

Intro to R

Part 3: Visualization

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/01/23

Slides Updated: 2023-01-22

Agenda

1. Recap of last lecture

- Using packages: `install.packages()` & `require()`
- Loading and manipulating data: `readRDS()` and `%>%`

2. Plotting in R

- `ggplot` (+ instead of `%>%`)

Loading Packages & Data

- Create an `.Rmd` file and save to your `code` folder
 - Accept defaults, Save As... (with a good name), then `knit`
- Load the `tidyverse` package

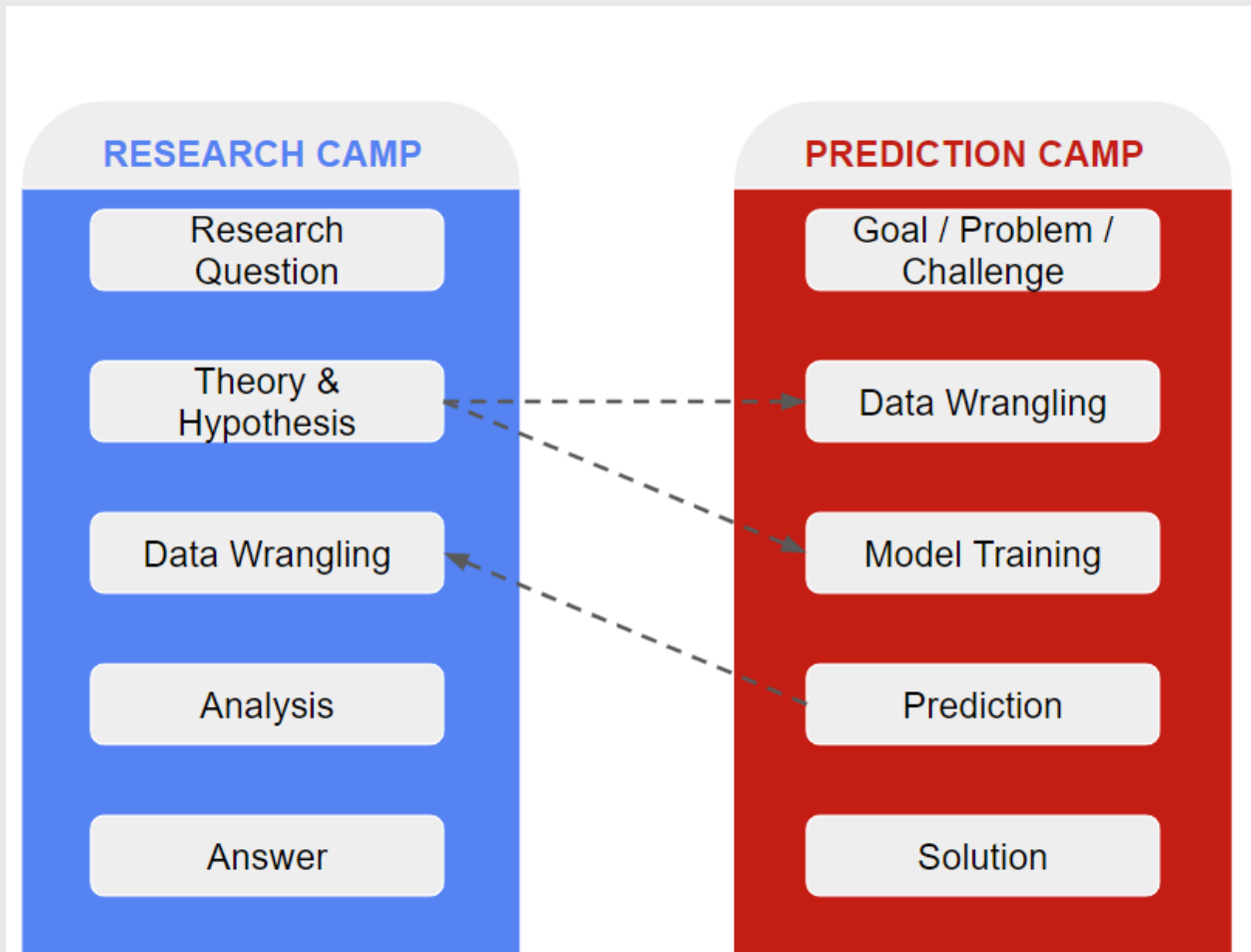
```
require(tidyverse)
```

- Download `sc_debt.Rds` from [GitHub](#) and save to your `./data` folder
- Now load the data with `readRDS("[PATH TO DATA]/sc_debt.Rds")`
 - We **create** an "object" to store the data using a left-arrow: `<-`

```
df <- readRDS("../data/sc_debt.Rds")
```

- NB: `../` means "go up one folder"

The Two Camps



The Research Camp

- RQ: How might admissions and SAT scores be **related**?
 - Theory: selective schools have stricter criteria
 - Hypothesis: admissions and SAT scores should be **negatively** related
- How can we test this hypothesis?

Previously: `summarise()`

- We can combine base R functions with `tidyverse` functions!
 - Base R: `mean()`
 - `tidyverse`: `summarise()` (aka `summarize()`)
- Overall average SAT scores

```
df %>%  
  summarise(mean_sat = mean(sat_avg, na.rm=T))
```

```
## # A tibble: 1 × 1  
##   mean_sat  
##   <dbl>  
## 1    1141.
```

Previously: `summarise()`

- Let's unpack this

```
df %>%  
  summarise(mean_sat = mean(sat_avg, na.rm=T))
```

- Create new variable `mean_sat` that contains the `mean()` of every school's average SAT score
- `na.rm=T` means we want to ignore missing data. If not?

```
df %>%  
  summarise(mean_sat = mean(sat_avg))
```

```
## # A tibble: 1 × 1  
##   mean_sat  
##   <dbl>  
## 1      NA
```

summarise() + filter()

- Recall we want see if more selective schools have higher SAT scores

```
df %>%  
  filter(adm_rate < .1) %>%  
  summarise(mean_sat_LT10 = mean(sat_avg, na.rm=T))
```

```
## # A tibble: 1 × 1  
##   mean_sat_LT10  
##           <dbl>  
## 1          1510.
```

```
df %>%  
  filter(adm_rate > .1 & adm_rate < .2) %>%  
  summarise(mean_sat_1020 = mean(sat_avg, na.rm=T))
```

```
## # A tibble: 1 × 1  
##   mean_sat_1020  
##           <dbl>  
## 1          1424.
```


summarise() + group_by()

