Example 6b: Modeling Change over Time Using Piecewise Trends
(complete data, syntax, and output available for SAS, STATA, and R electronically)

These data for these example models come from Hoffman (2015) chapter 6 (which also provides an example results section for most of these models). We will be examining change in response time (RT) in milliseconds over six practice sessions (i.e., balanced time) to a measure of processing speed in a sample of 101 older adults. This Example 6b builds on an empty means, random intercept model (as shown in Example 6a) to examine piecewise slopes for initial change and later change. Stay tuned for "truly" nonlinear exponential and latent basis models (Example 6c) and the use of log-transformed time to approximate a truly nonlinear exponential model (Example 6d).

STATA Syntax for Data Import and Manipulation:

```
// Define working directory for file location
   cd "C:\Dropbox\24_PSQF6271\PSQF6271_Example6"

// Import Example 6 six-occasion long-format data from excel
   clear // clear memory in case a dataset is already open
   import excel "Example6_Data.xlsx", firstrow case(preserve) sheet("Example6") clear

// Create predictors for piecewise time models assuming BALANCED DATA
// slope12 and slope26 for 2 Direct Slopes model, time for Slope + Deviation Slope model
// Both models have intercept at session 1, breakpoint at session 2
   gen slope12 = session
   recode slope12 (1=0) if session==1
   recode slope12 (2=1) if session==2
   recode slope12 (3=1) if session==3
   recode slope12 (4=1) if session==4
   recode slope12 (5=1) if session==5
   recode slope12 (6=1) if session==6
   gen slope26 = session
   recode slope26 (1=0) if session==1
   recode slope26 (2=0) if session==2
   recode slope26 (3=1) if session==3
   recode slope26 (4=2) if session==4
   recode slope26 (5=3) if session==5
   recode slope26 (6=4) if session==6
   label variable slope12 "slope12: Early Practice Slope (Session 1-2)"
   label variable slope26 "slope26: Later Practice Slope (Session 2-6)"
   gen time=session-1
   label variable time "time: Session (0=1)"

// Create session dummy codes for testing means model absolute fit in REML
   gen s4=0
   gen s5=0
   gen s6=0
   replace s4=1 if session==4
   replace s5=1 if session==5
   replace s6=1 if session==6
```

R Syntax for Data Import and Manipulation (after loading 4 custom functions and packages *readxl*, *TeachingDemos*, *nlme*, and *lmerTest*):

```
# Set working directory (to import and export files to)
setwd("C:/Dropbox/24_PSQF6271/PSQF6271_Example6")

# Import Example 6 six-occasion long-format data from excel -- path = file name
Example6 = read_excel(path="Example6_Data.xlsx", sheet="Example6")
# Convert to data frame to use for analysis
Example6 = as.data.frame(Example6)
# Sort by person and occasion (needed for correct R or V matrix)
Example6 = Example6[order(Example6$PersonID, Example6$session), ]
```

```
# Create predictors for piecewise time models assuming BALANCED DATA
# slope12 and slope26 for 2 Direct Slopes model, time for Slope + Deviation Slope model
# Both models have intercept at session 1, breakpoint at session 2
Example6$slope12=Example6$session
Example6$slope12[which(Example6$session==1)]=0
Example6$slope12[which(Example6$session==2)]=1
Example6$slope12[which(Example6$session==3)]=1
Example6$slope12[which(Example6$session==4)]=1
Example6$slope12[which(Example6$session==5)]=1
Example6$slope12[which(Example6$session==6)]=1
Example6$slope26=Example6$session
Example6$slope26[which(Example6$session==1)]=0
Example6$slope26[which(Example6$session==2)]=0
Example6$slope26[which(Example6$session==3)]=1
Example6$slope26[which(Example6$session==4)]=2
Example6$slope26[which(Example6$session==5)]=3
Example6$slope26[which(Example6$session==6)]=4

# Center time predictor for polynomial time models
Example6$time=Example6$session-1

# Create session dummy codes for testing means model absolute fit in REML
Example6$s4=0
Example6$s5=0
Example6$s6=0
Example6$s4[which(Example6$session==4)]=1
Example6$s5[which(Example6$session==5)]=1
Example6$s6[which(Example6$session==6)]=1

# Save number of occasions per person for use later
Ntimes = 6
# Save total number of observations for use later
Ntotal = 606
```

Model 4a: Fixed Slope12, Fixed Slope26, Random Intercept Model

Level 1: $y_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Slope12}_{ti}\right) + \beta_{2i}\left(\text{Slope26}_{ti}\right) + e_{ti}$

