Homework DataVisualization 3 4 5

Emilio Horner

2022-09-16

R Markdown

```
devtools::install_github("kjhealy/socviz")
```

```
## Skipping install of 'socviz' from a github remote, the SHA1 (eca80210) has not changed since last ins
##   Use `force = TRUE` to force installation
```

```
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 1.4.1
## 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()
```

```
# Read in the data
exercise_data <- read_csv("https://raw.githubusercontent.com/NicolasRestrep/223_course/main/Data/visuali
```

```
## 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`
```

```
glimpse(exercise_data)
```

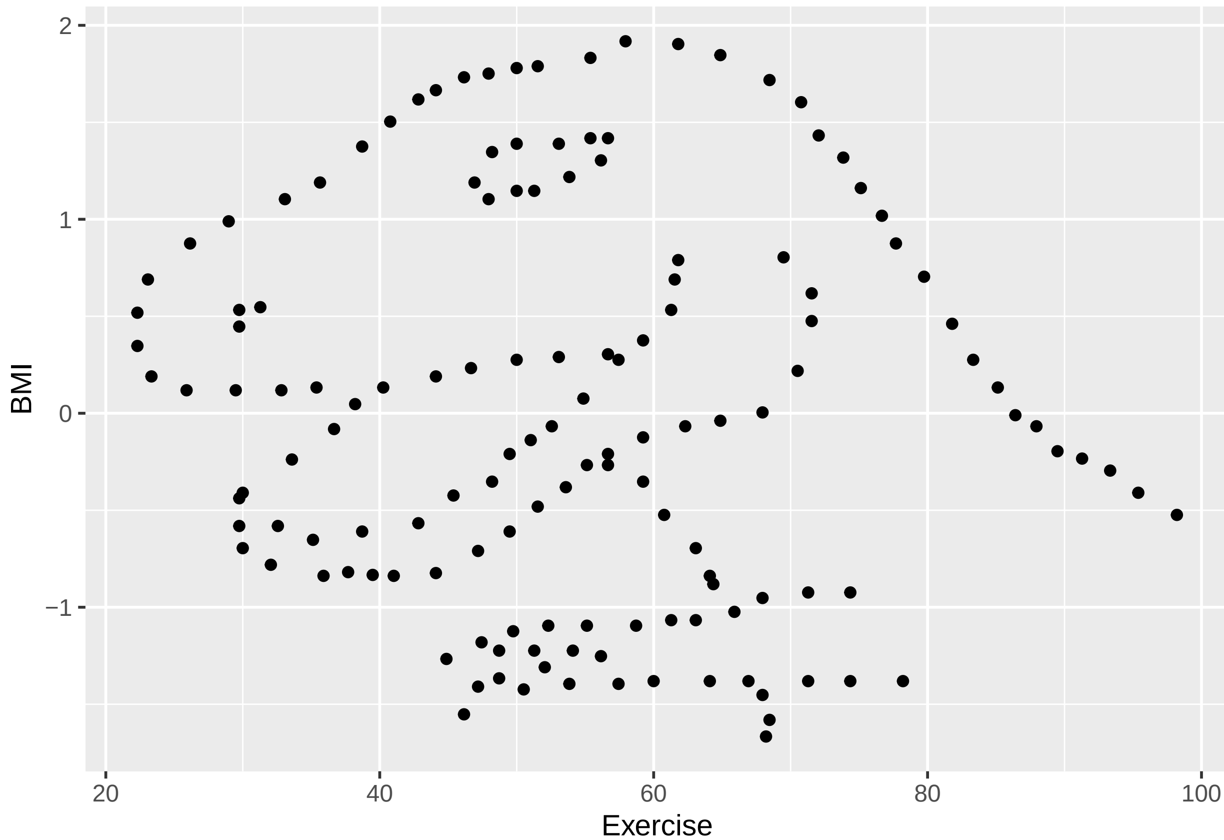
```
## Rows: 142
## Columns: 4
## $ ...1     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
## $ ...2     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
## $ Exercise <dbl> 55.3846, 51.5385, 46.1538, 42.8205, 40.7692, 38.7179, 35.6410~
## $ BMI      <dbl> 1.8320590,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

I would expect that people that exercise more have a lower BMI (though it depends on the type of exercise).

```
cor(exercise_data$Exercise, exercise_data$BMI)
```

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

```
ggplot(data = exercise_data, mapping = aes(x = Exercise, y = BMI)) +
  geom_point()
```



I see a dinosaur.

2.

```
library(causact)
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.~
```

```
corruptDF
```

```
## # A tibble: 174 x 7
##    country     region            countryCode regionCode popula~1 CPI2017 HDI2017
##    <chr>       <chr>             <chr>       <chr>         <int>   <int>   <dbl>
##  1 Afghanistan Asia Pacific      AFG         AP         35530081      15   0.498
##  2 Albania     East EU Cemt Asia ALB         ECA         2873457      38   0.785
##  3 Algeria     MENA              DZA         MENA       41318142      33   0.754
```

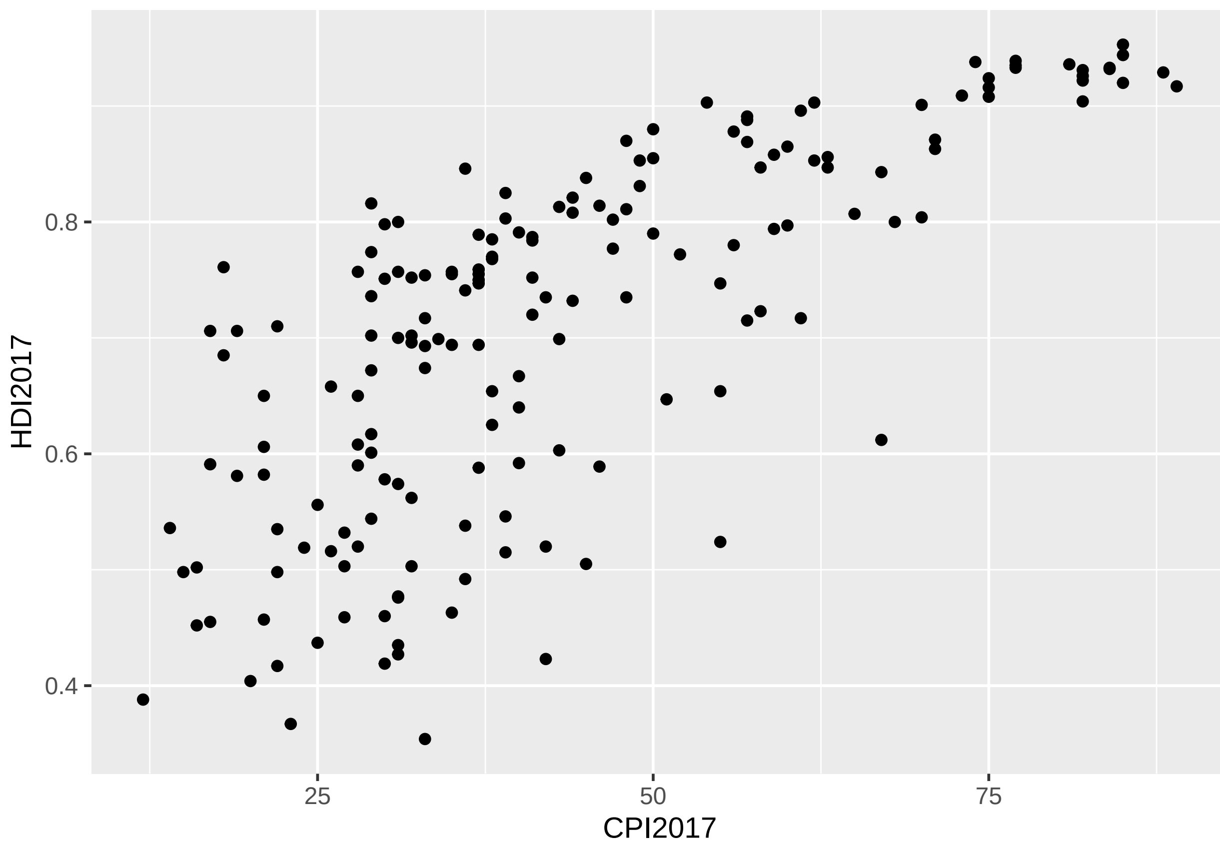
```
##   4 Angola      SSA               AGO         SSA        29784193      19    0.581
##   5 Argentina   Americas          ARG         AME        44271041      39    0.825
##   6 Armenia     East EU Cemt Asia ARM         ECA         2930450      35    0.755
##   7 Australia   Asia Pacific      AUS         AP         24598933      77    0.939
##   8 Austria     EU W. Europe      AUT         WE/EU       8809212      75    0.908
##   9 Azerbaijan  East EU Cemt Asia AZE         ECA         9862429      31    0.757
##  10 Bahamas     Americas          BHS         AME          395361      65    0.807
## # ... with 164 more rows, and abbreviated variable name 1: population
```

CPI2017 means the Consumer Price Index for 2017, essentially the average cost of a good

HDI2017 means the Human Development Index for 2017, which is a measurement of quality of life, life expectancy and access to knowledge

3.

```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017)) +
  geom_point()
```

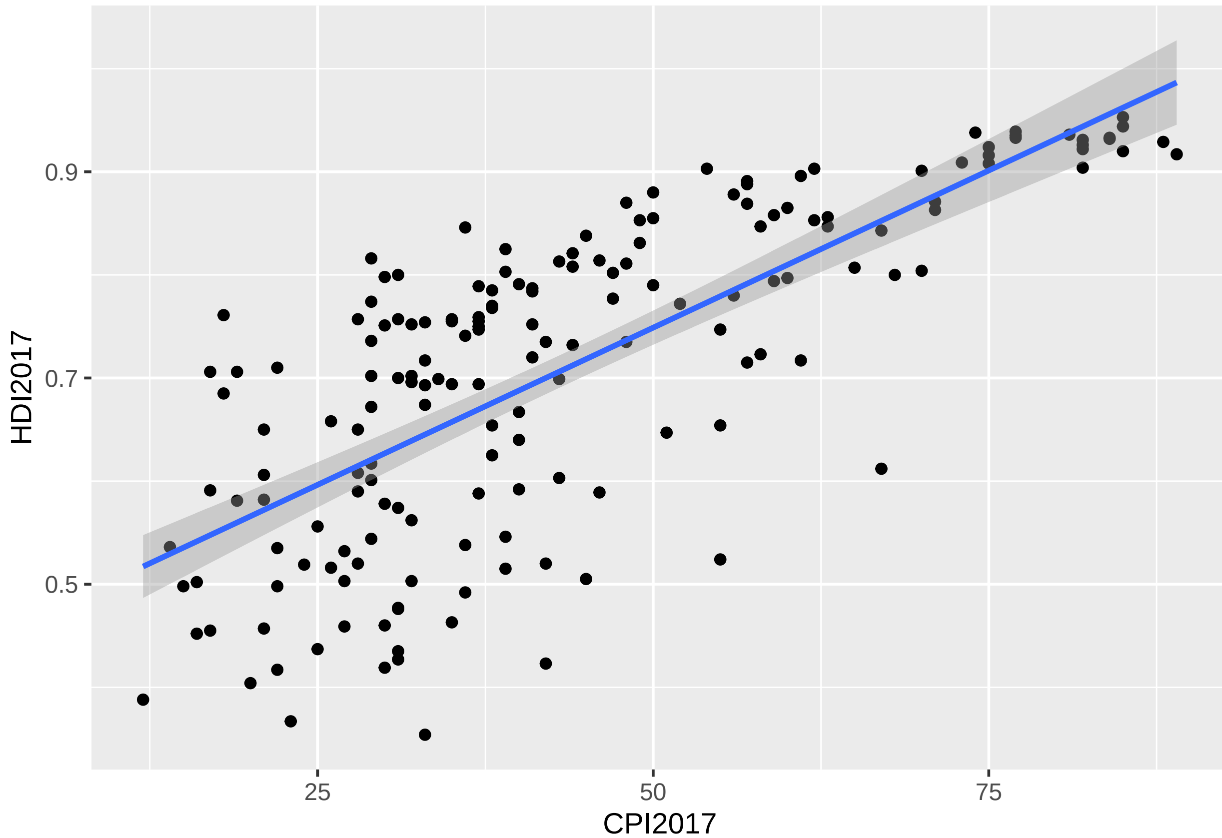


It appears the higher the CPI the higher the HDI. A positive sloping graph.

4.

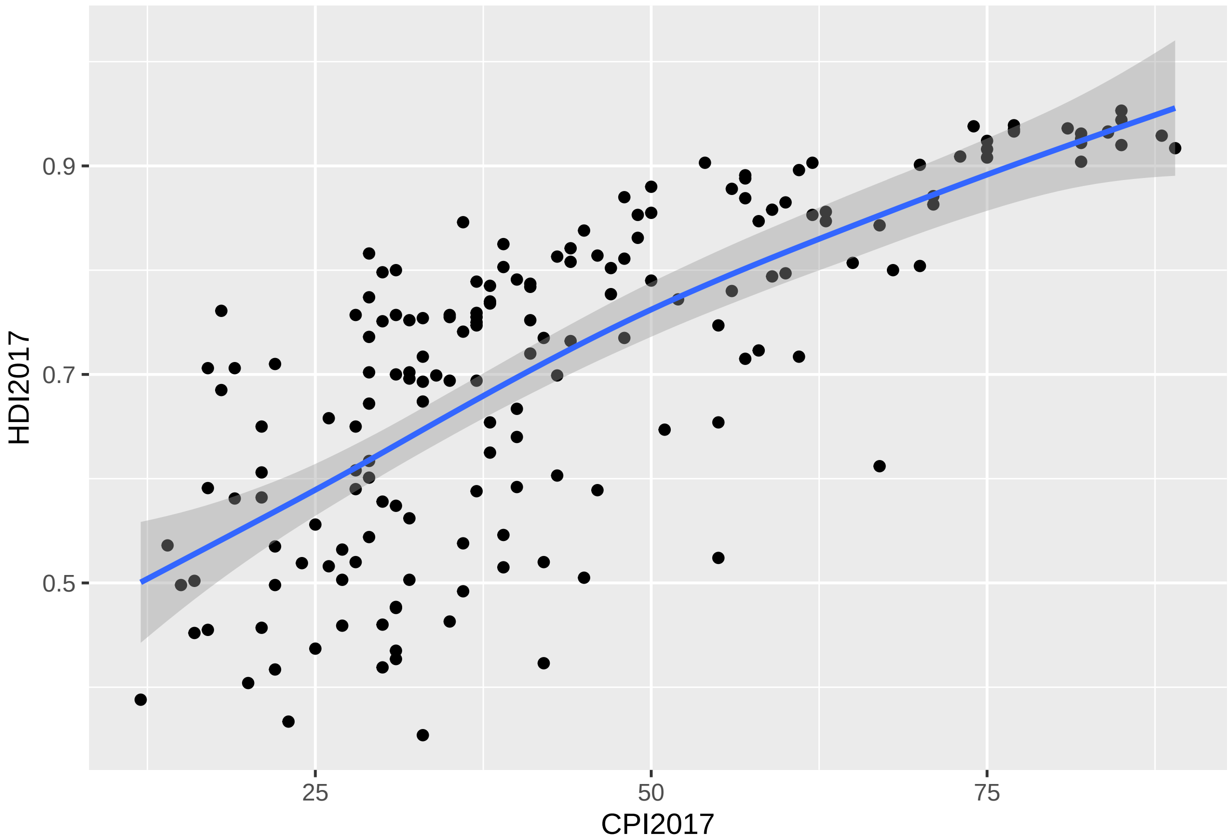
```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017)) +
  geom_point()+geom_smooth(method ="lm")
```

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




