Homework 04 T-tests, Sampling Distributions, and the Bootstrap

Due by 11:59pm, Saturday, February 18, 2023

S&DS 230/530/ENV 757

(1) Practice with Loops. *(10 points)* For this problem, use loops even if you could do the task without them.

(1.1) A Fibonacci sequence is a series of integers where each number after the 2nd number is found by adding together the two integers before it. Starting with 0 and 1, the sequence goes:

> 0, 1, 1, 2, 3, 5, . . .

Write a loop that fills a vector called `myFib` with this sequence, starting from 0 and 1 (first two entries), and going up to a total length of 30 numbers (that is, `length(myFib)` should be 30). Display the last value in `myFib`.

(1.2) Here is the link to the World Bank data :

> http://www.reuningscherer.net/s&ds230/data/WB.2016.csv

Read the data into a dataframe called `wb`. Write a loop to fill a vector called `naVals` having length equal to the number of columns in the World Bank data frame. The i-th entry in `naVals` should be a number (>= 0) equal to the total number of missing values in the i-th column of World Bank data frame. Make a histogram of `naVals` and label as appropriate.

(For full credit, use only one for-loop to do part b)

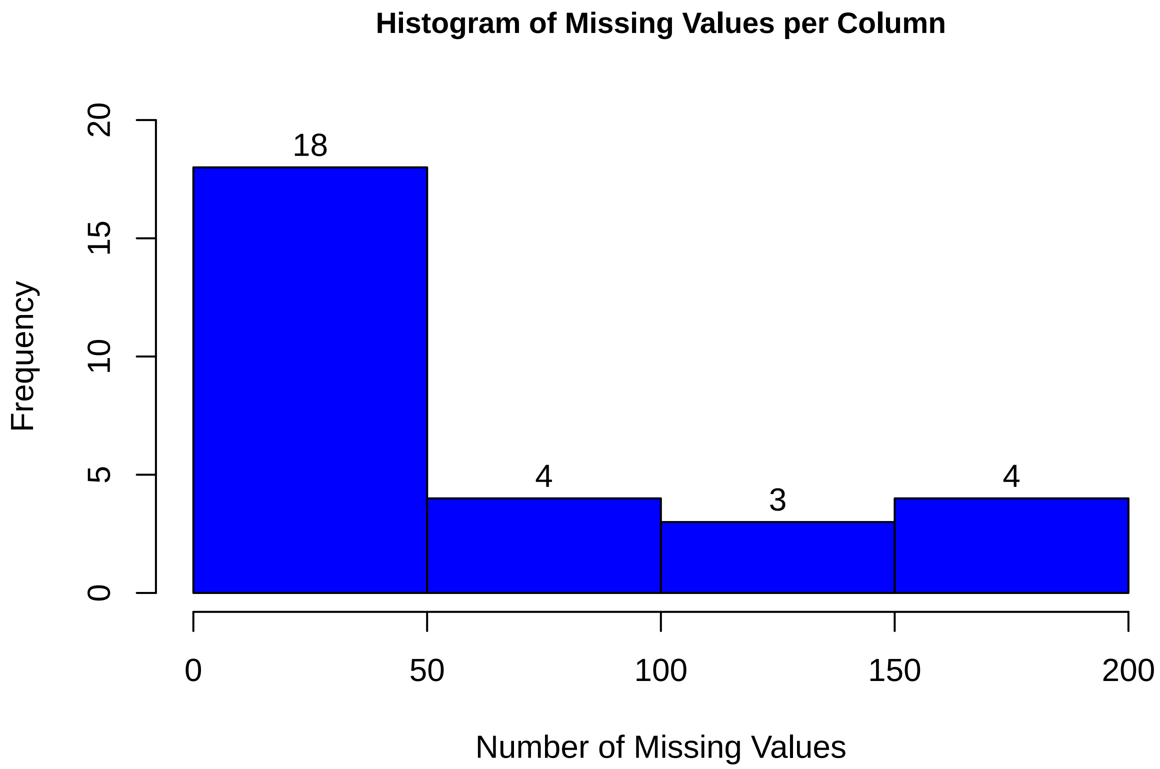
```
# part 1.1

myFib <- rep(NA,30)
myFib[1] <- 0
myFib[2] <- 1
for (i in 3:30) {
   myFib[i] <- myFib[i-1] + myFib[i-2]
}

tail(myFib,1)
```

```
## [1] 514229
```

```
# Part 1.2



wb <- read.csv("http://www.reuningscherer.net/s&ds230/data/WB.2016.csv")
dim(wb)
```

```
## [1] 217  29
```

```
naVals <- rep(NA< 29)
for (i in 1:29) {
  naVals[i] <- sum(is.na(wb[ ,i]))
}

hist(naVals,
     main = "Histogram of Missing Values per Column",
     xlab = "Number of Missing Values",
     col = "blue",
     ylim=c(0,20),
     labels=TRUE,
     cex.main=0.9)
```



