Project IV: PageRank and Anomaly Detection

Appiah Prince* University of Texas at El Paso (UTEP)

October 25, 2022

Contents

*pappiah@miners.utep.edu

1 PageRank

1.1 Obtain the link matrix L and input it into R

```
L <- matrix(c(0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 1, 1, 0, 0,
1, 0, 0, 0, 0, 1, 0,
1, 0, 1, 0, 1, 1, 0,
0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1, 0), nrow = 7, ncol = 7, byrow = F)
colnames(L)<-c("A","B","C","D","E","F","G")
row.names(L)<-c("A","B","C","D","E","F","G")

L
```

```
##   A B C D E F G
## A 0 0 1 1 0 0 0
## B 1 0 0 0 0 0 0
## C 0 0 0 1 0 0 0
## D 0 1 0 0 1 0 0
## E 0 1 0 1 0 0 1
## F 0 0 1 1 0 0 1
## G 0 0 0 0 0 0 0
```

Comments

- L is a 7 x 7 matrix
- Webpage G is a dead end since there is no outlink from it.

1.2 Reproduce the graph similar to Figure 1 to check if you have got the right link matrix L.

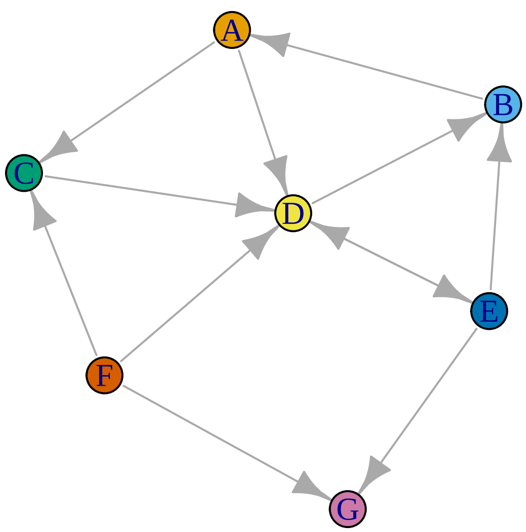
```
set.seed(12333333)
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
graph <- graph_from_adjacency_matrix(L)
par(mfrow=c(1,1), mar=rep(4,4))
plot(graph, vertex.color=c(1,2,3,4,5,6,7))
```



Comment

- The plot is the same as the given plot. Hence, our link matrix L is correct.

1.3 Compute the PageRank score for each webpage. Provide a barplot of the PageRank score. Which pages come to the top-3 list? Discuss the results.

