Welcome to DATA 151

I’m so glad you’re here!

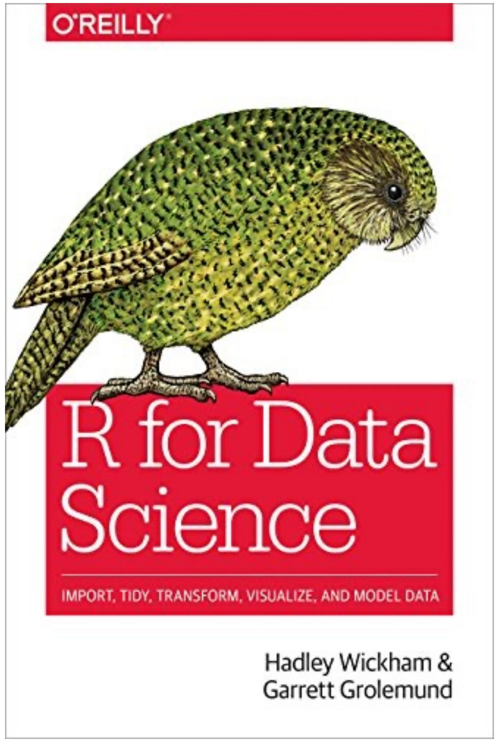
DATA 151: CLASS 7A
INTRODUCTION TO DATA SCIENCE (WITH R)

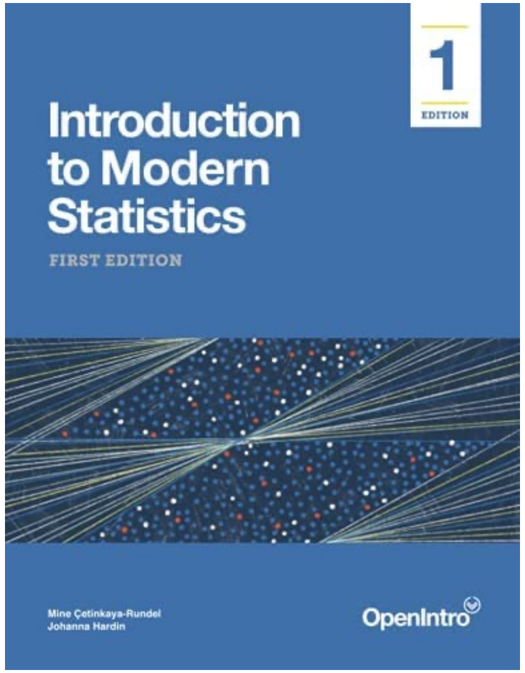
EXPLORATORY DATA ANALYSIS

NOTES PREPARED BY PROF. KITADA SMALLEY (FALL 2022)

ANNOUNCEMENTS

RELEVANT READING





Introduction to Data Science:

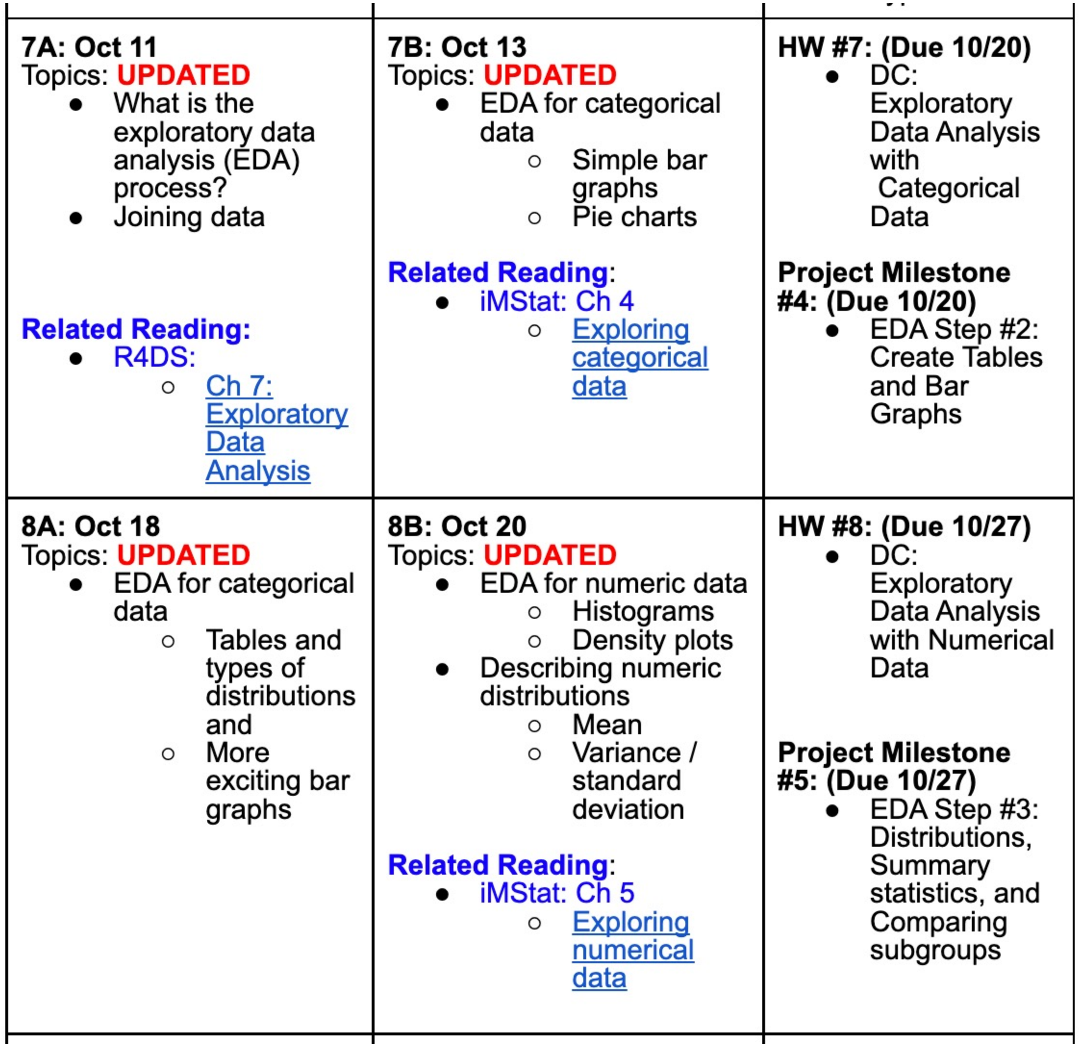
- Tuesday:
  - R for Data Science
  - Ch 7: Exploratory Data Analysis
- Thursday:
  - Introduction to Modern Statistics
  - Ch 4: Exploring Categorical Data

HOMEWORK REMINDER

Due this/next week: (EXTENSION DUE 10/17)

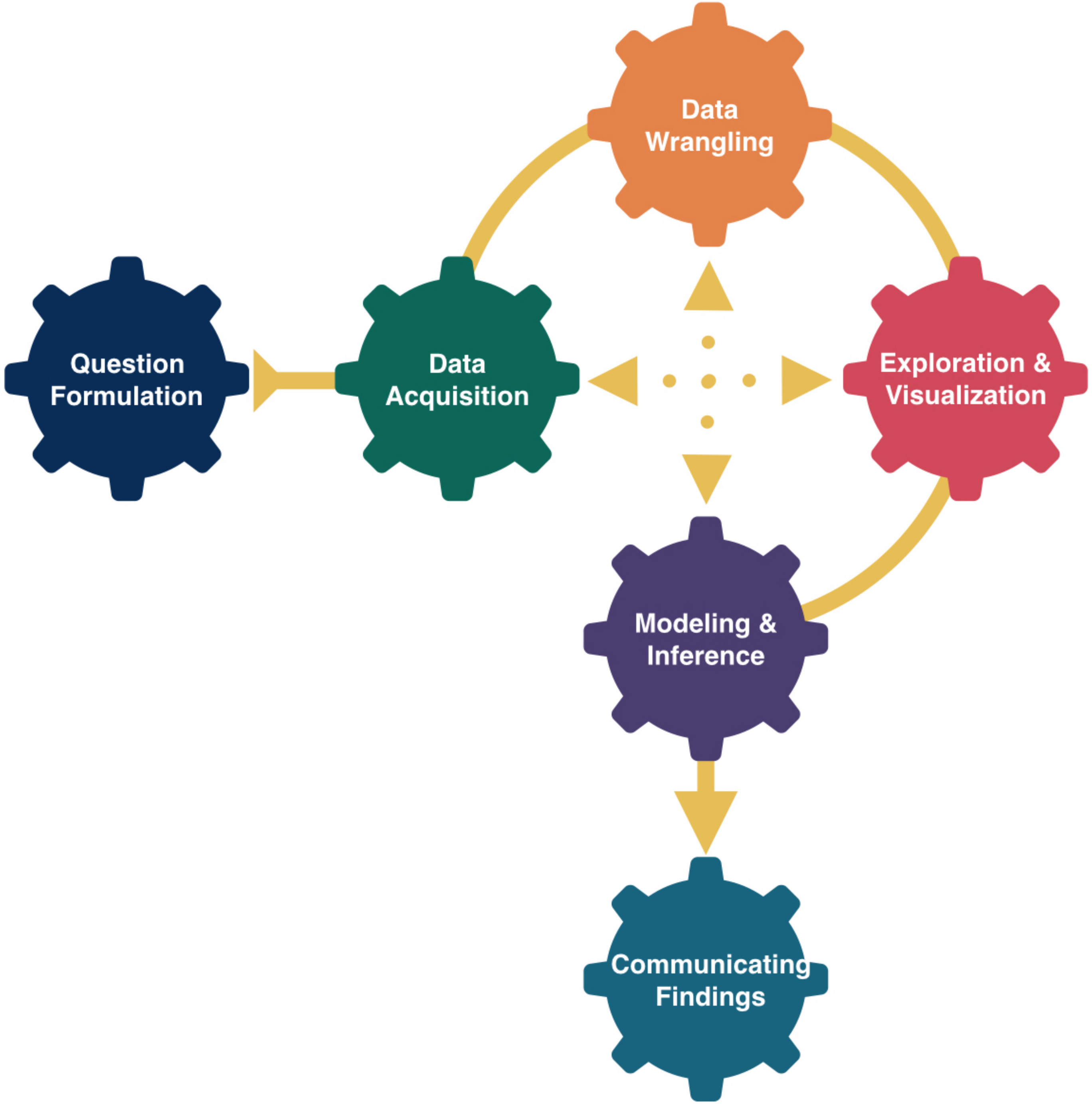
- *HW #6: DC Introduction to Data Visualization in ggplot2*
    - ***No submission on WISE necessary, do on DataCamp***
- *Project Milestone #3: EDA Step 1*
    - Ask questions and form hypotheses

