Permutational tests and bootstrap reverse percentile confidence intervals on prevalence

2023-07-05

TWO POPULATION PAIRED TEST H0: mean(male) = mean(female) H1: mean(male) ≠ mean(female) can be reformulated as H0: mean(male-female) = mu H1: mean(male-female) ≠ mu with mu = c(0)

Equivalent to the center of simmetry of one univariate population. The bootstrap is instead computed on the third quantile of the distribution.

I create the dataset of the differences for each year

Utils

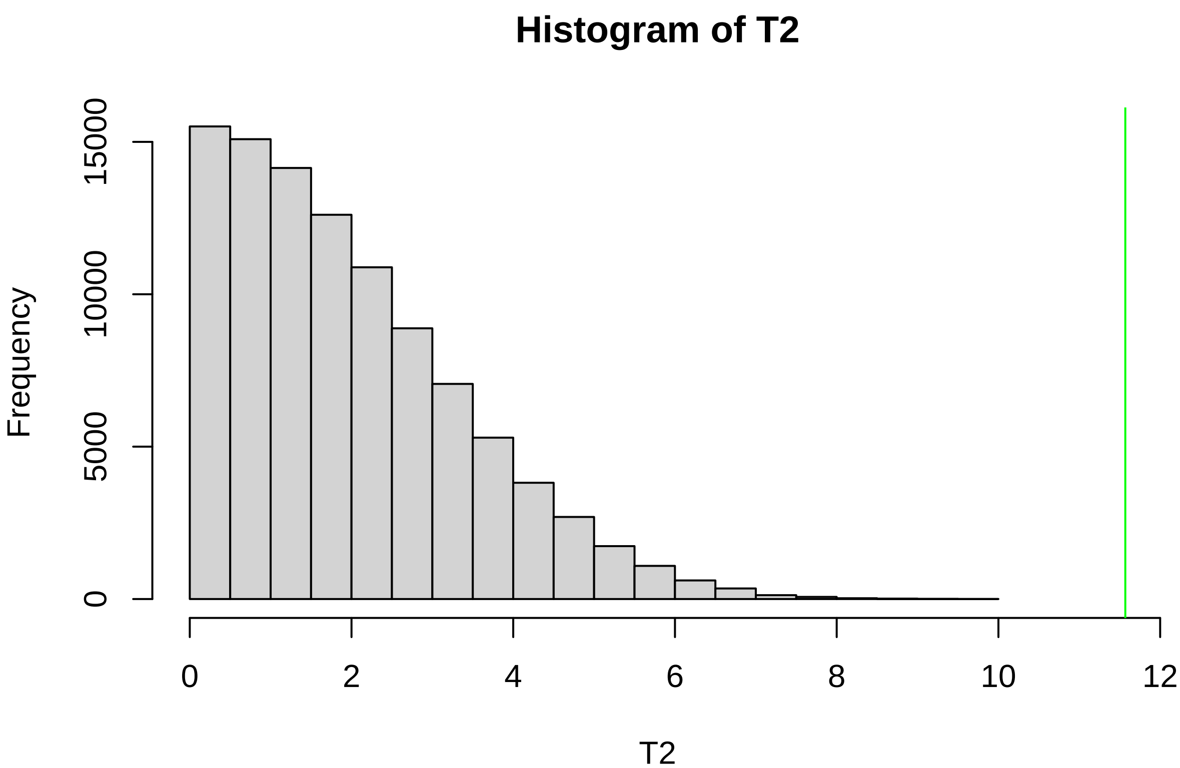
```
perm_t_test_paired <- function(diff, mu0, iter = 10000, test_statistic = mean){
  T20 <- abs(test_statistic(diff) - mu0)
  n <- length(diff)
  T2 <- numeric(iter)
  for(perm in 1:iter)
  {
    # Random permutation
    # obs: exchanging data within couples means changing the sign of the difference
    signs.perm <- rbinom(n, 1, 0.5)*2 - 1
    diff_perm <- (diff) * matrix(signs.perm,nrow=n ,ncol=1,byrow=FALSE)
    diff.mean_perm <- test_statistic(diff_perm)
    T2[perm] <- abs(diff.mean_perm-mu0)
  }
  hist(T2, xlim=c(0,T20))
  abline(v=T20, col='green')

  p.value <- sum(T2 >= T20) / B
  return(p.value)
}



###Compute bootstrap confint on the differences
diagnostic_bootstrap <- function(distro, obs, alpha) {
  variance_pred <- var(distro)
  print(paste("Variance: ", variance_pred))
  sd_pred <- sd(distro)
  print(paste("Standard deviation: ", sd_pred))
  bias_pred <- mean(distro) - obs
  print(paste("Bias: ", bias_pred))
  MSE_pred <- variance_pred + bias_pred^2
  print(paste("MSE: ", MSE_pred))
  right.quantile <- quantile(distro, 1 - alpha / 2)
  left.quantile <- quantile(distro, alpha / 2)
```

```
  CI <- c(
    obs - (right.quantile - obs),
    obs,
    obs - (left.quantile - obs)
  )
  names(CI) <- c("lower", "center", "upper")
  print(CI)
  plot(ecdf(distro), main = "Bootstrap distribution")
  abline(v = CI[2], col = 3, lwd = 2)
  abline(v = CI[c(1, 3)], lty = 3)
}
```

