HW_02

Yuke Han

2024-02-17

Problem 1:

```
library(tidyverse)
dt <- tribble(
  ~x, ~y, ~w, ~z,
  210, 300, 220, 180,
  102, 100, 119, 187,
  176, 175, 188, 173,
  87, 95, 91, 94,
  202, 210, 234, 218,
  110, 122, 131, 128
)
dt
```

```
## # A tibble: 6 x 4
##       x     y     w     z
##   <dbl> <dbl> <dbl> <dbl>
## 1   210   300   220   180
## 2   102   100   119   187
## 3   176   175   188   173
## 4    87    95    91    94
## 5   202   210   234   218
## 6   110   122   131   128
```

1_a:

- Use and show a map function to find the "mean" of each column of the dt data table

```
mean_dt <- map(dt, mean)
mean_dt
```

```
## $x
## [1] 147.8333
##
## $y
## [1] 167
##
## $w
## [1] 163.8333
```

```
##
## $z
## [1] 163.3333
```

1__b:

- Use and show a map function to find the "standard deviation" of each column of the dt data table.

```
sd_dt <- map(dt, ~sd(.x, na.rm = TRUE))
sd_dt
```

```
## $x
## [1] 54.45151
##
## $y
## [1] 79.12016
##
## $w
## [1] 58.40348
##
## $z
## [1] 44.66617
```

1__c:

- Use and show a map function that will calculate the "square root" of each value of each column of the data table dt.

```
sqrt_dt <- map_df(dt, sqrt)
sqrt_dt
```

```
## # A tibble: 6 x 4
##       x     y     w     z
##   <dbl> <dbl> <dbl> <dbl>
## 1 14.5  17.3  14.8  13.4
## 2 10.1  10    10.9  13.7
## 3 13.3  13.2  13.7  13.2
## 4  9.33  9.75  9.54  9.70
## 5 14.2  14.5  15.3  14.8
## 6 10.5  11.0  11.4  11.3
```

1__d:

- Use R code to find the "mean", "max", "1st Quartile", "3rd Quartile", "Median", and "Min" for each column of the dt data table. (Hint: You do not have to use a map function)

```
summary_stats <- dt %>%
  summarise(across(everything(), list(mean = ~mean(.x, na.rm = TRUE),
                                      max = ~max(.x, na.rm = TRUE),
                                      Q1 = ~quantile(.x, 0.25, na.rm = TRUE),
```

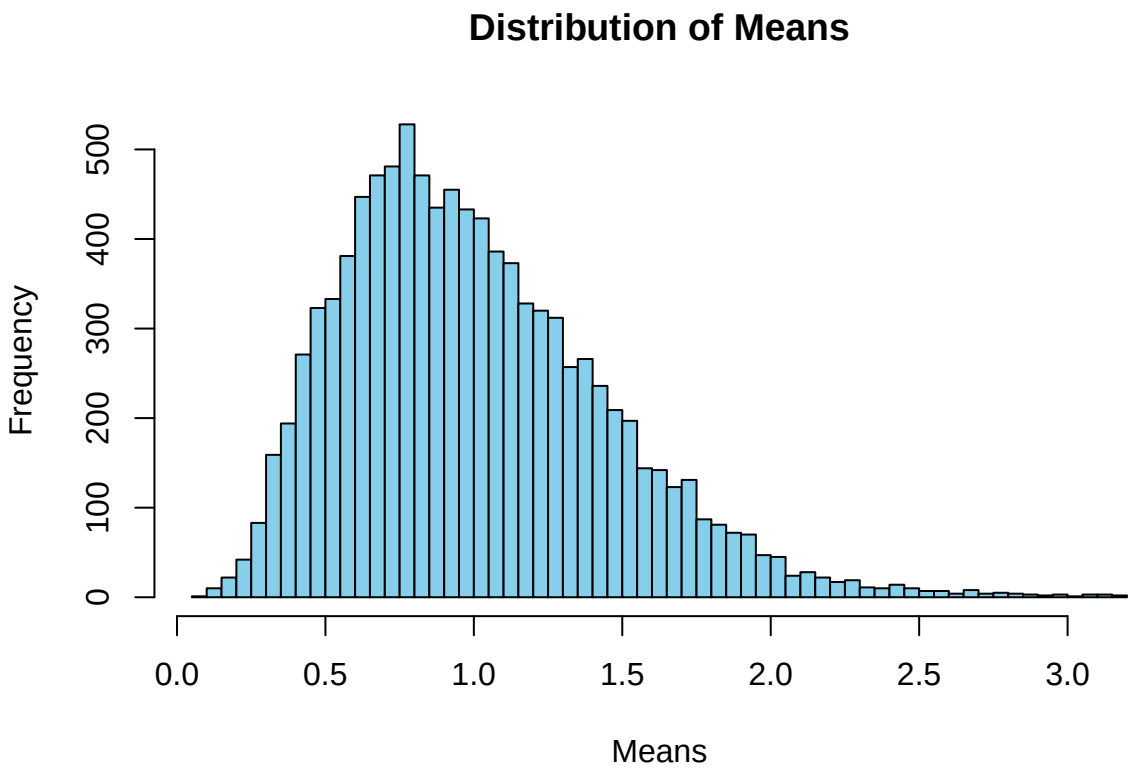
```
                                    median = ~median(.x, na.rm = TRUE),
                                    Q3 = ~quantile(.x, 0.75, na.rm = TRUE),
                                    min = ~min(.x, na.rm = TRUE)))) %>%
  pivot_longer(cols = everything(), names_to = c(".value", "statistic"), names_pattern = "(.*)_(.*)")
print(summary_stats)
```

```
## # A tibble: 6 x 5
##   statistic     x     y     w     z
##   <chr>     <dbl> <dbl> <dbl> <dbl>
## 1 mean       148.  167   164.  163.
## 2 max        210   300   234   218
## 3 Q1         104   106.  122   139.
## 4 median     143   148.  160.  176.
## 5 Q3         196.  201.  212   185.
## 6 min         87    95    91    94
```

