13 Multiple Testing

By: Udit (based on ISLR)

Review of T-test

Simple t-tests using 100 variables, each consisting of 10 observations The first 50 variables have a non-zero mean of 0.5 by design and variance of 1, while others have mean of 0.

```
set.seed(6)
x = matrix(rnorm(10*100),10,100)
x[,1:50] = x[,1:50] + 0.5
dim(x)
```

```
## [1]  10 100
```

```
t.test(x[,1],mu=0)
```

```
##
##  One Sample t-test
##
## data:  x[, 1]
## t = 2.0841, df = 9, p-value = 0.06682
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.05171076  1.26242719
## sample estimates:
## mean of x
## 0.6053582
```

```
p.values = rep(0,100)
for(i in 1:100){
  p.values[i] = t.test(x[,i], mu=0)$p.value
  }

decision = ifelse(p.values <= 0.05, "Reject Null", "Do Not Reject")
table(decision, true = c(rep("Reject Null",50), rep("Do Not Reject", 50)))
```

```
##                true
## decision        Do Not Reject Reject Null
##   Do Not Reject            47          40
##   Reject Null               3          10
```

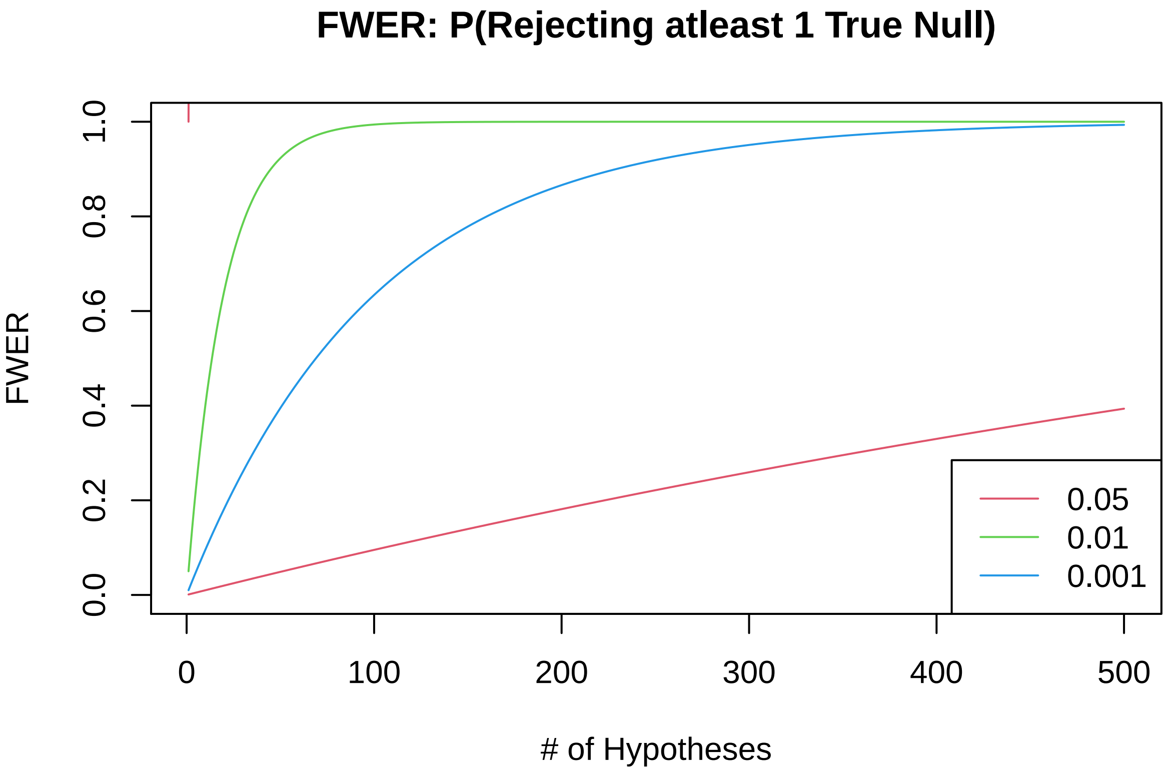
At $\alpha = 0.05$ we reject just 11 our of 50 false null hypotheses. And we would incorrectly reject 3 of the true null hypotheses.

```
# Using Stronger Signal/Noise ratio
x = matrix(rnorm(10*100),10,100)
x[,1:50] = x[,1:50] + 1
p.values = rep(0,100)
for(i in 1:100){
  p.values[i] = t.test(x[,i], mu=0)$p.value
  }

decision = ifelse(p.values <= 0.05, "Reject Null", "Do Not Reject")
table(decision, true = c(rep("Reject Null",50), rep("Do Not Reject", 50)))
```

```
##                  true
## decision          Do Not Reject Reject Null
##   Do Not Reject              49           9
##   Reject Null                 1          41
```

