Intro to R

Part 3: Visualization

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/01/23

Slides Updated: 2023-01-03

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

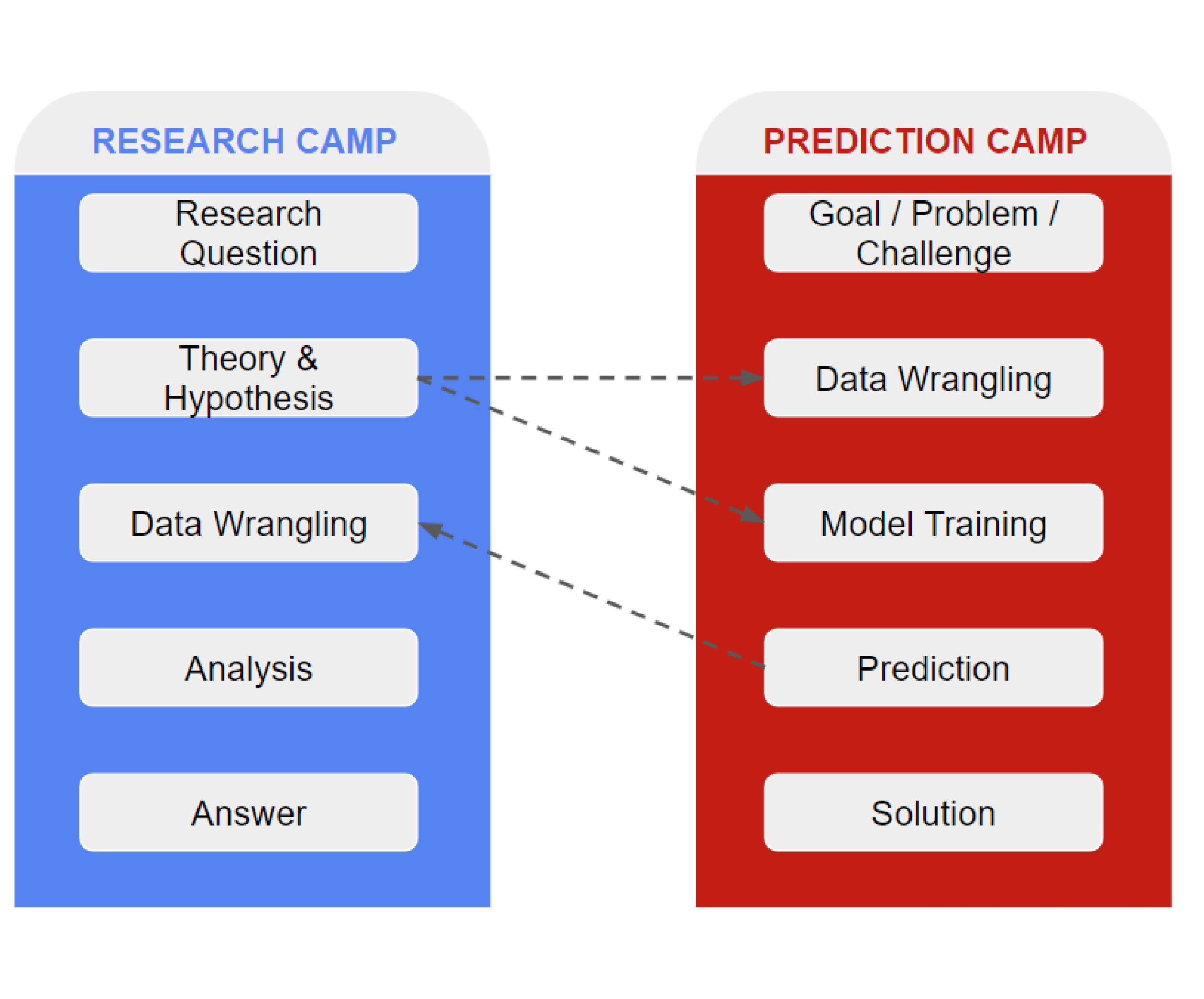
```r
require(tidyverse)
```

- 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: `<-`

```r
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
```

