Multivariate twin models:
Independent Pathway Model (IPM) and the Common Pathway Model (CPM)

ANSWERS to the QUESIONS

Conor Dolan & Dirk Pelt
(c.v.dolan@vu.nl, h.m.pelt@vu.nl)

VU VRIJE UNIVERSITEIT AMSTERDAM | Faculty of Behavioural and Movement Sciences

Nederlands Tweelingen Register

Pratical Part 1: Answers to 15 questions

Question 1.1. Check out the MZ and DZ phenotypic correlation (rmz, rdz) of each phenotype. The rule of thumb is: if 2*rdz>rmz, then choose an ACE model; if 2*rdz<rmz choose a ADE model. What are the correlations and what model would you choose?

Answer 1.1: the correlations are MZ: .787, .808, .783, .823, and DZ: .324, .272, .290, .288. As consistently 2*rdz>rmz, I'd choose a ADE model.

Question 1.2. Using "Falconer's equations" we can obtain preliminary estimates of the standardized A, D, E variance components, VA, VD, & VE. In the case of the ADE model, these equations are

VA = 4*rdz-rmz
VD= 2*rmz-4*rdz
VE= 1-VA-VD

Apply these to the correlations, to obtain the standardized components VA and VD. How much of the phenotypic variance is due to A (additive genetic effects)?

Answer 1.2: I like to use a small function for this.

```
F_ADE = function(rmz, rdz) { c(4*rdz-rmz, 2*rmz-4*rdz) }
rmz=c(.787, .808, .783, .823)
rdz=c(.324, .272, .290, .288)
print(F_ADE(rmz,rdz))
```

```
VA                              VD
0.509 0.280 0.377 0.329     0.278 0.528 0.406 0.494
```

Variance of the 4 phenotypes due to A in percentages: 50.9% 28.0% 37.7% 32.9%

Question 1.3. In the output, we can see the 8x8 covariance matrix of the MZ group (called Smz4) and the correlation matrix of the MZ group (called Rmz4). We can express the covariance matrix as a function of the correlation matrix Rmz4 and the diagonal matrix containing the standard deviations. Do this in R as follows 1) get the standard deviations from the covariance matrix, and put these in the diagonal of a 8x8 matrix, called Sdmz and carry out the multiplication: Sdmz%*%Rmz4%*%t(Sdmz)

```
# ->
# express covariance matrix using stdevs and correlation matrix
Vmz=(diag(Smz4))  # variances from diagonal to vector Vmz
Sdmz=diag(sqrt(Vmz))  # standard deviation into diagonal
Smz = Sdmz%*%Rmz4%*%t(Sdmz)  #### Useful to know
round(Smz,3)
round(Smz4-Smz,3) # compare with previous calculation
# <-
```

Answer 1.3:

Given the covariance matrix Smz4 (calculated earlier), you can obtain the correlation matrix as follows:

```
# ->
Rmz=cov2cor(Smz4)
round(Rmz,3)
#
Sdmzi=diag(sqrt(1/Vmz))   # Sdmzi diagonal with 1/std
Rmz=Sdmzi%*%Smz4%*%t(Sdmzi)
round(Rmz,3)
round(Rmz4-Rmz,3) # compare with previous calculation
# <-
```

```
> round(Smz,3)
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]
[1,] 12.478 10.379 13.047 16.254  9.571  8.586 10.661 12.264
[2,] 10.379 10.604 11.930 14.599  8.103  8.503  9.627 11.096
[3,] 13.047 11.930 20.500 22.392 10.823  9.954 16.650 17.481
[4,] 16.254 14.599 22.392 28.667 13.539 12.549 18.717 22.471
[5,]  9.571  8.103 10.823 13.539 11.845  9.894 13.398 15.001
[6,]  8.586  8.503  9.954 12.549  9.894 10.447 12.237 13.765
[7,] 10.661  9.627 16.650 18.717 13.398 12.237 22.081 21.755
[8,] 12.264 11.096 17.481 22.471 15.001 13.765 21.755 26.022
> round(Smz4-Smz,3) # compare with previous calculation
       bic_T1 tri_T1 ssc_T1 sil_T1 bic_T2 tri_T2 ssc_T2 sil_T2
bic_T1      0      0      0      0      0      0      0      0
tri_T1      0      0      0      0      0      0      0      0
ssc_T1      0      0      0      0      0      0      0      0
sil_T1      0      0      0      0      0      0      0      0
bic_T2      0      0      0      0      0      0      0      0
tri_T2      0      0      0      0      0      0      0      0
ssc_T2      0      0      0      0      0      0      0      0
sil_T2      0      0      0      0      0      0      0      0
> # <-
```

