STAT 231: Problem Set 1A

Jack Dove

due by 5 PM on Monday, August 31

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.

```
data(CIACountries)
summary(CIACountries)
```

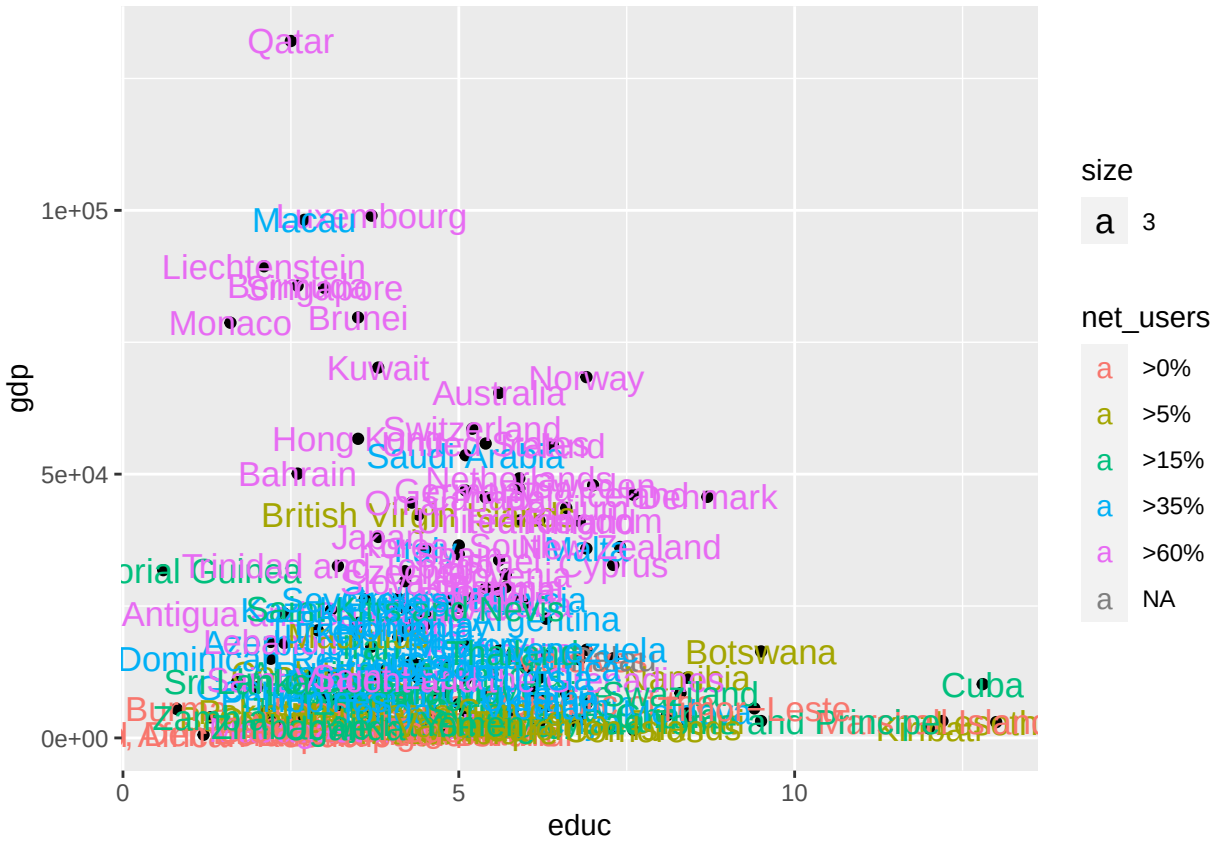
```
##    country                pop                 area              oil_prod
##  Length:236         Min.   :4.800e+01   Min.   :       0   Min.   :       0
##  Class :character   1st Qu.:3.435e+05   1st Qu.:    2498   1st Qu.:       0
##  Mode  :character   Median :5.311e+06   Median :   73580   Median :       0
##                     Mean   :3.075e+07   Mean   :  577876   Mean   :  373125
##                     3rd Qu.:1.835e+07   3rd Qu.:  414643   3rd Qu.:   51130
##                     Max.   :1.367e+09   Max.   :17098242   Max.   :10840000
##                                                            NA's   :23
##       gdp              educ           roadways        net_users
##  Min.   :   400   Min.   : 0.600   Min.   : 0.00639   >0% :24
##  1st Qu.:  4775   1st Qu.: 3.300   1st Qu.: 0.12239   >5% :30
##  Median : 13750   Median : 4.700   Median : 0.33474   >15%:36
##  Mean   : 21398   Mean   : 4.855   Mean   : 1.10849   >35%:59
##  3rd Qu.: 31650   3rd Qu.: 5.900   3rd Qu.: 1.15395   >60%:67
##  Max.   :132100   Max.   :13.000   Max.   :38.50000   NA's:20
##  NA's   :8        NA's   :63       NA's   :13
```

```
# define the plot object
g <- ggplot(data = CIACountries, mapping = aes(x = educ, y = gdp)) +
  geom_point() + geom_text(aes(label=country, color=net_users, size =3))

# print the plot
g
```