- One final `tidyverse` function: `group_by()`

```
df %>%  
  group_by(selective) %>%  
  summarise(mean_sat = mean(sat_avg, na.rm=T))
```

```
## # A tibble: 3 × 2  
##   selective mean_sat  
##   <dbl>     <dbl>  
## 1      0     1135.  
## 2      1     1510.  
## 3     NA      NaN
```

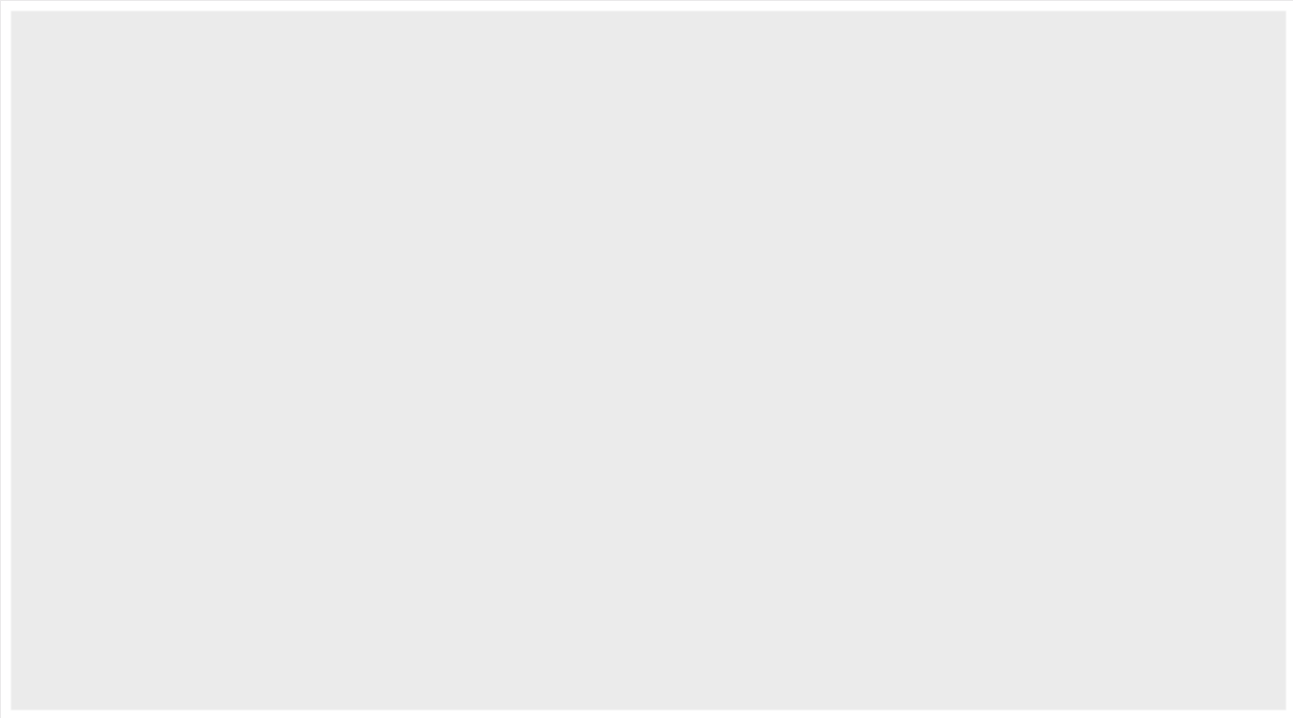
Plotting data

- Let's plot the data instead of writing many of these `summarise()` functions
- Visualization in R uses `ggplot()` function
 - Inputs: `aes(x,y,...)` (elipses `...` indicates many more inputs)
 - `x` is the x-axis (horizontal)
 - `y` is the y-axis (vertical)

ggplot()

- Attach `ggplot()` to your data with `%>%`

```
df %>%  
  ggplot()
```

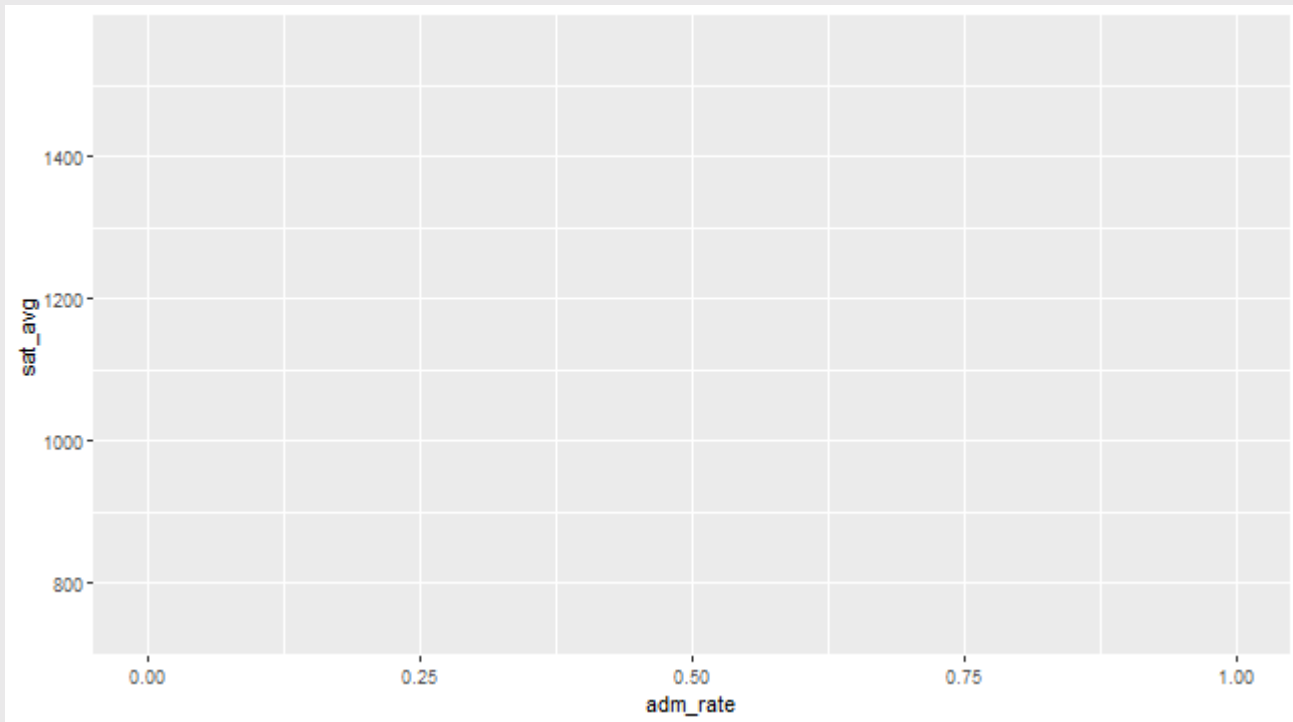


ggplot()

