# Midterm

## Noelle Wheeler

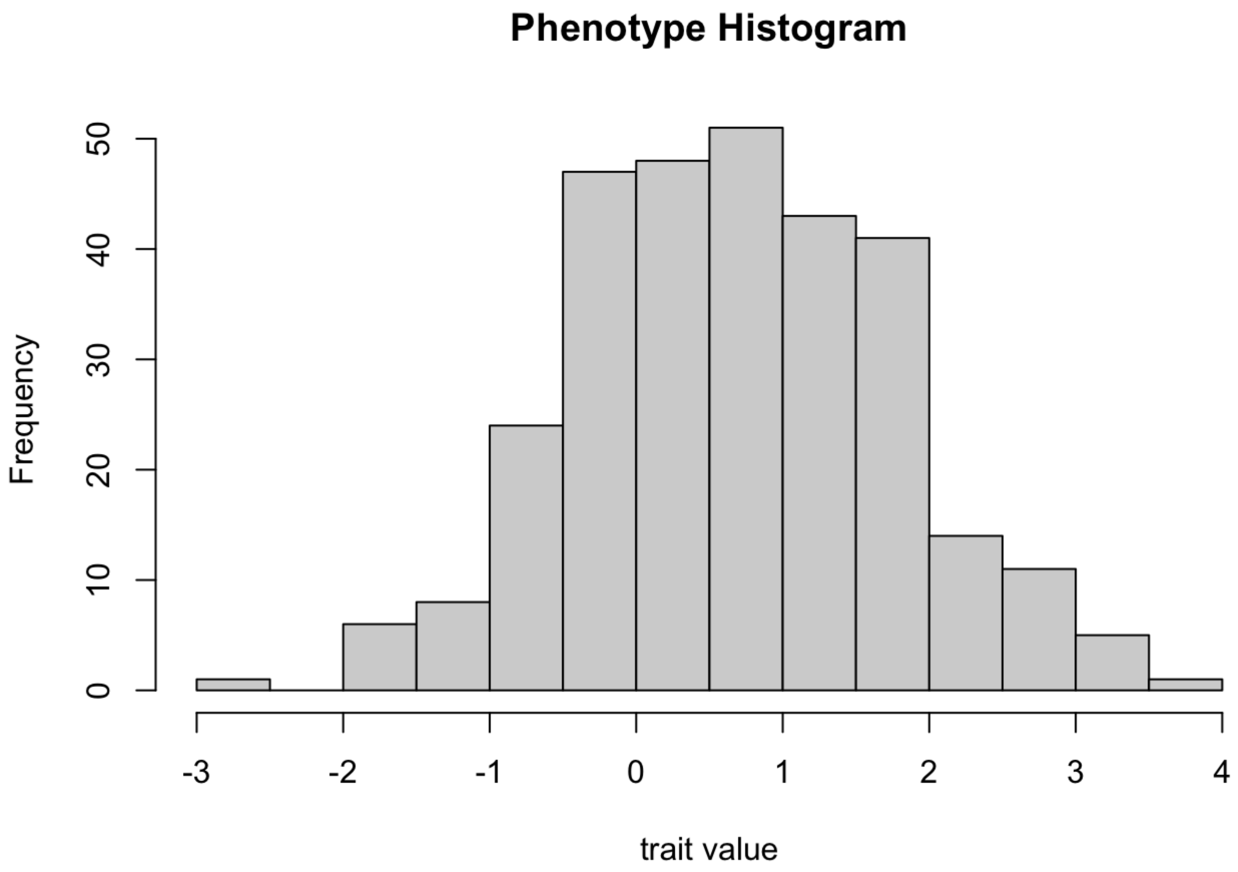
## 2023-03-29

1. Import the phenotype data from the file ‘midterm phenotypes.txt’ and (a) Calculate and report the total sample size n, (b) Plot a histogram of the phenotypes.

```
df_pheno <- read.table("/Users/noelawheeler/Desktop/quan genomics/Quan-genomics-assignme
nts/midterm/midterm_phenotypes.txt")
sample_size <- nrow(df_pheno)
print(paste0("There are ", sample_size, " samples."))
```

```
## [1] "There are 300 samples."
```

```
hist(df_pheno$V1, main='Phenotype Histogram', xlab='trait value')
```



