Lab: Genomic SEM

Using Genomic SEM to:
-Estimate a common factor model

-Run a user specified model

-Run a multivariate GWAS

Estimate a Common Factor Model

Step 1: Munge the summary statistics (*munge*)

Step 2: Run multivariable ld-score regression (*ldsc*)

Step 3: Estimate the common factor model (*commonfactor*)

Run a User Specified Model

Follow same **Steps 1-2** for common factor model, but specify your own model for **Step 3** (*usermodel*)

Multivariate GWAS in Genomic SEM

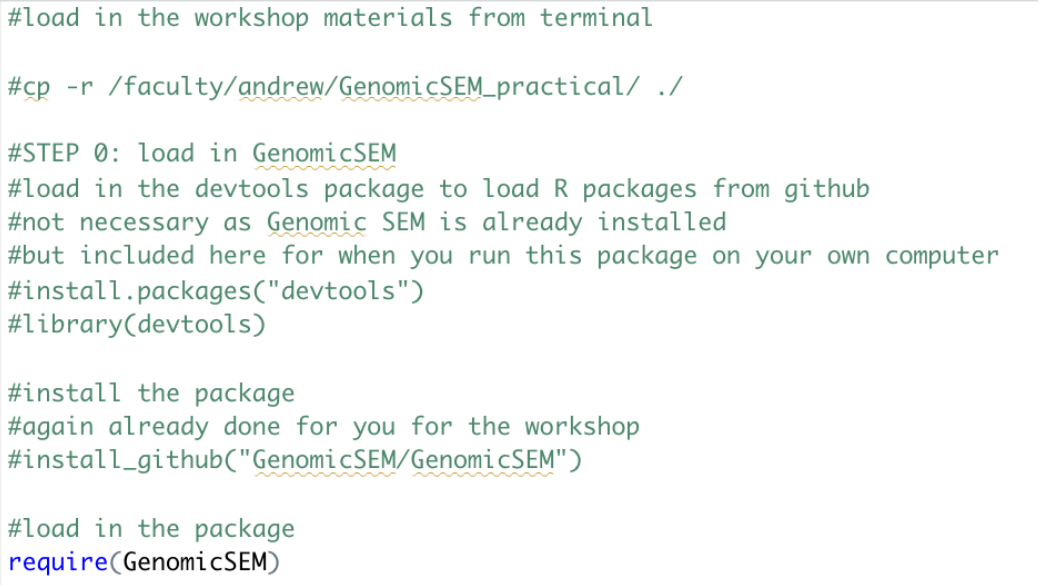
Step 1: Munge the summary statistics (*munge*)

Step 2: Run multivariable ld-score regression (*ldsc*)

Step 3: Prepare the summary statistics for GWAS (*sumstats*)

Step 4: Run the multivariate GWAS (*commonfactorGWAS* or *userGWAS*)

Step 0: Start up an R session, copy over workshop materials into your folder, and load in GenomicSEM



