Statistical Hypothesis Testing and Multiple Test Procedure

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(pwr)
library(broom)
```

Data

We will focus on gene expression data related to prostate cancer. The dataset contains measurements of the gene expression of 6033 genes for 102 observations: 50 healthy ($y = 0$) and 50 ill men ($y = 1$).

```
load("/Users/bottimacintosh/Documents/M2_CMB/S3/STATISTICAL_INFERENCE/Practice_1/prostate.rda")
```

In the matrix `prostate.x`:

- the 50 first rows are the healthy individuals,
- the 52 last rows are the individuals suffering from prostate cancer,
- the numerous columns are related to 6033 different genes.

Thus,

```
n <- nrow(prostate.x)      # num of men
m <- ncol(prostate.x)      # num of tests
n0 <- sum(prostate.y == 0) # num of healthy men
n1 <- n - n0               # num of ill men
```

The data come from:

D. Singh et al. 2002. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 1:203–209.

We will not take into account other covariates measured on the 102 men, and will focus only on applying statistical methods of testing.

About hypothesis testing

Statisticians have spent a lot of time to design tests, that is say to find statistics $s(\mathcal{D})$ that:

- have a different behavior depending whether H_0 or H_1 is true,
- have a known distribution under H_0.

Note that when H_i are defined in terms of joint distribution, we also set assumptions such as Gaussian distribution of the population, and so on.

Related to our problem that compare averages, assuming both sub-population are Gaussian, we can rely on two t-test statistics depending whether we assume that the covariances of both sub-populations are equal or not. For simplicity, we will assume that they are equal.

(Please do not do multiple tests to decide whether this is true of not based on data, before using another multiple test procedure…)

t-test

T-tests are simple enough to compute the power function explicitly.

At first, we will focus on gene #4227. The `t.test` function of R can do the computations for you.

```r
gene_id <- 4227
t.test(prostate.x[1:n0, gene_id], prostate.x[n0 + 1:n1, gene_id],
       alternative = "two.sided", var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  prostate.x[1:n0, gene_id] and prostate.x[n0 + 1:n1, gene_id]
## t = 2.0752, df = 100, p-value = 0.04054
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01139997 0.50726483
## sample estimates:
## mean of x mean of y
## 0.9540074 0.6946750
```

1. What is the observed value of T ? And the p-value? Which decision do we take? *Answer.* T=2.075195. p-value=0.04053543.The p-value<0.05 thus, we can say that this gene is deferentially expressed between the two conditions. This interpretation have to be carefully considered because of the very small difference between the obtained p.value and the level alpha=0.05. This difference is only about 0.0095.

```r
test_4227 <- t.test(prostate.x[1:n0, gene_id], prostate.x[n0 + 1:n1, gene_id],
                    alternative = "two.sided", var.equal = TRUE)
test_4227$statistic  #observed value of T
```

```
##        t
## 2.075195
```

```
test_4227$p.value     #observed p.value
```

```
## [1] 0.04053543
```

```
0.05-test_4227$p.value #difference between alpha and the obtained p.value.
```

```
## [1] 0.009464565
```

The observed value of T comes from:

$$T = \frac{\bar{X} - \bar{Y}}{\hat{\sigma}\sqrt{n_1^{-1} + n_2^{-1}}}, \quad \text{where } \hat{\sigma} = \sqrt{\frac{(n_1-1)\hat{\sigma}_1^2 + (n_2-1)\hat{\sigma}_2^2}{n_1+n_2-2}}.$$

and $\hat{\sigma}_1^2$, $\hat{\sigma}_2^2$ are estimates of the standard deviation within the two sub-populations. Note that $\hat{\sigma}\sqrt{n_1^{-1} + n_2^{-1}}$ is an estimate of the standard deviation of $\bar{X} - \bar{Y}$.

The observed values of the three statistics can be obtained as follows.

```
test_4227 <- t.test(prostate.x[1:n0, gene_id], prostate.x[n0 + 1:n1, gene_id],
                    alternative = "two.sided", var.equal = TRUE)
test_4227$estimate
```

```
## mean of x mean of y
## 0.9540074 0.6946750
```

```
test_4227$stderr
```

```
## [1] 0.1249677
```

We want to study the power function, namely the ability to decide in favor of the alternative when it is true. We modeled the data with

$$(X_1, \dots, X_{n_1}, Y_1, \dots, Y_{n_2}) \sim P = \mathcal{N}(\mu_1, \sigma^2)^{\otimes n_1} \otimes \mathcal{N}(\mu_2, \sigma^2)^{\otimes n_2}.$$

2. Quantiles of the t-distribution can be obtained with the `qt` function of R. Outside which interval do we decide in favor of H_1? *Answer.* We decided in favor of H_1 outside the interval [-1.98, 1.98]
Note: alpha=0.05 -> alpha/2=0.025.

