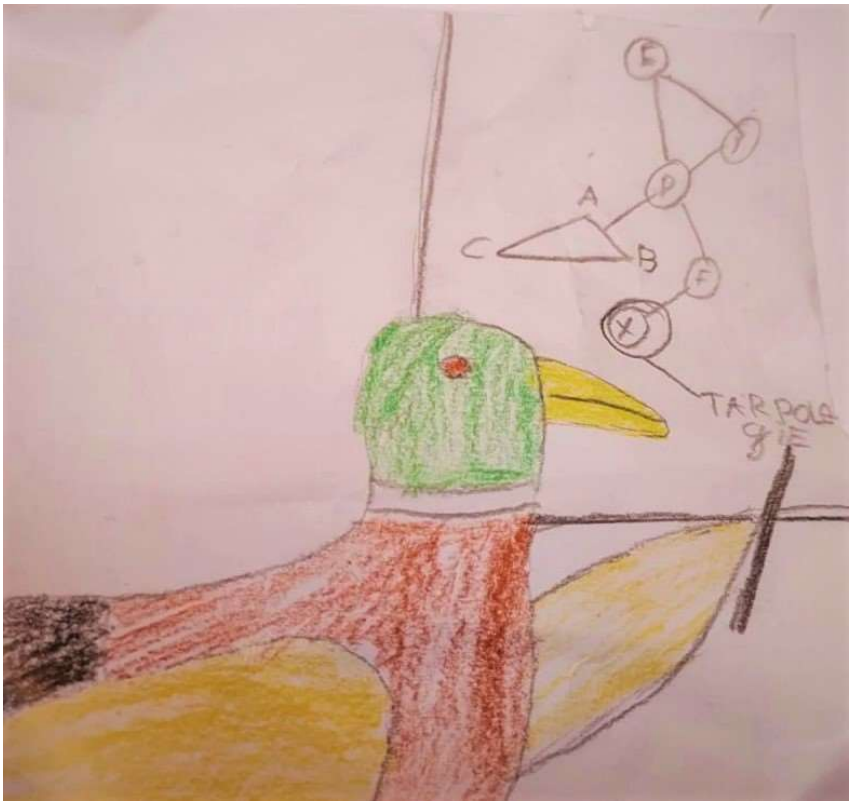
Boulder Genetics Course (online version June 2021).

Computer practical using R studio

updated 7-June (20:30 CET)



A duck called professor Rucola teaching graph theory

Conor Dolan
with edits, suggestions, and improvement by
Hermine Maes & Elizabeth Prom-Wormley
testing and suggestions by Daniel Bustamante, Morgan Driver, Daniel Zhou, Philip Vinh

R code is shown in blue bold `courier new` font

Part 1: Simulated data (dataset dataTw.dat) (Monday 7/6/21)

Estimating additive genetic variance and dominance variance
Linear regression of phenotype on SNPs
The classical twin design ... Falconer's equations

Part 2: Real data (weight in young female adults datasets mzHWB.dat and dzHWB.dat) (Tuesday 8/6/21)

Estimating additive genetic variance and dominance variance
The classical twin design ... umx and OpenMx

The main question that this practical is meant to answer:

What is the relationship between the variance explained in predicting a phenotype from a measured genetic variant (a QTL, e.g., a single nucleotide polymorphism; SNP) and the components of genetic variance that we estimate in the classical twin model?

This practical includes 5 backstories (included at the end of this document). These include additional information for students who would like some background information about the practical. Given time constraints, please do not read the backstories during the practical. The answers to the questions posed in this practical are included at the end.

In part 1 of this practical we require the dataset in the datafile dataTw.dat

This data set contains the following variables

"zygosity"

"T1QTL_A1" "T1QTL_A2" "T1QTL_A3" "T1QTL_A4" "T1QTL_A5"
"T1QTL_A6" "T1QTL_A7" "T1QTL_A8" "T1QTL_A9" "T1QTL_A10"
"T2QTL_A1" "T2QTL_A2" "T2QTL_A3" "T2QTL_A4" "T2QTL_A5"
"T2QTL_A6" "T2QTL_A7" "T2QTL_A8" "T2QTL_A9" "T2QTL_A10"

"T1QTL_D1" "T1QTL_D2" "T1QTL_D3" "T1QTL_D4" "T1QTL_D5"
"T1QTL_D6" "T1QTL_D7" "T1QTL_D8" "T1QTL_D9" "T1QTL_D10"
"T2QTL_D1" "T2QTL_D2" "T2QTL_D3" "T2QTL_D4" "T2QTL_D5"
"T2QTL_D6" "T2QTL_D7" "T2QTL_D8" "T2QTL_D9" "T2QTL_D10"

"pgsT1" "pgsT2" "phenoT1" "phenoT2"

zygosity is coded 1 (for MZ) and 2 (for DZ).
T1QTL_A1 to _A10 are 10 additively coded Quantitative Trait Loci (QTLs, e.g., SNPs) in twin 1 members
T2QTL_A1 to _A10 are 10 additively coded QTLs (SNPs) in twin 2 members.
The QTLs are diallelic with alleles A and a. The genotypes aa, Aa or aA, and AA are coded additively as follows: 0 (aa), 1 (Aa or aA) and 2 (AA)

T1QTL_D1 to _D10 are the same 10 QTLs (SNPs) in twin 1 members, in dominance coding
T2QTL_D1 to _D10 are the same 10 QTLs (SNPs) in twin 2 members, in dominance coding
The genotypes aa, Aa or aA, and AA are coded as follows: 0 (aa), 1 (Aa or aA) and 0 (AA).

"pgsT1" "pgsT2" are polygenic risk scores, i.e., sum of T1QTL_A1 to T1QTL_A10, and sum of T2QTL_A1 to T2QTL_A10, respectively. These are not used here (but see **Backstory #4**).

"phenoT1" "phenoT2" are the phenotypic scores of twin 1 and twin 2 members.

The data were simulated: the QTLs are mutually uncorrelated (in linkage equilibrium) and the allele frequency is .5, so the genotype frequencies are .25 (aa), .5 (Aa or aA), and .25 (AA). The QTLs are all associated with the phenotype with the same effect size. The gene action is both additive and dominant. Together the QTLs explain 50% of the variance of the phenotype.

The first aim of this practical is to demonstrate the estimation of additive and dominance variance components using linear regression of the phenotype on the QTLs (i.e., the QTLs are the predictors). The second aim is to do the same in the classical twin design.

The practical includes R code (to cut and paste), some questions about the R output. At the end, you will find some additional explanation (see **BACKSTORY #1 to #5**), and the answers to the questions about the R output. Please do not read the backstories now: they are provided as additional info for students who repeat this practical in their won time.

Part 1 (1.1. to 1.14, with 9 questions).

1.1. Download the data to your working directory. Start the RStudio program and set the working directory.

You can find the files in the folder faculty/conor/Boulder2021.

```
dataTw=read.table(file='dataTw.dat', header=T)
```

1.2. Check the variable names. You should see the variable names which are given on page 3.

```
colnames(dataTw)
```

1.3. Create a table of the zygosity variable and a table of T1QTL_A1, calculate the variance of the phenotype, and the conditional phenotypic means (phenotypic mean conditional genotype). If the relationship between the SNP and the phenotype is linear, the differences between the conditional means should be about equal. See the conditional means (i.e., c_means).

```
table(dataTw$zygosity)   # shows the number of MZ and DZ twinpairs
table(dataTw$T1QTL_A1)   # shows the distribution of QTL_A1
s2ph=var(dataTw$phenoT1)   # the variance of the phenotype
c_means=rep(0,3)      # a vector for the conditional phenotypic means
c_means[1]=mean(dataTw$phenoT1[dataTw$T1QTL_A1==0])
c_means[2]=mean(dataTw$phenoT1[dataTw$T1QTL_A1==1])
c_means[3]=mean(dataTw$phenoT1[dataTw$T1QTL_A1==2])
print(s2ph)     # show variance
print(c_means)  # show conditional means
```

Question 1.3. What is the variance of the phenotype, what are the genotype frequencies?

1.4. Regress "phenoT1" on T1QTL_A1. This involves estimating the parameters of the regression model (b0 and b1), and the proportion of variance explained (R^2). We can do this using the R function `lm()`.
Model: phenoT1 = b0 + b1* T1QTL_A1 + e

```
lin1A = lm(phenoT1~T1QTL_A1, data=dataTw) # T1QTL_A1 predicts
phenoT1
summary(lin1A)
```

If R^2 is the propotion of explained variance, the raw explained variance component is R^2 times the variance of the phenotype (`s2ph`).

Q.1.4. Is there association? Does the QTL predict the phenotype? Test the hypothesis b1=0 (alpha=0.005). The explained variance is additive genetic variance of the phenotype that is attributable to, or explained by, T1QTL_A1. The proportion is the additive genetic variance divided by the total ~~genetic~~ phenotypic variance. What is the proportion of explained variance?

