final

Yuanyuan Wu

5/5/2019

1a

```
phenotypes <- read.csv("final2019_pheno.csv",
                       stringsAsFactors = F, header = F)
```

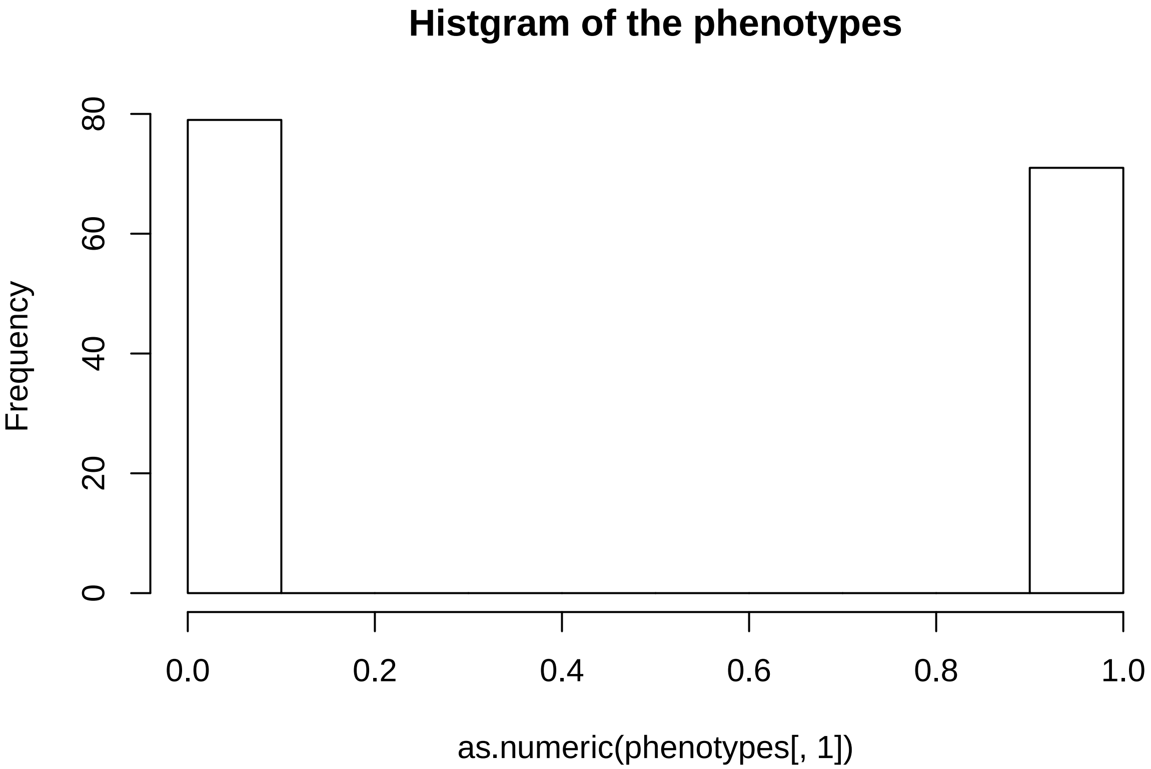
1b.

```
cat("The number of samples is: ",nrow(phenotypes),'\n')
```

```
## The number of samples is:  150
```

1c.

```
hist(as.numeric(phenotypes[,1]),main = 'Histgram of the phenotypes')
```



##2a.

```
genotypes <- read.csv("final2019_genotypes.csv",
                      stringsAsFactors = F, header = F)
```

