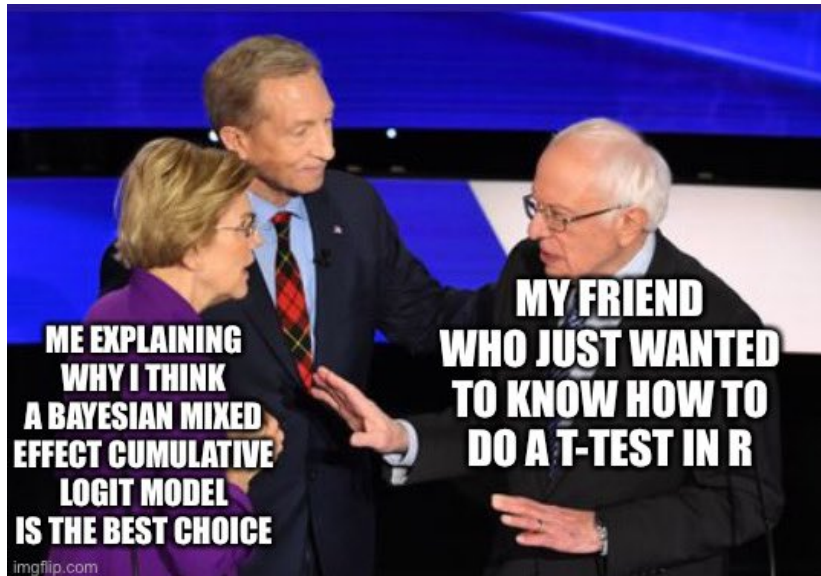# Tutorial - R

Blakemore Lab - Emily Towner

January 28, 2021

Why R?

Why R?

- Flexible
  - Multiple packages
  - Multiple functions
  - Create functions and packages yourself!

Why R?

- ▶ Data wrangling
  - ▶ Merge datasets
  - ▶ Score questionnaires
  - ▶ Transform scores
  - ▶ Exclude subjects
  - ▶ Group data together

Why R?

- Centralize workflow
  - Clean data
  - Score data
  - Analyze data
  - Create figures
  - Create tables
  - Write manuscripts
  - Create this presentation!

Why R?

- Reproducible
  - No point and click
  - Publish your code
  - Re-run your analyses with new data instantly

Why R?

- ▶ Re-usable
  - ▶ Make your own code repository
  - ▶ Use the same code for multiple studies
  - ▶ Make the same types of figures for multiple datasets

Basics

Basics - Software

- R
  - Runs in the background
  - Download **here**
    - https://www.r-project.org/
- R-Studio
  - Where you will do your work
  - Download **here**
    - https://www.rstudio.com/products/rstudio/download

Basics - Environment

Basics - Creating a script

- You can type and execute code straight from the console - but to save it and come back to it later, you want to use a script
- Make a folder on your computer where your data, scripts, and code will live
- Click: File -> New File -> R Markdown (recommended)

Programming in R

► What is a working directory?

```
> getwd()
[1] "/Users/emily"
>
```

► Benefits of R Markdown - sets working directory to root folder

tutorial_r		
Name		
data.csv		
script.Rmd		
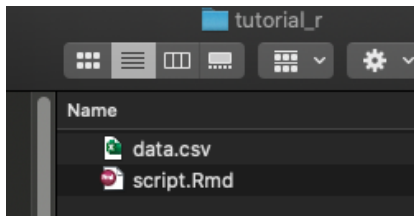
Programming in R

▶ Object-oriented programming
▶ You create objects and store them in the computer's "memory"

```r
an_object <- 2 + 1
an_object
```

```
## [1] 3
```

▶ It also shows up in your environment
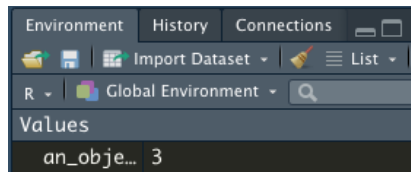
Programming in R

▶ Then you can do other things with it!
▶ Great for complex objects like sets of data.

```r
another_object <- 10
sum_of_objects <- an_object + another_object
sum_of_objects
```

```
## [1] 13
```

Programming in R - Data Types

- ▶ Objects can be more than just numbers
  - ▶ Numeric
  - ▶ Integer
  - ▶ Character/String
  - ▶ Logical
  - ▶ Factor

```r
class(sum_of_objects)
```

```
## [1] "numeric"
```

Programming in R - Data Structures

- ▶ Several but will talk about **Data Frames**
- ▶ Let's make our own

```r
participant <- as.factor(c(1,2,3,4,5))
sex <- c("Female", "Male", "Male", "Female", "Female")
age <- c(16.5, 21.4, 20.1, 18.2, 17.3)
score_1 <- c(85, 90, 73, 70, 92)
score_2 <- c(87, 89, 85, 88, 97)
my_data_frame <- cbind.data.frame(participant, sex, age,
                  score_1, score_2)
my_data_frame
```

```
##   participant    sex  age score_1 score_2
## 1           1 Female 16.5      85      87
## 2           2   Male 21.4      90      89
## 3           3   Male 20.1      73      85
## 4           4 Female 18.2      70      88
## 5           5 Female 17.3      92      97
```

