Project 3: Overfitting

Keenan Elliott

2023-12-16

Introduction

The Family Wise Error Rate (FWER) is the probability of making at least one false positive discovery while performing multiple statistical tests [1]. If uncorrected, the increased type 1 error (T1E) can lead to dozens or hundreds of false positive findings in a single study. This problem has been extensively discussed and exemplified in Genome-Wide Association Studies, where thousands to millions of tests can be performed to identify significantly correlated single nucleotide polymorphisms (SNP) and phenotypic traits. However, the issue of T1E inflation can be much more subtle, for example, in employing the minimum p-value approach. The minimum p-value approach involves a researcher selecting a test that produces the lowest possible p-value from an array of possible tests and subsequently only reports this smaller p-value. This methodology can increase the rate of false positives and be considered a form of 'p-hacking' [2]. Here, we will investigate the T1E of the minimum p-value approach and attempt to identify a significance threshold that can compensate for the increase in false positives. Notably, the minimum p-value approach can be a valid statistical procedure, although the significance level must be adjusted to account for the rise in T1E.

In this report, we investigate the influence of the minimum p-value approach using simulated trait and SNP data. To this end, we randomly generated 1000 individuals' genotype data using the Hardy Weinberg equilibrium with a minor allele frequency of 0.2 and randomly sampled phenotype data for a continuous trait, Y, from a normal distribution with a mean of 170 and variance of 7^2 (Y~N(170,7^2)). The randomly generated genotype-phenotype pairs were tested for statistical significance using four coding models: Dominant, Recessive, Additive, and Genotypic. Dominant coding refers to instances where one copy of a SNP leads to expression of a given phenotype; recessive coding refers to models where two copies of a SNP lead to a given phenotype; additive coding describes a model where a trait becomes more prominent as more copies of the SNP are present (i.e. two copies of the SNP produces a more pronounced effect); and genotypic coding provides a model where the individual copies of the SNP are considered in the model using dummy variables. These 1000 simulated individuals are reproduced for 10,000 replications and the data are plotted to observe the effects on the T1E.

Methods

Generating the data

All data were generated randomly in R (V. 4.3.1) accessed using Rstudio. Phenotype data was generated using the rnorm function with the mean of 170 and standard deviation of 7. Genotype data was generated with the sample function with replace=T, and sampling probabilities equal to Hardy Weinberg equilibrium proportions with the minor allele frequency equal to 0.2.

Coding SNP Data

Genotype data was coded as either, dominant, recessive, additive, or genotypic (described above). Coding Keys are described above each section of the for loop in the results section below.

Data visualizations

Visualizations were created in R, using the ggplot2 and qqman libraries.

Results

Running the simulation

Below is the code for the simulation study. The nRep variable assigns the 10,000 repitiions to be completed, the n variable is the number of individuals sampled, mean and standard deviation of the trait are asssigned with meanTrait and SdTrait variables, respectively. Finally, the minor allele frequency for generating HWE proportioned SNPs is assigned by maf. The setseed function was used for replicability and the seed was assigned each loop to the loop iteration value.

Broadly, the steps are as follows:

1. Create a table to store the 1000 phenotypes and genotypic models data.
2. Enter into a loop of 10,000 repetitions.

Steps 3-8 occur within the loop.

3. Generate 1000 phenotypes randomly from the same normal distribution.
4. Generate 1000 genotypes using HWE and the designated coding style.
5. Match up genotypes and phenotypes.
6. Perform test for statistical significance with each of the 4 coding methods.
7. Select the lowest p-value from the four tests.
8. Continue to the next loop with new simulated data until 10,000 iterations are complete.

```
nRep=10000
n=1000
meanTrait=170
SdTrait=7
maf=0.2
p=1-maf
q=maf

GenoPhenoTable=array(data=NA, dim=c(n,6))
colnames(GenoPhenoTable)=c("phenotypes","genotypesADD","genotypesDOM",
                           "genotypesREC","genotypesGENOsnp1", "genotypesGENOsnp2")
GenoPhenoTable=data.frame(GenoPhenoTable)


pVals=array(data=NA,dim=c(nRep,7))
colnames(pVals)=c("Iter","genotypesADDpValue", "genotypesDOMpValue",
                  "genotypesRECpValue", "genotypesGENOpValue", "MinPval", "NegLogMinP")
pVals=data.frame(pVals)

for( i in 1:nRep){
  set.seed(i)

  ### generate a placeholder for the lowest pValue
  minP=1

  ### Generate Phenotypes ###
```

```
phenotype=rnorm(n,meanTrait,SdTrait)
GenoPhenoTable$phenotypes=phenotype

### Generate Genotypes ###
genotypes=sample(c("AA","Aa","aa"), size=n, replace=T, prob=c(p^2, 2*p*q,q^2))

############################
### Dominant SNP Coding ###
############################

## 0=PP, 1=heterozygous Pq, or qq
genotypeDOM=ifelse(genotypes=="AA",0,1)
GenoPhenoTable$genotypesDOM=as.factor(genotypeDOM)

model=lm(formula = phenotypes~genotypesDOM, data=GenoPhenoTable)
summary=summary(model)
lmPvalDOM=summary$coefficients[,4][2]

pVals$genotypesDOMpValue[i]=lmPvalDOM

if (minP>lmPvalDOM){ minP=lmPvalDOM}

############################
### Recessive SNP Coding ###
############################

## 0=PP or heterozygous Pq, 1=qq
genotypeREC=ifelse(genotypes=="aa",1,0)
GenoPhenoTable$genotypesREC=as.factor(genotypeREC)

model=lm(formula = phenotypes~genotypeREC, data=GenoPhenoTable)
summary=summary(model)
lmPvalREC=summary$coefficients[,4][2]

pVals$genotypesRECpValue[i]=lmPvalREC

if (minP>lmPvalREC){ minP=lmPvalREC}


############################
### Additive SNP Coding ###
############################

## 0=PP, 1=heterozygous Pq, 2=qq
genotypeADD=array(data=NA, dim=n)

for(k in 1:n){
  if(genotypes[k]=="aa"){
    genotypeADD[k]=2
  }
  if(genotypes[k]=="Aa"){
    genotypeADD[k]=1
  }
```

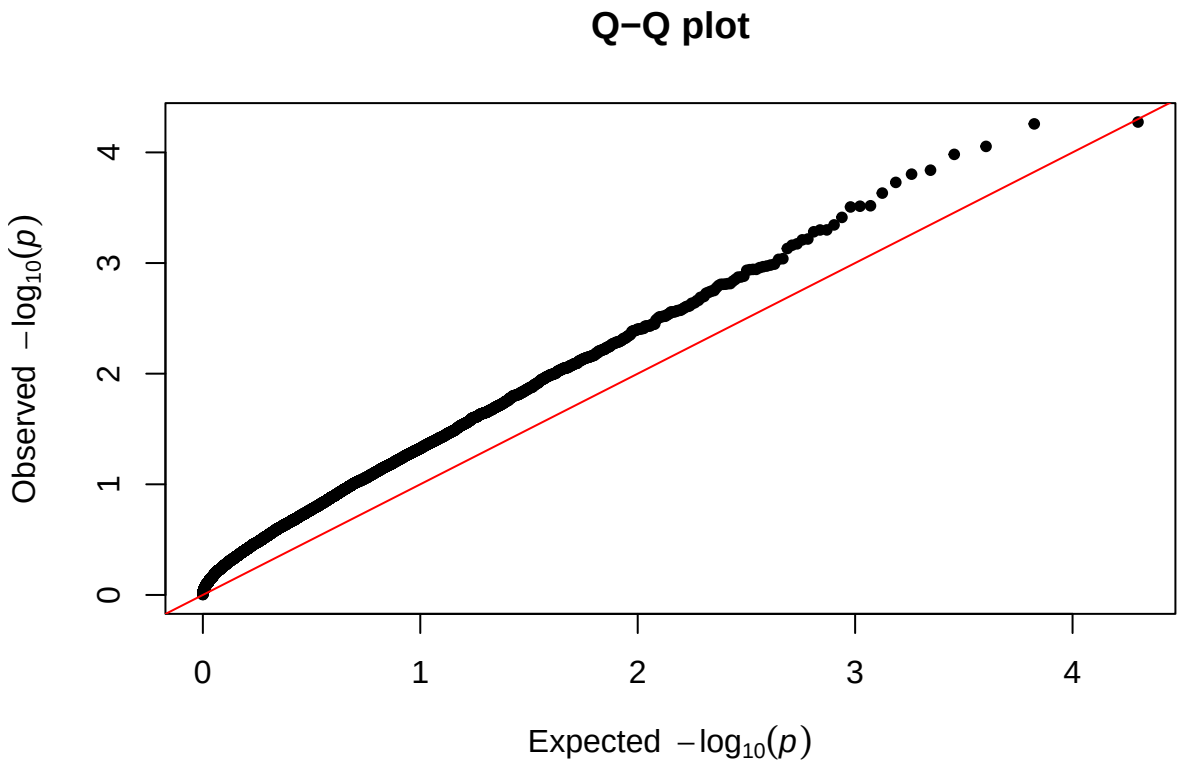
```
    if(genotypes[k]=="AA"){
      genotypeADD[k]=0
    }
  }
  GenoPhenoTable$genotypesADD=as.factor(genotypeADD)


  model=lm(formula=phenotypes~genotypeADD, data=GenoPhenoTable)
  summary=summary(model)
  lmPvalADD=summary$coefficients[,4][2]

  pVals$genotypesADDpValue[i]=lmPvalADD

  if (minP>lmPvalADD){ minP=lmPvalADD}

  ############################
  ### Genotypic SNP Coding ###
  ############################

  ## coding key:
  ## 0 SNP1copy and 0 SNP2copy=homozygous, no snp (PP)
  ## 1 SNP1copy and 0 SNP2copy=heterozygous with 1 copy SNP (pq)
  ## 0 SNP1copy and 1 SNP2copy=homozygous SNP, 2 copies SNP (qq)

  ## 0=PP, 1=heterozygous Pq, 2=qq
  snp1=ifelse(genotypes=="Aa",1,0)
  snp2=ifelse(genotypes=="aa",1,0)

  GenoPhenoTable$genotypesGENOsnp1=snp1
  GenoPhenoTable$genotypesGENOsnp2=snp2

  model=lm(phenotypes~genotypesGENOsnp1+genotypesGENOsnp2, data = GenoPhenoTable)
  modelSum=summary(model)
  fstat=modelSum$fstatistic
  pValGeno=pf(fstat[1], fstat[2],fstat[3], lower.tail = FALSE)

  pVals$genotypesGENOpValue[i]=pValGeno
  if (minP>pValGeno){ minP=pValGeno}


  ## ADD THE LOWEST P VALUE OF ALL TESTS
  pVals$MinPval[i]=minP
  pVals$NegLogMinP[i]=log10(minP)*-1
  pVals$Iter[i]=i

}
```

