Assignments

I want you to submit your assignment as a PDF, so I can keep a record of what the code looked like that day. I also want you to include your answers on your personal GitHub website. This will be good practice for editing your website and it will help you produce something you can keep after the class is over.

1. Download the Assignment1.Rmd file from Canvas. You can use this as a template for writing your answers. It's the same as what you can see on my website in the Assignments tab. Once we're done with this I'll edit the text on the website to include the solutions.
2. On RStudio, open a new R script in RStudio (File > New File > R Script). This is where you can test out your R code. You'll write your R commands and draw plots here.
3. Once you have finalized your code, copy and paste your results into this template (Assignment 1.Rmd). For example, if you produced a plot as the solution to one of the problems, you can copy and paste the R code in R markdown by using the ``{r} ``` command. Answer the questions in full sentences and Save.
4. Produce a PDF file with your answers. To do this, knit to PDF (use Knit button at the top of RStudio), locate the PDF file in your docs folder (it's in the same folder as the Rproj), and submit that on on Canvas in Assignment 1.
5. Build Website, go to GitHub desktop, commit and push. Now your solutions should be on your website as well.

Assignment 1

This assignment is due on Canvas on Monday 9/20 before class, at 10:15 am. Include the name of anyone with whom you collaborated at the top of the assignment.

Problem 1

Install the datasets package on the console below using `install.packages("datasets")`. Now load the library.

```
library(datasets)
```

Load the USArrests dataset and rename it `dat`. Note that this dataset comes with R, in the package datasets, so there's no need to load data from your computer. Why is it useful to rename the dataset?

Answer: Well, we want to replicate analyses. That's why it's nice to rename data.

```
dat <- USArrests
```

Problem 2

Use this command to make the state names into a new variable called State.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
dat$state <- tolower(rownames(USArrests))
```

I used the dplyr package in order to rename this variable. I am sure there are other ways to do this but I used old reliable!

This dataset has the state names as row names, so we just want to make them into a new variable. We also make them all lower case, because that will help us draw a map later - the map function requires the states to be lower case.

List the variables contained in the dataset `USArrests`.

```
names(dat)
```

```
## [1] "Murder"   "Assault"  "UrbanPop" "Rape"     "state"
```

Answer: The four variables are Murder, Assault, UrbanPop, and Rape.

Problem 3

What type of variable (from the DVB chapter) is `Murder`? **Answer**: quantitative variable

What R Type of variable is it?

