[PSY202A Lab] Statistical Modeling in Psychological Sciences

Ihnwhi Heo, M.Sc.

Fall 2024

Contents

Chapter 1

Introduction

Hi everyone! I'm Ihnwhi.

It is my great pleasure to be your guest lecturer for PSY202A.

Statistical modeling is a key component in conducting research in the psychological sciences. While many statistical toolkits are available to researchers, R is arguably one of the most useful free and open-source statistical software programs. It offers a dizzying array of analytic options to answer your important and exciting research questions. In the upcoming four lab sessions, you will be introduced to R and become familiar with its capabilities. These sessions are designed to help you get acquainted with the fundamentals of R and learn how to use it wisely as the next generation of psychologists. You will then be guided through summarizing and analyzing data using R.

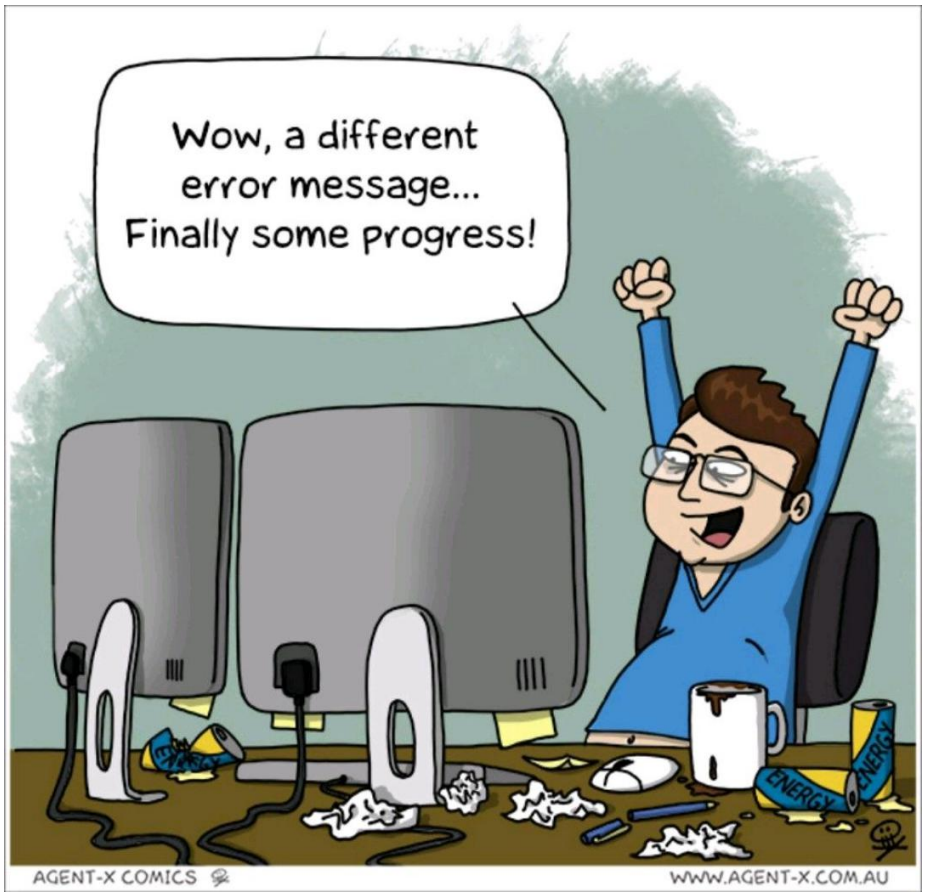
I must acknowledge that these materials are the result of the dedication and efforts of many remarkable researchers who have contributed to improving statistics resources. These include Fan Jia, Marieke Visser, Luca Marvin, Terrence Jorgensen, and Sunthud Pornprasertmanit.

Are you ready? Let's get it on!

Chapter 2

Introduction to R: Part 1

2.1 Word of wisdom

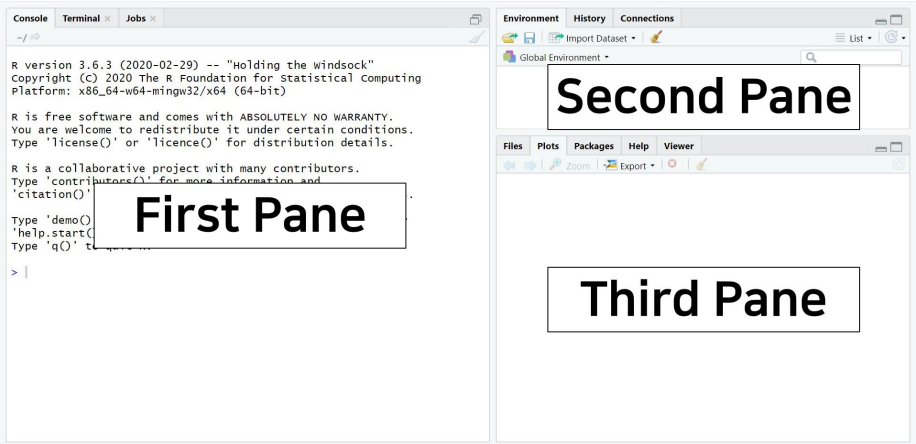


- Don't worry about making mistakes; it's part of the process.
- Feel free to ask questions to me or your peers.
- Remember, there isn't just one way to solve a problem.
- Be a wise user of resources like Google, YouTube, or AI.
- Keep in mind that you're not the only one struggling.

2.2 How R you?

2.2.1 Open RStudio

- Can you find three panes?

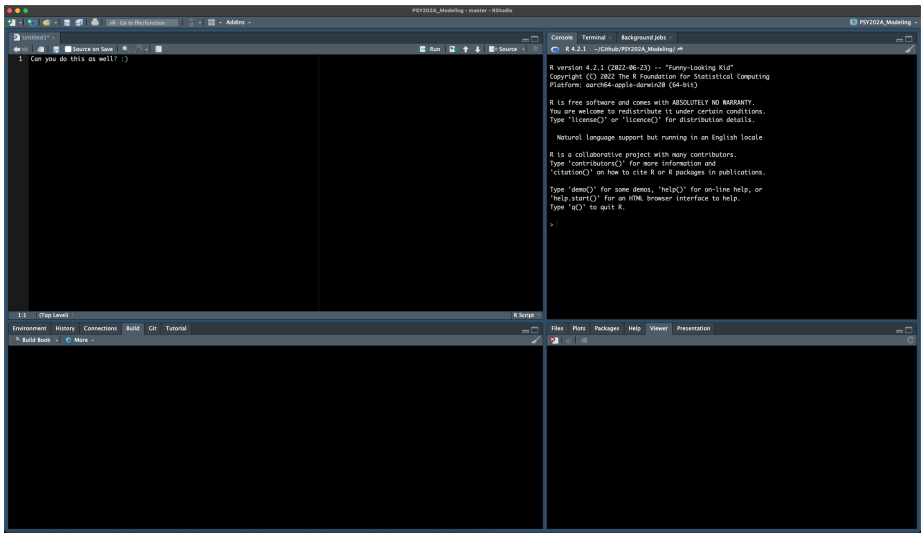


2.2.2 Open a new R script

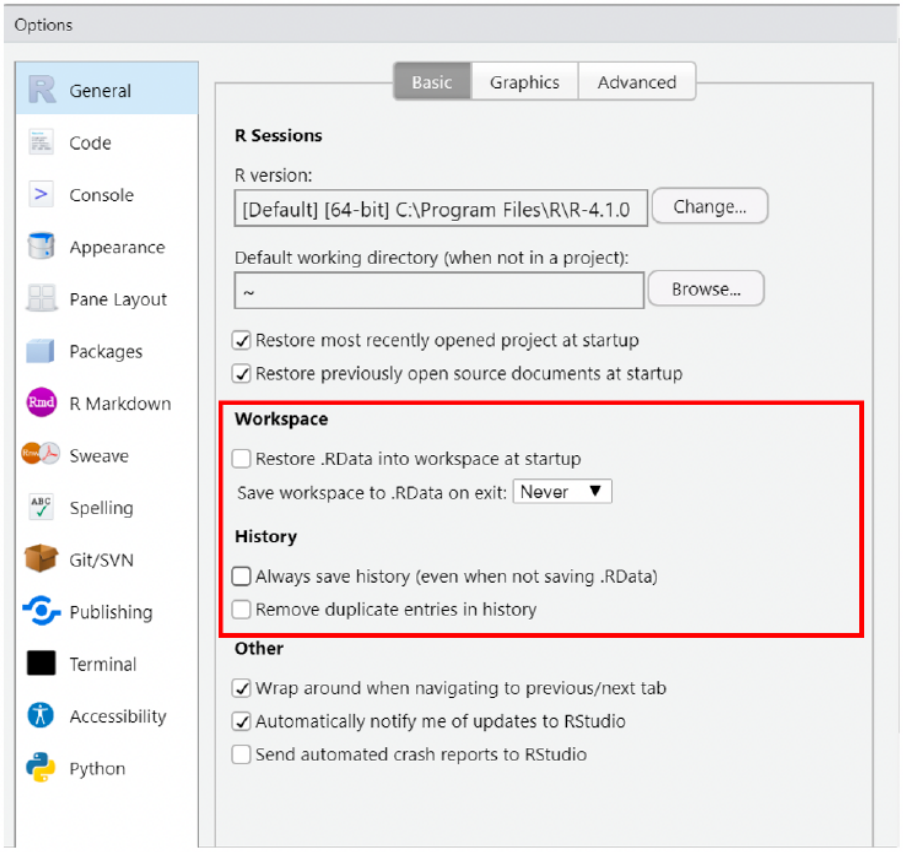
- Can you find four panes?

2.2.3 Individualize R

- Can you find your own R style?


PSY202A_Modeling - master - RStudio
Go to file/function
Addins
PSY202A_Modeling
Untitled1*
Source on Save
Run
Source
1 Can you do this as well? :)
1:1 (Top Level)
R Script
Console
Terminal
Background Jobs
R 4.2.1 · ~/Github/PSY202A_Modeling/
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (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.
>
Environment
History
Connections
Build
Git
Tutorial
Build Book
More
Files
Plots
Packages
Help
Viewer
Presentation

2.2.4 Some useful settings



- Under "Workspace":
  - Uncheck restore .RData into workspace at startup.
  - Save workshop to .Rdata on exit: "Never".
- Under "History":
  - Uncheck "Always save history (even when not saving .RData).
  - Uncheck "Remove duplicate entries in history".

2.3 Interacting with R

- How to run R commands?
  - Put your cursor on a line, or select the line(s), hit the Run button.
  - The hot key for Run the current line'selection is Ctrl + Enter (Windoes) or Command + Return (Mac).
- Can you run the below code?

```
greetings <- "Hello PSY202A"
print(greetings)
```

- Everything starting with a pound sign (#) is considered as a comment. R will skep all the characters after # until it finds a new line of command.
- Can you run the below code?

```
Comment

# Comment
```

2.4 Computation

2.4.1 Using R as a numerical calculator

- Can you run the code below?

```
2 + 4 # Addition
2 - 4 # Subtraction
2 * 4 # Multiplication
2 / 4 # Division
2 ** 4 # Power
sqrt(144) # Square root
log(144) # Natural logarithm
exp(5) # Exponential, i.e., power of e
(10 + 2*log(8) - (exp(8) - 4)/3) * 7
```

- Here, `sqrt()`, `log()`, and `exp()` are numerical functions.

2.4.1.1 Exercise: Numerical calculator

- Can you give me the answer to the mathematical formula below?

$$\frac{26}{\log(5)} \times e^3 + (-0.2)$$

- How about this?

$$-\sqrt{(\ln 7)}$$

2.4.2 Using R as a logical calculator

- You can also use R for logical statements.

```
1 == 1 # equal
1 != 2 # unequal
```

```r
10 > 1 # greater than
10 >= 1 # greater than or equal to
```

2.4.2.1 Exercise: Logical calculator

- Then... can you program your code to evaluate the below statements?
  - Try this first: 5 is lower than or equal to 10. What is the result? Does this make sense?
  - Then, try this second: 10 is not equal to 10. What is the result? Does this make sense?

2.5 Types of objects

- In R, anything with a name is an object.
  - You can give an object with any name you want insofar as that name is not taken in R already.
- An object can contain more than one value and have more complex data strutures, such as:
  - vector: a series of values of the same data type
  - matrix: a two-dimensional table of values with the same data type
  - data frame: a two-dimensional table of values, may not be the same data type
  - list: an object made of objects

2.5.1 Vector: Intro

- The concatenate function `c()` is usually used to put a set of values together.
  - An object `iq` is a vector of 5 IQ scores.

```r
# A vector of IQ scores
iq <- c(90, 100, 105, 110, 95)

# Here, 'iq' is an object

# Print the IQ object
print(iq)
iq
```

- Out of curiosity, what should we do if we want to know the sum of all the five iq scores? You can use the `sum()` function. Can you try?

```r
# Hint: put an object name within the sum()
```

- Many other functions also return vectors as results.
  - For instance, the sequence operator (:) generates consecutive numbers.

```
numbers <- 1:100
numbers
```

2.5.1.1 Exercise: Vector intro

- There are other specific functions. Can you guess what each of the functions does?

```
# Describe what the below function does:
values_1 <- seq(1, 5, by = 1)

# Describe what the below function does:
values_2 <- seq(1, 5, by = 0.5)

# Describe what the below function does:
values_3 <- rep(1, 5)

# Describe what the below function does:
values_4 <- rep(1:5, 3)
```

2.5.2 Vector: Advanced

- There are a couple of distribution functions that can be used to generate a vector of random numbers from a specific distribution.
- For example, you can generate 5 random numbers from a normal distribution with a mean of 0 and a standard deviation of 1:

```
random_normal <- rnorm(5, 0, 1)
random_normal
```

- As another example, you can generate 5 random numbers from a uniform distribution between 0 and 1:

```
random_uniform <- runif(5, 0, 1)
random_uniform
```

- The standard arithmetic operators and functions apply to vectors on a element-wise basis.

```
A <- c(1, 2, 3, 4) - 4
A

B <- c(1, 2, 3, 4)/4
B

C <- c(1, 2, 3, 4)/c(4, 3, 2, 1)
C
```

```
D <- log(c(1, 2, 3, 4))
D
```

- All elements in a vector must be the same type
  - Numeric, character (aka. string), logic

```
numeric_vector <- c(1, 2, 3, 4)

character_vector <- c("developmental", "health", "quantitative")

logical_vector <- c(FALSE, TRUE, FALSE, FALSE)
```

2.5.2.1 Exercise: Vector advanced

Can you run the code below? What do you see?

```
numeric_vector^2

1/numeric_vector

numeric_vector + character_vector

character_vector + logical_vector

numeric_vector + logical_vector

logical_vector + logical_vector

logical_vector+5
```

2.5.3 Matrix: intro

- We can create a matrix by combining multiple vectors (e.g., perhaps associated with multiple scales)

```
scale1 <- c(1, 4, 7, 4, 5)
scale2 <- c(5, 6, 8, 3, 2)
scale3 <- c(0, 9, 8, 9, 3)
```

- We could arrange these into a table, using `cbind()` function (bind by column).

```
my_data <- cbind(scale1, scale2, scale3)
my_data
```

- Or, if our data were in one long vector:

```
long_vec <- c(scale1, scale2, scale3)
my_matrix <- matrix(long_vec, ncol = 3, byrow = FALSE)
my_matrix
```

2.5.4 Matrix: advanced

- Find the size of the matrix (i.e., the number of rows and the number of columns) using `dim()`:

```
dim(my_matrix)
```

- ,where this parallels `length()` for vectors:

```
length(scale1)
```

2.5.4.1 Exercise: matrix advanced

- Can you create an arbitrary matrix with 5 rows and 3 columns? You can do anything to achieve this goal.
- If you are successful, can you transpose the matrix using the `t()` function?
- What is the size of the transposed matrix?

2.5.5 Data frame

- Similar to matrices, data frames can be created by combining multiple vectors, but data frames allow for different vector types (e.g., number, character, logical), but preserves the characteristics of each type.

```
v1 <- 1001:1006
v2 <- c(407.56, 442.20, 385.85, 295.31, 408.10, 280.52)
v3 <- c("A", "A", "A", "B", "B", "B")

my_dataframe <- data.frame(id = v1, reactiontime= v2, condition = v3)
my_dataframe
```

2.5.5.1 Exercise: data frame

- Can you describe the differences between a matrix and a data frame in R?