UPDATES TO CALENDAR TOPICS



7A: Oct 11 Topics: **UPDATED** ● What is the exploratory data analysis (EDA) process? ● Joining data **Related Reading:** ● R4DS: ○ Ch 7: Exploratory Data Analysis	**7B: Oct 13** Topics: **UPDATED** ● EDA for categorical data ○ Simple bar graphs ○ Pie charts **Related Reading**: ● iMStat: Ch 4 ○ Exploring categorical data	**HW #7: (Due 10/20)** ● DC: Exploratory Data Analysis with Categorical Data **Project Milestone #4: (Due 10/20)** ● EDA Step #2: Create Tables and Bar Graphs
8A: Oct 18 Topics: **UPDATED** ● EDA for categorical data ○ Tables and types of distributions and ○ More exciting bar graphs	**8B: Oct 20** Topics: **UPDATED** ● EDA for numeric data ○ Histograms ○ Density plots ● Describing numeric distributions ○ Mean ○ Variance / standard deviation **Related Reading**: ● iMStat: Ch 5 ○ Exploring numerical data	**HW #8: (Due 10/27)** ● DC: Exploratory Data Analysis with Numerical Data **Project Milestone #5: (Due 10/27)** ● EDA Step #3: Distributions, Summary statistics, and Comparing subgroups

EXPLORATORY DATA ANALYSIS



Kelly McConville

AN ITERATIVE CYCLE

EDA is an iterative cycle. You:

1. Generate questions about your data.
2. Search for answers by visualising, transforming, and modelling your data.
3. Use what you learn to refine your questions and/or generate new questions.

AN ITERATIVE CYCLE

"EDA is not a formal process with a strict set of rules. More than anything, EDA is a state of mind."

QUESTIONS TO ASK YOURSELF

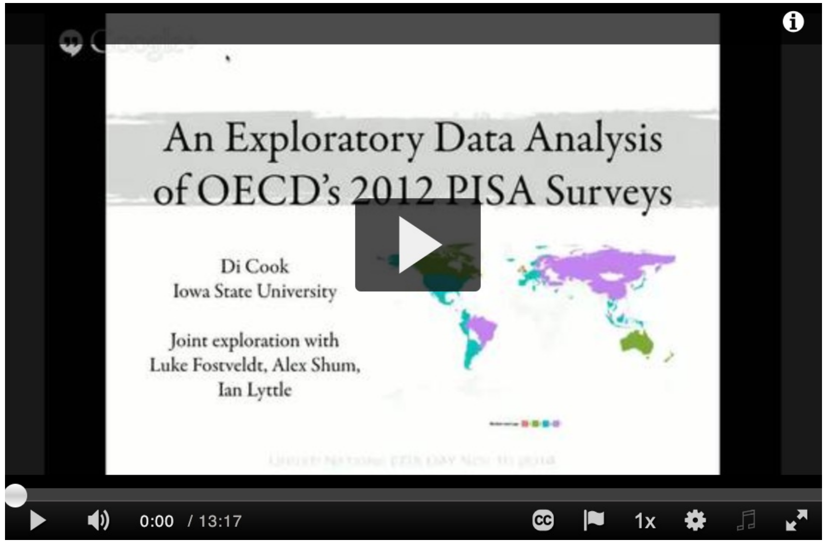
1. What type of variation occurs within my variables?
2. Which values are the most common? Why?
3. Which values are rare? Why? Does that match your expectations?
4. Can you see any unusual patterns? What might explain them?
5. What type of covariation occurs between my variables?

PHILOSOPHY AND STRATEGY OF EDA

Watch (after class) the following excerpt (~ 12mins) from a workshop on EDA given at the UN. Di Cook talks about EDA with respect to an OECD data set on education.

LINK

What strategies does she suggest for Exploratory Data Analysis?



An Exploratory Data Analysis of OECD's 2012 PISA Surveys

Di Cook
Iowa State University

Joint exploration with
Luke Fostveldt, Alex Shum,
Ian Lyttle

PHILOSOPHY AND STRATEGY OF EDA

Di suggests two key strategies:

1. **Write down your expectations ahead of time** This gives you a starting point for things to look at. Try to verify your expectations of the data, but be prepared to be surprised.

PHILOSOPHY AND STRATEGY OF EDA

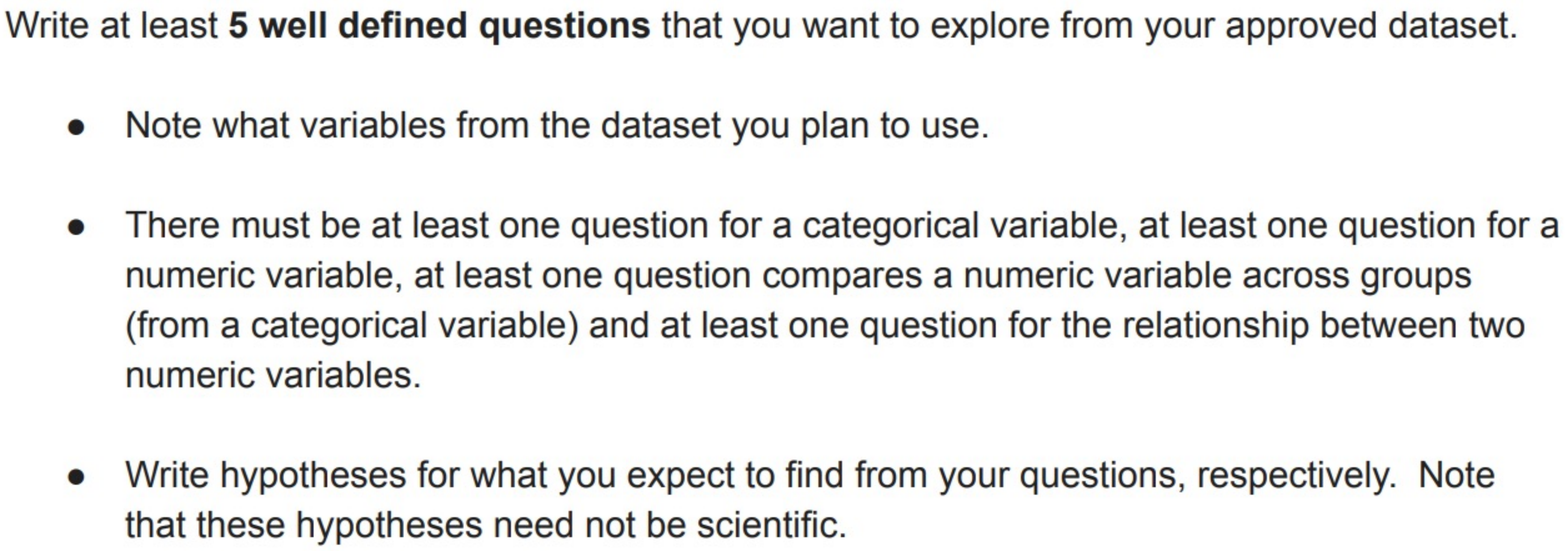
2. **Show the data** Don't over-process the data. Start with the rawest data possible, then refine it according to what you see (either to refine a question, or make a clearer display).

PHILOSOPHY AND STRATEGY OF EDA

3. **Note what surprises you** You can sometimes get pretty involved in an analysis and forgot how you got where you did. It's important to make notes along the way.

MEET WITH YOUR GROUP

MILESTONE #3- QUESTIONS OF INTEREST



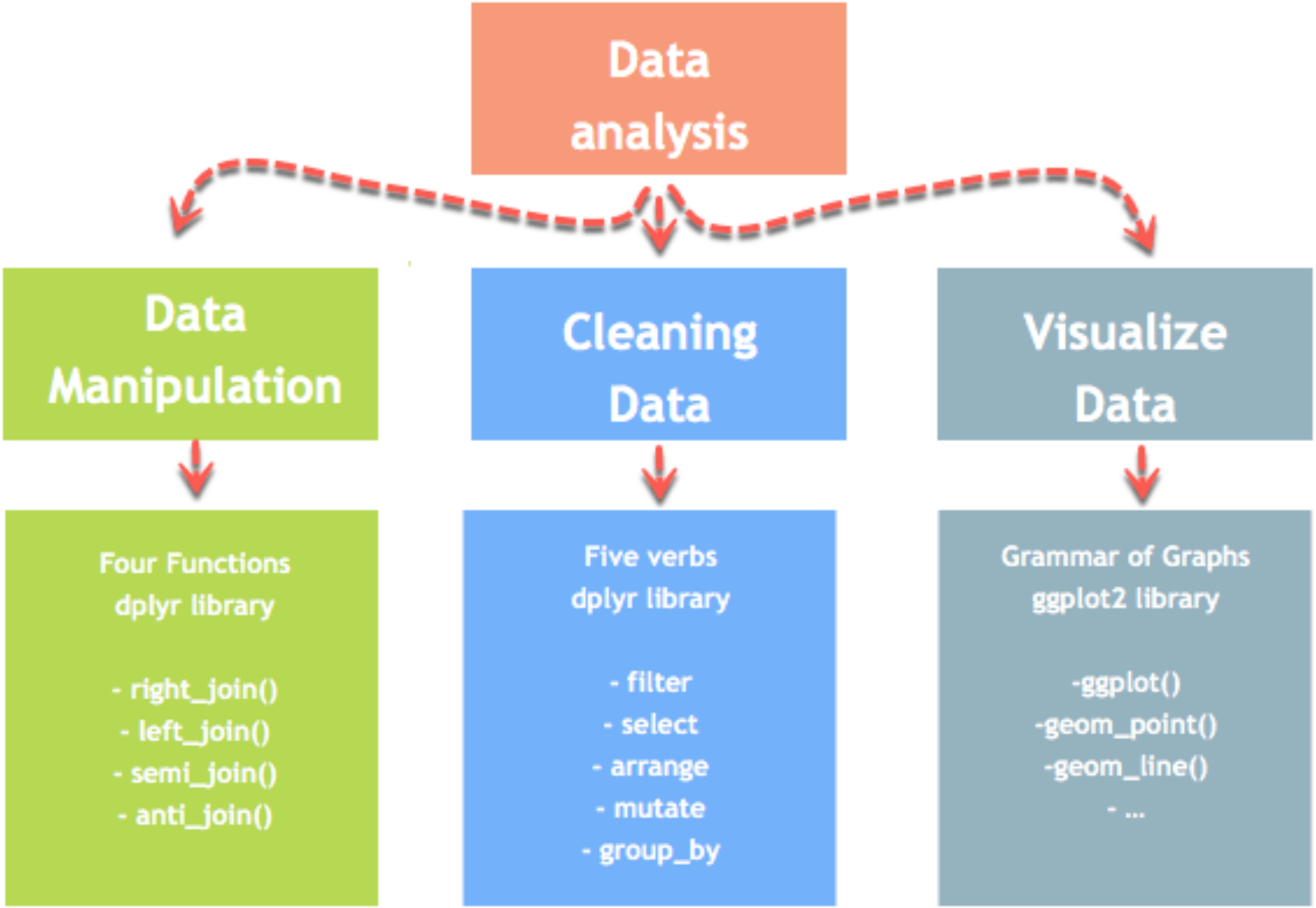
Write at least **5 well defined questions** that you want to explore from your approved dataset.