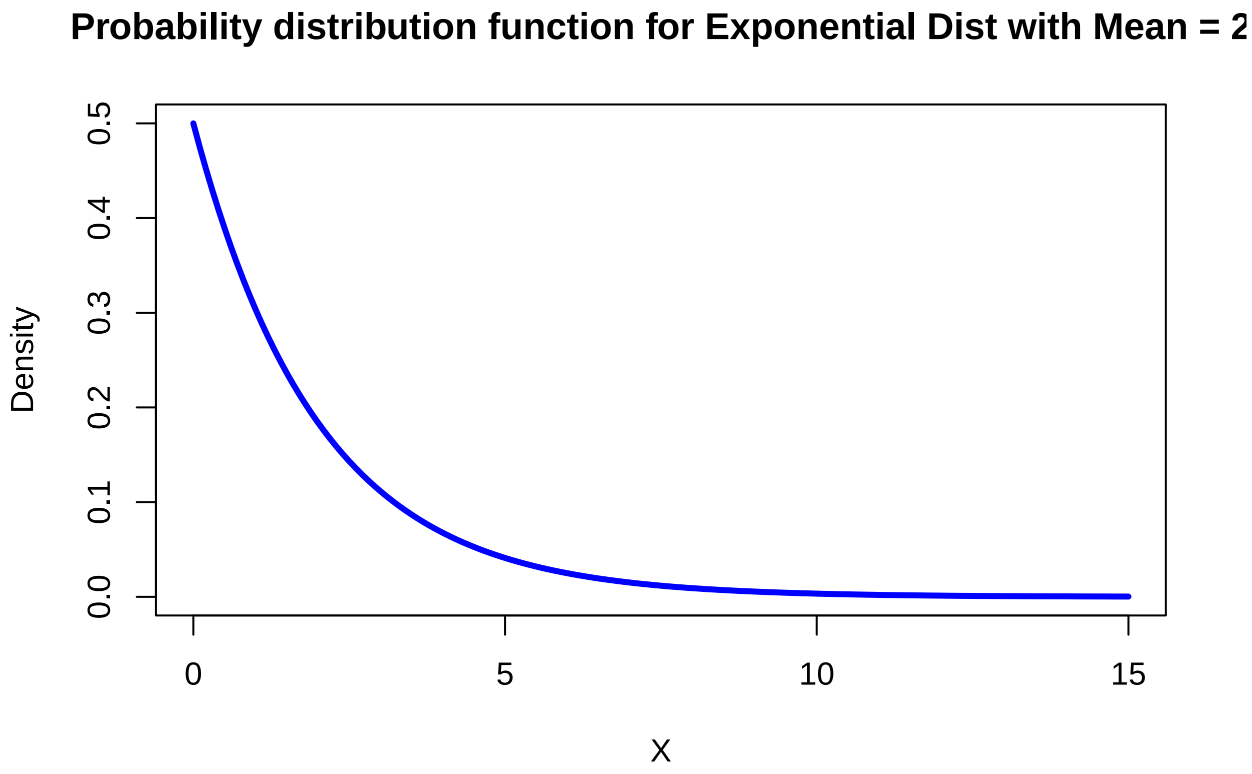
(2) Simulations with the Exponential Distribution *(50 points)*.

For this problem, we'll investigate the sampling distributional characteristics of three statistics. In particular, suppose we take a sample of size 15 from an exponential distribution. We can use the CLT to say something about how far the sample mean is likely to be from the true mean, but how far are the sample median or the sample variance likely to be from the true values in an exponential distribution where we take a sample of size 15? Also, what do we expect the distribution of these statistics to look like!

(2.1) *(8 points)* First, let's get a quick sense of what an exponential distribution looks like where the mean is 2. By the way, it's handy to know that for an exponential distribution with mean 2, the variance is 4 and the median is 2*ln(2). You can read about the exponential distribution HERE.

The code below gives a quick plot of this distribution. Your job is to succinctly answer what each part of the code does. You'll probably need to get help on the function `dexp()`, `seq()` and on a few of the graphics parameters in `par()`.

```
#Get exponential probabilities - note that rate = .5 gives us mean of 2 (mean is 1/rate)
probs <- dexp(seq(0,15, by = .1), rate = .5)

#dexp : gives the density.
#rate : vector of rates.
#by : number: increment of the sequence.

#Plot sampling distribution
plot(seq(0, 15, by = .1), probs,
     main = "Probability distribution function for Exponential Dist with Mean = 2",
     xlab = "X",
     ylab = "Density",
     type = 'l',
     lwd = 3,
     col = "blue"
)
```



```
#type = The type argument helps to customize the plot type. In this example 'l' means line plot.
#lwd : It is the argument to define the line width.
```

(2.2) *(7 points)* Following the example in class 8, get a random sample of 15 observations from an exponential distribution with mean 2. Repeat this process 10000 times. Save your results in a matrix called `samples` with 10,000 rows and 15 columns. The function you'll need is `rexp()`. Display the dimension of `samples'. Show the first 4 rows of`samples‘ but round the values to three decimal places.

```
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)

# FILL IN REMAINING CODE

# The sample size
N <- 15

# Define TIMES, number of times we take a sample of size N
TIMES <- 10000

# Saving results in a matrix - samples

samples <- matrix(rexp(N*TIMES, rate = .5), nrow = TIMES)

# Dimensions of the samples
dim(samples)
```

```
## [1] 10000    15
```

```
# First 4 rows of the samples data
round(samples[1:4, ],3)
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## [1,] 2.644 0.475 9.339 0.902 2.068 2.317 2.376 7.276 0.910 0.851 2.372 4.775
## [2,] 4.560 1.809 1.797 2.778 8.598 1.279 0.188 3.579 1.265 1.037 1.388 1.110
## [3,] 0.515 0.098 0.090 2.143 4.173 1.393 0.754 0.605 1.231 1.703 4.064 0.420
## [4,] 4.644 2.627 4.079 2.981 1.364 0.900 0.706 0.970 4.946 4.951 1.812 1.282
##      [,13] [,14] [,15]
## [1,] 2.650 0.459 0.366
## [2,] 5.233 1.894 1.404
## [3,] 0.938 0.650 0.080
## [4,] 0.883 2.891 0.194
```

(2.3) *(7 points)* Calculate the sample mean for each sample of size 15 (i.e. calculate the mean for each row of `samples`). Repeat this process to get the sample median and the sample variance for each sample of size 15. Save these values in objects called, respectively, `smeans, smedians, svariance`.

```
# Calculating the mean for each row of `samples`

smeans <- apply(samples, 1, mean)
length(smeans)
```

```
## [1] 10000
```

```
# Calculating the median for each row of `samples`

smedians <- apply(samples, 1, median)
length(smedians)
```

```
## [1] 10000
```

```
# Calculating the variance for each row of `samples`

svariance <- apply(samples, 1, var)
length(svariance)
```

