HW: Week 4

36-350 – Statistical Computing

Week 4 – Spring 2021

Name: Jacky Liu

Andrew ID: jackyl1

You must submit **your own** HW as a PDF file on Gradescope.

Question 1

(20 points)

You are given the following matrix:

```
set.seed(505)
mat = matrix(rnorm(900),30,30)
mat[sample(30,1),sample(30,1)] = NA
```

Compute the standard deviation for each row, using `apply()` and your own on-the-fly function, i.e., a function that is defined *within* the argument list being passed to `apply()`. **Do not use the function sd()!** Realize that since there is a missing value within the matrix, you need to define your function so as to only take into account the non-missing data in each row. If your vector of standard deviations has an `NA` in it, then your function isn't quite working yet.

```
stdev = function(row) {
  # create a new vector with no NA values
  r = row[!is.na(row)]
  # calculate mean
  mean = sum(r) / length(r)
  # implement stdev formula
  summation = 0
  for(num in r){
    summation = summation + (num - mean)^2
  }
  st_dev = sqrt(summation/(length(r)-1))
  return(st_dev)
}
# compute the standard deviation for each row
apply(mat, 1, stdev)
```

```
##  [1] 1.2235111 0.9996540 0.8324186 0.7935861 0.9546933 1.1166745 1.0264495
##  [8] 0.7135952 1.0357715 0.9023740 1.2146342 0.9665977 1.1364236 0.7335094
## [15] 0.8758855 1.0529671 1.0303302 0.8857679 1.1004938 0.9636788 0.9981597
## [22] 1.1224219 1.2828417 0.9777383 0.9223948 0.8506261 0.8840344 0.6538431
## [29] 0.8304627 1.0001846
```

Below we read in the data on the political economy of strikes.

```
strikes.df = read.csv("http://www.stat.cmu.edu/~mfarag/350/strikes.csv")
```

Question 2

(20 points)

Using `split()` and `sapply()`, compute the average unemployment rate, inflation rates, and strike volume for each year represented in the `strikes.df` data frame. The output should be a matrix of dimension 3 $\times$ 35. (You need not display the matrix contents... just capture the output from `sapply()` and pass that output to `dim()`.) Provide appropriate row names (see `rownames()` to your output matrix. Display the columns for 1962, 1972, and 1982. (This can be done in one line as opposed to three.)

```
unemp = strikes.df$unemployment
unemp = split(unemp, f=strikes.df$year)
unemployment.means = sapply(unemp, FUN=mean)
infl = strikes.df$inflation
infl = split(infl, f=strikes.df$year)
inflation.means = sapply(infl, FUN=mean)
strk = strikes.df$strike.volume
strk = split(strk, f=strikes.df$year)
strikevol.means = sapply(strk, FUN=mean)

mat = rbind(unemployment.means, inflation.means, strikevol.means)

mat = data.matrix(mat)
dim(mat)
```

```
## [1]  3 35
```

```
mat[,c(12,22,32)]
```

```
##                           1962       1972       1982
## unemployment.means    2.127778   2.705556   6.805882
## inflation.means       3.738889   6.238889   9.594118
## strikevol.means     214.555556 387.111111 227.882353
```

Question 3

(20 points)

Utilize piping and `group_by()`, etc., to compute the average unemployment rate for each country, and display that average for only those countries with the maximum and minimum averages. To be clear: your output should only show average unemployment for Ireland and Switzerland, and nothing else. (Hint: remember `slice()`, a less-often-used `dplyr` function.) Hint: arrange your output in order of descending average unemployment, then note that `n()` applied as an argument to the right function will return the last row.

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.1
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df = group_by(strikes.df, country) %>% summarize(unemp_rate = mean(unemployment))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
df = df[order(-df$unemp_rate),]
df %>% slice(c(1, n()))
```

```
## # A tibble: 2 x 2
##   country     unemp_rate
##   <fct>            <dbl>
## 1 Ireland          7.77
## 2 Switzerland      0.329
```