

# Workshop #1: Intro to R

Presented by: UF American Statistical Association



# Table of contents

01

## About ASA

How to get involved

02

## Why R?

.rmd vs .r, etc

03

## System Setup

Setting up your computer  
for the workshop!

04

## Hands On!

Coding in R with the  
RStudio software



01

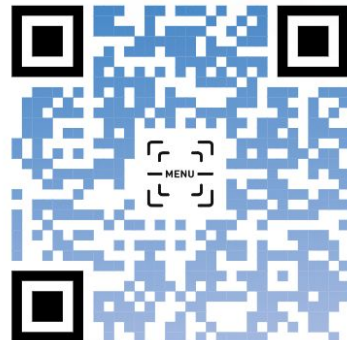
**ASA**

or “Stats club”



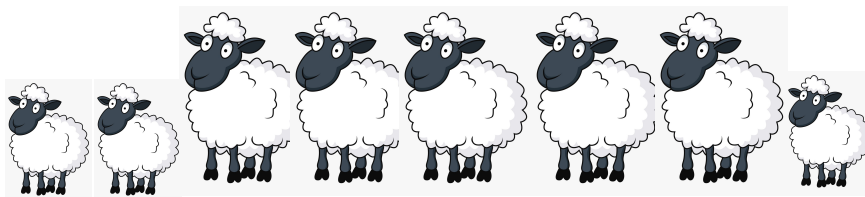
discord

We are a club dedicated to  
statistics education/ networking  
Mentorship program coming this  
week!



link tree

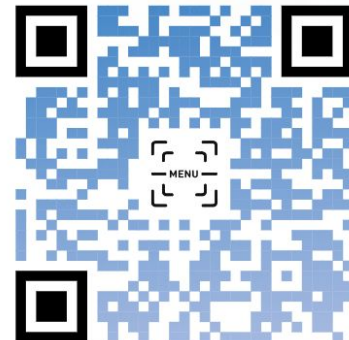
Self proclaimed free thinkers when they read  
the "SCAN ME" qr code





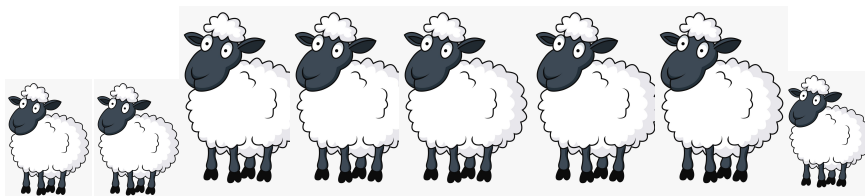
discord

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



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/) ( <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](#)

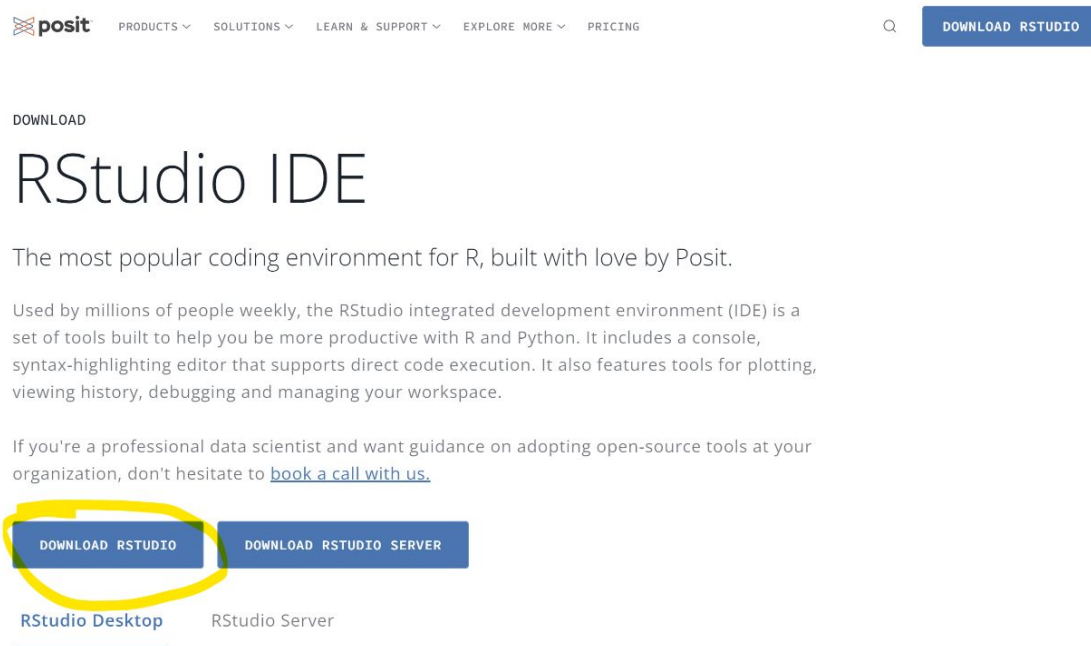
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

