Homework 2
STAT 4410/8416 Section 001
Fall 2014
Due: September 12, 2014 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;

```
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.

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

Answer: The following codes show how we can do the same operation without for loop. Our Result Y is same as X above and we demonstrate that using function `identical()`.

```
Y <- round(t(t(M)/V), 2)
identical(X,Y)

## [1] TRUE

# or if you want to use apply
Z <- t(round(apply(M,1,'/',V), 2))
identical(X,Z)

## [1] TRUE

# yet another solution
VV <- matrix(rep(V, each=n), ncol=k)
W <- round(M/VV, 2)
identical(X,W)

## [1] TRUE
```

The above three solutions demonstrate three concepts. Please make sure you understand those concepts. I leave it as an excersize to test which one performs better with large data.

(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:

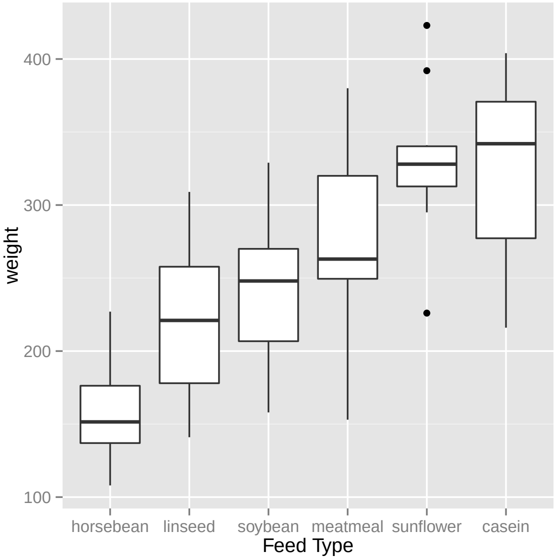
```
set.seed(4286)
n <- 400
k <- 500
V <- sample(seq(4), size=k, replace=T)
M <- matrix(rnorm(n*k), ncol=k)
```

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

##    user  system elapsed
##   0.031   0.000   0.032

system.time(round(t(t(M)/V), 2))

##    user  system elapsed
##   0.009   0.001   0.009
```

2. The data set `chickwts` contains the Chicken Weights by Feed Type. Draw a side by side boxplot of weight for each feed type. Order the feed type based on the median weight. Provide your codes and the plot. Which food type is responsible for highest median weight of the chicken?

   Answer: Food type casein is responsible for highest weights. The codes and the plot are given below.

```
library(ggplot2)
ggplot(chickwts, aes(reorder(feed, weight, median), weight)) +
  geom_boxplot() + xlab("Feed Type")
```



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.

```
library(maps)
mdat <- map_data("state")
head(mdat)

##     long   lat group order  region subregion
## 1 -87.46 30.39     1     1 alabama      <NA>
## 2 -87.48 30.37     1     2 alabama      <NA>
## 3 -87.53 30.37     1     3 alabama      <NA>
## 4 -87.53 30.33     1     4 alabama      <NA>
## 5 -87.57 30.33     1     5 alabama      <NA>
## 6 -87.59 30.33     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.

```
# obtaining the state names in lower case
state <- tolower(row.names(USArrests))

# Standardizing the crime rates as suggested
std_crime <- scale(USArrests)

arrest <- data.frame(state, std_crime)
head(arrest)
```

```
##                 state  Murder Assault UrbanPop      Rape
## Alabama       alabama 1.24256  0.7828  -0.5209 -0.003416
## Alaska         alaska 0.50786  1.1068  -1.2118  2.484203
## Arizona       arizona 0.07163  1.4788   0.9990  1.042878
## Arkansas     arkansas 0.23235  0.2309  -1.0736 -0.184917
## California california 0.27827  1.2628   1.7589  2.067820
## Colorado     colorado 0.02571  0.3989   0.8608  1.864967
```

(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`.

```
arrestdat <- merge(mdat, arrest, by.x='region', by.y='state', all.x=T)
odat <- arrestdat[order(arrestdat$order),]
head(odat)
```

```
##    region   long   lat group order subregion Murder Assault UrbanPop
## 1 alabama -87.46 30.39     1     1      <NA>  1.243  0.7828  -0.5209
## 2 alabama -87.48 30.37     1     2      <NA>  1.243  0.7828  -0.5209
## 6 alabama -87.53 30.37     1     3      <NA>  1.243  0.7828  -0.5209
## 7 alabama -87.53 30.33     1     4      <NA>  1.243  0.7828  -0.5209
## 8 alabama -87.57 30.33     1     5      <NA>  1.243  0.7828  -0.5209
## 9 alabama -87.59 30.33     1     6      <NA>  1.243  0.7828  -0.5209
##        Rape
## 1 -0.003416
## 2 -0.003416
## 6 -0.003416
## 7 -0.003416
## 8 -0.003416
## 9 -0.003416
```

(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.

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

