Homework 2
Brian Detweiler
STAT 4410/8416 Section 001/002
Fall 2016
Due: September 30, 2016 by midnight

1. We generate a nxk matrix M and a vector V of length k for some specific values of n and k as follows;

```r
set.seed(4286)
n <- 4
k <- 5
V <- sample(seq(4), size = k, replace = TRUE)
M <- matrix(rnorm(n * k), ncol = k)
```

   (a) Now, carefully review the following for loop. Rewrite the code that does the same job but doesn't use a for loop.

```r
X <- M
for(i in seq(n)){
  X[i,] <- round(M[i, ] / V, 2)
}
```

   Answer: We can use `apply` to apply a function to a particular dimension of a matrix, in this case, the rows. We must also transpose the matrix that `apply` returns.

```r
Y <- t(apply(M, 1, function(x) { round(x / V, 2) }))
```

   (b) Now do the same experiment for $n = 400$ and $k = 500$. Which code runs faster, your code or the for loop? Demonstrate that using function `system.time()`.

   Answer:

```r
n <- 400
k <- 500
V <- sample(seq(4), size = k, replace = TRUE)
M <- matrix(rnorm(n * k), ncol = k)

X <- M
x.time <- system.time(
  for(i in seq(n)){
    X[i,] <- round(M[i, ] / V, 2)
})

y.time <- system.time(t(apply(M, 1, function(x) { round(x / V, 2) })))
```

   The for-loop finished in 0.03 @ milliseconds. The `apply` function finished in 0.04 milliseconds.

2. The data set tips contains tip for different party size as well as total bill personal information about bill payer. We can get the data from reshape2 packages as follows;

```
library(reshape2)
attach(tips)
tips.dat <- tips
```

Now answer the following questions.

(a) Compute tip rate dividing tip by total bill and create a new column called tip.rate in the dataframe tips.dat. Demosntrate your result by showing the head of tips.dat.
Answer:

```
tips.dat$tip.rate <- tips.dat$tip / tips.dat$total_bill
head(tips.dat)
```

```
##   total_bill  tip    sex smoker day   time size   tip.rate
## 1      16.99 1.01 Female     No Sun Dinner    2 0.05944673
## 2      10.34 1.66   Male     No Sun Dinner    3 0.16054159
## 3      21.01 3.50   Male     No Sun Dinner    3 0.16658734
## 4      23.68 3.31   Male     No Sun Dinner    2 0.13978041
## 5      24.59 3.61 Female     No Sun Dinner    4 0.14680765
## 6      25.29 4.71   Male     No Sun Dinner    4 0.18623962
```

(b) Draw a side by side violin plot of tip rate for each party size. Order the party size based on the median tip rate. Provide your codes and the plot. Which party size is responsible for highest median tip rate?
Answer:
Here, we order by the median tip rate. This can probably be done with reshape2, but I couldn't figure out how to melt and cast the data correctly.
We can see that the single party is responsible for the highest median tip.

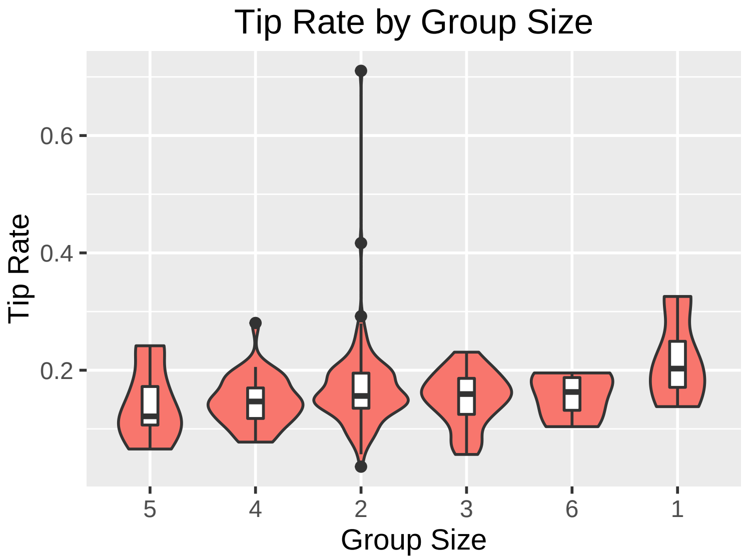
```
library(ggplot2)

tips.dat$median <- c(rep(0, length(tips.dat[, 1])))

for (i in 1:6) {
  tips.dat[tips.dat$size == i, ]$median <-
    median(tips.dat[tips.dat$size == i, ]$tip.rate)
}

tips.dat$size <- factor(tips.dat$size,
                        levels = tips.dat$size[order(tips.dat$median)])

ggplot(tips.dat, aes(x = factor(size), y = tip.rate)) +
  geom_violin(aes(fill = 'grey')) +
  geom_boxplot(width = 0.2) +
  labs(title = 'Tip Rate by Group Size', x = 'Group Size', y = 'Tip Rate') +
  guides(fill = FALSE)
```