```
gene_4227<-prostate.x[,6033]
qt(c(0.025,1-0.025),100) #interval requested. (answer Q2)
```

```
## [1] -1.983972  1.983972
```

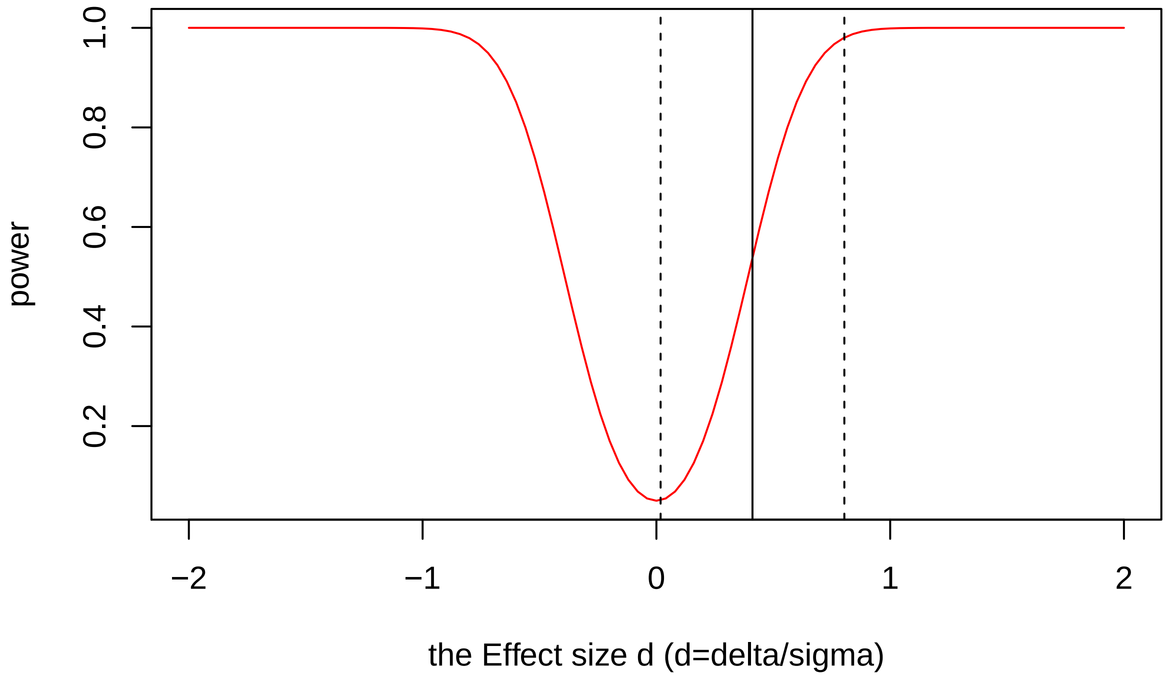
3. The R functions of the `pwr` returns the power of some heavy used tests. Read the vignette of the package to understand how `pwr.t2n.test` can be used. For example, here, the power at the estimated values of the parameters is:

```r
Delta = -diff(test_4227$estimate)

sig <- sqrt(var(prostate.x[1:n0, gene_id])*(n0 - 1) +
              var(prostate.x[n0 + 1:n1, gene_id])*(n1 - 1))/sqrt(100)

pwr.t2n.test(n1 = 50, n2 = 52, d = Delta/sig)
```

```
##
##      t test power calculation
##
##              n1 = 50
##              n2 = 52
##               d = 0.4110288
##       sig.level = 0.05
##           power = 0.5379826
##     alternative = two.sided
```

Plot the power as a function of its parameter d, the effect size, in red. Add a solid vertical black line at $d = (\bar{x} - \bar{y})/\hat{\sigma}$, and two dotted lines at $d = b_i/\hat{\sigma}$, where b_i are the bounds of the confidence interval on $\mu_1 - \mu_2$.
How much power do we have for the test on gene #4227? *Answer.* According to the chunk above, we have 0.54 power which is considered not a good power. In fact, this lower power value indicates that the probability to make a type II error is equal to: beta=1-power=1-0.54=0.46 which is very high.

```r
X_hat<-test_4227$estimate[1]
Y_hat<-test_4227$estimate[2]
conf_int<-test_4227$conf.int[1:2]

d=(X_hat-Y_hat)/sig

curve(pwr.t2n.test(n1 = 50, n2 = 52, d=x)$power, from = -2, to=2, ylab="power", xlab="the Effect size d
abline(v=d)
abline(v=(conf_int[1]/sig), lty="dashed")
abline(v=conf_int[2]/sig, lty="dashed")
```