Previously: `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
```

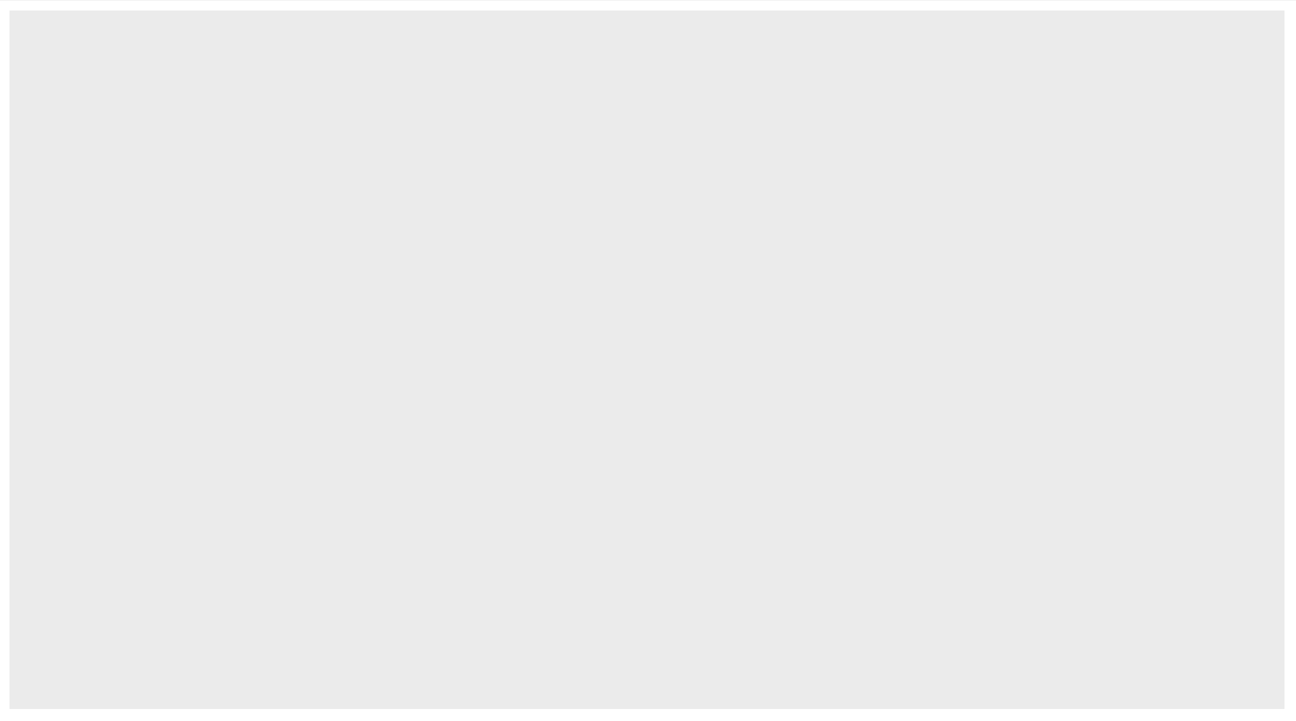
Plotting data

- 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()
```

