Example 5: Practice with Fixed and Random Effects of Time in Modeling Within-Person Change
(complete data, syntax, and output available for SAS, STATA, R, and Mplus electronically)

The models for this example come from Hoffman (2015) chapter 5. We will be examining the extent to which change in a test score outcome across four annual occasions can be described with fixed and random linear effects of time (indexed by years in study, in which 0 is baseline) in a sample of 25 persons. For an example results section, see the end of chapter 5. **NEW for 2022:** See the last section for the same models estimated as single-level structural equation models using maximum likelihood instead (in which the random effect variances are downwardly biased given the *N*= 25 sample).

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_Example5;
* Location for SAS files for these models (uses macro variable filesave);
LIBNAME filesave "&filesave.";

* Import chapter 5 stacked data into work library and center time at first occasion;
DATA work.Example5; SET filesave.SAS_Chapter5;
     time = wave - 1; LABEL time= "time: Time in Study (0=1)";
RUN;
```

STATA Syntax for Data Import and Manipulation:

```
// Define working directory for file location to be replaced in code below
   cd "C:\Dropbox\22_PSQF6271\PSQF6271_Example5"

// Import chapter 5 stacked data and center time at first occasion
   use "$STATA_Chapter5.dta", clear
   gen time = wave - 1
   label variable time "time: Time in Study (0=1)"
```

R Syntax for Data Import and Manipulation:

```
# Define variables for working directory and data name
filesave = "C:\\Dropbox/22_PSQF6271/PSQ6271_Example5/"
filename = "SAS_Chapter5.sas7bdat"
setwd(dir=filesave)

# Import chapter 5 stacked data with labels
Example5 = read_sas(data_file=paste0(filesave,filename))
# Convert to data frame as data frame without labels to use for analysis
Example5 = as.data.frame(Example5)

# Sort data by PersonID (needed for correct RCOV matrix)
Example5 = sort_asc(Example5,PersonID,wave)

# Center time at first occasion
Example5$time=Example5$wave-1
# Make new variable for wave with reference=4 to match other programs
Example5$wave4=relevel(factor(Example5$wave), ref=4)
```

A disclaimer about the R code in this example (and many subsequent longitudinal examples): I have been unsuccessful in finding an R package that does everything simultaneously that is available in SAS MIXED or STATA MIXED, and so this example uses three: GLS and LME (from NLME) and LMER (from LME4). As near as I can tell, GLS allows R-only models (whereas LME and LMER do not), LME allows G+R models with a correlation structure in R, and GLS and LME each easily display the model-predicted G, R, and V matrices. But I can't get either GLS or LME to provide the correct denominator degrees of freedom (DDF) for unstructured models for the variance. In contrast, LMER does provide correct DDF, but it does not allow any R correlation structures (VC diagonal only), and it does not easily provide the predicted V matrix. Consequently, I have provided two sets of code (using LME and LMER) for the example models that contain random effects—so refer to the LME output for the variance model predictions and to the LMER output for correct tests of the fixed effects. If anyone out there can help me solve this problem, please let me know!!!!

1. SAS, STATA, and R Syntax for a Saturated Means, Unstructured R-only Variance Model

This provides the ANSWER KEY for both the model for the means (via saturated means) and the model for the variance (via an unstructured **R** matrix of all possible variances and covariances), as called the "H1 model" in SEM terminology. This model is only possible to estimate directly (without rounding occasions) in balanced data. The predicted outcome from the (saturated) fixed effects is given by: $\widehat{outcome}_{ti} = \beta_0 + \beta_1(wave1_{ti}) + \beta_2(wave2_{ti}) + \beta_3(wave3_{ti})$, but the unstructured model for the variance cannot be easily summarized by scalar notation like this.

```
TITLE1 "SAS Ch 5: Saturated Means, Unstructured Variance Model";
TITLE2 "ANSWER KEY for both sides of the model";
PROC MIXED DATA=work.Example5 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID wave;
     MODEL outcome = wave / SOLUTION DDFM=Satterthwaite;
     REPEATED wave / R RCORR TYPE=UN SUBJECT=PersonID;
     LSMEANS wave / DIFF=ALL; * Means and mean differences per wave;
RUN; TITLE1; TITLE2;
```

```
display "STATA Ch 5: Saturated Means, Unstructured Variance Model"
display "ANSWER KEY for both sides of the model"
mixed outcome ib(last).wave, || personid: , noconstant variance reml  ///
               residuals(unstructured,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2        // Print -2LL for model
      estat ic, n(25)                   // AIC and BIC
      estat wcorrelation, covariance    // R matrix
      estat wcorrelation                // RCORR matrix
      contrast i.wave, small            // Omnibus F-test for wave
      margins  i.wave                   // Means per wave
      margins  i.wave, pwcompare(pveffects) df(24)  // Mean diffs by wave
```

```
print("R GLS Ch 5: Saturated Means, Unstructured Variance Model")
print("ANSWER KEY for both sides of the model")
SatUN = gls(data=Example5, method="REML", model=outcome~1+factor(wave4),
            correlation=corSymm(form=~as.numeric(wave4)|PersonID), # Unstructured correlations
            weights=varIdent(form=~1|wave4))                       # Heterogeneous variances
print("Show results with -2LL, incorrect DDF, and total leftover variance")
print("Total variance per occasion is created using SD multiplier")
summary(SatUN); -2*logLik(SatUN); summary(SatUN)$sigma^2

print("Show R and RCORR matrices for first person")
R=getVarCov(SatUN, individual="1", type="marginal"); R
RCORR=corMatrix(SatUN$modelStruct$corStruct)[[4]]; RCORR

print("Wave means and pairwise mean differences with Satterthwaite DDF")
lsmeans(SatUN, "wave4", mode="satterthwaite")
emmeans(ref_grid(SatUN), pairwise~wave4, adjust="none")
print("F-test p-value based on residual DDF instead"); anova(SatUN)
```

SAS Output:

```
             Dimensions
Covariance Parameters           10
Columns in X                     5
Columns in Z                     0
Subjects                        25
Max Obs Per Subject              4
```

```
          Estimated R Matrix for PersonID 1
 Row        Col1        Col2        Col3        Col4
   1      2.3618      2.7867      1.9566      2.4204
   2      2.7867      4.8900      4.0440      5.5525
   3      1.9566      4.0440      6.2172      7.7994
   4      2.4204      5.5525      7.7994     11.7437
```

Because this model uses REPEATED only (no RANDOM statement), the **R** matrix holds the total (marginal) variances and covariances over waves directly. Likewise, **RCORR** holds the total (marginal) correlations over waves directly.

Estimated R Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.8200	0.5106	0.4596
2	0.8200	1.0000	0.7334	0.7327
3	0.5106	0.7334	1.0000	0.9128
4	0.4596	0.7327	0.9128	1.0000

Covariance Parameter Estimates (total variance and covariance per occasions)

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	PersonID	2.3618	0.6818	3.46	0.0003
UN(2,1)	PersonID	2.7867	0.8971	3.11	0.0019
UN(2,2)	PersonID	4.8900	1.4116	3.46	0.0003
UN(3,1)	PersonID	1.9566	0.8783	2.23	0.0259
UN(3,2)	PersonID	4.0440	1.3958	2.90	0.0038
UN(3,3)	PersonID	6.2172	1.7947	3.46	0.0003
UN(4,1)	PersonID	2.4204	1.1831	2.05	0.0408
UN(4,2)	PersonID	5.5525	1.9176	2.90	0.0038
UN(4,3)	PersonID	7.7994	2.3615	3.30	0.0010
UN(4,4)	PersonID	11.7437	3.3901	3.46	0.0003

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
9	108.30	<.0001

This is the test of whether we need *anything* beyond a constant residual variance σ_e^2 (df=9)… and we do.

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
353.8	10	373.8	376.3	377.1	385.9	395.9

Because we are using REML, only the variance model parameters count towards AIC and BIC.

Solution for Fixed Effects

Effect	wave (1-4)	Estimate	Standard Error	DF	t Value	Pr > |t|	
Intercept		15.5516	0.6854	24	22.69	<.0001	Beta0
wave	1	-5.1468	0.6088	24	-8.45	<.0001	Beta1
wave	2	-3.6940	0.4703	24	-7.86	<.0001	Beta2
wave	3	-1.9672	0.3074	24	-6.40	<.0001	Beta3
wave	4	0	.	.	.	.	

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
wave	3	24	23.86	<.0001

This is the ANOVA test of omnibus mean differences across wave (note numerator df=3 for the 4 means across waves), assuming an unstructured **R** matrix (multivariate ANOVA).

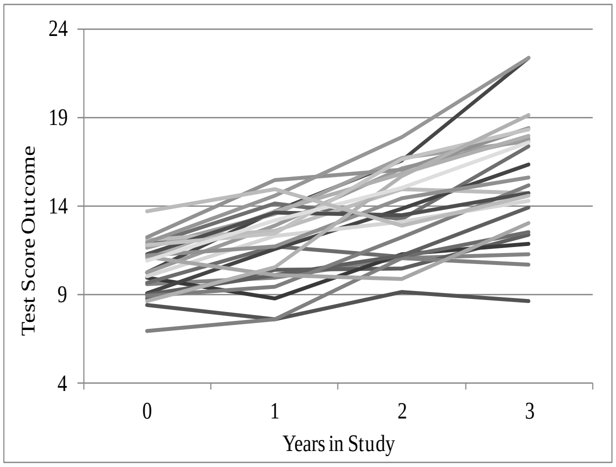
Because *wave* is on the CLASS statement, LSMEANS can provides mean per wave and pairwise mean differences.

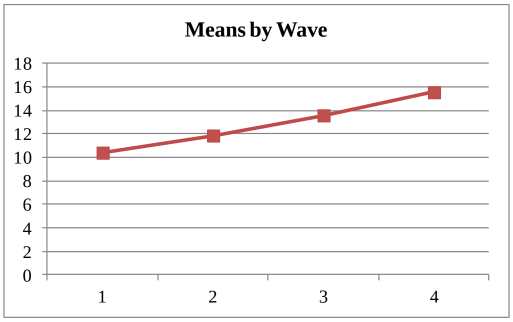
Least Squares Means

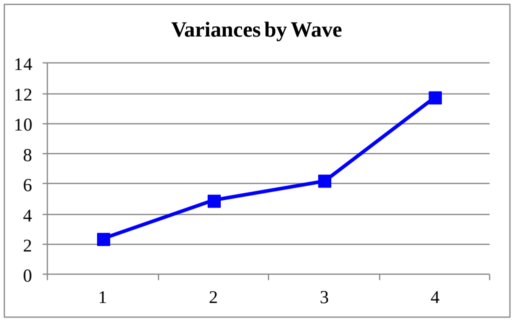
Effect	wave (1-4)	Estimate	Standard Error	DF	t Value	Pr > |t|	
wave	1	10.4048	0.3074	24	33.85	<.0001	Beta0+Beta1
wave	2	11.8576	0.4423	24	26.81	<.0001	Beta0+Beta2
wave	3	13.5844	0.4987	24	27.24	<.0001	Beta0+Beta3
wave	4	15.5516	0.6854	24	22.69	<.0001	Beta0

Differences of Least Squares Means

Effect	wave (1-4)	wave (1-4)	Estimate	Standard Error	DF	t Value	Pr > |t|
wave	1	2	-1.4528	0.2591	24	-5.61	<.0001
wave	1	3	-3.1796	0.4320	24	-7.36	<.0001
wave	1	4	-5.1468	0.6088	24	-8.45	<.0001
wave	2	3	-1.7268	0.3475	24	-4.97	<.0001
wave	2	4	-3.6940	0.4703	24	-7.86	<.0001
wave	3	4	-1.9672	0.3074	24	-6.40	<.0001







2. SAS, STATA, and R Syntax for a Saturated Means, Random Intercept Model

If an unstructured **R** matrix was *not* possible to estimate, I'd still examine the answer key for the model for the means (via a saturated means model) but estimate a random intercept only (which should always be possible).

$$outcome_{ti} = \beta_0 + \beta_1(wave1_{ti}) + \beta_2(wave2_{ti}) + \beta_3(wave3_{ti}) + U_{0i} + e_{ti}$$