Problem 2:

Write a function that uses a for loop that, for each iteration, randomly draws 5 observations from an exponential distribution with "rate" parameter 1 (use rexp()) and calculates its "mean". It should do this 10,000 times. Choose an appropriate plot to plot the distribution of "means".

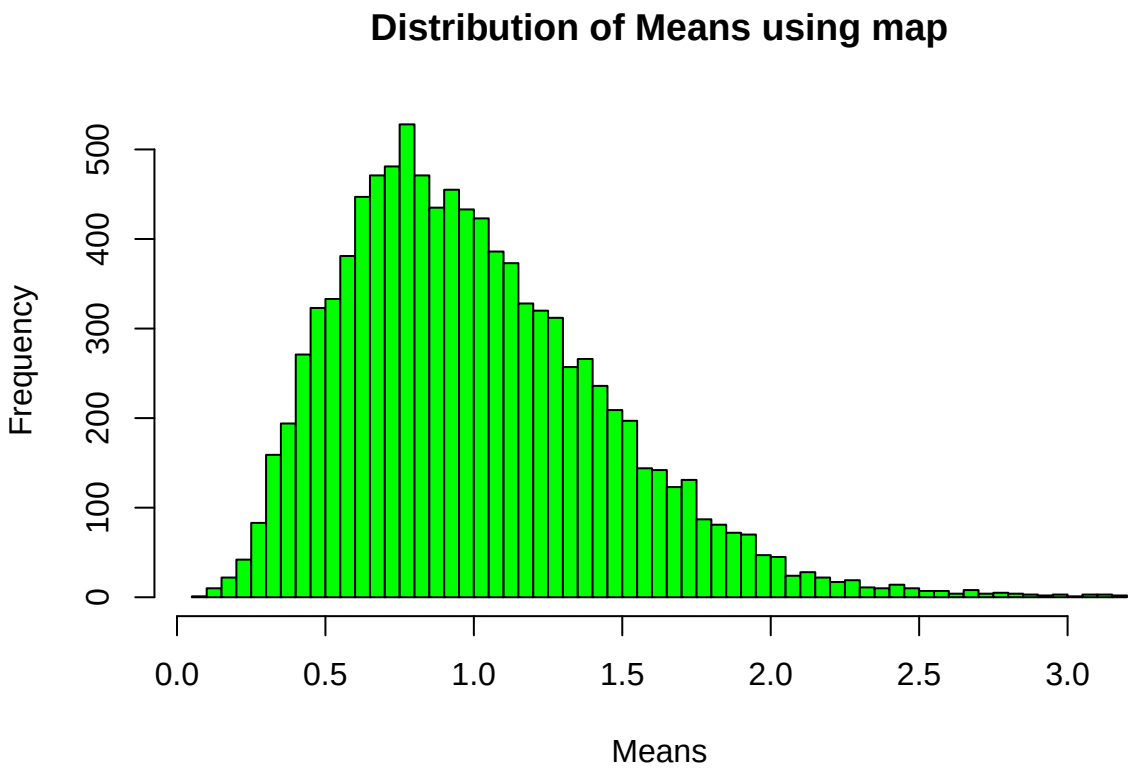
```
simulate_means <- function(n, size, rate) {
  means <- numeric(n)
  for (i in 1:n) {
    sample <- rexp(size, rate)
    means[i] <- mean(sample)
  }
  return(means)
}

set.seed(123)
means <- simulate_means(10000, 5, 1)
hist(means, breaks = 50, main = "Distribution of Means", xlab = "Means", col = 'skyblue')
```



2_a:

- Repeat part 1 by using a map_*() function.

