QIoT_vignette

The vignette is intended to show how to use this package to do simple inference with some examples. The code also serves as an approach to replicate the results in our paper.

Install QIoT package The installation of our QIoT package requires the help from package devtools. You can follow the code below.

```
library(gurobi)
library(devtools)
install_github("Yongchang-Su/QIoT",force = TRUE)
#> Rcpp          (1.0.9         -> 1.0.10       ) [CRAN]
#> RcppArmad...  (0.11.2.0.0    -> 0.12.0.1.0   ) [CRAN]
#> lpSolveAPI    (5.5.2.0-17.7 -> 5.5.2.0-17.9) [CRAN]
#> package 'Rcpp' successfully unpacked and MD5 sums checked
#> package 'RcppArmadillo' successfully unpacked and MD5 sums checked
#> package 'lpSolveAPI' successfully unpacked and MD5 sums checked
#>
#> The downloaded binary packages are in
#>  C:\Users\ycsuf\AppData\Local\Temp\RtmpeEWPtz\downloaded_packages
#> Warning message:
#> In normalizePath(path.expand(path), winslash, mustWork) :
#>   path[1]="C:/Users/ycsuf/OneDrive/??": The filename, directory name, or volume label syntax is inco
#> Warning message:
#> In normalizePath(path.expand(path), winslash, mustWork) :
#>   path[1]="C:/Users/ycsuf/OneDrive/??": The filename, directory name, or volume label syntax is inco
#> Warning message:
#> In normalizePath(path.expand(path), winslash, mustWork) :
#>   path[1]="C:/Users/ycsuf/OneDrive/??": The filename, directory name, or volume label syntax is inco
#> * checking for file 'C:\Users\ycsuf\AppData\Local\Temp\RtmpeEWPtz\remotes911035eefbf\Yongchang-Su-QI
#> * preparing 'QIoT':
#> * checking DESCRIPTION meta-information ... OK
#> * cleaning src
#> * installing the package to process help pages
#> * saving partial Rd database
#> * cleaning src
#> * checking for LF line-endings in source and make files and shell scripts
#> * checking for empty or unneeded directories
#>   NB: this package now depends on R (>= 3.5.0)
#>   WARNING: Added dependency on R >= 3.5.0 because serialized objects in
#>   serialize/load version 3 cannot be read in older versions of R.
#>   File(s) containing such objects:
#>     'QIoT/data/cadmium.rda'
#> * building 'QIoT_1.0.tar.gz'
#>
library(QIoT)
```

Load the data and specify ranking method The first thing to do is to load the data and specify the ranking method your want to apply. Here we simply use Wilcoxon rank sum statistics for all strata.

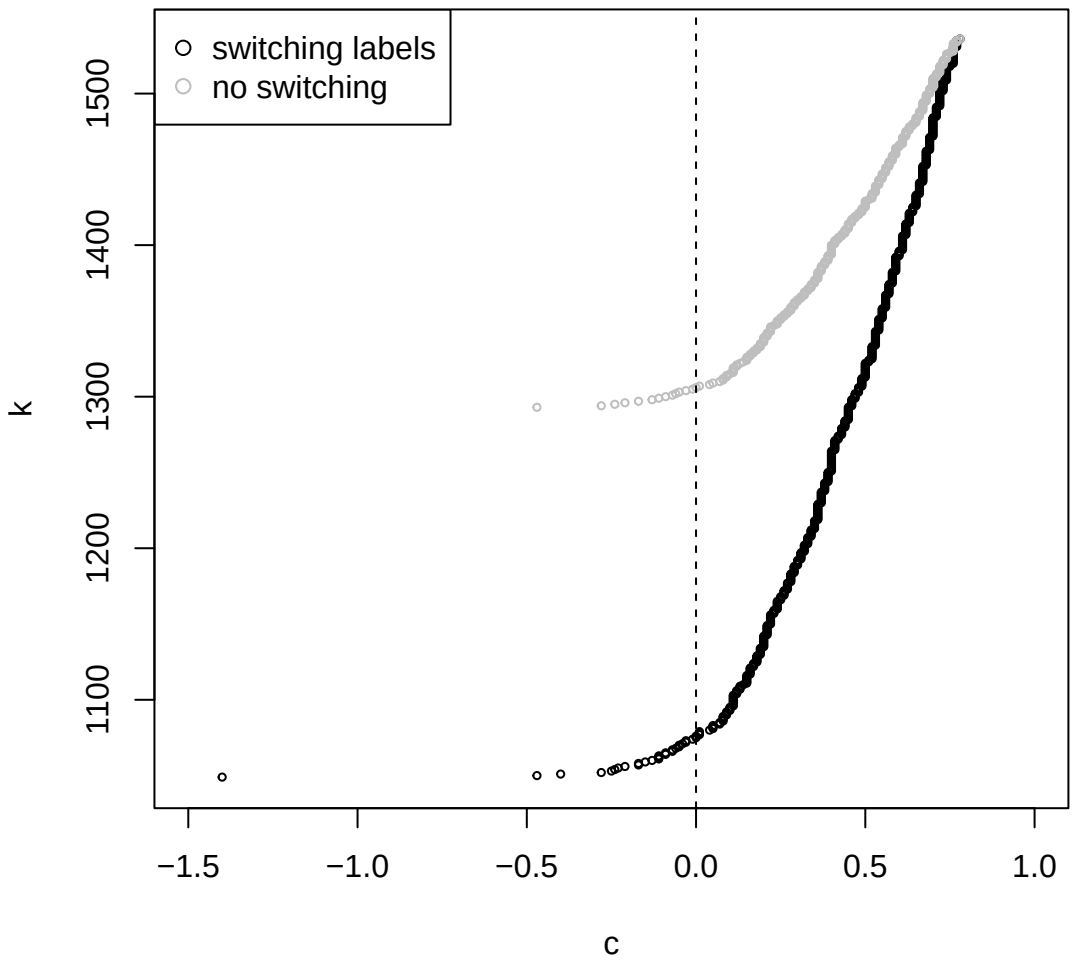
```
data("cadmium")
Y = cadmium$cadmium
block = as.factor(cadmium$mset)
Z = cadmium$z
method.list.all = list()
method.list.all[[1]] = list(name = "Wilcoxon")
```

