HW6

Kate Harline feat T-bone Wrighty McFlightsman

4/5/2019

Problem 1

A

The assumption of a linear regression is that $\epsilon \sim N(0, \sigma^2)$ regardless of the distribution of the independent variables (genotypes, X). This imposes the same assumption on the dependent variable (phenotypes, Y). For example, in the simplest case for one individual $\hat{y_i} = \hat{\beta_\mu} + x_{i,a}\hat{\beta_a} + x_{i,d}\hat{\beta_d} + \epsilon$ when a polymorphism is noncausal, $\hat{\beta_a} = \hat{\beta_d} = 0$ so $\hat{y_i} = \hat{\beta_\mu} + \epsilon \sim N(0, \sigma^2)$. Furhter, the distribution for the linear regression model follows $Pr(Y|X) \sim N(\beta_\mu + X_a\beta_a + X_d\beta_d, \sigma^2_\epsilon$ so the data we are testing will ideally also follow this type of distribution.

B

Correctly rejecting the null hypothesis in a GWAS signifies that the region that contains this "hit" contains the causal polymorphism, not that the "hit" itself is causal. Due to the structure of genomes and the current depth of sequencing data, a GWAS is a mapping experiment. Genomic structure and inheritance patterns dictate that through linkage disequilibrium, sites that are closer together are inherited together.

On the other hand, there is always a chance the null hypothesis was rejected incorrectly and therefore we have obtained a false positive. We control for the false positives by setting a lower alpha and doing multi-test corrections.

Problem 2

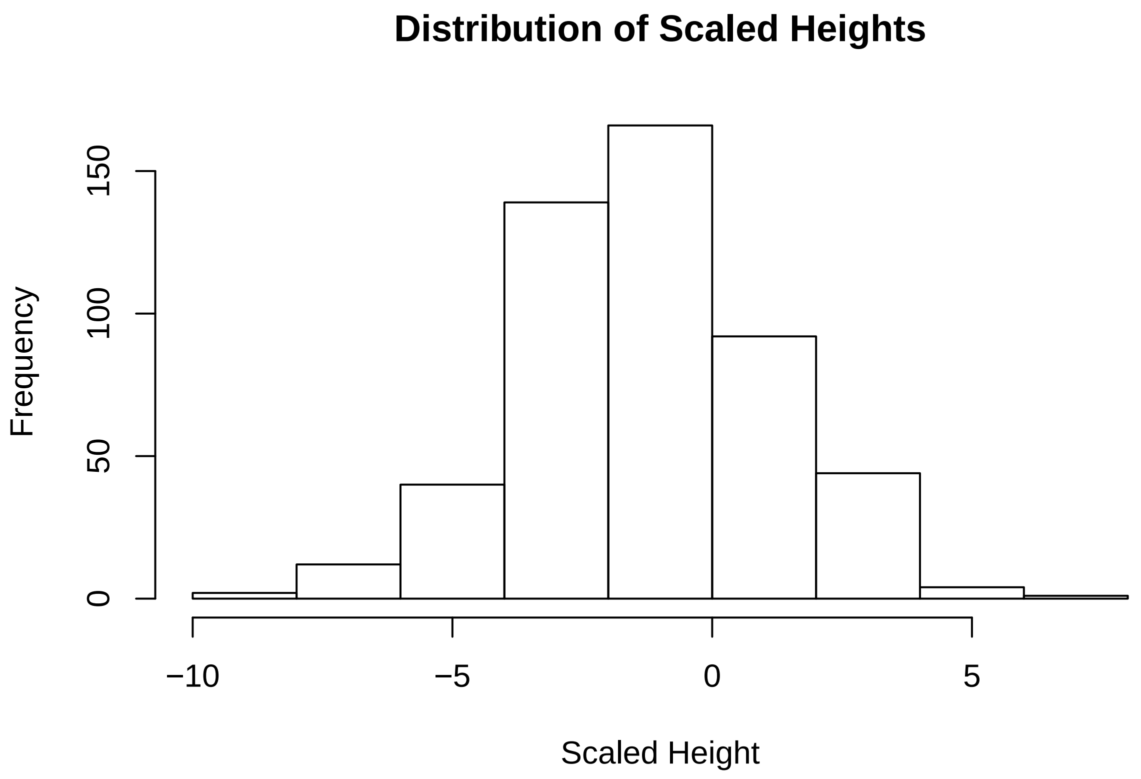
A

```r
# input the phenotype data and report the number of samples, s
phenotypes <- read.table("./QG19 - hw6_phenotypes.txt", header = F)
n <- length(phenotypes[,1])
cat('There are n = ', n, 'samples')
```

```
## There are n =  500 samples
```

