Part C

Quantitative genetics of disease

1 Single locus disease model

Parameters for single locus disease risk model

```
> v <- 0.5                          # Proportion of cases in sample
> p <- 0.2                          # Frequency of risk allele A
> R <- 1.2                          # Heterozygote genotype relative risk; homozygote GRR is R^2
> K <- 0.01                         # Overall risk (or probability) of disease in population
```

Parameters for power calculation

```
> N <- 10000                        # Total sample size
> alpha <- 5e-8                     # Level of significance that will be used in the association study
> t <- qnorm(alpha/2,0,1)           # Normal distribution threshold for declaring significance
```

Question. Derived single locus parameters ($f0$ and PG)

```
> f0 <- K/(1+p*(R-1))^2             # Probability of disease in those homozygote for non-risk allele aa
> PG <- c((1-p)^2, 2*p*(1-p), p^2)  # Probability of genotypes aa, Aa, AA assumes HW equilibrium
```

Question. Calculate p_{case}, allele frequency in cases

```
> PDgivG <- c(f0, f0*R, f0*R^2)     # Probability of disease given the genotype
> PDandG <- PDgivG*PG               # Probability of  genotype and disease
> # PDandG <- c(f0*(1-p)^2,f0*R*2*p*(1-p),f0*R^2*p^2)
> sum(PDandG); K                    # Check equals K
```

```
[1] 0.01
```

```
[1] 0.01
```

```
> PGgivD <- PDandG/K                # Probability of genotype in people who are diseased -
>                                   # compare PGgivD to PG
> pcase <- 0.5*PGgivD[2] + PGgivD[3]  # Frequency of allele A in cases
```

Question. Calculate $p_{control}$, allele frequency in controls

```
> PNDgivG <- 1 - PDgivG               # Probability of being not diseased given the genotype
> PGandND <- PNDgivG * PG             # Probability of  genotype and not diseased
> sum(PGandND); (1-K)               # Check equals 1-K
```

```
[1] 0.99
```

```
[1] 0.99
```

```
> PGgivND <- PGandND / (1-K)          # Probability of genotype in not diseased - compare PGgivND + PG
> pcont <- 0.5 * PGgivND[2] + PGgivND[3]# Frequency of allele A in controlsp
> #pcontchk = (p/(1-K))*(1-K*R/(1+p*(R-1)))
> # Checks
> p;K*pcase + (1-K)*pcont             # Check that population weighted average of pcase and pcontrol is p
```

```
[1] 0.2
```

```
[1] 0.2
```

Question. Calculate the Odds Ratio

```
> pcase-pcont
```

```
[1] 0.03108003
```

```
> OR = (pcase/(1-pcase)) / (pcont/(1-pcont))
> OR = (pcase/pcont) / ((1-pcase)/(1-pcont)) # equivalent
> OR
```

```
[1] 1.202335
```

2 Power

```
> pbar <- v*pcase + (1-v)*pcont       # Mean allele frequency in case-control sample
> NCP <- (pcase-pcont) * (pcase-pcont) * 2 * N * v * (1-v) / (pbar*(1-pbar))
>                                   # Chi-square non-centrality parameter
> pow <- pnorm(sqrt(NCP)+t)         # Power - normal distribution is sqrt of chi-square
> pow; NCP                          # Check agrees with genetic power calculator:
```

```
[1] 0.4586288
```

```
[1] 28.59492
```

```
>
> # N <- 10000,v<-0.5,p<-0.2,R<-1.2,K<-0.01,alpha<-5e-8,Dprime<-1
```

Question. Use the Genetic Power Calculator http://zzz.bwh.harvard.edu/gpc/cc2.html and compare the power to the direct calculation as coded above
$Power < -0.4586; NCP < -28.59$

Genetic Power Calculator

Case - control for discrete traits

High risk allele frequency (A)	:	0.2	(0 - 1)
Prevalence	:	0.01	(0.0001 - 0.9999)
Genotype relative risk Aa	:	1.2	(>1)
Genotype relative risk AA	:	1.44	(>1)
D-prime	:	1	(0 - 1)
Marker allele frequency (B)	:	0.2	(0 - 1)
Number of cases	:	5000	(0 - 10000000)
Control : case ratio	:	1	(>0)
			(1 = equal number of cases and controls)
		☐	Unselected controls? (* see below)
User-defined type I error rate	:	5e-8	(0.00000001 - 0.5)
User-defined power: determine N (1 - type II error rate)	:	0.8	(0 - 1)

Process Reset

Created by Shaun Purcell 24.Oct.2008

***Note** : unselected controls indicates a true random population sample (e.g. for a 1% disease, 1% of controls would also, b not having the disease.*

