Assignment 2.2 Part 2

Mahdi

September 29, 2020

Part 2

Task 1

1. How many samples are included in this dataset?

```
expr_ceu_20 <- read.table("data/expr_ceu_chr20.tab", header = T) #gene expr vs samps
gene_pos <- read.table("data/expr_chr20.pos", header = T)
snp_ceu_20 <- read.table("data/geno_ceu_chr20_strict.tab", header=T) #snp vs samps
snp_pos <- read.table("data/geno_ceu_chr20_strict.pos", header = T)

dim(expr_ceu_20)[2] -1
```

```
## [1] 91
```

There are 91 samples in the dataset.

2.How many variants are present on chromosome 20?

```
dim(snp_pos)[1]
```

```
## [1] 30000
```

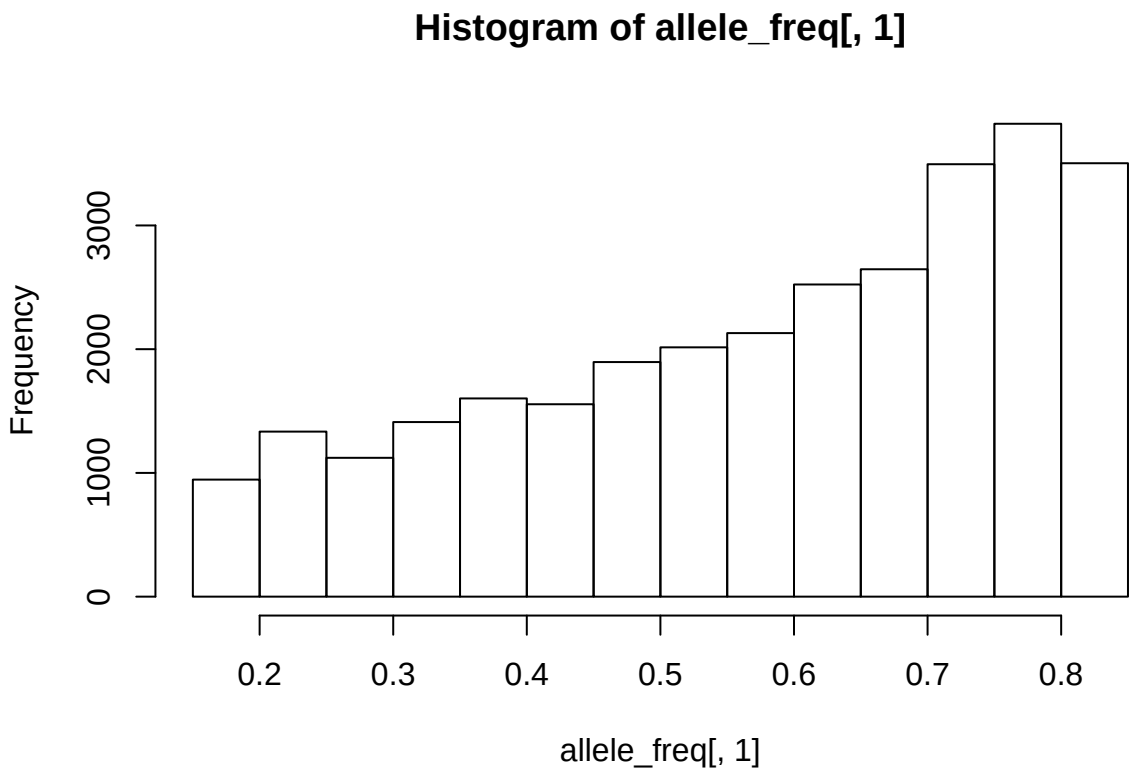
There are 30000 SNPS on chromosome 20.

3. Generate a histogram of allele frequencies for chromosome 20.

```
get_geno_frequency <- function(geno, snp_mat){
  boolean_matrix <- snp_mat == geno
  count <- apply(boolean_matrix, 1, sum)
  freq <- count/dim(snp_mat)[2]
  return (freq)
}

get_af <- function(snp_mat){
  genotypes <- c(0,1,2)
  #get matrix of genotype frequencies
  geno_freq <- sapply(genotypes, get_geno_frequency, snp_mat)
  colnames(geno_freq) <- genotypes
  allele_freq <- geno_freq + geno_freq[,2]/2
  allele_freq <- allele_freq[,-2]
  #maf <- apply(allele_freq, 1, min)
  return (allele_freq)
}
```

```
allele_freq <- get_af(snp_ceu_20)
hist(allele_freq[,1])
```




```
#hist(allele_freq[,2])
```

4. What is the lowest allele frequency observed?

```
min(allele_freq[,1])
```

```
## [1] 0.1521739
```

The lowest allele frequency is 0.1521739.

