



Experimental Syntax: Design, Analysis, and Application

Jon Sprouse
University of Connecticut

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1: Design

Section 2: Analysis

Section 3: Application

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

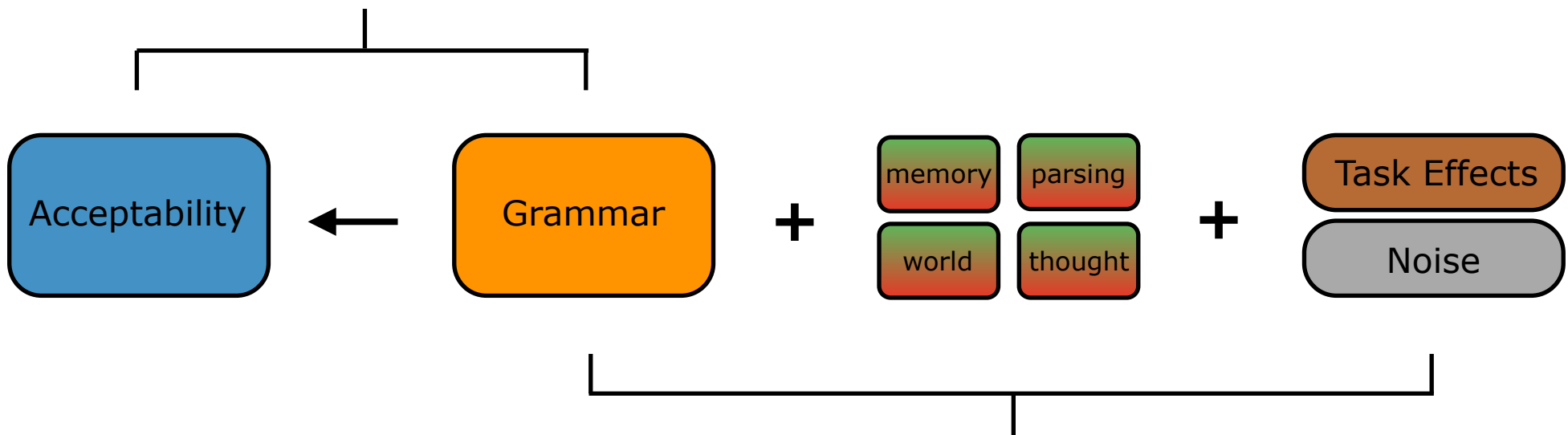
Section 3:
Application

What is an experiment?

The experimental method:

Manipulate one variable to elicit a change in a second variable. The goal is to establish a causal relationship between the first variable and the second.

















In syntax, the **causal relationship** we want to establish is between grammatical properties (structures, features, etc) and acceptability judgments.



The point of an experiment is to demonstrate this relationship while controlling for any possible **confounds** - effects that either (i) **obscure** the relationship you want to establish, or (ii) **create the illusion** of a relationship where none exists. Confounds can come from the grammar, other cognitive systems that influence acceptability judgments, the judgment task itself, and noise.

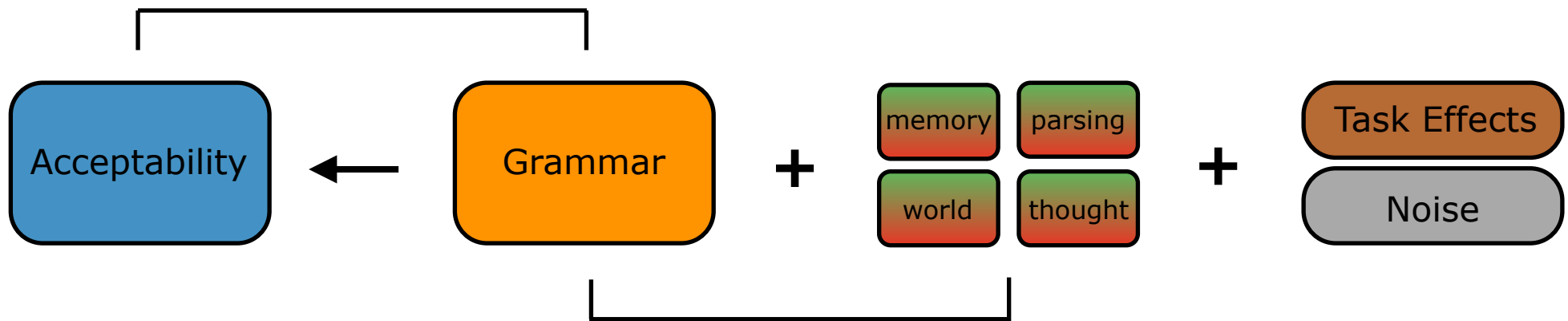
Traditional vs. Formal

The traditional approach to data collection in syntax is already experimental. Syntacticians already know how to construct an experiment. So we need to be clear about why we want to use formal experiments instead.

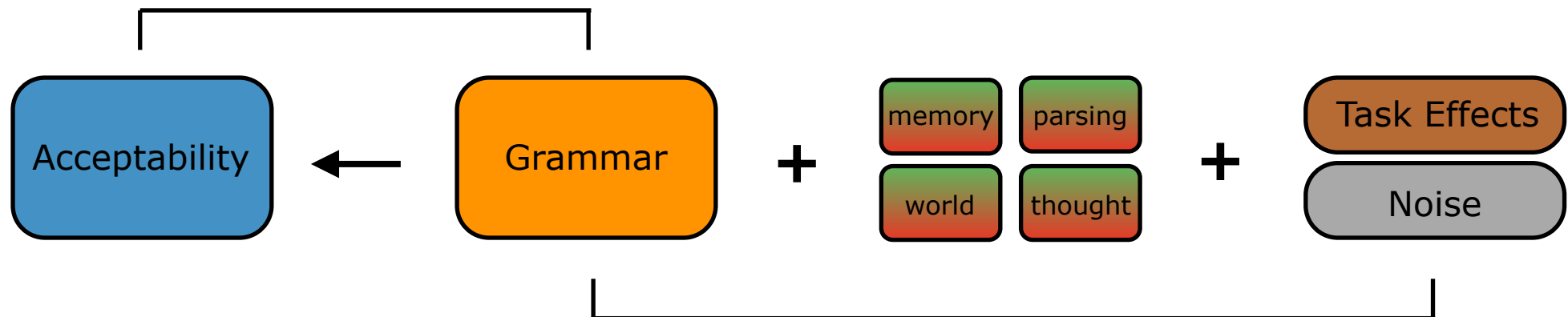
	Traditional	Formal
1. Establish a linking hypothesis		
2. Create conditions		
3. Create items		
4. Order the items for presentation		
5. Choose a task		
6. Recruit participants		
7. Explore/Analyze the results		
8. Report the results to others		

Traditional vs. Formal

Theoretical syntacticians are trained to design experiments that demonstrate the **causal relationship of interest** while controlling for **grammatical confounds** and **other cognitive confounds**.



Experimental syntax simply wants to add methods for controlling task effects and quantifying noise to this.



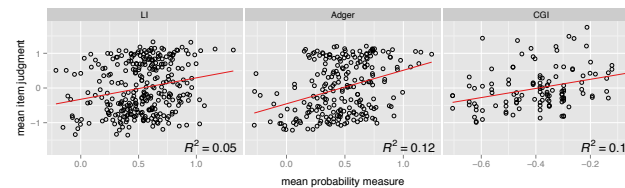
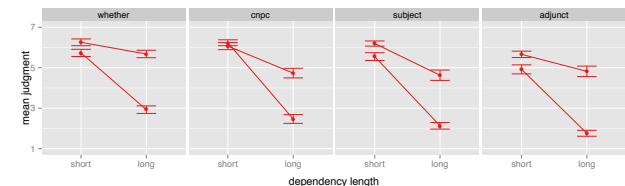
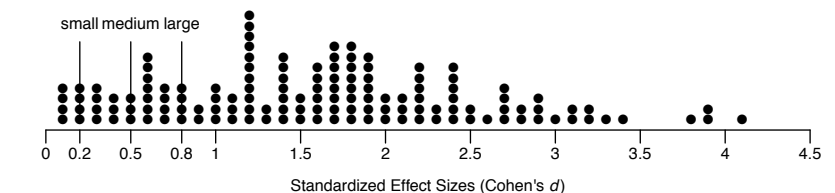
But what if you aren't worried about task effects and noise?

Comparisons of **traditional experiments** and **formal experiments** have yielded nearly identical results. This suggests that traditional methods haven't been led astray by **task effects** or noise.



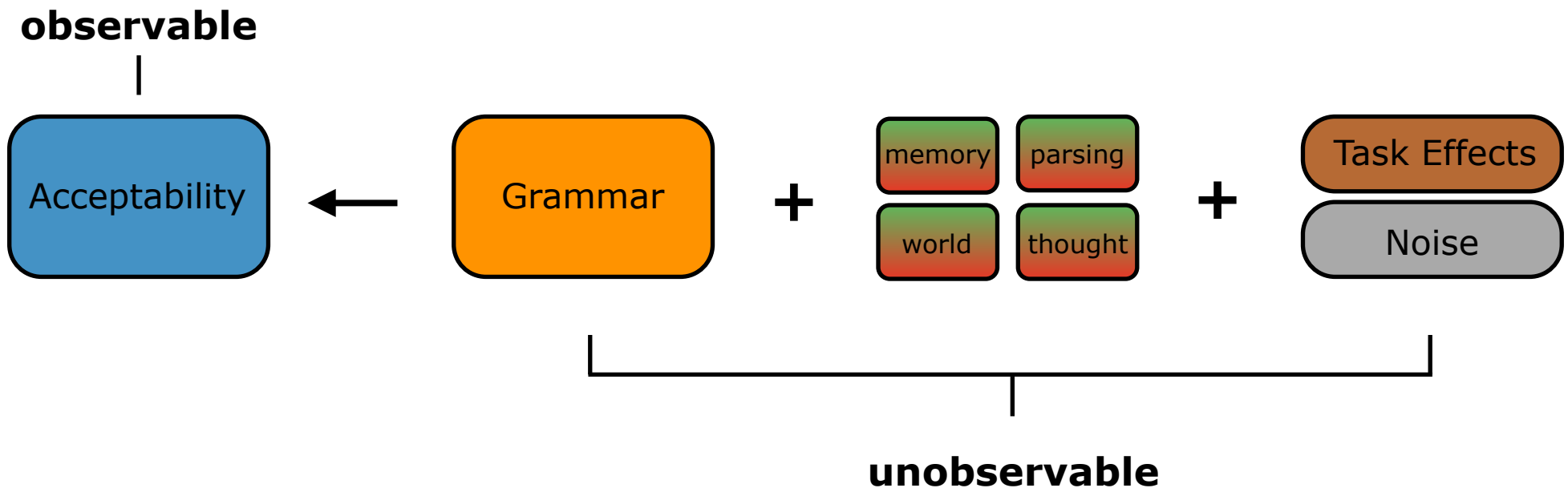
Three potential advantages of formal experiments:

1. Study smaller effect sizes.
2. Use more complicated designs.
3. Derive numbers.



All experiments need a linking hypothesis

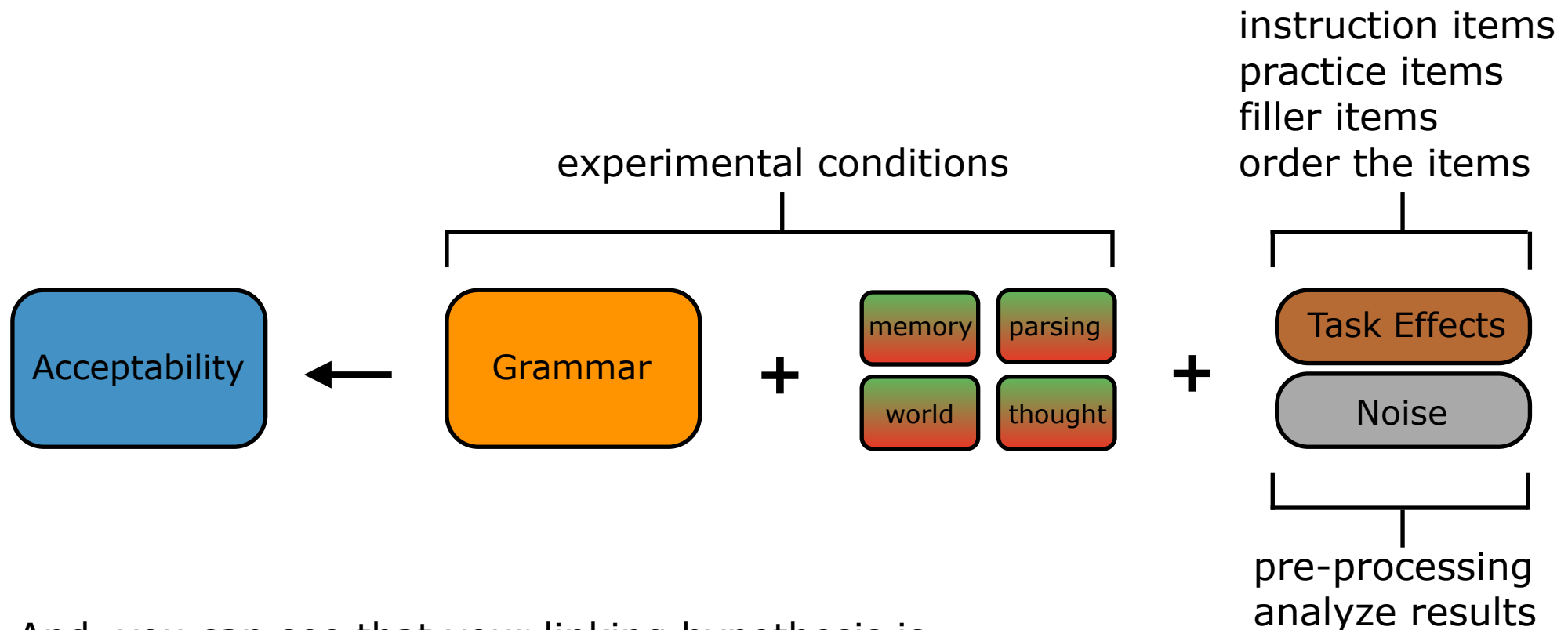
A **linking hypothesis** postulates a link between an observable measure and an unobservable construct. Linking hypotheses can't be directly tested (because one side of the link is unobservable). We decide that a linking hypothesis is reasonable by adopting it for a while, constructing a theory from the resulting data, and deciding if that theory seems reasonable.



The linking hypothesis for acceptability judgments is whatever our theory of acceptability judgments is. At the very least, we assume that grammatical status affects acceptability. This lets us use acceptability to study grammar.

Linking hypotheses are helpful for framing

Linking hypotheses can be useful as more than just a background assumption. Once you make your hypothesis explicit, you can begin to see how the different parts of your experiment are designed to minimize **confounds**.



$$\text{Acceptability} = \text{grammaticalFactors} + \text{processingFactors} + \text{taskFactors} + \text{noise}$$

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

Section 3:
Application

Conditions: Treatments and Controls

The terminology in experiments is borrowed from medicine.

A **treatment**, or **treatment condition**, is something that you do to the participant to see what happens. In syntax, a treatment would be the presentation of a sentence with a specific set of syntactic properties. We want to see how the treatment affects the participant's acceptability judgments.

A **control condition** is a lack of treatment. In syntax, this would be the presentation of a sentence without the specific set of syntactic properties under investigation. The control condition serves as a baseline to help rule out other explanations (**confounds**) for any **treatment effect** that we see.

So, the **conditions** in your experiment are the sentence types that you are going to present to your participants.

N.B. - Because medical terminology sounds strange in cognitive science, the term "treatment" is rarely used. We typically just talk about the conditions of the experiment. People sometimes use the term "target condition" or "experimental conditions", which can mean treatment conditions or treatment & control conditions, depending on the context. (It is often used to distinguish these conditions from "filler" conditions.)

Creating a treatments and controls

The fundamental logic in the creation of treatments and controls is that you want to create a **minimal pair**: a treatment condition and a control condition that vary by one property (the syntactic manipulation of interest).

$$\text{Acceptability}_T = \text{treatment} + \text{processingFactors} + \text{taskFactors} + \text{noise}$$

$$\text{Acceptability}_C = \text{processingFactors} + \text{taskFactors} + \text{noise}$$

If we subtract the acceptability of the two conditions, all that remains is the treatment effect.

$$\text{Acceptability}_T = \text{treatment} + \text{processingFactors} + \text{taskFactors} + \text{noise}$$

$$- \text{Acceptability}_C = \text{processingFactors} + \text{taskFactors} + \text{noise}$$

$$\text{Acceptability}_{T-C} = \text{treatment}$$

Pairs work in many simple cases

The first design you should consider is a simple pairwise comparison: one treatment condition and one control condition.

treatment: The children **is** tired.

control: The children are tired.

Acceptability_T = **treatment** + processingFactors + taskFactors + noise

agreement declarative

length (4 words)

— Acceptability_C = processingFactors + taskFactors + noise

declarative


length (4 words)

Acceptability_{T-C} = **agreement violation**

What about when pairs aren't enough?

Let's say you were interested in studying Whether Island effects in English, so you construct the obvious treatment condition - a whether island violation:

whether island: * What do you wonder [whether Jack stole ___]?



To determine what the best control will be, we first need to look at everything in the sentence that could affect acceptability. We can again use our linear equation for this:

Acceptability = grammaticalFactors + processingFactors + taskFactors + noise

island violation

wh-movement (2 clauses)

whether clause

length (7 words, 2 clauses)

To reiterate, the best control would be a sentence that has all of the same properties, except for the island violation (a minimal pair).

Let's try a couple of potential controls

whether island: * What do you wonder [whether Jack stole ___]?

that-control: What do you think [that Jack stole ___]?

Acceptability_T = island violation + wh-movement + whether clause + 7-words

– Acceptability_C = wh-movement + that clause + 7-words

Acceptability_{T-C} = island violation + (whether clause - that clause)

So this would not be a good control, because the effect that it shows us is confounded: it includes the difference between whether and that clauses.

Let's try a couple of potential controls

whether island: * What do you wonder [whether Jack stole ___]?

whether-control: Who ___ wonders [whether Jack stole a necklace]?

Acceptability_T = island violation + wh-movement-long + whether clause

– Acceptability_C = wh-movement-short + whether clause

Acceptability_{T-C} = island violation + (wh-long - wh-short)

So this would not be a good control, because the effect that it shows us is confounded: it includes the difference between long and short wh-movement.

The problem and its solution

The problem we face is an empirical one. We can't take away the island violation and keep everything else constant.

To take away the island violation, we either need to take away the whether clause, or we need to take away the long-distance dependency.

So the solution is to have multiple controls. But we have to be clever about this. We have to construct the controls in such a way that all of the effects that we want to eliminate will subtract out in the end. We can do this with four conditions:

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

This is called a crossed factorial design

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

Factor: A dimension/property that you can manipulate

The factors above are STRUCTURE and DEPENDENCY-LENGTH. Factors are often written in small caps in papers.

Level: The values that a factor can take

The values for STRUCTURE are non-island and island; the values for DEPENDENCY-LENGTH are short and long.

This is called a **crossed design** because every level of each factor is combined with every level of the other factors.

The subtraction logic

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

$$\begin{array}{rcl}
 \text{Acceptability}_4 & = & \text{violation} + \text{long} + \text{island} \\
 - \text{Acceptability}_2 & = & \text{long} + \text{non-island} \\
 \hline
 \text{Acceptability}_{4-2} & = & \text{violation} + (\text{island} - \text{non-island})
 \end{array}$$

$$\begin{array}{rcl}
 \text{Acceptability}_3 & = & \text{short} + \text{island} \\
 - \text{Acceptability}_1 & = & \text{short} + \text{non-island} \\
 \hline
 \text{Acceptability}_{3-1} & = & (\text{island} - \text{non-island})
 \end{array}$$

The subtraction logic

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

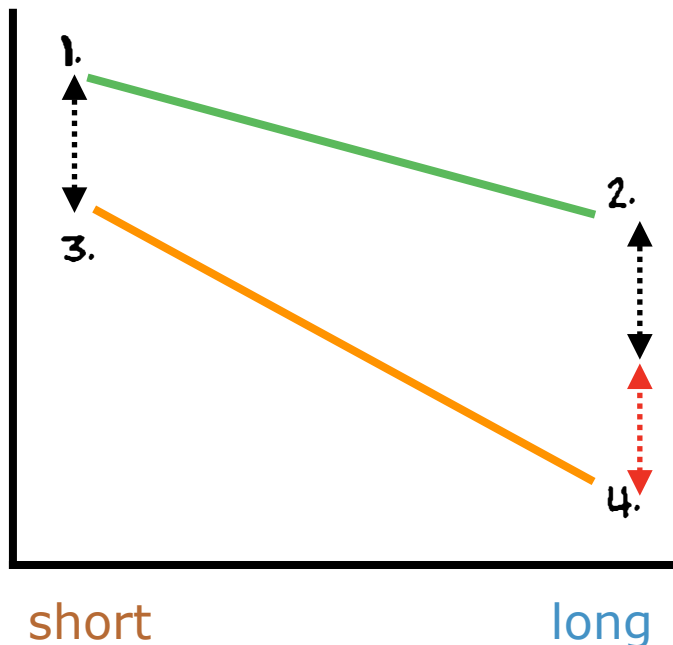
$$\begin{array}{rcl}
 & \text{Acceptability}_{4-2} = \text{violation} & + (\text{island} - \text{non-island}) \\
 - & \text{Acceptability}_{3-1} = & (\text{island} - \text{non-island}) \\
 \hline
 & \text{Acceptability}_{\text{DD}} = \text{violation} &
 \end{array}$$

The final step is to subtract the two differences (that we calculated in step one) from each other. This is called a **differences-in-differences** score.

Critically, the crossed factorial design and the two-step subtraction logic allows us to isolate the violation despite its co-occurrence with the two other factors!

The visual logic of differences-in-differences

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long



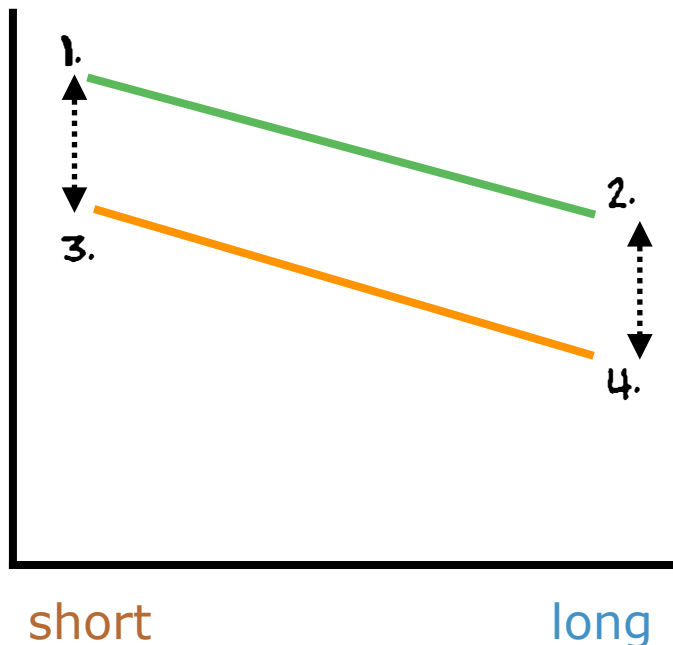
$$\begin{aligned} \text{Acceptability}_{4-2} &= \text{violation} + (\text{isl} - \text{non-isl}) \\ - \text{Acceptability}_{3-1} &= (\text{isl} - \text{non-isl}) \end{aligned}$$

$$\text{Acceptability}_{\text{DD}} = \text{violation}$$

If there is a violation present, the two lines will **not be parallel** because the two differences are different sizes.

The visual logic of differences-in-differences

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

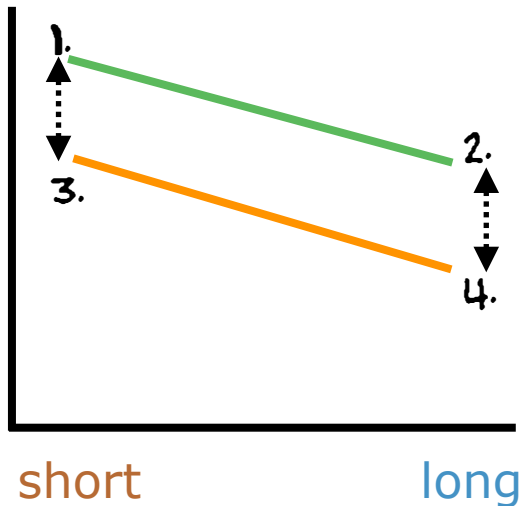


$$\begin{aligned}
 & \text{Acceptability}_{4-2} = + (\text{isl} - \text{non-isl}) \\
 - & \text{Acceptability}_{3-1} = (\text{isl} - \text{non-isl}) \\
 \hline
 & \text{Acceptability}_{DD} =
 \end{aligned}$$

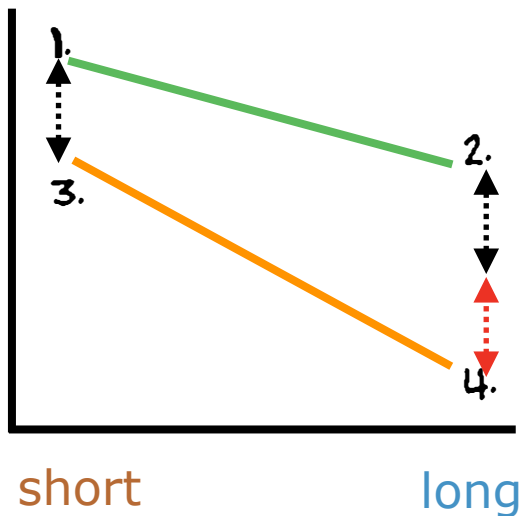
If there is no violation present, the two lines will be **parallel** because the two differences are the same size.

The logic of differences-in-differences

It is important to be clear about what you can and cannot interpret from these two patterns of results.



If the lines are parallel, there is no evidence of a violation at work. This is because we designed our experiment so that we could control for the effects of our two factors, isolating the effect of the violation (to condition 4).



If the lines are not parallel, there is evidence that something is affecting acceptability above and beyond our two factors. This could be a violation (or it could be something else, like a processing effect that arises from the interaction of the two factors). One way to think about this is that non-parallel lines is necessary, but not sufficient, for establishing the existence of a violation effect.

This is called a 2x2 design

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

The terminology here is that each digit represents a factor. Since there are two digits (2 and 2), there are two factors in this design.

And the value of each digit represents the number of levels for that factor. Both factors in this design have two levels, so both digits are 2.

A 2x3 design would have two factors, one with two levels and one with three.

A 2x2x2 design would have three factors, each with two levels.

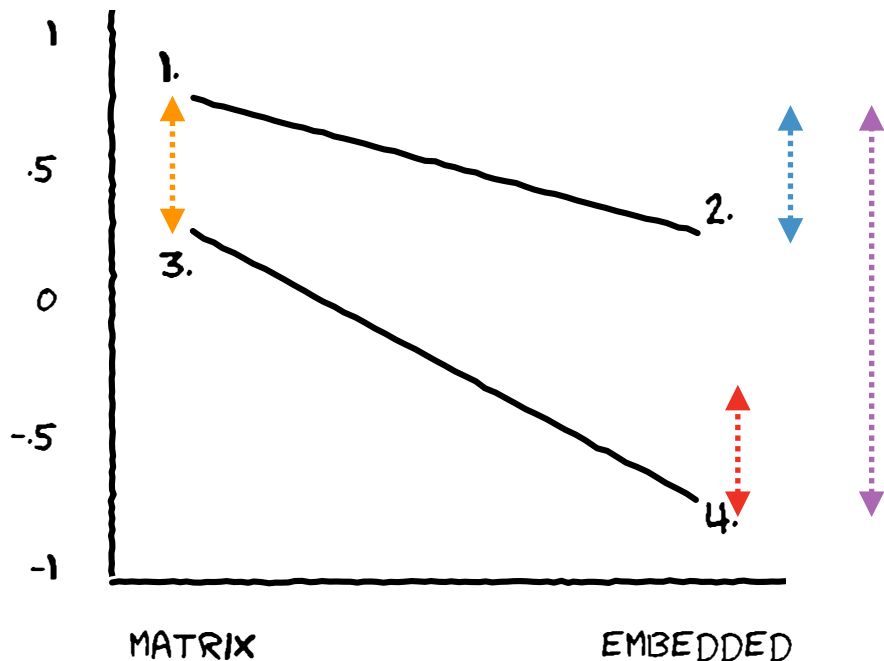
A 2x2 design can quantify 3 effects

isolate
structure

all three

1. Who ___ thinks that Jack stole the necklace?
2. What do you think that Jack stole ___?
3. Who ___ wonders whether Jack stole the necklace?
4. *What do you wonder whether Jack stole ___?

isolate
dependency



dependency effect (1-2)

complexity effect (1-3)

+	violation effect	+	X
<hr/>		<hr/>	
	dep + struc + X		(1-4)

This means that we can specifically state the contribution of these three effects. I am treating condition 1 as a neutral or baseline condition, and calculating the effects relative to condition 1.

Just to clarify, the DD score is part of the calculation of the three effects.

	Structure	Dependency
1. Who ___ thinks that Jack stole the necklace?	non-island	short
2. What do you think that Jack stole ___?	non-island	long
3. Who ___ wonders whether Jack stole the necklace?	island	short
4. *What do you wonder whether Jack stole ___?	island	long

The three effects:

$$\begin{array}{rcl}
 (1-4) & = & (1-2) + (1-3) + \text{violation} \\
 - (1-2) & & (1-2) \\
 \hline
 (2-4) & = & (1-3) + \text{violation} \\
 - (1-3) & & (1-3) \\
 \hline
 \text{DD score: } & (2-4) - (1-3) = & \text{violation}
 \end{array}$$

The DD scores earlier were shortcuts to directly calculate the violation effect.

But if we want, we can calculate all three effects (structure, dependency, and violation). The two are equivalent, they just reveal different bits of information.

2x2 designs can control potential confounds (as long as they don't interact with the factors)

		Structure	Dependency
1.	Who ___ thinks that Jack stole the necklace?	non-island	short
2.	What do you think that Jack stole ___?	non-island	long
3.	Who ___ wonders whether Jack stole the necklace?	island	short
4.	*What do you wonder whether Jack stole ___?	island	long

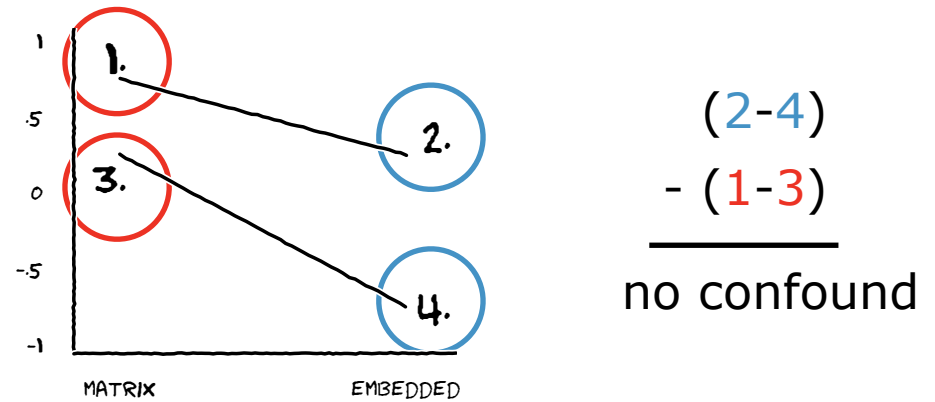
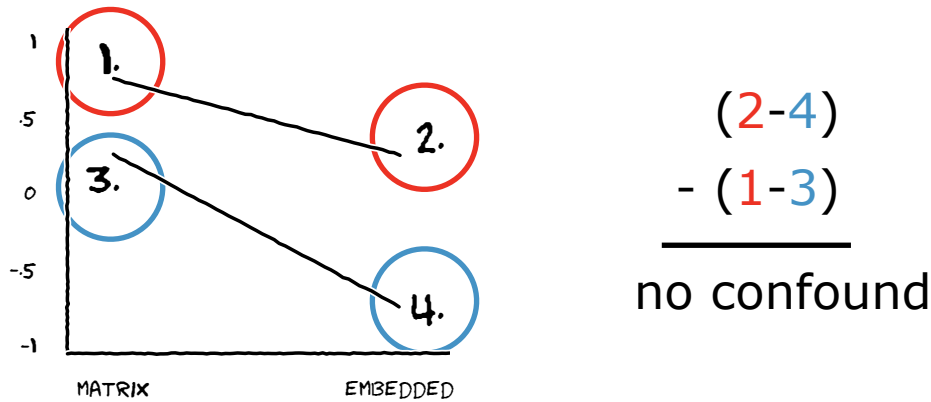
Let's say you are looking at this design, and you notice that the wh-word varies by condition. Should you be worried that this is a confound?

The answer is that it depends on how the variability is distributed around the DD subtraction. If the variability subtracts out, then everything is fine. If it doesn't, then this is a confound.

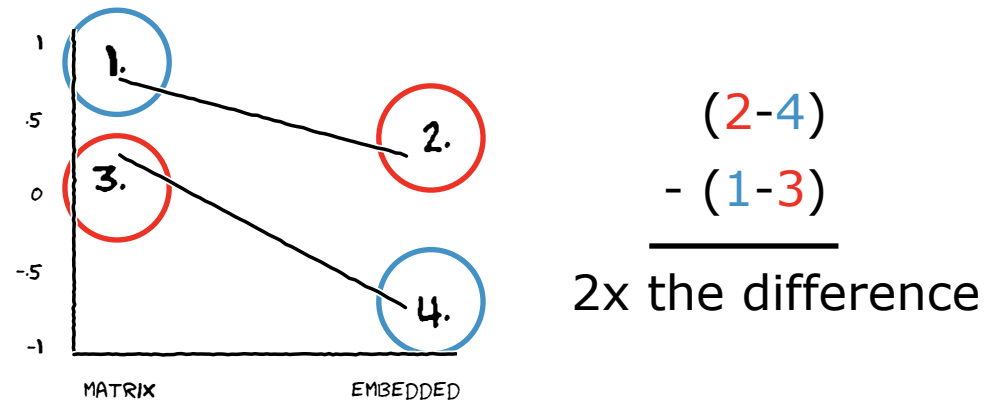
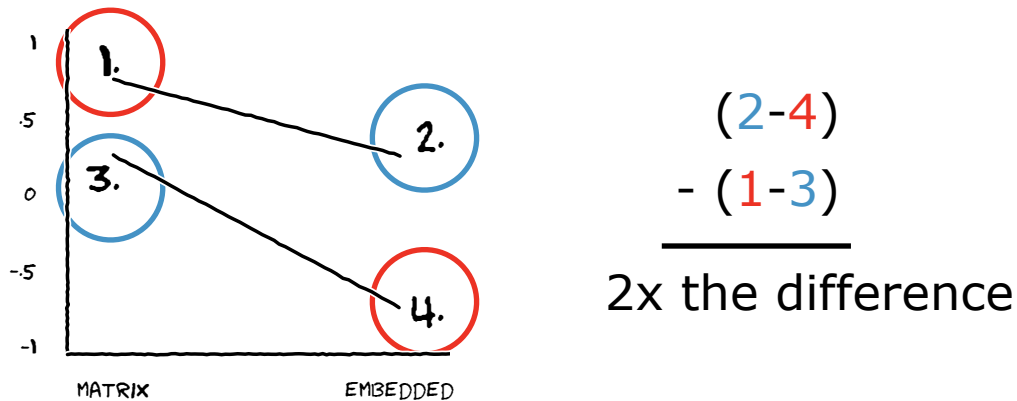
This is the hidden power of 2x2 designs. Although they can only **quantify 3 effects** (here length, structure, and violation), they can **control for an unlimited number of other effects as long as those effects don't interact with any of the factors**. By control, I mean they let you subtract them out so that they aren't a confound.

2x2 designs are very powerful

There is a quick way to see whether the potential confound will be subtracted out. Basically, if you imagine the conditions as forming a box, the confound will subtract out if it is distributed as if it is on the same side of the box.



If it is distributed on opposite corners, it won't subtract out, it will add or subtract to the appearance of the violation effect.



2x2 should be your “go to” design

If pairwise comparisons don't work for your design, then most likely a 2x2 design will work.

The only reason to move up to a larger design (2x3, 2x2x2, etc) is theoretical: if you are working with an effect that has several levels (e.g. 2x3), if you are working with two violation effects at once (e.g., 2x2x2, which is really just two 2x2 designs put together, one for each effect), or you need to quantify more than two factors, then you will need to move up. But most of the time we only work on one effect at a time, and most effects are either present/absent (not multi-leveled). So 2x2 designs are very likely to work with most syntactic effects.

Exercise 1: Creating 2x2 designs for island effects

The file exercise.1.xlsx is setup for you to create a 2x2 design for each of four different island effects (so, 4 different 2x2 designs). Your job is to create one example sentence for each condition in the designs (16 total sentences).

Be sure that any potential confounds are distributed such that they will be subtracted out.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

Section 3:
Application

Linguistics tends to use repeated measures

Repeated Measures



condition 1 condition 2

Repeated Measures:

If each participants sees every condition, we call it repeated measures. It is also called a **within-subjects** design.

Independent Measures:

If each participants sees only one condition, we call it independent measures. It is also called a **between-subjects** design.

Independent Measures



condition 1 condition 2

Linguistics tends to use repeated measures

Repeated Measures



Requires fewer participants

Individual differences between participants is not a confound

Increased statistical power

Interaction of two conditions is a potential confound

Independent Measures



Requires more participants

Individual differences between participants is a possible confound

Decreased statistical power

Interaction of two conditions is impossible

There are four types of items to create

After you have designed your conditions, the next step is to actually make the items that will go in your experiment. There are four types of items that you will need to construct:

Instruction items: These are the items that appear in your instructions. The goal there is to illustrate the task, and if necessary, [anchor](#) the response scale.

Practice items: These are items that occur at the beginning of the experiment. They help to familiarize the participant with the task. They are typically not analyzed in any way. They can be marked as separate (announced) or just part of the experiment (unannounced).

Experimental items: These are your treatment and control conditions.

Filler items: These are items that you add to the experiment for various reasons: filling out the scale, hiding the experiment's purpose, and balancing types of items.

Instruction items

The number and type of instruction items depends on your task.

If the task is a **scale task** with an **odd number of points** (e.g, 7-point scale), I recommend three instruction items: one at the bottom of the scale, one at the top, and one in middle. Here are three that I use. They were pre-tested in my massive LI replication study:

	LI-Mode	LI-Mean
The was insulted waitress frequently.	1	1
Tanya danced with as handsome a boy as her father.	4	4
This is a pen.	7	7

If the scale has an even number of points, you would probably just use two: the bottom and top of the scale.

If the task is yes/no, you might use three: a clear yes, a clear no, and one in between.

If the task is forced-choice, you might use 3 pairs: a pair with a large difference, a pair with a medium difference, and one with a small difference.

Practice items

Practice items give participants a chance to work out any bugs before they respond to items that you actually care about (the experimental items).

For [scale tasks](#), practice items give participants a chance to see the full range of variability in acceptability, so that they can use the scale appropriately. So in scale tasks, it is important to have practice items that span the range of acceptability. Here are 9 that I have pre-tested in the LI study. One for each point on a 7-point scale, plus one more for each endpoint.

	LI-Mode	LI-Mean
She was the winner.	7	7.00
Promise to wash, Neal did the car.	1	1.31
The brother and sister that were playing all the time had to be sent to bed	4	3.91
The children were cared for by the adults and the teenagers	6	6.08
Ben is hopeful for everyone you do to attend.	2	2.00
All the men seem to have all eaten supper	5	4.92
They consider a teacher of Chris geeky.	3	3.09
It seems to me that Robert can't be trusted.	7	6.92
There might mice seem to be in the cupboard.	1	1.25

Practice items

For non-scale tasks, the rationale behind the practice items might be different.

For **yes/no tasks**, you may want to give a mix of clear yes's, clear no's, and intermediate sentences, so that participants can sharpen their own internal boundary.

For **forced-choice tasks**, you may want to include a mix of large differences, small differences, and medium differences, so that participants can practice identifying each size of difference.

Announced practice is when you clearly indicate in the experiment that the items are practice items. This signals to the participants that it is ok to make mistakes. Announced practice is typical in psycholinguistic experiments, because it gives participants a chance to ask questions of the experimenter.

Unannounced practice is when the practice items simply appear as part of the main experiment. This is appropriate if the task is relatively intuitive, such that participants won't have questions. This is what I do with all of my judgment studies.

I typically present the (unannounced) practice items in the same order for all participants. You could also counterbalance the order (more on this later).

Experimental items

Here is a starting set of experimental items for the whether island experiment we started to construct in the previous section. Let's use these to see the issues that arise in creating experimental items.

Condition 1: non-island short

1. Who ___ thinks that Jack stole the car?
2. Who ___ thinks that Amy chased the bus?
3. Who ___ thinks that Dale sold the TV?
4. Who ___ thinks that Stacey wrote the letter?

Condition 3: island short

1. Who ___ wonders whether Jack stole the car?
2. Who ___ wonders whether Amy chased the bus?
3. Who ___ wonders whether Dale sold the TV?
4. Who ___ wonders whether Stacey wrote the letter?

Condition 2: non-island long

1. What do you think that Jack stole ___?
2. What do you think that Amy chased ___?
3. What do you think that Dale sold ___?
4. What do you think that Stacey wrote ___?

Condition 4: island long

1. What do you wonder whether Jack stole ___?
2. What do you wonder whether Amy chased ___?
3. What do you wonder whether Dale sold ___?
4. What do you wonder whether Stacey wrote ___?

Experimental items - Lexically matched sets

The first thing to note is that the items are created in **lexically matched sets**. The idea here is that the only thing you want varying between conditions is the syntactic manipulation. So, to the extent possible, you use the same lexical items in all 4 conditions.

Condition 1: non-island short

1. Who ___ thinks that Jack stole the car?
2. Who ___ thinks that Amy chased the bus?
3. Who ___ thinks that Dale sold the TV?
4. Who ___ thinks that Stacey wrote the letter?

Condition 2: non-island long

1. What do you think that Jack stole ___?
2. What do you think that Amy chased ___?
3. What do you think that Dale sold ___?
4. What do you think that Stacey wrote ___?

Condition 3: island short

1. Who ___ wonders whether Jack stole the car?
2. Who ___ wonders whether Amy chased the bus?
3. Who ___ wonders whether Dale sold the TV?
4. Who ___ wonders whether Stacey wrote the letter?

Condition 4: island long

1. What do you wonder whether Jack stole ___?
2. What do you wonder whether Amy chased ___?
3. What do you wonder whether Dale sold ___?
4. What do you wonder whether Stacey wrote ___?

This helps minimize confounds in the experiment. The only lexical confound left is if the syntactic manipulation interacts with the lexical items.

Experimental items - variability

The second thing to note is that the variability in the items is tightly controlled. In this case, I primarily varied content items, keeping functional items the same. There is a tension between variability and control. I tend to err on the side of control so that there are fewer chances for confounds.

Condition 1: non-island short

1. Who ___ thinks that Jack stole the car?
2. Who ___ thinks that Amy chased the bus?
3. Who ___ thinks that Dale sold the TV?
4. Who ___ thinks that Stacey wrote the letter?

Condition 3: island short

1. Who ___ wonders whether Jack stole the car?
2. Who ___ wonders whether Amy chased the bus?
3. Who ___ wonders whether Dale sold the TV?
4. Who ___ wonders whether Stacey wrote the letter?

Condition 2: non-island long

1. What do you think that Jack stole ___?
2. What do you think that Amy chased ___?
3. What do you think that Dale sold ___?
4. What do you think that Stacey wrote ___?

Condition 4: island long

1. What do you wonder whether Jack stole ___?
2. What do you wonder whether Amy chased ___?
3. What do you wonder whether Dale sold ___?
4. What do you wonder whether Stacey wrote ___?

However, variability is also important. When items vary, you can begin to see how well the effect generalizes across lexical items.

How much variability do you want?

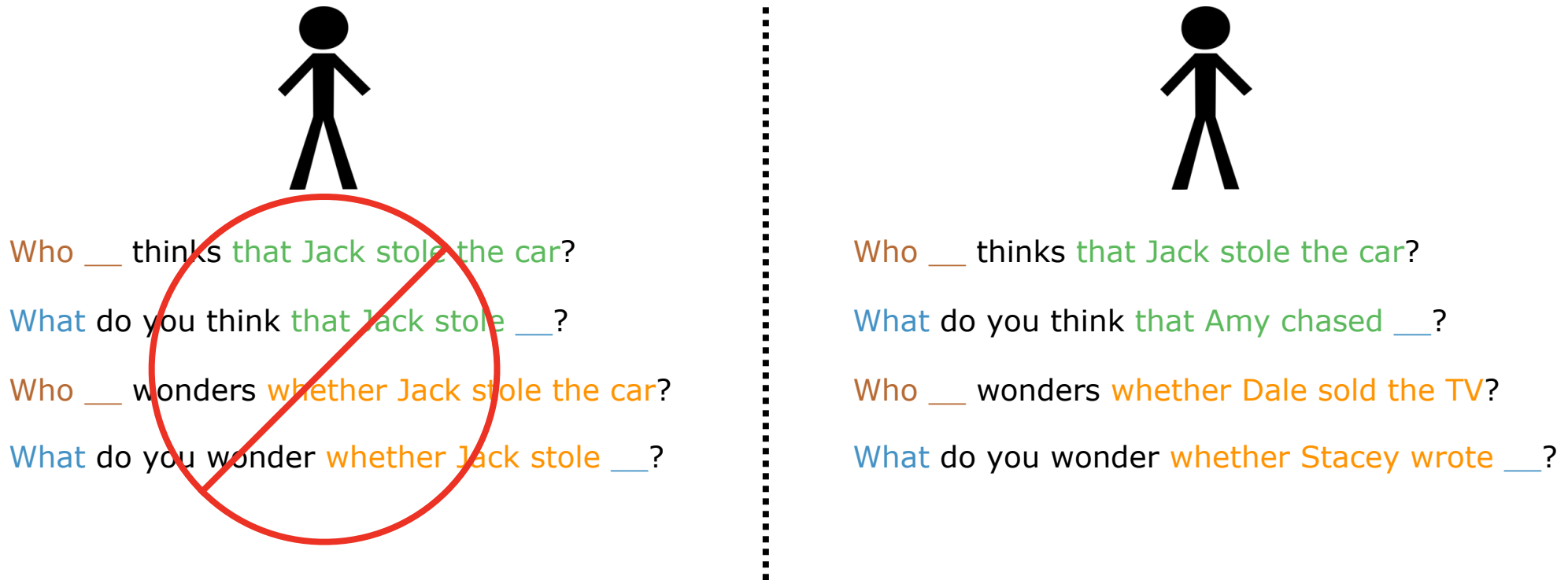
There is no set principle for how much variability you need. It will depend on the number of viable lexical items for the constructions you are testing, the likelihood that lexical items are driving your effect, and the potential confounds that could be introduced by lexical items.

What I can tell you is my approach to this:

1. I try to make every item in a single condition the same length. This means there are no extra PPs or clauses between items. Longer sentences often lead to lower ratings, so length is a potential confound.
2. It is often the case that some of the lexical items **cannot** vary because of the nature of the conditions. For example, in whether-islands you will always have **whether** in the embedded clause.
3. I try to be consistent about the use and position of pronouns versus nouns. The reason for this is that pronouns and nouns are processed differently; in fact, different pronouns are processed differently.
4. Everything else is a potential point of variation, as long as the lexical items have the relevant properties (e.g., subcategorization frames).

Lexical matching and repeated measures

In repeated measures designs (each participant sees every condition), lexical matching can be a problem. You don't want one participant to see the same lexical material in each condition, because then they might overlook the syntactic manipulation:



This leads to a straightforward relationship between (i) the number of conditions, (ii) the number of judgments per condition each participant will give, and (iii) the number of items that you need to make per condition.

Experimental items - number

If C is the number of conditions in your experiment, and O is the number of judgments (observations) each participant will give per condition, and I is the number of items per condition that you need to construct, then **$I = C \times O$** .

Condition 1: non-island short

1. Who ___ thinks that Jack stole the car?
2. Who ___ thinks that Amy chased the bus?
3. Who ___ thinks that Dale sold the TV?
4. Who ___ thinks that Stacey wrote the letter?

Condition 3: island short

1. Who ___ wonders whether Jack stole the car?
2. Who ___ wonders whether Amy chased the bus?
3. Who ___ wonders whether Dale sold the TV?
4. Who ___ wonders whether Stacey wrote the letter?

Condition 2: non-island long

1. What do you think that Jack stole ___?
2. What do you think that Amy chased ___?
3. What do you think that Dale sold ___?
4. What do you think that Stacey wrote ___?

Condition 4: island long

1. What do you wonder whether Jack stole ___?
2. What do you wonder whether Amy chased ___?
3. What do you wonder whether Dale sold ___?
4. What do you wonder whether Stacey wrote ___?

Here I've created 4 items per condition, so it must be the case that I only want 1 judgment per participant per condition. If I wanted 2, I'd need 8 items...

Filler items

Filler items are not strictly necessary. But there are three reasons to add filler items to your experiment. If you are worried about any of these issues, then you need fillers items. (As a practical matter, most reviewers expect filler items, so it is easier to include them if you can.)

Fill out the response scale:

Participants tend to keep track of how often they use each response option. If some options aren't being used, they may try to use them even if they aren't appropriate. Well-designed fillers can make sure that every response option is used an equal number of times.

Balancing other properties:

Some properties of your experimental items might be particularly salient, especially if you are studying a particular construction (wh-movement, ellipsis, etc). Fillers allow you to include other constructions, so that participants are less likely to be impacted by the salience of those features.

Hiding your intent:

Relatedly, some experimenters worry that participants might respond differently if they know the purpose of the experiment. Fillers can help disguise that purpose, by hiding the experimental items among other items.

Filler items

There is no easy formula for calculating the number of filler items that you need. The answer is that you need as many as you need to achieve your goals.

What I can tell you is that there are “rules of thumb” in the field that reviewers often look for. These can be violated if the science requires it, but in general, if you can follow these rules, it will make your reviewing experience easier.

1. The ideal ratio of filler items to experimental items is **2:1 or higher**. That means that 2/3 of the items that a participant sees are filler items, and 1/3 are experimental items.
2. The minimum ratio is 1:1. This means that half of the items that a participant sees are filler items.
3. Experimental items from a one experiment can serve as fillers for the experimental items from another experiment. So you can kill multiple birds with one stone. But the items need to be sufficiently distinct, and they still need to satisfy general filler properties (balancing responses, etc).

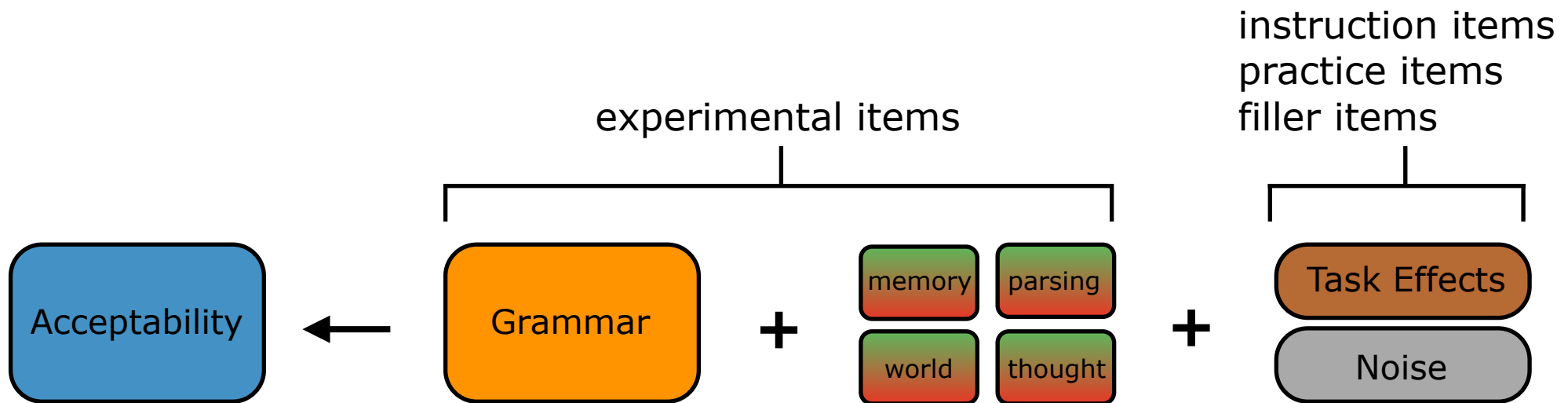
Fillers: Here is a set of filler items that I have constructed for an experiment with 8 experimental items (2 each of 4 conditions).

	LI-Mode	LI-Mean
Mike prefers tennis because Jon baseball.	1	1.17
Jenny cleaned her sister the table.	1	1.17
There had all hung over the fireplace the portraits by Picasso.	2	2.17
Lilly will dance who the king chooses.	2	2.00
The specimen thawed to study it more closely.	3	3.08
With that announcement were many citizens denied the opportunity to protest.	3	3.08
There is likely a river to run down the mountain.	4	4.15
Richard may have been hiding, but Blake may have done so too.	4	4.17
The ball perfectly rolled down the hill.	5	5.00
Lloyd Weber musicals are easy to condemn without even watching.	5	4.93
There are firemen injured.	6	6.00
Someone better sing the national anthem.	6	6.00
Laura is more excited than nervous.	7	6.92
I hate eating sushi.	7	6.92

What have we been controlling?

The construction of experimental items is primarily about controlling for **grammar confounds** and **other cognitive confounds**.

The construction of instruction items, practice items, and filler items is primarily about controlling for **task effects**.



Hands on practice

Exercise 2: 2x2 item practice

The file `exercise.2.xlsx` contains the 2x2 designs for the four island effects from exercise 1. Your job is to create four items for each condition (a total of 64 sentences). Be sure to create variability where you can, while still keeping the items tightly controlled.

Anchor, practice, filler items (not an exercise)

The file `anchor.practice.instruction.items.xlsx` includes the instruction, practice, and filler items that we discussed here. There is nothing you need to do. These just exist for you to use in your future experiments if you want.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

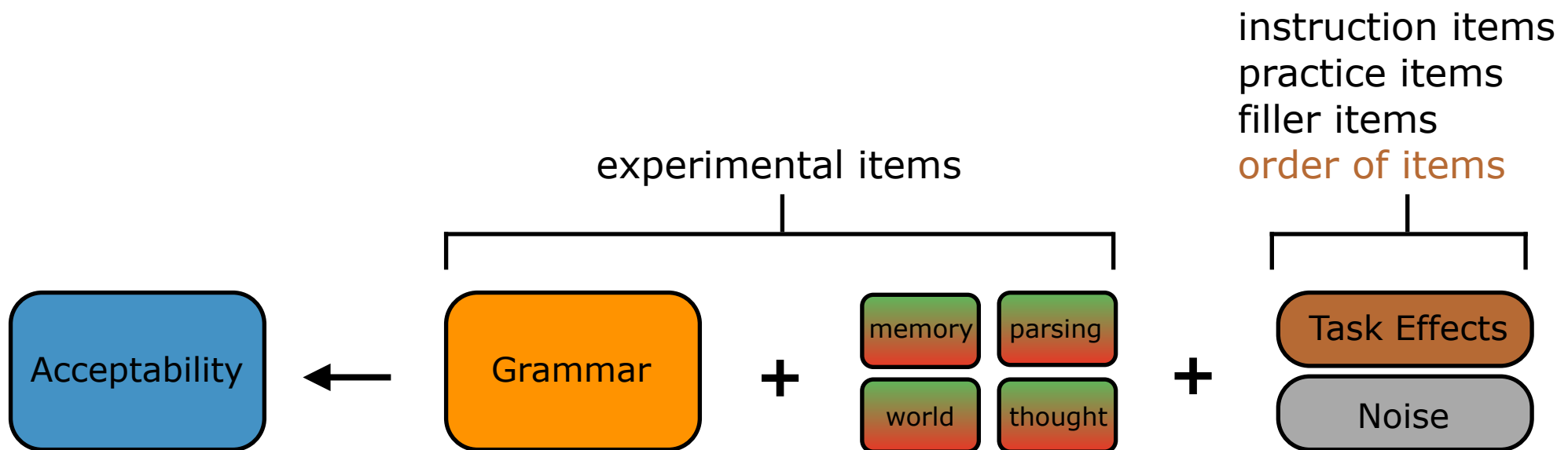
Section 3:
Application

This section is about task effects

The construction of experimental items is primarily about controlling for **grammar confounds** and **other cognitive confounds**.

The construction of instruction items, practice items, and filler items is primarily about controlling for **task effects**.

The arrangement of items into an actual experiment is also primarily about controlling for **task effects**.



Assign meaningful codes to your items

Before we start to manipulate our target items, let's talk about **item codes**.

item codes: Meaningful codes that you assign to each of your items. These will help you quickly identify the properties of each item, and will play an important role in later data analysis.

Item codes should contain all of the information about an item, such as the name of its condition (if you are naming your conditions), the levels of its factors (if you have a factorial design), and the lexically-matched item-set (or lexicalization-set) number that it is. Here is how I like to create item codes:

subdesign.factor1.factor2.item-set-number

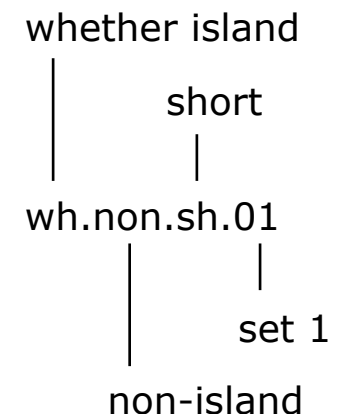
Condition 1: non-island short

wh.non.sh.01 Who thinks that Jack stole the car?

wh.non.sh.02 Who thinks that Amy chased the bus?

wh.non.sh.03 Who thinks that Dale sold the TV?

wh.non.sh.04 Who thinks that Stacey wrote the letter?



Assign meaningful codes to your items

wh.non.sh.01 Who __ thinks that Jack stole the car?
wh.non.sh.02 Who __ thinks that Amy chased the bus?
wh.non.sh.03 Who __ thinks that Dale sold the TV?
wh.non.sh.04 Who __ thinks that Stacey wrote the letter?

wh.non.lg.01 What do you think that Jack stole __?
wh.non.lg.02 What do you think that Amy chased __?
wh.non.lg.03 What do you think that Dale sold __?
wh.non.lg.04 What do you think that Stacey wrote __?

wh.isl.sh.01 Who __ wonders whether Jack stole the car?
wh.isl.sh.02 Who __ wonders whether Amy chased the bus?
wh.isl.sh.03 Who __ wonders whether Dale sold the TV?
wh.isl.sh.04 Who __ wonders whether Stacey wrote the letter?

wh.isl.lg.01 What do you wonder whether Jack stole __?
wh.isl.lg.02 What do you wonder whether Amy chased __?
wh.isl.lg.03 What do you wonder whether Dale sold __?
wh.isl.lg.04 What do you wonder whether Stacey wrote __?

Note that each item code is **unique** to that item. So they are unique identifiers.

However, each code captures all of the relationships among the items.

The global design is captured in the first part, the factors in the middle parts, and the lexical matching in the final part.

Using a separator like “.” makes it easy to pull this information apart in languages like R.

Divide items into lists

List: A list is a set of items that will be seen by a single participant. It is not yet ordered for presentation.

Let's assume that these are our 4 conditions. We've made 4 items per condition. We want each participant to see all 4 conditions, and 1 item per condition.

We don't want participants to see the same lexical material (because then they might not notice the differences). How many lists can we make at the same time?

wh.non.sh.01 Who __ thinks that Jack stole the car?

wh.non.sh.02 Who __ thinks that Amy chased the bus?

wh.non.sh.03 Who __ thinks that Dale sold the TV?

wh.non.sh.04 Who __ thinks that Stacey wrote the letter?

wh.non.lg.01 What do you think that Jack stole __?

wh.non.lg.02 What do you think that Amy chased __?

wh.non.lg.03 What do you think that Dale sold __?

wh.non.lg.04 What do you think that Stacey wrote __?

wh.isl.sh.01 Who __ wonders whether Jack stole the car?

wh.isl.sh.02 Who __ wonders whether Amy chased the bus?

wh.isl.sh.03 Who __ wonders whether Dale sold the TV?

wh.isl.sh.04 Who __ wonders whether Stacey wrote the letter?

wh.isl.lg.01 What do you wonder whether Jack stole __?

wh.isl.lg.02 What do you wonder whether Amy chased __?

wh.isl.lg.03 What do you wonder whether Dale sold __?

wh.isl.lg.04 What do you wonder whether Stacey wrote __?

Divide items into lists

The answer is that we can create 4 lists from this design.

We want each list to have all 4 conditions, but to have a different lexical item for each condition.

List 1	List 2	List 3	List 4
wh.non.sh.01	wh.non.sh.02	wh.non.sh.03	wh.non.sh.04
wh.non.lg.02	wh.non.lg.03	wh.non.lg.04	wh.non.lg.01
wh.isl.sh.03	wh.isl.sh.04	wh.isl.sh.01	wh.isl.sh.02
wh.isl.lg.04	wh.isl.lg.01	wh.isl.lg.02	wh.isl.lg.03

List 1

wh.non.sh.01 Who __ thinks that Jack stole the car?

wh.non.lg.02 What do you think that Amy chased __?

wh.isl.sh.03 Who __ wonders whether Dale sold the TV?

wh.isl.lg.04 What do you wonder whether Stacey wrote __?

List 3

wh.non.sh.03 Who __ thinks that Dale sold the TV?

wh.non.lg.04 What do you think that Stacey wrote __?

wh.isl.sh.01 Who __ wonders whether Jack stole the car?

wh.isl.lg.02 What do you wonder whether Amy chased __?

List 2

wh.non.sh.02 Who __ thinks that Amy chased the bus?

wh.non.lg.03 What do you think that Dale sold __?

wh.isl.sh.04 Who __ wonders whether Stacey wrote the letter?

wh.isl.lg.01 What do you wonder whether Jack stole __?

List 4

wh.non.sh.04 Who __ thinks that Stacey wrote the letter?

wh.non.lg.01 What do you think that Jack stole __?

wh.isl.sh.02 Who __ wonders whether Amy chased the bus?

wh.isl.lg.03 What do you wonder whether Dale sold __?

The analogy to Latin Squares

This design is often called a Latin Square design in experimental fields.

Latin Squares have been mathematical puzzles for centuries. Euler studied them using Latin characters, hence the name.

List 1	List 2	List 3	List 4
wh.non.sh.01	wh.non.sh.02	wh.non.sh.03	wh.non.sh.04
wh.non.lg.02	wh.non.lg.03	wh.non.lg.04	wh.non.lg.01
wh.isl.sh.03	wh.isl.sh.04	wh.isl.sh.01	wh.isl.sh.02
wh.isl.lg.04	wh.isl.lg.01	wh.isl.lg.02	wh.isl.lg.03

Latin Square (4 letters)

A	B	C	D
B	C	D	A
C	D	A	B
D	A	B	C

Latin Square Design (4 conditions)

	List 1	List 2	List 3	List 4
wh.non.sh	1	2	3	4
wh.non.lg	2	3	4	1
wh.isl.sh	3	4	1	2
wh.isl.lg	4	1	2	3

The number in the cells represent items numbers from the lexically-matched sets.

We only need one solution to the Latin Square

There are 576 possible solutions to a latin square of size 4, but we only need one. It is really easy to find a single solution to a latin square — simply increment the sequence by one step in each row, looping the sequence around when you get to the end. This will always give you a solution.

1	2	3	4
2	3	4	1
3	4	1	2
4	1	2	3

Once you have this solution memorized, you can see that it is the solution that I used to create the four lists for our experiment.

List 1	List 2	List 3	List 4
wh.non.sh.01	wh.non.sh.02	wh.non.sh.03	wh.non.sh.04
wh.non.lg.02	wh.non.lg.03	wh.non.lg.04	wh.non.lg.01
wh.isl.sh.03	wh.isl.sh.04	wh.isl.sh.01	wh.isl.sh.02
wh.isl.lg.04	wh.isl.lg.01	wh.isl.lg.02	wh.isl.lg.03

So all we need now is a quick way to create this latin square pattern for both our sentences and the item codes for our sentences.

