HW3

Alexandra Gibbons

2022-09-16

# DV 3

## Question 1

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.8 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.1  
## ✔ readr 2.1.2 ✔ forcats 0.5.2  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

# 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`

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…

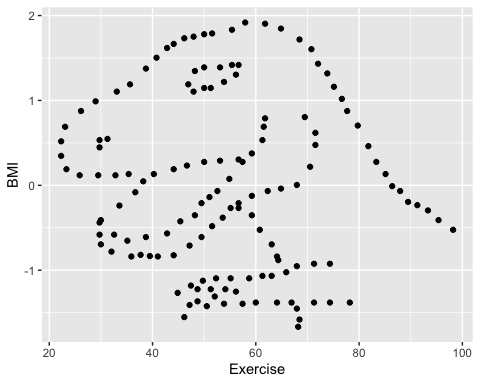
I expect that people who record more exercise will have lower BMIs.

cor(exercise\_data$Exercise, exercise\_data$BMI)

## [1] -0.06447185

The correlation coefficient is negative, which indicates that an increase in recorded exercise time is associated with a decrease in BMI.

exercise\_data %>%   
 ggplot(aes(x = Exercise,   
 y = BMI)) +  
 geom\_point()

 ## Question 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.…

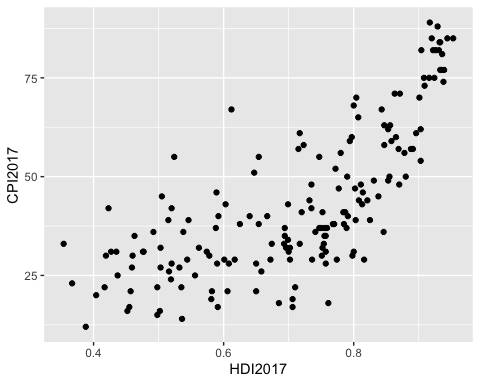
?corruptDF

CPI2017 is a country’s (or territory’s) score on the Corruption Perceptions Index in 2017, which measures the perception of the level of corruption in the public sector. Scores range from 0 to 100, with 0 meaning that a given country is percieved as being very corrupt, and a score of 100 meaning that the country is not percieved as corrupt in the public sector.

HDI2017 is a country’s or territory’s Human Development Index score from 2017, which represents human development based on life expectancy, education levels, and income.

## Question 3

corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017)) +  
 geom\_point()

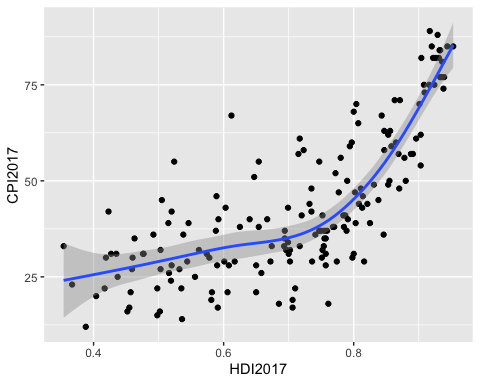


There seems to be a positive association between HDI and CPI scores in 2017. This relationship doesn’t look perfectly linear and may be best represented with a quadratic function.

## Question 4

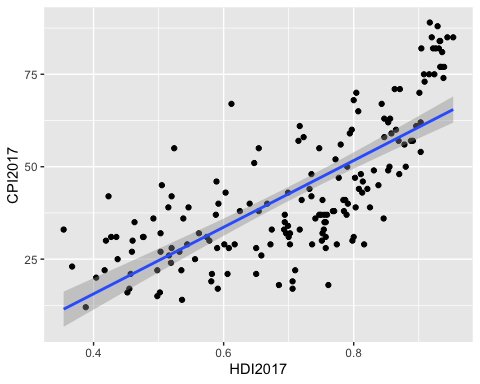
corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017)) +  
 geom\_point() +  
 geom\_smooth(method="gam")

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



corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017)) +  
 geom\_point() +  
 geom\_smooth(method="lm")