Question 1.4. Repeat the analyses in the twin 2 member. What do you conclude with respect to the main effect of sex? Check out the `Multiple R-squared` statistic and the p value associated with the regression coefficient in the regression of the skinfold measures on sex (`Pr(>|t|)`). You can get these as follows (e.g., `r11` output)

Answer 1.4:

```
#
r11$r.squared; r11$coefficients
#

# ->
# regression analyses in twin 1 members (MZ and DZ)
r11=summary(lm(bic_T1~sex_T1, data=skinfold))
r12=summary(lm(tri_T1~sex_T1, data=skinfold))
r13=summary(lm(ssc_T1~sex_T1, data=skinfold))
r14=summary(lm(sil_T1~sex_T1, data=skinfold))
# regression analyses in twin 2 members (MZ and DZ)
r21=summary(lm(bic_T2~sex_T2, data=skinfold))
r22=summary(lm(tri_T2~sex_T2, data=skinfold))
r23=summary(lm(ssc_T2~sex_T2, data=skinfold))
r24=summary(lm(sil_T2~sex_T2, data=skinfold))
# see the R2 – proportion of explained variance
print(c(r11$r.squared, r12$r.squared, r13$r.squared, r14$r.squared))
print(c(r21$r.squared, r22$r.squared, r23$r.squared, r24$r.squared))
# <-
```

```
> print(c(r11$r.squared, r12$r.squared, r13$r.squared, r14$r.squared))
[1] 0.06113459 0.04301970 0.05669036 0.03225223
> print(c(r21$r.squared, r22$r.squared, r23$r.squared, r24$r.squared))
[1] 0.07957842 0.06355664 0.09277487 0.06437159
```

All p values < .001, as shown in the output `r11`, `r22`, etc.

Question 1.5. What are the MZ and DZ correlations of the 4 phenotypes based on the OpenMx output, and why do they differ from those of question 2? The correlations were (question 2):

```
> rmz=c(.787, .808, .783, .823)
> rdz=c(.324, .272, .290, .288)
```

Answer 1.5.

The correlations are now

```
rmz=c(.770, .796, .769, .808)
rdz=c(.309, .267, .293, .297)
```

The correlations differ slightly because these correlations are corrected for sex effects.

Question 1.6. What do you conclude on the basis of the MZ and DZ cross-trait – cross-twin correlations of biceps and triceps (underlined), given MZ cor(bic_T1, tri_T2) > given DZ cor(bic_T1, tri_T2) ?

MZ cross-twin-cross-trait correlations			DZ cross-twin-cross-trait correlations		
	bic_T1	tri_T1		bic_T1	tri_T1
bic_T2	**0.769**	0.706	bic_T2	**0.312**	0.269
tri_T2	0.737	**0.795**	tri_T2	0.254	**0.270**

Answer 1.6: Genetic factors are probably contributing to the covariance between biceps and triceps, as the MZ correlations are much larger than the DZ correlations.

Question 1.7. What does the guess svVA=svS* .4 actually imply with respect to the variance of the 4 phenotype phenotypes? Is this reasonable given your answer to question 2? (From question 2, we know that the percentages are 50.9%, 28.0%, 37.7% and 32.9%**).**

Answer 1.7: This implies that 40% of the phenotypic variances and covariances is due to additive genetic effects. I.e., $\Sigma_A = .4*\Sigma_{Ph}$

Question 1.8. Verify that VA_ = sA_%*%RA_%*%t(sA_).