5. How many genes are included?

```
dim(expr_ceu_20)[1]
```

```
## [1] 561
```

There are 561 genes in the dataset.

6. What gene shows the highest mean expression?

```
mean_expr <- apply(expr_ceu_20[,-1], 1 , mean)
expr_ceu_20[mean_expr == max(mean_expr),1]
```

```
## [1] ENSG00000227063.4
## 561 Levels: ENSG00000000419.7 ENSG00000020256.14 ... ENSG00000261582.1
```

Gene ENSG00000227063.4 shows the highest mean expression.

Task 2

```
#Matrix eQTL
library(MatrixEQTL)

# Genotype file names
SNP_file_name = "data/geno_ceu_chr20_strict.tab" ; #Genotype file path
snps_location_file_name = "data/geno_ceu_chr20_strict.pos" ; #snp position file path

# Gene expression file names
expression_file_name = "data/expr_ceu_chr20.tab" ;#Expression file path
gene_location_file_name = "data/expr_chr20.pos" ;#gene position file path

# Only associations significant at this level will be saved
pvOutputThreshold_cis = 1; #p.value threshold for cis eqtls
pvOutputThreshold_tra = 0; #p.value threshold for trans eqtls

#Covariates file names
covariates_file_name = character();# Set to character() for no covariates

# Distance for local gene-SNP pairs
cisDist = 1e6; #Define cis distance

## Load genotype data
snps = SlicedData$new();
snps$fileDelimiter = "\t"; # the TAB character
snps$fileOmitCharacters = "NA"; # denote missing values;
snps$fileSkipRows = 1; # one row of column labels
snps$fileSkipColumns = 1; # one column of row labels
snps$fileSliceSize = 2000; # read file in slices of 2,000 rows
snps$LoadFile(SNP_file_name);
```

```
## Rows read: 2,000
## Rows read: 4,000
## Rows read: 6,000
## Rows read: 8,000
## Rows read: 10,000
## Rows read: 12,000
## Rows read: 14,000
## Rows read: 16,000
## Rows read: 18,000
## Rows read: 20,000
## Rows read: 22,000
## Rows read: 24,000
## Rows read: 26,000
## Rows read: 28,000
## Rows read: 30,000
```

```
## Rows read: 30000 done.
```

```
## Load gene expression data
gene = SlicedData$new();
gene$fileDelimiter = "\t"; # the TAB character
gene$fileOmitCharacters = "NA"; # denote missing values;
gene$fileSkipRows = 1; # one row of column labels
gene$fileSkipColumns = 1; # one column of row labels
gene$fileSliceSize = 2000; # read file in slices of 2,000 rows
gene$LoadFile(expression_file_name);
```

```
## Rows read: 561 done.
```

```
#Load position files
snpspos = read.table(snps_location_file_name, header = TRUE, stringsAsFactors = FALSE);
genepos = read.table(gene_location_file_name, header = TRUE, stringsAsFactors = FALSE);

## Run the analysis
me = Matrix_eQTL_main(
  snps = snps,
  gene = gene,
  output_file_name=NULL,
  pvOutputThreshold = pvOutputThreshold_tra,
  useModel = modelLINEAR,
  errorCovariance =numeric(),
  verbose = TRUE,
  output_file_name.cis = NULL, #Do not write out cis results
  pvOutputThreshold.cis = pvOutputThreshold_cis,
  snpspos = snpspos,
  genepos = genepos,
  cisDist = cisDist,
  min.pv.by.genesnp = FALSE,
  noFDRsaveMemory = FALSE,
  pvalue.hist = FALSE)
```

```
## Matching data files and location files

## 561 of 561 genes matched

## 30000 of 30000 SNPs matched

## Task finished in 0.02 seconds

## Reordering SNPs

## Task finished in 0.15 seconds

## Reordering genes

## Task finished in 0.07 seconds

## Processing covariates

## Task finished in 0 seconds

## Processing gene expression data (imputation, residualization)

## Task finished in 0.01 seconds

## Creating output file(s)

## Task finished in 0 seconds
```

```
## Performing eQTL analysis
##  6.66% done, 56,158 cis-eQTLs
## 13.33% done, 101,338 cis-eQTLs
## 20.00% done, 113,643 cis-eQTLs
## 26.66% done, 127,676 cis-eQTLs
## 33.33% done, 136,430 cis-eQTLs
## 40.00% done, 166,641 cis-eQTLs
## 46.66% done, 187,389 cis-eQTLs
## 53.33% done, 245,058 cis-eQTLs
## 60.00% done, 286,249 cis-eQTLs
## 66.66% done, 331,024 cis-eQTLs
## 73.33% done, 377,660 cis-eQTLs
## 80.00% done, 409,288 cis-eQTLs
## 86.66% done, 425,208 cis-eQTLs
## 93.33% done, 456,812 cis-eQTLs
## 100.00% done, 527,117 cis-eQTLs
## Task finished in 1.36 seconds
##
```

```
cis_eqtls = me$cis$eqtls[,-c(5)]
cis_eqtls["beta_se"] = cis_eqtls["beta"]/cis_eqtls["statistic"]
rm(me)
```

