Class 4: Exploratory Data Anslysis

MGSC 310

Prof. Jonathan Hersh

Class 4: Announcements

1. Data Analytics Week! October
2. Problem Set 1 posted – Due Sept 15
   - Must submit compiled HTML file using Rmarkdown
3. Quiz due Thursday @ midnight
4. Be sure you are following along with the course reading

First Meeting

DAA

Come listen to our guest speakers

Dr. Seth Benzell & Dave Holtz

Using Data Science to Fight COVID-19

Two researchers discuss their studies and publications on their analysis of COVID-19 w/ Q&A

Sept. 8 @ 7:00pm PST

Zoom Meeting ID: https://chapman.zoom.us/j/5094919512
Password: DAAFA2020

Hello!

Thank you to everyone who came to our first meeting of the semester! We were able to listen to Dr. Seth Benzell and Dave Holtz talk about their work on COVID-19 papers as well as a little about their backgrounds! As requested, We have provided a **link to the meeting recording** for those who were unable to make it. The recording starts right when the speakers were introduced:
[goog_1863276205] https://drive.google.com/file/d/1-2XdN4j-jJS3uHJH8F_sXYcfbc0VDSAE/view?usp=sharing

Lastly, there is **contact form** that I would ask you to give to your friends if they are interested, or if you are *interested in joining the executive team*! Here is the link https://forms.gle/FaoiZzqB5MaGvn1E7

Reminder **our next meeting will be Tuesday, October 13th @ 7pm PDT.** Be on the lookout for emails on internship opportunities or updates till then!

Best,
DAA Executive Team

Register on Handshake to get access to the following virtual events!

Careers in Data Analytics
Tuesday, October 6 | 12 p.m. PST
Hear from the renowned authors of Build a Career in Data Science, Jacqueline Nolis and Emily Robinson about careers in data analytics.

Data Analytics Industry Panel
Thursday, October 8 | 4:30 p.m. PST
This data analytics panel will feature industry experts in analytics from entertainment, healthcare, technology, and more.

Entertainment Analytics: Turning Data Into Insights
Friday, October 9| 12 p.m. PST
Come see a live demo and learn about turning data into actionable insights in Entertainment Analytics with Andre Vargas Head of the data department at leading entertainment and sports agency, Creative Artists Agency (CAA).

CHAPMAN UNIVERSITY | Argyros School of Business and Economics

May Use Problem Set Rmarkdown Template

Problem Set 1 (R Programming)

Published Edit

See the problem set 1 instructions here MGSC310_pset1.pdf, MGSC310_pset1.html

You might find it useful to use the RMarkdown template available RMarkdown_Pset_Template.Rmd

Datasets:

- IMDB_movies.csv
- IMDB_movies.txt

Points	30
Submitting	a file upload

May Use Problem Set Rmarkdown Template

lab_class_4_R_Exploratory_Data_Analysi... | RMarkdown_Pset_Template.Rmd

Preview | Insert | Run

```
---
title: "Problem Set ?"
author: "Super smart students"
subtitle: MGSC 310, Fall 2020, Zoom Professor Hersh
output:
  html_notebook: default
  html_document:
    df_print: paged
---

```{r setup, include=FALSE}
```


```{r}
# if you get an error on loading libraries install package first
#
library(ISLR)
data(Auto)
plot(cars)
```

Question 1
1. *First things first, make sure you are working in a RStudio Project.*
2. Install any needed packages in the console. If you include any `install.packages` in a code chunk it will install every time, which you don't want.
3. Make sure you have installed the packages `here`, `fs`, `rmarkdown`, `tidyverse` and `ISLR`.
4. If you have problem knitting to PDF, knit a file to HTML! It looks just as good, even better when uploaded for your problem sets.
5. Knit to HTML when you are done. You can "preview notebook" if you want to quickly see how your code is doing.
6. If you want to put all your code inside one code chunk with your comments in a separate Word file, that's perfectly fine.
7. If at any point your RMarkdown doesn't compile, don't panic. Email me or the TAs for help. If you *need* to submit something quickly, just create a word file with all your figure and text/regression output. I will dock some points for the latter, but not as much as if you just send uncompiled or unexecuted code.

Question 2 (These are headings, note the '#')
This is text. I can write text just like this and it will come out as a paragraph.

If I want to do a bulleted list

* I
* can
* do
```

