

Lab2 Report

Ninh (Vincent) DO

March 16, 2019

Data Example

```
set.seed(1)
population = unlist( read.csv("femaleControlsPopulation.csv") )
alpha <- 0.05
N <- 12
m <- 10000
pvals <- replicate(m,{
  control = sample(population,N)
  treatment = sample(population,N)
  t.test(treatment,control)$p.value
})
```

```
sum(pvals < 0.05)
```

```
## [1] 462
```

```
alpha <- 0.05
N <- 12
m <- 10000
p0 <- 0.90 ##10% of diets work, 90% don't
m0 <- m*p0
m1 <- m-m0
nullHypothesis <- c( rep(TRUE,m0), rep(FALSE,m1))
delta <- 3
```

```
set.seed(1)
calls <- sapply(1:m, function(i){
  control <- sample(population,N)
  treatment <- sample(population,N)
  if(!nullHypothesis[i]) treatment <- treatment + delta
  ifelse( t.test(treatment,control)$p.value < alpha,
    "Called Significant",
    "Not Called Significant")
})
```

```
null_hypothesis <- factor( nullHypothesis, levels=c("TRUE","FALSE"))
table(null_hypothesis,calls)
```

```
##              calls
## null_hypothesis Called Significant Not Called Significant
##          TRUE              421              8579
##          FALSE             520              480
```

```
B <- 10 ##number of simulations
VandS <- replicate(B,{
  calls <- sapply(1:m, function(i){
    control <- sample(population,N)
```

```

treatment <- sample(population,N)
if(!nullHypothesis[i]) treatment <- treatment + delta
t.test(treatment,control)$p.val < alpha
})
cat("V =",sum(nullHypothesis & calls), "S =",sum(!nullHypothesis & calls),"\n")
c(sum(nullHypothesis & calls),sum(!nullHypothesis & calls))
})

```

```

## V = 410 S = 564
## V = 400 S = 552
## V = 366 S = 546
## V = 382 S = 553
## V = 372 S = 505
## V = 382 S = 530
## V = 381 S = 539
## V = 396 S = 554
## V = 380 S = 550
## V = 405 S = 569

```

The Bonferroni Correction

```

B<-10000
minpval <- replicate(B, min(runif(10000,0,1))<0.01)
mean(minpval>=1)

```

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

```

set.seed(1)
pvals <- sapply(1:m, function(i){
  control <- sample(population,N)
  treatment <- sample(population,N)
  if(!nullHypothesis[i]) treatment <- treatment + delta
  t.test(treatment,control)$p.value
})

```

```
sum(pvals < 0.05/10000)
```

```
## [1] 2
```

False Discovery Rate

```

set.seed(1)
pvals <- sapply(1:m, function(i){
  control <- sample(population,6)
  treatment <- sample(population,6)
  if(!nullHypothesis[i]) treatment <- treatment + delta
  t.test(treatment,control)$p.value
})
sum(pvals < 0.05/10000)

