21BDS0340

Abhinav Dinesh Srivatsa

Exploratory Data Analysis Lab

Assignment – II

**Experiment 5**

**Code:**

library(dplyr)

library(missForest)

library(mice)

library(VIM)

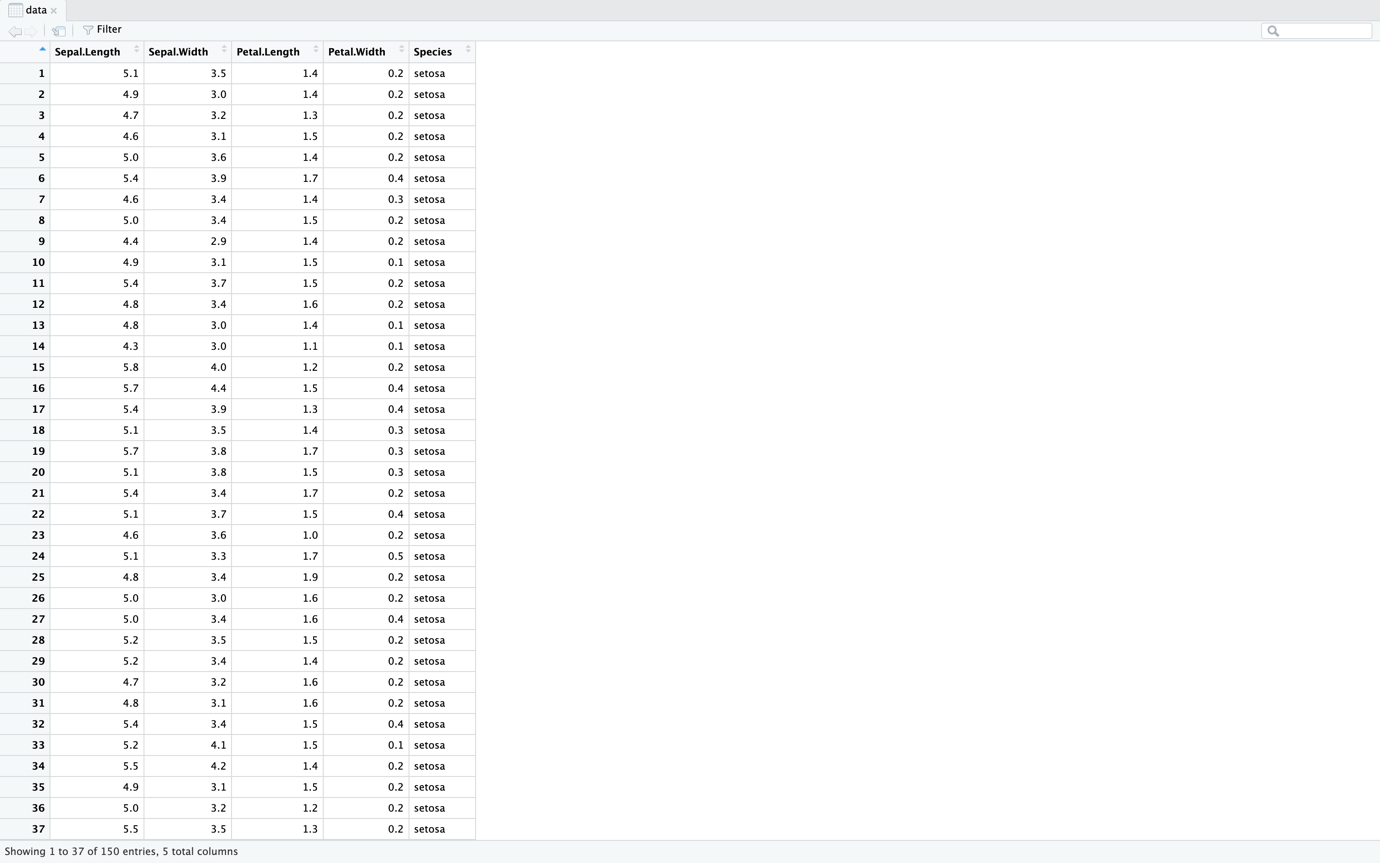
library(ggplot2)

library(cowplot)

data = iris

View(data)

