

Introduction to OpenMx

Sarah Medland

What is OpenMx?

- Free, Open-source, full-featured SEM package
- Software which runs on Windows, Mac OSX, and Linux
- A package in the R statistical programming environment
- Two main approaches to writing OpenMx models – Path or **Matrix** Specification

- R is a functional language
 - Easy to define new functions
 - Items are stored as Objects
- OpenMx uses functions to build objects
 - Arguments to the function have an order
 - Order can be changed by naming arguments

Matrices are the building blocks

Matrix: a rectangular array of elements arranged in rows and columns.

$$\mathbf{A} = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$$

(3x3)

Columns

Rows

The **order** or **dimension** of a matrix is defined by the number of row and columns in the matrix.

The order of a matrix is generally referred to as $M \times N$ (where M is the number of rows and N is the number of columns)

Matrix A is a 3 x 3 matrix.

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

Therefore, e is element $a_{(2,2)}$

Matrices are the building blocks

```
mxMatrix( type="Lower", nrow=nv, ncol=nv, free=TRUE,  
values=.6, label="a11", name="a" ), # additive genetic path  
coefficients
```

- Many types eg. type="Lower"
- Denoted by names eg. name="a"
- Size eg. nrow=nv, ncol=nv
- Estimated parameters must be placed in a matrix & Mx must be told what type of matrix it is

Matrices are the building blocks

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

```
? 0 0  
0 ? 0  
0 0 ?
```

`mxMatrix(type="Sdiag", nrow=3, ncol=3, free=TRUE, name="a")`

```
0 0 0  
? 0 0  
? ? 0
```

`mxMatrix(type="Stand", nrow=3, ncol=3, free=TRUE, name="a")`

```
1 ? ?  
? 1 ?  
? ? 1
```

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

```
? ? ?  
? ? ?  
? ? ?
```

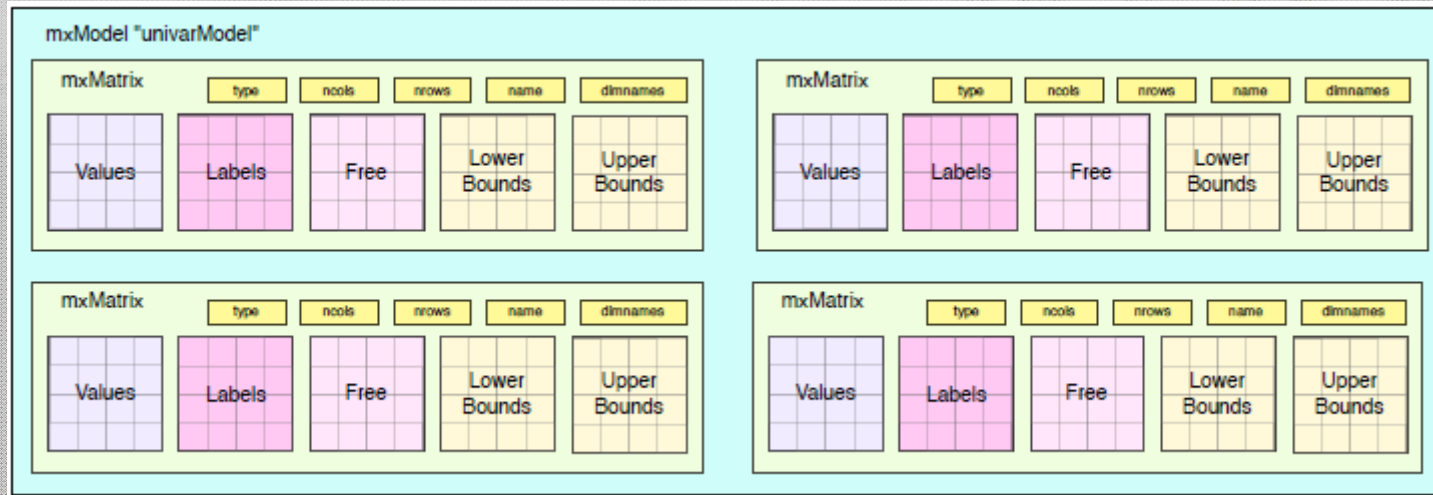
`mxMatrix(type="Lower", nrow=3, ncol=3, free=TRUE, name="a")`

```
? 0 0  
? ? 0  
? ? ?
```

