Day 1 Practical - Copy

Q1
Welcome! Before getting started, it's helpful to get to know your team members for today. Each team member is asked to answer the questions below. After you introduce yourself, announce the name of the next person to introduce themselves.
What is your name?
What is the name of your research institution and country of residence?
How comfortable are you with programming in R and OpenMx? (not at all comfortable, somewhat comfortable, very comfortable)
What is one thing that made you smile in the last 24 hours?
Once you have introduced yourselves, identify your room number in the text box below. One response per breakout room.

Please identify someone to record responses for the remainder of this practical.

TIME: This should take approximately 5 minutes to complete. We recommend that you identify someone to keep track of time.

____

Page Break

Q2
In order to keep your team moving smoothly, we suggest that team members consider how they will interact with one another. We offer suggestions below. Please review as a group and determine if you would like use these guidelines or if you would like to modify them.

Everyone’s thoughts and analysis are welcomed here. All students are encouraged to consistently and actively contribute to the discussion. Each group member brings their own expertise and experience to the discussion, and we encourage all opinions and ideas to be shared in an environment of mutual respect. Engaged active listening is encouraged. For example, using “I agree”/ “I disagree” and sharing the rationale as well as any references to it would be appreciated.

Zoom Etiquette. As a matter or courtesy to your fellow classmates, please consider the following recommendations to encourage clear discussion:

- Keep background noise to a minimum. This may include muting your microphone.
- If you cannot control the background noise, just let the group know and use the chat.
- Please keep video on during discussions as much as possible so that your team mates can engage with your facial expressions and body language.
- You are welcome to get up and move to go to the bathroom or answer urgent calls at any time. Just turn your camera and microphone off to let your classmates know that you are temporarily away via the chat box.
- Please feel free to eat during the workshop.

In the text box below, please indicate if your team decided to
(1) Adhere to these suggestions by typing "We'll stick to these suggestions" OR
(2) Make changes by typing "We made edits". Then identify your edits

TIME: This should take 5 minutes or less to complete

____

____

Page Break

Q3

Receiving Tutor Support

Please identify one team member to serve as the person who will request help from tutors using the raise your hand function in Zoom for the remainder of this session. This person will also articulate questions on behalf of the group to the tutor when they arrive. Once you have identified your person, raise your hand and ask for help. A tutor may not arrive immediately. While you're waiting, you can continue working on the tutorial.

When a tutor arrives:
1- Say hi
2- Ask them to introduce themself
3- Ask them to identify one thing that made them smile in the last 24 hours.

TIME: 1 minute

Page Break

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 (please note that there IS a period that must be included at the end of the second line):

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

Once this is complete, the following five files should be copied into your server folder:
1- Session1Activity2022.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 Session1Activity2022.R in RStudio in a separate browser tab.

```
https://workshop.colorado.edu/rstudio/
```

4- Once you have opened Session1Activity2022.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.

Page Break

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

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")
```

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 asssumptions 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)

 # Calcuating 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 (1) ____

○ Means for MZ Twin 2s (2) ____

○ Means for DZ Twin 1s (3) ____

○ Means for DZ Twin 2s (4) ____

○ Variances for MZ Twin 1s (5) ____

○ Variances for MZ Twin 2s (6) ____

○ Variances for DZ Twin 1s (7) ____

○ Variances for DZ Twin 2s (8) ____

○ Covariance for MZ twins (9) ____

○ Covariance for DZ twins (10) ____

Page Break

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" )
```

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" )
```

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" )
```

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") )
```

____

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

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

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

____

Page Break

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

```
 # --------------------------------------------------------------------
 # 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' )
```

TIME: 2 minutes

____

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 (1) ____

○ Equating means across twin order (oneEMOc) (2) ____

○ Equating means and variances across twin order (one EMVOc) (3) ____

○ Equating means and variances across twin order and zygosity (oneEMVZc) (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 (9)

○ False (10)

○ Unsure (11)

Page Break

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" )
```

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) )
```

Q23
Report the unstandardized parameter estimates from the best fitting model and their 95 % Confidence Intervals.
TIME: 3 minutes

○ A (1) ____

○ C (2) ____

○ E (3) ____

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.

____

____

____

____

____

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 (1) ____

○ C (2) ____

○ E (3) ____

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

○ True (1)

○ False (2)

○ Unsure (3)

Page Break

Q29
FALSE.
Although this model used data from both sites, the influence of site on weight was not included in this model.

Page Break

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" )
 pathBl <- mxMatrix( type="Full", nrow=1, ncol=1, free=TRUE, values=svBe,
label="b11", name="bl" )

 # 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%*%bl,defL%*%bl),
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 )
```

Page Break

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" )
```

Page Break

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) ____

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.

____

____

____

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?

Page Break

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?

____

Page Break

Q36
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. Which code did you use to demonstrate the importance of Site? Does Site have a significant contribution to the variance of weight? How do you know?

____

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

○ Yes (1)

○ No (2)

○ Unsure (3)

Q39 **Team Reflection**
As a team, consider:
1- What made your team successful today
2- What made your team less successful today
Although you will not be with this exact team over the next sessions, some of the strategies you identify here might help promote success in the future.

____

End of Block: Introductions