Project 4

Isaiah Thompson Ocansey

2022-10-25

PAGERANK

(1a) 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
```

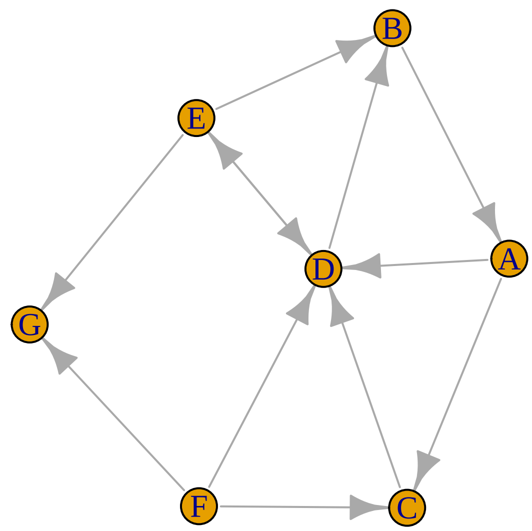
The above output shows a 7x7 link matrix L

(1b) Reproduce the graph similar to Figure 1 to check if you have got the right link matrix L

```
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)
```



The above figure shows the graph of the obtained link matrix which is similar to the one stated in the question

(1c) 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(L, method='eigen', d=.85, niter=100){
  # G is a connectivity matrix, with G[i,j]=1 if page j points to page i
  cvec <- apply(L,2,sum) # COMPUTING COLUMN SUMS
  cvec[cvec==0] <- 1 # nodes with indegree 0 will cause problems if we divide by 0.
  # rvec <- apply(G,1,sum)  # COMPUTING ROW SUMS.
  n <- nrow(L)
  delta <- (1-d)/n
  A <- matrix(delta,nrow(L),ncol(L))
  for (i in 1:n)   A[i,] <- A[i,] + d*L[i,]/cvec
#  print(A)
  if (method=='power'){
    # POWER METHOD
    x <- rep(1,n)
    for (i in 1:niter) x <- A%*%x
  } else {
    # EIGEN-DECOMPOSITION - 1ST ENGENVECTOR
    x <- Re(eigen(A)$vector[,1])
```

```
  }
  x/sum(x)
}
```

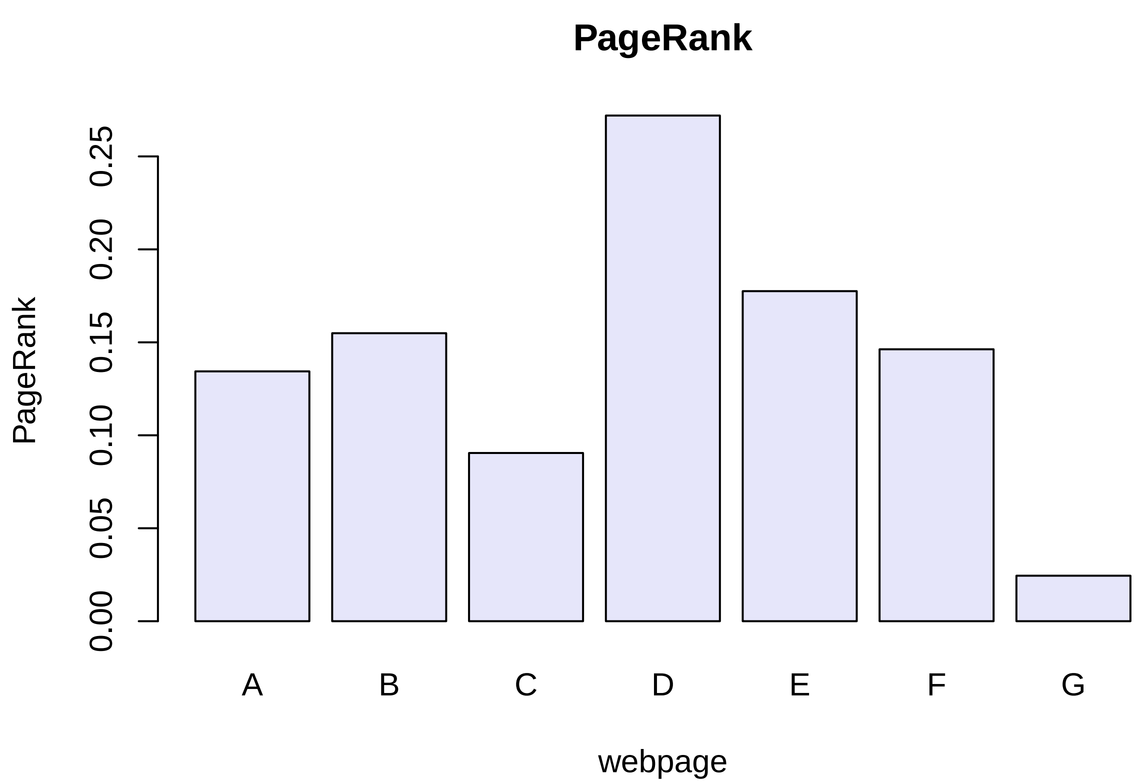
```
pg <- pagerank(L, method='eigen')
sum(pg)   # THE SUM SHOULD BE 1.
```

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

```
pg <- data.frame("WebPage"= c("A","B","C","D","E","F","G"), "PageRank"= pg)
pg
```

```
##   WebPage   PageRank
## 1       A 0.13438041
## 2       B 0.15491010
## 3       C 0.09047138
## 4       D 0.27197905
## 5       E 0.17753139
## 6       F 0.14625695
## 7       G 0.02447073
```

```
barplot(pg$PageRank, names=pg$WebPage, col="lavender", xlab="webpage", ylab="PageRank", main="PageRank"
```



The above bar plot shows the PageRank Scores for the various webpages. It can be observed from the bar plot that WebPage D has the highest score meaning it's highly accessible or has the most important Page Link than say Webpage G which has the lowest score.

```
Top3_pg <- pg[ order(pg$PageRank, decreasing = TRUE), ]
head(Top3_pg, n=3)
```

```
##   WebPage  PageRank
## 4       D 0.2719790
## 5       E 0.1775314
## 2       B 0.1549101
```

From the above output, we can see Pages D, E and B are the top 3 Webpages with most important WebPage links that are highly accessible