1. How many tests were conducted? 527,117 tests were conducted.
2. Using a bonferroni correction ($\alpha = 0.05$), how many genes are significant?

```
cis_eqtls$p.adj <- p.adjust(cis_eqtls$pvalue, method = "bonferroni")
sum(cis_eqtls$p.adj <= 0.05)
```

```
## [1] 71
```

71 genes are significant

3. What gene-snp pair show the lowest pvalue? What is the effect size of this snp-gene pair?

```
cis_eqtls[cis_eqtls$p.adj== min(cis_eqtls$p.adj),]
```

```
##                snps              gene statistic       pvalue      beta
## 1   snp_20_37055875 ENSG00000196756.5 -9.420136 5.066679e-15 -8.146604
## 2 snp_20_37055875.1 ENSG00000196756.5 -9.420136 5.066679e-15 -8.146604
##     beta_se        p.adj
## 1 0.8648075 2.670733e-09
## 2 0.8648075 2.670733e-09
```

The snp_20_37055875, ENSG00000196756.5 pair has the lowest p value. The effect size is 8.146604.

4. What is the biotype of this gene? ???????

Task 3

```
# Genotype file names
SNP_file_name = "data/geno_ceu_chr22_strict.tab" ; #Genotype file path
snps_location_file_name = "data/geno_ceu_chr22_strict.pos" ; #snp position file path

# Gene expression file names
expression_file_name = "data/expr_ceu_chr20.tab" ;#Expression file path
gene_location_file_name = "data/expr_chr20.pos" ;#gene position file path

# Only associations significant at this level will be saved
pvOutputThreshold_cis = 0; #p.value threshold for cis eqtls
pvOutputThreshold_tra = 1; #p.value threshold for trans eqtls

#Covariates file names
covariates_file_name = character();# Set to character() for no covariates

# Distance for local gene-SNP pairs
cisDist = 1e6; #Define cis distance

## Load genotype data
snps = SlicedData$new();
snps$fileDelimiter = "\t"; # the TAB character
snps$fileOmitCharacters = "NA"; # denote missing values;
snps$fileSkipRows = 1; # one row of column labels
snps$fileSkipColumns = 1; # one column of row labels
snps$fileSliceSize = 2000; # read file in slices of 2,000 rows
snps$LoadFile(SNP_file_name);
```

```
## Rows read: 1001 done.
```

```
## Load gene expression data
gene = SlicedData$new();
gene$fileDelimiter = "\t"; # the TAB character
gene$fileOmitCharacters = "NA"; # denote missing values;
gene$fileSkipRows = 1; # one row of column labels
gene$fileSkipColumns = 1; # one column of row labels
gene$fileSliceSize = 2000; # read file in slices of 2,000 rows
gene$LoadFile(expression_file_name);
```

```
## Rows read: 561 done.
```

```
#Load position files
snpspos = read.table(snps_location_file_name, header = TRUE, stringsAsFactors = FALSE);
genepos = read.table(gene_location_file_name, header = TRUE, stringsAsFactors = FALSE);

## Run the analysis
me = Matrix_eQTL_main(
  snps = snps,
  gene = gene,
  output_file_name=NULL,
  pvOutputThreshold = pvOutputThreshold_tra,
  useModel = modelLINEAR,
  errorCovariance =numeric(),
  verbose = TRUE,
```

```
  output_file_name.cis = NULL, #Do not write out cis results
  pvOutputThreshold.cis = pvOutputThreshold_cis,
  snpspos = snpspos,
  genepos = genepos,
  cisDist = cisDist,
  min.pv.by.genesnp = FALSE,
  noFDRsaveMemory = FALSE,
  pvalue.hist = FALSE)
```

```
## Processing covariates

## Task finished in 0 seconds

## Processing gene expression data (imputation, residualization)

## Task finished in 0.02 seconds

## Creating output file(s)

## Task finished in 0 seconds

## Performing eQTL analysis

## 100.00% done, 561,561 eQTLs

## Task finished in 0.42 seconds

##
```

```
trans_eqtls = me$all$eqtls[,-c(5)]
trans_eqtls["beta_se"] = trans_eqtls["beta"]/trans_eqtls["statistic"]
rm(me)
```

