Comparison between different local tests: Simes, Simes with Storey and Wilcoxon-Mann-Whitney

27-04-2023

The aim is to compare on real datasets 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.

R functions and libraries

```
library(nout)
library(R.matlab)
```

```
## R.matlab v3.7.0 (2022-08-25 21:52:34 UTC) successfully loaded. See ?R.matlab for help.

##
## Caricamento pacchetto: 'R.matlab'

## I seguenti oggetti sono mascherati da 'package:base':
##
##     getOption, isOpen
```

```
library(isotree)
library(farff)
library(tictoc)
```

```
sim_realdata = function(B, dataset, m1, m, n, l, in_index, out_index=NULL, alpha=m/(l+1), lambda = 0.5)

  m0=m-m1
  if(m1!=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(m1!=0){
    tr_ind = sample(in_index, size = n)
    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_index, tr_ind)

  crit=critWMW(m=m, n=n, 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 = l)
    in_index3 = setdiff(in_index2, cal_ind)
    tein_ind = sample(in_index3, size = m0)
    teout_ind = sample(out_index, size = m1)

    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")

    d_WMW[b] = d_mannwhitney(S_X=S_cal, S_Y=S_te, crit=crit)
    d_Simes[b] = d_Simes(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_StoSimes[b] = d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_BH[b] = d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_StoBH[b] = d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=alpha)
  }
}

else{
  tr_ind = sample(in_index, size = n)
  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_index, tr_ind)

  crit=critWMW(m=m, n=n, 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 = l)
    in_index3 = setdiff(in_index2, cal_ind)
    te_ind = sample(in_index3, size = m0)

    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")

      d_WMW[b] = d_mannwhitney(S_X=S_cal, S_Y=S_te, crit=crit)
      d_Simes[b] = d_Simes(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_StoSimes[b] = d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_BH[b] = d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=alpha)
      d_StoBH[b] = 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,
              "m1"=m1, "alpha"=alpha))

}
```

In the following we set the calibration set and the test set size, respectively l and m, so that the nominal level α is proportional to $\frac{m}{l+1}$. The train set size is equal to n and the number of iterations is $B = 10^3$.

Statlog (Shuttle) dataset

The dataset is available at http://odds.cs.stonybrook.edu/shuttle-dataset

```
set.seed(321)

# Initializing parameters
B=10^3
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=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)

tic()
```

```
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 194.28 sec elapsed
```

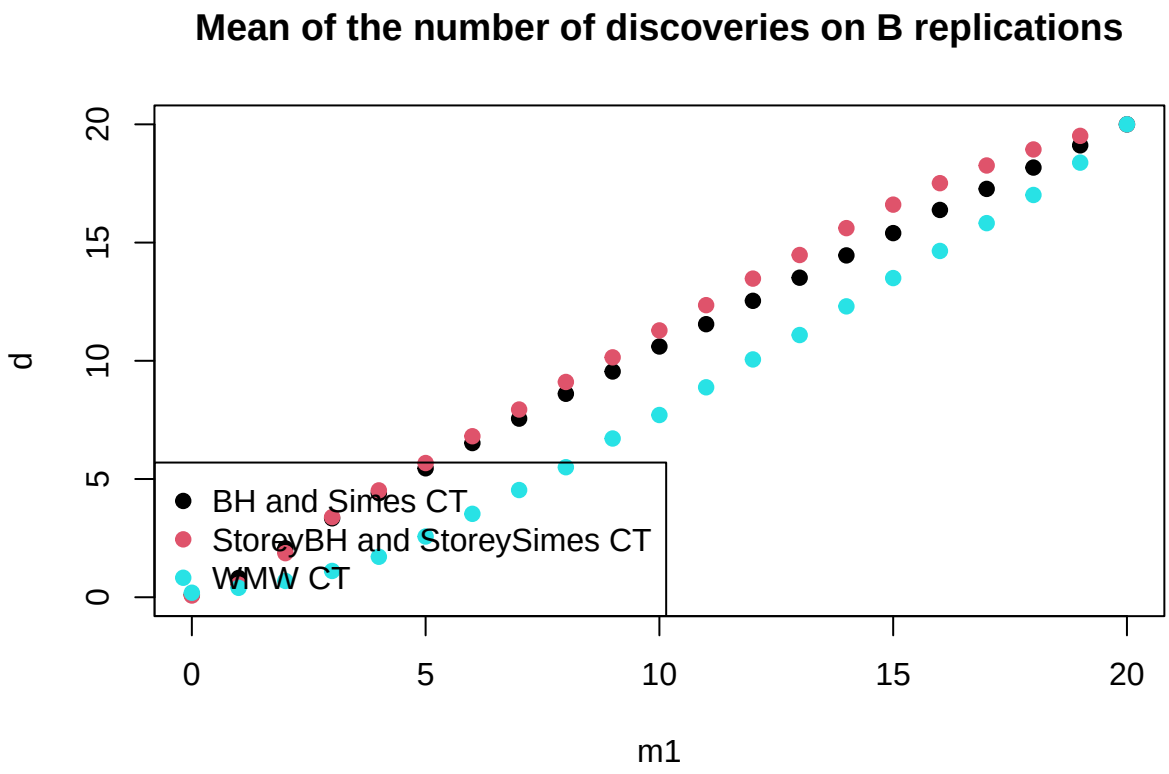
```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("m1 =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