2.5.6 List

- Data frames must be rectangular, what if our data are non-rectangular?

```
mod_A <- c(4.25, 2.36, 2.37)
mod_B <- c(4.26, 2.45, 2.31, 7.5)
mod_C <- c(4.21, 2.44, 2.29, 7.7, 4.1)
flag <- c(FALSE, FALSE, TRUE)
```

- A list will allow any set of R objects to be combined into a single object

```
my_list <- list(parA = mod_A, parB = mod_C, parC = mod_C, flag = flag)
my_list
```

2.5.7 Exercise: Selecting elements in objects

- We can select elements of vectors, matrices, data frames, and lists using brackets (i.e., []).
- For each chunk of the code below, think of the expected output before running the code, and see if your guess was correct after running the code:

2.5.7.1 For vectors

- Given the vector below:

```
scale1 <- c(1, 4, 7, 4, 5)
```

- Can you expect the outcome?

```
scale1[3]
```

2.5.7.2 For matrices

- Recall that this is our predefined matrix object:

```
my_matrix
```

- Can you expect the outcome?

```
my_matrix[ , 2]
my_matrix[1,  ]
my_matrix[1, 2]
my_matrix[1:2,  ]
my_matrix[ , 1:2]
```

2.5.7.3 For data frames

- Recall that this is our predefined data frame object:

```
my_dataframe
```

- Can you expect the outcome?

```
my_dataframe[1, 1]
my_dataframe[1:2,  ]
my_dataframe[ , 1:2]
my_dataframe$id
```

2.5.7.4 For lists

- Recall that this is our predefined list object:

```
my_list
```

- Can you expect the outcome?
  - Note that, for list objects, we sometimes need to use the double brackets to extract an element of a part of a list

```
my_list[1]
my_list[[1]]
my_list[2]
my_list[[2]]
my_list$parB
```

2.5.7.5 Another bonus exercise for you

- Create a new data frame using the following vectors:

```
id <- c(1001:1040)
x <- runif(40, 250, 500)
y <- x * 0.2 + runif(40, 200, 450)
gender <- sample(c("M", "F"), 40, replace = TRUE)
```

- Get the mean of y in the data frame you just created. (Hint: use $ to extract the variable; use function mean() to compute the mean.)

2.6 Workspace

- Recall that everything that has a name in R is called an object.
- The workspace is your current R working environment and includes every user-defined objects, such as vectors, matrices, data frames, lists, etc.
- The ls() function will list all the objects in a workspace.
- The object(s) can be removed from the workspace by the `rm()` function.

2.6.1 Functions to manage workspace

- Can you run the code below and see what happens? Can you understand what is happening?

```
ls()
```

- What about this? What happened?

```
rm(v1)
ls()
```

- Shall we remove all the objects?

```
# Remove all the objects in the workspace
rm(list=ls(all = TRUE))
```

- If you haven't unchecked the box about Workspace in the Global Options, at the end of an R session, you will be asked whether you want to save an image of the current workspace that is automatically reloaded the next time R is started. Choose NO.
- Note that the workspace consists only of R objects, not of any of the output that you have generated during a session (e.g., images). If you want to save your output, just copy it from the console.
- Also, there are ways to export your data and save them in some specific formats for further uses.

2.7 Working directory

- The working directory is the location (file path) on your computer where R will look for files and where it will save any files.
  - For more details, see https://www.rensvandeschoot.com/tutorials/r-for-beginners/
- To see your current working directory, use `getwd`:

```
getwd()
```

- You can also check the current working directory by looking at bar at the top of the Console pane.
- To change your working directory, use `setwd()`:
  - For Windows users, ## use / instead of
- You can also manually set the working directory in the Files tab.
- If you start an R session from a .R file, the default working directory will be set to where the .R file is located on your computer.

2.8 Importing external data

- The most common way is to use the function `read.table()` to read in .txt files or .dat files; or `read.csv()` for .csv files.

```
mydata1 <- read.table(file = "SAT.dat", header = TRUE)
class(mydata1)
write.csv(mydata1, file = "SAT.csv", row.names = FALSE)
```

- You can try this as well:

```
mydata2 <- read.csv(file = "SAT.csv", header = TRUE, sep = ",")
class(mydata2)
```

- You can also manually choose a file to import
  - But this will not work for all data formats, so be careful

```
mydata3 <- read.table(file.choose(), header = TRUE)
class(mydata3)
```

- To look at the whole dataset, directly type the name of it.

```
mydata1

mydata2

mydata3
```

- To obtain a single variable from the dataset, use a dollar sign ($)

```
mydata1$Math
```

- Can you get the sum and the mean of the math scores using the `sum()` and `mean()` functions?

2.9 Exporting

- Similar to `read.table()`, the most-used expoert function is `write.table()`.
- You can export your data into some different formats than it was originally imported.
- To export the mydata1 into a tab-delimited file without row names:

```
write.table(mydata1, file = "SAT.txt", sep = "\t", row.names = FALSE, col.names = TRUE)
## sep="\t" tells R to use tab as separator in the text file. You can also try sep=" "
```

- Similar to `read.csv()`, there is a function `write.csv()`.

2.10 Final exercise

1. Create a folder named `PSY202A_Lab 1`, move the data file `SAT.csv` in it. Then create a new R script file using the File menu -> New File -> R Script.
2. Change the working directory to `PSY202A_Lab 1`.
3. In the new R script, write R code to examine the data:

- Read in `SAT.csv` file, name it as datSAT;
- Check the first 6 rows of the data set;
- Find the total number of observations in the data set using the `nrow()` function;

- List all the values in the State variable;
- Compute the mean of Verbal scores;
- Create a new variable `Combined`, which equals the sum of Verbal and Math.
  - Hint: assign the sum of two variables to `datSAT$Combined`.

4. Save the data set datSAT (now with a new variable Combined) to your working directory `PSY202A_Lab 1`, and name it as `SATCombined.csv`.
5. Save your R script as `Exercise 1.R` in your working directory `PSY202A_Lab 1` (File menu -> Save).

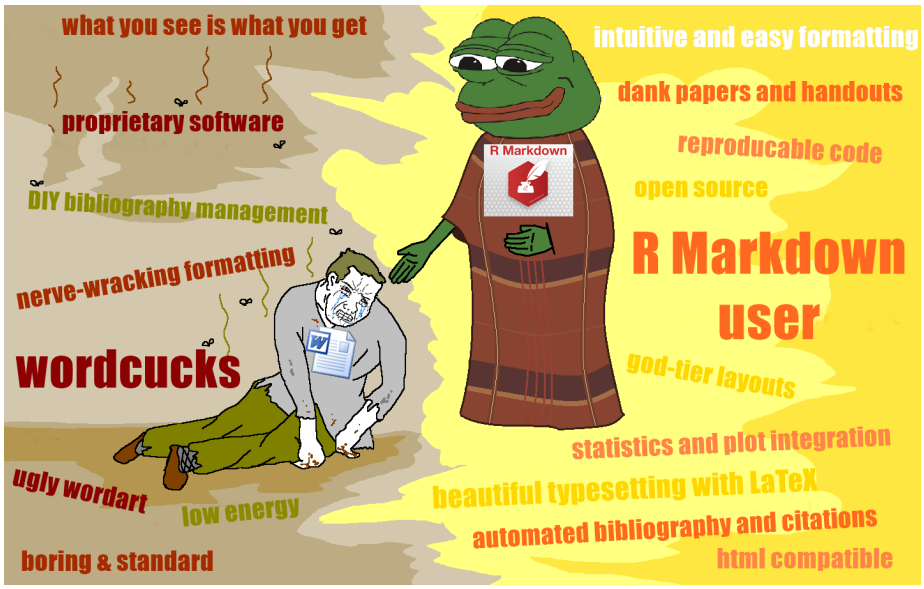
2.11 R Markdown

- Reproducibility is a key philosophical principle in the psychological sciences.
- An easy yet elegant way to ensure reproducibility in R programming is by using R Markdown for documentation.
- I strongly recommend downloading R Markdown before our next meeting. You can find resources at:
  - https://rmarkdown.rstudio.com/lesson-1.html
  - https://vimeo.com/178485416
  - https://yihui.org/tinytex/
- This is a great opportunity to learn how to use R Markdown, which you can then apply when submitting your upcoming homework assignments.

Chapter 3

Introduction to R: Part 2

3.1 What will we do?



3.2 Packages

- A package is a collection of previously programmed functions and data sets, often including functions for specific tasks.
- Some packages come with the base installation of R.
- There are thousands of user-contributed packages that you must manually download and install.
- To see which packages you have, click View -> Show Package.
- To see all available packages: https://cran.r-project.org/web/packages/
- Some packages can be very handy when working with psychological data.
  - The `psych` package is a general-purpose toolbox for analyzing psychological data.
  - The `haven` package can be used to import and export SPSS, Stata, and SAS files.

3.2.1 Installing packages

- The easiest way to install packages and add packages to the base version of R is to use the Tools -> Install Packages...
- Or use `insstall.packages()` function as follows:

```
install.packages("psych")
install.packages("haven")
```

3.2.2 Loading packages

- To access the package you have already installed, load it to the current R session using the function `library()`.

```
library(psych)
library(haven)
```

3.3 Getting help

- There are many ways to getting help for R programming.

3.3.1 Modern way of getting help

- Google is your friend.
- Stack Overflow is your cool friend.
- YouTube is your another wonderful friend.
- AI is your badass friend.

3.3.2 Traditional way of getting help

- When R was installed, HTML format help files were copied on your hard drive.
- To access these files, you can click Help -> R Help.
- Or just type:

```
help.start()
```

- To request an R document for a special function, use "?". To illustrate:

```
?log
# This will pull up an Internet window with everything about the function log()
```

- To request help by keywords, use "??". To demonstrate:

```
??logarithm
# This will give you an information window
# listing all the functions that contain the term "logarithm".
```

- To request help for a specific package, use `help(package = " ")`. That said:

```
help(package = "psych")
```

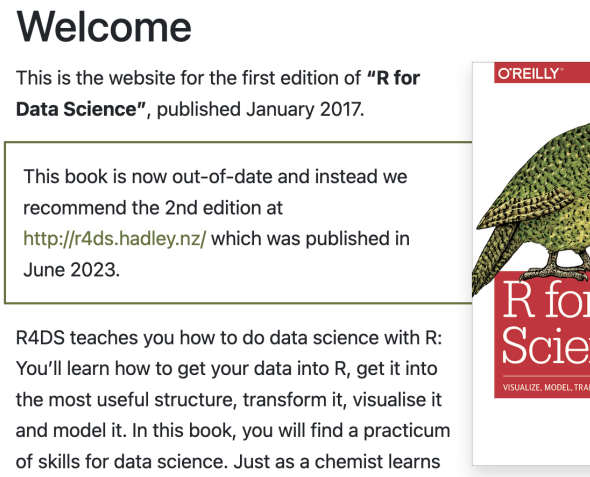
3.4 Tidyverse

- The tidyverse is a collection of R packages for data analysis that are developed with common ideas and norms - the tidyverse style.

- More and more popular in recent years.
- Website: https://www.tidyverse.org/packages
- To install and load the tidyverse packages:

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

- Purpose is to make R code easier to work with:
  - Data manipulation
  - Data visualization
  - Programming
  - Integration with other packages
  - . . . and more
- Book recommended: R for Data Science



  - Freely available at https://r4ds.had.co.nz/

3.4.1 Tidyverse basics

- As it is difficult to change how fundamental base R structures/functions work, the tidyverse suite of packages creates and uses data structures, functions, and operators to make working with data more intuitive.
- The two most basic changes are in the use of pipes and tibbles.

```
# Packages for tidyverse demo
library(datasets)
library(tidyverse)
```

3.4.2 Pipes

- Stringing together commands in R can be quite daunting. Also, trying to understand code that has many nested functions can be confusing.
- To make R code more human readable, the Tidyverse tools use the pipe, `%>%`, which was acquired from the magrittr package and comes installed

automatically with Tidyverse.
- The pipe allows the output of a previous command to be used as input to another command instead of using nested functions.
- Hint: The shortcut to write pipe is `shift + command + M`.

3.4.2.1 Exercise: pipes

- Base R method of running more than one command

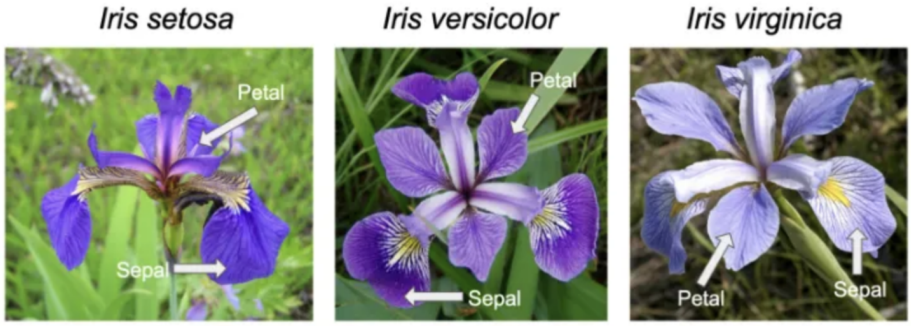
```
sqrt(83)
round(sqrt(83), digit = 2)
```

- Running more than one command with piping

```
sqrt(83) %>% round(digit = 2)
```

3.4.3 Tibbles

- A core component of the tidyverse is the tibble.
- Tibbles are a modern rework of the standard `data.frame`, with some internal improvements to make code more reliable.
- They are data frames but do not follow all of the same rules. For example, tibbles can have column names that are not normally allowed, such as numbers/symbols.
- Tibbles can be created directly using the `tibble()` function or data frames can be converted into tibbles using `as_tibble(name_of_df)`.
- In this section of code, the iris data frame is converted into a tibble. The iris dataset consists of five variables: `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`.




```
# Loading the iris dataset from the datasets package
data("iris")

# Converting the iris data.frame into a tibble object
tibble_iris <- as_tibble(iris)
```

```
# Using a data.frame as a tibble with a pipe operator
iris %>% as_tibble()
```

3.4.4 Differences between tibbles and data.frames

- The main differences between `tibbles` and `data.frames` relate to printing and subsetting.

3.4.4.1 Printing

- A nice feature of a tibble is that when printing a variable to screen, it will show only the first 10 rows and the columns that fit to the screen by default.

```
# Default printing of data.frame
iris # Prints 150 rows

# Default printing of tibble
iris %>%
  as_tibble() # Prints 10 row
```

- This is nice since you don't have to specify head() to take a quick look at your dataset.
- If it is desirable to view more of the dataset, the print() function can change the number of rows or columns displayed.

```
# Printing of tibble with print() - change defaults
iris %>%
  as_tibble() %>%
  print(n = 20, width = Inf)
```

3.4.4.2 Subsetting

- When subsetting base R data.frames the default behavior is to simplify the output to the simplest data structure (i.e., a vector).
- If you use piping to subset a data frame, then the notation is slightly different from base R, requiring a placeholder `.` prior to the `[ ]` or `$`.

```
# Subsetting the Species variable in base R
iris$Species
iris[ ,"Species"]

# Subsetting the Species variable in output using a pipe
iris %>% .$Species
iris %>% .[ ,"Species"]
```

- Note that some older functions do not work with tibbles, so if you need to convert a tibble to a data.frame, the function

as.data.frame(name_of_tibble) will easily convert it.

3.4.5 Tidyverse tools

- Tidyverse has many tools for data wrangling, cleaning, and visualization.

3.4.5.1 dplyr

- Perhaps the most useful tool in the tidyverse is dplyr. It's a Swiss-army knife for data wrangling.
- dplyr has many handy functions:
  - `select()` extracts columns and returns a tibble.
  - `arrange()` changes the ordering of the rows.
  - `filter()` picks cases based on their values.
  - `mutate()` adds new variables that are functions of existing variables.
  - `rename()` easily changes the name of a column(s).
  - `summarise()` reduces multiple values down to a single summary.
  - `pull()` extracts a single column as a vector.
  - `_join()` group of functions that merge two data frames together (e.g., `inner_join()`, `left_join()`, `right_join()`, and `full_join()`).
- Here is an example of the `select()`, `filter()`, and `summarise()` functions using the iris dataset.

```
# Only select the columns related to the Sepal of the iris
iris %>%
  select(Sepal.Length,Sepal.Width) %>%
  head()
```

```
#F ilter for irises with a Sepal.Length greater than 5 and Sepal.Width greater than 4
iris %>%
  filter(Sepal.Length > 5, Sepal.Width > 4)
```

```
# Calculate the average Sepal.Length and Sepal.Width for each iris Species
iris %>%
  group_by(Species) %>%
  summarise(mean.Sepal.Length = mean(Sepal.Length), mean.Sepal.width = mean(Sepal.Width))
```

3.5 Final exercise

1. Subset the `Species` column from the iris dataset using a pipe.

- Hint: Use a `.` before the `[ ]` or `$`.