```
TITLE1 "SAS Saturated Means, Random Intercept Model";
TITLE2 "ANSWER KEY for means side only";
PROC MIXED DATA=work.Example5 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID wave;
     MODEL outcome = wave / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT / G V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED wave / R TYPE=VC SUBJECT=PersonID;
     LSMEANS wave / DIFF=ALL; RUN; TITLE1; TITLE2;
```

```
display "STATA Ch 5: Saturated Means, Random Intercept Model"
display "ANSWER KEY for means side only"
mixed outcome ib(last).wave, || personid: , variance reml  ///
              residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2       // Print -2LL for model
      estat ic, n(25)                  // AIC and BIC
      estat icc                        // Conditional intraclass correlation
      estat wcorrelation, covariance   // R matrix
      estat wcorrelation               // RCORR matrix
      contrast i.wave, small           // Omnibus F-test for wave
      margins  i.wave                  // Means per wave
      margins  i.wave, pwcompare(pveffects) df(72)  // Mean diffs by wave
```

```
print("R LME Ch 5: Saturated Means, Random Intercept Model")
print("ANSWER KEY for means side only")
SatRI = lme(data=Example5, method="REML", outcome~1+factor(wave4), random=~1|PersonID)
print("Show results with -2LL and incorrect DDF")
summary(SatRI); -2*logLik(SatRI)
print("Show G, R, and V matrices for first person")
G=getVarCov(SatRI, individual="1", type="random.effects"); G
R=getVarCov(SatRI, individual="1", type="conditional"); R
V=getVarCov(SatRI, individual="1", type="marginal"); V
ICC=(V[[1]][2,1])/(V[[1]][1,1]); print("Show Conditional ICC"); ICC
print("Wave means, pairwise mean differences, and omnibus F-test")
emmeans(ref_grid(SatRI), pairwise~wave4, adjust="none"); anova(SatRI)
```

Here is the same model using the LME4 function LMER (to get the correct Satterthwaite DDF) instead of the NLME function LME (which more easily provides the V and R matrices for pedagogical purposes):

```
print("R LMER Ch 5: Saturated Means, Random Intercept Model")
SatRIr = lmer(data=Example5, REML=TRUE, formula=outcome~1+factor(wave4)+(1|PersonID))
print("Show results with -2LL using correct Satterthwaite DDF")
summary(SatRIr, ddf="Satterthwaite"); llikAIC(SatRIr, chkREML=FALSE)
print("Show Conditional ICC"); icc(SatRIr)
print("Wave means, pairwise mean differences, and omnibus F-test")
emmeans(ref_grid(SatRIr), pairwise~wave4, adjust="none"); anova(SatRIr)
```

SAS Output:

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	6.3032	4.0933	4.0933	4.0933
2	4.0933	6.3032	4.0933	4.0933
3	4.0933	4.0933	6.3032	4.0933
4	4.0933	4.0933	4.0933	6.3032

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.6494	0.6494	0.6494
2	0.6494	1.0000	0.6494	0.6494
3	0.6494	0.6494	1.0000	0.6494
4	0.6494	0.6494	0.6494	1.0000

The ICC is always given in the VCORR matrix. In this model, it is a *conditional* ICC (after controlling for fixed effects for the predictors for wave mean differences). Make sure you report what model the ICC is from to avoid confusion.

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z	
UN(1,1)	PersonID	4.0933	1.3443	3.04	0.0012	L2 random intercept U_{0i} variance in G
wave	PersonID	2.2099	0.3683	6.00	<.0001	L1 residual e_{ti} variance in R

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
1	49.51	<.0001

This likelihood ratio test says we need a random intercept (e-only fits significantly worse), indicating the conditional ICC is significantly > 0.

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
412.5	2	416.5	416.7	417.2	419.0	421.0

Solution for Fixed Effects

Effect	wave (1-4)	Estimate	Standard Error	DF	t Value	Pr > |t|	
Intercept		15.5516	0.5021	42.4	30.97	<.0001	Beta0
wave	1	-5.1468	0.4205	72	-12.24	<.0001	Beta1
wave	2	-3.6940	0.4205	72	-8.79	<.0001	Beta2
wave	3	-1.9672	0.4205	72	-4.68	<.0001	Beta3
wave	4	0	.	.	.	.	

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
wave	3	72	55.82	<.0001

This is the ANOVA test of omnibus mean differences across wave (note df=3 for the 4 means across waves), assuming a random intercept only (CS **V** matrix, which is univariate RM ANOVA).

Least Squares Means

Effect	wave (1-4)	Estimate	Standard Error	DF	t Value	Pr > |t|	
wave	1	10.4048	0.5021	42.4	20.72	<.0001	Beta0+Beta1
wave	2	11.8576	0.5021	42.4	23.62	<.0001	Beta0+Beta2
wave	3	13.5844	0.5021	42.4	27.05	<.0001	Beta0+Beta3
wave	4	15.5516	0.5021	42.4	30.97	<.0001	Beta0

Differences of Least Squares Means

Effect	wave (1-4)	wave (1-4)	Estimate	Standard Error	DF	t Value	Pr > |t|
wave	1	2	-1.4528	0.4205	72	-3.46	0.0009
wave	1	3	-3.1796	0.4205	72	-7.56	<.0001
wave	1	4	-5.1468	0.4205	72	-12.24	<.0001
wave	2	3	-1.7268	0.4205	72	-4.11	0.0001
wave	2	4	-3.6940	0.4205	72	-8.79	<.0001
wave	3	4	-1.9672	0.4205	72	-4.68	<.0001

---

3. SAS, STATA, and R Syntax for Equation 5.1: Empty Means, Random Intercept Model

```
TITLE1 "SAS Eq 5.1: Empty Means, Random Intercept Model";
PROC MIXED DATA=work.Example5 COVTEST NOCLPRINT
           NAMELEN=100 IC METHOD=REML;
     CLASS PersonID wave;
     MODEL outcome = / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT / G V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED wave / R TYPE=VC SUBJECT=PersonID; RUN;
```

Level 1: $y_{ti} = \beta_{0i} + e_{ti}$

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

Composite: $y_{ti} = \left(\gamma_{00} + U_{0i}\right) + e_{ti}$

```
display "STATA Eq 5.1: Empty Means, Random Intercept Model"
mixed outcome , || personid: , variance reml covariance(unstructured) ///
              residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2              // Print -2LL for model
      estat ic, n(25)                         // AIC and BIC
      estat icc                               // Intraclass correlation
      estat recovariance, relevel(personid)   // G matrix
      estat wcorrelation, covariance          // V matrix
      estat wcorrelation                      // VCORR matrix
```

```
print("R LME Eq 5.1: Empty Means, Random Intercept Model")
EmptyRI = lme(data=Example5, method="REML", outcome~1, random=~1|PersonID)
print("Show results with -2LL and incorrect DDF"); summary(EmptyRI); -2*logLik(EmptyRI)
print("Show G, R, and V matrices for first person")
G=getVarCov(EmptyRI, individual="1", type="random.effects"); G
R=getVarCov(EmptyRI, individual="1", type="conditional"); R
V=getVarCov(EmptyRI, individual="1", type="marginal"); V
ICC=(V[[1]][2,1])/(V[[1]][1,1]); print("Show Unconditional ICC"); ICC

print("R LMER Eq 5.1: Empty Means, Random Intercept Model")
EmptyRIr = lmer(data=Example5, REML=TRUE, formula=outcome~1+(1|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
summary(EmptyRIr, ddf="Satterthwaite"); llikAIC(EmptyRIr, chkREML=FALSE)
print("Show Unconditional ICC"); icc(EmptyRI)
print("Does random intercept improve model fit?")
ranova(EmptyRIr, reduce.term=TRUE) # LRT for removing random intercept
```

SAS Output:

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	7.0554			
2		7.0554		
3			7.0554	
4				7.0554

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1
1	Intercept	1	2.8819

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	9.9373	2.8819	2.8819	2.8819
2	2.8819	9.9373	2.8819	2.8819
3	2.8819	2.8819	9.9373	2.8819
4	2.8819	2.8819	2.8819	9.9373

Because this model uses the REPEATED and RANDOM statements, the **V** matrix holds the total (marginal) predicted variances and covariances over waves (from putting **G** and **R** back together through the **Z** matrix). Likewise, **VCORR** holds the total predicted correlations over waves.