1. How many tests were conducted? 527,117 tests were conducted.
2. Using a bonferroni correction ($\alpha = 0.05$), how many genes are significant?

```
trans_eqtls$p.adj <- p.adjust(trans_eqtls$pvalue, method = "bonferroni")
sum(trans_eqtls$p.adj <= 0.05)
```

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

None of the genes are significant

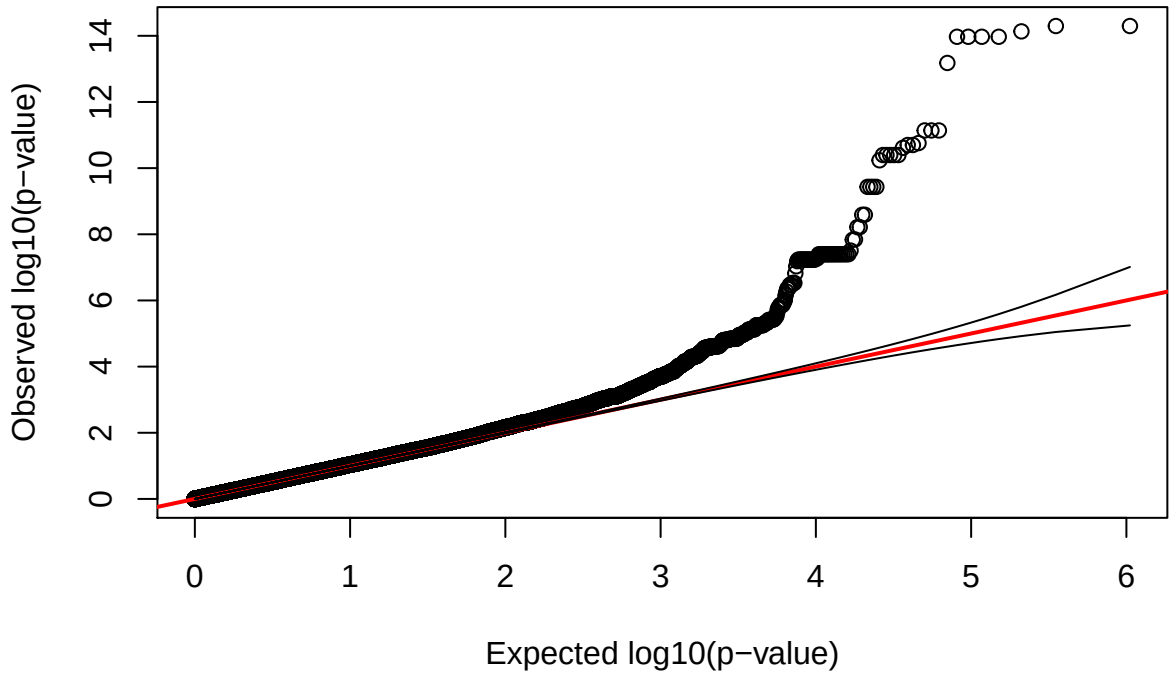
Task 4

1. Briefly explain what a QQ-plot can be used for (2-3 sentences) A QQ-plot can be used to check if a distribution is Normally distributed. For example, if we have some data and we assume it is normally ditributed, we can make a QQ-plot of the data and get a rough idea if our assumption is correct.
2. Compute the QQ-plot for both the cis and trans eqtl separately