2. Select the `Petal.Length` and `Petal.Width` columns from the iris dataset.
3. Find the average `Petal.Length` and `Petal.Width` for each iris Species.

3.6 Bonus: R Markdown

- If you have not downloaded the R Markdown yet, please go to https://rmarkdown.rstudio.com/lesson-1.html and download the `rmarkdown` package.
- R Markdown is a tool for reproducible documentation in statistics. R markdown generates a new file that contains selected text, code, and results from the `.Rmd` file.
- The newly created file can be a finished website, PDF document, Word document, slide (beamer, powerpoint, xaringan) show, notebook, handout, book, dashboard, Shiny Apps, package vignette, or more.

3.6.1 How does the R Markdown work?

- According to https://rmarkdown.rstudio.com/lesson-2.html,
  - When you run render, R Markdown feeds the `.Rmd` file to `knitr`, which executes all of the code chunks and creates a new markdown (.md) document which includes the code and its output.
  - The markdown file generated by `knitr` is then processed by `pandoc` which is responsible for creating the finished format.
  - This may sound complicated, but R Markdown makes it extremely simple by encapsulating all of the above processing into a single render function.

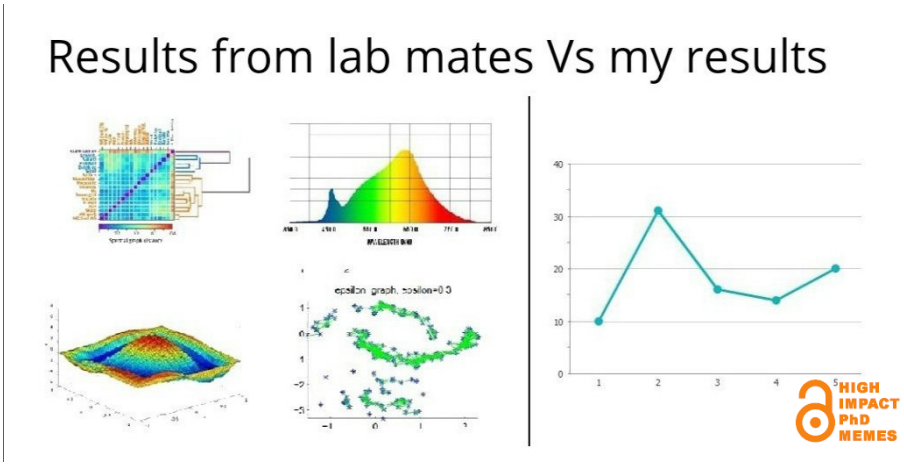
3.6.1.1 Practice

- Generate any PDF document from R Markdown by knitting your markdown file.
- You may need to visit:
  - https://yihui.org/tinytex/
  - https://yihui.org/tinytex/r/#debugging

Chapter 4

Summarizing Data

4.1 Are you ready?



4.1.1 Wait... you guys are getting results???!

4.2 Packages

- Here is a list of packages that we will use today:
  - `modeest` has the `mfv()` function for the mode.
  - `moments` has the `skewness()` and `kurtosis()` functions.
  - `tidyverse` contains the `ggplot2` package for data visualization.
- For those who hope to dive into the fancier data visualization tools, consider:
  - RShiny at https://shiny.posit.co/r/gallery/.
  - Extensions of the `ggplot2` package at https://exts.ggplot2.tidyverse.org/gallery/.

4.3 Numerical methods

4.3.1 Peabody data

- For the purpose of today's demonstration, we will use the following data from 40 children who completed the Peabody Developmental Motor Scales (PDMS).
- The PDMS is a popular standardized test used by physical and occupational therapists for children less than six.
- The variables include peabody scores, gender, and age (in months).

- The dataset is as follows:

```
# Peabody data
peabody <- c(123,99,80,81,90,104,69,86,107,94,82,99,83,98,86,74,67,90,106,102,
             68,80,76,65,69,81,76,68,84,88,68,92,92,97,78,77,94,74,89,68)

gender <- c("M","F","F","F","F","M","F","F","F","F","M","M","M","M","M","M","M",
            "M","M","M","F","F","F","M","F","F","F","M","M","F","M","M","M","M",
            "M","M","M","M","M","M")

age <- c(35,45,38,40,38,42,36,45,34,36,36,51,34,35,37,37,36,36,37,32,22,28,18,20,
         25,18,14,26,14,24,14,26,11,12,26,17,16,23,12,13)
```

- These variables can be stored in a data frame with the following code:

```
# Make a data frame
peabody_df <- data.frame(peabody = peabody, gender = gender, age = age)
```

4.3.2 Central tendency

- Mean

```
mean(peabody)
```

```
## [1] 85.1
```

```
mean(peabody_df$peabody)
```

```
## [1] 85.1
```

- Median

```
median(peabody)
```

```
## [1] 83.5
```

```
median(peabody_df$peabody)
```

```
## [1] 83.5
```

- Mode

```
library(modeest)
mfv(peabody)
```

```
## [1] 68
```

4.3.3 Variability

- Variance

```
var(peabody)
```

```
## [1] 181.1692
```

- Standard deviation

```
sd(peabody)
```

```
## [1] 13.45991
```

4.3.4 Skewness

- There are multiple methods that can be used to calculate skewness in R:
- You could use the formula for skewness.

```
# Define sample size
n <- length(peabody)

# Compute the skewness
(sum((peabody-mean(peabody))^3)/n)/(sum((peabody-mean(peabody))^2)/n)^(3/2)
```

```
## [1] 0.524655
```

- But the `skewness()` function in the `moments` package makes it even easier.

```
library(moments)
```

```
##
## Attaching package: 'moments'
```

```
## The following object is masked from 'package:modeest':
##
##     skewness
```

```
skewness(peabody)
```

```
## [1] 0.524655
```

4.3.5 Kurtosis

- There are multiple methods for calculating kurtosis in R.
- You can use the formula:

```
n <- length(peabody)
(sum((peabody-mean(peabody))^4)/n)/(sum((peabody-mean(peabody))^2)/n)^2
```

```
## [1] 2.911796
```

- Or you can use the `kurtosis()` function in the `moments` package.

```
kurtosis(peabody)
```

```
## [1] 2.911796
```

4.3.6 Using the tidyverse package

- tidyverse allows us to use the `summarise()` function.
  - which allows us to get various descriptive statistics such as minimum, maximum, mean, median, variance, standard deviation, skewness, kurtosis, etc.
- As a small exercise, try to run the code below. What do you see?

```
summarise(peabody_df,
          mean.peabody = mean(peabody),
          sd.peabody = sd(peabody),
          skew.peabody = moments::skewness(peabody),
          kurt.peabody = moments::kurtosis(peabody))
```

4.4 Visual methods

- Be sure to refer to the cheatsheet I have attached on CatCourses!
- Every useful piece of information is there!

4.4.1 Using the tidyverse package

- The `ggplot2` package is a core component of the tidyverse that allows us to build data visualizations.
- ChatGPT is so good at programming `ggplot2` code for you. But of course, sometimes it itself does produce wrong code, particularly when you want to do complicated visualizations.
  - It is suggested to know the fundamental grammar to be a smart user of the AI!
- The idea behind `ggplot2` is that every new concept we introduce will be layered on top of the information we've already learned.
- In this way, `ggplot2` uses layers of information added on top of each other to help build the graph.
  - This is most evidenced by how each line is separated by a plus sign (+). Each line of code is a different layer of the graph.
- You'll see what I mean as we go through several example graphs.
- Remember to work with a tibble instead of a data frame. Tibbles work much better in tidyverse.

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
```

```
peabody_tib <- as_tibble(peabody_df)
peabody_tib
```

```
## # A tibble: 40 x 3
##    peabody gender   age
##      <dbl> <chr>  <dbl>
##  1     123 M         35
##  2      99 F         45
##  3      80 F         38
##  4      81 F         40
##  5      90 F         38
##  6     104 M         42
##  7      69 F         36
##  8      86 F         45
##  9     107 F         34
## 10      94 F         36
## # i 30 more rows
```

- The type of plot you want to make is referred to as a `geom`. There are dozens of `geom` options available in `ggplot2`. For todays' lesson, we will focus on four specific `geoms`:
  - `geom_boxplot()`
  - `geom_histogram()`
  - `geom_bar()`
  - `geom_point()`
- When using `ggplot2`, plots will take the following general form:

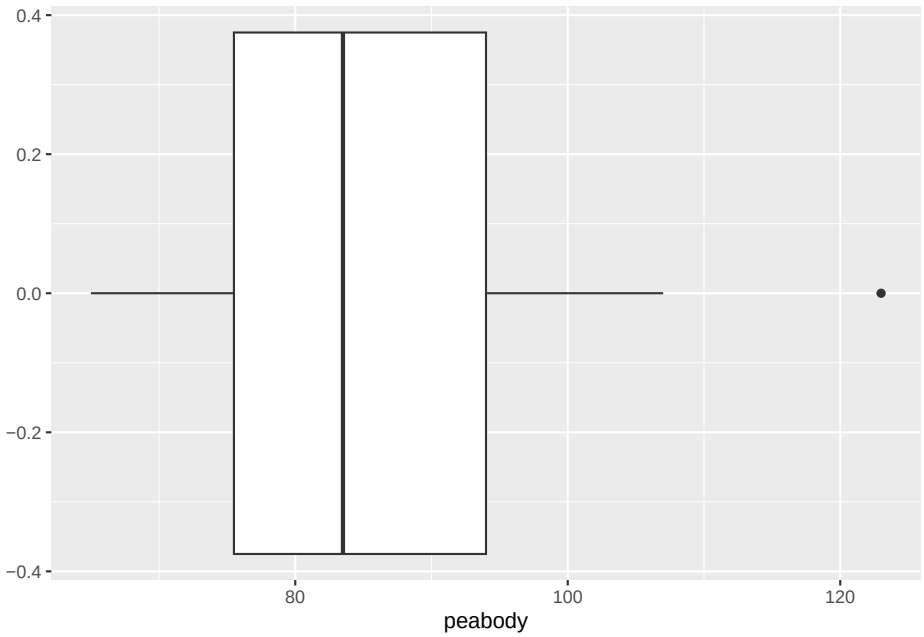
```
ggplot(data = DATASET, aes(VARIABLE(S))) +
  geom_PLOT_TYPE()
```

- First, you start with the `ggplot()` function, where you will specify the data.
- Second, you will include the `aes` argument, which is where you specify the variable(s) you want to plot.
- Third, you select the geom type (e.g., `geom_boxplot()`) you are interested in plotting. This is also where you can begin to customize your plots.

4.4.2 Boxplot

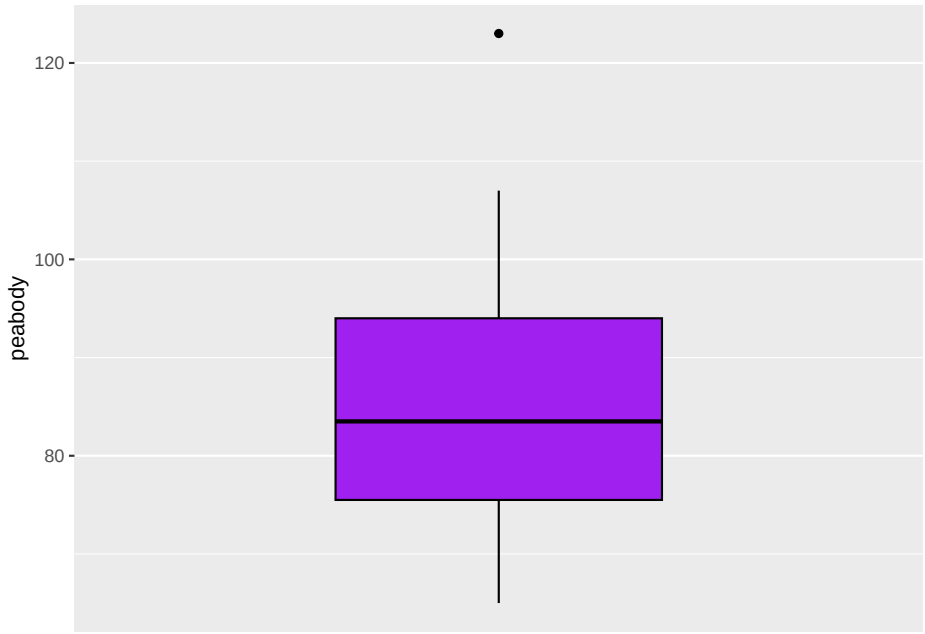
- Boxplots are a useful method for presenting the dispersion of the data, the symmetry of the distribution, and the potential outliers (or extreme scores).
- The following code provides a basic `ggplot2` setup for a boxplot of the peabody scorees:

```
ggplot(data=peabody_tib, aes(peabody)) +
  geom_boxplot()
```



- By adding another layer, you can further customize your plot. In the code below, I illustrate how you can change the coloring of the boxplot and remove the tick marks on the x-axis.

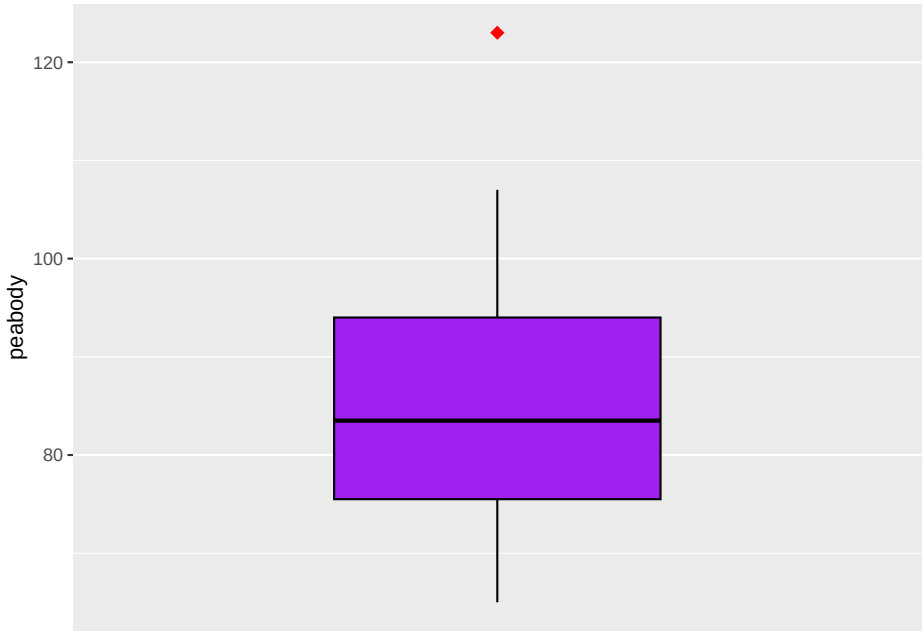
```
ggplot(data = peabody_tib, aes(y=peabody)) +
  geom_boxplot(fill = "purple", colour = "black") +
  scale_x_discrete( ) ## removes x-axis ticks
```



- You can also change the shape and color of the outlier.

- All you have to do is add another line of code that specifies the outlier color, shape, and size.

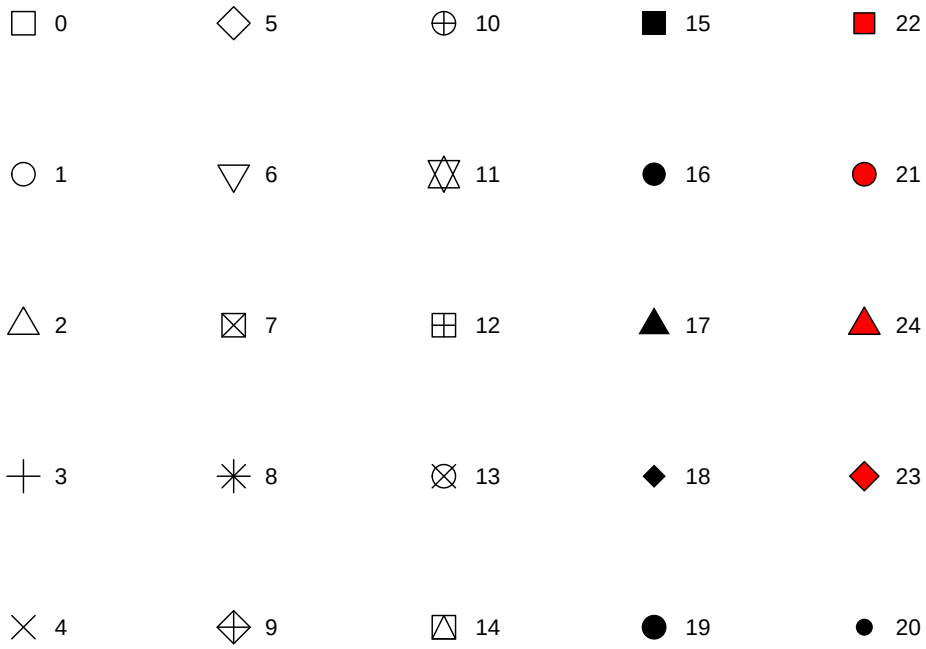
```
ggplot(data = peabody_tib, aes(y=peabody)) +
  geom_boxplot(fill = "purple", color = "black",
               outlier.color = "red", outlier.shape = 18, outlier.size = 3) +
  scale_x_discrete( ) ## removes x-axis ticks
```



- There are several available shapes in `ggplot2`.
  - I provide some examples below, but you can see more here: https://ggplot2.tidyverse.org/articles/ggplot2-specs.html

