Chapter 3, 4, & 5 - DV

Allyson Cameron

2022-09-13

Chapter 3

```r
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()
```

```r
# Read in the data
exercise_data <- read_csv("Data/visualize_data.csv")
```

```
## New names:
## Rows: 142 Columns: 4
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (4): ...1, ...2, Exercise, BMI
## 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.
## * ‘‘ -> ‘...1‘
## * ‘...1‘ -> ‘...2‘
```

```r
# Glimpse the data
glimpse(exercise_data)
```

```
## Rows: 142
## Columns: 4
## $ ...1     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
## $ ...2     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
## $ Exercise <dbl> 55.3846, 51.5385, 46.1538, 42.8205, 40.7692, 38.7179, 35.6410~
## $ BMI      <dbl> 1.8320590, 1.7892194, 1.7321050, 1.6178724, 1.5036362, 1.3751~
```

Question 1

Before, we examine anything from the data, write down what you expect the relationship would look like. **Do you think people who record more exercise will have more or less BMI?** I think people who exercise more will have a lower BMI.

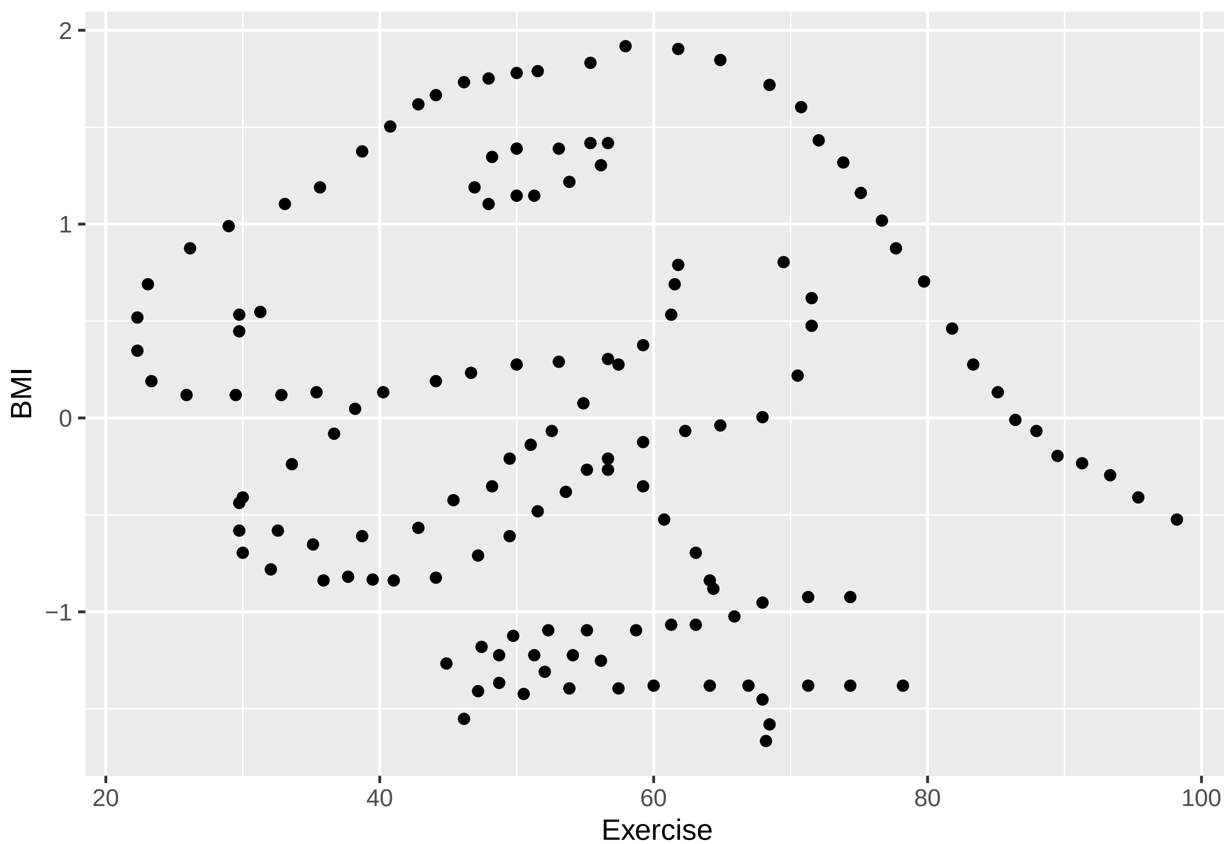
```
# see correlation
cor(exercise_data$Exercise, exercise_data$BMI)
```

```
## [1] -0.06447185
```

So far, it looks like my prediction is correct. We see a moderate, negative correlation. This means that as one increases their exercise their BMI should decrease, or as one decreases their exercise one's BMI should increase.

Let's plot the relationship to see it visually.

```
# create base of ggplot
a <- ggplot(exercise_data, aes(x = Exercise, y  = BMI))

# add type of ggplot (scatter)
a + geom_point()
```



I see a dinosaur! So the data is *definitely* not a negative correlation.

Question 2

First, let's install the `causact` package.

```
# Let's install the needed package
install.packages("causact")
```

Next, let's load the package and glimpse the dataset.

```
library(causact)

# Glimpse the data
glimpse(corruptDF)
```

```
## Rows: 174
## Columns: 7
## $ country     <chr> "Afghanistan", "Albania", "Algeria", "Angola", "Argentina"~
## $ region      <chr> "Asia Pacific", "East EU Cemt Asia", "MENA", "SSA", "Ameri~
## $ countryCode <chr> "AFG", "ALB", "DZA", "AGO", "ARG", "ARM", "AUS", "AUT", "A~
## $ regionCode  <chr> "AP", "ECA", "MENA", "SSA", "AME", "ECA", "AP", "WE/EU", "~
## $ population  <int> 35530081, 2873457, 41318142, 29784193, 44271041, 2930450, ~
## $ CPI2017     <int> 15, 38, 33, 19, 39, 35, 77, 75, 31, 65, 36, 28, 68, 44, 75~
## $ HDI2017     <dbl> 0.498, 0.785, 0.754, 0.581, 0.825, 0.755, 0.939, 0.908, 0.~
```

```
# Let's see what each variable captures
?corruptDF
```

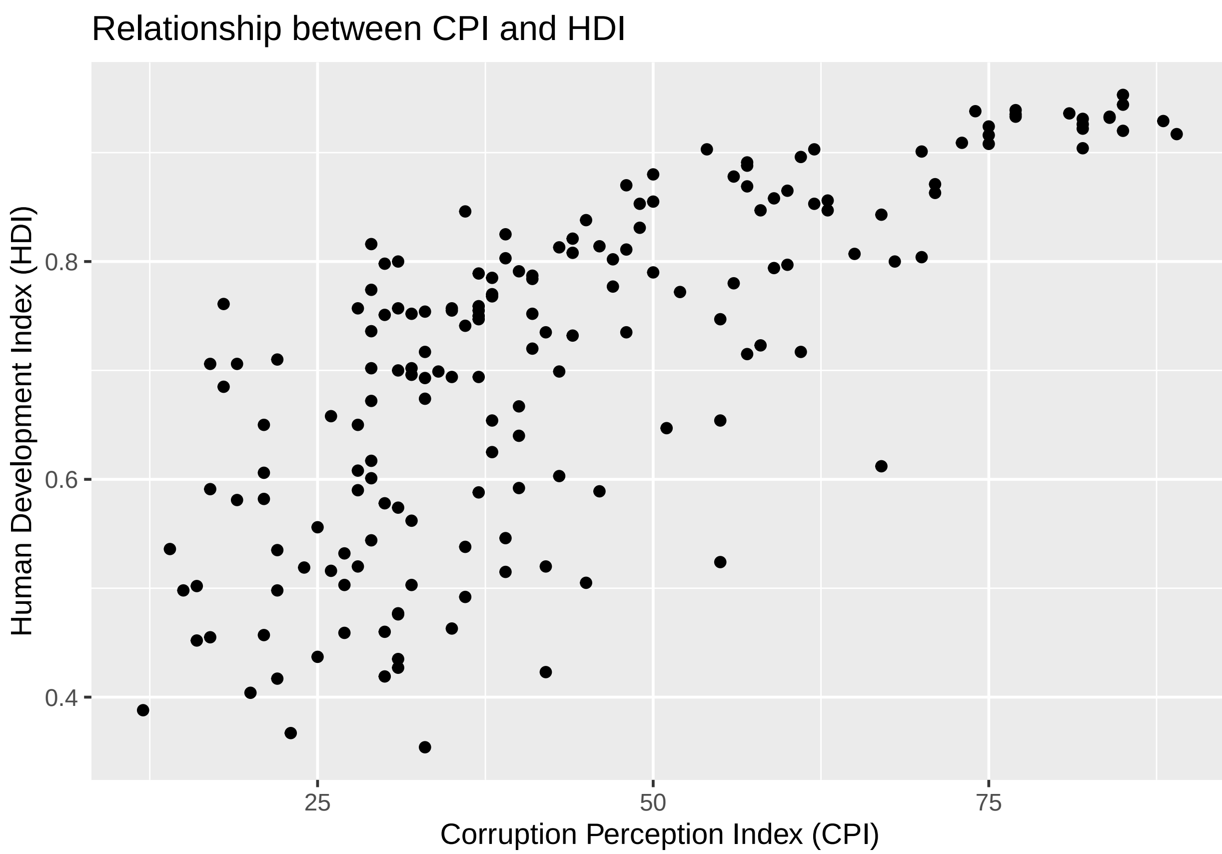
1. What does `CPI2017` capture? It is showing the percieved level of corruption within the public sector based on a scale from 0 (very corrupt) to 100 (not corrupt at all).

2. What does `HDI2017` capture? It is showing a countries level of human development based on how well they achieve certain dimensions (nation longevity, education and income) associated with (human) development.

Question 3

Now, let's make a scatterplot to see the relationship between these variables.

```
# Let's make a scatterplot, base first
b <- ggplot(corruptDF, aes(x = CPI2017, y = HDI2017))

# add geompoint() to base
b + geom_point() + labs(x = "Corruption Perception Index (CPI)",
                        y = "Human Development Index (HDI)",
                        title = "Relationship between CPI and HDI")
```



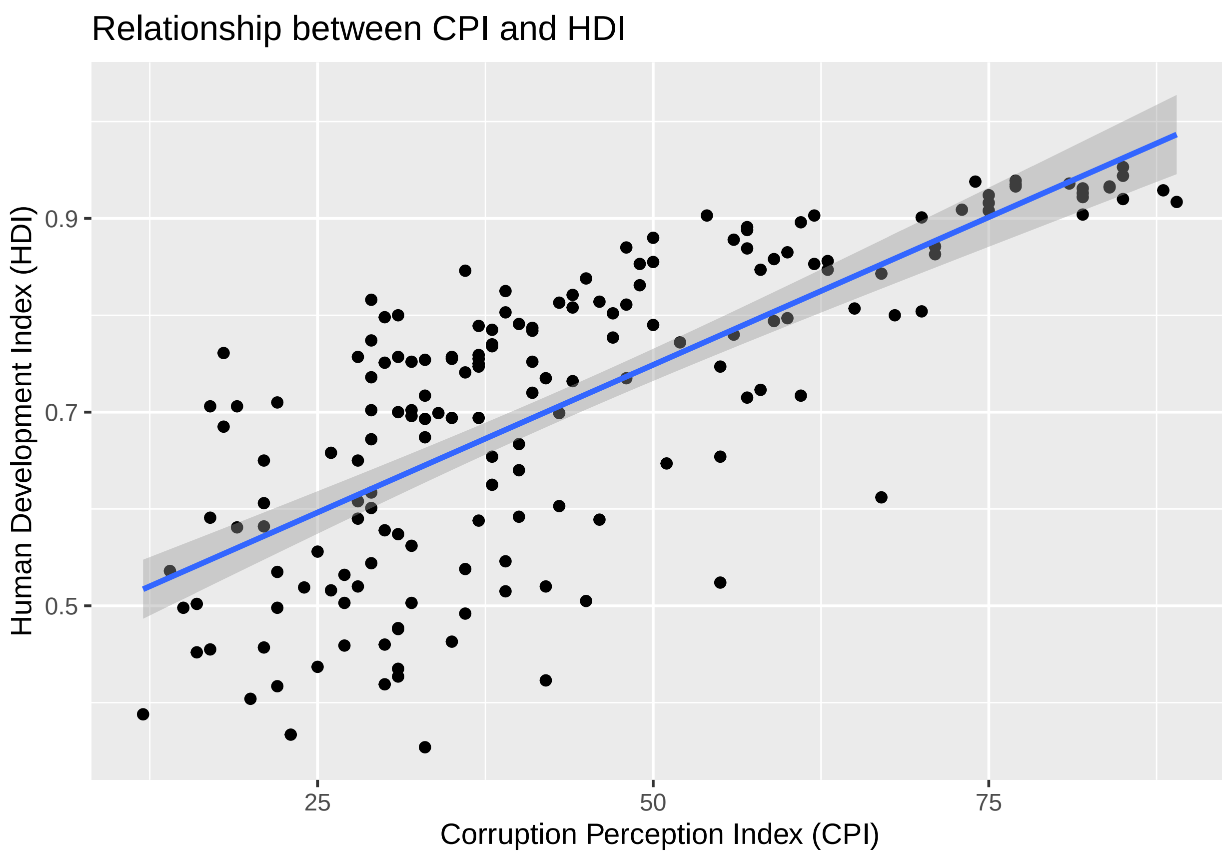
There is a positive relationship relationship between `CPI2017` and `HDI2017` this means that the more corrupt a country is perceived the less likely they are to achieve at dimensions of human development.

Question 4

Now, lets add a layer that captures the overall relationship between these two variables.