```
##             BH  StoBH  Simes StoSimes    WMW
## m1 = 0   0.105  0.073  0.085    0.050  0.190
## m1 = 1   0.815  0.536  0.692    0.415  0.398
## m1 = 2   2.068  1.864  1.488    1.172  0.690
## m1 = 3   3.339  3.390  2.891    2.574  1.112
## m1 = 4   4.402  4.519  3.358    3.168  1.712
## m1 = 5   5.453  5.676  4.417    4.336  2.568
## m1 = 6   6.521  6.809  5.355    5.364  3.527
## m1 = 7   7.554  7.940  6.657    6.687  4.534
## m1 = 8   8.605  9.104  7.199    7.289  5.499
## m1 = 9   9.548 10.146  8.296    8.346  6.713
## m1 = 10 10.604 11.285  8.897    9.007  7.707
## m1 = 11 11.549 12.353 10.821   10.665  8.879
## m1 = 12 12.537 13.476 11.885   11.760 10.058
## m1 = 13 13.514 14.474 12.625   12.579 11.089
## m1 = 14 14.457 15.611 13.692   13.703 12.302
## m1 = 15 15.400 16.604 15.043   14.950 13.499
## m1 = 16 16.380 17.512 16.063   15.924 14.646
## m1 = 17 17.270 18.260 17.099   17.005 15.822
## m1 = 18 18.172 18.938 18.090   18.015 17.013
## m1 = 19 19.112 19.513 19.112   19.057 18.381
## m1 = 20 20.000 20.000 20.000   20.000 20.000
```

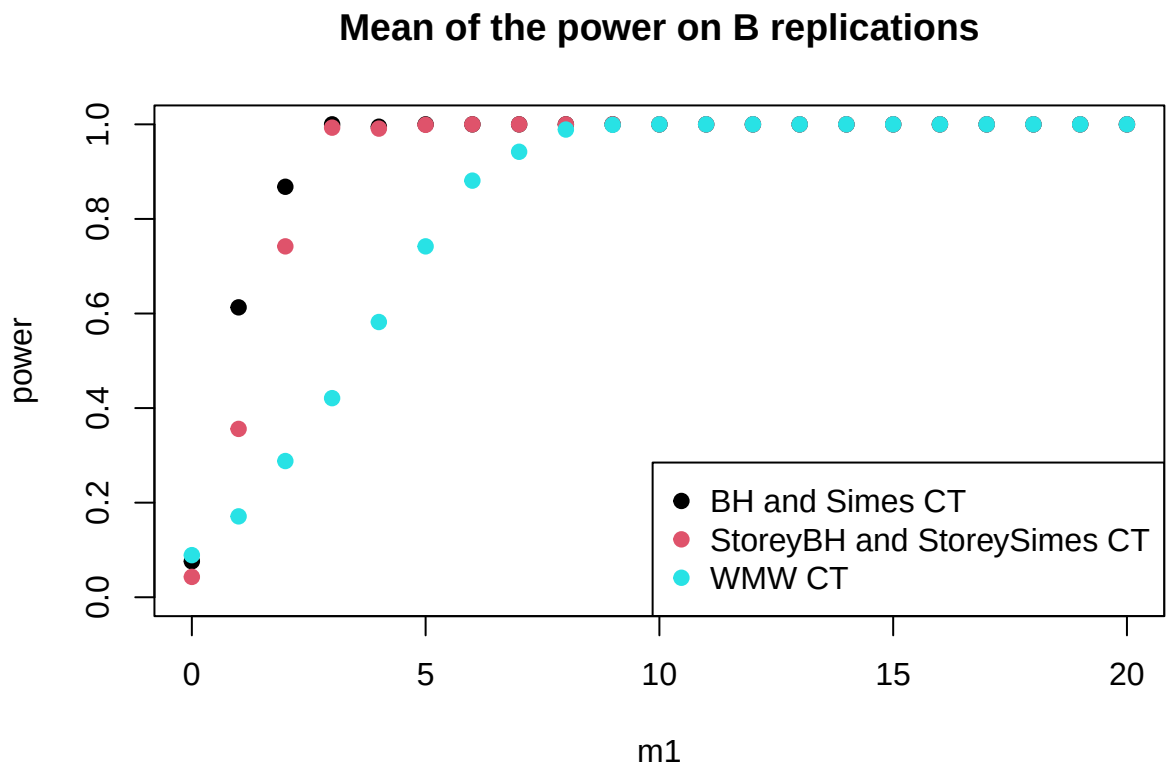
```
store_res$mean.powerGlobalNull
```

```
##            BH StoBH Simes StoSimes   WMW
## m1 = 0  0.076 0.043 0.076    0.043 0.089
## m1 = 1  0.613 0.356 0.613    0.356 0.171
## m1 = 2  0.868 0.742 0.868    0.742 0.288
## m1 = 3  1.000 0.993 1.000    0.993 0.421
## m1 = 4  0.995 0.991 0.995    0.991 0.582
## m1 = 5  1.000 0.999 1.000    0.999 0.742
## m1 = 6  1.000 1.000 1.000    1.000 0.881
## m1 = 7  1.000 1.000 1.000    1.000 0.942
## m1 = 8  1.000 1.000 1.000    1.000 0.989
## m1 = 9  1.000 1.000 1.000    1.000 0.999
## m1 = 10 1.000 1.000 1.000    1.000 1.000
## m1 = 11 1.000 1.000 1.000    1.000 1.000
## m1 = 12 1.000 1.000 1.000    1.000 1.000
## m1 = 13 1.000 1.000 1.000    1.000 1.000
## m1 = 14 1.000 1.000 1.000    1.000 1.000
## m1 = 15 1.000 1.000 1.000    1.000 1.000
## m1 = 16 1.000 1.000 1.000    1.000 1.000
## m1 = 17 1.000 1.000 1.000    1.000 1.000
## m1 = 18 1.000 1.000 1.000    1.000 1.000
## m1 = 19 1.000 1.000 1.000    1.000 1.000
## m1 = 20 1.000 1.000 1.000    1.000 1.000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("bottomleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Credit Card Fraud Detection dataset

