Electronic Appendix A: *Mplus Script for Gompertz Growth Curve*

```
TITLE:      NSLCM: Gompertz - ECLS;
DATA:       FILE = ecls_math_read_miss.dat;
VARIABLE:   NAMES = read_fk read_sk read_f1 read_s1 read_s3 read_s5 read_s8;
            MISSING = .;
ANALYSIS:   TYPE= MISSING MEANSTRUCTURE H1;
MODEL:
!Factor Loadings
        i BY    read_fk-read_s8@1;
        a BY    read_fk* (La1)
                read_sk  (La2)
                read_f1  (La3)
                read_s1  (La4)
                read_s3  (La5)
                read_s5  (La6)
                read_s8  (La7);
        d BY    read_fk* (Ld1)
                read_sk  (Ld2)
                read_f1  (Ld3)
                read_s1  (Ld4)
                read_s3  (Ld5)
                read_s5  (Ld6)
                read_s8  (Ld7);
        r BY    read_fk* (Lr1)
                read_sk  (Lr2)
                read_f1  (Lr3)
                read_s1  (Lr4)
                read_s3  (Lr5)
                read_s5  (Lr6)
                read_s8  (Lr7);
!Means
        [read_fk-read_s8@0];
        [i*-4] (mu_i); [a*134] (mu_a); [r@0 d@0];
!Variances
        read_fk-read_s8;
        i@0 a*600 r*.002 d*.059;
        i WITH a@0 r@0 d@0;
        a WITH r d;
        r WITH d;

MODEL CONSTRAINT:
   NEW(mu_r*0.228 mu_d*1.265);

La1 = exp(-1*exp(-mu_r*(0.0-mu_d)));
La2 = exp(-1*exp(-mu_r*(0.5-mu_d)));
La3 = exp(-1*exp(-mu_r*(1.0-mu_d)));
La4 = exp(-1*exp(-mu_r*(1.5-mu_d)));
La5 = exp(-1*exp(-mu_r*(3.5-mu_d)));
La6 = exp(-1*exp(-mu_r*(5.5-mu_d)));
La7 = exp(-1*exp(-mu_r*(8.5-mu_d)));

Ld1 = -mu_a*mu_r*exp(-exp(-mu_r*0.0)*((mu_r*0.0-mu_r*mu_d)*exp(mu_r*0.0)+exp(mu_r*mu_d)));
Ld2 = -mu_a*mu_r*exp(-exp(-mu_r*0.5)*((mu_r*0.5-mu_r*mu_d)*exp(mu_r*0.5)+exp(mu_r*mu_d)));
Ld3 = -mu_a*mu_r*exp(-exp(-mu_r*1.0)*((mu_r*1.0-mu_r*mu_d)*exp(mu_r*1.0)+exp(mu_r*mu_d)));
Ld4 = -mu_a*mu_r*exp(-exp(-mu_r*1.5)*((mu_r*1.5-mu_r*mu_d)*exp(mu_r*1.5)+exp(mu_r*mu_d)));
```

```
Ld5 = -mu_a*mu_r*exp(-exp(-mu_r*3.5)*((mu_r*3.5-mu_r*mu_d)*exp(mu_r*3.5)+exp(mu_r*mu_d)));
Ld6 = -mu_a*mu_r*exp(-exp(-mu_r*5.5)*((mu_r*5.5-mu_r*mu_d)*exp(mu_r*5.5)+exp(mu_r*mu_d)));
Ld7 = -mu_a*mu_r*exp(-exp(-mu_r*8.5)*((mu_r*8.5-mu_r*mu_d)*exp(mu_r*8.5)+exp(mu_r*mu_d)));

Lr1 = (mu_a*0.0*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*0.0)*(mu_r*0.0*exp(mu_r*0.0)+exp(mu_r*mu_d)));
Lr2 = (mu_a*0.5*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*0.5)*(mu_r*0.5*exp(mu_r*0.5)+exp(mu_r*mu_d)));
Lr3 = (mu_a*1.0*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*1.0)*(mu_r*1.0*exp(mu_r*1.0)+exp(mu_r*mu_d)));
Lr4 = (mu_a*1.5*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*1.5)*(mu_r*1.5*exp(mu_r*1.5)+exp(mu_r*mu_d)));
Lr5 = (mu_a*3.5*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*3.5)*(mu_r*3.5*exp(mu_r*3.5)+exp(mu_r*mu_d)));
Lr6 = (mu_a*5.5*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*5.5)*(mu_r*5.5*exp(mu_r*5.5)+exp(mu_r*mu_d)));
Lr7 = (mu_a*8.5*exp(mu_r*mu_d)-mu_a*mu_d*exp(mu_r*mu_d))*
                    exp(-exp(-mu_r*8.5)*(mu_r*8.5*exp(mu_r*8.5)+exp(mu_r*mu_d)));

OUTPUT:  SAMPSTAT STANDARDIZED TECH1 RESIDUAL;
```

Electronic Appendix B: *Mplus Script for Gompertz Growth Mixture Model*

```
TITLE:      GMM Gompertz - ECLS;