B

```r
# make a histogram of the phenotype data
heights_hist <- hist(phenotypes[,1], main = 'Distribution of Scaled Heights', xlab = 'Scaled Height')
```



```
print(heights_hist)
```

```
## $breaks
##  [1] -10  -8  -6  -4  -2   0   2   4   6   8
##
## $counts
## [1]   2  12  40 139 166  92  44   4   1
##
## $density
## [1] 0.002 0.012 0.040 0.139 0.166 0.092 0.044 0.004 0.001
##
## $mids
## [1] -9 -7 -5 -3 -1  1  3  5  7
##
## $xname
## [1] "phenotypes[, 1]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

C

```
# input genotype data, report the number of genotypes N, sample size n
genotypes <- read.table("./QG19 - hw6_genotypes.txt", header = F)
N <- ncol(genotypes)
```

```
n <- nrow(genotypes) / 2
cat('There are N = ', N, ' genotypes. There are n = ', n, ' samples.')
```

```
## There are N =  1194  genotypes. There are n =  500  samples.
```

D

```
# convert genotypes into xa and xd matrices


# initialize matrix to create encodings
xa_matrix <- matrix(NA, nrow = nrow(genotypes)/2, ncol = ncol(genotypes))
# store xa encodings
for (i in 1:(ncol(genotypes))){
  loci <- matrix(genotypes[,i], nrow = nrow(genotypes)/2, ncol = 2, byrow = TRUE)
  allele_freq <- table(loci)
  minor_allele <- names(allele_freq[allele_freq == min(allele_freq)])
  xa_encoding <- ifelse(loci == minor_allele, 1, 0)
  xa_matrix[,i] <- rowSums(xa_encoding) -1
}
# then convert x_a to x_d
xd_matrix <- -2*abs(xa_matrix)+1
```

E

```
# calculate MLE of b's, calculate f statistic,
# install.packages(MASS)
library(MASS)
# Function to calculate the pval given a set of individuals' phenotype, and genotype encodings.
pval_calculator <- function(pheno_input, xa_input, xd_input){
    n_samples <- length(xa_input)

    X_mx <- cbind(1,xa_input,xd_input)

    MLE_beta <- ginv(t(X_mx) %*% X_mx) %*% t(X_mx) %*% pheno_input
    y_hat <- X_mx %*% MLE_beta

    SSM <- sum((y_hat - mean(pheno_input))^2)
    SSE <- sum((pheno_input - y_hat)^2)

    df_M <- 2
    df_E <- n_samples - 3

    MSM <- SSM / df_M
    MSE <- SSE / df_E

    Fstatistic <- MSM / MSE

    pval <- pf(Fstatistic, df_M, df_E,lower.tail = FALSE)
    return(pval)
}

# Initialize some variables and constants
n_geno <- ncol(xa_matrix)
n_pheno <- length(phenotypes)
pvals <- c()

# Calculate and save pvals for each phenotype-genotype pair

  for (i in 1 : n_geno){
      pvals <- c(pvals, pval_calculator(as.matrix(phenotypes),
                                      xa_input = xa_matrix[,i],
                                      xd_input = xd_matrix[,i]))
  }
```

F

