Heterogeneity: Sex-limitation Models

Eveline de Zeeuw

(Meike, Brad, Sarah, Hermine, Ben, Elizabeth, and most of the rest of the faculty that has contributed bits and pieces to various versions of this talk)

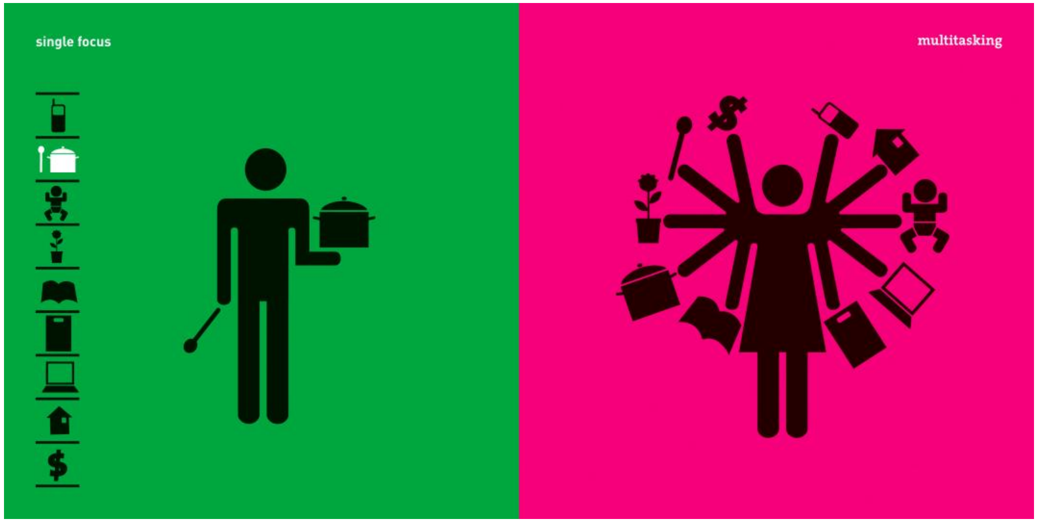
COPY FILES FROM:

Faculty/eveline/2018/sexLim

Heterogeneity

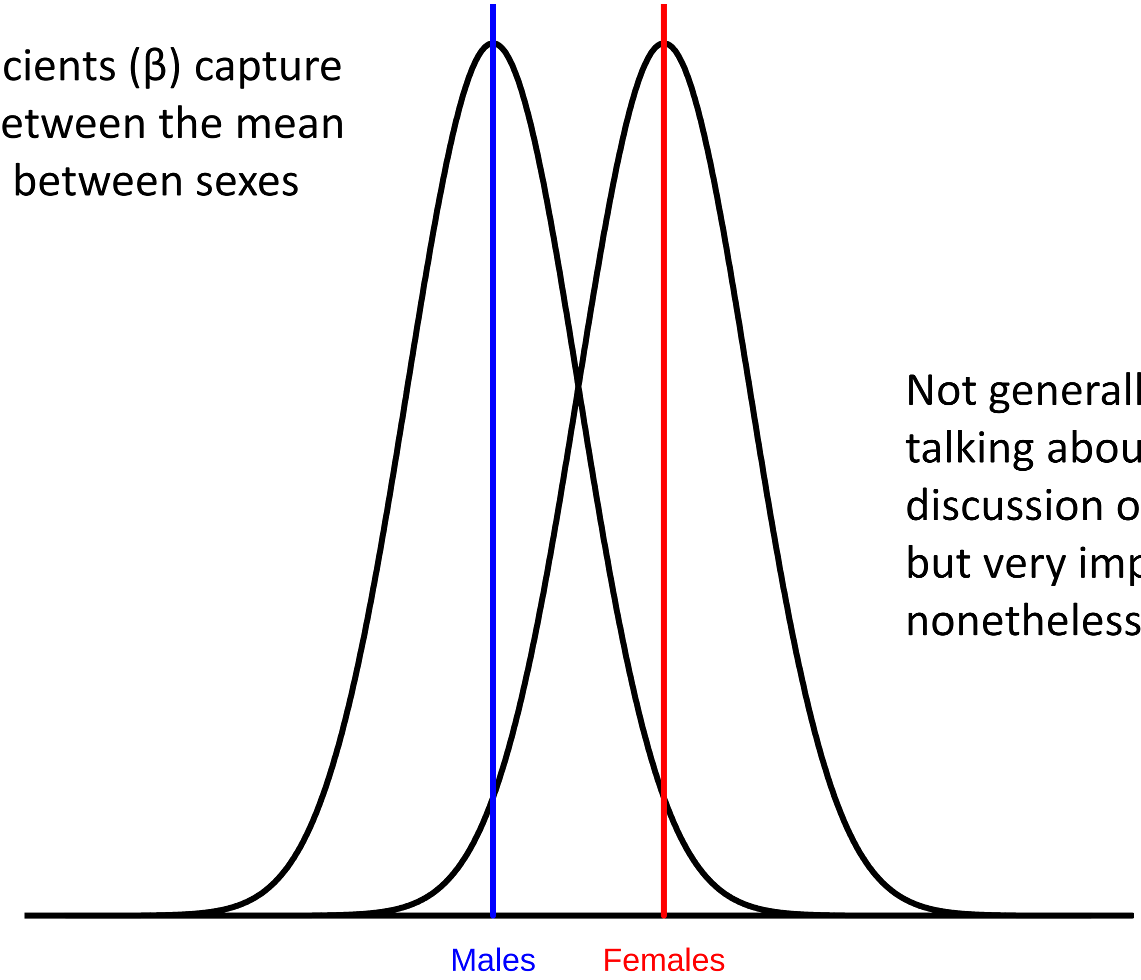
- Univariate Analysis:
  - What are the contributions of additive genetic, dominance/shared environmental and unique environmental factors to the variance?
- Heterogeneity:
  - Are the contributions of genetic and environmental factors equal for different groups, sex, cohort, SES, age, environmental exposure, etc.?

Sex Limitation = Sex Differences



Mean Differences

Regression coefficients (β) capture the differences between the mean levels of the trait between sexes

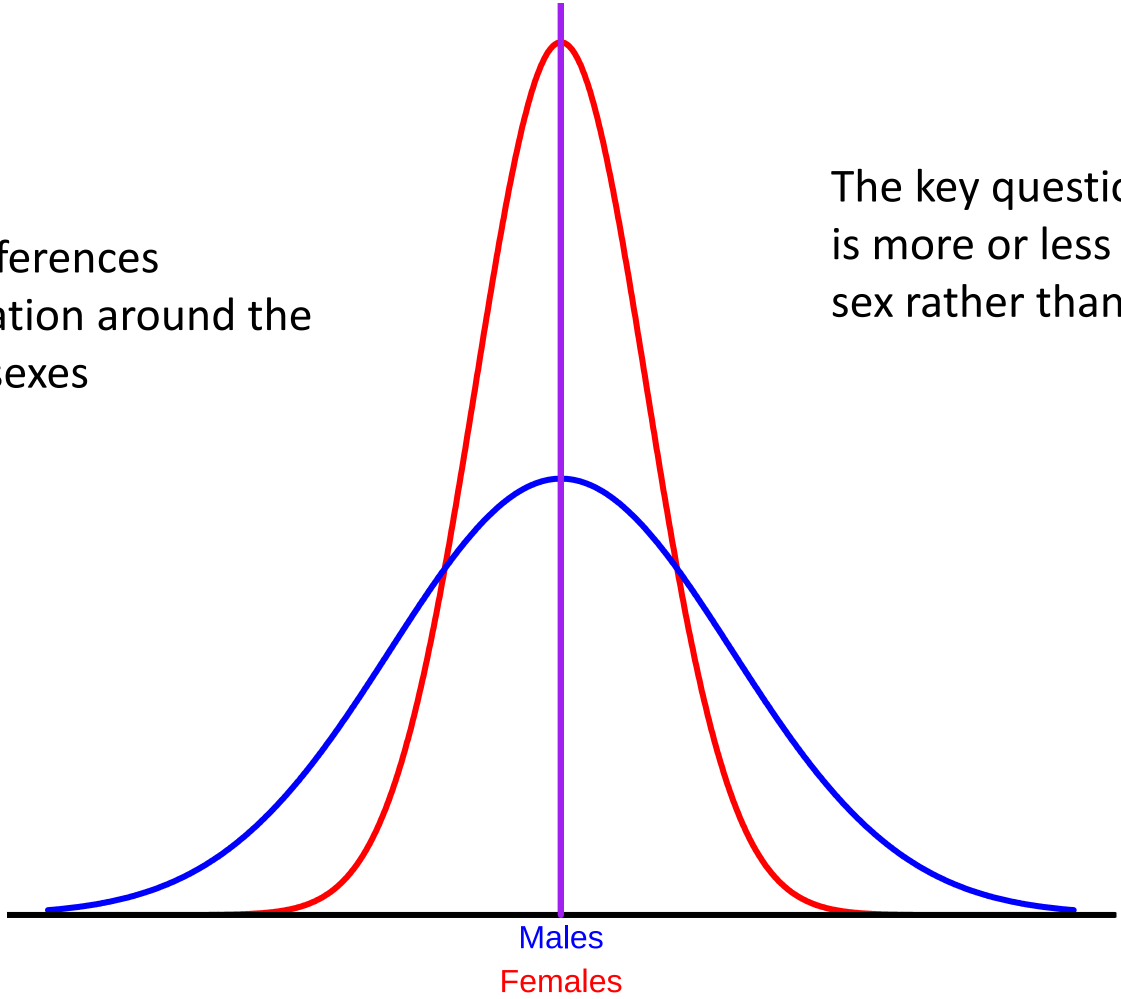


Not generally what we are talking about when discussion of sex limitation, but very important nonetheless.