Warning: To Do Well In This Class You Have To Do The Reading

Tentative Schedule

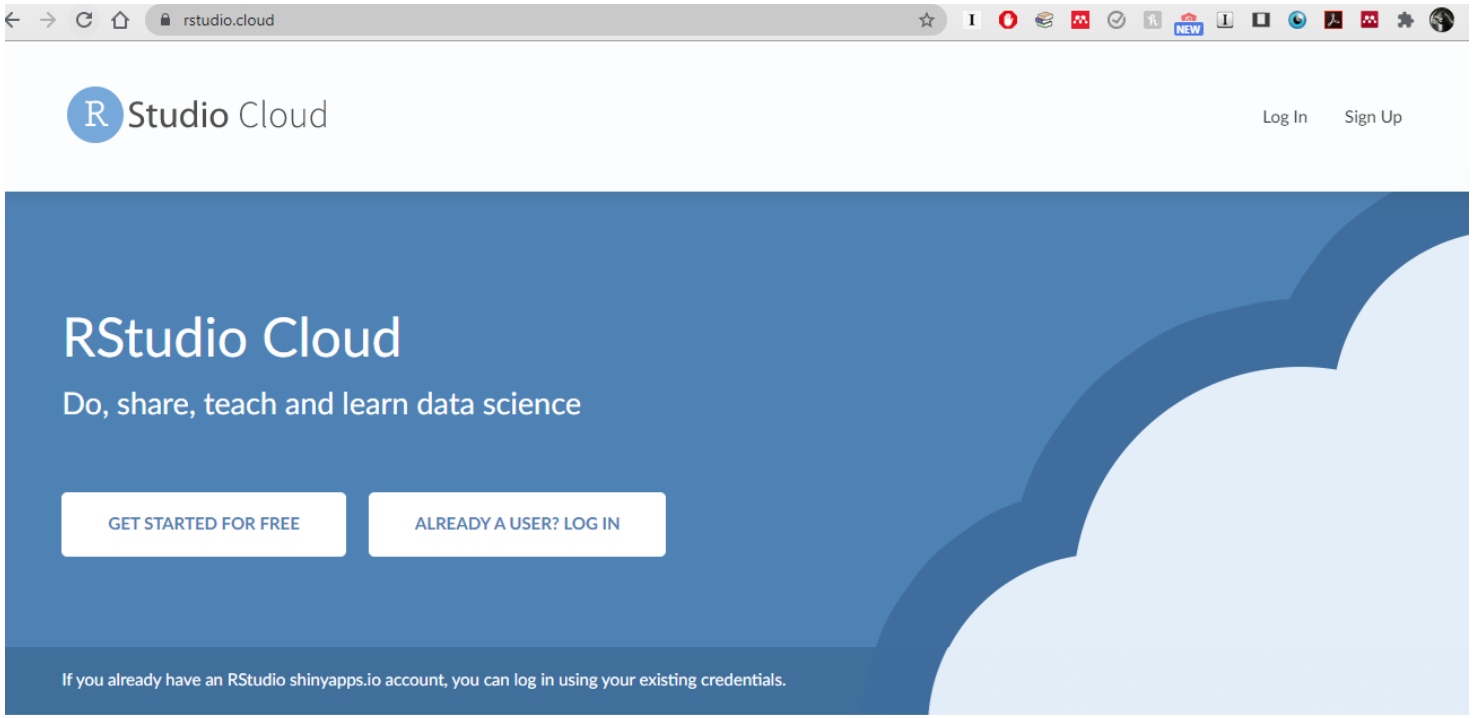
Date	Basic Topic	Topic	Text Reading Due	Assignment
Tue, Sep 1	Intro	Intro/Inference Vs Prediction		
Thu, Sep 3	Intro to R	Installing R, Installing and Loading Packages, Loading Datasets, Data Visualization in ggplot2	R for Everyone, Chp 1-3, 7	
Tue, Sep 8	Intro to R	Basic Data Types, and Advanced Data Structures, Functions, and Loops	R for Everyone, Chp 4-6, 8-9	
Thu, Sep 10	Intro to R	Exploratory Data Analysis and Data Manipulation with Dplyr	R for Everyone, Chp 12 ISLR: Chp 1	Quiz 1
Tue, Sep 15	Bias-Variance	Classification vs Regression, Assessing Model Accuracy, and Bias-Variance Trade-off	ISLR: pages 15-36	Problem Set 1
Thu, Sep 17	Linear Regression	Linear Regression 1:, Coefficient Hypothesis Testing, and Assessing Model Accuracy,	ISLR: pages 59-82	Quiz 2
Tue, Sep 22	Linear Regression	Linear Regression 2: Feature Engineering: Qualitative Predictors (Dummy Variables), Log Transformations, Squared Predictors, Interpreting Coefficients	ISLR: 82-92	