2. Import the genotype data from the file ‘midterm genotypes.txt’, (a) Calculate and report the number of SNPs N, (b) Calculate the MAF for every SNP and plot a histogram of the MAFs.

```
df_geno <- read.table("/Users/noelawheeler/Desktop/quan genomics/Quan-genomics-assignments/midterm/midterm_genotypes.txt", header = TRUE, sep = ",")
num_SNP <- ncol(df_geno)
print(paste0("There are ", num_SNP, " SNPs."))
```

```
## [1] "There are 3000 SNPs."
```

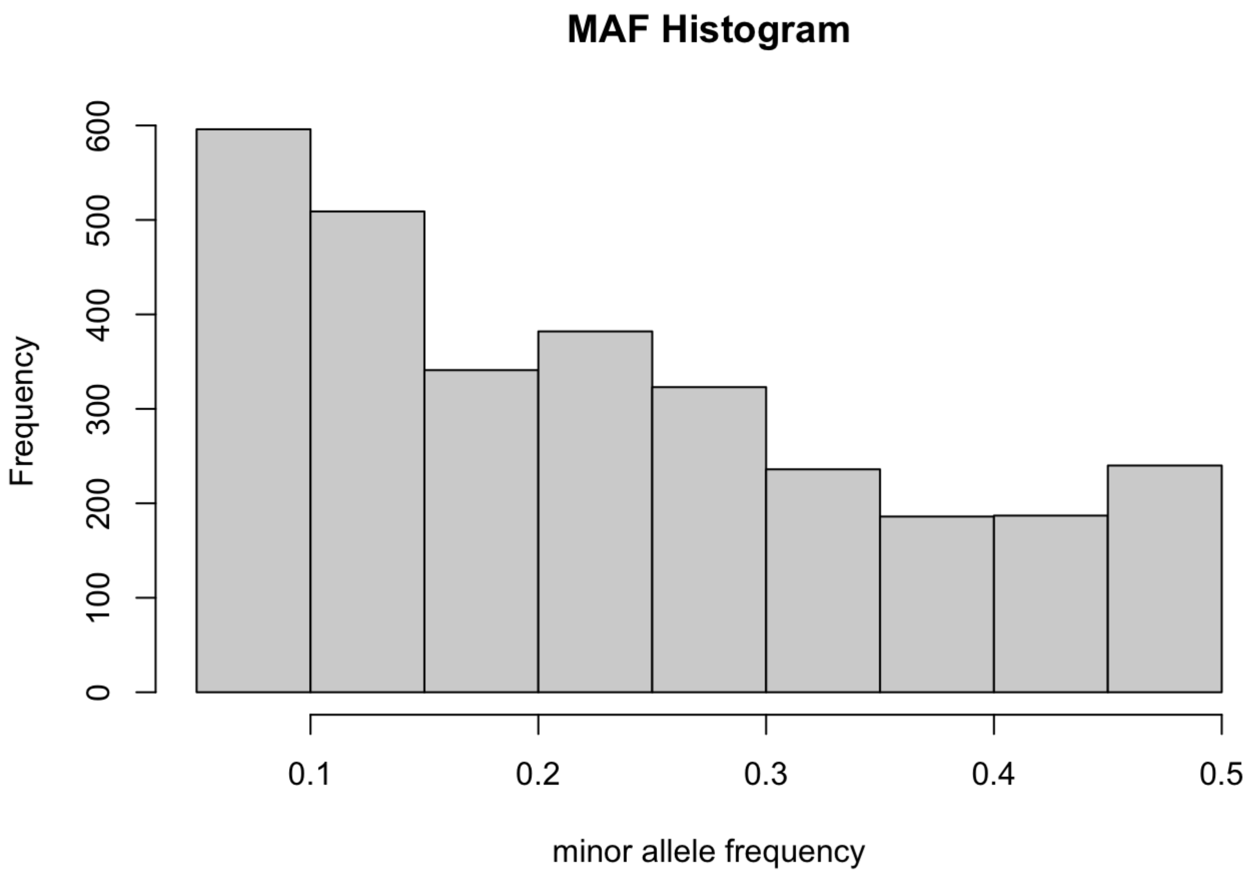
```
# get minor allele
allele1 <- df_geno[seq(1, nrow(df_geno), 2), ]
allele2 <- df_geno[seq(2, nrow(df_geno), 2), ]

# create function to find maf
get_maf <- function(x) {
  minor_allele <- head(names(sort(table(x))), 1)
  count_minor_allele <- sum(x==minor_allele)
  count_major_allele <- sum(x!=minor_allele)
  maf <- count_minor_allele/(count_minor_allele+count_major_allele)
  return(maf)
}

# function to get major allele, will use later
get_major_allele <- function(x){
  major_allele <- tail(names(sort(table(x))), 1)
  return(major_allele)
}

# apply function to all columns
maf <- apply(df_geno, 2, get_maf)

# plot histogram of maf
hist(maf, main='MAF Histogram', xlab='minor allele frequency')
```