Family Wise Error Rate (FWER)

```
m = 1:500
fwer1 = 1 - (1-0.05)^m
fwer2 = 1 - (1-0.01)^m
fwer3 = 1 - (1-0.001)^m

matplot(cbind(m,fwer1,fwer2, fwer3), type="l", ylim=c(0,1), col=c(2,3,4), lty=1,
        ylab = "FWER", xlab = "# of Hypotheses",
        main = "FWER: P(Rejecting atleast 1 True Null)")
legend("bottomright", legend=c(0.05, 0.01, 0.001), col=c(2,3,4), lty=1)
```



FWER - Fund Manager Dataset

```r
library(ISLR2)
fund.mini <- Fund[,1:5]
t.test(fund.mini[,1], mu=0)
```

```
##
##  One Sample t-test
##
## data:  fund.mini[, 1]
## t = 2.8604, df = 49, p-value = 0.006202
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.8923397 5.1076603
## sample estimates:
## mean of x
##         3
```

```r
fund.p = rep(0,5)
for (i in 1:5){
  fund.p[i] = t.test(fund.mini[,i], mu=0)$p.value
}

fund.p
```

```
## [1] 0.006202355 0.918271152 0.011600983 0.600539601 0.755781508
```

```r
# Bonderroni adjustment alpha/m
p.adjust(fund.p, method="bonferroni")
```

```
## [1] 0.03101178 1.00000000 0.05800491 1.00000000 1.00000000
```

```r
#Holm's adjustment
p.adjust(fund.p, method = "holm")
```

```
## [1] 0.03101178 1.00000000 0.04640393 1.00000000 1.00000000
```

Because the paired t-test below was conducted after visual inspection of the 5 fund managers, in essence, we already carried out 5C2 pairwise comparison through visual inspection. Therefore the p-value should be adjusted for this using **Tukey's HSD (Honest Significant Difference) method.**

```r
#Paired t-test
apply(fund.mini, 2, mean)
```

```
## Manager1 Manager2 Manager3 Manager4 Manager5
##      3.0     -0.1      2.8      0.5      0.3
```

```r
t.test(fund.mini[,1], fund.mini[,2], paired=T)
```