ANOMALY DETECTION

(2a) Bring in the data with the following R code

```
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
```

```
##             V.1          V.2           V.3           V.4           V.5
## 1  2.726436e-04  3.237405e-06  3.040539e-04  3.258806e-06  4.395332e-06
## 2 -1.268664e-04  5.179741e-05 -1.861661e-04  5.263881e-05  5.112533e-05
## 3  3.536364e-05  4.897405e-06  6.137388e-05  4.918806e-06  5.045332e-06
## 4 -3.900464e-04  1.663741e-05 -5.111861e-04  1.716881e-05  1.759533e-05
## 5 -5.985264e-04  5.097405e-06 -6.724361e-04  4.098806e-06  4.725332e-06
## 6 -7.890464e-04 -2.546259e-05 -8.462061e-04 -2.575119e-05 -2.593467e-05
##             V.6          V.7           V.8           V.9          V.10
## 1  1.949953e-04  2.604694e-06  1.314538e-04  0.0008319774  3.191322e-04
## 2 -5.665467e-05  5.025469e-05 -5.258623e-05 -0.0007294226 -3.276878e-04
## 3  3.327533e-05  4.074694e-06 -8.619623e-05 -0.0004575226  7.338218e-05
## 4 -2.185047e-04  1.458469e-05 -1.113562e-04 -0.0007819226 -5.975478e-04
## 5 -4.391347e-04  2.524694e-06 -1.264562e-04  0.0001531774 -6.203649e-04
## 6 -5.641747e-04 -2.741531e-05 -1.601762e-04  0.0003164774 -6.633731e-04
##            V.11          V.12          V.13          V.14          V.15
## 1  3.912975e-06  3.215580e-04  3.688435e-06  7.498488e-05  3.067430e-04
## 2  6.597298e-05 -2.260220e-04  5.214843e-05 -2.545119e-06 -3.322770e-04
## 3  2.742975e-06  8.162802e-05  5.148435e-06 -4.699512e-05  7.170302e-05
## 4  2.087298e-05 -5.750220e-04  1.698843e-05 -4.527512e-05 -5.864270e-04
## 5  5.672975e-06 -6.888920e-04  4.728435e-06 -5.844512e-05 -5.975832e-04
```

```
## 6 -2.855702e-05 -8.379420e-04 -2.593157e-05 -8.992512e-05 -6.341554e-04
##            V.16          V.17          V.18          V.19          V.20
## 1  3.287189e-06  2.915543e-04  1.292947e-04  5.546208e-06  2.690776e-06
## 2  4.909719e-05 -1.405857e-04 -1.832527e-05 -2.073792e-06  5.259078e-05
## 3  5.667189e-06  3.306428e-05  4.924473e-05  1.829621e-05  6.100776e-06
## 4  1.688719e-05 -4.314957e-04 -9.228527e-05 -1.120379e-05  1.691078e-05
## 5  5.037189e-06 -6.263457e-04 -2.904753e-04  4.762084e-07  5.270776e-06
## 6 -2.521281e-05 -8.188057e-04 -3.283753e-04  4.762084e-07 -2.569922e-05
##            V.21          V.22          V.23          V.24          V.25
## 1 -1.258796e-04  3.158345e-06  3.241261e-06 -4.091042e-05  2.604191e-04
## 2 -2.581296e-04  5.121835e-05  5.282126e-05  5.039958e-05 -9.240094e-05
## 3 -5.466396e-04  4.618345e-06  5.311261e-06  1.064958e-05 -1.139509e-04
## 4  6.859042e-05  1.737835e-05  1.715126e-05  3.969958e-05 -2.237309e-04
## 5 -3.685996e-04  5.938345e-06  4.791261e-06  3.829579e-06 -2.563409e-04
## 6 -3.250596e-04 -2.523165e-05 -2.525874e-05  1.757958e-05 -2.928209e-04
##            V.26          V.27          V.28          V.29          V.30
## 1  4.302839e-06 -1.622775e-05 -3.077118e-05  4.277639e-06  3.279396e-04
## 2  5.032284e-05 -3.373775e-05  1.339888e-04  4.978764e-05 -2.512904e-04
## 3  5.262839e-06  1.716225e-05  5.477882e-05  1.457639e-06  7.862959e-05
## 4  1.699284e-05 -3.617749e-06  1.028688e-04  1.430764e-05 -5.951304e-04
## 5  5.552839e-06  1.152523e-04  1.697688e-04  2.667639e-06 -6.847504e-04
## 6 -2.571716e-05  4.992251e-06  4.920882e-05 -3.125236e-05 -8.075404e-04
##            V.31         V.32          V.33          V.34          V.35
## 1 -6.030747e-07 0.0001997316  2.901919e-04  2.950068e-06  9.566039e-05
## 2  1.876385e-06 0.0015270316 -1.487181e-04  5.336007e-05 -9.959606e-06
## 3  8.165053e-07 0.0003932316  4.526186e-05  4.830068e-06  4.692039e-05
## 4  7.084353e-07 0.0002697316 -4.490381e-04  1.666007e-05 -5.015961e-05
## 5 -3.982147e-07 0.0009875316 -6.384581e-04  5.320068e-06 -2.397096e-04
## 6  4.684353e-07 0.0001635316 -8.303081e-04 -2.645993e-05 -2.389896e-04
##            V.36          V.37          V.38          V.39          V.40
## 1 -1.092576e-04 -2.045656e-05  2.309891e-04  3.163167e-04  5.269756e-05
## 2 -2.481976e-04  3.506344e-05 -8.244094e-05 -2.151833e-04 -6.306244e-05
## 3 -3.570176e-04  2.174344e-05  3.489906e-05  7.352674e-05  2.175610e-07
## 4 -5.331756e-05  2.096344e-05 -2.838009e-04 -5.567033e-04 -2.136244e-05
## 5 -2.556276e-04  9.913437e-06 -5.061909e-04 -6.835733e-04  7.071756e-05
## 6 -2.761976e-04  6.434368e-07 -6.748509e-04 -8.461033e-04  1.537561e-06
##            V.41          V.42          V.43          V.44          V.45
## 1  3.247248e-04  0.0002232093  4.880209e-06  0.0003201276  3.003677e-06
## 2 -2.621252e-04 -0.0001921907  5.232021e-05 -0.0001927824  5.197368e-05
## 3  8.469475e-05 -0.0003442907  4.400209e-06  0.0000677476  4.163677e-06
## 4 -6.017852e-04 -0.0002545907  1.695021e-05 -0.0005321124  1.660368e-05
## 5 -6.787152e-04  0.0003723093  5.310209e-06 -0.0006819024  3.843677e-06
## 6 -7.937471e-04  0.0002165093 -2.554979e-05 -0.0008634424 -2.630632e-05
##            V.46          V.47          V.48          V.49          V.50
## 1  0.0006315873  2.111365e-04  4.352026e-07  1.594326e-04 -6.199151e-05
## 2 -0.0009373127 -6.793345e-05  9.058426e-07 -3.317744e-05  3.087849e-05
## 3 -0.0009843127  2.850655e-05  3.516726e-07  4.349256e-05  9.531849e-05
## 4 -0.0008205127 -2.531135e-04 -6.437395e-09 -1.461174e-04  5.563849e-05
## 5 -0.0001046127 -4.731935e-04  3.893726e-07 -3.537474e-04  3.750185e-04
## 6  0.0002708873 -6.194835e-04  7.549726e-07 -4.385374e-04  8.896849e-05
##            V.51          V.52          V.53          V.54          V.55
## 1  3.343624e-04  4.261938e-04  3.041318e-04 -4.887982e-06  2.822660e-06
## 2 -1.225376e-04  8.544378e-05 -1.625982e-04  4.583202e-05  5.271266e-05
## 3  1.996242e-05 -5.385622e-05  4.706179e-05  5.122202e-05  5.302660e-06
```

```
## 4 -2.811376e-04 -1.897262e-04 -4.811382e-04  7.822018e-06  1.724266e-05
## 5  3.775624e-04 -4.276462e-04 -6.568382e-04  2.335202e-05  4.982660e-06
## 6  2.430624e-04 -4.580262e-04 -8.505282e-04  1.902018e-06 -2.669734e-05
##            V.56          V.57          V.58          V.59          V.60
## 1 -2.674375e-05  3.852197e-04  3.020560e-04  3.211581e-04  3.162044e-05
## 2  1.999625e-05 -3.127029e-05 -3.207240e-04 -2.664319e-04  3.974044e-05
## 3  1.216625e-05 -7.964029e-05  7.486604e-05  9.366806e-05 -1.807956e-05
## 4  1.749625e-05 -2.833003e-04 -5.875340e-04 -6.118219e-04 -5.759557e-06
## 5 -1.987375e-05 -4.108703e-04 -6.017588e-04 -6.804719e-04 -2.085956e-05
## 6  6.536253e-06 -4.664803e-04 -6.421109e-04 -7.871287e-04 -5.376956e-05
##            V.61          V.62          V.63          V.64          V.65
## 1  2.551865e-04  2.523415e-06 -1.751972e-05  3.235729e-04  3.323603e-06
## 2 -9.046352e-05  5.313341e-05  3.014028e-05 -3.158071e-04  5.209360e-05
## 3  2.789648e-05  5.213415e-06  4.020277e-06  8.353288e-05  5.603603e-06
## 4 -3.202635e-04  1.755341e-05  2.527028e-05 -6.100771e-04  1.621360e-05
## 5 -5.447535e-04  4.483415e-06  1.413028e-05 -6.389371e-04  4.873603e-06
## 6 -7.210835e-04 -2.617659e-05  1.310277e-06 -6.994114e-04 -2.598640e-05
##            V.66          V.67          V.68          V.69          V.70
## 1 -4.169452e-05  3.178148e-06  1.838077e-04  3.722320e-06  2.483639e-06
## 2  2.312548e-05  5.204815e-05 -3.980233e-05  5.760232e-05  5.217364e-05
## 3 -4.754523e-06  4.538148e-06  3.911767e-05  3.152320e-06  6.093639e-06
## 4  1.930548e-05  1.657815e-05 -1.792423e-04  1.722232e-05  1.721364e-05
## 5  8.995477e-06  5.448148e-06 -4.043023e-04  3.742320e-06  5.063639e-06
## 6  1.086548e-05 -2.572185e-05 -5.126923e-04 -2.721768e-05 -2.590636e-05
##            V.71          V.72          V.73          V.74          V.75
## 1  3.586079e-05  3.679196e-06  1.176750e-05  3.259493e-04  1.457141e-04
## 2 -7.087921e-05  5.337920e-05  4.259750e-05 -2.990307e-04 -2.348594e-05
## 3  3.596079e-05  4.839196e-06 -3.382502e-06  9.824927e-05  4.660406e-05
## 4 -2.051921e-05  1.769920e-05  6.917498e-06 -6.217807e-04 -1.204559e-04
## 5 -1.808992e-04  5.639196e-06 -5.542502e-06 -6.662907e-04 -3.239259e-04
## 6 -1.325792e-04 -2.603080e-05 -3.558250e-05 -7.413748e-04 -3.859359e-04
##            V.76          V.77          V.78          V.79          V.80
## 1  7.455461e-05 -0.0001527421  1.181130e-04  0.0009664207  0.0000613574
## 2 -9.185393e-06 -0.0002001521 -1.337699e-05  0.0007458207 -0.0000301626
## 3  4.406461e-05 -0.0006433421  4.530301e-05 -0.0002043793  0.0000420874
## 4 -3.179539e-05  0.0001423779 -7.149699e-05 -0.0005676793 -0.0000234526
## 5 -2.072654e-04 -0.0004419521 -2.734770e-04  0.0004198207 -0.0001997426
## 6 -1.850954e-04 -0.0003172321 -3.002370e-04  0.0003945207 -0.0001587826
##            V.81          V.82          V.83          V.84          V.85
## 1  3.140559e-04  3.304995e-04  6.671859e-05  3.385299e-06  2.476920e-04
## 2 -3.100041e-04 -2.801905e-04 -6.161141e-05  5.195530e-05 -1.079280e-04
## 3  7.483592e-05  8.852954e-05  7.458592e-06  4.645299e-06  3.029203e-05
## 4 -5.928241e-04 -6.221205e-04 -5.221408e-06  1.709530e-05 -3.499080e-04
## 5 -6.312941e-04 -6.785005e-04  8.011859e-05  5.455299e-06 -5.641680e-04
## 6 -6.998402e-04 -7.788191e-04 -1.716141e-05 -2.591470e-05 -7.471280e-04
##            V.86          V.87          V.88
## 1  2.129363e-06 -2.691822e-05  3.447164e-04
## 2  6.336936e-05  1.345918e-04  1.488464e-04
## 3  3.199363e-06  5.587178e-05 -7.975364e-05
## 4  1.980936e-05  1.015818e-04 -6.473636e-06
## 5  4.809363e-06  1.737518e-04 -2.378336e-04
## 6 -2.830064e-05  4.943178e-05 -1.816936e-04
```