3. Write code to calculate $MLE(\hat{\beta}) = [\hat{\beta}_{\mu}, \hat{\beta}_{a}, \hat{\beta}_{d}]$ for each SNP and plot a histogram of all the $\hat{\beta}_{\mu}$, plot a histogram of all the $\hat{\beta}_{a}$, plot a histogram of all the $\hat{\beta}_{d}$.

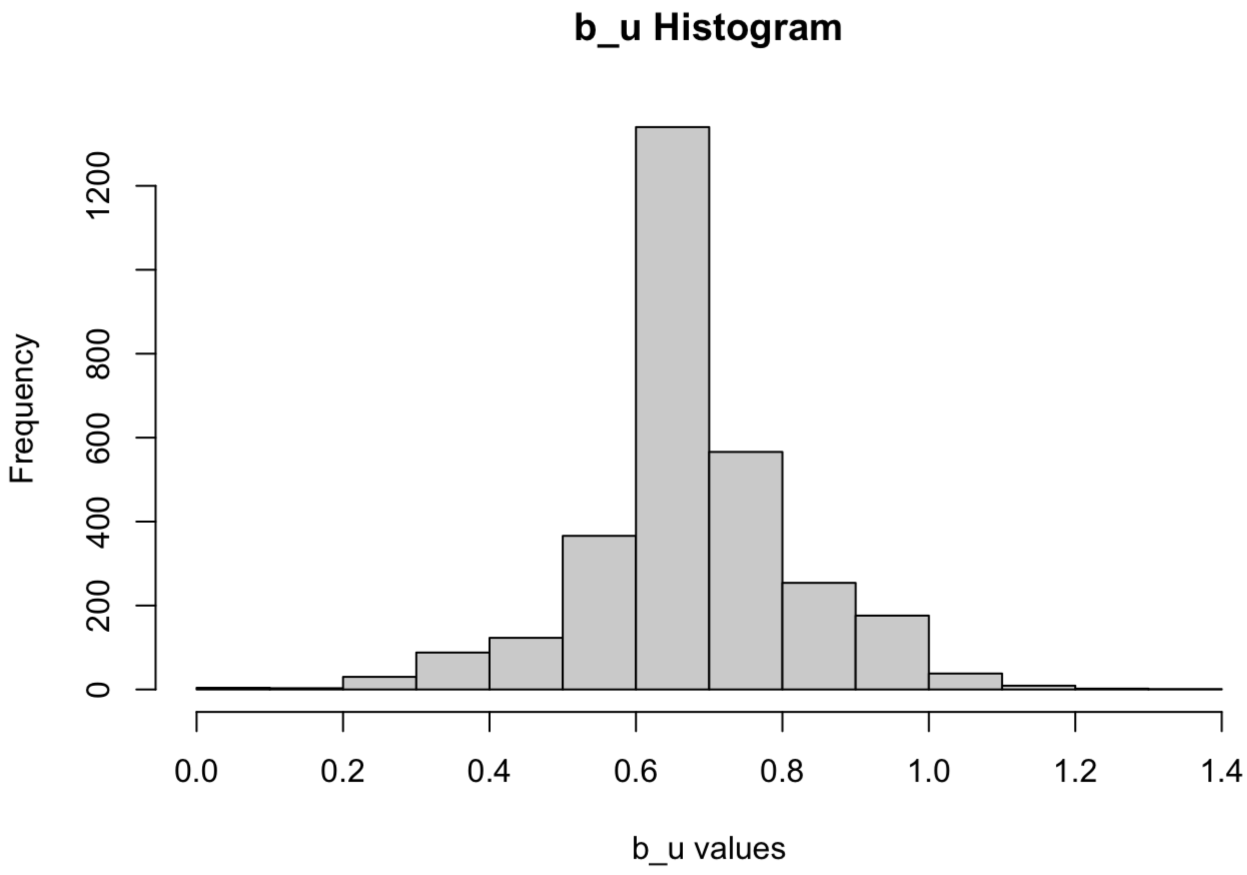
```
# create x_d matrix
x_d <- allele1[,] == allele2[,]
x_d <- ifelse(x_d, -1, 1)

# create x_a matrix
x_a <- x_d
x_a[x_a == 1] <- 0
samples <- 300
snp <- 3000
major_allele <- apply(df_geno, 2, get_major_allele)
mat <- matrix(replicate(sample_size, major_allele),nrow=num_SNP)
mat <- t(mat)
# if both alleles equal the minor allele, change -1 to 1
major_allele_even <-  allele1[,] == mat[,]
major_allele_odd <-  allele2[,] == mat[,]
indices <- which(major_allele_even == FALSE & major_allele_odd == FALSE, arr.ind = TRUE)
x_a[indices] <- 1
```