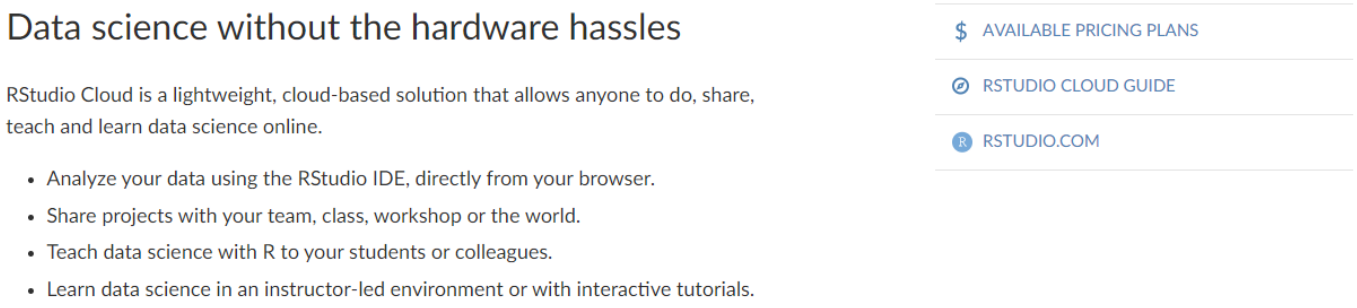
- Students ask me often how they can get an A in this course. Reading the course material is my #1 response.
- You will do better on problem sets, quizzes, midterm and projects
- Being a competent data scientists involves lots of reading.

Class 4: Outline

1. Qs from last week?
2. **Data Analysis**
   - Loading data
   - Glimpse to view
   - Pipe operator
   - slice() to select rows
   - arrange() to order data frame
   - select() to choose variables
   - rename() to rename variables
   - filter() to select rows matching characteristics
   - Remove duplicates with distinct
   - Outputting “clean” data file”
3. Data Analysis Lab Part 1
4. **Data Analysis by Groups**
   - group_by() function
   - summarize() to create group variables
5. Data Analysis Lab Part 2

