Project 1

Chris Harris / Stephan Kostreski

March 12, 2017

Problem 1 - Warm Up

Part A

Generate a sample of 1000 from the normal distribution with a mean of 10 and a standard deviation of 2.25.

Hint: Type ?rnorm into the console to see how to use this function to solve this part of the problem. Also note the arguments required:

- n: number of observations (set to 1000)
- mean: the mean value of the distribution (set to 10)
- standard deviation: the deviation of values from the mean (set to 2.25)

NOTE: Don't forget to assign the function result to a variable (using the <- operator) so you can refer to it later, like so:

data_dist <- rnorm(n=10, mean=1, sd=2)

```
#Generates a list of 1000 values from a normally distributed function with a mean of 10 and a standard
data_1000 <- rnorm(n=1000, mean=10, sd=2.25)
View(data_1000)
```

Part B

Create a histogram of the data with 10 bins (classes)

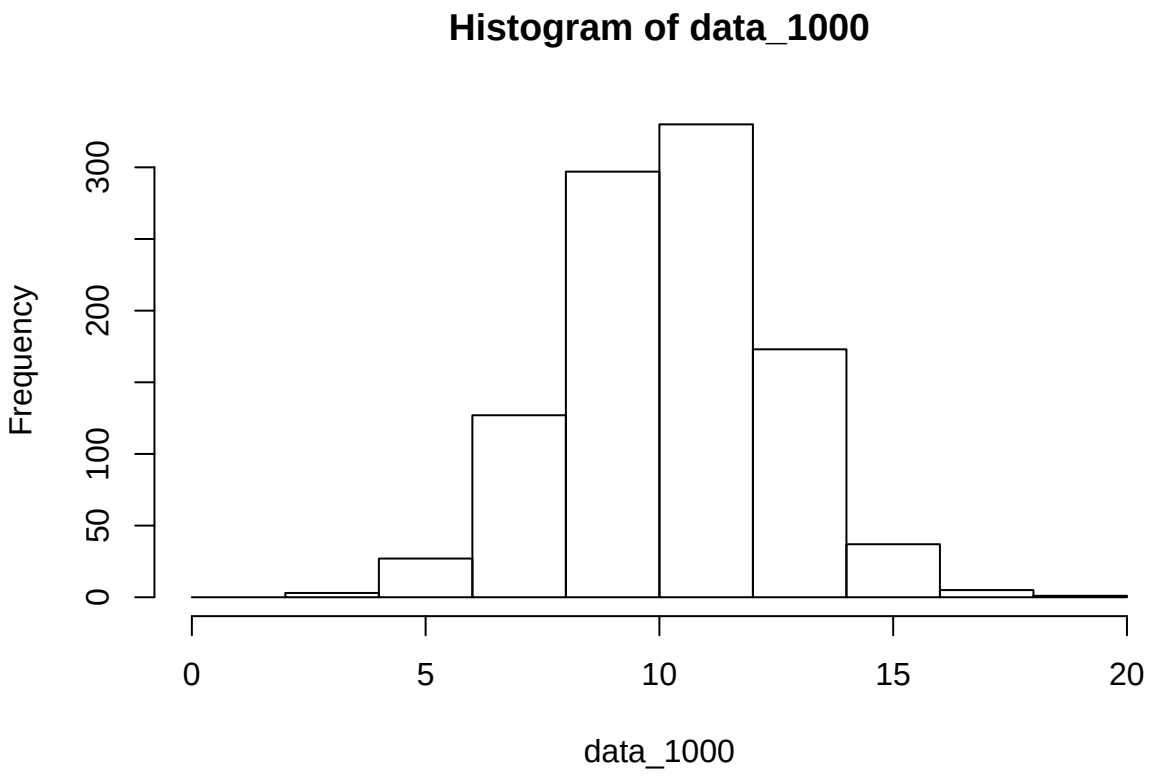
Functions/Arguments needed:

- hist
  - x: data to be tallied (i.e. the variable where you stored the rnorm function call)
  - xlab: label for the x-axis
  - breaks: number of bins (classes) to categorize data

Note: R may not create exactly 10 bins based on its algorithm. To gain precision you may provide exact breaks by using breaks= seq(0,20,by=2). This is what we did:

hist(x=data_dist, xlab="Sample Values", breaks=seq(0,20, by=2))

```
#Generates a histogram with 10 bins based upon our distribution
hist(x=data_1000, breaks=seq(0,20, by=2))
```



At this point, we recommend playing with different parameters to obtain colors, different titles and more. Hint: use ?hist to learn the parameters and/or google

