Univariate/ MonoPhenotype Modeling

Boulder Workshop 2020

Hermine H. Maes
with credit to Nick Martin, Elizabeth Prom-Wormley, Tim Bates, Katrina Grasby & many others

hmaes/2020/maes/twinModeling

hmaes/2020/maes/twinModeling

- oneSATc.R
- oneACEvc.R
- oneADEvc.R
- oneACEc.R
- oneADEc.R
- miFunctions.R
- miFunctionsDocs.pdf
- twinModeling2020.pdf

Questions

- Does a trait of interest run in families?
- Can familial resemblance be explained by genetic and/or environmental effects?
- Which sources of variance contribute significantly to the variance of the trait?
- How much of the variation in the trait is accounted for by genetic and environmental effects?

Roadmap for Univariate Analysis

- Use data to test basic assumptions (equal means & variances for twin 1/twin 2 and MZ/DZ pairs)
  - Saturated Model
- Estimate contributions of genetic/environmental effects on total variance of a phenotype
  - ACE or ADE Models
- Test ACE / ADE submodels to identify and report significant genetic and environmental contributions
  - AE / CE / E Only Model

Practical Example

- Dataset: NH&MRC Twin Register
- 1981 Questionnaire
- BMI (body mass index): weight/height squared
  - kg/m^2, transformed: 7*log(BMI), simulated based on real data
- Young Female Cohort: 18-30 years
- Sample Size:
  - MZf: 534 pairs (zyg=1; zygosity='MZFF' & cohort='younger')
  - DZf: 328 pairs (zyg=3; zygosity='DZFF' & cohort='younger')

Dataset

```
> head(twinData)
  fam age zyg part wt1 wt2    ht1    ht2   htwt1   htwt2    bmi1    bmi2
1   1  21   1    2  58  57 1.7000 1.7000 20.0692 19.7232 20.9943 20.8726
2   2  24   1    2  54  53 1.6299 1.6299 20.3244 19.9481 21.0828 20.9519
3   3  21   1    2  55  50 1.6499 1.6799 20.2020 17.7154 21.0405 20.1210
4   4  21   1    2  66  76 1.5698 1.6499 26.7759 27.9155 23.0125 23.3043
5   5  19   1    2  50  48 1.6099 1.6299 19.2894 18.0662 20.7169 20.2583
6   6  26   1    2  60  60 1.5999 1.5698 23.4375 24.3418 22.0804 22.3454
....
```

My Naming Conventions

```
name of variable(s)              vars     <- 'bmi'
number of variables              nv       <- 1
number of twin variables         ntv      <- nv*2
variables per twin pair          selVars <-c('bmi1','bmi2')
definition variables             covVars
number of factors                nf       <- 2
number of thresholds             nth      <- 3

starting values                  sv
lower bound / upper bound        lb / ub
labels                           lab

built model                      modelNAME
fitted model                     fitNAME
summary of fitted model          sumNAME
```

Classical Twin Study Background

- The Classical Twin Study (CTS) uses MZ and DZ twins reared together
  - MZ twins share 100% of their genes
  - DZ twins share on average 50% of their genes
- Expectation: Genetic factors are assumed to contribute to a phenotype when MZ twins are more similar than DZ twins

Classical Twin Study Assumptions

- Equal Environments of MZ and DZ pairs
- Random Mating
- No GE Correlation
- No G x E Interaction
- No Sex Limitation
- No G x Age Interaction

Classical Twin Study Basic Data Assumptions

- MZ and DZ twins are sampled from the same population, therefore we expect :
  - Equal means/variances in Twin 1 and Twin 2
  - Equal means/variances in MZ and DZ twins
- Further assumptions would need to be tested if we introduce male twins and opposite sex twin pairs

Observed Values

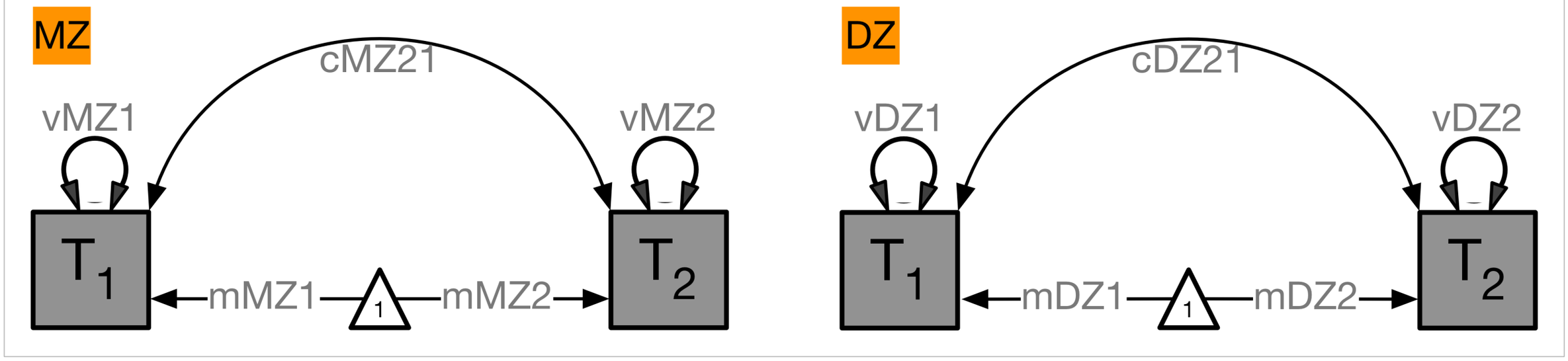
	Descriptive Statistics					
	MZ twins			DZ twins		
		T1	**T2**		**T1**	**T2**
mean	**MZ**	21.34	21.35	**DZ**	21.45	21.46
		T1	**T2**		**T1**	**T2**
cov	**T1**	0.73		**T1**	0.77	
	T2	0.59	0.79	**T2**	0.24	0.82

‘Old Fashioned’ Data Checking

Nice, but how can we actually be sure that these means and variances are truly the same?

Saturated Model

SAT model
oneSATc.R



Intuition behind Maximum Likelihood (ML)

- Likelihood: probability that an observation (data point) is predicted by specified model
- Maximum Likelihood Estimates (MLE): most likely values of population parameter values (e.g, μ, σ, β) given observed sample values
  - Define model
  - Define probability of observing a given event conditional on a particular set of parameters
  - Choose a set of parameters which are most likely to have produced observed results

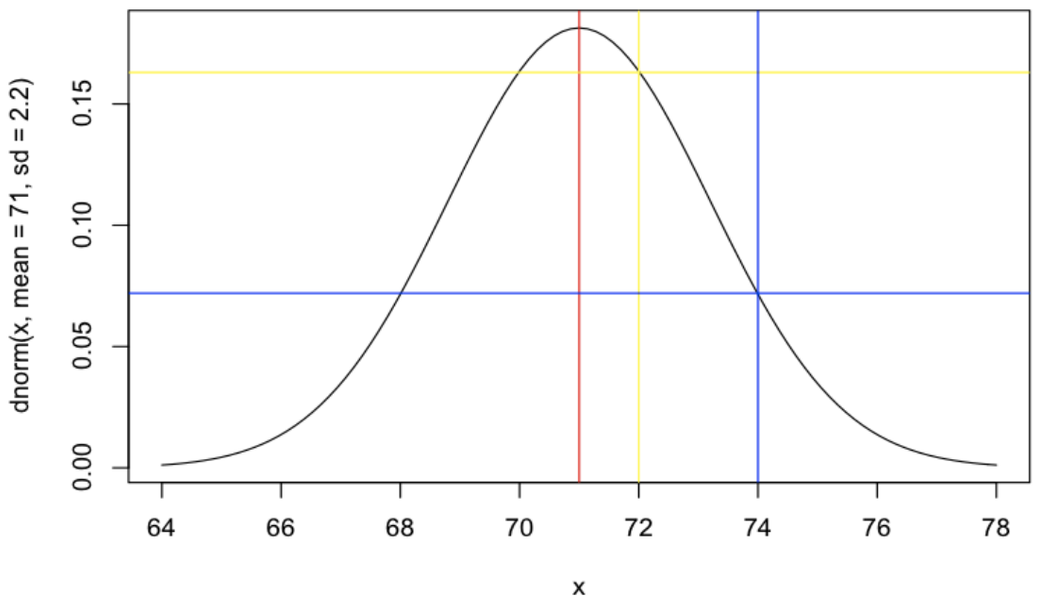
Likelihood Ratio Test

- Likelihood Ratio (LR) test: simple comparison of Log-Likelihoods under 2 separate models:
  - Model Mu: Unconstrained (has more parameters)
  - Model Mc: Constrained (has fewer parameters)
- LR statistic equals:
  - LR (Mc | Mu) = 2ln(L(Mu) - 2ln(L(Mc)
- LR is asymptotically distributed as χ^2 with df equal to number of constraints

Probability Density Function Φ(xi)

- Φ(xi): likelihood of data point x_i for particular mean and variance estimates
- Univariate: height of probability density function

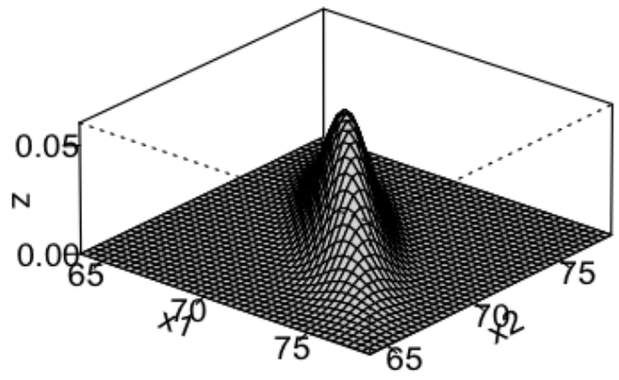
$$\Phi\left(x_i\right) = -\left|2\pi\sigma^2\right|^{-.5}\ e^{-5((x_i-\mu)^2/\sigma^2)}$$



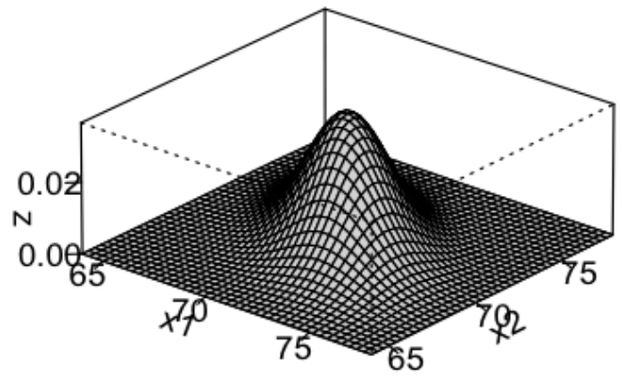
π: pi=3.14; x_i: observed value of variable i; μ: expected mean; σ: expected variance

Multinormal Probability Function

- $\Phi(x_i)$: likelihood of pair of data points x_i and y_i for particular means, variances & correlation estimates
- Multivariate: height of multinormal probability density function



rMZ=.85



rDZ=.49

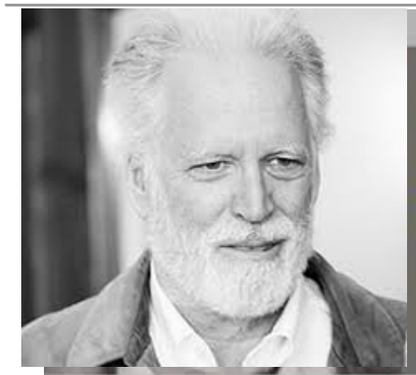
$$\Phi\left(x_i\right) = -|2\pi\Sigma|^{-n/2}\ e^{-5((x_i-\mu)\Sigma^{-1}(x_i-\mu)')}$$

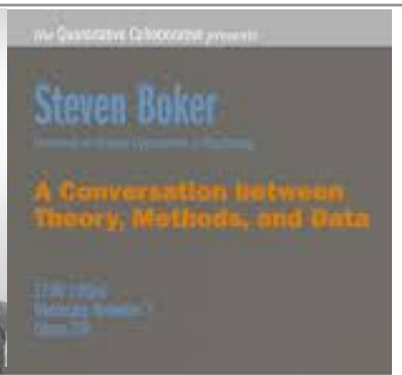
π= 3.14; x_i: value of variable i; μ: expected mean; ∑: expected covariance matrix

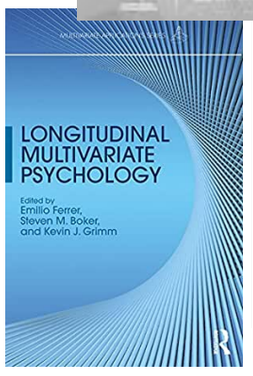
OpenMx Scripts

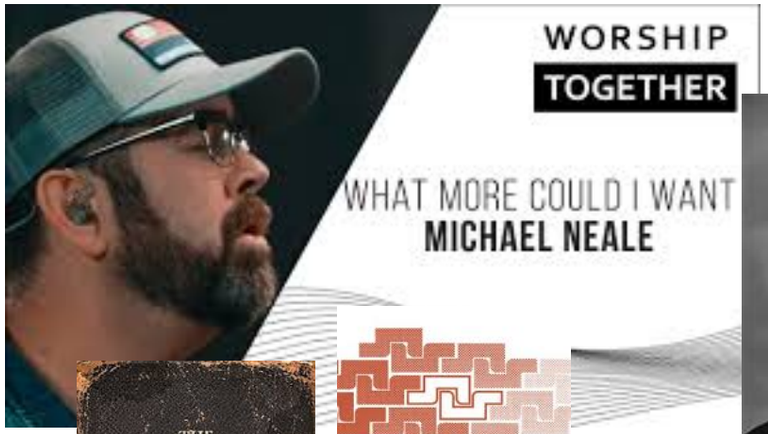
- oneSATc.R
  - Saturated model estimating means & variances for continuous data in MZ & DZ twins
- oneACEvc.R
  - Univariate/Monophenotype model estimating A, C & E components for continuous data in MZ & DZ twins
- oneADEvc.R
  - Univariate/Monophenotype model estimating A, D & E components for continuous data in MZ & DZ twins