1.5. Plot the data with the conditional means and the regression line. We can do this using the R functions **plot()**, **abline()**, and **lines()**.

```
plot(dataTw$phenoT1~dataTw$T1QTL_A1,col='grey')
abline(lin1A, lwd=3)
lines(c(0,1,2), c_means, type='p', col=2, lwd=5)
```

If the relationship between the QTL (or SNP) and the phenotype is perfectly linear, the regression line should pass through the conditional means, and the differences between the conditional means should be about equal. To test this "linearity" we can use dominant coding of the QTL and add the dominance term to the regression model. The coding is such that T1QTL_A1 and T1QTL_D1 are uncorrelated (see **BACKSTORY #1**). In the present case, the additive coding is 0 (aa), 1 (Aa or aA) and 2 (AA). The dominance coding is 0 (aa), 1 (Aa or aA), and 0 (AA).

1.6. Create a table of dataTw$T1QTL_D1 and the table of dataTw$T1QTL_A1 and dataTw$T1QTL_D1.

```
#
table(dataTw$T1QTL_D1)    # one-way table
table(dataTw$T1QTL_A1, dataTw$T1QTL_D1)    # 2 way table
#
```

In the output you can see that in the present case, the additive coding is 0 (aa), 1 (Aa or aA) and 2 (AA). The dominance coding is 0 (aa), 1 (Aa or aA), and 0 (AA).

1.7. Regress phenoT1 on T1QTL_A1 and T1QTL_D1 to estimate the parameters b0, b1 and b2 and the proportion of explained variance attributable to the QTL (additive genetic + dominance variance)
Model: phenoT1 = b0 + b1* T1QTL_A1 + b2* T1QTL_D1 + e

```
lin1AD=(lm(phenoT1~T1QTL_A1+T1QTL_D1,data=dataTw))
summary(lin1A)     # results lm(phenoT1~T1QTL_A1)
summary(lin1AD)      # results lm(phenoT1~T1QTL_A1+T1QTL_D1)
```

Q.1.7. What is the proportion of variance (R^2) explained by additively coded QTL?
What is the proportion of variance (R^2) explained by additively coded QTL and dominance coded QTL?
What is the difference in R^2?
Is contribution of T1QTL_D1 to the explained variance statistically significant (alpha=0.005)?

1.8. Plot the regression results (plot included in **BACKSTORY #1**).

```
lines(sort(dataTw$T1QTL_A1),sort(lin1AD$fitted.values),
type='b', col=6, lwd=3)
# closer look ... change the scale of Y
plot(dataTw$phenoT1~dataTw$T1QTL_A1,col='grey', ylim=c(3,7))
abline(lin1A, lwd=3)
lines(c(0,1,2), c_means, type='p', col=6, lwd=8)
lines(sort(dataTw$T1QTL_A1),sort(lin1AD$fitted.values),
type='b', col=3, lwd=3)
```

In the plot you can see the linear regression line (phenoT1~T1QTL_A1) and – in green the regression line associated with phenoT1~T1QTL_A1 + D1. The latter runs through the conditional means (purple cicles) exactly; the former does not: gene action is not perfectly additive.

The proportion of explained variance are 0.02732 (additive) and 0.03658 (additive + dominance). Because the predictors are uncorrelated, and given the phenotypic variance of 15.102 (`print(s2ph)`), we have the following variance components (note: 0.03658 - 0.02732 = .00926), as you can check for yourself (at your leisure, not necessarily during the practical).

Source	Proportion R^2	(raw) variance component
Total pheno		15.102
Total genetic	0.03658 (~3.65%)	0.03658*15.102 = ~0.552
Add genetic	0.02732 (~2.73%)	0.02732*15.102 = ~0.412
Dom genetic	0.00926 (0.926%)	0.00926*15.102 = ~0.139

Q 1.8. How much of the phenotypic variance is not explained?

We can relate the variance components to the biometric model associated with the QTL. This is important as it makes the connection between the biometric definition of additive genetic and dominance variance, and the explained variance components that we estimate in the regression (see **BACKSTORY #2**).

1.9. In GWAS, the phenotype is regressed on one additively coded SNP at a time (where the number of SNP is in the millions, so millions of regression analyses are carried out). Here, we have all the QTLs that are relevant to the phenotype. There are only 10, so this really is a toy example: in practice a polygenic phenotype is expected to be subject to hundreds, if not thousands, of QTLs. To make the link with the classical twin design, let's regress the phenotype on all 10 additively coded QTLs.

Model: phenoT1 = b0 + b1* T1QTL_A1 + b2* T1QTL_A2 + ... + b10* T1QTL_A10 + e

```
lin10A=(lm(
phenoT1~T1QTL_A1+T1QTL_A2+T1QTL_A3+T1QTL_A4+T1QTL_A5+
        T1QTL_A6+T1QTL_A7+T1QTL_A8+T1QTL_A9+T1QTL_A10,
         data=dataTw))
summary(lin10A)
```

Q. 1.9. What is the proportion of explained variance R^2? Given that the phenotypic variance is 15.102, how large is the additive genetic variance, i.e., R^2 times the phenotypic variance.

1.10. To make the link with the classical twin design (where it is possible to estimate both additive genetic variance and dominance variance), let's regress the phenotype on the dominantly coded QTLS.

Model: phenoT1 = b0 + b1* T1QTL_D1 + b2* T1QTL_D2 + ... + b10* T1QTL_D10 + e

```
lin10D=(lm(
phenoT1~T1QTL_D1+T1QTL_D2+T1QTL_D3+T1QTL_D4+T1QTL_D5+
        T1QTL_D6+T1QTL_D7+T1QTL_D8+T1QTL_D9+T1QTL_D10,
         data=dataTw))
summary(lin10D)
```

Q.1.10. What is the proportion of explained variance R^2? Given that the phenotypic variance is 15.102, how large is the dominance genetic variance?

1.11. The coding of the QTL is such that the additively coded QTLs and dominance coded QTLs are uncorrelated. Let's regress the phenotype on the QTLs coded additively and dominantly, to estimate the total genetic variance (should equal the additive genetic variance + the dominance variance)

Model: phenoT1 = b0 + b1* T1QTL_A1 + b2* T1QTL_A2 + ... + b10* T1QTL_A10 + b11* T1QTL_D1 + b12* T1QTL_D2 + ... + b20* T1QTL_D10 + e

```
lin10AD=(lm(
phenoT1~T1QTL_A1+T1QTL_A2+T1QTL_A3+T1QTL_A4+T1QTL_A5+
        T1QTL_A6+T1QTL_A7+T1QTL_A8+T1QTL_A9+T1QTL_A10+
        T1QTL_D1+T1QTL_D2+T1QTL_D3+T1QTL_D4+T1QTL_D5+
        T1QTL_D6+T1QTL_D7+T1QTL_D8+T1QTL_D9+T1QTL_D10,
         data=dataTw))
summary(lin10AD)
```

Q 1.11. What is the proportion of explained variance R^2? Given that the phenotypic variance is 15.102, how large is the total genetic variance (i.e., additive variance + dominance variance) and how large is the dominance variance?

Let's estimate the A (additive genetic) and D (dominance) variance in the classical twin design using Falconer's equations. Based on our regression results we have estimates of the total genetic variance and the A and D components. In practice, this is impossible because we do not know all the genes (their location, etc.) relevant to the (highly polygenic) phenotypes. How can we obtain estimates of A and D variance if we have not measured any QTLs at all? This is where the classical twin design comes in.

1.12 Based on the zygosity info, create MZ and DZ dataframe of the phenotypic data as follows.

```
dataMZ = dataTw[dataTw$zygosity==1, c('phenoT1', 'phenoT2')]   #
MZ data frame
dataDZ = dataTw[dataTw$zygosity==2, c('phenoT1', 'phenoT2')]   #
DZ data frame
```

1.13. Calculate the covariance matrices and the correlations using the R functions cov() and cor()

```
SMZ=cov(dataMZ)      # MZ covariance matrix
rMZ=cor(dataMZ)[2,1]  # element 2,1 in the MZ correlation matrix
SDZ=cov(dataDZ)      # DZ covariance matrix
rDZ=cor(dataDZ)[2,1]# element 2,1 in the DZ correlation matrix
```

Q.1.13. Based on the correlations, we see that MZ twins are phenotypically more alike than the DZ twins. What are the MZ and DZ correlations?

1.14. Use Faclconer's equations to obtain the standardized variance components based on the ADE model, based on the model var(pheno) = var(A) + var(D) + var(E), (A = additive genetic, D = dominance, E= unshared environmental). (see **BACKSTORY #3**).

Model for standardized phenotype:
var(pheno) = var(A) + var(D) + var(E) = $s^2_A + s^2_D + s^2_E$
cor(MZs) = rMZ= var(A) + var(D) = $s^2_A + s^2_D$
cor(DZs) = rDZ =.5*var(A) + .25*var(D) = .5*s^2_A + .25*s^2_D

solve for the unknowns:
var(A) = s^2_A = 4*rDZ - rMZ
var(D) = s^2_D = 2*rMZ – 4*rDZ
var(E) = s^2_E = 1- var(A) – var(D)

```
sA2 = 4*rDZ - rMZ
sD2 = 2*rMZ - 4*rDZ
sE2 = 1 - sA2 - sD2
print(c(sA2, sD2, sE2))
```

*Q 1.14 We know the proportion of A, D, and A+D variance from the regression analyses (1.9 – 1.11). Do these agree with the values based on the classical twin design? (**NOTE:** if you think that the answer is "NO, THEY DO NOT AGREE" then that is a good answer. To understand what is going on here please take the time – later on, at your leisure – to see the answer to the question provided below).*

In practical, we estimate variance components in the classical twin design using genetic covariance structure modeling, not Falconer's equations. We do this in part 2 of this practical using the OpenMx library and the umx library. Before we continue with Part 2 of the practical, you might ask: so if we have measured all the relevant QTLs and can estimate their combined effect on the phenotype (additive genetic and dominance effects), how do polygenic risk scores fit in to this? See – at your leisure, not during the practical - **BACKSTORY #4** for the answer to this.