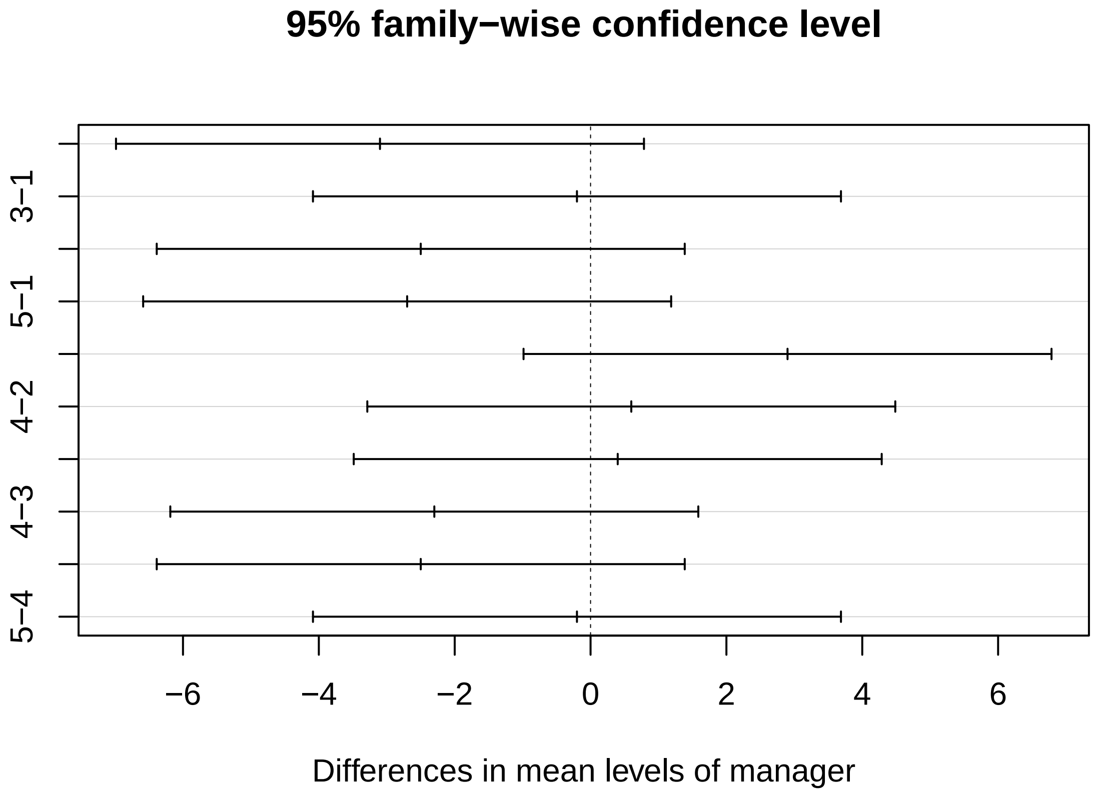
```
##
##  Paired t-test
##
## data:  fund.mini[, 1] and fund.mini[, 2]
## t = 2.128, df = 49, p-value = 0.03839
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  0.1725378 6.0274622
## sample estimates:
## mean of the differences
##                     3.1
```

```r
returns = as.vector(as.matrix(fund.mini))
manager = rep(c("1","2","3","4","5"), rep(50,5))
a1 = aov(returns ~ manager) #ANOVA
TukeyHSD(x = a1) #diff between M1 & M2 is no longer significant
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = returns ~ manager)
##
## $manager
##     diff        lwr       upr     p adj
## 2-1 -3.1 -6.9865435 0.7865435 0.1861585
## 3-1 -0.2 -4.0865435 3.6865435 0.9999095
## 4-1 -2.5 -6.3865435 1.3865435 0.3948292
## 5-1 -2.7 -6.5865435 1.1865435 0.3151702
## 3-2  2.9 -0.9865435 6.7865435 0.2452611
## 4-2  0.6 -3.2865435 4.4865435 0.9932010
## 5-2  0.4 -3.4865435 4.2865435 0.9985924
## 4-3 -2.3 -6.1865435 1.5865435 0.4819994
## 5-3 -2.5 -6.3865435 1.3865435 0.3948292
## 5-4 -0.2 -4.0865435 3.6865435 0.9999095
```

```r
plot(TukeyHSD(x = a1))
```



## False Discovery Rate (FDR)

Far too many tests to control for FWER (since that would be impossibly punitive and lead to extremely few 'discoveries'.) Instead, we focus on FDR: expected fration of rejected null hypotheses that are actually false positives.

```
fund.p = rep(0,2000)
for (i in 1:2000){
  fund.p[i] = t.test(Fund[,i],mu=0)$p.value
}
fund.p[1:5]
```

```
## [1] 0.006202355 0.918271152 0.011600983 0.600539601 0.755781508
```

```
# Benjamini-Hochberg adjustment
q.BH = p.adjust(fund.p, method="BH")
q.BH[1:10] # q.value is the lower FDR at which that H0 can be rejected
```

```
##  [1] 0.08988921 0.99149100 0.12211561 0.92342997 0.95603587 0.07513802
##  [7] 0.07670150 0.07513802 0.07513802 0.07513802
```