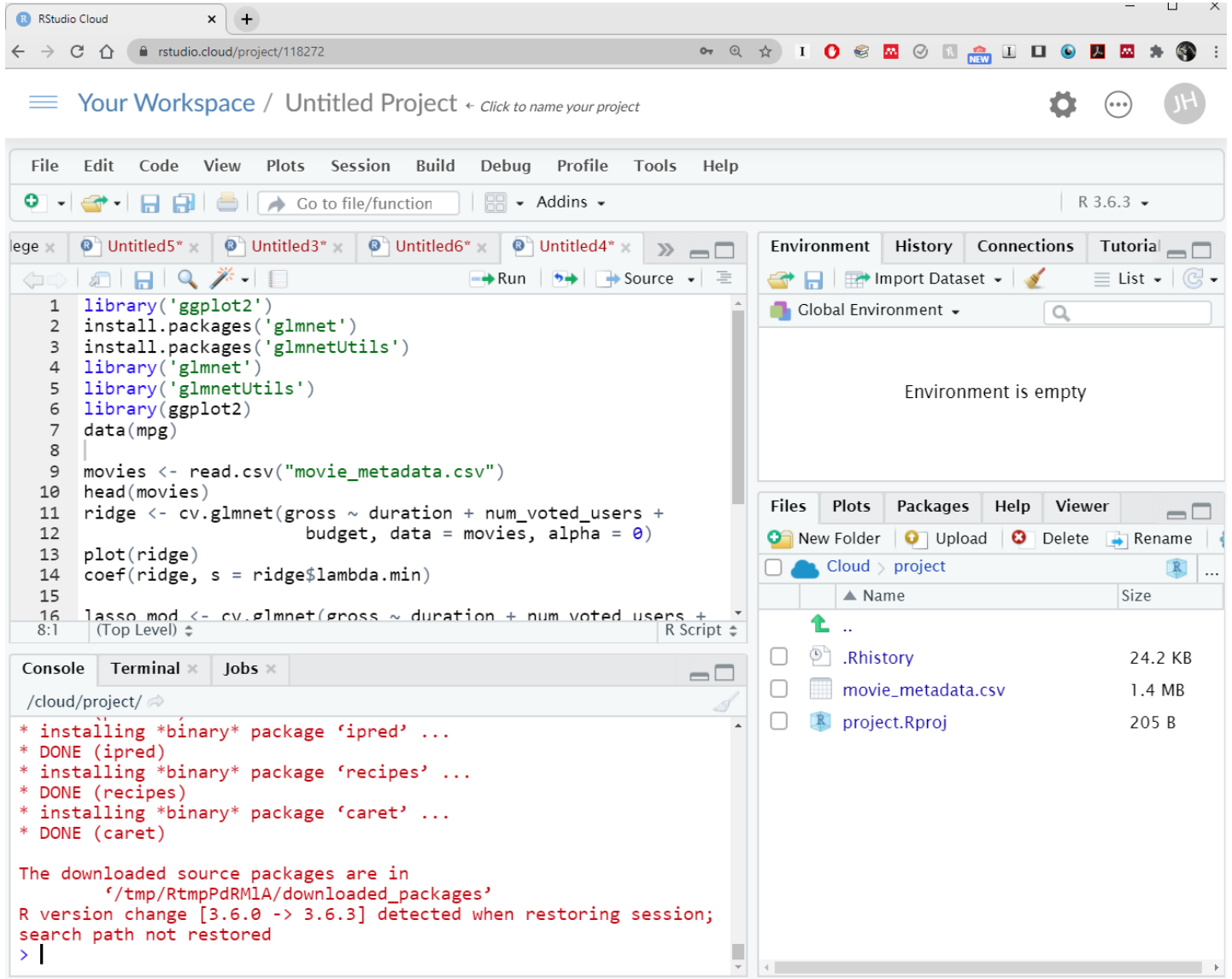
R Studio Cloud





- Go to rstudio.cloud if your version of R is ever not working

R Studio Cloud



- R Studio Cloud is a full featured version of R in your browser!

Loading Data

```r
# ----------------------------------------------
# Loading data
# ----------------------------------------------
library('tidyverse')

# the here package is very useful, it allows us to select across folders relative to our "home"
# directory of the project
# note here::here allows us to use the here function in the here package without loading it

# download the IMDB_movies.csv dataset here, and store it in a subfolder called "datasets"
# https://github.com/jonhersh/MGSC310/tree/master/datasets

# OR you can run the code below to
fs::dir_create(here::here("datasets"))

# this downloads a file from the net and stores it in your datasets folder
download.file("https://raw.githubusercontent.com/jonhersh/MGSC310/master/datasets/IMDB_movies.csv",
              here::here("datasets", "IMDB_movies.csv"),
              method = "curl",
              replace = TRUE)


movies <- read.csv(here::here("datasets", "IMDB_movies.csv"))
```