Level 2: Intercept: $\beta_{0i} = \gamma_{00} + U_{0i}$

Slope12: $\beta_{1i} = \gamma_{10}$

Slope26: $\beta_{2i} = \gamma_{20}$

Session	1	2	3	4	5	6
Early Practice → Slope12 =	0	1	1	1	1	1
Later Practice → Slope26 =	0	0	1	2	3	4

```
display "STATA Ch 6: 4a: Fixed Slope12, Fixed Slope26, Random Intercept Model"
mixed rt c.slope12 c.slope26, || PersonID: , reml nolog  ///
      residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitFix12Fix26                // Save for LRT
matrix Fix12Fix26 = r(table)                 // Save results for computations below
display "-2LL = " e(ll)*-2                   // Print -2LL for model

// Get conditional mean per occasion from values of time predictors, also slope difference
lincom _cons*1 + c.slope12*0 + c.slope26*0   // Intercept at Session=1
lincom _cons*1 + c.slope12*1 + c.slope26*0   // Intercept at Session=2
lincom _cons*1 + c.slope12*1 + c.slope26*1   // Intercept at Session=3
lincom _cons*1 + c.slope12*1 + c.slope26*2   // Intercept at Session=4
lincom _cons*1 + c.slope12*1 + c.slope26*3   // Intercept at Session=5
lincom _cons*1 + c.slope12*1 + c.slope26*4   // Intercept at Session=6
lincom c.slope12*-1 + c.slope26*1            // Difference of fixed slope12 vs slope26

matrix list Fix12Fix26                                  // Show saved results
// Variances are stored as log of SD instead
global Fix12Fix26ResVar = exp(Fix12Fix26[1,5])^2       // Save as L1 residual variance for pseudo-R2
display "Pseudo-R2 for L1 Residual Variance = " 1-($Fix12Fix26ResVar/$EmptyResVar)
```

```
print("R Ch 6 4a: Fixed Slope12, Fixed Slope26, Random Intercept Model")
Fix12Fix26 = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                  formula=rt~1+slope12+slope26+(1|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(Fix12Fix26, chkREML=FALSE); summary(Fix12Fix26, ddf="Satterthwaite")
```

```
$AICtab
      AIC       BIC    logLik  deviance  df.resid
 8392.687  8414.722 -4191.344  8382.687   601.000   → deviance = -2LL for model

Random effects:
 Groups   Name        Variance Std.Dev.
 PersonID (Intercept) 202684   450.2    Var(U0)
 Residual              34098   184.7    Var(e)

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  1961.89      48.42  128.66  40.519  < 2e-16  gamma00
slope12      -163.64      23.24  503.00  -7.041 6.29e-12  gamma10
slope26       -32.89       5.81  503.00  -5.661 2.53e-08  gamma20
```

```
print("Get conditional mean per occasion from values of time predictor, also slope difference")
print("Intercept at Session=1 Time=0"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(1,0,0))
print("Intercept at Session=2 Time=1"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(1,1,0))
print("Intercept at Session=3 Time=2"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(1,1,1))
print("Intercept at Session=4 Time=3"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(1,1,2))
print("Intercept at Session=5 Time=4"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(1,1,3))
print("Intercept at Session=6 Time=5"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(1,1,4))
print("Diff of fixed slope12 vs slope26"); contest1D(Fix12Fix26, ddf="Satterthwaite", L=c(0,-1,1))
```

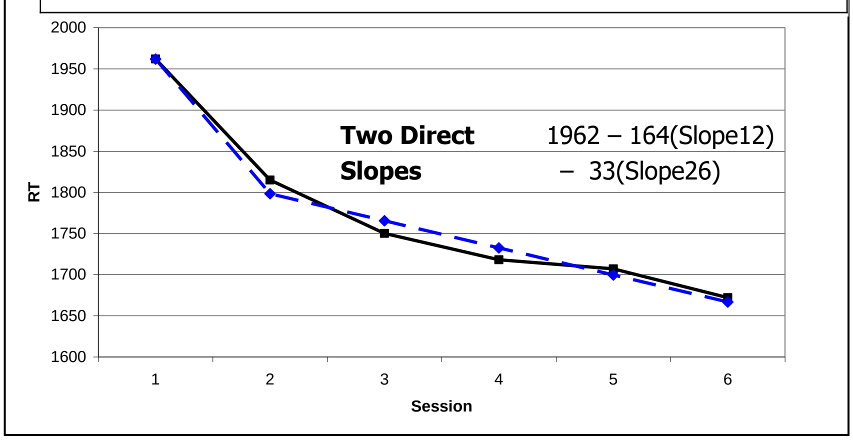
Estimates (from SAS for better organization)

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	g = gamma fixed effect
Intercept at Session=1 Time=0	1961.89	48.4187	129	40.52	<.0001	g00(1)+ g10(0)+ g20(0)
Intercept at Session=2 Time=1	1798.25	47.0035	115	38.26	<.0001	g00(1)+ g10(1)+ g20(0)
Intercept at Session=3 Time=2	1765.36	45.9134	104	38.45	<.0001	g00(1)+ g10(1)+ g20(1)
Intercept at Session=4 Time=3	1732.46	45.5443	101	38.04	<.0001	g00(1)+ g10(1)+ g20(2)
Intercept at Session=5 Time=4	1699.57	45.9134	104	37.02	<.0001	g00(1)+ g10(1)+ g20(3)
Intercept at Session=6 Time=5	1666.68	47.0035	115	35.46	<.0001	g00(1)+ g10(1)+ g20(4)
Difference of slope12 vs slope26	130.75	26.6265	503	4.91	<.0001	g20 – g10

```
print("Psuedo-R2 for variance accounted for by fixed piecewise time")
pseudoRSquaredinator(smallerModel=EmptyRI, largerModel=Fix12Fix26)
```

```
Pseudo R2 Estimates
R2 Random.(Intercept):
-0.00897
R2 L1.Residual.Variance:
0.24058
```

Session	1	2	3	4	5	6
Early Practice → Slope12 =	0	1	1	1	1	1
Later Practice → Slope26 =	0	0	1	2	3	4



Model 4b: Random Slope12, Fixed Slope26 Model

Level 1: $y_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Slope12}_{ti}\right) + \beta_{2i}\left(\text{Slope26}_{ti}\right) + e_{ti}$

Level 2: Intercept: $\beta_{0i} = \gamma_{00} + U_{0i}$

Slope12: $\beta_{1i} = \gamma_{10} + U_{1i}$

Slope26: $\beta_{2i} = \gamma_{20}$

```
display "STATA Ch 6: 4b: Random Slope12, Fixed Slope26 Model"
mixed rt c.slope12 c.slope26, || PersonID: slope12, reml nolog covariance(unstructured)  ///
      residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitRand12Fix26                        // Save for LRT

------------------------------------------------------------------
          rt |      Coef.   Std. Err.          DF       t    P>|t|
-------------+----------------------------------------------------
     slope12 |   -163.644   31.24619        122.6    -5.24   0.000
     slope26 |  -32.89317   4.891648        403.0    -6.72   0.000
       _cons |   1961.893    54.6805        100.0    35.88   0.000
------------------------------------------------------------------
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
PersonID: Unstructured       |
                var(slope12) |   59940.74    12743.1       39514.9       90925
                  var(_cons) |   277818.2   42741.21      205497.6    375590.5
        cov(slope12,_cons) |  -69063.12   18931.72     -106168.6   -31957.64
-----------------------------+------------------------------------------------
               var(Residual) |    24167.5   1702.528      21050.73    27745.74
------------------------------------------------------------------------------
LR test vs. linear model: chi2(3) = 868.87                Prob > chi2 = 0.0000

display "-2LL = " e(ll)*-2                            // Print -2LL for model
-2LL = 8319.6133

estat recovariance, relevel(PersonID) correlation // GCORR matrix

Random-effects correlation matrix for level PersonID
             |   slope12      _cons
-------------+----------------------
     slope12 |         1
       _cons | -.5351861          1

estat wcorrelation, covariance                        // V matrix
(printed in scientific notation in STATA; R version given below)

         1        2        3        4        5        6
1 301985.7 208754.9 208754.9 208754.9 208754.9 208754.9
2 208754.9 223799.9 199632.4 199632.4 199632.4 199632.4
3 208754.9 199632.4 223799.9 199632.4 199632.4 199632.4
4 208754.9 199632.4 199632.4 223799.9 199632.4 199632.4
5 208754.9 199632.4 199632.4 199632.4 223799.9 199632.4
6 208754.9 199632.4 199632.4 199632.4 199632.4 223799.9
```