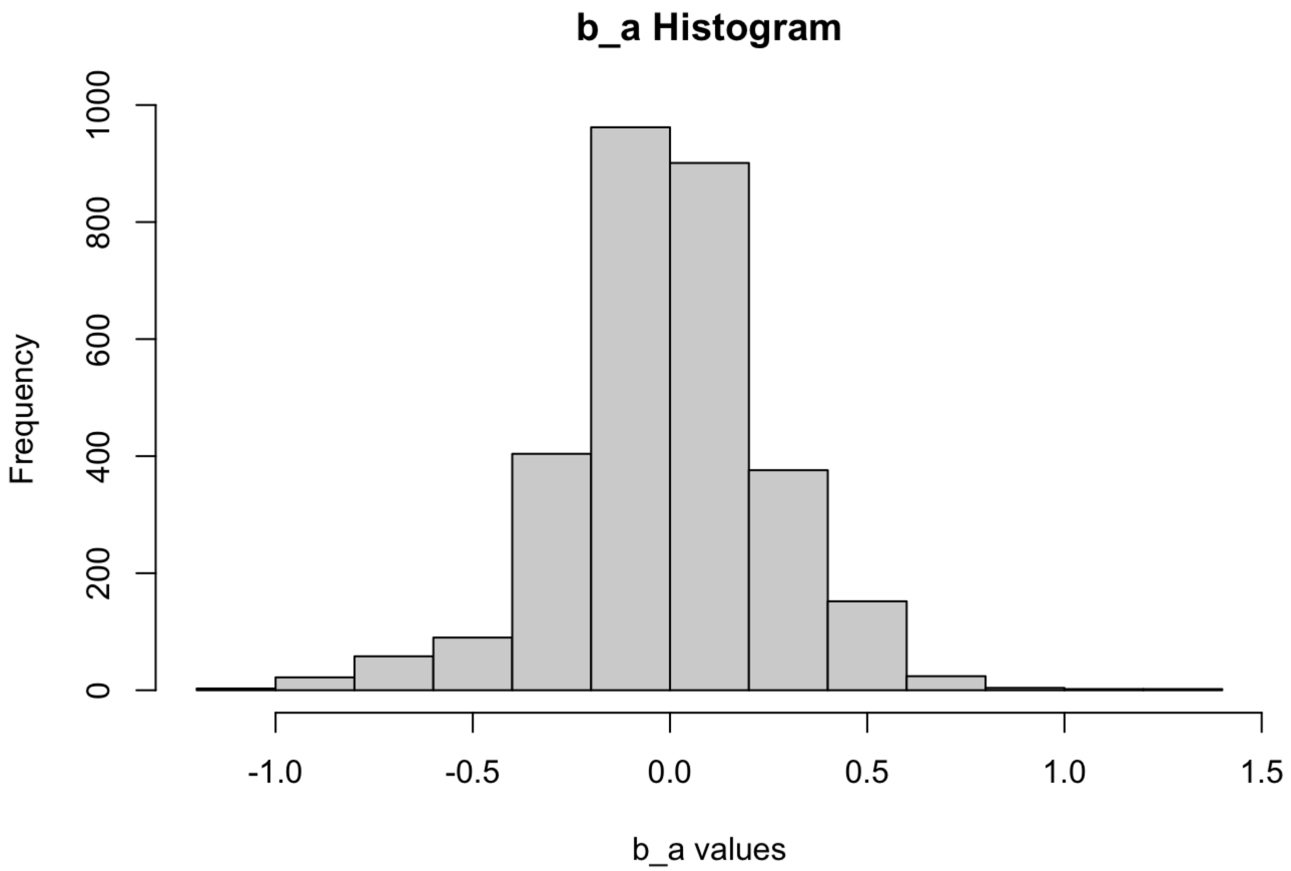
```
library(magrittr)
library(MASS)
# calculate beta_mu, beta_a, beta_d
beta <- function(y, xa_i, xd_i) { # Create X matrix
  X_mx_i <- cbind(1, xa_i, xd_i)
  # Compute MLE_beta
  beta_i <- ginv(t(X_mx_i) %*% X_mx_i) %*% t(X_mx_i) %*% y
  return(beta_i)
}

beta_vals <- lapply(c(1:ncol(x_a)),
                function(i) { beta(as.matrix(df_pheno$V1), x_a[, i], x_d[, i])} )
# get b_u, b_a, b_d
b_u <- c()
b_a <- c()
b_d <- c()
for (i in seq(1:3000)){
  b_u <- append(b_u, beta_vals[[i]][1])
  b_a <- append(b_a, beta_vals[[i]][2])
  b_d <- append(b_d, beta_vals[[i]][3])
}
```

