Chapter 3 and 4 - Modern Dive

Allyson Cameron

2022-09-06

Chapter 3

First, lets load `tidyverse` and load the data.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library(dplyr)
library(knitr)
mario_kart <- read_csv(paste0("/Users/allysoncameron/Documents/soc_722_stats/",
                              "Data/world_records.csv"))
```

```
## Rows: 2334 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): track, type, shortcut, player, system_played, time_period
## dbl  (2): time, record_duration
## date (1): date
##
## 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.
```

```
View(mario_kart)
```

Question 1

Now let's filter out only the races with `"Three Lap"` and take out laps from `"Rainbow Road"`.

```
three_laps <- mario_kart %>%
  filter(type == "Three Lap" & track != "Rainbow Road")

three_laps
```

```
## # A tibble: 1,112 x 9
##    track         type    short~1 player syste~2 date       time_~3  time recor~4
##    <chr>         <chr>   <chr>   <chr>  <chr>   <date>     <chr>   <dbl>   <dbl>
##  1 Luigi Raceway Three ~ No      Salam  NTSC    1997-02-15 2M 12.~  133.       1
##  2 Luigi Raceway Three ~ No      Booth  NTSC    1997-02-16 2M 9.9~  130.       0
##  3 Luigi Raceway Three ~ No      Salam  NTSC    1997-02-16 2M 8.9~  129.      12
##  4 Luigi Raceway Three ~ No      Salam  NTSC    1997-02-28 2M 6.9~  127.       7
##  5 Luigi Raceway Three ~ No      Gregg~ NTSC    1997-03-07 2M 4.5~  125.      54
##  6 Luigi Raceway Three ~ No      Rocky~ NTSC    1997-04-30 2M 2.8~  123.       0
##  7 Luigi Raceway Three ~ No      Launs~ NTSC    1997-04-30 2M 2.8~  123.       0
##  8 Luigi Raceway Three ~ No      Launs~ NTSC    1997-04-30 2M 2.7~  123.      27
##  9 Luigi Raceway Three ~ No      Launs~ NTSC    1997-05-27 2M 2.2~  122.       0
## 10 Luigi Raceway Three ~ No      Launs~ NTSC    1997-05-27 2M 2.2~  122.      64
## # ... with 1,102 more rows, and abbreviated variable names 1: shortcut,
## #   2: system_played, 3: time_period, 4: record_duration
## # i Use `print(n = ...)` to see more rows
```

Now, let's save a dataset that only contains the records achieved at Rainbow Road.

```
rainbow_road <- mario_kart %>%
  filter(type == "Three Lap" & track == "Rainbow Road")

rainbow_road
```

```
## # A tibble: 99 x 9
##    track        type     short~1 player syste~2 date       time_~3  time recor~4
##    <chr>        <chr>    <chr>   <chr>  <chr>   <date>     <chr>   <dbl>   <dbl>
##  1 Rainbow Road Three L~ No      Booth  NTSC    1997-05-27 6M 15.~  376.      92
##  2 Rainbow Road Three L~ No      Jonat~ NTSC    1997-08-27 6M 9.6~  370.     140
##  3 Rainbow Road Three L~ No      Zwart~ PAL     1998-01-14 6M 8.6~  369.      58
##  4 Rainbow Road Three L~ No      Jonat~ NTSC    1998-03-13 6M 5.5~  366.     173
##  5 Rainbow Road Three L~ No      Penev  PAL     1998-09-02 6M 4.1~  364.       9
##  6 Rainbow Road Three L~ No      Penev  PAL     1998-09-11 6M 3.8~  364.       2
##  7 Rainbow Road Three L~ No      Penev  PAL     1998-09-13 6M 2.1~  362.       9
##  8 Rainbow Road Three L~ No      Penev  PAL     1998-09-22 6M 1.9~  362.       8
##  9 Rainbow Road Three L~ No      Penev  PAL     1998-09-30 6M 1.7~  362.       9
## 10 Rainbow Road Three L~ No      Penev  PAL     1998-10-09 6M 1.6~  362.       1
## # ... with 89 more rows, and abbreviated variable names 1: shortcut,
## #   2: system_played, 3: time_period, 4: record_duration
## # i Use `print(n = ...)` to see more rows
```

Question 2

Now, lets get the average `time` at Rainbow Road and the standard deviations.