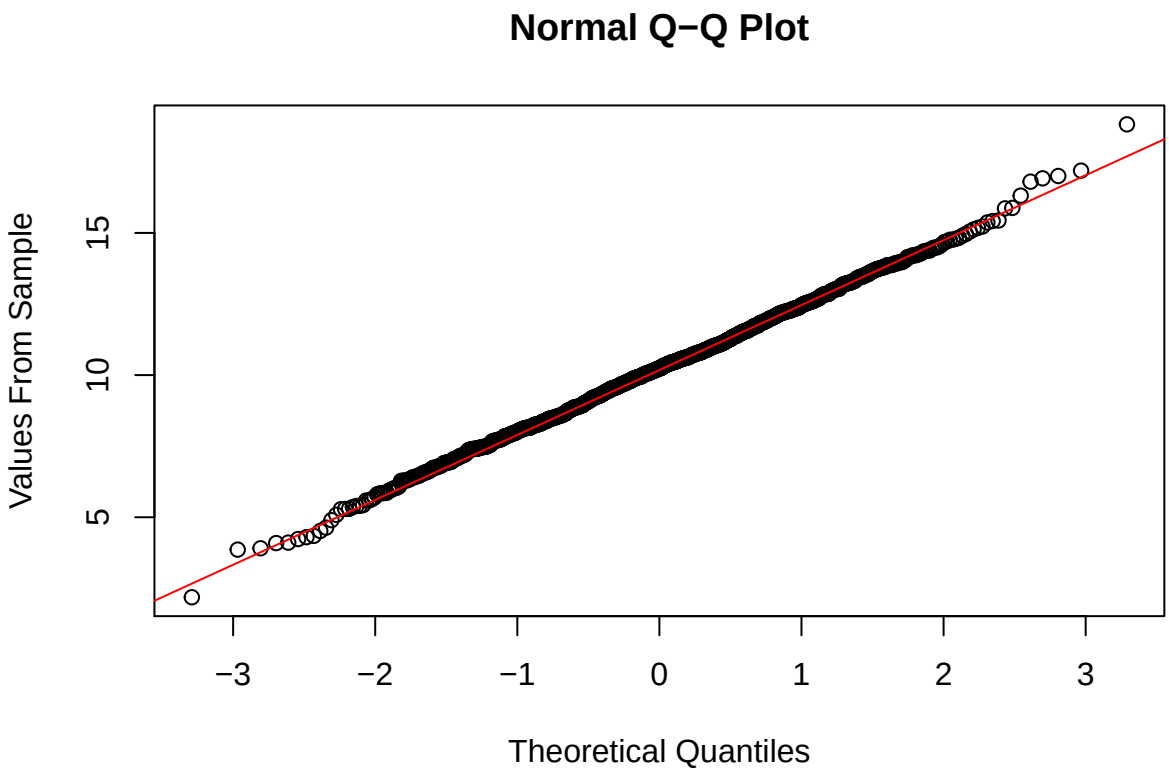
Part C

Create a normal probability plot and boxplot.

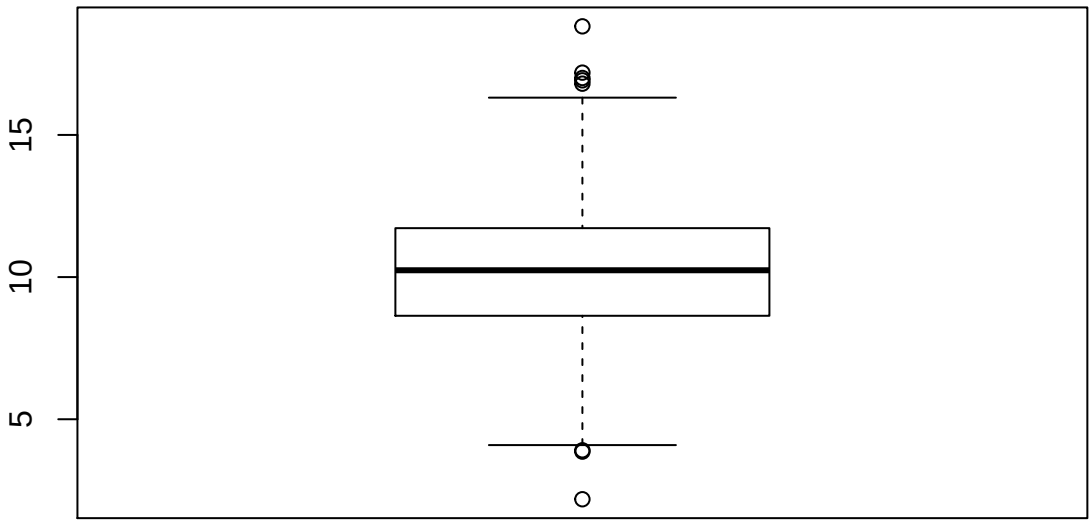
Functions/Arguments needed:

- boxplot - creates a boxplot
  - x: data to be plotted
- qqnorm - creates normal probability plot
  - y: data to be plotted
- qqline - adds a reference line to the qqnorm plot
  - y: data to add reference line
  - col: line color - col=2 creates a red line, col=3 creates a green line

```
#Create a normal probability plot
qqnorm(y=data_1000, ylab="Values From Sample")
qqline(y=data_1000, col=2)
```




```
#Create a boxplot
boxplot(data_1000)
```


15
10
5

Problem 2

Part A

Create 1000 samples of size 10.

Functions/Arguments needed:

- rnorm - same parameters as before
- replicate
  - n: number of repititions
  - expr: function to repeat 'n' times
- sample - takes a sample of a particular size from a "population distribution"
  - x: distribution to be sampled from (variable result of rnorm)
  - size: size of sample
  - replace: FALSE (sample without replacement)

```
#Create our overall population to be sampled
population_dist_norm <- rnorm(1000, mean=10, sd=2.25)


#Use the sample function to take a random sample with specified sample parameter. Repeat this process u
samples <- replicate(1000, sample(x=population_dist_norm, size=10, replace=FALSE))
```

Part B

Calculate the mean for these 1000 and graph the sampling distribution of the sample mean for these samples of size 10.

There are several ways to accomplish this, but we recommend using a looping construct to take the average of each sample and store each average in a list.

Using this method, you should wind up having a list of size 1000. Each element containing the average of that particular sample

Logical Constructs needed:

- For Loop

Functions/Arguments needed:

- c : creates a list with a specified argument range
- mean
  - x: list (function will take the average of this list)
- hist

As this can be tricky for those new to programming, here is an example of how to accomplish this task:

1. We define our distribution population as so:

distribution <- rnorm(n=100, mean=10, sd=2.25)

2. We take 10 random samples of size 10 from this distribution

samples <- replicate(n=10, sample(x=distribution, size=10, replace=FALSE))