```r
#delta=difference
#sigma=standard deviation
```

4. Often the power cannot be computed explicitly. We have to rely on simulations to get an approximation. Use a Monte Carlo method to approximate the power at the estimated values of the parameters. Do you get the same result?

Answer. The basic steps for calculating power using Monte Carlo simulations are (from the internet):
1. generate a dataset assuming the alternative hypothesis is true (for example, mean=75).
2. test the null hypothesis on the dataset 3. save the results of the test (for example, “reject” or “fail to reject”).
4. repeat steps 1–2-3 many times (usually 1,000 or more).
Rk: The power is the number of times we have a significant result (p-val<=5%)

```r
nb_healthy<-length(prostate.x[1:n0,gene_id])
nb_cancer<-length(prostate.x[1:n0,gene_id])

MonteCarlo<-function(sample_size){
  MC_output_vec<-rep(0,sample_size)
  for(i in 1:sample_size){
  rand.healthy.data<-rnorm(n=nb_healthy,
                           mean=mean(prostate.x[1:n0,gene_id]),
                           sd=sd(prostate.x[1:n0, gene_id]))
  rand.cancer.data<-rnorm(n=nb_cancer,
                          mean=mean(prostate.x[(n0+1):n1, gene_id]),
                          sd=sd(prostate.x[(n0+1):n1, gene_id]))
```

```
      sig_value=sqrt(var(rand.healthy.data)*(nb_healthy-1) +
                       var(rand.cancer.data)*(nb_cancer-1))/sqrt(100)
      Delta_value=-diff(test_4227$estimate)
      power_value_temp<-pwr.t2n.test(n1=nb_healthy, n2=nb_cancer,
                                     d=Delta_value/sig_value)
      MC_output_vec[i]<-power_value_temp$power
  }
  return(mean(MC_output_vec))
}

MonteCarlo(3000)
```

```
## [1] 0.7716875
```

5. How does the power impact the decision we have taken? To better understand what is happening, let us be almost Bayesian and assume that

$$Pr(d=1|H_1)=0.54, \quad Pr(d=1|H_0)=0.05, \quad Pr(H_0)=Pr(H_1)=0.5.$$

5.a) What is the value of $Pr(H_1|d=1)$? (And if $Pr(d=1|H_1)=0.3$?)
Answer. (done by hand)
- if $Pr(d=1|H_1)=0.54$, then $Pr(H_1|d=1)=0.92$
- if $Pr(d=1|H_1)=0.3$, then $Pr(H_1|d=1)=0.50$

5.b) Do you really think that, for a gene picked at random among the 6033 genes, a prior probability of $Pr(H_1)=0.5$ is reasonable?
Answer. From a biological point of viez, we expect a very few number of genes among the 6033 that are differentially expressed. From this prior belief, we should better set a out prior probabilities as $Pr(H_0) >> Pr(H_1)$, i.g. 0.9 and 0.9 respectively. Thus, a prior probability of $Pr(H_1)=0.5$ is not reasonable in this case.

5.c) What is the value $Pr(H_1|d=1)$ when $Pr(H_1)=10\%$ or $Pr(H_1)=1\%$?
Knowing that $Pr(d=1|H_1)=0.54$ (done by hand) :
- if $Pr(H_1)=0.10$, then $Pr(H_1|d=1)=0.5454545$
- if $Pr(H_1)=0.01$, then $Pr(H_1|d=1)=0.1071429$

6. Use `power.t.test` to find the minimal size of each sub-sample to obtain a power of 0.9 at $d=(\bar{x}-\bar{y})/\hat{\sigma}$. And to obtain a power 0.95? *Answer.* data.frame of next chunk

```
n_0.90<-power.t.test(power=0.90,delta=(X_hat-Y_hat)/sig, alternative = "two.sided" , type = c("two.sampl
n_0.95<-power.t.test(power=0.95,delta=(X_hat-Y_hat)/sig, alternative = "two.sided" , type = c("two.sampl
data.frame(power=c("0.90","0.95"),min_size_n=c(round(n_0.90),round(n_0.95)))
```

```
##   power min_size_n
## 1  0.90        125
## 2  0.95        155
```

7.a Does the power influence the quality of the decision when $d=0$?
Answer. When p-value>0.05, we accept H_0, i.e. $d=0$. However, by assuming H0, we can make a type II error (beta). Thus, the value of beta influence the quality of the decision $d=0$. Knowing that beta is related to the power by beta=1-power, we can then confirm that YES the power influence the quality of the decision when $d=0$.

7.b. Repeat all the above analysis on gene #877. *Answer.* See Code Chunk and Displayed Table
For gene 877, p.value>0.05. Thus, we accept H0, i.e. gene 877 is not dfferentially expressed.