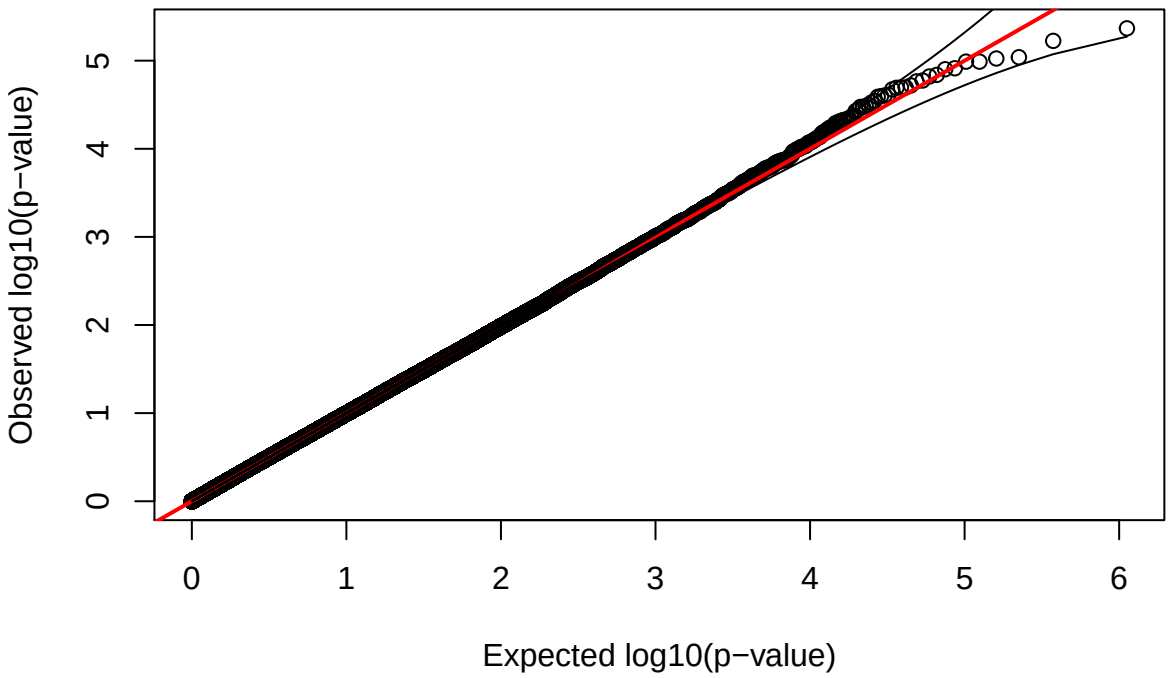
```
qqp<-function(x, maxLogP=30,...){
  x<-x[!is.na(x)]
  if(!missing(maxLogP)){
    x[x<10^-maxLogP]<-10^-maxLogP
  }
  N<-length(x)
  chi1<-qchisq(1-x,1)
  x<-sort(x)
  e<- -log((1:N-0.5)/N,10)
  plot(e,-log(x,10),ylab="Observed log10(p-value)",xlab="Expected log10(p-value)",...)
```

```
  abline(0,1,col=2,lwd=2)
  c95<-qbeta(0.95,1:N,N-(1:N)+1)
  c05<-qbeta(0.05,1:N,N-(1:N)+1)
  lines(e,-log(c95,10))
  lines(e,-log(c05,10))
}

qqp(cis_eqtls$pvalue)
```




```
qqp(trans_eqtls$pvalue)
```



3. Explain the plots The red line represents where the points would be if the expected and observed distributions were exactly the same, therfore when the black points fall on the red line, it means the two distributions are normally distributed. From the graphs we can see that the cis eQTLs are not normally distributed while the trans eQTLs are.
4. What is the main difference between these two QQ-plots? The main difference is that for the cis eQTLs the observed p values are much lower than the expected p values. This means we find significant sites more often than expected.
5. Explain what drives this? Cis eQTLs are SNPs that affect the gene expression of nearby genes. This means one cis eQTL could affect multiple genes in the regiion so these eQTLs are more likely to affect gene expression than random. In contrast, trans eQTLs affect distant genes so knowing the posititon of a trans eQTL doesn't say anything about which genes could be affected, so the p values are indepdendent and nromally distributed.