**Output:**



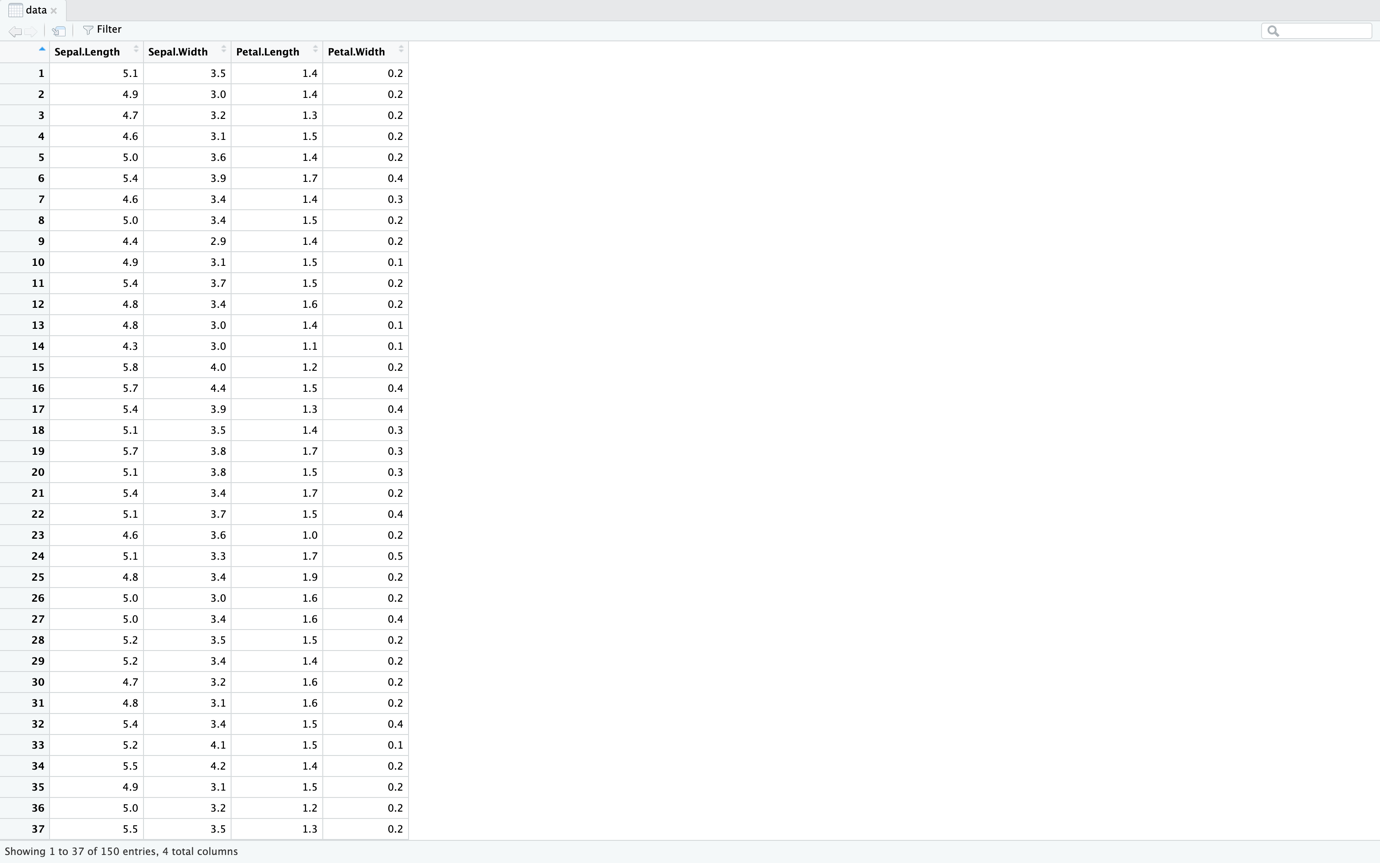
**Code:**

# dropping labels

data = data %>% select(-c("Species"))

View(data)