```
typeof(dat$Murder)
```

```
## [1] "double"
```

Answer: double

Problem 4

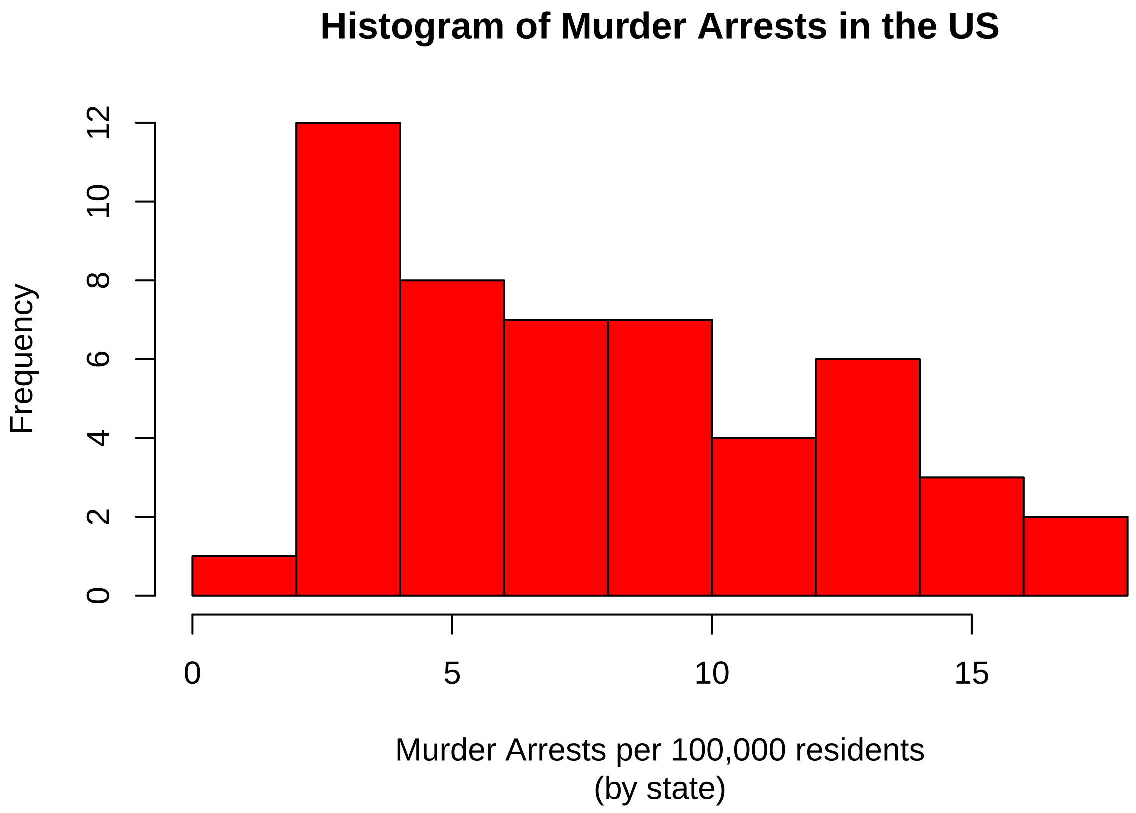
What information is contained in this dataset, in general? What do the numbers mean?

Answer: The information in this dataset is a set of the number of arrests for different violent crimes in the United States, as divided by state. The numbers represent the numeric value of arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states, also providing the percent of the population living in urban areas (UrbanPop).

Problem 5

