# Lab 1: Linear models for quantitative genetics

## BMI 206

### Chris Thompson

### Oct 18th 2024

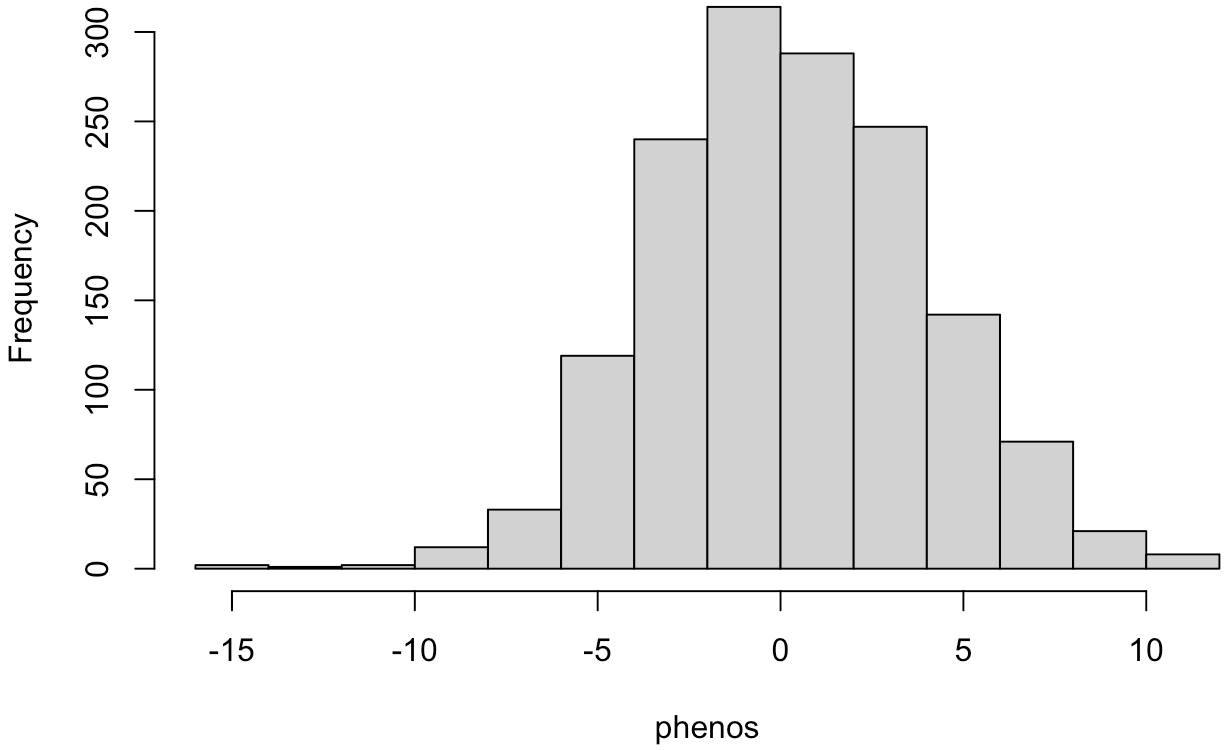
### PART1: Analyzing provided genotype and phenotype data.

**Prepare the data.** Read in the genotype and phenotype matrices.

```
genos = as.matrix(read.table("./genos.txt"))
phenos = as.matrix(read.table("./phenos.txt"))
```

Make a histogram of the phenotypes. Do they look normally distributed?

```
hist(phenos)
```




Yes, the histogram looks roughly normally distributed with mean around 0.

How are the genotypes encoded?

```
table(genos)
```

```
## genos
##       0       1       2
## 4773842 5447131 4779027
```

Genotypes are encoded as factors, with the value being number of some allele B. E.g. 0 would be 0 copies of allele B, 1 would be 1 copy of allele B, and 2 being 2 copies of allele B.

How many individuals are there in the dataset and how many SNPs? (Save them in `N` and `M`, respectively.)

```
dim(genos)
```

```
## [1]  1500 10000
```

```
dim(phenos)
```

```
## [1] 1500    1
```

```
N = 1500 # Individuals
M = 10000 # SNPs
```

1,500 individuals, 10,000 SNPs.

