hw1

Khoa Vu 705600710

Problem 1: Multiple Testing (20 pts)

We will use the NCI60 cancer cell line microarray data, which consist of 6, 830 gene expression measurements on 64 cancer cell lines. You need to install the ISLR package in R by

```
## Installing package into '/home/kvu1702/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

Then you load the package into R using

```r
library("ISLR")
```

Then you have the object `NCI60` in your R workspace. The object is a list of two elements `NCI60$data` and `NCI60$labs`, where `NCI60$data` is a numeric matrix of 64 rows (i.e., cancer cell lines) and 6, 830 columns (i.e., genes) and `NCI60$labs` is a 64-element character vector containing the cancer types of the cell lines. For example, you can explore the data using the following commands.

```
## [1]   64 6830
```

```
## [1] "CNS"    "CNS"    "CNS"    "RENAL"  "BREAST" "CNS"
```

```
##
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

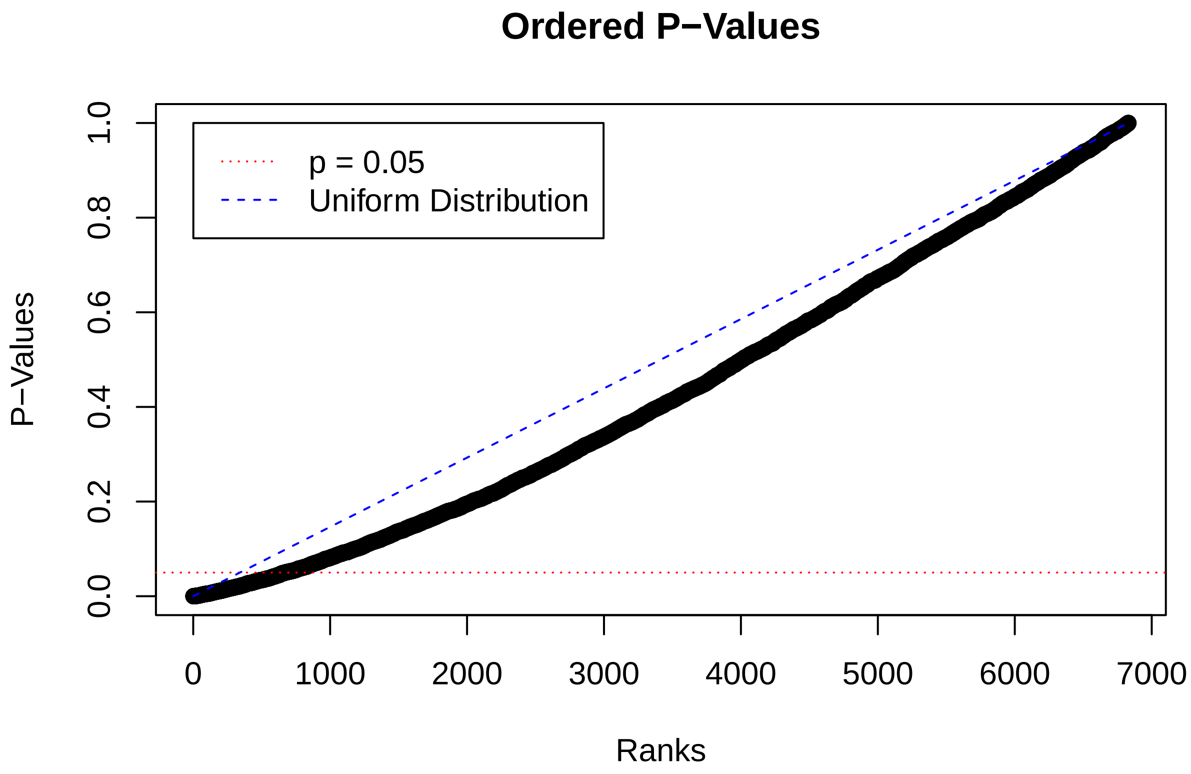
For every gene in the data set, perform a two-sample t-test with a two-sided alternative hypothesis to check if the gene is differentially expressed between the `NSCLC` cell lines and the `RENAL` cell lines. You can use the R function `t.test()`. This will result in 6, 830 tests in total.

1. Order the 6, 830 p-values from the smallest to the largest. Plot the ordered p-values as the y-axis and their corresponding ranks as the x-axis. If all the 6, 830 null hypotheses are true, the p-values should be uniformly distributed. Are the observed p-values likely from a uniform distribution?