Draw a histogram of ***`Murder`*** *with proper labels and title.*

```
hist(dat$Murder, col = 'red', ylab = 'Frequency', xlab = 'Murder Arrests per 100,000 residents', main =
```



Problem 6

Please summarize ***`Murder`*** *quantitatively. What are its mean and median? What is the difference between mean and median? What is a quartile, and why do you think R gives you the 1st Qu. and 3rd Qu.?*

```
summary(dat$Murder)
```

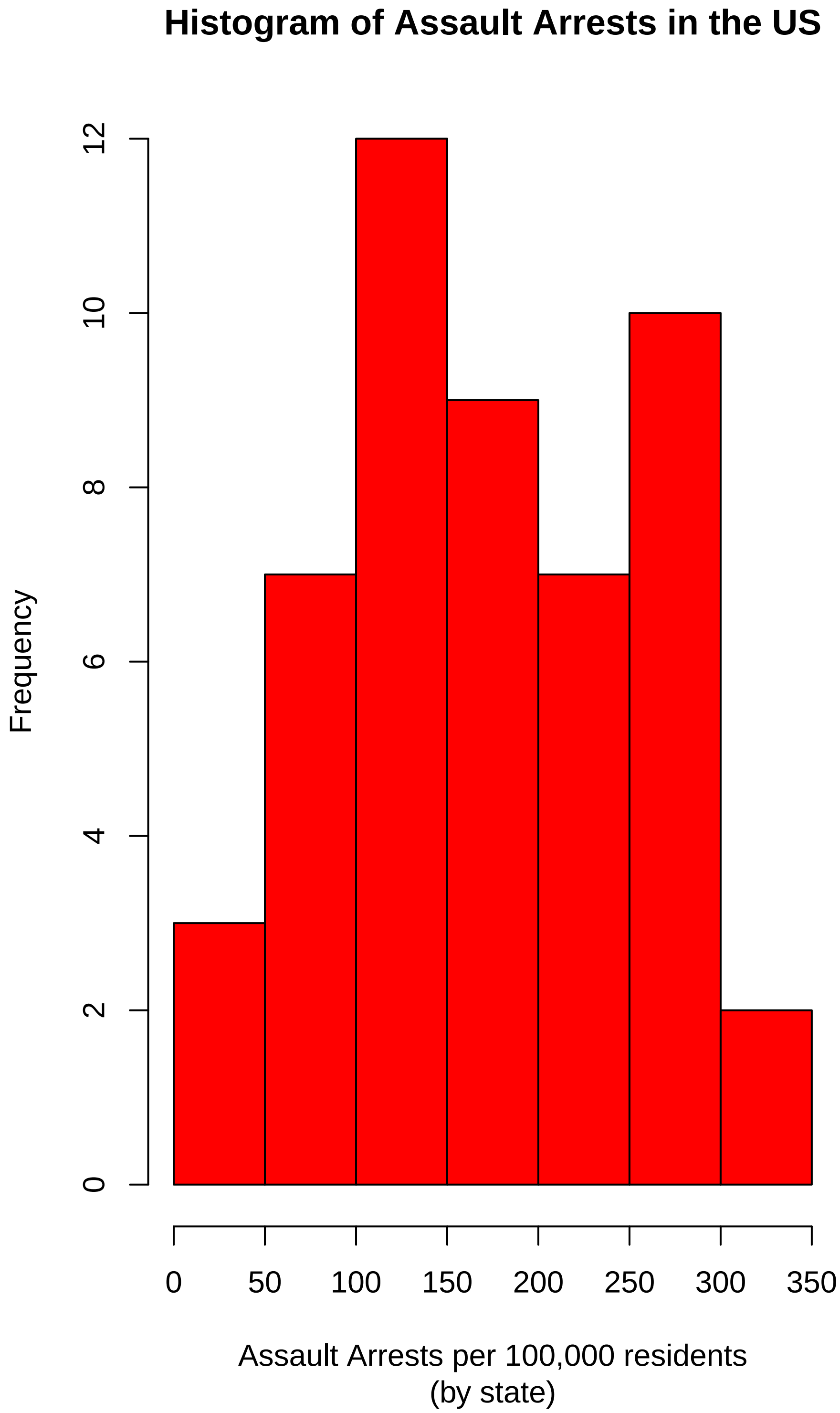
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.800   4.075   7.250   7.788  11.250  17.400
```

