Real Data Power Analysis

2023-05-11

R functions and libraries

```r
library(nout)
library(R.matlab)
library(isotree)
library(tictoc)

SIM_realdata = function(B, dataset, n1, n, m, l, in_index, out_index=NULL,
                        alpha, lambda = 0.5){

  n0=n-n1
  if(n1!=0 & is.null(out_index)){
    stop("Error: arg out_index must be initialized.")
  }

  # if(m!=(m1+m0)){
  #   stop("Error: equation m=m1+m0 must be verified.")
  # }

  if(n1!=0){
    tr_ind = sample(in_ind, size = l)
    tr = dataset[tr_ind,]
    iso.fo = isolation.forest(tr, ndim=ncol(dataset), ntrees=10, nthreads=1,
                              scoring_metric = "depth", output_score = TRUE)
    in_index2 = setdiff(in_ind, tr_ind)
    mycrit = nout::critWMW(m=n,n=m,alpha=alpha)

    d_WMW = rep(0,B)
    d_Simes = rep(0,B)
    d_StoSimes = rep(0,B)
    d_BH = rep(0,B)
    d_StoBH = rep(0,B)

    for (b in 1:B){
      cal_ind = sample(in_index2, size = m)
      in_index3 = setdiff(in_index2, cal_ind)
      tein_ind = sample(in_index3, size = n0)
      teout_ind = sample(out_index, size = n1)

      cal = dataset[cal_ind,]
      te = dataset[c(tein_ind, teout_ind),]

      S_cal = predict.isolation_forest(iso.fo$model, cal, type = "score")
      S_te = predict.isolation_forest(iso.fo$model, te, type = "score")
      S_Y = S_te
      S_X = S_cal

      # U_i = sapply(1:n, function(i) sum(S_Y[i]>S_X))
      # U = sum(U_i)

      d_WMW[b] = nout::d_mannwhitney(S_Y=S_Y, S_X=S_X, crit = mycrit)>0
      #res[b] = U >= mycrit
      d_Simes[b] = nout::d_Simes(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_StoSimes[b] = nout::d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=alpha)$d
      d_BH[b] = nout::d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_StoBH[b] = nout::d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=alpha)
    }
  }

  else{

    tr_ind = sample(in_ind, size = l)
    tr = dataset[tr_ind,]
    iso.fo = isolation.forest(tr, ndim=ncol(dataset), ntrees=10, nthreads=1,
                              scoring_metric = "depth", output_score = TRUE)
    in_index2 = setdiff(in_ind, tr_ind)
    mycrit = nout::critWMW(m=n,n=m,alpha=alpha)

    d_WMW = rep(0,B)
    d_Simes = rep(0,B)
    d_StoSimes = rep(0,B)
    d_BH = rep(0,B)
    d_StoBH = rep(0,B)

    for (b in 1:B){
      cal_ind = sample(in_index2, size = m)
      in_index3 = setdiff(in_index2, cal_ind)
      te_ind = sample(in_index3, size = n)
      cal = dataset[cal_ind,]
      te = dataset[te_ind,]

      S_cal = predict.isolation_forest(iso.fo$model, cal, type = "score")
      S_te = predict.isolation_forest(iso.fo$model, te, type = "score")

      S_Y = S_te
      S_X = S_cal

      # U_i = sapply(1:n, function(i) sum(S_Y[i]>S_X))
      # U = sum(U_i)

      d_WMW[b] = nout::d_mannwhitney(S_Y=S_Y, S_X=S_X, crit = mycrit)>0
      #res[b] = U >= mycrit
      d_Simes[b] = nout::d_Simes(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_StoSimes[b] = nout::d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=alpha)$d
      d_BH[b] = nout::d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_StoBH[b] = nout::d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=alpha)
    }
  }

  discov = as.data.frame(cbind("d_BH"=d_BH, "d_StoBH"=d_StoBH, "d_Simes"=d_Simes,
                               "d_StoSimes"=d_StoSimes, "d_WMW"=d_WMW))
  colnames(discov) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
  mean.discov = apply(discov, MARGIN = 2, FUN = mean)

  powerGlobalNull = as.data.frame(cbind("d_BH"=d_BH>0, "d_StoBH"=d_StoBH>0, "d_Simes"=d_Simes>0,
                                        "d_StoSimes"=d_StoSimes>0, "d_WMW"=d_WMW>0))
  colnames(powerGlobalNull) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
  mean.powerGlobalNull = apply(powerGlobalNull, MARGIN = 2, FUN = mean)

  return(list("discoveries"=discov, "mean.discoveries" = mean.discov, "powerGlobalNull"=powerGlobalNull
              "mean.powerGlobalNull"=mean.powerGlobalNull, "n1"=n1, "alpha"=alpha))
}
```