Question. Compare power for screened and unscreened controls

```
> # N =10000,v = 0.5,p=0.2,R=1.2,K=0.01,alpha=5e-8
> pcont=?
+ pbar <- v*pcase+(1-v)*pcont       # Mean allele frequency in case-control sample
> #NCP
> NCP <- (pcase-pcont)*(pcase-pcont)*2*N*v*(1-v)/(pbar*(1-pbar)) # Chi-square non-centrality parameter
> pow <- pnorm(sqrt(NCP)+t)          # Power - normal distribution is sqrt of chi-square
> pow; NCP
>
> #K=0.01
> #screened pow  = ??? ; NCP =???
> #unscreened pow = ???; NCP =???
>
> # the code above assumes controls are screened, for unscreened controls pcont=p
```

Question. Compare the impact on power of screening controls for schizophrenia $K = 0.01$ and Major depression $K = 0.15$.
For which disorders is screening of controls most recommended?

```
> K=0.15 # rerun the relevant code
> #screened pow  = ??? ; NCP =???
> #unscreened pow = ???; NCP =???
```

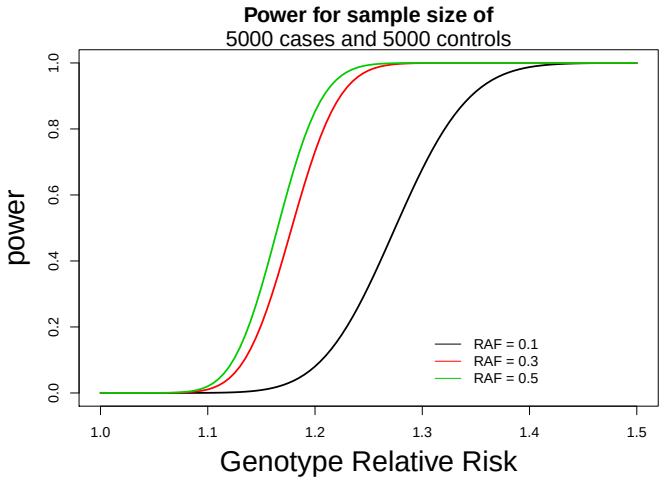
Question. Get a power graph

```
> getpower=function(p,GRR,Ncase,Ncont,A,K){
+   N=Ncase+Ncont
+   v<-Ncase/N
+   Thr<-qnorm(A/2,0,1)
+   pcase<-p*GRR/(p*GRR+1-p)
+   pcont<-(p/(1-K))*(1-K*GRR/(1+p*(GRR-1)))
+   pdiff=pcase-pcont
+   pbar<-pcase*v+pcont*(1-v)
+   #exactly the same as GPC
+   NCP<-pdiff*pdiff*2*N*v*(1-v)/(pbar*(1-pbar))
+   pow=pnorm(sqrt(NCP)+Thr)
+   pbar=0.5*(pcase+pcont)
+   Nequiv=NCP*pbar*(1-pbar)/pdiff^2   # sample size of equal cases and controls that gives same power
+   result=list(pow=pow,NCP=NCP,Nequiv=Nequiv)
+   result
+ }
> getpower(0.2,1.2,5000,5000,5e-8,0.01) #confirm same as programmed above
```