Answer: The mean of the Murder variable is 7.788 (murder arrests per 100,000 residents of a state). The median is 7.250 (murder arrests per 100,000 residents of a state). The difference between these two values quantitatively is 0.538. The difference qualitatively is that the mean represents the average number of murder arrests per 100,000 residents of all US states while the median is the middle value of the data set, showing the data has a slight right skew. A quartile isdivides the dataset into four parts, with Q1 being the median of the lower half of the dataset and Q3 being the median of the upper half of the data set. R gives you this information because quartiles can be helpful in determining where a specific data point falls in the set; for example, if a value for "Murders" is greater than Q3, 11.250, we can more reasonably assume that this state has an exceptionally high murder rate in comparison to other states in the US. If a state has a value for "Murders" that is less than 4.075, we can more reasonably assume that this state is relatively safer than other US states!

Problem 7

Repeat the same steps you followed for `Murder`*, for the variables* `Assault` *and* `Rape`*. Now plot all three histograms together. You can do this by using the command* `par(mfrow=c(3,1))` *and then plotting each of the three.*

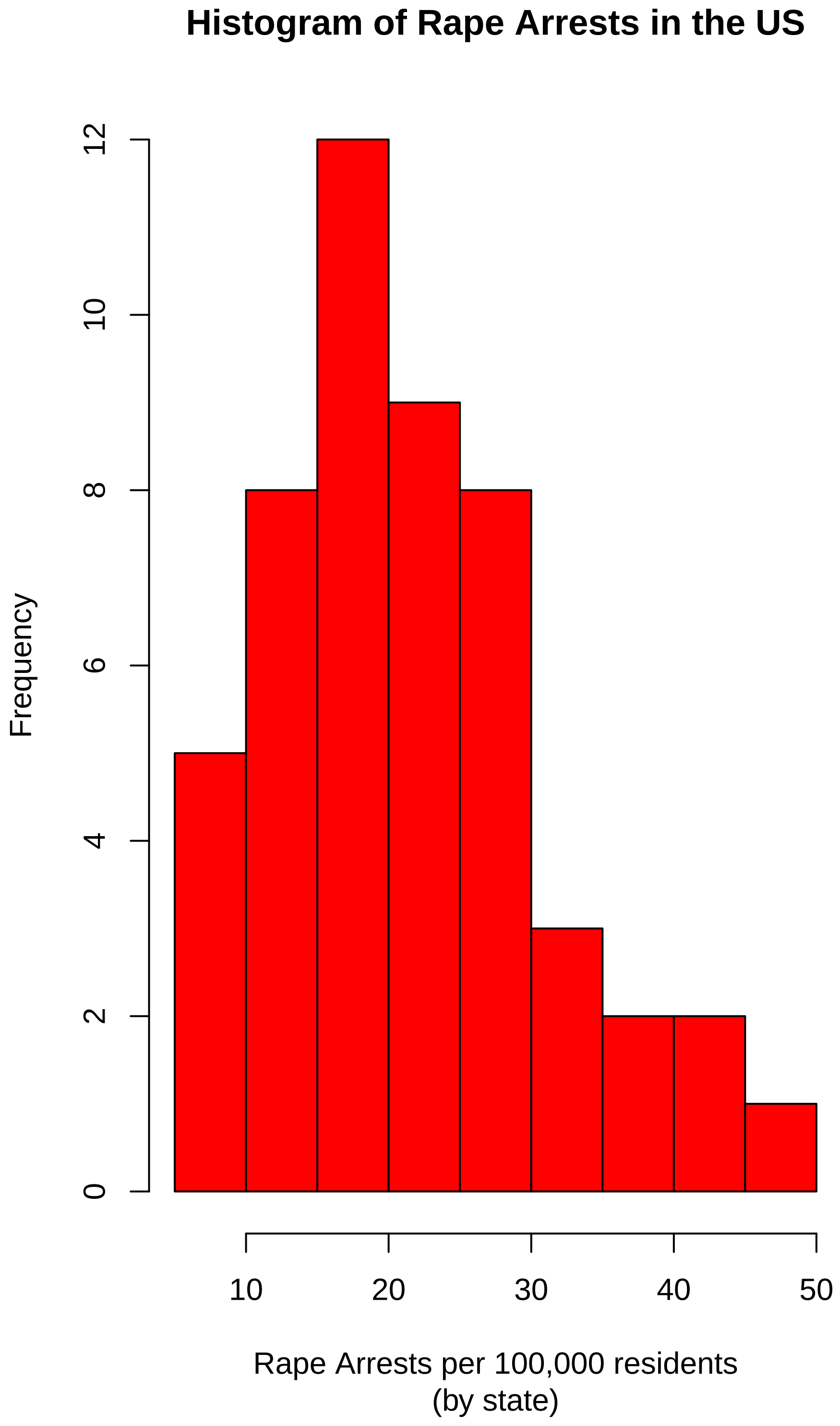
```
hist(dat$Assault, col = 'red', ylab = 'Frequency', xlab = 'Assault Arrests per 100,000 residents', main
```



```r
summary(dat$Assault)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    45.0   109.0   159.0   170.8   249.0   337.0
```

```r
hist(dat$Rape, col = 'red', ylab = 'Frequency', xlab = 'Rape Arrests per 100,000 residents', main = 'His
```



