

# Day 1 Tutorial – Getting Acquainted with Twin Modeling Using OpenMx

The text version of the Day 1 Qualtrics tutorial is below along with instructions on downloading files. This tutorial can be found at: [https://qimr.az1.qualtrics.com/jfe/form/SV\\_eqWgUTNJghdOBVA](https://qimr.az1.qualtrics.com/jfe/form/SV_eqWgUTNJghdOBVA)  
We'll start with Question 4 in the tutorial.

**Q4** This activity will help you learn the basics of running a simple twin analysis, including challenges and useful strategies. You and your team will explore the development and estimation of parameters from Saturated and ACE models. Let's begin!

**1- Visit the workshop website. This is your hub for all features related to conducting work related to the workshop.**

<https://workshop.colorado.edu/>

## 2- Copy files via SSH

Choose the SSH client from the workshop hub or go directly to the SSH client in a new browser tab by typing:

<https://workshop.colorado.edu/ssh>

Make sure you are in your home folder by typing:

```
cd ~
```

Create a directory to hold your day's work by typing:

```
mkdir day1
```

Go into the day1 directory by typing:

```
cd day1
```

Copy over the files/exercises from Elizabeth's directory into yours by typing the following:

```
cp /faculty/elizabeth/2022/* ./
```

Once this is complete, the following five files should be copied into your server folder:

1- Session1Activity5272022.R

The code we will be using for today's practical

2- miFunctions5272022.R

Code that contains several functions which makes it easier to see output from OpenMx

3- SimWtDataPairD1.csv

A Simulated dataset with 40,000 twin pairs and 7 variables containing data from 2 study site locations

4- SimWtDataSite1D1.csv

A Simulated dataset with 20,000 twin pairs and 7 variables containing data from Site 1

5- SimWtDataSite2D1.csv

A Simulated dataset with 20,000 twin pairs and 7 variables containing data from Site 2

**3- Open Session1Activity5272022.R in RStudio in a separate browser tab.**

<https://workshop.colorado.edu/rstudio/>

**4- Once you have opened Session1Activity5272022.R, go to line 16 to set YOUR working directory**  
`setwd("~/day1")`

**5- Please identify one person who is willing to run the code in R and is willing to share their screen with the rest of the team.**

All other team members are welcome to run the code on their own computers as this person is running and sharing their screen. All other team members will also be contributing to tutorial responses.

### **Q5 Your Scenario**

Your team has just been included as part of a larger group to conduct a univariate twin analysis of Weight (measured in kilograms) in a sample of  $N = 40,000$  same sex twin pairs ages 65 and older. The study was conducted at two locations (study sites). Recently, the investigators at the two locations have been reviewing their protocols and have been wondering whether minor differences between the two study sites could affect results in their twin analyses and ask for your help.

### **Q6 Task 1- Running Analyses Separately by Study Site**

Odd numbered rooms will run analyses for site 1.

Even numbered rooms will run analyses for site 2.

In Session1Activity2022.R , open the dataset for your room by un-commenting and running either line 37 or 38.  
ODD NUMBERED ROOMS- SimWtDataSite1D1.csv or  
EVEN NUMBERED ROOMS- SimWtDataSite2D1.csv

Enter the code you used to open the dataset in the text box below

### **ANSWER Q6 –**

```
Twins2 <- read.table(file="SimWtDataSite1D1.csv",header=T,sep=",") OR  
Twins2 <-read.table(file="SimWtDataSite2D1.csv",header=T,sep=",")
```

---

**Q7 Review and run lines 57-62 in Session1Activity2022.R for the following code and describe what you think is happening here.**

TIME: 2 minutes

```
library(OpenMx)  
mxOption( NULL, "Default optimizer", "NPSOL" )  
library(psych)  
library(polycor)  
source("miFunctions5272022.R")
```

### **ANSWER Q7 –**

- (1) We are opening the OpenMx package and requesting that OpenMx set the default optimizer to NPSOL.
- (2) We are opening two additional packages, psych and polycor
- (3) We are running another code called "miFunctions5272022.R".

A couple of additional notes. There were some questions regarding the optimizer line.

1- What is an optimizer? Very briefly, an optimizer is code that has been developed to find the best set of parameter estimates for a model to keep the misfit between data and a model as low as possible.

