```
*******************************************************************************
*  A unified multi-level model approach to assessing patient responsiveness
*  including; return to normal, minimally important differences, and minimally
*  clinical important differences for patient reported outcome measures.
*******************************************************************************
*
*  Sayers A*, Wylde V, Lenguerrand E, Gooberman-Hill R, Dawson J, Beard D,
*  Price A, Blom AW
*
*  1.   Musculoskeletal Research Unit, School of Clinical Sciences,
*       University of Bristol, Southmead Hospital, Westbury On Trym,
*              Bristol, BS10 5NB.
*  2.   Nuffield Department of Population Health, University of Oxford,
*              Old Road Campus, Headington, Oxford OX3 7LF, UK
*  3.   Biomedical Research Unit, Nuffield Department of Orthopaedics,
*              Rheumatology and Musculoskeletal Science, Nuffield Orthopaedic Centre,
*              Windmill Road, Oxford, OX2 9JA
*
*      Address for Correspondence
*   Adrian Sayers, Musculoskeletal Research Unit, School of Clinical Sciences,
*   University of Bristol, Learning and Research Building (Level 1),
*   Southmead Hospital, Westbury on Trym, Bristol, BS10 5NB
*
*
*   E-mail: adrian.sayers@bristol.ac.uk
*   Tel: 44 (0)117 4147880; Fax + 44(0)117 414 7924
*******************************************************************************
* Abstract
* Stata code to illustrate calculation of patient reponsiveness using existing
* and multi-level model methods.
* Do file should be run comlpletely inorder to simulate data from a linear model
* and perform calculations.
* File requires MLWin and copy of runmlwin downloaded for Stata.

*******************************************************************************
* 1. Simulate a dataset
*******************************************************************************
{
* Design matrix in OO Format
set seed 111
clear
set obs 100
gen id= _n

* Set Parameters values
        * Set Fixed Effect Parameters
        local b0 = 49.19
                local b1 = 44.35 / 3
                        local b2 = 39.12
```

```
                            local b3 = 47.66 / 3
* Set Random Effect Standard Deviations & Correlation Matrix
local u0 = 25.3
       local u1 = 23.4 / 3
              local u2 = 18.5
                     local u3 = 19.1 / 3
                            matrix u = (`u0',`u1',`u2',`u3')'
                                   matrix u_corr = (1   ,0.3 ,0.1 ,0.1 \  ///
                                                    0.3 ,1   ,0.1 ,0.1 \  ///
                                                    0.1 ,0.1 ,1   ,0.3 \  ///
                                                    0.1 ,0.1 ,0.3 ,1   )
* Draw Random Parameters
       drawnorm u0 u1 u2 u3 , sds(u) corr(u_corr)


* Create 4 measurement occassions
expand 4
by id , sort : gen t = _n-1

* Prepare for a reshape into double long
gen _1= 1
       gen _2= 1
              reshape long _ , i(id t) j(resp)
                     drop _

* Set error Standard Deviations & Correlation Matrix
local e1= 5
       local e2= 5
              matrix e = (`e1', `e2')'
                     matrix e_corr = (1   ,0.1 \  ///
                                      0.1 ,1   )  //

                            drawnorm e1 e2 , sds(e) corr(e_corr)


* Create response indicators for OO
gen w1 = 1 if resp==1
       replace w1 = 0 if resp==2
              gen w2 = 0 if resp==1
                     replace w2 = 1 if resp==2

* Generate a satisfaction indicator, uncorrelated with effects just for illustration
gen x = cond(uniform()>=0.3,1,0) if resp==1 & t==1
       by id : egen _x = min(x)
              *Create dummy variables
                     gen x1 = 1 if _x==1
                            replace x1 = 0 if _x==0
                                   gen x2 = 0 if _x==1
                                          replace x2 = 1 if _x==0
```

```
                                                drop x _x


* Predict response
gen y = (`b0' + u0)* w1 + (`b1' + u1)* w1 * t + e1* w1 + ///
               (`b2' + u2)* w2 + (`b3' + u3)* w2 * t + e2* w2 //

tempfile simdata
       save `simdata' , replace

}
*******************************************************************************
* 2.1 Existing Methods (n.b. only for first response)
*******************************************************************************
use  `simdata' , clear
       * Working with the first and last measurment occassion
              keep if  t ==0 | t==3
                     sort id resp t
                            by id resp : gen d_y = y[_n] - y[_n-1]
*********************
* 2.1.1 Existing RTN
*********************
{
       sum y if t==0 & resp==1
              local rtn = r(mean) + 2*r(sd)
                     by id resp: gen ex_rtn =cond(y>=`rtn',1 ,0) if _n==2 & resp==1
                            by id resp: gen ex_rci = cond((d_y / sqrt(2*(`r(sd)' * sqrt(1-0.9))^2))>=1.96,1,0) if _n==2 & resp==1
                                   by id resp: gen ex_rtn_rci = cond(ex_rtn==1 & ex_rci==1 ,1,0) if _n==2 & resp==1

                                   tab ex_rtn if resp==1         // Number of individuals returning to normal
                                   tab ex_rci if resp==1         // Number of individuals significant change
                                   tab ex_rtn_rci if resp==1     // Number of individuals significant change & returning to normal
                                   }
*********************
* 2.1.2 Existing MID
*********************
{
sum y if t==0 & resp==1
       local mid = r(sd)*0.5
              by id resp : gen ex_mid =cond(d_y>=`mid',1,0)  if _n==2 & resp==1

                     tab ex_mid if resp==1         // Number of individuals with minimally important difference
}
*********************
* 2.1.3 Existing MCID
* n.b using the 25th centile is pain is reverse coded.
