Example 1: Time-Invariant Predictors in Polynomial Models for Change over Time

complete data, syntax, and output available electronically for R (1a) and Mplus (1b)

These simulated data are from Hoffman (2015) chapter 7b (continued in chapter 9) and include 200 girls measured approximately annually from ages 12–18 on their risky behavior (the outcome, an item sum ranging from 10 to 50) and the extent to which their mothers monitored their activities (the time-varying predictor, an item mean ranging from 1 to 5, centered at 3). A time-invariant predictor of the conservativeness of mothers' attitudes about the smoking and drinking (a mean ranging from 1 to 5, centered at 4) was also collected at the age 12 occasion. In this example we will predict risky behavior from exact age (centered such that time 0 = age 18) and time-invariant mothers' attitudes.

The following R packages were used in this example: readxl, expss, nlme, emmeans, lme4, lmerTest, performance, lavaan, and TeachingDemos (the latter output results to a text file). To facilitate comparison across MLM and SEM variants, all models were estimated using maximum likelihood (ML), rather than (the better) residual ML (REML).

Data Manipulation (used in Example 1a):

```
# Define variables for working directory and data name -- CHANGE THESE
filesave = "C:\\Dropbox/SMiP/SMiP_Workshop/Download/Example1/"
filename = "Example1.xlsx"
setwd(dir=filesave)

# Import Example1 data from excel in sheet "Data"
Example1_wide = read_excel(paste0(filesave,filename), sheet="Data")
# Convert to data frame to use in analysis
Example1_wide = as.data.frame(Example1_wide)
# Sort data by PersonID
Example1_wide = sort_asc(Example1_wide,PersonID)

# Stack into long format (one row per occasion per person) by variable
Example1 = reshape(Example1_wide, direction="long", idvar="PersonID",
                   varying=list(age=c(3,4,5,6,7,8,9), # column numbers
                                risky=c(10,11,12,13,14,15,16),
                                monitor=c(17,18,19,20,21,22,23)),
                   v.names=c("age","risky","mon"),
                   timevar="occasion", times=c(1,2,3,4,5,6,7))
# Sort data by PersonID (will be needed for correct RCOV matrix)
Example1 = sort_asc(Example1,PersonID)

# Create predictors for analysis in stacked data
Example1$roundage=Example1$occasion+11
Example1$time=Example1$age-18
Example1$att4=Example1$Attitude12-4
```

Data Description via a Model with Saturated Means *by Rounded Occasion*, Unstructured Variances

This model had to be estimated using gls (from the nlme package) in order to allow an unstructured R matrix (with all possible variances and covariances estimated separately) but no random effects over level-2 persons.

Model 0: Model for the Means (in which each predictor is a dummy-coded indicator for that age):

$$\hat{\text{risky}}_{ti} = \gamma_{00} + \gamma_{10}\left(\text{age13}_{ti}\right) + \gamma_{20}\left(\text{age14}_{ti}\right) + \gamma_{30}\left(\text{age15}_{ti}\right) + \gamma_{40}\left(\text{age16}_{ti}\right) + \gamma_{50}\left(\text{age17}_{ti}\right) + \gamma_{60}\left(\text{age18}_{ti}\right)$$

```
print("Model 0: Saturated Means, Unstructured R Matrix -- TOTAL ANSWER KEY")
SatUN = gls(data=Example1, method="ML", model=risky~1+factor(roundage),
            correlation=corSymm(form=~occasion|PersonID), # all possible correlations
            weights=varIdent(form=~1|occasion))           # all separate variances
print("Show results using incorrect DDF, with total leftover variance")
summary(SatUN)
```

```
Generalized least squares fit by maximum likelihood
  Model: risky ~ 1 + factor(roundage)
  Data: Example1
        AIC       BIC     logLik
  7676.8312 7860.3792 -3803.4156

Correlation Structure: General → RCORR matrix (lower off-diagonals)
 Formula: ~occasion | PersonID
 Parameter estimate(s):
 Correlation:
  1     2     3     4     5     6
2 0.574
3 0.520 0.559
4 0.424 0.466 0.613
5 0.430 0.560 0.615 0.656
6 0.293 0.457 0.503 0.634 0.657
7 0.237 0.417 0.379 0.583 0.603 0.678

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | occasion
 Parameter estimates:
         1          2          3          4          5          6          7
1.00000000 0.96755050 0.98442775 0.99804210 1.02413906 1.13917769 1.18320440

Coefficients:
                        Value  Std.Error   t-value p-value  Fixed effects:
(Intercept)        16.7223330 0.32401124 51.610348  0.0000  gamma00
factor(roundage)13  0.4604405 0.29436297  1.564193  0.1180  gamma10
factor(roundage)14  1.1402300 0.31493893  3.620480  0.0003  gamma20
factor(roundage)15  2.2595160 0.34754921  6.501284  0.0000  gamma30
factor(roundage)16  3.0505475 0.35029644  8.708474  0.0000  gamma40
factor(roundage)17  4.9283580 0.41358068 11.916316  0.0000  gamma50
factor(roundage)18  6.7987695 0.43948355 15.469907  0.0000  gamma60

Residual standard error: 4.5707411 → SQRT of risky outcome variance at age 12
Degrees of freedom: 1400 total; 1393 residual → incorrect residual denominator DF

print("Total variance per occasion is created using SD multiplier")
summary(SatUN)$sigma^2

[1] 20.891674 → Variance at age 12

print("Show R and RCORR matrices for first person")
R=getVarCov(SatUN, individual="1", type="marginal"); R

Marginal variance covariance matrix
→ These the variances and covariances the random effects will try to recreate
        [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]
[1,] 20.8920 11.6030 10.7000  8.8322  9.1927  6.9823  5.8517
[2,] 11.6030 19.5580 11.1280  9.3911 11.5880 10.5350  9.9710
[3,] 10.7000 11.1280 20.2460 12.5830 12.9560 11.7780  9.2227
[4,]  8.8322  9.3911 12.5830 20.8100 14.0110 15.0480 14.3850
[5,]  9.1927 11.5880 12.9560 14.0110 21.9120 16.0060 15.2540
[6,]  6.9823 10.5350 11.7780 15.0480 16.0060 27.1120 19.0860
[7,]  5.8517  9.9710  9.2227 14.3850 15.2540 19.0860 29.2480
  Standard Deviations: 4.5707 4.4224 4.4996 4.5618 4.6811 5.2069 5.4081
```

RCORR=corMatrix(SatUN$modelStruct$corStruct)[[4]]; RCORR (Correlation version)

```
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]       [,7]
[1,] 1.00000000 0.57402106 0.52025861 0.42358914 0.42964430 0.29338207 0.23672836
[2,] 0.57402106 1.00000000 0.55923486 0.46550160 0.55975847 0.45749615 0.41689930
[3,] 0.52025861 0.55923486 1.00000000 0.61301137 0.61512160 0.50269910 0.37900105
[4,] 0.42358914 0.46550160 0.61301137 1.00000000 0.65614874 0.63353335 0.58306747
[5,] 0.42964430 0.55975847 0.61512160 0.65614874 1.00000000 0.65670677 0.60253119
[6,] 0.29338207 0.45749615 0.50269910 0.63353335 0.65670677 1.00000000 0.67779750
[7,] 0.23672836 0.41689930 0.37900105 0.58306747 0.60253119 0.67779750 1.00000000
```

```
print("session means and pairwise mean differences with incorrect DDF")
emmeans(ref_grid(SatUN), pairwise~roundage, adjust="none") # tried mode="df.error"

print("Error when trying to get Satterthwaite DDF, so had to switch to residual DDF")
lsmeans(SatUN, "roundage", mode="df.error")
```

