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. **Bonus:** 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).

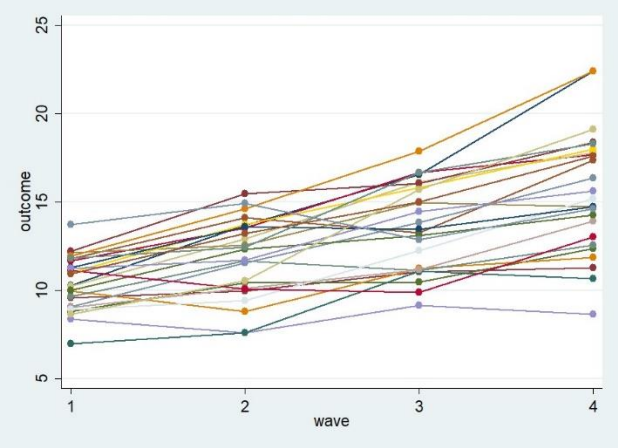
STATA Syntax for Data Import, Manipulation, and Description:

```
// Define working directory for file location
   cd "C:\Dropbox\24_PSQF6271\PSQF6271_Example5"

// Import Example 5 four-occasion long-format data from excel
   clear // clear memory in case a dataset is already open
   import excel "Example5_Data.xlsx", firstrow case(preserve) sheet("Example5") clear

// Center time at first occasion
   gen time = wave - 1
   label variable time "time: Time in Study (0=1)"

// To show what has been saved by default (but disappears unless named)
// return list
// ereturn list

display "STATA: Get Variance of Time Predictor for Slope Reliability (=1.263)"
summarize wave, detail
global TimeVar = r(Var) // Save variance of time predictor for slope reliability
display $TimeVar        // Show saved value to make sure it worked (= 1.263)
global Ntimes = 4       // Save number of occasions for use later
global LastTime = 3     // Save time value at last occasion for use later

display "STATA: Plot of Individual Trajectories"
xtset PersonID wave
xtline outcome, overlay legend(off) recast(connected)
graph export "STATA Individual Trajectories Plot.png", replace
```

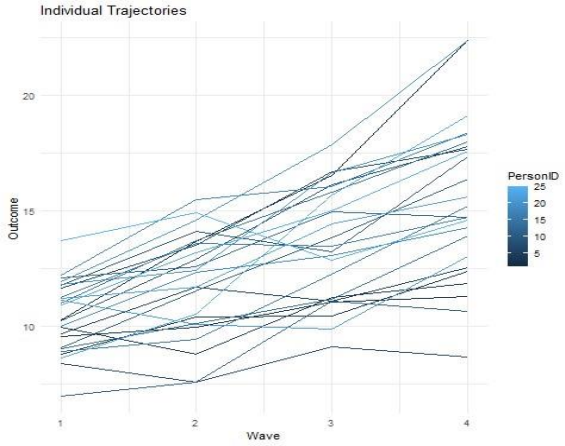
R Syntax for Data Import and Manipulation (after loading packages *readxl*, *TeachingDemos*, *psych*, *ggplot2*, *nlme*, *multcomp*, *emmeans*, *lmerTest*, *performance*, and *lavaan*):

```
# Set working directory (to import and export files to)
setwd("C:/Dropbox/24_PSQF6271/PSQF6271_Example5")

# Import Example 5 four-occasion long-format data from excel -- path = file name
Example5 = read_excel(path="Example5_Data.xlsx", sheet="Example5")
# Convert to data frame to use for analysis
Example5 = as.data.frame(Example5)
# Sort by person and occasion (needed for correct V matrix)
Example5 = Example5[order(Example5$PersonID, Example5$wave), ]

# Center time at first occasion
Example5$time=Example5$wave-1

print("R: Get Variance of Time Predictor for Slope Reliability (= 1.263)")
describe(x=Example5$time); TimeVar = var(Example5$time)
# Save number of occasions for use later
Ntimes = 4

png(file="R Individual Trajectories Plot.png")  # open file
ggplot(data=Example5, aes(x=wave, y=outcome, group=PersonID, color=PersonID)) +
       geom_line() + theme_minimal() + labs(title="Individual Trajectories", x="Wave", y="Outcome")
dev.off()  # close file
```

STATA Plot of Individual Trajectories:



R Plot of Individual Trajectories:



A disclaimer about the R code in this example (and in 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 for the fixed effects directly in the solution. In contrast, LMER does provide the requested Satterthwaite DDF, but it does not allow any R correlation structures (i.e., 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 most of the example models that contain random effects.

1. Syntax and R Output 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) for balanced time. The predicted outcome from the (saturated) fixed effects is given by: $\widehat{outcome}_{ti} = \beta_0 + \beta_1(wave2_{ti}) + \beta_2(wave3_{ti}) + \beta_3(wave4_{ti})$, in which the $wave_{ti}$ predictors are binary indicators for each wave (relative to a reference wave 1). The unstructured model for the variance cannot be easily summarized by scalar notation like this.

```
display "STATA Ch 5: Saturated Means, Unstructured Variance Model"
display "ANSWER KEY for both sides of the model"
mixed outcome i.wave, || PersonID: , noconstant reml nolog  ///
      residuals(unstructured,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
display "-2LL = " e(ll)*-2       // Print -2LL for model
estat wcorrelation, covariance   // R matrix
estat wcorrelation               // RCORR matrix
contrast i.wave, small           // Omnibus F-test for wave
margins  i.wave                  // Means per wave
marginsplot, xdimension(wave) name(predicted_wave, replace)
graph export "STATA Saturated Means Plot.png", replace
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(wave),
            correlation=corSymm(form=~as.numeric(wave)|PersonID), # Unstructured correlations
            weights=varIdent(form=~1|wave))                      # Heterogeneous variances

print("Show results with -2LL but incorrect residual DDF")
-2*logLik(SatUN); summary(SatUN)
```

```
'log Lik.' 353.7534 (df=14)  → -2LL for model
```

```
      AIC      BIC    logLik
  381.7534 417.6543 -176.8767

Correlation Structure: General
Parameter estimate(s):
 Correlation:
  1     2     3
2 0.820
3 0.511 0.733
4 0.460 0.733 0.913

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | wave
 Parameter estimates:
       1        2        3        4
1.000000 1.438901 1.622462 2.229871

Coefficients:
                Value Std.Error  t-value p-value
(Intercept)   10.4048 0.3073633 33.85180       0  beta0
factor(wave)2  1.4528 0.2591085  5.60692       0  beta1
factor(wave)3  3.1796 0.4320111  7.36000       0  beta2
factor(wave)4  5.1468 0.6087586  8.45458       0  beta3
```