```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017)) +
  geom_point()+geom_smooth(method ="gam")
```

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

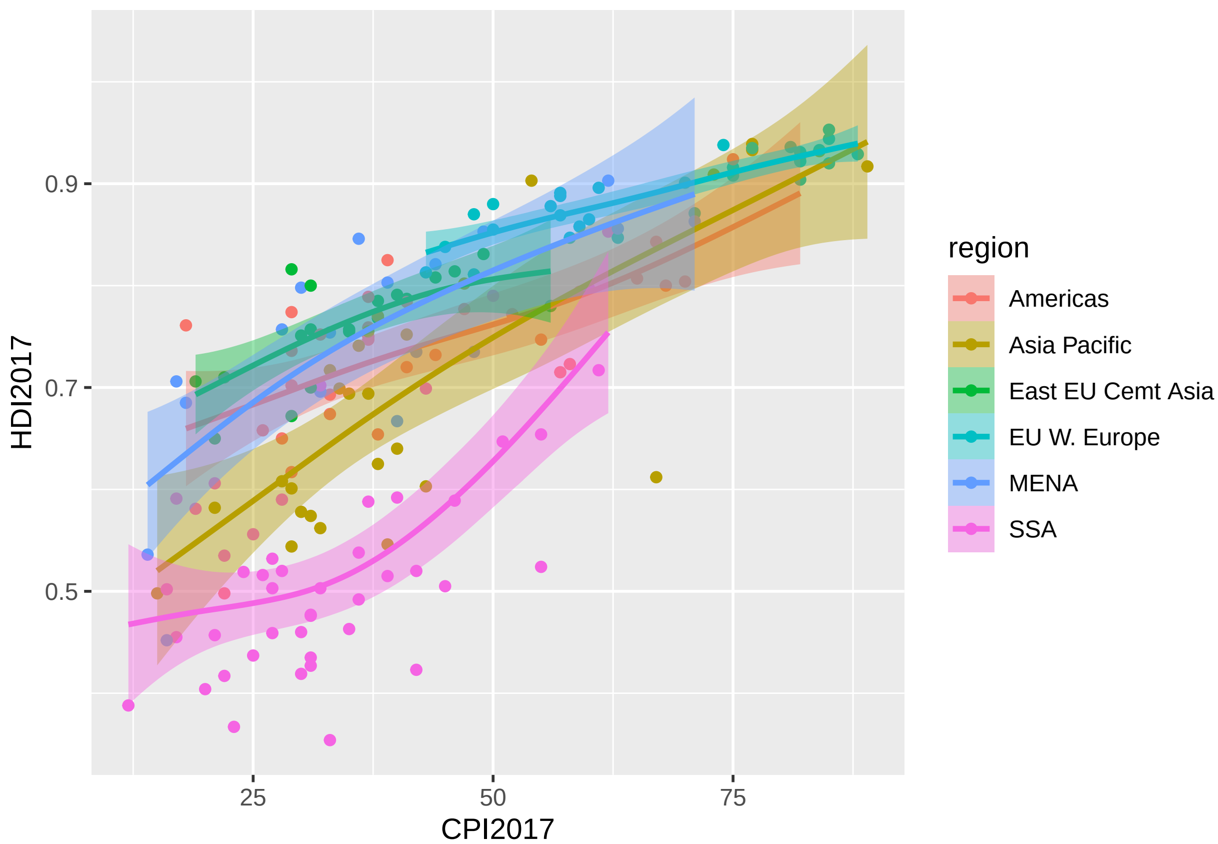


One is using the linear regression to create the line while other uses the generalized additive model. the lines are slightly different but both still show the postive sloping line. For this data I prefer the gam method because it appears that hte line more accurately represent the data.

5.

```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017, color = region, fill=region)) +
  geom_point()+geom_smooth(method ="gam")
```

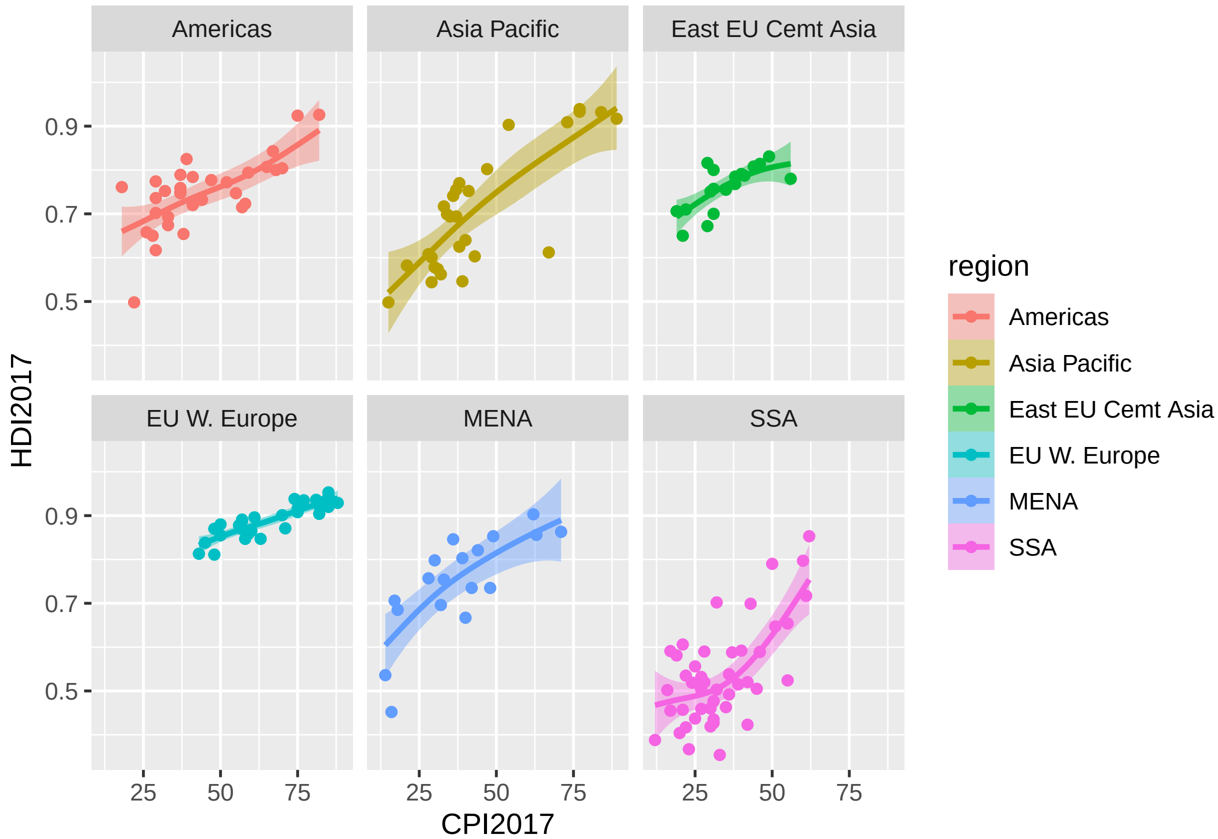
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



I think the lines are too cluttered. Another way would be to facetwrap the graph by region to create many different graphs.

```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017, color = region, fill=region)) +
  geom_point()+geom_smooth(method ="gam")+ facet_wrap(~region)
```

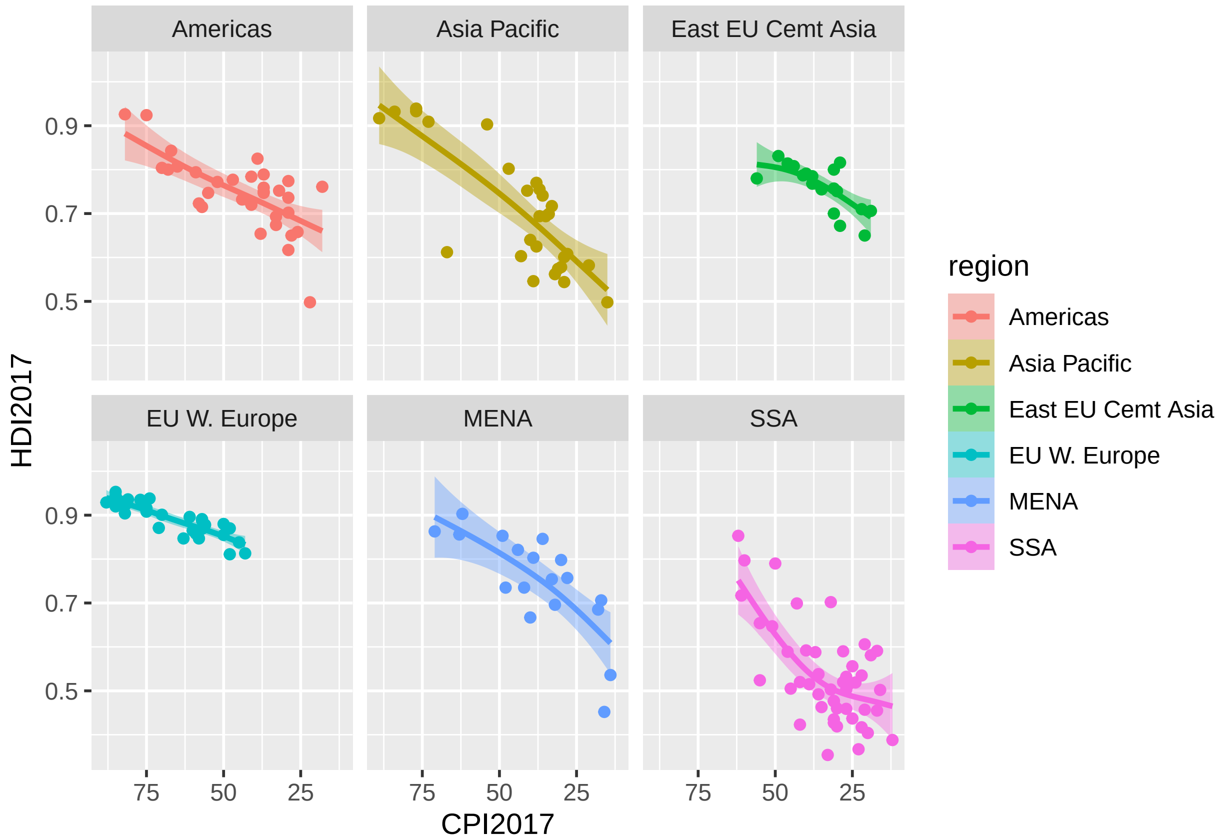
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



6.