```

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

Vectorizing Code

```
library(genefilter) ##rowttests is here
set.seed(1)
##Define groups to be used with rowttests
g <- factor( c(rep(0,N),rep(1,N)) )
B <- 1000 ##number of simulations
Qs <- replicate(B,{
  ##matrix with control data (rows are tests, columns are mice)
  controls <- matrix(sample(population, N*m, replace=TRUE),nrow=m)

  ##matrix with control data (rows are tests, columns are mice)
  treatments <- matrix(sample(population, N*m, replace=TRUE),nrow=m)

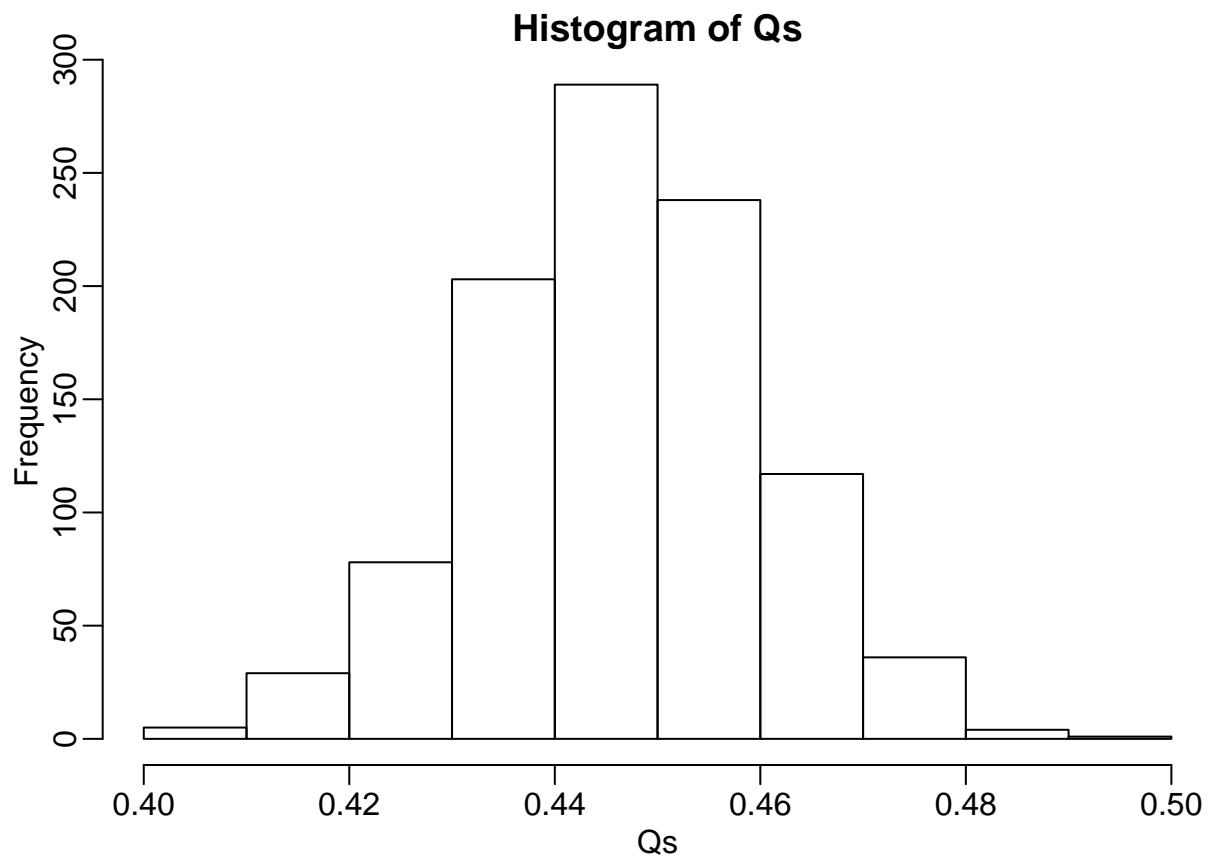
  ##add effect to 10% of them
  treatments[which(!nullHypothesis),]<-treatments[which(!nullHypothesis),]+delta

  ##combine to form one matrix
  dat <- cbind(controls,treatments)

  calls <- rowttests(dat,g)$p.value < alpha
  R=sum(calls)
  Q=ifelse(R>0,sum(nullHypothesis & calls)/R,0)
  return(Q)
})
```

Controlling FDR

```
library(rafalib)
mypar(1,1)
hist(Qs) ##Q is a random variable, this is its distribution
```