Latin Squares - mild automation

You should probably create latin squares by hand for the first few experiments that you run, so that you can be sure that you really understand the pattern. But once you know the pattern, you should feel free to automate the process. Here is some mild automation using excel:

Step 1: list all items in order by condition

Step 2: add a list number next to each item based on a Latin Square design

21				
22	1	wh.non.sh.01	Who thinks that Paul stole the necklace?	
23	2	wh.non.sh.02	Who thinks that Matt chased the bus?	
24	3	wh.non.sh.03	Who thinks that Tom sold the television?	
25	4	wh.non.sh.04	Who thinks that Stacey wrote the letter?	
26	2	wh.non.lg.01	What does the detective think that Paul stole?	
27	3	wh.non.lg.02	What does the police officer think that Matt chased?	
28	4	wh.non.lg.03	What does the manager think that Tom sold?	
29	1	wh.non.lg.04	What does the soldier think that Stacey wrote?	
30	3	wh.isl.sh.01	Who wonders whether Paul stole the necklace?	
31	4	wh.isl.sh.02	Who wonders whether Matt chased the bus?	
32	1	wh.isl.sh.03	Who wonders whether Tom sold the television?	
33	2	wh.isl.sh.04	Who wonders whether Stacey wrote the letter?	
34	4	wh.isl.lg.01	What does the detective wonder whether Paul stole?	
35	1	wh.isl.lg.02	What does the police officer wonder whether Matt chased?	
36	2	wh.isl.lg.03	What does the manager wonder whether Tom sold?	
37	3	wh.isl.lg.04	What does the soldier wonder whether Stacey wrote?	
38				

Latin Squares - mild automation

You should probably create latin squares by hand for the first few experiments that you run, so that you can be sure that you really understand the pattern. But once you know the pattern, you should feel free to automate the process. Here is some mild automation using excel:

Step 1: list all items in order by condition

Step 2: add a list number next to each item based on a Latin Square design

Step 3: sort by the list number to create four lists

1	wh.non.sh.01	Who thinks that Paul stole the necklace?	
1	wh.non.lg.04	What does the soldier think that Stacey wrote?	
1	wh.isl.sh.03	Who wonders whether Tom sold the television?	
1	wh.isl.lg.02	What does the police officer wonder whether Matt chased?	
2	wh.non.sh.02	Who thinks that Matt chased the bus?	
2	wh.non.lg.01	What does the detective think that Paul stole?	
2	wh.isl.sh.04	Who wonders whether Stacey wrote the letter?	
2	wh.isl.lg.03	What does the manager wonder whether Tom sold?	
3	wh.non.sh.03	Who thinks that Tom sold the television?	
3	wh.non.lg.02	What does the police officer think that Matt chased?	
3	wh.isl.sh.01	Who wonders whether Paul stole the necklace?	
3	wh.isl.lg.04	What does the soldier wonder whether Stacey wrote?	
4	wh.non.sh.04	Who thinks that Stacey wrote the letter?	
4	wh.non.lg.03	What does the manager think that Tom sold?	
4	wh.isl.sh.02	Who wonders whether Matt chased the bus?	
4	wh.isl.lg.01	What does the detective wonder whether Paul stole?	

What if you want participants to judge two items per condition?

Increasing the number of items per condition that a participant judges will increase the sensitivity of your experiment. (It will lead to less noise per participant.)

The first thing to remember is our equation: then $\mathbf{I} = \mathbf{C} \times \mathbf{O}$. If you want 2 observations, and have 4 conditions, you will need 8 items per condition:

Condition 1	Condition 2	Condition 3	Condition 4
item 1	item 1	item 1	item 1
item 2	item 2	item 2	item 2
item 3	item 3	item 3	item 3
item 4	item 4	item 4	item 4
item 5	item 5	item 5	item 5
item 6	item 6	item 6	item 6
item 7	item 7	item 7	item 7
item 8	item 8	item 8	item 8

Two items per condition - by hand

If you follow our Latin Square procedure, you will end up with 8 lists:

	List 1	List 2	List 3	List 4	List 5	List 6	List 7	List 8
condition 1	1	2	3	4	5	6	7	8
condition 2	2	3	4	5	6	7	8	1
condition 3	3	4	5	6	7	8	1	2
condition 4	4	5	6	7	8	1	2	3

	List 1	List 2	List 3	List 4
condition 1	1	2	3	4
condition 2	2	3	4	5
condition 3	3	4	5	6
condition 4	4	5	6	7
condition 1	5	6	7	8
condition 2	6	7	8	1
condition 3	7	8	1	2
condition 4	8	1	2	3

All you have to do is cut lists 5 -8, and paste them below lists 1-4.

The result is four lists with two items per condition, and no lexical overlap.

Two items per condition - mildly automated

To get two items per condition you simply use each list number twice:

30	1	wh.non.sh.01	Who thinks that Paul stole the necklace?
31	1	wh.non.sh.02	Who thinks that Matt chased the bus?
32	2	wh.non.sh.03	Who thinks that Tom sold the television?
33	2	wh.non.sh.04	Who thinks that Stacey wrote the letter?
34	3	wh.non.sh.05	Who thinks that Aaron bought the house?
35	3	wh.non.sh.06	Who thinks that George read the book?
36	4	wh.non.sh.07	Who thinks that Heather saw the movie?
37	4	wh.non.sh.08	Who thinks that Casey baked the cake?
38	2	wh.non.lg.01	What does the detective think that Paul stole?
39	2	wh.non.lg.02	What does the police officer think that Matt chased?
40	3	wh.non.lg.03	What does the manager think that Tom sold?
41	3	wh.non.lg.04	What does the soldier think that Stacey wrote?
42	4	wh.non.lg.05	What does the agent think that Aaron bought?
43	4	wh.non.lg.06	What does the teacher think that George read?
44	1	wh.non.lg.07	What does the girl think that Heather saw?
45	1	wh.non.lg.08	What does the guest think that Casey baked?
46	3	wh.isl.sh.01	Who wonders whether Paul stole the necklace?
47	3	wh.isl.sh.02	Who wonders whether Matt chased the bus?
48	4	wh.isl.sh.03	Who wonders whether Tom sold the television?
49	4	wh.isl.sh.04	Who wonders whether Stacey wrote the letter?
50	1	wh.isl.sh.05	Who wonders whether Aaron bought the house?
51	1	wh.isl.sh.06	Who wonders whether George read the book?
52	2	wh.isl.sh.07	Who wonders whether Heather saw the movie?
53	2	wh.isl.sh.08	Who wonders whether Casey baked the cake?
54	4	wh.isl.lg.01	What does the detective wonder whether Paul stole?
55	4	wh.isl.lg.02	What does the police officer wonder whether Matt chased?
56	1	wh.isl.lg.03	What does the manager wonder whether Tom sold?
57	1	wh.isl.lg.04	What does the soldier wonder whether Stacey wrote?
58	2	wh.isl.lg.05	What does the agent wonder whether Aaron bought?
59	2	wh.isl.lg.06	What does the teacher wonder whether George read?
60	3	wh.isl.lg.07	What does the girl wonder whether Heather saw?
61	3	wh.isl.lg.08	What does the guest wonder whether Casey baked?
62			

Two items per condition - mildly automated

And then when you sort you will have 4 lists, each with two items per condition.

1	wh.non.sh.01	Who thinks that Paul stole the necklace?
1	wh.non.sh.02	Who thinks that Matt chased the bus?
1	wh.non.lg.07	What does the girl think that Heather saw?
1	wh.non.lg.08	What does the guest think that Casey baked?
1	wh.isl.sh.05	Who wonders whether Aaron bought the house?
1	wh.isl.sh.06	Who wonders whether George read the book?
1	wh.isl.lg.03	What does the manager wonder whether Tom sold?
1	wh.isl.lg.04	What does the soldier wonder whether Stacey wrote?
2	wh.non.sh.03	Who thinks that Tom sold the television?
2	wh.non.sh.04	Who thinks that Stacey wrote the letter?
2	wh.non.lg.01	What does the detective think that Paul stole?
2	wh.non.lg.02	What does the police officer think that Matt chased?
2	wh.isl.sh.07	Who wonders whether Heather saw the movie?
2	wh.isl.sh.08	Who wonders whether Casey baked the cake?
2	wh.isl.lg.05	What does the agent wonder whether Aaron bought?
2	wh.isl.lg.06	What does the teacher wonder whether George read?
3	wh.non.sh.05	Who thinks that Aaron bought the house?
3	wh.non.sh.06	Who thinks that George read the book?
3	wh.non.lg.03	What does the manager think that Tom sold?
3	wh.non.lg.04	What does the soldier think that Stacey wrote?
3	wh.isl.sh.01	Who wonders whether Paul stole the necklace?
3	wh.isl.sh.02	Who wonders whether Matt chased the bus?
3	wh.isl.lg.07	What does the girl wonder whether Heather saw?
3	wh.isl.lg.08	What does the guest wonder whether Casey baked?
4	wh.non.sh.07	Who thinks that Heather saw the movie?
4	wh.non.sh.08	Who thinks that Casey baked the cake?
4	wh.non.lg.05	What does the agent think that Aaron bought?
4	wh.non.lg.06	What does the teacher think that George read?
4	wh.isl.sh.03	Who wonders whether Tom sold the television?
4	wh.isl.sh.04	Who wonders whether Stacey wrote the letter?
4	wh.isl.lg.01	What does the detective wonder whether Paul stole?
4	wh.isl.lg.02	What does the police officer wonder whether Matt chased?

Some item recommendations

For basic acceptability judgment experiments I generally recommend that you present 2 items per condition per participant. So for a 2x2 design, that means you need 8 items per condition.

I think that 8 items per condition is also sufficient to make (non-statistical) claims about the generalizability of the result to multiple items. So this is a nice starting point for most designs.

Of course, if you have reason to believe that participants will make errors with the items, you should present more than 2 items per condition. Similarly, if you need to demonstrate that the result generalizes to more than 8 items, by all means, use more than 8 items. These are just good starting points for basic acceptability judgment experiments.

Exercise 3

The file exercise.3.xlsx contains four worksheets for you to create: (i) a Latin Square by hand, (ii) a Latin Square that is mildly automated, (iii) a Latin Square with two items per condition by hand, and (iv) a Latin Square with two items per condition that is mildly automated.

Unordered lists

The next step is to combine the fillers with the experimental items to create **unordered lists**.

I like to do a little formatting here. The Latin Square procedure gives you 4 lists of experimental items. I put the item codes to the left of each list, and place a blank column to the left of the item codes. We'll use that column when we order the lists. I also number each list, above the item codes.

	1		2		3		4	
	wh.non.sh.01	Who thinks th	wh.non.sh.02	Who thinks th	wh.non.sh.03	Who thinks th	wh.non.sh.04	Who thinks th
	wh.non.lg.08	What does the	wh.non.lg.01	What does the	wh.non.lg.02	What does the	wh.non.lg.03	What does the
	wh.isl.sh.07	Who wonders	wh.isl.sh.08	Who wonders	wh.isl.sh.01	Who wonders	wh.isl.sh.02	Who wonders
	wh.isl.lg.06	What does the	wh.isl.lg.07	What does the	wh.isl.lg.08	What does the	wh.isl.lg.01	What does the
	wh.non.sh.05	Who thinks th	wh.non.sh.06	Who thinks th	wh.non.sh.07	Who thinks th	wh.non.sh.08	Who thinks th
	wh.non.lg.04	What does the	wh.non.lg.05	What does the	wh.non.lg.06	What does the	wh.non.lg.07	What does the
	wh.isl.sh.03	Who wonders	wh.isl.sh.04	Who wonders	wh.isl.sh.05	Who wonders	wh.isl.sh.06	Who wonders
	wh.isl.lg.02	What does the	wh.isl.lg.03	What does the	wh.isl.lg.04	What does the	wh.isl.lg.05	What does the

Unordered lists

The next step is to add the fillers to these lists.

You have three options when it comes to adding fillers:

**Identical fillers
items for each list:**

This is the most controlled option. Every participant sees the same filler items, so the fillers don't introduce any variability into the experiment.

**Different items
(but same types)
for each list:**

This basically treats the fillers like experimental items. I don't know why you would do this, unless you wanted to analyze the fillers. But I've seen this.

**Use a second
experiment as
the filler items:**

This saves time (and perhaps money). However, it means that your "fillers" are introducing variability between participants. You also have to be careful about which experiments to combine. You don't want the items from the two experiments influencing each other (so they should be relatively distinct phenomena.)

Unordered lists

In the example materials, I am going with option 1: identical filler items for each list. I think this should be the default option. You can use the other options if you have reason to.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		1			2			3			4		
2		wh.non.sh.01	Who thinks th		wh.non.sh.02	Who thinks th		wh.non.sh.03	Who thinks th		wh.non.sh.04	Who thinks th	
3		wh.non.lg.08	What does the		wh.non.lg.01	What does the		wh.non.lg.02	What does the		wh.non.lg.03	What does the	
4		wh.isl.sh.07	Who wonders		wh.isl.sh.08	Who wonders		wh.isl.sh.01	Who wonders		wh.isl.sh.02	Who wonders	
5		wh.isl.lg.06	What does the		wh.isl.lg.07	What does the		wh.isl.lg.08	What does the		wh.isl.lg.01	What does the	
6		wh.non.sh.05	Who thinks th		wh.non.sh.06	Who thinks th		wh.non.sh.07	Who thinks th		wh.non.sh.08	Who thinks th	
7		wh.non.lg.04	What does the		wh.non.lg.05	What does the		wh.non.lg.06	What does the		wh.non.lg.07	What does the	
8		wh.isl.sh.03	Who wonders		wh.isl.sh.04	Who wonders		wh.isl.sh.05	Who wonders		wh.isl.sh.06	Who wonders	
9		wh.isl.lg.02	What does the		wh.isl.lg.03	What does the		wh.isl.lg.04	What does the		wh.isl.lg.05	What does the	
10		1F.01	Mike prefers t		1F.01	Mike prefers t		1F.01	Mike prefers t		1F.01	Mike prefers t	
11		1F.02	Jenny cleanec		1F.02	Jenny cleanec		1F.02	Jenny cleanec		1F.02	Jenny cleanec	
12		2F.01	There had all t		2F.01	There had all t		2F.01	There had all t		2F.01	There had all t	
13		2F.02	Lily will dance		2F.02	Lily will dance		2F.02	Lily will dance		2F.02	Lily will dance	
14		3F.01	The specimen		3F.01	The specimen		3F.01	The specimen		3F.01	The specimen	
15		3F.02	With that ann		3F.02	With that ann		3F.02	With that ann		3F.02	With that ann	
16		4F.01	There is likely		4F.01	There is likely		4F.01	There is likely		4F.01	There is likely	
17		4F.02	Richard may h		4F.02	Richard may h		4F.02	Richard may h		4F.02	Richard may h	
18		5F.01	The ball perfe		5F.01	The ball perfe		5F.01	The ball perfe		5F.01	The ball perfe	
19		5F.02	Lloyd Webber		5F.02	Lloyd Webber		5F.02	Lloyd Webber		5F.02	Lloyd Webber	
20		6F.01	There are fire		6F.01	There are fire		6F.01	There are fire		6F.01	There are fire	
21		6F.02	Someone bett		6F.02	Someone bett		6F.02	Someone bett		6F.02	Someone bett	
22		7F.01	Laura is more		7F.01	Laura is more		7F.01	Laura is more		7F.01	Laura is more	
23		7F.02	I hate eating s		7F.02	I hate eating s		7F.02	I hate eating s		7F.02	I hate eating s	
24													

Notice that I've given item codes to the fillers. This allows us to look at their ratings later.

And now you have **unordered lists**.

Ordering the lists

The next step is to order the lists for actual presentation to participants.

The goal of this step is to make the order **appear random to the participant**, while still exerting **control over the order to eliminate confounds**.

We call an order that appears random, but isn't, **pseudorandom**. So, we want to pseudorandomize the lists.

What are some things that we want to control for in our pseudorandomization? (i.e., what are some of the constraints on the order?)

1. We don't want related experimental conditions to appear next to each other.
2. We don't want the experimental items to cluster together separately from the fillers.

... there may be others depending on your experiment ...

Notice that the reason that we can't use a random order is that random means any possible order. A random order could violate our constraints.

Pseudorandomizing by hand

You can use the excel function **=rand()** to generate a random number between 0 and 1 next to each item in a list.

You can then use the excel **sort command** to reorder the list based on the random number.

This will give you a random order. You can then look for yourself to see if it satisfies your constraints. If it does, you are finished. If it doesn't, you can simply use the sort command again to generate a new random order. The rand() function updates after the sort, so you don't need to run it again.

	A	B	C	D	E	F	G	H	I	J	K	L
1			1			2			3			4
2	0.678542385	wh.non.sh.01	Who thinks th		wh.non.sh.02	Who thinks th		wh.non.sh.03	Who thinks th		wh.non.sh.04	Who thinks th
3	0.85146555	wh.non.lg.08	What does the		wh.non.lg.01	What does the		wh.non.lg.02	What does the		wh.non.lg.03	What does the
4	0.763820205	wh.isl.sh.07	Who wonders		wh.isl.sh.08	Who wonders		wh.isl.sh.01	Who wonders		wh.isl.sh.02	Who wonders
5	0.937068547	wh.isl.lg.06	What does the		wh.isl.lg.07	What does the		wh.isl.lg.08	What does the		wh.isl.lg.01	What does the
6	0.31720942	wh.non.sh.05	Who thinks th		wh.non.sh.06	Who thinks th		wh.non.sh.07	Who thinks th		wh.non.sh.08	Who thinks th
7	0.399411995	wh.non.lg.04	What does the		wh.non.lg.05	What does the		wh.non.lg.06	What does the		wh.non.lg.07	What does the
8	0.748024381	wh.isl.sh.03	Who wonders		wh.isl.sh.04	Who wonders		wh.isl.sh.05	Who wonders		wh.isl.sh.06	Who wonders
9	0.755892716	wh.isl.lg.02	What does the		wh.isl.lg.03	What does the		wh.isl.lg.04	What does the		wh.isl.lg.05	What does the
10	0.619243597	1F.01	Mike prefers t		1F.01	Mike prefers t		1F.01	Mike prefers t		1F.01	Mike prefers t
11	0.520781678	1F.02	Jenny cleaned		1F.02	Jenny cleaned		1F.02	Jenny cleaned		1F.02	Jenny cleaned
12	0.339870172	2F.01	There had all f		2F.01	There had all f		2F.01	There had all f		2F.01	There had all f
13	0.378820448	2F.02	Lily will dance		2F.02	Lily will dance		2F.02	Lily will dance		2F.02	Lily will dance
14	0.338458306	3F.01	The specimen		3F.01	The specimen		3F.01	The specimen		3F.01	The specimen
15	0.596879677	3F.02	With that ann		3F.02	With that ann		3F.02	With that ann		3F.02	With that ann
16	0.278934611	4F.01	There is likely		4F.01	There is likely		4F.01	There is likely		4F.01	There is likely
17	0.932298128	4F.02	Richard may h		4F.02	Richard may h		4F.02	Richard may h		4F.02	Richard may h
18	0.799882156	5F.01	The ball perfe		5F.01	The ball perfe		5F.01	The ball perfe		5F.01	The ball perfe
19	0.311106355	5F.02	Lloyd Webber		5F.02	Lloyd Webber		5F.02	Lloyd Webber		5F.02	Lloyd Webber
20	0.318152556	6F.01	There are fire		6F.01	There are fire		6F.01	There are fire		6F.01	There are fire
21	0.887276886	6F.02	Someone bett		6F.02	Someone bett		6F.02	Someone bett		6F.02	Someone bett
22	0.303257151	7F.01	Laura is more		7F.01	Laura is more		7F.01	Laura is more		7F.01	Laura is more
23	=rand()	7F.02	I hate eating s		7F.02	I hate eating s		7F.02	I hate eating s		7F.02	I hate eating s

Counterbalancing order

At this stage, you have one pseudorandom order per list.

But the fact of the matter is that every order has at least one confound in it — **the order itself**. The order itself is going to have some effect, and we can't eliminate it.

When we can't eliminate a confound, one strategy is to **counterbalance** it. The term comes from weights on a scale — if the order is causing one effect, we can try to neutralize that effect by creating the opposite effect.

So, we can counterbalance the order of presentation by doing some simple manipulations:

1. We can create the exact reverse of the order. This new reversed-order will counterbalance the effects of being first or last in the order (practice, fatigue, etc.)
2. We can split the original order in half, and put the second half first and the first half second. This will counterbalance the effects of being in the middle of the order (because the middle items will now be either at the beginning or end of the order).
3. We can also reverse the split order to counterbalance for the new first/last endpoints. (Or split the reverse order, the two are equivalent.)

The split/reverse procedure

Original	Reversed	Split	Split-Reversed
item 1	item 8	item 5	item 4
item 2	item 7	item 6	item 3
item 3	item 6	item 7	item 2
item 4	item 5	item 8	item 1
item 5	item 4	item 1	item 8
item 6	item 3	item 2	item 7
item 7	item 2	item 3	item 6
item 8	item 1	item 4	item 5

This procedure gives you 4 orders per list. So if you have 4 lists to begin with, you will have 16 orders. This is sufficient for most experiments.

(Advanced thought: You can, in principle, get away with one order per list if you don't think that the different items will behave differently in different positions (an item x position interaction). You can create these 4 orders using conditions instead of items, and then apply one order to each of your four

Add practice items

The final step is to add the practice items to the beginning of each list. They will be in the same order for each participant, so this is just copy and paste.

	1			2			3			4	
5	7P	She was the		7P	She was the		7P	She was the		7P	She was the
7	1P	Promise to w		1P	Promise to w		1P	Promise to w		1P	Promise to w
8	4P	The brother		4P	The brother		4P	The brother		4P	The brother
9	6P	The children		6P	The children		6P	The children		6P	The children
0	2P	Ben is hopef		2P	Ben is hopef		2P	Ben is hopef		2P	Ben is hopef
1	5P	All the men s		5P	All the men s		5P	All the men s		5P	All the men s
2	3P	They consid		3P	They consid		3P	They consid		3P	They consid
3	7p	It seems to n		7p	It seems to n		7p	It seems to n		7p	It seems to n
4	1p	There might		1p	There might		1p	There might		1p	There might
5	3F.01	The specime		5F.01	The ball perf		4F.02	Richard may		5F.01	The ball perf
6	wh.isl.lg.02	What does th		wh.isl.sh.08	Who wonder		wh.non.lg.06	What does th		wh.isl.sh.02	Who wonder
7	4F.01	There is likel		2F.01	There had all		3F.02	With that an		6F.01	There are fir
8	7F.01	Laura is mor		6F.02	Someone be		wh.non.sh.07	Who thinks t		wh.non.lg.03	What does th
9	wh.non.sh.01	Who thinks t		wh.isl.lg.07	What does th		1F.02	Jenny cleane		2F.02	Lily will danc
0	2F.02	Lily will danc		5F.02	Lloyd Webbe		7F.02	I hate eating		wh.non.sh.04	Who thinks t
1	wh.non.lg.04	What does th		7F.02	I hate eating		5F.02	Lloyd Webbe		4F.01	There is likel
2	6F.01	There are fir		1F.02	Jenny cleane		wh.isl.lg.08	What does th		1F.01	Mike prefers
3	wh.isl.sh.03	Who wonder		wh.non.sh.06	Who thinks t		6F.02	Someone be		3F.01	The specime
4	1F.01	Mike prefers		3F.02	With that an		2F.01	There had all		wh.isl.lg.01	What does th
5	4F.02	Richard may		wh.non.lg.05	What does th		wh.isl.sh.05	Who wonder		7F.01	Laura is mor
6	wh.non.lg.08	What does th		4F.02	Richard may		1F.01	Mike prefers		wh.isl.sh.06	Who wonder
7	3F.02	With that an		1F.01	Mike prefers		3F.01	The specime		2F.01	There had all
8	wh.non.sh.05	Who thinks t		wh.isl.sh.04	Who wonder		wh.isl.lg.04	What does th		6F.02	Someone be
9	1F.02	Jenny cleane		6F.01	There are fir		4F.01	There is likel		wh.isl.lg.05	What does th
0	7F.02	I hate eating		wh.non.lg.01	What does th		7F.01	Laura is mor		5F.02	Lloyd Webbe
1	5F.02	Lloyd Webbe		2F.02	Lily will danc		wh.non.sh.03	Who thinks t		7F.02	I hate eating
2	wh.isl.lg.06	What does th		wh.non.sh.02	Who thinks t		2F.02	Lily will danc		1F.02	Jenny cleane
3	6F.02	Someone be		7F.01	Laura is mor		wh.non.lg.02	What does th		wh.non.sh.08	Who thinks t
4	2F.01	There had all		4F.01	There is likel		6F.01	There are fir		3F.02	With that an
5	wh.isl.sh.07	Who wonder		wh.isl.lg.03	What does th		wh.isl.sh.01	Who wonder		wh.non.lg.07	What does th
6	5F.01	The ball perf		3F.01	The specime		5F.01	The ball perf		4F.02	Richard may

Now you have complete lists! (NB: I am going back to one order per list for simplicity. But remember that the safest option is (at least) 4 orders per list.)

Make a set of item code keys

At this point, you should also make a file with just the item codes in the correct orders. We will use this when we analyze the data later.

1	2	3	4
7P	7P	7P	7P
1P	1P	1P	1P
4P	4P	4P	4P
6P	6P	6P	6P
2P	2P	2P	2P
5P	5P	5P	5P
3P	3P	3P	3P
7p	7p	7p	7p
1p	1p	1p	1p
3F.01	5F.01	4F.02	5F.01
wh.isl.lg.02	wh.isl.sh.08	wh.non.lg.06	wh.isl.sh.02
4F.01	2F.01	3F.02	6F.01
7F.01	6F.02	wh.non.sh.07	wh.non.lg.03
wh.non.sh.01	wh.isl.lg.07	1F.02	2F.02
2F.02	5F.02	7F.02	wh.non.sh.04
wh.non.lg.04	7F.02	5F.02	4F.01
6F.01	1F.02	wh.isl.lg.08	1F.01
wh.isl.sh.03	wh.non.sh.06	6F.02	3F.01
1F.01	3F.02	2F.01	wh.isl.lg.01
4F.02	wh.non.lg.05	wh.isl.sh.05	7F.01
wh.non.lg.08	4F.02	1F.01	wh.isl.sh.06
3F.02	1F.01	3F.01	2F.01
wh.non.sh.05	wh.isl.sh.04	wh.isl.lg.04	6F.02
1F.02	6F.01	4F.01	wh.isl.lg.05
7F.02	wh.non.lg.01	7F.01	5F.02
5F.02	2F.02	wh.non.sh.03	7F.02
wh.isl.lg.06	wh.non.sh.02	2F.02	1F.02
6F.02	7F.01	wh.non.lg.02	wh.non.sh.08
2F.01	4F.01	6F.01	3F.02
wh.isl.sh.07	wh.isl.lg.03	wh.isl.sh.01	wh.non.lg.07
5F.01	3F.01	5F.01	4F.02

The code I am going to give you requires that there be a number at the top of each list, and that there be no spaces between the lists.

In principle, you could write a script that doesn't care about these things. That is going to be up to you, and R.

The code also looks for this to be a separate CSV file. I've given you this separate file (keys.csv) in the packet of files that you've downloaded. I put this in the big excel workbook just for convenience.

Hands on practice

Exercise 4:

The file exercise.4.xlsx contains four worksheets that walk through the steps of ordering lists.

The first sheet is for pseudorandomizing the original lists.

The second sheet is for creating four orders per list based on the split/reverse procedure.

The third sheet is for adding practice items.

The fourth sheet is for creating item keys for later use.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

Section 3:
Application

Four basic tasks

There are four basic tasks in experimental syntax. I will briefly talk about all of them, but for most experiments, I believe the best choice is Likert Scale.

- Likert Scale:** Participants judge each sentence individually along a numerical scale. The scale generally has an odd number of points (so there is a middle point), but in theory it could be even.
- Magnitude Estimation:** Participants judge each sentence individually, but judge it relative to a reference sentence. The ratings are numerical.
- Yes-No:** Participants indicate whether a sentence is grammatical/ungrammatical (possible/impossible, acceptable/unacceptable). This is technically a two-alternative forced-choice task (2AFC), but I use that label for the next task.
- Forced-Choice:** Participants judge two (or more) sentences simultaneously, and indicate which is better (or worse). When there are two sentences, it is a two-alternative forced-choice (2AFC).

The Likert Scale Task

Likert Scale: Participants judge each sentence individually along a numerical scale. The scale generally has an odd number of points (so there is a middle point), but in theory it could be even.

	least acceptable						most acceptable
	1	2	3	4	5	6	7
1. Who thinks that John bought a car?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2. What do you think that John bought?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. Who wonders whether John bought a car?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. What do you wonder whether John bought?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

For Likert Scale tasks you have to choose the number of scale points. The trick is to choose a number that is high enough for participants to report as many differences as they want, but not so high that they won't use all of them. I like to use 7. It is also best to use an odd number so there is a middle point.

You also need to label the two ends of the scale. I like to use least/most acceptable. I also like to make the low numbers the low ratings. The reverse seems confusing to some participants.

The Likert Scale Task

What is the difference between an odd number and an even number of points?

I think this question is most salient if you assume (i) a binary grammar (two types of strings: grammatical and ungrammatical), and (ii) a linking hypothesis between acceptability and grammaticality whereby the location on the continuum of acceptability indicates grammaticality (higher is grammatical, lower is ungrammatical).

Both of these assumptions are open areas of research — there are plenty of non-binary approaches to grammar; and there are well-known examples of misalignment between acceptability and grammaticality:

Unacceptable, but probably grammatical:

*The reporter the senator the president insulted contacted filed the story.

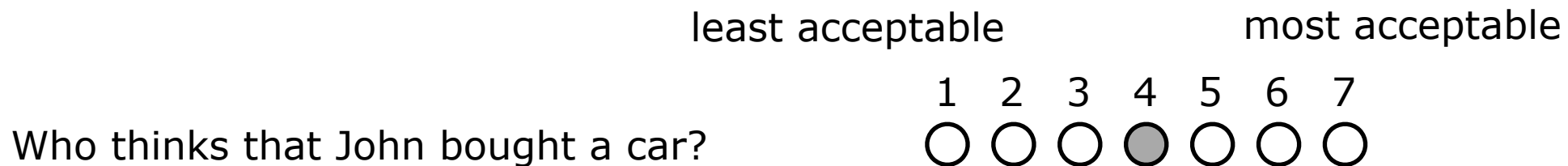
Initially acceptable, but ungrammatical:

More people have been to Russian than I have.

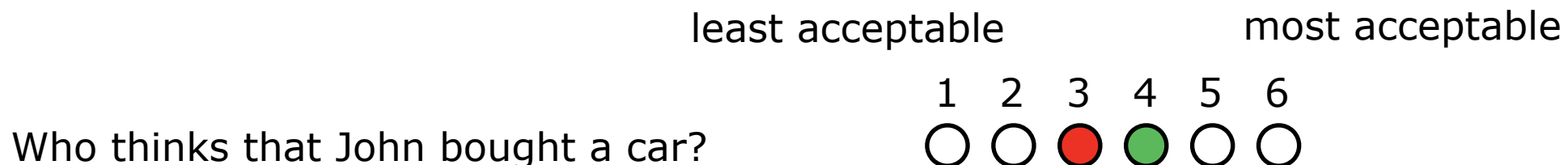
The Likert Scale Task

What is the difference between an odd number and an even number of points?

An odd number of points gives participants the option of saying that they don't know whether this should fall on the acceptable or unacceptable side of the spectrum.



An even number of points turns this into a type of binary forced-choice: participants have to choose a side of the scale. I like to keep the binary aspect out of the Likert scale because the nature of the relationship between acceptability and grammaticality is such an open question.



The Likert Scale Task

Why 7 points? Why not 5 or 9?

Bard et al. 1996 demonstrated that 5 was not enough. Participants can distinguish more than 5 levels of acceptability.

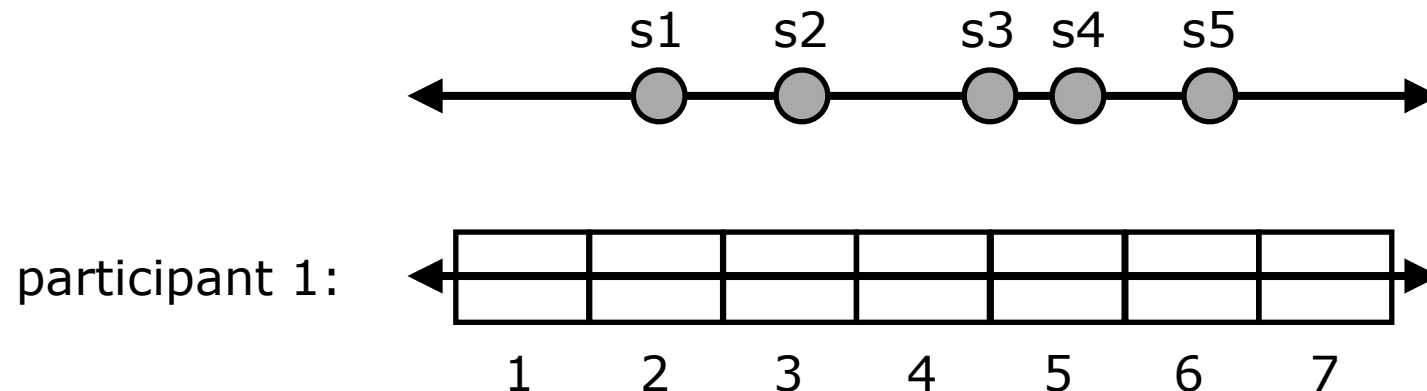
To my knowledge, nobody has demonstrated that 7 is not enough, or that some higher number is preferable. This is a gap in our methodological knowledge.

But a bit later in this lecture, I will show you that completely unconstrained scales do not increase statistical power over 7 point scales... suggesting that there is a finite number that is ideal.

And, I can tell you that I have never had a participant tell me that they felt constrained by a 7 point scale. I only ran in-person studies from 2004 to 2010. Since 2010, nearly all of my experiments have been online, so there is little opportunity for them to tell me (unless they email me).

LS Benefit: Effect sizes

One of the primary benefits of LS tasks is that they provide a clear mechanism for assessing the sizes of differences between conditions

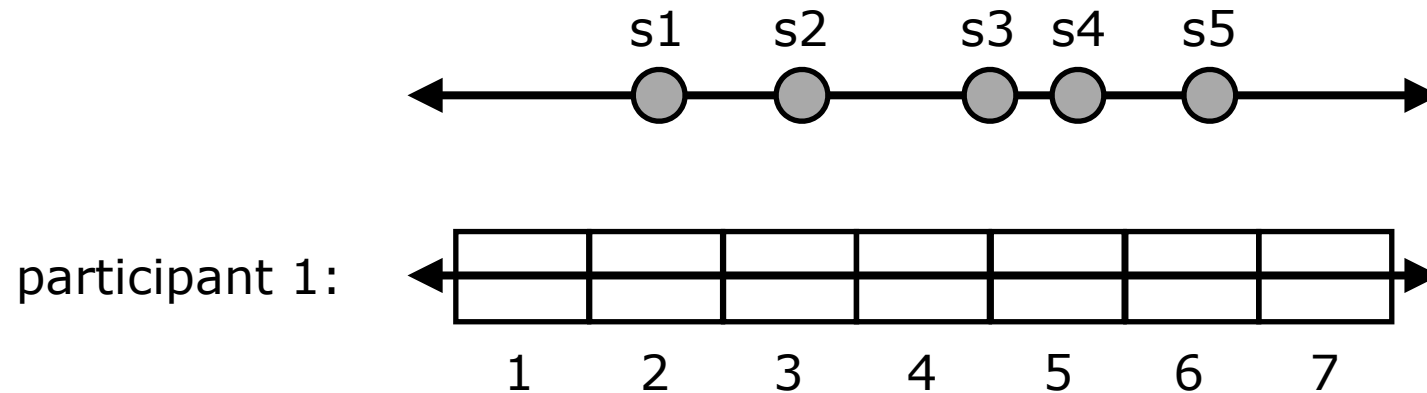


There will be some variability in the cases where a sentence falls on the boundary between two ratings (the way that s3 falls on the 4/5 boundary), but in general, the numerical ratings of LS tasks lend themselves to the types of analyses that we want for factorial designs.

However, this rests on several assumptions about how participants use the scales. **Can you think of what those assumptions are?** We will go through them in the “drawbacks” slides for LS!

LS Benefit: Multiple comparisons

Even though each sentence is rated in isolation in an LS task, because those ratings are made relative to a scale, it is possible to make comparisons between any and all of the sentences in the experiment.

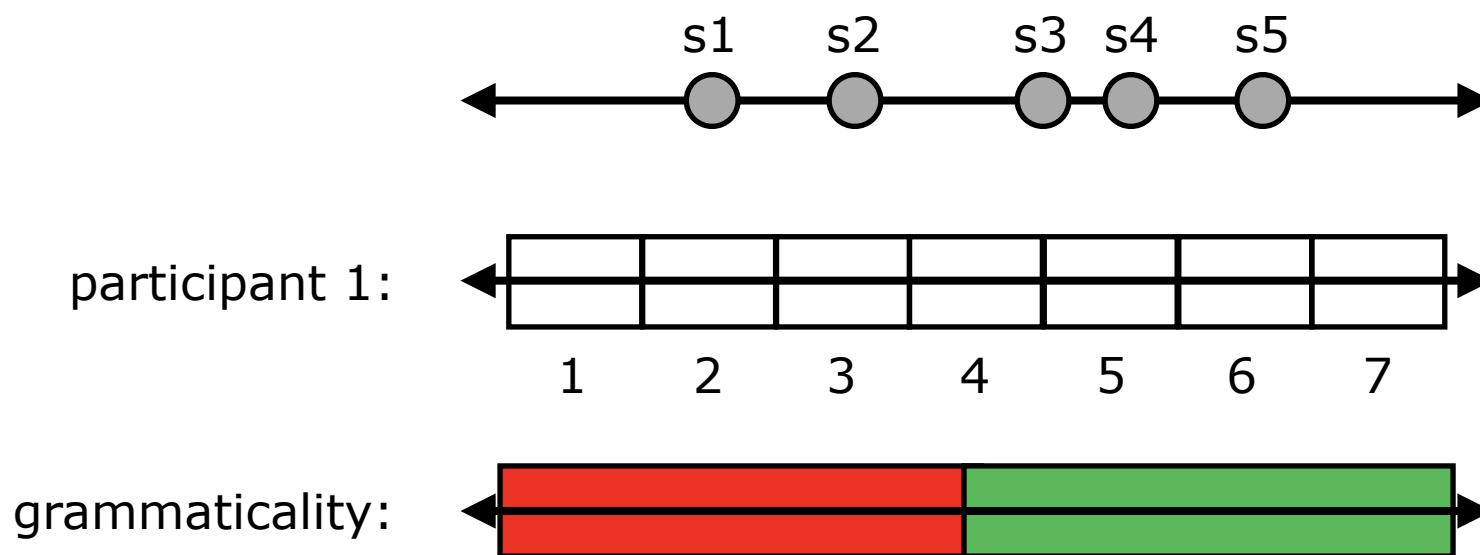


This means that you do not need to know which comparisons you are going to make before you run the experiment. Although in practice, there is no point in running an experiment if you don't know what you are looking for!

LS Benefit: Location on the scale

The responses in LS tasks tell you where along the scale a given sentence is. This means that you can interpret the location on the scale if you want to.

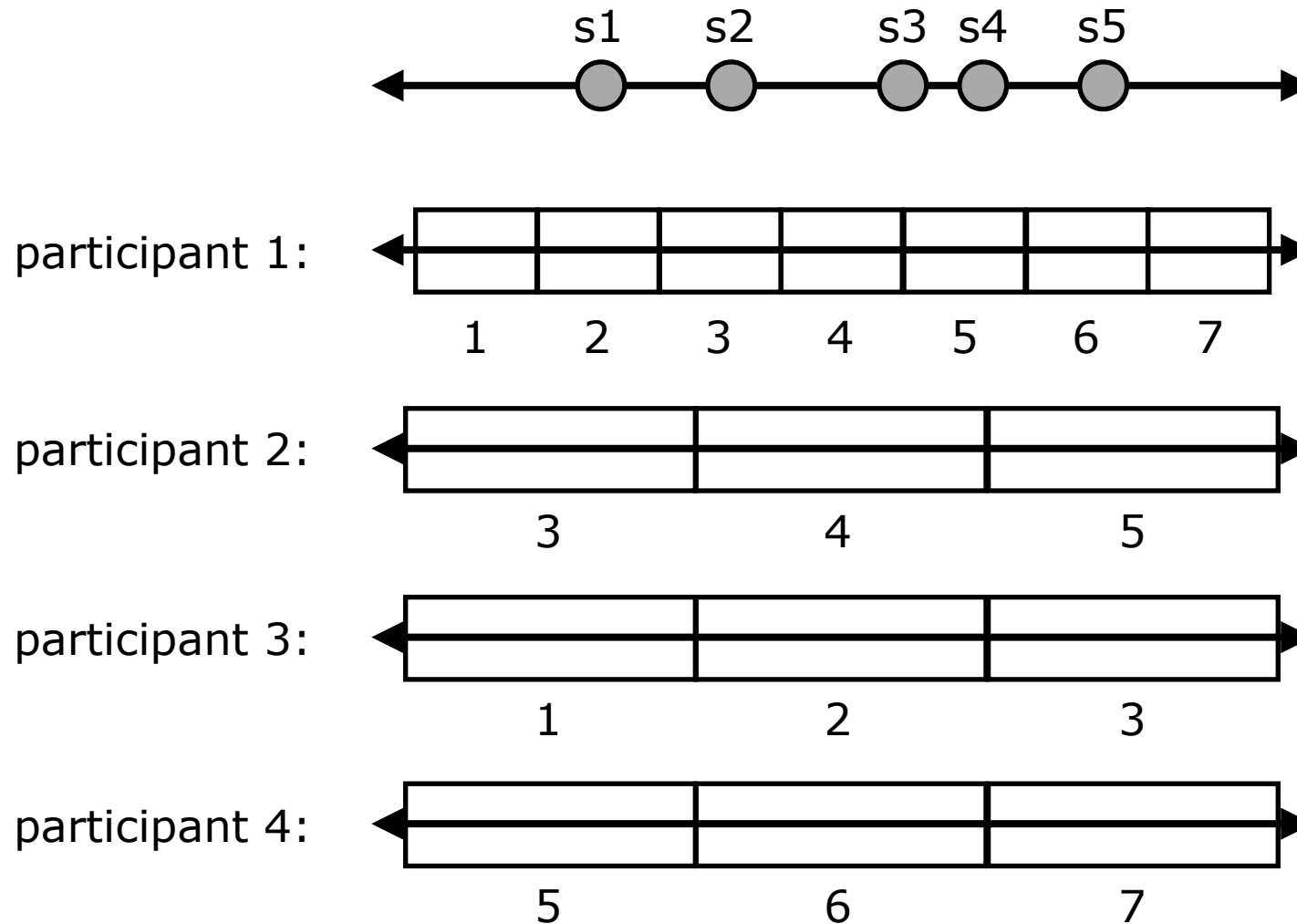
For example, if you assume a binary theory of grammaticality, you could interpret the location of the rating as indicative of the grammaticality of the sentence:



Of course, this rests on a number of assumptions about how the participant uses the scale, how grammars work, and how acceptability maps to grammaticality (a linking hypothesis)! So it isn't an argument, but rather an assumption, or better yet, a research question.

LS Drawback: Scale biases

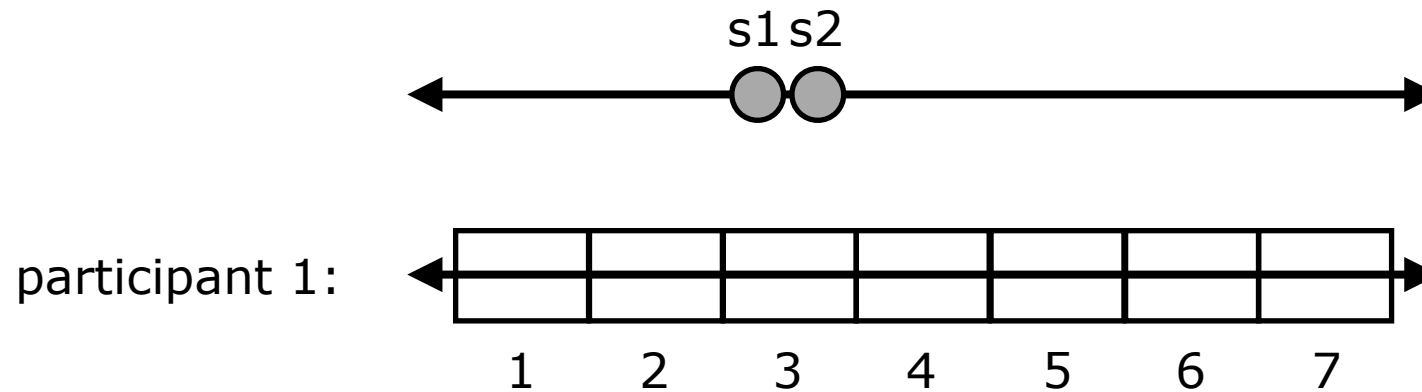
Scale Bias: Different participants might choose to use a scale in different ways.



We can eliminate basic scale bias with a z-score transformation, which we will talk about a bit later.

LS Drawback: Finite options

The LS task gives participants a finite number of response options. This means that there may be certain differences between conditions that they cannot report:



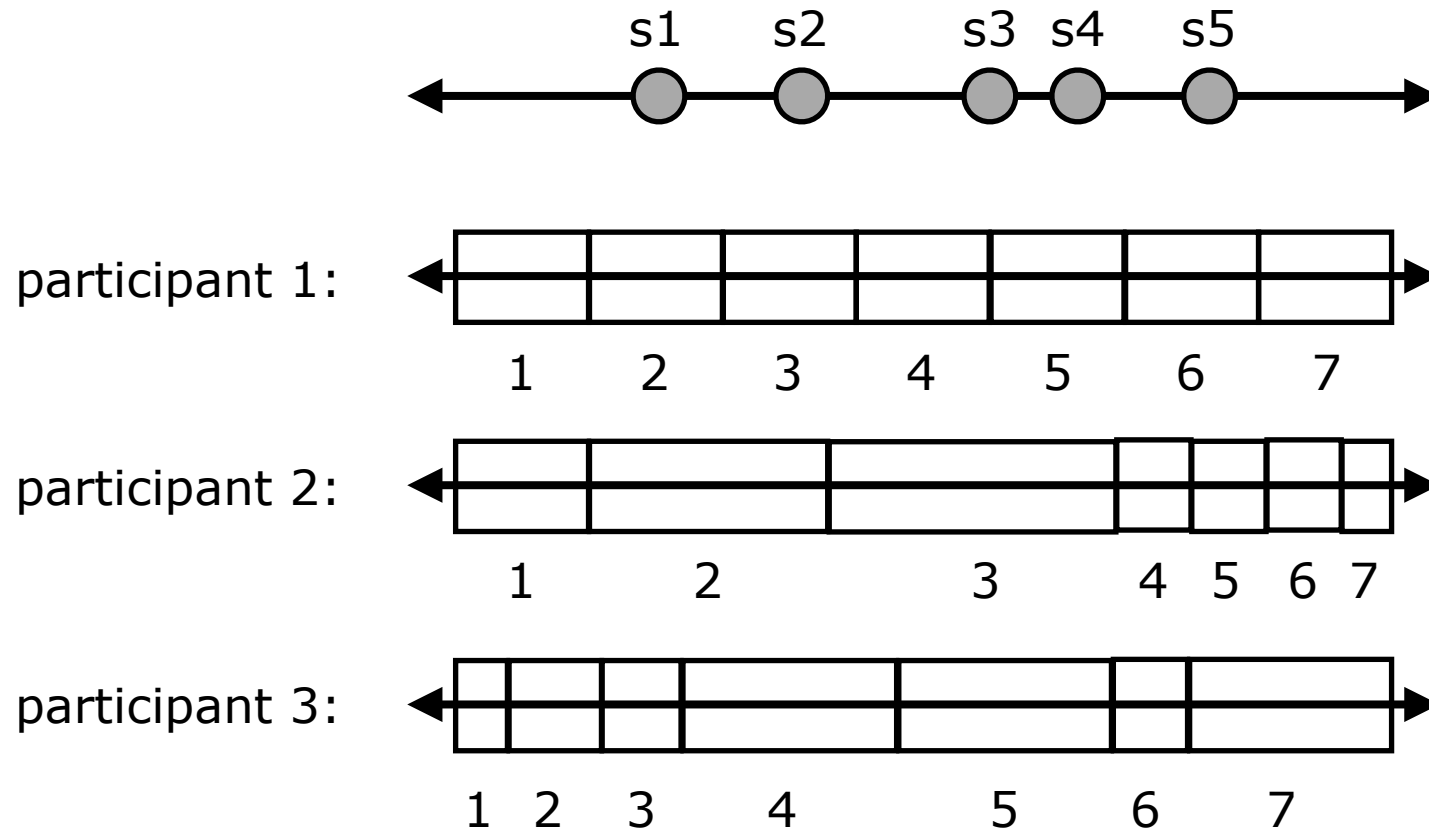
The two sentences above would both be rated a 3, even though they do have a small difference between them.

The obvious solution to this is to increase the number of responses in the scale.

But this runs the risk of introducing too many response options. If the scale defines units that are smaller than the units that humans can use, it could introduce noise in the measurements (or stress in the participants).

LS Drawback: Non-linear scales

One of the assumptions in the LS task is that each of the response categories is exactly the same size (that they define the same interval). But this need not be the case:



There is no easy solution to this (although one could imagine building a model to try to estimate these non-linearities for each participant).

The Magnitude Estimation Task

Magnitude Estimation: Participants judge each sentence individually, but judge it relative to a reference sentence. The ratings are numerical.

The first step is to define a reference stimulus. Usually this is chosen to be in approximately the middle of the range of acceptability.

The reference stimulus is called the **standard**. It is assigned a number that represents its acceptability rating. This number is called the **modulus**. Usually the modulus is a nice round number like 100.

Who said my brother was kept tabs on by the FBI? 100

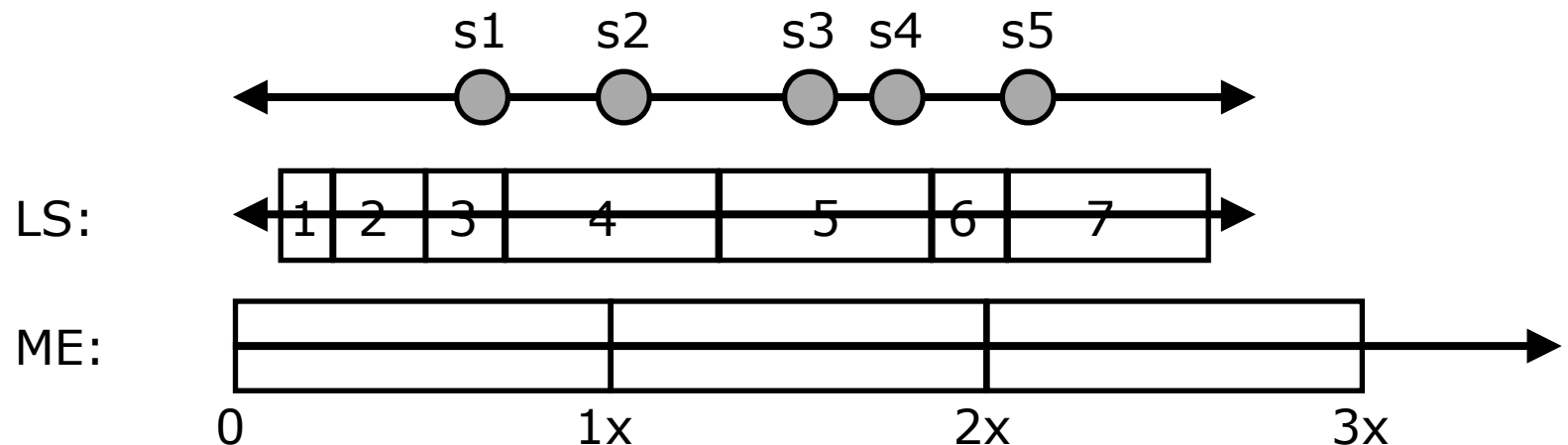
Participants are then asked to rate each sentence in the experiment relative to the standard and modulus. The idea is that if the sentence is twice as acceptable, they would rate the sentence as twice the modulus (e.g., 200). If it is half as acceptable, they would rate it as half the modulus (e.g., 50):

What do you wonder whether John bought? ???

The potential benefits of ME

ME was introduced into psychophysics by Stanley Smith Stevens in order to overcome two deficiencies in the Likert Scale task. It was introduced to syntax by Bard et al. (1996) for exactly the same reason.

1. The LS task uses a finite number of responses. In contrast, ME is usually defined over the positive number line, which is countably infinite. ME sidesteps the problem of defining too many responses by tying the response to a multiple of the standard. This could increase precision.
2. There is no guarantee that the intervals in LS tasks are stable (we called these non-linearities earlier). ME eliminates this problem by using the standard as the perceptual unit (a perceptual "inch"). Although this might differ from participant to participant, the responses within participant should be stable.



The cognitive assumptions of ME

ME makes two assumptions about the cognitive abilities of participants (see Narens 1996 and Luce 2002):

1. Participants must have the ability to make ratio judgments.
2. The number words (called *numerals*) that participants use must represent the mathematical numbers (called *numbers*) that the words denote.

Narens (1996) laid out empirical conditions that would test whether these two assumptions hold. He defined them in terms of a **magnitude production** - a task in which participants must **produce** a second stimulus that has the right proportion to the first stimulus (e.g., lights).

1. **Commutativity:** Magnitude assessments are commutative if the order in which successive adjustments (symbolized by $*$, X is the original stimulus) are made is irrelevant, such that $p * (q * X) \approx q * (p * X)$. Notice that this makes no reference to numbers (it is about matching the resulting stimuli), so it is only testing the ratio judgment assumption.
2. **Multiplicativity:** Magnitude assessments are multiplicative if the result of two successive adjustments matches the result of a single adjustment that is the numeric equivalent of the product of the two adjustments, such that $p * (q * X) \approx r * X$, when $p \cdot q = r$.

Testing commutativity with ME instead of MP

commutativity: $p * (q * X) \approx q * (p * X)$

Experiment 1

sentence X	100
sentence Y	150
sentence Z	200
...	

→ X

→ (p * X)

→ (q * X)

Experiment 2

sentence Y	150
...	
...	
sentence J	300

→ (p * X)

→ q * (p * X)

Experiment 3

sentence Z	200
...	
...	
sentence J	300

→ (q * X)

→ p * (q * X)

If commutativity holds, then both experiment 2 and experiment 3 will yield the same sentence when we look for the p*q value.

The only complicated thing here is that we need to run separate experiments for each participant using the results from experiment 1.

Testing commutativity with ME instead of MP

commutativity: $p * (q * X) \approx q * (p * X)$

Experiment 1

sentence X	100
sentence Y	150
sentence Z	200
...	

→ X

→ $(p * X)$

→ $(q * X)$

Experiment 2

sentence Y	100
...	
...	
sentence J	200

→ $(p * X)/p$

→ $q * (p * X)/p$

Experiment 3

sentence Z	100
...	
...	
sentence J	150

→ $(q * X)/q$

→ $p * (q * X)/q$

We can simplify the process by setting all standards to 100.

This allows us to run all three experiments without creating dependencies across the experiments.

Testing commutativity with ME instead of MP

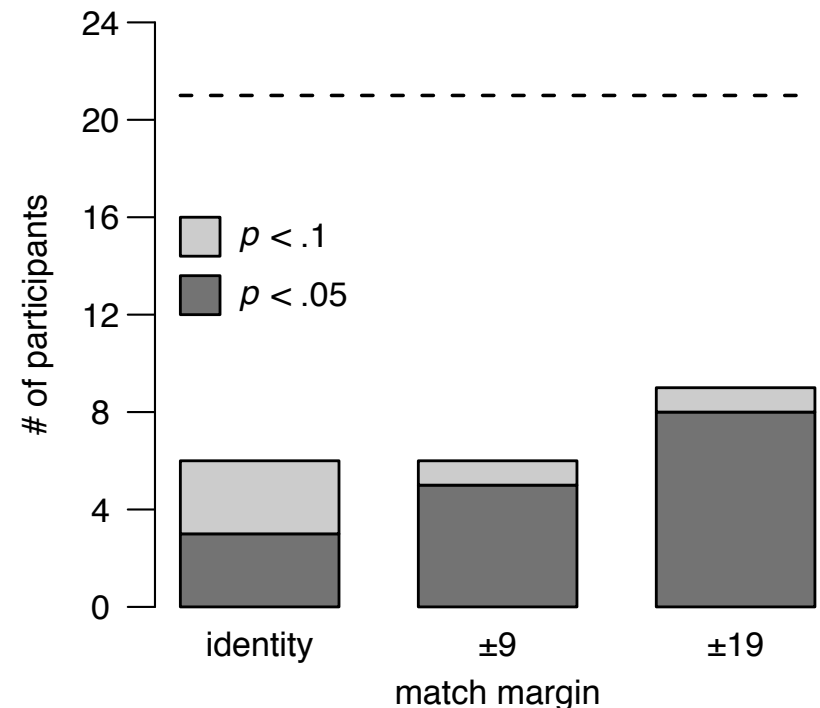
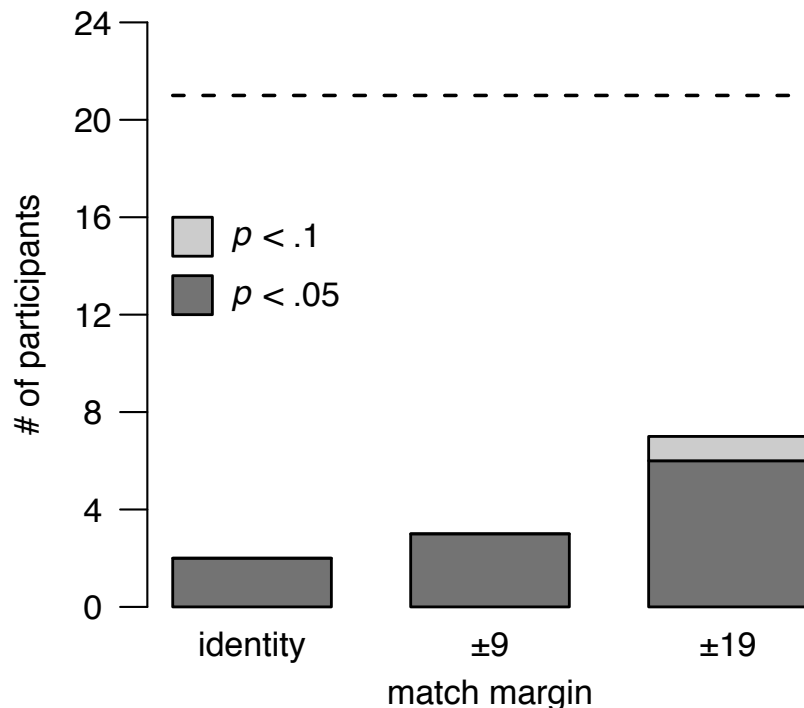
The logic of this experiment relies on finding an item that has the correct rating in both experiments 2 and 3. To increase the likelihood of finding that (should commutatively exist), Sprouse 2011 used 8 experiments instead of 3:

Experiment 1	Experiment 2		Experiment 8
sentence 1 100	sentence 2 100		sentence 8 100
sentence 2	sentence 1		sentence 1
sentence 3	sentence 3		sentence 2
sentence 4	sentence 4		sentence 3
sentence 5	sentence 5	...	sentence 4
sentence 6	sentence 6		sentence 5
sentence 7	sentence 7		sentence 6
sentence 8	sentence 8		sentence 7

Testing commutativity with ME instead of MP

Because of the novelty of this design, and the fact that chance plays such a big role, Sprouse 2011 designed a simulation test to see if the number of matches suggesting commutativity was greater than or less than what would be expected by chance in this design. Basically, a randomization test — which we will discuss in more detail when we do stats later in the course.

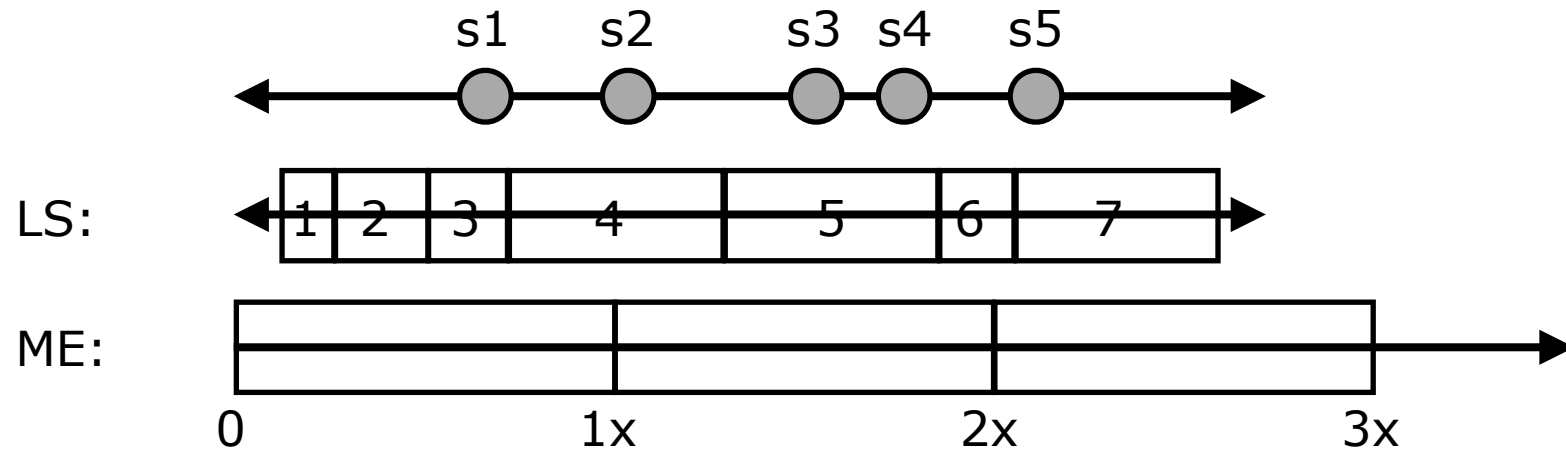
These figures show the number of participants (out of 24) that show evidence (above chance) of commutativity. Sprouse 2011 ran two experiments, so there are two graphs. The dotted line shows the expected number of participants if acceptability judgments had the same level of commutativity as magnitude estimation in psychophysics.



The problem with ME for acceptability

Although there are a number of potential benefits to using ME for psychophysics, it is not clear that these benefits extend to using ME for acceptability judgments because ME for acceptability does not respect the cognitive assumptions of ME (namely, [commutativity](#)).

Commutativity tests the ability of subjects to make ratio judgments. Sprouse 2011's results suggest that humans cannot make ratio judgments of acceptability.



It seems to me that problem is that **ratios require a zero point**. Without a zero point, it is impossible to state ratios. Therefore **ME requires a zero point**. However, it is not clear at all that acceptability has a zero point. What would it even mean for a sentence to have zero acceptability? This lack of meaningful zero point likely causes the breakdown of ME for acceptability.

The Yes-No Task

Yes-No:

Participants indicate whether a sentence is grammatical/ungrammatical (possible/impossible, acceptable/unacceptable). This could also be called a two-alternative forced-choice task, but I reserve that label for the next task.

What do you wonder whether John bought?

Yes ☐

No ☒

I think that Lisa wrote a book.

Yes ☒

No ☐

Who did you meet the man that married?

Yes ☐

No ☒

Although I like to call this the yes-no task, this isn't standardized. Part of the problem is that you could use any pair of categorical labels that you prefer. The other problem is that this is technically an instance of a two-alternative forced-choice task (where the choices are categories).

The Yes-No Task

Yes-No:

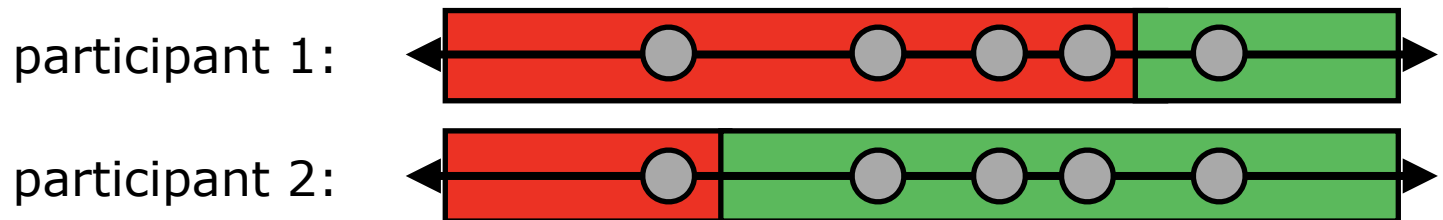
Participants indicate whether a sentence is grammatical/ungrammatical (possible/impossible, acceptable/unacceptable). This could also be called a two-alternative forced-choice task, but I reserve that label for the next task.