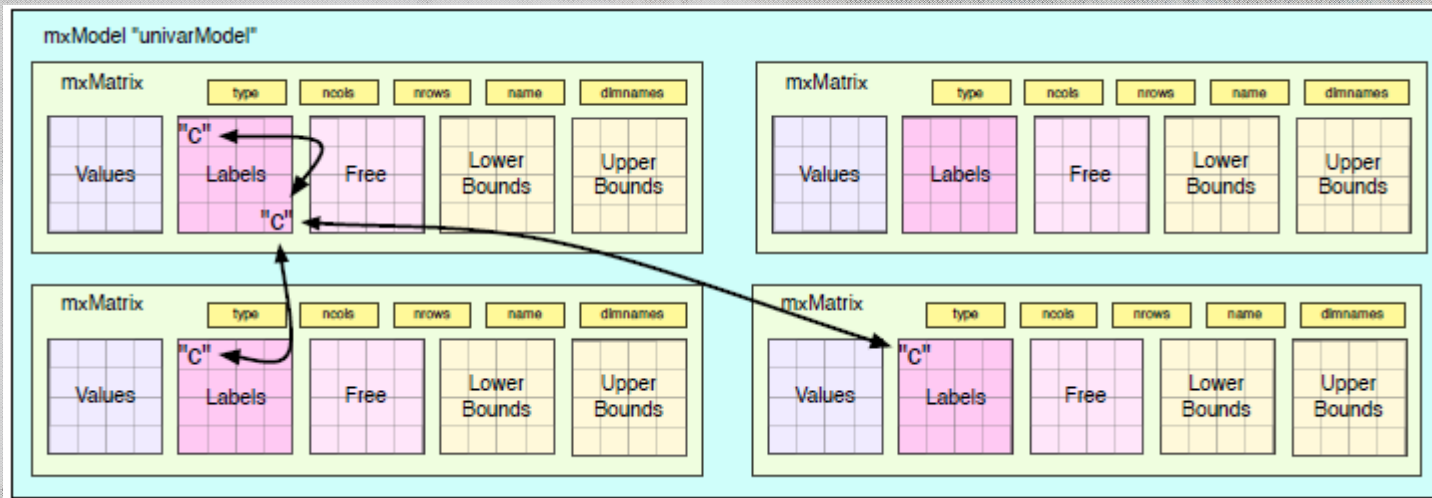
`mxMatrix(type="Full", nrow=2, ncol=4, free=TRUE, name="a")`

```
? ? ? ?  
? ? ? ?
```

A model can have many matrices



Equate parameters using labels



Matrix Operations

Matrix Addition and Subtraction:

- Matrices must be the same size

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

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

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

Dot Product

Also known as the **element-wise product**

OpenMx symbol *****

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

Matrix Multiplication (Star product)

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

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

OpenMx symbol `%*%`

$$\begin{aligned} \mathbf{C} &= \mathbf{A} \times \mathbf{B} \\ \mathbf{C} &= \begin{pmatrix} 3 & 4 & 7 \\ 5 & 6 & 1 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 5 \\ 6 & 2 \end{pmatrix} \\ &= \begin{pmatrix} 3*2 + 4*3 + 7*6 & 3*1 + 4*5 + 7*2 \\ 5*2 + 6*3 + 1*6 & 5*1 + 6*5 + 1*2 \end{pmatrix} = \begin{pmatrix} 60 & 37 \\ 34 & 37 \end{pmatrix} \end{aligned}$$

Kroneker Product

OpenMx symbol %x%

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

Quadratic Product

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

$$A \% \& \% B = ABAT =$$

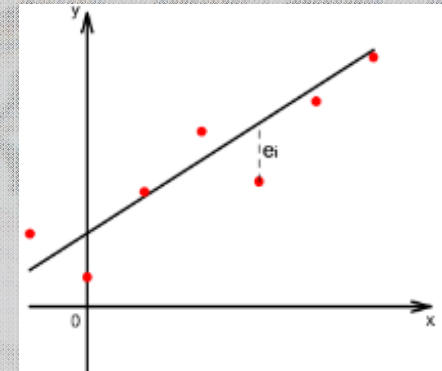
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}$$

Our first OpenMx script

- **Linear regression** (aka simple regression, Ordinary Least Squares Regression, linear model)
- Raw data
- Describe the relationship between two variables, X and Y, as a straight line
- The regression of BMI on variable Age
 - Does age predict BMI?