How would we describe the marginal pattern of variance and covariances in the **V** matrix now?

```
estat wcorrelation                                    // VCORR matrix

         obs |      1        2        3        4        5        6
-------------+----------------------------------------------------
           1 |  1.000
           2 |  0.803    1.000
           3 |  0.803    0.892    1.000
           4 |  0.803    0.892    0.892    1.000
           5 |  0.803    0.892    0.892    0.892    1.000
           6 |  0.803    0.892    0.892    0.892    0.892    1.000
```

```
lrtest FitRand12Fix26 FitFix12Fix26   // LRT for random slope12 variance and covariance

Likelihood-ratio test                                 LR chi2(2)  =     63.07
(Assumption: FitFix12Fix26 nested in FitRand12Fix26)  Prob > chi2 =    0.0000


print("R Ch 6 4b: Random Slope12, Fixed Slope26 Model")
Rand12Fix26 = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                   formula=rt~1+slope12+slope26+(1+slope12|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(Rand12Fix26, chkREML=FALSE); summary(Rand12Fix26, ddf="Satterthwaite")
print("Use custom function to get predicted V matrix"); PrintV(Rand12Fix26)
print("LRT for random slope12 variance and covariance"); ranova(Rand12Fix26, reduce.term=TRUE)
```

Model 4c: Random Slope12, Random Slope26 Model

Level 1: $y_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Slope12}_{ti}\right) + \beta_{2i}\left(\text{Slope26}_{ti}\right) + e_{ti}$

Level 2: Intercept: $\beta_{0i} = \gamma_{00} + U_{0i}$

Slope12: $\beta_{1i} = \gamma_{10} + U_{1i}$

Slope26: $\beta_{2i} = \gamma_{20} + U_{2i}$

```
display "STATA Ch 6: 4c: Random Slope12, Random Slope26 Model"
mixed rt c.slope12 c.slope26, || PersonID: slope12 slope26,  ///
      reml nolog covariance(unstructured)                    ///
      residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitRand12Rand26                        // Save for LRT
matrix Rand12Rand26 = r(table)                         // Save results for computations below
display "-2LL = " e(ll)*-2                             // Print -2LL for model
estat recovariance, relevel(PersonID) correlation // GCORR matrix
estat wcorrelation, covariance                         // V matrix
estat wcorrelation                                     // VCORR matrix
lrtest FitRand12Rand26 FitRand12Fix26   // LRT for random slope26 variance and covariance

matrix list Rand12Rand26                // Show saved results
// Save fixed effects and variances for computations below
global FixInt = Rand12Rand26[1,3]          // Save fixed intercept
global FixS12 = Rand12Rand26[1,1]          // Save fixed slope12
global FixS26 = Rand12Rand26[1,2]          // Save fixed slope26
global IntVar = exp(Rand12Rand26[1,6])^2   // Save L2 random intercept variance
global S12Var = exp(Rand12Rand26[1,4])^2   // Save L2 random slope16 variance
global S26Var = exp(Rand12Rand26[1,5])^2   // Save L2 random slope26 variance
// Check if correct saved variances
display $IntVar
display $S12Var
display $S26Var
display "STATA 95% Random Intercept CI"
display "Lower = " $FixInt - 1.96*sqrt($IntVar)
display "Upper = " $FixInt + 1.96*sqrt($IntVar)
display "STATA 95% Random Slope12 CI"
display "Lower = " $FixS12 - 1.96*sqrt($S12Var)
display "Upper = " $FixS12 + 1.96*sqrt($S12Var)
display "STATA 95% Random Slope26 CI"
display "Lower = " $FixS26 - 1.96*sqrt($S26Var)
display "Upper = " $FixS26 + 1.96*sqrt($S26Var)


print("R Ch 6 4c: Random Slope12, Random Slope26 Model -- reports convergence problem")
Rand12Rand26 = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                    formula=rt~1+slope12+slope26+(1+slope12+slope26|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(Rand12Rand26, chkREML=FALSE); summary(Rand12Rand26, ddf="Satterthwaite")
```

```
$AICtab
      AIC       BIC    logLik  deviance  df.resid
 8295.374  8339.443 -4137.687  8275.374   596.000

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 PersonID (Intercept) 284311   533.21
          slope12      63954   252.89   -0.40
          slope26       2617    51.16   -0.39 -0.13
 Residual              17673   132.94

Fixed effects:
            Estimate Std. Error       df t value    Pr(>|t|)
(Intercept) 1961.893     54.680  100.000  35.879     < 2e-16
slope12     -163.644     30.219  100.000  -5.415 0.000000422
slope26      -32.893      6.589  100.000  -4.992 0.000002529
```

```
print("Use custom function to get predicted V matrix"); PrintV(Rand12Rand26)
```

```
         1        2        3        4        5        6
1 301984.5 230041.6 219397.9 208754.1 198110.3 187466.6
2 230041.6 257399.0 227409.9 215093.9 202777.8 190461.7
3 219397.9 227409.9 235384.2 208012.4 198313.6 188614.8
4 208754.1 215093.9 208012.4 218603.9 193849.4 186767.9
5 198110.3 202777.8 198313.6 193849.4 207058.2 184920.9
6 187466.6 190461.7 188614.8 186767.9 184920.9 200747.0
```

