

Univariate/ MonoPhenotype Modeling

Hermine H. Maes, Nick Martin, Elizabeth Prom-Wormley, Tim Bates
& many others

[faculty/hmaes/2016/maes/UnivariateAnalysis](https://faculty.hmaes/2016/maes/UnivariateAnalysis)

Univariate Questions

- Does a trait of interest cluster among related individuals?
- Can clustering be explained by genetic or environmental effects?
- What is the best way to explain the degree to which genetic and environmental effects affect a trait?

scriptsOpenMx.shtml



OpenMx Scripts

2 groups	Continuous	Binary	Ordinal	Mean/Variance Ordinal	Joint	
Univariate Monophenotype	oneSATc oneACEc oneADEc	oneSATb oneACEb oneADEb	oneSATo oneACEo oneADEo	oneSATm oneACEm oneADEm		miFunctions
Bivariate DiPhenotype	twoSATc twoACEc twoADEc	twoSATb twoACEb twoADEb	twoSATo twoACEo twoADEo	twoSATm twoACEm twoADEm	twoSATj twoACEj twoADEj	
with Age Covariate	Continuous	Binary	Ordinal	Mean/Variance Ordinal	Joint	
Univariate Monophenotype	oneSATca oneACEca oneADEca	oneSATba oneACEba oneADEba	oneSAToa oneACEoa oneADEoa	oneSATma oneACEma oneADEma		
Bivariate DiPhenotype	twoSATca twoACEca twoADEca	twoSATba twoACEba twoADEba	twoSAToa twoACEoa twoADEoa	twoSATma twoACEma twoADEma	twoSATja twoACEja twoADEja	
5 groups	Continuous	Binary	Ordinal	Mean/Variance Ordinal	Joint	
Univariate Monophenotype	oneSAT5ca oneACE5ca oneADE5ca	oneSAT5ba oneACE5ba oneADE5ba	oneSAT5oa oneACE5oa oneADE5oa	oneSAT5ma oneACE5ma oneADE5ma		OpenMx Site hmaes@vcu.edu

OpenMx Scripts

- oneSATc

- Saturated model estimating means & variances for continuous data in MZ & DZ twins

- oneACEc

- Univariate/Monophenotype model estimating A, C & E components for continuous data in MZ & DZ twins

- oneADEc

- Univariate/Monophenotype model estimating A, D & E components for continuous data in MZ & DZ twins

miFunctions.R

```
# -----  
# Program: miFunctions.R  
# Author: Hermine Maes  
# Date: 02 25 2016  
#  
# 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="_") }  
labFullSq <- function(lab,nv) { paste(lab,1:nv,rep(1:nv,each=nv),sep="_") }  
labDiag <- function(lab,nv) { paste(lab,1:nv,1:nv,sep="_") }  
labSymm <- function(lab,nv) { paste(lab,rev(nv+1-sequence(1:nv)),rep(1:nv,nv:1),sep="_") }  
labFull <- function(lab,nr,nc) { paste(lab,1:nr,rep(1:nc,each=nr),sep="_") }  
  
...
```

Practical Example

- Dataset: NH&MRC Twin Register
- 1981 Questionnaire
- BMI (body mass index): $\text{weight}/\text{height squared}$
- Young Female Cohort: 18-30 years
- Sample Size:
 - MZf_young: 534 pairs (zyg=1)
 - DZf_young: 328 pairs (zyg=3)