Consequently, an optimizer is needed to produce a measure of model fit (-2 Log Likelihood). Visit here for a student thesis on optimizers. <https://scholarscompass.vcu.edu/cgi/viewcontent.cgi?article=6347&context=etd>

2- How do you know which optimizer you should use? It depends on your data and model. In general, it is safe to start with NPSOL or CSOLNP. After working with NPSOL, you may find it challenging to get multivariate models with binary/ordinal data to produce estimates that make sense. If this occurs, consider switching to SLSQP.

### Q8 Running a Saturated Model for One Site

In this section, we will use a Saturated model and its related submodels to assess the classical assumptions of the ACE model.

Review and run lines 70-128 for the following lines of code in Session1Activity2022.R.

```
# Select Variables for Analysis
vars <- 'WT' # 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="_T")

# Select Data for Analysis
mzData <- subset(Twins2, ZYG==1, selVars)
dzData <- subset(Twins2, ZYG==2, selVars)

# Calculating means, variances, and covariances by twin member (Twin 1 or Twin2)
and zygosity groups in OpenMx
colMeans(mzData,na.rm=TRUE)
colMeans(dzData,na.rm=TRUE)
cov(mzData,use="complete")
cov(dzData,use="complete")

# Set Starting Values
svMe <- 50 # start value for means
svVa <- .8 # start value for variance
lbVa <- .0001 # lower bound for variance
# -----
# 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()

# 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 )
# -----
# RUN MODEL

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

```

**Q9** After running lines 70-128 in Session1Activity2022.R review and report results using this object:

sumSAT

TIME: 5 minutes

- Means for MZ Twin 1s \_\_\_\_\_
- Means for MZ Twin 2s \_\_\_\_\_
- Means for DZ Twin 1s \_\_\_\_\_
- Means for DZ Twin 2s \_\_\_\_\_
- Variances for MZ Twin 1s \_\_\_\_\_
- Variances for MZ Twin 2s \_\_\_\_\_
- Variances for DZ Twin 1s \_\_\_\_\_
- Variances for DZ Twin 2s \_\_\_\_\_
- Covariance for MZ twins \_\_\_\_\_
- Covariance for DZ twins \_\_\_\_\_

**ANSWER Q9**

<b>Parameter</b>	<b>Site 1</b>	<b>Site 2</b>
Means for MZ Twin 1s	62.014	62.292
Means for MZ Twin 2s	62.0068	62.295
Means for DZ Twin 1s	61.998	62.3032
Means for DZ Twin 2s	61.986	62.3028
Variances for MZ Twin 1s	3.022	3.014
Variances for MZ Twin 2s	3.023	2.985
Variances for DZ Twin 1s	3.0189	3.029
Variances for DZ Twin 2s	3.031	3.019
Covariance for MZ twins	2.423	2.405
Covariance for DZ twins	1.351	1.373

**Q10** You need to explain your code to the newest member of your team. Please summarize the details of what you think is happening in this bit of code in lines 94-95 of Session1Activity2022.R.

What exactly is being produced?

Why is this code necessary?

TIME: 2 minutes

```
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" )
```

**ANSWER Q10** – Here, we are defining two matrices with names of meanMZ and meanDZ (name= "meanMZ" and name = "meanDZ"). Both of these matrices have 1 row and 2 columns. They also have start values of svMe which were defined earlier in the code. The cells in the matrices have names (labels=c("mMZ1", "mMZ2") and labels=c("mDZ1", "mDZ2"))

The information in each of these matrices is passed to two objects meanMZ and meanDZ. This code is necessary to tell OpenMx which matrices will contain the information for the means that will go into the saturated model.

---

**Q11** Please summarize the details of what you think is happening in lines 98-101 of Session1Activity2022.R.

What exactly is being produced?

Why is this code necessary?

TIME: 2 minutes

```
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" )
```

**ANSWER Q11** – Here, we are defining two matrices with names of covMZ and covDZ (name= "covMZ" and name = "covDZ"). Both of these matrices have 2 rows and 2 columns. They also have start values of valDiag(svVa, ntv) which were defined earlier in the code. The cells in the matrices have names (labels=c("vMZ1", "cMZ21", "vMZ2") and labels=c("vDZ1", "cDZ21", "vDZ2"))

The information in each of these matrices is passed to two objects covMZ and covDZ. This code is necessary to tell OpenMx which matrices will contain the information for the variances and covariances that will go into the saturated model.