For each year, I run a permutational test and compute bootstrap interval

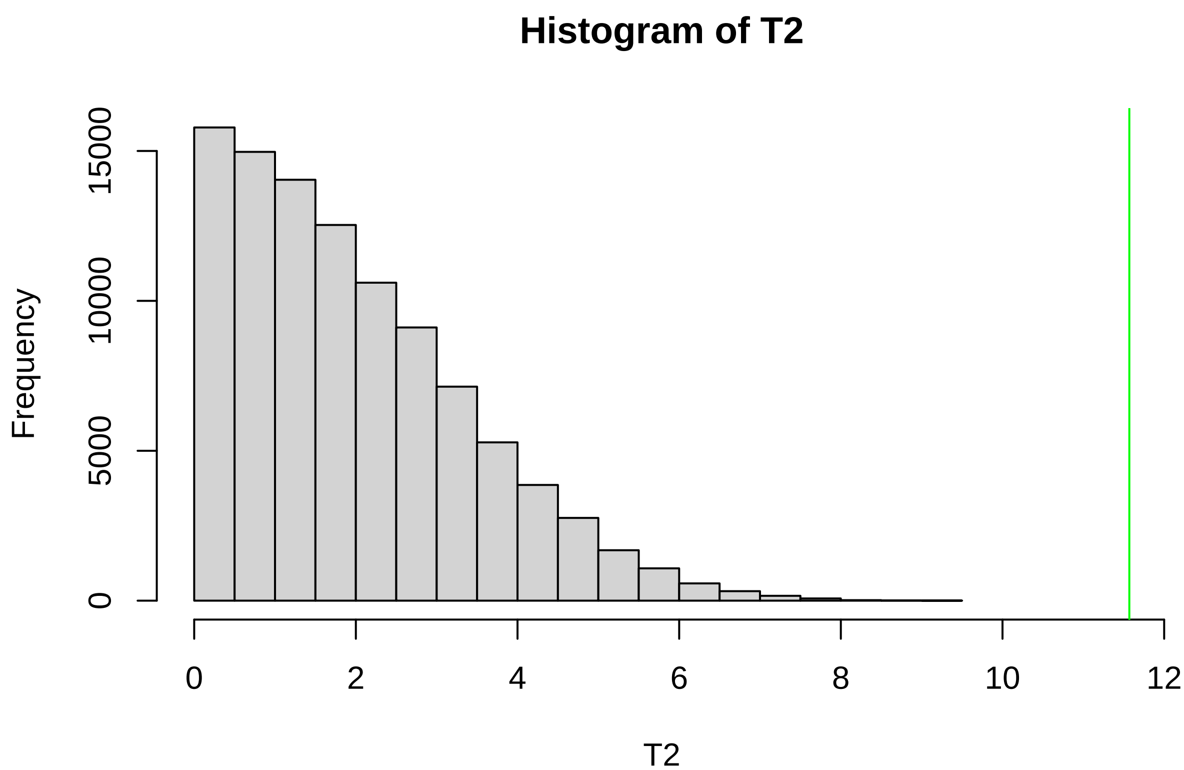
2007

```
mydf<-t(df_2007)
perm_t_test_paired(mydf, mu0, B, mean)
```




```
## [1] 0
```

```
p_val_2007<-perm_t_test_paired(mydf, mu0, B, mean)
```



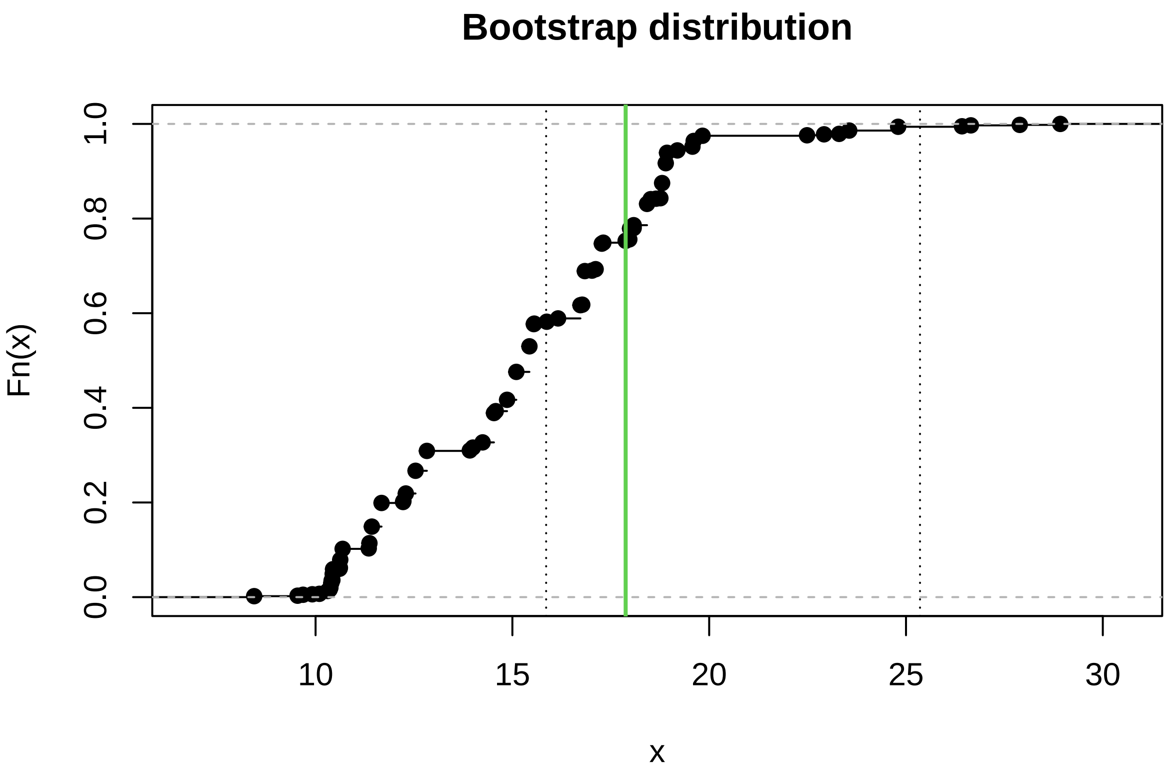

```
p_val_2007
```

```
## [1] 0
```

```
compute_bootstrap_sample <- function(df) {
  permutation <- sample(1:length(df), replace = T) #or length(df) if numeric vector
  df.boot <- df[permutation]
  quant0.75 <- quantile(df.boot)[4]
  return(quant0.75) # this can be 2D etc.
}

n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))
clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
B=1000
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

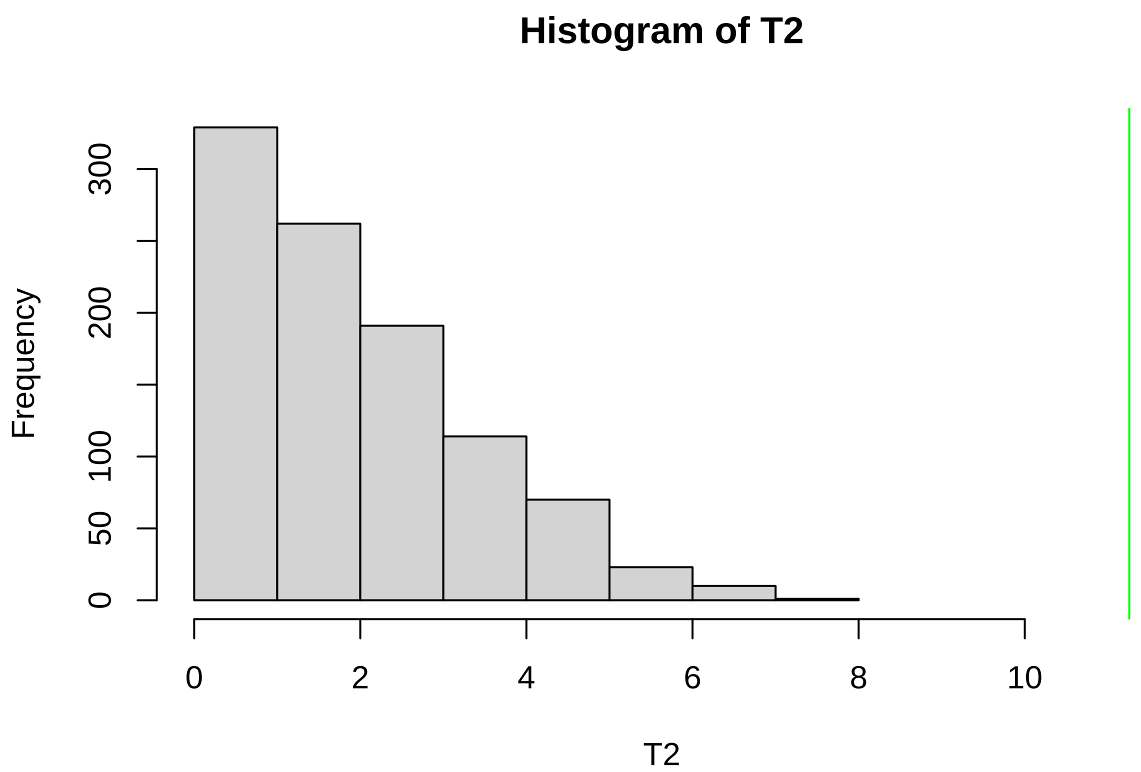
```
## [1] "Variance:  10.2082147438479"
## [1] "Standard deviation:  3.19502969373492"
## [1] "Bias:  -2.516836"
```