**Compute the *minor* allele frequency for every SNP. Check MAFs are <0.5.**

```
MAFs = array(0,M)
for(i in 1:M) {
      totalAlleles = sum(genos[,i])
      geneFreq = totalAlleles/(N*2)
      if (geneFreq > 0.5) {
        geneFreq = 1 - geneFreq
      }
      MAFs[i] = geneFreq
}
MAFs[1:10]
```

```
##  [1] 0.1516667 0.3226667 0.4126667 0.2156667 0.4626667 0.4826667 0.4863333
##  [8] 0.1516667 0.2990000 0.3310000
```

```
max(MAFs)
```

```
## [1] 0.5
```

**Run a GWAS under an additive model and save the p-values, z-scores, and effect sizes.**

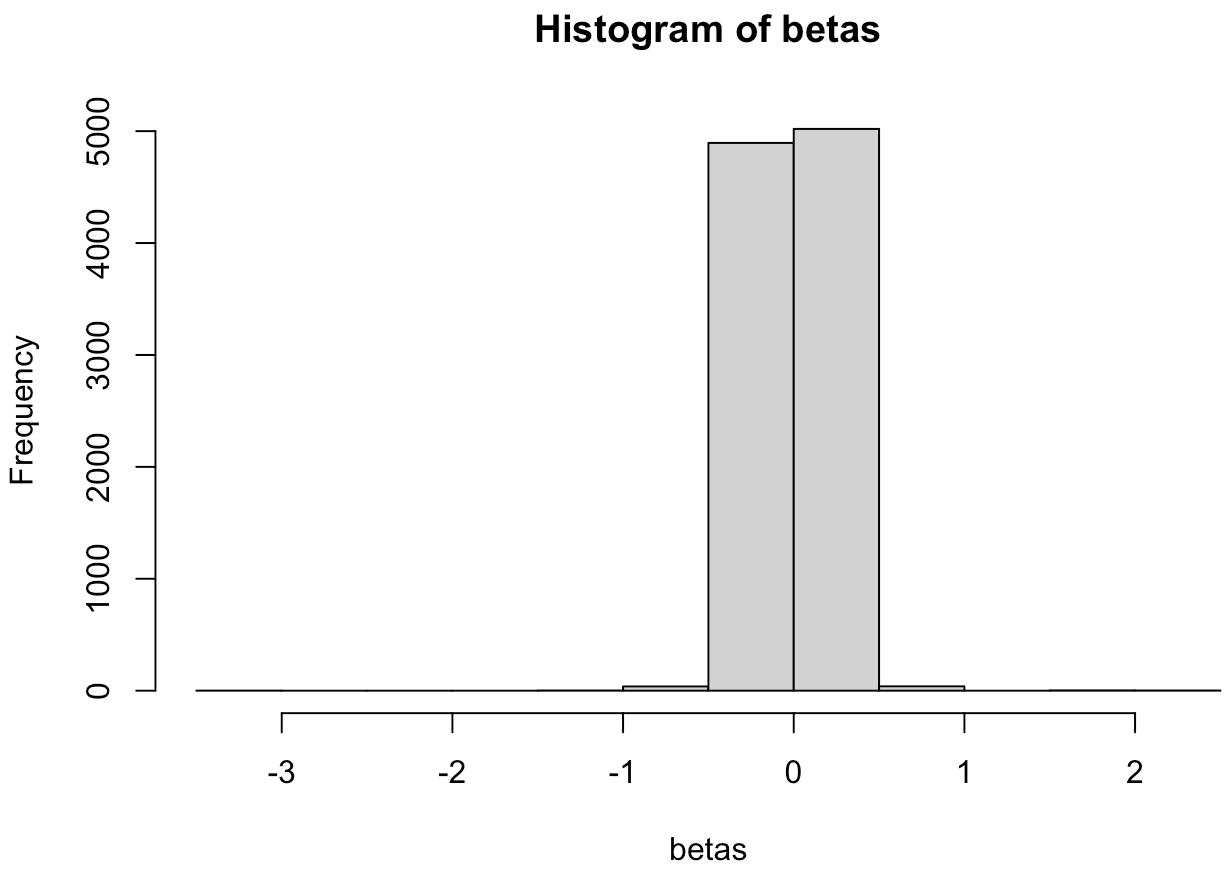
```
pvalues = array(0,M)
zscores = array(0,M)
betas = array(0,M)
for(i in 1:M) {
    g = genos[,i]
    res = summary(lm(phenos~g))
    zscores[i] = res$coefficients["g", "Estimate"] / res$coefficients["g", "Std. Error"]
    pvalues[i] = res$coefficients["g", "Pr(>|t|)"]
    betas[i] = res$coefficients["g", "Estimate"]
}
```

