Q1

In part 1 of this question, we had to use three methods to load data. When using scan function, we used what=character() to interpret all data as text. This was required since the data used had numbers and texts both while scan function expects the data to be of single type.

Since the result of the load function must be identical to the iris data given in R, we also removed the first column of ‘Sample index’ from the data.

The results from all three functions were identical.

Table

Description automatically generated

Part two of the question required to test the function produced in part 1 to show if the results of all three methods were identical.

Graphical user interface, text, application

Description automatically generated

The results were same for all three methods.

Q2) In Question 2, I used help from tutorial 5 to clean the data. We first looked for errors in text using unique() function. There were no textual errors in the data.



We then used summary() function to look for high values and negative values.

A picture containing text, receipt

Description automatically generated

Summary showed negative values as well as high values greater than 10.

We then plotted the data to look for outliers using the below code:

plot(Ndata[, -5], main = "Iris data with bad values",

pch =21, col= c('red', 'yellow', 'green4', 'blue'))

A picture containing diagram

Description automatically generated

The plot showed some outliers. We also used boxplots to confirm the outliers in the data.

par(mfrow = c(1,1))

boxplot(Ndata$Sepal.Length, Ndata$Sepal.Width, Ndata$Petal.Length, Ndata$Petal.Width,

main = "Boxplot of Iris data with bad values", xlab = 'Measurements',

col = 3:6)

legend(0.5, 28, inset=0, title = 'Legend',

c("Sepal Length", "Sepal Width", "Petal Length", "Petal width"),

fill = 3:6, cex = 0.8)

Chart, box and whisker chart

Description automatically generated

The boxplot confirms the presence of outliers.

We then created a function clean.indices() and used a block code to find the bad values in the data.

clean.indices <- function(x)

{

missval <- any(is.na(x))

negval <- any( x <= 0)

larval <- any( x > 10)

res <- missval | negval | larval

return(!res)

}

We used any() function to find any NA values, negative values, and large values in the data. We used 10 as a reference number for large values. After that we applied the filter to clean bad values from the data.

filter <- apply( Ndata[-5], 1, clean.indices

cleaned <- Ndata [filter,]

The summary and plotted data after cleaning showed improved data quality with no outliers or negative values.

A picture containing text

Description automatically generated

A picture containing calendar

Description automatically generated

We also produced a boxplot for cleaned data.

Chart, box and whisker chart

Description automatically generated

Q3i)

For Question 3, we were asked to create a repair data for the bad values in iris data. We decided to use median values to be used as replacement values for all bad values since median values are middle values and they do not affect the mean values or range of a dataset. We created a function to find median values of all measurements of the three species and combined them in a matrix form. We used build.repair.data() to do this.

A picture containing text

Description automatically generated

For the second and third part of the question where we were supposed to replace the values in the data with median values, I was not able to produce repaired data due to an unidentified error in my code for part 2. I have written the codes for both repaired data as well as the codes to produce plots and boxplots for comparison. The graphs would be plotted to compare the cleaned data and repaired data.

Q4i)

For part 1 of this question, we could use only apply, summary, cat function to produce the summary. Below code was used to produce the output:

apply(iris[1:4], 2, summary)

Text

Description automatically generated

To produce the boxplot as shown in the question, below code was used:

D <- apply(iris[1:4], 2, summary)

boxplot(D, main= "Measuerments using all species",

col= c('Red', 'Green', 'Blue', 'Skyblue'))

Chart, box and whisker chart

Description automatically generated

Q4ii)

For part 2 of the question, we were advised to generate the outputs as shown and structure the code in a way that each output is produced by a single function. Therefore, two functions in total.To produce the textual output, below block code was used:

spec.summary <- function()

{

Setos <- subset(iris, Species == "setosa")

Versic <- subset(iris, Species == "versicolor")

Virgini <- subset(iris, Species == "virginica")

Setosa <- apply(Setos[1:4], 2, summary)

cat('Species = Setosa', '\n', sep='')

print(Setosa)

Versicolor <- apply(Versic[1:4], 2, summary)

cat('Species = Versicolor', '\n', sep='')

print(Versicolor)

Virginica <- apply(Virgini[1:4], 2, summary)

cat('Species = Virginica', '\n', sep='')

print(Virginica)

}

spec.summary()

A picture containing text, receipt

Description automatically generated

To produce boxplots together in one frame, following function was used:

mul.bplots <- function()

{

par(mfrow=c(2,2))

boxplot(iris$Sepal.Length ~ iris$Species,

main = 'Sepal.length', xlab = '', ylab = '',

col = c('red', 'green', 'blue'))

boxplot(iris$Petal.Length ~ iris$Species,

main = 'Petal.length', xlab = '', ylab = '',

col = c('red', 'green', 'blue'))

boxplot(iris$Sepal.Width ~ iris$Species,

main = 'Sepal.Width', xlab = '', ylab = '',

col = c('red', 'green', 'blue'))

boxplot(iris$Petal.Width ~ iris$Species,

main = 'Petal.Width', xlab = '', ylab = '',

col = c('red', 'green', 'blue'))

}

mul.bplots()

The output:

Diagram, box and whisker chart

Description automatically generated