```
## [1] "MSE:  16.5426781947439"
##    lower   center    upper
## 15.85765 17.87800 25.35600
```



2008

```
mydf<-t(df_2008)
p_val_2008<-perm_t_test_paired(mydf, mu0, B, mean)
```

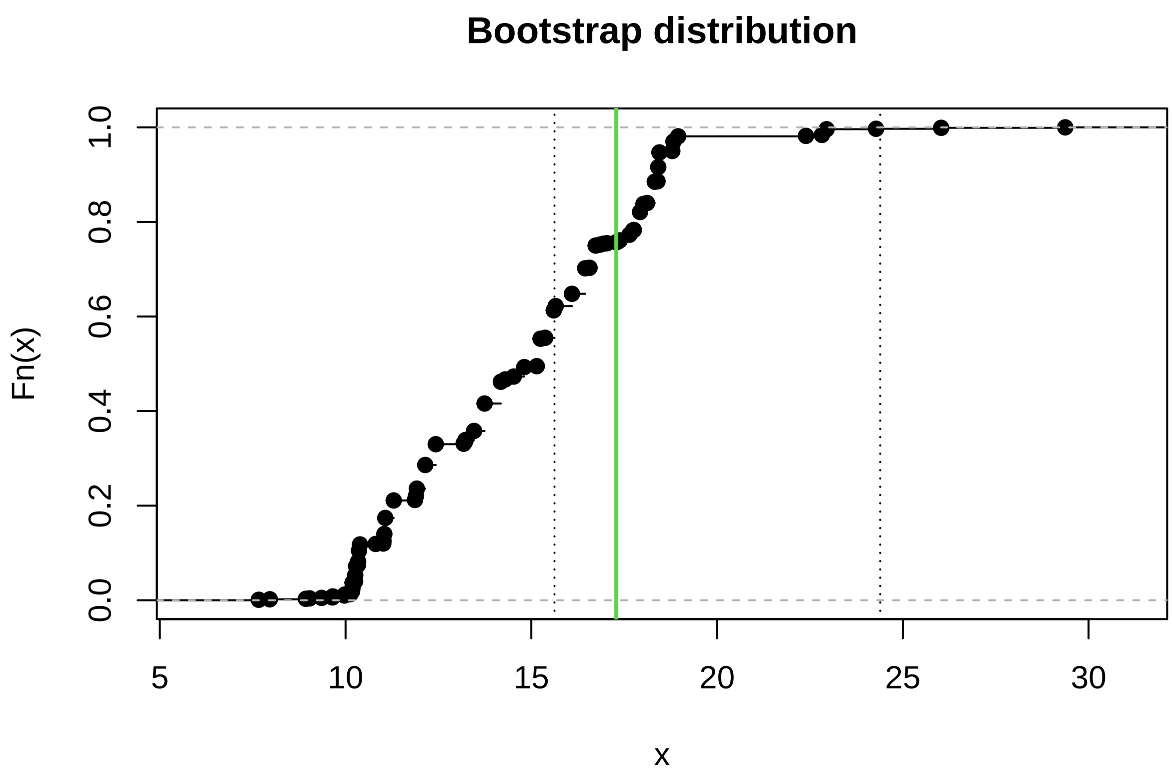



