Psychology 454: Latent Variable Modeling
Week 7: More on change

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA

November, 2016

Measuring structure at two (or more) time points

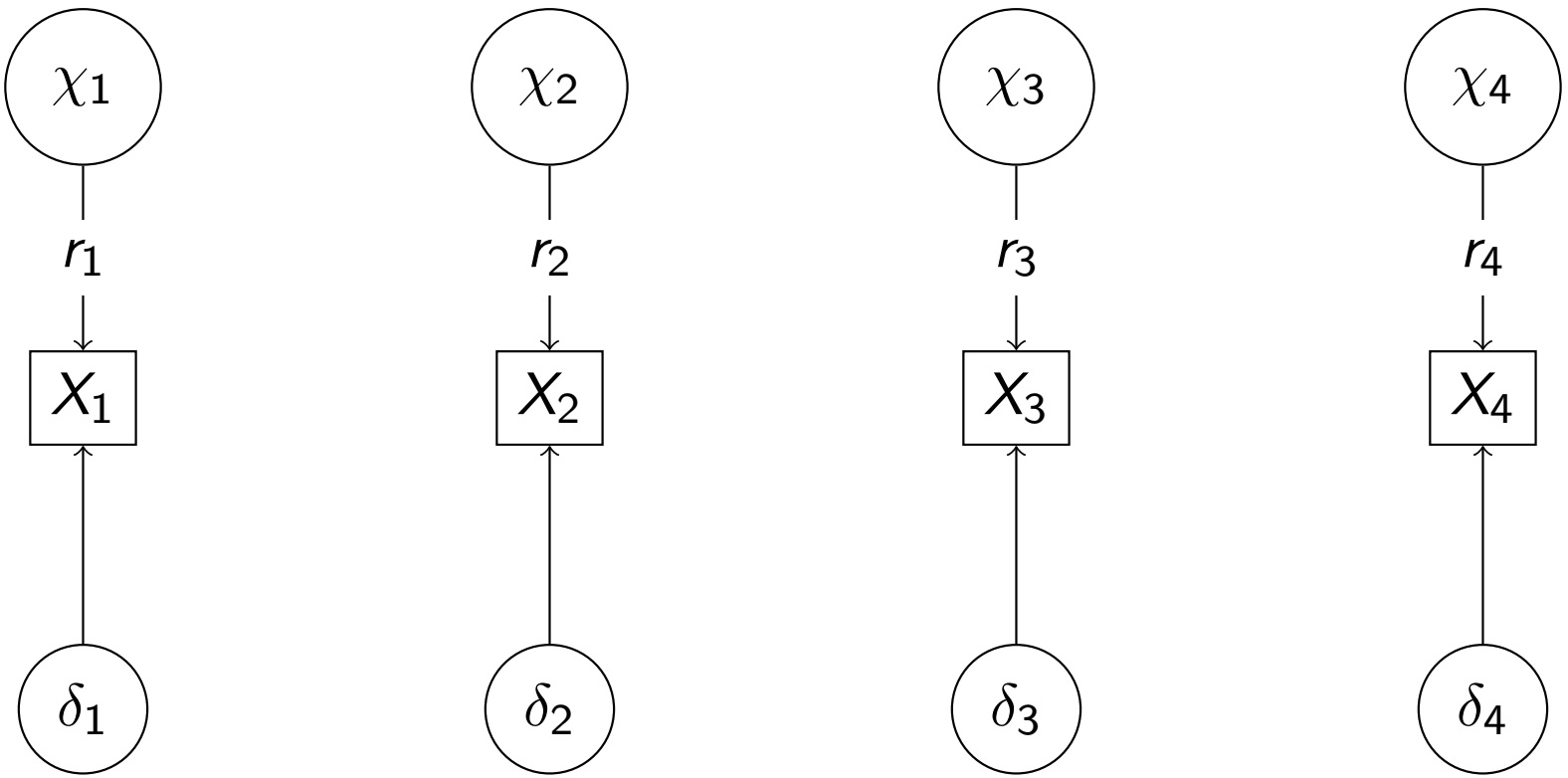
1. Is the structure the same
   - Structural Invariance (is the graph the same)
   - Measurement invariance (are the loadings the same)
   - Strong measurement invariance (are the item intercepts the same?)
   - Measuring change
2. Do the means change (is there growth)
   - This is the means of the latent trait, not the means of the items
3. Do the latent traits correlate across two or more occasions?
   - Just two occasions, can not separate trait from state effects
   - With > 2 occasions, can examine trait and state effects
4. Compare several different simulations

Observed scores over 4 time points

Do they differ in means? Do they measure the same thing?

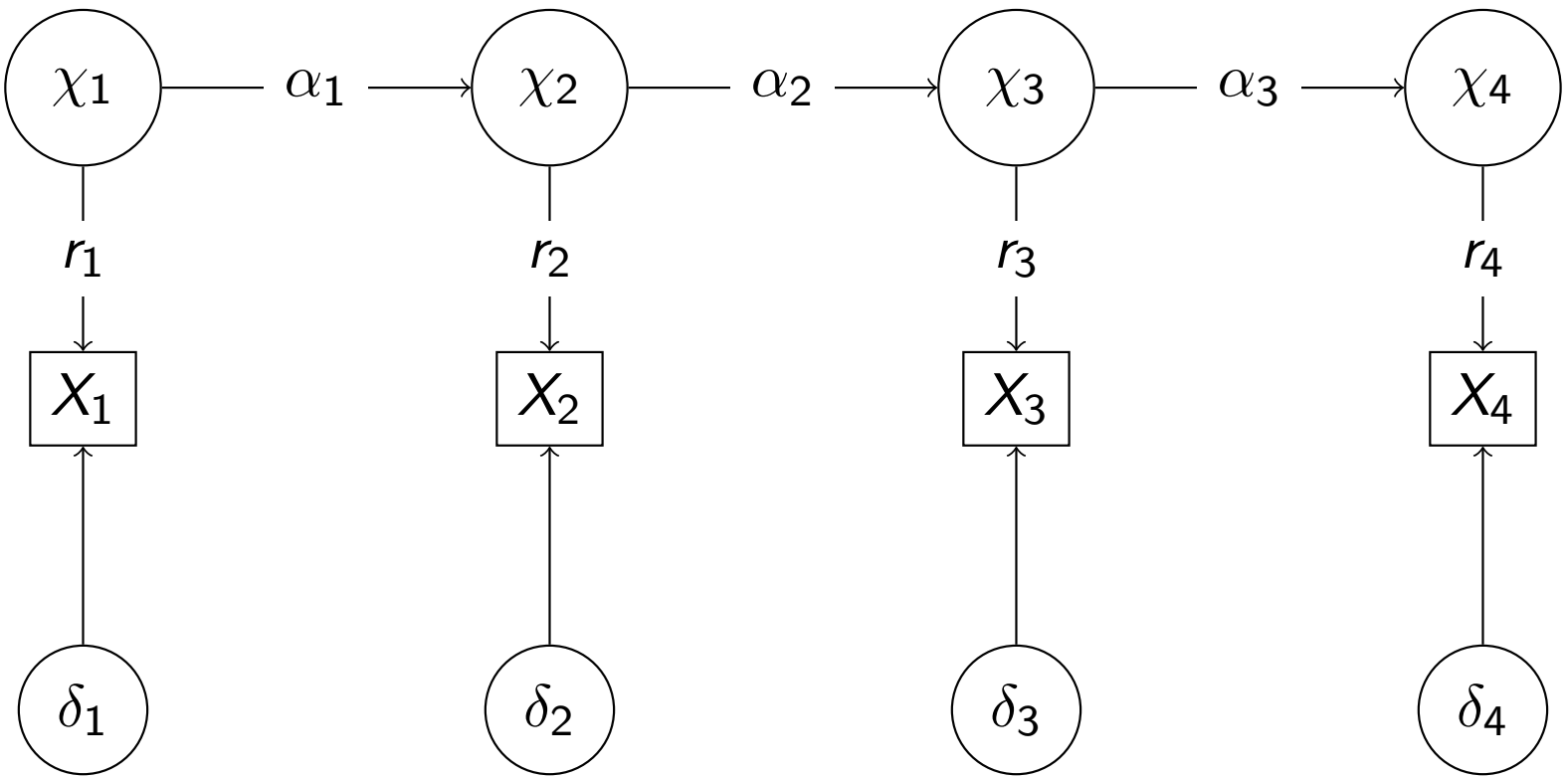
Modeled 4 time points

Are the measures measures of the same construct? Are the measures invariant?