Dataset

```
> head(twinData)
```

```
   fam age zyg part wt1 wt2   ht1   ht2  htwt1  htwt2   bmi1   bmi2
1  115  21   1   2  58  57 1.7000 1.7000 20.0692 19.7232 20.9943 20.8726
2  121  24   1   2  54  53 1.6299 1.6299 20.3244 19.9481 21.0828 20.9519
3  158  21   1   2  55  50 1.6499 1.6799 20.2020 17.7154 21.0405 20.1210
4  172  21   1   2  66  76 1.5698 1.6499 26.7759 27.9155 23.0125 23.3043
5  182  19   1   2  50  48 1.6099 1.6299 19.2894 18.0662 20.7169 20.2583
6  199  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	defVars	
number of factors	nf	<- 2
number of thresholds	nth	<- 3
data objects	mzData / dzData	
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

'Old Fashioned' Data Checking

	MZ		DZ	
	T1	T2	T1	T2
mean	21.34	21.35	21.45	21.46
variance	0.73	0.79	0.77	0.82
covariance	0.59		0.24	

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

Roadmap for Univariate Analysis

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

Intuition behind Maximum Likelihood (ML)

- Likelihood: probability that an observation (data point) is predicted by specified model
- For MLE, determine 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

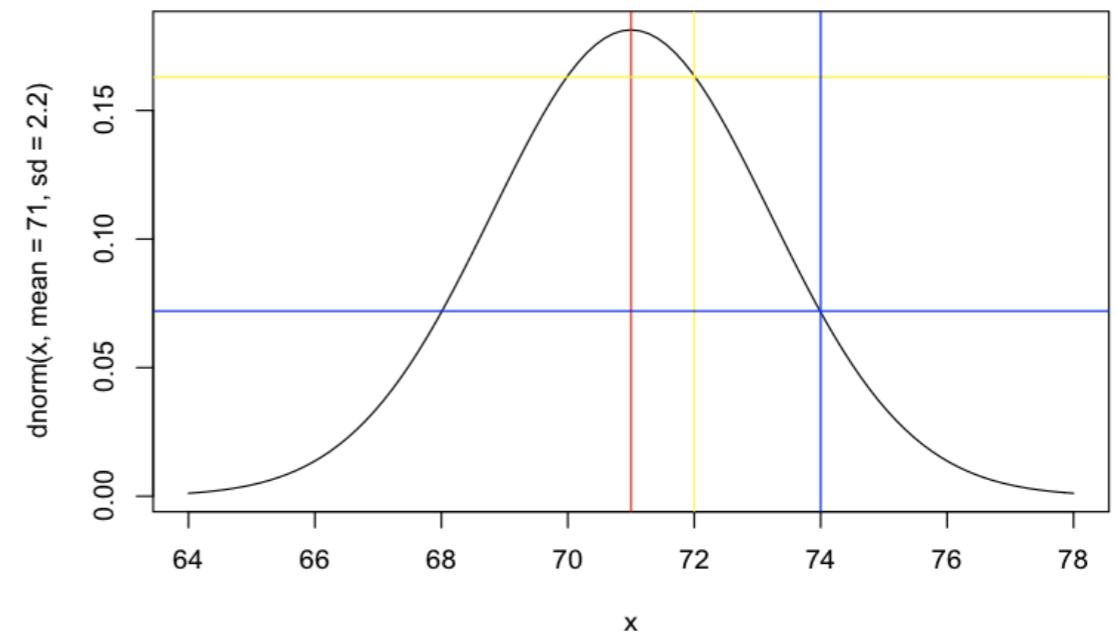
Likelihood Ratio Test

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

Probability Density Function $\Phi(x_i)$

- $\Phi(x_i)$ is likelihood of data point x_i for particular mean and variance estimates
- Univariate: height of probability density function

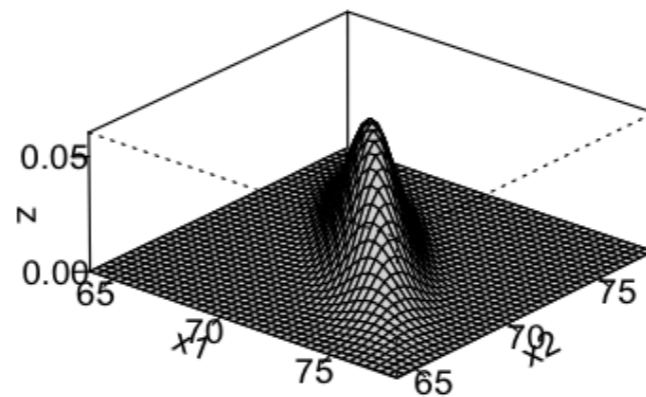
$$\Phi(x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-5((x_i-\mu)^2/\sigma^2)}$$



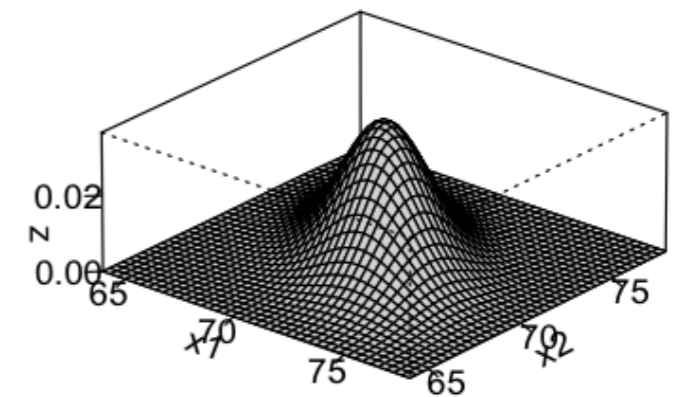
π : $\pi=3.14$; x_i : observed value of variable i ; μ : expected mean; σ : expected variance

Multinormal Probability Function

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



rMZ=.85

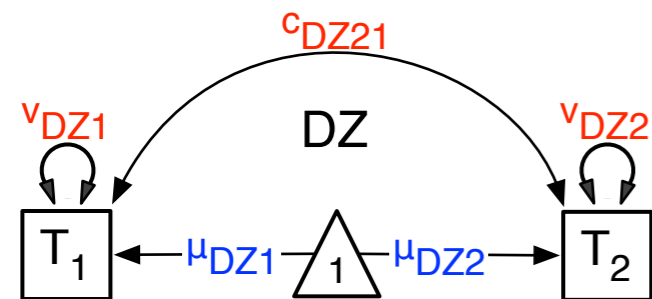
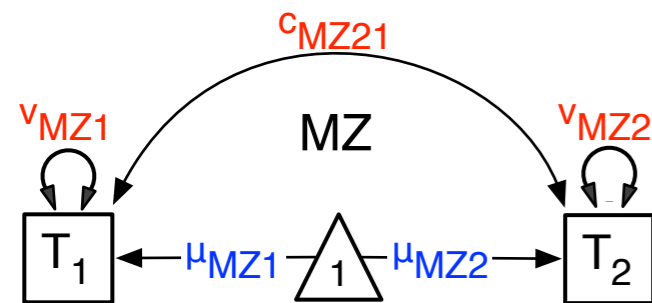


rDZ=.49

$$\Phi(x_i) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}((x_i-\mu)\Sigma^{-1}(x_i-\mu)')}$$

$\pi = 3.14$; x_i : value of variable i ; μ : expected mean; Σ : expected covariance matrix

Predicted Means



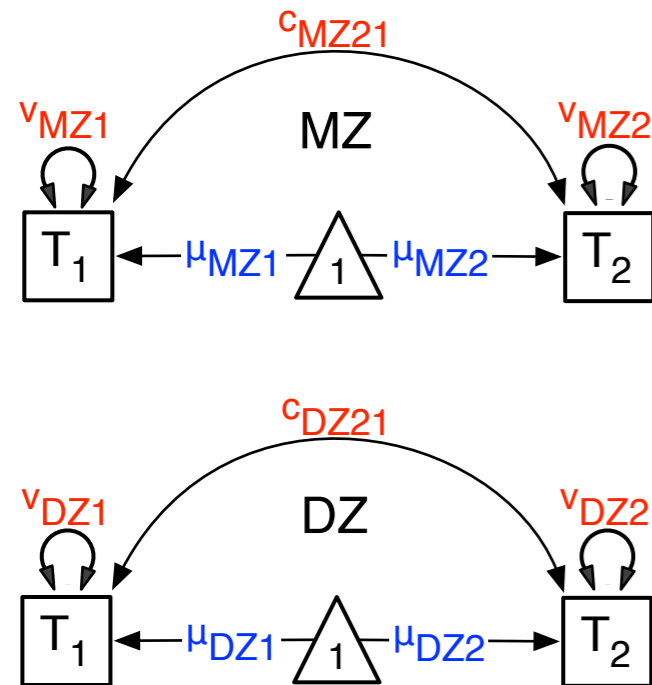
1x2 matrices

T1	T2
mMZ1	mMZ2

T1	T2
mDZ1	mDZ2