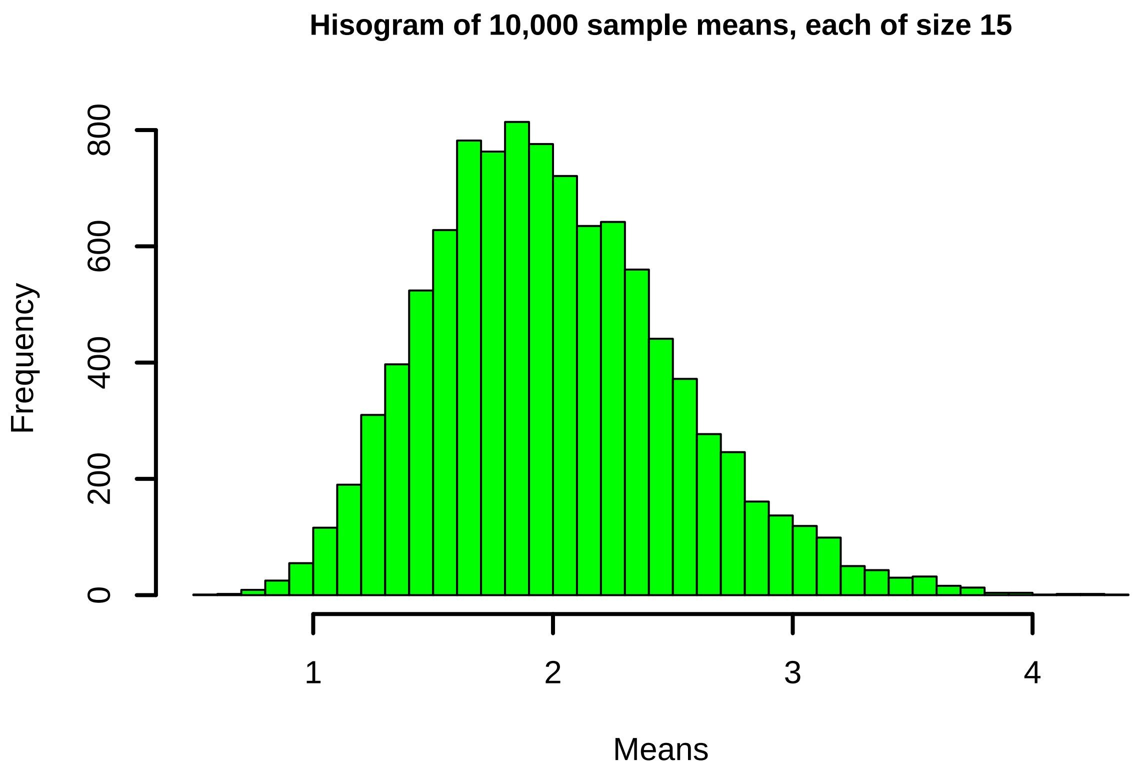
```
## [1] 10000
```

(2.4) *(10 points)* * Create a sample histogram of the sample means (make the bars green, make sure you label your axes and put on a clear title).
* Make a normal quantile plot of the sample means using the `qqPlot()` function in the `car` package. Comment on whether the CLT seems to be in effect. * Get summary statistics OF THE SAMPLE MEANS and save this to an object called `ans1`. Using code, display only the element of `ans` that is the sample mean, rounded to two decimal places. Is this the value you expect?
* Calculate and display the sample standard deviation of the sample means (use the function `sd()`)and display rounded to two decimal places. Then, use code to calculate the value you'd expect based on the CLT, again rounded to two decimal places. Are the two values similar?

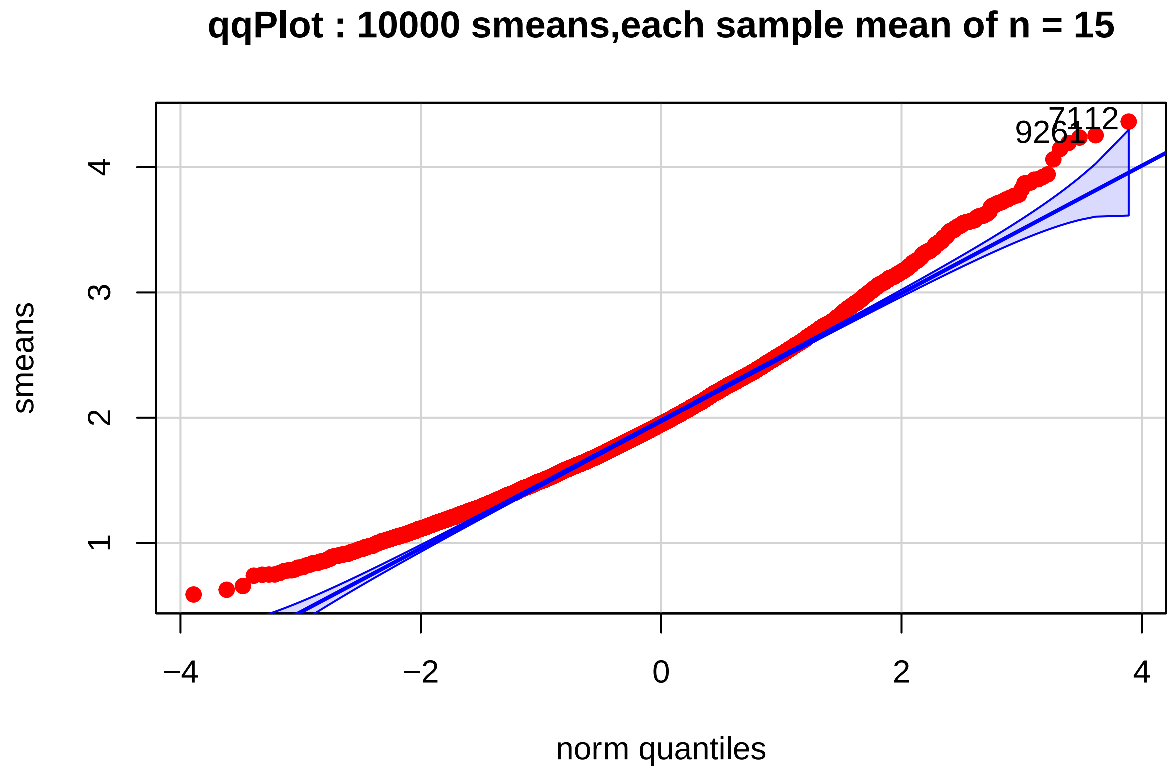
```
library(car)
```

```
## Loading required package: carData
```

```
hist(smeans,
     col = "green",
     main = "Hisogram of 10,000 sample means, each of size 15",
     breaks = 50,
     xlab = "Means",
     lwd = 2,
     cex.main=0.9)
```



```
# normal quantile plot

qqPlot(smeans,
       col = "red",
       pch = 19,
       main = paste("qqPlot : 10000 smeans,each sample mean of n =", N))
```



```
## [1] 7112 9261
```

```
# Summary Statistics of smeans
ans1 <- summary(smeans)
ans1
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5881  1.6317  1.9496  1.9991  2.3187  4.3642
```

```
ans <- round(ans1[4],2)
ans
```

```
## Mean
##    2
```

```
# Standard Deviation

print(paste("Sample SD of the sample means: ", round(sd(smeans), 2)))
```

```
## [1] "Sample SD of the sample means:  0.52"
```