Glimpse to Summarize Data

```
# ---------------------------------------------
# GLIMPSE to summarize data
# ---------------------------------------------
# let's summarize the data using the glimpse function
glimpse(movies)
```

Pipe Operator

```
# ------------------------------------------------
# Pipe Operator!
# ------------------------------------------------
# The pipe operator "%>%" is super useful!
# It allows us to execute a series of functions on an object in stages
# The general recipe is Data_Frame %>% function1() %>% function2() etc
# Functions are applied right to left

movies %>% glimpse()
glimpse(movies)

# cmd shift

movies %>% glimpse()
glimpse(movies)
```

Slice to View Rows

```
# ---------------------------------------------
# slice function: to select ROWS
# ---------------------------------------------
# SLICE: slice to view only the first 10 rows
movies %>% slice(1:10)

# SLICE to view only rows 300 to 310
movies %>% slice(300:310)
```

Arrange function: to ORDER dataset

```
# ----------------------------------------------
# Arrange function: to ORDER dataset
# ----------------------------------------------

# arrange the dataframe in descening order by budget, and store this back as movies
movies <- movies %>% arrange(desc(budget))

# arrange the dataframe in ascending order by budget and store this back as movies
movies <- movies %>% arrange(desc(budget))

# arrange via multipe columns, by budget and title year, then output rows 1 to 10
movies %>%
  arrange(desc(budget), desc(title_year)) %>%
  slice(1:10)
```

SELECT columns of the dataset using the 'select' function

```
# ------------------------------------------------
# SELECT columns of the dataset using the 'select' function
# ------------------------------------------------
# selecting columns using the select() function
# here we create a subset of the original dataset that only contains director_name and movie title
movies_keys <- movies %>%  select(director_name, movie_title)
glimpse(movies_keys)

# using select to programmatically select several variables that 'start with' a certain string
movies_actors <- movies %>% select(starts_with("actor"))
glimpse(movies_actors)

# here we
# everything() is a useful function, and
movies <- movies %>% select(director_name, movie_title, title_year, everything())
glimpse(movies)
```

RENAME variables using the RENAME function

```
# ---------------------------------------------
# RENAME variables using the RENAME function
# ---------------------------------------------

# use the rename function to rename variables
movies <- movies %>%  rename(director = director_name)
glimpse(movies)
```

FILTER and ONLY allow certain rows using the FILTER function

```
# ----------------------------------------------
# FILTER and ONLY allow certain rows using the FILTER function
# ----------------------------------------------
# filter removes any rows that DO NOT meet the logical operator


# ONLY select large budget movies and store this as a new data frame
movies_big <- movies %>% filter(budget > 100000000)
glimpse(movies_big)

# ONLY select english language films and store this as a new data frame
movies_eng <- movies %>% filter(language == "English")
glimpse(movies_eng)
dim(movies_eng)
```

Exercises - Lab

1. What are the highest grossing Steven Spielberg films?
2. Print a dataframe that only lists the films with the highest 10 budgets, the movie title, and the country of origin (hint, use select).
3. How many "PG-13" movies are there in the database? (hint: use nrow())
4. Create a new dataframe called “movies_actors” that contains all the actor variables, and the movie title. (hint use select(starts_with(…))
5. Change the name of the variable “content_rating” to “rating”
6. Make 1-2 interesting ggplots using the movies dataset

Missing Values

```r
# ---------------------------------------------
# MISSING VALUES are values that are unknown in your dataset
# ---------------------------------------------
# R stores missing values as as NAs
is.na(NA)
1 > NA
1 + 1 == NA
NA == NA
y <- NA
y
x <- 1
y == x
```

lab_class_4_R_Exploratory_Data_Analysi... | movies

Filter

	actor_1_facebook_likes	gross	genres	actor_1_name
	11000	200074175	Action|Adventure|Thriller	Christoph Waltz
	27000	448130642	Action|Thriller	Tom Hardy
	131	NA	Documentary	Doug Walker
	640	73058679	Action|Adventure|Sci-Fi	Daryl Sabara
	24000	336530303	Action|Adventure|Romance	J.K. Simmons
	799	200807262	Adventure|Animation|Comedy|Family|Fantasy|Musical|...	Brad Garrett