Solution:

```r
#Subsetting the data based on cell lines
renal <- subset(NCI60$data, NCI60$labs == 'RENAL')
nsclc <- subset(NCI60$data, NCI60$labs == 'NSCLC')

#Performing per gene t-tests
n_genes <- ncol(NCI60$data)
pvals <- vector(length = n_genes)
for (gene in seq_len(n_genes)) {
  res <- t.test(x = renal[, gene], y = nsclc[, gene], alternative = 'two.sided')
  pvals[gene] <- res$p.value
}
```

```r
#Ordering and plotting the pvalues
ordered_pvals <- sort(pvals)
plot(x = seq_along(ordered_pvals), y = ordered_pvals,
     xlab = "Ranks", ylab = "P-Values", main = "Ordered P-Values")

#We add a vertical line denoting the our significance value of p = 0.05
abline(h = 0.05, col = "red", lty = 3)

#Add a line denoting the uniform distribution
lines(
  x = seq_along(ordered_pvals),
  y = (seq_along(ordered_pvals) / length(ordered_pvals)),
  col = "blue", lty = 2
)

#Adding a legend for the lines
legend(x = 1, legend = c("p = 0.05", "Uniform Distribution"),
       col=c("red", "blue"), lty=c(3, 2))
```



The observed p-values are not likely from a uniform distribution as they deviate slightly from the diagonal line associated with the uniform distribution.

2. What will be the p-value cutoff if you would like to control the Family Wise Error Rate under 0.05 using the Bonferroni correction? How many genes will be identified as differentially expressed using this cutoff?

Solution:

```
#We can apply the Bonferroni correction by dividing our original p-value
#threshold by the number of tests performed.

sig_level_bonferroni <- 0.05/n_genes
DEGs <- 0
for (gene in pvals) {
  if (gene < sig_level_bonferroni) {
    DEGs <- DEGs + 1
  }
}
sig_level_bonferroni
```

```
## [1] 7.320644e-06
```

```
DEGs
```

```
## [1] 0
```

There are 0 genes that are classified as differentially expressed using the p-value of 7.320644e-06 obtained from controlling the Family Wise Error Rate. This p-value is derived by dividing our initial alpha by the number of tests performed.

3. What will be the p-value cutoff if you would like to control the False Discovery Rate under 0.05 using the Benjamini-Hochberg procedure? How many genes will be identified as differentially expressed using this cutoff?

Solution:

```
FDR_Cutoff <- 0.05 / n_genes
DEGs <- 0
for (gene_idx in seq_len(n_genes)) {
  if (ordered_pvals[gene_idx] < (FDR_Cutoff * gene_idx)) {
    FDR_Cutoff <- FDR_Cutoff * gene_idx
    DEGs <- DEGs + 1
    next
  }
  else {
    break
  }
}
FDR_Cutoff
```

```
## [1] 7.320644e-06
```

```
DEGs
```

```
## [1] 0
```

There are 0 genes that are classified as differentially expressed using the p-value of 7.320644e-06 obtained from controlling the False Discovery Rate. We derive the p-value from first ordering our p-values from most to least significant and then obtaining our significance threshold from the Bonferroni correction. We iterate through the ordered p-values and multiply our significance threshold by the number of iterations. Once we have reached a p-value that does not meet our significance threshold, it becomes our new p-value. However, since no genes are significant from the original Bonferroni correction, our p-value from controlling the False Discovery Rate is the same as if we had controlled the Family Wise Error Rate

4. What will be the p-value cutoff if you would like to control the Per-Comparison Error Rate (i.e., significance level) under 0.05? How many genes will be identified as differentially expressed using this

cutoff?

Solution:

```
sig_level <- 0.05
DEGs <- 0
for (gene in pvals) {
  if (gene < 0.05) {
    DEGs <- DEGs + 1
  }
}
DEGs
```

```
## [1] 670
```

There are 670 genes that are classified as differentially expressed using the p-value of 0.05 obtained from controlling the Per-Comparison Error Rate, since the p-value is simply our significance threshold, alpha.

Problem 2 (15 pts)

Use the NCI60 data, give an example to each of the following data analysis tasks (please refer to the Leek and Peng, Science (2015) article we studied in our first lecture). Note: this is an open question; different people may give difference examples to the same task.

