Reading Set 1

Ziji Zhou

Due by 10pm ET on Monday

Reading Set Information

A more thorough reading and light practice of the textbook reading prior to class allows us to jump into things more quickly in class and dive deeper into topics. As you actively read the textbook, you will work through the Reading Sets to help you engage with the new concepts and skills, often by replicating on your own the examples covered in the book.

These should be completed on your own without help from your peers. 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!

GitHub Workflow

1. Before editing this file, verify you are working on the copy saved in *your* repo for the course (check the filepath and the project name in the top right corner).
2. Before editing this file, make an initial commit of the file to your repo to add your copy of the problem set.
3. Change your name at the top of the file and get started!
4. You should *save, knit, and commit* the .Rmd file each time you've finished a question, if not more often.
5. You should occasionally *push* the updated version of the .Rmd file back onto GitHub. When you are ready to push, you can click on the Git pane and then click **Push**. You can also do this after each commit in RStudio by clicking **Push** in the top right of the *Commit* pop-up window.
6. When you think you are done with the assignment, save the pdf as "*Name_thisfilename_date*.pdf" before committing and pushing (this is generally good practice but also helps me in those times where I need to download all student homework files).

Gradescope Upload

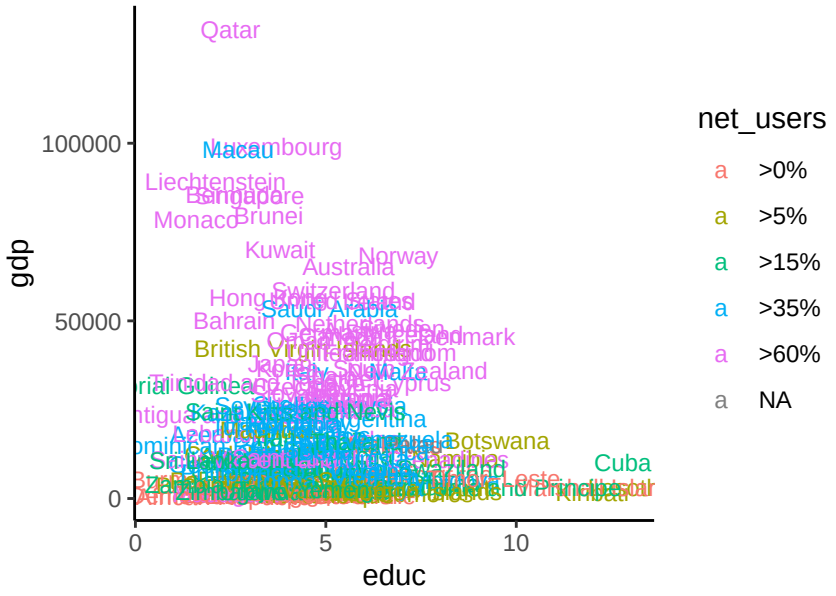
For each question (e.g., 3.1), allocate all pages associated with the specific question. If your work for a question runs onto a page that you did not select, you may not get credit for the work. If you do not allocate *any* pages when you upload your pdf, you may get a zero for the assignment.

You can resubmit your work as many times as you want before the deadline, so you should not wait until the last minute to submit some version of your work. Unexpected delays/crises that occur on the day the assignment is due do not warrant extensions (please submit whatever you have done to receive partial credit).