```
#get the gene column from the gene expression matrix
gene_877_col<-prostate.x[,877]

#t.test two sample , assuming var equal and H two sided
gene_877_ttest<-t.test(x=gene_877_col[0:n0],y=gene_877_col[(n0+1):n],
                       alternative = "two.sided", var.equal = TRUE)
gene_877_ttest_row<-tibble(tidy(gene_877_ttest))
gene_877_ttest_row$statistic #belong to [-1.98,1.98] <=> p-value>0.05 => d=0.
```

```
##        t
## 1.794056
```

```
#compute the power
#need d value as : d=(estimatedX-estimatedY)/estimated_sigma
Delta_877<-gene_877_ttest_row$estimate # <=> estimatedX-estimatedY :t.test outputs
Delta_877<-gene_877_ttest_row$estimate1-gene_877_ttest_row$estimate2 #same

sig_877<-sqrt(var(gene_877_col[1:n0])*(n0 - 1) +
              var(gene_877_col[n0 + 1:n1])*(n1 - 1))/sqrt(100) #formula

d_arg=Delta_877/sig_877

#find the minimal size of n1 and n2 to get a power=0.90 at d=Delta_877/sigma_877
min_n_for_power_90<-power.t.test(power=0.90, delta=Delta_877)$n

#Dataframe to display
gene_877_power<-pwr.t2n.test(n1=50, n2=52, d=d_arg)$power
outputs_df<-data.frame(variables=c("gene_nb","p-val t.test", "power","min size for power=0.90"),values=
outputs_df
```

```
##                 variables           values
## 1                 gene_nb              877
## 2            p-val t.test 1.79405642949373
## 3                   power  0.4274365755511
## 4 min size for power=0.90 1652.66300191726
```

Mutliple tests

Computing T-values and p-values

1. We have to run `t.test` as many times as genes in our dataset. This can be done easily with a `for`-loop. Yet, the value returned by this function is a list of 10 R objects. Some of them are 2-dimensional vectors. Which one? *Answer.* These 2 dimensional vectors are the confidence interval and the sample estimates.

The `tidy` function of the package named `broom` allows us to transform this list in a one-row table. Apply it on `test_4227` to see the result. Which fields of the list are dropped by `tidy`? *Answer.* See Chunk

```
colnames(tibble(tidy(test_4227)))
```

```
##  [1] "estimate"    "estimate1"   "estimate2"   "statistic"   "p.value"
##  [6] "parameter"   "conf.low"    "conf.high"   "method"      "alternative"
```

The table we will fill with the `for`-loop can be initialized as follows.

```
mytests <- tibble(tidy(test_4227), .rows = 6033) #table preparetion
```

```
for (i in 1:6033){
  test_result<-t.test(prostate.x[1:n0, i], prostate.x[(n0+1):(n0+n1) ,i],
                      alternative = "two.sided", var.equal = TRUE)
  mytests[i,]<-tibble(tidy(test_result))
}
head(mytests,2)
```

```
## # A tibble: 2 x 10
##   estimate estimate1 estimate2 statistic p.value parameter conf.low conf.high
##      <dbl>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl>
## 1    0.134    -0.555    -0.689      1.67 0.0984        100  -0.0254     0.294
## 2    0.227    -0.550    -0.777      3.16 0.00209       100   0.0844     0.369
## # ... with 2 more variables: method <chr>, alternative <chr>
```

2. Since `estimate` is $\bar{x} - \bar{y}$ and `statistic` is the observed value of

$$T = \frac{\bar{X} - \bar{Y}}{\hat{\sigma}\sqrt{n_1^{-1} + n_2^{-1}}}, \quad \text{where } \hat{\sigma} = \sqrt{\frac{(n_1 - 1)\hat{\sigma}_1^2 + (n_2 - 1)\hat{\sigma}_2^2}{n_1 + n_2 - 2}},$$

$$then, \hat{\sigma} = \frac{\bar{X} - \bar{Y}}{T\sqrt{n_1^{-1} + n_2^{-1}}}$$

Use `mutate` to add a new column `sd` to `mytests` that gives the observed value of $\hat{\sigma}$. *Answer.* Chunk

```
mytests<- mytests %>% mutate(sd_observed=estimate/(statistic*sqrt(1/n0+1/n1)))
```

3. Compute the power at the estimated values of the parameters and $Pr(H_1|d = 1)$ as in the first part, assuming the prior probability of H_1 is 50%. Add two columns named `power` and `posterior1` to the table with these values. *Answer.* Power : compute and add column: see Chunk

```
power_vec<-rep(0,nrow(mytests))

for(i in 1:6033){

  delta=mytests$estimate
  sigma=mytests$sd_observed
  d_vec=delta/sigma

  power_vec[i]<-power.t.test(power=NULL, #power= what we are searching for.
           n=n0,
             delta=d_vec[i],
             sd=mytests$sd_observed[i],
             sig.level=0.05,
             alternative = "two.sided" , type = c("two.sample"))$power
}
mytests<-mytests %>% mutate (power=power_vec)
```

Now calculate the posterior probability $Pr(H_1|d=1)$ (called posetior1), and add it to the table. *Answer.*

$$Pr(H_1|d=1) = frac(Pr(d=1|H1)Pr(H_1))Pr(d=1|H0)Pr(H0) + Pr(d=1|H1)Pr(H1)$$