- Note what variables from the dataset you plan to use.

- There must be at least one question for a categorical variable, at least one question for a numeric variable, at least one question compares a numeric variable across groups (from a categorical variable) and at least one question for the relationship between two numeric variables.

- Write hypotheses for what you expect to find from your questions, respectively. Note that these hypotheses need not be scientific.

BACK TO JOINS…



Data analysis

Data Manipulation

Four Functions
dplyr library

- right_join()
- left_join()
- semi_join()
- anti_join()

Cleaning Data

Five verbs
dplyr library

- filter
- select
- arrange
- mutate
- group_by

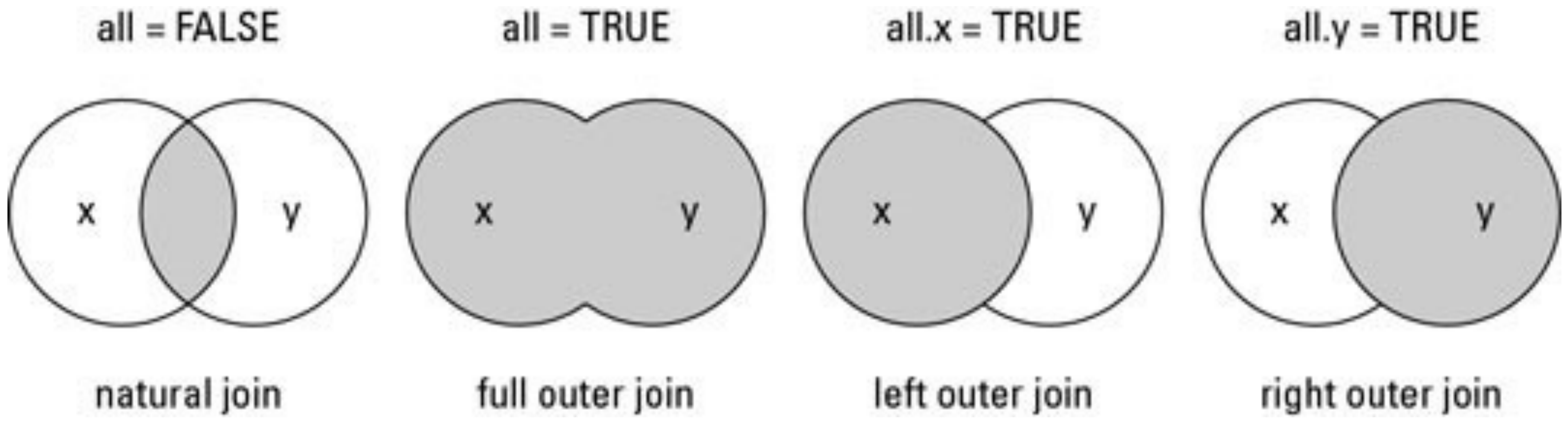
Visualize Data

Grammar of Graphs
ggplot2 library

-ggplot()
-geom_point()
-geom_line()
- ...

JOINS

- There are four types of join methods that can be used in R:
  - Left, right, inner, and full



- Note: the natural join is called “inner” join in R

JOINS

- **Natural join**: To keep only rows that match from the data frames, specify the argument all=FALSE.

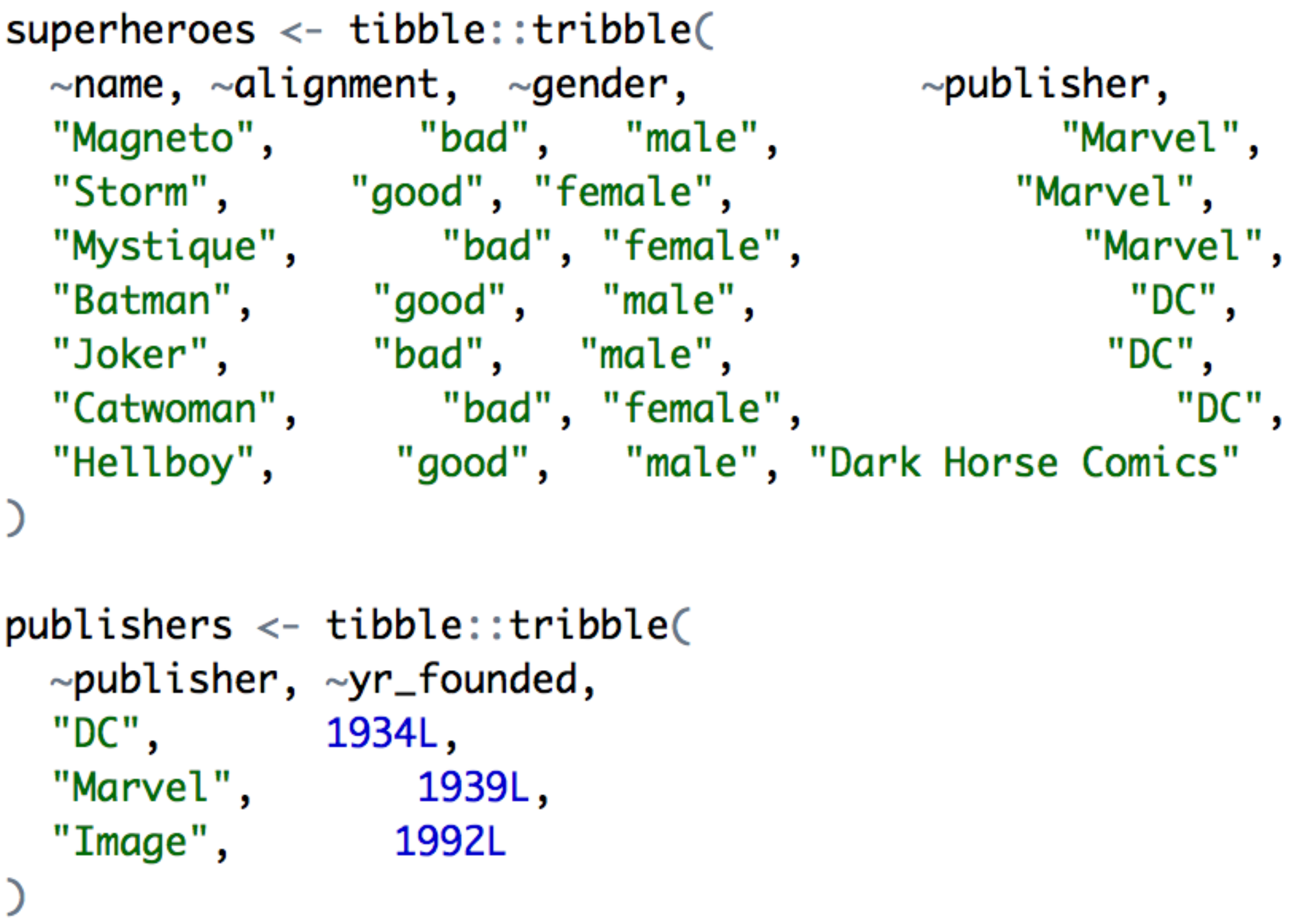
- **Full outer join:** To keep all rows from both data frames, specify all=TRUE.

- **Left outer join:** To include all the rows of your data frame x and only those from y that match, specify x=TRUE.

- **Right outer join:** To include all the rows of your data frame y and only those from x that match, specify y=TRUE.