```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017, color = region, fill=region)) +
  geom_point()+geom_smooth(method ="gam")+ facet_wrap(~region)+scale_x_reverse()
```

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

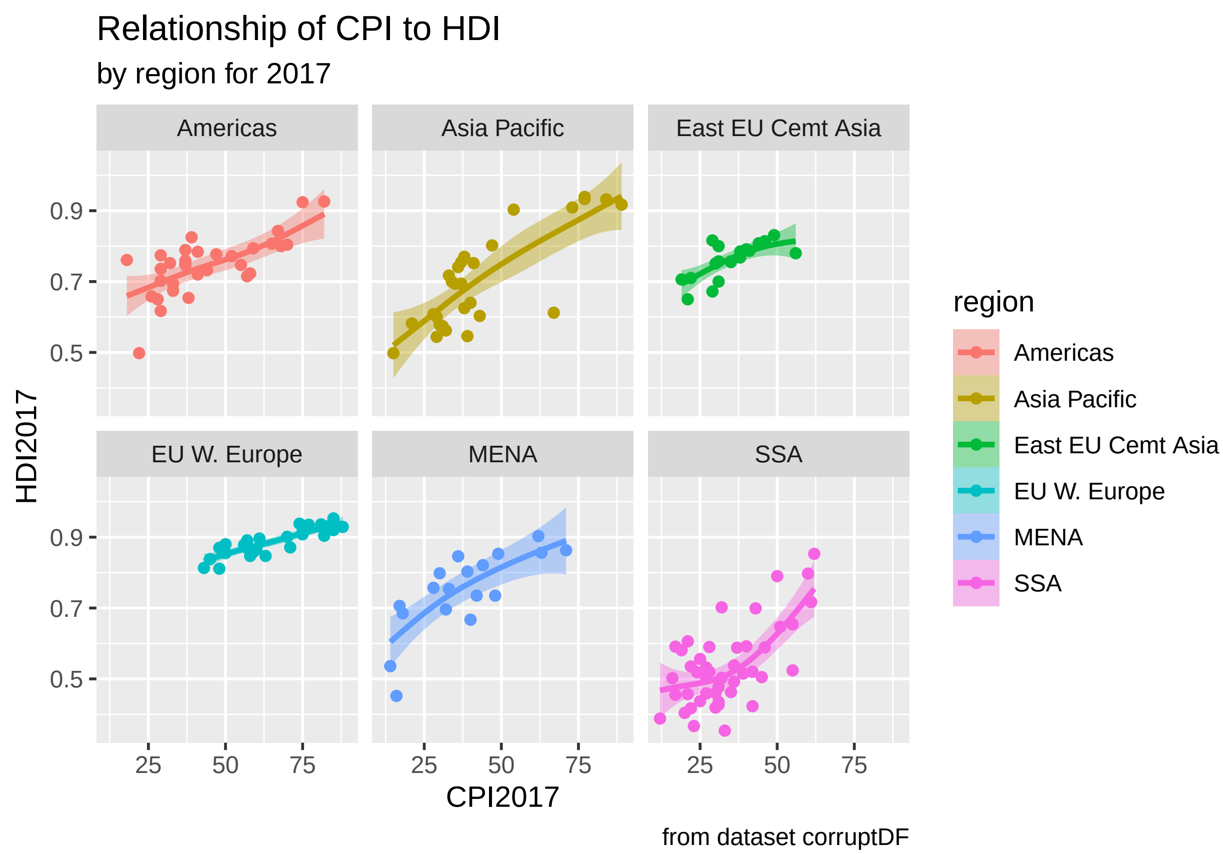


All the lines are downward sloping now

7.

```
ggplot(data = corruptDF , mapping = aes(x = CPI2017, y = HDI2017, color = region, fill=region)) +
  geom_point()+geom_smooth(method ="gam")+ facet_wrap(~region) + labs(title = "Relationship of CPI to HI
```

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



8.

```
ggsave(filename = "CPIgraph.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

Chapter 4.

1.

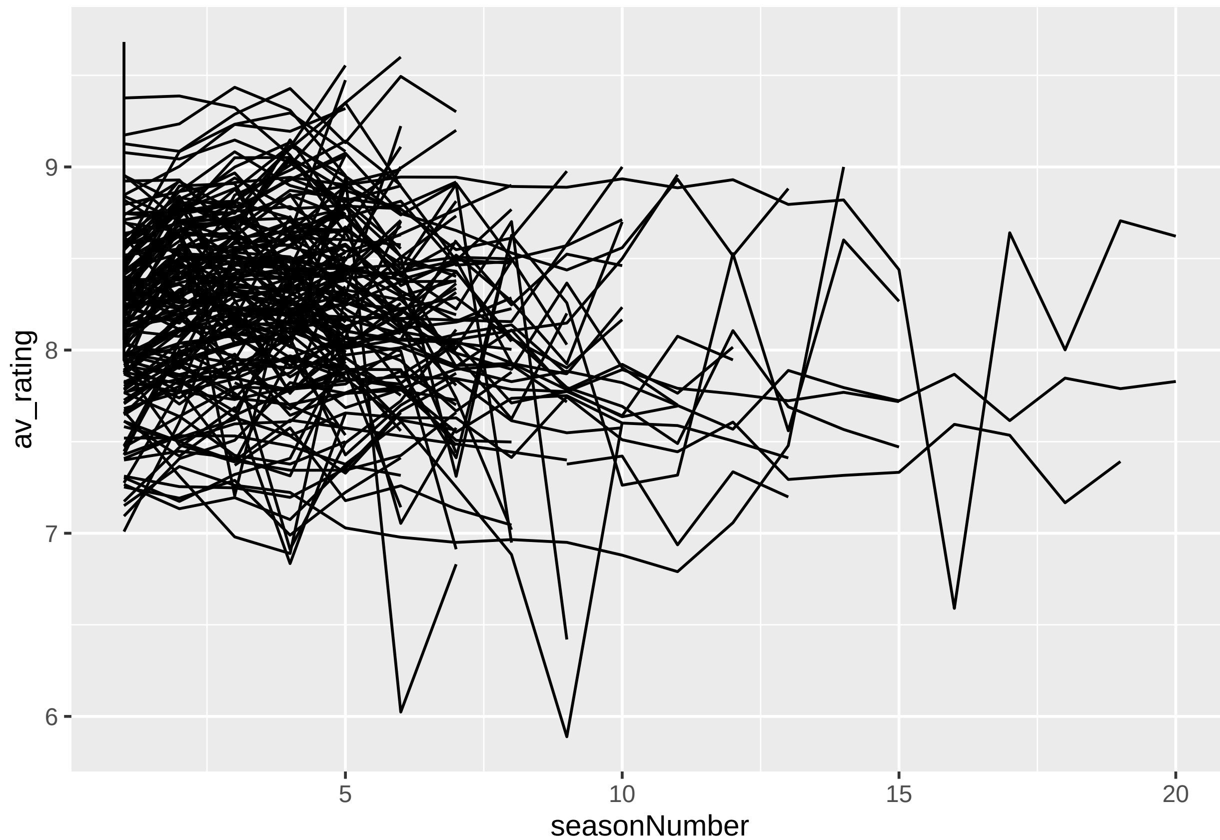
```
tv_ratings <- read_csv("https://raw.githubusercontent.com/NicolasRestrep/223_course/main/Data/tv_rating
```

```
## 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.
```

```
tv_long <- tv_ratings |>
  group_by(title) |>
  summarise(num_seasons = n()) |>
  ungroup() |>
  left_join(tv_ratings, by = "title")
```

```r
tv_long <- tv_long |>
  filter(num_seasons >= 5)
```

```r
ggplot(data=tv_long, mapping =aes(x= seasonNumber, y = av_rating))+geom_line(aes(group=title))
```



I don't think number of seasons determines how low or high the ratings will be.

2.

```r
tv2 <- tv_ratings |>
  group_by (title) |>
  mutate(num_seasons = max (seasonNumber)) |>
  filter (num_seasons >= 5)
```

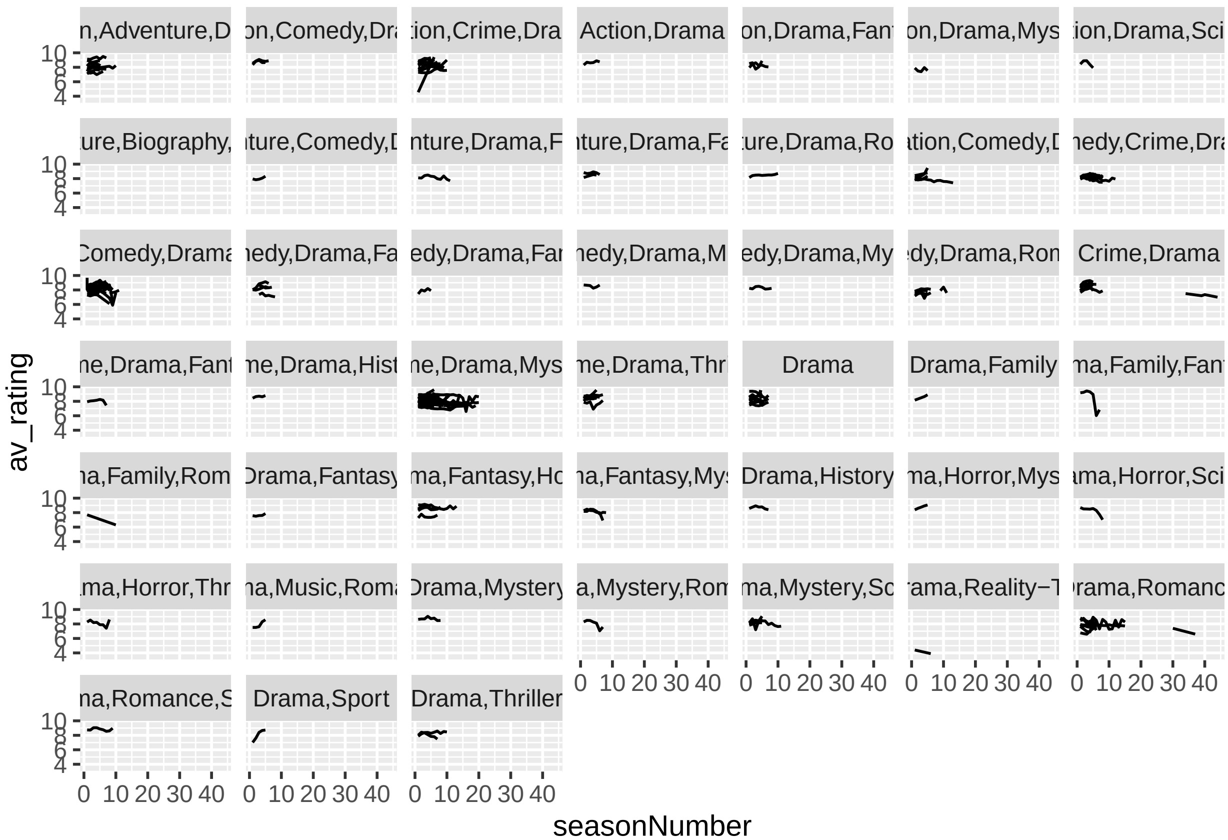
```r
tv2 |>
  filter(genres == "Drama,Family,Fantasy")
```

```
## # A tibble: 8 x 8
## # Groups:   title [2]
##   titleId   seasonNumber title            date       av_ra~1 share genres num_s~2
##   <chr>            <dbl> <chr>            <date>       <dbl> <dbl> <chr>    <dbl>
## 1 tt0103352            1 Are You Afraid~ 1993-04-17    9.17  8.27 Drama~       7
## 2 tt0103352            2 Are You Afraid~ 1993-08-10    9.24  6.98 Drama~       7
## 3 tt0103352            3 Are You Afraid~ 1994-02-23    9.43  2.6  Drama~       7
## 4 tt0103352            4 Are You Afraid~ 1994-11-18    9.31  2.15 Drama~       7
## 5 tt0103352            5 Are You Afraid~ 1995-12-15    8.95  2.31 Drama~       7
## 6 tt0103352            6 Are You Afraid~ 1999-03-22    6.02  0.93 Drama~       7
## 7 tt0103352            7 Are You Afraid~ 2000-04-24    6.83  0.68 Drama~       7
```

```
## 8 tt0108968            5 Touched by an ~ 1998-11-15    9.6   0.08 Drama~       5
## # ... with abbreviated variable names 1: av_rating, 2: num_seasons
```

```
tv2 |>
  ggplot(aes(x = seasonNumber, y = av_rating, group = title)) + geom_line()+ facet_wrap (~ genres)
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

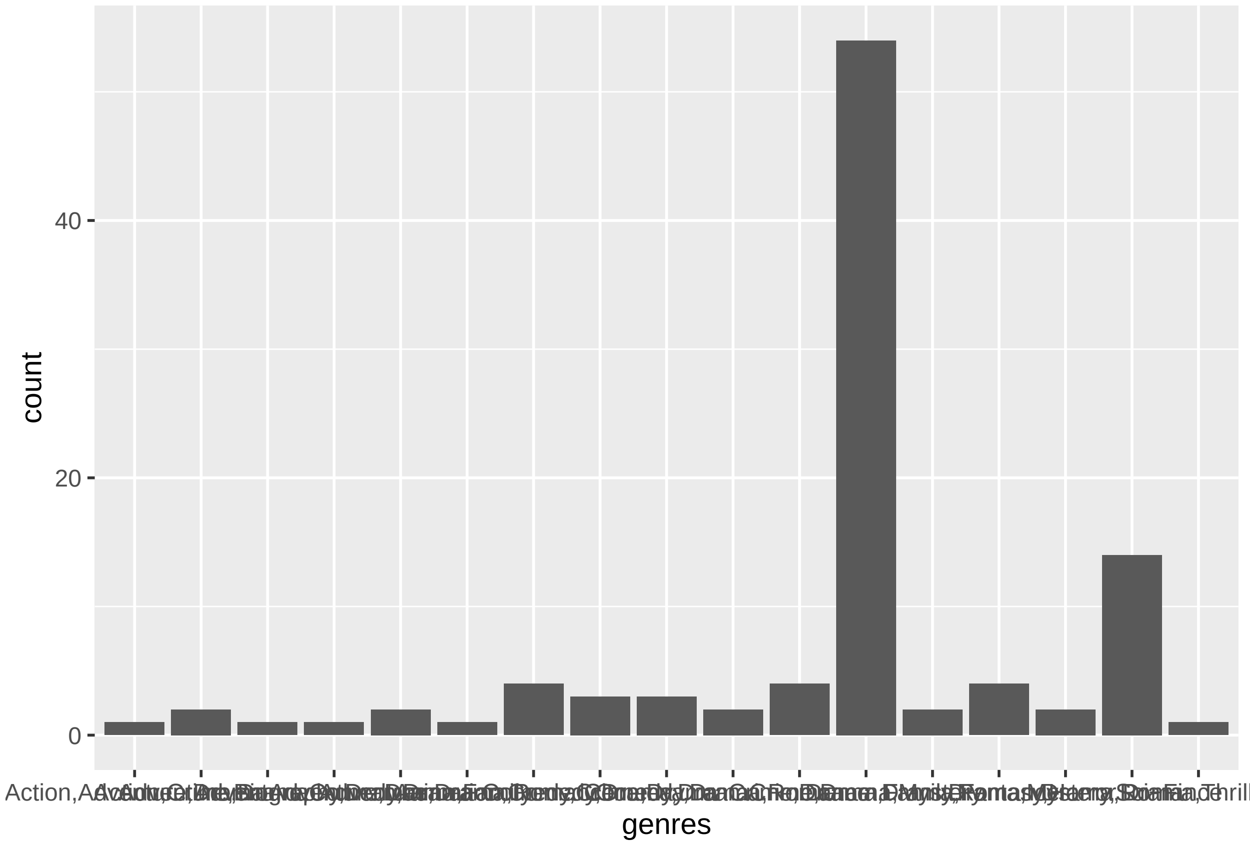


Are you Afraid of the Dark's ratings fell.

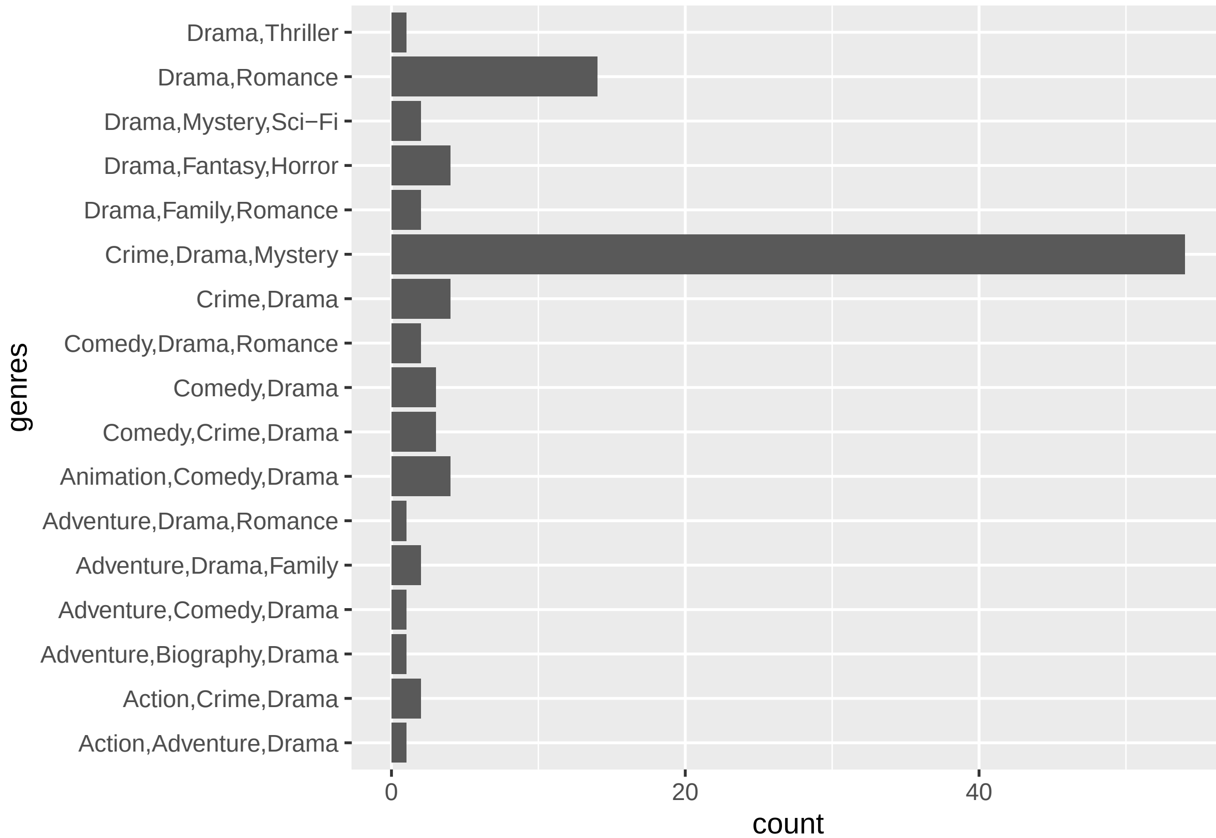
3.