```
# create Manhattan plot of the data
library(ggplot2)
```

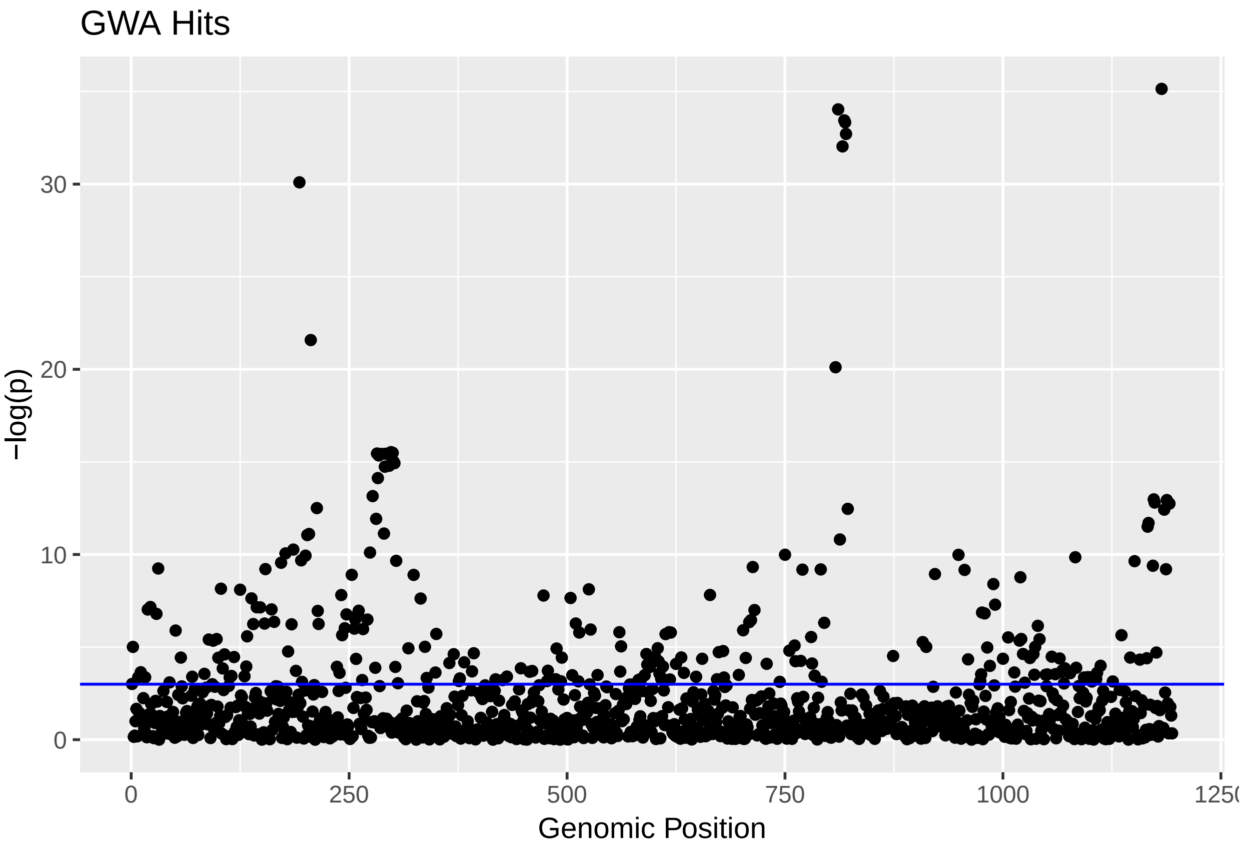
```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(gridExtra)

neg_log_p <- -1 * log(pvals)
x <- seq(1, length(neg_log_p))
y <- neg_log_p

plot1 <-ggplot(data = data.frame(x, y), mapping = aes(x, y)) + geom_point() + geom_hline(yintercept=-log