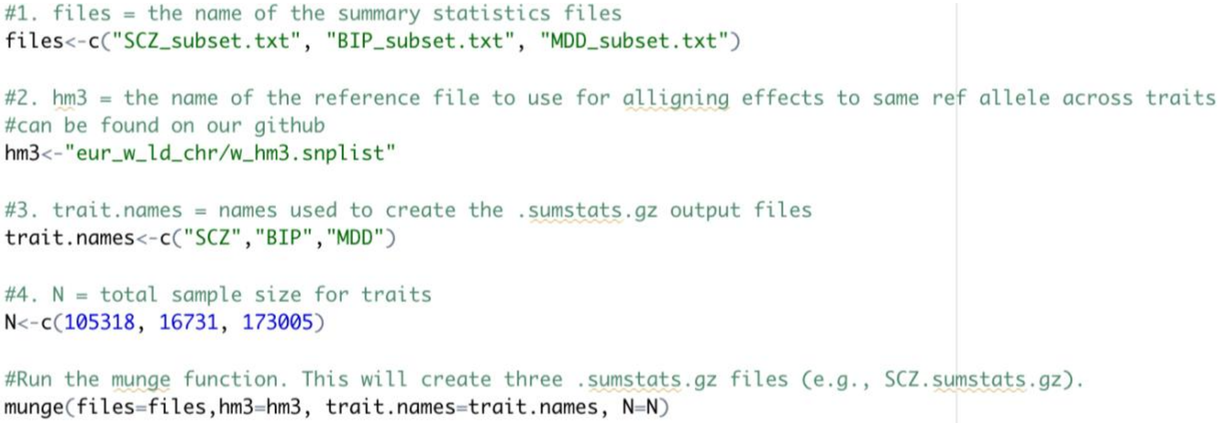
```
#load in the workshop materials from terminal

#cp -r /faculty/andrew/GenomicSEM_practical/ ./

#STEP 0: load in GenomicSEM
#load in the devtools package to load R packages from github
#not necessary as Genomic SEM is already installed
#but included here for when you run this package on your own computer
#install.packages("devtools")
#library(devtools)

#install the package
#again already done for you for the workshop
#install_github("GenomicSEM/GenomicSEM")

#load in the package
require(GenomicSEM)
```

Estimate a Common Factor Model

ACTIVITY: Fit a common factor model for three psychiatric traits: Schizophrenia, Bipolar Disorder and Major Depressive Disorder

Step 1: Munge the three provided summary statistics using the *munge* function

**Note that all code provided for Genomic SEM below is written in R.*

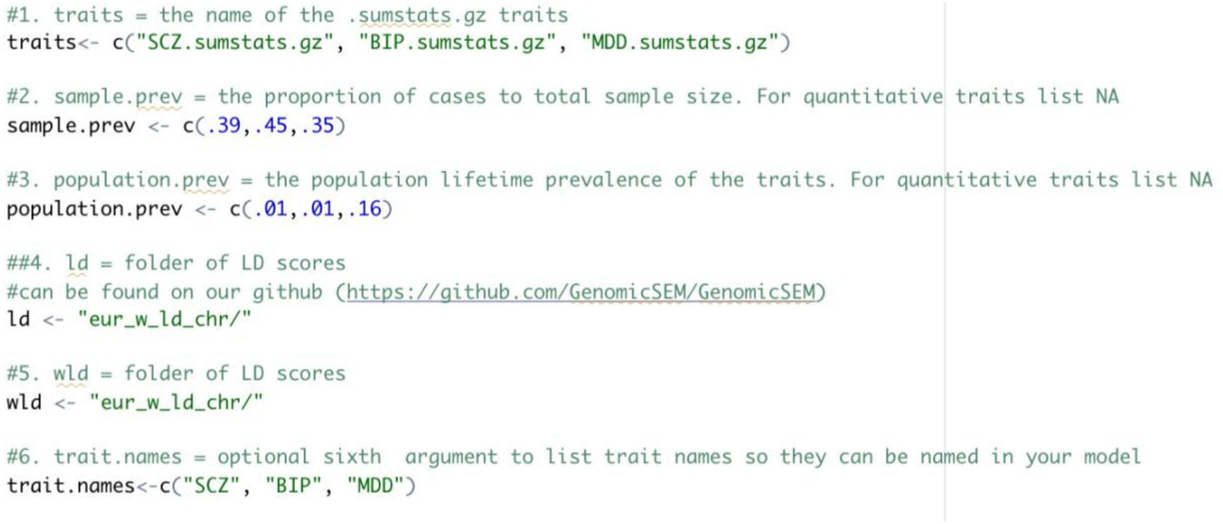


```
#1. files = the name of the summary statistics files
files<-c("SCZ_subset.txt", "BIP_subset.txt", "MDD_subset.txt")

#2. hm3 = the name of the reference file to use for alligning effects to same ref allele across traits
#can be found on our github
hm3<-"eur_w_ld_chr/w_hm3.snplist"

#3. trait.names = names used to create the .sumstats.gz output files
trait.names<-c("SCZ","BIP","MDD")

#4. N = total sample size for traits
N<-c(105318, 16731, 173005)

#Run the munge function. This will create three .sumstats.gz files (e.g., SCZ.sumstats.gz).
munge(files=files,hm3=hm3, trait.names=trait.names, N=N)
```