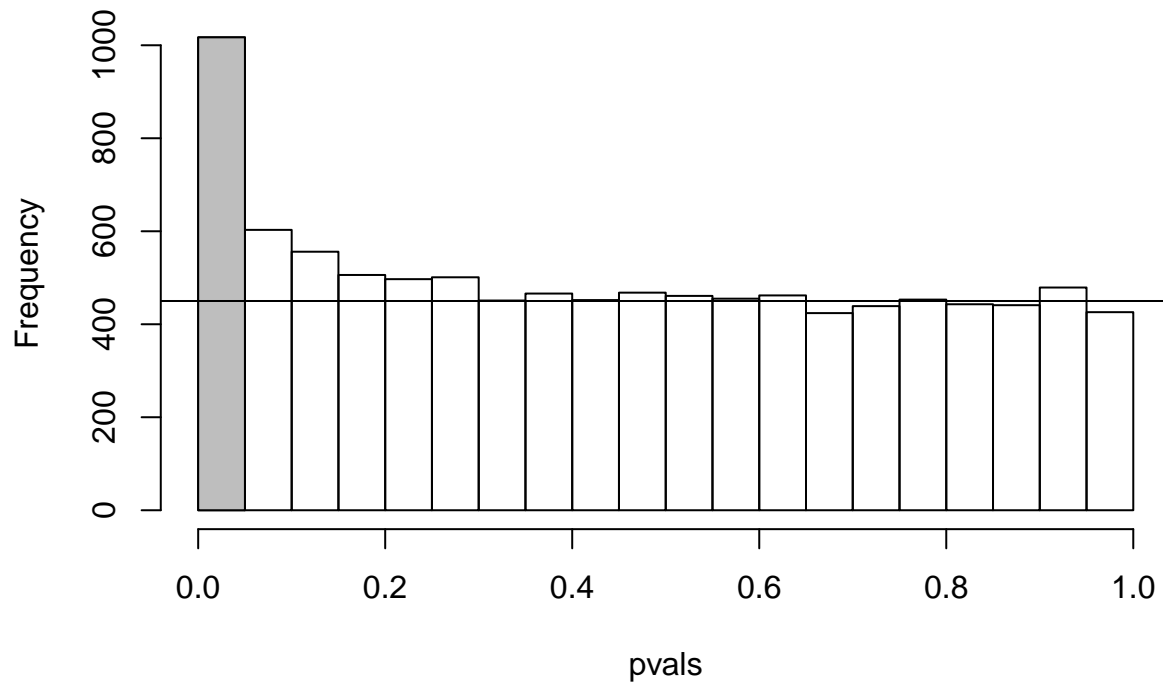
```
FDR=mean(Qs)
print(FDR)
```

```
## [1] 0.4463354
```

```
set.seed(1)
controls <- matrix(sample(population, N*m, replace=TRUE),nrow=m)
treatments <- matrix(sample(population, N*m, replace=TRUE),nrow=m)
treatments[which(!nullHypothesis),]<-treatments[which(!nullHypothesis),]+delta
dat <- cbind(controls,treatments)
pvals <- rowttests(dat,g)$p.value

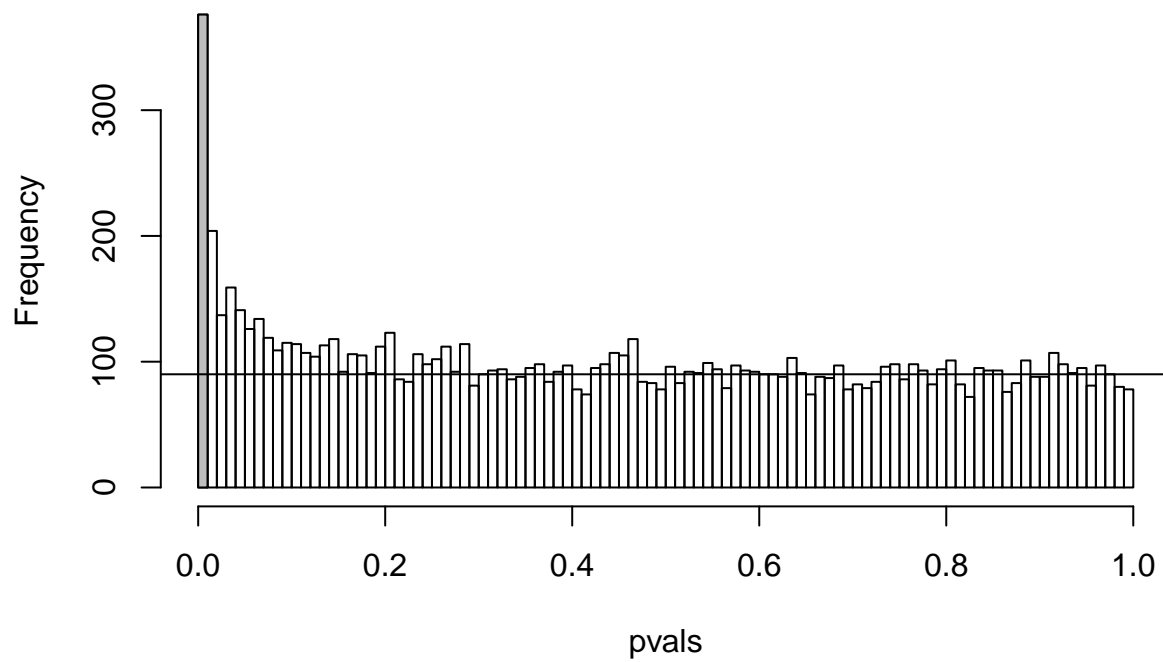
h <- hist(pvals,breaks=seq(0,1,0.05))
polygon(c(0,0.05,0.05,0),c(0,0,h$counts[1],h$counts[1]),col="grey")
abline(h=m0/20)
```