## `geom\_smooth()` using formula 'y ~ x'

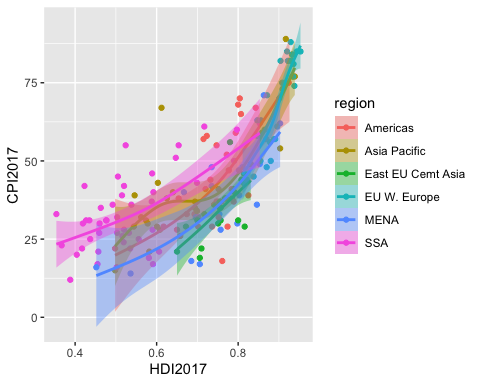


The gam method fits a generalized additive model, which is more flexible and fits this data much better than the lm method, which fits a simple linear model and thus renders a straight line.

## Question 5

corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017,  
 color=region,  
 fill=region)) +  
 geom\_point() +  
 geom\_smooth(method="gam")

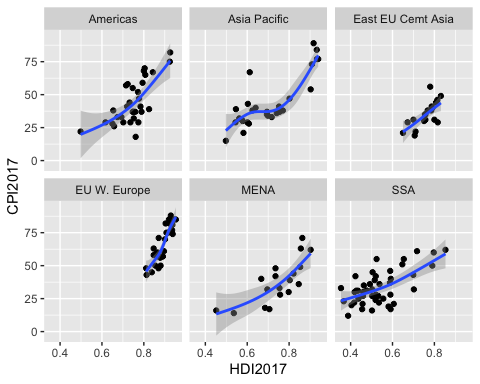
## `geom\_smooth()` using formula 'y ~ s(x, bs = "cs")'



Haha, this is way too cluttered. Let’s try again…

corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017)) +  
 geom\_point() +  
 geom\_smooth(method="gam") +  
 facet\_wrap(~region)

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

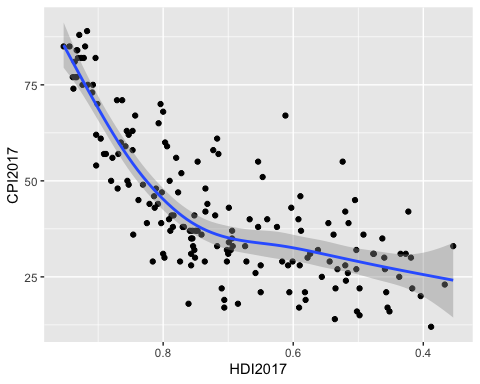


This is much more digestible.

## Question 6

corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017)) +  
 geom\_point() +  
 geom\_smooth(method="gam") +  
 scale\_x\_reverse()

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



Done, though I don’t particularly like it.

## Question 7

final\_plot <- corruptDF %>%   
 ggplot(aes(x=HDI2017,  
 y=CPI2017)) +  
 geom\_point() +  
 geom\_smooth(method="gam") +  
 labs(x="Human Development Index Score",  
 y="Corruption Perceptions Index Score",  
 title="Human Development and Corruption Perception in 2017",  
 subtitle="Data points are countries",  
 caption="Source: causact package in R")

## Question 8

#setwd("/Users/ajg109//Documents/Github/sociol\_722/HW3")  
#ggsave("hw3\_plot.pdf", plot=final\_plot)

I commented this out because it was preventing knitting but I promise it worked and I have since used ggsave to successfully save a graph I made for work!

# DV 4

## Question 1

library(tidyverse)  
  
# Read in the data   
tv\_ratings <- read\_csv("Data/tv\_ratings.csv")

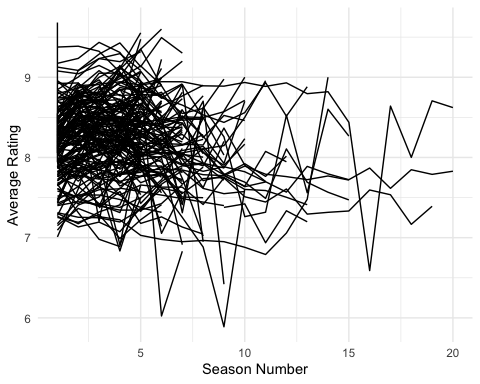
## Rows: 2266 Columns: 7  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): titleId, title, genres  
## dbl (3): seasonNumber, av\_rating, share  
## date (1): date  
##   
## ℹ Use `spec()` to retrieve the full column specification 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…

tv\_long <- tv\_ratings %>%   
 group\_by(title) %>%   
 summarise(num\_seasons = n()) %>%   
 ungroup() %>%   
 left\_join(tv\_ratings, by = "title")   
  
tv\_long <- tv\_long %>%   
 filter(num\_seasons >= 5)