Summarize the effect sizes.

```
summary(betas)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.400728 -0.105822  0.002475  0.004204  0.113524  2.245525
```

```
hist(betas)
```




The overwhelming majority of effect sizes are between 0.5 and -0.5 (1st and 3rd quantiles being around -0.1 and 0.1), and the effect sizes are centered around 0. Because the data is so tightly centered around the mean and the bucket widths of the histogram are large it's hard to determine if the data is normally distributed, but I'd believe it from what we see.

Are there any significantly associated SNPs? If so, which SNPs are they?

```
assoc = which(p.adjust(pvalues, method="bonferroni") <= 0.05/M)
assoc
```

```
## [1] 1 2 3 5 6 7 8 9
```

8 SNPs total are significantly associated after performing a bonferroni correction on the data (because an experiment with p-value of 0.05 done 10,000 times would, by chance, give 500 "significant" SNPs). These SNPs are the first 9 with the exception of the 4th SNP.

How big are their effect sizes? How significant are they?

```
betas[assoc]
```

```
## [1] -3.400728  2.134040  1.523892 -1.478701  1.985246 -1.222523  2.245525
## [8]  1.860513
```

```
zscores[assoc]
```

```
## [1] -21.021900  16.013141  11.615036 -11.436804  16.128209  -9.115941  12.602187
## [8]  13.341784
```

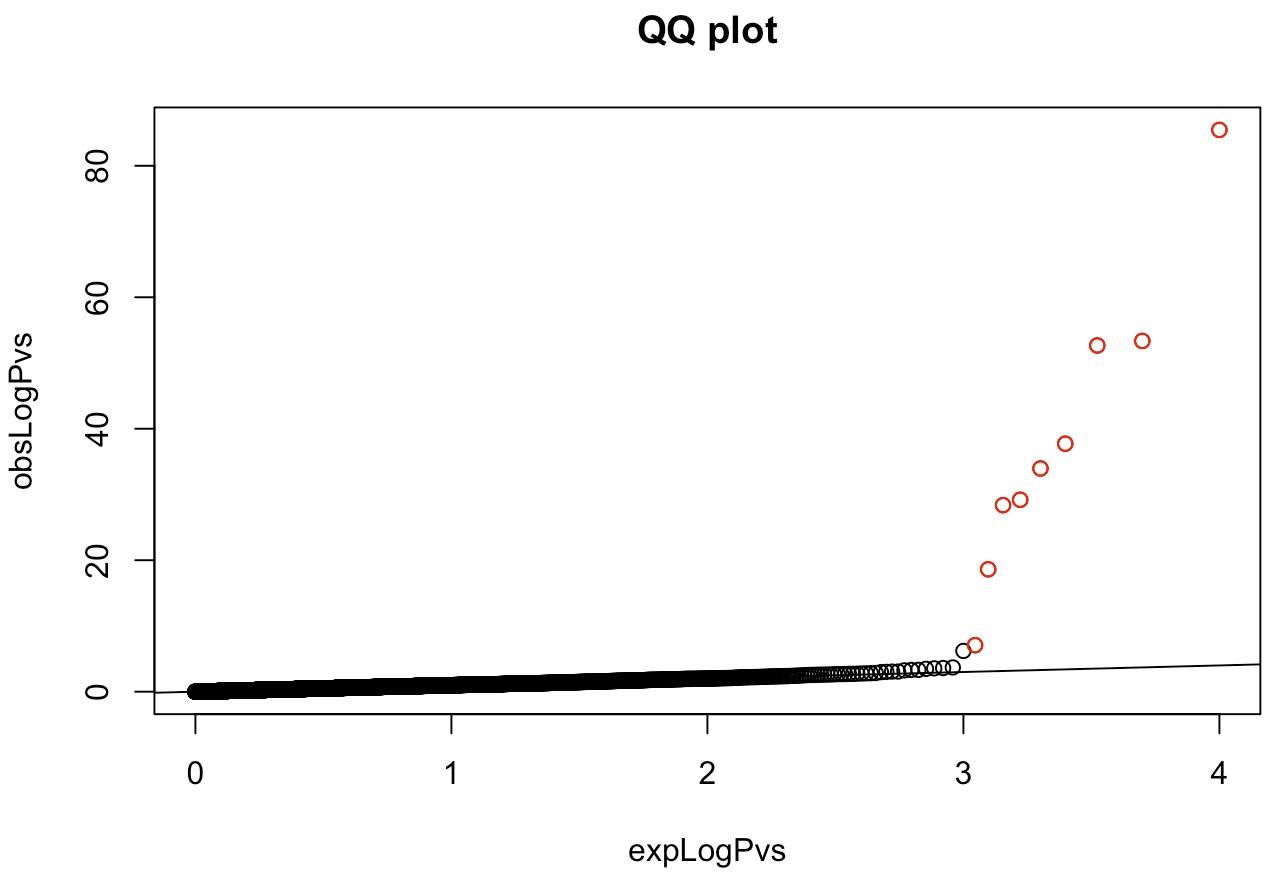
```
pvalues[assoc]
```

```
## [1] 3.488001e-86 2.148840e-53 6.341996e-30 4.244077e-29 4.411319e-54
## [6] 2.434694e-19 1.103317e-34 1.884171e-38
```