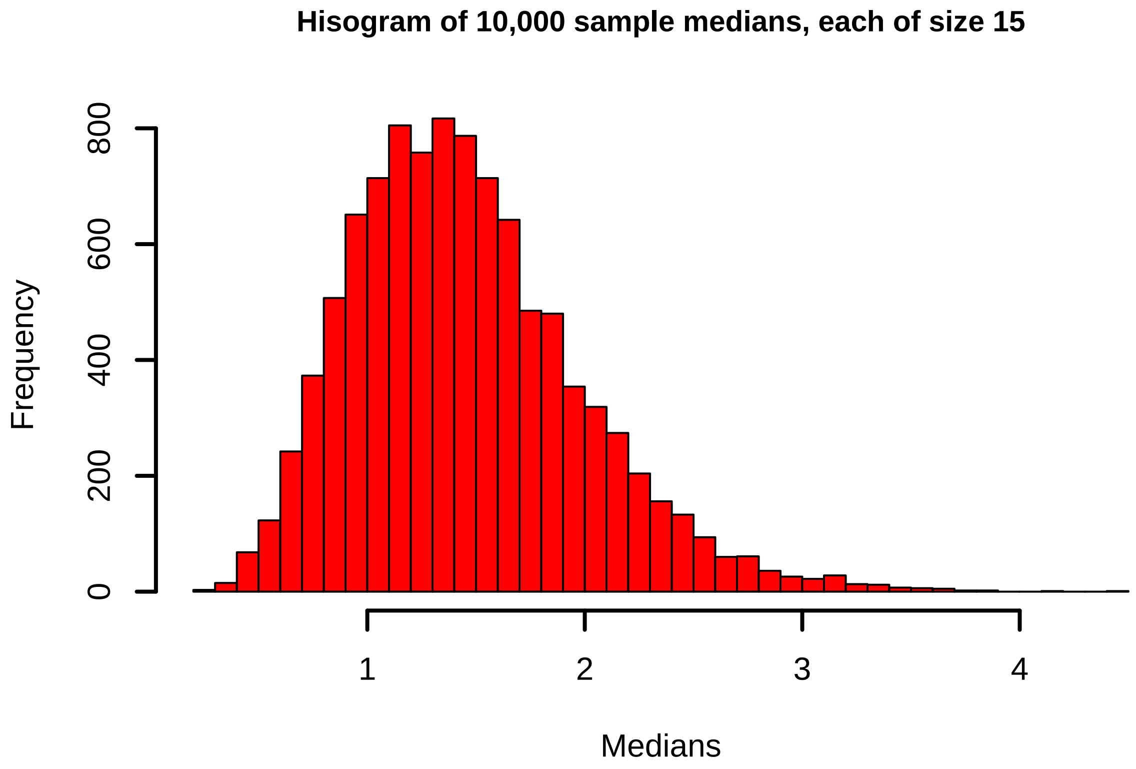
```
print (paste("Theoretical SD of the sample means:", round(sd(samples)/sqrt(15), 2)))
```

```
## [1] "Theoretical SD of the sample means: 0.52"
```

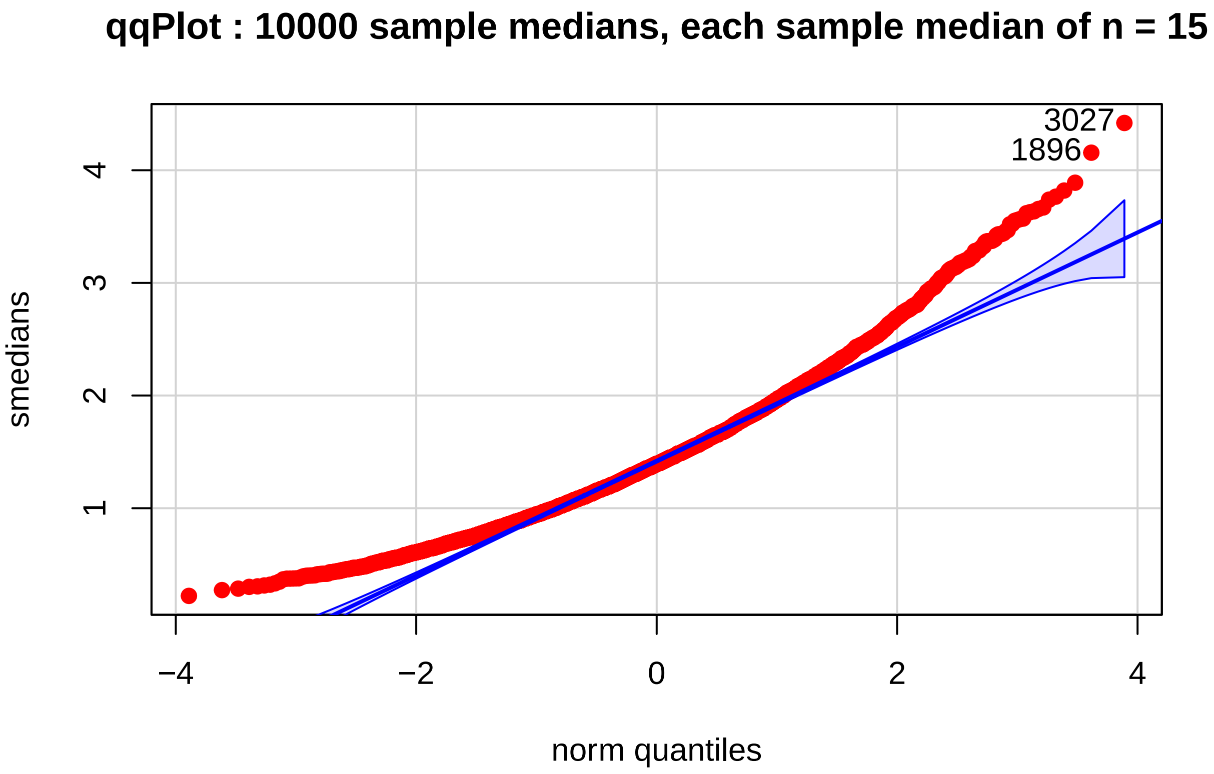
The central limit theorem (CLT) states that the distribution of sample means approximates a normal distribution as the sample size gets larger, regardless of the population's distribution. We know that the mean of the original distribution is 2 and we got the a sample mean of 2; which is same as the original distribution and this what we had expected.In this example the distribution of the sample means is a slightly right skewed (qqPlot does not follow a linear trend) due to which we can state the CLT is not in effect as we have a small sample size of 15; if we increase the sample size and then calculate the sample means, the distribution of the sample mean will be normally distributed.Also the theoretical value and the calculate values of SD are same i.e 0.52.