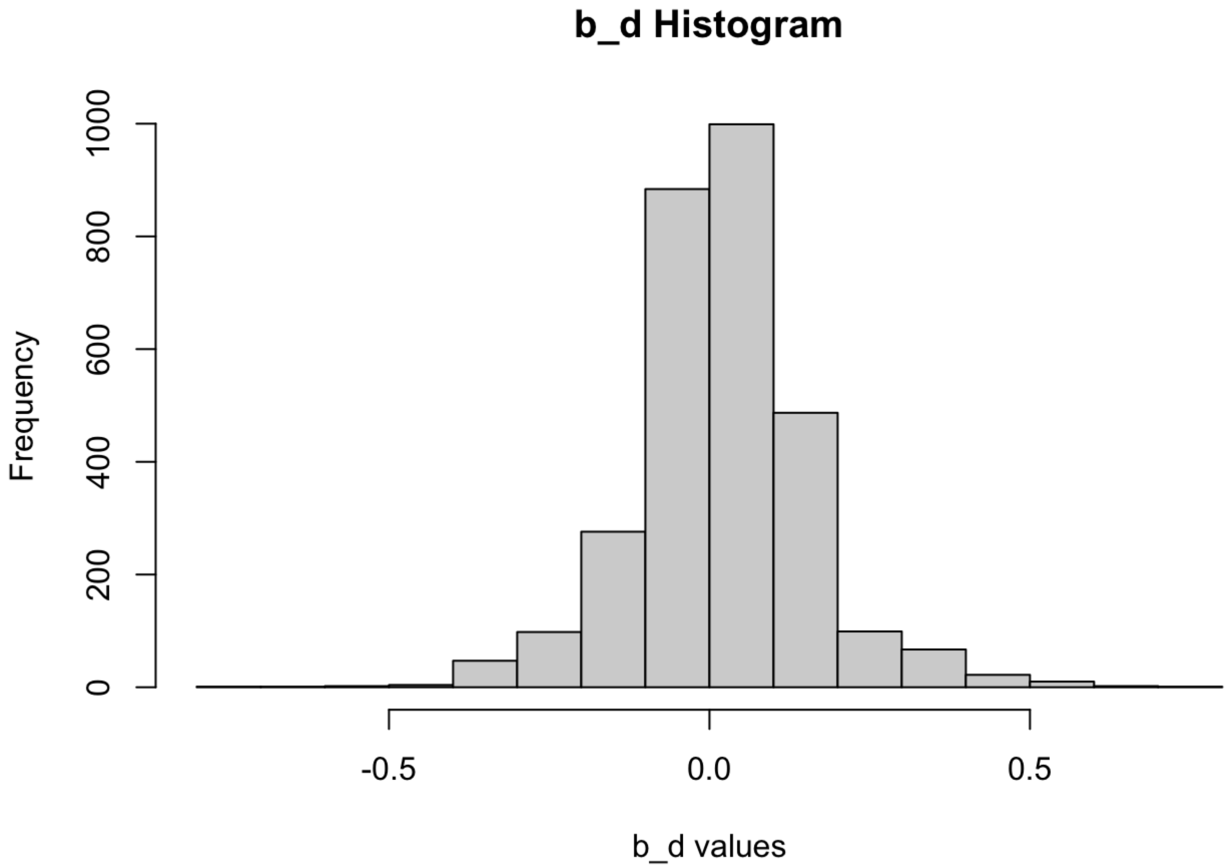
```
hist(b_u, main='b_u Histogram', xlab='b_u values')
```



```
hist(b_a, main='b_a Histogram', xlab='b_a values')
```



```
hist(b_d, main='b_d Histogram', xlab='b_d values')
```