Programming in R - Data Structures

- ▶ Indexing
  - ▶ I want to know what my third participant's score_1 was
  - ▶ Third participant is in Row = 3 and score_1 is in Column = 4

```
my_data_frame[3,4]
```

```
## [1] 73
```

- ▶ Adding variables
  - ▶ I want to create a difference score between Score 2 and Score 1 to see if students improved

```
my_data_frame$difference <- my_data_frame$score_2 -
  my_data_frame$score_1
my_data_frame$difference
```

```
## [1]  2 -1 12 18  5
```

Summary Statistics

```r
mean(my_data_frame$age)
```

```
## [1] 18.7
```

```r
range(my_data_frame$age)
```

```
## [1] 16.5 21.4
```

```r
levels(my_data_frame$sex)
```

```
## [1] "Female" "Male"
```

Summary Statistics

```r
mean(my_data_frame$score_1[my_data_frame$sex == "Female"])
```

```
## [1] 82.33333
```

```r
mean(my_data_frame$score_1[my_data_frame$sex == "Male"])
```

```
## [1] 81.5
```

Inferential Statistics

- ► Running inferential statistics in R is surprisingly easy
- ► Do females in this sample have statistically significantly higher test scores on score 1 than males?

```r
t.test(my_data_frame$score_1 ~ my_data_frame$sex)
```

```
##
##  Welch Two Sample t-test
##
## data:  my_data_frame$score_1 by my_data_frame$sex
## t = 0.077925, df = 2.1416, p-value = 0.9446
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##  -42.38243  44.04910
## sample estimates:
## mean in group Female   mean in group Male
##            82.33333              81.50000
```

Inferential Statistics

▶ On average, did participants get better from their first score to their second?

▶ In other words, was the difference between score 2 and score 1 statistically significantly different from 0?

```
t.test(my_data_frame$difference)
```

```
##
##  One Sample t-test
##
## data:  my_data_frame$difference
## t = 2.0837, df = 4, p-value = 0.1056
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -2.393813 16.793813
## sample estimates:
## mean of x
##       7.2
```

Plotting - then a demonstration!

- ► You will find you may need to install and load "packages" for different uses
- ► Plotting is a good example

```
# Install the needed package
# (you only need to do this once)
# Note how I've 'commented' this out
# using hashtags

# install.packages('tidyverse')
library(tidyverse)
```
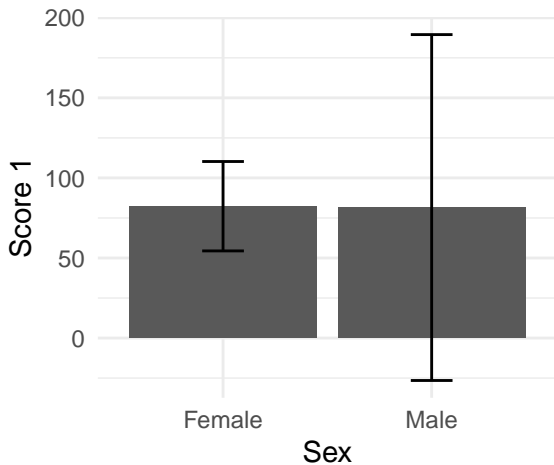
Plotting - histograms

```
ggplot(my_data_frame, aes(x = sex)) +
  geom_histogram(stat = "count") +
  xlab("Sex") + ylab("Count") +
  theme_minimal()
```
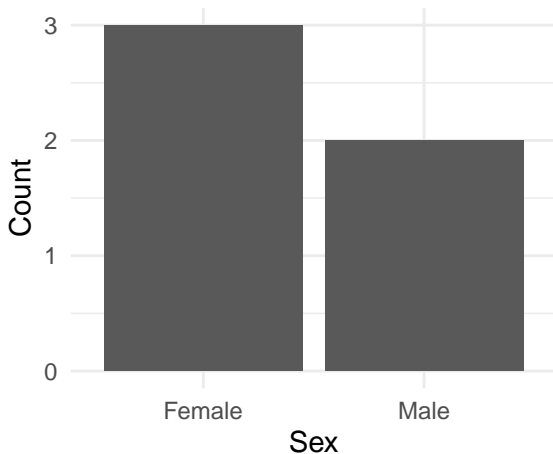
Plotting - bar plot

```
ggplot(my_data_frame, aes(x = sex, y = score_1)) +
  stat_summary(fun = mean, geom = "bar") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar'
  xlab("Sex") + ylab("Score 1") +
  theme_minimal()
```

R-guide

- ▶ I will send around an R-guide with more detailed instructions
- ▶ Please do not distribute as it is a work in-progress

A snippet from the guide:

7.4.1.2 Plotting a three-factor experimental design {#3-factor-plot}

Seriously? You're manipulating three factors in one experiment? I bet you're questioning some life decisions right now.

Thanks Ryan! Lolz

Now let's do a tutorial with a larger dataset!