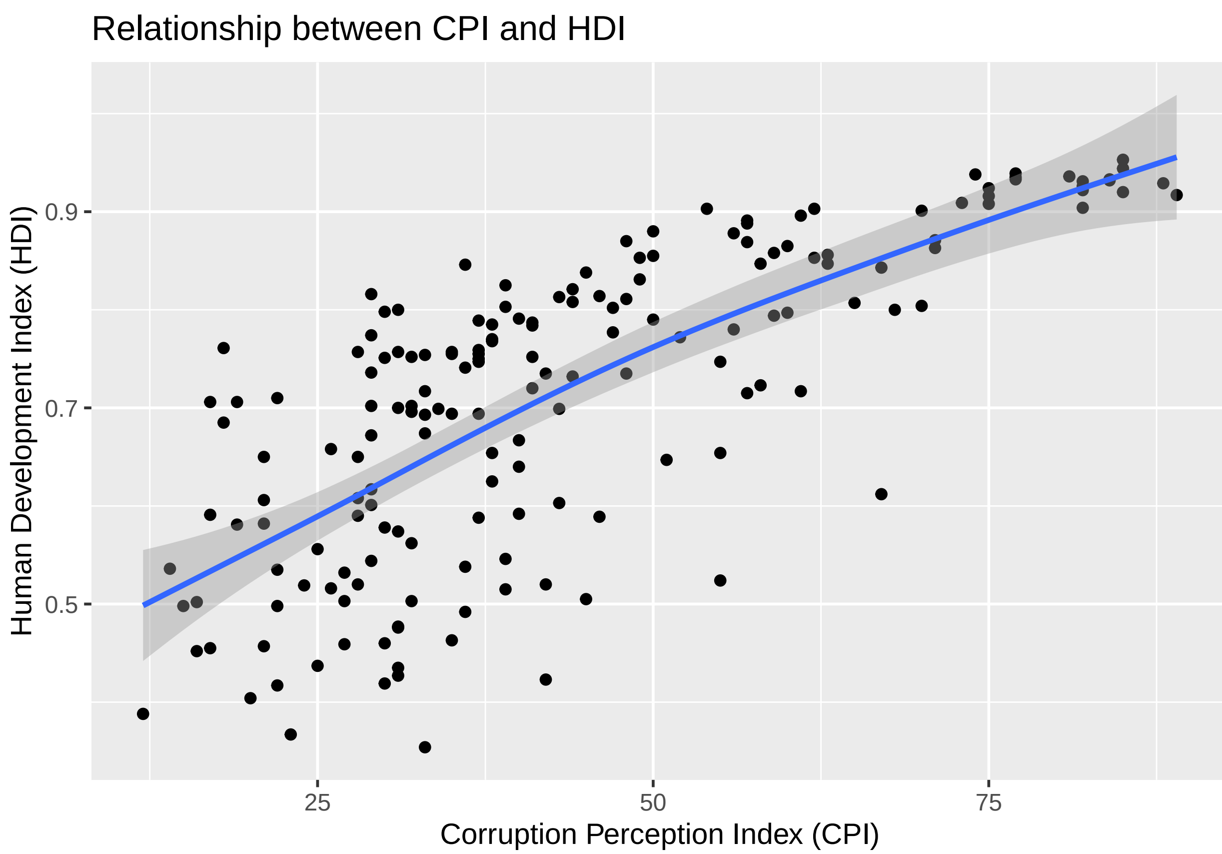
```
# lets use lm method
b + geom_point() + geom_smooth(method = "lm") +
  labs(x = "Corruption Perception Index (CPI)",
       y = "Human Development Index (HDI)",
       title = "Relationship between CPI and HDI")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# lets use gam method
b + geom_point() + geom_smooth(method = "gam") +
  labs(x = "Corruption Perception Index (CPI)",
       y = "Human Development Index (HDI)",
       title = "Relationship between CPI and HDI")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

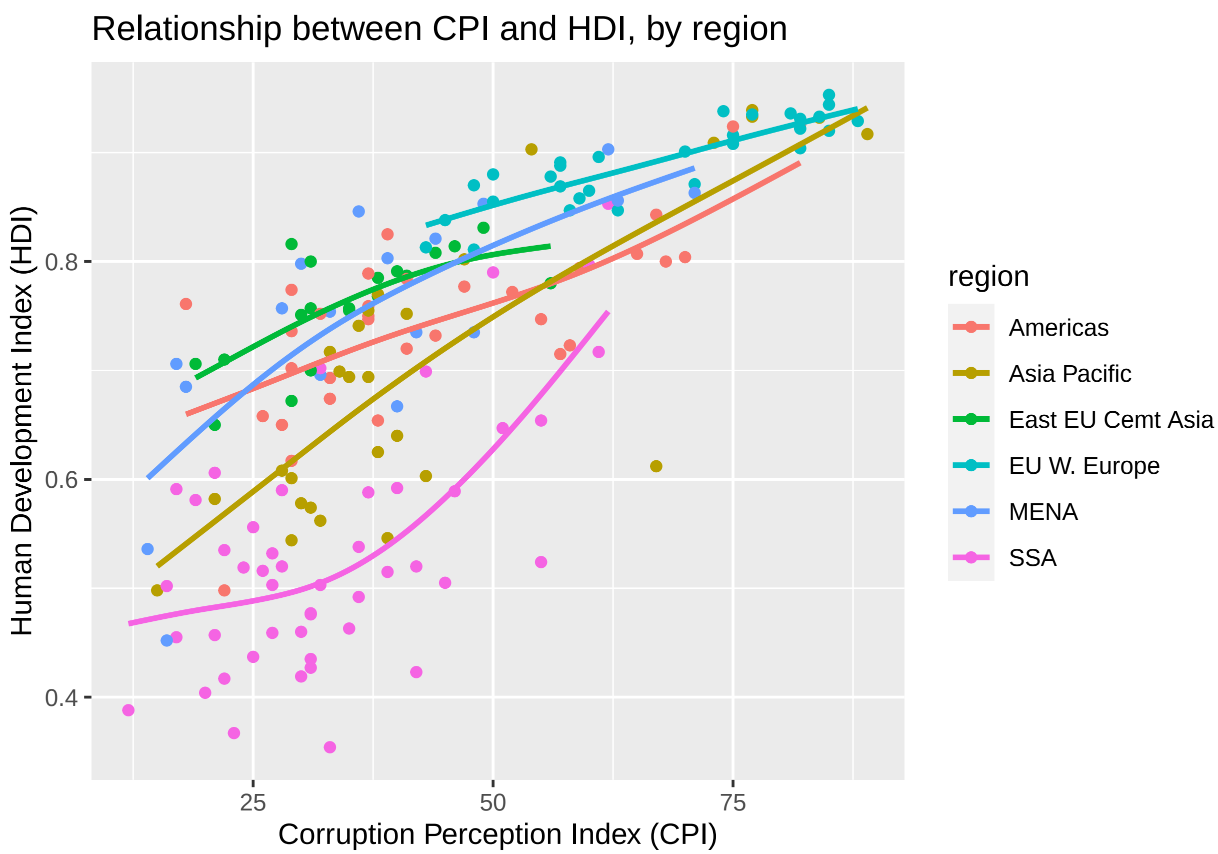


I prefer the `gam` method because although the standard error is larger for the line the standard error still seems to encompass the points present while the `lm` method seems to not align as well towards the top right of the line/graph.

Question 5

```
# create region base of ggplot
b_by_region <- ggplot(corruptDF, aes(x = CPI2017, y = HDI2017,
                                     color = region, fill = region))

# add geom_point and other details
b_by_region  + geom_point() + geom_smooth(method = "gam", se = FALSE) +
  labs(x = "Corruption Perception Index (CPI)",
       y = "Human Development Index (HDI)",
        title = "Relationship between CPI and HDI, by region")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



1. **What do you see?**
   I see that Sub-Saharan Africa has some of the most perceived corrupt countries along with the lowest achievement levels on the dimensions of human development. Additionally, EU W. Europe has some of the least perceived corruption and the highest achievement levels on the dimensions of human development.

2. **Are patterns clear or is the graph too cluttered? What would be another way to get these trends by region but in a way to would be more legible?**

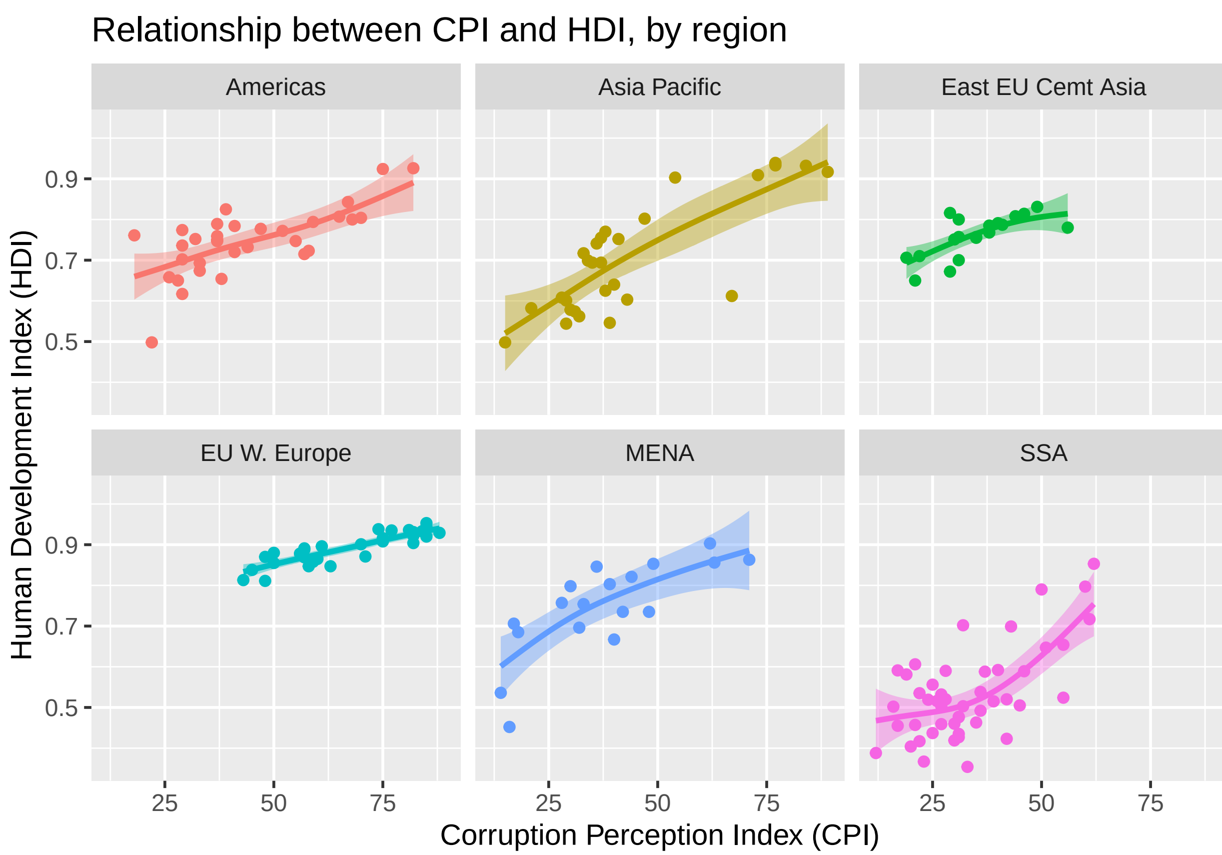
While I can kind of read the graph, it is way too cluttered (I even tried taking off the standard errors to make it more legible!). Below I will show you another way to see these trends that is more legible.

```
# create  new ggplot using base
# adding facet wrap for region and group by country
b_region_facet <- b_by_region  + geom_point(aes(group = country)) +
  geom_smooth(method = "gam") +
  facet_wrap(~region) +
  guides(fill = FALSE, color = FALSE) +
  labs(x = "Corruption Perception Index (CPI)",
       y = "Human Development Index (HDI)",
       title = "Relationship between CPI and HDI, by region")
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```r
b_region_facet
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



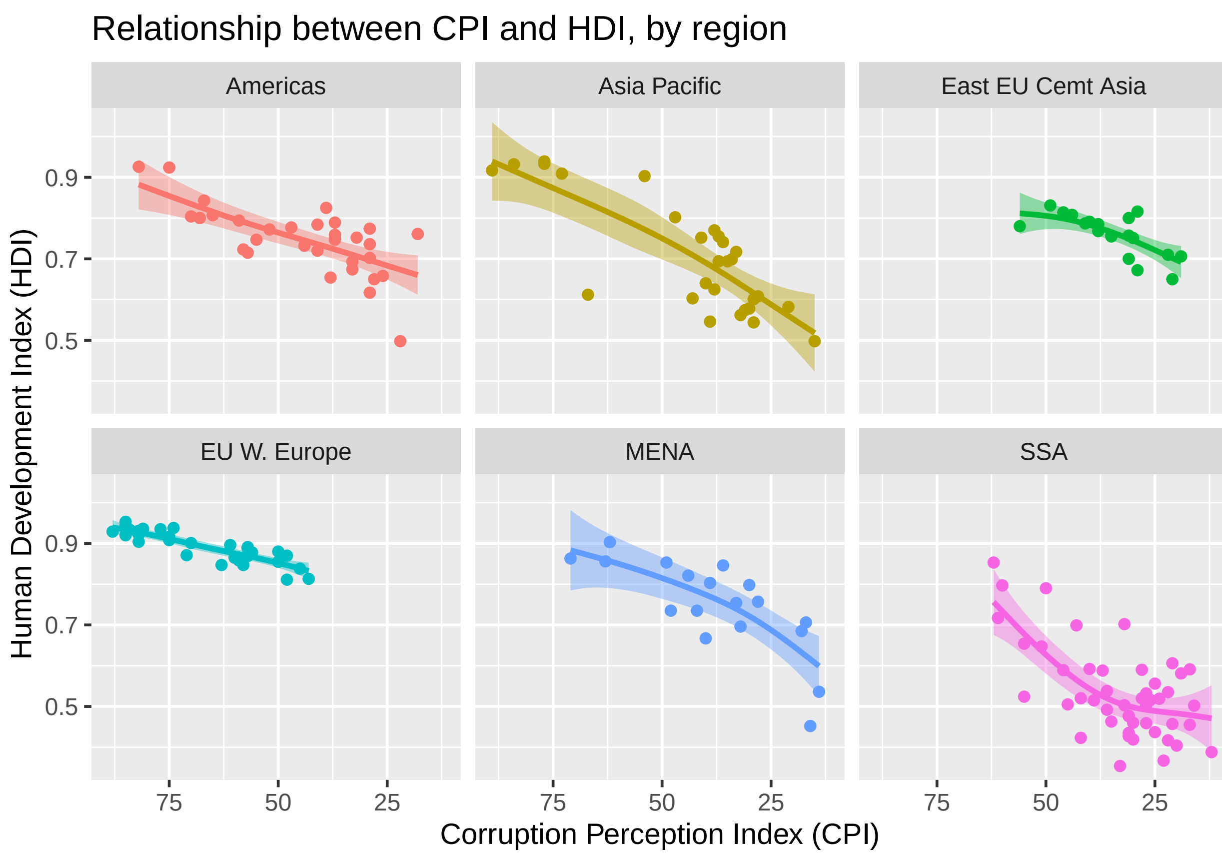
Now we can see the trends for each region separately instead of having them all overlapping.

Question 6

Now, lets reverse `CPI2017` so that the lower side of the graph shows low levels of corruption (100) instead of higher levels of corruption (0)

```r
#reverse the x scale
facet_reverse <- b_region_facet + scale_x_reverse()

facet_reverse
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

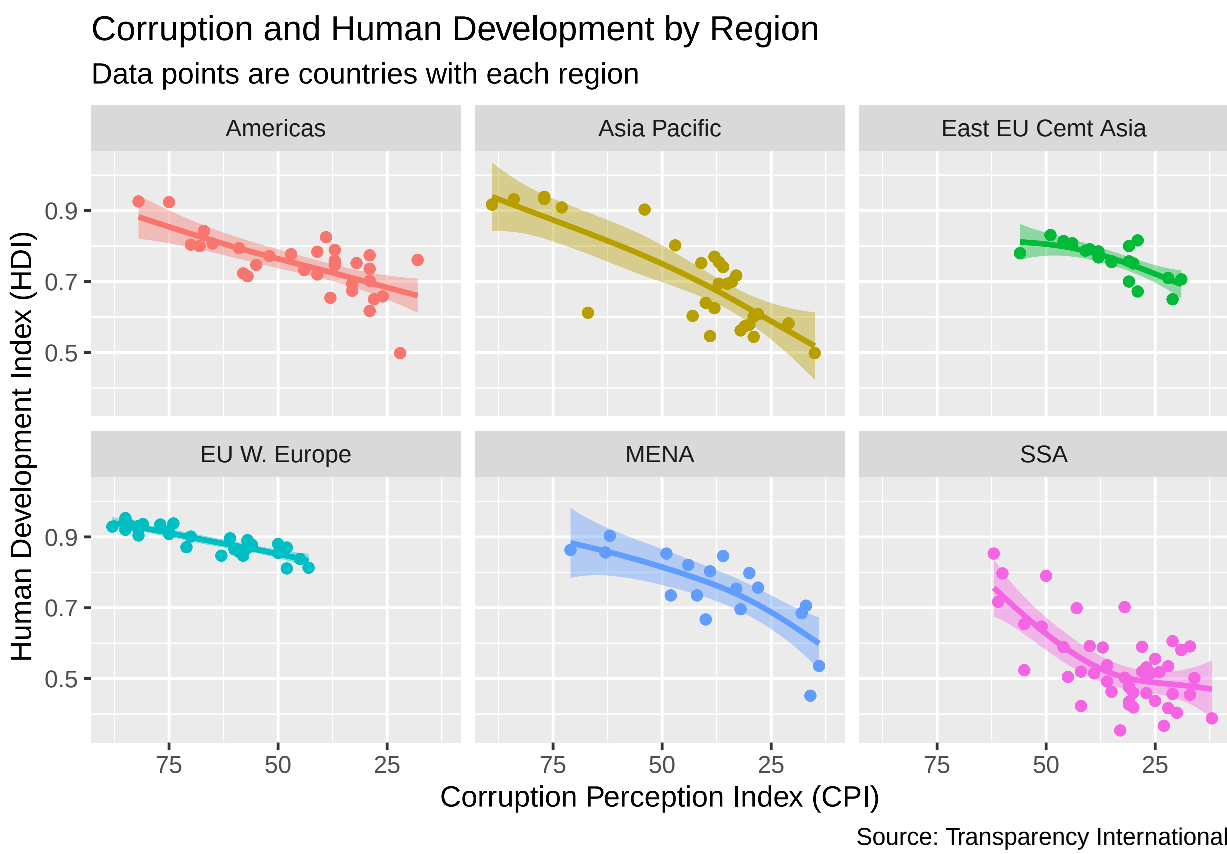


Question 7

Let's add a title and subtitle to the plot along with a caption.

```
# where is the data from
?corruptDF

# create output with titles and caption
output <- facet_reverse +
  labs(title = "Corruption and Human Development by Region",
                subtitle = "Data points are countries with each region",
                caption = "Source: Transparency International" )

output
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



Question 8

Now lets save it for my *wonderful* supervisor.

