Example 6d: Modeling Change over Time Using Log Time to Approximate Exponential Trends
(complete data, syntax, and output available for SAS electronically)

These data for these example models come from Hoffman (2015) chapter 6. We will be examining change in response time (RT) in milliseconds over six practice sessions (balanced time) to a measure of processing speed in a sample of 101 older adults. Previously in Example6c we used SAS NLMIXED to estimate a truly nonlinear model with an exponential curve, in which REML is not available. Below we examine a strategy to mimic the same shape that uses linear and quadratic effects of natural-log-transformed time instead, such that we can use REML in MIXED. At the end we will compare the fit of the "best" version of each family of models (polynomial, piecewise, exponential, and latent basis).

SAS Syntax for Data Import and Manipulation:

```
* Define global variable for file location to be replaced in code below;
%LET filesave=C:\Dropbox\22_PSQF6271\PSQF6271_Example6;
* Location for SAS files for these models (uses macro variable filesave);
LIBNAME filesave "&filesave.";

DATA work.Example6; SET filesave.SAS_Chapter6;
* Log-transform time as new predictor;
  logtime=LOG(session); LABEL logtime="logtime: Log-Transformed Time (0=1)";
* Create session dummy codes for testing means model absolute fit in REML;
  IF session=1 THEN s1=1; ELSE s1=0;
  IF session=2 THEN s2=1; ELSE s2=0;
  IF session=3 THEN s3=1; ELSE s3=0;
  IF session=4 THEN s4=1; ELSE s4=0;
RUN;
```

STATA Syntax for Data Import and Manipulation:

```
// Define global variable for file location to be replaced in code below
   global filesave "C:\Dropbox\22_PSQF6271\PSQF6271_Example6"
// Import chapter 6 stacked data and create time predictor variables
   use "$filesave\STATA_Chapter6.dta", clear

// Log-transform time as new predictor
   gen logtime=log(session)
   gen logtimesq=log(session)*log(session)
   label variable logtime   "logtime: Natural Log Session"
   label variable logtimesq "logtimesq: Quadratic Natural Log Session"

// Create session dummy codes for testing means model absolute fit in REML
   gen s1=0
   gen s2=0
   gen s3=0
   gen s4=0
   replace s1=1 if session==1
   replace s2=1 if session==2
   replace s3=1 if session==3
   replace s4=1 if session==4
```

R Syntax for Data Import and Manipulation:

```
# Define variables for working directory and data name
filesave = "C:\\Dropbox/22_PSQF6271/PSQF6271_Example6/"
filename = "SAS_Chapter6.sas7bdat"
setwd(dir=filesave)

# Import chapter 6 stacked data with labels
Example6 = read_sas(data_file=paste0(filesave,filename))
# Convert to data frame as data frame without labels to use for analysis
Example6 = as.data.frame(Example6)
# Sort data by PersonID (needed for correct RCOV matrix)
Example6 = sort_asc(Example6,PersonID,wave)

# Log-transform time as new predictor
Example6$logtime=log(Example6$session)
```

```
# Create session dummy codes for testing means model absolute fit in REML
Example6$s1=0
Example6$s2=0
Example6$s3=0
Example6$s4=0
Example6$s1[which(Example6$session==1)]=1
Example6$s2[which(Example6$session==2)]=1
Example6$s3[which(Example6$session==3)]=1
Example6$s4[which(Example6$session==4)]=1
```

Note—a TYPC=VC R matrix will be used in all of the models that follow (not shown to save space).

Model 7a. Fixed Linear Log Time, Random Intercept Model

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

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

Linear LogTime: $\beta_{1i} = \gamma_{10}$

```
TITLE1 "SAS: 7a: Fixed Linear Log Time, Random Intercept Model";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT / G V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED session / R TYPE=VC SUBJECT=PersonID;
     ODS OUTPUT CovParms=CovFixLinLog InfoCrit=FitFixLinLog; * Save for pseudo-R2 and LRT;
     * Get conditional mean per occasion from value of time predictor;
     ESTIMATE "Intercept at Session=1"    intercept 1 logtime 0;
     ESTIMATE "Intercept at Session=2"    intercept 1 logtime 0.6931;
     ESTIMATE "Intercept at Session=3"    intercept 1 logtime 1.0986;
     ESTIMATE "Intercept at Session=4"    intercept 1 logtime 1.3863;
     ESTIMATE "Intercept at Session=5"    intercept 1 logtime 1.6094;
     ESTIMATE "Intercept at Session=6"    intercept 1 logtime 1.7918;
RUN;
TITLE1 "Calculate pseudo R2 -- variance accounted for by fixed linear log time";
%PseudoR2(NCov=2, CovFewer=CovEmpty, CovMore=CovFixLinLog); TITLE1;
```

```
display "STATA: 7a: Fixed Linear Log Time, Random Intercept Model"
mixed rt c.logtime, || personid: , variance reml covariance(unstructured) ///
         residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2      // Print -2LL for model
      estat ic, n(101)                // AIC and BIC
      estat wcorrelation, covariance  // V matrix
      estat wcorrelation              // VCORR matrix
      // Get conditional mean per occasion from value of time predictor
      lincom _cons*1 + c.logtime*0          // Intercept at Session=1
      lincom _cons*1 + c.logtime*0.6931     // Intercept at Session=2
      lincom _cons*1 + c.logtime*1.0986     // Intercept at Session=3
      lincom _cons*1 + c.logtime*1.3863     // Intercept at Session=4
      lincom _cons*1 + c.logtime*1.6094     // Intercept at Session=5
      lincom _cons*1 + c.logtime*1.7918     // Intercept at Session=6
      estimates store FitFixLinLog          // Save for LRT
```

```
print("R 7a: Fixed Linear Log Time, Random Intercept Model")
FixLinLog = lmer(data=Example6, REML=TRUE, formula=rt~1+logtime+(1|PersonID))
print("Show results using Satterthwaite DDF including -2LL as deviance")
summary(FixLinLog, ddf="Satterthwaite"); llikAIC(FixLinLog, chkREML=FALSE)
print("Get conditional mean per occasion from value of time predictor")
print("Intercept at Session=1"); contest1D(FixLinLog, ddf="Satterthwaite", L=c(1,0))
print("Intercept at Session=2"); contest1D(FixLinLog, ddf="Satterthwaite", L=c(1,0.6931))
print("Intercept at Session=3"); contest1D(FixLinLog, ddf="Satterthwaite", L=c(1,1.0986))
print("Intercept at Session=4"); contest1D(FixLinLog, ddf="Satterthwaite", L=c(1,1.3863))
print("Intercept at Session=5"); contest1D(FixLinLog, ddf="Satterthwaite", L=c(1,1.6094))
print("Intercept at Session=6"); contest1D(FixLinLog, ddf="Satterthwaite", L=c(1,1.7918))
```

SAS Output:

Estimated V Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	**236852**	202669	202669	202669	202669	202669
2	202669	**236852**	202669	202669	202669	202669
3	202669	202669	**236852**	202669	202669	202669
4	202669	202669	202669	**236852**	202669	202669
5	202669	202669	202669	202669	**236852**	202669
6	202669	202669	202669	202669	202669	**236852**

The predicted **V** matrix still has a compound symmetry pattern because we have not yet added to the model for the variance (still only a random intercept variance in **G**).