1. Descriptive study
2. Exploratory study
3. Inferential study
4. Predictive study
5. Causal study
6. Mechanistic study

Solution:

1. Using the NCI60 data, a possible descriptive study calculates the mean, median, range, minimum, maximum, variance, and other summary statistics of the genes in each cancer cell line.

2. Using the NCI60 data, a possible exploratory study would be to visualize the summary statistics and compare the summary statistics of the same gene across different cancer cell lines.

3. Using the NCI60 data, a possible inferential study would be to perform statistical testing to determine if a gene is a biomarker for a specific cancer cell line.

4. Using the NCI60 data, a possible predictive study would be to perform supervised clustering to see if the gene expression data can be used to predict the cancer cell line or perform binary classification to determine whether a patient has cancer.

5. Using the NCI60 data, a possible causal study would be to determine if a significantly different expression of one gene is responsible for a specific cancer.

6. Using the NCI60 data, a possible mechanistic study would be to conduct gene ontology on differentially expressed genes to reveal possible pathways that are involved in each cancer cell line respectively to other cancer cell lines.

Problem 3 (20 pts)

We will use a data set babies.txt to demonstrate the use of the permutation test. You may use the following R commands to read the data set into R:

```
babies <- read.table("babies.txt", header=TRUE)
bwt.nonsmoke <- subset(babies, smoke==0)$bwt
bwt.smoke <- subset(babies, smoke==1)$bwt
```

The two R objects bwt.nonsmoke and bwt.smoke contain the rows that correspond to the weights of the babies with nonsmoking and smoking mothers respectively.

1. We will generate the following statistics based on a sample size of 10 and observe the following difference:

```r
n <- 10
set.seed(1)
nonsmokers <- sample(bwt.nonsmoke , n)
smokers <- sample(bwt.smoke , n)
diff <- mean(smokers)- mean(nonsmokers)
```

The question is whether this observed difference is statistically significant. We do not want to rely on the assumptions needed for the t test, so instead we will use permutations. We will reshuffle the data and recompute the mean. We can create one permuted sample with the following code:

```r
dat <- c(smokers, nonsmokers)
shuffle <- sample(dat)
smokers_star <- shuffle[1:n]
nonsmokers_star <- shuffle[(n+1):(2*n)]
diff_star <- mean(smokers_star)-mean(nonsmokers_star)
```

The last value is one observation from the null distribution we will conduct. Set the seed at 1, and then repeat the permutation for 1,000 times to create a null distribution. What is the permutation derived p-value for our observation diff?

Solution:

```r
set.seed(1)
n_permutations <- 1000
more_extreme <- 0
for (permutation in seq_len(n_permutations)) {
  shuffle <- sample(dat)
  smokers_star <- shuffle[1:n]
  nonsmokers_star <- shuffle[(n+1):(2*n)]
  diff_star <- mean(smokers_star)-mean(nonsmokers_star)
  if (abs(diff_star) >= abs(diff)) {
    more_extreme <- more_extreme + 1
  }
}
permutation_pval <- more_extreme / n_permutations
permutation_pval
```

```
## [1] 0.116
```

The permutation based p-value based on the differences in mean is 0.061.

2. Repeat the above exercise, but instead of the differences in mean, consider the differences in median.

Solution

```r
diff_med <- median(smokers)- median(nonsmokers)
```

What is the permutation based p-value?