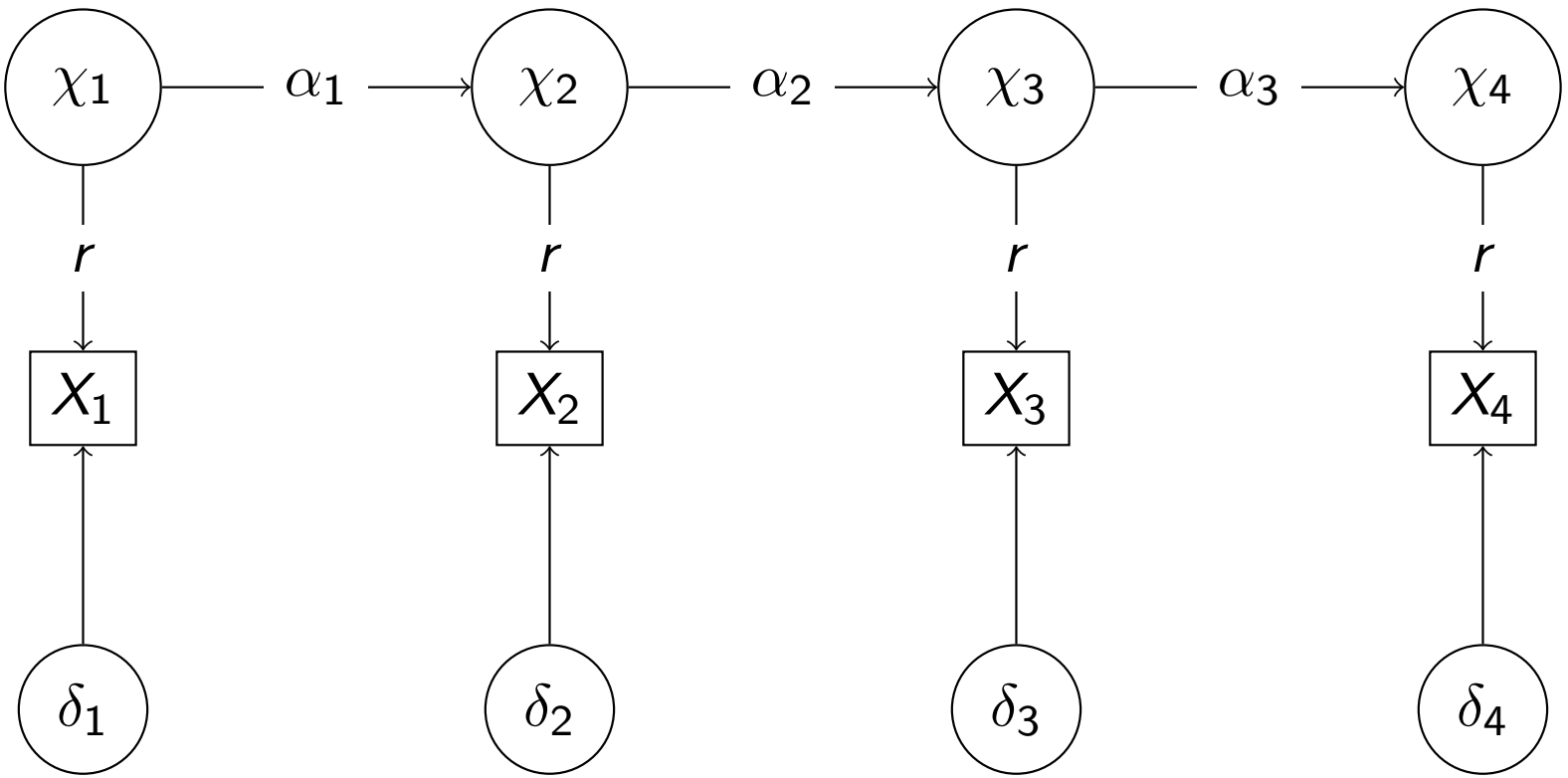
Model state change over 4 time points: simplex but with too many parameters

$\sqrt{r_i}$ is reliability, α_i is autocorrelation.



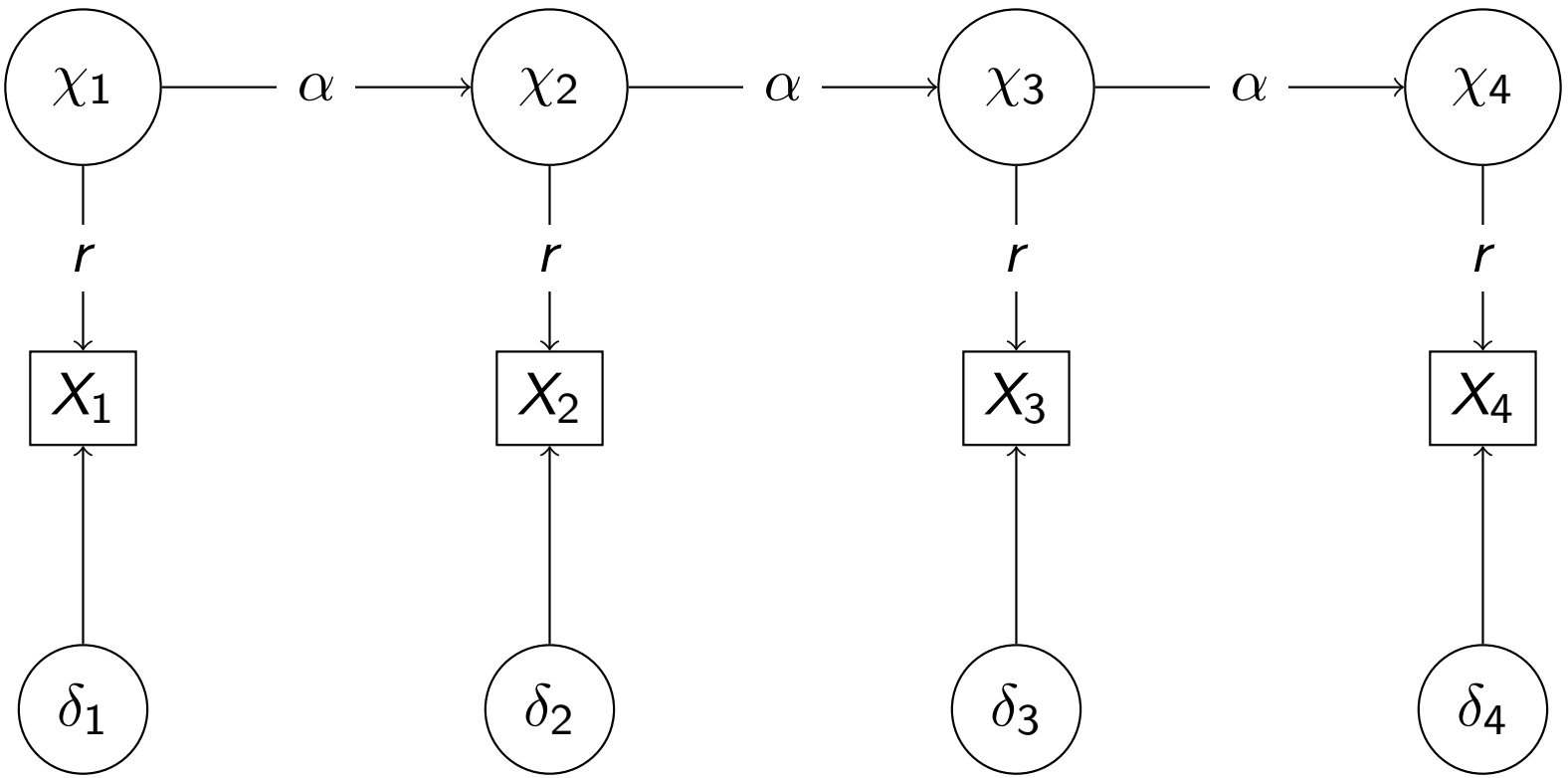
Model change over 4 time points: simplex: force some constraints

Equal reliabilities, unequal autocorrelation.