TOY EXAMPLE FOR JOINS

JOINS

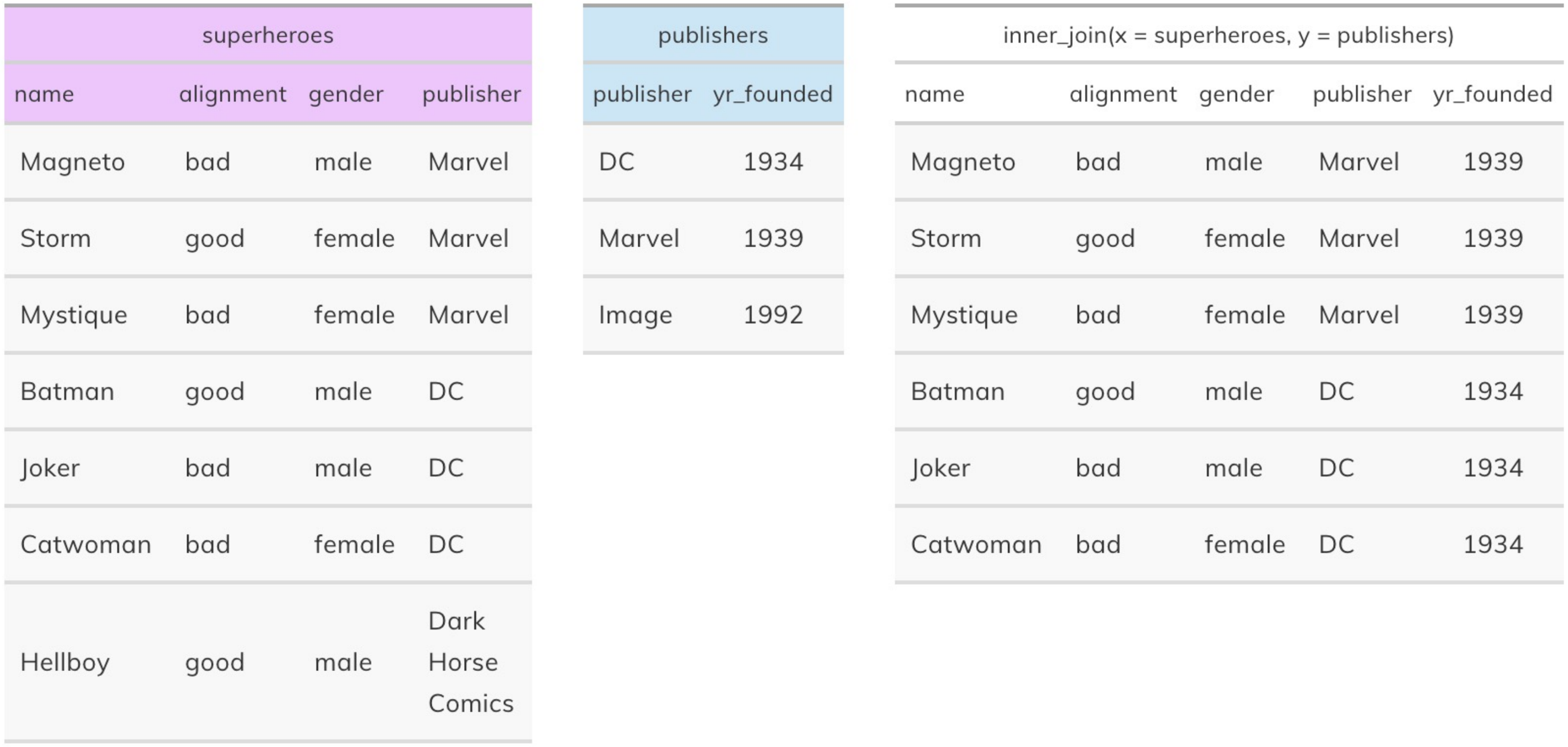


```
superheroes <- tibble::tribble(
  ~name, ~alignment,  ~gender,          ~publisher,
  "Magneto",      "bad",   "male",            "Marvel",
  "Storm",     "good", "female",            "Marvel",
  "Mystique",      "bad", "female",            "Marvel",
  "Batman",     "good",   "male",                "DC",
  "Joker",      "bad",   "male",                "DC",
  "Catwoman",      "bad", "female",                "DC",
  "Hellboy",     "good",   "male", "Dark Horse Comics"
)

publishers <- tibble::tribble(
  ~publisher, ~yr_founded,
  "DC",       1934L,
  "Marvel",       1939L,
  "Image",       1992L
)
```

JOINS

```
# inner join super hero and publisher
insp<-inner_join(superheroes, publishers)
insp
```



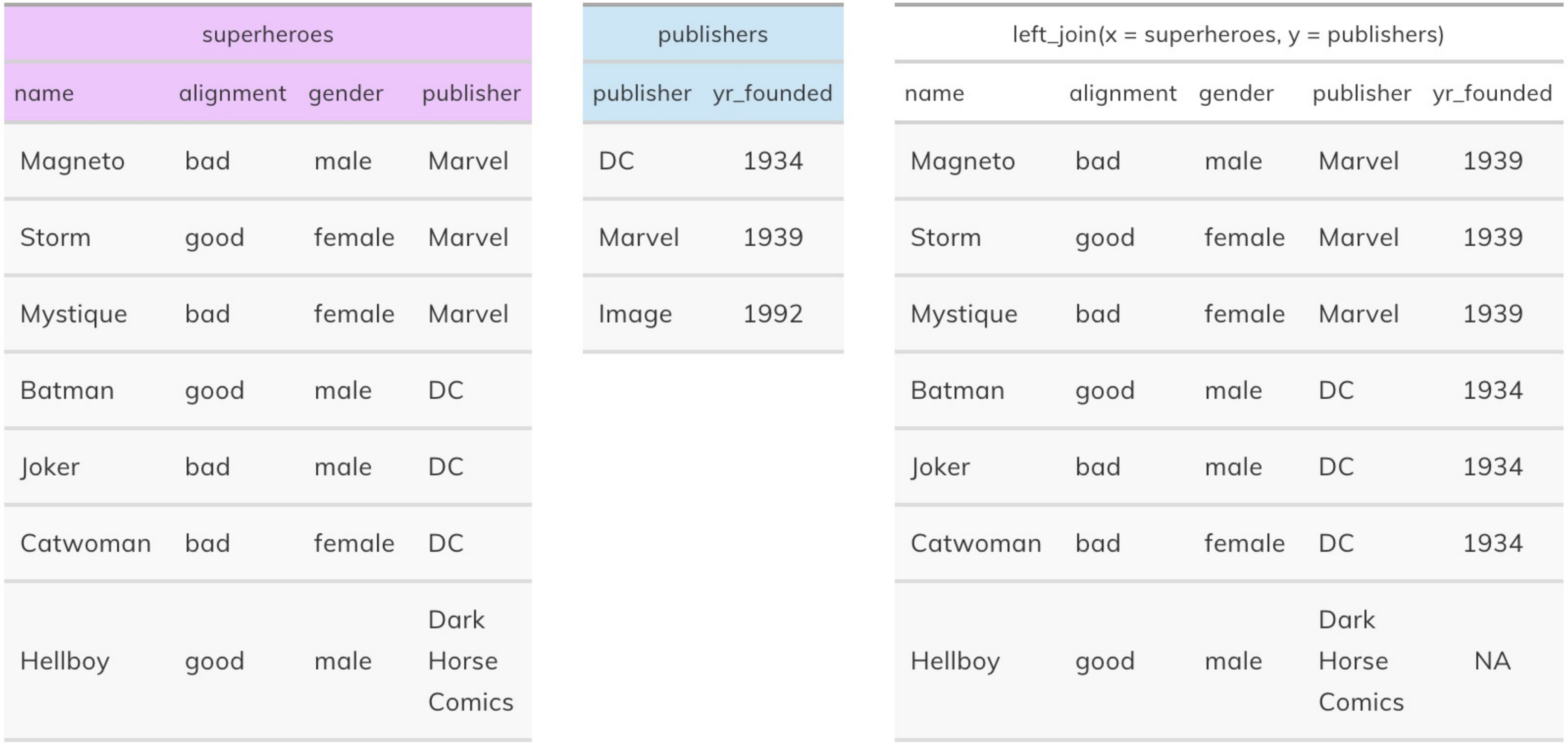
superheroes			
name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers	
publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

inner_join(x = superheroes, y = publishers)				
name	alignment	gender	publisher	yr_founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934

JOINS

```
# left join super hero and publisher
ljsp<-left_join(superheroes, publishers)
ljsp
```



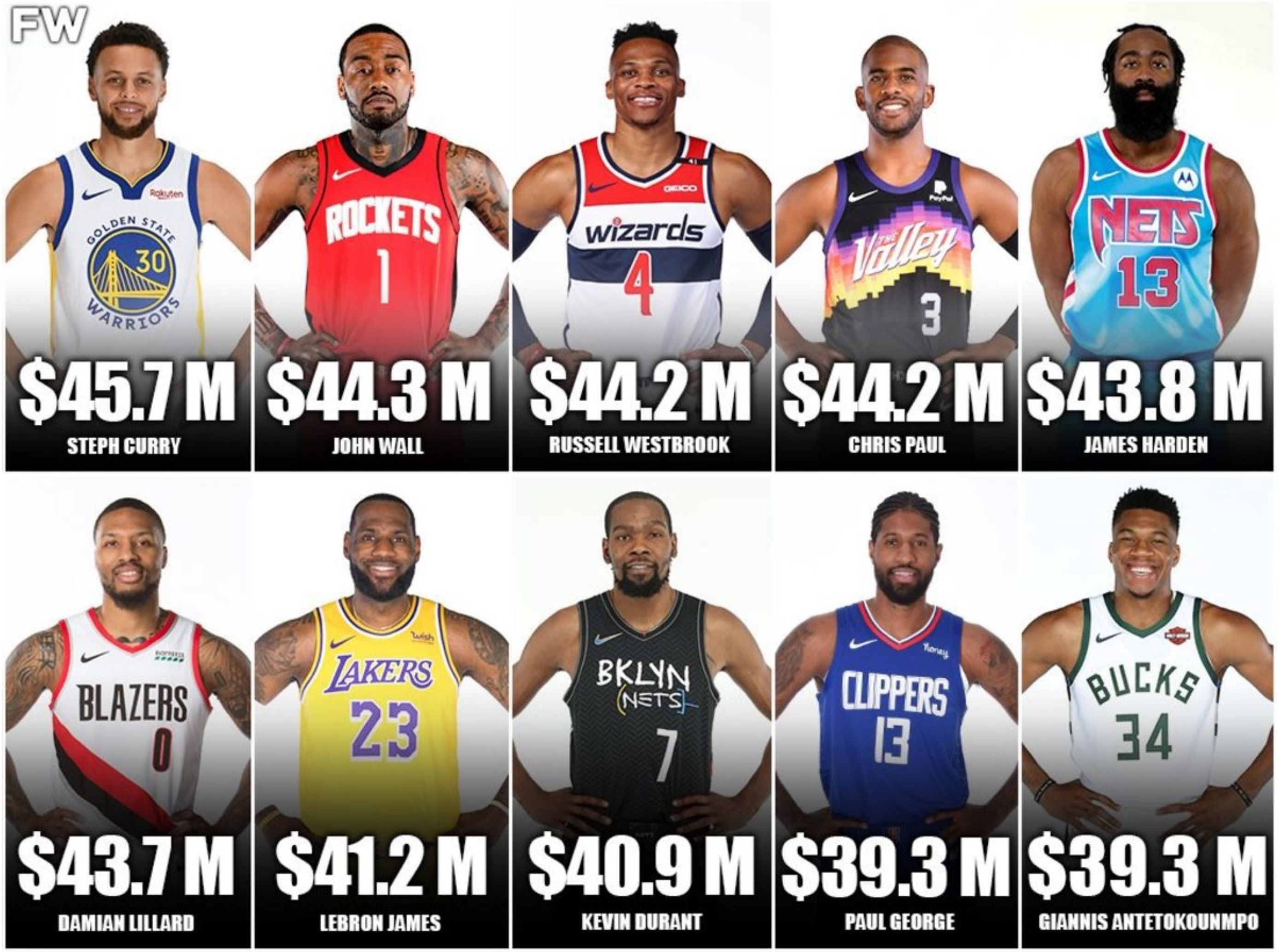
superheroes			
name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers	
publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