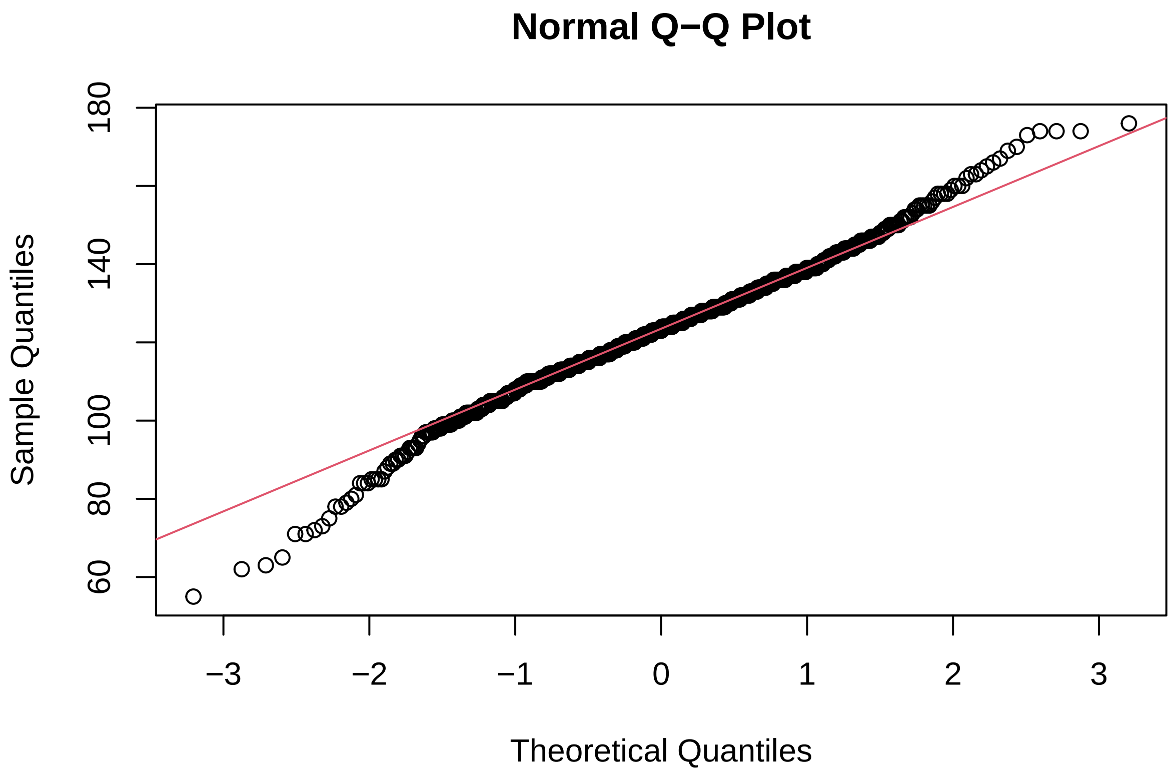
```r
set.seed(1)
n_permutations <- 1000
more_extreme <- 0
for (permutation in seq_len(n_permutations)) {
  shuffle <- sample(dat)
  smokers_star <- shuffle[1:n]
  nonsmokers_star <- shuffle[(n+1):(2*n)]
  diff_star <- median(smokers_star)-median(nonsmokers_star)
  #Because it is the two-sided test, we need to consider the possibility of
  #Greater than or equal than
  if (abs(diff_star) >= abs(diff_med)) {
    more_extreme <- more_extreme + 1
  }
}
permutation_pval <- more_extreme / n_permutations
permutation_pval
```

```
## [1] 0.286
```

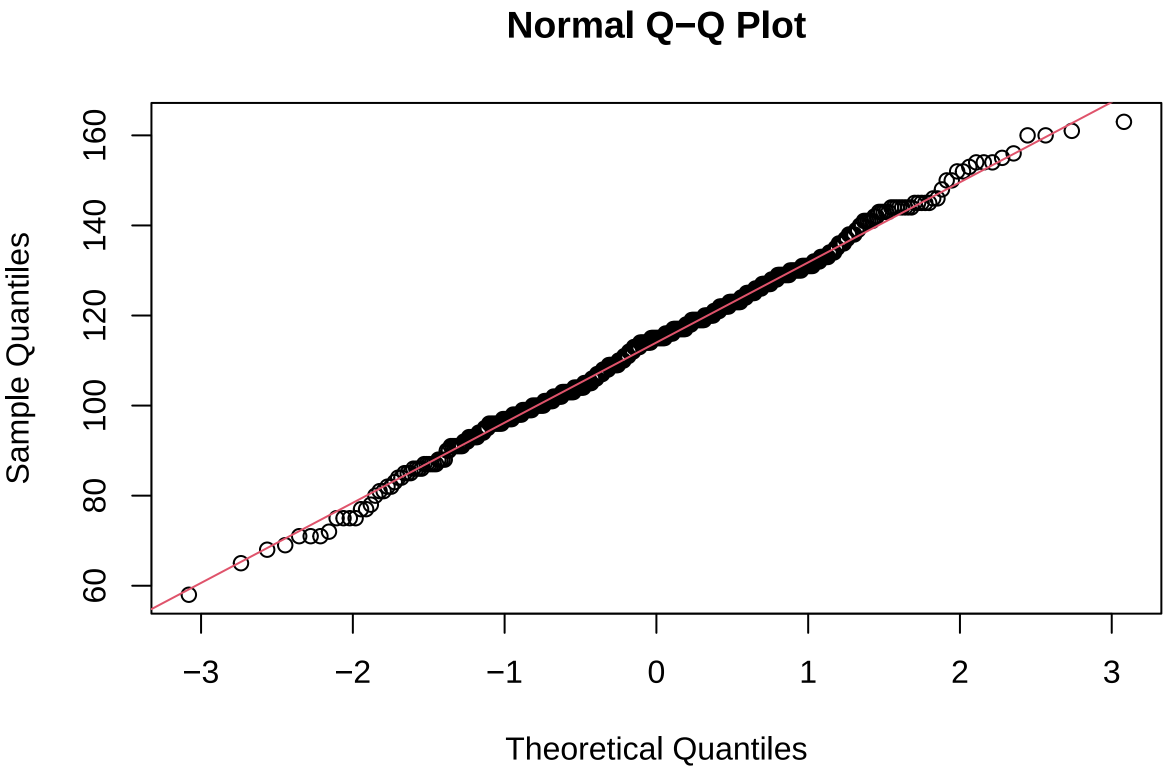
The permutation based p-value based on the differences in median is 0.119.