---

**Q12** Please summarize the details of what you think is happening in lines 104-105 of Session1Activity2022.R. What exactly is being produced?

Why is this code necessary?

TIME: 2 minutes

```
dataMZ <- mxData( observed=mzData, type="raw" )
dataDZ <- mxData( observed=dzData, type="raw" )
```

**ANSWER Q12** – Here, we produce two objects, dataMZ and dataDZ that contain the data from MZ twins and DZ twins. This code is necessary for OpenMx to know what data it should use to run the model.

**Q13** Please summarize the details of what you think is happening in lines 108-115 of Session1Activity2022.R. Is this code necessary? If so, why ?

TIME: 2 minutes

```
expMZ <- mxExpectationNormal( covariance="covMZ", means="meanMZ",
dimnames=selVars )
expDZ <- mxExpectationNormal( covariance="covDZ", means="meanDZ",
dimnames=selVars )
funML <- mxFitFunctionML()
```

```
modelMZ <- mxModel( meanMZ, covMZ, dataMZ, expMZ, funML, name="MZ")
modelDZ <- mxModel( meanDZ, covDZ, dataDZ, expDZ, funML, name="DZ")
multi <- mxFitFunctionMultigroup( c("MZ","DZ") )
```

**ANSWER Q12** –

- (1) Produce two objects (`expMZ` and `expDZ`) that contain the information on the matrices with means and variances/covariances to be used to produce the expected means, variances, and covariances that will be produced by the model.
  - (2) We are producing an object (`funML`) that will be used to compute the  $-2 \times (\log \text{likelihood})$  of the data given the current values of the free parameters and the expectation function (e.g., `mxExpectationNormal`) selected for the model. For more information on `mxFitFunctionML`, just type “`?mxFitFunctionML()`” in your console.
  - (3) We are producing two objects that contain all the previously defined objects that will be used for the MZ model (`modelMZ`) and the DZ model (`modelDZ`).
  - (4) Produce an object (`multi`) that contains the fit for the entire model using an aggregated fit statistics from multiple submodels (adding model fits from `modelMZ` and `modelDZ`).
-

**Q14** - Please summarize the details of what you think is happening in lines 122-128 of Session1Activity2022.R. What exactly is being produced?

What would you need to edit if you wanted to calculate 95% confidence intervals for your estimates?

TIME: 2 minutes

**ANSWER Q14 –**

```
modelSAT <- mxModel("oneSATc", modelMZ, modelDZ, multi, ciCov, ciMean)

fitSAT <- mxRun( modelSAT, intervals=F )
sumSAT <- summary( fitSAT )
```

(1) We are producing an object (`modelSAT`) that contains all objects that will be used to estimate the model.

Two additional objects (`ciCov` and `ciMean`) are named. These objects will be used to estimate 95% CI.

(2) Produce an object (`fitSAT`) that contains the parameter estimates generated from the Saturated model as run through the `mxRun()` function.

(3) Produce an object (`sumSAT`) that takes the results from the `fitSAT` model and summarizes them

**Q15** Review and run lines 135-159 in Session1Activity2022.R

TIME: 2 minutes

```
# -----
# RUN SUBMODELS
# Constrain expected Means to be equal across Twin Order
modelEMO <- mxModel( fitSAT, 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) )

```

---

**Q16** What is the purpose of this code?

```

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' )

```

**ANSWER Q16 –**

This code is taking the information from the saturated model (fitSAT) and equating the labels from the Saturated model `label=c("mMZ1","mMZ2")` and equating their estimates to be equal to one another using a new label (`newlabels='mMZ'`). This change is stored within the object `modelEMO`.

