Problem Set 1

Yifan Li

Problem 1

a

Using `read.table` to read a file written in txt format. For the separation, using ',’. Then according to the description file, 'wine.names', there are 14 attributes in the data file with a class number listed in the first column. So adding `col.names` in the code `read.table`. Such that, one can produce a `data.frame` object with appropriate columns names.

```
wines_data <- read.table("wine.data",
                         sep = ",",
                         col.names=c('class_number',
                                     'Alcohol',
                                     'Malic_acid',
                                     'Ash',
                                     'Alcalinity_of_ash',
                                     'Magnesium',
                                     'Total_phenols',
                                     'Flavanoids',
                                     'Nonflavanoid_phenols',
                                     'Proanthocyanins',
                                     'Color_intensity',
                                     'Hue',
                                     'OD280_OD315_of_diluted_wines',
                                     'Proline'))
```

b

First, using `wines_data['class_number]==i` for `i in [1, 2, 3]` to create a new `data.frame` that has `True` only if the class numbers match with `i`. After that, using

a `sum` function to compute the number of `True`, which is the number of the wine class.

```
num_class_one <- sum(wines_data['class_number'] == 1)
num_class_two <- sum(wines_data['class_number'] == 2)
num_class_three <- sum(wines_data['class_number'] == 3)
```

The results are:

```
num_class_one
```

```
[1] 59
```

```
num_class_two
```

```
[1] 71
```

```
num_class_three
```

```
[1] 48
```

So, the number of wines within each class is correct as reported in the file "wine.names".

c

1. The correlation between alcohol content and color intensity can be derived from a function `cor`. The alcohol content has variable name `Alcohol`, the color intensity has variable name `Color_intensity`. So the input of the function will be:

```
cor(wines_data['Alcohol'], wines_data['Color_intensity'])
```

```
        Color_intensity
Alcohol       0.5463642
```

2. For each class, first the whole data from that class will be extracted, then the correlation between alcohol content and color intensity will be calculated.

For class one:

```r
class_one <- wines_data[wines_data['class_number'] == 1, ]
class_one_cor <- cor(class_one['Alcohol'], class_one['Color_intensity'])
class_one_cor
```

```
        Color_intensity
Alcohol       0.4082913
```

For class two:

```r
class_two <- wines_data[wines_data['class_number'] == 2, ]
class_two_cor <- cor(class_two['Alcohol'], class_two['Color_intensity'])
class_two_cor
```

```
        Color_intensity
Alcohol       0.2697891
```

For class three:

```r
class_three <- wines_data[wines_data['class_number'] == 3, ]
class_three_cor <- cor(class_three['Alcohol'], class_three['Color_intensity'])
class_three_cor
```

```
        Color_intensity
Alcohol       0.3503777
```

Through comparison, one will find that class one has the highest correlation which is `0.4082913`, while class two has the lowest correlation which is `0.2697891`.

3. To find the wine with highest color intensity, using **`which.max`** function, with attributes **`wines_data$Color_intensity`**. This will yield the index of the wine with highest color intensity. Then using this index to find the wine, after that extract its alcohol content.

```r
index <- which.max(wines_data$Color_intensity)
target_wine <- wines_data[index, ]
target_wine$Alcohol
```

```
[1] 14.34
```

Finally extract the alcohol content from the target wine, which is `14.34`.

4. First, find the number of wines that have a higher content of proanthocyanins than ash. Then divide it by the sum of three classes of wines, which will give us the percentage of wines had a higher content of proanthocyanins compare to ash, which is `8.426966%`.

```
num <- sum(wines_data$'Proanthocyanins' > wines_data$'Ash')
percentage <- num * 100 / (num_class_one + num_class_two + num_class_three)
percentage
```

```
[1] 8.426966
```