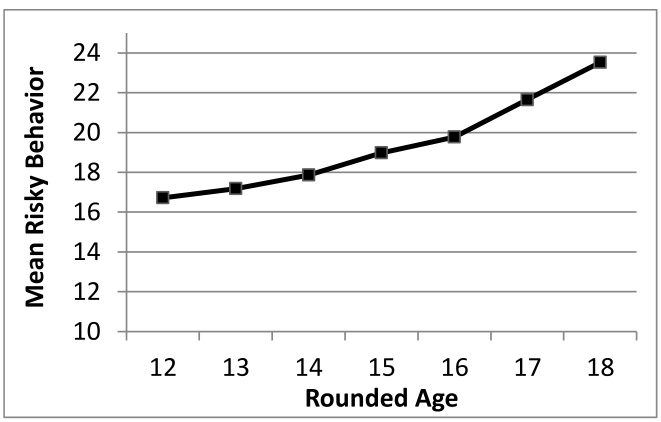
These are the means the fixed effects of time will be trying to recreate

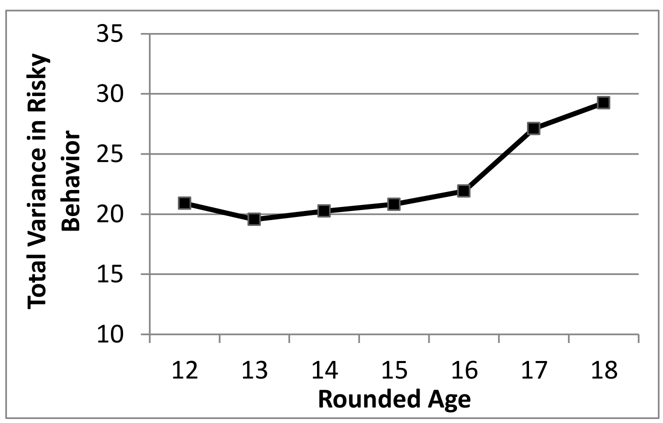
roundage	lsmean	SE	df	lower.CL	upper.CL	**Fixed effects:**
12	16.7	0.324	1365	16.1	17.4	**gamma00**
13	17.2	0.313	1365	16.6	17.8	**gamma00+gamma10**
14	17.9	0.319	1365	17.2	18.5	**gamma00+gamma20**
15	19.0	0.323	1365	18.3	19.6	**gamma00+gamma30**
16	19.8	0.332	1365	19.1	20.4	**gamma00+gamma40**
17	21.7	0.369	1365	20.9	22.4	**gamma00+gamma50**
18	23.5	0.383	1365	22.8	24.3	**gamma00+gamma60**

```
print("F-test p-value using Satterthwaite DDF even though it doesn't say so");
anova(SatUN)
```

```
Denom. DF: 1393
                 numDF   F-value p-value
(Intercept)          1 5474.6949  <.0001
factor(roundage)     6   55.1939  <.0001 → significant mean differences over time
```

So here is what are we trying to model—the black lines are means and variances from model 0, the data:





Model 1. Most Conservative Baseline—Empty Means, Random Intercept

Level 1: $\text{risky}_{ti} = \beta_{0i} + e_{ti}$

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

I am switching to lmer (from the lme4 package) for all models with a diagonal (independent) R matrix, that way the fixed effects will be tested using Satterthwaite denominator degrees of freedom (instead of residual).

```
print("Model 1: Empty Means, Random Intercept Model")
EmptyRI = lmer(data=Example1, REML=FALSE, formula=risky~1+(1|PersonID))
print("Show results using Satterthwaite DDF"); summary(EmptyRI, ddf="Satterthwaite")

Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method
['lmerModLmerTest']
Formula: risky ~ 1 + (1 | PersonID)
   Data: Example1

     AIC      BIC   logLik deviance df.resid
  8302.6   8318.3  -4148.3   8296.6     1397 → deviance = -2LL

Random effects:
 Groups   Name        Variance Std.Dev.
 PersonID (Intercept) 10.843   3.2929   → L2 BP random intercept variance
 Residual             17.241   4.1523   → L1 WP residual variance
Number of obs: 1400, groups:  PersonID, 200

Fixed effects:
             Estimate Std. Error        df t value  Pr(>|t|)  Fixed effects:
(Intercept)  19.38488    0.25794 200.00000  75.154 < 2.2e-16  gamma00


print("Get ICC"); icc(EmptyRI)

# Intraclass Correlation Coefficient
     Adjusted ICC: 0.386
  Conditional ICC: 0.386
```

VCORR: $\text{ICC} = \frac{10.843}{10.843 + 17.241} = .386$

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

ANOVA-like table for random-effects: Single term deletions
               npar  logLik     AIC     LRT Df Pr(>Chisq)
<none>            3 -4148.31 8302.62
(1 | PersonID)    2 -4321.16 8646.32 345.707  1 < 2.22e-16 → ICC is significant
```

V matrix (marginal predicted variance–covariance matrix) from SAS MIXED output:

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	**28.0843**	10.8431	10.8431	10.8431	10.8431	10.8431	10.8431
2	10.8431	**28.0843**	10.8431	10.8431	10.8431	10.8431	10.8431
3	10.8431	10.8431	**28.0843**	10.8431	10.8431	10.8431	10.8431
4	10.8431	10.8431	10.8431	**28.0843**	10.8431	10.8431	10.8431
5	10.8431	10.8431	10.8431	10.8431	**28.0843**	10.8431	10.8431
6	10.8431	10.8431	10.8431	10.8431	10.8431	**28.0843**	10.8431
7	10.8431	10.8431	10.8431	10.8431	10.8431	10.8431	**28.0843**

Model 2a. Fixed Linear Exact Age (using *time* = age − 18), Random Intercept

$$\begin{aligned}&\text{Level 1: } \text{risky}_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Age}_{ti} - 18\right) + e_{ti}\\ &\text{Level 2: Intercept: } \quad \beta_{0i} = \gamma_{00} + U_{0i}\\ &\qquad\qquad \text{Linear Age: } \beta_{1i} = \gamma_{10}\end{aligned}$$

```
print("Model 2a: Fixed Linear Time, Random Intercept Model")
FixLin = lmer(data=Example1, REML=FALSE, formula=risky~1+time+(1|PersonID))
print("Show results using Satterthwaite DDF"); summary(FixLin, ddf="Satterthwaite")
```

```
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method
['lmerModLmerTest']
Formula: risky ~ 1 + time + (1 | PersonID)
   Data: Example1
     AIC      BIC   logLik deviance df.resid
    7806     7827    -3899     7798     1396 → deviance = -2LL

Random effects:
 Groups   Name        Variance Std.Dev.
 PersonID (Intercept) 11.659   3.4146   → L2 BP random intercept variance UP 7.5%!
 Residual             11.382   3.3737   → L1 WP residual variance down 34.0%
Number of obs: 1400, groups:  PersonID, 200
```

Relative to the empty means, random intercept model 1, the fixed linear effect of time explained 34% of the residual variance (which made the random intercept variance increase by 7.5% due to its smaller correction factor). In total, we have explained 34% of the 61% of the variance that was originally within persons over time (from the empty means, random intercept model 1).