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

The HTP data set has 902 observations and 88 variables

(2b) First obtain robust estimates of the mean vector $\hat{\mu}$ and the VCOV matrix $\hat{\Sigma}$ of the data with MCD with a breakdown point of your choice. Then compute the robust Mahalanobis distance of each observation with repect to the MCD estimates ($\hat{\mu}$, $\hat{\Sigma}$) and plot them. You may add a threshold based on the distribution and highlight the two defective parts. Are the two defective parts in your top list of potential outliers?

```r
library(robustbase)

fit.robust <- covMcd(dat, cor = FALSE, alpha = 0.80)
```

The above is the MCD estimates with a breakdown point of 20%

```r
Mean_vector <- fit.robust$center
Mean_vector
```

```
##           V.1           V.2           V.3           V.4           V.5
##  1.554054e-05 -3.263036e-07  5.650411e-06 -3.262559e-07 -3.055800e-07
##           V.6           V.7           V.8           V.9          V.10
##  2.740012e-05 -3.242202e-07  7.990267e-06 -9.570366e-05 -1.700507e-05
##          V.11          V.12          V.13          V.14          V.15
## -3.621558e-07 -1.232412e-06 -3.000408e-07  2.968625e-06 -1.887928e-05
##          V.16          V.17          V.18          V.19          V.20
## -3.211508e-07  1.242645e-05  2.865364e-05  5.010131e-07 -3.143619e-07
##          V.21          V.22          V.23          V.24          V.25
## -3.233804e-06 -3.025326e-07 -3.302853e-07  2.074239e-06  1.865508e-05
##          V.26          V.27          V.28          V.29          V.30
## -3.111565e-07  1.834421e-06  6.725844e-06 -3.612078e-07 -5.108829e-06
##          V.31          V.32          V.33          V.34          V.35
##  1.063921e-07 -2.270386e-05  1.167567e-05 -3.002464e-07  2.404799e-05
##          V.36          V.37          V.38          V.39          V.40
## -5.615028e-06  1.941078e-06  2.363152e-05  9.180380e-07  9.081713e-08
##          V.41          V.42          V.43          V.44          V.45
## -6.555489e-06 -9.963700e-05 -3.217333e-07  4.174405e-06 -3.240930e-07
##          V.46          V.47          V.48          V.49          V.50
## -1.644013e-04  2.574089e-05  1.034790e-07  2.999431e-05  2.535181e-05
##          V.51          V.52          V.53          V.54          V.55
## -3.201573e-05  6.036546e-05  9.426361e-06  2.931410e-06 -3.365812e-07
##          V.56          V.57          V.58          V.59          V.60
##  9.812166e-07  3.730185e-05 -1.834855e-05 -7.658910e-06  5.033523e-07
##          V.61          V.62          V.63          V.64          V.65
##  2.103621e-05 -3.126335e-07  1.494373e-06 -1.466965e-05 -3.212794e-07
##          V.66          V.67          V.68          V.69          V.70
##  7.727271e-07 -3.034190e-07  2.866046e-05 -3.439176e-07 -2.776363e-07
##          V.71          V.72          V.73          V.74          V.75
##  2.685564e-06 -3.021946e-07 -2.178893e-07 -1.193592e-05  2.954968e-05
##          V.76          V.77          V.78          V.79          V.80
##  1.952435e-05 -2.601964e-06  2.659005e-05 -9.150836e-05  1.316938e-05
##          V.81          V.82          V.83          V.84          V.85
## -1.390503e-05 -9.025224e-06  9.445309e-08 -3.218053e-07  1.926770e-05
##          V.86          V.87          V.88
## -3.458167e-07  6.766604e-06  8.432421e-05
```

The output above shows the robust estimates of the mean vector

```
VCOV  <- fit.robust$cov
```

The above function shows the robust estimates of the variance covariance matrix of the data with MCD with a breakdown point of 20%

```
RD <- mahalanobis(dat, Mean_vector , VCOV)
head(RD)
```

```
## [1]  97.68606 204.83381  86.21858  76.43213  76.33860  76.04470
```

The above shows the robust Mahalanobis distance of each observation with respect to the MCD estimates

```
cutoff.chi.sq <- qchisq(0.975, df = ncol(dat)); cutoff.chi.sq
```

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

The above is the Cut-off point based on the chi-square distribution

```
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
```

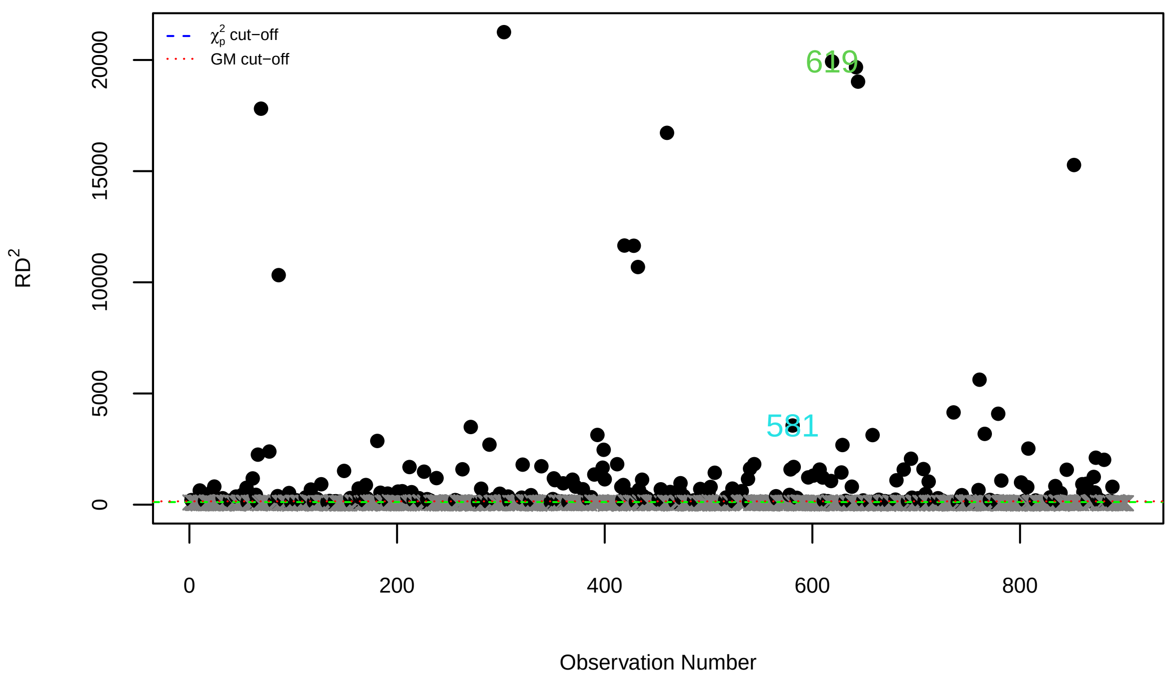
```
which(RD >= cutoff.GM)  # OUTLIER IDs
```

```
##   [1]   2  10  15  22  24  32  39  45  51  55  61  64  65  66  67  69  77  85
##  [19]  86  91  96 103 108 112 113 117 123 127 135 140 141 149 155 160 163 164
##  [37] 165 167 169 170 171 181 184 185 191 201 205 210 212 214 216 221 226 229
##  [55] 230 231 232 238 256 257 263 271 281 284 289 290 294 299 303 307 308 310
##  [73] 320 321 328 329 332 339 350 351 352 354 360 369 372 379 384 386 387 390
##  [91] 393 398 399 400 412 416 417 418 419 424 428 432 433 436 437 438 441 452
## [109] 453 454 456 457 460 463 472 473 474 476 486 492 500 502 506 516 517 520
## [127] 523 524 526 527 528 532 538 540 544 565 566 578 579 581 582 585 596 601
## [145] 607 610 611 615 618 619 628 629 632 638 642 644 649 658 664 665 670 680
## [163] 681 688 692 695 696 702 703 707 708 709 712 721 725 736 743 744 760 761
## [181] 766 771 772 778 779 782 801 805 807 808 815 829 833 834 839 845 852 860
## [199] 861 864 865 871 872 873 874 876 878 881 886 888 889
```