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

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



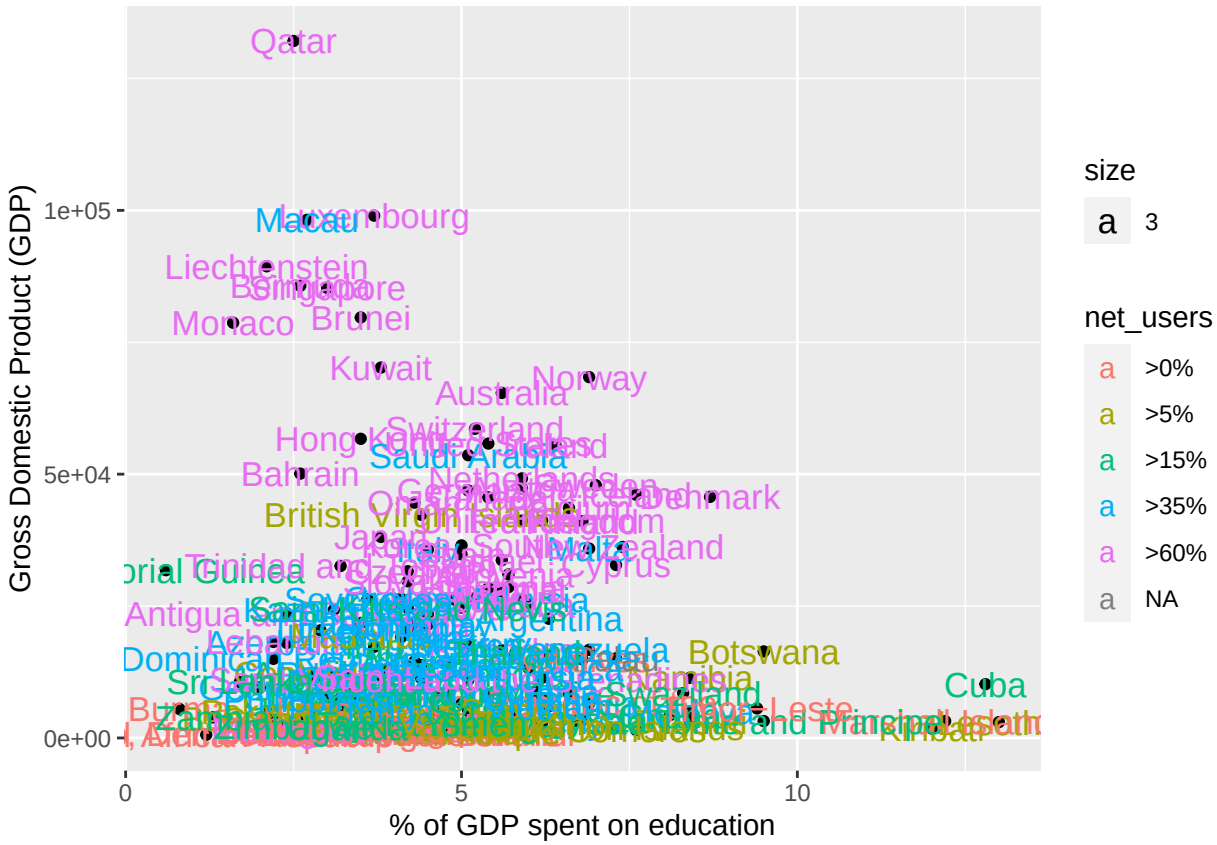
b.