DATA:       FILE = ecls_math_read_miss.dat;
VARIABLE:   NAMES = read_fk read_sk read_f1 read_s1 read_s3 read_s5 read_s8;
            MISSING = .;
            CLASSES = c(2);
ANALYSIS:   TYPE= MISSING MIXTURE;
MODEL:
%OVERALL%
        i BY    read_fk-read_s8@1;
        a BY    read_fk* read_sk read_f1 read_s1 read_s3 read_s5 read_s8;
        d BY    read_fk* read_sk read_f1 read_s1 read_s3 read_s5 read_s8;
        r BY    read_fk* read_sk read_f1 read_s1 read_s3 read_s5 read_s8;
!Means & Intercepts
        [read_fk-read_s8@0]; [i*-4 a*134 r@0 d@0];
!Variances & Covariances
        read_fk-read_s8;
        i@0; a*600 (v_a); r*.002 (v_r); d*.059 (v_d);
        i WITH a@0; i WITH r@0; i WITH d@0;
        a WITH r (c_ar); a WITH d (c_ad); r WITH d (c_rd);
%c#1%
        a BY    read_fk* (La11)
                read_sk  (La21)
                read_f1  (La31)
                read_s1  (La41)
                read_s3  (La51)
                read_s5  (La61)
                read_s8  (La71);
        d BY    read_fk* (Ld11)
                read_sk  (Ld21)
                read_f1  (Ld31)
                read_s1  (Ld41)
                read_s3  (Ld51)
                read_s5  (Ld61)
                read_s8  (Ld71);
        r BY    read_fk* (Lr11)
                read_sk  (Lr21)
                read_f1  (Lr31)
                read_s1  (Lr41)
                read_s3  (Lr51)
                read_s5  (Lr61)
                read_s8  (Lr71);

      [i*-4] (mu_i); [a*134] (mu_a1);

%c#2%
        a BY    read_fk* (La12)
                read_sk  (La22)
                read_f1  (La32)
                read_s1  (La42)
                read_s3  (La52)
                read_s5  (La62)
                read_s8  (La72);
        d BY    read_fk* (Ld12)
                read_sk  (Ld22)
```

```
                read_f1  (Ld32)
                read_s1  (Ld42)
                read_s3  (Ld52)
                read_s5  (Ld62)
                read_s8  (Ld72);
        r BY    read_fk* (Lr12)
                read_sk  (Lr22)
                read_f1  (Lr32)
                read_s1  (Lr42)
                read_s3  (Lr52)
                read_s5  (Lr62)
                read_s8  (Lr72);

      [i*-4] (mu_i); [a*158] (mu_a2);

MODEL CONSTRAINT:
new(mu_r1*0.228 mu_d1*1.265 mu_r2*0.228 mu_d2*2.265);

!Constraints for Class 1;

La11 = exp(-1*exp(-mu_r1*(0.0 - mu_d1)));
La21 = exp(-1*exp(-mu_r1*(0.5 - mu_d1)));
La31 = exp(-1*exp(-mu_r1*(1.0 - mu_d1)));
La41 = exp(-1*exp(-mu_r1*(1.5 - mu_d1)));
La51 = exp(-1*exp(-mu_r1*(3.5 - mu_d1)));
La61 = exp(-1*exp(-mu_r1*(5.5 - mu_d1)));
La71 = exp(-1*exp(-mu_r1*(8.5 - mu_d1)));

Ld11  = -mu_a1*mu_r1*exp(-exp(-mu_r1*0.0)*((mu_r1*0.0-mu_r1*mu_d1)*
            exp(mu_r1*0.0)+exp(mu_r1*mu_d1)));
Ld21  = -mu_a1*mu_r1*exp(-exp(-mu_r1*0.5)*((mu_r1*0.5-mu_r1*mu_d1)*
            exp(mu_r1*0.5)+exp(mu_r1*mu_d1)));
Ld31  = -mu_a1*mu_r1*exp(-exp(-mu_r1*1.0)*((mu_r1*1.0-mu_r1*mu_d1)*
            exp(mu_r1*1.0)+exp(mu_r1*mu_d1)));
Ld41  = -mu_a1*mu_r1*exp(-exp(-mu_r1*1.5)*((mu_r1*1.5-mu_r1*mu_d1)*
            exp(mu_r1*1.5)+exp(mu_r1*mu_d1)));
Ld51  = -mu_a1*mu_r1*exp(-exp(-mu_r1*3.5)*((mu_r1*3.5-mu_r1*mu_d1)*
            exp(mu_r1*3.5)+exp(mu_r1*mu_d1)));