```
p_val_2008
```

```
## [1] 0
```

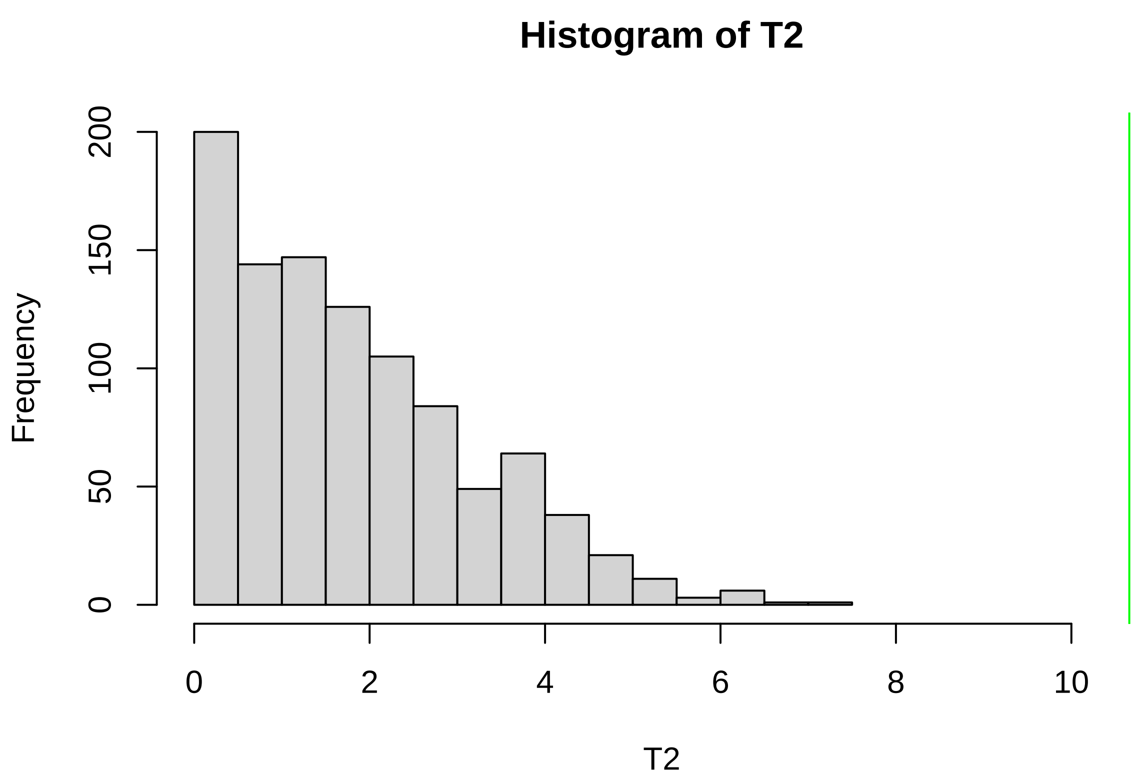
```
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))
clusterExport(cl, varlist = list("mydf",  #or "df"
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

myalpha=0.05 #change!
T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  9.43542421331329"
## [1] "Standard deviation:  3.07171356303176"
## [1] "Bias:  -2.55743"
## [1] "MSE:  15.9758724182133"
##  lower center  upper
## 15.624 17.288 24.391
```



2010

```
mydf<-t(df_2010)
p_val_2010<-perm_t_test_paired(mydf, mu0, B, mean)
```

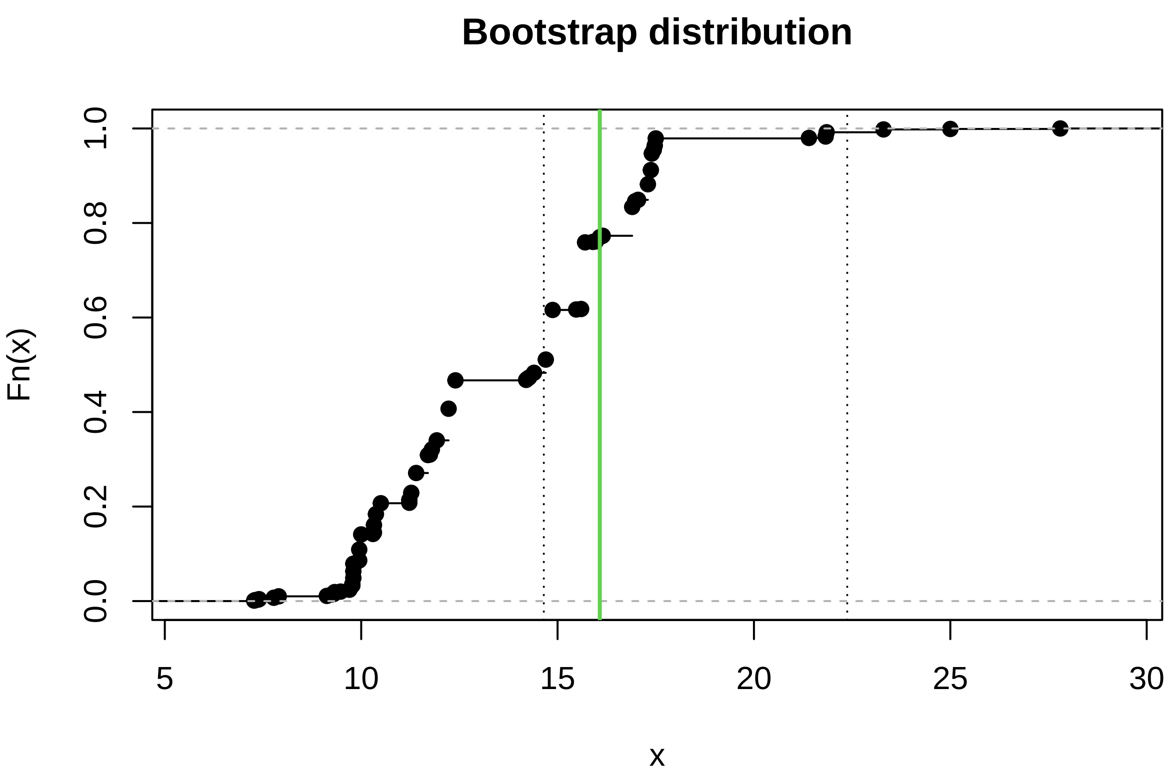