$$\text{BMI} = \text{weight in kg} / (\text{height in m})^2$$

Linear Regression

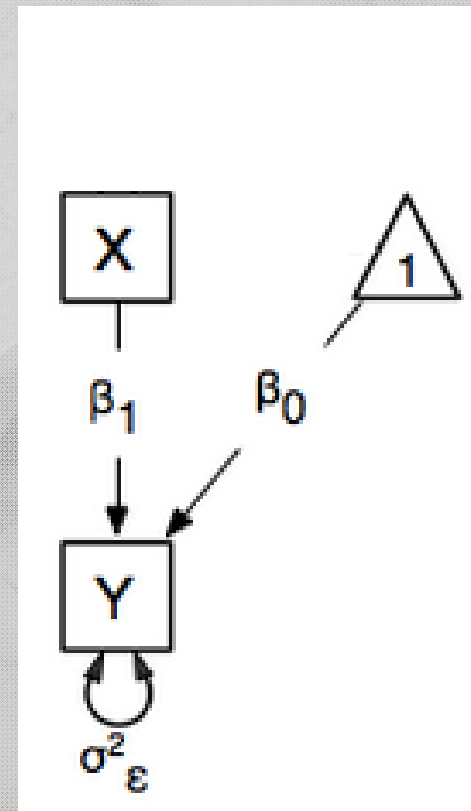
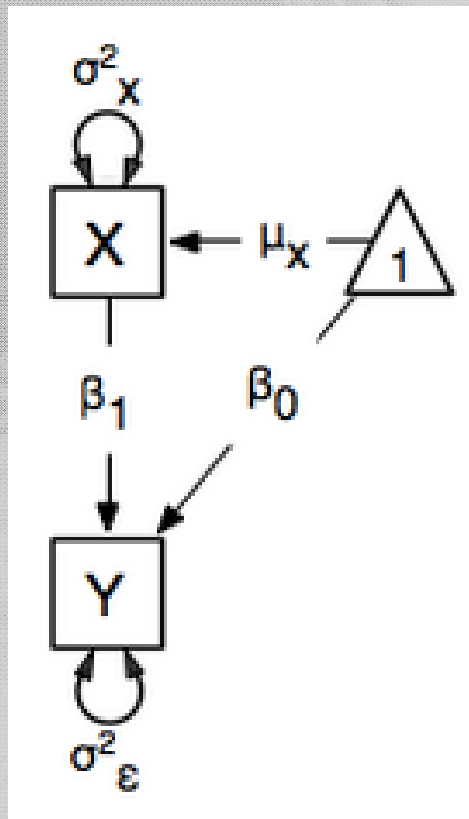


- The regression of variable Y on variable X is given by:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad i = 1, \dots, n$$

- where:
 - β_0 (Intercept) value of y when $x=0$
 - β_1 (Slope) increase in Y for each unit change in X
 - ε_i (Random Error) $\varepsilon_i \sim N(0, \sigma^2)$
 - Linear Function: $\beta_0 + \beta_1 x_i = E(Y | X = x_i)$

As a path diagram



Starting at the beginning...

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

Saving your output

- Numerous options
 - Saving workspace most common but could potentially lead to IRB issues
 - Sink and history both miss sections
- Today we will use the teaching demos package records everything that goes in and comes out
- **require (TeachingDemos)**
- **txtStart(file="linearRegression.omx", commands=TRUE, results=TRUE)**
- **txtStop()**

Getting your data into R

- Open RStudio
- Example data: ozbmi2.txt
- `OZbmi<-read.table("ozbmi2.txt", header=T, na.strings = "NA")`
- `head(data)`

```
> head(data)
  fam agecat  age zyg part wt1 wt2  ht1  ht2  htwt1  htwt2  bmi1  bmi2
1 115      0 0.21  1   2  58  57   1.7   1.7 20.0692 19.7232 20.9943 20.8726
2 121      0 0.24  1   2  54  53 1.6299 1.6299 20.3244 19.9481 21.0828 20.9519
3 158      0 0.21  1   2  55  50 1.6499 1.6799  20.202 17.7154 21.0405  20.121
4 172      0 0.21  1   2  66  76 1.5698 1.6499 26.7759 27.9155 23.0125 23.3043
5 182      0 0.19  1   2  50  48 1.6099 1.6299 19.2894 18.0662 20.7169 20.2583
6 199      0 0.26  1   2  60  60 1.5999 1.5698 23.4375 24.3418 22.0804 22.3454
```

- # using subset function create new dataset without missing data
- `OZbmi <- subset(data, age != "NA" , select=c(bmi1, age))`

