# r_intro PS 2

Hnin Ei Phyu

2022-11-10

## Practice Summary Statistics with atomic vector

First thing first, you definitely need `tidyverse` pkg for this exercise. Don't forgot to load it at the top of your answer `rmd` file.

Let's go back to the previous exercise on implementation of the sampling distribution of the means.

2. Construct the list of 100 means values from the `Sepal.Length` and each mean value should construct from 10 sample sizes. (sample size = 10, replication 100 times)
3. Assigned the result (vector with 100 numbers) as vector name called `means_list` and .
4. Construct the another vector called `test` and assigned `1` hundred time in that vector. Use `rep()` function for generation of same value for 100 times.
5. Then, using `cbind` function to combined those two vectors" `test` and `mean_list`, and treat the result as `data.frame` and assigned to the object name `mean_list_1`. You can try with below demo code.

```
means_list_1 <- data.frame(cbind(number, mean_list))
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df<-iris
means_list<-c(replicate(n = 100, expr = mean(sample(df$Sepal.Length, size = 10, replace=TRUE))))
test<-rep(1,100)
means_list_1 <- data.frame(cbind(test, means_list))
```

Repeat the same process from numbers 1 to 4, but using different sample size and replication number this time.

- for `means_list_2`, using a sample size 30 and replication time 200.
- for `means_list_3`, using a sample size 50 and replication time 1,000.

- for `means_list_4`, using a sample size 50 and replication time 3,000.

- for `means_list_5`, using a sample size 50 and replication time 10,000.

```
##means_list_2
means_list<-c(replicate(n = 200, expr = mean(sample(df$Sepal.Length, size = 30, replace=TRUE))))
test<-rep(2,200)
```

```
means_list_2 <- data.frame(cbind(test, means_list))

##means_list_3
means_list<-c(replicate(n = 1000, expr = mean(sample(df$Sepal.Length, size = 50, replace=TRUE))))
test<-rep(3,1000)
means_list_3 <- data.frame(cbind(test, means_list))

##means_list_4
means_list<-c(replicate(n = 3000, expr = mean(sample(df$Sepal.Length, size = 50, replace=TRUE))))
test<-rep(4,3000)
means_list_4 <- data.frame(cbind(test, means_list))

##means_list_5
means_list<-c(replicate(n = 10000, expr = mean(sample(df$Sepal.Length, size = 50, replace=TRUE))))
test<-rep(5,10000)
means_list_5 <- data.frame(cbind(test, means_list))
```

This time, we are going to create the list to store all those `means_list_x` (where `x 1:5`) and assigned that list as `means_seris`.

```
means_list_x<-list(means_list_1,means_list_2,means_list_3,means_list_4,means_list_5)
means_seris<-means_list_x
```

And, perform the following function from that list.

1. calculate the mean of `means_list_x` (where `x 1:5`) from that list (using the command related to filtering list from lecture 2.

```
mean1<-mean(means_seris[[1]][[2]])
mean2<-mean(means_seris[[2]][[2]])
mean3<-mean(means_seris[[3]][[2]])
mean4<-mean(means_seris[[4]][[2]])
mean5<-mean(means_seris[[5]][[2]])

mean_means_list_x<-mean(c(mean1,mean2,mean3,mean4,mean5))
mean_means_list_x
```

```
## [1] 5.842093
```

For example, if we use the following command, we can get the first dataframe. You work is to calculate the mean value of column `mean_list`.

```
means_seris[[1]]
```

2. Using `bind_rows()` function to combined all dataset from `means_seris` and assigned into object called `df_means_combined`. Please make sure that your result dataset should have `16100` observations and 2 variables; `test` and `mean_list`.

```
df_means_combined<-bind_rows(means_seris)
```

3. Finally, calculate the mean value for each group of `test` using following example. In this exercise, we are going to use `group_by` function from the `tidyverse` packge and `%>%` operator (pipe operator).

```
library(tidyverse)
df<-df_means_combined
df%>%
  group_by(test) %>%
  summarise(mean = mean(means_list))
```

```
## # A tibble: 5 x 2
##    test  mean
##   <dbl> <dbl>
## 1     1  5.84
## 2     2  5.84
## 3     3  5.84
## 4     4  5.84
## 5     5  5.84
```