Model change over 4 time points: simplex: fully constrained: 2 parameters

But, even though there are 6 observed correlations, the model is unidentified, because $r_{x_1x_2} = r_{x_2x_3} ... = r^2\alpha$



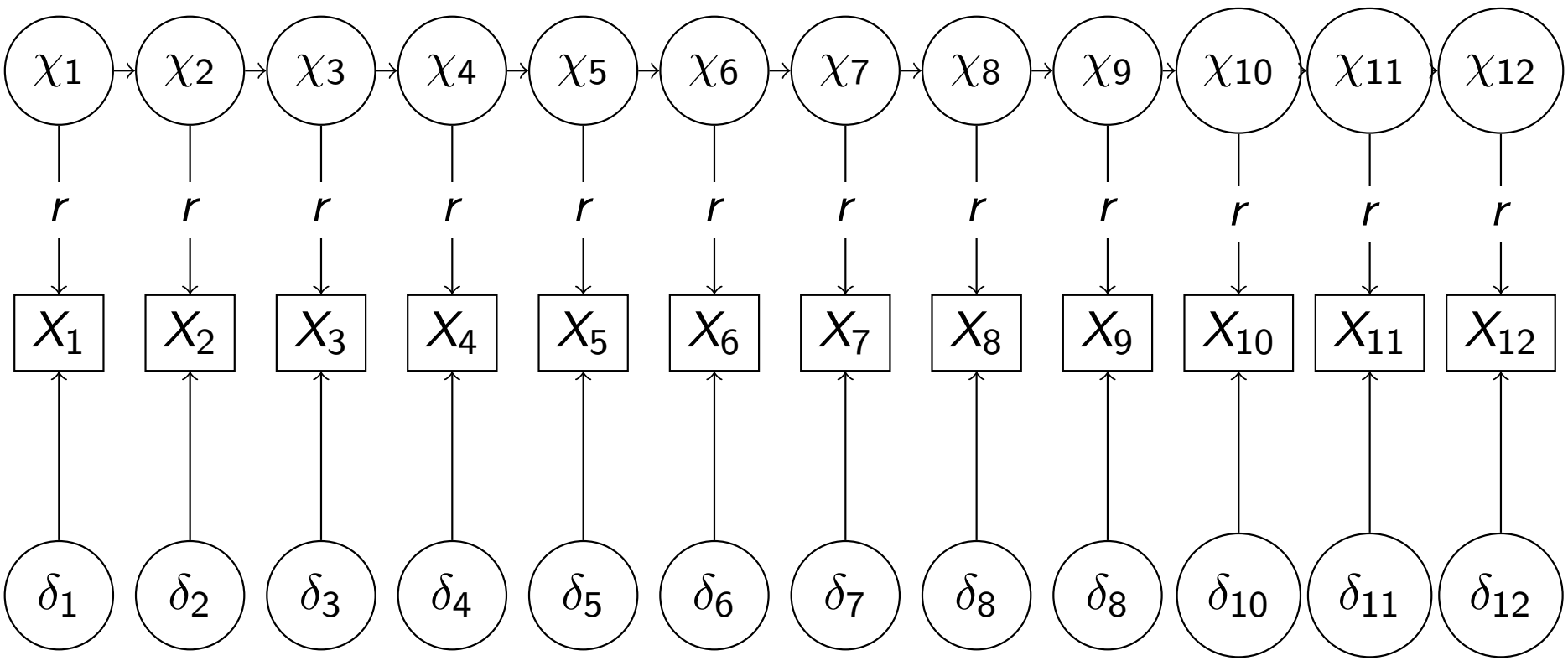
Generating a 4 variable simplex (autocorrelation confounded with reliability)

R code

```
simplex <- sim.simplex(nvar=4,alpha=.8)
simplex
```

```
      V1    V2    V3     V4
V1 1.000 0.80 0.64 0.512
V2 0.800 1.00 0.80 0.640
V3 0.640 0.80 1.00 0.800
V4 0.512 0.64 0.80 1.000
```

Model a 12 variable simplex



A 12 variable simplex:

Notice how the correlations will tend towards 0 because the further away in time, the smaller the effect of the autocorrelation. This is in some sense, a one factor model with all correlations varying as $(r\alpha)^t$ where t is the time between observations. However, this is not a congeneric model. In addition, note that the autocorrelations and reliabilities are confounded.

R code

```
simplex12 <- sim.simplex()    #default is 12 variables
```

```
lowerMat(simplex12)
     V1   V2   V3   V4   V5   V6   V7   V8   V9   V10  V11  V12
V1  1.00
V2  0.80 1.00
V3  0.64 0.80 1.00
V4  0.51 0.64 0.80 1.00
V5  0.41 0.51 0.64 0.80 1.00
V6  0.33 0.41 0.51 0.64 0.80 1.00
V7  0.26 0.33 0.41 0.51 0.64 0.80 1.00
V8  0.21 0.26 0.33 0.41 0.51 0.64 0.80 1.00
V9  0.17 0.21 0.26 0.33 0.41 0.51 0.64 0.80 1.00
V10 0.13 0.17 0.21 0.26 0.33 0.41 0.51 0.64 0.80 1.00
V11 0.11 0.13 0.17 0.21 0.26 0.33 0.41 0.51 0.64 0.80 1.00
V12 0.09 0.11 0.13 0.17 0.21 0.26 0.33 0.41 0.51 0.64 0.80 1.00
```

A naive factor analysis of a simplex

R code

```
> set.seed(42)
> simplex12 <- sim.simplex(n=1000)
> f1 <- fa(simplex12$observed)
> f1
```

```
Factor Analysis using method =  minres
Call: fa(r = simplex12$observed)
Standardized loadings (pattern matrix) based upon correlation matrix
    MR1   h2   u2 com
1  0.44 0.19 0.81   1
2  0.55 0.30 0.70   1
3  0.64 0.41 0.59   1
4  0.74 0.54 0.46   1
5  0.82 0.67 0.33   1
6  0.85 0.72 0.28   1
7  0.83 0.69 0.31   1
8  0.79 0.62 0.38   1
9  0.71 0.51 0.49   1
10 0.61 0.37 0.63   1
11 0.52 0.27 0.73   1
12 0.44 0.20 0.80   1

                MR1
SS loadings    5.49
Proportion Var 0.46

Mean item complexity =  1
Test of the hypothesis that 1 factor is sufficient.
```

one factor simplex – continued

```
The degrees of freedom for the null model are  66  and the objective function was  11.22
                with Chi Square of  11155.94
The degrees of freedom for the model are 54  and the objective function was  5.78

The root mean square of the residuals (RMSR) is  0.2
The df corrected root mean square of the residuals is  0.22

The harmonic number of observations is  1000 with the empirical chi square
                    5175.14  with prob <  0
The total number of observations was  1000  with MLE Chi Square =  5741.88  with prob <  0

Tucker Lewis Index of factoring reliability =  0.373
RMSEA index =  0.326  and the 90 % confidence intervals are  0.317 0.332
BIC =  5368.86
Fit based upon off diagonal values = 0.84
Measures of factor score adequacy
                                                 MR1
Correlation of scores with factors              0.96
Multiple R square of scores with factors        0.93
Minimum correlation of possible factor scores   0.86
```

Residuals of the 1 factor model are very large

This model could be interpreted as a one factor model, but not in the conventional sense. It is certainly not a congeneric model.