```
     Estimated V Correlation Matrix for PersonID 1
 Row        Col1        Col2        Col3        Col4
   1      1.0000      0.2900      0.2900      0.2900
   2      0.2900      1.0000      0.2900      0.2900
   3      0.2900      0.2900      1.0000      0.2900
   4      0.2900      0.2900      0.2900      1.0000
```

VCORR provides the ICC as: IntVar/TotalVar
Notice how much lower this unconditional ICC is relative to the conditional ICC from saturated means (i.e., this random intercept variance is much smaller before the residual variance due to mean differences over time has been explained).

```
                  Covariance Parameter Estimates
                                       Standard         Z
Cov Parm    Subject       Estimate       Error      Value       Pr > Z
UN(1,1)     PersonID        2.8819      1.3717       2.10       0.0178  L2 random intercept U0i variance in G
wave        PersonID        7.0554      1.1521       6.12       <.0001  L1 residual eti variance in R

  Null Model Likelihood Ratio Test
    DF    Chi-Square      Pr > ChiSq
     1          9.79          0.0018
```

This is the test of whether we need the random intercept variance (so df=1)… and we do. This means the unconditional ICC is > 0.

```
                             Information Criteria
Neg2LogLike     Parms         AIC        AICC        HQIC         BIC        CAIC
      502.2         2       506.2       506.3       506.9       508.7       510.7

                   Solution for Fixed Effects
                         Standard
Effect        Estimate      Error      DF    t Value    Pr > |t|
Intercept      12.8496     0.4311      24      29.81      <.0001 gamma00
```

---

4. SAS, STATA, and R Syntax for Equation 5.3: Fixed Linear Time, Random Intercept Model

```
TITLE1 "SAS Eq 5.3: Fixed Linear Time, Random Intercept Model";
PROC MIXED DATA=work.Example5 COVTEST NOCLPRINT
           NAMELEN=100 IC METHOD=REML;
  CLASS PersonID wave;
  MODEL outcome = time / SOLUTION DDFM=Satterthwaite;
  RANDOM INTERCEPT / G V VCORR TYPE=UN SUBJECT=PersonID;
  REPEATED wave / R TYPE=VC SUBJECT=PersonID;
* Save output for computations;
  ODS OUTPUT CovParms=CovFixLin InfoCrit=FitFixLin;
  ESTIMATE "Pred Mean at Time=0"  int 1 time 0;
  ESTIMATE "Pred Mean at Time=1"  int 1 time 1;
  ESTIMATE "Pred Mean at Time=2"  int 1 time 2;
  ESTIMATE "Pred Mean at Time=3"  int 1 time 3;
RUN;
* Call macro to calculate pseudo R2;
  %PseudoR2(NCov=2, CovFewer=CovEmpty, CovMore=CovFixLin);
```

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

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

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

Composite: $y_{ti} = \left(\gamma_{00} + U_{0i}\right) + \gamma_{10}\left(\text{Time}_{ti}\right) + e_{ti}$

Note the two different versions of the "time" variable in the SAS syntax. Both are necessary here because they do different things. "**Wave**" is treated as a **categorical** predictor, and it is being used as a level-1 ID variable to structure the **R** matrix in the event of missing occasions. Therefore, "**wave**" goes on the **CLASS** and **REPEATED** statements. In contrast, "**time**" is treated as a **quantitative** predictor, and its role is to index linear effects of time (and it is centered such that wave 1 = time 0). Accordingly, in the **ESTIMATE** statements, only one value after "**time**" is needed.

```
display "STATA Eq 5.3: Fixed Linear Time, Random Intercept Model"
mixed outcome c.time, || personid: , variance reml ///
              residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2             // Print -2LL for model
      estat ic, n(25)                        // AIC and BIC
      estat recovariance, relevel(personid)  // G matrix
      estat wcorrelation, covariance         // V matrix
      estat wcorrelation                     // VCORR matrix
      lincom _cons*1 + c.time*0, small       // Pred mean at Time=0
      lincom _cons*1 + c.time*1, small       // Pred mean at Time=1
      lincom _cons*1 + c.time*2, small       // Pred mean at Time=2
```

```
lincom _cons*1 + c.time*3, small       // Pred mean at Time=3
margins, at(c.time=(0(1)3)) vsquish    // Pred mean at all times (start(by)end)
estimates store FitFixLin              // Save for LRT
```

```
print("R LME Eq 5.3: Fixed Linear Time, Random Intercept Model")
FixLinRI = lme(data=Example5, method="REML", outcome~1+time, random=~1|PersonID)
print("Show results using incorrect DDF"); summary(FixLinRI); -2*logLik(FixLinRI)
print("Show G, R, and V matrices for first person")
G=getVarCov(FixLinRI, individual="1", type="random.effects"); G
R=getVarCov(FixLinRI, individual="1", type="conditional"); R
V=getVarCov(FixLinRI, individual="1", type="marginal"); V

print("R LMER Eq 5.3: Fixed Linear Time, Random Intercept Model")
FixLinRIr = lmer(data=Example5, REML=TRUE, formula=outcome~1+time+(1|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
summary(FixLinRIr, ddf="Satterthwaite"); llikAIC(FixLinRIr, chkREML=FALSE)
print("Get conditional mean per occasion from value of time predictor")
print("Pred Mean at Time 0"); contest1D(FixLinRIr, ddf="Satterthwaite", L=c(1,0))
print("Pred Mean at Time=1"); contest1D(FixLinRIr, ddf="Satterthwaite", L=c(1,1))
print("Pred Mean at Time=2"); contest1D(FixLinRIr, ddf="Satterthwaite", L=c(1,2))
print("Pred Mean at Time=3"); contest1D(FixLinRIr, ddf="Satterthwaite", L=c(1,3))
print("Predicted outcome means using margins package (but SEs don't work)")
prediction(model=FixLinRIr, calculate_se=TRUE, at=list(time=0:3))
```

SAS Output:

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	2.1725			
2		2.1725		
3			2.1725	
4				2.1725

After controlling for the fixed linear effect of time, the residual variance was reduced from σ_e^2 = 7.06 in the empty means, random intercept model to σ_e^2 = 2.17 in this model. This is a pseudo-R^2 reduction of (7.06 – 2.17) / 7.06 = .69 (or 69% of the residual variance is accounted for by a fixed linear time).

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1
1	Intercept	1	4.1026

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	6.2751	4.1026	4.1026	4.1026
2	4.1026	6.2751	4.1026	4.1026
3	4.1026	4.1026	6.2751	4.1026
4	4.1026	4.1026	4.1026	6.2751

However, the random intercept variance actually increased from 2.88 to 4.10. This is because of how $\tau_{U_0}^2$ is found:

true $\tau_{U_0}^2$= observed $\tau_{U_0}^2$– (σ_e^2/ n)

So reducing σ_e^2 will make $\tau_{U_0}^2$ (and the conditional ICC, shown in VCORR) increase!

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.6538	0.6538	0.6538
2	0.6538	1.0000	0.6538	0.6538
3	0.6538	0.6538	1.0000	0.6538
4	0.6538	0.6538	0.6538	1.0000

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z	
UN(1,1)	PersonID	4.1026	1.3441	3.05	0.0011	L2 random intercept U_{0i} variance in G
wave	PersonID	2.1725	0.3572	6.08	<.0001	L1 residual e_{ti} variance in R

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
1	51.12	<.0001

This LRT tests whether we need the random intercept variance (so df=1)… and we (still) do.

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
415.1	2	419.1	419.2	419.8	421.5	423.5

```
                 Solution for Fixed Effects
                      Standard
Effect       Estimate    Error     DF    t Value    Pr > |t|
Intercept     10.2745   0.4743   34.7      21.66      <.0001 gamma00
time           1.7167   0.1318     74      13.02      <.0001 gamma10

          Estimates → These are the predicted outcome means from a fixed linear time model
                                Standard
Label                 Estimate     Error     DF    t Value    Pr > |t|
Pred Mean at Time=0    10.2745    0.4743   34.7      21.66      <.0001 gamma00 + gamma10(0)
Pred Mean at Time=1    11.9912    0.4361   25.1      27.50      <.0001 gamma00 + gamma10(1)
Pred Mean at Time=2    13.7080    0.4361   25.1      31.43      <.0001 gamma00 + gamma10(2)
Pred Mean at Time=3    15.4247    0.4743   34.7      32.52      <.0001 gamma00 + gamma10(3)

PsuedoR2 (% Reduction) for CovEmpty vs. CovFixLin → These come from my SAS %FitTest macro
  Name       CovParm    Subject     Estimate     StdErr    ZValue     ProbZ     PseudoR2
CovEmpty     UN(1,1)    PersonID      2.8819     1.3717      2.10    0.0178        .
CovEmpty     wave       PersonID      7.0554     1.1521      6.12    <.0001        .
CovFixLin    UN(1,1)    PersonID      4.1026     1.3441      3.05    0.0011     -0.42359
CovFixLin    wave       PersonID      2.1725     0.3572      6.08    <.0001      0.69208
```

---

5. SAS, STATA, and R Syntax for Equation 5.5: Random Linear Time Model

```
TITLE1 "SAS Eq 5.5: Random Linear Time Model";
PROC MIXED DATA= work.Example5 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
  CLASS PersonID wave;
  MODEL outcome = time / SOLUTION DDFM=Satterthwaite;
  RANDOM INTERCEPT time / G V GCORR VCORR TYPE=UN SUBJECT=PersonID;
  REPEATED wave / R TYPE=VC SUBJECT=PersonID;
  ODS OUTPUT InfoCrit=FitRandLin; * Save for computations;
  ESTIMATE "Pred Mean at Time=0" int 1 time 0;
  ESTIMATE "Pred Mean at Time=1" int 1 time 1;
  ESTIMATE "Pred Mean at Time=2" int 1 time 2;
  ESTIMATE "Pred Mean at Time=3" int 1 time 3;
RUN;
* Does random linear time slope improve model fit?
%FitTest(FitFewer=FitFixLin, FitMore=FitRandLin);
```

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

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

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

Composite: $y_{ti} = \left(\gamma_{00} + U_{0i}\right) + \left(\gamma_{10} + U_{1i}\right)\left(Time_{ti}\right) + e_{ti}$

Note that the “time” variable gets included in the SAS RANDOM statement, not “wave”—including “wave” would result in model non-convergence, because it would try to estimate a random slope variance for each possible difference between waves (instead of a single variance for a single linear random slope throughout).