d

```
average_table <- data.frame(id = 1: 4,
                            class_number = c('overall', '1', '2', '3'),

                            Mean_Alcohol = c(mean(wines_data$Alcohol),
                                             mean(class_one$Alcohol),
                                             mean(class_two$Alcohol),
                                             mean(class_three$Alcohol)),

                            Mean_Malic_acid = c(mean(wines_data$Malic_acid),
                                                mean(class_one$Malic_acid),
                                                mean(class_two$Malic_acid),
                                                mean(class_three$Malic_acid)),

                            Mean_Ash = c(mean(wines_data$Ash),
                                         mean(class_one$Ash),
                                         mean(class_two$Ash),
                                         mean(class_three$Ash)),

                            Mean_Alcalinity_of_ash = c(
                              mean(wines_data$Alcalinity_of_ash),
                              mean(class_one$Alcalinity_of_ash),
                              mean(class_two$Alcalinity_of_ash),
                              mean(class_three$Alcalinity_of_ash)),

                            Mean_Magnesium = c(mean(wines_data$Magnesium),
                                               mean(class_one$Magnesium),
                                               mean(class_two$Magnesium),
```

```
                                        mean(class_three$Magnesium)),

                          Mean_Total_phenols = c(mean(wines_data$Total_phenols),
                                                 mean(class_one$Total_phenols),
                                                 mean(class_two$Total_phenols),
                                                 mean(class_three$Total_phenols)),

                          Mean_Flavanoids = c(mean(wines_data$Flavanoids),
                                              mean(class_one$Flavanoids),
                                              mean(class_two$Flavanoids),
                                              mean(class_three$Flavanoids)),

                          Mean_Nonflavanoid_phenols = c(
                            mean(wines_data$Nonflavanoid_phenols),
                            mean(class_one$Nonflavanoid_phenols),
                            mean(class_two$Nonflavanoid_phenols),
                            mean(class_three$Nonflavanoid_phenols)),

                          Mean_Proanthocyanins = c(mean(wines_data$Proanthocyanins),
                                                   mean(class_one$Proanthocyanins),
                                                   mean(class_two$Proanthocyanins),
                                                   mean(class_three$Proanthocyanins)),

                          Mean_Color_intensity = c(mean(wines_data$Color_intensity),
                                                   mean(class_one$Color_intensity),
                                                   mean(class_two$Color_intensity),
                                                   mean(class_three$Color_intensity)),

                          Mean_Hue = c(mean(wines_data$Hue), mean(class_one$Hue),
                                       mean(class_two$Hue), mean(class_three$Hue)),

                          Mean_OD280_OD315_of_diluted_wines = c(
                              mean(wines_data$OD280_OD315_of_diluted_wines),
                              mean(class_one$OD280_OD315_of_diluted_wines),
                              mean(class_two$OD280_OD315_of_diluted_wines),
                              mean(class_three$OD280_OD315_of_diluted_wines)),

                          Mean_Proline = c(mean(wines_data$Proline),
                                           mean(class_one$Proline),
                                           mean(class_two$Proline),
                                           mean(class_three$Proline)))
average_table
```

```
  id class_number Mean_Alcohol Mean_Malic_acid Mean_Ash Mean_Alcalinity_of_ash
1  1      overall     13.00062        2.336348 2.366517                19.49494
2  2            1     13.74475        2.010678 2.455593                17.03729
3  3            2     12.27873        1.932676 2.244789                20.23803
4  4            3     13.15375        3.333750 2.437083                21.41667
  Mean_Magnesium Mean_Total_phenols Mean_Flavanoids Mean_Nonflavanoid_phenols
1       99.74157           2.295112       2.0292697                 0.3618539
2      106.33898           2.840169       2.9823729                 0.2900000
3       94.54930           2.258873       2.0808451                 0.3636620
4       99.31250           1.678750       0.7814583                 0.4475000
  Mean_Proanthocyanins Mean_Color_intensity  Mean_Hue
1             1.590899             5.058090 0.9574494
2             1.899322             5.528305 1.0620339
3             1.630282             3.086620 1.0562817
4             1.153542             7.396250 0.6827083
  Mean_OD280_OD315_of_diluted_wines Mean_Proline
1                         2.611685     746.8933
2                         3.157797    1115.7119
3                         2.785352     519.5070
4                         1.683542     629.8958
```

e

Since there are three different classes, one will need to do 3 comparisons, class 1 vs. class 2, class 1 vs class 3 and class 2 vs class 3. Firstly, extracting the data of level of phenols of each classes:

```
class_one_phenols <- class_one['Total_phenols']
class_two_phenols <- class_two['Total_phenols']
class_three_phenols <- class_three['Total_phenols']
```

For existing R function.

```
t_test_1_2 <- t.test(class_one_phenols, class_two_phenols)
t_test_1_2
```

```

    Welch Two Sample t-test

data:  class_one_phenols and class_two_phenols
```

```
t = 7.4206, df = 119.14, p-value = 1.889e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.4261870 0.7364055
sample estimates:
mean of x mean of y
 2.840169  2.258873
```

```r
t_test_1_3 <- t.test(class_one_phenols, class_three_phenols)
t_test_1_3
```

```
    Welch Two Sample t-test

data:  class_one_phenols and class_three_phenols
t = 17.12, df = 98.356, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.026801 1.296038
sample estimates:
mean of x mean of y
 2.840169  1.678750
```

```r
t_test_2_3 <- t.test(class_two_phenols, class_three_phenols)
t_test_2_3
```

```
    Welch Two Sample t-test

data:  class_two_phenols and class_three_phenols
t = 7.0125, df = 116.91, p-value = 1.622e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.4162855 0.7439610
sample estimates:
mean of x mean of y
 2.258873  1.678750
```

For manually conducting the t-test.