R code

```
resid(f1)
```

```
      C1    C2    C3    C4    C5    C6    C7    C8    C9   C10   C11   C12
R1  0.81
R2  0.56  0.70
R3  0.36  0.45  0.59
R4  0.17  0.23  0.33  0.46
R5  0.02  0.05  0.11  0.19  0.33
R6 -0.06 -0.05 -0.03  0.01  0.12  0.28
R7 -0.11 -0.12 -0.12 -0.10 -0.04  0.08  0.31
R8 -0.15 -0.16 -0.17 -0.16 -0.09 -0.01  0.15  0.38
R9 -0.14 -0.17 -0.19 -0.16 -0.13 -0.08  0.04  0.22  0.49
R10 -0.11 -0.13 -0.16 -0.15 -0.15 -0.12 -0.03  0.11  0.35  0.63
R11 -0.11 -0.14 -0.18 -0.17 -0.16 -0.11 -0.03  0.08  0.26  0.48  0.73
R12 -0.10 -0.10 -0.13 -0.14 -0.14 -0.11 -0.03  0.05  0.19  0.38  0.59  0.80
```

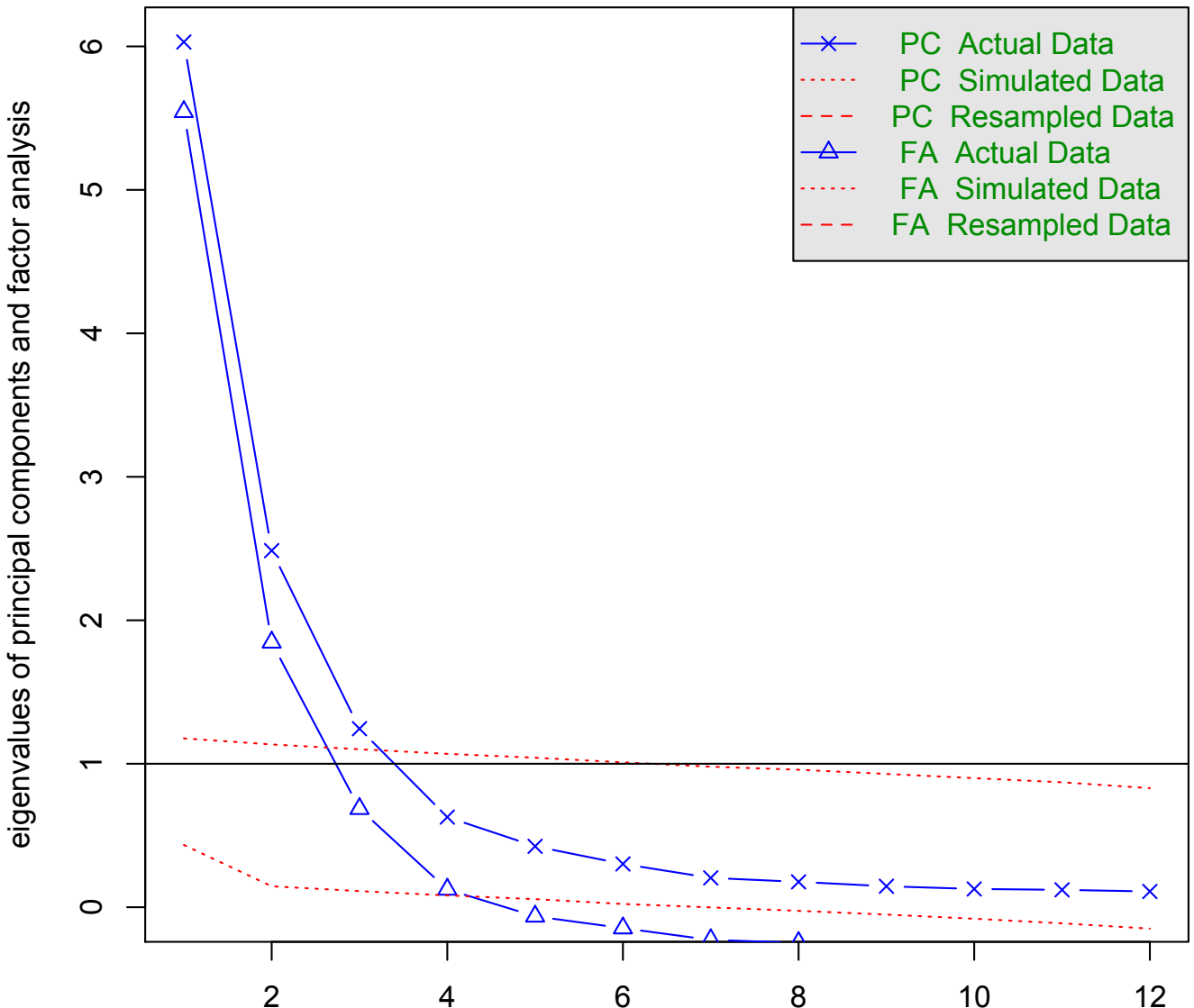
Number of factors by conventional methods:

R code

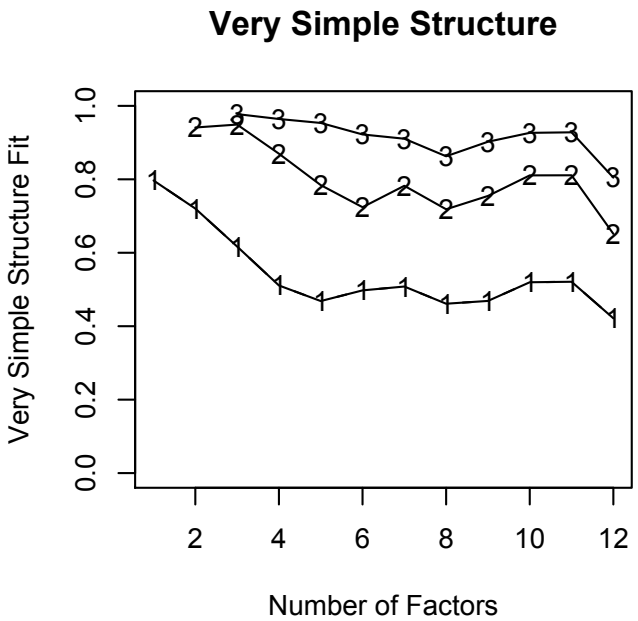
```
nfactors(simplex12$observed)
```

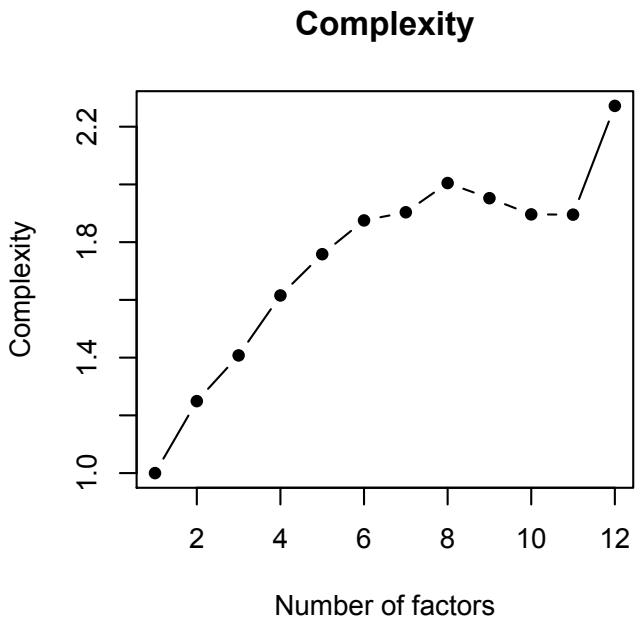
```
Number of factors
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
    n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximimum of 0.8  with  1  factors
VSS complexity 2 achieves a maximimum of 0.95  with  3  factors
The Velicer MAP achieves a minimum of 0.07  with  4  factors
Empirical BIC achieves a minimum of  -97.37  with  4  factors
Sample Size adjusted BIC achieves a minimum of  -25.44  with  6  factors

Statistics by number of factors
   vss1 vss2   map dof   chisq     prob sqresid  fit RMSEA  BIC  SABIC complex  eChisq    S
1  0.80 0.00 0.151  54 5.7e+03  0.0e+00   9.120 0.80  0.33 5369 5540.4     1.0 5.2e+03 2.0e
2  0.72 0.94 0.108  43 3.0e+03  0.0e+00   2.644 0.94  0.26 2728 2864.6     1.2 1.1e+03 9.0e
3  0.62 0.95 0.075  33 1.4e+03 3.4e-273   1.007 0.98  0.20 1171 1276.2     1.4 2.2e+02 4.1e
4  0.51 0.87 0.074  24 6.3e+02 1.4e-118   0.608 0.99  0.16  469  544.9     1.6 6.8e+01 2.3e
5  0.47 0.78 0.084  16 2.1e+02  2.4e-35   0.459 0.99  0.11   97  147.8     1.8 1.5e+01 1.1e
6  0.50 0.72 0.103   9 8.1e+00  5.2e-01   0.408 0.99  0.00  -54  -25.4     1.9 1.1e+00 2.9e
7  0.51 0.78 0.134   3 5.7e+00  1.3e-01   0.333 0.99  0.03  -15   -5.5     1.9 7.1e-01 2.3e
8  0.46 0.72 0.188  -2 2.5e-01       NA   0.216 1.00    NA   NA     NA     2.0 2.1e-02 4.0e
9  0.47 0.75 0.282  -6 8.3e-04       NA   0.141 1.00    NA   NA     NA     2.0 3.9e-05 1.7e
10 0.52 0.81 0.474  -9 7.0e-07       NA   0.194 1.00    NA   NA     NA     1.9 3.0e-08 4.8e
11 0.52 0.81 1.000 -11 0.0e+00       NA   0.192 1.00    NA   NA     NA     1.9 2.0e-15 1.2e
12 0.42 0.65    NA -12 1.8e-12       NA   0.013 1.00    NA   NA     NA     2.3 3.5e-25 1.6e
>
```

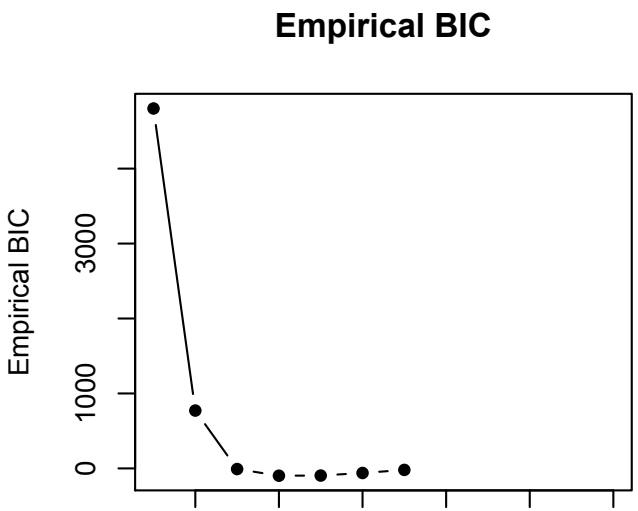
parallel analysis for the 12 item simplex

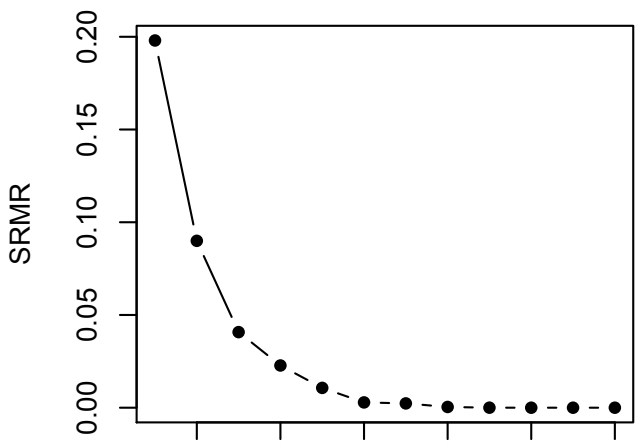


number of factors for the 12 item simplex









Lets model this with lavaan

R code

```
library(lavaan)
modsimp <- ' x2 ~  a*x1
             x3 ~  a*x2
             x4 ~  a*x3
             x5 ~  a* x4
             x6 ~  a*x5
             x7 ~  a*x6
             x8 ~  a* x7
             x9 ~  a*x8
             x10 ~ a*x9
             x11~  a* x10
             x12~  a* x11'
fitsimp <- sem(model=modsimp,data=simplex12$observed)
summary(fitsimp)
```

```
summary(fitsimp)
lavaan (0.5-19) converged normally after  16 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               64.004
  Degrees of freedom                                65
  P-value (Chi-square)                           0.512
```

lavaan simplex output continued: slopes are correct

```
Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  x2 ~
    x1         (a)    0.796    0.006  140.017    0.000
  x3 ~
    x2         (a)    0.796    0.006  140.017    0.000
  x4 ~
   ...
  x11 ~
    x10        (a)    0.796    0.006  140.017    0.000
  x12 ~
    x11        (a)    0.796    0.006  140.017    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x2                0.390    0.017   22.361    0.000
    x3                0.356    0.016   22.361    0.000
    x4                0.342    0.015   22.361    0.000
  ...
    x11               0.344    0.015   22.361    0.000
    x12               0.335    0.015   22.361    0.000
```

lavaan simplex output continued: residuals are tiny

R code

```
resid(fitsimp)
```

```
resid(fitsimp)
$type
[1] "raw"

$cov
    x2     x3     x4     x5     x6     x7     x8     x9     x10    x11    x12    x1
x2   0.013
x3  -0.023 -0.045
x4  -0.028 -0.044 -0.041
x5  -0.020 -0.026 -0.018 -0.002
x6   0.005 -0.014 -0.010  0.009  0.015
x7   0.003 -0.010 -0.006  0.003 -0.001 -0.011
x8   0.014  0.002  0.008  0.039  0.027  0.002  0.009
x9   0.020  0.005  0.034  0.052  0.036  0.005  0.002 -0.003
x10  0.042  0.023  0.043  0.035  0.012 -0.015 -0.022 -0.002 -0.002
x11  0.009 -0.019 -0.003  0.002  0.004 -0.013 -0.027 -0.033 -0.040 -0.062
x12  0.037  0.017  0.021  0.022  0.016  0.021  0.010  0.001  0.009  0.003  0.044
x1   0.008 -0.020 -0.041 -0.035 -0.024 -0.012 -0.017  0.006  0.029  0.002  0.011  0.000

$mean
 x2  x3  x4  x5  x6  x7  x8  x9 x10 x11 x12  x1
  0   0   0   0   0   0   0   0   0   0   0   0
```