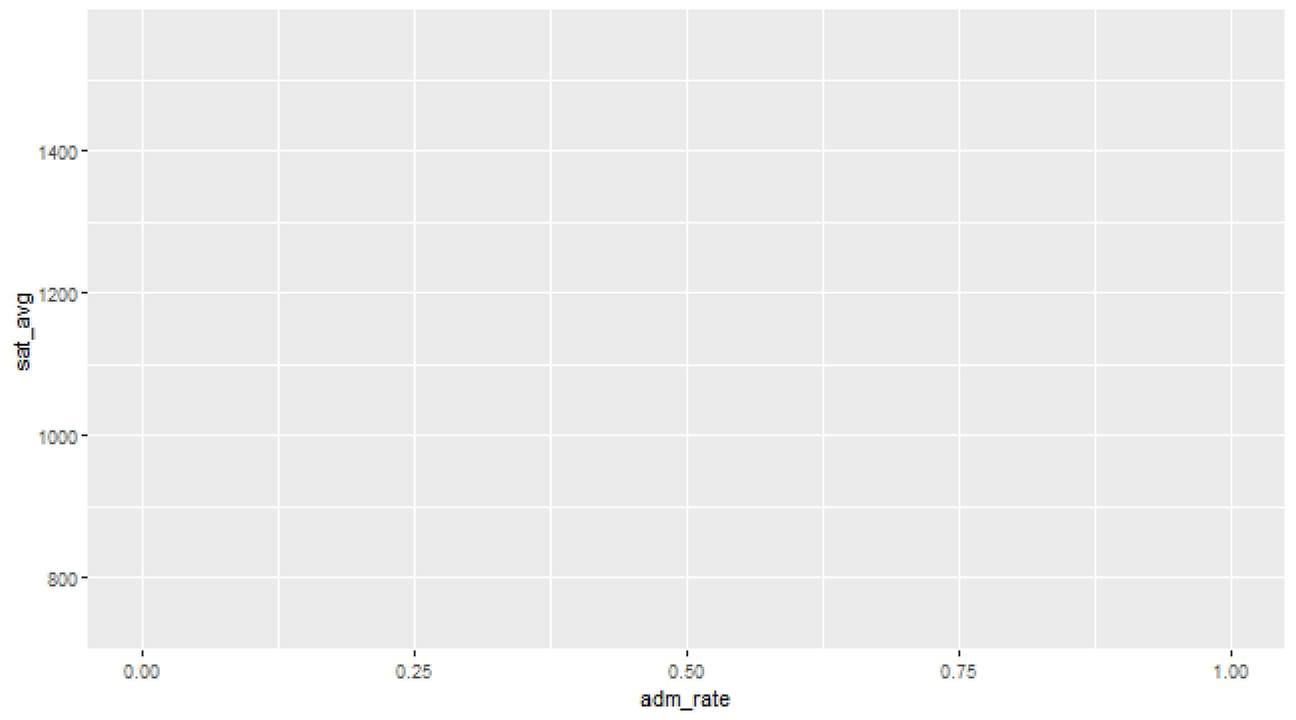


`ggplot()`

- 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()`

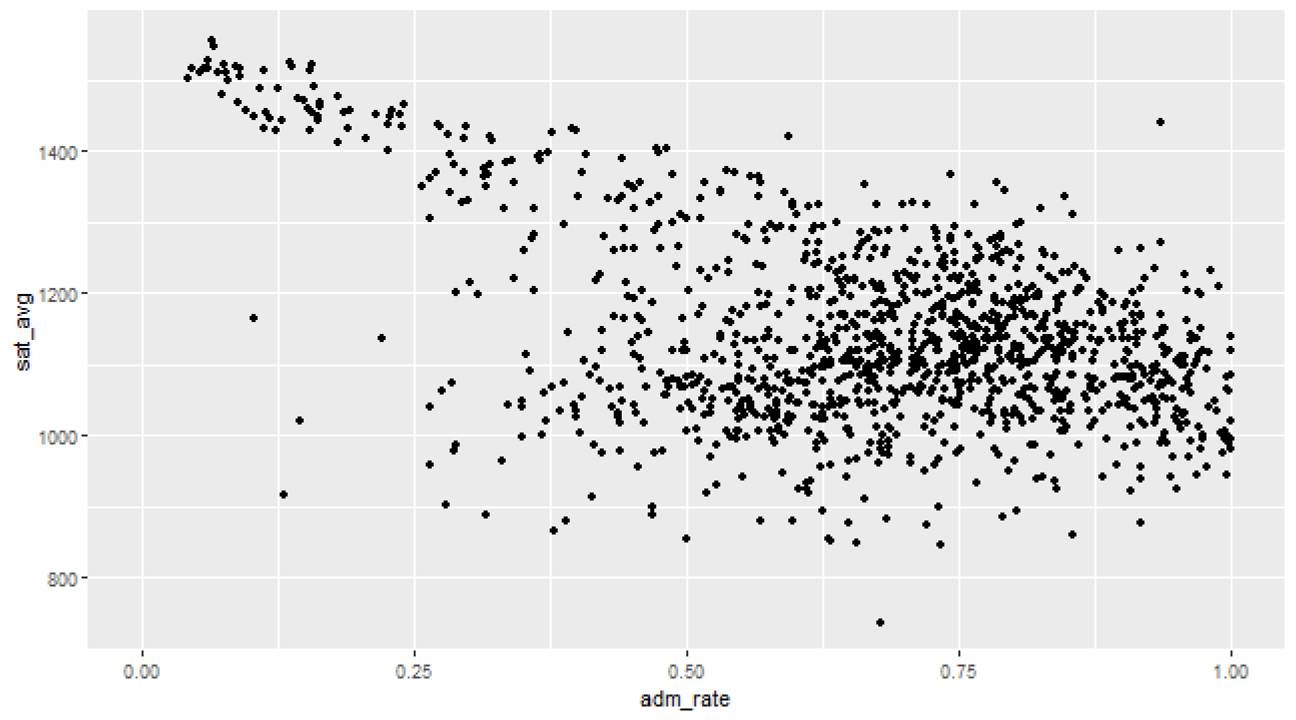
- This gives us an empty plot
- We have the correct variables on the correct axes...
- ...but we need to choose how to display them
- There are many different `ggplot()` functions to choose from
  - `geom_point()` creates one point for each x and y coordinate
  - `geom_bar()` creates a barplot
  - `geom_histogram()`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