```
Fixed effects:
               Estimate  Std. Error          df t value  Pr(>|t|)  Fixed effects:
(Intercept)   22.742068    0.290969  320.826562  78.160 < 2.2e-16  gamma00
time           1.119446    0.045029 1200.347992  24.861 < 2.2e-16  gamma10
```

```
print("Get conditional mean per occasion from value of time predictor")
print("Y-hat Int: Age=12 Time=-6"); contest1D(FixLin, ddf="Satterthwaite", L=c(1,-6))
print("Y-hat Int: Age=13 Time=-5"); contest1D(FixLin, ddf="Satterthwaite", L=c(1,-5))
print("Y-hat Int: Age=14 Time=-4"); contest1D(FixLin, ddf="Satterthwaite", L=c(1,-4))
print("Y-hat Int: Age=15 Time=-3"); contest1D(FixLin, ddf="Satterthwaite", L=c(1,-3))
print("Y-hat Int: Age=16 Time=-2"); contest1D(FixLin, ddf="Satterthwaite", L=c(1,-2))
print("Y-hat Int: Age=17 Time=-1"); contest1D(FixLin, ddf="Satterthwaite", L=c(1,-1))
print("Y-hat Int: Age=18 Time= 0"); contest1D(FixLin, ddf="Satterthwaite", L=c(1, 0))
```

```
[1] "Y-hat Int: Age=12 Time=-6"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, -6))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 16.025391 0.29101186 321.00584 55.067829 1.3198206e-165

[1] "Y-hat Int: Age=13 Time=-5"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, -5))
   Estimate Std. Error        df  t value       Pr(>|t|)
1 17.144837 0.27303083 251.24093 62.79451 1.3610652e-155

[1] "Y-hat Int: Age=14 Time=-4"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, -4))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 18.264284 0.26164656 212.38659 69.805174 1.9397015e-148
```

```
[1] "Y-hat Int: Age=15 Time=-3"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, -3))
  Estimate Std. Error        df   t value        Pr(>|t|)
1 19.38373 0.25773467 199.99996 75.208079 1.2571042e-148

[1] "Y-hat Int: Age=16 Time=-2"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, -2))
   Estimate Std. Error        df   t value        Pr(>|t|)
1 20.503176 0.26163057 212.33496 78.366897 1.1001159e-158

[1] "Y-hat Int: Age=17 Time=-1"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, -1))
   Estimate Std. Error        df  t value        Pr(>|t|)
1 21.622622 0.27300019 251.13072 79.20369 1.2036821e-179

[1] "Y-hat Int: Age=18 Time= 0"
> contest1D(FixLin, ddf = "Satterthwaite", L = c(1, 0))
   Estimate Std. Error        df   t value        Pr(>|t|)
1 22.742068 0.29096873 320.82656 78.159836 6.5050635e-211
```

Model 2b. Random Linear Exact Age

$$\begin{array}{|l|}\hline \text{Level 1: } \text{risky}_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Age}_{ti} - 18\right) + e_{ti} \\ \text{Level 2: Intercept: } \quad \beta_{0i} = \gamma_{00} + U_{0i} \\ \qquad\qquad \text{Linear Age: } \beta_{1i} = \gamma_{10} + U_{1i} \\ \hline \end{array}$$

How to calculate the predicted V matrix variances and covariances in a random linear age model:

$$\mathbf{V}_i \text{ matrix: Variance}\left[y_{time}\right] = \tau^2_{U_0} + \left[\left(\text{Age}-18\right)^2 \tau^2_{U_1}\right] + \left[2\left(\text{Age}-18\right)\tau_{U_{01}}\right] + \sigma^2_e$$

$$\mathbf{V}_i \text{ matrix: Covariance}\left[y_A, y_B\right] = \tau^2_{U_0} + \left[\left(A+B\right)\tau_{U_{01}}\right] + \left[\left(AB\right)\tau^2_{U_1}\right]$$

V matrix (marginal predicted variance–covariance matrix) from SAS MIXED output:

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	**22.2046**	11.7266	10.7657	9.3141	8.6628	7.2868	6.2674
2	11.7266	**19.9982**	11.0367	10.6686	10.5035	10.1546	9.8961
3	10.7657	11.0367	**19.9026**	11.4082	11.5084	11.7203	11.8773
4	9.3141	10.6686	11.4082	**21.2432**	13.0265	14.0854	14.8699
5	8.6628	10.5035	11.5084	13.0265	**22.4255**	15.1466	16.2127
6	7.2868	10.1546	11.7203	14.0854	15.1466	**26.1067**	19.0497
7	6.2674	9.8961	11.8773	14.8699	16.2127	19.0497	**29.8692**

```
print("Model 2b: Random Linear Time Model")
RandLin = lmer(data=Example1, REML=FALSE, formula=risky~1+time+(1+time|PersonID))
print("Show results using Satterthwaite DDF"); summary(RandLin, ddf="Satterthwaite")
```

```
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method
['lmerModLmerTest']
Formula: risky ~ 1 + time + (1 + time | PersonID)
   Data: Example1