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

Marginal variance covariance matrix → **R = V in R-only model**

```
       [,1]   [,2]   [,3]    [,4]
[1,] 2.3618 2.7867 1.9566  2.4204
[2,] 2.7867 4.8900 4.0440  5.5526
[3,] 1.9566 4.0440 6.2172  7.7994
[4,] 2.4204 5.5526 7.7994 11.7440
```

```
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.8199940 0.5105950 0.4595814
[2,] 0.8199940 1.0000000 0.7334295 0.7327197
[3,] 0.5105950 0.7334295 1.0000000 0.9127755
[4,] 0.4595814 0.7327197 0.9127755 1.0000000
```

→ **RCORR = VCORR in R-only model**

```
print("F-test p-value using Satterthwaite DDF")
joint_tests(object=ref_grid(SatUN), adjust="none", mode="satterthwaite")
```

```
model term df1   df2 F.ratio p.value
 wave         3 23.01  23.861  <.0001
```

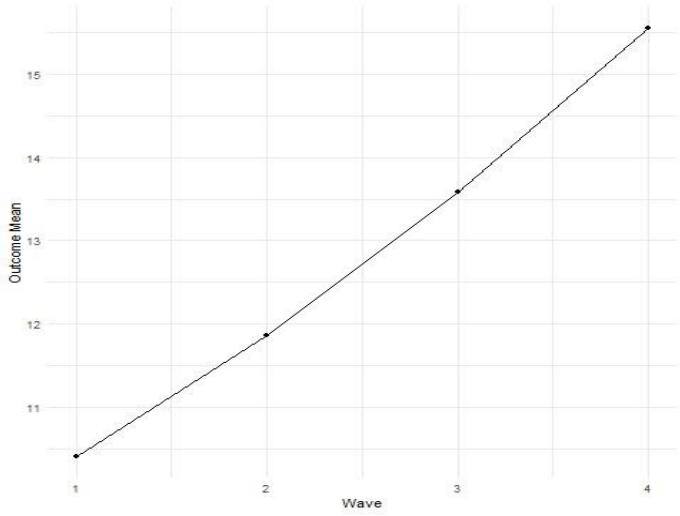
This is the multivariate ANOVA test of omnibus mean differences across wave (note numerator df=3 for the 4 means across waves).

```
print("Wave means and pairwise mean differences with Satterthwaite DDF")
emmeans(ref_grid(SatUN), pairwise~wave, adjust="none", mode="satterthwaite")
```

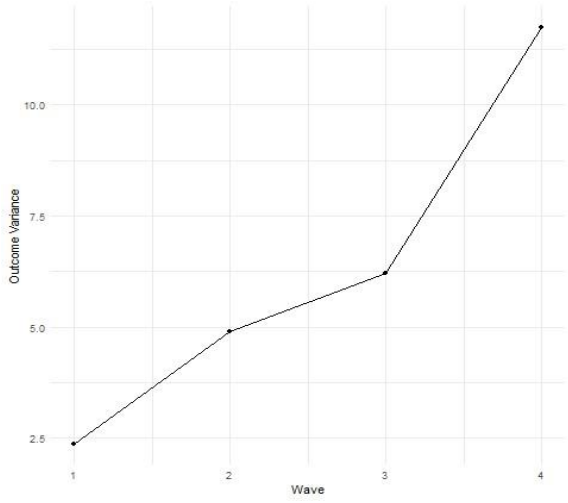
```
wave emmean    SE   df lower.CL upper.CL
    1   10.4 0.307 24.1     9.77     11.0  beta0
    2   11.9 0.442 24.1    10.95     12.8  beta0+beta1
    3   13.6 0.499 24.1    12.56     14.6  beta0+beta2
    4   15.6 0.685 23.9    14.14     17.0  beta0+beta3

Degrees-of-freedom method: appx-satterthwaite
Confidence level used: 0.95

contrast      estimate    SE   df t.ratio p.value
 wave1 - wave2    -1.45 0.259 23.7  -5.607  <.0001  (beta0)       - (beta0+beta1)
 wave1 - wave3    -3.18 0.432 24.1  -7.360  <.0001  (beta0)       - (beta0+beta2)
 wave1 - wave4    -5.15 0.609 23.9  -8.455  <.0001  (beta0)       - (beta0+beta2)
 wave2 - wave3    -1.73 0.348 24.0  -4.969  <.0001  (beta0+beta1) - (beta0+beta2)
 wave2 - wave4    -3.69 0.470 24.0  -7.855  <.0001  (beta0+beta1) - (beta0+beta3)
 wave3 - wave4    -1.97 0.307 23.4  -6.400  <.0001  (beta0+beta2) - (beta0+beta3)
```

```
png(file="R Saturated Means Plot.png")  # open file
SatMeans = emmeans(object=SatUN, specs="wave")  # Save saturated means to new object
SatMeans = as.data.frame(summary(SatMeans))     # Convert to data frame for plot
ggplot(data=SatMeans, aes(x=wave, y=emmean)) + geom_line() + geom_point() + theme_minimal() +
labs(x="Wave", y="Outcome Mean"); dev.off()  # close file

png(file="R Saturated Variances Plot.png")  # open file
SatVar = as.matrix(diag(SatUN_R)); SatVar = as.data.frame(SatVar) # Save sat means to new object
SatMeans = cbind(SatMeans, SatVar)                          # Concatenate to sat means to get wave
ggplot(data=SatMeans, aes(x=wave, y=V1)) + geom_line() + geom_point() + theme_minimal() +
labs(x="Wave", y="Outcome Variance"); dev.off()  # close file
```

Saturated Means by Wave (from R):



Saturated Variances by Wave (from R):