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.

```
gnew <- g + ylab("Gross Domestic Product (GDP)") +xlab("% of GDP spent on education")
gnew
```

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

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



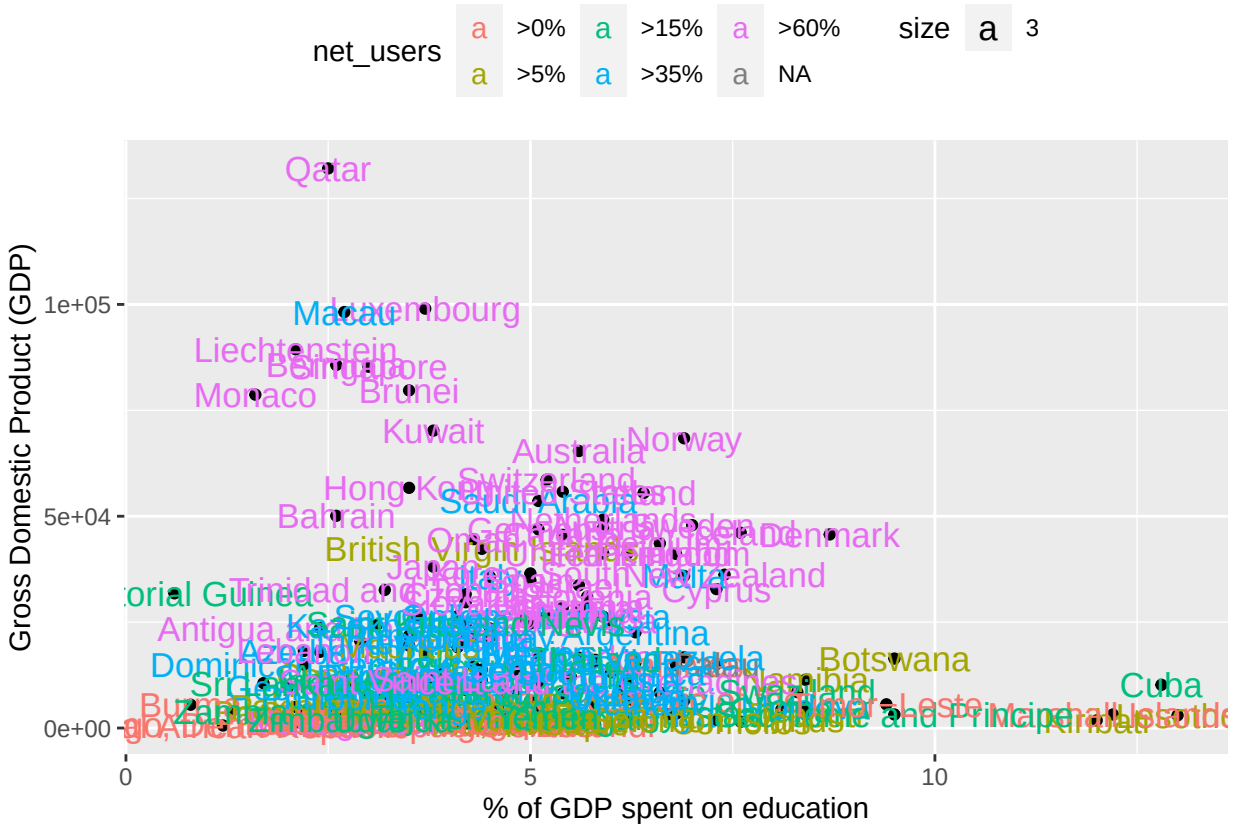
c.