(c) Generate the similar plot you did in question 2b for each day (instead of party size) and facet by sex and smoker. Is the shape of violin plot similar for each faceted condition?

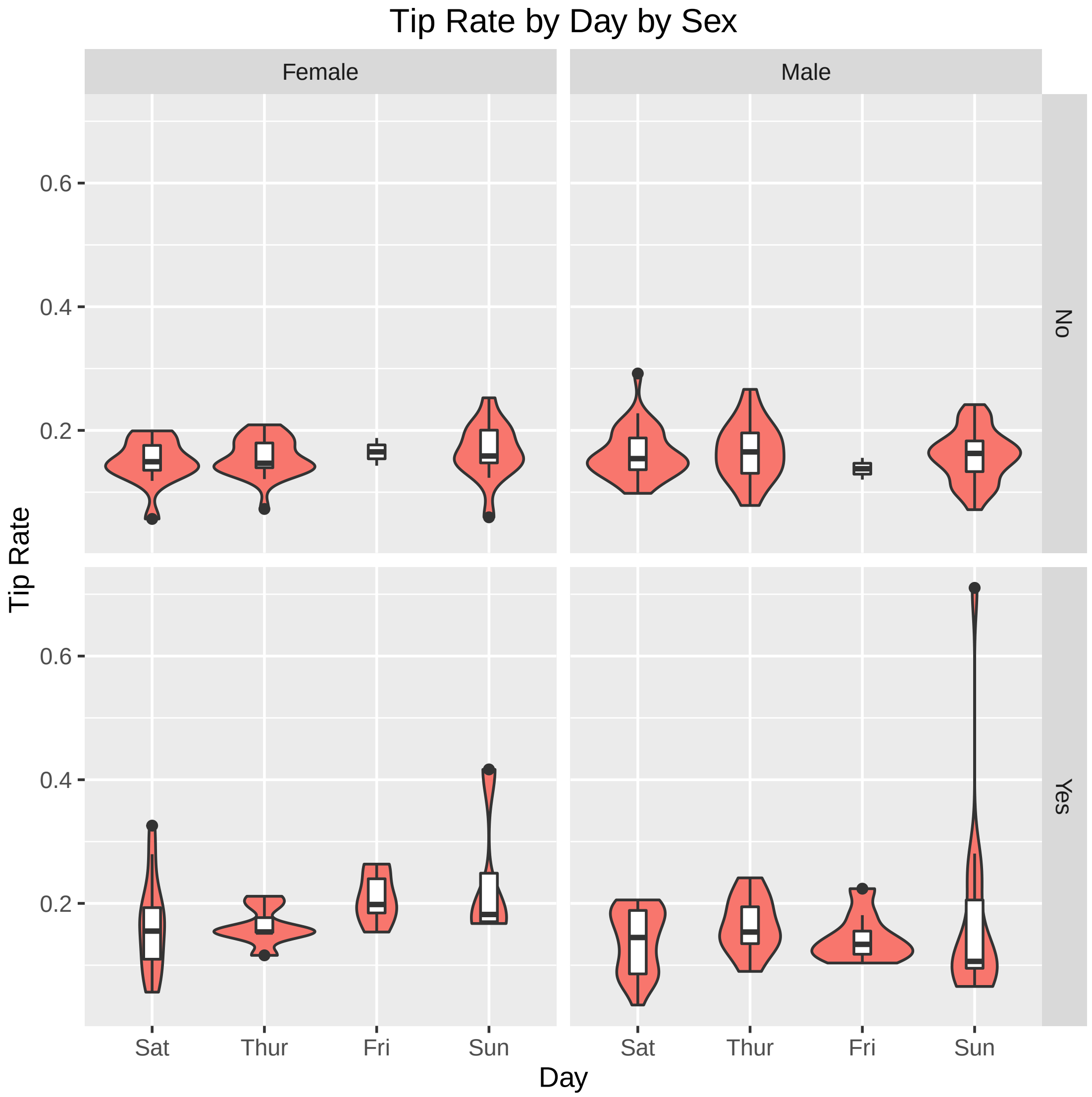
Answer:

```
tips.dat <- tips
tips.dat$tip.rate <- tips.dat$tip / tips.dat$total_bill
tips.dat$median <- c(rep(0, length(tips.dat[, 1])))

for (i in tips.dat$day) {
  tips.dat[tips.dat$day == i, ]$median <-
    median(tips.dat[tips.dat$day == i, ]$tip.rate)
}

tips.dat$day <- factor(tips.dat$day,
                        levels = tips.dat$day[order(tips.dat$median)])

ggplot(tips.dat, aes(x = factor(day), y = tip.rate)) +
  geom_violin(aes(fill = 'grey')) +
  geom_boxplot(width = 0.2) +
  facet_grid(smoker ~ sex) +
  labs(title = 'Tip Rate by Day by Sex', x = 'Day', y = 'Tip Rate') +
  guides(fill = FALSE)
```



