Code ▾

Statistical Foundations

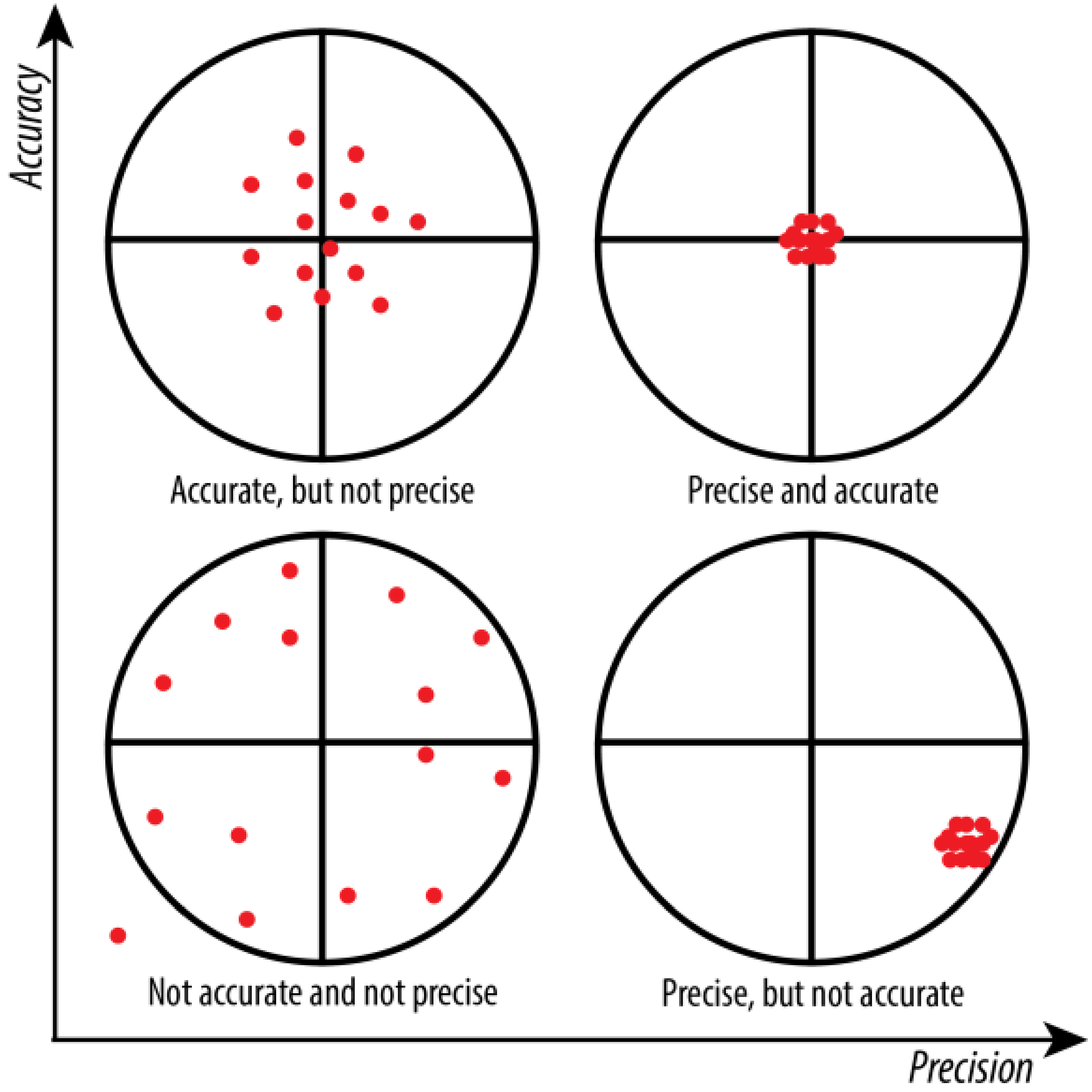
Instructor - Soumya Mukherjee

Content Credit- Dr. Matthew Beckman and Olivia Beck

July 25, 2023

Key Ideas

- Statistics is the area of science concerned with characterizing case-to-case variation and the collective properties of cases
- It’s one thing to simply calculate quantities from data, but a central topic in statistics is the **precision** with which we can estimate those quantities
- **Accommodating and quantifying uncertainty due to randomness** is perhaps the backbone of inferential statistics
- You **ALWAYS** need to report uncertainty and/or variation


Accuracy
Accurate, but not precise
Precise and accurate
Not accurate and not precise
Precise, but not accurate
Precision

Key Ideas (continued)

- it’s often useful to look at distributions of data graphically, and you were presented several useful ways to do so
  - density plots (perhaps overlaid or faceted to highlight group comparisons)

    - box plots (again, side-by-side or faceted to highlight group comparisons)
    - violin plots (alternative to box plots with greater detail in the density)
    - note, all of these plots show variation
- model functions help describe a relationship between an “input” (X, explanatory, independent) variable and an “output” (Y, response, dependent) variable
    - smoothers show relationships between variables that bend with the data collectively
    - linear functions are also useful in some contexts, but sometimes miss important features of the relationship between variables
- confidence intervals (when estimating a single quantity) and confidence bands (when estimating model functions) are **essential** to communicate uncertainty
    - small sample sizes result in a great deal of uncertainty
    - we can be more confident in estimates produced by large sample sizes (asymptotic/ large sample theory)
    - error bars, notched box plots, and confidence bands all help communicate uncertainty in a graph.

Recall: Key goals of a careful Exploratory Data Analysis

1. **Examine the data source:** variable types, coding, missingness, summary statistics/plots, who/what/when/where/why/how data were collected
2. **Discover features that influence may modeling decisions:** investigate potential outliers, consideration for recoding variables (e.g., numeric data that’s functionally dichotomous), evaluate correlation structure (e.g., autocorrelation, hierarchy, spatial/temporal proximity)
3. **Address research questions:** build intuition and note preliminary observations/conclusions related to each research question. Also, note observations that prompt you to refine your research questions or add new questions to investigate

…this is often an iterative process, but the order shown might help you organize your approach.

Examples of Graphs

Means

Hide

```
#Set up data
#?AirPassengers
class(AirPassengers)
```

```
[1] "ts"
```

Hide

```
dat_passengers <- data.frame(1949:1960,
                             matrix(AirPassengers, ncol = 12))
colnames(dat_passengers) <- c("Year", month.abb)

head(dat_passengers)
```

	Year <int>	Jan <dbl>	Feb <dbl>	Mar <dbl>	Apr <dbl>	May <dbl>	Jun <dbl>	Jul <dbl>	Aug <dbl> ▸
1	1949	112	115	145	171	196	204	242	284
2	1950	118	126	150	180	196	188	233	277
3	1951	132	141	178	193	236	235	267	317
4	1952	129	135	163	181	235	227	269	313
5	1953	121	125	172	183	229	234	270	318
6	1954	135	149	178	218	243	264	315	374

6 rows | 1-10 of 13 columns

Hide

```
# Wrangle
GlyphReadyData <- dat_passengers %>%
  pivot_longer(!Year, names_to = "Month", values_to = "Val") %>%
  group_by(Year) %>%
  summarise(Mean = mean(Val, na.rm = T),
            SD = stats::sd(Val, na.rm = T))
head(GlyphReadyData)
```

Year <int>	Mean <dbl>	SD <dbl>
1949	241.7500	101.0330

Year	Mean	SD
<int>	<dbl>	<dbl>
1950	235.0000	89.6194
1951	270.1667	100.5592
1952	267.0833	107.3748
1953	271.8333	114.7399
1954	311.6667	134.2199

6 rows

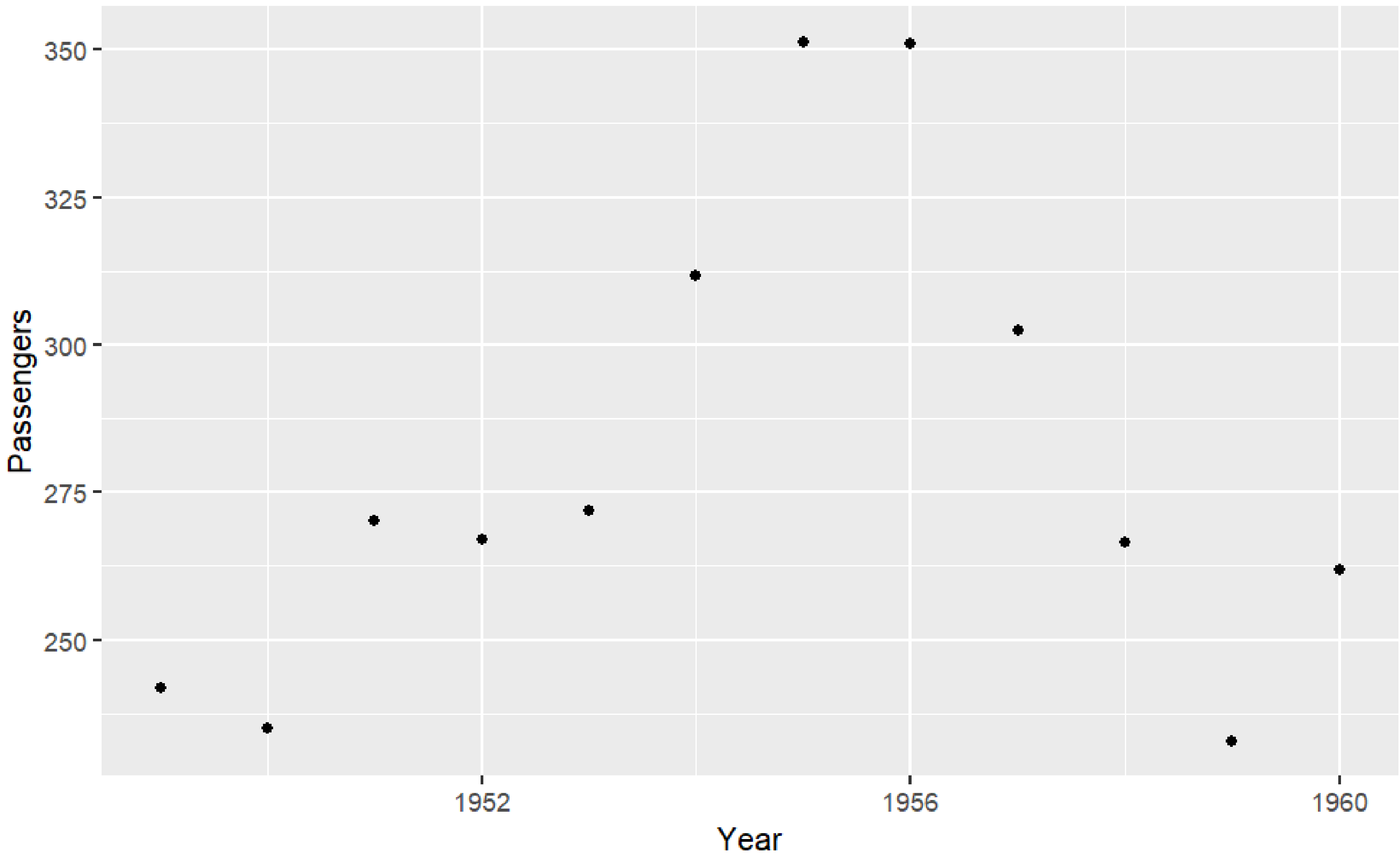
Hide

```
NA
NA
```

Here are multiple ways of displaying the same data

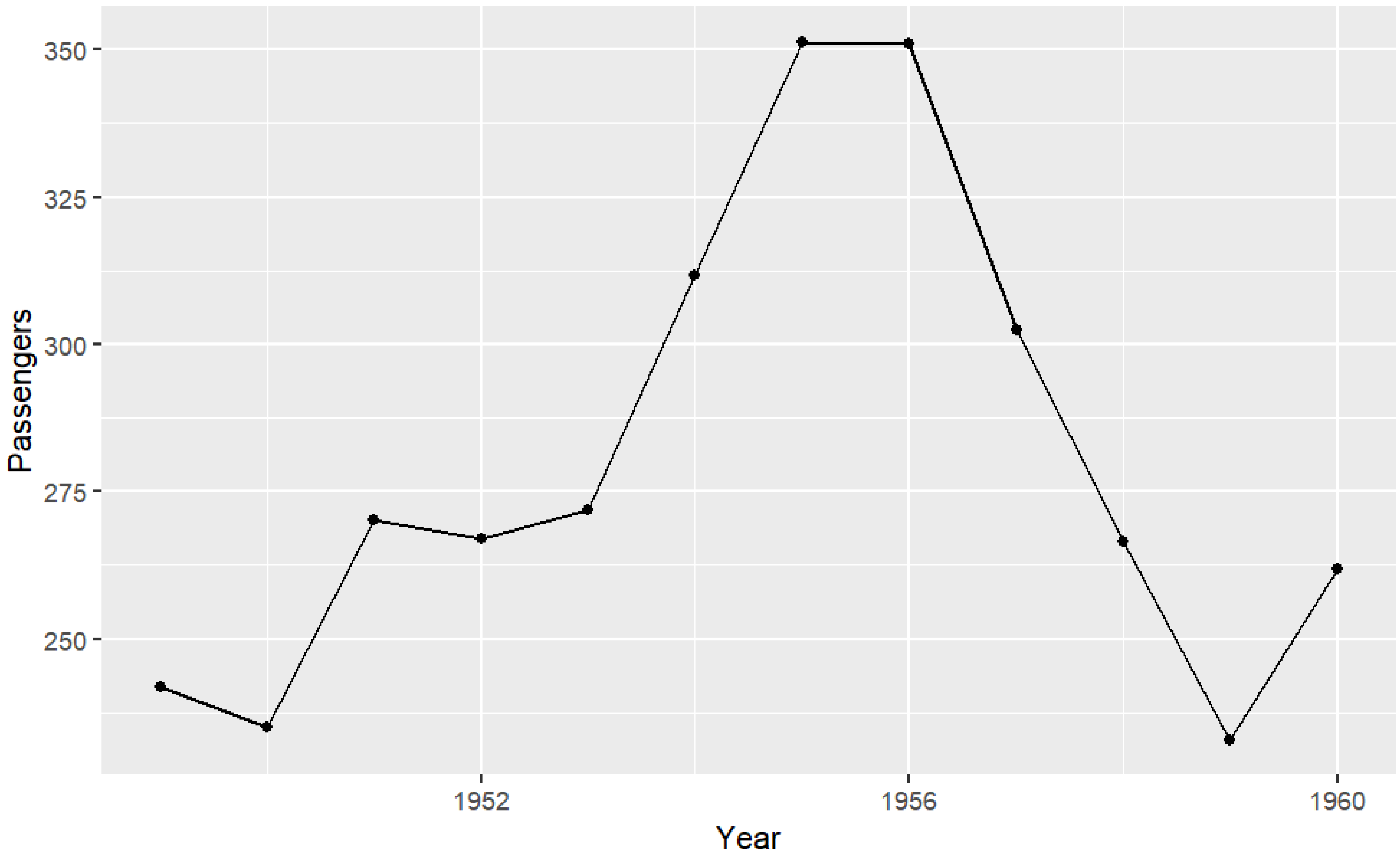
Hide

```
ggplot(GlyphReadyData, aes(x = Year, y = Mean)) +
  geom_point()+
  xlab("Year") +
  ylab("Passengers")
```