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.

```
gnew2 <- gnew + theme(legend.position = "top")
gnew2
```

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

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



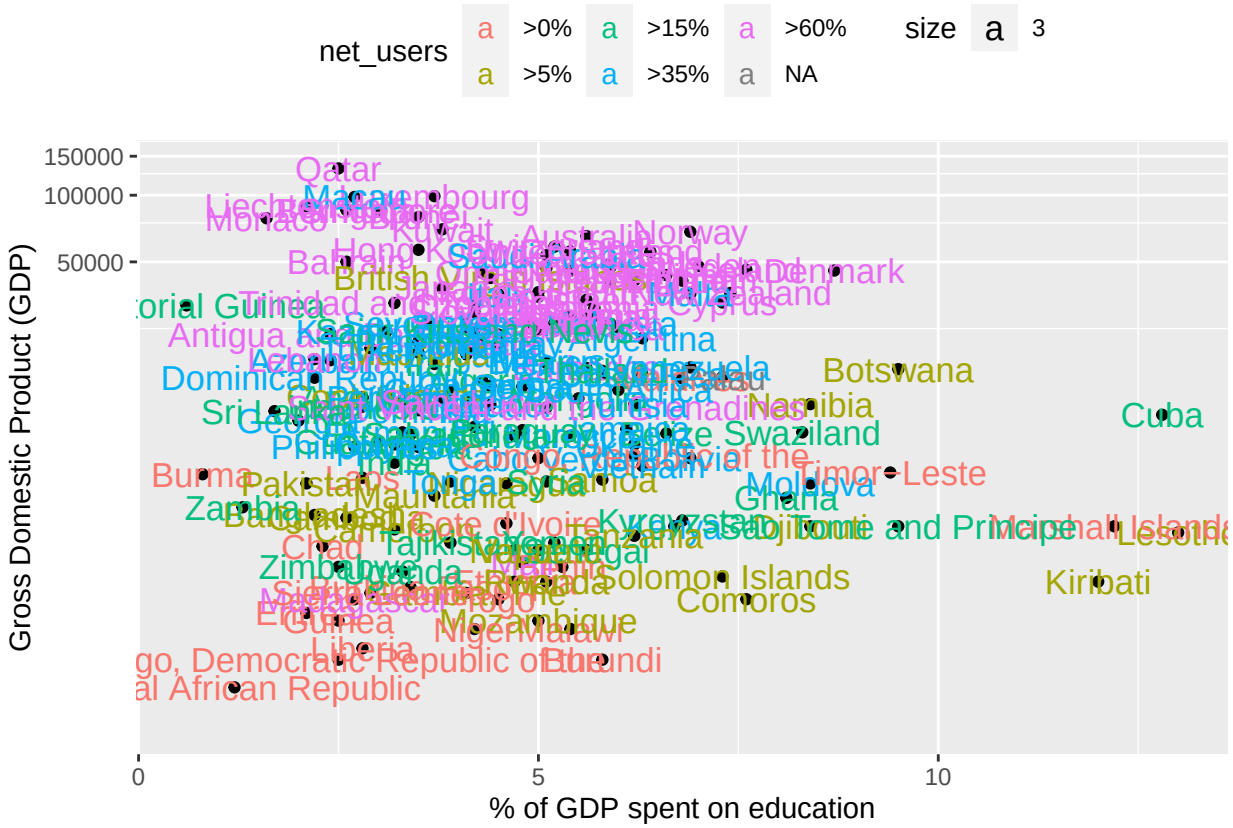
d.

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.

```
gnew3 <- gnew2 + coord_trans(y="log10")
gnew3
```

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

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


net_users
>0%
>5%
>15%
>35%
>60%
NA
size
3
Gross Domestic Product (GDP)
150000
100000
50000
0
5
10
% of GDP spent on education
Qatar
Norway
Denmark
Cyprus
Argentina
Botswana
Namibia
Cuba
Swaziland
Moldova
Ghana
Lesotho
Kiribati
Solomon Islands
Comoros
Mozambique
Liberia
Burundi
Zimbabwe
Zambia
Pakistan
Timor-Leste

2. Medical procedures

a.