Estimated V Correlation Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.8557	0.8557	0.8557	0.8557	0.8557
2	0.8557	1.0000	0.8557	0.8557	0.8557	0.8557
3	0.8557	0.8557	1.0000	0.8557	0.8557	0.8557
4	0.8557	0.8557	0.8557	1.0000	0.8557	0.8557
5	0.8557	0.8557	0.8557	0.8557	1.0000	0.8557
6	0.8557	0.8557	0.8557	0.8557	0.8557	1.0000

This random intercept only model for the variance (CS) translates into equal correlation over time as shown in the VCORR matrix (the ICC).

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z	
UN(1,1)	PersonID	202669	29470	6.88	<.0001	Level-2 random intercept variance of U_{0i}
session	PersonID	34183	2153.35	15.87	<.0001	Level-1 residual variance of e_{ti}

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
1	805.80	<.0001

This LRT is for the conditional ICC > 0.

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
8391.3	2	8395.3	8395.3	8397.4	8400.5	8402.5

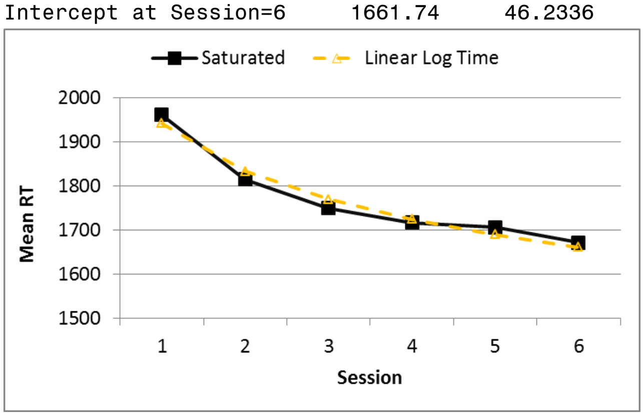
Only variance model parameters are "counted" as parms using REML.

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	
Intercept	1942.55	47.4172	119	40.97	<.0001	gamma00
logtime	-156.72	12.4160	504	-12.62	<.0001	gamma10

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	**gamma00(1)+gamma10(logtime)**
Intercept at Session=1	1942.55	47.4172	119	40.97	<.0001	
Intercept at Session=2	1833.93	45.6960	102	40.13	<.0001	
Intercept at Session=3	1770.38	45.4206	100	38.98	<.0001	
Intercept at Session=4	1725.29	45.5629	101	37.87	<.0001	
Intercept at Session=5	1690.33	45.8648	104	36.85	<.0001	
Intercept at Session=6	1661.74	46.2336	107	35.94	<.0001	



The fixed linear effect of log time mimics an exponential curve (and appears to fit pretty well to the saturated means).

As shown below, the fixed linear slope for log time accounted for 24% of the level-1 residual variance.

```
Calculate pseudo R2 -- variance accounted for by fixed linear log time
PsuedoR2 (% Reduction) for CovEmpty vs. CovFixLinLog
```

Name	CovParm	Subject	Estimate	StdErr	ZValue	ProbZ	PseudoR2
CovEmpty	UN(1,1)	PersonID	200883	29471	6.82	<.0001	.
CovEmpty	session	PersonID	44900	2825.63	15.89	<.0001	.
CovFixLinLog	UN(1,1)	PersonID	202669	29470	6.88	<.0001	-0.00889
CovFixLinLog	session	PersonID	34183	2153.35	15.87	<.0001	**0.23868**

Model 7b. Random Linear Log Time

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

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

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

```
TITLE1 "SAS: 7b: Random Linear Log Time Model";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT logtime / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED session / R TYPE=VC SUBJECT=PersonID;
     ODS OUTPUT CovParms=CovRandLinLog  InfoCrit=FitRandLinLog; * Save for pseudo-R2 and LRT;
RUN; TITLE1 "Calculate LRT -- does random linear log time slope improve model fit?";
%FitTest(FitFewer=FitFixLinLog, FitMore=FitRandLinLog); TITLE1;
```

```
display "STATA: 7b: Random Linear Log Time Model"
mixed rt c.logtime,  || personid: logtime, variance reml covariance(unstructured) ///
          residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2                            // Print -2LL for model
      estat ic, n(101)                                      // AIC and BIC
      estat recovariance, relevel(personid)                 // G matrix
      estat recovariance, relevel(personid) correlation     // GCORR matrix
      estat wcorrelation, covariance                        // V matrix
      estat wcorrelation                                    // VCORR matrix
      estimates store FitRandLinLog      // Save for LRT
      lrtest FitRandLinLog FitFixLinLog  // Does random linear log time slope improve model fit?
```

```
print("R 7b: Random Linear Log Time Model")
RandLinLog = lmer(data=Example6, REML=TRUE, formula=rt~1+logtime+(1+logtime|PersonID))
print("Show results using Satterthwaite DDF including -2LL as deviance")
summary(RandLinLog, ddf="Satterthwaite"); llikAIC(RandLinLog, chkREML=FALSE)
print("Does random linear time slope improve model fit?")
ranova(RandLinLog, reduce.term=TRUE) # Remove random slope and covariance
```

SAS Output:

Estimated G Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	101	274252	-44541
2	logtime	101	**-44541**	**23101**

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	101	1.0000	-0.5596
2	logtime	101	**-0.5596**	1.0000

This **level-2 G matrix** (always **unstructured**) still contains a random intercept variance (1,1), and we have added a random linear log time slope variance (2,2), as well as a covariance between the random intercept and random linear log time slope for the same person (2,1). The intercept–slope correlation is shown in GCORR (2,1).

Estimated V Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	**298372**	243378	225318	212505	202566	194445
2	243378	**247724**	212036	203829	197463	192262
3	225318	212036	**228387**	198754	194478	190984
4	212505	203829	198754	**219274**	192360	190078
5	202566	197463	194478	192360	**214838**	189375
6	194445	192262	190984	190078	189375	**212922**

The marginal **V** matrix now predicts that **variance changes quadratically over time** (as a function of logtime^2).

```
               Estimated V Correlation Matrix for PersonID 101
 Row        Col1        Col2        Col3        Col4        Col5        Col6
   1      1.0000      0.8952      0.8631      0.8308      0.8001      0.7714
   2      0.8952      1.0000      0.8914      0.8746      0.8559      0.8371
   3      0.8631      0.8914      1.0000      0.8882      0.8780      0.8661
   4      0.8308      0.8746      0.8882      1.0000      0.8863      0.8797
   5      0.8001      0.8559      0.8780      0.8863      1.0000      0.8854
   6      0.7714      0.8371      0.8661      0.8797      0.8854      1.0000
```