```
shapes <- data.frame(
  shape = c(0:19, 22, 21, 24, 23, 20),
  x = 0:24 %/% 5,
  y = -(0:24 %% 5)
)

ggplot(shapes, aes(x, y)) +
  geom_point(aes(shape = shape), size = 5, fill = "red") +
  geom_text(aes(label = shape), hjust = 0, nudge_x = 0.15) +
  scale_shape_identity() +
  expand_limits(x = 4.1) +
  theme_void()
```



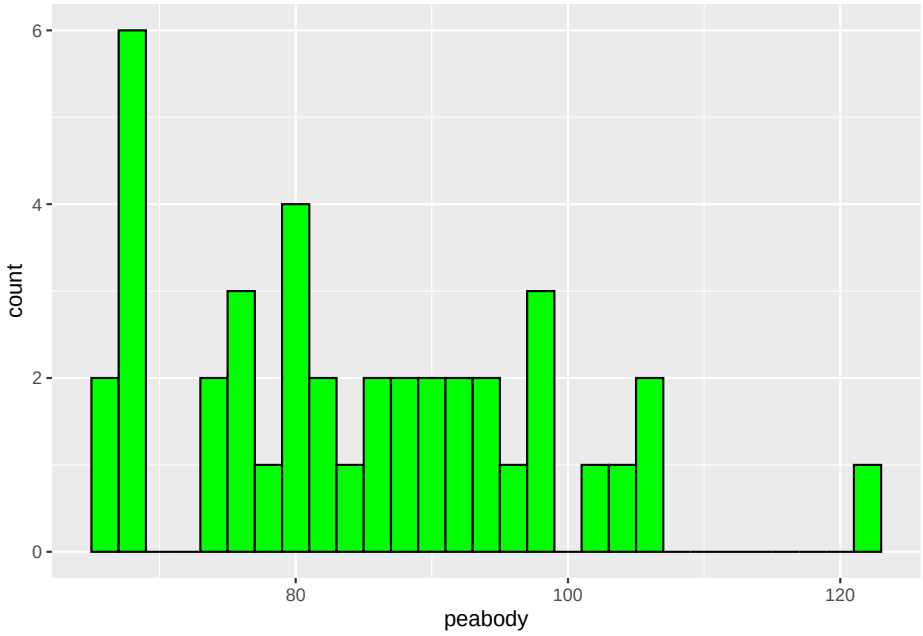
4.4.3 Histograms

- Histograms are a useful method for presenting the dispersion of the data and the symmetry of the distribution.

- The following code provides the setup for a histogram of the peabody scores:

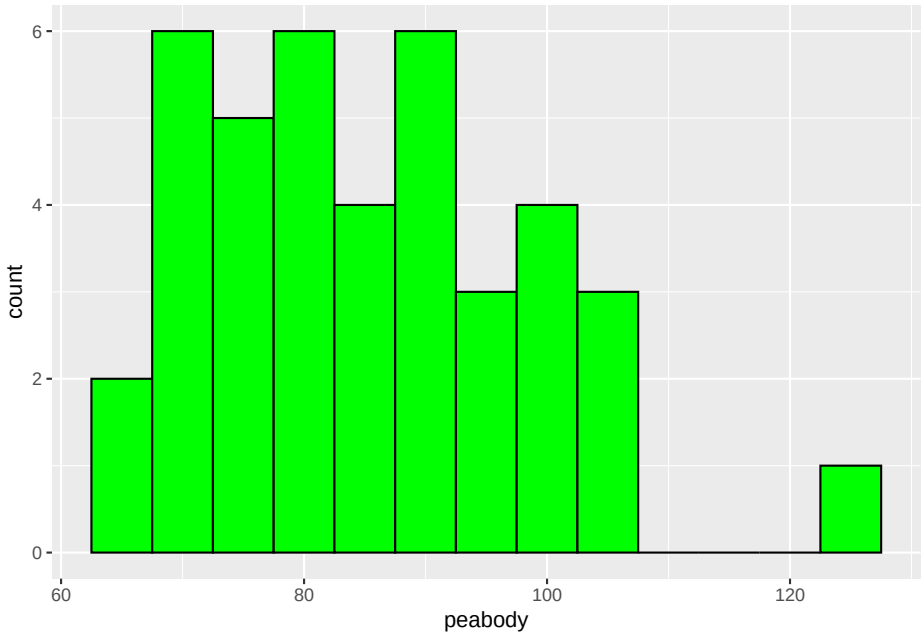
```
ggplot(data = peabody_tib, aes(peabody)) +
  geom_histogram(color = "black", fill = "green")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- You can change the bindwidth of the histogram using the following code:

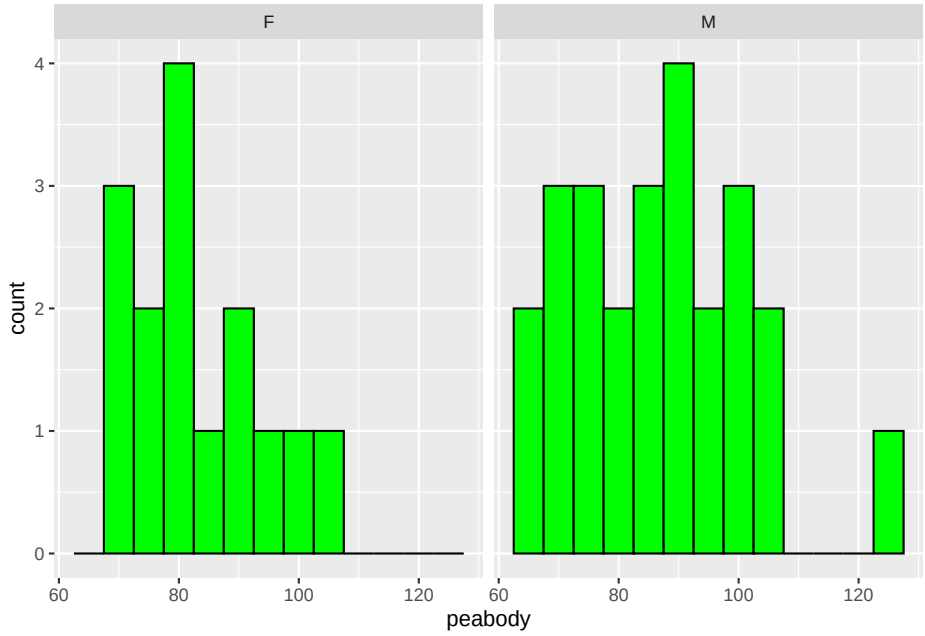
```
ggplot(data = peabody_tib, aes(peabody)) +
  geom_histogram(color = "black", fill = "green",
                 binwidth = 5)
```



- Alternatively, you could set the total number of bins with the `bins` argument.

- Sometimes you want to display plots for a subset of your data. This can be accomplished using the `facet_wrap()` function.

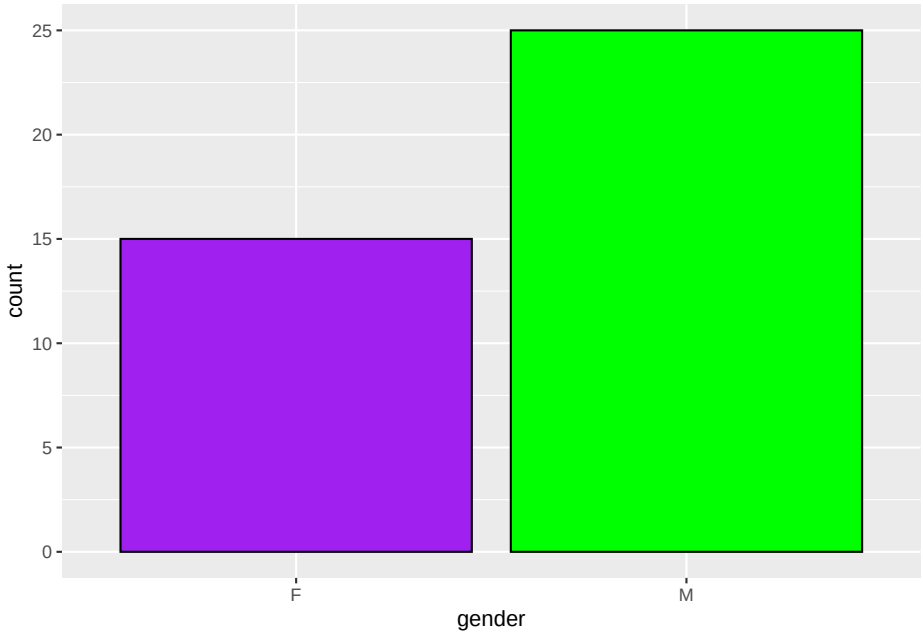
```
ggplot(data = peabody_tib, aes(peabody)) +
  geom_histogram(color = "black", fill = "green",
                 binwidth = 5) +
  facet_wrap(~gender, nrow = 1)
```



4.4.4 Bar charts

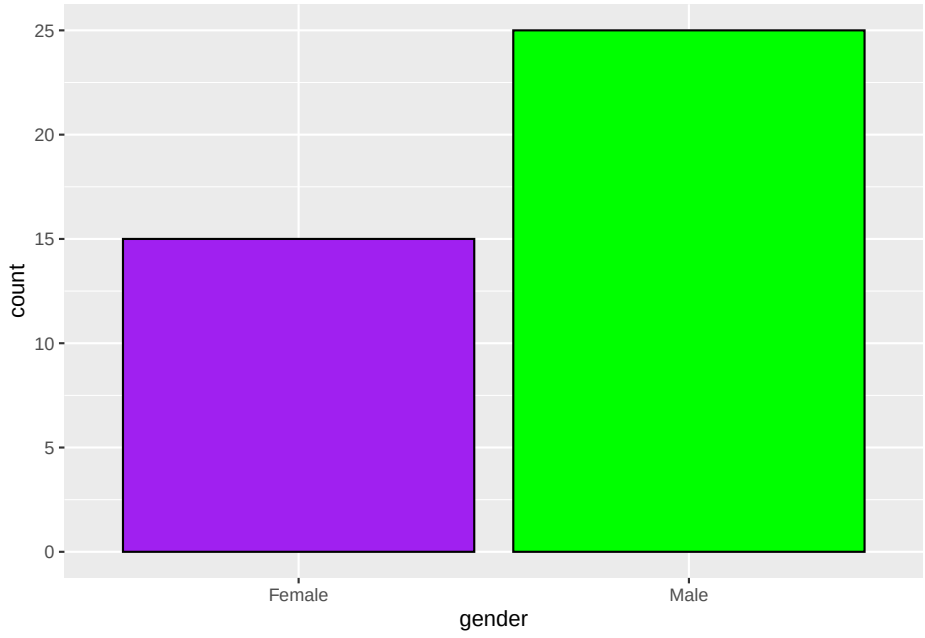
- Bar charts are a useful method for displaying categorical data. They can show the number of cases in each of the categories.

```
ggplot(data = peabody_tib, aes(gender)) +
  geom_bar(color = "black", fill = c("purple","green"))
```



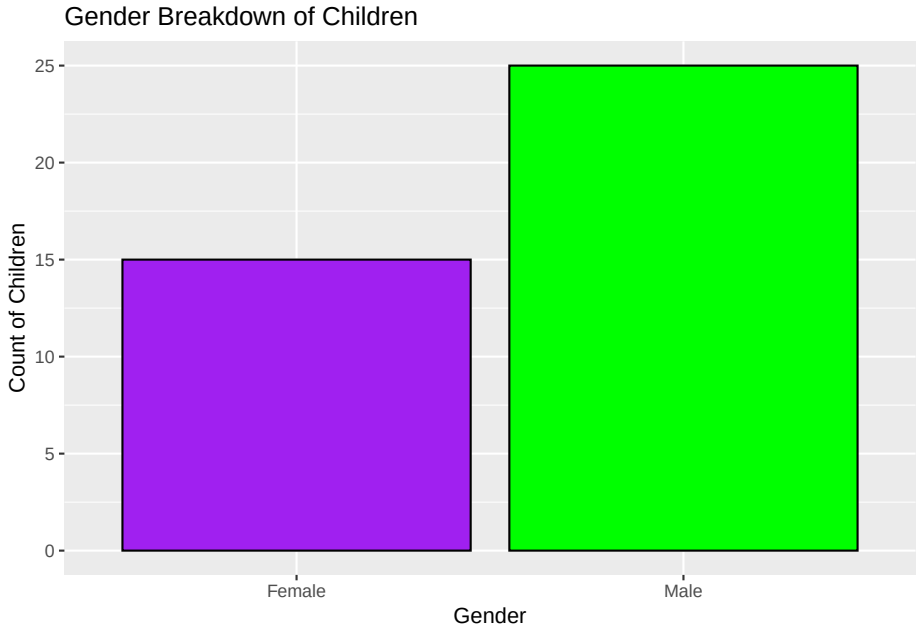
- Perhaps you would like to change the `F` to Female and the `M` to Male. The x-axis ticks can be changed using the following trick:

```
ggplot(data = peabody_tib, aes(gender)) +
  geom_bar(color = "black", fill = c("purple","green")) +
  scale_x_discrete(labels = c("Female", "Male")) ## Changes the labels on the x-axis ticks
```



- Sometimes it can be helpful to change the plot title, x-axis and y-axis.

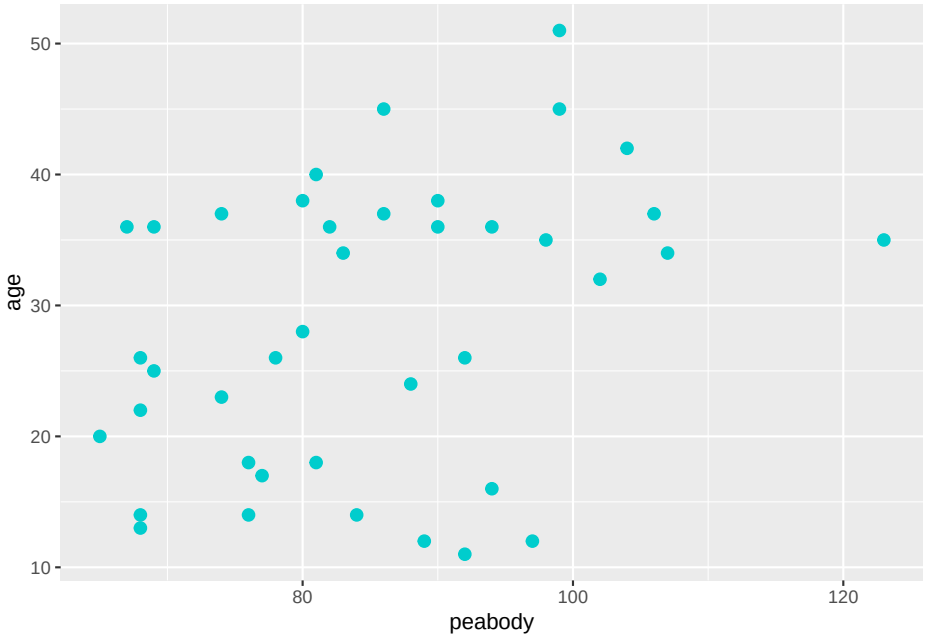
```
ggplot(data = peabody_tib, aes(gender)) +
  geom_bar(color = "black", fill = c("purple","green")) +
  scale_x_discrete(labels = c("Female", "Male")) +
  labs(x='Gender',
       y='Count of Children',
       title='Gender Breakdown of Children') ## Allows you to change the title and axi
```



4.4.5 Scatterplots

- Scatterplots are a useful method for visualizing the relationship between two numerical variables. We will not discuss how to interpret the scatterplot today, but I thought it might be helpful for you to see how the `geom_point()` function works.

```
ggplot(data = peabody_tib, aes(peabody,age)) +
  geom_point(shape = 19, size = 2.5, color = "cyan3")
```



4.4.6 Important points about data visualization

- At its core, the term 'data visualization' refers to any visual display of data that helps us understand the underlying data better.
- Generally, there are a few charactersitics of all godo plots [Note: This is not an exhaustive list].
  - Clearly-labeled axes.
  - Text that are large enough to see.
  - Axes that are not misleading.
  - Data that are displayed appropriately considering the type of data you have.