Regression using lm

- `BMIfit <- lm(bmi1 ~ age, data=OZbmi)`
- `summary(BMIfit)` # show results
- `coefficients(BMIfit)` # model coefficients

```
> #Running the regression using lm
> BMIfit <- lm(bmi1 ~ age, data=OZbmi)
> summary(BMIfit) # show results

Call:
lm(formula = bmi1 ~ age, data = OZbmi)

Residuals:
    Min       1Q   Median       3Q      Max
-3.4504 -0.5906 -0.0712  0.5188  4.3832

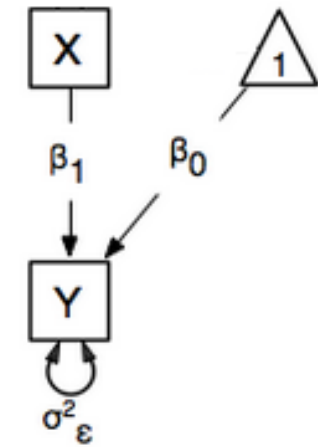
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.01355    0.03912   537.20  <2e-16 ***
age          2.03457    0.10483   19.41  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8944 on 3671 degrees of freedom
(133 observations deleted due to missingness)
Multiple R-squared:  0.09305, Adjusted R-squared:  0.09281
F-statistic: 376.6 on 1 and 3671 DF, p-value: < 2.2e-16

> coefficients(BMIfit) # model coefficients
(Intercept)          age
21.013545         2.034573
```

Regression using OpenMx

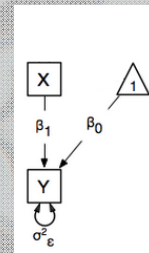
- Model contains:
 - 4 matrices and 3 estimated parameters
 - BMI
 - Free observed variable
 - β_0 and σ^2_ϵ are estimated
 - Age
 - Fixed observed variable
 - Regression of BMI on Age
 - β_1 is estimated



Regression using OpenMx

Variance/Covariance matrix

```
Variance <- mxMatrix(  
  type="Full",  
  nrow=1,  
  ncol=1,  
  free=TRUE,  
  values=11,  
  labels='resid',  
  name="residualVar" )
```



Regression using OpenMx

```
require (OpenMx)
```

```
depVar <- 'bmi1'
```

```
# Variance/Covariance matrix
```

```
Variance <- mxMatrix( type="Full", nrow=1, ncol=1, free=TRUE,  
                      values=11, labels='resid', name="residualVar" )
```

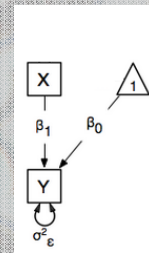
```
# Regression betas
```

```
b0 <-mxMatrix(type="Full", nrow=1, ncol=1, free=T, values=22,  
             labels="beta0", name="Intercept" )
```

```
b1 <-mxMatrix(type="Full", nrow=1, ncol=1, free=T, values=0,  
             labels="beta1", name="bAge" )
```

```
# Independent variable
```

```
x <-mxMatrix(type="Full", nrow=1, ncol=1, free=F,  
            labels="data.age", name="Age" )
```



```
> b0
FullMatrix 'Intercept'

@labels
      [,1]
[1,] "beta0"

@values
      [,1]
[1,]    22

@free
      [,1]
[1,] TRUE

@lbound: No lower bounds assigned.
@ubound: No upper bounds assigned.
```

```
> x
FullMatrix 'Age'

@labels
      [,1]
[1,] "data.age"

@values
      [,1]
[1,]    0

@free: No free parameters.
@lbound: No lower bounds assigned.
@ubound: No upper bounds assigned.
```

Building the model ($y_i = \beta_0 + \beta_1 x_i$)

```
expMean <- mxAlgebra(intercept + bAge%x%Age,  
                      name="regress")
```

```
> expMean  
mxAlgebra 'regress'  
@formula: Intercept + bAge * Age  
@result: (not yet computed) <0 x 0 matrix>  
dimnames: NULL
```

Specify the data

```
regData <- mxData( observed=OZbmi, type="raw" )
```

```
inclusions <- list(Variance, b0, b1, bAge, expMean)
```

Finally, we call up the results of the algebras as the arguments for the expectation function.

The dimnames map the data to the model.

NOTE- The matrix names defined in mxMatrix() statements are used here NOT the objects that are used to store the matrices.

```
exp      <- mxExpectationNormal( covariance="residualVar",  
                                means="regress", dimnames=depVar )
```

```
> exp  
MxExpectationNormal 'expectation'  
$covariance : 'residualVar'  
$means      : 'regress'  
$dims       : 'bmi1'  
$thresholds : NA  
$threshnames : 'bmi1'
```

#The fit function declares that the model is fit using maximum likelihood.