Now open the .log file produced by munge to answer questions below:
Checking Understanding Questions (to discuss as group):

1. How many SNPs are left for MDD?
2. How many SNPs are pruned for imputation quality (INFO score) for SCZ?
3. How many rows are removed from BIP due to low minor allele frequency (MAF)?

Step 2: Run multivariable LD-Score Regression. We specifically use the *ldsc* function from Genomic SEM as this accounts for sample overlap



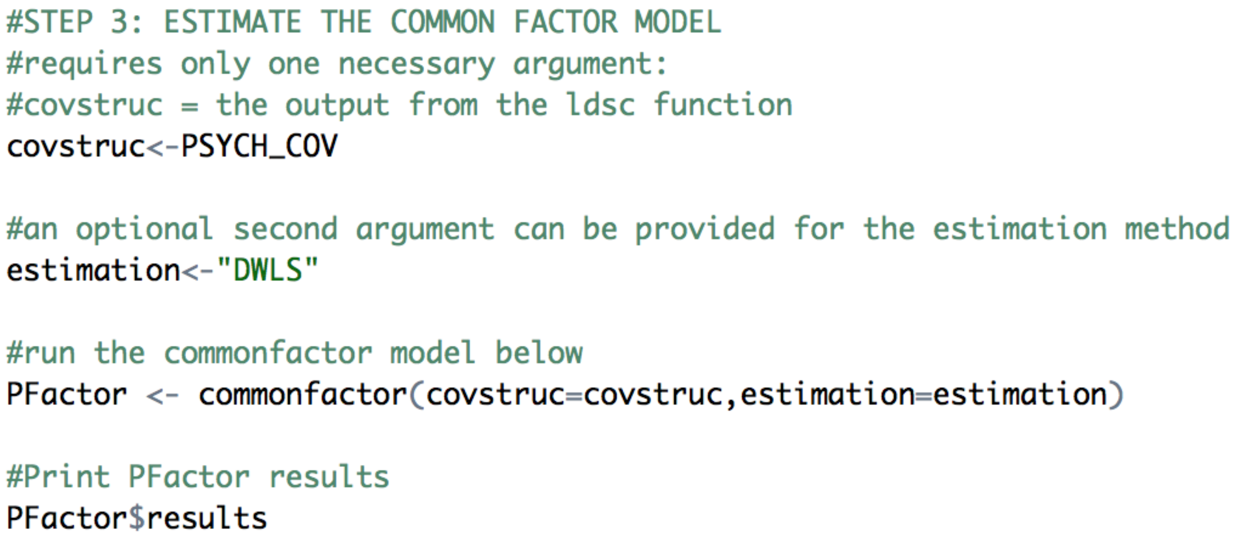
```
#1. traits = the name of the .sumstats.gz traits
traits<- c("SCZ.sumstats.gz", "BIP.sumstats.gz", "MDD.sumstats.gz")

#2. sample.prev = the proportion of cases to total sample size. For quantitative traits list NA
sample.prev <- c(.39,.45,.35)

#3. population.prev = the population lifetime prevalence of the traits. For quantitative traits list NA
population.prev <- c(.01,.01,.16)

##4. ld = folder of LD scores
#can be found on our github (https://github.com/GenomicSEM/GenomicSEM)
ld <- "eur_w_ld_chr/"

#5. wld = folder of LD scores
wld <- "eur_w_ld_chr/"

#6. trait.names = optional sixth  argument to list trait names so they can be named in your model
trait.names<-c("SCZ", "BIP", "MDD")
```

Answers to these questions will be in the results Rstudio prints to the screen:

Checking Understanding Questions (to discuss as group):

1. What is the liability scale h2 for BIP?
2. What is the observed scale h2 for SCZ?
3. What is the genetic correlation between SCZ and MDD?