Digits dataset

The aim is to compare on Digits dataset (available at http://odds.cs.stonybrook.edu/pendigits-dataset) the performance of three closed testing procedures, which respectively use Simes local test with and without Storey estimator for the proportion of true null hypotheses and Wilcoxon-Mann-Whitney local test.

Digits dataset consists of 6870 observations, among which $n_{inliers} = 6714$ items are inliers and the remaining $n_{outliers} = 156$ are outliers. We will denote by n, l, m respectively the train set, the calibration set and the test set size. And reproducing the same setting as in [1], we have that $m + n = n_{inliers}/2$, $m = min\{2000, l/2\}$ and $n = min\{2000, l/3\}$. In the case of Digits dataset we obtain

$$m + n = 6714/2, \quad m = l/3, \quad n = l/3$$

from which $n = 2517.75, \quad l = 839.25, \quad m = 839.25$. Arbitrarily, we choose to set

$$n = 1258, \quad l = 2099, \quad m = 420$$

in order to have exact control of type I errors at the significance level $\alpha = 0.2$.

Load the data and set the parameters as described above.

```
set.seed(321)

# Initializing parameters
l = 2518
m = 2099
n = 420
myalpha=0.2
#alpha = n/(m+1)

data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)
```

We fixed the train set on which we train the isolation forest algorithm and we generate $B = 10^4$ calibration and test sets. For each $b = 1, \dots, B$ we compute the number of discoveries obtained by Benjamini-Hochberg procedure with and without Storey's estimator for the proportion of true null hypotheses, by closed testing using Simes local test with and without Storey's estimator and by closed testing using Wilcoxon-Mann-Whitney local test.

All inliers

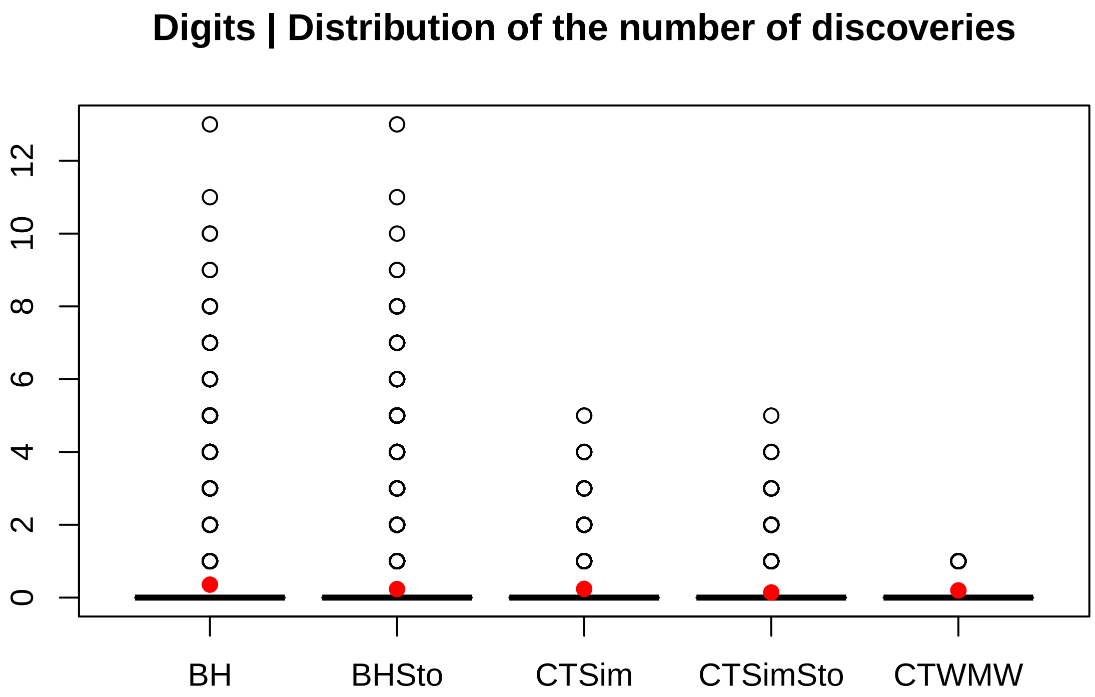
We now set the proportion of inliers equal to 1, so that the number of outliers $n_1 = 0$.

```
B=10^4
n1=0

tic()
res = SIM_realdata(B=B, dataset=dataset, n1=n1, n=n, m=m, l=l, in_index=in_ind,
                   out_index=out_ind, alpha=myalpha, lambda = 0.5)
toc()
```

```
## 610.77 sec elapsed
```

```
boxplot(res$discoveries, main="Digits | Distribution of the number of discoveries")
points(x=1:5, y=res$mean.discoveries, pch=19, col="red")
```



```
res$mean.discoveries
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.3562   0.2341   0.2396   0.1420   0.1970
```

```
res$mean.powerGlobalNull
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.1968   0.1152   0.1968   0.1152   0.1970
```

```
resDigits_exact0 = res
save(resDigits_exact0,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&Tidy/resDigits_exact0")
```