Part 2 {2.1 to 2.7, with about 5 questions).

In part 1, we ended with estimating variance components using Falconer's equations (Backstory #3). In use umx and OpenMx to estimaet variance components using maximum likelihood estimates. To this end, we require the umx and the OpenMx R libraries. We first fit the ADE model to the simulated data using umx.

In practical one, we read a data frame called dataTw from the file dataTw.dat, and we created the data frames called dataMZ and dataDZ. Lets' start with reading and creating the data frames. Open RStudio, set the working memory to the folder containing the data files **(faculty/conor/Boulder2021),** and start the practical:

start

```
dataTw=read.table(file='dataTw.dat', header=T)
dataMZ = dataTw[dataTw$zygosity==1, c('phenoT1', 'phenoT2')]  #
MZ data frame
dataDZ = dataTw[dataTw$zygosity==2, c('phenoT1', 'phenoT2')]  #
DZ data frame
```

At this point, you should have the data as data frames in R "ready to go".

The library umx is an OpenMx interface. The umx library is convenient as it allows one to fit a ACE or ADE model in a single umx function call (i.e., a single line of R code). We fit the ADE model using the umx function `umxACEv()`, which provide maximum likelihood estimates of the variance components.

2.1.

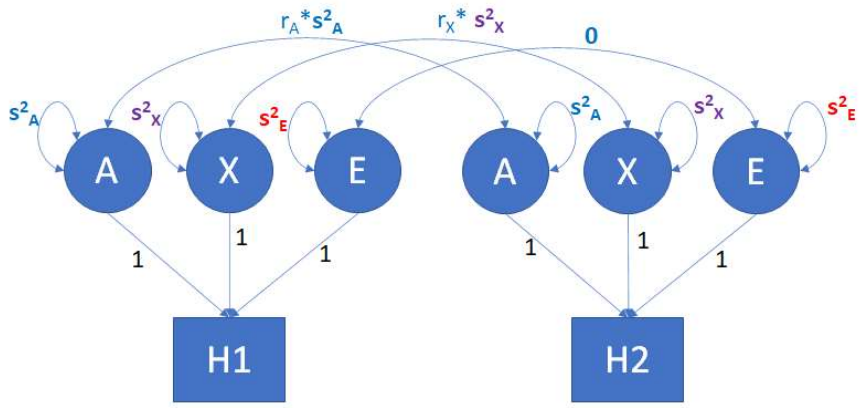
```
library(umx)  # load the umx library
selDVs=c("phenoT")  # the name of the phenotype phenoT1, phenoT2
ADEmodel_1=umxACEv(
name='ADE',
selDVs=selDVs,      # the phenotype
selCovs=NULL,        # fixed covariates ... none here
dzData=dataDZ,      # the DZ data frame
mzData=dataMZ,            # the MZ data frame
sep='',         # phenoT""1 = phenoT1, phenoT""2 = phenoT2
dzCr=.25 )  # in the ACE model dzCr=1, in the ADE model dzCr=.25
summary(ADEmodel_1)  # show the results.
```

Above you see some annotation. You can get more information about this R function by entering `?umxACEv` in R, and reading the help page associated with this function.
The umx function `umxACEv()` fits an ACE model by default. By setting `dzCr=.25`, the ADE model is fitted.

NOTE: in the umx output the notation C is still used, even though in with dzCr=.25 the ADE model is fitted. So in the umx output you will see reference to the parameters `"C_r1c1"` but actually this is refers to dominance variance. That is confusing. So remember dzCR=1 implies a ACE model, dzCR=.25

implies a ADE model, even if the umx output still refers to C (i.e., `"C_r1c1"` , where `"D_r1c1"` would be more approriate).

How dzCR=1 vs dzCr=.25 works:



A: Additive genetic variable (s^2_A is the additive genetic variance; r_A=1 in MZs and r_A =.5 in DZ);
E: Unshared environmental variance (s^2_E unshared environmental variance).
X: Shared environmental variable ($s^2_X = s^2_C$, shared environmental variance) if r_X=1 in MZs and r_X=1 in DZ
X: Dominance variable ($s^2_X = s^2_D$, dominance variance) if r_X=1 in MZs and r_X=.25 in DZ, so the dzCr in the `umx` function call is the r_X in the DZ group: given dzCr=1, X=C (shared environment); given dzCr=.25, X=D (dominance).

Q.2.1. Check the **`summary(ADEmodel_1)`** *output. What are the parameters in this model? What are the values of the variance components?*

Q.2.2. The observed phenotypic covariance matrices are

```
MZ          phenoT1   phenoT2
phenoT1 15.004166  8.187556
phenoT2  8.187556 15.130558

DZ          phenoT1   phenoT2
phenoT1 15.216090  2.833538
phenoT2  2.833538 14.558097
```

What are the expected covariance matrices based on the estimates of the variance components (see 1.14)?

2.2. The model includes three variance components, $\mathbf{s^2_A}$, $\mathbf{s^2_D}$, and $\mathbf{s^2_E}$. Suppose that we want to test the hypothesis that $\mathbf{s^2_D} = 0$, i.e., that the gene action is additive, the genetic variance is additive. We can do this by fixing the variance component to zero, i.e. imposing the constraint $\mathbf{s^2_D} = 0$ and refitting the model. We can revise the model using the umx function **`umxModify()`**. Remember that guiven dzCr=.25, the variance component denoted "C_r1c1" is a dominance variance component.

```
# the umxModify function
```

```
AEmodel_1=umxModify(
ADEmodel_1,  # the model that we want to modify
regex = c("C_r1c1"), # the parameter that we want to modify
                     # (given dzCr=.25, C is actually D, s2D)
free = FALSE,  # we want to fix this parameter,
      # so in the new model, it is not free (to be estimated)
value = 0, # the value that we fix the parameter to zero
          # i.e., s2D = 0 (fixed)
name='A_noD_E')  # a sensible name for the new AE model
summary(AEmodel_1)  # see the results
```

Q. 2.2. What are the values of s^2_A, s^2_D, and s^2_E in the revised model?

2.3. At this point, you might think: "OK, we fitted the ADE model and then the AE model, but how do I know whether we should drop the $s^2_D = 0$, can we test the hypothesis $s^2_D = 0$?". We can test this by means of the likelihood ratio test, LRT. In the output `summary(AEmodel_1)` you can see a number called "Fit (-2lnL units)", which equals 21794.14.

```
Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
      Model:              3                    3997             21794.14
```

In the output `summary(ADEmodel_1)`, the value is 21787.55

```
Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
      Model:              4                    3996             21787.55
```

The likelihood ratio test of $s^2_D = 0$ is based on this information. Specifically the difference in these values is a test statistic, which we'll call LRT. If, in truth, $s^2_D = 0$, LRT follows a chi-square distribution with degrees of freedom equal to the difference in the number of parameter, i.e., 4 – 3 = 1. Here is the R code for the LRT:

```
LRT=21794.14-21787.55
df=1
pval=pchisq(LRT,df,lower=F)
print(c(LRT, df, pval))
```

So LRT=6.59, df=1, and pval=.0102. What conclusion? Given alpha=0.05, we would conclude that we cannot set $s^2_D = 0$ (i.e, pvalue < alpha, or .0102 < .05). As this is an important test, umx library includes a dedicated function to carry it out, `umxCompare()`:

```
umxCompare(ADEmodel_1, AEmodel_1)
```

The output, which includes some additional information, is

```
|Model    | EP|Δ -2LL    |Δ df |p     |      AIC|    Δ AIC|Compare with Model |
|:-------|--:|:---------|:----|:-----|--------:|--------:|:------------------|
|ADE      |  4|          |     |      | 13795.55| 0.000000|                   |
|A_noD_E |  3|6.5915688 |1    |0.010 | 13800.14| 4.591569|ADE                |
```