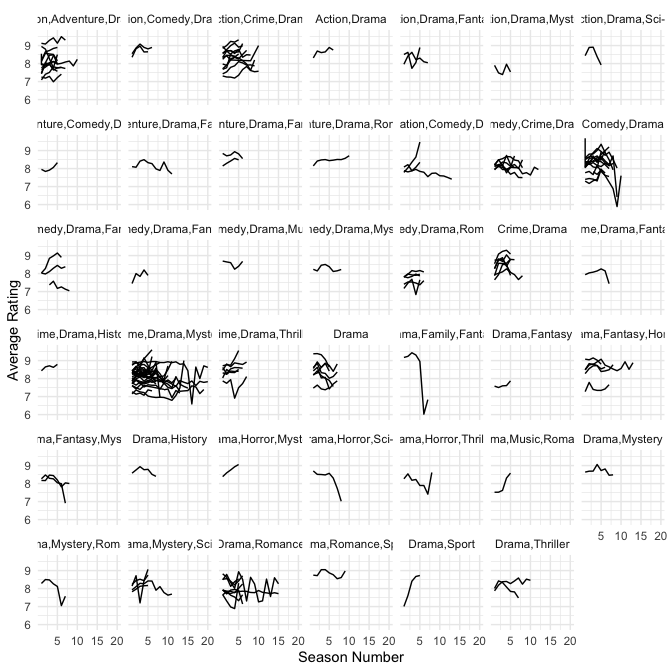
tv\_long %>%   
 ggplot(aes(x=seasonNumber,  
 y=av\_rating,  
 group=title)) +  
 geom\_line() +   
 labs(x="Season Number", y="Average Rating") +  
 theme\_minimal()



This graph is extremely messy and it is difficult to draw any conclusions.

## Question 2

tv\_long %>%   
 ggplot(aes(x=seasonNumber,  
 y=av\_rating,   
 group=title)) +  
 geom\_line() +  
 facet\_wrap(~genres) +   
 labs(x="Season Number", y="Average Rating") +  
 theme\_minimal()



The following genres of shows seem to last especially long: drama, romance and crime, drama, mystery. Ratings do seem to change across seasons.

tv\_long %>%   
 filter(genres=="Drama,Family,Fantasy") %>%   
 dplyr::select(title)

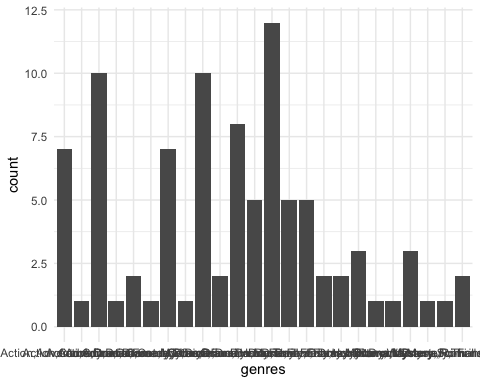
## # A tibble: 7 × 1  
## title   
## <chr>   
## 1 Are You Afraid of the Dark?  
## 2 Are You Afraid of the Dark?  
## 3 Are You Afraid of the Dark?  
## 4 Are You Afraid of the Dark?  
## 5 Are You Afraid of the Dark?  
## 6 Are You Afraid of the Dark?  
## 7 Are You Afraid of the Dark?

I assumed that the show that plummeted would be GOT, but it is actually “Are You Afraid of the Dark?”.