Visualizing the Results

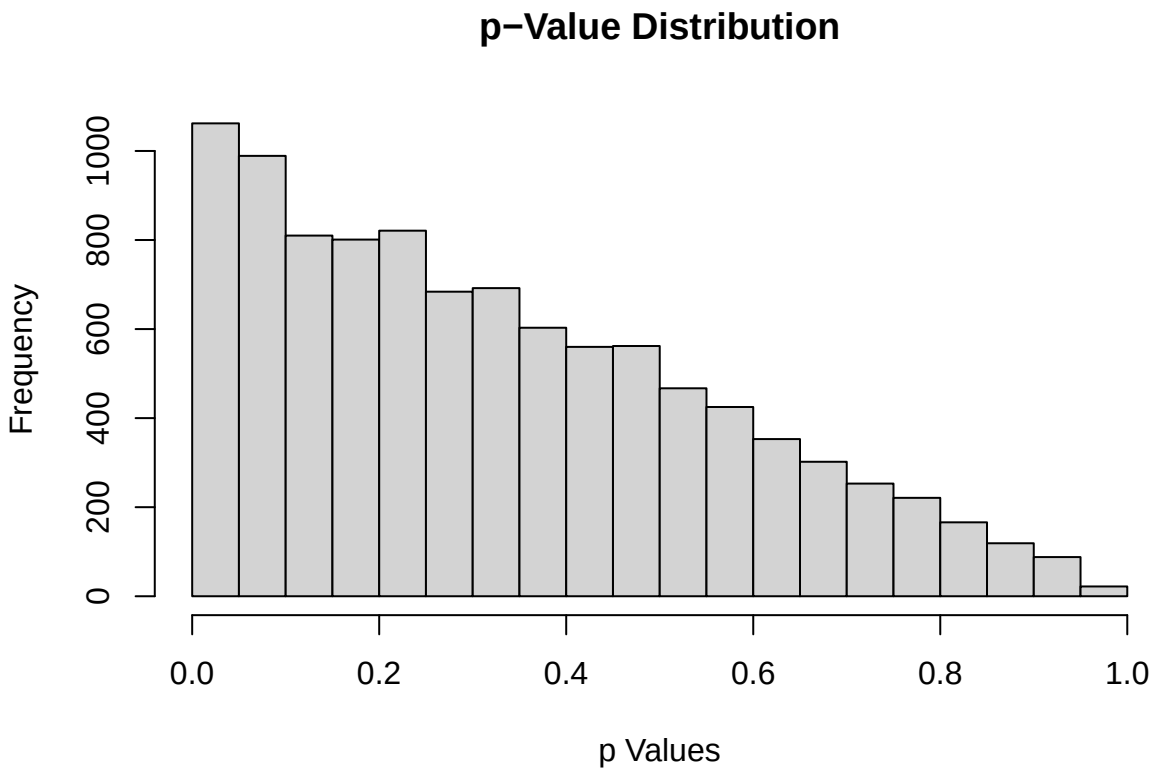
The QQ plot, hisogram of p-values, and manhattan plot for the 10,000 replications can be viewed below. On the manhattan plot, three horizontal lines have been added. The green line corresponds to a -log10(p-value) of 0.05 and the blue line corresponds to a bonferroni adjusted p-value of -log(significance level/number of tests); because we are performing 4 tests, the alpha level of 0.05 was adjusted by dividing alpha by 4. Finally

a red line was added that corresponded to the significance that was required to control the T1E rate at 5% for this simulation study. This value was obtained by sorting all minimum p-values from the 10,000 repetitions and obtaining the value that represents the 95th percentile. This value corresponded to a p-value of ~0.022.

```
library(ggplot2)
suppressMessages(library(qqman))

qq(pVals$MinPval,
main = "Q-Q plot")
```




```
hist(pVals$MinPval, xlab = "p Values", main = "p-Value Distribution")
```



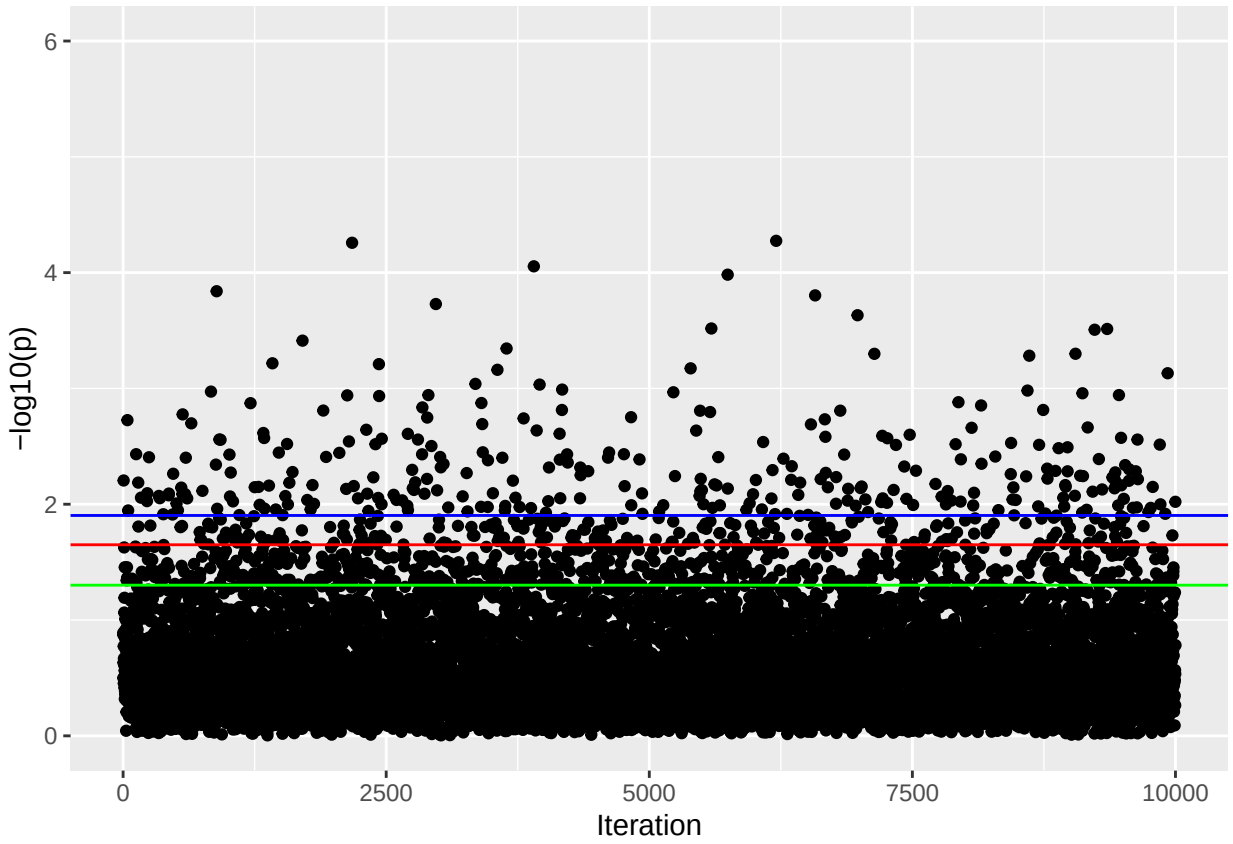

```
# ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+ xlab("Iteration") + ylab("-log10(p)")+ ylim(0

pvalsVector=pVals$MinPval
pvalsVector=sort(pvalsVector)


fivepercent=pvalsVector[500]
sprintf("Using the minimum p-value approach, 5percent of tests had p-value<= %f", fivepercent)
```

```
## [1] "Using the minimum p-value approach, 5percent of tests had p-value<= 0.022387"
```

```
T1ER=length(which(pvalsVector<=0.05))/10000
sprintf("The T1ER was: %f", T1ER)
```

```
## [1] "The T1ER was: 0.106200"
```

```
FWERbonfCorrected=length(which(pvalsVector<=0.05/4))/10000

T1Ecorrection=-log10(pvalsVector[500])
bonfCorrected=-log10((0.05/4))

ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+xlab("Iteration") + ylab("-log10(p)")+ ylim(0, 6
```