```r
p_val_2010
```

```
## [1] 0
```

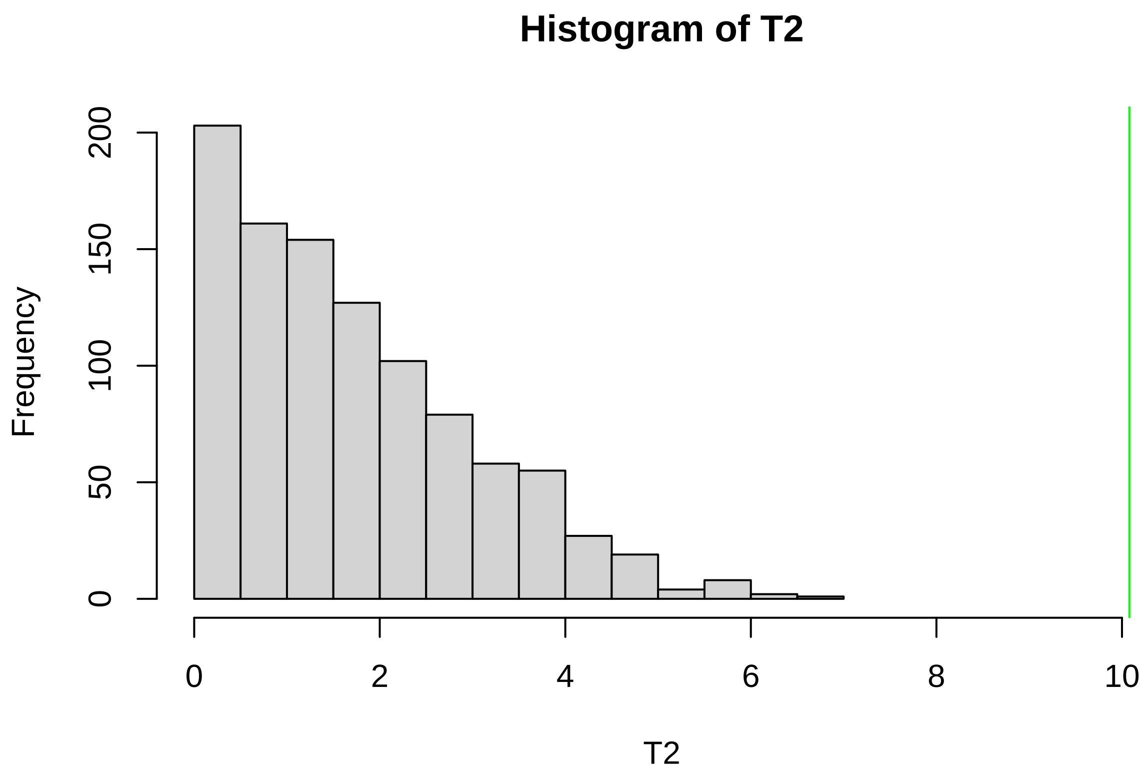
```r
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))
clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  9.00701870870871"
## [1] "Standard deviation:  3.0011695568076"
## [1] "Bias:  -2.2259"
## [1] "MSE:  13.9616495187087"
##  lower center  upper
## 14.650 16.075 22.375
```



2012

```
mydf<-t(df_2012)
p_val_2012<-perm_t_test_paired(mydf, mu0, B, mean)
```

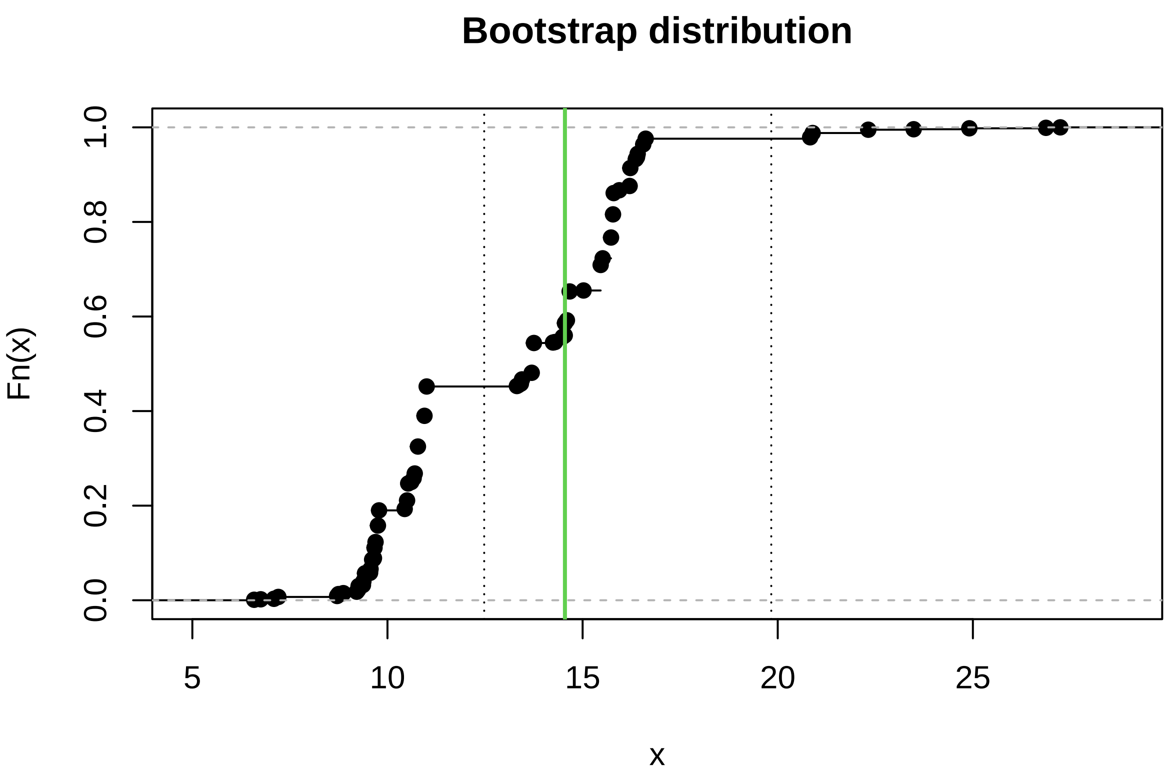



```
p_val_2012
```

```
## [1] 0
```

```
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))

clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  8.8892734034675"
## [1] "Standard deviation:  2.98148845435757"
## [1] "Bias:  -1.388456"
## [1] "MSE:  10.8170834674035"
##  lower center  upper
## 12.478 14.547 19.834
```



2014

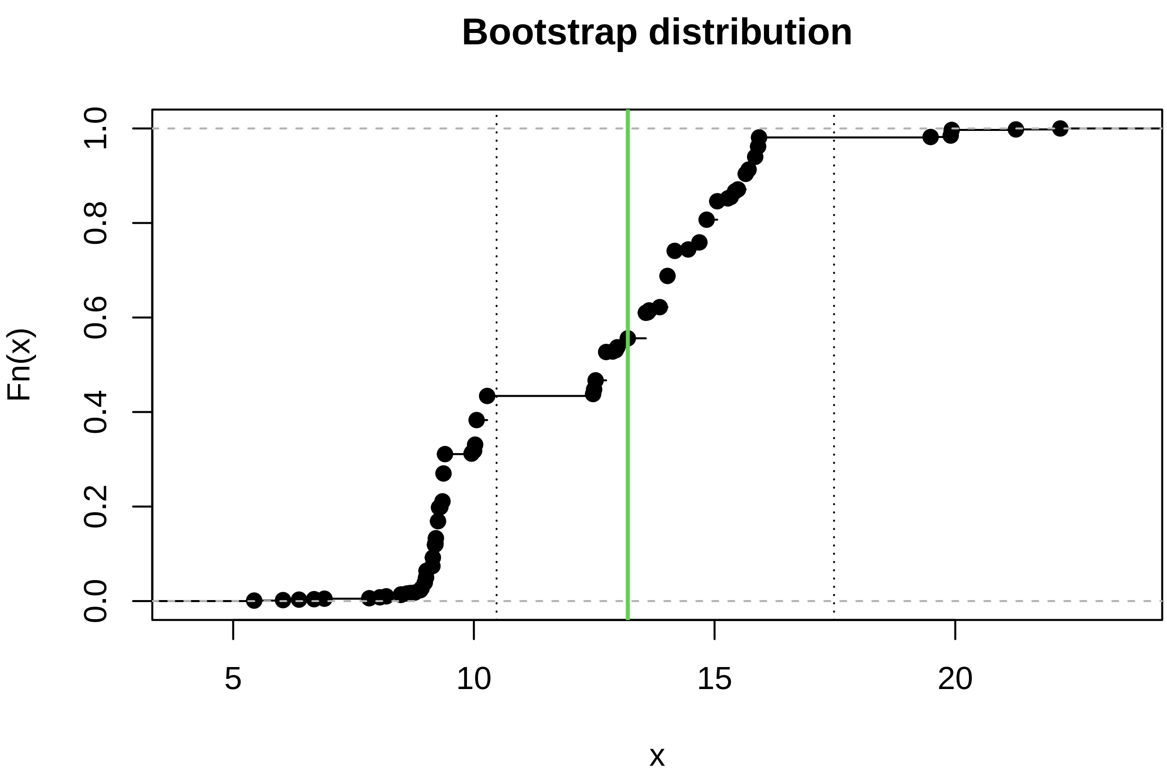
```r
mydf<-t(df_2014)
p_val_2014<-perm_t_test_paired(mydf, mu0, B, mean)
```

```
p_val_2014
```

```
## [1] 0
```

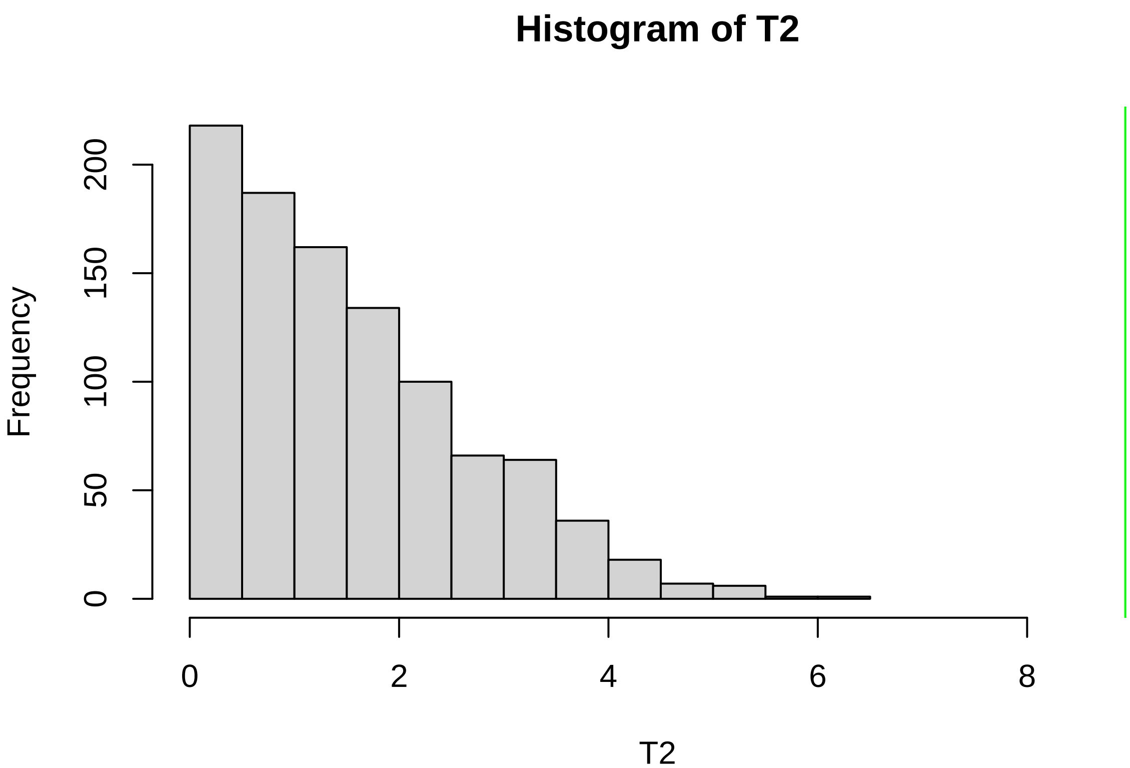
```
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))
clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  7.72846005712816"
## [1] "Standard deviation:  2.78001080162077"
## [1] "Bias:  -0.899923000000003"
## [1] "MSE:  8.53832146305717"
##  lower center  upper
## 10.472 13.198 17.482
```



2016