Problem 4 (10 pts) We can use the same data set babies.txt to demonstrate the use of t test. For a sample with sample size n = 10 like in Problem 3, the Central Limit Theorem does not apply, and we need to check whether the data are approximately Gaussian under each condition. The following R codes can be used to check the Gaussian assumption:

```
qqnorm(bwt.nonsmoke)
qqline(bwt.nonsmoke,col=2)
```



```
qqnorm(bwt.smoke)
qqline(bwt.smoke,col=2)
```



1. Please display the Q-Q plots and argue whether the Gaussian assumption is reasonable for this data set.

Solution:

The Q-Q plots are displayed above. Since most of the points fall along the 45-degree reference line, the Gaussian assumption is reasonable for this data set. However, the nonsmoking data set has a slight deviation from the reference line and is slightly skewed.

2. Perform the t test using the following R codes. Compare the resulting p-value with what you obtain from the permutation test in Problem 3. Do you reach the same conclusion?

```
result <- t.test(nonsmokers, smokers)
result$p.value
```

```
## [1] 0.1032748
```

Solution:

The resulting p-value of the t-test is 0.103, which is higher than our significance level of 0.05. We thus cannot reject the null hypothesis, a conclusion consistent with the results from problem 3, since those derived p-values were also higher than our significance level of 0.05.

Problem 5 (10 pts)

A test for cystic fibrosis has an accuracy of 99%. Specifically, we mean that: P(+|D) = 0.99 and P(−|ND) = 0.99, where + and − stand for positive and negative test results respectively, and D and ND represent the existence and nonexistence of the disease respectively. The cystic fibrosis rate in the general population is 1 in 3,900, that is P(D) = 0.00025. If we select a random person and they test positive, what is probability that the person has the disease?

Hint: Use the Bayes Theorem.

Note: The posterior mean you derive in this problem is often called the "maximum a posteriori" (MAP) estimator of µ.

Hint: Please follow my derivation for the Bayesian estimator of 2 in the t-test setting.

Solution: We are given:

P(+|D) = 0.99

P(−|ND) = 0.99

P(D) = 0.00025

We can also derive:

P(ND) = 1 - P(D) = 1 - 0.00025 = 0.99975

We can also derive:

P(Error) = 1 - P(Acc)

P(Error) = 1 - 0.99 = 0.1

An error occurs when a person tests negative when they have the disease or test postive when they do not have the disease, -|D and +|D. So P(-|D) = 0.01 and P(+|D) = 0.01.

We use Bayes theorem, the law of total probability, and definition of conditional probability.
From the law of total probability:

$$P(B) = \sum_{i=1}^{k} P(B \cap A_i)$$

From the definition of conditional probability:

$$P(B \cap A) = P(A)|P(B|A)$$

From Bayes Rules and combinig the above two equations:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_{i=1}^{k} P(B|A_i)P(A_i)}$$

We want to find P(have the disease | test positive) = P(D|+). We can use Bayes theorem by substituting in for our events:

$$P(D|+) = \frac{P(+|D)P(D)}{P(+)} = \frac{P(+|D)P(D)}{P(+|D)P(D) + P(+|ND)P(ND)}$$

We can now substitute in our given values:

$$= \frac{(0.99)(0.00025)}{(0.99)(0.00025) + (0.01)(0.99975)} = 0.02416$$

Problem 6 (10 pts) This problem is to help you understand what we mean by a random sample, and why we say that the sample mean is a random variable. Use the following R code to simulate a sample with sample size n = 10 and calculate the sample mean:

```r
n <- 10
x <- rnorm(n)
mean(x)
```

```
## [1] 0.1945933
```

Repeat the simulation for one million times, plot the distribution of the one million sample means.