The shapes are similar for non-smokers, but are quite different between female and male smokers.

3. We want to generate a plot of US arrest data (USArrests). Please provide the detailed codes to answer the following questions.

   (a) Obtain USA state boundary coordinates data for USA map using function `map_data()` and store the data in `mdat`. Display first few data from `mdat` and notice that there is a column called `order` that contains the true order of coordinates.

   Answer:

```
mdat <- map_data('state')
head(mdat)

##        long      lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
## 2 -87.48493 30.37249     1     2 alabama      <NA>
## 3 -87.52503 30.37249     1     3 alabama      <NA>
## 4 -87.53076 30.33239     1     4 alabama      <NA>
## 5 -87.57087 30.32665     1     5 alabama      <NA>
## 6 -87.58806 30.32665     1     6 alabama      <NA>
```

   (b) You will find USA crime data in the data frame called `USArrests`. Standardize the crime rates and create a new column called `state` so that all the state names are lower case. Store the new data in `arrest` and report first few data.

   Answer:

```
state <- tolower(rownames(USArrests))
scaled.arrests <- scale(USArrests)
pop <- as.data.frame(scaled.arrests[,'UrbanPop'])
murder <- as.data.frame(scaled.arrests[,'Murder'])
assault <- as.data.frame(scaled.arrests[,'Assault'])
rape <- as.data.frame(scaled.arrests[,'Rape'], )
arrest <- data.frame(pop, murder, assault, rape, state)
colnames(arrest) <- c('UrbanPop', 'Murder', 'Assault', 'Rape', 'state')
head(arrest)

##              UrbanPop     Murder   Assault         Rape      state
## Alabama    -0.5209066 1.24256408 0.7828393 -0.003416473    alabama
## Alaska     -1.2117642 0.50786248 1.1068225  2.484202941     alaska
## Arizona     0.9989801 0.07163341 1.4788032  1.042878388    arizona
## Arkansas   -1.0735927 0.23234938 0.2308680 -0.184916602   arkansas
## California  1.7589234 0.27826823 1.2628144  2.067820292 california
## Colorado    0.8608085 0.02571456 0.3988593  1.864967207   colorado
```

   (c) Merge the two data sets `mdat` and `arrest` by state name. Merging will change the order of coordinates data. So, order the data back to the original order and store the merged-ordered data in `odat`. Report first few data from `odat`.