The marginal **VCORR** matrix now predicts that the covariance changes over time as well (in a time-dependent pattern).

```
                    Covariance Parameter Estimates
                                            Standard         Z
Cov Parm     Subject      Estimate         Error       Value        Pr Z
UN(1,1)      PersonID       274252         41240        6.65      <.0001   Level-2 random intercept variance of U0i
UN(2,1)      PersonID       -44541         11493       -3.88      0.0001   Level-2 random intercept-linear covariance
UN(2,2)      PersonID        23101       4882.26        4.73      <.0001   Level-2 random linear slope variance of U1i
session      PersonID        24120       1697.11       14.21      <.0001   Level-1 residual variance of eti

  Null Model Likelihood Ratio Test
    DF     Chi-Square      Pr > ChiSq
     3         873.64          <.0001
```

This LRT tells us whether we need the full 2x2 **G** matrix (so it is not helpful).

```
                          Information Criteria
Neg2LogLike       Parms         AIC        AICC        HQIC         BIC        CAIC
     8323.4           4      8331.4      8331.5      8335.7      8341.9      8345.9

                     Solution for Fixed Effects
                                Standard
Effect         Estimate         Error       DF    t Value    Pr > |t|
Intercept       1942.55       53.7211      100      36.16      <.0001
logtime         -156.72       18.3711      100      -8.53      <.0001

Calculate LRT -- does random linear log time slope improve model fit?
Likelihood Ratio Test for FitFixLinLog vs. FitRandLinLog
                Neg2Log
    Name          Like       Parms         AIC         BIC      DevDiff      DFdiff        Pvalue
FitFixLinLog    8391.3         2       8395.3      8400.5         .            .               .
FitRandLinLog   8323.4         4       8331.4      8341.9      67.8448         2      1.8874E-15
```

This LRT tells us that the random log time slope improves model fit.

<u>95% Random Effects Confidence Intervals that describe the *predicted* range of *individual* random effects:</u>

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

$$\text{Intercept 95\% CI} = \gamma_{00} \pm \left(1.96*\sqrt{\tau^2_{U_0}}\right) \rightarrow 1{,}942.55 \pm \left(1.96*\sqrt{274{,}252}\right) = 916 \text{ to } 2{,}969$$

$$\text{Linear LogTime Slope 95\% CI} = \gamma_{10} \pm \left(1.96*\sqrt{\tau^2_{U_1}}\right) \rightarrow -156.72 \pm \left(1.96*\sqrt{23{,}101}\right) = -455 \text{ to } 141$$

The absolute fit of the linear log time model for the means can be tested by mimicking a saturated means model using the *same random linear log time slope* (i.e., holding the model for the variance constant):

```
TITLE1 "SAS: Test Absolute Fit of Linear Log Time Means (Using Random Linear Log Time Variance Model)";
TITLE2 "Add 4 session dummy codes to saturate the means model";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime s1 s2 s3 s4 / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT logtime / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED session / R TYPE=VC SUBJECT=PersonID;
     CONTRAST "Does fixed linear log time reproduce saturated means?" s1 1, s2 1, s3 1, s4 1 / CHISQ;
RUN; TITLE1; TITLE2;
```

```
display "STATA: Test Absolute Fit of Linear Log Time Means (Using Random Linear Log Time Variance Model)"
display "Add 3 session dummy codes to saturate the means model"
mixed rt c.logtime c.s1 c.s2 c.s3 c.s4, ///
         || personid: logtime, variance reml covariance(unstructured) ///
         residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2                        // Print -2LL for model
      test (c.s1=0)(c.s2=0)(c.s3=0)(c.s4=0), small  // Does fixed linear log time reproduce sat means?

print("R: Test Absolute Fit of Linear Log Time Means (Using Random Linear Log Time Variance Model)")
print("Add 3 session dummy codes to second piece to saturate the means model")
LinLogMean = lmer(data=Example6, REML=TRUE, formula=rt~1+logtime+s1+s2+s3+s4+(1+logtime|PersonID))
print("Show results using Satterthwaite DDF including -2LL as deviance")
summary(LinLogMean, ddf="Satterthwaite"); llikAIC(LinLogMean, chkREML=FALSE)
print("Does fixed two-piece slope reproduce saturated means?")
contestMD(LinLogMean, ddf="Satterthwaite",
          L=rbind(c(0,0,1,0,0,0),c(0,0,0,1,0,0),c(0,0,0,0,1,0) ,c(0,0,0,0,0,1)))
```

SAS Output (relevant tables only):

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	2016.49	210.04	449	9.60	<.0001
logtime	-192.19	120.43	418	-1.60	0.1113
s1	-54.5944	204.05	400	-0.27	0.7892
s2	-68.1019	121.83	400	-0.56	0.5765
s3	-55.3148	74.3533	400	-0.74	0.4573
s4	-32.2644	42.0226	400	-0.77	0.4431

Because there are now **6 fixed effects for the 6 means**, this model is equivalent to **saturated means** (even if the linear log time fixed slope is largely uninterpretable). In an SEM context, this model would be specified by letting four of the observed occasions' intercepts be estimated (as discrepancies).

The multivariate Wald test using CONTRAST indicates that the 4 extra session contrasts did not improve model fit (which is good news here).

Contrasts

Label	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
Does fixed linear log time reproduce saturated means?	4	400	6.69	1.67	0.1531	**0.1554**

The absolute fit of the random linear log time model for the variance can be tested against a UN variance model using the *same fixed linear log time slope* (i.e., holding the model for the means constant):