     AIC      BIC   logLik deviance df.resid
  7689.2   7720.7  -3838.6   7677.2     1394
```
→ deviance = −2LL

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 PersonID (Intercept) 21.5157  4.63850        → L2 BP random intercept variance
          time         0.5701  0.75505  0.695 → L2 BP random linear slope variance
 Residual              8.7175  2.95255        → L1 WP residual variance
Number of obs: 1400, groups:  PersonID, 200

Fixed effects:
             Estimate Std. Error        df t value  Pr(>|t|)  Fixed effects:
(Intercept)  22.74344    0.35751 199.82934  63.615 < 2.2e-16  gamma00
time          1.11989    0.06639 199.86684  16.868 < 2.2e-16  gamma10
```

```
print("Get conditional mean per occasion from value of time predictor")
print("Y-hat Int: Age=12 Time=-6"); contest1D(RandLin, ddf="Satterthwaite", L=c(1,-6))
print("Y-hat Int: Age=13 Time=-5"); contest1D(RandLin, ddf="Satterthwaite", L=c(1,-5))
print("Y-hat Int: Age=14 Time=-4"); contest1D(RandLin, ddf="Satterthwaite", L=c(1,-4))
print("Y-hat Int: Age=15 Time=-3"); contest1D(RandLin, ddf="Satterthwaite", L=c(1,-3))
print("Y-hat Int: Age=16 Time=-2"); contest1D(RandLin, ddf="Satterthwaite", L=c(1,-2))
print("Y-hat Int: Age=17 Time=-1"); contest1D(RandLin, ddf="Satterthwaite", L=c(1,-1))
print("Y-hat Int: Age=18 Time= 0"); contest1D(RandLin, ddf="Satterthwaite", L=c(1, 0))
```

```
[1] "Y-hat Int: Age=12 Time=-6"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, -6))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 16.024121 0.29060005 200.21585 55.141492 5.2297177e-123

[1] "Y-hat Int: Age=13 Time=-5"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, -5))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 17.144007 0.26389054 200.02622 64.966358 2.0101315e-136

[1] "Y-hat Int: Age=14 Time=-4"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, -4))
   Estimate Std. Error       df   t value       Pr(>|t|)
1 18.263894 0.25227627 199.9485 72.396401 2.0815199e-145

[1] "Y-hat Int: Age=15 Time=-3"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, -3))
  Estimate Std. Error        df   t value       Pr(>|t|)
1 19.38378 0.25780552 199.99493 75.187607 1.3330775e-148

[1] "Y-hat Int: Age=16 Time=-2"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, -2))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 20.503667 0.27946258 200.01165 73.368202 1.4784211e-146

[1] "Y-hat Int: Age=17 Time=-1"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, -1))
   Estimate Std. Error       df   t value       Pr(>|t|)
1 21.623554 0.31392715 199.9377 68.880802 3.0229488e-141

[1] "Y-hat Int: Age=18 Time= 0"
> contest1D(RandLin, ddf = "Satterthwaite", L = c(1, 0))
  Estimate Std. Error        df   t value       Pr(>|t|)
1 22.74344 0.35751427 199.82934 63.615477 1.3641649e-134
```

```
print("Does random linear time slope improve model fit?")
ranova(RandLin, reduce.term=TRUE)
# LRT for removing random linear slope and covariance

ANOVA-like table for random-effects: Single term deletions

Model:
risky ~ time + (1 + time | PersonID)
                              npar   logLik     AIC     LRT Df Pr(>Chisq)
<none>                           6 -3838.59 7689.19
time in (1 + time | PersonID)    4 -3898.99 7805.97 120.785  2 < 2.22e-16
```

Is the random linear age model (2b) better than the fixed linear age, random intercept model (2a)?
Yes, −2ΔLL= 120, which is bigger than the critical value of 5.99ish on df = ~2ish

We will not calculate pseudo-R^2 for this random linear age slope model relative to the previous fixed linear age slope, random intercept model because random effects *do not* explain variance—they partition it instead.

Model 2c. Fixed Quadratic, Random Linear Exact Age

$$\text{Level 1: } \text{risky}_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Age}_{ti} - 18\right) + \beta_{2i}\left(\text{Age}_{ti} - 18\right)^2 + e_{ti}$$
$$\text{Level 2: Intercept: } \beta_{0i} = \gamma_{00} + U_{0i}$$
$$\text{Linear Age: } \beta_{1i} = \gamma_{10} + U_{1i}$$
$$\text{Quadratic Age: } \beta_{2i} = \gamma_{20}$$

```
print("Model 2c: Fixed Quadratic, Random Linear Time Model")
FixQuad = lmer(data=Example1, REML=FALSE,
formula=risky~1+time+I(time^2)+(1+time|PersonID))
print("Show results using Satterthwaite DDF"); summary(FixQuad, ddf="Satterthwaite")

Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method
['lmerModLmerTest']
Formula: risky ~ 1 + time + I(time^2) + (1 + time | PersonID)
   Data: Example1

     AIC      BIC   logLik deviance df.resid
  7648.8   7685.5  -3817.4   7634.8     1393

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 PersonID (Intercept) 21.69614 4.65791         → L2 BP random int variance up 0.83%
          time         0.58484 0.76475  0.695 → L2 BP random linear variance up 2.58%
 Residual              8.35155 2.88991         → L1 WP residual variance down 4.20%
Number of obs: 1400, groups:  PersonID, 200
```

Relative to the random linear time model 2b, fixed quadratic age explained another 4% of the residual variance (which increased the random intercept variance and the random linear time slope variance).

```
Fixed effects:
               Estimate  Std. Error          df t value          Pr(>|t|) Fixed effects:
(Intercept)   23.465495    0.374026  238.632073 62.7376         < 2.2e-16 gamma00
time           1.987721    0.147608 1187.654639 13.4662         < 2.2e-16 gamma10
I(time^2)      0.144630    0.021966 1010.047530  6.5843 0.00000000007335 gamma20

print("Get conditional mean per occasion from values of time predictors")
print("Int: Age=12 Time=-6"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1,-6,36))
print("Int: Age=13 Time=-5"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1,-5,25))
print("Int: Age=14 Time=-4"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1,-4,16))
```

```
print("Int: Age=15 Time=-3"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1,-3, 9))
print("Int: Age=16 Time=-2"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1,-2, 4))
print("Int: Age=17 Time=-1"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1,-1, 1))
print("Int: Age=18 Time= 0"); contest1D(FixQuad, ddf="Satterthwaite", L=c(1, 0, 0))
```

```
[1] "Int: Age=12 Time=-6"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, -6, 36))
   Estimate Std. Error        df   t value      Pr(>|t|)
1 16.745845 0.31078751 259.74601 53.881974 5.355379e-143

[1] "Int: Age=13 Time=-5"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, -5, 25))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 17.142637 0.26402623 200.02909 64.927782 2.2447017e-136

[1] "Int: Age=14 Time=-4"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, -4, 16))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 17.828689 0.26085757 228.09759 68.346453 6.5734557e-154

[1] "Int: Age=15 Time=-3"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, -3, 9))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 18.804001 0.27245463 248.65911 69.016998 3.4538463e-164

[1] "Int: Age=16 Time=-2"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, -2, 4))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 20.068573 0.28718734 222.84258 69.879727 1.5657245e-153

[1] "Int: Age=17 Time=-1"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, -1, 1))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 21.622404 0.31396479 199.92102 68.868882 3.1842962e-141

[1] "Int: Age=18 Time= 0"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(1, 0, 0))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 23.465495 0.37402615 238.63207 62.737579 2.6829317e-150
```

```
print("Get instantaneous linear slope per occasion from 2*value of time predictor")
print("Linear: Age=12 Time=-6"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1,-12))
print("Linear: Age=13 Time=-5"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1,-10))
print("Linear: Age=14 Time=-4"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1, -8))
print("Linear: Age=15 Time=-3"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1, -6))
print("Linear: Age=16 Time=-2"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1, -4))
print("Linear: Age=17 Time=-1"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1, -2))
print("Linear: Age=18 Time= 0"); contest1D(FixQuad, ddf="Satterthwaite", L=c(0,1,  0))
```

```
[1] "Linear: Age=12 Time=-6"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1, -12))
    Estimate Std. Error        df  t value    Pr(>|t|)
1 0.25216237 0.14759362 1188.1102 1.708491 0.087806465

[1] "Linear: Age=13 Time=-5"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1, -10))
    Estimate Std. Error        df   t value        Pr(>|t|)
1 0.54142216 0.11015724 931.86547 4.9149938 0.0000010487003

[1] "Linear: Age=14 Time=-4"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1, -8))
    Estimate  Std. Error        df   t value      Pr(>|t|)
1 0.83068194 0.079657679 396.08942 10.428147 1.1665323e-22

[1] "Linear: Age=15 Time=-3"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1, -6))
```

```
   Estimate  Std. Error        df   t value       Pr(>|t|)
1 1.1199417 0.066453445 199.81591 16.853027 3.1145521e-40

[1] "Linear: Age=16 Time=-2"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1, -4))
   Estimate  Std. Error        df   t value       Pr(>|t|)
1 1.4092015 0.079666567 396.31147 17.688744 4.9919836e-52

[1] "Linear: Age=17 Time=-1"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1, -2))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 1.6984613  0.1101701 931.93033 15.416718 6.1817713e-48

[1] "Linear: Age=18 Time= 0"
> contest1D(FixQuad, ddf = "Satterthwaite", L = c(0, 1,  0))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 1.9877211 0.14760801 1187.6546 13.466214 1.4298239e-38
```

```
print("Save yhat and correlation of yhat with y")
Example1$PredUnc = predict(FixQuad, re.form=NA)
rUnc = cor.test(Example1$PredUnc, Example1$risky, method="pearson")
print("Total R2 for Unconditional Time Model"); rUnc$estimate^2
```