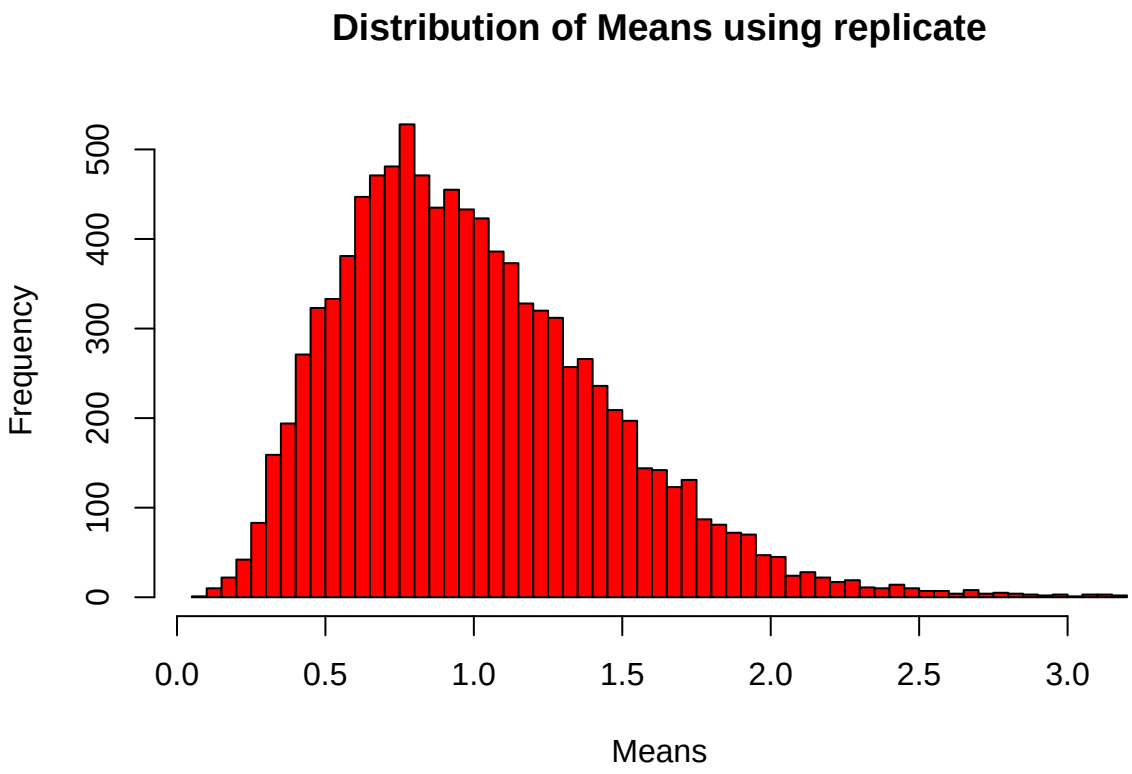
```
set.seed(123)
means_map <- map_dbl(1:10000, ~mean(rexp(5, 1)))
hist(means_map, breaks = 50, main = "Distribution of Means using map", xlab = "Means", col = 'green')
```



2_b:

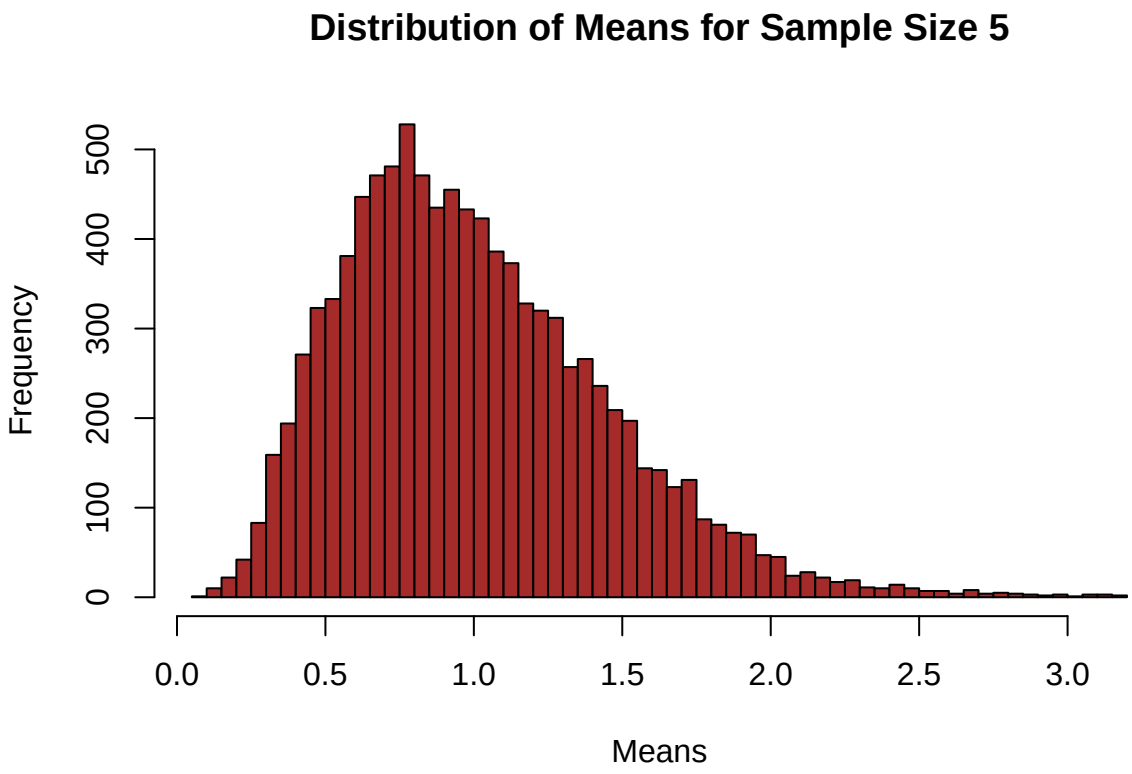
- Repeat part 1 by using the replicate() function.

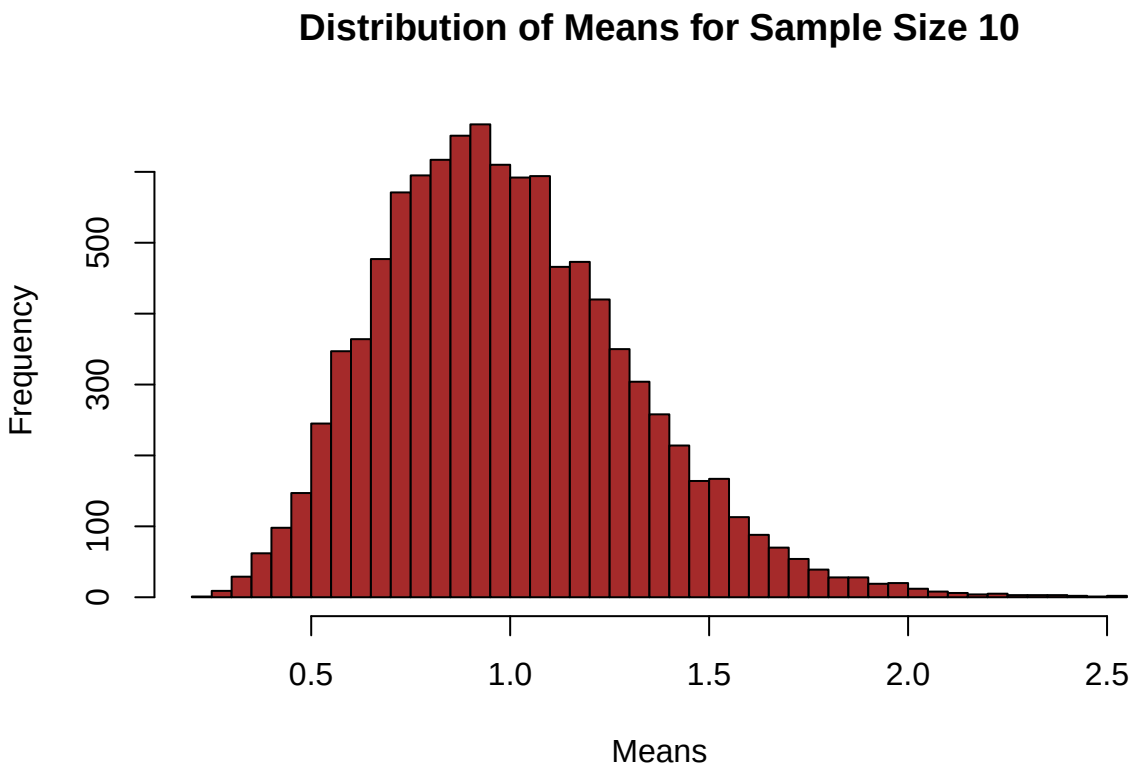
```
set.seed(123)
means_replicate <- replicate(10000, mean(rexp(5, rate = 1)))
hist(means_replicate, breaks = 50, main = "Distribution of Means using replicate", xlab = "Means", col =
```