left_join(x = superheroes, y = publishers)				
name	alignment	gender	publisher	yr_founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934
Hellboy	good	male	Dark Horse Comics	NA

REAL WORLD EXAMPLE: JOINS

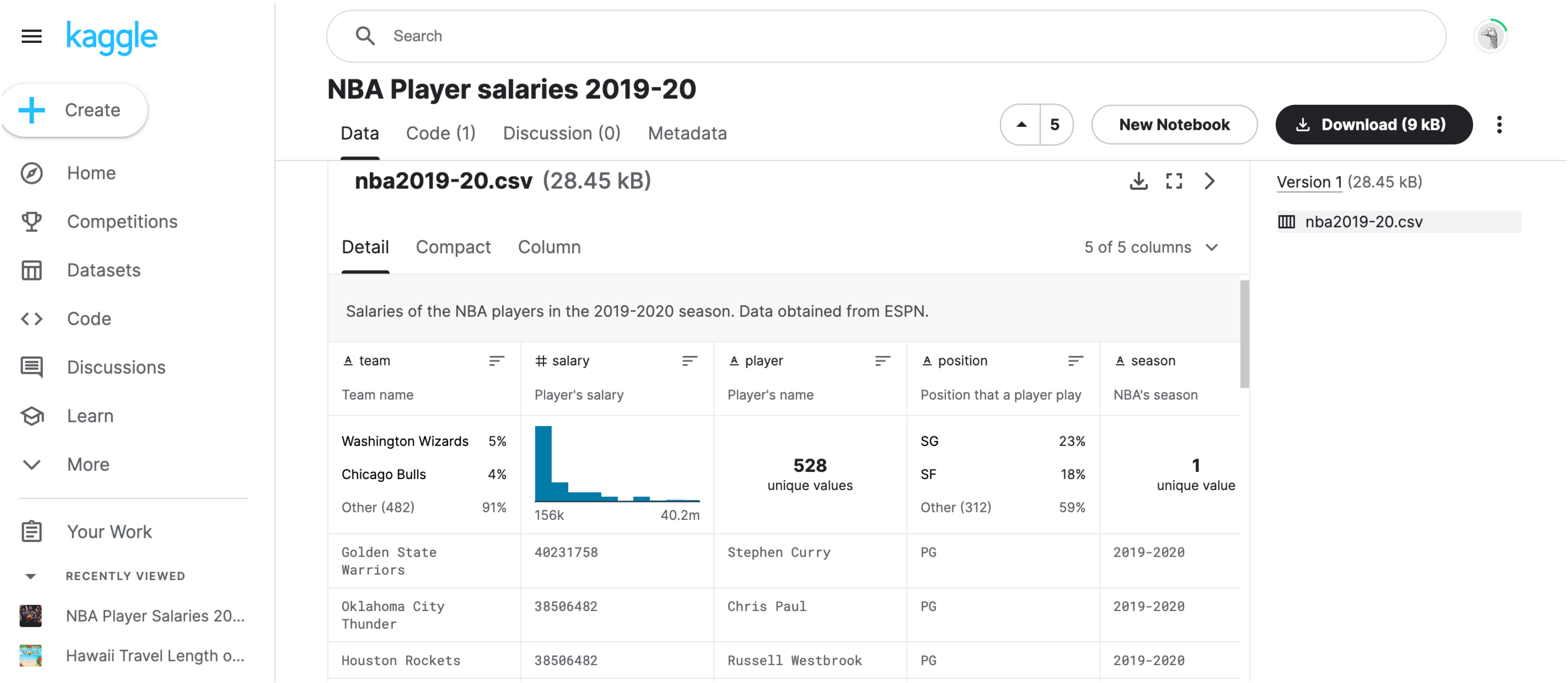
REAL WORLD EXAMPLE: JOINS



MOTIVATING QUESTION

Is player salary related to player performance?

STEP 0: DOWNLOAD THE DATA



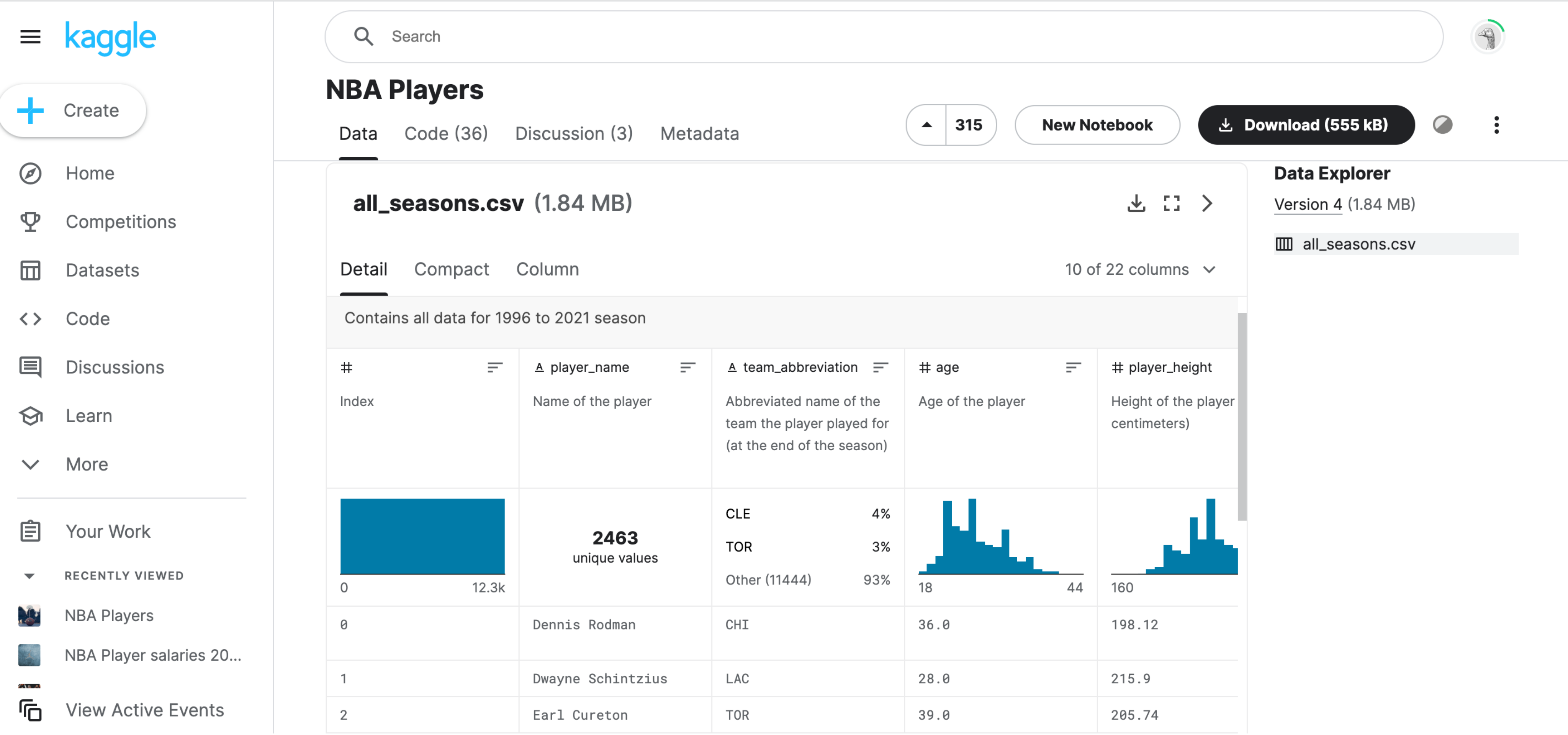
NBA Player salaries 2019-20

nba2019-20.csv (28.45 kB)

Salaries of the NBA players in the 2019-2020 season. Data obtained from ESPN.

team	salary	player	position	season
Team name	Player's salary	Player's name	Position that a player play	NBA's season
Washington Wizards 5% Chicago Bulls 4% Other (482) 91%		528 unique values	SG 23% SF 18% Other (312) 59%	1 unique value
Golden State Warriors	40231758	Stephen Curry	PG	2019-2020
Oklahoma City Thunder	38506482	Chris Paul	PG	2019-2020
Houston Rockets	38506482	Russell Westbrook	PG	2019-2020