Now lets examine change in the means of a simplex model

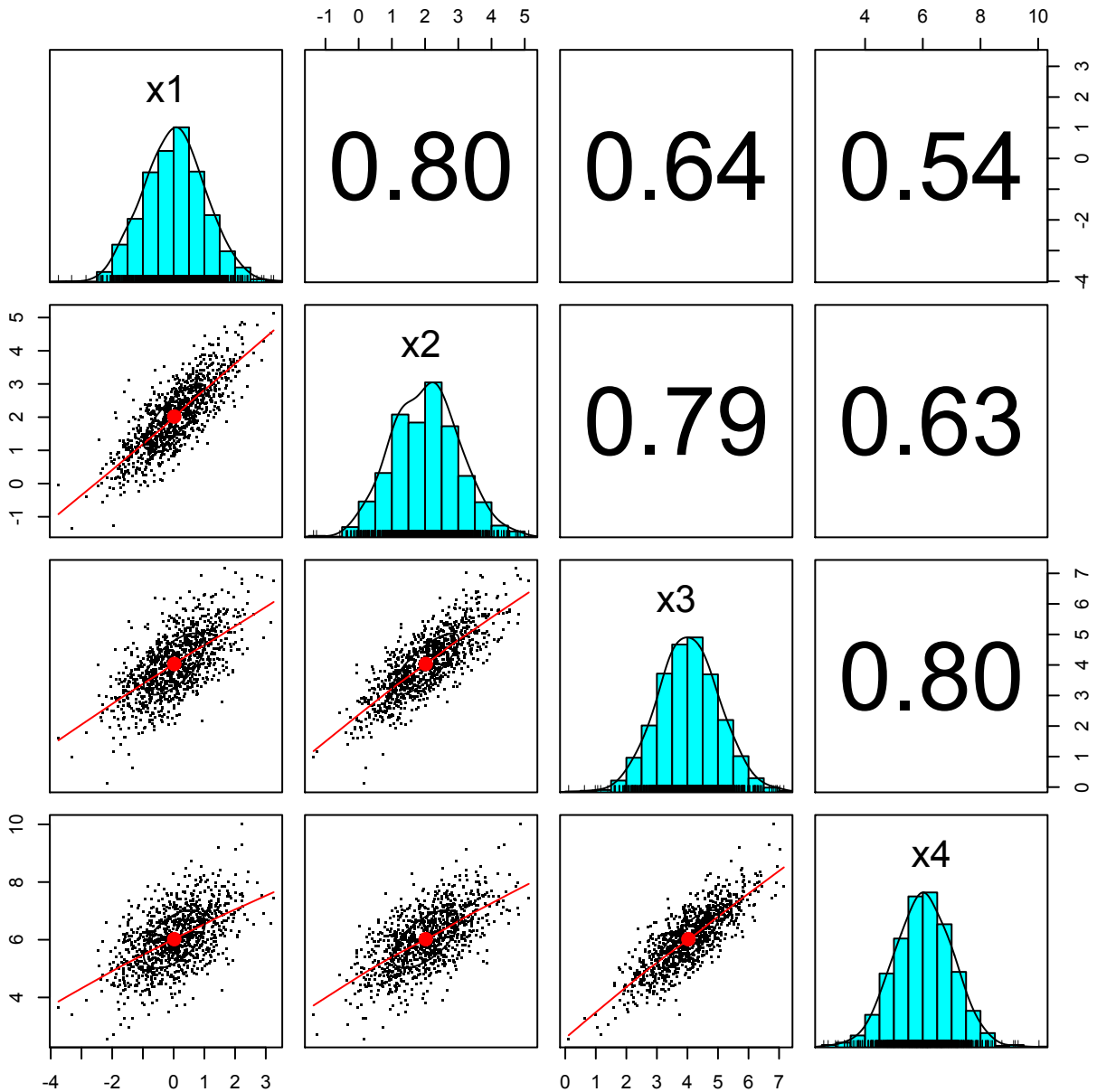
Put in a change in the means in our simplex.

R code

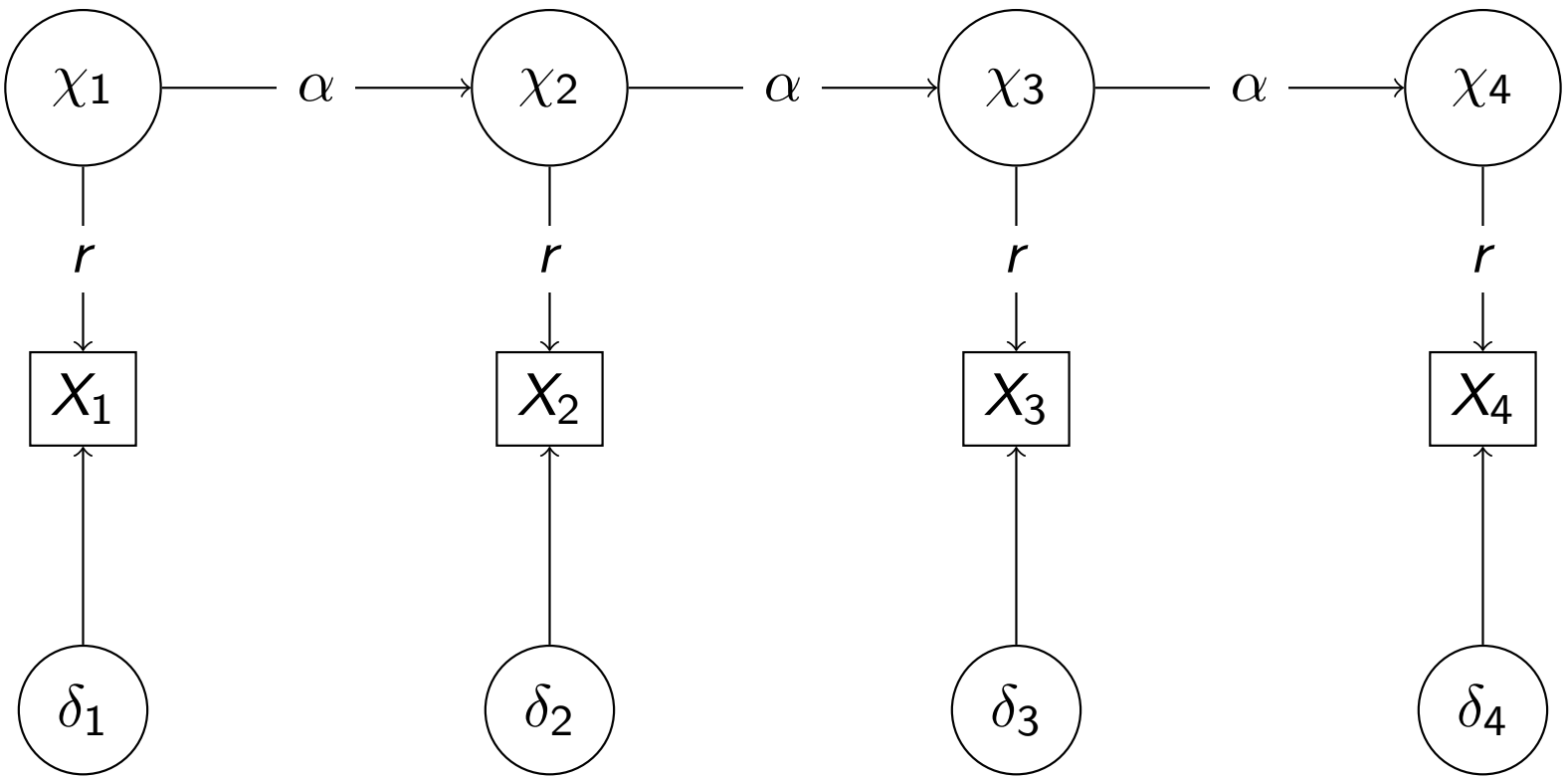
```
set.seed(42)
simp4 <- sim.simplex(4,mu=c(0,2,4,6),n=1000)
colnames(simp4$observed) <- paste0("x",1:4)
describe(simp4$observed)
```

```
   vars    n mean   sd median trimmed  mad   min   max range  skew kurtosis   se
x1    1 1000 0.02 1.00   0.03    0.02 1.02 -3.75  3.26  7.01  0.01     0.05 0.03
x2    2 1000 2.02 1.00   2.05    2.01 1.03 -1.36  5.12  6.48  0.07    -0.08 0.03
x3    3 1000 4.03 1.00   4.02    4.03 0.98  0.11  7.16  7.05 -0.01     0.24 0.03
x4    4 1000 6.02 1.01   6.02    6.02 0.99  2.57 10.03  7.46 -0.01     0.28 0.03
```

The simplex structure is a state auto correlation structure



Model change over 4 time points: states are auto correlated



Model it using lavaan growth, with a state model: 0 df!