```
tv_9seasons <- tv_ratings |>
  filter(seasonNumber > 9)
```

```
ggplot(data = tv_9seasons, mapping=aes(x=genres))+geom_bar()
```




```
ggplot(data = tv_9seasons, mapping=aes(x=genres))+geom_bar()+coord_flip()
```



Crime, Drama, and Mystery has the most top rated shows

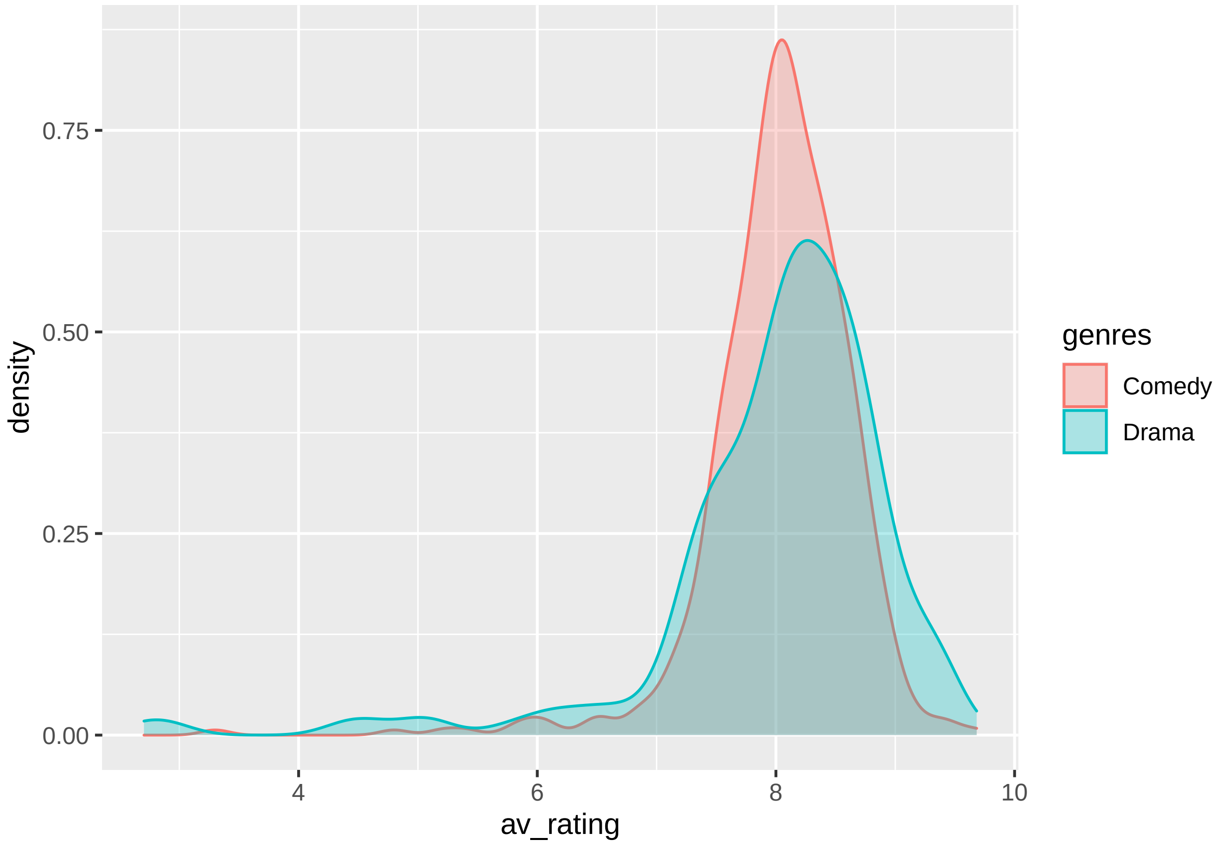
4.

```
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: 8
## $ 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", "Comedy", "Comedy", "Comedy", "Comedy", "Comedy~
## $ is_comedy    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

```
ggplot(data = comedies_dramas, mapping = aes(x=av_rating, fill = genres, color = genres))+geom_density(a
```

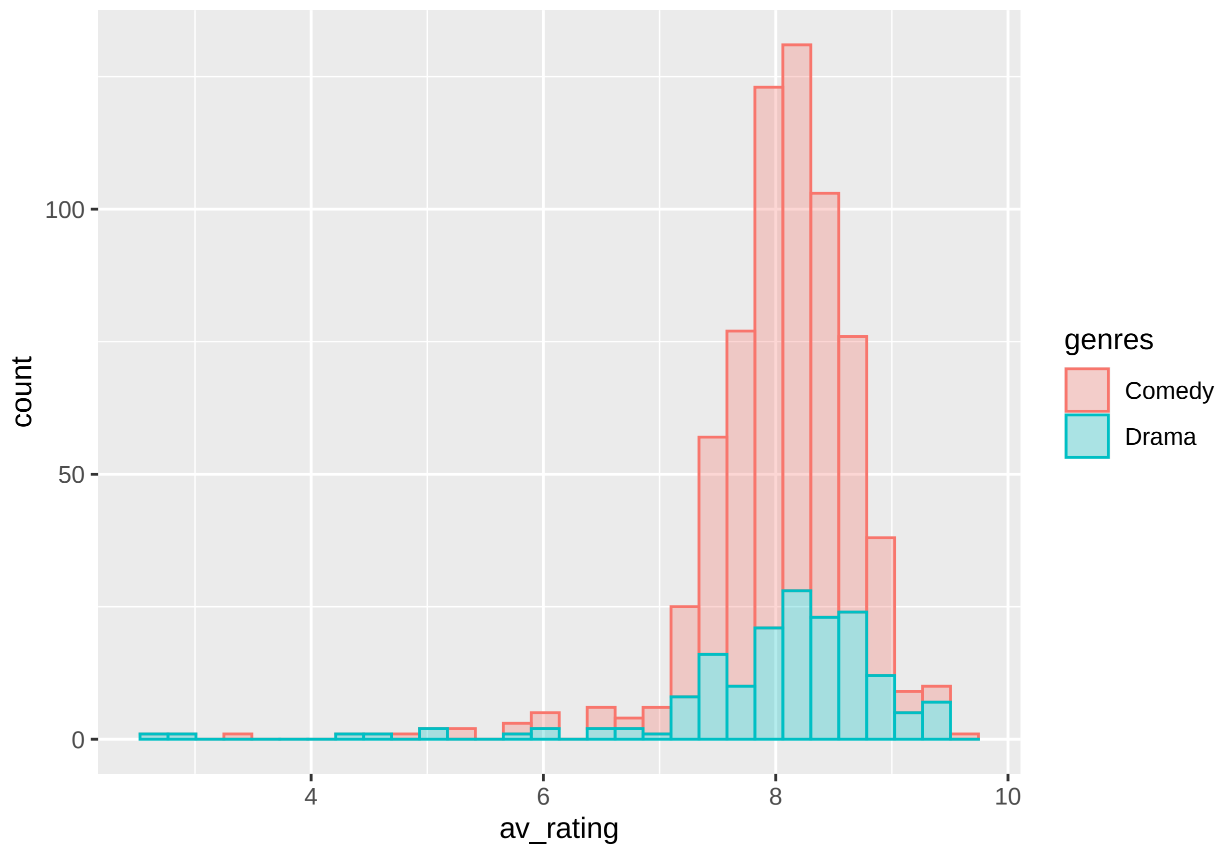


Dramas are still rated higher.