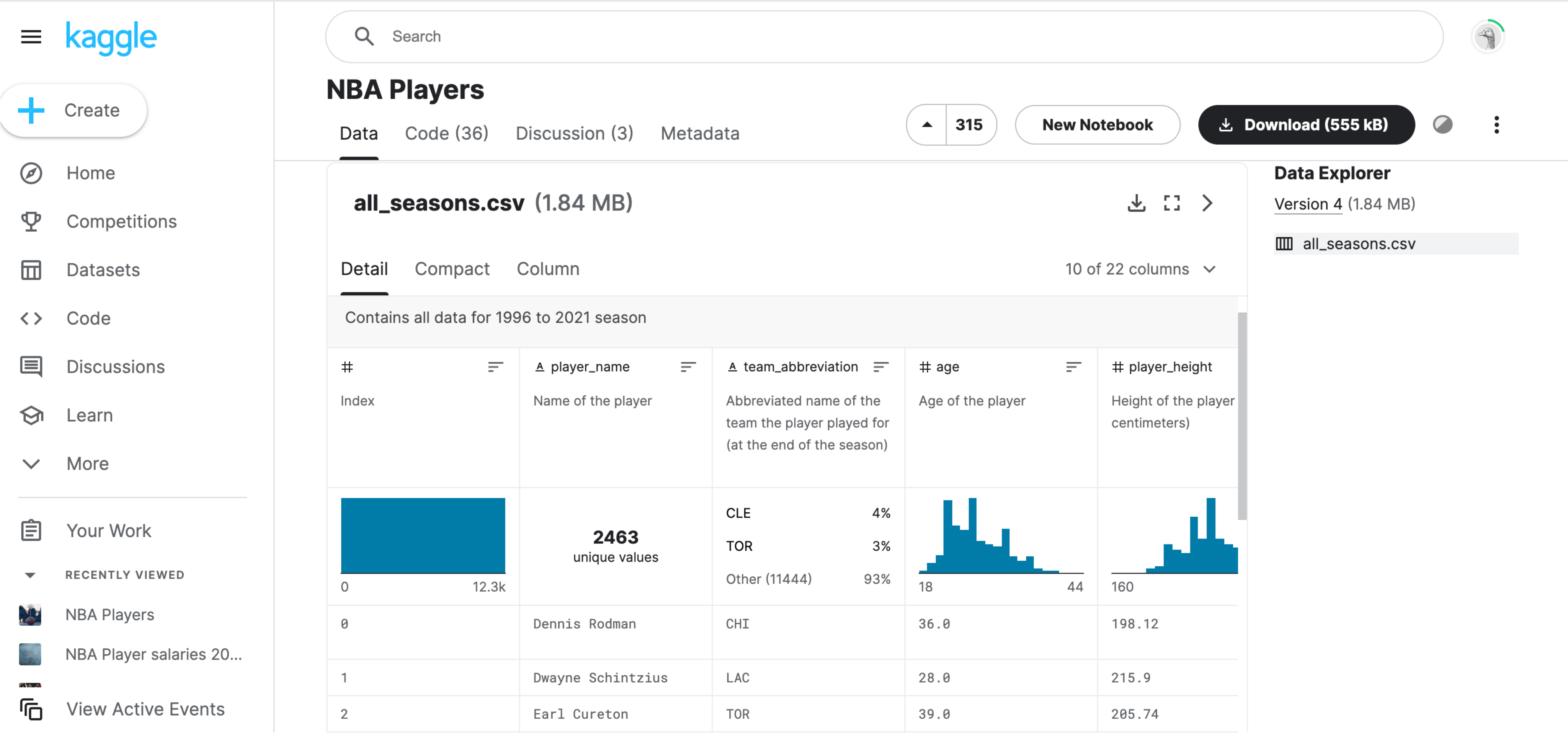
STEP 0: DOWNLOAD THE DATA



Contains all data for 1996 to 2021 season

#	player_name	team_abbreviation	age	player_height
Index	Name of the player	Abbreviated name of the team the player played for (at the end of the season)	Age of the player	Height of the player centimeters)
0 12.3k	2463 unique values	CLE 4% TOR 3% Other (11444) 93%	18 44	160
0	Dennis Rodman	CHI	36.0	198.12
1	Dwayne Schintzius	LAC	28.0	215.9
2	Earl Cureton	TOR	39.0	205.74

STEP 0: DOWNLOAD THE DATA



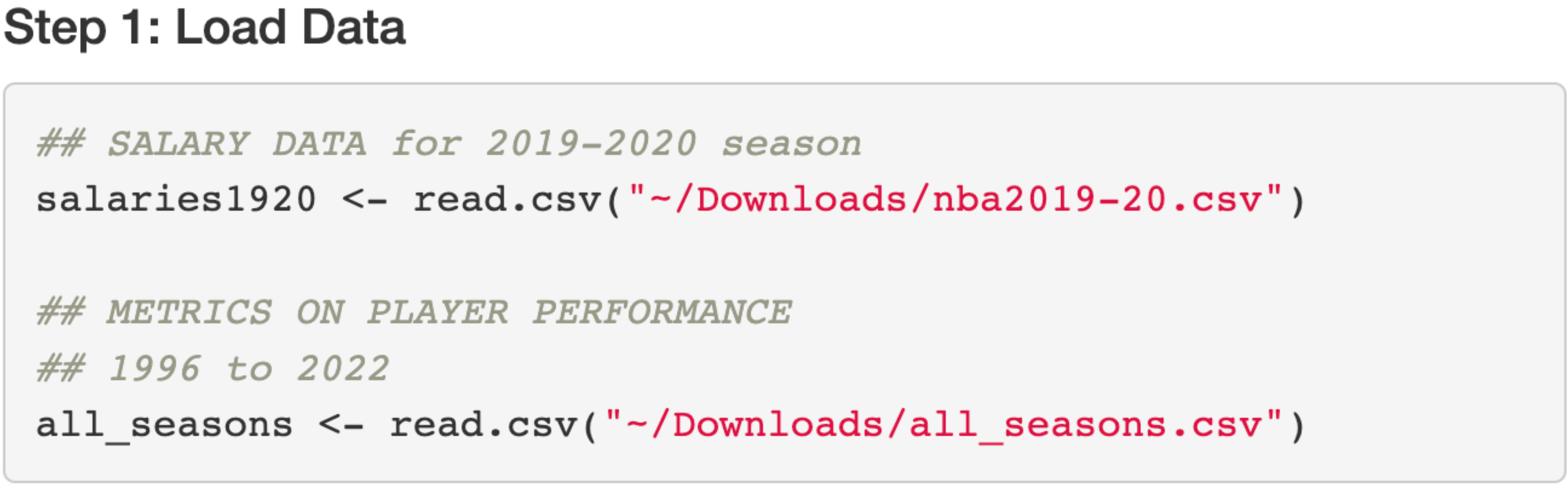
NBA Players

all_seasons.csv (1.84 MB)

Contains all data for 1996 to 2021 season

#	player_name	team_abbreviation	age	player_height
Index	Name of the player	Abbreviated name of the team the player played for (at the end of the season)	Age of the player	Height of the player centimeters)
0 12.3k	2463 unique values	CLE 4% TOR 3% Other (11444) 93%	18 44	160
0	Dennis Rodman	CHI	36.0	198.12
1	Dwayne Schintzius	LAC	28.0	215.9
2	Earl Cureton	TOR	39.0	205.74

STEP 1: LOAD THE DATA

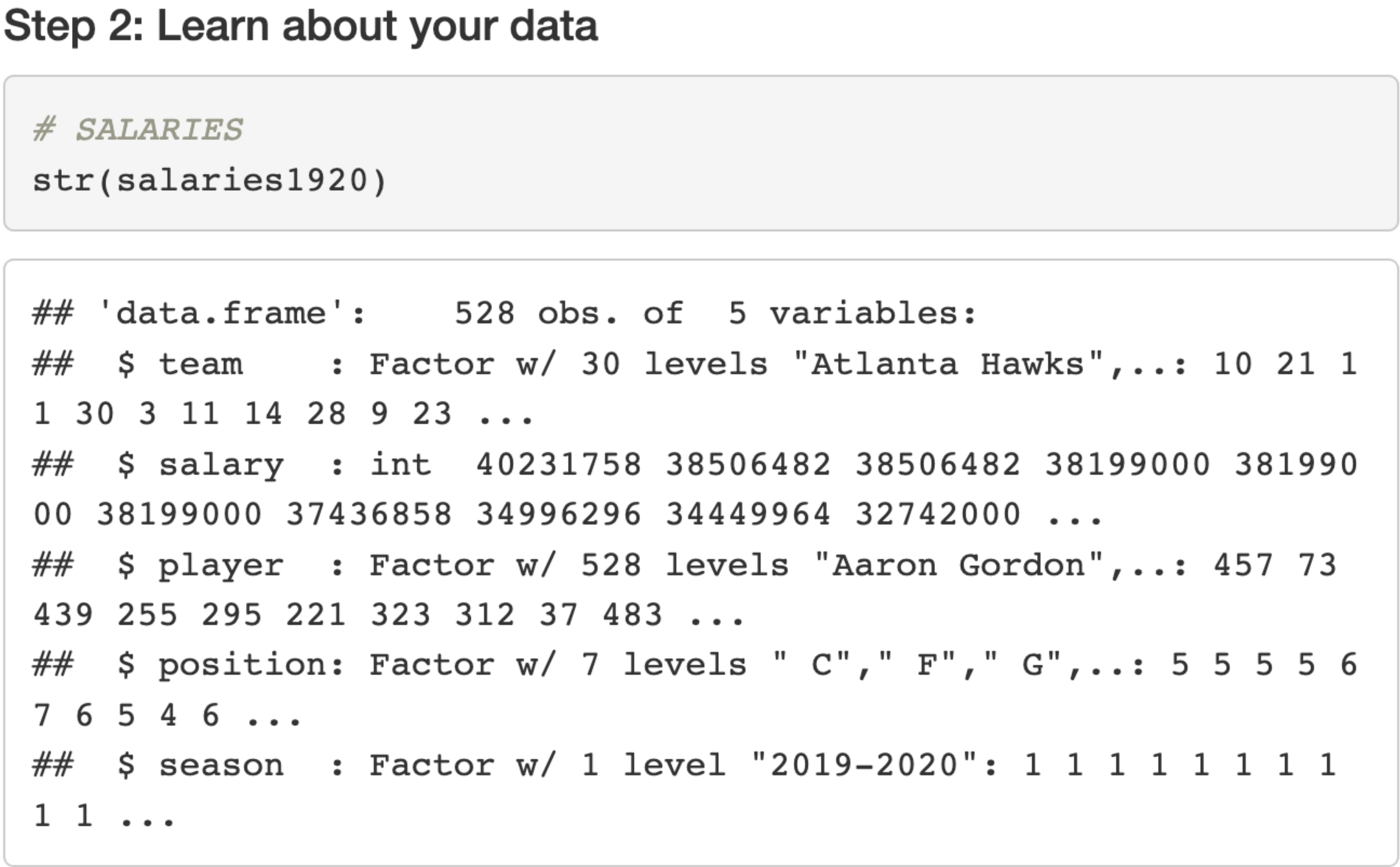


Step 1: Load Data

```r
## SALARY DATA for 2019-2020 season
salaries1920 <- read.csv("~/Downloads/nba2019-20.csv")

## METRICS ON PLAYER PERFORMANCE
## 1996 to 2022
all_seasons <- read.csv("~/Downloads/all_seasons.csv")
```