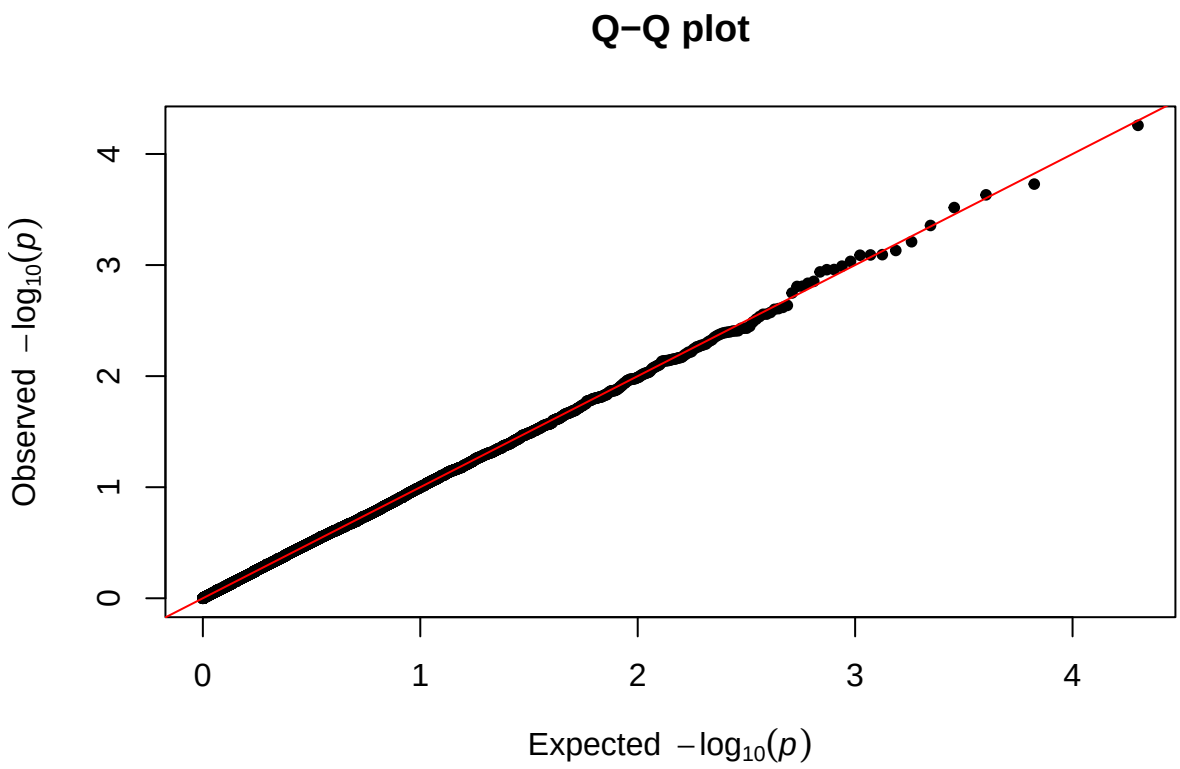
P-value Uniform Under the Null: Observing Effect of Single Test

Below is the same study but performing only one test per repetition. This simulation helps illustrate the p-value is uniformly distributed under the null. Note the green and red lines on the Manhattan plot over overlapping as the expected 5% T1ER overlaps with the result obtained from the simulation. The simulations are completed for each coding type: dominant, recessive, additive and genotypic.

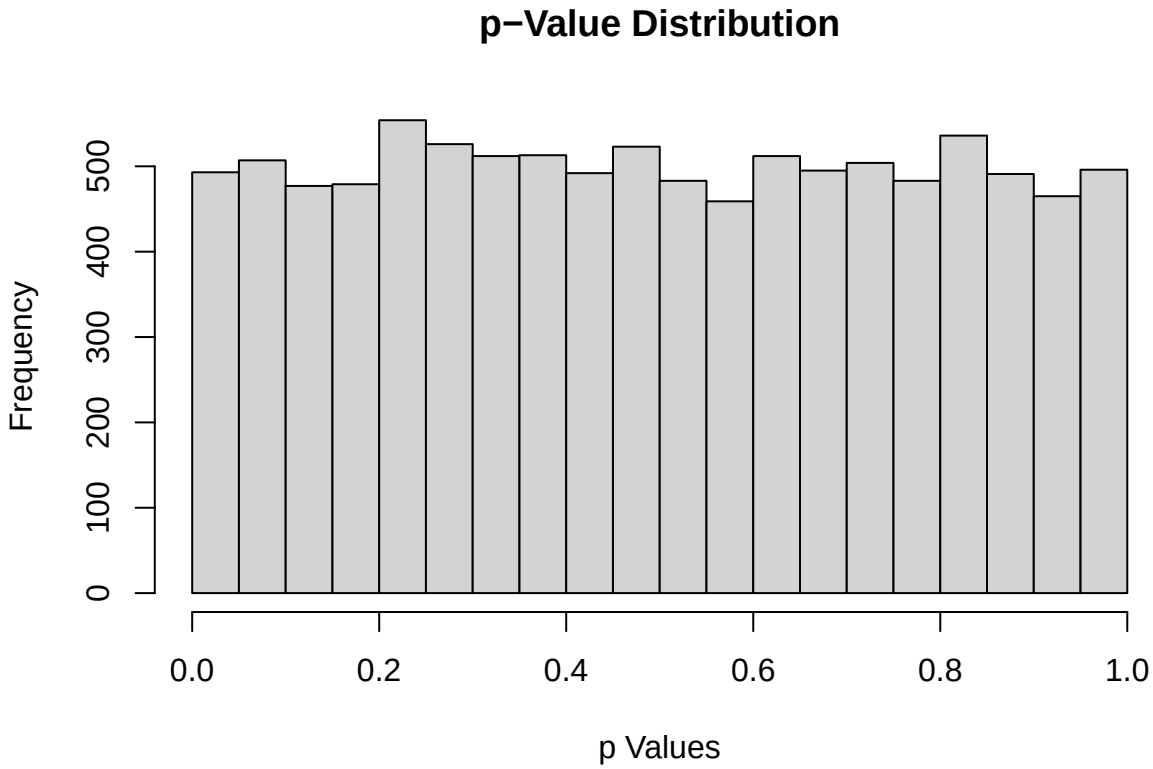
Effect of Single Test: Dominant coding

```
nRep=10000
n=1000
meanTrait=170
SdTrait=7
maf=0.2
p=1-maf
q=maf

GenoPhenoTable=array(data=NA, dim=c(n,6))
colnames(GenoPhenoTable)=c("phenotypes","genotypesADD","genotypesDOM",
                           "genotypesREC","genotypesGENOsnp1", "genotypesGENOsnp2")
GenoPhenoTable=data.frame(GenoPhenoTable)


pVals=array(data=NA,dim=c(nRep,7))
colnames(pVals)=c("Iter","genotypesADDpValue", "genotypesDOMpValue",
                  "genotypesRECpValue", "genotypesGENOpValue", "MinPval", "NegLogMinP")
pVals=data.frame(pVals)
```

```r
for( i in 1:nRep){
  set.seed(i)

  ### generate a placeholder for the lowest pValue
  minP=1

  ### Generate Phenotypes ###
  phenotype=rnorm(n,meanTrait,SdTrait)
  GenoPhenoTable$phenotypes=phenotype

  ### Generate Genotypes ###
  genotypes=sample(c("AA","Aa","aa"), size=n, replace=T, prob=c(p^2, 2*p*q,q^2))

  ############################
  ### Dominant SNP Coding ###
  ############################

  ## 0=PP, 1=heterozygous Pq, or qq
  genotypeDOM=ifelse(genotypes=="AA",0,1)
  GenoPhenoTable$genotypesDOM=as.factor(genotypeDOM)

  model=lm(formula = phenotypes~genotypesDOM, data=GenoPhenoTable)
  summary=summary(model)
  lmPvalDOM=summary$coefficients[,4][2]

  pVals$genotypesDOMpValue[i]=lmPvalDOM

  if (minP>lmPvalDOM){ minP=lmPvalDOM}

  ## ADD THE LOWEST P VALUE OF ALL TESTS
  pVals$MinPval[i]=minP
  pVals$NegLogMinP[i]=log10(minP)*-1
  pVals$Iter[i]=i

}

library(ggplot2)
suppressMessages(library(qqman))

qq(pVals$MinPval,
main = "Q-Q plot")
```




```
hist(pVals$MinPval, xlab = "p Values", main = "p-Value Distribution")
```