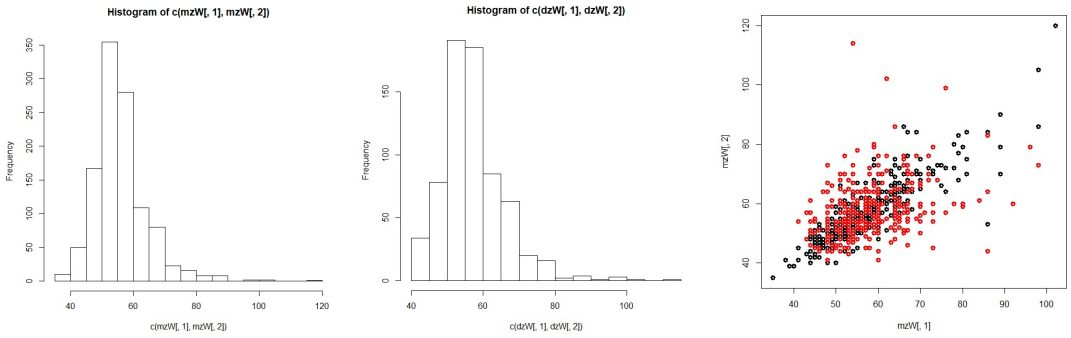

MAJESTIC
MAJESTIC

2.4. Real data: weight in young females. In the final part of this practical, we'll fit the ADE model to weight measured in young adult female twins. The sample sizes N=569 MZ pairs and N=351 DZ pairs. This dataset is part of a dataset included in the OpenMx library (the dataset is called twinData). The datafames are called mzHWB.dat and dzHWB.dat. These include height in centimeters (ht), weight in kilograms (wt), and body mass index (bmi; i.e., weight / height^2, where height is expressed in meters, rather than centimeters). Read the dataframes, and look at the first 6 lines of data.

```
mzHWB=read.table(file='mzHWB.dat', header=T)
dzHWB=read.table(file='dzHWB.dat', header=T)
head(mzHWB)  # first 6 rows of data
head(dzHWB) # first 6 rows of data
```

As you can see the variable names are ht1, wt1, bmi1, ht2, wt2, bmi2. To ease this presentation, let's create dataframes for weight only, and calculate some descriptives, and make some graphs.

```
mzW=mzHWB[,c('wt1', 'wt2')]  # only weight
dzW=dzHWB[,c('wt1', 'wt2')] # only weight
dim(mzW) # dimension ... N=569
dim(dzW) # dimension ... N=351
apply(mzW,2,mean, na.rm=T)  # phenotypic means
apply(dzW,2,mean, na.rm=T)  # phenotypic means
cov(mzW, use="pairwise.complete.obs") # covariance
cor(mzW, use="pairwise.complete.obs") # correlation
cov(dzW, use="pairwise.complete.obs")
cor(dzW, use="pairwise.complete.obs")
#x11() - this is not necessary in R Studio
plot(mzW[,1], mzW[,2], type='p', col=1, lwd=3)
lines(dzW[,1], dzW[,2], type='p', col=2, lwd=3)
#x11()
hist(c(mzW[,1],mzW[,2]),20)
#x11()
hist(c(dzW[,1],dzW[,2]),20)
```



The phenotypic correlations are rMZ = .843 (N=569, MZs) and rDZ = .334 (N=351, DZs). Given 2*rDZ < rMZ, we suspect an ADE model (see **BACKSTORY #4**). We could apply Falconer's equations to obtain estimates of the standardized variance components (as in 1.14).

```
rMZ = cor(mzW, use="pairwise.complete.obs")[2,1]
rDZ = cor(dzW, use="pairwise.complete.obs")[2,1]
#
#rMZ = .843
#rDZ = .334
sA2 = 4*rDZ - rMZ
sD2 = 2*rMZ - 4*rDZ
sE2 = 1 - sA2 - sD2
print(c(sA2, sD2, sE2))
```

2.5. Using umx. We can use umx to obtain maximum likelihood estimates of the A, D, and E variance components. The R code is almost the same as above (2.1). We have already loaded the umx library in 2.1.

```
selDVs=c("wt")  # the name of the phenotype wt1, wt2
ADEmodel_1=umxACEv(
name='ADE',
selDVs=selDVs,      # the phenotype
selCovs=NULL,        # fixed covariates ... none here
dzData=dzW,      # the DZ data frame
mzData=mzW,          # the MZ data frame
sep='',         # wt""1 = wt1, wt""2 = wt2
dzCr=.25)     # in the ACE model dzCr=1, in the ADE model
dzCr=.25
summary(ADEmodel_1)
```

Q.2.5. Check the ***summary(ADEmodel_1)*** *output. What are the parameters in this model? What are the values of the variance components? What are the standardized variance components? (i.e.,* $s^2_{Ph} = s^2_A + s^2_D + s^2_E$; s^2_A / s^2_{Ph}, *etc.)*

2.6. We can test the hypothesis that $s^2_D = 0$, in the same way as in 2.3, using the same R code

```
# the umxModify function
AEmodel_1=umxModify(
ADEmodel_1,  # the model that we want to modify
regex = c("C_r1c1"),      # the parameter that we want to modify
                    # (given dzCr=.25, C is actually D, s2D)
free = FALSE,        # we want to fix this parameter,
      # so in the new model, it is not free (to be estimated)
value = 0,      # the value that we fix the parameter to zero
          # i.e., s2D = 0 (fixed)
name='A_noD_E')  # a sensible name for the new AE model
summary(AEmodel_1)    # see the results
umxCompare(ADEmodel_1, AEmodel_1) # LRT
```

Q 2.6. Given alpha = 0.05, what would you conclude concerning the hypothesis $s^2_D = 0$*? Given alpha = 0.01, what would you conclude concerning the hypothesis* $s^2_D = 0$*?*

2.7. The umx library provides an interface to OpenMx. Here finally is the OpenMx script to carry out the same analysis (ADE model). See **BACKSTORY #5** for some annotation. Hermine Maes has a practical on OpenMx syntax. You can run this at your leisure in your own time. At this point in the practical, this code should run "as-is", and produce the same results as we just obtained using umx.

```
library(OpenMx)

nv <- 1
ntv <- nv*2
svVA=35   # starting values of var(A), a guess
svVC=.0   # starting values of var(C), zero: this is an ADE model
svVD=30   # starting values of var(D)
svVE=10  # starting value of var(E)
svMe=60  # starting value phenotypic mean
selVars=c('wt1','wt2')
# Create matrix for expected Mean Matrices
meanPh     <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe,
                                  labels=c("mean","mean"), name="meanPh" )
#ACDE model
# Create Matrices for Variance Components
covA      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE,
                       values=svVA, label=" VA11", name="VA" )
covC      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=FALSE,
                       values=svVC, label="VC11", name="VC" )
covD      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE,
                       values=svVD, label="VD11", name="VD" )
covE      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE,
                       values=svVE, label="VE11", name="VE" )
#ACDE model
# Create Algebra for expected Variance/Covariance Matrices in MZ & DZ twins
covP      <- mxAlgebra( expression= VA+VC+VD+VE, name="V" )
covMZ     <- mxAlgebra( expression= VA+VC+VD, name="cMZ" )
covDZ     <- mxAlgebra( expression= 0.5%x%VA +0.25%x%VD+ VC, name="cDZ" )
expCovMZ  <- mxAlgebra( expression= rbind( cbind(V, cMZ), cbind(t(cMZ), V)),
                                                    name="expCovMZ" )
expCovDZ  <- mxAlgebra( expression= rbind( cbind(V, cDZ), cbind(t(cDZ), V)),
                                                    name="expCovDZ" )
#ACDE model
# Create Data Objects for Multiple Groups
#ACDE model
dataMZ    <- mxData( observed=mzW, type="raw" )
dataDZ    <- mxData( observed=dzW, type="raw" )
#ACDE model
# Create Expectation Objects for Multiple Groups
expMZ     <- mxExpectationNormal( covariance="expCovMZ", means="meanPh",
                                                   dimnames=selVars )
expDZ     <- mxExpectationNormal( covariance="expCovDZ", means="meanPh",
                                                   dimnames=selVars )
funML     <- mxFitFunctionML()
#ACDE model
# Create Model Objects for Multiple Groups
pars      <- list( meanPh, covA, covC, covD, covE, covP )
modelMZ   <- mxModel( pars, covMZ, expCovMZ, dataMZ, expMZ, funML, name="MZ" )
modelDZ   <- mxModel( pars, covDZ, expCovDZ, dataDZ, expDZ, funML, name="DZ" )
multi     <- mxFitFunctionMultigroup( c("MZ","DZ") )
# Build Model with Confidence Intervals
modelACDE  <- mxModel( "oneACDEvc", pars, modelMZ, modelDZ, multi)
```

```
#ACDE model run
fitACDE    <- mxRun( modelACDE)
sumACDE    <- summary( fitACDE)
# Run AE model
modelAE   <- mxModel( fitACDE, name="oneAEvc" )
modelAE   <- omxSetParameters( modelAE, labels="VD11", free=FALSE, values=0 )
fitAE     <- mxRun( modelAE)
sumAE = summary(fitAE)
mxCompare(fitACDE, fitAE)
sumACDE
#
```

BACKSTORY #1.

In GWAS of a continuous variable, the phenotype is regressed on the additively coded QTL (quantitative trait locus, e.g., a SNP). Given a diallelic locus, with alleles A and a, there are three possible genotypes, AA, Aa (same as aA) and aa. The genotypes are typicaly coded as shown in the table:

Table : coding the QTL		
	additive	dominance
genotype	GV_A	GV_D
AA	2	4p-2
Aa or aA	1	2p
aa	0	0

The basic model is (note that in practice other covariates may be included, such as sex, age, etc.