Benefit:

If you believe the grammar is binary, then you might also believe that acceptability might reflect that. So, asking people which category sentences belong to could be helpful.

Drawback:

Participants could have different boundary locations. This will create noise in the ratings for some sentences.



Drawback:

This task has less sensitivity to detect differences between sentences that are on the same side of the boundary. This can be problematic for larger designs (e.g. 2x2s).

The Forced-Choice Task

Forced-Choice: Participants judge two (or more) sentences simultaneously, and indicate which is better (or worse). When there are two sentences, it is a two-alternative forced-choice (2AFC).

What do you wonder whether John bought? ☐

What do you think that John bought? ☒

You could in principle have as many sentences as you like per group (2AFC, 3AFC, 4AFC); however, I find it difficult to think of a scenario where this would be useful in building a syntactic theory. The fact that one sentence is better than the other two in a 3AFC doesn't tell you anything about the other two sentences relative to each other. So in practice, this will just be a task for situations when you want to see a difference between two conditions.

FC Benefit

The primary benefit of the forced-choice task is that it is explicitly designed to reveal differences between two conditions. If that is the goal of your hypothesis, you can't get a more perfectly designed task:

What do you wonder whether John bought? ☐

What do you think that John bought? ☒

Notice that the two sentences are the same lexicalization. This means that there is no chance that variability in the lexical items is leading to the difference that is reported by participants.

This also means that there is less of a chance that differences in meaning are driving the difference (only differences in meaning that are tied to the structure could be causing the difference).

Normally, we don't recommend using the same lexicalization. But in this case, the paired presentation means that the difference in structure is going to stand out, so we don't worry about them not noticing it.

FC Drawback: Pre-plan your comparisons

One obvious drawback to the forced-choice task is that you can only compare two conditions if they are **presented as a pair** in the experiment.

- 4. What do you wonder whether John bought?
- 2. What do you think that John bought?
- 1. Who thinks that John bought a car?
- 3. Who wonders whether John bought a car?

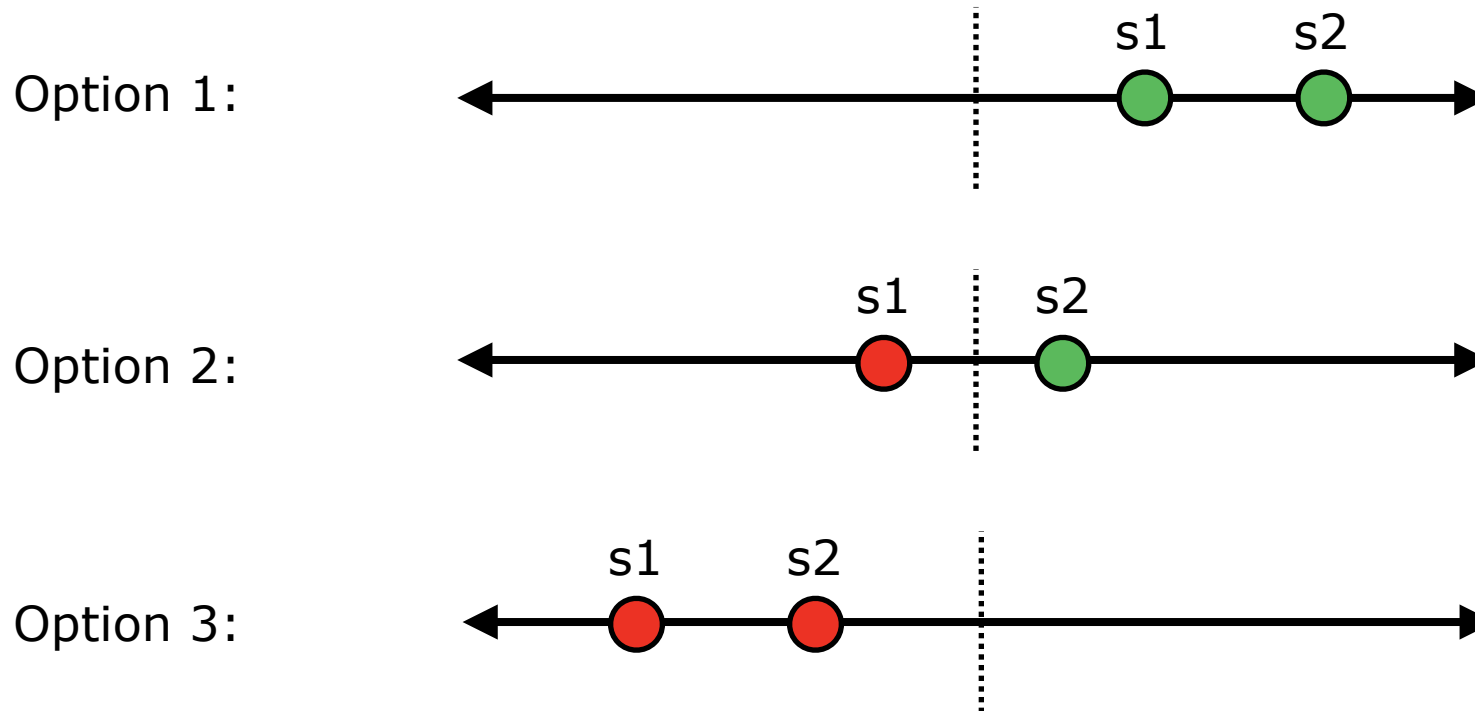
If you wanted to compare 1 and 2 or 3 and 4, you'd have to add another pair containing those sentences to your experiment.

In practice this means that in order to use a forced-choice experiment, you have to know ahead of time exactly which comparisons you want to make so that you can build them into the design of the experiment.

FC Drawback: No location information

Another drawback of the FC task is that it provides **no information** about **where the sentences are on the scale** of relative acceptability.

Let's say that you run an FC experiment, and see that two sentence are different. They could still be anywhere on the scale:



FC Drawback: More complicated assembly

Because the FC task is predicated upon pairs of sentence, the assembly of the task is a bit more complicated than the other tasks.

The first complication is that when you are creating your Latin Squares, you have to keep the pairs of items together. Basically, in an FC task, each “condition” is really a pairing of two sentence types together:

	list 1	list 2	list 3	list 4
condition 1	1-1	2-2	3-3	4-4
condition 2	2-2	3-3	4-4	1-1
condition 3	3-3	4-4	1-1	2-2
condition 4	4-4	1-1	2-2	3-3

FC Drawback: More complicated assembly

The second complication is that you don't want the two sentence types to appear in the same order each time. Half the time you want the better sentence on top in the pair, and half the time you want the worse sentence on top in the pair. This makes sure that participants can't take the strategy "always choose top" or "always choose bottom" with any success.

So after creating your latin square, you have to go through and make sure that half of the pairs are in one order, and the other half are in the other order:

	list 1	list 2	list 3	list 4
C1	1-1	2-2	3-3	4-4
C2	2-2	3-3	4-4	1-1
C3	3-3	4-4	1-1	2-2
C4	4-4	1-1	2-2	3-3

Notice that each **list/column** has two **red items** first, and two **green items** first.

Notice that each **row/condition-pair** has two **red items** first, and two **green items** first.

This is not easy, and I know of no software to automate this.

Comparing Tasks: Qualitative evaluation

- Likert Scale:** LS has the best combination of properties for most experiments. It gives effect size information and location information, and allows for flexible analyses. Its drawbacks are either correctable or mostly theoretical.
- Magnitude Estimation:** Participants can't do ME of acceptability, so it turns into something like an LS task. I would not use it.
- Yes-No:** YN is terrific if you want participants to divide sentences into two groups. But it is not well suited for other types of experiments. The boundary increases noise, and makes the task blind to differences that fall on one side of the boundary.
- Forced-Choice:** FC is terrific you want to detect a difference between two sentences. But it is not well suited for other types of experiments. It provides no location information, and can only be analyzed in direct (pre-planned) pairs.

Comparing Tasks: Statistical Power

Statistical power: The probability that a statistical test will favor the alternative hypothesis when the alternative hypothesis is in fact true.

This definition will make much more sense later in the course when we discuss stats. For now, we can think of it this way: **statistical power is the probability of detecting a difference between conditions when there really is a difference between the conditions.** It can also be thought of as a measure of **sensitivity.**

As a probability, statistical power ranges from 0 to 1, where 0 means something will never happen, and 1 means it is certain to happen.

Probabilities can also be converted to percentages if you like that better: 0% to 100%.

Since power is the probability of detecting an effect when one really exists, we want it to be as high as possible... 1 or 100% would be ideal, though in practice this is difficult to achieve (for reasons that we will discuss when we get to stats).

In psychology, a good rule of thumb is that .8 or 80% power is a good level of power for a given test.

Comparing Tasks: Statistical Power

Statistical power is dependent on a number of factors:

1. The size of the difference to be detected. Larger differences are easier to detect, thus increasing power.
2. The size of the sample of participants. Larger samples provide better estimates (with less noise), thus increasing power..
3. The inherent noise in the task. Less-noisy tasks lead to higher power.
4. The rate of false positives that you are willing to tolerate. It is easy to have perfect (1 or 100%) power: just call everything significant!

So, if you want to compare the statistical power of different tasks, you have to either hold some of these factors constant, or vary some of them to see the impact of different values.

Sprouse and Almeida 2017 did just that for the four tasks that we've been discussing.

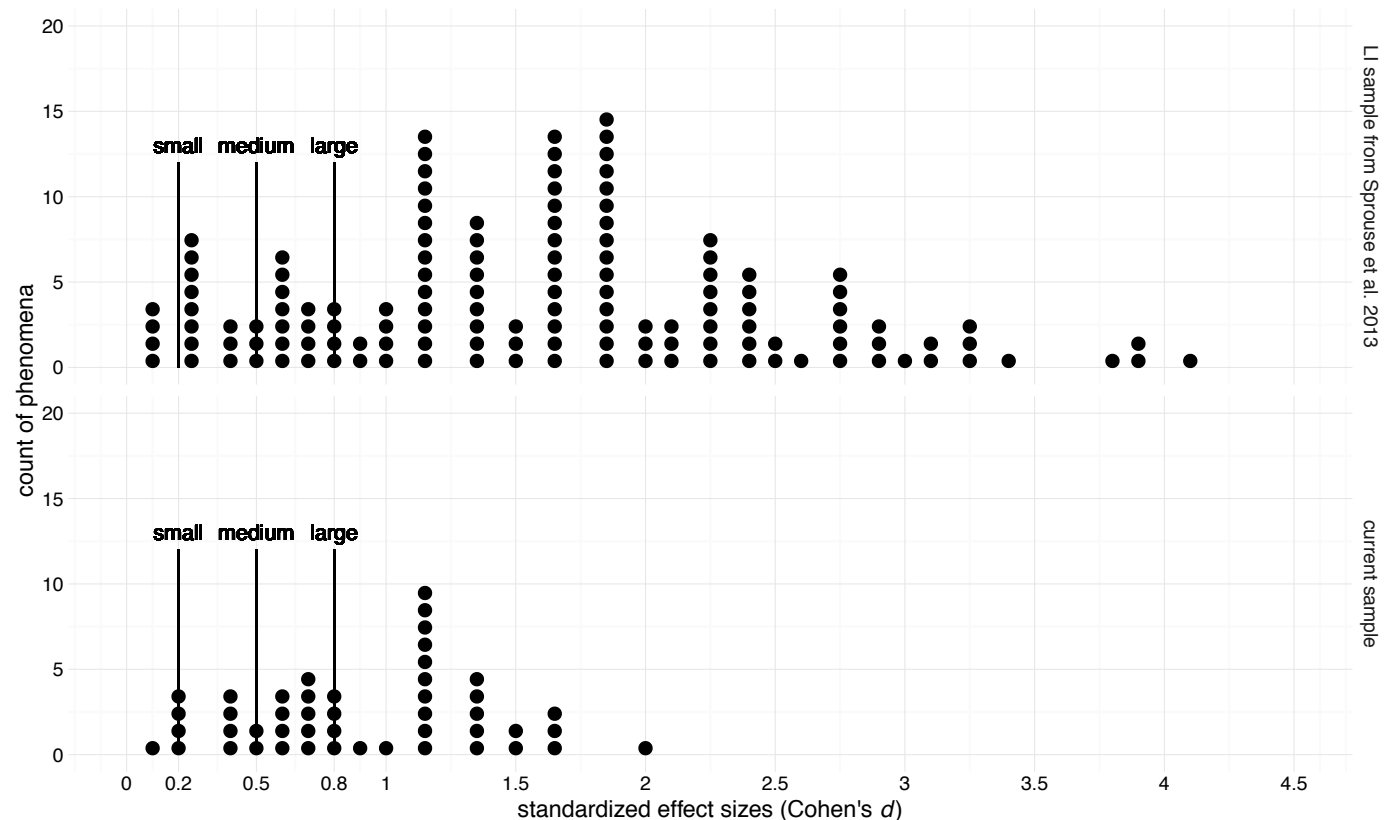
The phenomena

Sprouse et al. 2013 tested 150 phenomena that were randomly sampled from Linguistic Inquiry between 2001 and 2010. Each phenomenon had two conditions: a target condition that was marked unacceptable in the journal article, and a control condition that was marked acceptable.

Sprouse and Almeida 2017 chose 47 of those phenomena to use as critical test cases for comparing power. We chose the 47 to span the lower half of the range of effect sizes. We chose the lower range because that is where the action will be!

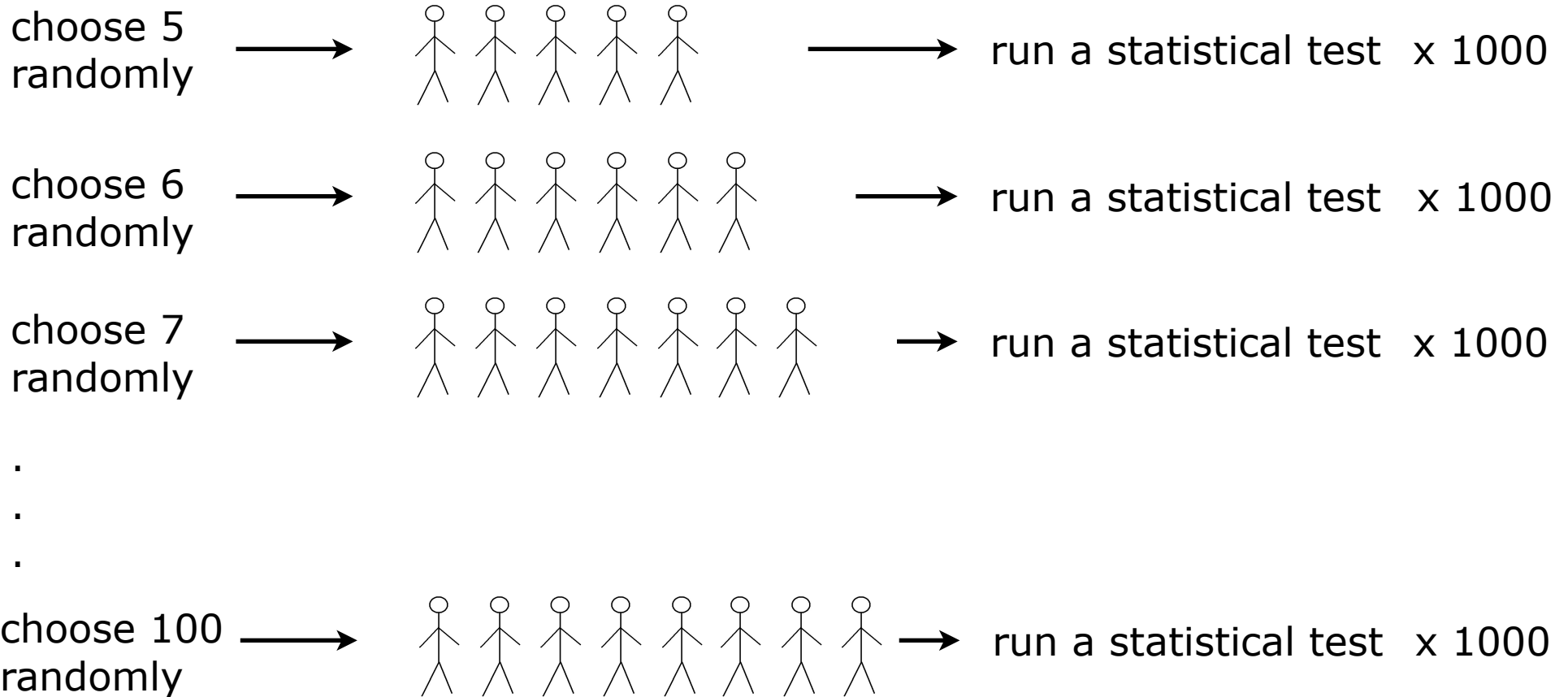
These are standardized effect sizes called Cohen's d .

By standardizing the effect sizes, you can compare across fields!



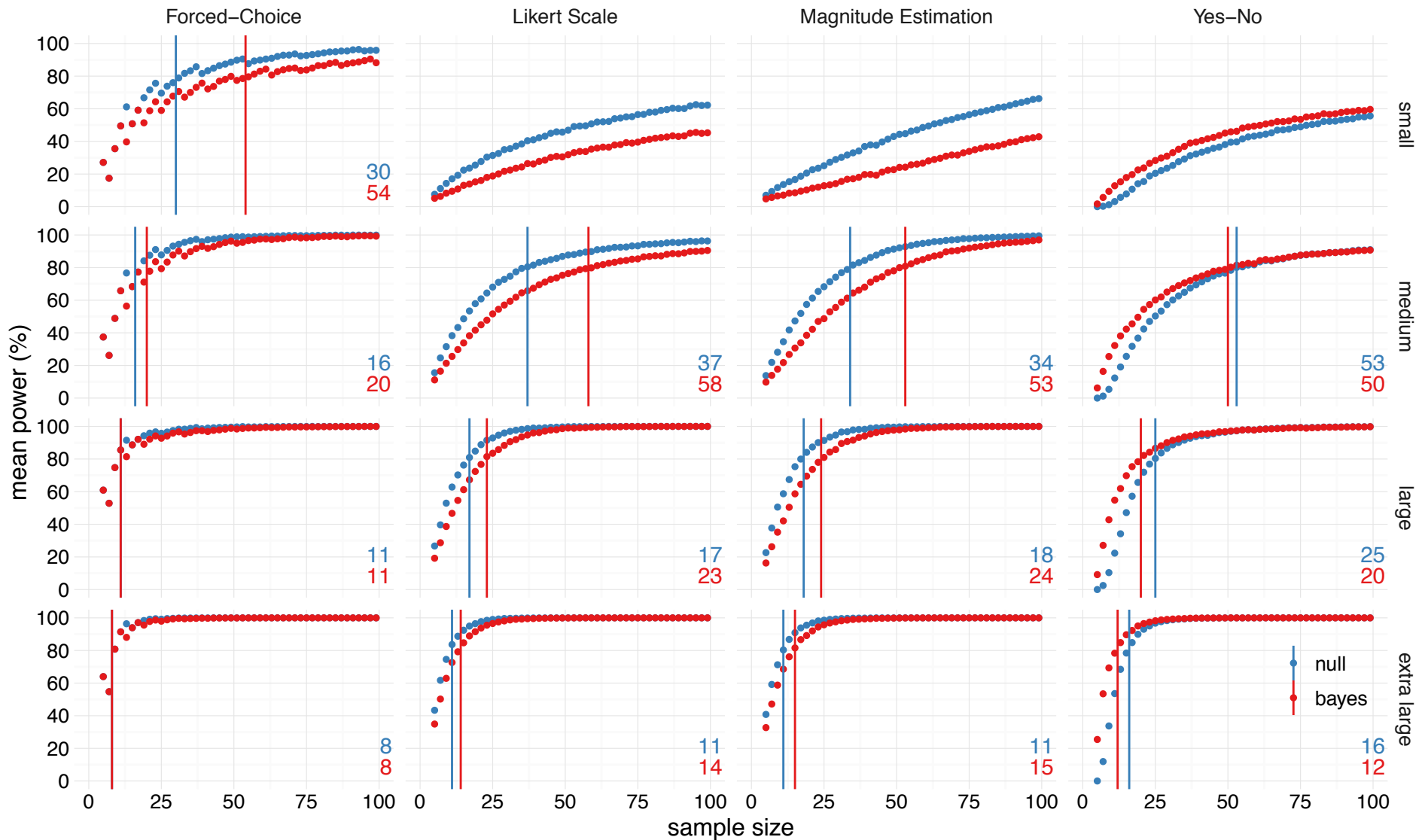
The experiments and simulations

Sprouse and Almeida 2017 collected 144 participants x 4 tasks (=576) for each of the 47 phenomena. This allowed us to create re-sampling simulations to estimate the statistical power of each task for each phenomenon for sample sizes ranging from 5 to 100.



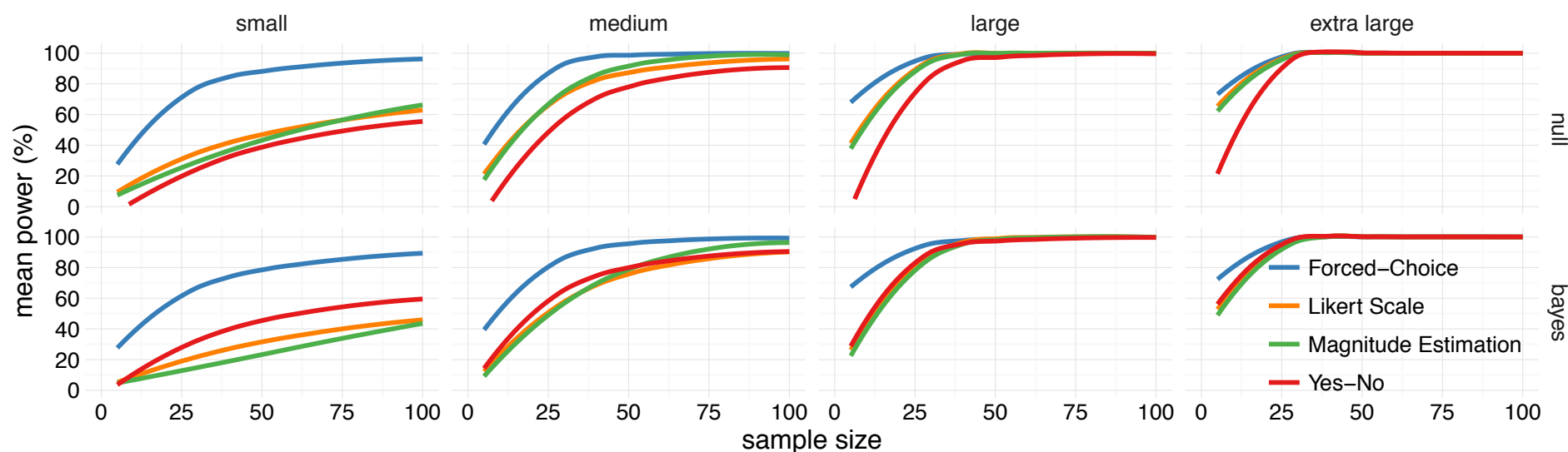
Comparing Tasks: Statistical Power

These graphs show an estimate of statistical power at each sample size from 5 to 100 (x-axis) for each task (columns) for two types of statistical tests (**blue** is null hypothesis testing; **red** is bayes factors). The vertical lines indicate 80%.



Comparing Tasks: Statistical Power

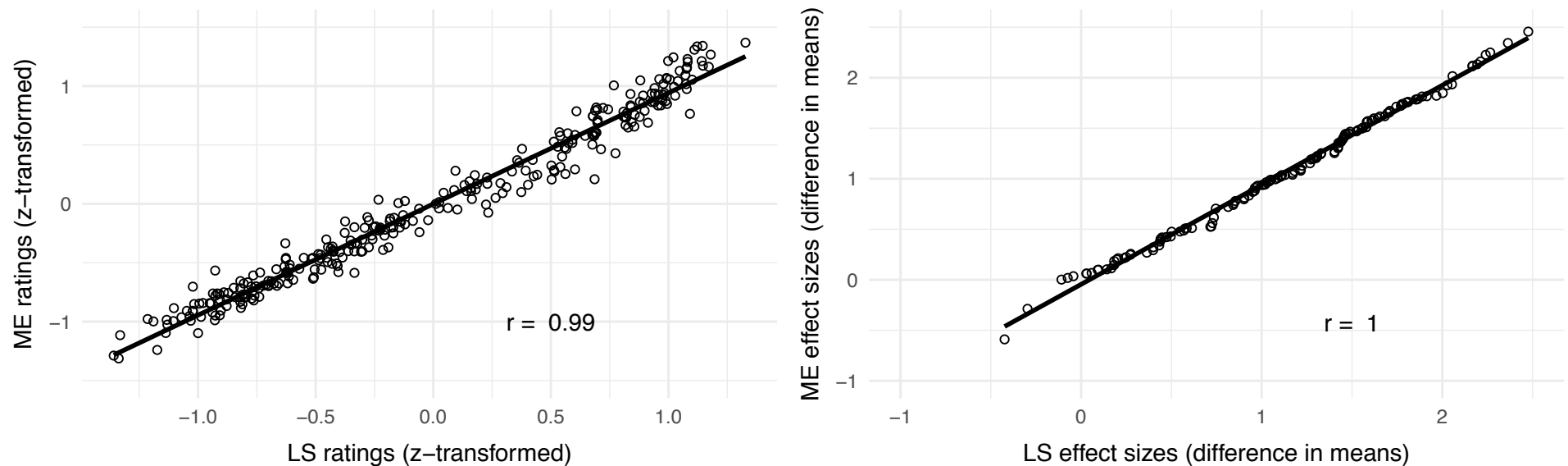
These graphs show an estimate of statistical power at each sample size from 5 to 100 (x-axis) for each group of phenomena (columns) for two types of statistical tests (rows) for each task (colored lines)



What we see is the FC has the most power, which is unsurprising given that it is designed to detect differences between conditions. LS and ME are roughly the same, with some minor advantages for LS (matching the findings of Weskott and Fanselow 2011 for some German phenomena). YN has the lowest power (most of the time), which is unsurprising given that it is not designed to detect differences between conditions, but rather categorize sentence types.

Comparing LS and ME: ratings and effect sizes

As a quick aside, to substantiate my belief that participants turn ME tasks into LS tasks, we can compare the ratings of the same set of 300 conditions from Sprouse et al. 2013 using each task:



The correlations are ridiculous. Pearson's r ranges from -1 (perfectly negatively correlated) to 1 (perfectly positively correlated). The r 's here are .99 for ratings, and 1 for effect sizes.

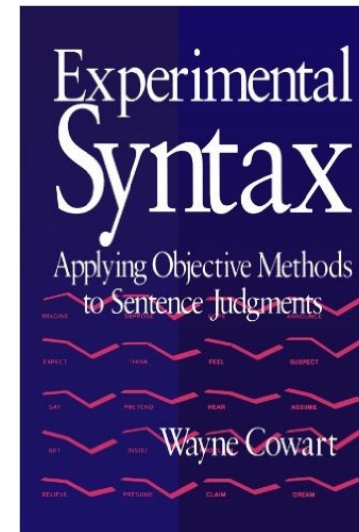
The ratings slope is .95 and the effect sizes slope is 1, again suggesting a really high degree of equivalence between the two tasks. This suggests the power loss in the previous slides is likely due to the higher variability in ME ratings (because of more response options), as noted by Weskott and Fanselow 2011. This is evidence that unlimited response scales are not necessarily ideal. 108

Instructions

It is fairly common for non-linguists to wonder about the instructions that we give participants. I get the sense that in other fields, the instructions can really impact the results.

The only systematic study of this that I know is reported in Cowart's 1997 textbook.

He reports that his manipulations of the instructions led to no differences in the pattern of results that he obtained. All he could do was move judgments (of all sentences) up or down on the scale.



I've never studied this myself, though I've also never noticed any artifacts in my results that might suggest a problem with the instructions.

I have provided **HTML templates** for each of these tasks that can be used on Amazon Mechanical Turk (we'll look at them soon). The instructions that I use are contained in these templates, so you can take a look at them if you'd like some inspiration for instructions for your tasks.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1: Design

Section 2: Analysis

Section 3: Application

Institutional Review Board (IRB) Approval

In the US, before you recruit human participants, you will need approval from your university's Institutional Review Board (the IRB). This is generally a painless process, but it can take a month or more, so you should start planning early.

The process varies from institution to institution, so I can't give you detailed instructions. But I do have some general recommendations:

1. If possible, I would suggest requesting approval for all four possible tasks, both online and offline, and for all possible languages you might study in one application. I would also request approval to test several thousand participants. This will save you time down the road.
2. Acceptability judgments generally fall under survey procedures, which means that they are **exempt (category 2)**. This means that they are exempt from full board review, and instead only require review by the chair of the board. This generally means that the review process will be a bit faster. (The other levels of review are "expedited", which also doesn't require full board review, and "full", which is full board review)
3. Most IRBs require some sort of training before you can submit a proposal for review. So be sure to complete that before you submit your proposal.

Amazon Mechanical Turk

If you are going to be working on US English, Amazon Mechanical Turk can be a great resource for recruiting participants.

The screenshot shows the Amazon Mechanical Turk homepage. At the top, there's a navigation bar with links for 'Your Account', 'HITs', and 'Qualifications'. To the right, it says 'Already have an account? Sign in as a Worker | Requester'. Below this is a blue banner with the text 'Mechanical Turk is a marketplace for work. We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it's convenient. 208,088 HITs available. View them now.' Below the banner, there are two main sections: 'Make Money by working on HITs' and 'Get Results from Mechanical Turk Workers'. The 'Make Money' section includes a flowchart: 'Find an interesting task' (represented by a gear icon) -> 'Work' (represented by a gear icon) -> 'Earn money' (represented by a dollar sign icon). Below this flowchart is a 'Find HITs Now' button. The 'Get Results' section includes a flowchart: 'Fund your account' (represented by a wallet icon) -> 'Load your tasks' (represented by a document icon) -> 'Get results' (represented by a star icon). Below this flowchart is a 'Get Started' button. At the bottom of the page, there's a footer with links for 'FAQ', 'Contact Us', 'Careers at Mechanical Turk', 'Developers', 'Press', 'Policies', 'State Licensing', 'Blog', and 'Service Health Dashboard'. It also includes the copyright notice '©2005-2015 Amazon.com, Inc. or its Affiliates' and the text 'An amazon.com company'.

amazonmechanicalturk
Artificial Intelligence

Already have an account?
Sign in as a [Worker](#) | [Requester](#)

[Your Account](#) [HITs](#) [Qualifications](#)

[Introduction](#) | [Dashboard](#) | [Status](#) | [Account Settings](#)

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.
208,088 HITs available. [View them now.](#)

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. [Find HITs now.](#)

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task → **Work** → **Earn money**

[Find HITs Now](#)

or [learn more about being a Worker](#)

Get Results
from Mechanical Turk Workers

Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. [Get Started.](#)

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Fund your account → **Load your tasks** → **Get results**

[Get Started](#)

[FAQ](#) | [Contact Us](#) | [Careers at Mechanical Turk](#) | [Developers](#) | [Press](#) | [Policies](#) | [State Licensing](#) | [Blog](#) | [Service Health Dashboard](#)

©2005-2015 Amazon.com, Inc. or its Affiliates

An [amazon.com](#) company

Pros: Fast! You can collect a hundred participants in an hour.
More diverse than a university participant pool.

Cons: Not free. You must pay participants (and Amazon).
Less control over the properties of the participants.

AMT Sandbox

The first step is to create a [requester account](#). (AMT divides users into requesters, who post tasks, and workers, who complete them).

If you want to practice using AMT without having to put up a real survey, you can use the [requester's sandbox](#). This is a simulated AMT environment where you can test your experiments without any risk (and without paying anything).

The screenshot shows the Amazon Mechanical Turk Developer Sandbox page. At the top, there's a navigation bar with links: "← go to MTurk.com", "ucisynlab | My Account | Sign Out | Help". Below this, the "amazonmechanical turk" logo is on the left, and "REQUESTER" is on the right. A secondary navigation bar contains "Home", "Create", "Manage", "Developer" (which is highlighted), and "Help". Below this, there's a row with "Resources", "Tools", and "Sandbox" (which is highlighted). On the right side of this row, there's a link "We're Hiring! Learn More". The main content area is titled "Developer Sandbox" with the subtitle "A simulated environment to test your HITs." Below this, there's a section "About the Sandbox" which explains that the sandbox is a simulated environment for testing applications and HITs before publication. It lists three benefits: 1. Free to use for registered Mechanical Turk requesters. Fees will not be withdrawn and payments are not made to Worker accounts. 2. Has functional parity with the production website. 3. Requires only a URL change to configure your application to work against the developer sandbox or the production website. To the right of this section, there's a "Get Started" section with a castle icon. It states that to access the sandbox, you need a Mechanical Turk Requester account and, for programmatic access, an Amazon Web Services (AWS) account. At the bottom right of this section is an orange button labeled "Requester Sandbox ►".

← go to MTurk.com ucisynlab | My Account | Sign Out | Help

amazonmechanical turk Beta REQUESTER

Home Create Manage **Developer** Help

Resources Tools Sandbox We're Hiring! Learn More

Developer Sandbox

A simulated environment to test your HITs.

About the Sandbox

The Mechanical Turk Developer Sandbox is a simulated environment that lets you test your applications and Human Intelligence Tasks (HITs) prior to publication in the marketplace.

Benefits:

- Free to use for registered Mechanical Turk requesters. Fees will not be withdrawn and payments are not made to Worker accounts.
- Has functional parity with the production website.
- Requires only a URL change to configure your application to work against the developer sandbox or the production website.

Get Started

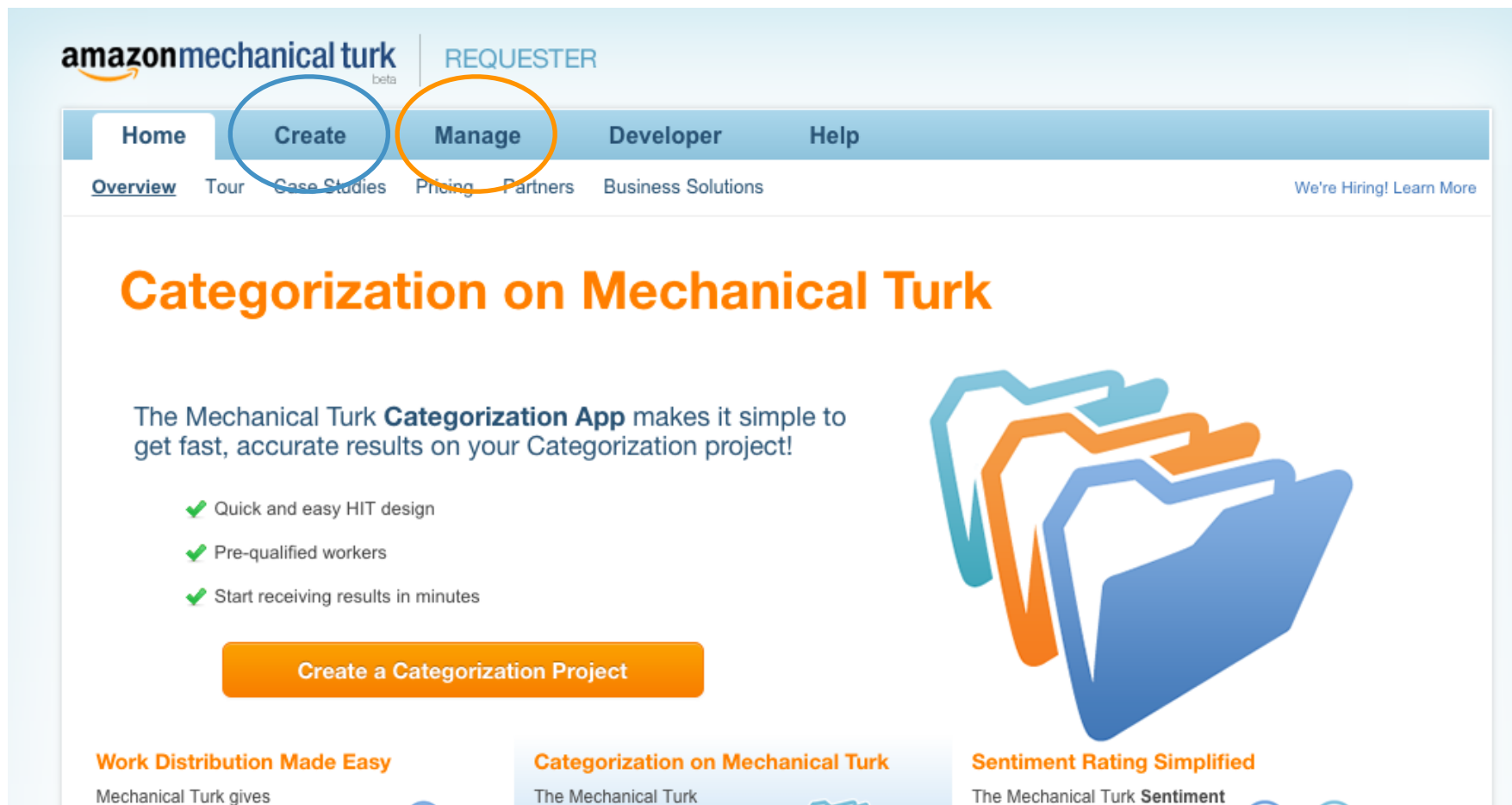
To access the sandbox, you will need a [Mechanical Turk Requester account](#) and, in order to access the sandbox programmatically, you will need an [Amazon Web Services \(AWS\) account](#).

[Requester Sandbox ►](#)

Two stages: create and manage

I am going to use my real account to show you what creating an experiment looks like.

There are basically two stages: the **create** stage, where you create your experiment, and the **manage** stage, where you deploy your experiment and watch the results come in.



amazonmechanicalturk beta REQUESTER

Home Create Manage Developer Help

Overview Tour Case Studies Pricing Partners Business Solutions We're Hiring! Learn More

Categorization on Mechanical Turk

The Mechanical Turk **Categorization App** makes it simple to get fast, accurate results on your Categorization project!

- ✓ Quick and easy HIT design
- ✓ Pre-qualified workers
- ✓ Start receiving results in minutes

Create a Categorization Project

Work Distribution Made Easy
Mechanical Turk gives

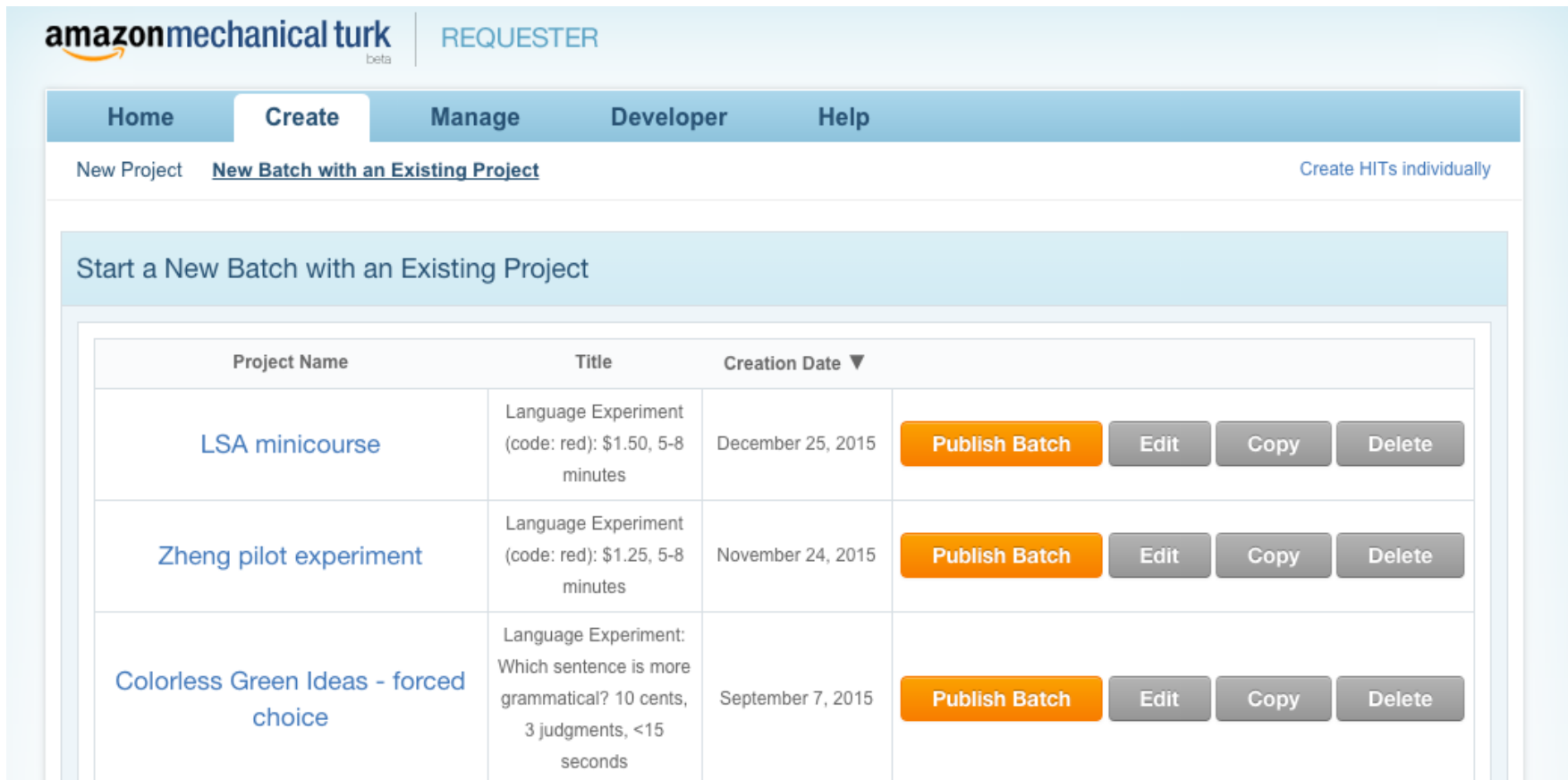
Categorization on Mechanical Turk
The Mechanical Turk

Sentiment Rating Simplified
The Mechanical Turk Sentiment

Creating an experiment

When you click on Create, you will see a list of all of the experiments that you've run in the past. This lets you easily re-use them (or edit them) if you need to.

Your list will be empty (or have demos in it). But you can easily create a new one using one of the AMT HTML templates I have made available.



The screenshot shows the Amazon Mechanical Turk requester interface. At the top, there's a header with the Amazon Mechanical Turk logo and the word "REQUESTER". Below this is a navigation bar with tabs: Home, Create, Manage, Developer, and Help. Under the "Create" tab, there are links for "New Project", "New Batch with an Existing Project", and "Create HITs individually". The main content area is titled "Start a New Batch with an Existing Project" and displays a table of existing experiments.

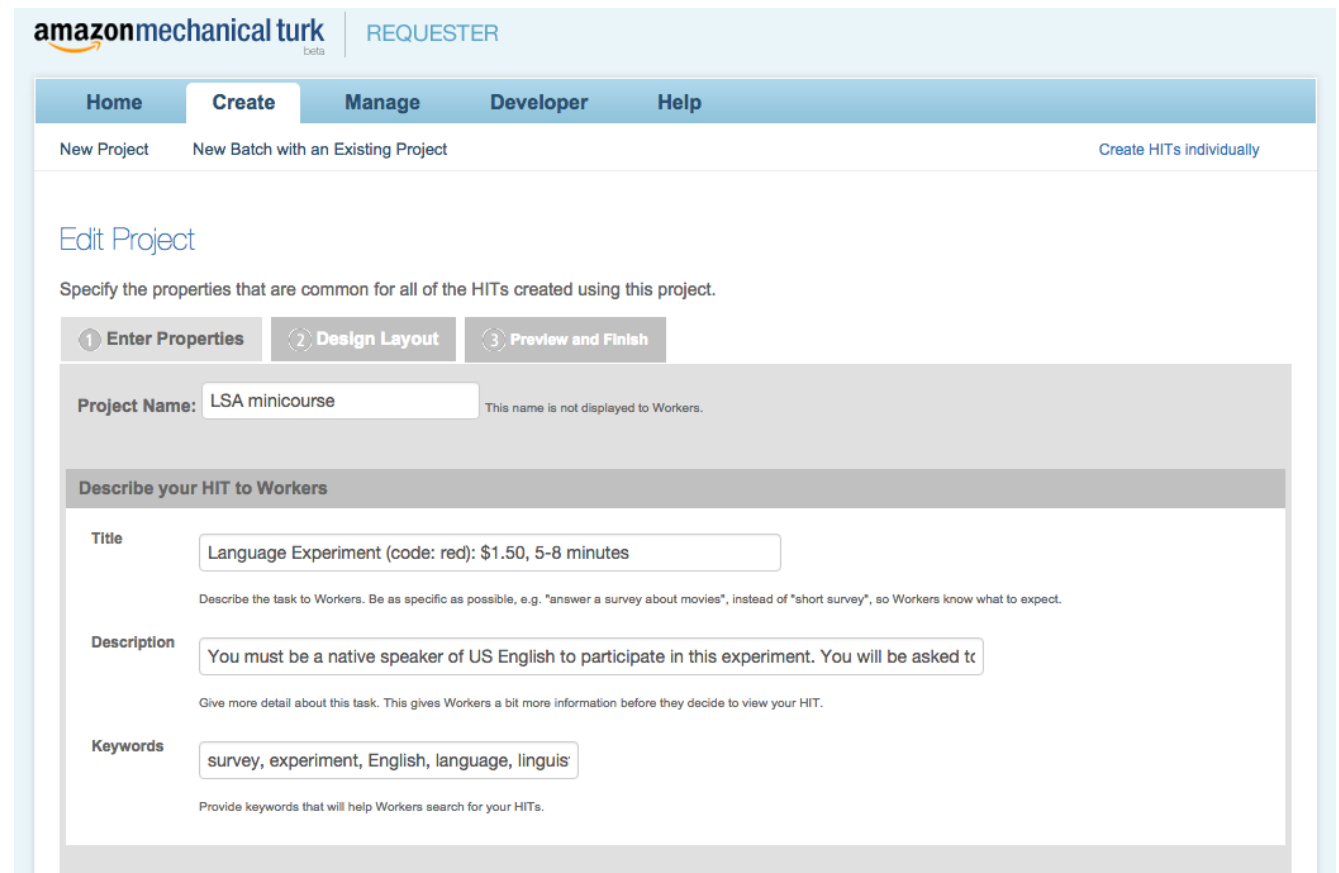
Project Name	Title	Creation Date ▼	
LSA minicourse	Language Experiment (code: red): \$1.50, 5-8 minutes	December 25, 2015	Publish Batch Edit Copy Delete
Zheng pilot experiment	Language Experiment (code: red): \$1.25, 5-8 minutes	November 24, 2015	Publish Batch Edit Copy Delete
Colorless Green Ideas - forced choice	Language Experiment: Which sentence is more grammatical? 10 cents, 3 judgments, <15 seconds	September 7, 2015	Publish Batch Edit Copy Delete

Create: Enter Properties

There are three parts to creating an experiment: entering its properties, designing the layout, and then looking for errors. We start with entering the properties.

The first box is where you enter information that the workers will see.

I like to tell them how long I think it will take, how much I am going to pay, and any requirements that I have (that aren't enforced by AMT - more on this soon.)



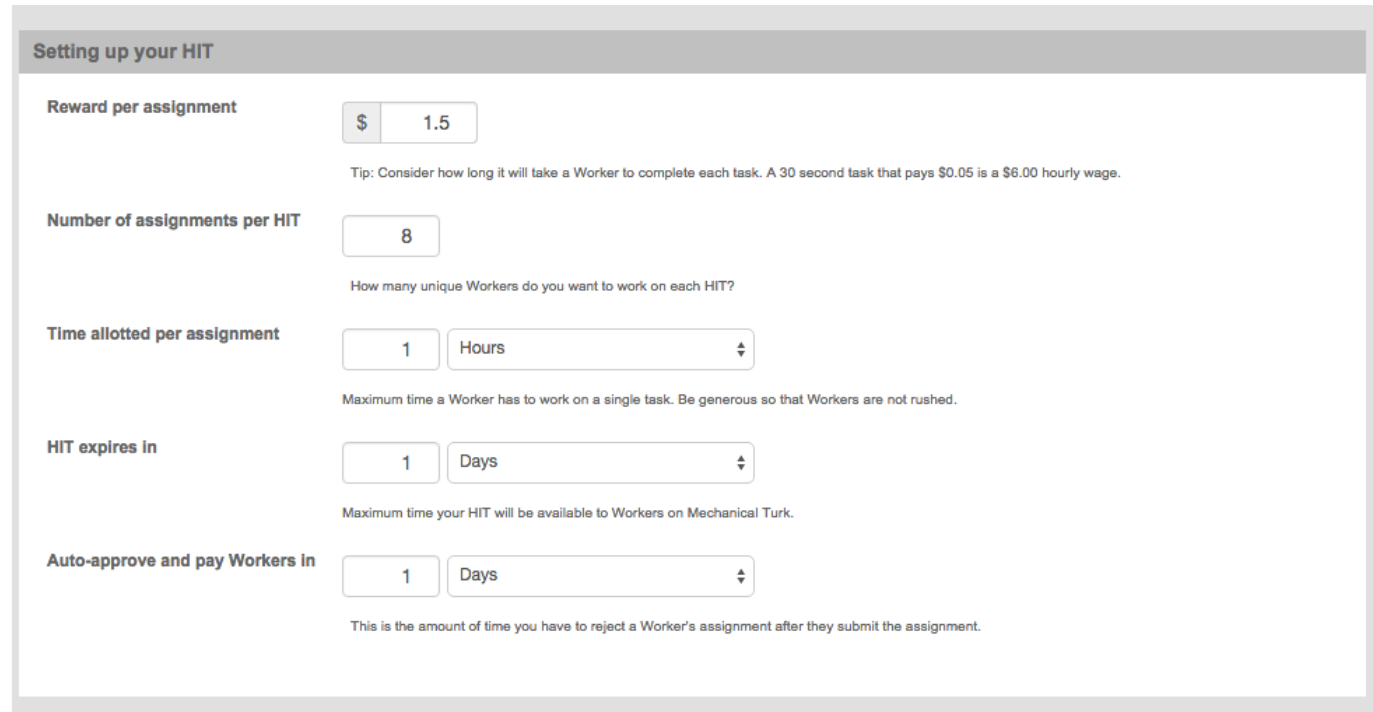
The screenshot shows the 'Create' page on the Amazon Mechanical Turk requester interface. The page has a light blue header with the 'amazonmechanicalturk' logo and 'REQUESTER' text. Below the header is a navigation bar with tabs: 'Home', 'Create' (selected), 'Manage', 'Developer', and 'Help'. Under the 'Create' tab, there are links for 'New Project', 'New Batch with an Existing Project', and 'Create HITs individually'. The main content area is titled 'Edit Project' and includes the instruction: 'Specify the properties that are common for all of the HITs created using this project.' There are three numbered steps: '1 Enter Properties' (active), '2 Design Layout', and '3 Preview and Finish'. The 'Enter Properties' section contains a 'Project Name' field with the value 'LSA minicourse' and a note 'This name is not displayed to Workers.' Below this is a section titled 'Describe your HIT to Workers' with three fields: 'Title' (containing 'Language Experiment (code: red): \$1.50, 5-8 minutes'), 'Description' (containing 'You must be a native speaker of US English to participate in this experiment. You will be asked to'), and 'Keywords' (containing 'survey, experiment, English, language, linguistics'). Each field has a small instructional text below it.

Create: Enter Properties

The second box in “enter properties” is where you set the specific properties of this HIT (Human Intelligence Task — this is what AMT calls a task).

The first box is how much you will the participant.

The second is the number of participants you want to recruit per HIT. Each ordered list you have is a HIT, so you have to do some math here.



The screenshot shows the 'Setting up your HIT' interface. It contains several input fields and dropdown menus for configuring a HIT. The fields are: 'Reward per assignment' (set to \$1.5), 'Number of assignments per HIT' (set to 8), 'Time allotted per assignment' (set to 1 hour), 'HIT expires in' (set to 1 day), and 'Auto-approve and pay Workers in' (set to 1 day). Each field has a corresponding tip or instruction below it.

Property	Value	Unit	Tip/Note
Reward per assignment	1.5	\$	Tip: Consider how long it will take a Worker to complete each task. A 30 second task that pays \$0.05 is a \$6.00 hourly wage.
Number of assignments per HIT	8		How many unique Workers do you want to work on each HIT?
Time allotted per assignment	1	Hours	Maximum time a Worker has to work on a single task. Be generous so that Workers are not rushed.
HIT expires in	1	Days	Maximum time your HIT will be available to Workers on Mechanical Turk.
Auto-approve and pay Workers in	1	Days	This is the amount of time you have to reject a Worker's assignment after they submit the assignment.

If you have 8 ordered lists, and want 24 participants in your sample, then you need 3 participants per list. Since each list is a HIT, you need 3 assignments per HIT. More generally:

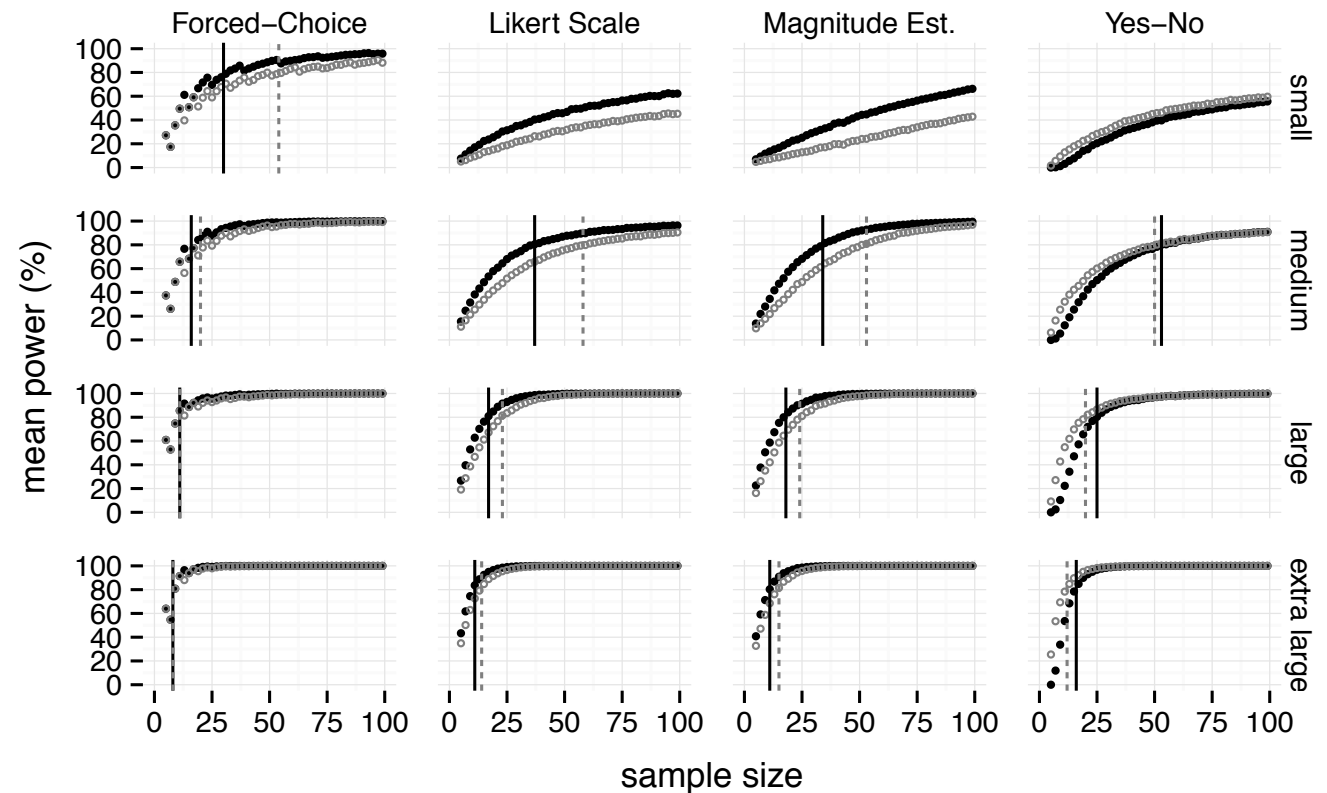
Number of assignments per HIT = total sample size / number of ordered lists

Quick aside - How many participants?

The number of participants that you need is a complex function of, at least, (i) the size of the effect you want to detect, (ii) sensitivity/noise of the task, and (iii) the statistical power you want to achieve (the probability of detecting the effect if it is present).

We can use the graph I showed you before to estimate this relationship.

This graph is based on 50 phenomena from LI, and 1 observation per participant per condition.



There is also a general rule of thumb in statistics that says that you need at least 25 participants (or 24 if your lists are based on multiples of 4). So I suggest using the graph above to calculate a number, and treating 24 as the absolute minimum.

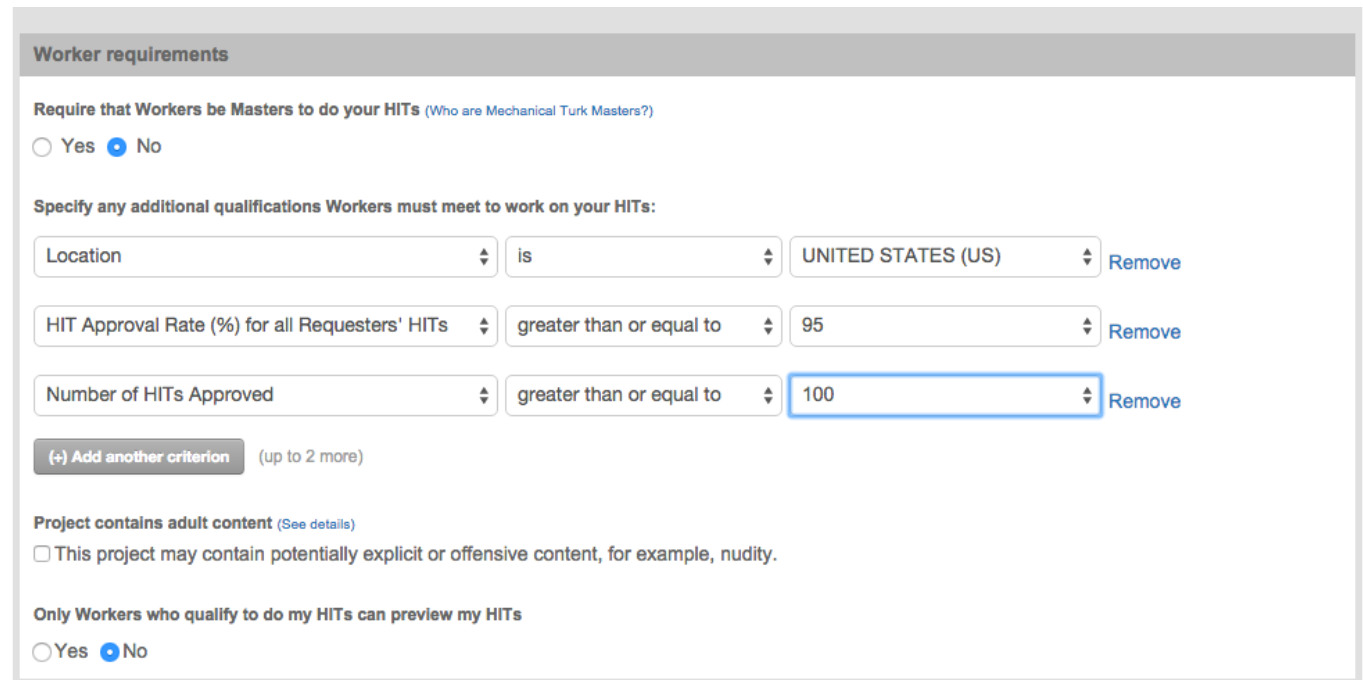
Create: Enter Properties

The final box is where you can enter restrictions on your HIT. Technically, AMT allows you to set very strict requirements — you simply have to create a qualifying task, and then only allow participants who pass your qualifying task to participate in your experiment.

The problem is that there is a trade-off between restricting access and recruiting (diverse) participants. So I try to use a minimum of qualifications.

I set IP location to US to try to limit the number of non-native speakers (more on this later).

I set HIT approval rates and number of HITs approved to numbers that will weed out very bad participants and very new participants.



The screenshot shows the 'Worker requirements' section of the Amazon Mechanical Turk interface. It includes a header 'Worker requirements' and a sub-header 'Require that Workers be Masters to do your HITs (Who are Mechanical Turk Masters?)'. Below this, there are three radio buttons: 'Yes' and 'No', with 'No' selected. The next section is 'Specify any additional qualifications Workers must meet to work on your HITs:'. It contains three criteria: 'Location' is 'UNITED STATES (US)', 'HIT Approval Rate (%) for all Requesters' HITs' is 'greater than or equal to' '95', and 'Number of HITs Approved' is 'greater than or equal to' '100'. Each criterion has a 'Remove' button. There is a button to 'Add another criterion' (up to 2 more). The bottom section is 'Project contains adult content (See details)' with a checkbox 'This project may contain potentially explicit or offensive content, for example, nudity.' which is unchecked. The final section is 'Only Workers who qualify to do my HITs can preview my HITs' with radio buttons 'Yes' and 'No', with 'No' selected.

Worker requirements

Require that Workers be Masters to do your HITs (Who are Mechanical Turk Masters?)

☐ Yes ☒ No

Specify any additional qualifications Workers must meet to work on your HITs:

Location is UNITED STATES (US) Remove

HIT Approval Rate (%) for all Requesters' HITs greater than or equal to 95 Remove

Number of HITs Approved greater than or equal to 100 Remove

(+) Add another criterion (up to 2 more)

Project contains adult content (See details)

☐ This project may contain potentially explicit or offensive content, for example, nudity.

Only Workers who qualify to do my HITs can preview my HITs

☐ Yes ☒ No

Design Layout

The next step is to design the layout of the HIT itself. The basic AMT interface uses HTML. Amazon has tried to make this easy by using a WYSIWYG editor for the HTML. But I find that the only way to really use this for an experiment is to have some familiarity with HTML.

The screenshot shows the Amazon Mechanical Turk (AMT) interface for a requester. The top navigation bar includes 'Home', 'Create', 'Manage', 'Developer', and 'Help'. Under the 'Create' tab, there are links for 'New Project', 'New Batch with an Existing Project', and 'Create HITs individually'. The main heading is 'Edit Project'. Below this, a message states: 'Use the HTML editor below to design the layout of your HIT. This layout is common for all of the HITs created with this project. You can define variables for data that will vary from HIT to HIT ([Learn more](#)).'

The interface features a three-step progress bar: '1 Enter Properties', '2 Design Layout' (the current step), and '3 Preview and Finish'. Below the progress bar, the 'Project Name' is set to 'LSA minicourse', with a note that 'This name is not displayed to Workers.' The 'Frame Height' is set to '600', with a note: 'Height in pixels of the frame your HIT will be displayed in to Workers. Adjust the height appropriately to minimize scrolling for Workers.'

The HTML editor toolbar includes buttons for 'Format', 'Font', 'U' (underline), 'I' (italic), 'B' (bold), 'A' (text color), 'I_x' (background color), alignment options (left, center, right, justified), list options (bulleted, numbered), undo/redo, and a 'Source' button. The editor area contains the following text, which is highlighted with a red border in the image:

This experiment is code named the **red experiment**. You may only take the **red experiment** once per day for payment. If you take this experiment twice in one day, your second one will be rejected. We may have other color experiments up at the same time as this experiment. You may take each color only once per day.

Below the highlighted text, the beginning of another paragraph is visible: 'This is an experiment about English sentences. It will take about 5-8 minutes to complete, and you will be paid \$1.50 for your time. This experiment is being'

Design Layout: Parts of the experiments

Color coding: Amazon isn't made for experiments. It treats each HIT (each ordered list) separately, so workers can take more than one if they want. But we want workers to take only one ordered list per experiment. So I use color coding to link separate HITs (ordered lists) that are related. I tell participants that they can only take a survey of this color once per day. This also lets me post more than one experiment per day if I want.

Format

Font

A*I_x*

Source

This experiment is code named the **red experiment**. You may only take the **red experiment** once per day for payment. If you take this experiment twice in one day, your second one will be rejected. We may have other color experiments up at the same time as this experiment. You may take each color only once per day.