5.

```
ggplot(data = comedies_dramas, mapping = aes(x=av_rating, fill = genres, color = genres))+geom_histogram
```

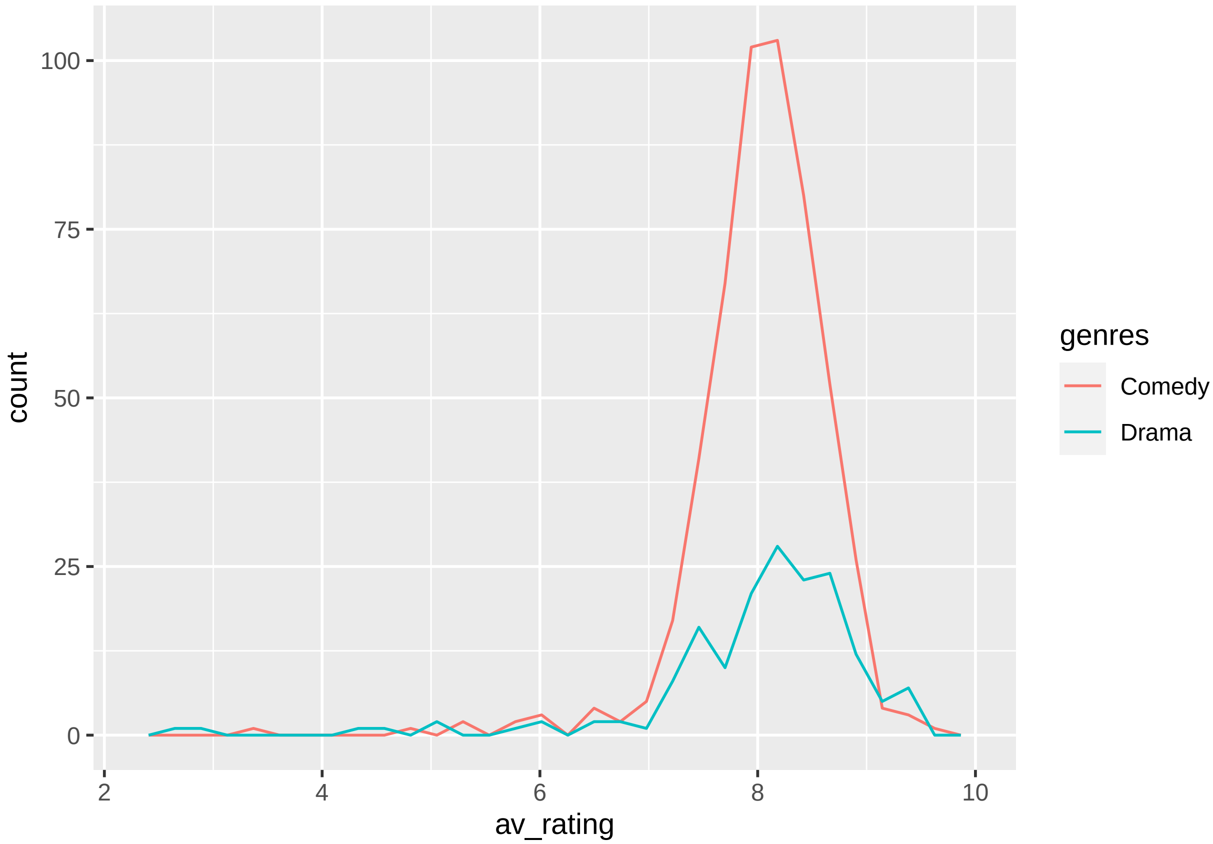
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This shows that there are lots of high rated comedies in the dataset.

```
ggplot(data = comedies_dramas, mapping = aes(x=av_rating, fill = genres, color = genres))+geom_freqpoly
```

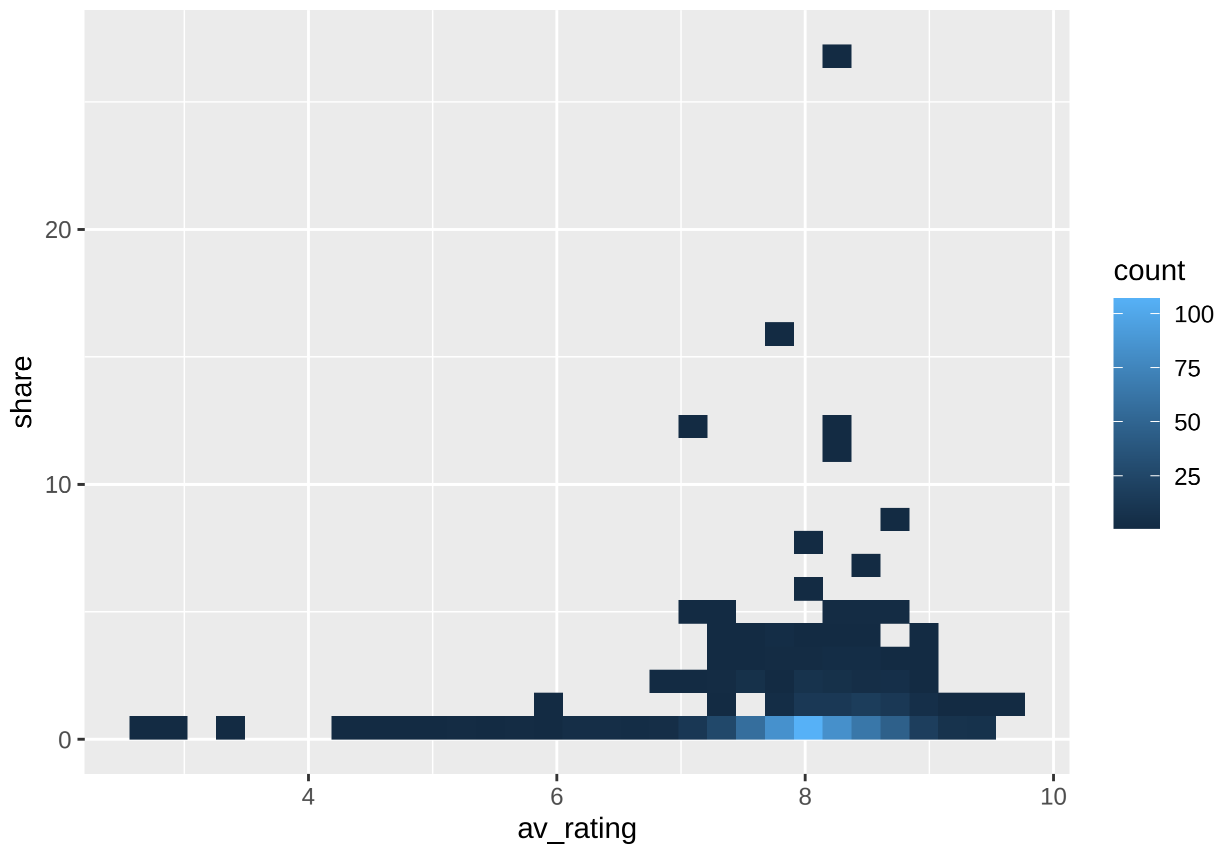
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The difference here is that its more precise in the line graph form. I like the last graph in that it accurately shows the number of comedies and dramas at their different ratings. The first one is helpful in that it shows the average ratings per genre.

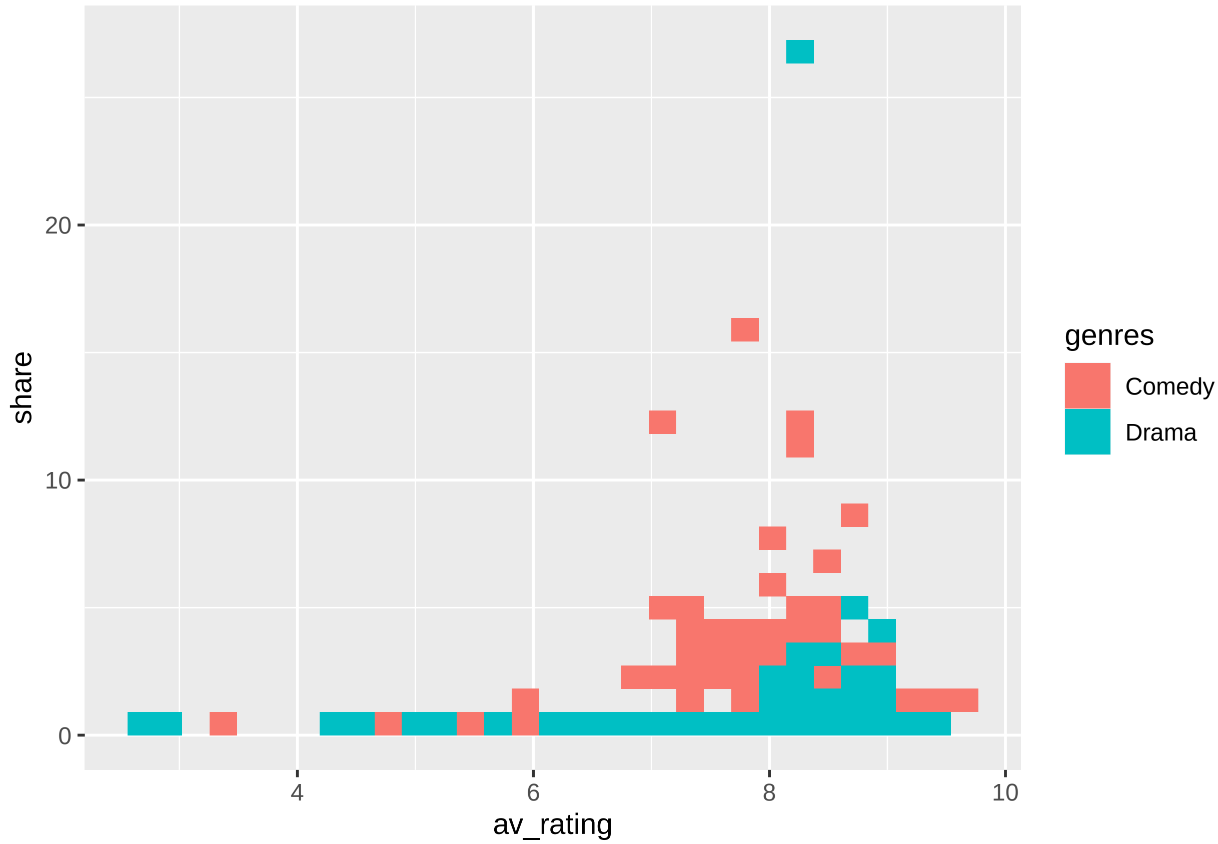
6.

```
ggplot(data = comedies_dramas, mapping=aes(x=av_rating, y=share))+ geom_bin_2d()
```



most shows are around an 8 in average rating and not watched by many people.

```
ggplot(data = comedies_dramas, mapping=aes(x=av_rating, y=share, fill=genres))+ geom_bin_2d()
```



There was one drama that is quality and highly watched. I would say its Breaking Bad.

chapter 5

1.

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

```
wncaaChamp <- wncaa |>
  filter(tourney_finish=="Champ")
```

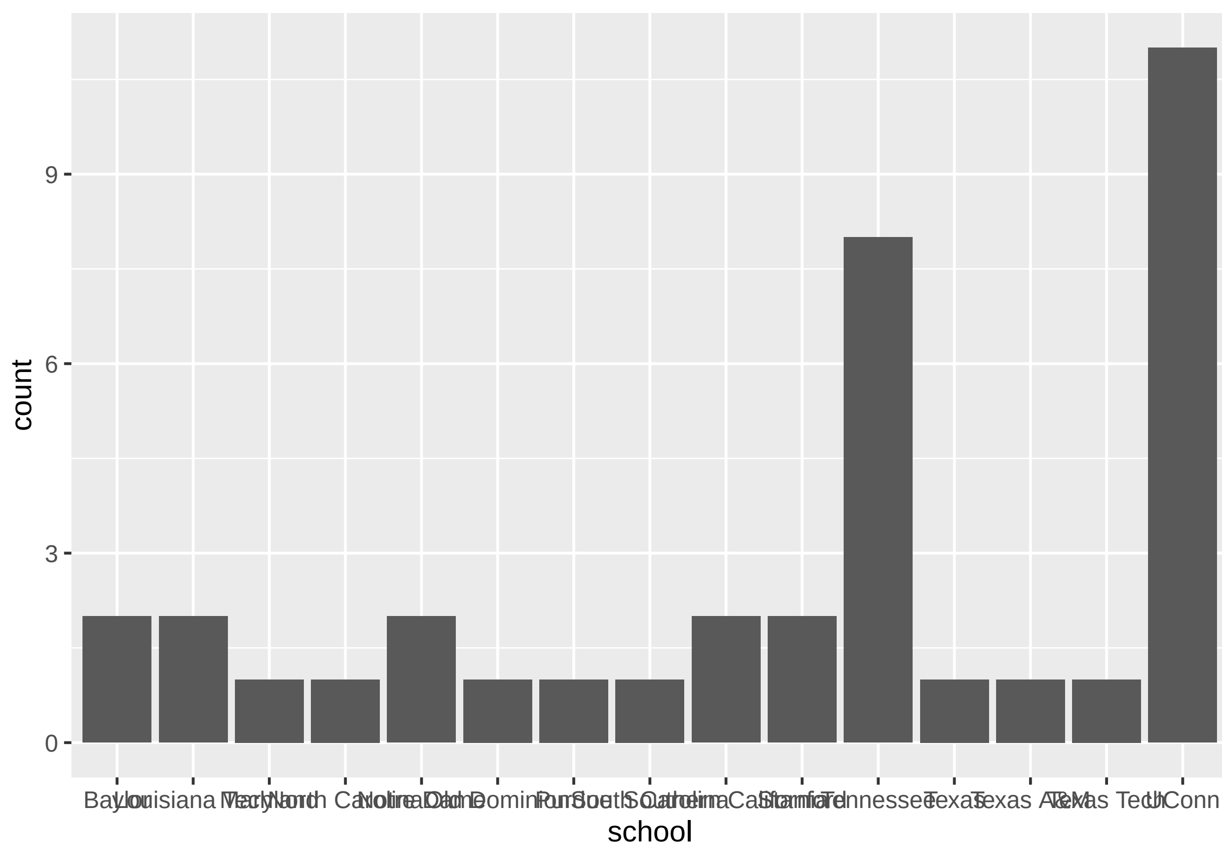
```
wncaaChamp |>
  group_by(school, tourney_finish) |>
  summarise(N=n()) |>
  mutate(freq=N/sum(N), pct=round(freq*100), 0)
```