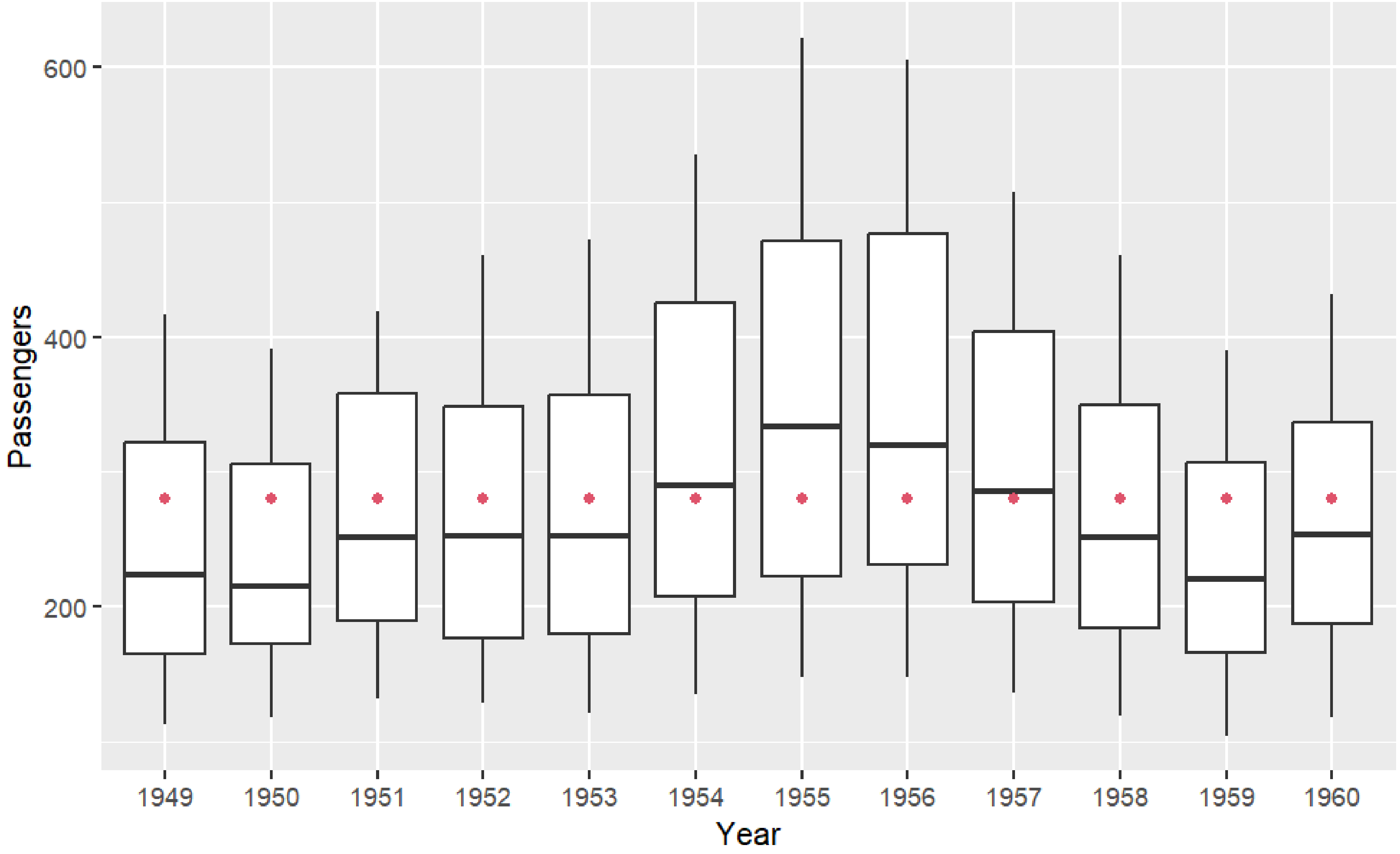
Hide

```
ggplot(GlyphReadyData, aes(x = Year, y = Mean)) +
  geom_point() +
  geom_line()+
  xlab("Year") +
  ylab("Passengers")
```



Hide

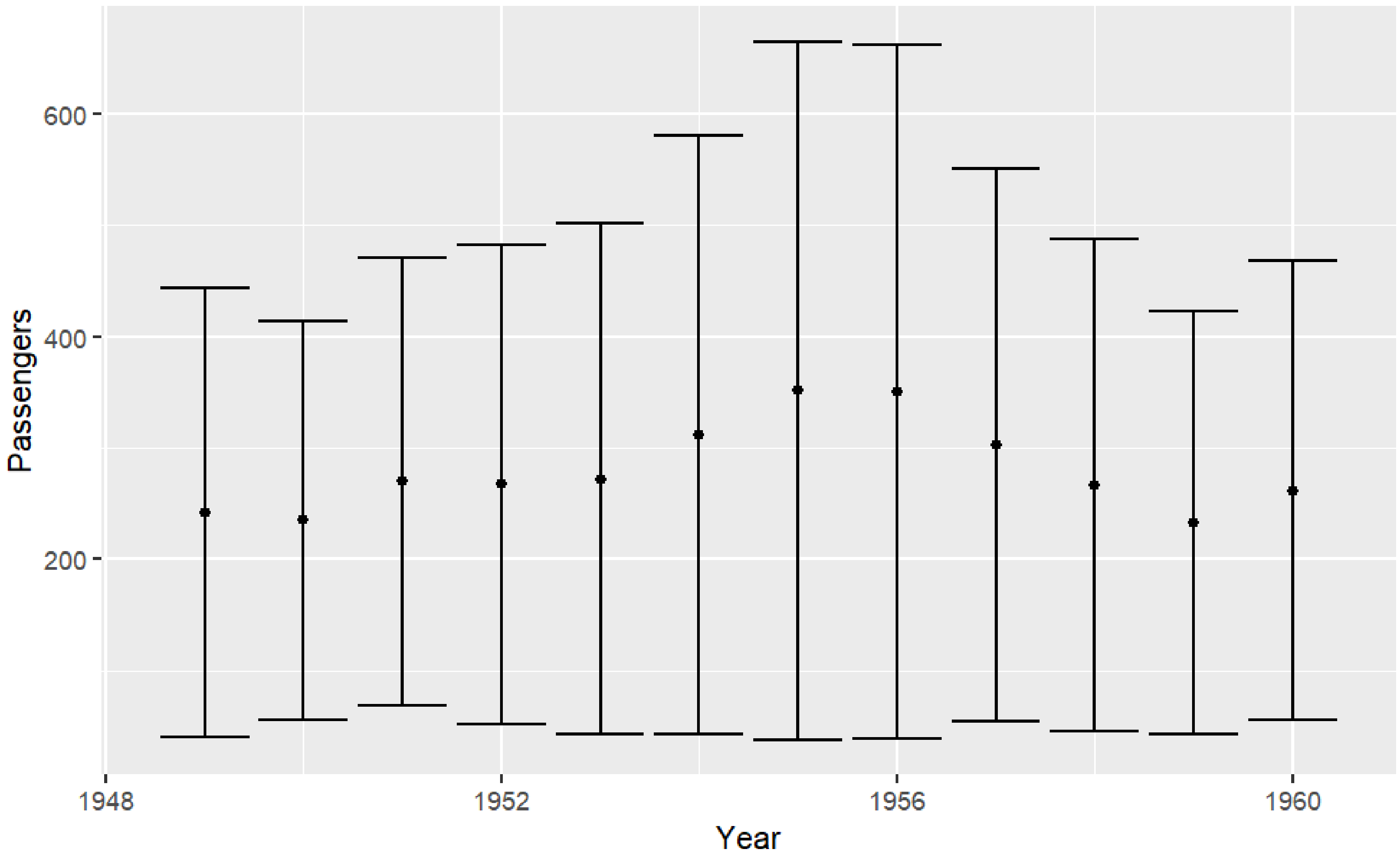
```
dat_passengers %>%
  pivot_longer(!Year, names_to = "Month", values_to = "Val") %>%
  ggplot( aes(x = as.factor(Year), y = Val)) +
  geom_boxplot() +
  geom_point(aes(y = mean(Val)), col = 2) +
  xlab("Year") +
  ylab("Passengers")
```


600
400
200
Passengers
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
Year

Hide

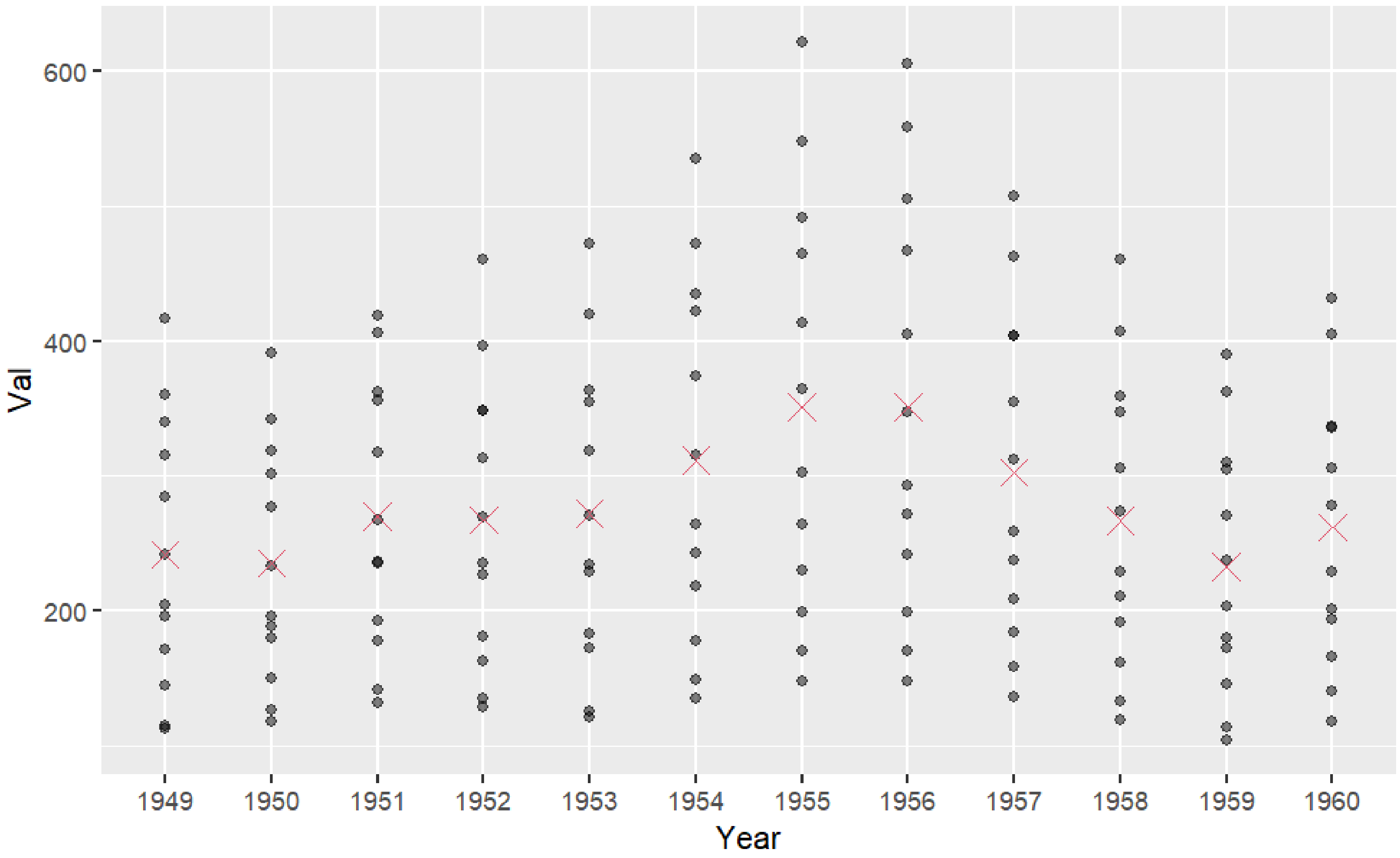
```
#color 2 is red, 3 is green, 4 is blue, 5 is cyan, 6 is purple, 7 is yellow, and so on.....
# R also understands HEX codes as colors

ggplot(GlyphReadyData, aes(x = Year, y = Mean)) +
  geom_point()+
  geom_errorbar(aes(ymin = Mean - 2 * SD, ymax = Mean + 2 * SD))+
  xlab("Year") +
  ylab("Passengers")
```



Hide

```
dat_passengers %>%
  pivot_longer(!Year, names_to = "Month", values_to = "Val") %>%
  ggplot()+
  geom_point(aes(x = as.factor(Year), y = Val), alpha = 0.5) +
  geom_point(data = GlyphReadyData, aes(x = as.factor(Year), y = Mean), col = 2, shape = 4, size = 4) +
  xlab("Year")
```



Hide

NA

Questions

- Which graph would you choose to display? why?
- How does your perception of the data change between each graph?

- What other graphs might you want to make?
- Would this information better be displayed as a table? Why or why not?

Guided Example: SAT Data exploration

Statement of Research Question

Are higher teacher salaries associated with better state-wide SAT scores?