When combined with raw data this means full information maximum likelihood (FIML) is optimized.

```
funML <- mxFitFunctionML()
```

```
> funML
MxFitFunctionML 'fitfunction'
 $vector : FALSE
 $rowDiagnostics : FALSE
 $fellner : NA
 $verbose : 0
<0 x 0 matrix>
```

#Build the model - specify the name of the model, the objects referenced, the data and the objective

```
regModel <- mxModel( "Regression101", inclusions,  
                    regData, exp, funML )
```

```
> regModel  
MxModel 'Regression101'  
type : default  
$matrices : 'residualVar', 'Intercept', 'bAge', and 'Age'  
$algebras : 'regress'  
$constraints : NULL  
$intervals : NULL  
$latentVars : none  
$manifestVars : none  
$data : 3806 x 2  
$data means : NA  
$data type: 'raw'  
$submodels : NULL  
$expectation : MxExpectationNormal  
$fitfunction : MxFitFunctionML  
$compute : NULL  
$independent : FALSE  
$options :  
$output : FALSE  
> |
```

Run the model & summarize output

```
regFit <- mxRun( regModel, intervals=FALSE )
```

```
regSum <- summary( regFit )
```

```

> regFit      <- mxRun( regModel, intervals=FALSE)
Running Regression101 with 3 parameters
>
> ( regSum    <- summary( regFit ) )
Summary of Regression101

free parameters:
  name      matrix  row  col  Estimate  Std.Error A
1 resid residualVar bmi1 bmi1  0.799574  0.01865852
2 beta0    Intercept  1   1  21.013549  0.03911028
3 beta1      bAge    1   1  2.034564  0.10481854

observed statistics:  3673
estimated parameters:  3
degrees of freedom:  3670
fit value ( -2lnL units ):  9601.954
number of observations:  3806
Information Criteria:
      |  df Penalty |  Parameters Penalty |  Sample-Size Adjusted
AIC:      2261.954          9607.954          NA
BIC:      -20654.752        9626.687        9617.155
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: 2016-03-07 17:37:26
Wall clock time (HH:MM:SS.hh): 00:00:01.13
optimizer:  NPSOL
OpenMx version number: 2.5.2
Need help?  See help(mxSummary)

```

Intuitive Logic of Optimization

1. Start with an arbitrary set of initial parameters values.
(Starting Values)
2. Determine a direction of movement for the parameters (larger or smaller)
3. Determine a step length to move (how much larger or smaller)
4. Rinse and repeat until some termination criteria is reached and then stop.

Maximum Likelihood Estimation

Let us assume that X is an i.i.d. random variable with a probability of $p(x | \theta) = P(X=x)$, where θ is a parameter

We know that if observations are independent, then the joint probability of their occurrence is the product of the individual probabilities.

Hence:
$$L(\theta | x) = \prod_{i=1}^n p_X(X_i | \theta)$$

#Looking at the optimization process

ReRunning to look at optimization

```
regModel <- mxOption(regModel,"Checkpoint Units",  
                    "iterations")
```

```
regModel <- mxOption(regModel,"Checkpoint Count", 1)
```

```
regFit <- mxRun( regModel, intervals=FALSE, checkpoint=T )
```

OpenMx vs lm

OpenMx

- regSum\$parameters

```
> regSum$parameters
  name      matrix row col Estimate Std. Error lbound ubound lboundMet uboundMet
1 resid residualVar bmi1 bmi1  0.799574 0.01865790      NA      NA      FALSE      FALSE
2 beta0 Intercept    1    1 21.013549 0.03911028      NA      NA      FALSE      FALSE
3 beta1      bAge    1    1  2.034564 0.10481854      NA      NA      FALSE      FALSE
```

Lm

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.01355    0.03912   537.20  <2e-16 ***
age          2.03457    0.10483   19.41  <2e-16 ***
```

- Is beta1 different from 0?

Go back and pickup the model so that we can run significance tests

```
ageEfModel <- regFit
```

```
#set beta 1 to 0
```

```
ageEfModel <- omxSetParameters( ageEfModel, label="beta1",  
                                free=FALSE, values=0 )
```

```
ageEfFit <- mxRun(ageEfModel, intervals=FALSE)
```

```
(ageEfSumm <- summary(ageEfFit))
```

```
# difference in fit
```

```
deltaLL <-ageEfSumm$Minus2LogLikelihood -  
  regSum$Minus2LogLikelihood
```

```
# difference in df
```

```
deltaDF <-ageEfSumm$degreesOfFreedom -  
  regSum$degreesOfFreedom
```

```
# significance test
```

```
pchisq(deltaLL, lower.tail=F, deltaDF)
```

Questions?



Bus shelter on the road to
Sintra (Portugal)