BIOSTATS 540 – Introductorye Biostatistics
Fall 2022

Introduction to R
02 – R Essentials II

Welcome

In this second introduction to R essentials, get a little taste of using R with data! We begin with setting the working directory and a brief introduction to packages (enough so that you appreciate their virtue and know how to work them). Then we look at some data and produce some numerical and graphical summaries. Enjoy!!

#1. Highlights of Introduction to R 01 – R Essentials

We will be doing all our work in R Studio	R Studio is an application that sits "on top" of R. R is then "under the hood". R Studio provides a very friendly environment for doing lots of things: writing and executing code, managing files and directories, and working with packages.
`>` is the command prompt **`#` denotes a comment; R ignores the rest of the line**	
R is case sensitive	And is unforgiving!
Use the function `c( )` to create vectors of data and separate arguments by commas	**For example:** `v1 <- c(14,35,81,99)`
A "dataset" (analogous to excel spreadsheet or SAS dataset or Stata dataset) is called a `data frame` in R	
R can work with more than one dataset at a time.	
To identify a variable in a dataframe, R utilizes a two-part naming convention: **dataframename$variablename**	**For example:** `arthritis$Age`
Consider using the package `{swirl}` to learn R interactively	

#2. Introduction to the Arthritis Dataset

Citation:
Edward Gracely, "Arthritis Treatment Dataset", *TSHS Resources Portal (2020)*. Available: https://www.causeweb.org/tshs/arthritis-treatment/.

This was a study of treatment for rheumatoid arthritis (RA) in elderly patients. With the availability of effective anti-RA agents, disease activity measurements can inform the specification of treatment and are part of an approach known as "treat to target".

Research Question: Compared to younger RA patients, are older RA patients less likely to have their disease activity measured and less likely to receive aggressive treatment?

Study Design: Retrospective cohort study that compared two groups: Controls = patients age 40-70 versus Elderly = patients age 75+ years.

Sample Size: n = 530 (459 controls + 71 elderly)
Variables (partial listing): 14

arthritis.xlsx Data Dictionary (for 4 variables only)

Position	Variable	Variable Label	Units	Codes	Missing data
1	ID	Unique subject ID			None
2	Age	Age in years	Years	90 = 90+ for HIPAA purposes	None
3	AgeGp	Age Group		1 = 40 to 70 years ("control") 2 = 75 and older ("elderly")	None
4	Sex	Patient sex		0 = Female 1 = Male	None

#3. Set Your Working Directory
setwd( ) and getwd( )

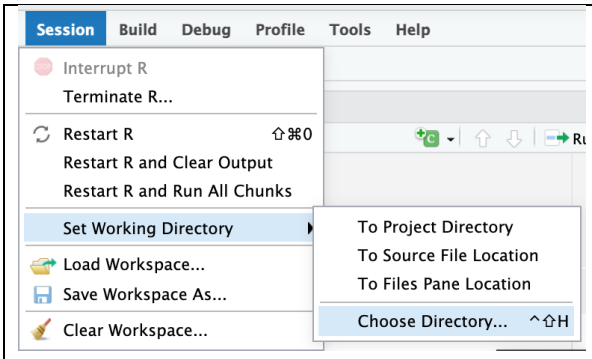
What is the working directory.

R needs to know where to find the files to **read from** and where to **write to.** This location is a directory with an associated path and is known as your **working directory**.

setwd( ) - Set your working directory

getwd( ) - Show current working directory

How to Set Your Working Directory using the R Studio Menus
From the top menu bar, click Session > Set Working Directory > Choose Directory
Browse to navigate to your desired folder. Click **CHOOSE.**