```
display "STATA Eq 5.5: Random Linear Time Model"
mixed outcome c.time, || personid: time, variance reml covariance(unstructured) ///
             residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2                          // Print -2LL for model
      estat ic, n(25)                                     // 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
      margins, at(c.time=(0(1)3)) vsquish     // Pred mean at all times (start(by)end)
      estimates store FitRandLin              // Save for LRT
      lrtest FitRandLin FitFixLin             // Does random linear time slope improve fit?
```

```
print("R LME Eq 5.5: Random Linear Time Model")
RandLin = lme(data=Example5, method="REML", outcome~1+time, random=~1+time|PersonID)
print("Show results with -2LL using incorrect DDF"); summary(RandLin); -2*logLik(RandLin)
print("Show G, R, and V matrices for first person")
G=getVarCov(RandLin, individual="1", type="random.effects"); G
R=getVarCov(RandLin, individual="1", type="conditional"); R
V=getVarCov(RandLin, individual="1", type="marginal"); V
```

```
print("R LMER Eq 5.5: Random Linear Time Model")
RandLinr = lmer(data=Example5, REML=TRUE, formula=outcome~1+time+(1+time|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
summary(RandLinr, ddf="Satterthwaite"); llikAIC(RandLinr, chkREML=FALSE)
print("Get conditional mean per occasion from value of time predictor")
print("Pred Mean at Time 0"); contest1D(RandLinr, ddf="Satterthwaite", L=c(1,0))
print("Pred Mean at Time=1"); contest1D(RandLinr, ddf="Satterthwaite", L=c(1,1))
print("Pred Mean at Time=2"); contest1D(RandLinr, ddf="Satterthwaite", L=c(1,2))
print("Pred Mean at Time=3"); contest1D(RandLinr, ddf="Satterthwaite", L=c(1,3))
print("Predicted outcome means using margins package (but SEs don't work)")
prediction(model= RandLinr, calculate_se=TRUE, at=list(time=0:3))
print("Does random linear time slope improve fit?")
ranova(RandLinr, reduce.term=TRUE) # Remove random slope and covariance
```

SAS Output:

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	0.6986			
2		0.6986		
3			0.6986	
4				0.6986

After adding a random linear effect of time, the residual variance is smaller, but it is not correct to say that it has been reduced. Random effects do not explain variance; they simply re-allocate it. Here, this means that part of what was residual is now individual differences in the linear effect of time as a new pile of variance in the **G** matrix below.

Estimated G Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	1	2.2624	0.05454
2	time	1	0.05454	0.9089

The **G** matrix provides the variances and covariances of the individual random effects. Now **G** is a 2x2 matrix because we have 2 random effects (intercept, linear slope).

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	1	1.0000	0.03803
2	time	1	0.03803	1.0000

The **GCORR** matrix provides the correlation(s) among the individual random effects. Here, the individual intercepts and slopes are correlated r = .04.

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	2.9611	2.3170	2.3715	2.4260
2	2.3170	3.9790	4.2438	5.2073
3	2.3715	4.2438	6.8148	7.9885
4	2.4260	5.2073	7.9885	11.4684

The **V** matrix holds the total (marginal) variances and covariances over waves (from putting **G** and **R** back together through the **Z** matrix). Likewise, **VCORR** holds the total correlations over waves. Note that all of these are now predicted to differ as a function of which wave it is (see Table 5.2 for a description of how this works).

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.6750	0.5279	0.4163
2	0.6750	1.0000	0.8150	0.7709
3	0.5279	0.8150	1.0000	0.9036
4	0.4163	0.7709	0.9036	1.0000

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z	
UN(1,1)	PersonID	2.2624	0.8003	2.83	0.0023	L2 random intercept U_{0i} variance in G
UN(2,1)	PersonID	0.05454	0.3507	0.16	0.8764	L2 random intercept-slope covariance in G
UN(2,2)	PersonID	0.9089	0.3040	2.99	0.0014	L2 random linear time slope U_{1i} var in G
wave	PersonID	0.6986	0.1397	5.00	<.0001	L1 residual e_{ti} variance in R

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	99.47	<.0001

This tests whether we need *anything* in the **G** matrix (so df=3). Note this does NOT tell us if we need the random linear time slope specifically! We have to do a separate LRT to answer that question (see my %FitTest macro results below).

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
366.7	4	374.7	375.2	376.1	379.6	383.6

```
                    Solution for Fixed Effects
                          Standard
Effect        Estimate       Error      DF    t Value    Pr > |t|
Intercept      10.2745      0.3318      24      30.97      <.0001 gamma00
time            1.7167      0.2048      24       8.38      <.0001 gamma10

                Estimates → These are the predicted outcome means from a random linear time model
                                    Standard
Label                     Estimate       Error      DF    t Value    Pr > |t|
Intercept at Time 0        10.2745      0.3318      24      30.97      <.0001 gamma00 + gamma10(0)
Intercept at Time 1        11.9912      0.3736      24      32.09      <.0001 gamma00 + gamma10(1)
Intercept at Time 2        13.7080      0.5030      24      27.25      <.0001 gamma00 + gamma10(2)
Intercept at Time 3        15.4247      0.6711      24      22.98      <.0001 gamma00 + gamma10(3)

Likelihood Ratio Test for FitFixLin vs. FitRandLin (from %FitTest macro)
              Neg2Log
   Name          Like       Parms        AIC         BIC      DevDiff     DFdiff        Pvalue
FitFixLin       415.1         2        419.1       421.5        .            .               .
FitRandLin      366.7         4        374.7       379.6     48.3539         2      3.1629E-11
```

A random time slope helps the model.

Two Ways of Conveying Effect Size for This Model's Random Effects:

(1) 95% Random Effects Confidence Intervals that describe the *predicted* range of *individual* random effects:

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

$$\text{Intercept 95\% CI} = \gamma_{00} \pm \left(1.96 * \sqrt{\tau^2_{U_0}}\right) = 10.27 \pm \left(1.96 * \sqrt{2.26}\right) = 7.32 \text{ to } 13.22$$

$$\text{Linear Time Slope 95\% CI} = \gamma_{10} \pm \left(1.96 * \sqrt{\tau^2_{U_1}}\right) = 1.72 \pm \left(1.96 * \sqrt{0.91}\right) = -0.15 \text{ to } 3.59$$

(2) Intercept Reliability (IR; ICC2) and Slope Reliability (SR) using these formulae from Lecture 5 slide 38 (1.26 = variance of the *time* predictor variable, as found by requesting its descriptive statistics):

$$\text{IR} = \frac{\tau^2_{U0}}{\tau^2_{U0} + \frac{\sigma^2_e}{L1n}} = \frac{2.26}{2.26 + \frac{.70}{4}} = .93 \qquad \text{SR} = \frac{\tau^2_{U1}}{\tau^2_{U1} + \frac{\sigma^2_e}{L1n * \sigma^2_{L1}}} = \frac{0.91}{0.91 + \frac{.70}{4 * 1.26}} = .87$$

Last but not least: there may still be residual covariances after modeling individual differences in the linear effect of time (i.e., adding a random linear time slope to the **G** matrix). We can test alternative **R** matrix assumptions besides VC (which assumes no residual covariance/correlation over time) to see if this is the case:

6. SAS, STATA, and R Syntax for Random Linear Time in G + Auto-Regressive Residual Corr in R

```
TITLE1 "SAS Ch 5: Random Linear Time Model + AR1 R Matrix";
PROC MIXED DATA= work.Example5 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID wave;
     MODEL outcome = time / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT time / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED wave / R RCORR TYPE=AR(1) SUBJECT=PersonID;
     ODS OUTPUT InfoCrit=FitRandLinAR1; * Save for computations;
RUN;
* Does AR1 residual correlation improve fit?;
  %FitTest(FitFewer=FitRandLin, FitMore=FitRandLinAR1);
```

```
display "STATA Ch 5: Random Linear Time Model with AR1 R Matrix"
mixed outcome c.time,  || personid: time, variance reml covariance(unstructured) ///
           residuals(ar1,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
    display "-2LL = " e(ll)*-2                               // Print -2LL for model
    estat ic, n(25)                                          // 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 FitRandLinAR1     // Save for LRT
    lrtest FitRandLinAR1 FitRandLin   // Does AR1 residual correlation improve fit?
```

```
print("R Ch 5: LME Random Linear Time Model + AR1 R Matrix")
RandLinAR1 = lme(data=Example5, method="REML", outcome~1+time, random=~1+time|PersonID,
                 correlation=(corAR1(form=~as.numeric(time)|PersonID)))
print("Show results using incorrect DDF"); summary(RandLinAR1)
print("Show G, R, and V matrices for first person")
G=getVarCov(RandLinAR1, individual="1", type="random.effects"); G
R=getVarCov(RandLinAR1, individual="1", type="conditional"); R
V=getVarCov(RandLinAR1, individual="1", type="marginal"); V
print("Does AR1 residual correlation improve fit?")
anova(RandLinAR1, RandLin) # anova compares using LME versions
```

SAS Output:

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	0.7193	0.01841	0.000471	0.000012
2	0.01841	0.7193	0.01841	0.000471
3	0.000471	0.01841	0.7193	0.01841
4	0.000012	0.000471	0.01841	0.7193

Estimated R Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.02560	0.000655	0.000017
2	0.02560	1.0000	0.02560	0.000655
3	0.000655	0.02560	1.0000	0.02560
4	0.000017	0.000655	0.02560	1.0000

The AR1 correlation shows up in the **R** matrix for the lag-1 correlation (with AR^2 for lag-2 and AR^3 for lag-3).

Estimated G Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	1	2.2216	0.06949
2	time	1	0.06949	0.9015

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	1	1.0000	0.04910
2	time	1	0.04910	1.0000

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	2.9409	2.3095	2.3610	2.4301
2	2.3095	3.9814	4.2516	5.2046
3	2.3610	4.2516	6.8250	7.9967
4	2.4301	5.2046	7.9967	11.4717

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.6749	0.5270	0.4184
2	0.6749	1.0000	0.8156	0.7701
3	0.5270	0.8156	1.0000	0.9037
4	0.4184	0.7701	0.9037	1.0000