```
summary_rr <- rainbow_road %>%

  summarize(mean_time = mean(time),
            sd_record_rr = sd(time))

summary_rr
```

```
## # A tibble: 1 x 2
##   mean_time sd_record_rr
##       <dbl>        <dbl>
## 1      276.         91.8
```

Let's do the same things for the other dataset with all of the other tracks.

```
summary_three_laps <- three_laps %>%
  summarize(mean_time_3 = mean(time),
            sd_record__3 = sd(time))

summary_three_laps
```

```
## # A tibble: 1 x 2
##   mean_time_3 sd_record__3
##         <dbl>        <dbl>
## 1        114.         53.0
```

Notice any differences? The average time for Rainbow Road was significantly longer (275.63) than the average for all other tracks doing three-laps (113.80). Additionally, there is more variation in the times of the records at Rainbow Road (91.82) than at the other tracks with three-laps (52.98).

Question 3

Next we are going to create `three_laps_by_track` which will first look in `three_laps`, then (%>%), `group_by` tracks, then (%>%), `filter` to only count cases of individuals who actually currently hold a record, then (%>%), `summarize` to count how many different records have been established on each track. After this, I will arrange the counts in descending order so that I can see which track has the most records.

```
three_laps_by_track <- three_laps %>%
  group_by(track) %>%
  filter(record_duration != 0) %>%
  summarize(num_three_laps_records = n()) %>%
  arrange(desc(num_three_laps_records))

three_laps_by_track
```

```
## # A tibble: 15 x 2
##    track                num_three_laps_records
##    <chr>                                 <int>
##  1 Toad's Turnpike                          86
##  2 Frappe Snowland                          82
```

```
##  3 D.K.'s Jungle Parkway                    80
##  4 Mario Raceway                            80
##  5 Choco Mountain                           77
##  6 Kalimari Desert                          70
##  7 Royal Raceway                            70
##  8 Yoshi Valley                             70
##  9 Luigi Raceway                            65
## 10 Wario Stadium                            64
## 11 Sherbet Land                             55
## 12 Banshee Boardwalk                        53
## 13 Koopa Troopa Beach                       50
## 14 Moo Moo Farm                             42
## 15 Bowser's Castle                          39
```

Toad's Turnpike has the most, with 86 current records.

Question 4

Now we want to investigate if there are drivers who have multiple records at each track, and how many records they have.

For this, we will be grouping by both driver and track.

```
by_player_each_track <- three_laps %>%
  group_by(player, track) %>%
  filter(record_duration != 0) %>%
  summarize(num_by_player_track = n()) %>%
  arrange(desc(num_by_player_track))
```

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

```
by_player_each_track
```

```
## # A tibble: 277 x 3
## # Groups:   player [52]
##    player   track                num_by_player_track
##    <chr>    <chr>                              <int>
##  1 Penev    Choco Mountain                        24
##  2 Lacey    D.K.'s Jungle Parkway                 23
##  3 MR       Frappe Snowland                       17
##  4 abney317 Kalimari Desert                       16
##  5 MR       Toad's Turnpike                       16
##  6 abney317 Choco Mountain                        15
##  7 Penev    Toad's Turnpike                       14
##  8 MR       Banshee Boardwalk                     13
##  9 Penev    Frappe Snowland                       13
## 10 Penev    Royal Raceway                         13
## # ... with 267 more rows
## # i Use `print(n = ...)` to see more rows
```

Who is the player that has recorded the most records at any one track and what track was it? Player Penev is the player who holds the most records overall (24) and this is on a track called Choco Mountain.

Question 5

Now, I will show you the best time recorded on each track by using `group_by`, `arrange`, and `slice` to see the first (best) time for each. I will also limit which columns show using `select`.

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

best_time_by_track
```

```
## # A tibble: 15 x 2
## # Groups:   track [15]
##    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 Royal Raceway         119.
## 12 Sherbet Land           91.6
## 13 Toad's Turnpike        30.3
## 14 Wario Stadium          14.6
## 15 Yoshi Valley           33.4
```