**Output:**



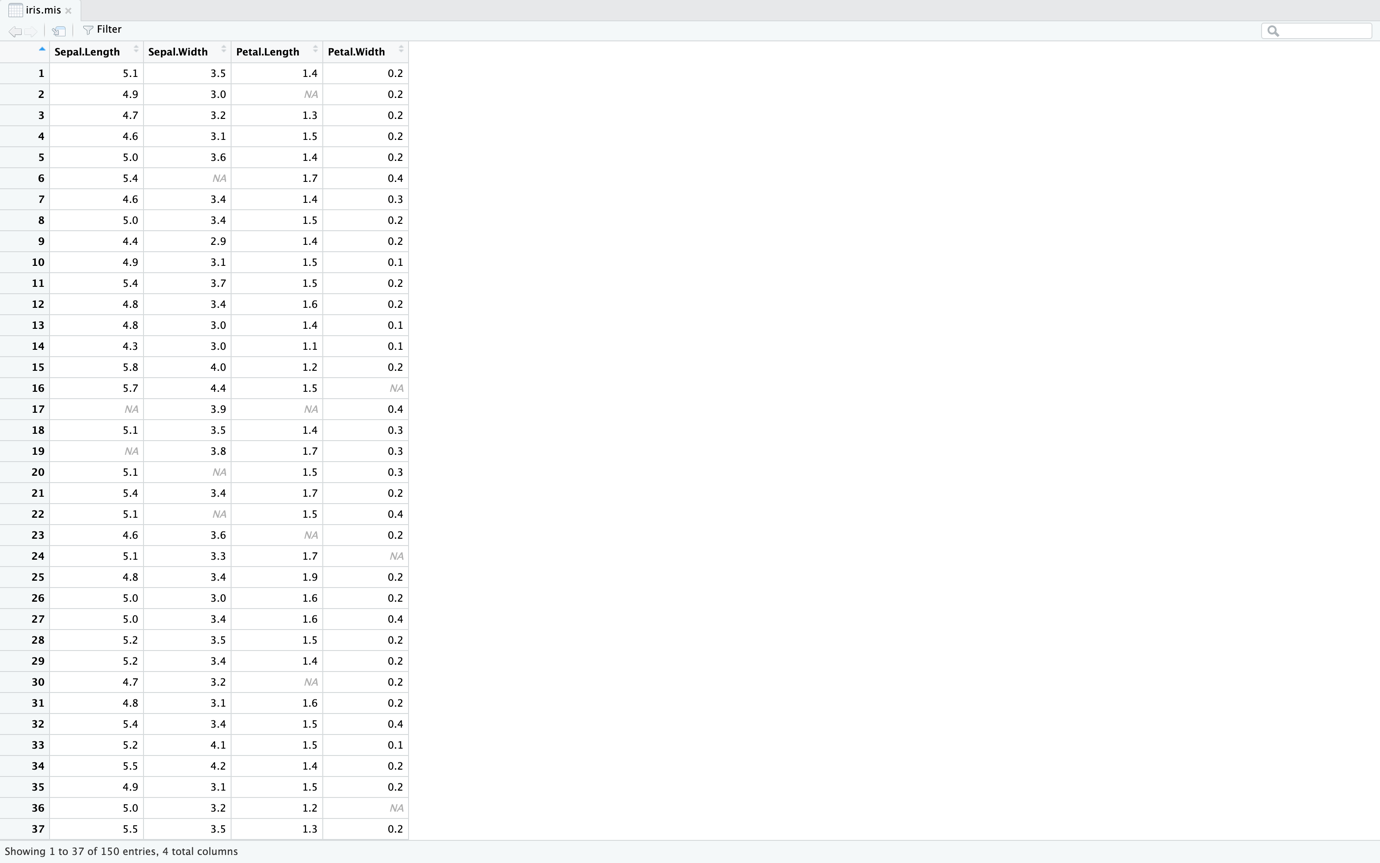
**Code:**

# adding 10% random values

iris.mis <- prodNA(data, noNA = 0.1)

View(iris.mis)

**Output:**

****

**Code:**

ggplot(iris.mis, aes(x = Sepal.Length)) +

geom\_histogram(color="black", fill="#0099F8")

**Output:**

**A graph of blue bars

Description automatically generated**

**Code:**

# simple imputations for Sepal.Length

imputed = data.frame(

Original = iris.mis$Sepal.Length,

Imp.Zero = replace(iris.mis$Sepal.Length, is.na(iris.mis$Sepal.Length), 0),

Imp.Mean = replace(iris.mis$Sepal.Length, is.na(iris.mis$Sepal.Length), mean(iris.mis$Sepal.Length, na.rm = TRUE)),