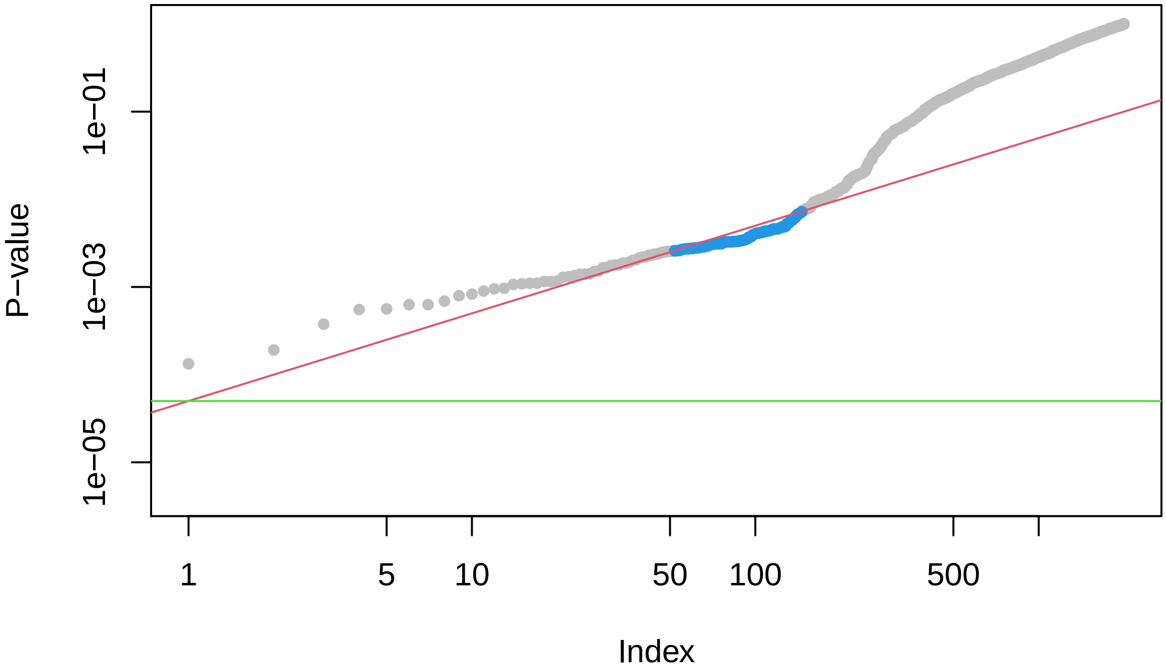
```
# Rejected Nulls for FDR of 10%
sum(q.BH <= 0.1) # 146.. we can expect ~15 of these to be false positives
```

```
## [1] 146
```

```
# Bonferroni would be extremely punitive
sum(fund.p <= (0.1/2000)) # 0 discoveries
```

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

```
# BH method - arrange p-values, compare with q*j/m
m = length(fund.p)
p = sort(fund.p)
q = .1
idx = which(p < q*(1:m)/m)

plot(p, log="xy", ylim=c(4e-6,1), ylab="P-value", xlab="Index", main="", pch=20, col="gray")
points(idx, p[idx], col=4, pch=20)
abline(a=0, b = (q/m), col=2, untf=TRUE)
abline(h=0.1/2000, col=3)
```



Resampling Approach

```
attach(Khan) #Khan gene data
x = rbind(xtrain, xtest)
y = c(ytrain, ytest)
table(y) # Four classes of cancer
```

```
## y
##  1  2  3  4
## 11 29 18 25
```

```
# Comparing 11 gene's difference b/w class 2 and 4
x1 = x[which(y==2),]
x2 = x[which(y==4),]
t.out = t.test(x1[,11], x2[,11], var.equal=TRUE)
t.out$statistic; t.out$p.value   # p-value based on 'theoretical' distribution
```

```
##         t
## -2.093633
```

