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

Due by 11:59pm, Saturday, February 17, 2024, 11:59pm

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)

```
#1.1
myFib = c(0,1)
for(x in 3:30){
  myFib[x] = myFib[x-1] + myFib[x-2]
}
myFib[30]
```

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

```
#1.2
wb = read.csv("http://www.reuningscherer.net/s&ds230/data/WB.2016.csv")
naVals = c()
for(i in 1:ncol(wb)){
  naVals[i] = sum(is.na(wb[,i]))
}
hist(naVals, main = "Histogram of Missing Values in World Bank Data", xlab = "Number of Missing Values"
```

(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.

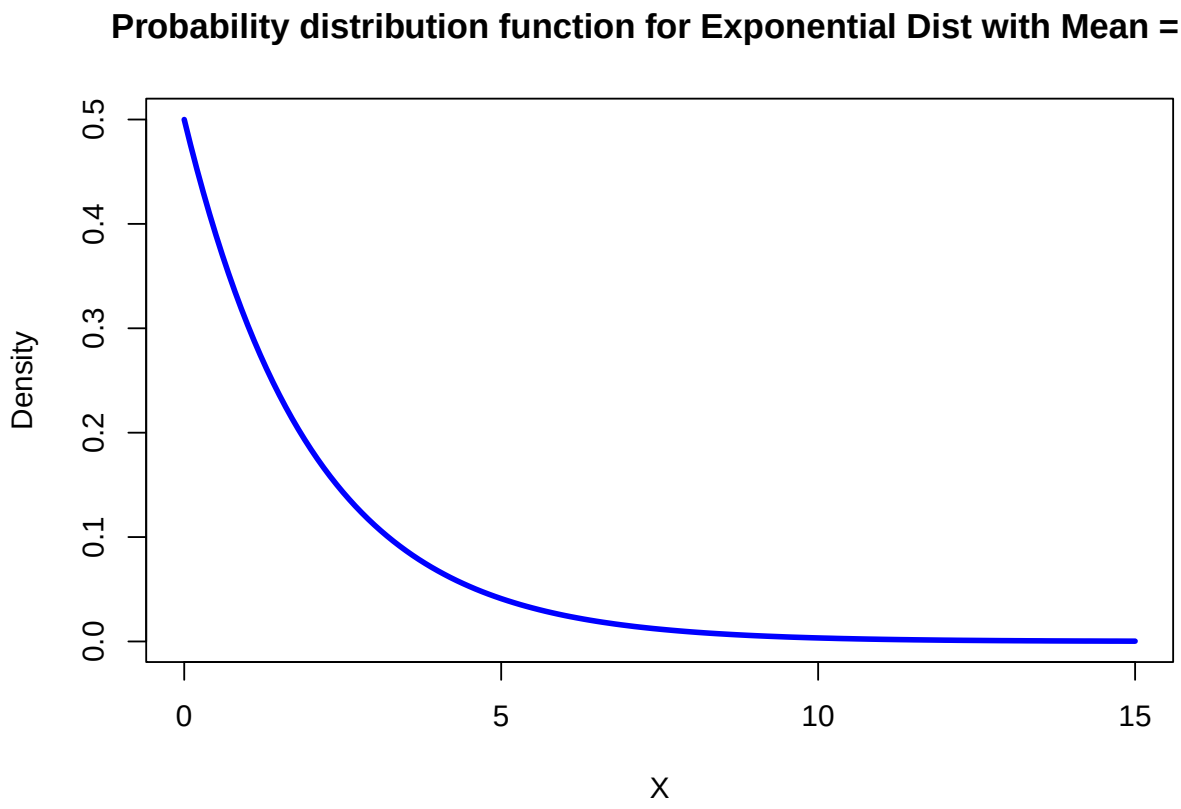
The code below gives a quick plot of this distribution. Your job is to succinctly answer what each part of the code does. You'll probably need to get help on the function `dexp()`, `seq()` and on a few of the graphics parameters in `par()`.

```
#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 returns values of an exponential density given in the first argument, and the rate in the second
#rate = .5 is the parameter that specifies 1/mean gives us mean of 2 (mean is 1/rate)
#by = .1 gives us the step size 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 = 'l' means give type line plot
#lwd = 3 means make the line width 3
```

(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)
N <- 15
TIMES <- 10000
samples <- matrix(rexp(N*TIMES, rate = 0.5), ncol = N)
dim(samples)
```

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

```
head(round(samples, 3),4)
```

```
##       [,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`.

