Univariate Twin Analysis

OpenMx Tc24 2010
Nick Martin, Hermine Maes

Upgrade OpenMx

- quit R
- go to Workshop website
- find OpenMx website
- open R
- copy line in R console
  - source('http://openmx.psyc.virginia.edu/getOpenMx.R')
- execute line
- quit R

Files to Copy

- create new directory: e.g. H:\hermine
- copy files from F:\hmaes\hermine to new directory
- close R
- double-click on first script:
  - UnivariateTwinSaturated_MatrixRawCon.R
  - UnivariateTwinAnalysis_MatrixRawConADE.R
  - UnivariateTwinAnalysis_MatrixRawConACE.R
  - GenEpiHelperFunctions.R

Practical Example

- Dataset: NH&MRC Twin Register
- 1981 Questionaire
- BMI (body mass index): weight/height squared
- Young 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

....
```

Univariate Twin Saturated

UnivariateTwinSaturated_MatrixRawCon.R [1]

```
# Prepare Data
# ------------------------------------------------------------------------------
data(twinData)
require(psych)
describe(twinData)
Vars <- 'bmi'
nv <- 1
selVars <- paste(Vars,c(rep(1,nv),rep(2,nv)),sep="")   #c('bmi1','bmi2')
ntv <- nv*2
mzData <- subset(twinData, zyg==1, selVars)
dzData <- subset(twinData, zyg==3, selVars)

# Print Descriptive Statistics
# ------------------------------------------------------------------------------
describe(mzData)
colMeans(mzData,na.rm=TRUE)
cov(mzData,use="complete")
cor(mzData,use="complete")
....
```

Naming Conventions for this workshop

variable	Vars <- 'bmi'
number of variables	nv <- 1
variables per twin pair	selVars <- c('bmi1','bmi2')
number of twin variables	ntv <- nv*2
number of factors	nf <- 2
number of thresholds	nth <- 3
MZ data	mzData
DZ data	dzData

Univariate Twin Saturated

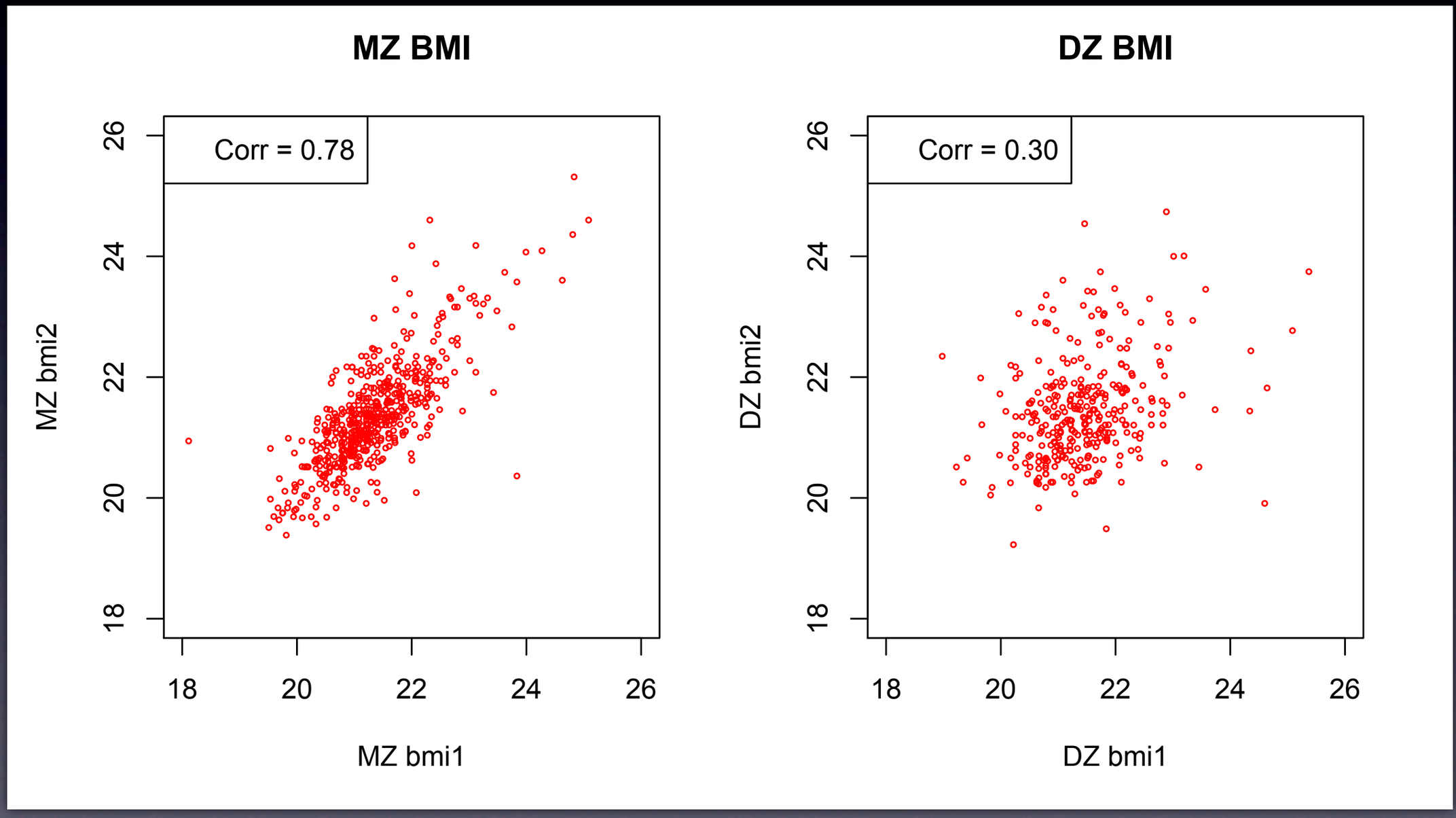
UnivariateTwinSaturated_MatrixRawCon.R [2]

```
# Graph Descriptive Statistics
# ------------------------------------------------------------------
attach(twinData)
par(mfcol=c(1,2))
hist(bmi1[zyg==1])
hist(bmi2[zyg==1])
hist(bmi1[zyg==3])
hist(bmi2[zyg==3])

plot(bmi1[zyg==1],bmi2[zyg==1])
plot(bmi1[zyg==3],bmi2[zyg==3])