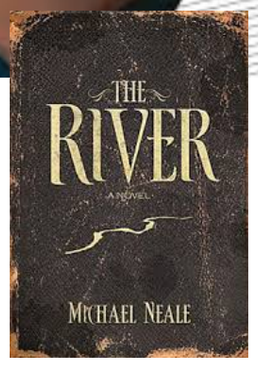
OpenMx Development Team (most active ones)

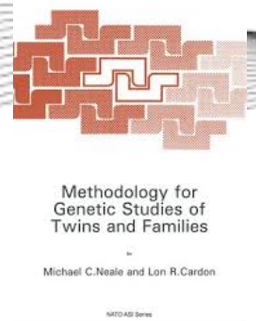


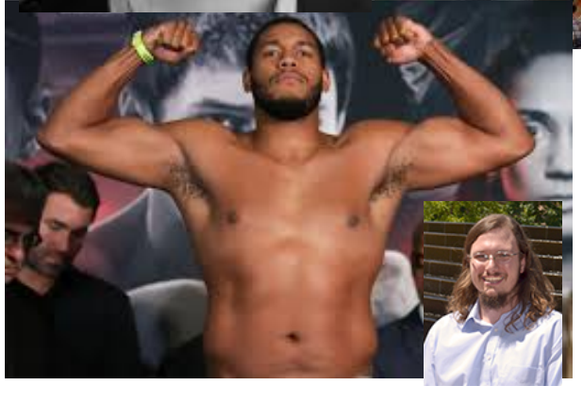


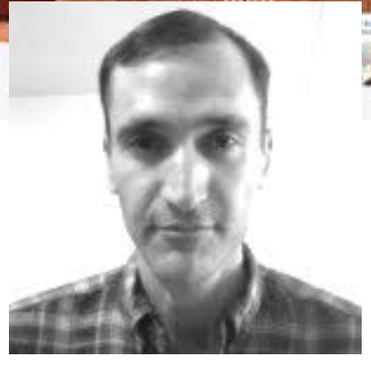












OpenMx Development Team (most active ones)

Steve Boker

Mike Neale

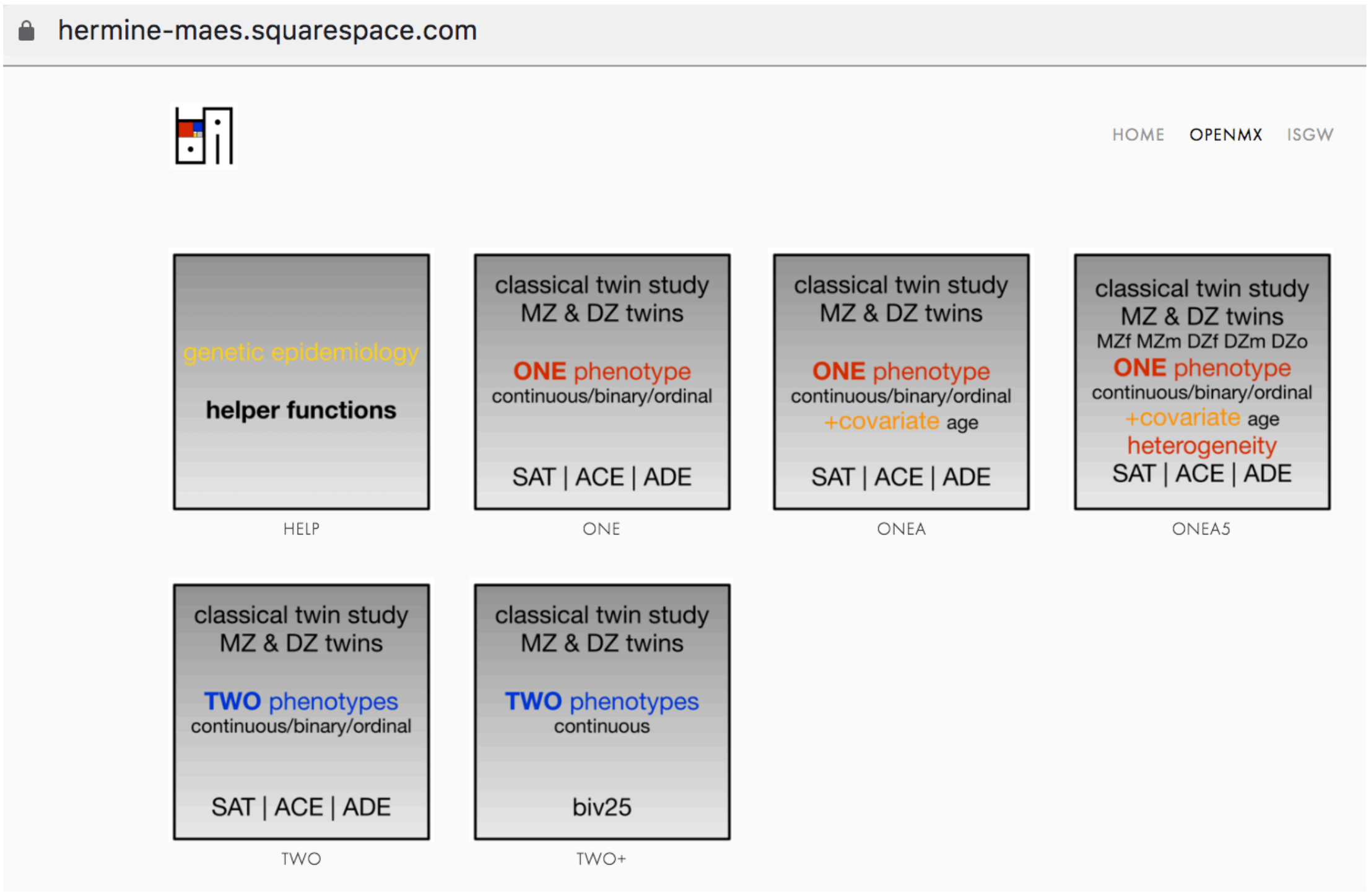
Mike Hunter

Rob Kirkpatrick

Joshua Pritikin

Tim Bates

https://hermine-maes.squarespace.com



helper functions

HOME OPENMX ISGW

Parallel standard OpenMx scripts & matching umx versions

This site offers OpenMx and matching umx scripts to fit standard biometrical models to data collected in MZ and DZ twins. The models can be used to estimate the role of genetic (A: additive genetic factors; D: dominance genetic factors) and environmental factors (C: common/shared environmental factors, E: unique environmental factors) to the variance of phenotypes of interest and covariances between phenotypes of interest.

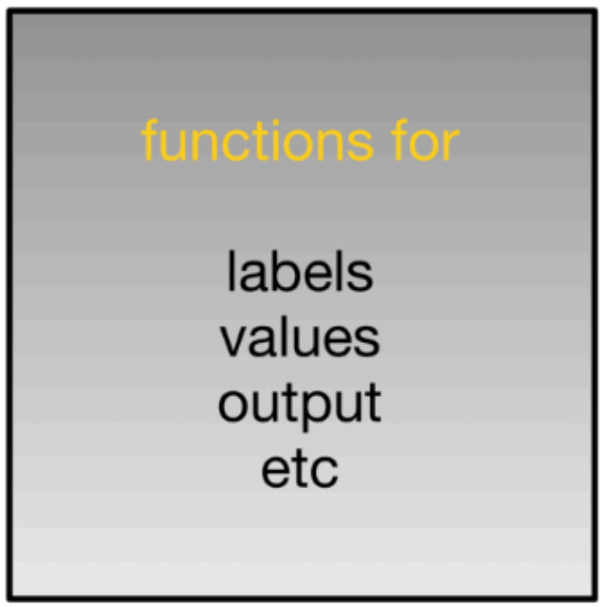
Scripts are organized in pages by number of phenotypes, addition of covariates etc. Within each page, scripts are organized by type of model in rows (Saturated, ACE estimating variance components, ADE estimating variance components, ACE estimating path coefficients, ADE estimating path coefficients) and by type of data in columns (continuous, binary, ordinal (estimating all thresholds), ordinal (fixing two thresholds and estimating means/variances) using standard code, and using umx (for different data types) in the last column. All scripts source the R code attached here that includes a number of functions that automate various aspects of the models such as labels, starting values, output generated etc.

Note that each of the scripts is represented by a path diagram. If you click on the diagram, a PDF of the associated script will be displayed in a separate window. If you click on the filename.R below the diagram, the R script will be downloaded.

Comments, suggestions, corrections welcome!! **Email hmaes@vcu.edu.**

New pages/scripts will be added as they are ready to go! If you have scripts that you'd like to add, send them my way!

Last updated: 02/28/2020



miFunctions.R

miFunctions.R

```
# ---------------------------------------------------------------------------
# Program: miFunctions.R
#  Author: Hermine Maes
#    Date: 02 28 2020
#
# Set of my options & functions used in basic twin methodology scripts
#   Email: hmaes@vcu.edu
# -------|---------|---------|---------|---------|---------|---------|---------|

# Options
mxOption( NULL, "Default optimizer", "NPSOL" )
#mxOption( NULL, "Checkpoint Prefix", filename )
mxOption( NULL, "Checkpoint Units", "iterations" )
mxOption( NULL, "Checkpoint Count", 1 )
options(width=120)
options(digits=8)
mxVersion()


# Functions to assign labels
# ---------------------------------------------------------------------------
labLower  <- function(lab,nv) { paste(lab,rev(nv+1-sequence(1:nv)),rep(1:nv,nv:1),sep="") }
labSdiag  <- function(lab,nv) { paste(lab,rev(nv+1-sequence(1:(nv-1))),rep(1:(nv-1),(nv-1):1),sep="") }
labOdiag  <- function(lab,nv) { paste(lab,c(rev(nv+1-sequence(1:(nv-1))),rep(1:(nv-1),(nv-1):1)),c(rep(1:(nv-1),
 (nv-1):1),rev(nv+1-sequence(1:(nv-1)))),sep="") }
```

….

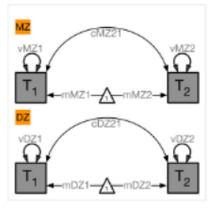
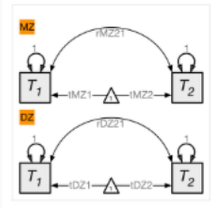
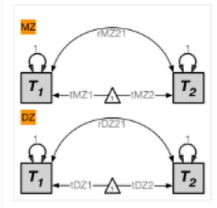
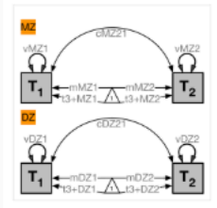
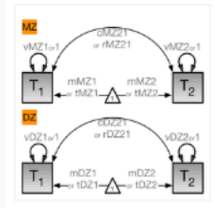
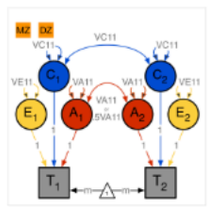
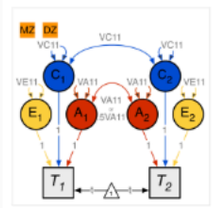
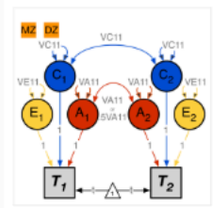
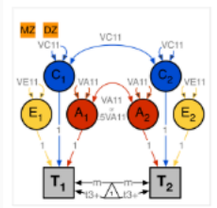
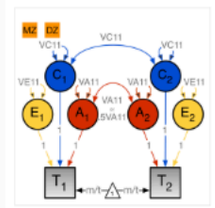
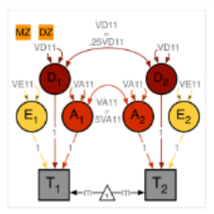
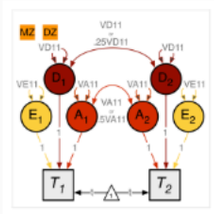
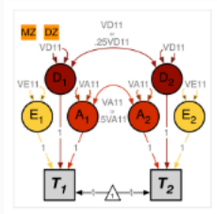
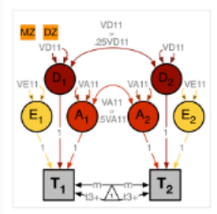
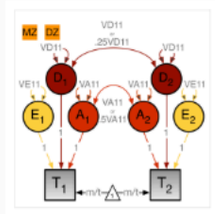
miFunctionsDocs.pdf

one

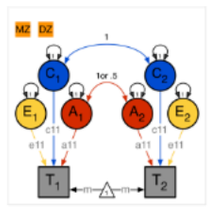
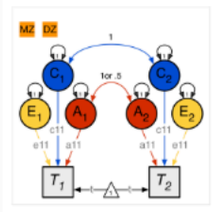
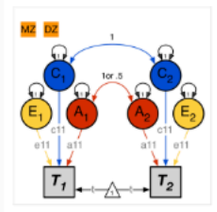
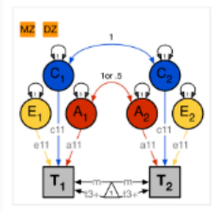
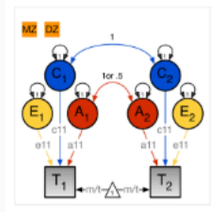
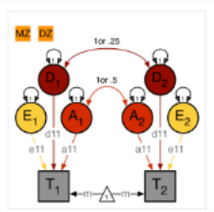
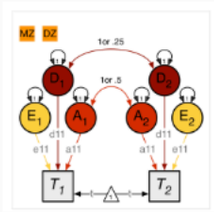
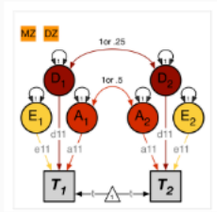
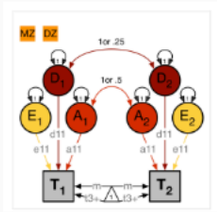
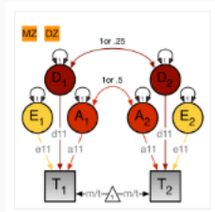
classical twin study
MZ & DZ twins

ONE phenotype
continuous/binary/ordinal

SAT | ACE | ADE

.	One Phenotype **CONTINUOUS** c	One Phenotype **BINARY** b	One Phenotype **ORDINAL** o	One Phenotype **ORDINAL** m *	One Phenotype **UMX** c/b/m
SAT estimating means/thresholds, variances & covariances * > 2 categories					
	oneSATc.R	**oneSATb.R**	**oneSATo.R**	**oneSATm.R**	**oneSATu.R**
ACE estimating variance components					
	oneACEvc.R	**oneACEvb.R**	**oneACEvo.R**	**oneACEvm.R**	**oneACEvu.R**
ADE estimating variance components					
	oneADEvc.R	**oneADEvb.R**	**oneADEvo.R**	**oneADEvm.R**	**oneADEvu.R**

NOTE! Models below estimating path coefficients may provide biased estimates of the parameters. Use of these scripts is discouraged.