$phenotype_i = b_0 + b_1*GV_{Ai} + e_i$,

This model is an additive model, because the effect of allele A on the phenotype is additive. You can see this if you consider the predicted values (or equivalently the conditional means):

$Mean(phenotype \mid GV_A = 0) = b_0 + b_1*0 = b_0$
$Mean(phenotype \mid GV_A = 1) = b_0 + b_1*1 = b_0 + b_1$
$Mean(phenotype \mid GV_A = 2) = b_0 + b_1*2 = b_0 + b_1 + b_1 = b_0 + 2*b_1$

This implies that the predicted values fall on a straight line, i.e., on the regression line. Whether this model "fits the data", depends on the relationship between the genotypes and the phenotype. If in truth the observed conditional means (the phenotypic means calculated within each genotype group) fall on a straight line, the gene action is called additive. If that is not the case, the model does not "fit the data" perfectly, and the gene action is called dominant. That does NOT mean that the linear regression is useless: given association, the predictor GVA will still account for phenotypic variance, but the linear model will not provide a 100% accurate account of the genotype – phenotype relationship. We can extend the model as follows to include the QTL in additive coding (GV_A) and in dominance coding (GV_D).

$phenotype_i = b_0 + b_1*GV_{Ai} + b_2*GV_{Di} + e_i$,

Note that the coding of the GV_D involves the allele frequency. Coding in this way ensures that the GV_A and GV_D are not correlated, so that the decomposition of variance is $\sigma_{ph}^2 = b_1^2*\sigma_{GVA}^2 + b_1^2*\sigma_{GVD}^2 + \sigma_e^2$. The path diagram of the extended model is shown in the Figure below.

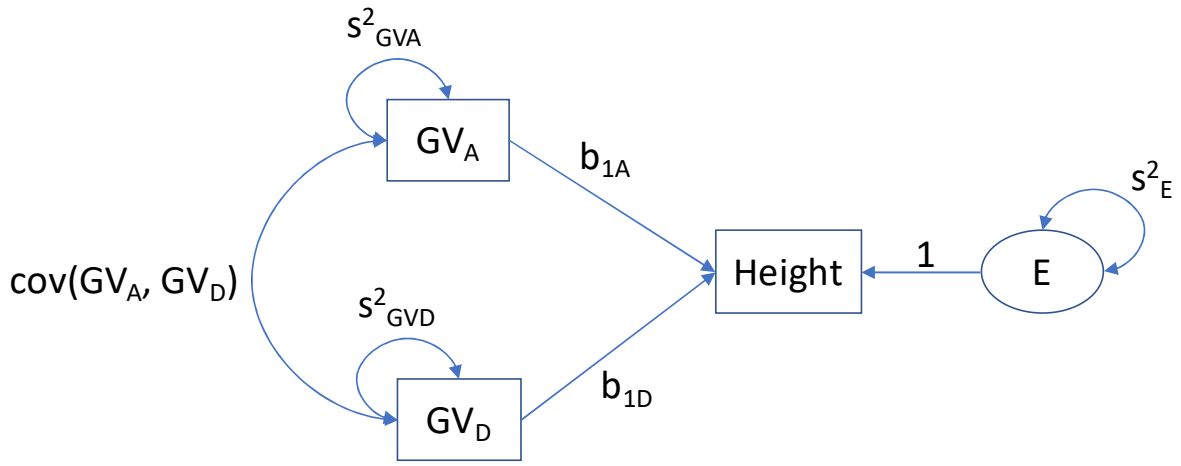


Figure 1.1. Given the dominance coding shown in Table above, $cov(GV_A, GV_D) = 0$.

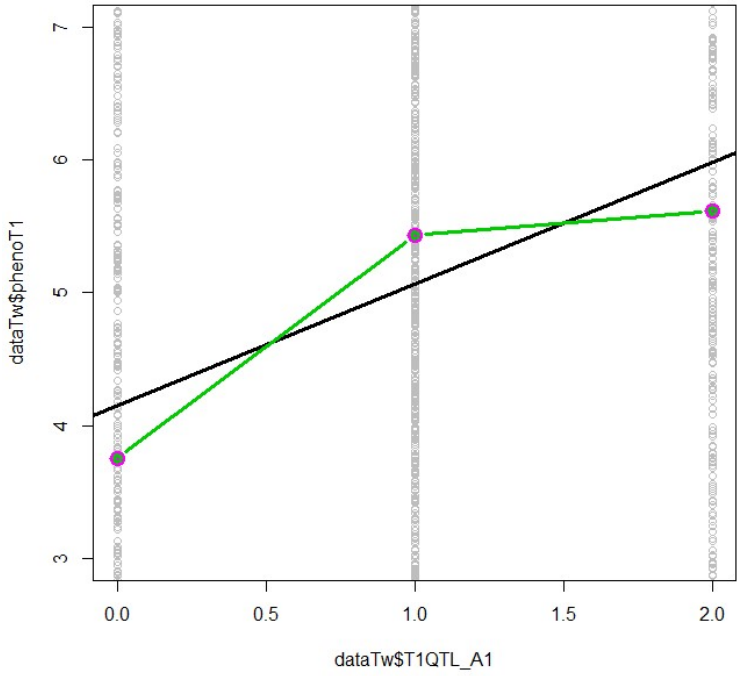


Figure 1.2. Purple points: the observed conditional phenotypic means (conditional on the genotype)
Black regression line $phenotype_i = b_0 + b_1*GV_{Ai} + e_i$,
Green regression line $phenotype_i = b_0 + b_1*GV_{Ai} + b_2*GV_{Di} + e_i$,

BACKSTORY #2. A bit of biometrical genetics

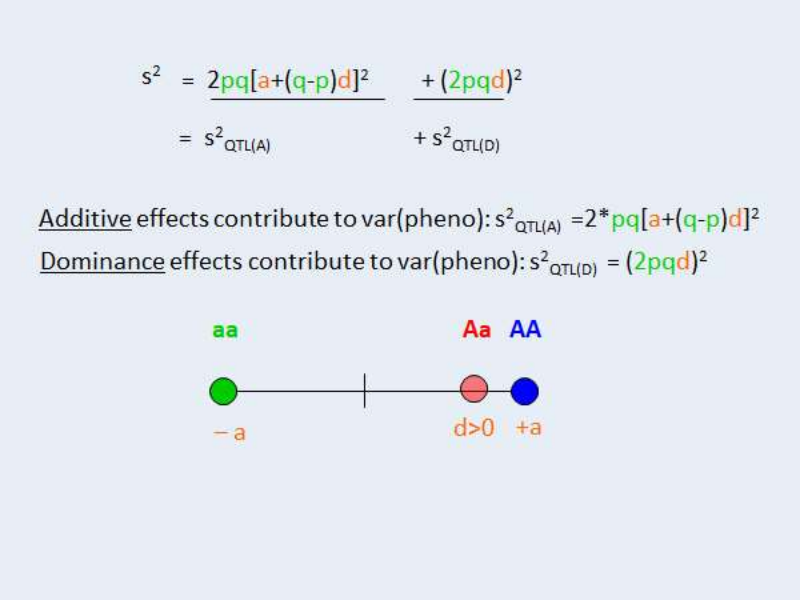


Figure 2.1 (sorry about the bad notation: green a is an allele, orange a is a genotype effect). In this Figure dominance is present as d, the effect of the heterozygote Aa, is not equal to zero (i.e., not intermediate in effect to the effects of the two homozygotes) d>0. Note that d=0 implies dominance, where either d<0 or d>0 are possible.

Let p denote the frequency of allele A and q=1-p the frequency of allele a (in a well defined population), where A and a are the alleles at a given autosomal locus. Given Hardy-Weinberg Equilibrium (HWE), the genotype frequencies are p^2 (AA), 2*p*q (Aa or aA), and q^2 (aa). Associated with the genotypes are the effects -a (aa), d (Aa or aA) and +a (AA). The total genetic variance attributable to this locus equals

$2*p*q*[(a+(q-p)*d)]^2 + [2*p*q*d]^2$,

where $2*p*q*[(a+(q-p)*d)]^2$ the additive variance and $[2*p*q*d]^2$ is dominance variance.

In our data simulation, we have 10 QTLs in linkage equilibrium. For each QTL we set – in the simulation - p=.5 (q=.5), a=1 and d=1.

source	raw variance component	expected
Add genetic	0.02732*15.102 = ~0.412	$2*p*q*[(a+(q-p)*d)]^2 = 2*.5*.5*1^2 = .5$
Dom genetic	0.00926*15.102 = ~0.139	$[2*p*q*d]^2$= (2*.5*.5*1)^2 = .25
Total genetic	0.03658*15.102 = ~0.552	$2*p*q*[(a+(q-p)*d)]^2 + [2*p*q*d]^2 = .75$

The discrepancy between the estimated variance components and the expected variance components (e.g. observed .412 vs expected .5) is simply due to sampling fluctuation.

BACKSTORY #3. Falconer's equations.

