

BTTEST-demo

Babak M. Khsoroo

2023-04-27

```
library(BTTEST)
```

Function/Constructor code: myttest()

```
myttest <- function(x,y,alpha, paired)
{
  #handling unequal length of x and y if needed:
  if (length(x) > length(y))
  {
    y <- c(y, rep(NA,(length(x)-length(y))))
  } else if (length(y) > length(x))
  {
    x <- c(x, rep(NA,(length(y)-length(x))))
  }

  if(paired == FALSE)      #If the variables are not paired
  {
    v = var.test(x,y)      #checking if the variances are different

    if(v$p.value > 0.05)    #If we fail to reject NULL (variances of the same)
    {
      #T-test:
      t <- t.test(x, y, var.equal = TRUE)
      t_type <- "T-test"
    }

    else      #If we reject the NULL (variances are different)
    {
      #Welch:
      t <- t.test(x,y, var.equal = FALSE)
      t_type <- "Welch"
    }
  }

  else    #'paired' is TRUE and we want to do a paired test
  {
    t <- t.test(x,y,paired=TRUE)
    t_type <- "Paired"
  }

  #Setting if we have rejected our NULL in the relevant t-test or not:
  if(t$p.value > alpha)
    rej_null <- "NO"
  else
    rej_null <- "YES"

  #creating the object/list:
  object = list(
    test_type = t_type,
    reject_null = rej_null,
    summary_stats = t,
    data = list(x,y),
    alpha = alpha
  )
}
```

```
class(object) <- "Rttest"    #attributing the class Rttest to the object  
  
invisible(object)    #returning the object in an invisible way  
}
```

Function code: print()

```
print.Rttest <- function(x, ...)  
{  
  t <- x$summary_stats  
  cat("$ci\n")  
  print(t$conf.int)  
  cat("\n$Test_type\n")  
  print(x$test_type)  
}
```

Function code: plot()

```
plot.Rttest <- function(x,...)
{
  #If we have done a paired t.test:
  if(x$test_type == "Paired")
  {
    #Extracting each data set in the list as a vector
    x_v <- unlist(x$data[1])
    y_v <- unlist(x$data[2])

    #Creating the difference vector
    d <- x_v - y_v

    #Converting d to a data frame
    d_df <- data.frame(d)

    #For the confidence interval, first extracting conf.int from the test
    #summary:
    ci <- x$summary_stats$conf.int

    #Converting ci to data frame
    ci_df <- data.frame(ci)

    #Plotting the differences (boxplot) with the confidence interval inside it:
    ggplot() +
      geom_boxplot(data=d_df, aes(x="",y=d), fill = "orange") +
      geom_line(data=ci_df,aes(x="",y=ci), linewidth = 2.5) +
      labs(x = "Difference", y = "myd") +
      ggtitle("Differences Boxplot")
  }

  #If we have done non-paired t.test:
  else
  {
    #Extracting each data set in the data as a vector and binding
    #the vectors as a data frame:
    X <- unlist(x$data[1])
    Y <- unlist(x$data[2])
    data_df <- data.frame(X,Y)

    #Reshaping our data frame so that it could be used for side-by-side
    # boxplots:
    data_long <- melt(data_df,)

    #changing the columns name appropriately:
    colnames(data_long) <- c("Category", "Samples")

    #Making side-by-side boxplots:
    ggplot(data_long, aes(x=Category, y=Samples, fill=Category)) +
      geom_boxplot() +
      ggtitle('Boxplots')
```

```
}  
}
```

RESULTS

T-test

```
set.seed(32); x=rnorm(30,mean=10,sd=15)  
set.seed(35); y=rnorm(30,mean=8,sd=15)
```