## Question 3

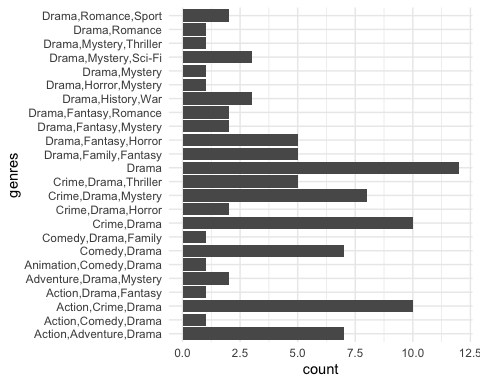
tv\_best <- tv\_ratings %>%   
 filter(av\_rating>=9)

tv\_best %>%   
 ggplot(aes(x=genres)) +  
 geom\_bar() + theme\_minimal()



First try, impossible to read the x-axis.

tv\_best %>%   
 ggplot(aes(x=genres)) +  
 geom\_bar() +  
 coord\_flip() +   
 theme\_minimal()



labs(title="Number of Show Seasons with an Avereage Rating of 9/10 or Greater by Genre")

## $title  
## [1] "Number of Show Seasons with an Avereage Rating of 9/10 or Greater by Genre"  
##   
## attr(,"class")  
## [1] "labels"

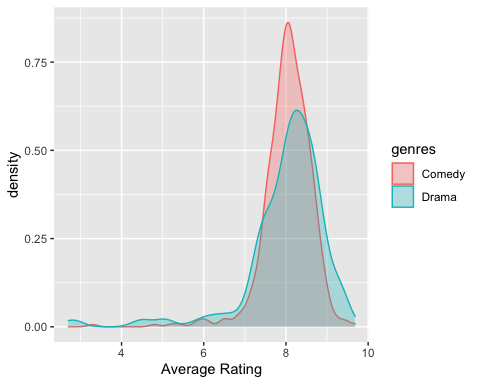
Wow, much better. coord\_flip flips the coordinates so that the original x-axis is shown on the y-axis and vice versa. Drama is the genre with the most top-rated shows.

## Question 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, …

comedies\_dramas %>%   
 ggplot(aes(x=av\_rating,  
 fill=genres,  
 color=genres)) +  
 geom\_density(alpha=.3) +  
 labs(x="Average Rating")

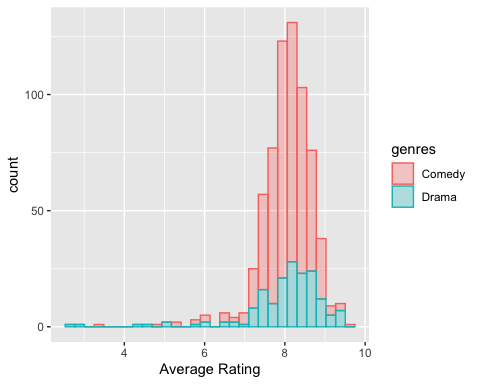


No, sorry Nico, dramas have a slightly higher peak (most common rating), and also have more extremely highly rated tv seasons than do comedies.

## Question 5

comedies\_dramas %>%   
 ggplot(aes(x=av\_rating,  
 fill=genres,  
 color=genres)) +  
 geom\_histogram(alpha=.3) +  
 labs(x="Average Rating")

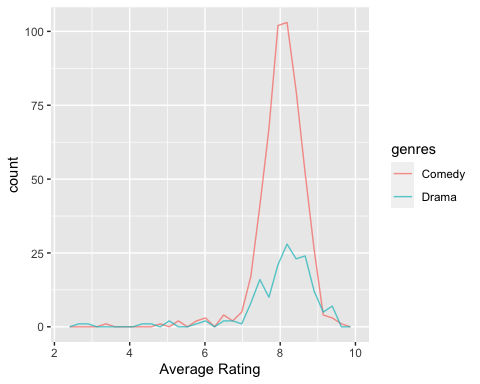
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



The histogram is helpful because it shows raw counts and makes it clear that we have many more comedies in our dataset than dramas. However, it is difficult to compare the relative frequencies of different ratings.

comedies\_dramas %>%   
 ggplot(aes(x=av\_rating,  
 fill=genres,  
 color=genres)) +  
 geom\_freqpoly(alpha=.7) +  
 labs(x="Average Rating")

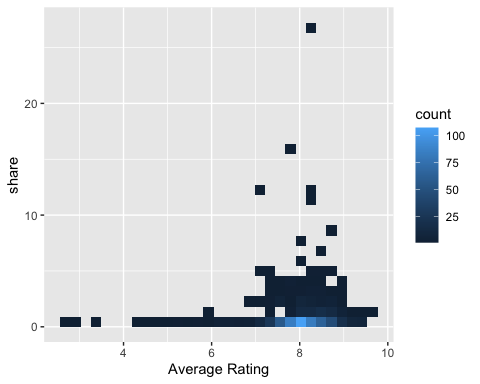
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



This plot seems to show the same information as the histogram except using lines in place of bars. I think that for our purposes, the density plot is most informative because it accounts for the differences in the number of observations for each genre.