# Make them prettier
plot(bmi1[zyg==1],bmi2[zyg==1],xlab="MZ bmi1",ylab="MZ bmi2",
main="MZ BMI",xlim=c(18,26),ylim=c(18,26),col="red",cex=.4)
legend("topleft","Corr = 0.78")
plot(bmi1[zyg==3],bmi2[zyg==3],xlab="DZ bmi1",ylab="DZ bmi2",
main="DZ BMI",xlim=c(18,26),ylim=c(18,26),col="red",cex=.4)
legend("topleft","Corr = 0.30")
```

Twin Correlations



ML Means & Variances

Multinormal Probability Density Function:

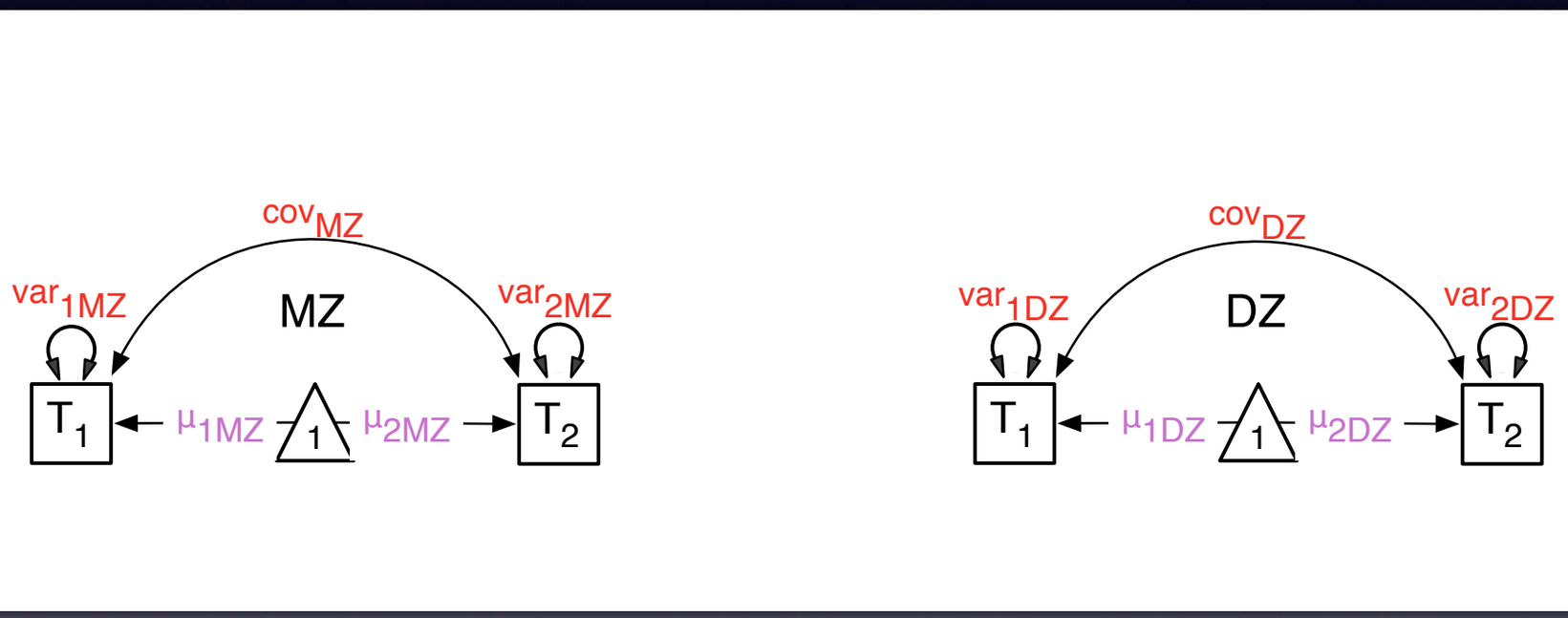
$$-|2\pi\Sigma|^{-n/2}\, e^{-.5((x_i - \mu)\,\Sigma^{-1}\,(x_i - \mu)')}$$

make use of all available data

get unbiased estimates if missing data are missing at random

Use Maximum Likelihood to estimate free Parameters:
2 means, 2 variances, 1 covariance

Saturated Twin Model



OpenMx Commands

mxModel(model = NA, ..., manifestVars = NA, latentVars = NA,
remove = FALSE, independent = NA, type = NA, name = NA)

```
mxModel("univTwinSat",....
mxModel(univTwinSatModel,
```

mxMatrix(type = "Full", nrow = NA, ncol = NA, free = FALSE, values = NA,
labels = NA, lbound = NA, ubound = NA, byrow = getOption('mxByrow'),
dimnames = NA, name = NA)

```
mxMatrix( type="Lower", nrow=ntv, ncol=ntv, free=TRUE, values=.5, name="CholMZ" )
```

mxAlgebra(expression, name = NA, dimnames = NA)

```
mxAlgebra( expression=CholMZ %*% t(CholMZ), name="expCovMZ" )
```

OpenMx Commands

mxData(observed, type = NA, means = NA, numObs = NA)

```
mxData( observed=mzData, type="raw" )
```

mxFIMLObjective(covariance, means, dimnames, thresholds)

```
mxFIMLObjective( covariance="expCovMZ",means="expMeanMZ", dimnames=selVars)
```

mxAlgebraObjective(algebra)

```
mxAlgebraObjective( algebra="-2sumll")
```

mxRun(mxModel)

```
mxRun(univTwinSatModel)
```

mxEval(expression, model, compute, show)

```
mxEval( expression= cbind(ACE.a,ACE.d,ACE.e), model=univACEFit)
```

R Matrix operators

classic Mx	R OpenMx		classic Mx	R OpenMx	
~	solve()	inversion	\trace()	tr()	trace
‘	t()	transposition	\det()	det()	determinant
+	+	addition	\sum()	sum()	sum
-	-	subtraction	\max()	max()	maximum
*	%*%	matrix multiplication	\min()	min()	minimum
.	*	dot product	\abs()	abs()	absolute value
%	/	element division	\exp()	exp()	exponent
@	%x%	Kronecker product	\ln()	log()	natural logarithm
&	%&%	quadratic product	\sqrt()	sqrt()	square root
|	cbind()	horizontal adhesion	\d2v()	diag2vec()	diagonal 2 vector
_	rbind()	vertical adhesion	\v2d()	vec2diag()	vector 2 diagonal
^		power	\vech()	vech()	lower 2 vector

OpenMx uses R precedence rules

http://openmx.psyc.virginia.edu/wiki/matrix-operators-and-functions

Univariate Twin Saturated

UnivariateTwinSaturated_MatrixRawCon.R [3]