---

**Q17** - Report the values of the model fits (minus2LL) and their respective number of parameters for each of the following models

TIME: 3 minutes

- Saturated model
- Equating means across twin order (oneEMO<sub>c</sub>)
- Equating means and variances across twin order (one EMVO<sub>c</sub>)
- Equating means and variances across twin order and zygoty (oneEMVZ<sub>c</sub>)

**ANSWER Q17 –**

---

<b>Model</b>	<b>Site 1</b>		<b>Site 2</b>	
	<b>-2ll</b>	<b>Parameters</b>	<b>-2ll</b>	<b>Parameters</b>
Saturated model	145259.92	10	145013.63	10
Equating means across twin order (oneEMOc)	145260.81	8	145013.69	8
Equating means and variances across twin order (one EMVOc)	145260.86	6	145014.36	6
Equating means and variances across twin order and zygosity (oneEMVZc)	145261.53	4	145014.74	4

---

-----

**Q18** Based on these results we conclude that a model where the means and variances across twin order and zygosity explain the data as well as the saturated model where all means and variances are estimated. Therefore, we feel comfortable moving on to testing an ACE model.

TIME: 3 minutes

- True
- False
- Unsure

**Q19 TRUE.**

Your analyses from the saturated model allowed you to conclude that a model where the means and variances across twin order and zygosity explain the data as well as the saturated model where all means and variances are estimated. Therefore, we feel comfortable moving on to testing an ACE model.

-----

Page Break

---

## Q20 Running an ACE Model for One Site (Site 1 or Site 2)

In this section, we will run an ACE model using data for one study site.

Review and run code from lines 163-248 in Session1Activity2022.R

```
#### ACE Model 1- One Site Only ####
# Set Starting Values
svMe <- 50 # start value for means
svPa <- .2 # start value for path coefficient
svPe <- .5 # start value for path coefficient for e
# -----
-----

# 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" )
covC <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VC11", name="VC" )
covE <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VE11", name="VE" )

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

# 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, covC, 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 Unstandardized and Standardized Variance Components
rowUS <- rep('US',nv)
colUS <- rep(c('VA','VC','VE','SA','SC','SE'),each=nv)
estUS <- mxAlgebra( expression=cbind(VA,VC,VE,VA/V,VC/V,VE/V), name="US",
dimnames=list(rowUS,colUS) )

# Create Confidence Interval Objects
ciACE <- mxCI( "US[1,1:6]" )

# Build Model with Confidence Intervals
modelACE <- mxModel( "oneACEvc", pars, modelMZ, modelDZ, multi, estUS, ciACE )

# -----
# RUN MODEL
# Run ACE Model
fitACE <- mxRun( modelACE, intervals=T )
sumACE <- summary( fitACE )

# Compare with Saturated Model if saturated model was fitted in same session and
if saturated model prior to genetic model
mxCompare( fitSAT, fitACE )

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

# -----
# RUN SUBMODELS
# Run AE model
modelAE <- mxModel( fitACE, name="oneAEvc" )
modelAE <- omxSetParameters( modelAE, labels="VC11", free=FALSE, values=0 )
fitAE <- mxRun( modelAE, intervals=T )
fitGofs(fitAE); fitEstCis(fitAE)

# Run CE model
modelCE <- mxModel( fitACE, name="oneCEvc" )
modelCE <- omxSetParameters( modelCE, labels="VA11", free=FALSE, values=0 )
modelCE <- omxSetParameters( modelCE, labels=c("VE11","VC11"), free=TRUE,
values=.6 )
fitCE <- mxRun( modelCE, intervals=T )
fitGofs(fitCE); fitEstCis(fitCE)

# Run E model
modelE <- mxModel( fitAE, name="oneEvc" )

```

```

modelE <- omxSetParameters( modelE, labels="VA11", free=FALSE, values=0 )
fitE <- mxRun( modelE, intervals=T )
fitGofs(fitE); fitEstCis(fitE)

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

```

---

**Q21** Describe what is happening in lines 179-181 of Session1Activity2022.R

TIME: 2 minutes

```

covP <- mxAlgebra( expression= VA+VC+VE, name="V" )
covMZ <- mxAlgebra( expression= VA+VC, name="cMZ" )
covDZ <- mxAlgebra( expression= 0.5*x%VA+ VC, name="cDZ" )

```

**ANSWER Q21** – Here, we produce 3 objects that contain the results of 3 different algebras.

The first object (`covP`) contains the result from an `mxAlgebra` function to add the matrices VA, VC, and VE.