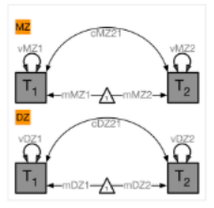
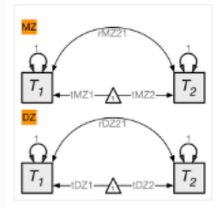
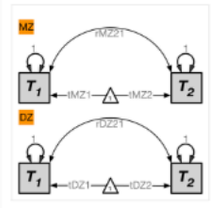
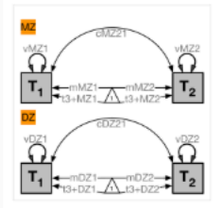
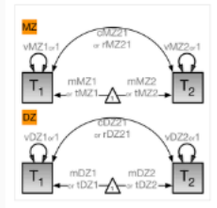
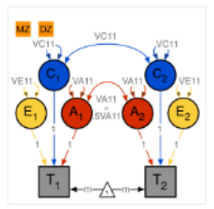
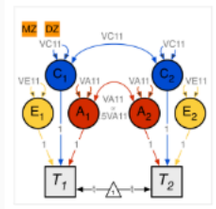
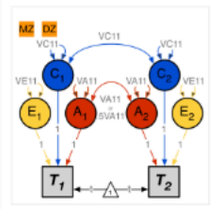
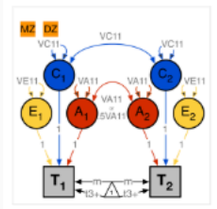
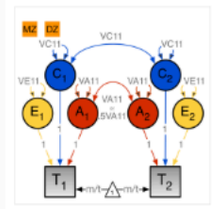
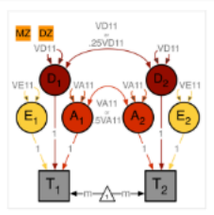
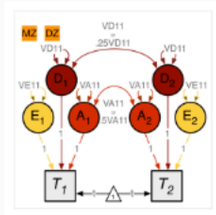
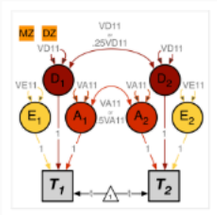
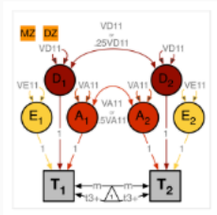
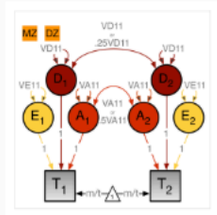
ACE estimating path coefficients					
	oneACEc.R	**oneACEb.R**	**oneACEo.R**	**oneACEm.R**	**oneACEu.R**
ADE estimating path coefficients					
	oneADEc.R	**oneADEb.R**	**oneADEo.R**	**oneADEm.R**	**oneADEu.R**

one

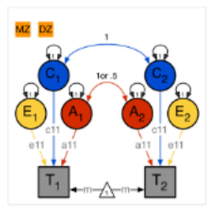
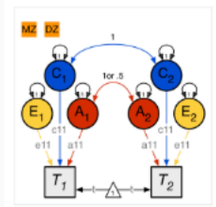
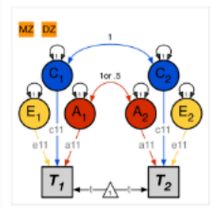
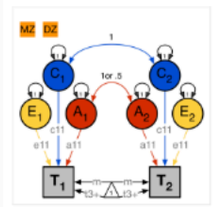
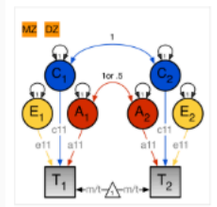
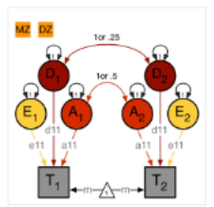
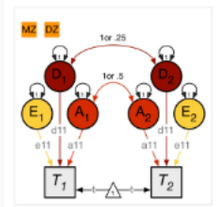
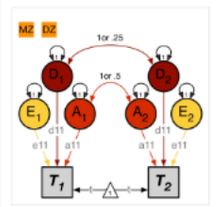
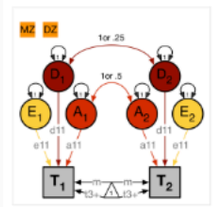
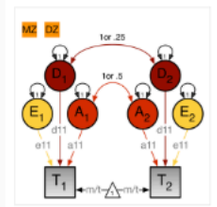
classical twin study
MZ & DZ twins

ONE phenotype
continuous/binary/ordinal

SAT | ACE | ADE

.	One Phenotype **CONTINUOUS** c	One Phenotype **BINARY** b	One Phenotype **ORDINAL** o	One Phenotype **ORDINAL** m *	One Phenotype **UMX** c/b/m
SAT estimating means/thresholds, variances & covariances * > 2 categories	 **oneSATc.R**	 **oneSATb.R**	 **oneSATo.R**	 **oneSATm.R**	 **oneSATu.R**
ACE estimating variance components	 **oneACEvc.R**	 **oneACEvb.R**	 **oneACEvo.R**	 **oneACEvm.R**	 **oneACEvu.R**
ADE estimating variance components	 **oneADEvc.R**	 **oneADEvb.R**	 **oneADEvo.R**	 **oneADEvm.R**	 **oneADEvu.R**

NOTE! Models below estimating path coefficients may provide biased estimates of the parameters. Use of these scripts is discouraged.

ACE estimating path coefficients	 **oneACEc.R**	 **oneACEb.R**	 **oneACEo.R**	 **oneACEm.R**	 **oneACEu.R**
ADE estimating path coefficients	 **oneADEc.R**	 **oneADEb.R**	 **oneADEo.R**	 **oneADEm.R**	 **oneADEu.R**

Univariate Saturated Model oneSATc.R

```
# ------------------------------------------------------------------------------
# Program: oneSATc.R
#  Author: Hermine Maes
#    Date: 10 22 2018
#
# Twin Univariate Saturated model to estimate means and (co)variances across multiple groups
# Matrix style model - Raw data - Continuous data
# -------|---------|---------|---------|---------|---------|---------|---------|

# Load Libraries & Options
rm(list=ls())
library(OpenMx)
library(psych); library(polycor)
source("miFunctions.R")

# Create Output
filename    <- "oneSATc"
sink(paste(filename,".Ro",sep=""), append=FALSE, split=TRUE)
```

load OpenMx

my functions which you can edit as you like

creates output file with extension .Ro

Preparing Data

oneSATc.R

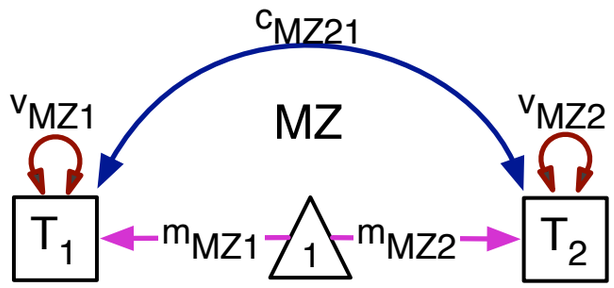
```
# ------------------------------------------------------------------------------
# PREPARE DATA

# Load Data
data(twinData)
dim(twinData)
describe(twinData[,1:12], skew=F)

# Select Variables for Analysis
vars      <- 'bmi'                    # 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
mzData    <- subset(twinData, zyg==1, selVars)  # zygosity='MZFF' & cohort='younger'
dzData    <- subset(twinData, zyg==3, selVars)

# Generate Descriptive Statistics
colMeans(mzData,na.rm=TRUE)
colMeans(dzData,na.rm=TRUE)
cov(mzData,use="complete")
cov(dzData,use="complete")

# Set Starting Values
svMe      <- 20                       # start value for means
svVa      <- .8                       # start value for variance
lbVa      <- .0001                    # lower bound for variance
```

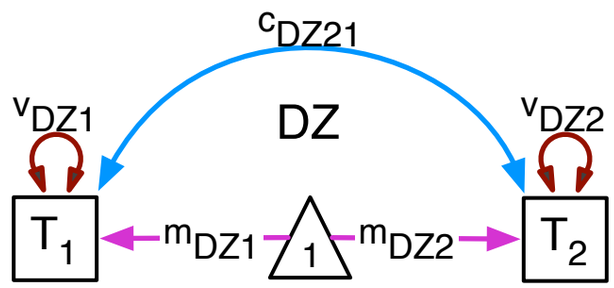
load 'twinData' or read in your own

analyzing c('bmi1','bmi2')

get right codes for zygosity

SAT Deconstructed: *Covariance Matrices & Means*






```
meanMZ     <- mxMatrix( type="Full", nrow=1, ncol=ntv,
 free=TRUE, values=svMe, labels=c("mMZ1","mMZ2"),name="meanMZ" )
meanDZ     <- mxMatrix( type="Full", nrow=1, ncol=ntv,
 free=TRUE, values=svMe, labels=c("mDZ1","mDZ2"),name="meanDZ" )
```

mMZ1	mMZ2

meanMZ *1x2*

mDZ1	mDZ2

meanDZ *1x2*

```
covMZ      <- mxMatrix( type="Symm", nrow=ntv, ncol=ntv,
 free=TRUE, values=svVas, lbound=lbVas,
 labels=c("vMZ1","cMZ21","vMZ2"), name="covMZ" )
covDZ      <- mxMatrix( type="Symm", nrow=ntv, ncol=ntv,
 free=TRUE, values=svVas, lbound=lbVas,
 labels=c("vDZ1","cDZ21","vDZ2"), name="covDZ" )
```

vMZ1	cMZ21
cMZ21	vMZ2

covMZ *2x2*

vDZ1	cDZ21
cDZ21	vDZ2

covDZ *2x2*

Preparing Model

oneSATc.R

```
# ------------------------------------------------------------------------------
# PREPARE MODEL


# Create Algebra for expected Mean Matrices
meanMZ     <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe, labels=c("mMZ1","mMZ2"), name="meanMZ" )
meanDZ     <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe, labels=c("mDZ1","mDZ2"), name="meanDZ" )

# Create Algebra for expected Variance/Covariance Matrices
covMZ      <- mxMatrix( type="Symm", nrow=ntv, ncol=ntv, free=TRUE, values=valDiag(svVa,ntv), lbound=valDiag(lbVa,ntv),
 labels=c("vMZ1","cMZ21","vMZ2"), name="covMZ" )
covDZ      <- mxMatrix( type="Symm", nrow=ntv, ncol=ntv, free=TRUE, values=valDiag(svVa,ntv), lbound=valDiag(lbVa,ntv),
 labels=c("vDZ1","cDZ21","vDZ2"), name="covDZ" )

# Create Data Objects for Multiple Groups
dataMZ     <- mxData( observed=mzData, type="raw" )
dataDZ     <- mxData( observed=dzData, type="raw" )

# Create Expectation Objects for Multiple Groups
expMZ      <- mxExpectationNormal( covariance="covMZ", means="meanMZ", dimnames=selVars )
expDZ      <- mxExpectationNormal( covariance="covDZ", means="meanDZ", dimnames=selVars )
funML      <- mxFitFunctionML()
```

full matrix for means

symmetric matrix for covariances

fitting to raw data

link to data

using FIML: full information maximum likelihood

Run Model

oneSATc.R

```
# Create Model Objects for Multiple Groups
modelMZ    <- mxModel( meanMZ, covMZ, dataMZ, expMZ, funML, name="MZ" )
modelDZ    <- mxModel( meanDZ, covDZ, dataDZ, expDZ, funML, name="DZ" )
multi      <- mxFitFunctionMultigroup( c("MZ","DZ") )

# Create Confidence Interval Objects
ciCov      <- mxCI( c('MZ.covMZ','DZ.covDZ') )
ciMean     <- mxCI( c('MZ.meanMZ','DZ.meanDZ') )

# Build Saturated Model with Confidence Intervals
modelSAT     <- mxModel( "oneSATc", modelMZ, modelDZ, multi, ciCov, ciMean )
```

model object contains all matrices etc.