```
##     long   lat group  region Murder Assault UrbanPop      Rape
## 1 -87.46 30.39     1 alabama  1.243  0.7828  -0.5209 -0.003416
## 2 -87.48 30.37     1 alabama  1.243  0.7828  -0.5209 -0.003416
## 6 -87.53 30.37     1 alabama  1.243  0.7828  -0.5209 -0.003416
## 7 -87.53 30.33     1 alabama  1.243  0.7828  -0.5209 -0.003416
## 8 -87.57 30.33     1 alabama  1.243  0.7828  -0.5209 -0.003416
## 9 -87.59 30.33     1 alabama  1.243  0.7828  -0.5209 -0.003416
```

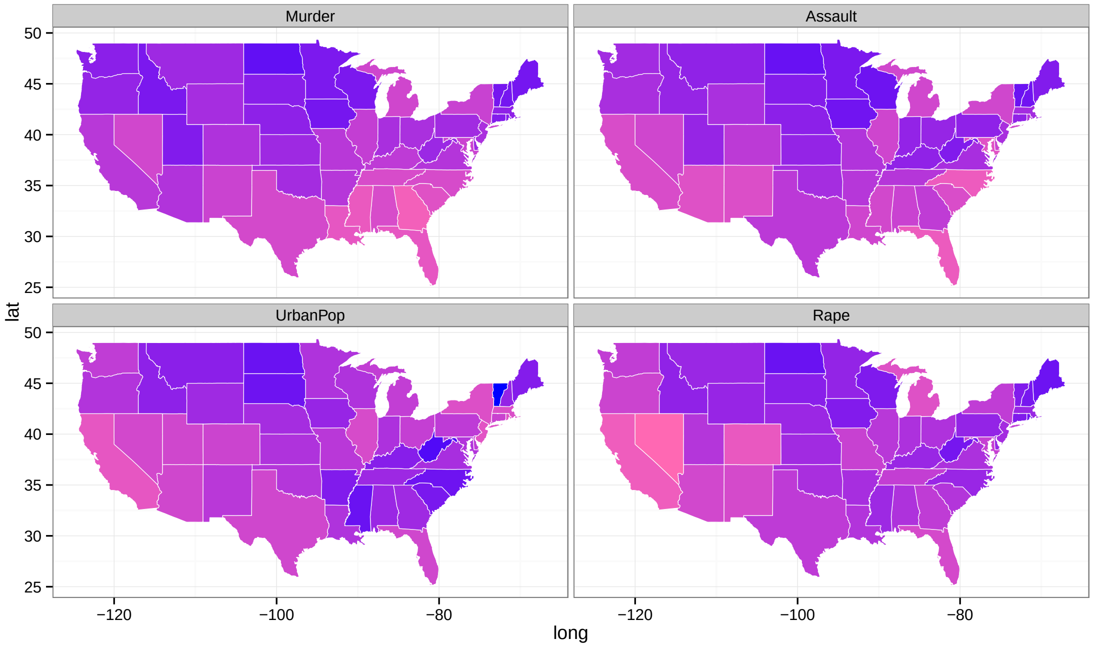
(e) Melt the data frame `sdat` with id variables long, lat, group, region. Store the molten data in `msdat` and report first few rows of data.

```
library(reshape2)
msdat <- melt(sdat, id=c("long", "lat", "group", "region"))
head(msdat)
```

```
##     long   lat group  region variable value
## 1 -87.46 30.39     1 alabama   Murder 1.243
## 2 -87.48 30.37     1 alabama   Murder 1.243
## 3 -87.53 30.37     1 alabama   Murder 1.243
## 4 -87.53 30.33     1 alabama   Murder 1.243
## 5 -87.57 30.33     1 alabama   Murder 1.243
## 6 -87.59 30.33     1 alabama   Murder 1.243
```

(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.

```
library(scales)
ggplot(msdat, aes(x=long, y=lat,group=group)) +
  geom_polygon(aes(fill=value), colour = alpha("white", 1/2), size = 0.2) +
  theme_bw() + theme(legend.position = "none") +
  scale_fill_continuous(low="blue", high="hotpink") + 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: By visual inspection we see the following states with highest crime rates for the respective crimes as shown in the table below.

crimes	states
Murder	Mississipi, Georgia
Assault	North Carolina, Florida
Rape	Nevada, California

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

Answer: No, we don't think it is true. Murder rates are highest for Mississippi and Georgia but their urban population is among the smallest. On the other hand the urban population was larger for California or New York but their murder rates are not among the tops.