Answer 1.8:

```
# ->
VA_ = fitADE4$VA$result
RA_ = fitADE4$RA$values
sA_ = fitADE4$sA$values
#
print(VA_)
sA_%*%RA_%*%t(sA_)
# <-
```

```
> print(VA_)
         [,1]     [,2]     [,3]     [,4]
[1,] 4.853464 2.893920 4.035718 5.282234
[2,] 2.893920 2.229991 1.852565 2.049091
[3,] 4.035718 1.852565 7.205262 7.053915
[4,] 5.282234 2.049091 7.053915 9.303209
> sA_%*%RA_%*%t(sA_)
         [,1]     [,2]     [,3]     [,4]
[1,] 4.853464 2.893920 4.035718 5.282234
[2,] 2.893920 2.229991 1.852565 2.049091
[3,] 4.035718 1.852565 7.205262 7.053915
[4,] 5.282234 2.049091 7.053915 9.303209
```

Question 1.9. Extract the covariance matrices of A (already done!), D and E, call these VA_, VD_, and VE_. But here we only need the 2x2 matrices of the phenotypes biceps and triceps. Do this as follows.

```
# ->
VA_ = fitADE4$VA$result[1:2,1:2]
VD_ = fitADE4$VD$result[1:2,1:2]
VE_ = fitADE4$VE$result[1:2,1:2]
# <-
```

```
> VA_
         [,1]     [,2]
[1,] 4.852640 2.893227
[2,] 2.893227 2.229423
> VD_
         [,1]     [,2]
[1,] 5.029458 5.771617
[2,] 5.771617 6.734171
> VE_
         [,1]     [,2]
[1,] 2.725765 1.894279
[2,] 1.894279 2.114601
```

Question 1.10. Look at the values in VA_, VD_, and VE_, they should resemble approximately the values in *Figure slide 26 left*. NOTE: they are not exactly equal, because the results in the slides were obtain by fitting the bivariate model. but the present results were obtained by fitting the 4-variate model. Verify that the parameters in *Figure slide 26 left* resemble the values in VA_, VD_, and VE_.

Answer 1.10:

```
> VA_
      [,1]     [,2]
[1,] 4.853464 2.893920
[2,] 2.893920 2.229991
> VD_
      [,1]     [,2]
[1,] 5.028658 5.770940
[2,] 5.770940 6.733608
> VE_
      [,1]     [,2]
[1,] 2.725781 1.894291
[2,] 1.894291 2.114610
```

Question 1.11. To obtain the values in *Figure slide 26 right* we need the A, D, and E correlations and the A D and E standard deviations. In this path diagram, the path coefficients are the A, D, and E standard deviations and the A, D, E variables are standardized, so that the coefficients association with the double headed arrows are correlations. Get these results by getting the standard deviations and the correlations. You can extract these from the output. E.g., for A:

```
# ->
VA_ = fitADE4$VA$result[1:2,1:2]
RA_ = fitADE4$RA$values[1:2,1:2]
sA_ = fitADE4$sA$values[1:2,1:2]
# <-
```

Alternatively, you can use `cov2cor()` applied to the covariance matrices (to obtain the correlations), and `sqrt(diag())` applied to the covariance matrices (to obtain the standard deviations). Verify that the parameter in Figure slide 26 right resemble your values.

Answer 1.11:

```
# ->
cov2cor(VA_)  # correlations
sqrt(diag(VA_)) # st devs
cov2cor(VD_)  # correlations
sqrt(diag(VD_)) # st devs
cov2cor(VE_)  # correlations
sqrt(diag(VE_)) # st devs
# <-
```

```
> cov2cor(VA_)  # correlations
          [,1]      [,2]
[1,] 1.0000000 0.8796483
[2,] 0.8796483 1.0000000
> sqrt(diag(VA_)) # st devs
[1] 2.203058 1.493316
> cov2cor(VD_)  # correlations
          [,1]      [,2]
[1,] 1.0000000 0.9917372
[2,] 0.9917372 1.0000000
> sqrt(diag(VD_)) # st devs
[1] 2.242467 2.594920
> cov2cor(VE_)  # correlations
          [,1]      [,2]
[1,] 1.0000000 0.7890167
[2,] 0.7890167 1.0000000
> sqrt(diag(VE_)) # st devs
[1] 1.650994 1.454170
```

Question 1.12. The representation in *Figure slide 27 right* is the easiest to interpret, because all variables are standardized. However, it is more complicated to calculate the path coefficients. We already have the correlations (Question 11). We now require the standardized variance components. To calculate these:

```
# ->
VPh_ = VA_ + VD_ + VE_
diag(VA_ / VPh_) # the standardized A variance component
diag(VD_ / VPh_) # the standardized D variance component
diag(VE_ / VPh_) # the standardized E variance component
# <-
```

Verify that your parameter values resemble those in *Figure slide 27 right*.