(2.5) *(10 points)* * Create a sample histogram of the sample MEDIANS (make the bars red, make sure you label your axes and put on a clear title).
* Make a normal quantile plot of the sample medians using the `qqPlot()` function in the `car` package. Do the medians seem normally distributed? * Display summary statistics OF THE SAMPLE MEDIANS. Is the median of the sample medians the value you expect?
* Calculate and display the sample standard deviation of the sample medians and display rounded to two decimal places. Is this value similar to the sd of the sample means?

```
library(car)

hist(smedians,
     col = "red",
     main = "Hisogram of 10,000 sample medians, each of size 15",
     breaks = 50,
     xlab = "Medians",
     lwd = 2,
     cex.main=0.9)
```



```
# normal quantile plot

qqPlot(smedians,
       col = "red",
       pch = 19,
       main = paste("qqPlot : 10000 sample medians, each sample median of n =", N))
```



```
## [1] 3027 1896
```

```
# Summary Statistics of smedians
ans1 <- summary(smedians)
ans1
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2217  1.0753  1.3925  1.4522  1.7600  4.4195
```

```
ans <- round(ans1[3],2)
ans
```

```
## Median
##   1.39
```

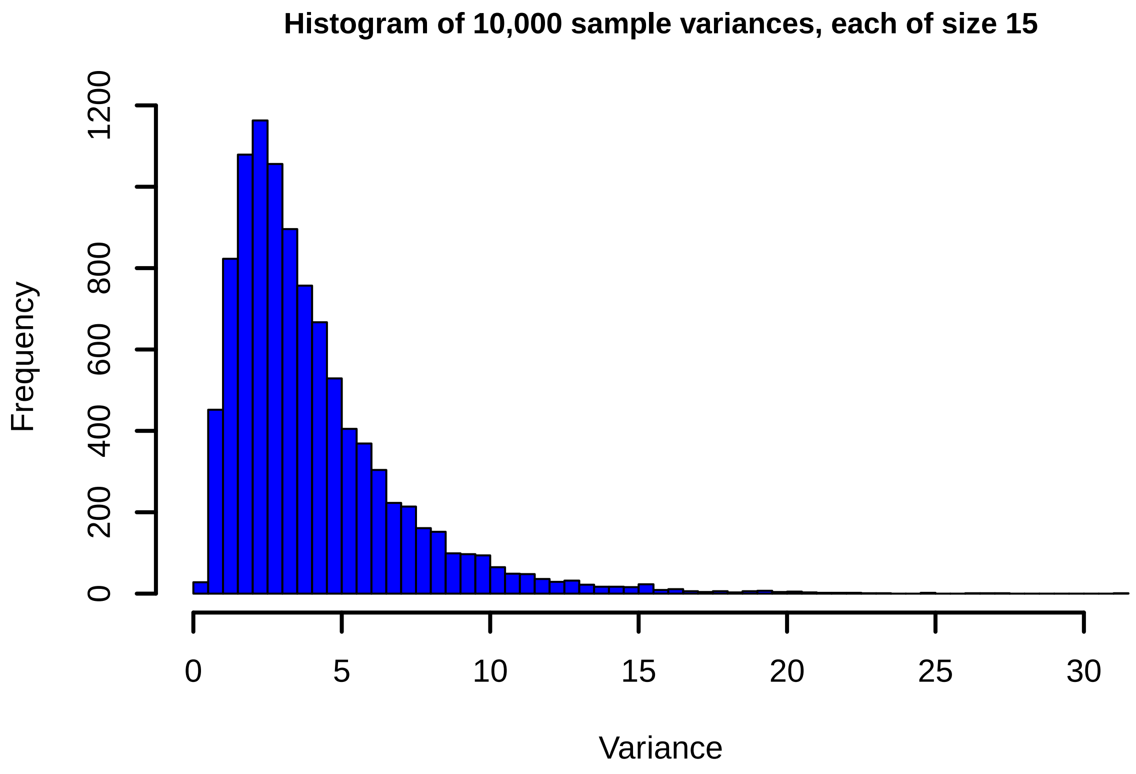
```
print(paste("Sample SD of the sample medians: ", round(sd(smedians), 2)))
```

```
## [1] "Sample SD of the sample medians:  0.52"
```