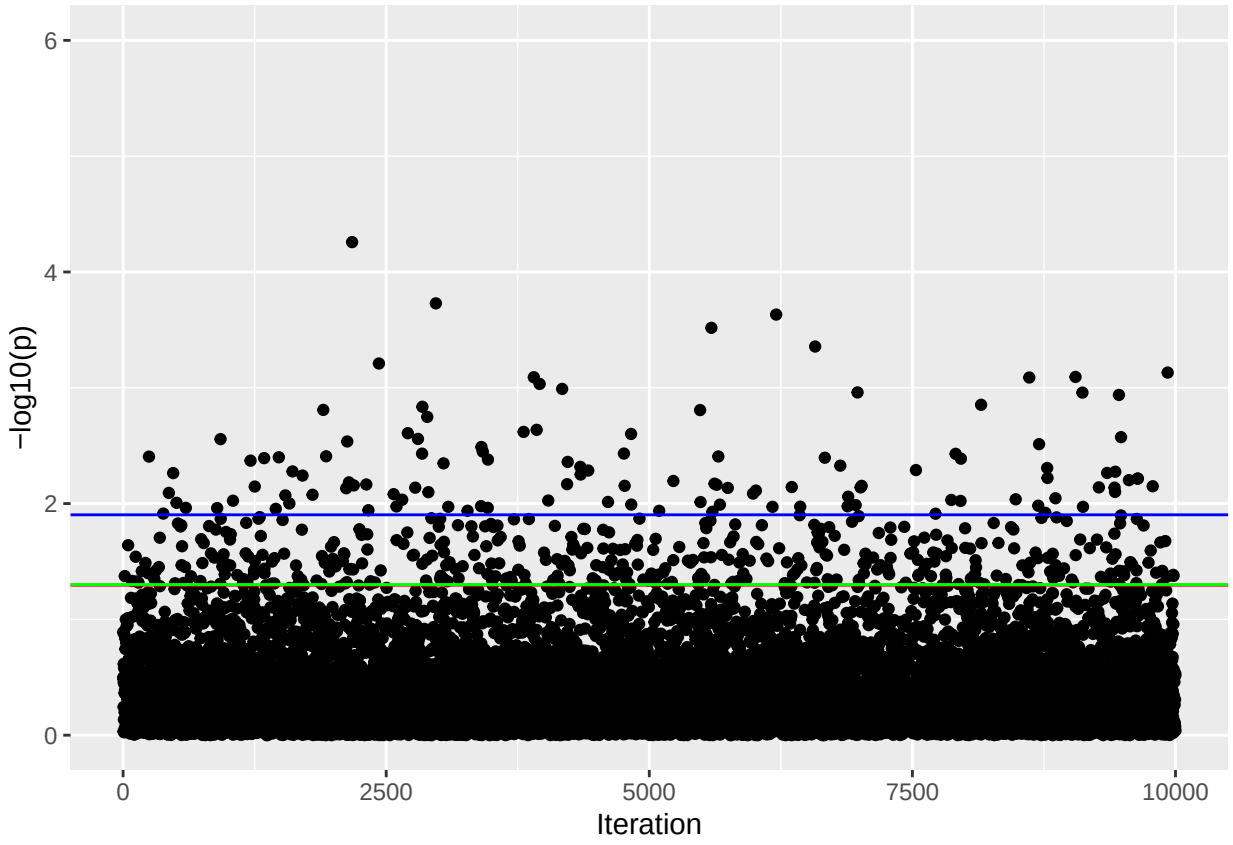



```
# ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+ xlab("Iteration") + ylab("-log10(p)")+ ylim(0

pvalsVector=pVals$MinPval
```

```
pvalsVector=sort(pvalsVector)


T1ER=length(which(pvalsVector<=0.05))/10000
sprintf("The T1ER was: %f", T1ER)
```

```
## [1] "The T1ER was: 0.049300"
```

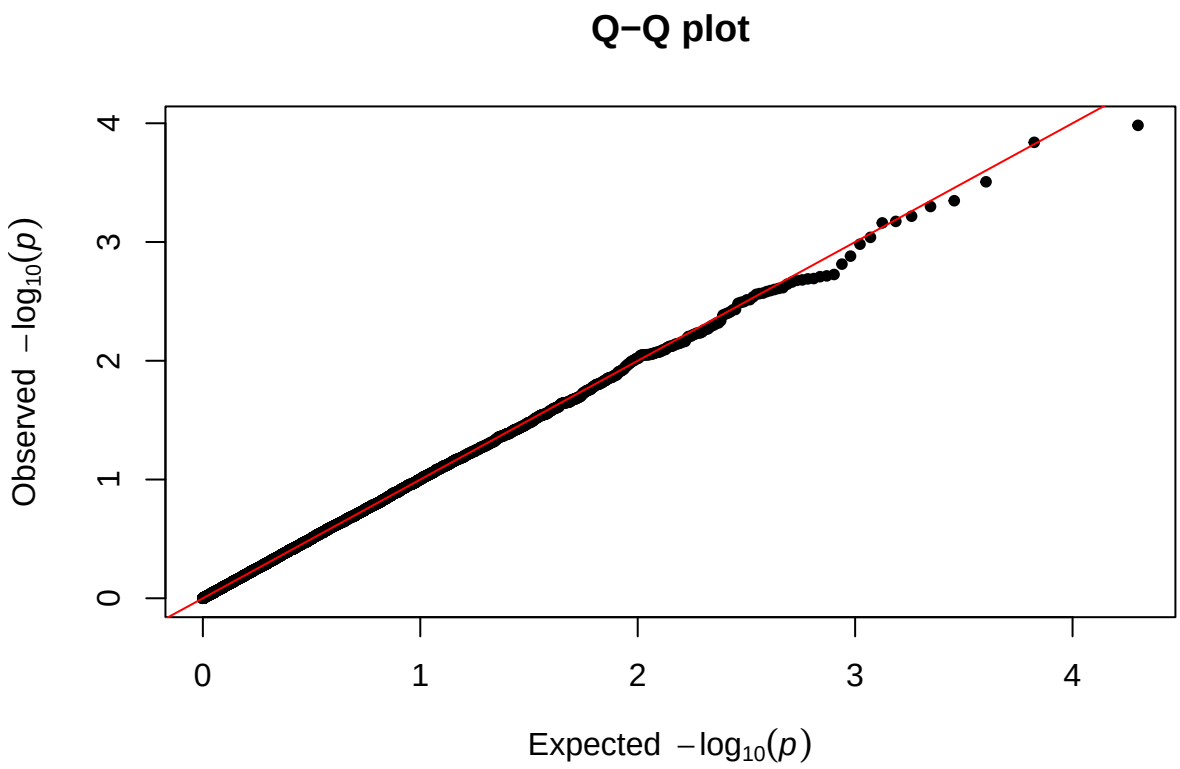
```
FWERbonfCorrected=length(which(pvalsVector<=0.05/4))/10000

T1Ecorrection=-log10(pvalsVector[500])
bonfCorrected=-log10((0.05/4))

ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+xlab("Iteration") + ylab("-log10(p)")+ ylim(0, 6)
```