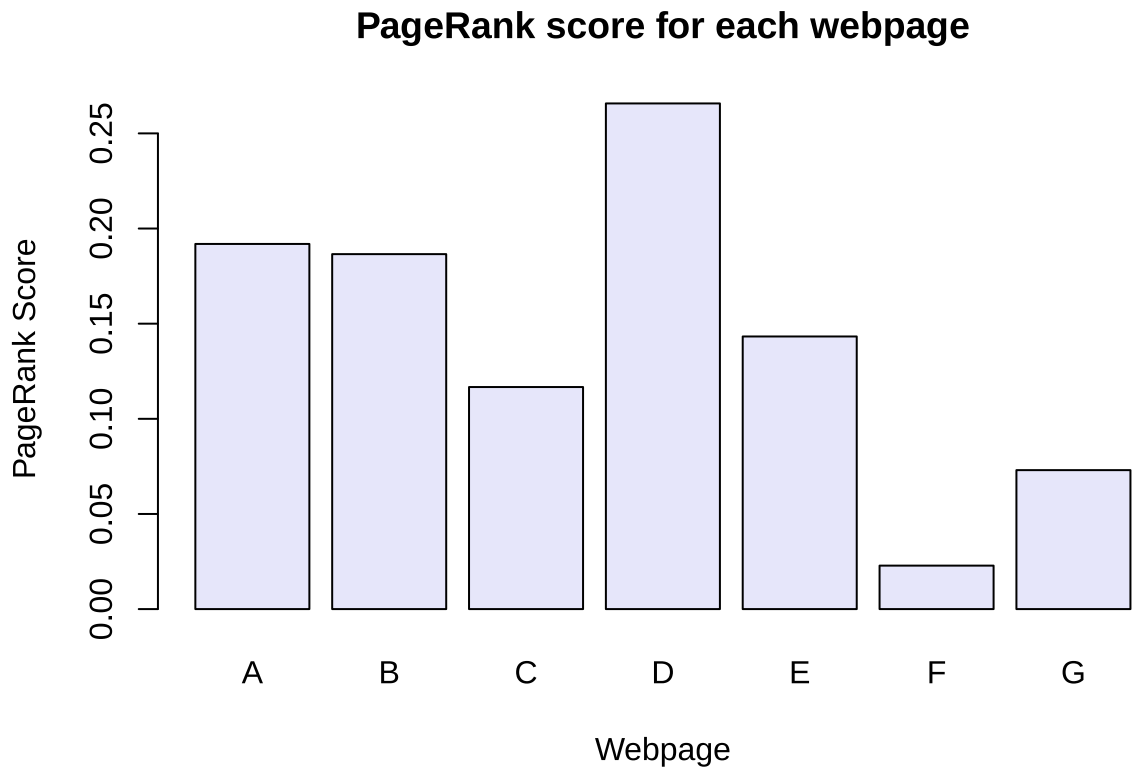
```
pagerank <- function(G, method='eigen',d=.85,niter=100){
  cvec <- apply(G,2,sum)
  cvec[cvec==0] <- 1
  n <- nrow(G)
  delta <- (1-d)/n
  A <- matrix(delta,nrow(G),ncol(G))
  for (i in 1:n)   A[i,] <- A[i,] + d*G[i,]/cvec
```

```r
  if (method=='power'){
    x <- rep(1,n)
    for (i in 1:niter) x <- A%*%x
  } else {
    x <- Re(eigen(A)$vector[,1])
  }
  x/sum(x)
}
```

```r
#PageRank score for each webpage
L0 <- t(L)
pg <- pagerank(L0, method='power')
pg <- data.frame("WebPage"= c("A","B","C","D","E","F","G"), "PageRank"= pg)
pg
```

```
##   WebPage   PageRank
## 1       A 0.19189501
## 2       B 0.18653569
## 3       C 0.11670092
## 4       D 0.26573770
## 5       E 0.14325934
## 6       F 0.02284669
## 7       G 0.07302466
```

```r
#Barplot of the PageRank score.
barplot(pg$PageRank, names= pg$WebPage, col="lavender", xlab="Webpage",
        ylab="PageRank Score", main="PageRank score for each webpage")
```



Comment

- Webpage D has the highest score which indicates that most people visit it.
- Webpage F has the lowest score.

```
#The top-3 list of the PageRank score
top3 <- pg[ order(pg$PageRank, decreasing = TRUE), ]
head(top3, 3)
```

```
##   WebPage  PageRank
## 4       D 0.2657377
## 1       A 0.1918950
## 2       B 0.1865357
```

Comment

- The top-3 list of the pagerank score in descending order are D,A,B.

2 Anomaly Detection

We consider the HTP (high tech part) data available from R Package ICSOutlier. This data set contains the results of p = 88 numerical tests for n = 902 high-tech parts. Based on these results the producer considered all parts functional and all of them were sold. However two parts, 581 and 619, showed defects in use and were returned to the manufacturer. These two observations can thus be considered as outliers and the objective is to detect them by re-examining the test data.

2.1 Bring in the data with the following R code

```
#install.packages("ICSOutlier")
        library("ICSOutlier")
```

```
## Loading required package: ICS

## Loading required package: mvtnorm

## Loading required package: moments
```

```
        data(HTP)
        dat <- HTP; dim(dat); #head(dat)
```

```
## [1] 902  88
```

```
        outliers.true <- c(581, 619)
```

Comment

- The dimension of the data is 902 observations with 88 variables.

