Experiment 10

Student Name: Mohit Kumar

UID: 20BCS9473

Branch: CSE

Section/Group: 20 BCS-DM-714-A

Subject Name: DATA MINING LAB

Aim:

Outlier detection using R programming.

Theory and Output:

Outlires

In R, the IQR() function is used to compute the interquartile range of a given object of numerical values. The interquartile range of these values is a range where 25% on either side is cut off. Statistically, the interquartile range is the difference between the upper quartile and the lower quartile.

Then, we calculate the interquartile range (IQR) using the IQR() function in R.

The quantile function divides the data into equal halves, in which the median acts as middle and over that the remaining lower part is lower quartile and upper part is upper quartile.

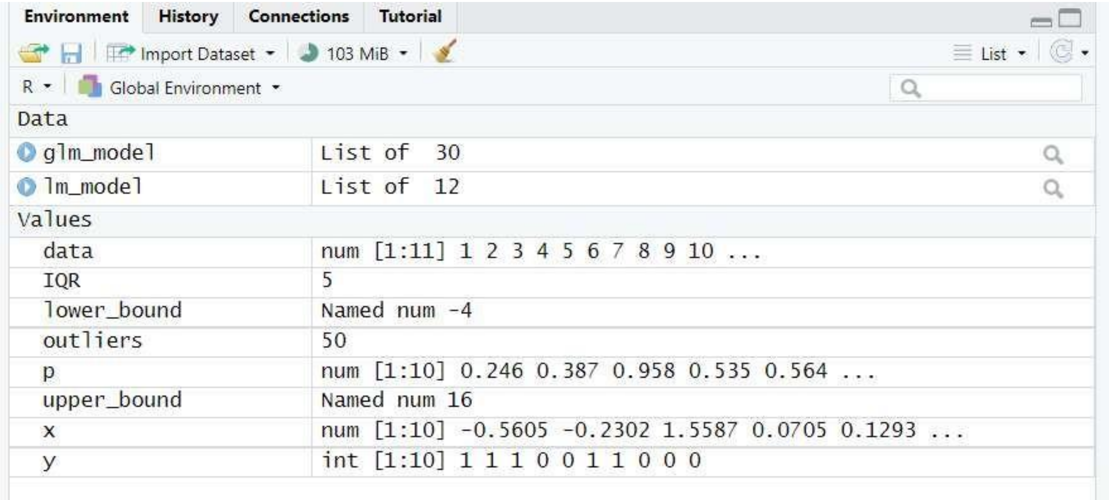
Next, we use the Tukey method to calculate the lower and upper bounds. Any data points that fall below the lower bound or above the upper bound are considered outliers. Finally, we print the lower and upper bounds as well as any identified outliers.

Note that this is just one method for outlier detection and there are many other methods that can be used depending on the specific needs of your analysis.

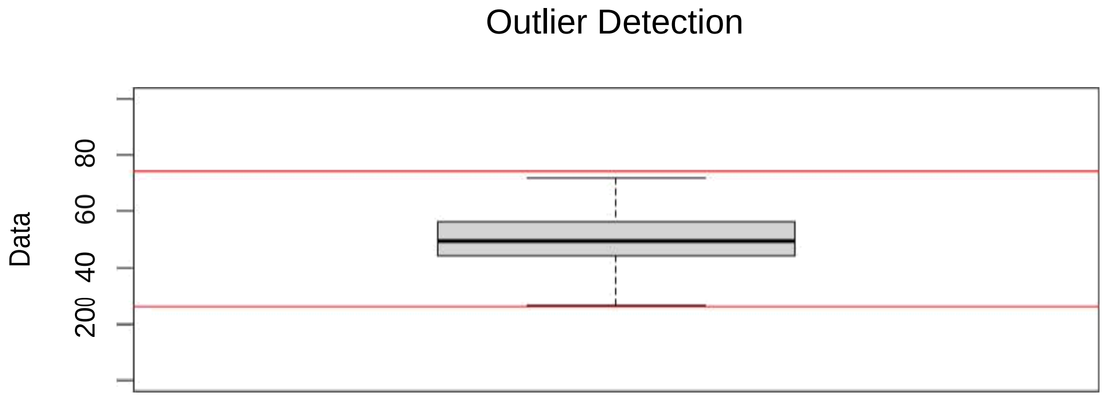
```
# Outlier Program
data <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50)
# Calculate interquartile range
IQR<- IQR(data)
# Calculate lower and upper bounds using Tukey method
lower bound <- quantile(data, 0.25) - 1.5 * IQR upper bound
<- quantile(data, 0.75) + 1.5 * IQR
# Identify outliers
outliers <- data[data < lower bound | data > upper bound]
#Print results
```

```
cat("Lower bound:", lower bound, "\n")
cat("Upper bound:", upper bound, "\n")
cat("Outliers:", outliers, "\n")
# Create a vector of random data
data <- rnorm(100, mean = 50, sd = 10)

# Calculate the quartiles and interquartile range
q1 <- quantile(data, 0.25)
q3 <- quantile(data,
0.75) iqr <- q3 - q1

# Calculate the lower and upper bounds for outliers
lower <- q1 - 1.5 * iqr
upper <- q3 + 1.5 * iqr

# Create a box plot of the data
boxplot(data, main = "Outlier Detection", ylim = c(0, 100), ylab = "Data")
ab1ine(h = lower, col = "red")
ab1ine(h = upper, col = "red")

# Identify the outliers and plot them as points
outliers <- data[data < lower | data > upper]
points(rep(1, length(outliers)), outliers, col = "red", pch = 19)
```

Output:



```
Data
glm_model       List of  30
lm_model        List of  12
Values
  data          num [1:11] 1 2 3 4 5 6 7 8 9 10 ...
  IQR           5
  lower_bound   Named num -4
  outliers      50
  p             num [1:10] 0.246 0.387 0.958 0.535 0.564 ...
  upper_bound   Named num 16
  x             num [1:10] -0.5605 -0.2302 1.5587 0.0705 0.1293 ...
  y             int [1:10] 1 1 1 0 0 1 1 0 0 0
```


Outlier Detection
Data
200
40
60
80