## Question 6

comedies\_dramas %>%   
 ggplot(aes(x=av\_rating, y=share)) +  
 geom\_bin\_2d() +   
 labs(x="Average Rating")



This gives you information about the relationship between two different variables and about the distribution of tv shows along these dimensions. It also addresses issues related to overplotting in a scatterplot as the fill gives information about the count of shows within each 2d bin. I’m curious about the show with such a viewer share.

comedies\_dramas %>%   
 ggplot(aes(x=av\_rating, y=share, fill=genres)) +  
 geom\_bin\_2d() +  
 labs(x="Average Rating")



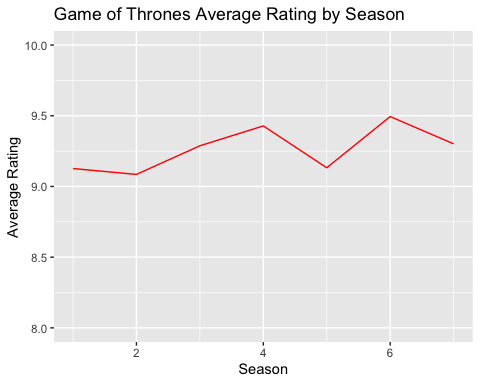
Comedy shows with lower ratings tend to have higher viewership shares than do dramas with higher ratings, which is interesting (besides the one outlier).

mystery\_show <- comedies\_dramas %>%   
 filter(share>20)

The show is “Dekalog.” Never heard of it…

## Just for fun

got <- tv\_ratings %>%   
 filter(title=="Game of Thrones")  
  
got %>%   
 ggplot(aes(x=seasonNumber,  
 y=av\_rating)) +  
 geom\_line(color="red") +  
 labs(x="Season",  
 y="Average Rating",  
 title="Game of Thrones Average Rating by Season") +ylim(8, 10)



# DV 5

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.

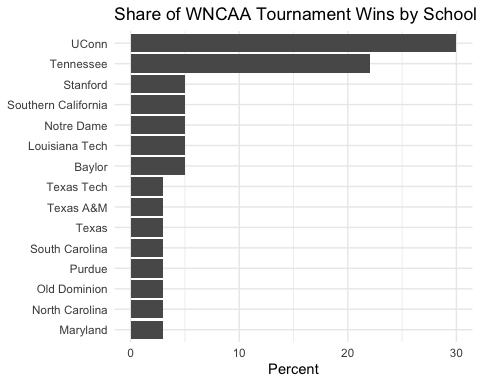
# Glimpse the data   
glimpse(wncaa)

## 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

tourney\_wins <- wncaa %>%   
 filter(tourney\_finish=="Champ") %>%   
 group\_by(school) %>%   
 summarize(N=n()) %>%   
 mutate(freq=N/ sum(N),  
 pct= round((freq\*100), 0))

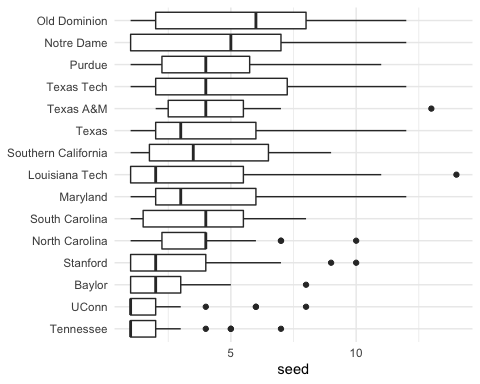
tourney\_wins %>%   
 ggplot(aes(x=reorder(school, pct), y=pct)) +  
 geom\_col(position="dodge") +  
 coord\_flip() +   
 labs(x=NULL, y="Percent",  
 title="Share of WNCAA Tournament Wins by School") +  
 theme\_minimal()

 Together, UConn and Tennessee have won the majority of WNCAA tournaments. The rest of these teams have won a significantly lower percentage of tournaments.