3. We create a new list that will eventually hold our sample averages:

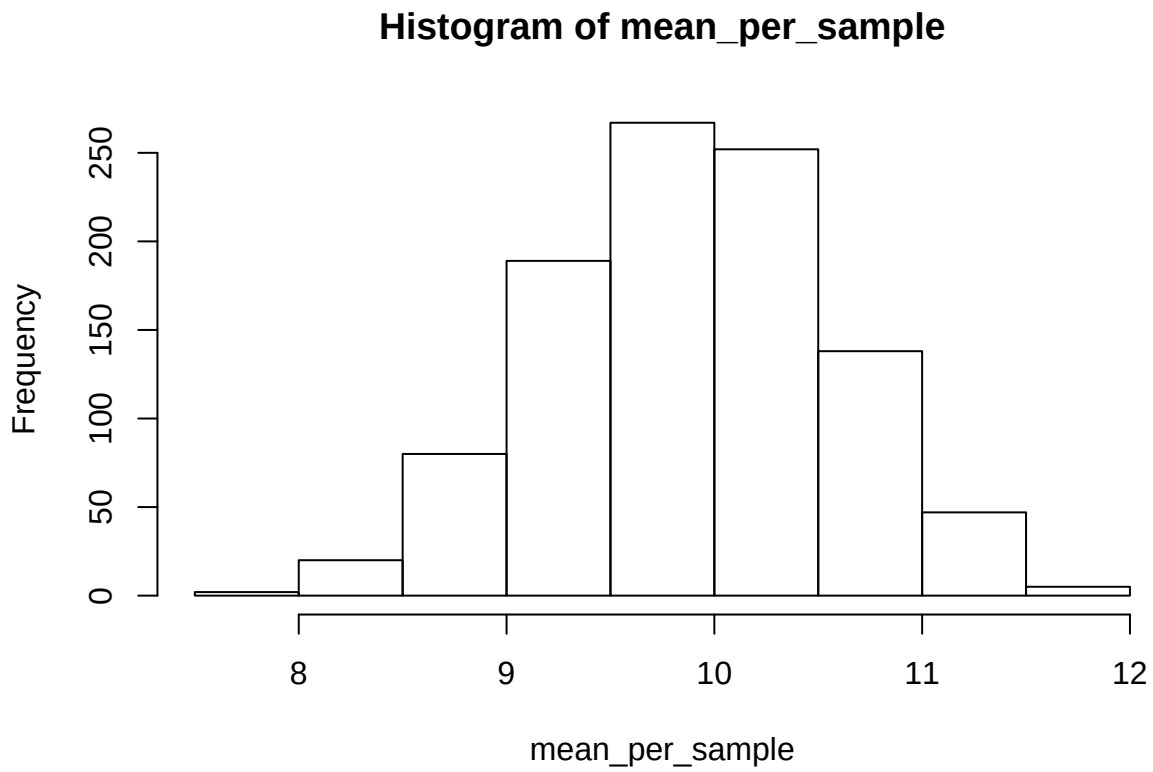
mean_per_sample <- c(1:10) - Creates a list of size 10

4. We then go through each of our individual samples, get the mean of the 10 values in each sample, and store that result in our list.

for (index in 1:10) { mean_per_sample[index] <- mean(samples[,index]) }

```
#Create a vector (a list) with 1000 elements with default values
mean_per_sample <- c(1:1000)
for (index in 1:1000) {
  mean_per_sample[index] <- mean(samples[,index])
}

hist(x=mean_per_sample, breaks=10)
```