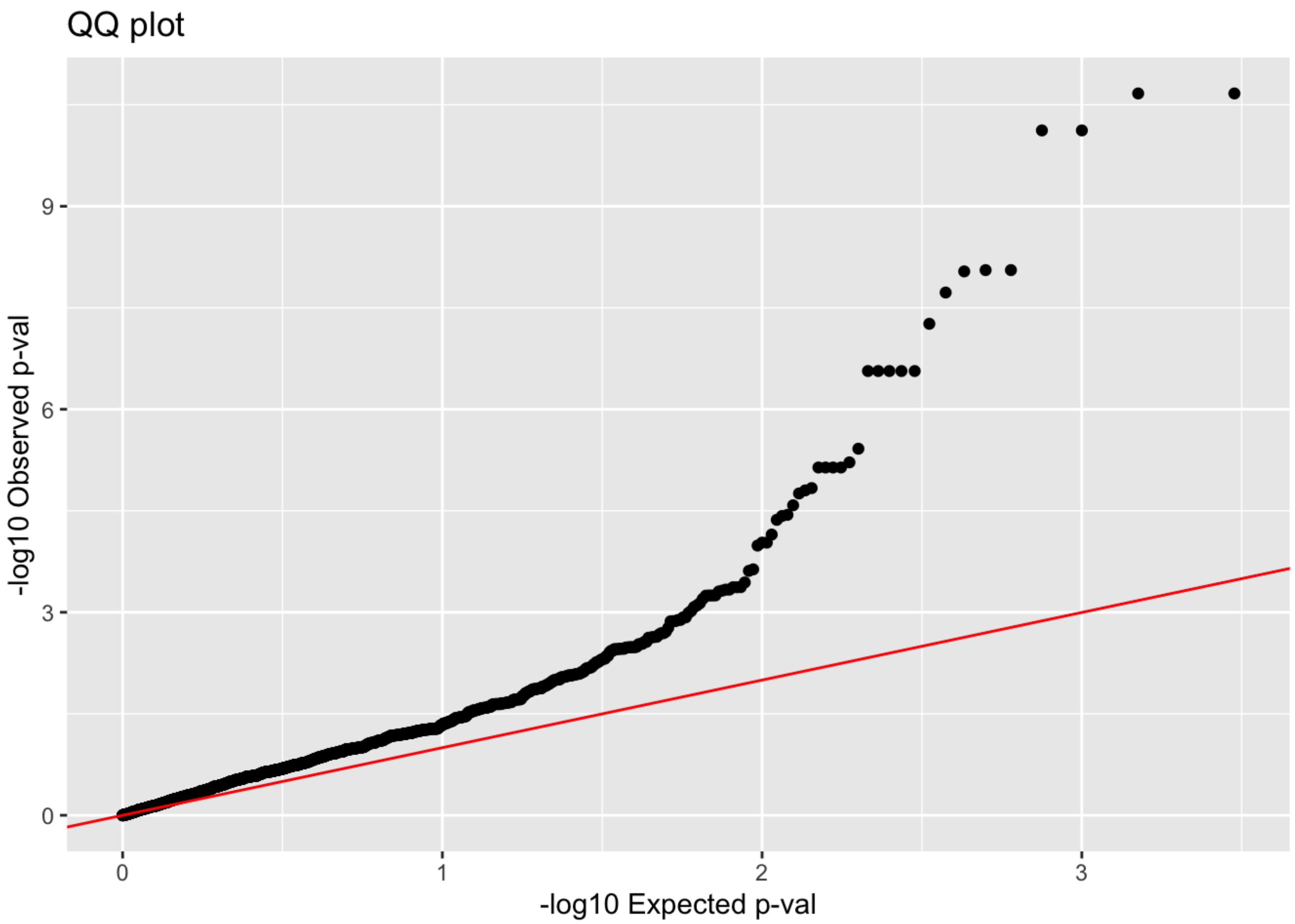


4. For each SNP, calculate p-values for the null hypothesis $H_0 : \beta_a = 0 \cap \beta_d = 0$ versus the alternative hypothesis $H_A : \beta_a \neq 0 \cup \beta_d \neq 0$ when applying the genetic linear regression model.

```
get_p_vals <- function(y, x_a, x_d) { # Create X matrix
  # Calculate SSM,SSE,MSM,MSE,F-statistic, and p-value
  X_mx_i <- cbind(1, x_a, x_d)
  beta_i <- ginv(t(X_mx_i) %*% X_mx_i) %*% t(X_mx_i) %*% y
  y_hat <- X_mx_i %*% beta_i
  SSM <- sum((y_hat - mean(y))^2)
  SSE <- sum((y - y_hat)^2)
  df_M <- 3 - 1
  df_E <- sample_size - 3
  MSM <- SSM / df_M
  MSE <- SSE / df_E
  # Get a test statistic for a likelihood ratio test
  f_stat <- MSM / MSE
  # Get pvalue
  pval <- pf(f_stat, df_M, df_E, lower.tail = FALSE)
  return(pval)
}
pvals <- lapply(c(1:ncol(x_a)),
function(i) { get_p_vals(as.matrix(df_pheno$V1), x_a[, i], x_d[,i])} ) %>% do.call(rbind, .)
```