- Then tell it what to put in the x-axis and y-axis
- What should go on these axes?
- **Theory**: Selective schools choose higher scoring students
 - Selective schools **explain** higher scores
 - Selective schools: **independent variable** / **explanatory variable** / **predictor** / X
 - Higher scores: **dependent variable** / **outcome variable** / Y
- Selective schools go on the x-axis, SAT scores go on the y-axis

ggplot()

```
df %>%  
  ggplot(aes(x = adm_rate, y = sat_avg))
```



ggplot()

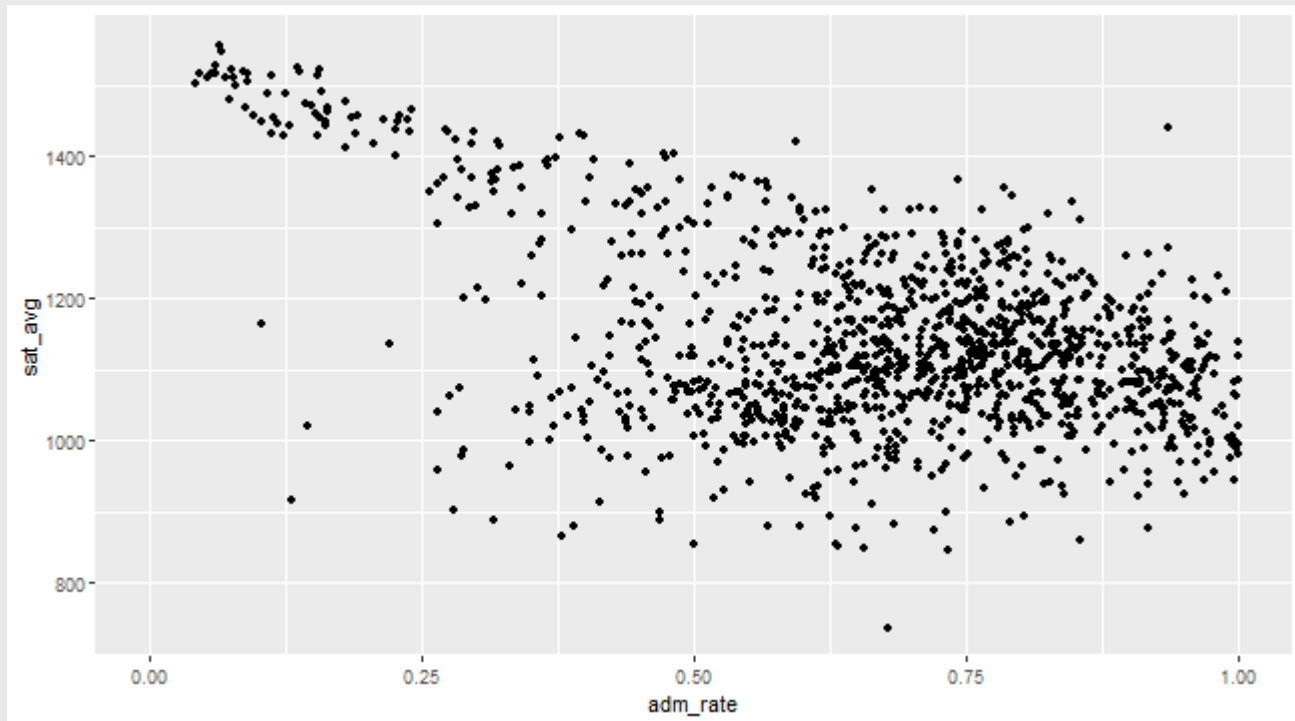
- This gives us an empty plot
- We have the correct variables on the correct axes...
- ...but we need to choose how to display them
- There are many different `ggplot()` functions to choose from
 - `geom_point()` creates one point for each x and y coordinate
 - `geom_bar()` creates a barplot
 - `geom_histogram()` creates a histogram
 - `geom_density()` creates a density plot
 - `geom_boxplot()` creates a box-and-whisker plot

ggplot()

- We **add** a second `ggplot()` function to the first with a plus sign +
 - **NB:** This is JUST LIKE THE PIPE OPERATOR `%>%` in `tidyverse`!
- Since `adm_rate` (the x-axis variable) and `sat_avg` (the y-axis variable) are both numeric ("continuous") measures, we will use `geom_point()`
 - We will come back to **variable types** and how to visualize them later

ggplot()

```
df %>%  
  ggplot(aes(x = adm_rate, y = sat_avg)) +  
  geom_point()
```