10% outliers

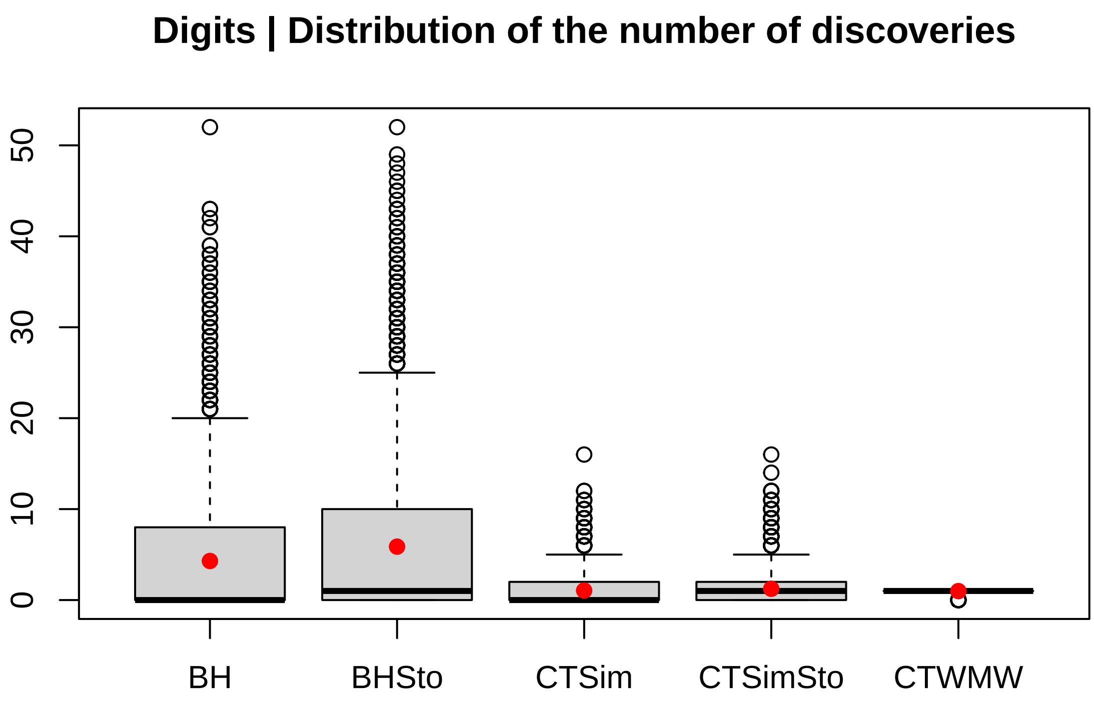
We now set the proportion of inliers equal to 0.9. Referring to Digits dataset we have that the number of inliers is $n_0 = 378$ and the number of outliers is $n_1 = 42$.

```
B=10^4
n1=round(0.1*n)

tic()
res = SIM_realdata(B=B, dataset=dataset, n1=n1, n=n, m=m, l=l, in_index=in_ind,
                   out_index=out_ind, alpha=myalpha, lambda = 0.5)
toc()
```

```
## 609.14 sec elapsed
```

```
boxplot(res$discoveries, main="Digits | Distribution of the number of discoveries")
points(x=1:5, y=res$mean.discoveries, pch=19, col="red")
```



```
res$mean.discoveries
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   4.2935   5.8736   1.0500   1.2383   0.9830
```

```
res$mean.powerGlobalNull
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.4668   0.5043   0.4668   0.5042   0.9830
```

```
resDigits_exact10 = res
save(resDigits_exact10,
    file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&Tidy/resDigits_exact10")
```

Credit Card Fraud Detection dataset

The aim is to compare on Digits dataset (available at https://www.kaggle.com/mlg-ulb/creditcardfraud) the performance of three closed testing procedures, which respectively use Simes local test with and without Storey estimator for the proportion of true null hypotheses and Wilcoxon-Mann-Whitney local test.