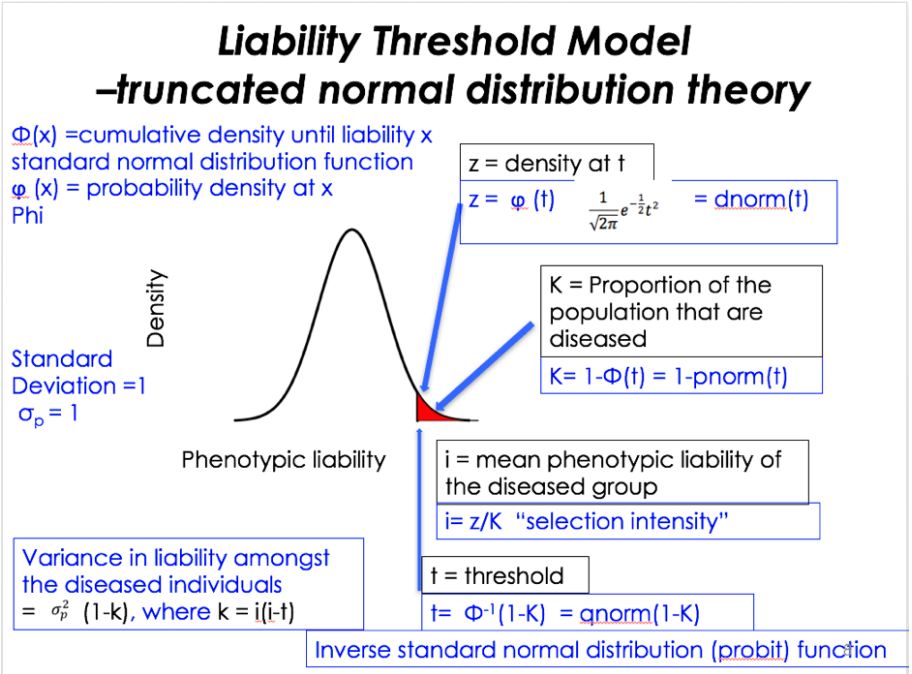
```
$pow
[1] 0.4586288

$NCP
[1] 28.59492

$Nequiv
[1] 5000
```

```
> #-------------------------
> # run this section to get a power graph
> #-------------------------
> opar<-par(mfrow=c(1,1))
> Ncase=5000
> Ncont=5000
> maxGRR=1.5
> A<-5e-8
> x<-c(1,maxGRR)
> y<-c(0,1)
> plot(x,y,xlim=c(1,maxGRR),ylim=c(0,1),xlab="Genotype Relative Risk",ylab="power",
+      main="Power for sample size of",type="n",cex.lab=2,cex.main=1.5)
```

```
> mtext(paste(Ncase,"cases and",Ncont,"controls"),3,0,cex =1.5)
> curve(getpower(0.1,x,Ncase,Ncont,A,K)$pow,from=1,to=maxGRR,col=1,lty=1,lwd=2,add=TRUE)
> curve(getpower(0.3,x,Ncase,Ncont,A,K)$pow,from=1,to=maxGRR,col=2,lty=1,lwd=2,add=TRUE)
> curve(getpower(0.5,x,Ncase,Ncont,A,K)$pow,from=1,to=maxGRR,col=3,lty=1,lwd=2,add=TRUE)
> legend(x=1.3,y=0.2,col=c(1:3),lty=1,legend=paste("RAF =",c(0.1,0.3,0.5)),box.col="white")
```



3 Liability Threshold Model



Exercise 1.
- In part B we simulated disease for a quantitative trait which could represent the phenotypic liability of disease. Use this code to simulate a disease trait of lifetime risk K.
- Check that the mean liability of the disease group is $i \times \sigma_p$
- Calculate the increased risk of disease in parents, full-siblings and monozygotic twins of affected individuals
- Report these risks for four disease, from all combinations of K=0.01, K=0.15, h2=0.3, h2=0.8.
- Explain the results in words.

Question. In part B we simulated disease for a quantitative trait which could represent the phenotypic liability of disease. Use this code to simulate a disease trait of lifetime risk K.

Reminder of part B.

```
> set.seed(615) # for reproducibility
> m = 500                                      # number of SNPs
> maf = runif(m, 0, .5)                        # random MAF for each SNP
> N = 400                                      # number of individuals
> x012 = t(replicate(N, rbinom(m, 2, maf)))    # n x m genotype matrix
> #
> # scaling the data
> scale_x012 = scale(x012)
> scale_x012[which(is.na(scale_x012))] = 0
> #
> h2 = 0.9  # heretability
> beta = rnorm(m, 0, sqrt(h2/m))
> g_parent = scale_x012 %*% beta
> e_parent = rnorm(N, 0, sqrt(1-h2))
> #
> # phenotype parent
> y_parent = g_parent + e_parent
```

```
> K = 0.2
> t = qnorm(1-K, 0, 1)
> D_parent = rep(0, N)
> D_parent[y_parent>t] = 1 # P: phenotype
```

Question. Check that the mean liability of the disease group is $i \times \sigma_p$

```
> mean(y_parent[D_parent==0])
> mean(y_parent[D_parent==1])
> i = dnorm(t)/K #i = z/K
>
```

Question. Calculate the increased risk of disease in parents, full-siblings and monozygotic twins of affected individuals

Using the *getOffspring* function from Part B, we now simulate $N = 400$ full-sblings and $N = 400$ monozygotic twins.

```
> fullsib = NULL # where we will store the offsprings
> for(i in 1:(N/2)){
+   # randomly pick two parents
+   parents = sample (1:N,2)
+   # generate 2 offsprings from these two parents
+   kid_1 = getOffspring(x012[parents[1], ], x012[parents[2], ])
+   kid_2 = getOffspring(x012[parents[1], ], x012[parents[2], ])
+
+   # store the new offspring with the other ones
+   fullsib = rbind(fullsib, rbind(kid_1,kid_2))
+ }
> #
> # scaling the data
> scale_fullsib = scale(fullsib)
> scale_fullsib[which(is.na(scale_fullsib))] = 0
> #
> # phenotype y_fullsib
> g_fullsib = scale_fullsib %*% beta
> e_fullsib = rnorm(N, 0, sqrt(1-h2))
> #
> y_fullsib = g_fullsib + e_fullsib
> #
> # liability of the disease
> D_fullsib = rep(0, N)
> D_fullsib[y_fullsib>t] = 1 # P: phenotype
```

