```r
# ------------------------------------------------------------------------------------------------------------------------
# Program: oneSATu.R
#  Author: Hermine Maes
#    Date: 02 17 2018
#
# Twin Univariate Saturated model to estimate means and (co)variances across multiple groups
# umx style model - Raw data - Continuous data / Binary data / Ordinal Data
# -------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|

# Load Libraries & Options
rm(list=ls())
library(OpenMx)
library(umx)
source('miFunctions.R')

# Create Output
filename    <- 'oneSATu'
sink(paste(filename,'.Ro',sep=''), append=FALSE, split=TRUE)

# ------------------------------------------------------------------------------------------------------------------------
# ------------------------------------------------------------------------------------------------------------------------
# PREPARE DATA - CONTINUOUS PHENOTYPE

# Load Continuous Data
selVars   <- c('bmi1','bmi2')
mzData    <- subset(twinData, zyg==1, selVars) #zygosity='MZFF' & cohort='younger'
dzData    <- subset(twinData, zyg==3, selVars) #zygosity='DZFF' & cohort='younger'

# ------------------------------------------------------------------------------------------------------------------------
# PREPARE & RUN MODEL & SUBMODELS for Continuous Data

# Saturated Model  # Add suffix to distinguish MZ paths from those of DZs
modelMZc  <- umxRAM( 'MZ', data=mzData, suffix='_MZ', umxPath(v.m.=selVars), umxPath(selVars[1], with = selVars[2]),
 autoRun=FALSE )
modelDZc  <- umxRAM( 'DZ', data=dzData, suffix='_DZ', umxPath(v.m.=selVars), umxPath(selVars[1], with = selVars[2]),
 autoRun=FALSE )
fitSATcu  <- umxSuperModel( 'oneSATcu', modelMZc, modelDZc )
fitGofs(fitSATcu); fitEsts(fitSATcu)

# Constrain expected Means to be equal across Twin Order
fitEMOcu  <- umxEquate( fitSATcu, name='oneEMOcu', autoRun=TRUE, comp=TRUE,
                        master=c('one_to_bmi1_MZ', 'one_to_bmi1_DZ'), slave=c('one_to_bmi2_MZ', 'one_to_bmi2_DZ') )
fitGofs(fitEMOcu); fitEsts(fitEMOcu)

# Alternative Code 2 Constrain expected Means to be equal across Twin Order
fitEMOcu2 <- umxModify( fitSATcu, update=c('one_to_bmi1_MZ', 'one_to_bmi2_MZ'), newlabels='one_to_bmi_MZ', name='oneEMOcu2',
 autoRun=FALSE )
fitEMOcu2 <- umxModify( fitEMOcu2, update=c('one_to_bmi1_DZ', 'one_to_bmi2_DZ'), newlabels='one_to_bmi_DZ', autoRun=TRUE,
 comp=TRUE )
fitGofs(fitEMOcu2); fitEsts(fitEMOcu2)
fitEMOcu3 <- umxModify( fitSATcu, regex='one_to_bmi[12]_MZ', newlabels='one_to_bmi_MZ', name='oneEMOcu3', autoRun=FALSE )
fitEMOcu3 <- umxModify( fitEMOcu3, regex='one_to_bmi[12]_DZ', newlabels='one_to_bmi_DZ', autoRun=TRUE, comp=TRUE )
fitGofs(fitEMOcu3); fitEsts(fitEMOcu3)

# Constrain expected Means and Variances to be equal across Twin Order
fitEMVOcu <- umxEquate( fitEMOcu, name='oneEMVOcu', autoRun=TRUE, comp=TRUE,
                        master=c('bmi1_with_bmi1_MZ', 'bmi1_with_bmi1_DZ'), slave =c('bmi2_with_bmi2_MZ', 'bmi2_with_bmi2_DZ') )
