Lab #2 Activities

Adejare Taiwo

A reminder that the R code we have covered in class is available on the lecture section UM Learn page, under **Content** > **Course Material**.

Knit this file to pdf to see the questions in a more readable format.

Question 1:

We can generate values from a normal distribution using the line of R code `rnorm(n,mu,sigma)`, where **n** is the number of values to generate, **mu** is the mean of the normal distribution and **sigma** is the standard deviation of the normal distribution. When we generate these **n** values, each time that line of code is run, it will generate a new set of **n** values so you can never reproduce the same set of values, unless you use the `set.seed()` function just before you use `rnorm()` with a number passed in to `set.seed()` (any number).

(a) Generate 1000 values from a normal distribution with mean 100 and standard deviation 10 and store these values in a vector `x`. Before generating the values, enter the following line of code into the code chunk: `set.seed(25)`.

```
set.seed(25);
x <- rnorm(1000, mean = 100, sd = 10)
```

(b) Create a vector called `indices` that will contain the indices of `x` that have values less than 80.

```
indicies <- which(x < 80)
```

(c) Now write the R code that determines how many values of `x` are less than 80.

```
length(indicies)
```

```
## [1] 20
```

(d) What proportion of the 1000 values are less than 80?

0.02

(e) Based on the 68-95-99.7% rule, what proportion of the population has a value of `x` less than 80?

2

Question 2:

Suppose a university course has an enrollment of 100 students. The table below gives the letter grades for the students enrolled in the course:

Letter Grade	Frequency
A	40
B	35
C	15
D	7
F	3

Without manually typing 100 letter grades one-by-one, create a vector called `grades` that contains the 100 letter grades earned by students in this course. Print the `grades` vector.

Hint: if we have two vectors `x` and `y`, we can combine them into one vector `z` as follows:

```
x = 1:3
y = 4:6
z = c(x,y)
z
```

```
## [1] 1 2 3 4 5 6
```

```
grades <- c(rep("A", 40), rep("B", 35), rep("C", 15), rep("D", 7), rep("F", 3))
```

Question 3:

Suppose the suggested retail price of an item being sold in a store is $50.

(a) Create a vector called `discountPercent` with each of the numbers from 0 to 50, in increments of 5, using the `seq()` function. Then print the `discountPercent` vector.

```
discountPercent <- seq(from = 0, to = 50, by = 5)
print(discountPercent)
```

```
##  [1]  0  5 10 15 20 25 30 35 40 45 50
```

(b) Each component of `discountPercent` will be used to determine each component of a new vector called `finalPrice`, where each component of `finalPrice` will contain the price the customer must pay if the percentage discount is given by the corresponding component of `discountPercent`. Create the `finalPrice` vector as described and print the vector.

Do not create `finalPrice` by manually typing `c(50,47.5,45,...)`. Instead, use a formula that calculates the final price based on the suggested retail price and the percentage discount.

```
finalPrice <- 50 * (1 - discountPercent / 100)
print(finalPrice)
```

```
##  [1] 50.0 47.5 45.0 42.5 40.0 37.5 35.0 32.5 30.0 27.5 25.0
```

(c) Create a data frame with two columns, consisting of the `discountPercent` and `finalPrice` vectors. Print the data frame.

```r
priceFrame <- data.frame(DiscountPercent = discountPercent, FinalPrice = finalPrice)
print(priceFrame)
```

```
##    DiscountPercent FinalPrice
## 1                0       50.0
## 2                5       47.5
## 3               10       45.0
## 4               15       42.5
## 5               20       40.0
## 6               25       37.5
## 7               30       35.0
## 8               35       32.5
## 9               40       30.0
## 10              45       27.5
## 11              50       25.0
```

Question 4:

In this question, we will use use the built-in `mtcars` dataset. The dataset can be viewed by typing `mtcars` at the Console and the help documentation can be seen by typing `?mtcars` at the Console. The dataset is stored in an R data frame.

Create an R list of length 3 with the following statistics on the **rear axle ratio** variable in the three elements of the list. Then print the list.

1. The sample mean (average) rear axle ratio
2. The sample standard deviation of the rear axle ratios. (If we have a vector `x` of numerical values, then `sd(x)` calculates the sample standard deviation.)
3. The five number summary of the rear axle ratios. (If we have a vector `x` of numerical values, then `fivenum(x)` calculates the five-number summary.)

```r
mean_mt <- mean(mtcars$drat)
sd_mt <- sd(mtcars$drat)
summary_mt <- fivenum(mtcars$drat)

stats_list <- list(mean_mt,sd_mt, summary_mt)

print(stats_list)
```

```
## [[1]]
## [1] 3.596563
##
## [[2]]
## [1] 0.5346787
##
## [[3]]
## [1] 2.760 3.080 3.695 3.920 4.930
```