Step 3: Run the model using the *commonfactor* function



```
#STEP 3: ESTIMATE THE COMMON FACTOR MODEL
#requires only one necessary argument:
#covstruc = the output from the ldsc function
covstruc<-PSYCH_COV

#an optional second argument can be provided for the estimation method
estimation<-"DWLS"

#run the commonfactor model below
PFactor <- commonfactor(covstruc=covstruc,estimation=estimation)

#Print PFactor results
PFactor$results
```

Checking Understanding Questions (to discuss as group):

1. What is the unstandardized factor loading for SCZ?
2. What is the standardized residual variance of MDD?

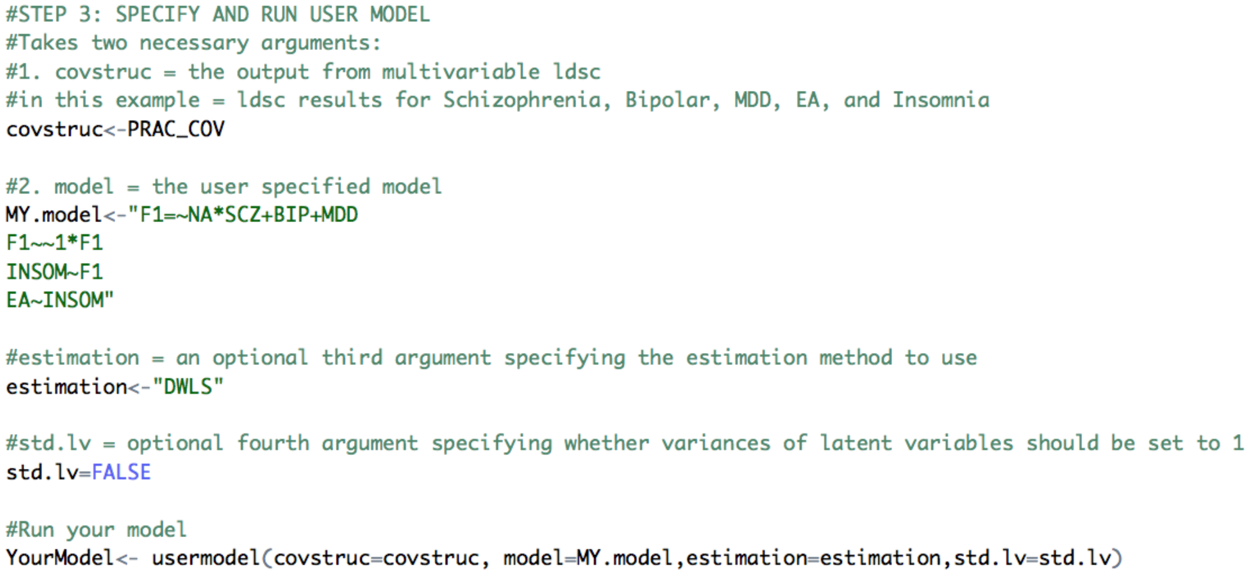
Pause here; we will discuss next steps as a group.

Estimate a User Specified Model

ACTIVITY: Fit a user specified model for three psychiatric traits (Schizophrenia, Bipolar Disorder and Major Depressive Disorder), and two external phenotypes (Educational Attainment and Insomnia)

Steps 1 and 2: Already done for you, but as with the common factor model, involved munging the summary statistics (**Step 1)** and then running multivariable ld-score regression (**Step 2**)