Loops in R

```
# ---------------------------------------------
# LOOP through numbers using the FOR loop
# ---------------------------------------------

# for loops are created using the synthax
# for(i in start:end){
# do something with i
# }
```

```
> for(i in 1:10){
+   print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

LOOP through numbers using the FOR loop

```
# how to see how many missings you have in each column?
# well, we want to sum through every column using a for loop
# then print the variable name using names(movies[i])
# then print the sum of is.na() for just that variable

# for each column in the movies
for(i in 1:ncol(movies)){

  # print the following
  print(

    # first print "Variable: "
    paste0("Variable: ",

           # then print the variable name, then "NAs: "
           names(movies)[i], " NAs: ",

           # then print the sum of the number of missing values
           # for that variable
           sum(is.na(movies %>% select(i)))
          )

        )
}
```

Functions in R

```r
# ---------------------------------------------
# Creating functions
# ---------------------------------------------
# we create a function in R by writing
# function_name <- function(input1, input2,...){
  # function arguments
# }

print_names <- function(data_frame){
  print(names(data_frame))
}
```

```
> print_names(movies)
 [1] "color"                    "director_name"            "num_critic_for_reviews"
 [4] "duration"                 "director_facebook_likes"  "actor_3_facebook_likes"
 [7] "actor_2_name"             "actor_1_facebook_likes"   "gross"
[10] "genres"                   "actor_1_name"             "movie_title"
[13] "num_voted_users"          "cast_total_facebook_likes" "actor_3_name"
[16] "facenumber_in_poster"     "plot_keywords"            "movie_imdb_link"
[19] "num_user_for_reviews"     "language"                 "country"
[22] "content_rating"           "budget"                   "title_year"
[25] "actor_2_facebook_likes"   "imdb_score"               "aspect_ratio"
[28] "movie_facebook_likes"
```

Build a function that prints number of missing values for each variable

```r
# Let's take the code we wrote above and translate
# it to a function called "num_missing".
# We can then call the function and pass our movies dataframe
# to it to export
num_missing <- function(data_frame){
  for(i in 1:ncol(movies)){
    print(
      paste0("Variable: ",
             names(movies)[i], " NAs: ",
             sum(is.na(movies %>% select(i)))
      )

    )
  }
}
```

```
> num_missing(movies)
[1] "Variable: color NAs: 0"
[1] "Variable: director_name NAs: 0"
[1] "Variable: num_critic_for_reviews NAs: 50"
[1] "Variable: duration NAs: 15"
[1] "Variable: director_facebook_likes NAs: 104"
[1] "Variable: actor_3_facebook_likes NAs: 23"
[1] "Variable: actor_2_name NAs: 0"
[1] "Variable: actor_1_facebook_likes NAs: 7"
[1] "Variable: gross NAs: 884"
[1] "Variable: genres NAs: 0"
[1] "Variable: actor_1_name NAs: 0"
[1] "Variable: movie_title NAs: 0"
[1] "Variable: num_voted_users NAs: 0"
[1] "Variable: cast_total_facebook_likes NAs: 0"
[1] "Variable: actor_3_name NAs: 0"
[1] "Variable: facenumber_in_poster NAs: 13"
[1] "Variable: plot_keywords NAs: 0"
[1] "Variable: movie_imdb_link NAs: 0"
[1] "Variable: num_user_for_reviews NAs: 21"
[1] "Variable: language NAs: 0"
[1] "Variable: country NAs: 0"
[1] "Variable: content_rating NAs: 0"
[1] "Variable: budget NAs: 492"
[1] "Variable: title_year NAs: 108"
[1] "Variable: actor_2_facebook_likes NAs: 13"
[1] "Variable: imdb_score NAs: 0"
[1] "Variable: aspect_ratio NAs: 329"
[1] "Variable: movie_facebook_likes NAs: 0"
```

MUTATE to Transform variables in your dataset