---

2. Syntax and Output 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).

$$\boldsymbol{outcome_{ti} = \beta_0 + \beta_1(wave2_{ti}) + \beta_2(wave3_{ti}) + \beta_3(wave4_{ti}) + U_{0i} + e_{ti}}$$

```
display "STATA Ch 5: Saturated Means, Random Intercept Model"
display "ANSWER KEY for means side only"
mixed outcome i.wave, || PersonID: , reml nolog  ///
      residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
```

```
                                                F(3,    72.00)    =     55.82
Log restricted-likelihood = -206.26874          Prob > F          =    0.0000
------------------------------------------------------------------------------
     outcome |      Coef.   Std. Err.           DF       t    P>|t|
-------------+----------------------------------------------------------------
        wave |
          2  |     1.4528   .4204656          72.0     3.46   0.001  beta1
          3  |     3.1796   .4204656          72.0     7.56   0.000  beta2
          4  |     5.1468   .4204656          72.0    12.24   0.000  beta3
       _cons |    10.4048   .5021216          42.4    20.72   0.000  beta0
------------------------------------------------------------------------------
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
PersonID: Identity           |
                  var(_cons) |   4.093261   1.344266      2.150438    7.791339  Var(U0) in G
-----------------------------+------------------------------------------------
               var(Residual) |   2.209892   .3683152      1.594058    3.063641  Var(e) in R
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 49.51         Prob >= chibar2 = 0.0000
```

```
display "-2LL = " e(ll)*-2      // Print -2LL for model
-2LL = 412.53749

estat wcorrelation, covariance  // V matrix

Covariances for PersonID = 1:
         obs |      1       2       3       4
-------------+--------------------------------
           1 |  6.303
           2 |  4.093   6.303
           3 |  4.093   4.093   6.303
           4 |  4.093   4.093   4.093   6.303

estat icc                       // Conditional intraclass correlation

Residual intraclass correlation
------------------------------------------------------------------------------
                       Level |        ICC   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
                    PersonID |    .649399   .0861367      .4687651    .7954181
------------------------------------------------------------------------------

contrast i.wave, small          // Omnibus F-test for wave

Contrasts of marginal linear predictions
Margins      : asbalanced
-------------------------------------------------------------
             |         df        ddf           F        P>F
-------------+-----------------------------------------------
outcome      |
        wave |          3      72.00       55.82     0.0000
-------------------------------------------------------------

margins  i.wave                 // Means per wave
------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        wave |
          1  |    10.4048   .5021216    20.72   0.000      9.42066    11.38894  beta0
          2  |    11.8576   .5021216    23.61   0.000     10.87346    12.84174  beta0+beta1
          3  |    13.5844   .5021216    27.05   0.000     12.60026    14.56854  beta0+beta2
          4  |    15.5516   .5021216    30.97   0.000     14.56746    16.53574  beta0+beta3
------------------------------------------------------------------------------

margins  i.wave, pwcompare(pveffects) df(72)  // Mean diffs by wave
-------------------------------------------------------
             |            Delta-method    Unadjusted
             |   Contrast   Std. Err.      t    P>|t|
-------------+-----------------------------------------
        wave |
     2 vs 1  |     1.4528   .4204656     3.46   0.001  (beta0+beta1) - (beta0)
     3 vs 1  |     3.1796   .4204656     7.56   0.000  (beta0+beta2) - (beta0)
     4 vs 1  |     5.1468   .4204656    12.24   0.000  (beta0+beta3) - (beta0)
     3 vs 2  |     1.7268   .4204656     4.11   0.000  (beta0+beta2) - (beta0+beta1)
     4 vs 2  |      3.694   .4204656     8.79   0.000  (beta0+beta3) - (beta0+beta1)
     4 vs 3  |     1.9672   .4204656     4.68   0.000  (beta0+beta3) - (beta0+beta2)
-------------------------------------------------------


print("R LME Ch 5: Saturated Means, Random Intercept Model")
print("ANSWER KEY for means side only")
SatRI = lme(data=Example5, method="REML", fixed=outcome~1+factor(wave), random=~1|PersonID)
print("Show results with -2LL but incorrect residual DDF")
-2*logLik(SatRI); summary(SatRI)
```

```
'log Lik.' 412.5375 (df=6) → -2LL for model
       AIC      BIC    logLik
  424.5375 439.9236 -206.2687

Random effects:
 Formula: ~1 | PersonID   Note that SDs are given instead of variances!
        (Intercept) Residual
StdDev:     2.02318 1.486571  → SD(U0) in G, SD(e) in R

Fixed effects:  outcome ~ 1 + factor(wave)
                Value Std.Error DF   t-value p-value
(Intercept)   10.4048 0.5021215 72 20.721677  0.0000  beta0
factor(wave)2  1.4528 0.4204657 72  3.455217  0.0009  beta1
factor(wave)3  3.1796 0.4204657 72  7.562092  0.0000  beta2
factor(wave)4  5.1468 0.4204657 72 12.240714  0.0000  beta3

print("Show V matrix for first person")
SatRI_V = getVarCov(SatRI, individual="1", type="marginal"); SatRI_V

Marginal variance covariance matrix
       1      2      3      4
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

print("Compute and show conditional ICC")
SatRI_ICC = (SatRI_V[[1]][2,1])/(SatRI_V[[1]][1,1]); SatRI_ICC
[1] 0.6493988

print("F-test p-value using Satterthwaite DDF")
joint_tests(object=ref_grid(SatRI), adjust="none", mode="satterthwaite")

model term df1 df2 F.ratio p.value
 wave         3  72  55.817  <.0001

print("Wave means and pairwise mean differences") → Omitted for brevity (interpreted as above)
emmeans(ref_grid(SatRI), pairwise~wave, adjust="none", mode="satterthwaite")
```

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")
print("ANSWER KEY for means side only")
SatRIr = lmer(data=Example5, REML=TRUE, formula=outcome~1+factor(wave)+(1|PersonID))
print("Show results with -2LL using Satterthwaite DDF")
llikAIC(SatRIr, chkREML=FALSE); summary(SatRIr, ddf="Satterthwaite")

$AICtab
      AIC       BIC    logLik  deviance  df.resid  → deviance = -2LL for model
 424.5375  440.1685 -206.2687  412.5375   94.0000