```
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" )
```

Predicted Covariances



2x2 matrices

	T1	T2
T1	$v1MZ$	$c21MZ$
T2	$c21MZ$	$v2MZ$

	T1	T2
T1	$v1DZ$	$c21DZ$
T2	$c21DZ$	$v2DZ$

```

covMZ      <- mxMatrix( type="Symm", nrow=ntv, ncol=ntv, free=TRUE,
values=svVas, lbound=lbVas, labels=c("vMZ1", "cMZZ1", "vMZ2"), name="covMZ" )
covDZ      <- mxMatrix( type="Symm", nrow=ntv, ncol=ntv, free=TRUE,
values=svVas, lbound=lbVas, labels=c("vDZ1", "cDZZ1", "vDZ2"), name="covDZ" )
    
```

Univariate Saturated Model

oneSATc.R

```
# -----  
# Program: oneSATc.R  
# Author: Hermine Maes  
# Date: 02 25 2016  
#  
# Twin Univariate Saturated model to estimate means and (co)variances across multiple groups  
# Matrix style model - Raw data - Continuous data  
# -----|-----|-----|-----|-----|-----|-----|-----|
```

```
# Load Libraries & Options  
library(OpenMx)  
library(psych)
```

```
source("miFunctions.R")
```



my functions which you can edit as you like

```
# Create Output
```

```
filename <- "oneSATc"
```

```
sink(paste(filename, ".Ro", sep=""), append=FALSE, split=TRUE)
```



creates output file with extension .Ro

Preparing Data

oneSATc.R

```
# -----  
# PREPARE DATA  
  
# Load Data  
data(twinData)  
dim(twinData)  
describe(twinData, 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)  
dzData    <- subset(twinData, zyg==3, selVars)  
  
# Set Starting Values  
svMe      <- 20             # start value for means  
svVa      <- .8            # start value for variance  
svVas     <- diag(svVa,ntv,ntv) # assign start values to diagonal of matrix  
lbVa      <- .0001        # start value for lower bounds  
lbVas     <- diag(lbVa,ntv,ntv) # assign lower bounds values to diagonal of matrix
```

analyzing c('bmi1','bmi2')

get right codes for zygosity

Preparing Model

oneSATc.R

```
# -----  
# PREPARE MODEL  
  
# Saturated 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=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" )  
  
# Create Algebra for Maximum Likelihood Estimates of Twin Correlations  
matI <- mxMatrix( type="Iden", nrow=ntv, ncol=ntv, name="I" )  
corMZ <- mxAlgebra( solve(sqrt(I*covMZ)) %&% covMZ, name="corMZ" )  
corDZ <- mxAlgebra( solve(sqrt(I*covDZ)) %&% covDZ, name="corDZ" )  
  
# 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()  
  
# Create Model Objects for Multiple Groups  
modelMZ <- mxModel( name="MZ", meanMZ, covMZ, matI, corMZ, dataMZ, expMZ, funML )  
modelDZ <- mxModel( name="DZ", meanDZ, covDZ, matI, corDZ, dataDZ, expDZ, funML )  
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  
model <- mxModel( "oneSATc", modelMZ, modelDZ, multi, ciCov, ciMean )
```

formula to calculate correlations

fitting to raw data

using FIML: full information maximum likelihood

evaluating 2 groups simultaneously

built model

Run Model

oneSATc.R

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

```
# Run Saturated Model
```

```
fit <- mxRun( model, intervals=F )
```

```
sum <- summary( fit )
```

```
round(fit$output$estimate,4)
```

```
fitGofs(fit)
```

standard summary function in OpenMx

fitted model

my short summary function in miFunctions.R

summary(oneSATc)

free parameters:

	name	matrix	row	col	Estimate	Std.Error	A	lbound	ubound
1	mMZ1	MZ.meanMZ	1	bmi1	21.3443773	0.03606189			
2	mMZ2	MZ.meanMZ	1	bmi2	21.3490123	0.03765093			
3	vMZ1	MZ.covMZ	bmi1	bmi1	0.7276713	0.04365878		1e-04	
4	cMZ21	MZ.covMZ	bmi1	bmi2	0.5916404	0.04079381		0	
5	vMZ2	MZ.covMZ	bmi2	bmi2	0.7932021	0.04764761		1e-04	
6	mDZ1	DZ.meanDZ	1	bmi1	21.4475210	0.04757192			
7	mDZ2	DZ.meanDZ	1	bmi2	21.4578443	0.04923334			
8	vDZ1	DZ.covDZ	bmi1	bmi1	0.7691912	0.05900857		1e-04	
9	cDZ21	DZ.covDZ	bmi1	bmi2	0.2400401	0.04520164		0	
10	vDZ2	DZ.covDZ	bmi2	bmi2	0.8216316	0.06315491		1e-04	

observed statistics: 1777

estimated parameters: 10

degrees of freedom: 1767

fit value (-2lnL units): 4055.935

number of observations: 920

Information Criteria:

	df	Penalty	Parameters	Penalty	Sample-Size	Adjusted
AIC:		521.9346		4075.935		NA

Estimated Values

		T1	T2		T1	T2
Saturated Model						
mean	MZ	21.34	21.35	DZ	21.45	21.46
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, mDZ1, vDZ1, vDZ2, cDZ21

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
Sat	10	4055.94	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
diff -2ll	likelihood ratio Chi-square		
diff df	difference in degrees of freedom		

Fitting Nested Models

oneSATc.R

```
# Constrain expected Means to be equal across twin order
modelEMO <- mxModel(fit, name="eqMeansTwin" )
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 )
mxCompare(fit, fitEMO)
round(fitEMO$output$estimate,4)
fitGofs(fitEMO)
```

changing parameters

generate likelihood ratio test

```
# Constrain expected Means and Variances to be equal across twin order
modelEMVO <- mxModel(fitEMO, name="eqMVarsTwin" )
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 )
mxCompare(fit, subs <- list(fitEMO, fitEMVO) )
round(fitEMVO$output$estimate,4)
fitGofs(fitEMVO)
```