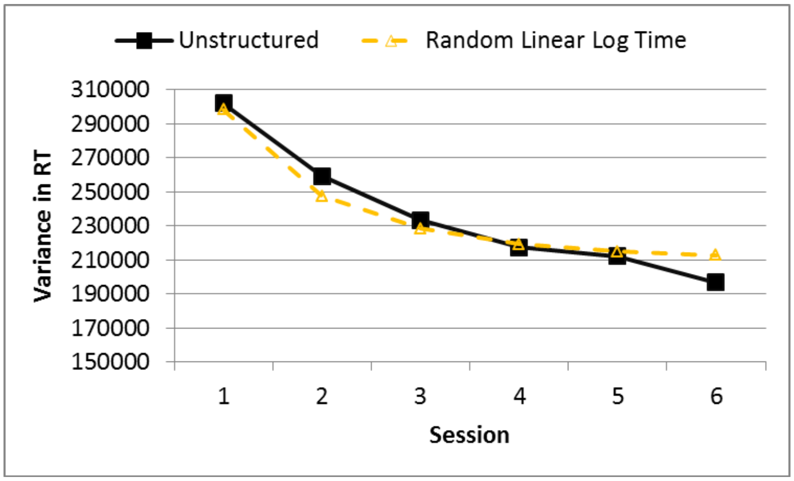
```
TITLE1 "SAS: Test Absolute Fit of Linear Log Time Variance (Using Fixed Linear Log Time Means Model)";
TITLE2 "Change to Unstructured R matrix as variance model answer key";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime / SOLUTION DDFM=Satterthwaite;
     REPEATED session / R RCORR TYPE=UN SUBJECT=PersonID;
     ODS OUTPUT InfoCrit=FitFixLinLogUN; * Save for LRT;
RUN; TITLE2;
TITLE1 "Calculate LRT -- does random linear log time reproduce UN variance model?";
 %FitTest(FitFewer=FitRandLinLog, FitMore=FitFixLinLogUN); TITLE1;


display "STATA: Test Absolute Fit of Linear Log Time Variance (Using Fixed Linear Log Time Means Model)"
display "Change to Unstructured R matrix as variance model answer key"
mixed rt c.logtime, || personid: , noconstant variance reml ///
         residuals(unstructured,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2           // Print -2LL for model
      estimates store FitFixLinLogUN       // Save for LRT
      lrtest FitFixLinLogUN FitRandLinLog // Does random linear log time reproduce UN variance model?


print("R: Test Absolute Fit of Linear Log Time Variance Model (Using Fixed Linear Log Time Means Model)")
print("Change to Unstructured R matrix as variance model answer key in GLS")
LinLogUN = gls(data=Example6, method="REML", model=rt~1+logtime,
               correlation=corSymm(form=~as.numeric(session)|PersonID),  # Unstructured correlations
               weights=varIdent(form=~1|session))                         # Heterogeneous variances
print("Have to re-run random linear log time model using LME to get LRT")
RandLinLoglme = lme(data=Example6, method="REML", rt~1+logtime, random=~1+logtime|PersonID)
anova(LinLogUN,RandLinLoglme) # anova does LRT using LME versions
```

SAS Output (relevant table only):

```
Calculate LRT -- does random linear log time reproduce UN variance model?
Likelihood Ratio Test for FitRandLinLog vs. FitRandUN
                  Neg2Log
    Name           Like      Parms       AIC        BIC     DevDiff    DFdiff       Pvalue
FitRandLinLog     8323.4       4       8331.4     8341.9       .          .       .
FitRandUN         8265.4      21       8307.4     8362.3    58.0097      17      .000002230
```



The random linear slope of log time approximates most of the observed variances from the unstructured model, but not well enough according to the LRT (and the same is likely true of the covariances, not shown here).

Model 7c. Fixed Quadratic, Random Linear Log Time

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

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

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

Quadratic LogTime: $\beta_{2i} = \gamma_{20}$

Predicted intercept at any occasion:

$= \gamma_{00} + \gamma_{10}(\text{LogTime}_{ti}) + \gamma_{20}(\text{LogTime}_{ti})^2$

Instantaneous linear time slope at any occasion:

$= \gamma_{10} + 2\gamma_{20}(\text{LogTime}_{ti})$

```
TITLE1 "SAS: 7c: Fixed Quadratic, Random Linear Log Time Model";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime logtime*logtime / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT logtime / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED session / R TYPE=VC SUBJECT=PersonID;
     ODS OUTPUT CovParms=CovFixQuadLog InfoCrit=FitFixQuadLog; * Save for pseudo-R2 and LRT;
     * Get conditional mean per occasion from values of time predictors;
     ESTIMATE "Intercept at Session=1"    intercept 1 logtime 0      logtime*logtime 0;
     ESTIMATE "Intercept at Session=2"    intercept 1 logtime 0.6931 logtime*logtime 0.4805;
     ESTIMATE "Intercept at Session=3"    intercept 1 logtime 1.0986 logtime*logtime 1.2069;
     ESTIMATE "Intercept at Session=4"    intercept 1 logtime 1.3863 logtime*logtime 1.9218;
     ESTIMATE "Intercept at Session=5"    intercept 1 logtime 1.6094 logtime*logtime 2.5903;
     ESTIMATE "Intercept at Session=6"    intercept 1 logtime 1.7918 logtime*logtime 3.2104;
     * Get instantaneous linear slope per occasion from 2*value of time predictor;
     ESTIMATE "Linear Slope at Session=1" logtime 1 logtime*logtime 0;
     ESTIMATE "Linear Slope at Session=2" logtime 1 logtime*logtime 1.3863;
     ESTIMATE "Linear Slope at Session=3" logtime 1 logtime*logtime 2.1972;
     ESTIMATE "Linear Slope at Session=4" logtime 1 logtime*logtime 2.7726;
     ESTIMATE "Linear Slope at Session=5" logtime 1 logtime*logtime 3.2189;
     ESTIMATE "Linear Slope at Session=6" logtime 1 logtime*logtime 3.5835;
RUN;
TITLE1 "Calculate pseudo R2 -- variance accounted for by fixed quadratic log time";
%PseudoR2(NCov=4, CovFewer=CovRandLinLog, CovMore=CovFixQuadLog); TITLE1;
```

Because twice the quadratic slope is how the linear slope changes per unit time, the value for log time used in estimating the linear slope per session gets multiplied by 2.

```
display "STATA: 7c: Fixed Quadratic, Random Linear Log Time Model"
mixed rt c.logtime c.logtime#c.logtime, || personid: logtime, variance reml covariance(unstructured) ///
         residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2                        // Print -2LL for model
      estat ic, n(101)                                  // AIC and BIC
      estat recovariance, relevel(personid)             // G matrix
      estat recovariance, relevel(personid) correlation // GCORR matrix
      estat wcorrelation, covariance                    // V matrix
      estat wcorrelation                                // VCORR matrix
      * Get conditional mean per occasion from values of time predictors
      lincom _cons*1 + c.logtime*0      + c.logtime#c.logtime*0         // Intercept at Session=1
      lincom _cons*1 + c.logtime*0.6931 + c.logtime#c.logtime*0.4805    // Intercept at Session=2
      lincom _cons*1 + c.logtime*1.0986 + c.logtime#c.logtime*1.2069    // Intercept at Session=3
      lincom _cons*1 + c.logtime*1.3863 + c.logtime#c.logtime*1.9218    // Intercept at Session=4
      lincom _cons*1 + c.logtime*1.6094 + c.logtime#c.logtime*2.5903    // Intercept at Session=5
      lincom _cons*1 + c.logtime*1.7918 + c.logtime#c.logtime*3.2104    // Intercept at Session=6
      // Get instantaneous linear slope per occasion from 2*value of time predictor
      lincom c.logtime*1 + c.logtime#c.logtime*0     , small    // Linear Slope at Session=1
      lincom c.logtime*1 + c.logtime#c.logtime*1.3863, small    // Linear Slope at Session=2
      lincom c.logtime*1 + c.logtime#c.logtime*2.1972, small    // Linear Slope at Session=3
      lincom c.logtime*1 + c.logtime#c.logtime*2.7726, small    // Linear Slope at Session=4
      lincom c.logtime*1 + c.logtime#c.logtime*3.2189, small    // Linear Slope at Session=5
      lincom c.logtime*1 + c.logtime#c.logtime*3.5835, small    // Linear Slope at Session=6
      estimates store FitFixQuadLog    // Save for LRT
```

```
print("R 7c: Fixed Quadratic, Random Linear Log Time Model")
FixQuadLog = lmer(data=Example6, REML=TRUE, formula=rt~1+logtime+I(logtime^2)+(1+logtime|PersonID))
print("Show results using Satterthwaite DDF including -2LL as deviance")
summary(FixQuadLog, ddf="Satterthwaite"); llikAIC(FixQuadLog, chkREML=FALSE)
print("Get conditional mean per occasion from values of time predictors")
print("Intercept at Session=1"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(1,0,0))
print("Intercept at Session=2"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(1,0.6931,0.4805))
print("Intercept at Session=3"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(1,1.0986,1.2069))
print("Intercept at Session=4"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(1,1.3863,1.9218))
print("Intercept at Session=5"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(1,1.6094,2.5903))
print("Intercept at Session=6"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(1,1.7918,3.2104))
print("Get instantaneous linear slope per occasion from 2*value of time predictor")
print("Linear Slope at Session=1 Time=0"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(0,1,0))
print("Linear Slope at Session=2 Time=1"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(0,1,1.3863))
print("Linear Slope at Session=3 Time=2"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(0,1,2.1972))
print("Linear Slope at Session=4 Time=3"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(0,1,2.7726))
print("Linear Slope at Session=5 Time=4"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(0,1,3.2189))
print("Linear Slope at Session=6 Time=5"); contest1D(FixQuadLog, ddf="Satterthwaite", L=c(0,1,3.5835))
```