4.5 Final exercise

- Using the age variable, find the following summary statistics:
  - mean
  - median
  - standard deviation
  - skewness
  - kurtosis
- Using the age variable, create a histogram with the following features:
  - colored bars.

- 10 bins [Hint: use `bins` argument].
- a title of "Age of Children (in months)" [Hint: use `labs()` function].

Chapter 5

Regression and ANOVA

5.1 Your first Ph.D. statistics course is almost done

- Keep up the excellent work! You're nearly there!


Ready for PSY202A!
No More PSY202A...
imgflip.com

5.1.1 Regression vs. ANOVA

- In general, remember that both regression and ANOVA are part of the broader class of general linear models.
- The term 'general linear model' refers to conventional linear regression models for a continuous response variable given continuous and/or categorical predictors.
  - For your information, the term 'generalized linear model' refers to a larger class of models that is assumed to follow an exponential family distribution. See https://online.stat.psu.edu/stat504/lesson/6/6.1.
  - For your additional information, the term 'generalized linear mixed model' refers to an even further extension of general linear models that permis random effects as well as fixed effects in the linear predictor. See https://online.stat.psu.edu/stat504/lesson/generalized-linear-mixed-models.
- Out of curiosity... are you familiar with ANCOVA...?

5.2 What Do We Do Today?

5.2.1 Goals

- HW7 (one-way ANOVA) and HW8 (factorial ANOVA) are upcoming. Therefore, we will first discuss ANOVA, focusing on helping you succeed with these assignments.
- We will also discuss regression analysis. Let's approach it with a relaxed and review-oriented mindset.

5.2.2 Packages

- Initial setting

```
# Clean the workspace
rm(list=ls());gc()
```

```
##           used (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells 1444103 77.2    2204462 117.8         NA  2204462 117.8
## Vcells 2532237 19.4    8388608  64.0      16384  4387657  33.5
```

```
# Set the working directory
setwd("/Users/ihnwhiheo/Github/PSY202A_Modeling")
getwd()
```

```
## [1] "/Users/ihnwhiheo/Github/PSY202A_Modeling"
```

- Let's load R packages that will be used throughout.

```
# List of packages
packages <- c("psych", "QuantPsyc", "car", "apaTables", "report", "multcomp",
              "palmerpenguins", "tidyverse", "ggformula", "emmeans") # Add the package

# Check, install, and load each package
for (package_name in packages) {
  if (!requireNamespace(package_name, quietly = TRUE)) {
    install.packages(package_name, dependencies = TRUE)
  }
  library(package_name, character.only = TRUE)
}
```

```
## Warning: package 'ggformula' was built under R version 4.2.3
```

```
## Warning: package 'scales' was built under R version 4.2.3
```

```
## Warning: package 'ggridges' was built under R version 4.2.3
```

- We will be using with the `sat.act` dataset in the R `psych` package.
  - The dataset consists of self-reported scores of 700 subjects on the SAT Verbal, SAT Quantitative and ACT. Additional variables include age, gender (0=female, 1=male), and education ().

```
# Load data
data(sat.act)
dat <- sat.act
```

- In the original dataset, gender is coded as 1 for males and 2 for females.
- The education variable is categorical, ranging from 1 (high school) to 5 (graduate work). For the variables coded as 0, I could not find the exact coding scheme, so we assume that 0 refers to no education.

- While the other variables are continuous and require no data preprocessing, we will perform some quick data engineering for gender and education.
- Specifically, we will recode gender so that females (originally coded as 2) become the reference category by assigning them a value of 0. Additionally, we will treat the education variable as categorical.

```
# Data engineering
dat$gender_original <- dat$gender
dat$gender <- ifelse(dat$gender == 1, 1, 0)
dat$education <- as.factor(dat$education)

# Quick overview of the data
head(dat, 20)
```

```
##       gender education age ACT SATV SATQ gender_original
## 29442      0         3  19  24  500  500               2
## 29457      0         3  23  35  600  500               2
## 29498      0         3  20  21  480  470               2
## 29503      1         4  27  26  550  520               1
## 29504      1         2  33  31  600  550               1
## 29518      1         5  26  28  640  640               1
## 29527      0         5  30  36  610  500               2
## 29529      1         3  19  22  520  560               1
## 29543      0         4  23  22  400  600               2
## 29547      0         5  40  35  730  800               2
## 29578      1         3  23  32  760  710               1
## 29592      0         4  34  29  710  600               2
## 29617      1         4  32  21  600  600               1
## 29619      0         4  41  35  780  725               2
## 29633      0         3  20  27  640  630               2
## 29671      0         4  24  27  640  590               2
## 29676      0         3  19  33  640  650               2
## 29685      0         4  24  32  700  620               2
## 29710      1         4  35  28  640  580               1
## 29711      0         4  46  32  610  680               2
```

- Shall we look at the descriptive statistics?

```
# Descriptive statistics
describe(dat)
```

```
##                vars   n   mean     sd median trimmed    mad min max range
## gender            1 700   0.35   0.48      0    0.32   0.00   0   1     1
## education*        2 700   4.16   1.43      4    4.31   1.48   1   6     5
## age               3 700  25.59   9.50     22   23.86   5.93  13  65    52
## ACT               4 700  28.55   4.82     29   28.84   4.45   3  36    33
## SATV              5 700 612.23 112.90    620  619.45 118.61 200 800   600
```

```
## SATQ               6 687 610.22 115.64    620  617.25 118.61 200 800   600
## gender_original    7 700   1.65   0.48      2    1.68   0.00   1   2     1
##                   skew kurtosis   se
## gender            0.61    -1.62 0.02
## education*       -0.68    -0.07 0.05
## age               1.64     2.42 0.36
## ACT              -0.66     0.53 0.18
## SATV             -0.64     0.33 4.27
## SATQ             -0.59    -0.02 4.41
## gender_original -0.61    -1.62 0.02
```

- For categorical variables (i.e., gender and education), let's make a table and view the frequencies.

```
# Descriptive statistics
tab <- table(Gender = dat$gender, Education = dat$education)
addmargins(tab)
```

```
##        Education
## Gender   0   1   2   3   4   5 Sum
##    0    30  25  21 195  87  95 453
##    1    27  20  23  80  51  46 247
##    Sum  57  45  44 275 138 141 700
```

5.3 One-way ANOVA

5.3.1 Assumptions

Trailer: You will literally enjoy assumptions in PSY202B with Sarah!

1. Normality: We assume that the populations from which the samples were taken are normally distributed.

- Numerically, we can check the skewness and kurtosis of the variables. For instance,

```
# Skewness and kurtosis
skew(dat$ACT) # Little bit negatively skewed (if 0, no skew; if positive, positively s
```

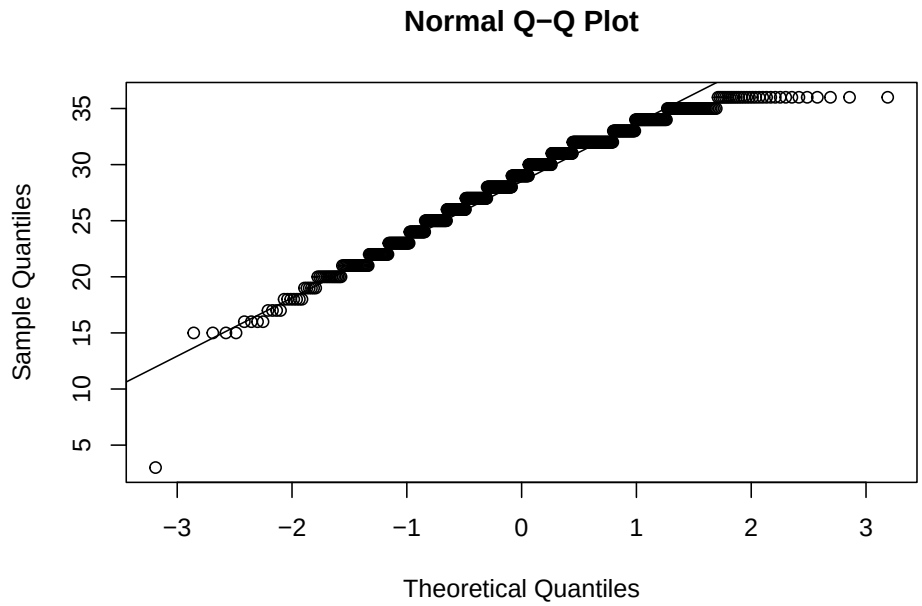
```
## [1] -0.6564026
```

```
kurtosi(dat$ACT) # Little bit leptokurtic (if 0, mesokurti;, if negative, platykurtic)
```

```
## [1] 0.5349691
```

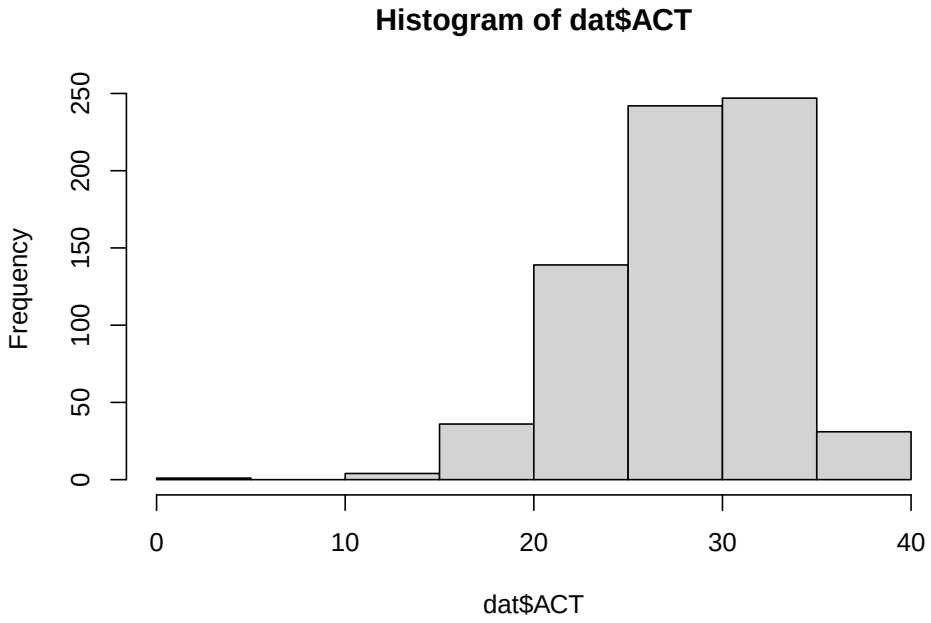
- Visually, we can check the QQ plot. For instance:

```
# QQ plot
qqnorm(dat$ACT)
qqline(dat$ACT)
```



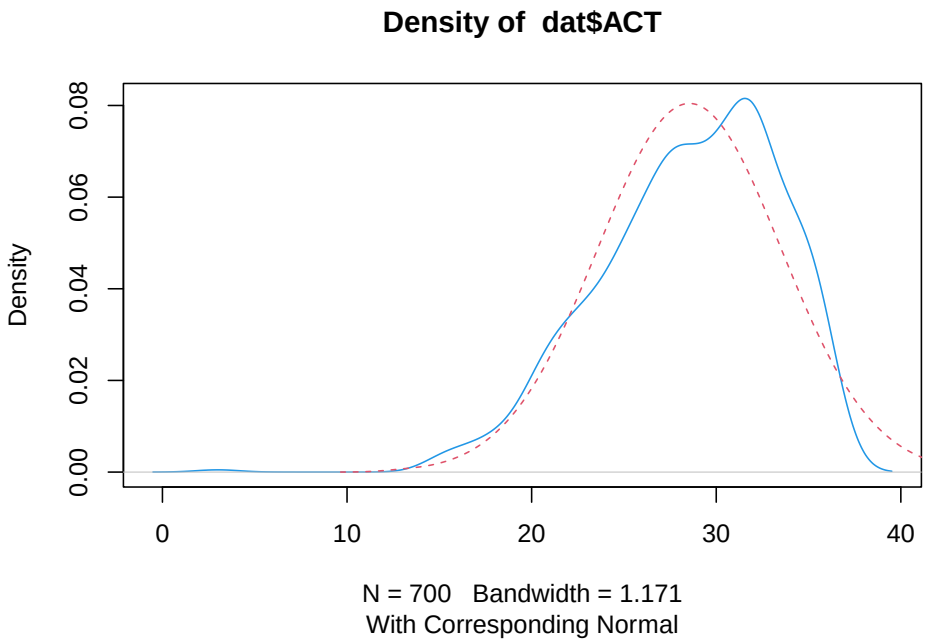
- Or simply we can check the histogram.

```
# Histogram
hist(dat$ACT)
```



- Or we can examine the density plot to compare the distribution of the variable against the ideal normal distribution.

```
# Density plot
plotNormX(dat$ACT)
```



2. Random sampling: We assume that the data we measure were obtained from a sample that we selected using a random sampling procedure.

- We cannot statistically test whether random sampling has occurred. Think about the data collection method you used.

3. Independence: We assume that the probabilities of each measured outcome in a study are independent or equal.

- For the independence assumption, we cannot test this statistically. Just think about the data collection method you used.

4. Homogeneity of variance (i.e., homoscedasticity): We assume that the samples were drawn from populations of equal variances.