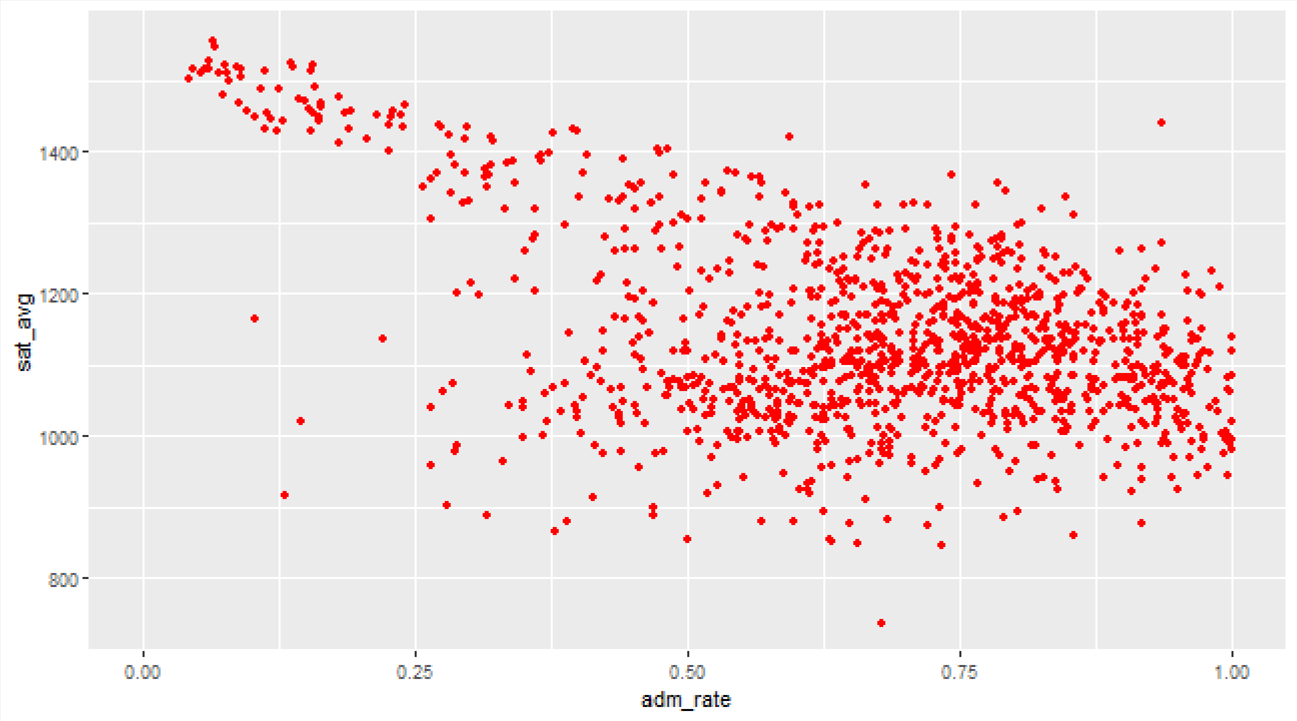


Plotting data

- 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`!
  - We can force aesthetics by setting code outside the `aes()`

```
df %>%
  ggplot(aes(x = adm_rate_pct,y = sat_avg)) +
  geom_point(color = 'red')
```