5. For the p-values you calculated in question [4], (a) Produce a QQ plot for these p-values (label your plot and your axes using informative names!), (b) USING NO MORE THAN TWO SENTENCES answer the following question: based on this QQ plot, do you think you have achieved an appropriate model fit with your analysis and why do you think this is the case?

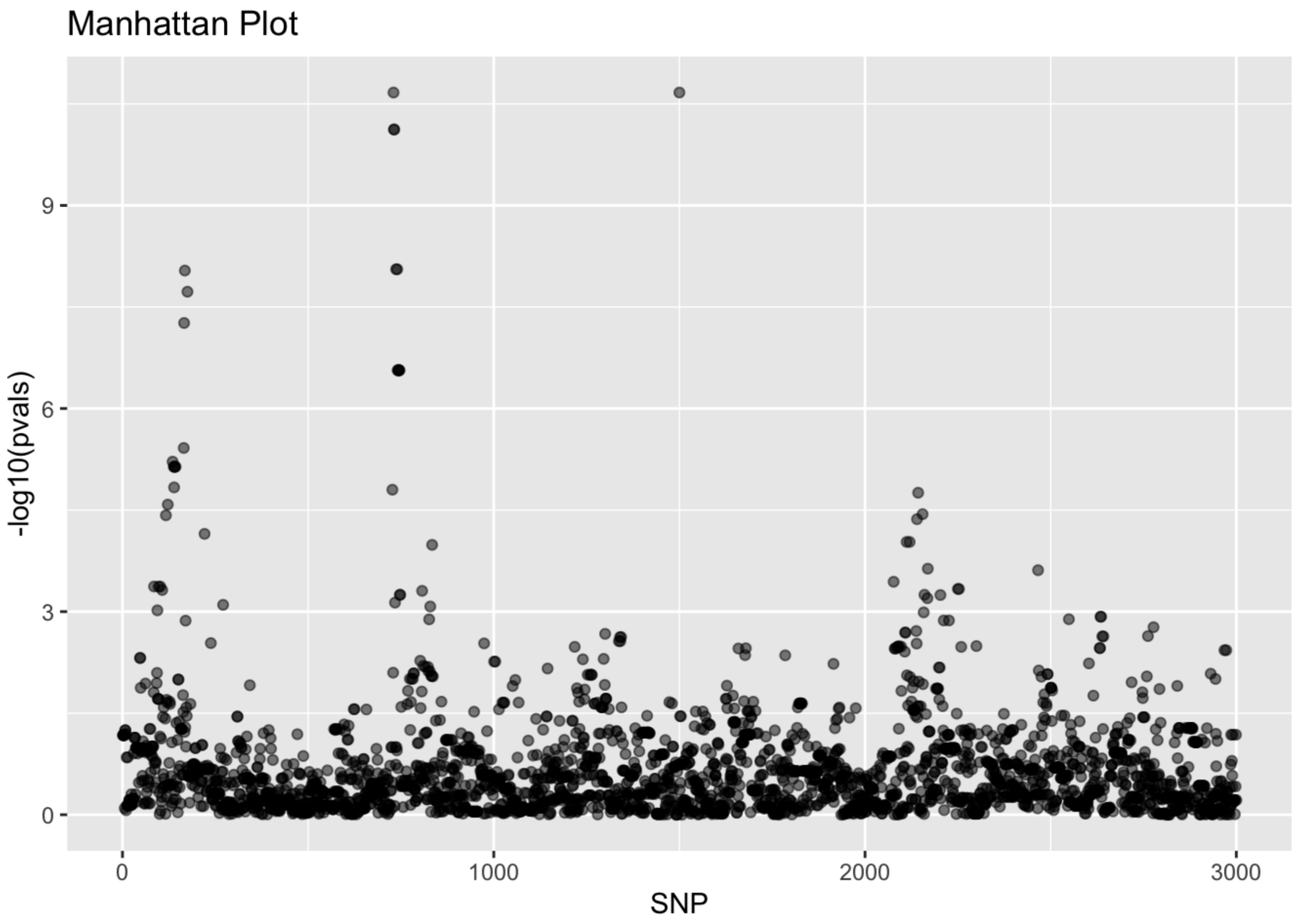
```
library(ggplot2)
# get sorted observed and expected p-values
observed_pvals = sort(pvals)
expected_pvals = qunif(seq(0, 1, length.out = length(observed_pvals) + 2), min = 0, max
= 1)
expected_pvals = expected_pvals[expected_pvals != 0 & expected_pvals != 1]
# combine into dataframe
p_df = data.frame(observed = -log10(observed_pvals),
                  expected = -log10(expected_pvals))
#plot
ggplot(p_df, aes(x = expected, y = observed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = 'red') +
  labs(x = '-log10 Expected p-val',
       y = '-log10 Observed p-val',
       title = 'QQ plot')
```