The sum of VA + VC + E produces the total phenotypic variance.

The second object (`covMZ`) contains a result from an `mxAlgebra` function to add the matrices VA and VC to produce the MZ covariance. The result from this algebra is contained within an object called `covMZ`.

The third object (`covDZ`) contains a result from an `mxAlgebra` function to add the matrices 0.5VA + VC to produce the DZ covariance.

---

**Q22** Look at the model fit comparison table (line 247). Based on those results, which model do you think best describes the data and why?

TIME: 3 minutes

```

mxCompare( fitACE, nested <- list(fitAE, fitCE, fitE) )

```

**ANSWER Q22** – An ACE model best fits the data. We know this because the model fit values of each submodel (AE model- Row 2, CE model- Row 3, and E Only model- Row 4) are significantly larger than the ACE model. Additionally, the AIC values for each of the submodels are larger for each submodel compared to the ACE model.

```

> mxCompare( fitACE, nested <- list(fitAE, fitCE, fitE) )

```

	base	comparison	ep	minus2LL	df	AIC	diffLL	diffdf	p
1	oneACEvc	<NA>	4	290642.81	79996	290650.81	NA	NA	NA
2	oneACEvc	oneAEvc	3	290730.37	79997	290736.37	87.56137	1	8.1704937e-21
3	oneACEvc	oneCEvc	3	295894.83	79997	295900.83	5252.02148	1	0.0000000e+00
4	oneACEvc	oneEvc	2	315984.74	79998	315988.74	25341.92987	2	0.0000000e+00

**Q23** Report the unstandardized parameter estimates from the best fitting model and their 95 % Confidence Intervals.

TIME: 3 minutes

- A \_\_\_\_\_
- C \_\_\_\_\_
- E \_\_\_\_\_

**ANSWER Q23 -**

Parameter	Site 1	Site 2
<b>A</b>	2.1492 (2.0549, 2.246)	2.1084 (2.0154, 2.2039)
<b>C</b>	0.2756 (0.1831, 0.3667)	0.311 (0.2196, 0.4012)
<b>E</b>	0.5993 (0.5831, 0.6163)	0.5947 (0.5786, 0.6115)

**Q24** Please note the parameter estimates and be prepared to share with the large group. We will reconvene as a large group shortly. If you are waiting, as a group, discuss and note any possible strategies the two study sites should use and implement to reduce variation due to study location.

**ANSWER Q24** - It might be nice for study sites to use the same equipment to measure weight. Also, it would be useful to train team members collecting weight data to measure weight the same way between both sites. Documenting the protocol in manual that is shared across site would also be useful.

## Q25 Task 2- Running Analyses Together Across Study Site.

We will now turn to analyzing data from the full sample and using data from both study sites.

A colleague previously ran the code for running an ACE model from lines 254-319 in Session1Activity2022.R.

```
#### ACE Model1 ####
Twins2<- read.table(file='SimWtDataPairD1.csv',header=T,sep=",")

# Select Variables for Analysis
vars <- 'WT' # 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="_T")

# Select Data for Analysis
mzData <- subset(Twins2, ZYG==1, selVars)
dzData <- subset(Twins2, ZYG==2, selVars)

# Set Starting Values
svMe <- 50 # start value for means
svPa <- .2 # start value for path coefficient
svPe <- .5 # start value for path coefficient for e
# -----
# 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" )
covC <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VC11", name="VC" )
covE <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VE11", name="VE" )

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

# 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, covC, 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 Unstandardized and Standardized Variance Components
rowUS <- rep('US',nv)
colUS <- rep(c('VA','VC','VE','SA','SC','SE'),each=nv)
estUS <- mxAlgebra( expression=cbind(VA,VC,VE,VA/V,VC/V,VE/V), name="US",
dimnames=list(rowUS,colUS) )

# Create Confidence Interval Objects
ciACE <- mxCI( "US[1,1:6]" )

# Build Model with Confidence Intervals
modelACE <- mxModel( "oneACEvc", pars, modelMZ, modelDZ, multi, estUS, ciACE )

# -----
# RUN MODEL
# Run ACE Model
fitACE <- mxRun( modelACE, intervals=TRUE )
sumACE <- summary( fitACE )