The above is another Cut-off Suggested by Green and Martin (2017)

```
colPoints <- ifelse(RD >= min(c(cutoff.chi.sq, cutoff.GM)), 1, grey(0.5))
pchPoints <- ifelse(RD >= min(c(cutoff.chi.sq, cutoff.GM)), 16, 4)
plot(seq_along(RD), RD, pch = pchPoints, col = colPoints,
ylim=c(0, max(RD, cutoff.chi.sq, cutoff.GM) + 2), cex.axis = 0.7, cex.lab = 0.7,
```

```
ylab = expression(RD**2), xlab = "Observation Number")
abline(h = c(cutoff.chi.sq, cutoff.GM), lty = c("dashed", "dotted"), col=c("green", "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, RD[619], labels=619, col=619)
text(581, RD[581], labels=581, col=581)
```



Using the threshold according to the chi-square distribution and one suggested by Green and Martin (2017), We can observe the defective parts are on the top list of potential outliers

(2c) Apply isolation forest (iForest), local outlier factor (LOF), and, optionally, one-class SVM for the same task. Choose the involved parameters appropriately based on your own judgment and you may compare results by varying the parameters. Plot the results. Comment on the similarities and differences of their results. In particular, pay attention to whether the two defective parts are deemed anomalies by each method.

```
library(isofor)
# help(package="isofor")

fit.isoforest <- iForest(dat, nt=100, phi=256)
pred <- predict(fit.isoforest, newdata=dat)
pred # Higher scores correspond to more isolated observations
```

```
##   [1] 0.4075197 0.4379984 0.3846837 0.3899160 0.4197782 0.4428606 0.3945396
##   [8] 0.4347594 0.3954978 0.3845581 0.4302465 0.4508016 0.4583920 0.3942170
##  [15] 0.6077039 0.4153783 0.5167661 0.3780448 0.3818856 0.3967165 0.3997048
##  [22] 0.4595605 0.3989609 0.4059214 0.4007960 0.4159920 0.3936232 0.4158151
```

```
##  [29] 0.4339360 0.4321472 0.3911837 0.6378410 0.5379088 0.3810954 0.4037336
##  [36] 0.4528047 0.4728030 0.4556375 0.5424621 0.4054102 0.3968807 0.3833722
##  [43] 0.4522670 0.3847227 0.3863855 0.4695157 0.4195649 0.5395623 0.3970525
##  [50] 0.3742683 0.4342688 0.4656726 0.4737054 0.3892082 0.5336524 0.4556565
##  [57] 0.4399216 0.3823067 0.4013177 0.3828288 0.4420389 0.3994373 0.4099340
##  [64] 0.4048121 0.4223116 0.4383571 0.4396853 0.4137395 0.6954813 0.4007189
##  [71] 0.3885775 0.3943080 0.4220197 0.3900362 0.4207761 0.3952254 0.4029991
##  [78] 0.3879934 0.4056275 0.4205894 0.3747315 0.5822228 0.4859812 0.3767980
##  [85] 0.4078896 0.6101668 0.4021414 0.4112117 0.4027499 0.4232868 0.3976816
##  [92] 0.3946688 0.4169852 0.4190931 0.3880979 0.4994611 0.4373718 0.4211563
##  [99] 0.4100497 0.4265044 0.5252288 0.5099693 0.5234745 0.4410786 0.4230315
## [106] 0.3958507 0.4088996 0.5157417 0.4606954 0.3955465 0.3906906 0.4126562
## [113] 0.4384368 0.4034623 0.3905213 0.4849691 0.4281853 0.4617063 0.3920448
## [120] 0.4240418 0.4212622 0.4028514 0.4653446 0.3918748 0.4116803 0.4023021
## [127] 0.4212319 0.3978927 0.4055349 0.4079117 0.4403517 0.4479147 0.3987206
## [134] 0.4762434 0.4284581 0.4530249 0.4499953 0.4139956 0.3964227 0.4285304
## [141] 0.4453938 0.3775094 0.3973217 0.4651419 0.4089237 0.4103029 0.4250346
## [148] 0.3715523 0.4879794 0.4056408 0.4168780 0.4233511 0.3876105 0.3916592
## [155] 0.4059426 0.3887739 0.4826612 0.4033792 0.3797539 0.4471010 0.4361441
## [162] 0.4077624 0.4120292 0.3944081 0.4336888 0.4190320 0.4461519 0.4458993
## [169] 0.4162887 0.4124294 0.5120862 0.4182163 0.3861798 0.3955262 0.4306673
## [176] 0.3870044 0.4818497 0.5386807 0.4108918 0.4916226 0.4734105 0.4800502
## [183] 0.4148132 0.4871991 0.4440254 0.4178277 0.4092654 0.4147133 0.4205075
## [190] 0.4587623 0.4921874 0.5799746 0.4741544 0.3936816 0.4088549 0.4898790
## [197] 0.4424259 0.3943465 0.4155079 0.4385506 0.4224465 0.4162225 0.3940401
## [204] 0.4025172 0.5037430 0.4217291 0.4297366 0.4006133 0.3892509 0.3979136
## [211] 0.3864548 0.5194190 0.4102098 0.4260546 0.5572636 0.4685673 0.4666969
## [218] 0.5429849 0.4963630 0.4010177 0.4229740 0.4150937 0.4852297 0.3984925
## [225] 0.3854872 0.5695280 0.4627766 0.4214782 0.4362284 0.5151577 0.4021267
## [232] 0.3918843 0.4105503 0.4491243 0.4029956 0.3886109 0.4087335 0.4356638
## [239] 0.3897759 0.4007446 0.3999743 0.4246969 0.4482791 0.3880801 0.3838348
## [246] 0.3829015 0.4098427 0.4676545 0.5109573 0.4072530 0.4105800 0.3805017
## [253] 0.3778129 0.3884026 0.4085641 0.4346284 0.4224322 0.3928863 0.3996202
## [260] 0.4417900 0.3790095 0.4578820 0.4083107 0.3845032 0.5243667 0.3932649
## [267] 0.3718120 0.4469686 0.4157094 0.3947725 0.4566254 0.4173185 0.4011719
## [274] 0.3946321 0.3850404 0.4295310 0.4577897 0.4467777 0.4284090 0.3926725
## [281] 0.4282588 0.3869649 0.3997188 0.4089661 0.4192628 0.3798988 0.3986670
## [288] 0.3912772 0.4571619 0.4273764 0.3980057 0.4247481 0.4154669 0.4305003
## [295] 0.4242446 0.3971669 0.4013450 0.3948067 0.4569727 0.3955431 0.4583616
## [302] 0.4086035 0.7169310 0.4063003 0.4040384 0.4124101 0.3957010 0.4241052
## [309] 0.3983717 0.3976652 0.4247126 0.3815413 0.4312645 0.3799176 0.3770766
## [316] 0.3774477 0.4842198 0.3931144 0.4278090 0.4053311 0.3893182 0.4293652
## [323] 0.3843791 0.4021764 0.4158462 0.3888265 0.4378662 0.4084190 0.4958701
## [330] 0.3921714 0.4089414 0.3958843 0.3851265 0.4382253 0.4031087 0.3971416
## [337] 0.4134263 0.4578275 0.4561925 0.4691588 0.4521505 0.4134182 0.4427136
## [344] 0.4122907 0.4970307 0.3982062 0.4336519 0.4641428 0.4422023 0.4114716
## [351] 0.4240150 0.4420571 0.4115820 0.4637084 0.5271798 0.4544634 0.4007327
## [358] 0.3991585 0.3898466 0.4232118 0.3856088 0.3901958 0.4123041 0.3992194
## [365] 0.4547331 0.4416408 0.4185041 0.4374108 0.5476587 0.3859308 0.3860577
## [372] 0.3971584 0.3969331 0.4167729 0.4599158 0.3858413 0.4057633 0.4337341
## [379] 0.4196974 0.3827109 0.4128256 0.4109999 0.3946343 0.4556975 0.4259346
## [386] 0.4558744 0.3836730 0.4701785 0.4575545 0.5109609 0.4132272 0.3805028
## [393] 0.5477467 0.3959987 0.4510731 0.4691425 0.4991416 0.3898238 0.4287348
## [400] 0.4273458 0.4374255 0.4194528 0.4592295 0.4133646 0.4193689 0.3882223
```