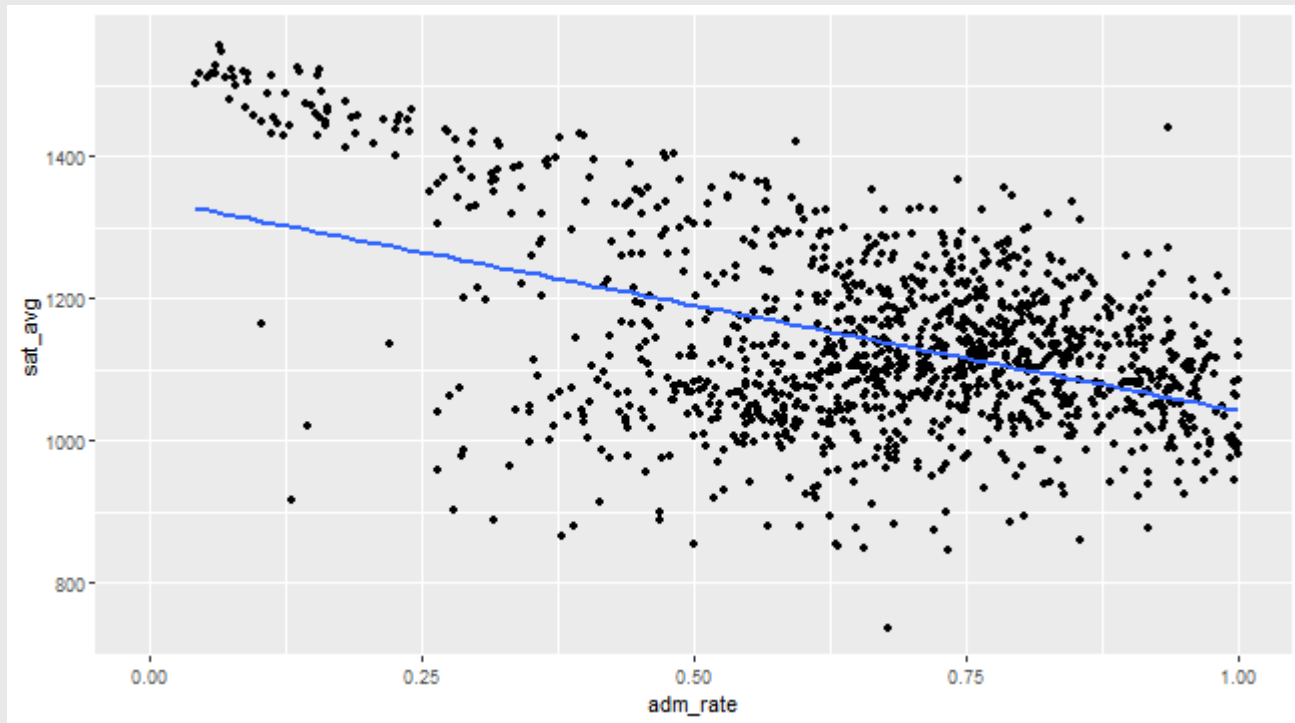
Plotting data

- Let's unpack this
 - `aes(x,y)` sets the basic aesthetics for the plot
 - `geom_point()` tells `ggplot()` how to visualize those aesthetics
 - These two parts are linked with the `+`. Similar to...?
 - ...the `%>%` in `tidyverse`!

Interpreting the plot

- We **hypothesized** that admissions and SAT scores are negatively related
 - Is this supported in the data?
- Let's add a line of best fit with `geom_smooth()`

```
df %>%  
  ggplot(aes(x = adm_rate, y = sat_avg)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = F)
```



The Research Camp

- **RQ**: How might future earnings and SAT scores be **related**?
 - **Theory**: SATs measure student ability.
 - **Theory**: Student ability is valued by the labor market.
 - **Theory**: Firms pay more for students with higher SAT scores.
 - **Hypothesis**: Earnings and SAT scores should be **positively** related

Plotting Quiz

- Which variable goes on the x-axis?
 - **SAT scores**
- Which variable goes on the y-axis?
 - **Earnings**
- In our theory, SAT scores **cause** earnings
- Why might this **not** be the case?
 - Spurious 1: SAT scores **and** earnings are caused by student ability
 - Spurious 2: SAT scores **and** earnings are caused by socio-economic privilege

Let's Plot!

```
df %>%  
  ggplot(aes(x = sat_avg, y = md_earn_wne_p6)) + # Build axes  
  geom_point() + # Add points  
  geom_smooth(method = 'lm', se = F) # Add line of best fit
```

Outliers

- Which schools are furthest from the line?
 - These are **outliers**
 - These schools are the **furthest** from our [theory](#)

```
df %>%  
  mutate(out = ifelse(md_earn_wne_p6 > 100000,  
                      instnm, # Value if TRUE  
                      NA)) %>% # Value if FALSE  
  drop_na(out,sat_avg) %>%  
  select(instnm,md_earn_wne_p6,sat_avg)
```

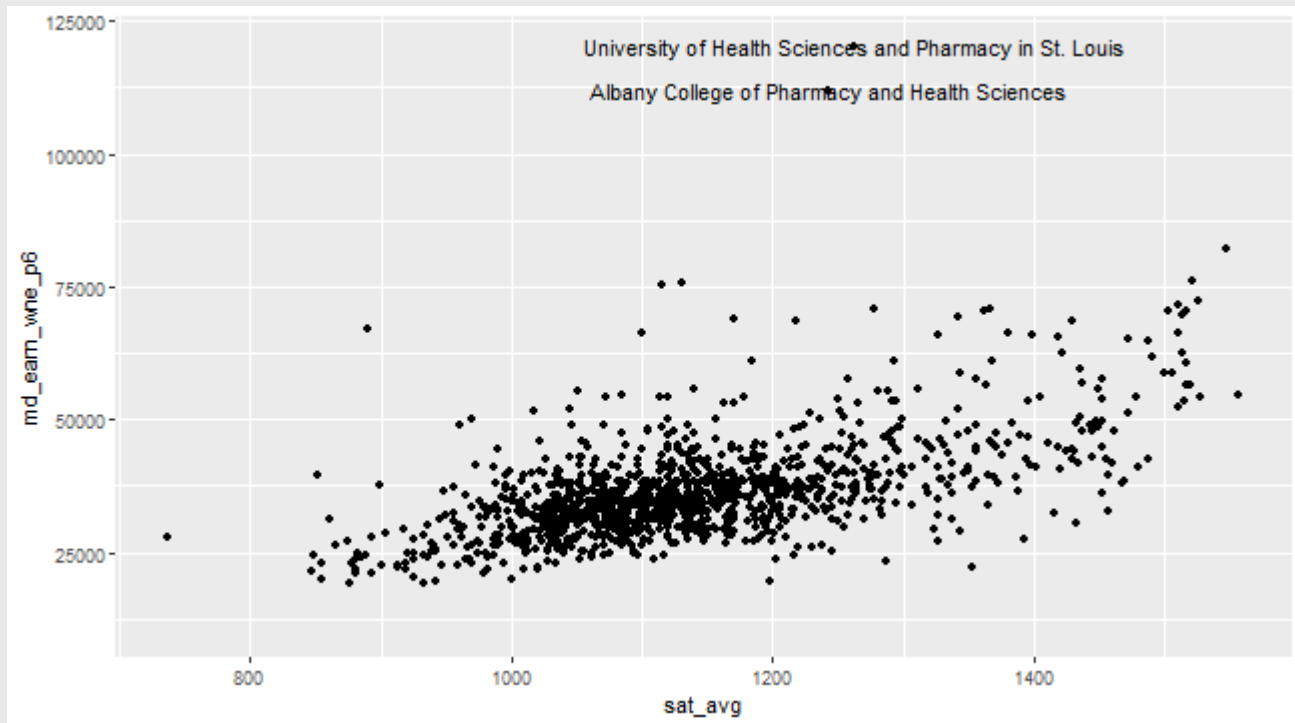
```
## # A tibble: 2 × 3  
##   instnm                                md_ea...1 sat_avg  
##   <chr>                                <int>    <int>  
## 1 University of Health Sciences and Pharmac... 120400    1262  
## 2 Albany College of Pharmacy and Health Sci... 112100    1242  
## # ... with abbreviated variable name 1md_earn_wne_p6
```