print(plot1)
```

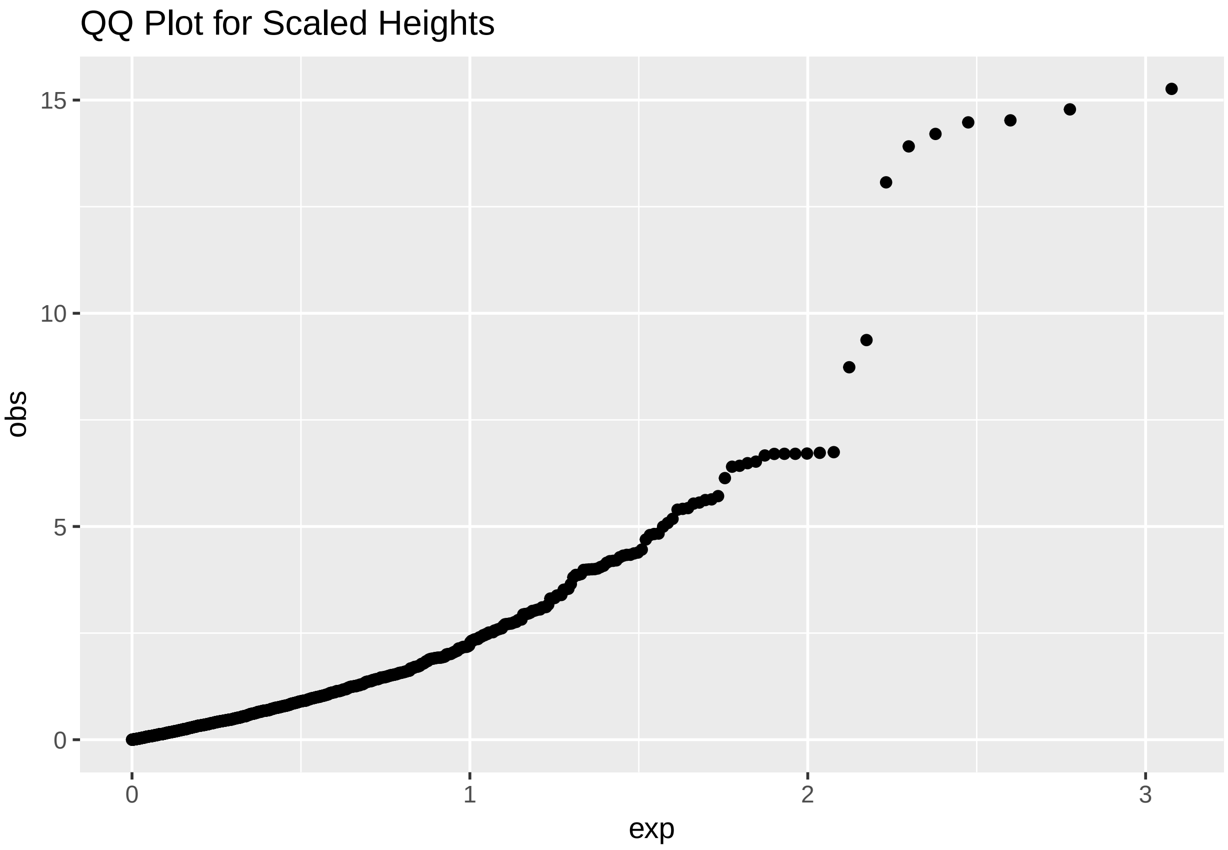


G

```
# create QQ plot of the data
# plot expected v observed
obs <- sort(-log10(pvals))
# generate range
exp <- seq(from = (1 / length(obs)), to = 1, length.out = length(obs))
# - log 10
exp <- sort(-log10(exp))

plot2 <-ggplot(data = data.frame(exp, obs), mapping = aes(exp, obs)) + geom_point() +  labs(title = 'QQ

print(plot2)
```



H

The QQ plot for the data is still pretty squiggly. The line is not as straight as would be ideal for interpreting the significance of hits. We would expect the observed p-values to track linearly for the most part with normally distributed p-values of the same size at least until the suspected causal polymorphisms. However, we do not see this. Likely covariates need to be added to the model in order to correctly interpret the data.

I

```r
# use bonferroni correction to reject the null with alpha = 0.5,
r <- sum(pvals < 0.05/ncol(xa_matrix))
no_b <- sum(pvals < 0.05)

cat('With a Bonferroni correction, the null was rejected ', r, ' times. ')
```

```
## With a Bonferroni correction, the null was rejected  38  times.
```

```r
cat('Without the Bonferroni correction, the null was rejected ', no_b, ' times.')
```

```
## Without the Bonferroni correction, the null was rejected  247  times.
```

J

I would estimate with the Bonferroni correction, we are only actually measuring about 3 main causal genotypes as there are only 3 major towers in the Manhattan plot. There may be a few others where there are hillier peaks. I would need to see some LD block data in order to have a better sense of the width of genomic regions in which the SNPs that were measured are linked. It's likely that we are able to reject the null for SNPs that are in linkage disequilibrium and thus are only mapping significance to one area of the genome. In fact, it's likely that they are all surrounding the one to few actually causal polymorphisms that we may or may not have measured in the current dataset.

Problem 3

Given the F-statistic, $F_{[2,n-3]} = \frac{\frac{SSE(\hat{\theta}_0)-SSE(\hat{\theta}_1)}{2}}{\frac{SSE(\hat{\theta}_1)}{n-3}}$ Which can be simplified, $F_{[2,n-3]} = \frac{\frac{1}{2}SSE(\hat{\theta}_0)-SSE(\hat{\theta}_1)}{\frac{1}{n-3}SSE(\hat{\theta}_1)}$

$F_{[2,n-3]} = \frac{n-3}{2}[\frac{SSE(\hat{\theta}_0)}{SSE(\hat{\theta}_1)} - 1]$ Such that $\frac{2}{n-3}F_{[2,n-3]} + 1 = \frac{SSE(\hat{\theta}_0)}{SSE(\hat{\theta}_1)}$ and the Likelihood for a linear regression, $L(\beta, \sigma_\epsilon^2, x|y) = \frac{1}{2\pi^{\frac{n}{2}}\sigma_\epsilon^n}e^{\frac{-1}{2\sigma_\epsilon^2}\sum(y_i-\hat{y_i})^2}$ Which contains the "Sum of Squares of the Errors" as $L(\beta, \sigma_\epsilon^2, x|y) = \frac{1}{2\pi^{\frac{n}{2}}\sigma_\epsilon^n}e^{\frac{-1}{2\sigma_\epsilon^2}SSE(\hat{\theta})}$ Plugging this into the Likelihood Ratio test $\Lambda = \frac{L(\hat{\theta_0})}{L(\hat{\theta_1})}$

$\Lambda = \frac{\frac{1}{2\pi^{\frac{n}{2}}\sigma_\epsilon^n}e^{\frac{-1}{2\sigma_\epsilon^2}SSE(\hat{\theta}_0)}}{\frac{1}{2\pi^{\frac{n}{2}}\sigma_\epsilon^n}e^{\frac{-1}{2\sigma_\epsilon^2}SSE(\hat{\theta}_1)}}$ So we can see that $\Lambda = (\frac{2}{n-3}F_{[2,n-3]} + 1)^{-n/2}$