For the monozygotic twins

```
> MZ = NULL # where we will store the offsprings
> for(i in 1:(N/2)){
+   # randomly pick two parents
+   parents = sample (1:N,2)
+   # generate twice the same offspring from these two parents
+   kid = getOffspring(x012[parents[1], ], x012[parents[2], ])
+
+   # store the new offspring with the other ones
+   MZ = rbind(MZ, rbind(kid,kid))
+ }
> #
> # scaling the data
> scale_MZ = scale(MZ)
> scale_MZ[which(is.na(scale_MZ))] = 0
> #
> # phenotype y_MZ
> g_MZ = scale_MZ %*% beta
> e_MZ = rnorm(N, 0, sqrt(1-h2))
> #
```

```
> y_MZ = g_MZ + e_MZ
> #
> # liability of the disease
> D_MZ = rep(0, N)
> D_MZ[y_MZ>t] = 1 # P: phenotype
```

Increased risk calculated in parents: probability that one is affected when the other is, divided by the probability to be affected when the other is not.

```
> # disease state of randomly selected pair of parents
> parents = matrix(D_parent[sample(1:N,N, replace=TRUE)],ncol=2)
> #
>
> # parent1 affected
> affected_parent1 = which(parents[,1] == 1)
> # how many parents are affected when parent1 is
> a = apply(parents[affected_parent1,],1,sum)
> # a==2: the second parent is affected when parent1 is
> risk_affected_parent1 = sum(a==2)/length(affected_parent1)
> # parent1 non-affected
> non_affected_parent1 = which(parents[,1] == 0)
> # how many parents are affected when parent1 is not
> a = apply(parents[non_affected_parent1,],1,sum)
> # a==1: the second parent is affected when parent1 is not
> risk_non_affected_parent1 = sum(a==1)/length(non_affected_parent1)
> risk_affected_parent1/risk_non_affected_parent1
```

```
[1] 1.111111
```

Increased risk calculated in fullsib: probability that one sibling is affected when the other is, divided by the probability that one sibling is affected when the other is not.

```
> per_parent = matrix(D_fullsib,ncol=2,byrow=T) # each siblings are in row
> #
>
> # sib1 affected
> affected_sib1 = which(per_parent[,1] == 1)
> # how many siblings are affected when sib1 is
> a = apply(per_parent[affected_sib1,],1,sum)
> # a==2: the second sibling is affected when sib1 is
> risk_affected_sib1 = sum(a==2)/length(affected_sib1)
> # sib1 non-affected
> non_affected_sib1 = which(per_parent[,1] == 0)
> # how many siblings are affected when sib1 is not
> a = apply(per_parent[non_affected_sib1,],1,sum)
> # a==1: the second sibling is affected when sib1 is not
> risk_non_affected_sib1 = sum(a==1)/length(non_affected_sib1)
> risk_affected_sib1/risk_non_affected_sib1
```

```
[1] 3.347826
```

Increased risk calculated in MZ: probability that one MZ is affected when the other is, divided by the probability that one MZ is affected when the other is not.

```
> per_parent = matrix(D_MZ,ncol=2,byrow=T) # each sibling are in row
> #
>
> # twin1 affected
> affected_twin1 = which(per_parent[,1] == 1)
> # how many twins are affected when twin1 is
> a = apply(per_parent[affected_twin1,],1,sum)
> # a==2: the second twin is affected when twin1 is
> risk_affected_twin1 = sum(a==2)/length(affected_twin1)
> # twin1 non-affected
> non_affected_twin1 = which(per_parent[,1] == 0)
> # how many twins are affected when twin1 is not
> a = apply(per_parent[non_affected_twin1,],1,sum)
> # a==1: the second twin is affected when twin1 is not
> risk_non_affected_twin1 = sum(a==1)/length(non_affected_twin1)
> risk_affected_twin1/risk_non_affected_twin1
```

```
[1] 16.64815
```

Question. Report these risks for four disease, from all combinations of K=0.01, K=0.15, h2=0.3, h2=0.8.
Explain the results in words.

```
Writing to file Additional Files For Students/Rcode/PartF - Quantitative Genetics of Disease.R
```