Problem 1 **GDP and education**

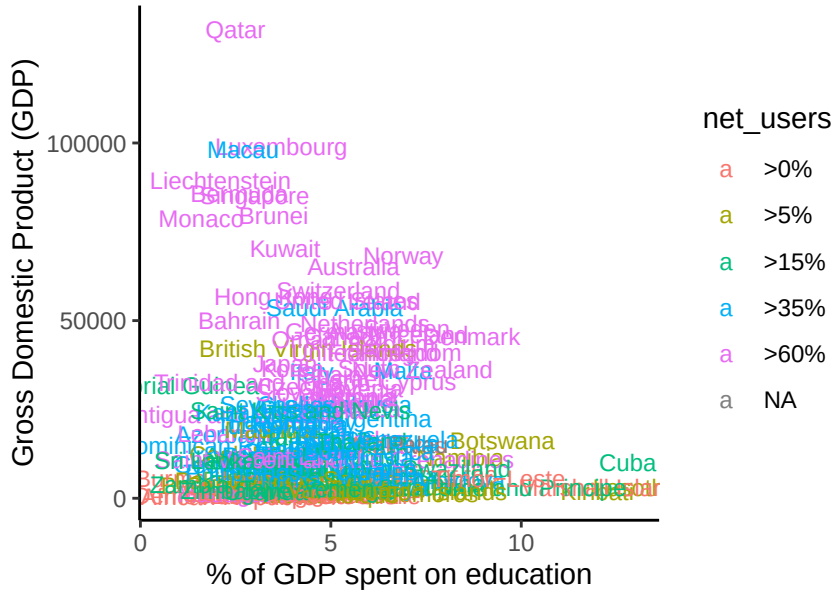
1.1 Figure 3.3 in Section 3.1.1 shows a scatterplot that uses both location and label as aesthetics. Reproduce this figure. *Hint: you'll need to define 'g' based on code from earlier in Section 3.1.1. Also, make sure you load packages in the `setup` chunk as needed!*

```
data(CIACountries)

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



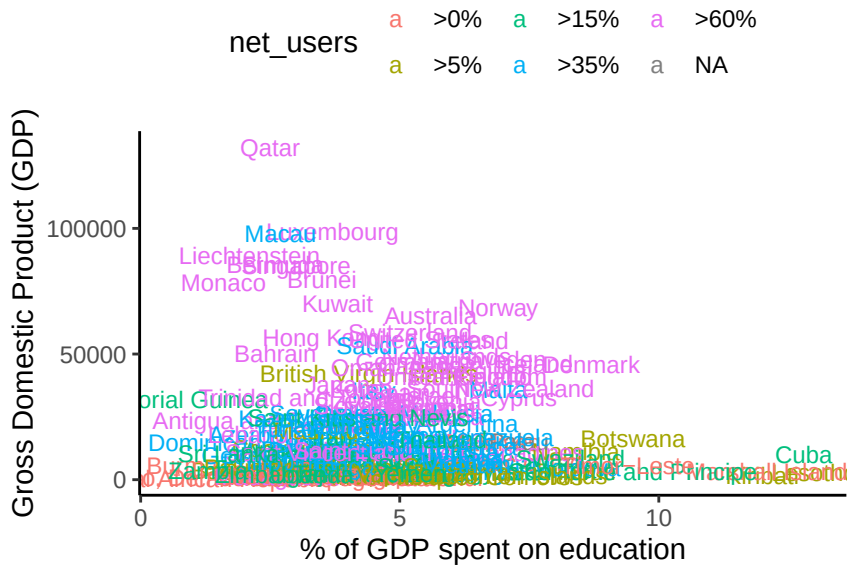
1.2 Now, update the plot with more informative labels. Label the x-axis "% of GDP spent on education" and the y-axis "Gross Domestic Product (GDP)". *Hint: see Section 3.2.1 for an example of one way to label the axes.*

```
data(CIACountries)

# define the plot object
g <- ggplot(data = CIACountries, aes(x=educ,y=gdp))
g = g + geom_text(aes(label = country, color = net_users), size = 3) +
  labs(x = "% of GDP spent on education", y = "Gross Domestic Product (GDP)")
# print the plot
g
```



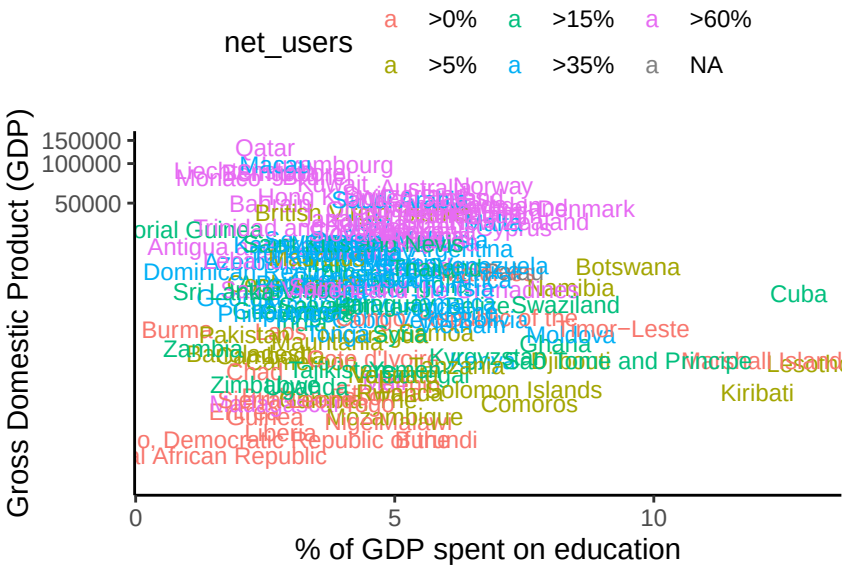
1.3 Next, move the legend so that it's located on the top of the plot as opposed to the right of the plot. *Hint: see Section 3.1.4 for an example on how to change the legend position.*

