Workshop #1:

Intro to R

Presented by: UF American Statistical Association

ASA
University of Florida

Table of contents

01

ASA

or “Stats club”

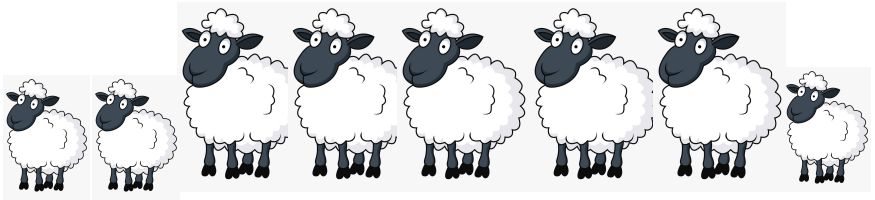
We are a club dedicated to statistics education/ networking

Mentorship program coming this week!

discord

link tree

Self proclaimed free thinkers when they read the “SCAN ME” qr code

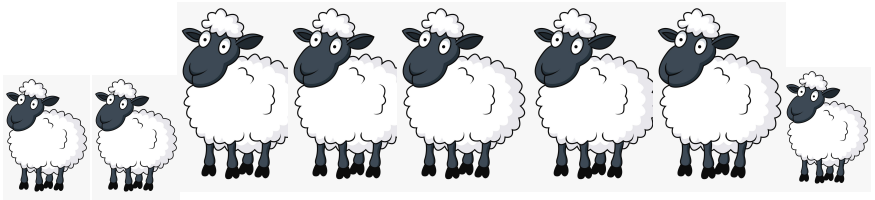


- 10/25 - DSI/ACM/More Halloween Painting Social
- 10/30 - Stats Halloween Social
- 11/6 - Workshop #2 (TBA)

discord

link tree

Self proclaimed free thinkers when they read the “SCAN ME” qr code



Getting started with R

02

Why R?

Why R?

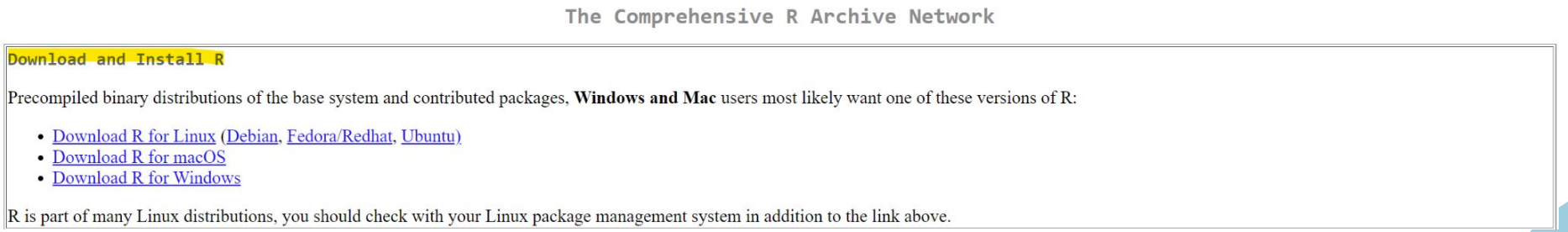
- It’s free and easy to use
- Best for statistical computing and graphics
- Visualize data for large and messy datasets
- Easy to clean data and find which specific datasets are useful
- Finally and most importantly, you have to use it to PASS YOUR STATISTICS CLASSES!!