0.18858759 → Total variance in risky behavior explained by fixed quadratic time

---

Model 2d. Random Quadratic Exact Age

$$\begin{aligned}
&\text{Level 1:}\ \ y_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Age}_{ti} - 18\right) + \beta_{2i}\left(\text{Age}_{ti} - 18\right)^2 + e_{ti} \\
&\text{Level 2:}\ \ \text{Intercept:} \qquad \beta_{0i} = \gamma_{00} + U_{0i} \\
&\qquad\qquad\ \ \text{Linear Age:} \qquad \beta_{1i} = \gamma_{10} + U_{1i} \\
&\qquad\qquad\ \ \text{Quadratic Age:} \ \ \beta_{2i} = \gamma_{20} + U_{2i}
\end{aligned}$$

How to calculate the predicted V matrix variances and covariances in a random quadratic age model:

Predicted Variance at Age *T*:

$$\text{Var}(y_T) = \sigma^2_e + \tau^2_{U_0} + 2*T*\tau_{U_{01}} + T^2*\tau^2_{U_1} + 2*T^2*\tau_{U_{02}} + 2*T^3*\tau_{U_{12}} + T^4*\tau^2_{U_2}$$

Predicted Covariance between Time *A* and *B*:

$$\text{Cov}(y_A, y_B) = \tau^2_{U_0} + (A+B)*\tau_{U_{01}} + (AB)*\tau^2_{U_1} + (A^2+B^2)*\tau_{U_{02}} + (AB^2)+(A^2B)*\tau_{U_{12}} + (A^2B^2)*\tau^2_{U_2}$$

V matrix (marginal predicted variance–covariance matrix) from SAS MIXED output:

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	**21.5024**	11.2582	10.1076	8.4858	7.8036	6.4548	5.5367
2	11.2582	**19.6663**	11.6825	11.3511	11.0763	10.2386	9.3932
3	10.1076	11.6825	**20.1821**	12.5393	12.5125	12.0949	11.4698
4	8.4858	11.3511	12.5393	**21.8080**	14.2123	14.6183	14.5677
5	7.8036	11.0763	12.5125	14.2123	**22.7692**	15.6407	15.9425
6	6.4548	10.2386	12.0949	14.6183	15.6407	**25.5552**	18.8163
7	5.5367	9.3932	11.4698	14.5677	15.9425	18.8163	**28.8959**

```
print("Model 2d: Random Quadratic Time Model -- reports convergence problem")
RandQuad = lmer(data=Example1, REML=FALSE,
formula=risky~1+time+I(time^2)+(1+time+I(time^2)|PersonID))
print("Show results using Satterthwaite DDF"); summary(RandQuad, ddf="Satterthwaite")
```

```
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method
['lmerModLmerTest']
Formula: risky ~ 1 + time + I(time^2) + (1 + time + I(time^2) | PersonID)
   Data: Example1

     AIC      BIC   logLik deviance df.resid
  7649.5   7701.9  -3814.8   7629.5     1390
Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 PersonID (Intercept) 21.301046 4.61531                → L2 BP random intercept var
          time         1.357314 1.16504   0.462        → L2 BP random linear slope var
          I(time^2)    0.021513 0.14667  -0.006  0.748 → L2 BP random quad slope var
 Residual              7.977601 2.82446                → L1 WP residual variance
Number of obs: 1400, groups:  PersonID, 200

Fixed effects:
              Estimate Std. Error         df t value       Pr(>|t|)  Fixed Effects:
(Intercept)  23.468793   0.369629 199.721279 63.4928      < 2.2e-16  gamma00
time          1.991168   0.157760 192.748323 12.6215      < 2.2e-16  gamma10
I(time^2)     0.145098   0.023889 190.687536  6.0738 0.00000000664  gamma20

optimizer (nloptwrap) convergence code: 0 (OK)
Model failed to converge with max|grad| = 0.0420868 (tol = 0.002, component 1)
```

```
print("Get conditional mean per occasion from values of time predictors")
print("Int: Age=12 Time=-6"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1,-6,36))
print("Int: Age=13 Time=-5"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1,-5,25))
print("Int: Age=14 Time=-4"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1,-4,16))
print("Int: Age=15 Time=-3"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1,-3, 9))
print("Int: Age=16 Time=-2"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1,-2, 4))
print("Int: Age=17 Time=-1"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1,-1, 1))
print("Int: Age=18 Time= 0"); contest1D(RandQuad, ddf="Satterthwaite", L=c(1, 0, 0))
```

```
[1] "Int: Age=12 Time=-6"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, -6, 36))
   Estimate Std. Error        df   t value      Pr(>|t|)
1 16.745307 0.30618375 200.19863 54.690386 2.4852507e-122

[1] "Int: Age=13 Time=-5"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, -5, 25))
   Estimate Std. Error        df  t value      Pr(>|t|)
1 17.140399 0.26420646 200.08004 64.87502 2.4824813e-136

[1] "Int: Age=14 Time=-4"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, -4, 16))
   Estimate Std. Error        df  t value      Pr(>|t|)
1 17.825687 0.26851167 200.17284 66.38701 2.7376689e-138

[1] "Int: Age=15 Time=-3"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, -3, 9))
  Estimate Std. Error        df  t value      Pr(>|t|)
1 18.80117 0.28285256 200.33005 66.46986 1.8155268e-138

[1] "Int: Age=16 Time=-2"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, -2, 4))
   Estimate Std. Error        df   t value      Pr(>|t|)
1 20.066849 0.29427814 200.04089 68.190075 1.8618247e-140

[1] "Int: Age=17 Time=-1"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, -1, 1))
   Estimate Std. Error        df   t value      Pr(>|t|)
1 21.622723 0.31408725 199.87051 68.843047 3.6223026e-141
```

```
[1] "Int: Age=18 Time= 0"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(1, 0, 0))
   Estimate Std. Error        df   t value       Pr(>|t|)
1 23.468793 0.36962944 199.72128 63.492759 2.2078691e-134
```

```
print("Get instantaneous linear slope per occasion from 2*value of time predictor")
print("Linear: Age=12 Time=-6"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1,-12))
print("Linear: Age=13 Time=-5"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1,-10))
print("Linear: Age=14 Time=-4"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1, -8))
print("Linear: Age=15 Time=-3"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1, -6))
print("Linear: Age=16 Time=-2"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1, -4))
print("Linear: Age=17 Time=-1"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1, -2))
print("Linear: Age=18 Time= 0"); contest1D(RandQuad, ddf="Satterthwaite", L=c(0,1,  0))
```

```
[1] "Linear: Age=12 Time=-6"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1, -12))
    Estimate Std. Error        df   t value   Pr(>|t|)
1 0.24999433 0.15825474 192.25437 1.5796957 0.11582043

[1] "Linear: Age=13 Time=-5"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1, -10))
    Estimate Std. Error        df   t value        Pr(>|t|)
1 0.54018986 0.11663858 194.11039 4.6313139 0.0000066450715

[1] "Linear: Age=14 Time=-4"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1, -8))
   Estimate  Std. Error        df   t value      Pr(>|t|)
1 0.8303854 0.082037322 198.20623 10.122044 1.1186139e-19

[1] "Linear: Age=15 Time=-3"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1, -6))
   Estimate Std. Error        df   t value      Pr(>|t|)
1 1.1205809 0.06649254 199.65869 16.852732 3.1955649e-40

[1] "Linear: Age=16 Time=-2"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1, -4))
   Estimate  Std. Error        df  t value     Pr(>|t|)
1 1.4107765 0.081718813 195.72192 17.26379 3.433291e-41

[1] "Linear: Age=17 Time=-1"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1, -2))
  Estimate Std. Error        df   t value      Pr(>|t|)
1 1.700972 0.11619054 193.80045 14.639505 3.5917278e-33

[1] "Linear: Age=18 Time= 0"
> contest1D(RandQuad, ddf = "Satterthwaite", L = c(0, 1,  0))
  Estimate Std. Error        df   t value      Pr(>|t|)
1 1.9911675  0.1577596 192.74832 12.62153 5.1933749e-27
```

```
print("Does random quadratic time slope improve model fit?")
ranova(RandQuad, reduce.term=TRUE)
# LRT for removing random quadratic slope and covariances
```

