STAT 231: Problem Set 1A

Joshua Kim

due by 5 PM on Monday, February 22

In order to most effectively digest the textbook chapter readings – and the new R commands each presents – series A homework assignments are designed to encourage you to read the textbook chapters actively and in line with the textbook's Pro Tip on page 33:

"**Pro Tip**: If you want to learn how to use a particular command, we highly recommend running the example code on your own"

A more thorough reading and light practice of the textbook chapter prior to class allows us to dive quicker and deeper into the topics and commands during class. Furthermore, learning a programming lanugage is like learning any other language – practice, practice, practice is the key to fluency. By having two assignments each week, I hope to encourage practice throughout the week. A little coding each day will take you a long way!

Series A assignments are intended to be completed individually. While most of our work in this class will be collaborative, it is important each individual completes the active readings. The problems should be straightforward based on the textbook readings, but if you have any questions, feel free to ask me!

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps1A.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps1A.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

1. GDP and education

a.

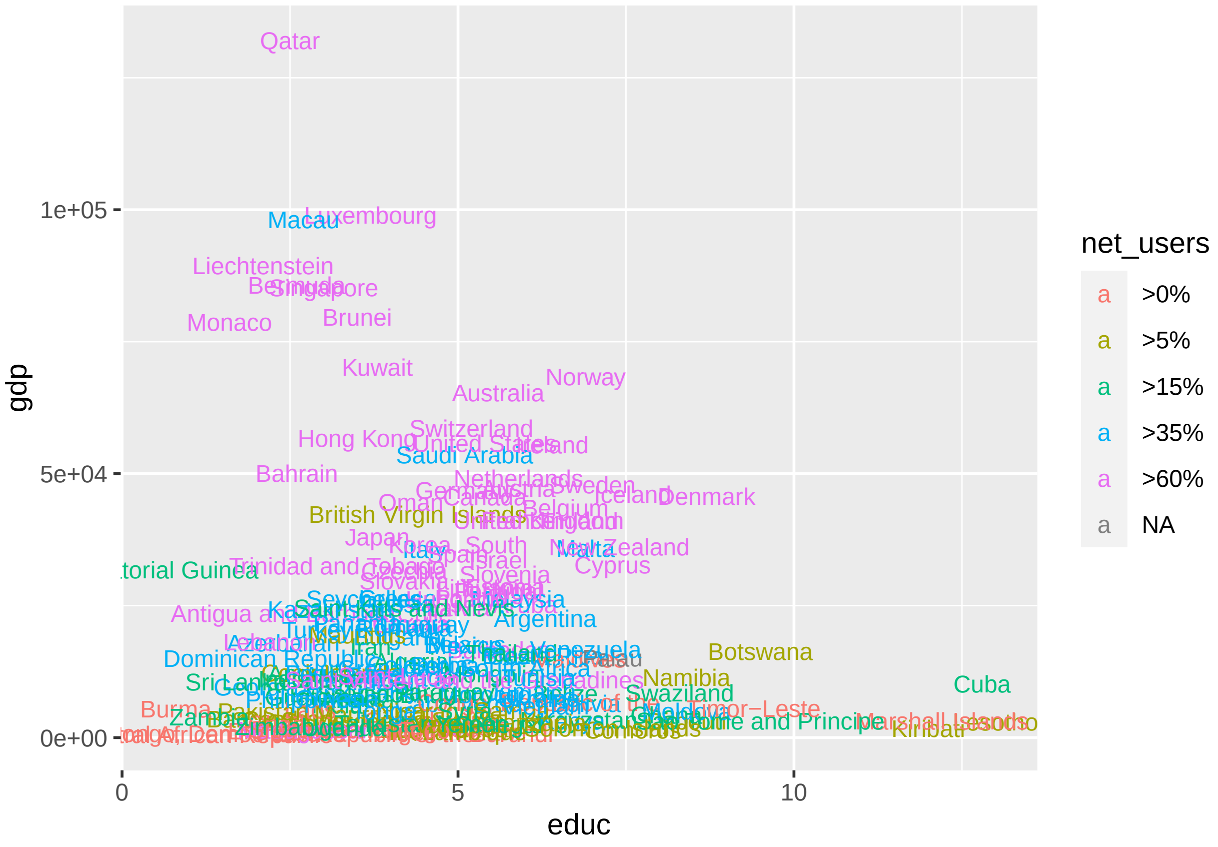
Figure 3.3 in Section 3.1.1 shows a scatterplot that uses both location and label as aesthetics. Reproduce this figure. Hint: you'll need to define 'g' based on code from earlier in Section 3.1.1.

```r
data(CIACountries)

# define the plot object
g <- ggplot (data = CIACountries, aes(y = gdp, x= educ))
geom_point(size = 3)
```

```
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

