```
install.packages("ggplot2")
library(ggplot2)

install.packages("diagram")
library(diagram)

install.packages("readxl")
library(readxl)

install.packages("Rcpp")
library(Rcpp)

install.packages("dplyr")
library(dplyr)


# This is the document for Homework 2
# Eizabeth Hollis

# 1. Keeping track of stock prices
# 1. a. How would you input this data as a vector?
US_stock = c(58, 57, 55, 52, 51, 46, 48, 49, 49, 51)
View(US_stock)
#This is the US Stock Prices as a vector list

# 1. b. How would you find the mean of the data above?
summary(US_stock) #Basic descriptions - this is how you find the mean
# The mean is 51.60

# 1. c. How would you sort the above values in descending order?
sort(US_stock, decreasing = TRUE)


# 2. Convert the values from US dollars to Euros. One US dollar is equivalent
to 0.88 Euros.
# 2. a. What formula would you use to convert all the values?
exchange = US_stock / 0.88
round(exchange, digits = 0)
#Here I make it so the vector is the actual rounded up values
exchange = c(66, 65, 62, 59, 58, 52, 55, 56, 56, 58)


# 2. b. What is your output in R after applying this conversion?
# The output is 66, 65, 62, 59, 58, 52, 55, 56, 56, 58 after being rounded up
(from console)


# 2. c. What is the difference between Us dollar and Euros for each value?
difference_between = exchange - US_stock
round(difference_between, digits = 0)
# The difference between exchange and us_stock (rounded up) is 8, 8, 8, 7, 7,
6, 7, 7, 7, 7.
#Here I make it so the vector is the actual rounded up values
difference_between = c(8, 8, 8, 7, 7, 6, 7, 7, 7, 7)
```

```
# 3. You are interested in knowing how many values are over 50 in US dollars
and less than 50 in Euros
sum(US_stock > 50) # Tells us six values are over 50

sum(exchange < 50) # Tells us no values are under 50


# 4. How many values are the same between US dollars and Euros?
# 4. a. What formula is used to find this out?
intersect(US_stock, exchange)

# 4. b. Are there any values that are the same between the two?
# Yes! The values they share are 58, 55, and 52.


# 5. Answer each of the following points:
# 5. a. What code would you use to find if any values in the US stock prices
are divisible by 4?
divide_code = US_stock %% 4


# 5. b. How many values are there and what are these values?
# Values are (with corresponding from US_stock) is 58 % 4 = 2
# (not divisible), 57 % 4 = 1 (not divisible), 55 % 4 = 3 (not divisible)
# 52 % 4 = 0 (is divisible), 51 % 4 = 3 (not divisible),
# 46 % 4 = 2 (not divisible), 48 % 4 = 0 (is divisible),
# 49 % 4 = 1 (not divisible), 49 % 4 = 1 (not divisible),
# 51 % 4 = 3 (not divisible). 10 values in all.


# 5. c. What code would you use to find if any values in the
# Euro stock prices are even values?
even_euro = exchange[exchange %% 2 == 0]
# The code above gives a vector for only the even values


# 5. d. How many values are there and what are these values?
# There are 7 values that are even within the exchange or euro stock
# vector which are 66, 62, 58, 52, 56, 56, and 58.


# 5. e. What code would be used to find if any of them greater than 50
# or if all of them are greater than 50, in those Euro stock prices?

# I wasn't too sure if you wanted the euro stock price vector or
# the new vector we created from the previous question so I did both.
sum(exchange > 50) # Tells us all 10 values are over 50
sum(even_euro > 50) # Tells us all 7 values over 50

all(exchange > 50) # Tells us all values are over 50
all(even_euro > 50) # Tells us all values are over 50
```

```
# The next questions are using the mcds dataset
# 6. a. How many rows and columns are in your dataset?
dim(mcds) #There are 260 rows and 24 columns


# 6. b. What is the name of the 4th column in the dataset?
# 4th column is the Calories column
four_column = select(mcds, Calories)
# The above code just selected the Calories column from the table


# 6. c. Show the summary statistics for the 4th column in the
# dataset (like min, 1st quartile, median, etc.)
summary(four_column)
# Min - 0.0; 1st Qu - 210.0; Median - 340.0; Mean - 368.3;
# 3rd Qu - 500.0; Max - 1880.0


# 7. a. Create a boxplot showing the range of Calories by Category.
boxplot(mcds$Calories ~ mcds$Category)


# 7. b. Add color to the boxplot
boxplot(mcds$Calories ~ mcds$Category,
        main = "Boxplot showing range of Calories against Category of food",
        xlab = "Category", ylab = "Calories",
        col = c("deeppink", "mediumblue", "coral", "brown", "darkgrey",
                "darkgreen", "cadetblue", "cornsilk", "bisque", radius = 1))



# 7. c. What are one or two observations you see based on the data?
# Many of the boxplots are roughly about the same size so that means
# they have nearly the same range for different categories. There is
# a lot of suspected outliers within the orange boxplot which is
# under the chicken % Fish category.



# 8. a. Filter or subset the data by those items that are greater
# than 800 calories.
big_cal = subset(mcds, Calories > 800)
View(big_cal)


# 8. b. How many items are greater than 800?
dim(big_cal) # 12 items greater than 800


# 8. c. Which Category names are listed in this dataset where you
# are only looking at those items greater than 800?
cat_names = select(big_cal, Category)
# Category names include Breakfast, Chicken & Fish, and Smoothies & Shakes
```

```
cat_names1 = table(big_cal$Category)
# The above code transforms the single Category data into values
# for a pie chart

# 8. d. Create a pie chart with these categories represented. Add title and
color.
pie(cat_names1, main = "Pie Chart of different Categories of items over 800
Calories",
    col = c("deeppink", "mediumblue", "coral", radius = 1))
```