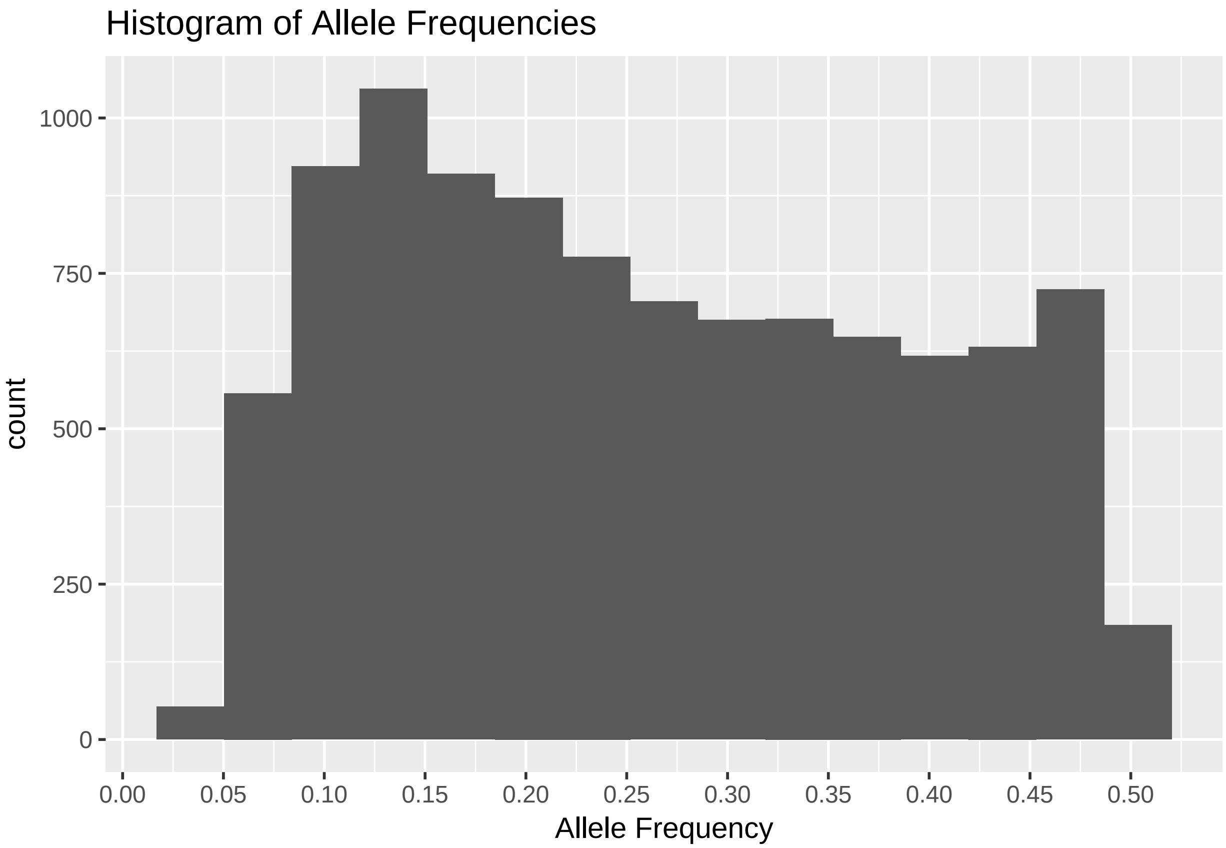
2b.

```r
cat("The number of SNPs N is: ",ncol(genotypes),'\n')
```

```
## The number of SNPs N is:  10000
```

2c.

```r
library(ggplot2)
maf_calc <- function(x){
  tab_x <- table(x)
  af <- 1-((2*max(tab_x[names(tab_x) %in% c(0,2)])+sum(x==1))/(2*nrow(genotypes)))
  return(af)
}

af <- apply(genotypes, 2, maf_calc)
ggplot(data.frame(af),aes(af))+geom_histogram(bins=15)+
  scale_x_continuous(breaks = seq(0,0.55,by=0.05))+
  labs(x="Allele Frequency", title="Histogram of Allele Frequencies")
```



3a.