```
# Fit Saturated Model
# ------------------------------------------------------------------------------
univTwinSatModel <- mxModel("univTwinSat",
    mxModel("MZ",
        mxMatrix( type="Lower", nrow=ntv, ncol=ntv, free=TRUE, values=.5, name="CholMZ" ),
        mxAlgebra( expression=CholMZ %*% t(CholMZ), name="expCovMZ" ),
        mxMatrix( type="Full", nrow=1, ncol=ntv, free=T, values=20, name="expMeanMZ" ),
        mxData( observed=mzData, type="raw" ),
        mxFIMLObjective( covariance="expCovMZ",means="expMeanMZ", dimnames=selVars)
    # Algebra's needed for equality constraints
       mxAlgebra( expression=expMeanMZ[1,1:nv], name="expMeanMZt1"),
        mxAlgebra( expression=expMeanMZ[1,(nv+1):ntv], name="expMeanMZt2"),
        mxAlgebra( expression=t(diag2vec(expCovMZ)), name="expVarMZ"),
        mxAlgebra( expression=expVarMZ[1,1:nv], name="expVarMZt1"),
        mxAlgebra( expression=expVarMZ[1,(nv+1):ntv], name="expVarMZt2")
    ),
```

Cholesky Decomposition

UnivariateTwinSaturated_MatrixRawCon.R [4]

```
....
    mxModel("DZ",
        mxMatrix( type="Lower", nrow=ntv, ncol=ntv, free=TRUE, values=.5, name="CholDZ" ),
        mxAlgebra( expression=CholDZ %*% t(CholDZ), name="expCovDZ" ),
        mxAlgebra( expression=diag2vec(expCovDZ), name="expVarDZ"),
        mxMatrix( type="Full", nrow=1, ncol=ntv, free=T, values=20, name="expMeanDZ" ),
        mxData( observed=dzData, type="raw" ),
        mxFIMLObjective( covariance="expCovDZ", "expMeanDZ", dimnames=selVars)
      # Algebra's needed for equality constraints
        mxAlgebra( expression=expMeanDZ[1,1:nv], name="expMeanDZt1"),
        mxAlgebra( expression=expMeanDZ[1,(nv+1):ntv], name="expMeanDZt2"),
        mxAlgebra( expression=t(diag2vec(expCovDZ)), name="expVarDZ"),
        mxAlgebra( expression=expVarDZ[1,1:nv], name="expVarDZt1"),
        mxAlgebra( expression=expVarDZ[1,(nv+1):ntv], name="expVarDZt2")
    ),
```

Model - Fit - Summ

UnivariateTwinSaturated_MatrixRawCon.R [5]

```
....
    mxAlgebra( MZ.objective + DZ.objective, name="-2sumll" ),
    mxAlgebraObjective("-2sumll")
 )

univTwinSatFit <- mxRun(univTwinSatModel)
univTwinSatSumm <- summary(univTwinSatFit)
```

OpenMx Process

testModel ←

```
mxModel( name="myModelName"
    mxMatrix(),
    mxMatrix(),
    mxAlgebra(),
    mxConstraint(),
    mxModel( mxMatrix(),
             mxData(),
             mxObjective(),
    ),
)
```

testFit ← mxRun(testModel)

testSumm ← summary(testFit)

output ← mxEval(algebra, testFit)

MxModel R Objects

R Objects

summary(MxModel)

```
>univTwinSatSumm
   name       matrix row  col   Estimate  Std.Error
1  <NA>    MZ.CholMZ   1    1  0.8530361 0.01789560
2  <NA>    MZ.CholMZ   2    1  0.6935690 0.02226836
3  <NA>    MZ.CholMZ   2    2  0.5587162 0.01188896
4  <NA> MZ.expMeanMZ   1 bmi1 21.3443766 0.02537514
5  <NA> MZ.expMeanMZ   1 bmi2 21.3490128 0.02655154
....
10 <NA> DZ.expMeanDZ   1 bmi2 21.4578451 0.03459388

Observed statistics:  1777
Estimated parameters:  10
Degrees of freedom:  1767
-2 log likelihood:  4055.935
numObs:  920
AIC (Mx):  521.9346
BIC (Mx):  -4001.367
```

Free Parameters

>univTwinSatFit$MZ@matrices

LowerMatrix 'CholMZ'

@values
[,1] [,2]
[1,] 0.8530361 0.0000000
[2,] 0.6935690 0.5587162

@free
[,1] [,2]
[1,] TRUE FALSE
[2,] TRUE TRUE

FullMatrix 'expMeanMZ'

@values
bmi1 bmi2
[1,] 21.34438 21.34901

@free
bmi1 bmi2
[1,] TRUE TRUE

Derived Quantities

>univTwinSatFit$MZ@algebras

mxAlgebra 'expCovMZ'
@formula: CholMZ %*% t(CholMZ)
@result:
	bmi1	bmi2
bmi1	0.7276707	0.5916394
bmi2	0.5916394	0.7932017

mxAlgebra 'expMeanMZt1'
@formula: expMeanMZ[1, 1:nv]
@result:
	[,1]
[1,]	21.34438

mxAlgebra 'expMeanMZt2'
@formula: expMeanMZ[1, (nv + 1):ntv]
@result:
	[,1]
[1,]	21.34901

mxAlgebra 'expVarMZ'
@formula: t(diag2vec(expCovMZ))
@result:
	[,1]	[,2]
[1,]	0.7276707	0.7932017

mxAlgebra 'expVarMZt1'
@formula: expVarMZ[1, 1:nv]
@result:
	[,1]
[1,]	0.7276707

mxAlgebra 'expVarMZt2'
@formula: expVarMZ[1, (nv + 1):ntv]
@result:
	[,1]
[1,]	0.7932017

Extract Information

univTwinSatFit	all information in MxModel fitted object
univTwinSatFit@algebras univTwinSatFit@algebras$'-2sumll'	list of algebras of container model specific algebra
univTwinSatFit@submodels univTwinSatFit@submodels$MZ =univTwinSatFit$MZ	list of all items in all child models list of all items in specific child model
univTwinSatFit$MZ@matrices univTwinSatFit$MZ@matrices$CholMZ =univTwinSatFitMZCholMZ	list of all matrices in specific child model specific matrix in specific child model
univTwinSatFit$MZ@algebras univTwinSatFit$MZ@algebras$expCovMZ =univTwinSatFitMZexpCovMZ	list of all algebras in specific child model specific algebra in specific child model
univTwinSatFitMZobjective univTwinSatFitMZdata	objective of specific child model data of specific child model

Estimated Values

		T1	T2		T1	T2
Saturated Model						
mean	MZ			DZ		
cov	T1			T1		
	T2			T2		

Results

		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

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
Sat							

os	observed statistics	
ep	estimated parameters	
-2ll	-2 LogLikelihood	
df	degrees of freedom	os - ep
AIC	Akaike's Information Criterion	-2ll -2df
BIC	Bayesian Information Criterion	5(-2ll-df*ln(N)
diff -2ll	likelihood ratio Chi-square	
diff df	difference in degrees of freedom	

GoF Results

	ep	-2ll	df	AIC	diff -2ll	diff df	p
Sat	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
BIC	Bayesian Information Criterion	5(-2ll-df*ln(N)
diff -2ll	likelihood ratio Chi-square	
diff df	difference in degrees of freedom	

Helper Functions

UnivariateTwinSaturated_MatrixRawCon.R [6]

....