Imp.Median = replace(iris.mis$Sepal.Length, is.na(iris.mis$Sepal.Length), median(iris.mis$Sepal.Length, na.rm = TRUE))

)

# plotting the simple imputations

h1 = ggplot(imputed, aes(x=Original)) +

geom\_histogram(fill="red", color="black", position="identity") +

ggtitle("Original distribution")

h2 = ggplot(imputed, aes(x=Imp.Zero)) +

geom\_histogram(fill="green", color="black", position="identity") +

ggtitle("Zero-imputed distribution")

h3 = ggplot(imputed, aes(x=Imp.Mean)) +

geom\_histogram(fill="blue", color="black", position="identity") +

ggtitle("Mean-imputed distribution")

h4 = ggplot(imputed, aes(x=Imp.Median)) +

geom\_histogram(fill="yellow", color="black", position="identity") +

ggtitle("Median-imputed distribution")

plot\_grid(h1, h2, h3, h4, nrow=2, ncol=2)

**Output:**

**A group of different colored bars

Description automatically generated**

**Code:**

# viewing missing values

md.pattern(iris.mis, rotate.names=TRUE)

**Output:**

**A blue and pink squares

Description automatically generated**

**Code:**

# performing imputations with mice algorithms

mice\_imputed = data.frame(

Original = iris.mis$Sepal.Length,

Imp.PMM = complete(mice(iris.mis, method="pmm"))$Sepal.Length,

Imp.CART = complete(mice(iris.mis, method="cart"))$Sepal.Length,

Imp.Lasso = complete(mice(iris.mis, method="lasso.norm"))$Sepal.Length

)

# plotting the mice imputations

h1 = ggplot(mice\_imputed, aes(x=Original)) +

geom\_histogram(fill="red", color="black", position="identity") +

ggtitle("Original distribution")

h2 = ggplot(mice\_imputed, aes(x=Imp.PMM)) +

geom\_histogram(fill="green", color="black", position="identity") +

ggtitle("PMM-imputed distribution")

h3 = ggplot(mice\_imputed, aes(x=Imp.CART)) +

geom\_histogram(fill="blue", color="black", position="identity") +

ggtitle("CART-imputed distribution")

