Introduction to Analysis of Twin Data Using R and OpenMx- Part 1

Elizabeth Prom-Wormley

Special Thanks to Sarah Medland

Files for this video can be found at:
/home/elizabeth/2022/

Copy 3 files to your server folder:
IntroToRAndOpenMxMay2022.R
SimWtDataInd.csv
SimWtDataPair.csv

Session Objectives

By the end of the session, you will be able to

- Recognize the major steps involved in an OpenMx model
- Translate implementation of a linear regression between a statistical equation, structural equation model, and an OpenMx model

What is OpenMx?

- Free, Open-source software package for use in R

- Estimation of advanced multivariate statistical models, particularly structural equation modelling

- Runs on Windows, Mac OSX, and Linux/GNU

- Two main approaches to writing OpenMx models
  - Path or Matrix Specification

How Does OpenMx Work?

- OpenMx uses functions to build objects
- Arguments to the function have an order
- Order can be changed by naming arguments

Data Preparation Considerations

- The algebra style used in OpenMx expects 1 line per case/family
- (Almost) limitless number of families and variables
- Data needs to be read into R before it can be analyzed
  - (the commands to read the data can be nested within the R script)
- Default missing code is NA

Matrix Algebra Basics and Its Application in OpenMx

Matrix: A rectangular array of elements arranged in rows and columns

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

ROWS

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

COLUMNS

Order or dimension of a matrix is defined by the number of ROWS then COLUMNS in the matrix.

Matrix A is a 3 x 2 matrix.

Each element in the matrix is referred to by its placement in a row and column, where a_{ij} is the element in Matrix A in the i_{th} row and j_{th} column.

In Matrix A, the number 4 is element a(1,2)

OpenMx Matrices

```
mxMatrix( type=“Zero", nrow=2, ncol=3, name="a" )
0 0 0
0 0 0

mxMatrix( type=“Unit", nrow=2, ncol=3, name="a" )
1 1 1
1 1 1

mxMatrix( type=“Ident", nrow=3, ncol=3, name="a" )
1 0 0
0 1 0
0 0 1
```

```
mxMatrix( type=“Diag", nrow=3, ncol=3, free=TRUE, name="a" )
? 0 0
0 ? 0
0 0 ?

mxMatrix( type=“Sdiag", nrow=3, ncol=3, free=TRUE, name="a" )
0 0 0
? 0 0
? ? 0

mxMatrix( type=“Stand", nrow=3, ncol=3, free=TRUE, name="a" )
1 ? ?
? 1 ?
? ? 1

mxMatrix( type=“Symm", nrow=3, ncol=3, free=TRUE, name="a" )
? ? ?
? ? ?
? ? ?

mxMatrix( type=“Lower", nrow=3, ncol=3, free=TRUE, name="a" )
? 0 0
? ? 0
? ? ?

mxMatrix( type=“Full", nrow=2, ncol=4, free=TRUE, name="a" )
? ? ? ?
? ? ? ?
```

Matrix Operations

Matrix Addition and Subtraction:

- Matrices must be the same size

$$\begin{pmatrix} 2 & 1 \\ 3 & 5 \\ 6 & 2 \end{pmatrix} + \begin{pmatrix} 4 & 8 \\ 7 & 2 \\ 9 & 6 \end{pmatrix} = \begin{pmatrix} 2+4 & 1+8 \\ 3+7 & 5+2 \\ 6+9 & 2+6 \end{pmatrix} = \begin{pmatrix} 6 & 9 \\ 10 & 7 \\ 15 & 8 \end{pmatrix}$$

If the matrices are of different orders, it is impossible to add them

$$\begin{pmatrix} 2 & 1 \\ 3 & 5 \\ 6 & 2 \end{pmatrix} + \begin{pmatrix} 4 & 7 & 9 \\ 8 & 2 & 6 \end{pmatrix} = \text{Undefined}$$

Dot Product

Also known as the element-wise product

OpenMx symbol *

$$A \circ B = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} . \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} & A_{12}B_{12} & A_{13}B_{13} \\ A_{21}B_{21} & A_{22}B_{22} & A_{23}B_{23} \\ A_{31}B_{31} & A_{32}B_{32} & A_{33}B_{33} \end{pmatrix}$$

Matrix Multiplication (Star product)

Number of columns of the first matrix must equal the number of rows of the second matrix.

Product will have as many rows as the first matrix and as many columns as the second matrix.