Plotting data

- We can add these as labels!

```
df %>%  
  mutate(out = ifelse(md_earn_wne_p6 > 100000,  
                      instnm, # Value if TRUE  
                      NA)) %>% # Value if FALSE  
  ggplot(aes(x = sat_avg, y = md_earn_wne_p6,  
             label = out)) +  
  geom_point() +  
  geom_text()
```


Plotting data

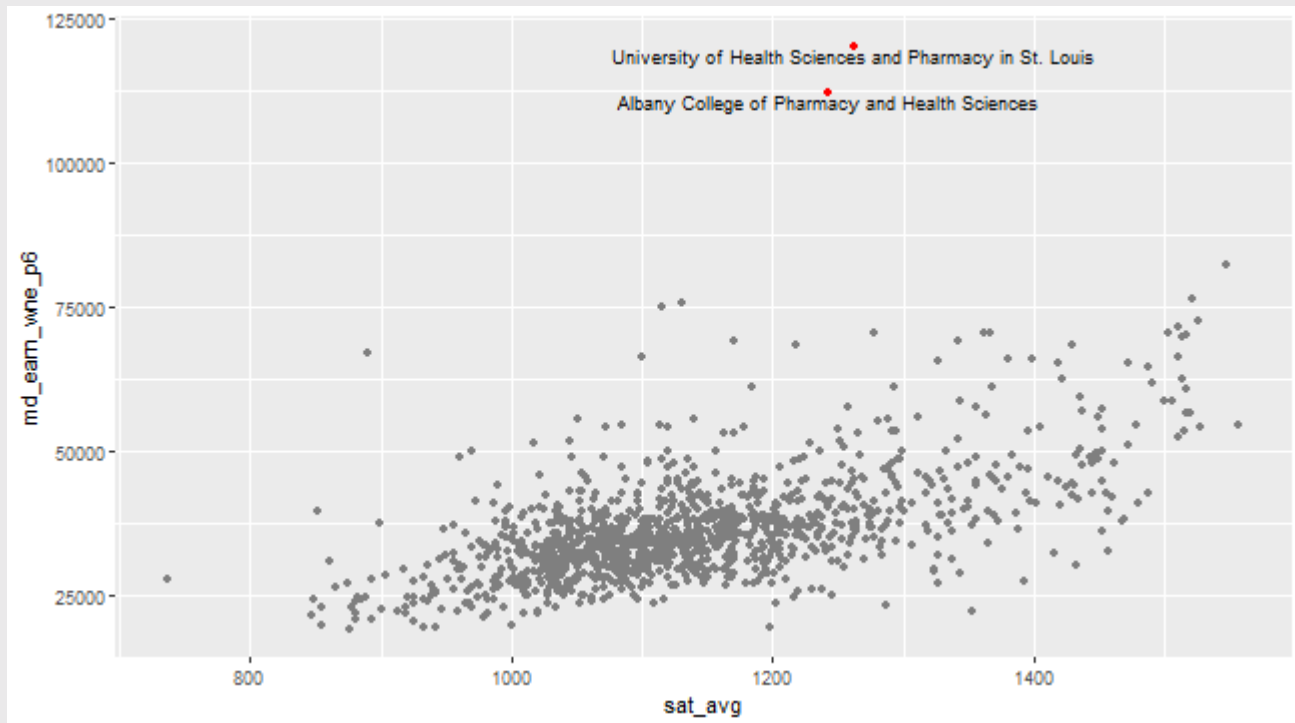


Plotting data

- Let's accentuate the outlier more with color

```
p <- df %>%  
  mutate(out = ifelse(md_earn_wne_p6 > 100000,  
                      instnm, # Value if TRUE  
                      NA)) %>% # Value if FALSE  
  
  drop_na(sat_avg) %>%  
  ggplot(aes(x = sat_avg, y = md_earn_wne_p6,  
            label = out, color = out)) +  
  geom_point() +  
  scale_color_manual(name = "Outlier", values =  
c('red', 'red', 'black')) +  
  geom_text(hjust = .5, vjust = 1, color = 'black', size = 3)
```

Plotting data



Categorical Data

- Thus far, plotting two continuous variables with `geom_point()`
- What if we wanted to see which state has the most selective schools?
- Use `group_by()` and `summarise()`

```
df %>%  
  group_by(stabbr) %>%  
  summarise(selective_avg = mean(adm_rate, na.rm=T))
```

```
## # A tibble: 51 × 2  
##   stabbr selective_avg  
##   <chr>         <dbl>  
## 1 AK           0.827  
## 2 AL           0.654  
## 3 AR           0.676  
## 4 AZ           0.843  
## 5 CA           0.592  
## 6 CO           0.768  
## 7 CT           0.589  
## 8 DC           0.529  
## 9 DE           0.627
```