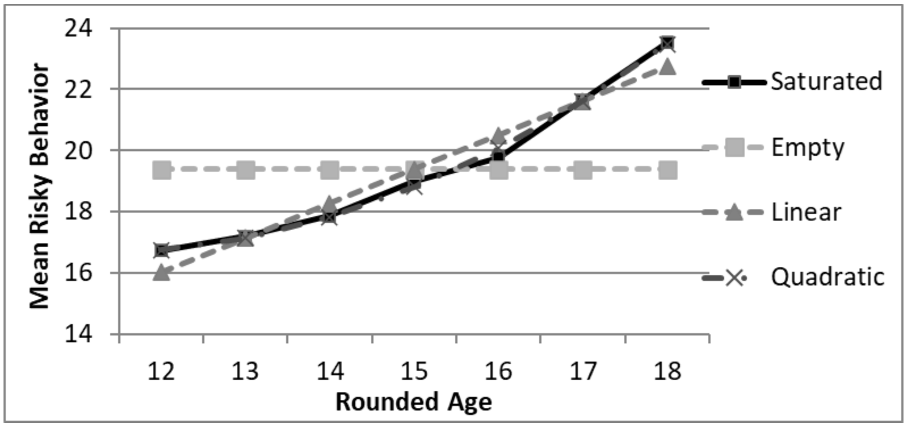
Is the random quadratic model (2d) better than the fixed quadratic, random linear model (2c)?

No, −2ΔLL= 5.26, which is smaller than the critical value of 7.82ish on df= ~3ish

```
ANOVA-like table for random-effects: Single term deletions

Model:
risky ~ time + I(time^2) + (1 + time + I(time^2) | PersonID)
                                              npar   logLik     AIC      LRT Df Pr(>Chisq)
<none>                                          10 -3814.75 7649.50
time in (1 + time + I(time^2) | PersonID)        7 -3825.44 7664.89 21.38519  3 0.00008756
I(time^2) in (1 + time + I(time^2) | PersonID)   7 -3817.38 7648.77  5.26451  3    0.15342
```

We will not calculate pseudo-R^2 for this random quadratic age model relative to the previous fixed linear age, random intercept model because random effects *do not* explain variance—they partition it instead.



Given how well the quadratic age model appears to fit the rounded means, we can move on by adding a time-invariant predictor of mothers' attitudes (as measured at age 12).