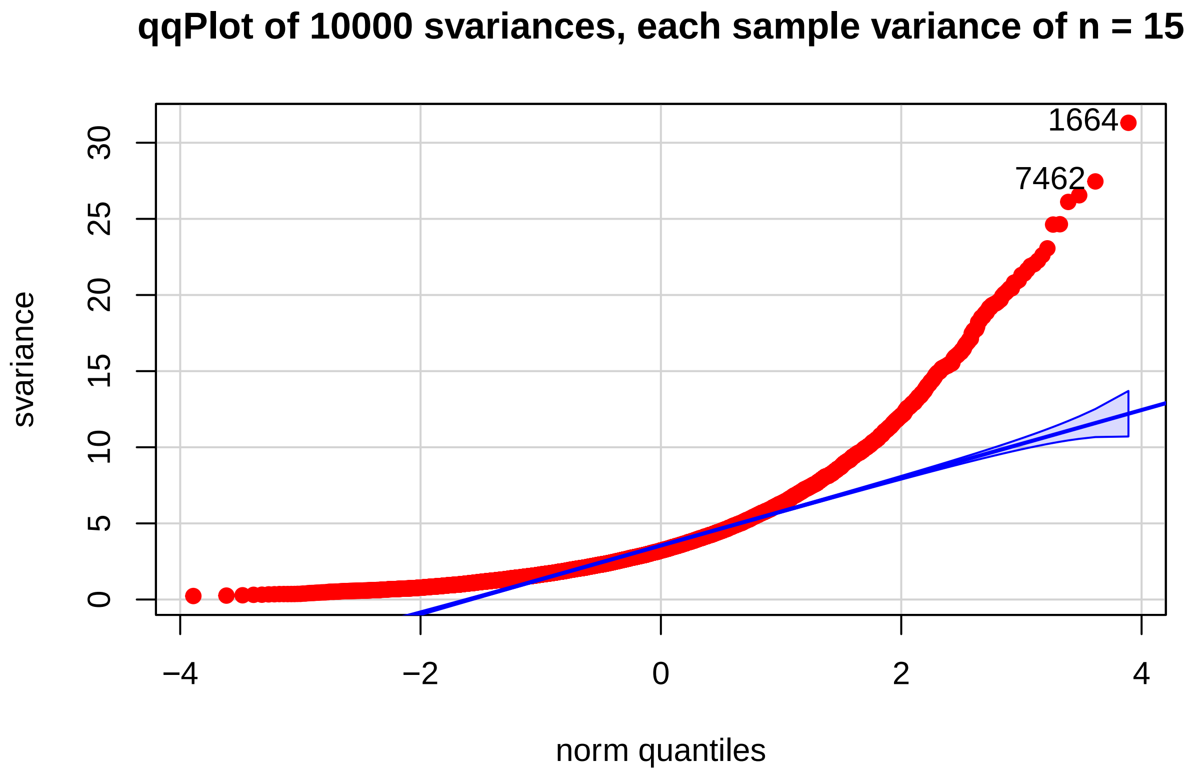
Based on the histogram and qqPlot it seems that the medians are not normally distributed, the distribution seems to be right skewed. Yes the median of the sample medians is similar to the value expected (2 X ln(2) = 1.39). The sample standard deviation of the sample medians is the value similar to the sd of the sample means i.e 0.52.

(2.6) *(10 points)* * Create a sample histogram of the sample VARIANCES (make the bars blue, make sure you label your axes and put on a clear title).
* Make a normal quantile plot of the sample VARIANCES using the `qqPlot()` function in the `car` package. Do the variances seem normally distributed? * Display summary statistics OF THE SAMPLE VARIANCES Is the mean of the sample variances the value you expect?
* Calculate and display the sample standard deviation of the sample variances. Just a note that without messy math, there's no easy way to know this number.

```
library(car)

hist(svariance,
     col = "blue",
     main = "Histogram of 10,000 sample variances, each of size 15",
     breaks = 50,
     xlab = "Variance",
     lwd = 2,
     cex.main=0.9)
```



```
qqPlot(svariance,
       col = "red",
       pch = 19,
       main = "qqPlot of 10000 svariances, each sample variance of n = 15")
```



```
## [1] 1664 7462
```

```
# Summary Statistics of svariance
ans1 <- summary(svariance)
ans1
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2294  2.0471  3.2168  4.0014  5.0481 31.3092
```

```
ans <- round(ans1[4],2)
ans
```

```
## Mean
##    4
```

```
print(paste("Sample SD of the sample variances: ", round(sqrt(var(svariance)), 2)))
```

```
## [1] "Sample SD of the sample variances:  2.92"
```

Based on the histogram and the qqPlot we can say that the sample variance are not normally distributed and are heavily right skewed. The qqPlot does not have a linear trend. The mean of the sample variances (4.04) is almost similar to the mean of the original distribution (mean = 4), which we expected.

(3) Cloud Seeding and the Bootstrap *(40 points, 5 points each part, part e) counts double.*

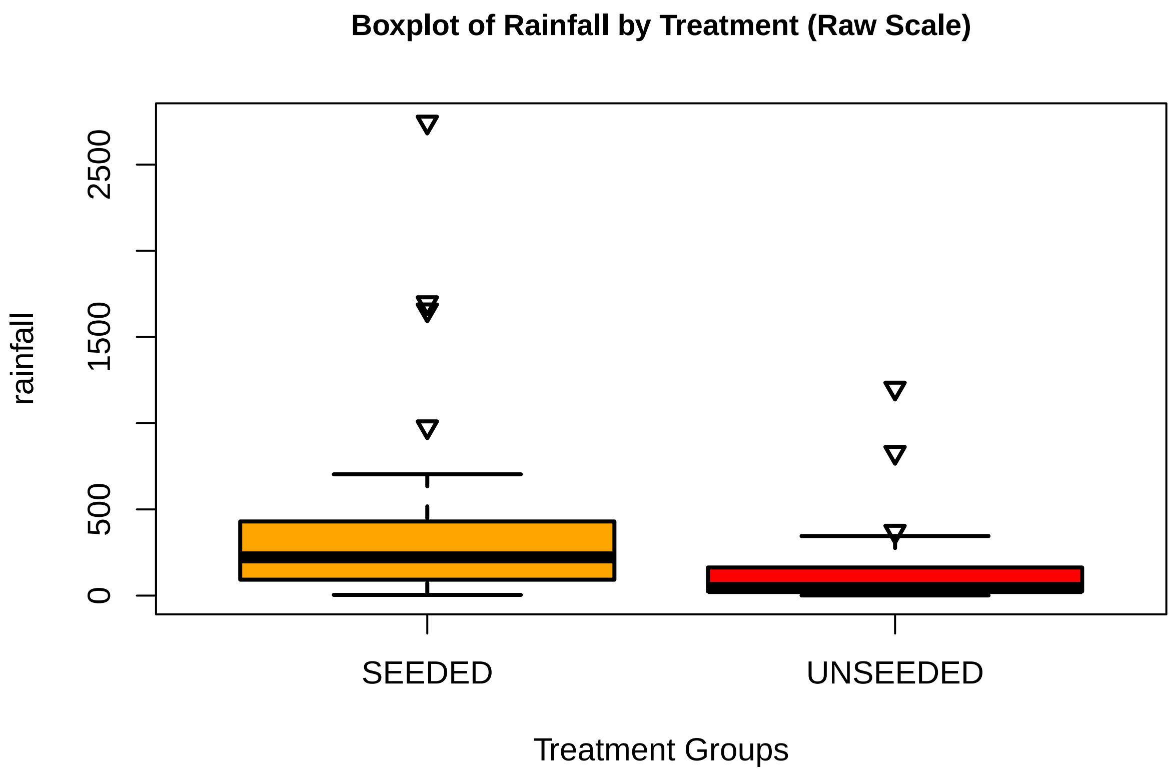
This problem examines results of a study of cloud seeding. The data is HERE. The variables are `rainfall` and `treatment` (SEEDED and UNSEEDED).