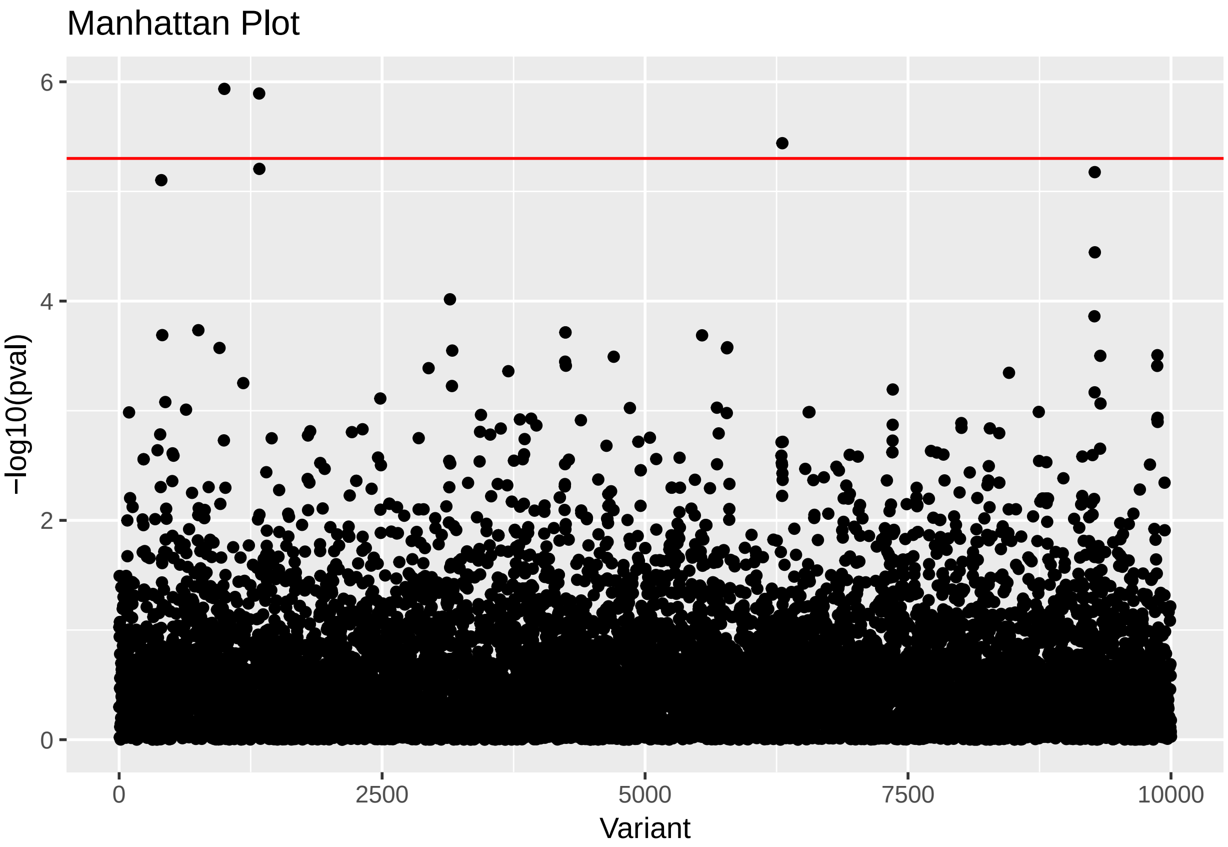
```r
#calculate p-val with IRLS
#most code are from Lab12
```

```r
#recursion and apply functions are used for speed
gamma_inv_calc <- function(X_mx, beta_t){
    #initialize gamma
    # K is the part which goes into the exponent
    K <- X_mx %*% beta_t
    gamma_inv <- exp(K)/(1+exp(K))
    return(gamma_inv)
}

W_calc <- function(gamma_inv){
        W <- diag(as.vector(gamma_inv * (1- gamma_inv)))
    return(W)
}

beta_update <- function(X_mx, W, Y, gamma_inv, beta){
  beta_up <- beta + ginv(t(X_mx)%*%W%*%X_mx)%*%t(X_mx)%*%(Y-gamma_inv)
    return(beta_up)
}

dev_calc <- function(Y, gamma_inv){
    deviance <- 2*( sum(Y[Y==1]*log(Y[Y==1]/gamma_inv[Y==1])) +
                    sum((1-Y[Y==0])*log((1-Y[Y==0])/(1-gamma_inv[Y==0]))) )
    return(deviance)
}

loglik_calc <- function(Y, gamma_inv){
    loglik <- sum(Y*log(gamma_inv)+(1-Y)*log(1-gamma_inv))
    return(loglik)
}

logistic.IRLS.recursive <- function(Y, X_mx, beta_t, dpt1, gamma_inv, iter, d.stop.th = 1e-6, it.max =
    # create empty matrix W
        W <- W_calc(gamma_inv)

        beta_t <- beta_update(X_mx, W, Y, gamma_inv, beta_t)

        #update gamma since it's a function of beta
        gamma_inv <- gamma_inv_calc(X_mx, beta_t)

        #calculate new deviance
        dt <- dev_calc(Y, gamma_inv)
        absD <- abs(dt - dpt1)

        if(absD < d.stop.th | iter > it.max) {
            #cat("Convergence at iteration:", i, "at threshold:", d.stop.th, "\n")
            logl <- loglik_calc(Y, gamma_inv)
            return(list(beta_t,logl))
        }   else {
          return(logistic.IRLS.recursive(Y, X_mx, beta_t, dt, gamma_inv, iter+1, d.stop.th = 1e-6, it.ma
        }
}
```

```
logistic.IRLS.pval.recursive <- function(Xa,Xd,Y, beta.initial.vec = c(0,0,0), d.stop.th = 1e-6, it.max
  #Initialize
  beta_t <- beta.initial.vec
    dt <- 0

  X_mx <- cbind(rep(1,nrow(Y)), Xa, Xd)
  gamma_inv <- gamma_inv_calc(X_mx, beta_t)
    h1 <- logistic.IRLS.recursive(Y, X_mx, beta_t, dt, gamma_inv, 1, d.stop.th = 1e-6, it.max = 100)

    X_mx <- cbind(rep(1,nrow(Y)), rep(0,nrow(Y)),rep(0,nrow(Y)))
  gamma_inv <- gamma_inv_calc(X_mx, beta_t)
    h0 <- logistic.IRLS.recursive(Y, X_mx, beta_t, dt, gamma_inv, 1, d.stop.th = 1e-6, it.max = 100)

    LRT <- 2*h1[[2]]-2*h0[[2]] #likelihood ratio test statistic
  pval <- pchisq(LRT, 2, lower.tail = F)
    return(pval)
}

Y <- as.matrix(phenotypes)
colnames(Y) <- NULL
xa_matrix <- as.matrix(genotypes)-1

xd_matrix <- 1 - 2*abs(xa_matrix)

allPvals <- apply(rbind(xa_matrix,xd_matrix), 2, function(x) logistic.IRLS.pval.recursive(Xa=x[1:nrow(x

log.allPval <- as.data.frame(-log10(allPvals),ncol=1)
adjusted_p <- -log10(0.05/ncol(genotypes))
```

