STAT 102: Week 3

Ricky's Section

Introductions and Attendance

Introduction: Name

Question of the Week: What is one word to describe how you're feeling? Try not to repeat words!

Important Reminders

Join Our Slack: #section-d001

Content Review: Week 2

Grammar of Graphics

https://drive.google.com/file/d/1y13AW7qNlvMnHGtNv1FspjUb4Vfr9Ohn/view?usp=drive_link

Choosing the Right Graph

https://drive.google.com/file/d/1GlfYFuUYMPxMgnvBtojxzQn6MGh5yl-H/view?usp=drive_link

Coding

- Make sure your code isn't running off the screen in your PDF
- Hit "Return" to start a new line
  - Best to do this after commas and plus signs

```
## 12 1      United Talmudical Seminary                    21200
```

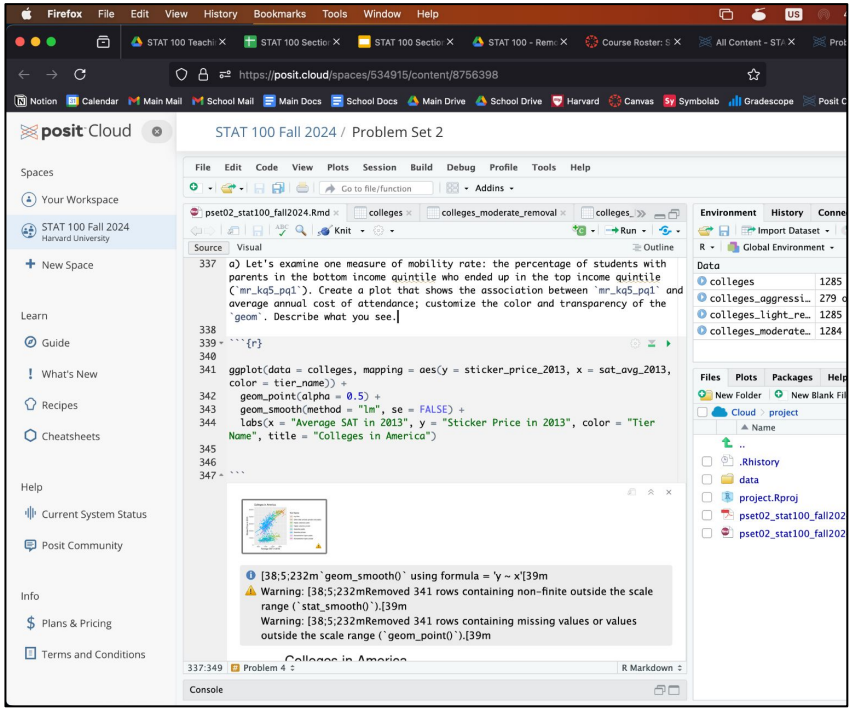
Problem 4

a) Let's examine one measure of mobility rate: the percentage of students with parents in the bottom income quintile who ended up in the top income quintile (`mr_kq5_pq1`). Create a plot that shows the association between `mr_kq5_pq1` and average annual cost of attendance; customize the color and transparency of the `geom`. Describe what you see.

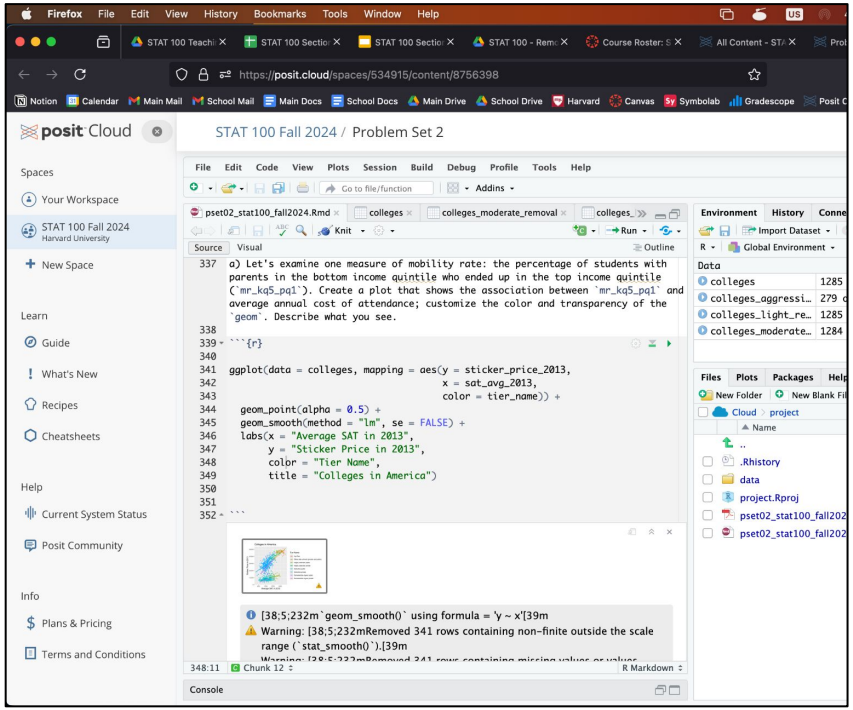
```
ggplot(data = colleges, mapping = aes(y = sticker_price_2013, x = sat_avg_2013, color =
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Average SAT in 2013", y = "Sticker Price in 2013", color = "Tier Name", titl
```