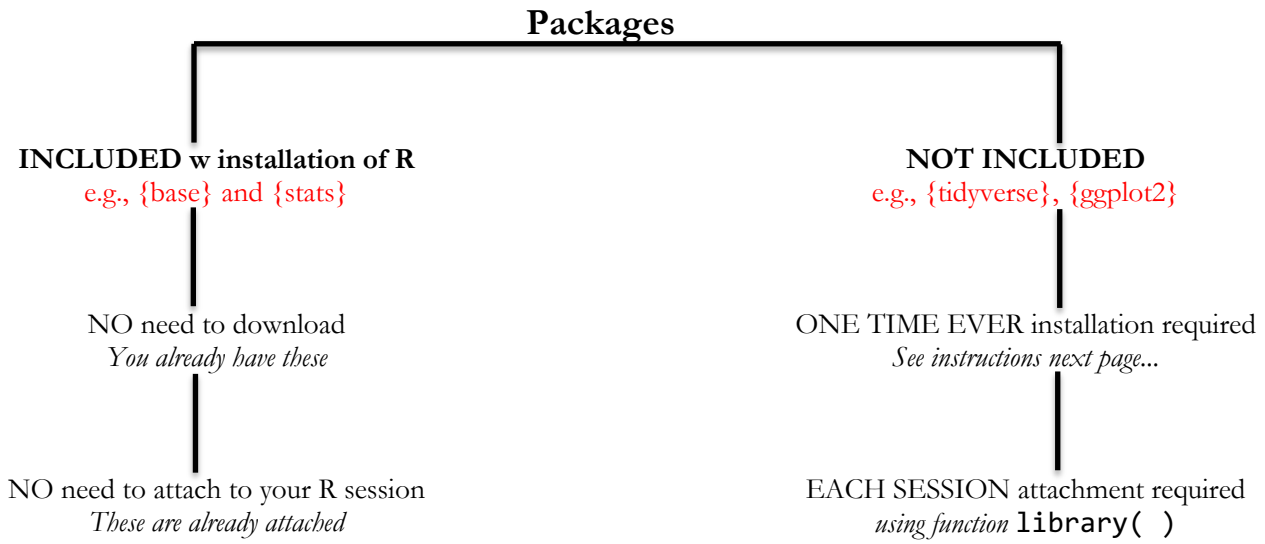
How to Set Your Working Directory using the `setwd( )` function in the Console
IMPORTANT. The path name must be enclosed in quotes.
Example (Windows): `setwd("My Documents/BIOSTATS 540/homeworks")`
Example (Mac): `setwd("~Desktop/BIOSTATS 540/homeworks")`

How to Show Your Current Working Directory using the `getwd( )` function in the Console
`getwd( )`
IMPORTANT. The path name must be enclosed in quotes.
Example (Windows): `setwd("My Documents/BIOSTATS 540/homeworks")`
Example (Mac): `setwd("~Desktop/BIOSTATS 540/homeworks")`

#4. Introduction to Packages

A **package** is a collection of functions and datasets

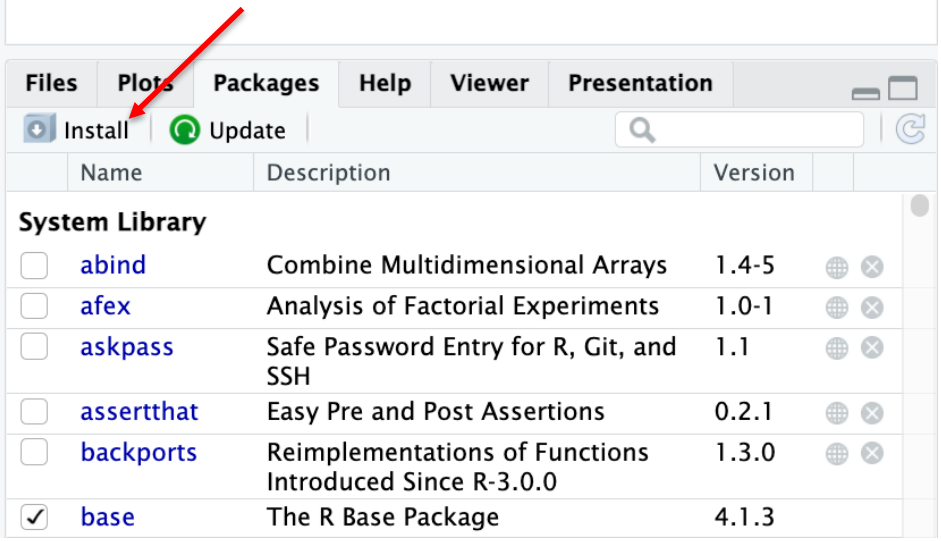
There are two types of packages.