*********************
{
centile d_y if resp==1 & x1==1 , c(25)
```

```
        local mcid = r(c_1)
                by id resp: gen ex_mcid = cond(d_y>=`mcid',1,0) if _n==2 & resp==1

                        tab ex_mcid if resp==1        // Number of individuals meeting the MCID criteria
}

*******************************
* 2.1.4 Existing (OO) OMERACT-OARSI
*******************************
{
* 50% relative, 20% absolute single
* 20% relative, 10% absolute both

* Calculate Relative Change
        by id resp: gen d_rely= (d_y/y[_n-1])*100
                * Mark Single Changes
                        by id resp: gen ex_oo_single =1 if (d_y>=20 & d_y<.) | (d_rely>=50 & d_rely<.) & _n==2
                                * Mark Double Changes
                                        by id resp: gen ex_oo_double =1 if (d_y>=10 & d_y<.) | (d_rely>=20 & d_rely<.) & _n==2
                                                * Sum double changes
                                                        by id  : egen ex_oo_double_sum = total(ex_oo_double) if  d_y!=.

* Mark OO criteria
        by id : gen _ex_oo = cond(ex_oo_single==1 | ex_oo_double_sum==2 , 1,0) if d_y!=.
                by id : egen ex_oo = max(_ex_oo) if d_y!=.

                tab ex_oo if resp==1   // Number of individuals meeting the oo criteria
}
*******************************************************************************
* 2.2 Multi-level Methods
*******************************************************************************
// Set the global macro to identify the location and version of mlwin
global MLwiN_path "C:\Program Files (x86)\MLwiN v2.36\i386\MLwiN.exe"
        use  `simdata' , clear
                keep if resp==1
* Create a constant
gen cons=1

******************
* 2.2.1 MLM RTN / MID Model
******************
{
*                       0-----------------1
*                       1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,
matrix a = (1,1,1)

runmlwin y cons t if resp==1 ,                                                          /// Fixed effect
        level1(t: cons, residuals(_e, )  )                                  /// Level 1 variance
                level2(id: cons t, elements(a) residuals(_u, ) )     /// Level 2 varaince
```

```
                      maxiterations(10)  corr sd nopause                          //      Modelling options

* Predict Individual effects
gen xb_fe =  _b[cons] + _b[t]*t
       gen xb_re =  _u0 + _u1*t
              gen xb = xb_fe + xb_re

* Predict to asses responsiveness at (3month)
gen xb_t =  (_b[cons]+_u0) + (_b[t]+_u1)*3

* RTN threshold
local mlm_rtn = _b[FP1:cons] + 2*(_b[RP2:var(cons)]^0.5)

* Mark RTN
gen mlm_rtn = cond(xb_t>=`mlm_rtn',1,0)

* Calculate RCI
       gen xb_d = _b[FP1:t] + _u1
              gen se_d = (_se[FP1:t]^2 +_u1se^2)^0.5
                     gen z_d = xb_d / se_d

* Mark RCI
gen mlm_rci = cond(z_d>=1.96,1,0)

* Mark RTN RCI composite
gen mlm_rtn_rci = cond(mlm_rtn==1 & mlm_rci==1, 1, 0)
       egen pickone = tag(id)

              tab mlm_rtn_rci if pickone==1 // Number of individuals meeting the MLM RTN RCI  criteria
}
**********************
* 2.2.2 MLM MID
**********************
{
* MID Threshold @ 3 months
local mlm_mid = 0.5*(_b[RP2:var(cons)]^0.5)
       gen mlm_mid = cond( (_b[t]+_u1)*3>= `mlm_mid' ,1 ,0 )
              tab mlm_mid if pickone==1 // Number of individuals meeting the MLM MID criteria

* Drop previous residual and predictions
drop  _u0 _u1 _u0se _u1se _e0 _e0se  xb_fe xb_re xb xb_t xb_d se_d z_d
}
********************
* 2.2.3  MLM MCID
********************
{