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

## Warning: Removed 341 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 341 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Messy Code



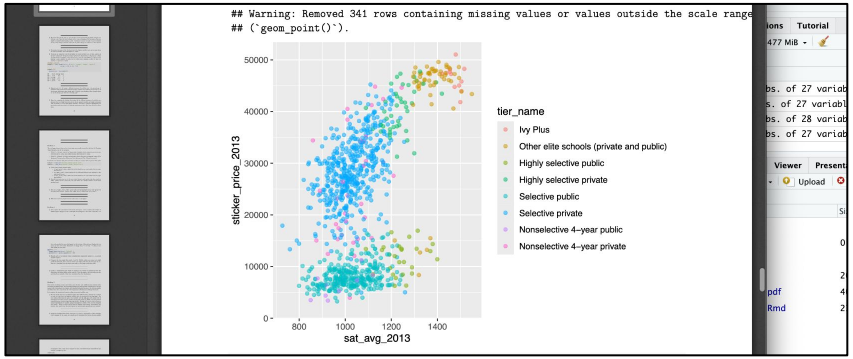
Clean Code



Reading Code in English

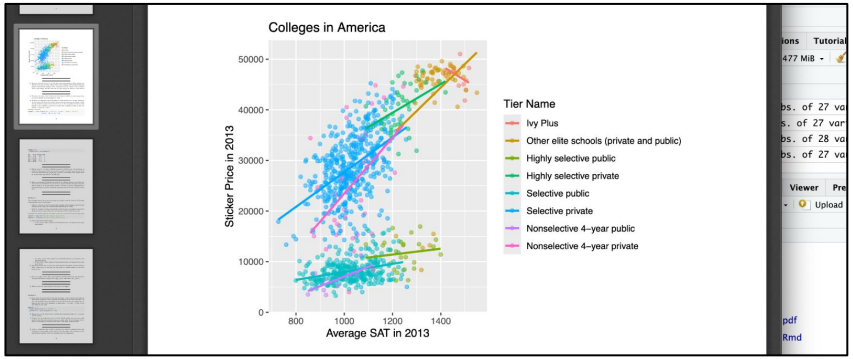
In R: ggplot(data = colleges, mapping = aes(y = sticker_price_2013, x = sat_avg_2013, color = tier_name)) + geom_point(alpha = 0.5)

In English: Create a **plot** using the "colleges" **dataset**. Map to the **aesthetic** of **y-direction** the **variable** of sticker price, map to the **aesthetic** of **x-direction** the **variable** of average SAT, and map to the **aesthetic** of **color** the **variable** tier name. Use **points** as the **geom**, and make **alpha/transparency** equal to 0.5.