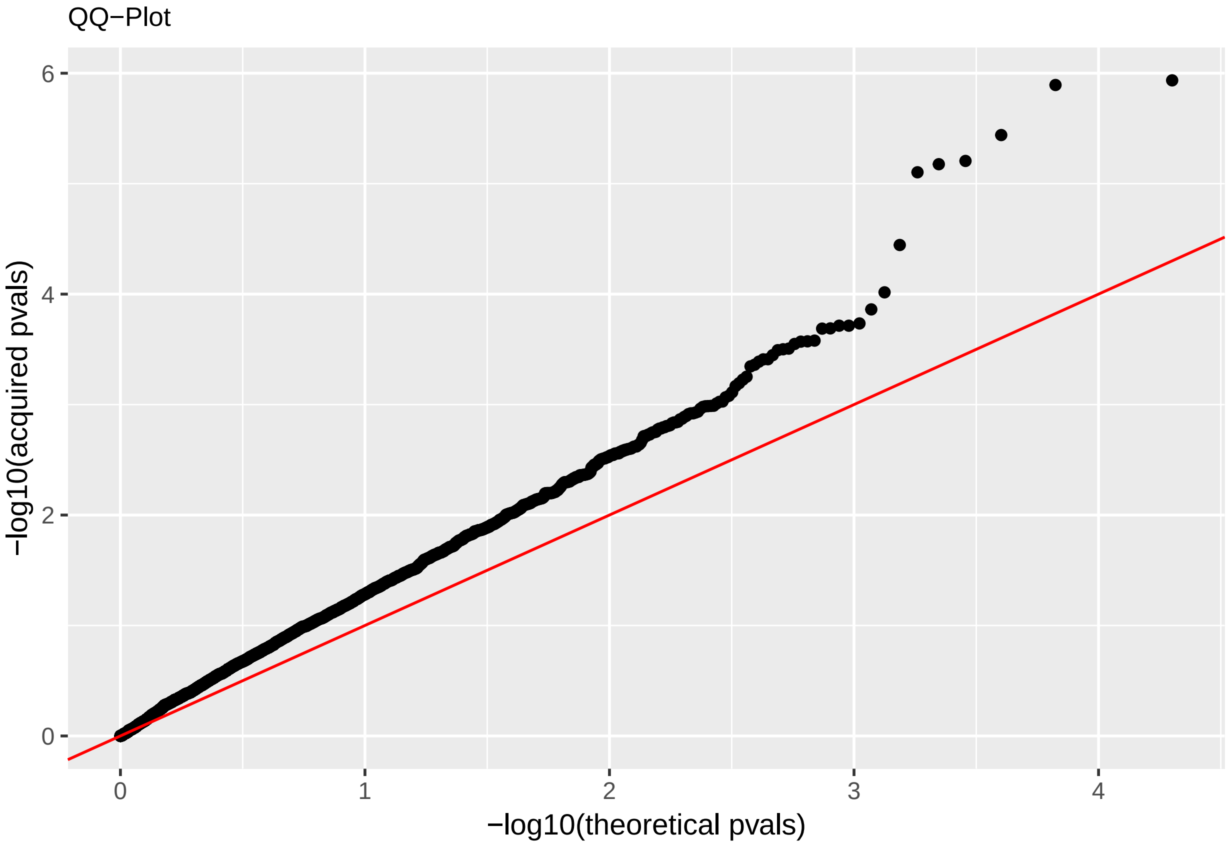
3b

```
library(ggplot2)
log.allPval <- as.data.frame(-log10(allPvals),ncol=1)
adjusted_p <- -log10(0.05/ncol(genotypes))


ggplot(log.allPval,aes(1:nrow(log.allPval),log.allPval[,1])) +
  geom_point() +
  labs(x="Variant",y="-log10(pval)",title=c("Manhattan Plot"))+
  geom_hline(yintercept=adjusted_p,color = "red")
```