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

```
## # A tibble: 15 x 6
## # Groups:   school [15]
```

```
##    school              tourney_finish     N  freq   pct   `0`
##    <chr>               <chr>          <int> <dbl> <dbl> <dbl>
##  1 Baylor              Champ              2     1   100     0
##  2 Louisiana Tech      Champ              2     1   100     0
##  3 Maryland            Champ              1     1   100     0
##  4 North Carolina      Champ              1     1   100     0
##  5 Notre Dame          Champ              2     1   100     0
##  6 Old Dominion        Champ              1     1   100     0
##  7 Purdue              Champ              1     1   100     0
##  8 South Carolina      Champ              1     1   100     0
##  9 Southern California Champ              2     1   100     0
## 10 Stanford            Champ              2     1   100     0
## 11 Tennessee           Champ              8     1   100     0
## 12 Texas               Champ              1     1   100     0
## 13 Texas A&M           Champ              1     1   100     0
## 14 Texas Tech          Champ              1     1   100     0
## 15 UConn               Champ             11     1   100     0
```

```
ggplot(data = wncaaChamp, mapping=aes(x=school))+geom_bar()
```



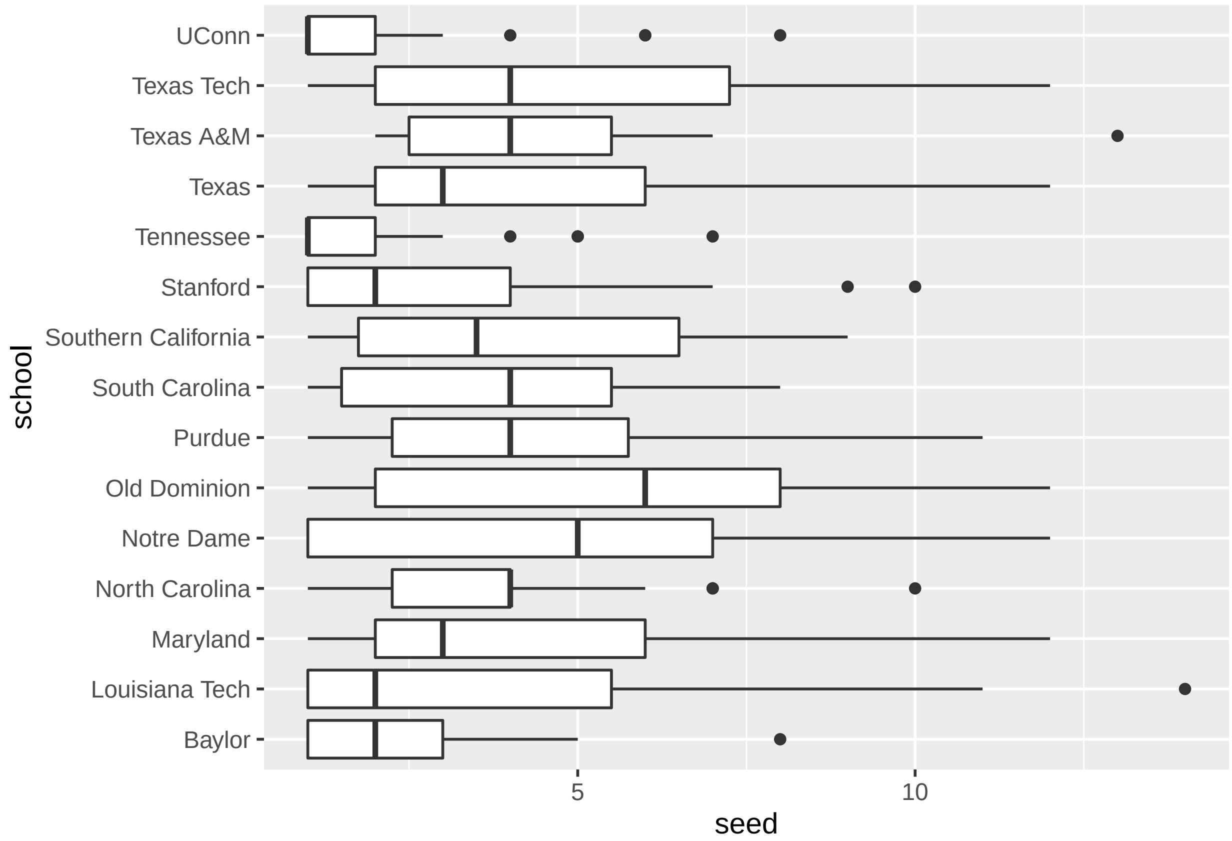
It appears that UCONN and Tennessee have won the most times.

2.

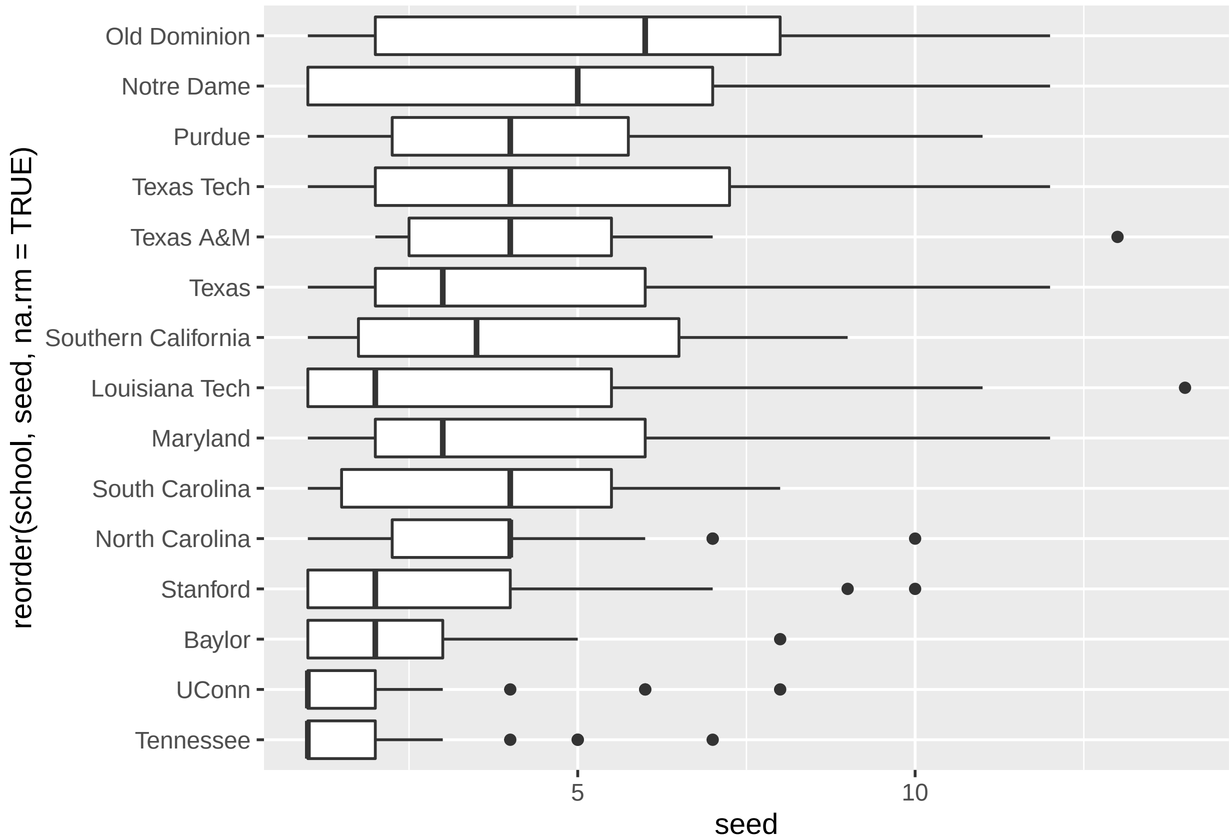
```
champ_names <- unique(wncaaChamp$school)
```

```
winners <- wncaa %>%
  filter(school %in% champ_names)
```

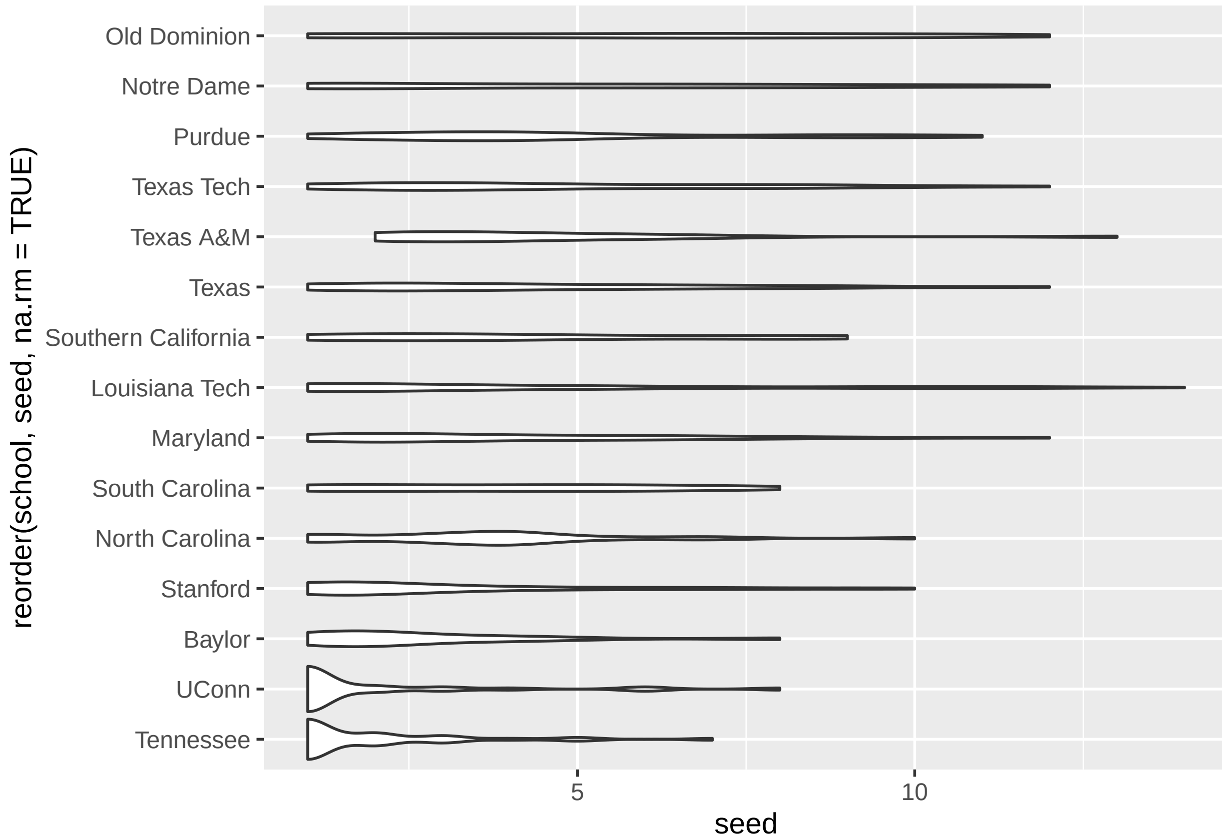
```
ggplot(data = winners, mapping = aes(x=school, y=seed))+geom_boxplot()+coord_flip()
```




```
ggplot(data = winners, mapping = aes(x= reorder(school, seed, na.rm=TRUE), y=seed))+geom_boxplot()+ coo
```



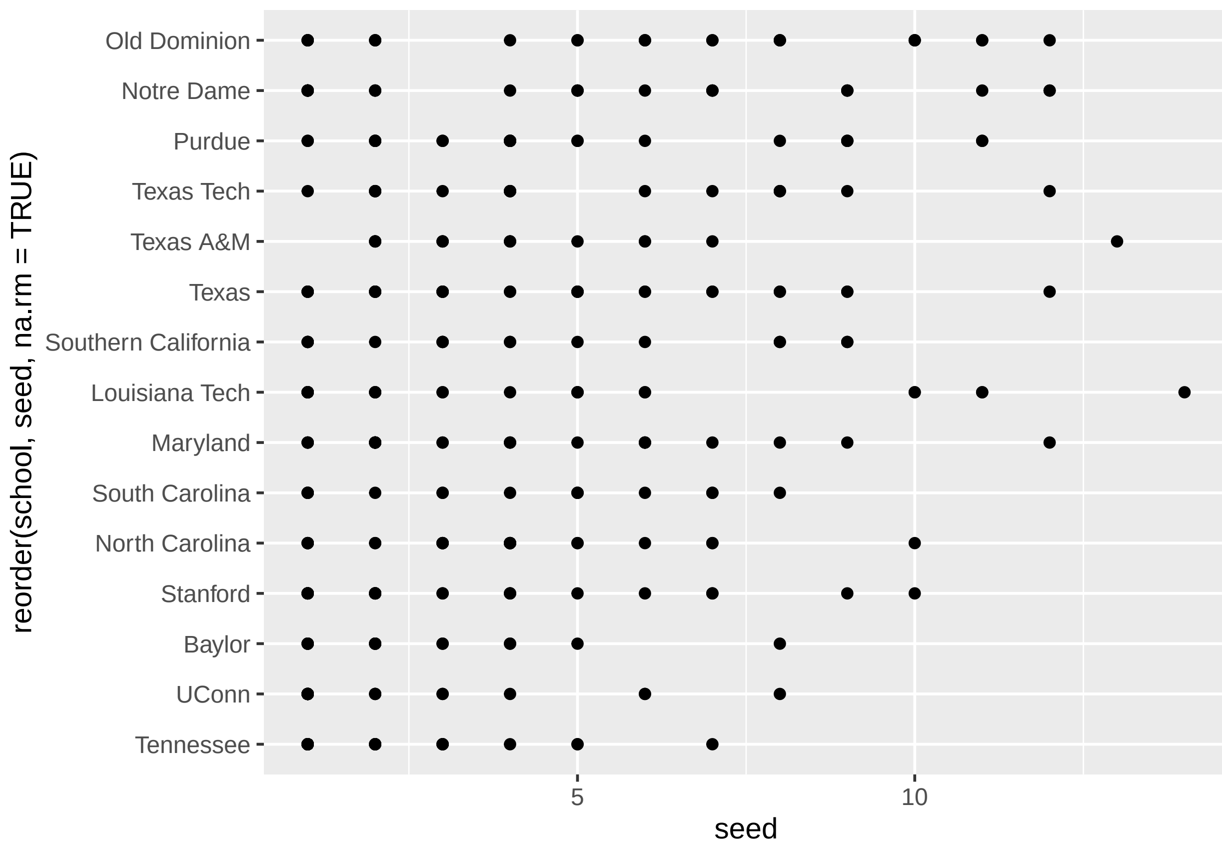

```
ggplot(data = winners, mapping = aes(x= reorder(school, seed, na.rm=TRUE), y=seed))+geom_violin()+ coord
```



I think the boxplot is slightly more aesthetically appealing for me. There are some outliers in that lower seeds still won some years.