Variance Differences

σ^2 capture the differences between the variation around the mean across the sexes

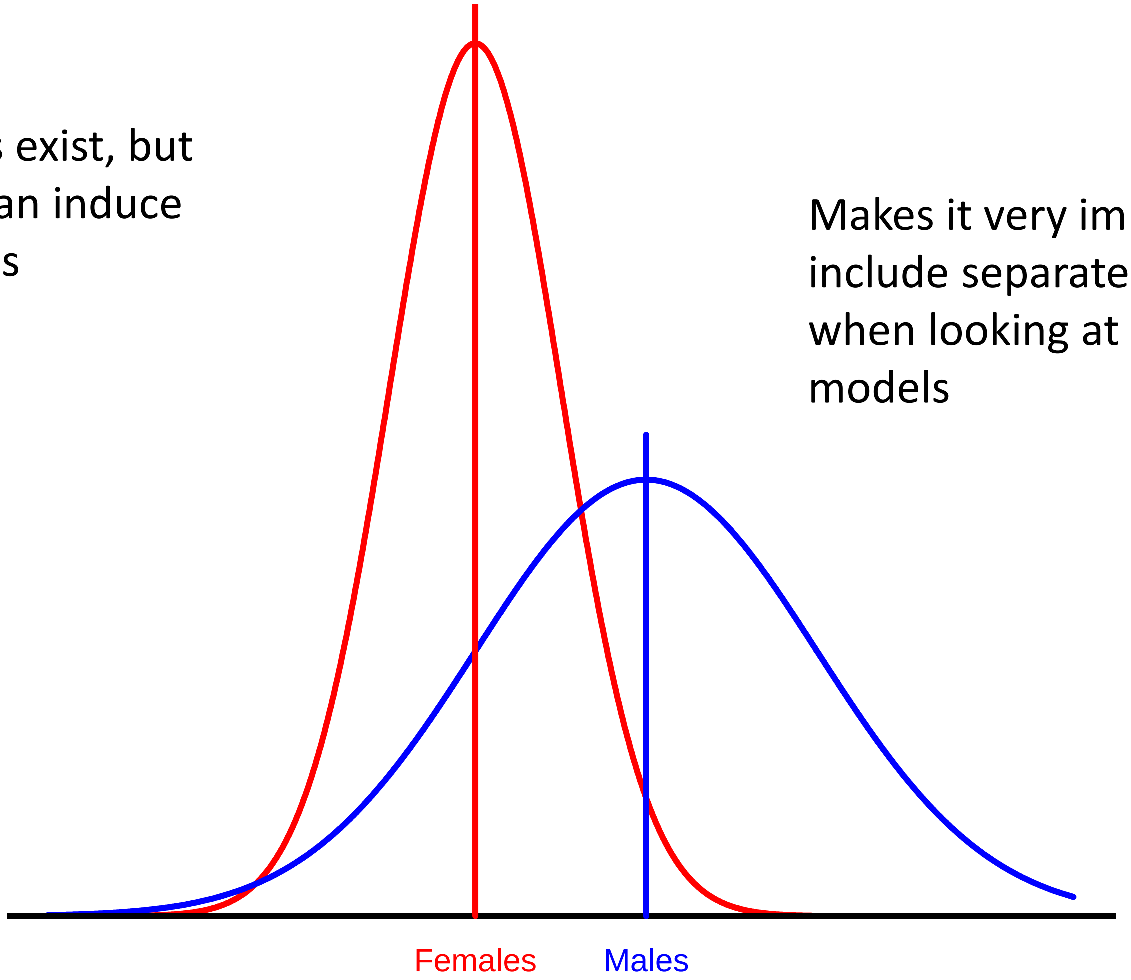
The key question is why there is more or less variation in one sex rather than the other



Both Mean and Variance Differences

If mean differences exist, but are ignored, they can induce variance differences

Makes it very important to include separate means for sex when looking at sex limitation models



Causes of Variance Differences

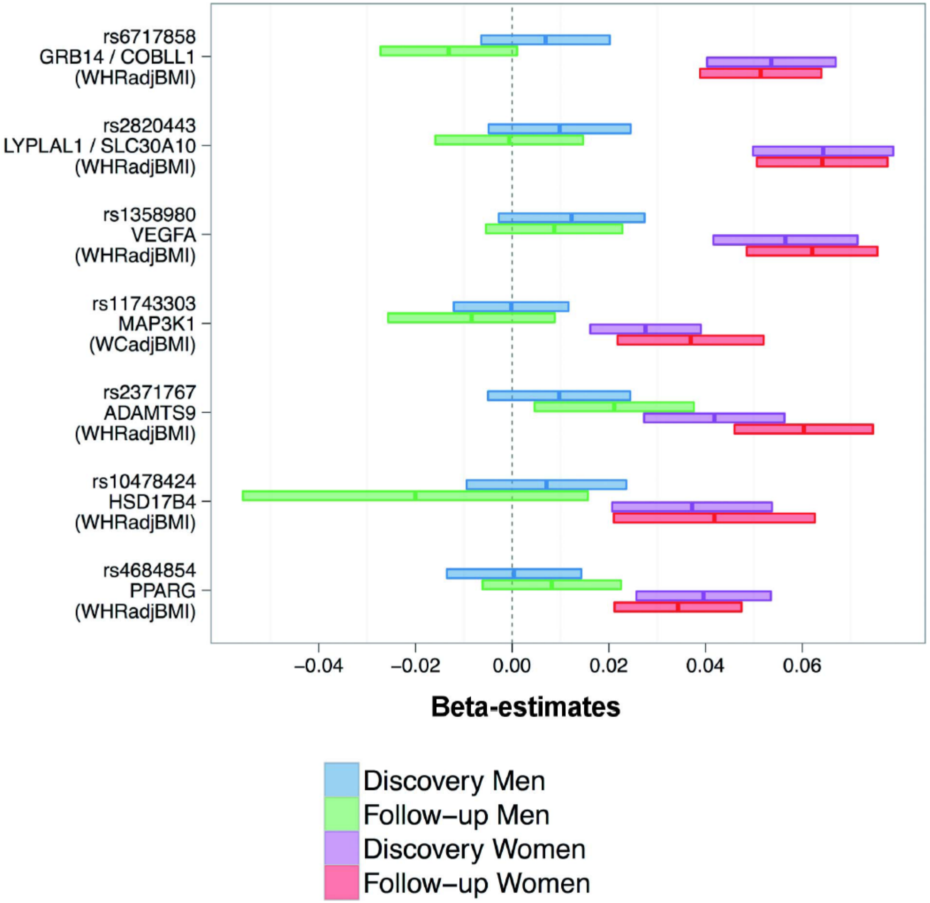
- Independent variables (millions of them) can influence the trait to different extents in different groups

or

- Different independent variables can influence the trait in the different groups.

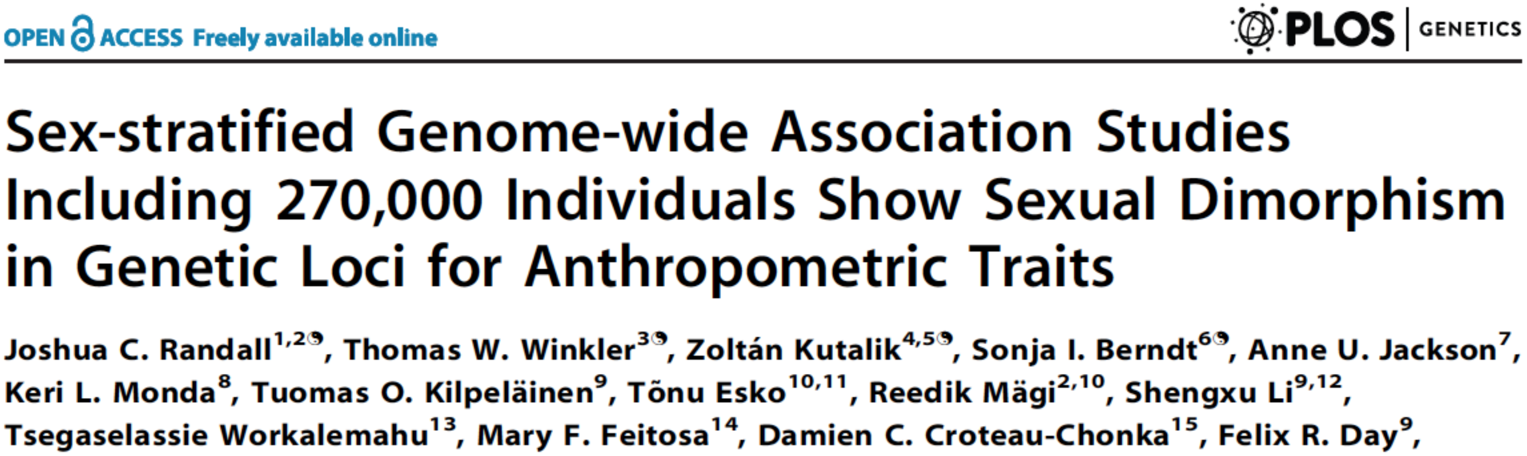
The Language of Heterogeneity

- Are the differences due to differences in the magnitude of the effects (quantitative differences)?
  - Is the contribution of genetic/environmental factors greater/smaller in males than in females?
- Are the differences due to differences in the nature of the effects (qualitative differences)?
  - Are there different genetic/environmental factors influencing the trait in males and females?



On all of the SNPs presented, women are affected by the polymorphism, while men are not.

Ergo, different genes "cause" the trait in males and females!
Or
Molecular evidence of qualitative sex limitation



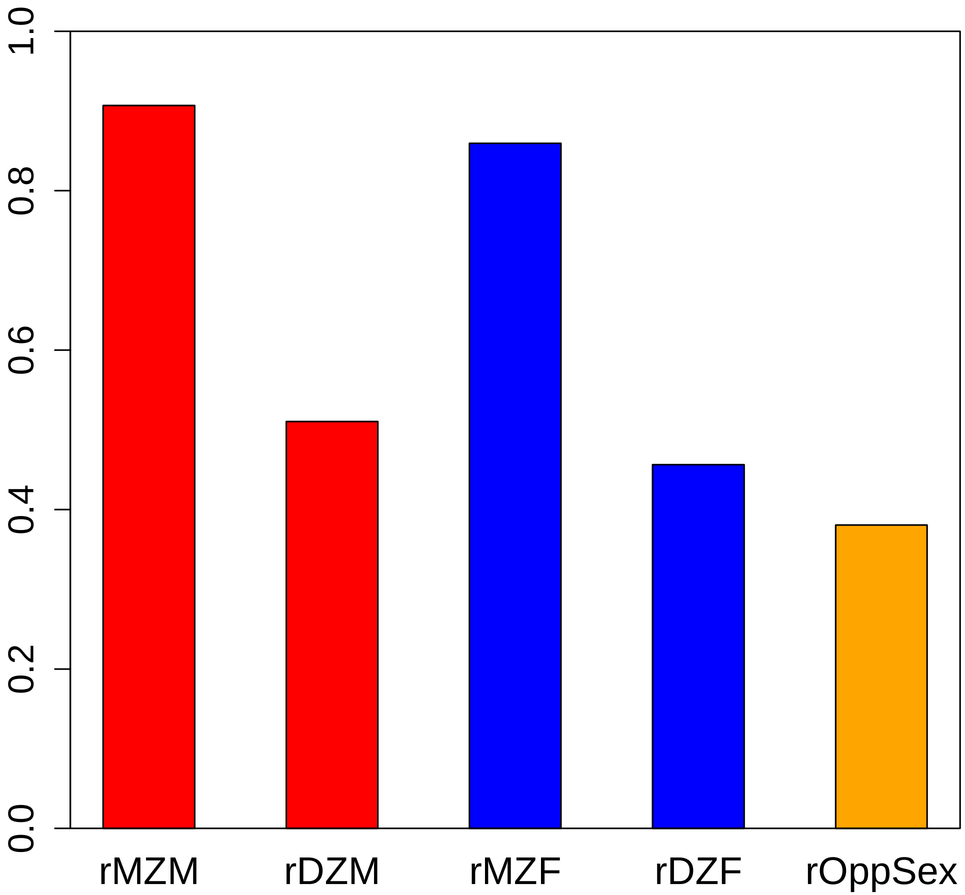
OPEN ACCESS Freely available online

PLOS | GENETICS

Sex-stratified Genome-wide Association Studies Including 270,000 Individuals Show Sexual Dimorphism in Genetic Loci for Anthropometric Traits