STEP 2: LOOK AT THE DATA STRUCTURE

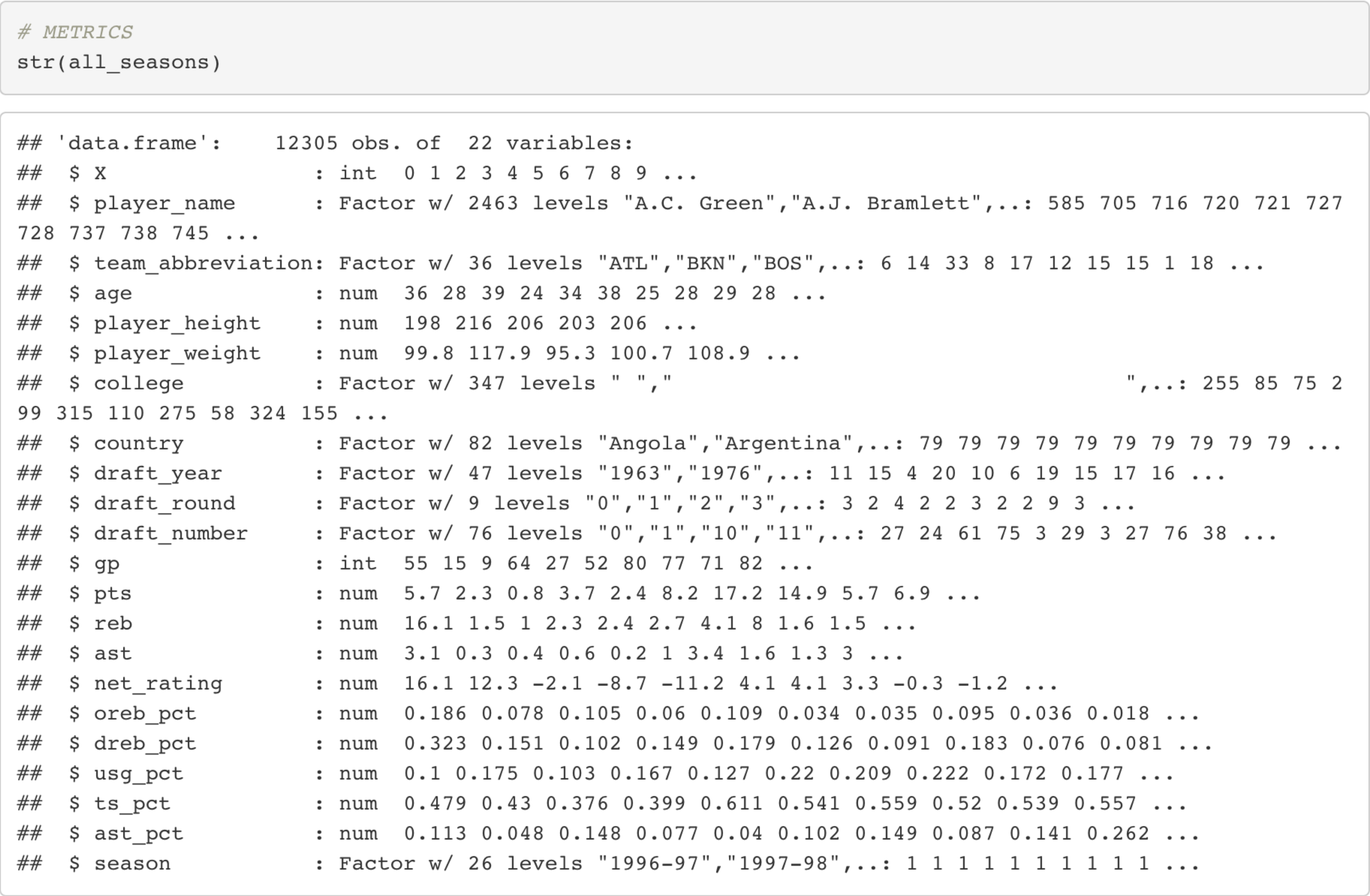


Step 2: Learn about your data

```
# SALARIES
str(salaries1920)
```

```
## 'data.frame':    528 obs. of  5 variables:
##  $ team    : Factor w/ 30 levels "Atlanta Hawks",..: 10 21 1
1 30 3 11 14 28 9 23 ...
##  $ salary  : int  40231758 38506482 38506482 38199000 381990
00 38199000 37436858 34996296 34449964 32742000 ...
##  $ player  : Factor w/ 528 levels "Aaron Gordon",..: 457 73
439 255 295 221 323 312 37 483 ...
##  $ position: Factor w/ 7 levels " C"," F"," G",..: 5 5 5 5 6
7 6 5 4 6 ...
##  $ season  : Factor w/ 1 level "2019-2020": 1 1 1 1 1 1 1 1
1 1 ...
```

STEP 2: LOOK AT THE DATA STRUCTURE

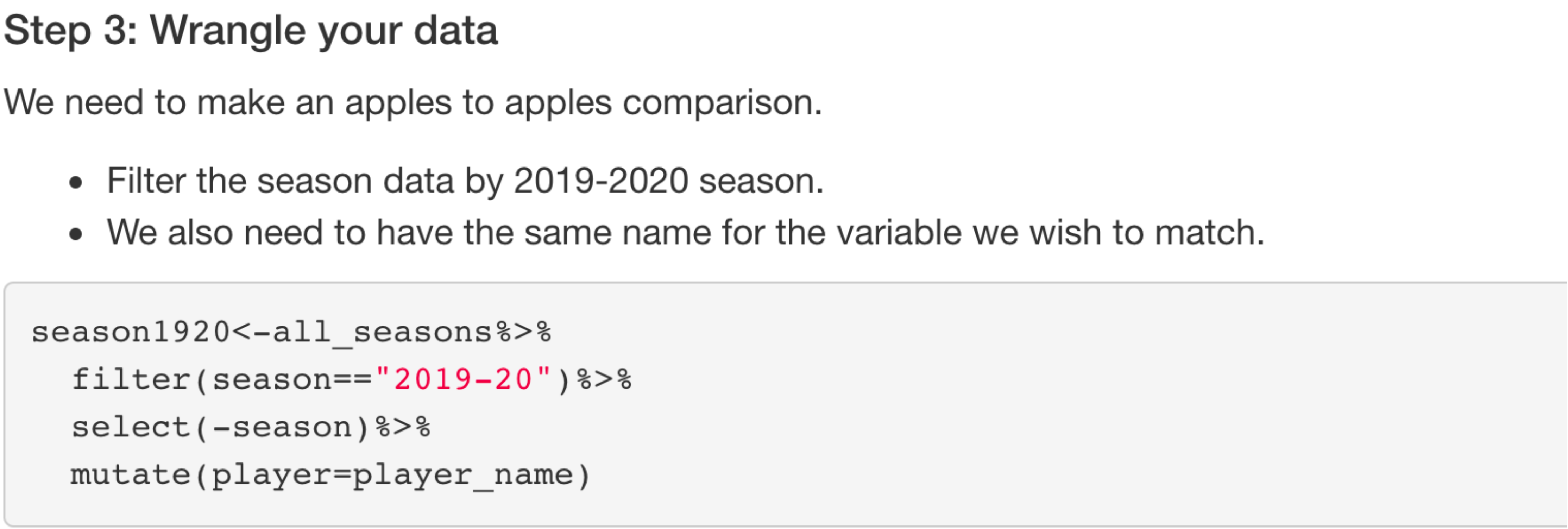


```
# METRICS
str(all_seasons)
```

```
## 'data.frame':    12305 obs. of  22 variables:
##  $ X                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ player_name      : Factor w/ 2463 levels "A.C. Green","A.J. Bramlett",..: 585 705 716 720 721 727
728 737 738 745 ...
##  $ team_abbreviation: Factor w/ 36 levels "ATL","BKN","BOS",..: 6 14 33 8 17 12 15 15 1 18 ...
##  $ age              : num  36 28 39 24 34 38 25 28 29 28 ...
##  $ player_height    : num  198 216 206 203 206 ...
##  $ player_weight    : num  99.8 117.9 95.3 100.7 108.9 ...
##  $ college          : Factor w/ 347 levels " ","                 ",..: 255 85 75 2
99 315 110 275 58 324 155 ...
##  $ country          : Factor w/ 82 levels "Angola","Argentina",..: 79 79 79 79 79 79 79 79 79 79 ...
##  $ draft_year       : Factor w/ 47 levels "1963","1976",..: 11 15 4 20 10 6 19 15 17 16 ...
##  $ draft_round      : Factor w/ 9 levels "0","1","2","3",..: 3 2 4 2 2 3 2 2 9 3 ...
##  $ draft_number     : Factor w/ 76 levels "0","1","10","11",..: 27 24 61 75 3 29 3 27 76 38 ...
##  $ gp               : int  55 15 9 64 27 52 80 77 71 82 ...
##  $ pts              : num  5.7 2.3 0.8 3.7 2.4 8.2 17.2 14.9 5.7 6.9 ...
##  $ reb              : num  16.1 1.5 1 2.3 2.4 2.7 4.1 8 1.6 1.5 ...
##  $ ast              : num  3.1 0.3 0.4 0.6 0.2 1 3.4 1.6 1.3 3 ...
##  $ net_rating       : num  16.1 12.3 -2.1 -8.7 -11.2 4.1 4.1 3.3 -0.3 -1.2 ...
##  $ oreb_pct         : num  0.186 0.078 0.105 0.06 0.109 0.034 0.035 0.095 0.036 0.018 ...
##  $ dreb_pct         : num  0.323 0.151 0.102 0.149 0.179 0.126 0.091 0.183 0.076 0.081 ...
##  $ usg_pct          : num  0.1 0.175 0.103 0.167 0.127 0.22 0.209 0.222 0.172 0.177 ...
##  $ ts_pct           : num  0.479 0.43 0.376 0.399 0.611 0.541 0.559 0.52 0.539 0.557 ...
##  $ ast_pct          : num  0.113 0.048 0.148 0.077 0.04 0.102 0.149 0.087 0.141 0.262 ...
##  $ season           : Factor w/ 26 levels "1996-97","1997-98",..: 1 1 1 1 1 1 1 1 1 1 ...
```

THESE DATA SETS ARE APPLES AND ORANGES!