```
# Constrain expected Means and Variances to be equal across twin order and zygosity
modelEMVZ <- mxModel(fitEMVO, name="eqMVarsZyg" )
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 )
mxCompare(fit, subs <- list(fitEMO, fitEMVO, fitEMVZ) )
round(fitEMVZ$output$estimate,4)
fitGofs(fitEMVZ)
```

Estimated Values

		T1	T2		T1	T2
Equate Means & Variances across Twin Order						
mean	MZ			DZ		
cov	T1			T1		
	T2			T2		
Equate Means Variances across Twin Order & Zygosity						
mean	MZ			DZ		
cov	T1			T1		
	T2			T2		

Goodness-of-Fit Stats

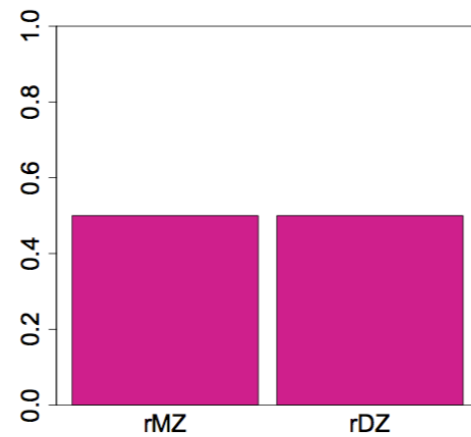
	ep	-2ll	df	AIC	diff -2ll	diff df	p
Saturated	10	4055.93	1767	521.93			
mT1=mT2	8	4056.00	1769	518.00	0.07	2	0.97
mT1=mT2 varT1=varT2	6	4058.94	1771	516.94	3.01	4	0.56