R code

```
mod.auto =  ' X1 =~ r*x1
              X2  =~ r* x2
              X3 =~ r * x3
              X4 =~ r* x4
              X2 =~  X1
              X3 =~ X2
              X4 =~ X3'


fit.auto <- growth(mod.auto,data= simp4$observed)
fit.auto
```

```
Warning messages:
1: In lavData(data = data, group = group, group.label = group.label,  :
  lavaan WARNING: data argument has been coerced to a data.frame
2: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING: could not compute standard errors!
  lavaan NOTE: this may be a symptom that the model is not identified.

> fit.auto
lavaan (0.5-19) converged normally after  29 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               0.811
  Degrees of freedom                                0
  Minimum Function Value                0.0004057403763
```

Model it using lavaan growth, with a state model: 2 df!

R code

```
mod.auto =  ' X1 =~ r*x1
              X2  =~ r* x2
              X3 =~ r * x3
              X4 =~ r* x4
              X2 =~  a * X1
              X3 =~  a* X2
              X4 =~  .8*X3'
fit.auto <- growth(mod.auto,data= simp4$observed)
summary(fit.auto)
```

```
lavaan (0.5-19) converged normally after  29 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               1.411
  Degrees of freedom                                2
  P-value (Chi-square)                          0.494

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  X1 =~
    x1         (r)    1.000
```

auto correlated 4 variable simplex

Latent Variables:

		Estimate	Std.Err	Z-value	P(>|z|)
X1 =~					
x1	(r)	1.000			
X2 =~					
x2	(r)	1.000			
X3 =~					
x3	(r)	1.000			
X4 =~					
x4	(r)	1.000			
X1 =~					
X2	(alph)	0.922	0.029	32.185	0.000
X2 =~					
X3	(alph)	0.922	0.029	32.185	0.000
X3 =~					
X4	(alph)	0.922	0.029	32.185	0.000

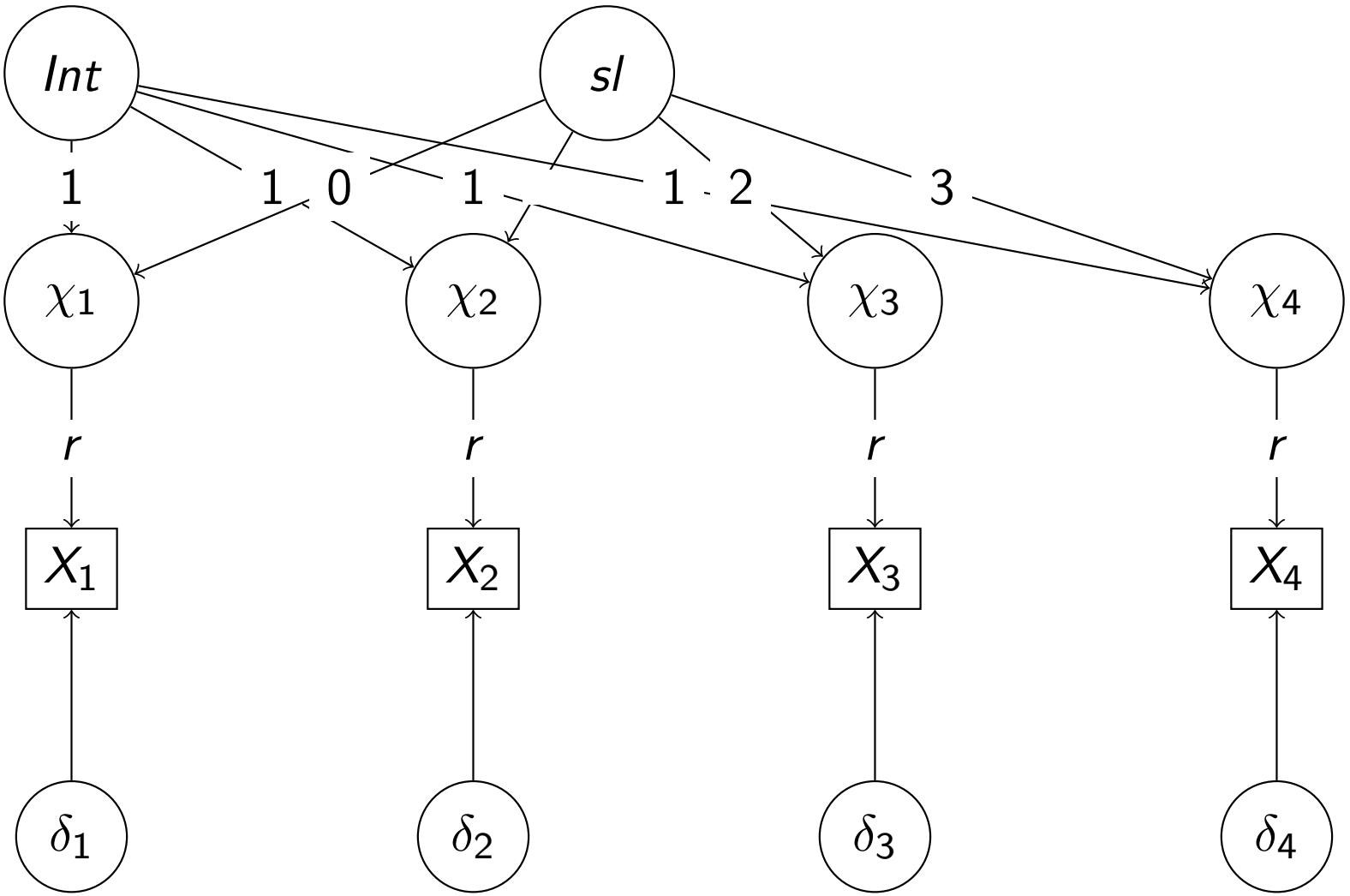
Intercepts:

	Estimate	Std.Err	Z-value	P(>|z|)
x1	0.000			
x2	0.000			
x3	0.000			
x4	0.000			
X1	0.056	0.031	1.778	0.075
X2	-0.028	0.029	-0.967	0.334
X3	-0.030	0.032	-0.935	0.350
X4	1.623	0.033	49.235	0.000

Variances:

	Estimate	Std.Err	Z-value	P(>|z|)
x1	0.376	0.044	8.561	0.000
x2	0.484	0.031	15.727	0.000
x3	0.610	0.037	16.539	0.000

Model change over 4 time points: traits are stable



Model it using lavaan growth, but this is not the simplex model!