Ld61  = -mu_a1*mu_r1*exp(-exp(-mu_r1*5.5)*((mu_r1*5.5-mu_r1*mu_d1)*
            exp(mu_r1*5.5)+exp(mu_r1*mu_d1)));
Ld71  = -mu_a1*mu_r1*exp(-exp(-mu_r1*8.5)*((mu_r1*8.5-mu_r1*mu_d1)*
            exp(mu_r1*8.5)+exp(mu_r1*mu_d1)));

Lr11 = (mu_a1*0.0*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*0.0)*(mu_r1*0.0*exp(mu_r1*0.0)+exp(mu_r1*mu_d1)));
Lr21 = (mu_a1*0.5*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*0.5)*(mu_r1*0.5*exp(mu_r1*0.5)+exp(mu_r1*mu_d1)));
Lr31 = (mu_a1*1.0*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*1.0)*(mu_r1*1.0*exp(mu_r1*1.0)+exp(mu_r1*mu_d1)));
Lr41 = (mu_a1*1.5*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*1.5)*(mu_r1*1.5*exp(mu_r1*1.5)+exp(mu_r1*mu_d1)));
Lr51 = (mu_a1*3.5*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*3.5)*(mu_r1*3.5*exp(mu_r1*3.5)+exp(mu_r1*mu_d1)));
Lr61 = (mu_a1*5.5*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*5.5)*(mu_r1*5.5*exp(mu_r1*5.5)+exp(mu_r1*mu_d1)));
Lr71 = (mu_a1*8.5*exp(mu_r1*mu_d1)-mu_a1*mu_d1*exp(mu_r1*mu_d1))*
                    exp(-exp(-mu_r1*8.5)*(mu_r1*8.5*exp(mu_r1*8.5)+exp(mu_r1*mu_d1)));
```

```
!Constraints for Class 2;

La12 = exp(-1*exp(-mu_r2*(0.0 - mu_d2)));
La22 = exp(-1*exp(-mu_r2*(0.5 - mu_d2)));
La32 = exp(-1*exp(-mu_r2*(1.0 - mu_d2)));
La42 = exp(-1*exp(-mu_r2*(1.5 - mu_d2)));
La52 = exp(-1*exp(-mu_r2*(3.5 - mu_d2)));
La62 = exp(-1*exp(-mu_r2*(5.5 - mu_d2)));
La72 = exp(-1*exp(-mu_r2*(8.5 - mu_d2)));

Ld12  = -mu_a2*mu_r2*exp(-exp(-mu_r2*0.0)*((mu_r2*0.0-mu_r2*mu_d2)*
            exp(mu_r2*0.0)+exp(mu_r2*mu_d2)));
Ld22  = -mu_a2*mu_r2*exp(-exp(-mu_r2*0.5)*((mu_r2*0.5-mu_r2*mu_d2)*
            exp(mu_r2*0.5)+exp(mu_r2*mu_d2)));
Ld32  = -mu_a2*mu_r2*exp(-exp(-mu_r2*1.0)*((mu_r2*1.0-mu_r2*mu_d2)*
            exp(mu_r2*1.0)+exp(mu_r2*mu_d2)));
Ld42  = -mu_a2*mu_r2*exp(-exp(-mu_r2*1.5)*((mu_r2*1.5-mu_r2*mu_d2)*
            exp(mu_r2*1.5)+exp(mu_r2*mu_d2)));
Ld52  = -mu_a2*mu_r2*exp(-exp(-mu_r2*3.5)*((mu_r2*3.5-mu_r2*mu_d2)*
            exp(mu_r2*3.5)+exp(mu_r2*mu_d2)));
Ld62  = -mu_a2*mu_r2*exp(-exp(-mu_r2*5.5)*((mu_r2*5.5-mu_r2*mu_d2)*
            exp(mu_r2*5.5)+exp(mu_r2*mu_d2)));
Ld72  = -mu_a2*mu_r2*exp(-exp(-mu_r2*8.5)*((mu_r2*8.5-mu_r2*mu_d2)*
            exp(mu_r2*8.5)+exp(mu_r2*mu_d2)));

Lr12 = (mu_a2*0.0*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*0.0)*(mu_r2*0.0*exp(mu_r2*0.0)+exp(mu_r2*mu_d2)));
Lr22 = (mu_a2*0.5*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*0.5)*(mu_r2*0.5*exp(mu_r2*0.5)+exp(mu_r2*mu_d2)));
Lr32 = (mu_a2*1.0*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*1.0)*(mu_r2*1.0*exp(mu_r2*1.0)+exp(mu_r2*mu_d2)));
Lr42 = (mu_a2*1.5*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*1.5)*(mu_r2*1.5*exp(mu_r2*1.5)+exp(mu_r2*mu_d2)));
Lr52 = (mu_a2*3.5*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*3.5)*(mu_r2*3.5*exp(mu_r2*3.5)+exp(mu_r2*mu_d2)));
Lr62 = (mu_a2*5.5*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*5.5)*(mu_r2*5.5*exp(mu_r2*5.5)+exp(mu_r2*mu_d2)));
Lr72 = (mu_a2*8.5*exp(mu_r2*mu_d2)-mu_a2*mu_d2*exp(mu_r2*mu_d2))*
                    exp(-exp(-mu_r2*8.5)*(mu_r2*8.5*exp(mu_r2*8.5)+exp(mu_r2*mu_d2)));