The dataset is available at https://www.kaggle.com/mlg-ulb/creditcardfraud.

```
set.seed(321)

# Initializing parameters
B=10^3
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=1)

dataset = read.csv("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 492.39 sec elapsed
```

```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("m1 =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

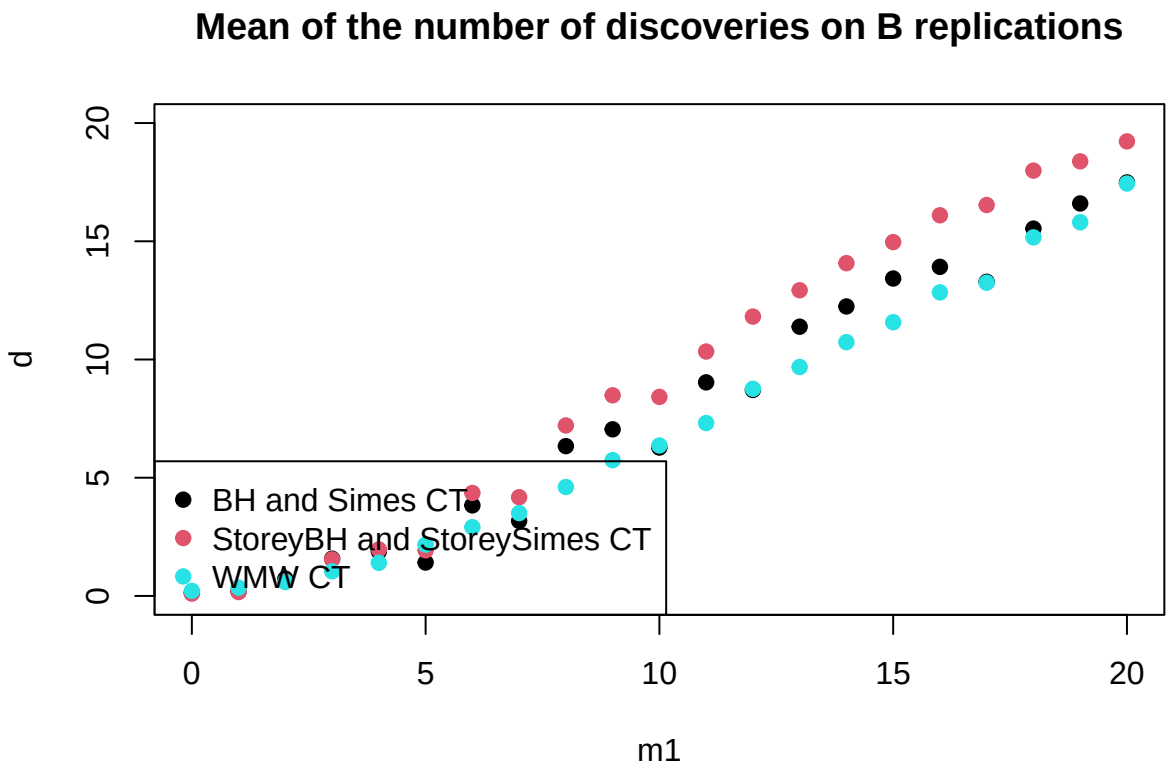
```
##             BH  StoBH  Simes StoSimes    WMW
## m1 = 0   0.131  0.100  0.113    0.075  0.215
## m1 = 1   0.190  0.165  0.140    0.103  0.351
## m1 = 2   0.733  0.620  0.503    0.371  0.588
## m1 = 3   1.576  1.547  1.032    0.893  1.043
## m1 = 4   1.882  1.966  1.078    1.041  1.402
## m1 = 5   1.413  1.928  0.752    0.876  2.170
## m1 = 6   3.831  4.357  2.399    2.481  2.908
## m1 = 7   3.173  4.176  1.641    1.987  3.504
## m1 = 8   6.337  7.212  3.884    4.251  4.608
## m1 = 9   7.047  8.486  3.632    4.514  5.742
## m1 = 10  6.277  8.419  3.017    4.139  6.363
## m1 = 11  9.034 10.338  5.686    6.406  7.314
## m1 = 12  8.711 11.816  4.104    6.154  8.762
## m1 = 13 11.387 12.929  8.404    9.042  9.684
## m1 = 14 12.243 14.075  8.547    9.564 10.734
## m1 = 15 13.425 14.966  9.871   10.817 11.578
## m1 = 16 13.918 16.100  9.861   11.489 12.839
## m1 = 17 13.286 16.533  7.951   11.039 13.248
## m1 = 18 15.535 17.990 11.420   13.535 15.166
## m1 = 19 16.600 18.381 13.155   14.745 15.804
## m1 = 20 17.489 19.225 14.266   16.227 17.443
```

```
store_res$mean.powerGlobalNull
```

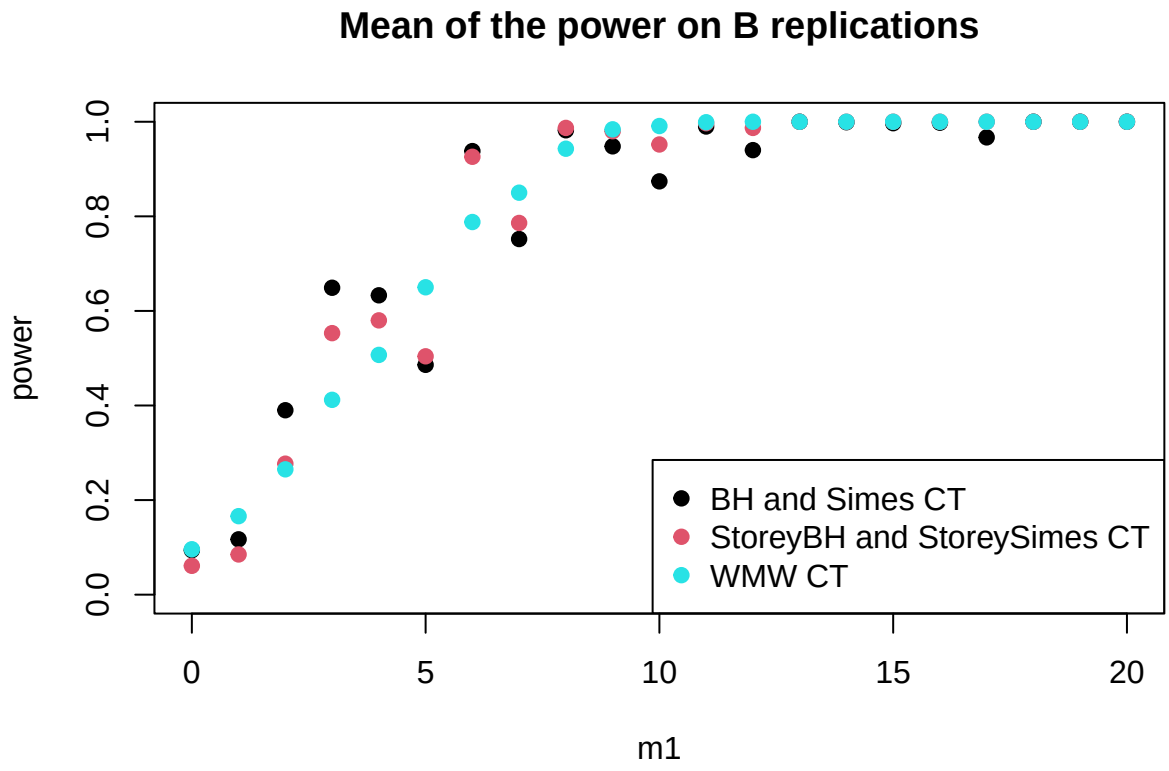
```
##            BH StoBH Simes StoSimes   WMW
## m1 = 0  0.094 0.061 0.094    0.061 0.096
## m1 = 1  0.117 0.085 0.117    0.085 0.166
## m1 = 2  0.390 0.277 0.390    0.277 0.265
## m1 = 3  0.649 0.553 0.649    0.553 0.412
## m1 = 4  0.633 0.580 0.633    0.580 0.507
```