   Answer:

```
odat <- merge(mdat, arrest, by.x = 'region', by.y = 'state')

# Order the data
odat <- odat[order(odat$order), ]
head(odat)

##     region      long       lat group order subregion   UrbanPop    Murder
## 1 alabama -87.46201 30.38968     1     1      <NA> -0.5209066 1.242564
## 2 alabama -87.48493 30.37249     1     2      <NA> -0.5209066 1.242564
## 6 alabama -87.52503 30.37249     1     3      <NA> -0.5209066 1.242564
## 7 alabama -87.53076 30.33239     1     4      <NA> -0.5209066 1.242564
## 8 alabama -87.57087 30.32665     1     5      <NA> -0.5209066 1.242564
## 9 alabama -87.58806 30.32665     1     6      <NA> -0.5209066 1.242564
##     Assault         Rape
## 1 0.7828393 -0.003416473
## 2 0.7828393 -0.003416473
## 6 0.7828393 -0.003416473
## 7 0.7828393 -0.003416473
## 8 0.7828393 -0.003416473
## 9 0.7828393 -0.003416473
```

(d) All the columns of `odat` is not necessary for our analysis. So, subset by selecting only columns long, lat, group, region, Murder, Assault, UrbanPop, Rape. Store the data in `sdat` and report first few rows.

Answer:

```
sdat <- subset(odat, select = c('long',
                                'lat',
                                'group',
                                'region',
                                'Murder',
                                'Assault',
                                'UrbanPop',
                                'Rape'))
head(sdat)
```

```
##        long      lat group  region   Murder   Assault   UrbanPop
## 1 -87.46201 30.38968     1 alabama 1.242564 0.7828393 -0.5209066
## 2 -87.48493 30.37249     1 alabama 1.242564 0.7828393 -0.5209066
## 6 -87.52503 30.37249     1 alabama 1.242564 0.7828393 -0.5209066
## 7 -87.53076 30.33239     1 alabama 1.242564 0.7828393 -0.5209066
## 8 -87.57087 30.32665     1 alabama 1.242564 0.7828393 -0.5209066
## 9 -87.58806 30.32665     1 alabama 1.242564 0.7828393 -0.5209066
##           Rape
## 1 -0.003416473
## 2 -0.003416473
## 6 -0.003416473
## 7 -0.003416473
## 8 -0.003416473
## 9 -0.003416473
```

(e) Melt the data frame `sdat` with id variables long, lat, group, region. Store the molten data in `msdat` and report first few rows of data.

Answer:

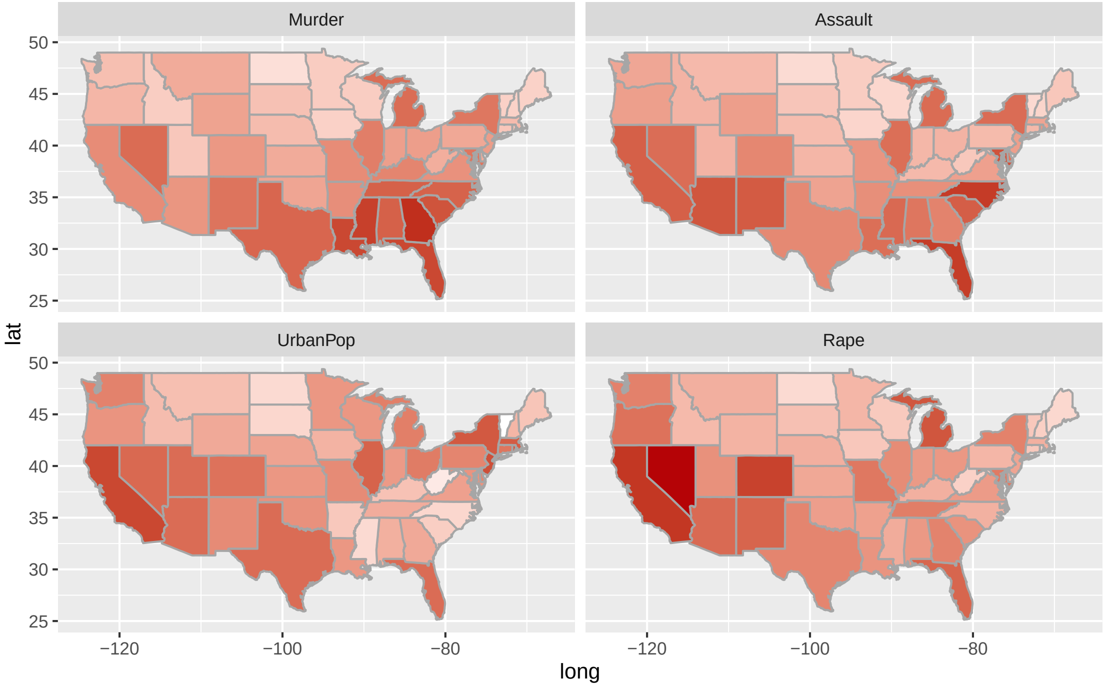
```
msdat <- melt(sdat, id.vars = c('long', 'lat', 'group', 'region'))
head(msdat)
```

```
##        long      lat group  region variable    value
## 1 -87.46201 30.38968     1 alabama   Murder 1.242564
## 2 -87.48493 30.37249     1 alabama   Murder 1.242564
## 3 -87.52503 30.37249     1 alabama   Murder 1.242564
## 4 -87.53076 30.33239     1 alabama   Murder 1.242564
## 5 -87.57087 30.32665     1 alabama   Murder 1.242564
## 6 -87.58806 30.32665     1 alabama   Murder 1.242564
```