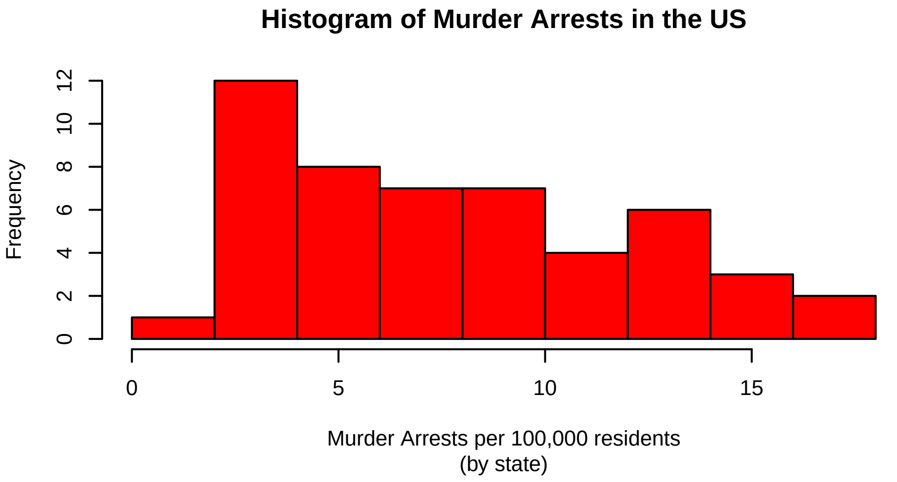
```
summary(dat$Rape)
```

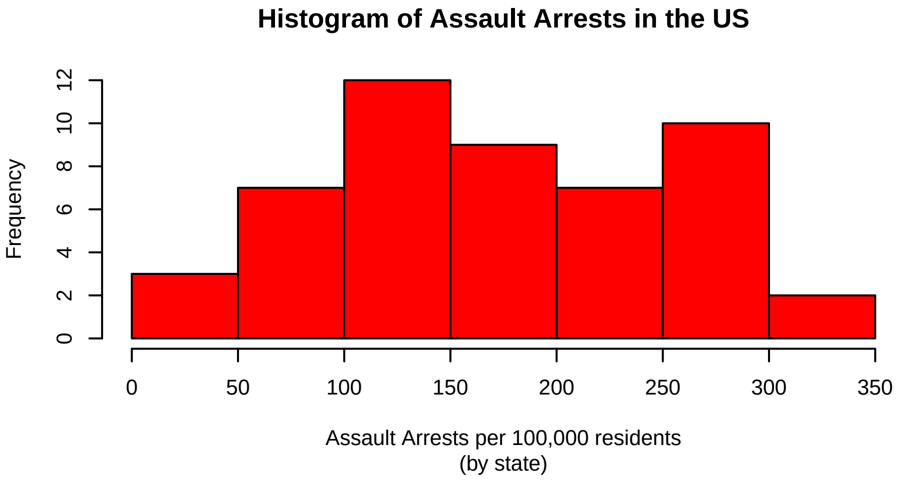
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.30   15.07   20.10   21.23   26.18   46.00
```

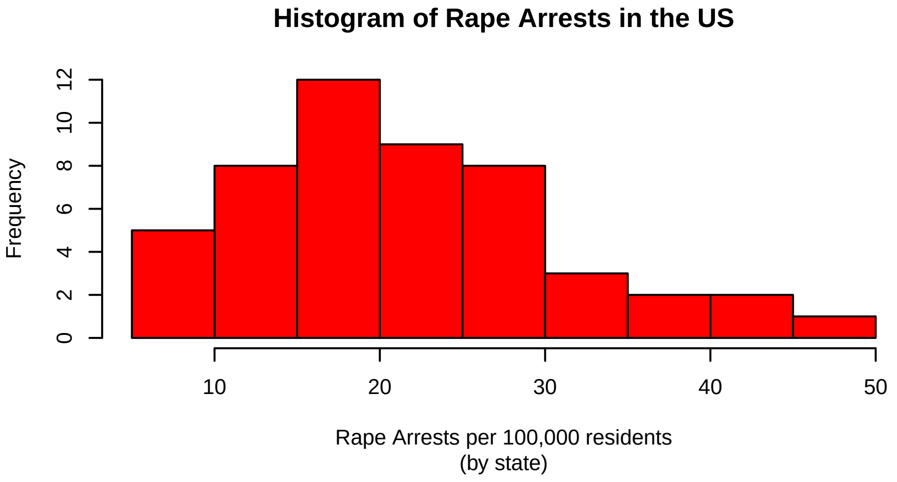
```
par(mfrow=c(3,1))
hist(dat$Murder, col = 'red', ylab = 'Frequency', xlab = 'Murder Arrests per 100,000 residents', main =
hist(dat$Assault, col = 'red', ylab = 'Frequency', xlab = 'Assault Arrests per 100,000 residents', main
summary(dat$Assault)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    45.0   109.0   159.0   170.8   249.0   337.0
```