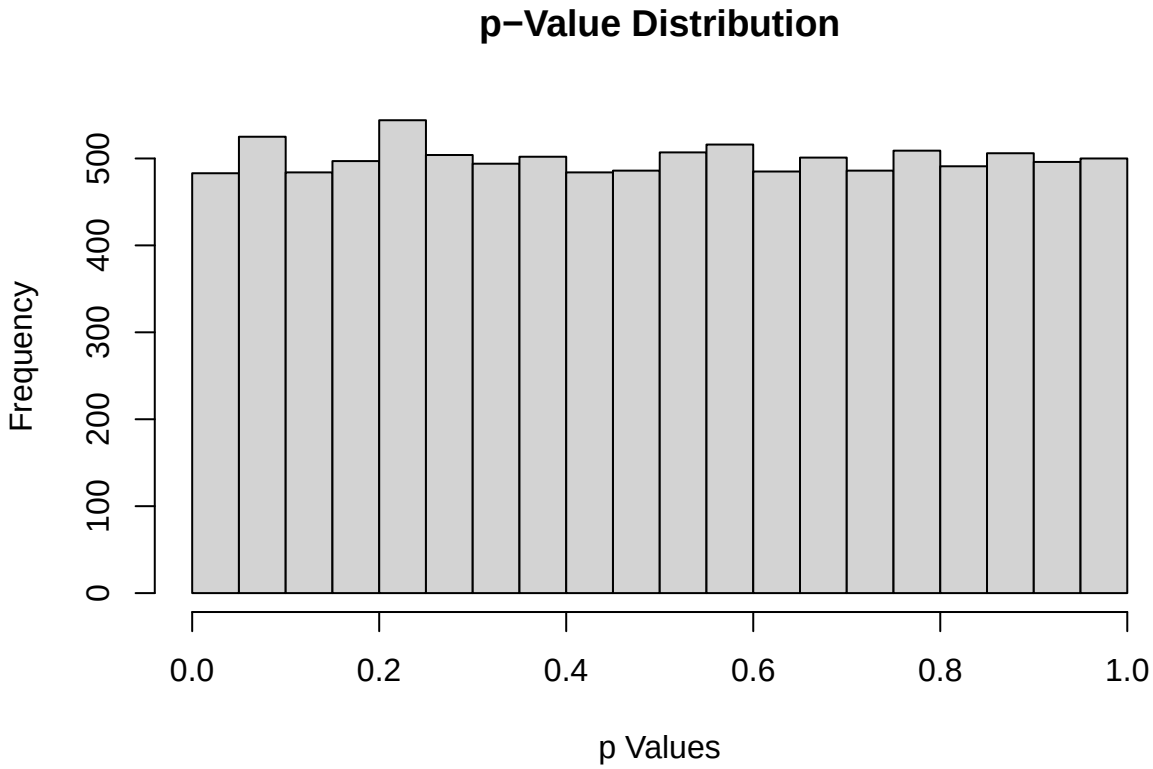
Effect of Single Test: Recessive coding

```
nRep=10000
n=1000
meanTrait=170
SdTrait=7
maf=0.2
p=1-maf
q=maf

GenoPhenoTable=array(data=NA, dim=c(n,6))
colnames(GenoPhenoTable)=c("phenotypes","genotypesADD","genotypesDOM",
```

```
                              "genotypesREC","genotypesGENOsnp1", "genotypesGENOsnp2")
GenoPhenoTable=data.frame(GenoPhenoTable)


pVals=array(data=NA,dim=c(nRep,7))
colnames(pVals)=c("Iter","genotypesADDpValue", "genotypesDOMpValue",
                  "genotypesRECpValue", "genotypesGENOpValue", "MinPval", "NegLogMinP")
pVals=data.frame(pVals)

for( i in 1:nRep){
  set.seed(i)

  ### generate a placeholder for the lowest pValue
  minP=1

  ### Generate Phenotypes ###
  phenotype=rnorm(n,meanTrait,SdTrait)
  GenoPhenoTable$phenotypes=phenotype

  ### Generate Genotypes ###
  genotypes=sample(c("AA","Aa","aa"), size=n, replace=T, prob=c(p^2, 2*p*q,q^2))

  ############################
  ### Recessive SNP Coding ###
  ############################

  ## 0=PP or heterozygous Pq, 1=qq
  genotypeREC=ifelse(genotypes=="aa",1,0)
  GenoPhenoTable$genotypesREC=as.factor(genotypeREC)

  model=lm(formula = phenotypes~genotypeREC, data=GenoPhenoTable)
  summary=summary(model)
  lmPvalREC=summary$coefficients[,4][2]

  pVals$genotypesRECpValue[i]=lmPvalREC

  if (minP>lmPvalREC){ minP=lmPvalREC}

  ## ADD THE LOWEST P VALUE OF ALL TESTS
  pVals$MinPval[i]=minP
  pVals$NegLogMinP[i]=log10(minP)*-1
  pVals$Iter[i]=i

}

library(ggplot2)
suppressMessages(library(qqman))

qq(pVals$MinPval,
main = "Q-Q plot")
```




```
hist(pVals$MinPval, xlab = "p Values", main = "p-Value Distribution")
```

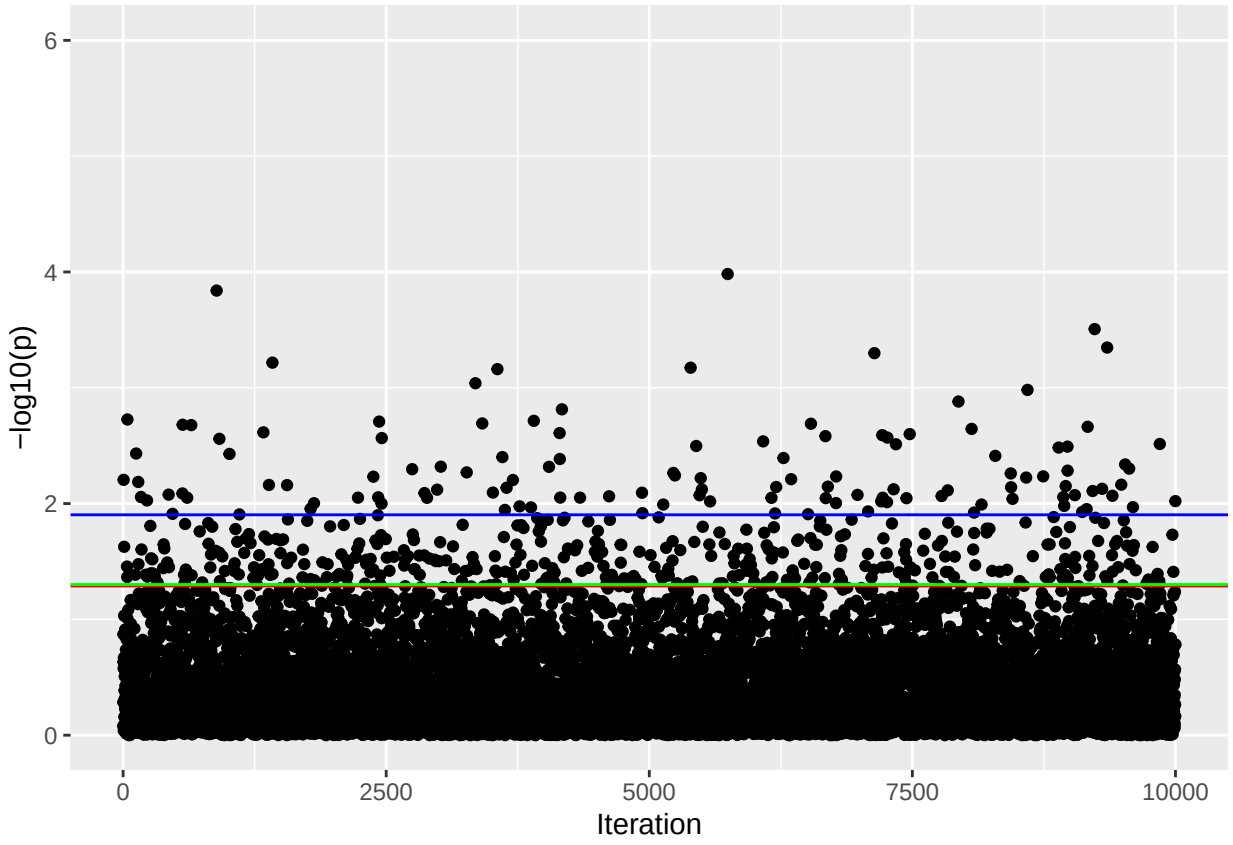



```
# ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+ xlab("Iteration") + ylab("-log10(p)")+ ylim(0

pvalsVector=pVals$MinPval
```

```r
pvalsVector=sort(pvalsVector)


T1ER=length(which(pvalsVector<=0.05))/10000
sprintf("The T1ER was: %f", T1ER)
```

```
## [1] "The T1ER was: 0.048300"
```

```r
FWERbonfCorrected=length(which(pvalsVector<=0.05/4))/10000

T1Ecorrection=-log10(pvalsVector[500])
bonfCorrected=-log10((0.05/4))

ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+xlab("Iteration") + ylab("-log10(p)")+ ylim(0, 6)
```