## Question 2

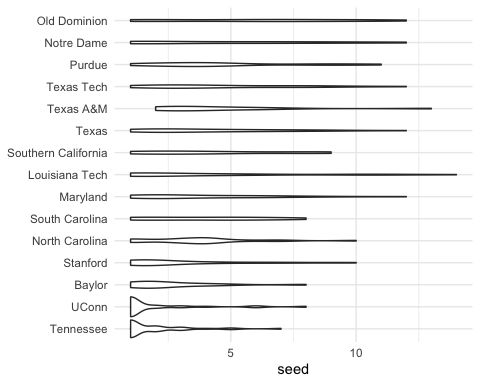
champ\_names <- unique(tourney\_wins$school)  
  
winners <- wncaa %>%   
 filter(school %in% champ\_names)

winners %>% ggplot(aes(x=reorder(school, seed, na.rm=TRUE),  
 y=seed)) +  
 geom\_boxplot() +  
 coord\_flip() +   
 labs(x=NULL) +  
 theme\_minimal()



Honestly, don’t know what seeds are. Based on my brief read on the subject we would expect UConn and Tennessee to have low seeds. Interesting that Notre Dame has the second highest mean and median given they are in the second tier of schools based on the graph from Q1.

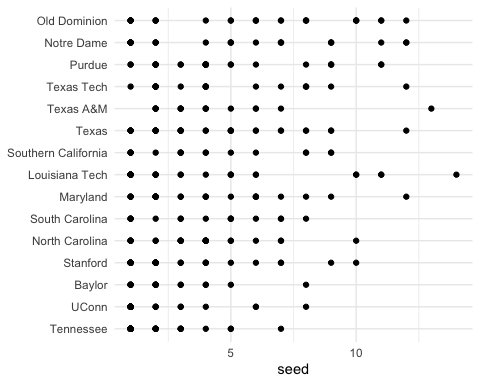
winners %>% ggplot(aes(x=reorder(school, seed, na.rm=TRUE),  
 y=seed)) +  
 geom\_violin() +  
 coord\_flip() +   
 labs(x=NULL) +  
 theme\_minimal()



I personally find the boxplot more informative and digestible but I appreciate that the violin plot also shows the full distribution of data.

## Question 3

winners %>% ggplot(aes(x=reorder(school, seed, na.rm=TRUE),  
 y=seed)) +  
 geom\_point() +  
 coord\_flip() +   
 labs(x=NULL) +  
 theme\_minimal()



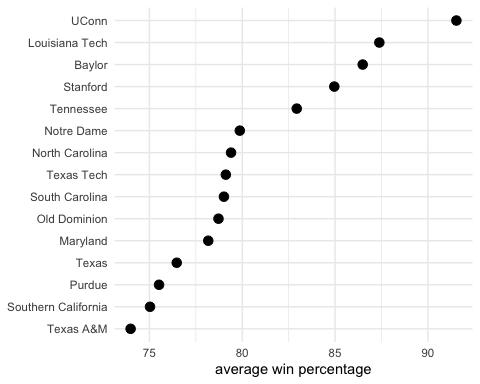
This is not helpful because seed only takes interger values and it’s impossible to tell how many observations occurred at each seed value; this only shows if there was at least one seed set at a given value.

## Question 4

winners\_sts <- winners %>% group\_by(school) %>%   
 summarize\_if(is.numeric, funs(mean, sd), na.rm = TRUE) %>%  
 ungroup()