A simple estimation method based on standardized phenotypes. .... Falconer's equations for the ACE model and ADE model. Given the ACE model, we assume that the phenotypic variance equals

$s^2_{Ph} = s^2_A + s^2_C + s^2_E$, where

s^2_A is additive genetic variance, s^2_C is shared environmental variance, and s^2_E is unshared environmental variance. According to the classical twin design, we have the expected twin covariance matrices shown in the Figure below

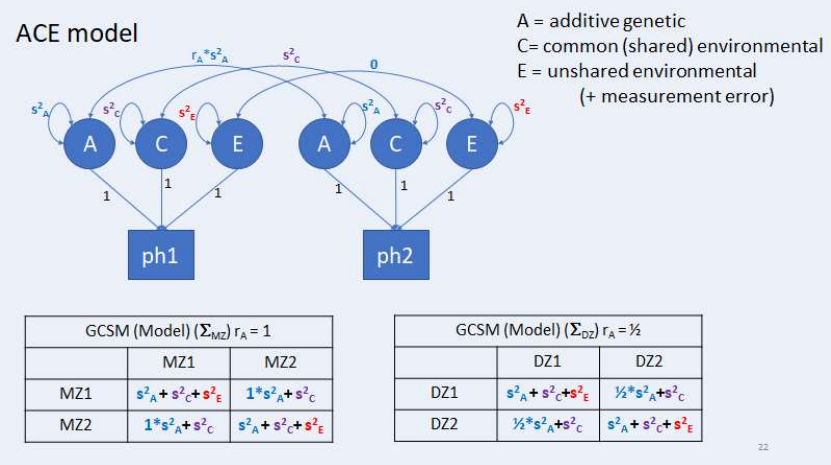


Figure 3.1: ACE model path diagram and expected covariance matrices

Given standardization of the phenotype, we have

$s^2_{Ph} = s^2_A + s^2_C + s^2_E = 1$
cov(MZ) = cor(MZ) = rMZ = $s^2_A + s^2_C$
cov(DZ) = cor(DZ) = rDZ = $.5*s^2_A + s^2_C$

which implies (solving for the unknowns) the following, i.e., Falconer's equations for the ACE model

$s^2_A = 2*(r_{MZ}-r_{DZ})$
$s^2_C = 2*r_{DZ}-r_{MZ}$
$s^2_E = 1- s^2_A - s^2_C$

Given the ADE model, we assume that the phenotypic variance equals

$s^2_{Ph} = s^2_A + s^2_D + s^2_E$, where

s^2_A is additive genetic variance, s^2_D is dominance variance, and s^2_E is unshared environmental variance. According to the classical twin design, we have the expected twin covariance matrices shown below in Figure 4

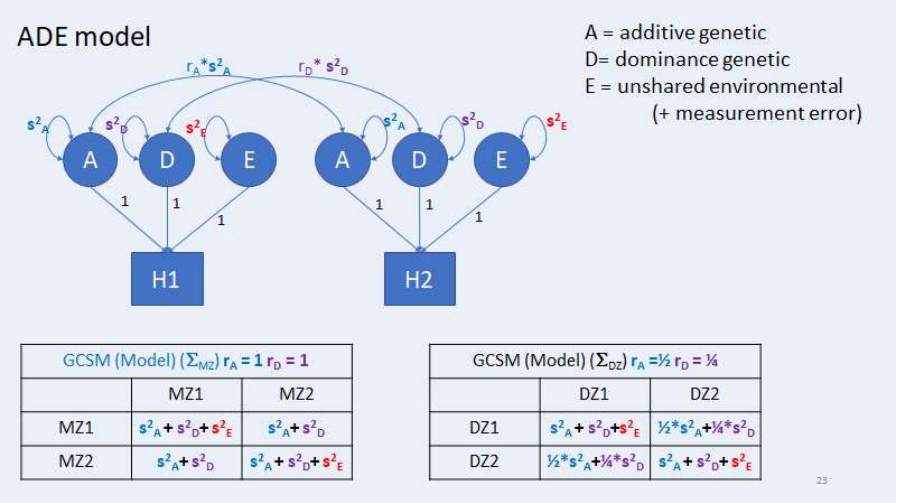


Figure 3.2: ADE model path diagram and expected covariance matrices

Given standardization of the phenotype, we have

$\mathbf{s^2_{Ph}} = \mathbf{s^2_A} + \mathbf{s^2_D} + \mathbf{s^2_E} = 1$
$cov(MZ) = cor(MZ) = rMZ = \mathbf{s^2_A} + \mathbf{s^2_D}$
$cov(DZ) = cor(DZ) = rDZ = .5*\mathbf{s^2_A} + .25*\mathbf{s^2_D}$

which implies (solving for the unknowns) the following, i.e., Falconer's equations for the ACE model

$\mathbf{s^2_A} = 4*r_{DZ}-r_{MZ}$
$\mathbf{s^2_D} = 2*r_{MZ}-4*r_{DZ}$
$\mathbf{s^2_E} = 1- \mathbf{s^2_A} - \mathbf{s^2_D}$

How to decide between an ACE and ADE model? The rule of thumb is

ADE model, if $(2*r_{DZ})<r_{MZ}$
ACE model, if $(2*r_{DZ})>r_{MZ}$

Note that $(2*r_{DZ}) = r_{MZ}$ suggests an AE model, i.e., a special case of the ACE model (drop C variance, i.e., $\mathbf{s^2_C} = 0$) and the AE model (drop D variance, i.e., $\mathbf{s^2_D} = 0$).

BACKSTORY #4: polygenic risk scores.

In 1.9. we fitted the following regression model:

Model: phenoT1 = b0 + b1* T1QTL_A1 + b2* T1QTL_A2 + ... + b10* T1QTL_A10 + e

That is, the regression of the phenotype on the measured QTLs (10 QTLs, all relevant to the phenotype). So what are polygenic scores and how – in this toy example – do they relate to additive genetic variance in the classical twin design? The following is relatively simple because the QTLs are uncorrelated (in linkage equilibrium). We have measured the QTLs and have coded them additively (0,1,2). Note that the "predictor part" of the linear regression model is

b1* T1QTL_A1 + b2* T1QTL_A2 + ... + b10* T1QTL_A10

This part of the model includes observed variables T1QTL_A1 to T1QTL_A10 and unknown parameters b1 to b10. If we know the values of b1 to b10, we simplify the model as follows:

Model: phenoT1 = b0 + A + e

where **A = b1* T1QTL_A1 + b2* T1QTL_A2 + ... + b10* T1QTL_A10.** We actually know the values of b1 to b10 (simulated data...), so we really can calculate the variable A. This A variable is the same as the A variable in the classical twin design. So in the context of our toy dataset, we expect the regression of the phenotype of the 10 QTLs (additively coded) to produce the same R^2 as obtained in the regression of the phenotype of the A, i.e., the polygenic risk score as defined above.

It is important to understand that the polygenic risk scores are based on 1) a subset of relevant QTLs (SNPs), 2) the QTLs (SNPs) may be correlated due to linkage disequilibrium (LD) – so that call for some correction to take the LD into account, 3) in practice the regression weights are estimated, not known, so that the variation in the score A is a function of both variation in the QTLs and the variation in the regression coefficients. In the case of our toy data set we can regress the phenotype on the 10 QTLs or on our polygenic risk score and obtain about the same proportion of explained variance R^2. Is that so? Yes: 33.44% vs. 33.95%. See below.

```
> summary(linPGSA)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -15.00235    0.63775  -23.52   <2e-16 ***
pgsT1         1.00105    0.03159   31.68   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.171 on 1998 degrees of freedom
Multiple R-squared:  0.3344,    Adjusted R-squared:  0.3341
F-statistic:  1004 on 1 and 1998 DF,  p-value: < 2.2e-16

> summary(lin10A)


Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.98016    0.32547 -15.301   <2e-16 ***
```

```
T1QTL_A1     0.86835    0.10136   8.567   <2e-16 ***
T1QTL_A2     0.96957    0.10139   9.563   <2e-16 ***
T1QTL_A3     0.93857    0.09918   9.464   <2e-16 ***
T1QTL_A4     0.96458    0.09929   9.715   <2e-16 ***
T1QTL_A5     0.84703    0.10211   8.296   <2e-16 ***
T1QTL_A6     1.15908    0.10013  11.576   <2e-16 ***
T1QTL_A7     0.91373    0.10076   9.068   <2e-16 ***
T1QTL_A8     1.16146    0.09896  11.737   <2e-16 ***
T1QTL_A9     1.21302    0.10060  12.058   <2e-16 ***
T1QTL_A10    0.97548    0.10247   9.519   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.166 on 1989 degrees of freedom
Multiple R-squared:  0.3395,    Adjusted R-squared:  0.3362
F-statistic: 102.2 on 10 and 1989 DF,  p-value: < 2.2e-16.
```

BACKSTORY #5: Annotation of the OpenMx script

```
library(OpenMx)
#
svVA=35   # starting values of var(A), a guess
svVC=.0   # starting values of var(C), zero: this is an ADE model
svVD=30   # starting values of var(D)
svVE=10  # starting value of var(E)
svMe=60  # starting value phenotypic mean
```