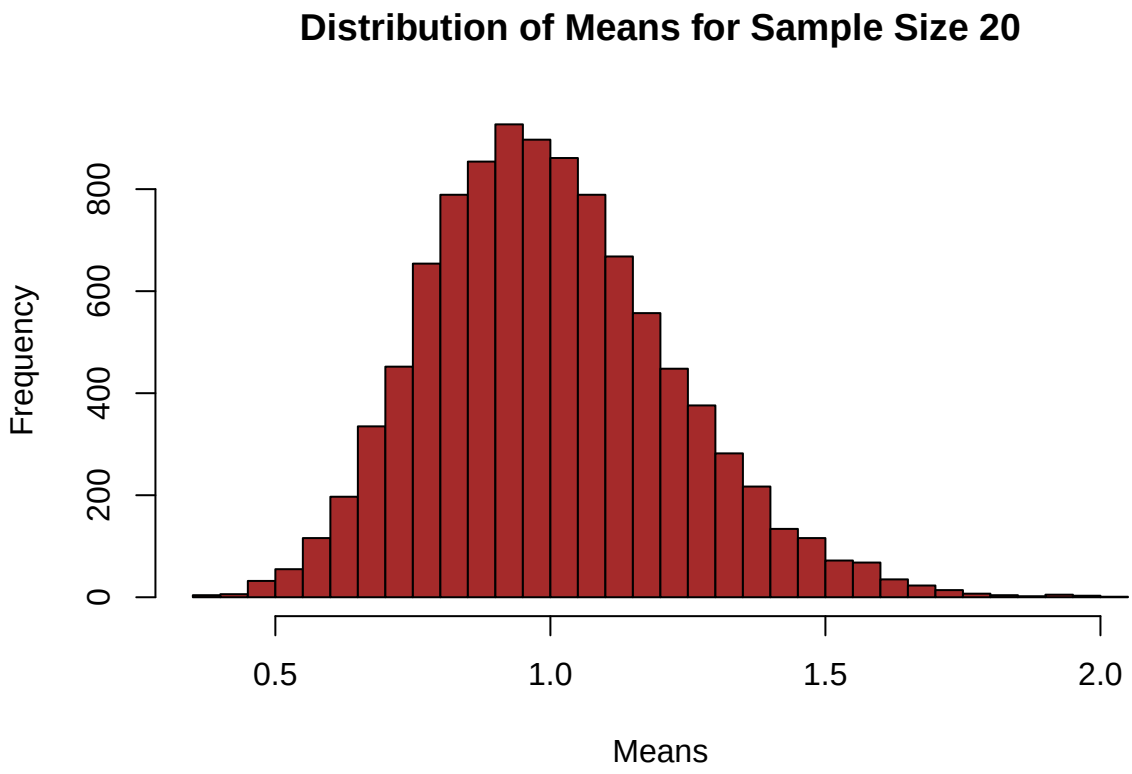


2_c: • Use a another for loop that will print out plots for sample sizes of 5, 10, and 20 observations (instead of just 5).

```
set.seed(123)
sample_sizes <- c(5, 10, 20)
for (size in sample_sizes) {
  means <- replicate(10000, mean(rexp(size, rate = 1)))
  hist(means, breaks = 50, main = paste("Distribution of Means for Sample Size", size), xlab = "Means",
}
```


Distribution of Means for Sample Size 5
Frequency
0
100
200
300
400
500
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Means


Distribution of Means for Sample Size 10
Frequency
0
100
300
500
0.5
1.0
1.5
2.0
2.5
Means



Problem 3:

• Use and show R coding to calculate the "standard deviation" for each variable of the data table mtcars using the "Special For Loop Method".

```
data(mtcars)
output <- vector("double", ncol(mtcars))
for (i in seq_along(mtcars)) {
  output[[i]] <- sd(mtcars[[i]], na.rm = TRUE)
}
names(output) <- names(mtcars)
print(output)
```

```
##         mpg         cyl        disp          hp        drat          wt
##   6.0269481   1.7859216 123.9386938  68.5628685   0.5346787   0.9784574
##        qsec          vs          am        gear        carb
##   1.7869432   0.5040161   0.4989909   0.7378041   1.6152000
```