Random effects:
 Groups   Name        Variance Std.Dev.
 PersonID (Intercept) 4.093    2.023    for U0
 Residual             2.210    1.487    for e
Number of obs: 100, groups:  PersonID, 25

Fixed effects:
             Estimate Std. Error      df t value     Pr(>|t|)
(Intercept)   10.4048     0.5021 42.3812  20.722      < 2e-16  beta0
factor(wave)2  1.4528     0.4205 72.0000   3.455     0.000926  beta1
factor(wave)3  3.1796     0.4205 72.0000   7.562 0.0000000001  beta2
factor(wave)4  5.1468     0.4205 72.0000  12.241      < 2e-16  beta3
```

```
print("Show Conditional ICC"); icc(SatRIr)

# Intraclass Correlation Coefficient
    Adjusted ICC: 0.649
  Unadjusted ICC: 0.408

print("F-test p-value using Satterthwaite DDF"); anova(SatRIr, ddf="Satterthwaite")

Type III Analysis of Variance Table with Satterthwaite's method
             Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
factor(wave) 370.05  123.35     3    72  55.817 < 2.2e-16

print("Wave means and pairwise mean differences with Satterthwaite DDF") → Omitted for brevity
emmeans(ref_grid(SatRIr), pairwise~wave, adjust="none", lmer.df="satterthwaite")
```

I used `anova` instead of `joint_test` because the latter insists on Kenward-Roger DDF and I couldn't change it!

---

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

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

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

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

```
display "STATA Eq 5.1: Empty Means, Random Intercept Model"
mixed outcome , || PersonID: ,  reml nolog  ///
      residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
matrix EmptyRI = r(table)        // Save results for computations below

                                              F(0,     0.00)    =           .
Log restricted-likelihood = -251.11191        Prob > F          =           .
------------------------------------------------------------------------------
     outcome |      Coef.   Std. Err.           DF       t    P>|t|
-------------+----------------------------------------------------------------
       _cons |    12.8496   .4310792         24.0    29.81   0.000   gamma0
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
PersonID: Identity           |
                  var(_cons) |   2.881871    1.37169      1.133773    7.325256  Var(U0) in G
-----------------------------+------------------------------------------------
               var(Residual) |   7.055445   1.152149      5.122993    9.716839  Var(e) in R
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 9.79          Prob >= chibar2 = 0.0009
```

The **chibar2** test above is a likelihood ratio (LR) test comparing this model to a single-level regression (without a random intercept, as **linear model**) using a chi-square (χ^2) distribution with a mixture of DF=0 (for which $\chi^2 = 0$ always) and DF=1. Consequently, in this case you can obtain the mixture *p*-value by weighting each contribution to the χ^2 by 0.5, which means cutting the regular *p*-value in half. **Here, this LRT is a significance test of the intraclass correlation (ICC), which in turn provides an effect size for the amount of constant dependency attributed to person mean differences in the outcome.**

```
display "-2LL = " e(ll)*-2       // Print -2LL for model
-2LL = 502.22381

estat icc                        // Unconditional intraclass correlation
```

$$ICC = \frac{\tau^2_{U_0}}{\tau^2_{U_0} + \sigma^2_e} = \frac{2.882}{2.882 + 7.0558} = .290$$

```
Intraclass correlation
------------------------------------------------------------------------------
                       Level |        ICC   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
                    PersonID |    .290005   .1100872       .125289    .5380638
------------------------------------------------------------------------------
```

```
matrix list EmptyRI              // Show saved results

EmptyRI[9,3]
          outcome:  lns1_1_1:   lnsig_e:  // Variances are stored as log of SD instead
            _cons       _cons       _cons
     b    12.8496   .52921985  .97689982
    se   .43107921   .23798611   .08164965
     t   29.807979   2.2237426   11.964531
pvalue   1.788e-20   .02616577   5.450e-33
    ll   11.959896   .06277565   .81686944
    ul   13.739304   .99566405   1.1369302
    df         24           .           .
  crit   2.0638986    1.959964    1.959964
 eform          0           0           0

global EmptyResVar = exp(EmptyRI[1,3])^2 // Save as L1 residual variance
display $EmptyResVar                     // Show saved value to make sure it worked
7.0554448  → matches L1 residual variance in output

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")
llikAIC(EmptyRIr, chkREML=FALSE); summary(EmptyRIr, ddf="Satterthwaite")
print("Show Unconditional ICC"); icc(EmptyRIr)

print("Does random intercept improve model fit?")
ranova(EmptyRIr, reduce.term=TRUE) # LRT for removing random intercept

print("Print stored variance components table for reference")
as.data.frame(VarCorr(EmptyRIr))
# Save L1 residual variance
EmptyResVar = as.data.frame(VarCorr(EmptyRIr))[2,4]
```

---

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

Level 1: $outcome_{ti} = \beta_{0i} + \beta_{1i}(time_{ti}) + e_{ti}$

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

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

Composite: $outcome_{ti} = (\gamma_{00} + U_{0i}) + (\gamma_{10})(time_{ti}) + e_{ti}$

```
display "STATA Eq 5.3: Fixed Linear Time, Random Intercept Model"
mixed outcome c.time, || PersonID: , reml nolog  ///
      residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
estimates store FitFixLin            // Save for LRT
matrix FixLin = r(table)             // Save results for computations below
display "-2LL = " e(ll)*-2           // Print -2LL for model
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)$LastTime)) vsquish  // Intercept at each occasion (start(by)end)

matrix list FixLin  // Show saved results
global FixLinResVar = exp(FixLin[1,4])^2 // Save as L1 residual variance
display $FixLinResVar                    // Show saved value to make sure it worked
display "STATA Pseudo-R2 for L1 ResVar = " 1-($FixLinResVar/$EmptyResVar)
// Longer version of formula: R2 = (was-is)/was

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")
llikAIC(FixLinRIr, chkREML=FALSE); summary(FixLinRIr, ddf="Satterthwaite")
```

```
$AICtab
      AIC       BIC    logLik  deviance  df.resid
 423.0954  433.5161 -207.5477  415.0954    96.0000

Random effects:
 Groups    Name        Variance Std.Dev.
 PersonID (Intercept) 4.103    2.025    for U0
 Residual             2.173    1.474    for e
Number of obs: 100, groups:  PersonID, 25

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  10.2745     0.4743 34.6660   21.66   <2e-16  gamma00
time          1.7167     0.1318 74.0000   13.02   <2e-16  gamma10
```

Note that the random intercept variance actually increased from 2.882 to 4.103. This is because of how $\tau^2_{U_0}$ is found:

true $\tau^2_{U_0}$ = observed $\tau^2_{U_0}$ – (σ^2_e/ n)

So reducing σ^2_e will make $\tau^2_{U_0}$ (and the conditional ICC, as given in the off-diagonals of VCORR) increase!