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



# R studio

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



The screenshot shows the RStudio download page. At the top, there is a navigation bar with the Posit logo and links for PRODUCTS, SOLUTIONS, LEARN & SUPPORT, EXPLORE MORE, and PRICING. A search icon and a 'DOWNLOAD RSTUDIO' button are also present. The main heading is 'RStudio IDE'. Below it, a paragraph describes RStudio as the most popular coding environment for R, built with love by Posit. Another paragraph mentions that RStudio is used by millions of people weekly and includes a console, syntax-highlighting editor, and tools for plotting, viewing history, debugging, and managing the workspace. A link to 'book a call with us' is provided for professional data scientists. At the bottom, there are two buttons: 'DOWNLOAD RSTUDIO' (highlighted with a yellow circle) and 'DOWNLOAD RSTUDIO SERVER'. Below these buttons, the text 'RStudio Desktop' and 'RStudio Server' are displayed, with 'RStudio Desktop' being the selected option.

posit PRODUCTS SOLUTIONS LEARN & SUPPORT EXPLORE MORE PRICING

Q DOWNLOAD RSTUDIO

DOWNLOAD

## RStudio IDE

The most popular coding environment for R, built with love by Posit.

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution. It also features tools for plotting, viewing history, debugging and managing your workspace.

If you're a professional data scientist and want guidance on adopting open-source tools at your organization, don't hesitate to [book a call with us](#).

DOWNLOAD RSTUDIO DOWNLOAD RSTUDIO SERVER

RStudio Desktop RStudio Server

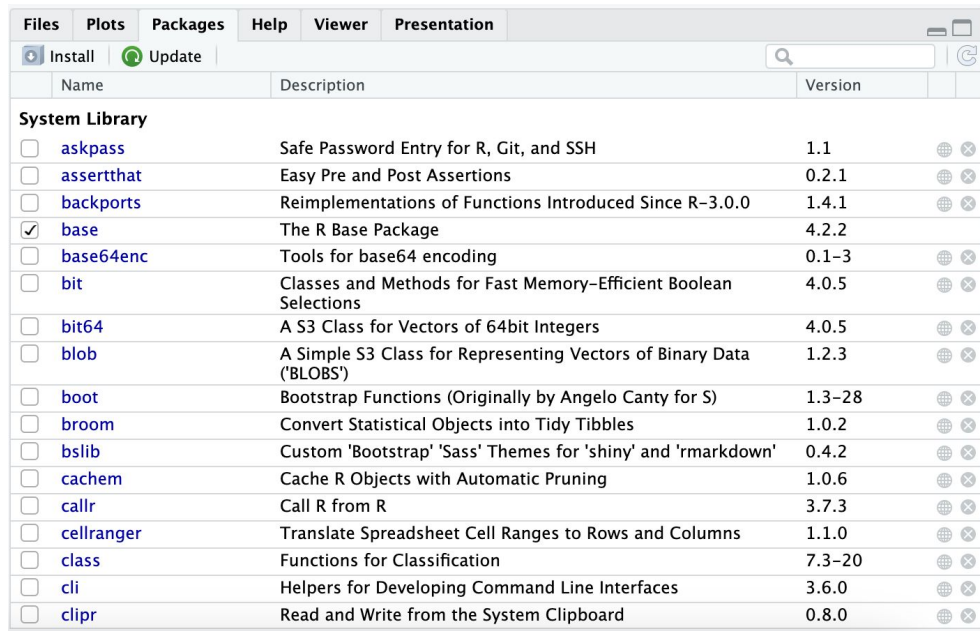
## After successfully download R and R studio...