The marginal V matrix now predicts the variances to change in a **quadratic pattern (time^2) within each piece separately.**

```
print("LRT for random slope26 variance and covariances"); ranova(Rand12Rand26, reduce.term=TRUE)
```

```
                                          npar  logLik    AIC    LRT Df Pr(>Chisq)
<none>                                      10 -4137.7 8295.4
slope12 in (1 + slope12 + slope26 | PersonID)  7 -4173.6 8361.1 71.727  3  1.821e-15
slope26 in (1 + slope12 + slope26 | PersonID)  7 -4159.8 8333.6 44.239  3  1.343e-09
```

95% Random Effect Confidence Intervals that describe the *predicted* range of *individual* random effects:

$$\text{Random Effect 95\% CI} = \text{fixed effect} \pm \left(1.96 * \sqrt{\text{Random Variance}}\right)$$

$$\text{Intercept 95\% CI} = \gamma_{00} \pm \left(1.96 * \sqrt{\tau^2_{U_0}}\right) \rightarrow 1{,}961.9 \pm \left(1.96 * \sqrt{284{,}312}\right) = 917 \text{ to } 3{,}007$$

$$\text{Slope12 95\% CI} = \gamma_{10} \pm \left(1.96 * \sqrt{\tau^2_{U_1}}\right) \rightarrow -163.6 \pm \left(1.96 * \sqrt{63{,}954}\right) = -659 \text{ to } 322$$

$$\text{Slope26 95\% CI} = \gamma_{20} \pm \left(1.96 * \sqrt{\tau^2_{U_2}}\right) \rightarrow -32.9 \pm \left(1.96 * \sqrt{2{,}617}\right) = -133 \text{ to } 67$$

```
# Get ingredients for 95% random effect confidence intervals
# Print each object first to see which row and column values to extract
as.data.frame(fixef(Rand12Rand26)); as.data.frame(VarCorr(Rand12Rand26))
```

```
            fixef(Rand12Rand26)
(Intercept)          1961.89337
slope12              -163.64399
slope26               -32.89317

       grp        var1    var2        vcov       sdcor
1 PersonID (Intercept)    <NA> 284311.486 533.2086701
2 PersonID     slope12    <NA>  63954.180 252.8916367
3 PersonID     slope26    <NA>   2617.280  51.1593622
4 PersonID (Intercept) slope12 -54269.854  -0.4024639
5 PersonID (Intercept) slope26 -10643.761  -0.3901870
6 PersonID     slope12 slope26  -1672.293  -0.1292566
7 Residual        <NA>    <NA>  17673.032 132.9399568
```

```
# Save fixed effects and variances for computations below
FixInt = as.data.frame(fixef(Rand12Rand26))[1,1]     # Save fixed intercept
FixS12 = as.data.frame(fixef(Rand12Rand26))[2,1]     # Save fixed slope12
FixS26 = as.data.frame(fixef(Rand12Rand26))[3,1]     # Save fixed slope26
IntVar = as.data.frame(VarCorr(Rand12Rand26))[1,4]  # Save L2 random intercept variance
S12Var = as.data.frame(VarCorr(Rand12Rand26))[2,4]  # Save L2 random slope12 variance
S26Var = as.data.frame(VarCorr(Rand12Rand26))[3,4]  # Save L2 random slope26 variance
print("R 95% Random Intercept Confidence Interval")
print("Lower = "); FixInt - 1.96*sqrt(IntVar)
print("Upper = "); FixInt + 1.96*sqrt(IntVar)
print("R 95% Random Slope12 Confidence Interval")
print("Lower = "); FixS12 - 1.96*sqrt(S12Var)
print("Upper = "); FixS12 + 1.96*sqrt(S12Var)
print("R 95% Random Slope26 Confidence Interval")
print("Lower = "); FixS26 - 1.96*sqrt(S26Var)
print("Upper = "); FixS26 + 1.96*sqrt(S26Var)
```

So far we've examined one way to fit piecewise slopes models—two direct slopes that represent the change during each time period. Let's now examine an alternative specification—**slope + deviation slope**, which can be useful in examining individual differences in **differential change between time periods**.

Model 5a: Fixed Slope, Fixed Deviation Slope, Random Intercept Model (Equivalent to 4a)

Level 1: $y_{ti} = \beta_{0i} + \beta_{1i}\left(Time_{ti}\right) + \beta_{2i}\left(Slope26_{ti}\right) + e_{ti}$