Examine the data source

for example:

- who/what/when/where/why/how data were collected
- review data intake
- variable types,
- coding,
- missingness,
- basic summary statistics and plots to learn about variables

Hide

```
data("SAT_2010")

# review data intake & variable coding
glimpse(SAT_2010)
```

```
Rows: 50
Columns: 9
$ state               <fct> Alabama, Alaska, Arizona, Arka…
$ expenditure         <int> 10, 17, 9, 10, 10, 10, 16, 13,…
$ pupil_teacher_ratio <dbl> 15.3, 16.2, 21.4, 14.1, 24.1, …
$ salary              <int> 49948, 62654, 49298, 49033, 71…
$ read                <int> 556, 518, 519, 566, 501, 568, …
$ math                <int> 550, 515, 525, 566, 516, 572, …
$ write               <int> 544, 491, 500, 552, 500, 555, …
$ total               <int> 1650, 1524, 1544, 1684, 1517, …
$ sat_pct             <int> 8, 52, 28, 5, 53, 19, 87, 74, …
```

Hide

```
head(SAT_2010)
```

	state <fctr>	expenditure <int>	pupil_teacher_ratio <dbl>	salary <int>	read <int>	math <int>	write <int>	total <int>	sat_pct <int>
1	Alabama	10	15.3	49948	556	550	544	1650	8
2	Alaska	17	16.2	62654	518	515	491	1524	52
3	Arizona	9	21.4	49298	519	525	500	1544	28
4	Arkansas	10	14.1	49033	566	566	552	1684	5
5	California	10	24.1	71611	501	516	500	1517	53
6	Colorado	10	17.4	51660	568	572	555	1695	19

6 rows

Hide

```
tail(SAT_2010)
```

	state <fctr>	expenditure <int>	pupil_teacher_ratio <dbl>	salary <int>	read <int>	math <int>	write <int>	total <int>	sat_pct <int>
45	Vermont	17	11.6	51537	519	521	506	1546	67
46	Virginia	11	17.6	52514	512	512	497	1521	71
47	Washington	10	19.4	55651	524	532	508	1564	57
48	West Virginia	12	13.9	48255	515	507	500	1522	17
49	Wisconsin	12	15.1	53826	595	604	579	1778	5
50	Wyoming	17	12.5	58652	570	567	546	1683	5

6 rows

Hide

```
NA
```

Hide

```
# missingness & summary statistics
favstats( ~ salary, data = SAT_2010)
```

min	Q1	median	Q3	max	mean	sd	n	missing
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>
40778	48999.25	52062.5	58728.5	75212	54721.28	7755.489	50	0

1 row

Hide

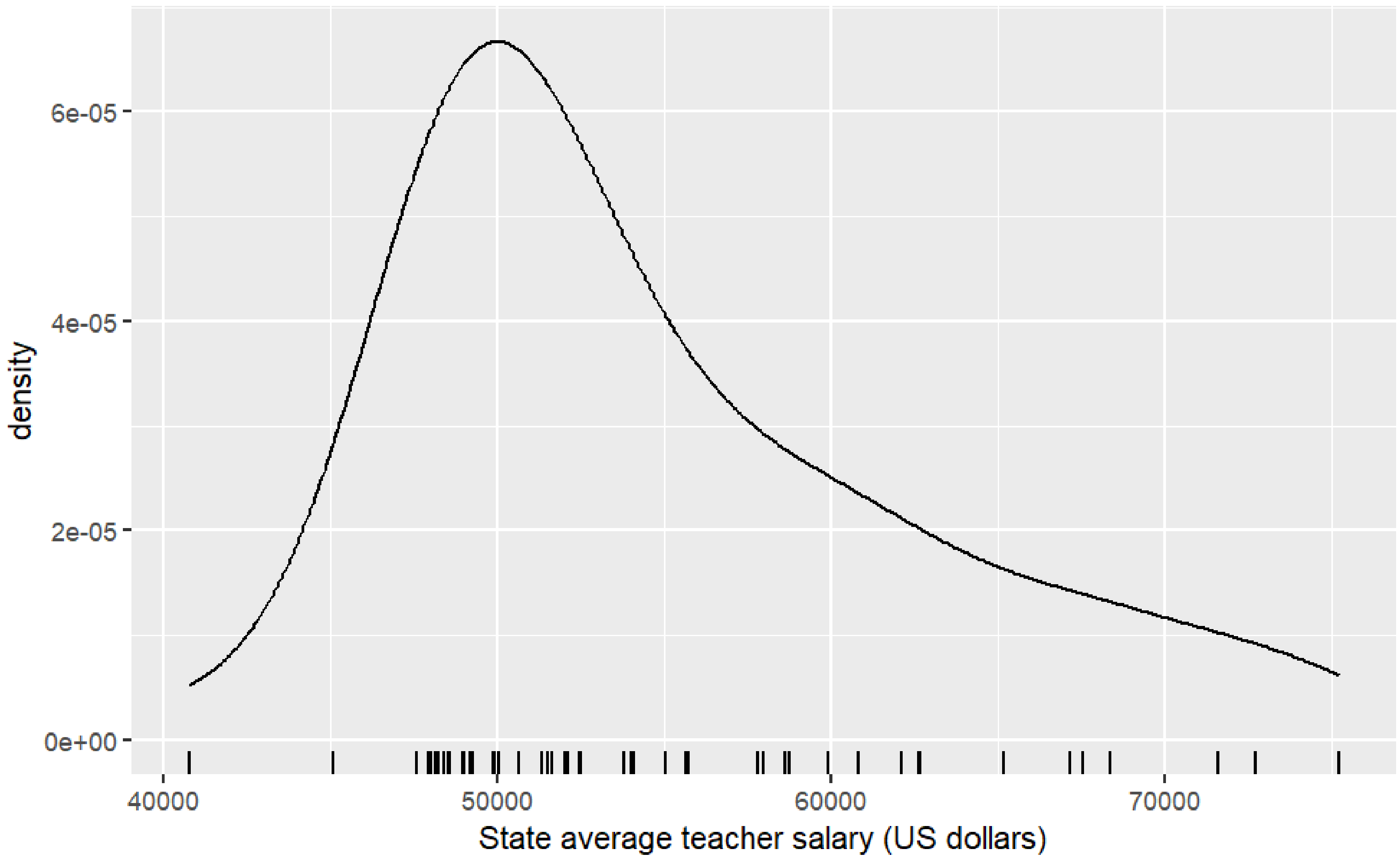
```
favstats( ~ total, data = SAT_2010)
```

min	Q1	median	Q3	max	mean	sd	n	missing
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>
1389	1489.25	1559	1692.25	1798	1596.46	116.7002	50	0

1 row

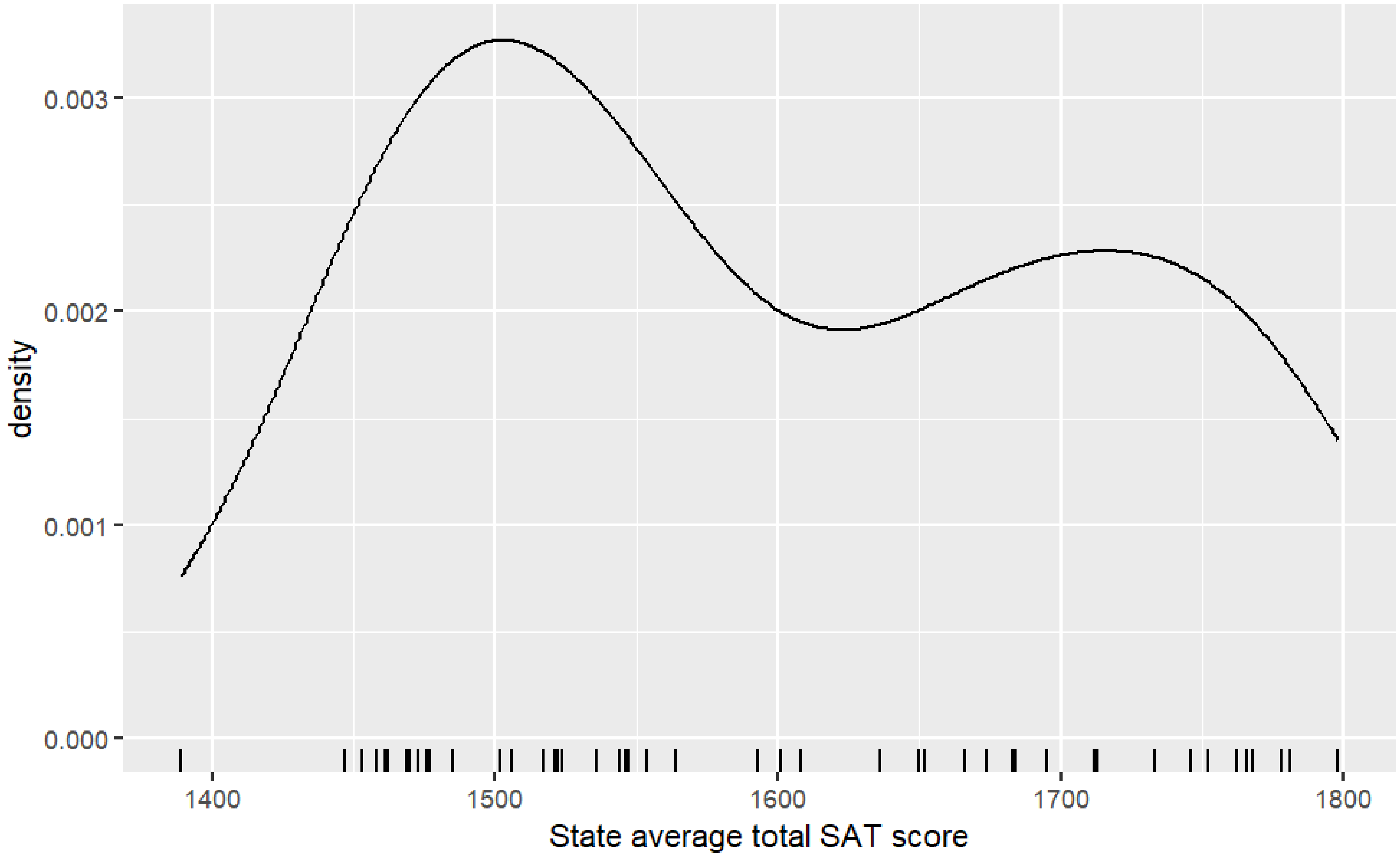
Hide

```
SAT_2010 %>%
    ggplot(aes(x = salary)) +
    geom_density() +
    geom_rug() +
    xlab("State average teacher salary (US dollars)")
```


density
6e-05
4e-05
2e-05
0e+00
40000
50000
60000
70000
State average teacher salary (US dollars)

Hide

```
SAT_2010 %>%
    ggplot(aes(x = total)) +
    geom_density() +
    geom_rug() +
    xlab("State average total SAT score")
```


0.003
0.002
0.001
0.000
density
1400
1500
1600
1700
1800
State average total SAT score

Hide

```
NA
NA
```

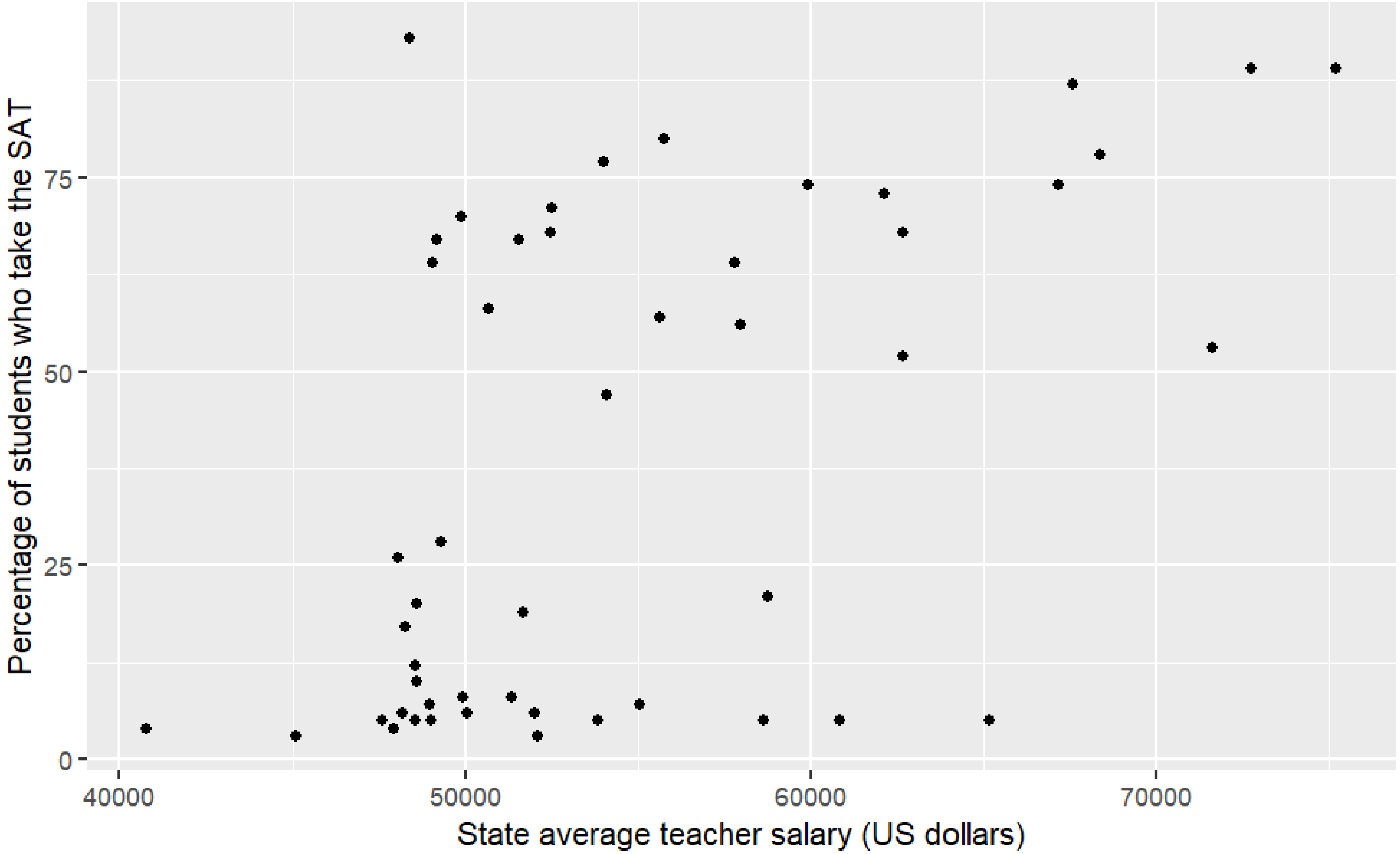
Discover features in the data that may impact modeling decisions

Some of this is based on scrutiny of the data collection practices (and study design), but much of it can be substantiated in EDA

- investigate outliers
- functionally dichotomous variables–e.g., survey asks people to rate job approval of president on scale of 1-7, but most people choose either 1 or 7 and the options in between are rarely used
- highly correlated predictor variables
- hierarchy or nesting–e.g., data from students within classrooms within schools
- repeated observations of the same “case”–e.g., medical study follows up with the same group of patients every 6 months