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!!!

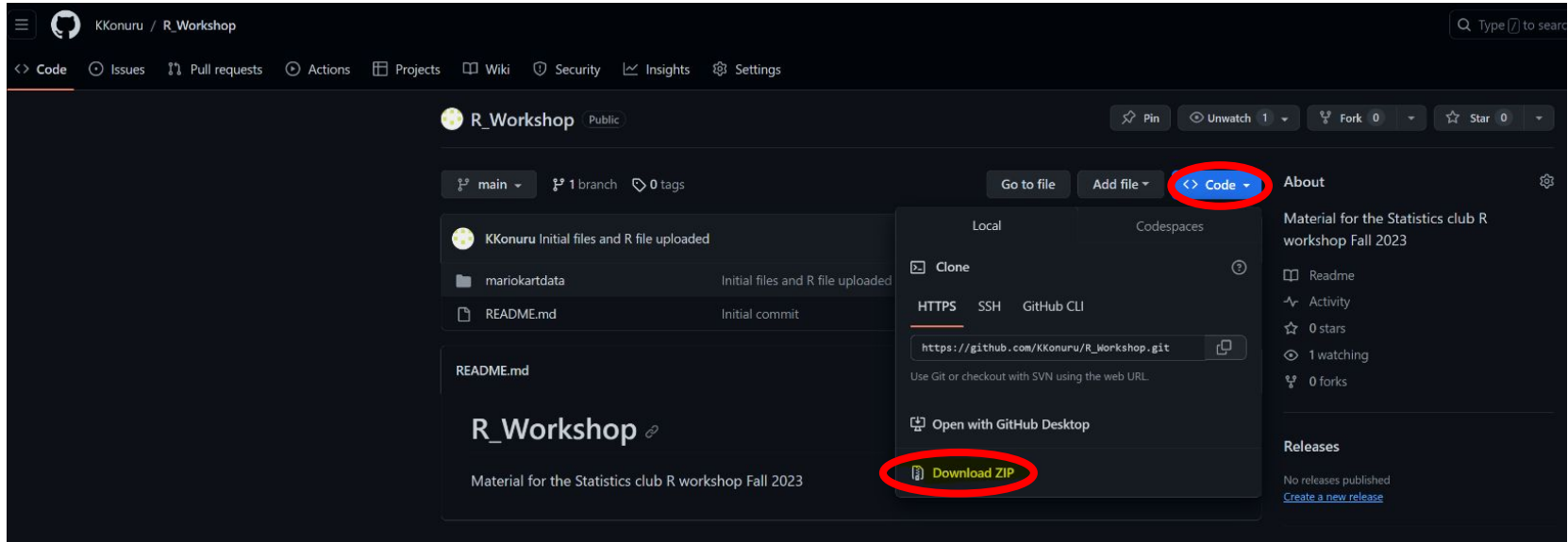


The screenshot shows the RStudio interface with the 'Packages' pane open. The 'System Library' section is visible, listing various installed and available packages. The 'base' package is checked, indicating it is installed. Other packages like 'askpass', 'assertthat', 'backports', 'base64enc', 'bit', 'bit64', 'blob', 'boot', 'broom', 'bslib', 'cachem', 'callr', 'cellranger', 'class', 'cli', and 'clipr' are listed with their descriptions and versions.