* Stratify intercept and slope by satisfaction
gen consx1= cons*x1
       gen consx2 = cons*x2
```

```
               gen tx1 = t*x1
                       gen tx2 = t*x2

* Specify RE variance matrix
*                      0-----------------1-------------------2
*                      1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1
matrix u = (1,1,1,0,0,1,0,0,1,1)

* Specify RE variance matrix
*                      0-----------------1-------------------2
*                      1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1
matrix e = (1,0,1)

runmlwin y consx1 tx1 consx2 tx2 if resp==1 ,                                                                   /// Fixed effect
       level1(t: consx1 consx2, elements(e)  residuals(_e, norecode ))                          /// Level 1 variance
               level2(id: consx1 tx1 consx2 tx2 ,elements(u)  residuals(_u, norecode ))   /// Level 2 varaince
                       maxiterations(10)  corr sd nopause                                                                      //      Modelling
options

* Estimate the Change  for all individuals
gen xb_slope = (_b[tx1]+_u1)*x1 + (_b[tx2]+_u3)*tx2

* Find the 75th (inverse coding 25th) centile of those satisfied
centile xb_slope if tx1==3 , c(25)
       local mlm_mcid = r(c_1)

*tag observations which have improvements greater than mcid
       gen mlm_mcid = cond(xb_slope>=`mlm_mcid',1,0) if t==3
               tab mlm_mcid if t==3 // Number of individuals meeting the MCID  criteria
       }
*********************
* 2.2.4  MLM (OO) OMERACT-OARSI
*********************
{
* 50% relative, 20 absolute single assuming a 0-100 score
* 20% relative, 10 absolute both assuming a 0-100 score
use  `simdata' , clear
       sort id t resp


* Create response indicators
gen cons =1
       gen consw1 = cons*w1
               gen consw2 = cons*w2
                       gen tw1 = t*w1
                              gen tw2 = t*w2

runmlwin y  consw1 tw1 consw2  tw2 ,                                                                    /// Fixed Effect
       level1(resp:)                                                                                                   /// Level 1 variance
```

```
        level2(t: consw1 consw2,  residuals(_e, norecode ))                              /// Level 2 variance
                level3(id: consw1 tw1 consw2 tw2 , residuals(_u, norecode ))        /// Level 3 varaince
                        maxiterations(10)  corr sd nopause                                                      //      Modelling options

* Calculate predicted changes
gen mlm_d = (_b[tw1] + _u1 )*tw1  + (_b[tw2] + _u3 )*tw2
        gen mlm_bl = (_b[consw1] + _u0)*consw1 + (_b[consw2] + _u2)*consw2
                gen mlm_relyd=  (mlm_d /mlm_bl)*100

* Mark out responders
                        by id resp ,sort: gen mlm_oo_single =1 if (( mlm_d>=20 & mlm_d<.) | (mlm_relyd>=50 & mlm_relyd<.)) & t==3
                                * Mark Double Changes
                                        by id resp ,sort: gen mlm_oo_double =1 if ((mlm_d>=10 & mlm_d<.) | (mlm_relyd>=20 & mlm_relyd<.)) & t==3
                                                * Sum double changes
                                                        by id ,sort  : egen mlm_oo_double_sum = total(mlm_oo_double) if t==3

* Mark OO criteria
by id : gen _mlm_oo = cond(mlm_oo_single==1 | mlm_oo_double_sum==2 , 1,0) if t==3
        by id : egen mlm_oo = max(_mlm_oo) if t==3

                tab mlm_oo if resp==1  // Number of individuals meeting the MLM OO criteria
}
```