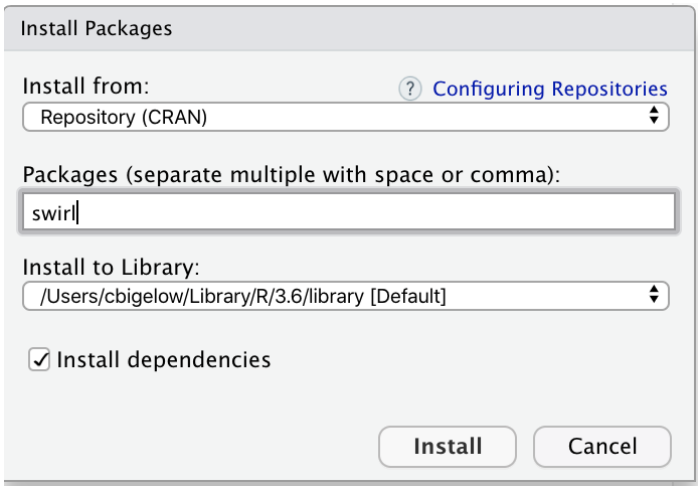
How to Install a Package Using R Studio Menus (recommended)

Example: `{ swirl }`

From the **FILES/PLOTS/PACKAGES** pane choose **PACKAGES** and click **Install**



Key:



In Install from: **default (Repository CRAN) is fine**

In Packages (separate multiple with space or comma:) **swirl**

In Install to Library: **leave as is**

Check box for "Install dependencies": **check**

At bottom, click **Install**

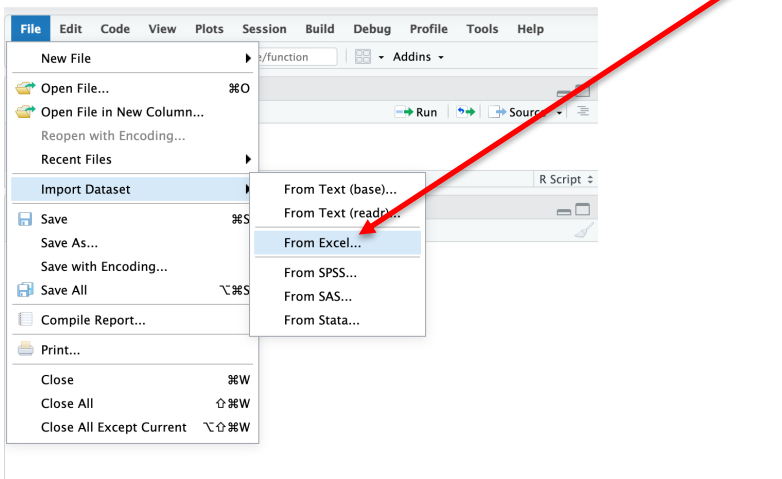
#5. Import Excel Data

Preliminaries (Important):

(1) Make sure that you have downloaded from the course website the dataset arthritis.xlsx.
(2) Strongly encouraged: *(Source: marinstats lectures)* Importing Excel Data into R (video, 8:12)