Plotting data

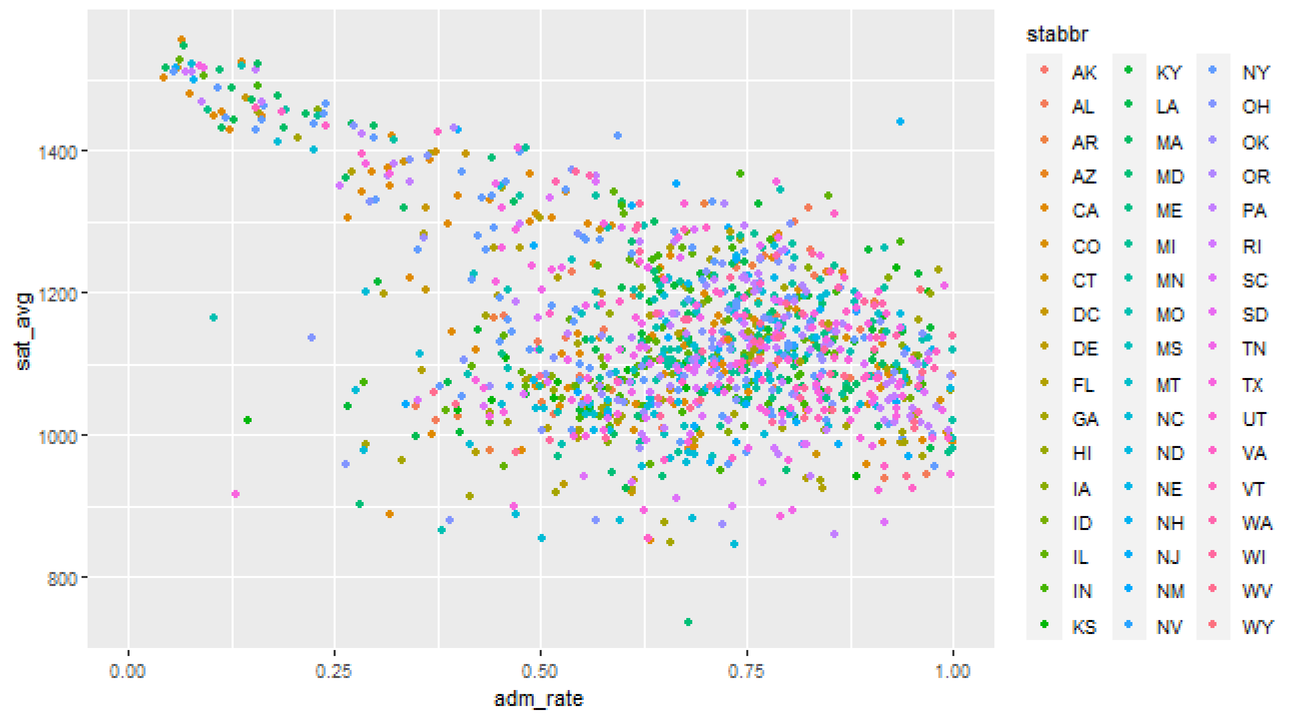


Plotting data

- Or we can make more aesthetics dependent on the data

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

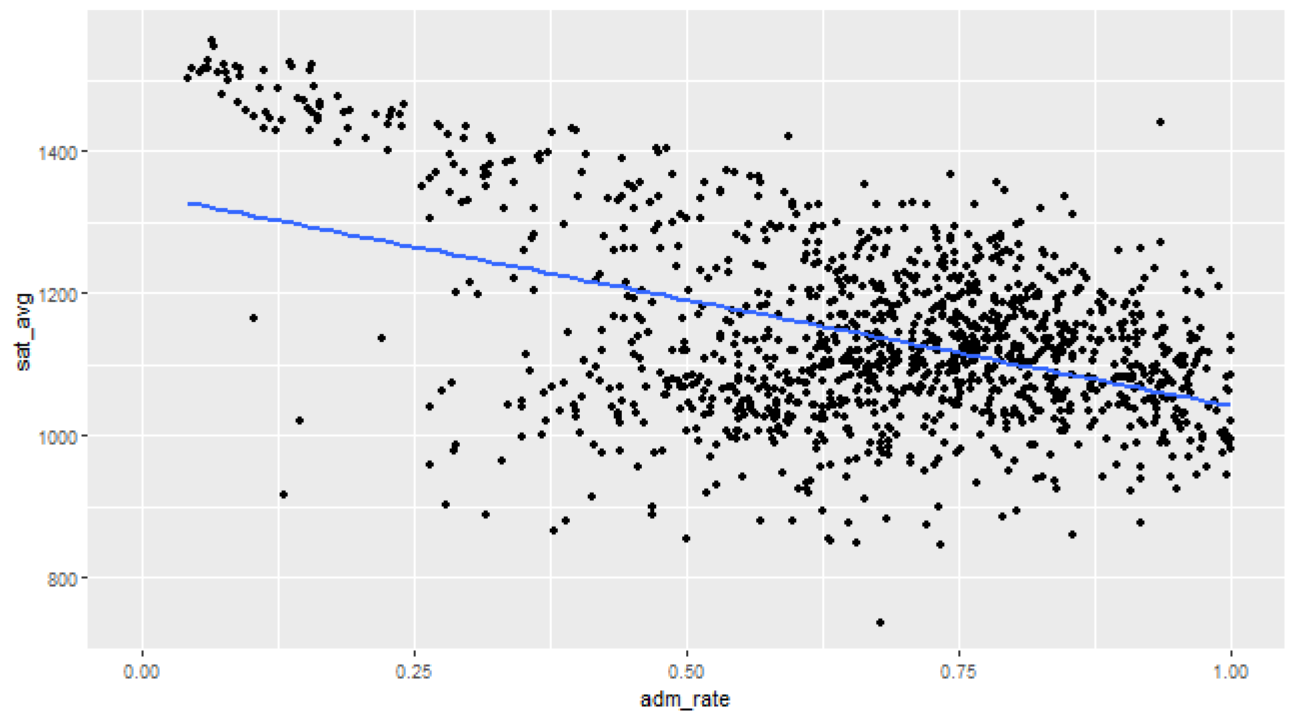
Plotting data



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)
```


1400
1200
1000
800
sat_avg
0.00
0.25
0.50
0.75
1.00
adm_rate

Plotting data

- Which school is most selective but also with the lowest SAT?
  - This is an **outlier**
  - This school is the **furthest** from our theory