Annotation: svVA etc. are starting value. Maximum liklihood estimation is an iterative process that requires starting values for the parameters. The start values chosen here are good starting values as they were based on the umx output.

```
selVars=c('wt1','wt2')
```

Annotation: the variable that we want to analyze (weight)

```
# Create matrix for expected Mean Matrices
meanPh     <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe, labels=c("mean","mean"),
name="meanPh" )
```

Annotation: This OpenMx matrix contains the mean of the phenotype

```
#ACDE model
# Create Matrices for Variance Components
covA      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svVA, label="VA11", name="VA" )
covC      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=FALSE, values=svVC, label="VC11", name="VC" )
covD      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svVD, label="VD11", name="VD" )
covE      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svVE, label="VE11", name="VE" )
#ACDE model
```

Annotation: covA to covE are the covariance matrices of A, C, D, and E. In the present case, we have 1 phenotype (weight), so that these are actually 1x1 covariance matrices, or simple variances. Note that the C variance component is fixed to zero (values=svVC, where scVC=0, and free=FALSE, i.e., not estimated), so this is actually an ADE model. Note that VA is the 1x1 covariance matrix, which contains one parameter (the additive genetic variance). This variance is a parameter in the model, which is called "VA11". So here there is a one-to-one correspondence between VA11 and VA, because there is only one phenotype. In a multivariate analysis, say p phenotype, the matrix VA would be pxp, and would include p*(p+1)/2 parameters (p variance and p*(p-1)/2 covariances).

```
# Create Algebra for expected Variance/Covariance Matrices in MZ & DZ twins
covP      <- mxAlgebra( expression= VA+VC+VD+VE, name="V" )

Annotation: the phenotypic variance is the some of the variance components.

covMZ     <- mxAlgebra( expression= VA+VC+VD, name="cMZ" )
covDZ     <- mxAlgebra( expression= 0.5%x%VA +0.25%x%VD+ VC, name="cDZ" )
expCovMZ  <- mxAlgebra( expression= rbind( cbind(V, cMZ), cbind(t(cMZ), V)), name="expCovMZ" )
expCovDZ  <- mxAlgebra( expression= rbind( cbind(V, cDZ), cbind(t(cDZ), V)), name="expCovDZ" )
#ACDE model
```

Annotation: create the MZ and DZ covariance matrices, there are 2x2 matrices.

```
# Create Data Objects for Multiple Groups
#ACDE model
dataMZ    <- mxData( observed=mzW, type="raw" )
dataDZ    <- mxData( observed=dzW, type="raw" )
#ACDE model
```

Annotation: define the datasets as OpenMX data object

```
#ACDE model
# Create Expectation Objects for Multiple Groups
expMZ     <- mxExpectationNormal( covariance="expCovMZ", means="meanPh", dimnames=selVars )
expDZ     <- mxExpectationNormal( covariance="expCovDZ", means="meanPh", dimnames=selVars )
funML     <- mxFitFunctionML()
#ACDE model
# Create Model Objects for Multiple Groups
pars      <- list( meanPh, covA, covC, covD, covE, covP )
modelMZ   <- mxModel( pars, covMZ, expCovMZ, dataMZ, expMZ, funML, name="MZ" )
modelDZ   <- mxModel( pars, covDZ, expCovDZ, dataDZ, expDZ, funML, name="DZ" )
```

```
multi     <- mxFitFunctionMultigroup( c("MZ","DZ") )
# Build Model with Confidence Intervals
modelACDE  <- mxModel( "oneACDEvc", pars, modelMZ, modelDZ, multi)
#ACDE model run
fitACDE    <- mxRun( modelACDE)
sumACDE    <- summary( fitACDE)
```

Annotation: run the model

```
# Run AE model
modelAE   <- mxModel( fitACDE, name="oneAEvc" )
modelAE   <- omxSetParameters( modelAE, labels="VD11", free=FALSE, values=0 )
```

Annotation: drop the parameter VD11 (dominance variance), i.e. VD11 =0, reducing the model from ADE to AE.

```
fitAE     <- mxRun( modelAE)
sumAE = summary(fitAE)
mxCompare(fitACDE, fitAE)
```

Annotation: Run the model, and do the likelihood ratio test (is VD11 = 0?)

The model fitted is shown below

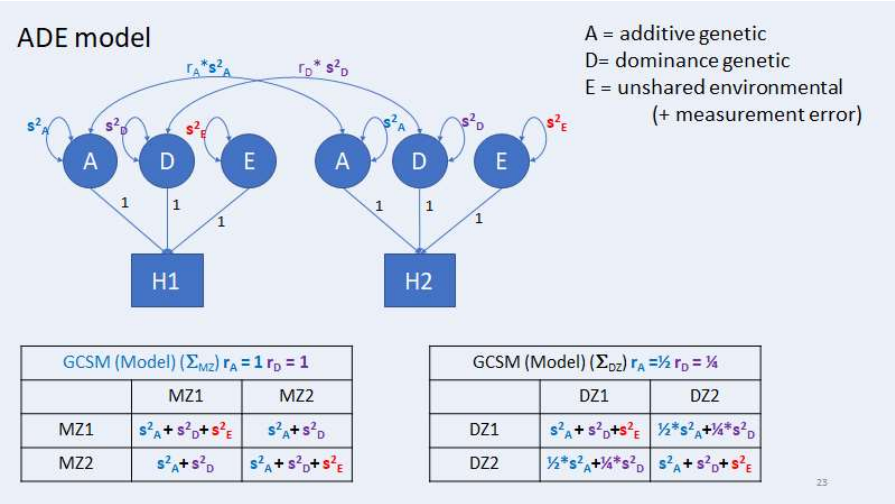


Figure 5.1 ADE model path diagram and covariance matrices

In terms of the OpenMx specification, the covariance matrices are specified as follows:

```
covMZ     <- mxAlgebra( expression= VA+VC+VD, name="cMZ" )
covDZ     <- mxAlgebra( expression= 0.5%x%VA +0.25%x%VD+ VC, name="cDZ" )
expCovMZ  <- mxAlgebra( expression= rbind( cbind(V, cMZ), cbind(t(cMZ), V)), name="expCovMZ" )
expCovDZ  <- mxAlgebra( expression= rbind( cbind(V, cDZ), cbind(t(cDZ), V)), name="expCovDZ" )
#ACDE model
```

that is:

	MZ1	MZ2
MZ1	VA+VC+VD+VE	VA+VC+VD
MZ2	VA + VC + VD	VA+VC+VD+VE

	DZ1	DZ2
DZ1	VA+VC+VD+VE	.5*VA+VC+.25*VD
DZ2	.5*VA + VC + .25*VD	VA+VC+VD+VE

where VA=[VA11], VD=[VD11], VC=[VC11], VE=[VE11]. However, we are fitting an ADE model, so VC11 = 0, meaning that VC=0, so that the variances are effectively VA+VD+VE and the covariances are VA+VD (MZ), and .5*VA+.25*VD (DZ). Finally let's take the means and covariance matrices involved in this analysis:

```
MZ observed mean
56.65468 56.50090
DZ observed means
58.18182 57.73469

MZ covariance matrix
         wt1      wt2
wt1 73.44090 63.35853
wt2 63.35853 77.66200
MZ correlation = 63.358 / sqrt(73.440*77.662) = 0.8437

DZ covariance matrix
         wt1      wt2
wt1 74.63155 26.77257
wt2 26.77257 84.56391
DZ correlation = 26.772 / sqrt(74.631*84.564) = 0.3344

MZ model expected phenotypic mean
meanPh     meanPh
DZ model expected phenotypic mean
meanPh     meanPh

MZ expected covariance matrix
      MZ1              MZ2
MZ1   VA + VD + VE     VA + VD
MZ2   VA + VD          VA + VD+VE

DZ expected covariance matrix
      DZ1                    DZ2
DZ1   VA + VD + VE           .5*VA + .25*VD
DZ2   .5*VA + .25*VD         VA + VD+VE
```

Given the maximum likelihood estimates: VA=36.53555, VD=29.65004, VE=11.7815, m=57.20859.

```
MZ model phenotypic mean
57.20859 57.20859
DZ model phenotypic mean
57.20859 57.20859

MZ covariance matrix
> fitACDE$MZ$expCovMZ$result
         [,1]     [,2]
[1,] 77.96709 66.18559
[2,] 66.18559 77.96709
```

```
cor = 66.185 / 77.967  = .8488
```

DZ covariance matrix

```
> fitACDE$DZ$expCovDZ$result
         [,1]     [,2]
[1,] 77.96709 25.68029
[2,] 25.68029 77.96709
cor = 25.680 / 77.967  = .329
```

Based on the results we obtain the following standardized variance components

36.53555/77.96709 = 0.468 (narrow-sense heritability; 46.8% additive genetic)

(36.53555+29.65004)/77.96709 = 0.848 (broad-sense heritability; 84.8% additive genetic and dominance)

11.7815/77.96709 = 0.151 (15.1% unshared environmental + measurement error).

Anwers to the questions Part 1.

Q.1.3. What is the variance of the phenotype, what are the genotype frequencies?