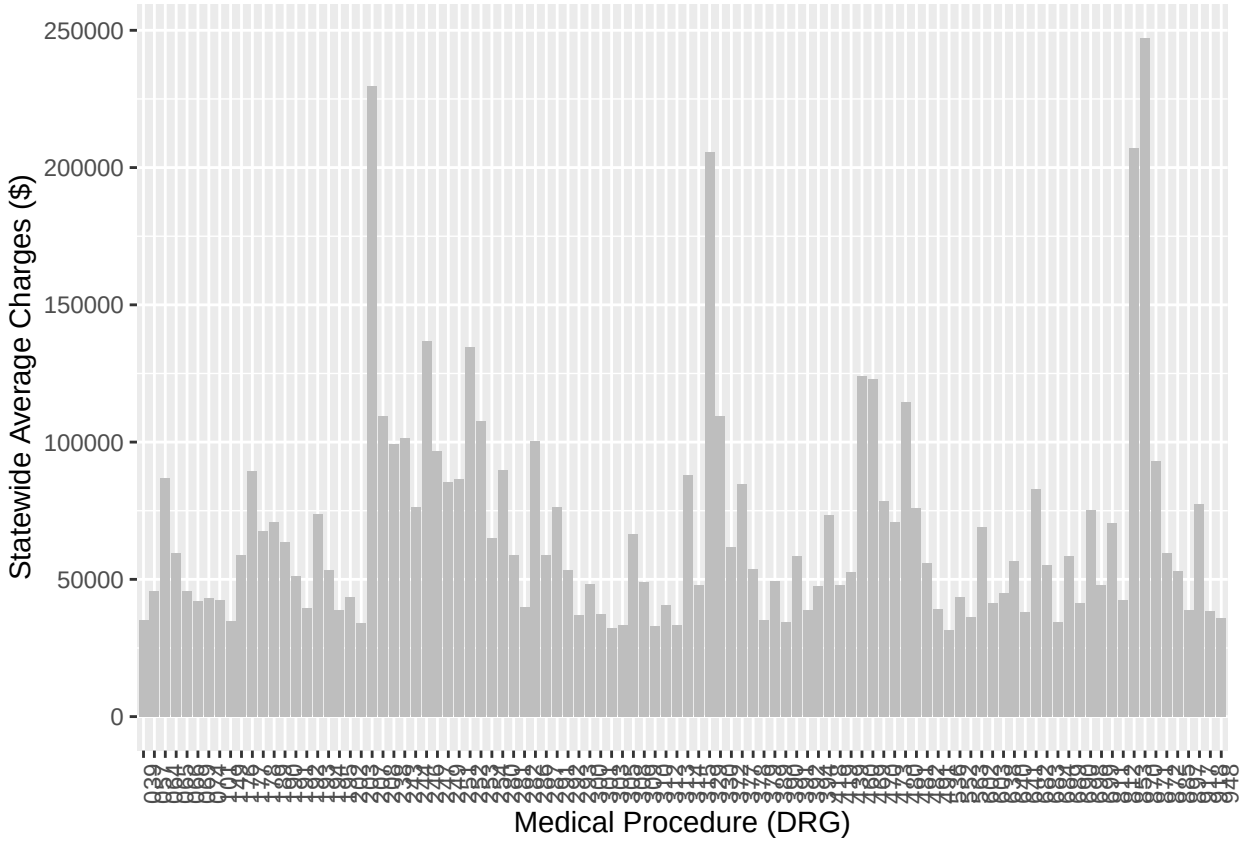
Consider Figure 3.7 in Section 3.2.1. What does `reorder(drg, mean_charge)` do? Recreate the plot, but use `x = drg` instead of `x = reorder(drg, mean_charge)`. What happens?

> ANSWER: The reorder operator sorts the medical procedures from least to greatest mean charge, not based on their medical procedure DRG number.