Step 3a: Specify and run a user-specified model that is provided for you

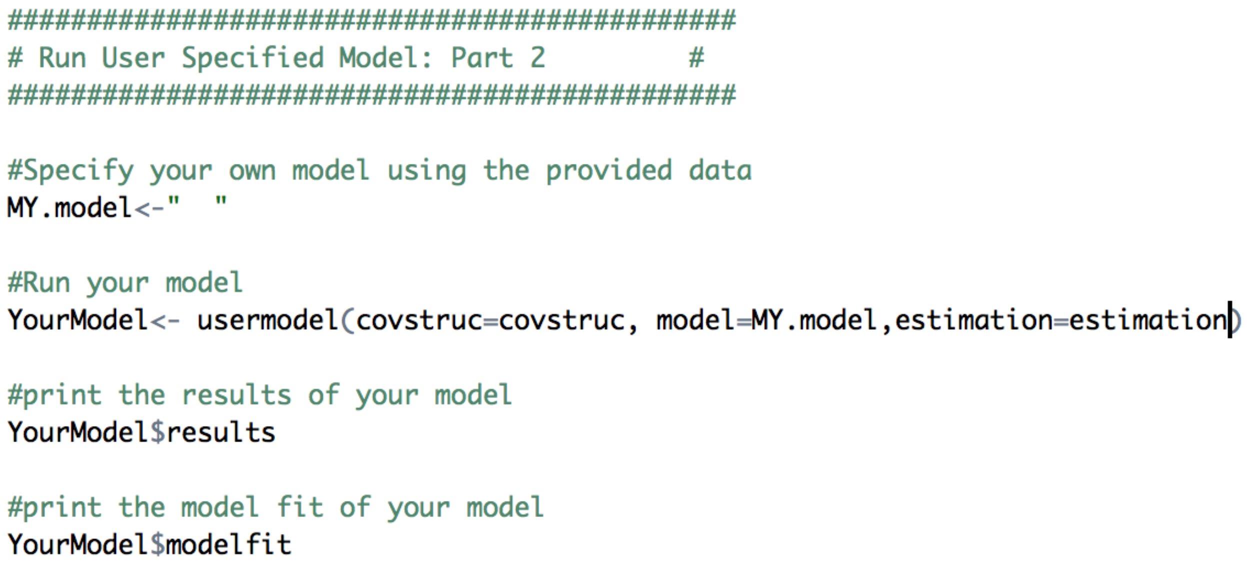


```
#STEP 3: SPECIFY AND RUN USER MODEL
#Takes two necessary arguments:
#1. covstruc = the output from multivariable ldsc
#in this example = ldsc results for Schizophrenia, Bipolar, MDD, EA, and Insomnia
covstruc<-PRAC_COV

#2. model = the user specified model
MY.model<-"F1=~NA*SCZ+BIP+MDD
F1~~1*F1
INSOM~F1
EA~INSOM"

#estimation = an optional third argument specifying the estimation method to use
estimation<-"DWLS"

#std.lv = optional fourth argument specifying whether variances of latent variables should be set to 1
std.lv=FALSE

#Run your model
YourModel<- usermodel(covstruc=covstruc, model=MY.model,estimation=estimation,std.lv=std.lv)
```

ACTIVITY: Run your own user-specified model using the same dataset

Steps 1 and 2: As before, already done for you.

Step 3: Specify your own model! Pre-register the model by it down on paper beforehand



```
##############################################
# Run User Specified Model: Part 2          #
##############################################

#Specify your own model using the provided data
MY.model<-"  "

#Run your model
YourModel<- usermodel(covstruc=covstruc, model=MY.model,estimation=estimation)

#print the results of your model
YourModel$results

#print the model fit of your model
YourModel$modelfit
```

Checking Understanding Questions (to discuss as group):

1. How would you verbally explain your model?
2. What is the CFI and AIC for your model?
3. Would you describe your model as fitting the data well?