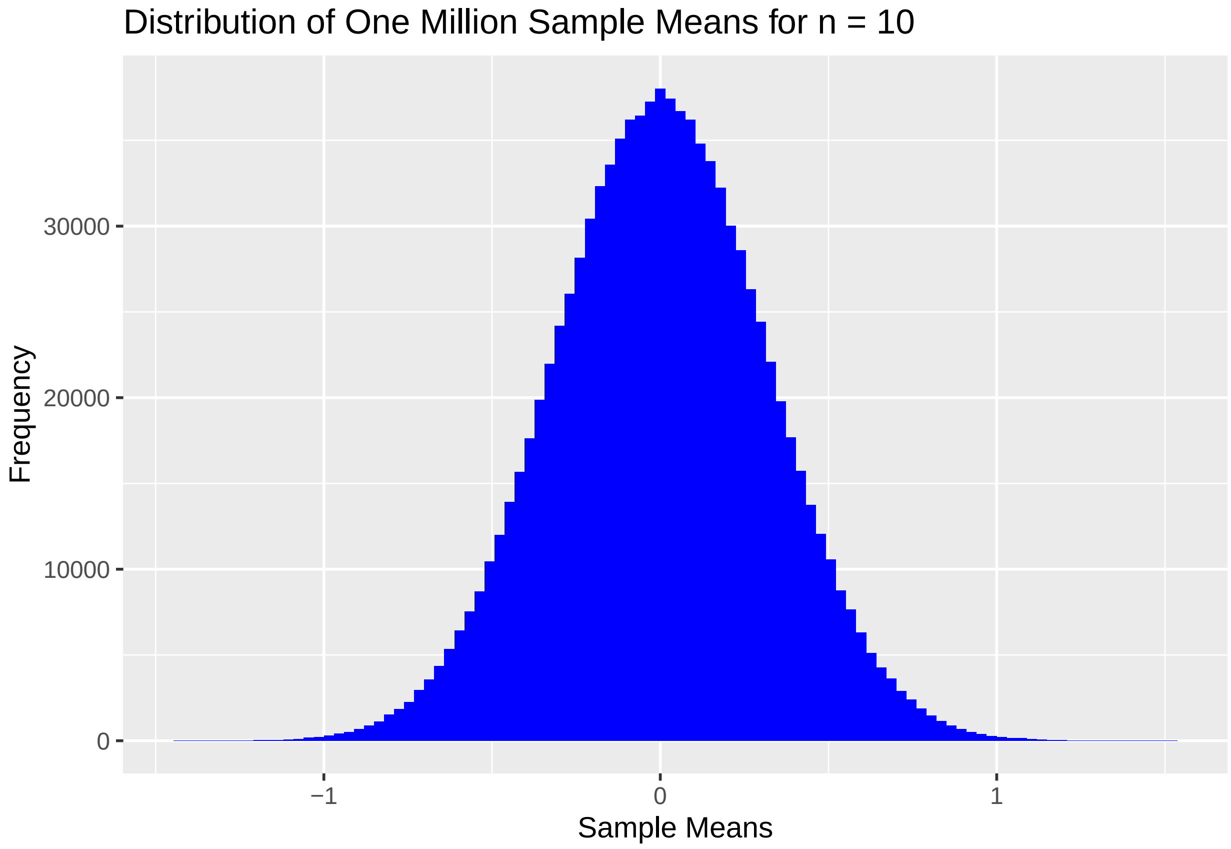
Now increase n to 100. Repeat the above simulation, how does the distribution of the one million sample means change?

Now further increase n to 1000. Repeat the above simulation, how does the distribution of the one million sample means change?

What do we conclude about the distribution of the sample mean?