03

System Setup

Install R

1. Go to CRAN (**https://cran.r-project.org/**) and select the download option that matches your system


The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

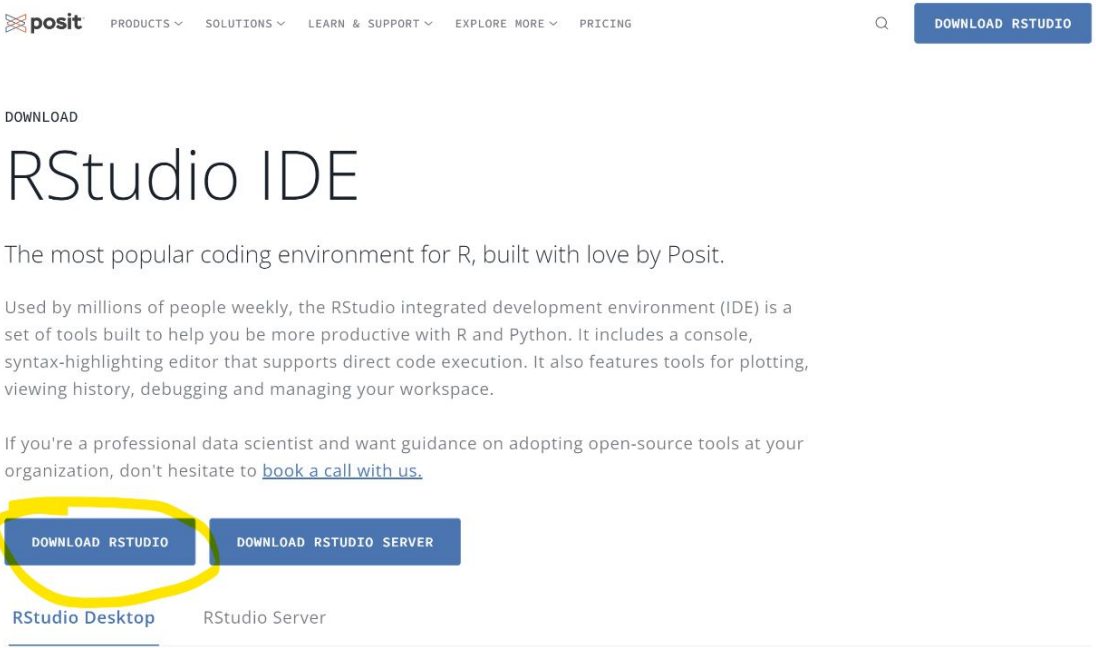
- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

We have now installed base R!!!
Now we need to install a environment to write our code

R studio

1. Go to the [RStudio download page](**https://posit.co/downloads/**) to install the free version of Rstudio for your computer.



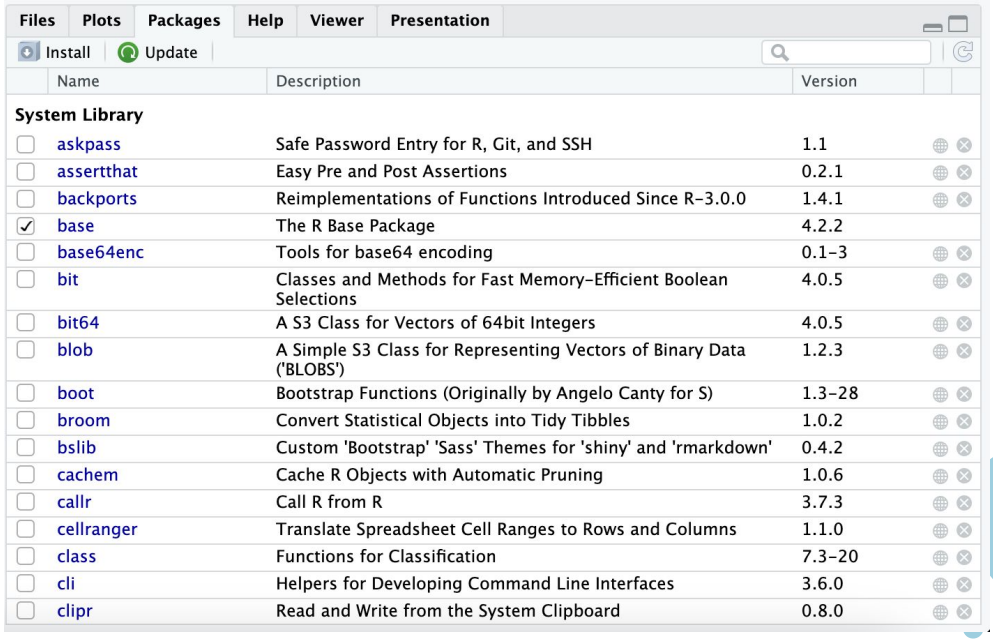
After successfully download R and R studio...