```r
# ----------------------------------------------
# MUTATE to Transform variables in your dataset
# ----------------------------------------------

# adding new variables using mutate()
# note %<>% == DF <- DF %>%
# let's create new varibles budgetM and grossM that
# are budget and gross in units of millions
movies %<>% mutate(budgetM = budget/1000000,
                   grossM = gross/1000000,
                   profitM = grossM - budgetM)

movies %>% glimpse()

# so it looks like there's some outliers
# The most expensive movie ever made was Pirates of
# the Caribbean: On Stranger Tides
# which cost $387.8m. Any movies with a budget higher
# than this must be a data anomaly

# Let's use the filter command to remove these

movies_clean <- movies %>% filter(budgetM < 400)
```

Find Duplicate Rows with duplicated()

```
# ----------------------------------------------
# Find Duplicate Rows with duplicated()
# and find_duplicates() (must install hablar package)
# ----------------------------------------------
# number of duplicated rows
movies %>% duplicated() %>% sum()

# view duplicated rows
# install.packages(hablar)
movies %>% hablar::find_duplicates()
```

Output final clean version of dataset

```r
# ------------------------------------------------
# Output final clean version of dataset
# ------------------------------------------------
# remove duplicate rows, create new budget and gross variables,
# rename director and title
# remove budgets greater than 400M,
# order title, year, budget, director and gross first, then store in new file
movies_clean <-
  movies %>%
  distinct() %>%
  mutate(budgetM = budget/1000000,
         grossM = gross/1000000,
         profitM = grossM - budgetM) %>%
  rename(director = director_name,
         title = movie_title,
         year = title_year) %>%
  relocate(title, year, country, director, budgetM, grossM, imdb_score) %>%
  filter(budgetM < 400)

movies_clean %>% glimpse()
```

- Generally we do pre-processing on our dataset starting from a raw file.
- After these transformations we save a “clean” version of the dataset that is used for analysis

Create summary statistics by GROUP using group_by()

```
# ------------------------------------------------
# Create summary statistics by GROUP using group_by()
# ------------------------------------------------
# group summaries using summarise and group_by
director_avg <-
  movies_clean %>%
    # group_by() is used to indicate the grouping variable
  group_by(director) %>%

    # summarize creates a new variable based on this group
    # here we create averages by director using the 'mean'
    # function
  summarize(gross_avg_director = mean(grossM, na.rm = TRUE))


# view results
director_avg %>% arrange(-gross_avg_director) %>% print()
```

Create averages, count and standard deviation by groups

```
# ---------------------------------------------
# Create grouped variables using the Summarize function
# n() creates counts by
# sd() creates standard deviations
# ---------------------------------------------
# let's create budget by director, gross by director, profit by director,
# number films by director
director_df <-
  movies_clean %>%
  group_by(director) %>%
  summarize(

      # create average budget by director
      budget_avg_director = mean(budgetM, na.rm = TRUE),
      # create average gross by directory
      gross_avg_director = mean(grossM, na.rm = TRUE),
      # create average movie profit by director
      profit_avg_director = mean(profitM, na.rm = TRUE),
      # create variable that lists number of films
      # by director
      num_films = n(),
      # create a standard deviation of profit
      # by director
      profit_sd_director = sd(profitM, na.rm = TRUE)

      )
```

Exercises - 2

1. Print a dataframe with the film title, director name, and number of films for the 10 directors with the most films in the dataset
2. Print a dataframe with all of George Lucas' films ordered by budget
3. Why do some directors have "NA" for profit_sd? How many directors have Nas for profit_sd?
4. Print a scatter plot of average budget against average profit for the top 20 directors by average profit
5. Make 1-2 more interesting ggplots using the director_df
6. What movie genres have the highest average profit? (hint, must use a new group_by() command)
7. What movie ratings have the highest average profit?