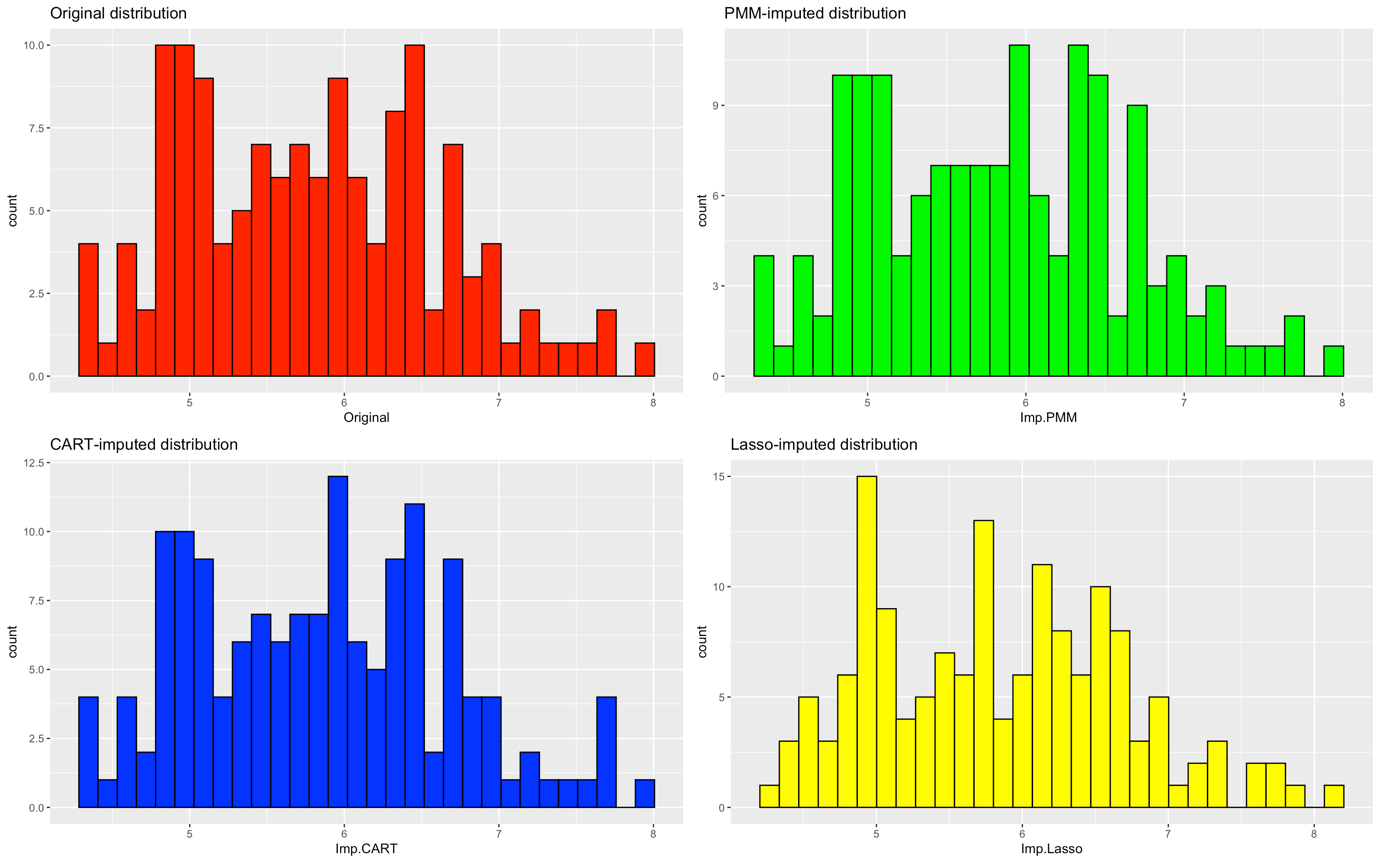
h4 = ggplot(mice\_imputed, aes(x=Imp.Lasso)) +

geom\_histogram(fill="yellow", color="black", position="identity") +

ggtitle("Lasso-imputed distribution")

plot\_grid(h1, h2, h3, h4, nrow=2, ncol=2)

**Output:**

****

**Code:**

# imputations with missForest

missforest\_imputed = data.frame(

Original = iris.mis$Sepal.Length,

Imp.Missforest = missForest(iris.mis)$ximp$Sepal.Length

)

# plotting the missForest imputations

h1 = ggplot(missforest\_imputed, aes(x=Original)) +

geom\_histogram(fill="red", color="black", position="identity") +

ggtitle("Original distribution")