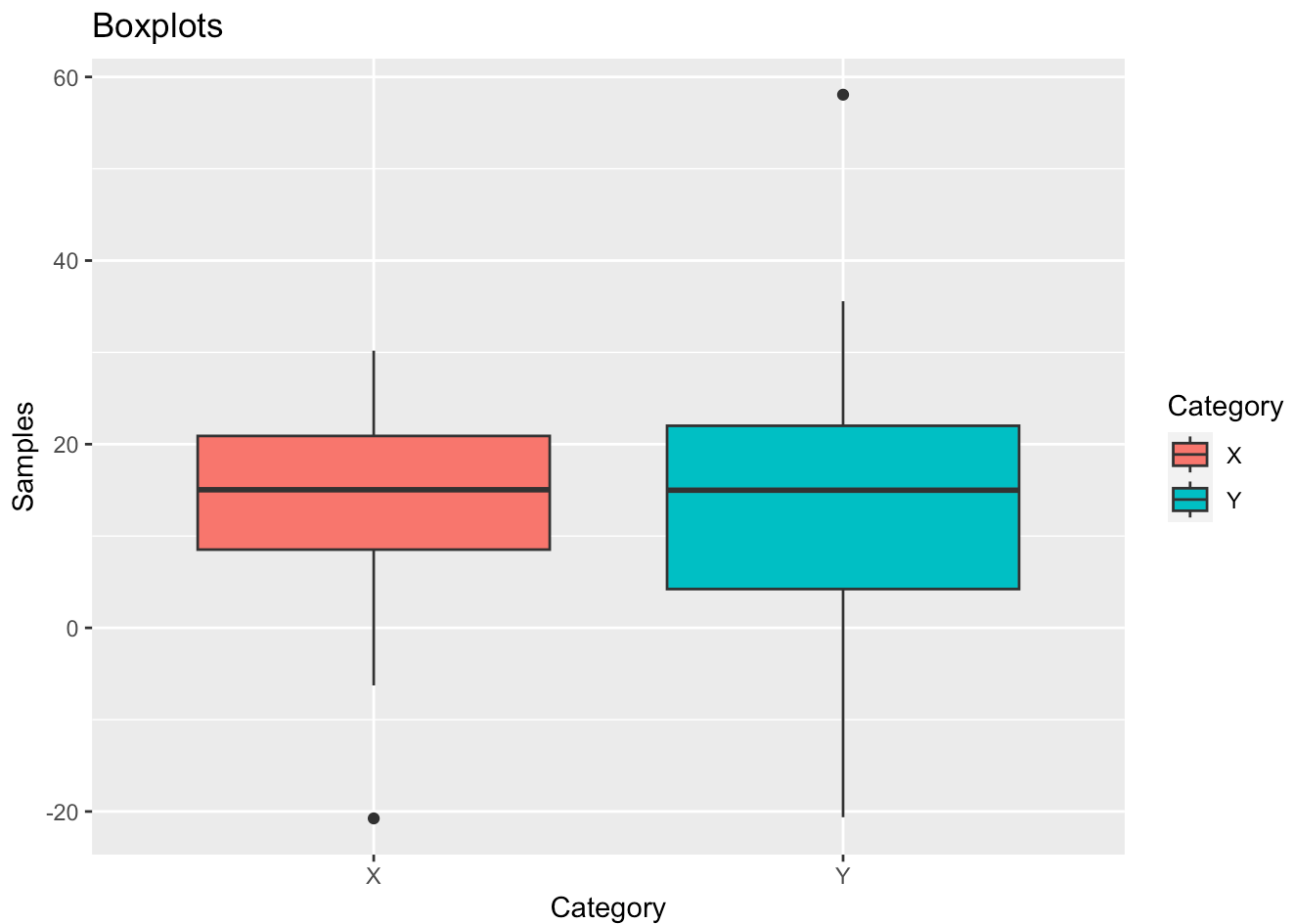
```
ans1=BTTEST::myttest(x,y,alpha=0.05,paired=FALSE)
```

```
print(ans1)
```

```
## $ci  
## [1] -7.623130 6.100201  
## attr("conf.level")  
## [1] 0.95  
##  
## $Test_type  
## [1] "T-test"
```

```
plot(ans1)
```

```
## No id variables; using all as measure variables
```



Welch

```
set.seed(32); x=rnorm(30,mean=10,sd=5)
set.seed(35); y=rnorm(30,mean=8,sd=15)
```