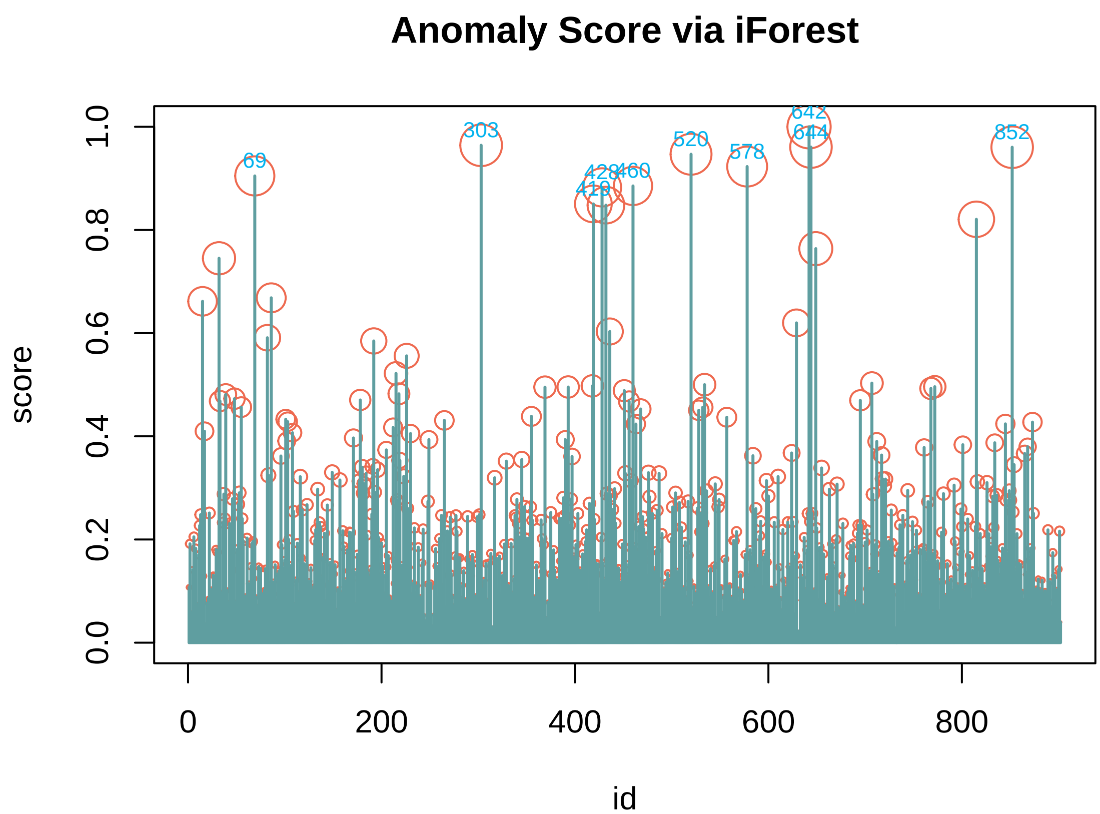
```
## [407] 0.3936327 0.4308889 0.3912420 0.3836337 0.4396433 0.4477763 0.3834974
## [414] 0.4054190 0.4662317 0.4229875 0.4144254 0.5485088 0.6758784 0.4552294
## [421] 0.4024697 0.4244575 0.4165536 0.4238738 0.3831198 0.3891084 0.4426468
## [428] 0.6874504 0.4244869 0.4089380 0.4124925 0.6750359 0.4731557 0.4266372
## [435] 0.3798251 0.5865942 0.4717840 0.4379544 0.4620885 0.3921234 0.4765840
## [442] 0.4523821 0.4197887 0.4072820 0.4210929 0.4062304 0.3786156 0.4114568
## [449] 0.4378435 0.4159736 0.5451975 0.4873901 0.4237317 0.4154973 0.3967042
## [456] 0.5376056 0.4354794 0.4822362 0.3895406 0.6884835 0.3847314 0.4357099
## [463] 0.5218309 0.4319074 0.4009161 0.4500790 0.3726107 0.5323470 0.4054107
## [470] 0.4569603 0.4483889 0.4262721 0.4427207 0.4142584 0.4131318 0.4877697
## [477] 0.4709884 0.4319400 0.4206203 0.4592443 0.4104059 0.4157294 0.4185150
## [484] 0.4331634 0.4613827 0.4126473 0.4873094 0.3861842 0.3821101 0.4451563
## [491] 0.3851738 0.4058800 0.3952601 0.4111058 0.3926273 0.4175895 0.3858614
## [498] 0.4004181 0.4136510 0.4419276 0.4637543 0.4076190 0.4168671 0.4736448
## [505] 0.3898835 0.4184818 0.4063803 0.4669138 0.3842500 0.4496260 0.3734743
## [512] 0.4066789 0.4066126 0.4393054 0.3834685 0.4001983 0.4677353 0.4300331
## [519] 0.4119241 0.7106547 0.3819160 0.3930824 0.4080162 0.3967784 0.4032575
## [526] 0.3786935 0.4632349 0.5314638 0.4463518 0.4052984 0.4597619 0.5335145
## [533] 0.4521434 0.5493858 0.4000570 0.4750862 0.4075644 0.3995228 0.3832616
## [540] 0.4205686 0.3885478 0.4033466 0.3984997 0.4227833 0.4799312 0.3806206
## [547] 0.3862483 0.4639294 0.4689366 0.4072102 0.4075004 0.3896055 0.3760549
## [554] 0.4010385 0.4272822 0.3828469 0.5265765 0.3838876 0.3924950 0.4086867
## [561] 0.3829185 0.3983583 0.4004298 0.4403386 0.4401318 0.4016126 0.4465184
## [568] 0.3960555 0.3730895 0.4059488 0.4165530 0.4050513 0.3972833 0.3899800
## [575] 0.3978268 0.4304793 0.3830367 0.7020297 0.4146650 0.4079674 0.4338024
## [582] 0.4288981 0.3905649 0.4996344 0.4060550 0.3789799 0.4625713 0.4209068
## [589] 0.4195516 0.3992481 0.4435898 0.4537545 0.3786673 0.4068201 0.4287271
## [596] 0.3967148 0.4308226 0.4822078 0.4251093 0.4714290 0.3869774 0.3935498
## [603] 0.4064729 0.3909333 0.4051868 0.4531252 0.3935715 0.4077854 0.3948149
## [610] 0.4850358 0.4216632 0.3995758 0.3896596 0.3912508 0.4480264 0.4126037
## [617] 0.4033594 0.4071689 0.4009844 0.4531079 0.4110938 0.4218781 0.3971028
## [624] 0.5015777 0.4534081 0.3885387 0.3782812 0.4294814 0.5926176 0.3772404
## [631] 0.3747611 0.3741409 0.4232807 0.3985009 0.4143726 0.3813349 0.4425187
## [638] 0.3963400 0.4055690 0.4336969 0.4591258 0.7298508 0.4535770 0.7156682
## [645] 0.4592503 0.3915521 0.3952443 0.4053627 0.6445852 0.4490633 0.3980381
## [652] 0.3891176 0.4355365 0.3939143 0.4910911 0.3980042 0.4308891 0.3951834
## [659] 0.3942788 0.3922190 0.4148482 0.3822316 0.4762207 0.4111908 0.3933066
## [666] 0.4378641 0.4228247 0.3893763 0.3811055 0.4410292 0.4797690 0.3934075
## [673] 0.3709644 0.3915256 0.4041215 0.4159211 0.4522048 0.3731085 0.4005212
## [680] 0.3936770 0.3956852 0.4037544 0.4071328 0.4294502 0.4369291 0.3810097
## [687] 0.4275368 0.4381746 0.3765985 0.3884198 0.3972047 0.4453755 0.4510682
## [694] 0.3835982 0.5384070 0.4509934 0.3888218 0.3944683 0.3904996 0.4392049
## [701] 0.4064946 0.4476201 0.4173988 0.3743696 0.4191138 0.4064912 0.5504790
## [708] 0.4726366 0.3888804 0.4308333 0.4376257 0.5096250 0.3886754 0.4167708
## [715] 0.3925564 0.4338891 0.5001243 0.4834644 0.3774519 0.4784086 0.4832978
## [722] 0.4086580 0.4157471 0.4397015 0.4339104 0.3874472 0.4616269 0.4386129
## [729] 0.4023115 0.4082237 0.3791040 0.3688089 0.4316094 0.3826845 0.4355425
## [736] 0.4512649 0.3943060 0.4087626 0.4579401 0.3893521 0.3750208 0.4099570
## [743] 0.4262439 0.4753576 0.3884350 0.4178621 0.4301108 0.4000974 0.4534985
## [750] 0.3792336 0.4126380 0.4332581 0.4475372 0.3744086 0.3963745 0.4022135
## [757] 0.4339047 0.3976383 0.4003353 0.4351302 0.5054318 0.4323115 0.3914390
## [764] 0.4251213 0.4674178 0.4263825 0.4299389 0.5467420 0.3883422 0.3789630
## [771] 0.4312101 0.5479526 0.3887796 0.4219470 0.4260719 0.3904000 0.3906306
## [778] 0.4116892 0.4461089 0.4227790 0.4731226 0.3984463 0.3881265 0.4016953
```

```
## [785] 0.4240891 0.4111221 0.4079775 0.4046198 0.4133662 0.4044807 0.3998819
## [792] 0.4790604 0.4396435 0.4214005 0.4087636 0.4047432 0.4062677 0.4326741
## [799] 0.4623191 0.4500006 0.5073522 0.4037897 0.3735443 0.4084113 0.4034643
## [806] 0.4553760 0.4164890 0.4297159 0.3809650 0.3983384 0.4194623 0.4087524
## [813] 0.3947241 0.4500658 0.6651196 0.4813764 0.4122785 0.3802205 0.4451867
## [820] 0.4070384 0.4205662 0.3887845 0.4088474 0.4041520 0.4016833 0.4807863
## [827] 0.4223613 0.4449648 0.4292388 0.4358308 0.4228431 0.4713298 0.4494030
## [834] 0.5087160 0.3830030 0.4720342 0.4265146 0.4148888 0.4024177 0.4241978
## [841] 0.3908136 0.4350168 0.3905925 0.3951182 0.5219270 0.4686489 0.4311044
## [848] 0.4089245 0.4751345 0.4685167 0.3934081 0.7155575 0.4603002 0.4933783
## [855] 0.3933014 0.4185331 0.4450598 0.3976222 0.4241730 0.3923857 0.4234270
## [862] 0.4044272 0.3981987 0.3896282 0.5011603 0.3971523 0.4164593 0.5059338
## [869] 0.3968471 0.4262109 0.4350181 0.4116273 0.5231941 0.4593309 0.4046699
## [876] 0.4047582 0.4065590 0.4063876 0.4133677 0.3878084 0.4053653 0.3798104
## [883] 0.4124934 0.4025437 0.4032075 0.4050203 0.3924104 0.3880049 0.4477567
## [890] 0.3850112 0.4115842 0.3878776 0.3932225 0.4317903 0.3985978 0.4157722
## [897] 0.3847828 0.4057049 0.3857488 0.4202054 0.4468403 0.3831532
```

```r
score <- scale(pred, center = min(pred), scale = max(pred)-min(pred))
par(mfrow=c(1,1), mar=rep(4,4))
plot(x=1:length(score), score, type="p", pch=21,
     main="Anomaly Score via iForest",
     xlab="id", ylab="score", cex=score*3, 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
```

```r
eps <- 0.99
id.outliers <- which(score > quantile(score, eps))
text(id.outliers, score[id.outliers]+0.03, label=id.outliers,
     col="deepskyblue2", cex=0.7)
```