Zyg MZ=DZ	4	4063.45	1773	517.45	7.52	6	0.28

Patterns of Twin Correlations

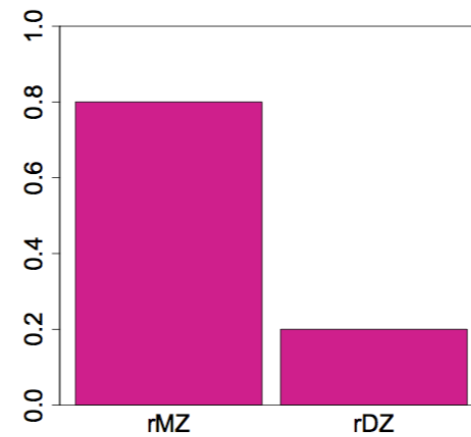


$r_{MZ} = 2r_{DZ}$
A

DZ's on average share 50% of additive genetic effects



$r_{MZ} = r_{DZ}$
C



$r_{MZ} > 2r_{DZ}$
A+D

DZ's on average share 25% of dominance genetic effects



$r_{MZ} < 2r_{DZ}$
A+C

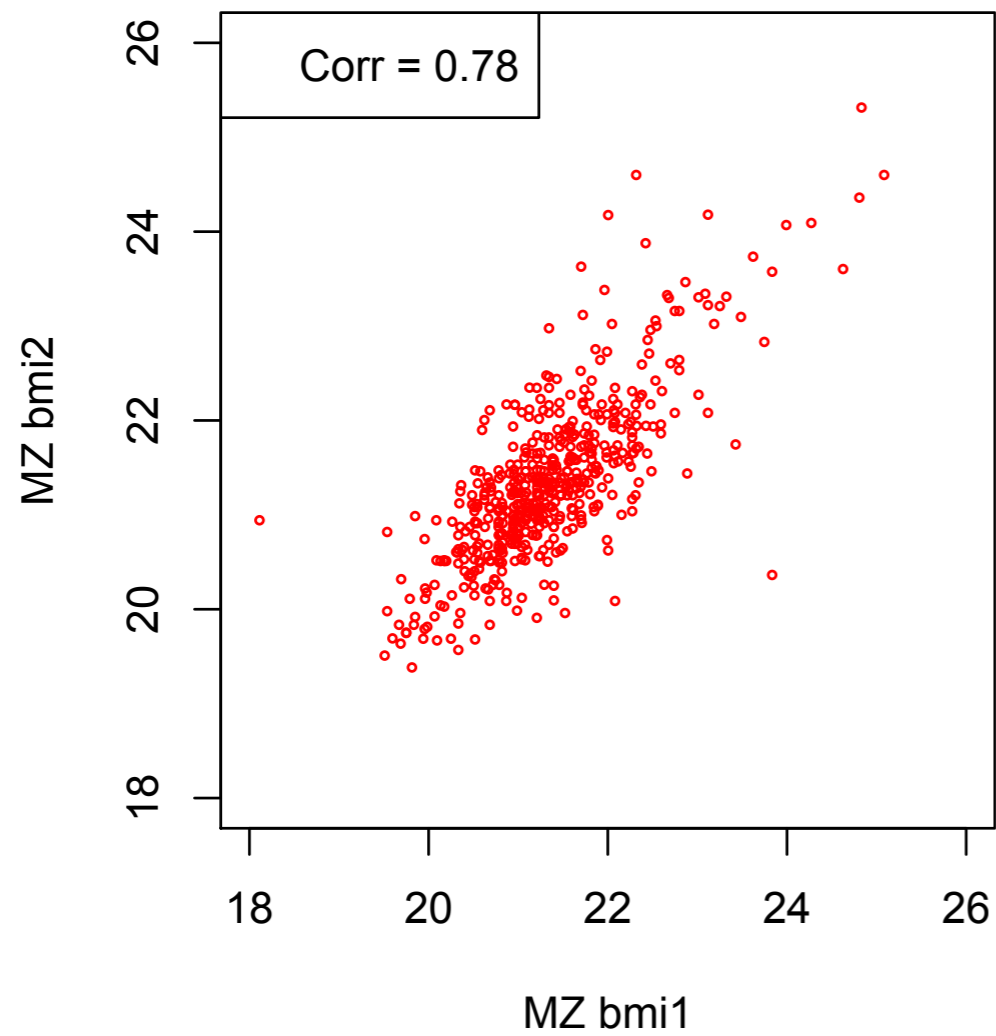
$A = 2(r_{MZ} - r_{DZ})$
 $C = 2r_{DZ} - r_{MZ}$
 $E = 1 - r_{MZ}$

Twin Correlations ~ Sources of Variance

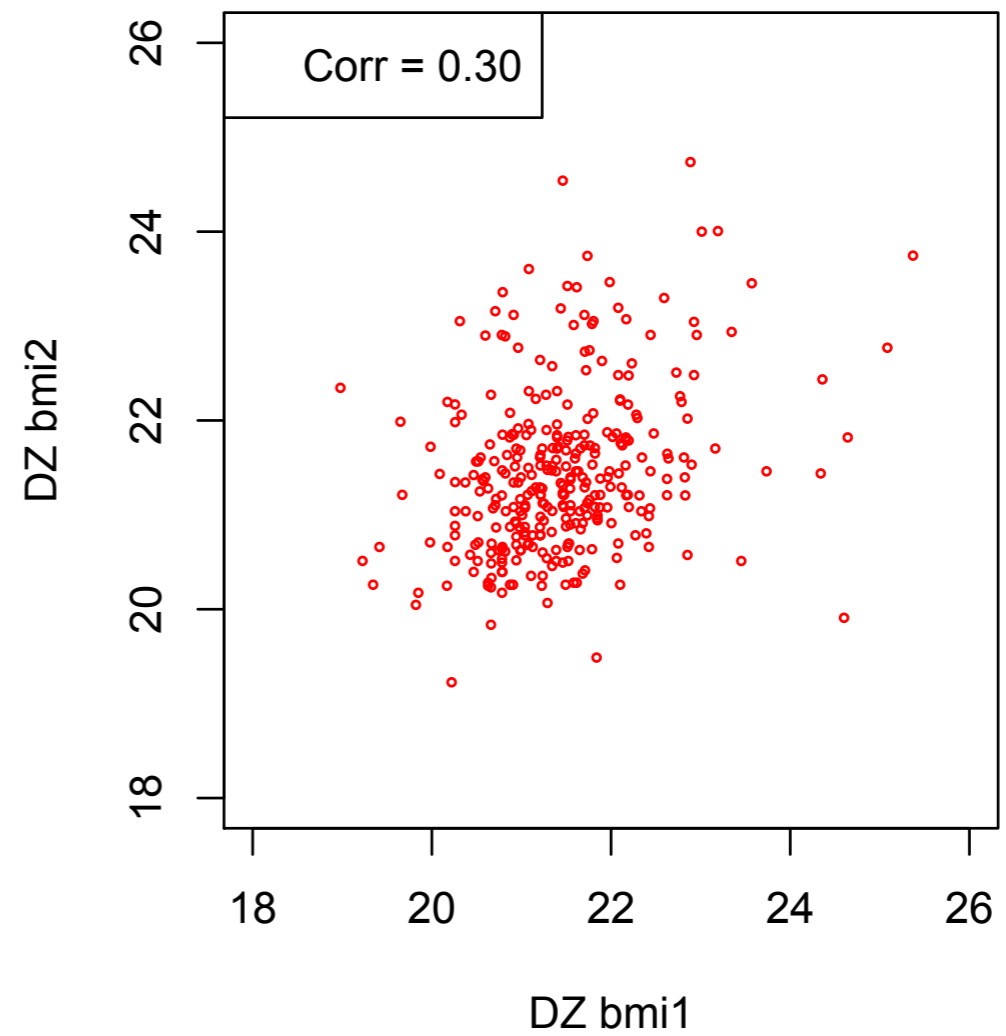
$1 - r_{MZ}$	E
$r_{MZ} > r_{DZ}$	A
$r_{MZ} = 2 r_{DZ}$	only A
$r_{MZ} = r_{DZ}$	only C
$r_{MZ} < 1/2 r_{DZ}$	A & C
$r_{MZ} > 1/2 r_{DZ}$	A & D

Example Twin Correlations

MZ BMI