```
# Generate Saturated Output
# ------------------------------------------------------------------------------
parameterSpecifications(univTwinSatFit)
expectedMeansCovariances(univTwinSatFit)
tableFitStatistics(univTwinSatFit)
```

Parameter Specifications

```
> parameterSpecifications(univTwinSatFit)

model:MZ, matrix:CholMZ
     [,1] [,2]
[1,] [NA] 0
[2,] [NA] [NA]

model:DZ, matrix:CholDZ
     [,1] [,2]
[1,] [NA] 0
[2,] [NA] [NA]

model:MZ, matrix:expMeanMZ
     bmi1 bmi2
[1,] [NA] [NA]

model:DZ, matrix:expMeanDZ
     bmi1 bmi2
[1,] [NA] [NA]
```

free parameters in square brackets with labels
fix parameters with values

Expected Values

```
> expectedMeansCovariances(univTwinSatFit)

model:MZ, covariance:expCovMZ
          bmi1      bmi2
bmi1 0.7276707 0.5916394
bmi2 0.5916394 0.7932017

model:DZ, covariance:expCovDZ
          bmi1      bmi2
bmi1 0.7691893 0.2400395
bmi2 0.2400395 0.8216317

model:MZ, means:expMeanMZ
          bmi1      bmi2
[1,] 21.34438 21.34901

model:DZ, means:expMeanDZ
          bmi1      bmi2
[1,] 21.44752 21.45785
```

expected covariances, means and thresholds
for each of the mxModels with an objective function

Fit Statistics

```
> tableFitStatistics(univTwinSatFit)

                Name          ep      -2LL      df      AIC
Model 1 : univTwinSat 10 4055.93 1767 521.93
```

ep: # of estimated parameters
-2LL: -2 LogLikelihood of data
df: degrees of freedom
AIC: Akaike's Information Criterion

Fitting Nested Models

- Saturated Model
  - likelihood of data without any constraints
  - fitting as many means and (co)variances as possible
- Equality of means & variances by twin order
  - test if mean of twin 1 = mean of twin 2
  - test if variance of twin 1 = variance of twin 2
- Equality of means & variances by zygosity
  - test if mean of MZ = mean of DZ
  - test if variance of MZ = variance of DZ

Equate Means & Variances

UnivariateTwinSaturated_MatrixRawCon.R [7]

```
# Fit Model with Equal Means & Variances across Twin Order and Zygosity
# ------------------------------------------------------------------------------
# Constrain expected means and variances to be equal across twin order
equateMeansVarsTwinModel <- mxModel(univTwinSatFit, name="equateMeansVarsTwin",
    mxConstraint( alg1="MZ.expVarMZt1", "=", alg2="MZ.expVarMZt2", name="VarMZt1=t2"),
    mxConstraint( alg1="DZ.expVarDZt1", "=", alg2="DZ.expVarDZt2", name="VarDZt1=t2"),
    mxConstraint( alg1="MZ.expMeanMZt1", "=", alg2="MZ.expMeanMZt2", name="MeanMZt1=t2"),
    mxConstraint( alg1="DZ.expMeanDZt1", "=", alg2="DZ.expMeanDZt2", name="MeanDZt1=t2")
)
equateMeansVarsTwinFit <- mxRun(equateMeansVarsTwinModel)
equateMeansVarsTwinSumm <- summary(equateMeansVarsTwinFit)
equateMeansVarsTwinSumm

parameterSpecifications(equateMeansVarsTwinFit)
expectedMeansCovariances(equateMeansVarsTwinFit)
tableFitStatistics(univTwinSatFit, equateMeansVarsTwinFit)
```

Twin Order & Zygosity

UnivariateTwinSaturated_MatrixRawCon.R [8]

```
# Constrain expected means and variances to be equal across twin order and zygosity
equateMeansVarsTwinZygModel <- mxModel(equateMeansVarsTwinFit,
name="equateMeansVarsTwinZyg",
    mxConstraint( alg1="MZ.expVarMZt1", "=", alg2="DZ.expVarDZt1", name="VarMZ=DZ"),
    mxConstraint( alg1="MZ.expMeanMZt1", "=", alg2="DZ.expMeanDZt1", name="MeanMZ=DZ")
)
equateMeansVarsTwinZygFit <- mxRun(equateMeansVarsTwinZygModel)
equateMeansVarsTwinZygSumm <- summary(equateMeansVarsTwinZygFit)
equateMeansVarsTwinZygSumm