```
# Save the data
ggsave(filename = "Chapter 3 Figure.pdf", plot = output)
```

```
## Saving 6.5 x 4.5 in image
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

Chapter 4

Question 1

Lets load `tidyverse` and read the data.

```
library(tidyverse)

# Read in the data
tv_ratings <- read_csv("Data/tv_ratings.csv")
```

```
## Rows: 2266 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): titleId, title, genres
## dbl  (3): seasonNumber, av_rating, share
## 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.
```

```
# Glimpse the data
glimpse(tv_ratings)
```

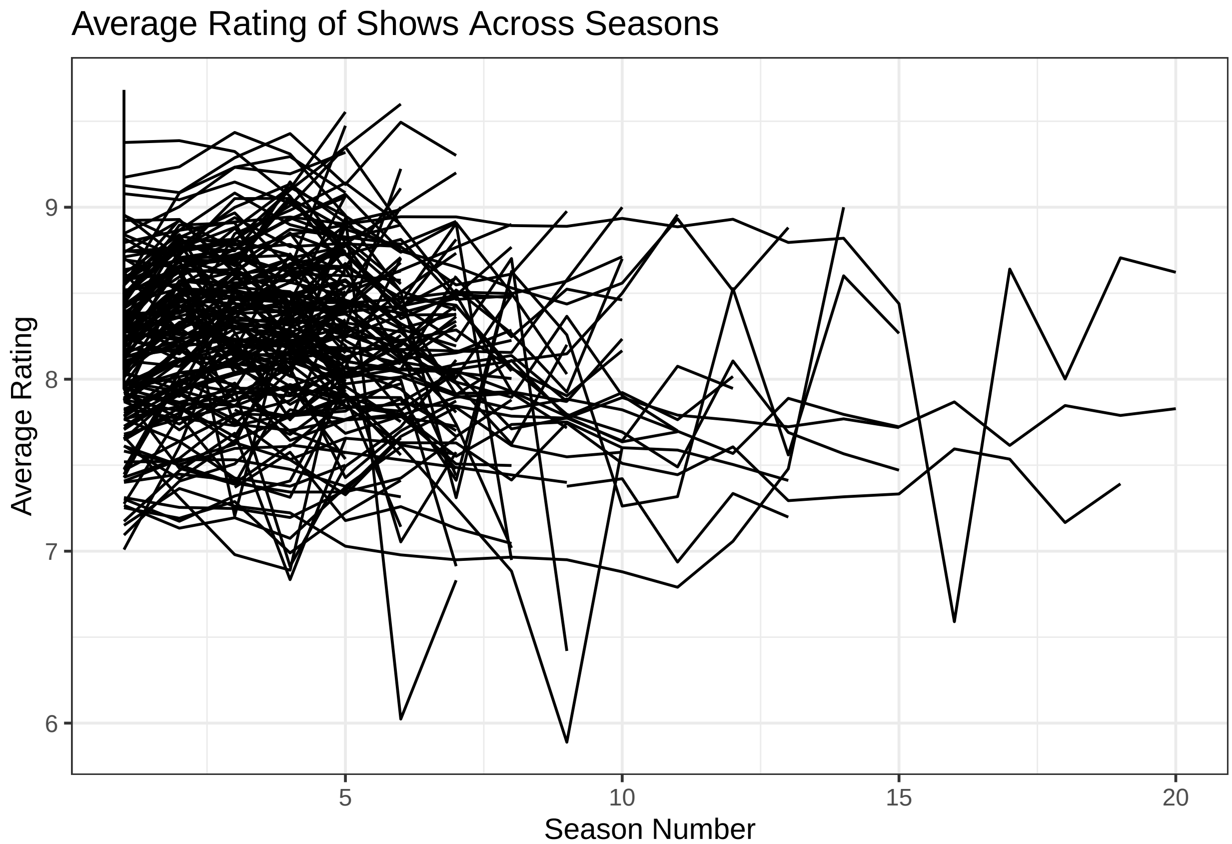
```
## Rows: 2,266
## Columns: 7
## $ titleId      <chr> "tt2879552", "tt3148266", "tt3148266", "tt3148266", "tt31~
## $ seasonNumber <dbl> 1, 1, 2, 3, 4, 1, 2, 1, 2, 3, 4, 5, 6, 7, 8, 1, 1, 1, 1, ~
## $ title        <chr> "11.22.63", "12 Monkeys", "12 Monkeys", "12 Monkeys", "12~
## $ date         <date> 2016-03-10, 2015-02-27, 2016-05-30, 2017-05-19, 2018-06--~
## $ av_rating    <dbl> 8.4890, 8.3407, 8.8196, 9.0369, 9.1363, 8.4370, 7.5089, 8~
## $ share        <dbl> 0.51, 0.46, 0.25, 0.19, 0.38, 2.38, 2.19, 6.67, 7.13, 5.8~
## $ genres       <chr> "Drama,Mystery,Sci-Fi", "Adventure,Drama,Mystery", "Adven~
```

Next let's find out how many shows have 5 seasons or more.

```
# create var with total number of seasons
tv_long <- tv_ratings %>%
  group_by(title) %>%
  summarize(num_seasons = n()) %>%
  ungroup() %>%
  left_join(tv_ratings, by = "title")

# filter for >5 seasons
tv_long <- tv_long %>%
  filter(num_seasons >= 5)

# create dataframe with only 1 entry per show
number_by_title <- tv_long %>%
  group_by(title) %>%
  slice(1) %>%
  select(title, num_seasons) %>%
  arrange(desc(num_seasons))
```

Now, using `tv_long` lets create a line plot across seasons for average ratings.

```
# here is the base of the plot
rate_season <- ggplot(tv_long, aes(x = seasonNumber, y = av_rating))

# here is the complete plot
rate_season_complete <- rate_season + geom_line(aes(group = title)) +
  labs(x = "Season Number", y = "Average Rating",
       title = "Average Rating of Shows Across Seasons") + theme_bw()

rate_season_complete
```

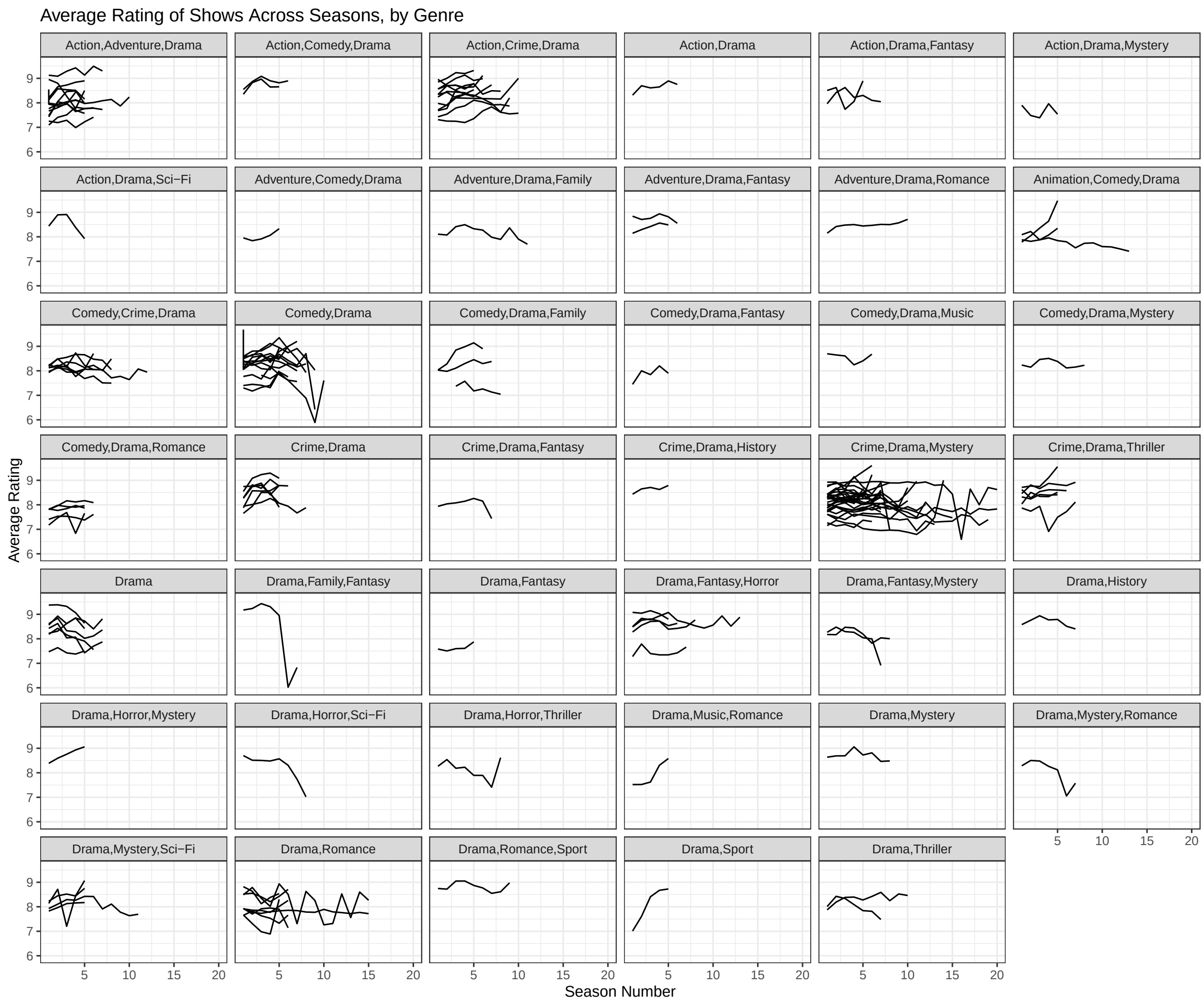


This plot is extremely messy. From it though, I gather than not many shows make it past about 12 seasons. Additionally, most shows start out with a rating of at least 7 out of 10.

Question 2

Now, lets make it easier to read by facet wrapping.

```
rate_season_complete + facet_wrap(~genres, ncol = 6) +
  labs(title = "Average Rating of Shows Across Seasons, by Genre")
```



What shows tend to last longer? Do ratings change much across seasons? Shows in the `Crime, Drama, Mystery` genre tend to last longer. In most cases, shows seem to have small rates of change in their ratings across seasons, but there is definitely a change. There are a few exceptions like `Drama, Family, Fantasy` or `Drama, Sport`.

Can you identify that show on Drama, Family, Fantasy whose ratings just plummeted?

```
# tidy helps you see every step in the process, gives you back a tibble
plummeted <- tv_long %>%
  filter(genres == "Drama,Family,Fantasy") %>%
  select(title)

plummeted
```

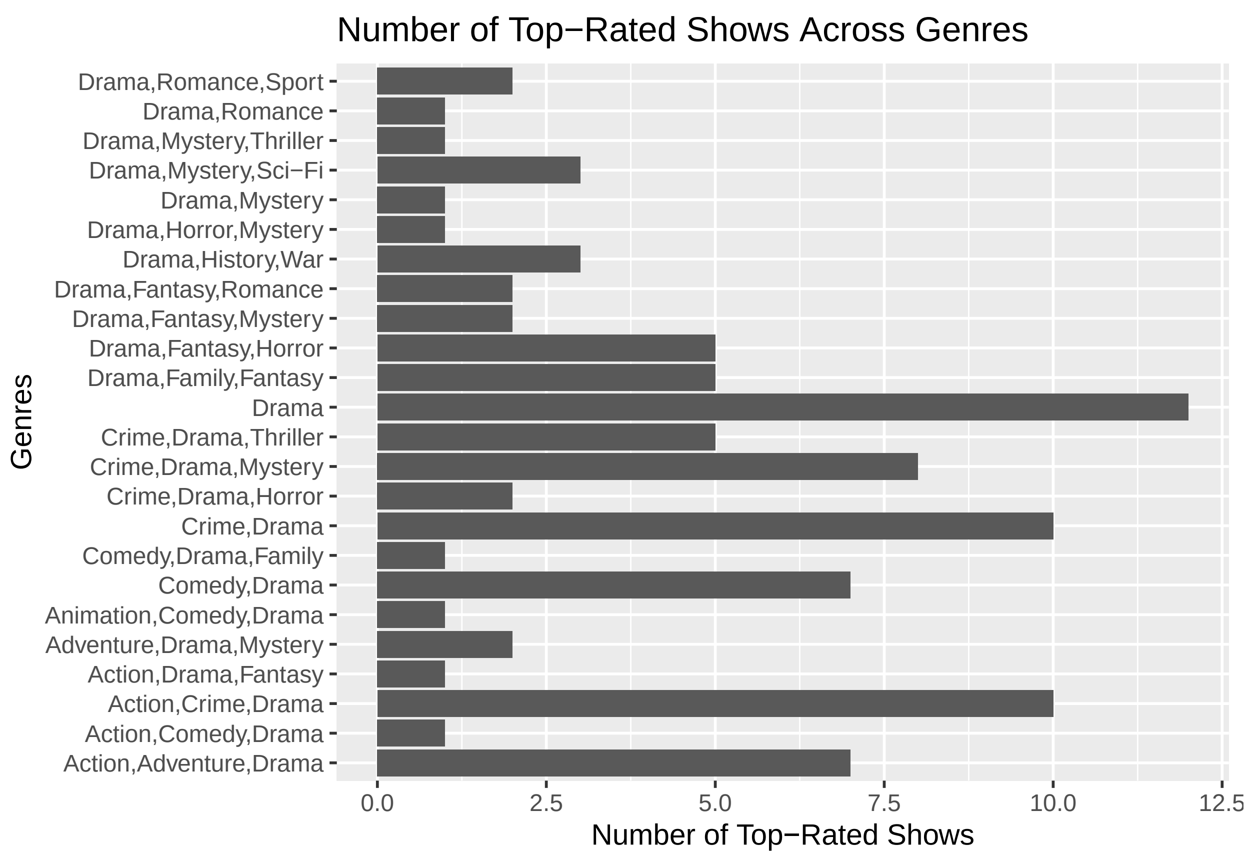
```
## # A tibble: 7 x 1
##   title
##   <chr>
## 1 Are You Afraid of the Dark?
## 2 Are You Afraid of the Dark?
## 3 Are You Afraid of the Dark?
## 4 Are You Afraid of the Dark?
```

```
## 5 Are You Afraid of the Dark?
## 6 Are You Afraid of the Dark?
## 7 Are You Afraid of the Dark?
```

The show is "Are You Afraid of the Dark?"