Categorical Data

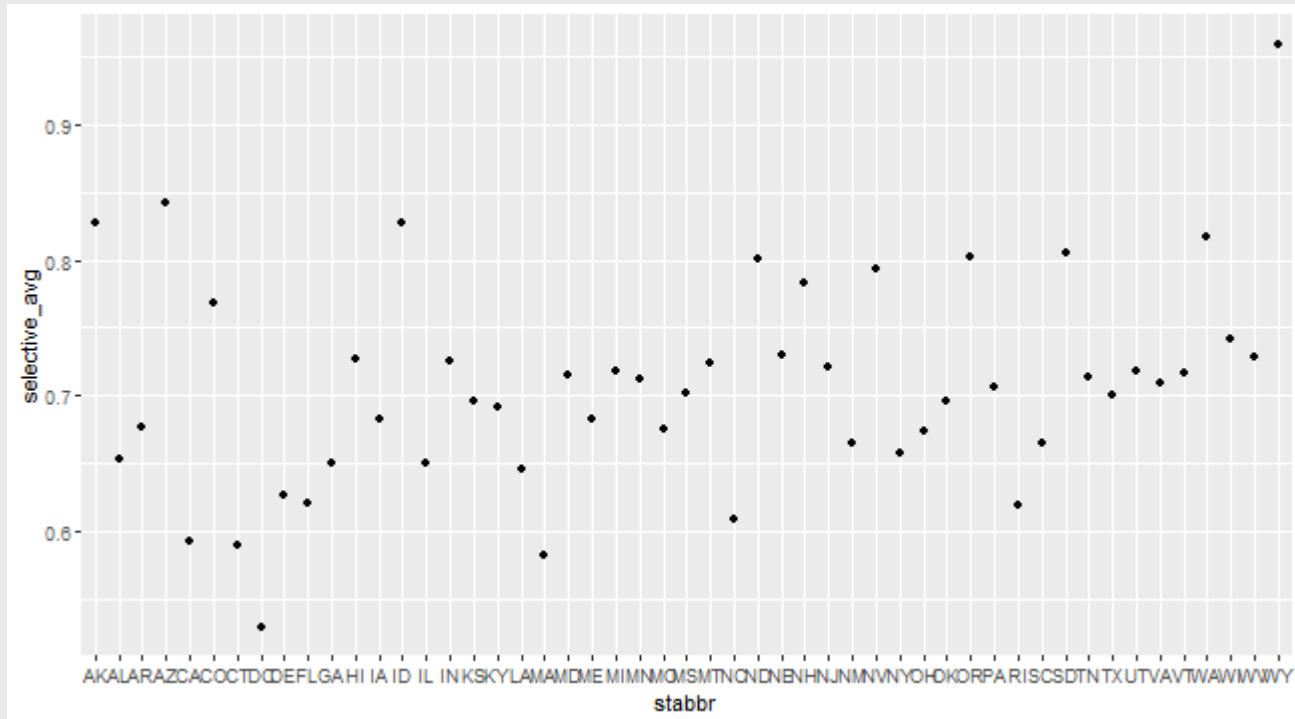
- Two variables (`stabbr` and `selective_avg`), but one of them is now a `character` type
- Can we plot this as a scatterplot?

```
p <- df %>%  
  group_by(stabbr) %>%  
  summarise(selective_avg = mean(adm_rate, na.rm=T)) %>%  
  ggplot(aes(x = stabbr, y = selective_avg)) +  
  geom_point()
```

Categorical Data

- Yes...but it isn't very pretty

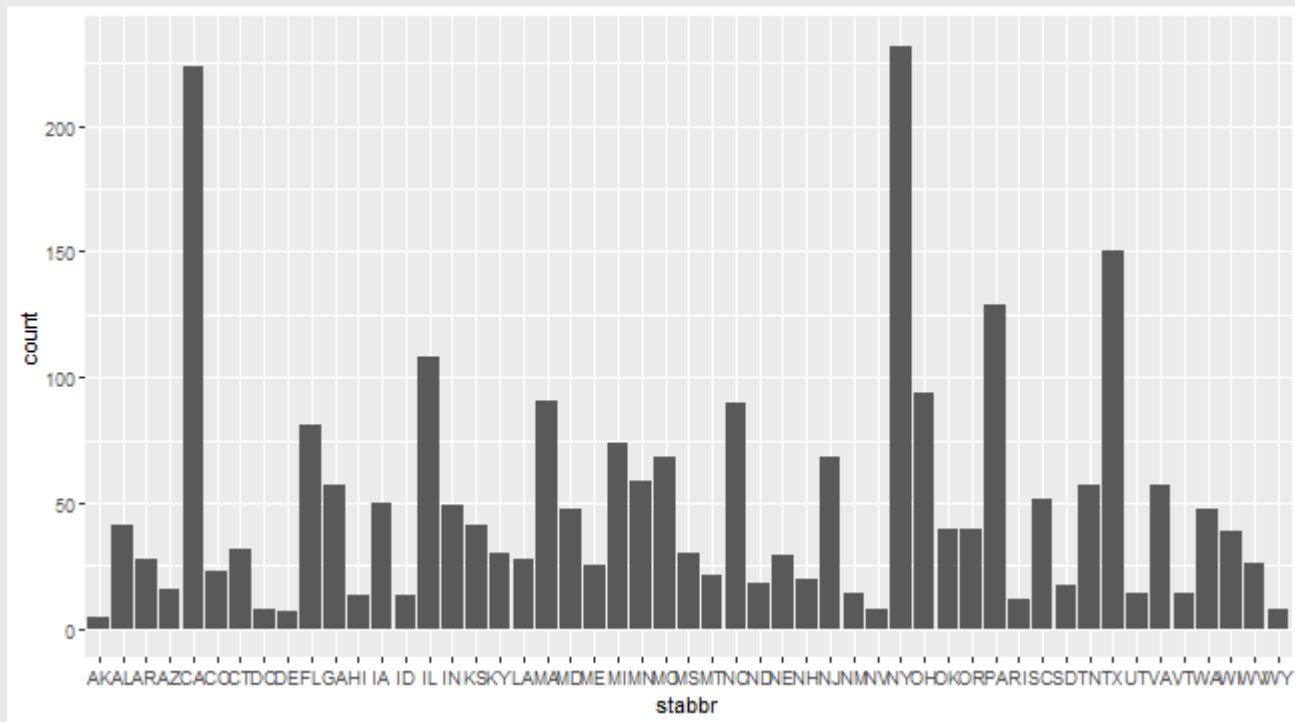
p



Categorical Data: `geom_bar()`

- NB: `geom_bar()` will automatically count the values on the x-axis

```
df %>%  
  ggplot(aes(x = stabbr)) +  
  geom_bar()
```