##3c

```
qqDf1 <- data.frame(exp = sort(allPvals),theo = sort(qunif(ppoints(nrow(log.allPval)))))
ggplot(qqDf1,aes(-log10(theo), -log10(exp)))+geom_point()+
  geom_abline(slope = 1, color = "red")+
  labs(title="QQ-Plot",x="-log10(theoretical pvals)",y="-log10(acquired pvals)")+
  theme(plot.title = element_text(size=10))
```

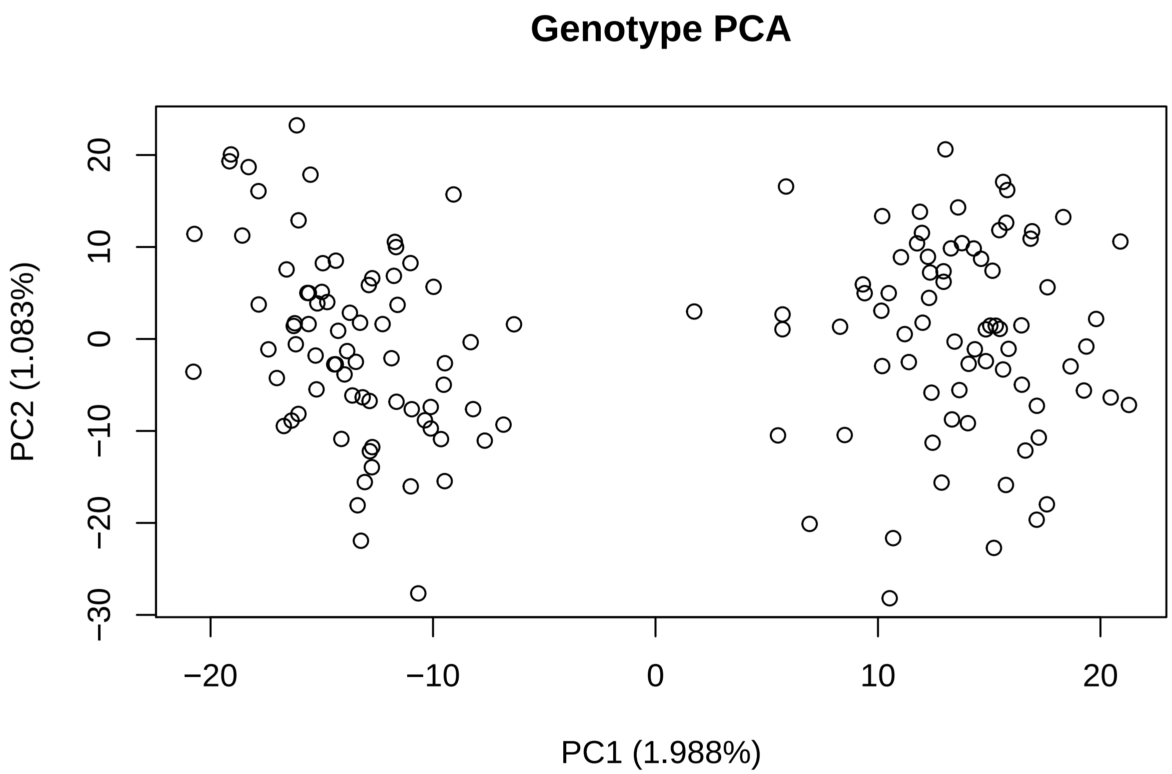


4a

```r
#perform PCA
geno_pca <- prcomp(genotypes,scale = T)
```

4b

```r
plot(geno_pca$x[,1],geno_pca$x[,2],main = "Genotype PCA",
     xlab = paste("PC1 (", 100*as.numeric(summary(geno_pca)$importance[, 1][2]), "%)", sep = ""),
     ylab = paste("PC2 (", 100*as.numeric(summary(geno_pca)$importance[, 2][2]), "%)", sep = ""))
```