The above shows the plot of the anomaly scores using iForest. Using a probability of 0.99, it can be seen that 642,644, 520, 303 are among the top potential outliers. However, the iForest is unable to detect the defective parts,581 and 619 clearly as among the top outliers

```
library(Rlof)
```

```
## Loading required package: doParallel

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

```
outlier.scores <- lof(dat, k=5);  outlier.scores
```

```
##   [1] 1.0130646 1.0026690 1.2698831 0.9937883 1.0698406 1.0526181 0.9969470
##   [8] 0.9652022 1.0531495 1.0953653 1.2359542 1.0505848 1.0022157 1.0138839
##  [15] 1.2125433 1.0679045 1.0861551 1.0422267 1.0476032 0.9927402 1.0348282
##  [22] 1.0412071 1.0411443 1.0528434 1.1913782 1.0205773 1.1987251 1.0908414
##  [29] 1.0412421 1.0920116 1.1278609 1.2323410 1.3934765 0.9597107 1.0149273
##  [36] 1.0021007 1.1685739 0.9987907 1.1888641 1.1150786 1.0433465 1.0172610
##  [43] 1.0008317 1.0815763 1.0099252 1.0921757 0.9879717 1.1592862 1.1311863
##  [50] 0.9602683 1.0102893 1.0784421 1.0954553 1.0284864 0.9953941 1.1488052
```

```
##  [57] 1.1345856 1.0764542 1.0128263 1.1625717 1.4263298 1.0300384 1.0297903
##  [64] 1.0430390 1.0687539 0.9927208 1.0123170 0.9743408 0.9996940 1.1225367
##  [71] 1.0212177 1.0937460 1.1715180 0.9806828 1.0828983 1.0238431 0.9666242
##  [78] 0.9866503 1.1610968 1.0173475 1.0255940 1.4756065 1.2729940 0.9758654
##  [85] 1.2057328 1.4508796 1.0006321 1.0559245 1.0810407 0.9869268 1.0752885
##  [92] 0.9412999 1.0142409 1.1269510 1.0235266 0.9928256 1.1961625 1.1637700
##  [99] 1.1315115 1.0761006 1.1677101 1.1111824 1.1748065 0.9656787 1.0102432
## [106] 1.0788452 1.0499539 1.0923672 1.0491671 1.0698388 1.1730374 1.0074821
## [113] 1.0287733 1.1606395 1.0364324 1.1327557 1.0729796 1.0933785 1.0942299
## [120] 1.0284986 0.9857282 0.9889188 1.0399860 1.0220014 1.0563384 1.0466014
## [127] 1.0245755 1.0056989 1.1968950 1.0004316 1.0584493 1.0462879 0.9958494
## [134] 1.0377807 0.9955719 1.0065005 1.2901005 1.0503985 1.3484727 1.0255315
## [141] 1.1846866 0.9654542 1.0771807 1.1133173 0.9585507 0.9961500 1.1048431
## [148] 1.0007747 1.0828579 1.1034519 1.0042254 1.0739349 1.2144210 1.0597764
## [155] 1.0035612 0.9701836 1.0343926 1.1866356 1.0114284 1.0288939 0.9527717
## [162] 1.1084940 1.0549243 1.2000004 1.2387126 1.0059415 1.1566921 0.9873227
## [169] 1.0920757 1.0827306 1.1870144 1.0087105 1.0186671 1.0078896 1.0015129
## [176] 0.9938276 1.0276783 1.3254272 1.0255891 1.1743835 1.2398211 0.9995837
## [183] 1.0257649 0.9954422 1.1399999 1.0936163 1.1521294 1.1646778 1.0079993
## [190] 1.0810200 1.0473263 1.1586383 1.0765432 1.0280875 0.9892320 1.1878829
## [197] 1.0363444 0.9970665 1.0065067 0.9932427 1.1198150 1.1781049 1.0146516
## [204] 1.0093421 0.9663997 1.1270069 1.2162074 1.0931077 0.9988626 1.1400659
## [211] 0.9958146 1.2639255 1.1445671 1.0421322 1.0080941 1.0397546 0.9858636
## [218] 1.1258235 1.0944326 1.0283477 1.4265478 0.9799097 1.3493515 0.9986482
## [225] 0.9659403 1.0082739 1.0165181 1.0555367 1.0865813 1.2176622 1.1761632
## [232] 1.0231147 0.9588439 1.0261502 0.9828582 0.9880944 1.2434063 1.2784328
## [239] 1.0914913 1.1935626 1.0666438 1.0119777 0.9658219 1.0792523 1.0999443
## [246] 1.0219181 1.1070315 1.0902321 1.2004736 1.1082614 1.2146400 1.0221925
## [253] 1.1228421 1.3050182 1.2510039 1.0219540 1.0449219 1.1450080 1.0655788
## [260] 0.9786190 1.0624256 1.2440309 1.0604504 1.0258162 1.1636297 1.0891161
## [267] 1.0880370 1.4230477 1.0260997 1.1539802 1.1621714 1.0900728 1.1618709
## [274] 1.0846786 1.3545524 1.1248324 1.1368312 0.9801392 1.5012061 1.0880805
## [281] 1.0816213 0.9973747 1.1569497 1.1521188 1.2437652 1.0459988 0.9762938
## [288] 1.0196020 1.3596719 1.6082813 0.9898272 0.9690175 0.9936501 1.1153705
## [295] 0.9722559 0.9986041 1.0220608 1.1115897 1.0654129 1.1320447 1.0805123
## [302] 0.9885485 1.0048114 1.0880115 1.0003509 1.0302452 1.2768035 1.0075890
## [309] 0.9988453 1.1754472 1.0064778 1.0296740 0.9986320 0.9749650 1.0111630
## [316] 1.0269960 0.9576067 1.0855831 1.0685549 1.0318585 0.9711277 0.9674809
## [323] 0.9960580 0.9864436 1.1184428 1.2075821 1.1157432 1.0822654 0.9874509
## [330] 1.0259759 1.0272710 1.1450585 0.9854368 1.0139917 1.0939248 0.9891636
## [337] 1.0341667 1.0025147 1.1808678 1.1540984 1.1019994 1.0470703 1.0397311
## [344] 0.9997979 1.0248350 1.0105304 1.2592090 0.9712481 1.0557178 1.0798388
## [351] 1.2165826 1.0839467 1.0752544 1.0159154 1.1834509 1.0007520 1.0004349
## [358] 1.0298982 1.0805208 1.1188893 1.0822428 1.0218352 1.0490995 1.0170330
## [365] 0.9743307 1.0165861 1.0288895 1.0875091 1.1382554 0.9990900 1.0764894
## [372] 1.0785795 0.9783215 0.9983899 1.0758608 1.0038558 1.0676153 1.0633339
## [379] 1.2870614 1.1866207 0.9897159 1.0151156 1.0306545 1.1795275 1.0681156
## [386] 1.0045502 1.0128882 1.0366492 1.0786204 1.0117433 0.9796916 0.9904850
## [393] 0.9967803 1.2044915 1.1485363 1.0184977 1.1981469 1.1623622 1.0065533
## [400] 1.4144770 1.0001564 1.0758723 1.1293332 1.0415203 1.0920131 1.1576401
## [407] 1.0169572 1.0988079 1.0048566 1.0129009 1.3314054 1.3018919 1.0161853
## [414] 1.0090154 0.9722915 0.9992230 0.9770953 1.0529857 0.9715481 1.1807189
## [421] 0.9771963 1.4086051 1.0128506 1.0203101 1.0579707 1.0601357 0.9694717
## [428] 0.9735998 1.0688397 1.0549755 1.0122931 1.0056704 1.1383799 1.1017357
```