(f) The molten data frame `msdat` is now ready to be plotted. Create a plot showing USA state map, fill with value and facet_wrap with variable. Please don't add any legend and make sure that faceting labels are identified so that we can compare the facetted plots.

Answer:

```
ggplot(msdat, aes(x = long,
                  y = lat)) +
  geom_polygon(aes(group = group,
                   fill = value),
               color = "grey65") +
  scale_fill_gradient(low = "#ffffff",
                      high = "#b50306",
                      space = "Lab",
                      na.value = "grey50",
                      guide = FALSE) +
  facet_wrap(~variable)
```



(g) Now examine the plot you have generated in question (3f) and answer the following questions based on what you see in the plot.

i. For each of the crimes, name two states with the highest crime rate.

Answer:
For murder, Georgia is the highest followed by a few other southern states, such as Mississippi. In assault, Florida and North Carolina clock in as the highest. And for rape, Nevada stands out as bright red, followed by California.

ii. Do you think larger urban population is an indicative of larger murder rate? Why or why not?

Answer:

Definitely not. If this were the case, California, New York, Texas, Illinois, and Florida would stand out on the murder plots, but instead, southern states, which are some of the lightest in terms of urban population, dominate in the murder category.

(h) In question (3b) we standardized the crime rates. Why do you think we did this? Explain what would happen if we would not do this.

Answer:

By standardizing the crime rates, we put everything on the same scale. Measuring things against each other is meaningless if they are on different scales.

(i) In question (3c) we ordered the data after merging. Why do you think we have to order? Explain what would happen if we would not order.

Answer:

The order of the data is what determines how the shape files are drawn. It's like playing ”connect the dots”. If we didn't have an order in which to connect them, our polygon would likely not look like a state.

4. For the following questions please use data frame `tips`

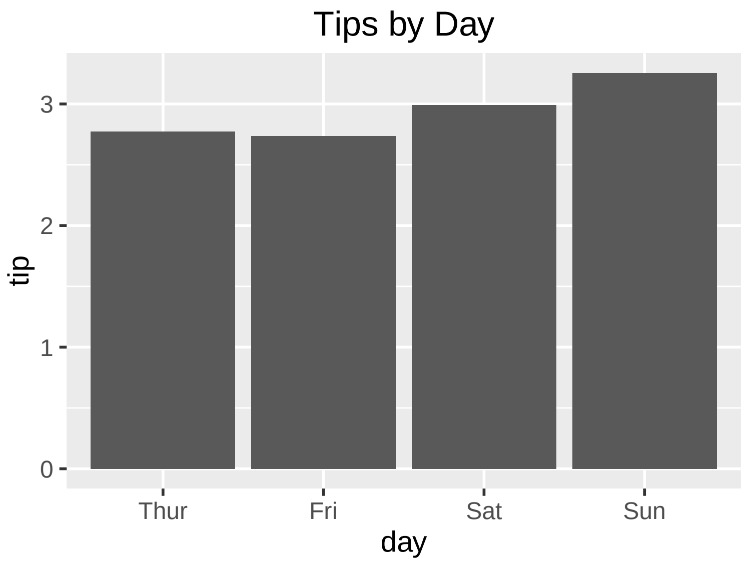
(a) Create a bar chart that shows average tip by day.