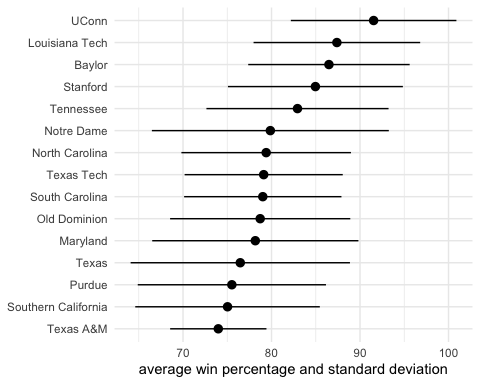
## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(.,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

winners\_sts %>%   
 ggplot(aes(x=reorder(school, reg\_percent\_mean),  
 y=reg\_percent\_mean)) +  
 geom\_point(size=3) +  
 coord\_flip() +  
 theme\_minimal() +  
 labs(x=NULL,   
 y="average win percentage")



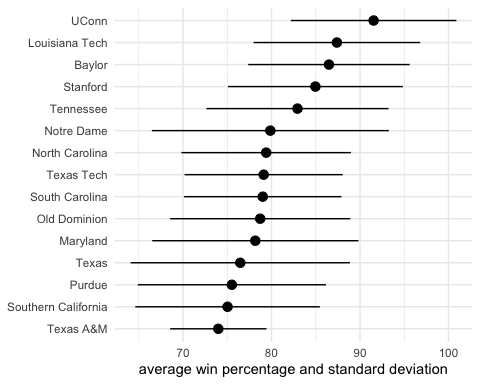
Interesting that while UConn remains in the top two in terms of average win percentage, Tennessee is not in the top two based on this statistic.

winners\_sts %>%   
 ggplot(aes(x=reorder(school, reg\_percent\_mean),  
 y=reg\_percent\_mean)) +  
 geom\_pointrange(aes(ymin=reg\_percent\_mean-reg\_percent\_sd,  
 ymax=reg\_percent\_mean+reg\_percent\_sd)) +  
 coord\_flip() +  
 theme\_minimal() +  
 labs(x=NULL,   
 y="average win percentage and standard deviation")



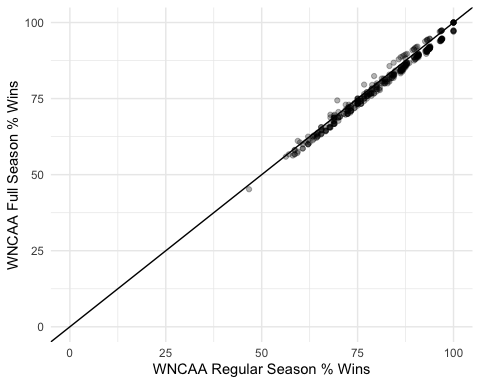
Texas A&M has the most narrow interval.

p <- winners\_sts %>%   
 ggplot(aes(x=reorder(school, reg\_percent\_mean),  
 y=reg\_percent\_mean)) +  
 geom\_point(size=3)  
  
p + geom\_linerange(aes(ymin=reg\_percent\_mean-reg\_percent\_sd,  
 ymax=reg\_percent\_mean+reg\_percent\_sd)) +  
 coord\_flip() +  
 theme\_minimal() +  
 labs(x=NULL,   
 y="average win percentage and standard deviation")



## Question 5

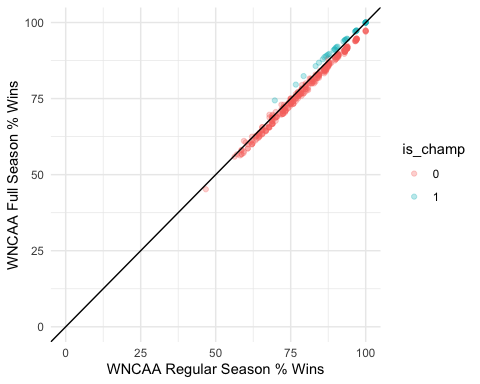
winners %>% ggplot(aes(x=reg\_percent,  
 y=full\_percent)) +  
 geom\_point(alpha=.3) +  
 geom\_abline() +  
 theme\_minimal() +  
 xlim(0,100) +  
 ylim(0,100) +  
 labs(x="WNCAA Regular Season % Wins",   
 y="WNCAA Full Season % Wins")



Most dots are below the line as expected. Dots above the line occur mostly at the upper bounds of reg\_percent, so it seems that teams that overperform at tournaments also tend to do quite well during the regular season.