evaluating 2 groups simultaneously

built model

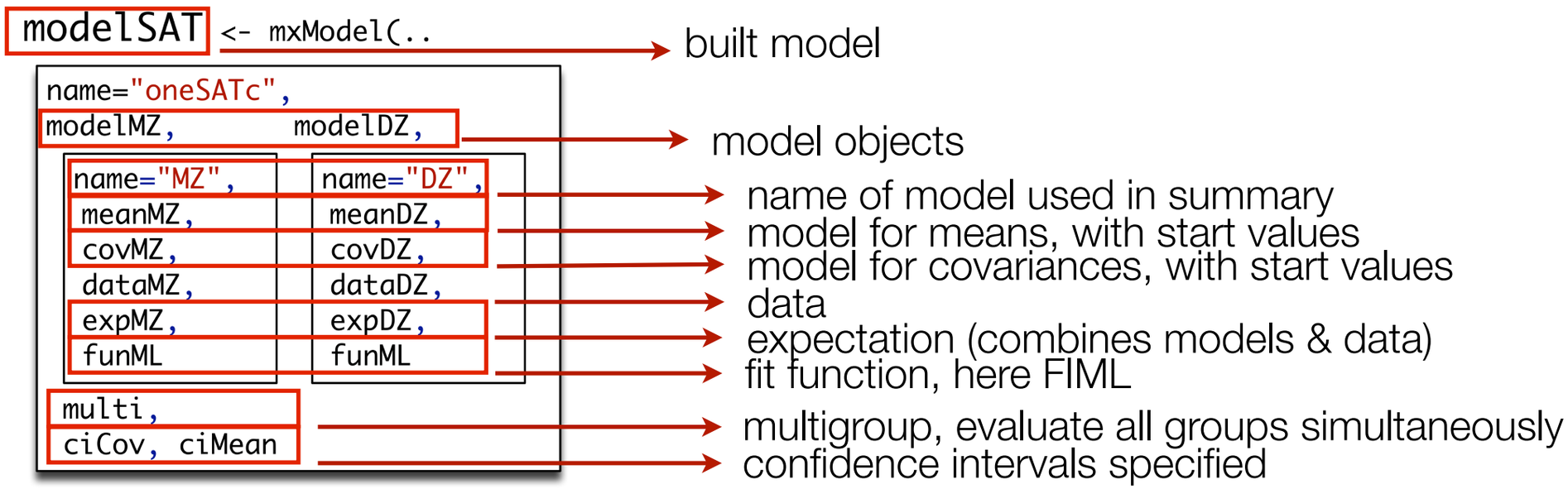
```
# ----------------------------------------------------------------------------
# RUN MODEL

# Run Saturated Model
fitSAT     <- mxRun( modelSAT, intervals=F )
sumSAT     <- summary( fitSAT )

# Print Goodness-of-fit Statistics & Parameter Estimates
fitGofs(fitSAT)
fitEsts(fitSAT)
mxGetExpected( fitSAT, c("means","covariance") )
```

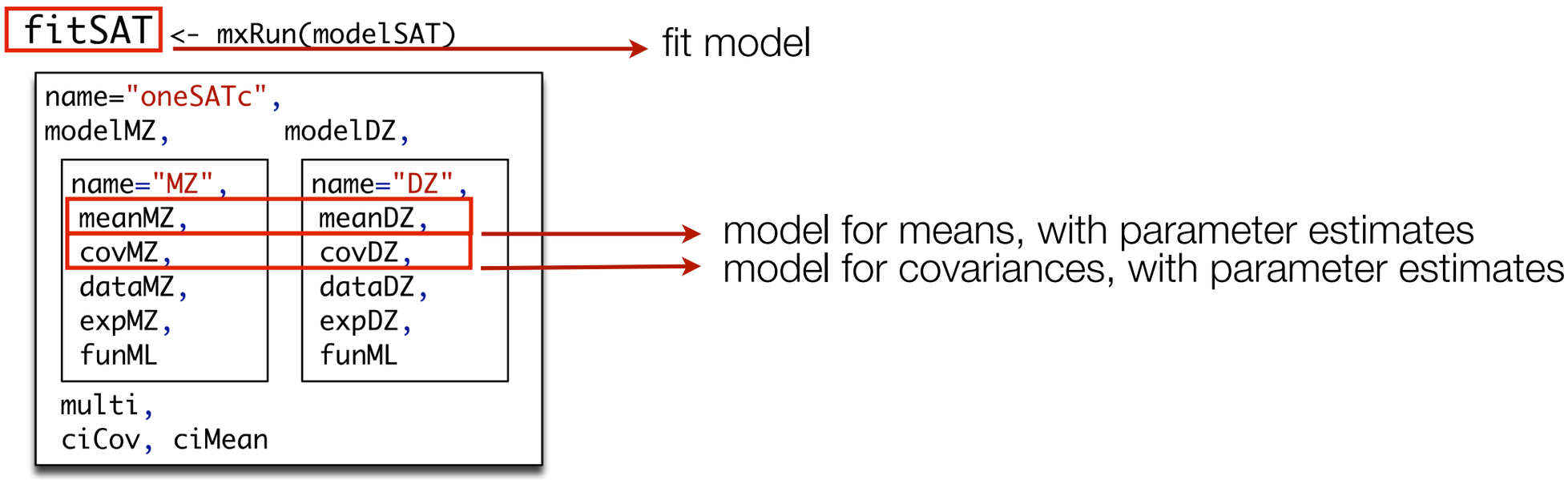
fitted model

standard summary function in OpenMx

my short summary function in miFunctions.R

Model Building - Model Fitting





more of miFunctions.R

```
# Functions to generate output

fitGofs   <- function(fit) {
          summ <- summary(fit)
          cat(paste("Mx:", fit$name,"  os=", summ$ob,"  ns=", summ$nu,"   ep=", summ$es,
                    "   co=", sum(summ$cons),"  df=", summ$de, "  ll=", round(summ$Mi,4),
                    "  cpu=", round(summ$cpu,4),"  opt=", summ$op,"  ver=", summ$mx,
                    "  stc=", fit$output$status$code, "\n",sep=""))
}


fitEsts   <- function(fit) {
  print(round(fit$output$estimate,4))
}

fitEstCis   <- function(fit) {
  print(round(fit$output$estimate,4))
  print(round(fit$output$confidenceIntervals,4))
}
```

Print Goodness-of-Fit Statistics

```
> summary(fitSAT)
Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:             10                  1767             4055.9346
   Saturated:             NA                    NA                   NA
Independence:             NA                    NA                   NA
Number of observations/statistics: 920/1777

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:      521.93461              4075.9346                     NA
BIC:    -8002.73367              4124.1783               4092.4195
CFI: NA
TLI: 1   (also known as NNFI)
RMSEA:  0  [95% CI (NA, NA)]
Prob(RMSEA <= 0.05): NA
To get additional fit indices, see help(mxRefModels)
timestamp: 2020-03-01 16:55:41
Wall clock time: 0.095155001 secs
optimizer:  NPSOL
OpenMx version number: 2.17.2
Need help?  See help(mxSummary)

> fitGofs(fitSAT)
Mx:oneSATc  os=1777  ns=920   ep=10   co=0  df=1767  ll=4055.9346  cpu=0.0952  opt=NPSOL  ver=2.17.2  stc=0
```

Print Estimates

```
> summary(fitSAT)$parameters
free parameters:
   name    matrix  row  col    Estimate   Std.Error A lbound ubound
1   mMZ1 MZ.meanMZ    1 bmi1 21.34437690 0.036061832
2   mMZ2 MZ.meanMZ    1 bmi2 21.34901242 0.037650856
3   vMZ1  MZ.covMZ bmi1 bmi1  0.72766891 0.043658984    1e-04
4  cMZ21  MZ.covMZ bmi1 bmi2  0.59163768 0.040794161        0
5   vMZ2  MZ.covMZ bmi2 bmi2  0.79319915 0.047647906    1e-04
6   mDZ1 DZ.meanDZ    1 bmi1 21.44752035 0.047571928
7   mDZ2 DZ.meanDZ    1 bmi2 21.45784215 0.049233334
8   vDZ1  DZ.covDZ bmi1 bmi1  0.76919130 0.059007266    1e-04
9  cDZ21  DZ.covDZ bmi1 bmi2  0.24004049 0.045201541        0
10  vDZ2  DZ.covDZ bmi2 bmi2  0.82163163 0.063154677    1e-04

> fitEsts(fitSAT)
   mMZ1    mMZ2    vMZ1   cMZ21    vMZ2    mDZ1    mDZ2    vDZ1   cDZ21    vDZ2
21.3444 21.3490  0.7277  0.5916  0.7932 21.4475 21.4578  0.7692  0.2400  0.8216
```

Estimated Values

	Saturated Model					
	MZ twins			DZ twins		
		T1	**T2**		**T1**	**T2**
mean	**MZ**	21.34	21.35	**DZ**	21.45	21.46
		T1	**T2**		**T1**	**T2**
cov	**T1**	0.73		**T1**	0.77	
	T2	0.59	0.79	**T2**	0.24	0.82

10 parameters estimated:

`mMZ1, mMZ2, vMZ1, vMZ2, cMZ21`

`mDZ1, mDZ2, vDZ1, vDZ2, cDZ21`

Goodness-of-Fit Statistics

	os	ep	-2ll	df	AIC	diff -2ll	diff df	p
Saturated	1777	10	4055.93	1767	521.93			

os	observed statistics	
ep	estimated parameters	
-2ll	-2 LogLikelihood	
df	degrees of freedom	os - ep
AIC	Akaike's Information Criterion	-2ll -2df

Fitting Nested Models

oneSATc.R

```
# Constrain expected Means to be equal across twin order
modelEMO  <- mxModel(fit, name="oneEMOc" )
modelEMO  <- omxSetParameters( modelEMO, label=c("mMZ1","mMZ2"), free=TRUE, values=svMe, newlabels='mMZ' )
modelEMO  <- omxSetParameters( modelEMO, label=c("mDZ1","mDZ2"), free=TRUE, values=svMe, newlabels='mDZ' )
fitEMO    <- mxRun( modelEMO, intervals=F )
fitGofs(fitEMO); fitEsts(fitEMO)

# Constrain expected Means and Variances to be equal across twin order
modelEMVO <- mxModel(fitEMO, name="oneEMVOc" )
modelEMVO <- omxSetParameters( modelEMVO, label=c("vMZ1","vMZ2"), free=TRUE, values=svVa, newlabels='vMZ' )
modelEMVO <- omxSetParameters( modelEMVO, label=c("vDZ1","vDZ2"), free=TRUE, values=svVa, newlabels='vDZ' )
fitEMVO   <- mxRun( modelEMVO, intervals=F )
fitGofs(fitEMVO); fitEsts(fitEMVO)

# Constrain expected Means and Variances to be equal across twin order and zygosity
modelEMVZ <- mxModel(fitEMVO, name="oneEMVZc" )
modelEMVZ <- omxSetParameters( modelEMVZ, label=c("mMZ","mDZ"), free=TRUE, values=svMe, newlabels='mZ' )
modelEMVZ <- omxSetParameters( modelEMVZ, label=c("vMZ","vDZ"), free=TRUE, values=svVa, newlabels='vZ' )
fitEMVZ   <- mxRun( modelEMVZ, intervals=F )
fitGofs(fitEMVZ); fitEsts(fitEMVZ)

# Print Comparative Fit Statistics
mxCompare( fitSAT, subs <- list(fitEMO, fitEMVO, fitEMVZ) )

# ------------------------------------------------------------------------------
sink()
save.image(paste(filename,".Ri",sep=""))
```

changing parameters

existing parameters

new parameters

generate likelihood ratio test

close .Ro file & save image as file with .Ri extension

Goodness-of-Fit Stats

	os	ep	-2ll	df	AIC	diff -2ll	diff df	p
Saturated	1777	10	4055.93	1767	521.93			
mT1=mT2	1777	8	4056.00	1769	518.00	0.07	2	0.97
mT1=mT2 varT1=varT2	1777	6	4058.94	1771	516.94	3.01	4	0.56
Zyg MZ=DZ	1777	4	4063.45	1773	517.45	7.52	6	0.28

diff -2ll	likelihood ratio Chi-square	
diff df	difference in degrees of freedom	
p	probability of Chi-square	