```
data(CIACountries)

# define the plot object
g <- ggplot(data = CIACountries, aes(x=educ,y=gdp))
g = g + geom_text(aes(label = country, color = net_users), size = 3) +
  labs(x = "% of GDP spent on education", y = "Gross Domestic Product (GDP)") +
  theme(legend.position = "top")
# print the plot
g
```



1.4 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.

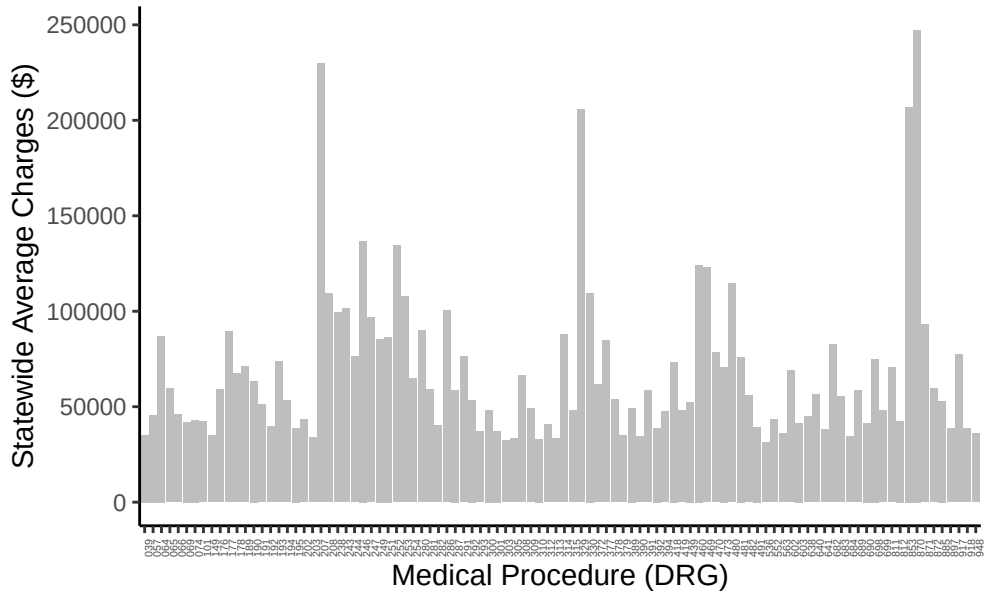
```
data(CIACountries)

# define the plot object
g <- ggplot(data = CIACountries, aes(x=educ,y=gdp))
g = g + geom_text(aes(label = country, color = net_users), size = 3) +
  coord_trans(y = "log10") +
  labs(x = "% of GDP spent on education", y = "Gross Domestic Product (GDP)") +
  theme(legend.position = "top")
# print the plot
g
```


net_users
a >0%
a >15%
a >60%
a >5%
a >35%
a NA
Gross Domestic Product (GDP)
150000
100000
50000
0
5
10
% of GDP spent on education
Qatar
Macau
Norway
Denmark
Botswana
Cuba
Namibia
Swaziland
Timor−Leste
Lesotho
Kiribati
Comoros
Solomon Islands
Mozambique
Malawi
Liberia
Burundi

Problem 2 **Medical procedures**

2.1 Consider Figure 3.8 in Section 3.1.5. What does `reorder(drg, mean_charge)` do? To figure this out, recreate the plot, but use `x = drg` instead of `x = reorder(drg, mean_charge)`. What happens?

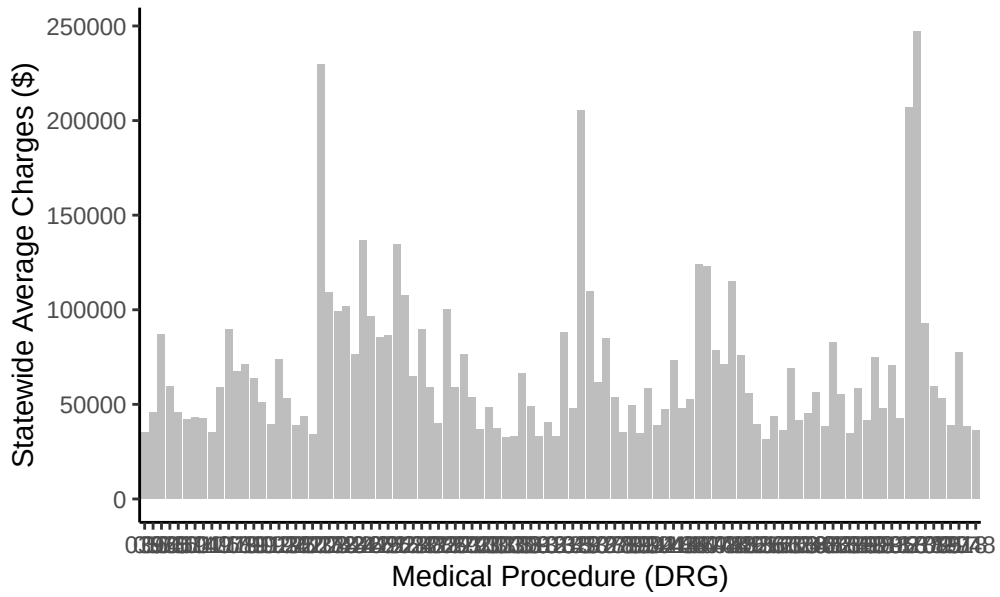
```
data(MedicareCharges)