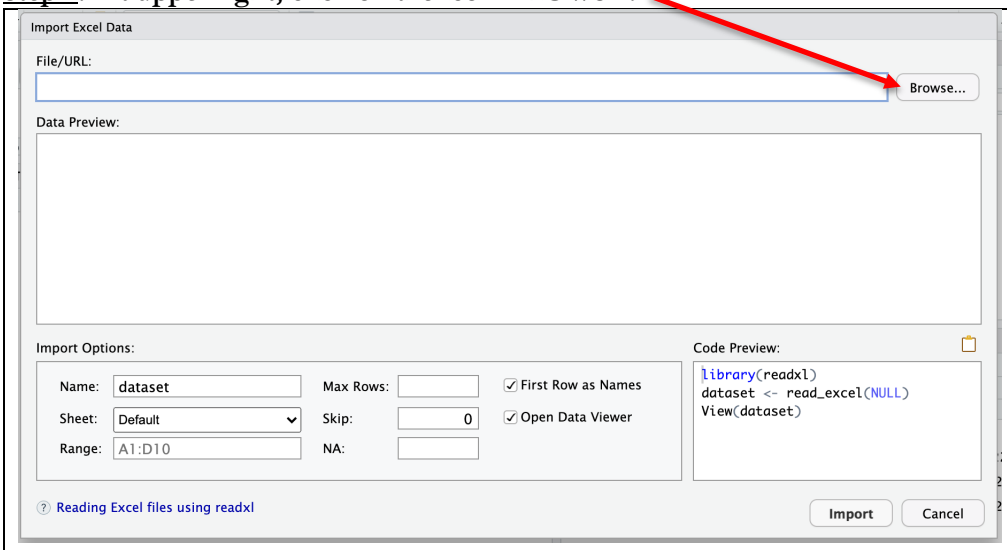
How to Import Excel Data Using R Studio Menus

Step 1: At upper left; FILE > IMPORT DATASET > FROM EXCEL

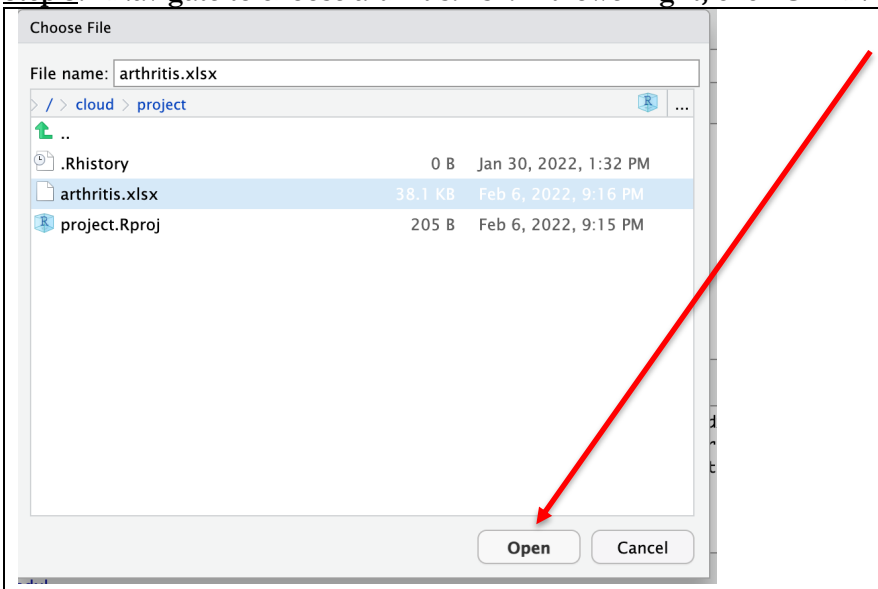


Note: R may return a message saying that you need to install readxl. Click YES. Then wait until you get a prompt.