Histogram of pvals



```
h <- hist(pvals,breaks=seq(0,1,0.01))  
polygon(c(0,0.01,0.01,0),c(0,0,h$counts[1],h$counts[1]),col="grey")  
abline(h=m0/100)
```

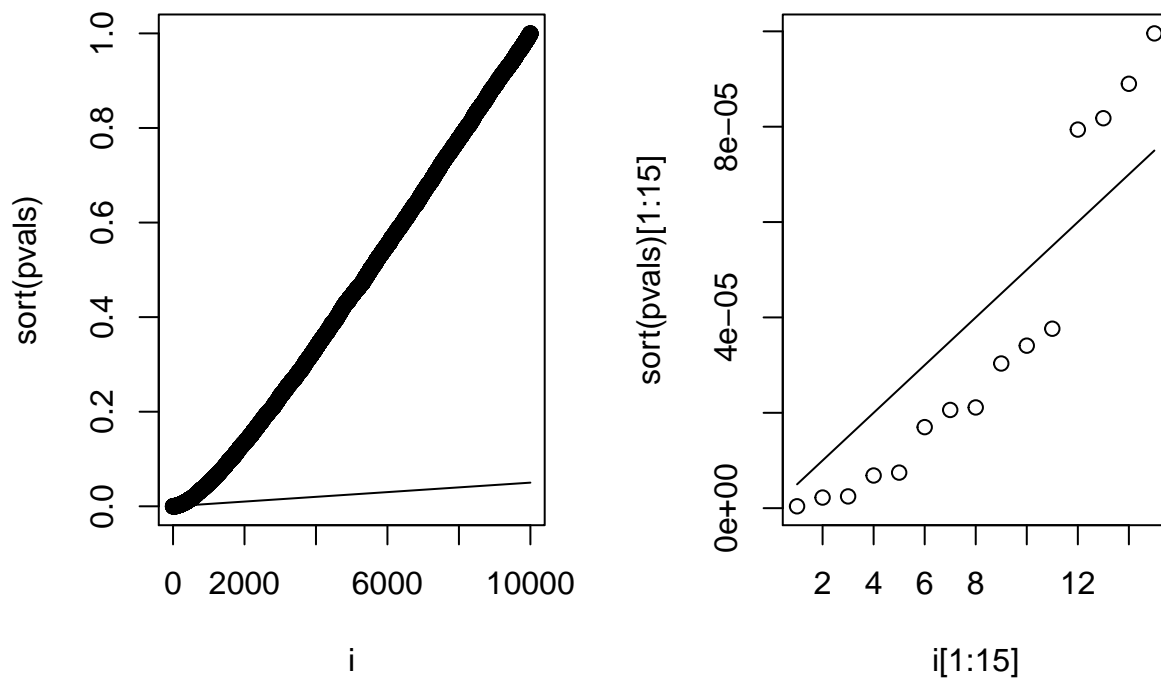
Histogram of pvals



Benjamini-Hochberg

```
par(mfrow=c(1,2))
i = c(1:m)
plot(i, sort(pvals))
lines(i[c(1, m)], i[c(1, m)] * alpha / m)
plot(i[1:15], sort(pvals)[1:15], main="Close-up")
lines(i[c(1, 15)], i[c(1, 15)] * alpha / m)
```