```r
# print the plot
g + geom_text(aes(label = country, color = net_users), size =3)
```

```
## Warning: Removed 64 rows containing missing values (geom_text).
```



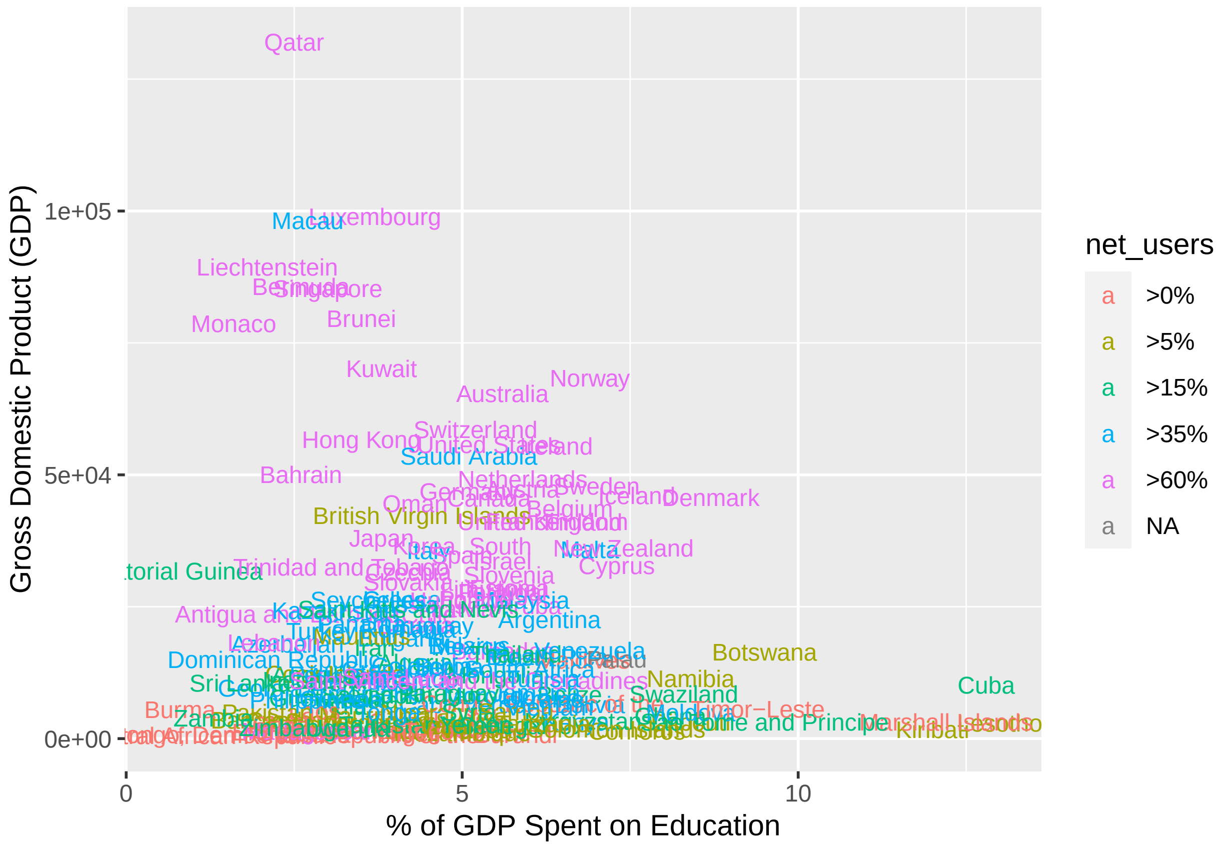
b.

Now, update the plot with more informative labels. Label the x-axis "% of GDP spent on education" and the y-axis "Gross Domestic Product (GDP)". Hint: see Section 3.2.2 for an example of one way to label the axes.

