Supplementary material: The power of visualizing distributional differences: Formal graphical n-sample tests.

Konstantinos Konstantinou

2024-08-14

This supplementary material contains supporting material for the article "The power of visualizing distributional differences: Formal graphical n-sample tests" by Konstantinos Konstantinou, Tomáš Mrkvička, and Mari Myllymäki. The following material presents a template of the methodology followed in the simulation study to investigate the power of the proposed tests.

Installing GET

To install the GET from CRAN, start R and type

```
install.packages("GET")
```

Data simulation function

The code below implements a function that simulates data for each Experiment in the simulation study (DO NOT MODIFY).

The following options should be specified when simulating new data:

1. Experiment_ID: This variable defines the ID of the simulation experiment. Valid values are integers from 1-5 (corresponding to Experiments I-V in the paper).
2. N: The sample size for each sample. The total number of observations is then 2*N.
3. df: The number of degrees of freedom for the t-distribution in Experiment 1.

```
Sim_data <- function( Experiment = 1, N = 100,df =2){
  if( !Experiment %in% c(1:5)) stop("Please provide a valid Experiment ID.")
  #Simulate first sample
  if( Experiment == 5){ sample1 = rnorm(N, 8, 3.34) }
  else{ sample1 = rnorm(N) }
  #Simulate second sample
  r = seq(-3,3,len=100)
  switch(Experiment,
         #Experiment 1
         { sample2 = rt(N, df)
           switch (df,
             {r = seq(-7,7,len=100)},
```

```
            {r = seq(-5,5,len=100)},
            {r = seq(-4,4,len=100)})
        },
        #Experiment 2
        { sample2 = rnorm(N, 0.3, 1)},
        #Experiment 3
        { sample2 = rnorm(N, 0, 1.3)},
        #Experiment 4
        { Z = rbinom(N, 1, 0.5)
          sample2 = numeric(N)
          for( i in 1:N){
          if( Z[i] == 1){ sample2[i] = rnorm(1,1,0.5)}
          else{ sample2[i] = rnorm(1,-1,0.5)}},
        #Experiment 5
        { sample2 = rlnorm(N, 2, 0.4)
          r = seq(0,15,len=100)}
        )
data.frame(sample1 = sample1,
           sample2 = sample2,
           r = r)
}
```

Simulation study

Due to the computational costs, the simulation study was run on a cluster of CSC – IT Center for Science. The code below displays a "toy example" of how the simulation study was carried out. The following simulation options should be specified:

1. nsim_end: Number of data resamples
2. nsim: Number of permutations for the global test
3. N : The sample sizes of the data resamples (for 2 groups the total amount of datapoints is 2*N)
4. Experiment_ID: This variable defines the ID of the simulation experiment
5. stat: The test statistic considered. Valid values are any subset of the set c("qq","qr","diff","ecdf","den", "shift"). The "C2" test corresponds to the option c("qq","den") and the "C4" test to the option c("qq","diff","ecdf","den").

```
library(GET) # Loads the GET package
library(ggplot2) # For plotting
set.seed(0)
############################ Simulation options ##################
#Script was run with different simulation options.

nsim_end <- 10            # number of resamples
# NOTICE: Increasing nsim_end will increase computational time
# nsim_end = 1000 was used in the simulation study.

nsim <- 5000        # number of permutations
N <- 50             # Sample size
Experiment_ID <- 4   # Experiment ID
df = 2     # Degrees of freedom for the t distribution in Experiment 1.
stat <- c("qq","den")    # Test statistic used for testing
```

```
##########################################

############################ Simulation study(DO NOT MODIFY) ##################
res <- NULL
for( i in 1: nsim_end){
Simulated_data <- Sim_data(Experiment = Experiment_ID,
                           N = N,
                           df = df)
res <- c(res,
        list(
         GET.distrequal(
           x = list ( first_sample = Simulated_data$sample1,
                      second_sample = Simulated_data$sample2),
           stat = stat,
           nsim = nsim,
           r = Simulated_data$r)
           )
        )
}
```