Joshua C. Randall[1,2☯], Thomas W. Winkler[3☯], Zoltán Kutalik[4,5☯], Sonja I. Berndt[6☯], Anne U. Jackson[7], Keri L. Monda[8], Tuomas O. Kilpeläinen[9], Tõnu Esko[10,11], Reedik Mägi[2,10], Shengxu Li[9,12], Tsegaselassie Workalemahu[13], Mary F. Feitosa[14], Damien C. Croteau-Chonka[15], Felix R. Day[9],

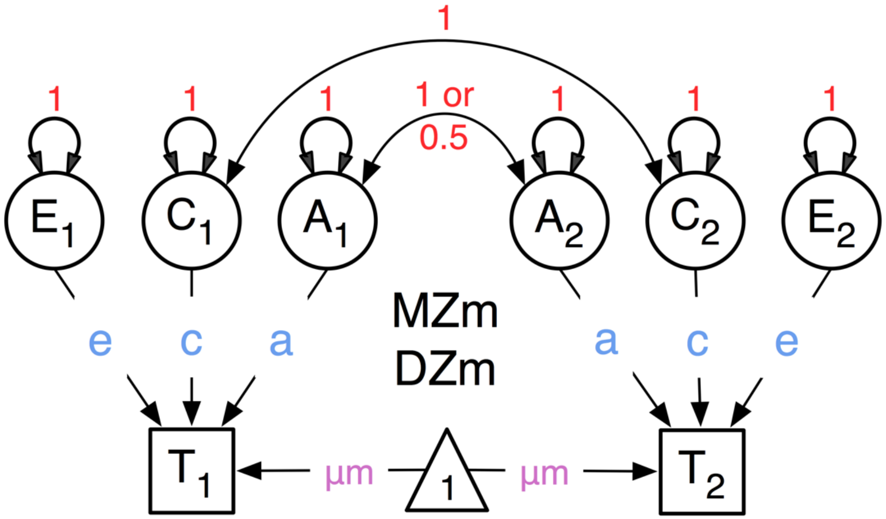
Look at the Correlations!

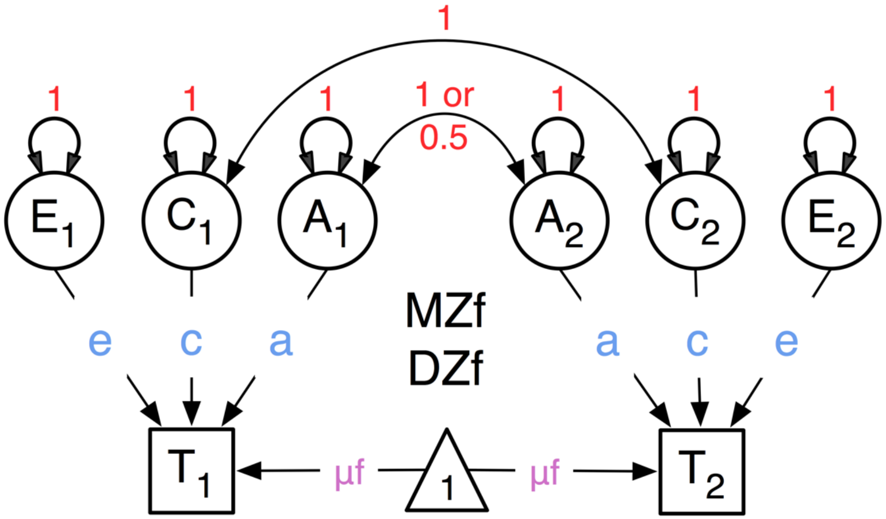


Non Sex-Limitation Model

- No heterogeneity
- The same proportion (%) of variance due to A, C, E equal between groups
- Total variance equal between groups
  - $V_m = V_f$
- Variance Components are equal between groups
  - $A_m = A_f$
  - $C_m = C_f$
  - $E_m = E_f$

Non Sex-Limitation Model





	Male	Male
Male	$a^2 + c^2 + e^2$	$(½)a^2 + c^2$
Male	$(½)a^2 + c^2$	$a^2 + c^2 + e^2$

	Female	Female
Female	$a^2 + c^2 + e^2$	$(½)a^2 + c^2$
Female	$(½)a^2 + c^2$	$a^2 + c^2 + e^2$

Multiple Non Sex-Limitation Models

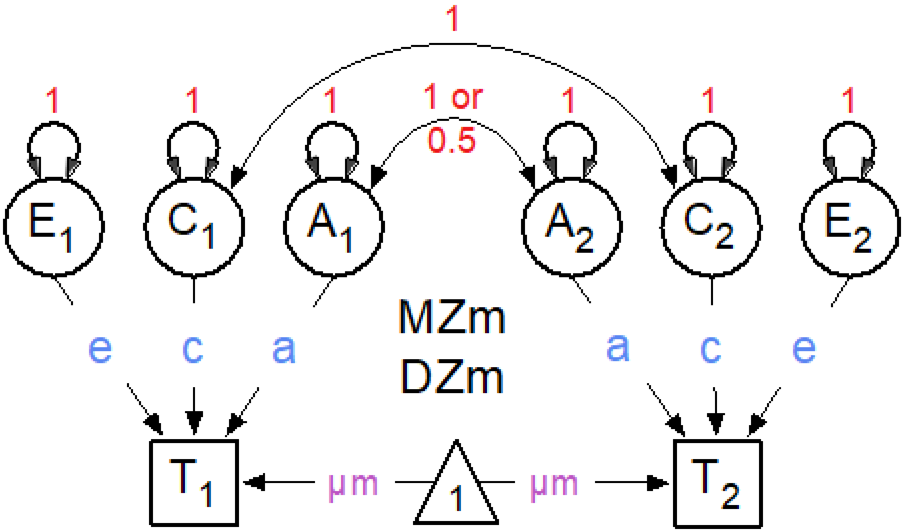
- It does not test whether the heterogeneity is significant
- It does not attempt to explain the sex differences
- It does not include useful information from dizygotic opposite-sex twins

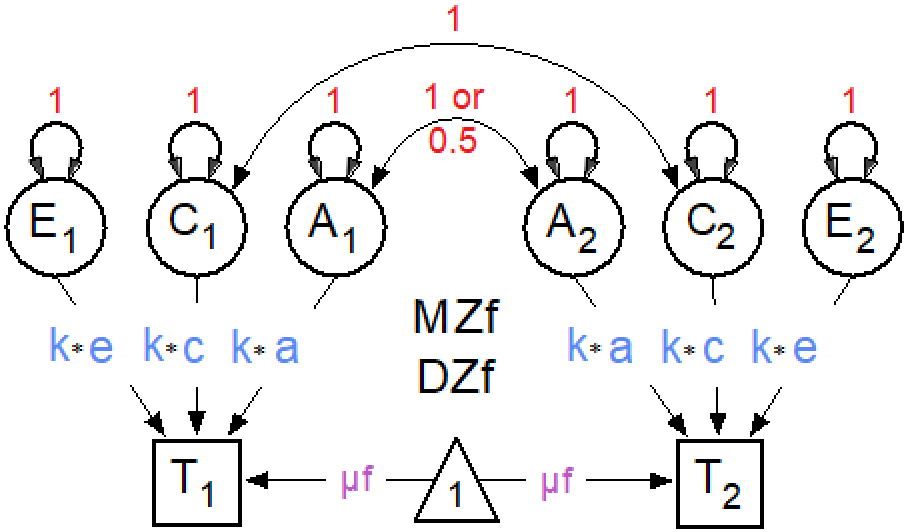
Scalar Sex-limitation Model

- Scalar sex-limitation (a quantitative model)
- The proportion (%) of variance due to A, C, E alters by a scalar (single value)
- Total variance not equal between groups
  - Vm = k* Vf
  - Am = k* Af
  - Cm = k* Cf
  - Em = k* Ef

k is scalar

Scalar Sex-limitation Model





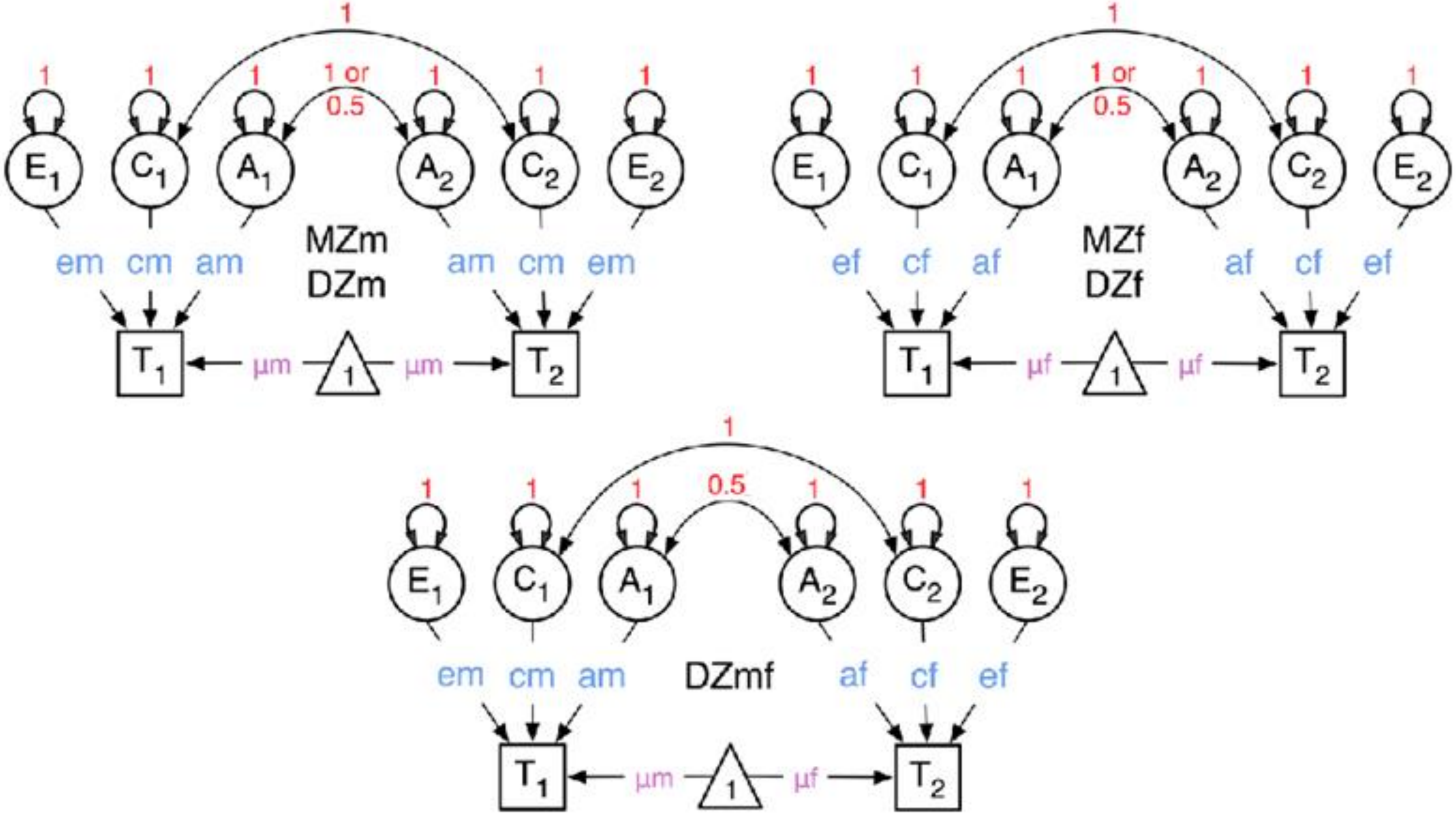
	Male	Male
Male	$a^2 + c^2 + e^2$	$(½)a^2 + c^2$
Male	$(½)a^2 + c^2$	$a^2 + c^2 + e^2$