(3.1) Read the data into an object called `clouds`.

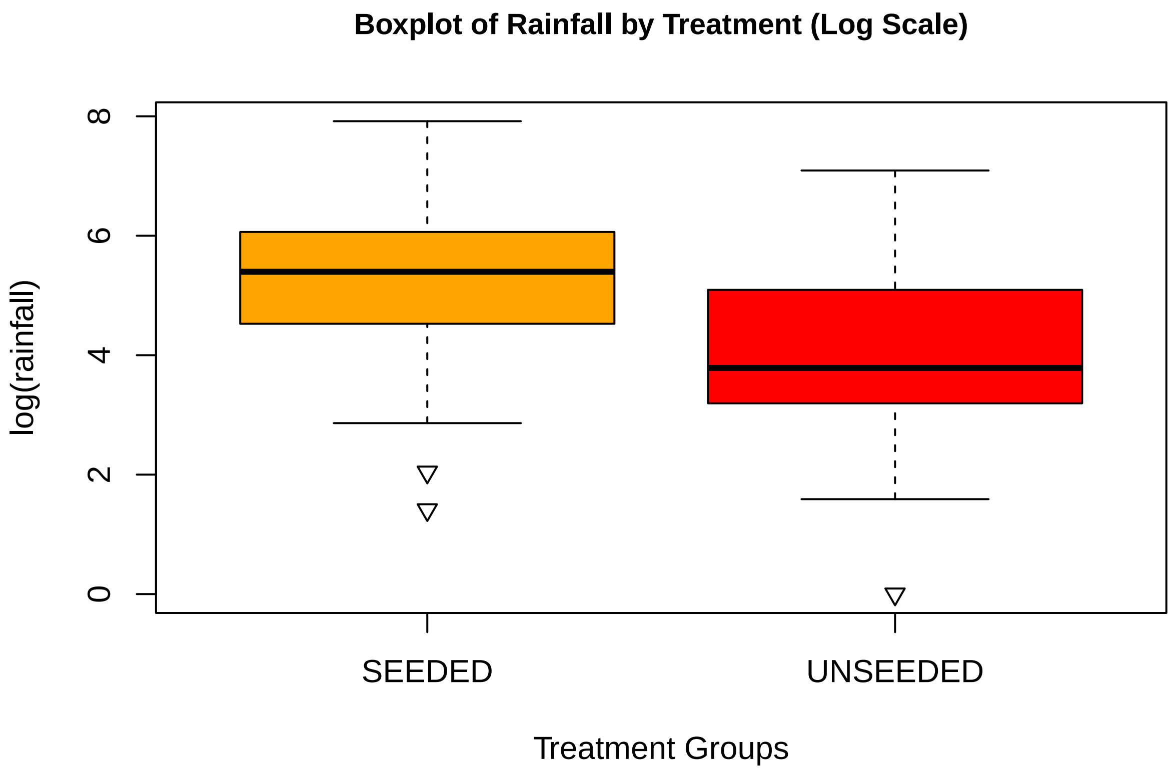
```
clouds <- read.csv("http://reuningscherer.net/S&DS230/data/rainandseedingclouds.csv", header = TRUE)
```

(3.2) Make side by side boxplots of rainfall by treatment. Make side by side boxplots of log(rainfall) by treatment. Write a sentence or two about what you observe. Which scale do you prefer?

```
boxplot(rainfall ~ treatment,
        data = clouds,
        pch = 19,
        outpch = 25,
        col = c("orange","red"),
        lwd = 2,
        xlab = "Treatment Groups",
        main = "Boxplot of Rainfall by Treatment (Raw Scale)",
        cex.main=0.9)
```



```
boxplot(log(rainfall) ~ treatment,
        data = clouds,
        pch = 19,
        outpch = 25,
        col = c("orange","red"),
        xlab = "Treatment Groups",
         main = "Boxplot of Rainfall by Treatment (Log Scale)",
        cex.main=0.9)
```



I would prefer a boxplot of log(rainfall) by treatment as it helps to easily understand the distribution between the two groups (for example; the mean of unseeded is less than the seeded. Also, the outliers are less on the log scale.

(3.3) Calculate summary statistics for rainfall by treatment on the raw scale and the log scale.

```
# Summary Statistics

print("summary statistics for rainfall by treatment on the Raw scale")
```

```
## [1] "summary statistics for rainfall by treatment on the Raw scale"
```

```
tapply(clouds$rainfall, clouds$treatment, summary)
```

```
## $SEEDED
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.10   98.12  221.60  441.98  406.02 2745.60
##
## $UNSEEDED
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   24.82   44.20  164.59  159.20 1202.60
```

```
print("summary statistics for rainfall by treatment on the Log scale")
```

```
## [1] "summary statistics for rainfall by treatment on the Log scale"
```

```
tapply(log(clouds$rainfall), clouds$treatment, summary)
```

```
## $SEEDED
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.411   4.581   5.396   5.134   6.001   7.918
##
## $UNSEEDED
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.211   3.786   3.990   5.069   7.092
```

(3.4) Calculate a two-sample t-test comparing mean log rainfall between treatments. Save results in an object called `test1` and display the results. Use alpha = .01 (i.e. make a 99% CI). Is there evidence of a difference between groups?

```
test1 <- t.test(log(rainfall) ~ treatment, data = clouds, conf.level = 0.99)
test1
```

```
##
##  Welch Two Sample t-test
##
## data:  log(rainfall) by treatment
## t = 2.5444, df = 49.966, p-value = 0.01408
## alternative hypothesis: true difference in means between group SEEDED and group UNSEEDED is not equal
## 99 percent confidence interval:
##  -0.06001102  2.34757335
## sample estimates:
##   mean in group SEEDED mean in group UNSEEDED
##               5.134187               3.990406
```

There is no significant difference in the two groups mean. The p value is more than 0.01 (the alpha) and therefore, we fail to reject the null hypothesis. Based on the confidence interval of (-0.6 , 2.37) we can see that 0 is included and therefore, at a 99% confidence level, we would say that there is not a significant difference in the means between the two groups.