Question 6

Let's create a new variable that is a 1 if record_duration is higher than 100 or 0 otherwise.

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

three_laps
```

```
## # A tibble: 1,112 x 10
##    track   type  short~1 player syste~2 date       time_~3  time recor~4 rec_d~5
##    <chr>   <chr> <chr>   <chr>  <chr>   <date>     <chr>   <dbl>   <dbl>   <dbl>
```

```
##  1 Luigi ~ Thre~ No      Salam  NTSC    1997-02-15 2M 12.~  133.       1       0
##  2 Luigi ~ Thre~ No      Booth  NTSC    1997-02-16 2M 9.9~  130.       0       0
##  3 Luigi ~ Thre~ No      Salam  NTSC    1997-02-16 2M 8.9~  129.      12       0
##  4 Luigi ~ Thre~ No      Salam  NTSC    1997-02-28 2M 6.9~  127.       7       0
##  5 Luigi ~ Thre~ No      Gregg~ NTSC    1997-03-07 2M 4.5~  125.      54       0
##  6 Luigi ~ Thre~ No      Rocky~ NTSC    1997-04-30 2M 2.8~  123.       0       0
##  7 Luigi ~ Thre~ No      Launs~ NTSC    1997-04-30 2M 2.8~  123.       0       0
##  8 Luigi ~ Thre~ No      Launs~ NTSC    1997-04-30 2M 2.7~  123.      27       0
##  9 Luigi ~ Thre~ No      Launs~ NTSC    1997-05-27 2M 2.2~  122.       0       0
## 10 Luigi ~ Thre~ No      Launs~ NTSC    1997-05-27 2M 2.2~  122.      64       0
## # ... with 1,102 more rows, and abbreviated variable names 1: shortcut,
## #   2: system_played, 3: time_period, 4: record_duration, 5: rec_duration_mod
## # i Use `print(n = ...)` to see more rows
```

Now, let's look at the total amount of long duration records each player holds.

```
long_duration_by_player <- three_laps %>%
  group_by(player) %>%
  summarize(sum_rec_duration = sum(rec_duration_mod)) %>%
  arrange(desc(sum_rec_duration))

long_duration_by_player
```

```
## # A tibble: 57 x 2
##    player   sum_rec_duration
##    <chr>               <dbl>
##  1 MR                     76
##  2 MJ                     47
##  3 Penev                  24
##  4 Zwartjes               24
##  5 Lacey                  23
##  6 VAJ                    23
##  7 abney317               21
##  8 Dan                    20
##  9 Booth                  16
## 10 Karlo                  16
## # ... with 47 more rows
## # i Use `print(n = ...)` to see more rows
```

What player has the most long duration records? Player MR has the most long-duration records (76).

Question 7

Now, lets import the a data set to join it with our `three_laps` dataset.

