# MATERIEL COVERED

CLASS 1 : BASIC DATA STRUCTURES, VARS, DATA FRAMES, SEQ, TYPE\_OF, CHAR\_VECTORS

CLASS 2: LIBRARIES, ARRANGE, SELECT, FILTER, TIBBLS FROM TIDYVERSE; MUTATE

CLASS 3: FILTER, GROUPING, MUTATE, ADVACNED TIDYVERSE

CLASS 4: RECAP, FACTOR VARS, SIMPLE GGPLOT (HISTOGRAMS, COLUMN CHARTS, SUMMARISE)

CLASS 5: Scatterplot/geom\_point, geom\_line, chart bifurcation with facet\_wrap,

CLASS 6: Recap, Names(), viewing data

Class 7: Plot groups, colour args, point sizes, labs()/labels, theme & visuals,

Class 8: Syntax errors, groupings, paste command

Class 9: Joins intro

Class 10: Joins, nObs, - see prep file

Class 11: Subsetting Vectors, I think I missed this

Class 12: Paste to append, %in% review, and read.csv plus file paths in R

Class 13: Gather, Spread, separate, combine, tidy data

Class 14: if statements,

Class 15: All, any, if\_else, case\_whens

# LOCATIONS

Repo for class (Hari’s): <https://github.com/puzhu/R4D>

My class repo: <https://github.com/noosphereintern/R4D-class>

Cheat sheets: <https://www.rstudio.com/resources/cheatsheets/>

Tidyverse portal: <https://www.tidyverse.org/>

# \*\*Keyboard Shortcuts\*\*:

1. Assignment operator (<-) is `Alt+-` for Windows and `Option+-` for Mac

2. Insert new code chunk `Ctrl+Alt+I` for Windows and `Command+Option+I` for Mac

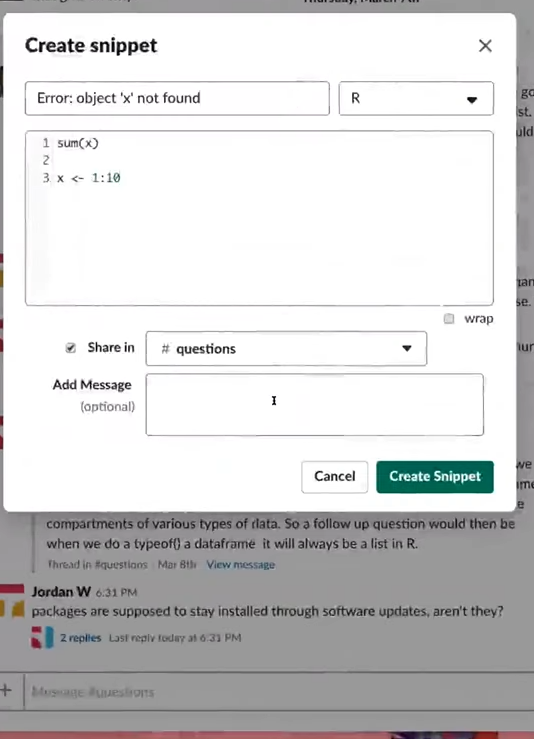
3. Run a line of code `Ctrl+Enter` for Windows and `Command+Enter` for Mac

4. Run all the code within a chunk `Ctrl+Shift+Enter` for Windows and `Command+Shift+Enter` for Mac

5. Insert a pipe operator (%>%) `Ctrl+Shift+M` for Windows and `Command+Shift+M` for Mac

6. Command + Shift + C will comment all the selected lines

# Errors in Slack:



Use the + sign at the bottom of slack to get this menu. Code in main, error in title, post in questions.

# RSTUDIO POINTERS

Note- Global Environment button shows packages loaded, it’s a subtab of the environment tab

Edit > COLLAPSE ALL OUTPUT what a lifesaver.

# R MARKDOWN HINTS

Renaming your RMD will eventually rename your html file, but you need to rerun all your chunks and/or preview first.

# Run All Chunks In a Notebook

You don't even have to use purl(): if you knit the document in the R console, the code is evaluated in the global environment (by default, see the envir= option to knit()).

So, if your file is my.Rmd, then just run

library(knitr)

knit('my.Rmd')

A handy trick: if you want to only run up to a certain point in the document, insert an error like:

stop('here')

at the point in a code chunk you want it to stop, and set the following knitr option:

opts\_chunk$set(error=FALSE)

in the console before running knit().

# GITHUB STUFF

##GITHUB STEPS

create local repo

OPTIONS: Git ignore "R"

MIT License

Publish to github

Repos are public

Here’s what you REALLY NEED to know about R on git: downloading individual files is stupid hard. Generally, you have to download the repo.

If you can figure out the direct URL path – not the one you get with the GUI nav – to an nb.html file, it will render with a link to download the notebook. DIRECT URLS to RMDS WILL FAIL.

HOW TO SUBMIT ASSIGNMENTS

Push

#Use settings to HOST part of my repo, a particular folder or file

#then you can make a URL out of the repo plus the html file name of your notebok

#the link is the url of your repo plus the file path, terminating

#in the notebook html file name

#video segment at 917PM or so of today's thing (CLASS 5 VIDEO)

IMPORTANT NOTE ABOUT HOSTING FILES. FIRST IS "COPY PATH BUTTON" when you're in the code UI.

Next, here is the root folder for my repo:

**https://noosphereintern.github.io/R4D-class/**

then for rendering any one file, just add the relative path from here.

Note this is NOT what you see in the URL to your browser Github GUI:

Which is this: **https://github.com/noosphereintern/R4D-class**

# CLASS 1 POINTERS

IMPORTANT: JUST TYPE THE VARIABLE NAME TO PRINT IT

Data types include atomic vectors, data frames, lists, and factors, among others.

RSTUDIO: Broom button clears environment

R expects homogenous data types in atomic vectors and coerces to get them

*“Under the hood, a data frame is a list of equal-length vectors. This makes it a 2-dimensional structure, so it shares properties of both the matrix and the list. This means that a data frame has names(), colnames(), and rownames(), although names() and colnames() are the same thing. The length() of a data frame is the length of the underlying list and so is the same as ncol(); nrow() gives the number of rows.”*

# CLASS 2 POINTER: Dots and Pipes

flights %>%

arrange(desc(dep\_delay))

arrange( ., desc(dep\_time))

```

note that you need to describe what you’re arranging (i.e., an input, a subject)

unless you have a pipe. then it knows it's your last pipe’s output

the period syntax is where you'd put the input, as 1st argument. It’s not needed here, but if you didn’t have a pipe, you’d need either a dot of the name of the dataframe?

COLON OPERATOR: Generate sequences. From:To. Numeric only.

# Row Numbers in tidyverse:

<https://stackoverflow.com/questions/25994307/filtering-data-frame-based-on-row-number>

row\_number() does return row numbers.

# CLASS 3 POINTER:

flights %>%

filter(dense\_rank(desc(arr\_delay)) %in% 10:40)

# %in% is a matching operator, see notes

#ties are eliminated b/c dense\_rank, ties broken by data order

Windowed rank operators, there’s various of them and I don’t understand the difference.

# RESOURCES:

https://r4ds.had.co.nz/