```
                 Covariance Parameter Estimates
                                      Standard         Z
Cov Parm      Subject      Estimate      Error     Value        Pr Z
UN(1,1)       PersonID       2.2216     1.2181      1.82      0.0341 L2 random intercept U0i variance in G
UN(2,1)       PersonID      0.06949     0.4842      0.14      0.8859 L2 random intercept-slope covariance in G
UN(2,2)       PersonID       0.9015     0.3459      2.61      0.0046 L2 random linear time slope U1i var in G
AR(1)         PersonID      0.02560     0.5688      0.05      0.9641 L1 auto-regressive correlation
Residual                     0.7193     0.4969      1.45      0.0739 L1 residual eti variance in R

  Null Model Likelihood Ratio Test
    DF     Chi-Square      Pr > ChiSq
     4          99.48          <.0001

                           Information Criteria
Neg2LogLike      Parms        AIC        AICC        HQIC        BIC        CAIC
      366.7          5      376.7       377.4       378.4      382.8       387.8

                                      Solution for Fixed Effects
                             Standard
Effect         Estimate         Error       DF     t Value     Pr > |t|      Alpha        Lower        Upper
Intercept       10.2763        0.3308     20.2       31.06       <.0001       0.05       9.5867      10.9658
time             1.7167        0.2047     23.7        8.39       <.0001       0.05       1.2940       2.1394

Likelihood Ratio Test for FitRandLin vs. FitRandLinAR1
                Neg2Log
    Name           Like       Parms         AIC         BIC        DevDiff     DFdiff       Pvalue
FitRandLin        366.7          4        374.7       379.6    .                  .            .
FitRandLinAR1     366.7          5        376.7       382.8    .002107160         1        0.96339
```

Adding an AR1 correlation to the **R** matrix does not help.

---

7. SAS and STATA Syntax for Random Linear Time in G + Toeplitz Lag-1 Residual Covariance in R

```
TITLE1 "SAS Random Linear Time Model + TOEP2 R Matrix";
PROC MIXED DATA=example5 COVTEST NOCLPRINT NAMELEN=100 IC METHOD=REML;
     CLASS PersonID wave;
     MODEL outcome = time / SOLUTION DDFM=Satterthwaite;
     RANDOM INTERCEPT time / G GCORR V VCORR TYPE=UN SUBJECT=PersonID;
     REPEATED wave / R RCORR TYPE=TOEP(2) SUBJECT=PersonID;
     ODS OUTPUT InfoCrit=FitRandLinTOEP2; * Save for computations;
RUN;
* Does lag-1 residual correlation improve fit?;
  %FitTest(FitFewer=FitRandLin, FitMore=FitRandLinTOEP2);


display "STATA Ch 5: Random Linear Time Model with TOEP2 R Matrix"
mixed outcome c.time, || personid: time, variance reml covariance(unstructured) ///
               residuals(toeplitz1,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
      display "-2LL = " e(ll)*-2                             // Print -2LL for model
      estat ic, n(25)                                        // 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 FitRandLinTOEP2    // Save for LRT
      lrtest FitRandLinTOEP2 FitRandLin // Does lag-1 residual correlation improve fit?

# Toeplitz is not a pre-defined structure in R LME
```

SAS Output:

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	0.7127	0.01259		
2	0.01259	0.7127	0.01259	
3		0.01259	0.7127	0.01259
4			0.01259	0.7127

The Toeplitz lag-1 covariance shows up in the **R** matrix for adjacent occasions only.

Estimated R Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.01766		
2	0.01766	1.0000	0.01766	
3		0.01766	1.0000	0.01766
4			0.01766	1.0000

Estimated G Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	1	2.2342	0.06496
2	time	1	0.06496	0.9038

Estimated G Correlation Matrix

Row	Effect	PersonID	Col1	Col2
1	Intercept	1	1.0000	0.04571
2	time	1	0.04571	1.0000

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	2.9470	2.3118	2.3641	2.4291
2	2.3118	3.9807	4.2493	5.2055
3	2.3641	4.2493	6.8220	7.9944
4	2.4291	5.2055	7.9944	11.4709

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.6750	0.5273	0.4178
2	0.6750	1.0000	0.8154	0.7703
3	0.5273	0.8154	1.0000	0.9037
4	0.4178	0.7703	0.9037	1.0000

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z	
UN(1,1)	PersonID	2.2342	1.0786	2.07	0.0192	L2 random intercept U_{0i} variance in G
UN(2,1)	PersonID	0.06496	0.4407	0.15	0.8828	L2 random intercept-slope covariance in G
UN(2,2)	PersonID	0.9038	0.3309	2.73	0.0032	L2 random linear time slope U_{1i} var in G
TOEP(2)	PersonID	0.01259	0.3232	0.04	0.9689	L1 lag-1 residual covariance in R
Residual		0.7127	0.3908	1.82	0.0341	L1 residual e_{ti} variance in R

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
4	99.48	<.0001

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
366.7	5	376.7	377.4	378.4	382.8	387.8

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Intercept	10.2757	0.3311	21.3	31.04	<.0001	0.05	9.5878	10.9636
time	1.7167	0.2047	23.8	8.39	<.0001	0.05	1.2940	2.1394

```
Likelihood Ratio Test for FitRandLin vs. FitRandLinTOEP2
                        Neg2Log
Name                     Like      Parms       AIC        BIC        DevDiff     DFdiff      Pvalue
FitRandLin               366.7       4       374.7      379.6     .                 .          .
FitRandLinTOEP2          366.7       5       376.7      382.8     .001489262        1        0.96922
```

Adding a lag-1 correlation to the **R** matrix does not help.

So how do we know that this model is "good enough" in terms of fit: (a) of the fixed linear time slope for predicting the means for each wave, and (b) of the level-2 random intercept, level-2 random linear time slope, the covariance of the level-2 random intercept and linear time slope, and level-1 residual for predicting the variances and covariances across waves? This is trickier to do when using REML but not impossible—stay tuned for a demonstration in Example 6!

Single-Level SEM Versions of All Models (after Reshaping: Long → Wide)

Below: Mplus Syntax Only; R Syntax and Output (full results available in electronic files)

```
TITLE:    PSQF 6271 Example 5: Single-Level SEM Versions in Mplus
DATA:     FILE = MPLUS_Chapter5.csv;    ! Syntax in same folder as data

! Unstacking from long to format
DATA LONGTOWIDE:
! Names of old stacked former variables (without numbers)
  LONG = wave|outcome;
! Names of new multivariate variables (that use numbers)
  WIDE = time0-time3|outcome0-outcome3;
! Variable with level-2 ID info
  IDVARIABLE = PersonID;
! Old level-1 identifier
  REPETITION = wave (1 2 3 4);

VARIABLE:
! List of variables in original LONG data file
  NAMES = PersonID wave outcome;
! Variables to be analyzed in this model
  USEVARIABLE = outcome0-outcome3;
! Missing data identifier
  MISSING = ALL (-999);

ANALYSIS:    TYPE = GENERAL; ESTIMATOR = ML; MODEL = NOCOVARIANCES;
OUTPUT:      RESIDUAL;  ! Model-implied means, variances, covs, corrs

MODEL:  !!! Model-specific code goes here, to be replaced per model
```

```
# R: Un-Stack from long to wide format (becomes one row per person)
Example5_wide = reshape(Example5, direction="wide",
                        v.names="outcome", idvar="PersonID", timevar="time")
```

---

1. Saturated Means, Unstructured Variance (Total "Answer Key") Model

This provides the ANSWER KEY for both the model for the means (via saturated means) and the model for the variance (via an unstructured **R** matrix of all possible variances and covariances), as called the "H1 model" in SEM terminology. This model is only possible to estimate directly (without rounding occasions) in balanced data. The predicted outcome from the (saturated) fixed effects is given by: $\widehat{outcome}_{ti} = \beta_0 + \beta_1(wave1_{ti}) + \beta_2(wave2_{ti}) + \beta_3(wave3_{ti})$, but the unstructured model for the variance cannot be easily summarized by scalar notation like this.

```
!!!! Mplus Ch 5: Saturated Means, Unstructured Variance Model
! Occasion means all estimated
  [outcome0-outcome3];
! Occasion variances all estimated
  outcome0-outcome3;
! Occasion covariances all estimated
  outcome0-outcome3 WITH outcome0-outcome3;
```

```
print("R lavaan Ch 5: Saturated Means, Unstructured Variance Model")
print("ANSWER KEY for both sides of the model")
SyntaxSatUN = "
# Occasion means all estimated
  outcome.0 ~ 1; outcome.1 ~ 1; outcome.2  ~ 1; outcome.3 ~ 1
# Occasion variances all estimated
  outcome.0 ~~ outcome.0; outcome.1 ~~ outcome.1; outcome.2 ~~ outcome.2; outcome.3 ~~ outcome.3
# Occasion covariances all estimated
  outcome.0 ~~ outcome.1; outcome.0 ~~ outcome.2; outcome.0 ~~ outcome.3
  outcome.1 ~~ outcome.2; outcome.1 ~~ outcome.3; outcome.2 ~~ outcome.3
"
ModelSatUN = lavaan(model=SyntaxSatUN, data=Example5_wide, estimator="ML", mimic="mplus")
summary(ModelSatUN, fit.measures=TRUE, standardized=FALSE)
```

```
Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  outcome.0 ~~
    outcome.1         2.675    0.844    3.170    0.002
    outcome.2         1.878    0.826    2.274    0.023
    outcome.3         2.324    1.113    2.088    0.037
  outcome.1 ~~
    outcome.2         3.882    1.313    2.957    0.003
    outcome.3         5.330    1.804    2.955    0.003
  outcome.2 ~~
    outcome.3         7.487    2.221    3.371    0.001

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
    outcome.0        10.405    0.301   34.550    0.000
    outcome.1        11.858    0.433   27.364    0.000
    outcome.2        13.584    0.489   27.802    0.000
    outcome.3        15.552    0.672   23.158    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    outcome.0         2.267    0.641    3.536    0.000
    outcome.1         4.694    1.328    3.536    0.000
    outcome.2         5.968    1.688    3.536    0.000
    outcome.3        11.274    3.189    3.536    0.000
```

```
print("Model-implied marginal means, variances, and covariances; add correlations")
fitted(object=ModelSatUN); lavInspect(object=ModelSatUN, "cor.ov")
```

```
$cov → This is sigma from the H1 model
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0  2.267
outcome.1  2.675  4.694
outcome.2  1.878  3.882  5.968
outcome.3  2.324  5.330  7.487 11.274

$mean
outcome.0 outcome.1 outcome.2 outcome.3
   10.405    11.858    13.584    15.552