```
## [435] 1.0133527 1.6830372 1.0044776 1.0136494 1.0469198 1.1048160 1.3923614
## [442] 1.0019087 0.9992524 0.9811626 1.0576777 1.2082887 1.0764464 1.0841825
## [449] 1.0615895 1.2542881 1.5812256 1.0766399 1.1252896 1.0197454 1.0371316
## [456] 1.1505286 1.1039965 1.0029194 1.0684320 0.9969274 1.1054781 1.0729905
## [463] 1.2374157 1.0080852 0.9995241 0.9657746 0.9885775 1.2399520 1.0326449
## [470] 1.3195545 1.0696822 1.1345612 1.3397083 1.0263855 1.0704554 1.0053051
## [477] 1.1020905 1.0132420 1.1468438 1.3856037 0.9774703 1.0187438 1.1004398
## [484] 1.0946014 1.0684148 1.0305858 0.9833160 1.0736620 1.0423099 1.0689004
## [491] 1.0459239 1.1498370 1.0038621 1.0389985 0.9730947 1.0560618 1.1273801
## [498] 1.2188858 1.0103403 1.0709333 1.0228220 1.0510950 1.0184951 1.4609758
## [505] 1.0635580 1.3877042 1.0045244 1.2426502 1.0248707 1.0056254 0.9834993
## [512] 1.0062921 0.9660655 0.9925610 0.9635593 0.9856036 1.3097304 1.1643811
## [519] 1.0447378 1.9518776 1.0035070 0.9790542 1.0166563 1.0529972 1.2170195
## [526] 1.0669603 1.3688542 1.2814918 1.0141841 0.9811145 1.0081243 1.0370307
## [533] 1.0761644 1.3374429 1.0426483 1.2075501 1.0648116 1.0065172 1.0119261
## [540] 1.1162424 1.0707156 1.1803482 1.0288762 1.1410294 1.0337221 1.0087169
## [547] 0.9916397 0.9802044 1.1541607 1.4432978 1.1202533 1.0089844 0.9812433
## [554] 0.9561498 0.9851213 1.0291243 1.3926354 1.0718699 0.9889277 1.0974761
## [561] 1.0131651 1.0077566 0.9793564 1.0172115 0.9711823 1.1535213 1.0605011
## [568] 1.0516300 1.0047677 0.9629523 1.2057347 1.0615985 1.0009550 1.0360280
## [575] 1.0185912 1.0264614 1.0310560 1.5937547 1.0486723 1.0591770 4.2382897
## [582] 1.1966075 1.1625672 1.0036927 0.9915464 0.9867868 1.0093720 1.0081381
## [589] 0.9772677 1.0836528 1.0880515 1.0297665 1.0200910 1.0536863 1.2898953
## [596] 1.0177875 1.0087090 1.0218583 1.0764164 1.0413310 1.0084858 1.1269785
## [603] 1.0156970 1.0087854 1.0382801 1.0040679 1.0369606 0.9983303 1.0029689
## [610] 0.9968897 1.2144893 0.9840248 0.9856978 0.9971793 1.0710014 1.0573352
## [617] 1.0002604 1.0609848 7.3220520 1.1380486 0.9815292 0.9727417 1.1583611
## [624] 1.2708569 1.0221824 1.0063558 0.9837231 0.9911841 1.2054781 0.9944349
## [631] 1.0321372 0.9977260 1.0077058 1.1326280 1.0025956 0.9866378 1.0552356
## [638] 1.0302064 1.0276442 1.3259123 1.0160705 0.9996940 1.0005378 1.0339254
## [645] 1.0132566 1.0167865 0.9931817 0.9858083 1.2370792 1.0778218 1.1896613
## [652] 0.9968706 1.0243149 1.2155019 1.0939158 1.1844107 1.0866330 1.1995043
## [659] 1.0606121 1.0508463 0.9681459 0.9860271 1.0695815 1.1094119 0.9763116
## [666] 0.9864141 1.1648564 0.9950696 1.0288047 1.0196918 0.9840839 1.1546474
## [673] 0.9924850 1.1763810 1.0063768 0.9762670 1.0638969 1.0144675 1.0132549
## [680] 1.0194510 1.0575572 1.0778957 1.0050773 1.2161174 1.0906200 0.9764446
## [687] 1.5341184 1.1432272 1.0360148 1.0376493 1.0102114 0.9987022 1.0126925
## [694] 1.0075872 1.0007035 1.0750525 1.0153091 1.0170392 1.1260024 0.9535692
## [701] 1.0687025 1.0856806 1.0044923 1.0086735 1.4143607 0.9782802 1.2445775
## [708] 1.1807553 1.0185318 0.9691683 0.9824138 0.9464486 1.0338908 1.0892135
## [715] 0.9787250 0.9842926 1.0986530 1.0149301 1.0431164 1.0096337 1.1447525
## [722] 0.9892045 1.0434144 0.9948280 1.0515876 1.0484922 1.1017171 1.1047400
## [729] 0.9800752 0.9654431 0.9753930 0.9582700 1.0231590 1.0323722 1.0617539
## [736] 1.6292530 1.0038605 1.1026378 1.2037967 1.0781636 1.0237629 1.2100024
## [743] 0.9751365 1.0959747 1.0178401 0.9909123 1.0232132 0.9815586 1.0673657
## [750] 1.0207401 1.1381464 0.9986293 1.0431753 0.9631941 1.0138606 1.0425954
## [757] 0.9902437 1.0281492 0.9739677 1.0317531 1.0423040 1.0757644 0.9823034
## [764] 1.0146865 0.9881496 1.1366843 1.1291484 1.0579229 0.9883354 1.0018877
## [771] 1.0170715 1.1730072 0.9534513 1.0745439 1.0360475 1.0876241 0.9663949
## [778] 1.0405971 1.0240479 1.0897889 1.1712160 1.0089757 1.0572490 1.0634080
## [785] 1.0285556 0.9758924 1.6725776 1.3733251 0.9454878 1.0653647 1.3307922
## [792] 1.0576813 0.9741730 1.1236857 1.0826346 1.0797518 0.9809344 0.9987712
## [799] 1.0473324 1.1620525 1.0254505 1.0514404 0.9882108 1.1373046 1.2100058
## [806] 1.0824997 1.0113614 1.0679669 1.0141413 1.0280397 1.0580221 1.0030706
```