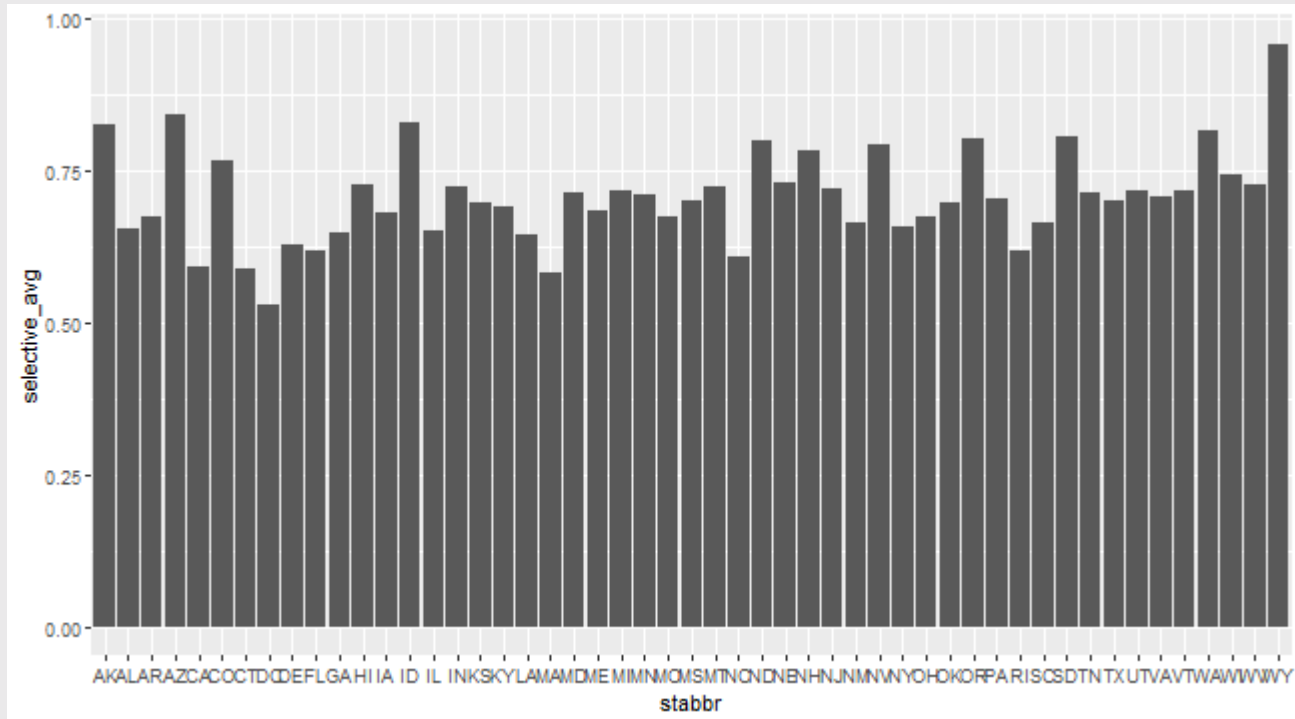
Categorical Data: `geom_bar()`

- This is fine if we just want to know which states have the most schools in our data
- But we want to put the average admissions rate on the y-axis instead
 - Need to **override** `geom_bar()` default behavior

```
p <- df %>%  
  group_by(stabbr) %>%  
  summarise(selective_avg = mean(adm_rate, na.rm=T)) %>%  
  ggplot(aes(x = stabbr, y = selective_avg)) +  
  geom_bar(stat = 'identity')
```


Categorical Data

p



Categorical Data

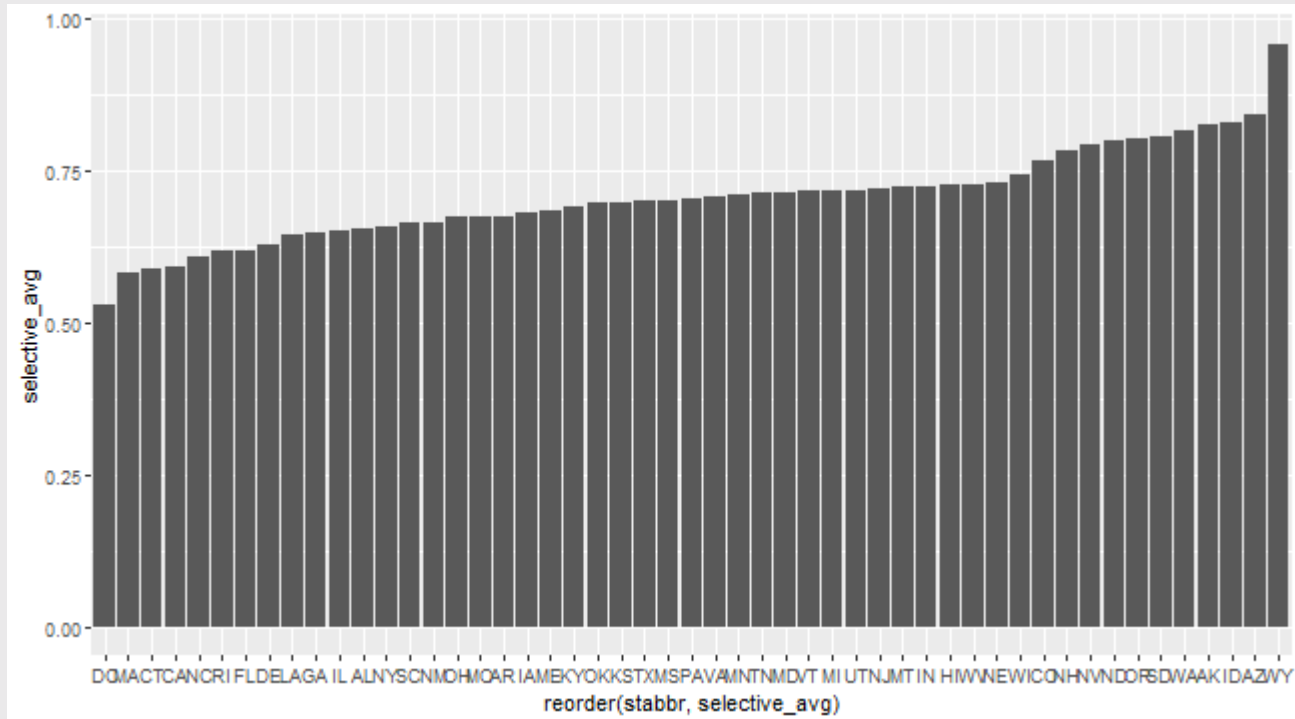
- Getting a little better, but still ugly
- Use `reorder()` to sort the x-axis values by the y-axis

```
p <- df %>%  
  group_by(stabbr) %>%  
  summarise(selective_avg = mean(adm_rate, na.rm=T)) %>%  
  ggplot(aes(x = reorder(stabbr, selective_avg), y = selective_avg)) +  
  geom_bar(stat = 'identity')
```

Categorical Data

- Even better!