```
prior_H0<-0.5
prior_H1<-1-0.5
alpha<-0.05

mytests<- mytests %>% mutate(posterior1=(power*prior_H1)/(prior_H0*alpha+prior_H1*power))
head(mytests,2)
```

```
## # A tibble: 2 x 13
##   estimate estimate1 estimate2 statistic p.value parameter conf.low conf.high
##      <dbl>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl>
## 1    0.134    -0.555    -0.689      1.67 0.0984        100  -0.0254     0.294
## 2    0.227    -0.550    -0.777      3.16 0.00209       100   0.0844     0.369
## # ... with 5 more variables: method <chr>, alternative <chr>,
## #   sd_observed <dbl>, power <dbl>, posterior1 <dbl>
```

And another column named `posterior0` with the values of $Pr(H_1|d=0)$. *Answer.*

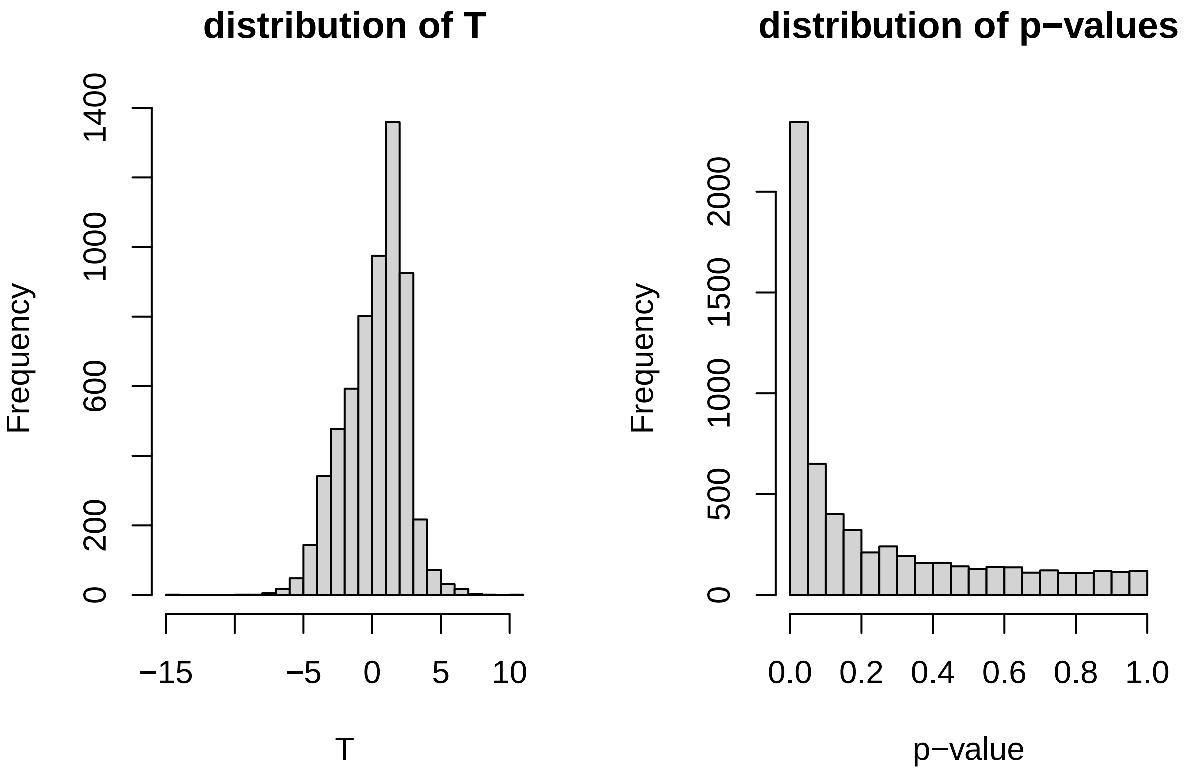
```
prior_H0<-0.5
prior_H1<-1-0.5
alpha<-0.05

mytests<- mytests %>% mutate(posterior0=(alpha*prior_H1)/(prior_H0*alpha+prior_H1*power))
head(mytests,2)
```

```
## # A tibble: 2 x 14
##   estimate estimate1 estimate2 statistic p.value parameter conf.low conf.high
##      <dbl>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl>
## 1    0.134    -0.555    -0.689      1.67 0.0984        100  -0.0254     0.294
## 2    0.227    -0.550    -0.777      3.16 0.00209       100   0.0844     0.369
## # ... with 6 more variables: method <chr>, alternative <chr>,
## #   sd_observed <dbl>, power <dbl>, posterior1 <dbl>, posterior0 <dbl>
```

Naive multiple test procedures

1. Plot the distribution of T's and p-values observed on the 6033 genes of the dataset.