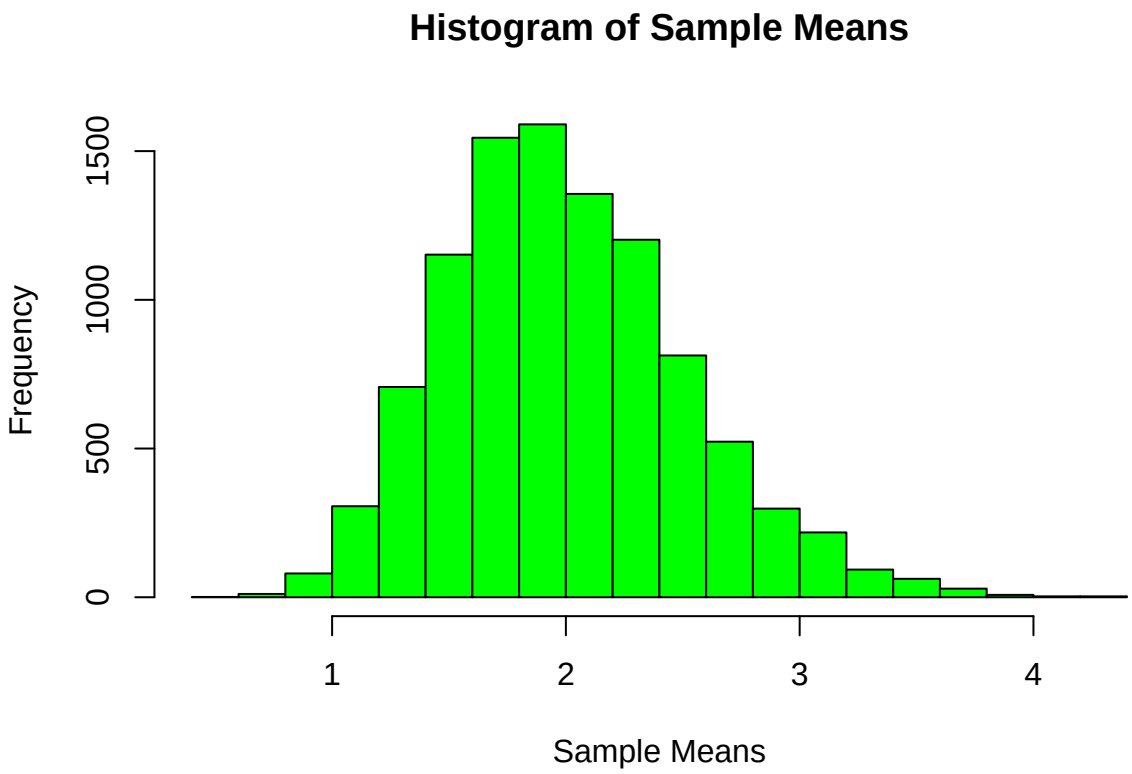
```
smeans <- apply(samples, 1, mean)
smedians <- apply(samples, 1, median)
svariance <- apply(samples, 1, var)
```

(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?