	Female	Female
Female	$k(a^2 + c^2 + e^2)$	$k((½)a^2 + c^2)$
Female	$k((½)a^2 + c^2)$	$k(a^2 + c^2 + e^2)$

Non-scalar Sex-limitation Model

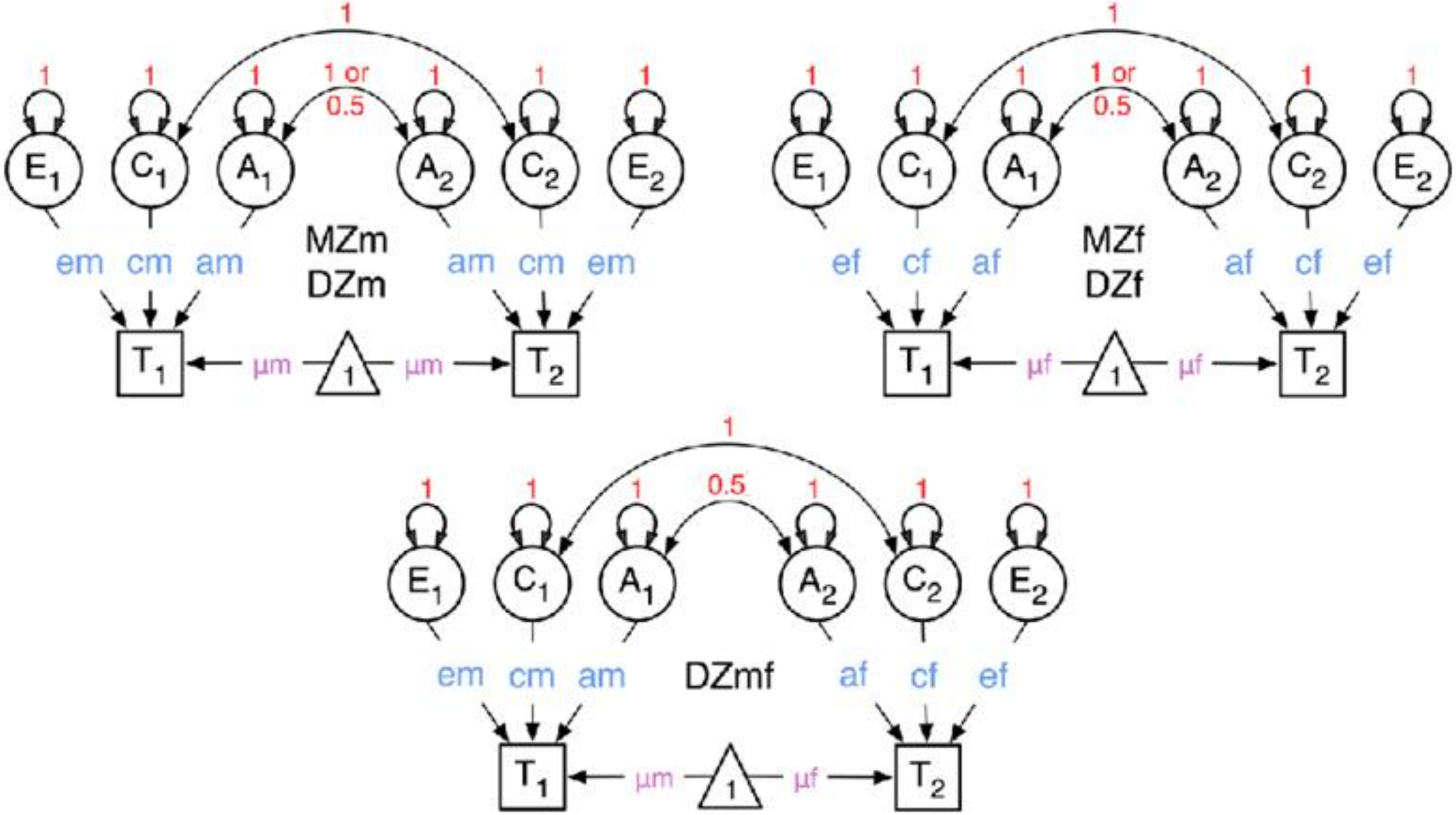
- Non-scalar sex-limitation with opposite sex pairs (a quantitative model)
- The total variance and proportion (%) of variance due to A, C, E are estimated separately for each group
  - $Vm \neq Vf$
  - $Am \neq Af$
  - $Cm \neq Cf$
  - $Em \neq Ef$

Non-scalar Sex-limitation Model



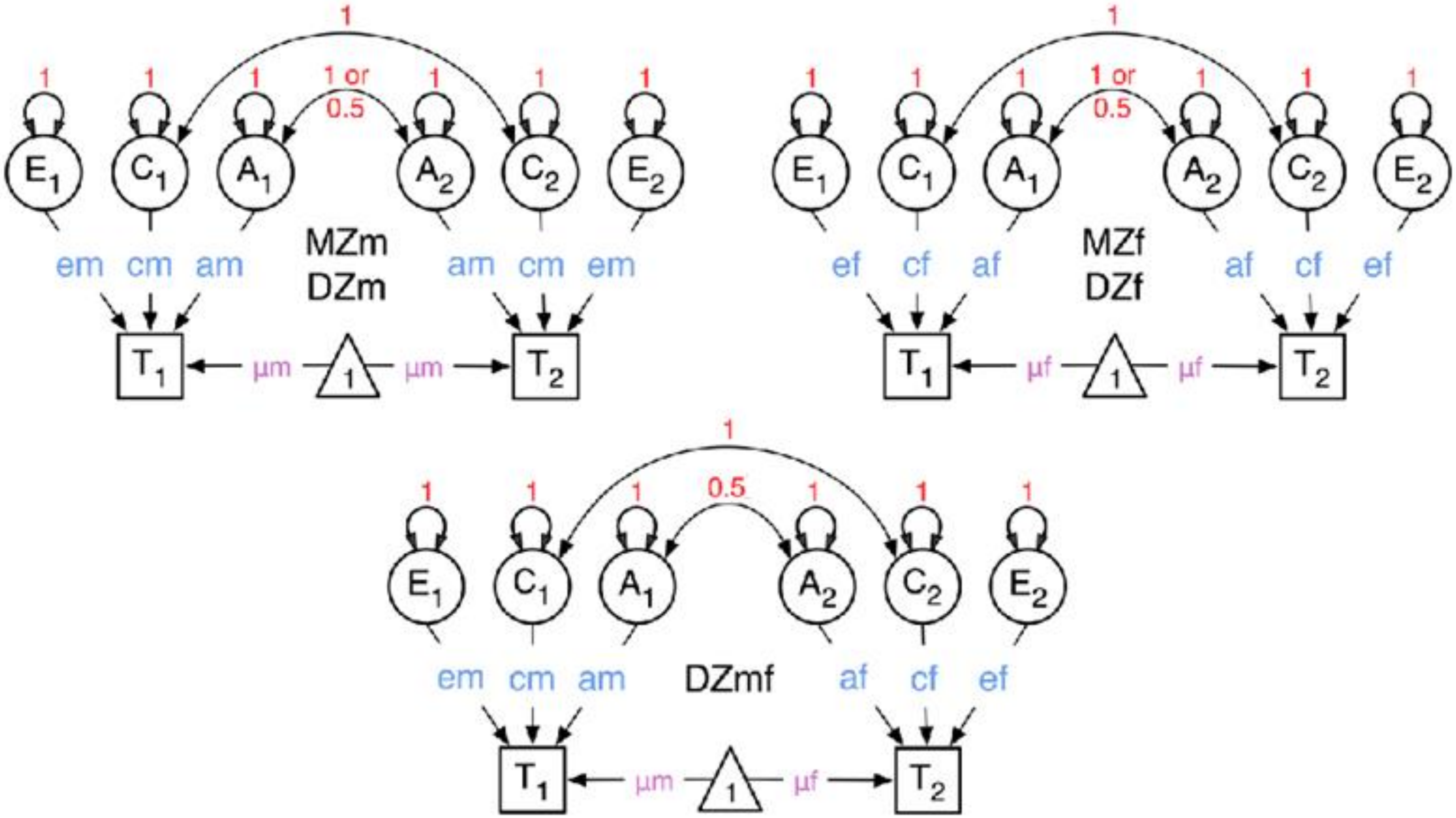
	Male	Male
Male	$a_m{}^2 + c_m{}^2 + e_m{}^2$	$(½)a_m{}^2 + c_m{}^2$
Male	$(½)a_m{}^2 + c_m{}^2$	$a_m{}^2 + c_m{}^2 + e_m{}^2$

Non-scalar Sex-limitation Model



	Male	Male
Male	$a_f^2 + c_f^2 + e_f^2$	$(½)a_f^{\ 2}+ c_f^2$
Male	$(½)a_f^2 + c_f^2$	$a_f^2 + c_f^2 + e_f^2$