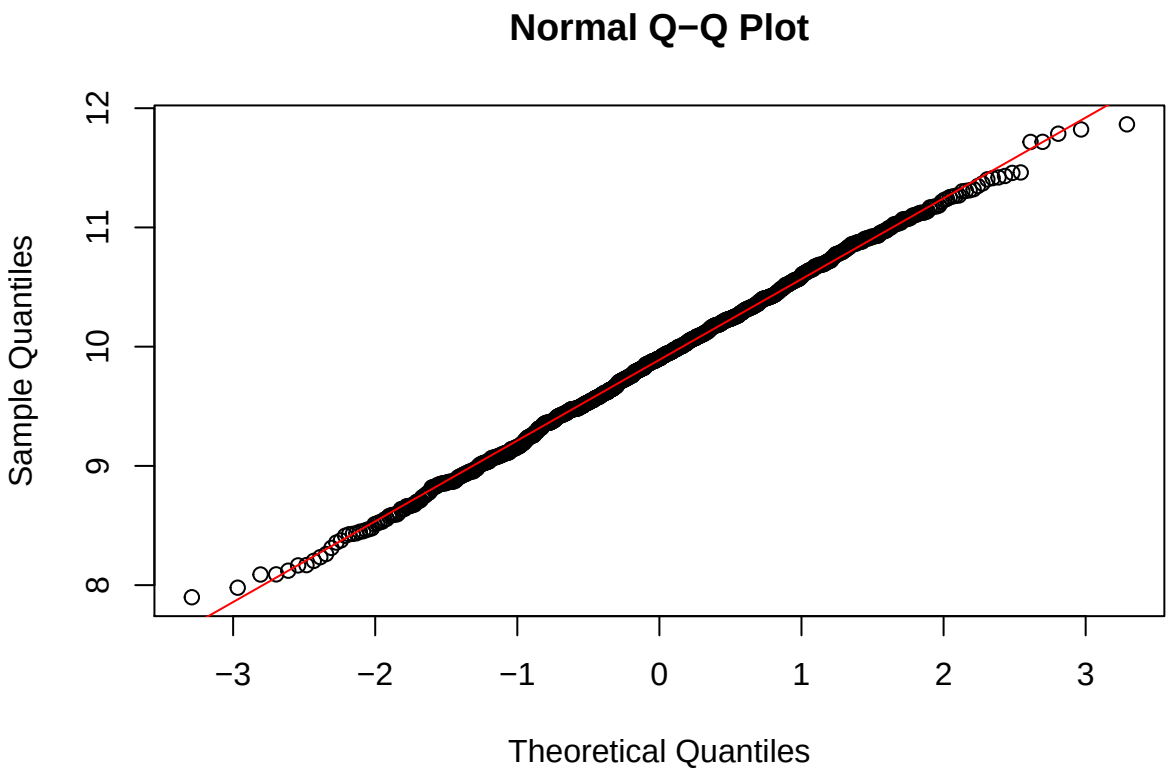


Part C

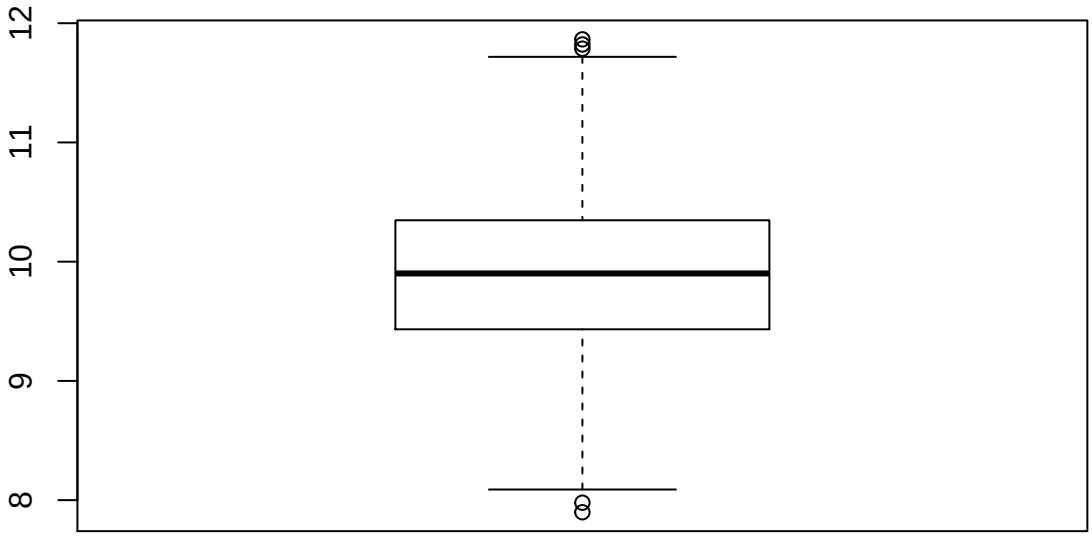
Create a normal probability plot and boxplot of this sampling distribution.

```
#Same as Problem 1c

qqnorm(y=mean_per_sample)
qqline(y=mean_per_sample, col=2)
```




```
boxplot(x=mean_per_sample)
```



Part D: Create the 90, 95 and 99 % confidence intervals.

1. Get the mean, standard deviation, and number of samples taken

Functions needed:

- mean
- sd
- length

```
#In order to calculate CI, we need the mean and standard deviation of our list of sample means.
a <- mean(mean_per_sample)
s <- sd(mean_per_sample)

#We also need the number of samples we took
n <- length(mean_per_sample)
```

2. Calculate 90% CONFIDENCE INTERVAL:

We are going to use the standard CI forumla.

Functions needed:

- qnorm - gives us the z-score given a probility
  - p: probability

Example:

qnorm(0.95) = 1.644854 qnorm(0.975) = 1.959964

```
error <- qnorm(0.95)*(s/sqrt(n)) # 90%
left <- a-error
right <- a+error
```

90% CI: [9.8531994, 9.9246034]

3. Calculate 95% CONFIDENCE INTERVAL:

```
error <- qnorm(0.975)*(s/sqrt(n))
left <- a-error
right <- a+error
```

95% CI: [9.8463598, 9.931443]

4. Calculate 99% CONFIDENCE INTERVAL:

```
error <- qnorm(0.995)*(s/sqrt(n))
left <- a-error
right <- a+error
```

99% CI: [9.8329923, 9.9448105]

Problem 3

Part A

Generate a sample of size 1000 from the exponential with mean 2

Functions/Arguments needed:

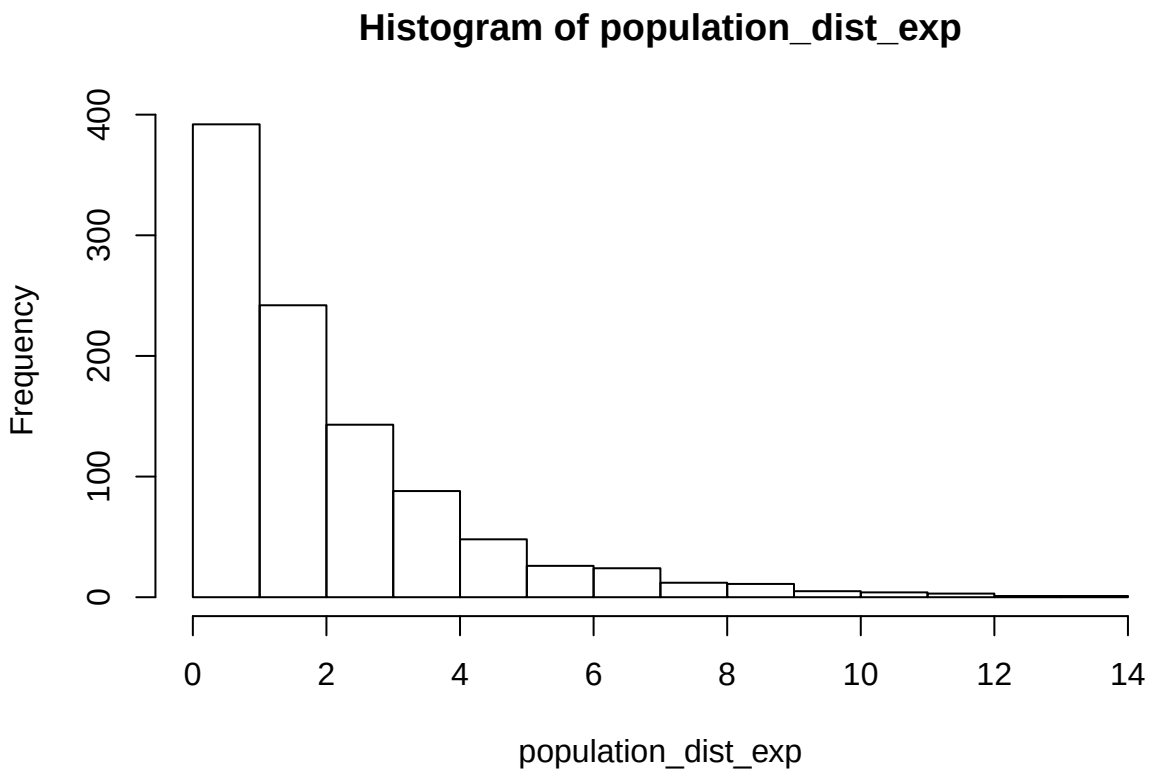
- rexp : randomly generates values from the exponential distribution
  - n: number of values to be generated
  - rate: rate of exponential growth (mean = 1/rate)

```
#Generate population for exponential distribution with mean '2'. Mean = 1/rate. Therefore, rate = 0.5.
population_dist_exp <- rexp(n=1000, rate=0.5)
```