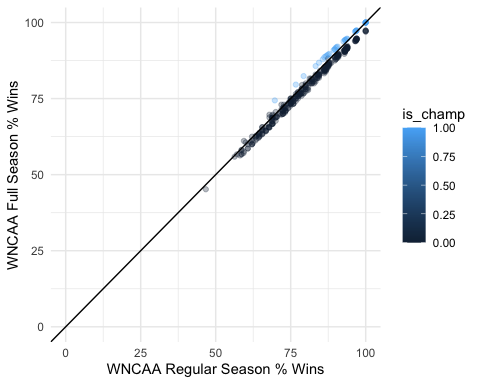
## Question 6

winners <- winners %>%   
 mutate(is\_champ = if\_else(tourney\_finish == "Champ", 1, 0),   
 is\_champ = as.factor(is\_champ))  
winners %>% ggplot(aes(x=reg\_percent,  
 y=full\_percent,  
 color=is\_champ)) +  
 geom\_point(alpha=.3) +  
 geom\_abline() +  
 theme\_minimal() +  
 xlim(0,100) +  
 ylim(0,100) +  
 labs(x="WNCAA Regular Season % Wins",   
 y="WNCAA Full Season % Wins")



Wow, almost all of the dots above the line represent teams that were champions in their tournament that year. This makes sense because they won the tournament which is why their full season percentage of wins is higher than the regular season percentage.

winners <- winners %>%   
 mutate(is\_champ = if\_else(tourney\_finish == "Champ", 1, 0))  
winners %>% ggplot(aes(x=reg\_percent,  
 y=full\_percent,  
 color=is\_champ)) +  
 geom\_point(alpha=.3) +  
 geom\_abline() +  
 theme\_minimal() +  
 xlim(0,100) +  
 ylim(0,100) +   
 labs(x="WNCAA Regular Season % Wins",   
 y="WNCAA Full Season % Wins")

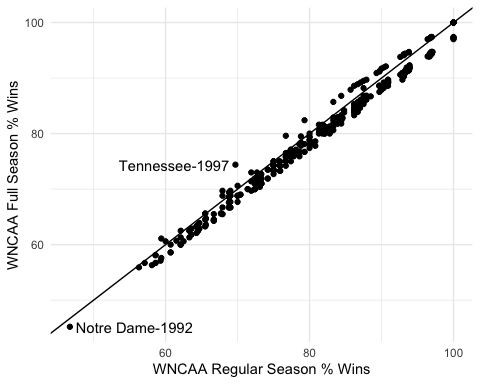


Without knowing that is\_champ is a factor variable, R assumes that it is continuous and colors points with a continuous color scale. You can still see the same patterns but the legend would be confusing for someone else to see because it implies that is\_champ can take on values other than 0 or 1, which is not the case.

## Question 7

winners <- winners %>%   
 mutate(plot\_label = paste(school, year, sep = "-"))  
winners <- winners %>%   
 mutate(difference = full\_percent - reg\_percent)

library(ggrepel)  
winners$plot <- winners$reg\_percent<50 | (winners$reg\_percent<75 & winners$is\_champ==1)  
p1 <- winners %>% ggplot(aes(x=reg\_percent,  
 y=full\_percent))   
p1 + geom\_point() +  
 geom\_text\_repel(data=subset(winners, plot),  
 mapping=aes(label=plot\_label), nudge\_x=-.2) +  
 geom\_abline() +  
 theme\_minimal() +  
 labs(x="WNCAA Regular Season % Wins",   
 y="WNCAA Full Season % Wins")



I have labeled the points of interest, but just those two.

winners %>% filter(reg\_percent==100 & full\_percent==100) %>%   
 dplyr::select (plot\_label)

## # A tibble: 8 × 1  
## plot\_label   
## <chr>   
## 1 Texas-1986   
## 2 UConn-1995   
## 3 UConn-2002   
## 4 UConn-2009   
## 5 UConn-2010   
## 6 Baylor-2012  
## 7 UConn-2014   
## 8 UConn-2016

Most of the observations in which teams have won 100% of regular and fulls season games are UConn, which isn’t surprising because previous questions have shown that they are an extremely strong team. Texas is the most surprising (and to a lesser extent Baylor) because they have a much lower average win percentage and have won a much lower share of tournaments than UConn.