Non-scalar Sex-limitation Model

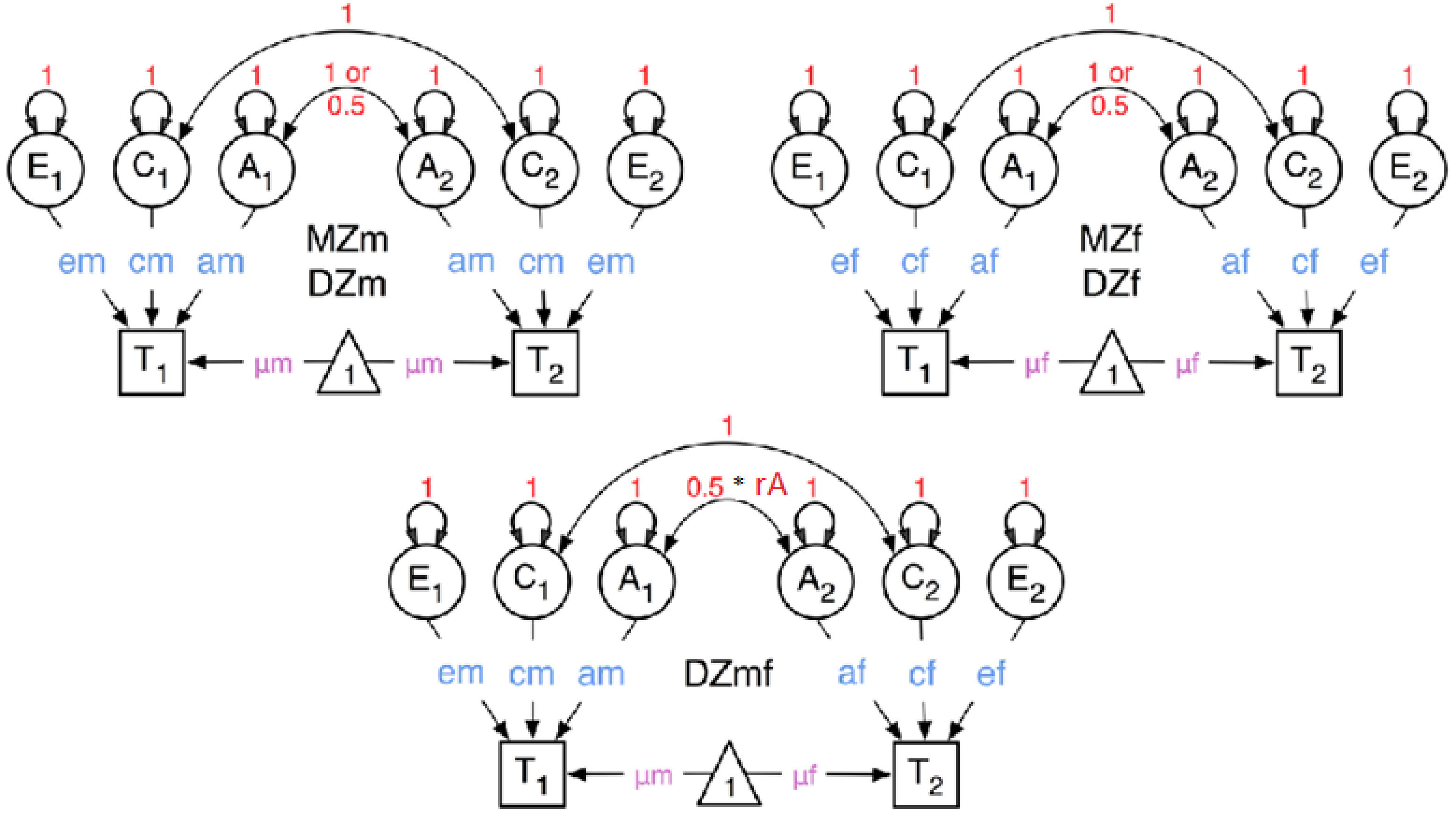


	Male	Female
Male	$a_f^2 + c_f^2 + e_f^2$	$\frac{1}{2}a_m a_f + c_m c_f$
Female	$\frac{1}{2}a_f a_m + c_f c_m$	$a_f^2 + c_f^2 + e_f^2$

General Non-scalar Sex-limitation Model

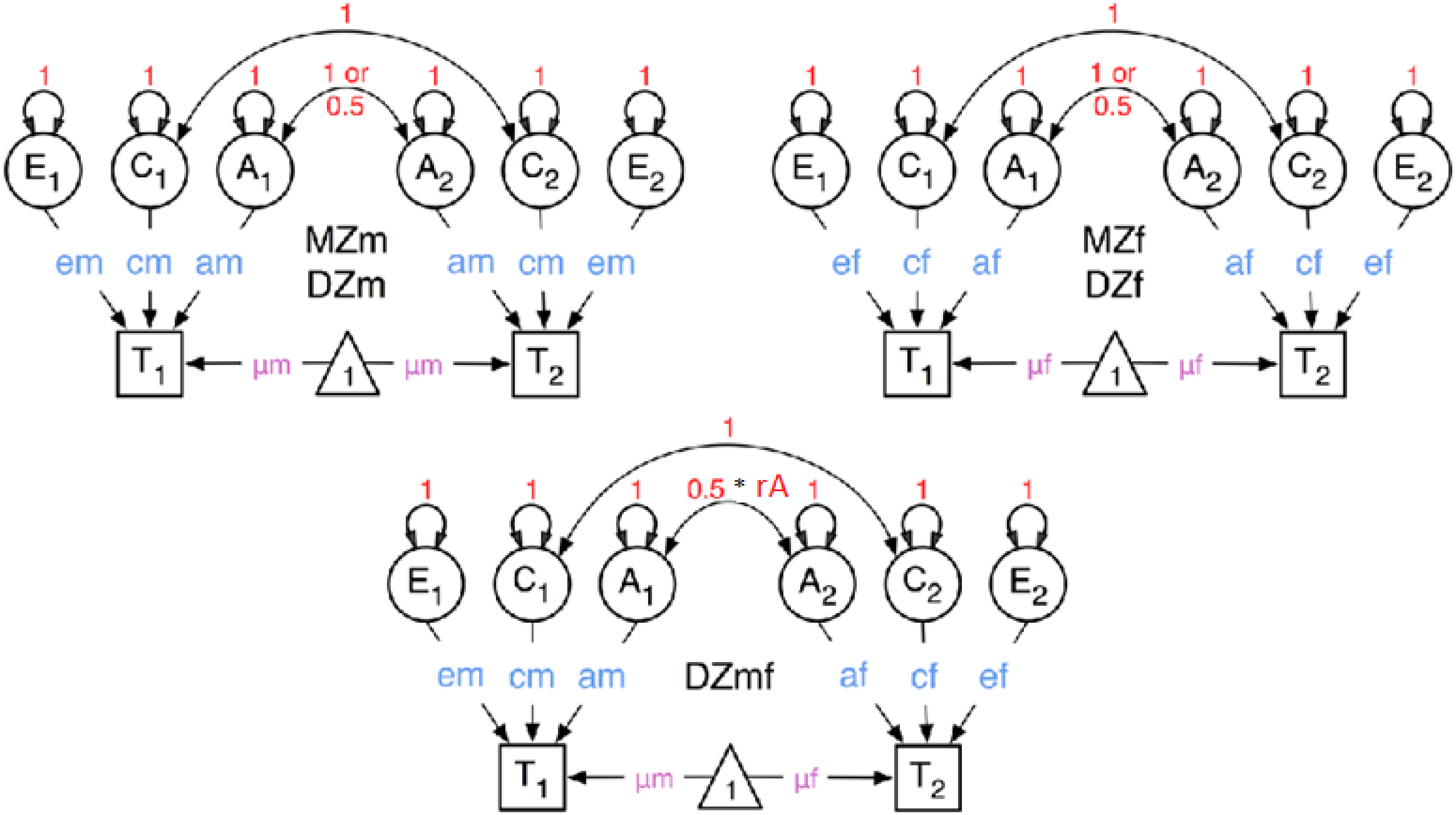
- Non-scalar sex-limitation with opposite sex pairs (a quantitative & qualitative model)
- The total variance and proportion (%) of variance due to A, C, E are estimated separately for each group
  - Vm ≠ Vf
  - Am ≠ Af
  - Cm ≠ Cf
  - Em ≠ Ef
- Genetic correlation (by means of rA) between DZO twins is estimated freely

General Non-scalar Sex-limitation Model



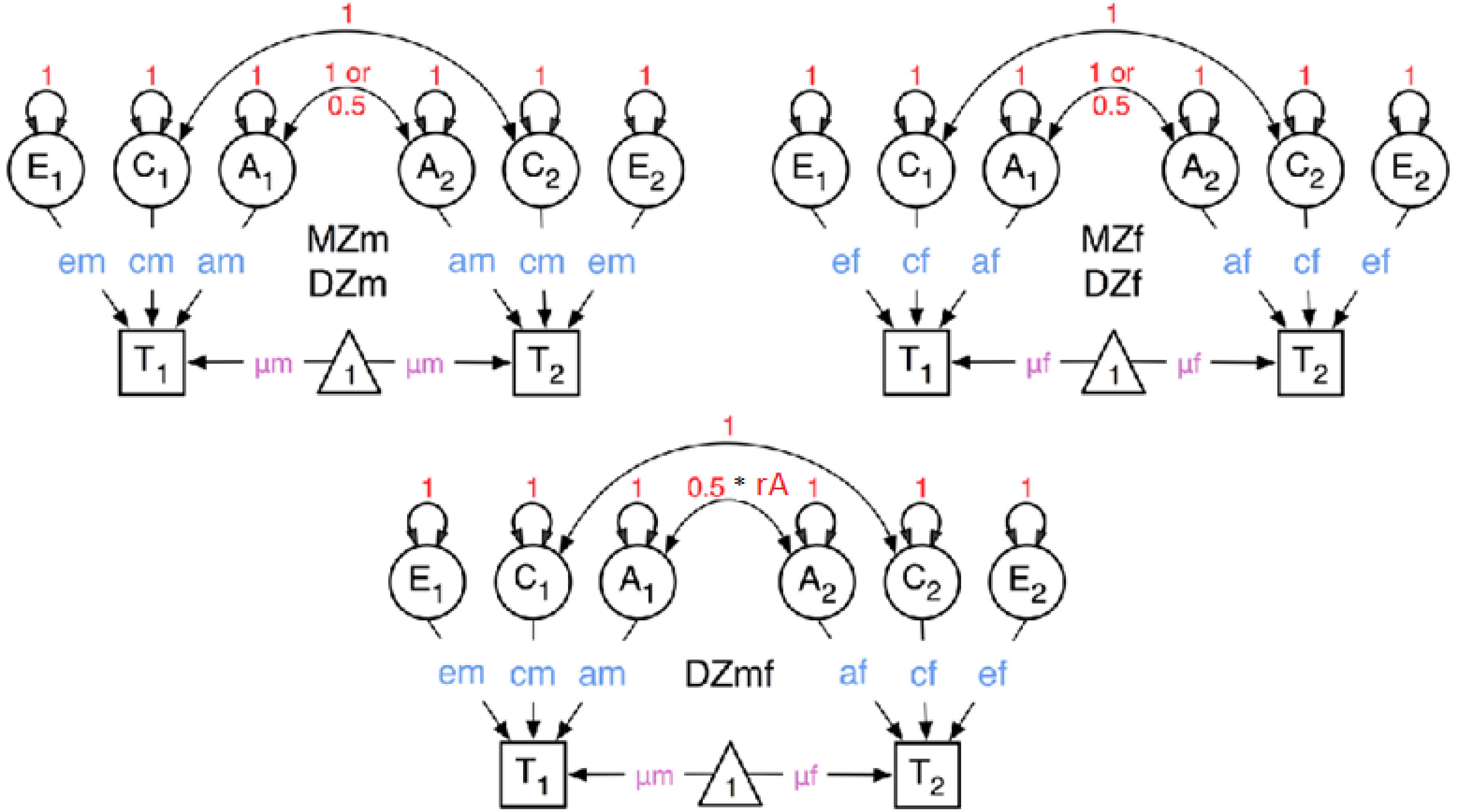
	Male	Male
Male	$a_m^2 + c_m^2 + e_m^2$	$(½)a_m^2 + c_m^2$
Male	$(½)a_m^2 + c_m^2$	$a_m^2 + c_m^2 + e_m^2$

General Non-scalar Sex-limitation Model



	Male	Male
Male	$a_f^2 + c_f^2 + e_f^2$	$(½)a_f{}^2 + c_f^2$
Male	$(½)a_f^2 + c_f^2$	$a_f^2 + c_f^2 + e_f^2$

General Non-scalar Sex-limitation Model



	Male	Female
Male	$a_m^2 + c_m^2 + e_m^2$	$\frac{1}{2} r_A a_m a_f + c_m c_f$
Female	$\frac{1}{2} r_A a_f a_m + c_f c_{mf}$	$a_f^2 + c_f^2 + e_f^2$

Sex Limitation Models

Model Type	**Data Requirements**
Non Sex-Limitation Model (Classical Twin Model)	MZ & DZ Twins
Scalar Sex-limitation Model	MZm, MZf, DZm & DZf Twins
Non-scalar Sex-Llmitation Model	MZm, MZf, DZm, DZf & DZo Twins
General Non-scalar Sex-limitation Model	MZm, MZf, DZm, DZf & DZo Twins

Example

bent u meerling en staat u nog niet ingeschreven bij het NTR? klik hier om u aan te melden en mee te doen aan ons onderzoek

DOUBLE DUTCH: fotoportretten van tweelingen

De opening op dinsdag 2 februari was een groot succes! De fototentoonstelling in het Hoofdgebouw van de Vrije Universiteit is nog vrij te bezichtigen tot en met 30 juni.