We should see most the of the points fall along the expected (red) line and a few point towards the right above the line - indicating that they are significant. However, we see a large number of points falling above the line which means we may be getting false positive due to covariates or some other factor, so it does not seem like we achieved the appropriate model fit.

6. For the p-values you calculated in question [4], (a) Produce a Manhattan plot, (b) Report HOW MANY SNPs (not which, just how many!) you find to be significant when controlling the study-wide type 1 error of 0.05 using a Bonferroni correction (note: do NOT use adjusted p-values! Just use the p-values you calculated in question [4])

```
man_plot <- data.frame(index = 1:length(pvals), pval = pvals)
ggplot(man_plot, aes(index, -log10(pvals))) +
  geom_point(alpha=0.5) + ggtitle("Manhattan Plot") + xlab("SNP")
```



```
# significant SNPs
num_significant <- sum(pvals <= .05)
print(paste0("There are ",num_significant, " significant SNPs"))
```

```
## [1] "There are 307 significant SNPs"
```

```
# significant SNPs after correction
p_adj <- p.adjust(pvals, method = "bonferroni")
adj_num_sig <- sum(p_adj <= .05)
print(paste0("After bonferroni correction there are ",adj_num_sig, " significant SNPs"))
```

```
## [1] "After bonferroni correction there are 22 significant SNPs"
```

7. USING NO MORE THAN TWO SENTENCES answer the following question: based on your answer the question [6], how many distinct `peaks’ do you think you have identified and why?

I think I have three distinct peaks. This is because there appears to be three “towers” in my manhattan plot where multiple SNPs show significance.

8.

a. What is a casual polymorphism?
A casual polymorphism is a SNP that has a significant effect on the phenotype of interest.

b. Why do you observe ‘peaks’ in your Manhattan plot?
We observe peaks in the manhattan plot because we take the -log of our p-values so that small (significant) locations show up as large values in our manhattan plot. These significant locations are grouped together because SNPs that are close together on the genome tend to be correlated with each other.

c. Describe why the peaks in your Manhattan plot may indicate the genomic position of a causal polymorphism but not (necessarily) the actual causal polymorphism?
SNPs that are close together tend to be correlated because of the way that genetic recombination occurs. A SNP that is highly correlated with a casual SNP will also show up as casual, even if it is not necessarily causing an effect on the phenotype, so many SNPs in a genomic position will appear to be casual.

d. Provide one reason why a peak may NOT indicate the position of a causal polymorphism.
A peak may not indicate the position of a casual polymorphism if there is an unaccounted for covariate.

9.

a. Provide a rigorous definition of the ‘power’ of a hypothesis test.
Power is the probability of incorrectly rejecting the null hypothesis when it is false. It is the ability of a test to detect a real effect.
b. List three factors that could impact the power of a hypothesis test in a GWAS.
Sample size, effect size, minor allele frequency.

10.

a. Provide a rigorous definition of a random variable.
A random variable is a real valued function on a sample space.
b. Provide a rigorous definition of a statistic.
A statistic is a function on a sample that is used to provide information about that sample.
c. Provide a rigorous definition of a p-value.
A p-value is the probability of getting your test statistic or a more extreme value, under the assumption that the null hypothesis is true.