```r
g <- ggplot(data = CIACountries, aes(y = gdp, x= educ)) +
  xlab("% of GDP Spent on Education") +
  ylab("Gross Domestic Product (GDP)")
```

```
g + geom_text(aes(label = country, color = net_users), size =3)
```

```
## Warning: Removed 64 rows containing missing values (geom_text).
```

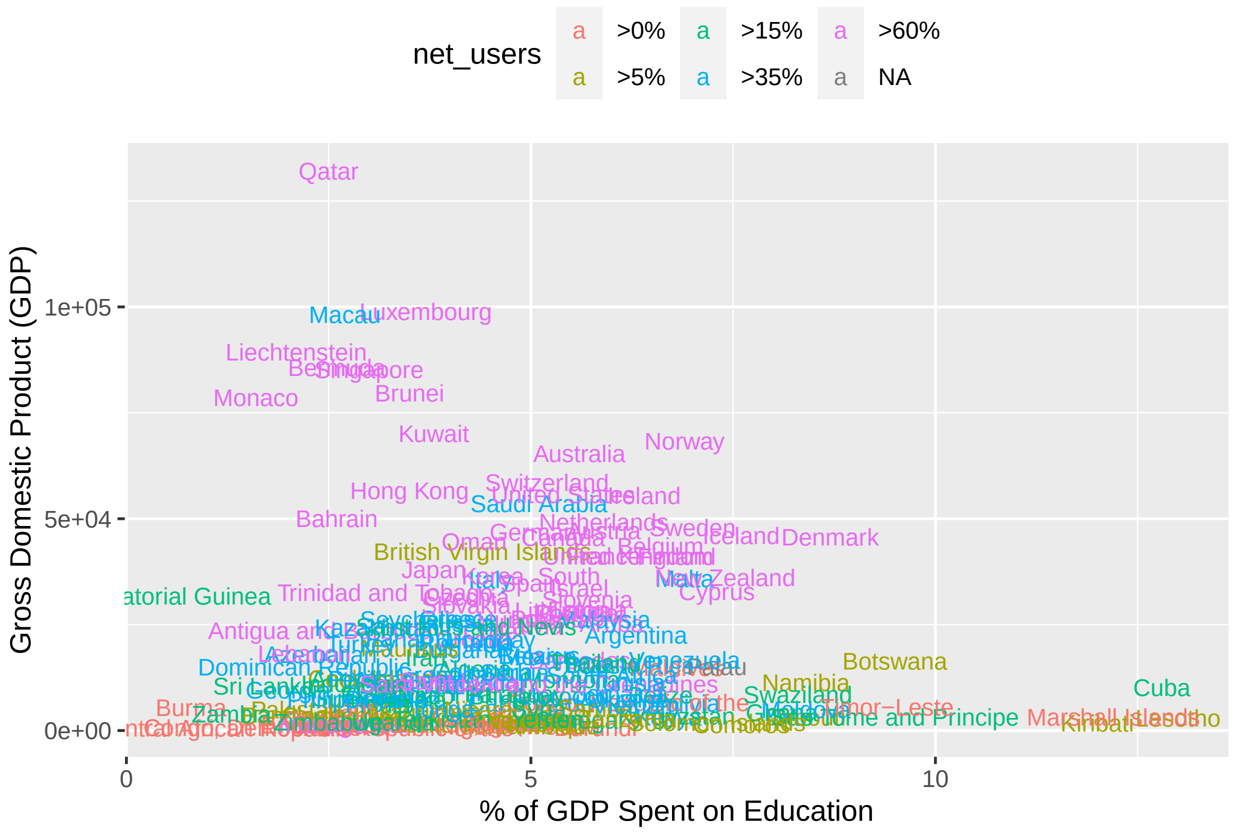


c.

Next, move the legend so that it's located on the top of the plot as opposed to the right of the plot. Hint: see Section 3.1.4 for an example on how to change the legend position.