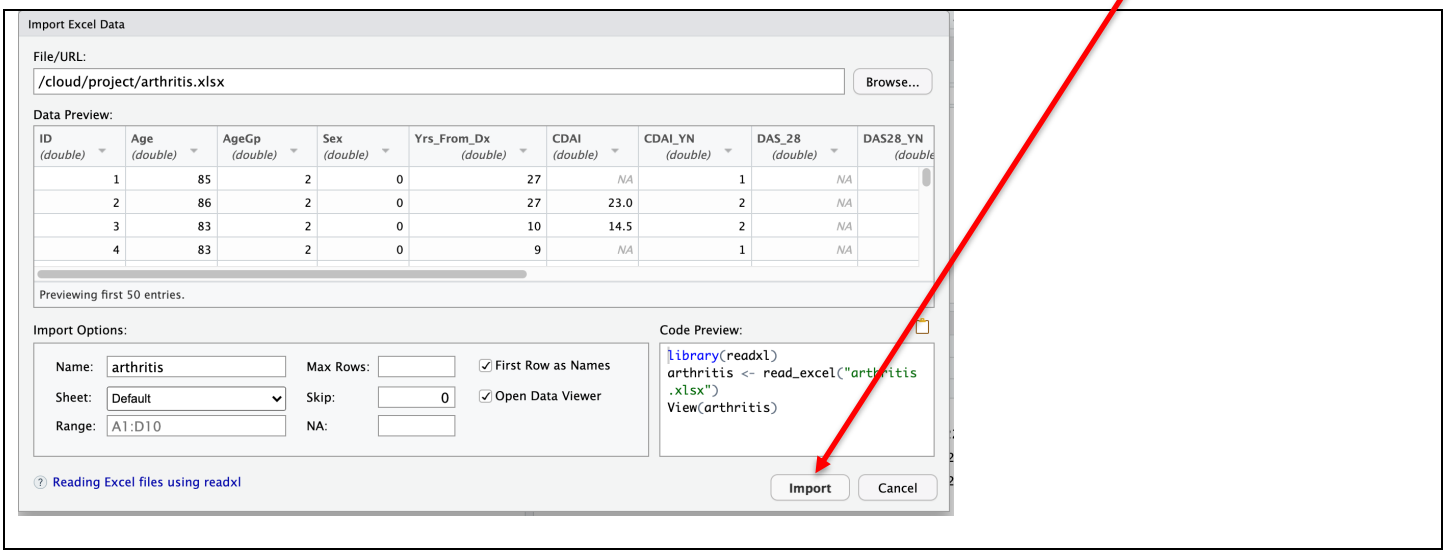
Step 2: At upper right, click on the icon BROWSE.



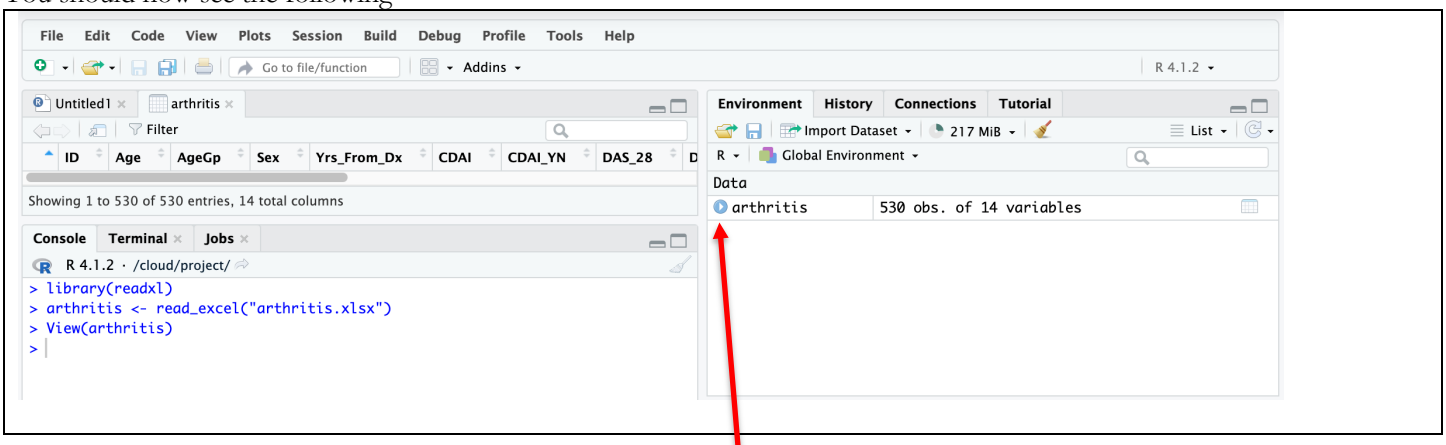
Step 3: Navigate to choose arthritis.xlsx. At lower right, click OPEN