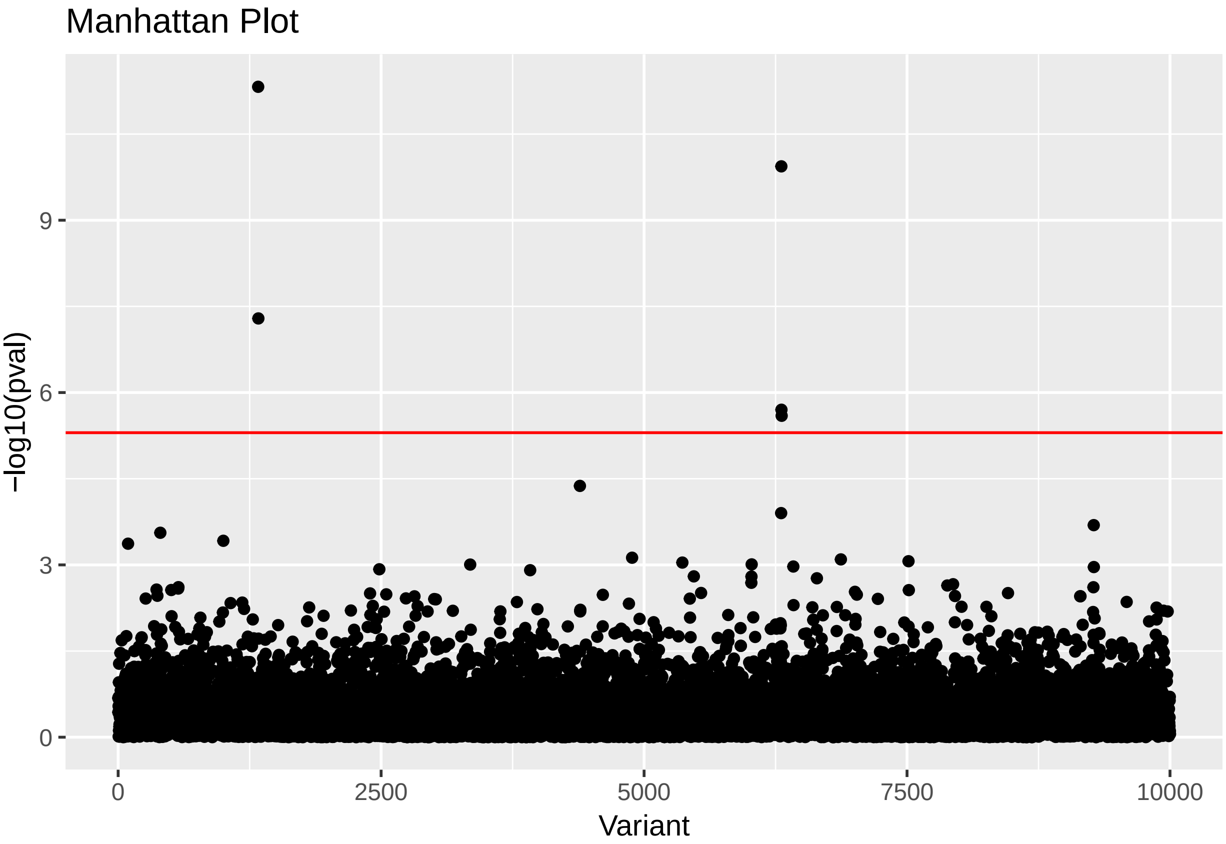


5a

```
#include PC1, 4 beta's
cov.logistic.IRLS.pval.recursive <- function(Xa,Xd,Y,PC1, beta.initial.vec = c(0,0,0,0), d.stop.th = 1e
  #Initialize
  beta_t <- beta.initial.vec
    dt <- 0

  X_mx <- cbind(rep(1,nrow(Y)), Xa, Xd,PC1)
  gamma_inv <- gamma_inv_calc(X_mx, beta_t)
    h1 <- logistic.IRLS.recursive(Y, X_mx, beta_t, dt, gamma_inv, 1, d.stop.th = 1e-6, it.max = 100)

    X_mx <- cbind(rep(1,nrow(Y)), rep(0,nrow(Y)),rep(0,nrow(Y)),PC1)
  gamma_inv <- gamma_inv_calc(X_mx, beta_t)
    h0 <- logistic.IRLS.recursive(Y, X_mx, beta_t, dt, gamma_inv, 1, d.stop.th = 1e-6, it.max = 100)

    LRT <- 2*h1[[2]]-2*h0[[2]] #likelihood ratio test statistic