SAS Output:

Estimated G Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	101	274453	-44682
2	logtime	101	-44682	23229

The model for the variance has not changed, so it has the same structure as before (**G** is 2x2).

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	101	1.0000	-0.5596
2	logtime	101	-0.5596	1.0000

Estimated V Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	**298292**	243482	225365	212511	202540	194394
2	243482	**247510**	212083	203861	197483	192273
3	225365	212083	**228152**	198801	194525	191032
4	212511	203861	198801	**219050**	192426	190151
5	202540	197483	194525	192426	**214637**	189468
6	194394	192273	191032	190151	189468	**212749**

Estimated V Correlation Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.8961	0.8639	0.8314	0.8005	0.7717
2	0.8961	1.0000	0.8925	0.8755	0.8568	0.8379
3	0.8639	0.8925	1.0000	0.8893	0.8790	0.8671
4	0.8314	0.8755	0.8893	1.0000	0.8874	0.8808
5	0.8005	0.8568	0.8790	0.8874	1.0000	0.8866
6	0.7717	0.8379	0.8671	0.8808	0.8866	1.0000

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z	
UN(1,1)	PersonID	274453	41239	6.66	<.0001	Level-2 random intercept variance of U_{0i}
UN(2,1)	PersonID	-44682	11493	-3.89	0.0001	Level-2 random intercept-linear covariance
UN(2,2)	PersonID	23229	4880.99	4.76	<.0001	Level-2 random linear time slope variance of U_{1i}
session	PersonID	23839	1679.38	14.20	<.0001	Level-1 residual variance of e_{ti}

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	876.51	<.0001

This LRT tells us whether we need the full 2x2 **G** matrix (so it is not helpful).

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
8309.9	4	8317.9	8318.0	8322.1	8328.4	8332.4

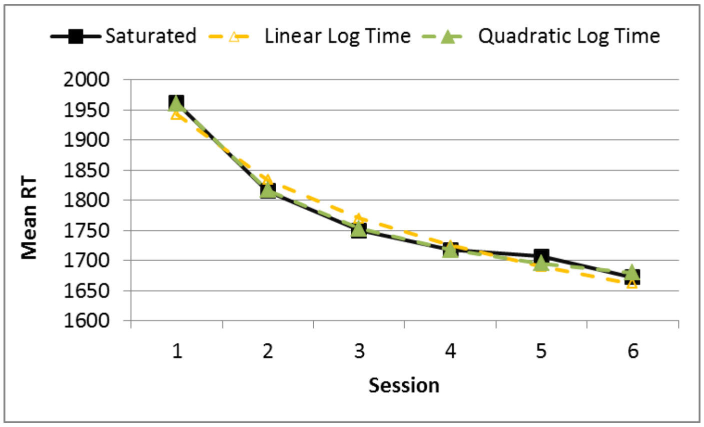
Is the fixed quadratic log time slope significant? How do we know?

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	
Intercept	1960.96	54.2653	104	36.14	<.0001	gamma00
logtime	-240.61	39.4593	502	-6.10	<.0001	gamma10
logtime*logtime	46.9146	19.5288	403	2.40	0.0167	gamma20

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	**g = gamma, logt = logtime**
Intercept at Session=1	1960.96	54.2653	104	36.14	<.0001	g00(1)+ g10(logt)+ g20(logt^2)
Intercept at Session=2	1816.74	48.1936	105	37.70	<.0001	
Intercept at Session=3	1753.25	45.9683	105	38.14	<.0001	
Intercept at Session=4	1717.57	44.6261	101	38.49	<.0001	
Intercept at Session=5	1695.25	44.2821	100	38.28	<.0001	
Intercept at Session=6	1680.45	44.9706	106	37.37	<.0001	
Linear Slope at Session=1	-240.61	39.4593	502	-6.10	<.0001	g10(1) + 2*g20(logt)
Linear Slope at Session=2	-175.57	19.9776	139	-8.79	<.0001	
Linear Slope at Session=3	-137.53	20.0322	140	-6.87	<.0001	
Linear Slope at Session=4	-110.53	26.5904	338	-4.16	<.0001	
Linear Slope at Session=5	-89.5967	33.4381	472	-2.68	0.0076	
Linear Slope at Session=6	-72.4917	39.5812	502	-1.83	0.0676	



The quadratic effect of log time appears to make the predicted line a little more bendy, such that it fits the saturated means slightly (but significantly) better.

```
Calculate pseudo R2 -- variance accounted for by fixed quadratic log time
PsuedoR2 (% Reduction) for CovRandLinLog vs. CovFixQuadLog
```

Name	CovParm	Subject	Estimate	StdErr	ZValue	ProbZ	PseudoR2
CovRandLinLog	UN(1,1)	PersonID	274252	41240	6.65	<.0001	.
CovRandLinLog	UN(2,2)	PersonID	23101	4882.26	4.73	<.0001	.
CovRandLinLog	session	PersonID	24120	1697.11	14.21	<.0001	.
CovFixQuadLog	UN(1,1)	PersonID	274453	41239	6.66	<.0001	-0.000733
CovFixQuadLog	UN(2,2)	PersonID	23229	4880.99	4.76	<.0001	-0.005551
CovFixQuadLog	session	PersonID	23839	1679.38	14.20	<.0001	**0.011672**

Model 7d. Random Quadratic Log Time

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

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

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

$$\text{Quadratic LogTime: } \beta_{2i} = \gamma_{20} + U_{2i}$$

Predicted intercept at any occasion:
$= \gamma_{00} + \gamma_{10}(\text{LogTime}_{ti}) + \gamma_{20}(\text{LogTime}_{ti})^2$