p



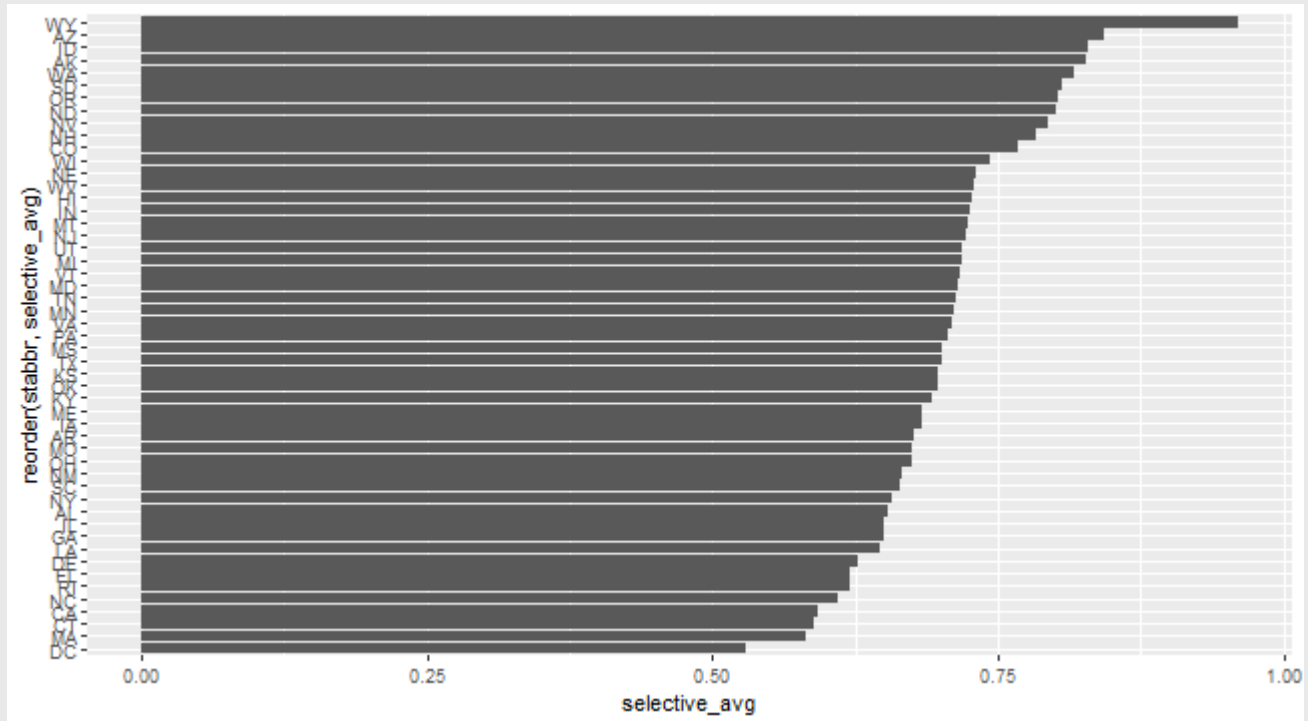
Plot Tweaking

- We could go even further and swap the x and y-axes (although this isn't always a good idea!)

```
p <- df %>%  
  group_by(stabbr) %>%  
  summarise(selective_avg = mean(adm_rate,na.rm=T)) %>%  
  ggplot(aes(y = reorder(stabbr,selective_avg),x = selective_avg)) +  
  geom_bar(stat = 'identity')
```

Plot Tweaking

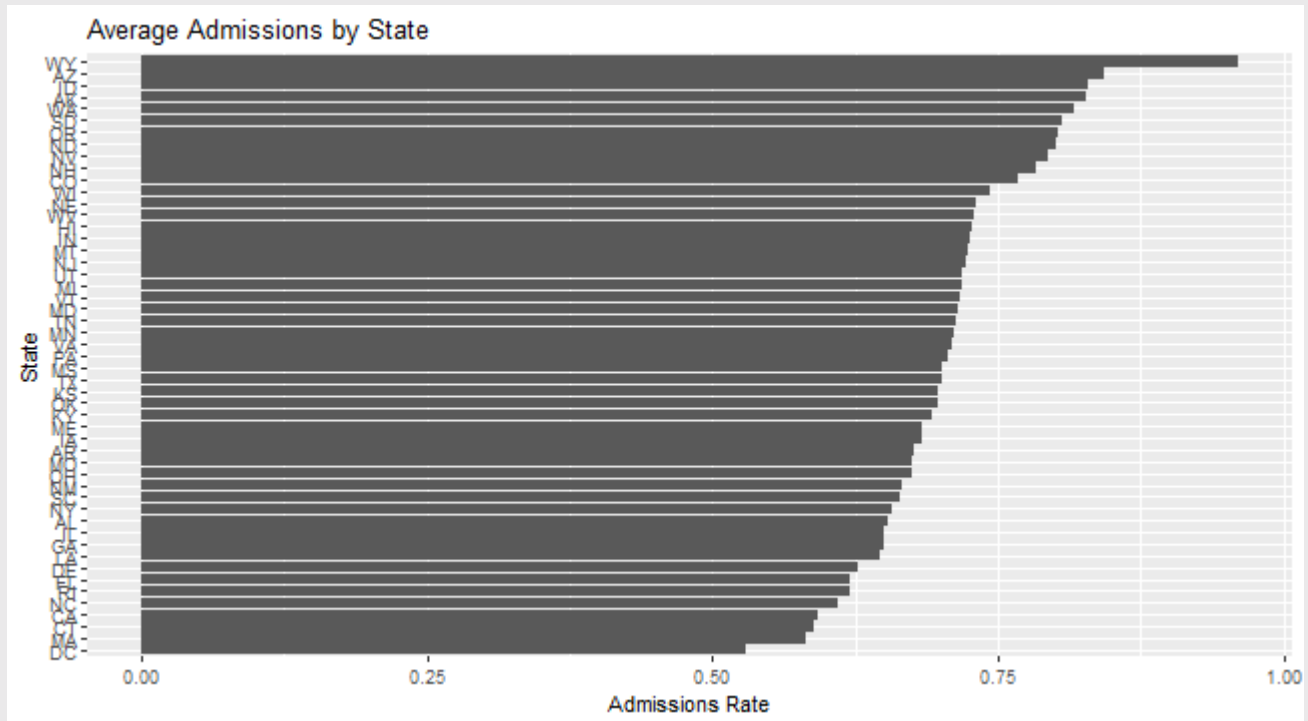
p



- Still ugly though! We want to tweak the labels with `labs()`

Plot Tweaking

```
p +  
  labs(title = "Average Admissions by State",  
        x = "Admissions Rate",  
        y = "State")
```



Conclusion

- What to take away
 1. Which variables go on which axes
 2. How to put these on a `ggplot()` figure
 3. How to create a visualization of these variables
- This wraps up the crash course in `R`
 - **REMEMBER:** This class is *inherently* challenging because of `R`
 - The course is graded leniently to reflect the inherent difficulty of the material

Quiz & Homework

- Go to Brightspace and take the fourth quiz
 - The password to take the quiz is ####
- **Homework:**
 1. Work through Intro_to_R_Part3_hw.Rmd
 2. Complete Problem Set 1