OpenMx symbol %*%

C = A x B

$$C = \begin{pmatrix} 3 & 4 & 7 \\ 5 & 6 & 1 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 5 \\ 6 & 2 \end{pmatrix}$$

$$= \begin{pmatrix} 3*2 + 4*3 + 7*6 & 3*1 + 4*5 + 7*2 \\ 5*2 + 6*3 + 1*6 & 5*1 + 6*5 + 1*2 \end{pmatrix} = \begin{pmatrix} 60 & 37 \\ 34 & 37 \end{pmatrix}$$

Kroneker Product

OpenMx symbol %x%

$$A \otimes B = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}a_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Quadratic Product

- The quadratic product is extremely useful in statistical analysis (particularly in Structural Equation Modeling)
- OpenMx symbol %&%

$$A\%\&\%B = ABA^T =$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{x} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \text{x} \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}$$

The Challenge

- You have been handed a dataset and you have been asked to consider the degree to which Study Site influences Weight in Twin 1 members of your sample
  - Correlation
  - Regression
  - Structural Equation Modeling (via OpenMx)

Linear Regression in R Using lm()

- Open this file:

/home/elizabeth/2022/IntroToOpenMx/IntroToOpenMxMay2022.R

```
WeightFit_T1 <- lm(WT_T1 ~ Site_T1, data=Twins2)
summary(WeightFit_T1)
coefficients(WeightFit_T1)
```

- Intercept = 51.713723
- Beta1 = 0.294491
- Variance = residual standard error^2

  = 1.0052^2

  = 1.010427

Traditional Linear Regression

$$\text{Weight}_i = \beta_0 + \beta_1 * \text{Sitei} + \varepsilon_i$$

$$y_1 = \beta_0 + \beta_1 \mathrm{x}_1 + \varepsilon_1$$

$$y_2 = \beta_0 + \beta_1 \mathrm{x}_2 + \varepsilon_2$$

$$y_3 = \beta_0 + \beta_1 \mathrm{x}_3 + \varepsilon_3$$

$$\vdots$$

$$y_i = \beta_0 + \beta_1 \mathrm{x}_\mathrm{i} + \varepsilon_i$$

$$y = \beta X + \varepsilon$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \end{bmatrix} = \begin{bmatrix} \beta 0 \\ \beta 1 \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_i \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_i \end{bmatrix}$$

- 4 matrices
  - Weight- Observed variable
  - SITE- Observed variable
  - Beta
  - Epsilon

- 3 estimated parameters
  - β_0: Intercept
  - β_1: Regression of Weight on SITE
  - σ^2_{weight} : Error variance

$$\mathrm{Y} = \beta X + \varepsilon$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \end{bmatrix} = [\beta 0\ \beta 1\ ] \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & x_i \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_i \end{bmatrix}$$

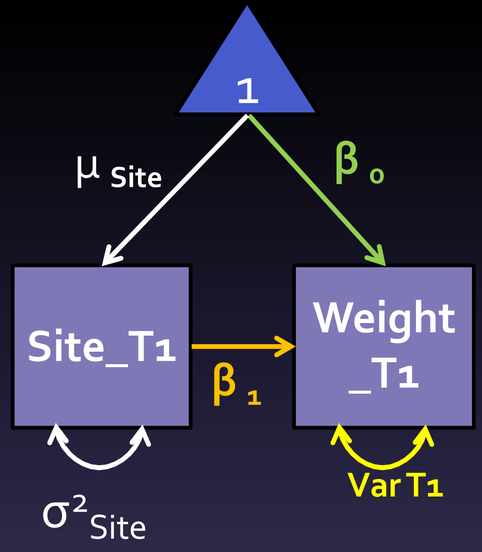
- Y = n x 1 column vector
- X = n x 2 matrix
- β = 2 x 1 column vector
- ε = n x 1 column vector

- βX is calculated using matrix multiplication
- βX + ε is calculated using matrix addition

Linear Regression in OpenMx

- Download data and code from /home/elizabeth/2022/IntroToOpenMxMay2022.R
- Intercept = 51.71372117
- Beta1 = 0.29449237
- Variance = 1.01047002

Regression Across All Twin 1 Participants




```
depVar <- ‘Weight_T1‘

# Variance/Covariance matrix
eVar     <- mxMatrix( type="Full", nrow=1, ncol=1, free=TRUE,
                 values=10, labels=‘varT1', name="residualVar" )

# Regression betas
b0       <- mxMatrix(type="Full", nrow=1, ncol=1, free=TRUE,
                 values=30, labels="beta0T1", name="Intercept" )

b1       <- mxMatrix(type="Full", nrow=1, ncol=1, free=TRUE,
                 values=0, labels="beta1T1", name="bSite" )

# Independent variable
x      <- mxMatrix(type="Full", nrow=1, ncol=1, free=FALSE,
                 labels="data.Site_T1", name=”Site" )
```