```
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))

  Estimate Std. Error     df  t value     Pr(>|t|)
1 10.27452   0.474273 34.666 21.66373 9.854775e-22
1 11.99124  0.4360898 25.131 27.49718 2.813423e-20
1 13.70796  0.4360898 25.131 31.43380 1.068453e-21
1 15.42468   0.474273 34.666 32.52279 1.453392e-27
```

Separate results per **contest1D** compiled for convenience from:

$$\widehat{outcome}_{ti} = \gamma_{00} + \gamma_{10}(time_{ti})$$

```
print("Print stored variance components table for reference")
as.data.frame(VarCorr(EmptyRIr))

       grp        var1 var2     vcov    sdcor
1 PersonID (Intercept) <NA> 4.102598 2.025487
2 Residual        <NA> <NA> 2.172533 1.473952

# Save L1 residual variance
FixLinResVar = as.data.frame(VarCorr(FixLinRIr))[2,4]
print("Pseudo-R2 for L1 Residual Variance")
1-(FixLinResVar/EmptyResVar)
# Longer version of formula: R2 = (was-is)/was
[1] 0.6920771
```

After controlling for the fixed linear effect of time, the residual variance was reduced from σ^2_e = 7.06 in the empty means, random intercept model to σ^2_e = 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). An equivalent shorter version of the formula is used here as 1 – (is/was).

---

5. Syntax and Output for Equation 5.5: Random Linear Time Model

Level 1: $outcome_{ti} = \beta_{0i} + \beta_{1i}(time_{ti}) + e_{ti}$

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

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

Composite: $outcome_{ti} = (\gamma_{00} + U_{0i}) + (\gamma_{10} + U_{1i})(time_{ti}) + e_{ti}$

```
display "STATA Eq 5.5: Random Linear Time Model"
mixed outcome c.time, || PersonID: time, reml nolog covariance(unstructured)  ///
      residuals(independent,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
matrix RandLin = r(table)             // Save results for computations below

                                              F(1,    24.00)    =      70.26
Log restricted-likelihood = -183.37075        Prob > F          =     0.0000
------------------------------------------------------------------------------
     outcome |      Coef.   Std. Err.           DF       t    P>|t|
-------------+----------------------------------------------------------------
        time |    1.71672   .2048056         24.0     8.38    0.000  gamma10
       _cons |   10.27452   .3317511         24.0    30.97    0.000  gamma00
------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.    [95% Conf. Interval]  G matrix is now 2x2:
-----------------------------+------------------------------------------------
PersonID: Unstructured       |
                  var(time) |    .908907   .3040016    .4718659    1.750735  Var(U1) in G
                 var(_cons) |   2.262429   .8002807    1.131052    4.525508  Var(U0) in G
            cov(time,_cons) |   .0545357   .3506837   -.6327918    .7418632  Covar(U0,U1) in G
-----------------------------+------------------------------------------------
              var(Residual) |   .6986308   .1397261    .4720712    1.033923
------------------------------------------------------------------------------
LR test vs. linear model: chi2(3) = 99.47                 Prob > chi2 = 0.0000
```

The LRT above tests whether we need **anything** in the **G** matrix (which is why 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 below).

After adding a random linear time slope variance, 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 level-1 residual variance (in the **R** matrix) is now individual differences in the linear time slope as a new pile of variance in the **G** matrix (along with the covariance between the random intercepts and random linear time slopes in the **G** matrix).

```
display "-2LL = " e(ll)*-2                          // Print -2LL for model
-2LL = 366.7415

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

Random-effects correlation matrix for level PersonID
             |      time      _cons
-------------+----------------------
        time |         1
       _cons |  .0380306          1

estat wcorrelation, covariance                      // V matrix

Covariances for PersonID = 1:
         obs |      1       2       3       4
-------------+--------------------------------
           1 |  2.961
           2 |  2.317   3.979
           3 |  2.372   4.244   6.815
           4 |  2.426   5.207   7.989  11.468

estat wcorrelation                                  // VCORR matrix

Correlations:
         obs |      1       2       3       4
-------------+--------------------------------
           1 |  1.000
           2 |  0.675   1.000
           3 |  0.528   0.815   1.000
           4 |  0.416   0.771   0.904   1.000

margins, at(c.time=(0(1)$LastTime)) vsquish  // Intercept at each occasion (start(by)end)

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]  g = gamma
-------------+----------------------------------------------------------------
         _at |
          1  |   10.27452   .3317511    30.97   0.000       9.6243    10.92474  g00 + g10(0)
          2  |   11.99124   .3736306    32.09   0.000     11.25894    12.72354  g00 + g10(1)
          3  |   13.70796   .5030224    27.25   0.000     12.72205    14.69387  g00 + g10(2)
          4  |   15.42468   .6710841    22.98   0.000     14.10938    16.73998  g00 + g10(3)
------------------------------------------------------------------------------
```

The **GCORR** matrix provides the correlation(s) among the individual random effects. Here, the individual intercepts and linear time slopes are correlated *r* = .038 (from covariance = .054 above).

```
estimates store FitRandLin            // Save for LRT
lrtest FitRandLin FitFixLin           // Does random linear time slope improve fit?