(3.5) Get 10,000 boostrap samples from the data and compare the mean log rainfall between sample means. Save these means in an object called `diffRain`.

```
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)

# FILL IN REMAINING CODE
```

```
#Set number of bootstrap samples to take
N <- 10000

#Make empty vector for sample mean differences
diffRain <- rep(NA, N)

for (i in 1:N) {
  rfUS <- sample(log(clouds$rainfall)[clouds$treatment == "UNSEEDED"],
                 sum(clouds$treatment == "UNSEEDED"), replace = TRUE)
  rfS <- sample(log(clouds$rainfall)[clouds$treatment == "SEEDED"],
                 sum(clouds$treatment == "SEEDED"), replace = TRUE)
  diffRain[i] <- mean(rfS) - mean(rfUS)
}
```

(3.6) Calculate a 99% Bootstrap confidence interval. How to results compare to the theoretical interval in part d)?

```
# Calculating a 99% Bootstrap confidence interval
ci <- quantile(diffRain, c(0.005,0.995))
ci
```

```
##        0.5%       99.5%
## -0.04566067  2.22803856
```

```
# Report these values
round(ci,1)
```

```
##  0.5% 99.5%
##   0.0   2.2
```

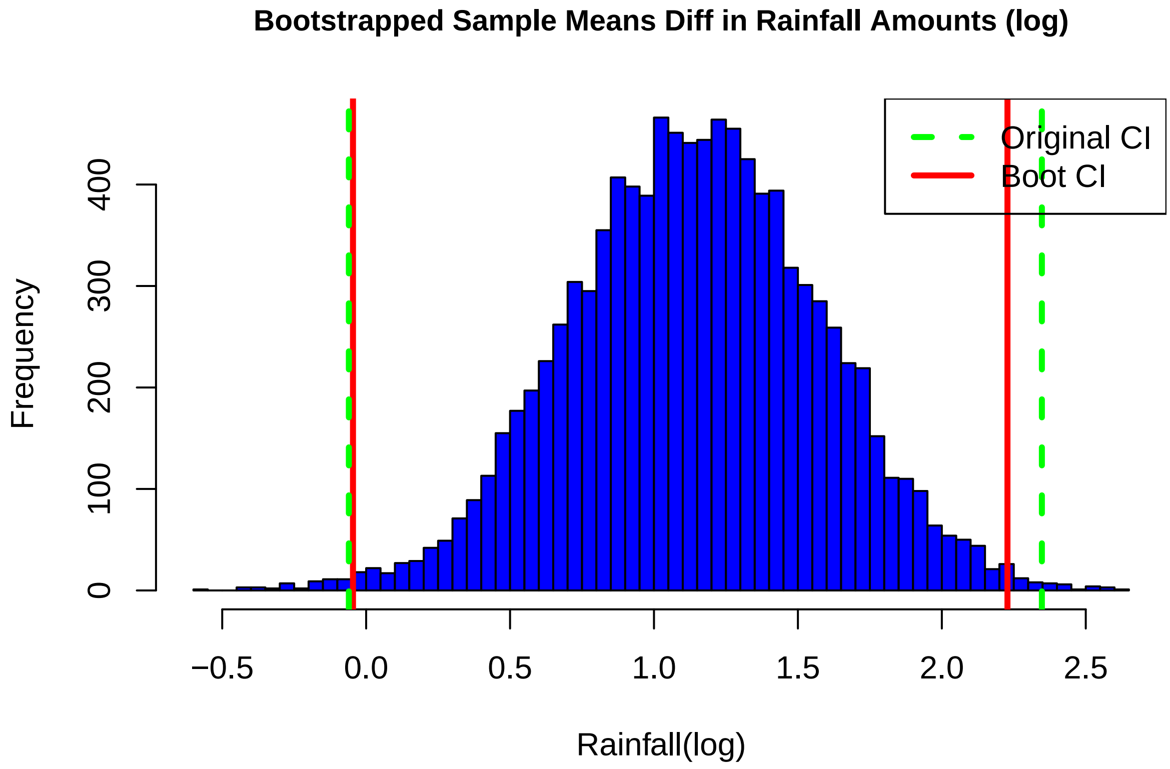
```
# compare to original 99% CI for the mean
Rainfall_test <- test1$conf.int
round(Rainfall_test,1)
```

```
## [1] -0.1  2.3
## attr(,"conf.level")
## [1] 0.99
```

The rounded theoretical interval in the part (d) is (-0.1, 2.3) and the rounded 99% Bootstrap confidence interval is (0.0, 2.2). The bootstrap CI is somewhat similar to the theoretical interval of two-sample t.test CI. However, we can say that bootstrap CI is a bit narrower than the than the theoretical CI.

(3.7) Make a histogram of bootstrap differences in means and add vertical lines for the theoretical and bootstrapped confidence intervals.

```
#Make histogram of bootstrap sample means
hist(diffRain,
     col = "blue",
     main = "Bootstrapped Sample Means Diff in Rainfall Amounts (log)",
     xlab = "Rainfall(log)",
     breaks = 50,
      cex.main=0.9)

#Add lines to histogram for CI's
abline(v = ci, lwd = 3, col = "red")
abline(v = Rainfall_test, lwd = 3, col = "green", lty = 2)
legend("topright",
       c("Original CI","Boot CI"),
       lwd = 3,
       col = c("green","red"),
       lty = c(2,1))
```


Bootstrapped Sample Means Diff in Rainfall Amounts (log)
Original CI
Boot CI
Frequency
Rainfall(log)