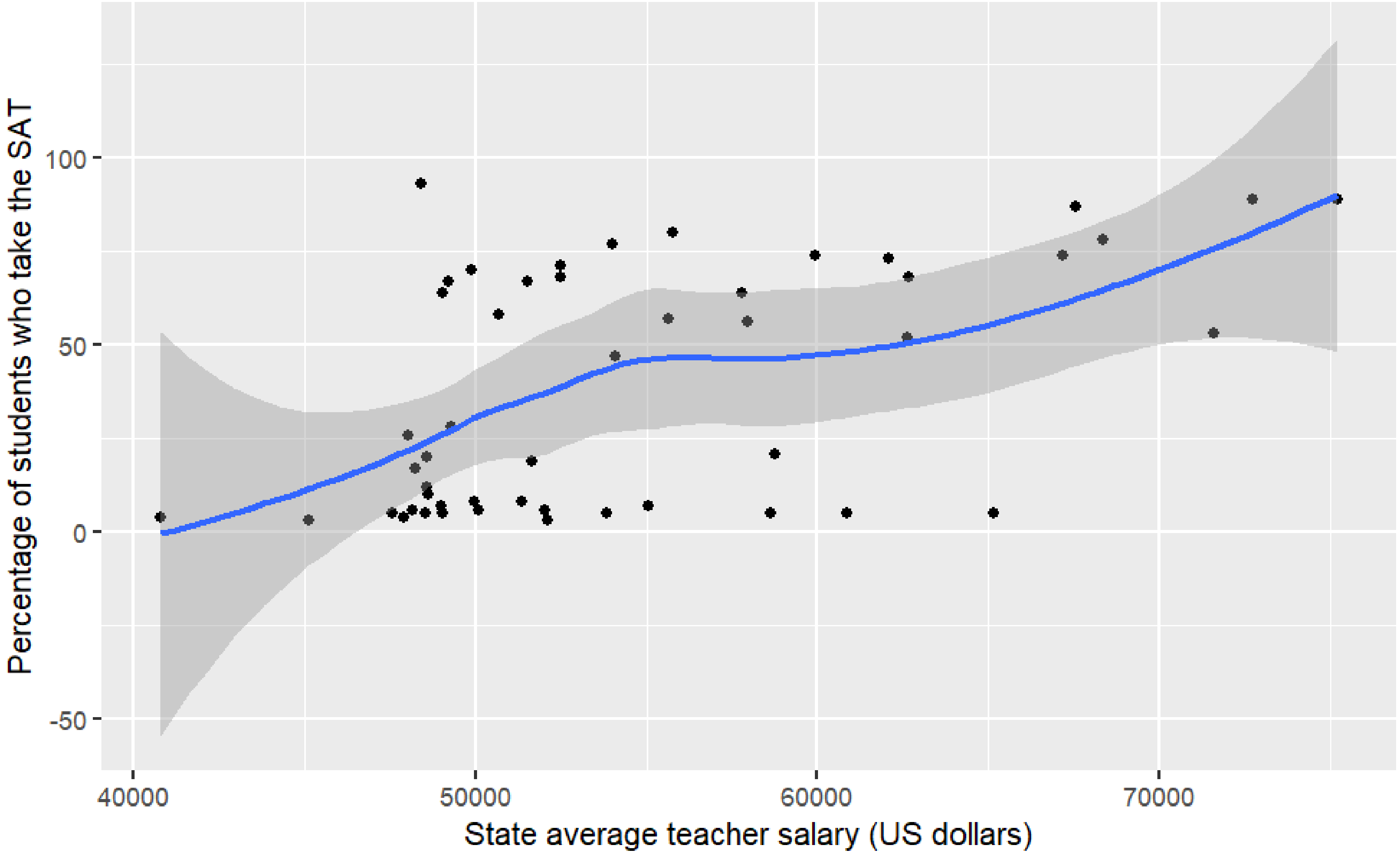
Hide

```
SAT_2010 %>%
    ggplot(aes(x = salary, y = sat_pct)) +
    geom_point() +
    # geom_smooth() +     # if linear is reasonable, consider method = "lm"
    xlab("State average teacher salary (US dollars)") +
    ylab("Percentage of students who take the SAT")
```


Percentage of students who take the SAT
0
25
50
75
40000
50000
60000
70000
State average teacher salary (US dollars)

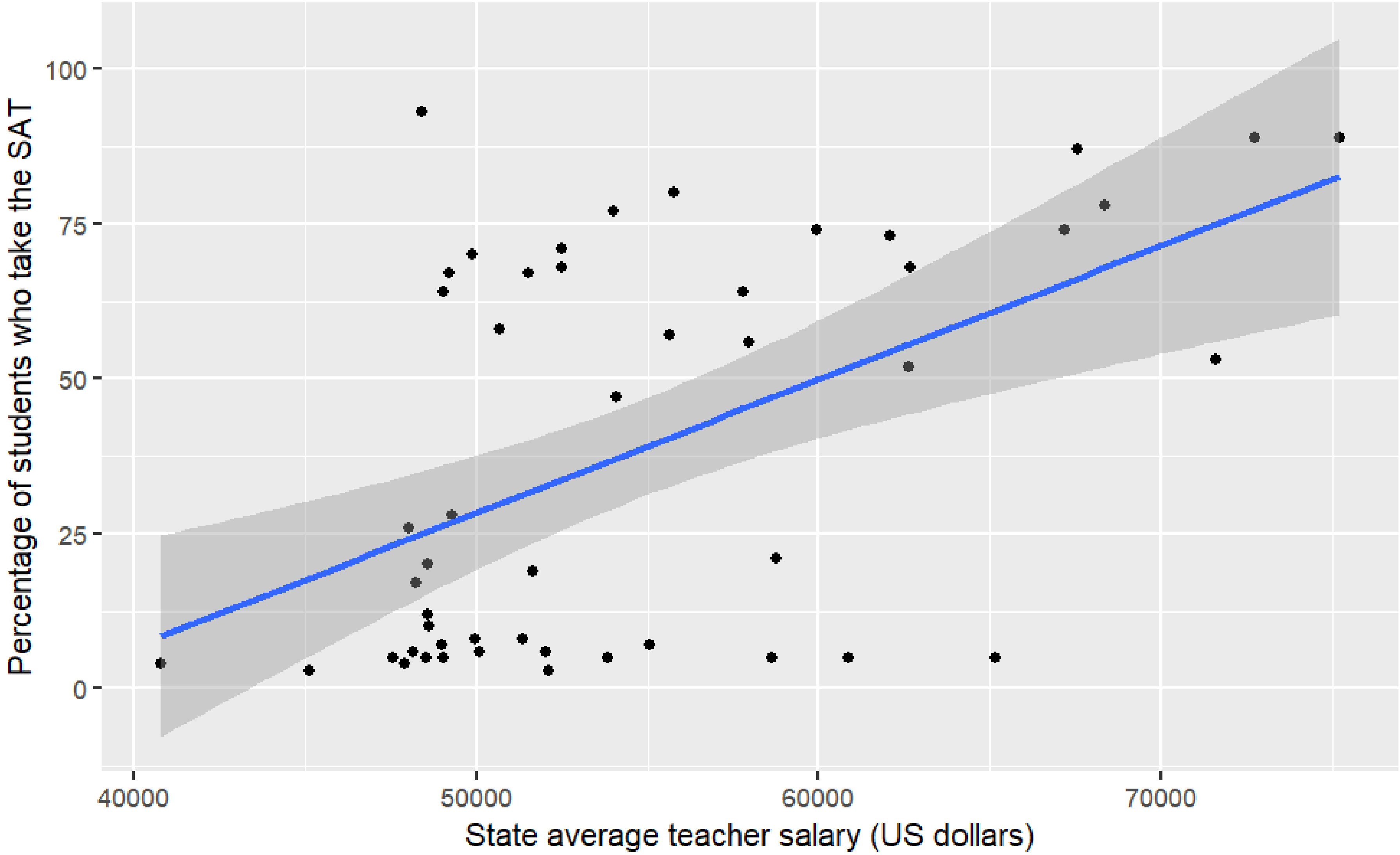
Hide

```
SAT_2010 %>%
    ggplot(aes(x = salary, y = sat_pct)) +
    geom_point() +
    geom_smooth() +
    xlab("State average teacher salary (US dollars)") +
    ylab("Percentage of students who take the SAT")
```


Percentage of students who take the SAT
100
50
0
-50
40000
50000
60000
70000
State average teacher salary (US dollars)

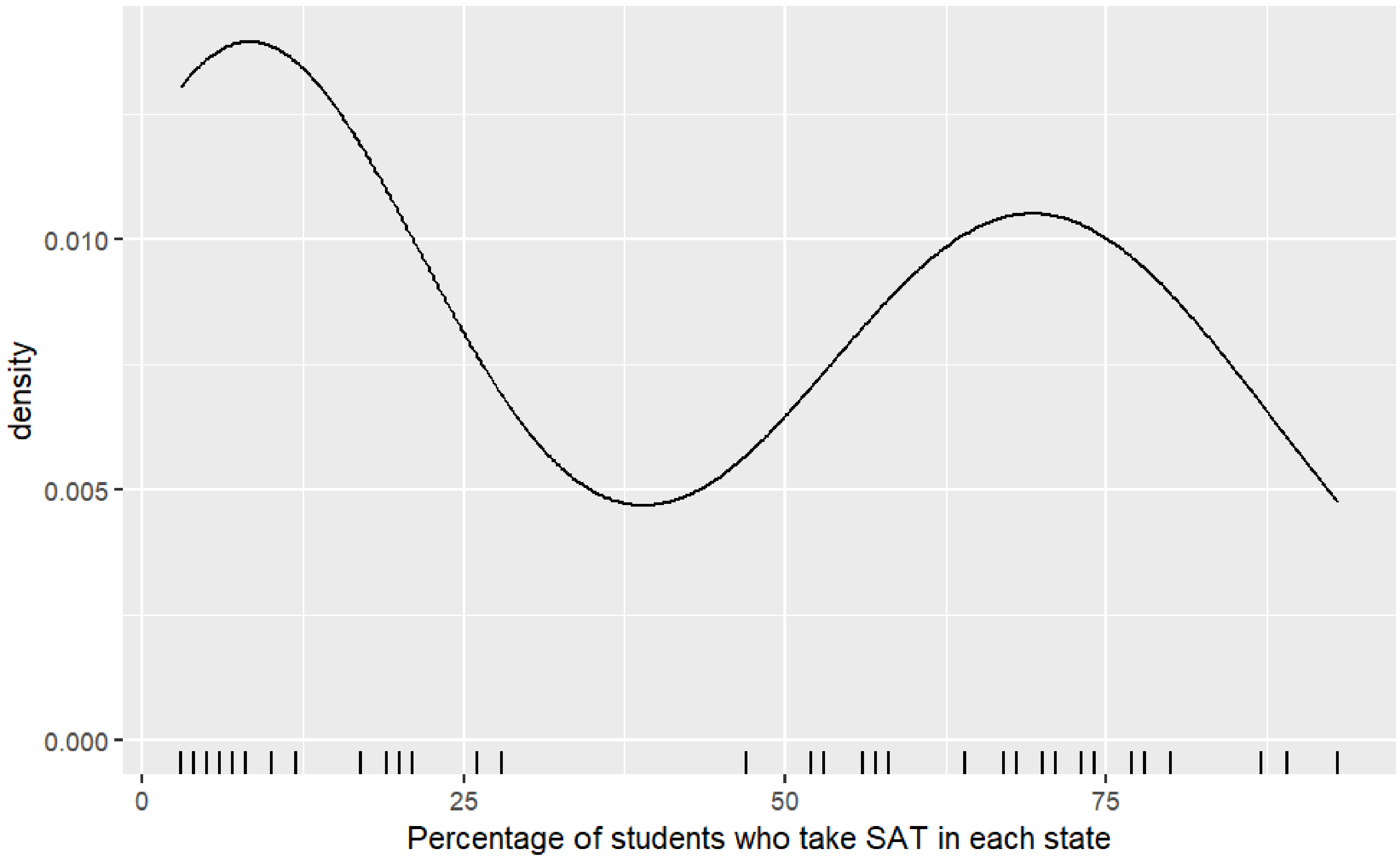
Hide

```
SAT_2010 %>%
    ggplot(aes(x = salary, y = sat_pct)) +
    geom_point() +
    geom_smooth(method = "lm") +
    xlab("State average teacher salary (US dollars)") +
    ylab("Percentage of students who take the SAT")
```


Percentage of students who take the SAT
100
75
50
25
0
40000
50000
60000
70000
State average teacher salary (US dollars)

Hide

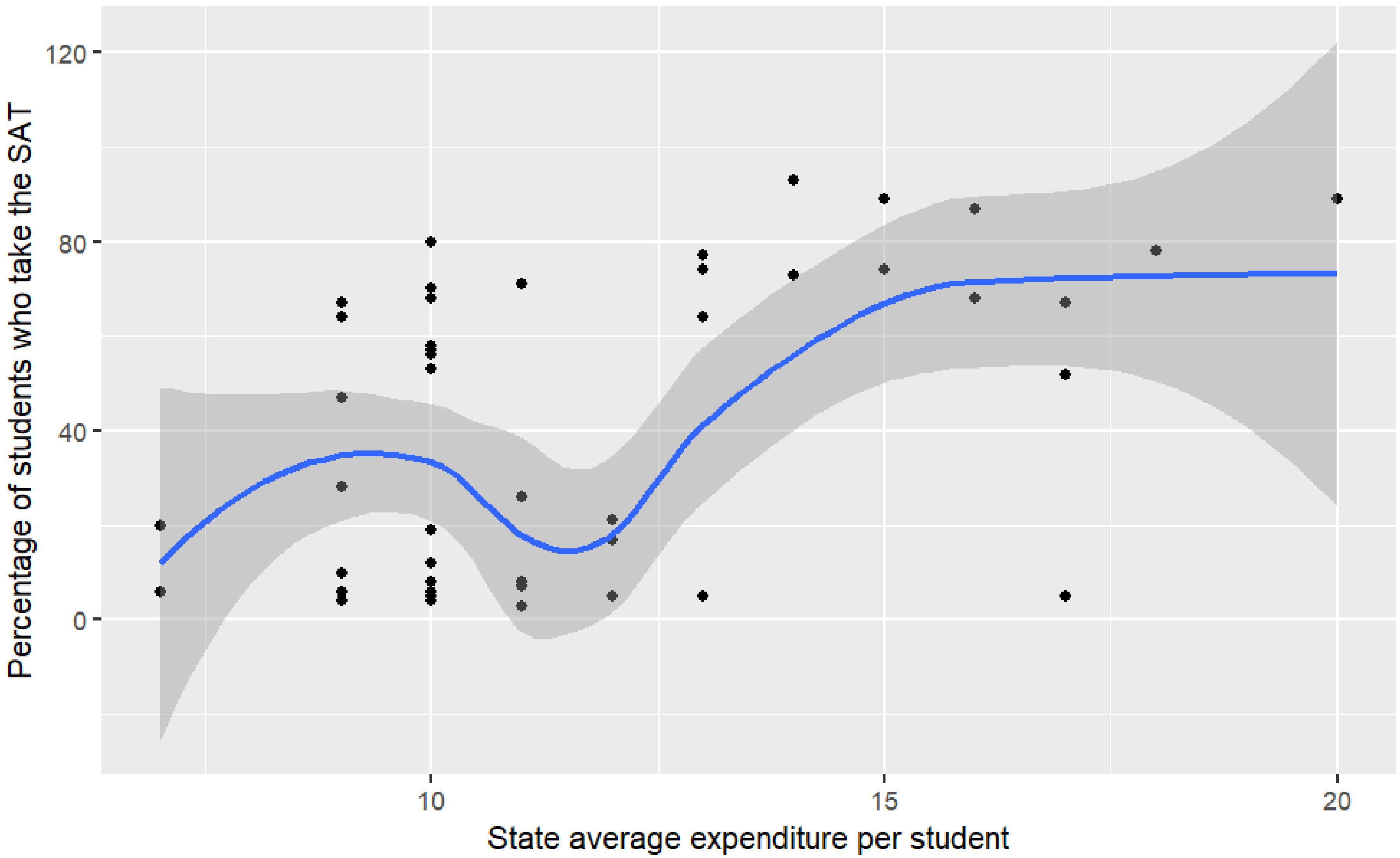
```
# density of sat_rate... apparent gap (why??)
SAT_2010 %>%
    ggplot(aes(x = sat_pct)) +
    geom_density() +
    geom_rug() +
    xlab("Percentage of students who take SAT in each state")
```


density
0.010
0.005
0.000
0
25
50
75
Percentage of students who take SAT in each state