  pval <- pchisq(LRT, 2, lower.tail = F)
    return(pval)
}

Y <- as.matrix(round(phenotypes) )
colnames(Y) <- NULL
xa_matrix <- as.matrix(genotypes)-1
xd_matrix <- 1 - 2*abs(xa_matrix)


allPvals.PC1 <- apply(rbind(xa_matrix,xd_matrix), 2, function(x) cov.logistic.IRLS.pval.recursive(Xa=x[
```

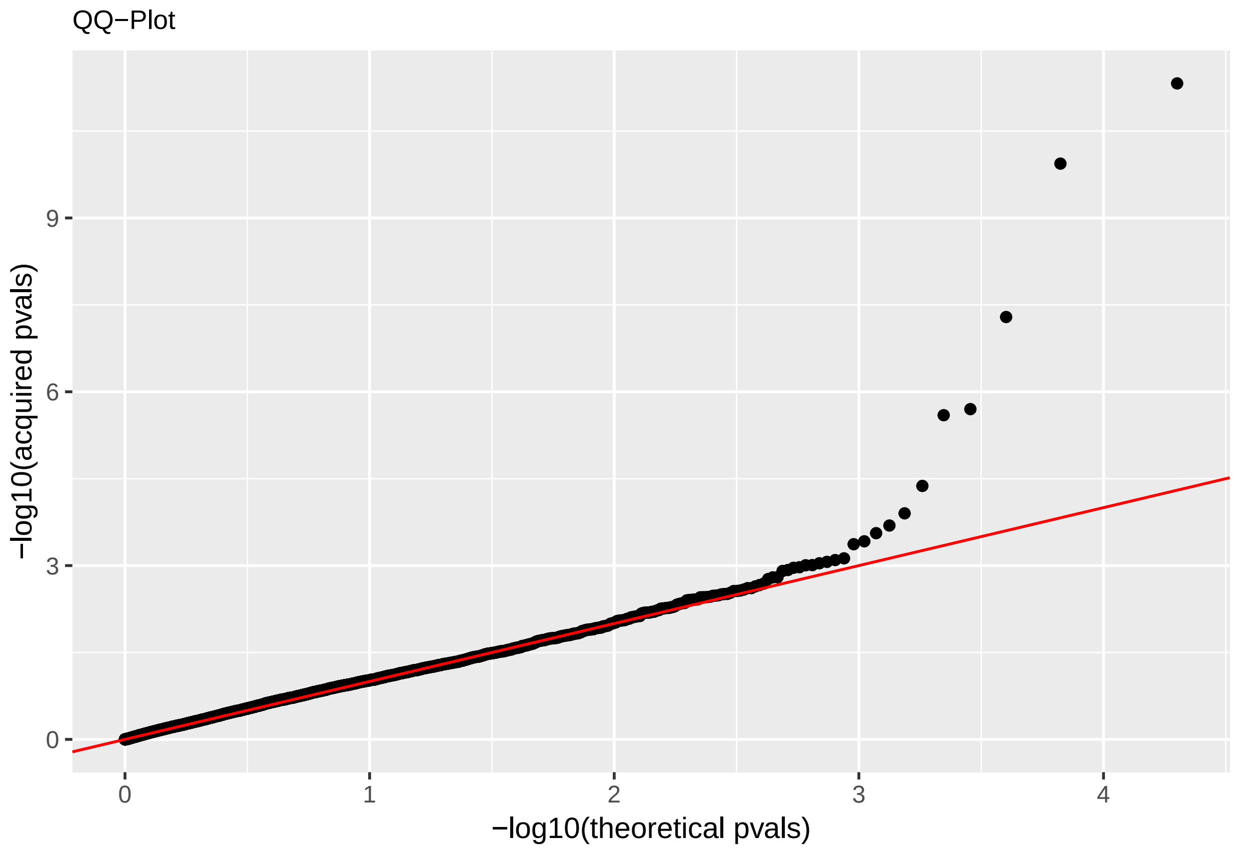
5b

```
log.allPval.PC1 <- as.data.frame(-log10(allPvals.PC1),ncol=1)
adjusted_p <- -log10(0.05/ncol(genotypes))


ggplot(log.allPval.PC1,aes(1:nrow(log.allPval.PC1),log.allPval.PC1[,1])) +
  geom_point() +
  labs(x="Variant",y="-log10(pval)",title=c("Manhattan Plot"))+
  geom_hline(yintercept=adjusted_p,color = "red")
```