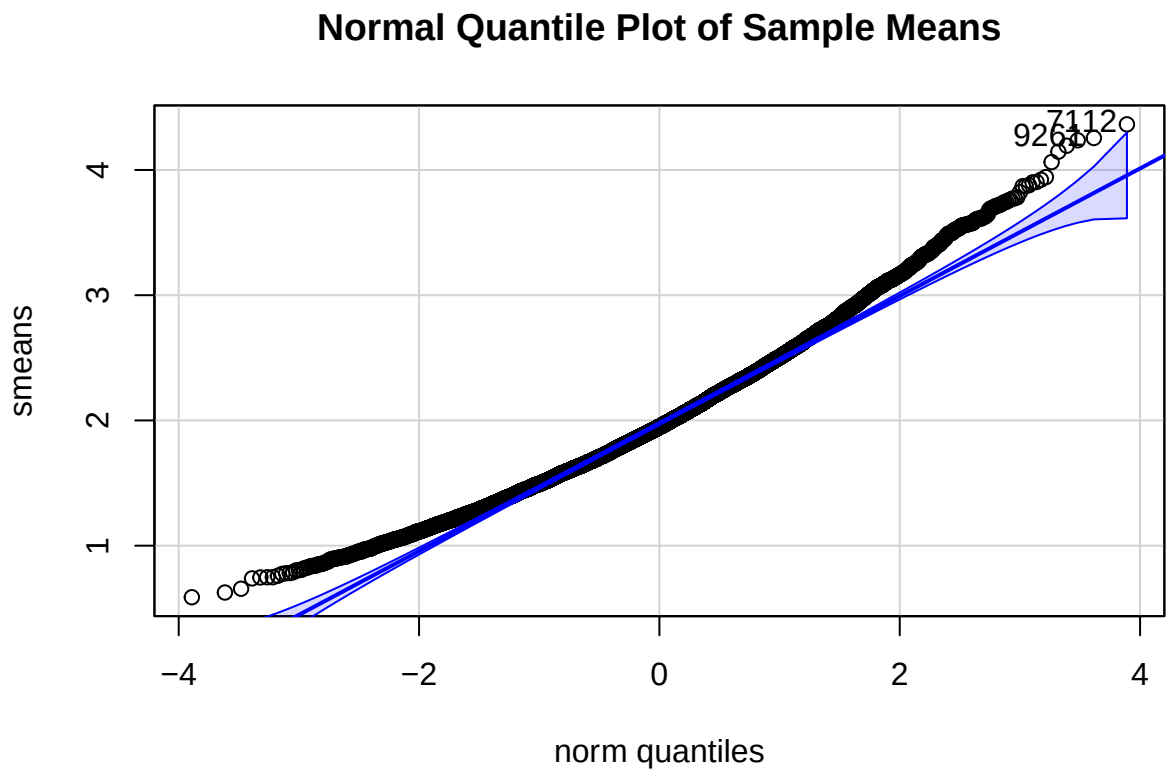
```
#2.4
library(car)
```

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

```
hist(smeans, col = "green", main = "Histogram of Sample Means", xlab = "Sample Means", ylab = "Frequenc
```




```
qqPlot(smeans, main = "Normal Quantile Plot of Sample Means")
```




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

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

```
## [1] 2
```

```
round(sd(smeans), 2)
```

```
## [1] 0.52
```