```r
df %>%
  mutate(out = ifelse(adm_rate < .25 & sat_avg < 1000,
                          instnm,  # Value if TRUE
                          NA)) %>% # Value if FALSE
  filter(!is.na(out)) %>%
  select(instnm,adm_rate,sat_avg)
```

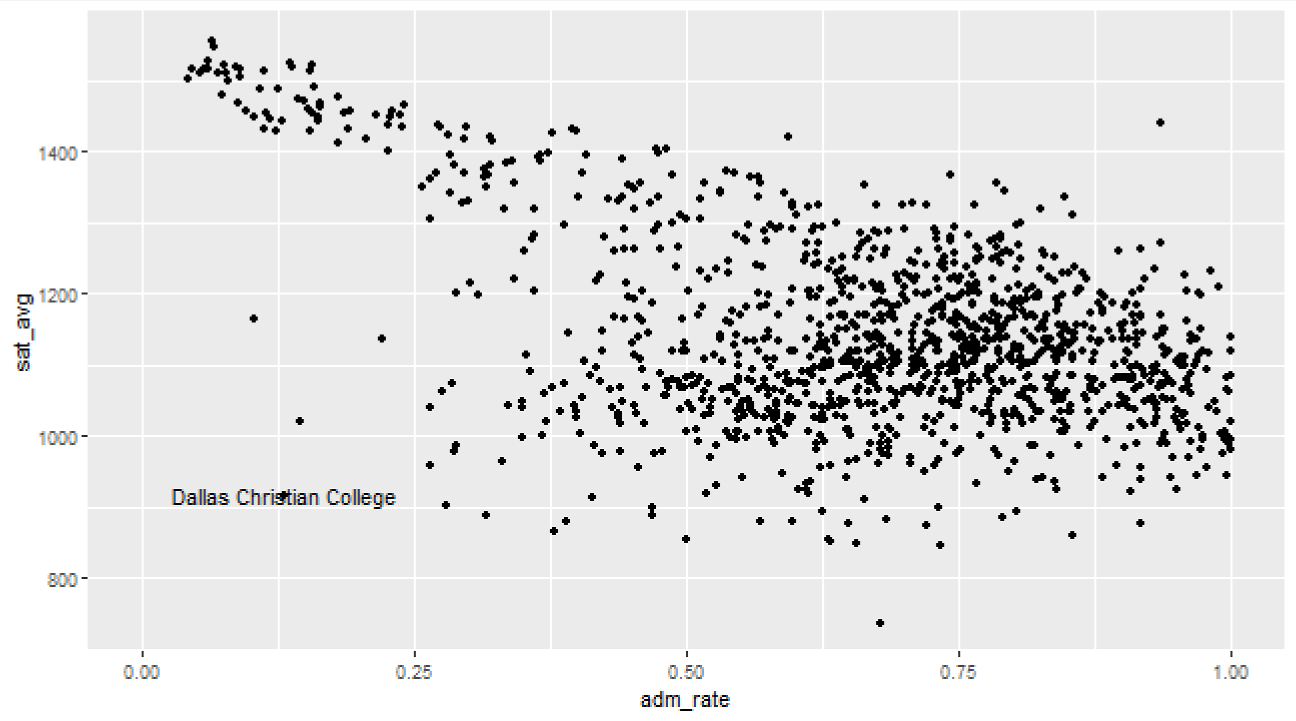
```
## # A tibble: 1 × 3
##   instnm                   adm_rate sat_avg
##   <chr>                       <dbl>   <int>
## 1 Dallas Christian College    0.130     917
```

Plotting data

- We can add this as a label!

