---

title: “MD_CH3.4_HW” author: “Rachel Kaufman” date: “2022-09-06” output: pdf_document

Chapter 3 & 4 Homework Assignment

CHAPTER 3 ASSIGNMENT QUESTIONS

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.0     v forcats 0.5.2
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
mario_kart <- read.csv("Data/world_records.csv")
glimpse(mario_kart)
```

```
## Rows: 2,334
## Columns: 9
## $ track           <chr> "Luigi Raceway", "Luigi Raceway", "Luigi Raceway", "Lu~
## $ type            <chr> "Three Lap", "Three Lap", "Three Lap", "Three Lap", "T~
## $ shortcut        <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", ~
## $ player          <chr> "Salam", "Booth", "Salam", "Salam", "Gregg G", "Rocky ~
## $ system_played   <chr> "NTSC", "NTSC", "NTSC", "NTSC", "NTSC", "NTSC", "NTSC"~
## $ date            <chr> "1997-02-15", "1997-02-16", "1997-02-16", "1997-02-28"~
## $ time_period     <chr> "2M 12.99S", "2M 9.99S", "2M 8.99S", "2M 6.99S", "2M 4~
## $ time            <dbl> 132.99, 129.99, 128.99, 126.99, 124.51, 122.89, 122.87~
## $ record_duration <int> 1, 0, 12, 7, 54, 0, 0, 27, 0, 64, 3, 0, 90, 132, 1, 74~
```

Question #1:

```
three_laps <- mario_kart %>% filter(type == "Three Lap")
glimpse(three_laps)
```

```
## Rows: 1,211
## Columns: 9
## $ track           <chr> "Luigi Raceway", "Luigi Raceway", "Luigi Raceway", "Lu~
## $ type            <chr> "Three Lap", "Three Lap", "Three Lap", "Three Lap", "T~
## $ shortcut        <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", ~
## $ player          <chr> "Salam", "Booth", "Salam", "Salam", "Gregg G", "Rocky ~
## $ system_played   <chr> "NTSC", "NTSC", "NTSC", "NTSC", "NTSC", "NTSC", "NTSC"~
## $ date            <chr> "1997-02-15", "1997-02-16", "1997-02-16", "1997-02-28"~
## $ time_period     <chr> "2M 12.99S", "2M 9.99S", "2M 8.99S", "2M 6.99S", "2M 4~
## $ time            <dbl> 132.99, 129.99, 128.99, 126.99, 124.51, 122.89, 122.87~
## $ record_duration <int> 1, 0, 12, 7, 54, 0, 0, 27, 0, 64, 3, 0, 90, 132, 1, 74~
```

```
##okay, so now I need to remove Rainbow Road (this course is a bitch) and its times
##first step is to make a new data.frame, removing the rainbow road times
salt_worthy_tracks <- three_laps %>%
  filter(!(track %in% c("Rainbow Road")))


##then creating the rainbow road only data.frame from the three_laps data frame
Rainbow_road_data <- three_laps %>%
  filter(track == "Rainbow Road")
glimpse(Rainbow_road_data)
```

```
## Rows: 99
## Columns: 9
## $ track           <chr> "Rainbow Road", "Rainbow Road", "Rainbow Road", "Rainb~
## $ type            <chr> "Three Lap", "Three Lap", "Three Lap", "Three Lap", "T~
## $ shortcut        <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", ~
## $ player          <chr> "Booth", "Jonathan", "Zwartjes", "Jonathan", "Penev", ~
## $ system_played   <chr> "NTSC", "NTSC", "PAL", "NTSC", "PAL", "PAL", "PAL", "P~
## $ date            <chr> "1997-05-27", "1997-08-27", "1998-01-14", "1998-03-13"~
## $ time_period     <chr> "6M 15.83S", "6M 9.67S", "6M 8.69S", "6M 5.51S", "6M 4~
## $ time            <dbl> 375.83, 369.67, 368.69, 365.51, 364.15, 363.86, 362.15~
## $ record_duration <int> 92, 140, 58, 173, 9, 2, 9, 8, 9, 1, 14, 113, 65, 8, 35~
```

Question #2:

```
## Find the average track times for Rainbow Road and the SD of the records which would be the # of rows
Rainbow_road_data %>%
  summarize(
    mean_record_time = mean(time),
    sd_record_time = sd(time))
```

```
##   mean_record_time sd_record_time
## 1         275.6336       91.81962
```

```r
##Now finding the average and SD for the tracks EXLCUDING rainbow road
salt_worthy_tracks %>%
  summarize(
    mean_record_time = mean(time), #don't forget the comma!
    std_record_time = sd(time))
```

```
##   mean_record_time std_record_time
## 1         113.7984        52.97595
```

For rainbow road, the mean time is 275 seconds with an SD of 91 for the variation in times. The mean time for all the other tracks is 113 seconds with an SD of 52 for record duration. Rainbow road has a quite longer average time for three track races in comparison to all of the other tracks. this is not very surprising considering I have never made it through Rainbow Road without falling off at least once. As for the Standard Deviation, they are fairly close (granted you would need to standardize the SD to come to any definitive conclusions) comparing rainbow road to all other courses. They both have pretty high (varying may be a better word here) SD for the duration in which the record is held. I wonder if that could be related to players frequency in choosing to play more fun tracks (obviously Moo Moo farm for me) so the turn over rate for the duration of record holding might me quicker!

Question #3:

```r
three_laps_records_count <- three_laps %>%
  group_by(track) %>%
  summarize(three_laps_records_count = n()
          ) %>%
  arrange(desc(three_laps_records_count)) #the arrange has to be on outside of summarize function, henc
```

Toad's Turnpike has the most records established at 124. Toad's turnpike is obviously the easier of the races....

Question #4:

```r
nerds_with_records <- three_laps %>%
  group_by(track, player) %>%
  summarize(count = n()) #warning that summarise is grouped output by player
```

```
## `summarise()` has grouped output by 'track'. You can override using the
## `.groups` argument.
```

```r
nerds_with_records %>%
  arrange(desc(count))
```

```
## # A tibble: 306 x 3
## # Groups:   track [16]
##    track                player   count
##    <chr>                <chr>    <int>
##  1 Choco Mountain       Penev       26
##  2 D.K.'s Jungle Parkway Lacey      24
##  3 Rainbow Road         abney317    21
##  4 Toad's Turnpike      MR          20
##  5 Frappe Snowland      MR          18
##  6 Toad's Turnpike      Penev       18
```

```
##  7 Kalimari Desert       abney317    16
##  8 Sherbet Land          MR          16
##  9 Banshee Boardwalk     MR          15
## 10 Choco Mountain        abney317    15
## # ... with 296 more rows
```

Cool so the biggest nerd is Penev with 26 records at Choco Mountain

Question 5:

```
three_laps %>%
  group_by(track) %>%
  arrange(time) %>%
  slice(1) %>%
  select(track, time)
```

```
## # A tibble: 16 x 2
## # Groups:   track [16]
##    track                  time
##    <chr>                 <dbl>
##  1 Banshee Boardwalk     124.
##  2 Bowser's Castle       132
##  3 Choco Mountain         17.3
##  4 D.K.'s Jungle Parkway  21.4
##  5 Frappe Snowland        23.6
##  6 Kalimari Desert       122.
##  7 Koopa Troopa Beach     95.2
##  8 Luigi Raceway          25.3
##  9 Mario Raceway          58.5
## 10 Moo Moo Farm           85.9
## 11 Rainbow Road           50.4
## 12 Royal Raceway         119.
## 13 Sherbet Land           91.6
## 14 Toad's Turnpike        30.3
## 15 Wario Stadium          14.6
## 16 Yoshi Valley           33.4
```

Wario stadium has the fastest time at 14.59 seconds which is very confusing and frankly sounds fake because three laps???? for that short of time??? Fake news.

Question 6:

```
duration_of_records <- three_laps %>%
  mutate(
    longterm_records = as.numeric(three_laps$record_duration >= 100))

##part 2 of Question 6
duration_of_records %>%
  group_by(player) %>%
  summarize(sum_of_records = sum(longterm_records)) %>%
    ##if this was group by player, ... confused... FIX*****
  arrange(desc(sum_of_records))
```