Level 2: Intercept: $\beta_{0i} = \gamma_{00} + U_{0i}$

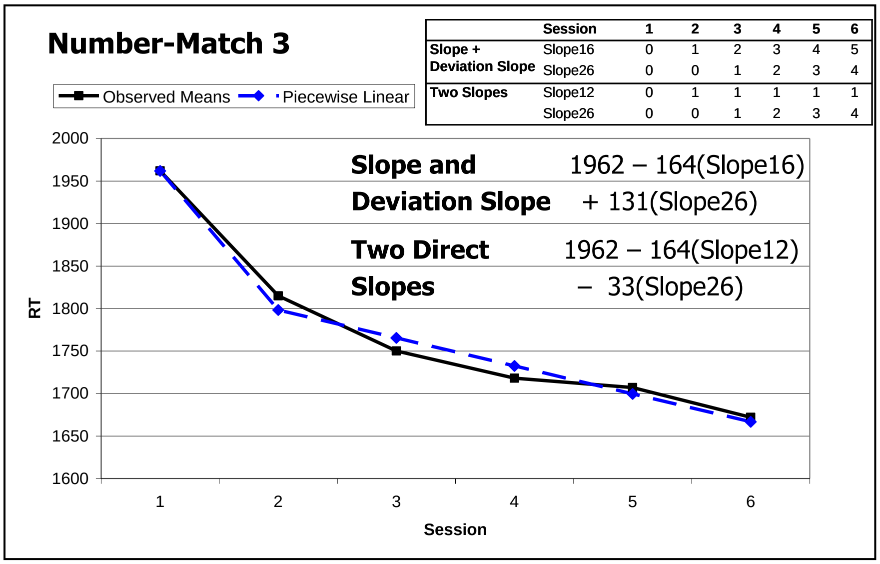
Time: $\beta_{1i} = \gamma_{10}$

Slope26: $\beta_{2i} = \gamma_{20}$

Session	1	2	3	4	5	6
Time → Slope16 =	0	1	2	3	4	5
Deviation → Slope26 =	0	0	1	2	3	4

```
mixed rt c.time c.slope26, || PersonID: , reml nolog  ///
         residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitFix16Fix26                // Save for LRT

------------------------------------------------------------------
          rt |      Coef.   Std. Err.           DF        t    P>|t|
-------------+----------------------------------------------------
        time |   -163.644   23.24149         503.0    -7.04    0.000
     slope26 |   130.7508   26.62648         503.0     4.91    0.000
       _cons |   1961.893    48.4187         128.7    40.52    0.000
------------------------------------------------------------------
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
PersonID: Identity           |
                  var(_cons) |   202683.4   29469.65      152424.8    269513.6
-----------------------------+------------------------------------------------
               var(Residual) |   34098.04   2150.109      30133.91    38583.65
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 805.80        Prob >= chibar2 = 0.0000

display "-2LL = " e(ll)*-2                   // Print -2LL for model
-2LL = 8382.6871

// Get conditional mean per occasion from values of time predictors, also slope26
lincom _cons*1 + c.time*0 + c.slope26*0      // Intercept at Session=1
lincom _cons*1 + c.time*1 + c.slope26*0      // Intercept at Session=2
lincom _cons*1 + c.time*1 + c.slope26*1      // Intercept at Session=3
lincom _cons*1 + c.time*1 + c.slope26*2      // Intercept at Session=4
lincom _cons*1 + c.time*1 + c.slope26*3      // Intercept at Session=5
lincom _cons*1 + c.time*1 + c.slope26*4      // Intercept at Session=6
lincom c.time*1 + c.slope26*1, small         // Fixed rate of change from session 2 to 6
```

```
                                Estimates (from SAS for better organization)
                                                  Standard
Label                                 Estimate       Error     DF    t Value   Pr > |t|   g = gamma fixed effect
Intercept at Session=1 Time=0          1961.89     48.4187    129      40.52     <.0001   g00(1)+ g10(0)+ g20(0)
Intercept at Session=2 Time=1          1798.25     47.0035    115      38.26     <.0001   g00(1)+ g10(1)+ g20(0)
Intercept at Session=3 Time=2          1765.36     45.9134    104      38.45     <.0001   g00(1)+ g10(2)+ g20(1)
Intercept at Session=4 Time=3          1732.46     45.5443    101      38.04     <.0001   g00(1)+ g10(3)+ g20(2)
Intercept at Session=5 Time=4          1699.57     45.9134    104      37.02     <.0001   g00(1)+ g10(4)+ g20(3)
Intercept at Session=6 Time=5          1666.68     47.0035    115      35.46     <.0001   g00(1)+ g10(5)+ g20(4)
Rate of change from session 2 to 6    -32.8932      5.8104    503      -5.66     <.0001   g10 + g20
```



	Session	1	2	3	4	5	6
Slope + Deviation Slope	Slope16	0	1	2	3	4	5
	Slope26	0	0	1	2	3	4
Two Slopes	Slope12	0	1	1	1	1	1
	Slope26	0	0	1	2	3	4

```
print("R Ch 6 5a: Fixed Time, Fixed Slope26, Random Intercept Model")
Fix16Fix26 = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                  formula=rt~1+time+slope26+(1|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(Fix16Fix26, chkREML=FALSE); summary(Fix16Fix26, ddf="Satterthwaite");
print("Get conditional mean per occasion from values of time predictors, also slope26")
print("Intercept at Session=1 Time=0"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(1,0,0))
print("Intercept at Session=2 Time=1"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(1,1,0))
print("Intercept at Session=3 Time=2"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(1,2,1))
print("Intercept at Session=4 Time=3"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(1,3,2))
print("Intercept at Session=5 Time=4"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(1,4,3))
print("Intercept at Session=6 Time=5"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(1,5,4))
print("Fixed rate of change from session 2 to 6"); contest1D(Fix16Fix26, ddf="Satterthwaite", L=c(0,1,1))
```

Model 5b: Random Slope, Fixed Deviation Slope Model

$$\text{Level 1: } y_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Time}_{ti}\right) + \beta_{2i}\left(\text{Slope26}_{ti}\right) + e_{ti}$$

$$\text{Level 2: Intercept: } \beta_{0i} = \gamma_{00} + U_{0i}$$

$$\text{Time: } \beta_{1i} = \gamma_{10} + U_{1i}$$

$$\text{Slope26: } \beta_{2i} = \gamma_{20}$$

```
display "STATA Ch 6: 5b: Random Time, Fixed Slope26 Model"
mixed rt c.time c.slope26, || PersonID: time, reml nolog covariance(unstructured) ///
      residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitRand16Fix26                         // Save for LRT
display "-2LL = " e(ll)*-2                             // Print -2LL for model
estat recovariance, relevel(PersonID) correlation // GCORR matrix
estat wcorrelation, covariance                        // V matrix
```

```
estat wcorrelation                                  // VCORR matrix
lrtest FitRand16Fix26 FitFix16Fix26   // LRT for random time variance and covariance

print("R Ch 6 5b: Random Time, Fixed Slope26 Model")
Rand16Fix26 = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                   formula=rt~1+time+slope26+(1+time|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(Rand16Fix26, chkREML=FALSE); summary(Rand16Fix26, ddf="Satterthwaite")

      AIC       BIC    logLik  deviance  df.resid
 8347.381  8378.229 -4166.691  8333.381   599.000

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 PersonID (Intercept) 254289   504.27
          time          2346    48.44   -0.53
 Residual              25934   161.04
Number of obs: 606, groups:  PersonID, 101

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  1961.89      52.67  109.38  37.246  < 2e-16
time         -163.64      20.83  466.59  -7.854 2.79e-14
slope26       130.75      23.22  403.00   5.631 3.37e-08

print("Use custom function to get predicted V matrix"); PrintV(Rand16Fix26)

         1        2        3        4        5        6
1 280223.8 241307.1 228324.8 215342.6 202360.3 189378.0
2 241307.1 256605.7 220035.5 209399.7 198763.8 188128.0
3 228324.8 220035.5 237680.5 203456.7 195167.4 186878.0
4 215342.6 209399.7 203456.7 223448.2 191570.9 185628.0
5 202360.3 198763.8 195167.4 191570.9 213908.9 184378.0
6 189378.0 188128.0 186878.0 185628.0 184378.0 209062.4
```