Name	Description	Version
<b>System Library</b>		
<input type="checkbox"/> askpass	Safe Password Entry for R, Git, and SSH	1.1
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.1
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0	1.4.1
<input checked="" type="checkbox"/> base	The R Base Package	4.2.2
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3
<input type="checkbox"/> bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.0.5
<input type="checkbox"/> bit64	A S3 Class for Vectors of 64bit Integers	4.0.5
<input type="checkbox"/> blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.3
<input type="checkbox"/> boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28
<input type="checkbox"/> broom	Convert Statistical Objects into Tidy Tibbles	1.0.2
<input type="checkbox"/> bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'rmarkdown'	0.4.2
<input type="checkbox"/> cachem	Cache R Objects with Automatic Pruning	1.0.6
<input type="checkbox"/> callr	Call R from R	3.7.3
<input type="checkbox"/> cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
<input type="checkbox"/> class	Functions for Classification	7.3-20
<input type="checkbox"/> cli	Helpers for Developing Command Line Interfaces	3.6.0
<input type="checkbox"/> clipr	Read and Write from the System Clipboard	0.8.0

# Getting ready to code

1. First visit this github link: [https://github.com/KKonuru/R\\_Workshop/](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

```
#Lets examine the first 6 lines of the code
```

```
```{r}  
head(kart)  
```
```

- 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

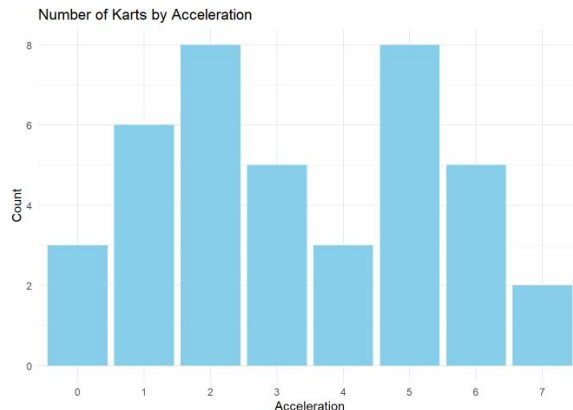
# Analysis of Karts dataset (cont.)

- 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

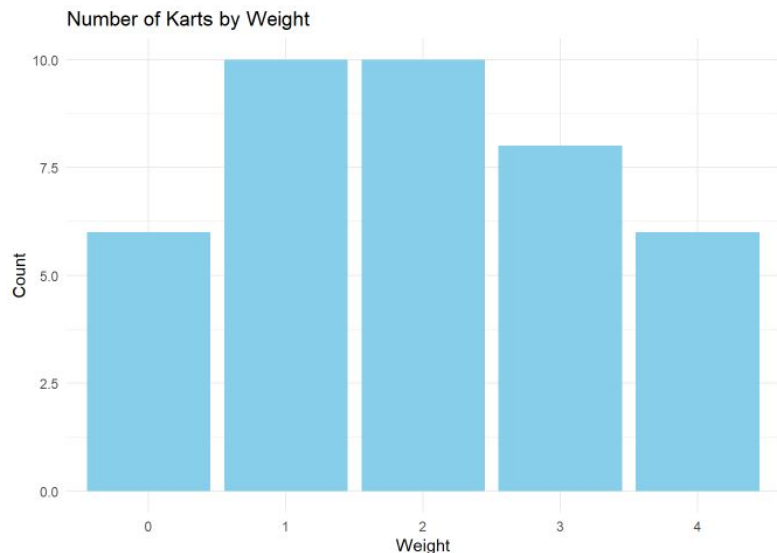
- We can use ggplot to generate bar graphs of the number of karts given
  - First we do `acc_counts <- as.data.frame(table(kart$Acceleration))` where `table()` makes a frequency table of acceleration and we convert it to a dataframe
  - Next we do `colnames(acc_counts) <- c("Acceleration", "Count")` 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](#)



**CREDITS:** This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