Credit card dataset consists of 284807 observations, among which $n_{inliers} = 284315$ items are inliers and the remaining $n_{outliers} = 492$ are outliers.

In the case of Credit Card dataset we obtain

$$m + n = 284315/2, \quad m = 2000, \quad n = 2000.$$

Arbitrarily, we choose to set

$$l = 132158, \quad m = 9999, \quad n = 2000$$

in order to have exact control of type I errors at the significance level $\alpha = 0.2$.

Load the data and set the parameters ad described above.

```
set.seed(321)

# Initializing parameters
l = 132158
m = 9999
n = 2000
myalpha = 0.2

dataset = read.csv("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazion
out_ind = which(dataset$Class==1)
in_ind = which(dataset$Class==0)
```

We fixed the train set on which we train the isolation forest algorithm and we generate $B = 10^4$ calibration and test sets. For each $b = 1, \dots, B$ we compute the number of discoveries obtained by Benjamini-Hochberg procedure with and without Storey's estimator for the proportion of true null hypotheses, by closed testing using Simes local test with and without Storey's estimator and by closed testing using Wilcoxon-Mann-Whitney local test.

All inliers

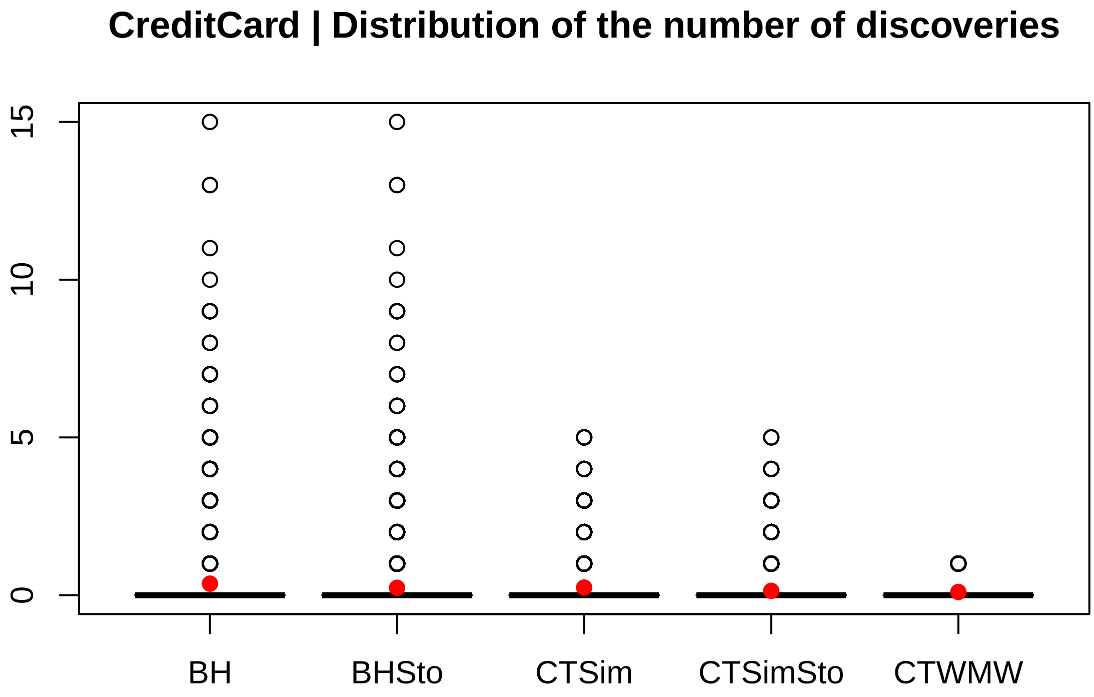
We now set the proportion of inliers equal to 1, so that the number of outliers $n_1 = 0$.

```
B=10^4
n1=0

tic()
res = SIM_realdata(B=B, dataset=dataset, n1=n1, n=n, m=m, l=l, in_index=in_ind,
                   out_index=out_ind, alpha=myalpha, lambda = 0.5)
toc()
```

```
## 8000.93 sec elapsed
```

```
boxplot(res$discoveries, main="CreditCard | Distribution of the number of discoveries")
points(x=1:5, y=res$mean.discoveries, pch=19, col="red")
```



```
res$mean.discoveries
```

```
##        BH     BHSto     CTSim  CTSimSto     CTWMW
##    0.3654    0.2329    0.2414    0.1372    0.1016
```

```
res$mean.powerGlobalNull
```

```
##        BH     BHSto     CTSim  CTSimSto     CTWMW
##    0.1951    0.1110    0.1951    0.1110    0.1016
```

```
resCredit_exact0 = res
save(resCredit_exact0,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&Tidy/resCredit_exact0")
```