- Numerically, we can conduct Barlett's or Levene's test.
  - If significant, the assumption of homogeneity of variances is violated.

```
# Bartlett's test
bartlett.test(ACT ~ gender, data = dat)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  ACT by gender
```

```
## Bartlett's K-squared = 1.9164, df = 1, p-value = 0.1663
```

```
bartlett.test(ACT ~ education, data = dat)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  ACT by education
## Bartlett's K-squared = 21.624, df = 5, p-value = 0.0006172
```

```
# Levene's test
leveneTest(dat$ACT, group = dat$gender)
```

```
## Warning in leveneTest.default(dat$ACT, group = dat$gender): dat$gender coerced
## to factor.
```

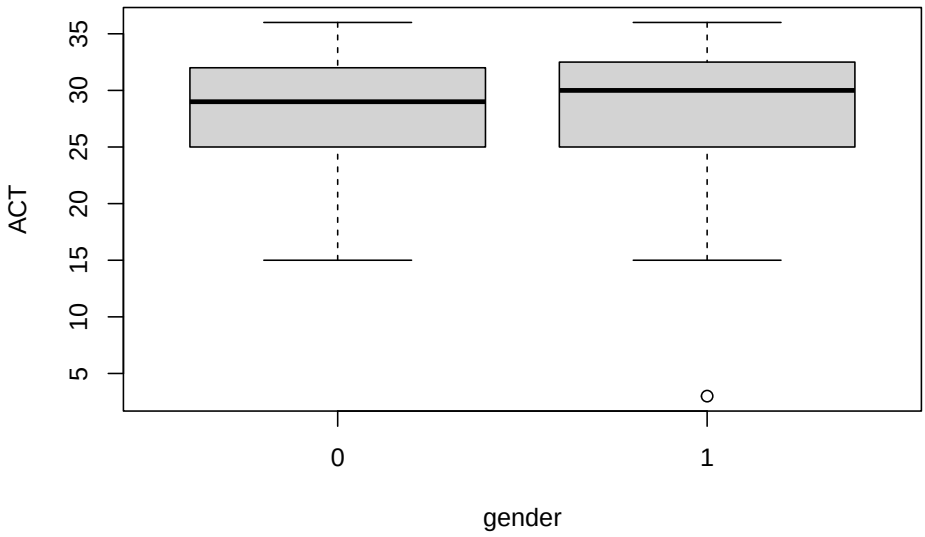
```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.2899 0.5904
##       698
```

```
leveneTest(dat$ACT, group = dat$education)
```

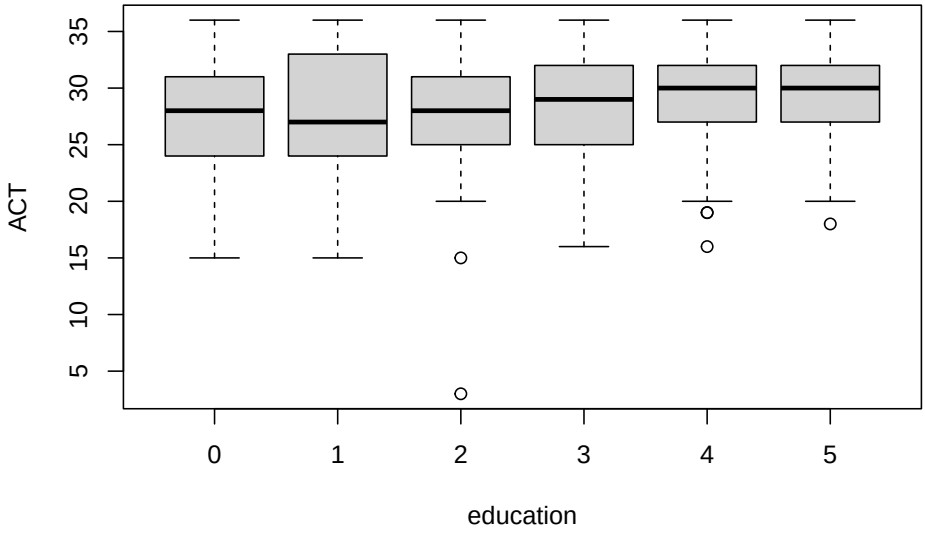
```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   5   3.279 0.006194 **
##       694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Visually, we can look at the box plots

```
# Boxplots
boxplot(ACT ~ gender, data = dat)
```




```r
boxplot(ACT ~ education, data = dat)
```



5.3.2 Analysis

- Let's conduct a one-way ANOVA to test whether there are significant mean differences in ACT scores across the different education levels.

```r
# Perform a one-way ANOVA
model_oaov <- aov(ACT ~ education, data = dat)

# Print the output
summary(model_oaov)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## education     5    470   93.90   4.126 0.00106 **
## Residuals   694  15794   22.76
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Make the APA-style table
apa.aov.table(model_oaov)
```

```
##
##
## ANOVA results using ACT as the dependent variable
##
##
##    Predictor       SS  df       MS       F    p partial_eta2 CI_90_partial_eta2
##  (Intercept) 43023.79   1 43023.79 1890.50 .000
##    education   469.50   5    93.90    4.13 .001          .03         [.01, .05]
##        Error 15793.94 694    22.76
##
## Note: Values in square brackets indicate the bounds of the 90% confidence interval for partial
```

```
# Describe the result following the APA-style; but for reference ONLY!
report(model_oaov)
```

```
## The ANOVA (formula: ACT ~ education) suggests that:
##
##   - The main effect of education is statistically significant and small (F(5,
## 694) = 4.13, p = 0.001; Eta2 = 0.03, 95% CI [7.33e-03, 1.00])
##
## Effect sizes were labelled following Field's (2013) recommendations.
```

5.3.3 Can you explain to me how those results have been obtained?

- I'll leave this as food for thought for you!

5.3.4 How can we interpret the results?

- According to the ANOVA summary table, the omnibus F-test is statistically significant, as the p-value is below the alpha level of 0.05. This indicates that the null hypothesis of equal population means across the six education groups is rejected. Consequently, we conclude that the six education levels are associated with different ACT scores.

5.3.5 Given the results, we are now interested in testing the differences among all the pairs of the means. Because using multiple t-tests can inflate familywise error rates, we will use one of the correction methods. What should we do?

- Note that Bonferroni and Holm-Bonferroni are a priori comparison methods. This means they are used when we have a planned set of comparisons before collecting data.
  - In our case, we did not have such a plan (or did you have something in mind? If then, you are amazing!). Therefore, we will use either Fisher's LSD or Tukey's HSD for post hoc comparisons.
- Fisher's LSD is typically used when there are only three means to compare. This is because, with three means, the family-wise error rate remains controlled at the significance level of alpha.
  - However, when comparing more than three means, the overall F-test does not adequately protect the family-wise error rate if the complete null hypothesis is not true but a subset of it is.
- Tukey's HSD, on the other hand, is specifically designed to perform all possible pairwise comparisons. It ensures that the researcher has only a 5% chance of finding a significant mean difference when the null hypothesis is true.
  - In the current example, there are 15 comparisons to be made (Can you understand why? If yes, please explain that to me!). Thus, Tukey's HSD is the appropriate method among the four correction techniques.

```
# Tukey's HSD
TukeyHSD(model_oaov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ACT ~ education, data = dat)
##
## $education
##            diff         lwr      upr     p adj
## 1-0  0.01520468 -2.70335124 2.733761 1.0000000
## 2-0 -0.49641148 -3.23217649 2.239354 0.9954530
## 3-0  0.82086124 -1.16316374 2.804886 0.8453156
## 4-0  1.78718535 -0.35927170 3.933642 0.1648941
## 5-0  2.12915267 -0.01061924 4.268925 0.0520133
## 2-1 -0.51161616 -3.40192518 2.378693 0.9959541
## 3-1  0.80565657 -1.38656404 2.997877 0.9005816
## 4-1  1.77198068 -0.56826588 4.112227 0.2562269
## 5-1  2.11394799 -0.22016851 4.448064 0.1015015
## 3-2  1.31727273 -0.89625276 3.530798 0.5317982
```

```
## 4-2  2.28359684 -0.07661879 4.643812 0.0644453
## 5-2  2.62556415  0.27142658 4.979702 0.0186900
## 4-3  0.96632411 -0.45584423 2.388492 0.3775325
## 5-3  1.30829142 -0.10376689 2.720350 0.0874803
## 5-4  0.34196731 -1.29046383 1.974398 0.9911202
```

```
# Test whether the adjusted p-values are lower than 0.05
TukeyHSD(model_oaov)$education[,4] < 0.05
```

```
##   1-0   2-0   3-0   4-0   5-0   2-1   3-1   4-1   5-1   3-2   4-2   5-2   4-3
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##   5-3   5-4
## FALSE FALSE
```

- The adjusted p-values based on Tukey's HSD indicate a significant difference when comparing the means of group 2 and group 5. Therefore, the group means for 2 and 5 are significantly different.

5.3.6 Something more about a priori contrasts and post-hoc comparison

- For our valuable learning experiences, we can also use other packages for multiple comparisons.
- Let's start with the usage of the `multcomp` package.
- The `glht` function is used to create a test for multiple contrasts. The contrasts can be all-possible pairwise comparisons, which is referred to as "Tukey" in this package. Then, the result of multiple contrasts can be summarized and adjusted for familywise error rate.
- The Bonferroni method is used in this example. Note that if the test argument is not specified, the single-step approach is used. The adjusted p values is computed from the joint normal or t distribution of the z statistics such that the p value represents the probability of getting at least one significant result by chance if all z or t values are the same in all contrasts. The Tukey method and the single-step approach will provide the same results if the group sizes are equal.

```
# Multiple comparisons
pairwise <- glht(model_oaov, linfct = mcp(education = "Tukey"))
summary(pairwise, test = adjusted(type = "bonferroni"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
```

```
## Fit: aov(formula = ACT ~ education, data = dat)
##
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)
## 1 - 0 == 0   0.0152     0.9513   0.016   1.0000
## 2 - 0 == 0  -0.4964     0.9573  -0.519   1.0000
## 3 - 0 == 0   0.8209     0.6943   1.182   1.0000
## 4 - 0 == 0   1.7872     0.7511   2.379   0.2642
## 5 - 0 == 0   2.1292     0.7488   2.844   0.0689 .
## 2 - 1 == 0  -0.5116     1.0114  -0.506   1.0000
## 3 - 1 == 0   0.8057     0.7671   1.050   1.0000
## 4 - 1 == 0   1.7720     0.8189   2.164   0.4623
## 5 - 1 == 0   2.1139     0.8168   2.588   0.1478
## 3 - 2 == 0   1.3173     0.7746   1.701   1.0000
## 4 - 2 == 0   2.2836     0.8259   2.765   0.0877 .
## 5 - 2 == 0   2.6256     0.8238   3.187   0.0225 *
## 4 - 3 == 0   0.9663     0.4977   1.942   0.7886
## 5 - 3 == 0   1.3083     0.4941   2.648   0.1243
## 5 - 4 == 0   0.3420     0.5712   0.599   1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

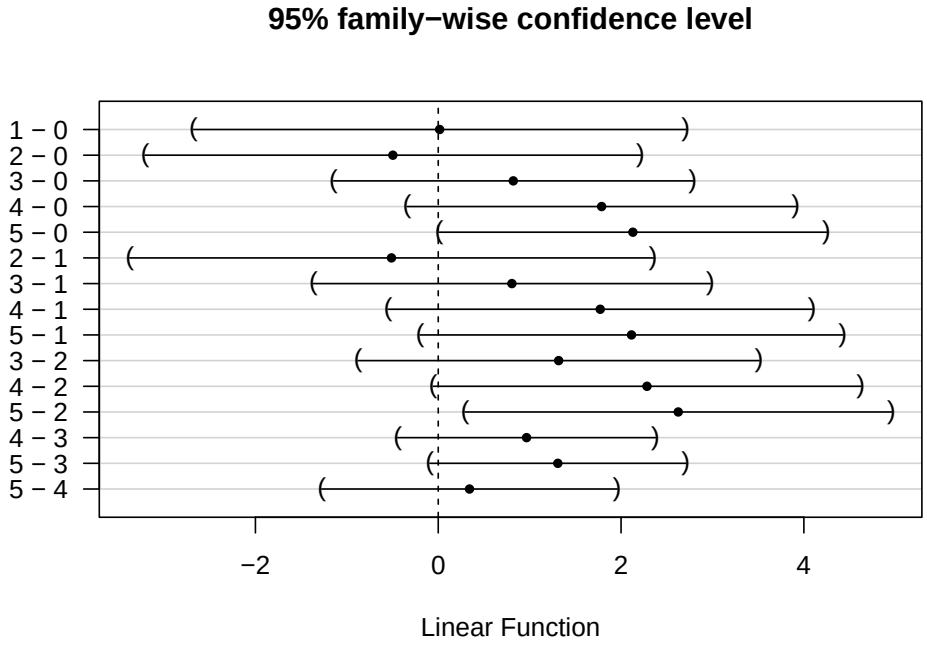
- The simultaneous confidence interval can be computed by the `confint` function:

```
# Confidence interval
confint(pairwise)
```

```
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = ACT ~ education, data = dat)
##
## Quantile = 2.8293
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##            Estimate lwr      upr
## 1 - 0 == 0  0.01520 -2.67634  2.70675
## 2 - 0 == 0 -0.49641 -3.20500  2.21218
## 3 - 0 == 0  0.82086 -1.14345  2.78518
## 4 - 0 == 0  1.78719 -0.33795  3.91232
```

```
## 5 - 0 == 0  2.12915  0.01064  4.24767
## 2 - 1 == 0 -0.51162 -3.37321  2.34998
## 3 - 1 == 0  0.80566 -1.36479  2.97610
## 4 - 1 == 0  1.77198 -0.54502  4.08898
## 5 - 1 == 0  2.11395 -0.19698  4.42488
## 3 - 2 == 0  1.31727 -0.87426  3.50881
## 4 - 2 == 0  2.28360 -0.05317  4.62036
## 5 - 2 == 0  2.62556  0.29481  4.95631
## 4 - 3 == 0  0.96632 -0.44172  2.37436
## 5 - 3 == 0  1.30829 -0.08974  2.70632
## 5 - 4 == 0  0.34197 -1.27425  1.95818
```

```
plot(confint(pairwise))
```



- Instead of post hoc comparison, researchers may have a priori contrasts from their research hypotheses. For example, researchers expect a linear trend in the impact of education on ACT score. The contrast coefficient of the linear trend for six groups can be created:

```
# Contrast
ctr <- matrix(c(-2.5, -1.5, -0.5, 0.5, 1.5, 2.5), 1, 6)
```

- The polynomial contrast coefficient is based on Table A10 in Maxwell and Delaney (2004). The contrast is specified in a matrix where rows represent different contrasts and columns represent the coefficient of each group. The contrast can be evaluated using the glht function from the `multcomp` package:

```r
# Simultanoues tests for general linear hypotheses
linear <- glht(model_oaov, linfct = mcp(education = ctr))
summary(linear)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = ACT ~ education, data = dat)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0    8.639      2.272   3.802 0.000156 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

- The contrast is provided in the `linfct` argument. Because only one contrast is tested, the familywise error rate correction is not needed.
- Next, all polynomial contrasts up to the fifth order are investigated for the difference in ACT between education levels:

```r
# Different contrasts as examples
linear <- c(-5, -3, -1, 1, 3, 5)
quadratic <- c(5, -1, -4, -4, -1, 5)
cubic <- c(-5, 7, 4, -4, -7, 5)
quartic <- c(1, -3, 2, 2, -3, 1)
quintic <- c(-1, 5, -10, 10, -5, 1)
mctr <- rbind(linear, quadratic, cubic, quartic, quintic)
```

- As mentioned above, the row of the contrast matrix represents different contrasts. The matrix of multiple contrasts can be used in the glht function:

```r
# Multiple testing
polynomial <- glht(model_oaov, linfct = mcp(education = mctr))
summary(polynomial, test=adjusted(type = "bonferroni"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = ACT ~ education, data = dat)
##
```

```
## Linear Hypotheses:
##                Estimate Std. Error t value Pr(>|t|)
## linear == 0      17.279      4.544   3.802  0.00078 ***
## quadratic == 0    7.546      4.928   1.531  0.63099
## cubic == 0       -7.027      7.515  -0.935  1.00000
## quartic == 0     -2.629      2.999  -0.877  1.00000
## quintic == 0      6.442      8.793   0.733  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

5.4 Factorial ANOVA

- Shall we use a different dataset this time?