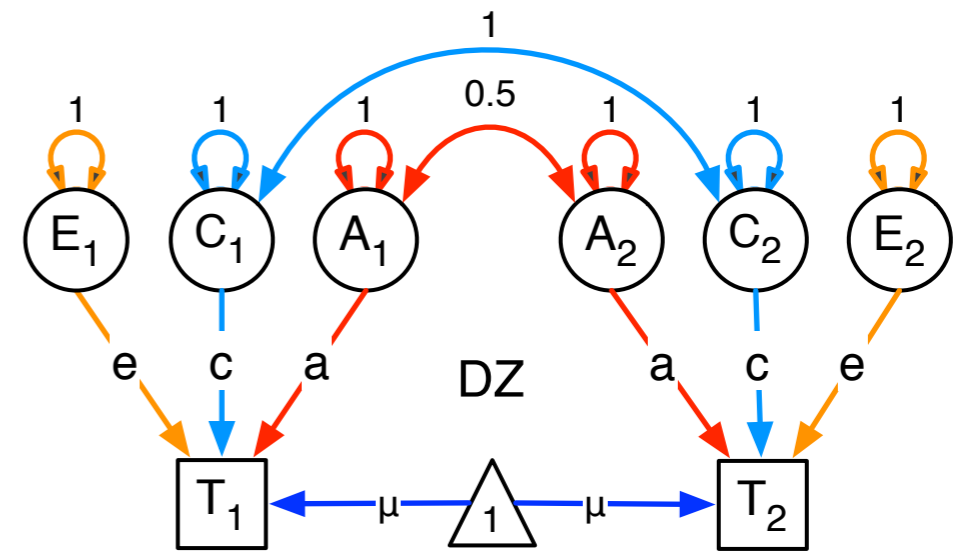
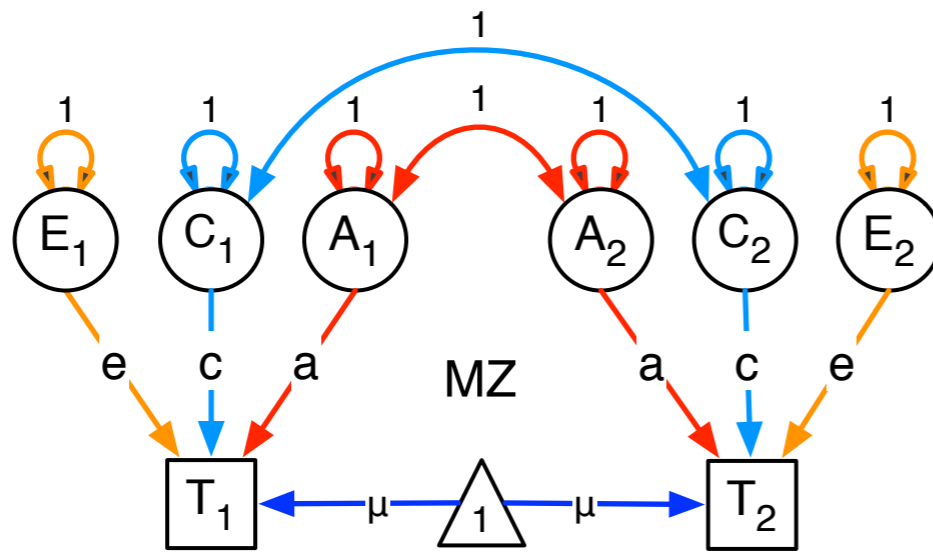


DZ BMI

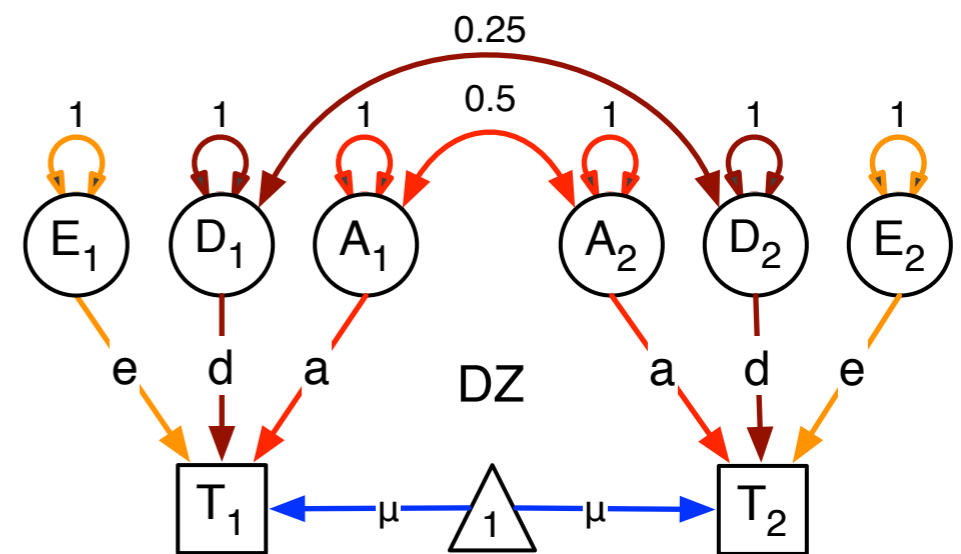
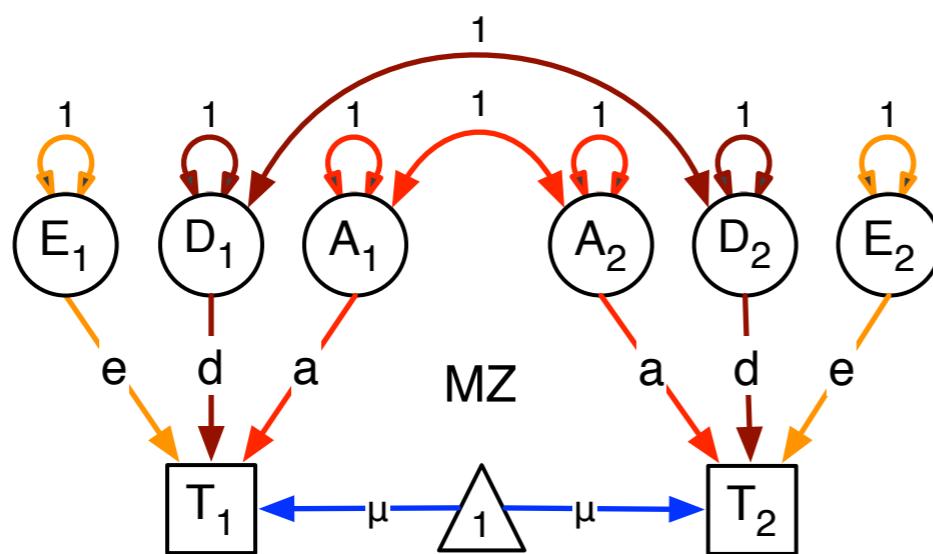


Univariate ACE / ADE Model

ACE



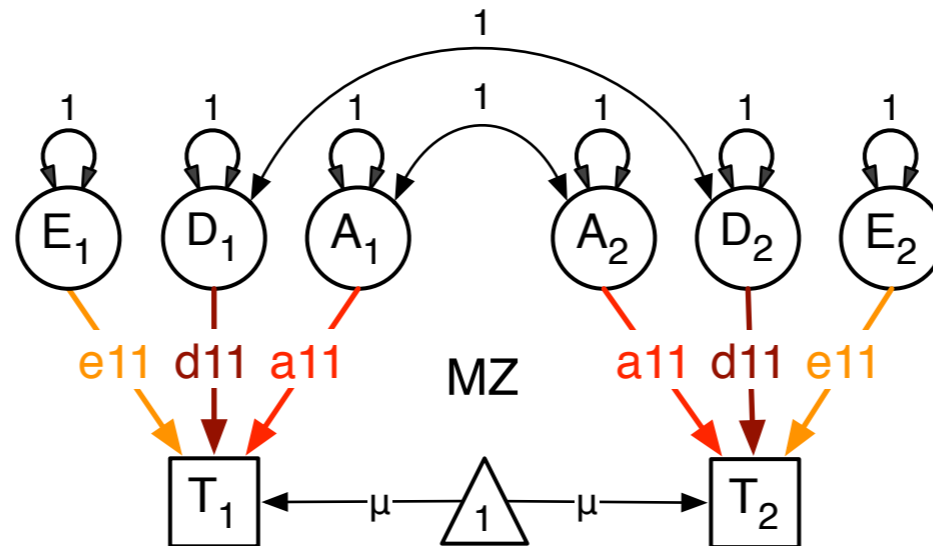
ADE



Roadmap for Univariate Analysis

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

ADE Deconstructed: *Path Coefficients*



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

a 1 1
a 1 x 1

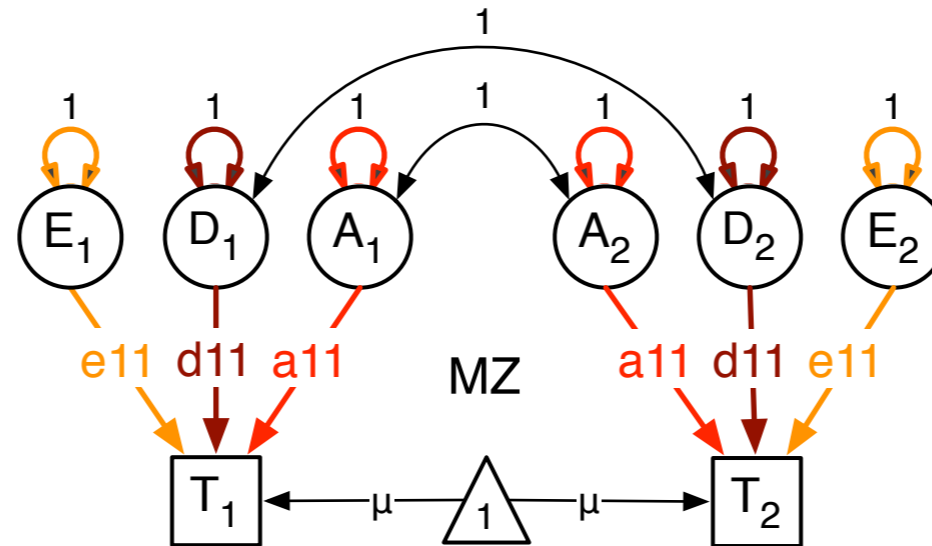
```
pathD      <- mxMatrix( type="Lower", nrow=nv, ncol=nv,
  free=TRUE, values=svPa, label="d11", lbound=lbPa, name="d" )
```

d 1 1
d 1 x 1

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

e 1 1
e 1 x 1

ADE Deconstructed: *Variance Components*