10% outliers

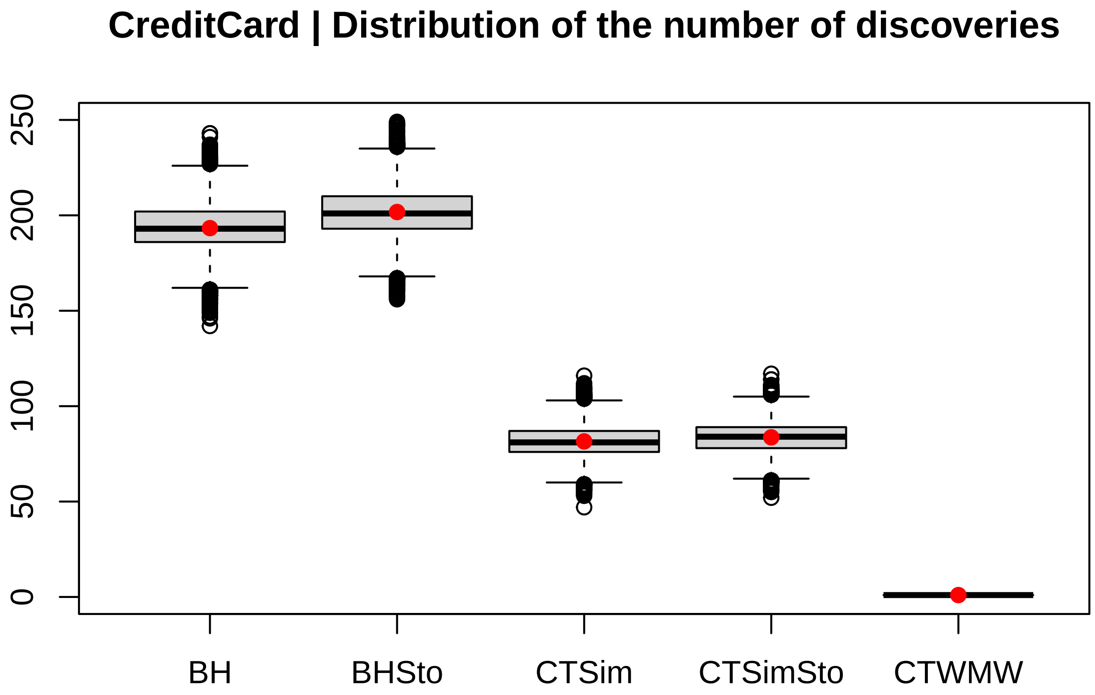
We now set the proportion of inliers equal to 0.9. Referring to Credit Card dataset we have that the number of inliers is $n_0 = 1800$ and the number of outliers is $n_1 = 200$.

```
B=10^4
n1=round(0.1*n)

tic()
res = SIM_realdata(B=B, dataset=dataset, n1=n1, n=n, m=m, l=l, in_index=in_ind,
                   out_index=out_ind, alpha=myalpha, lambda = 0.5)
toc()
```

```
## 7753.08 sec elapsed
```

```
boxplot(res$discoveries, main="CreditCard | Distribution of the number of discoveries")
points(x=1:5, y=res$mean.discoveries, pch=19, col="red")
```



```
res$mean.discoveries
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
## 193.3018 201.7384  81.5388  83.6672   1.0000
```

```
res$mean.powerGlobalNull
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##        1        1        1        1        1
```

```
resCredit_exact10 = res
save(resCredit_exact10,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&Tidy/resCredit_exact10")
```

Statlog (Shuttle) dataset

The aim is to compare on Digits dataset (available at http://odds.cs.stonybrook.edu/shuttle-dataset) the performance of three closed testing procedures, which respectively use Simes local test with and without Storey estimator for the proportion of true null hypotheses and Wilcoxon-Mann-Whitney local test.