This random slope model predicts the same kind of **V** matrix as would a random linear time model —on the variance side, that's exactly what this is!

```
print("LRT for random time slope variance and covariance"); ranova(Rand16Fix26, reduce.term=TRUE)

                              npar  logLik    AIC    LRT Df Pr(>Chisq)
<none>                           7 -4166.7 8347.4
time in (1 + time | PersonID)    5 -4191.3 8392.7 49.306  2  1.965e-11
```

Model 5c: Random Slope, Random Deviation Slope Model (Equivalent to 4c)

Level 1: $y_{ti} = \beta_{0i} + \beta_{1i}\left(Time_{ti}\right) + \beta_{2i}\left(Slope26_{ti}\right) + e_{ti}$

Level 2: Intercept: $\beta_{0i} = \gamma_{00} + U_{0i}$

Time: $\beta_{1i} = \gamma_{10} + U_{1i}$

Slope26: $\beta_{2i} = \gamma_{20} + U_{2i}$

```
display "STATA Ch 6: 5c: Random Time, Random Slope26 Model"
mixed rt c.time c.slope26, || PersonID: time slope26,  ///
      reml nolog covariance(unstructured)              ///
      residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitRand16Rand26                     // Save for LRT
matrix Rand16Rand26 = r(table)                      // Save results for computations below
display "-2LL = " e(ll)*-2                          // Print -2LL for model
estat recovariance, relevel(PersonID) correlation // GCORR matrix
estat wcorrelation, covariance                      // V matrix
estat wcorrelation                                  // VCORR matrix
lrtest FitRand16Rand26 FitRand16Fix26   // LRT for random slope26 variance and covariances
```

```
matrix list Rand12Rand26                     // Show saved results
// Save fixed effects and variances for computations below
global FixInt = Rand16Rand26[1,3]            // Save fixed intercept
global FixS16 = Rand16Rand26[1,1]            // Save fixed slope16
global FixS26 = Rand16Rand26[1,2]            // Save fixed slope26
global IntVar = exp(Rand16Rand26[1,6])^2     // Save L2 random intercept variance
global S16Var = exp(Rand16Rand26[1,4])^2     // Save L2 random slope16 variance
global S26Var = exp(Rand16Rand26[1,5])^2     // Save L2 random slope26 variance
// Check if correct saved variances
display $IntVar
display $S16Var
display $S26Var
display "STATA 95% Random Intercept CI"
display "Lower = " $FixInt - 1.96*sqrt($IntVar)
display "Upper = " $FixInt + 1.96*sqrt($IntVar)
display "STATA 95% Random Slope16 CI"
display "Lower = " $FixS16 - 1.96*sqrt($S16Var)
display "Upper = " $FixS16 + 1.96*sqrt($S16Var)
display "STATA 95% Random Slope26 CI"
display "Lower = " $FixS26 - 1.96*sqrt($S26Var)
display "Upper = " $FixS26 + 1.96*sqrt($S26Var)
```

```
print("R Ch 6 5c: Random Time, Random Slope26 Model")
Rand16Rand26 = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                    formula=rt~1+time+slope26+(1+time+slope26|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(Rand16Rand26, chkREML=FALSE); summary(Rand16Rand26, ddf="Satterthwaite")
```

```
$AICtab
      AIC       BIC    logLik  deviance  df.resid
 8295.374  8339.443 -4137.687  8275.374   596.000

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 PersonID (Intercept) 284313   533.2
          time         63954   252.9    -0.40
          slope26      69916   264.4     0.31 -0.98
 Residual              17673   132.9
```

The **level-2 G matrix** (always **unstructured**) still contains variances for the random intercept and random linear time slope, **which is now just slope12 change (as in model 4c)**, as well as their covariance. We have now added a variance for the **random slope26, which allows *differences* in change from sessions 2 to 6**, as well as its covariances with the random intercept and random slope12 change for the same person. All covariances are given as correlations above.

```
Fixed effects:
            Estimate Std. Error     df t value    Pr(>|t|)
(Intercept)  1961.89      54.68 100.00  35.879     < 2e-16
time         -163.64      30.22 100.00  -5.415 0.000000422
slope26       130.75      32.55 100.00   4.017    0.000114
```

```
print("Use custom function to get predicted V matrix"); PrintV(Rand16Rand26)
         1        2        3        4        5        6
1 301986.0 230042.2 219398.4 208754.6 198110.8 187467.0
2 230042.2 257398.9 227409.8 215093.7 202777.7 190461.6
3 219398.4 227409.8 235384.1 208012.2 198313.4 188614.6
4 208754.6 215093.7 208012.2 218603.7 193849.1 186767.6
5 198110.8 202777.7 198313.4 193849.1 207057.9 184920.6
6 187467.0 190461.6 188614.6 186767.6 184920.6 200746.6
```

```
print("LRT for random slope26 variance and covariances"); ranova(Rand16Rand26, reduce.term=TRUE)

                                         npar  logLik    AIC    LRT Df Pr(>Chisq)
<none>                                     10 -4137.7 8295.4
time in (1 + time + slope26 | PersonID)     7 -4173.6 8361.1 71.727  3  1.821e-15
slope26 in (1 + time + slope26 | PersonID)  7 -4166.7 8347.4 58.007  3  1.567e-12
```

95% Random Effect Confidence Intervals that describe the *predicted* range of *individual* random effects:

$$\text{Random Effect 95\% CI} = \text{fixed effect} \pm \left(1.96 * \sqrt{\text{Random Variance}}\right)$$

$$\text{Intercept 95\% CI} = \gamma_{00} \pm \left(1.96 * \sqrt{\tau^2_{U_0}}\right) \rightarrow 1{,}961.9 \pm \left(1.96 * \sqrt{284{,}312}\right) = 917 \text{ to } 3{,}007$$

$$\text{Time 95\% CI} = \gamma_{10} \pm \left(1.96 * \sqrt{\tau^2_{U_1}}\right) \rightarrow -163.6 \pm \left(1.96 * \sqrt{63{,}954}\right) = -659 \text{ to } 322$$