Hide

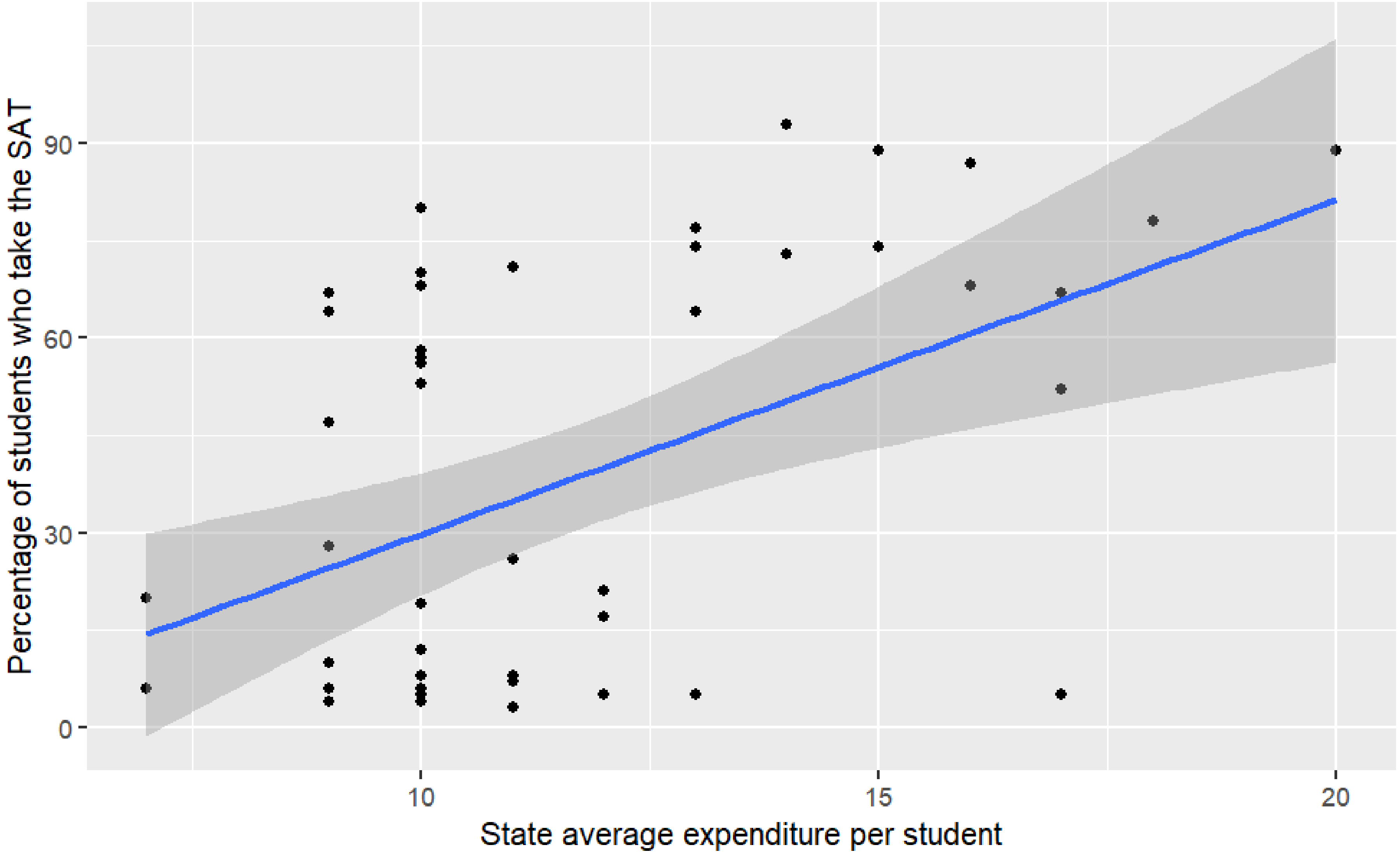
```
SAT_2010 <-
    SAT_2010 %>%
    mutate(sat_rate = cut(sat_pct, breaks = c(0, 30, 70, 100),
                          labels = c("low", "med", "high")))
    # mutate(sat_rate = cut(sat_pct, breaks = c(0, 40, 100),
    #                       labels = c("lower", "higher")))

SAT_2010 %>%
    ggplot(aes(x = expenditure, y = sat_pct)) +
    geom_point() +
    geom_smooth() +
    xlab("State average expenditure per student") +
    ylab("Percentage of students who take the SAT")
```


120
80
40
0
10
15
20
Percentage of students who take the SAT
State average expenditure per student

Hide

```
SAT_2010 %>%
    ggplot(aes(x = expenditure, y = sat_pct)) +
    geom_point() +
    geom_smooth(method = "lm") +
    xlab("State average expenditure per student") +
    ylab("Percentage of students who take the SAT")
```



Hide

```
SAT_2010 %>%
    filter(sat_pct < 25, expenditure > 15)
```

state	expenditure	pupil_teacher_ratio	salary	read	math	write	total	sat_pct	sat_rate
<fctr>	<int>	<dbl>	<int>	<int>	<int>	<int>	<int>	<int>	<fctr>
Wyoming	17	12.5	58652	570	567	546	1683	5	low

1 row

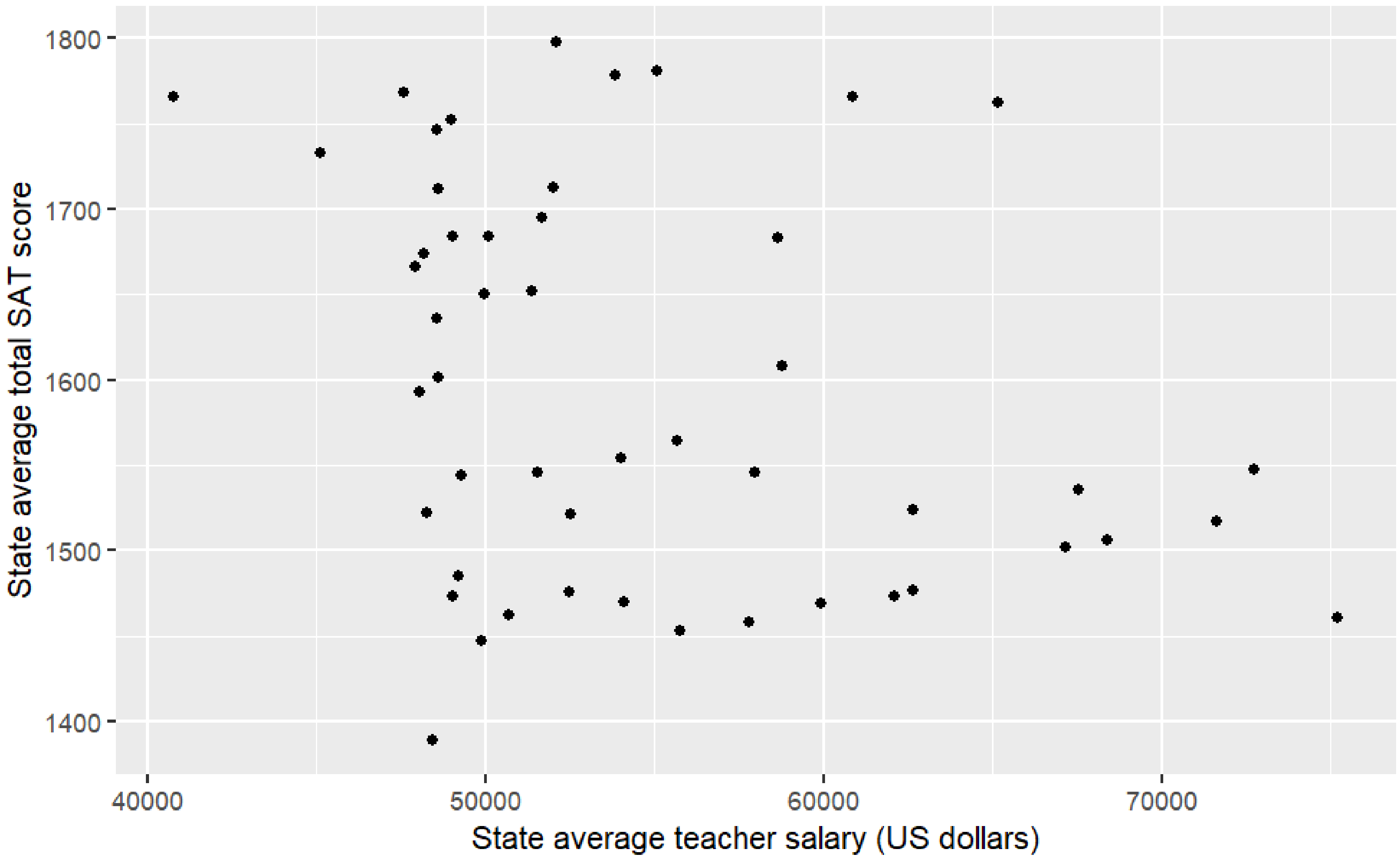
Address research question

- one or a few key data visualizations that are most informative to a reader/observer
- include data visualization (but not exclusively)
- often requires exploring many data visualizations to find the one or few that most effectively communicate intuition for your research question
- we may even do some exploratory modeling here

Hide

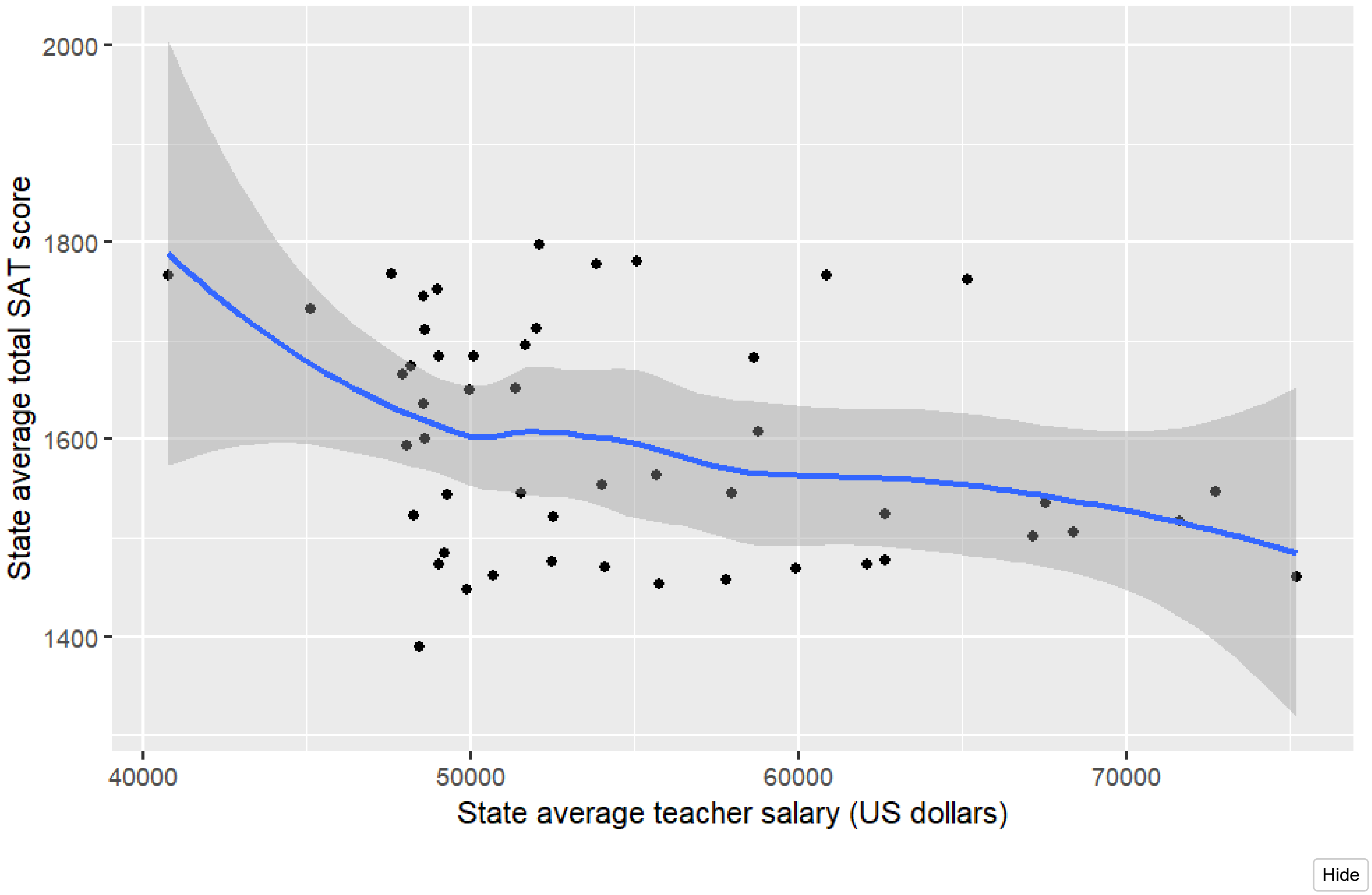
```
## Relationship b/w salary and Average total SAT scores

SAT_2010 %>%
    ggplot(aes(x = salary, y = total)) +
    geom_point() +
    # geom_smooth() +
    # geom_smooth(method = "lm") +
    xlab("State average teacher salary (US dollars)") +
    ylab("State average total SAT score")
```