```
g <- ggplot(data = CIACountries, aes(y = gdp, x= educ)) +
  xlab("% of GDP Spent on Education") +
  ylab("Gross Domestic Product (GDP)") +
  theme(legend.position = "top")


g + geom_text(aes(label = country, color = net_users), size =3)
```

```
## Warning: Removed 64 rows containing missing values (geom_text).
```

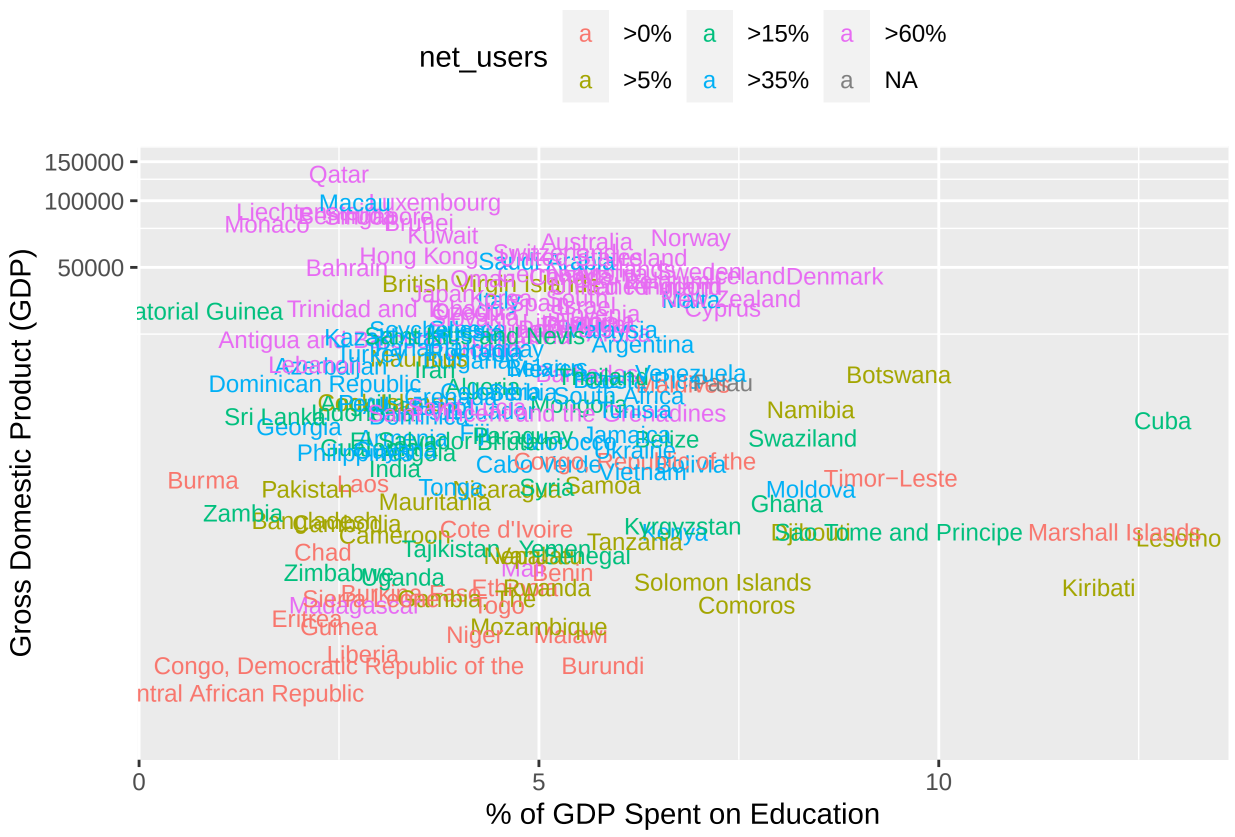


d.

Lastly, Section 3.1.2 discusses *scale*, and demonstrates how to display GDP on a logarithmic scale to better discern differences in GDP. Update the figure so GDP is on a log10 scale.