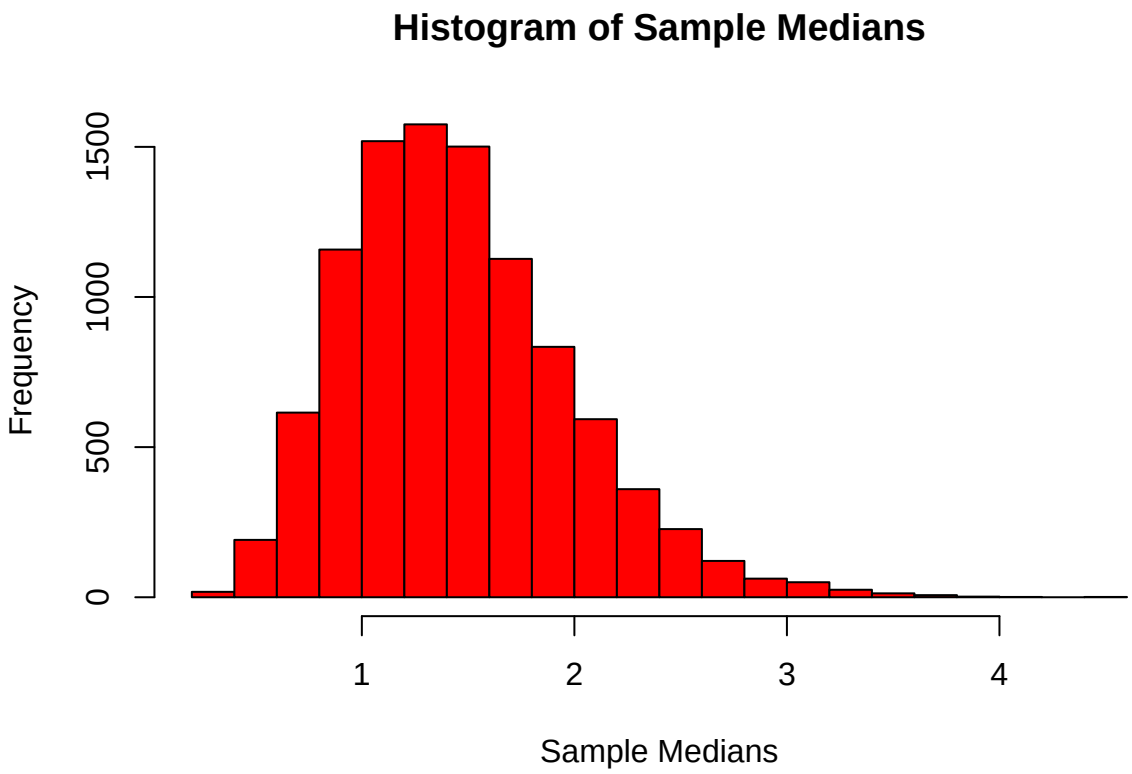
```
round(2/sqrt(15), 2)
```

```
## [1] 0.52
```

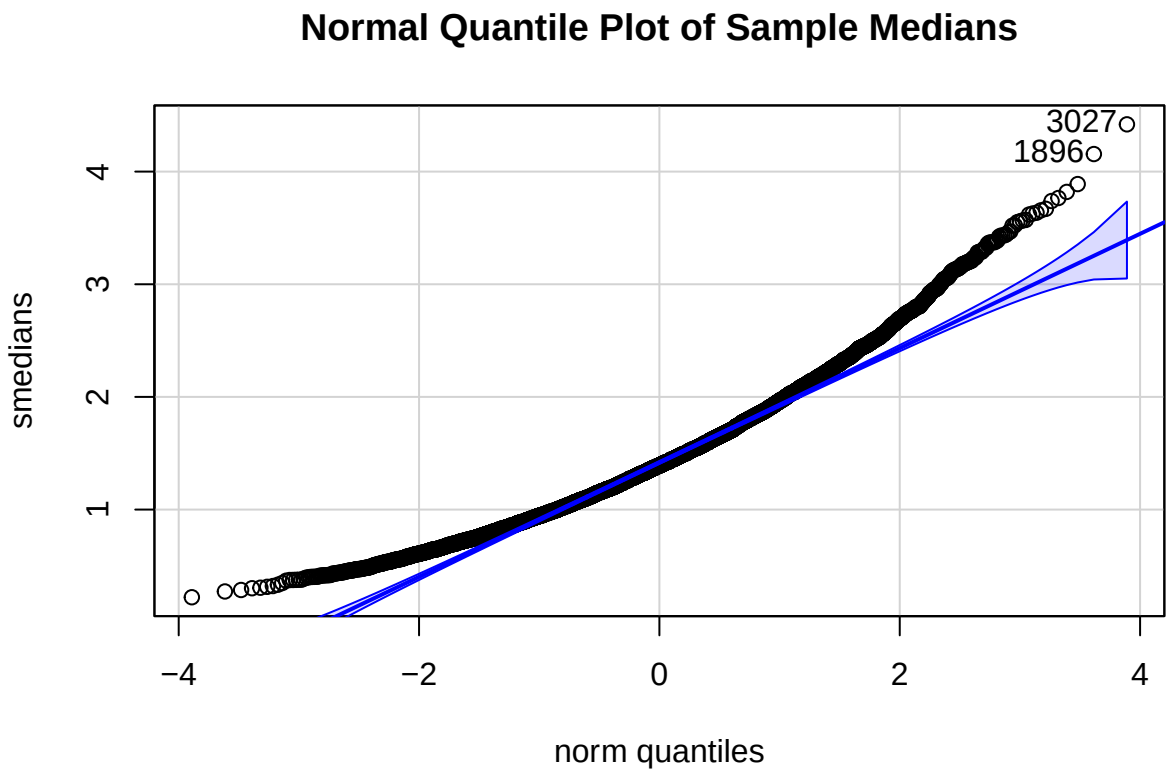
As the quantile plot shows a more normally distributed, and less skewed plot, we can infer that from adding more data, we approach a normal distribution, thus the CLT is in effect By construction, we would expect a mean of 2, which is also what is oberved The sample standard deviation is 0.52, and the expected value is 0.52, which are similar