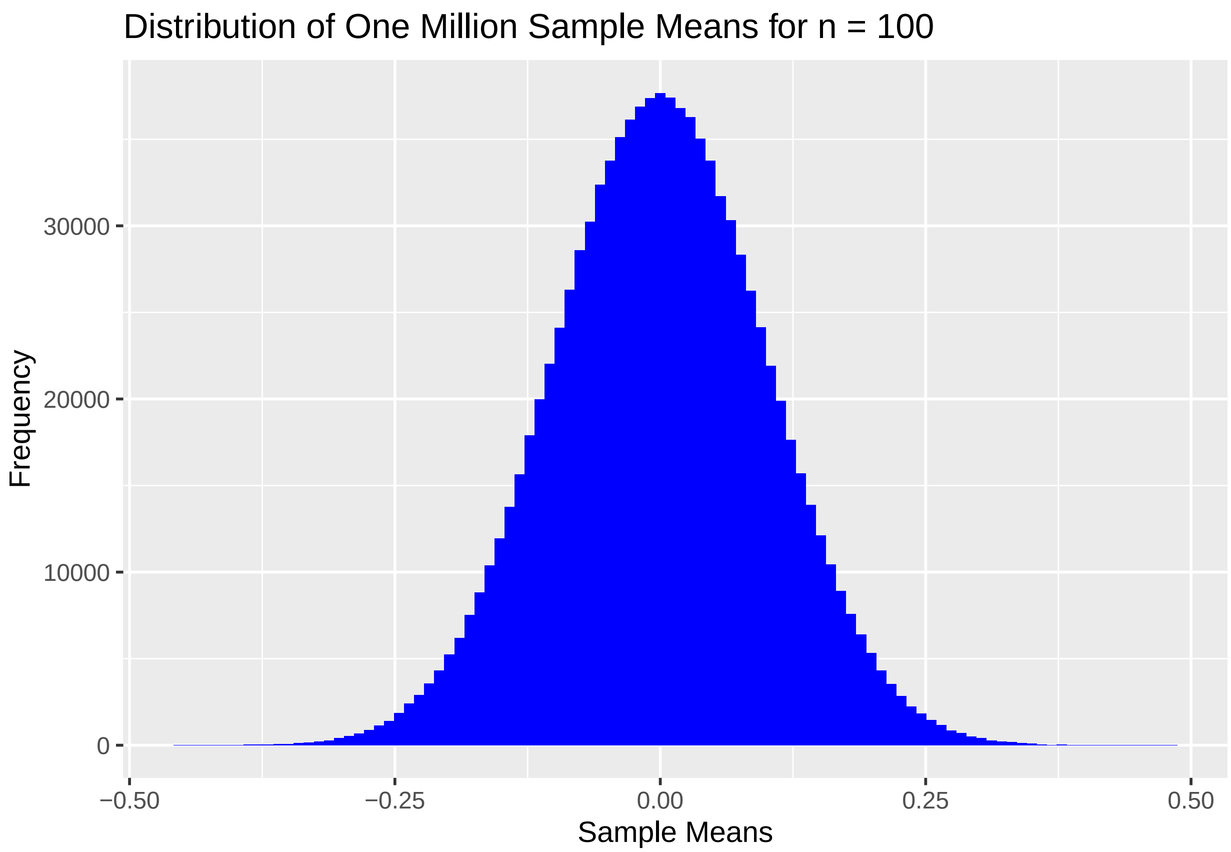
Solution:

```
library("ggplot2")

#Repeat the simulation for one million times
n <- 10
n_iter <- 1000000
means <- vector(length = n_iter)
for (i in seq_len(n_iter)) {
  x <- rnorm(n)
  means[i] <- mean(x)
}

ggplot(data = data.frame(means), aes(x = means)) +
  geom_histogram(bins = 100, fill = "blue") +
  xlab("Sample Means") +
  ylab("Frequency") +
  ggtitle("Distribution of One Million Sample Means for n = 10")
```