```



**Q26** Your colleague produced the following results were reported from the ACE Model with both sites together

Summary of oneACEvc

free parameters:

	name	matrix	row	col	Estimate	Std.Error	A
1	meanWT	meanG	1	1	62.14989513	0.0078237786	
2	VA11	VA	1	1	2.12787805	0.0342171371	
3	VC11	VC	1	1	0.31656728	0.0330053664	
4	VE11	VE	1	1	0.59712159	0.0059533170	

confidence intervals:

	lbound	estimate	ubound	note
oneACEvc.US[1,1]	2.061406995	2.12787805	2.19562933	
oneACEvc.US[1,2]	0.251534329	0.31656728	0.38100015	
oneACEvc.US[1,3]	0.585601373	0.59712159	0.60894360	
oneACEvc.US[1,4]	0.678133872	0.69959929	0.72138682	
oneACEvc.US[1,5]	0.082944489	0.10408033	0.12486296	
oneACEvc.US[1,6]	0.192053378	0.19632038	0.20069131	

Model Statistics:

	Parameters	Degrees of Freedom	Fit (-2lnL units)
Model:	4	79996	290642.81
Saturated:	NA	NA	NA
Independence:	NA	NA	NA

Number of observations/statistics: 40000/80000

Information Criteria:

	df	Penalty	Parameters	Penalty	Sample-Size Adjusted
AIC:		130650.81		290650.81	290650.81
BIC:		-557045.58		290685.20	290672.49

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: 2022-06-05 18:13:26

Wall clock time: 1.2562194 secs

optimizer: NPSOL

OpenMx version number: 2.20.6

Need help? See help(mxSummary)

**Q27** Report the unstandardized parameter estimates from the above results of the following:  
TIME- 5 minutes

- A \_\_\_\_\_
- C \_\_\_\_\_
- E \_\_\_\_\_

**ANSWER Q27 –**

	lbound	estimate	ubound
<b>A</b> oneACEvc.US [1,1]	2.061406995	2.12787805	2.19562933
<b>C</b> oneACEvc.US [1,2]	0.251534329	0.31656728	0.38100015
<b>E</b> oneACEvc.US [1,3]	0.585601373	0.59712159	0.60894360

---

**Q28** This model accounted for the variance due to site on the mean values of weight.

- True (1)
- False (2)
- Unsure (3)

**Q29 FALSE.**

Although this model used data from both sites, the influence of site on weight was not included in this model.

### Q30 Task 3- Re-Running and Accounting for Site

We will explore the role of site on the A, C, and E parameter estimates by running an ACE model that includes site in the means portion of the model.

Review and run lines 504-577 in Session1Activity2022.R.

TIME: 15 minutes

```
Twins3a <- subset(Twins3, !is.na(Twins3$Site))
covVars <- 'Site'

# Select Data for Analysis
mzData <- subset(Twins3a, ZYG==1, c(selVars,covVars))
dzData <- subset(Twins3a, ZYG==2, c(selVars,covVars))

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

# Set Starting Values
svBe <- 0.01 # start value for regressions
svMe <- 50 # start value for means
svPa <- .2 # start value for path coefficient
svPe <- .5 # start value for path coefficient for e

# -----
# PREPARE MODEL
# Create Matrices for Covariates and linear Regression Coefficients
defL <- mxMatrix( type="Full", nrow=1, ncol=1, free=FALSE,
labels=c("data.Site"), name="defL" )
pathB1 <- mxMatrix( type="Full", nrow=1, ncol=1, free=TRUE, values=svBe,
label="b11", name="b1" )

# Create Algebra for expected Mean Matrices
meanG <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe,
labels=labVars("mean",vars), name="meanG" )
expMean <- mxAlgebra( expression= meanG + cbind(defL%*%b1,defL%*%b1),
name="expMeanG" )

# Create Matrices for Variance Components
covA <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VA11", name="VA" )
covC <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VC11", name="VC" )
covE <- mxMatrix( type="Symm", nrow=nv, ncol=nv, free=TRUE, values=svPa,
label="VE11", name="VE" )
```

```

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

# 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="expMeanG",
dimnames=selVars )
expDZ <- mxExpectationNormal( covariance="expCovDZ", means="expMeanG",
dimnames=selVars )
funML <- mxFitFunctionML()

# Create Model Objects for Multiple Groups
pars <- list( pathBl, meanG, covA, covC, covE, covP )
defs <- list( defL )
modelMZ <- mxModel( pars, defs, expMean, covMZ, expCovMZ, dataMZ, expMZ, funML,
name="MZ" )
modelDZ <- mxModel( pars, defs, expMean, 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','VC','VE','SA','SC','SE'),each=nv)
estUS <- mxAlgebra( expression=cbind(VA,VC,VE,VA/V,VC/V,VE/V), name="US",
dimnames=list(rowUS,colUS) )

# Create Confidence Interval Objects
ciACE <- mxCI( "US[1,1:6]" )

# Build Model with Confidence Intervals
modelACE <- mxModel( "oneACEvca", pars, modelMZ, modelDZ, multi, estUS, ciACE )

# -----
# RUN MODEL
# Run ACE Model
fitACE <- mxRun( modelACE, intervals=T )
sumACE <- summary( fitACE )