Op 17 maart wordt het fotoboek gepresenteerd dat bij de fototentoonstelling hoort. Het boek kan nu al worden besteld, het wordt dan bezorgd of voor u klaargelegd om 17 maart op te halen. Klik op de foto voor meer informatie.

Welkom op de website van het Nederlands Tweelingen Register
klik hier om meer te lezen over het NTR

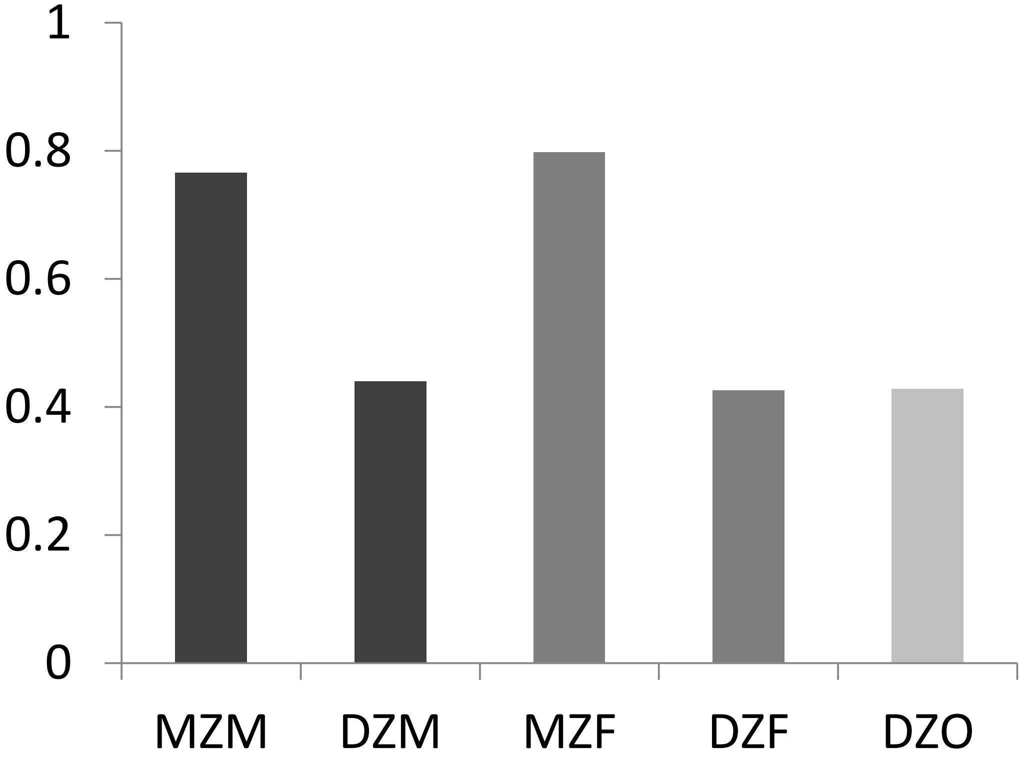
NTR-deelnemers kunnen hier klikken om hun e-mailadres aan ons door te geven

- Questions about the NTR? Click here to see our FAQ (in Dutch only)
- Nieuwe genen ontdekt die cholesterolgehalte en lichaamslengte kunnen beïnvloeden
- NTR krijgt prachtige bijdrage uit NWO-groot programma
- NTR-onderzoeker én tweeling op Radio 1 over onderzoek naar Alzheimer
- Honderdste tweeling gescand voor onderzoek naar oorzaken Alzheimer
- Klik hier om het boek 'Tweelingonderzoek - Wat meerlingen vertellen over de mens' te lezen
- Meike Bartels bij de Universiteit van Nederland over Geluk (via YouTube)
- The Guardian: Top 10 twins in children's books

Measurement Instrument

- Educational achievement test
  - Total score: Ranked from highest (1) to lowest (50) and standardized
  - Subscores: Mathematics, Language, Study Skills, World Studies
- Administered in last grade of primary school (~age 12)
- 1181 MZM, 1185 DZM, 1445 MZF, 1175 DZF and 2396 DZO pairs

Twin Correlations



Practical

1. Open oneACEcm (left) or oneACEcf (right)
2. Walk through the script
3. Run the script
4. Report the mean and the estimates for the variance components
5. Be sure that you know what you are doing

Results

	Boys	**Girls**
Mean	-.02	.11
A	.69	.72
C	.09	.08
E	.22	.20

Practical

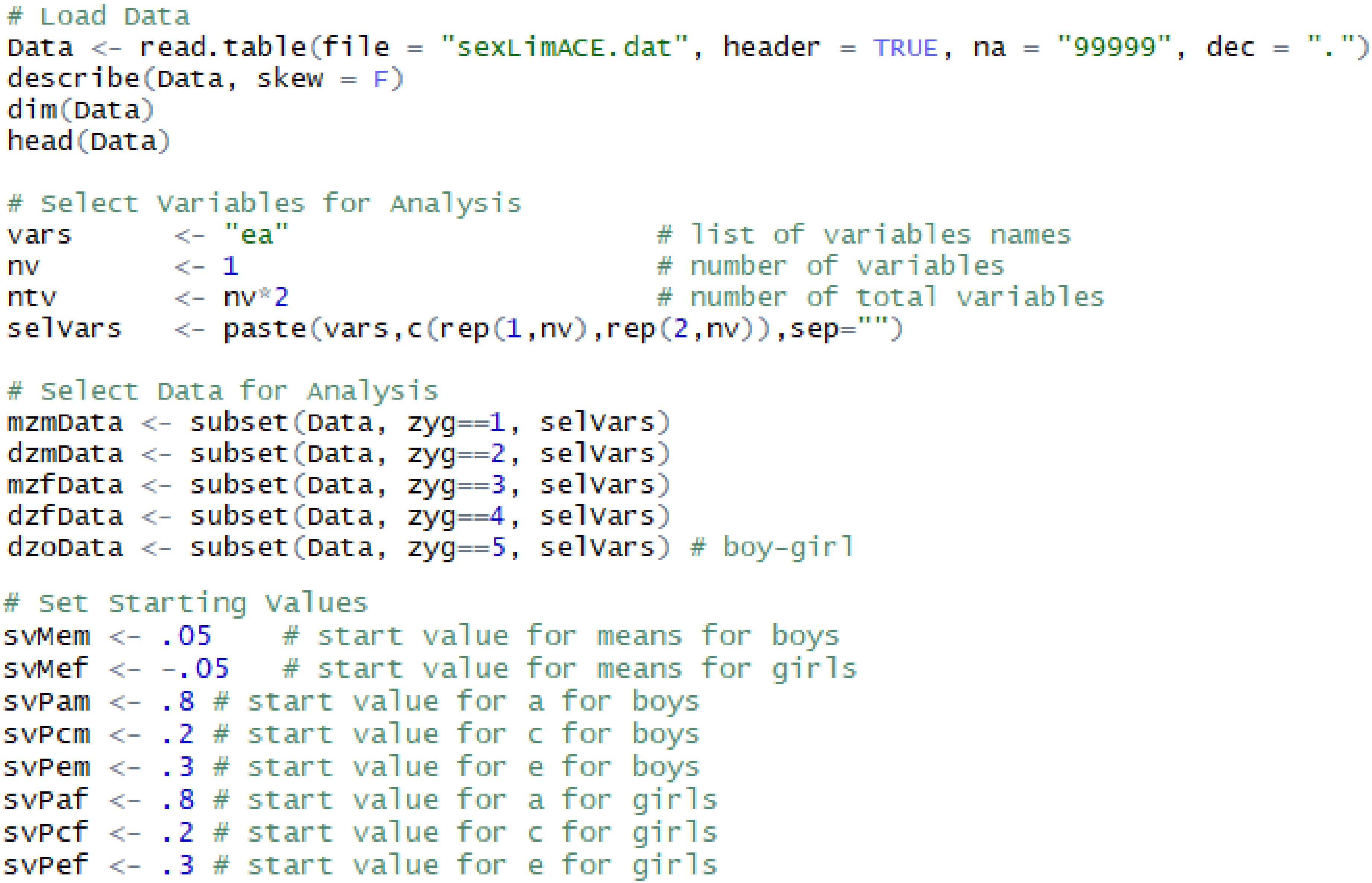
1. Open oneACE5c
2. Walk through the first part of the script
3. Run it
4. You run the submodels
    - For each model fill in the question marks
5. Be sure that you know what you are doing

Results Model Fitting

Model	EP	-2LL	df	AIC	Δ-2LL	Δdf	P
oneACErq5c	9	35764.08	13678	8408.08	-	-	-
oneACEq5c	8	35764.19	13679	8406.19	.11	1	.738
oneACE5c	5	35766.18	13682	8402.18	2.11	4	.716

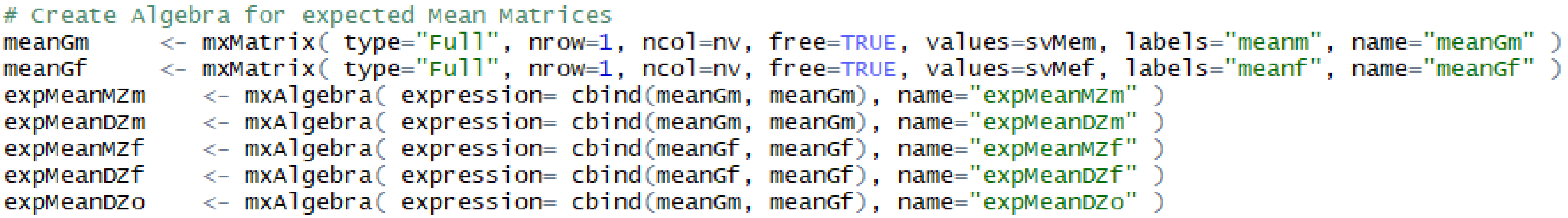
Model	EP	-2LL	df	AIC	Δ-2LL	Δdf	P
oneACE5c	5	35766.18	13682	8402.18	-	-	-
oneAE5c	4	35775.05	13683	8409.05	8.87	1	.003
oneCE5c	4	36462.77	13683	9096.77	696.58	1	<.001
oneE5c	3	38865.18	13684	11497.18	3098.99	2	<.001