```
## m1 = 5  0.486 0.504 0.486    0.504 0.650
## m1 = 6  0.938 0.926 0.938    0.926 0.788
## m1 = 7  0.752 0.786 0.752    0.786 0.850
## m1 = 8  0.982 0.987 0.982    0.987 0.943
## m1 = 9  0.948 0.980 0.948    0.980 0.984
## m1 = 10 0.874 0.952 0.874    0.952 0.991
## m1 = 11 0.990 0.997 0.990    0.997 0.999
## m1 = 12 0.940 0.987 0.940    0.987 1.000
## m1 = 13 1.000 1.000 1.000    1.000 1.000
## m1 = 14 0.999 1.000 0.999    1.000 1.000
## m1 = 15 0.997 1.000 0.997    1.000 1.000
## m1 = 16 0.998 1.000 0.998    1.000 1.000
## m1 = 17 0.967 1.000 0.967    1.000 1.000
## m1 = 18 1.000 1.000 1.000    1.000 1.000
## m1 = 19 1.000 1.000 1.000    1.000 1.000
## m1 = 20 1.000 1.000 1.000    1.000 1.000
```

```r
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("bottomleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Covertype dataset

The dataset is available at http://odds.cs.stonybrook.edu/forestcovercovertype-dataset.

```
set.seed(321)

# Initializing parameters
B=10^3
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=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"
```

```r
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 186.92 sec elapsed
```

```r
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("theta =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

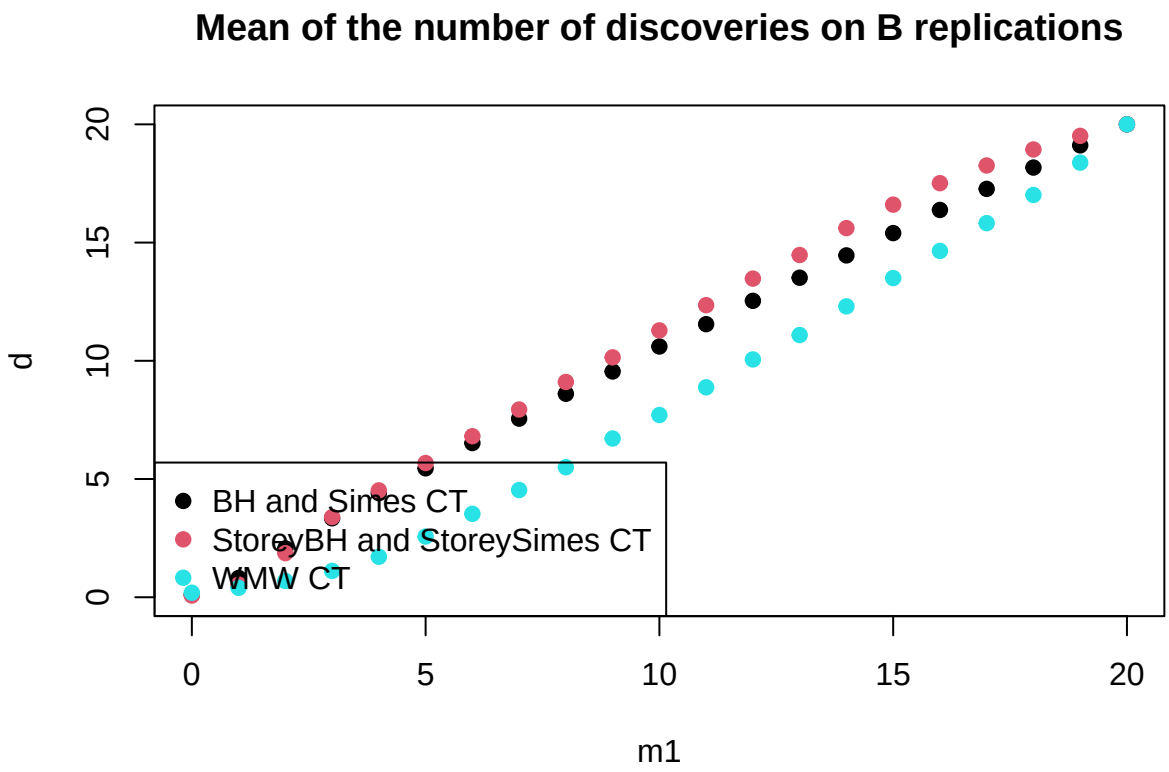
```
##                BH  StoBH  Simes StoSimes    WMW
## theta = 0   0.105  0.073  0.085    0.050  0.190
## theta = 1   0.815  0.536  0.692    0.415  0.398
## theta = 2   2.068  1.864  1.488    1.172  0.690
## theta = 3   3.339  3.390  2.891    2.574  1.112
## theta = 4   4.402  4.519  3.358    3.168  1.712
## theta = 5   5.453  5.676  4.417    4.336  2.568
## theta = 6   6.521  6.809  5.355    5.364  3.527
## theta = 7   7.554  7.940  6.657    6.687  4.534
## theta = 8   8.605  9.104  7.199    7.289  5.499
## theta = 9   9.548 10.146  8.296    8.346  6.713
## theta = 10 10.604 11.285  8.897    9.007  7.707
## theta = 11 11.549 12.353 10.821   10.665  8.879
## theta = 12 12.537 13.476 11.885   11.760 10.058
## theta = 13 13.514 14.474 12.625   12.579 11.089
## theta = 14 14.457 15.611 13.692   13.703 12.302
## theta = 15 15.400 16.604 15.043   14.950 13.499
## theta = 16 16.380 17.512 16.063   15.924 14.646
## theta = 17 17.270 18.260 17.099   17.005 15.822
## theta = 18 18.172 18.938 18.090   18.015 17.013
## theta = 19 19.112 19.513 19.112   19.057 18.381
```