(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?

```
#2.5
hist(smedians, col = "red", main = "Histogram of Sample Medians", xlab = "Sample Medians", ylab = "Freq
```




```
qqPlot(smedians, main = "Normal Quantile Plot of Sample Medians")
```




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

```
summary(smedians)
```

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

```
round(sd(smedians), 2)
```

```
## [1] 0.52
```

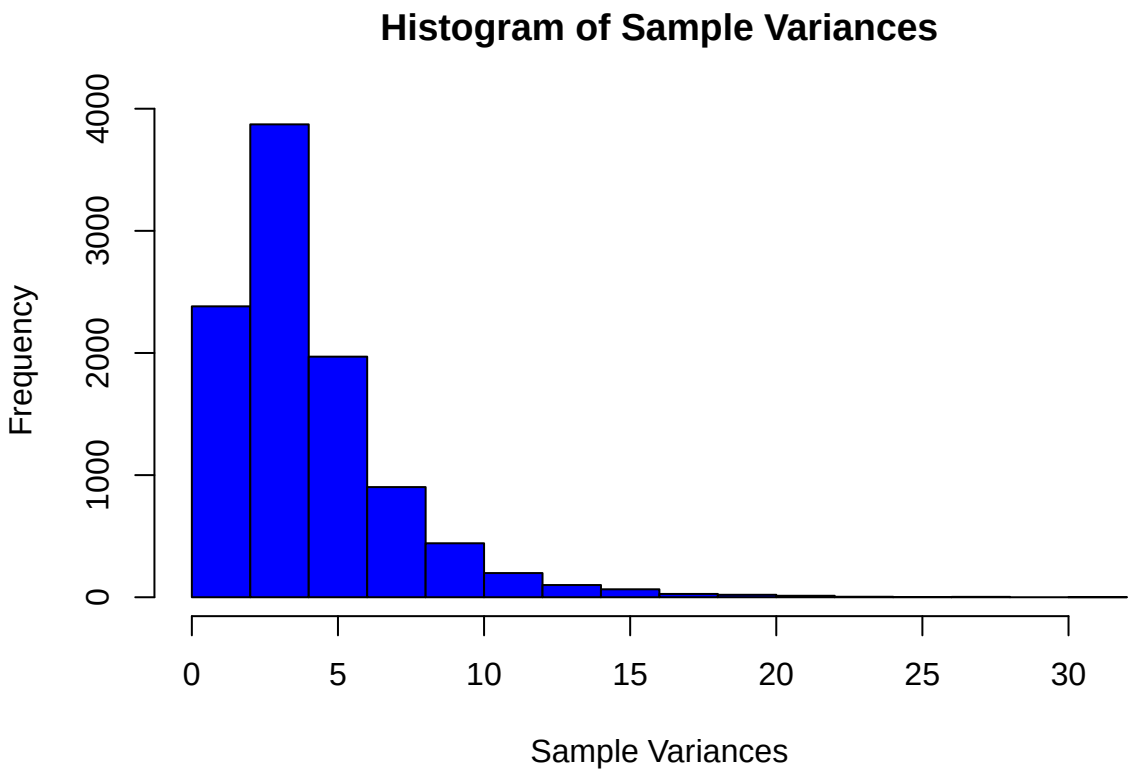
```
round(sd(smeans), 2)
```

```
## [1] 0.52
```