Part B

Create a histogram of the data with 10 bins(classes)

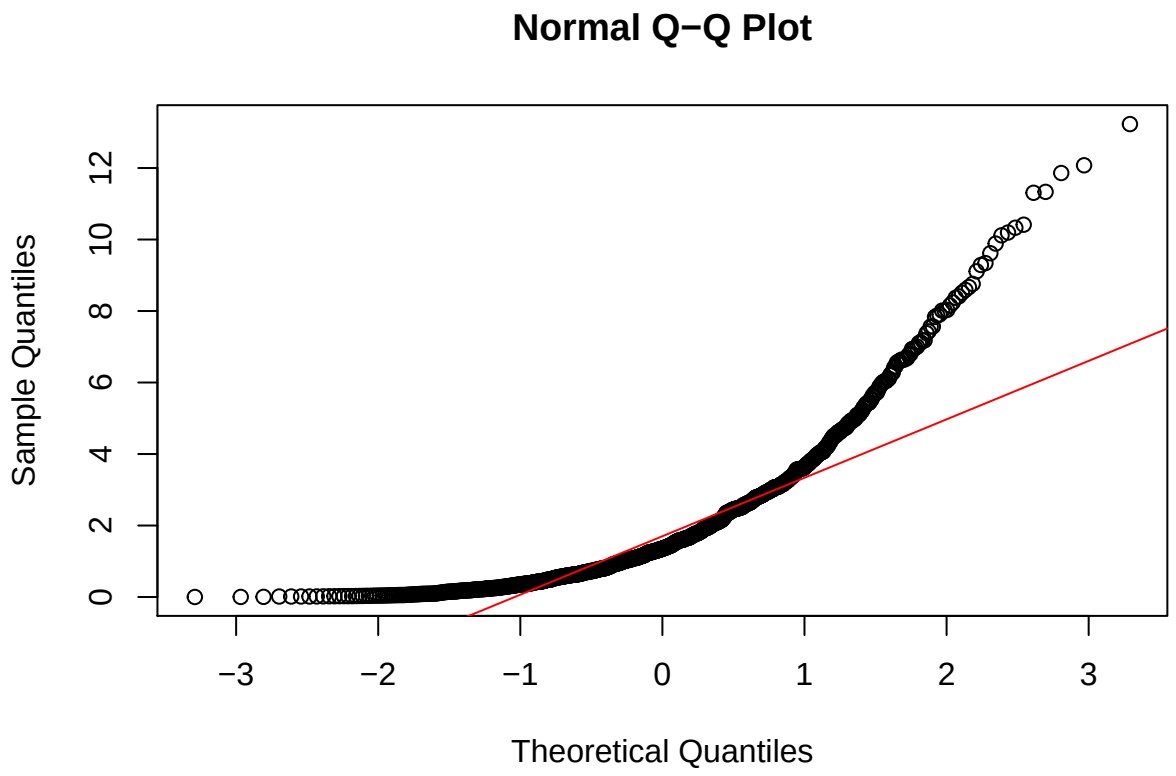
```
#Same as Problem 1B
hist(x=population_dist_exp, breaks=10)
```



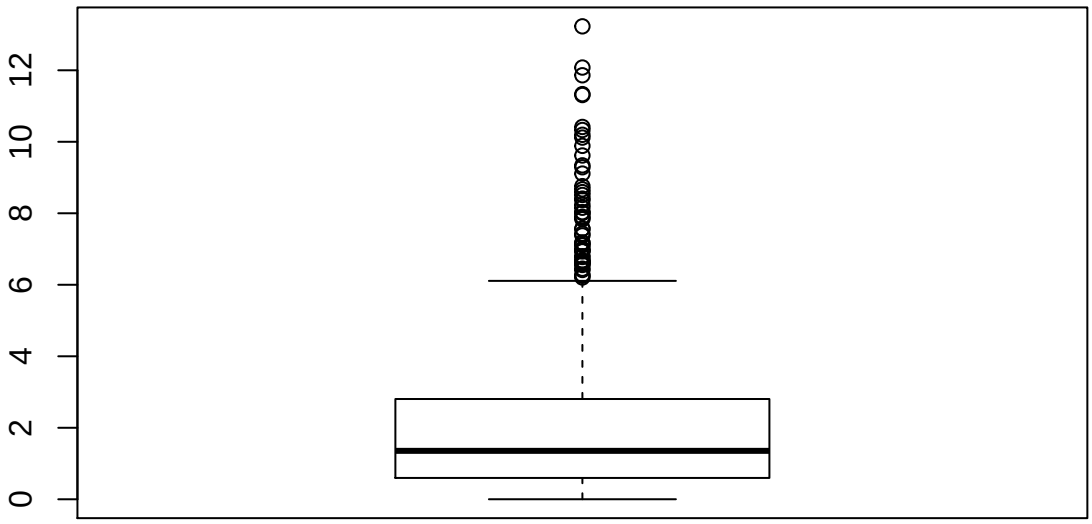
Part C

Create a normal probability plot and boxplot.

```r
#Same as Problem 1C
qqnorm(y=population_dist_exp)
qqline(y=population_dist_exp, col=2)
```




```r
boxplot(x=population_dist_exp)
```