Coding: In Words

In R: ggplot(data = colleges, mapping = aes(y = sticker_price_2013, x = sat_avg_2013, color = tier_name)) + geom_point(alpha = 0.5) + geom_smooth(method = "lm", se = FALSE) + labs(x = "Average SAT in 2013", y = "Sticker Price in 2013", color = "Tier Name", title = "Colleges in America")



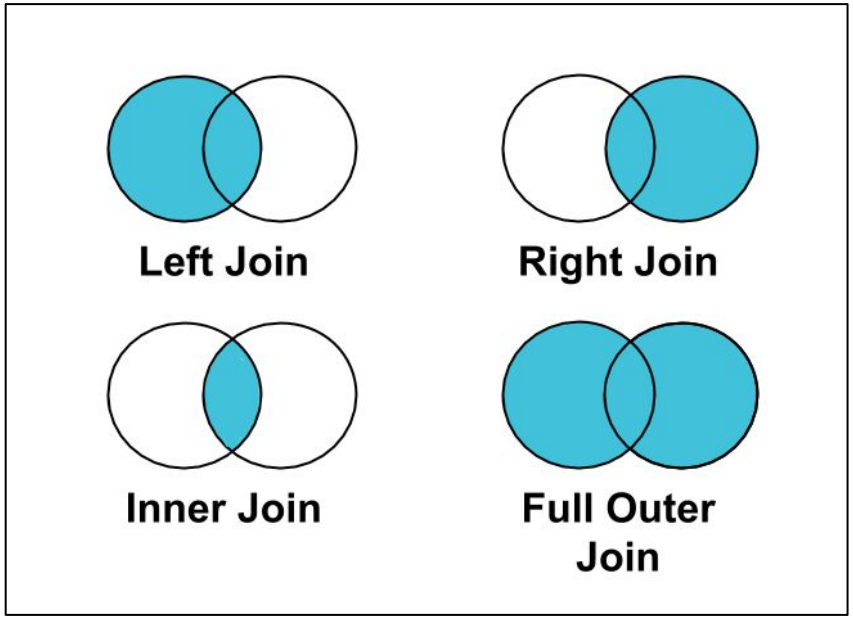
In English: Create a **plot** using the "colleges" **dataset**. Map to the **aesthetic** of **y-direction** the **variable** of sticker price, map to the **aesthetic** of **x-direction** the **variable** of average SAT, and map to the **aesthetic** of **color** the **variable** tier name. Use **points** as the **geom**, and make **alpha/transparency** equal to 0.5.

Additionally, use the **geom** of **smooth/line** to create lines of best fit. **Additionally**, insert **labels** to the **x-axis**, **y-axis**, **legends**, and **title**.

Content Review: Week 3

Data Joins

- Use to join **datasets** via a **key** (variable to link the 2 datasets)
- Left, Right, Inner, and Full



Left Join

- `left_join(houses, students, join_by("name" == "house"))`
- Combine 2 datasets via key, keeping all original observations from LEFT-HAND dataset while adding matching observations from RIGHT-HAND dataset

right

```
students
##    id  conc    house sleep
## 1 001   CPB Winthrop     7
## 2 002  HDRB  Currier     8
## 3 003  Stat Winthrop     8
## 4 004  Econ   Mather     9
## 5 005 Psych    Pfoho     6
## 6 006  Stat Winthrop     7
## 7 007    IB    Pfoho     8
```

left

```
houses
##        name built        area
## 1   Dunster  1930 River East
## 2 Winthrop  1931 River West
## 3   Currier  1970        Quad
## 4    Mather  1970 River East
```

0 matches → 1 row
3 matches → 3 rows
1 matches → 1 row
1 matches → 1 row

6 rows after left_join()

no match, but kept from original "left" dataset

```
left_join(houses, students,
          join_by("name" == "house"))
```

5 matches

```
##        name built        area   id conc sleep
## 1   Dunster  1930 River East <NA> <NA>    NA
## 2 Winthrop  1931 River West  001  CPB     7
## 3 Winthrop  1931 River West  003 Stat     8
## 4 Winthrop  1931 River West  006 Stat     7
## 5   Currier  1970        Quad  002 HDRB     8
## 6    Mather  1970 River East  004 Econ     9
```

Inner Join

- `inner_join(houses, students, join_by("name" == "house"))`
- Combine 2 datasets via key, keeping only matching observations between BOTH datasets (most constrained)

```
students
##    id  conc    house sleep
## 1 001   CPB Winthrop     7
## 2 002  HDRB  Currier     8
## 3 003  Stat Winthrop     8
## 4 004  Econ   Mather     9
## 5 005 Psych    Pfoho     6
## 6 006  Stat Winthrop     7
## 7 007    IB    Pfoho     8
houses
##        name built        area
## 1   Dunster  1930 River East
## 2 Winthrop  1931 River West
## 3   Currier  1970        Quad
## 4    Mather  1970 River East
```

```
inner_join(houses, students,
           join_by("name" == "house"))
```

```
##        name built        area  id conc sleep
## 1 Winthrop  1931 River West 001  CPB     7
## 2 Winthrop  1931 River West 003 Stat     8
## 3 Winthrop  1931 River West 006 Stat     7
## 4  Currier  1970        Quad 002 HDRB     8
## 5   Mather  1970 River East 004 Econ     9
```