Data



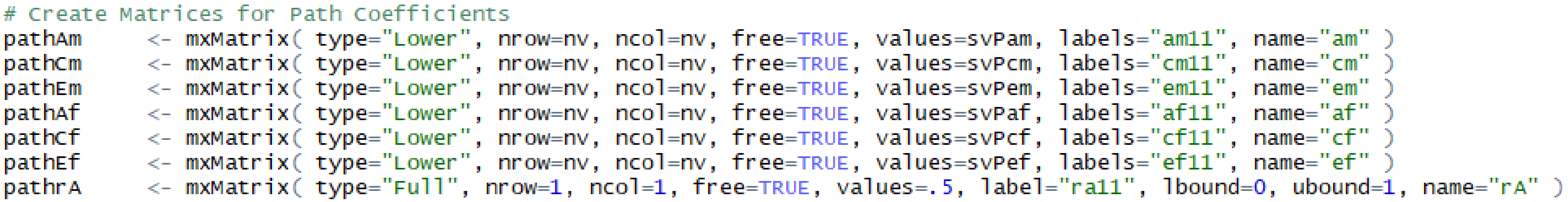
```
# Load Data
Data <- read.table(file = "sexLimACE.dat", header = TRUE, na = "99999", dec = ".")
describe(Data, skew = F)
dim(Data)
head(Data)

# Select Variables for Analysis
vars      <- "ea"                     # list of variables names
nv        <- 1                        # number of variables
ntv       <- nv*2                     # number of total variables
selVars   <- paste(vars,c(rep(1,nv),rep(2,nv)),sep="")

# Select Data for Analysis
mzmData <- subset(Data, zyg==1, selVars)
dzmData <- subset(Data, zyg==2, selVars)
mzfData <- subset(Data, zyg==3, selVars)
dzfData <- subset(Data, zyg==4, selVars)
dzoData <- subset(Data, zyg==5, selVars) # boy-girl

# Set Starting Values
svMem <- .05    # start value for means for boys
svMef <- -.05   # start value for means for girls
svPam <- .8 # start value for a for boys
svPcm <- .2 # start value for c for boys
svPem <- .3 # start value for e for boys
svPaf <- .8 # start value for a for girls
svPcf <- .2 # start value for c for girls
svPef <- .3 # start value for e for girls
```

Means

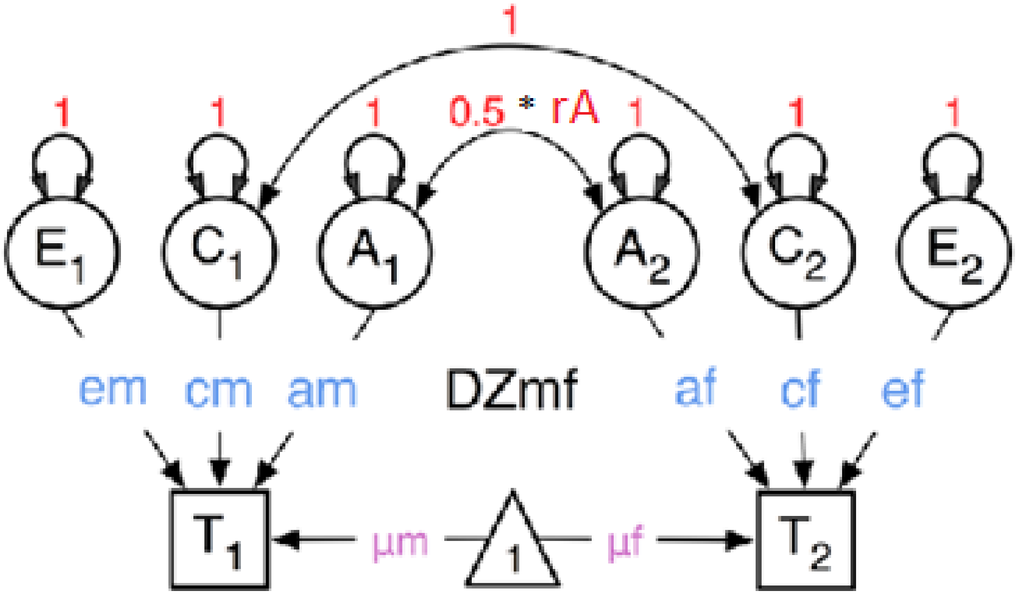


```
# Create Algebra for expected Mean Matrices
meanGm     <- mxMatrix( type="Full", nrow=1, ncol=nv, free=TRUE, values=svMem, labels="meanm", name="meanGm" )
meanGf     <- mxMatrix( type="Full", nrow=1, ncol=nv, free=TRUE, values=svMef, labels="meanf", name="meanGf" )
expMeanMZm    <- mxAlgebra( expression= cbind(meanGm, meanGm), name="expMeanMZm" )
expMeanDZm    <- mxAlgebra( expression= cbind(meanGm, meanGm), name="expMeanDZm" )
expMeanMZf    <- mxAlgebra( expression= cbind(meanGf, meanGf), name="expMeanMZf" )
expMeanDZf    <- mxAlgebra( expression= cbind(meanGf, meanGf), name="expMeanDZf" )
expMeanDZo    <- mxAlgebra( expression= cbind(meanGm, meanGf), name="expMeanDZo" )
```

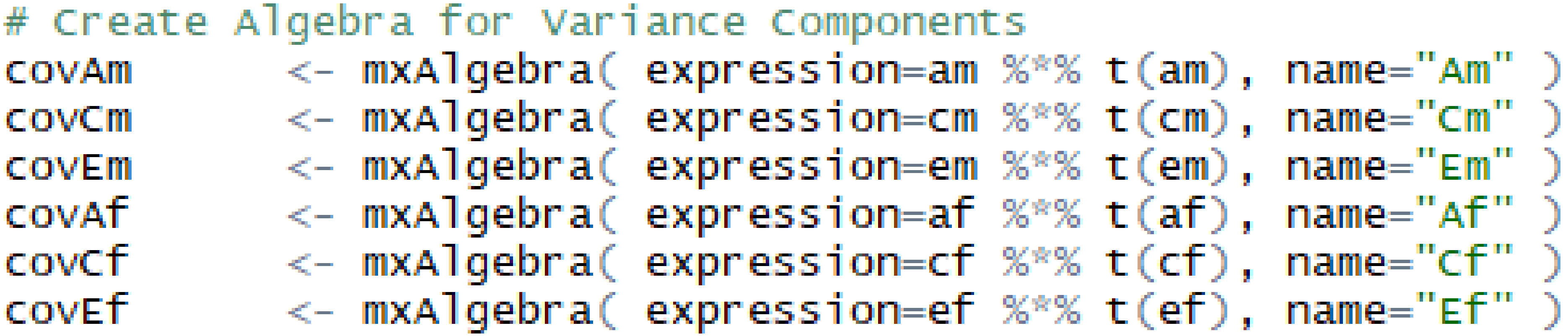
Path Coefficients



```
# Create Matrices for Path Coefficients
pathAm     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPam, labels="am11", name="am" )
pathCm     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPcm, labels="cm11", name="cm" )
pathEm     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPem, labels="em11", name="em" )
pathAf     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPaf, labels="af11", name="af" )
pathCf     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPcf, labels="cf11", name="cf" )
pathEf     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPef, labels="ef11", name="ef" )
pathrA     <- mxMatrix( type="Full", nrow=1, ncol=1, free=TRUE, values=.5, label="ra11", lbound=0, ubound=1, name="rA" )
```

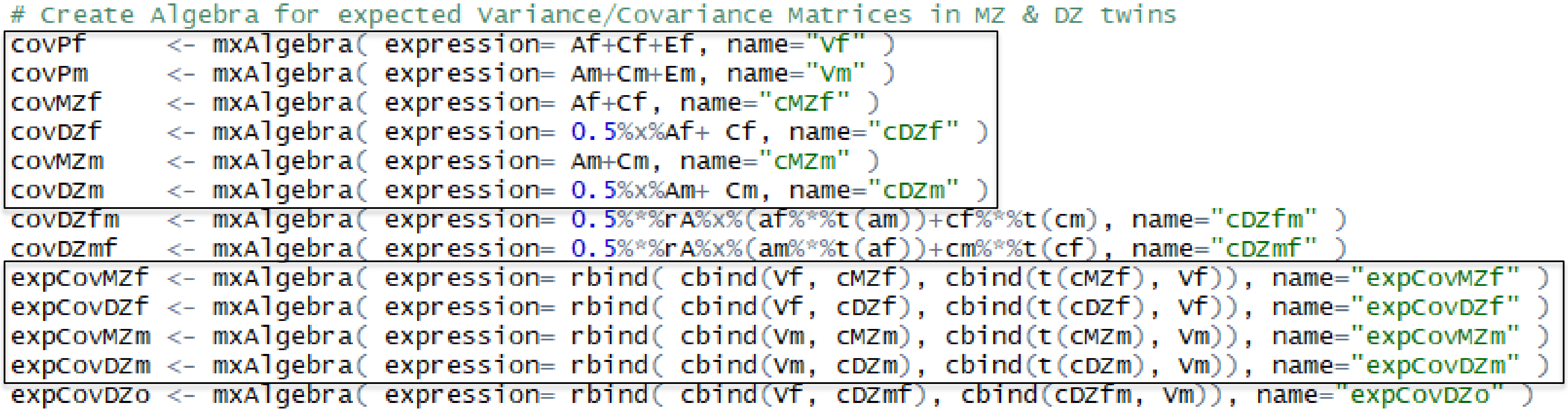


Variance Components

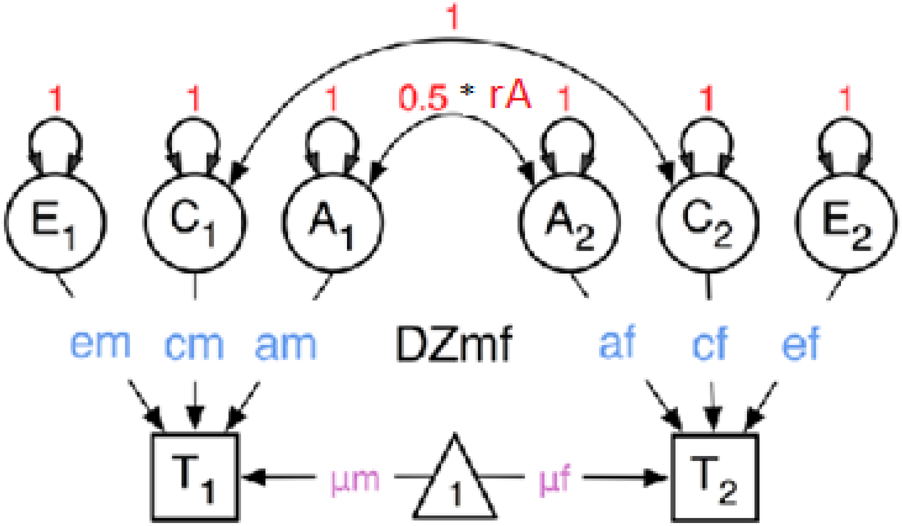


```
# Create Algebra for Variance Components
covAm        <- mxAlgebra( expression=am %*% t(am), name="Am" )
covCm        <- mxAlgebra( expression=cm %*% t(cm), name="Cm" )
covEm        <- mxAlgebra( expression=em %*% t(em), name="Em" )
covAf        <- mxAlgebra( expression=af %*% t(af), name="Af" )
covCf        <- mxAlgebra( expression=cf %*% t(cf), name="Cf" )
covEf        <- mxAlgebra( expression=ef %*% t(ef), name="Ef" )
```