Task 5

1. Calculate the PVE for all cis SNP-gene pairs and make a histogram of them

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

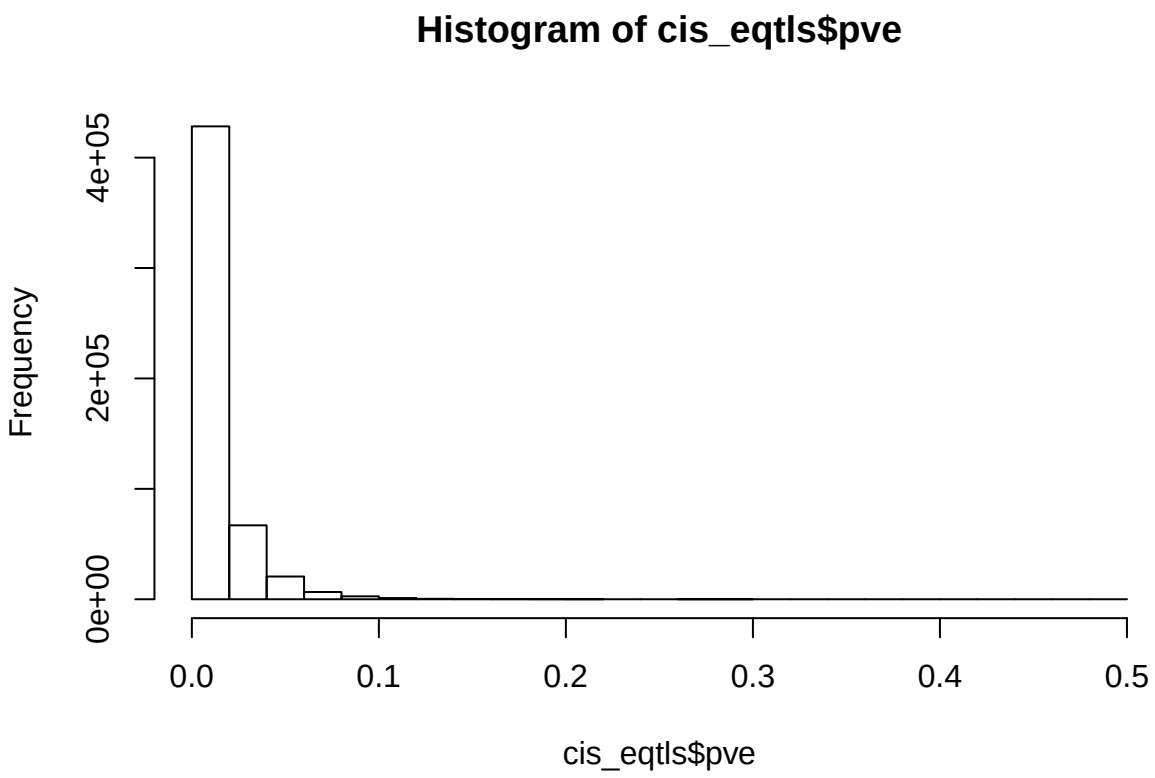
```
## x dplyr::lag()    masks stats::lag()
```

```
get_geno_frequency <- function(geno, snp_mat){
  boolean_matrix <- snp_mat == geno
  count <- apply(boolean_matrix, 1, sum)
  freq <- count/dim(snps)[2]
  return (freq)
}

get_maf <- function(snp_mat){
  genotypes <- c(0,1,2)
  #get matrix of genotype frequencies
  geno_freq <- sapply(genotypes, get_geno_frequency, snp_mat)
  colnames(geno_freq) <- genotypes
  allele_freq <- geno_freq + geno_freq[,2]/2
  allele_freq <- allele_freq[,-2]
  maf <- apply(allele_freq, 1, min)
  return (maf)
}

pve <- function(b, b_se, maf, N){
  return(
    (2*(b^2)*maf*(1-maf))/(
      (2*(b^2)*maf*(1-maf)) + (b_se^2)*2*N*maf*(1-maf)
    )
  )
}

maf_df <- data.frame("id" = snp_ceu_20$id, "maf" = get_maf(snp_ceu_20))
temp_df <- data.frame("id" = cis_eqtls$snps)
temp_df <- merge(temp_df, maf_df, by="id", all.x = T)
cis_eqtls %>% arrange(snps) -> cis_eqtls
cis_eqtls$maf <- temp_df$maf
N <- dim(snp_ceu_20)[2] - 1
cis_eqtls$pve <- pve(cis_eqtls$beta, cis_eqtls$beta_se, cis_eqtls$maf, N)
hist(cis_eqtls$pve)
```



2. What gene has the highest PVE

```
cis_eqtls %>%
  arrange(desc(pve)) %>%
  head(1) %>%
  select(gene)
```

```
##                gene
## 1 ENSG00000196756.5
```

3. What other factors can explain the remaining variance (mention 2)? Contribution form other SNPs, expression of other genes and noise.