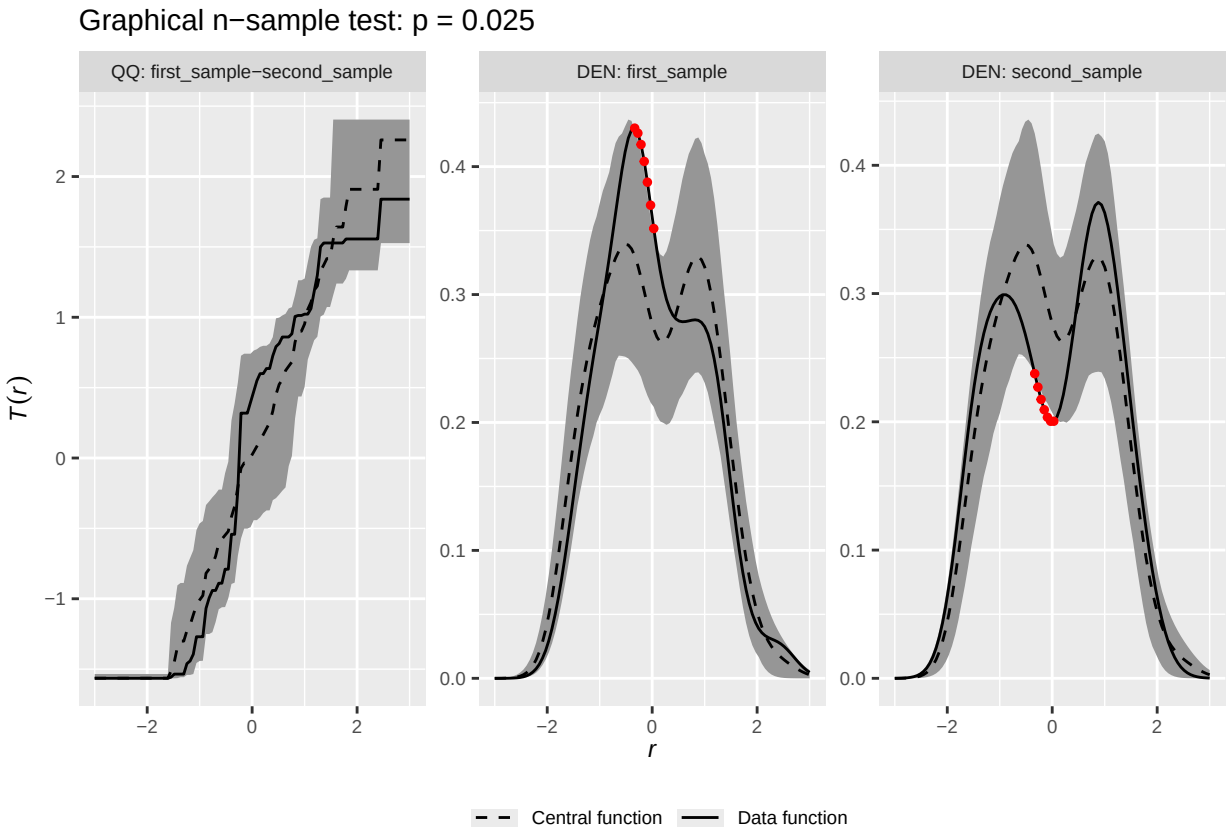
Analysing the results

Plotting the global envelope test

The code chunk above produced a list of global_envelope objects called "res".

- To plot the test results (for each resample) one can use the plot method (for options in plotting see ?plot.global_envelope)

```
# visualizing the global envelope for the first resample
plot(res[[1]], scales="free") +
  theme(text = element_text(size = 9))
```



Calculating Power

- The p-values of the tests can be accessed through the "p" attribute of the global_envelope objects.
- The Power of the test is calculated as the average number of times the p-value is less than 0.05.

```
# Calculate the p-values for each resample
pvalues <- sapply(res, function(x) attr(x,"p"))
pvalues
```

```
##  [1] 0.02499500 0.13357329 0.00139972 0.00859828 0.00339932 0.00079984
##  [7] 0.00319936 0.56028794 0.00399920 0.01679664
```

```
# Calculate Power/Type I error
Power <- mean( pvalues < 0.05)
print(paste(" Power =", Power))
```

```
## [1] " Power = 0.8"
```