```
g <- ggplot(data = CIACountries, aes(y = gdp, x= educ)) +
  xlab("% of GDP Spent on Education") +
  ylab("Gross Domestic Product (GDP)") +
  theme(legend.position = "top") +
  coord_trans(y = "log10")


g + geom_text(aes(label = country, color = net_users), size =3)
```

```
## Warning: Removed 64 rows containing missing values (geom_text).
```


net_users
>0%
>5%
>15%
>35%
>60%
NA
Gross Domestic Product (GDP)
% of GDP Spent on Education
150000
100000
50000
0
5
10
Qatar
Luxembourg
Macau
Monaco
Brunei
Kuwait
Norway
Australia
Bahrain
Hong Kong
Saudi Arabia
Denmark
New Zealand
Cyprus
Malta
Trinidad and Tobago
Argentina
Botswana
Dominican Republic
Sri Lanka
Georgia
Namibia
Swaziland
Cuba
Burma
Pakistan
Laos
India
Tonga
Samoa
Moldova
Timor−Leste
Ghana
Zambia
Mauritania
Cote d'Ivoire
Kyrgyzstan
Kenya
Tanzania
Djibouti
Sao Tome and Principe
Marshall Islands
Lesotho
Chad
Zimbabwe
Uganda
Benin
Solomon Islands
Comoros
Kiribati
Eritrea
Guinea
Mozambique
Niger
Malawi
Liberia
Congo, Democratic Republic of the
Burundi
Central African Republic

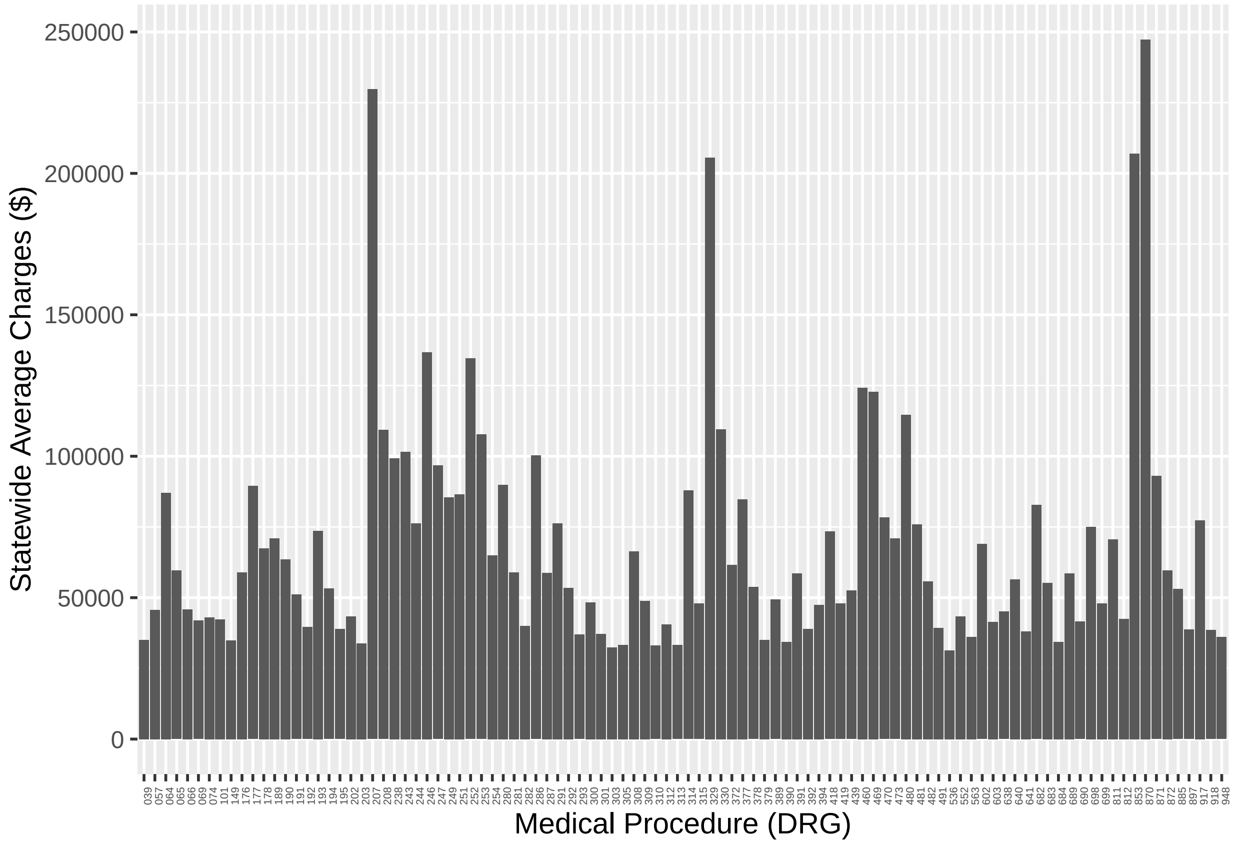
2. Medical procedures

a.

Consider Figure 3.7 in Section 3.2.1. What does `reorder(drg, mean_charge)` do? Recreate the plot, but use `x = drg` instead of `x = reorder(drg, mean_charge)`. What happens?

> ANSWER: `reorder(drg, mean_charge)` literally reorders the plot to be in order of average cost for each medical procedure. When I change it to x=drg, the graph is no longer in order of smallest to biggest average cost of the procedure, but the graph instead shows the average price for each medical procedure based on their name and number.