The effect sizes are between -3.400728 and 2.245525 (given the phenotype range, this seems significant in terms of impact). The statistical significance varies between p = 2.43e-19 and p = 3.48e-86, which are obviously very different but all (especially excluding 2.43e-19) are comfortably above the significance threshold.

Draw a QQ plot for log10(p) values.

```
obsLogPvs = sort(-log10(pvalues))
expLogPvs = sort(-log10(seq(1/M,1,1/M)))
plot(expLogPvs,obsLogPvs,main='QQ plot')
abline( a=0, b=1 )
#label the significant SNPs red
points(expLogPvs[(M-length(assoc)):M],obsLogPvs[(M-length(assoc)):M],col="red")
```




Is there inflation? Use the chi-square statistics to check.

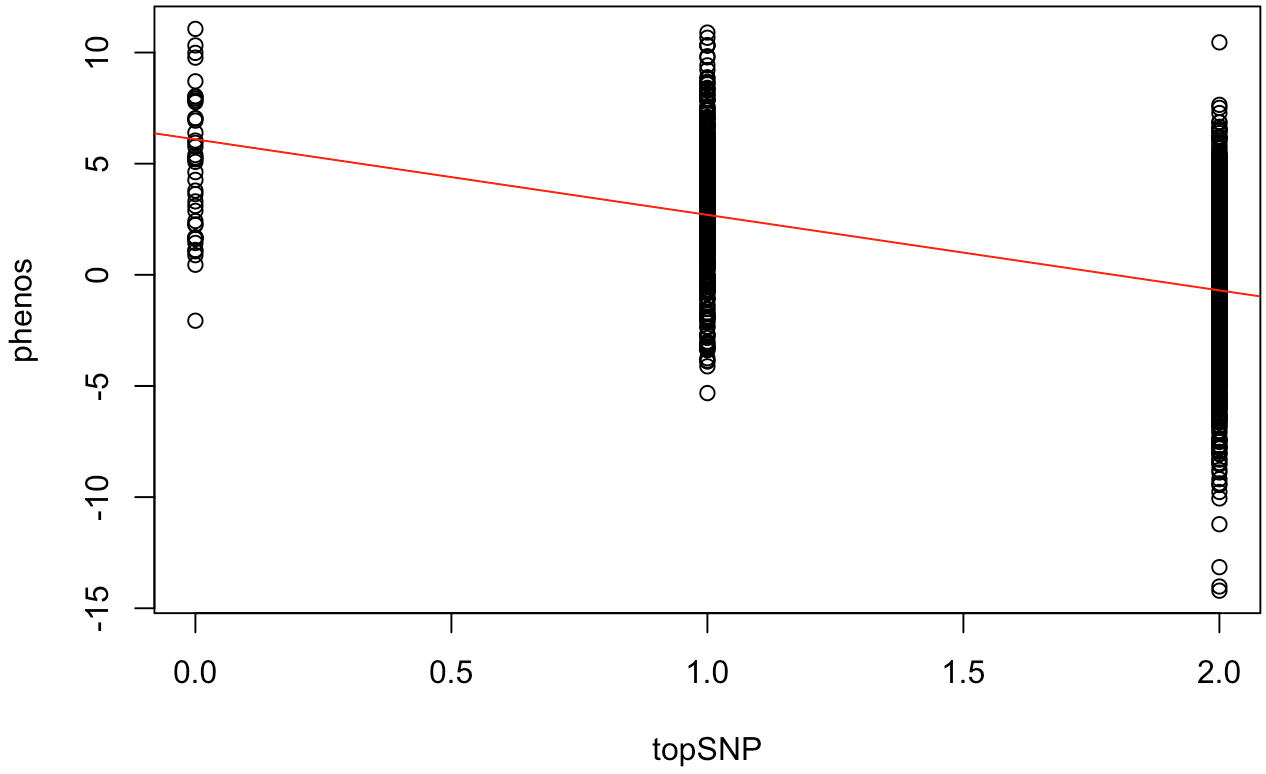
```
chis = zscores^2
lambdaGC = median(chis)/0.454 # why .454? (qchisq(0.5, df=1))
lambdaGC
```

```
## [1] 1.007873
```