Then, calculating the mean, variance for each groups:

```r
mean_one <- mean(class_one_phenols[,])
mean_one
```

```
[1] 2.840169
```

```r
variance_one <- var(class_one_phenols[,])
variance_one
```

```
[1] 0.1148948
```

```r
mean_two <- mean(class_two_phenols[,])
mean_two
```

```
[1] 2.258873
```

```r
variance_two <- var(class_two_phenols[,])
variance_two
```

```
[1] 0.2974187
```

```r
mean_three <- mean(class_three_phenols[,])
mean_three
```

```
[1] 1.67875
```

```r
variance_three <- var(class_three_phenols[,])
variance_three
```

```
[1] 0.1274282
```

For different comparisons, assuming that the variances are different, first compute the t-statistics with formula: $t = \frac{(\hat{X}_1 - \hat{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$, where $\hat{X}_1$ and $\hat{X}_2$ are the sample means, μ_1 and μ_2 are the means, S_1^2 and S_2^2 are the sample variances, n_1 and n_2 are the sizes.

Since the null hypothesis is that there is no difference between each class, $\mu_1 - \mu_2 = 0$, thus the t-statistics are:

```
t_1_2 <- (mean_one - mean_two) /
  (sqrt((variance_one / num_class_one) + variance_two / num_class_two))
t_1_2
```

```
[1] 7.420649
```

```
t_1_3 <- (mean_one - mean_three) /
  (sqrt((variance_one / num_class_one) + variance_three / num_class_three))
t_1_3
```

```
[1] 17.12025
```

```
t_2_3 <- (mean_two - mean_three) /
  (sqrt((variance_two / num_class_two) + variance_three / num_class_three))
t_2_3
```

```
[1] 7.012505
```

Next the degrees of freedom are defined as $\nu = \frac{\left(\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}\right)^2}{\frac{\left(\frac{S_1^2}{n_1}\right)}{n_1-1}+\frac{\left(\frac{S_2^2}{n_2}\right)}{n_2-1}}$, then rounding it down to find the degree of freedom. The results are:

```
nu_1_2 <- floor(
  (variance_one / num_class_one + variance_two / num_class_two)^2 /
    ((variance_one / num_class_one)^2 / (num_class_one - 1) +
       (variance_two / num_class_two)^2 / (num_class_two - 1))
  )
nu_1_2
```

```
[1] 119
```

```
nu_1_3 <- floor(
  (variance_one / num_class_one + variance_three / num_class_three)^2 /
    ((variance_one / num_class_one)^2 / (num_class_one - 1) +
       (variance_three / num_class_three)^2 / (num_class_three - 1))
  )
nu_1_3
```

```
[1] 98
```

```
nu_2_3 <- floor(
  (variance_two / num_class_two + variance_three / num_class_three)^2 /
    ((variance_two / num_class_two)^2 / (num_class_two - 1) +
       (variance_three / num_class_three)^2 / (num_class_three - 1))
  )
nu_2_3
```

```
[1] 116
```

Define a function to manually compute the p-value of give t-statistics and degree of freedom:

```
compute_two_tail_p_value <- function(t_statistics, df){
  #inputs: t_statistics : the t-statistics, df: the degree of freedom
  #outputs: the p-value
  t_pdf <- function(x, df){
    # inputs x: variable, df: the degree of freedom
    # output the value of the probability density function
    return(gamma((df+1)/2) / (sqrt(df*pi) * gamma(df/2)) * (1 + (x^2)/df)^(-(df+1)/2))
  }

  p_value_two_tail <- 2 * integrate(t_pdf, t_statistics, Inf, df = df)$value

  return(p_value_two_tail)
}
p_1_2 <- compute_two_tail_p_value(t_1_2, nu_1_2)
p_1_2
```

```
[1] 1.897952e-11
```

```
p_1_3 <- compute_two_tail_p_value(t_1_3, nu_1_3)
p_1_3
```

```
[1] 3.267661e-31
```

```
p_2_3 <- compute_two_tail_p_value(t_2_3, nu_2_3)
p_2_3
```

```
[1] 1.664716e-10
```

Through calculation, one can observe that the p-values of all three comparisons are extremely small. Thus one can argue that there is extremely strong evidence against the null hypothesis for each pairwise comparison. The differences in phenol levels between all the classes are statistically significant.