The genotype counts are
```
  0   1   2
474 1021  505
```

So the genotype freqs are
> 474/2000
[1] 0.237 (aa)
> 1021/2000
[1] 0.5105 (Aa, aA)
> 505/2000
[1] 0.2525 (AA)

The phenotypic variance equals 15.10257.

Q.1.4. Is there association? Does the QTL predict the phenotype? Test the hypothesis b1=0 (alpha=0.005). The explained variance is additive genetic variance of the phenotype that is attributable to, or explained by T1QTL_A1. The proportion is the additive genetic variance divided by the total genetic variance. What is the proportion of explained variance?

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.1464     0.1511  27.438  < 2e-16 ***
T1QTL_A1      0.9180     0.1226   7.491 1.02e-13 ***
```

Multiple R-squared: 0.02732

Given alpha=0.005, we reject the null hypothesis b1=0, as p < alpha. We conclude that there is association. The proportion of explained variance equal .02732, so the additive coded QTL explain 2.732% of the phenotypic variance (the variance component equals .02732*15.10257 = 0.412).

Q.1.7. Proportion of variance (R^2) explained by additively coded QTL?
Proportion of variance (R^2) explained by additively coded QTL and dominance coded QTL?
Difference in R^2?
Is contribution of T1QTL_D1 to the explained variance statistically significant (alpha=0.005)?

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.1464     0.1511  27.438  < 2e-16 ***
T1QTL_A1      0.9180     0.1226   7.491 1.02e-13 ***
Multiple R-squared:  0.02732
```

The proportion of explained variance is .02732, i.e., 2.732%

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.7522     0.1753  21.405  < 2e-16 ***
```

```
T1QTL_A1      0.9301     0.1220   7.622 3.83e-14 ***
T1QTL_D1      0.7483     0.1708   4.382 1.24e-05 ***
Multiple R-squared:  0.03658
```

The proportion of explained variance is .03658, i.e., 3.658%.

The total genetic variance (due to the QTL): 3.658%
The additive genetic variance (due to the additively coded QTL): 2.732%
The difference is 3.658 – 2.732 = 0.926%, which is dominance variance due to the QTL

The test of whether this 0.926 is statistically significant is the test of the regression coefficient associated with the dominance coded QTL: .7483 (st err .1708). Given alpha=0.005, we conclude that the dominance variance is not zero (p=1.24E-05, i.w., p < alpha).

Q 1.8.How much of the phenotypic variance is not explained?

Given the results:

source	proportion R^2	raw variance component
Total pheno		15.102
Total genetic	0.03658 (~3.65%)	0.03658*15.102 = ~0.552
Add genetic	0.02732 (~2.73%)	0.02732*15.102 = ~0.412
Dom genetic	0.00926 (0.926%)	0.00926*15.102 = ~0.139

The total explained proportion is .03658, so that the unexplained proportion is
1-.03658 = 0.96342
The unexplained variance is (1-.03658)*15.102 = 14.54957

Q. 1.9. What is the proportion of explained variance R^2? Given that the phenotypic variance is 15.102, how large is the additive genetic variance?

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.98016    0.32547 -15.301   <2e-16 ***
T1QTL_A1     0.86835    0.10136   8.567   <2e-16 ***
T1QTL_A2     0.96957    0.10139   9.563   <2e-16 ***
T1QTL_A3     0.93857    0.09918   9.464   <2e-16 ***
T1QTL_A4     0.96458    0.09929   9.715   <2e-16 ***
T1QTL_A5     0.84703    0.10211   8.296   <2e-16 ***
T1QTL_A6     1.15908    0.10013  11.576   <2e-16 ***
T1QTL_A7     0.91373    0.10076   9.068   <2e-16 ***
T1QTL_A8     1.16146    0.09896  11.737   <2e-16 ***
T1QTL_A9     1.21302    0.10060  12.058   <2e-16 ***
T1QTL_A10    0.97548    0.10247   9.519   <2e-16 ***
Multiple R-squared:  0.3395
```

The proportion is .3395, i.e., 33.95% of the variance. The explained variance equals .3395*15.102 = 5.127.

Q 1.11. What is the proportion of explained variance R^2? Given that the phenotypic variance is 15.102, how large is the total genetic variance (does this equal additive variance + dominance variance) and how large is the dominance variance?

The proportion of explained variance is multiple R-squared: 0.509, so 50.9% is explained by the QTL (additive + dominance). The additive part is 33.95% (see previous question).

Total genetic variance: .509*15.102 = 7.686
Additive genetic variance: .3395*15.102 = 5.127
Dominance variance: (.509 - .3395)*15.102 =0.1695*(15.102) = 2.559

Q.1.13. Based on the correlations, we see that MZ twins are phenotypically more alike than the DZ twins. What are the MZ and DZ correlations?

```
> rMZ
[1] 0.5434016
> rDZ
[1] 0.1903817
```

Q 1.14 We know the proportion of A, D, and A+D variance from the regression analyses (1.9 – 1.11). Do these agree with the values based on the classical twin design?

```
> sA2 = 4*rDZ - rMZ
> sD2 = 2*rMZ - 4*rDZ
> sE2 = 1 - sA2 - sD2
> print(c(sA2, sD2, sE2))
[1] 0.2181251 0.3252764 0.4565984
```

Note: In Q.1.11, we found .3395 (sA2), .1695 (sD2), so this seems to differ a lot. The values .3395 and .1695 are definitely better (closer to the truth) as these are based on the analysis of the actually observed QTLs.

The apparent discrepancy is due to the imprecision of the estimates based on the twin design. Specifically the standardized variance components with 95% confidence intervals (calculated using OpenMx) are

```
confidence intervals:
      lbound           estimate         ubound           Based on 1.11
```

s^2_A	-0.02019899	0.2241315	0.4595139	.3395
s^2_D	0.07307716	0.3170788	0.5697425	.1695

If we take the results of 1.11 to be the true values, we note that the confidence interval includes the values which we obtained in section 1.11.

Anwers to the questions Part 2.

Q.2.1. Check the `summary(ADEmodel_1)` output. What are the parameters in this model? What are the values of the variance components?

The parameters in the model are 1) the mean; 2) the three variance components (A, D, E). The variance components are given below:

```
2      A_r1c1    top.A  1  1 3.351658
3      C_r1c1    top.C  1  1 4.741629
4      E_r1c1    top.E  1  1 6.860765
```

Q.2.2. The observed phenotypic covariance matrices are

```
MZ          phenoT1   phenoT2
phenoT1 15.004166  8.187556
phenoT2  8.187556 15.130558

DZ          phenoT1   phenoT2
phenoT1 15.216090  2.833538
phenoT2  2.833538 14.558097
```

What are the expected covariance matrices based on the estimates of the variance components (see 1.14)?

The variance components are
2 A_r1c1 top.A 1 1 3.351658
3 C_r1c1 top.C 1 1 4.741629
4 E_r1c1 top.E 1 1 6.860765
The phenotypic variance is 3.351 + 4.741 + 6.860 = 14.95
The MZ covariance is 3.351 + 4.741 = 8.093
The DZ covariance is .5*3.351 + .25*4.741 = 2.861

We can extract the expected covariance matrices from the umx output:

```
> ADEmodel_1$top$expCovMZ$result
          phenoT1   phenoT2
phenoT1 14.954052  8.093287
phenoT2  8.093287 14.954052
> ADEmodel_1$top$expCovDZ$result
          phenoT1   phenoT2
phenoT1 14.954052  2.861236
phenoT2  2.861236 14.954052
```

Q. 2.2. What are the values of s^2_A, s^2_D, and s^2_E in the revised model?

2 A_r1c1 top.A 1 1 7.881474 is the value of s^2_A
3 E_r1c1 top.E 1 1 7.133863 is the value of s^2_E

The variance component s^2_D is fixed to zero in this model, so $s^2_D = 0$.

Q.2.5. Check the `summary(ADEmodel_1)` output. What are the parameters in this model? What are the values of the variance components?
What are the standardized variance components? (i.e., $s^2_{Ph} = s^2_A + s^2_D + s^2_E$; s^2_A / s^2_{Ph}, etc.)

The parameters are three variance components and the phenotypic mean.
The values of the variance components are given below:

```
2       A_r1c1        top.A    1    1 36.53490
3       C_r1c1        top.C    1    1 29.65062
4       E_r1c1        top.E    1    1 11.78150
```

The phenotypic variance is 36.534 + 29.650 + 11.781 = 77.96
The standardized variance components are

```
> 36.534 / 77.96
```
s^2_A / s^2_{Ph} = 0.4686249
```
> 29.650 / 77.96
```
s^2_D / s^2_{Ph} = 0.3803232
```
> 11.781/ 77.96
```
s^2_E / s^2_{Ph} = 0.151116

Q 2.6. Given alpha = 0.05, what would you conclude concerning the hypothesis $s^2_D = 0$?

```
> umxCompare(ADEmodel_1, AEmodel_1) # LRT
```

Model	EP	Δ -2LL	Δ df	p	AIC	Δ AIC	Compare with Model
ADE	4				8599.476	0.000000	
A_noD_E	3	4.8025856	1	0.028	8602.278	2.802586	ADE

Given alpha=0.05, we would reject hypothesis $s^2_D = 0$ and conclude that $s^2_D > 0$?