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

# create the plot object
p <- ggplot(data = ChargesNJ, aes(x=drg, y=mean_charge))+geom_bar(fill="gray", stat="identity")+ylab("S


# print the plot
p
```



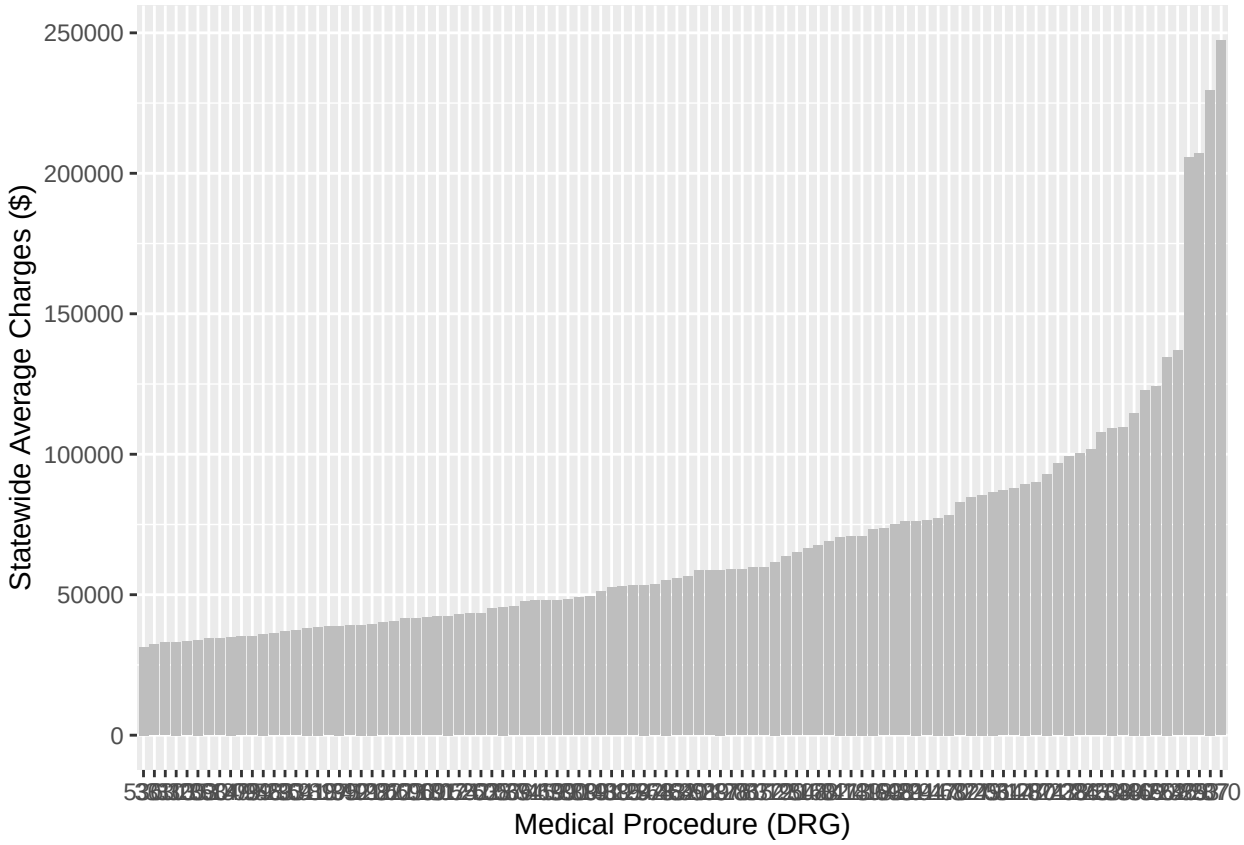
b.

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: The theme line rotates the x axis point labels 90 degrees, allowing them to be legible in a vertical fashion.

```
pnew <- ggplot(data = ChargesNJ, aes(x=reorder(drg, mean_charge), y=mean_charge))+geom_bar(fill="gray",
pnew
```



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: 1. Does the shrinking distance between the blue chart and black line represent an initial boom in baby name assignment, and then the age of that baby name decreasing as time goes on, until the black and blue are contangent? 2. Is the slash n operator a consistent new line creator in R code, or is that only used within ggplot functions?

```
# to get you started following along . . .
library(babynames)
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")

name_plot
```