Correlations: Standardized H1 sigma
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 1.000
outcome.1 0.820  1.000
outcome.2 0.511  0.733  1.000
outcome.3 0.460  0.733  0.913  1.000
```

Because the SEM is estimated using ML instead of REML, the variance–covariance matrix entries are all under-estimated— here is the REML version of **V** from SAS MIXED for comparison:

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4
1	2.3618	2.7867	1.9566	2.4204
2	2.7867	4.8900	4.0440	5.5525
3	1.9566	4.0440	6.2172	7.7994
4	2.4204	5.5525	7.7994	11.7437

The means are estimated accurately, but their standard errors are a little too small (because all the ML variances are too small, *N* = 25).

2. Saturated Means, Random Intercept Model ("Answer Key" for Means Side Only)

If an unstructured **R** matrix was *not* possible to estimate, I'd still examine the answer key for the model for the means (via a saturated means model) but estimate a random intercept only (which should always be possible).

$$outcome_{ti} = \beta_0 + \beta_1(wave1_{ti}) + \beta_2(wave2_{ti}) + \beta_3(wave3_{ti}) + U_{0i} + e_{ti}$$

```
!!!! Mplus Ch 5: Saturated Means, Random Intercept Model
! Define intercept factor by fixing all factor loadings to 1
  FactInt BY outcome0-outcome3@1;
! Fix factor mean to 0 to estimate per-occasion intercepts instead
  [FactInt@0];
! Estimate intercept factor variance = random intercept variance
  FactInt (IntVar);
! Occasion intercepts all estimated
  [outcome0-outcome3];
! Occasion residual variances all constrained equal
  outcome0-outcome3 (ResVar);
```

```
print("R lavaan Ch 5: Saturated Means, Random Intercept Model")
print("ANSWER KEY for means side only")
SyntaxSatRI = "
# Define intercept factor by fixing all factor loadings to 1
  FactInt =~ 1*outcome.0 + 1*outcome.1 + 1*outcome.2 + 1*outcome.3
# Fix factor mean to 0 to estimate per-occasion intercepts instead
  FactInt ~ 0
# Estimate intercept factor variance = random intercept variance
  FactInt ~~ (IntVar)*FactInt
# Occasion intercepts all estimated
  outcome.0 ~ 1; outcome.1 ~ 1; outcome.2  ~ 1; outcome.3 ~ 1
# Occasion residual variances all constrained equal
  outcome.0 ~~ (ResVar)*outcome.0; outcome.1 ~~ (ResVar)*outcome.1
  outcome.2 ~~ (ResVar)*outcome.2; outcome.3 ~~ (ResVar)*outcome.3
"
ModelSatRI = lavaan(model=SyntaxSatRI, data=Example5_wide, estimator="ML", mimic="mplus")
summary(ModelSatRI, fit.measures=TRUE, standardized=FALSE)
```

```
Latent Variables:   Estimate  Std.Err  z-value  P(>|z|)
  FactInt =~
    outcome.0          1.000
    outcome.1          1.000
    outcome.2          1.000
    outcome.3          1.000

Intercepts:         Estimate  Std.Err  z-value  P(>|z|)
    FactInt            0.000
   .outcome.0         10.405    0.492   21.149    0.000
   .outcome.1         11.858    0.492   24.102    0.000
   .outcome.2         13.584    0.492   27.612    0.000
   .outcome.3         15.552    0.492   31.610    0.000

Variances:Estimate  Std.Err  z-value  P(>|z|)
    FactInt (IntV)     3.930    1.264    3.108    0.002 → vs 4.0933 in REML
   .outcm.0 (RsVr)     2.121    0.346    6.124    0.000 → vs 2.2099 in REML
   .outcm.1 (RsVr)     2.121    0.346    6.124    0.000
   .outcm.2 (RsVr)     2.121    0.346    6.124    0.000
   .outcm.3 (RsVr)     2.121    0.346    6.124    0.000
```

```
print("Model-implied marginal means, variances, and covariances; add correlations")
fitted(object=ModelSatRI); lavInspect(object=ModelSatRI, "cor.ov")
```

```
$cov → Compound symmetry sigma
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 6.051
outcome.1 3.930  6.051
outcome.2 3.930  3.930  6.051
outcome.3 3.930  3.930  3.930  6.051
```

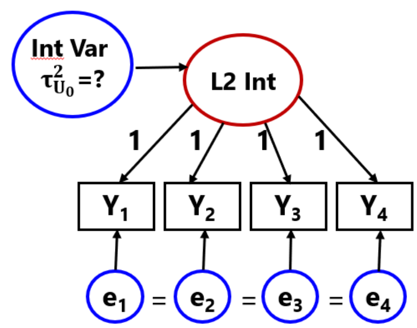
```
$mean → From saturated means
outcome.0 outcome.1 outcome.2 outcome.3
   10.405    11.858    13.584    15.552

Correlations → Conditional ICC
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 1.000
outcome.1 0.649  1.000
outcome.2 0.649  0.649  1.000
outcome.3 0.649  0.649  0.649  1.000
```

---

3. Equation 5.1: Empty Means, Random Intercept Model

Level 1:	$y_{ti} = \beta_{0i} + e_{ti}$
Level 2:	$\beta_{0i} = \gamma_{00} + U_{0i}$
Composite:	$y_{ti} = (\gamma_{00} + U_{0i}) + e_{ti}$




```
!!!! Mplus Ch 5: Empty Means, Random Intercept Model
! Define intercept factor by fixing all factor loadings to 1
  FactInt BY outcome0-outcome3@1;
! Estimate intercept factor mean = fixed intercept
  [FactInt] (FixInt);
! Estimate intercept factor variance = random intercept variance
  FactInt (IntVar);
! Occasion intercepts all fixed to 0
  [outcome0-outcome3@0];
! Occasion residual variances all constrained equal
  outcome0-outcome3 (ResVar);
```

```
print("R lavaan Eq 5.1: Empty Means, Random Intercept Model")
SyntaxEmptyRI = "
# Define intercept factor by fixing all factor loadings to 1
  FactInt =~ 1*outcome.0 + 1*outcome.1 + 1*outcome.2 + 1*outcome.3
# Estimate intercept factor mean = fixed intercept
  FactInt ~ (FixInt)*1
# Estimate intercept factor variance = random intercept variance
  FactInt ~~ (IntVar)*FactInt
# Occasion intercepts all fixed to 0
  outcome.0 ~ 0; outcome.1 ~ 0; outcome.2  ~ 0; outcome.3 ~ 0
# Occasion residual variances all constrained equal
  outcome.0 ~~ (ResVar)*outcome.0; outcome.1 ~~ (ResVar)*outcome.1
  outcome.2 ~~ (ResVar)*outcome.2; outcome.3 ~~ (ResVar)*outcome.3
"
ModelEmptyRI = lavaan(model=SyntaxEmptyRI, data=Example5_wide, estimator="ML", mimic="mplus")
summary(ModelEmptyRI, fit.measures=TRUE, standardized=FALSE)
```

```
Latent Variables:   Estimate  Std.Err  z-value  P(>|z|)
  FactInt =~
    outcome.0          1.000
    outcome.1          1.000
    outcome.2          1.000
    outcome.3          1.000

Intercepts:          Estimate  Std.Err  z-value  P(>|z|)
    FactInt (FxIn)   12.850    0.422   30.423    0.000
   .outcm.0           0.000
   .outcm.1           0.000
   .outcm.2           0.000
   .outcm.3           0.000

Variances:          Estimate  Std.Err  z-value  P(>|z|)
    FactInt (IntV)    2.696    1.294    2.084    0.037 → vs 2.8819 in REML
   .outcm.0 (RsVr)    7.055    1.152    6.124    0.000
   .outcm.1 (RsVr)    7.055    1.152    6.124    0.000
   .outcm.2 (RsVr)    7.055    1.152    6.124    0.000
   .outcm.3 (RsVr)    7.055    1.152    6.124    0.000
```

```
print("Model-implied marginal means, variances, and covariances; add correlations=ICC")
fitted(object=ModelEmptyRI); lavInspect(object=ModelEmptyRI, "cor.ov")
```

```
$cov → V is compound symmetry
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 9.751
outcome.1 2.696  9.751
outcome.2 2.696  2.696  9.751
outcome.3 2.696  2.696  2.696  9.751
$mean → From empty means
outcome.0 outcome.1 outcome.2 outcome.3
    12.85     12.85     12.85     12.85

Correlations → VCORR unconditional ICC
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 1.000
outcome.1 0.276  1.000
outcome.2 0.276  0.276  1.000
outcome.3 0.276  0.276  0.276  1.000
```

VCORR provides the ICC as: IntVar/TotalVar
Notice how much lower this unconditional ICC is relative to the conditional ICC from saturated means (i.e., this random intercept variance is much smaller before the residual variance due to mean differences over time has been explained).

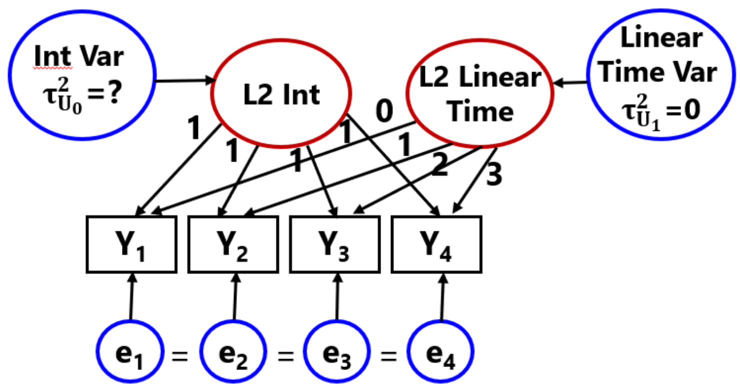
4. Equation 5.3: Fixed Linear Time, Random Intercept Model

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

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

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

Composite: $y_{ti} = \left(\gamma_{00} + U_{0i}\right) + \gamma_{10}\left(\text{Time}_{ti}\right) + e_{ti}$




```
!!!! Mplus Eq 5.3: Fixed Linear Time,
     Random Intercept Model
! Define intercept factor by fixing all
  factor loadings to 1
  FactInt BY outcome0-outcome3@1;
! Define linear time slope factor by fixing factor loadings to time values
  FactLin BY outcome0@0 outcome1@1 outcome2@2 outcome3@3;
! Estimate factor means = fixed intercept and fixed linear time slope
  [FactInt FactLin] (FixInt FixLin);
! Estimate intercept factor variance = random intercept variance
  FactInt (IntVar);
! Fix linear time slope factor variance to 0 (not random yet)
  FactLin@0;
! Occasion intercepts all fixed to 0
  [outcome0-outcome3@0];
! Occasion residual variances all constrained equal
  outcome0-outcome3 (ResVar);
```

Note that Mplus TSCORES would be needed if we have unbalanced time instead (for person-specific factor loadings of time passed).