1x1 matrix
name “residualVar”

varT1

1x1 matrix
name “Intercept”

beta0T1

1x1 matrix
name “bSite”

beta1T1

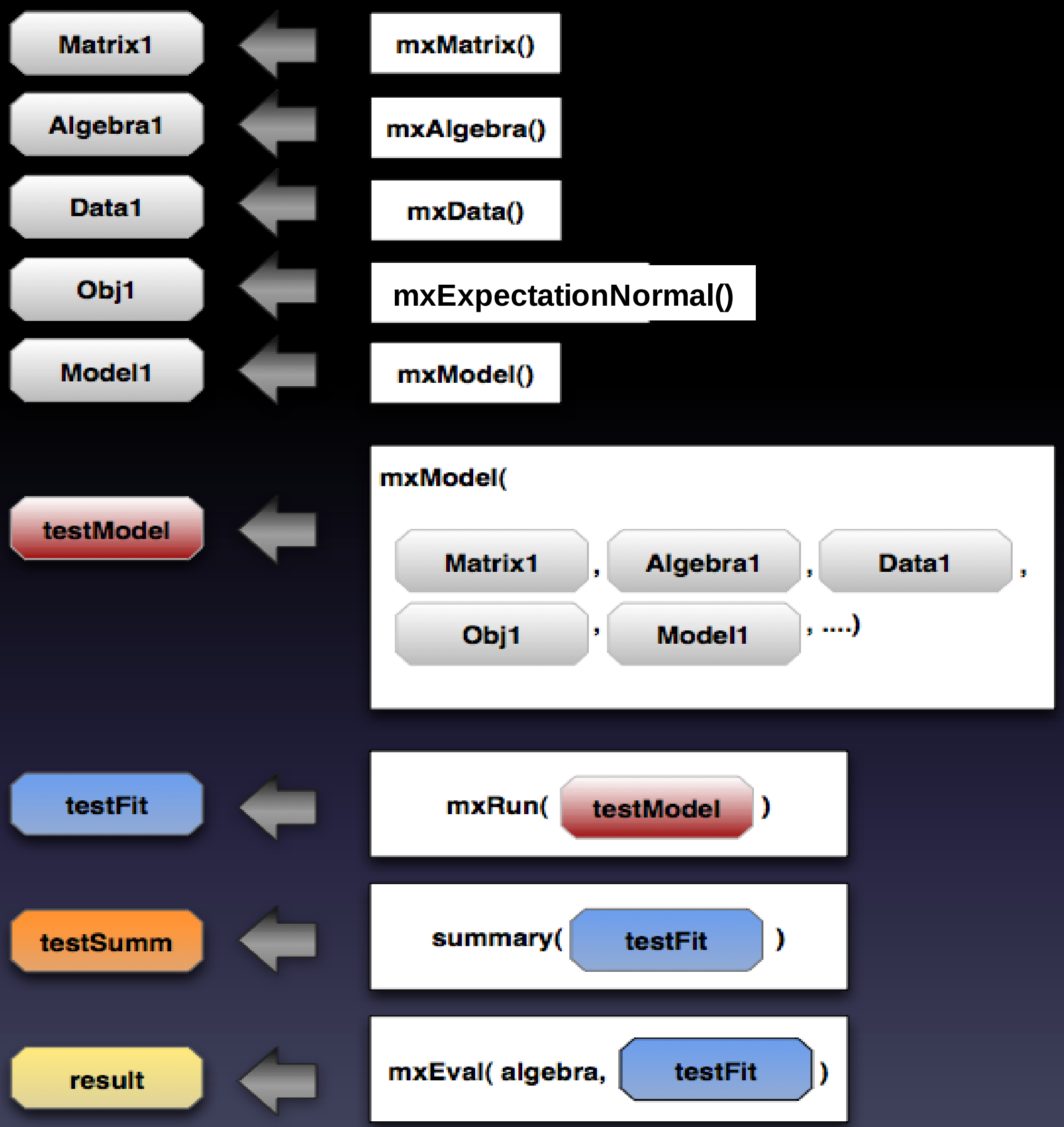
1x1 matrix
name “Site”

data.Site_T1

```
# Building the model  ()
expMean      <- mxAlgebra(intercept + bSite%x%Site, name="regress")

# Specify the data
regData      <- mxData( observed=Twins2a_1, type="raw"  )
inclusions   <- list(eVar, b0,  b1, bSite,  expMean)
exp          <- mxExpectationNormal( covariance="residualVar",
                means="regress",  dimnames=depVar )

# Run the model & summarize output
funML                  <- mxFitFunctionML()
regModel   <- mxModel( "Regression101", inclusions, regData, exp, funML )
regFit                 <- mxRun( regModel, intervals=FALSE )
regSum                 <- summary( regFit )
```



