# 36350-A HW4

Joong Ho Choi

TOTAL POINTS

**60 / 60**

QUESTION 1

## 1 Q1 **20 / 20**

✓ **- 0 pts** Correct

**- 4 pts** does not define function within argument list

**- 3 pts** does not take into missing values within function at all

**- 2 pts** removes some but not all missing values within function

**- 2 pts** sd calculation is incorrect (may use n or n-1)

**- 4 pts** did not link question

**- 20 pts** blank

**- 2 pts** partially linked question

QUESTION 2

## 2 Q2 **20 / 20**

✓ **- 0 pts** Correct

**- 4 pts** does not use sapply()

**- 3 pts** does not use split()

**- 3 pts** does not display the dimensions

**- 2 pts** does not add appropriate row names

**- 2 pts** the displayed dimensions are incorrect

**- 2 pts** does not display example output from 1962, 1972, and 1982

**- 2 pts** incorrect output for 1962, 1972, and 1982

**- 4 pts** did not link question

**- 20 pts** blank answer

**- 2 pts** partially linked question

QUESTION 3

## 3 Q3 **20 / 20**

✓ **- 0 pts** Correct

**- 0 pts** Please make sure to limit the length of each line of code next time, especially when you are using piping

**- 5 pts** does not use piping at all

**- 2 pts** only partially uses piping

**- 1 pts** does not use group_by

**- 4 pts** does not display output

**- 3 pts** the displayed output is incorrect

**- 4 pts** did not link question

**- 20 pts** blank answer

**- 2 pts** partially linked question

QUESTION 4

## 4 Late Penalties **0 / 0**

✓ **- 0 pts** Correct

**- 54 pts** Late, submitted Saturday 8:50PM (100% off)

**- 10 pts** Late, submitted before Friday 12AM

# HW: Week 4

36-350 – Statistical Computing

Week 4 – Spring 2021

Name: Joong Ho Choi

Andrew ID: joonghoc

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.

```
# FILL ME IN
#apply(mat,1,function(x){sd(x,na.rm=TRUE)} )
apply(mat,1,function(x){sqrt(sum((x - mean(x,na.rm=TRUE))^2,na.rm=TRUE) / (length(x[!is.na(x)]) - 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")
```

---

1 Q1 **20 / 20**

- ✓ **- 0 pts** Correct
- **- 4 pts** does not define function within argument list
- **- 3 pts** does not take into missing values within function at all
- **- 2 pts** removes some but not all missing values within function
- **- 2 pts** sd calculation is incorrect (may use n or n-1)
- **- 4 pts** did not link question
- **- 20 pts** blank
- **- 2 pts** partially linked question

## 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 × 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.)

```
# FILL ME IN
ans<-split(strikes.df,strikes.df$year)
help=function(x){return (c("mn.unemplyment"=mean(x$unemployment),
                           "mn.inflation"=mean(x$inflation),
                           "mn.strike_volume"=mean(x$strike.volume)))}
res<-sapply(ans,FUN=help)
dim(res)
```

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

```
rownames(res)
```

```
## [1] "mn.unemplyment"   "mn.inflation"     "mn.strike_volume"
```

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

```
##                        1962       1972       1982
## mn.unemplyment     2.127778   2.705556   6.805882
## mn.inflation       3.738889   6.238889   9.594118
## mn.strike_volume 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.

```
# FILL ME IN
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

## 2 Q2 20 / 20

✓ **- 0 pts** Correct

**- 4 pts** does not use sapply()

**- 3 pts** does not use split()

**- 3 pts** does not display the dimensions

**- 2 pts** does not add appropriate row names

**- 2 pts** the displayed dimensions are incorrect

**- 2 pts** does not display example output from 1962, 1972, and 1982

**- 2 pts** incorrect output for 1962, 1972, and 1982

**- 4 pts** did not link question

**- 20 pts** blank answer

**- 2 pts** partially linked question

## 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 × 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.)

```
# FILL ME IN
ans<-split(strikes.df,strikes.df$year)
help=function(x){return (c("mn.unemplyment"=mean(x$unemployment),
                           "mn.inflation"=mean(x$inflation),
                           "mn.strike_volume"=mean(x$strike.volume)))}
res<-sapply(ans,FUN=help)
dim(res)
```

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

```
rownames(res)
```

```
## [1] "mn.unemplyment"   "mn.inflation"     "mn.strike_volume"
```

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

```
##                        1962       1972       1982
## mn.unemplyment     2.127778   2.705556   6.805882
## mn.inflation       3.738889   6.238889   9.594118
## mn.strike_volume 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.

```
# FILL ME IN
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
result<-strikes.df %>% group_by(country) %>%
  summarize(mn.unemployment=mean(unemployment,na.rm=TRUE))%>%
  arrange(.,decs=(mn.unemployment))%>%
  slice(.,n=1,n())
result
```

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

## 3 Q3 20 / 20

✓ **- 0 pts** Correct

**- 0 pts** Please make sure to limit the length of each line of code next time, especially when you are using piping

**- 5 pts** does not use piping at all

**- 2 pts** only partially uses piping

**- 1 pts** does not use group_by

**- 4 pts** does not display output

**- 3 pts** the displayed output is incorrect

**- 4 pts** did not link question

**- 20 pts** blank answer

**- 2 pts** partially linked question

## 4 Late Penalties **0 / 0**

✓ **- 0 pts** **Correct**

**- 54 pts** Late, submitted Saturday 8:50PM (100% off)

**- 10 pts** Late, submitted before Friday 12AM