Model 2e. Random Quadratic Exact Age + AR1 residual correlation

Let us see if the assumption of independent level-1 residuals holds by examining the improvement in fit from adding a first-order auto-regressive correlation in the level-1 residual R matrix… to do so, we need to switch from lmer (from the lme4 package) to lme (from the nlme package), although the latter doesn't (usually) provide the correct Satterthwaite denominator degrees of freedom (at least in most of the cases I've seen).

```
print("Model 2e: Test AR1 Residual Correlation in Random Quadratic Time Model in LME")
RandQuadAR1 = lme(data=Example1, method="ML", risky~1+time+I(time^2),
                  random=~1+time+I(time^2)|PersonID,
                  correlation=(corAR1(form=~as.numeric(occasion)|PersonID)))
print("Show results using incorrect DDF"); summary(RandQuadAR1)

Linear mixed-effects model fit by maximum likelihood
  Data: Example1
        AIC       BIC    logLik
  7651.3703 7709.0568 -3814.6852

Random effects:
 Formula: ~1 + time + I(time^2) | PersonID
 Structure: General positive-definite, Log-Cholesky parametrization

             StdDev     Corr
(Intercept) 4.58892176 (Intr)   time
time        1.11368087  0.463
I(time^2)   0.13650457 -0.028  0.726
Residual    2.84971411

Correlation Structure: AR(1)
 Formula: ~as.numeric(occasion) | PersonID
 Parameter estimate(s):
       Phi
0.020333681 → lag-1 residual correlation r (so r^2 for lag 2, r^3 for lag 3…)


Fixed effects:  risky ~ 1 + time + I(time^2) → Note incorrect denominator DF!
                Value  Std.Error   DF   t-value p-value
(Intercept) 23.4697341 0.36975920 1198 63.473022       0
time         1.9911650 0.15774535 1198 12.622654       0
I(time^2)    0.1450396 0.02388494 1198  6.072428       0

print("Show G, R, and V matrices for first person")
G=getVarCov(RandQuadAR1, individual="1", type="random.effects"); G
```

```
Random effects variance covariance matrix
             (Intercept)     time  I(time^2)
(Intercept)    21.058000 2.36840 -0.017428
time            2.368400 1.24030  0.110320
I(time^2)      -0.017428 0.11032  0.018633
  Standard Deviations: 4.5889 1.1137 0.1365
```

```
R=getVarCov(RandQuadAR1, individual="1", type="conditional"); R
```

```
PersonID 1
Conditional variance covariance matrix
                 1               2           3           4            5              6                7
1 8.12090000000000 0.165130000000 0.0033576000 0.000068273 0.0000013882 0.000000028228 0.00000000057398
2 0.16513000000000 8.120900000000 0.1651300000 0.003357600 0.0000682730 0.000001388200 0.00000002822800
3 0.00335760000000 0.165130000000 8.1209000000 0.165130000 0.0033576000 0.000068273000 0.00000138820000
4 0.00006827300000 0.003357600000 0.1651300000 8.120900000 0.1651300000 0.003357600000 0.00006827300000
5 0.00000138820000 0.000068273000 0.0033576000 0.165130000 8.1209000000 0.165130000000 0.00335760000000
6 0.00000002822800 0.000001388200 0.0000682730 0.003357600 0.1651300000 8.120900000000 0.16513000000000
7 0.00000000057398 0.000000028228 0.0000013882 0.000068273 0.0033576000 0.165130000000 8.12090000000000
  Standard Deviations: 2.8497 2.8497 2.8497 2.8497 2.8497 2.8497 2.8497
```

```
V=getVarCov(RandQuadAR1, individual="1", type="marginal"); V
```

```
PersonID 1
Marginal variance covariance matrix
        1       2      3       4       5       6       7
1 21.3730 11.3350 10.091  8.5226  7.8485  6.4815  5.5188
2 11.3350 19.7100 11.751 11.2760 11.0120 10.2130  9.4053
3 10.0910 11.7510 20.210 12.5910 12.4140 12.0440 11.4770
4  8.5226 11.2760 12.591 21.8200 14.2520 14.5380 14.5410
5  7.8485 11.0120 12.414 14.2520 22.7880 15.7100 15.8920
6  6.4815 10.2130 12.044 14.5380 15.7100 25.5800 18.8500
7  5.5188  9.4053 11.477 14.5410 15.8920 18.8500 28.8300
  Standard Deviations: 4.6231 4.4395 4.4955 4.6711 4.7737 5.0576 5.3694
```

```
print("Does AR1 residual correlation improve fit?")
print("Have to re-run random quadratic time model using lme to get LRT")
RandQuadlme = lme(data=Example1, method="ML", risky~1+time+I(time^2),
                  random=~1+time+I(time^2)|PersonID)
anova(RandQuadAR1,RandQuadlme) # anova does LRT using lme versions
```

```
> anova(RandQuadAR1, RandQuadlme)
            Model df       AIC       BIC    logLik   Test    L.Ratio p-value
RandQuadAR1     1 11 7651.3703 7709.0568 -3814.6852
RandQuadlme     2 10 7649.5045 7701.9468 -3814.7522 1 vs 2 0.13413486  0.7142
```

The LRT result indicates that the AR1 residual correlation does not improve model fit (so we can remove it).

Model 3. Add Attitudes at Age 12 as Predictor of Intercept, Linear, and Quadratic Age Slopes

Level 1: $\text{risky}_{ti} = \beta_{0i} + \beta_{1i}\left(\text{Age}_{ti} - 18\right) + \beta_{2i}\left(\text{Age}_{ti} - 18\right)^2 + e_{ti}$

Level 2:

Intercept: $\beta_{0i} = \gamma_{00} + \gamma_{01}\left(\text{Attitudes}_i - 4\right) + U_{0i}$

Linear: $\beta_{1i} = \gamma_{10} + \gamma_{11}\left(\text{Attitudes}_i - 4\right) + U_{1i}$

Quadratic: $\beta_{2i} = \gamma_{20} + \gamma_{21}\left(\text{Attitudes}_i - 4\right)$

Note that we have not included the random quadratic age slope variance given the results of model 2d. That means the quadratic effect of age is systematically varying in this model (instead of random, like the intercept and linear age effects), such that it will reduce the WP residual variance.

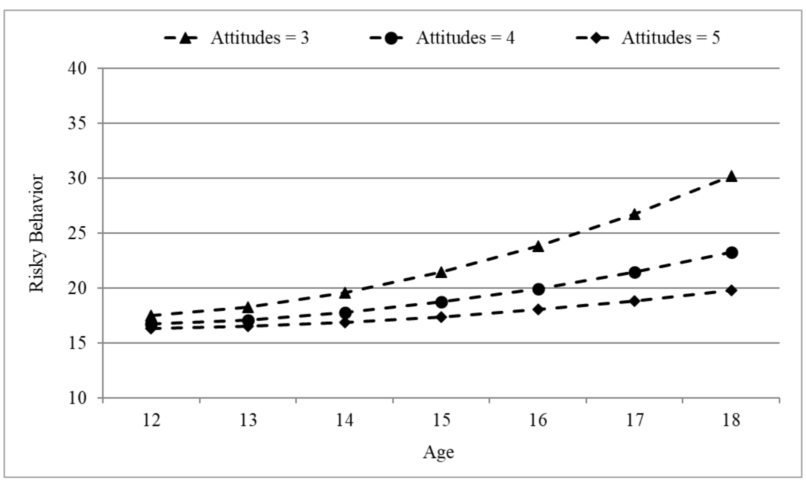
```
print("Model 3: Add Attitudes at Age 12 as
       Predictor of Intercept, Linear, and Quadratic Age Slopes")
Att = lmer(data=Example1, REML=FALSE, formula=risky~1+time+I(time^2)
                                     +att4+att4:time+att4:I(time^2)+(1+time|PersonID))
print("Show results using Satterthwaite DDF"); summary(Att, ddf="Satterthwaite")
```

```
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method
['lmerModLmerTest']
Formula: risky ~ 1 + time + I(time^2) + att4 + att4:time + att4:I(time^2) +
(1 + time | PersonID)
   Data: Example1

     AIC      BIC   logLik deviance df.resid
  7619.4   7671.9  -3799.7   7599.4     1390  → deviance = -2LL

Random effects:
 Groups   Name        Variance Std.Dev. Corr  Changes in Variance Relative to Model 2c:
 PersonID (Intercept) 18.08165 4.25225        → L2 BP random int var down by 16.63%
          time         0.48975 0.69982  0.634 → L2 BP random linear var down by 16.27%
 Residual              8.32597 2.88548        → L1 WP residual var down by 0.31%
Number of obs: 1400, groups:  PersonID, 200

Fixed effects:
                  Estimate  Std. Error          df t value        Pr(>|t|)
(Intercept)      23.299303    0.349986  244.791715 66.5722       < 2.2e-16 gamma00
time              1.948027    0.146148 1196.512435 13.3292       < 2.2e-16 gamma10
I(time^2)         0.142159    0.021989 1010.617242  6.4649 0.0000000001574 gamma20
att4             -3.476358    0.580431  244.877452 -5.9893 0.0000000074724 gamma01
time:att4        -0.900438    0.242102 1196.165674 -3.7193       0.0002091 gamma11
I(time^2):att4   -0.064085    0.036363 1011.398973 -1.7624       0.0783068 gamma21
```



The more conservative the mothers' attitude (higher values), the less positive (and almost less accelerating) the daughters' trajectory.