This is an experiment about English sentences. It will take about 5-8 minutes to complete, and you will be paid \$1.50 for your time. This experiment is being conducted by Dr. Jon Sprouse at the University of Connecticut, and has been approved by the UConn Institutional Review Board. Please [click here](#) to download a study information sheet (pdf) that contains the UConn IRB's seal of approval.

Basic Info

What is your biological sex?	Female <input type="radio"/> Male <input type="radio"/>
What is your age?	<input type="text"/>
1. Did you live in the United States from birth until (at least) age 13?	Yes <input type="radio"/> No <input type="radio"/>
2. Did both of your parents speak English to you at home?	Yes <input type="radio"/> No <input type="radio"/>
Which state did you grow up in (two letter abbreviation)?	<input type="text"/>

Instructions

Design Layout: Parts of the experiments

IRB Approval: In the second paragraph, I provide a link to my IRB approval document (called a study information sheet). This is a requirement of my IRB. Yours may be different (but most likely it will be the same).

Basic Info: In the third paragraph, I collect information that may be useful during data analysis (approved by the IRB). Crucially, I ask two questions that help me to screen out non-native speakers. Note that I don't reject them for answering no, they are still paid, that way there is no incentive to lie.

Format

Font

A*I*_x

Source

This experiment is code named the **red experiment**. You may only take the **red experiment** once per day for payment. If you take this experiment twice in one day, your second one will be rejected. We may have other color experiments up at the same time as this experiment. You may take each color only once per day.

This is an experiment about English sentences. It will take about 5-8 minutes to complete, and you will be paid \$1.50 for your time. This experiment is being conducted by Dr. Jon Sprouse at the University of Connecticut, and has been approved by the UConn Institutional Review Board. Please [click here](#) to download a study information sheet (pdf) that contains the UConn IRB's seal of approval.

Basic Info

What is your biological sex?	Female <input type="radio"/> Male <input type="radio"/>
What is your age?	<input type="text"/>
1. Did you live in the United States from birth until (at least) age 13?	Yes <input type="radio"/> No <input type="radio"/>
2. Did both of your parents speak English to you at home?	Yes <input type="radio"/> No <input type="radio"/>
Which state did you grow up in (two letter abbreviation)?	<input type="text"/>

Instructions

Design Layout: Parts of the experiments

Instructions: The next section is the instructions, along with the three instruction/anchor items, which are pre-filled with ratings.

FormatFontU^uIⁱB^bA^a_x

Source

Instructions

In this experiment you will read English sentences, and determine if they sound grammatical to you. By grammatical, we mean whether you think a native speaker of English could say this sentence in a conversation. In other words, do you think it would sound odd for your friends to say this to you, as if they don't speak English natively?

We are not concerned with whether the sentence would be graded highly by a writing teacher: we do not care about points of style or clarity, and we do not care about the grammar rules that you learned in school (who versus whom, ending a sentence with a preposition, etc). Instead, we are interested in whether these sentences could be said by a native speaker of English in normal daily speech.

You will rate the sentence on a scale from 1 (very bad) to 7 (very good). Here are three examples: one that is very bad, one that is in middle of the scale, and one that is very good. You do not need to change these ratings. They are just examples for you.

Example sentences

The was insulted waitress frequently.	very bad	1	2	3	4	5	6	7	very good
		<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Tanya danced with as handsome a boy as her father.	very bad	1	2	3	4	5	6	7	very good
		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
This is a pen.	very bad	1	2	3	4	5	6	7	very good
		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	

Design Layout: Parts of the experiments

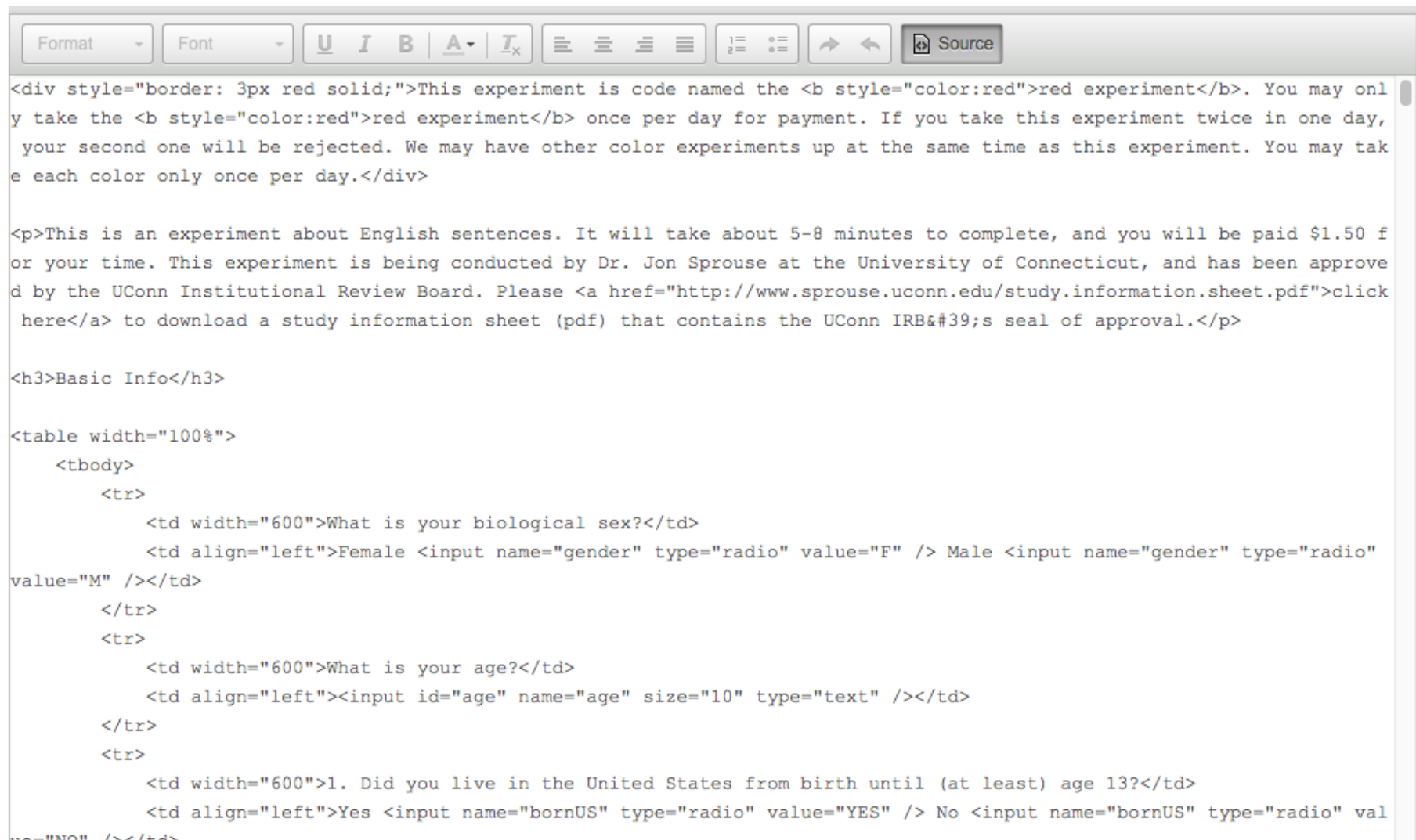
The main experiment: The next section is the experiment itself. Notice that there are symbols on the left: $\${1}$. These are variables used by AMT. They will look for sentences in an input file that match these variables (more soon).

The screenshot shows a web-based experiment interface. At the top is a toolbar with various formatting options like bold, italic, underline, and text color. Below the toolbar, the text "Sentences for you to judge" is displayed. A paragraph of instructions follows: "OK. Now you are ready to rate the rest on your own. There are 31 sentences for you to judge. You must rate all of them in order to be paid for the HIT." Below this, there is a table with four rows, each representing a sentence to be judged. The first row is labeled $\${1}$ on the left. To the right of the label is a rating scale with the text "very bad" on the left, followed by seven numbered circles (1 through 7), and "very good" on the right. The second row is labeled $\${2}$, the third row is labeled $\${3}$, and the fourth row is labeled $\${4}$. Each row has a corresponding rating scale. The interface is designed for participants to rate sentences based on the variables $\${1}$ through $\${4}$.

Sentences for you to judge	very bad	1	2	3	4	5	6	7	very good
$\${1}$		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
$\${2}$		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
$\${3}$		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
$\${4}$		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Design Layout: HTML source

While it is technically possible to create this experiment using the WYSIWYG editor that amazon provides, it is easier to use the HTML source directly. In fact, you can copy in the HTML templates I've provided directly into the source window:



The screenshot shows a web editor interface with a top toolbar containing buttons for 'Format', 'Font', text formatting (underline, italic, bold), text color, background color, list creation, and a 'Source' button. The main area displays the HTML source code for a page. The code includes a red-bordered div, a paragraph about the experiment, a link to a study information sheet, a section header 'Basic Info', and a table with two rows of form fields for gender and age.

```
<div style="border: 3px red solid;">This experiment is code named the <b style="color:red">red experiment</b>. You may only take the <b style="color:red">red experiment</b> once per day for payment. If you take this experiment twice in one day, your second one will be rejected. We may have other color experiments up at the same time as this experiment. You may take each color only once per day.</div>

<p>This is an experiment about English sentences. It will take about 5-8 minutes to complete, and you will be paid $1.50 for your time. This experiment is being conducted by Dr. Jon Sprouse at the University of Connecticut, and has been approved by the UConn Institutional Review Board. Please <a href="http://www.sprouse.uconn.edu/study.information.sheet.pdf">click here</a> to download a study information sheet (pdf) that contains the UConn IRB's seal of approval.</p>

<h3>Basic Info</h3>

<table width="100%">
  <tbody>
    <tr>
      <td width="600">What is your biological sex?</td>
      <td align="left">Female <input name="gender" type="radio" value="F" /> Male <input name="gender" type="radio" value="M" /></td>
    </tr>
    <tr>
      <td width="600">What is your age?</td>
      <td align="left"><input id="age" name="age" size="10" type="text" /></td>
    </tr>
    <tr>
      <td width="600">1. Did you live in the United States from birth until (at least) age 13?</td>
      <td align="left">Yes <input name="bornUS" type="radio" value="YES" /> No <input name="bornUS" type="radio" value="NO" /></td>
    </tr>
  </tbody>
</table>
```

Preview and Finish

The final preview step shows you what the experiment will look like to workers. It doesn't (yet) contain the sentences for your experiment, so those are missing, but this is very close to the final format of the experiment.

1 Enter Properties

2 Design Layout

3 Preview and Finish

Project Name: LSA minicourse

This name is not displayed to Workers.

Language Experiment (code: red): \$1.50, 5-8 minutes

Requester: ucisynlab

Reward: \$1.50 per HIT

HITs available: 0

Duration: 1 Hours

Qualifications Required: Location is US , HIT Approval Rate (%) for all Requesters' HITs greater than or equal to 95 , Number of HITs Approved greater than or equal to 100

HIT Preview

This experiment is code named the **red experiment**. You may only take the **red experiment** once per day for payment. If you take this experiment twice in one day, your second one will be rejected. We may have other color experiments up at the same time as this experiment. You may take each color only once per day.

This is an experiment about English sentences. It will take about 5-8 minutes to complete, and you will be paid \$1.50 for your time. This experiment is being conducted by Dr. Jon Sprouse at the University of Connecticut, and has been approved by the UConn Institutional Review Board. Please [click here](#) to download a study information sheet (pdf) that contains the UConn IRB's seal of approval.

Basic Info

What is your biological sex? Female ☐ Male ☐

What is your age?

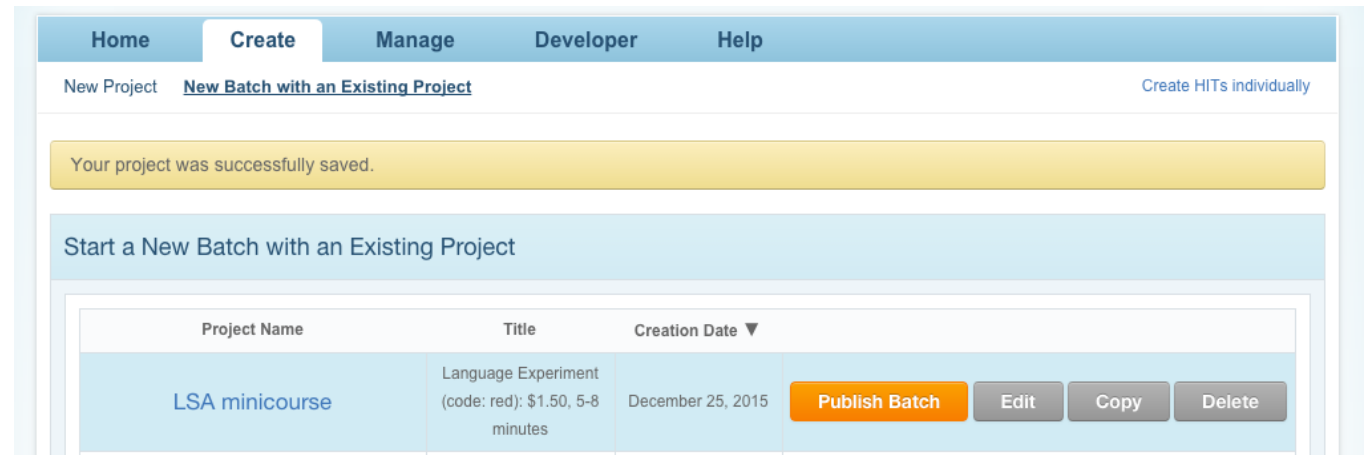
1. Did you live in the United States from birth until (at least) age 13? Yes ☐ No ☐

2. Did both of your parents speak English to you at home? Yes ☐ No ☐

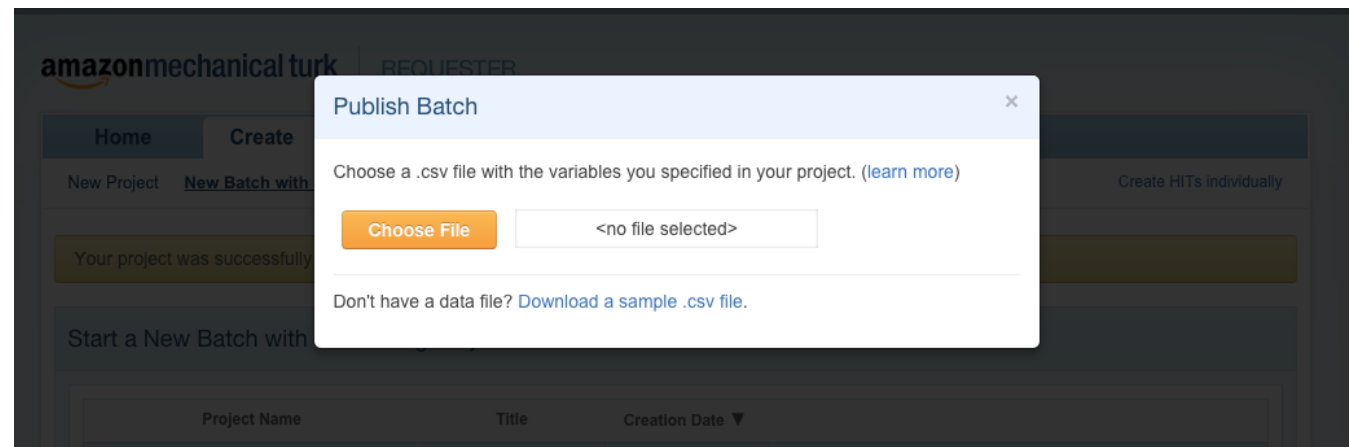
Which state did you grow up in (two letter abbreviation)?

Publish Batch

The next step is to “publish” your “batch” of HITs. You do that by going back to the main “create” page, and clicking the orange button.



When you do that, it is going to ask you to choose a file to upload your HITs. We haven't talked about this input file yet...



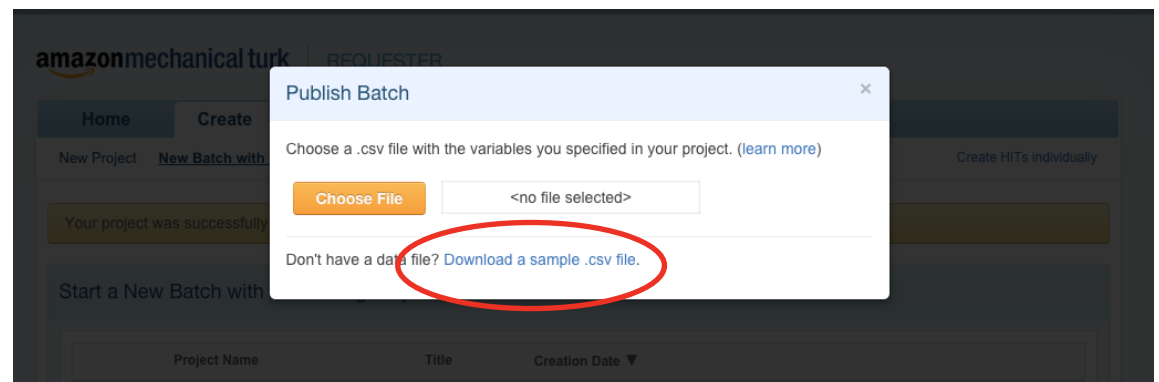
The input file

The input file must be a CSV file. It must contain a **column** for every **variable** in your HIT. There should be one variable in your HIT that tells you which ordered list it is. I call this variable **surveycode**. Then, there should be one variable for every item in your list. In this experiment there are **31 items**, so there are 32 total variables, and therefore 32 columns in the input file.

Each column is named after the variable. Then, you simply need to paste-transpose each ordered list into a row:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	surveycode	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
2		1	She was the	Promise to w	The brother	The children	Ben is hopef	All the men s	They consid	It seems to n	There might	The specime	What does t	There is lik	Laura is mor	Who thinks t	Lily will danc	What does t	There are fir	Who wc
3		2	She was the	Promise to w	The brother	The children	Ben is hopef	All the men s	They consid	It seems to n	There might	The ball perf	Who wonder	There had all	Someone be	What does t	Lloyd Webbe	I hate eating	Jenny cleane	Who thi
4		3	She was the	Promise to w	The brother	The children	Ben is hopef	All the men s	They consid	It seems to n	There might	Richard may	What does t	With that an	Who thinks t	Jenny cleane	I hate eating	Lloyd Webbe	What does t	Someor
5		4	She was the	Promise to w	The brother	The children	Ben is hopef	All the men s	They consid	It seems to n	There might	The ball perf	Who wonder	There are fir	What does t	Lily will danc	Who thinks t	There is lik	Mike prefers	The spe
6																				
7																				

You don't need to construct this file from scratch. AMT will generate a template for your input file that you can download.



Publish Batch

When you upload your input file, AMT will check it to make sure that there are no errors in the coding (that all of the variables match, and that it can read the file.)

The screenshot shows a 'Publish Batch' dialog box with a light blue header and a close button (X) in the top right corner. Below the header, there is a text prompt: 'Choose a .csv file with the variables you specified in your project. ([learn more](#))'. This is followed by an orange 'Choose File' button, a text input field containing 'input.csv', and an orange 'Upload' button. A green success bar with the text 'File validation completed successfully' spans the width of the dialog. Below this, a section titled 'Validated' in green text contains a table of file details. At the bottom, there is a link: 'Don't have a data file? [Download a sample .csv file.](#)'

Validated	
File Name:	input.csv
File Validated:	Yes
File Size:	5.78 KB
Line Count:	5

Publish Batch

AMT will then show you a new preview of your HITs, this time with the real sentences included.

HIT Preview

Sentences for you to judge

OK. Now you are ready to rate the rest on your own. There are 31 sentences for you to judge. You must rate all of them in order to be paid for the HIT.

She was the winner.

very bad 1 2 3 4 5 6 7 very good
☐ ☐ ☐ ☐ ☐ ☐ ☐

Promise to wash, Neal did the car.

very bad 1 2 3 4 5 6 7 very good
☐ ☐ ☐ ☐ ☐ ☐ ☐

The brother and sister that were playing all the time had to be sent to bed.

very bad 1 2 3 4 5 6 7 very good
☐ ☐ ☐ ☐ ☐ ☐ ☐

The children were cared for by the adults and the teenagers.

very bad 1 2 3 4 5 6 7 very good
☐ ☐ ☐ ☐ ☐ ☐ ☐

Publish Batch

Finally, it will show you a summary page that includes all of the information about the HIT, including how much money it will cost you.

You need a credit card to fund your account to actually run the experiment.

Batch Summary

Batch Name: LSA minicourse 1

Description: You must be a native speaker of

Batch Properties	
Title:	Language Experiment (code: red): \$1.50, 5-8 minutes
Description:	You must be a native speaker of US English to participate in this experiment. You will be asked to rate the grammaticality of sentences. You may only take "code name: red" once per day for payment.
Batch expires in:	1 Days
Results are auto-approved and Workers are paid after:	1 Days
Workers must meet the following Qualifications to work on these HITs:	Location score is <u>US</u> HIT Approval Rate (%) for all Requesters' HITs score greater than or equal to 95 Number of HITs Approved score greater than or equal to 100

HITs

Number of HITs in this batch:	4
Number of assignments per HIT:	x 8
Total number of assignments in this batch:	32

Cost

Reward per Assignment:	\$1.50
	x 32 (total number of assignments in this batch)
Estimated Total Reward:	\$48.00
Estimated Fees to Mechanical Turk:	+ \$9.60 (fees paid to Mechanical Turk) (fee details)
Estimated Total Cost:	\$57.60 (this is the amount that will be deducted from your Available Balance when you click "Publish HITs")

Your Available Balance:

\$0.00 (before clicking "Publish HITs")

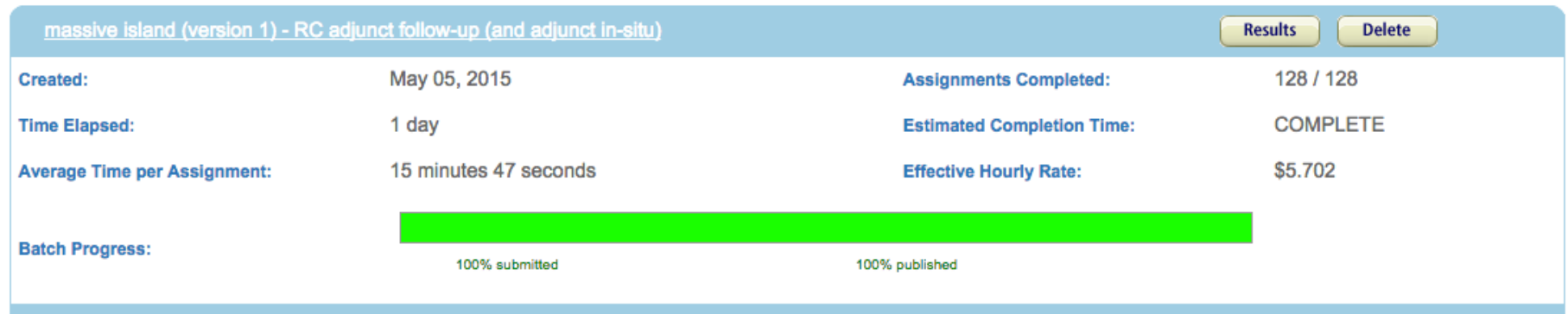
Your Projected Balance:

\$57.60 (after clicking "Publish HITs")

Your account does not have sufficient funds to publish these items. Click [here](#) to fund your account.

Manage: While the experiment is running

While the experiment is running, you can watch its progress under the Manage tab. You will see a progress bar like this:



Pro Tip: You must associate your AMT account with an email address. While the experiment is running, you should be actively monitoring that email address. Workers who run into problems (e.g., accidentally submitting the survey before it is complete) will email you. If you don't respond, they will leave you negative feedback on sites like Turkopticon (a website where workers leave reviews for requesters).

Another tip: incomplete surveys

Workers are very protective of their **approval rates** - the proportion of HITs that are approved. They need to maintain high approval rates to qualify for the best paying HITs.

The problem is that the **only way to not pay a worker** is to reject their HIT. So, if they accidentally submit an unfinished survey, you either have to pay them for the unfinished work, or reject them. Nobody is happy about either option. That is why they email you when this happens. They want to find a solution.

If you are feeling nice, you can do the following. Look at the incoming results by clicking the results button at the top right of the progress bar. Find the worker's incomplete HIT (usually it is the only one with empty responses, but you can also use their worker ID number). Then send them the ordered list in an excel spreadsheet, and tell them that if they finish it in the excel spreadsheet, and send it back to you, then you will approve their HIT. It takes work on your end, but it gets you the data, and saves them a rejection.

The results view

If you want to see the results as they are coming in, you can, by clicking the results button:

massive island (version 1) - RC adjunct follow-up (and adjunct in-situ)

ResultsDelete

Created:

May 05, 2015

Assignments Completed:

128 / 128

Time Elapsed:

1 day

Estimated Completion Time:

COMPLETE

Average Time per Assignment:

15 minutes 47 seconds

Effective Hourly Rate:

\$5.702

Batch Progress:

100% submitted

100% published

This generates a (super wide) table of the results. If you want, you can approve results from this view, you can reject results from this view, you can sort by various properties (workerID, completion time, etc), etc. **Remember to approve the results for all workers after the experiment is finished.**

ApproveReject											
HIT ID ▲	Worker ID	Lifetime Approval Rate	Input.Surveycode	Input.1	Input.2	Input.3	Input.4	Input.5	Input.6	Input.7	Input.8
32204AGAABCLUCJXWDTSSSEMRU3HG4	A2BA9Y6VGW6WS1	100% (2/2)	55.1	She was the winner.	Promise to wash, Neal did the car.	The brother and sister that were playing all th...	The children were cared for by the adults and L...	Ben is hopeful for everyone have all you do to eaten attend.	All the men seem to a teacher of Chris geeky.	They consider to me that Robert can't be trusted.	It seems to me that Robert can't be trusted.
32204AGAABCLUCJXWDTSSSEMRU3HG4	AW0MG225VXWCN	100% (17/17)	55.1	She was the winner.	Promise to wash, Neal did the car.	The brother and sister that were playing all th...	The children were cared for by the adults and L...	Ben is hopeful for everyone have all you do to eaten attend.	All the men seem to a teacher of Chris geeky.	They consider to me that Robert can't be trusted.	It seems to me that Robert can't be trusted.

Input.30	Input.31	1	10	11	12	13	14	15	16	17	18	19	2	20	21	22	23	24	25	26	27	28	29	3	30	31	4	5	6	7	8	9	A1	A2	A3	Age	Born State	Born Us	Gender	Parents English
I understand the employee who will call a repor...	The ball perfectly rolled down the hill.	7	3	4	6	6	4	5	3	6	4	4	4	6	4	4	5	4	7	7	4	7	3	6	4	7	7	4	7	4	6	3	1	4	7	36	ky	YES	F	YES
I understand the employee who will call a repor...	The ball perfectly rolled down the hill.	7	5	4	3	7	7	4	5	7	5	2	3	5	5	3	7	3	7	6	4	7	3	6	5	6	7	3	5	6	7	4	1	4	7	28	LA	YES	M	YES
I understand	The ball																																							

There is also a button to generate a CSV of the results. Ultimately, when the experiment is finished, this is what you are going to want to do.

Exercise 5

Part 1: Complete the CITI training for working with human subjects.

This is required by UConn for you to run experiments using human participants. You only need to do this once. If you've already completed it, move on to part 2.

<https://www.citiprogram.org/>

You must complete the course called [Human Subjects Research Course, Social/Behavioral Research](#).

Part 2: Put our experiment up on the mechanical turk sandbox.

You have everything you need to put the experiment up online.

<https://requestersandbox.mturk.com/>

Submit the following to me: (i) a mechanical turk input file (csv) for our materials, (ii) a screenshot of the batch summary page that they give you right before you publish, and (iii) a screenshot of the list of available experiments that shows your experiment available.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1: Design

Section 2: Analysis

Section 3: Application

What is pre-processing

Pre-processing is any manipulation you do to your data before the actual statistical analysis.

For organizational purposes, I am going to lump two types of pre-processing together in this section, though they are distinct in principle.

One type of pre-processing that you will always have to do is **data formatting**. You need to arrange your data in such a way that you can easily do the analysis (modeling, plotting, etc) that you need to do. Data formatting doesn't change your data, so you should feel free to do whatever you need to do to make things work.

Another type of pre-processing that you may have to do is **data transformation**. This is where you take your raw data and perform some number of calculations to derive new data (e.g., averaging, z-score transformations, log transformations, or in EEG, filtering).

Data transformations should always be theoretically justified, and if possible, kept to a minimum. **They change your data!**

I am going to cover both in this section because (i) they both use R, and (ii) the result is a data file that you can use for statistical analysis and plotting.

Formatting your data

Two formats: wide and long

When humans enter experimental data into a table, they tend to do it in **wide format**. It is a very intuitive format for data.

In **wide format**, each row represents a **participant**. Each column represents something about the participant, such as a **property** or an **experimental trial**. And each cell contains the value for that property.

	age	trial 1	trial 2	trial 3	trial 4
participant 1	18	2	7	6	1
participant 2	22	2	6	5	1
participant 3	23	3	7	4	2

Wide format has some uses in computer-aided analysis, typically as part of a calculation of a new value; but it is not the dominant format. I would say that I use wide format less than **5%** of the time. **95%** of the time, the analyses that you will perform will call for **long format**.

Two formats: wide and long

When humans enter experimental data into a table, they tend to do it in **wide format**. It is a very intuitive format for data.

In **wide format**, each row represents a **participant**. Each column represents something about the participant, such as a **property** or an **experimental trial**. And each cell contains the value for that property.

	age	Wide format grows longer by one row every time you add a participant, and by one column every time you add a trial/response/measurement/property. Because many experiments will have more trials/responses/properties than participants, the table will often look like a rectangle whose width is greater than its height.
participant 1	18	
participant 2	22	
participant 3	23	

Wide format has some uses in computer-aided analysis, typically as part of a calculation of a new value; but it is not the dominant format. I would say that I use wide format less than **5%** of the time. **95%** of the time, the analyses that you will perform will call for **long format**.

Two formats: wide and long

The primary format for computer-aided statistical analysis is **long format**. At first, long format is less intuitive than wide format, but you will very quickly learn to appreciate its logic.

In **long format**, each row represents a **trial**. Each column represents a property of that trial, such as the ID of the participant in that trial, the condition of that trial, the item used in that trial, and ultimately the rating (or response) that came from that trial.

	participant	age	condition	item	rating
trial 1	1	21	long.island	1	1
trial 2	1	21	short.non	4	7
trial 3	1	21	long.non	2	5
trial 4	1	21	short.island	3	5

Two formats: wide and long

The primary format for computer-aided statistical analysis is **long format**. At first, long format is less intuitive than wide format, but you will very quickly learn to appreciate its logic.

In **long format**, each row represents a **trial**. Each column represents a property of that trial, such as the ID of the participant in that trial, the condition of that trial, the item used in that trial, and ultimately the rating (or response) that came from that trial.

	participant	age
trial 1	1	21
trial 2	1	21
trial 3	1	21
trial 4	1	21

Long format is called “long” because it leads to really long tables. Each subject will have a number of rows equal to the number of trials in the experiment. So 40 participants x 100 items = 4000 rows. Both formats grow longer with additional participants, but long format grows longer much faster. And long format grows longer with additional trials (wide format grows wider with additional trials).

AMT gives you results in wide format (IBEX gives results in its own hybrid format)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	HITId	HITTypeId	Title	Description	Keywords	Reward	CreationTime	MaxAssignm	RequesterAn	AssignmentC	AutoApprov	Expiration	NumberOfSi	LifetimeInSe	AssignmentId	WorkerId	Assignment
2	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			31LVTDXBL7	A3OV174HQ	Approved
3	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			35K3O9HUA1	ADOB8J5ANJ	Approved
4	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			38F710A9G1	ARLSOH5YM	Approved
5	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3F6KKYWMN	AETIZKQNUS	Approved
6	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3KMS4QQVK	AYKZ9H4BNV	Approved
7	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3NL0RFNU0F	AS1QMPXIT1	Approved
8	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			35KRO2GZ71	ARNVB51ESK	Approved
9	3ATYLI1PRT2	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3YGXWBAF7	A2NJ7N8INZ	Approved
0	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			308XBLVESI4	AU2NVT51E7	Approved
1	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			32SVAV9L3F	ALEJV7D94ZI	Approved
2	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			39RP059MEI	A3STVJG6VL	Approved
3	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3QL2OF5M9	AMZE7O09X	Approved
4	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3QXNC7EIPV	A177EXELD1	Approved
5	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3R2UR8A0IA	A6INY1UVFY	Approved
6	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3RANCT1ZVF	A2AMI7BVAL	Approved
7	388FBO7JZR	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			3RKNTXVS3N	AM155T4U3	Approved
8	3BFFODJK8X	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			34T446B1C0	AHDBHMH3	Approved
9	3BFFODJK8X	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			352YTHGROV	A1TFWB4P6I	Approved
0	3BFFODJK8X	36SUH4ZPJU	Language Ex	You must be	survey, expe	\$1.50	Mon Apr 20	8	BatchId:1909	3600	86400	Tue Apr 21 11:17:35 PDT 2015			35LDD5557A	A2CGAOF4G	Approved

But that is ok, we can use R to convert the results to long format.

Exercise 6: convert wide format AMT data to long format

In the document exercise.6.pdf, I give you a list of functions that you can (and probably will) use to do this. The trick with this, and any script you write, is to start by writing out the steps that you want to achieve in plain English. Then you can figure out how to make R perform those steps. In this case, you are re-arranging the data. So figure out how you would do that (with cutting and pasting, and filling in labels), and then convert those steps to R.

There are two solution scripts on the website

I've created two scripts that can convert wide AMT data to long format: **convert.to.long.format.v1.R** and **convert.to.long.format.v2.R**.

Version 1 works very similarly to the way you would convert from wide to long if you were cutting and pasting in excel. It cuts away different pieces of the dataset, stacks the columns that need to be stacked, and pastes them back together.

Version 2 uses functions from two packages that were specifically designed to make manipulating data easier (including converting from wide format to long format). These packages are **tidyr** and **dplyr**. These two packages are now available in a single package called **tidyverse**. Tidyverse also includes other packages that are useful for data manipulation and visualization, including **ggplot2**, which we will use next time to make plots!

We will go through these later so that you can see what the code looks like. You can also add them to your growing library of R scripts (and use them in future experiments).

Next step: adding item information

```
> dataset.long
  subject survey order judgment
218 A2NJ7N8INZOB00      1      1      7
219 A2NJ7N8INZOB00      1      2      1
220 A2NJ7N8INZOB00      1      3      4
221 A2NJ7N8INZOB00      1      4      1
222 A2NJ7N8INZOB00      1      5      1
223 A2NJ7N8INZOB00      1      6      5
224 A2NJ7N8INZOB00      1      7      2
225 A2NJ7N8INZOB00      1      8      6
226 A2NJ7N8INZOB00      1      9      1
227 A2NJ7N8INZOB00      1     10      3
228 A2NJ7N8INZOB00      1     11      5
229 A2NJ7N8INZOB00      1     12      5
230 A2NJ7N8INZOB00      1     13      7
231 A2NJ7N8INZOB00      1     14      7
232 A2NJ7N8INZOB00      1     15      7
233 A2NJ7N8INZOB00      1     16      6
234 A2NJ7N8INZOB00      1     17      7
235 A2NJ7N8INZOB00      1     18      7
236 A2NJ7N8INZOB00      1     19      2
237 A2NJ7N8INZOB00      1     20      6
238 A2NJ7N8INZOB00      1     21      6
```

The csv file called **results.long.format.no.items.csv** contains the results of converting from wide to long format.

Although it is technically possible to upload item keys to AMT, and then have the AMT results contain item keys, I typically don't do that (and IBEX cannot do that). AMT didn't have the item or condition labels, so we need to add that ourselves.

This means we need to add the item keys to our long format dataset.

This is where our **keys.csv** file comes into play. We are going to use it to add item codes to the dataset. Then, we can use R to convert the item codes into condition codes and factors for each item!

I have already written a script to add item keys, derive condition codes, and derive factor/level codes. It is called **add.items.conditions.factors.r**.

Next step: Correcting scale bias (z-scores)

Recall that **pre-processing** is any manipulation you do to your data before the actual statistical analysis. As a general rule, you should keep the pre-processing to a minimum (pre-processing changes your data!). But there is at least one property of judgment data that people agree should be corrected before analysis: **scale bias**.

Scale Bias: Different participants might choose to use a scale in different ways.

There are two types of scale bias that are relatively straightforward to correct.

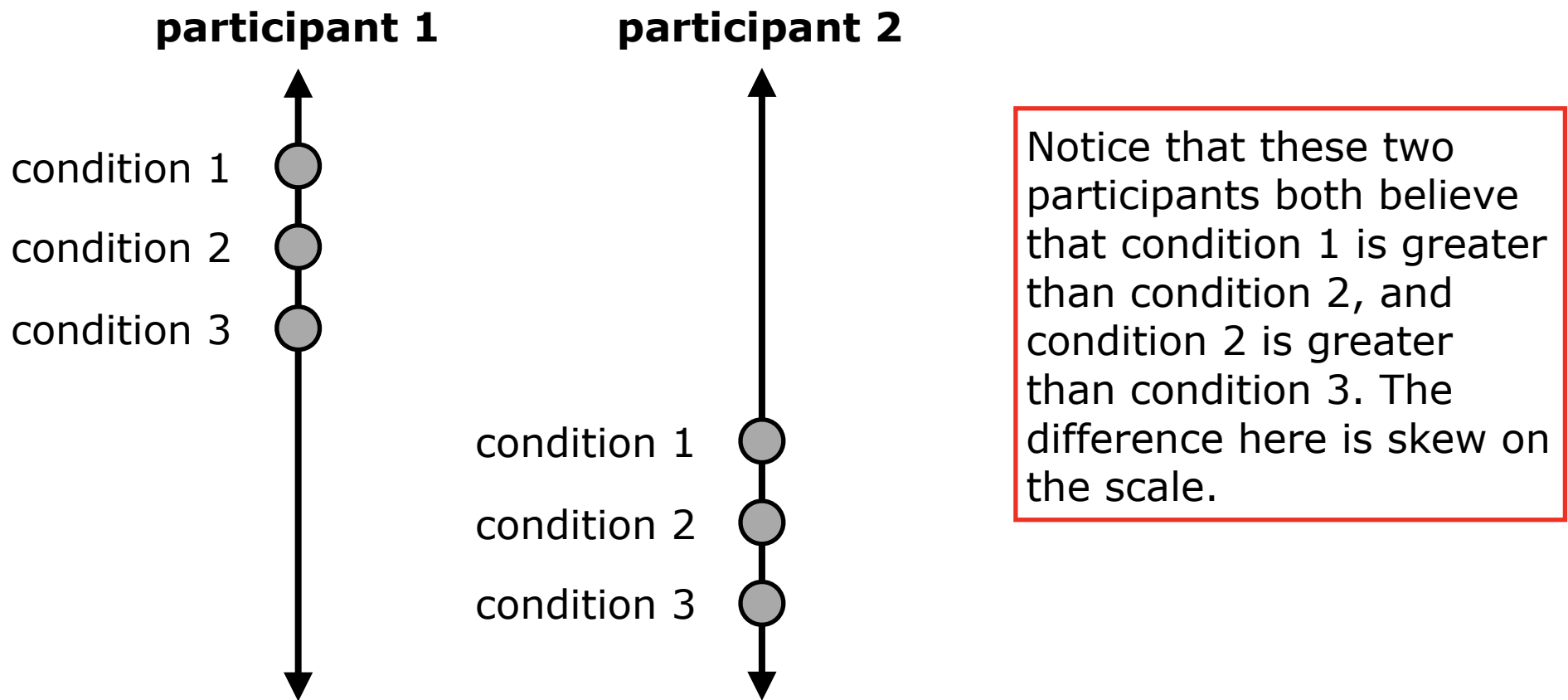
Skew: Different participants might use different parts of the scale, such as one using the high end, and another the low end).

Compression/Expansion: Different participants might use different amounts of the scale, such as one using only 3/7 responses, and another using the full 7 responses.

PRO TIP: The best defense against scale bias is a well-designed experiment. Try to have the mean rating of your items equal the mid-point of your scale. Make sure all of your responses will be used, will be used an equal number of times!

Here is an example of **skew**

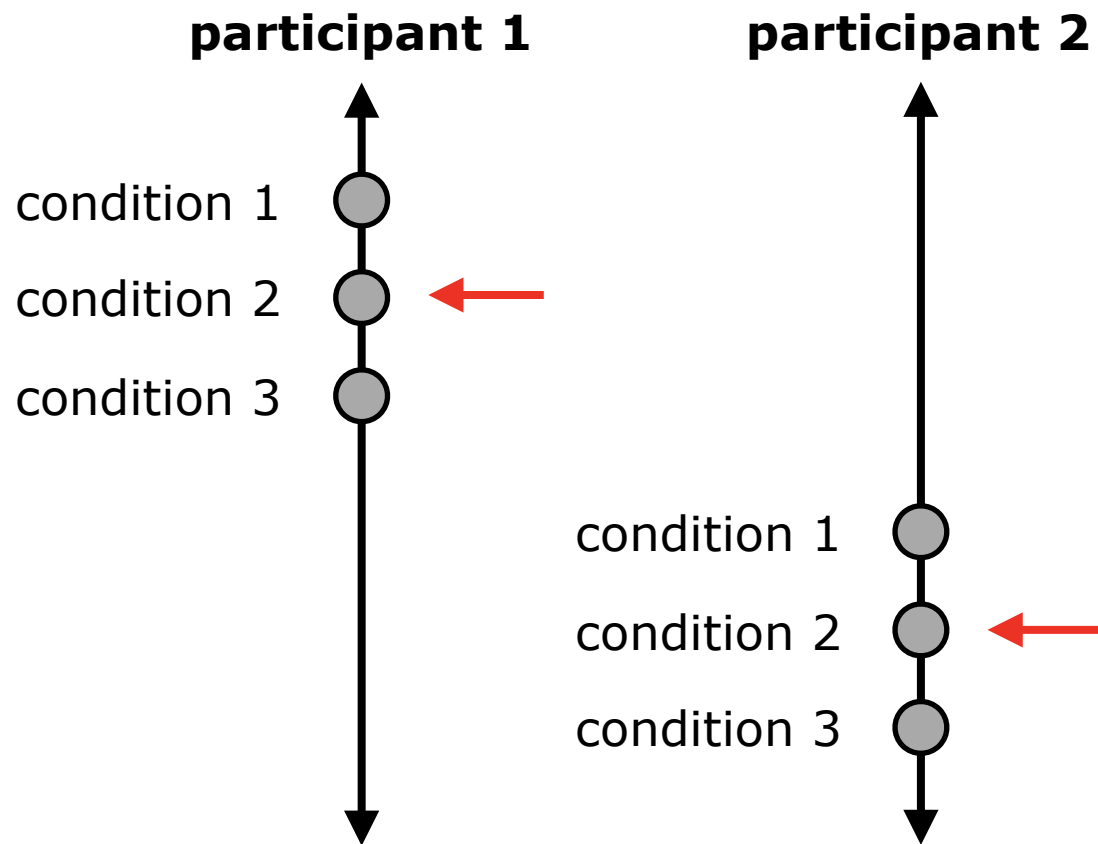
If two participants are skewed in different ways, we are basically saying that their two private scales are separated from each other, but not because of meaningful differences in their judgments.



If you were to average their results together, you would end up with the same pattern, but there would be a lot of (non-meaningful) variability (or spread) in your data.

Skew can be corrected with **centering**

The way to correct skew is to identify the **center point of all of the ratings** for each participant, and then **align** the center points.

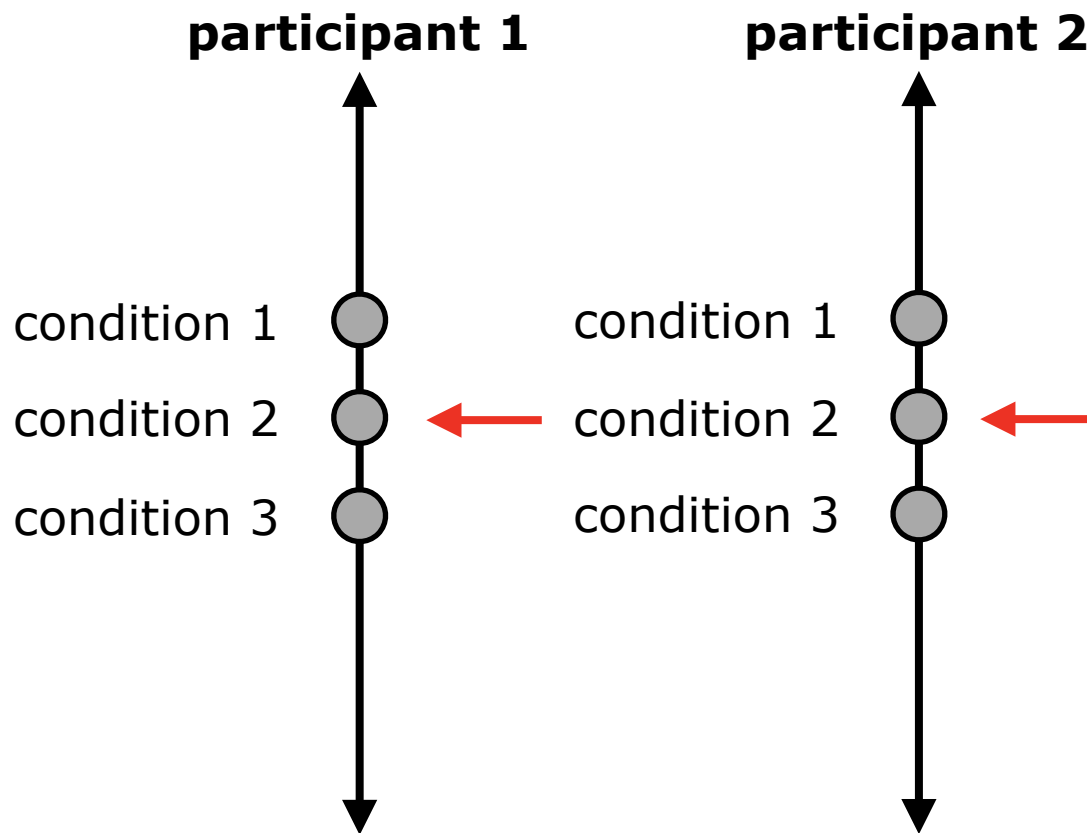


You have several options for choosing a center point (we will discuss these next time). The **mean** is the most common choice for centering to remove scale bias.

Important note: In this toy example, the center point is also the rating for a condition. This is not necessary. The mean of all of the ratings of a participant could be a number that isn't the rating of a condition.

Skew can be corrected with **centering**

If we align the center points, the same relationship holds among the conditions, and the same distances hold between the conditions. But the variability due to skew is removed.



One way to align the centers is to subtract the mean from each data point:

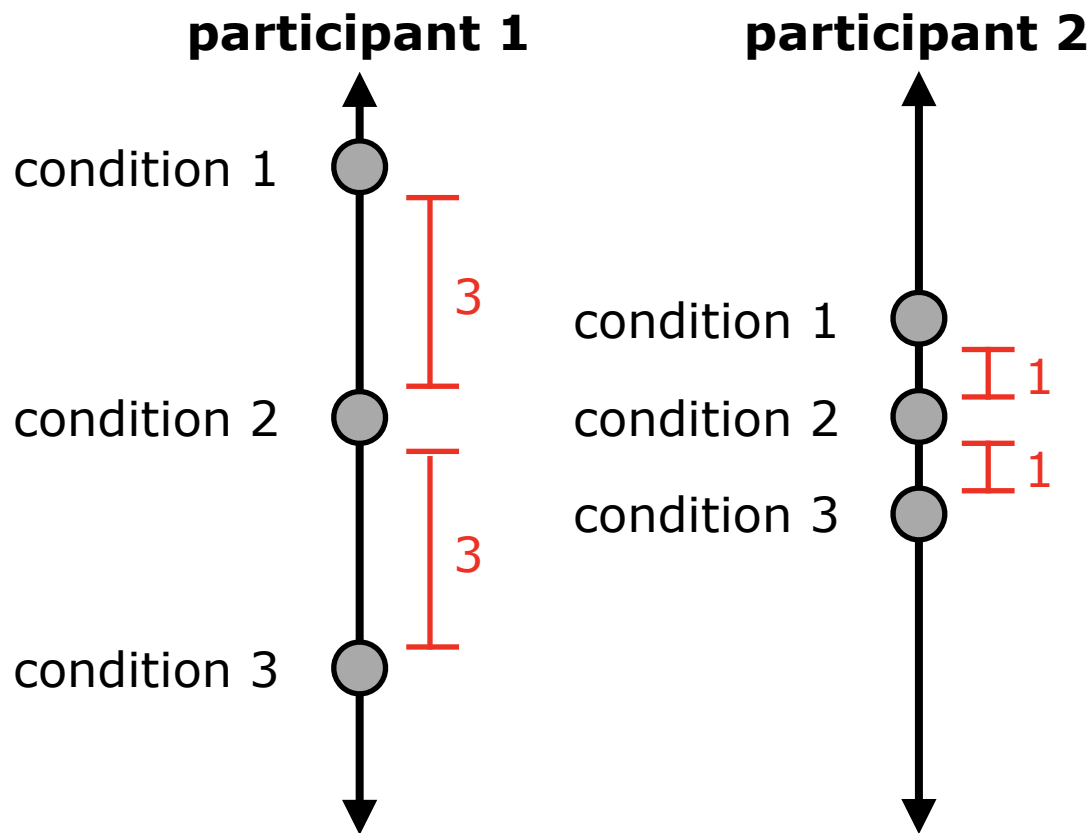
new point = old point - mean

This has the effect of turning the mean in 0, and arranging the points around 0 based on their distance from the mean.

This process is called **mean centering**; obviously, if you used a different center point, it would be a different kind of centering.

Here is an example of compression/expansion

Here we have two participants that use different amounts of the scale. This means that the distances between the points is different for each of them. Notice that their centers are the same, so there is no skew.



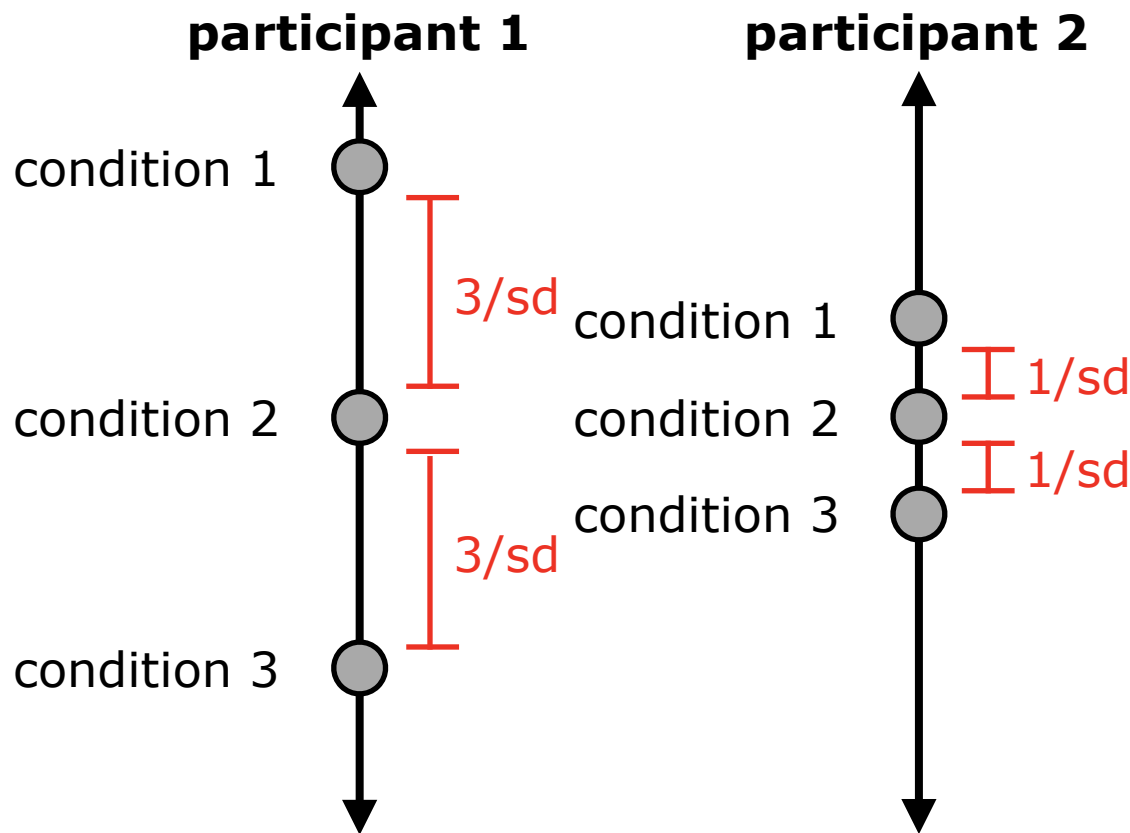
If we take the mean of these conditions, the means will be somewhere between the two, and there will be variability (spread) in our data set.

But looking at the points, we see the same relative position, and we see that the distance differences affect all of the points. This suggests a scale bias issue, not a meaningful difference.

If we use the **mean** as a center to calculate distances, what we can see is that each participant is characterized by very different **distances from the mean**.

We need a **standard unit of distance**

Again, you have a number of choices for a standard unit of distance. The most common choice is to use the participant's **standard deviation**.



We will discuss standard deviation in detail next class.

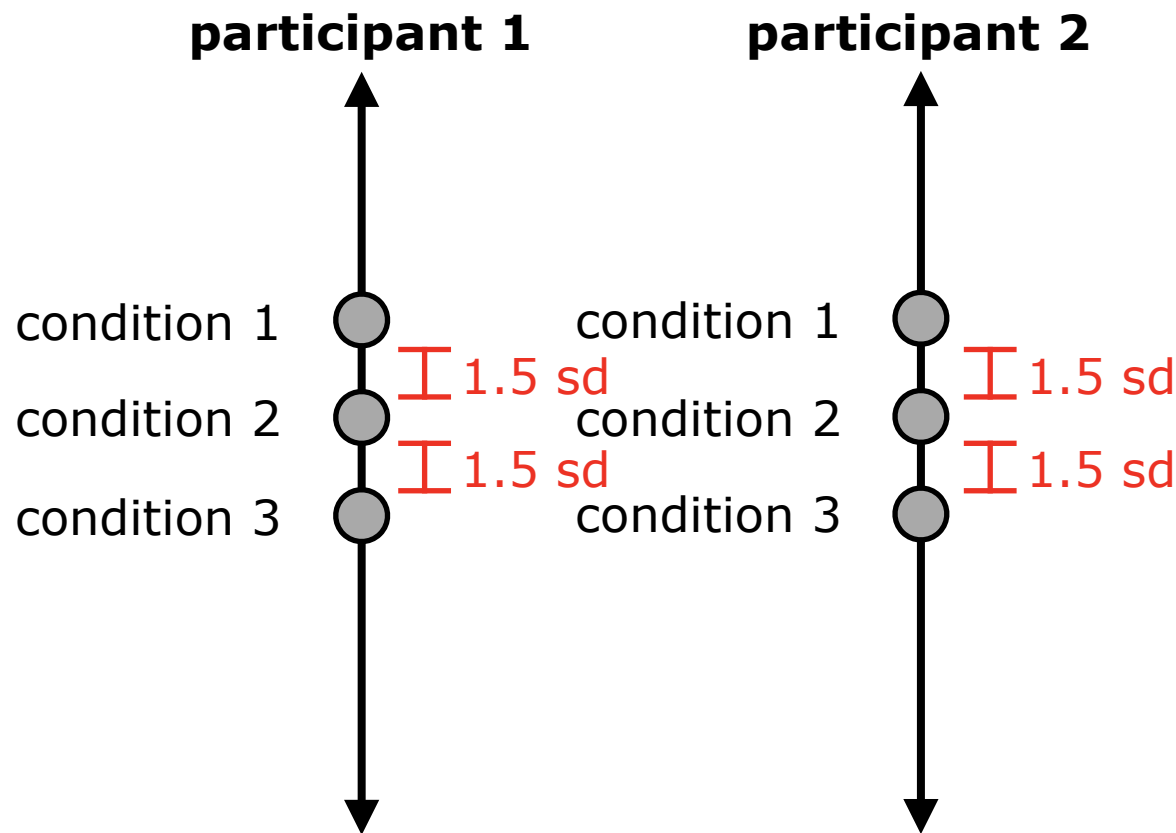
For the curious, it is the root mean square error (square the distances from the mean, sum them, divide by n , then take the square root).

But for now, just think of it as an average measure of the distance that each data point is from the mean (for each participant).

We can use the standard deviation to standardize the distances for each participant. Basically, we just divide each distance by the standard deviation!

We need a **standard unit of distance**

The result is that both participants use the same distance unit, “standard deviation units”. Because it was simple unit conversion (division), the structure of the data is unchanged.



By using a standard unit of distance, we can see the structure of the data without the interfering compression/expansion issue.

Here we see that both participants share the same center, and they share the same relative distance from the mean (I just made these numbers up).

One way people talk about this is that participants 1 and 2 are using different scales. But finding a common unit of distance, we can put them on the same scale. This can be done for any two scales — even qualitatively different ones.

Putting both steps together: z-scores

The **z-score transformation** combines both steps: it centers the scores around the mean, and it converts the units to standard deviations.

$$Z = \frac{\text{the judgment} - \text{participant's mean}}{\text{participant's standard deviation}}$$

Pro Tip 1: It is crucial that the z-score transformation is applied to each participant separately. That way you are eliminating the scale bias for each individual participant. If you z-score transform the entire sample at once, it won't eliminate any scale bias, it will just convert the values to z-units (think about this offline to see why!).

Pro Tip 2: If your goal is to eliminate scale bias, you have to use all of the data points from the participant (target items and fillers, not just the target items). I would also recommend not including the practice items. The practice items are there to help people learn how to use the scale. So those items might have different bias properties than the later items. So, my suggestion is to perform the z-score transformation using all of the item except the practice items.

Some thoughts about z-scores

Advantages:

The primary benefit of the z-transformation is that it will help to eliminate the most common forms of scale bias, making the comparison of judgments across participants less noisy.

This reduction in noise results in a noticeable increase in statistical power (scale bias introduces additional variance into the model).

The z-score scale is also intuitive: 0 represents the mean, the sign of the score indicates if it is above or below the mean, and the number represents the number of standard deviations!

Finally, it is relatively easy to compute. So all we need to do is apply it to each participant.

Disadvantages:

Because the z-transformation does not alter any of the information in a data set (it is called a linear transformation), there are not many risks at all.

The only real risk would be if the the skew that you saw as scale bias was actually meaningful. You need to be sure it is not meaningful. Typically, if each participant saw the same items, then any bias is an artifact; but if you give participants wildly different items, scale differences might be meaningful.

Exercise 7: Adding z-scores to our dataset

Exercise 7: add z-scores to the dataset

In the document `exercise.7.pdf`, I give you a list of functions that you can (and probably will) use to do this.

The trick with this is to figure out how you would calculate z-scores for each participant, then figure out how to make R perform these calculations.

My scripts: Adding z-scores to our dataset

On the website I have two versions of the z-score script. Once again, I've made two: a longer one and a shorter one.

The first is the long way: [add.z.scores.v1.r](#). This calculates z-scores the same way you would do it if you were using excel by hand.

The second takes advantage of two built-in functions in R: `split()` and `scale()`. Once you understand how the z-score works, [add.z.scores.v2.r](#) will save you time.

[add.z.scores.v1.r](#)

```
1 #####
2 #This script takes the output of "add.items.conditions.factors.r" and calculates the
3 #The z-score transformation corrects for some forms of scale bias.
4 #This is the long (step-by-step) way of doing this. It is exactly what you'd do if y
5 #####
6
7 #NOTE: This script assumes your dataset is called dataset.factors, which is the o
8
9 #remove practice items
10 #By definition, we don't want these contributing to the scale bias removal proces
11 out their scale.
12 dataset.working=subset(dataset.factors, condition != "7P" & condition != "6P" & c
13 != "2P" & condition != "1P" & condition != "7p" & condition != "1p")
14
15 Nsubjects = length(levels(dataset.working$subject))
16
17 Nitems = nrow(dataset.working)/Nsubjects #This is the number of experim
```

[add.z.scores.v2.r](#)

```
1 #####
2 #This script takes the output of "add.items.conditions.factors.r" and calculates th
3 #The z-score transformation corrects for some forms of scale bias.
4 #This is the quicker way, using split() and unsplit()
5 #####
6
7 #NOTE: This script assumes your dataset is called dataset.factors, which is the o
8
9 #####
10 #We still need to remove practice items; the fancy commands below work on whi
11 #By definition, we don't want these contributing to the scale bias removal proces
12 out their scale.
13 dataset.working=subset(dataset.factors, condition != "7P" & condition != "6P" & c
14 != "2P" & condition != "1P" & condition != "7p" & condition != "1p")
15
16 #sort by subject in ascending order (descending would be -subject)
17 #this is NECESSARY, but the choice between ascending/descending is not critical
18 #the dataset should already be sorted this way. But just in case it isn't, we do it
```


Removing outliers

An **outlier** is an experimental unit (either a participant or a judgment) that is substantially different from other experimental units. Outliers add noise to your data, which can lead to errors (a null result when there is a real difference, or a false positive result when there is no real difference).

There are a number of ways to deal with outliers. Here I will review three common approaches, in the order in which I recommend them (with colors indicating the danger!).

1. **Run more participants.** This will diminish the impact of an outlier. The nice thing about this approach is that you don't have to make any assumptions. You just report the data you have with no changes.
2. **Use gold-standard questions.** If you include sentences with known ratings, you can identify participants who rate these known sentences substantially differently (than expected), and eliminate those participants. There are two nice properties of this approach: (i) it does not rely on the experimental items, and (ii) you remove entire participants.
3. **Trim (or Windsorize) the data.** You can also look at the distribution of judgments for each experimental item, and remove outliers. The risk here is that bias can creep in (you are looking at the experimental items directly, so you could make choices that bias toward one outcome or another).

An approach that uses gold-standard questions

My preferred approach is to simply **run more participants**. AMT makes this very easy. The only downside is that it increases the cost of the experiment.

If I can't run many participants, my second choice is to use gold-standard questions. In the design that we have been using, all of our filler items can serve as gold-standard questions because we pre-tested the fillers, and know exactly what their expected rating should be (the mean or mode).

Of course, there is some noise in judgments, so we don't expect every participant to give the precise mean rating for each filler. So we don't just want to eliminate everyone whose response differs from expected value. That would probably eliminate everybody. Remember, we expect variation in humans.

So what we want to do is **quantify the variation** that each participant shows from the expected judgments, and then **eliminate any participant that shows substantially more variation** than the other participants.

One common way to do this is with a **sum of squares** measure of error.

Calculating variation using sum of squares

We run the calculation for each participant separately.

First, we calculate the difference between the expected value of each filler and the value that we observed from the participant

We can't sum this value directly, because it could be either positive or negative, and the two will cancel each other out (given the appearance of good fit). So, next, we square those difference scores to eliminate the negative signs.

Finally, we sum the squared differences to obtain a **final variation score** for the participant.

item	expected	observed	difference	difference ²
1F	1	2	1	1
4F	4	2	-2	4
6F	6	7	1	1
		sum:	0	6

Setting a criterion for exclusion

After we run this for each participant, we will end up with a distribution of scores like this (these are derived from `identify.and.remove.outliers.r`):

	subject	sumsquares
2	AU2NVT51E7	7
3	A2LQ33BQ8K	9
4	AM155T4U3	10
5	AMZE7009X	12
6	A3N5L136KK	13
7	ARLSOH5YM	13
8	AS1QMPXIT1	15
9	A6POINAX7C	16
10	ADVIE0ZHLW	16
11	A2AMI7BVAL	17
12	ADOB8J5ANJ	19
13	A2J7PEUIO2Y	20
14	AKM7BAAH9	20
15	A2CGAOF4G	23
16	A6INY1UVFY	23
17	A1TFWB4P6I	24
18	AYKZ9H4BNV	24
19	AVI7UDWV0	26
20	A1XOXGWB4	27
21	A1ZR0Q2OU	27
22	AETIZKQNUS	29
23	AKG7VBOY9	29
24	ALEJV7D94ZI	34
25	A3OV174HQ	35
26	AHDBHMH3A	39
27	ARNVB51ESK	39
28	A2V3P1XE33	41
29	A2NJ7N8INZ	47
30	A3STVJG6VL	69
31	A339I8W36K	98

One common way to identify outliers (in general) is to take the **mean** and **standard deviation** of some distribution of values, and then call any value that is some number of standard deviations away from the mean (in either direction) an outlier. Since only high scores are bad here, we need only look in the positive direction.