```r
mydf<-t(df_2016)
p_val_2016<-perm_t_test_paired(mydf, mu0, B, mean)
```

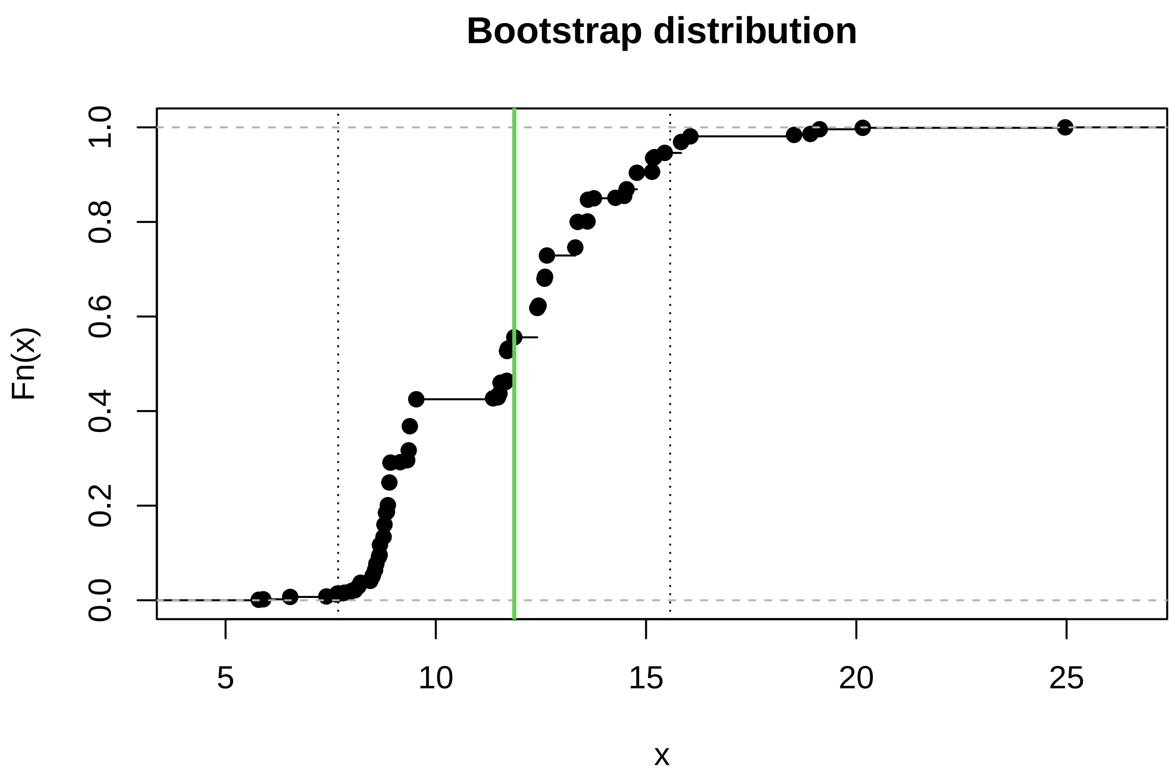



```r
p_val_2016
```

```
## [1] 0
```

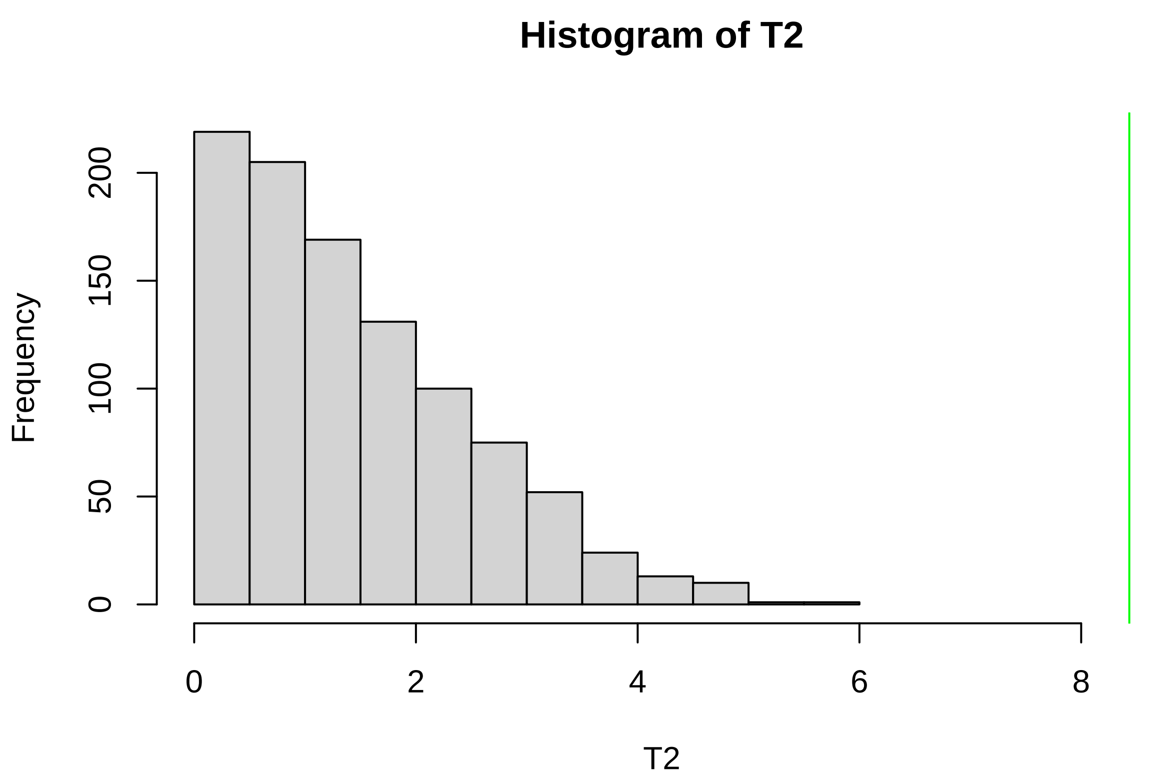
```r
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))
clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  6.74721197844719"
## [1] "Standard deviation:  2.59753960093916"
## [1] "Bias:  -0.39965055555556"
## [1] "MSE:  6.90693254500305"
##     lower    center     upper
##  7.677778 11.865556 15.573333
```



2018