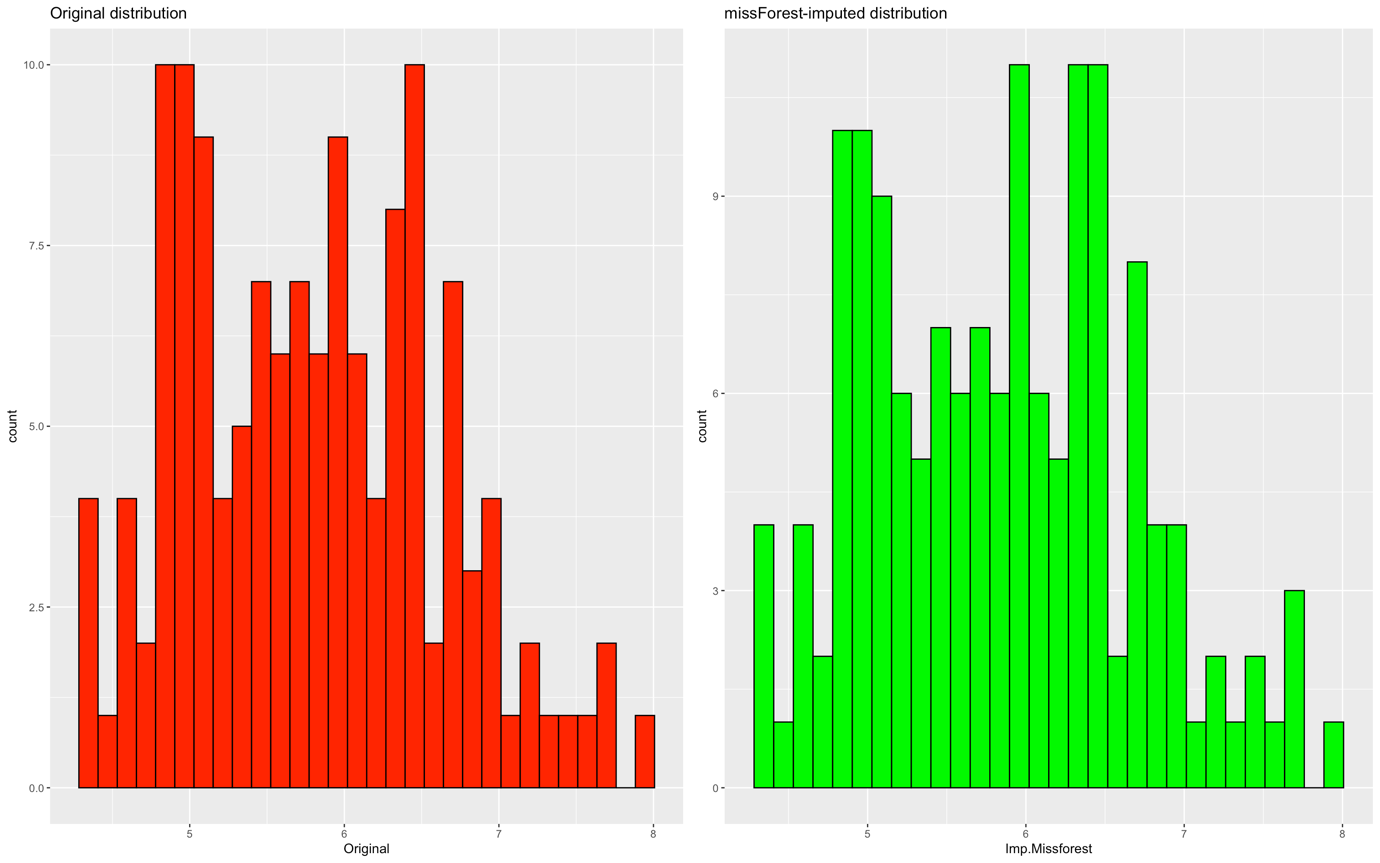
h2 = ggplot(missforest\_imputed, aes(x=Imp.Missforest)) +

geom\_histogram(fill="green", color="black", position="identity") +

ggtitle("missForest-imputed distribution")

plot\_grid(h1, h2, nrow=1, ncol=2)

**Output:**

****

**Experiment 6**

**Code:**

X = scan(what = numeric())

Y = scan(what = numeric())

cov(X, Y)

cor(X, Y)

**Output:**

> X = scan(what = numeric())

1: 1 2 3 4 5 6 7 8 9 10

11:

Read 10 items

> Y = scan(what = numeric())

1: 9 3 4 1 5 6 2 7 10 9

11:

Read 10 items

> cov(X, Y)

[1] 3.777778

> cor(X, Y)

[1] 0.3981308

**Experiment 7**

**Code:**

# z score method

data = c(10, 10, 11, 11, 12, 12, 13, 12, 11, 14, 13, 15, 100)

mean.data = mean(data)

std.data = sd(data)

z.scores = (data - mean.data) / std.data

# outliers have -3 < z.score < 3

outliers = data[abs(z.scores) > 3]

outliers

**Output:**

> # z score method

> data = c(10, 10, 11, 11, 12, 12, 13, 12, 11, 14, 13, 15, 100)

>

> mean.data = mean(data)

> std.data = sd(data)

>

> z.scores = (data - mean.data) / std.data

>

> # outliers have -3 < z.score < 3

> outliers = data[abs(z.scores) > 3]

> outliers

[1] 100

**Code:**

# inter quartile range method

data = c(10, 10, 11, 11, 12, 12, 13, 12, 11, 14, 13, 15, 100)

q1 = quantile(data, 0.25)

q3 = quantile(data, 0.75)

iqr = q3 - q1

# outliers lie outside of the inter quartile range

outliers <- data[data < q1 | data > q3]

outliers

**Output:**

> # inter quartile range method

> data = c(10, 10, 11, 11, 12, 12, 13, 12, 11, 14, 13, 15, 100)