The number of SDs that you choose determines how many outliers you will have. A low number like 2 will yield more outliers, a high number like 4 will yield fewer.

The **mean** of these values is 27.367

The **standard deviation** of these values is 16.694

So any value **above 60.756** would be an outlier.

There are only two subjects that are above this threshold, so by this procedure there are **two outliers**.

Now, let's look at R and the scripts!

What is R?

R is a programming language designed for statistics. That's it.

It is called R for two reasons: there is a proprietary statistical language called S that serves as a model for R, and the two creators of R are Ross Ihaka and Robert Gentleman.



Why do so many people love R?

It is free, open-source, and cross-platform.

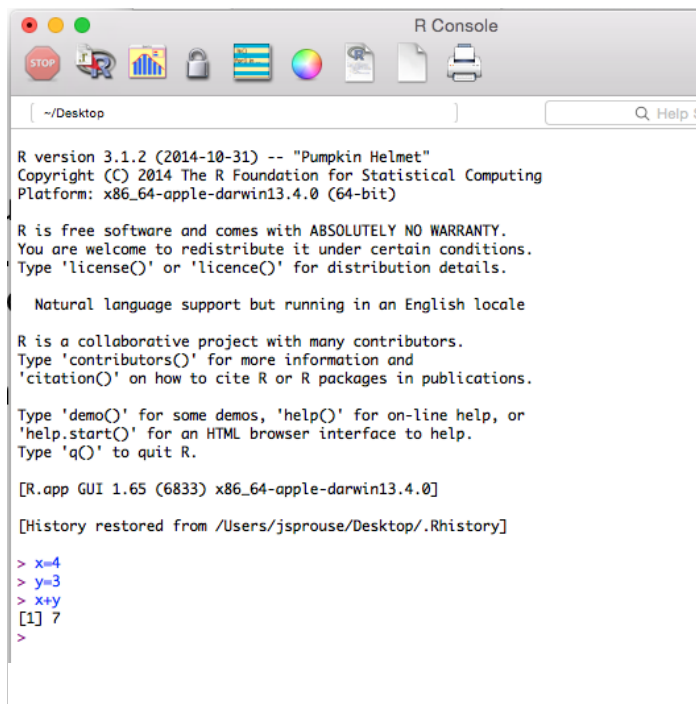
It has a giant user community. Anything you want to do has probably been done before, so there are pre-built packages and internet help groups galore.

It allows you to do three things that you need to do: (i) manipulate data/text files, (ii) analyze your data, and (iii) create publication-quality figures (no, you can't use excel for figures in publications).

Yes, Matlab (proprietary) and Python (free) can do the same things. You can absolutely use those if you prefer. But R is specifically designed for stats and graphics, whereas Matlab is designed for matrix algebra, and python is a general computing language.

Interacting with R: the R console

R is a programming language. You need to find a way to interact with the language. The R-project (the developers of R) provide a “console” to allow you to interact with the R language. You type a command into the console, and the R language implements that command.



```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.65 (6833) x86_64-apple-darwin13.4.0]

[History restored from /Users/jsprouse/Desktop/.Rhistory]

> x=4
> y=3
> x+y
[1] 7
>
```

If you want to save your code, you can type it into a text editor like TextWrangler (Mac) or Notepad++ (Windows). Then you just have to move the text from the editor to the console window to run it (you can copy and paste, or create a shortcut key that does it for you).

```
1 dataset=read.csv(raw.results)
2
3 #Keep the following, discard the rest
4 #WorkerID
5 #AssignmentStatus
6 #surveycode
7 #all responses
8
9 keep.columns=c(
10   grep("WorkerId",colnames(dataset)),
11   grep("AssignmentStatus",colnames(dataset)),
12   grep("Input.surveycode",colnames(dataset)),
13   grep("Answer.",colnames(dataset)) #All responses are prefaced with Answer.
14 )
15
16 data.subset1=dataset[,keep.columns]
17
18 #remove Answer. and Input. from column names
19 colnames(data.subset1)=sub('Answer.',"",colnames(data.subset1))
20 colnames(data.subset1)=sub('Input.',"",colnames(data.subset1))
21
22 #remove subjects that were rejected through the MTurk interface (i.e. not paid)
```

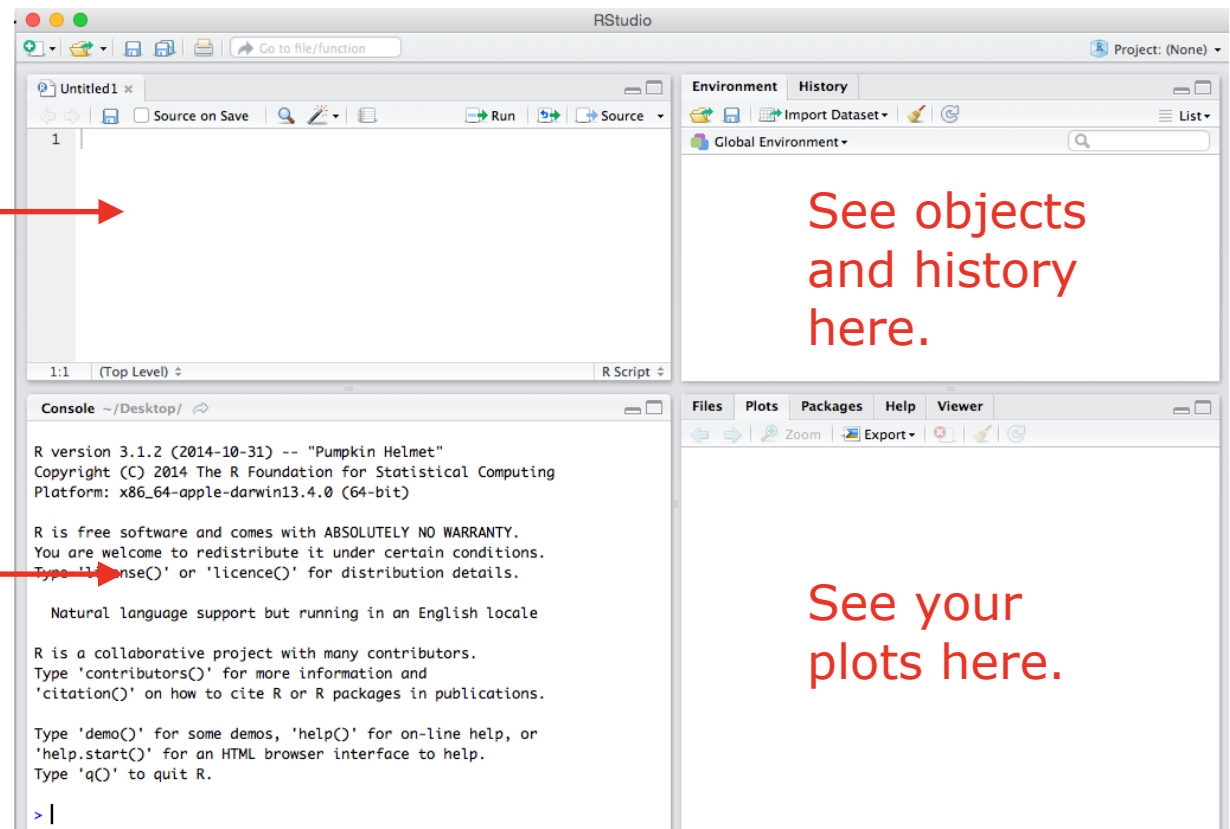
Interacting with R: R Studio

R Studio is a third-party piece of software (free) that allows you to interact with the R language in a single, unified environment.



A text editor for saving your code. →

The R console that runs your commands. →



See objects and history here.

See your plots here.

R is an interpreted language

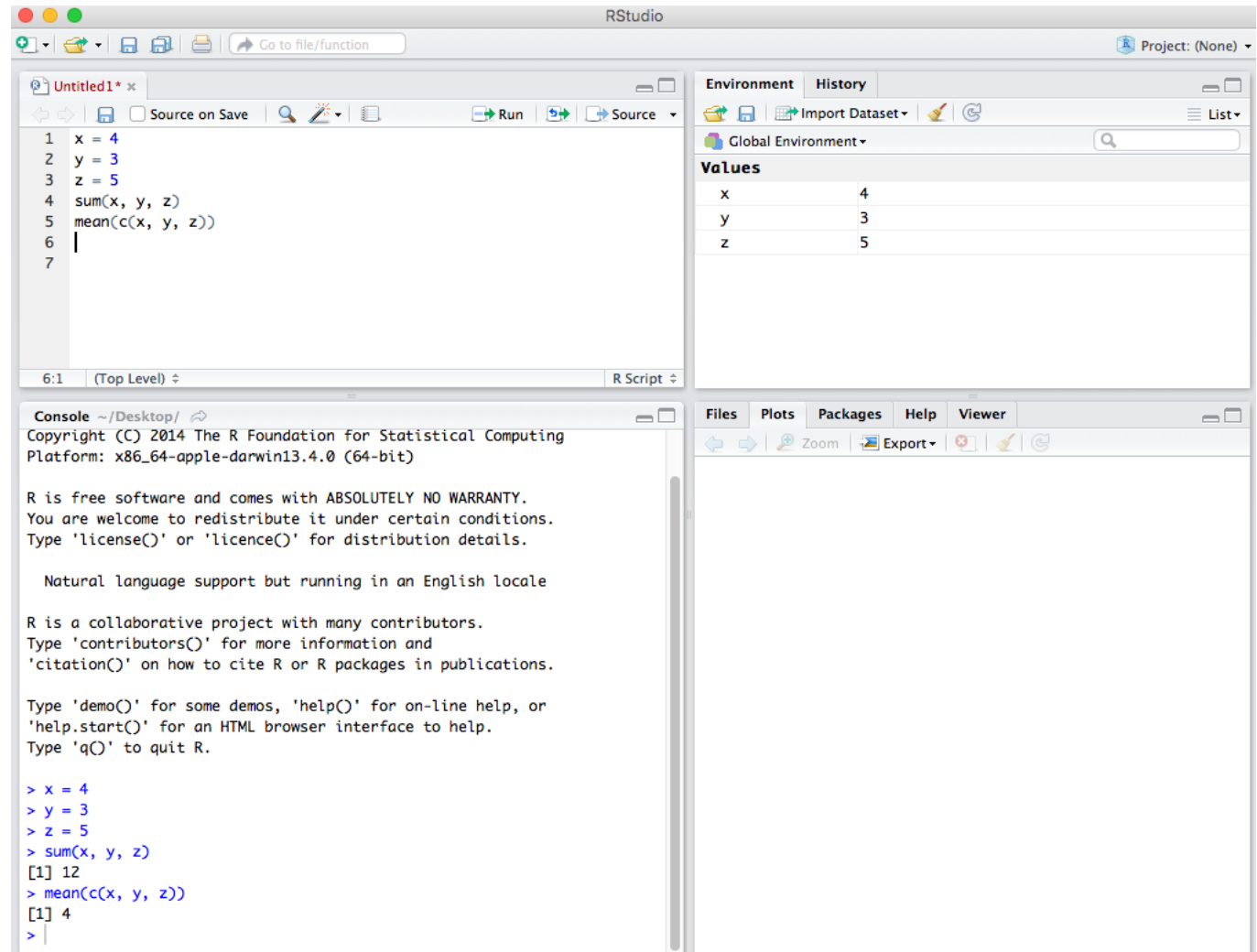
R is an interpreted language. That means that you tell it to run a **function**, and it does. You can run one function at a time, or several in sequence.

Here are 5 functions.

The first three only have one argument. They assign a number to a variable.

The fourth one has multiple arguments. It calculates the sum of the three variables. The fifth takes one complex argument.

Notice that R runs each function. If it is a calculation, it gives you the result.



The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains a script with the following code:

```
1 x = 4
2 y = 3
3 z = 5
4 sum(x, y, z)
5 mean(c(x, y, z))
6 |
7
```
- Environment Pane:** Shows the 'Global Environment' with the following values:

Variable	Value
x	4
y	3
z	5
- Console:** Shows the output of the script execution:

```
> x = 4
> y = 3
> z = 5
> sum(x, y, z)
[1] 12
> mean(c(x, y, z))
[1] 4
> |
```
- Files Pane:** Shows the file structure of the project.

Setting the working directory

R needs a working directory to do its work. The working directory is the directory (or folder) on your computer where it looks for files. This is also where it will save files.

To see the current working directory that R is using, you can type the command **getwd()**. R will print the current working directory in the console window.

To change the working directory, you can use the command **setwd()**. Unlike `getwd()`, `setwd()` needs an argument inside of the parentheses. The argument it needs is the name of the new working directory. I like to use my desktop for small projects, so I type the following **setwd("/Users/jsprouse/Desktop")**. Notice that the directory must be in quotes. Character strings must be in quotes in R (either double or single, it is your choice).

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

```
> getwd()  
[1] "/Users/jsprouse/Desktop"  
> setwd("/Users/jsprouse/Desktop")  
> |
```

Also note that nothing seems to happen when you set the working directory. R just does that in the background, and waits for a new command.

Base functions and add-on packages

The language R comes with a (very large) set of functions built-in. This built-in library of functions is called the **base**.

But R is also a complete (as in Turing-complete) programming language, so you can easily create new functions for yourself. There is literally a function called **function()** that you can use to define a new function of your own.

When people write functions that they think are useful enough to share with the world, they combine them together into something called a **package** (or **library**). Packages often consist of several thematically-related functions that help you run a specific kind of task (or analysis).

These user-created packages are part of the reason that R is so incredibly useful. Nearly every analysis you can think of has been implemented by somebody in an R package. You just need to do some searching to find the right package for the job you want to do. (And if it can be done using base functions, somebody on the internet has posted the code to do it.)

Installing and loading packages

Once you know the name of a package that you want to use, you can install it right from the command line in the console of R.

To install a new package, you can use the command **install.packages()**. Then, you just put the name of the package, in quotes, inside of the parentheses.

For example, if you want to install the tidyr package, you would type **install.packages("tidyr")** and hit enter. R will find the package in an online repository (called CRAN for comprehensive R archive network), and install it on your machine.

R does not load every package that you install when you open R. You have to tell it to load a package (some of them are large, so it would take time and memory to load them all). To do this, you use the function **library()**. You put the name of the package inside the parentheses, this time without quotes.

For example, if you want to use the functions in tidyr, you would run the command **library(tidyr)** and hit enter.

Reading/Writing csv files

To read in data from a CSV file, you use the function `read.csv()`:

```
amtdata = read.csv("raw.data.from.AMT.csv")
```

If you type the name of the data set, `amtdata`, and press enter, R will print it out for you on the screen.

You can also use the functions `head()` and `tail()`. `Head(amtdata)` will show you the first 6 rows of the data set; `tail(amtdata)` will show you the last 6 rows.

To write an existing piece of data in R to a CSV file, you use the aptly named function `write.csv()`, where "x" is the argument specifying the data you want to write, and "file" is the name of the CSV file you want R to create:

```
write.csv(x=amtdata, file="my.first.file.csv", row.names=FALSE)
```

Notice that I've used the optional argument `row.names=FALSE` here to suppress R's default action of putting row names in the first column of the CSV.

Reading about functions inside of R

R comes with a fairly complete help system, and you should use it.

The primary use of the help system is to see all of the arguments that you can pass to a function, along with descriptions of what they do, and examples that demonstrate them.

To see the help page for a function, just type a question mark followed by the function name, and press enter:

```
?write.csv
```

Go ahead and do that now, and take a look at all of the information it provides. It may take a while to get used to reading this information (it is dry, without much hand-holding), but trust me, over time, you will find the help files really useful.

Data types in R

R recognizes several different data types:

- vector:** A one dimensional object, like a sequence of numbers.
- matrix:** A two dimensional object, where all of the items in the matrix are of the same type (all numbers, or all character strings, etc).
- array:** Like a matrix, but can have more than two dimensions.
- data frame:** Two dimensions, and perfectly suited to data analysis. Each column can be of a different type (numbers, strings, etc).
- list:** Just a collection of objects. This is the most general data type. It allows you to collect multiple (possibly unrelated) objects together under one name.

For experimental data analysis, the goal is to put your results into a data frame. Along the way, you may construct vectors, matrices, etc. But the final object will be a data frame that you can use to run analyses and create plots.

Indexing data types

Indexing means identifying a specific element in your data. As you can imagine, the way that you index an element depends on the data type that you have.

vector: A one dimensional object, like a sequence of numbers.

You can create a vector using the `c()` function (it stands for “combine”):

```
x = c(1, 3, 5, 7, 9)
```

And you can index an element in a vector by using bracket notation, and referring to the ordinal number of the element:

```
x[2]    #this will return 3          (the hash mark indicates a comment in R)
```

```
x[5]    #this will return 9
```

You can also change an element in a vector, while leaving everything else the same, by using the bracket notation:

```
x[2] = 17          #this will make x the sequence 1, 17, 5, 7, 9
```

```
x[5] = 23          #this will make x the sequence 1, 17, 5, 7, 23
```


Indexing data types

matrix: A two dimensional object

You can create a matrix using the `matrix()` function:

```
y = matrix(1:16, nrow=4, ncol=4)
```

And you can index an element in a matrix by using bracket notation with two numbers. The first is the row number, the second is the column number.

```
y[2,4]          #this will return the element in row 2 / column 4
```

```
y[2,]           #this will return the entire second row
```

```
y[,4]           #this will return the entire fourth column
```

```
y[1:2,3:4]      #this will return the first two rows of columns three and four
```

Just like with vectors, you can replace elements in a matrix using the bracket notation. I'll leave that to you.

Indexing data types

data frame: A two dimensional object, optimized for data analysis.

You can create a data frame using the `data.frame()` function:

```
names = c("Mary", "John", "Sue")
```

```
ages = c(22, 25, 27)
```

```
colors = c("red", "blue", "green")
```

```
people = data.frame(names, ages, colors)
```

You can index data frames using bracket notation:

```
people[2,3]          #this will return "blue"
```

You can also index data frames by naming the columns using the `$` operator:

```
people$names          #this will return the names column as a vector
```

```
people$names[2]       #this will return "John"
```

Indexing data types

list: A collection of objects

You can create a list using the `list()` function:

```
mylist = list(x, y, people)
```

You can index elements of a list using a double bracket:

```
mylist[[1]]          #this will return the vector x
```

```
mylist[[2]]          #this will return the matrix y
```

```
mylist[[3]]          #this will return the data frame people
```

Once you've indexed a list element, you can use bracket notation to index specific elements inside that element:

```
mylist[[2]][2,4]      #this will return 14
```

Assignment operators

You've already seen one assignment operator in action - the equal sign. An assignment operator allows you to assign an object (like a vector or matrix) to a variable name (like `x`, or `mylist`).

There are three assignment operators in R:

- | | |
|-------------------------------|---|
| <code>x = c(1,2,3)</code> | The equal sign assigns the element on the right to the variable name on the left. |
| <code>x <- c(1,2,3)</code> | The left arrow (made from a less than sign and a dash) assigns to the left. |
| <code>c(1,2,3) -> x</code> | The right arrow (made from a greater than sign and a dash) assigns to the right. |

Logical operators check to see if a given mathematical statement is true. I put them here because they shouldn't be confused with assignment operators:

- | | |
|-----------------------|--|
| <code>5 == 2+3</code> | This is logical equals. It checks to see if the values on either side are equal to each other. Notice it is two equal signs. |
|-----------------------|--|

Logical operators

Logical operators check to see if a given mathematical statement is true. I put them here because they shouldn't be confused with assignment operators:

<code>5 == 2+3</code>	This is logical equals. It checks to see if the values on either side are equal to each other. Notice it is two equal signs.
<code>5 != 4</code>	No equal to.
<code>5 > 2</code>	Greater than
<code>5 < 8</code>	Less than
<code>5 >= 2</code>	Greater than or equal to
<code>5 <= 8</code>	Less than or equal to

You can apply logical operators to any data type, including matrixes, data frames, etc.

<code>y <= 5</code>	#Remember that y is a 4x4 matrix. This will return a 4x4 matrix of TRUEs and FALSEs.
------------------------	--

Learning R

The assignments in this course will help you learn R. I'll give you a list of functions that will help you with the assignment, and then it will be up to you to work with them to complete the assignment.

I am doing it this way because the only way to learn a programming language is to jump in and do it. That said, I realize that you won't be able to do it without help. So here are the ways to get more knowledge:

1. Google your question (the answer will be on StackOverflow)

R has a huge user community. Google your questions. You will likely find an answer, or an answer to a similar question.

2. Read an R tutorial

There are tons of free tutorials out there. I am not going to recommend any specific ones, because they all cover the same stuff for the most part.

3. Read a book

There are tons of free R books out there. Again, just google for them.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1: Design

Section 2: Analysis

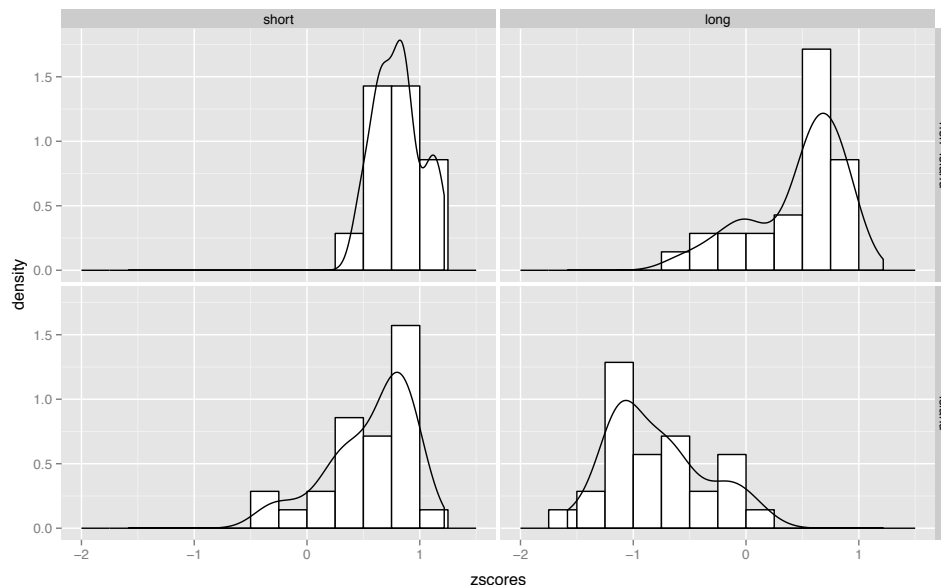
Section 3: Application

Before anything else — Look at your data!

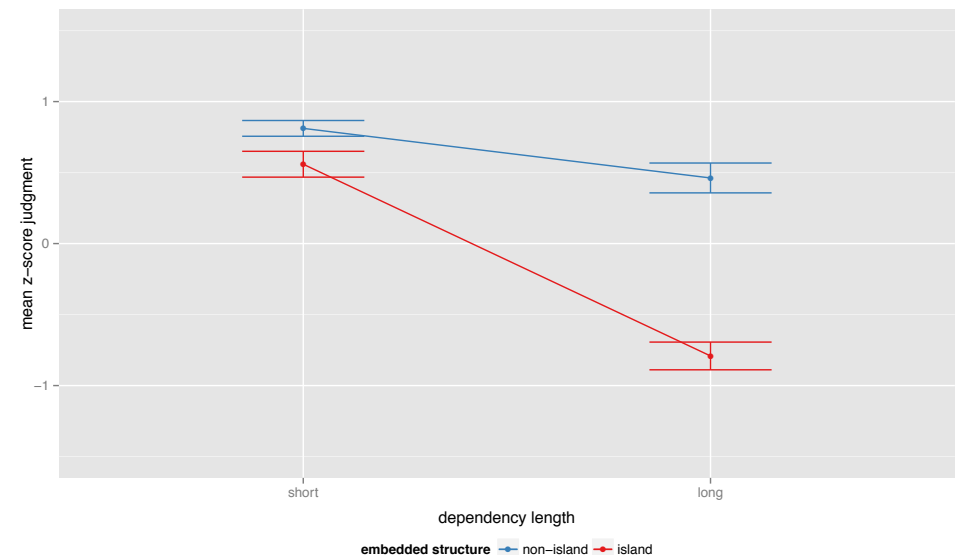
I cannot stress this enough. **You have to look at your data.** You can't just plop it into a statistical test and report that result. Well, you can, but you may miss something important. (And, to be fair, I am guilty of not looking at my data enough, so I say this with real experience behind it — look at your data!)

There are lot of different ways to “look at” your data, and there is no prescribed way that will work for all experiments. But there are two graphs that are going to be important for nearly all experiments: (i) the distribution of responses per condition, and (ii) the means and standard errors per condition.

distribution by condition



means and se by condition



Plotting in R: base vs ggplot2

One of the major benefits of R is the ability to make publication quality figures easily (and in the same environment as your statistical analysis).

R's base comes with all of the functions that you might need to create beautiful figures. The primary function is `plot()`, with a long list of additional functions that will add tick marks, add labels, format the plotting area, draw shapes, etc.

If you spend the time to become proficient at plotting with base functions, you will find that you end up drawing your figures in `layers`: you draw the plot area, you add points, you add lines, you add error bars, you add a legend, etc.

There is a package, written by Hadley Wickham (also the creator of `dplyr` and `tidyr`), called `ggplot2` that takes this fact to its logical conclusion. The two `g`'s in the name stand for "grammar of graphics". The idea is that the functions in `ggplot` allow you to construct a beautiful figure layer by layer, without having to spend as much effort as you would with the base R functions.

The received wisdom is that `base R functions` give you the `most flexibility`, but require `the most effort` to create a good looking figures, while `ggplot` requires `the least effort` to create good looking figures, but you `lose some flexibility` (or rather, deviating substantially from the default look in `ggplot` will lead to complex code, just like base R).

Why do we look at distributions?

A **distribution** is simply a description of the number of times that an event (in this case, a judgment or rating) occurs relative to the other possible events.

For each sentence type in our experiment, we assume that it has a single underlying acceptability value. However, there are other factors affecting its judgment — the lexical items and meaning of the specific item, the noise of the judgment process itself, any biases that the subject has, etc. So, in practice, we expect that there will be a distribution of judgments for a sentence type.

The first thing we want to do is look at that distribution for each of our experimental conditions. In theory, we expect the distribution of judgments to be relatively **normal** (or gaussian, or bell-shaped). The reason for this is that we expect the other factors that are influencing the judgments to be relatively random. When you mix a bunch of random factors together on top of a non-random factor (the sentence type), you get a normal (gaussian, bell-shaped) distribution.

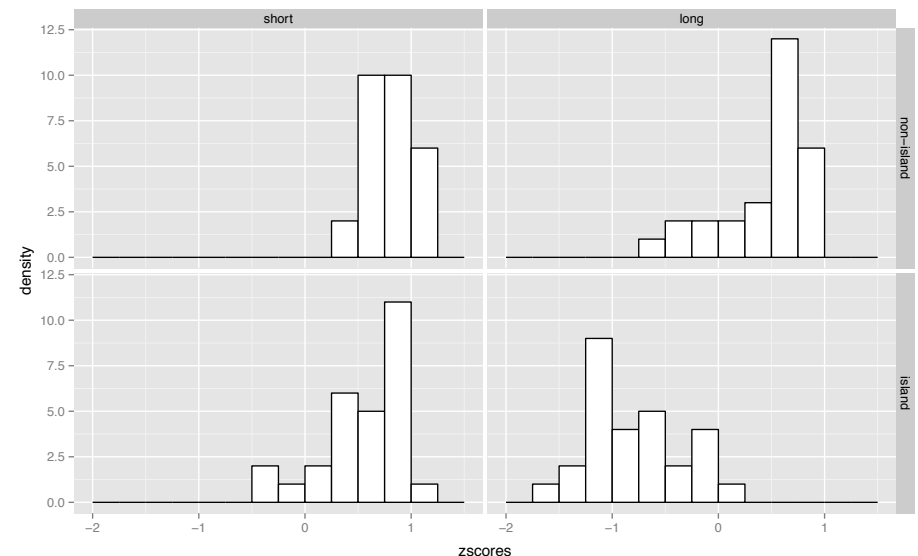
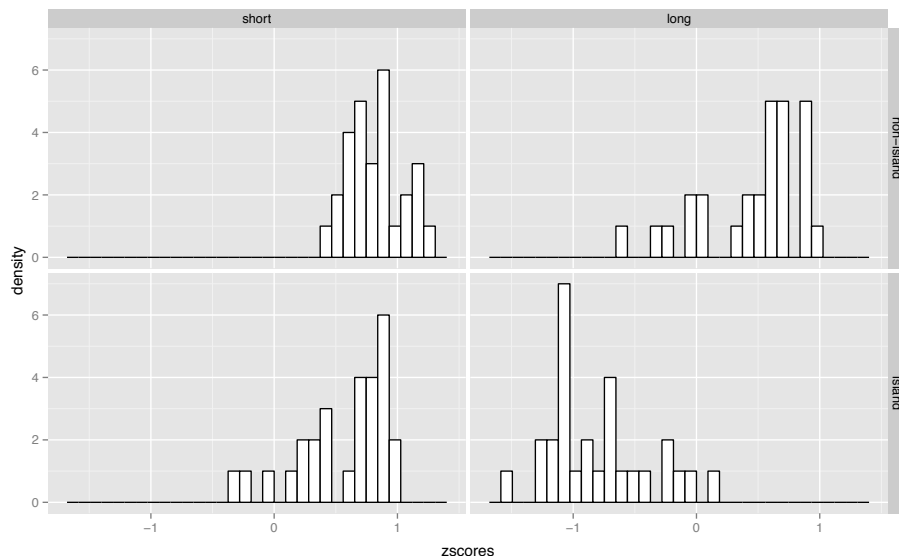
So what we want to do is look at the distribution of each of our experimental items to make sure that they are roughly normally distributed. If they aren't roughly normal, then something might be wrong in our experiment (an outlier or two, a non-random bias for some number of participants, a non-random factor that we failed to control for, etc.)

Histograms

A histogram shows the **counts** of each response type. The benefit of a histogram is that the y-axis, counts, is very intuitive, and shows you what the raw data looks like.

One drawback of a histogram is that the shape of the distribution in a histogram is strongly dependent on the size of the **bins** that you choose (with continuous data, like z-scores, you have to define bins). If the bins are too small, a normal distribution will look non-normal, and if the bins are too big, a non-normal distribution can look normal.

You can use the code in [distribution.plots.r](#) to generate histograms with different bin-widths and see the effect:

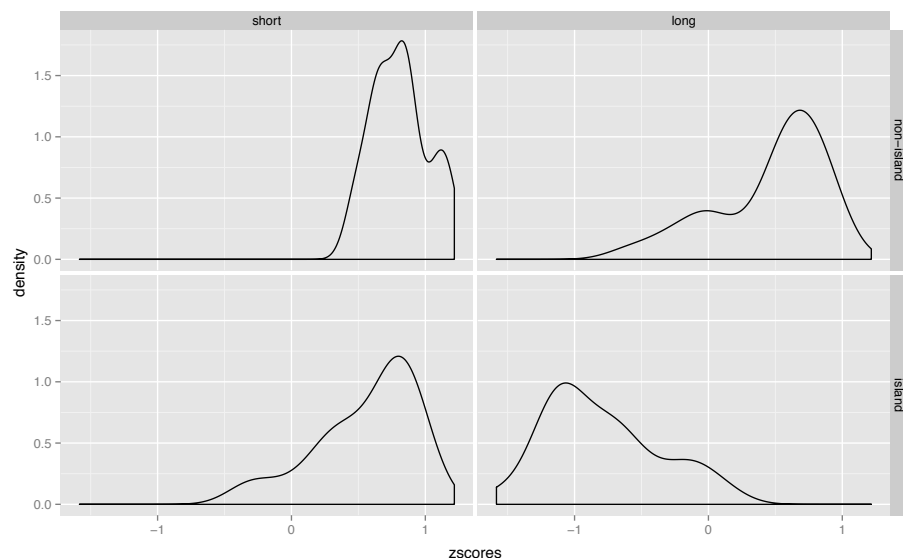


Density plots

A **density plot** shows you the **probability density function** for your distribution. The “curve” that people think of when they think about distributions is a probability density function. The idea behind a probability density function is that it shows the relative likelihood that a certain judgment will occur.

Speaking more precisely, the **total area under the curve of a pdf will be 1**, and the area under the curve between two points will be the probability that a judgment will be between those two values.

Much like binning, pdfs are necessary because there are an infinite number of possible values on a continuous scale (like z-scores), so the probability of any given judgment is infinitesimal. That isn’t helpful. So we use the pdf to calculate the probability that a judgment is between two possible values.



Like histograms and binning, pdfs will vary based on the kernel density estimation method that you use to calculate them. R tries its best to do this in a reasonable way.

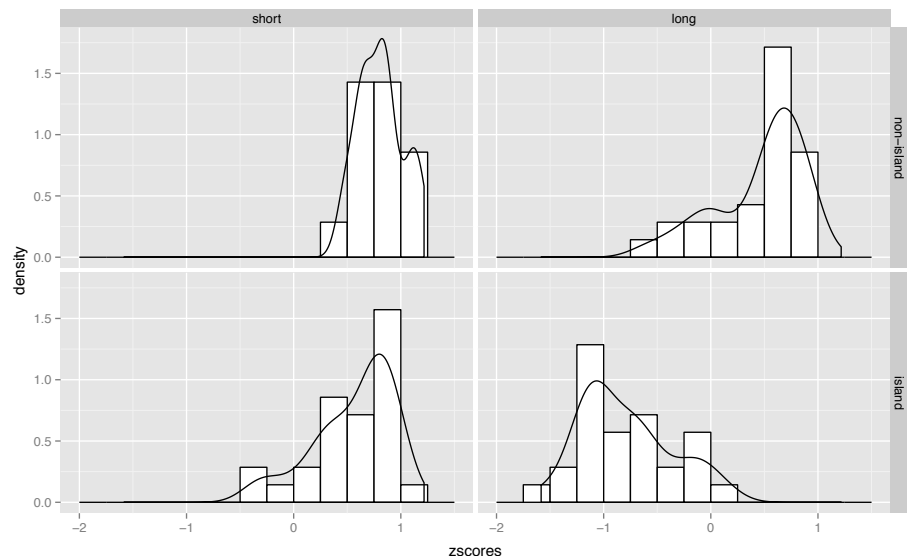
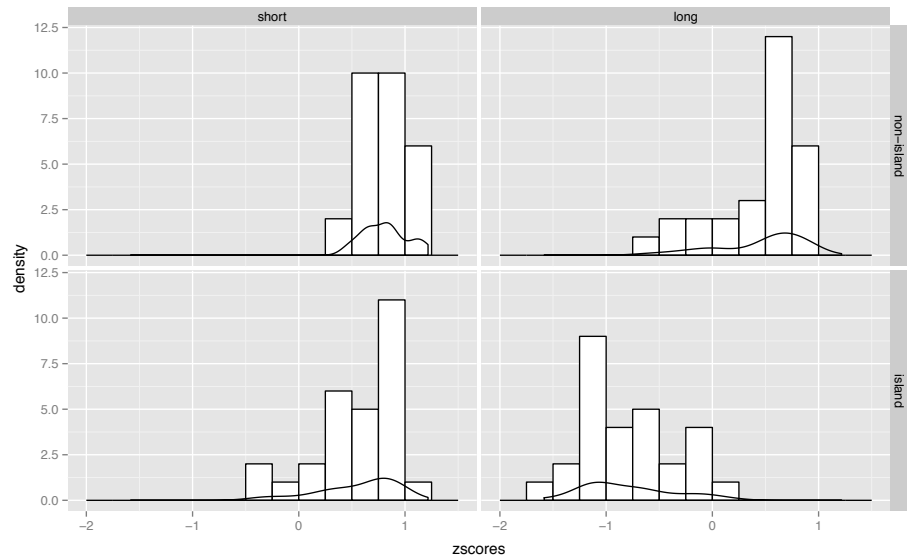
You can use the code in the script to generate density plots using R’s default kernel density estimation.

Combining histograms and density plots

You can combine histograms and density plots into one figure if you want. The code in [distribution.plots.r](#) shows you how to do this.

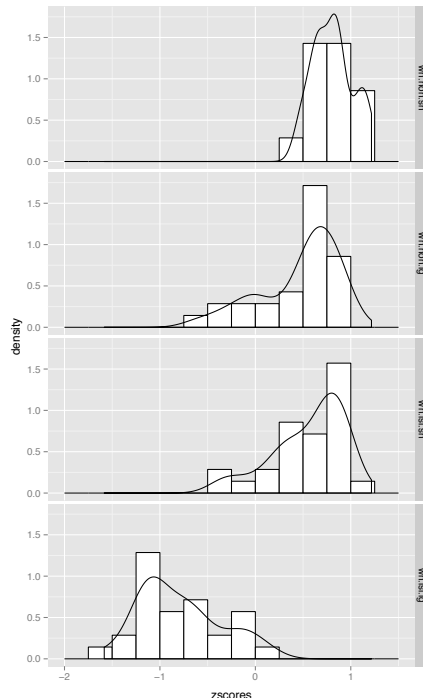
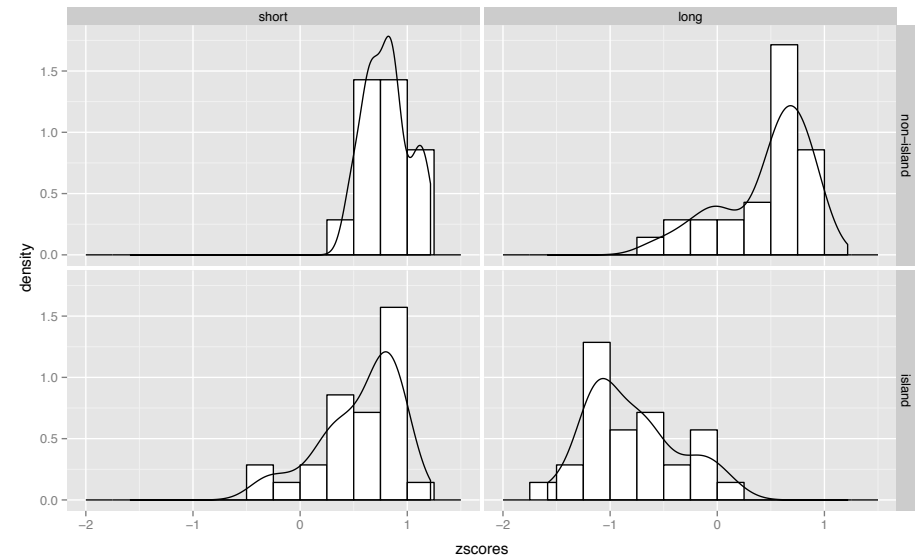
One thing to note is that frequencies and density are typically on different scales. Frequency is typically much larger than density. So if you plot the two together, the density curve will be flattened.

So what we probably want to do is use density alone for the y-axis, and scale the histogram to fit. R does this very easily (see the code). The result makes the histogram harder to interpret, but allows you to compare the raw responses to the estimated density function nicely.



Arranging the plots in different ways

You may have noticed that the distribution plots have been arranged according to the two factors and their levels. This is called **faceting**, and is a very convenient way to organize multiple plots.



You can organize faceting based on any factor you want. You can also do it based on one factor alone (creating a single column or a single row).

The trick is to choose an arrangement that helps readers understand the data. For example, if you aligned the four conditions in a column, you can highlight the different locations of the distributions on the x-axis. This makes it clear that the fourth condition tends to have lower acceptability than the other three.

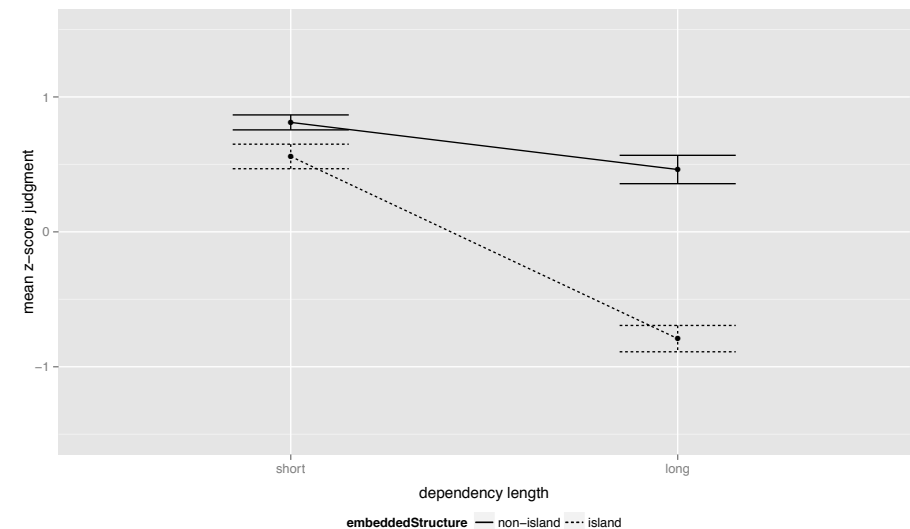
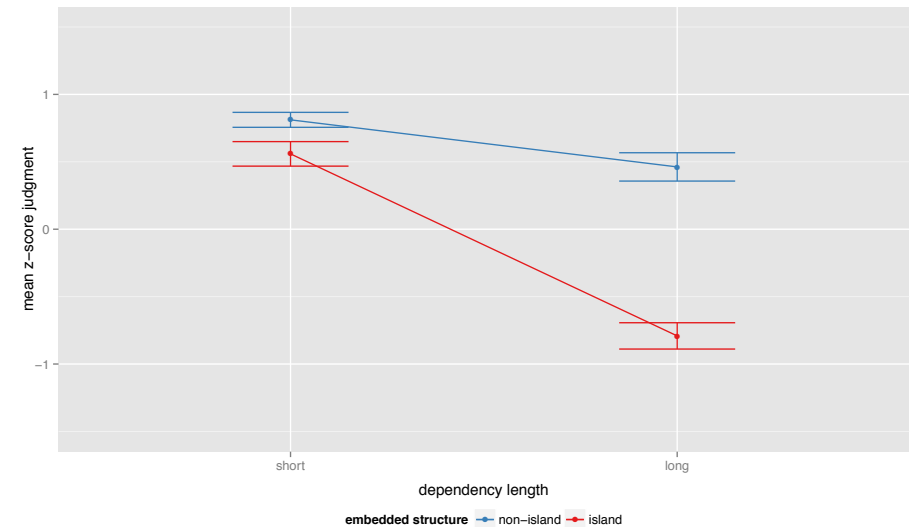
Plotting means and standard errors

The second major plot type that you will (pretty much always) want to create is a plot of the condition means and their (estimated) standard errors.

For any design that has more than one factor (two factors, three factors, etc), you will probably want to create something called an **interaction plot**. An interaction plot is a line-plot arranged by the levels of the factors.

In a 2-D plot, you can only directly specify one axis. The other is the value of the responses. Typically, you specify the x-axis, and let the y-axis be the value of the responses.

So, if we specify the x-axis to be the two levels of the DEPENDENCY LENGTH factor, we then need to use something else to specify the levels of EMBEDDED STRUCTURE. We can either use **color** or the type of line.



Plotting means and standard errors

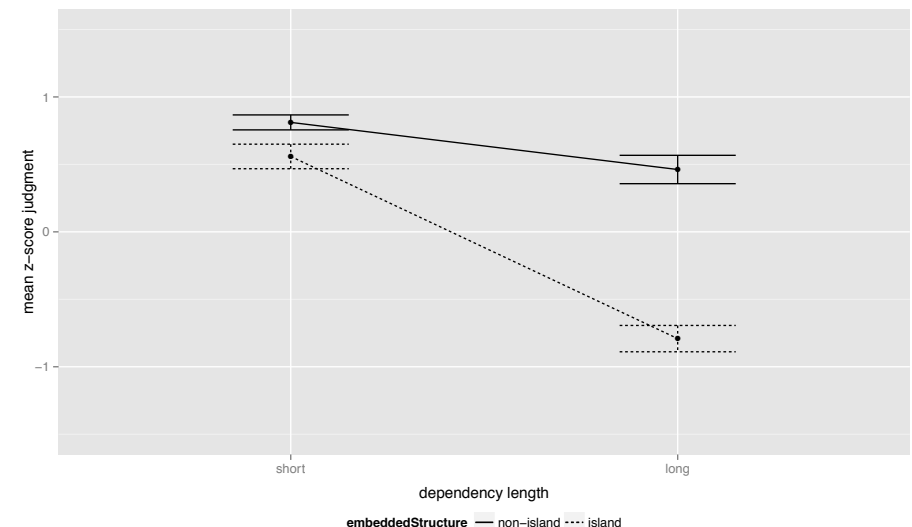
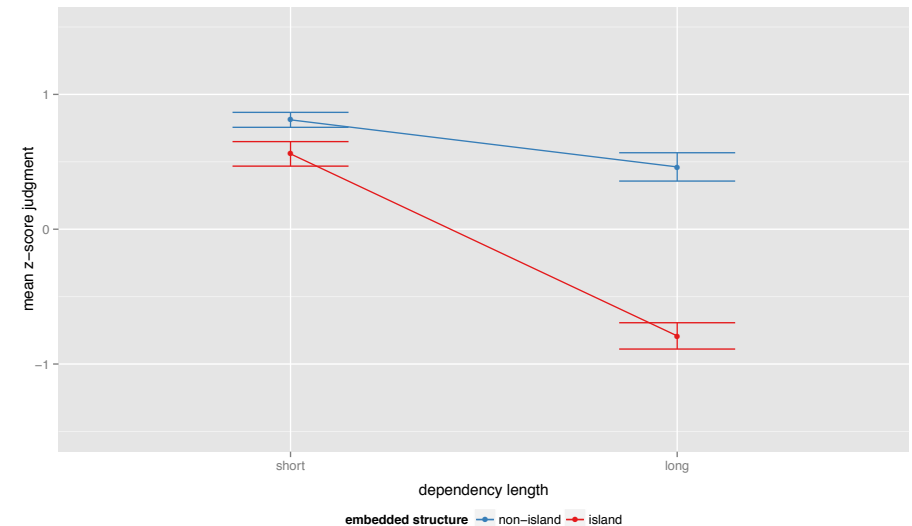
If you look at the code in `interaction.plot.r`, you can see that we use the `summarize()` function from `dplyr` to calculate three numbers.

We calculate the **mean** of each condition. We plot these means as the points in the interaction plot.

We calculate the **standard deviation** of each condition. We **do not** plot the standard deviation. We just use it to calculate the standard error.

We calculate the **estimated standard error** of each mean. The formula for this is **standard deviation** divided by the square root of the number of participants. The **error bars** in the plot are 1 standard error above and 1 standard error below the mean.

As a rule of thumb, non-overlapping error bars tend to be statistically significant in a null hypothesis test.



Digression: Basic Statistics Concepts

In order to understand why we plotted means and standard errors, we need to understand a little bit about statistics — what sorts of information we are looking for, and how it is calculated.

Concepts we need to know

1. Population vs sample
2. Parameter vs statistic
3. Central Tendency
4. Variability (or spread)
5. Parameter estimation (from a statistic)
6. Testing hypotheses about populations (sampling distribution of the mean, and standard error of the mean)

With these concepts, everything about the plots makes sense. If you already know these concepts, you can skip to the next section. If you don't, you (or we) should work through these.

I have created an R script called [parameters.statistics.r](#) that helps to illustrate some of these concepts using simulations.

Population versus Sample

Population: The **complete set** of items/values. This is most commonly thought of as people (e.g., all of the people in the US is the population of the US), but it can also be other units such as judgments (the complete set of acceptability judgments would be the population of judgments). A population can be defined using whatever criteria you want (e.g., the population of people born in NJ; or the population of judgments given to a certain sentence).

Sample: A **subset** of a population. The process of selecting the subset from a population is called sampling. Sampling is usually necessary because most populations of interest are too large to measure in their entirety. Samples can be chosen randomly, or they can be chosen non-randomly. How a sample is chosen matters for the types of inferences you can make. (Random is best... everything else limits your inferences.)

Parameter versus Statistic

Because both populations and samples can be characterized as distributions, you can calculate things like means, medians, variances, and standard deviations for both of them.

Parameter: A number that describes an aspect of a population. Usually written with a greek letter.

Statistic: A number that describes an aspect of a sample. Usually written with a Roman (English) letter.

And now you can see where the word “statistics” comes from. Statistics are the numbers we use to characterize samples... and since experiments are conducted on samples (not populations), we are usually manipulating statistics, not parameters. There are different types of statistics:

Descriptive Statistic: A statistic that describes an aspect of a sample.

Estimator: A statistic that can be used to estimate a population parameter.

Test Statistic: A statistic that can be used to make inferences.

Describing Distributions (population or sample)

It is great to look at distributions, and it is great to use the probability density function to predict probability. But sometimes, we want single numbers that can describe some aspect of the distribution.

There are different types of information that one could be interested in. Two types that arise frequently are:

Location, or Central Tendency: A measure of location/central tendency gives a single value that is representative of the distribution as a whole (its expected value). The three most common measures of this are the **mean**, **median**, and **mode**.

Variability, or Dispersion/Spread: A measure of variability/dispersion/spread gives a single value that indicates how different the values in a distribution are from each other. The most common measures are **variance** and **standard deviation**, although you may also encounter the **absolute deviation**.

We will see these over and over again, but for now, I will simply define them so that we are all on the same page mathematically when they come up later.

Central Tendency: Mean

Let's start with the (arithmetic) **mean**, which is commonly called the average.

Mean: The sum of the values, divided by the number of values (the count) that were summed.

$$\text{Mean} = \frac{x_1 + x_2 + \dots x_n}{n}$$

The mean is by far the most common measure of central tendency, so you will encounter (and use it often). The primary benefit of the mean is that it takes the “weight” of the values into consideration. But this is also a drawback, as it means that it is distorted by very large (or very small) values.

$$\text{mean}(1, 2, 3, 4, 5) = 3$$

$$\text{mean}(1, 2, 3, 4, 10) = 4$$

$$\text{mean}(1, 2, 3, 4, 100) = 22$$

Central Tendency: Median

The next most common measure of central tendency is the **median**.

Median: The median is the value in a set of values that divides the set into two halves (an upper half and a lower half). If there is an odd number of values in the set, the median will be one of the values in the set. If there is an even number, the median will be the mean of the two middle values.

The median is interesting for a number of reasons, but perhaps the most valuable aspect of the median is that it is **robust to outliers**. This is just a fancy way of saying that the median is not influenced by very large (or very small) numbers. This is in stark contrast to the mean, which is not.

$$\text{mean}(1, 2, 3, 4, 5) = 3$$

$$\text{median}(1, 2, 3, 4, 5) = 3$$

$$\text{mean}(1, 2, 3, 4, 10) = 4$$

$$\text{median}(1, 2, 3, 4, 10) = 3$$

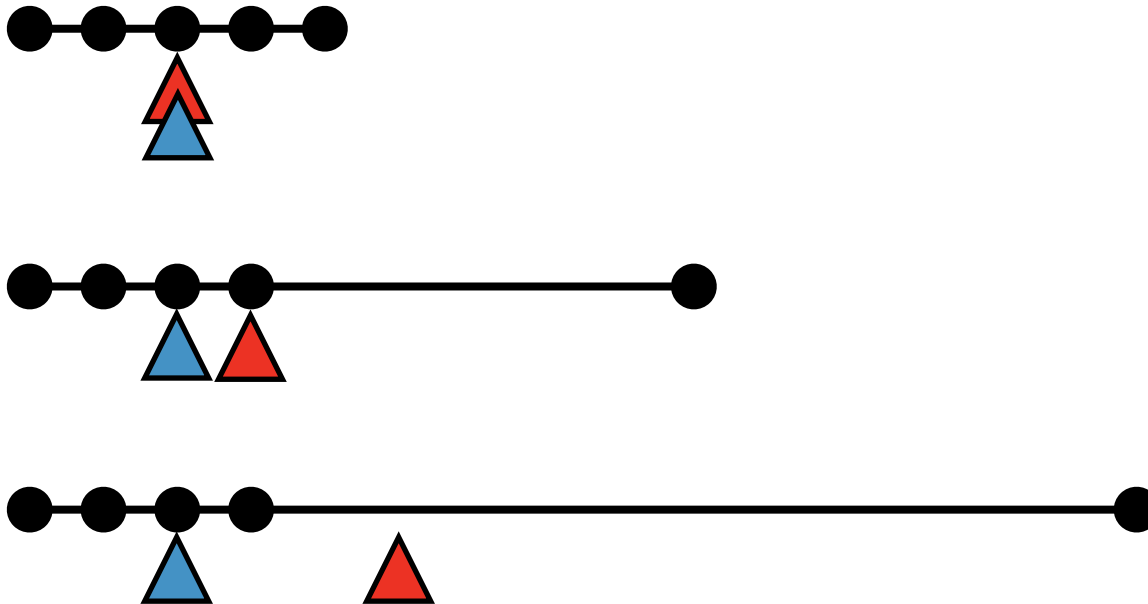
$$\text{mean}(1, 2, 3, 4, 100) = 22$$

$$\text{median}(1, 2, 3, 4, 100) = 3$$

The Mean/Median see-saw analogy

I am not kidding when I say that there is a nifty visual analogy for means and medians involving a seesaw.

If you imagine that the values in your set indicate the location on a seesaw where people (of identical weight) are sitting, then the **mean** is the point where you would place the fulcrum in order to **balance** the seesaw. The **median** is the point where you would place the fulcrum in order to put **half** of the people on each side of the seesaw.

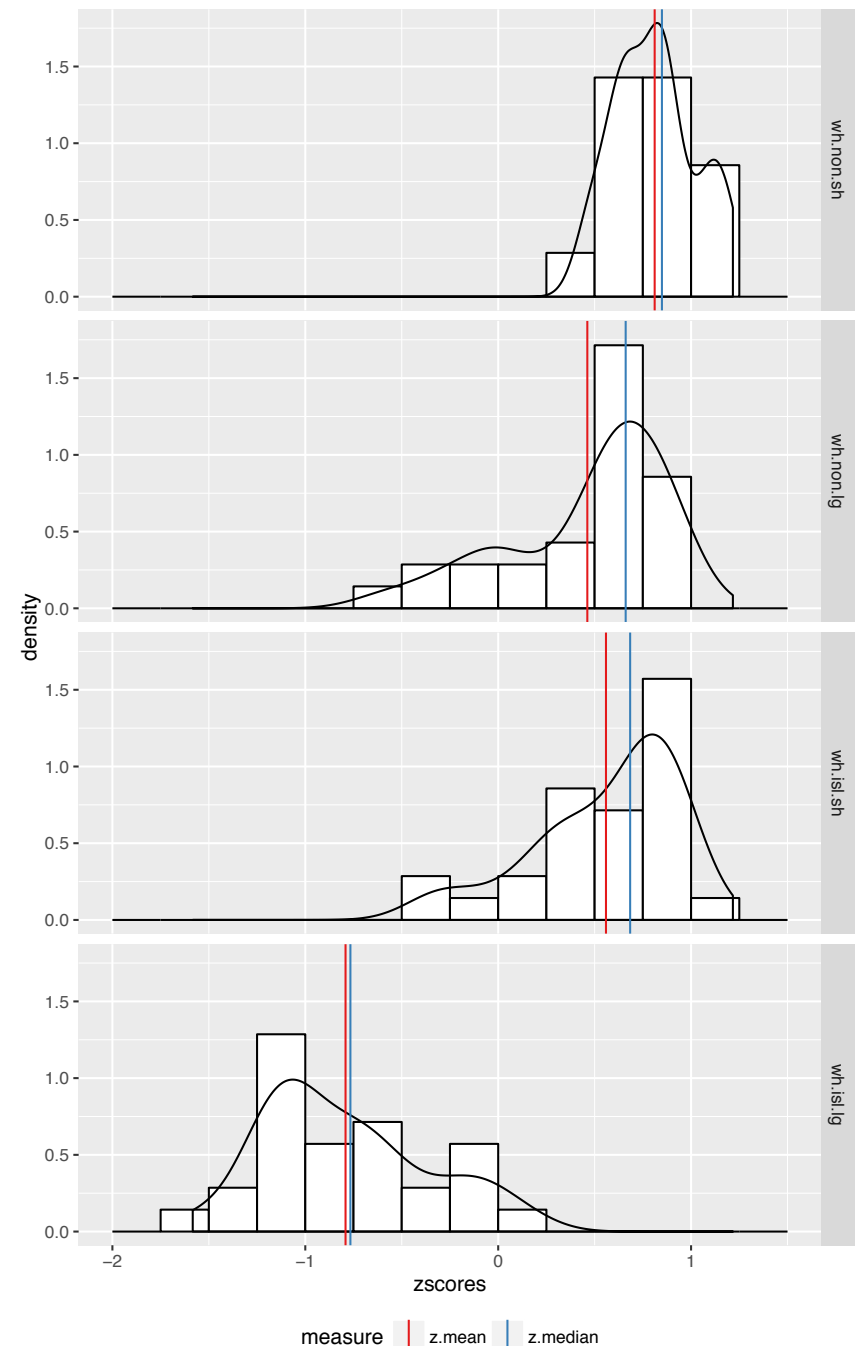


Mean and Median in real distributions

To see the difference between means and medians, we can calculate the means and medians of each of our experimental conditions, and then overlay a vertical line for the **mean** and **median**.

The **script mean.median.lines.r** shows you how to do this.

As you can see, the mean tends to be pulled to the side by long tails. In a perfectly symmetric distribution (like the normal distribution), the mean and median will be identical.



Variability

Let's build up this idea in a several steps.

Step 1: The variability of a data point must be measured against a reference point. This will probably be the central tendency (or expected value) of the set of values (the distribution). Most likely this will be the mean. So the variability of a data point is simply its difference from the mean:

$$\text{variability of } x = (x - \text{mean})$$

Now, you may think that the variability of a set of values (a distribution) can be derived by adding up the variability of all of the values in it. Let's try this and see what happens. Let's say your set is (1, 2, 3, 4, 10). The mean is 4.

variability of 1 = $1 - 4 = -3$

variability of 2 = $2 - 4 = -2$

variability of 3 = $3 - 4 = -1$

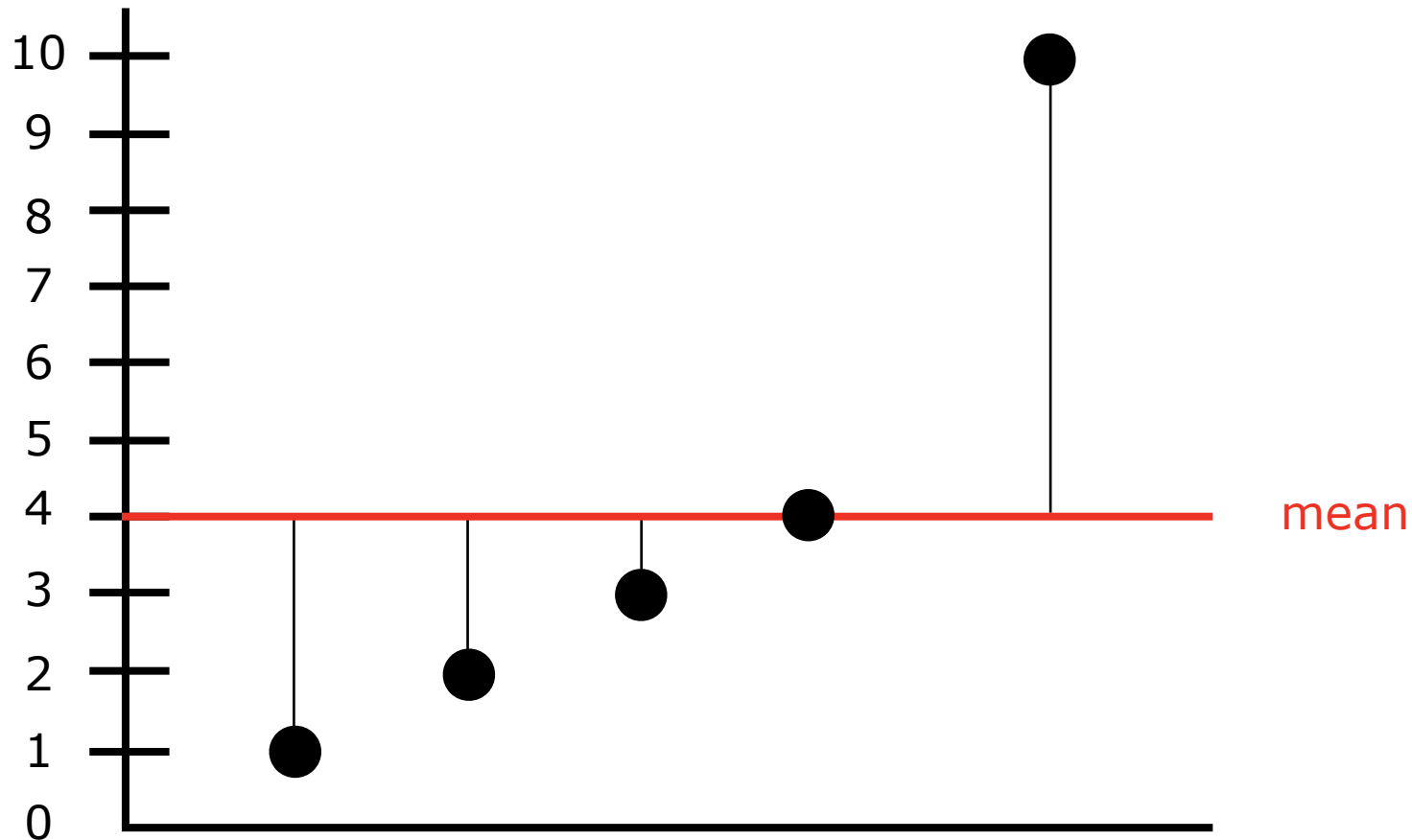
variability of 4 = $4 - 4 = 0$

variability of 10 = $10 - 4 = 6$

sum of the variability:

$$-3 + -2 + -1 + 0 + 6 = 0$$

Variability



Because of the definition of the mean, the deviation (from the mean) of the points below the mean will always equal the deviation of the points above the mean. So it is impossible to simply sum this deviation.

Variability

OK, so now we know that we can't just sum $(x - \text{mean})$, because that will always yield a sum of 0. What we need is a measure that of variability for each data point that is **always positive**. That way, when we add them up, the total will be positive.

The most common solution to this problem (although not necessarily the most intuitive) is to square the difference:

$$\text{variability of } x = (x - \text{mean})^2$$

Since squares are always positive, this will avoid the summation problem that we saw before:

variability of 1 = $(1 - 4)^2 = 9$

variability of 2 = $(2 - 4)^2 = 4$

variability of 3 = $(3 - 4)^2 = 1$

variability of 4 = $(4 - 4)^2 = 0$

variability of 10 = $(10 - 4)^2 = 36$

sum of the variability:

$$9 + 4 + 1 + 0 + 36 = 50$$

Variance

We can call this the **sum of squares**:

$$\text{sum of squares} = (x_1 - \text{mean})^2 + (x_2 - \text{mean})^2 + \dots + (x_n - \text{mean})^2$$

Now, we could try to use the sum of squares as our measure of variability. But one problem with the **sum of squares is that its size is dependent upon the number of values** in the set. Larger sets could have larger sum of squares simply because they have more values, even though there might really be less variation.

One solution for this is to divide the sum of squares by the number of values. This is similar to the mean — it is like an **average measure of variability** for each point. We call it the **variance**:

$$\text{variance} = \frac{(x_1 - \text{mean})^2 + (x_2 - \text{mean})^2 + \dots + (x_n - \text{mean})^2}{n}$$

Standard Deviation

Although variance is a useful measure, it does have one problem. It is in really strange units - the units of measure squared!

$$\text{variance} = \frac{(x_1 - \text{mean})^2 + (x_2 - \text{mean})^2 + \dots + (x_n - \text{mean})^2}{n}$$

 acceptability judgments squared?

The fix for this should be obvious. We can simply take the square root of the variance to change it **back into un-squared units**. We call this the **standard deviation**:

$$\text{standard deviation} = \sqrt{\frac{(x_1 - \text{mean})^2 + (x_2 - \text{mean})^2 + \dots + (x_n - \text{mean})^2}{n}}$$

 same units as the original values

Absolute Deviation

The first time you see standard deviation you might find yourself wondering why we square the deviations from the mean to eliminate the negative signs. Couldn't we just take the absolute value? The answer is yes. It is called the **absolute deviation**:

$$\text{absolute deviation} = \frac{|x_1 - CT| + |x_2 - CT| + \dots + |x_n - CT|}{n}$$

OK, so how do we choose between the **standard deviation** and the **absolute deviation**? In practice, **standard deviations** tend to accompany **means**, and **absolute deviations** tend to accompany **medians**. Here's why:

The **mean** is the measure of central tendency that **minimizes variance (and standard deviation)**. The variance of the mean will always be smaller than (or equal to) the variance of the median.