Calculate lower limits of CIs with or without applying the switching treatment trick One important contribution of our paper is that we can obtain simultaneous confidence region for quantiles of treatment effects. Here is the code to implement them. We also encourage the users to apply our switching treatment trick, which can make the confidence region more informative.

```
LB_switch = ci_quantile_scre(Z,Y,block,method.list.all=method.list.all,
                             opt.method = "Greedy", switch = TRUE, null.max=10^5)$LB

LB_noswitch = ci_quantile_scre(Z,Y,block,method.list.all=method.list.all,
                             opt.method = "Greedy", switch = FALSE, null.max=10^5)$LB
```

Visual aids for CIs Below is the visual results of confidence region for quantiles of treatment effect. The points are the lower limits of one-sided confidence intervals. For this specific case, we are 90% confident that all treatment effects fall to the right side of these points. Based on the plot, it's obvious to conclude that switching labels does promote information utilization, as it gives tighter confidence region.

```
plot(LB_switch[LB_switch>-Inf], (1:length(LB_switch))[LB_switch>-Inf], type = "p",
     xlim = c(-1.5, 1), xlab = "c", ylab = "k", cex = 0.5)
points(LB_noswitch[LB_noswitch>-Inf], (1:length(LB_noswitch))[LB_noswitch>-Inf]
       , cex = 0.5, col = "grey")

abline(v = 0, lty = 2)
legend("topleft", pch = 1, col = c("black", "grey"),
       legend = c("switching labels", "no switching"))
```



Find desired Γ's for sensitivity analysis In sensitivity analysis, we gradually release the model constraints until major conclusions are changed. You can think of Γ as a parameter that controls how strict the constraint is. We are interested in the values of Γ that makes confidence interval of a certain quantile change from not including 0 to include it.