Likelihood-ratio test                                 LR chi2(2)  =     48.35
(Assumption: FitFixLin nested in FitRandLin)          Prob > chi2 =    0.0000

matrix list RandLin                   // Show saved results

RandLin[9,6]
            outcome:     outcome:    lns1_1_1:    lns1_1_2:  atr1_1_1_2:     lnsig_e:
               time        _cons        _cons        _cons        _cons        _cons
     b      1.71672     10.27452    -.04775627    .40821943    .03804899    -.17931642
    se     .20480558    .33175113     .1672347    .17686318    .24690656     .09999997
     t     8.3821936    30.970566    -.28556437    2.3081086    .15410279    -1.7931648
pvalue     1.368e-08    7.298e-21     .77521179     .0209931    .87752869     .07294658
    ll     1.2940221    9.5898193    -.37553026    .06157397   -.44587897    -.37531276
    ul     2.1394179    10.959221     .28001771     .7548649    .52197695     .01667992
    df            24           24             .             .             .             .
  crit     2.0638986    2.0638986     1.959964     1.959964     1.959964     1.959964
 eform             0            0            0            0            0            0

global RandLinFixLin = RandLin[1,1]   // Save fixed linear slope
global RandLinFixInt = RandLin[1,2]   // Save fixed intercept
global RandLinLinVar = exp(RandLin[1,3])^2  // Save as L2 random linear slope variance
global RandLinIntVar = exp(RandLin[1,4])^2  // Save as L2 random intercept variance
global RandLinResVar = exp(RandLin[1,6])^2  // Save as L1 residual variance
```

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:

Random Effect 95% CI = fixed effect + (1.96*$\sqrt{\text{random variance}}$)

Intercept 95% CI $= \gamma_{00} \pm \left(1.96\sqrt{\tau^2_{U_0}}\right) \rightarrow 10.27 \pm \left(1.96\sqrt{2.26}\right) = 7.33$ to 13.22

Linear Time Slope 95% CI $= \gamma_{10} \pm \left(1.96\sqrt{\tau^2_{U_1}}\right) \rightarrow \quad 1.72 \pm \left(1.96\sqrt{0.91}\right) = -0.15$ 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 at the beginning):

$$\text{IR} = \frac{\tau^2_{U_0}}{\tau^2_{U_0} + \frac{\sigma^2_e}{L1n}} = \frac{2.26}{2.26 + \frac{.70}{4}} = .93 \qquad \text{SR} = \frac{\tau^2_{U_1}}{\tau^2_{U_1} + \frac{\sigma^2_e}{L1n * \sigma^2_{L1}}} = \frac{0.91}{0.91 + \frac{.70}{4 * 1.26}} = .87$$

```
display "STATA 95% Random Intercept CI"
display "Lower = " $RandLinFixInt - 1.96*sqrt($RandLinIntVar)
display "Upper = " $RandLinFixInt + 1.96*sqrt($RandLinIntVar)
Lower = 7.3264111
Upper = 13.222629

display "STATA 95% Random Linear Time Slope CI"
display "Lower = " $RandLinFixLin - 1.96*sqrt($RandLinLinVar)
display "Upper = " $RandLinFixLin + 1.96*sqrt($RandLinLinVar)
Lower = -.1518776
Upper = 3.5853176

display "STATA Intercept Reliability = ICC2"  // See Lecture 5 slide 38
display $RandLinIntVar/($RandLinIntVar+($RandLinResVar/$Ntimes))
.9283334

display "STATA Slope Reliability"  // See Lecture 5 slide 38
display $RandLinLinVar/($RandLinLinVar+($RandLinResVar/($Ntimes*$TimeVar)))
.86791046
```

```
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")
llikAIC(RandLinr, chkREML=FALSE); summary(RandLinr, ddf="Satterthwaite")
```

```
$AICtab
      AIC       BIC    logLik  deviance  df.resid
 378.7415  394.3725 -183.3708  366.7415   94.0000

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 PersonID (Intercept) 2.2624   1.5041         for U0 in G
          time        0.9089   0.9534   0.04  for U1 in G
 Residual             0.6986   0.8358         for e  in R
```

`lmer` provides the random effect correlation instead of the random effect covariance, but you can compute the covariance as:

$cov = r * SD(intercept) * SD(slope)$

$.054 = .038 * \sqrt{2.2624} * SD\sqrt{0.9089}$

```
Fixed effects:
            Estimate Std. Error      df t value     Pr(>|t|)
(Intercept)  10.2745     0.3317 24.0002  30.971      < 2e-16  gamma00
time          1.7167     0.2048 24.0001   8.382 0.0000000137  gamma10
```

```
# R code to generate V matrix after lmer (not built into lme4)
# Adapted from: https://stackoverflow.com/questions/45650548/get-residual-variance-covariance-matrix-in-lme4
# Parameterization explained here:
# https://stats.stackexchange.com/questions/156438/what-is-the-variance-component-parameter-in-mixed-effect-model

# Lambdat = Transpose of "relative covariance factor" of G matrix (element = SD random/SD res)
RandLinGtm = crossprod(getME(RandLinr,"Lambdat")) → will become correct G matrix below
# Transpose of Z for predictors with random effects
RandLinZtm = getME(RandLinr,"Zt")
# Residual variance as scalar
RandLinRes = sigma(RandLinr)^2
# Build parts of V = ZGZ' + R --> V = L2part + L1part
L2part = RandLinRm*(t(RandLinZtm) %*% RandLinGtm %*% RandLinZtm)
L1part = RandLinRes*Diagonal(nrow(Example5))
RandLinVm = L2part + L1part

print("Show first block diagonal of V and VCORR (1 to # occasions)")
RandLinVm[1:Ntimes,1:Ntimes]; cov2cor(RandLinVm)[1:Ntimes,1:Ntimes]
```

```
         1        2        3         4
1 2.961042 2.316945 2.371483  2.426021
2 2.316945 3.979018 4.243823  5.207261
3 2.371483 4.243823 6.814797  7.988502
4 2.426021 5.207261 7.988502 11.468378

          1         2         3         4
1 1.0000000 0.6750026 0.5279238 0.4163142
2 0.6750026 1.0000000 0.8149722 0.7708509
3 0.5279238 0.8149722 1.0000000 0.9036243
4 0.4163142 0.7708509 0.9036243 1.0000000
```

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 matrix elements are now predicted to differ as a function of which wave it is (see Table 5.2 in Hoffman 2015 chapter 5 for a description of how this works).