Instantaneous linear time slope at any occasion:
$= \gamma_{10} + 2\gamma_{20}(\text{LogTime}_{ti})$

```
TITLE1 "SAS: 7d: Random Quadratic Log Time Model";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime logtime*logtime / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT logtime logtime*logtime / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED session / R TYPE=VC SUBJECT=PersonID;
     ODS OUTPUT InfoCrit=FitRandQuadLog; * Save for LRT;
RUN;
TITLE1 "Calculate LRT -- does random quadratic log time slope improve model fit?";
%FitTest(FitFewer=FitFixQuadLog, FitMore=FitRandQuadLog); TITLE1;
```

```
display "STATA: 7d: Random Quadratic Log Time Model"
mixed rt c.logtime c.logtime#c.logtime, ///
         || personid: logtime logtimesq, variance reml covariance(unstructured) ///
         residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2                   // Print -2LL for model
      estat ic, n(101)                             // AIC and BIC
      estat recovariance, relevel(personid)        // G matrix
      estat recovariance, relevel(personid) correlation // GCORR matrix
      estat wcorrelation, covariance               // V matrix
      estat wcorrelation                           // VCORR matrix
      estimates store FitRandQuadLog       // Save for LRT
      lrtest FitRandQuadLog FitFixQuadLog  // Does random quadratic log time slope improve model fit?
```

```
print("R 7d: Random Quadratic Log Time Model -- reports convergence problem")
RandQuadLog = lmer(data=Example6, REML=TRUE,
formula=rt~1+logtime+I(logtime^2)+(1+logtime+I(logtime^2)|PersonID))
print("Show results using Satterthwaite DDF including -2LL as deviance")
summary(RandQuadLog, ddf="Satterthwaite"); llikAIC(RandQuadLog, chkREML=FALSE)
print("Does random quadratic time slope improve model fit?")
ranova(RandQuadLog, reduce.term=TRUE) # Remove random slope and covariance
```

SAS Output:

Estimated G Matrix

Row	Effect	PersonID	Col1	Col2	Col3
1	Intercept	101	285072	-80403	15959
2	logtime	101	-80403	199200	-86830
3	logtime*logtime	101	**15959**	**-86830**	**43025**

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1	Col2	Col3
1	Intercept	101	1.0000	-0.3374	0.1441
2	logtime	101	-0.3374	1.0000	-0.9379
3	logtime*logtime	101	**0.1441**	**-0.9379**	1.0000

The **level-2 G matrix** (always **unstructured**) still contains variances for the random intercept (1,1) and random linear log time slope (2,2), as well as their covariance (2,1). We have now added a variance for the random quadratic log time slope (3,3) and its covariances with the random intercept (3,1) and random linear log time slope (3,2) for the same person. **GCORR** provides the corresponding correlations among the random effects.

Estimated V Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	**302304**	237009	216002	204280	197007	192244
2	237009	**253981**	226105	213856	201669	189971
3	216002	226105	**236995**	209779	198899	187976
4	204280	213856	209779	**219785**	194472	186263
5	197007	201669	198899	194472	**206863**	184764
6	192244	189971	187976	186263	184764	**200661**

The marginal **V** matrix now predicts the variances to change in a **quartic pattern ($\mathbf{logtime^4}$)**.

Estimated V Correlation Matrix for PersonID 101

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.8553	0.8070	0.7925	0.7878	0.7805
2	0.8553	1.0000	0.9216	0.9052	0.8798	0.8415
3	0.8070	0.9216	1.0000	0.9192	0.8983	0.8620
4	0.7925	0.9052	0.9192	1.0000	0.9120	0.8869
5	0.7878	0.8798	0.8983	0.9120	1.0000	0.9069
6	0.7805	0.8415	0.8620	0.8869	0.9069	1.0000

The marginal **VCORR** matrix now predicts that the covariance changes over time in a more complex time-dependent pattern than for random linear log time.

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z	
UN(1,1)	PersonID	285072	42684	6.68	<.0001	Level-2 random intercept variance of U_{0i}
UN(2,1)	PersonID	-80403	31879	-2.52	0.0117	Level-2 random intercept-linear slope covariance
UN(2,2)	PersonID	199200	42606	4.68	<.0001	Level-2 random linear log time slope variance of U_{1i}
UN(3,1)	PersonID	15959	14893	1.07	0.2839	Level-2 random intercept-quadratic slope covariance
UN(3,2)	PersonID	-86830	20318	-4.27	<.0001	Level-2 random linear-quadratic slope covariance
UN(3,3)	PersonID	43025	10274	4.19	<.0001	Level-2 random quadratic time slope variance of U_{2i}
session	PersonID	17232	1399.97	12.31	<.0001	Level-1 residual variance of e_{ti}

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
6	920.29	<.0001

This LRT tells us whether we need the full 3x3 **G** matrix (so it is not helpful).

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
8266.1	7	8280.1	8280.3	8287.5	8298.4	8305.4

```
                    Solution for Fixed Effects
                                Standard
Effect             Estimate        Error      DF    t Value    Pr > |t|
Intercept           1960.96      54.6521     100      35.88      <.0001   gamma00
logtime             -240.61      54.1434     100      -4.44      <.0001   gamma10
logtime*logtime     46.9146      26.4888     100       1.77      0.0796   gamma20
```

```
Calculate LRT -- does random quadratic log time slope improve model fit?
Likelihood Ratio Test for FitFixQuadLog vs. FitRandQuadLog
                   Neg2Log
     Name           Like      Parms       AIC        BIC     DevDiff     DFdiff       Pvalue
FitFixQuadLog      8309.9       4       8317.9     8328.4       .           .              .
FitRandQuadLog     8266.1       7       8280.1     8298.4    43.7825        3        1.6786E-9
```

Is the random quadratic time slope significant? How do we know?

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

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

$$\text{Intercept 95\% CI} = \gamma_{00} \pm \left(1.96*\sqrt{\tau^2_{U_0}}\right) \rightarrow 1{,}960.96 \pm \left(1.96*\sqrt{285{,}072}\right) = 914 \text{ to } 3{,}007$$

$$\text{Linear LogTime Slope 95\% CI} = \gamma_{10} \pm \left(1.96*\sqrt{\tau^2_{U_1}}\right) \rightarrow -240.61 \pm \left(1.96*\sqrt{199{,}200}\right) = -115 \text{ to } 634$$

$$\text{Quadratic LogTime Slope 95\% CI} = \gamma_{20} \pm \left(1.96*\sqrt{\tau^2_{U_2}}\right) \rightarrow 46.91 \pm \left(1.96*\sqrt{43025}\right) = -360 \text{ to } 453$$

Is it a problem that the CIs for the linear and quadratic log time slopes overlap 0? What does this mean?