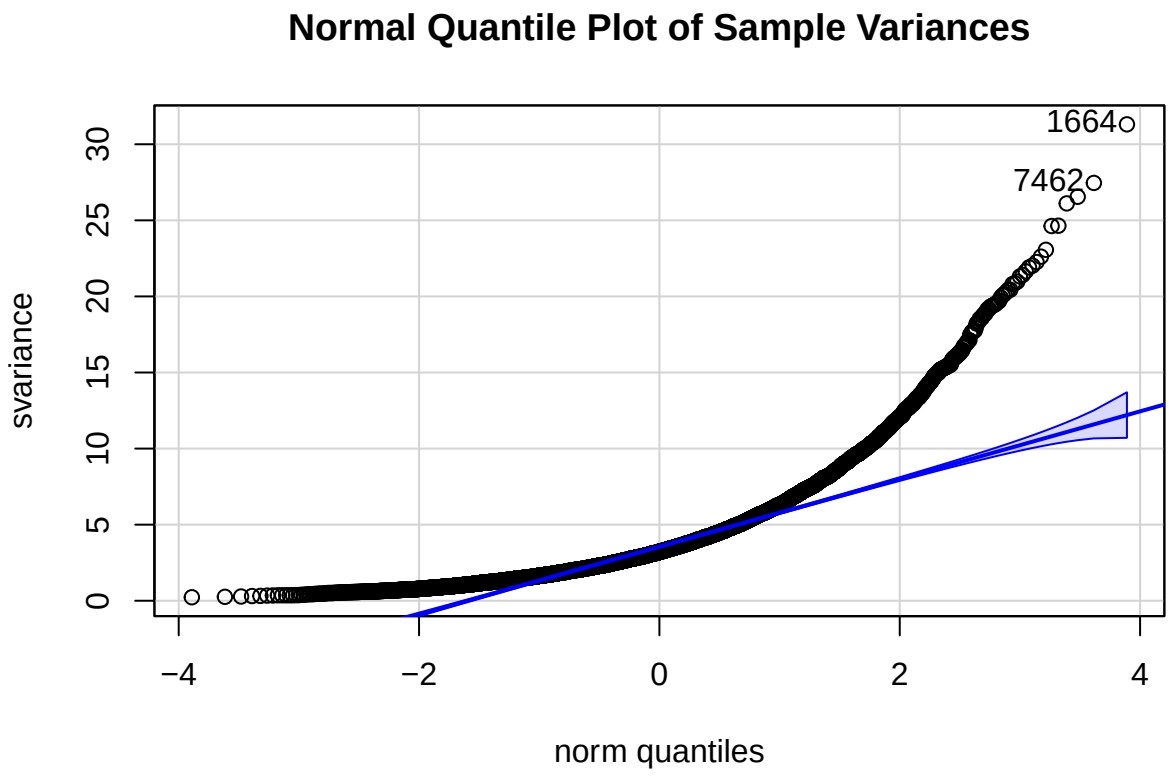
Looking at the normal quantiule plot, the points curve, indicating a right skew, which is also seen in the histogram. The median of the sample medians is 1.3985, which is very close to the expected value of 1.3863, or 2ln(2). The sample standard deviation of the medians is 0.52, which is similar to the standard deviation of the means, which is 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.

```
#2.6
hist(svariance, col = "blue", main = "Histogram of Sample Variances", xlab = "Sample Variances", ylab =
```




```
qqPlot(svariance, main = "Normal Quantile Plot of Sample Variances")
```




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

```
round(summary(svariance), 2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.23    2.05    3.22    4.00    5.05   31.31
```

```
round(sd(svariance), 2)
```

```
## [1] 2.92
```

The variances do not seem to be normally distributed. The observed mean of the sample variances is 4, which is the same as the expected 2^2 = 4.

(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`.