Question 3

```r
high_rating <- tv_ratings %>%
  filter(av_rating >= 9) %>%
  mutate(Genres = genres) %>%
  select(Genres,av_rating)


# create barplot
high_rating_box <- ggplot(high_rating, aes(x = Genres))

high_rating_box + geom_bar() + coord_flip() +
  labs(y = "Number of Top-Rated Shows",
       title = "Number of Top-Rated Shows Across Genres")
```



`coord_flip` changes the x and y axes to the opposite coordinate positions. Drama has the most top-rated shows.

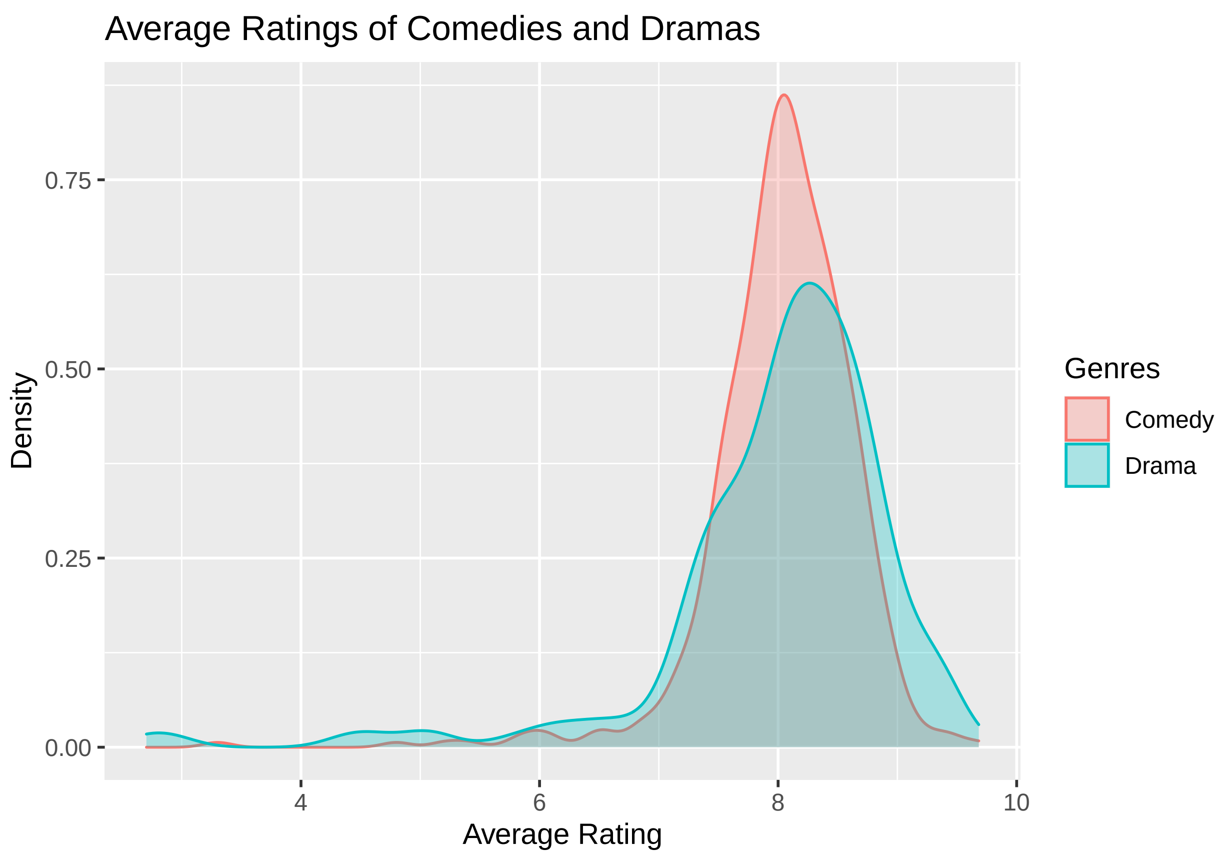
Question 4

```
# lets create an object with all genre categories with comedgy in it as comedy
# or with all dramas under drama
comedies_dramas <- tv_ratings %>%
  mutate(is_comedy = if_else(str_detect(genres, "Comedy"),
                              1,
                              0)) %>%
                             # If it contains the word comedy then 1, else 0
  filter(is_comedy == 1 | genres == "Drama") %>% # Keep comedies and dramas
  mutate(Genres = if_else(genres == "Drama",
                          # Make it so that we only have those two genres
                          "Drama",
                          "Comedy"))

glimpse(comedies_dramas)
```

```
## Rows: 684
## Columns: 9
## $ titleId      <chr> "tt0312081", "tt0312081", "tt0312081", "tt1225901", "tt12~
## $ seasonNumber <dbl> 1, 2, 3, 1, 2, 3, 4, 5, 1, 2, 1, 25, 1, 1, 2, 3, 4, 5, 1,~
## $ title        <chr> "8 Simple Rules", "8 Simple Rules", "8 Simple Rules", "90~
## $ date         <date> 2002-09-17, 2003-11-04, 2004-11-12, 2009-01-03, 2009-11-~
## $ av_rating    <dbl> 7.5000, 8.6000, 8.4043, 7.1735, 7.4686, 7.6858, 6.8344, 7~
## $ share        <dbl> 0.03, 0.10, 0.06, 0.40, 0.14, 0.10, 0.04, 0.01, 0.48, 0.4~
## $ genres       <chr> "Comedy,Drama", "Comedy,Drama", "Comedy,Drama", "Comedy,D~
## $ is_comedy    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Genres       <chr> "Comedy", "Comedy", "Comedy", "Comedy", "Comedy", "Comedy~
```

Now, let's make a density plot (exciting).

```
cd_plot <- ggplot(comedies_dramas, aes(x = av_rating, fill = Genres,
                                        color = Genres))
cd_plot + geom_density(alpha = 0.3) +
  labs(x = "Average Rating", y = "Density",
       title = "Average Ratings of Comedies and Dramas")
```



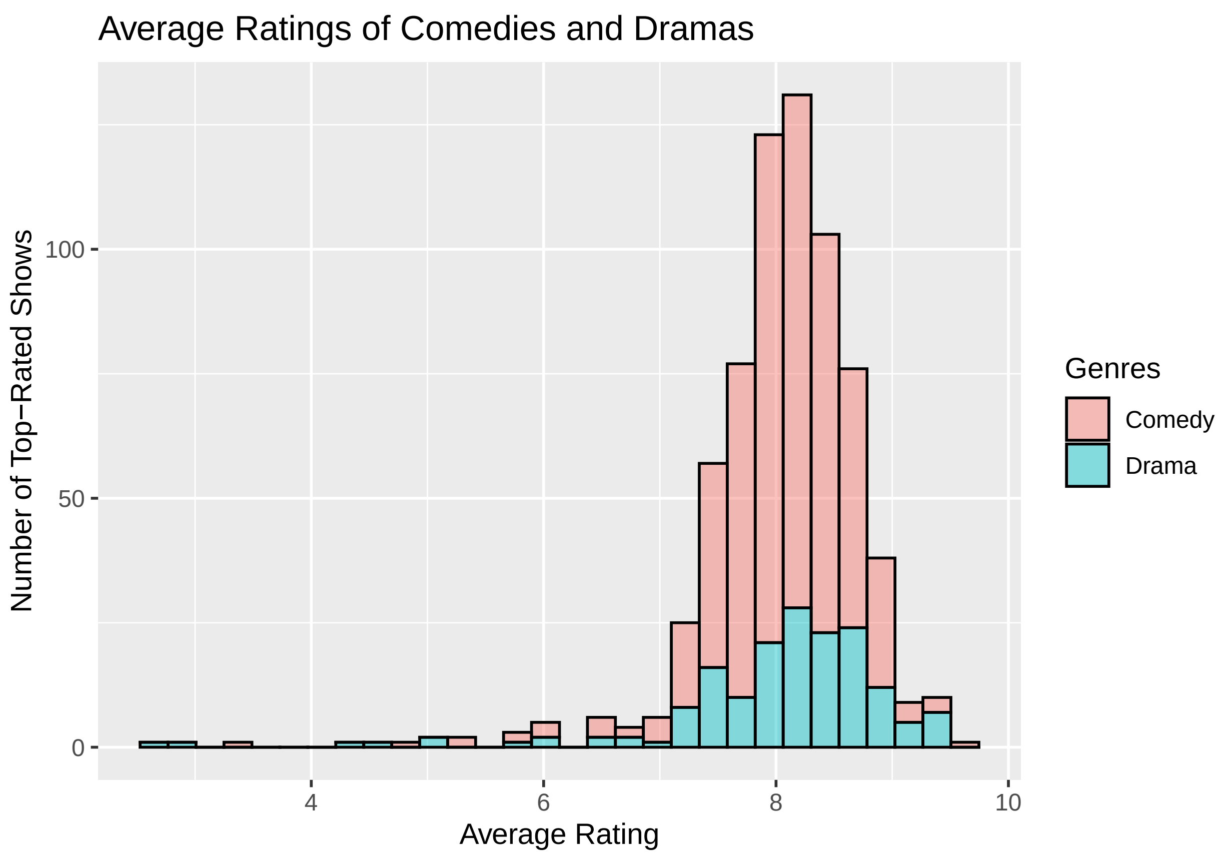
How does my prediction above hold? Are dramas rated higher?
I believe that this is showing that there are actually a lot of comedies that are still highly rated, they are just more often rated an 8, which was past our cutoff.

Question 5

Let's try some other ways of visualizing this data

```
# let's try a histogram
cd_hist <- ggplot(comedies_dramas, aes(x = av_rating, fill = Genres))
cd_hist + geom_histogram(color = "black", alpha = 0.45) +
  labs(x = "Average Rating", y = "Number of Top-Rated Shows",
       title = "Average Ratings of Comedies and Dramas")
```

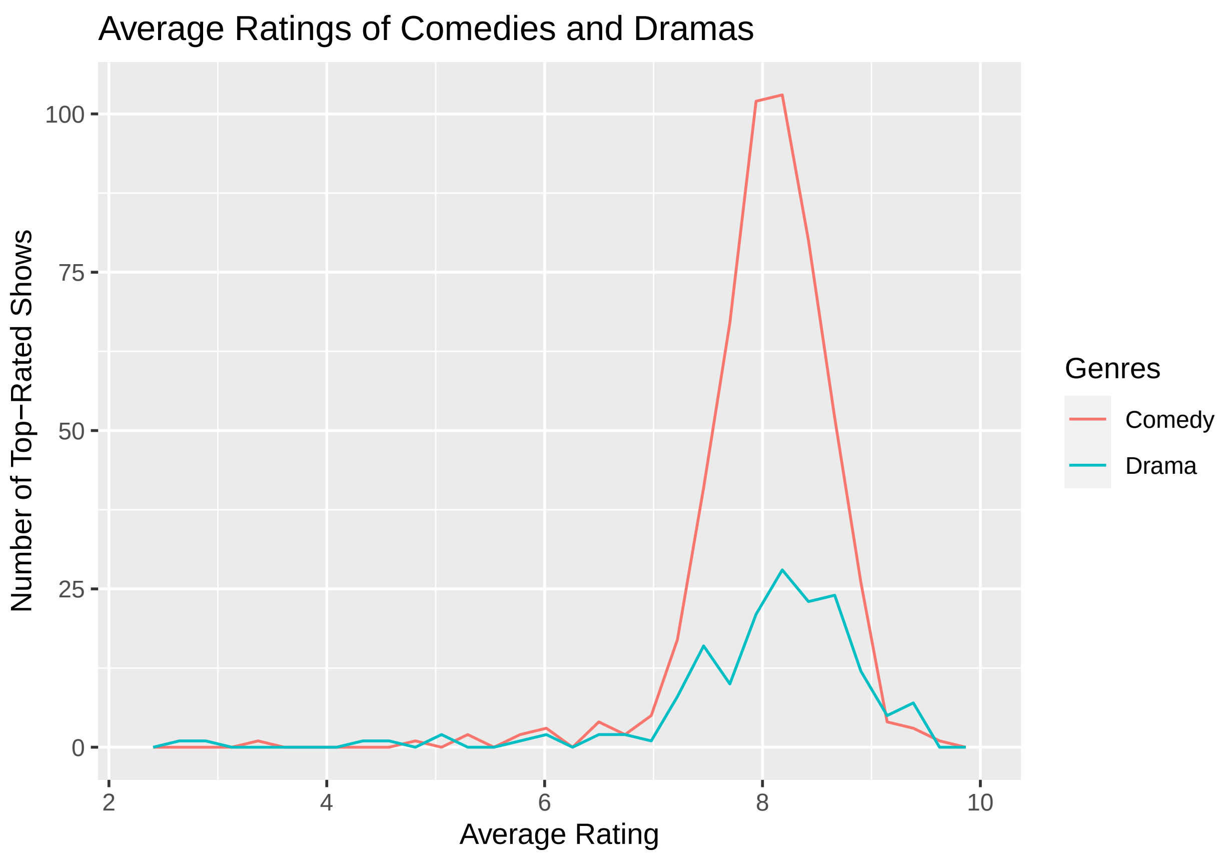
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



With the histogram, we can reach the same conclusion on the number of comedies still being higher. However, now we are also able to see a count of how many shows are at what rating.

```
# let's try a frequency poly
cd_freqpoly <- ggplot(comedies_dramas, aes(x = av_rating,
                                           color = Genres))
cd_freqpoly + geom_freqpoly() +
    labs(x = "Average Rating", y = "Number of Top-Rated Shows",
         title = "Average Ratings of Comedies and Dramas")
```