Part D

1. Create 1000 samples of size 10.

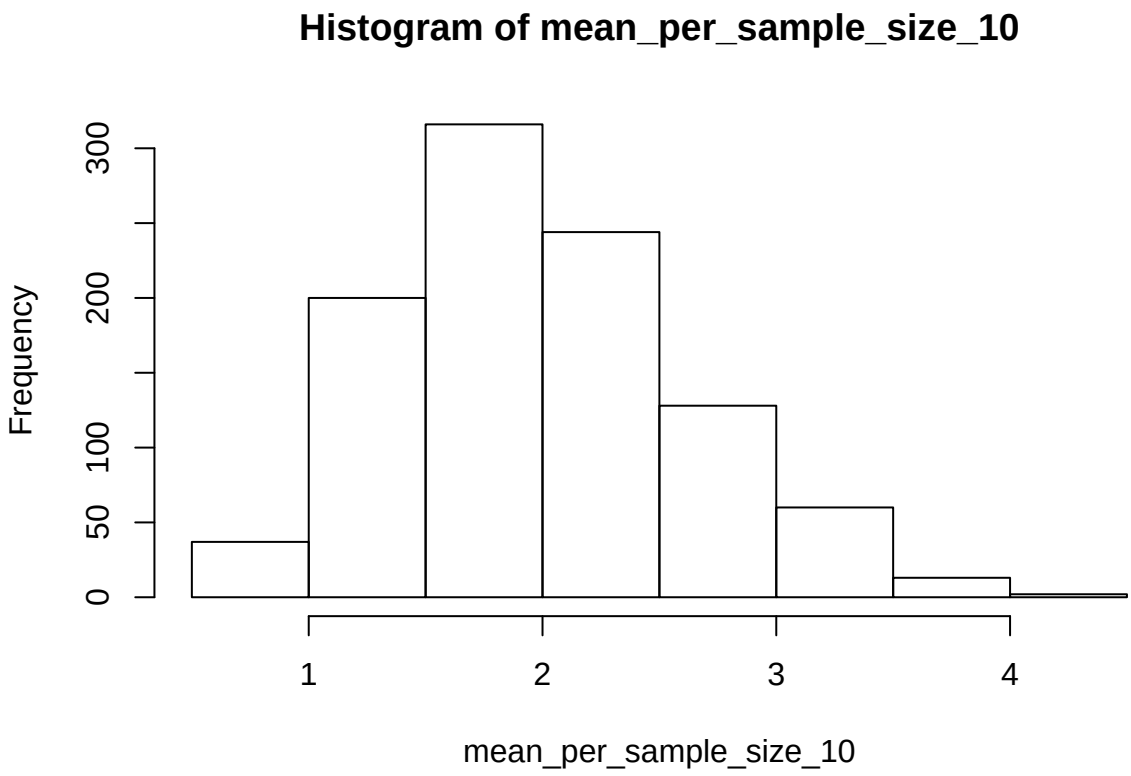
```
samples_size_10 <- replicate(1000, sample(x=population_dist_exp, size=10, replace=FALSE))
```

2. Calculate the mean for these 1000 and graph the sampling distribution of the sample mean for these samples of size 10

```
mean_per_sample_size_10 <- c(1:1000)
for (index in 1:1000) {
  mean_per_sample_size_10[index] = mean(samples_size_10[,index])
}
```

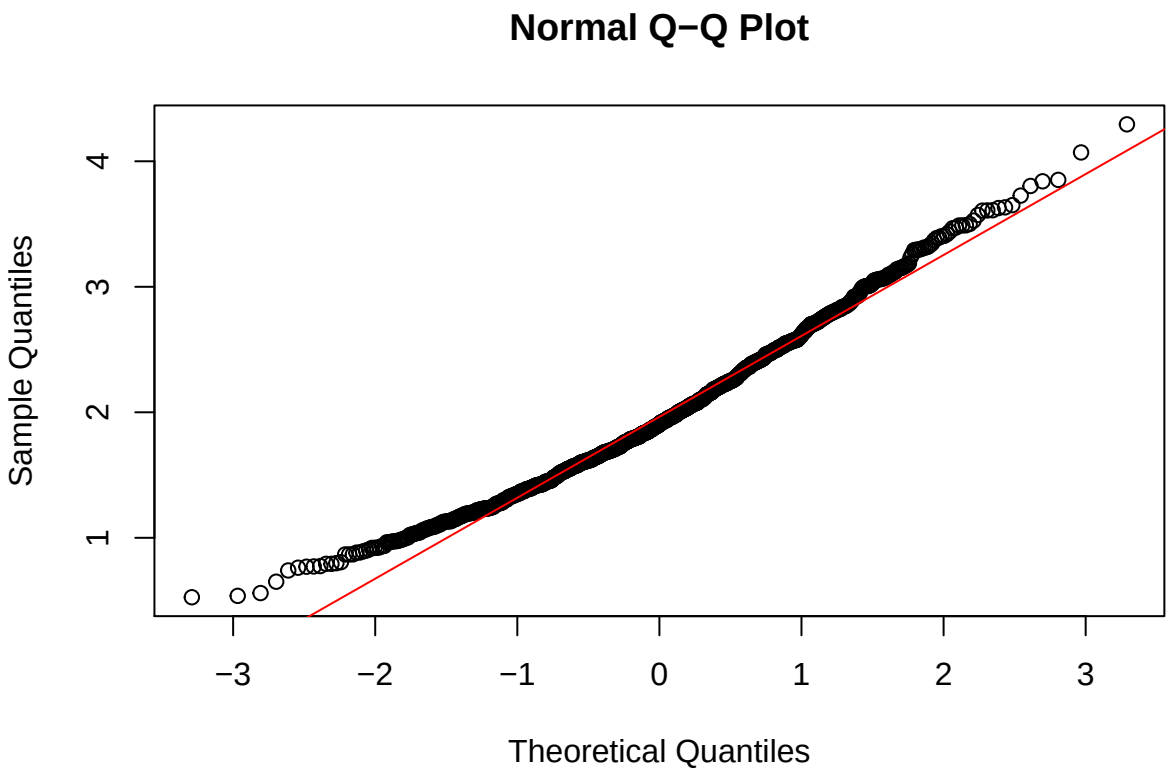
3. Create the histogram of this data with 10 bins.

```
hist(x=mean_per_sample_size_10, breaks=10)
```