Yes, but since severe inflation only occurs at the last 10 or so points (out of 10,000), overall it's pretty good.

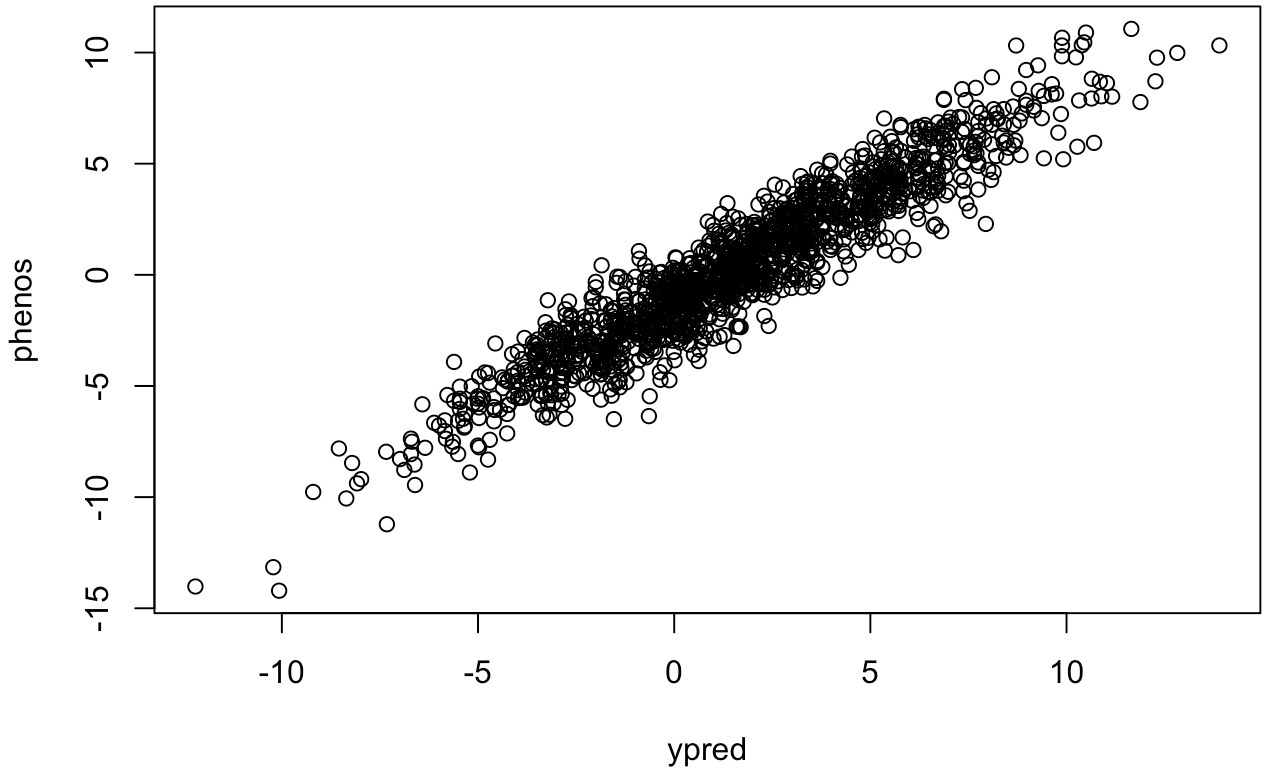
Plot the phenotype predictions for the most significant SNP.

```
topSNP = genos[,order(pvalues)[1]]
plot(topSNP,phenos)
abline(lm(phenos~topSNP)$coeff,col="red")
```




**Build a linear predictor of the phenotype using the associated SNPs.**

```
ypred = array(0,N)
beta0 =
for(i in 1:N) {
      ypred[i] = genos[i,assoc] %*% betas[assoc]
}
plot(ypred,phenos)
```




What is the correlation between the predicted phenotype and the true phenotype?

```
cor(ypred,phenos)
```

```
##             V1
## [1,] 0.9451631
```

The correlation is 0.9451631, which if I may editorialize, is pretty good.