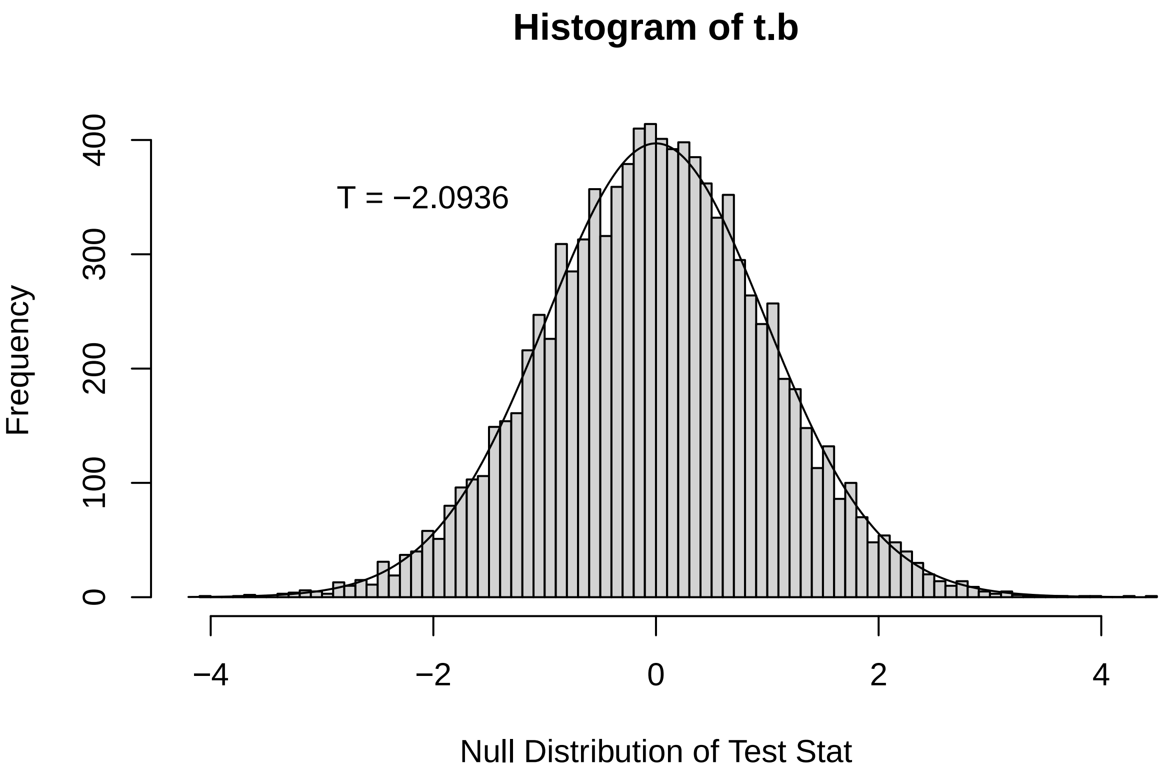
```
## [1] 0.04118644
```

```
# Re-sampling to build empirical distribution
n1 = nrow(x1)
n2 = nrow(x2)

set.seed(1)
```

```
b = 10000
t.b = rep(NA,b)
for(i in 1:b){
  dat = sample(c(x1[,11],x2[,11])) # jumbles up all values
  t.b[i] = t.test(dat[1:n1], dat[(n1+1):(n1+n2)], var=T)$statistic
}
mean(abs(t.b) >= abs(t.out$statistic)) #0.0416.. same as theoritical distribution
```

```
## [1] 0.0416
```

```
hist(t.b, breaks=100, xlim=c(-4.2, 4.2), xlab="Null Distribution of Test Stat")
lines(seq(-4.2,4.2,len=1000),dt(seq(-4.2,4.2,len=1000), df=(n1+n2-2))*1000)
text(t.out$statistic, 350, paste("T = ", round(t.out$statistic,4), sep=""))
```