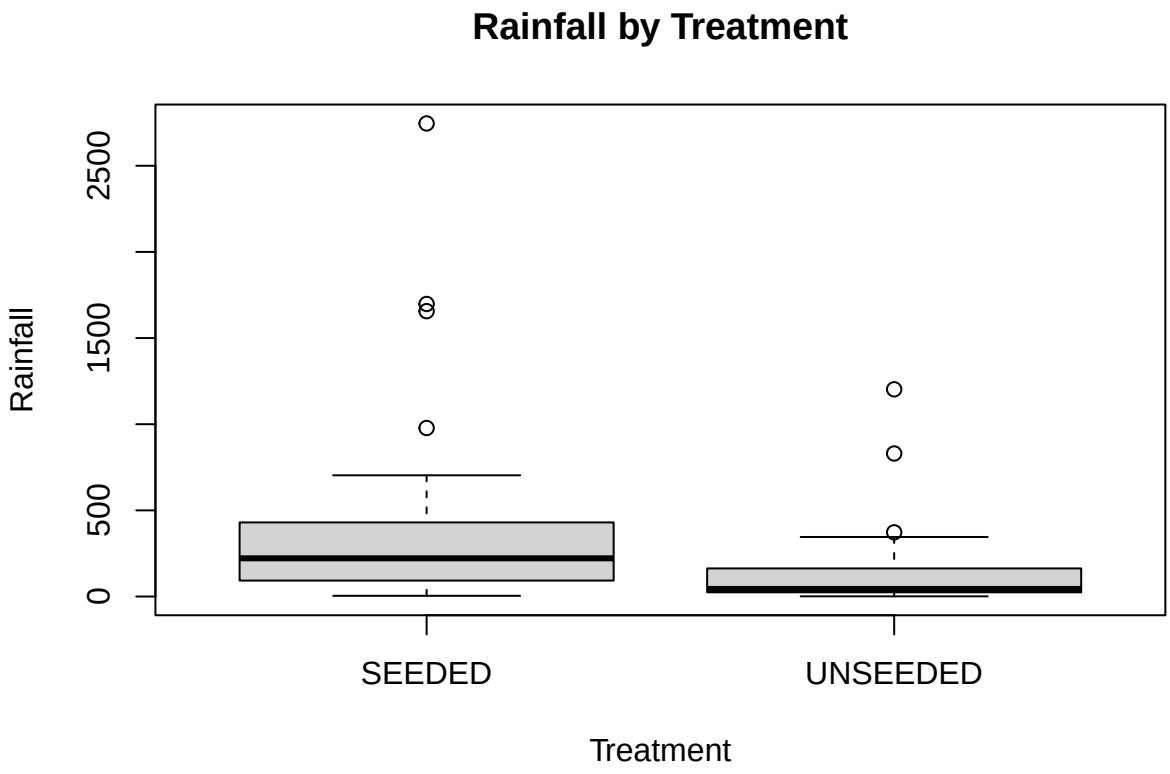
```
#3.1
clouds = read.csv("http://reuningscherer.net/S&DS230/data/rainandseedingclouds.csv")
head(clouds)
```

```
##   rainfall treatment
## 1   1202.6  UNSEEDED
## 2    830.1  UNSEEDED
## 3    372.4  UNSEEDED
## 4    345.5  UNSEEDED
## 5    321.2  UNSEEDED
```