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
```



"Reorder" rearranges the different drugs by the mean charge of it. When leaving it as x = drg, the graph is no longer sorted in terms of numbers.

2.2 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.*

```
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
```



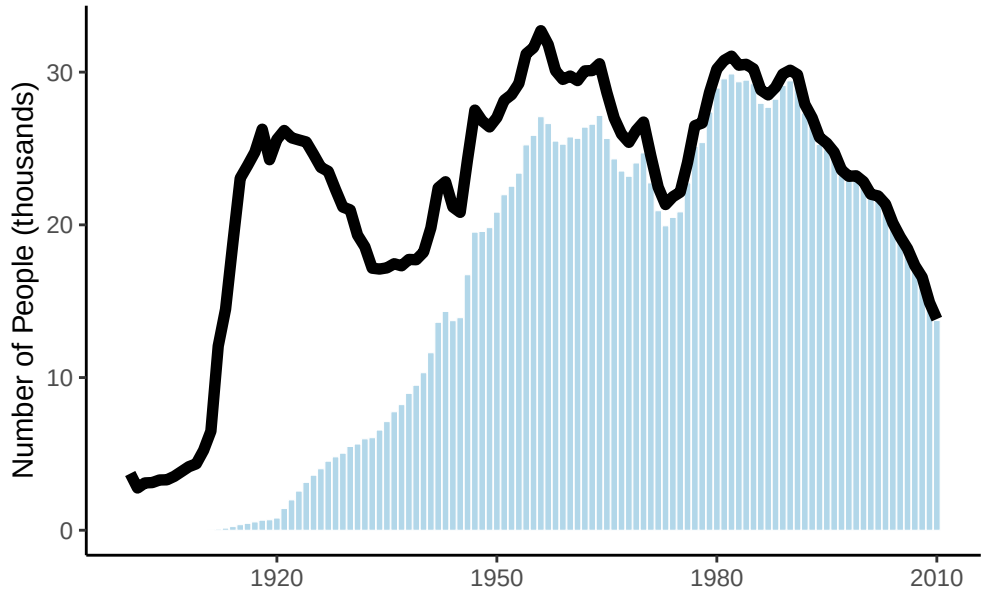
Without the theme function the x axis of the graph become unreadable with how tight the ticks become. The purpose of theme() helps delegate the visual representation of the legend and of the graph itself, adjusting the size of the x axis.

Problem 3 **Historical baby names** As you read through (or, even better, as you code along with. . .) 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.

1.What does the xlab(NULL) do for the line of people born that year?

2.How does the tribble formatting and visual theme functions work in order to make the context text make more sense?

```
library(babynames)
BabynamesDist <- make_babynames_dist()

joseph <- BabynamesDist %>%
  filter(name == "Joseph" & sex == "M")
name_plot <- ggplot(data = joseph, aes(x = year)) +
    geom_col(
       aes(y = count_thousands * alive_prob),
       fill = "#b2d7e9",
       color = "white",
       size = 0.1) +
    geom_line(aes(y = count_thousands), size = 2) +
    ylab("Number of People (thousands)") +
    xlab(NULL)
name_plot
```



```
BabynamesDist <- make_babynames_dist()

joseph <- BabynamesDist %>%
  filter(name == "Joseph" & sex == "M")
name_plot <- ggplot(data = joseph, aes(x = year))

name_plot <- name_plot +
  geom_col(
    aes(y = count_thousands * alive_prob),
    fill = "#b2d7e9",
    color = "white",
    size = 0.1
  )

name_plot
```