```
drivers <- read_csv(paste0("/Users/allysoncameron/Documents/soc_722_stats/",
                           "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.
```

```
View(drivers)
```

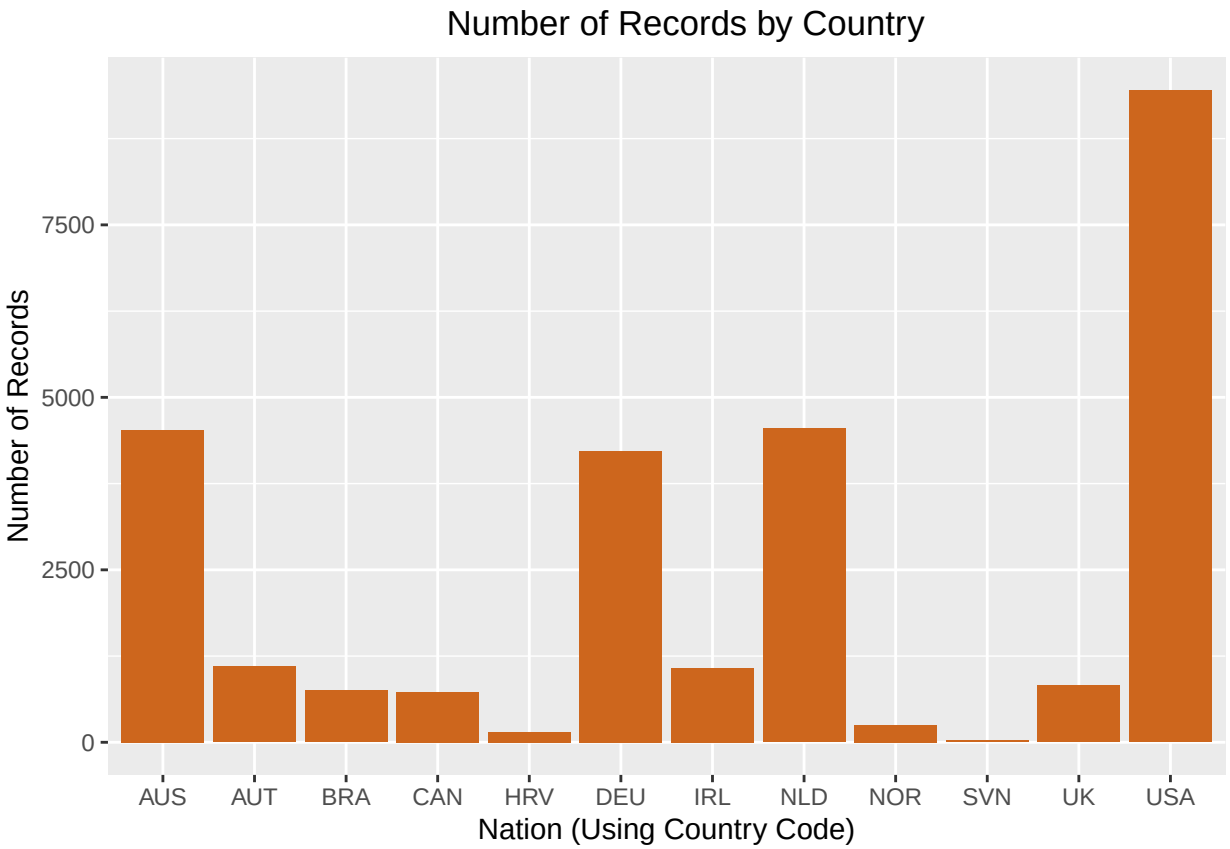
Let's complete the join using `left_join`.

```
# Let's take out year as it is information we already have.
three_laps_drivers <- three_laps %>%
  left_join(drivers, by = "player") %>%
  select(-year)


View(three_laps_drivers)
```

Lastly, let's try to plot a bar chart of number of records by country. Here are the countries and there corresponding codes for your reference.

Country	Country Code
Australia	AUS
Austria	AUT
Brazil	CAN
Canada	HRV
Germany	DEU
Ireland	IRL
Netherlands	NLD
Norway	NOR
Slovenia	SVN
United Kingdom	UK
United States	USA