```r
df %>%
  mutate(out = ifelse(adm_rate < .25 & sat_avg < 1000,
                      instnm,
                      NA)) %>%
  ggplot(aes(x = adm_rate,y = sat_avg,
             label = out)) +
  geom_point() +
  geom_text()
```

Plotting data

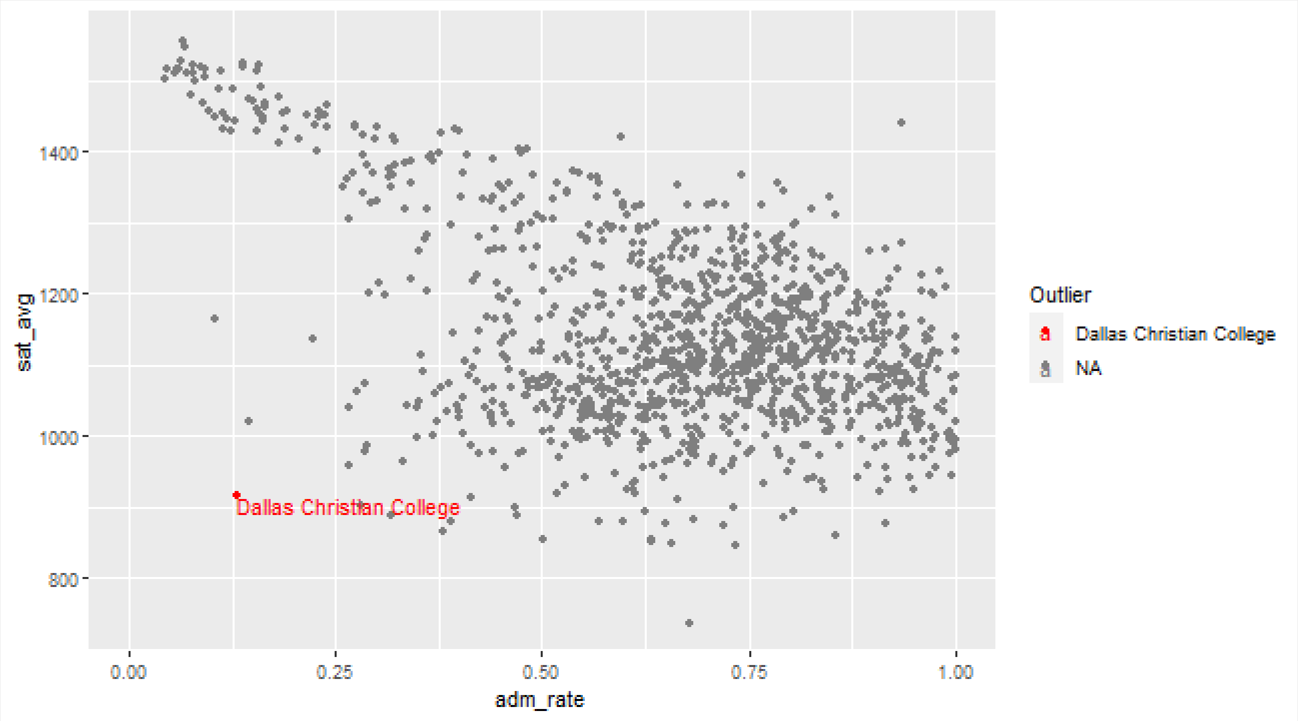


Plotting data

- Let's accentuate the outlier more with color

```
df %>%
  mutate(out = ifelse(adm_rate < .25 & sat_avg < 1000,
                      instnm,
                      NA)) %>%
  ggplot(aes(x = adm_rate,y = sat_avg,
             color = out,label = out)) +
  geom_point() +
  scale_color_manual(name = "Outlier",values = c('red','black')) +
  geom_text(hjust = 0,vjust = 1,color = 'black',size = 3)
```

Plotting data



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 3326
- **Homework:**
  1. Work through Lecture_2023_01_23_hw.Rmd
  2. Complete Problem Set 1