Answer:

```r
tips.ordered <- tips

tips.ordered$day <- factor(tips$day, levels = c('Thur', 'Fri', 'Sat', 'Sun'))

ggplot(tips.ordered) +
  geom_bar(aes(day, tip),
           position = "dodge",
           stat = "summary",
           fun.y = "mean") +
  labs(title = 'Tips by Day')
```



(b) Compute the average tip, total tip and average size grouped by smoker and day. i.e., For each combination of smoker and day you should have a row of these summaries. Report the result in a nice table.

Answer:

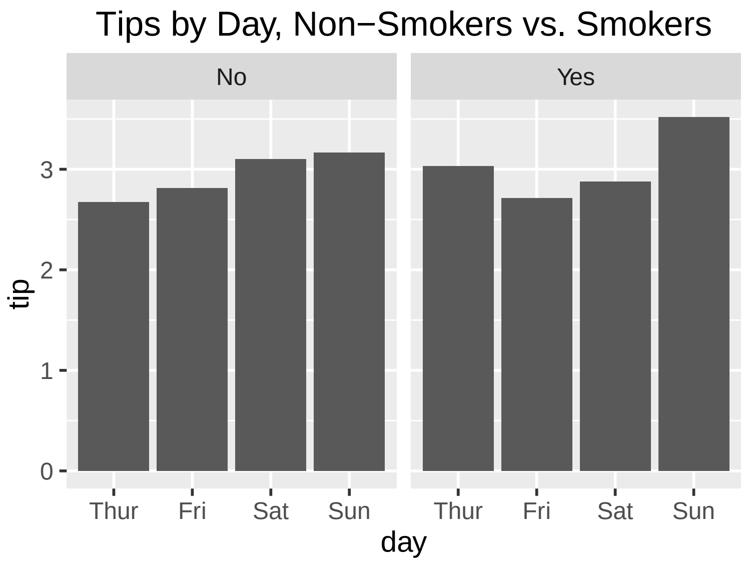
	Average Tip	Total Tip	Average Group Size
NonSmoker	2.991854	451.77	2.668874
Smoker	3.008710	279.81	2.408602
Thu	2.771452	171.83	2.451613
Fri	2.734737	51.96	2.105263
Sat	2.993103	260.40	2.517241
Sun	3.255132	247.39	2.842105

(c) Create a bar chart that shows average tip by day and also faceted by smoker.

Answer:

```r
tips.ordered <- tips
tips.ordered$day <- factor(tips$day, levels = c('Thur', 'Fri', 'Sat', 'Sun'))

ggplot(tips.ordered) +
  geom_bar(aes(day, tip),
           position = "dodge",
           stat = "summary",
           fun.y = "mean") +
  facet_grid(. ~ smoker) +
  labs(title = 'Tips by Day, Non-Smokers vs. Smokers')
```

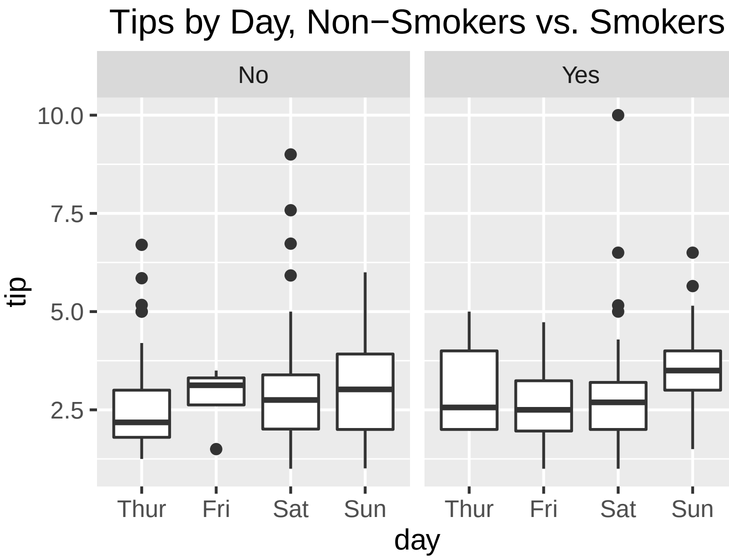


(d) In questions 4a and 4c we plotted the summary of data which does not show us the whole picture. In practice we like to see the whole data. What plot do you suggest to serve the same purpose similar to what we did in question 4c? In other words, what would be a better plot to show tips by day and facetted by smoker? Please produce that plot and include your codes.