```

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

```

```

covC      <- mxAlgebra( expression=c %*% t(c),
name="C"  )

```

```

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

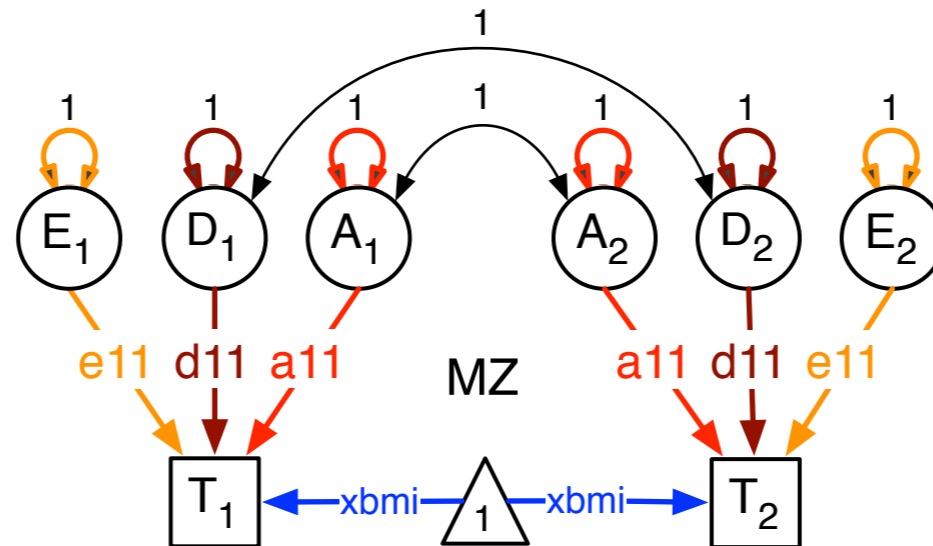
```

$$\begin{matrix} \boxed{\text{a11}} & * & \boxed{\text{a11T}} \\ A & 1 \times 1 & \end{matrix}$$

$$\begin{matrix} \boxed{\text{d11}} & * & \boxed{\text{d11T}} \\ D & 1 \times 1 & \end{matrix}$$

$$\begin{matrix} \boxed{\text{e11}} & * & \boxed{\text{e11T}} \\ E & 1 \times 1 & \end{matrix}$$

ADE Deconstructed: Means & Variances



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

xbmi	xbmi
------	------

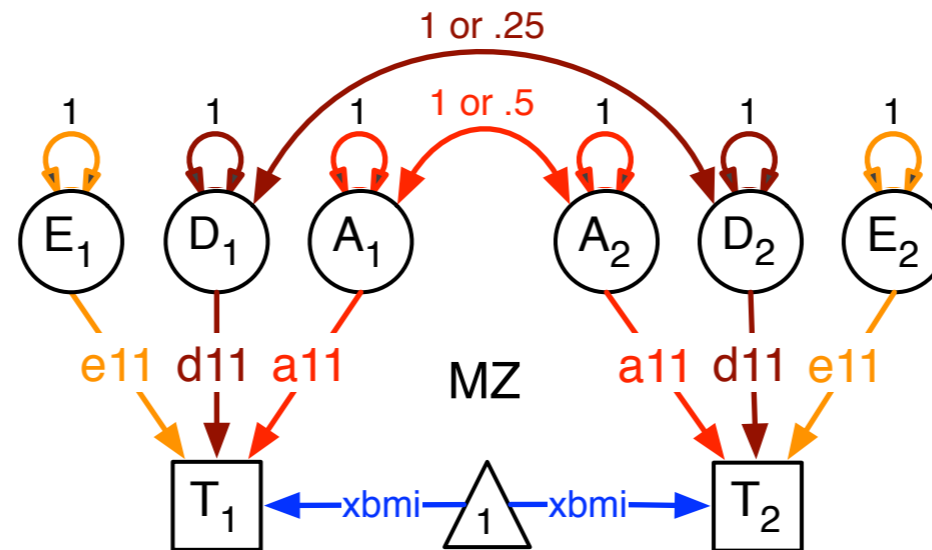
expMean 1 x 2

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

V	
	V

expCovMZ 2 x 2
expCovDZ 2 x 2

ADE Deconstructed: $A+D$ Covariances