```
## theta = 20 20.000 20.000 20.000   20.000 20.000
```

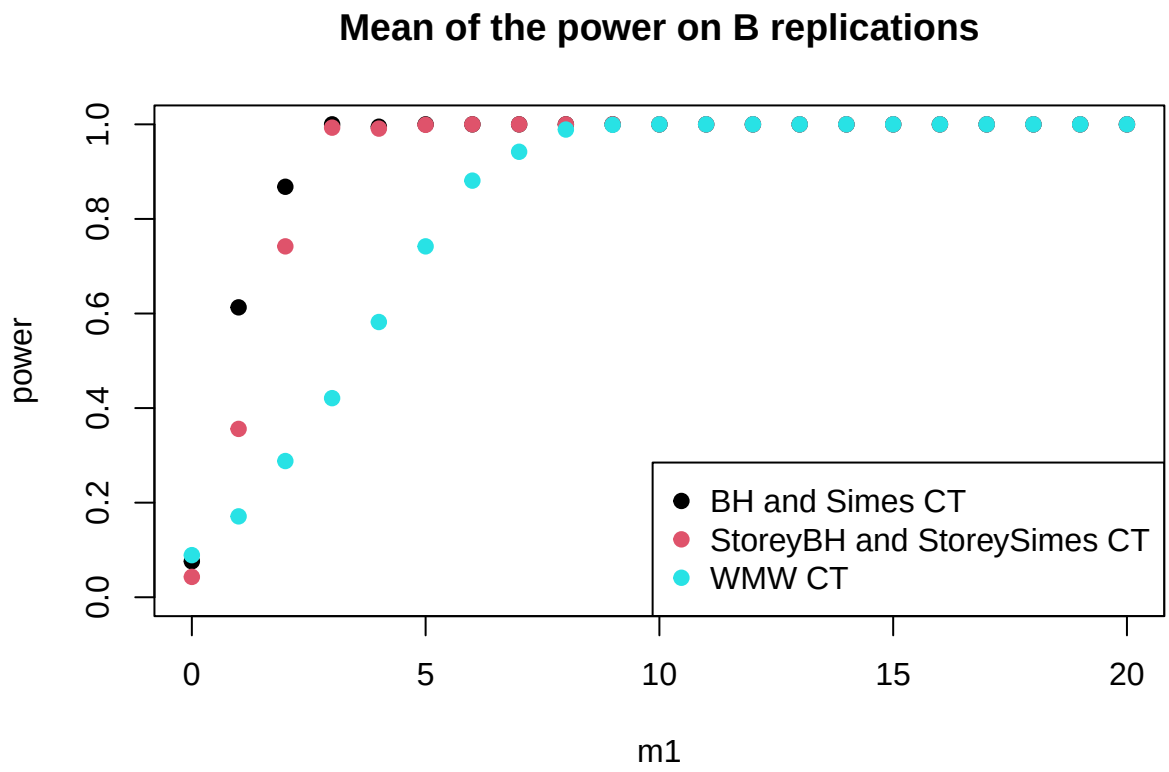
```
store_res$mean.powerGlobalNull
```

```
##               BH StoBH Simes StoSimes   WMW
## theta = 0  0.076 0.043 0.076    0.043 0.089
## theta = 1  0.613 0.356 0.613    0.356 0.171
## theta = 2  0.868 0.742 0.868    0.742 0.288
## theta = 3  1.000 0.993 1.000    0.993 0.421
## theta = 4  0.995 0.991 0.995    0.991 0.582
## theta = 5  1.000 0.999 1.000    0.999 0.742
## theta = 6  1.000 1.000 1.000    1.000 0.881
## theta = 7  1.000 1.000 1.000    1.000 0.942
## theta = 8  1.000 1.000 1.000    1.000 0.989
## theta = 9  1.000 1.000 1.000    1.000 0.999
## theta = 10 1.000 1.000 1.000    1.000 1.000
## theta = 11 1.000 1.000 1.000    1.000 1.000
## theta = 12 1.000 1.000 1.000    1.000 1.000
## theta = 13 1.000 1.000 1.000    1.000 1.000
## theta = 14 1.000 1.000 1.000    1.000 1.000
## theta = 15 1.000 1.000 1.000    1.000 1.000
## theta = 16 1.000 1.000 1.000    1.000 1.000
## theta = 17 1.000 1.000 1.000    1.000 1.000
## theta = 18 1.000 1.000 1.000    1.000 1.000
## theta = 19 1.000 1.000 1.000    1.000 1.000
## theta = 20 1.000 1.000 1.000    1.000 1.000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("bottomleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Mammography dataset