```
mydf<-t(df_2018)
p_val_2018<-perm_t_test_paired(mydf, mu0, B, mean)
```

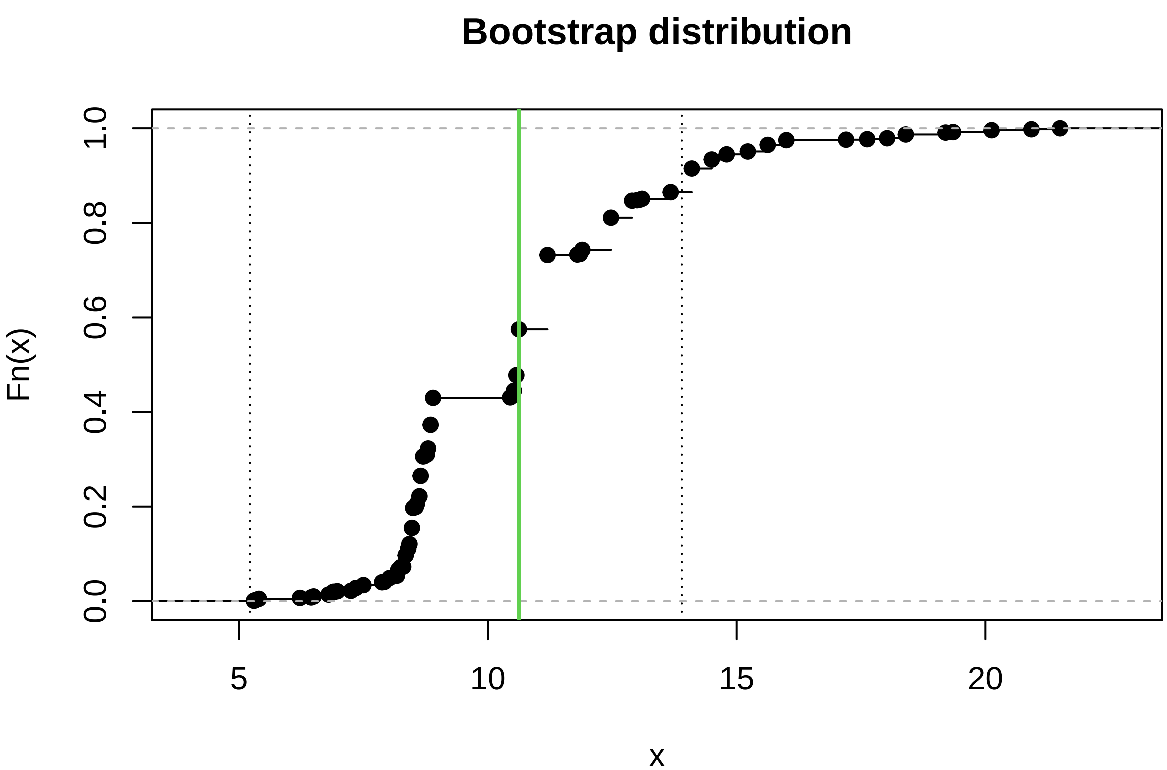


```
p_val_2018
```

```
## [1] 0
```

```
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))
clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  6.51274827577573"
## [1] "Standard deviation:  2.55200867470621"
## [1] "Bias:  0.0883500000000019"
## [1] "MSE:  6.52055399827573"
##  lower center  upper
##  5.220 10.625 13.900
```



2020

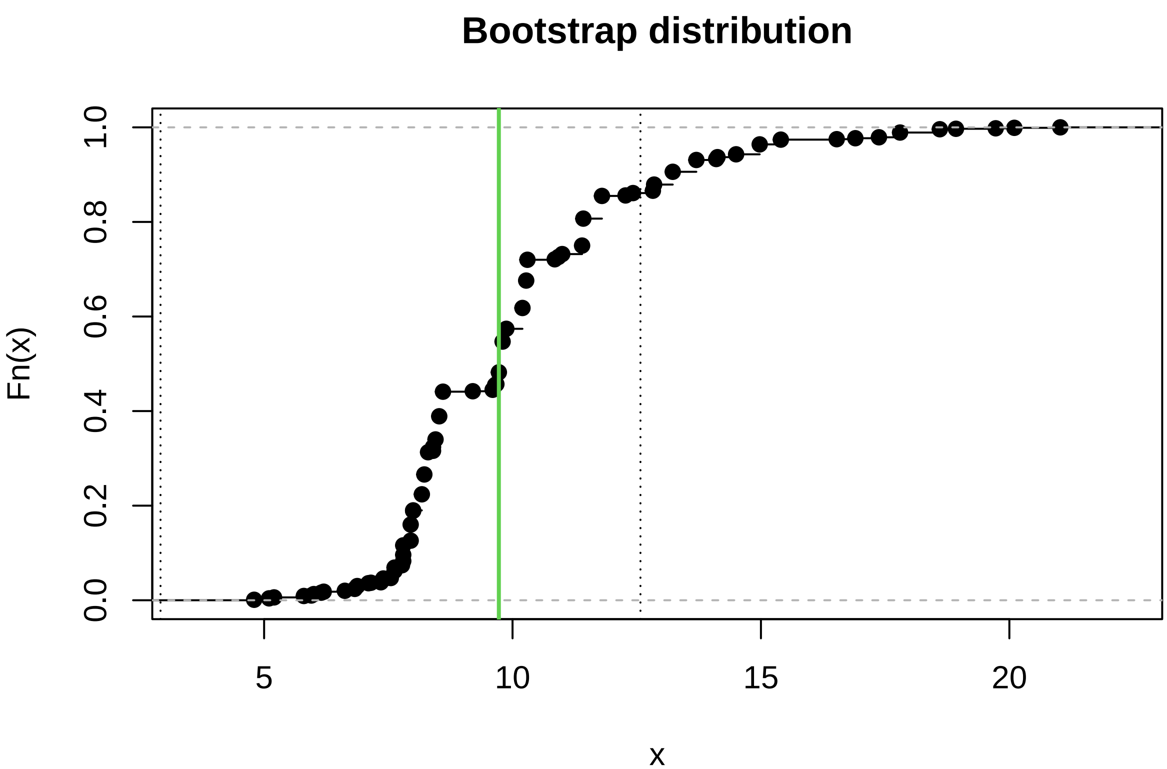
```
mydf<-t(df_2020)
p_val_2020<-perm_t_test_paired(mydf, mu0, B, mean)
```


```
p_val_2020
```

```
## [1] 0
```

```
n_cores <- detectCores()
cl <- makeCluster(n_cores)
invisible(clusterEvalQ(cl, library(np)))

clusterExport(cl, varlist = list("mydf",
                                 "compute_bootstrap_sample",
                                 "seed"))
set.seed(seed)
T.boot <- pbreplicate(B, compute_bootstrap_sample(mydf), cl = cl)
stopCluster(cl)

T.obs=compute_bootstrap_sample(mydf)
diagnostic_bootstrap(T.boot, T.obs, alpha = myalpha)
```

```
## [1] "Variance:  5.84607877314816"
## [1] "Standard deviation:  2.41786657471999"
## [1] "Bias:  0.284575"
## [1] "MSE:  5.92706170377316"
##     lower    center     upper
##  2.915625  9.725000 12.575000
```




```r
#P-value
c(p_val_2007,
  p_val_2008,
  p_val_2010,
  p_val_2012,
  p_val_2014,
  p_val_2016,
  p_val_2018,
  p_val_2020)
```

```
## [1] 0 0 0 0 0 0 0 0
```

Bootstrap

```r
# Year lower center  upper
# 2010 14.650 16.075 22.375
# 2012 8.421575 14.547000 19.846900
# 2014 10.4915 13.1980 17.4820 ---->2014
# 2016 7.677778 11.865556 15.522222
# 2018 3.225 10.625 13.750
# 2020  4.050  9.725 12.350
```