```
print("Predicted outcome means using predict on fake cases")
times = data.frame(time=seq(from=0, to=Ntimes-1, by=1))
RandLinPred = predict(object=RandLinr, newdata=times, se.fit=TRUE, re.form=NA)
cbind(times, as.data.frame(RandLinPred)) # Combine occasions with predictions
```

```
  time      fit    se.fit  g = gamma
1    0 10.27452 0.3317500  g00 + g10(0)
2    1 11.99124 0.3736294  g00 + g10(1)
3    2 13.70796 0.5030210  g00 + g10(2)
4    3 15.42468 0.6710823  g00 + g10(3)
```

```
print("Does random linear time slope improve fit?")
ranova(RandLinr, reduce.term=TRUE) # Remove random slope and covariance

                              npar  logLik    AIC    LRT Df       Pr(>Chisq)
<none>                           6 -183.37 378.74
time in (1 + time | PersonID)    4 -207.55 423.10 48.354  2 0.00000000003163

# Print each object first to see which row and column values to extract
as.data.frame(fixef(RandLinr)); as.data.frame(VarCorr(RandLinr))

            fixef(RandLinr)
(Intercept)        10.27452
time                1.71672

       grp        var1 var2       vcov      sdcor
1 PersonID (Intercept) <NA> 2.26240692 1.50412996
2 PersonID        time <NA> 0.90890103 0.95336301
3 PersonID (Intercept) time 0.05453787 0.03803247
4 Residual        <NA> <NA> 0.69863477 0.83584375

# Save fixed effects and variances for computations below
RandLinFixInt = as.data.frame(fixef(RandLinr))[1,1]
RandLinFixLin = as.data.frame(fixef(RandLinr))[2,1]
RandLinIntVar = as.data.frame(VarCorr(RandLinr))[1,4]
RandLinLinVar = as.data.frame(VarCorr(RandLinr))[2,4]
RandLinResVar = as.data.frame(VarCorr(RandLinr))[4,4]

print("R 95% Random Intercept Confidence Interval")
print("Lower = "); RandLinFixInt - 1.96*sqrt(RandLinIntVar)
print("Upper = "); RandLinFixInt + 1.96*sqrt(RandLinIntVar)
[1] 7.326425
[1] 13.22261

print("R 95% Random Linear Time Slope Confidence Interval")
print("Lower = "); RandLinFixLin - 1.96*sqrt(RandLinLinVar)
print("Upper = "); RandLinFixLin + 1.96*sqrt(RandLinLinVar)
[1] -0.1518715
[1] 3.585311

print("R Intercept Reliability = ICC2")
RandLinIntVar/(RandLinIntVar+(RandLinResVar/Ntimes))
[1] 0.9283324

print("R Slope Reliability")
RandLinLinVar/(RandLinLinVar+(RandLinResVar/(Ntimes*TimeVar)))
[1] 0.8679091
```

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

6. Syntax and R Output for Random Linear Time in G + Auto-Regressive Residual Correlation in R

```
display "STATA Ch 5: Random Linear Time Model with AR1 R Matrix"
mixed outcome c.time,  || PersonID: time, reml nolog covariance(unstructured)  ///
      residuals(ar1,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
display "-2LL = " e(ll)*-2                          // Print -2LL for model
estat recovariance, relevel(PersonID) correlation  // GCORR matrix
estat wcorrelation, covariance                      // V matrix
estat wcorrelation                                  // VCORR matrix
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", fixed=outcome~1+time, random=~1+time|PersonID,
                 correlation=(corAR1(form=~as.numeric(time)|PersonID)))
print("Show results using incorrect DDF"); summary(RandLinAR1)

       AIC      BIC    logLik
  380.7394 398.8342 -183.3697

Random effects:
 Formula: ~1 + time | PersonID  Note that SDs are given instead of variances!
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.4905003 (Intr)  for U0 in G
time        0.9494917 0.049   for U1 in G
Residual    0.8481307         for e in R

Correlation Structure: AR(1)
 Formula: ~as.numeric(time) | PersonID
 Parameter estimate(s):
       Phi
0.02560753  → AR1 correlation in R
Fixed effects:  outcome ~ 1 + time
                Value Std.Error DF   t-value p-value
(Intercept) 10.276252 0.3308219 74 31.062795       0  gamma00
time         1.716677 0.2046617 74  8.387877       0  gamma10

print("Show R and V matrices for first person")
getVarCov(RandLinAR1, individual="1", type="conditional") # R matrix

Conditional variance covariance matrix
            1           2           3           4
1 0.719330000 0.01842000 0.00047169 0.000012079
2 0.018420000 0.71933000 0.01842000 0.000471690
3 0.000471690 0.01842000 0.71933000 0.018420000
4 0.000012079 0.00047169 0.01842000 0.719330000

getVarCov(RandLinAR1, individual="1", type="marginal")    # V matrix

Marginal variance covariance matrix
       1      2      3       4
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 11.4720

# Run random linear time model in lme instead for LRT
RandLin = lme(data=Example5, method="REML", outcome~1+time, random=~1+time|PersonID)
print("Does AR1 residual correlation improve fit?")
anova(RandLinAR1, RandLin) # anova compares using LME versions