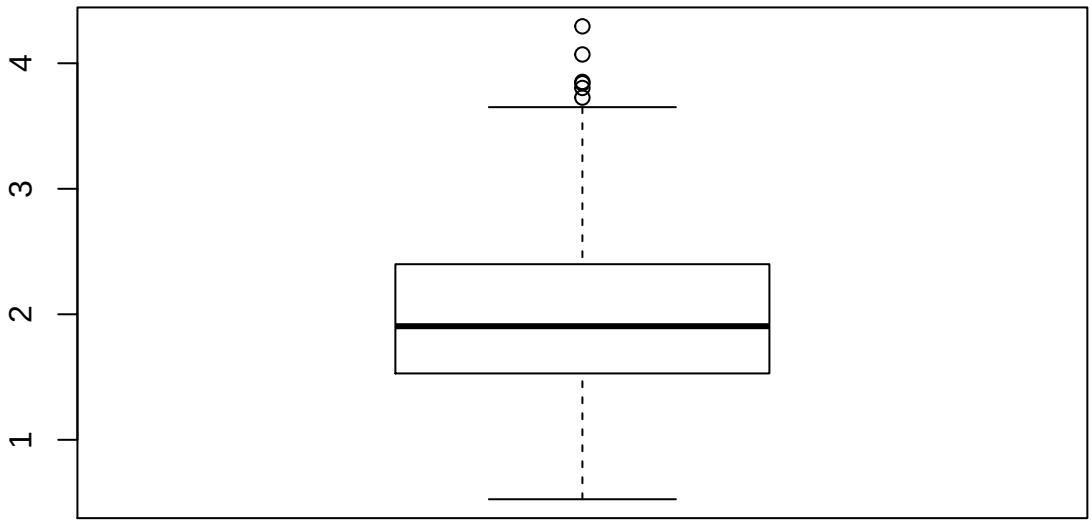


4. Create a normal probability plot and boxplot of this sampling distribution.

```
qqnorm(y=mean_per_sample_size_10)
qqline(y=mean_per_sample_size_10, col=2)
```




```
boxplot(x=mean_per_sample_size_10)
```



Part E

1. Create 1000 samples of size 40

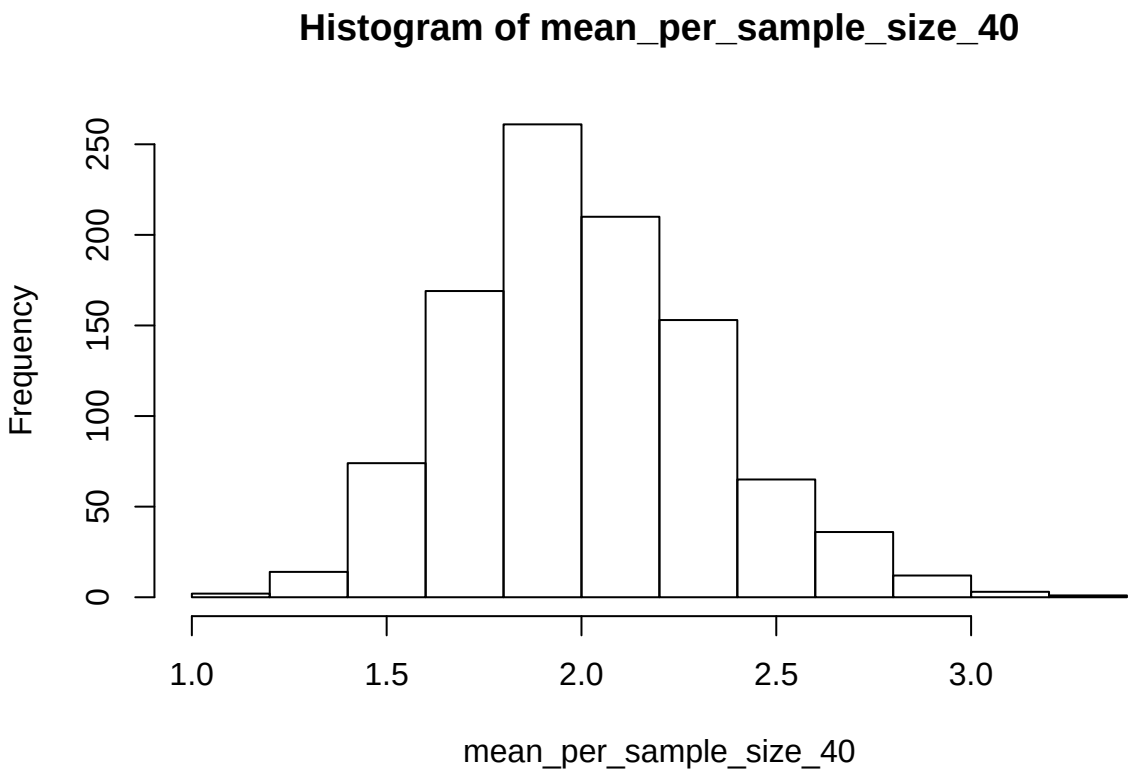
```
samples_size_40 <- replicate(1000, sample(x=population_dist_exp, size=40, replace=FALSE))
```

2. Calculate the mean for these 1000 and graph the sampling distribution of the sample mean for these samples of size 40

```
mean_per_sample_size_40 <- c(1:1000)
for (index in 1:1000) {
  mean_per_sample_size_40[index] = mean(samples_size_40[,index])
}
```