2.2 Part b

2.2.1 Obtain MCD estimates

```
# Obtain MCD estimates with a breakdown point of 20%
library(robustbase)
fit.robust <- covMcd(dat, cor = FALSE, alpha = 0.80)
```

Comment

- A breakdown of 20% has been used.

2.2.2 Robust estimates of the mean vector

```r
# Robust estimates of the mean vector for 5 variables
Mean_vector <- fit.robust$center
Mean_vector[1:5]
```

```
##           V.1           V.2           V.3           V.4           V.5
##  1.679338e-05 -3.970916e-07  6.971464e-06 -3.967705e-07 -3.760885e-07
```

2.2.3 Robust estimates of the VCOV matrix

```r
Cov_matrix <- fit.robust$cov
Cov_matrix[1:5, 1:5]
```

```
##               V.1           V.2           V.3           V.4           V.5
## V.1  1.542990e-07 -4.342807e-09  1.794573e-07 -4.296804e-09 -4.295646e-09
## V.2 -4.342807e-09  1.025305e-09 -5.294162e-09  1.018760e-09  1.012309e-09
## V.3  1.794573e-07 -5.294162e-09  2.101122e-07 -5.239670e-09 -5.236121e-09
## V.4 -4.296804e-09  1.018760e-09 -5.239670e-09  1.012768e-09  1.006158e-09
## V.5 -4.295646e-09  1.012309e-09 -5.236121e-09  1.006158e-09  1.000130e-09
```

2.2.4 Robust (squared) Mahalanobis distance

```r
Mahalanobis_Dist <- mahalanobis(dat,Mean_vector,Cov_matrix)
head(Mahalanobis_Dist)
```

```
## [1]  96.91542 205.18058  86.25564  75.77243  76.46848  74.63073
```

```r
# Cut-off based on the chi-square distribution
cutoff.chi.sq <- qchisq(0.975, df = ncol(dat))
cutoff.chi.sq
```

```
## [1] 115.8414
```

Comment

- I used a threshold $p = 0.975$ for the the chi-square distribution.

```r
# Another Cut-off Suggested by Green and Martin (2014)
library("CerioliOutlierDetection")
n <- nrow(dat)
p <- ncol(dat)
cutoff.GM <- hr05CutoffMvnormal(n.obs = n, p.dim=p, mcd.alpha = 0.75,
    signif.alpha = 0.025, method = "GM14",
    use.consistency.correction = TRUE)$cutoff.asy
cutoff.GM
```

```
## [1] 149.9075
```

Comment

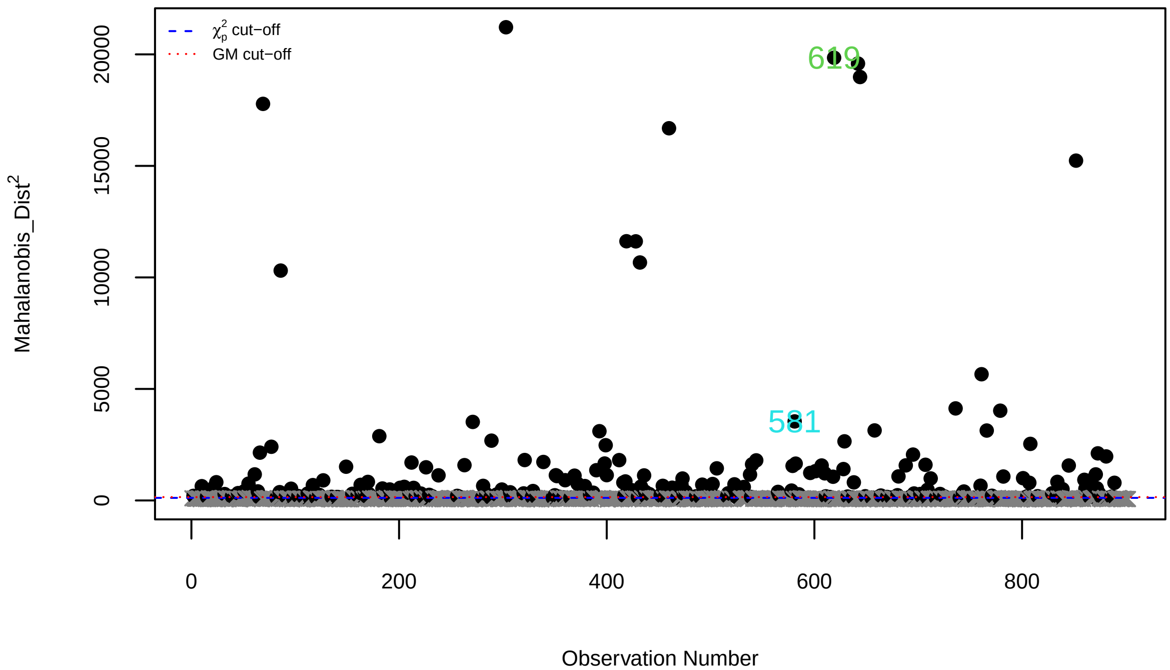
- I used a threshold of $\alpha = 0.025$ or 0.025 level of significance.