3.

```
ggplot(data = winners, mapping = aes(x= reorder(school, seed, na.rm=TRUE), y=seed))+geom_point()+ coord_
```



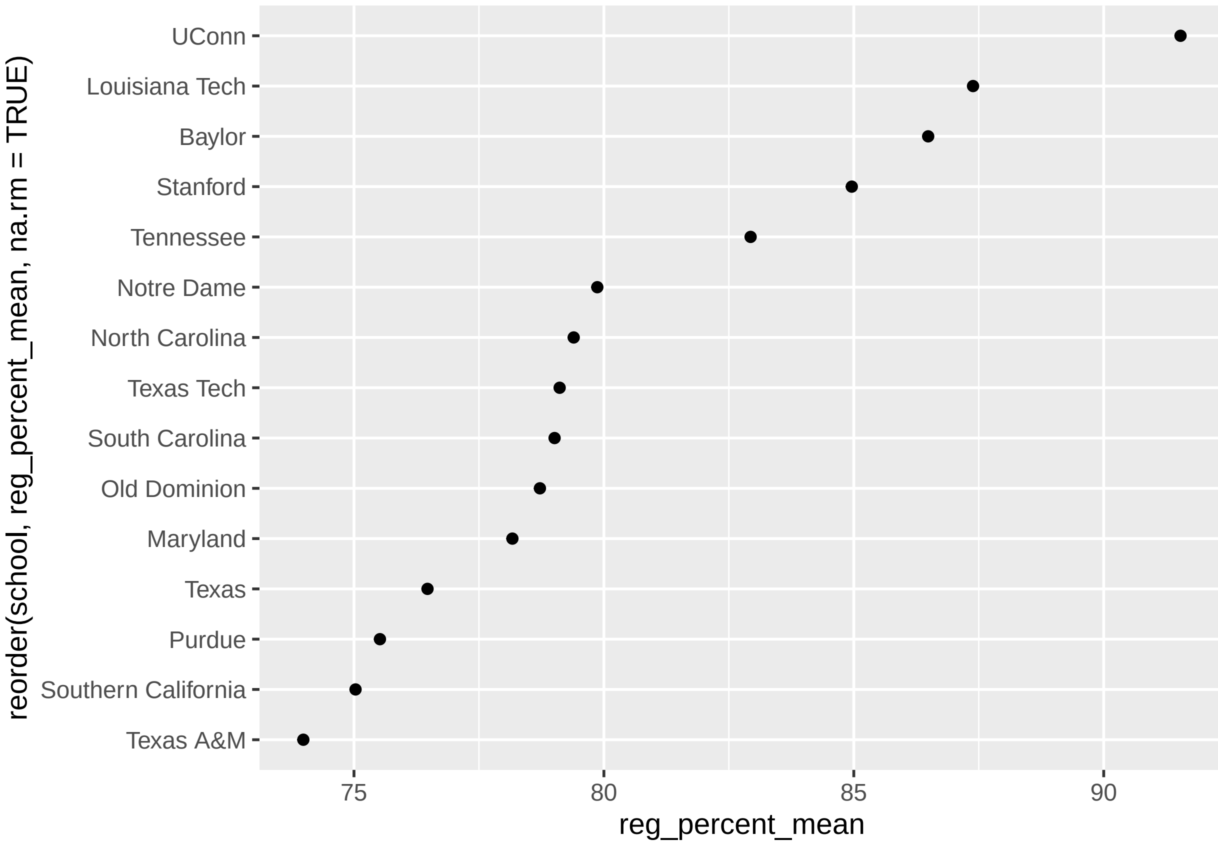
The point doesn't show the range as well. Its less clear what seed the school commonly is.

4.

```
winnersmean <-winners |>
  group_by(school) |>
  summarize_if(is.numeric, funs(mean, sd), na.rm =TRUE)
```

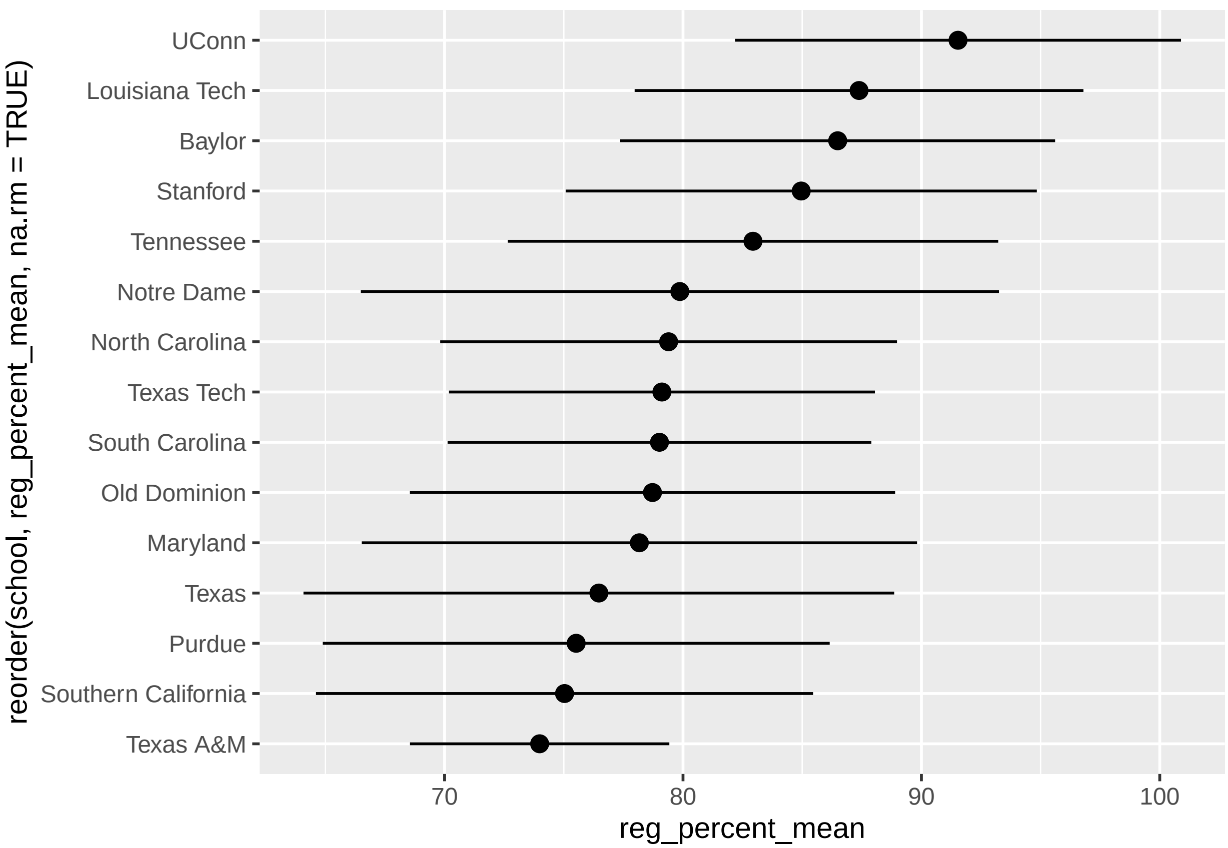
```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
ggplot(data = winnersmean, mapping=aes(x= reorder(school, reg_percent_mean, na.rm=TRUE), y=reg_percent_
```



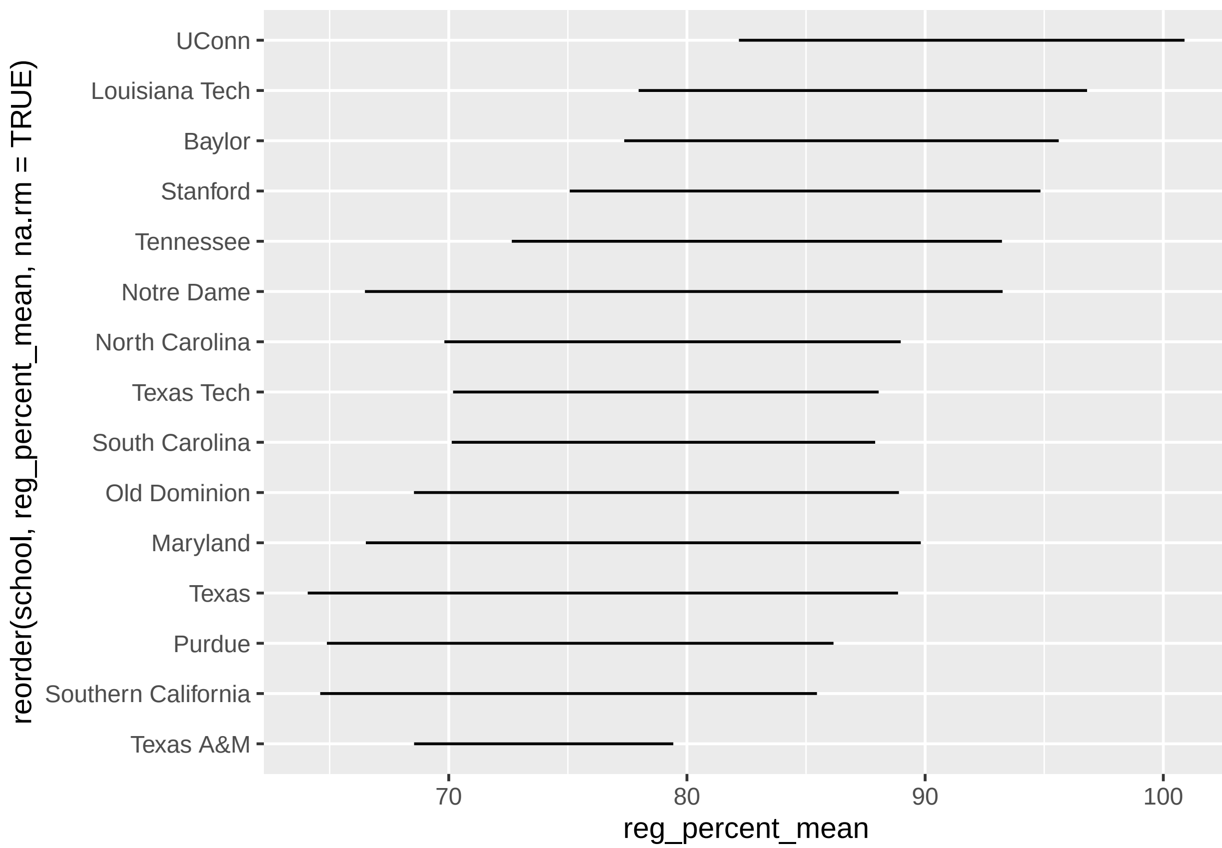
UCONN has won the most championships and has the highest regular season win percentage.

```
ggplot(data = winnersmean, mapping=aes(x= reorder(school, reg_percent_mean, na.rm=TRUE), y=reg_percent_r
```