The **median** is the measure of central tendency that **minimizes the absolute deviation**. The absolute deviation of the median will always be smaller than (or equal to) the absolute deviation of the mean.

Estimating a parameter from a statistic

Let's say you are trying to estimate the variance in a population. But you don't know the mean of the population. What do you do? **You estimate the parameter from your sample.**

This is simple enough. You can use the mean of your sample as an estimate of the mean of the population. I've been loose with notation up to now. Let's do it right. We use greek letter for parameters, and roman letters for statistics:

$$\text{true } \sigma^2 = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + \dots + (x_n - \mu)^2}{n} \quad \mu = \text{population mean}$$

$$\text{estimated } \sigma^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n} \quad \bar{x} = \text{sample mean}$$

Some statistics are better at estimating parameters than others. It turns out that estimating the variance using the sample mean will **underestimate the population variance**. When an estimate systematically under- or over-estimates a parameter, we call it a **biased estimator**.

For a really nice analytic explanation of why this will always underestimate the population variance, see the wikipedia page for Bessel's correction: https://en.wikipedia.org/wiki/Bessel%27s_correction. For a simulation that demonstrates this empirically, see the script [parameters.statistics.r](#).

The right way: Bessel's correction and df

The right way to estimate the population variance using the sample mean is to apply **Bessel's correction** to the equation:

$$\sigma^2 = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + \dots + (x_n - \mu)^2}{n} \quad \mu = \text{population mean}$$
$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n-1} \quad \bar{x} = \text{sample mean}$$


The reason this works is contained in the proof of Bessel's correction, which is far beyond this class. However, the intuition behind it is simple.

When you calculate a statistic, a certain number of values have the freedom to vary. We call this number the **degrees of freedom**.


When you calculate the first statistic from a sample, all of the values are free. You have n degrees of freedom. But when you've calculated one statistic, and are calculating the second one, you only have $n-1$ degrees of freedom.

Think about it. If you know a sample has 5 values, and a mean of 7. How many of the values are free? Just 4. Once you set those 4, the 5th is constrained to be whatever makes the mean equal 7. In the equation above, we already know the mean, so we only have $n-1$ degrees of freedom.

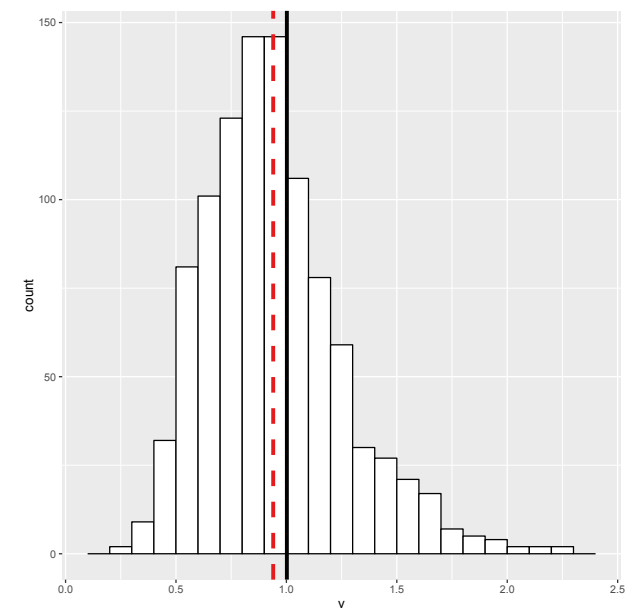
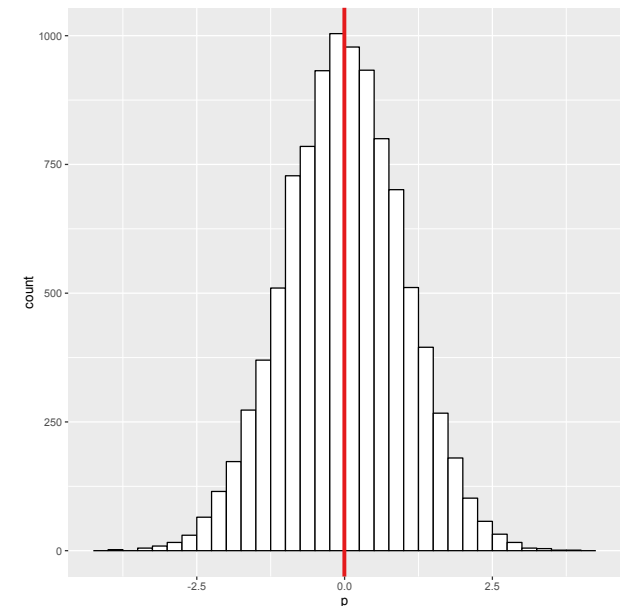
We can see the bias using a simulation

population  x 10,000

In the script `parameters.statistics.r`, I used R to generate a population of 10,000 values with a mean of 0 ($\mu=1$) and a variance of 1 ($\sigma^2=1$). The **mean** is in **red**.

sample  x 20 = \bar{x}_1 and σ^2

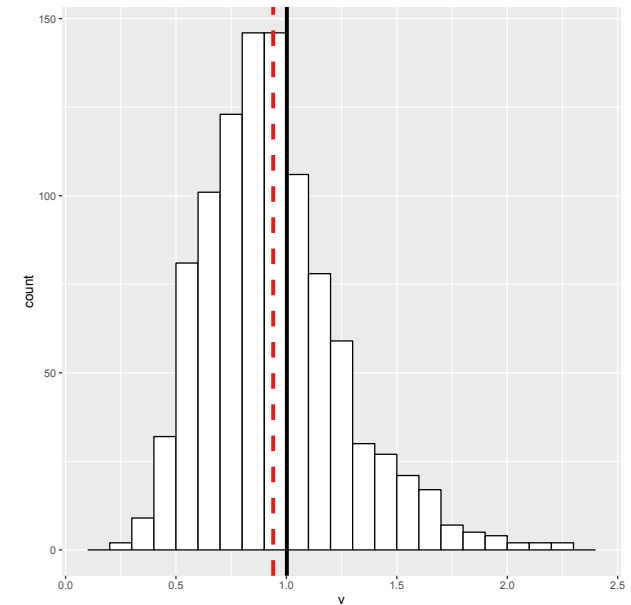
I then took 1,000 samples from the population, each with 20 values. I calculated the variance using the mean for each one. That gives us 1000 variance estimates. We can plot the distribution of variance estimates, with the **mean of the estimates** as a **dashed red line**. We can compare that to the **actual variance**, which is a **black solid line**. As you can see, the mean estimates is low!



And we can see the effect of the correction!

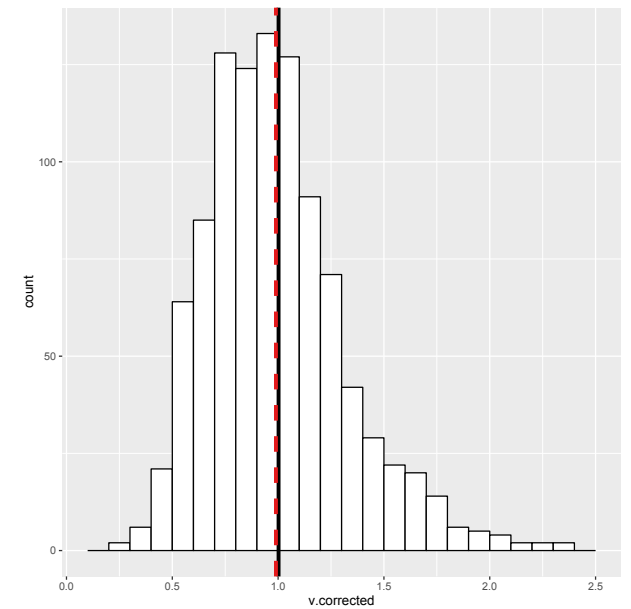
This simulation was without correction

This is just a repeat of the graph from the last slide. These are the uncorrected variance estimates. The mean of these estimates is lower than the population variance. This is the bias we talked about.



Now let's simply use Bessel's correction

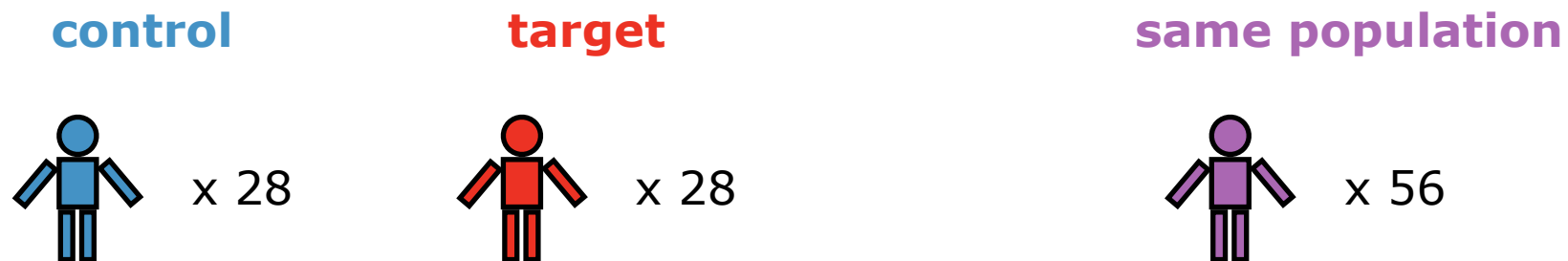
This plot uses the same 1000 samples from the population. The only difference is that the variance is calculated using $(n-1)$ rather than n . Again, the **mean of the estimates** as a **dashed red line** and the **actual variance** is a **solid black line**. They now partially overlap. In general, this will be a much closer estimate, with no systematic bias. It will approach the true value as the number of samples increases.



Why all of this talk of populations, parameters, samples, and statistics?

For simplicity, let's imagine that we only have two conditions in our experiment. And let's imagine that we test our conditions on two different sets of 28 people (that's a between-participant design).

We want to know if the two conditions are different (or have different effects on our participants). One way of phrasing this question is that we want to know if our two samples come from different populations, or whether they come from the same population:


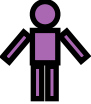
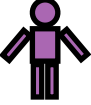
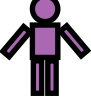


So here is one mathematical thing we can do to try to answer this question. We can calculate the mean for each sample, and treat them as estimates of a population mean. Then we can look at those estimates and ask whether we think they are two estimates of one population mean, or whether they are two distinct estimates of two distinct population means.

Standard Error: How much do samples vary?

How can we tell if two sample means are from the same population or not? Well, one logic is as follows: First, we expect sample means to vary even though they are from the same population. Each sample that we draw from a population will be different, so their means will be different. The question is how much will they vary?

We could, in principle, figure this out by collecting every possible sample from a population. If we calculated a mean for each one, those sample means would form a distribution. We could then calculate the variance and standard error of that distributions. That would tell us how much sample means vary when they come from the same population!

population		x 10,000	
sample 1		x 20	= \bar{x}_1
sample 2		x 20	= \bar{x}_2
sample 3		x 20	= \bar{x}_3


We call this the sampling distribution of the mean.

Its mean is the mean of the population that the samples come from.

Its standard deviation is called the **standard error of the mean**.

... to 10,000 choose 20 ...

Plotting the sampling distribution of the mean

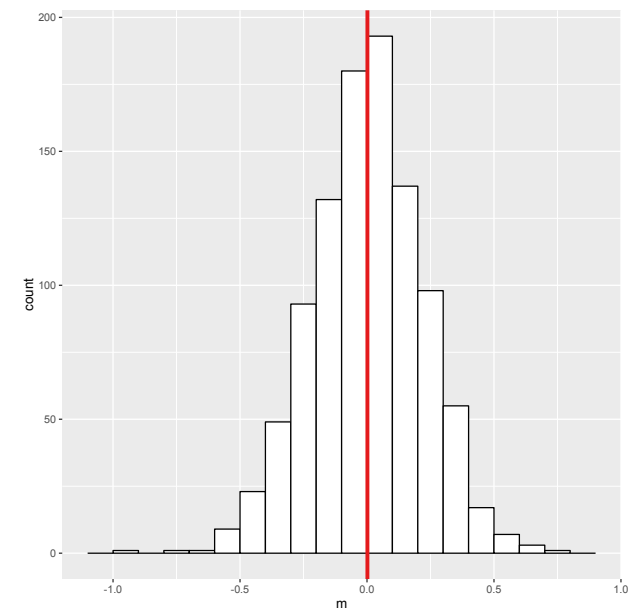
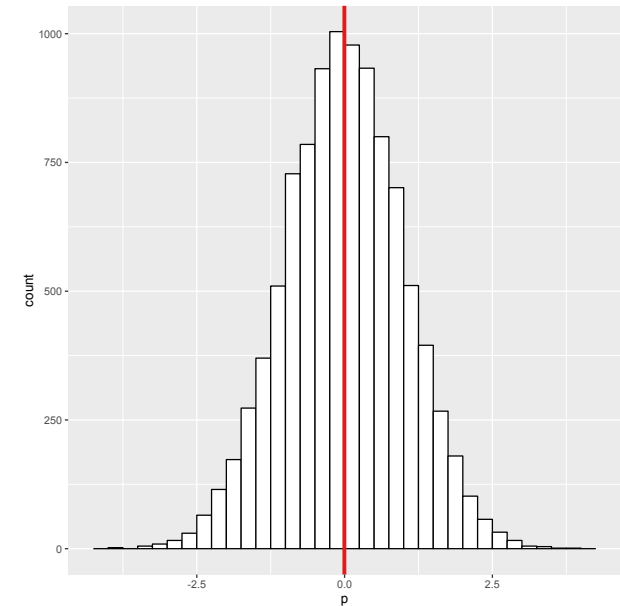
population  x 10,000

In the script [parameters.statistics.r](#), I used R to generate a population of 10,000 values with a mean of 0 and a standard deviation of 1. We've already seen this.

sample  x 20 = \bar{x}_1

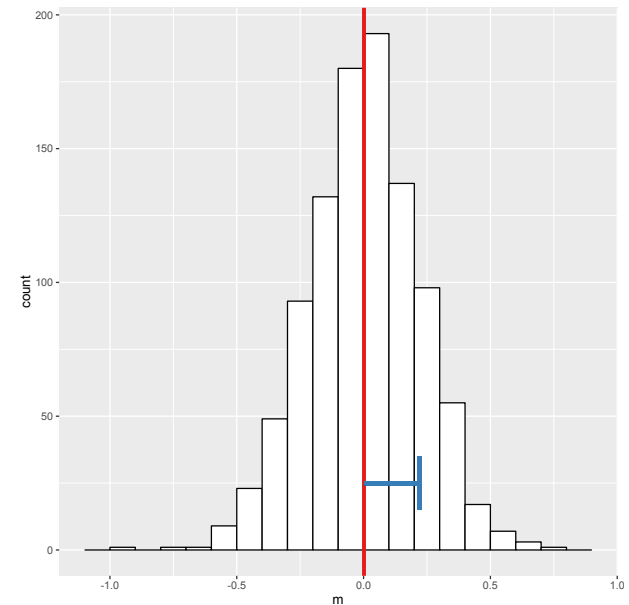
I then took 1,000 samples from the population, each with 20 values. I calculated the mean for each one, and plotted that distribution. This is a simulation of the sampling distribution of the mean.

The mean of the sampling distribution of the means is the population mean!

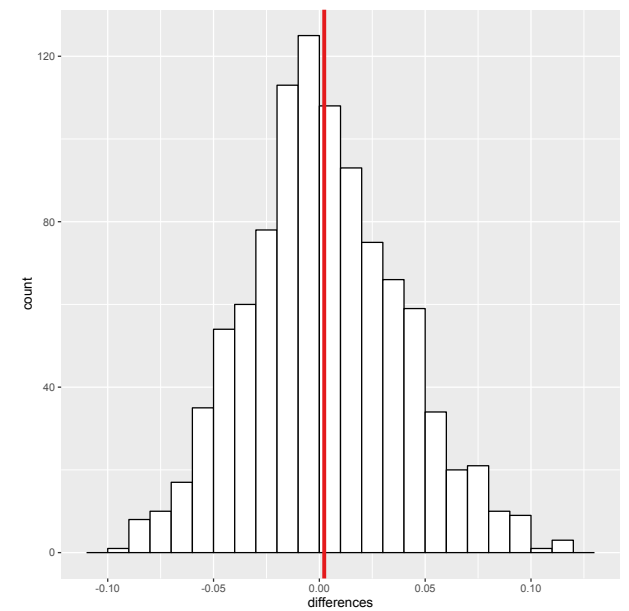


Estimating the standard error

The standard deviation of the sampling distribution of the mean is called the **standard error**. We can calculate it from the simulated distribution using the [standard deviation formula](#). The result for our simulation is plotted in [blue](#) to the right. (We typically don't have this distribution in real life, so we can't simply calculate it. We have to estimate it.)



To estimate standard error from a sample we use the formula: s/\sqrt{n} . In real life, you usually have one sample to do this. But we have 1000 samples in our simulation, so we can calculate 1000 estimates. To see how good they are, we can calculate the difference between each estimate and the empirical standard error calculated above. Here is the distribution of those differences. As you can see, the mean is very close to 0. They are good estimates!



And now we can explain why we use standard error in our graphs

OK, so now we know that the standard error is a measure of how much sample means from the same population will vary.

So now we can use the following logic. If two sample means differ by a lot relative to the standard error, then they are either from different populations, or something relatively rare has occurred (e.g., something rare like we drew samples from two ends of the sampling distribution of the mean).

Cashing this logic out quantitatively is the domain of statistics (and we will learn some of this soon). But at least you can see why we use standard errors in our figure.

Since we are comparing means in our figures, the standard errors allow us to compare the size of the variability between means.

Again, the formula for the estimated standard error is standard deviation divided by the square root of the sample size, or s/\sqrt{n} . There is no built-in function for this in R, so it is good to memorize it.

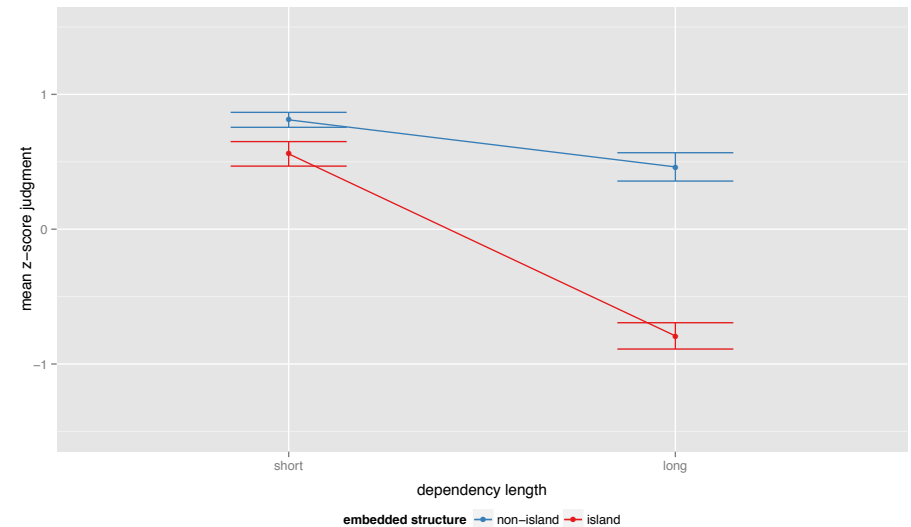


Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

Section 3:
Application

The most important lesson in stats: Statistics is a field of study, not a tool

Statistics is its own field. There is a ton to learn, and more is being discovered every day. Statisticians have different philosophies, theories, tastes, etc. They can't tell you the "correct" theory any more than we can tell them the "correct" theory of linguistics.

What we want to do is take this large and vibrant field, and convert it into a tool for us to use when we need it. This is a category mismatch.



Imagine if somebody tried to do that with linguistics. We would shake our heads and walk away...

But statistics is in a weird position, because other sciences do need the tools that they develop to get work done. And statistics wants to solve those problems for science. So we have to try to convert the field into a set of tools.

What you will run for (most) papers

Obviously, I am not qualified to teach you the actual field of statistics. And there is no way to give you a complete understanding of the “tool version” of statistics that we use in experimental syntax in the time we have here.

So here is my idea. I am going to start by showing you the R commands that you are going to run for (most) of your experimental syntax papers. Then we will work backwards to figure out exactly what information these commands are giving you.

1. Load the lmerTest package

```
library(lmerTest)
```

2. Create a linear mixed effects model with your fixed factors (e.g., factor1 and factor2) and random factors for subjects and items.

```
model.lmer=lmer(responseVariable~factor1*factor2 + (1+factor1*factor2|  
subject) + (1|item), data=yourDataset)
```

3. Run the anova() function to derive F statistics and *p*-values using the Satterthwaite approximation for degrees of freedom.

```
anova(model.lmer)
```

The results for our data

If we run the following code in the script called `linear.mixed.effects.models.r`:

```
wh.lmer = lmer(zscores~embeddedStructure*dependencyLength + (1|subject)
+ (1|item), data=wh)
```

And then use the `summary()` and `anova()` functions, we get the following results:

`summary(wh.lmer)`

```
Fixed effects:
              Estimate Std. Error      df t value Pr(>|t|)
(Intercept)    0.81129    0.06415 171.34000  12.647 < 2e-16 ***
embeddedStructure2 -0.25244    0.08789 192.15000  -2.872 0.004531 **
dependencyLength2 -0.34913    0.08789 192.15000  -3.973 0.000101 ***
embeddedStructure2:dependencyLength2 -1.00138    0.12458 192.33000  -8.038 9.06e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`anova(wh.lmer)`

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom
              Sum Sq Mean Sq NumDF  DenDF  F.value    Pr(>F)
embeddedStructure    31.616   31.616     1 192.32  146.189 < 2.2e-16 ***
dependencyLength     40.255   40.255     1 192.32  186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1 192.32   64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this section we want to try to understand what the model above is modeling, and what the information in the summaries is telling us.

Theories, models, and hypothesis tests

Substantive Theories

As scientists, theories are what we really care about. Substantive theories are written in the units of that science; e.g., syntactic theories are written in terms of features, operations, tree-structures, etc.

Mathematical Models

We want to find evidence for our theories. But what counts as evidence? One possible answer (among many) is: (i) a successful theory will predict observable data, therefore (ii) we can use a measure of how well a theory predicts the data as evidence for/against a theory. If we adopt this view, we need to link our theories to observable data in a way that lets us quantify that relationship. In short, we need a mathematical model that relates our theory to the data. This opens up lots of doors for us. We can create metrics to evaluate how good a model is, and compare models for goodness. And we can use probability theory to answer questions like “how likely is this data given this theory?”, “how likely is this theory given this data?”.

Hypothesis Tests

Once we have models, and metrics for comparing them, we may want to formalize a criterion for choosing one model/theory over another. In other words, a test.

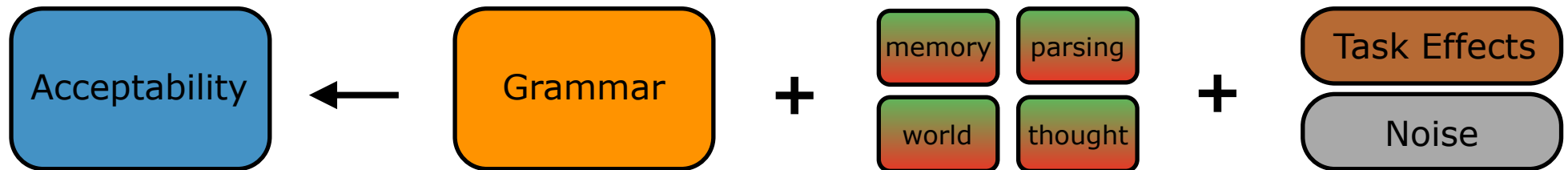
Constructing a model for our theory

The theory of wh-islands:

Our theory is that there is constraint on the extraction of wh-words out of embedded questions.

Our model:

We already have a model in mind for our theory. We think that this constraint will affect acceptability. So we need a model of acceptability that has a spot for this constraint.



So all we need to do is translate this model of acceptability into a specific equation for our experiment. Here is what it is going to look like:

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$

Now let's spend the next several slides building this equation so you can see where it came from.

This is a model to predict every judgment

We have 224 judgments in our dataset. We want a model that can explain every one of them. We capture this with the i subscript:

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$



This is shorthand for 1 to 224

$$\text{acceptability}_1 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_1$$

$$\text{acceptability}_2 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_2$$

$$\text{acceptability}_3 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_3$$

$$\text{acceptability}_4 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_4$$

...

$$\text{acceptability}_{224} = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_{224}$$

Also notice that when we write out the individual equations for each judgment in our dataset, certain other numbers become concrete. The subscript on the structure and dependencies factors becomes a specific number (0 or 1), and the i subscript on the ε term takes the same value as the judgment.

Coding the variables

The factors in our experiment are **categorical** (non-island/island, short/long).

Categorical variables can either be turned into 0 and 1 (treatment coding), or into -1 and 1 (effects coding). There is a difference between them that we will talk about in a few minutes. But for now, let's choose 0 and 1, like so:

structure

non-island = 0

island = 1

dependency

short = 0

long = 1

Now look at the first four equations below. Can you see which condition each one represents?

$$\text{acceptability}_1 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_1$$

$$\text{acceptability}_2 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_2$$

$$\text{acceptability}_3 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_3$$

$$\text{acceptability}_4 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_4$$

...

The first is **non-island** because its structure is 0, and it is **short** because its dependency is also 0. The fourth is **island** because its structure is 1, and it is **long** because its dependency is 1.

What are the Betas?

The betas in this equation are coefficients. They are the numbers that turn the 0s and 1s into an actual effect on acceptability.

The idea is that you multiply the beta by the 0 or 1 in the factor to get an effect. So when the factor is 0, there is no effect. And when the factor is 1, you get an effect that is the same size as the beta.

$$\text{acceptability}_1 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_1$$

$$\text{acceptability}_2 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_2$$

$$\text{acceptability}_3 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_3$$

$$\text{acceptability}_4 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_4$$

...

It is important to note that each beta is constant. β_1 is always β_1 . It doesn't have another subscript that varies for each judgment (unlike the ε term). This is why each beta can be seen as an effect.

β_1 is the effect of having **island** structure.

β_2 is the effect of having a **long** dependency.

structure₁:dependency₁ is the violation

The structure₁:dependency₁ term looks strange because it is the **interaction term** (the colon is a way of notating this). It is the special extra effect that occurs when the levels of the two factors are both 1. Basically, you multiply the two numbers together (0*0, 0*1, 1*0, or 1*1), and then multiply the result by β_3 .

In our **substantive theory**, this mathematical term captures the effect of a **violation**. The **island/long** condition (1,1) is the only condition that meets the structural description of the island constraint.

$$\text{acceptability}_1 = \beta_0 + \beta_1\text{structure}_0 + \beta_2\text{dependency}_0 + \beta_3\text{structure}_0:\text{dependency}_0 + \varepsilon_1$$

$$\text{acceptability}_2 = \beta_0 + \beta_1\text{structure}_0 + \beta_2\text{dependency}_1 + \beta_3\text{structure}_0:\text{dependency}_1 + \varepsilon_2$$

$$\text{acceptability}_3 = \beta_0 + \beta_1\text{structure}_1 + \beta_2\text{dependency}_0 + \beta_3\text{structure}_1:\text{dependency}_0 + \varepsilon_3$$

$$\text{acceptability}_4 = \beta_0 + \beta_1\text{structure}_1 + \beta_2\text{dependency}_1 + \beta_3\text{structure}_1:\text{dependency}_1 + \varepsilon_4$$

...

The interaction term does nothing for the first three conditions, because it is equivalent to a 0 then. In the fourth condition (1,1) it is a 1. In this condition, that 1 is multiplied by β_3 to add to the effect. This means that **β_3 is the size of the violation effect** (it is the **DD score** from earlier!). Note that this is only true with treatment (0,1) coding. The coefficients have different interpretations with different codings.

ε is the error term

If you just look at the betas and factors, you will quickly see that we can only generate 4 acceptability judgments: one for each condition in our experiment (00, 01, 10, 11). But we have 224 values that we need to model. And that is where the ε term comes in.

The ε term is an error term. It is the difference between the value that the model predicts and the actual value of the judgment. This is why it varies in its subscript: we need a different ε term for each judgment.

$$\text{acceptability}_1 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_1$$

$$\text{acceptability}_2 = \beta_0 + \beta_1 \text{structure}_0 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_2$$

$$\text{acceptability}_3 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_0 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_3$$

$$\text{acceptability}_4 = \beta_0 + \beta_1 \text{structure}_1 + \beta_2 \text{dependency}_1 + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_4$$

...

This may seem like a hack, but it is principled. The other parts of our model capture the things that we manipulated in our experiment. The error term captures all of the things that we couldn't control: individual differences in the participants, differences in the items, effects of the task, etc. (And we will see later that we can model some of these things, at least a little bit).

We minimize the ε 's to estimate β 's

Once you've specified your model (as we have here), the next step is to find the coefficients that make for a good model.

One way to define "good" is to say that a good model will minimize the amount of stuff that is unexplained. Well, all of our unexplained stuff is captured by the ε terms, so this means that we want to minimize ε .

Here is a toy example with 3 values and a simple model with only one beta:

Let's imagine we have three judgments to model (2,3,4). If we choose the value 4 for the coefficient of β_0 , we get ε terms (-2, -1, 0), which we can square and sum to derive a sum of squares.

$acc_i = \beta_0$	+	ε_i	
2 = 4	+	-2	
3 = 4	+	-1	SS=5
4 = 4	+	0	

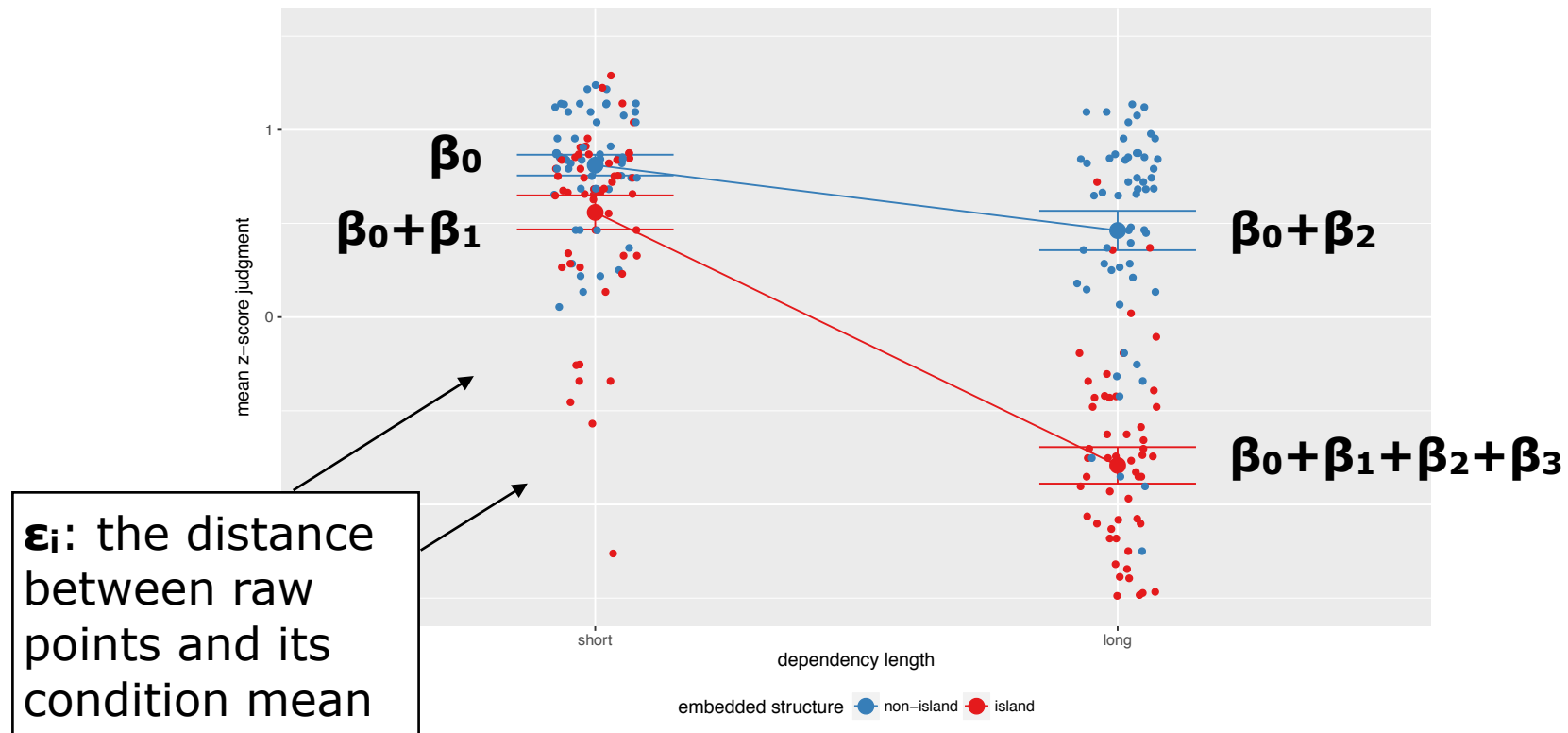
Now, let's imagine we have the same data, but we choose 3 for the coefficient of β_0 . Now we get smaller error terms, and consequently a smaller SS. This is a better model, because less is unexplained.

$acc_i = \beta_0$	+	ε_i	
2 = 3	+	-1	
3 = 3	+	0	SS=2
4 = 3	+	1	

Putting it all together

You specify the model for R. That was the command we entered into the console. R will then find the best value of the coefficients for the data that you gave it.

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$



And you might recall that this is exactly the 2x2 logic that we discussed earlier.

The R command

Now that we understand our linear model, we can compare it to the R command that we ran at the beginning of this section. I will color parts so that you can see the correspondence:

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$



```
lmer(zscores ~ embeddedStructure + dependencyLength + embeddedStructure:dependencyLength +  
(1|subject) + (1|item), data=wh)
```

You don't need to specify the intercept (β_0) in the command. R includes one by default (you can, however, tell it not to estimate an intercept if you want).

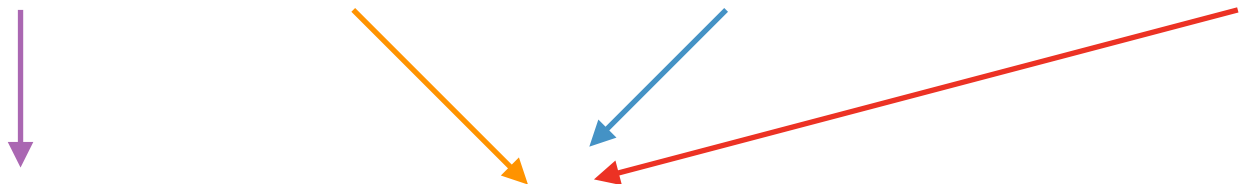
You don't need to specify the error term (ε_i) in the command. Again, R includes one by default.

You will also notice that lmer() formula contains extra bits: (1|subject) and (1|item). That is because the top model only has **fixed effects**. The (1|subject) and (1|item) terms are **random effects**. We will turn to those next.

The R command - a shortcut

You may have noticed that the command I just showed you is not exactly the command in the script (or on the slide at the beginning of this section). That is because there is a shortcut in R for specifying two factors and an interaction:

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$



```
lmer(zscores ~ embeddedStructure * dependencyLength + (1|subject) + (1|item), data=wh)
```

When you want all three effects, you can use the `*` operator instead of a `+`. R will automatically expand this to all three components:

embeddedStructure
dependencyLength
embeddedStructure:dependencyLength

It is a nice shortcut that really saves you time if you have more than two factors, because they grow in squares (remember, a 2x2x2 will have 8 components, and 2x2x2x2 will have 16).

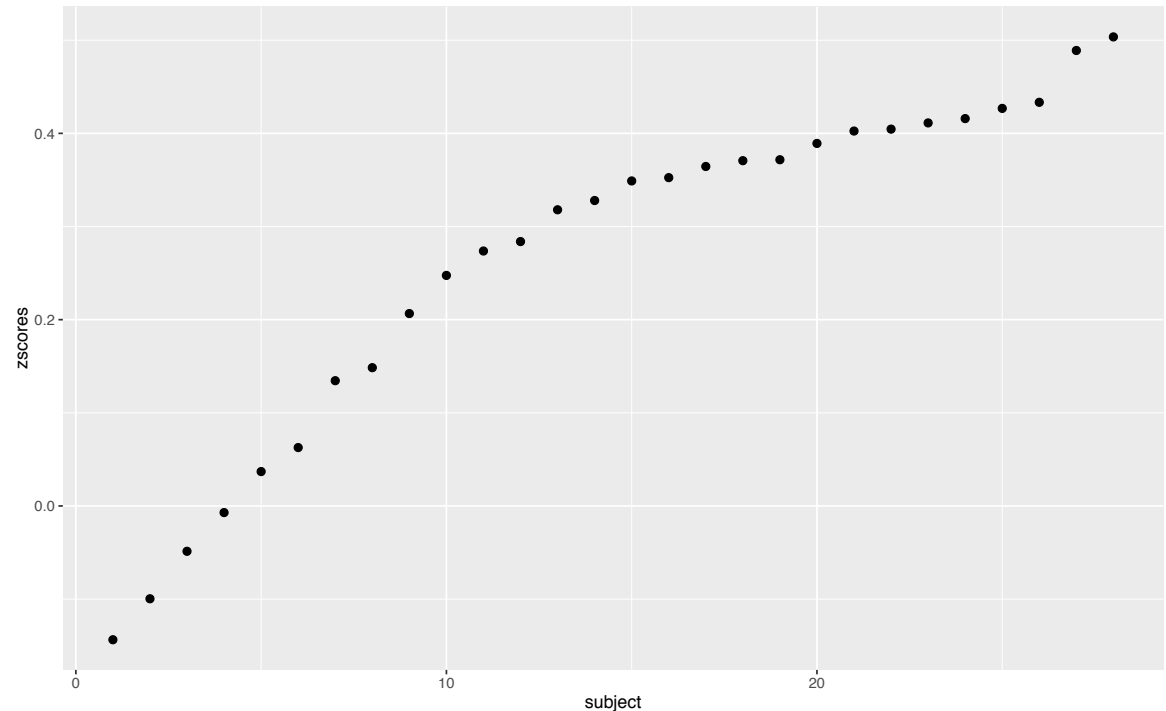
Subject differences

Let's talk about the first term (1|subject). As the name suggests, this term captures differences between the subjects in our dataset.

```
lmer(zscores ~ embeddedStructure * dependencyLength + (1|subject) + (1|item), data=wh)
```

The plot at the right shows the mean rating of the 4 experimental conditions for each subject. As you can see, there is quite a bit of variability.

The (1|subject) term in the model tells R to estimate an intercept for each subject. This intercept is added to each subjects judgments to try to account for these differences.



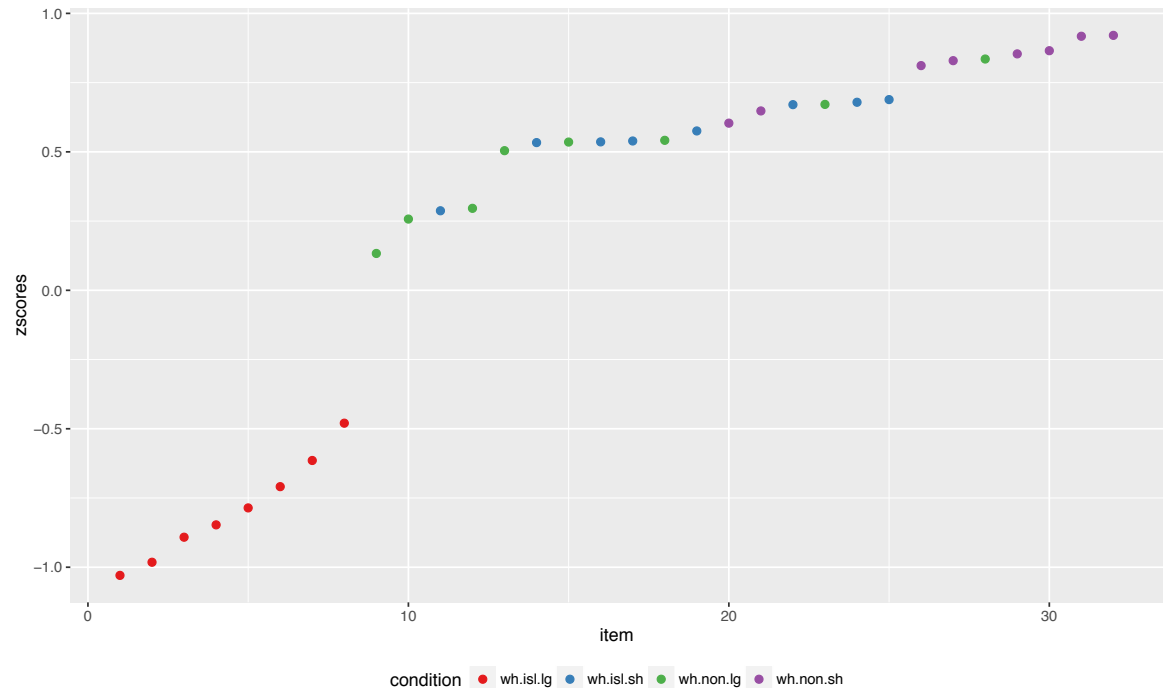
Basically, instead of having these subject differences contaminate the effects of interest, or having these differences sit in an error term, this asks the model to estimate them. The code for this plot is in [subject.item.differences.r](#).

Item differences

The second term, (1|item), is similar. As the name suggests, this term captures differences between the items in our dataset.

```
lmer(zscores ~ embeddedStructure * dependencyLength + (1|subject) + (1|item), data=wh)
```

Once again, we can plot the means of each item to see their differences. Now, we expect differences between items based on their condition. But as you can see by the colors (colors = condition), there are differences between items within a single condition.



This code asks R to estimate an intercept for each item, and add it whenever that item is being modeled. This makes sure that it isn't contributing to the other (important) effects, or to the error term. The code for this plot is in [subject.item.differences.r](#).

Fixed factors vs Random factors

Now, you may have noticed that our experimental factors look different from these subject and item factors in the R command. This is because the former are **fixed factors** and the latter are **random factors**.

```
lmer(zscores ~ embeddedStructure * dependencyLength + (1|subject) + (1|item), data=wh)
```



There are two common ways to define the difference between fixed and random factors. The first is operational, the second is mathematical:

1. **Fixed factors** are factors whose **levels must be replicated exactly** in order for a replication to count as a replication.

Random factors are factors whose **levels will most likely not be replicated exactly** in a replication of the experiment.

2. **Fixed factors** are factors whose **levels exhaust the full range of possible level values** (as they are defined in the experiment).

Random factors are factors whose **levels do not exhaust the full range of possible level values**.

Random intercepts and slopes

One last note about random factors. So far, we've only specified random intercepts — one value for each subject and one value for each item. But we can also specify **random slopes**. A random slope specifies a different value based on the values of the fixed factors (remember in our linear model, it is the fixed factors that specify the slopes of the lines).

The code for this looks complicated at first glance, but it isn't. We simply copy the fixed factor structure into the random subject term:

```
lmer(zscores ~ embeddedStructure * dependencyLength +  
(1+embeddedStructure*dependencyLength|subject) + (1|item), data=wh)
```

The 1 in the code tells R to estimate an intercept for each subject. The next bit tells R to estimate three more random coefficients per subject: one for embeddedStructure, one for dependencyLength, and one for the interaction embeddedStructure:dependencyLength.

There is a “best practices” claim in the field (Barr et al. 2013) that you should specify the “maximal” random effects structure licensed by your design. These means specifying random slopes if your design allows it.

The problem is that maximal random effects structure sometimes don't converge (R can't find a solution). In that case, you need to use a simpler model like an intercepts-only model.

This is a linear mixed effects model

A model that only has fixed effects is usually just called a linear model, though it is perhaps more correctly a linear fixed effects model.

A model that has both fixed factors and random factors is called a mixed model, so if it is linear, it is a linear mixed effects model.

```
lmer(zscores ~ embeddedStructure * dependencyLength + (1|subject) + (1|item), data=wh)
```



In R, there is a package called lme4 that exists to model linear mixed effects models. You could load lme4 directly, and create the linear mixed effects model above. The function lmer() is a function from lme4.

We are using the package lmerTest to run our models. The lmerTest package calls lme4 directly (when you installed it, it also installed lme4). The reason we are using lmerTest is that lmerTest also includes some functions that let us calculate inferential statistics, like the F-statistic, and p-values. The lme4 package doesn't do that by itself.

The Random slopes model in our script

Our script [linear.mixed.effects.models.r](#) contains the code for both an intercept-only model and a random slopes model. You should try running them.

What you will find is that the intercept-only model runs fine, but the slopes model fails to converge. Like I said, this happens with random slopes models.

It turns out that the problem with the model is our coding of the factors. We used [treatment coding](#), but for some reason (that I don't understand), the coding is causing a problem for the model.

The model will converge with a different coding scheme called [effect coding](#). This appears to be a pattern: many random slopes models will fail to converge with treatment coding, but succeed with effect coding.

So what should we do? Well, the coding doesn't affect things like F-statistics, t-statistics, and p-values. Those will be the same regardless of the coding scheme. So if that is all you care about, go ahead and change the coding.

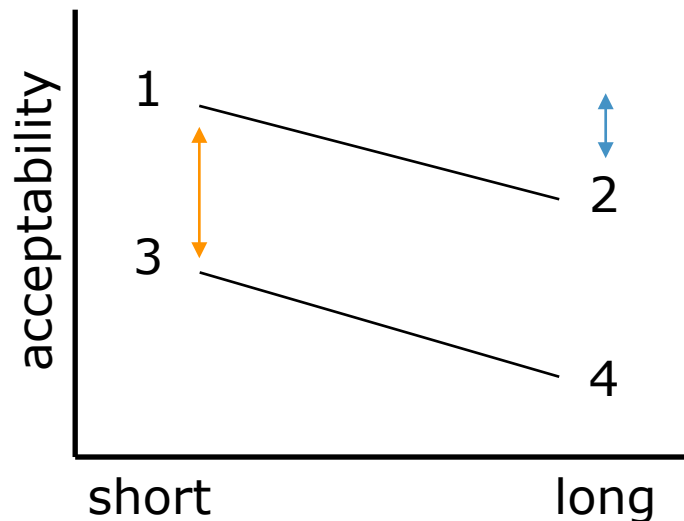
What does change is the interpretation of the coefficients in the model. In the next few slides, I will show you this change in interpretation. But the bottom line is that if the interpretation is important to you, you either need to drop the random slopes, or translate the effect coding estimates into treatment coding estimates by hand.

Simple effects vs Main effects

The first step to understanding the difference between treatment coding and effect coding is to understand the difference between simple effects and main effects:

Simple effects are a difference between two conditions.

Typically, a simple effect is defined relative to one condition, the baseline condition. So if condition 1 were the baseline condition, we could define two simple effects:



The effect of 1 vs 2.

The effect of 1 vs 3.

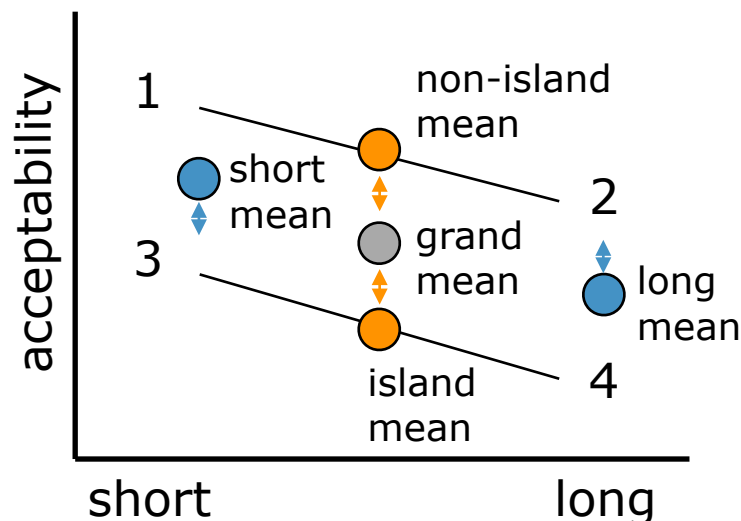
The effect of 1 vs 4 is the sum of these two (in this example).

Simple effects vs Main effects

The first step to understanding the difference between treatment coding and effect coding is to understand the difference between simple effects and main effects:

Main effects are the difference between the grand mean of all conditions and the average of one level across both levels of the other factor.

Again, in a 2x2 design we can define two main effects: `embeddedStructure` and `dependencyLength`. Each one goes in two directions (one positive, one negative)



The blue arrows are the main effect of `dependencyLength` (positive and negative change from the grand mean)

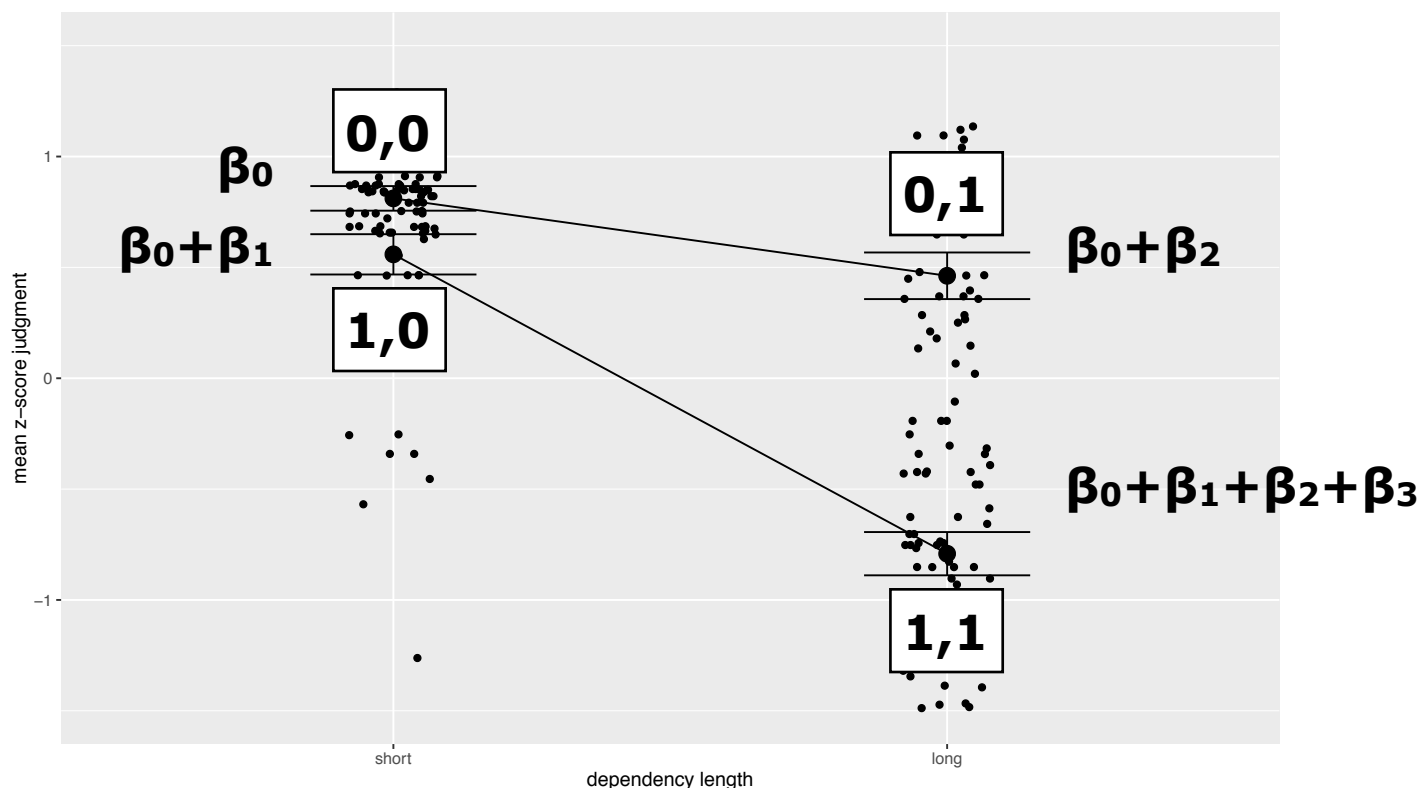
The orange arrows are the main effect of `embeddedStructure` (positive and negative change from the grand mean)

Each condition is a combination of the two main effects (in this example).

Treatment coding reveals simple effects

In [treatment coding](#), each level is either 0 or 1. This is what we've been using so far. Treatment coding is great when one of your conditions can be considered a baseline in your theory.

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$

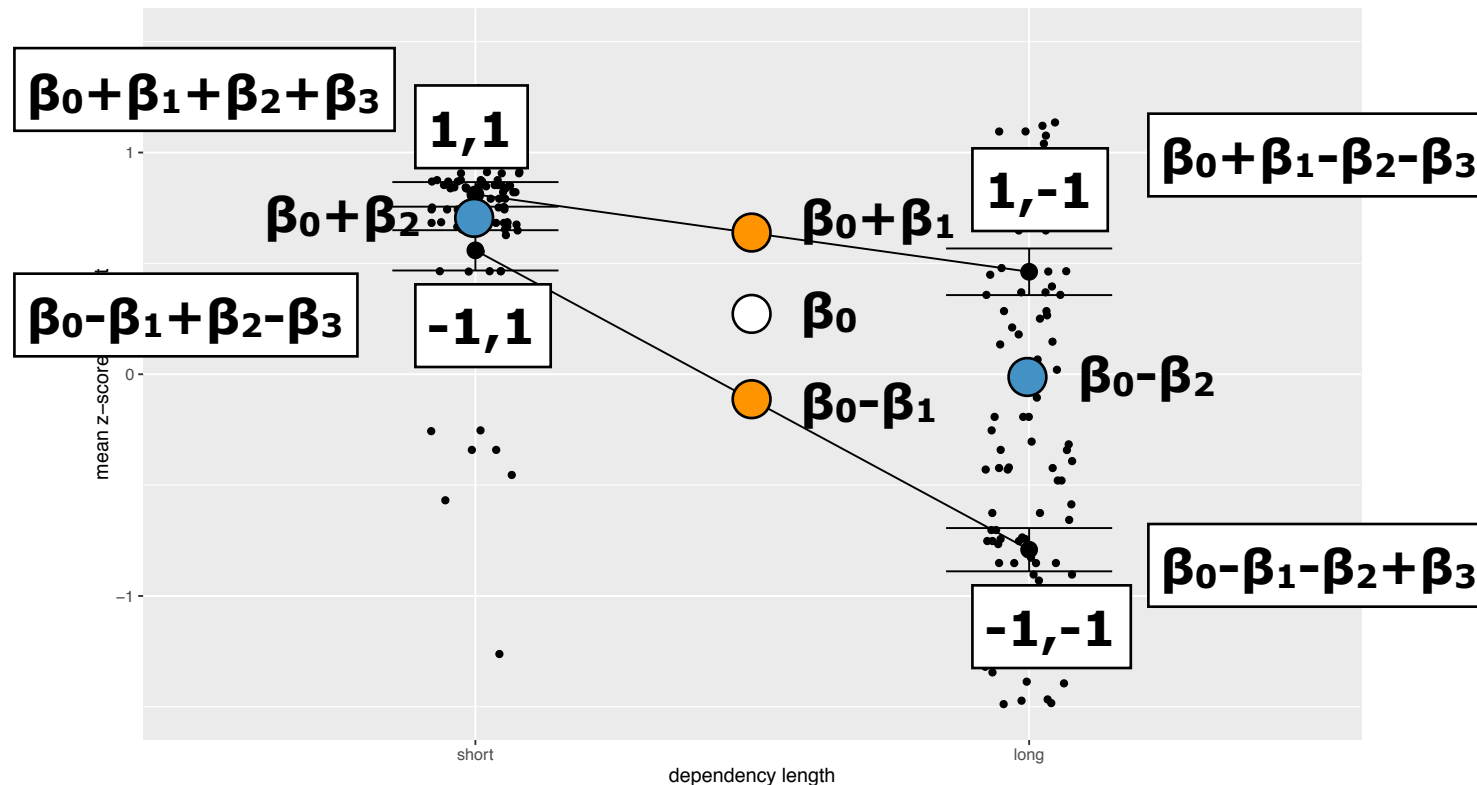


Treatment coding coefficients show you **simple effects**: the difference between the baseline condition and another condition. It works well for some designs, and less so for others (e.g., when you have no clear baseline).

Effect coding reveals main effects

In **effect coding**, the factors are given the values **1 or -1**. This doesn't change the model that we specify, but it changes the interpretation of the coefficients. Effects coding is helpful when there is no clear "baseline" condition.

$$\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(1,-1)} + \beta_2 \text{dependency}_{(1,-1)} + \beta_3 \text{structure}_1 : \text{dependency}_1 + \varepsilon_i$$



Effect coding coefficients show you main effects. But be careful. Main effects are not straightforward to interpret when there is an interaction (because the interaction contaminates them).

Choosing a contrast coding

Contrast coding is primarily about interpreting the coefficients in your model. If you don't care about trying to interpret those, then the contrast coding scheme will rarely matter. Contrast coding has **no** effect on statistics like F and t, and will not impact the p-values that F-tests and t-tests give you.

If you care about interpreting the coefficients, then you have to use your scientific knowledge to figure out which one is best for you.

Treatment coding is best if you have a clear baseline condition, and care about simple effects (differences from the baseline).

Effect coding is best when you don't have a clear baseline, or when you care about main effects (average effects of a factor). If you do care about main effects, remember that the presence of an interaction makes it impossible to interpret main effects (because the interaction contaminates them).

Finally, there are two times where it is better, mathematically, to use **effect coding**:

1. Some random slopes models won't converge with **treatment coding**, but will converge with **effect coding** (like our random slope model).
2. If you are mixing categorical and continuous factors, **treatment coding** can introduce heteroscedasticity (variable variance). **Effect coding** does not.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1: Design

Section 2: Analysis

Section 3: Application

Let's look at the coefficients of the intercept model (wh.lmer)

Here is the output of `summary(wh.lmer)` for treatment coding:

```
Fixed effects:
              Estimate Std. Error    df t value Pr(>|t|)
(Intercept)    0.81129    0.06415 171.34000  12.647  < 2e-16 ***
embeddedStructure2 -0.25244    0.08789 192.15000  -2.872  0.004531 **
dependencyLength2 -0.34913    0.08789 192.15000  -3.973  0.000101 ***
embeddedStructure2:dependencyLength2 -1.00138    0.12458 192.33000  -8.038  9.06e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The β 's for the model are listed under Estimate. Go ahead and check these numbers against the graph of our condition means.