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



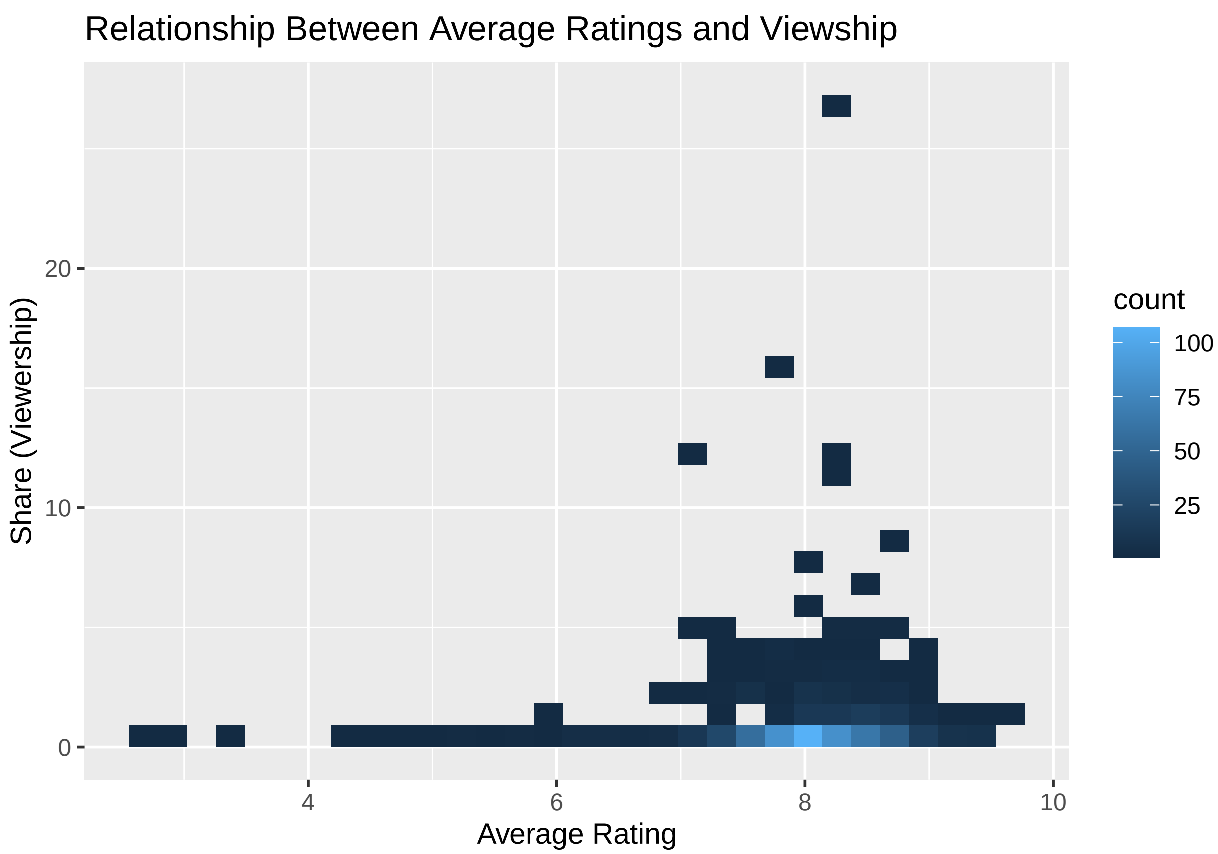
With the frequency polygon, we can now see the count along with the same graph structure as the density plot.

I believe the frequency polygon is the most informative because it allows for a structure that is easier to read while also giving us the count which is easiest to conceptualize over density.

Question 6

Now lets explore whether the actual quality of the show corresponded to viewership.

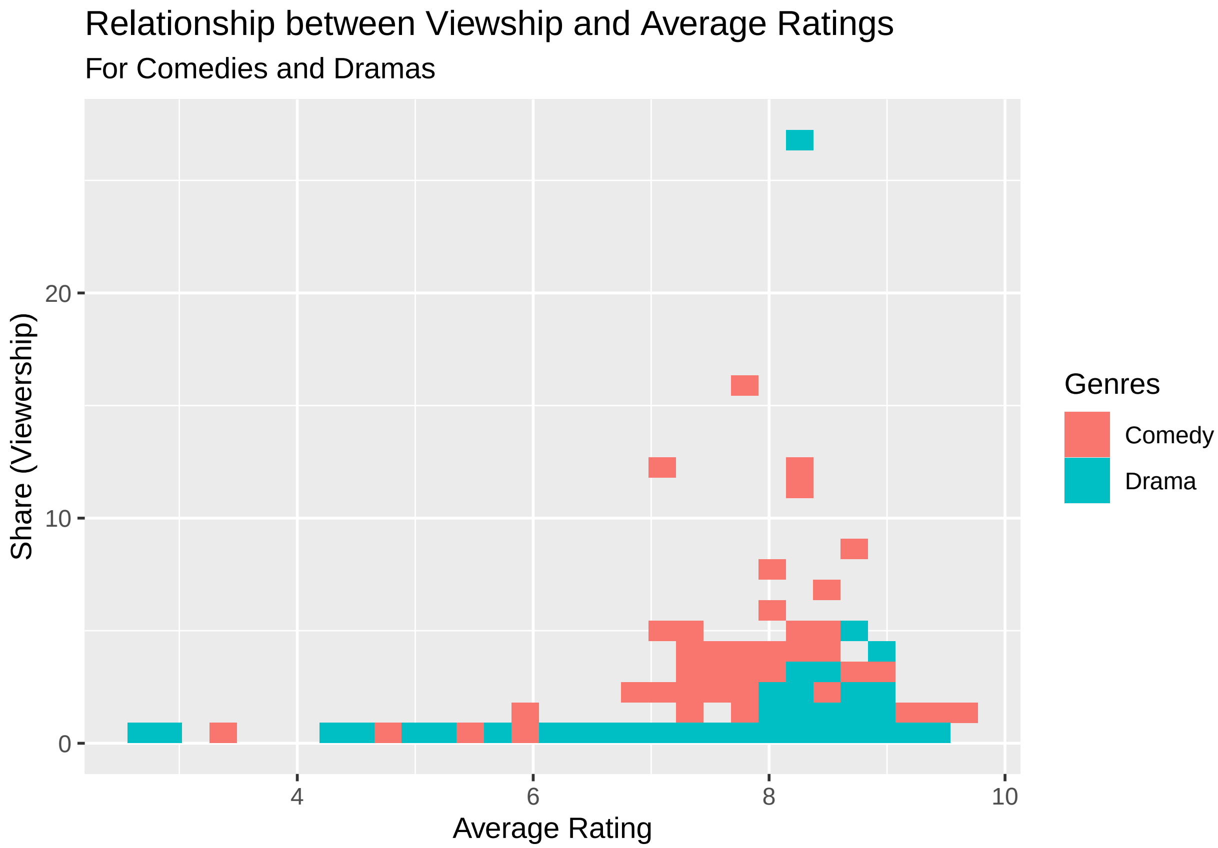
```
rating_viewship_plot <- ggplot(comedies_dramas, aes(x = av_rating, y = share))

rating_viewship_plot + geom_bin_2d() +
  labs (x = "Average Rating", y = "Share (Viewership)",
        title = "Relationship Between Average Ratings and Viewship")
```



Now we know that there were many shows with low viewership with pretty high ratings. This graph gives us additional information of the relative count of shows with each respective average rating *and* their viewership (in other words, we can see the counts of two variables).

Now, let's see how this looks with genre in the fill aesthetic.

```
rvg_plot <- ggplot(comedies_dramas, aes(x = av_rating, y = share,
                                        fill = Genres))

rvg_plot + geom_bin_2d() +
  labs(x = "Average Rating",y = "Share (Viewership)",
       title = "Relationship between Viewship and Average Ratings",
       subtitle = "For Comedies and Dramas")
```



What pattern do you see?

I see that comedies seem to have more viewership than dramas, especially the higher the rating (except for the one outlier which is a drama).

Lastly, let's find out the title of this outlier.

```
# I'm going to utilize the graph
# since I know that all other viewership numbers were less than 20
# lets just filter for the share that is greater than 20.

comedies_dramas %>%
  filter(share > 20)
```

```
## # A tibble: 1 x 9
##   titleId   seasonNumber title    date       av_rat~1 share genres is_co~2 Genres
##   <chr>            <dbl> <chr>    <date>        <dbl> <dbl> <chr>    <dbl> <chr>
## 1 tt0092337            1 Dekalog 1990-04-13     8.22  27.2 Drama        0 Drama
## # ... with abbreviated variable names 1: av_rating, 2: is_comedy
```

The show is called Dekalog.

Chapter 5

First, let's begin by reading the data and loading `tidyverse`.

```
library(tidyverse)
# Read in the data
wncaa <- read_csv("Data/wncaa.csv")
```

```
## Rows: 2092 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): school, conference, conf_place, how_qual, x1st_game_at_home, tourn...
## dbl (13): year, seed, conf_w, conf_l, conf_percent, reg_w, reg_l, reg_percen...
##
## 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.
```

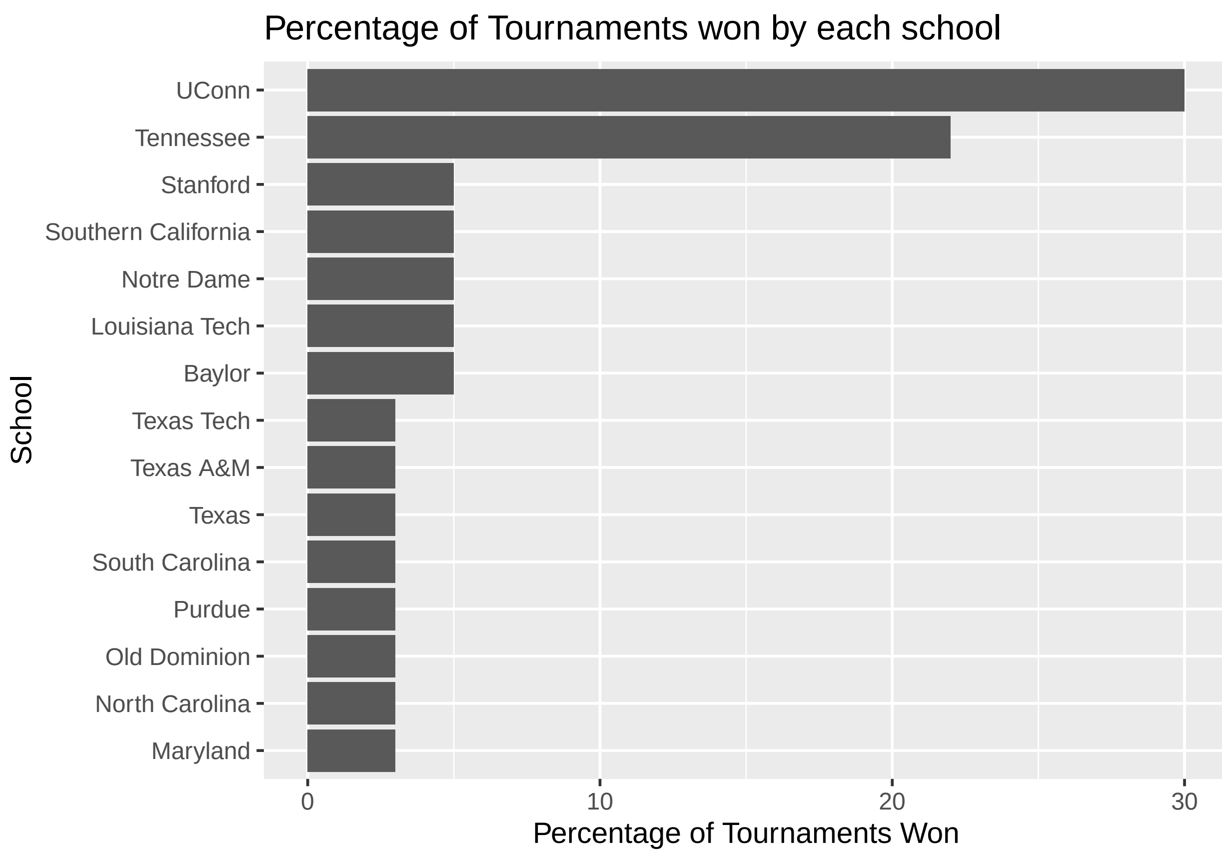
```
# Glimpse the data
glimpse(wncaa)
```

```
## Rows: 2,092
## Columns: 19
## $ year              <dbl> 1982, 1982, 1982, 1982, 1982, 1982, 1982, 1982, 1982~
## $ school            <chr> "Arizona St.", "Auburn", "Cheyney", "Clemson", "Drak~
## $ seed              <dbl> 4, 7, 2, 5, 4, 6, 5, 8, 7, 7, 4, 8, 2, 1, 1, 2, 3, 6~
## $ conference        <chr> "Western Collegiate", "Southeastern", "Independent",~
## $ conf_w            <dbl> NA, NA, NA, 6, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ conf_l            <dbl> NA, NA, NA, 3, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ conf_percent      <dbl> NA, NA, NA, 66.7, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ conf_place        <chr> "-", "-", "-", "4th", "-", "-", "-", "-", "-", "-", ~
## $ reg_w             <dbl> 23, 24, 24, 20, 26, 19, 21, 14, 21, 28, 24, 17, 22, ~
## $ reg_l             <dbl> 6, 4, 2, 11, 6, 7, 8, 10, 8, 7, 5, 13, 7, 5, 1, 6, 4~
## $ reg_percent       <dbl> 79.3, 85.7, 92.3, 64.5, 81.3, 73.1, 72.4, 58.3, 72.4~
## $ how_qual          <chr> "at-large", "at-large", "at-large", "at-large", "aut~
## $ x1st_game_at_home <chr> "Y", "N", "Y", "N", "Y", "N", "N", "N", "N", "N", "Y~
## $ tourney_w         <dbl> 1, 0, 4, 0, 2, 0, 0, 0, 0, 0, 2, 0, 2, 1, 5, 3, 1, 1~
## $ tourney_l         <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1~
## $ tourney_finish    <chr> "RSF", "1st", "N2nd", "1st", "RF", "1st", "1st", "1s~
## $ full_w            <dbl> 24, 24, 28, 20, 28, 19, 21, 14, 21, 28, 26, 17, 24, ~
## $ full_l            <dbl> 7, 5, 3, 12, 7, 8, 9, 11, 9, 8, 6, 14, 8, 6, 1, 7, 5~
## $ full_percent      <dbl> 77.4, 82.8, 90.3, 62.5, 80.0, 70.4, 70.0, 56.0, 70.0~
```

Question 1

```
# create percentages of tournaments one by school
champ_by_school <- wncaa %>%
  filter(tourney_finish == "Champ") %>%
  group_by(school) %>%
```

```r
  summarize(N = n()) %>%
  mutate(freq = N/sum(N), pct = round((freq*100),0)) %>%
  arrange(desc(pct))


# now let's make a bar plot
cs <- ggplot(champ_by_school, aes(x = reorder(school, pct), y = pct))
cs + geom_col() + coord_flip() +
  labs(x = "School", y = "Percentage of Tournaments Won",
       title = "Percentage of Tournaments won by each school")
```



The *first* thing I notice is that most of the Texas schools won small amounts of the tournaments they were in (~2.5% - lame). I also wonder why most teams seem to fall into either 5% or ~2.5% of wins, why is there so low variation?

Tennessee and UCon have won the most.