5 matches

Full Join

- `full_join(houses, students, join_by("name" == "house"))`
- Combine 2 datasets via key, keeping all observations between BOTH datasets and putting N/A if an observation didn't have corresponding value for a variable (most expansive)

```
students
##    id  conc     house sleep
## 1 001   CPB  Winthrop     7
## 2 002  HDRB   Currier     8
## 3 003  Stat  Winthrop     8
## 4 004  Econ    Mather     9
## 5 005 Psych     Pfoho     6
## 6 006  Stat  Winthrop     7
## 7 007    IB     Pfoho     8
houses
##        name built        area
## 1   Dunster  1930 River East
## 2 Winthrop  1931 River West
## 3   Currier  1970        Quad
## 4    Mather  1970 River East
```

```
full_join(houses, students,
          join_by("name" == "house"))
```

```
##        name built        area   id  conc sleep
## 1   Dunster  1930 River East <NA>  <NA>    NA
## 2 Winthrop  1931 River West  001   CPB     7
## 3 Winthrop  1931 River West  003  Stat     8
## 4 Winthrop  1931 River West  006  Stat     7
## 5   Currier  1970        Quad  002  HDRB     8
## 6    Mather  1970 River East  004  Econ     9
## 7     Pfoho    NA        <NA>  005 Psych     6
## 8     Pfoho    NA        <NA>  007    IB     8
```

5 matches

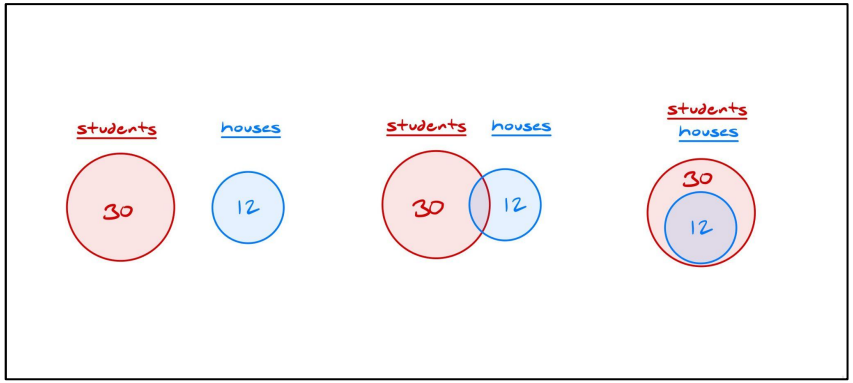
no matches, but still added

Practice: Answer These Questions

- LH dataset has 12 houses; RH dataset has 30 students (but not every student has to be in a house!)
- For `left_join()`, where houses is the LH dataset, what is the min # of rows possible?
- For `inner_join()` and `full_join()`, what is the min/max # of rows possible?

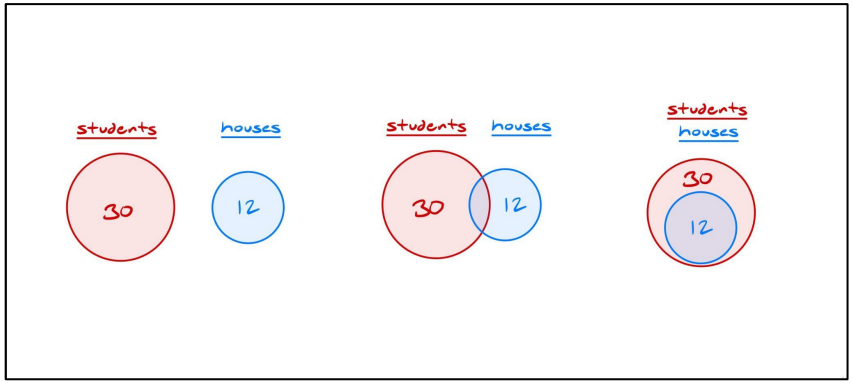
Solution: Left Join

- For `left_join()`, there must be at least 12 rows because each of the 12 rows in houses has to be represented, even if there are no matches for students (for example, if all 30 students are first-years)