OUTPUT:  SAMPSTAT STANDARDIZED TECH1 RESIDUAL;
```

Electronic Appendix C: *OpenMx Script for a Gompertz Growth Curve*

```
library(OpenMx)

# Read Data
rawData <- read.table("ecls_math_read_miss.dat", header=F, na.string='.')

# Create Vector of Variable Names
varNames <- c("read_fk", "read_sk", "read_f1", "read_s1", "read_s3", "read_s5", "read_s8")

# Assign Column (Variable) Names to Raw Data
names(rawData) <- varNames

# Create an mxData Object
data <- mxData(rawData[,read], type="raw")

# # Model Specification

# Factor means
factorMeans <- mxMatrix("Full", 1, 4,  free = c(TRUE, TRUE, FALSE, FALSE),
    values = c(21, 144, 0, 0), labels = c("i", "a", NA, NA),
    name = "mu")

# Factor covariances
factorCov <- mxMatrix("Symm", 4, 4,
    free = c(FALSE,
             FALSE, TRUE,
             FALSE, TRUE, TRUE,
             FALSE, TRUE, TRUE, TRUE),
    values = c(0,
               0, 6000,
               0,   0, .002,
               0,   0,    0, .002),
    labels = c("var_i" ,
               NA, "var_a",
               NA, "cov_ad",  "var_d",
               NA, "cov_ar", "cov_rd", "var_r"),
    byrow=TRUE,
    name="phi")

# Residual variances
manCov <- mxMatrix("Diag", 7, 7,  free = TRUE,
    values = c(20, 20, 20, 20, 20, 20, 20),
    labels = paste("e", c(1,2,3,4,5,6,7), sep=""),
    name = "resid"
)

# Matrices for constraints

#Matrix of ones
unit <- mxMatrix("Unit", 7, 1, name="one")

#Observation times
time <- mxMatrix("Full", 7, 1,  free = FALSE,
    values = c(0.0, 0.5, 1.0, 1.5, 3.5, 5.5, 8.5),
    name = "t")
```

```
aParameter <- mxMatrix("Full", 7, 1,  free = TRUE,
    values = 144,
    labels = "a",
    name = "alpha")

dParameter <- mxMatrix("Full", 7, 1,  free = TRUE,
    values = 1.265,
    labels = "d",
    name = "delta")

rParameter <- mxMatrix("Full", 7, 1,  free = TRUE,
    values = 0.228,
    labels = "r",
    name = "rho")

## Factor loading matrix

# Build the individual columns of lambda based on Gompertz Constraints

l1 <- mxMatrix("Unit", 7, 1, name="lambda1")
l2 <- mxAlgebra(
      exp(-1*exp(-rho*(t-delta))), name="lambda2")
l3 <- mxAlgebra(
      -alpha*rho*exp(-exp(-rho*t)*((rho*t-rho*delta)*exp(rho*t)+exp(rho*delta))),
      name="lambda3")
l4 <- mxAlgebra(
      (alpha*t*exp(rho*delta)-alpha*delta*exp(rho*delta))*
           exp(-exp(-rho*t)*(rho*t*exp(rho*t)+exp(rho*delta))),
      name="lambda4")

# assemble the columns of lambda

loadings <- mxAlgebra(cbind(lambda1, lambda2, lambda3, lambda4), name="lambda")

## Model Expectations

meanAlg <- mxAlgebra(mu %*% t(lambda), name="mean")

covAlg <- mxAlgebra(lambda %*% phi %*% t(lambda) + resid, name="cov")

# The model

model <- mxModel("Gompertz Growth Curve",         # name for the model
    data,                                         # data of some type
    factorMeans, factorCov, manCov,               # latent and manifest variable parameters
    unit, time,                                   # shortcuts for unit and time information
    aParameter, dParameter, rParameter,           # required pieces of lambda columns
    l1, l2, l3, l4, loadings,                     # factor loadings
    meanAlg, covAlg,                              # expected means and covariances