**BONUS: Test each of the associated SNPs for non-linearity.**

```
hp = array(0,length(assoc))
for (i in 1:length(assoc)) {
  g = genos[,assoc[i]]
  h = g
  h[h==2]=0
  #Hint: can use anova(lm(?),lm(?)) or summary(lm(?))
  hp[i] <- anova( lm(?), lm(?) )$Pr[2] #skip multiple test correction for now
}
hp
```

BONUS: Visualize a linear SNP and a non-linear SNP.

```
par( mfrow=c(1,2) )
plot( ?, ? )
points( c(0,1,2), tapply( ?, ?, mean ), col=2, pch=16, cex=3 )
lines( c(0,1,2), tapply( ?, ?, mean ), col=2, lwd=2  )
plot( ?, ? )
points( c(0,1,2), tapply( ?, ?, mean ), col=2, pch=16, cex=3 )
lines( c(0,1,2), tapply( ?, ?, mean ), col=2, lwd=2  )
```

**Repeat the GWAS to test for recessive rather than additive genetic effects.**

```
genos2 = genos
genos2[genos>1]=1 # Modified because 0/1 binary makes more sense (to me).
pvalues2 = array(0,M)
zscores2 = array(0,M)
betas2 = array(0,M)
for(i in 1:M) {
  g = genos2[,i]
  res = summary(lm(phenos~g))
    zscores2[i] = res$coefficients["(Intercept)", "Estimate"] /
                  res$coefficients["(Intercept)", "Std. Error"]
    pvalues2[i] = res$coefficients["(Intercept)", "Pr(>|t|)"]
    betas2[i] = res$coefficients["(Intercept)", "Estimate"]
}
```

**Are the same SNPs significant or not?**

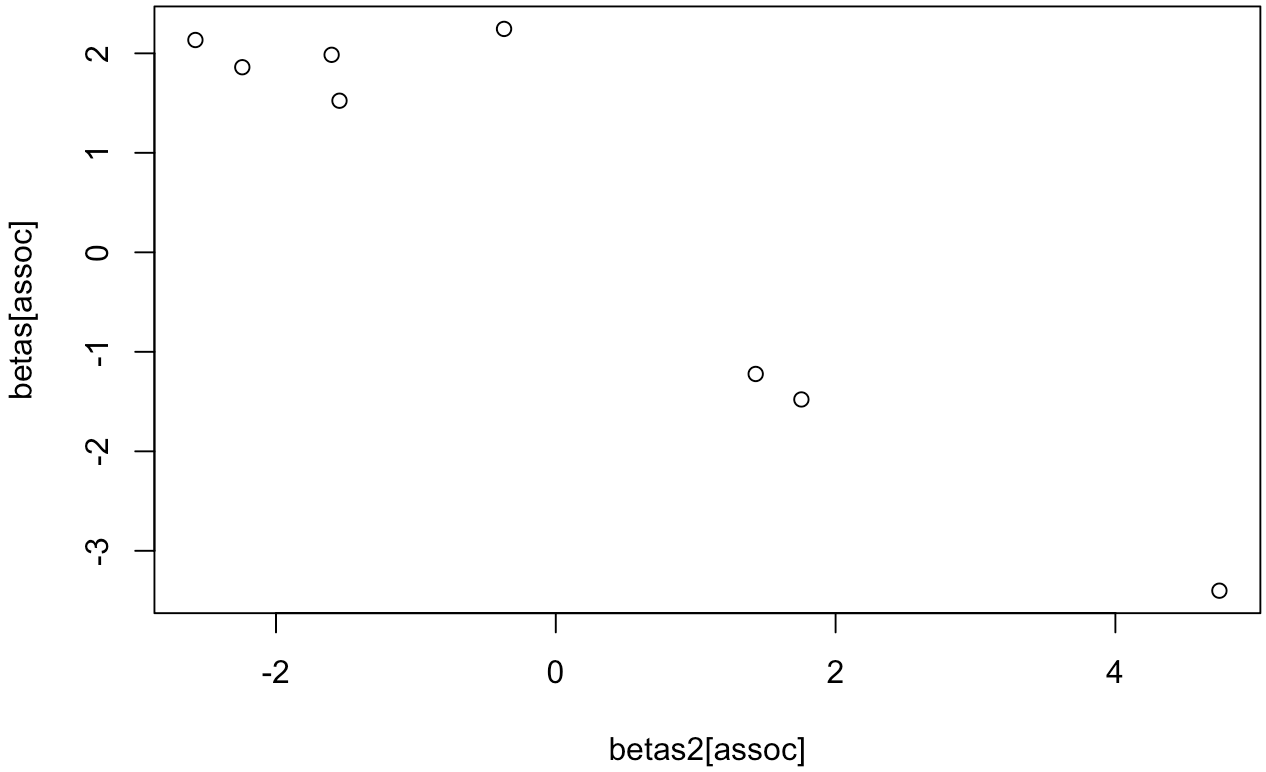
```
assoc2 = which(p.adjust(pvalues2, method="bonferroni") <= 0.05/M)
# assoc2 = which(pvalues2<?)
assoc2
```

```
## [1] 1 2 3 5 6 7 9
```