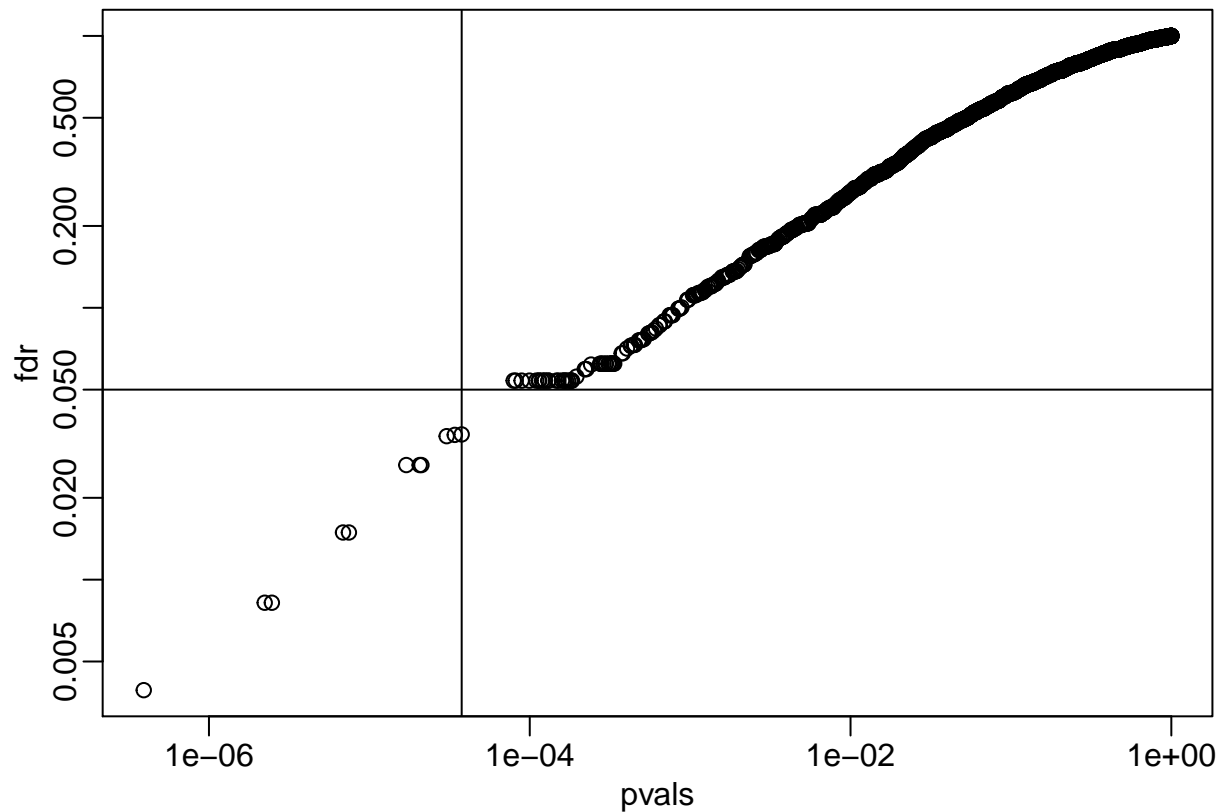
Close-up



```
k <- max( which( sort(pvals) < i/m*alpha) )
cutoff <- sort(pvals)[k]
cat("k =",k,"p-value cutoff=",cutoff)
```

```
## k = 11 p-value cutoff= 3.763357e-05
```

```
fdr <- p.adjust(pvals, method="fdr")
mypar(1,1)
plot(pvals,fdr,log="xy")
abline(h=alpha,v=cutoff) ##cutoff was computed above
```



```

set.seed(1)
Qs_controlled_FDR <- replicate(B,{
  ##matrix with control data (rows are tests, columns are mice)
  controls <- matrix(sample(population, N*m, replace=TRUE),nrow=m)

  ##matrix with control data (rows are tests, columns are mice)
  treatments <- matrix(sample(population, N*m, replace=TRUE),nrow=m)

  ##add effect to 10% of them
  treatments[which(!nullHypothesis),]<-treatments[which(!nullHypothesis),]+delta

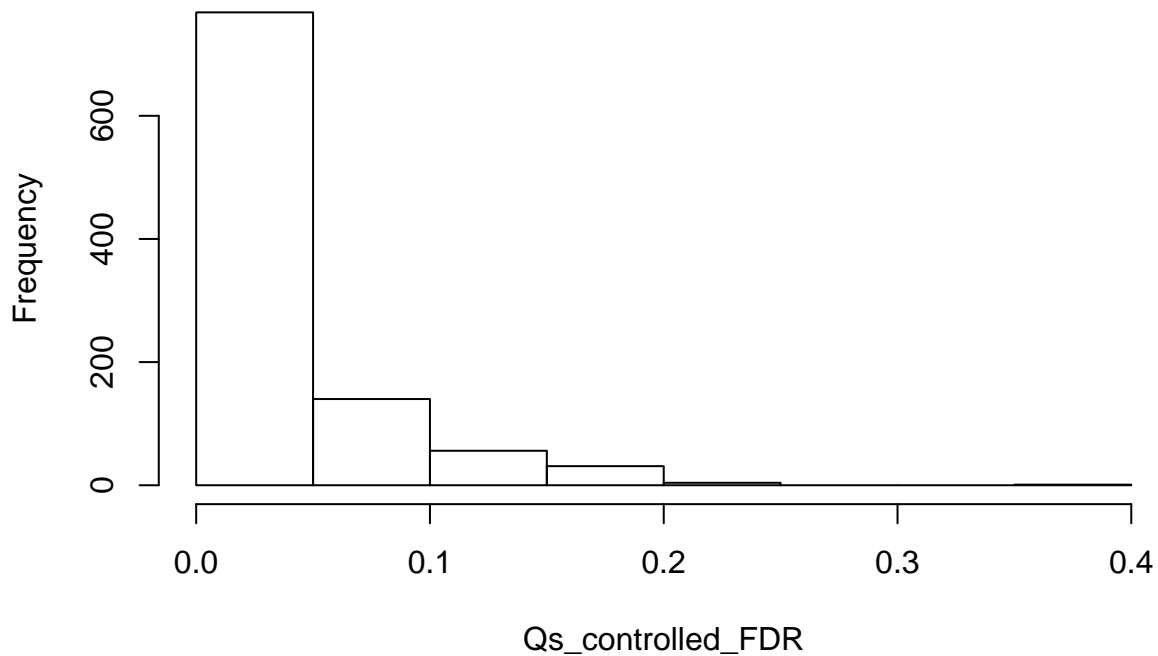
  ##combine to form one matrix
  dat <- cbind(controls,treatments)

  calls <- rowttests(dat,g)$p.value <= cutoff
  R=sum(calls)
  Q=ifelse(R>0,sum(nullHypothesis & calls)/R,0)
  return(Q)
})

hist(Qs_controlled_FDR)

```

Histogram of Qs_controlled_FDR



```
FDR_controlled=mean(Qs_controlled_FDR)
print(FDR_controlled)
```

```
## [1] 0.02512187
```

```
set.seed(1)
no_discovery_rates <- replicate(B,{
  ##matrix with control data (rows are tests, columns are mice)
  controls <- matrix(sample(population, N*m, replace=TRUE),nrow=m)

  ##matrix with control data (rows are tests, columns are mice)
  treatments <- matrix(sample(population, N*m, replace=TRUE),nrow=m)

  ##add effect to 10% of them
  treatments[which(!nullHypothesis),]<-treatments[which(!nullHypothesis),]+delta

  ##combine to form one matrix
  dat <- cbind(controls,treatments)

  no_calls <- rowttests(dat,g)$p.value > cutoff
  m_R=sum(no_calls)
  no_discovery = m_R / length(no_calls)
  return(no_discovery)
})
```

```
no_discovery_mean = mean(no_discovery_rates)
print(no_discovery_mean)
```

```
## [1] 0.9989413
```