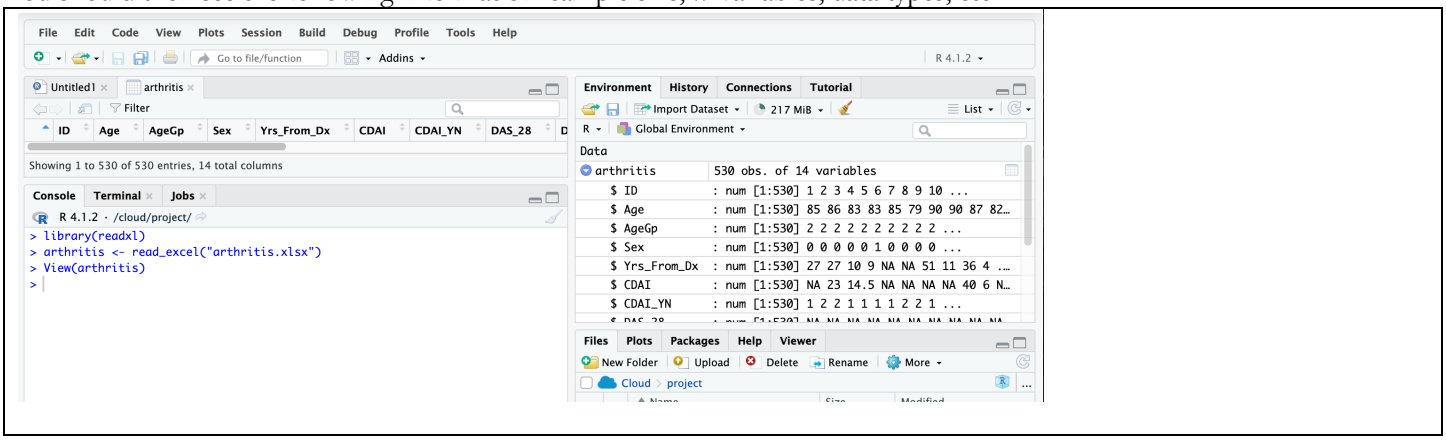
Step 4: Take your time here in making your selections. All set? At lower right, click IMPORT



You should now see the following



Step 5: At right, in the Environment tab, click on the down arrow next to the dataset name arthritis.
You should then see the following information: sample size, # variables, data types, etc.



#6. Numerical Descriptives Using the Package {summarytools}

Preliminaries (Required):