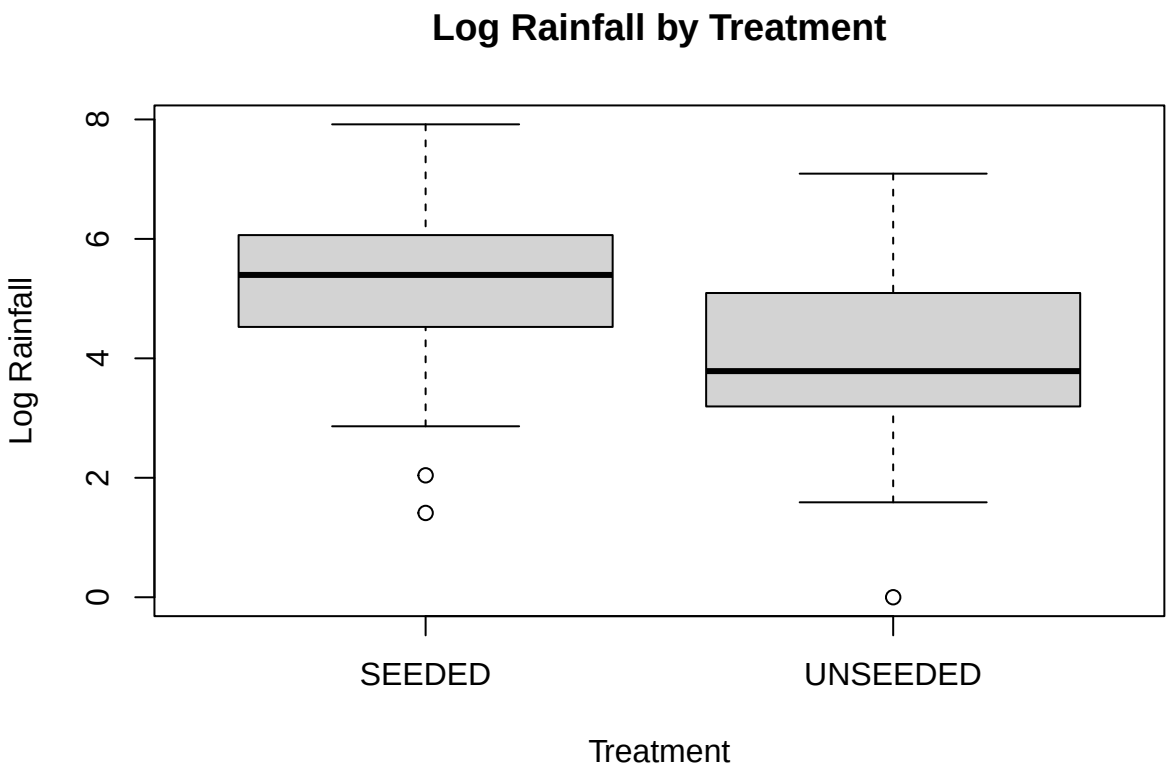
```
## 6    244.3  UNSEEDED
```

(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, main = "Rainfall by Treatment", xlab = "Treatment", ylab =
```




```
boxplot (log(rainfall) ~ treatment, data = clouds, main = "Log Rainfall by Treatment", xlab = "Treatmen
```



We observe in both graphs that the seeded treatment had a there was a higher average rainfall. The log scale is prefered, as it amplifies the small differences in data

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

```
#3.3
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
```

```
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?

```r
#3.4
test1 <- t.test(log(rainfall) ~ treatment, data = clouds, conf.level = .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
```

Since the p value is 0.01408 is greater than 0.01, we fail to reject the null hypothesis, where there is no statistically signifigant difference at the %99 confidence interval

(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`.

```r
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)
attach(clouds)
N <- 10000
diffRain <- rep(NA, N)
for (i in 1:N){
sSeed <- sample(rainfall[treatment == "SEEDED"],
                  sum(treatment == "SEEDED"), replace = TRUE)
  sUnseed <- sample(rainfall[treatment == "UNSEEDED"],
                  sum(treatment == "UNSEEDED"), replace = TRUE)
  diffRain[i] <- mean(log(sSeed)) - mean(log(sUnseed))
}
```

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

```r
#3.6
round(quantile(diffRain, c(.01, .99)),2)
```

```
##   1%  99%
## 0.14 2.20
```

The confidence interval is 0.14, 2.2 , which is different from the -0.06001102, 2.34757335 in test1. These intervals are very similar

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

```r
hist(diffRain, main = "Samples Means Diff in Log(Rainfall)",
     xlab = "Difference in Means", ylab = "Frequency", col = "lightblue")

legend("topright",
       c("Original CI","Boot CI"),
       lwd = 3,
       col = c("red","blue"),
       lty = c(2,1),
```

```
       cex = 0.6)

abline(v = test1$conf.int, col = "red")
abline(v = quantile(diffRain, c(0.005, 0.995)), col = "blue")
```