parameterSpecifications(equateMeansVarsTwinZygFit)
expectedMeansCovariances(equateMeansVarsTwinZygFit)
tableFitStatistics(univTwinSatFit, list(equateMeansVarsTwinFit, equateMeansVarsTwinZygFit))
```

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
Sat							
Twin t1=t2							
Zyg MZ=DZ							

Estimates

		T1	T2		T1	T2
Equate Means & Variances across Twin Order						
mean	MZ	21.35	21.35	DZ	21.45	21.45
cov	T1	0.76		T1	0.79	
	T2	0.59	0.76	T2	0.24	0.79
Equate Means Variances across Twin Order & Zygosity						
mean	MZ	21.39	21.39	DZ	21.39	21.39
cov	T1	0.78		T1	0.78	
	T2	0.61	0.78	T2	0.23	0.78

Stats

	ep	-2ll	df	AIC	diff -2ll	diff df	P
Sat	10	4055.93	1767	521.93			
Twin t1=t2	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

- Genes? Environment?

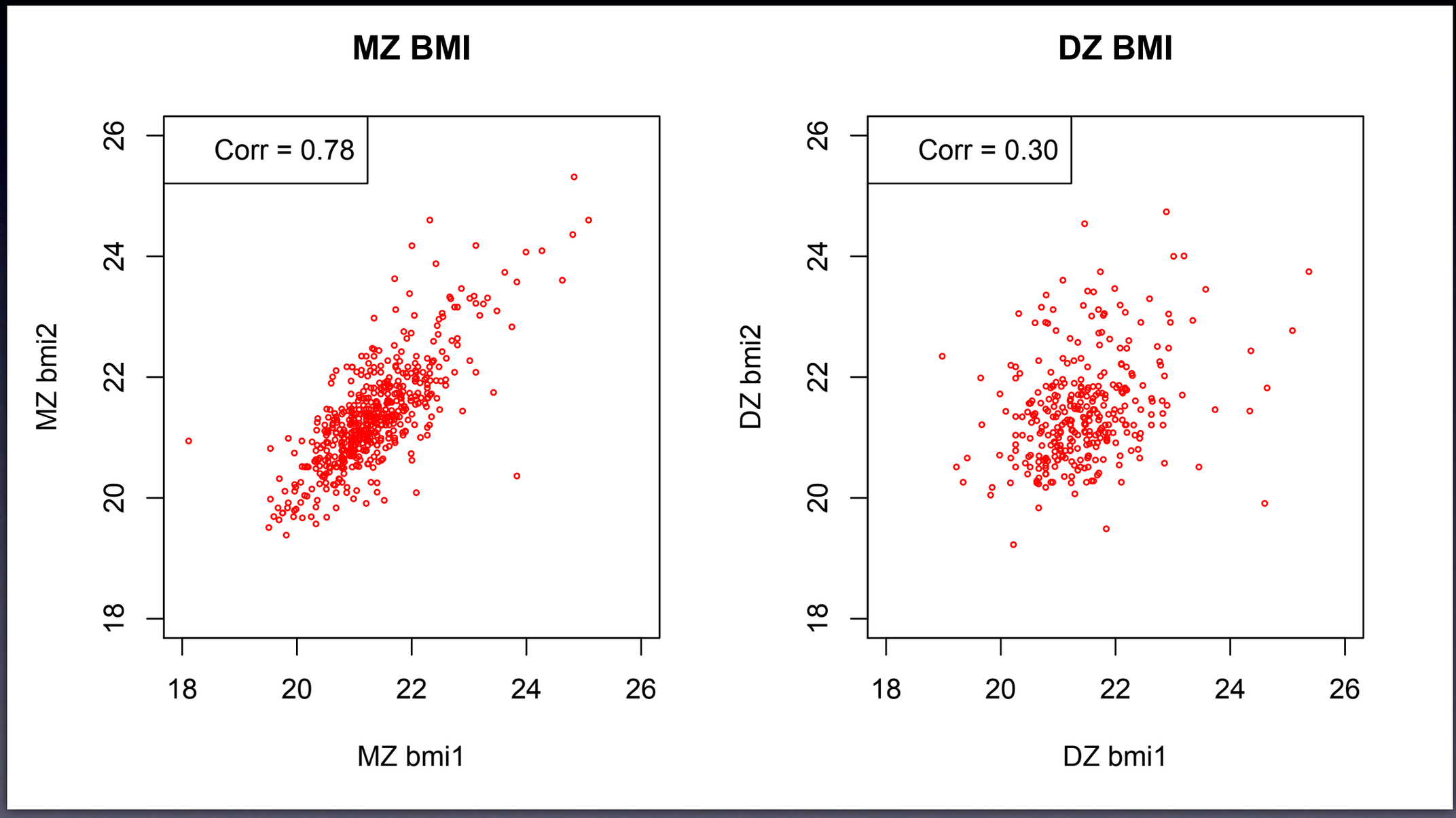
Univariate Twin Continuous

UnivariateTwinAnalysis_MatrixRawCon.R [1]

```
# Prepare Data
# ------------------------------------------------------------------------
data(twinData)
summary(twinData)
Vars <- 'bmi'
nv <- 1
selVars <- paste(Vars,c(rep(1,nv),rep(2,nv)),sep="")
ntv <- nv*2
mzData <- subset(twinData, zyg==1, selVars)
dzData <- subset(twinData, zyg==3, selVars)

# Print Descriptive Statistics
# ------------------------------------------------------------------------
summary(mzData)
colMeans(mzData,na.rm=TRUE)
cov(mzData,use="complete")
summary(dzData)
colMeans(dzData,na.rm=TRUE)
cov(dzData,use="complete")
```

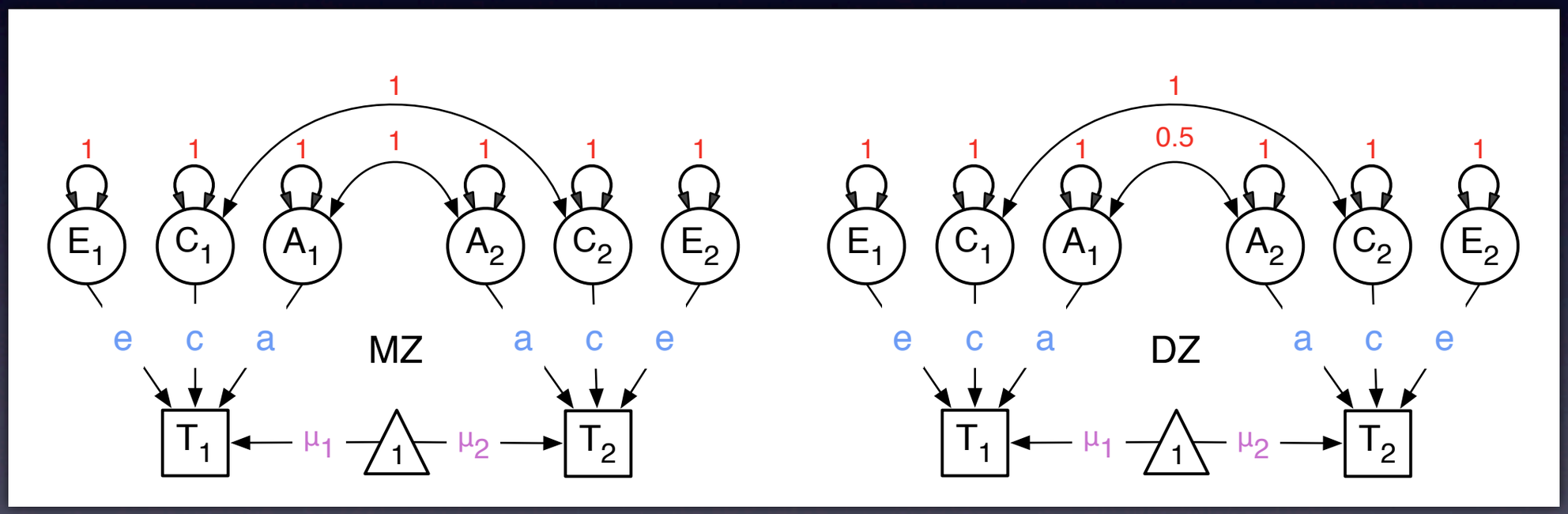
Twin Correlations



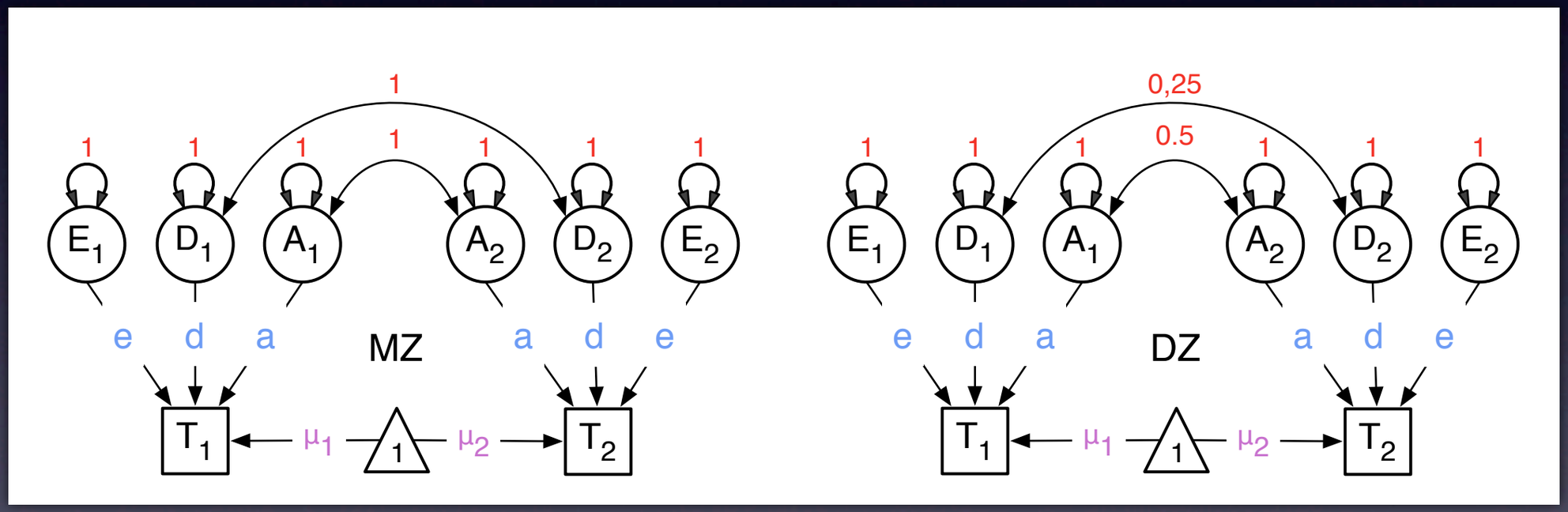
Twin Correlations ~ Sources of Variance

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

Univariate ACE Model



Univariate ADE Model



ADE

UnivariateTwinAnalysis_MatrixRawConADE.R [2]