```
par(mfrow=c(1,2))
hist(mytests$statistic, breaks=30, main="distribution of T", xlab="T")
hist(mytests$p.value, breaks=20, main="distribution of p-values", xlab="p-value")
```



Can you draw a conclusion based on the distribution of the p-values? *Answer.* The p-value distribution plot shows that very high number (computed in the next question) of genes show a p-value<0.05. As we are performing multiple hypothesis testing, we cannot conclude by simply considering the p-values. We may use correction procedure to get the adjusted p.values.

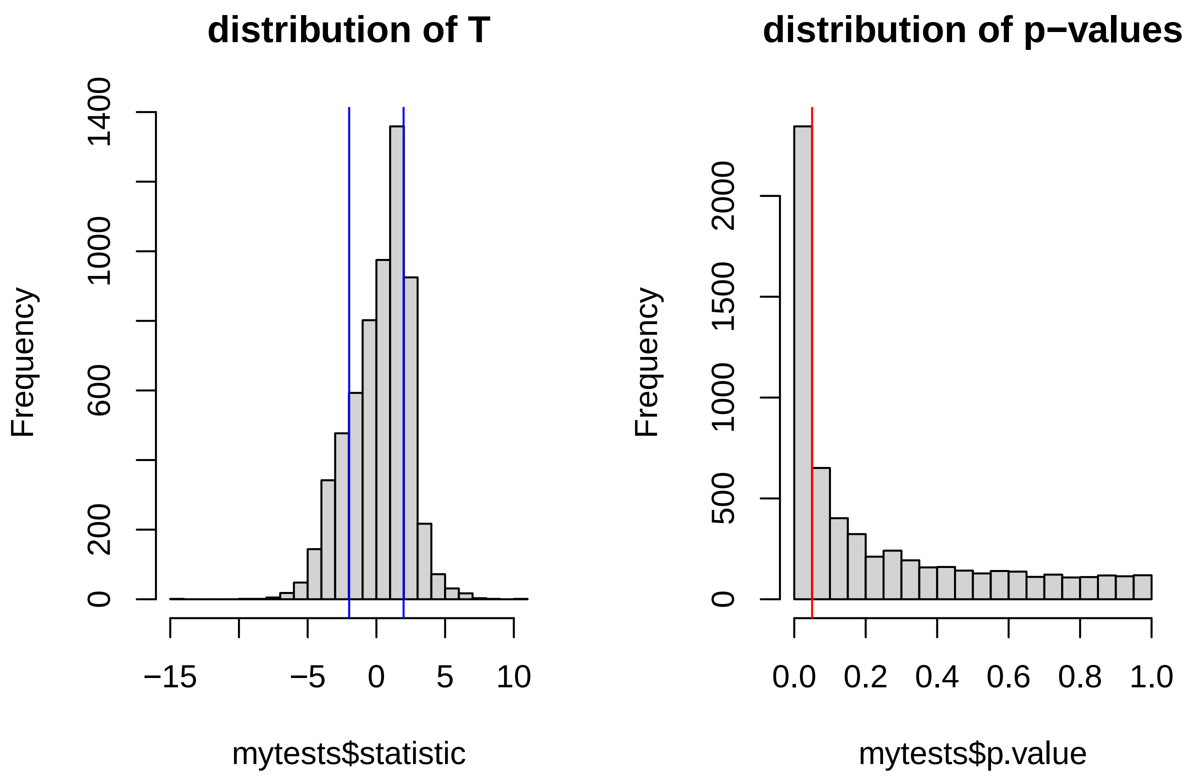
2. Which and how many genes satisfy $p_i(\mathcal{D}) \leq \alpha$? *Answer.* number of significant p-values : 2345 (over 6033 genes).

```
significant_genes_nb<-mytests %>% filter(p.value<=0.05) %>% nrow()
significant_genes_list<-which(mytests$p.value<=0.05)
names(significant_genes_nb)<-"significant_genes_nb"
significant_genes_nb
```

```
## significant_genes_nb
##                 2345
```

3. On the above plots, show the values of T and p that satisfy this inequality. ***Answer***
On the T plot, the T values satisfying the inquality pi(D)<0.05 are those outside the interval $[-1.96; 1.96]$. They are outside the zone delimited by the vertical two blue lines. On the p plot, the p values satisfying the inquality pi(D)<0.05 are presented on the left side of the red vertical line.

```
par(mfrow=c(1,2))
hist(mytests$statistic, breaks=30, main="distribution of T")
abline(v=-1.98,col="blue") ; abline(v=1.98,col="blue")
hist(mytests$p.value, breaks=20, main="distribution of p-values")
abline(v=0.05, col="red")
```



4. Show on a dataset where individuals are swapped at random that the naive procedure is unsatisfactory.

Answer.: main idea : step 1= redistribute the rows of mytests randomly
step 2= apply a multiple test to compare between the first 50 rows and the last 52.
If this test shows a significant difference between the two groups, it means that the previous procedure is not a good one.
We get 1470 p.value<0.05 in the random dataset (901»>0)
=> Thus, this approach is unsatisfactory.