fitGofs(fitEMVOcu); fitEsts(fitEMVOcu)

# Constrain expected Means and Variances to be equal across Twin Order and Zygosity
fitEMVZcu <- umxEquate( fitEMVOcu, name='oneEMVZcu', autoRun=TRUE, comp=TRUE,
                        master=c('one_to_bmi1_DZ', 'bmi1_with_bmi1_DZ'), slave =c('one_to_bmi1_MZ', 'bmi1_with_bmi1_MZ') )
fitGofs(fitEMVZcu); fitEsts(fitEMVZcu)

# Print Comparative Fit Statistics
umxCompare( fitSATcu, c(fitEMOcu, fitEMVOcu, fitEMVZcu) )

# ------------------------------------------------------------------------------------------------------------------------
# ------------------------------------------------------------------------------------------------------------------------
# PREPARE DATA - BINARY PHENOTYPE

# Load Data and Create Binary Variables
binData   <- data.frame(bbmi1= rep(ifelse(twinData[,'bmi1']>22,1.00,0.00)),
                        bbmi2= rep(ifelse(twinData[,'bmi2']>22,1.00,0.00)), zyg= twinData[,'zyg'])
binVars   <- c('bbmi1','bbmi2')
bmzData   <- subset(binData, zyg==1, binVars)
bdzData   <- subset(binData, zyg==3, binVars)
bmzDataF  <- mxFactor( x=bmzData, levels=c(0:1) )
```

```
bdzDataF  <- mxFactor( x=bdzData, levels=c(0:1) )

# ------------------------------------------------------------------------------------------------------------------------
# PREPARE & RUN MODEL & SUBMODELS for Binary Data

# Saturated Model  # Add suffix to distinguish MZ paths from those of DZs
modelMZb  <- umxRAM( 'MZ', data=bmzDataF, suffix='_MZ', umxPath(v1m0=binVars), umxPath(binVars[1], with = binVars[2]),
 autoRun=FALSE )
modelMZb  <- umxModify( modelMZb, update=c('bbmi1_dev1','bbmi2_dev1'), newlabels=c('bbmi1_dev1_MZ','bbmi2_dev1_MZ'),
 autoRun=FALSE )
modelDZb  <- umxRAM( 'DZ', data=bdzDataF, suffix='_DZ', umxPath(v1m0=binVars), umxPath(binVars[1], with = binVars[2]),
 autoRun=FALSE )
modelDZb  <- umxModify( modelDZb, update=c('bbmi1_dev1','bbmi2_dev1'), newlabels=c('bbmi1_dev1_DZ','bbmi2_dev1_DZ'),
 autoRun=FALSE )
fitSATbu  <- umxSuperModel( 'oneSATbu', modelMZb, modelDZb )
fitGofs(fitSATbu); fitEsts(fitSATbu)

# Constrain expected Thresholds to be equal across Twin Order
fitETObu  <- umxEquate( fitSATbu, name='oneETObu', autoRun=TRUE, comp=TRUE,
                        master=c('bbmi1_dev1_MZ', 'bbmi1_dev1_DZ'), slave=c('bbmi2_dev1_MZ', 'bbmi2_dev1_DZ') )
fitGofs(fitETObu); fitEsts(fitETObu)

# Constrain expected Thresholds to be equal across Twin Order and Zygosity
fitETZbu <- umxEquate( fitETObu, name='oneETZbu', autoRun=TRUE, comp=TRUE,
                        master='bbmi1_dev1_DZ', slave ='bbmi1_dev1_MZ' )
fitGofs(fitETZbu); fitEsts(fitETZbu)

# Print Comparative Fit Statistics
umxCompare( fitSATbu, c(fitETObu, fitETZbu) )

# ------------------------------------------------------------------------------------------------------------------------
# ------------------------------------------------------------------------------------------------------------------------
# PREPARE DATA - ORDINAL PHENOTYPE

# Load Data and Create Ordinal Variables
nth       <- 3       # number of thresholds