2.2.5 Plot the results

```r
colPoints <- ifelse(Mahalanobis_Dist >= min(c(cutoff.chi.sq, cutoff.GM)), 1,
                    grey(0.5))
pchPoints <- ifelse(Mahalanobis_Dist >= min(c(cutoff.chi.sq, cutoff.GM)), 16, 4)

plot(seq_along(Mahalanobis_Dist), Mahalanobis_Dist, pch = pchPoints,
     col = colPoints,ylim=c(0, max(Mahalanobis_Dist, cutoff.chi.sq, cutoff.GM) + 2),
     xlab = "Observation Number")

abline(h = c(cutoff.chi.sq, cutoff.GM), lty = c("dashed", "dotted"), col=c("blue", "red"
legend("topleft", lty = c("dashed", "dotted"), cex = 0.5, ncol = 1, bty = "n",
legend = c(expression(paste(chi[p]**2, " cut-off")), "GM cut-off"), col=c("blue", "red")
text(619, Mahalanobis_Dist[619], labels=619, col=619)
text(581, Mahalanobis_Dist[581], labels=581, col=581)
```



Comment

- From the graph, we see that the observations *581* and *619* are indeed outliers.
- Observation *619* may be in the top list of potential outliers while *581* may not.

2.3 Part c

2.3.1 Isolation forest (iForest)

```
# Since I use MACBOOK I decided to try the R PACKAGE IsolationForest which you
# stated in your R code in class.

library(IsolationForest)
```

```
## IsolationForest 0.0-26
```