Here is the output of `summary(wh.lmer)` for effect coding:

```
Fixed effects:
              Estimate Std. Error    df t value Pr(>|t|)
(Intercept)    0.26016    0.03497  27.13000   7.439 5.14e-08 ***
embeddedStructure1  0.37657    0.03114 192.33000  12.091  < 2e-16 ***
dependencyLength1  0.42491    0.03114 192.33000  13.643  < 2e-16 ***
embeddedStructure1:dependencyLength1 -0.25034    0.03114 192.33000  -8.038  9.06e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Just for fun, we can also look at the β 's from effect coding. As you can see, they are very different. You can check them against the β 's for treatment coding (you can translate between the two using the formulae in the previous slides, though it takes some effort).

Also notice there are some statistical things to the right in these readouts, such as t values and p-values... and notice that **they don't change based on coding!**

Anova(wh.lmer) yields F-statistics and p-values

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom

              Sum Sq Mean Sq NumDF  DenDF F.value    Pr(>F)
embeddedStructure      31.616   31.616     1 192.32 146.189 < 2.2e-16 ***
dependencyLength       40.255   40.255     1 192.32 186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1 192.32  64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Although the `summary()` function had statistics in it (t statistics and p-values), I want to focus on the `anova()` function. This is the same information that you would get from a fixed effects ANOVA, which I think is useful for relating mixed effects models to standard linear models.

There are two pieces of information here that I want to explain in more detail: the **F statistic** and the **p-value**. These are the two pieces of information that `anova()` adds to our interpretation. With that, we will have (i) the graphs, (ii) the model and its estimates, (iii) the F statistic, and (iv) the p-value. Together, those 4 pieces of information provide a relatively comprehensive picture of our results.

Someday, it will be worth it for you to explore the Sum of Squares and df values, but for now, we can set them aside as simply part of the calculation of F's and p's respectively.

The F statistic is about evaluating models

There are two common dimensions along which models are evaluated: their adequacy and their simplicity.

1. **Adequacy:** We want a model that minimizes error

We've already encountered this. We used **sum of squares** to evaluate the amount of error in a model. We chose the coefficients (the model) that minimized this error.

2. **Simplicity:** We want a model that estimates the fewest parameters

We can measure simplicity with the **number of parameters that are estimated from the model**. A model that estimates more parameters is more complicated, and one that estimates fewer parameters is simpler.

The intuition behind this is that **models are supposed to teach us something**. The more the model uses the data, the less the model itself is contributing.

The models we've been constructing are estimating 4 parameters from the data: β_0 , β_1 , β_2 , and β_3

Degrees of Freedom as a measure of simplicity

We can use **degrees of freedom** as a measure of **simplicity**.

$df = \text{number of data points} - \text{number of parameters estimated}$

$$df = n - k$$

Notice that df makes a natural metric for simplicity for three reasons:

1. It is based on the number of parameters estimated, which is our metric.
2. It captures the idea that a model that estimates 1 parameter to explain 100 data points ($df=99$) is better than a model that estimates 1 parameter to explain 10 ($df=9$).
3. The values of df work in an intuitive direction: higher df is better (simpler) and lower df is worse.

In practice, there is a tension between adequacy and simplicity

Adequacy seeks to minimize error. **Simplicity** seeks to minimize the number of parameters that are estimated from the data.

Imagine that you have 224 data points, just like our data set. A model with 224 parameters would predict the data with no error whatsoever because each parameter would simply be one of the data points. (This the old saying “the best model of the data is the data.”). This would have perfect adequacy.

But this model would also be the most complicated model that one can have for 224 data points. It would teach us nothing about the data.

This tension is not a necessary truth. There could be a perfect model that predicts all of the data without having to look at the data first. But in practice, there is a **tension between adequacy and simplicity**.

To put this in terms of our metrics, this means there will be a tension between **sum of squares** and **degrees of freedom**.

So what we want is a way to **balance this tension**. We want a way to know if the df we are giving up for lower error is a good choice or not.

A transactional metaphor

One way to think about this is with a metaphor. As a modeler, you want to eliminate error. You can do this by spending df. If you spend all of your df, you would have zero error. But you'd also have no df left. We have to assume that df is inherently valuable (you lose out on learning something) since you can spend it for stuff (lower error). So you only want to spend your df when it is a good value to do so.

Thinking about it this way, the question when comparing models is **whether you should spend a df to decrease your error**. The simple model keeps more df. The complex model spends it. The simple model has more error. The complex model has less error because it spent some df. Which one should you use?

Simple: spends no df

$Y_i = \beta_0$	+	ϵ_i
2 = 4	+	-2
3 = 4	+	-1
4 = 4	+	0
df=3		SS=5

Complex: spent a df

$Y_i = \beta_0$	+	ϵ_i
2 = 3	+	-1
3 = 3	+	0
4 = 3	+	1
df=2		SS=2

A transactional metaphor

When you are faced with the prospect of spending df, there are two questions you ask yourself:

1. How much (lower) error can I buy with my df?
2. How much error does df typically buy me?

In other words, you want to compare the value of your df (in this particular instance), with the value of your df in general. If the value here is more than the value in general, you should spend it. If it is less, you probably shouldn't spend it, because that isn't a good deal.

We can capture this with a ratio:

How much error can I buy with my df?

How much error does df typically buy me?

If the ratio is high, it is a good deal, so you spend your df. If the ratio is low, it is a bad deal, so you don't spend your df.

The F ratio

To cash out this intuition, all we need to do is calculate how much you can buy with your df, and then calculate the value you can expect for a df, and see if you are getting a good deal by spending the df.

the amount of error you can buy with a df = $(SS_{\text{simple}} - SS_{\text{complex}}) / (df_{\text{simple}} - df_{\text{complex}})$

the amount of error df typically buys = $SS_{\text{complex}} / df_{\text{complex}}$

Let's take a moment to really look at these equations.

The first takes the difference in error between the models and divides it by the difference in df. So that is telling you how much error you can eliminate with the df that you spent moving from one model to the next. Ideally, you would only be moving by 1 df to keep things simple.

The second equation takes the error of the complex model and divides it by the number of df in that model, giving you the value in error-elimination for each df. The complex model has the lowest error of the two models, so it is a good reference point for the average amount of error-elimination per df.

The F ratio

So now what we can do is take these two numbers, and create a ratio:

$$F = \frac{(SS_{\text{simple}} - SS_{\text{complex}})/(df_{\text{simple}} - df_{\text{complex}})}{SS_{\text{complex}}/df_{\text{complex}}}$$

If F stands for the ratio between the amount of error we can buy for a df and a typical value for a df, then we can interpret it as follows:

If F equals 1 or less, then we aren't getting a good deal for our df. We are buying relatively little error by spending it. So we shouldn't spend it. We should use the simpler model, which doesn't spend the df.

If F equals more than 1, we are getting a good deal for our df. We are buying relatively large amounts of error-reduction by spending it. So we should spend it. We should use the more complex model (which spends the df) in order to eliminate the error (at a good value).

The F ratio is named after Ronald Fisher (1890-1962), who developed it, along with a lot of methods in 20th century inferential statistics.

Our toy example

Here are our two models:

simple			complex		
$Y_i = \beta_0$	+	ε_i	$Y_i = \beta_0$	+	ε_i
2 = 4	+	-2	2 = 3	+	-1
3 = 4	+	-1	3 = 3	+	0
4 = 4	+	0	4 = 3	+	1
df=3		SS=5	df=2		SS=2

$$F = \frac{(SS_{\text{simple}} - SS_{\text{complex}})/(df_{\text{simple}} - df_{\text{complex}})}{SS_{\text{complex}}/df_{\text{complex}}} = \frac{(5-2)/(3-2)}{2/2} = 3$$

So in this case the F ratio is 3, which says that we can buy three times more error-elimination for this df than we would typically expect to get. So that is a good deal, and we should use that df. So the complex model is better by this metric (the F ratio).

Our real example

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom

              Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
embeddedStructure      31.616   31.616     1 192.32 146.189 < 2.2e-16 ***
dependencyLength      40.255   40.255     1 192.32 186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1 192.32  64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now let's look again at the output of the `anova()` function (which calculates F's) for our example data.

The first F in the list is for the factor `embeddedStructure`. This F is comparing two models:

simple: $\text{acceptability}_i = \beta_0$

complex: $\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)}$

The resulting F ratio is 146:1, so yes, the structure factor is pretty good value for the df spent.

Our real example

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom

              Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
embeddedStructure      31.616   31.616     1 192.32 146.189 < 2.2e-16 ***
dependencyLength      40.255   40.255     1 192.32 186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1 192.32  64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The second F in the results is for `dependencyLength`. Again, this is comparing two models:

simple: $\text{acceptability}_i = \beta_0$

complex: $\text{acceptability}_i = \beta_0 + \beta_2 \text{dependency}_{(0,1)}$

The resulting F ratio is 186:1, so yes, the dependency factor is pretty good value for the df spent.

Our real example

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom

              Sum Sq Mean Sq NumDF  DenDF  F.value    Pr(>F)
embeddedStructure      31.616   31.616     1 192.32  146.189 < 2.2e-16 ***
dependencyLength      40.255   40.255     1 192.32  186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1 192.32   64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The final F is for the interaction of the two factors. This is still comparing two models, but in this case, the simpler model is the model with the two main effects present with no interaction (+), and the complex model adds the interaction (*):

simple: $\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} + \beta_2 \text{dependency}_{(0,1)}$

complex: $\text{acceptability}_i = \beta_0 + \beta_1 \text{structure}_{(0,1)} * \beta_2 \text{dependency}_{(0,1)}$

The resulting F ratio is 64:1, which again, is a good value, and suggests that it was a good idea to add the interaction term.

Model comparison is not hypothesis testing

Let's be clear: model construction and comparison is its own exercise. Nothing we have done so far has been a formalization of a hypothesis test. We've just been talking about how to construct models, and how to compare two models that we've constructed using information that seems useful.

Also, there are other metrics for model evaluation and comparison that you should explore: adjusted R^2 , BIC, AIC, etc.

I want to stress the fact that you can be interested in model construction and model comparison for its own purposes. Models are a tool that allows you to better understand your research question. You can see exactly how different factors contribute to the dependent variable.

This distinction between model construction/comparison and hypothesis testing is why lme4 doesn't come with p-values. It is a tool for model construction and comparison, while p-values are a tool for hypothesis testing.

That being said, I wouldn't make you learn about F ratios if they couldn't be used for hypothesis testing. And lmerTest, which as the name suggests is designed to turn linear mixed effects models into hypothesis tests, wouldn't give you the F's if they weren't useful for tests. So let's do that now.

Null Hypothesis Significance Testing

When people think of hypothesis testing, the first approach that comes to mind is [Null Hypothesis Significance Testing](#).

NHST was not the first approach to statistics that was developed ([Bayes Theorem](#) is from 1763, Karl Pearson developed many components of statistics in the 1890s and 1900s, Gosset developed the t-test in 1908). NHST is also not the currently ascendant approach (Bayesian statistics are ascending).

But NHST dominated 20th century statistics (both in theory and practice) so it is still a standard approach in experimental psychology, and it is very much necessary for reading papers published in the last 75 years.

Pedagogically speaking, I am not sure if it is better to begin with NHST, and then move to Bayes, or better to start with Bayes, and then move to NHST. For now, I think it is safer to start with NHST, and move to Bayes if you are interested.

That way, even if you don't have the time to look into Bayes in detail, you still have the NHST tools necessary to (i) publish papers, and (ii) read existing papers. You can cross the Bayes bridge if the field ever comes to it.

Two approaches to NHST

It turns out that there are two major approaches to NHST. They are very similar in mathematical appearance, so it is easy to think that they are identical. But they differ philosophically (and in some details), so it is important to keep them separated.

Ronald Fisher was the first person to try to wrangle the growing field of statistics into a unified approach to hypothesis testing. His NHST was the first attempt, and may still be the closest to the way scientists think about NHST. We'll start with the **Fisher approach**.

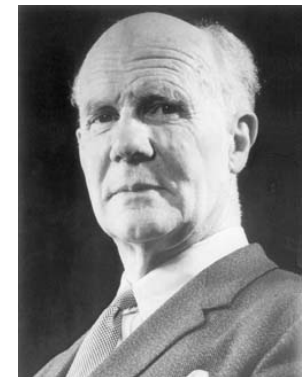


Ronald A. Fisher
(1890-1962)

Neyman and **Pearson** were fans of Fisher's work, but thought there were some deficiencies in his approach. So they tried to rectify that. It turns out that they simply had a different conception of probability and hypothesis testing. We'll talk about the Neyman-Pearson approach second.



Jerzy Neyman
(1894-1981)



Egon Pearson
(1895-1980)

Fisher's NHST

Under Fisher's NHST, there is only one hypothesis under consideration. Perhaps ironically, it is the most uninteresting hypothesis you could consider. It is called the **null hypothesis**, or H_0 .

H₀: The **null hypothesis**. This states that there is no effect in your data (e.g., no difference between conditions, no interaction term, etc).

For Fisher's NHST, the goal of an experiment is to **disprove the null hypothesis**.



"Every experiment may be said to exist only in order to give the facts a chance of disproving the null hypothesis." - Fisher (1966)

To do this, Fisher's NHST calculates the probability of **the observed data** under **the assumption that the null hypothesis is true**, or $p(\text{data} | \text{null hypothesis})$.

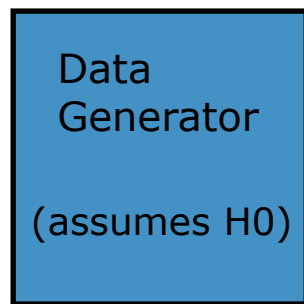
This leads to **Fisher's disjunction**:

If $p(\text{data} | \text{null hypothesis})$, called the p-value, is low, then you can conclude either: (i) the null hypothesis is incorrect, or (ii) a rare event occurred.

Fisher's NHST logic, stated a different way

There are two steps to a statistical test under Fisher's NHST approach:

Step 1: Calculate $P(\text{data} \mid \text{null hypothesis})$



data1
data2
data3
...

One way to think about this is that you are creating a data generating device that assumes the null hypothesis, and generates **all possible data sets**.

Then you use the distribution of generated data to calculate the probability of the observed data

$$P(\text{data} \mid H_0) = \frac{\text{observed data}}{\text{generated data}}$$

Step 2: Make an inference about the null hypothesis

For Fisher, $p(\text{data} \mid H_0)$ is a measure of the strength of evidence against the null hypothesis. If it is low, that is either strong evidence against the null hypothesis, or evidence that something really rare occurred.

This logic is enough to interpret our results

Once we have the logic of NHST, we can go back to the results that R and lmerTest gave us, and interpret those results. (Sure, it would be nice to be able to calculate the results for ourselves, but R does this for us.)

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom

              Sum Sq Mean Sq NumDF  DenDF  F.value    Pr(>F)    ***
embeddedStructure      31.616   31.616     1  192.32  146.189 < 2.2e-16 ***
dependencyLength       40.255   40.255     1  192.32  186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1  192.32   64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The first thing to note is that in this case, our p-values are in scientific notation. This is because they are really small:

structure	p = .0000000000000000022
length	p = .0000000000000000022
interaction	p = .00000000000000009048

These are incredibly small, so under Fisher's logic, we say that there is either very strong evidence that the null hypothesis is false, or something very rare occurred (i.e., the null hypothesis is true, but we got a result at the very end of the distribution of possible null hypothesis results).

This logic is enough to interpret our results

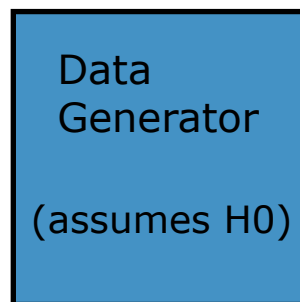
Once we have the logic of NHST, we can go back to the results that R and lmerTest gave us, and interpret those results. (Sure, it would be nice to be able to calculate the results for ourselves, but R does this for us.)

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom

              Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)    ***
embeddedStructure      31.616   31.616     1 192.32 146.189 < 2.2e-16 ***
dependencyLength       40.255   40.255     1 192.32 186.133 < 2.2e-16 ***
embeddedStructure:dependencyLength 13.973   13.973     1 192.32  64.611 9.048e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The second thing to note is that these p-values are based on the F statistics that lmerTest calculated for each effect.



F1
F2
F3
...

In principle, you can use any summary statistic you want (and you may know that there are many summary statistics in the literature). You could even use the sample mean.

The F is a nice statistic to use because it gives us even more information than just a p-value — remember, it tells us how much value we got for that df.

This logic is enough to interpret our results


Once we have the logic of NHST, we can go back to the results that R and lmerTest gave us, and interpret those results. (Sure, it would be nice to be able to calculate the results for ourselves, but R does this for us.)

Here is the output of `anova(wh.lmer)` for both coding types:

```
Analysis of Variance Table of type III with Satterthwaite
approximation for degrees of freedom
```

	Sum Sq	Mean Sq	NumDF	DenDF	F.value	Pr(>F)	
embeddedStructure	31.616	31.616	1	192.32	146.189	< 2.2e-16	***
dependencyLength	40.255	40.255	1	192.32	186.133	< 2.2e-16	***
embeddedStructure:dependencyLength	13.973	13.973	1	192.32	64.611	9.048e-14	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



Finally, note that the readout puts asterisks next to the p-values to tell you if they are below .05, .01, etc.

It is tempting to think of this as just a nice way to quickly visualize the results, but there is something much deeper going on here. The **precise p-value is necessary for the Fisher approach to NHST**, the **asterisks are there for the Neyman-Pearson approach**.

We will talk about this more later, but in a nutshell, the Neyman-Pearson approach asks whether the p-value is below a pre-specified threshold. The exact number doesn't matter, it is just whether it is below the threshold. These asterisks implement several common thresholds.

The logic of Fisher p-values

First and foremost, p-values are only one small piece of information. You also have your graphs, the model coefficients, and evaluation statistics like F.

But if you are going to use p-values, you need to be clear about what the p-value is telling you. It is the probability of obtaining the observed results, or results more extreme, under the data generation model of the null hypothesis).

Here are some other bits of information you may want to know. Unfortunately, p-values are not these other things:

1. The probability of the null hypothesis being true: $p(H_0 \mid \text{data})$
2. The probability of your hypothesis of interest being true: $p(H_1 \mid \text{data})$
3. The probability of incorrectly rejecting the null hypothesis (a false rejection).
4. The probability that you can replicate your results with another experiment.

The problem is that plenty of people think that p-values give these bits of information. That is false. There are literally dozens of papers out there trying to correct these misconceptions.

The math underlying NHST

Though the logic is enough to interpret the results that R and lmerTest give us, you may want to study the math that NHST approaches use to generate the reference distribution for the null hypothesis. It will give you the flexibility to run (and even create) your own analyses, and it will help you understand the hypothesis tests at a deeper level.

There are basically three approaches to generating the null reference distribution in NHST. I will review each briefly in the next few slides:

1. Randomization methods.

The basic idea is to take your observed data points, and randomize the condition labels that you attach to them.

2. Bootstrap methods.

The basic idea is to use your sample as a population, and sample from it to generate a (population-based) reference distribution.

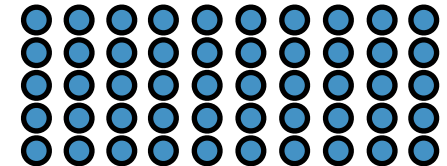
3. Analytic methods.

Most people imagine analytic methods when they think of stats. The idea here is that there are test statistics whose distribution is invariant under certain assumptions. We can use these known distributions to calculate p-values analytically (with an equation).

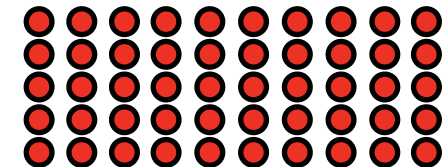
Randomization Methods

Let's use an example to demonstrate how to generate a reference distribution for the null hypothesis using randomization. Let's focus on two conditions for simplicity:

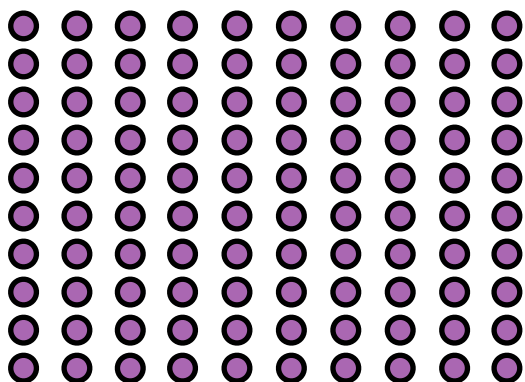
control: What do you think that Jack stole ___?



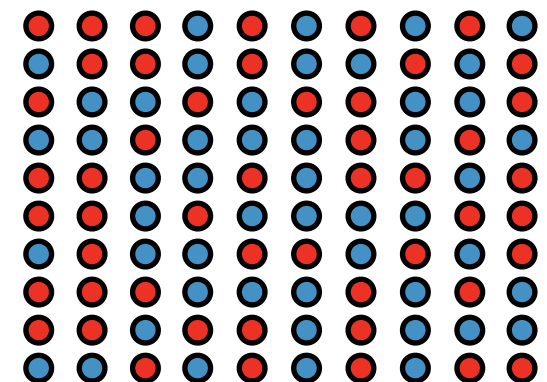
target: * What do you wonder whether Jack stole ___?



Here is the critical insight of randomization tests: Even though I have **labeled** these observations **control** and **target**, under the null hypothesis they are all just from the same label, **null**. So, this assignment of labels is arbitrary under the null hypothesis. And if the assignment is arbitrary, then I should be able to randomly re-arrange the labels.

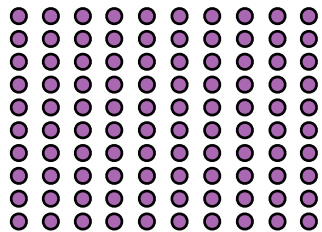


Randomly assign labels to these points because these labels are arbitrary under the null hypothesis.

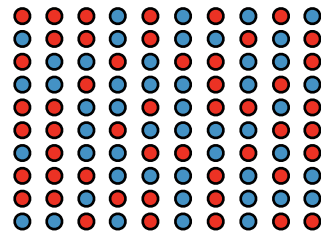


Randomization: generating the distribution

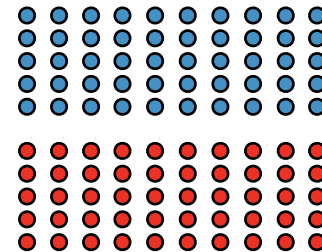
Start with the full data set



Randomly assign labels



Calculate the test statistic

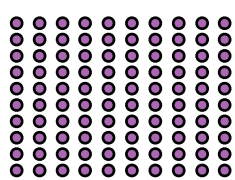


$$\rightarrow \bar{x}_c = 1$$

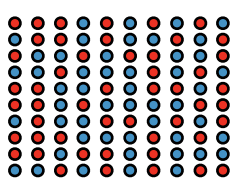
$$\rightarrow \bar{x}_t = .3$$

$$\rightarrow \bar{x}_c - \bar{x}_t = .7$$

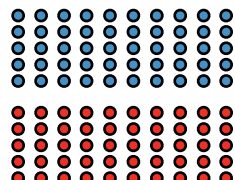
Then we repeat the process. With small samples, we can create every possible combination of labels, and have a complete distribution of possible test statistics. With large samples, this isn't possible, so we collect a large number of randomizations, like 10,000, and approximate the distribution.



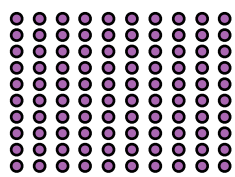
→



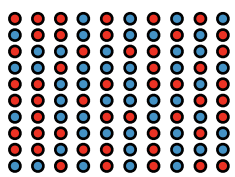
→



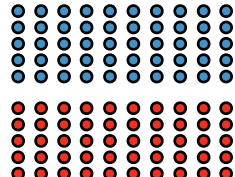
$$\bar{x}_c - \bar{x}_t = -.5$$



→



→



$$\bar{x}_c - \bar{x}_t = .3$$

We then collect all of the test statistics together to form a reference distribution under the null hypothesis.

See [randomization.r](#) for code to do this!

... to completion or 10,000

Randomization: calculating a p-value

Now that we have a reference distribution, we ask the following question: **What is the probability of obtaining the observed result, or one more extreme, given this reference distribution?**

We say “or one more extreme” for two reasons. First, we can’t just ask about one value because our response scale is continuous (most likely, the probability of one value is 1/the number of values in our distribution). Second, if we have to define a bin, “more extreme” results make sense, because those are also results that would be less likely under the null hypothesis.

If you calculated all possible randomizations, then you can use this formula for p-values:

$$p = \frac{\text{observations equal, or more extreme}}{\text{number of randomizations}}$$

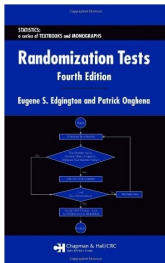
If you randomly sampled the randomizations, then the above will underestimate the true p-value (because your sampled distribution is missing some extreme values). You can correct for this by adding 1 to the numerator and denominator:

$$p = \frac{\text{observations equal, or more extreme} + 1}{\text{randomly sampled randomizations} + 1}$$

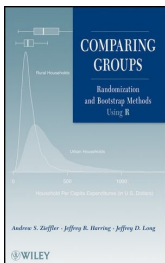
Randomization: More info

Randomization tests are incredibly powerful and incredibly flexible. I would say that if you want to do pure NHST, without mixed effects, then randomization tests should be your first choice.

Even Fisher admitted that randomization tests should be the gold standard for NHST. But in the 1930s, computers weren't accessible enough to make randomization tests feasible for anything but very small experiments. So he developed analytic methods for larger experiments. But he said that the analytic methods are only valid insofar as they give approximately the same result as randomization methods.



The best reference for randomization tests is Edgington and Onghena (now 2007), **Randomization Tests**. Be warned that it is written like a reference, and not like a textbook. But if you need to know something about randomization tests, it is fantastic.



For a textbook experience, I like Zieffler, Harring, and Long's (2011) **Comparing Groups: Randomization and Bootstrap Methods using R**. It is an introduction to NHST using Randomization and Bootstrap methods, which is a nice idea in the computer age.

Inferences: samples vs populations

If you want to make **causal inferences** about whether your treatment had an effect in your **sample**, you have to **randomly assign** units to treatments.

If you can't randomly assign your units to treatments, you can't be sure that your treatment is causing the effect. The effect could be caused by properties of the two groups.

As the name implies, randomization tests assume that you randomly assigned units to treatments. And because of this assumption, randomization tests allow you to make **causal inferences about the effect of your treatment in the sample**.

If you want to make **inferences about populations**, you have to **randomly sample the units from the population**.

If you can't randomly sample your units, you can't be sure that your results hold for the entire population. You can still make inferences about **your sample**, which is generally all you want to do anyway, but you can't claim that your treatment will have an effect in a population.

If you want to make **claims about a population**, then you can't use randomization tests. You have to add the idea of sampling from a population to the test. What you want are **bootstrap methods**.

Bootstrap methods

If you are running a bootstrap, I assume you are interested in making inferences about populations. Here are the first three steps:

- Step 1:** Randomly sample your participants (or other experimental units) when running your experiment. This is necessary if you want to make inferences about population parameters from the samples.
- Step 2:** Choose a test statistic. Usually this is the mean, but it could be one of the other possible statistics.
- Step 3:** Define the null hypothesis as no difference between population parameters, e.g., $\mu_A - \mu_B = 0$

Let's assume that we did randomly sample (not true, but let's assume it for demonstration purposes, and let's use means/mean differences for our test statistic.

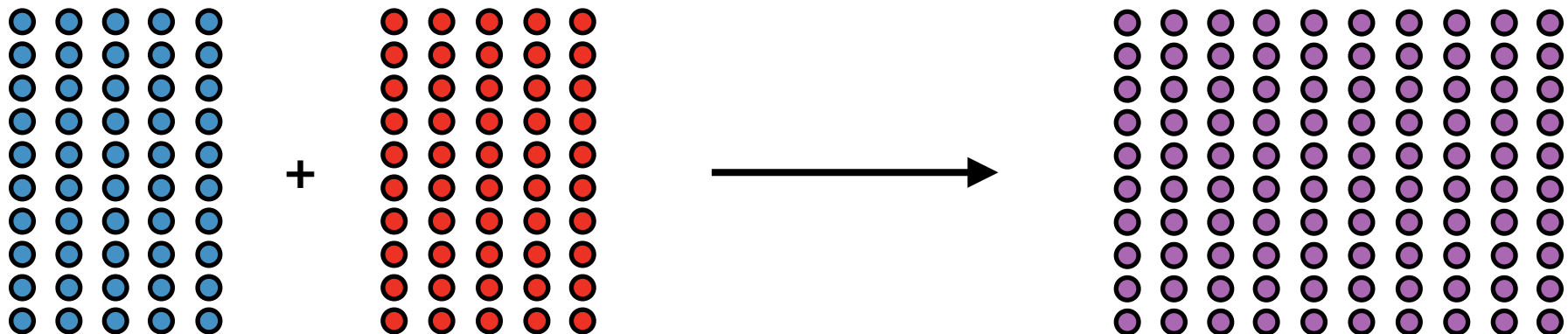
Bootstrap methods

Step 4: Define the single population under the null hypothesis.

This is our first tricky step. We don't have an empirical measurement of our population. If we did, we wouldn't need to sample from it! So what do we do?

Well, we can use our experimental sample as an approximation because it was randomly sampled.

Under the null hypothesis, we only have one population, so that means we can combine all of the values from both conditions together into one group:



Bootstrap methods

Step 5: Calculate the reference distribution for your test statistic under the null hypothesis (one population).

This is our next tricky step. We want to use our sample as an approximation of our distribution. This means randomly sampling from our sample in order to derive a reference distribution.

In this case we want to **randomly sample with replacement**, which means that after each participant is selected, we replace it so that it could be **selected again in the very same sample!** Up until now, we have been sampling **without replacement**, which means that each participant could only be **selected once per sample**.

with replacement				without replacement			
$\{1,2,3\}$, choose 2	\rightarrow	$\{1,2\}$	$\{1,1\}$	$\{1,2,3\}$, choose 2	\rightarrow	$\{1,2\}$	
		$\{1,3\}$	$\{2,2\}$			$\{1,3\}$	
		$\{2,3\}$	$\{3,3\}$			$\{2,3\}$	

We do this to approximate a population that is much larger than our sample (possibly infinitely large). Values will still be chosen according to their probability, but they won't artificially disappear because our sample size is small.

Bootstrap methods

Step 5: Calculate the reference distribution for your test statistic under the null hypothesis (one population).

So here is what we do:

First, we randomly sample with replacement two samples from our observed sample. We call these **bootstrap replicates**. They are replicates because they are other possible samples that we could have obtained in our experiment. They are bootstrap replicates because this procedure is called the bootstrap method.

Second, we calculate the mean for each bootstrap replicate, and then calculate the mean difference.

Third, we save this mean difference (as the first value in our reference distribution).

Then we repeat this process a large number of times (e.g., 10,000) to derive a reference distribution called the **bootstrap distribution**.

Bootstrap methods

Step 6: Calculate the probability of your observed statistic (e.g., difference between means in your experiment), or one more extreme, from your reference distribution.

Now that we have a reference distribution that approximates the exact distribution pretty well, all we need to do is use our old formula (plus correction) to calculate the p-value:

$$p = \frac{\text{outcomes equal, or more extreme} + 1}{\text{randomly sampled outcomes} + 1}$$

What we've just done is called a **non-parametric bootstrap**, because we didn't make any assumptions about the parameters of the population. Instead, we used our (combined) sample as a proxy for the population.

Parametric: A parametric test is one in which the parameters of the population are known (or assumed)

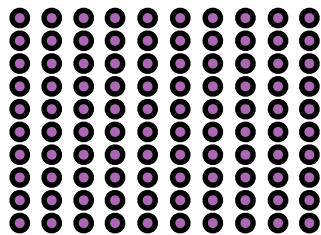
Non-parametric: A non-parametric test is one in which the parameters of the population are unknown (or not assumed)

A parametric bootstrap

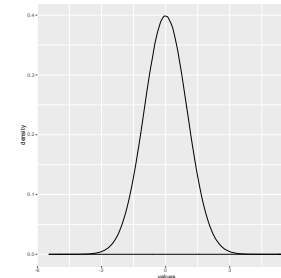
The parametric bootstrap has the same steps as the non-parametric bootstrap. The only difference is in the population that the replicates are drawn from!

1. Define a population to draw the replicates from:

non-parametric: the sample is used as a proxy



parametric: a probability model for the population with certain parameters



2. Sample with replacement from the population to derive a reference distribution.
3. Calculate the probability of the data given the reference set.

A parametric bootstrap

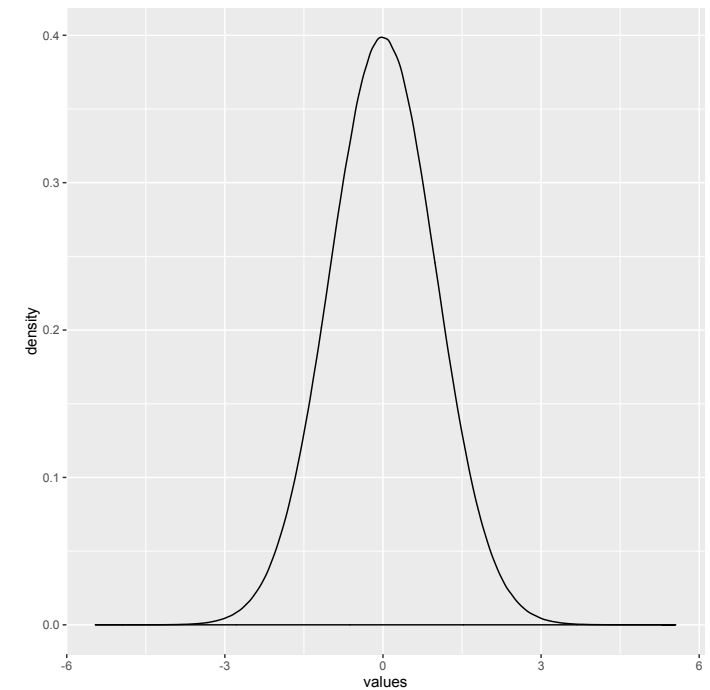
The only challenge in the parametric bootstrap is picking the correct probability model for your population. How do you know what parameters to pick?

In principle, you could pick any probability model that you think underlies the generation of your data. In practice, if you are ever doing one of these analyses, you will probably choose a **normal distribution**.

The normal distribution is the “bell curve”. It has some useful properties that make it a good choice for many applications:

1. It is the probability model underlying a large number of phenomena.
2. It can be completely parameterized with 2 parameters: the mean and the standard deviation using the following equation:

$$y = \frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma}$$

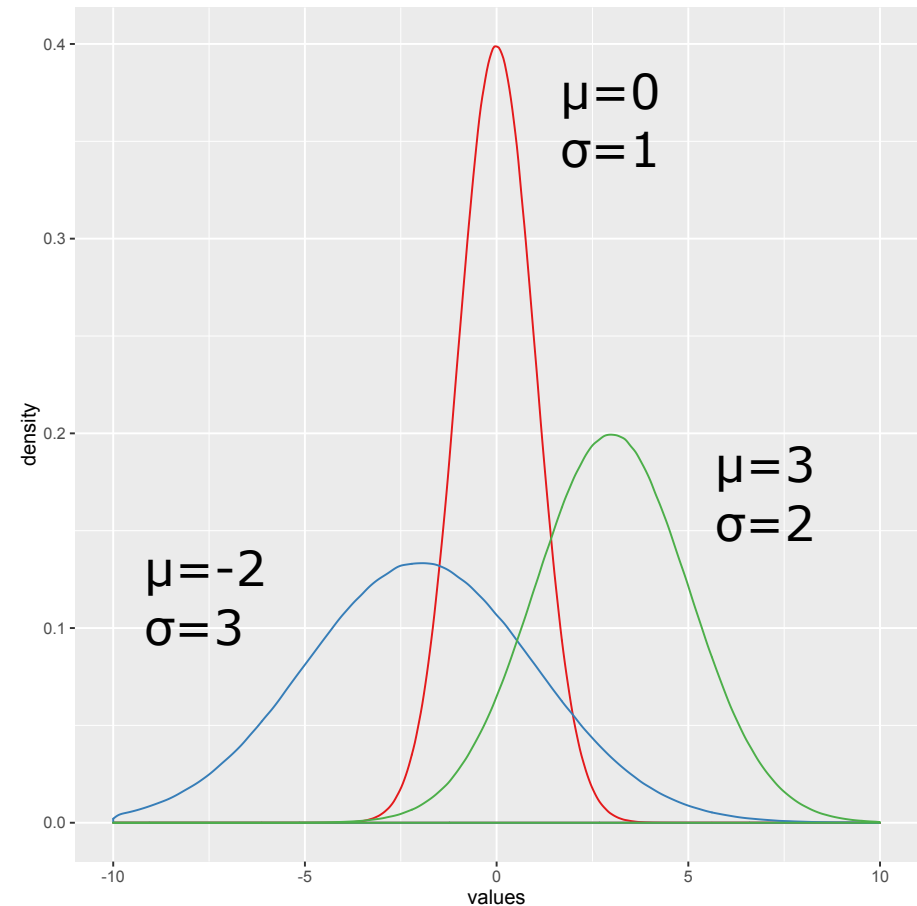


A parametric bootstrap

The only problem with the normal distribution is that it is a **family of distributions**. Every member of the family follows the equation, but they each use a different value for the mean and standard deviation:

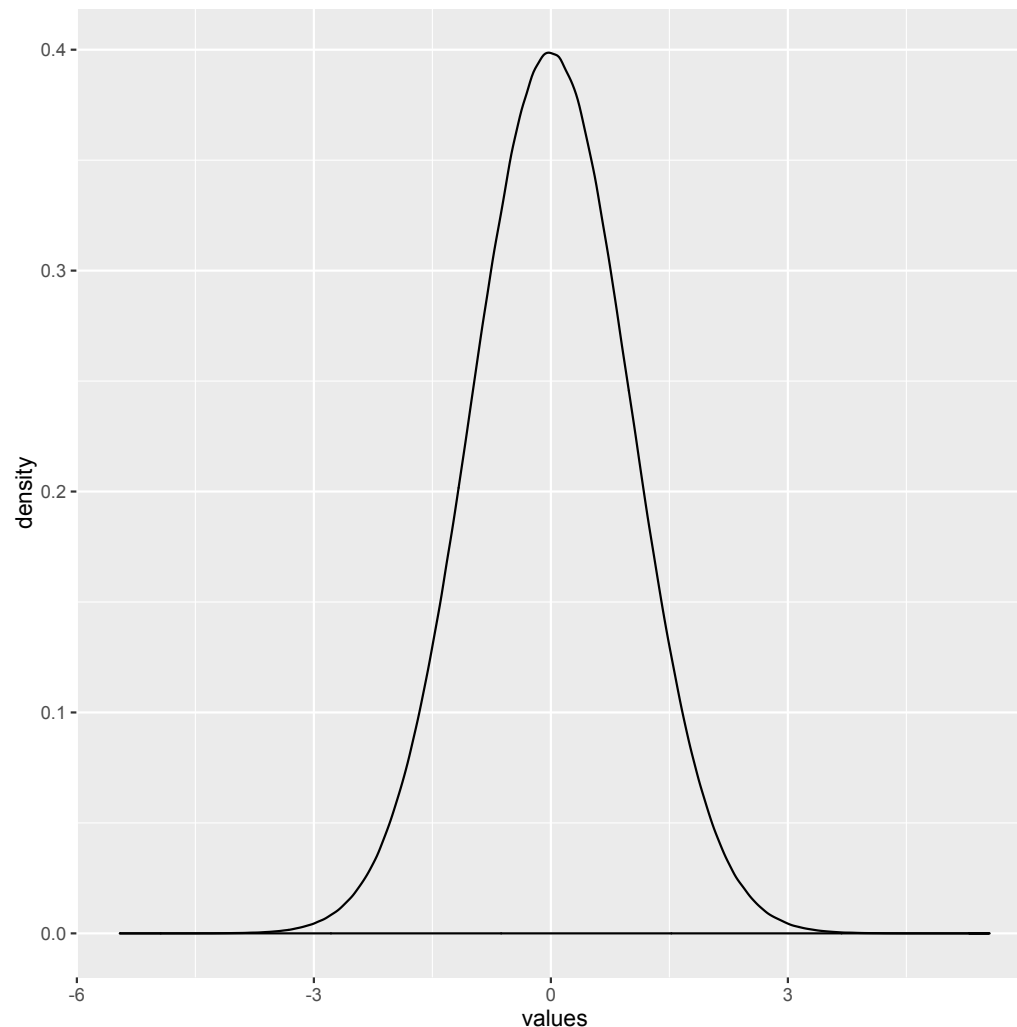
Believe it or not, these are all normal distributions.

The difference is that each one has a different mean (so a different location on the x-axis), and a different standard deviation (a different width on the x-axis, which also means a different height).



A parametric bootstrap

Instead of trying to guess the mean and standard deviation, you can use the **standard normal distribution**, which is just a normal distribution with a mean of 0, and a standard deviation of 1. It is easy to work with.



Converting our data to the standard normal distribution: z-scores appear again!

It is easy enough to use the standard normal distribution as our probability model for the population. R even gives us the built-in function `rnorm()`, which randomly samples from the standard normal distribution by default.

The problem is that our observed values are not on the same scale (the mean of the combined group of both of our condition is not 0). So we won't be able to compare our observed values to the reference distribution.

This is actually easy to fix. We can simply convert the values in our combined group into the standard normal distribution scale using our old friend the **z-score transformation**.

In this case, we are applying the z-score transformation to our combined data set (the thing that represents the full population), not each participant. That's the only difference. It is the same equation:

$$Z = \frac{X - \text{mean}}{\text{standard deviation}}$$

The result is that each value will be equal to its distance from the mean (as if the mean were 0), and that distance will be measured in units equal to the standard deviation. So our observed values will be on the same scale as our reference distribution!

A parametric bootstrap

Now that we've decided on a probability function, and re-scaled our observed data, we simply carry out the bootstrap procedure like before:

First, we randomly sample with replacement two samples from our probability model. We call these **bootstrap replicates**. They are replicates because they are other possible samples that we could have obtained in our experiment. They are bootstrap replicates because this procedure is called the bootstrap method.

Second, we calculate the mean for each bootstrap replicate, and then calculate the mean difference.

Third, we save this mean difference (as the first value in our reference distribution).

Then we repeat this process a large number of times (e.g., 10,000) to derive a reference distribution called the **bootstrap distribution**.

Finally we calculate a p-value using the standard formula (and correction).

The script **bootstrap.r** contains code to run both a non-parametric and parametric bootstrap.

Analytic methods

Because randomization and bootstrap methods are so computationally intensive, early 20th century statisticians could not use them. These people were smart. They developed analytic methods that give approximately the same result as randomization and bootstrap methods. And then shared them with the world.

The basic idea of analytic methods is that we need test statistics that have known, or easily calculable, reference distributions. We can't use the mean, because the distribution of the mean will vary based on the experiment (the data type, the design, etc). We need statistics that are relatively invariant, so that we can calculate the distribution once, and use it for every experiment in all of the different areas of science.

There are both **parametric** and **non-parametric** analytic methods, just like there are both parametric and non-parametric bootstrap methods. And there are a ton of different test statistics with different properties that are suited for different experimental situations.

For pedagogical reasons, I am going to focus on the **F statistic**.

The F statistic is parametric

When people talk about parametric statistics, there is a typical cluster of three assumptions that they usually have in mind. The F statistic is parametric in this way - its distribution is predictable only if these assumptions are met:

Normally distributed errors:

The error terms in the linear model are normally distributed (which will be true if the population(s) of participants are normally distributed)

Independence:

The observed responses are independent (in repeated-measures designs this means the pairs of responses are independent)

Homogeneity of variance:

The variances of the samples are equal (homogeneous). This is always true when the null hypothesis is true, but also must be true when the null hypothesis is false.

There is a fourth assumption that typically accompanies these four under the rubric “parametric”, but it is not about the distribution of the statistic. It is about the inferences that can be drawn from it.

Random Sampling:

Participants are randomly sampled from a population

The F-distribution

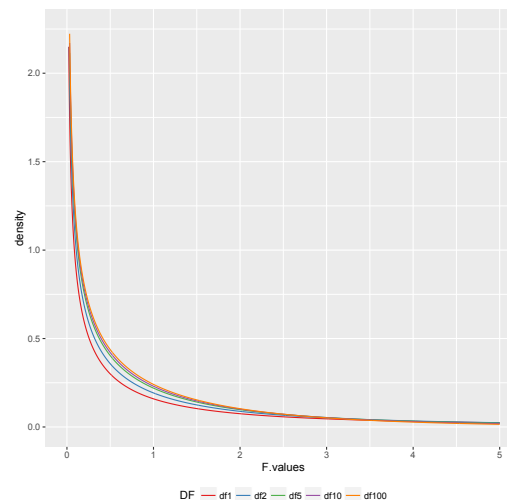
The distribution of the F statistic (called the Fisher-Snedecor distribution), is useful for analytic methods because it does not vary based on things like the mean or scale of the data. Instead, it is completely determined by two numbers, typically called df_1 and df_2 , or df_{num} and df_{den} , because of their relationships to the degrees of freedom in our calculation of F.

$$F = \frac{(SS_{simple} - SS_{complex}) / (df_{simple} - df_{complex})}{SS_{complex} / df_{complex}}$$

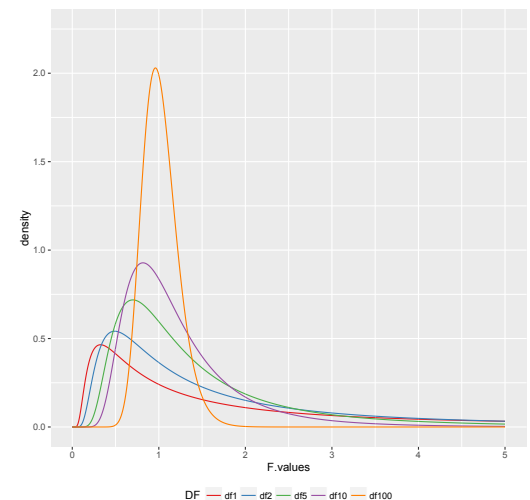
$$df_{num} = df_{simple} - df_{complex}$$

$$df_{den} = df_{complex}$$

If you want the equation for the probability density function, you can see it on the wikipedia page for the F distribution: <https://en.wikipedia.org/wiki/F-distribution>. It is fairly complicated, so I won't reproduce it here. But I will show you how the distribution varies with different dfs. In a 2x2, our df will be 1, as in the left figure. I include the right just to show you the full range of the F distribution. These plots are in f.distribution.r.



$$df_{num} = 1$$



$$df_{num} = 100$$

$$df_{den} = 1, 2, 10, 100$$

The ANOVA approach to F

The term ANOVA is just another way of saying F-test. It is actually the primary way, because most people think about tests, not about the statistics that they are using in that test.

ANOVA stands for **AN**alysis Of **V**ariance. What you should be thinking at this point is that we have never once discussed analyzing variance, so how is it that the F-tests that we have been discussing are analyses of variance?

Well, it turns out that there is a completely different, but equally valid, way of thinking about the F-ratio. Instead of a measure of error minimization per degrees of freedom, you can think of it as a ratio between two estimates of the population variance: the numerator is an estimate based on the sample means, and the denominator is an estimate based on the sample variance. (Don't worry, this will make more sense soon!)

$$F = \frac{\text{estimated } \sigma^2, \text{ based on sample means}}{\text{estimated } \sigma^2, \text{ based on the two sample variances}}$$

This is mathematically equivalent to the model comparison approach that I taught you, but conceptually different. I prefer model comparison; but most stats courses prefer the analysis of variance method. So now I will connect them for you!

Analysis of Variance

The first thing to realize about what we've been doing so far is that we've seen two ways to use samples to estimate the variance of a population.

Option 1: Use the variance of the sample as an estimate

Recall from our first lecture that the variance of a sample (s^2) can be used as an unbiased estimate of the population variance (σ^2) if we use $(n-1)$ in the calculation:

$$s^2 = \frac{\sum(Y_i - \bar{Y})^2}{(n-1)} = \text{estimate of } \sigma^2$$

In the case of an independent measures ANOVA, you actually have **two samples**! So you can come up with an even better estimate of σ^2 by **averaging** the two estimates! (If one estimate is good, the average of two estimates will be better!) Here is a formula to let you do that for two samples:

$$\text{mean } s^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{(n_1-1) + (n_2-1)}$$

Analysis of Variance

The first thing to realize about what we've been doing so far is that we've seen two ways to use samples to estimate the variance of a population.

Option 2: Use the variance of two (or more) means

Now, this estimate you probably didn't even notice. The basic idea has two steps.

First, the variance of two (or more) means provides an estimate of the variance of the sampling distribution of means (the variability in all of the means that you could get if you repeatedly sampled from a population: $\sigma_{\bar{Y}}^2$).

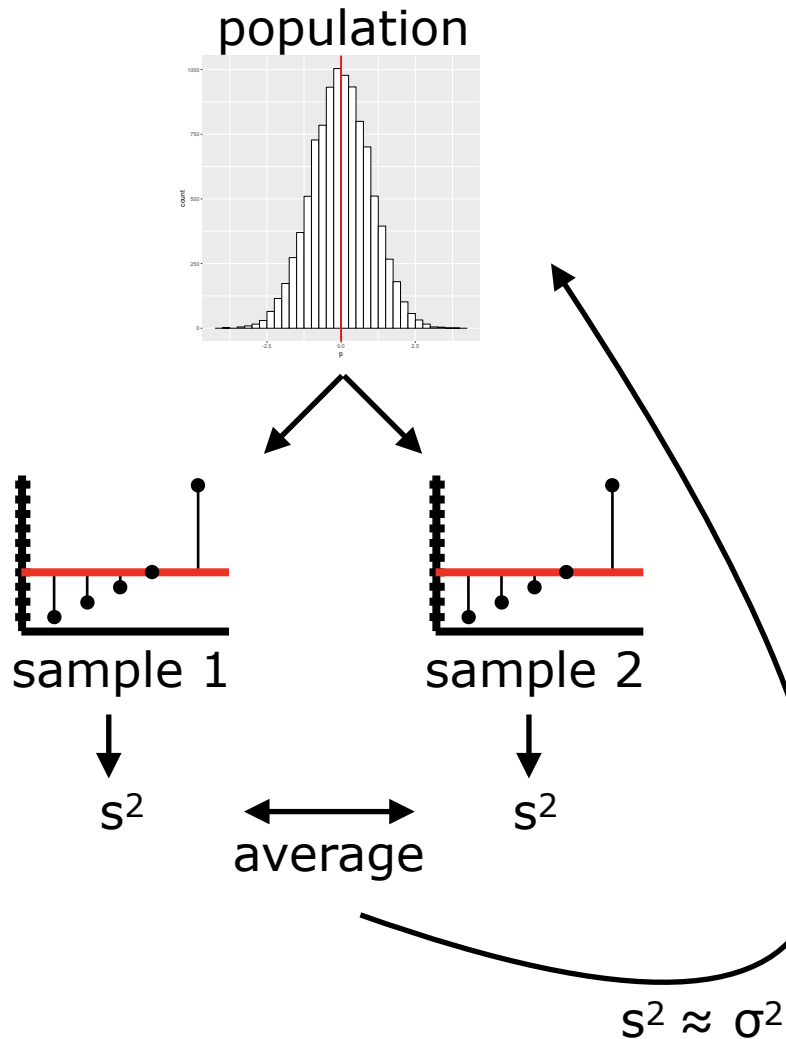
$$\text{estimate of } \sigma_{\bar{Y}}^2 = \frac{\sum(\bar{Y}_j - \bar{\bar{Y}})^2}{(j-1)}$$

Second, the variance of sampling means ($\sigma_{\bar{Y}}^2$) can be used to calculate the population mean:

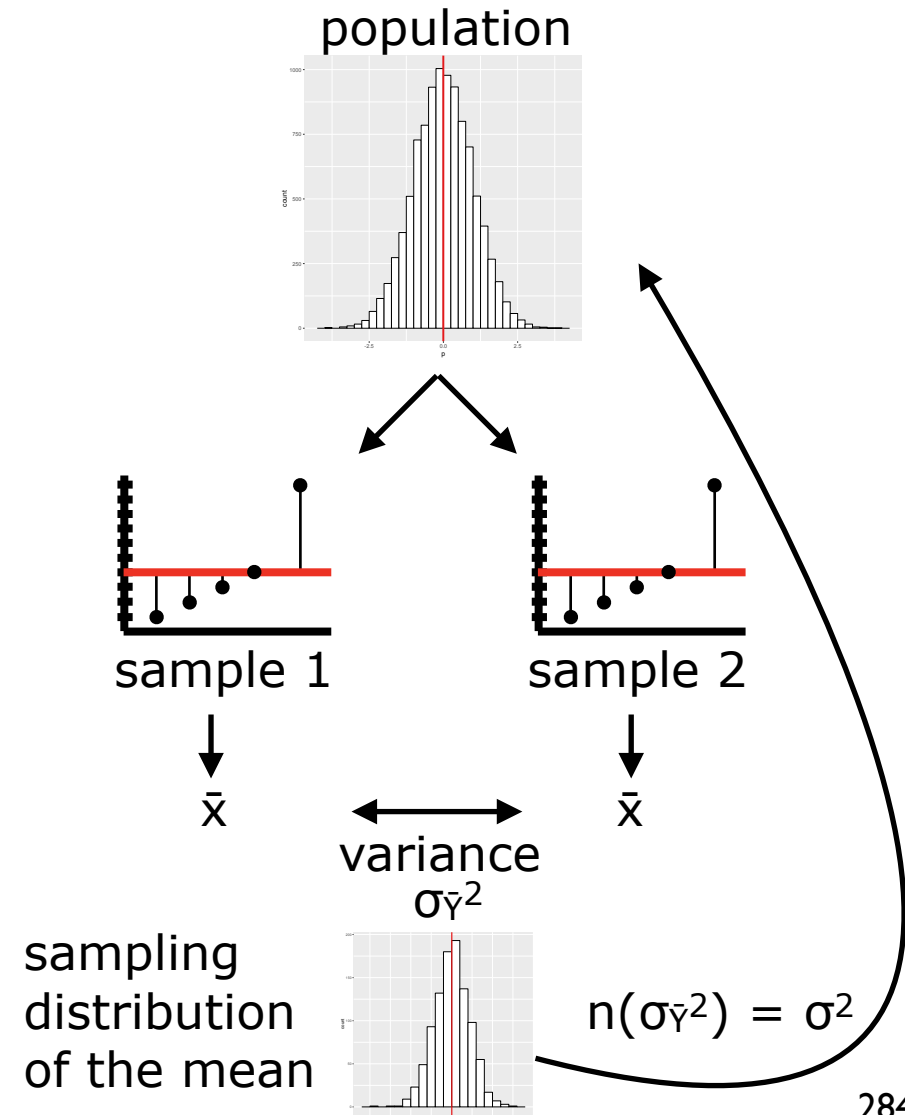
$$\sigma_{\bar{Y}}^2 = \frac{\sigma^2}{n} \quad \text{therefore,} \quad \sigma^2 = n(\sigma_{\bar{Y}}^2)$$

Two estimates of population variance (σ^2)

Based on sample variance
aka denominator
aka within groups



Based on sample means
aka numerator
aka between groups



Comparing the two estimates

We call the estimated variance based on the sample variances (Option 1) the **Within Groups Mean Squared Error, or MS_W** .

The reason we call it this is because “mean squared error” is just another way to say variance; and it was an estimate that was calculated by averaging the variance of the two groups (within the groups).

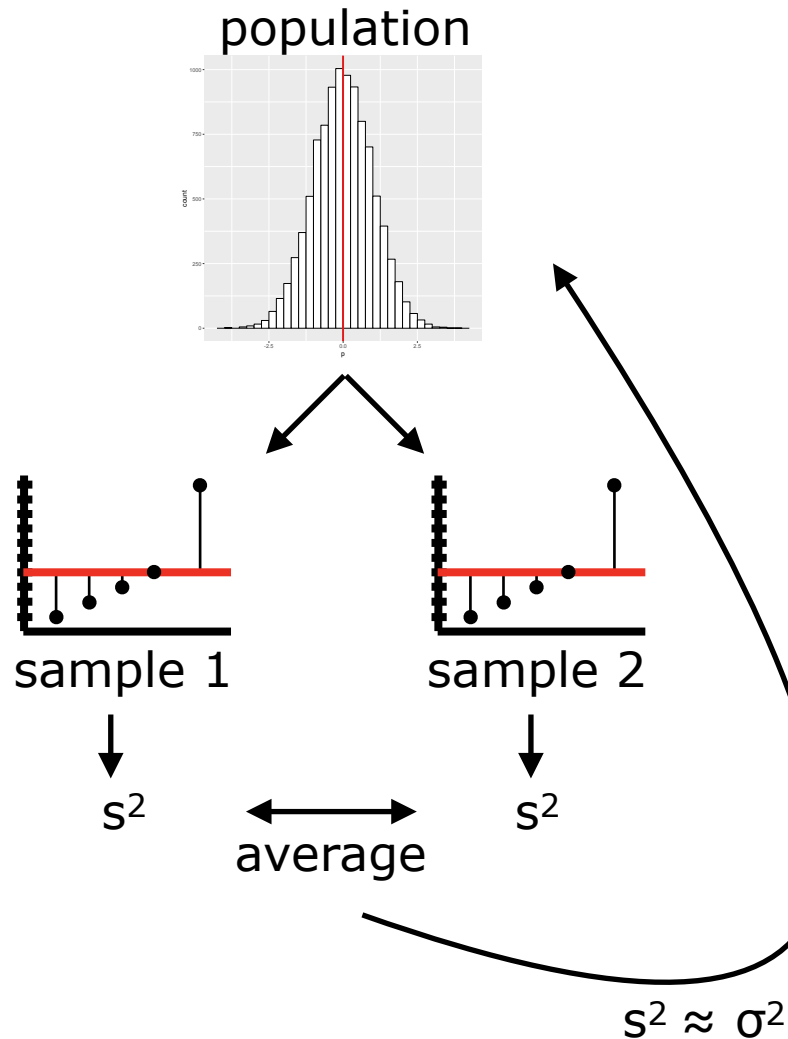
Assuming that variances are equal in both groups regardless of the hypothesis (null or alternative), which is an important assumption of ANOVAs, the MS_W **will not change based on whether the null hypothesis is true or false!**

We call the estimated variance based on the sample means (Option 2) the **Between Groups Mean Squared Error, or MS_B** . This is because it used the variance between the means of the two groups to estimate the variance (mean squared error) of the population.

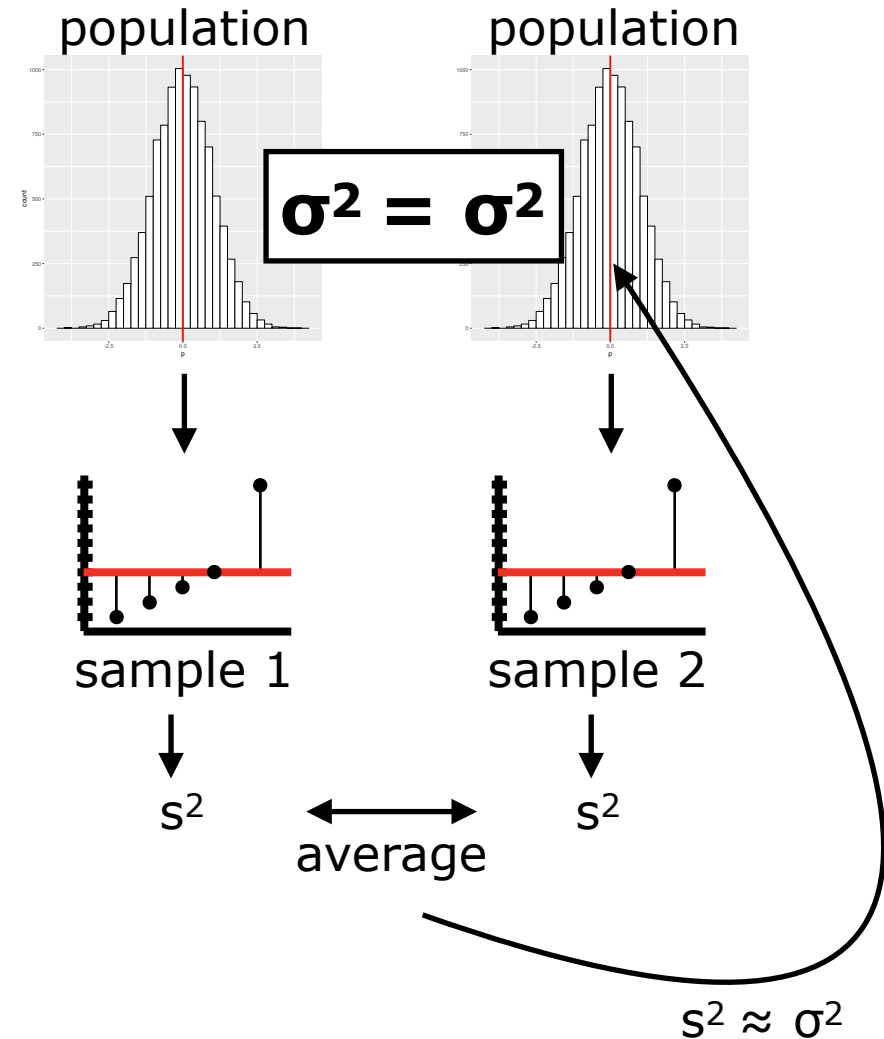
Now here is the neat thing. The MS_B will absolutely change depending on whether the null hypothesis is true or false. If the **null hypothesis is true**, then this estimate will be approximately the same as MSE_{WG} . But if the **null hypothesis is false**, this estimate will be **larger**. This is because the two means don't come from the same population, so they will likely be more different than two means that come from the same population.

Within groups variance does not change based on the hypothesis

Within Groups
Null Hypothesis is True

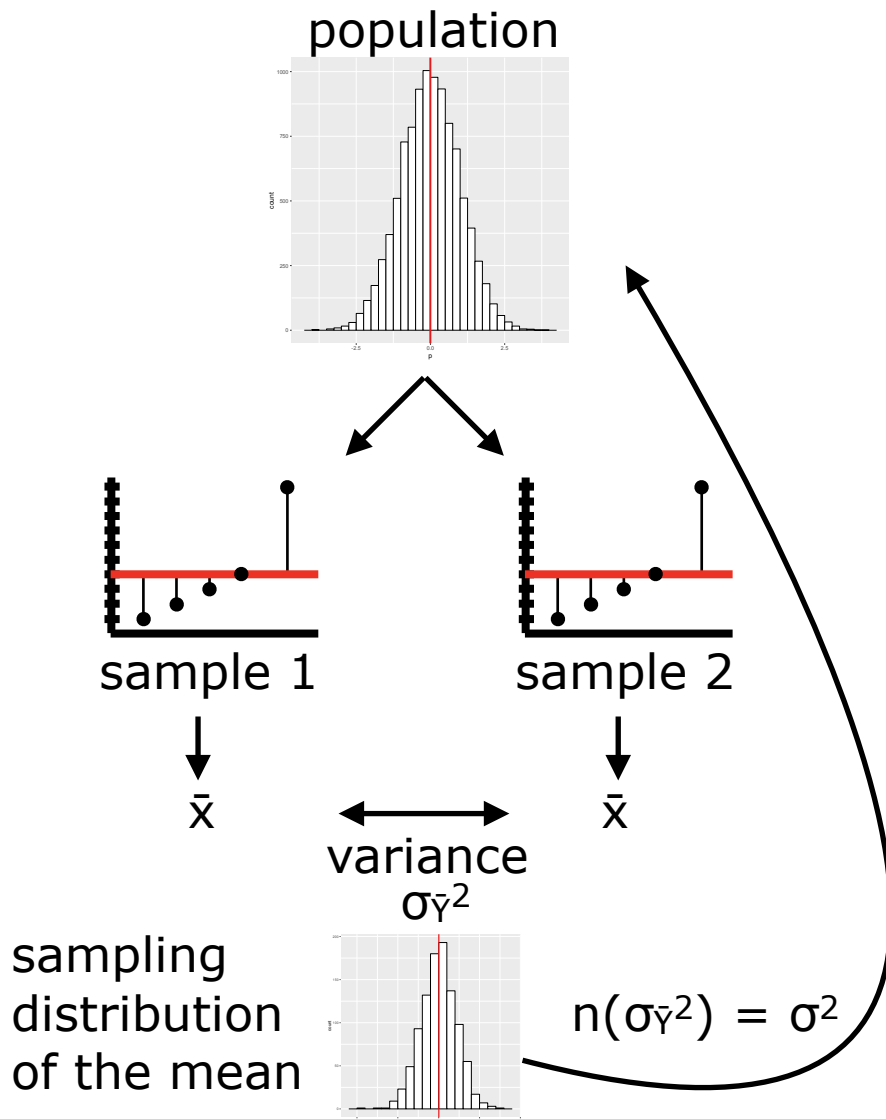


Within Groups
Null Hypothesis is False

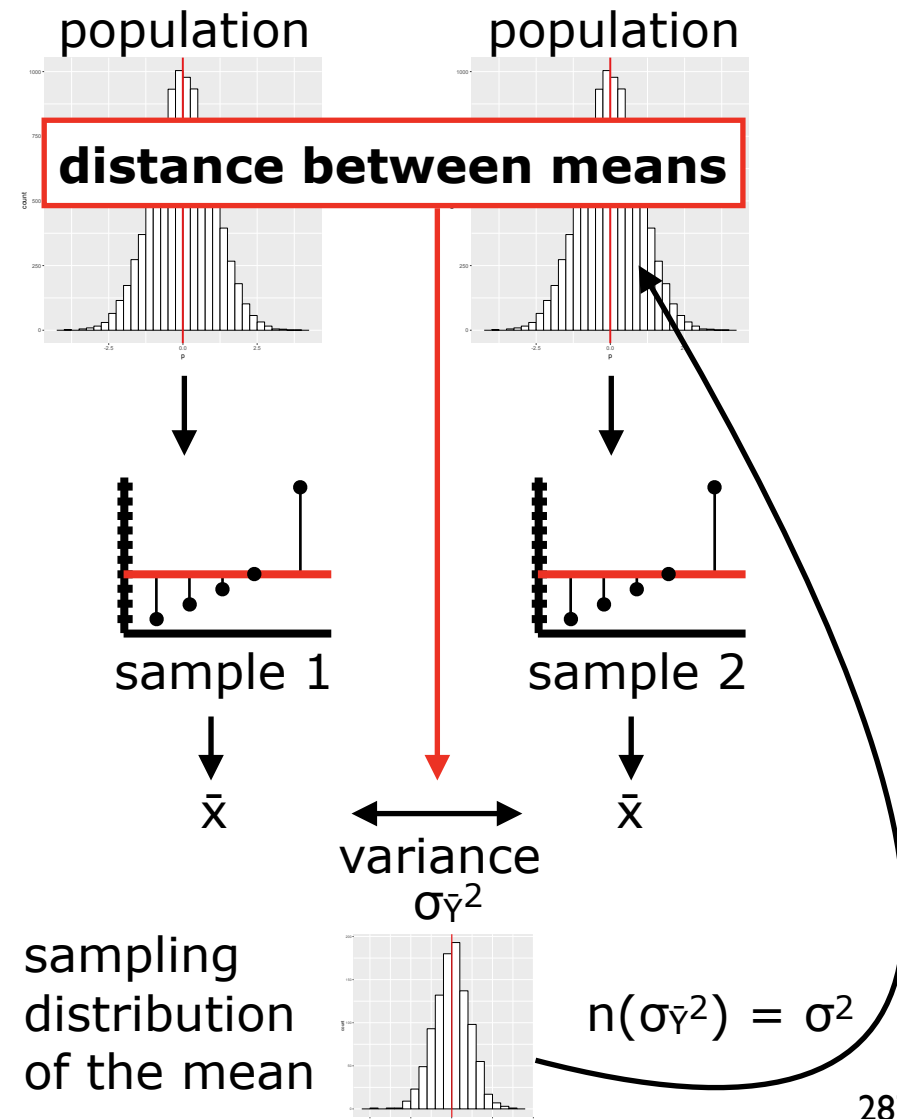


Between groups variance does change based on the hypothesis

Between Groups
Null Hypothesis is True



Between Groups
Null Hypothesis is False



This is also the F-ratio!

And now, yet another mind blowing moment:

$$F = \frac{MS_B}{MS_W}$$

Since MS_B gets larger when the null hypothesis is false, F will be larger (and will be close to 1 when the null is true).

Yup, the ratio between the estimate of the population variance based on mean variation and the estimate of the population variance based on sample variances is identical to the F-ratio that we've been talking about!

$$F = \frac{(SS_{\text{simple}} - SS_{\text{complex}})/(df_{\text{simple}} - df_{\text{complex}})}{SS_{\text{complex}}/df_{\text{complex}}}$$

We call the way we've been talking about the F-ratio the **model comparison approach**, because it emphasizes the comparison of two models. We call the new approach the **analysis of variance approach**, hence ANOVA. They are mathematically equivalent (I will leave it to you to work out the math), and they are equally valid for defining the F statistic for a test. Although I prefer using the model comparison approach, both are equally valid ways of thinking about F-tests.

And just FYI, $F = t^2$

We haven't looked at t-tests at all in this class, but some of you may have heard of them. A t-test is a way of comparing one mean to 0, or two means to each other, using the t-statistic. What you may find interesting is that F and t are related. F is t^2 .

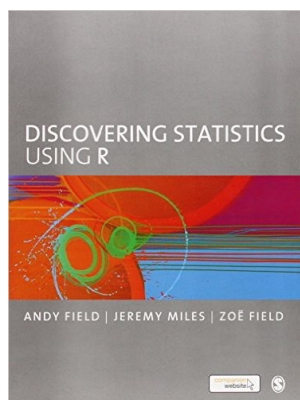
We can see this easily with our toy example from earlier. Let's calculate both an F for these two models, and a t for the complex model versus the constant in the simple model.

simple			complex		
Y_i	$= \beta_0 X_{0-i}$	$+ \epsilon_i$	Y_i	$= \beta_0 X_{0-i}$	$+ \epsilon_i$
2	= 4	+ -2	2	= 3	+ -1
3	= 4	+ -1	3	= 3	+ 0
4	= 4	+ 0	4	= 3	+ 1
df=3			df=2		
SS=5			SS=2		

$$F = \frac{(SS_{\text{simple}} - SS_{\text{complex}})/(df_{\text{simple}} - df_{\text{complex}})}{SS_{\text{complex}}/df_{\text{complex}}} = \frac{(5-2)/(3-2)}{2/2} = 3$$

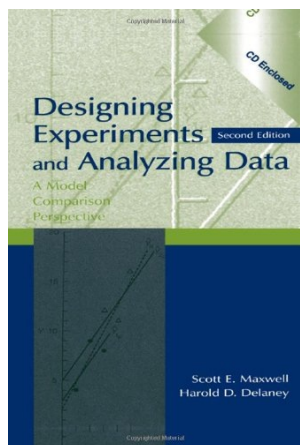
$$t = \frac{\bar{Y} - \mu}{\sqrt{\left(\frac{s^2}{n}\right)}} = \frac{3 - 4}{\sqrt{\left(\frac{1}{3}\right)}} = -1.732051$$

Analytic methods: more information



Discovering Statistics Using R Field, Miles, and Field

This book is a comprehensive introduction to (analytic) statistics, and it is a great introduction to R (and plotting with R). It is very readable (and at times, amusing), and covers all of the things that are covered in fundamental statistics courses.



Designing Experiments and Analyzing Data Maxwell and Delaney

This is probably the best book on the model comparison approach to F-tests there is. It is also a beast of a book. But well worth it if you really want to understand F-tests. There is no R here. This is math.

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1: Design

Section 2: Analysis

Section 3: Application

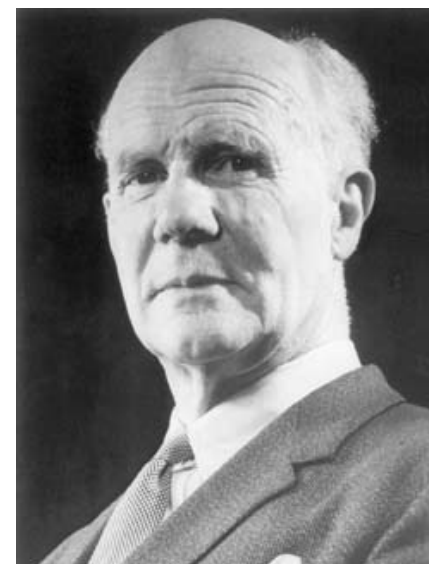
Going further: Neyman-Pearson NHST

Neyman and Pearson were fans of Fisher's work, but they thought there was a logical problem with his approach.

While it is all well and good to say that the p-value is a measure of strength of evidence against the null hypothesis, at some point you have to **make a decision to reject the null hypothesis, or not.**



Jerzy Neyman
(1894-1981)



Egon Pearson
(1895-1980)

Fisher himself had suggested that $p < .05$ was a good criterion for deciding whether to reject the null hypothesis or not.

Neyman and Pearson decided to take this one step further, and really work out what it would mean to base a statistical theory on the idea of decisions to reject the null hypothesis.

Going further: Neyman-Pearson NHST

Tenet 1: There are two states of the world: the null hypothesis is either true or false.

Tenet 2: You can never know if the null hypothesis is true or false.

This actually follows from the philosophy of science and the problem of induction.

In the absence of certainty about the state of the world, all you can do is make a decision about how to proceed based on the results of your experiment. You can choose to reject the null hypothesis, or you can choose not to reject the null hypothesis.

This sets up four possibilities: two states of the world and two decisions that you could make.

		State of the World	
		H_0 True	H_0 False
Decision	Reject H_0	Type I Error	Correct Action
	Accept H_0	Correct Action	Type II Error

Going further: Neyman-Pearson NHST

		State of the World	
		H_0 True	H_0 False
Decision	Reject H_0	Type I Error	Correct Action
	Accept H_0	Correct Action	Type II Error

Type I Error: This is when the null hypothesis is true, but you mistakenly reject it.

Type II Error: This is when the null hypothesis is false, but you mistakenly fail to reject it.

Take a moment to really think about what these two errors are. What do you think about the relative importance of each one?

Going further: Neyman-Pearson NHST

		State of the World	
		H_0 True	H_0 False
Decision	Reject H_0	Type I Error	Correct Action
	Accept H_0	Correct Action	Type II Error

Neyman-Pearson, and many others, have suggested that Type I errors are more damaging than Type II errors.

The basic idea is that science is focused on rejecting the null hypothesis, not accepting it. (To publish a paper, you have to reject the null hypothesis.) So a Type I error would mean making a decision (or publishing a result) that misleads science.

Type II errors are also important, but not equally so. Failing to reject the null hypothesis is simply a failure to advance science. It doesn't (necessarily) mislead the way that a Type I error does.

Going further: Neyman-Pearson NHST

Type I Error: This is when the null hypothesis is true, but you mistakenly reject it.

If you accept the importance of Type I errors, then you will want to keep the rate of Type I errors as low as possible.

Under the Neyman-Pearson approach, which emphasizes the decision aspect of science, you can control your Type I error rate by **always using the same criterion for your decisions**.

alpha level / alpha criterion: This is the criterion that you use to make your decision. By keeping it constant, you keep the number of Type I errors that you will make constant too. For example, if you set your alpha level to .05, then you only decide to reject the null hypothesis if your p-value is less than .05. Similarly, if you set your alpha level to .01, then you only decide to reject the null hypothesis if your p-value is less than .01.

Take a moment to think about how setting an alpha level will control your Type I error rate.

Going further: Neyman-Pearson NHST

There is an important relationship between your alpha level and the number of Type I errors that you will make:

If you apply the same alpha level consistently over the long-run, your Type I error rate will be less than or equal to your alpha level.

Here's a thought experiment:

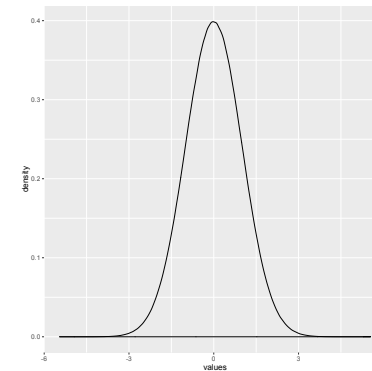
1. Imagine that the null hypothesis is TRUE.
2. Now, imagine that you run an experiment and derive a test statistic.
3. Next, imagine that you run a second experiment and derive a test statistic.
4. And then, imagine that you ran the experiment 10,000 times...
5. This should be familiar. You just derived a reference distribution of the test statistic under the null hypothesis!
6. Now set your alpha level (decision criterion) to .05. Given the distribution you just derived, how often will you derive a p-value less than .05? In short, how often would you make a Type I Error?

We can run this in R. There is code for it in [alpha.demonstration.r.](https://alpha.demonstration.r/)

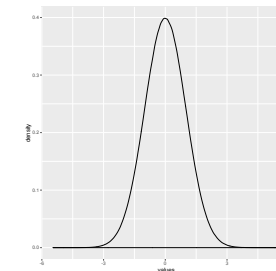
Graphical version: the alpha level

Here is how the alpha level works:

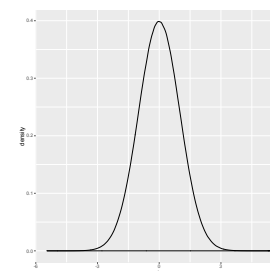
- 1.** Imagine that the null hypothesis is true for your phenomenon.
- 2.** And let's run an experiment testing this difference 10,000 times, saving the statistic each time.
- 3.** The result will be a distribution of real-world test statistics, obtained from experiments where the null hypothesis is true.
- 4.** But also notice that this distribution will be nearly identical to the hypothetical null distribution for your test statistic (because the null hypothesis was true in the real world). This will be important later.



real world
distribution of stats

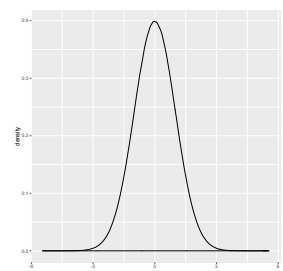


real world
distribution



null distribution

=

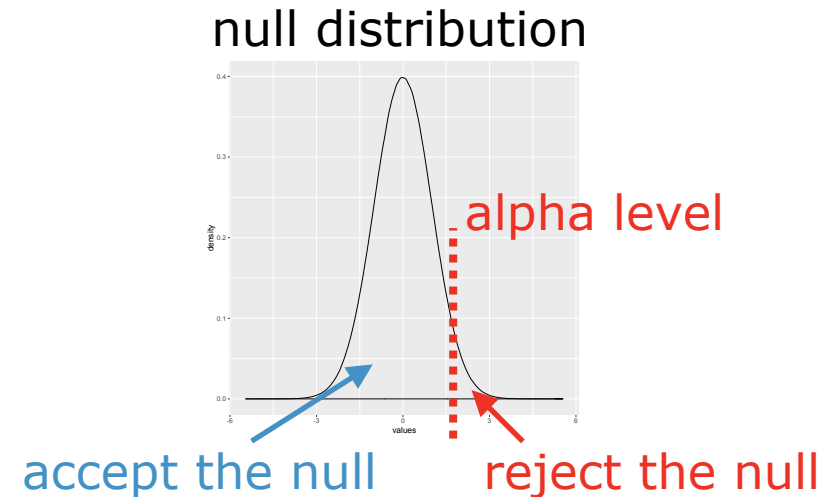


Graphical version: the alpha level

5. Now let's choose a **threshold** to cut the distribution into two decisions: **non-significant** and **significant**

Remember we call this the **alpha level**.

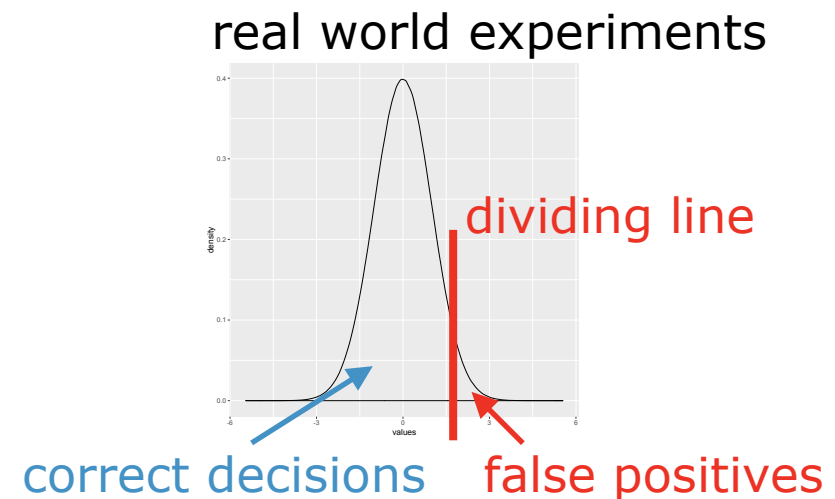
Also remember that this is a criterion chosen based on the null distribution (because this is a null hypothesis test).



6. Now we apply this threshold to each of our 10,000 experiments, one at a time as we run them.

So for each experiment, we can label it as a **correct decision** (accept the null) or a **false positive** (reject the null).

And to make life easier, we can visualize this as a distribution of results, with a dividing line between the two types.

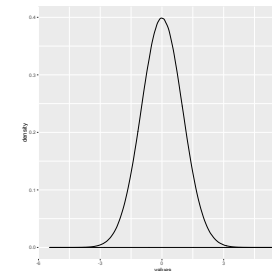


Graphical version: the alpha level

7. Now here is the final question. How many false positives happened in our 10,000 experiments?

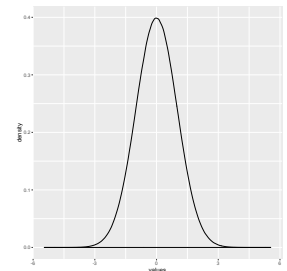
We could count them. But what I want to show you is the consequence of the identity that happened back in step 4.

real world distribution



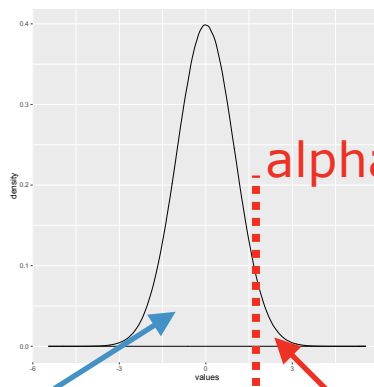
null distribution

=



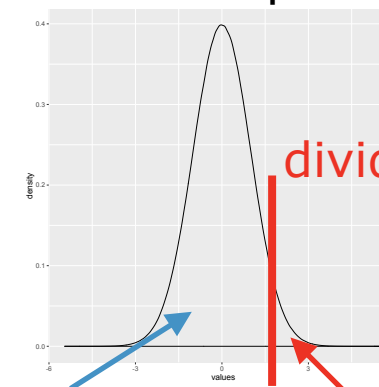
Because our real world distribution is identical to the null distribution (the null hypothesis is true), our alpha level is identical to the dividing line between correct decisions and false positives:

null distribution



=

real world experiments



accept the null

reject the null

correct decisions

false positives

In this way, the alpha level is the **maximum type I error rate** (because the maximum number of errors occurs when the null is true).

Going further: Neyman-Pearson NHST

It is important to understand the relationship between these concepts:

- p-value:** The probability of obtaining a test statistic equal to, or more extreme than, the one you observed under the null hypothesis.
- α -level:** The threshold below which you decide to reject the null hypothesis
- Type I Error:** This is when the null hypothesis is true, but you mistakenly reject it.

If you consistently base your decisions on the alpha level, then your Type I error rate will either be less than or equal to your alpha level!




We say that it might be less because we admit that the null hypothesis might be false for some experiments. Every time the null hypothesis is false, you make one less Type I Error, so the rate goes down a bit!

Multiple comparisons

Multiple comparisons

When people say “multiple comparisons”, what they mean is running more than one statistical test on a single set of experimental data.

The simplest design where this will arise is a one-factor design with three levels. Maybe something like this:

-  What do you think that John bought?
-  What do you wonder whether John bought?
-  What do you wonder who bought?

An F-test (ANOVA) or linear mixed effects model on this design will ask the following question:

What is the probability of the data under the assumption that the three means are equal?

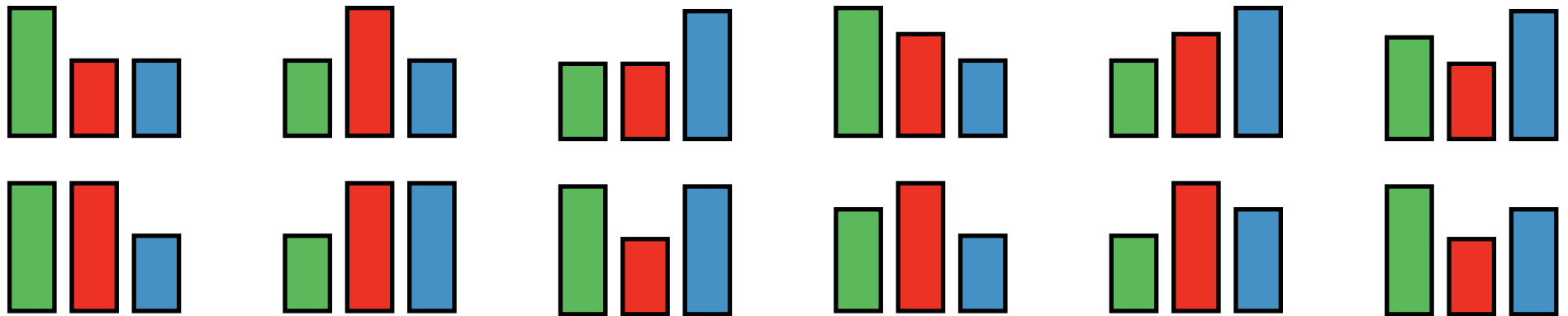
null hypothesis



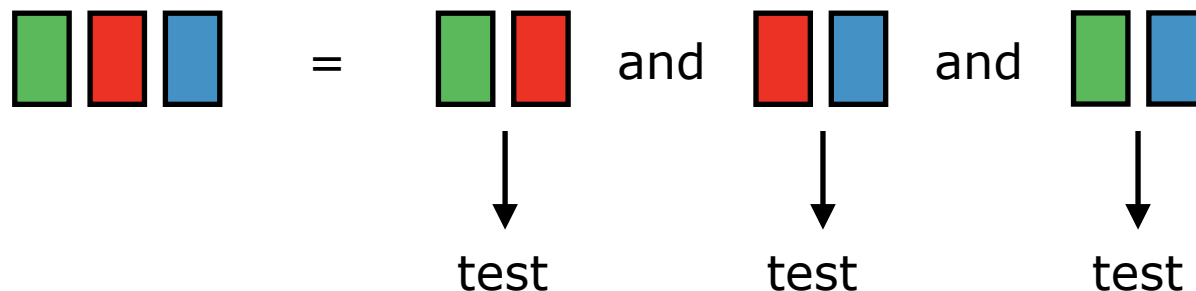
How many patterns of results will yield a low p-value under this null hypothesis?

A significant result tells us relatively little

Here are all (I think?) of the patterns of results that will yield a significant result in a one-way / three-level test. As you can see, a significant result doesn't tell us very much.



If we want to know which of these patterns is the one in our data, we need to compare each level to every other level one pair at a time:



The multiple comparison problem

Review: Neyman-Pearson NHST

		State of the World	
		H_0 True	H_0 False
Decision	Reject H_0	Type I Error	Correct Action
	Accept H_0	Correct Action	Type II Error

Type I Error: This is when the null hypothesis is true, but you mistakenly reject it.

Type II Error: This is when the null hypothesis is false, but you mistakenly fail to reject it.

α -level: The threshold at which you decide to reject the null hypothesis.

There are different error rates

Per Comparison Error Rate:

The probability that any one comparison is a Type I error (number of errors/number of comparisons). You set this by choosing a threshold for your decisions. We call the threshold α . Let's call the error rate PCER.

$$\text{PCER} = \frac{\text{number of errors}}{\text{number of statistical tests}}$$

Experimentwise Error Rate:

The probability that an experiment contains at least one Type I error. We can call this rate EWER.

$$\text{EWER} = \frac{\text{number of experiments with 1 or more errors}}{\text{number of experiments}}$$

Familywise Error Rate:

This is just like experimentwise error, but allows you to define sub-groups of comparisons inside of an experiment called a "family". So this is the probability that a family contains at least one error. In most experiments, there is just one family, so this will be equal to the experimentwise error rate. Let's call it FWER.

Visualizing the different error rates

Imagine your experiment has 3 comparisons, and you run that experiment 20 times. Let's say you set α to .05. Here are your results:

	e1	e2	e3	...															e20
comp1																			
comp2																			
comp3																			

$$\text{PCER} = \frac{\text{number of errors}}{\text{number of statistical tests}} = \frac{3}{60} = .05$$

$$\text{EWER} = \frac{\text{number of experiments w/errors}}{\text{number of experiments}} = \frac{3}{20} = .15$$

When you make multiple comparisons, EWER is larger than PCER. This is the **multiple comparisons problem!**

An equation for relating EWER to α

The relationship between α and EWER is lawful, and follows this equation:

	e1	e2	e3	...														e20
comp1																		
comp2																		
comp3																		

$$\text{EWER} = 1 - (1 - \alpha)^C \quad \text{where } C \text{ is the number of comparisons.}$$

So for 3 comparisons and an α set to .05, the maximum EWER will be:

$$\text{EWER} = 1 - (1 - .05)^3 = .142625$$

There is code in [multiple.comparisons.r](#) to demonstrate EWER, and how the EWER will always be more than PCER.

The take-home message is that multiple comparisons increases your type I error rate for the entire experiment!

Controlling Experimentwise Error

The Dunn Correction

(Mistakenly called the “Bonferroni correction” in the literature)

Controlling EW/FW error

So now you can see that setting an alpha level of .05 for each comparison only controls error at the comparison level. If you want to control errors at the experiment (or family) level, you need to make an adjustment to your decision criterion.

Luckily, the equation for EW/FW error tells us exactly how to do that:

$$\text{EWER} = 1 - (1 - \alpha)^C$$

Since EW/FW error is dependent on α , all we have to do is choose an α that gives us the EWER that we want!



You could do this through guessing-and-testing if you want, but statistician Olive Dunn figured out a much faster way using one of mathematician Carlo Bonferroni's inequalities:

$$X \geq 1 - (1 - (X/C))^C$$

As you can see, this inequality looks very similar to the EWER equation above...

The Dunn Correction (“Bonferroni correction”)

Here is how you can use Bonferroni’s inequality to set your α , and control EWER:

First, replace the X with EWER because that is what we care about. (And C is the number of comparisons).

Next, notice that the term EWER/C is in the position that α occupies in the EWER equation.

From that, it follows that if we set α to EWER/C we can keep our EWER at or below the number we want!

$$\begin{array}{ccc} X \geq 1 - (1 - (X/C))^C & & \\ \downarrow & & \downarrow \\ \text{EWER} \geq 1 - (1 - (\text{EWER}/C))^C & & \\ & & \swarrow \\ \text{EWER} = 1 - (1 - \alpha)^C & & \end{array}$$

$$\alpha = \frac{\text{EWER}}{C}$$

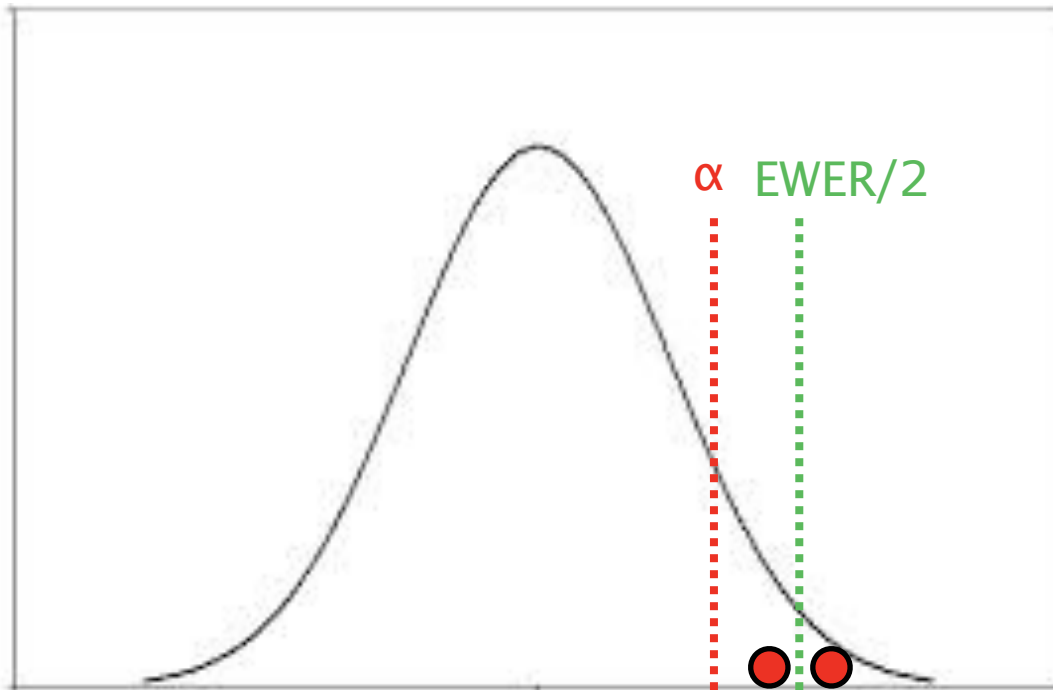
The **Dunn correction** states that we can control our experimentwise error rate (EWER) by setting our decision threshold (α) to our intended experimentwise error rate divided by the number of comparisons (EWER/C). See [multiple.comparisons.r](https://www.psychstat.org/multiplecomparisons.r) for a demo!

Why does EWER/C eliminate errors?

To see why the Dunn correction eliminates errors (over the long run!), all you need to do is think about the distribution of p-values.

The original α divides the distribution of p-values into those that lead to acceptance of H_0 , and those that lead to an error (rejection of H_0)

If you have two comparisons per experiment, you will basically double the number of errors over the long run. These errors will be evenly distributed throughout the error zone in the tail.



The Dunn correction cuts the tail. In this case, it cuts it in half. This means that you will eliminate half of the errors, which is what you want!

The same logic scales up to any number of comparisons. By cutting the zone, on average, you will move $C-1$ errors into the non-error zone!

A note on the name: Dunn vs Bonferroni

Olive Dunn proposed the correction in a paper in 1961, but didn't name it. She cites Bonferroni once for the use of the inequality. The paper is super mathy!

Bonferroni was a male mathematician who never worked on statistics.

The field seems to have named the correction sometime after Dunn's 1961 paper, and chose Bonferroni for the name. The question is why.

One possibility is to give credit for the use of the inequalities. But that doesn't go through. All statistical tests are named after the statisticians who discovered them, including correction procedures: Turkey, Scheffe, Fisher, etc. We don't name things after the mathematicians whose math they used (Euler, Leibniz, etc).

Another possibility is that Dunn did not invent the correction. Perhaps it was around before her, and called the Bonferroni correction, and she just did the mathematical work to figure out its properties. In that case, the name should be Dunn-Bonferroni. All other modifications of existing tests do this appending: the Holm-Bonferroni correction is a modification of Dunn's correction proposed by Holm in 1979 (where he doesn't cite Dunn!).

Another possibility is **sexism in science**.

Planned versus Post-Hoc Comparisons

Two types of comparisons

Planned Comparison:

This is a comparison that you specify before running your experiment (and crucially before looking at any data). Basically, you have a specific hypothesis, and decide that the best way to test it is to compare certain levels to each other.

Post-hoc Comparison:

This is a comparison that you decide to run after looking at your data. Basically, you see a difference in your data, and are curious to know if it is significant. This isn't theory-driven testing, this is data-driven testing.

I know it sounds strange, but under NHST, this difference matters for the probabilities of Type I errors.

Planned Comparisons and correction

Imagine your experiment has 3 comparisons, but you decide before you look at your data that you are only going to look at comparison 1.

	e1	e2	e3	...															e20
comp1																			
comp2																			
comp3																			

This eliminates the errors from the comparisons that you are not looking at (because you never see them). So you only need to correct for the one comparison that you are looking at:

$$\alpha = \frac{\text{EWER}}{C}$$

$$\alpha = \frac{.05}{1} = .05$$

The idea here is that, when you use **planned comparisons**, the C in equation is the number of **comparisons that you are actually looking at**.

Planned Comparisons and correction

Imagine your experiment has 3 comparisons, but you decide before you look at your data that you are going to look at **comparison 1 and comparison 2**.

	e1	e2	e3	...														e20
comp1																		
comp2																		
comp3																		

This eliminates the errors from comparison 3 (because you never see them). But you still see the errors from comparison 1 and comparison 2, so you need to correct for that:

$$\alpha = \frac{\text{EWER}}{C}$$

$$\alpha = \frac{.05}{2} = .025$$

The idea here is that, when you use **planned comparisons**, the C in equation is the number of **comparisons that you are actually looking at**.

Post-hoc comparisons and correction

Imagine your experiment has 3 comparisons, but you decide to look for the largest effect in each experiment:

	e1	e2	e3	...														e20
comp1																		
comp2																		
comp3																		

Notice that the **errors** are the largest effect in their experiments. This means that you will necessarily find errors in all three comparisons. So this process of choosing the largest effect **eliminates no errors**. So you have to correct:

$$\alpha = \frac{\text{EWER}}{C}$$

$$\alpha = \frac{.05}{3} = .0167$$

The idea here is that, when you use **post-hoc comparisons**, the C in equation is the **total number of possible comparisons**.

OK, so what do we do?

If you have **planned comparisons**, just run the Dunn correction with your actual number of comparisons (C).

If you have **post-hoc comparisons**, you can't use the actual number of comparisons, because you chose C after looking at the data.

Option 1: Run the Dunn correction with C equal to the maximum number of comparisons licensed by your experimental design.

The only downside of this option is that this could be a very extreme correction (imagine 10 possible comparisons, which would be $.05/10 = .005$). If the number of comparisons you are actually running is small, and the number of possible comparisons is large, you may be over-correcting, and thus making it less likely that you will detect significant differences that are really there.

Option 2: Run one of the methods that have been proposed to replace the Dunn method, like Tukey's Honestly Significant Difference (Tukey's HSD) or Scheffe's method. These were designed to provide good control of EWER without sacrificing as much power as the Dunn method.

Optional Stopping is Multiple Comparison

Optional Stopping is when you look at your results, and decide whether or not to collect more data based on what you see. (e.g., if the results are significant, you stop; if they are not significant, you collect more participants)

Optional Stopping increases the experimentwise error rate, just like a multiple comparison. So **you have to choose the number of participants you are going to collect before your experiment**

add slide or animation for dance of the p-values and optional stopping.

If you do collect more data, you need to apply a correction like Dunn.

Homework assignment: Adapt the code in [multiple.comparisons.R](#) to demonstrate that optional stopping increases the experimentwise error rate. Then show that the Dunn correction fixes the problem.

Criticisms of NHST

Despite the ubiquity of NHST as the analysis method for psychology, most people who think seriously about data analysis are critical of it.

I would love to spend a couple of weeks talking about these criticisms and really diving into the heart of the data analysis problem. But there is not time.

But you all should know enough now to read papers that are critical of NHST and think about the problems for yourself. So I've collected a bunch of good ones into folder you can download from the website. They are:

Balluerka, N., Gómez, J., & Hidalgo, D. (2005). The controversy over null hypothesis significance testing revisited. *Methodology*, 1(2), 55-70.

Cohen, J. (1994). The earth is round ($p < .05$). *American Psychologist*, 49, 997-1003.

Gigerenzer, G. (2004). Mindless statistics. *The Journal of Socio-Economics*, 33, 587-606.

Hubbard, R., & Lindsay, R. M. (2008). Why p values are not a useful measure of evidence in statistical significance testing. *Theory and Psychology* 18(1), 69-88.

Hubbard, R & M. J. Bayarri. (2003) P Values are not Error Probabilities.

Nickerson, R. (2000). Null hypothesis significance testing: A review of an old and continuing controversy. *Psychological Methods*, 5(2), 241-301

Table of contents

1. Introduction: You are already an experimentalist
2. Conditions
3. Items
4. Ordering items for presentation
5. Judgment Tasks
6. Recruiting participants
7. Pre-processing data (if necessary)
8. Plotting
9. Building linear mixed effects models
10. Evaluating linear mixed effects models using Fisher
11. Neyman-Pearson and controlling error rates
12. Bayesian statistics and Bayes Factors
13. Validity and replicability of judgments
14. The source of judgment effects
15. Gradience in judgments

Section 1:
Design

Section 2:
Analysis

Section 3:
Application

Bayes Theorem

Probability Basics

Probability: A mathematical statement about how likely an event is to occur. It takes a value between 0 and 1, where 0 means the event will never occur, and 1 means the event is certain to occur. (You can also think of it as a percentage 0% to 100%)

Here is an example:

Let's say you have a standard deck of cards. Cards have **values** and **suits**. There are 13 values and 4 suits, leading to 52 cards:

	2	3	4	5	6	7	8	9	10	J	Q	K	A
♠	•	•	•	•	•	•	•	•	•	•	•	•	•
♣	•	•	•	•	•	•	•	•	•	•	•	•	•
♦	•	•	•	•	•	•	•	•	•	•	•	•	•
♥	•	•	•	•	•	•	•	•	•	•	•	•	•

Let's say you pull a card at random from the deck. What is the probability of drawing a **Jack**?

Probability Basics

There are 52 possible cards. 4 of them are Jacks. So the probability of drawing a Jack is:

	2	3	4	5	6	7	8	9	10	J	Q	K	A
♠	•	•	•	•	•	•	•	•	•	•	•	•	•
♣	•	•	•	•	•	•	•	•	•	•	•	•	•
♦	•	•	•	•	•	•	•	•	•	•	•	•	•
♥	•	•	•	•	•	•	•	•	•	•	•	•	•

$$P(\text{J}) = \frac{\text{number of events you care about}}{\text{total number of events}} = \frac{4}{52} \approx .08$$

↑
This means "probability of J"

And what is the probability of drawing a heart?

$$P(\heartsuit) = \frac{\text{number of events you care about}}{\text{total number of events}} = \frac{13}{52} = .25$$

Conditional Probability

Conditional Probability: The probability of an event **given that** another event has occurred.

Let's say you draw a card, but can't see it. Your friend tells you it is a heart. What is the probability that it is a Jack?

This is a conditional probability. It is asking what the probability of a Jack is given that the card is a heart.

$$P(\text{Jack} \mid \text{heart}) = \frac{\text{number of events that are both Jack and heart}}{\text{number of heart events}} = \frac{1}{13}$$

The pipe symbol means "given that"

[illegible]

Conditional Probability

Conditional Probability:

The probability of an event **given that** another event has occurred.

$$P(\text{B}|\text{A}) = \frac{P(\text{A and B})}{P(\text{A})}$$

Notice that the format is very similar to the general probability equation that we've already seen:

$$\text{Probability(Event)} = \frac{\text{outcomes in the event}}{\text{total possible outcomes}}$$

The difference is that the denominator is not all possible outcomes, but just the outcomes that have the first event (A).

This is the mathematical way of saying that we are **restricting our attention** to just the A outcomes, and then looking for a specific event that is a subset of A outcomes.

Reversing the order makes a difference!

Notice that we can ask two different questions about Jacks and hearts:

What is the probability of a Jack given that the card is a heart?

$$P(\text{J} \mid \heartsuit) = \frac{1}{13}$$

What is the probability of a heart given that the card is a Jack?

$$P(\text{♥} \mid \text{J}) = \frac{1}{4}$$

[illegible]

Reversing the order makes a difference!

What is the probability of **being a movie star** given that **you live in LA**?

$$\frac{\text{number of movie stars in LA}}{\text{number of people that live in LA}} = \frac{250?}{\sim 4,000,000} = \text{very low!}$$

What is the probability of **living in LA** given that **you are a movie star**?

$$\frac{\text{number of movie stars in LA}}{\text{number of movie stars}} = \frac{250?}{\sim 300} = \text{very high!}$$

Reversing the order makes a difference!

What is the probability of **being a dark wizard** given that **are in slytherin**?

$$\frac{\text{number of dark wizards from Slytherin}}{\text{number of students from slytherin}} = \frac{30?}{5,000?} = \textbf{fairly low!}$$

What is the probability of **being from Slytherin** given that **you are a dark wizard**?

$$\frac{\text{number of dark wizards from Slytherin}}{\text{number of dark wizards}} = \frac{30?}{30?} = \textbf{very high!}$$

Bayes Theorem states the relationship between inverse conditional probabilities

Even though the two directions of the probabilities are not identical, Bayes Theorem tells us that they are related to each other:

$$P(\text{J} | \text{♥}) = \frac{P(\text{♥} | \text{J}) \times P(\text{J})}{P(\text{♥})}$$

Bayes Theorem

Since we already have these numbers, we can verify this pretty easily:

$$\frac{1}{13} = \frac{\frac{1}{4} \times \frac{4}{52}}{\frac{13}{52}} = \frac{\cancel{\frac{1}{4}} \times \frac{1}{13}}{\cancel{\frac{1}{4}}}$$

[illegible]

Bayes Theorem, general form

Bayes Theorem: $P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$

Historical Note: Thomas Bayes (1701-1761) was a minister in England who was the first to use the rules of probability to show us this relationship. It is now called **Bayes' Theorem** in his honor.



I know it seems like I pulled this equation out of thin air, but it is actually a very simple (algebraic) consequence of the definition of conditional probabilities.

Deriving Bayes Theorem

Here is the derivation of Bayes Theorem. As you can see, it is actually fairly simple. (The real work is in calculating the different components when you want to use it.)

1. Definition of conditional probability:

$$P(B|A) = \frac{P(A \text{ and } B)}{P(A)}$$

Algebra - multiply by denominator:

$$P(B|A) * P(A) = P(A \text{ and } B)$$

2. Definition of conditional probability:

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$$

Algebra - multiply by denominator:

$$P(A|B) * P(B) = P(A \text{ and } B)$$

3. Set 1 and 2 equal to each other:

Algebra - divide by P(A):

$$P(B|A) * P(A) = P(A|B) * P(B)$$

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

Some philosophy

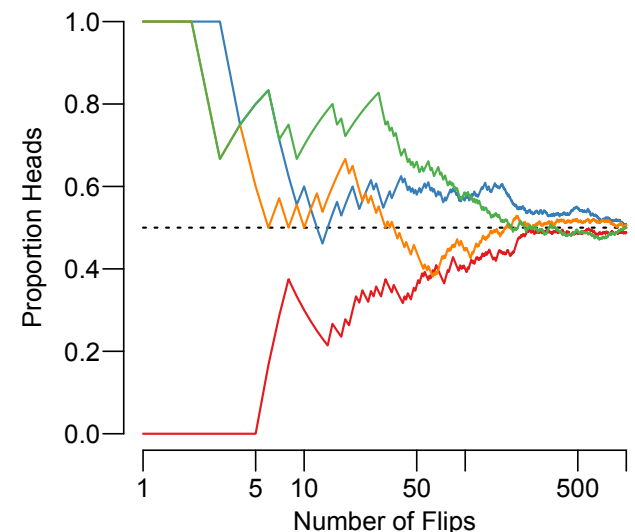
Two approaches to probability

Philosophically speaking, there are two ways of thinking about probabilities. People disagree about labels for these, but two common ones are **objective** and **subjective**.

Roughly speaking, **objective probabilities are descriptions of the lack of predictability that is inherent in some events**, like flipping a coin. This unpredictability can be measured with real-world observations.

Objective probabilities can be thought of as **long-run relative frequencies**. If you were to repeat the event over and over, probability is the proportion that you would get.

Here is a plot of coin flips over time (run four times). As you can see, objective probabilities don't tell you anything about individual events, but over time, the proportion becomes .5



Two approaches to probability

Roughly speaking, **subjective probabilities are descriptions of our uncertainty of knowledge about an event**. We use subjective probabilities when we say “there is a 10% chance of rain tomorrow”. This is not about long-run relative frequency. We aren’t going to repeat the event each day to see if it rains 10% of the time. Instead, we are talking about the strength of our beliefs in an event.

NHST approaches to statistics (Fisher and Neyman-Pearson) are (mostly) aligned with the **objective approach to probability**. The probabilities that we calculate are the hypothetical proportions that we would obtain if we actually ran the experiments over and over again. They are intended to be interpreted as long-run relative frequencies. This is why people call NHST approaches to statistics **frequentist**. The probabilities are related to objective frequencies.

Bayesian statistics are aligned with the **subjective approach** to probability. The probabilities in Bayesian statistics are not intended to be interpreted as long-run relative frequencies. For Bayesians, it makes no sense to talk about hypothetical repeated experiments. There is one experiment, and we want to know **how strong our beliefs should be in different theories** (similar to the example about rain).

Bayes Theorem for Science

Bayes Theorem for science

We can use Bayes Theorem to tell us how strongly we should believe in a hypothesis given the data that we observed.

$$\begin{array}{ccccc} \text{posterior} & & \text{likelihood} & & \text{prior} \\ \downarrow & & \downarrow & & \downarrow \\ P(\text{hypothesis} \mid \text{data}) & = & \frac{P(\text{data} \mid \text{hypothesis}) \times P(\text{hypothesis})}{P(\text{data})} \\ & & \uparrow & & \\ & & \text{evidence} & & \end{array}$$

The idea here is that you have a **prior belief** about a hypothesis (the prior probability). Then you get some evidence (data) from an experiment. Bayes Theorem tells you how to **update your prior beliefs using that evidence** and the **likelihood of the data given that hypothesis**. Your updated beliefs are then called your **posterior** beliefs, or posterior probability.

A real world example

Medical tests are a classic example of trying to prove a theory (that you have a disease) with positive evidence (that you have symptoms of the disease).

Let's try an example.

100% of people with Disease X will test positive using a test.

1.5% of people without Disease X will also test positive using a test.

Let's say someone goes to the doctor to take a test, and the result comes back **positive**. What is the probability that they have Disease X?

This example is about **updating** beliefs. Prior to the test, you have beliefs about having the disease. After the test, you have more evidence, and need to update those beliefs. The question is what the new beliefs should be. Most people, and an unfortunately large number of doctors, will say 98.5%. But this is **wrong**! To really calculate the probability of our theory, we need to use Bayes Theorem to calculate the posterior probability!

Bayes Theorem and Medical Tests

This is what we want to know

This is how good the test is when the disease is present. For X, it is 100% or 1.

This is the likelihood of having X in the US, period. Let's say it is 0.35% or .0035. People often ignore this number!

$$P(\text{having } X \mid \text{a positive test}) = \frac{P(\text{a positive test} \mid \text{having } X) \times P(\text{having } X)}{P(\text{a positive test})}$$

This is a tricky number to calculate. It is the total likelihood of getting a positive result, whether you have Disease X or not. You add up all of the **true positives** (0.35%) and all of the **false positives** (1.5% of the 99.65% of the population that doesn't have X). For X, this total is 1.84% or .0184.

Plugging in the numbers

$$P(\text{having } X \mid \text{a positive test}) = \frac{1 \times .0035}{.0184}$$

$$P(\text{having } X \mid \text{a positive test}) = .19 = 19\%$$

The probability that any random person in the US has Disease X is 0.35%.

Given the numbers that I gave you (which are fairly accurate for some deadly diseases), we see that a positive test means that the probability of having Disease X increases from 0.35% to 19%. So we should **update our beliefs** from .35% to 19%

This is a far cry from the 98.5% that many people (and some doctors) believe when they hear about a positive test.

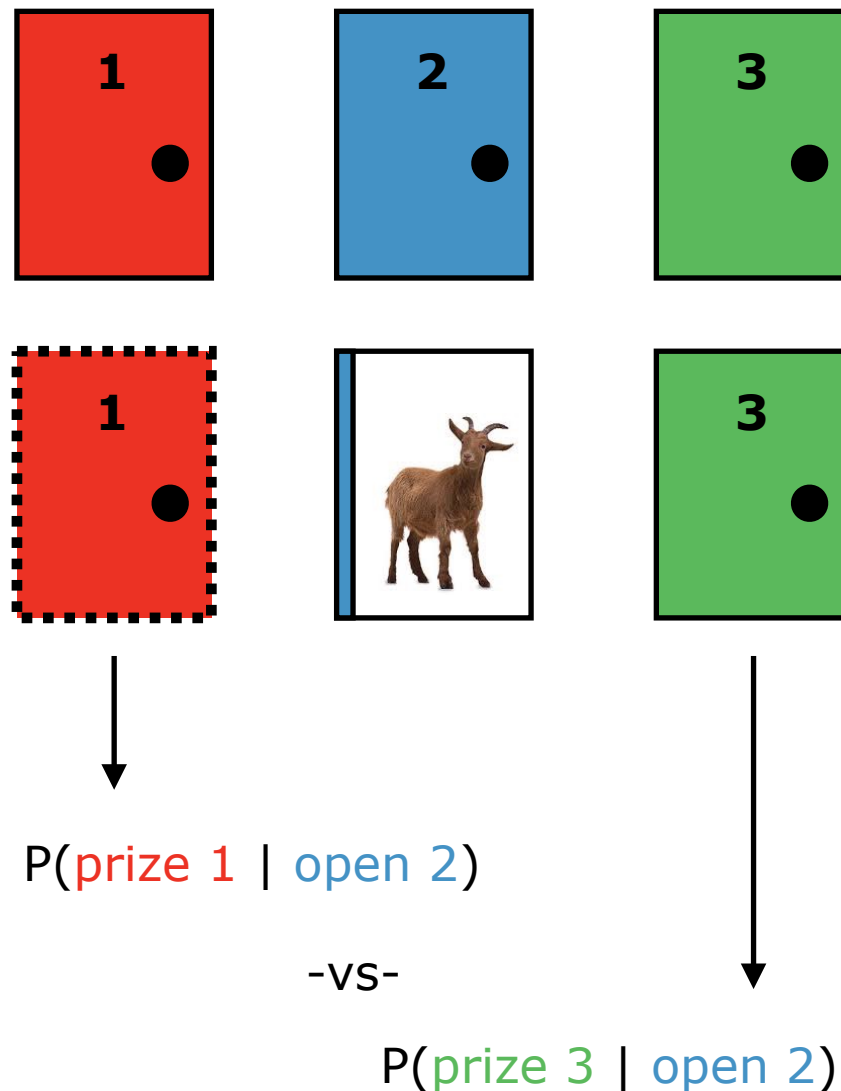
A fun example: the Monty Hall problem

There was a gameshow in the 70s hosted by Monty Hall that had a game as follows. There are 3 doors. One has money behind it, the other two have goats.

The contestant picks door number 1. Monty then opens door 2, and shows them a goat.

Monty then offers them a choice: keep their door, or switch to door 3. What should they do?

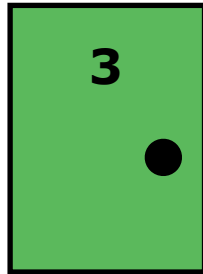
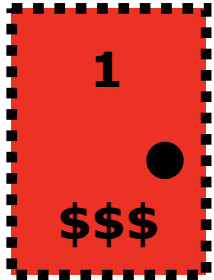
The contestant should choose the door with highest probability of a prize. So they need to know: (i) the probability that the money is behind door 1 given that Monty opened 2, and (ii) the probability that the money is behind 3 given that Monty opened 2.



The two conditional probabilities

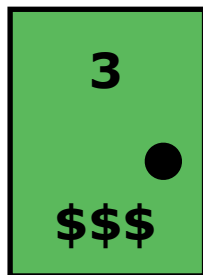
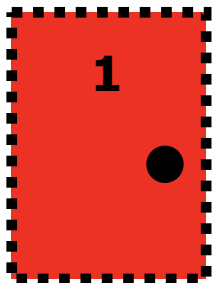
P(prize 1 | open 2)

$$= \frac{P(\text{open 2} \mid \text{prize 1}) \times P(\text{prize 1})}{P(\text{open 2})}$$



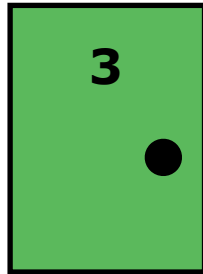
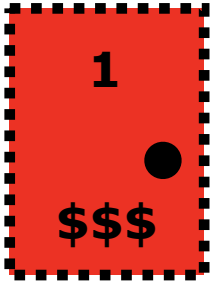
P(prize 3 | open 2)

$$= \frac{P(\text{open 2} \mid \text{prize 3}) \times P(\text{prize 3})}{P(\text{open 2})}$$



The first conditional probability

$$P(\text{prize 1} \mid \text{open 2}) = \frac{P(\text{open 2} \mid \text{prize 1}) \times P(\text{prize 1})}{P(\text{open 2})}$$



$P(\text{prize 1})$

This is $1/3$. There are 3 doors, and the TV show could choose any of them. The starting probability (prior probability) for each door is $1/3$.

$P(\text{open 2} \mid \text{prize 1})$

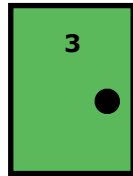
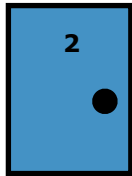
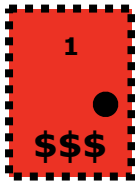
This is $1/2$. If the prize is behind door 1, then Monty can choose either door 2 or door 3.

$P(\text{open 2})$

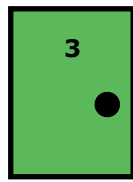
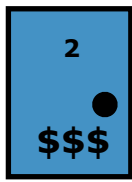
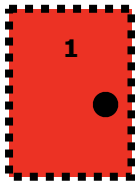
This is the tricky one. The answer is $1/2$, but I need the entire next slide to show you how we get that.

Calculating $p(\text{open } 2)$

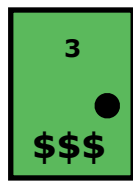
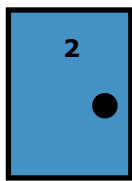
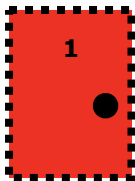
The tricky for calculating the evidence is that you have to consider every possible theory (prize 1, prize 2, prize 3), and calculate the probability of the data (open 2) under each theory.



If the prize is behind door 1, the probability of opening 2 is $1/2$.



If the prize is behind door 2, the probability of opening 2 is 0. Monty can't open that door.



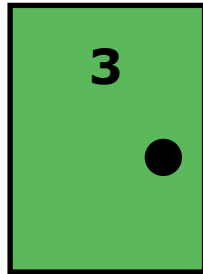
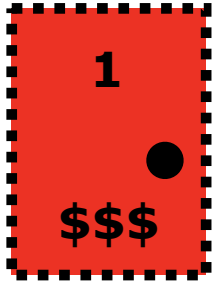
If the prize is behind door 3, the probability of opening 2 is 1. Monty can't open door 1 because the contestant chose it. He can't open 3 because it has the prize. So he has to choose door 2.

There are 3 theories, each with a prior probability of $1/3$. We weight each theory by its prior probability, which means multiplying each by $1/3$, and then sum. This tells us how likely open 2 would be out of all possible worlds:

$$P(\text{open } 2) = (1/3 \times 1/2) + (1/3 \times 0) + (1/3 \times 1) = 1/2$$

The first conditional probability

$$P(\text{prize 1} \mid \text{open 2}) = \frac{P(\text{open 2} \mid \text{prize 1}) \times P(\text{prize 1})}{P(\text{open 2})}$$



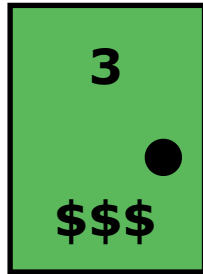
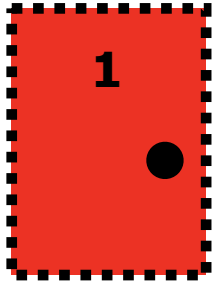
$$P(\text{prize 1} \mid \text{open 2}) = \frac{1/2 \times 1/3}{1/2}$$

$$P(\text{prize 1} \mid \text{open 2}) = 1/3$$

We can actually take a shortcut now. Since the prize has to be behind door 1 or door 3, and we know door 1 is 1/3, and we know probability must equal 1, then that means that probability for door 3 must be 2/3! But let's do the calculation anyway.

The second conditional probability

$$P(\text{prize 3} \mid \text{open 2}) = \frac{P(\text{open 2} \mid \text{prize 3}) \times P(\text{prize 3})}{P(\text{open 2})}$$



$P(\text{prize 3})$

This is $1/3$. There are 3 doors, and the TV show could choose any of them. The starting probability (prior probability) for each door is $1/3$.

$P(\text{open 2} \mid \text{prize 3})$

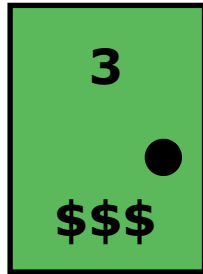
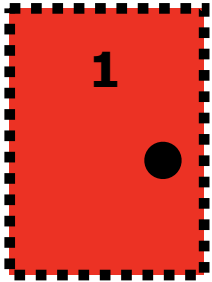
This is 1. Monty can't choose 1 because the contestant chose it. He can't choose 3 because it has the prize. He has to choose 2.

$P(\text{open 2})$

We already know that this is $1/2$ by the previous calculation.

The second conditional probability

$$P(\text{prize 3} \mid \text{open 2}) = \frac{P(\text{open 2} \mid \text{prize 3}) \times P(\text{prize 3})}{P(\text{open 2})}$$



$$P(\text{prize 3} \mid \text{open 2}) = \frac{1 \times 1/3}{1/2}$$

$$P(\text{prize 3} \mid \text{open 2}) = 2/3$$

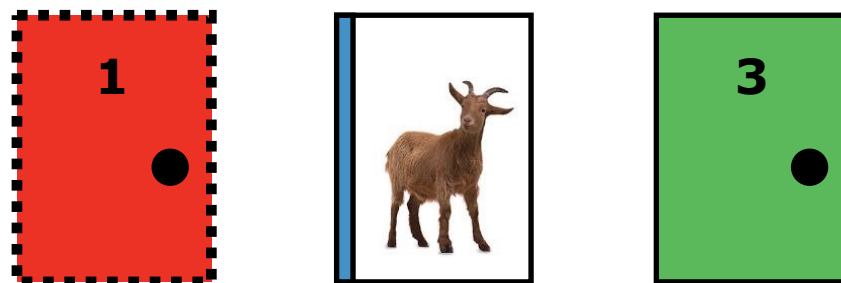
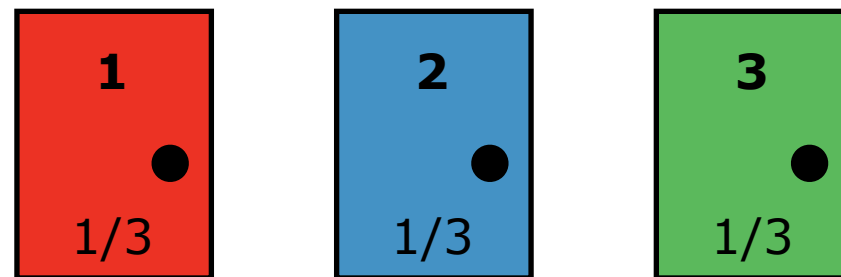
This is exactly what we calculated with our shortcut. So we can see that Bayes Theorem really works.

A fun example: the Monty Hall problem

When we begin the gameshow, each door has the same probability of having a prize.

But once Monty chooses a door, he is actually (perhaps unintentionally) giving us more information with which to **update our beliefs**. We can use Bayes Theorem to figure out how to update our beliefs.

Bayes Theorem tells us that we should switch doors. If we switch, we'll win 2/3 of the time. If we stay, we'll only win 1/3 of the time. (Though this example was phrased as about door 1, it is really about the contestant's door versus the unopened door.)



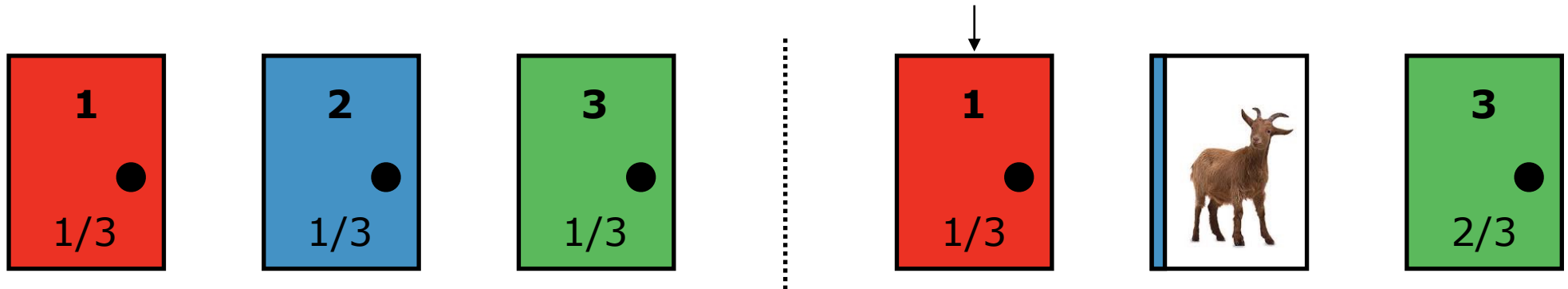
$$P(\text{prize 1} \mid \text{open 2}) = 1/3$$

-vs-

$$P(\text{prize 3} \mid \text{open 2}) = 2/3$$

Bayesian updating is NOT intuitive

In 1990, a reader asked columnist Marilyn vos Savant to solve Monty Hall's problem. She did, correctly (using logic rather than Bayes Theorem):



But people didn't believe her. The problem is that the result is not intuitive. With two choices left, many people believe that the answer is $1/2$ for both door 1 and door 3.

You can read some of the responses that people wrote to her answer. The embarrassing thing is that a number of them were academics/math teachers.

<http://marilynvossavant.com/game-show-problem/>

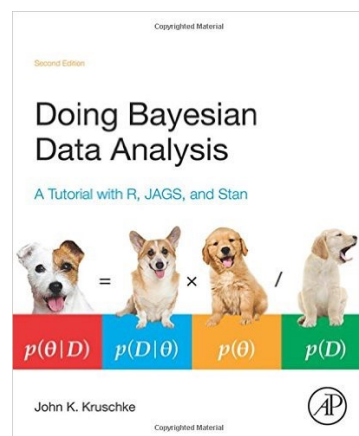
Today, we can run simulations to prove this. The script [monty.hall.r](#) contains a simulation to show you that the answer is $1/3$ and $2/3$, not $1/2$ and $1/2$.

Bayesian Statistics

Doing full Bayesian statistics can be very complicated, because some of the components of Bayes Theorem are difficult to calculate for real-world scientific hypotheses.

$$P(\text{hypothesis} \mid \text{data}) = \frac{P(\text{data} \mid \text{hypothesis}) \times P(\text{hypothesis})}{P(\text{data})}$$

This is why Bayesian analysis is very common in the computational modeling world (where they develop tools to estimate complex probabilities), but less common in the experimental world.



Kruschke's **Doing Bayesian Data Analysis** is a great one-stop shop for beginning Bayesian statistics. It has a gentle introduction to probability and Bayes, and even organizes Bayesian models around the NHST tests that they are most like. It is a great way to transition, but it makes it clear that Bayesian statistics is more about modeling than about creating statistical tests.

Bayes Factors

Bayes Factors

Because full Bayesian statistics can be very complicated, some statisticians have suggested that experimentalists could use **Bayes Factors** to do a Bayesian analysis without having to become computational modelers.

Bayes Factors are the ratio of the probability of the data under one hypothesis to the probability of the data under a second hypothesis. Typically, the two hypotheses are the experimental hypothesis (H1) and the null hypothesis (H0), though they could be any two hypotheses that you want:

$$\text{Bayes Factor}_{1,0} = \frac{P(\text{data} \mid H1)}{P(\text{data} \mid H0)}$$

This is the ratio of the probability of the data under H1 to H0.

You can setup the ratio in whichever direction is most convenient for your question (do you care more about H1, or more about H0):

$$\text{Bayes Factor}_{0,1} = \frac{P(\text{data} \mid H0)}{P(\text{data} \mid H1)}$$

This is the ratio of the probability of the data under H0 to H1.

Interpreting Bayes Factors

Bayes Factors are a ratio, so they will range from 0 to infinity. For example, a $BF_{1,0}$ of 3 means that the data is 3x more likely under H_1 than H_0 .

Jeffries (1939/1961) suggested some rules of thumb for interpreting Bayes Factors.

$$BF_{1,0} = \frac{P(\text{data} \mid H_1)}{P(\text{data} \mid H_0)}$$

$$BF_{0,1} = \frac{P(\text{data} \mid H_0)}{P(\text{data} \mid H_1)}$$

BF	Evidence
0 to .01	extreme for H_0
.01 to .1	strong for H_0
.1 to .33	substantial for H_0
.33 to 1	anecdotal for H_0
1 to 3	anecdotal for H_1
3 to 10	substantial for H_1
10 to 100	strong for H_1
100 to ∞	extreme for H_1

BF	Evidence
0 to .01	extreme for H_1
.01 to .1	strong for H_1
.1 to .33	substantial for H_1
.33 to 1	anecdotal for H_1
1 to 3	anecdotal for H_0
3 to 10	substantial for H_0
10 to 100	strong for H_0
100 to ∞	extreme for H_0

Deriving Bayes Factors from Bayes Theorem

Bayes Factors come directly from Bayes theorem. The basic idea is to set up a ratio between two tokens of Bayes theorem - one for H1 and one for H0:

$$\frac{P(\text{H1} \mid \text{data})}{P(\text{H0} \mid \text{data})} = \frac{P(\text{data} \mid \text{H1}) \times P(\text{H1})}{P(\text{data})}$$

$$= \frac{P(\text{data} \mid \text{H0}) \times P(\text{H0})}{P(\text{data})}$$

This is the ratio of the probability of H1 to H0 given some data. In other words, this would tell us how much more likely H1 is than H0 given some data. That would be really useful, but it is difficult to calculate.

Then we can simplify using basic algebra:

$$\frac{P(\text{H1} \mid \text{data})}{P(\text{H0} \mid \text{data})} = \frac{P(\text{data} \mid \text{H1}) \times P(\text{H1})}{\cancel{P(\text{data})}} \times \frac{\cancel{P(\text{data})}}{P(\text{data} \mid \text{H0}) \times P(\text{H0})}$$

$$\frac{P(\text{H1} \mid \text{data})}{P(\text{H0} \mid \text{data})} = \frac{P(\text{data} \mid \text{H1})}{P(\text{data} \mid \text{H0})} \times \frac{P(\text{H1})}{P(\text{H0})}$$

Deriving Bayes Factors from Bayes Theorem

Posterior Odds	Bayes Factor	Prior Odds
$\frac{P(\text{H1} \mid \text{data})}{P(\text{H0} \mid \text{data})}$	$= \frac{P(\text{data} \mid \text{H1})}{P(\text{data} \mid \text{H0})}$	$\times \frac{P(\text{H1})}{P(\text{H0})}$

One neat thing about seeing how Bayes Factors are derived is that you can see how useful they can be.

They are useful on their own. They tell you the (odds) ratio of the data under the two hypotheses.

They are also useful in combination with the priors. If you know the prior odds (the ratio of the two priors to each other), then BFs tell you how to update those priors into posteriors!

For example, a BF of 10 tells you that you should multiple your prior odds by 10 to get your posterior odds. So, if you thought H1 was 2 times more like than H0, a BF of 10 tells you to update this to 20x more likely!

Using Bayes Factors

There are two practical reasons to use Bayes Factors over full Bayesian models.

1. Bayes Factors don't require priors.

Prior probabilities are often very **subjective**. They are simply how likely you think a theory is. Different scientists will disagree on the priors for any given hypothesis. Bayes Factors sidestep this issue by sidestepping priors. They simply tell you how to update your priors based on the evidence.

2. Bayes Factors are very easy to calculate under certain assumptions

If you are willing to grant some simplifying assumptions, BFs are much easier to calculate than full Bayesian models.



The **BayesFactor R package** and accompanying blog at bayesfactor.blogspot.org

Jeff Rouder and colleagues have developed an R package that will calculate (simplified) BFs for you for a number of standard designs in experimental psychology. It is as easy as any test in R.

Why the excitement?

One reason that so many people are excited about Bayesian statistics is that it appears to give us (scientists) the information that we really want.

We want to know how likely a theory is to be true based on the evidence we have. That is what Bayes promises us... based on the axioms of probability!

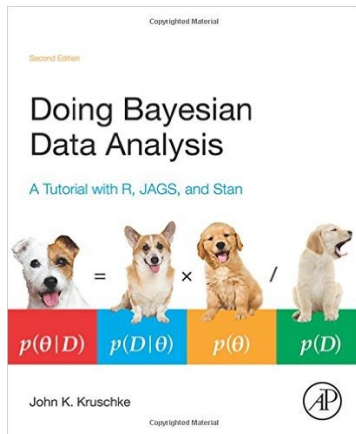
This is in stark contrast to NHST, which gives the probability of the data assuming the uninteresting theory is true. It is only through logical acrobatics (Fisher's disjunction, or hypothetical infinite experiments) that we can convert that into something usable.

Bayesian statistics: $P(\text{experimental hypothesis} \mid \text{data})$

NHST: $P(\text{data} \mid \text{null hypothesis})$

Bayesian statistics also overcomes some other limitations of NHST, such as not being able to test the null hypothesis directly (for Bayes, it is just another hypothesis), and not being able to peak at the data without increasing Type I errors (it is called the optional stopping problem; for Bayes, data is data). There is no space to cover these here, but I want to mention them so you can search for them.

Where to find more about Bayes



Kruschke's **Doing Bayesian Data Analysis** is a great one-stop shop for beginning Bayesian statistics. It has a gentle introduction to probability and Bayes, and even organizes Bayesian models around the NHST tests that they are most like. It is a great way to transition, but it makes it clear that Bayesian statistics is more about modeling than about creating statistical tests.



The **BayesFactor R package** and accompanying blog at bayesfactor.blogspot.org

Comparing Frequentist and Bayesian statistics

Subjectivity, Subjectivity, and the Null

The comparison of frequentist and Bayesian stats is a large and complex topic. I can't do it justice here, but I can mention three major differences:

1. The philosophy of probability

As previously mentioned, frequentists are more closely aligned with objective probability, and Bayesians are aligned with subjective probability.

2. The “subjectivity” of the calculation

Fisher explicitly developed his NHST in response to Bayesian statistics! He thought the specification of priors was too “subjective”, so he focused on the likelihood, which he found to be more “objective”.

$$P(\text{hypothesis} \mid \text{data}) = \frac{P(\text{data} \mid \text{hypothesis}) \times P(\text{hypothesis})}{P(\text{data})}$$

3. Proving the null

NHST can't make any claims about the null hypothesis, because it is assumed to be true. Bayesian stats can “prove the null”. It is simply another hypothesis.