Open the Rstudio file we downloaded and follow the instructions to install.

When we can open R studio, we will now have the base R package.

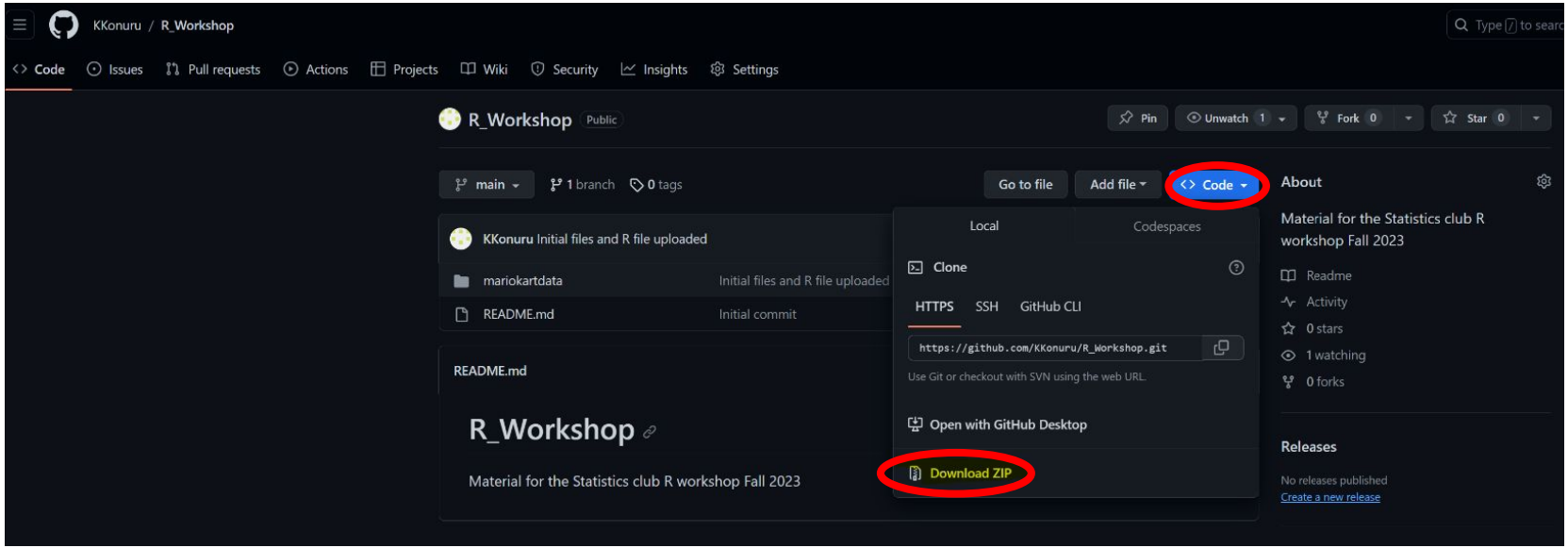
Select the other packages you want to use and also download them.

Now we are all set up for start using R studio!!!



Getting ready to code

1. First visit this github link: **https://github.com/KKonuru/R_Workshop/** to download the zip folder
2. Then open the R_workshop_student.rmd file in R studio



04

Hands On

Please switch to R Studio

R basics

- Before getting started with our interactive workshop lets go over some basic syntax
- Note: to execute these commands type these in the console section on the bottom tab of R studio
- We have arithmetic operators in R

```
> 3+2 #Addition
[1] 5
> 9/2 #Division
[1] 4.5
```

- We can create vectors that hold numbers by using the c() function

```
> x <- c(3,4,-5)
> x
[1]  3  4 -5
```

- Vectors can hold any of R's five atomic types
  - Logical (True or False)
  - Character
  - integer
  - double
  - complex
- Numerical vectors hold typically hold a type double

R basics (cont.)