```r
data(MedicareCharges)
ChargesNJ <- MedicareCharges %>%
  ungroup() %>%
  filter(stateProvider == "NJ")

# create the plot object
p <- ggplot(data = ChargesNJ, aes(x = drg, y = mean_charge)) +
  geom_col(fill = "gray") +
  ylab("Statewide Average Charges ($)") +
  xlab("Medical Procedure (DRG)") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = rel(0.5)))


# print the plot
p + geom_col(data = ChargesNJ, size = 1)
```

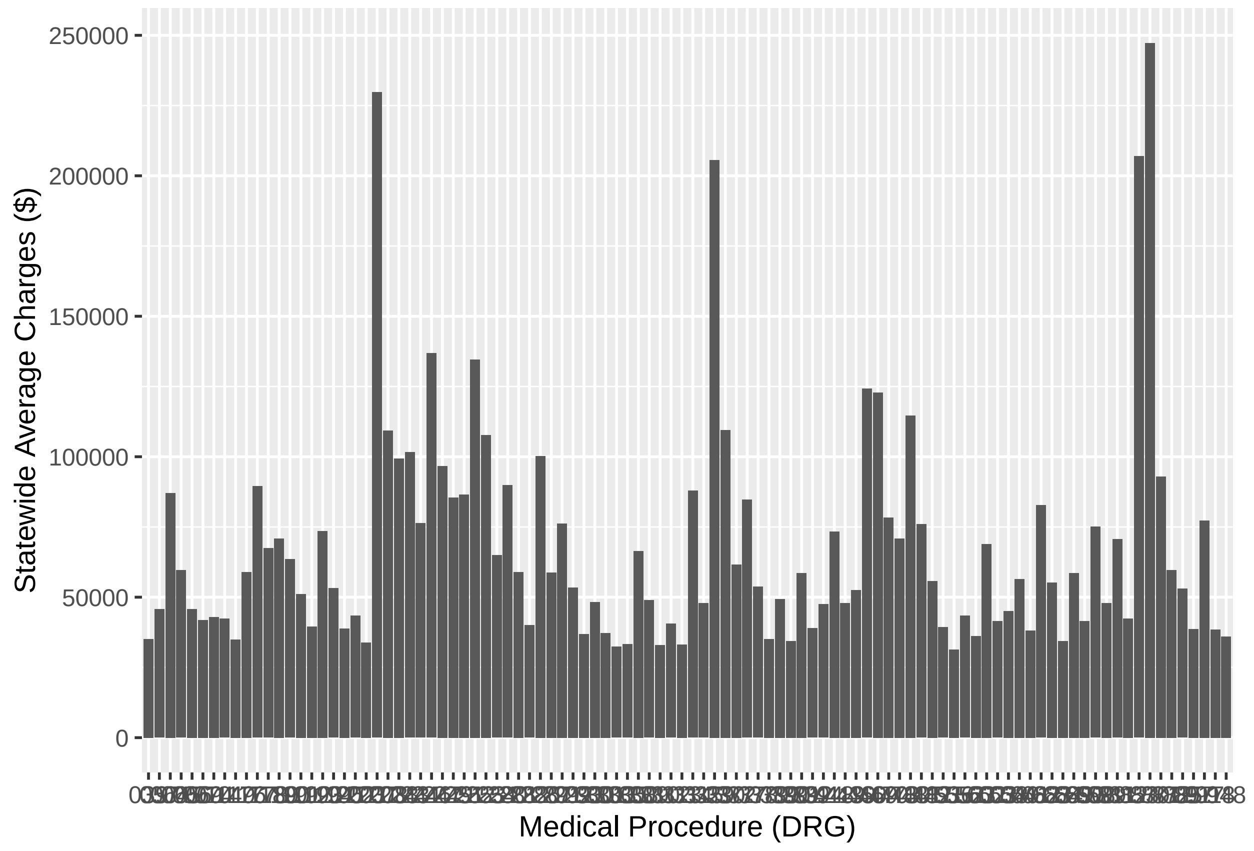


b.

Replace `x = drg` with `x = reorder(drg, mean_charge)`, but also remove the `theme()` line. Now what happens? What was the purpose of the `theme()` line? Hint: You may need to knit the document and look at the pdf to better observe what's happening.

> ANSWER: By removing the theme code, we are able to see that the text on the x axis become much more crowded because the text is no longer vertical, but horizontal.

```
data(MedicareCharges)
ChargesNJ <- MedicareCharges %>%
  ungroup() %>%
  filter(stateProvider == "NJ")

# create the plot object
p <- ggplot(data = ChargesNJ, aes(x = drg, y = mean_charge)) +
  geom_col(fill = "gray") +
  ylab("Statewide Average Charges ($)") +
  xlab("Medical Procedure (DRG)")