```
## # A tibble: 60 x 2
```

```
##    player   sum_of_records
##    <chr>             <dbl>
##  1 MR                   81
##  2 MJ                   50
##  3 Penev                27
##  4 abney317             26
##  5 VAJ                  26
##  6 Zwartjes             25
##  7 Lacey                23
##  8 Dan                  21
##  9 Karlo                18
## 10 Booth                17
## # ... with 50 more rows
```

Player name "MR" has the holds the most records (n= 81) that have a duration of 100 days or more. I wonder what MR does for a living.

Question 7

```
drivers <- read_csv("Data/drivers.csv")
```

```
## Rows: 2250 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): player, nation
## dbl (4): position, total, year, records
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
drivers_joined <- three_laps %>%
  left_join(drivers, by = "player")
```

CHAPTER 4 ASSIGNMENTS!

Question #1:

```
Ew_Football <- read_csv("https://raw.githubusercontent.com/NicolasRestrep/223_course/main/Data/nfl_sala
```

```
## Rows: 800 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (11): year, Cornerback, Defensive Lineman, Linebacker, Offensive Lineman...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Question #2:

```
glimpse(Ew_Football)
```

```
## Rows: 800
## Columns: 11
## $ year                <dbl> 2011, 2011, 2011, 2011, 2011, 2011, 2011, 2011, 20~
## $ Cornerback          <dbl> 11265916, 11000000, 10000000, 10000000, 10000000, ~
## $ `Defensive Lineman` <dbl> 17818000, 16200000, 12476000, 11904706, 11762782, ~
## $ Linebacker          <dbl> 16420000, 15623000, 11825000, 10083333, 10020000, ~
## $ `Offensive Lineman` <dbl> 15960000, 12800000, 11767500, 10358200, 10000000, ~
## $ Quarterback         <dbl> 17228125, 16000000, 14400000, 14100000, 13510000, ~
## $ `Running Back`      <dbl> 12955000, 10873833, 9479000, 7700000, 7500000, 703~
## $ Safety              <dbl> 8871428, 8787500, 8282500, 8000000, 7804333, 76527~
## $ `Special Teamer`    <dbl> 4300000, 3725000, 3556176, 3500000, 3250000, 32250~
## $ `Tight End`         <dbl> 8734375, 8591000, 8290000, 7723333, 6974666, 61333~
## $ `Wide Receiver`     <dbl> 16250000, 14175000, 11424000, 11415000, 10800000, ~
```