Answer:

A box-and-whisker plot gives us much more information than a bar plot. In this one plot, we get the upper and lower quantiles, the median, the range, and even outliers.

```r
ggplot(tips.ordered) +
  geom_boxplot(aes(day, tip)) +
  facet_grid(. ~ smoker) +
  labs(title = 'Tips by Day, Non-Smokers vs. Smokers')
```


Tips by Day, Non−Smokers vs. Smokers
No
Yes
10.0
7.5
5.0
2.5
tip
Thur
Fri
Sat
Sun
Thur
Fri
Sat
Sun
day

5. Life expectancy data for four countries are obtained from the world bank database which you will find on blackboard. It contains life expectancy in years for different genders. Download the data from the blackboard and save it on your hard drive. Now answer the following questions.

   (a) Read the file from your hard drive and display first few rows of the data.
   Answer:

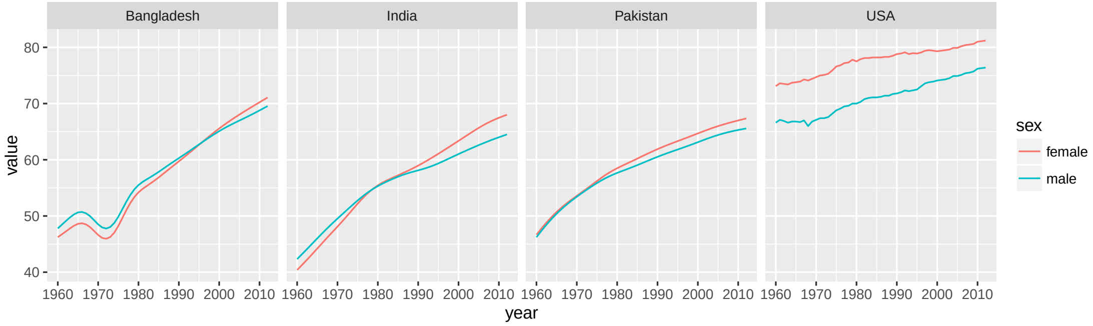
```
life.expectancy <- read.csv(file='life-expectancy.csv', header=TRUE)
head(life.expectancy)

##   year    sex Bangladesh  India Pakistan  USA
## 1 1960 female     46.224 40.391   46.655 73.1
## 2 1960   male     47.787 42.329   46.223 66.6
## 3 1961 female     46.731 41.125   47.564 73.6
## 4 1961   male     48.445 43.052   47.156 67.1
## 5 1962 female     47.254 41.876   48.426 73.5
## 6 1962   male     49.104 43.784   48.044 66.9
```

   (b) Generate a plot showing trend line of life expectancy over different year. Color them by sex and facet by country. Include your code and the plot.
   Answer:

```
life.expect.melt <- melt(life.expectancy, id.vars=c('year', 'sex'))

ggplot(life.expect.melt) +
  geom_line(aes(year, value, colour = sex)) +
  facet_grid(. ˜ variable)
```



   (c) Explain what interesting features you notice in the plot of question 5b.
   Answer:
   Perhaps the most surprising feature is how much higher life expectancy is in the US. People in the US can expect to live nearly ten years longer than in Bangladesh, India, or Pakistan. Also interesting is that women in the US can expect to live between five and ten years longer than men. Lastly, a positive note, life expectancy has increased across the board over the years.

6. **Ordering the factor** In class, we have seen how to order the factors. Suppose we have the following data about different class of students;

```r
class <- c("freshman", "graduate", "junior", "senior", "sophomore")
rate <- c(56, 35, 56, 48, 60)
df <- data.frame(class, rate)
```

Now please answer the following questions.

(a) Convert the class column of dataframe 'df' into a factor column. Demonstrate that it is indeed converted into a factor column.