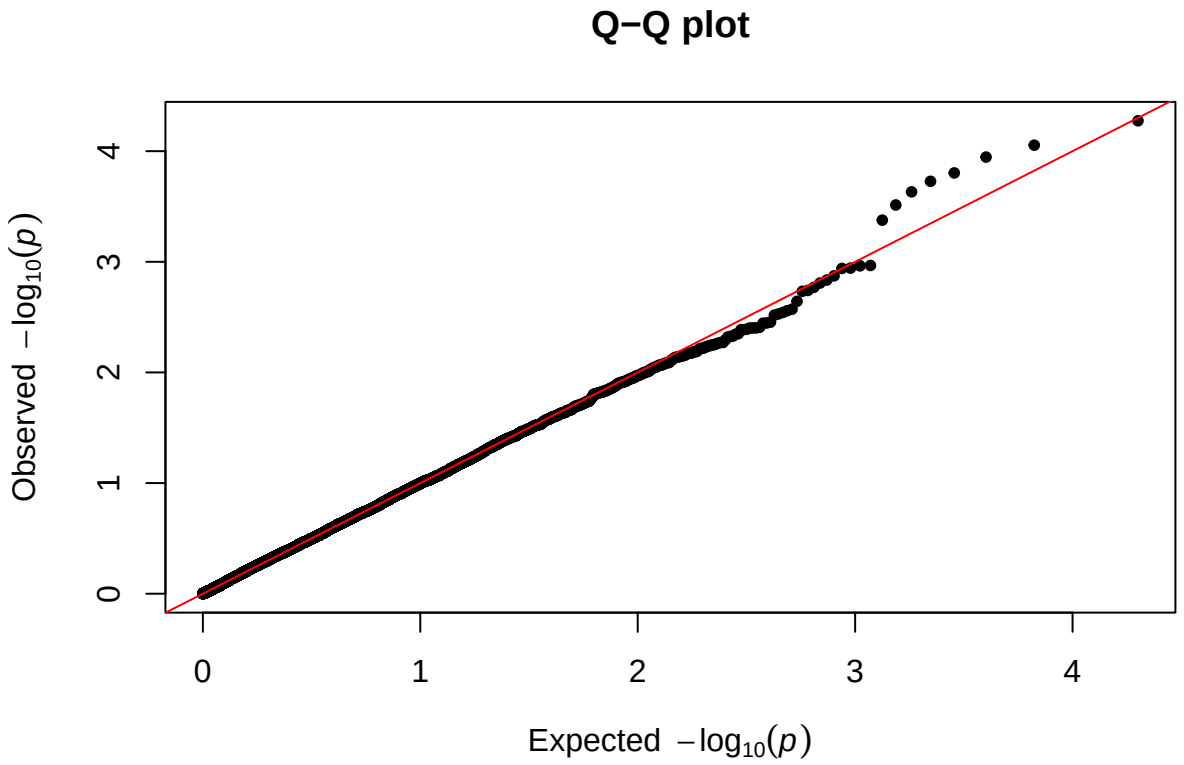


Effect of Single Test: Additive coding

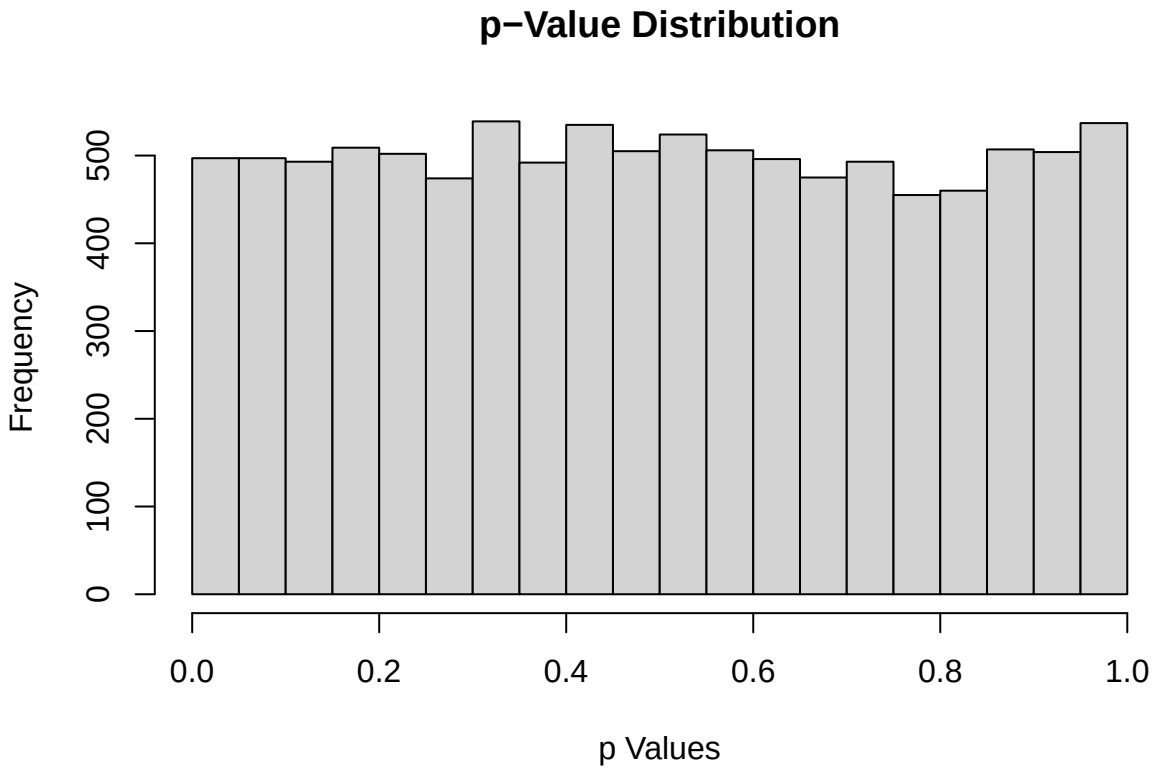
```r
nRep=10000
n=1000
meanTrait=170
SdTrait=7
maf=0.2
p=1-maf
q=maf

GenoPhenoTable=array(data=NA, dim=c(n,6))
colnames(GenoPhenoTable)=c("phenotypes","genotypesADD","genotypesDOM",
```

```
                              "genotypesREC","genotypesGENOsnp1", "genotypesGENOsnp2")
GenoPhenoTable=data.frame(GenoPhenoTable)


pVals=array(data=NA,dim=c(nRep,7))
colnames(pVals)=c("Iter","genotypesADDpValue", "genotypesDOMpValue",
                  "genotypesRECpValue", "genotypesGENOpValue", "MinPval", "NegLogMinP")
pVals=data.frame(pVals)

for( i in 1:nRep){
  set.seed(i)

  ### generate a placeholder for the lowest pValue
  minP=1

  ### Generate Phenotypes ###
  phenotype=rnorm(n,meanTrait,SdTrait)
  GenoPhenoTable$phenotypes=phenotype

  ### Generate Genotypes ###
  genotypes=sample(c("AA","Aa","aa"), size=n, replace=T, prob=c(p^2, 2*p*q,q^2))

  ############################
  ### Additive SNP Coding ###
  ############################

  ## 0=PP, 1=heterozygous Pq, 2=qq
  genotypeADD=array(data=NA, dim=n)

  for(k in 1:n){
    if(genotypes[k]=="aa"){
      genotypeADD[k]=2
    }
    if(genotypes[k]=="Aa"){
      genotypeADD[k]=1
    }
    if(genotypes[k]=="AA"){
      genotypeADD[k]=0
    }
  }
  GenoPhenoTable$genotypesADD=as.factor(genotypeADD)


  model=lm(formula=phenotypes~genotypeADD, data=GenoPhenoTable)
  summary=summary(model)
  lmPvalADD=summary$coefficients[,4][2]

  pVals$genotypesADDpValue[i]=lmPvalADD

  if (minP>lmPvalADD){ minP=lmPvalADD}

  ## ADD THE LOWEST P VALUE OF ALL TESTS
  pVals$MinPval[i]=minP
```

```
  pVals$NegLogMinP[i]=log10(minP)*-1
  pVals$Iter[i]=i

}

library(ggplot2)
suppressMessages(library(qqman))

qq(pVals$MinPval,
main = "Q-Q plot")
```




```
hist(pVals$MinPval, xlab = "p Values", main = "p-Value Distribution")
```

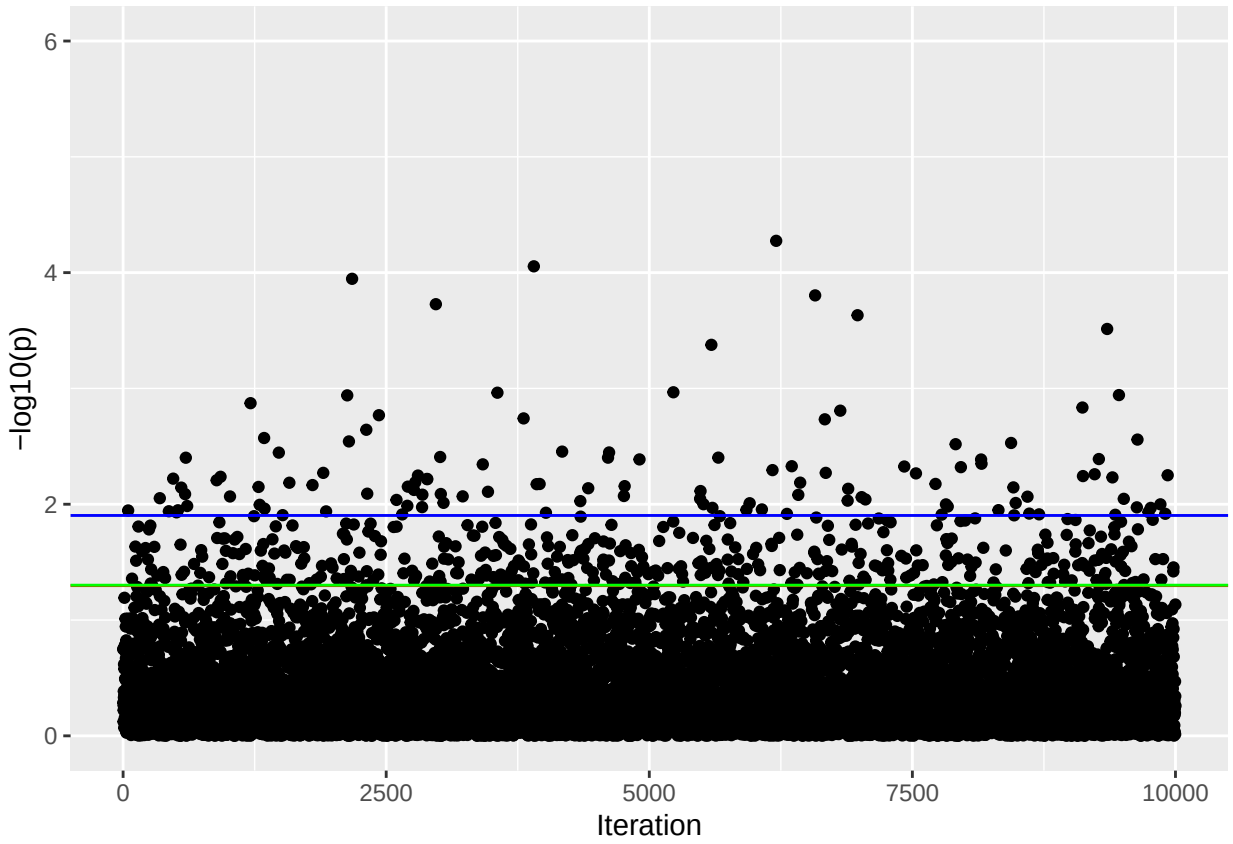



```
# ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+ xlab("Iteration") + ylab("-log10(p)")+ ylim(0
```

```
pvalsVector=pVals$MinPval
pvalsVector=sort(pvalsVector)


T1ER=length(which(pvalsVector<=0.05))/10000
sprintf("The T1ER was: %f", T1ER)
```

```
## [1] "The T1ER was: 0.049700"
```