```
# Let's take out the NA values from nations
three_laps_drivers %>%
    filter(!is.na(nation)) %>%
  ggplot(aes(x = nation)) +
  geom_bar(fill = "chocolate3") + labs(x = "Nation (Using Country Code)",
                                       y = "Number of Records",
                                     title = "Number of Records by Country") +
  scale_x_discrete(labels = c("AUS", "AUT", "BRA", "CAN", "HRV", "DEU", "IRL",
                              "NLD", "NOR", "SVN", "UK", "USA")) +
  theme(plot.title = element_text(hjust = 0.53))
```



Chapter 4

```
library(tidyverse)
library(dplyr)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

Question 1

First, let's import a raw data file from a git hub link.

```
nfl_salaries <- read.csv(paste0("https://raw.githubusercontent.com/",
                                "NicolasRestrep/223_course/main/Data/",
                                "nfl_salaries.csv"))
View(nfl_salaries)
```

Question 2

Now let's tidy up the data and combine the different columns/positions into one column called `positions` and add their values into a separate column called `salaries`.

```
nfl_salaries_tidy <- nfl_salaries %>%
  pivot_longer(names_to = "position",
               values_to = "salaries",
               cols = -year)
nfl_salaries_tidy
```

```
## # A tibble: 8,000 x 3
##     year position          salaries
##    <int> <chr>                <int>
##  1  2011 Cornerback        11265916
##  2  2011 Defensive.Lineman 17818000
##  3  2011 Linebacker        16420000
##  4  2011 Offensive.Lineman 15960000
##  5  2011 Quarterback       17228125
##  6  2011 Running.Back      12955000
##  7  2011 Safety             8871428
##  8  2011 Special.Teamer     4300000
##  9  2011 Tight.End          8734375
## 10  2011 Wide.Receiver     16250000
## # ... with 7,990 more rows
## # i Use 'print(n = ...)' to see more rows
```