The dataset is available at http://odds.cs.stonybrook.edu/mammography-dataset/.

```
set.seed(321)

# Initializing parameters
B=10^3
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=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)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 69.7 sec elapsed
```

```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("theta =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

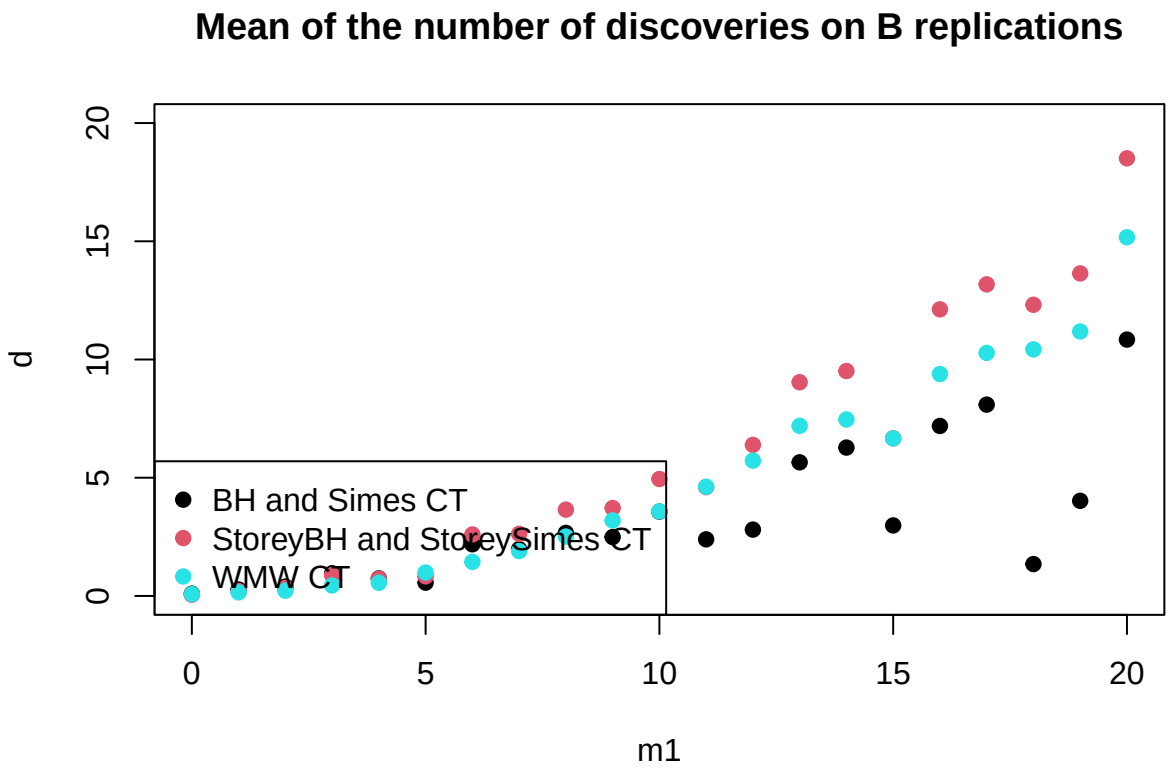
```
##                BH  StoBH Simes StoSimes    WMW
## theta = 0   0.095  0.053 0.078    0.041  0.077
## theta = 1   0.273  0.208 0.210    0.140  0.152
## theta = 2   0.396  0.337 0.289    0.210  0.222
## theta = 3   0.960  0.882 0.676    0.544  0.443
## theta = 4   0.741  0.727 0.557    0.487  0.557
## theta = 5   0.562  0.829 0.329    0.393  0.989
## theta = 6   2.184  2.602 1.126    1.301  1.440
## theta = 7   1.979  2.627 1.015    1.270  1.898
## theta = 8   2.667  3.650 1.415    1.749  2.511
## theta = 9   2.491  3.719 1.469    1.865  3.193
## theta = 10  3.566  4.946 1.916    2.545  3.575
## theta = 11  2.394  4.601 1.123    2.060  4.618
## theta = 12  2.806  6.389 1.405    2.552  5.722
## theta = 13  5.651  9.040 3.234    4.597  7.194
## theta = 14  6.275  9.512 3.424    5.107  7.464
## theta = 15  2.983  6.668 1.567    2.895  6.664
## theta = 16  7.189 12.123 3.581    6.369  9.385
## theta = 17  8.092 13.178 4.417    7.171 10.279
## theta = 18  1.347 12.315 0.431    4.913 10.429
## theta = 19  4.023 13.641 1.761    6.227 11.184
## theta = 20 10.842 18.509 6.124   12.046 15.169
```

```
store_res$mean.powerGlobalNull
```

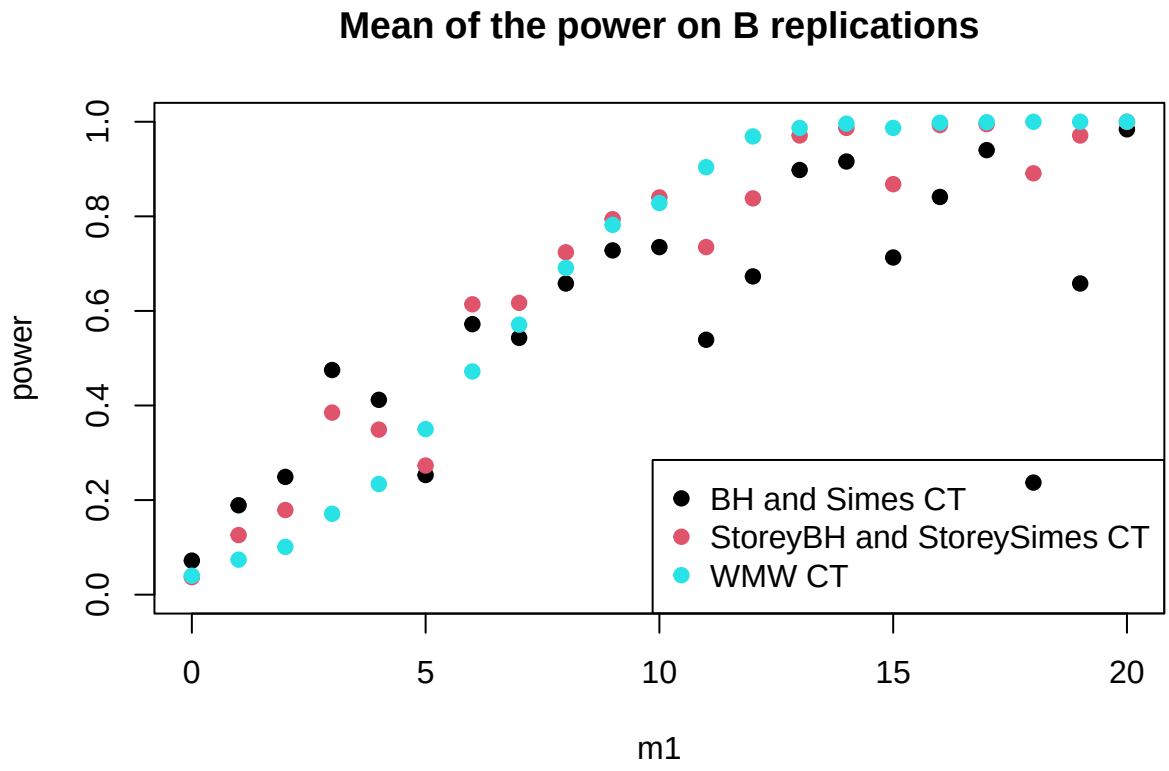
```
##               BH StoBH Simes StoSimes   WMW
## theta = 0  0.072 0.037 0.072    0.037 0.040
## theta = 1  0.189 0.126 0.189    0.126 0.074
## theta = 2  0.249 0.179 0.249    0.179 0.101
## theta = 3  0.475 0.385 0.475    0.385 0.171
## theta = 4  0.412 0.349 0.412    0.349 0.234
```