Question 2

First, lets create a dataset that includes just the top teams

```r
# Get the names of each of the schools through using the champ dataset
champ_names <- unique(champ_by_school$school)

champ_names
```

```
##  [1] "UConn"               "Tennessee"           "Baylor"
##  [4] "Louisiana Tech"      "Notre Dame"          "Southern California"
##  [7] "Stanford"            "Maryland"            "North Carolina"
## [10] "Old Dominion"        "Purdue"              "South Carolina"
## [13] "Texas"               "Texas A&M"           "Texas Tech"
```

```
# now lets use the champ names to get the school champs from the orig. data set

# we're going to group by school for the next step
winners <- wncaa %>%
  filter(school %in% champ_names) %>%
  mutate(seed2 = as.factor(seed)) #create character value of seed for fill

# I noticed later on we are called to use as.factor,
# let's see what this is about.
?as.factor
```

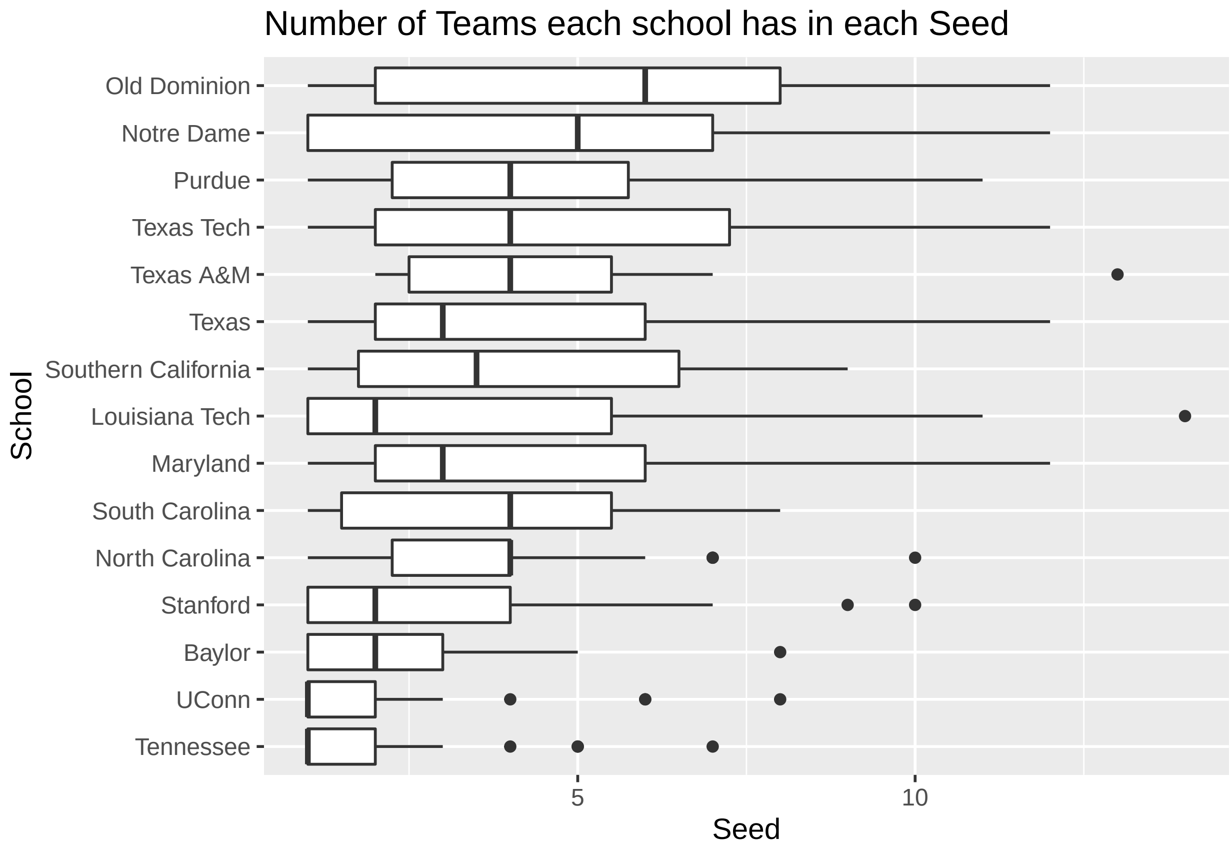
```
## Help on topic 'as.factor' was found in the following packages:
##
##   Package               Library
##   base                  /Library/Frameworks/R.framework/Resources/library
##   generics              /Library/Frameworks/R.framework/Versions/4.2/Resources/library
##
##
## Using the first match ...
```

```
winners
```

```
## # A tibble: 368 x 20
##     year school   seed confe~1 conf_w conf_l conf_~2 conf_~3 reg_w reg_l reg_p~4
##    <dbl> <chr>   <dbl> <chr>    <dbl>  <dbl>   <dbl> <chr>   <dbl> <dbl>   <dbl>
##  1  1982 Louisi~     1 Indepe~     NA     NA    NA   -          30     1    96.8
##  2  1982 Maryla~     2 Atlant~      6      1    85.7 1st        22     6    78.6
##  3  1982 Old Do~     1 Indepe~     NA     NA    NA   -          21     5    80.8
##  4  1982 South ~     3 Indepe~     NA     NA    NA   -          21     7    75
##  5  1982 Southe~     1 Wester~     NA     NA    NA   -          20     3    87
##  6  1982 Stanfo~     7 Northe~      9      3    75   2nd        19     7    73.1
##  7  1982 Tennes~     2 Southe~     NA     NA    NA   -          19     9    67.9
##  8  1983 Louisi~     1 Indepe~     NA     NA    NA   -          27     1    96.4
##  9  1983 Maryla~     3 Atlant~     10      3    76.9 T2nd       25     4    86.2
## 10  1983 North ~     7 Atlant~     10      3    76.9 T2nd       22     7    75.9
## # ... with 358 more rows, 9 more variables: how_qual <chr>,
## #   x1st_game_at_home <chr>, tourney_w <dbl>, tourney_l <dbl>,
## #   tourney_finish <chr>, full_w <dbl>, full_l <dbl>, full_percent <dbl>,
## #   seed2 <fct>, and abbreviated variable names 1: conference, 2: conf_percent,
## #   3: conf_place, 4: reg_percent
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Next, lets make a plot that shows the distribution of `seeds` for each school.

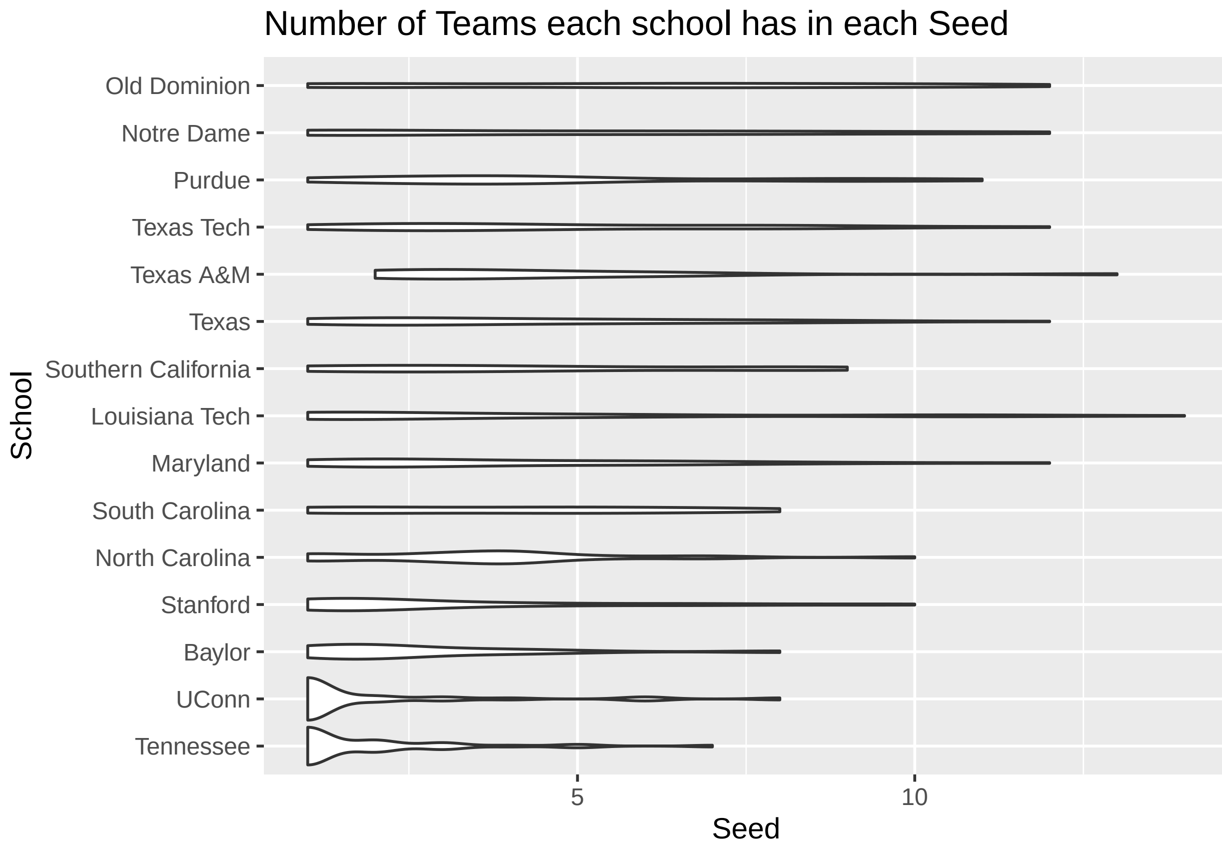
```
w <- ggplot(winners, aes(x = reorder(school,seed), y = seed))
w + geom_boxplot() + coord_flip() +
  labs( x = "School",
       y = "Seed", title = "Number of Teams each school has in each Seed")
```



Tennessee and UConn, the schools with the highest tournament wins, have the most seed one teams. **Any surprises?** I am surprised that Maryland seems to have so many high seed teams (closer to 1) but still is one of the lowest teams for percent of tournaments won.

Now lets make the same plot using `geom_violin`.

```
w2 <- ggplot(winners, aes(x = reorder(school,seed), y = seed))
w2 + geom_violin() + coord_flip() +
  labs(title = "Number of Teams each school has in each Seed", x = "School",
       y = "Seed")
```

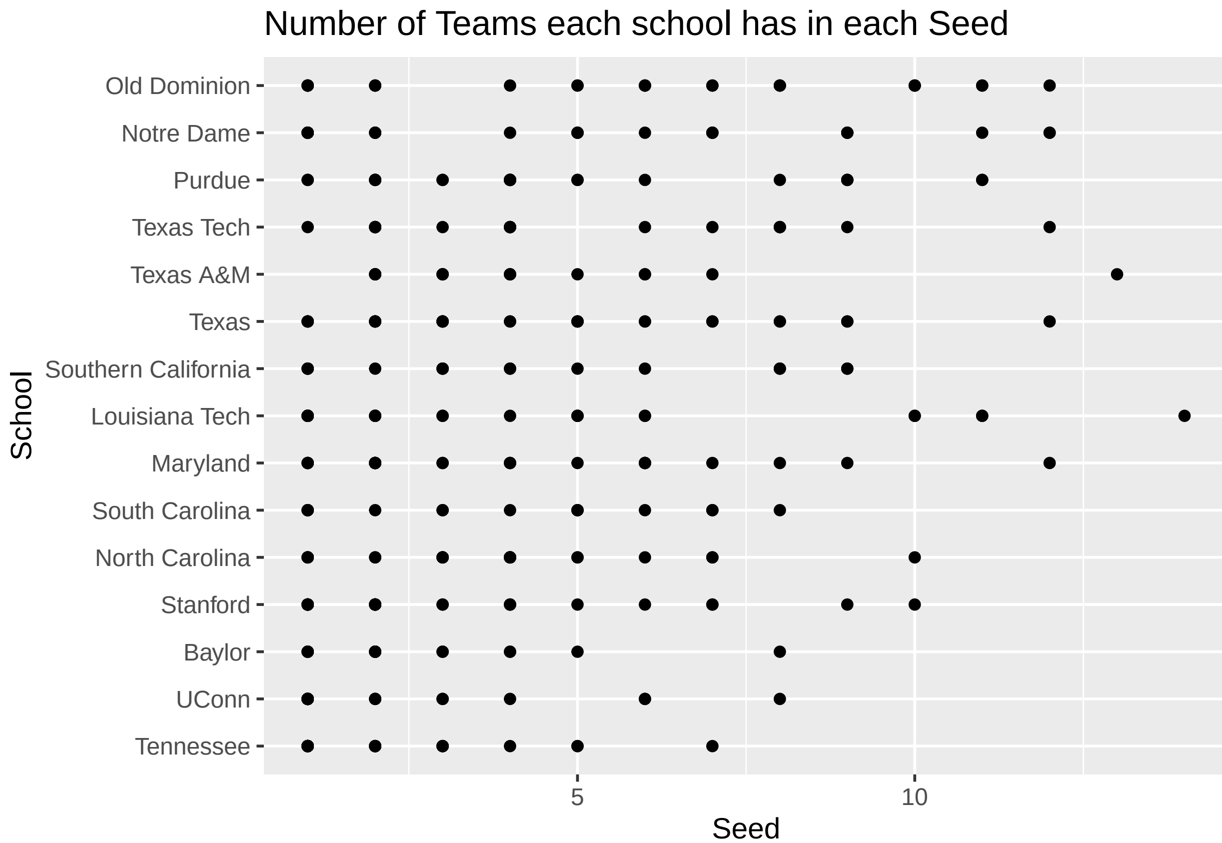


I find this graph way easier to read, because I can tell where each school has large amounts of seed one, two, etc. teams. However, I think the other graph gives you more precise information on the amount of teams each school has within each seed.

Question 3

Now, let's try visualizing the data with a scatterplot.

```
w2 + geom_point() + coord_flip() +
  labs(x = "School", y = "Seed",
       title = "Number of Teams each school has in each Seed")
```



As you can see, this doesn't work very well because the values available for "seed" are discrete and thus the options for each team are stacked on top of each other. The most we can tell from this is something like "Old Dominion and Notre Dame don't have any seed 3 teams." We can't see how many teams are within each seed for each school .

Question 4

Now, lets try the `summarize_if` verb. We're going to use the `winners` dataset.