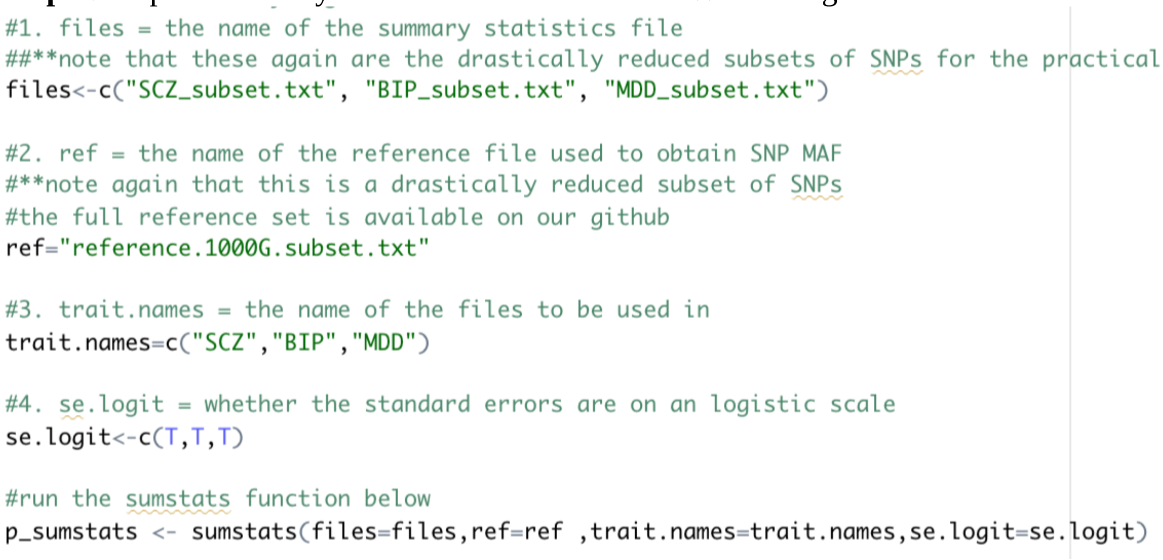
Pause here; we will discuss next steps as a group.

Multivariate GWAS in Genomic SEM

ACTIVITY: Estimate Multivariate GWAS for common factor using the three psychiatric traits (Schizophrenia, Bipolar Disorder, Major Depressive Disorder)

Steps 1 and 2: You already did them in the first example!

Step 3: Prepare summary statistics for multivariate GWAS using the *sumstats* function



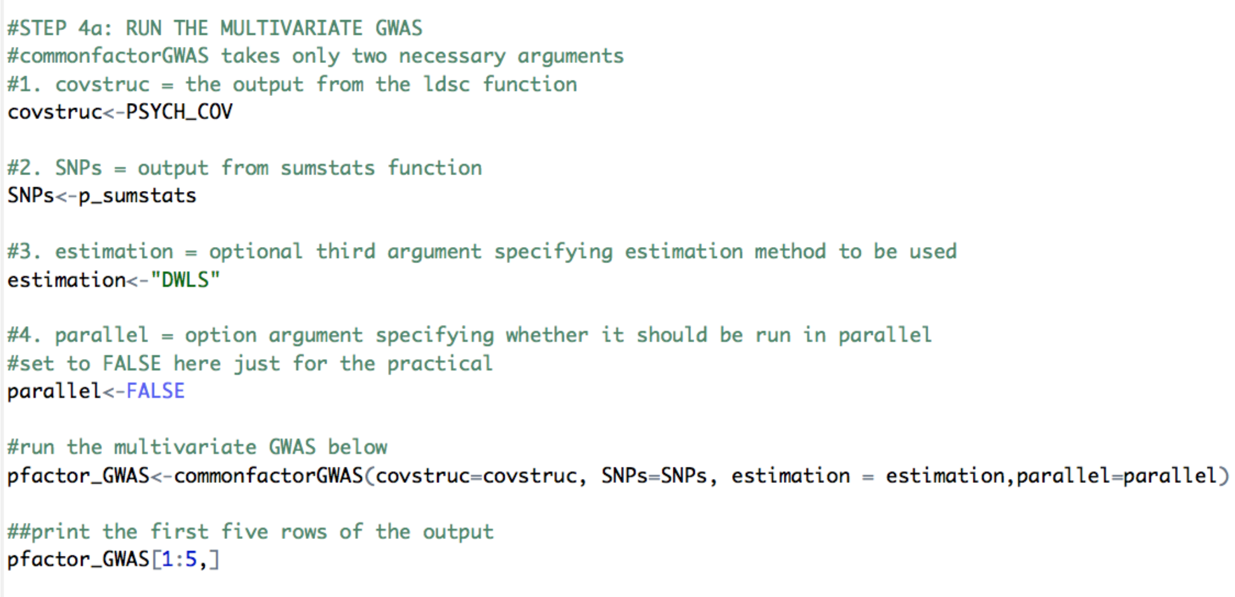
```
#1. files = the name of the summary statistics file
##**note that these again are the drastically reduced subsets of SNPs for the practical ONLY
files<-c("SCZ_subset.txt", "BIP_subset.txt", "MDD_subset.txt")

#2. ref = the name of the reference file used to obtain SNP MAF
#**note again that this is a drastically reduced subset of SNPs
#the full reference set is available on our github
ref="reference.1000G.subset.txt"

#3. trait.names = the name of the files to be used in
trait.names=c("SCZ","BIP","MDD")

#4. se.logit = whether the standard errors are on an logistic scale
se.logit<-c(T,T,T)

#run the sumstats function below
p_sumstats <- sumstats(files=files,ref=ref ,trait.names=trait.names,se.logit=se.logit)
```