```
## theta = 5  0.253 0.273 0.253    0.273 0.350
## theta = 6  0.572 0.614 0.572    0.614 0.472
## theta = 7  0.543 0.617 0.543    0.617 0.571
## theta = 8  0.658 0.724 0.658    0.724 0.691
## theta = 9  0.728 0.794 0.728    0.794 0.782
## theta = 10 0.735 0.840 0.735    0.840 0.828
## theta = 11 0.539 0.735 0.539    0.735 0.904
## theta = 12 0.673 0.838 0.673    0.838 0.969
## theta = 13 0.898 0.971 0.898    0.971 0.987
## theta = 14 0.916 0.987 0.916    0.987 0.996
## theta = 15 0.713 0.868 0.713    0.868 0.987
## theta = 16 0.841 0.993 0.841    0.993 0.998
## theta = 17 0.940 0.995 0.940    0.995 0.999
## theta = 18 0.237 0.891 0.237    0.891 1.000
## theta = 19 0.658 0.971 0.658    0.971 1.000
## theta = 20 0.984 1.000 0.984    1.000 1.000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("bottomleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



```r
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Pen-Based Recognition of Handwritten Digits dataset

The dataset is available at http://odds.cs.stonybrook.edu/pendigits-dataset.

```r
set.seed(321)

# Initializing parameters
B=10^3
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=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)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 98 sec elapsed
```

```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("theta =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

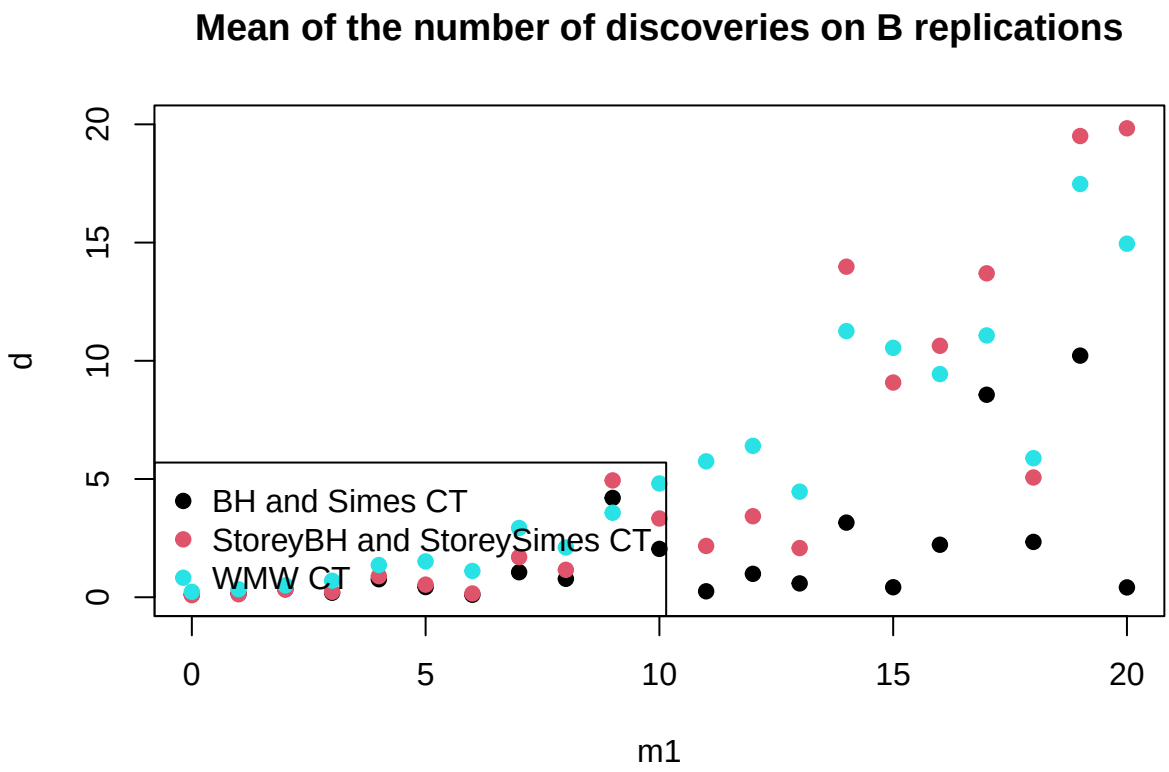
```
##                BH  StoBH Simes StoSimes    WMW
## theta = 0   0.122  0.088 0.098    0.061  0.230
## theta = 1   0.156  0.126 0.135    0.090  0.335
## theta = 2   0.344  0.322 0.269    0.197  0.502
## theta = 3   0.190  0.230 0.135    0.118  0.698
## theta = 4   0.762  0.891 0.458    0.453  1.362
## theta = 5   0.431  0.533 0.317    0.329  1.517
## theta = 6   0.104  0.158 0.086    0.100  1.110
## theta = 7   1.057  1.697 0.641    0.839  2.931
## theta = 8   0.779  1.160 0.465    0.589  2.104
## theta = 9   4.200  4.942 2.441    2.730  3.572
## theta = 10  2.044  3.330 1.311    1.718  4.812
## theta = 11  0.251  2.169 0.123    0.669  5.748
## theta = 12  0.990  3.427 0.529    1.398  6.402
## theta = 13  0.585  2.080 0.419    0.869  4.466
## theta = 14  3.156 13.980 1.195    5.756 11.256
## theta = 15  0.423  9.081 0.264    2.930 10.548
## theta = 16  2.220 10.634 1.126    4.140  9.438
## theta = 17  8.564 13.699 4.887    7.705 11.073
## theta = 18  2.340  5.065 1.465    2.555  5.881
## theta = 19 10.219 19.505 4.624   14.917 17.476
```

```
## theta = 20  0.416 19.832 0.284   12.291 14.952
```