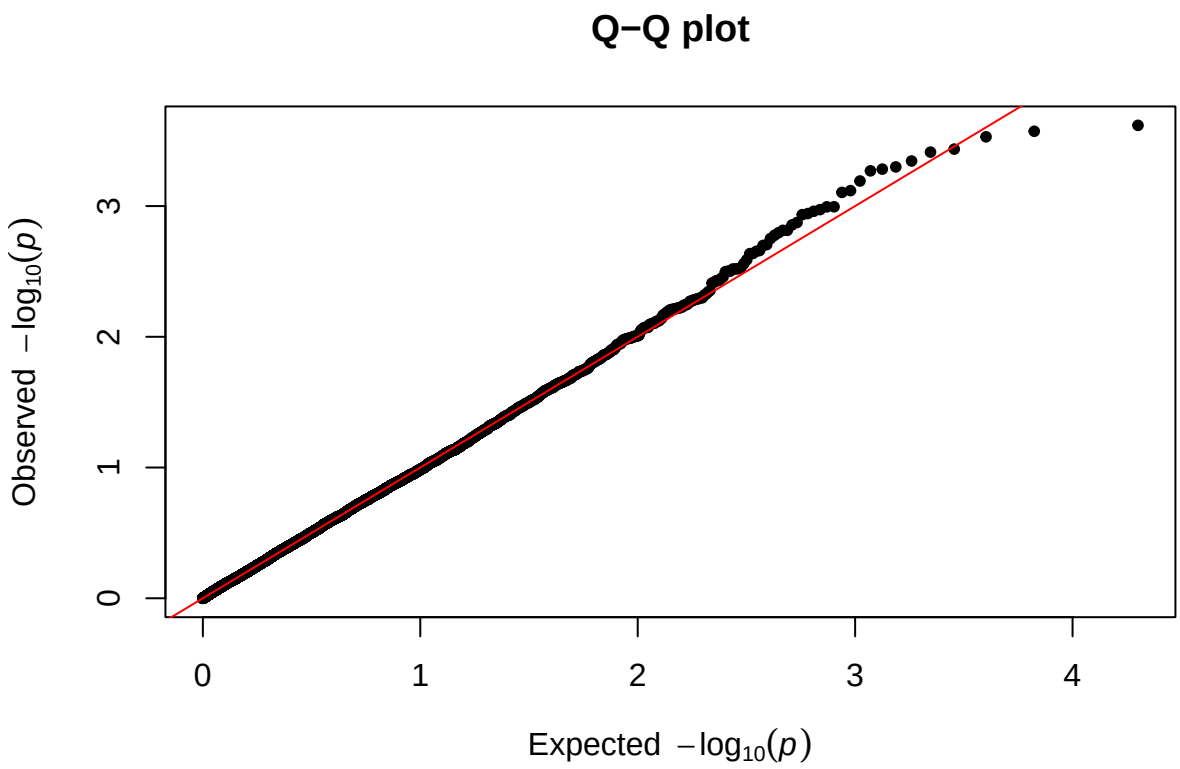
```
FWERbonfCorrected=length(which(pvalsVector<=0.05/4))/10000


T1Ecorrection=-log10(pvalsVector[500])
bonfCorrected=-log10((0.05/4))

ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+xlab("Iteration") + ylab("-log10(p)")+ ylim(0, 6
```



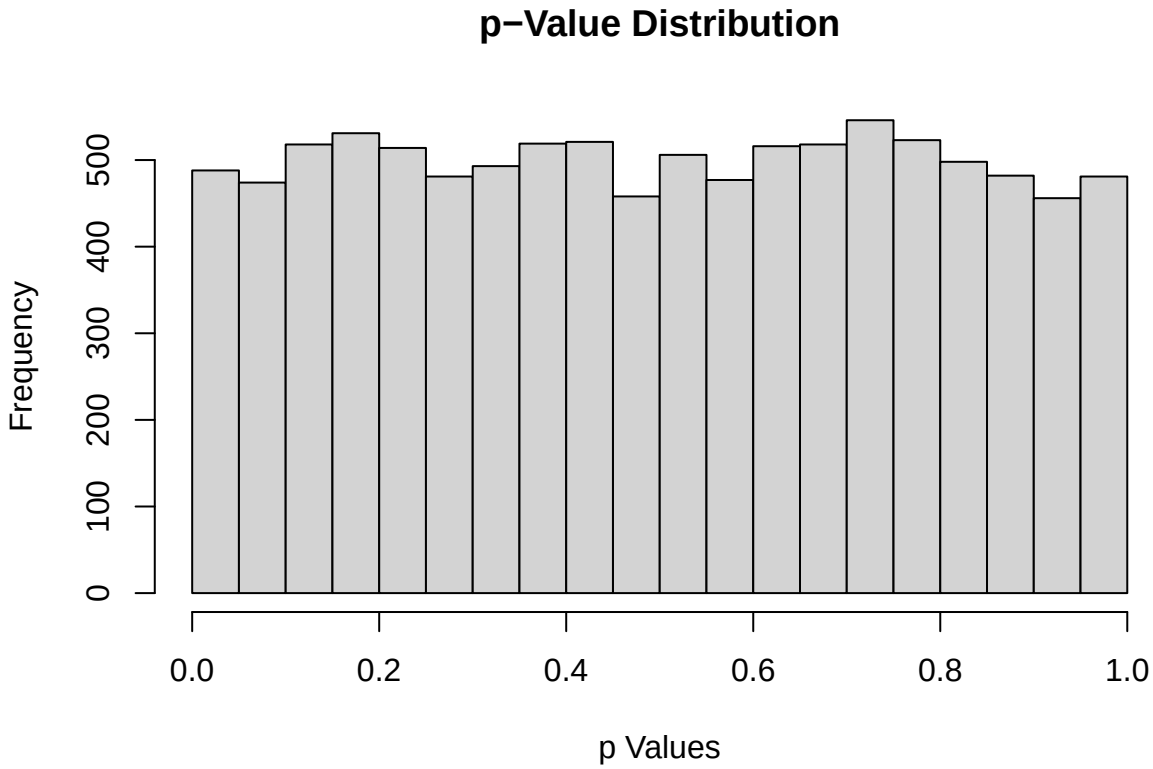
Effect of Single Test: Genotypic coding

```
nRep=10000
n=1000
meanTrait=170
SdTrait=7
maf=0.2
p=1-maf
q=maf

GenoPhenoTable=array(data=NA, dim=c(n,6))
colnames(GenoPhenoTable)=c("phenotypes","genotypesADD","genotypesDOM",
                           "genotypesREC","genotypesGENOsnp1", "genotypesGENOsnp2")
GenoPhenoTable=data.frame(GenoPhenoTable)


pVals=array(data=NA,dim=c(nRep,7))
colnames(pVals)=c("Iter","genotypesADDpValue", "genotypesDOMpValue",
                  "genotypesRECpValue", "genotypesGENOpValue", "MinPval", "NegLogMinP")
pVals=data.frame(pVals)

for( i in 1:nRep){
  set.seed(i)

  ### generate a placeholder for the lowest pValue
  minP=1
```

```
  ### Generate Phenotypes ###
  phenotype=rnorm(n,meanTrait,SdTrait)
  GenoPhenoTable$phenotypes=phenotype

  ### Generate Genotypes ###
  genotypes=sample(c("AA","Aa","aa"), size=n, replace=T, prob=c(p^2, 2*p*q,q^2))

  ############################
  ### Genotypic SNP Coding ###
  ############################

  ## coding key:
  ## 0 SNP1copy and 0 SNP2copy=homozygous, no snp (PP)
  ## 1 SNP1copy and 0 SNP2copy=heterozygous with 1 copy SNP (pq)
  ## 0 SNP1copy and 1 SNP2copy=homozygous SNP, 2 copies SNP (qq)

  ## 0=PP, 1=heterozygous Pq, 2=qq
  snp1=ifelse(genotypes=="Aa",1,0)
  snp2=ifelse(genotypes=="aa",1,0)

  GenoPhenoTable$genotypesGENOsnp1=snp1
  GenoPhenoTable$genotypesGENOsnp2=snp2

  model=lm(phenotypes~genotypesGENOsnp1+genotypesGENOsnp2, data = GenoPhenoTable)
  modelSum=summary(model)
  fstat=modelSum$fstatistic
  pValGeno=pf(fstat[1], fstat[2],fstat[3], lower.tail = FALSE)

  pVals$genotypesGENOpValue[i]=pValGeno
  if (minP>pValGeno){ minP=pValGeno}

  ## ADD THE LOWEST P VALUE OF ALL TESTS
  pVals$MinPval[i]=minP
  pVals$NegLogMinP[i]=log10(minP)*-1
  pVals$Iter[i]=i

}

library(ggplot2)
suppressMessages(library(qqman))

qq(pVals$MinPval,
main = "Q-Q plot")
```




```
hist(pVals$MinPval, xlab = "p Values", main = "p-Value Distribution")
```