# print the plot
p + geom_col(data = ChargesNJ, size = 1)
```

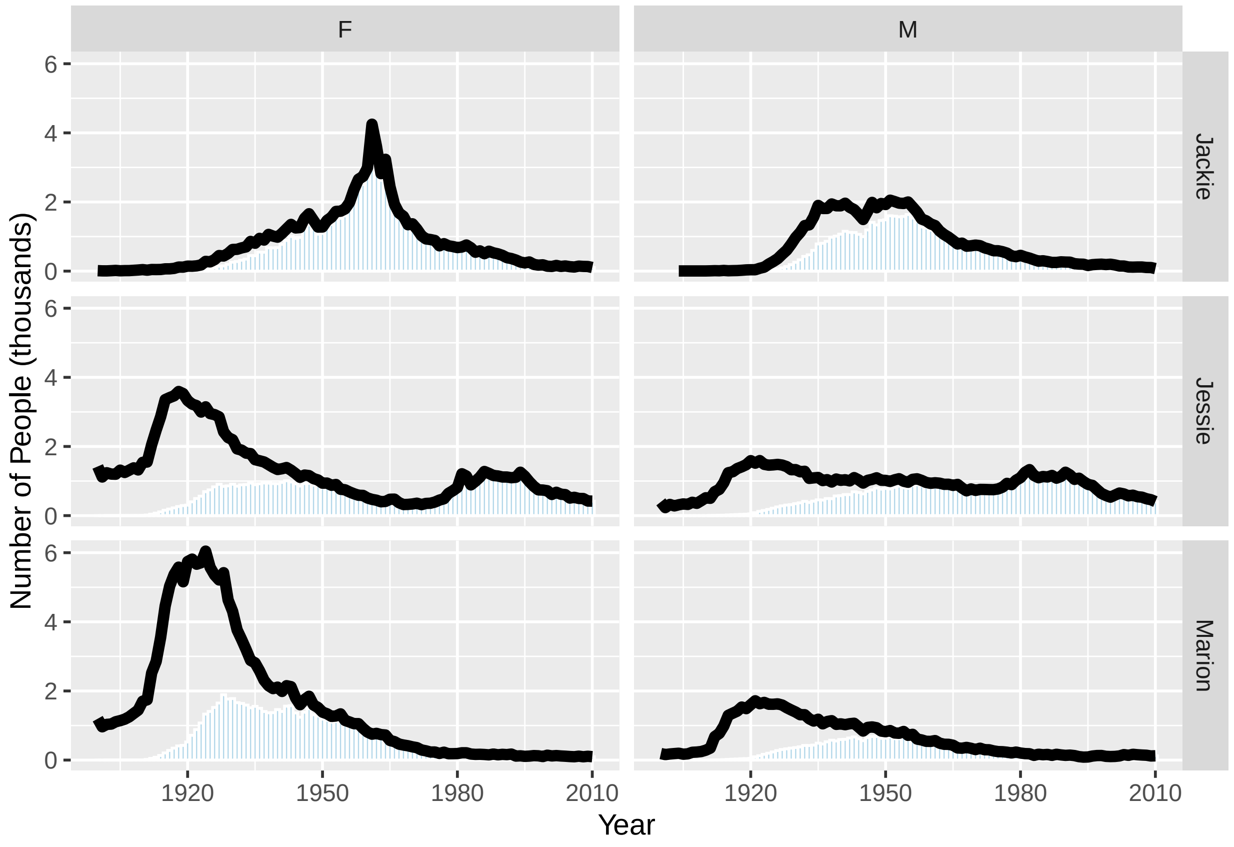


3. Historical baby names

As you read through (and, better yet – code along with (not required, but useful practice!)) – the extended example on historical baby names in section 3.3.1, write down two questions you have about any of the R code used in that example. (Your questions could be about what a specific part of the code – ggplot or not – is actually doing, or a more general question about any of the commands used.) Please be thoughtful about your questions; we will use them (anonymously) in an exercise in class this week.

> ANSWER: Regarding the Hmisc package, I do not really understand why .5 would give the median, and what would it would represent if we changed the value to something like .2. My best guess is that this is the percentile but I wonder how we can find the median and mean since they are likely different values. In addition, I had questions about whether we needed to use libary('package') everytime we run code. I've also learned that the function %>% is defined by dplyr, so I was wondering if we need to run the dplyr package everytime as well. It also makes me wonder what are some other functions that are reliant on specific packages. I was also a little bit unsure of the ifelse function, which I believe highlights one bar, but I would appreciate a bit more explanation on this function as well.

```r
# to get you started following along

library(mdsr)
library(dplyr)
library(babynames)
library(Hmisc)


BabynamesDist <- make_babynames_dist()

joseph <- BabynamesDist %>%
  filter(name == "Joseph" & sex == "M")

name_plot <- ggplot(data = joseph, aes(x = year)) +
  geom_bar(stat = "identity", aes(y = count_thousands * alive_prob), fill = "#b2d7e9", color = "white")
  geom_line(aes(y = count_thousands), size = 2) +
  ylab("Number of People (thousands)") +
  xlab("Year")

many_names_plot <- name_plot +
  facet_grid(name ~ sex)
mnp <- many_names_plot %+% filter(
  BabynamesDist,
  name %in% c("Jessie", "Marion", "Jackie"))
mnp
```


F
M
Jackie
Jessie
Marion
Number of People (thousands)
0
2
4
6
1920
1950
1980
2010
Year