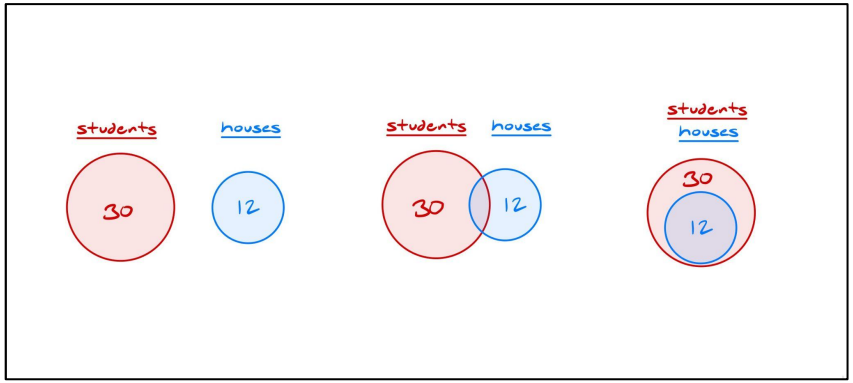
Solution: Inner Join

- For `inner_join()`, there can be 0 rows if there are no matches between students and houses, and there can be up to 30 rows if all students are matched to a house
- It may seem like there'd be a maximum of 12 rows, but if all 30 students had Winthrop, there'd be 30 rows (see Slide 17)



Solution: Full Join

- For `full_join()`, there can be 30 rows in the event that all students are matched to a house, and there can be up to 42 rows if there are no matches between students and houses



Removing Missing Values

https://drive.google.com/file/d/1poOyUn9yBChKcrBxGZDiTleQfKIqDWYm/view?usp=drive_link

Important Code for Data Wrangling

https://drive.google.com/file/d/1Mr7SHkdQ06N_G8k4BfHrr5HSg1CEJrrL/view?usp=drive_link

Pipe

- %>%: Takes dataset and “pipes” it as the first argument in the next line
  - The first argument of most wrangling verbs is a dataset
  - This is read as “and then” when reading code aloud
- ‘Command’ + ‘Shift’ + ‘M’

Pipe: These Are Equivalent Statements

```
mythbusters %>%
    summarize(count =
n())
```

```
summarize(mythbusters,
count = n())
```

Practice 1: What Does This Code Do?

```
women_in_stem <- people %>%
   filter(gender == "Female", jobtitle ==
"Software Engineer" | jobtitle ==
"Mathematician")
```

Solution

- We **define** a new **dataset** “women_in_stem” **assigned** from the **dataset** “people”...
- **And then** we **filter** for the gender of “female” **and** the job title of “software engineer” **or** “mathematician”

Practice 2: What Does This Code Do?

```
students_new <- students %>%
    mutate(seniority_new = case_when(
        seniority <= 2 ~ "junior",
        seniority == 3 ~ "mid",
        seniority >= 4 ~ "senior"))
```

Solution

- We **assign** the **dataset** “students” to itself (so that it gets updated)...
- **And then** we **mutate** a new variable “seniority_new”, which is...
  - Equal to “junior” **when** the variable “seniority” is less than/equal to 2
  - Equal to “mid” **when** the variable “seniority” is equal to 3
  - Equal to “senior” **when** the variable “seniority” is greater than/equal to 3

Practice 3: What Does This Code Do?

```
people %>%
    drop_na(pay) %>%
    filter(gender == "Female", jobtitle ==
"Financial Analyst") %>%
    slice_max(pay, n = 10) %>%
    select(pay, education, name)
```

Solution

- We take the **dataset** “people”…
- **And then** we **drop** all observations that have **NA** for the variable “pay”…
- **And then** we **filter** for the gender of “female” **and** the job title of “financial analyst”
- **And then** we **slice** for the observations with the top 10 **maximum** values for the variable “pay”
- **And then** we **select** (to display) the variables “pay,” “education,” and “name”

Questions?

P-Set 2

Have a great rest of your week!