```
# Fit ADE Model with RawData and Matrices Input
# ------------------------------------------------------------------------------
univADEModel <- mxModel("univADE",
      mxModel("ADE",
      # Matrices a, d, and e to store a, d, and e path coefficients
            mxMatrix( "Lower", nv, nv, free=TRUE, values=.6, label="a11", name="a" ),
            mxMatrix( "Lower", nv, nv, free=TRUE, values=.6, label="c11", name="d" ),
            mxMatrix( "Lower", nv, nv, free=TRUE, values=.6, label="e11", name="e" ),
      # Matrices A, D, and E compute variance components
            mxAlgebra( a %*% t(a), name="A" ),
            mxAlgebra( d %*% t(d), name="D" ),
            mxAlgebra( e %*% t(e), name="E" ),
      # Algebra to compute total variances and inverse of standard deviations (diagonal only)
            mxAlgebra( A+D+E, name="V" ),
            mxMatrix( "Iden", nv, nv, name="I"),
            mxAlgebra( solve(sqrt(I*V)), name="iSD"),
```

“ADE”,“MZ”,“DZ”

UnivariateTwinAnalysis_MatrixRawConADE.R [3]

```
## Note that following mxModel statements do not change for bivariate/multivariate case
    # Matrix & Algebra for expected means vector
        mxMatrix( "Full", 1, nv, free=TRUE, values= 20, label="mean", name="Mean" ),
        mxAlgebra(  cbind(Mean,Mean), name="expMean"),
    # Algebra for expected variance/covariance matrix in MZ
        mxAlgebra(  rbind  ( cbind(A+D+E , A+C),
                             cbind(A+C   , A+D+E)), name="expCovMZ" ),
    # Algebra for expected variance/covariance matrix in DZ; 0.5, converted to 1*1 matrix
        mxAlgebra(  rbind  ( cbind(A+D+E     , 0.5%x%A + 0.25%x%D),
                             cbind(0.5%x%A + 0.25%x%D , A+D+E)),  name="expCovDZ" ) ),
    mxModel("MZ",
        mxData( observed=mzData, "raw" ),
        mxFIMLObjective( "ADE.expCovMZ", "ADE.expMean", dimnames=selVars )     ),
    mxModel("DZ",
        mxData( observed=dzData, "raw" ),
        mxFIMLObjective( "ADE.expCovDZ", "ADE.expMean", dimnames=selVars )     ),
    mxAlgebra( MZ.objective + DZ.objective, name="-2sumll" ),
    mxAlgebraObjective("-2sumll")
)
```

Run

UnivariateTwinAnalysis_MatrixRawConADE.R [4]

```
univADEFit <- mxRun(univADEModel)
univADESumm <- summary(univADEFit)

# Generate ADE Output
# ------------------------------------------------------------------
parameterSpecifications(univADEFit)
expectedMeansCovariances(univADEFit)
tableFitStatistics(univADEFit)

# Generate Table of Parameter Estimates using mxEval
pathEstimatesADE <- print(round(mxEval(cbind(ADE.a,ADE.c,ADE.e), univADEFit),4))
varComponentsADE <- print(round(mxEval(cbind(ADE.A/ADE.V,ADE.C/ADE.V,
  ADE.E/ADE.V), univADEFit),4))
      rownames(pathEstimatesADE) <- 'pathEstimates'
      colnames(pathEstimatesADE) <- c('a','d','e')
      rownames(varComponentsADE) <- 'varComponents'
      colnames(varComponentsADE) <- c('a^2','d^2','e^2')
pathEstimatesADE
varComponentsADE
```

summary

```
> univADESumm
  name   matrix row col   Estimate  Std.Error
1  a11    ADE.a   1   1  0.5033079 0.11410744
2  d11    ADE.d   1   1  0.5304309 0.10629047
3  e11    ADE.e   1   1  0.3706477 0.01168032
4 mean ADE.Mean   1   1 21.6948896 0.02323177
```

Observed statistics: 910
Estimated parameters: 4
Degrees of freedom: 906
-2 log likelihood: 1952.722
Saturated -2 log likelihood: NA
numObs: 479
Chi-Square: NA
p: NA
AIC (Mx): 140.7223
BIC (Mx): -1819.419

Specifications