ordData   <- data.frame(obmi1= twinData[,'bmi1'],
                        obmi2= twinData[,'bmi2'], zyg= twinData[,'zyg'])
quant     <- quantile(ordData[,c('obmi1','obmi2')],(0:(nth+1))/(nth+1),na.rm=TRUE)
 for (i in c('obmi1','obmi2')) { ordData[,i] <- cut(ordData[,i], breaks=quant, labels=c(0:nth)) }
ordVars   <- c('obmi1','obmi2')
omzData   <- subset(ordData, zyg==1, ordVars)
odzData   <- subset(ordData, zyg==3, ordVars)
omzDataF  <- mxFactor( x=omzData, levels=c(0:nth) )
odzDataF  <- mxFactor( x=odzData, levels=c(0:nth) )

# ------------------------------------------------------------------------------------------------------------------------
# PREPARE & RUN MODEL & SUBMODELS for Ordinal Data

# Saturated Model  # Add suffix to distinguish MZ paths from those of DZs
modelMZm  <- umxRAM( 'MZ', data=omzDataF, suffix='_MZ', umxPath(v.m.=ordVars), umxPath(ordVars[1], with = ordVars[2]) )
modelMZm  <- umxModify( modelMZm, update=c('obmi1_dev1','obmi2_dev1'), value=0, autoRun=FALSE )
modelMZm  <- umxModify( modelMZm, update=c('obmi1_dev2','obmi2_dev2'), value=1, autoRun=FALSE )
modelMZm  <- umxModify( modelMZm, update=c('obmi1_dev3','obmi2_dev3'), newlabels=c('obmi1_dev3_MZ','obmi2_dev3_MZ'),
 autoRun=FALSE )
modelDZm  <- umxRAM( 'DZ', data=odzDataF, suffix='_DZ', umxPath(v.m.=ordVars), umxPath(ordVars[1], with = ordVars[2]) )
modelDZm  <- umxModify( modelDZm, update=c('obmi1_dev1','obmi2_dev1'), value=0, autoRun=FALSE )
modelDZm  <- umxModify( modelDZm, update=c('obmi1_dev2','obmi2_dev2'), value=1, autoRun=FALSE )
modelDZm  <- umxModify( modelDZm, update=c('obmi1_dev3','obmi2_dev3'), newlabels=c('obmi1_dev3_DZ','obmi2_dev3_DZ'),
 autoRun=FALSE )
fitSATmu  <- umxSuperModel( 'oneSATmu', modelMZm, modelDZm )
fitGofs(fitSATmu); fitEsts(fitSATmu)

# Constrain expected Means, Variances & Thresholds to be equal across Twin Order
fitEMVOmu <- umxEquate( fitSATmu, name='oneEMVOmu', autoRun=TRUE, comp=TRUE,
                        master=c('one_to_obmi1_MZ','one_to_obmi1_DZ', 'obmi1_with_obmi1_MZ','obmi1_with_obmi1_DZ',
 'obmi1_dev3_MZ','obmi1_dev3_DZ'),
                         slave=c('one_to_obmi2_MZ','one_to_obmi2_DZ', 'obmi2_with_obmi2_MZ','obmi2_with_obmi2_DZ',
 'obmi2_dev3_MZ','obmi2_dev3_DZ') )
fitGofs(fitEMVOmu); fitEsts(fitEMVOmu)

# Constrain expected Means and Variances to be equal across Twin Order and Zygosity
fitEMVZmu <- umxEquate( fitEMVOmu, name='oneEMVZmu', autoRun=TRUE, comp=TRUE,
                        master=c('one_to_obmi1_DZ', 'obmi1_with_obmi1_DZ', 'obmi1_dev3_DZ'), slave =c('one_to_obmi1_MZ',
 'obmi1_with_obmi1_MZ', 'obmi1_dev3_MZ') )
fitGofs(fitEMVZmu); fitEsts(fitEMVZmu)

# Print Comparative Fit Statistics
```

```r
umxCompare( fitSATmu, c(fitEMVOmu, fitEMVZmu) )

# ------------------------------------------------------------------------------------------------------------------------
sink()
save.image(paste(filename,'.Ri',sep=''))
```