You can to open the .log file created by *sumstats* to answer some of these questions or examine what R is printing to screen when running *sumstats*.

Checking Understanding Questions (to discuss as group):

1. What columns is *sumstats* using for MDD to compute total sample size?
2. How many total SNPs are left across all three traits?
3. What is being interpreted as the "effect" column for SCZ?

Step 4a: Run the multivariate GWAS using the *commonfactorGWAS* function

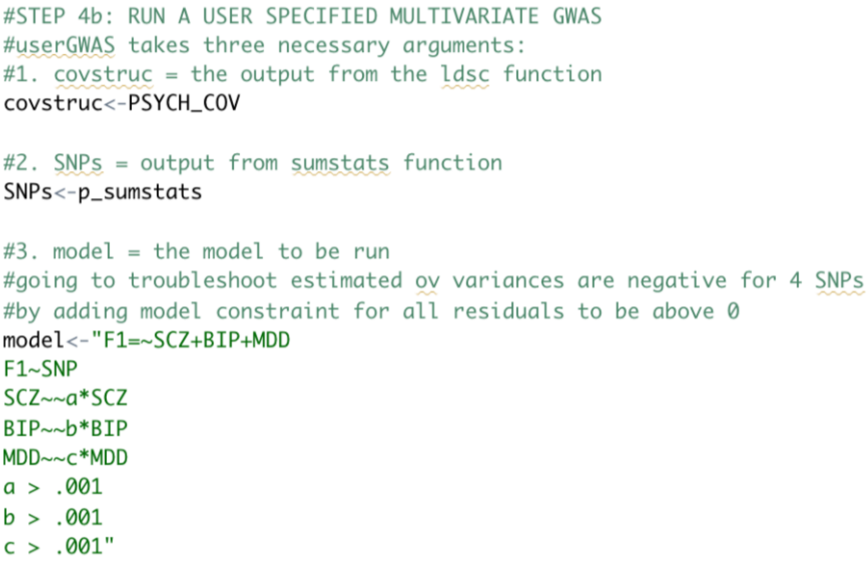


```
#STEP 4a: RUN THE MULTIVARIATE GWAS
#commonfactorGWAS takes only two necessary arguments
#1. covstruc = the output from the ldsc function
covstruc<-PSYCH_COV

#2. SNPs = output from sumstats function
SNPs<-p_sumstats

#3. estimation = optional third argument specifying estimation method to be used
estimation<-"DWLS"

#4. parallel = option argument specifying whether it should be run in parallel
#set to FALSE here just for the practical
parallel<-FALSE

#run the multivariate GWAS below
pfactor_GWAS<-commonfactorGWAS(covstruc=covstruc, SNPs=SNPs, estimation = estimation,parallel=parallel)

##print the first five rows of the output
pfactor_GWAS[1:5,]
```