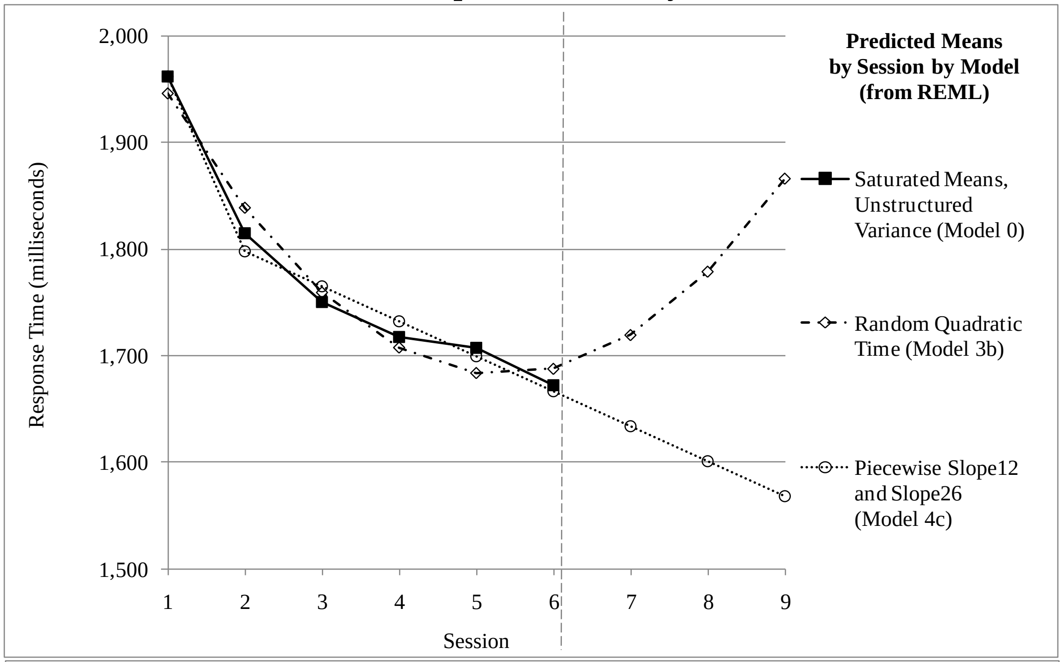
$$\text{Slope26 95\% CI} = \gamma_{20} \pm \left(1.96 * \sqrt{\tau^2_{U_2}}\right) \rightarrow 130.8 \pm \left(1.96 * \sqrt{69{,}916}\right) = -338 \text{ to } 649$$

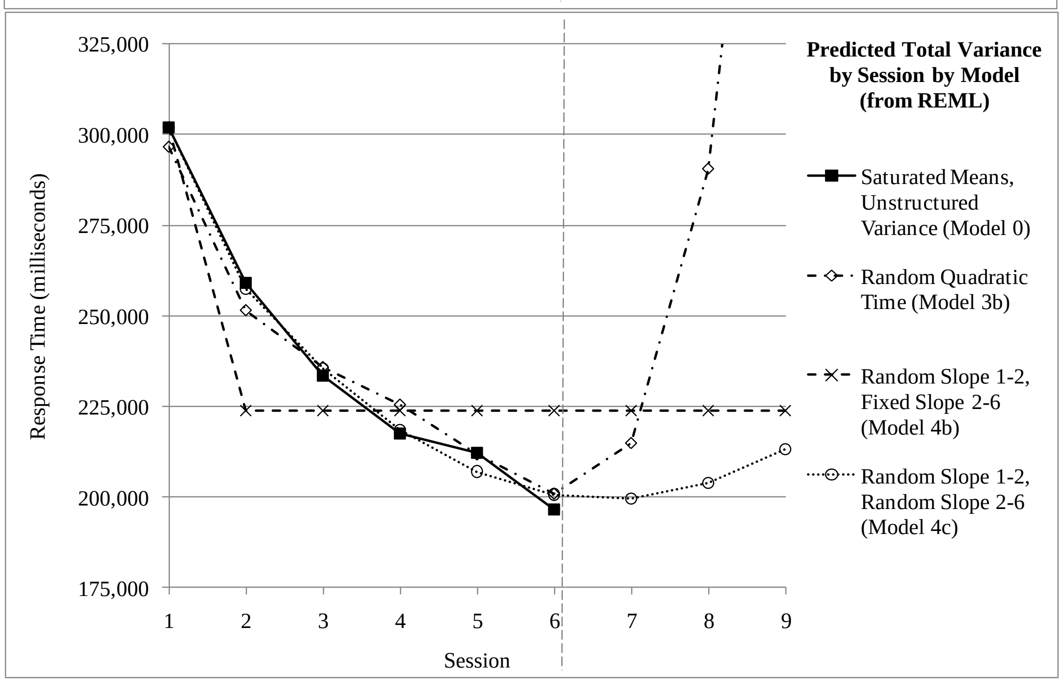
```
# Get ingredients for 95% random effect confidence intervals
# Print each object first to see which row and column values to extract
as.data.frame(fixef(Rand12Rand26)); as.data.frame(VarCorr(Rand12Rand26))
```

```
            fixef(Rand12Rand26)
(Intercept)          1961.89337
slope12              -163.64399
slope26               -32.89317

       grp        var1    var2        vcov       sdcor
1 PersonID (Intercept)    <NA> 284311.486 533.2086701
2 PersonID     slope12    <NA>  63954.180 252.8916367
3 PersonID     slope26    <NA>   2617.280  51.1593622
4 PersonID (Intercept) slope12 -54269.854  -0.4024639
5 PersonID (Intercept) slope26 -10643.761  -0.3901870
6 PersonID     slope12 slope26  -1672.293  -0.1292566
7 Residual        <NA>    <NA>  17673.032 132.9399568
```

```
# Save fixed effects and variances for computations below
FixInt = as.data.frame(fixef(Rand16Rand26))[1,1]    # Save fixed intercept
FixS16 = as.data.frame(fixef(Rand16Rand26))[2,1]    # Save fixed slope16
FixS26 = as.data.frame(fixef(Rand16Rand26))[3,1]    # Save fixed slope26
IntVar = as.data.frame(VarCorr(Rand16Rand26))[1,4]  # Save L2 random intercept variance
S16Var = as.data.frame(VarCorr(Rand16Rand26))[2,4]  # Save L2 random slope16 variance
S26Var = as.data.frame(VarCorr(Rand16Rand26))[3,4]  # Save L2 random slope26 variance

print("R 95% Random Intercept Confidence Interval")
print("Lower = "); FixInt - 1.96*sqrt(IntVar)
print("Upper = "); FixInt + 1.96*sqrt(IntVar)
print("R 95% Random Slope12 Confidence Interval")
print("Lower = "); FixS16 - 1.96*sqrt(S16Var)
print("Upper = "); FixS16 + 1.96*sqrt(S16Var)
print("R 95% Random Slope26 Confidence Interval")
print("Lower = "); FixS26 - 1.96*sqrt(S26Var)
print("Upper = "); FixS26 + 1.96*sqrt(S26Var)
```