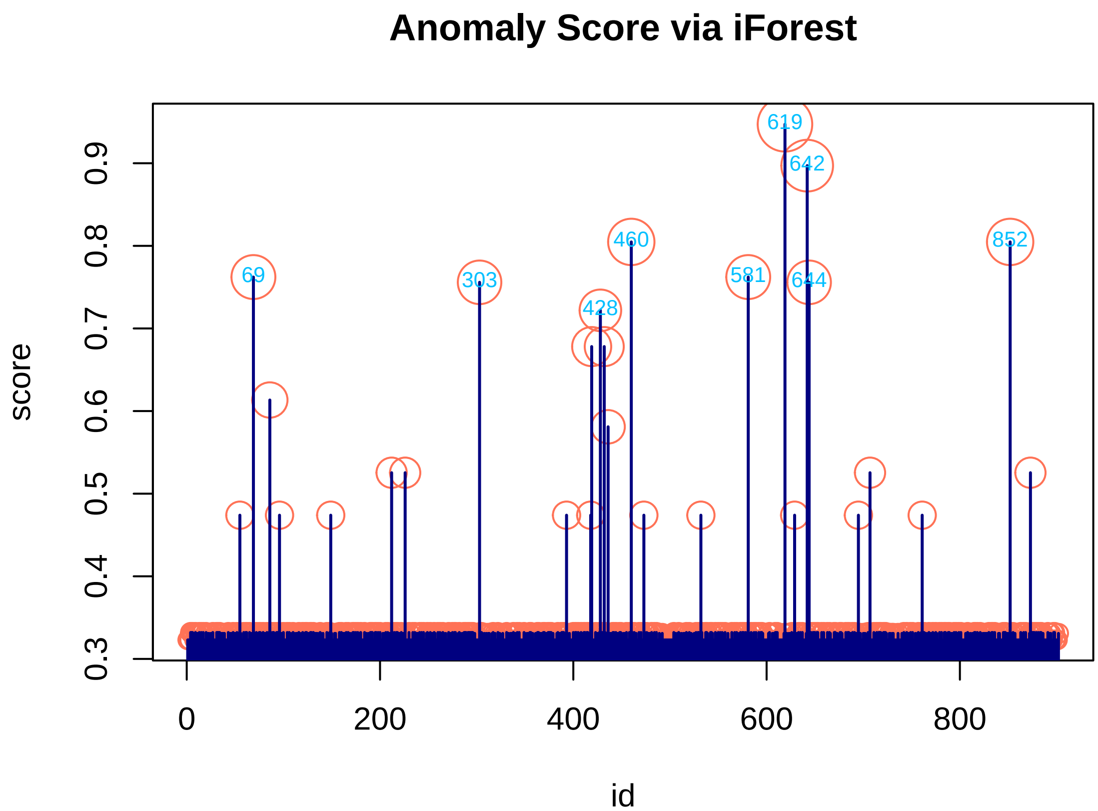
```
iso_tree <- IsolationTrees(dat, rFactor=0, ntree = 80)
anomaly_score <- AnomalyScore(dat,iso_tree)
Ascore <- anomaly_score$outF

# PLOT OF THE SCORES
par(mfrow=c(1,1), mar=rep(4,4))
plot(x=1:length(Ascore), Ascore, type="p", pch=1,
```

```
    main="Anomaly Score via iForest",
        xlab="id", ylab="score", cex=Ascore*4, col="coral1")
add.seg <- function(x) segments(x0=x[1], y0=0, x1=x[1], y1=x[2],
    lty=1, lwd=1.5, col="navyblue")
apply(data.frame(id=1:length(Ascore), score=Ascore), 1, FUN=add.seg)
```

```
## NULL
```

```
eps <- 0.99
id.outliers <- which(Ascore > quantile(Ascore, eps))
text(id.outliers, Ascore[id.outliers]+0.003, label=id.outliers,
    col="deepskyblue1", cex=0.7)
```



Comment

- We used the parameters rFactor=0, ntree = 80 for the IsolationForest.
- We observe that the observations *581* and *619* are deemed anomalies.