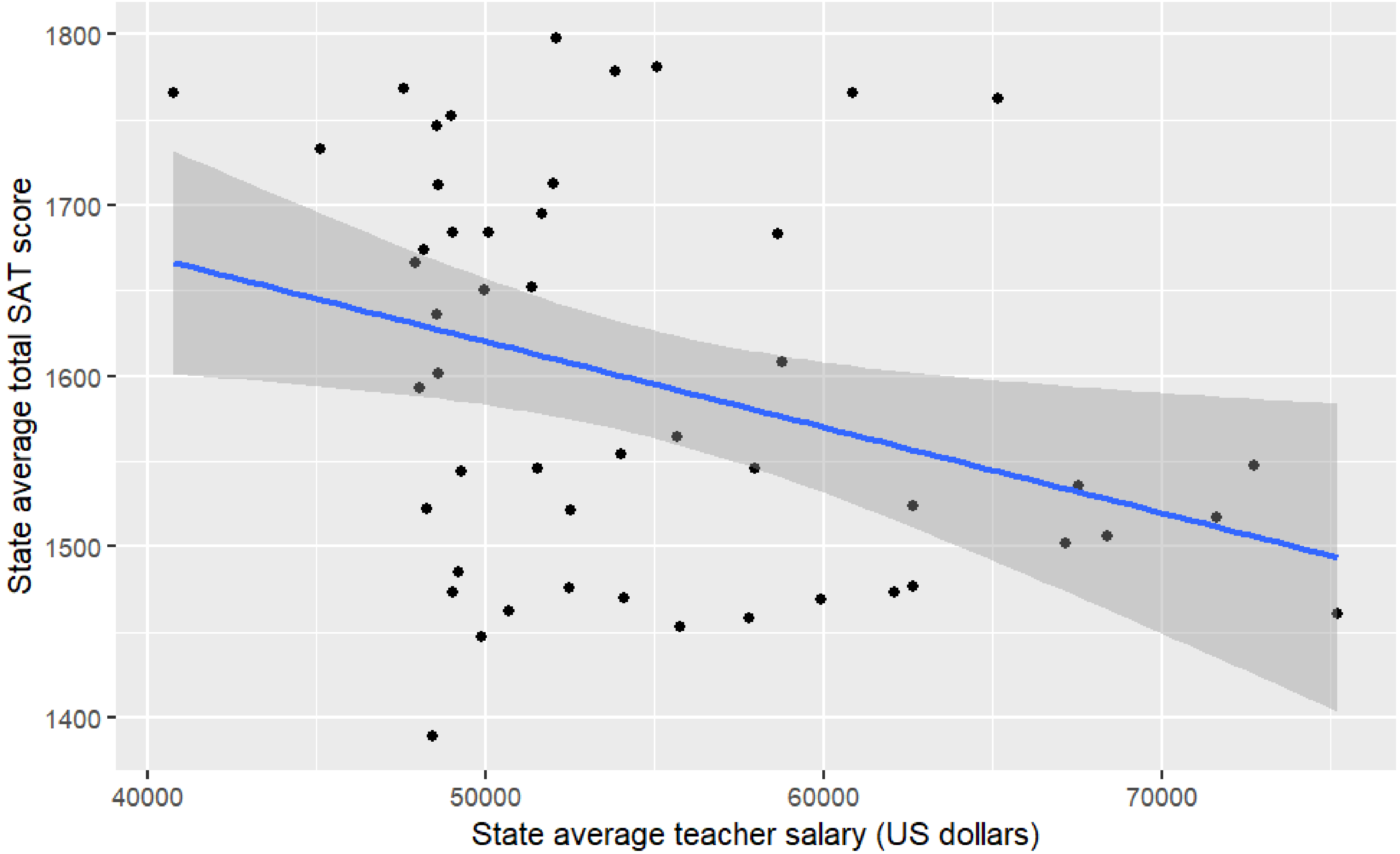

1800
1700
1600
1500
1400
State average total SAT score
40000
50000
60000
70000
State average teacher salary (US dollars)

Hide

```r
SAT_2010 %>%
    ggplot(aes(x = salary, y = total)) +
    geom_point() +
    geom_smooth() +
    # geom_smooth(method = "lm") +
    xlab("State average teacher salary (US dollars)") +
    ylab("State average total SAT score")
```


State average total SAT score
2000
1800
1600
1400
40000
50000
60000
70000
State average teacher salary (US dollars)
Hide

```
SAT_2010 %>%
    ggplot(aes(x = salary, y = total)) +
    geom_point() +
    geom_smooth(method = "lm") +
    xlab("State average teacher salary (US dollars)") +
    ylab("State average total SAT score")
```


1800
1700
1600
1500
1400
40000
50000
60000
70000
State average total SAT score
State average teacher salary (US dollars)

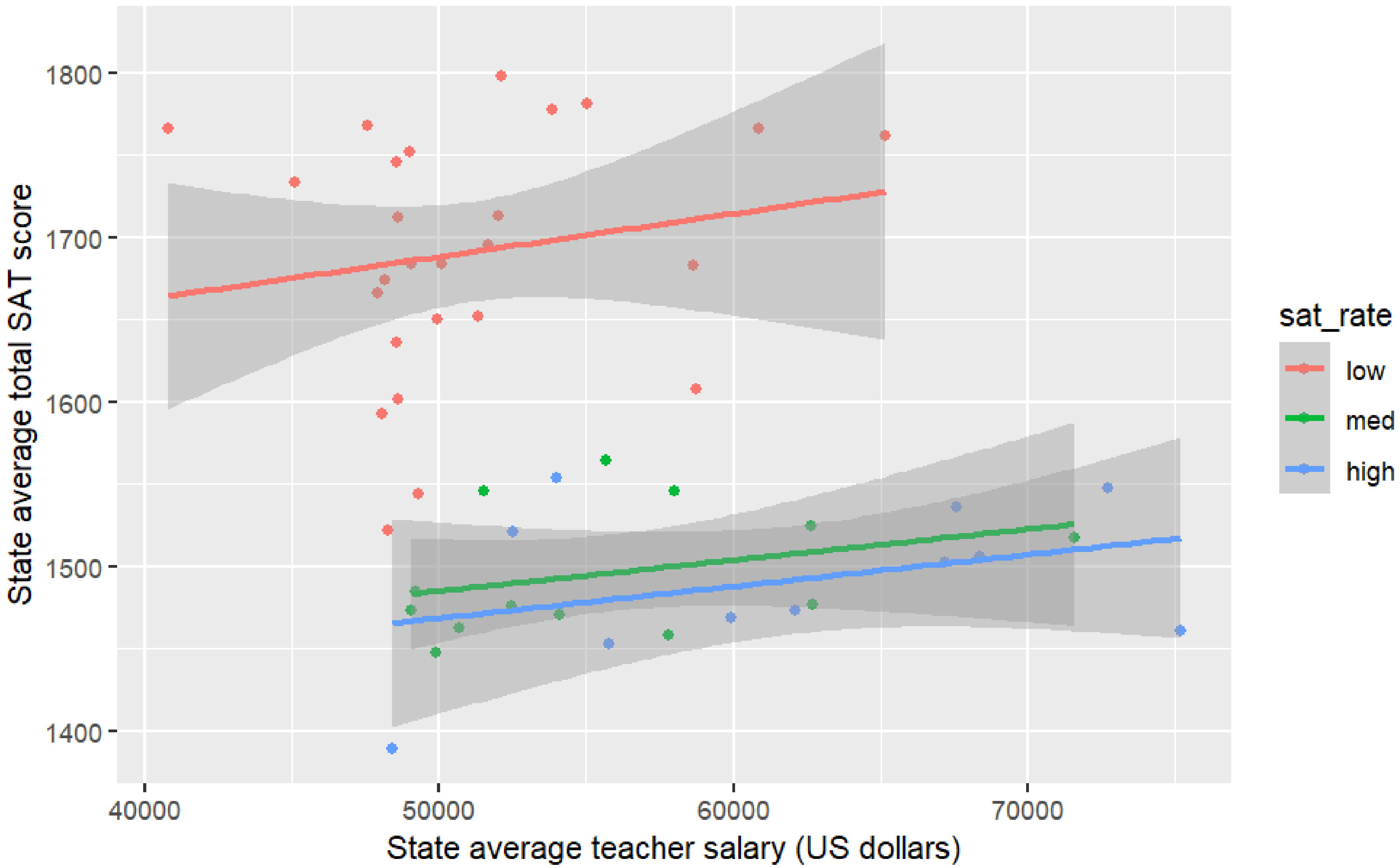
Hide

```
SAT_2010 %>%
    ggplot(aes(x = salary, y = total, color = sat_rate)) +
    geom_point() +
    geom_smooth() +
    # geom_smooth(method = "lm") +
    xlab("State average teacher salary (US dollars)") +
    ylab("State average total SAT score")

SAT_2010 %>%
    ggplot(aes(x = salary, y = total, color = sat_rate)) +
    geom_point() +
    geom_smooth(method = "lm") +
    xlab("State average teacher salary (US dollars)") +
    ylab("State average total SAT score")
```

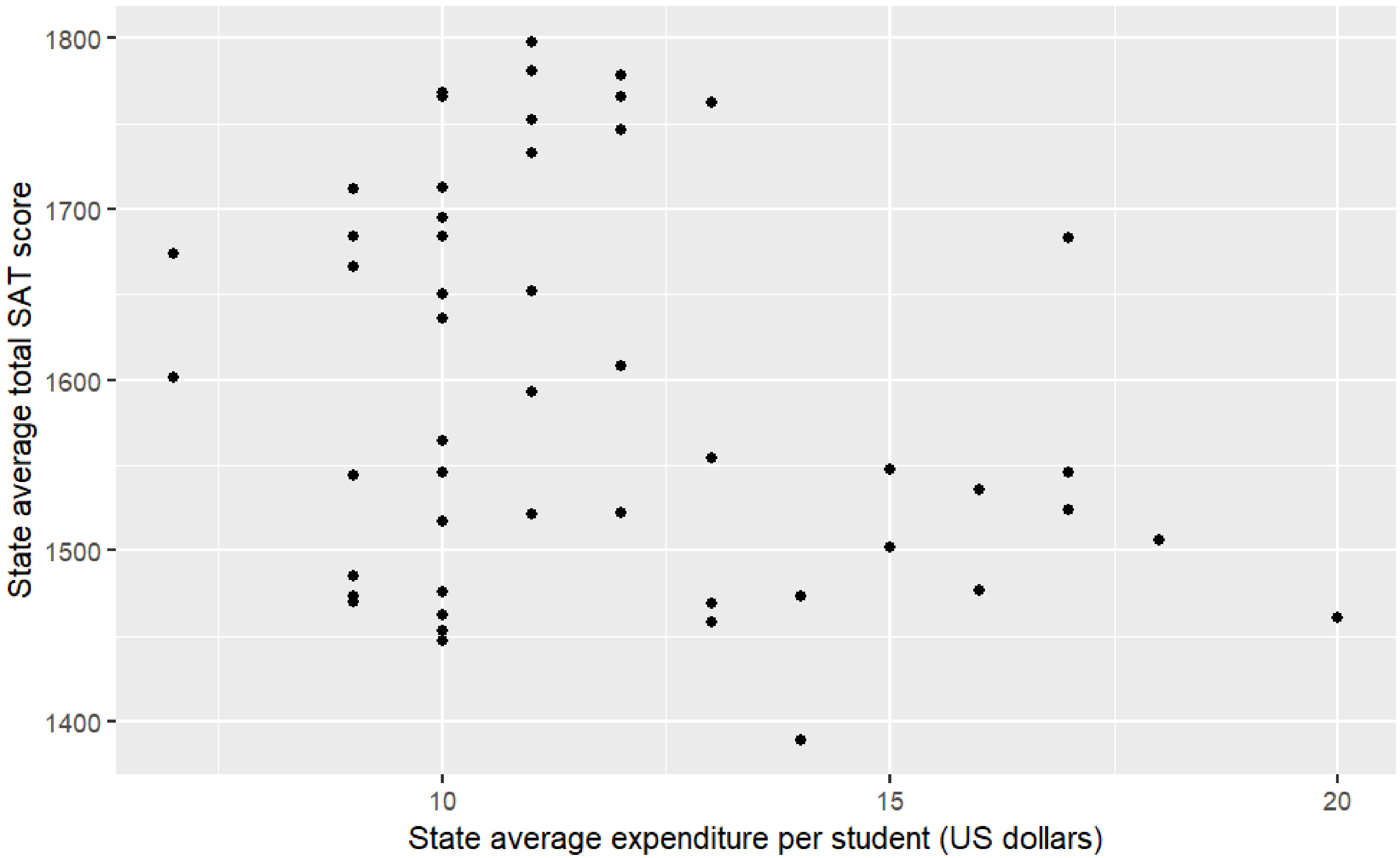
Hide

```
SAT_2010 %>%
    ggplot(aes(x = expenditure, y = total)) +
    geom_point() +
    # geom_smooth() +
    # geom_smooth(method = "lm") +
    xlab("State average expenditure per student (US dollars)") +
    ylab("State average total SAT score")
```


State average total SAT score
1800
1700
1600
1500
1400
40000
50000
60000
70000
State average teacher salary (US dollars)
sat_rate
low
med
high

Hide

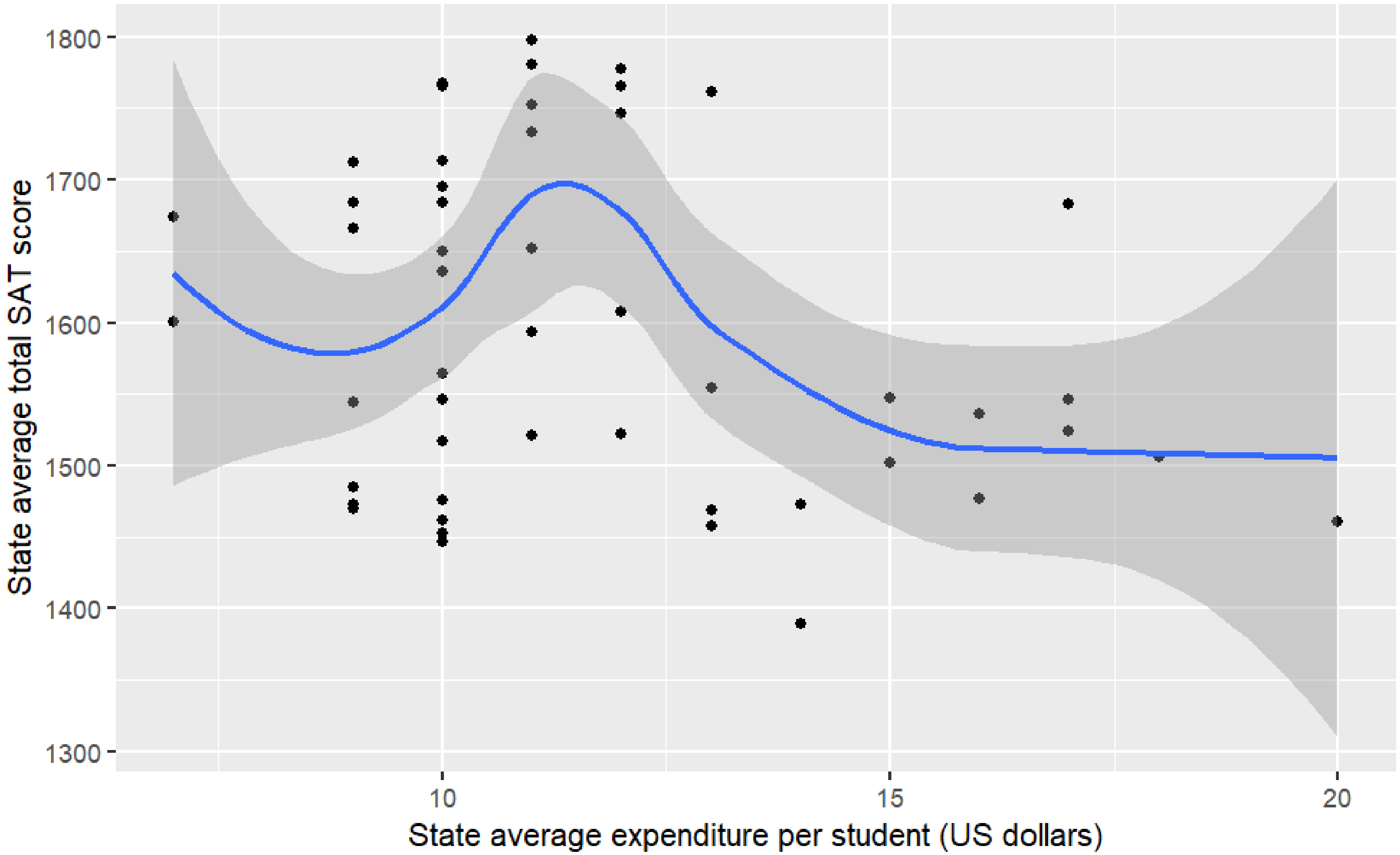
```r
SAT_2010 %>%
    ggplot(aes(x = expenditure, y = total)) +
    geom_point() +
    geom_smooth() +
    xlab("State average expenditure per student (US dollars)") +
    ylab("State average total SAT score")
```