           Model df      AIC      BIC    logLik   Test     L.Ratio p-value
RandLinAR1     1  7 380.7394 398.8342 -183.3697
RandLin        2  6 378.7415 394.2513 -183.3707 1 vs 2 0.002107161  0.9634 → AR1 corr doesn't help
```

The AR1 correlation shows up in the **R** matrix as a covariance instead, computed as $cov = r^x \sigma_e^2$ for lag x

---

The AR1 correlation didn't help model fit, but what about a lag-1 Toeplitz residual covariance?

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

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

display "STATA Ch 5: Random Linear Time Model with TOEP2 R Matrix"
mixed outcome c.time, || PersonID: time, reml nolog covariance(unstructured)  ///
             residuals(toeplitz1,t(wave)) dfmethod(satterthwaite) dftable(pvalue)
```

```
                                                F(1,    24.00)    =      70.33
Log restricted-likelihood = -183.37001          Prob > F          =     0.0000
------------------------------------------------------------------------------
     outcome |      Coef.   Std. Err.           DF       t    P>|t|
-------------+----------------------------------------------------------------
        time |    1.71669   .2047041          24.0     8.39   0.000  gamma10
       _cons |    10.2757   .3311023          24.0    31.03   0.000  gamma00
------------------------------------------------------------------------------
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
PersonID: Unstructured       |
                  var(time) |   .9038457   .3309504      .4409811    1.852544  Var(U1) in G
                 var(_cons) |   2.234488   1.078639      .8675279    5.755362  Var(U0) in G
            cov(time,_cons) |   .0648451   .4406703     -.7988528    .9285431  Covar(U0,U1) in G
-----------------------------+------------------------------------------------
Residual: Toeplitz(1)        |
                       cov1 |   .0124363   .3232059     -.6210356    .6459082  Lag-1 cov in R
                    var(e) |   .7125808   .3908035      .2432239     2.08767  Var(2) in R
------------------------------------------------------------------------------
LR test vs. linear model: chi2(4) = 99.48                 Prob > chi2 = 0.0000
```

```
display "-2LL = " e(ll)*-2                          // Print -2LL for model
-2LL = 366.74001

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

Random-effects correlation matrix for level PersonID
             |      time      _cons
-------------+----------------------
        time |         1
       _cons |   .045629          1


estat wcorrelation, covariance                      // V matrix

Covariances for PersonID = 1:
        wave |      1       2       3       4
-------------+--------------------------------
           1 |  2.947   2.312   2.364   2.429
           2 |  2.312   3.981   4.249   5.205
           3 |  2.364   4.249   6.822   7.994
           4 |  2.429   5.205   7.994  11.471

estat wcorrelation                                  // VCORR matrix

Correlations:
        wave |      1       2       3       4
-------------+--------------------------------
           1 |  1.000
           2 |  0.675   1.000
           3 |  0.527   0.815   1.000
           4 |  0.418   0.770   0.904   1.000

estimates store FitRandLinTOEP2   // Save for LRT
lrtest FitRandLinTOEP2 FitRandLin // Does lag-1 residual correlation improve fit?
Likelihood-ratio test                                 LR chi2(1)  =      0.00
(Assumption: FitRandLin nested in FitRandLinTO~2)     Prob > chi2 =    0.9692
```

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! For a sample results section, please see the end of Hoffman (2015) chapter 5.

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).

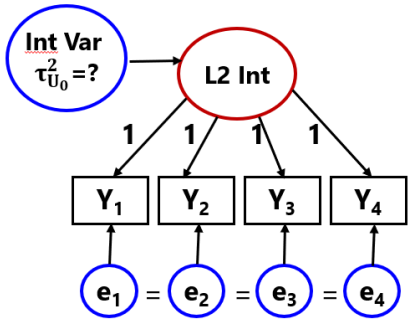
$$\boldsymbol{outcome_{ti} = \beta_0 + \beta_1(wave1_{ti}) + \beta_2(wave2_{ti}) + \beta_3(wave3_{ti}) + U_{0i} + e_{ti}}$$

```
!!!! 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

$$\textbf{L1:} \quad outcome_{ti} = \beta_{0i} + e_{ti}$$
$$\textbf{L2:} \quad \beta_{0i} = \gamma_{00} + U_{0i}$$
$$\textbf{Comp:} \quad outcome_{ti} = (\gamma_{00} + U_{0i}) + e_{ti}$$




```
!!!! 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
```

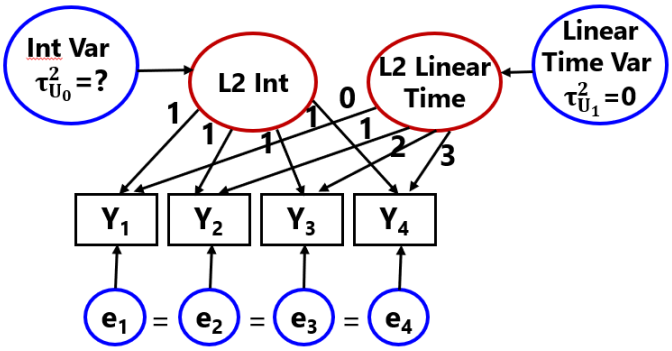
VCORR provides the ICC as: IntVar/TotalVar
Notice how much lower this unconditional ICC is relative to the conditional ICC from saturated means (i.e., this random intercept variance is much smaller before the residual variance due to mean differences over time has been explained).

```
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
```

---

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

L1: $outcome_{ti} = \beta_{0i}\beta_{1i}(time_{ti}) + e_{ti}$
L2: $\beta_{0i} = \gamma_{00} + U_{0i}$
$\beta_{1i} = \gamma_{10}$
Comp: $outcome_{ti} = (\gamma_{00} + U_{0i}) + (\gamma_{10})(time_{ti}) + e_{ti}$




```
!!!! 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
```

---

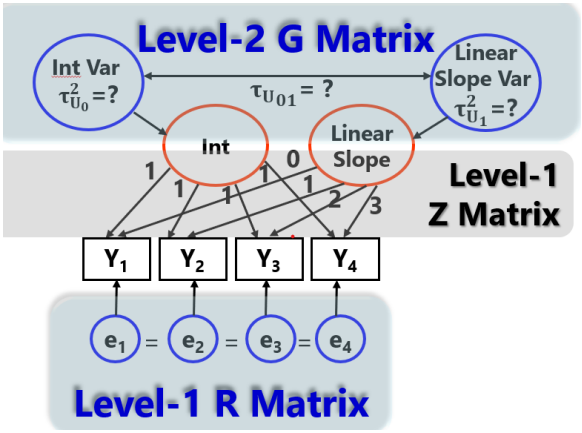
5. Equation 5.5: Random Linear Time Model

L1: $outcome_{ti} = \beta_{0i}\beta_{1i}(time_{ti}) + e_{ti}$

L2: $\beta_{0i} = \gamma_{00} + U_{0i}$

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

Comp: $outcome_{ti} = (\gamma_{00} + U_{0i}) + (\gamma_{10} + U_{1i})(time_{ti}) + e_{ti}$




```
!!!! Mplus Eq 5.5: Random Linear Time Model
! Define intercept factor by fixing 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
```

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).

```
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

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)

                                                    Two-Tailed
                  Estimate       S.E.  Est./S.E.    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).