Calculating FDR for all 2,308 genes.

```
m = 50  #taking 50 genes at random
set.seed(1)
index = sample(ncol(x), m)
Ts = rep(NA, m)
Ts.star = matrix(NA, ncol = m, nrow = b)
for(j in 1:m){
  k = index[j]
  Ts[j] = t.test(x1[,k], x2[,k], var.equal=TRUE)$statistic
  for (i in 1:b){
    dat = sample(c(x1[,k],x1[,k]))
    Ts.star[i,j] = t.test(dat[1:n1], dat[(n1+1):(n1+n2)], var.equal=TRUE)$statistic
  }
}
```

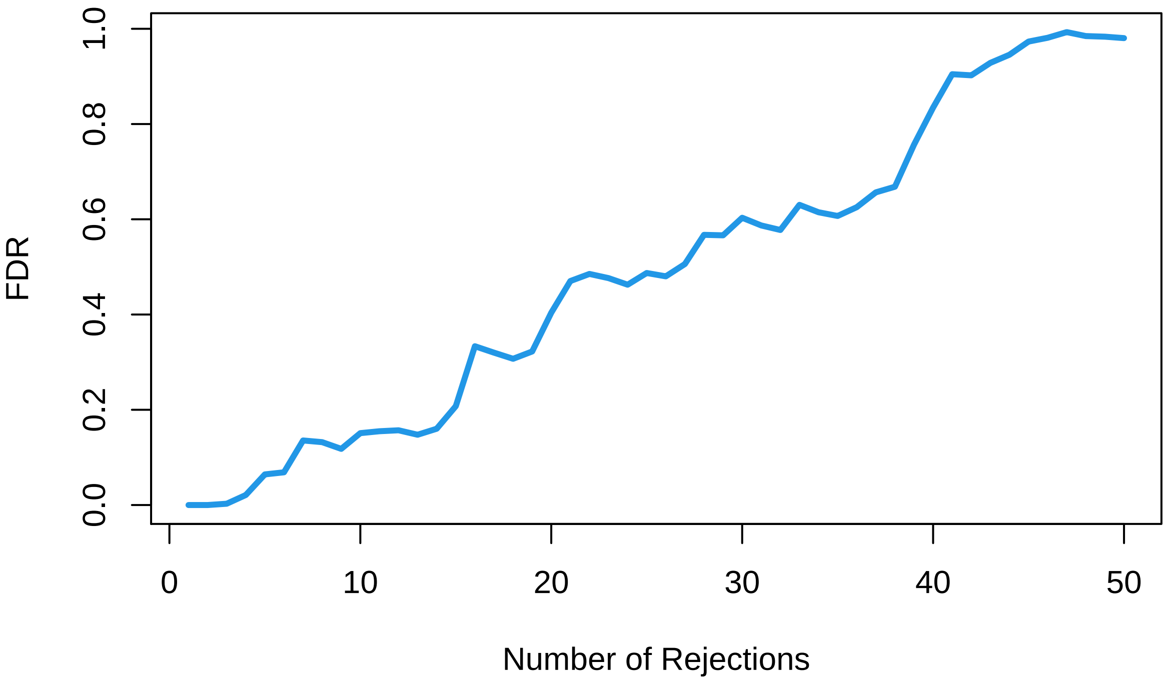
```
cs = sort(abs(Ts))
FDRs = Rs = Vs = rep(NA,m)
for(j in 1:m){
  Vs[j] = sum(abs(Ts.star) >= cs[j])/b     # V = false rejections
  Rs[j] = sum(abs(Ts) >= cs[j])            # R = total rejected nulls.. 50:1
  FDRs[j] = Vs[j]/Rs[j]
}

max(Rs[FDRs <= 0.1]) #6 out of 50 nulls can be rejected.. expect ~1 false positive
```

```
## [1] 6
```

```
max(Rs[FDRs <= 0.3]) #15 out of 50 nulls can be rejected.. expect ~2 false positive
```

```
## [1] 15
```

```
plot(Rs, FDRs, xlab="Number of Rejections", type="l", ylab="FDR", col=4, lwd=3)
```


FDR
Number of Rejections