See chapter 7 for all the models and a results section.

Getting total R^2 as squared correlation between fixed-effect-predicted and actual outcomes:

```
print("Save yhat and correlation of yhat with y")
Example1$PredAtt = predict(Att, re.form=NA)
rAtt = cor.test(Example1$PredAtt, Example1$risky, method="pearson")
print("Total R2 from Time + Attitudes Model"); rAtt$estimate^2
```

0.23644833 → new total R² for risky predicted by time + attitudes

```
print("Total R2 change for attitudes relative to unconditional time model")
rAtt$estimate^2-rUnc$estimate^2
```

0.047860738 → change in total R² for addition of 3 attitudes slopes

Model 3. Add Attitudes at Age 12 as Predictor of Intercept, Linear, and Quadratic Age Slopes Single-level SEM version using lavaan on original wide-format data assuming balanced time

```
# Center predictor for analysis in wide data
Example1_wide$att4=Example1_wide$Attitude12-4

Model3Syntax = "
# Factor loadings fixed by *
Int =~  1*Risky12 +  1*Risky13 +  1*Risky14 +  1*Risky15 +  1*Risky16 +  1*Risky17 + 1*Risky18
Lin =~ -6*Risky12 + -5*Risky13 + -4*Risky14 + -3*Risky15 + -2*Risky16 + -1*Risky17 + 0*Risky18
Qua =~ 36*Risky12 + 25*Risky13 + 16*Risky14 +  9*Risky15 +  4*Risky16 +  1*Risky17 + 0*Risky18

# Factor intercepts estimated = fixed effects
  Int ~ 1;  Lin ~ 1;  Qua ~ 1
# Level-2 factor variances estimated for intercept and linear age (in G)
  Int ~~ Int;  Lin ~~ Lin;  Qua ~~ 0*Qua
# Level-2 factor covariances estimated (in G)
  Int ~~ Lin

# Per-occasion intercepts fixed to 0
  Risky12 ~ 0; Risky13 ~ 0; Risky14 ~ 0; Risky15 ~ 0
  Risky16 ~ 0; Risky17 ~ 0; Risky18 ~ 0

! Level-1 residual variances held equal (in R)
  Risky12 ~~ (ResVar)*Risky12; Risky13 ~~ (ResVar)*Risky13; Risky14 ~~ (ResVar)*Risky14
  Risky15 ~~ (ResVar)*Risky15; Risky16 ~~ (ResVar)*Risky16; Risky17 ~~ (ResVar)*Risky17
  Risky18 ~~ (ResVar)*Risky18

# Fixed effects of reasoning --> latent factors
  Int + Lin + Qua ~ att4
"
Model3SEM = lavaan(data=Example1_wide, model=Model3Syntax,
                     estimator="ML", mimic="mplus")
summary(Model3SEM, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

```
lavaan 0.6-10 ended normally after 53 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        16
  Number of equality constraints                     6

  Number of observations                           200
  Number of missing patterns                         1

Model Test User Model:
  Test statistic                                36.655 → LRT: ours vs saturated model
  Degrees of freedom                                32
  P-value (Chi-square)                           0.262

Model Test Baseline Model:
  Test statistic                               765.694 → LRT: null vs saturated model
  Degrees of freedom                                28
  P-value                                        0.000

User Model versus Baseline Model:
  Comparative Fit Index (CFI)                    0.994
  Tucker-Lewis Index (TLI)                       0.994

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)              -3803.896
  Loglikelihood unrestricted model (H1)      -3785.569

  Akaike (AIC)                                7627.793
  Bayesian (BIC)                              7660.776
  Sample-size adjusted Bayesian (BIC)         7629.095
```

```
Root Mean Square Error of Approximation:
  RMSEA                                          0.027
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.061
  P-value RMSEA <= 0.05                          0.844

Standardized Root Mean Square Residual:
  SRMR                                           0.045

Parameter Estimates:
  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Int =~
    Risky12           1.000                               4.733    1.025
    Risky13           1.000                               4.733    1.062
    Risky14           1.000                               4.733    1.072
    Risky15           1.000                               4.733    1.051
    Risky16           1.000                               4.733    1.002
    Risky17           1.000                               4.733    0.932
    Risky18           1.000                               4.733    0.853
  Lin =~
    Risky12          -6.000                              -5.164   -1.118
    Risky13          -5.000                              -4.303   -0.965
    Risky14          -4.000                              -3.442   -0.780
    Risky15          -3.000                              -2.582   -0.573
    Risky16          -2.000                              -1.721   -0.364
    Risky17          -1.000                              -0.861   -0.169
    Risky18           0.000                               0.000    0.000
  Qua =~
    Risky12          36.000                               1.178    0.255
    Risky13          25.000                               0.818    0.183
    Risky14          16.000                               0.523    0.119
    Risky15           9.000                               0.294    0.065
    Risky16           4.000                               0.131    0.028
    Risky17           1.000                               0.033    0.006
    Risky18           0.000                               0.000    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Int ~
    att4             -3.438    0.583   -5.900    0.000   -0.726   -0.437 → gamma01
  Lin ~
    att4             -0.846    0.246   -3.435    0.001   -0.984   -0.591 → gamma11
  Qua ~
    att4             -0.054    0.037   -1.462    0.144   -1.664   -1.000 → gamma21

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
 .Int ~~
   .Lin               1.882    0.356    5.292    0.000    0.637    0.637 → BP cov of U0 and U1

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Int              23.299    0.351   66.296    0.000    4.923    4.923 → gamma00
   .Lin               1.958    0.149   13.181    0.000    2.276    2.276 → gamma10
   .Qua               0.145    0.022    6.448    0.000    4.424    4.424 → gamma20
   .Risky12           0.000                               0.000    0.000
   .Risky13           0.000                               0.000    0.000
   .Risky14           0.000                               0.000    0.000
   .Risky15           0.000                               0.000    0.000
   .Risky16           0.000                               0.000    0.000
   .Risky17           0.000                               0.000    0.000
   .Risky18           0.000                               0.000    0.000
```

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Int              18.130    2.210    8.203    0.000    0.809    0.809 → L2 BP rand int var
   .Lin               0.482    0.079    6.072    0.000    0.651    0.651 → L2 BP rand linear var
   .Qua               0.000                               0.000    0.000 → L2 BP rand quad var=0
   .Risky12 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.394 → L1 WP residual var(s)
   .Risky13 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.423
   .Risky14 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.431
   .Risky15 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.415
   .Risky16 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.377
   .Risky17 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.326
   .Risky18 (RsVr)    8.408    0.376   22.361    0.000    8.408    0.273

R-Square:
                   Estimate
    Int               0.191
    Lin               0.349
    Qua               1.000
    Risky12           0.606
    Risky13           0.577
    Risky14           0.569
    Risky15           0.585
    Risky16           0.623
    Risky17           0.674
    Risky18           0.727
```

Activity: Repeat the analyses above predicting time-varying monitoring as an outcome instead. What do you conclude about its best-fitting unconditional model for time and the effects of attitudes?