>

> q1 = quantile(data, 0.25)

> q3 = quantile(data, 0.75)

> iqr = q3 - q1

>

> # outliers lie outside of the inter quartile range

> outliers <- data[data < q1 | data > q3]

> outliers

[1] 10 10 14 15 100

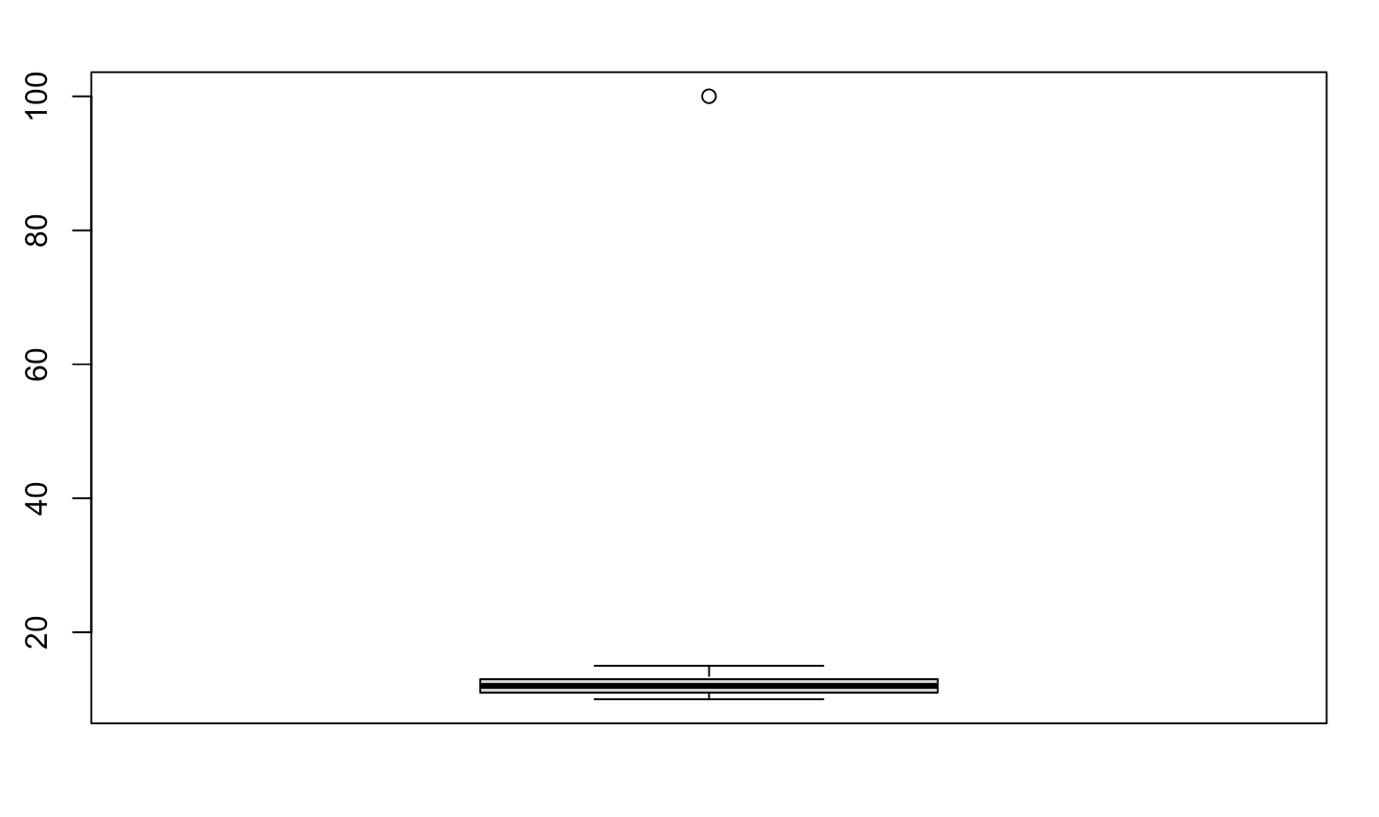
**Code:**

# boxplot method (purely visualisation)

data = c(10, 10, 11, 11, 12, 12, 13, 12, 11, 14, 13, 15, 100)

boxplot(data)

**Output:**

****