Question 3

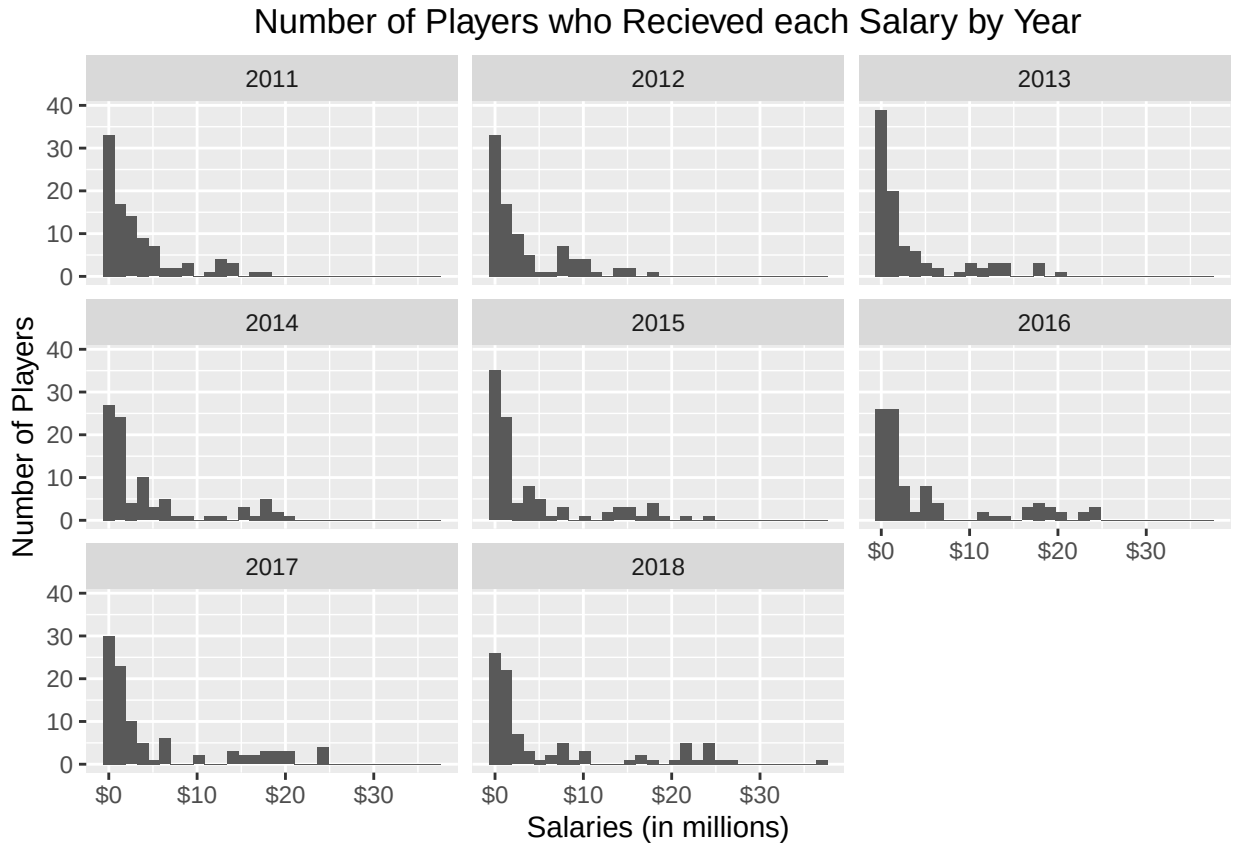
Let's make histograms for each year for quarter backs.

```
# Let's filter out quarter backs first and convert salaries to "in thousands"
qb_only <- nfl_salaries_tidy %>%
  filter(position == "Quarterback")

qb_only  <- qb_only %>%
   mutate(sal_in_millions = qb_only$salaries/1000000)


#First lets convert our salaries to "in thousands"

# Now let's create our histogram
ggplot(qb_only, mapping = aes(x = sal_in_millions)) +
  geom_histogram() +  facet_wrap(~year) + labs(x = "Salaries (in millions)",
```

```
                                        y = "Number of Players",
                                    title = paste0("Number of",
                                                   " Players who",
                                                   " Recieved each",
                                                   " Salary by Year")) +
  scale_x_continuous(labels = dollar) +
  theme(plot.title = element_text(hjust = 0.53))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 55 rows containing non-finite values (stat_bin).
```



What patterns do you notice? I notice that the the distribution is positively (right) skewed.This means a players make almost three times their counterparts while the rest (majority) make less than 10,000,000.

Question 4

Now, let's create a new dataset that contains the average salary for each position each year.

```
avg_pos_sal <- nfl_salaries_tidy %>%
  group_by(position,year) %>%
  summarize(avg_salaries = mean(salaries))
```