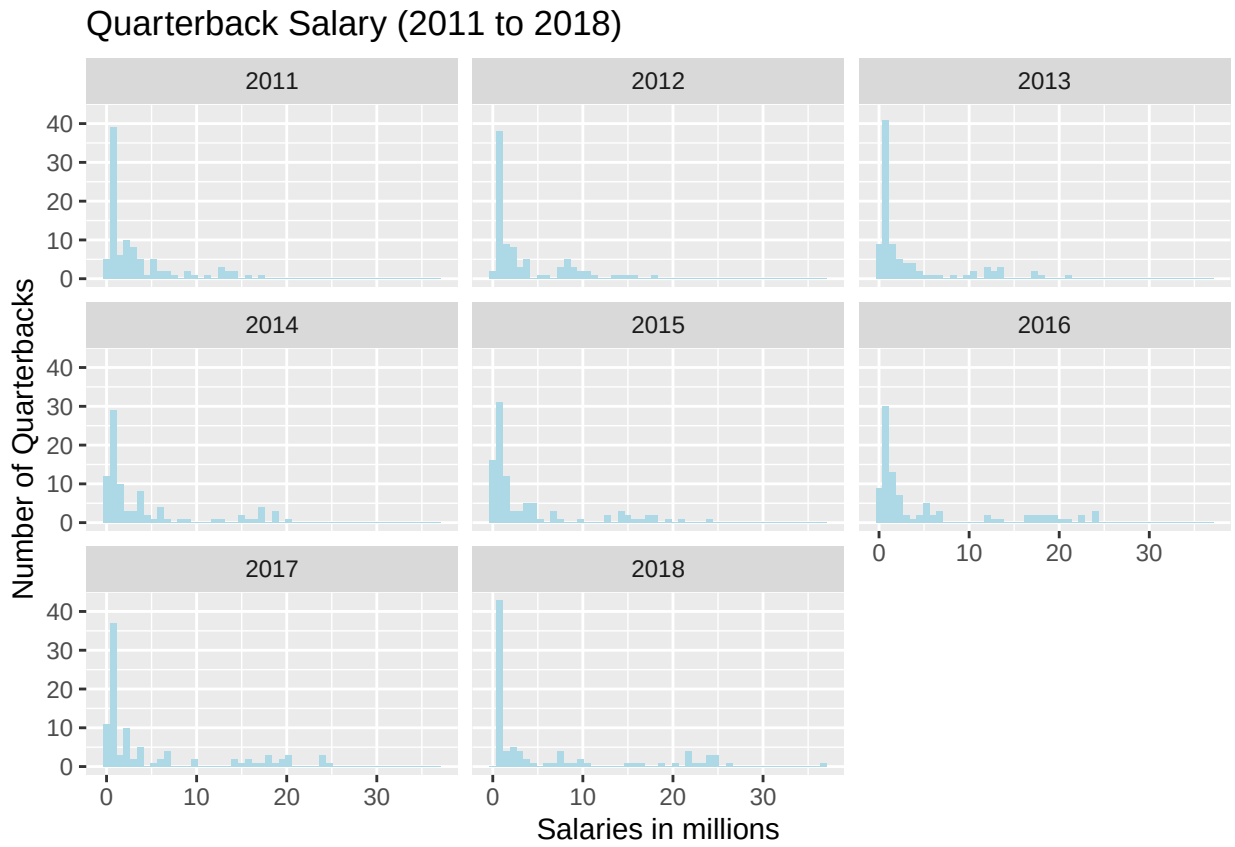
```
Ew_football_tidy <- Ew_Football %>%
  pivot_longer(
    names_to = "Positions",
    values_to = "salaries",
      cols = -year
  )
```

Question #3: Of course, there are many folks in each position and their salaries vary widely. Let's look at quarterbacks for example. Filter your newly created dataset so that it only contains quarterbacks. Then, make a histogram where salary is in the x-axis. Then use facet_wrap to get the histogram for each year. What patterns do you notice?

```
quaterback_salary <- Ew_football_tidy %>%
  filter(Positions == "Quarterback") %>%
  group_by(year) %>%
  drop_na()
glimpse(quaterback_salary)
```

```
## Rows: 745
## Columns: 3
## Groups: year [8]
## $ year      <dbl> 2011, 2011, 2011, 2011, 2011, 2011, 2011, 2011, 2011, 2011, ~
## $ Positions <chr> "Quarterback", "Quarterback", "Quarterback", "Quarterback", ~
## $ salaries  <dbl> 17228125, 16000000, 14400000, 14100000, 13510000, 13250000, ~
```

```
ggplot(quaterback_salary, aes(x = salaries)) +
  geom_histogram(fill = "light blue", binwidth = 750000) +
  labs(title = "Quarterback Salary (2011 to 2018)",
       x = "Salaries in millions",
       y = "Number of Quarterbacks") +
  scale_x_continuous(labels = scales::label_number(suffix = "", scale = 1e-6)) +
  facet_wrap(~year)
```



In terms of patterns, it looks like over time salaries are becoming more dispersed and the averagees are being pulled in the later years for people who are getting paid in the upwards of 30 million. **Question #4:** Let's calculate the average salary for each position, each year. Create a new dataset that contains the average salary for each position each year. To do this, you will need the group_by and summarize combo.

```
avg_absurd_salary <- Ew_football_tidy %>%
  group_by(year, Positions) %>%
  summarize("average salary" = mean(salaries)) %>%
  drop_na()
```