- R has ways to generate sequences as well

```
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> seq(from=35,to=80,by=3)
 [1] 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77 80
```

- We can index elements by position using square brackets [] and a positive index to specify elements
- In R the first element is accessed using 1 as the index not 0

```
> x<-1:35
> x[2:5]
[1] 2 3 4 5
> x<-1:35
> x[2]
[1] 2
> x[2:5]
[1] 2 3 4 5
> x[-(1:33)]
[1] 34 35
```

- We can also use logical vectors to index in a list of 35 elements.

```
> x[c(TRUE,FALSE,FALSE,FALSE,FALSE)]
[1]  1  6 11 16 21 26 31
```

Lists

- A list in R can contain any combination of types

```
> cars <- list("Toyota","Honda","Mercedes","BMW",c("Japan","Japan","Germany","Germany"))
> cars
[[1]]
[1] "Toyota"

[[2]]
[1] "Honda"

[[3]]
[1] "Mercedes"

[[4]]
[1] "BMW"

[[5]]
[1] "Japan"   "Japan"   "Germany" "Germany"

> cars <- list("Toyota","Honda","Mercedes","BMW",countries=c("Japan","Japan","Germany","Germany"))
> cars
[[1]]
[1] "Toyota"

[[2]]
[1] "Honda"

[[3]]
[1] "Mercedes"

[[4]]
[1] "BMW"

$countries
[1] "Japan"   "Japan"   "Germany" "Germany"

> cars[2]
[[1]]
[1] "Honda"

> cars[[2]]
[1] "Honda"
> cars["countries"]
$countries
[1] "Japan"   "Japan"   "Germany" "Germany"

> cars$countries
[1] "Japan"   "Japan"   "Germany" "Germany"
```

- We can access a element using [] and it returns a list
- To get the actual value use [[]]
- If the value in the list has a name then we can use the $ operator

Dataframes

- This is what we will be working with during our workshop
- We create a dataframe after reading a file such as a csv
- A dataframe is a list of columns
    - Each item in this list is a column and we can access a column using the $ operator or [[]]
- Example code:

```
> data
       State Population Area_SquareMiles
1   New York   19530000            54555
2 California   39540000           163696
3      Texas   29150000           268596
4    Florida   21530000            65755
5   Illinois   12740000            57914
> data$State
[1] "New York"   "California" "Texas"      "Florida"    "Illinois"
> data[[3]]
[1]  54555 163696 268596  65755  57914
```

Creating our dataframe

- We have opened up a R markdown file where we can write code in code chunks like this

- Anything written outside the code chunk is text in our html file
- At the top right of each code chunk is a green run button. You must click that button for your code to execute in that code chunk
- We are using karts.csv which contains data on each of the karts in mario kart
- First we use read.csv("karts.csv")
  - This will load the data into a **data frame**
- Next we can use the function head(x) which prints out the first 6 rows in the data frame
  - We can also specify the number of rows by passing a second parameter of a integer ex. head(kart,10) prints 10 rows

Analyzing the karts dataset

- Question 1 part a asks to find how many karts are in the dataset
  - we can use the nrow(kart) to get the number of rows
- It is your turn to answer part b to get the number of columns
  - Hint: use the same command but replace row with col
- Now we want to access data in a column but we need the names
  - For part c we use the names(x) command which returns the names of the columns
- For part d, write the code to get the values stored in the Acceleration column
  - Hint: Remember how a dataframe is a list of columns
- Next for part e we will rename the column Body to Kart
  - We use names(kart) to get the list of names
  - names(kart)[1] will get the first index in names and we can assign it Kart
  - names(kart)[1] <- “Kart”
- Part f will arrange the table so that it goes from the greatest acceleration to the least
  - We use the function arrange(kart,desc(Acceleration) which returns a new dataframe
  - It arranges the rows in descending order for acceleration
  - We then use head(kart) to get the 6 rows with greatest acceleration