So how did we do? Let's compare model predictions in terms of means (top) and variances (bottom)? Note that models 4c and 5c are equivalent, so only 4c is shown.





Bonus Material: Testing Absolute Fit of Each Side of the Model When Using REML

As shown as Model 0, the saturated means, unstructured variance model is the best-fitting model for each side (means and variances). However, when using REML, we cannot do a model comparison against our random two-piece slopes model, because models cannot differ in their fixed effects for the −2LL (LRT) to be valid. Instead, we can test the absolute fit for each side of the model separately as shown next. (This will be included in the second edition of my textbook, in progress).

The absolute fit of the two-piece model for the means can be tested by mimicking a saturated means model using the *same random two-piece slopes* (i.e., holding the model for the variance constant). Because the change from sessions 1–2 is already predicted perfectly, the session dummy codes must be added to the second slope (i.e., that still has degrees of freedom for not fitting the change from 2–6 perfectly).

```
display "STATA: Test Absolute Fit of Two-Piece Means Model"
display "Using Random Two-Piece Variance Model"
display "Add 3 session dummy codes to SECOND PIECE to saturate the means model"
mixed rt c.time c.slope26 c.s4 c.s5 c.s6, || PersonID: time slope26,  ///
      reml nolog covariance(unstructured)                              ///
      residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
```

```
------------------------------------------------------------------
          rt |      Coef.   Std. Err.          DF        t    P>|t|
-------------+----------------------------------------------------
        time |   -146.721   31.34924        115.6    -4.68    0.000
     slope26 |   81.58325   41.69216        236.7     1.96    0.052
          s4 |   32.89955   32.30947        300.0     1.02    0.309
          s5 |   87.41655   49.35353        300.0     1.77    0.078
          s6 |   117.5146   67.25752        300.0     1.75    0.082
       _cons |   1961.893   54.68056        100.0    35.88    0.000
------------------------------------------------------------------
```

```
test (c.s4=0)(c.s5=0)(c.s6=0), small  // Wald test for fixed two-piece slopes vs saturated means?

       F(  3,300.00) =    1.58
            Prob > F =    0.1946
```

Because there are now **6 fixed effects for the 6 means**, this model is equivalent to **saturated means** (even if the two fixed piecewise slopes are largely uninterpretable). In an SEM context, this model would be specified by letting three of the observed occasions' intercepts be estimated (as discrepancies). The multivariate Wald test indicates that the 3 extra session contrasts did not improve model fit (which is good news here).

```
print("R: Test Absolute Fit of Two-Piece Means Model (Using Random Two-Piece Variance Model)")
print("Add 3 session dummy codes to SECOND PIECE to saturate the means model")
PieceMean = lmer(data=Example6, REML=TRUE, control=lmerControl(optimizer="Nelder_Mead"),
                 formula=rt~1+time+slope26+s4+s5+s6+(1+time+slope26|PersonID))
print("Show results using Satterthwaite DDF"); summary(PieceMean, ddf="Satterthwaite")
print("Wald test for fixed two-piece linear slopes vs saturated means")
contestMD(PieceMean, ddf="Satterthwaite", L=rbind(c(0,0,0,1,0,0),c(0,0,0,0,1,0),c(0,0,0,0,0,1)))
```

The absolute fit of the random two-piece model for the variance can be tested against a UN variance model using the *same fixed two-piece slopes* (i.e., holding the model for the means constant):

```
display "STATA: Test Absolute Fit of Two-Piece Variance Model"
display "Using Fixed Two-Piece Means Model"
display "Change to Unstructured R matrix as variance model answer key"
mixed rt c.time c.slope26, || PersonID: , noconstant reml nolog difficult  ///
      residuals(unstructured,t(session)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitFixPieceUN         // Save for LRT
display "-2LL = " e(ll)*-2            // Print -2LL for model
lrtest FitFixPieceUN FitRand16Rand26  // LRT for random two-piece slopes vs unstruct variance model
```

```
print("R: Test Absolute Fit of Two-Piece Variance Model")
print("Using Fixed Two-Piece Means Model")
print("Change to Unstructured R matrix as variance model answer key in GLS")
FixPieceUN = gls(data=Example6, method="REML", model=rt~1+time+slope26,
                 correlation=corSymm(form=~as.numeric(session)|PersonID),  # Unstructured corrs
                 weights=varIdent(form=~1|session))                         # Het variances
print("LRT for random piecewise slopes vs unstructured variance model")
# Have to re-run random two-piece model using LME to get LRT")
RandPiecelme = lme(data=Example6, method="REML", rt~1+time+slope26,
                   random=~1+time+slope26|PersonID)
anova(FixPieceUN,RandPiecelme) # anova does LRT using LME versions
```

What does this nonsignificant LRT result indicate about our random two-piece slopes model?

```
             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
FixPieceUN       1 24 8307.601 8413.247 -4129.801
RandPiecelme     2 10 8295.374 8339.393 -4137.687 1 vs 2 15.77284  0.3274
```