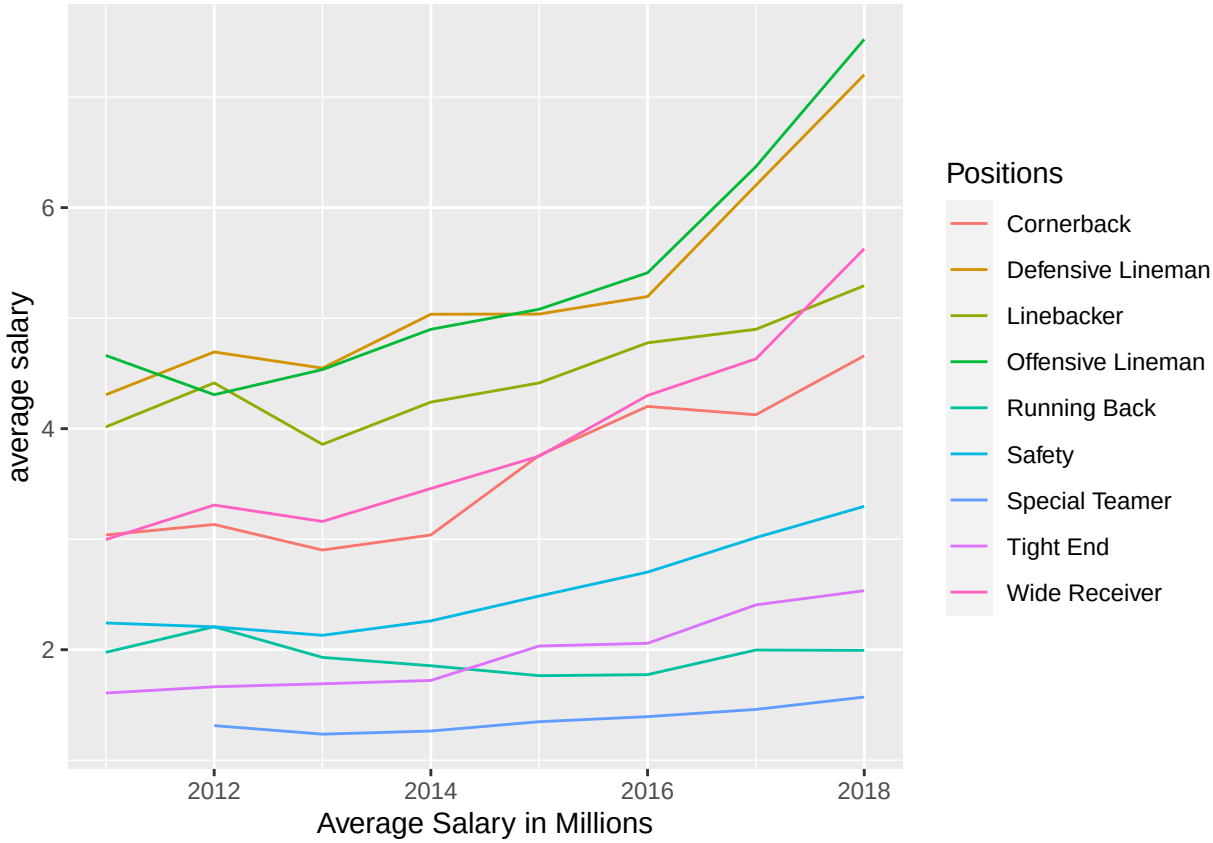
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

Question #5: Make a linegraph that traces the evolution of each position's average salary across the years. You can use use different strategies to distinguish between positions - color or facets for example. What is important is that you see each position's trajectory clearly and that they are comparable.

Describe at least two trends that are apparent to you.

```
##that means my line should represent positions and then my salary
ggplot(data = avg_absurd_salary, mapping = aes(x = year, y = `average salary`, color = Positions)) +
  scale_y_continuous(labels = scales::label_number(suffix = "", scale = 1e-6)) +
  labs(x = "Average Salary in Millions") +
geom_line(se = FALSE)
```

```
## Warning: Ignoring unknown parameters: se
```



In terms of two trends apparent to me, there is a more prominent slope for Offensive Lineman and Defensive Linemen that starts at 2016, even though overall every position seems to be at least relatively trending upwards in terms of salary. Sort of makes sense as they are some dangerous positions (but arent they all.... anyways) That is, unless you're a kicker. My second observation would be that "Special Teamer" get too much pressure for making less than all the other players. There seems to be a lack of general uptrend for this positions and for running backs. I feel like running backs is on the lower end for average salaries likely because teams rotate a lot of these dudes, and the guys who don't get the spotlight probably drag down thee average. But hey, I am guessing here. No idea how payment structure works.