```
# lets summarize values if they are numeric
# and take out NA values for each school

school_m_sd <- winners %>%
  group_by(school) %>%
  mutate(year = as.factor(year)) %>%
  summarize_if(is.numeric, funs(mean, sd), na.rm = TRUE) %>%
  select(school, reg_percent_mean, reg_percent_sd)
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
```

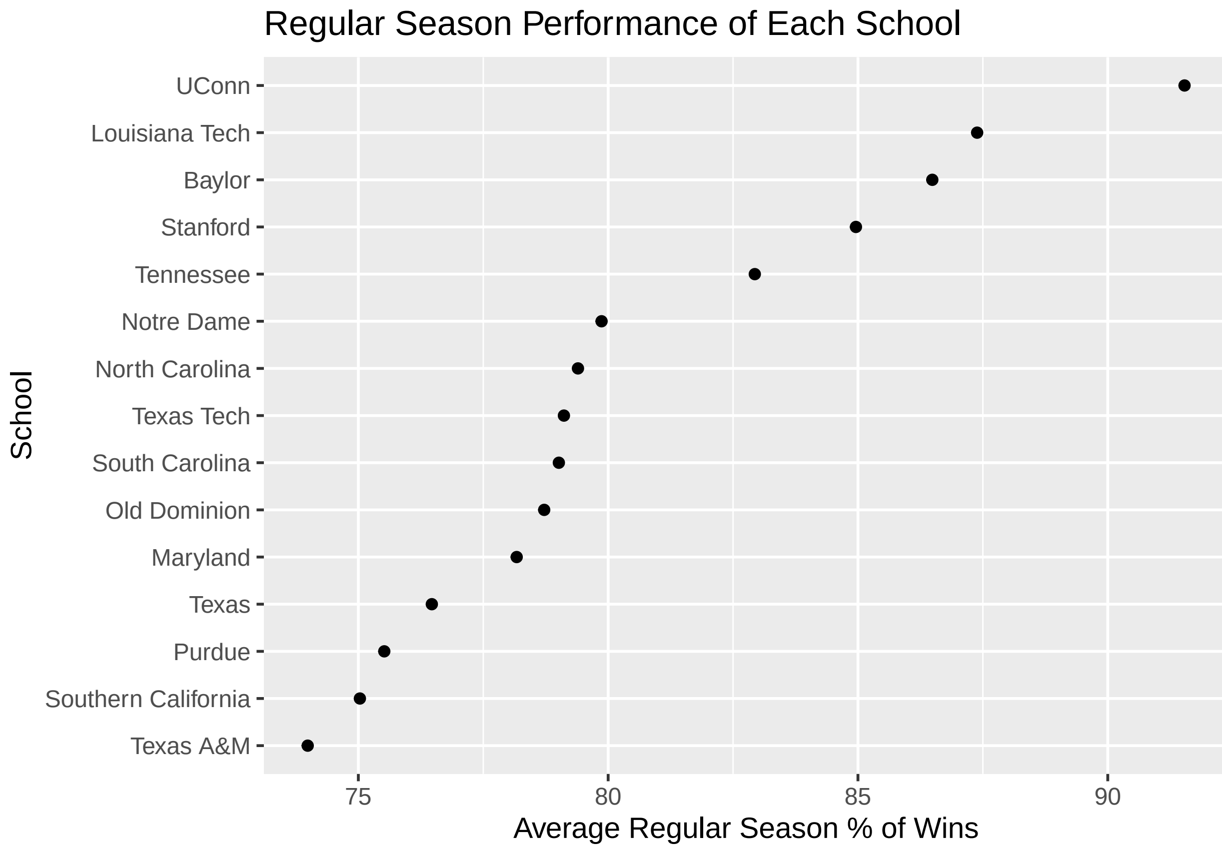
```
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
# lets explore average win percentages and standard deviations.
school_m_sd
```

```
## # A tibble: 15 x 3
##    school              reg_percent_mean reg_percent_sd
##    <chr>                          <dbl>          <dbl>
##  1 Baylor                          86.5           9.12
##  2 Louisiana Tech                  87.4           9.41
##  3 Maryland                        78.2          11.6
##  4 North Carolina                  79.4           9.58
##  5 Notre Dame                      79.9          13.4
##  6 Old Dominion                    78.7          10.2
##  7 Purdue                          75.5          10.6
##  8 South Carolina                  79.0           8.89
##  9 Southern California             75.0          10.4
## 10 Stanford                        85.0           9.88
## 11 Tennessee                       82.9          10.3
## 12 Texas                           76.5          12.4
## 13 Texas A&M                       74.0           5.44
## 14 Texas Tech                      79.1           8.93
## 15 UConn                           91.5           9.35
```

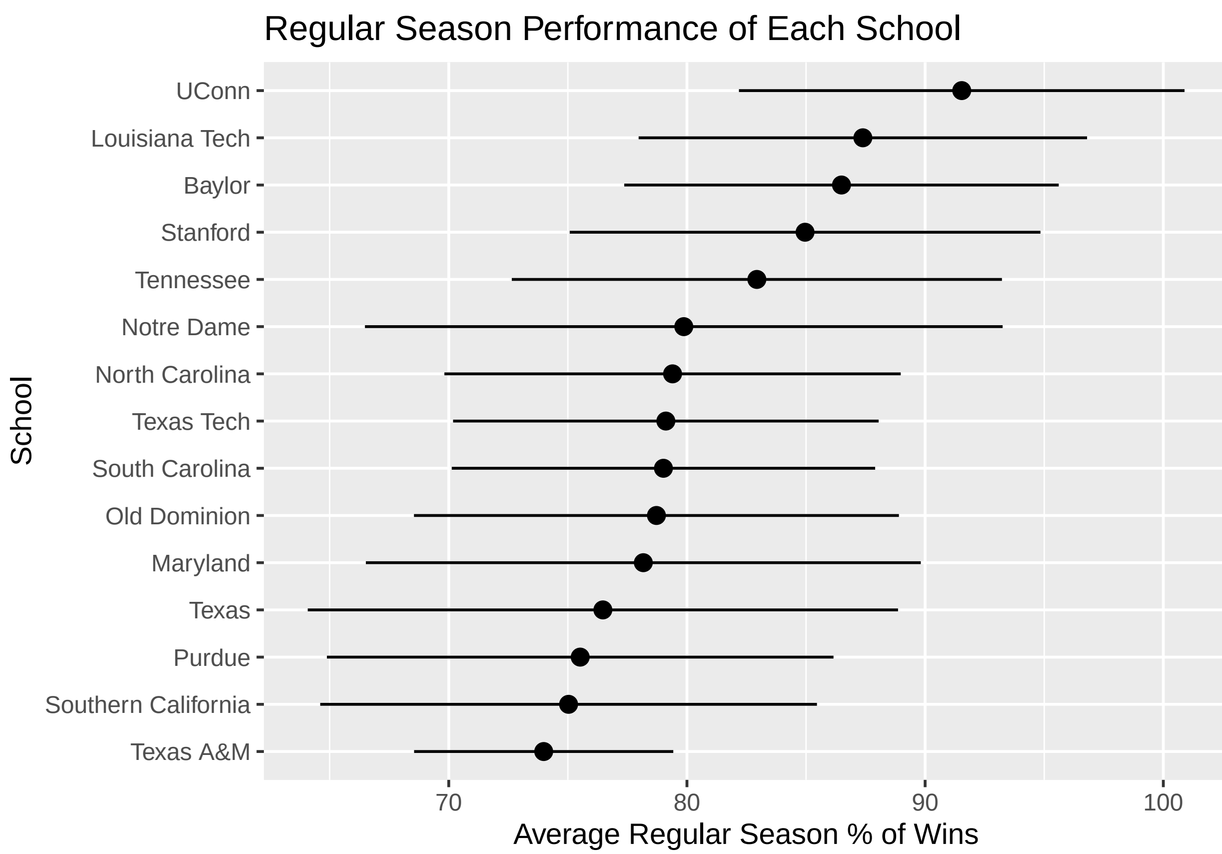
Perfect, now let's make a dot plot.

```
win_percent_plot <- ggplot(school_m_sd, aes(x = reorder(school,
                                                        reg_percent_mean),
                                            y = reg_percent_mean))
win_percent_plot + geom_point() + coord_flip() +
  labs(x = "School", y = "Average Regular Season % of Wins",
       title = "Regular Season Performance of Each School")
```



UConn and Lousiana Tech have the highest percent of regular season wins. Southern California and Texas A&M have the lowest percent of regular season wins. All teams had over 60% of wins in their regular season.

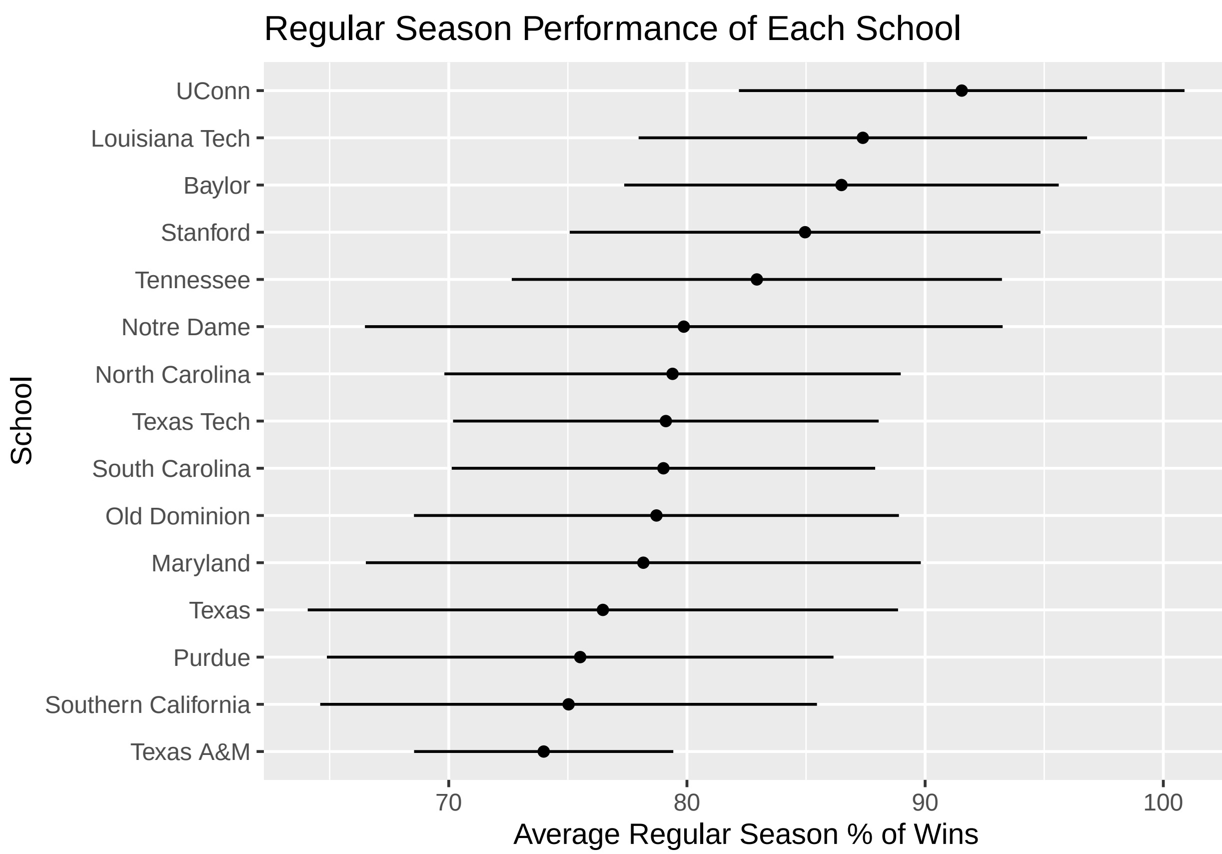
```
win_percent_plot +
  geom_pointrange(aes(ymin = reg_percent_mean - reg_percent_sd,
                      ymax = reg_percent_mean + reg_percent_sd)) +
   coord_flip() + labs(x = "School", y = "Average Regular Season % of Wins",
                       title = "Regular Season Performance of Each School")
```



What school has the most narrow interval Texas A&M has the most narrow interval.

Now, let's try to make a plot using `geom_linerage`.

```
win_percent_plot2 <- ggplot(school_m_sd,
                            aes(x = reorder(school, reg_percent_mean), y =
                                  reg_percent_mean)) + geom_point()

win_percent_plot2 +
  geom_linerange(aes(ymin = reg_percent_mean - reg_percent_sd,
                     ymax = reg_percent_mean + reg_percent_sd)) +
   coord_flip() + labs(x = "School", y = "Average Regular Season % of Wins",
                      title = "Regular Season Performance of Each School")
```

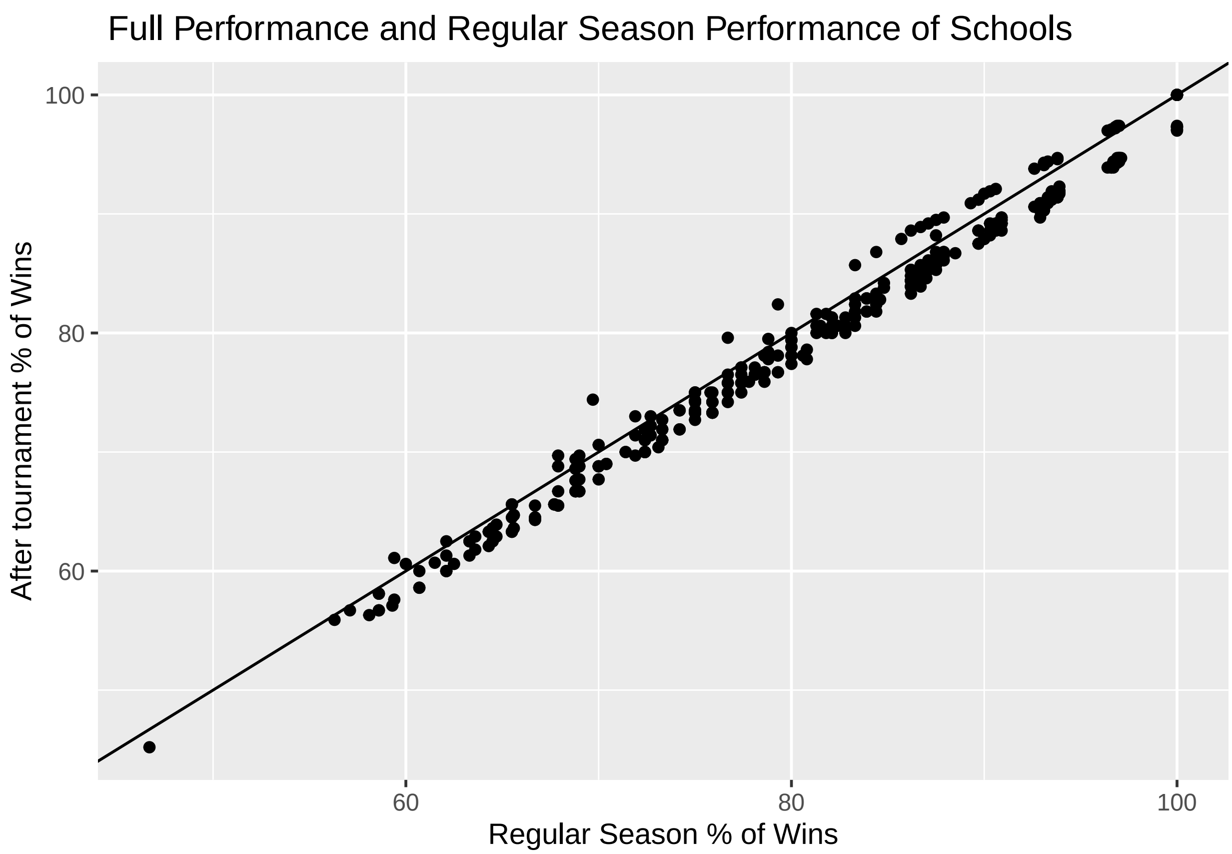


Can you produce the same graph?
Yes! You just combine `geom_point` and `geom_linerange`.

Question 5

Now lets explore how regular season performance is related to full performance.