```
#randomize the rows on the x.prostate datset
row_nb_rand<-sample(1:102,102,replace=FALSE)

prostate.random<-as.data.frame(matrix(0,nrow=102,ncol=6033))

for(i in 1:102){
  rand_row_temp<-row_nb_rand[i]
  prostate.random[i,]<-prostate.x[rand_row_temp,]
}
```

```
#initialisation
mytests.random<-as.data.frame(matrix(nrow=6033,ncol=10))

#test
for (i in 1:6033){
  test_temp<-t.test(prostate.random[1:n0, i], prostate.random[(n0+1):(n0+n1) ,i],alternative = "two.sid
  mytests.random[i,]<-tibble(tidy(test_temp))
}
```

```
colnames(mytests.random)<-colnames(mytests[1:10])
head(mytests.random,2)
```

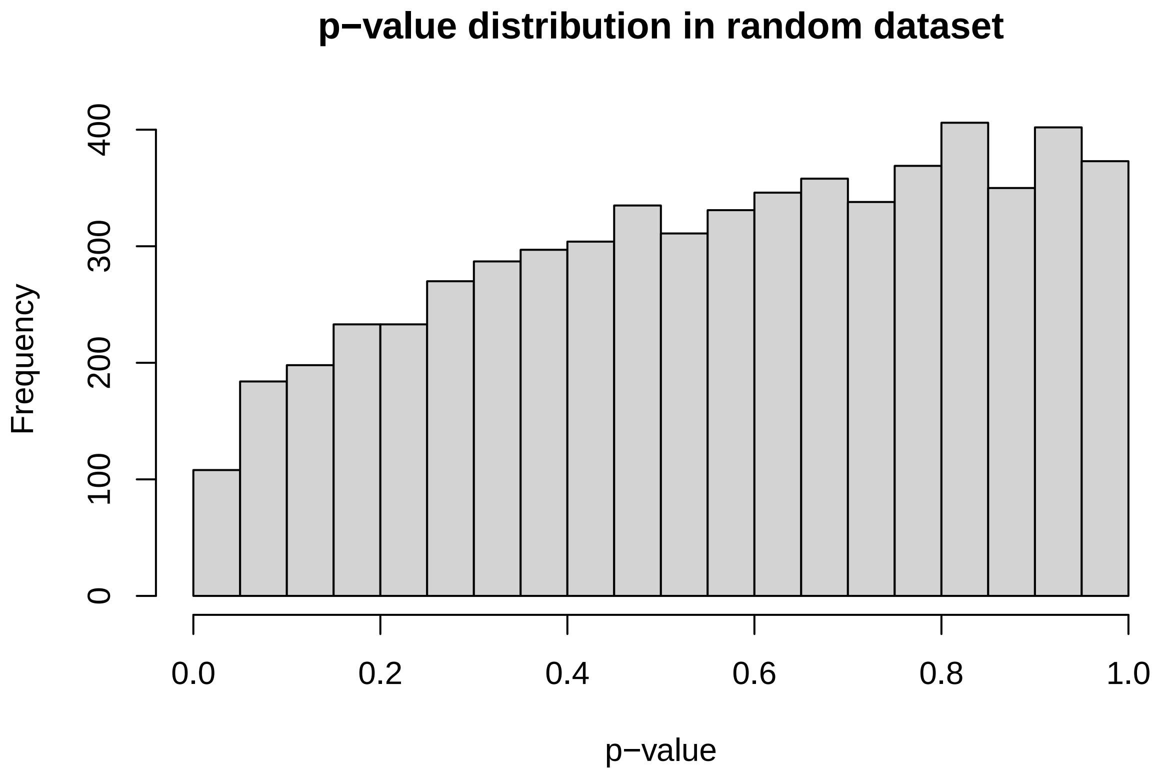
```
##     estimate  estimate1  estimate2  statistic    p.value parameter   conf.low
## 1 -0.0533715 -0.6502947 -0.5969232 -0.6555153 0.51364127       100 -0.2149048
## 2 -0.1243743 -0.7292151 -0.6048408 -1.6750800 0.09704304       100 -0.2716837
##    conf.high            method alternative
## 1 0.10816177 Two Sample t-test   two.sided
## 2 0.02293512 Two Sample t-test   two.sided
```

```
mytests.random.pval.vec<-mytests.random$p.value
significant_pval_count_random<-sum(mytests.random.pval.vec<=0.05)
names(significant_pval_count_random)<- "number of significant p-val in random table"
significant_pval_count_random
```

```
## number of significant p-val in random table
##                                         108
```

5. Plot the distribution of the p-values computed on the swapped dataset. Can you draw a conclusion based on the distribution of the p-values? *Answer.* As we are performing a multiple hypothesis testing procedure, we cannot draw a conclusion without p.values adjustment.