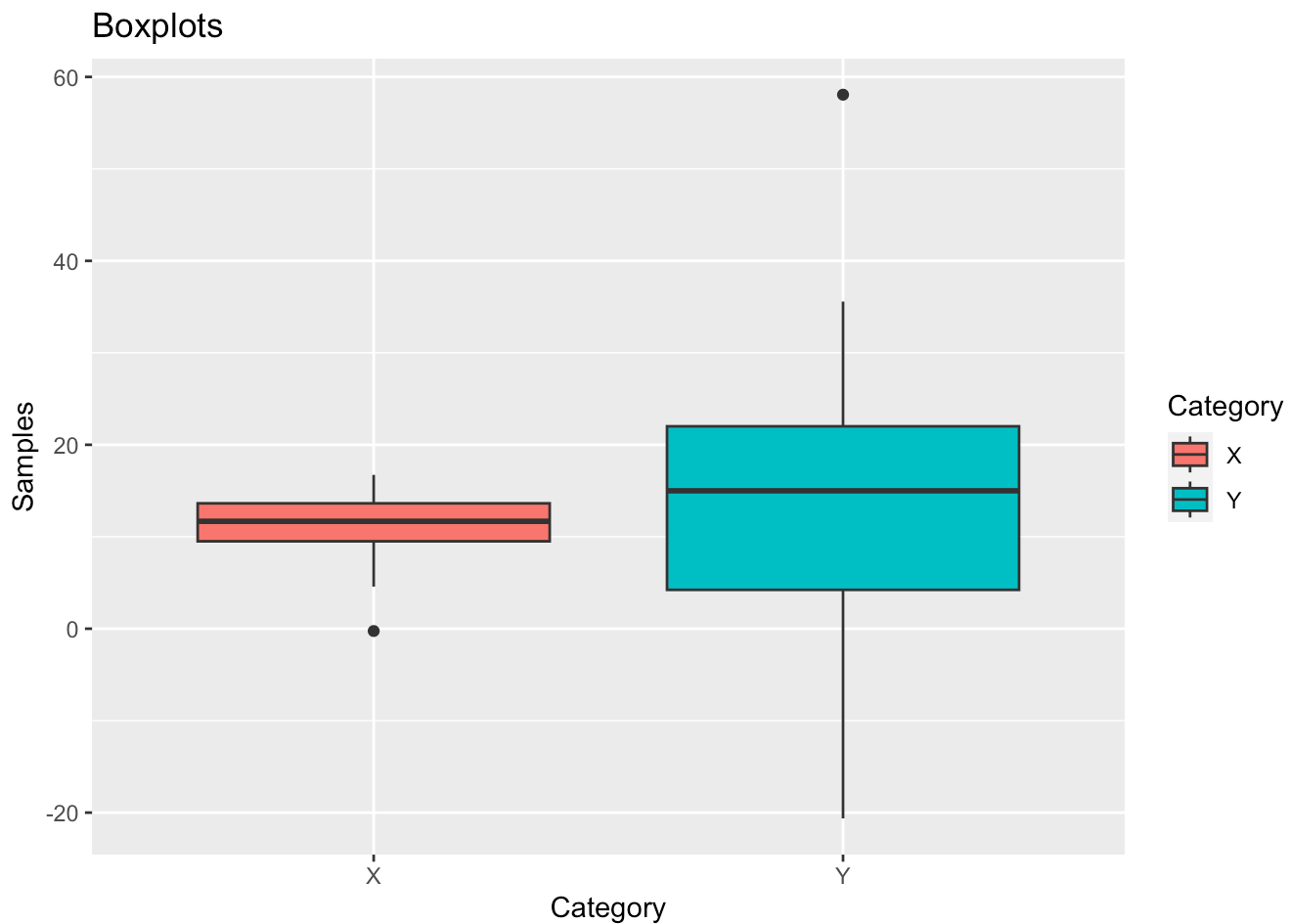
```
ans2=BTTEST::myttest(x,y,alpha=0.05,paired=FALSE)
```

```
print(ans2)
```

```
## $ci
## [1] -8.672098  3.086584
## attr("conf.level")
## [1] 0.95
##
## $Test_type
## [1] "Welch"
```

```
plot(ans2)
```

```
## No id variables; using all as measure variables
```



Paired

```
set.seed(32); x=rnorm(30,mean=10,sd=15)
set.seed(35); y = x+ rnorm(30, 5 ,4)
```

```
ans3=BTTEST::myttest(x,y,alpha=0.05,paired=TRUE)
```

```
print(ans3)
```

```
## $ci
## [1] -8.082480 -5.015335
## attr("conf.level")
## [1] 0.95
##
## $Test_type
## [1] "Paired"
```

```
plot(ans3)
```