Answer 1.12:

```
> cov2cor(VA_)  # correlations
          [,1]      [,2]
[1,] 1.0000000 0.8851769
[2,] 0.8851769 1.0000000
> cov2cor(VD_)  # correlations
          [,1]      [,2]
[1,] 1.0000000 0.9888395
[2,] 0.9888395 1.0000000
> cov2cor(VE_)  # correlations
          [,1]      [,2]
[1,] 1.0000000 0.7891396
[2,] 0.7891396 1.0000000

> VPh_ = VA_ + VD_ + VE_
> diag(VA_ / VPh_) # the standardized A variance component
[1] 0.3849541 0.2012953
> diag(VD_ / VPh_) # the standardized D variance component
[1] 0.3988497 0.6078246
> diag(VE_ / VPh_) # the standardized E variance component
[1] 0.2161962 0.1908801
```

Question 1.13: If I tell you that in the model the environmental correlation equals .789, does that mean that E is contributing greatly to the phenotypic correlation? The actual expected phenotypic correlation is .893. Answer this by referring to the actual contribution of E to the phenotypic correlation.

Answer 1.13:
rE = .789 does not tell us that the contribution is large. The contribution depends on the correlation, but also on the standard deviations of E.

```
> sqrt(.216)*.789*sqrt(.190)
[1] 0.1598382
```

The phenotypic correlation is .893, but the contribution to this of E is only .159.

Question 1.14: What is wrong with the 4x4 correlation matrix of D? You can extract this as follows:

```
# ->
RD_ = fitADE4$RD$values
# <-
```

Answer 1.14:
Some correlations are larger than 1.

```
> RD_
          [,1]      [,2]     [,3]      [,4]
[1,] 1.0000000 0.9917372 1.031062 0.9630709
[2,] 0.9917372 1.0000000 1.071759 1.0803190
[3,] 1.0310616 1.0717595 1.000000 1.0037344
[4,] 0.9630709 1.0803190 1.003734 1.0000000
```

Question 1.15: use the `omxSetParameters()` function to fix the D correlations to 1.0, and use the `mxCompare()` function to test whether these constraints are ok statistically. If OK, what does this result suggest (interpretation)? Why is df=6 (`diffdf=6`)?

Answer 1.15:

```
# ->
tmp=omxSetParameters(fitADE4,
labels=c("rD21","rD31","rD41","rD32","rD42","rD43"), values=1.0,
free=FALSE)
fittmp=mxRun(tmp)
mxCompare(fitADE4, fittmp)
# <-
```

```
> mxCompare(fitADE4, fittmp)
  base comparison ep minus2LL   df      AIC   diffLL diffdf         p
1 ACE4       <NA> 38 13292.05 3118 13368.05       NA     NA        NA
2 ACE4       ACE4 32 13294.74 3124 13358.74 2.687771      6 0.8468896
```

Note the df (diffdf) equals 6, because we fixed 6 parameters (the D correlations) to one.

We can conclude that the correlations are equal to 1 (p=.846). This result suggests that dominance effects are common to all phenotypes. There are no phenotype-specific dominance effects.

Practical part 2 Answers to 12 questions

Question 2.1: Consider the twin correlations of each item. Remember that rmz>2*rdz suggests an ADE model and rmz<2*rdz suggests an ACE model. However, rmz=2*rdz suggests an AE model and rmz = rdz suggests a CE model. Based on the correlations, which model is most likely to hold?

Answer 2.1:
MZ cors .212, .330, .236, .270
DZ cors .100, .170, .114, .120

Probably AE, as the DZ correlations are approximately half the MZ correlations (rmz about equal to 2*rdz).

Question 2.2: Are the means equal in the MZs and DZs? And why is df=4 (`diffdf=4`).

Answer 2.2: Yes, they are equal given any reasonable alpha (.05): T(4)=6.059, df=4.

```
> mxCompare(fitSAT4A, fitSAT4B)
  base comparison ep minus2LL    df      AIC   diffLL diffdf        p
1  Sat       <NA> 80 60109.33 22360 60269.33       NA     NA       NA
2  Sat        Sat 76 60115.39 22364 60267.39 6.059427      4 0.194754
```

Why df=4 (diffdf=4)? Because we went from 8 means
`labels=c("b01mz","b02mz","b03mz","b04mz","b01dz","b02dz","b03dz","b04dz")`
to 4 means
`newlabels=c("b01","b02","b03","b04","b01","b02","b03","b04")`
So the differences in the number of parameters is 8-4 = 4.