    mxFIMLObjective("cov", "mean", dimnames=read) # objective function
)

results <- mxRun(model)

summary(results)
```

Electronic Appendix D: *OpenMx Script for the Gompertz Growth Mixture Model*

```
library(OpenMx)

# Read Data
rawData <- read.table("ecls_math_read_miss.dat", header=F, na.string='.')

# Create Vector of Variable Names
varNames <- c("read_fk", "read_sk", "read_f1", "read_s1", "read_s3", "read_s5", "read_s8")

# Assign Column (Variable) Names to Raw Data
names(rawData) <- varNames

# Create an mxData Object
data <- mxData(rawData[,read], type="raw")

# # Model Specification

# factor means
factorMeans1 <- mxMatrix("Full", 1, 4, free = c(TRUE, TRUE, FALSE, FALSE),
     values = c(21, 144, 0, 0),
     labels = c("i", "a1", NA, NA),
     name = "mu1"
     )

factorMeans2 <- mxMatrix("Full", 1, 4, free = c(TRUE, TRUE, FALSE, FALSE),
     values = c(21, 157, 0, 0),
     labels = c("i", "a2", NA, NA),
     name = "mu2"
     )

# factor covariances

factorCov <- mxMatrix("Symm", 4, 4,
     free = c(FALSE,
              FALSE, TRUE,
              FALSE, TRUE, TRUE,
              FALSE, TRUE, TRUE, TRUE),
     values = c(0,
                0, 507,
                0,   0,  0.002,
                0,   0,      0, 0.2),
     labels = c("var_i",
                NA,  "var_a",
                NA, "cov_ad",  "var_d",
                NA, "cov_ar", "cov_rd", "var_r"),
     byrow=TRUE,
     name="phi"
     )

#residual variances

manCov <- mxMatrix("Diag", 7, 7, free = TRUE,
     values = c(20, 20, 20, 20, 20, 20, 20),
     labels = paste("e", c(1,2,3,4,5,6,7), sep=""),
     name = "resid" )
```

```
# # useful matrices for constraints
#matrix of ones
unit <- mxMatrix("Unit", 7, 1,
     name="one"
     )

#observation times
time <- mxMatrix("Full", 7, 1,
     free = FALSE,
     values = c(0.0, 0.5, 1.0, 1.5, 3.5, 5.5, 8.5),
     name = "t"
     )

aParameter1 <- mxMatrix("Full", 7, 1,
     free = TRUE,
     values = 144,
     labels = "a1",
     name = "alpha1"
     )

aParameter2 <- mxMatrix("Full", 7, 1,
     free = TRUE,
     values = 157,
     labels = "a2",
     name = "alpha2"
     )

dParameter1 <- mxMatrix("Full", 7, 1,
     free = TRUE,
     values = 1.9,
     labels = "d1",
     name = "delta1"
     )

dParameter2 <- mxMatrix("Full", 7, 1,
     free = TRUE,
     values = 0.9,
     labels = "d2",
     name = "delta2"
     )

rParameter1 <- mxMatrix("Full", 7, 1,
     free = TRUE,
     values = 0.3,
     labels = "r1",
     name = "rho1"
     )

rParameter2 <- mxMatrix("Full", 7, 1,
     free = TRUE,
     values = 0.4,
     labels = "r2",
     name = "rho2"
     )

# # Factor loading matrix
```

```
# build the individual columns of lambda

l11 <- mxMatrix("Unit", 7, 1,
     name="lambda11"
     )
l12 <- mxAlgebra(
      exp(-1*exp(-rho1*(t-delta1))),
     name="lambda12"
     )
l13 <- mxAlgebra(
      -alpha1*rho1*exp(-exp(-rho1*t)*((rho1*t-rho1*delta1)*exp(rho1*t)+exp(rho1*delta1))),
      name="lambda13"
     )
l14 <- mxAlgebra(
      (alpha1*t*exp(rho1*delta1)-alpha1*delta1*exp(rho1*delta1))*
           exp(-exp(-rho1*t)*(rho1*t*exp(rho1*t)+exp(rho1*delta1))),
      name="lambda14"
     )