```
hist(dat$Rape, col = 'red', ylab = 'Frequency', xlab = 'Rape Arrests per 100,000 residents', main = 'Hi
```







What does the command par do, in your own words (you can look this up by asking R `?par`)?

Answer: The par function allows an R user to look at the graphical parameters that control how graphs are displayed. In the case of the par(mfrow=c(3,1)) function, we are able to not only look at the graphical parameters, but control them so that we can decide how many subplots we want displayed.

What can you learn from plotting the histograms together?

Answer: From plotting the histograms together, we can view a comparison of the distributions in relation to each other, examining skew and quartile arrangement to see how distributions vary by crime type. On a more general note, plotting the histograms together allows us to see generally how frequency and distribution relationships among multiple variables interact, allowing a viewer a more clear image of the relationship between variables while still being able to view a more individualistic look at each histogram.

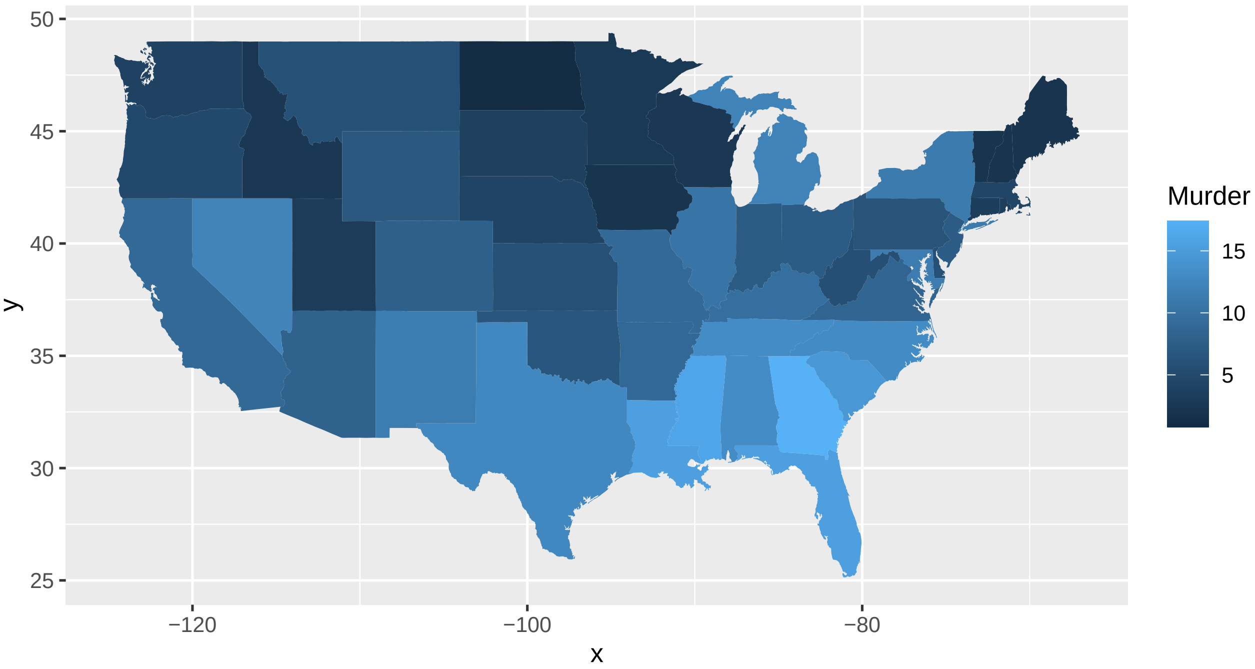
Problem 8

In the console below (not in text), type `install.packages("maps")` *and press Enter, and then type* `install.packages("ggplot2")` *and press Enter. This will install the packages so you can load the libraries.*

Run this code:

```
library(maps)
library(ggplot2)

ggplot(dat, aes(map_id=state, fill=Murder)) +
  geom_map(map=map_data("state")) +
  expand_limits(x=map_data("state")$long, y=map_data("state")$lat)
```



What does this code do? Explain what each line is doing.

Answer: ggplot is the most basic implementation of any data visualization in R. In this instance, the first line of code is filling out the basic information of the data visual by specifying the variables which are involved in the plot. The second line of code employs the maps package and ggplot2 package to choose specifically to create a mapping of the selected variables, using states as the selected implementation of the map. The third line employs the expand_limit function in order to ensure that the limits of the visualization include a single value for all aspects of the plot, implying what should be included in the scale.