Answer:

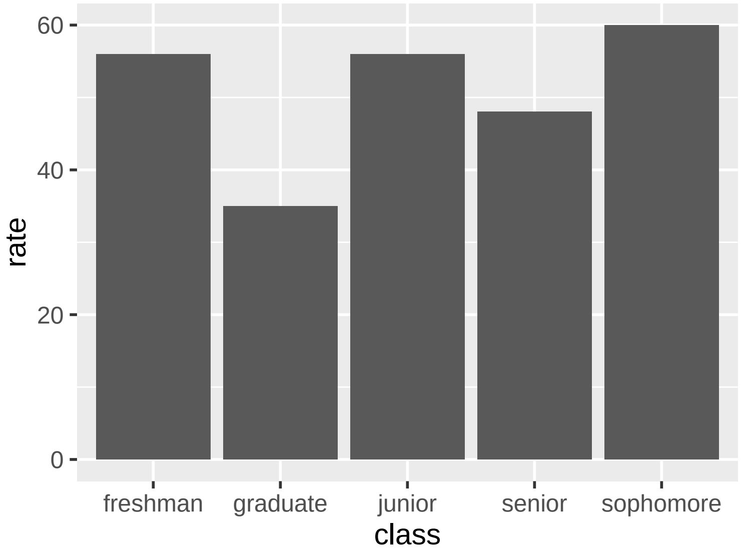
```r
df$class <- factor(class, levels=class)
str(df$class)

##  Factor w/ 5 levels "freshman","graduate",..: 1 2 3 4 5
```

(b) Now generate a bar chart showing the rate of different class.

Answer:

```r
ggplot(df, aes(x = class, y = rate)) +
  geom_bar(stat = "identity")
```

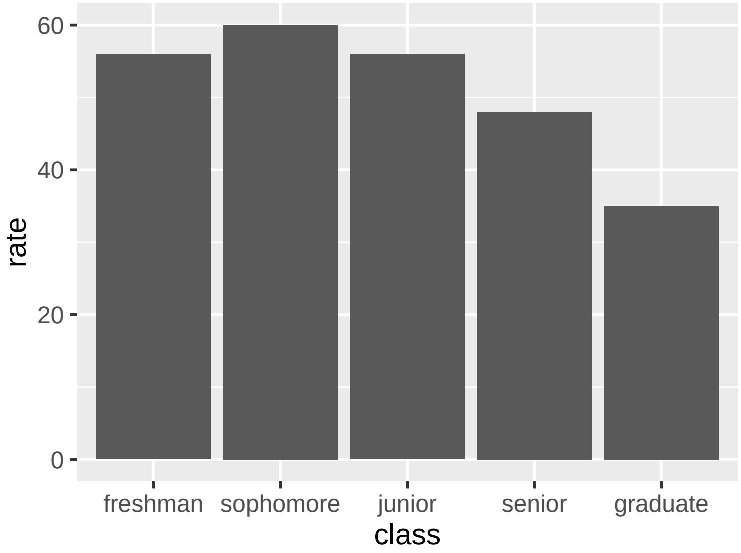


(c) Notice the order of the levels of class is not natural, instead the plot shows the dictionary order. Now, order the bars according to the natural order of the levels of the class (freshman, sophomore, junior, senior, graduate) and regenrate the bar graph.

Answer:

```r
df$class <- factor(class, levels = c("freshman",
                                     "sophomore",
                                     "junior",
                                     "senior",
                                     "graduate"))

ggplot(df, aes(x = class, y = rate)) +
  geom_bar(stat = "identity")
```


60
40
20
0
rate
freshman
sophomore
junior
senior
graduate
class

7. **Bonus (2 points)** for undergraduates and mandatory for graduate students. Suppose we have a vector of data as follows:

```
myVector <- c(1, 2, 3, 4, 5, 6, 7, 8)
```

(a) Using function `tapply()` compute the mean of first three values, next two values and rest of the three values. Show your codes and your result should be 2, 4.5 and 7.

Answer:

```
myVector <- c(1, 2, 3, 4, 5, 6, 7, 8)

tapply(myVector, c(1, 1, 1, 2, 2, 3, 3, 3), FUN=mean)

##   1   2   3
## 2.0 4.5 7.0
```

(b) Now compute the sum of squares instead of mean that you have done in question 7a. Show your codes and your result should be 14, 41 and 149.

Answer:

```
myVector <- c(1, 2, 3, 4, 5, 6, 7, 8)

sumSq <- function(x) {
  return(sum(x^2))
}
tapply(myVector, c(1, 1, 1, 2, 2, 3, 3, 3), FUN=sumSq)

##   1   2   3
##  14  41 149
```