PSYC 5316 – Week 3

Thomas J. Faulkenberry, Ph.D.

September 10, 2018

Before getting started

We need to install two packages for our work tonight:

```
install.packages("tidyverse")
install.packages("BayesFactor")
```

Building a *flanker task* (Eriksen & Eriksen, 1974)

- basic idea – when people are asked to respond to a stimulus that is surrounded ("flanked") by irrelevant stimuli, the irrelevant stimuli can interfere with the response.

- two types of trials:

congruent	»»>	SSSSS
incongruent	»<»	SSHSS

- *flanker effect* = RTs and error rates increase on incongruent trials (relative to congruent trials)

- the flanker effect is a popular measure of attention mechanisms

Goal of Lab 1

- our experiment will be based on Heitz and Engle (2007)
- stimuli will be five letter strings consisting of S and H

congruent	SSSSS	HHHHH
incongruent	SSHSS	HHSHH

- three blocks of 80 trials each (240 trials total)
- will impose RT deadlines on Block 2 (600 ms) and Block 3 (300 ms)

Tasks for tonight

1. get data from Github into R

2. clean the data

3. look at the data

4. analyze the data

   - traditional t-test
   - Bayesian t-test

Task 1 - getting data from Github

The data is stored on Github – `https://git.io/fAVTQ`

Take a look at these data in a browser:

- what variables do we *really* need?
- what variables can we ignore?

Task 1 - getting data from Github

```
rawdata = data.frame()
filestem = "https://raw.githubusercontent.com/ \
   tomfaulkenberry/courses/master/fall2018/psyc5316/ \
   lab1/subject-"
for (n in 1:25){
  file = paste(filestem, n, ".csv", sep="")
  temp = read.csv(file)
  d = select(temp, condition, correct, live_row, \
    response_time)
  d$subject_nr = n
  rawdata = rbind(rawdata,d)
}
```

Task 2 - cleaning data

Even though we selected just a few of the available variables from the data, it still contains more than we want.

Heitz and Engle (2007) got rid of "practice trials". Take a look at their method section and see how they did this.

```
clean = rawdata %>%
  filter(live_row > 19) %>%
  mutate(rt = response_time)
```

Task 3 - visualizing the RTs

Let's take a look at the distribution of RTs:

```
clean %>%
  ggplot(aes(x=rt)) +
  geom_density()
```

Let's see if the distribution depends on `condition`:

```
clean %>%
  ggplot(aes(x=rt, group=condition)) +
  geom_density(aes(fill=condition))
```

Oops!

Let's fix the `condition` levels:

```
clean$condition = recode(clean$condition,
       conguent="congruent",
       Congruent="congruent",
       incongurent="incongruent",
       Incongruent="incongruent")
```

Now look again:

```
clean %>%
  ggplot(aes(x=rt, group=condition)) +
  geom_density(aes(fill=condition))  # much better :)
```

Task 3 - look at descriptives

Let's take a look at overall performance:

```
clean %>%
  group_by(condition) %>%
  summarize(mRT = mean(rt),
            sd = sd(rt),
            correct = sum(correct))
```

Task 4 - analyze data

In this experiment, we have one independent variable (condition: congruent, incongruent) nested within subjects ($n = 28$). Thus, we have a $S \times A$ design:

	Congruent	Incongruent
subj 1		
subj 2		
subj 3		
.		
.		
.		
subj 28		

So, we should have $28 \times 2 = 56$ different measurements. What should they be?

Task 4 - analyze data

The most common solution is to use the mean (but see Faulkenberry, 2017). Thus, we need to **collapse** our dataset into two means for each participant. Note that for RT analyses, we usually just look at *correct* trials:

```
collapsed = clean %>%
  filter(correct==1) %>%
  group_by(subject_nr,condition) %>%
  summarize(mRT = mean(rt))
```

Task 4 - analyze data

Let's now construct two separate vectors that represent the RTs (by subject) in each condition:

```
attach(collapsed)
congruent = mRT[condition=="congruent"]
incongruent = mRT[condition=="incongruent"]
```

Task 4 - analyze data

Let's do a *paired samples t-test*. To do this, we look at difference scores for each subject (incongruent - congruent) and compare to 0.

```
diff = incongruent-congruent
hist(diff)
t.test(diff, mu=0)
```

Task 4 - analyze data

We can also get a measure of effect size with Cohen's d. Recall that

$$d = \frac{\mu_1 - \mu_2}{\sigma}$$

In R, this is easy to code:

```
d = mean(diff)/sd(diff)
```

Task 4 - analyze data

We can also do a Bayesian version of the t-test.

Bayesian inference

The machinery underlying Bayesian inference is *Bayes Theorem*:

$$\underbrace{p(\mathcal{H} \mid \text{data})}_{\substack{\text{Posterior beliefs}\\ \text{about hypothesis}}} = \underbrace{p(\mathcal{H})}_{\substack{\text{Prior beliefs}\\ \text{about hypothesis}}} \times \underbrace{\frac{p(\text{data} \mid \mathcal{H})}{p(\text{data})}}_{\text{predictive updating factor}}$$

Bayesian inference

Natural action in our discipline is to *compare* two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$.