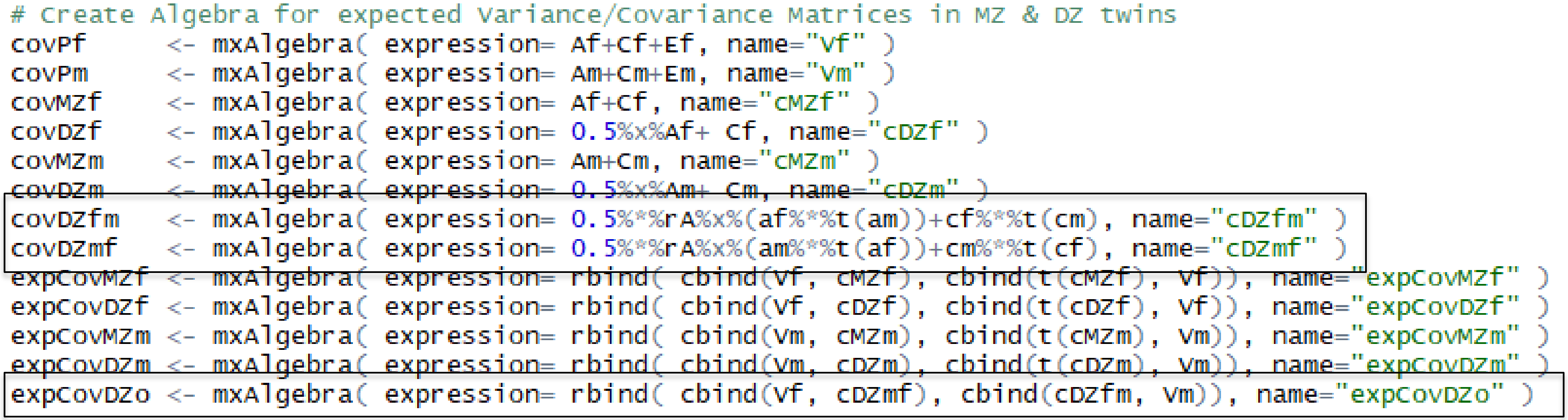
Covariances



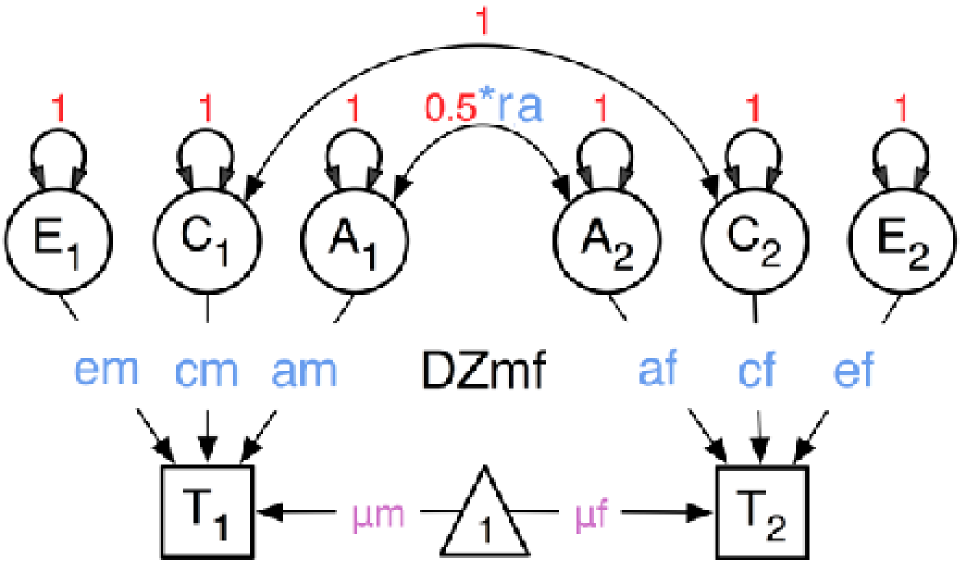
```
# Create Algebra for expected Variance/Covariance Matrices in MZ & DZ twins
covPf       <- mxAlgebra( expression= Af+Cf+Ef, name="Vf" )
covPm       <- mxAlgebra( expression= Am+Cm+Em, name="Vm" )
covMZf      <- mxAlgebra( expression= Af+Cf, name="cMZf" )
covDZf      <- mxAlgebra( expression= 0.5%x%Af+ Cf, name="cDZf" )
covMZm      <- mxAlgebra( expression= Am+Cm, name="cMZm" )
covDZm      <- mxAlgebra( expression= 0.5%x%Am+ Cm, name="cDZm" )
covDZfm     <- mxAlgebra( expression= 0.5%*%rA%x%(af%*%t(am))+cf%*%t(cm), name="cDZfm" )
covDZmf     <- mxAlgebra( expression= 0.5%*%rA%x%(am%*%t(af))+cm%*%t(cf), name="cDZmf" )
expCovMZf <- mxAlgebra( expression= rbind( cbind(Vf, cMZf), cbind(t(cMZf), Vf)), name="expCovMZf" )
expCovDZf <- mxAlgebra( expression= rbind( cbind(Vf, cDZf), cbind(t(cDZf), Vf)), name="expCovDZf" )
expCovMZm <- mxAlgebra( expression= rbind( cbind(Vm, cMZm), cbind(t(cMZm), Vm)), name="expCovMZm" )
expCovDZm <- mxAlgebra( expression= rbind( cbind(Vm, cDZm), cbind(t(cDZm), Vm)), name="expCovDZm" )
expCovDZo <- mxAlgebra( expression= rbind( cbind(Vf, cDZmf), cbind(cDZfm, Vm)), name="expCovDZo" )
```



Covariances



```
# Create Algebra for expected Variance/Covariance Matrices in MZ & DZ twins
covPf     <- mxAlgebra( expression= Af+Cf+Ef, name="Vf" )
covPm     <- mxAlgebra( expression= Am+Cm+Em, name="Vm" )
covMZf    <- mxAlgebra( expression= Af+Cf, name="cMZf" )
covDZf    <- mxAlgebra( expression= 0.5%x%Af+ Cf, name="cDZf" )
covMZm    <- mxAlgebra( expression= Am+Cm, name="cMZm" )
covDZm    <- mxAlgebra( expression= 0.5%x%Am+ Cm, name="cDZm" )
covDZfm   <- mxAlgebra( expression= 0.5%*%rA%x%(af%*%t(am))+cf%*%t(cm), name="cDZfm" )
covDZmf   <- mxAlgebra( expression= 0.5%*%rA%x%(am%*%t(af))+cm%*%t(cf), name="cDZmf" )
expCovMZf <- mxAlgebra( expression= rbind( cbind(Vf, cMZf), cbind(t(cMZf), Vf)), name="expCovMZf" )
expCovDZf <- mxAlgebra( expression= rbind( cbind(Vf, cDZf), cbind(t(cDZf), Vf)), name="expCovDZf" )
expCovMZm <- mxAlgebra( expression= rbind( cbind(Vm, cMZm), cbind(t(cMZm), Vm)), name="expCovMZm" )
expCovDZm <- mxAlgebra( expression= rbind( cbind(Vm, cDZm), cbind(t(cDZm), Vm)), name="expCovDZm" )
expCovDZo <- mxAlgebra( expression= rbind( cbind(Vf, cDZmf), cbind(cDZfm, Vm)), name="expCovDZo" )
```



```
# Create Data Objects for Multiple Groups
dataMZm    <- mxData( observed=mzmData, type="raw" )
dataDZm    <- mxData( observed=dzmData, type="raw" )
dataMZf    <- mxData( observed=mzfData, type="raw" )
dataDZf    <- mxData( observed=dzfData, type="raw" )
dataDZo    <- mxData( observed=dzoData, type="raw" )

# Create Expectation Objects for Multiple Groups
expMZm     <- mxExpectationNormal( covariance="expCovMZm", means="expMeanMZm", dimnames=selVars )
expDZm     <- mxExpectationNormal( covariance="expCovDZm", means="expMeanDZm", dimnames=selVars )
expMZf     <- mxExpectationNormal( covariance="expCovMZf", means="expMeanMZf", dimnames=selVars )
expDZf     <- mxExpectationNormal( covariance="expCovDZf", means="expMeanDZf", dimnames=selVars )
expDZo     <- mxExpectationNormal( covariance="expCovDZo", means="expMeanDZo", dimnames=selVars )
funML     <- mxFitFunctionML()

# Create Model Objects for Multiple Groups
parsm      <- list( meanGm, pathAm, pathCm, pathEm, covAm, covCm, covEm, covPm, pathrA )
parsf      <- list( meanGf, pathAf, pathCf, pathEf, covAf, covCf, covEf, covPf, pathrA )
modelMZm   <- mxModel( parsm, covMZm, expCovMZm, expMeanMZm, dataMZm, expMZm, funML, name="MZm" )
modelDZm   <- mxModel( parsm, covDZm, expCovDZm, expMeanDZm,  dataDZm, expDZm, funML, name="DZm" )
modelMZf   <- mxModel( parsf, covMZf, expCovMZf, expMeanMZf,  dataMZf, expMZf, funML, name="MZf" )
modelDZf   <- mxModel( parsf, covDZf, expCovDZf, expMeanDZf,  dataDZf, expDZf, funML, name="DZf" )
modelDZo   <- mxModel( parsm, parsf, covDZmf, covDZfm, expCovDZo, expMeanDZo,  dataDZo, expDZo, funML, name="DZo" )
multi     <- mxFitFunctionMultigroup( c("MZm","DZm","MZf","DZf","DZo") )
```

```
# Create Algebra for Variance Components
rowVC     <- rep('VC',nv)
colVCm    <- rep(c('Am','Cm','Em','SAm','SCm','SEm'),each=nv)
colVCf    <- rep(c('Af','Cf','Ef','SAf','SCf','SEf'),each=nv)
estVCm    <- mxAlgebra( expression=cbind(Am,Cm,Em,Am/Vm,Cm/Vm,Em/Vm), name="VCm", dimnames=list(rowVC,colVCm))
estVCf    <- mxAlgebra( expression=cbind(Af,Cf,Ef,Af/Vf,Cf/Vf,Ef/Vf), name="VCf", dimnames=list(rowVC,colVCf))

# Create Confidence Interval Objects
ciACE     <- mxCI( c("VCf[1,1:3]","VCm[1,1:3]") )

# Build Model with Confidence Intervals
modelACErq  <- mxModel( "oneACErq5c", parsm, parsf, modelMZm, modelDZm, modelMZf, modelDZf, modelDZo, multi, estVCm, estVCf, ciACE )

# ------------------------------------------------------------------------------------------------------------------------
# RUN MODEL

# Run General Sex Limitation ACE Model
fitACErq    <- mxRun( modelACErq, intervals=T )
sumACErq    <- summary( fitACErq )

# Print Goodness-of-fit Statistics & Parameter Estimates
fitGofs(fitACErq)
fitEstVCfm(fitACErq)
```