5c.

```
qqDf1 <- data.frame(exp = sort(allPvals.PC1),theo = sort(qunif(ppoints(nrow(log.allPval)))))
ggplot(qqDf1,aes(-log10(theo), -log10(exp)))+geom_point()+
  geom_abline(slope = 1, color = "red")+
  labs(title="QQ-Plot",x="-log10(theoretical pvals)",y="-log10(acquired pvals)")+
  theme(plot.title = element_text(size=10))
```



6a.

```
#Bonferroni correction
adjusted_p <- 0.05/ncol(genotypes)
```

6b.

there are two peaks. They are separated by genotypes with p-values that fall below the cutoff.

7a.

(1,3) ##7b. (2,3) ##7c. (1,2)

8a

$X_a\beta_\alpha = X_a\beta_a + X_d\beta_d + \beta'_u + \epsilon$ ##8b 0

9a

A1A1B1B1 A1A2B1B1 A2A2B1B1 A1A1B1B2 A1A2B1B2 A2A2B1B2 A1A1B2B2 A1A2B2B2 A2A2B2B2

9b

$X_{a,1}, X_{d,1}, X_{a,2}, X_{d,2}$ =1,-1,1,-1

9c

based on the interaction linear regression $y = \beta_u + \beta_{a,1} * X_{a,1} + \beta_{a,2} * X_{a,2} + \beta_{d,1} * X_{d,1} + \beta_{d,2} * X_{d,2} + \beta_{a1a2} * X_{a,1} * X_{a,2} + \beta_{a1d2} * X_{a,1} * X_{d,2} + \beta_{d1a2} * X_{d,1} * X_{a,2} + \beta_{d1d2} * X_{d,1} * X_{d,2} = 0.2 + 0.1 * 1 + 0.2 * (-1) + (-0.3) * 1 + 0.17 * (-1) + (-0.11) * 1 * 1 + 0.32 * 1 * (-1) + 0.08 * (-1) * 1 + (-0.03) * (-1) * (-1) = -0.91$

10a

{H,T}

10b

∅, {H,T},{H},{T},

10c

in a fair game, $Pr(H) = Pr(T) = 0.5$ $Pr(\emptyset) = 0$ $Pr(H) = Pr(T) = 0.5$ $Pr(H \cup T) = 1$

10d

define X= 1 if Head, X=0 if tail

10e

E(x) = Pr(X=1)*1+Pr(X=0)*0 = 0.5

10f

X can be 0,1,2,3,. . . .,10 11 possible outcomes

10g

An example of wrong estimator: $T = 1/N^2$

10h

$MLE(\hat{p}) = x/N$, where N is the number of total toss, x is the heads

10i

based on the static, there are 10 tests, hence: p = choose(10,8)/choose(10,5)

p= 0.178

10j

p>0.05, fail to reject