```

**Q31** What is happening in lines 531-532 of Session1Activity5272022.R?

```
meanG <- mxMatrix( type="Full", nrow=1, ncol=ntv, free=TRUE, values=svMe,
labels=labVars("mean",vars), name="meanG" )
expMean <- mxAlgebra( expression= meanG + cbind(defL%*%bl,defL%*%bl),
name="expMeanG" )
```

**ANSWER Q31-**

- (1) We are producing a matrix called meanG that has 1 row and 2 columns. The information from this matrix is passed to an object called meanG
- (2) We are producing conducting the algebra that will account for the variance due to site on mean weight.

---

**Q32** Run lines 589-609 in Session1Activity2022.R. Report the model fit comparisons, identify the model that best explains the data, and report the parameter estimates along with 95 % CI from the model that best explains the data (you may or may not need to report all parameter estimates).

```
modelAE <- mxModel( fitACE, name="oneAEvca" )
modelAE <- omxSetParameters( modelAE, labels="VC11", free=FALSE, values=0 )
fitAE <- mxRun( modelAE, intervals=T )
fitGofs(fitAE); fitEstCis(fitAE)

# Run CE model
modelCE <- mxModel( fitACE, name="oneCEvca" )
modelCE <- omxSetParameters( modelCE, labels="VA11", free=FALSE, values=0 )
modelCE <- omxSetParameters( modelCE, labels=c("VE11","VC11"), free=TRUE,
values=.6 )
fitCE <- mxRun( modelCE, intervals=T )
fitGofs(fitCE); fitEstCis(fitCE)

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

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

- A (4) \_\_\_\_\_
- C (5) \_\_\_\_\_
- E (6) \_\_\_\_\_

**ANSWER Q32-** An ACE model best explains the data. Unstandardized parameter estimates:

	lbound	estimate	ubound	note
<b>A</b> oneACEvca.US [1,1]	2.06230653	2.128769416	2.19650576	
<b>C</b> oneACEvca.US [1,2]	0.22849360	0.293353691	0.35761143	
<b>E</b> oneACEvca.US [1,3]	0.58555301	0.597070250	0.60888995	

**Q33** Compare these results against those that were reported in Q26.

Which parameter estimate appears to vary the most as a result of including Site in the model? Please describe the differences.

**ANSWER Q33-** Of all parameter estimates, C varies the most.

Parameter	Model Without Site	Model With Site
VA (unstandardized)	2.1279	2.1288
VC (unstandardized)	0.3166	0.2934
VE (unstandardized)	0.5971	0.5971

**Q34** Based on these results, what would you report as the estimate of heritability? Why did you choose that estimate? How would you write about your result if you had to report for publication?

**ANSWER Q34-** Heritability is the proportion of the total phenotypic variance due to additive genetic effects. Therefore, the heritability of weight as estimated from the ACE model that accounted for the influence of site would be estimated as

$$\frac{VA}{VA + VC + VE}$$
$$\frac{2.1288}{2.1288 + 0.2934 + 0.5971} = 0.70$$

In a publication, we might say that that the heritability was 70% for weight in this simulated sample. We could also say that additive genetic effects accounted for 70% of the variance for weight.

### **Q35 Responsible Conduct of Research**

The research group is finalizing text to submit a manuscript for submission using the results from the analyses that your colleague conducted earlier which DID NOT include Site. You have been working on this project for one week.

Given that another person who has been on the team for one year started the analyses and you improved them, to what extent are you responsible for :

- 1- Letting the other person know the conclusions of your analyses?
- 2- Letting both Site leaders know the conclusions of your analyses

If you discuss, what will your strategy be to make them aware of what you have learned? Would you request to be included as a co-author on the manuscript?

**ANSWER Q35** - There were a lot of perspectives. See below. Generally, the group felt that sharing knowledge was very important but wanted to get a better feel of the culture of the team before pushing this issue strongly.

We are responsible for 1 and 2.