- Bayes theorem gives us a natural way to do this by computing **relative** likelihoods

$$\frac{p(\mathcal{H}_1 \mid \text{data})}{p(\mathcal{H}_0 \mid \text{data})} = \frac{\frac{p(\text{data}|\mathcal{H}_1)\cdot p(\mathcal{H}_1)}{p(\text{data})}}{\frac{p(\text{data}|\mathcal{H}_0)\cdot p(\mathcal{H}_0)}{p(\text{data})}}$$

which implies

$$\underbrace{\frac{p(\mathcal{H}_1 \mid \text{data})}{p(\mathcal{H}_0 \mid \text{data})}}_{\substack{\text{posterior beliefs}\\ \text{about hypotheses}}} = \underbrace{\frac{p(\mathcal{H}_1)}{p(\mathcal{H}_0)}}_{\substack{\text{prior beliefs}\\ \text{about hypotheses}}} \times \underbrace{\frac{p(\text{data} \mid \mathcal{H}_1)}{p(\text{data} \mid \mathcal{H}_0)}}_{\text{predictive updating factor}}$$

Bayesian inference

The predictive updating factor

$$B_{10} = \frac{p(\text{data} \mid \mathcal{H}_1)}{p(\text{data} \mid \mathcal{H}_0)}$$

tells us how much better $\mathcal{H}_1$ predicts our observed data than $\mathcal{H}_0$.

This ratio is called the **Bayes factor** (Jeffreys, 1961)

Example: suppose $B_{10} = 5$.

Interpretation: the observed data are 5 times more likely under the alternative hypothesis $\mathcal{H}_1$ than the null hypothesis $\mathcal{H}_0$.

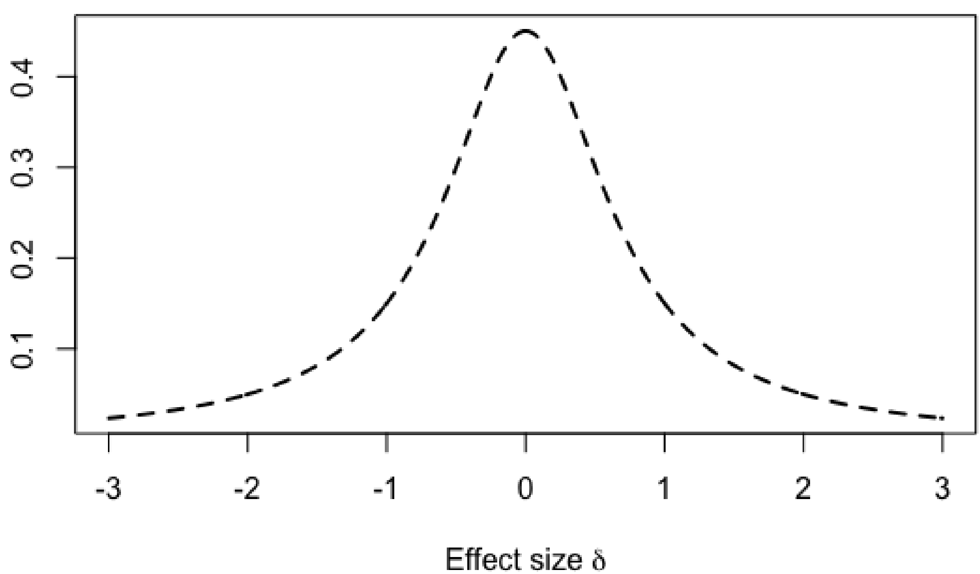
This is taken as **positive evidence** for the alternative $\mathcal{H}_1$

Bayesian \$t\$-test

The Bayesian \$t\$-test (Rouder et al., 2009) measures evidence by indexing the likelihood of the observed **effect size** under both the null and alternative:

$$\mathcal{H}_0 : \delta = 0$$
$$\mathcal{H}_1 : \delta \neq 0$$

Prior distribution for effect size δ – *Cauchy* distribution:



Bayesian \$t\$-test

Basic workflow:

- specify prior on effect size (default = Cauchy distribution)
- collect data
- estimate *posterior* distribution on effect size
- compare likelihood of $\delta = 0$ under prior and posterior (*updating*)

Task 4 - analyze data

Let's do a Bayesian \$t\$-test. The code is very simple:

```
bf = ttestBF(diff,mu=0)
```

We can also plot the prior and posterior to see the *updating*:

```
chains = posterior(bf, iterations = 10000)
plot(density(chains[,3]), xlim=c(-1,5), xlab="effect size")
x=seq(-1,5,0.01)
lines(x,dcauchy(x, scale=.707), lty=2)
legend(x=3, y=0.6, legend=c("Prior","Posterior"), \
  lty=c(2,1), bty="n")
```

Next time

For next week, we'll do the following:

- do these analyses with the full data set.
- learn about how to integrate *errors* into the analysis
- test a theoretical model of the flanker effect by constructing *conditional accuracy functions*