The absolute fit of the quadratic log time model for the means can be tested by mimicking a saturated means model using the *same random quadratic log time slopes* (i.e., holding the variance model constant):

```
TITLE1 "SAS: Test Absolute Fit of Quadratic Log Time Means (Using Random Quadratic Log Time Variance)";
TITLE2 "Add 3 session dummy codes to saturate the means model";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime logtime*logtime s1 s2 s3 / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT logtime logtime*logtime / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED session / R TYPE=VC SUBJECT=PersonID;
     CONTRAST "Does fixed quadratic log time reproduce saturated means?" s1 1, s2 1, s3 1 / CHISQ;
RUN; TITLE1; TITLE2;
```

```
display "STATA: Test Absolute Fit of Quadratic Log Time Means (Using Random Quadratic Log Time Variance)"
display "Add 3 session dummy codes to saturate the means model"
mixed rt c.logtime c.logtime#c.logtime c.s1 c.s2 c.s3, ///
         || personid: logtime logtimesq, variance reml covariance(unstructured) ///
         residuals(independent,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2            // Print -2LL for model
      test (c.s1=0)(c.s2=0)(c.s3=0), small   // Does fixed quadratic log time reproduce saturated means?
```

```
print("R: Test Absolute Fit of Quadratic Log Time Means (Using Random Quadratic Log Time Variance)")
print("Add 3 session dummy codes to second piece to saturate the means model")
QuadLogMean = lmer(data=Example6, REML=TRUE,
                   formula=rt~1+logtime+I(logtime^2)+s1+s2+s3+(1+logtime+I(logtime^2)|PersonID))
print("Show results using Satterthwaite DDF including -2LL as deviance")
summary(QuadLogMean, ddf="Satterthwaite"); llikAIC(QuadLogMean, chkREML=FALSE)
print("Does fixed quadratic log time reproduce saturated means?")
contestMD(QuadLogMean, 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)))
```

SAS Output (relevant tables only):

Because there are now **6 fixed effects for the 6 means**, this model is equivalent to **saturated means** (even if the linear and quadratic fixed 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 using CONTRAST indicates that the 3 extra session contrasts did not improve model fit (which is good news here).

```
                     Solution for Fixed Effects
                             Standard
Effect            Estimate      Error     DF    t Value    Pr > |t|
Intercept           988.14     986.36    302       1.00      0.3172
logtime            1020.69    1254.21    301       0.81      0.4164
logtime*logtime    -356.60     395.51    303      -0.90      0.3680
s1                  973.75     985.01    300       0.99      0.3237
s2                  290.87     306.89    300       0.95      0.3440
s3                 70.9505    87.1698    300       0.81      0.4163

                                                           Contrasts
                                                             Num    Den
Label                                                         DF     DF   Chi-Square   F Value   Pr > ChiSq   Pr > F
Does fixed quadratic log time reproduce saturated means?       3    300         1.31      0.44       0.7257   0.7259
```

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

```
TITLE1 "SAS: Test Absolute Fit of Quadratic Log Time Variance (Using Fixed Quadratic Log Time Means)";
TITLE2 "Change to Unstructured R matrix as variance model answer key";
PROC MIXED DATA=work.Example6 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID session;
     MODEL rt = logtime logtime*logtime / SOLUTION DDFM=Satterthwaite;
     REPEATED session / R RCORR TYPE=UN SUBJECT=PersonID;
     ODS OUTPUT InfoCrit=FitFixQuadLogUN; * Save for LRT;
RUN; TITLE2;
TITLE1 "Calculate LRT -- does random quadratic log time reproduce UN variance model?";
%FitTest(FitFewer=FitRandQuadLog, FitMore=FitFixQuadLogUN);

display "STATA: Test Absolute Fit of Quadratic Log Time Variance (Using Fixed Quadratic Log Time Means)"
display "Change to Unstructured R matrix as variance model answer key"
mixed rt c.logtime c.logtime#c.logtime, || personid: , noconstant variance reml ///
         residuals(unstructured,t(session)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2            // Print -2LL for model
      estimates store FitFixQuadLogUN       // Save for LRT
      lrtest FitFixQuadLogUN FitRandQuadLog // Does random quadratic log time reproduce UN variance?

print("R: Test Absolute Fit of Quadratic Log Time Variance (Using Fixed Quadratic Log Time Means)")
print("Change to Unstructured R matrix as variance model answer key in GLS")
QuadLogUN = gls(data=Example6, method="REML", model=rt~1+logtime+I(logtime^2),
                correlation=corSymm(form=~as.numeric(session)|PersonID),  # Unstructured correlations
                weights=varIdent(form=~1|session))                        # Heterogeneous variances
print("Have to re-run random quadratic log time model using LME to get LRT")
RandQuadLoglme = lme(data=Example6, method="REML", rt~1+logtime+I(logtime^2),
                     random=~1+logtime+I(logtime^2)|PersonID)
anova(QuadLogUN,RandQuadLoglme) # anova does LRT using LME versions
```

SAS Output (relevant table only):

```
Calculate LRT -- does random quadratic log time reproduce UN variance model?
Likelihood Ratio Test for FitRandQuadLog vs. FitRandUN
                  Neg2Log
Name                 Like     Parms       AIC        BIC    DevDiff    DFdiff     Pvalue
FitRandQuadLog     8266.1       7      8280.1     8298.4      .           .         .
FitRandUN          8254.1      21      8296.1     8351.1    11.9846      14     0.60754
```

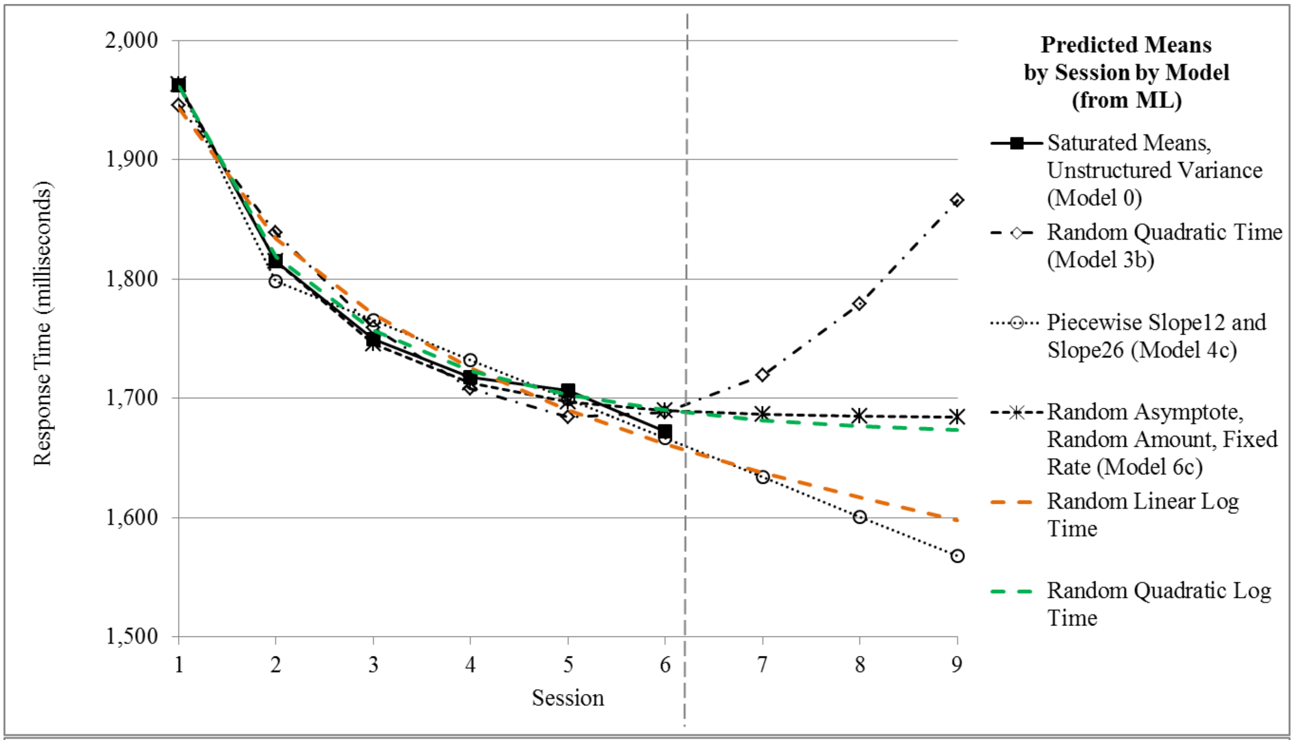
What does this nonsignificant LRT result indicate about our random quadratic log time model?