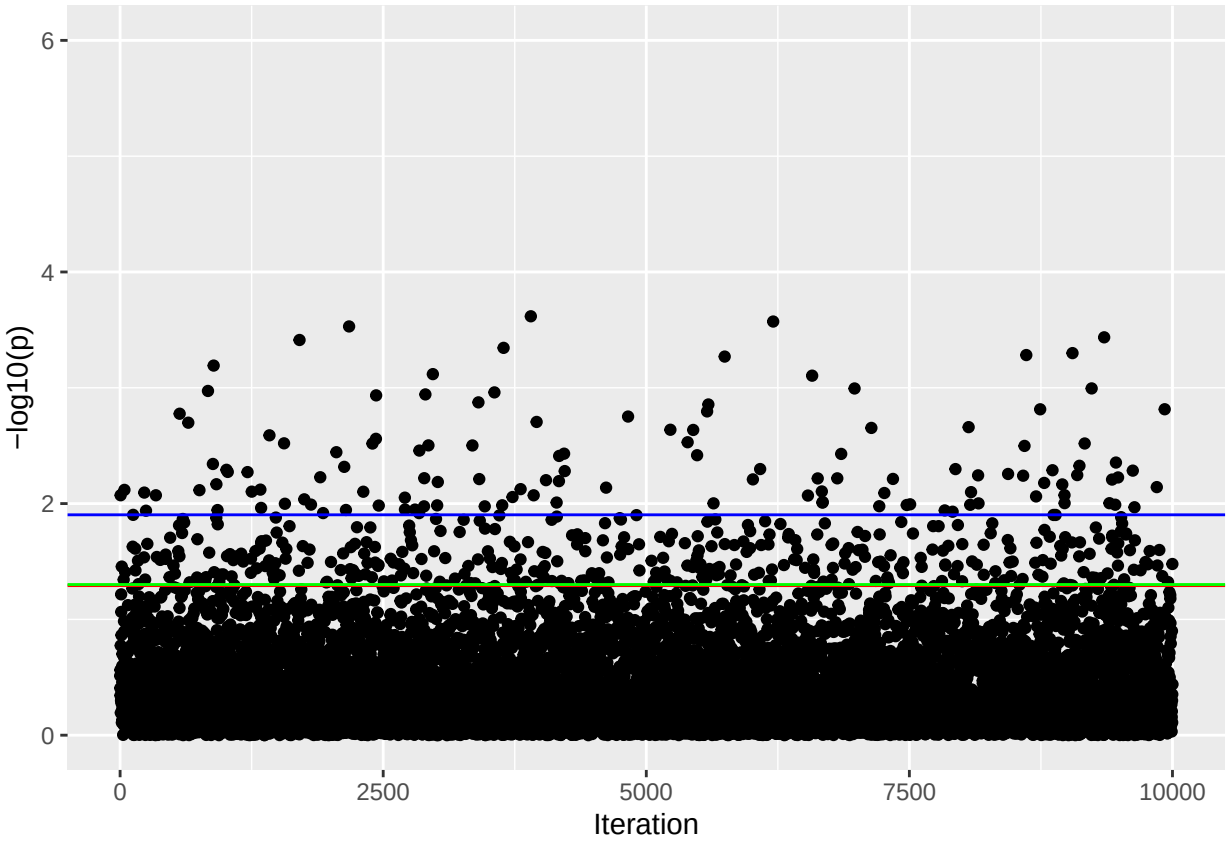



```
# ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+ xlab("Iteration") + ylab("-log10(p)")+ ylim(0

pvalsVector=pVals$MinPval
```

```
pvalsVector=sort(pvalsVector)


T1ER=length(which(pvalsVector<=0.05))/10000
sprintf("The T1ER was: %f", T1ER)
```

```
## [1] "The T1ER was: 0.048800"
```

```
FWERbonfCorrected=length(which(pvalsVector<=0.05/4))/10000

T1Ecorrection=-log10(pvalsVector[500])
bonfCorrected=-log10((0.05/4))

ggplot(pVals, aes(x=Iter, y=NegLogMinP))+ geom_point()+xlab("Iteration") + ylab("-log10(p)")+ ylim(0, 6)
```



Discussion

A simulation study was performed to assess the T1E of the minimum p-value approach and observe the effect of this approach on the p-value distribution. The simulated phenotype data was generated from one normal distribution, and the genotype data was randomly generated from a HWE with a MAF=0.2. The genotype and phenotype data were then paired and coded with either dominant, recessive, additive or genotypic coding before being tested for statistical significance. Given the phenotypes were generated from the same normal distribution and the genotypes were randomly assigned to phenotype data, any significant results are indicative of a false positive or a type one error. Moreover, because we are performing four statistical tests and selecting the smallest p-value to be recorded, we expect to see a significant inflation in the TIE.

Under the null, we expect to see a uniform distribution of p-values. To observe this effect and contrast

our results of the simulation, we repeated the study but with only one statistical test being performed for each of the 10,000 repetitions (see P-value Uniform Under the Null: Observing Effect of Single Test). The histogram of the p-values displays the typical characteristic of a uniform distribution. Contrasting these results with the p-value histogram obtained for the simulation study, we see a significant departure from a uniform distribution, whereby a gradient is produced towards smaller p-values. One interesting phenomenon of the p-value histogram is the shape of the distribution; there is a linear trend from the bottom right to the top left of increasing frequency of smaller p-values. This is a logical relationship, given the setup of our testing. Under a biased selection of smaller p-values, we should obtain an increased probability of higher values. Indeed, this is observed; the frequency of P-values in the smallest bucket of the histogram (0.00-0.05) is approximately two times greater in the minimum p-value simulation compared to the single test simulation. Given that we have a T1E rate of ~10%, twice as many significant tests at alpha=0.05 are expected.

It is important to note that the minimum p-value approach is not inherently an incorrect testing methodology. As we discussed in class, there is ambiguity when performing genetic association studies for what the correct coding method is (dominant recessive, etc.), and the minimum p-value approach can be used to assess multiple biological methods. However, when using the minimum p-value approach, one must adequately adjust the significance threshold to control the T1E. Using the typical alpha=0.05 significance level, the T1E rate of the minimum p-value approach was calculated to be approx. 10%. To control the T1E of this approach at 5%, these simulation studies can be used to select thresholds that account for the increase in false positives. If we are comfortable with a T1E rate of 5%, we can calculate the 95%th percentile of p-values obtained and select a significance threshold that matches this threshold. Using this methodology corresponded to a p-value threshold of ~0.022 to control the T1E rate of the minimum p-value approach at 5%.

As discussed in the lecture, there is a desire to provide smaller p-values when possible, and this can create erroneous associations or cause researchers to select models based on significance rather than biological plausibility. Moreover, certain programs will provide non-statistically significant thresholds that can lead researchers with limited understanding to consider findings non-significant findings as significant; this phenomenon was observed while using Plink, which reports values that do not achieve the standard identified to be p<5*10^-8. Here, I was able to complete a simulation study that observed the effect of multiple hypothesis testing and the minimum p-value approach. This project allowed the visualization of the inflation of the T1E that follows from selectively choosing models that provide smaller p-values. Moreover, we were able to obtain a new significance threshold to adequately control the T1E rate at 5% for the minimum p-value approach.

References

1. Nicholson KJ, Sherman M, Divi SN, Bowles DR, Vaccaro AR. The Role of Family-wise Error Rate in Determining Statistical Significance. Clin Spine Surg. 2022 Jun 1;35(5):222–3.
2. Head ML, Holman L, Lanfear R, Kahn AT, Jennions MD. The Extent and Consequences of P-Hacking in Science. PLoS Biol [Internet]. 2015 Mar 13 [cited 2023 Dec 10];13(3). Available from: /pmc/articles/PMC4359000/