```
print("R lavaan Eq 5.3: Fixed Linear Time, Random Intercept Model")
SyntaxFixLinRI = "
# Define intercept factor by fixing all factor loadings to 1
  FactInt =~ 1*outcome.0 + 1*outcome.1 + 1*outcome.2 + 1*outcome.3
# Define linear time slope factor by fixing factor loadings to time values
  FactLin =~ 0*outcome.0 + 1*outcome.1 + 2*outcome.2 + 3*outcome.3
# Estimate factor means = fixed intercept and fixed linear time slope
  FactInt ~ (FixInt)*1; FactLin ~ (FixLin)*1
# Estimate intercept factor variance = random intercept variance
  FactInt ~~ (IntVar)*FactInt
# Fix linear time slope factor variance to 0 (not random yet)
  FactLin ~~ 0*FactLin
# Occasion intercepts all fixed to 0
  outcome.0 ~ 0; outcome.1 ~ 0; outcome.2  ~ 0; outcome.3 ~ 0
# Occasion residual variances all constrained equal
  outcome.0 ~~ (ResVar)*outcome.0; outcome.1 ~~ (ResVar)*outcome.1
  outcome.2 ~~ (ResVar)*outcome.2; outcome.3 ~~ (ResVar)*outcome.3
"
ModelFixLinRI = lavaan(model=SyntaxFixLinRI, data=Example5_wide, estimator="ML", mimic="mplus")
summary(ModelFixLinRI, fit.measures=TRUE, standardized=FALSE)
```

```
Latent Variables:   Estimate  Std.Err  z-value  P(>|z|)
  FactInt =~
    outcome.0          1.000
    outcome.1          1.000
    outcome.2          1.000
    outcome.3          1.000
  FactLin =~
    outcome.0          0.000
    outcome.1          1.000
    outcome.2          2.000
    outcome.3          3.000

Intercepts:         Estimate  Std.Err  z-value  P(>|z|)
    FactInt (FxIn)    10.275    0.466   22.057    0.000
    FactLin (FxLn)     1.717    0.131   13.109    0.000
   .outcm.0            0.000
   .outcm.1            0.000
   .outcm.2            0.000
   .outcm.3            0.000

Variances:          Estimate  Std.Err  z-value  P(>|z|)
    FactInt (IntV)     3.924    1.264    3.103    0.002 → vs 4.1026 in REML
    FactLin            0.000                            → Random linear slope variance = 0
   .outcm.0 (RsVr)     2.144    0.350    6.124    0.000 → vs 2.1725 in REML
   .outcm.1 (RsVr)     2.144    0.350    6.124    0.000
   .outcm.2 (RsVr)     2.144    0.350    6.124    0.000
   .outcm.3 (RsVr)     2.144    0.350    6.124    0.000
```

```
print("Model-implied marginal means, variances, and covariances; add correlations")
fitted(object=ModelFixLinRI); lavInspect(object=ModelFixLinRI, "cor.ov")
```

```
$cov → V is still compound symmetry
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 6.068
outcome.1 3.924  6.068
outcome.2 3.924  3.924  6.068
outcome.3 3.924  3.924  3.924  6.068

$mean → predicted by linear time slope
outcome.0 outcome.1 outcome.2 outcome.3
   10.275    11.991    13.708    15.425

Correlations → VCORR conditional ICC
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 1.000
outcome.1 0.647  1.000
outcome.2 0.647  0.647  1.000
outcome.3 0.647  0.647  0.647  1.000
```

---

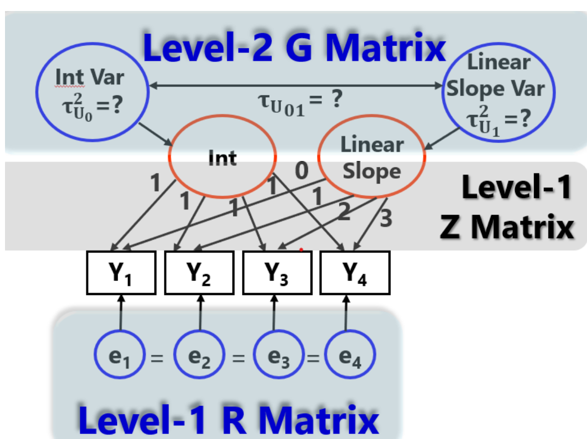
5. Equation 5.5: Random Linear Time Model

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

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

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

Composite: $y_{ti} = \left(\gamma_{00} + U_{0i}\right) + \left(\gamma_{10} + U_{1i}\right)\left(\text{Time}_{ti}\right) + e_{ti}$



```
!!!! Mplus Eq 5.5: Random Linear Time Model
! Define intercept factor by fixing all factor loadings to 1
  FactInt BY outcome0-outcome3@1;
! Define linear time slope factor by fixing factor loadings to time values
  FactLin BY outcome0@0 outcome1@1 outcome2@2 outcome3@3;
! Estimate factor means = fixed intercept and fixed linear time slope
  [FactInt FactLin] (FixInt FixLin);
! Estimate intercept factor variance = random intercept variance
  FactInt (IntVar);
! Estimate linear time slope factor variance = random slope variance
  FactLin (LinVar);
! Estimate random effects covariance
  FactInt WITH FactLin (ILcov);
! Occasion intercepts all fixed to 0
  [outcome0-outcome3@0];
! Occasion residual variances all constrained equal
  outcome0-outcome3 (ResVar);
```

Note that Mplus TSCORES would be needed if we have unbalanced time instead (for person-specific factor loadings of time passed).

```
print("R lavaan Eq 5.5: Random Linear Time Model")
SyntaxRandLin = "
# Define intercept factor by fixing all factor loadings to 1
  FactInt =~ 1*outcome.0 + 1*outcome.1 + 1*outcome.2 + 1*outcome.3
# Define linear time slope factor by fixing factor loadings to time value
  FactLin =~ 0*outcome.0 + 1*outcome.1 + 2*outcome.2 + 3*outcome.3
# Estimate factor means = fixed intercept and fixed linear time slope
  FactInt ~ (FixInt)*1; FactLin ~ (FixLin)*1
# Estimate intercept factor variance = random intercept variance
  FactInt ~~ (IntVar)*FactInt
# Estimate linear time slope factor variance = random slope variance
  FactLin ~~ (LinVar)*FactLin
# Estimate random effects covariance
  FactInt ~~ (ILcov)*FactLin
# Occasion intercepts all fixed to 0
  outcome.0 ~ 0; outcome.1 ~ 0; outcome.2  ~ 0; outcome.3 ~ 0
# Occasion residual variances all constrained equal
  outcome.0 ~~ (ResVar)*outcome.0; outcome.1 ~~ (ResVar)*outcome.1
  outcome.2 ~~ (ResVar)*outcome.2; outcome.3 ~~ (ResVar)*outcome.3
"
ModelRandLin = lavaan(model=SyntaxRandLin, data=Example5_wide, estimator="ML", mimic="mplus")
summary(ModelRandLin, fit.measures=TRUE, standardized=FALSE)
```

```
Latent Variables:   Estimate  Std.Err  z-value  P(>|z|)
  FactInt =~
    outcome.0         1.000
    outcome.1         1.000
    outcome.2         1.000
    outcome.3         1.000
  FactLin =~
    outcome.0         0.000
    outcome.1         1.000
    outcome.2         2.000
    outcome.3         3.000

Covariances:        Estimate  Std.Err  z-value  P(>|z|)
  FactInt ~~
    FactLin (ILcv)    0.061    0.330    0.184    0.854

Intercepts:         Estimate  Std.Err  z-value  P(>|z|)
    FactInt (FxIn)   10.275    0.325   31.609    0.000
    FactLin (FxLn)    1.717    0.201    8.555    0.000
   .outcm.0           0.000
   .outcm.1           0.000
   .outcm.2           0.000
   .outcm.3           0.000

Variances:          Estimate  Std.Err  z-value  P(>|z|)
    FactInt (IntV)    2.152    0.753    2.857    0.004 → vs 2.2624 in REML
    FactLin (LnVr)    0.867    0.286    3.030    0.002 → vs 0.9089 in REML
   .outcm.0 (RsVr)    0.699    0.140    5.000    0.000 → vs 0.6986 in REML
   .outcm.1 (RsVr)    0.699    0.140    5.000    0.000
```

```
   .outcm.2 (RsVr)     0.699    0.140    5.000    0.000
   .outcm.3 (RsVr)     0.699    0.140    5.000    0.000
```

```
print("Model-implied marginal means, variances, and covariances; add correlations")
fitted(object=ModelRandLin); lavInspect(object=ModelRandLin, "cor.ov")
$cov → V = sigma = not compound symmetry
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0  2.851
outcome.1  2.213  3.839
outcome.2  2.274  4.069  6.562
outcome.3  2.335  4.996  7.658 11.018

$mean → Predicted by linear time slope
outcome.0 outcome.1 outcome.2 outcome.3
   10.275    11.991    13.708    15.425

Correlations → VCORR differ by pair
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 1.000
outcome.1 0.669  1.000
outcome.2 0.526  0.811  1.000
outcome.3 0.417  0.768  0.901  1.000
```

The **V** matrix holds the total (marginal) variances and covariances over waves (from putting **G** and **R** back together through the **Z** matrix). Likewise, **VCORR** holds the total correlations over waves. Note that all of these are now predicted to differ as a function of which wave it is (see Table 5.2 for a description of how this works).