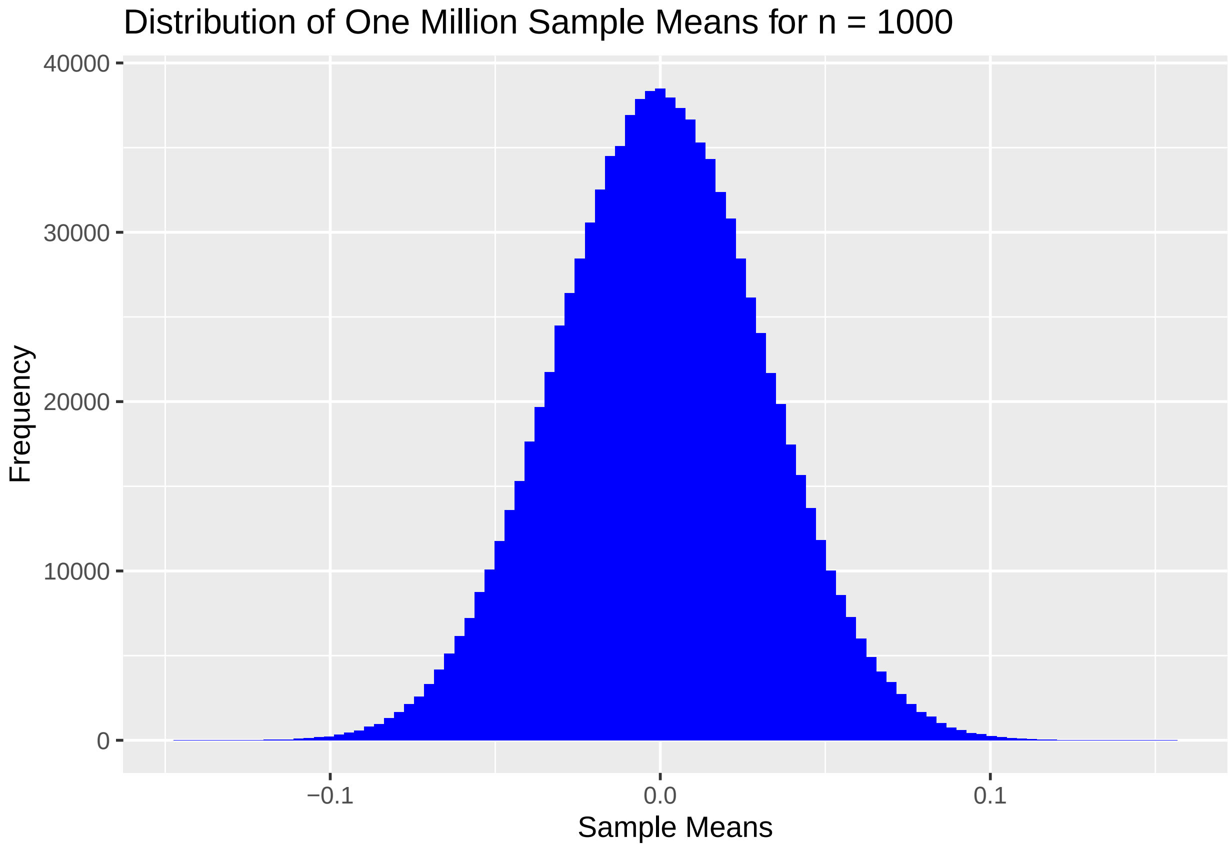
The distribution of the sample mean resembles a normal/Gaussian distribution.

```r
library("ggplot2")

#Increase n to 100
n <- 100
n_iter <- 1000000
means <- vector(length = n_iter)
for (i in seq_len(n_iter)) {
  x <- rnorm(n)
  means[i] <- mean(x)
}

ggplot(data = data.frame(means), aes(x = means)) +
  geom_histogram(bins = 100, fill = "blue") +
  xlab("Sample Means") +
  ylab("Frequency") +
  ggtitle("Distribution of One Million Sample Means for n = 100")
```



The sample mean distribution appears smoother when increasing n to 100 versus n = 10.

```
library("ggplot2")

#Increase n to 1000
n <- 1000
n_iter <- 1000000
means <- vector(length = n_iter)
for (i in seq_len(n_iter)) {
  x <- rnorm(n)
  means[i] <- mean(x)
}

ggplot(data = data.frame(means), aes(x = means)) +
  geom_histogram(bins = 100, fill = "blue") +
  xlab("Sample Means") +
  ylab("Frequency") +
  ggtitle("Distribution of One Million Sample Means for n = 1000")
```



The sample mean distribution appears smoothest when increasing n to 1000 versus n = 10 and n = 100. We can include the distribution of the sample mean resembles a normal/Gaussian distribution and becomes smoother as n increases. This is consistent with the central limit theorem, which states that the sampling distribution is approximately normal if the samples are large enough.