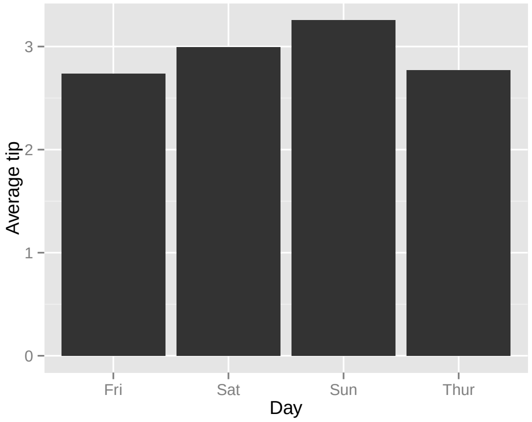
(h) In question (3b) we standardized the crime rates. Why do you think we did this? Explain what would happen if we would not do this.

(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.

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

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

```r
avg_tip <- tapply(tips$tip, tips$day, mean)
avg_dat <- melt(avg_tip)
ggplot(avg_dat, aes(Var1, value)) + geom_bar(stat="identity") +
  xlab("Day") + ylab("Average tip")
```



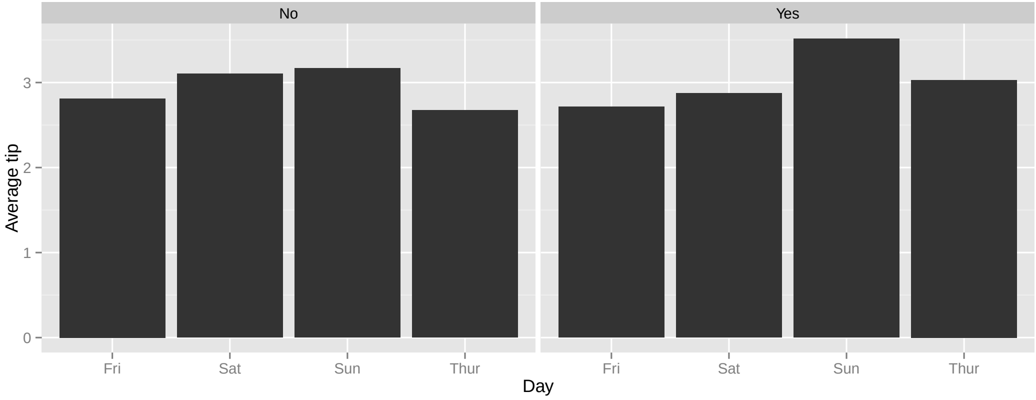
(b) Compute the average tip, total tip and average size grouped by smoker and day. i.e., For each combination of smoker and day you should have a row of these summaries. Report the result in a nice table.

```r
library(dplyr)
df <- tips %>%
  group_by(smoker, day) %>%
  summarize(avg.tip = mean(tip),
            tot.tip = sum(tip),
            avg.size = mean(size))
library(knitr)
kable(df, format = 'latex', booktabs = TRUE, digits=2)
```

smoker	day	avg.tip	tot.tip	avg.size
No	Fri	2.81	11.25	2.25
No	Sat	3.10	139.63	2.56
No	Sun	3.17	180.57	2.93
No	Thur	2.67	120.32	2.49
Yes	Fri	2.71	40.71	2.07
Yes	Sat	2.88	120.77	2.48
Yes	Sun	3.52	66.82	2.58
Yes	Thur	3.03	51.51	2.35