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

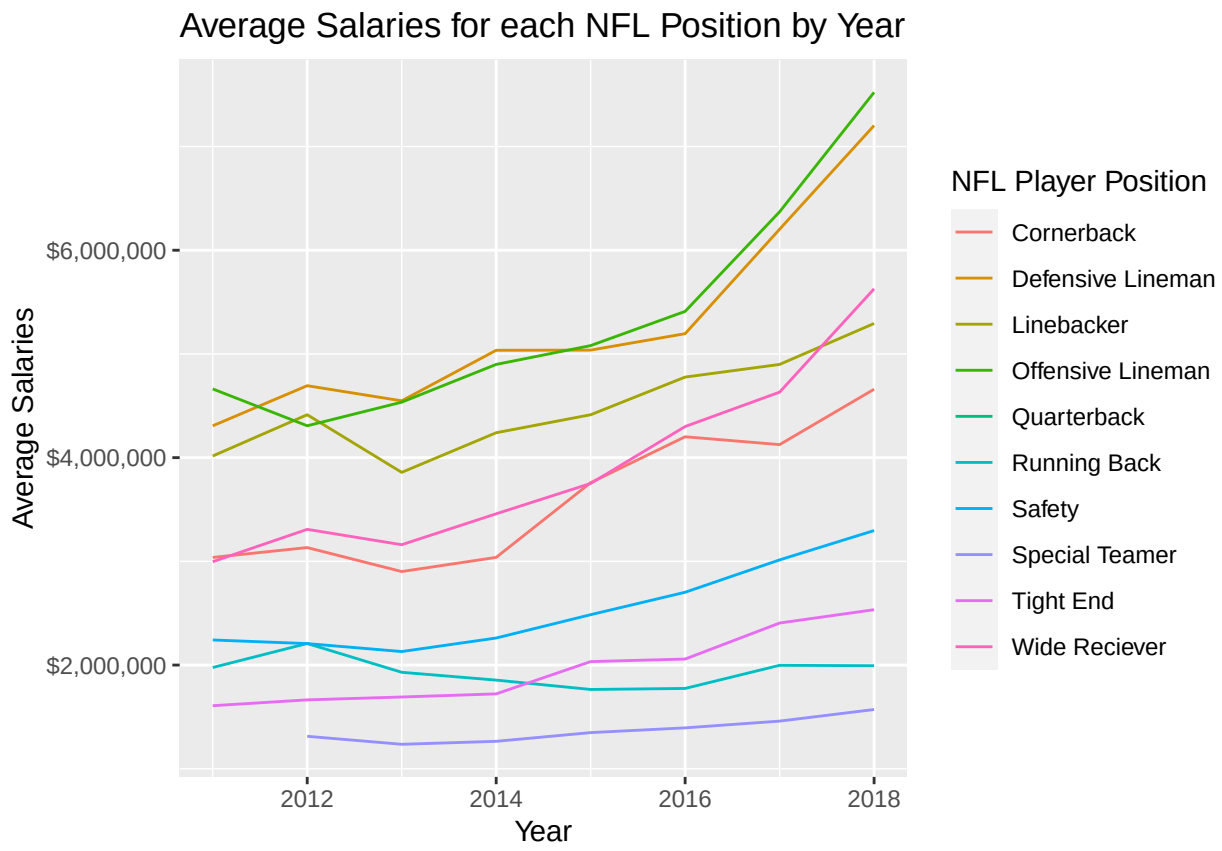
```
avg_pos_sal
```

```
## # A tibble: 80 x 3
## # Groups:   position [10]
##    position           year avg_salaries
##    <chr>             <int>        <dbl>
##  1 Cornerback         2011     3037766.
##  2 Cornerback         2012     3132916.
##  3 Cornerback         2013     2901798.
##  4 Cornerback         2014     3038278.
##  5 Cornerback         2015     3758543.
##  6 Cornerback         2016     4201470.
##  7 Cornerback         2017     4125692.
##  8 Cornerback         2018     4659704.
##  9 Defensive.Lineman  2011     4306995.
## 10 Defensive.Lineman  2012     4693730.
## # ... with 70 more rows
## # i Use `print(n = ...)` to see more rows
```

Question 5

```
ggplot(avg_pos_sal, mapping = aes(x = year, y = avg_salaries,
                                        col = position)) +
  geom_line() +
  scale_y_continuous(name = "Average Salaries", labels = dollar) +
  labs(x = "Year", title = "Average Salaries for each NFL Position by Year",
       color = "NFL Player Position") +
  scale_color_discrete(labels = c("Cornerback", "Defensive Lineman",
                                  "Linebacker", "Offensive Lineman",
                                  "Quarterback","Running Back", "Safety",
                                  "Special Teamer", "Tight End",
                                  "Wide Reciever" ))
```

```
## Warning: Removed 9 row(s) containing missing values (geom_path).
```



Describe at least two trends that are apparent to you.

1. Linemen positions have consistently made the most each year.
2. Overtime, most positions have seen a salary increase. Some noticeable positions where this is not the case is: Linebacker, Running Back, and Special Teamer.