- Here, assume we are interested in testing whether there are significant mean differences in body mass across the different species and sex levels.
- There are three different types for the species variable and two levels for the sex variable.

```
# Use penguins data
dat <- penguins

# Quick look at the data
str(dat)
```

```
## tibble [344 x 8] (S3: tbl_df/tbl/data.frame)
##  $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
##  $ bill_depth_mm    : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
##  $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
##  $ body_mass_g      : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
##  $ sex              : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...
```

```
##  $ year             : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007
```

```
describe(dat)
```

```
##                    vars   n    mean     sd  median trimmed    mad    min    max
## species*              1 344    1.92   0.89    2.00    1.90   1.48    1.0    3.0
## island*               2 344    1.66   0.73    2.00    1.58   1.48    1.0    3.0
## bill_length_mm        3 342   43.92   5.46   44.45   43.91   7.04   32.1   59.6
## bill_depth_mm         4 342   17.15   1.97   17.30   17.17   2.22   13.1   21.5
## flipper_length_mm     5 342  200.92  14.06  197.00  200.34  16.31  172.0  231.0
## body_mass_g           6 342 4201.75 801.95 4050.00 4154.01 889.56 2700.0 6300.0
## sex*                  7 333    1.50   0.50    2.00    1.51   0.00    1.0    2.0
## year                  8 344 2008.03   0.82 2008.00 2008.04   1.48 2007.0 2009.0
##                     range  skew kurtosis    se
## species*              2.0  0.16    -1.73  0.05
## island*               2.0  0.61    -0.91  0.04
## bill_length_mm       27.5  0.05    -0.89  0.30
## bill_depth_mm         8.4 -0.14    -0.92  0.11
## flipper_length_mm    59.0  0.34    -1.00  0.76
## body_mass_g        3600.0  0.47    -0.74 43.36
## sex*                  1.0 -0.02    -2.01  0.03
## year                  2.0 -0.05    -1.51  0.04
```

```
summary(dat)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

- For the ease of interpretation and implementation, I excluded penguins where the sex variable is missing:

```
# Data engineering for exclusing cases where sex variable is missing
dat <- dat %>% filter(!is.na(sex))
```

```
# Quick look at the modified data
str(dat)
```

```
## tibble [333 x 8] (S3: tbl_df/tbl/data.frame)
##  $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ bill_length_mm   : num [1:333] 39.1 39.5 40.3 36.7 39.3 38.9 39.2 41.1 38.6 34.6 ...
##  $ bill_depth_mm    : num [1:333] 18.7 17.4 18 19.3 20.6 17.8 19.6 17.6 21.2 21.1 ...
##  $ flipper_length_mm: int [1:333] 181 186 195 193 190 181 195 182 191 198 ...
##  $ body_mass_g      : int [1:333] 3750 3800 3250 3450 3650 3625 4675 3200 3800 4400 ...
##  $ sex              : Factor w/ 2 levels "female","male": 2 1 1 1 2 1 2 1 2 2 ...
##  $ year             : int [1:333] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

```
describe(dat)
```

```
##                    vars   n    mean     sd median trimmed    mad    min    max
## species*              1 333    1.92   0.89    2.0    1.90   1.48    1.0    3.0
## island*               2 333    1.65   0.71    2.0    1.57   1.48    1.0    3.0
## bill_length_mm        3 333   43.99   5.47   44.5   43.98   6.97   32.1   59.6
## bill_depth_mm         4 333   17.16   1.97   17.3   17.19   2.22   13.1   21.5
## flipper_length_mm     5 333  200.97  14.02  197.0  200.36  16.31  172.0  231.0
## body_mass_g           6 333 4207.06 805.22 4050.0 4159.46 889.56 2700.0 6300.0
## sex*                  7 333    1.50   0.50    2.0    1.51   0.00    1.0    2.0
## year                  8 333 2008.04   0.81 2008.0 2008.05   1.48 2007.0 2009.0
##                     range  skew kurtosis    se
## species*              2.0  0.16    -1.72  0.05
## island*               2.0  0.62    -0.85  0.04
## bill_length_mm       27.5  0.04    -0.90  0.30
## bill_depth_mm         8.4 -0.15    -0.91  0.11
## flipper_length_mm    59.0  0.36    -0.98  0.77
## body_mass_g        3600.0  0.47    -0.75 44.13
## sex*                  1.0 -0.02    -2.01  0.03
## year                  2.0 -0.08    -1.49  0.04
```

```
summary(dat)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :146   Biscoe   :163   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :123   1st Qu.:39.50   1st Qu.:15.60
##  Gentoo   :119   Torgersen: 47   Median :44.50   Median :17.30
##                                  Mean   :43.99   Mean   :17.16
##                                  3rd Qu.:48.60   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172       Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190       1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197       Median :4050                Median :2008
```

```
##  Mean   :201       Mean   :4207                Mean   :2008
##  3rd Qu.:213       3rd Qu.:4775                3rd Qu.:2009
##  Max.   :231       Max.   :6300                Max.   :2009
```

5.4.1 How can we characterize this design?

- There are two factors in this study. The first factor is species, and the second factor is sex. Species has three levels (Adelie, Chinstrap, and Gentoo), and sex has two levels (female and male). In total, there are six combinations formed by the three levels of species and the two levels of sex, with penguins assigned to all six combinations. This indicates that the current study employs a 3×2 factorial design. For this design, a two-way ANOVA should be conducted. The three null hypotheses being tested are as follows:
- H_0 (regarding species): Species does not have a significant main effect on body mass (group means on body mass across the three levels of species are not different).
- H_0 (regarding sex): Sex does not have a significant main effect on body mass (group means on body mass across the two levels of sex are not different).
- H_0 (regarding interaction): There is no significant interaction between species and sex on body mass.

5.4.2 Cells means for the data

```
# Mean by groups in a wide format
dat.tab <- dat %>% group_by(species, sex) %>%
  summarise(mean_bodymass = mean(body_mass_g)) %>%
  spread(sex, mean_bodymass)
```

```
## `summarise()` has grouped output by 'species'. You can override using the
## `.groups` argument.
```

```
# Add the marginal means for species
dat.tab$Means <- rowMeans(dat.tab[2:3])

# Add the marginal means for sex
col_means <- c(colMeans(dat.tab[2:4]))
dat.tab <- rbind(dat.tab, col_means)
dat.tab$species <- c("Adelie", "Chinstrap", "Gentoo", "Means")

# Print the table
dat.tab
```

```
## # A tibble: 4 x 4
```

```
## # Groups:   species [4]
##   species   female  male Means
##   <chr>      <dbl> <dbl> <dbl>
## 1 Adelie     3369. 4043. 3706.
## 2 Chinstrap  3527. 3939. 3733.
## 3 Gentoo     4680. 5485. 5082.
## 4 Means      3859. 4489. 4174.
```

- Can you understand the means by groups? In particular:
  - Can you find any simple effect? (i.e., the effect of one IV at a specific level of the other IV)
  - Can you find any main effect? (i.e., the average effect of the IV across the levels of all other IVs)
  - Can you find any interaction effect? (i.e., the simple effects of one IV are not constant across all levels of the other IV; there is a differential effect)

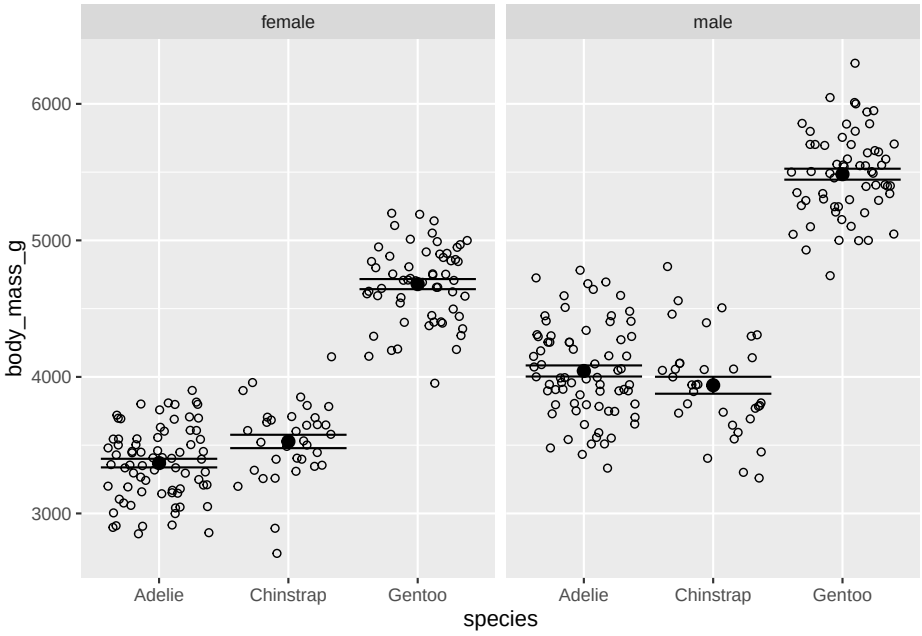
5.4.3 Visualizing cell means

```
# Jittered scatterplot with mean and SE
dat %>%
  ggplot(aes(x = species, y = body_mass_g)) +
  facet_wrap(~sex) +
  stat_summary(fun.dat = "mean_se", geom = "errorbar", wide = 0.5) +
  stat_summary(fun.dat = "mean_se", geom = "pointrange") +
  geom_jitter(cex = 1.5, pch = 1.0)
```

```
## Warning in stat_summary(fun.dat = "mean_se", geom = "errorbar", wide = 0.5):
## Ignoring unknown parameters: `fun.dat` and `wide`
```

```
## Warning in stat_summary(fun.dat = "mean_se", geom = "pointrange"): Ignoring
## unknown parameters: `fun.dat`
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



- What can you observe from the plot?

5.4.4 Analysis

Let's conduct a two-way ANOVA to test whether there are mean differences in body mass across the different species and sex levels.

```
# Perform a two-way ANOVA
# Both of the code below do the same job
model_taov <- aov(body_mass_g ~ species + sex + species:sex, data = dat)
## The colon represents the interaction effect only

model_taov <- aov(body_mass_g ~ species*sex, data = dat)
## The asterisk represents the interaction effect and the main effect of each variable
## (and all lower-order interactions)

# Results
summary(model_taov)
```

```
##              Df    Sum Sq  Mean Sq F value   Pr(>F)
## species       2 145190219 72595110 758.358  < 2e-16 ***
## sex           1  37090262 37090262 387.460  < 2e-16 ***
## species:sex   2   1676557   838278   8.757 0.000197 ***
## Residuals   327  31302628    95727
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Make the APA-style table
apa.aov.table(model_taov)
```

```
##
##
## ANOVA results using body_mass_g as the dependent variable
##
##
##      Predictor           SS  df           MS       F    p partial_eta2
##    (Intercept) 828480898.97   1 828480898.97 8654.65 .000
##        species  60350016.02   2  30175008.01  315.22 .000          .66
##            sex  16613441.78   1  16613441.78  173.55 .000          .35
##  species x sex   1676556.74   2    838278.37    8.76 .000          .05
##          Error  31302628.28 327     95726.69
##  CI_90_partial_eta2
##
##          [.61, .69]
##          [.28, .41]
##          [.02, .09]
##
##
## Note: Values in square brackets indicate the bounds of the 90% confidence interval for partial
```

```
# Describe the result following the APA-style; but for reference ONLY!
report(model_taov)
```

```
## The ANOVA (formula: body_mass_g ~ species * sex) suggests that:
##
##   - The main effect of species is statistically significant and large (F(2, 327)
## = 758.36, p < .001; Eta2 (partial) = 0.82, 95% CI [0.80, 1.00])
##   - The main effect of sex is statistically significant and large (F(1, 327) =
## 387.46, p < .001; Eta2 (partial) = 0.54, 95% CI [0.49, 1.00])
##   - The interaction between species and sex is statistically significant and
## small (F(2, 327) = 8.76, p < .001; Eta2 (partial) = 0.05, 95% CI [0.02, 1.00])
##
## Effect sizes were labelled following Field's (2013) recommendations.
```

5.4.5 Can you explain to me how those results have been obtained?

- I'll leave this as food for thought for you!

5.4.6 How can we interpret the results?

- Let's use an alpha level of 0.05 as the significance threshold. According to the results, the p-value for species is lower than the significance level,

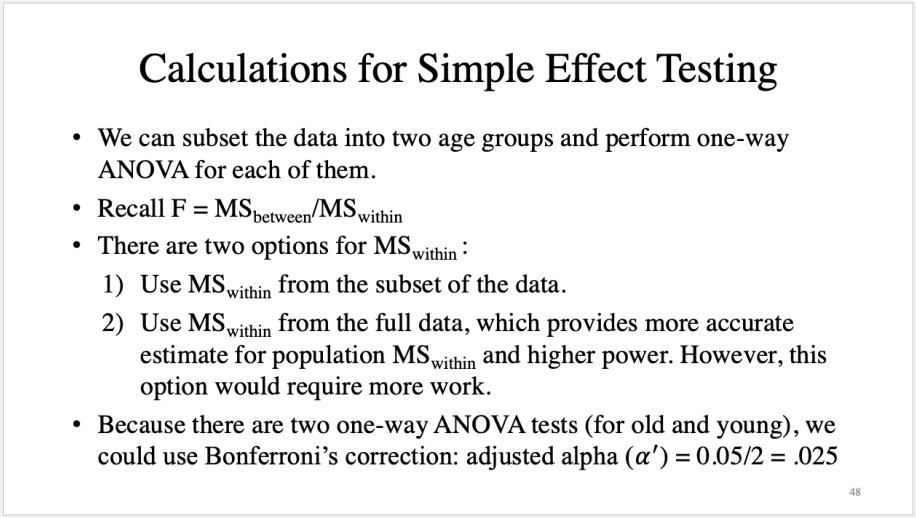
indicating that species has a significant effect on body mass. In other words, the group means for body mass differ significantly across the three species.

- The p-value for sex is also below the alpha level of 0.05, suggesting that sex has a significant effect on body mass. In other words, there are differences in body mass between female and male penguins.
- Lastly, the p-value for the interaction between species and sex (denoted as species:sex in the output) is below the significance threshold. This indicates a significant interaction between species and sex. That is, the effect of species on body mass is not consistent across the levels of sex, as observed in the plot above.

5.4.7 What to do with the interaction effect?

- The interaction in the analysis suggests that it would be beneficial to examine the simple effects. The next step is to analyze the differences in body mass due to species within each level of sex.
  - To do this, we will split the dataset and conduct a separate analysis for each level of sex.
  - Importantly, to do this, we use the mean-squared-within (i.e., MS_{within}) from the full data to compute the F value, and then use the `pf` function to find the p-value.
  - To control the overall Type I error rate, we will use an alpha level of 0.025 for each test.

5.4.7.1 What are you talking about, Ihnwhi...?



- Let's proceed with the analyses!

```
# Split data
## female
dat_f <- filter(dat, sex == "female")
model_taov_f <- aov(body_mass_g ~ species, data = dat_f)
summary(model_taov_f)
```

```
##              Df   Sum Sq  Mean Sq F value Pr(>F)
## species       2 60350016 30175008   393.2 <2e-16 ***
## Residuals   162 12430757    76733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# MS_between from the subset of data
MS_b_f <- summary(model_taov_f)[[1]][1,3]

# df_between frmo the subset of data
df_b_f <- summary(model_taov_f)[[1]][1,1]

# MS_within from the full data
MS_w <- summary(model_taov)[[1]][4,3]

# df_within from the full data
df_w <- summary(model_taov)[[1]][4,1]

# Calculate the F-value
F_f <- MS_b_f/MS_w

# Obtain the p-value
p_f <- pf(F_f, df1 = df_b_f, df2 = df_w, lower.tail = FALSE)

# Make a statistical decision - Is p-value lower than the adjusted alpha?
p_f < 0.025
```

```
## [1] TRUE
```

- The results indicate that the p-value is lower than the adjusted alpha level of 0.025 based on Bonferroni's correction. This suggests that there are significant mean differences across the three levels of species when the level of sex is female.
- We do the same for the subset of the data where the level of sex is male.

```
# Split data
## male
dat_m <- filter(dat, sex == "male")
model_taov_m <- aov(body_mass_g ~ species, data = dat_m)
summary(model_taov_m)
```

```
##              Df   Sum Sq  Mean Sq F value Pr(>F)
## species       2 84728125 42364062   370.4 <2e-16 ***
## Residuals   165 18871872   114375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# MS_between from the subset of data
MS_b_m <- summary(model_taov_m)[[1]][1,3]

# df_between frmo the subset of data
df_b_m <- summary(model_taov_m)[[1]][1,1]

# MS_within from the full data
MS_w <- summary(model_taov)[[1]][4,3]

# df_within from the full data
df_w <- summary(model_taov)[[1]][4,1]

# Calculate the F-value
F_m <- MS_b_m/MS_w

# Obtain the p-value
p_m <- pf(F_m, df1 = df_b_m, df2 = df_w, lower.tail = FALSE)

# Make a statistical decision - Is p-value lower than the adjusted alpha?
p_m < 0.025
```

```
## [1] TRUE
```

- Similarly, the p-value is lower than the adjusted alpha level of 0.025 based on Bonferroni's correction. This indicates that there are significant mean differences across the three levels of species when the level of sex is male.

5.4.8 This time, let's use pairwise comparison with Tukey's HSD to elaborate on the results of F.

- We will use the `lsmeans()` function to perform pairwise comparisons based on Tukey's HSD.

```
# Pairwise comparison
lsmeans(model_taov, pairwise ~ species, by = "sex", adjust = "tukey")
```

```
## $lsmeans
## sex = female:
##  species   lsmean   SE  df lower.CL upper.CL
##  Adelie      3369 36.2 327     3298     3440
```

```
##  Chinstrap   3527 53.1 327     3423     3632
##  Gentoo      4680 40.6 327     4600     4760
##
## sex = male:
##  species   lsmean   SE  df lower.CL upper.CL
##  Adelie      4043 36.2 327     3972     4115
##  Chinstrap   3939 53.1 327     3835     4043
##  Gentoo      5485 39.6 327     5407     5563
##
## Confidence level used: 0.95
##
## $contrasts
## sex = female:
##  contrast           estimate   SE  df t.ratio p.value
##  Adelie - Chinstrap     -158 64.2 327  -2.465  0.0377
##  Adelie - Gentoo       -1311 54.4 327 -24.088  <.0001
##  Chinstrap - Gentoo    -1153 66.8 327 -17.246  <.0001
##
## sex = male:
##  contrast           estimate   SE  df t.ratio p.value
##  Adelie - Chinstrap      105 64.2 327   1.627  0.2357
##  Adelie - Gentoo       -1441 53.7 327 -26.855  <.0001
##  Chinstrap - Gentoo    -1546 66.2 327 -23.345  <.0001
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

- Can you understand the results? Focus on the `$contrasts` section!
- Can you expand on your results? Feel free to refer to the lecture slides for a template to guide your write-up.