```
ggplot(winners, aes(x = reg_percent, y = full_percent)) + geom_point() + geom_abline() +
labs(x = "Regular Season % of Wins", y = "After tournament % of Wins",
     title = " Full Performance and Regular Season Performance of Schools")
```

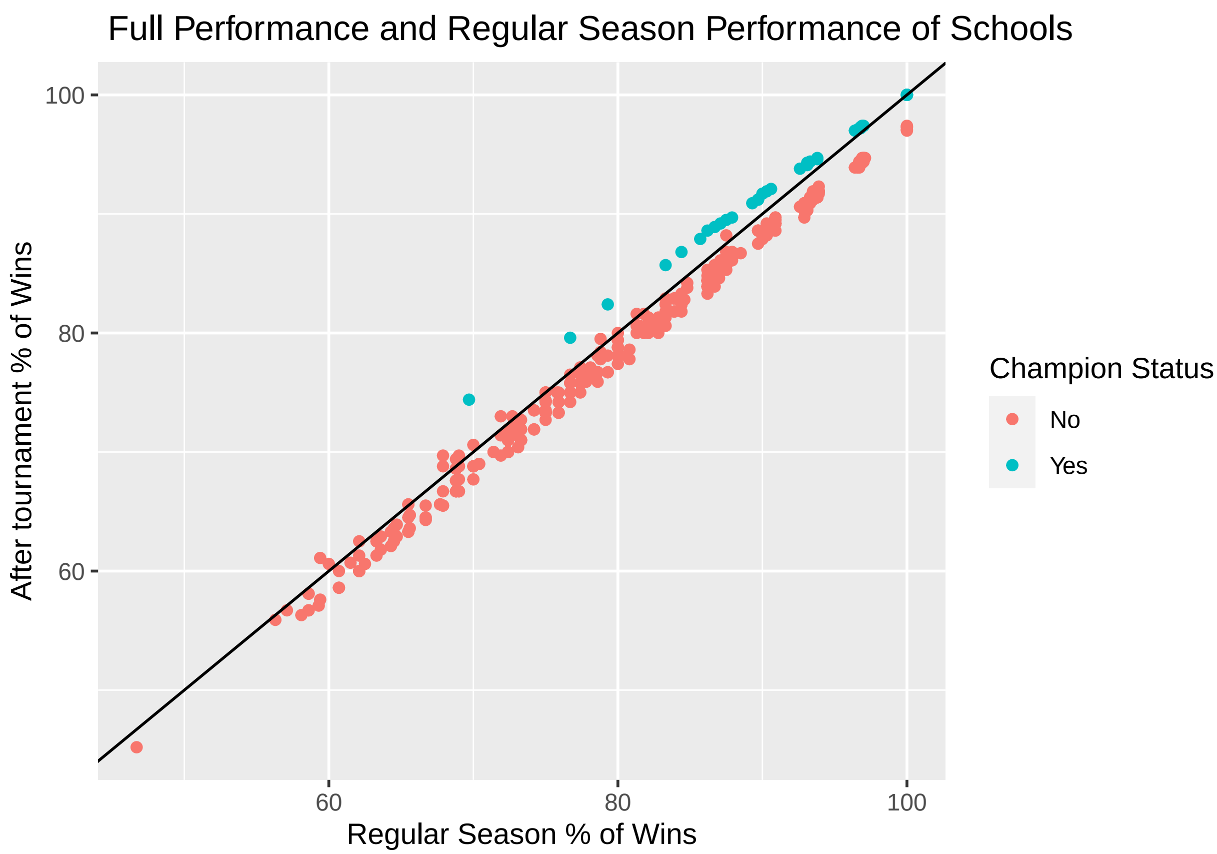


Most teams did not improve after the tournament compared to their regular season performance. However, quiet a few did.

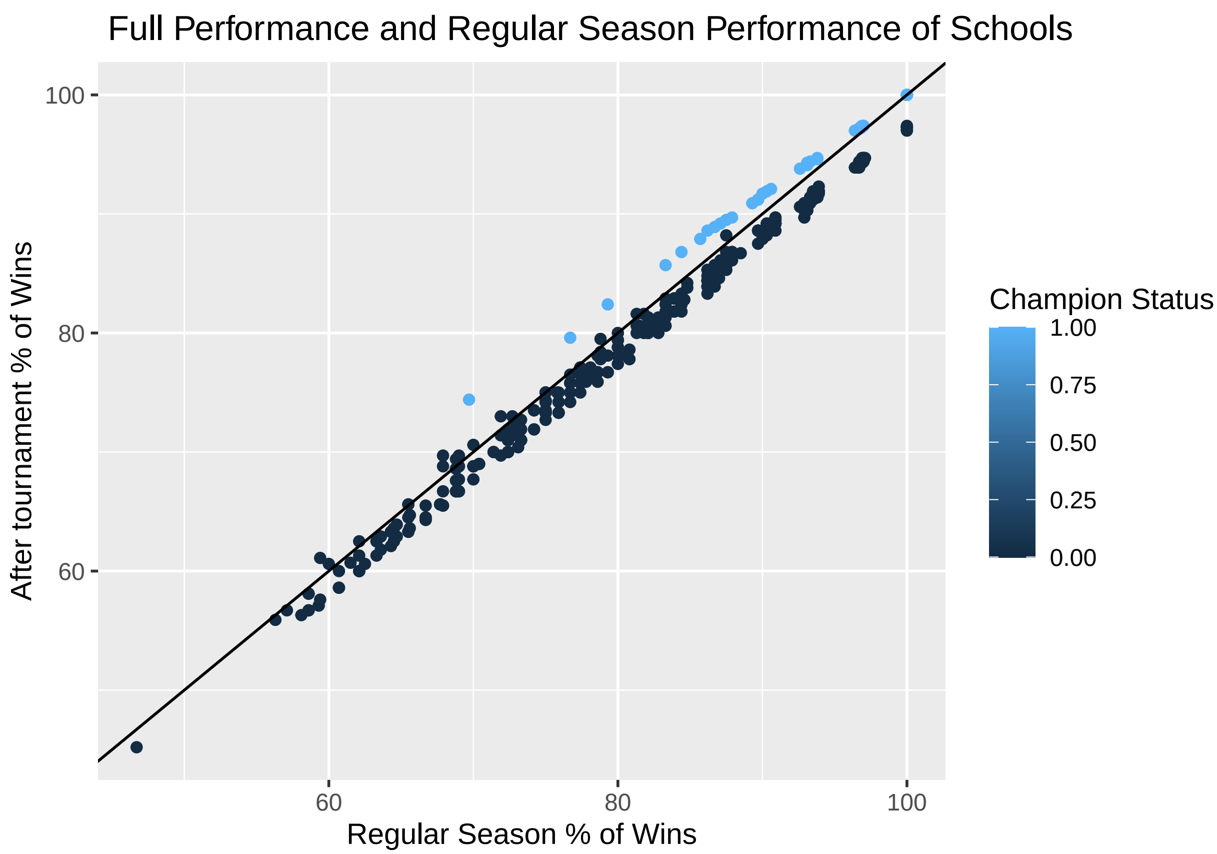
Additionally, the amount of teams who improved increases as we go up in their regular season performance. For example, there are fewer teams who improved in full performance whose regular season performance is 60% while there are more teams who's full performance improved from their 90% regular season performance.

Question 6

```r
# create a variable for champs
winners <- winners %>%
  mutate(is_champ = if_else(tourney_finish == "Champ", 1, 0),
         is_champ = as.factor(is_champ))


dubs <- ggplot(winners, aes(x = reg_percent, y = full_percent,
                            color = is_champ))
dubs + geom_point() + geom_abline() +
  labs(x = "Regular Season % of Wins",
      y = "After tournament % of Wins",
     title = " Full Performance and Regular Season Performance of Schools",
     col = "Champion Status") +
  scale_colour_discrete(labels = c("No", "Yes"))
```




```r
# lets see what happens if we use is_champ without as.factor
winners <- winners %>%
  mutate(is_champ2 = if_else(tourney_finish == "Champ", 1, 0))

ggplot(winners, aes(x = reg_percent,y = full_percent,
                    color = is_champ2)) +
  geom_point() + geom_abline() +
  labs(x = "Regular Season % of Wins", y = "After tournament % of Wins",
       title = " Full Performance and Regular Season Performance of Schools",
       col = "Champion Status")
```



Without `as.factor` the variable produces a scale from the numeric values from 0 to 1 instead of as discrete values of 0 and 1.

Do you see any patterns? Do they make sense to you?
Right away, I see a pattern of champions as being the ones who had improvement from their regular season performance to their full performance. This makes sense to me because these teams had improvement and thus were able to come out on top.

Question 7

```
winners2 <- winners %>%
  mutate(plot_label = paste(school, year, sep = " - ")) %>%
  mutate(difference = full_percent - reg_percent)

# now let's find these teams
winners2 %>%
  filter(reg_percent < 50 | reg_percent < 71 & full_percent > 71)
```

```
## # A tibble: 2 x 24
##    year school     seed confe~1 conf_w conf_l conf_~2 conf_~3 reg_w reg_l reg_p~4
##   <dbl> <chr>     <dbl> <chr>    <dbl>  <dbl>   <dbl> <chr>   <dbl> <dbl>   <dbl>
## 1  1992 Notre D~     12 Midwes~      8      4    66.7 2nd        14    16    46.7
## 2  1997 Tenness~      3 Southe~      8      4    66.7 5th        23    10    69.7
## # ... with 13 more variables: how_qual <chr>, x1st_game_at_home <chr>,
```

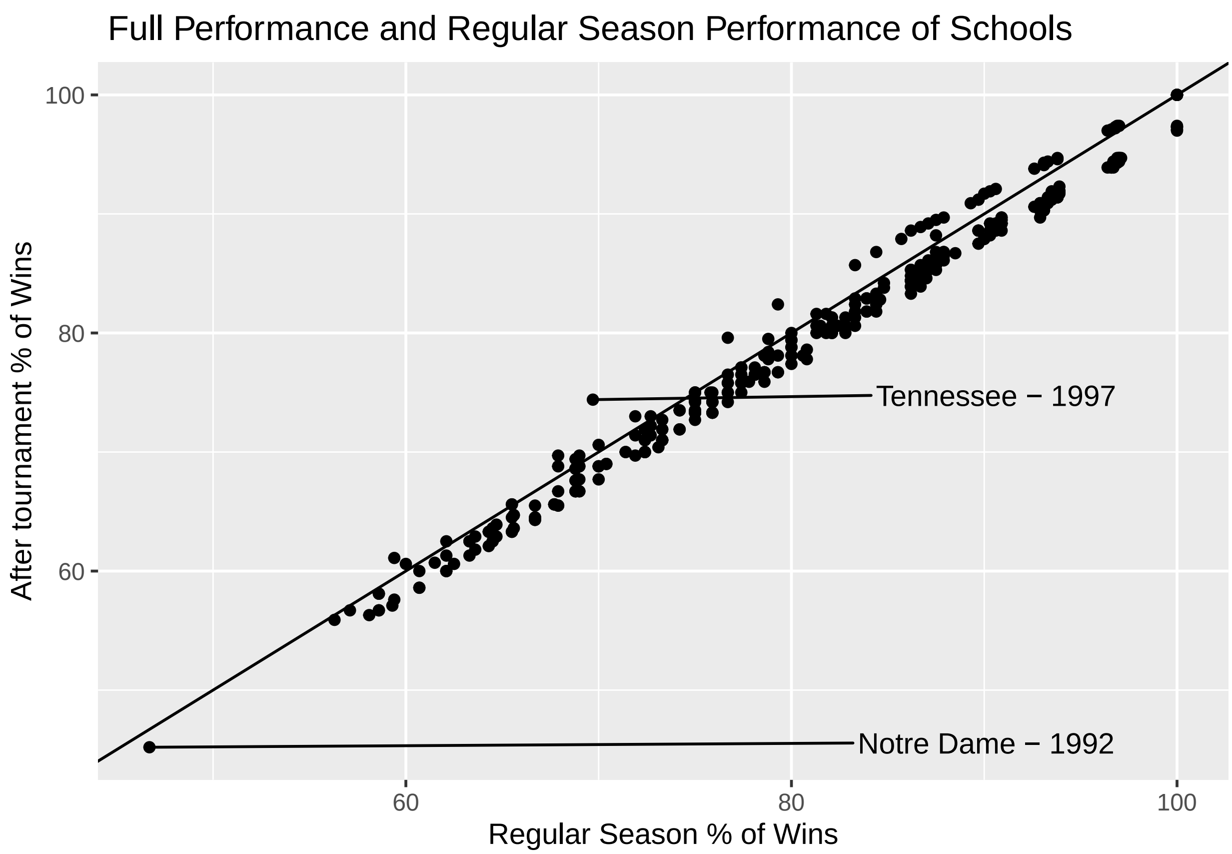
```
## #   tourney_w <dbl>, tourney_l <dbl>, tourney_finish <chr>, full_w <dbl>,
## #   full_l <dbl>, full_percent <dbl>, seed2 <fct>, is_champ <fct>,
## #   is_champ2 <dbl>, plot_label <chr>, difference <dbl>, and abbreviated
## #   variable names 1: conference, 2: conf_percent, 3: conf_place,
## #   4: reg_percent
## # i Use `colnames()` to see all variable names
```

Now lets create the plot with labels. First we need to install `ggrepel` so that it makes labelling our graph easier.

```
#first we need to install the ggrepel package
install.packages("ggrepel")
```

Next, let's laod the package and create our plot.

```
library(ggrepel)

champ_plot <- ggplot(winners2, aes(x = reg_percent,
                                   y = full_percent)) +
  geom_point() + geom_abline()


champ_plot +
  geom_text_repel(data = subset(winners2, reg_percent < 50 |
                                  reg_percent < 71 & full_percent > 71),
                  mapping = aes(label = plot_label),
                  hjust = -3.5, vjust = .4) +
  labs(x = "Regular Season % of Wins", y = "After tournament % of Wins",
     title = " Full Performance and Regular Season Performance of Schools")
```



Question 8

Lastly, let's find what teams have gone unbeaten (meaning they have 100% performance in the regular and full seasons).

```
winners %>%
  group_by(school) %>%
  filter(full_percent == 100 & reg_percent == 100)
```

```
## # A tibble: 8 x 22
## # Groups:   school [3]
##    year school  seed confere~1 conf_w conf_l conf_~2 conf_~3 reg_w reg_l reg_p~4
##   <dbl> <chr>  <dbl> <chr>      <dbl>  <dbl>   <dbl> <chr>   <dbl> <dbl>   <dbl>
## 1  1986 Texas      1 Southwest     16      0     100 1st        29     0     100
## 2  1995 UConn      1 Big East      18      0     100 1st        29     0     100
## 3  2002 UConn      1 Big East      16      0     100 1st        33     0     100
## 4  2009 UConn      1 Big East      16      0     100 1st        33     0     100
## 5  2010 UConn      1 Big East      16      0     100 1st        33     0     100
## 6  2012 Baylor     1 Big 12        18      0     100 1st        34     0     100
## 7  2014 UConn      1 American~     18      0     100 1st        34     0     100
## 8  2016 UConn      1 American~     18      0     100 1st        32     0     100
## # ... with 11 more variables: how_qual <chr>, x1st_game_at_home <chr>,
## #   tourney_w <dbl>, tourney_l <dbl>, tourney_finish <chr>, full_w <dbl>,
## #   full_l <dbl>, full_percent <dbl>, seed2 <fct>, is_champ <fct>,
## #   is_champ2 <dbl>, and abbreviated variable names 1: conference,
```

```
## #   2: conf_percent, 3: conf_place, 4: reg_percent
## # i Use ‘colnames()‘ to see all variable names
```

The teams that have gone unbeaten are: Texas (hook ’em), UConn, and Baylor.

Any patterns? Surprises?
I’m surprised Tennessee isn’t listed when they have a high number of seed 1 teams and high percentage of wins. I think a pattern I notice though is that UConn has a high number of years where they went unbeaten which makes sense since they have a high number of seed 1 teams and won a large percentage of games.