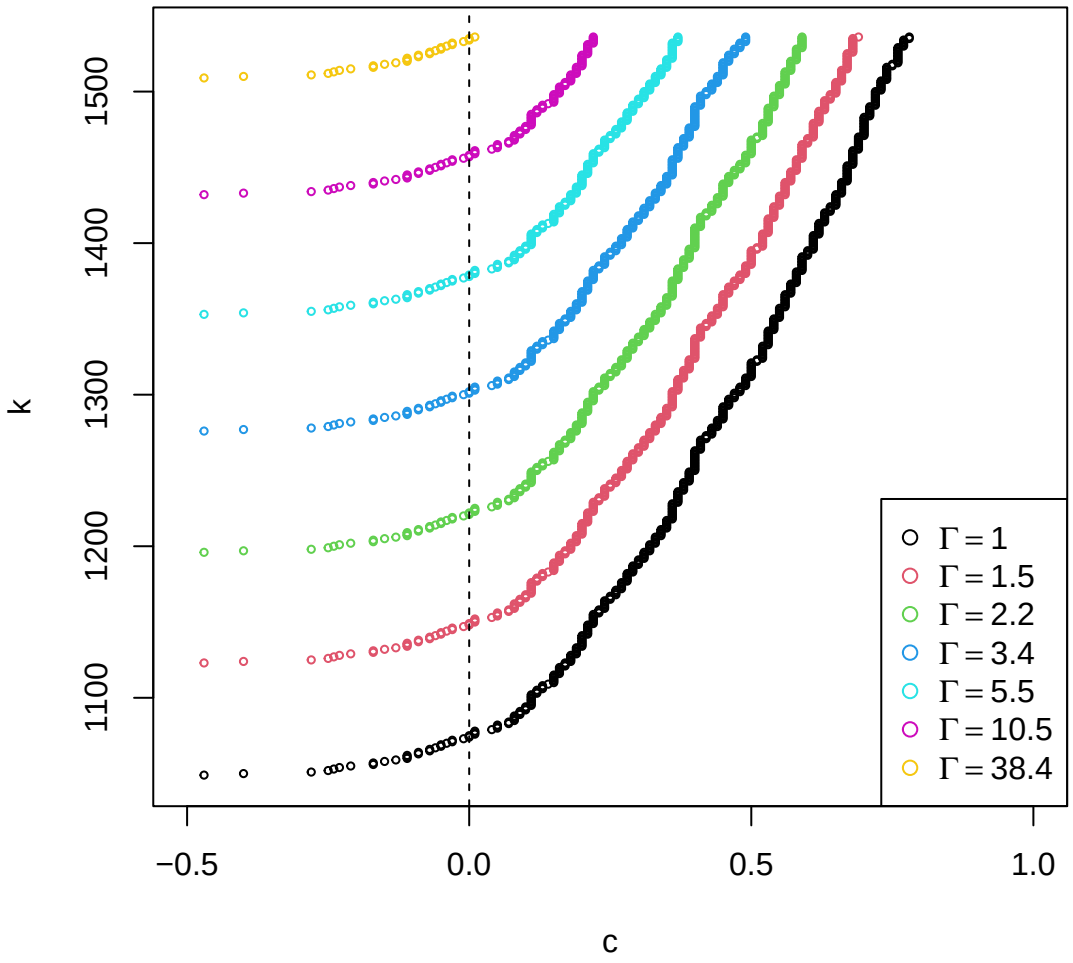
Here we first figure out the corresponding Γ's to 70%, 75%, ..., 100%.

```
gammas = rep(0, 7)
enum = 1
gam = 1

#### Find corresponding gammas
for(quant in 0.65 + 0.05*(1:7)){
  k = floor(quant * length(Z))
  while(1){
    pval = pval_quantile_sen(Z, Y, block, k, 0, gam=gam,
                method.list.all=method.list.all, ties = "upper", switch = TRUE)$upper
```

```
    if(pval > 0.1){

      gammas[enum] = max(gam - 0.1, 1)
      enum = enum + 1
      break
    }
    gam = gam + 0.1
  }
}
print(gammas)
#> [1]  1.0  1.5  2.2  3.4  5.5 10.5 38.4
```

```
CIlist = list()
for(i in 1:7){
  gam = gammas[i]
  CIlist[[i]] = ci_quantile_sen(Z, Y, block, gam=gam,
                          method.list.all=method.list.all,switch = TRUE)$LB
}
```

```
plot(CIlist[[1]][CIlist[[1]]>-Inf], (1:length(CIlist[[1]]))[CIlist[[1]]>-Inf],
     type = "p", xlim = c(-0.5, 1), xlab = "c", ylab = "k", cex = 0.5)
for(i in 2:7){
  points(CIlist[[i]][CIlist[[i]]>-Inf], (1:length(CIlist[[i]]))[CIlist[[i]]>-Inf],
         cex = 0.5, col = i)
}

abline(v = 0, lty = 2)

legend("bottomright", col = 1:7, pch = 1, legend = c(expression(Gamma == 1),
                                                     expression(Gamma == 1.5),
                                                     expression(Gamma == 2.2),
                                                     expression(Gamma == 3.4),
                                                     expression(Gamma == 5.5),
                                                     expression(Gamma == 10.5),
                                                     expression(Gamma == 38.4)))
```



Calculate CIs for those Γ's

Comparison of power between switching labels or not We generate the data as following. Set number of strata $S = 200$, stratum size $n = 10$. Within each stratum, there is 1 treatment unit and 9 control units. The response is generated by

$$Y = Z + \mathcal{N}(0, 1),$$

i.e. the treatment effect is constant 1 in this setting. We want to calculate the empirical power of our method with or without applying switching labels trick.

```
S = 200
n = 10
N = S*n
Z = rep(c(1, rep(0,n-1)), S)
block = rep(1:S, each = n)

method.list.all = list()
method.list.all[[1]] = list(name = "Stephenson", s=4)
```

```
pval_quantile_sen(Z, Y, block, k, 0, gam=gam,
                  method.list.all=method.list.all, switch = TRUE)$upper
#> [1] 1

sw = rep(0,10)
no_sw = rep(0,10)
rep_time = 100
for(i in 1:rep_time){
  Y = rnorm(S*n) + Z
  for(j in 1:10){
    if(pval_quantile_sen(Z,Y,block,floor(0.95*N),0,gam=1+0.1*(j-1),
                         method.list.all=method.list.all)$lower<0.1){
      no_sw[j] = no_sw[j] + 1
    }
    if(pval_quantile_sen(Z,Y,block,floor(0.95*N),0, gam=1+0.1*(j-1),
                         method.list.all=method.list.all,switch = TRUE)$lower<0.1){
      sw[j] = sw[j] + 1
    }
  }
}


power = rbind(sw, no_sw)/rep_time
colnames(power) = 1+(0:9)*0.1
print(power)
#>          1  1.1  1.2  1.3  1.4 1.5  1.6  1.7 1.8 1.9
#> sw    0.96 0.88 0.69 0.42 0.19 0.1 0.08 0.02   0   0
#> no_sw 0.00 0.00 0.00 0.00 0.00 0.0 0.00 0.00   0   0
```