Hide

```
# since we have state data, maybe we should map it!
library(mosaic)
mUSMap(SAT_2010, key = "state", "fill" = "sat_rate")
```

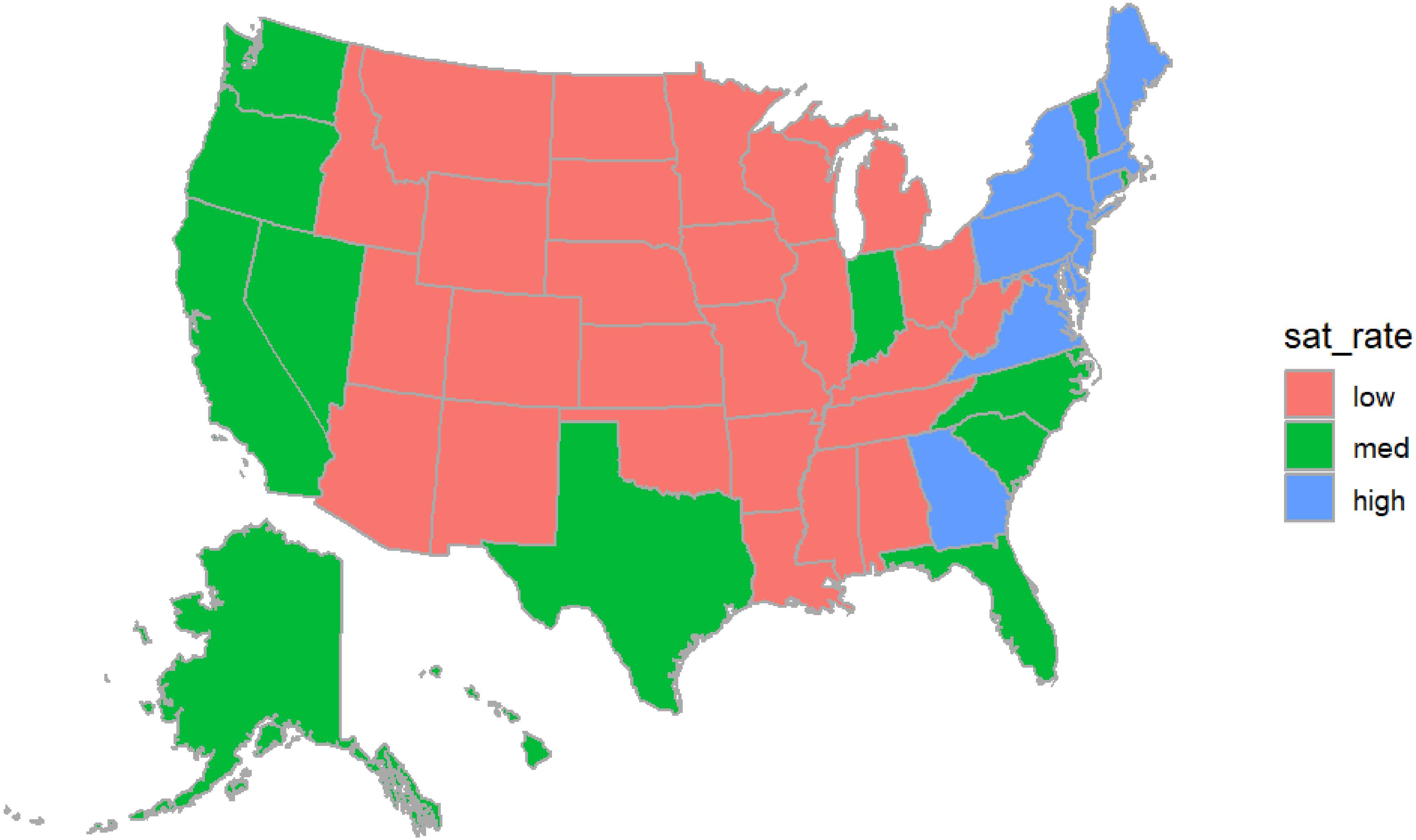
```
Mapping API still under development and may change in future releases.
```



Hide

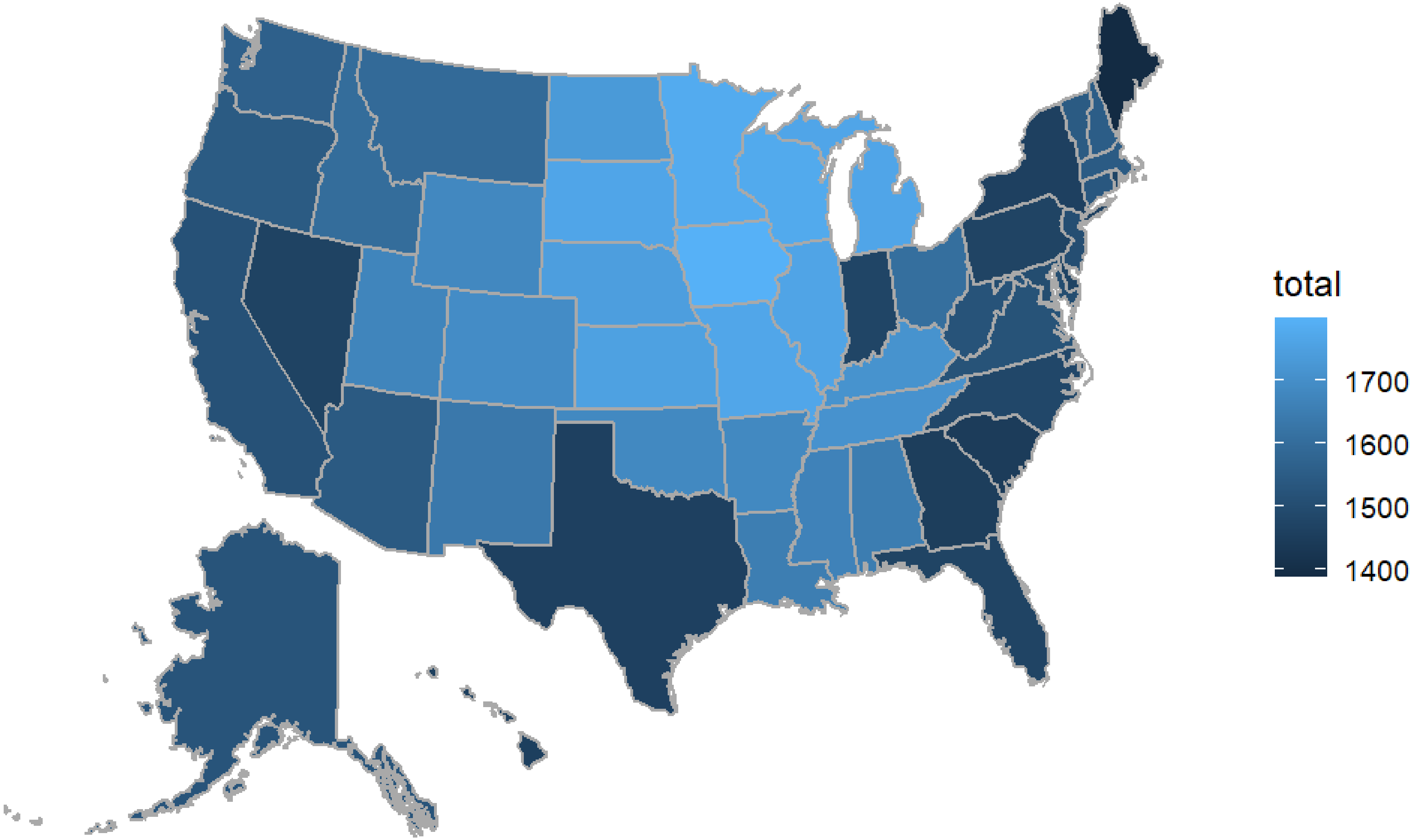
```
mUSMap(SAT_2010, key = "state", "fill" = "total")
```

```
Mapping API still under development and may change in future releases.
```



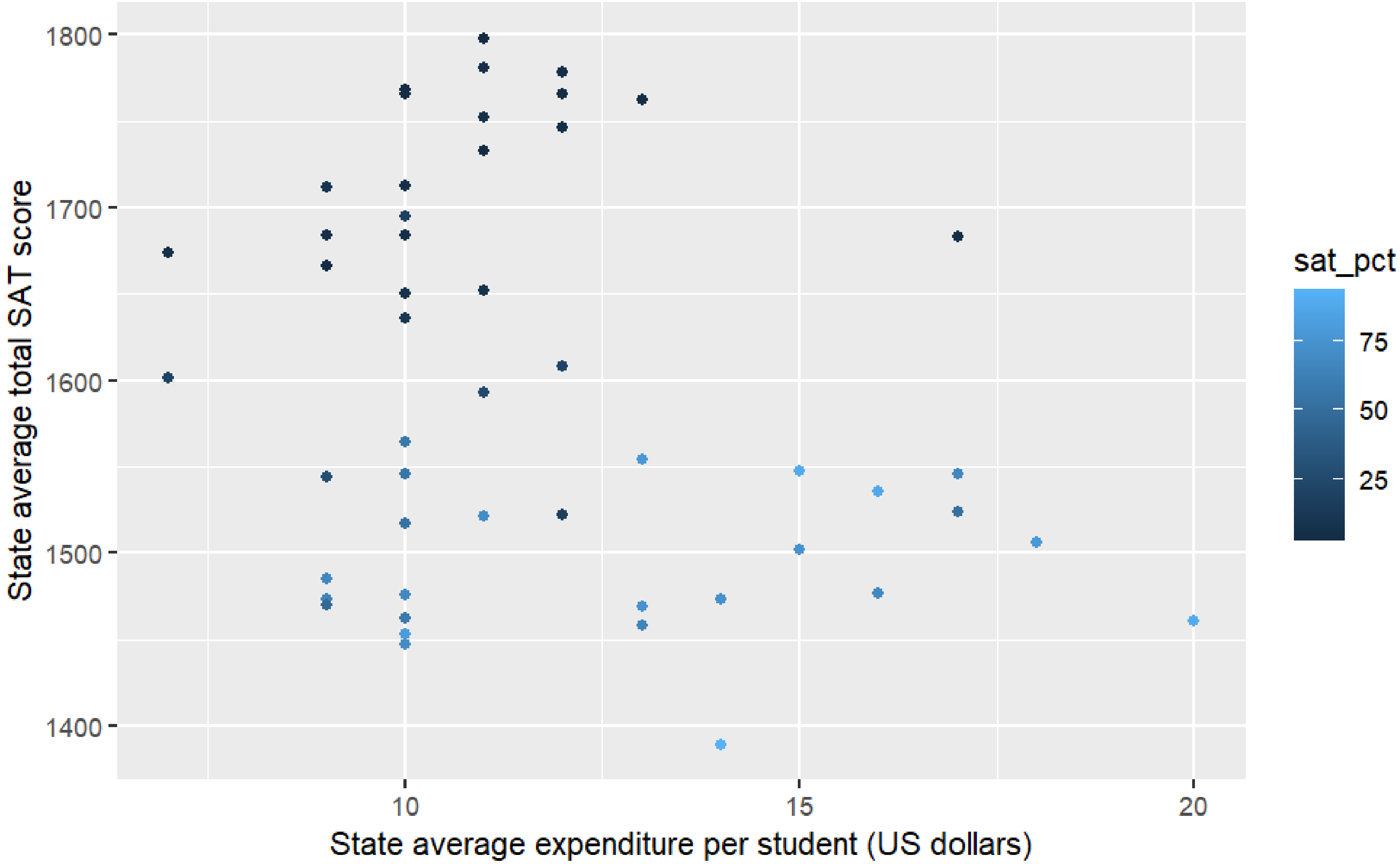
Hide

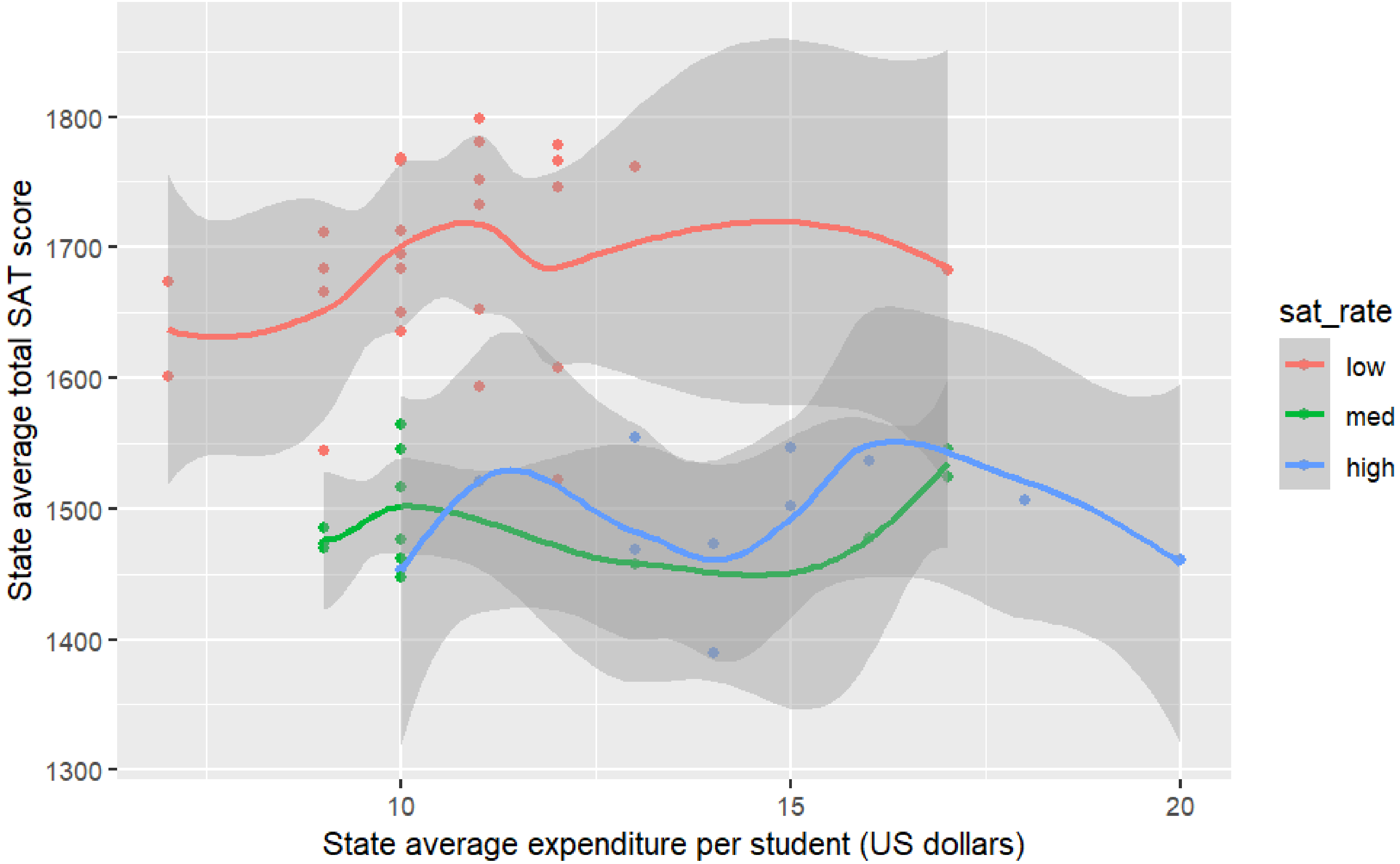
```
## relationship between expenditure and salary
SAT_2010 %>%
    ggplot(aes(x = expenditure, y = total, color = sat_pct)) +
    geom_point() +
    # geom_smooth() +
    # geom_smooth(method = "lm") +
    xlab("State average expenditure per student (US dollars)") +
    ylab("State average total SAT score")
```



Hide

```
SAT_2010 %>%
    ggplot(aes(x = expenditure, y = total, color = sat_rate)) +
    geom_point() +
    geom_smooth() +
    xlab("State average expenditure per student (US dollars)") +
    ylab("State average total SAT score")
```


1800
1700
1600
1500
1400
State average total SAT score
10
15
20
State average expenditure per student (US dollars)
sat_pct
75
50
25



Statistical modeling

After we have completed a thorough EDA, we are ready for inferential or predictive modeling.