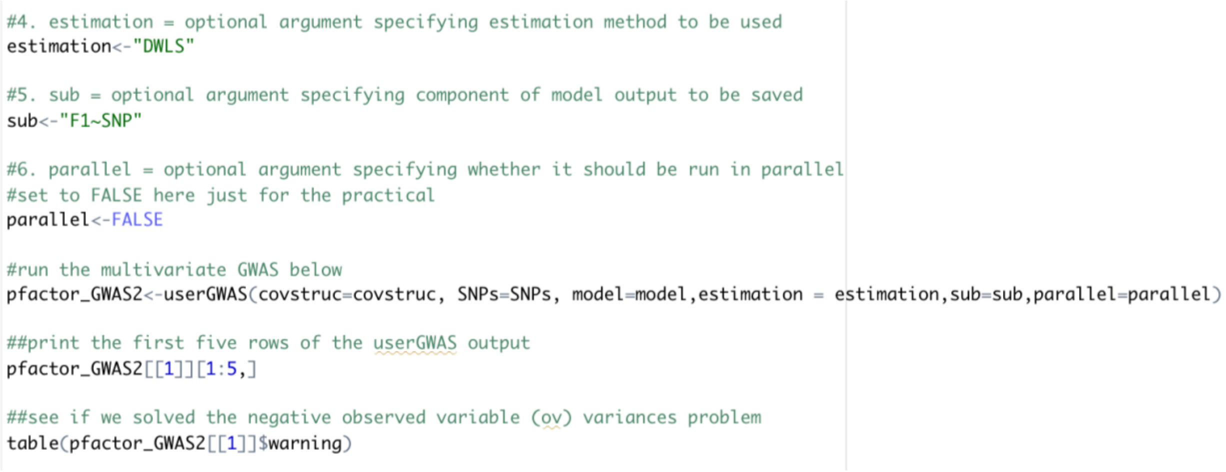
Checking Understanding Questions (to discuss as group):

1. How many warnings are there?
2. What is the p-value for the SNP effect on the factor for rs100053?

Step 4b: Run the same common factor model (with constraints on the residual variances) using the *userGWAS* function



```
#STEP 4b: RUN A USER SPECIFIED MULTIVARIATE GWAS
#userGWAS takes three necessary arguments:
#1. covstruc = the output from the ldsc function
covstruc<-PSYCH_COV

#2. SNPs = output from sumstats function
SNPs<-p_sumstats

#3. model = the model to be run
#going to troubleshoot estimated ov variances are negative for 4 SNPs
#by adding model constraint for all residuals to be above 0
model<-"F1=~SCZ+BIP+MDD
F1~SNP
SCZ~~a*SCZ
BIP~~b*BIP
MDD~~c*MDD
a > .001
b > .001
c > .001"
```



```
#4. estimation = optional argument specifying estimation method to be used
estimation<-"DWLS"

#5. sub = optional argument specifying component of model output to be saved
sub<-"F1~SNP"

#6. parallel = optional argument specifying whether it should be run in parallel
#set to FALSE here just for the practical
parallel<-FALSE

#run the multivariate GWAS below
pfactor_GWAS2<-userGWAS(covstruc=covstruc, SNPs=SNPs, model=model,estimation = estimation,sub=sub,parallel=parallel)

##print the first five rows of the userGWAS output
pfactor_GWAS2[[1]][1:5,]

##see if we solved the negative observed variable (ov) variances problem
table(pfactor_GWAS2[[1]]$warning)
```

Feel free to move on to anthropometric traits example at the end of the R code if you have time.