```
print("Does random linear time slope improve fit?")
anova(ModelRandLin, ModelFixLinRI) # Remove random slope and covariance

Chi-Squared Difference Test (chi-square = 48.3539 in REML)

              Df     AIC     BIC   Chisq Chisq diff Df diff        Pr(>Chisq)
ModelRandLin   8 376.990 384.303 13.9907
ModelFixLinRI 10 421.009 425.884 62.0096    48.0189       2 0.000000000037396
```

---

6. Random Linear Time in G + Auto-Regressive Residual Correlation in R

```
#### R NOTE: I COULD NOT GET THE Random Linear Time Model + AR1 R Matrix MODEL TO WORK ####

!!!! Mplus Ch.5: Random Linear Time in G + AR1 Residual Correlation in R
! Define intercept factor by fixing all factor loadings to 1
  FactInt BY outcome0-outcome3@1;
! Define linear time slope factor by fixing factor loadings to time values
  FactLin BY outcome0@0 outcome1@1 outcome2@2 outcome3@3;
! Estimate factor means = fixed intercept and fixed linear time slope
  [FactInt FactLin] (FixInt FixLin);
! Estimate intercept factor variance = random intercept variance
  FactInt (IntVar);
! Estimate linear time slope factor variance = random slope variance
  FactLin (LinVar);
! Estimate random effects covariance
  FactInt WITH FactLin (ILcov);
! Occasion intercepts all fixed to 0
  [outcome0-outcome3@0];
! Occasion residual variances all constrained equal
  outcome0-outcome3 (ResVar);
! Occasion residual covariances all constrained equal within lags
  outcome0-outcome2 PWITH outcome1-outcome3 (ResCov1);
  outcome0-outcome1 PWITH outcome2-outcome3 (ResCov2);
  outcome0          PWITH outcome3          (ResCov3);

MODEL CONSTRAINT: ! Constraining residual covariances to AR1 pattern
NEW (AR1cor);     ! New parameter to be estimated
ResCov1 = AR1cor**1*SQRT(ResVar)*SQRT(ResVar); ! Lag 1
ResCov2 = AR1cor**2*SQRT(ResVar)*SQRT(ResVar); ! Lag 2
ResCov3 = AR1cor**3*SQRT(ResVar)*SQRT(ResVar); ! Lag 3
```

MPLUS MODEL RESULTS (SINCE R LAVAAN WOULD NOT WORK FOR ME)

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value	
FACTINT BY					
OUTCOME0	1.000	0.000	999.000	999.000	
OUTCOME1	1.000	0.000	999.000	999.000	
OUTCOME2	1.000	0.000	999.000	999.000	
OUTCOME3	1.000	0.000	999.000	999.000	
FACTLIN BY					
OUTCOME0	0.000	0.000	999.000	999.000	
OUTCOME1	1.000	0.000	999.000	999.000	
OUTCOME2	2.000	0.000	999.000	999.000	
OUTCOME3	3.000	0.000	999.000	999.000	
FACTINT WITH					
FACTLIN	0.101	0.496	0.204	0.838	→ **L2 Random Effects Covariance**
OUTCOME0 WITH					
OUTCOME1	0.053	0.514	0.102	0.918	
OUTCOME2	0.004	0.068	0.053	0.957	
OUTCOME3	0.000	0.007	0.036	0.971	
OUTCOME1 WITH					
OUTCOME2	0.053	0.514	0.102	0.918	
OUTCOME3	0.004	0.068	0.053	0.957	
OUTCOME2 WITH					
OUTCOME3	0.053	0.514	0.102	0.918	
Means					
FACTINT	10.279	0.326	31.564	0.000	→ **Fixed Intercept**
FACTLIN	1.717	0.200	8.570	0.000	→ **Fixed Linear Time Slope**
Intercepts					
OUTCOME0	0.000	0.000	999.000	999.000	
OUTCOME1	0.000	0.000	999.000	999.000	
OUTCOME2	0.000	0.000	999.000	999.000	
OUTCOME3	0.000	0.000	999.000	999.000	
Variances					
FACTINT	2.040	1.292	1.579	0.114	→ **L2 Random Intercept Variance**
FACTLIN	0.847	0.341	2.480	0.013	→ **L2 Random Linear Slope Variance**
Residual Variances					
OUTCOME0	0.758	0.606	1.251	0.211	→ **L1 Residual Variance**
OUTCOME1	0.758	0.606	1.251	0.211	
OUTCOME2	0.758	0.606	1.251	0.211	
OUTCOME3	0.758	0.606	1.251	0.211	
New/Additional Parameters					
AR1COR	0.069	0.624	0.111	0.911	→ **L1 AR1 correlation in R matrix**

7. Random Linear Time in G + Toeplitz Lag-1 Residual Covariance in R

```
!!!! Mplus Ch.5: Random Linear Time in G + Toeplitz Lag-1 Residual Covariance in R
! Define intercept factor by fixing all factor loadings to 1
  FactInt BY outcome0-outcome3@1;
! Define linear time slope factor by fixing factor loadings to time values
  FactLin BY outcome0@0 outcome1@1 outcome2@2 outcome3@3;
! Estimate factor means = fixed intercept and fixed linear time slope
  [FactInt FactLin] (FixInt FixLin);
! Estimate intercept factor variance = random intercept variance
  FactInt (IntVar);
! Estimate linear time slope factor variance = random slope variance
  FactLin (LinVar);
! Estimate random effects covariance
  FactInt WITH FactLin (ILcov);
! Occasion intercepts all fixed to 0
  [outcome0-outcome3@0];
! Occasion residual variances all constrained equal
  outcome0-outcome3 (ResVar);
! Occasion residual covariances for lag-1 only constrained equal
  outcome0-outcome2 PWITH outcome1-outcome3 (ResCov1);
```

```
print("R lavaan Ch 5: Random Linear Time in G + Toeplitz Lag-1 Residual Covariance in R")
SyntaxRandLinTP2 = "
# Define intercept factor by fixing all factor loadings to 1
  FactInt =~ 1*outcome.0 + 1*outcome.1 + 1*outcome.2 + 1*outcome.3
# Define linear time slope factor by fixing factor loadings to time value
  FactLin =~ 0*outcome.0 + 1*outcome.1 + 2*outcome.2 + 3*outcome.3
# Estimate factor means = fixed intercept and fixed linear time slope
  FactInt ~ (FixInt)*1; FactLin ~ (FixLin)*1
# Estimate intercept factor variance = random intercept variance
  FactInt ~~ (IntVar)*FactInt
# Estimate linear time slope factor variance = random slope variance
  FactLin ~~ (LinVar)*FactLin
# Estimate random effects covariance
  FactInt ~~ (ILcov)*FactLin
# Occasion intercepts all fixed to 0
  outcome.0 ~ 0; outcome.1 ~ 0; outcome.2  ~ 0; outcome.3 ~ 0
# Occasion residual variances all constrained equal
  outcome.0 ~~ (ResVar)*outcome.0; outcome.1 ~~ (ResVar)*outcome.1
  outcome.2 ~~ (ResVar)*outcome.2; outcome.3 ~~ (ResVar)*outcome.3
# Lag-1 covariances all constrained equal
  outcome.0 ~~ (Rcov1)*outcome.1; outcome.1 ~~ (Rcov1)*outcome.2; outcome.2 ~~ (Rcov1)*outcome.3
"
ModelRandLinTP2 = lavaan(model=SyntaxRandLinTP2, data=Example5_wide, estimator="ML", mimic="mplus")
summary(ModelRandLinTP2, fit.measures=TRUE, standardized=FALSE)
```

```
Latent Variables:  Estimate  Std.Err  z-value  P(>|z|)
  FactInt =~
    outcome.0         1.000
    outcome.1         1.000
    outcome.2         1.000
    outcome.3         1.000
  FactLin =~
    outcome.0         0.000
    outcome.1         1.000
    outcome.2         2.000
    outcome.3         3.000

Covariances:       Estimate  Std.Err  z-value  P(>|z|)
  FactInt ~~
    FactLin (ILcv)    0.087    0.423    0.204    0.838
 .outcome.0 ~~
   .outcm.1 (Rcv1)    0.032    0.328    0.096    0.923 → Lag-1 residual covariance
 .outcome.1 ~~
   .outcm.2 (Rcv1)    0.032    0.328    0.096    0.923
 .outcome.2 ~~
   .outcm.3 (Rcv1)    0.032    0.328    0.096    0.923
```

```
Intercepts:         Estimate  Std.Err  z-value  P(>|z|)
    FactInt (FxIn)   10.277    0.325   31.643    0.000
    FactLin (FxLn)    1.717    0.200    8.565    0.000
   .outcm.0           0.000
   .outcm.1           0.000
   .outcm.2           0.000
   .outcm.3           0.000

Variances:          Estimate  Std.Err  z-value  P(>|z|)
    FactInt (IntV)    2.082    1.041    2.000    0.046
    FactLin (LnVr)    0.854    0.314    2.717    0.007
   .outcm.0 (RsVr)    0.734    0.401    1.830    0.067
   .outcm.1 (RsVr)    0.734    0.401    1.830    0.067
   .outcm.2 (RsVr)    0.734    0.401    1.830    0.067
   .outcm.3 (RsVr)    0.734    0.401    1.830    0.067
```

```
print("Model-implied marginal means, variances, and covariances; add correlations")
fitted(object=ModelRandLinTP2); lavInspect(object=ModelRandLinTP2, "cor.ov")
```

```
$cov
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0  2.817
outcome.1  2.201  3.844
outcome.2  2.255  4.082  6.580
outcome.3  2.342  4.991  7.672 11.024

$mean
outcome.0 outcome.1 outcome.2 outcome.3
   10.277    11.994    13.711    15.427

Correlations
          otcm.0 otcm.1 otcm.2 otcm.3
outcome.0 1.000
outcome.1 0.029  1.000
outcome.2 0.026  0.053  1.000
outcome.3 0.019  0.052  0.079  1.000
```

```
print("Does random linear time slope improve fit?")
anova(ModelRandLinTP2, ModelRandLin) # Test TOEP lag-1 covariance
```

```
Chi-Squared Difference Test

                Df    AIC     BIC   Chisq Chisq diff Df diff Pr(>Chisq)
ModelRandLinTP2  7 378.98 387.512 13.9813
ModelRandLin     8 376.99 384.303 13.9907 0.00943379       1    0.92262
```

So how do we know that this model is “good enough” in terms of fit: (a) of the fixed linear time slope for predicting the means for each wave, and (b) of the level-2 random intercept, level-2 random linear time slope, the covariance of the level-2 random intercept and linear time slope, and level-1 residual for predicting the variances and covariances across waves? In single-level SEMs estimated using ML, this is possible to do easily via the usual indices of global model fit whenever you have balanced data (but not for unbalanced data, for which there is no single H1 saturated model).