Again, statistical modeling can definitely serve exploratory and descriptive purposes that are appropriate during EDA (e.g., we fit smoothers & regression lines above), but they do impose a kind of structure on the data that influences (biases) our expectations. It's a good idea to learn as much as we can about the data while imposing as little structure as possible, and then gradually adding more structure to progressively refine our understanding.

Ideally, we want to let the data speak for itself, and then use appropriate analytical results like models to simply refine interpretations/predictions and more precisely quantify the uncertainty of our conclusions.

A cool side note (if we have time)

- `mUSMap` is part of the `mosaic` package, not the `tidyverse` .
- There is a way to plot maps with `ggplot` but it is slightly more complicated syntax than `mUSMap` .
- There are 2 types of ggplot maps
    - polygon maps
        - you can create very simple maps
        - this is essentially treating each border as a shape, then mapping the shape to each geographic coordinate (latitude and longitude)
        - simple “longitude-latitude” data format is not usually used in real world mapping
    - simple features maps
        - can make beautiful maps with lots and lots of features
            - can handle map projections, labels, colors, adding additional points (like cities), and so on
        - much more versatile (at the price of complexity)
        - uses vector data maps (GIS) https://en.wikipedia.org/wiki/Vector_Map (https://en.wikipedia.org/wiki/Vector_Map)
            - standard by the Open Geospatial Consortium
        - uses the `sf` package
        - essentially, data contains 2 columns, first column is location name, second column contains the “sf” polygon information (any additional columns are characteristics of the location)
            - these data sets are extremely tedious (and complex) to write. It is usually better to find someone else’s data set and adapt it to your needs. The `rnaturalearth` package is a good place to start has countries and US states. The `sf` package has some starter data sets for playing around with.
    - “raster” maps
        - uses geo-spatial data
        - “Unlike the simple features format, in which geographical entities are specified in terms of a set of lines, points and polygons, rasters take the form of images.”
        - think satellite images
- Here is a tutorial https://ggplot2-book.org/maps.html (https://ggplot2-book.org/maps.html) on polygon, sf, and raster maps.

Here are the basic steps for a simple features map of our state data colored by pupil-teacher-ratio:

Hide

```
library(sf)
```

```
Warning: package ‘sf’ was built under R version 4.2.3
```

Hide

```
library(ggspatial)
```

```
Warning: package ‘ggspatial’ was built under R version 4.2.3
```

Hide

```
library(rnaturalearth)
```

```
Warning: package ‘rnaturalearth’ was built under R version 4.2.3
```

Hide

```
library(tidygeocoder)
```

```
Warning: package ‘tidygeocoder’ was built under R version 4.2.3
```

Hide

```
library(maps)
```

```
Warning: package ‘maps’ was built under R version 4.2.3
```

Hide

```
library(ggrepel)
```

```
Warning: package ‘ggrepel’ was built under R version 4.2.3
```

Hide

```
#Get state map data
state_map_data <- map('state', fill = TRUE, plot = FALSE) %>% st_as_sf()

#inspect state map data
class(state_map_data)
```

```
[1] "sf"         "data.frame"
```

Hide

```
head(state_map_data)
```

```
Simple feature collection with 6 features and 1 field
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -124.3834 ymin: 30.24071 xmax: -71.78015 ymax: 42.04937
Geodetic CRS:  +proj=longlat +ellps=clrk66 +no_defs +type=crs
                     ID                           geom
alabama         alabama MULTIPOLYGON (((-87.46201 3...
arizona         arizona MULTIPOLYGON (((-114.6374 3...
arkansas       arkansas MULTIPOLYGON (((-94.05103 3...
california   california MULTIPOLYGON (((-120.006 42...
colorado       colorado MULTIPOLYGON (((-102.0552 4...
connecticut connecticut MULTIPOLYGON (((-73.49902 4...
```

Hide

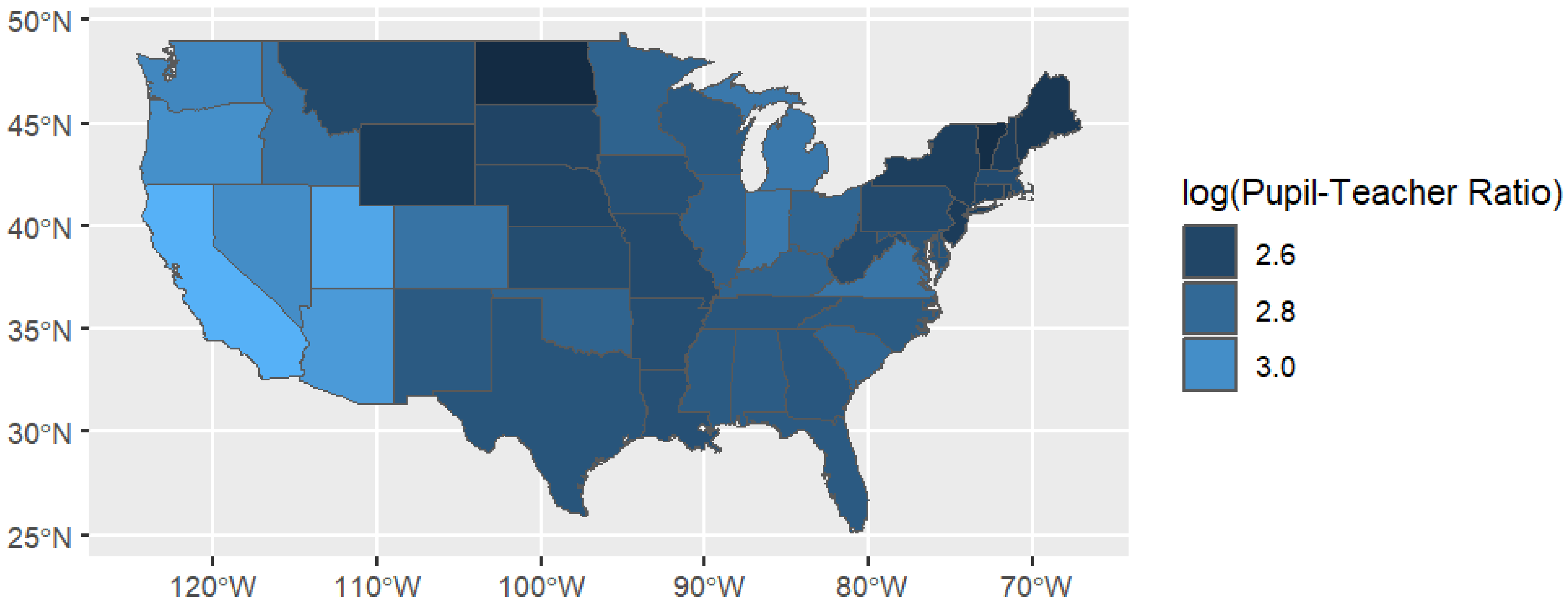
```
# Merge it with our SAT data
SAT2 <- SAT_2010 %>%
  mutate(state = tolower(state)) %>%
  filter(!(state) %in% c("alaska", "hawaii"))

state_map_data <- state_map_data %>%
  filter(ID != "district of columbia")

#color on the log scale
state_map_data$color <- log(SAT2$pupil_teacher_ratio)

#make the ggplot
gg_sat <-  ggplot() +
  geom_sf(data = state_map_data, aes(fill = color)) +
  guides(fill=guide_legend(title="log(Pupil-Teacher Ratio)"))

gg_sat
```



This is great! But it is essentially the exact same map as using `usMap` but with a ton more work.

A few notes:

- I personally prefer to plot color on the log scale (usually but not always). This is not necessary, but I find it easier to see broad trends as it dampens the extremity of the maximum. Not everyone prefers this method.

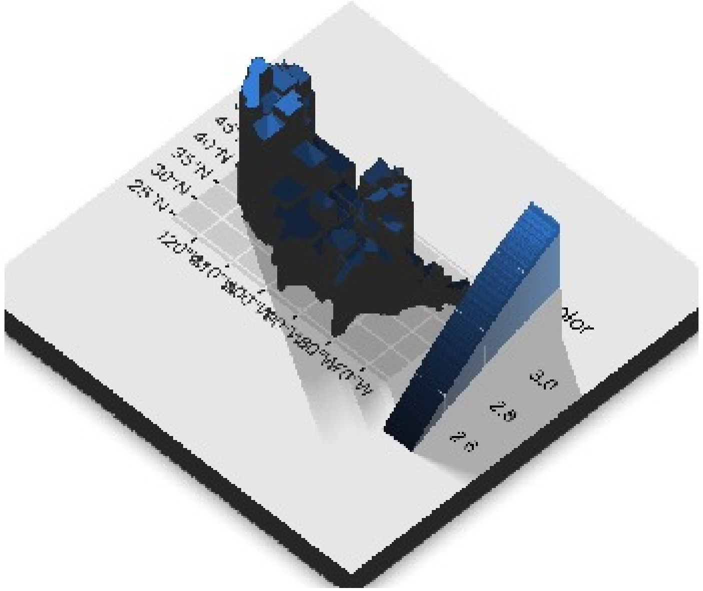
- Notice the order on the legend goes smallest at the top to largest at the bottom. This is counterintuitive. If I were to add this graphic to a report, I would make sure the legend went largest to smallest.
- Notice this map and the `mUSMap` have different map projections. `mUSMap` defaults to a polyconic projection, and `geom_sf` defaults to whatever projection your data frame is in (here it using a Mercator projection). Both methods allow you to change the map projection.

Side, Side note about plotting in 3D

With this ggplot and sf method of plotting maps, it is possible to plot your maps in 3D using the `rayshader` package.

Hide

```
#devtools::install_github("tylermorganwall/rayshader")
library(rayshader)
plot_gg(gg_sat, multicore = TRUE,
        scale = 300,  zoom = 0.75,
        phi = 50, sunangle = -60, theta = 45)
render_snapshot()
```



While this may look super cool, did plotting in the 3D add to our interpretation of the graph? Did we just add unnecessary graphics?

In data visualization it is always important to balance glyphs with interpretability. We need to add enough so that the viewer understands the story we are trying to tell, but not so much that the graph is “messy” or “cluttered”.

Examples when adding 3D does add value

- Election Results
    - https://www.arcgis.com/apps/MinimalGallery/index.html?appid=b3d1fe0e8814480993ff5ad8d0c62c32 (https://www.arcgis.com/apps/MinimalGallery/index.html?appid=b3d1fe0e8814480993ff5ad8d0c62c32)
- Population Density
    - https://www.visualcapitalist.com/3d-mapping-the-worlds-largest-population-densities/ (https://www.visualcapitalist.com/3d-mapping-the-worlds-largest-population-densities/)
- Cartography (particularly elevation)
    - `elmat` example https://www.rayshader.com (https://www.rayshader.com)
- Contours and Joint Densities
    - https://plotly.com/r/3d-surface-plots/ (https://plotly.com/r/3d-surface-plots/)
    - https://rviews.rstudio.com/2020/12/14/plotting-surfaces-with-r/ (https://rviews.rstudio.com/2020/12/14/plotting-surfaces-with-r/)
    - http://www.countbio.com/web_pages/left_object/R_for_biology/R_fundamentals/3D_surface_plot_R.html (http://www.countbio.com/web_pages/left_object/R_for_biology/R_fundamentals/3D_surface_plot_R.html)
    - diamonds example https://www.rayshader.com (https://www.rayshader.com)

Assignments

- Reading Quiz DC Ebook Chapter 15 (due Thursday, July 27, 9:59am)
- Suggested Reading : Chapter 17 Regular expressions

A few words about the final project

- Will be individual assignements
- Similar in nature to the Activities that you are doing
- You will need to explore and analyze using EDA atleast 2 different data sets. The primary dataset should not be part of an R package (could be a csv file, could be a dataset hosted on github, could be from a webpage), but the secondary dataset could be anything.
- For now, try to think about interesting topics you might want to explore and where could you find relevant datasets.
- You will need to submit a topic for the final project within August 2.
- For some it will seem daunting to start from scratch looking for one or more “interesting” data sets. There are lots of useful repositories out there. Here are a few links to get you started, but please feel free to use any data that interest you!

  - https://www.springboard.com/blog/free-public-data-sets-data-science-project/ (https://www.springboard.com/blog/free-public-data-sets-data-science-project/)
  - https://www.dataquest.io/blog/free-datasets-for-projects/ (https://www.dataquest.io/blog/free-datasets-for-projects/)
  - https://data.cityofnewyork.us/ (https://data.cityofnewyork.us/)
  - http://www.icpsr.umich.edu/icpsrweb/ICPSR/ (http://www.icpsr.umich.edu/icpsrweb/ICPSR/)
  - https://github.com/awesomedata/awesome-public-datasets (https://github.com/awesomedata/awesome-public-datasets)
  - https://github.com/fivethirtyeight/data (https://github.com/fivethirtyeight/data)
- I repeat: **you can use any data set you want and it may not be in the above list. The list is just a starting point** .