5.5 Regression analysis

- **Purpose of Regression Analysis**: Regression analysis is a statistical method used to examine the relationship between a dependent variable (outcome) and one or more independent variables (predictors). It helps quantify the strength and direction of these relationships.
- **Key Objective**: The primary goal is to model the relationship between variables, allowing researchers to explain variations in the dependent variable and make predictions about future or unobserved data points.
- **Types of Regression**: Regression analysis comes in various forms, such as simple regression (one predictor), multiple regression (multiple predictors), and specialized models (e.g., logistic regression for binary outcomes or polynomial regression for nonlinear relationships).

- **Assumptions and Applications**: Regression models rely on assumptions such as linearity, independence, homoscedasticity, and normality of residuals. It is widely applied in fields like psychology, economics, and engineering to analyze trends, test hypotheses, and predict outcomes. You will have a deeper understanding into the assumptions in PSY202B with Sarah!
- **Tips When Programming**: Be sure to differentiate what is regression on others!

5.5.1 Simple regression analysis

- A simple regression model is used to describe the relationship between two variables: an independent variable (IV) and a dependent variable (DV).
- The research question addressed by simple regression is whether the IV has an effect on the DV. If a significant effect is found, the model can be used to predict DV values for new IV values.
- Assuming a linear relationship between the two variables, the mathematical expression for a simple regression model is $y_i = b_0 + b_1 x_i + \varepsilon_i$, where b_0 and b_1 are the regression coefficients (intercept and slope, respectively), and ε_i represents the error term.
- The DV must be a continuous variable, while the IV can be either continuous or categorical.
  - For example, one-way ANOVA is a special case of simple regression, where the IV is categorical.

5.5.1.1 Simple regression analysis with a continuous predictor

- Say our goal is to predict the ACT score using age through a simple regression analysis.
- What would be the null hypothesis?

```
# Use the sat-act data again
dat_satact <- sat.act

# Simple regression analysis, with a continuous predictor
model_reg_s_cont <- lm(ACT ~ age, data = dat_satact)
summary(model_reg_s_cont)
```

```
##
## Call:
## lm(formula = ACT ~ age, data = dat_satact)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -25.3454  -3.1874   0.4301   3.6546   7.9915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.11035    0.52148  51.988  < 2e-16 ***
## age          0.05614    0.01910   2.939  0.00341 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.797 on 698 degrees of freedom
## Multiple R-squared:  0.01222,    Adjusted R-squared:  0.01081
## F-statistic: 8.635 on 1 and 698 DF,  p-value: 0.003406
```

- Can you understand the output and interpret the result?

5.5.1.2 Simple regression analysis with a categorical predictor

- This time, say we would like to predict the ACT score using education through a simple regression analysis.

```r
# Simple regression analysis, with a continuous predictor
model_reg_s_cate <- lm(ACT ~ as.factor(education), data = dat_satact)
summary(model_reg_s_cate)
```

```
##
## Call:
## lm(formula = ACT ~ as.factor(education), data = dat_satact)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.9773  -3.2945   0.5263   3.7055   9.0227
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            27.4737     0.6319  43.480  < 2e-16 ***
## as.factor(education)1   0.0152     0.9513   0.016  0.98725
## as.factor(education)2  -0.4964     0.9573  -0.519  0.60425
## as.factor(education)3   0.8209     0.6943   1.182  0.23748
## as.factor(education)4   1.7872     0.7511   2.379  0.01761 *
## as.factor(education)5   2.1292     0.7488   2.844  0.00459 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.771 on 694 degrees of freedom
## Multiple R-squared:  0.02887,    Adjusted R-squared:  0.02187
## F-statistic: 4.126 on 5 and 694 DF,  p-value: 0.001063
```

- Can you notice something from the above results compared to what we

did as below?

```
# Perform a one-way ANOVA
model_oaov <- aov(ACT ~ education, data = dat_satact)

# Print the output
summary(model_oaov)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## education     1    390   389.9   17.14 3.89e-05 ***
## Residuals   698  15874    22.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.2 Multiple regression analysis

- Multiple linear regression allows more than one IVs to predict the dependent variable (DV). To examine effects of multiple IVs on the DV, we can simply add IVs in the linear model, $y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_p x_{pi} + \varepsilon_i$.

5.5.2.1 Multiple regression analysis with two continuous variables

- Say our goal is to predict the body mass of penguins using flipper length and bill depth through a multiple regression analysis.
- What would be the null hypothesis?

```
# Load data again
dat_penguin <- penguins

# Multiple regression analysis
model_reg_m_1 <- lm(body_mass_g ~ flipper_length_mm + bill_depth_mm, data = dat_penguin
summary(model_reg_m_1)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + bill_depth_mm,
##     data = dat_penguin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1029.78  -271.45   -23.58   245.15  1275.97
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6541.907    540.751 -12.098   <2e-16 ***
## flipper_length_mm    51.541      1.865  27.635   <2e-16 ***
## bill_depth_mm        22.634     13.280   1.704   0.0892 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393.2 on 339 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.761,  Adjusted R-squared:  0.7596
## F-statistic: 539.8 on 2 and 339 DF,  p-value: < 2.2e-16
```

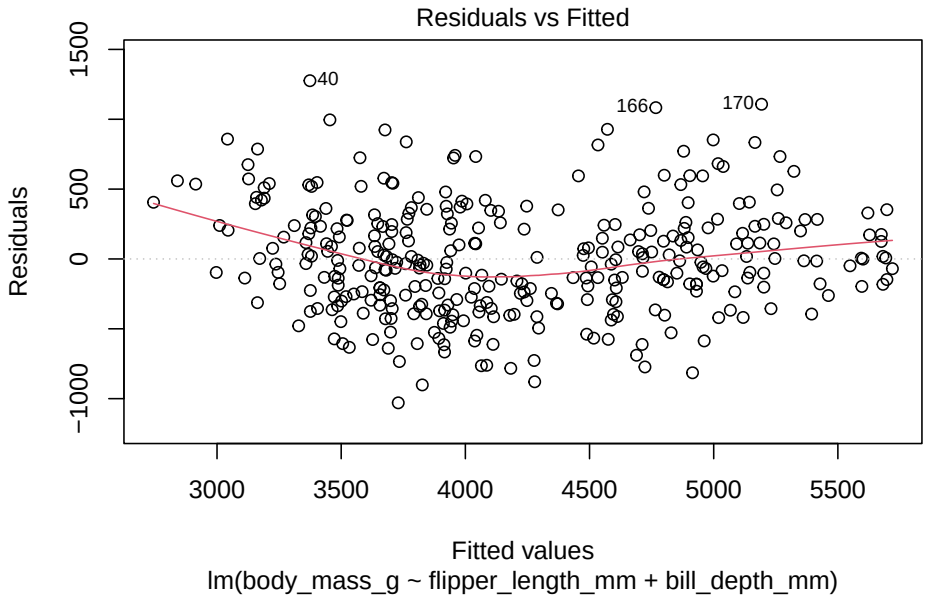
- Can you understand the output and interpret the result?

5.5.2.2 Model diagnostics for multiple regression analysis

Trailer: Again, much more in Sarah's PSY202B!

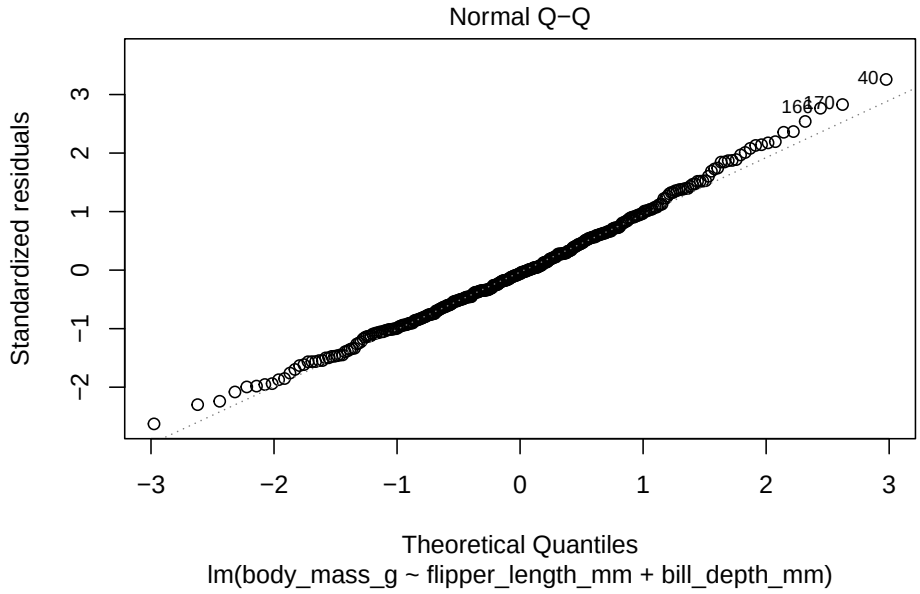
- For the purpose of diagnosing model assumptions, we can check four diagnostic plots: (1) residuals vs. fitted, (2) normal Q-Q, (3) scale-loation, (4) Cook's distance, and (5) residuals vs. leverage.

```
# Model diagnostics using visual tools
plot(model_reg_m_1, which = 1)
```



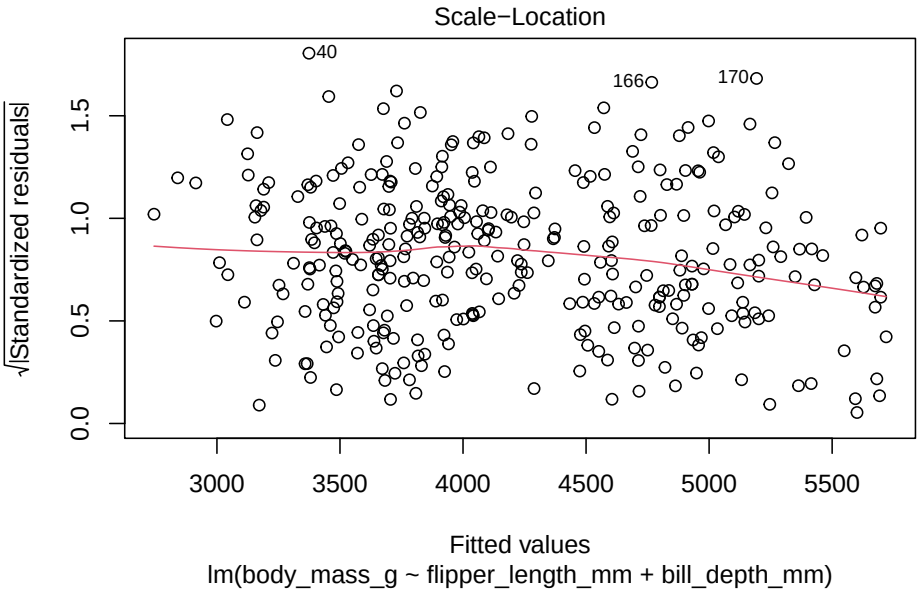
- A visual examination of the residuals vs. fitted plot indicates that there is equal variance along the line of best fit (i.e., regression line). Therefore, the assumption of homoscedasticity is met. In addition, the points are randomly scattered across the place, and there are no discernible nonlinear trends. This means that the assumption of linearity is also met.

```
# Model diagnostics using visual tools
plot(model_reg_m_1, which = 2)
```



- The Q-Q plot shows that there is a slight deviation in the lower-left corner, but this is not severe and looks okay. Also, most of the data points hover around the normal line. Thus, the normality assumption is met.

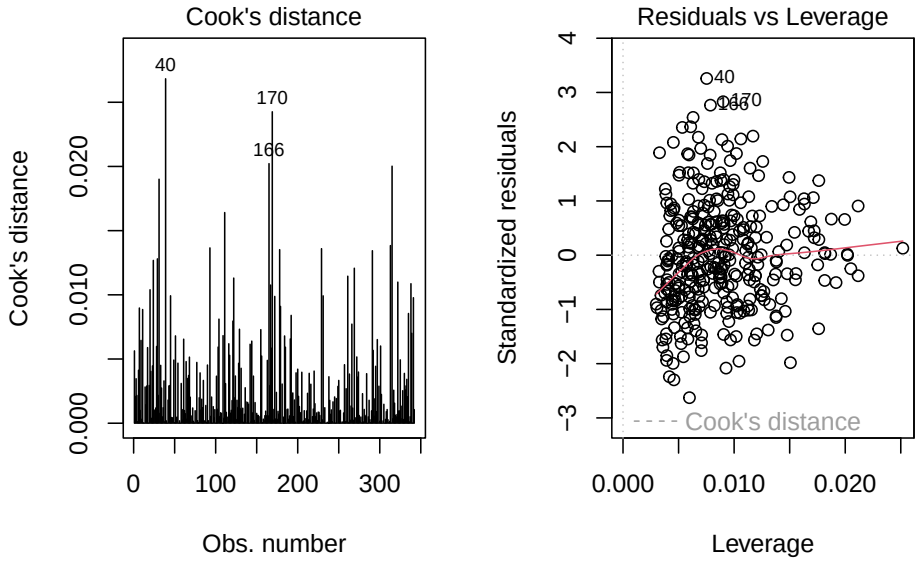
```
# Model diagnostics using visual tools
plot(model_reg_m_1, which = 3)
```



- The scale-location plot tells us whether the assumption of homoscedasticity

is satisfied or violated. If a model satisfies this assumption, the points should ideally be equally spread around the red horizontal line. According to the plot, the red horizontal line is fairly flat, which suggests that the assumption of homoscedasticity is met. While observations such as 40, 166, and 170 deviate slightly more, I do not believe this indicates a severe violation of the homoscedasticity assumption.

```
# Model diagnostics using visual tools
par(mfrow=c(1,2))
plot(model_reg_m_1, which = 4)
plot(model_reg_m_1, which = 5)
```




```
par(mfrow=c(1,1))
```

- To diagnose the existence of influential points that can alter the results of the regression analysis upon the inclusion or exclusion of a value, I checked the Cook's distance plot and the residuals vs. leverage plot. Cook's distance measures the influence of each data point on the fitted model. Larger values of Cook's distance, such as those greater than 0.5, indicate more influential points. According to the Cook's distance plot, values from observations 40, 170, 166 could be influential, so those values are worthy of closer scrutiny.
- I checked the residuals vs. leverage plot. According to this plot, observations with standardized residuals greater than 3 in absolute value are possible outliers. Indeed, there were three values whose absolute standardized residuals exceeded 3 (values from observations 40, 166, and 170), indicating the potential presence of outliers.
- In sum, the assumption of no influential points is likely violated, at least

for three specific observations. These values should be closely examined in the substantive research context.

5.5.2.3 Multiple regression analysis with one continuous variable and one categorical variable

- Say our goal is to predict the body mass of penguins using flipper length and sex through a multiple regression analysis.
- We use female as a reference category (i.e., coded as 0), where male is coded as 1.
- What would be the null hypothesis?

```
# Quick data engineering
dat_penguin <- dat_penguin %>%
  mutate(sex_dummy = if_else(sex == "male", 1, 0))

# Multiple regression analysis
model_reg_m_2 <- lm(body_mass_g ~ flipper_length_mm + sex_dummy, data = dat_penguin)
summary(model_reg_m_2)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + sex_dummy, data = dat_penguin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -910.28 -243.89   -2.94  238.85 1067.73
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5410.300    285.798 -18.931  < 2e-16 ***
## flipper_length_mm    46.982      1.441  32.598  < 2e-16 ***
## sex_dummy           347.850     40.342   8.623 2.78e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 355.9 on 330 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.8058, Adjusted R-squared:  0.8047
## F-statistic: 684.8 on 2 and 330 DF,  p-value: < 2.2e-16
```

- Can you understand the output and interpret the result?