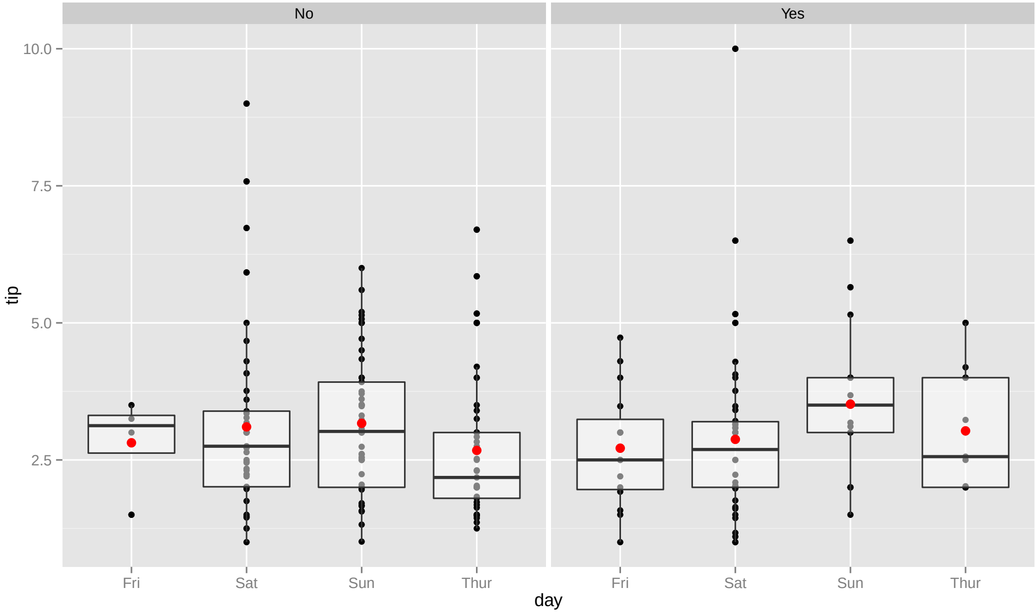
(c) Create a bar chart that shows average tip by day and also faceted by smoker.

```r
ggplot(df, aes(day, avg.tip)) + geom_bar(stat="identity") +
  xlab("Day") + ylab("Average tip") + facet_wrap(~smoker)
```



(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.

```
ggplot(tips, aes(day, tip)) + geom_point() +
 # geom_violin(alpha=1/2) +
  geom_boxplot(alpha=1/2) +
  stat_summary(fun.y = mean, geom = "point",  color='red', size=3) +
  facet_wrap(~smoker)
```



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.

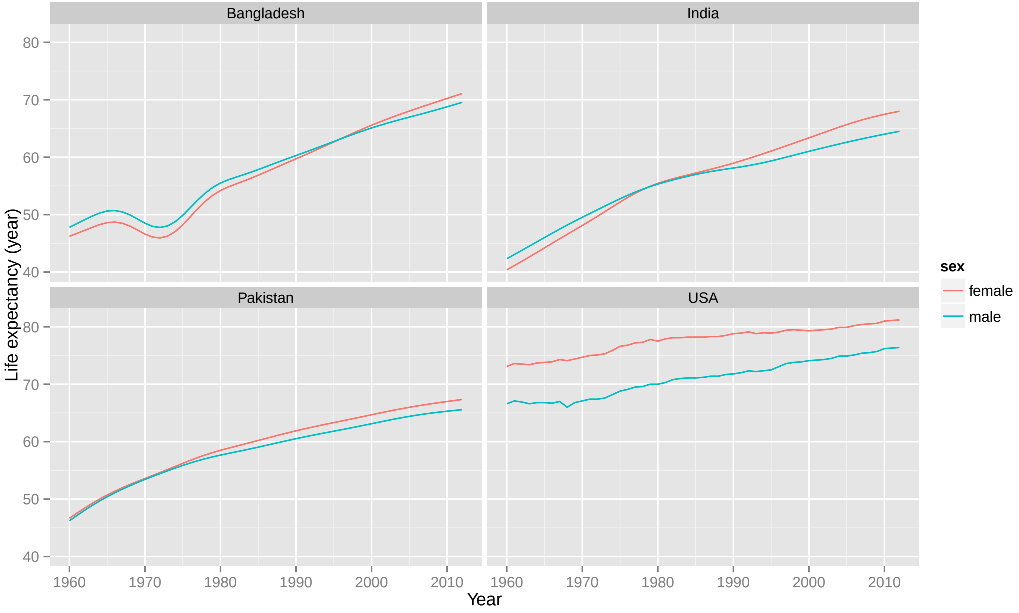
(a) Read the file from your hard drive and display first few rows of the data.

```
dat <- read.csv("life-expectancy.csv")
head(dat)
```

```
##   year    sex Bangladesh India Pakistan  USA
## 1 1960 female      46.22 40.39    46.66 73.1
## 2 1960   male      47.79 42.33    46.22 66.6
## 3 1961 female      46.73 41.12    47.56 73.6
## 4 1961   male      48.45 43.05    47.16 67.1
## 5 1962 female      47.25 41.88    48.43 73.5
## 6 1962   male      49.10 43.78    48.04 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.

```
mdat <- melt(dat, id=c("year","sex"))
ggplot(mdat, aes(as.numeric(year), value)) + geom_line(aes(color=sex)) +
  facet_wrap(˜variable) + xlab("Year") + ylab("Life expectancy (year)")
```



(c) Explain what interesting features you notice in the plot of question 5b.