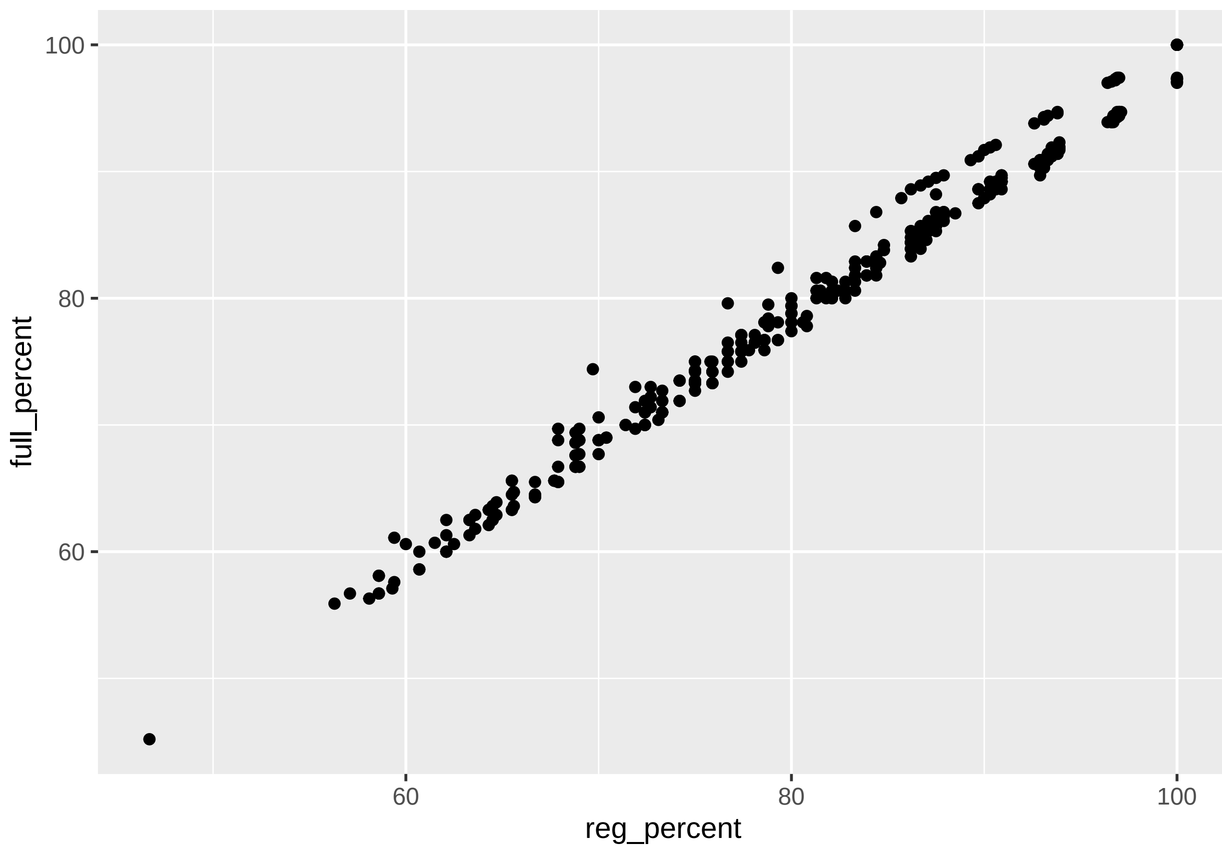
Texas A&M has the narrowest interval. They performed similarly in the regular season every year they won a championship.

```
ggplot(data = winnersmean, mapping=aes(x= reorder(school, reg_percent_mean, na.rm=TRUE), y=reg_percent_m
```

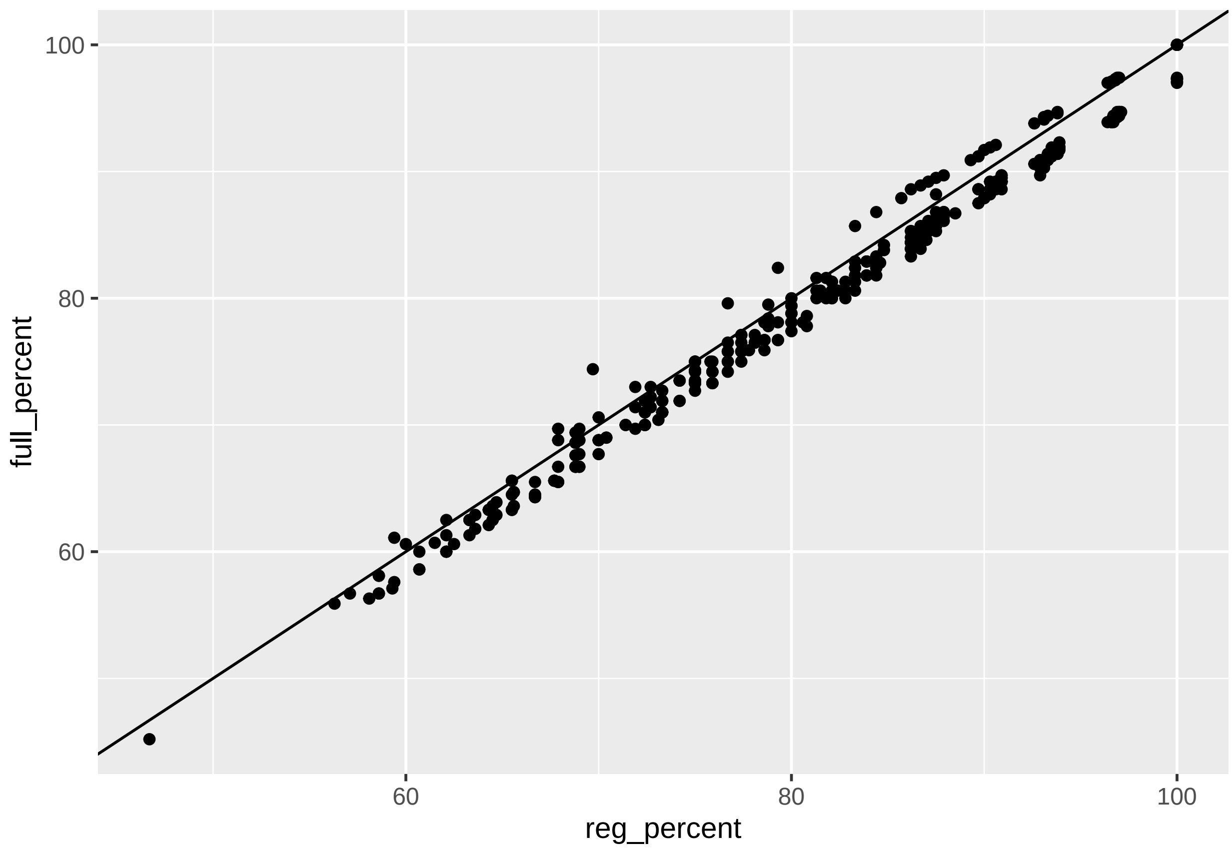


5.

```
ggplot(data = winners, mapping=aes(x=reg_percent, y=full_percent))+geom_point()
```




```
ggplot(data = winners, mapping=aes(x=reg_percent, y=full_percent))+geom_point()+geom_abline()
```

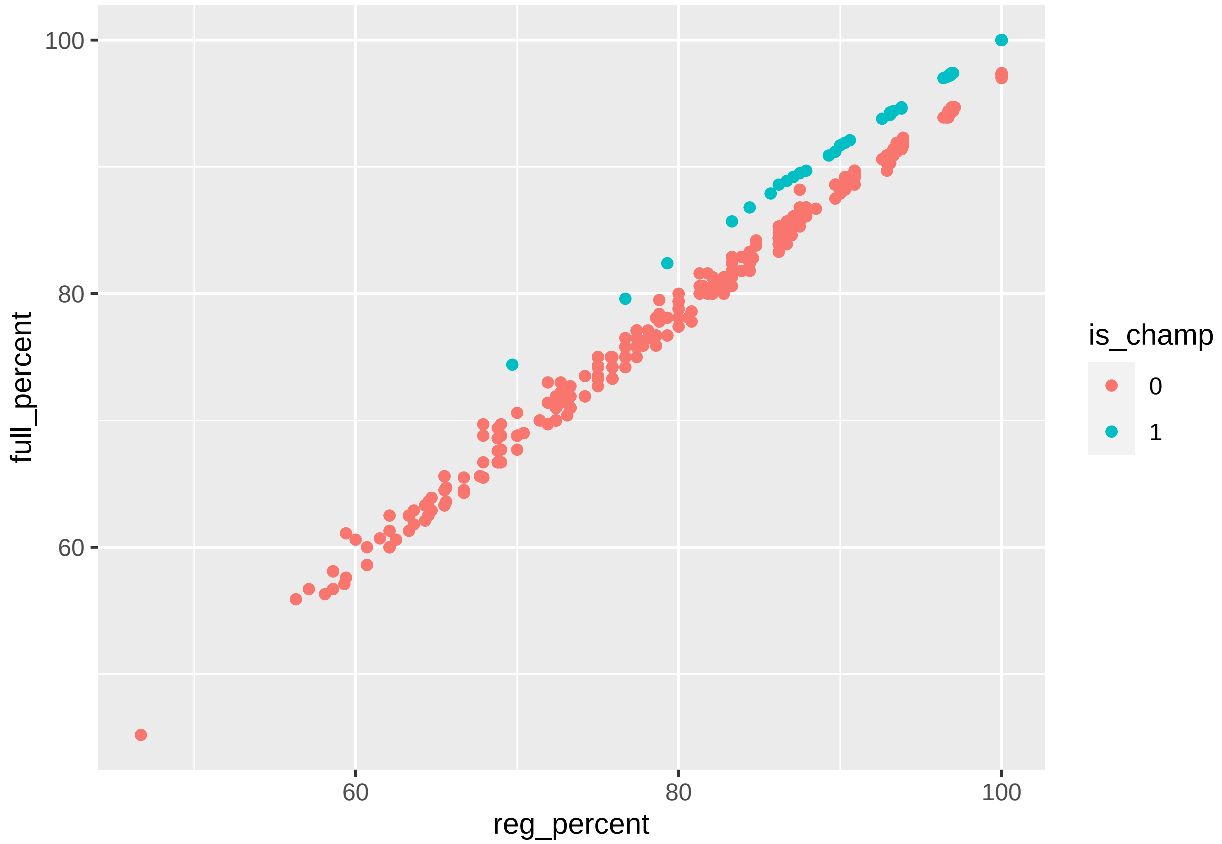


I feel like if we are looking at the winners data the teams that won should have had their postseason be more successful than the regular season just because in college basketball the march madness tournament is single game elimination. This might be including other postseason tournaments other than March Madness though.

6.

```
winners <- winners %>%
  mutate(is_champ = if_else(tourney_finish == "Champ", 1, 0),
         is_champ = as.factor(is_champ))
```

```
ggplot(data = winners, mapping=aes(x=reg_percent, y=full_percent, color=is_champ))+geom_point()
```



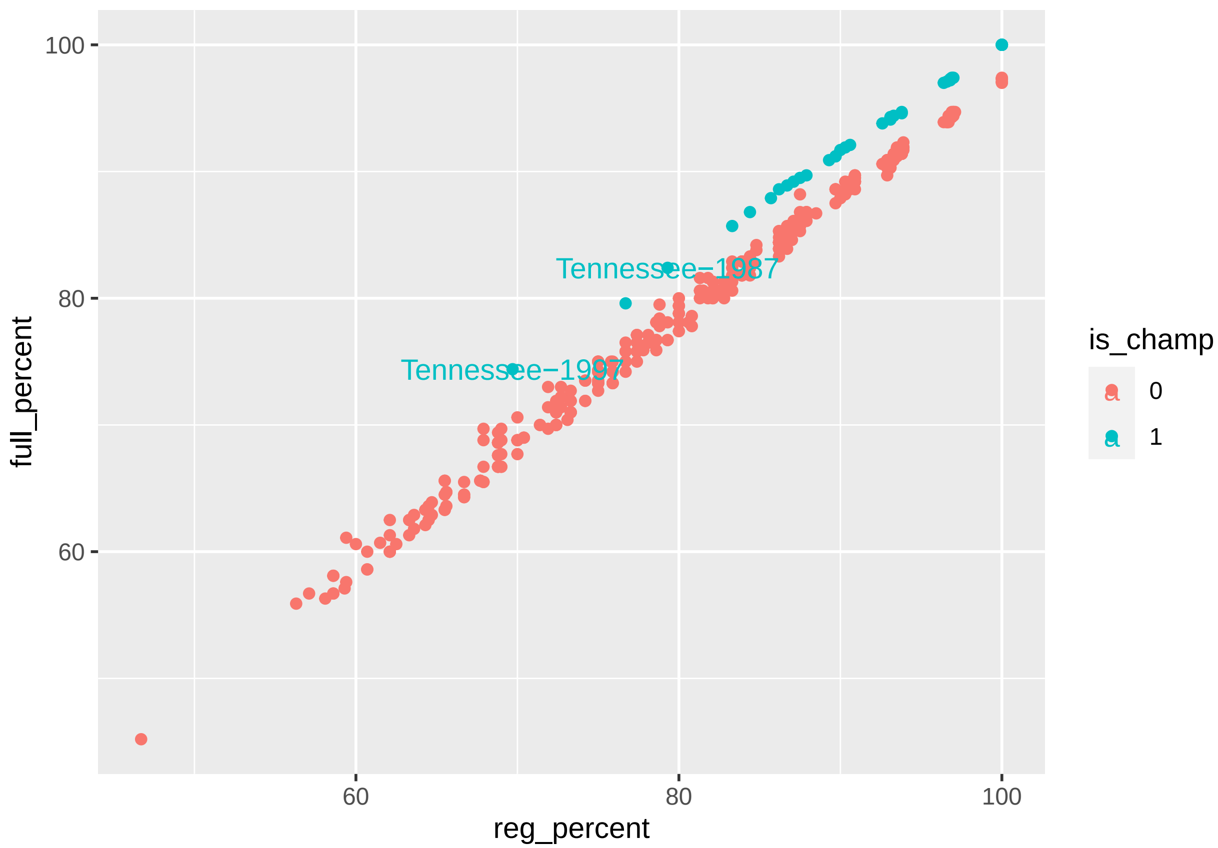
It wouldn't be able to know who was the champ and who wasn't if it wasn't made a factor. The pattern is that championship teams did better in the post season than regular season

7.

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

```
winners <- winners %>%
  mutate(difference = full_percent - reg_percent)
```

```
ggplot(data = winners, mapping=aes(x=reg_percent, y=full_percent, color=is_champ))+geom_point() +geom_t
```



The school was Tennessee

8.

```
winners |>
  filter(full_percent==100)
```

```
## # A tibble: 8 x 22
##    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>, is_champ <fct>, plot_label <chr>,
## #   difference <dbl>, and abbreviated variable names 1: conference,
## #   2: conf_percent, 3: conf_place, 4: reg_percent
```

UConn in 1995, 2002, 2009, 2010, 2014, and 2016. Texas in 1986, and Baylor in 2012

This makes sense because UConn dominates women's basketball.