Mostly, although SNP 8 is no longer included, so the remaining significant SNPs are 1, 2, 3, 5, 6, 7, 9.

**How did the effect sizes change?**

```
plot(betas2[assoc],betas[assoc])
```




They've mostly reversed signs but otherwise are similar, e.g. SNP 1 has gone from 4.7 to -3.4. Others are more pronounced, such as SNP 4 which went from 0.79 to -0.47, which is close to doubling (but also SNP 4 lost significance with the recessive study).

# PART2: Simulating genotypes with LD.

**Establish some important simulation parameters.**

```
N = 1000 #number of individuals
M = 30   #number of non-causal SNPs
gs = matrix(0,nrow=N,ncol=M)
```

**Simulate a GWAS data set.** First, simulate the causal variant.

```
set.seed = (42) #set random seed so we all get the same numbers
MAF = 0.5
gC = rbinom(N,1,MAF) #causal variant
```

Then, simulate the phenotypes given the causal variant.

```
beta = 0.3 #association of causal variant
pheno = gC*beta + rnorm(N)
```

Generate 10 SNPS in tight LD with the causal SNP.

```
rho = 0.9
for(i in 1:10) {
  idx = rbinom(N,1,rho)
  gs[,i]=gC*idx+rbinom(N,1,MAF)*(1-idx)
  # test they have the right LD empirically
  cat( 'Observed LD = ', cor( gs[,i], gC ), '\n' )
  # Bonus: prove they have the right LD theoretically
}
```

```
## Observed LD =  0.9020675
## Observed LD =  0.9120382
## Observed LD =  0.9076052
## Observed LD =  0.8800695
## Observed LD =  0.9197995
## Observed LD =  0.8915558
## Observed LD =  0.9019553
## Observed LD =  0.9166066
## Observed LD =  0.8998077
## Observed LD =  0.8955723
```

Do the same for 10 moderate LD partners (rho=0.6).

```
rho = 0.6
for(i in 11:20) {
  idx = rbinom(N,1,rho)
  gs[,i]=gC*idx+rbinom(N,1,MAF)*(1-idx)
  # test they have the right LD empirically
  cat( 'Observed LD = ', cor( gs[,i], gC ), '\n' )
}
```

```
## Observed LD =  0.5815334
## Observed LD =  0.6182154
## Observed LD =  0.5993123
## Observed LD =  0.5704789
## Observed LD =  0.609425
## Observed LD =  0.5880787
## Observed LD =  0.648607
## Observed LD =  0.6298397
## Observed LD =  0.5471717
## Observed LD =  0.5948462
```

Do the same for 10 independent SNPs (rho=0).

```
rho = 0
for(i in 21:30) {
  idx = rbinom(N,1,rho)
  gs[,i]=gC*idx+rbinom(N,1,MAF)*(1-idx)
  # test they have the right LD empirically
  cat( 'Observed LD = ', cor( gs[,i], gC ), '\n' )
}
```