```
> parameterSpecifications(univADEFit)
model:ADE, matrix:a
     [,1]
[1,] [a11]

model:ADE, matrix:d
     [,1]
[1,] [d11]

model:ADE, matrix:e
     [,1]
[1,] [e11]

model:ADE, matrix:Mean
     [,1]
[1,] [mean]
```

Expected Values

```
> expectedMeansCovariances(univADEFit)
model:MZ, covariance:ADE.expCovMZ
          bmi1      bmi2
bmi1 0.6720555 0.5346758
bmi2 0.5346758 0.6720555

model:MZ, means:ADE.expMean
          bmi1     bmi2
[1,] 21.69489 21.69489

model:DZ, covariance:ADE.expCovDZ
          bmi1      bmi2
bmi1 0.6720555 0.1969987
bmi2 0.1969987 0.6720555

model:DZ, means:ADE.expMean
          bmi1     bmi2
[1,] 21.69489 21.69489
```

Table of Fit Statistics

```
> tableFitStatistics(univADEFit)
                Name    ep      -2LL    df      AIC
Model 1 : univADE 4  1952.72 906 140.72
```

Table of Estimates

```
> # Generate Table of Parameter Estimates using mxEval
> pathEstimatesADE <- mxEval(cbind(ADE.a,ADE.d,ADE.e), univADEFit)
>     rownames(pathEstimatesADE) <- 'pathEstimates'
>     colnames(pathEstimatesADE) <- c('a','d','e')
> pathEstimatesADE
                      a         d         e
pathEstimates 0.5033079 0.5304309 0.3706477
>

> varComponentsADE <- mxEval(cbind(ADE.A/ADE.V,ADE.D/ADE.V,ADE.E/ADE.V),
 univADEFit)
>     rownames(varComponentsADE) <- 'varComponents'
>     colnames(varComponentsADE) <- c('a^2','d^2','e^2')
> varComponentsADE
                          a^2       d^2       e^2
varComponents 0.3769315 0.4186513 0.2044172
```

List of Matrices & Labels

UnivariateTwinAnalysis_MatrixRawConADE.R [5]

```
# Generate List of Parameter Estimates and Derived Quantities using formatOutputMatrices
ADEpathMatrices <- c("ADE.a","ADE.d","ADE.e",
 "ADE.iSD","ADE.iSD %*% ADE.a","ADE.iSD %*% ADE.d","ADE.iSD %*% ADE.e")
ADEpathLabels <- c("pathEst_a","pathEst_d","pathEst_e",
  "iSD","stParEst_a","stPathEst_d","stPathEst_e")
formatOutputMatrices(univADEFit,ADEpathMatrices,ADEpathLabels,4)
```

```
ADEcovMatrices <- c("ADE.A","ADE.D","ADE.E",
 "ADE.V","ADE.A/ADE.V","ADE.D/ADE.V","ADE.E/ADE.V")
ADEcovLabels <- c("covComp_A","covComp_D","covComp_E",
 "Var","stCovComp_A","stCovComp_D","stCovComp_E")
formatOutputMatrices(univADEFit,ADEcovMatrices,ADEcovLabels,4)
```

List of Path Estimates

```
> formatOutputMatrices(univADEFit,ADEpathMatrices,ADEpathLabels,Vars,4)

[1] "Matrix ADE.a"
    pathEst_a1
bmi 0.5033

[1] "Matrix ADE.d"
    pathEst_d1
bmi 0.5304

[1] "Matrix ADE.e"
    pathEst_e1
bmi 0.3706

[1] "Matrix ADE.iSD"
    iSD1
bmi 1.2198

[1] "Matrix ADE.iSD %*% ADE.a"
    stPathEst_a1
bmi 0.6139

[1] "Matrix ADE.iSD %*% ADE.d"
    stPathEst_d1
bmi 0.6470

[1] "Matrix ADE.iSD %*% ADE.e"
    stPathEst_e1
bmi 0.4521
```

List of Variance Components

```
> formatOutputMatrices(univADEFit,ADEcovMatrices,ADEcovLabels,Vars,4)
```

```
[1] "Matrix ADE.A"
    covComp_A1
bmi 0.2533
```

```
[1] "Matrix ADE.A/ADE.V"
    stCovComp_A1
bmi 0.3769
```

```
[1] "Matrix ADE.D"
    covComp_D1
bmi 0.2814
```

```
[1] "Matrix ADE.D/ADE.V"
    stCovComp_D1
bmi 0.4187
```

```
[1] "Matrix ADE.E"
    covComp_E1
bmi 0.1374
```

```
[1] "Matrix ADE.E/ADE.V"
    stCovComp_E1
bmi 0.2044
```

```
[1] "Matrix ADE.V"
    Var1
bmi 0.6721
```

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

AE - E

UnivariateTwinAnalysis_MatrixRawConADE.R [6]

```
# Fit AE model
# ------------------------------------------------------------------------------
univAEModel <- mxModel(univADEFit, name="univAE",
    mxModel(univADEFit$ADE,
        mxMatrix( "Full", 1, 1, free=FALSE, values=0, label="d11", name="d" ) # drop d at 0
    )
)
univAEFit <- mxRun(univAEModel)
univAESumm <- summary(univAEFit)

# Fit E model
# ------------------------------------------------------------------------------
univEModel <- mxModel(univAEFit, name="univE",
    mxModel(univAEFit$ADE,
        mxMatrix( "Full", 1, 1, free=FALSE, values=0, label="a11", name="a" ) # drop a at 0
    )
)
univEFit <- mxRun(univEModel)
univESumm <- summary(univEFit)
```

Comparative Fit

UnivariateTwinAnalysis_MatrixRawConADE.R [7]

```
# Print Comparative Fit Statistics
# ------------------------------------------------------------------------
univADENested <- list(univAEFit, univEFit)
tableFitStatistics(univADEFit,univADENested)
```

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
ADE							

os	observed statistics	
ep	estimated parameters	
-2ll	-2 LogLikelihood	
df	degrees of freedom	os - ep
AIC	Akaike's Information Criterion	-2ll -2df
BIC	Bayesian Information Criterion	5(-2ll-df*ln(N)
diff -2ll	likelihood ratio Chi-square	
diff df	difference in degrees of freedom	

Goodness-of-Fit Statistics

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

Estimated Values

	a	d	e	a^2	d^2	e^2
ADE						
AE						
E						

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
ADE	4	4063.45	1773	517.45	-		
AE	3	4067.66	1774	519.66	4.21	1	0.06
E	2	4591.79	1775	1041.79	528.3	2	0.00

Estimated Values

	a	d	e	a^2	d^2	e^2
ADE	0.57	0.54	0.41	0.41	0.37	0.22
AE	0.78	-	0.42	0.78	-	0.22
E	-	-	0.88	-	-	1.00

ACE

UnivariateTwinAnalysis_MatrixRawConACE.R [2]