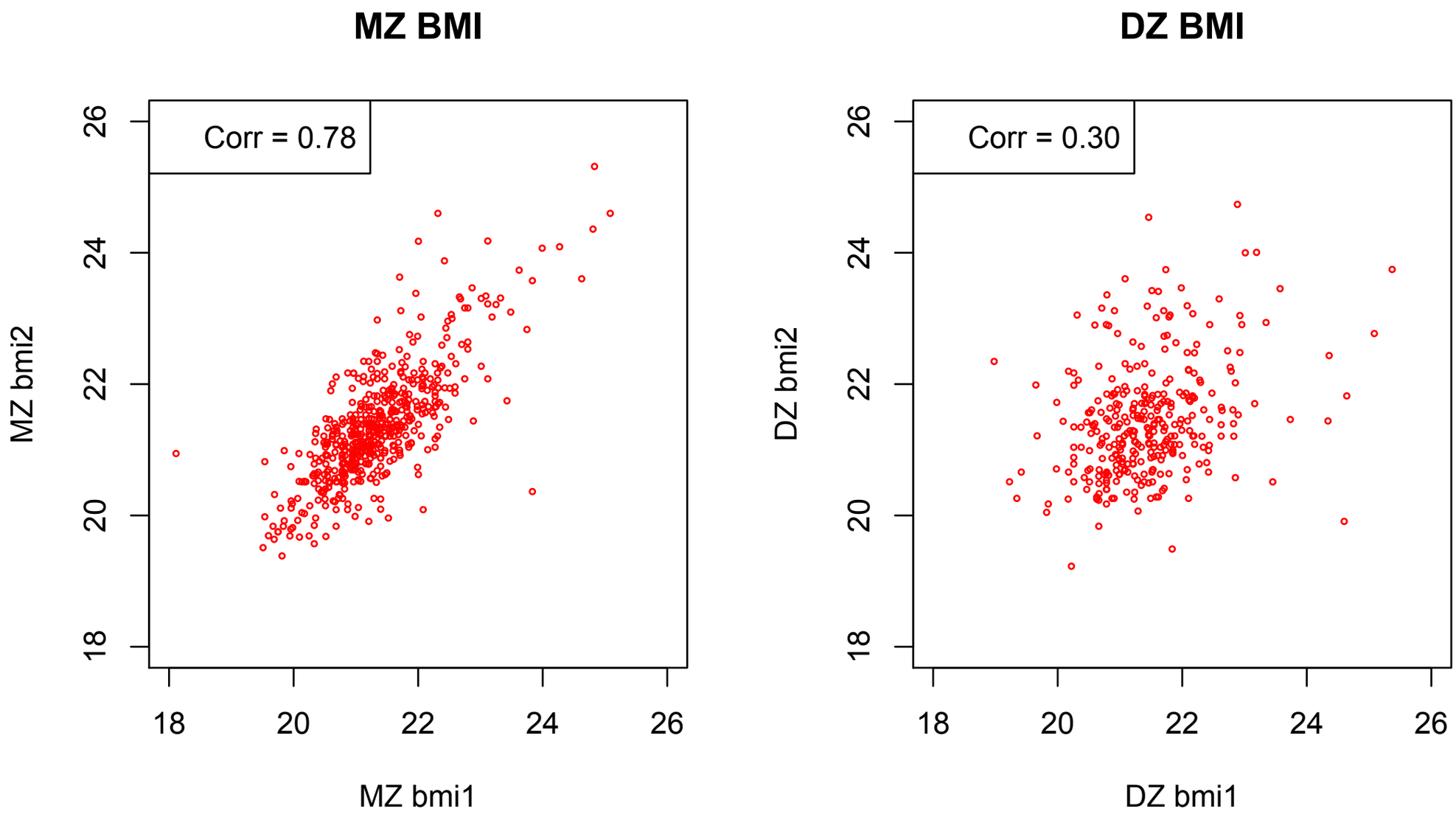
Conclusions so far

- BMI in young OZ females (age 18-30)
  - means of twin 1 and twin 2 not significantly different from one another in MZ & DZ pairs
  - variances of twin 1 and twin 2 not significantly different from one another in MZ & DZ pairs
  - means and variances of MZs and DZs not significantly different from one another
  - basic data assumptions about CTS met

Twin Correlations ~ Sources of Variance

1-rMZ	**E** +
rMZ > rDZ	**E** + **A**
rMZ = 2*rDZ	**E** + only **A**
rMZ = rDZ	**E** + only **C**
rDZ > 1/2 rMZ	**E** + **A** & **C**
rDZ < 1/2 rMZ	**E** + **A** & **D**

Example Twin Correlations



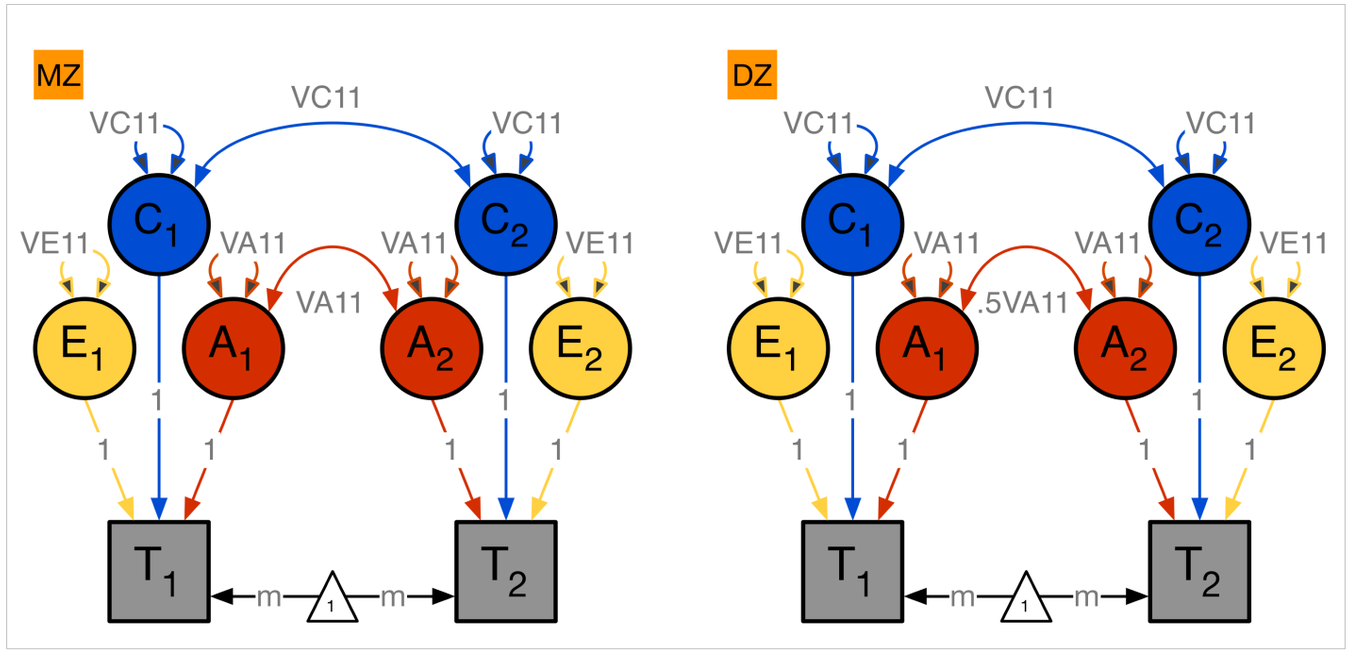
Roadmap for Univariate Analysis

- Use data to test basic assumptions (equal means & variances for twin 1/twin 2 and MZ/DZ pairs)
  - Saturated Model
- Estimate contributions of genetic/environmental effects on total variance of a phenotype
  - ACE or ADE Models
- Test ACE / ADE submodels to identify and report significant genetic and environmental contributions
  - AE / CE / E Only Models

Univariate ACE / ADE Model **variance estimation**

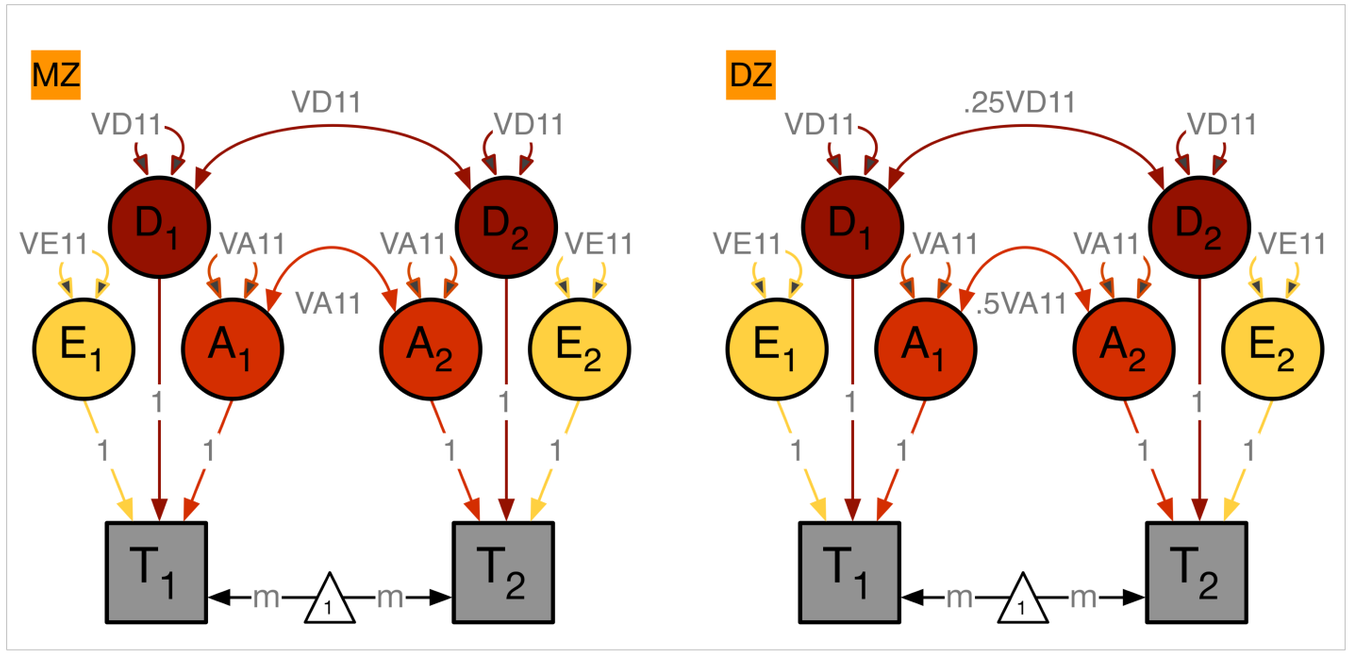
ACE model

oneACEvc.R

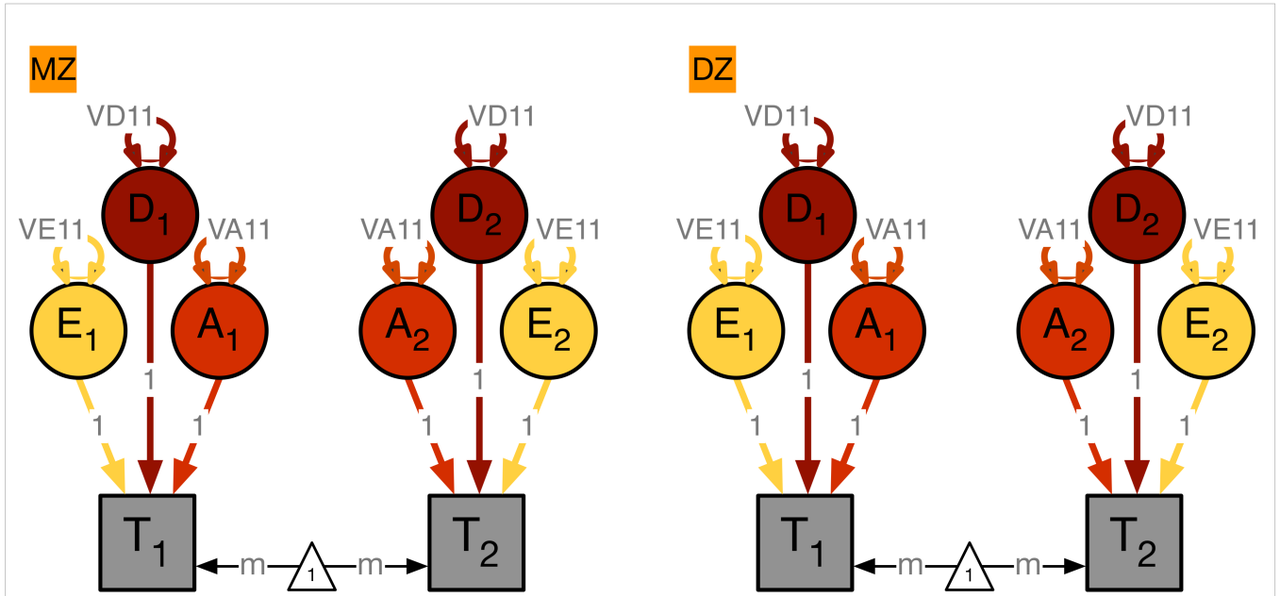


ADE model

oneADEvc.R



ADE Deconstructed: *Variance Components*




```
covA      <- mxMatrix( type="Symm", nrow=nv, ncol=nv,
free=TRUE, values=svPa, label="VA11", name="VA" )

covD      <- mxMatrix( type="Symm", nrow=nv, ncol=nv,
free=TRUE, values=svPa, label="VD11", name="VD" )

covE      <- mxMatrix( type="Symm", nrow=nv, ncol=nv,
free=TRUE, values=svPe, label="VE11", name="VE" )
```