Shuttle dataset consists of 49097 observations, among which $n_{inliers} = 45586$ items are inliers and the remaining $n_{outliers} = 3511$ are outliers. In the case of Digits dataset we obtain

$$m + n = 45586/2, \quad m = 2000, \quad n = 2000$$

Arbitrarily, we choose to set

$$l = 12794, \quad m = 9999, \quad n = 2000$$

in order to have exact control of type I errors at the significance level $\alpha = 0.2$.

Load the data and set the parameters ad described above.

```
set.seed(321)

# Initializing parameters
l = 12794
m = 9999
n = 2000
myalpha = 0.2

data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
out_ind = which(dataset[,ncol(dataset)]==1)
in_ind = which(dataset[,ncol(dataset)]==0)
```

We fixed the train set on which we train the isolation forest algorithm and we generate $B = 10^4$ calibration and test sets. For each $b = 1, \dots, B$ we compute the number of discoveries obtained by Benjamini-Hochberg procedure with and without Storey's estimator for the proportion of true null hypotheses, by closed testing using Simes local test with and without Storey's estimator and by closed testing using Wilcoxon-Mann-Whitney local test.

All inliers

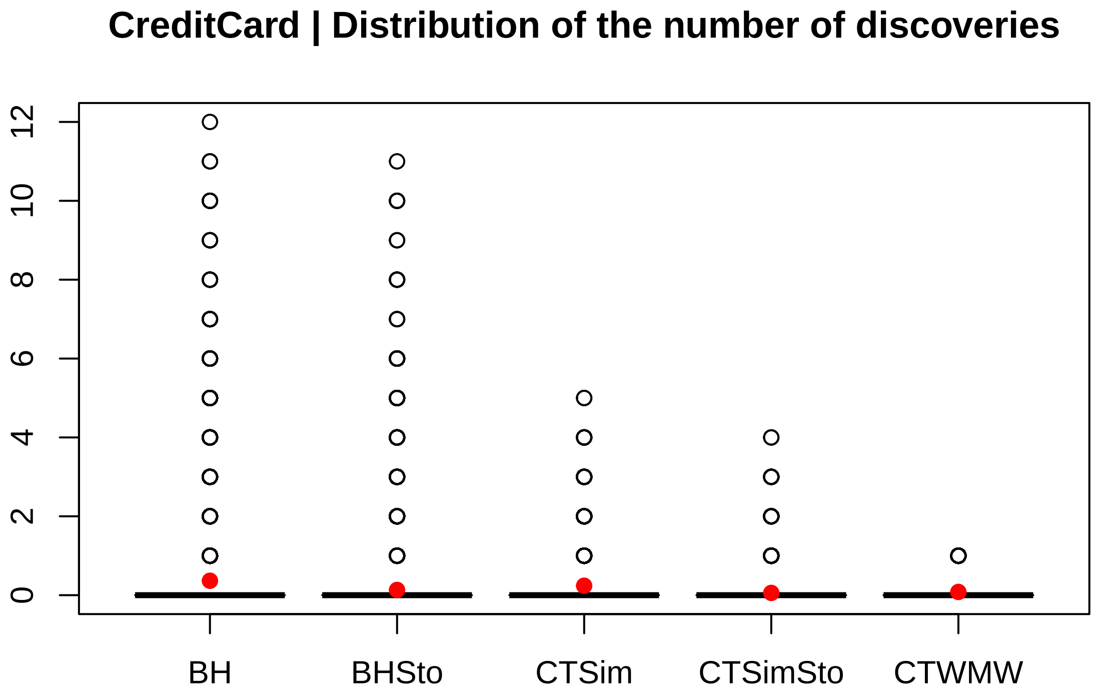
We now set the proportion of inliers equal to 1, so that the number of outliers $n_1 = 0$.

```
B=10^4
n1=0

tic()
res = SIM_realdata(B=B, dataset=dataset, n1=n1, n=n, m=m, l=l, in_index=in_ind,
                   out_index=out_ind, alpha=myalpha, lambda = 0.5)
toc()
```

```
## 5956.72 sec elapsed
```

```
boxplot(res$discoveries, main="CreditCard | Distribution of the number of discoveries")
points(x=1:5, y=res$mean.discoveries, pch=19, col="red")
```



```
res$mean.discoveries
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.3656   0.1325   0.2393   0.0570   0.0814
```

```
res$mean.powerGlobalNull
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.1936   0.0470   0.1936   0.0470   0.0814
```

```
resCredit_exact0 = res
save(resCredit_exact0,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&Tidy/resShuttle_exact0")
```