STEP 3: WRANGLE YOUR DATA



Step 3: Wrangle your data

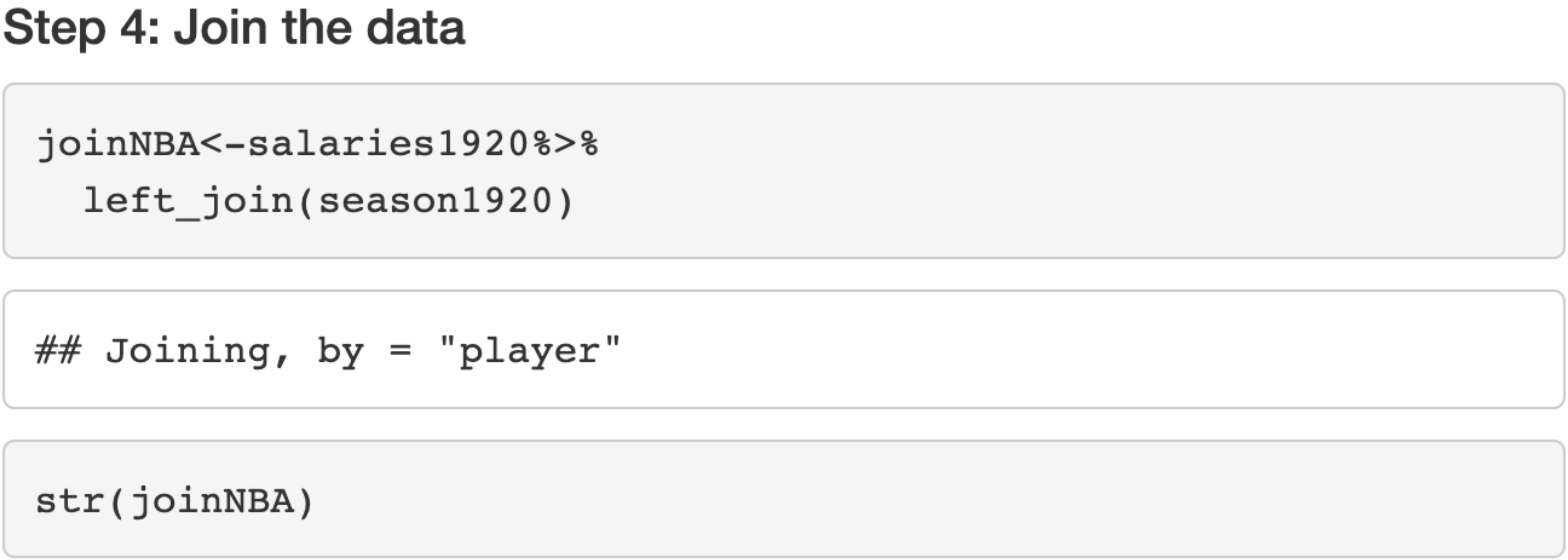
We need to make an apples to apples comparison.

- Filter the season data by 2019-2020 season.
- We also need to have the same name for the variable we wish to match.

```
season1920<-all_seasons%>%
  filter(season=="2019-20")%>%
  select(-season)%>%
  mutate(player=player_name)
```

NOW WE HAVE APPLES TO APPLES

STEP 4: JOIN THE DATA



Step 4: Join the data

```
joinNBA<-salaries1920%>%
  left_join(season1920)
```

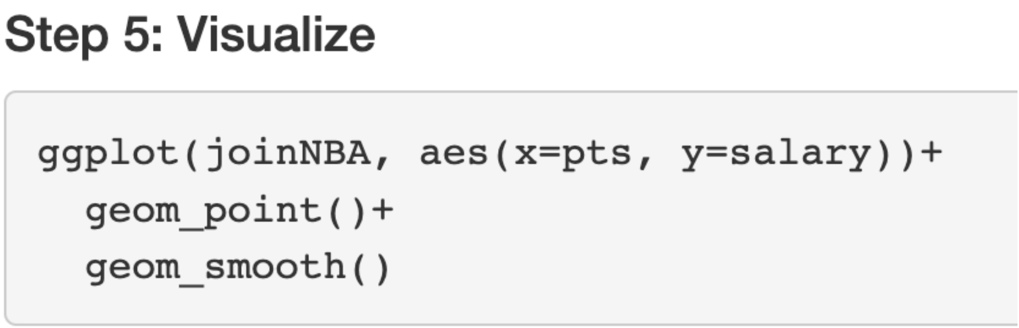
```
## Joining, by = "player"
```

```
str(joinNBA)
```

MOTIVATING QUESTION

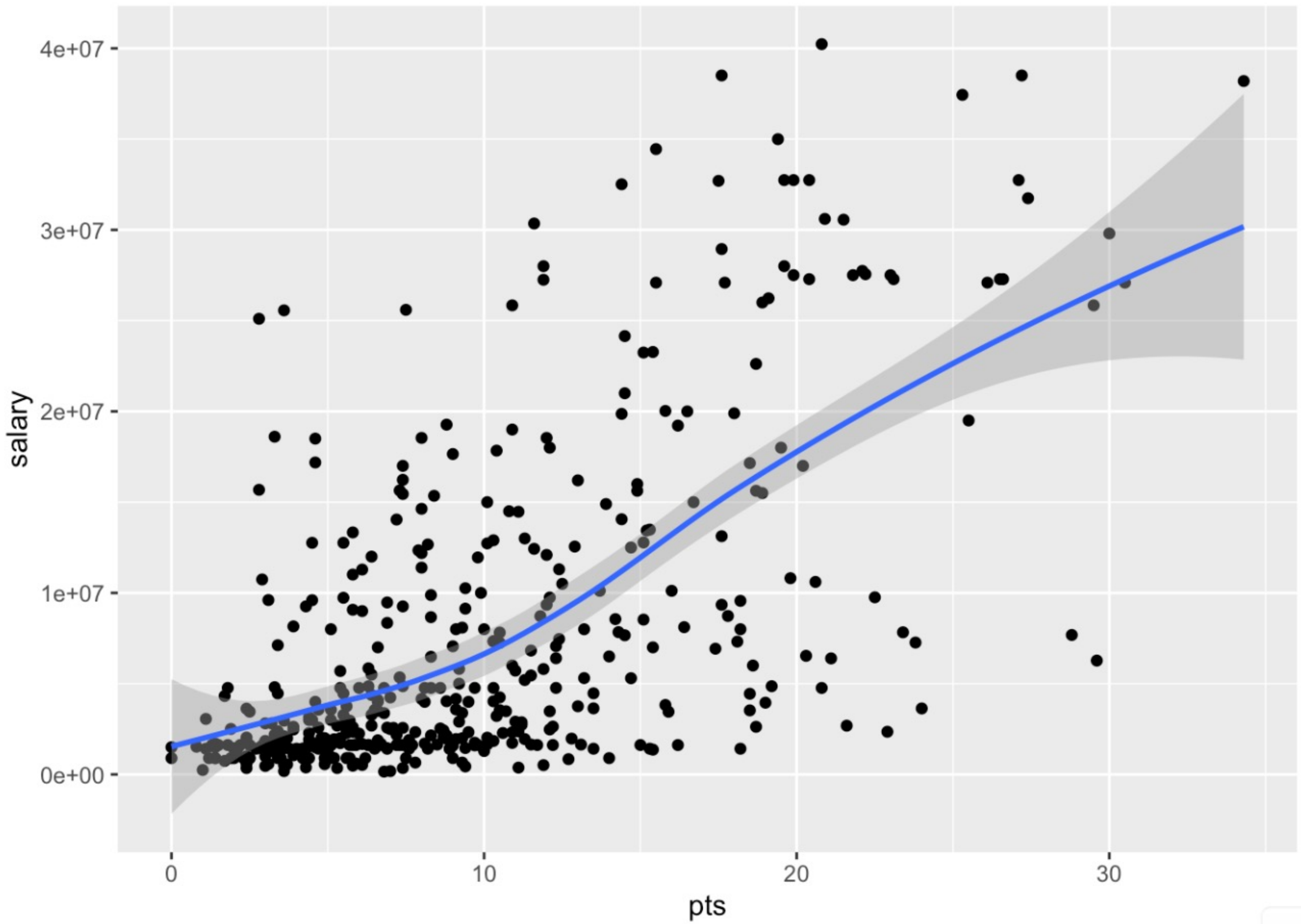
Is player salary related to player performance?

STEP 5:VISUALIZE



Step 5: Visualize

```
ggplot(joinNBA, aes(x=pts, y=salary))+
  geom_point()+
  geom_smooth()
```


4e+07
3e+07
2e+07
1e+07
0e+00
salary
pts