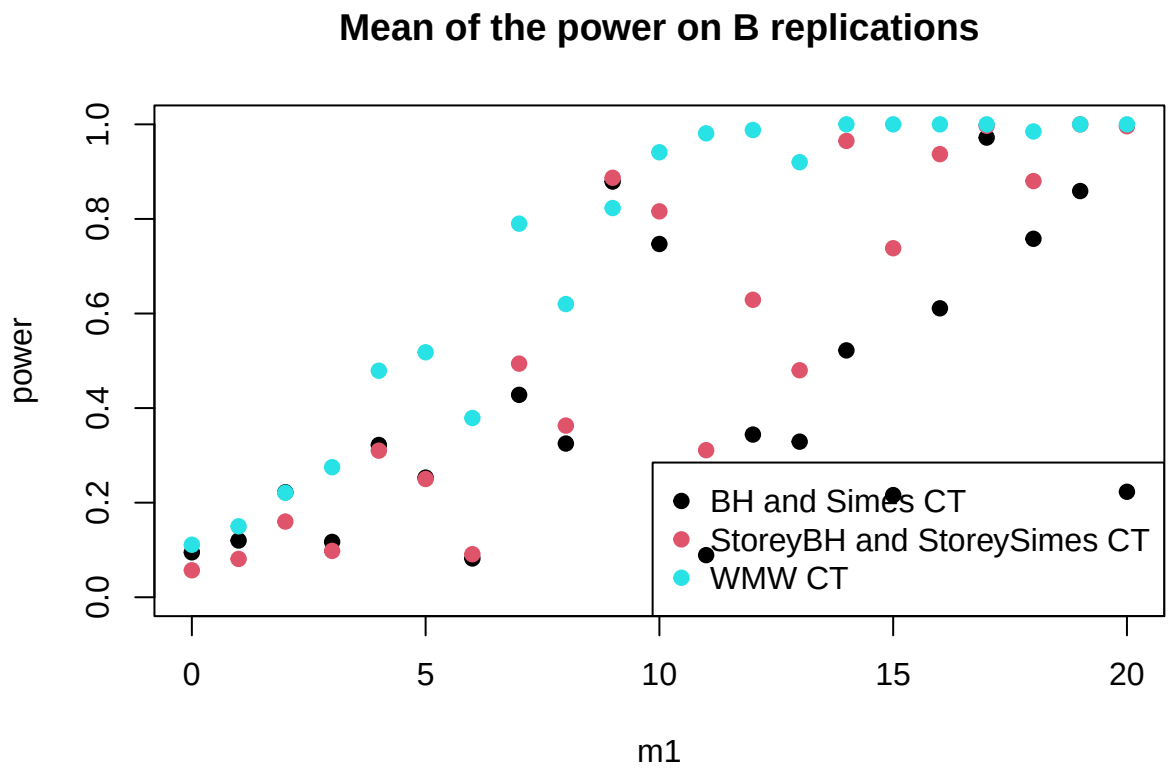
```
store_res$mean.powerGlobalNull
```

```
##               BH StoBH Simes StoSimes   WMW
## theta = 0  0.095 0.057 0.095    0.057 0.111
## theta = 1  0.120 0.081 0.120    0.081 0.150
## theta = 2  0.222 0.160 0.222    0.160 0.221
## theta = 3  0.117 0.098 0.117    0.098 0.275
## theta = 4  0.322 0.310 0.322    0.310 0.479
## theta = 5  0.253 0.250 0.253    0.250 0.518
## theta = 6  0.082 0.091 0.082    0.091 0.379
## theta = 7  0.428 0.494 0.428    0.494 0.790
## theta = 8  0.325 0.363 0.325    0.363 0.620
## theta = 9  0.879 0.887 0.879    0.887 0.823
## theta = 10 0.747 0.816 0.747    0.816 0.941
## theta = 11 0.089 0.311 0.089    0.311 0.981
## theta = 12 0.344 0.629 0.344    0.629 0.988
## theta = 13 0.329 0.480 0.329    0.480 0.920
## theta = 14 0.522 0.965 0.522    0.965 1.000
## theta = 15 0.216 0.738 0.216    0.738 1.000
## theta = 16 0.611 0.937 0.611    0.937 1.000
## theta = 17 0.972 0.997 0.972    0.997 1.000
## theta = 18 0.758 0.880 0.758    0.880 0.985
## theta = 19 0.859 1.000 0.859    1.000 1.000
## theta = 20 0.223 0.996 0.223    0.996 1.000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("bottomleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```


Mean of the power on B replications
power
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
m1
BH and Simes CT
StoreyBH and StoreySimes CT
WMW CT