Question 2.3: Does the ADE model fit relative to the saturated model?

```
# ->
mxCompare(fitSAT4B, fitADE4V)
# <-
```

Answer 2.3: The ADE model appears to fit (alpha = .05 …)

```
> mxCompare(fitSAT4B, fitADE4)
  base comparison ep minus2LL    df      AIC   diffLL diffdf          p
1  Sat       <NA> 76 60115.39 22364 60267.39       NA     NA         NA
2  Sat       ADE4 34 60173.04 22406 60241.04 57.64847     42 0.05447915
```

Question 2.4: Use `omxSetParameters()` to drop the D component from the ADE model, and fit the AE 4 variate model. The D component parameter labels are given in the OpenMx script:

```
SD      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svVD,
label=c("SD11","SD21","SD31","SD41","SD22","SD32","SD42","SD33","SD43","SD44"),
name="VD" )
```

Answer 2.4:

```
# ->
modelAE4 <- omxSetParameters(fitADE4V,
 labels=c("SD11","SD21","SD31","SD41","SD22","SD32","SD42","SD33","SD43","SD44"),
       free=FALSE, values=c(0))
fitAE4V    <- mxTryHard( modelAE4, 20)
# <-
```

Question 2.5. Use `mxCompare()` to carry out the Likelihood ratio test that the D component is zero (`fitADE4` vs `fitAE4`) What do you conclude? And why is df=10 (`diffdf=10`)?

Answer 2.5: The D component can be dropped given any reasonable alpha (.05).

```
# ->
mxCompare(fitADE4V, fitAE4V)
# <-
```

```
> mxCompare(fitADE4, fitAE4)
  base comparison ep minus2LL    df      AIC   diffLL diffdf         p
1 ADE4       <NA> 34 60173.04 22406 60241.04       NA     NA        NA
2 ADE4       ADE4 24 60179.03 22416 60227.03 5.987076     10 0.8163479
```

Note that the df (diffdf) equals 10, because the D covariance matrix has 10 elements (4 variances, 6 covariances). Or – equivalently – the D covariance matrix is a function of 6 correlations (`"rD21","rD31","rD41","rD32","rD42","rD43"`) and 4 standard deviations (`"sD1","sD2","sD3","sD4"`) .

Question 2.6. You can see that the A correlations are much higher than the E correlations. Does that mean that A contributes more to the phenotypic correlations that does E? Check your answer against the output of the following code.

```
# ->
SA=diag(sA)%*%rA%*% diag(sA)  # A covariance matrix
SE= diag(sE)%*%rE%*% diag(sE)     # E covariance matrix
SPh=SA+SE  # expected phenotypic covariance matrix
round(SA/SPh,3)       # proportions
round(SE/SPh,3)      # proportions
# <-
```

```
> round(SA/SPh,3)       # proportions
      [,1]  [,2]  [,3]  [,4]
[1,] 0.207 0.457 0.454 0.433
[2,] 0.457 0.333 0.521 0.492
[3,] 0.454 0.521 0.233 0.464
[4,] 0.433 0.492 0.464 0.264
> round(SE/SPh,3)      # proportions
      [,1]  [,2]  [,3]  [,4]
[1,] 0.793 0.543 0.546 0.567
[2,] 0.543 0.667 0.479 0.508
[3,] 0.546 0.479 0.767 0.536
[4,] 0.567 0.508 0.536 0.736
```

Answer 3.6. No it does not mean that A contributes most to the phenotypic covariances. In fact, we see in the output above that E contribute slightly more (see the off-diagonal proportions). This is because the heritabilities are much lower than the standardized environmental variances.

Question 2.7. Does the AE IPM model fit the data relative to the 4 variate AE model? Why is df=4 (`diffdf = 4`)?

```
# ->
mxCompare(fitAE4,fitAE4IPM)
# <-
```

Answer 2.7.

```
> mxCompare(fitAE4,fitAE4IPM)

  base comparison ep minus2LL    df      AIC   diffLL diffdf         p
1 ADE4       <NA> 24 60179.03 22416 60227.03       NA     NA        NA
2 ADE4     AE_IPM 20 60180.20 22420 60220.20 1.170279      4 0.8829673
```

The model fits well. df=4 because in the saturated model we estimated 10 (ΣA) + 10 (ΣE) parameters, but now 8 + 8, the differences 20-16 = 4.