Step 1: Install (one time) the package **{summarytools}**

Step 2: Load/Attach (once each session) the package using the command **library(summarytools)**

```
# Inspect data using structure function str( )
str(arthritis)
```

```
tibble [530 × 14] (S3: tbl_df/tbl/data.frame)
 $ ID           : num [1:530] 1 2 3 4 5 6 7 8 9 10 ...
 $ Age          : num [1:530] 85 86 83 83 85 79 90 90 87 82 ...
 $ AgeGp        : num [1:530] 2 2 2 2 2 2 2 2 2 2 ...
 $ Sex          : num [1:530] 0 0 0 0 0 1 0 0 0 0 ...
 $ Yrs_From_Dx  : num [1:530] 27 27 10 9 NA NA 51 11 36 4 ...
 $ CDAI         : num [1:530] NA 23 14.5 NA NA NA NA 40 6 NA ...
 $ CDAI_YN      : num [1:530] 1 2 2 1 1 1 1 2 2 1 ...
 $ DAS_28       : num [1:530] NA NA NA NA NA NA NA NA NA NA ...
 $ DAS28_YN     : num [1:530] 1 1 1 1 1 1 1 1 1 1 ...
 $ Steroids_GT_5: num [1:530] 0 1 1 1 0 0 0 1 0 0 ...
 $ DMARDs       : num [1:530] 1 1 1 1 0 0 1 0 0 1 ...
 $ Biologics    : num [1:530] 0 0 1 0 0 0 1 0 1 0 ...
 $ sDMARDS      : num [1:530] 0 0 0 0 0 0 0 0 0 0 ...
 $ OsteopScreen : num [1:530] 0 1 1 1 0 0 0 1 1 1 ...
```

```
# descr( ) in package {summarytools} to produce descriptive statistics of ONE continuous variable
descr(arthritis$Age)
```

```
Descriptive Statistics
arthritis$Age
N: 530

                      Age
----------------- --------
             Mean    60.69
          Std.Dev    10.47
              Min    42.00
               Q1    54.00
           Median    59.00
               Q3    66.00
              Max    90.00
              MAD     8.90
              IQR    12.00
               CV     0.17
         Skewness     0.76
      SE.Skewness     0.11
         Kurtosis     0.27
          N.Valid   530.00
        Pct.Valid   100.00
```

```
# option stats=c( ) to choose statistics to report
# option transpose=TRUE to display output horizontally (instead of default vertical display)
descr(arthritis$Age,
                stats = c("n.valid","mean", "sd","min","q1", "med", "q3", "max"),
                transpose = TRUE)
```

```
Descriptive Statistics
arthritis$Age
N: 530

            N.Valid    Mean   Std.Dev     Min      Q1   Median      Q3     Max
--------- --------- ------- --------- ------- ------- -------- ------- -------
      Age    530.00   60.69     10.47   42.00   54.00    59.00   66.00   90.00
```

#7. Produce a Graph Using the Package {ggplot2}

Preliminaries (Required):

Step 1: Install (one time) the package **{ggplot2}**

Step 2: Load/Attach (once each session) the package using the command **library(ggplot2)**

Good to Know.
A graph produced using ggplot is built "layer by layer". Once you get the hang of it, it's kind of fun!

Step 1: Attach the package {ggplot2} using the command `library( )`
library(ggplot2)

Step 2 *(optional, recommended):* Create each plot in its own R Markdown chunk (stay tuned!)

Step 3 *(optional, recommended):* Code your graph line by line, layering as you go.

First line: # Execute and correct as needed
data = *dataframename*

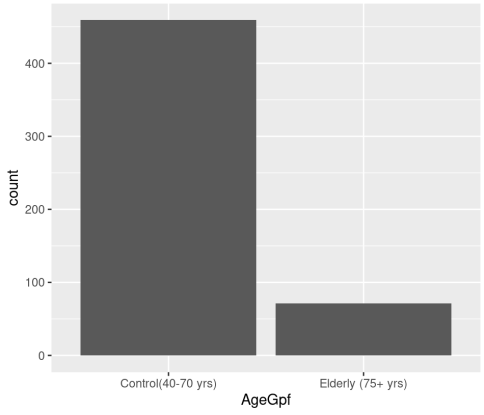
Second line is added # Execute and correct as needed
data = *dataframename* **+**
aes( )

Third line is added: # Execute and correct as needed
data = *dataframename* **+**
aes( ) +
geom_*xxx*( )

Tips

- The continuation **+** MUST BE at the end of the line (not at the start of the next line)
- As you add lines, execute all of the accumulating layers
- By building your graph layer by layer, it is easier to trouble shoot errors

```
# ggplot( ) and geom_bar( ) in package {ggplot2} to produce Bar Graph of ONE categorical variable
ggplot(data=arthritis) +
  aes(x=AgeGpf) +
  geom_bar(na.rm=T)
```




```
# ggplot( ) and geom_bar( ) in package {ggplot2} to produce Bar Graph of ONE categorical variable
# plus layer labs( ) to produce labels
ggplot(data=arthritis) +
  aes(x=AgeGpf) +
  geom_bar(na.rm=T) +
  labs(x="Age Group",
       y="Frequency",
       title="Bar Graph of Age Group")
```