```
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" )
```

V	A+D
A+D	V

expCovMZ 2 x 2

V	.5A+.25D
.5A+.25D	V

expCovDZ 2 x 2

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="xbmi", 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= 0.5x%A+ 0.25x%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" )
  
```

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

paths

V	cMZ
cMZ	V

V	cDZ
cDZ	V

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( name="MZ", pars, covMZ, expCovMZ, dataMZ, expMZ, funML )
modelDZ <- mxModel( name="DZ", pars, covDZ, expCovDZ, dataDZ, expDZ, funML )
multi   <- mxFitFunctionMultigroup( c("MZ", "DZ") )

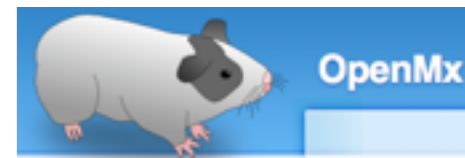
# Create Algebra for Variance Components
rowVC   <- rep('VC',nv)
colVC   <- rep(c('A','D','E','SA','SD','SE'),each=nv)
estVC   <- mxAlgebra( expression=cbind(A,D,E,A/V,D/V,E/V), name="VC", dimnames=list(rowVC,colVC))

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

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

list of common elements

ADE model



Run Model

oneADEc.R

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

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

```
# Compare with Saturated Model  
mxCompare( fit, fitADE )  
#lrtSAT( fitADE, 4055.9346, 1767)
```

function in miFunctions.R to provide $-2\ln$ & df of previously fit model

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

summary(oneADEc)

free parameters:

	name	matrix	row	col	Estimate	Std.Error	A	lbound	ubound
1	xbmi	meanG	1	1	21.3946487	0.02597351			
2	a11	a	1	1	0.5665073	0.13323546		1e-04	
3	d11	d	1	1	0.5379822	0.13748771		1e-04	
4	e11	e	1	1	0.4115217	0.01259157		1e-04	

confidence intervals:

	lbound	estimate	ubound	note
oneADEc.VC[1,1]	0.02648494	0.3209305	0.6111811	
oneADEc.VC[1,2]	0.01256943	0.2894248	0.5898489	
oneADEc.VC[1,3]	0.15055641	0.1693501	0.1913860	

observed statistics: 1777

estimated parameters: 4

degrees of freedom: 1773

fit value (-2lnL units): 4063.45

number of observations: 920

Information Criteria:

	df	Penalty
AIC:	517.4496	

Goodness-of-Fit Stats

	ep	-2ll	df	AIC	diff -2ll	diff df	p
Saturated	10	4055.93	1767	521.93			
ADE	5	4067.66	1773	521.66	11.73	6	0.07

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=""))  
}
```

```
fitGofS <- function(fit) {  
  summ <- summary(fit)  
  cat(paste("Mx:", fit$name, " #statistics=", summ$ob, " #records=", summ$nu, " #parameters=", summ$es,  
           " #constraints=", sum(summ$cons), " df=", summ$de, " -2LL=", round(summ$Mi,4),  
           " cpu=", round(summ$cpu,4), " optim=", summ$op, " version=", summ$mx,  
           " code=", fit$output$status$code, "\n", sep=""))  
}
```

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

miFunctions: fitGofs & fitEsts

```
> fitGofs(fitADE)
Mx:oneADEc  os=1777  ns=920  ep=4  co=0  df=1773  ll=4063.4496
  cpu=0.8816  opt=NPSOL  ver=2.3.1  stc=0

> fitEsts(fitADE)
  xbmi      a11      d11      e11
21.3946  0.5665  0.5380  0.4115
      A      D      E      SA      SD      SE
VC 0.3209  0.2894  0.1694  0.4116  0.3712  0.2172
      lbound estimate ubound
oneADEc.VC[1,1] 0.0163  0.3209  0.6121
oneADEc.VC[1,2] 0.0119  0.2894  0.5956
oneADEc.VC[1,3] 0.1694  0.1694  0.1914
```

Roadmap for Univariate Analysis

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

Nested Models

- 'Full' ADE Model
- Nested Models
 - AE Model: test significance of D
 - E Model vs AE Model: test significance of A
 - E Model vs ADE Model: test combined significance of A & D



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$VC$result, fitAE$VC$result, fitE$VC$result ),4)  
  
# -----  
sink()  
save.image(paste(filename, ".Ri", sep=""))
```

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
ADE	5	4063.45	1773	517.45			
AE	4	4067.66	1774	519.66	4.21	1	0.04
E	3	4591.79	1775	1041.79	528.34	2	0.00

should be divided by 2, as ADE parameters are bounded to be positive

Under the null hypothesis,
the test is distributed 50:50 as mixture of 0 and a chi-square with 1 df

Estimated Values

	path coefficients			unstandardized variance components			standardized variance components		
	a	d	e	a	d	e	a	d	e
ADE	0.57	0.44	0.41	0.32	0.29	0.17	0.41	0.37	0.22
AE	0.77	-	0.41	0.62	-	0.17	0.78	-	0.22
E	-	-	0.87	-	-	0.79	-	-	1.00

What about C?

- 'Full' ACE Model
- Nested Models
 - AE Model: test significance of C
 - CE Model: test significance of A
 - E Model vs AE Model: test significance of A
 - E Model vs ACE Model: test combined significance of A & C

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
ADE							
AE							
ACE							
CE							
E							

Estimated Values

	a	d	e	c	a	d	e	c
ADE								
AE								
ACE								
AE								
E								

Conclusions

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