Question 2.8. Consider the A and E item residual variances. With respect to item specific A and E effect, what do you conclude? Bear in mind that the E residual variance include measurement error. The correlation between skinfold measures and ultrasound measures of subcutaneous fat is between .80 and .90 (the ultrasound measurement is the *golden standard).*

```
> diag(estTA) # A residual variances
[1] 0.024 0.154 0.028 0.010
> diag(estTE)  # E residual variances
[1] 0.590 0.557 0.603 0.453
```

Answer 2.8: The A residuals are relatively small, suggesting that there are few if any item-specific genetic influences. The E residuals are much larger, suggesting relatively strong item-specific unshared influences, which include (possibly large) measurement error.

Question 2.9. The AE IPM has 4 means, 2x4 (A and E) factor loadings and 2x4 (A and E) residual variances, i.e., 20 parameters. How many independent parameters does the CPM model have, and why is the df in `mxCompare(fitAE4IPM,fitAE4CPM)` equal to 3?

Answer 2.9. The AE CPM model has 4 means, 4 phenotypic factor loadings, 1 common A variance (common E variance is given $sA^2 + sE^2 = 1$), 4 E residuals and 4 A residuals. So that is 17. The df=3 is the differences in the number of parameters of the IPM (20) and the CPM (17): 20-17=3.
Looking at the output, however, something weird is going on! Notice how under **ep** we first have 20 estimated parameters in the IPM, and it says 18 estimated parameters in the CPM. So 20-18 is 2, but in the **diffdf** column it says 3! This is because in the CPM, we need to fix the variance of the latent variable to 1 (for identifiability) causing us to "win" one degree of freedom because we have one less parameter to estimate.

```
> mxCompare(fitAE4IPM,fitAE4CPM2)
    base comparison ep minus2LL    df      AIC   diffLL diffdf         p
1 AE_IPM       <NA> 20 60180.20 22420 60220.20       NA     NA        NA
2 AE_IPM    AE_CPM2 18 60184.32 22423 60220.32 4.119611      3 0.2488346
```

The other (constrained) AE CPM model (see Appendix) has 4 means, 4 E factor loadings, 1 A factor loading, 4 E residuals and 4 A residuals. So that is 17. The other 3 A factor loadings are constrained. The df=3 is the differences in the number of parameters of the IPM (20) and the CPM (17): 20-17=3.

Question 2.10. Does the AE CPM (version B) fit the data relative to the AE IPM? (We have already answered this question!)

```
# ->
mxCompare(fitAE4IPM,fitAE4CPM)
mxCompare(fitAE4IPM,fitAE4CPM2)# for this see appendix
# <-
```

Answer 2.10: The CPM model fits will compared to the IPM.

```
> mxCompare(fitAE4IPM,fitAE4CPM2)
    base comparison ep minus2LL    df      AIC   diffLL diffdf         p
1 AE_IPM       <NA> 20 60180.20 22420 60220.20       NA     NA        NA
2 AE_IPM    AE_CPM2 18 60184.32 22423 60220.32 4.119611      3 0.2488346
```

Question 3.11. What is the heritability of the latent variable Neuroticism, according to the fitAE4CPM2 output?

Answer 3.11: The heritability of the latent variable Neuroticism equals .465.

Question 3.12. What is the heritability of the Neuroticism sum score and why does it differ from the heritability based on the CPM (see answer to question 3.11)? The variance components can be obtained from the OpenMx output as follows `fitAE$VarComp`.

Answer 3.12:

```
> fitAE$VarComp
mxAlgebra 'VarComp'
$formula:  cbind(VA, VD, VE, VA/V, VD/V, VE/V)
$result:
              VA VD       VE        SA SD        SE
VarComp 3.920567  0 6.429901 0.3787816  0 0.6212184
```

The heritability is 0.378, which is appreciably lower that the .465. The difference lies in the fact that the Neuroticism sum score includes error, while the latent variable Neuroticism in the CPM does not.

End of practical