2.3.2 Local Outlier Factor(LOF)

```
library(Rlof)
```

```
## Loading required package: doParallel

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

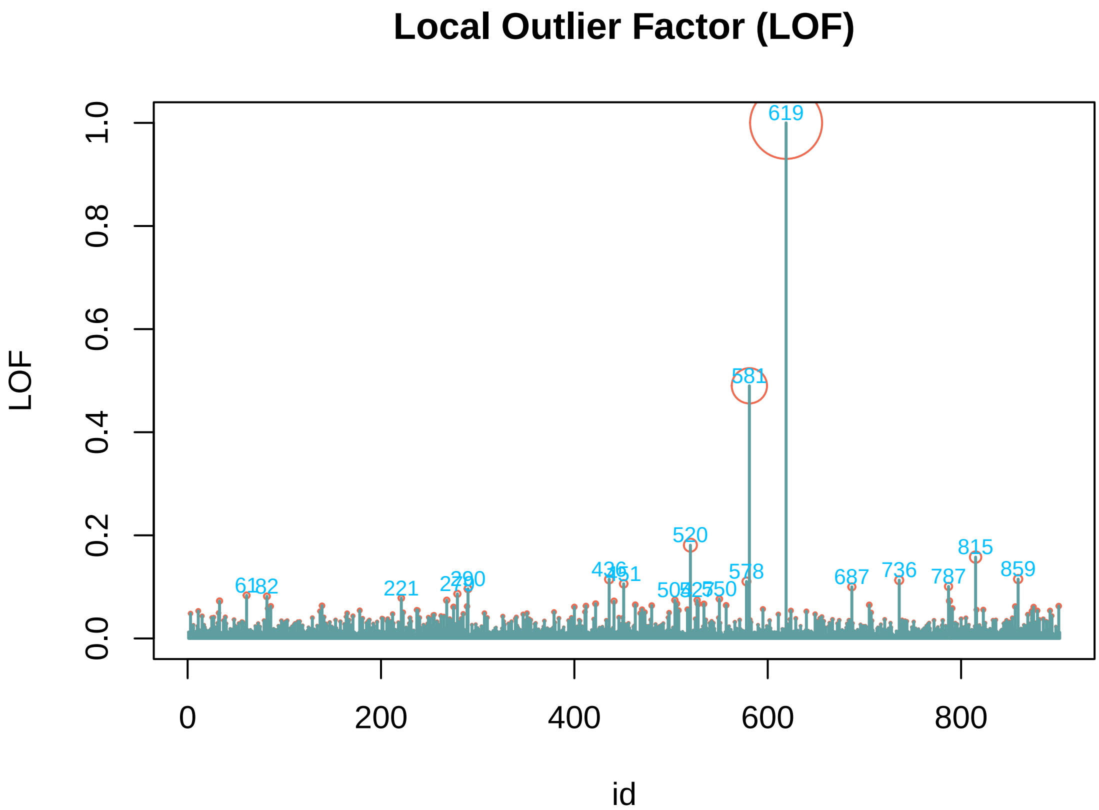
```
outlier.scores <- lof(dat, k=6)
which(outlier.scores > quantile(outlier.scores, 0.95))
```

```
##  [1]  33  61  82  83  86 139 221 268 275 279 289 290 400 412 422 436 441 451 463
## [20] 470 480 504 506 517 520 527 528 534 550 557 578 581 595 619 687 705 736 787
## [39] 788 791 815 823 856 859 875 901
```

```
# PLOT OF THE LOF SCORES
score <- scale(outlier.scores, center = min(outlier.scores),
    scale = max(outlier.scores)-min(outlier.scores))
par(mfrow=c(1,1), mar=rep(4,4))
plot(x=1:length(score), score, type="p", pch=1,
    main="Local Outlier Factor (LOF)",
        xlab="id", ylab="LOF", cex=score*5, col="coral2")
add.seg <- function(x) segments(x0=x[1], y0=0, x1=x[1], y1=x[2],
    lty=1, lwd=1.5, col="cadetblue")
apply(data.frame(id=1:length(score), score=score), 1, FUN=add.seg)
```

```
## NULL
```

```
eps <- 0.98
id.outliers <- which(outlier.scores > quantile(outlier.scores, eps))
text(id.outliers, score[id.outliers]+0.02, label=id.outliers,
    col="deepskyblue1", cex=0.7)
```



Comment

- We used k=6 that is, the 6th distance was used to calculate the LOFs.
- We observe that the observations *581* and *619* are deemed anomalies.

2.3.3 Comparison of the results of the two methods

```
par(mfrow=c(1,2), mar=rep(4,4))