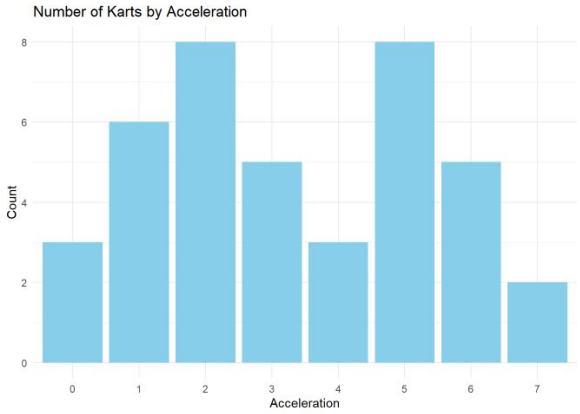
Analysis of Karts dataset (cont.)

- Now it is your turn to complete part g to get the value in the third row and fourth column
  - Hint: Use indexing for the dataset. You should get the result 3
- Part h will only print the rows with acceleration of 5 and only the columns weight and acceleration
  - kart[kart$Acceleration==5,c("Weight", "Acceleration")]
  - This command will get the rows of acceleration of 5 and the columns specified

Generating graphs

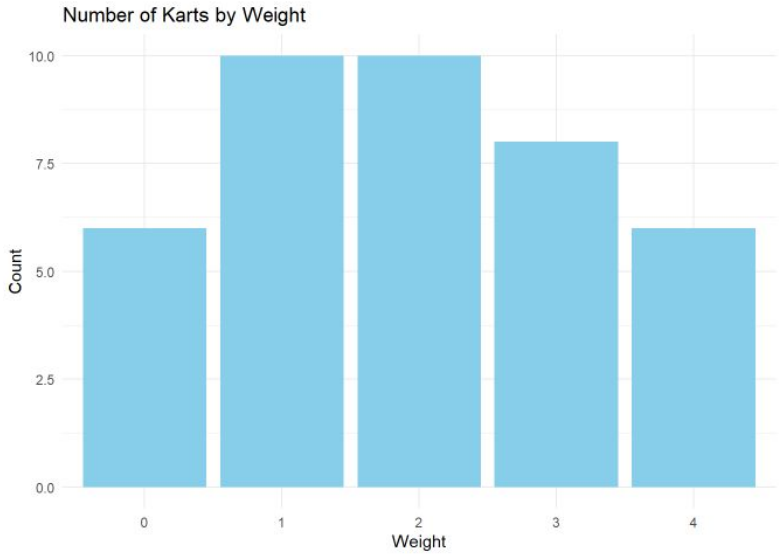
- We can use ggplot to generate bar graphs of the number of karts given
  - First we do acc_counts <- as.data.frame(table(kart$Acceleration)) where table() makes a frequency table of acceleration and we convert it to a dataframe
  - Next we do colnames(acc_counts) <- c("Acceleration", "Count") to assign the names for the columns to access it when making the bar graph
  - Next we use the ggplot function to get the result

```
acc_counts <- as.data.frame(table(kart$Acceleration))

colnames(acc_counts) <- c("Acceleration", "Count")

ggplot(acc_counts, aes(x = Acceleration, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Number of Karts by Acceleration",
       x = "Acceleration",
       y = "Count") +
  theme_minimal()
```



Generating graphs (cont.)

- Now it is your turn to complete part b and generate a similar bar graph of the number of karts for each weight
  - Try and see how you can use the code in part a to accomplish this task
- Your graph should look like this:



Thanks

Do you have any questions?
ufstatsclub@gmail.com
Link Tree