10% outliers

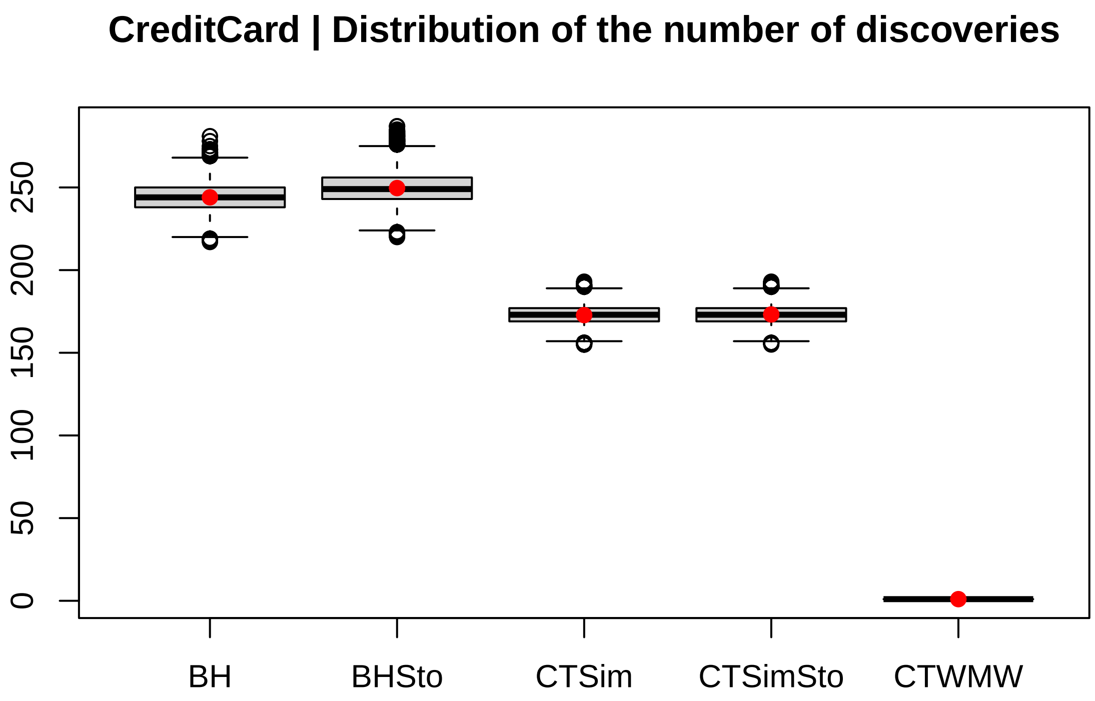
We now set the proportion of inliers equal to 0.9. Referring to Shuttle dataset we have that the number of inliers is $n_0 = 1800$ and the number of outliers is $n_1 = 200$.

```
B=10^4
n1=round(0.1*n)

tic()
res = SIM_realdata(B=B, dataset=dataset, n1=n1, n=n, m=m, l=l, in_index=in_ind,
                   out_index=out_ind, alpha=myalpha, lambda = 0.5)
toc()
```

```
## 5999.13 sec elapsed
```

```
boxplot(res$discoveries, main="CreditCard | Distribution of the number of discoveries")
points(x=1:5, y=res$mean.discoveries, pch=19, col="red")
```



```
res$mean.discoveries
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
## 244.0126 249.6119 172.7924 173.1773   1.0000
```

```
res$mean.powerGlobalNull
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##        1        1        1        1        1
```

```
resCredit_exact10 = res
save(resCredit_exact10,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&Tidy/resShuttle_exact10")
```

References

[1] Bates, S., E. Candes, L. Lei, Y. Romano, and M. Sesia (2023). Testing for outliers with conformal p-values. *Annals of Statistics*, **{51}, 149–178.**