#iForest
plot(x=1:length(Ascore), Ascore, type="p", pch=1,
    main="Anomaly Score via iForest",
        xlab="id", ylab="score", cex=Ascore*4, col="coral1")
add.seg <- function(x) segments(x0=x[1], y0=0, x1=x[1], y1=x[2],
    lty=1, lwd=1.5, col="navyblue")
apply(data.frame(id=1:length(Ascore), score=Ascore), 1, FUN=add.seg)
```

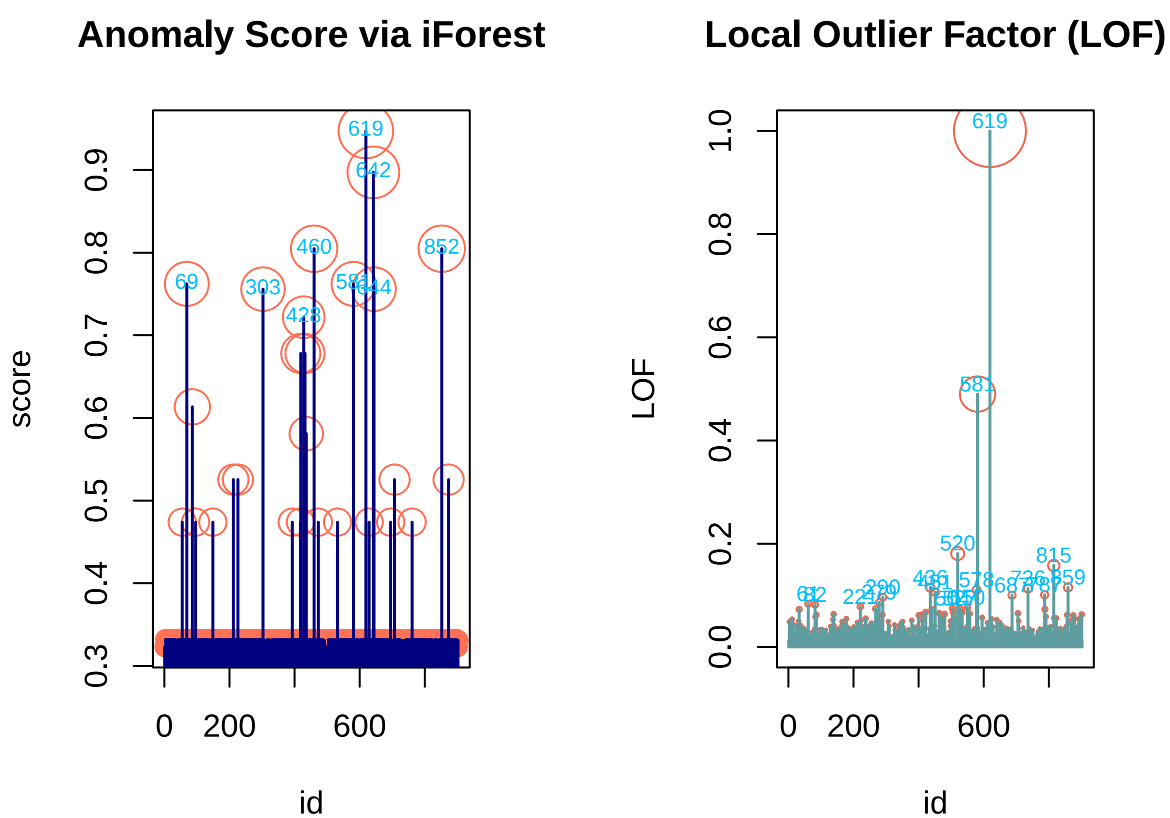
```
## NULL
```

```
eps <- 0.99
id.outliers <- which(Ascore > quantile(Ascore, eps))
text(id.outliers, Ascore[id.outliers]+0.003, label=id.outliers,
    col="deepskyblue1", cex=0.7)


#LOF
plot(x=1:length(score), score, type="p", pch=1,
    main="Local Outlier Factor (LOF)",
        xlab="id", ylab="LOF", cex=score*5, col="coral2")
add.seg <- function(x) segments(x0=x[1], y0=0, x1=x[1], y1=x[2],
    lty=1, lwd=1.5, col="cadetblue")
apply(data.frame(id=1:length(score), score=score), 1, FUN=add.seg)
```

```
## NULL
```

```
eps <- 0.98
id.outliers <- which(outlier.scores > quantile(outlier.scores, eps))
text(id.outliers, score[id.outliers]+0.02, label=id.outliers,
    col="deepskyblue1", cex=0.7)
```



Comment

- First, we observe that both plots indicate that the two methods(iForest and LOF) deemed the observations *581* and *619* as anomalies or outliers.
- Secondly, we see from the plot of the LOF that it is obvious that the observations *581* and *619* are separated from the potential outliers. However, this is not the case for the iForest.
- Hence, we conclude that the Local Outlier Factor (LOF) works better in this problem than the iForest.