```
# Fit ACE Model with RawData and Matrices Input
# ------------------------------------------------------------------------------
univACEModel <- mxModel("univACE",
      mxModel("ACE",
      # Matrices a, c, and e to store a, c, and e path coefficients
            mxMatrix( "Lower", nv, nv, free=TRUE, values=.6, label="a11", name="a" ),
            mxMatrix( "Lower", nv, nv, free=TRUE, values=.6, label="c11", name="c" ),
            mxMatrix( "Lower", nv, nv, free=TRUE, values=.6, label="e11", name="e" ),
      # Matrices A, C, and E compute variance components
            mxAlgebra( a %*% t(a), name="A" ),
            mxAlgebra( c %*% t(c), name="C" ),
            mxAlgebra( e %*% t(e), name="E" ),
      # Algebra to compute total variances and inverse of standard deviations (diagonal only)
            mxAlgebra( A+C+E, name="V" ),
            mxMatrix( "Iden", nv, nv, name="I"),
            mxAlgebra( solve(sqrt(I*V)), name="iSD"),
```

“ACE”, “MZ”, “DZ”

UnivariateTwinAnalysis_MatrixRawConACE.R [3]

```
## Note that following mxModel statements do not change for bivariate/multivariate case
    # Matrix & Algebra for expected means vector
        mxMatrix( "Full", 1, nv, free=TRUE, values= 20, label="mean", name="M" ),
        mxAlgebra(  cbind(M,M), name="expMean"),
    # Algebra for expected variance/covariance matrix in MZ
        mxAlgebra(  rbind  ( cbind(A+C+E , A+C),
                             cbind(A+C   , A+C+E)), name="expCovMZ" ),
    # Algebra for expected variance/covariance matrix in DZ; 0.5, converted to 1*1 matrix
        mxAlgebra(  rbind  ( cbind(A+C+E     , 0.5%x%A+C),
                             cbind(0.5%x%A+C , A+C+E)),  name="expCovDZ" ) ),
    mxModel("MZ",
        mxData( observed=mzData, "raw" ),
        mxFIMLObjective( "ACE.expCovMZ", "ACE.expMean", dimnames=selVars )     ),
    mxModel("DZ",
        mxData( observed=dzData, "raw" ),
        mxFIMLObjective( "ACE.expCovDZ", "ACE.expMean", dimnames=selVars )     ),
    mxAlgebra( MZ.objective + DZ.objective, name="-2sumll" ),
    mxAlgebraObjective("-2sumll")
)
```

Table of Estimates

UnivariateTwinAnalysis_MatrixRawConACE.R [4]

```
univACEFit <- mxRun(univACEModel)
univACESumm <- summary(univACEFit)

# Generate ACE Output
# ------------------------------------------------------------------------------
parameterSpecifications(univACEFit)
expectedMeansCovariances(univACEFit)
tableFitStatistics(univACEFit)

# Generate Table of Parameter Estimates using mxEval
pathEstimatesACE <- print(round(mxEval(cbind(ACE.a,ACE.c,ACE.e), univACEFit),4))
varComponentsACE <- print(round(mxEval(cbind(ACE.A/ACE.V,ACE.C/ACE.V,
  ACE.E/ACE.V), univACEFit),4))
      rownames(pathEstimatesACE) <- 'pathEstimates'
      colnames(pathEstimatesACE) <- c('a','c','e')
      rownames(varComponentsACE) <- 'varComponents'
      colnames(varComponentsACE) <- c('a^2','c^2','e^2')
pathEstimatesACE
varComponentsACE
```

List of Matrices & Labels

UnivariateTwinAnalysis_MatrixRawConACE.R [5]

```
# Generate List of Parameter Estimates and Derived Quantities using formatOutputMatrices
ACEpathMatrices <- c("ACE.a","ACE.c","ACE.e",
 "ACE.iSD","ACE.iSD %*% ACE.a","ACE.iSD %*% ACE.c","ACE.iSD %*% ACE.e")
ACEpathLabels <- c("pathEst_a","pathEst_c","pathEst_e",
  "iSD","stParEst_a","stPathEst_c","stPathEst_e")
formatOutputMatrices(univACEFit,ACEpathMatrices,ACEpathLabels,4)
```

```
ACEcovMatrices <- c("ACE.A","ACE.C","ACE.E",
 "ACE.V","ACE.A/ACE.V","ACE.C/ACE.V","ACE.E/ACE.V")
ACEcovLabels <- c("covComp_A","covComp_C","covComp_E",
 "Var","stCovComp_A","stCovComp_C","stCovComp_E")
formatOutputMatrices(univACEFit,ACEcovMatrices,ACEcovLabels,4)
```

Nested Models

- ‘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

AE - CE

UnivariateTwinAnalysis_MatrixRawConACE.R [6]

```
# Fit AE model
# ------------------------------------------------------------------------------
univAEModel <- mxModel(univACEFit, name="univAE",
    mxModel(univACEFit$ACE,
        mxMatrix( "Full", 1, 1, free=FALSE, values=0, label="c11", name="c" ) # drop c at 0
    )
)
univAEFit <- mxRun(univAEModel)
univAESumm <- summary(univAEFit)

# Fit CE model
# ------------------------------------------------------------------------------
univCEModel <- mxModel(univACEFit, name="univCE",
    mxModel(univACEFit$ACE,
        mxMatrix( "Full", 1, 1, free=FALSE, values=0, label="a11", name="a" ) # drop a at 0
    )
)
univCEFit <- mxRun(univCEModel)
univCESumm <- summary(univCEFit)
```

E, Comparative Fit

UnivariateTwinAnalysis_MatrixRawConACE.R [7]

```
# Fit E model
# ------------------------------------------------------------------------------
univEModel <- mxModel(univAEFit, name="univE",
    mxModel(univAEFit$ACE,
        mxMatrix( "Full", 1, 1, free=FALSE, values=0, label="a11", name="a" ) # drop a at 0
    )
univEFit <- mxRun(univEModel)
univESumm <- summary(univEFit)

# Print Comparative Fit Statistics
# ------------------------------------------------------------------------------
univACENested <- list(univAEFit, univCEFit, univEFit)
tableFitStatistics(univACEFit,univACENested)
```

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^2	d^2	e^2	c^2
ADE								
AE								
ACE								
AE								
E								

Goodness-of-Fit Statistics

	ep	-2ll	df	AIC	diff -2ll	diff df	p
ADE	4	4063.45	1773	517.45	-		
AE	3	4067.66	1774	519.66	4.21	1	0.06
ACE	4	4067.66	1773	521.66	-		
CE	3	4220.31	1774	672.31	152.6	1	0.00
E	2	4591.79	1775	1041.79	528.3	2	0.00

Estimated Values

	a	d	e	c	a^2	d^2	e^2	c^2
ADE	0.57	0.54	0.41	-	0.41	0.37	0.22	-
AE	0.78	-	0.42	-	0.78	-	0.22	-
ACE	0.78	0.00	0.42	-	0.78	0.00	0.22	-
AE	-	-	0.56	0.68	-	-	0.41	0.59
E	-	-	0.88	-	-	-	1.00	-

Conclusions

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