```
hist(mytests.random$p.value,breaks=20, main = "p-value distribution in random dataset", xlab="p-value")
```



Procedures that controls the FWER

We now focus on procedures that controls the FWER.

1. Compute with your own function the adjusted p-values of the Bonferroni procedure ($k = 1$), and find which and how many genes are "discovered" by this procedure?

Answer.

```
GetAdjustedPvaluesBonferroni<-function(pval_vec,k){
    m<-length(pval_vec)
    padj_vec<-as.vector(rep(0,m))
    for(i in 1:m){
      padj_temp<-(pval_vec[i]*m)/k
      padj_vec[i]<-padj_temp
    }
    return(padj_vec)
}
```

```
mytests_pval_vec<-mytests$p.value
mytests_padj_bonf_vec<-GetAdjustedPvaluesBonferroni(mytests_pval_vec,1)

#nb of discoveries with and without Bonferroni
significant_pval_count<-sum(mytests_pval_vec<=0.05)
significant_padj_bonf_count<-sum(mytests_padj_bonf_vec<=0.05)
data.frame(name=c("nb of p-val<0.05","nb of p-adj<0.05 (bonferroni)"),
           values=c(significant_pval_count, significant_padj_bonf_count))
```

```
##                            name values
## 1              nb of p-val<0.05   2345
## 2 nb of p-adj<0.05 (bonferroni)    168
```

```
#To identify genes that have p.adj<0.05, use the next command
#which(mytests_padj_bonf_vec<=0.05)
```

```
sum(p.adjust(p=mytests_pval_vec, method="bonferroni", n=6033)<=0.050)
```

```
## [1] 168
```

```
# We obtain exactly the same result.
```

2. Which procedure is better than Bonferroni's one to control the FWER? Use the R function `p.adjusted` to compute the adjusted p-values associated to this procedure and draw the conclusion. *Answer.* For k=1, the Holm procedure is known to be better than the Bonferroni procedure. (Note that if genes were considered independent from each other, Sidak would be better.)

```
mytests_pval_vec<-mytests$p.value
mytests_padj_holm_vec<-p.adjust(p=mytests_pval_vec, method = "holm",
                                n=6033)

#nb of p.adj holm <0.05
significant_padj_holm_count<-sum(mytests_padj_holm_vec<=0.05) ; significant_padj_holm_count
```

```
## [1] 169
```

Conclusion : Using Holm correction, the number of p-values<0.05 is equal to 169. Thus, we now can conclude that only 169 of the 6033 genes of our dataset are differentially expressed between normal patients and patients with prostate cancer.

3. What are the results returned by both procedures on the swapped dataset? *Answer.* see table output table below.

```
padj_bonf_vec_random<-p.adjust(p=mytests.random$p.value, method="bonferroni",
                                                 n=6033)
padj_holm_vec_random<-p.adjust(p=mytests.random$p.value, method="holm",
                                                 n=6033)

significant_padj_bonf_count_random<-sum(padj_bonf_vec_random<=0.05)
significant_padj_holm_count_random<-sum(padj_holm_vec_random<=0.05)

data.frame(
  correction=c("bonferroni","holm"),
  nb_padj_sig=c(significant_padj_bonf_count_random,
                significant_padj_holm_count_random)
)
```

```
##   correction nb_padj_sig
## 1 bonferroni           0
## 2       holm           0
```

Benjamini-Hochberg

1. Use the same R function as before to compute the adjusted p-value of the BH procedure. How many and which genes are "discovered"?

```
padj_bh_vec<-p.adjust(p=mytests$p.value, method = "fdr", n=6033)
padj_bh_vec_random<-p.adjust(p=mytests.random$p.value,method = "fdr", n=6033)

significant_padj_bh_count<-sum(padj_bh_vec<=0.05);
significant_padj_bh_random_count<-sum(padj_bh_vec_random<=0.05);

#resume with data frame

BH_df<-data.frame(
  dataset=c("original","random"),
  nb_sign_pval_no_correction<-c(sum(mytests$p.value<=0.05),
                 sum(mytests.random$p.value<=0.05)),
  nb_sign_pval_BH<-c(significant_padj_bh_count,significant_padj_bh_random_count))
colnames(BH_df)<-c("dataset","nb of pval<0.05","nb of padj<0.05 (BH)")
BH_df
```

```
##    dataset nb of pval<0.05 nb of padj<0.05 (BH)
## 1 original            2345                 1349
## 2   random             108                    0
```

2. Compare with the results on the swapped dataset. *Answer.* Done in the previous question. See displayed data.frame.

## The Romano-Wolf procedure : NOT TO DO