VA$_{11}$

VA *1x1*

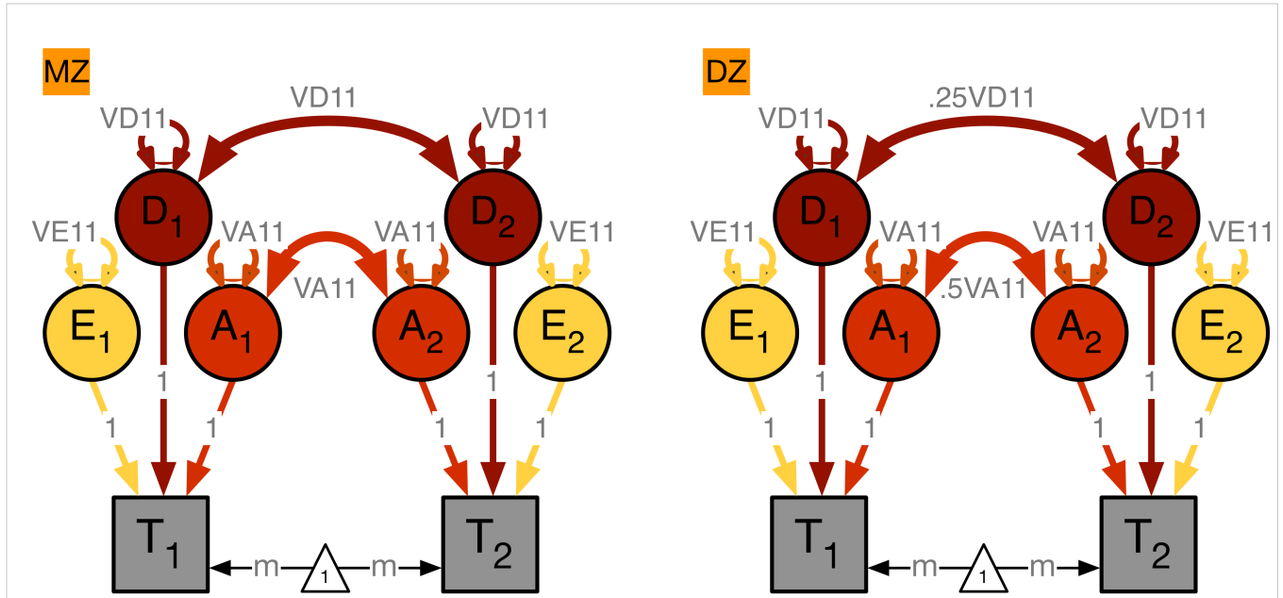
VD$_{11}$

VD *1x1*

VE$_{11}$

VE *1x1*

ADE Deconstructed: *Variances + Covariances*




```
covP      <- mxAlgebra( expression= VA+VD+VE,
 name="V" )

covMZ     <- mxAlgebra( expression= VA+VD,
 name="cMZ" )

covDZ     <- mxAlgebra( expression= 0.5%x%VA+ 0.25%x%VD,
 name="cDZ" )
```

V	VA+VD+VE

V *1x1*

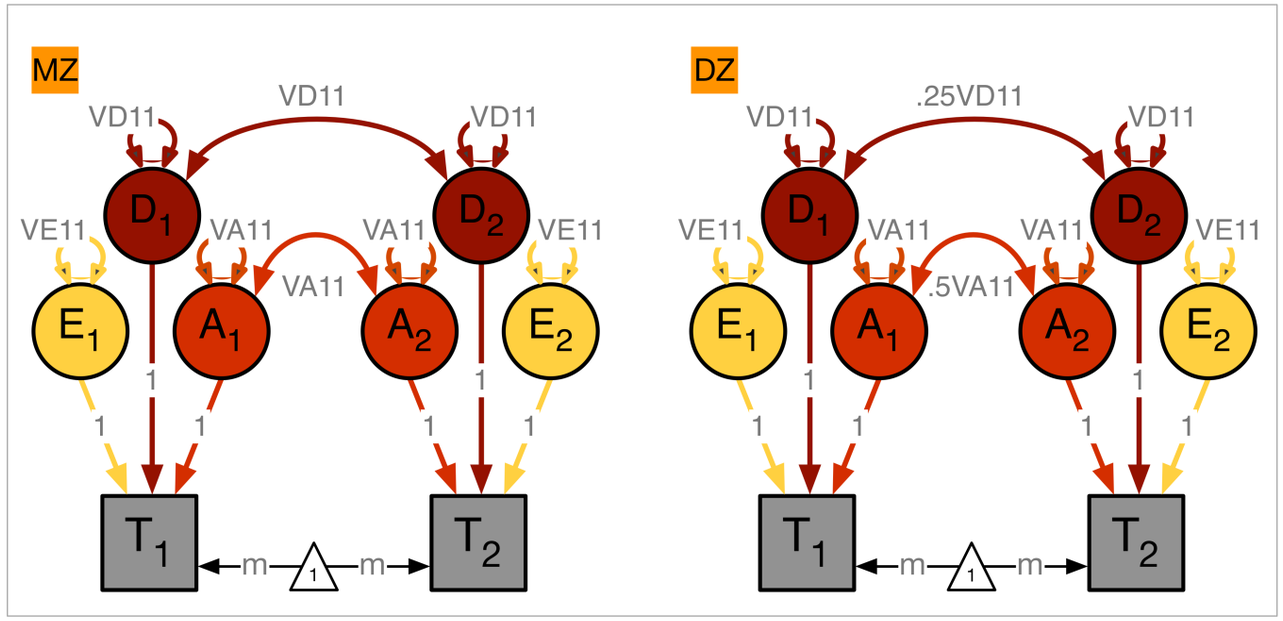
cMZ	VA+VD

cMZ *1x1*

cDZ	.5VA+.25VD

cDZ *1x1*

ADE Deconstructed: *Covariance Matrices & Means*




```
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" )
```

V	cMZ
cMZ	V

expCovMZ *2x2*

V	cDZ
cDZ	V

expCovDZ *2x2*

```
meanG     <- mxMatrix( type="Full", nrow=1, ncol=ntv,
 free=TRUE, values=svMe, labels=labVars("mean",vars),
 name="meanG" )
```

x_1	x_1

meanG *1x2*

Model Specification

oneADEvc.R

```
# ------------------------------------------------------------------------------
# PREPARE MODEL

# Create Algebra for expected Mean Matrices
meanG     <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe, labels=labVars("mean",vars),
 name="meanG" )

# Create Matrices for Variance Components
covA      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa, label="VA11", name="VA" )
covD      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa, label="VD11", name="VD" )
covE      <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPe, label="VE11", name="VE" )

# Create Algebra for expected Variance/Covariance Matrices in MZ & DZ twins
covP      <- mxAlgebra( expression= VA+VD+VE, name="V" )
covMZ     <- mxAlgebra( expression= VA+VD, name="cMZ" )
covDZ     <- mxAlgebra( expression= 0.5%x%VA+ 0.25%x%VD, 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" )
```

variance components: VA, VD & VE

Model Specification 2

oneADEvc.R

```
# Create Data Objects for Multiple Groups
dataMZ    <- mxData( observed=mzData, type="raw" )
dataDZ    <- mxData( observed=dzData, type="raw" )

# Create Expectation Objects for Multiple Groups
expMZ     <- mxExpectationNormal( covariance="expCovMZ", means="meanG", dimnames=selVars )
expDZ     <- mxExpectationNormal( covariance="expCovDZ", means="meanG", dimnames=selVars )
funML     <- mxFitFunctionML()

# Create Model Objects for Multiple Groups
pars      <- list(meanG, covA, 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") )

# Create Algebra for Variance Components
rowUS     <- rep('US',nv)
colUS     <- rep(c('VA','VD','VE','SA','SD','SE'),each=nv)
estUS     <- mxAlgebra( expression=cbind(VA,VD,VE,VA/V,VD/V,VE/V), name="US", dimnames=list(rowUS,colUS))

# Create Confidence Interval Objects
ciADE     <- mxCI( "US[1,1:3]" )

# Build Model with Confidence Intervals
modelADE  <- mxModel( "oneADEvc", pars, modelMZ, modelDZ, multi, estUS, ciADE )
```

list of common elements

calculate standardized variance components

list of matrix elements to calculate confidence intervals (CI)

ADE model object

Run Model

oneADEvc.R

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

# Run ADE Model
fitADE    <- mxRun( modelADE, intervals=T )
sumADE    <- summary( fitADE )

# Compare with Saturated Model
#if saturated model fitted in same session
mxCompare( fit, fitADE )
#if saturated model prior to genetic model
#lrtSAT( fitADE,4055.9346,1767)
```

estimate CI's

function in miFunctions.R to provide -2ll & df of previously fit model

```
# Print Goodness-of-fit Statistics & Parameter Estimates
fitGofs(fitADE)
fitEstCis(fitADE)

round(fitADE$US$result,4)
```

print estimates of variance components

summary(fitADE)

```
free parameters:
     name matrix row col    Estimate   Std.Error A
1 meanbmi  meanG   1   1 21.39464927 0.025973494
2    VA11     VA   1   1  0.32092995 0.150909584
3    VD11     VD   1   1  0.28942518 0.147886812
4    VE11     VE   1   1  0.16935016 0.010363413

confidence intervals:
                       lbound   estimate     ubound note
oneADEvc.US[1,1] 0.016290870 0.32092995 0.61208265
oneADEvc.US[1,2] 0.011924028 0.28942518 0.59556124
oneADEvc.US[1,3] 0.150553156 0.16935016 0.19139089

Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:              4                  1773             4063.4496
   Saturated:             NA                    NA                   NA
Independence:             NA                    NA                   NA
Number of observations/statistics: 920/1777

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:       517.44962             4071.4496                     NA
BIC:    -8036.16490             4090.7471              4078.0436
```

miFunctions: fitGofs & fitEsts

```
> fitGofs(fitADE)
Mx:oneADEvc  os=1777  ns=920   ep=4   co=0  df=1773  ll=4063.4496  cpu=0.1513  opt=NPSOL  ver=2.17.2  stc=0
>
> fitEstCis(fitADE)
meanbmi    VA11    VD11    VE11
21.3946  0.3209  0.2894  0.1694
                lbound estimate ubound
oneADEvc.US[1,1] 0.0163   0.3209 0.6121
oneADEvc.US[1,2] 0.0119   0.2894 0.5956
oneADEvc.US[1,3] 0.1506   0.1694 0.1914
```

```
> round(fitADE$US$result,4)

       VA     VD     VE     SA     SD     SE
US 0.3209 0.2894 0.1694 0.4116 0.3712 0.2172
```

Goodness-of-Fit Stats & Estimates

	os	ep	-2ll	df	AIC	diff -2ll	diff df	p
Saturated	1777	10	4055.93	1767	521.93			
ADE	1777	4	4063.45	1773	517.45	7.51	6	0.27

	unstandardized variance components			standardized variance components		
	VA	VD	VE	SA	SD	SE
ADE	0.32	0.29	0.17	0.41	0.37	0.22

Roadmap for Univariate Analysis

- Use data to test basic assumptions (equal means & variances for twin 1/twin 2 and MZ/DZ pairs)
  - Saturated Model
- Estimate contributions of genetic/environmental effects on total variance of a phenotype
  - ACE or ADE Models
- Test ACE / ADE submodels to identify and report significant genetic and environmental contributions
  - AE / CE / E Only Models

Fitting Nested Models

oneADEvc.R

```
# ---------------------------------------------------------------------------
# RUN SUBMODELS

# Run AE model
modelAE   <- mxModel( fitADE, name="oneAEc" )
modelAE   <- omxSetParameters( modelAE, labels="VD11", free=FALSE, values=0 )
fitAE     <- mxRun( modelAE, intervals=T )
fitGofs(fitAE); fitEsts(fitAE)

# Run E model
modelE    <- mxModel( fitAE, name="oneEc" )
modelE    <- omxSetParameters( modelE, labels="VA11", free=FALSE, values=0 )
fitE      <- mxRun( modelE, intervals=T )
fitGofs(fitE); fitEsts(fitE)

# Print Comparative Fit Statistics
mxCompare( fitADE, nested <- list(fitAE, fitE) )
round(rbind(fitADE$US$result,fitAE$US$result,fitE$US$resulT ),4)
```

dropping parameters

Nested Models

- ‘Full’ ADE Model
- Nested Models
  - AE Model vs ADE Model: test significance of **D**
  - E Model vs AE Model: test significance of **A**
  - E Model vs ADE Model: test significance of both **A & D**

Goodness-of-Fit Statistics **variance estimation**

	os	ep	-2ll	df	AIC	diff -2ll	diff df	p
Saturated	1777	10	4055.93	1767	521.93			
ADE	1777	4	4063.45	1773	517.45	7.51	6	0.27
AE	1777	3	4067.66	1774	519.66	4.21	1	0.04
E	1777	2	4591.79	1775	1041.79	528.34	2	0.00

Under the null hypothesis, test is distributed as a chi-square with 1df

Estimated Values **variance estimation**

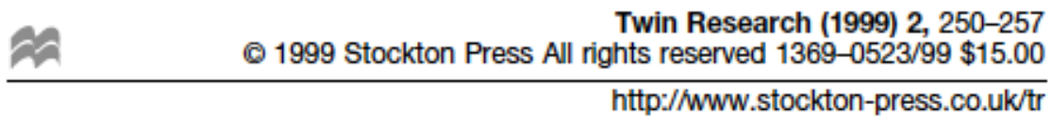
	unstandardized variance components			standardized variance components		
	VA	VD	VE	SA	SD	SE
ADE	0.32 0.02-0.61	0.29 0.01-0.60	0.17 0.15-0.19	0.41	0.37	0.22
AE	0.62 0.56-0.68	-	0.17 0.17-0.19	0.78	-	0.22
E	-	-	0.78 0.73-0.83	-	-	1.00

Conclusions

- BMI in young OZ females (age 18-30)
  - **additive** genetic factors: highly significant
  - **dominance**: borderline significant
  - **specific environment**al factors: significant
  - **shared environment**al factors: not

Publications

- Eaves LJ: Inferring the causes of human variation. J. R. Stat. Soc. Ser. A 140, 324–355,1977.
- Neale MC, Cardon LR: Methodology for Genetic Studies of Twins and Families (NATO ASI Series), Dordrecht, The Netherlands: Kluwer Academic Publishers, 496p, 1992.
- Posthuma P, Beem AL, de Geus EJC, van Baal GCM, von Hjelmborg JB, Iachine I, Boomsma DI: Theory and Practice in Quantitative Genetics. Twin Research 6:361-376, 2003.
- Eaves LJ, Chen S, Neale M, Maes HH, Silberg J: Questions, Models and Methods in Psychiatric Genetics, in Psychiatric Genetics (Review of Psychiatry Vol 24), Kendler KS & Eaves LJ (Eds). Washington, DC: American Psychiatric Publishing, Inc., 2005.
- Maes HH: The ACE model, in Encyclopedia for Behavioral Statistics (Wiley Series in Probability and Statistics), Purcell S (Volume Editor). John Wiley & Sons, Inc., 2005.
- Neale MC: Biometrical Models in Behavioral Genetics, in Handbook of Behavior Genetics, Yong-Kyu, K. (Volume Editor). Springer, 2009.
- Evans DM, Frazer IH, Martin NG: Genetic and environmental causes of variation in basal levels of blood cells, Twin Research 2: 250-257, 1999.



Twin Research (1999) 2, 250–257

http://www.stockton-press.co.uk/tr

Genetic and environmental causes of variation in basal levels of blood cells

David M Evans[1], Ian H Frazer[2] and Nicholas G Martin[1]

Calculate Correlations

- add calculations to openMx scripts
  - as part of script (calculated with every iteration)