```
## [813] 1.0510272 1.1343277 1.8063230 1.3004030 1.0059051 0.9972065 1.1353984
## [820] 0.9861923 1.0614838 1.0886189 1.3183589 1.0305846 1.1308191 1.0387196
## [827] 1.0262654 1.0141392 1.0414720 1.0669359 1.0248249 0.9786253 1.1712553
## [834] 1.0778066 1.0514808 1.0926923 1.0097366 1.0064897 0.9827597 0.9866504
## [841] 1.0590335 1.0484697 1.0591964 1.0968599 1.0876035 1.0554484 1.1493213
## [848] 1.0678045 1.0578915 1.1318099 1.0827058 0.9993336 1.1684106 1.1075509
## [855] 0.9877014 1.4133050 1.0145561 1.0473551 1.7119995 1.0478900 1.0325222
## [862] 1.0459093 1.0181239 1.0359605 1.0639474 1.0083003 1.1061334 0.9922888
## [869] 1.2020277 1.1012068 1.1003768 1.0036653 1.2796509 1.1771345 1.3545812
## [876] 1.0333214 1.0945959 1.0416362 1.2689260 1.0103737 1.1573883 1.0217002
## [883] 0.9902162 0.9564204 1.1881265 1.0346652 0.9820663 1.1430713 1.1418755
## [890] 1.0057725 1.0437466 1.2941310 0.9841797 1.2029818 1.0285486 0.9894848
## [897] 0.9881815 1.1340093 1.0253844 1.0496553 1.3667986 0.9922547
```

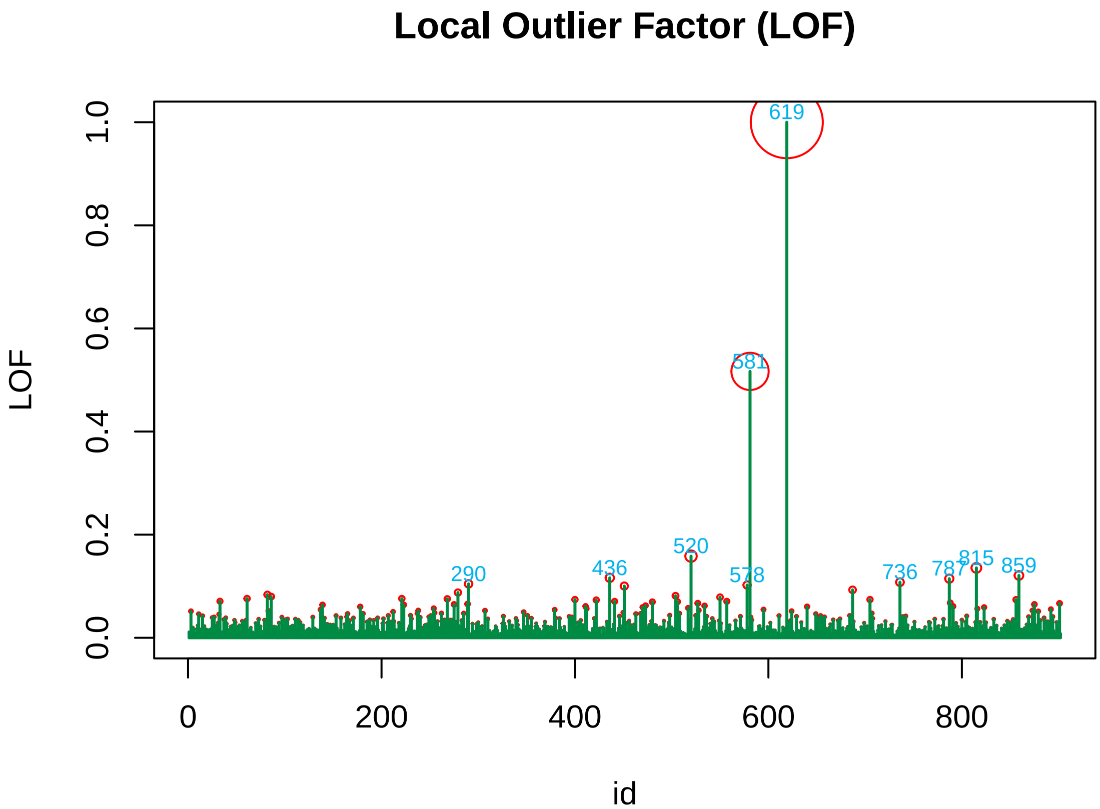
```r
which(outlier.scores > quantile(outlier.scores, 0.95))
```

```
##  [1]  33  61  82  86 139 178 221 223 268 275 279 289 290 400 411 422 436 441 451
## [20] 470 473 480 504 506 517 520 527 534 550 557 578 581 619 640 687 705 736 787
## [39] 788 791 815 823 856 859 875 901
```

```r
score <- scale(outlier.scores, center = min(outlier.scores),
    scale = max(outlier.scores)-min(outlier.scores)) # NORMALIZED TO RANGE[0,1]
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="red")
add.seg <- function(x) segments(x0=x[1], y0=0, x1=x[1], y1=x[2],
    lty=1, lwd=1.5, col="springgreen4")
apply(data.frame(id=1:length(score), score=score), 1, FUN=add.seg)
```

```
## NULL
```

```r
eps <- 0.99
id.outliers <- which(outlier.scores > quantile(outlier.scores, eps))
text(id.outliers, score[id.outliers]+0.02, label=id.outliers,
    col="deepskyblue2", cex=0.7)
```



From the graph, using a neighborhood size, k=5 and a probability of 0.99, We can observe that the defective parts,581 and 619 are the top most outliers

COMPARING LOCAL OUTLIER FACTOR (LOF) AND IFOREST

```
score <- scale(pred, center = min(pred), scale = max(pred)-min(pred))
par(mfrow=c(1,2), mar=rep(4,4))
plot(x=1:length(score), score, type="p", pch=21,
     main="Anomaly Score via iForest",
     xlab="id", ylab="score", cex=score*3, 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)
```

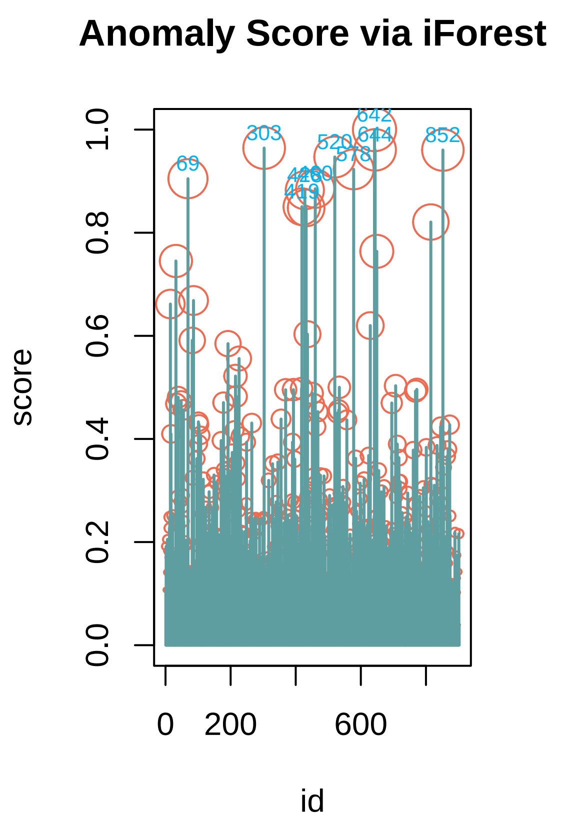
```
## NULL
```

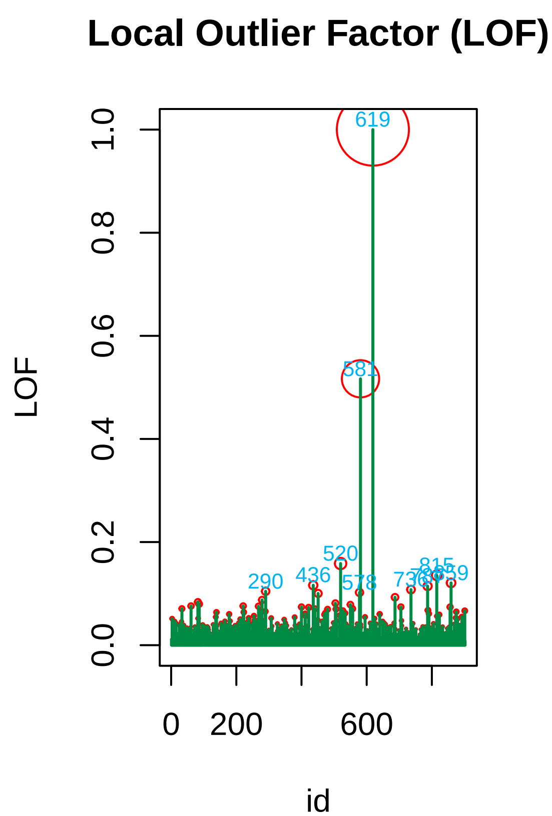
```
eps <- 0.99
id.outliers <- which(score > quantile(score, eps))
text(id.outliers, score[id.outliers]+0.03, label=id.outliers,
     col="deepskyblue2", cex=0.7)



score <- scale(outlier.scores, center = min(outlier.scores),
    scale = max(outlier.scores)-min(outlier.scores)) # NORMALIZED TO RANGE[0,1]
#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="red")
add.seg <- function(x) segments(x0=x[1], y0=0, x1=x[1], y1=x[2],
    lty=1, lwd=1.5, col="springgreen4")
apply(data.frame(id=1:length(score), score=score), 1, FUN=add.seg)
```

```
## NULL
```

```
eps <- 0.99
id.outliers <- which(outlier.scores > quantile(outlier.scores, eps))
text(id.outliers, score[id.outliers]+0.02, label=id.outliers,
    col="deepskyblue2", cex=0.7)
```





From the above, It is evident that the local outlier factor(LOF) was able to detect the the defective parts clearly as top most outliers whiles the iForest was unable to detect the defective parts,581 and 619 clearly among the top outliers so in comparism with respect to detecting the two defective parts, 581 and 619, LOF is a better method.