Make matrices *Variance, bo, b1, x*

Do Matrix Algebra w Matrices *expMean*

Call Data for Use in the Model *regData*

Build Model from Matrices/Algebras
exp via mxExpectationNormal

Build/Compile Overall Model from Matrices/Algebras and identify fit function
regModel / funML

Run Overall Model *regFit*

Get Summary Information from Overall Model
regSum

Generate Parameter Estimates from Overall Model *Resid, $Beta_0$, $Beta_1$*

The Challenge- Part 2

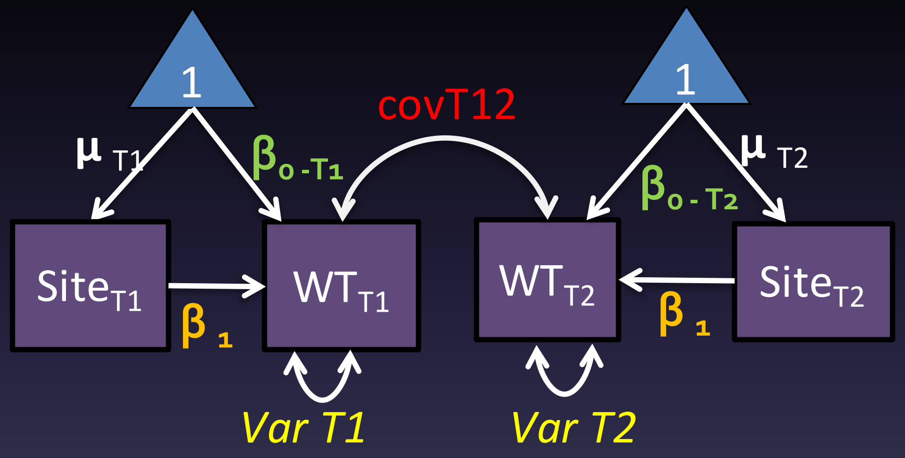
- You have been handed a dataset and you have been asked to consider the degree to which SITE influences both members of a twin pairs on the outcome (Weight).
  - Do the means differ in both twins for Weight or are they the same?
  - Does the influence of SITE on Weight differ in both twins or are they the same?

Summary of Regression201

free parameters:

	name	matrix	row	col	Estimate	Std.Error
1	varA	residualVar	WT_T1	WT_T1	1.01048364	0.0071452444
2	covAB	residualVar	WT_T1	WT_T2	0.40612385	0.0054232990
3	varB	residualVar	WT_T2	WT_T2	1.00106729	0.0070786532
4	beta0A	Intercept	1	1	51.70100780	0.0135685778
5	beta0B	Intercept	1	2	51.69860085	0.0135598998
6	beta1	bSite	1	1	0.30296722	0.0084022273

Regression with Two Members of a Twin Pair




```
eVar <- mxMatrix( type="Symm",
nrow=2, ncol=2, free=TRUE, values=c(10 ,1, 10),
labels=c('varT1 ,'covT12', 'varT2'),
name="residualVar" )
```

	T1	T2
T1	varT1	covT12
T2	covT12	varT2

cov = 2 x 2 matrix

```
bo <- mxMatrix(type="Full", nrow=1, ncol=2,
free=T, values=22,
labels=c("betaoT1","betaoT2"),
name="Intercept" )
```

T1	T2
betaoT1	betaoT2

```
x <- mxMatrix(type="Full", nrow=1, ncol=2,
free=F, labels=c("data.Site_T1", "data.Site_T2"),
name="Site" )
```

T1	T2
Site_T1	Site_T2

Summary

- First attempt at running a regression model using OpenMx and tested our results against a typical way of conducting regression in one member of a twin pair. Similar results!
- Extended this approach to both members of a twin pair
- Several steps in developing an OpenMx model
  - Advantage: Allows for considerable user flexibility to address all kinds of issues and models
  - Disadvantage: Learning the OpenMx language can initially be overwhelming. A complete understanding of OpenMx is not at all expected at this stage and many questions are likely to arise. PLEASE share your questions.

Thank You!

Elizabeth.Prom-Wormley@vcuhealth.org