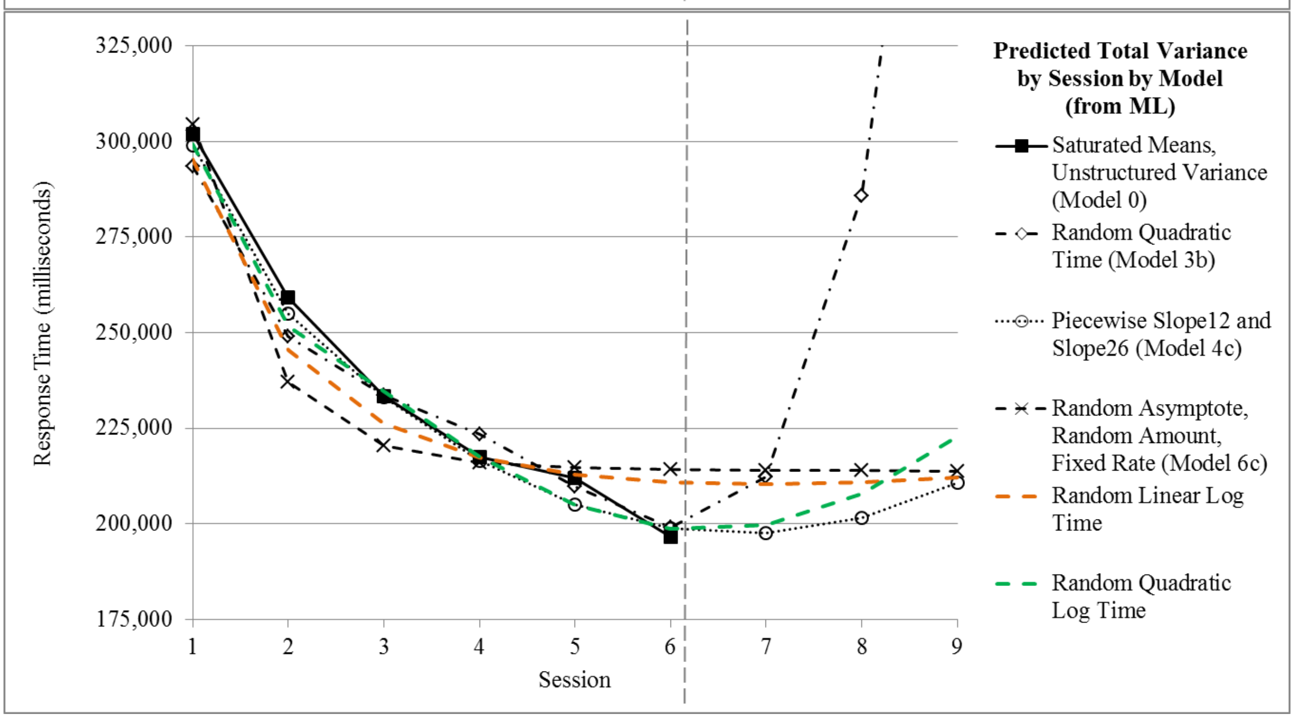
At last, the grand finale comparison: Tests of absolute fit of each side of the model for the "best" version of each model as presented in Examples 6a–6d, as well as tests of absolute fit of both sides of the model combined (see example syntax and output online for the latter)....

Summary of Tests of Absolute Model Fit For Each Side of the Model		Means Side Test			Variance Side Test		
		Chi-Square	Test DF	p-value	Chi-Square	Test DF	p-value
3b	Polynomial: Random Quadratic Time	9.07	3	0.030	35.76	14	0.001
4c	**Piecewise: Random Slope12, Random Slope26**	4.74	3	**0.195**	15.77	14	**0.327**
6c	Negative Exponential: Random Asymptote, Random Amount, Fixed Rate	0.87	3	**0.404**	46.50	17	0.000
7c	**Random Linear Log Time**	**6.69**	**4**	**0.155**	58.01	17	0.000
7d	**Random Quadratic Log Time**	**1.31**	**3**	**0.726**	**11.98**	**14**	**0.608**

Model		Total # Parameters	ML -2LL	ML AIC	ML BIC
7c	Random Linear Log Time	6	8340.49	8352.5	8368.2
6c	Negative Exponential: Random Asymptote, Random Amount, Fixed Rate	7	8327.30	8341.3	8359.6
3b	Polynomial: Random Quadratic Time	10	8321.77	8341.8	8367.9
4c	Piecewise: Random Slope12, Random Slope26	10	8298.95	8318.9	8345.1
7d	**Random Quadratic Log Time**	10	8291.51	**8311.5**	**8337.7**
8	**Latent Basis**	10	8326.19	8346.2	8372.3
0	Least Parsimonious Baseline: Saturated Means, Unstructured Variance	27	**8278.09**	8332.1	8402.7
0	Absolute Fit worse than Saturated Means, Unstructured Variance Model?	df	Chi-Square	p-value	
7c	Random Linear Log Time	21	62.41	.000	
6c	Negative Exponential: Random Asymptote, Random Amount, Fixed Rate	20	49.21	.000	
3b	Polynomial: Random Quadratic Time	17	43.68	.000	
4c	Piecewise: Random Slope12, Random Slope26	17	20.86	.233	
7d	**Random Quadratic Log Time**	**17**	**13.42**	**.708**	
8	**Latent Basis**	**17**	**48.10**	**.000**	

Predicted means/variances all in ML for comparability





I think random quadratic log time wins—and that's going in chapter 6 when I revise my book!

For sample results sections that go with Example 6, please see the end of Chapter 6.