l21 <- mxMatrix("Unit", 7, 1,
     name="lambda21"
     )
l22 <- mxAlgebra(
      exp(-1*exp(-rho2*(t-delta2))),
     name="lambda22"
     )
l23 <- mxAlgebra(
      -alpha2*rho2*exp(-exp(-rho2*t)*((rho2*t-rho2*delta2)*exp(rho2*t)+exp(rho2*delta2))),
      name="lambda23"
     )
l24 <- mxAlgebra(
      (alpha2*t*exp(rho2*delta2)-alpha2*delta2*exp(rho2*delta2))*
           exp(-exp(-rho2*t)*(rho2*t*exp(rho2*t)+exp(rho2*delta2))),
      name="lambda24"
     )

# assemble the columns of lambda
loadings1 <- mxAlgebra(
              cbind(lambda11, lambda12, lambda13, lambda14), name="lambda1")
loadings2 <- mxAlgebra(
              cbind(lambda21, lambda22, lambda23, lambda24), name="lambda2")

# # Model Expectations
# here are the mean and cov algebras for the individual classes

meanAlg1 <- mxAlgebra(mu1 %*% t(lambda1), name="mean1")
covAlg1 <- mxAlgebra(lambda1 %*% phi %*% t(lambda1) + resid, name="cov1")

meanAlg2 <- mxAlgebra(mu2 %*% t(lambda2), name="mean2")
covAlg2 <- mxAlgebra(lambda2 %*% phi %*% t(lambda2) + resid, name="cov2")

# # Model for the two classes

model1 <- mxModel("Gompertz1",                    # name for the model
     factorMeans1, factorCov, manCov,             # latent and manifest variable parameters
     unit, time,                                  # shortcuts for unit and time information
     aParameter1, dParameter1, rParameter1,       # required pieces of lambda columns
```

```
     l11, l12, l13, l14, loadings1,              # factor loadings
     meanAlg1, covAlg1,                          # expected means and covariances
     mxFIMLObjective("cov1", "mean1",
     dimnames=read, vector=TRUE) )

model2 <- mxModel("Gompertz2",                   # name for the model
     factorMeans2, factorCov, manCov,            # latent and manifest variable parameters
     unit, time,                                 # shortcuts for unit and time information
     aParameter2, dParameter2, rParameter2,      # required pieces of lambda columns
     l21, l22, l23, l24, loadings2,              # factor loadings
     meanAlg2, covAlg2,                          # expected means and covariances
     mxFIMLObjective("cov2", "mean2",
     dimnames=read, vector=TRUE) )

# # Model Expectations for Mixture (Parent) Model

classP <- mxMatrix("Full", 2, 1, free=TRUE, lbound=0,
           labels = c("pclass1", "pclass2"), name="classProbs")

classS <- mxAlgebra(sum(classProbs), name="classSum")

constant <- mxMatrix("Iden", 1, name="con")

classC <- mxConstraint(classSum==con, name="classCon")

# Objective Function
algObj <- mxAlgebra(-2*sum(
           log(pclass1%x%Gompertz1.objective + pclass2%x%Gompertz2.objective)),
          name="mixtureObj")

obj <- mxAlgebraObjective("mixtureObj")

# Model for Mixture Model

mixedModel <- mxModel("Mixture Model",           # name for the model
     model1, model2,                             # models for each class
     data,                                       # data
     classP, classS, constant, classC,           # requirements for class probabilities
     algObj, obj                                 # objective function
     )

mixedResults <- mxRun(mixedModel)

summary(mixedResults)
```