```
# Create Algebra for Maximum Likelihood Estimates of Twin Correlations

corMZ      <- mxAlgebra( cov2cor(covMZ), name="corMZ" )
corDZ      <- mxAlgebra( cov2cor(covDZ), name="corDZ" )
```

→ function to calculate correlations from covariances

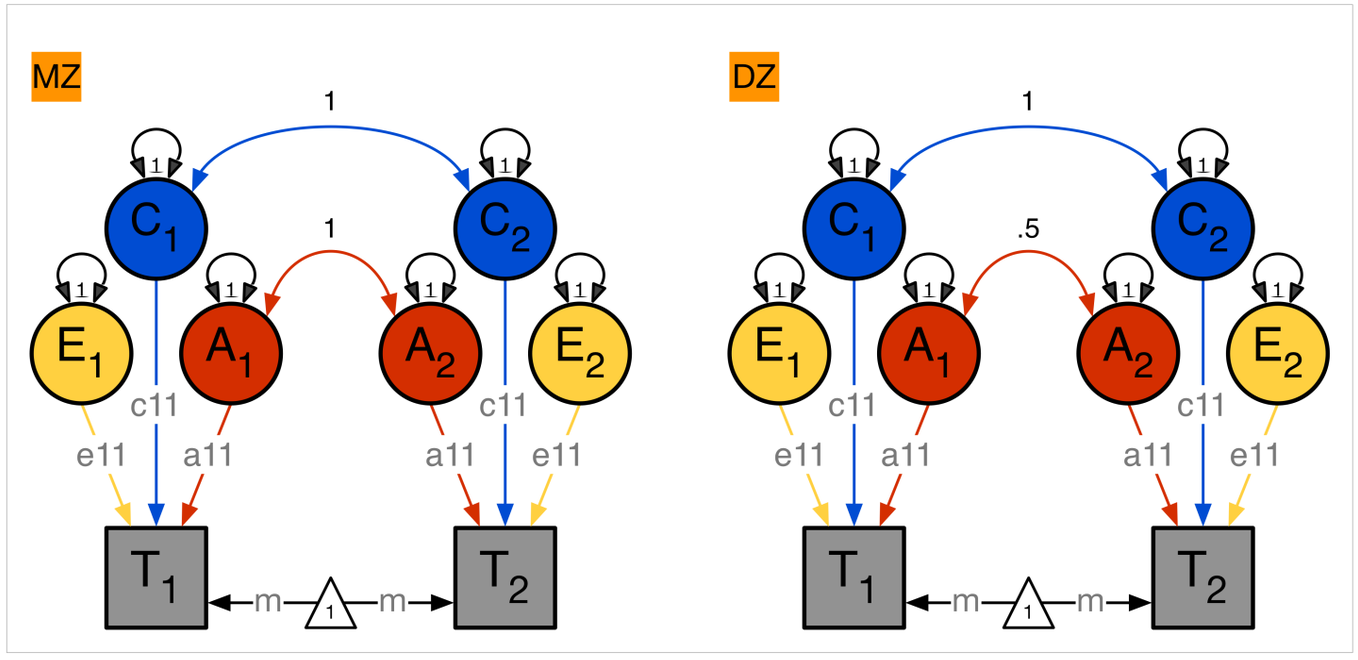
  - after script has been run

```
# Create Algebra for Maximum Likelihood Estimates of Twin Correlations

corMZ      <- mxEval( cov2cor(covMZ), fitSAT$MZ )
corDZ      <- mxEval( cov2cor(covDZ), fitSAT$DZ )
```

Univariate ACE / ADE Model - **path estimation**

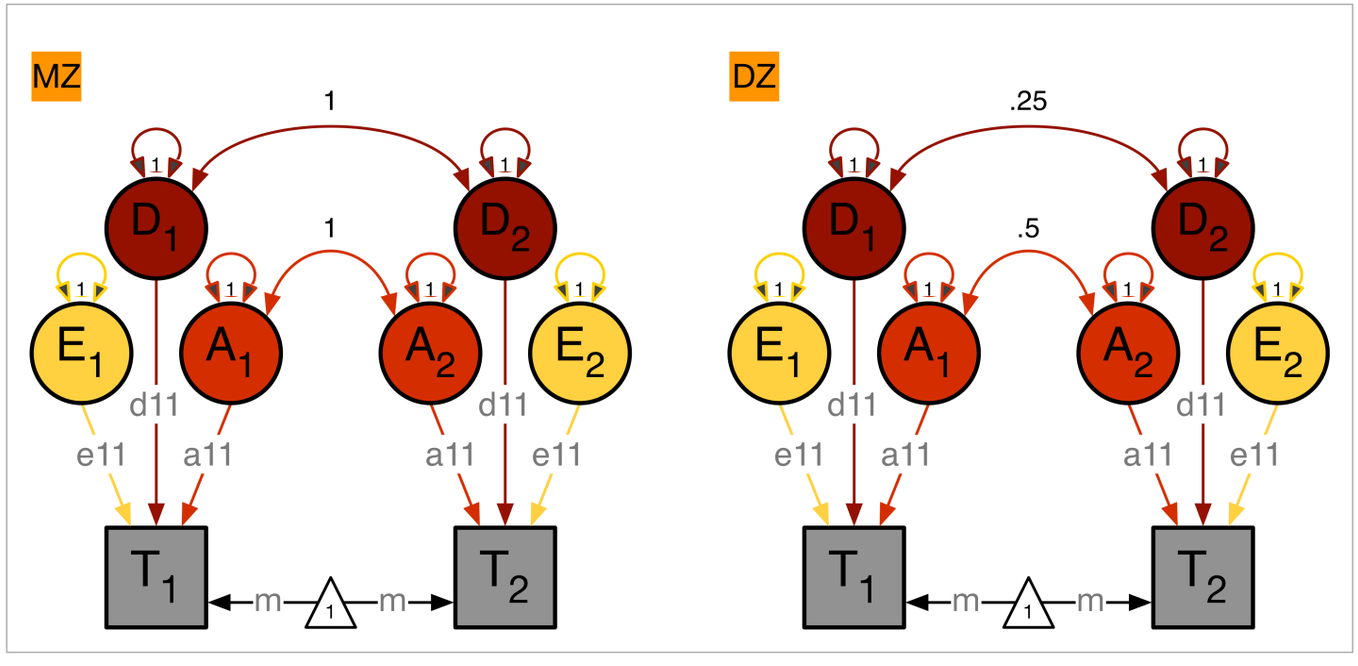
ACE model

oneACEc.R

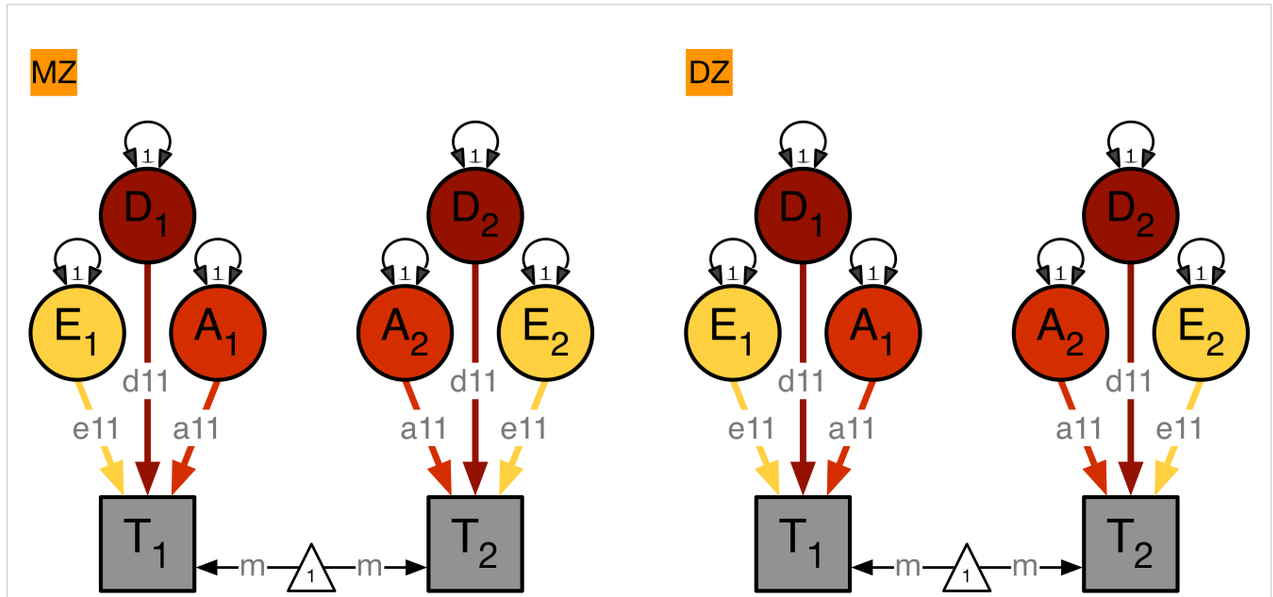


ADE model

oneADEc.R



ADE Deconstructed: *Path Coefficients*




```
pathA     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE,
values=svPa, label="a11", lbound=lbPa, name="a" )
```

a_{11}

a *1x1*

```
pathD     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE,
values=svPa, label="d11", lbound=lbPa, name="d" )
```

d_{11}

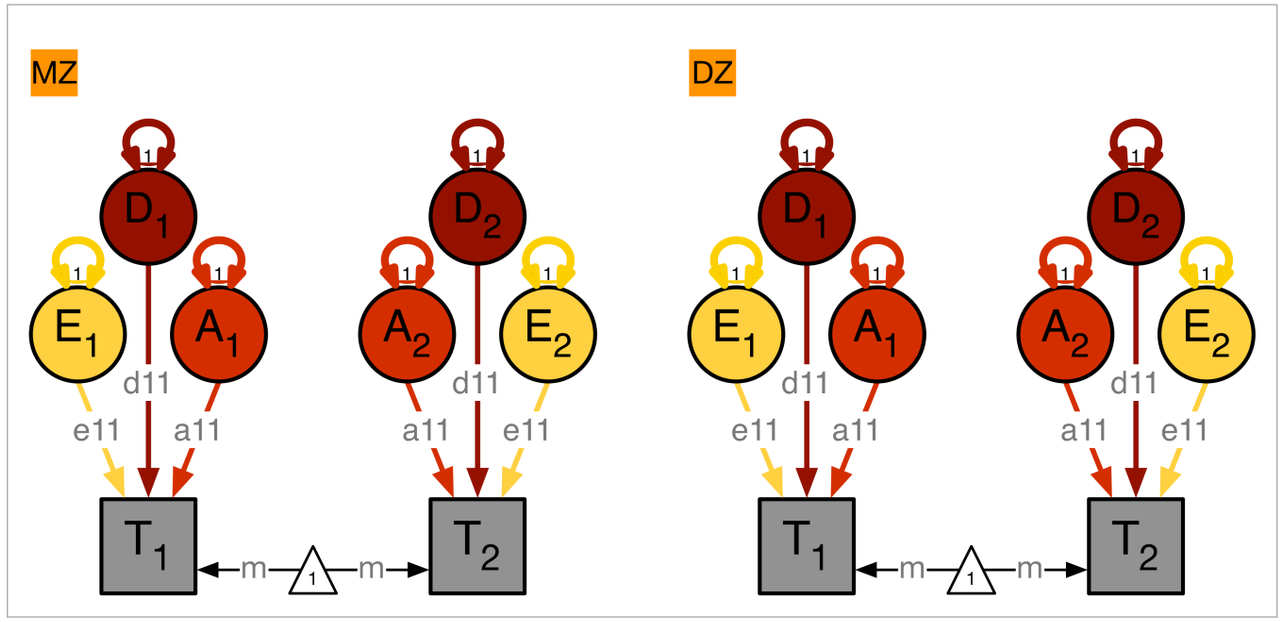
d *1x1*

```
pathE     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE,
values=svPe, label="e11", lbound=lbPa, name="e" )
```

e_{11}

e *1x1*

ADE Deconstructed: *Variance Components*