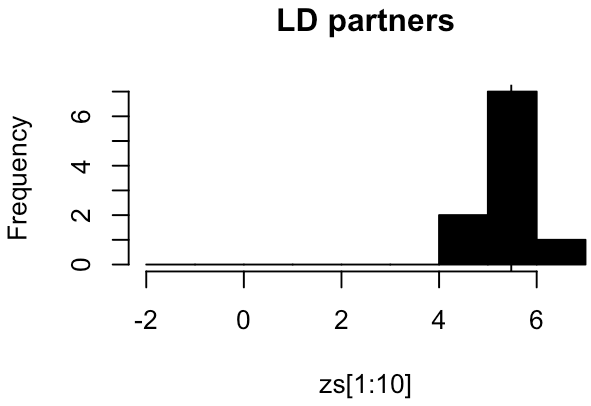
```
## Observed LD =  0.05159542
## Observed LD =  -0.005548971
## Observed LD =  0.03897972
## Observed LD =  -0.007503623
## Observed LD =  0.02967737
## Observed LD =  -0.01390053
## Observed LD =  -0.005884075
## Observed LD =  0.03897972
## Observed LD =  -0.01169901
## Observed LD =  -0.06549754
```

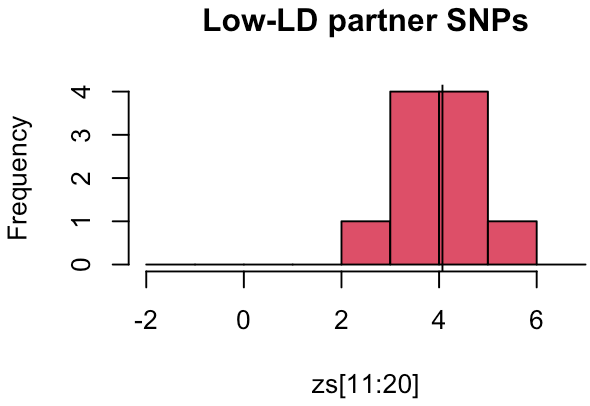
**Run GWAS on the causal variant. Then run GWAS on the other variants. Keep track of the zscores only.**

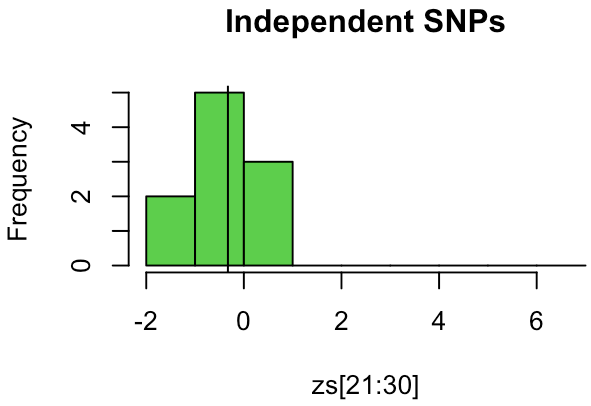
```
zsC = summary(lm(pheno~gC))$coef[2,3]
zs = sapply( 1:M, function(i) summary(lm(pheno~gs[,i]))$coef[2,3] )
```

Visualize the relationship between the mean z-scores at the tag SNPs and the z-score at the causal SNP.

```
par( mfrow=c(2,2) )
breaks = hist(c(0,zsC,zs),plot=F)$breaks
hist(zs[1:10],breaks=breaks, col=1, main='LD partners')
abline(v=mean(zs[1:10]))
hist(zs[11:20],breaks=breaks, col=2, main='Low-LD partner SNPs')
abline(v=mean(zs[11:20]))
hist(zs[21:30],breaks=breaks, col=3, main='Independent SNPs')
abline(v=mean(zs[21:30]))
```




**BONUS: Perform LD score regression. First, calculate the LD scores. There should be M+1 of them.**

```
ldscores = ?
ldscores
```

BONUS: Visualize LD score regression.

```
chis = c( ?, ? )^2
plot( ?, chis, ylab=expression(chi^2) )
#test for inflation
lambdaGC = median(chis)/0.454
lambdaGC
```

BONUS: Estimate heritability.

```
summary( lm( ? )$coef[2,1] * M/N
```

BONUS: What is the true heritability?

```
var(?) / var(?)
```