HW: Week 4

36-350 – Statistical Computing

Week 4 – Spring 2022

Name: Ethan Vertal

Andrew ID: evertal

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

Question 1

(40 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.

```
apply(mat, 1, function(r)
{
  t = r[!is.na(r)]
  return(sqrt(sum((t-mean(t))^2)/(length(t)-1)))
})
```

```
##  [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

(40 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 × 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.)

```
strikes.sapply <- sapply(split(strikes.df, f=strikes.df$year),
       function(x) {
         c("Unemployment Rate" = mean(x$unemployment),
           "Inflation Rates" = mean(x$inflation),
           "Strike Volume" = mean(x$strike.volume))
       })

dim(strikes.sapply)
```

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

```
rownames(strikes.sapply)
```

```
## [1] "Unemployment Rate" "Inflation Rates"   "Strike Volume"
```

```
strikes.sapply[,c("1962", "1972", "1982")]
```

```
##                         1962       1972       1982
## Unemployment Rate   2.127778   2.705556   6.805882
## Inflation Rates     3.738889   6.238889   9.594118
## Strike Volume     214.555556 387.111111 227.882353
```

Question 3

(40 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.

```
if ( require(tidyverse) == FALSE ) {
  install.packages("tidyverse",repos="https://cloud.r-project.org")
  suppressWarnings(library(tidyverse))
}
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
strikes.df %>% group_by(., country) %>% summarise_at(., vars(unemployment),
                                                      list(unemployment = mean)) %>%
                                                      arrange(., -unemployment) %>%
                                                      slice(., 1, n())
```

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