```
covA      <- mxAlgebra( expression=a %*% t(a),
 name="A" )

covC      <- mxAlgebra( expression=d %*% t(d),
 name="D" )

covE      <- mxAlgebra( expression=e %*% t(e),
 name="E" )
```

$a_{11} * t(a_{11})$

A *1x1*

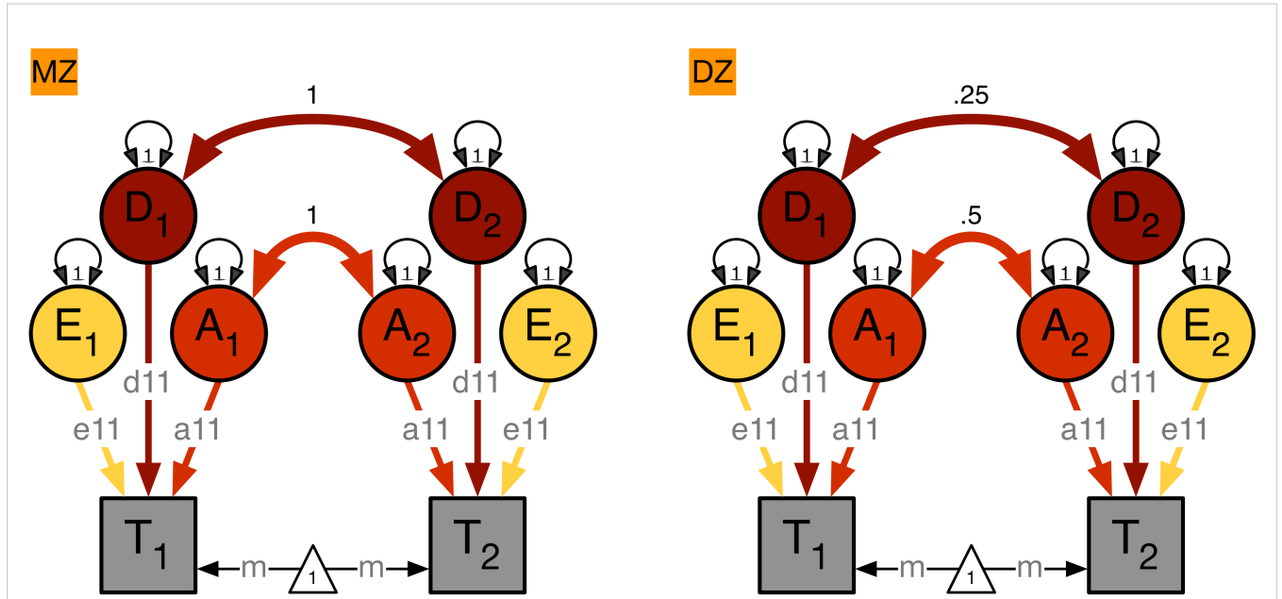
$d_{11} * t(d_{11})$

D *1x1*

$e_{11} * t(e_{11})$

E *1x1*

ADE Deconstructed: *Variances + Covariances*




```
covP       <- mxAlgebra( expression= A+D+E,
 name="V" )

covMZ      <- mxAlgebra( expression= A+D,
name="cMZ" )

covDZ      <- mxAlgebra( expression= 0.5%x%A+ 0.25%x%D,
name="cDZ" )
```

V	A+D+E

V *1x1*

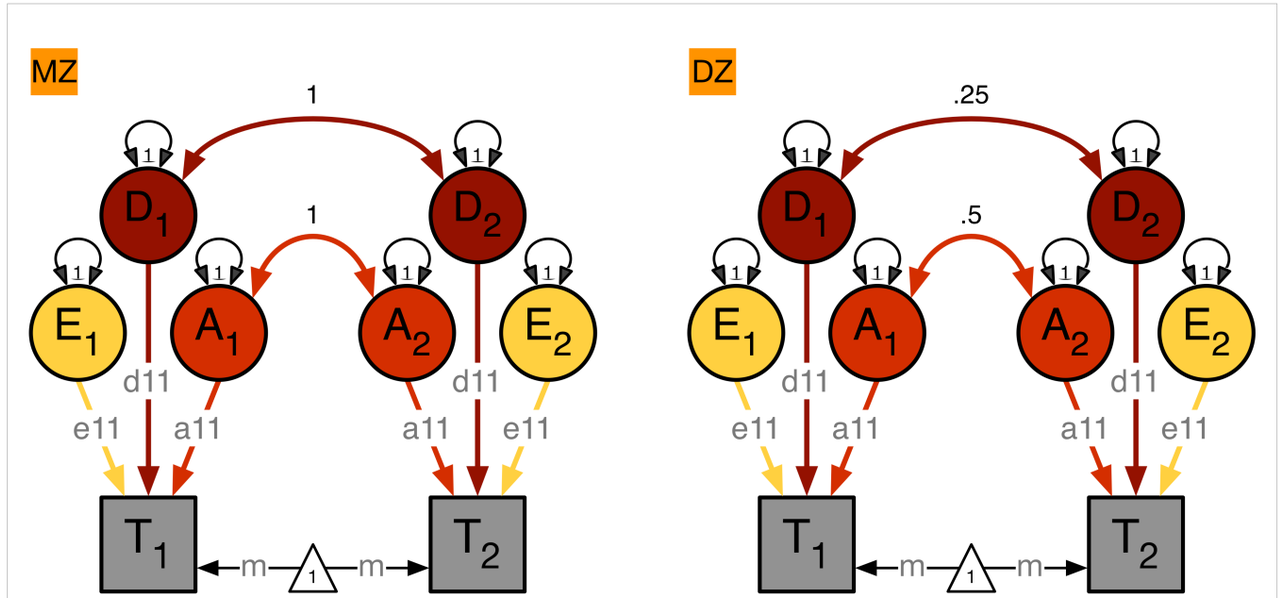
cMZ	A+D

cMZ *1x1*

cDZ	.5A+.25D

cDZ *1x1*

ADE Deconstructed: *Covariance Matrices & Means*




```
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" )



meanG     <- mxMatrix( type="Full", nrow=1, ncol=ntv,
 free=TRUE, values=svMe, labels=labVars("mean",vars),
 name="meanG" )
```

V	cMZ
cMZ	V

expCovMZ *2x2*

V	cDZ
cDZ	V

expCovDZ *2x2*

x_1	x_1

meanG *1x2*

Model Specification

oneADEc.R

```
# ------------------------------------------------------------------------------
# PREPARE MODEL

# ADE Model
# Create Algebra for expected Mean Matrices
meanG     <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe, labels="x1", name="meanG" )

# Create Matrices for Path Coefficients
pathA     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPa, label="a11", lbound=lbPa, name="a" )
pathD     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPa, label="d11", lbound=lbPa, name="d" )
pathE     <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE, values=svPe, label="e11", lbound=lbPa, name="e" )

# Create Algebra for Variance Components
covA      <- mxAlgebra( expression=a %*% t(a), name="A" )
covD      <- mxAlgebra( expression=d %*% t(d), name="D" )
covE      <- mxAlgebra( expression=e %*% t(e), name="E" )

# Create Algebra for expected Variance/Covariance Matrices in MZ & DZ twins
covP      <- mxAlgebra( expression= A+D+E, name="V" )
covMZ     <- mxAlgebra( expression= A+D, name="cMZ" )
covDZ     <- mxAlgebra( expression= 0.5%x%A+ 0.25%x%D, 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" )
```

path matrices: a, d & e

variance components: a^2, d^2 & e^2

Model Specification 2

oneADEc.R

```
# Create Data Objects for Multiple Groups
dataMZ    <- mxData( observed=mzData, type="raw" )
dataDZ    <- mxData( observed=dzData, type="raw" )

# Create Expectation Objects for Multiple Groups
expMZ     <- mxExpectationNormal( covariance="expCovMZ", means="meanG", dimnames=selVars )
expDZ     <- mxExpectationNormal( covariance="expCovDZ", means="meanG", dimnames=selVars )
funML     <- mxFitFunctionML()

# Create Model Objects for Multiple Groups
pars      <- list(meanG, pathA, pathD, pathE, covA, 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") )

# Create Algebra for Variance Components
rowUS     <- rep('US',nv)
colUS     <- rep(c('A','D','E','SA','SD','SE'),each=nv)
estUS     <- mxAlgebra( expression=cbind(A,D,E,A/V,D/V,E/V), name="US", dimnames=list(rowUS,colUS))

# Create Confidence Interval Objects
ciADE     <- mxCI( "US[1,1:3]" )

# Build Model with Confidence Intervals
modelADE  <- mxModel( "oneADEc", pars, modelMZ, modelDZ, multi, estUS, ciADE )
```

list of common elements

ADE variance components

Fitting Nested Models

oneADEc.R

```
# ------------------------------------------------------------------------------
# RUN SUBMODELS

# Run AE model
modelAE   <- mxModel( fitADE, name="oneAEc" )
modelAE   <- omxSetParameters( modelAE, labels="d11", free=FALSE, values=0 )
fitAE     <- mxRun( modelAE, intervals=T )
mxCompare( fitADE, fitAE )
fitGofs(fitAE)
fitEsts(fitAE)

# Run E model
modelE    <- mxModel( fitAE, name="oneEc" )
modelE    <- omxSetParameters( modelE, labels="a11", free=FALSE, values=0 )
fitE      <- mxRun( modelE, intervals=T )
mxCompare( fitAE, fitE )
fitGofs(fitE)
fitEsts(fitE)

# Print Comparative Fit Statistics
mxCompare( fitADE, nested <- list(fitAE, fitE) )
round(rbind(fitADE$US$result,fitAE$US$result,fitE$US$resulT ),4)

# ------------------------------------------------------------------------------
sink()
save.image(paste(filename,".Ri",sep=""))
```

dropping path parameters

Goodness-of-Fit Stats **ADE** **variance estimation**

	os	ep	-2ll	df	AIC	diff -2ll	diff df	p	p/2
Saturated	1777	10	4055.93	1767	521.93				
ADE	1777	4	4063.45	1773	517.45	7.51	6	0.27	
AE	1777	3	4067.66	1774	519.66	4.21	1	0.04	
E	1777	2	4591.79	1775	1041.79	528.34	2	0.00	
path ADE	1777	4	4063.45	1773	517.45	7.51	6	0.27	0.13
path AE	1777	3	4067.66	1774	519.66	4.21	1	0.04	0.02
path E	1777	2	4591.79	1775	1041.79	528.34	2	0.00	0.00

path estimation

Should be divided by 2, as ADE parameters are bounded to be positive
Under the null hypothesis, test is distributed 50:50 as mixture of 0 and a chi-square with 1df

Estimated Values **ADE**

	path coefficients			unstandardized variance components			standardized variance components		
	a	d	e	VA	VD	VE	SA	SD	SE
ADE				0.32 .02-.61	0.29 .01-.60	0.17 .15-.19	0.41	0.37	0.22
AE				0.62	-	0.17	0.78	-	0.22
E				-	-	0.78	-	-	1.00
path ADE	0.57	0.44	0.41	0.32 .02-.61	0.29 .01-.60	0.17 .15-.19	0.41	0.37	0.22
path AE	0.77	-	0.41	0.62	-	0.17	0.78	-	0.22
path E	-	-	0.87	-	-	0.79	-	-	1.00

Goodness-of-Fit Stats **ACE** variance estimation

	os	ep	-2ll	df	AIC	diff -2ll	diff df	p	p/2
Saturated	1777	10	4055.93	1767	521.93				
ACE	1777	4	4063.45	1773	517.45	7.51	6	0.27	
AE	1777	3	4067.66	1774	519.66	4.21	1	0.04	
CE	1777	3	4220.31	1774	672.31	156.86	1	0.00	
E	1777	2	4591.79	1775	1041.79	528.34	2	0.00	
path ACE	1777	4	4067.66	1773	519.66	4.21	6	0.27	0.13
path AE	1777	3	4067.66	1774	519.66	0	1	0.04	0.02
path CE	1777	3	4220.31	1774	672.31	152.65	1	0.00	0.00
path E	1777	2	4591.79	1775	1041.79	524.13	2	0.00	0.00

path estimation

Estimated Values **ACE**

	path coefficients			unstandardized variance components			standardized variance components		
	a	c	e	VA	VC	VE	SA	SC	SE
ACE				0.75	-0.14	0.17	0.97	-0.19	0.22
AE				0.62	-	0.17	0.78	-	0.22
CE				-	0.46	0.32	-	0.59	0.41
E				-	-	0.78	-	-	1.00
path ACE	0.79	0.00	0.41	0.62	0.29	0.17	0.41	0.37	0.22
path AE	0.77	-	0.41	0.62	-	0.17	0.78	-	0.22
path CE	-	0.68	0.56	-	0.46	0.32	-	0.59	0.41
path E	-	-	0.87	-	-	0.79	-	-	1.00

Courtesy of Matt Keller

- When D' is negative in ADE model, then estimates you *would* get in ACE model are:
  - **C' = -1/2 D'** & **A’ = covMZ + 1/2 D'**
  - So i.e., if D' estimate = -.20 & covMZ = .50 in ADE, then in ACE you'll get C'=.10 & A'=.40
- Similarly, when C’ is negative in ACE model, estimates you *would* get in ADE model are:
  - **D' = -2C’** & **A’ = covMZ + 2C'**
  - So i.e., if C' estimate = -.20 & covMZ = .50 in ACE, then in ADE you'll get A'=.10 & D'=.40

Publications 2

- Hao Wu, Michael C Neale: On the Likelihood Ratio Tests in Bivariate ACDE Models. Psychometrika 78 (3), 441-63 Jul 2013.
- Brad Verhulst, Elizabeth Prom-Wormley, Matthew Keller, Sarah Medland, Michael C Neale: Type I Error Rates and Parameter Bias in Multivariate Behavioral Genetic Models. Behav Genet 49 (1), 99-111 Jan 2019.

Thank you !

- functions to run saturated / ADE / ACE models
- **umx**
- Tim Bates