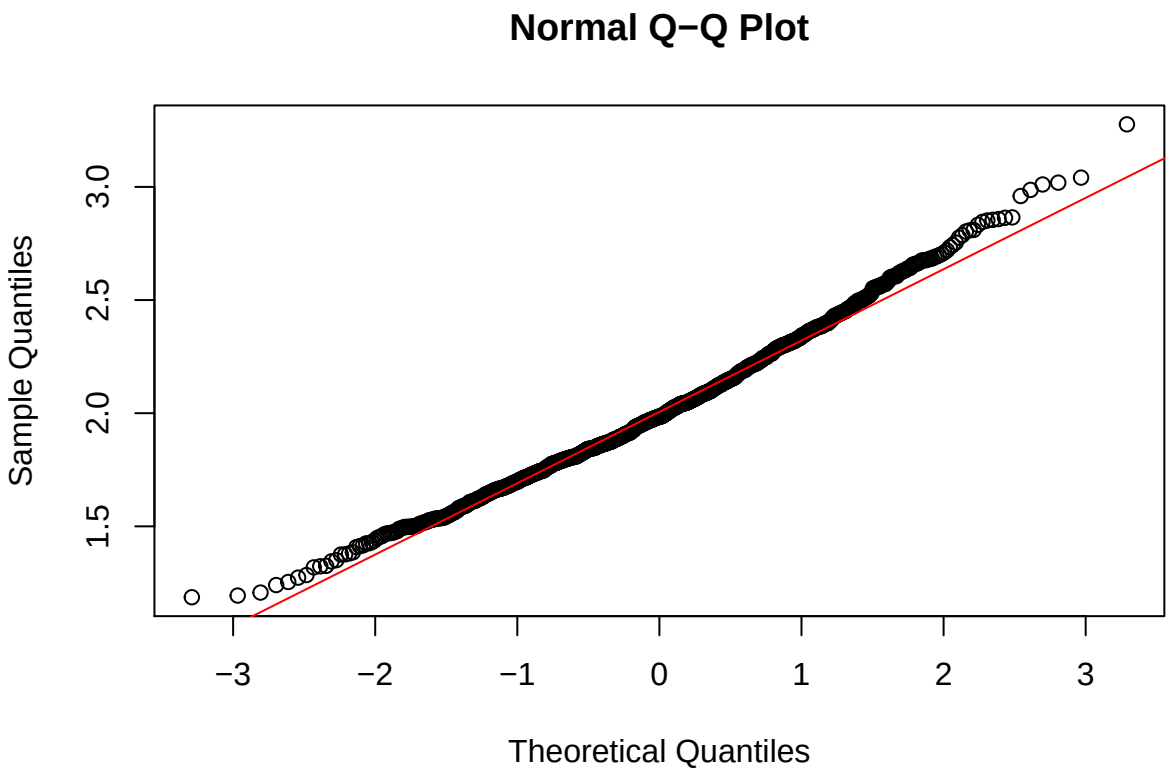
3. Create the histogram of this data with 10 bins.

```
hist(x=mean_per_sample_size_40, breaks=10)
```

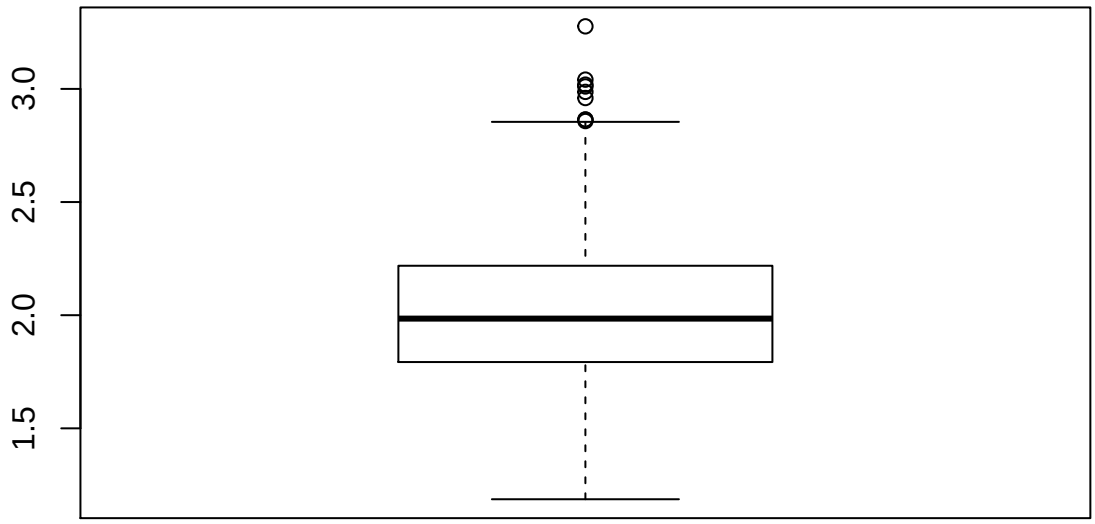


4. Create a normal probability plot and boxplot of this sampling distribution.

```
qqnorm(y=mean_per_sample_size_40)
qqline(y=mean_per_sample_size_40, col=2)
```




```
boxplot(x=mean_per_sample_size_40)
```



5. Create the 90, 95 and 99 % confidence intervals.

```
a <- mean(mean_per_sample_size_40)
s <- sd(mean_per_sample_size_40)
n <- length(mean_per_sample_size_40)

error_90 <- qnorm(0.95)*(s/sqrt(n))
left_90 <- a - error_90
right_90 <- a + error_90

error_95 <- qnorm(0.975)*(s/sqrt(n))
left_95 <- a - error_95
right_95 <- a + error_95

error_99 <- qnorm(0.995)*(s/sqrt(n))
left_99 <- a - error_99
right_99 <- a + error_99
```

90% Confidence Interval [1.99666,2.0303781]

95% Confidence Interval [1.9934303,2.0336079]

99% Confidence Interval [1.9871179,2.0399202]

‘‘‘