Approach the coworker in charge of the analysis, ask if they had considered controlling for site in their previous analysis. If so, why or why not? Discuss what I found my analysis of site. I would leave the decision for changing the paper in their hands, as well as setting up the meeting to inform the Site leaders (offering to attend to share results). We would not request co-authorship specifically, however, if we were asked to continue with this site analysis for the paper we would ask for co-authorship.

Site seems to be an important covariate that changes the results of the model, and therefore your colleagues should know. Authorship should be discussed.

You are very responsible for reporting the conclusions of the improved analyses to both the prior researcher and the site leaders. We would request to be included as a co-author on the manuscript should this analysis be used.

Share my results for the analysis by site. I would politely ask if I could be included as a co-author on the manuscript.

Yes to all of those things

You are responsible for letting the person know the conclusions are, and let them know. The results don't change very much (heritability changes only very slightly), and they can decide on whether to inform the site leaders and how to progress the manuscript as necessary. I would not request to be co-author on the manuscript, as I had only been working on it a week, but I think this is nuanced to different lab cultures.

1, not requesting co-authorship

Let them both know so the paper is updated -- come about it with humility, brief description of what you did, etc. We would not ask to be a coauthor based solely on one analysis that did not change discussion/implications.

It depends

You should let the research team know the results of accounting for site. We also discussed requesting to be included as a coauthor as it is a significant contribution.

We would consult with our mentor to communicate these findings most appropriately. If site-specific results are included, we would ask to be included on the manuscript as a co-author.

It depends :)

We would let the other person and both site leaders know about our analyses, since it was their prior data collection and analyses that allowed use to conduct the project. We would also ask to be included as a co-author for our work on the submitted manuscript.

We should tell the other person and the site leaders

I would definitely share the results of the analyses, but I wouldn't expect them to update the results because the chosen model did not change. If they decided to update their results, I would want to have a conversation about authorship

**This section is optional if there is additional time to take your reflection a bit deeper.**

### Q37 Thinking More about Site

After reviewing the code in Session1Activity5272022.R you notice that there is large block of code (lines 357-569) that was not run. If you wanted to prove to yourself the importance of Site on weight you could run this code.

#### ANSWER Q37-

Which code did you use to demonstrate the importance of Site?

```
mxCompare( fitSAT, subs <- list(fitCOV, fitEMO, fitEMVO, fitEMVZ) )
```

Does Site have a significant contribution to the variance of weight?

Yes.

How do you know?

	base	comparison	ep	minus2LL	df	AIC	diffLL	diffdf	p
1	oneSATca	<NA>	11	290276.17	79989	290298.17	NA	NA	NA
2	oneSATca	oneCOVca	10	290641.94	79990	290661.94	365.77077691	1	1.5597005e-81
3	oneSATca	oneEMOca	9	290276.48	79991	290294.48	0.31643860	2	8.5366255e-01
4	oneSATca	oneEMVOca	7	290276.78	79993	290290.78	0.61733580	4	9.6111794e-01

Comparing the model fit between a model with site (Row 1) against a model without site (Row 2) produced a p-value of  $1.56 \times 10^{-81}$ . Therefore, we conclude that the effect of Site on Weight is significant.

**Q38** Given the knowledge you have discovered above, should a model that includes Site be used?

- Yes (1)
- No (2)
- Unsure (3)

**ANSWER Q38-** Yes, include site in your model.