R code

```
mod.growth =  ' i =~ 1*x1 + 1*x2 + 1*x3 + 1* x4
                s =~ 0*x1 + 1*x2 + 2 *x3  + 3*x4'
fit <- growth(mod.growth,data= simp4$observed)
```

```
lavaan (0.5-19) converged normally after  43 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic             138.565
  Degrees of freedom                                5
  P-value (Chi-square)                          0.000

Parameter Estimates:
```

lavaan growth model continued

```
Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  i =~
    x1                1.000
    x2                1.000
    x3                1.000
    x4                1.000
  s =~
    x1                0.000
    x2                1.000
    x3                2.000
    x4                3.000

Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)
  i ~~
    s                -0.104    0.012   -8.380    0.000

Intercepts:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    i                 0.021    0.032    0.648    0.517
    s                 2.001    0.010  193.227    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1                0.178    0.020    9.000    0.000
    x2                0.184    0.012   15.481    0.000
    x3                0.175    0.012   15.065    0.000
    x4                0.183    0.020    9.229    0.000
```

lavaan residuals

R code

```
resid(fit)
```

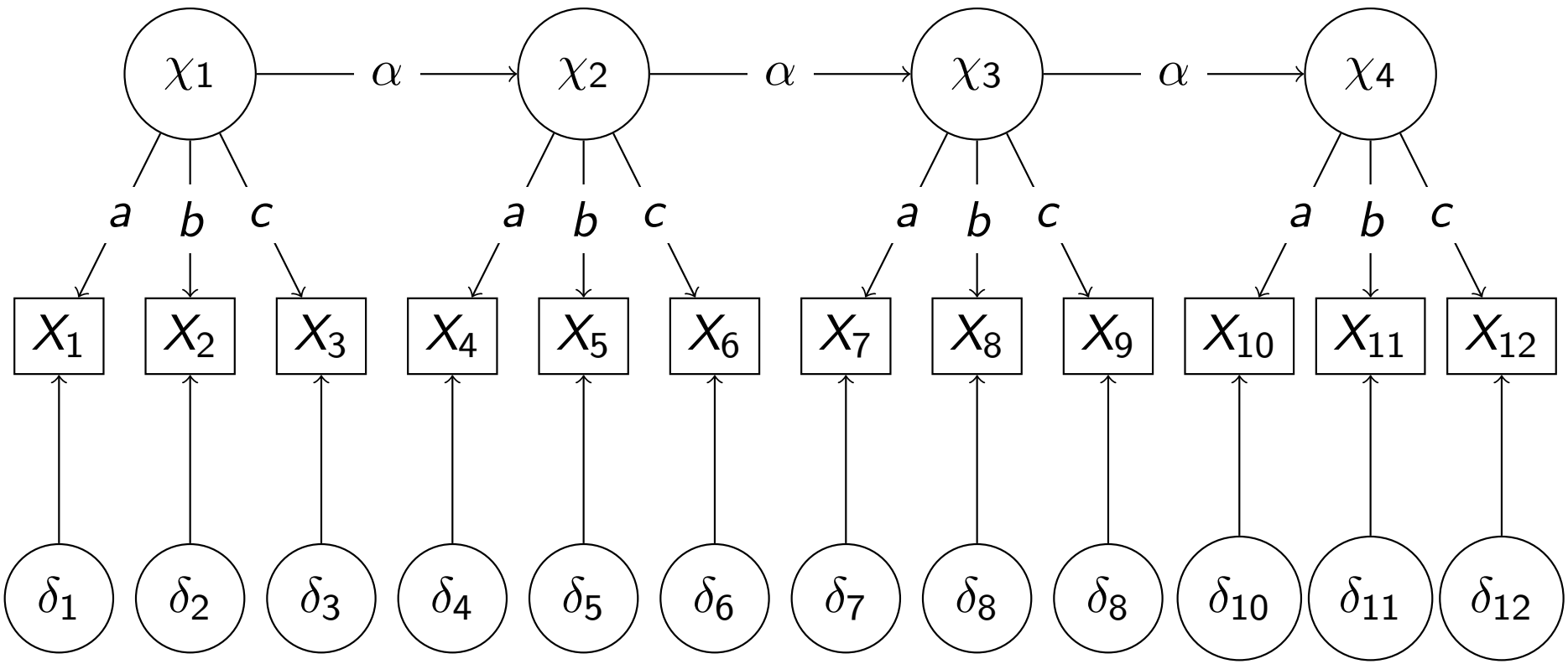
```
$type
[1] "raw"

$cov
   x1      x2      x3      x4
x1 -0.067
x2  0.016  0.063
x3 -0.033  0.079  0.078
x4 -0.033 -0.050  0.019 -0.076

$mean
    x1      x2      x3      x4
 0.000 -0.004  0.007 -0.004
```

Try a change model with autocorrelations

R code

```
mod.auto.change =  '
             X1 =~ r * x1
             X2 =~  r*x2
             X3 =~  r *  x3
              X4 =~  r * x4
            i =~ 1*X1 + 1*X2 + 1*X3 + 1* X4
                s =~ 0*X1 + 1*X2 + 2 *X3  + 3*X4'
fit.auto.change<- growth(mod.auto.change,data= simp4$observed)
fit.auto.change
resid(fit.auto)
```

Model change over 4 time points: states with measurement



Create four factor-simplex data

R code

```
set.seed(42)
#explicitly stating some of the default values
simp4 <- sim(mu=c(0,2,4,6),n=1000,alpha=.8,lambda=0)
colnames(simp4$observed) <- paste0("x",1:12)
describe(simp4$observed)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
x1	1	1000	0.06	0.99	0.02	0.06	0.98	-3.52	3.30	6.82	0.00	0.07	0.03
x2	2	1000	0.02	1.02	0.03	0.02	1.05	-3.77	3.13	6.90	-0.05	0.16	0.03
x3	3	1000	-0.01	1.02	0.01	0.00	1.01	-3.49	3.23	6.72	-0.08	0.10	0.03
x4	4	1000	1.62	0.99	1.58	1.60	1.01	-2.15	5.18	7.33	0.10	0.03	0.03
x5	5	1000	1.39	1.02	1.40	1.39	1.01	-1.89	4.70	6.59	-0.04	0.09	0.03
x6	6	1000	1.21	0.98	1.19	1.20	0.97	-2.45	4.43	6.88	0.08	0.10	0.03
x7	7	1000	3.24	1.00	3.25	3.25	0.96	-0.32	6.63	6.95	-0.07	0.20	0.03
x8	8	1000	2.84	0.96	2.85	2.84	1.00	-0.14	5.80	5.93	-0.03	-0.25	0.03
x9	9	1000	2.41	1.02	2.39	2.41	1.01	-1.19	6.13	7.32	0.02	0.13	0.03
x10	10	1000	4.81	1.03	4.79	4.80	1.07	1.76	7.97	6.21	0.05	-0.27	0.03
x11	11	1000	4.20	1.00	4.15	4.19	1.00	0.64	7.18	6.54	0.08	-0.05	0.03
x12	12	1000	3.61	1.00	3.64	3.62	0.99	0.57	6.61	6.05	-0.07	-0.03	0.03

Create four factor-simplex data

R code

```
lowerCor(simp4$observed)
```

```
     x1   x2   x3   x4   x5   x6   x7   x8   x9   x10  x11  x12
x1   1.00
x2   0.57 1.00
x3   0.47 0.46 1.00
x4   0.50 0.46 0.42 1.00
x5   0.44 0.40 0.35 0.59 1.00
x6   0.32 0.29 0.26 0.47 0.42 1.00
x7   0.41 0.38 0.31 0.50 0.43 0.39 1.00
x8   0.33 0.32 0.26 0.44 0.41 0.31 0.53 1.00
x9   0.32 0.28 0.22 0.40 0.39 0.32 0.47 0.38 1.00
x10 0.33 0.31 0.29 0.41 0.38 0.30 0.51 0.43 0.38 1.00
x11 0.26 0.26 0.21 0.38 0.36 0.28 0.45 0.40 0.34 0.54 1.00
x12 0.23 0.24 0.18 0.33 0.26 0.20 0.37 0.25 0.29 0.51 0.37 1.00
```

How many factors are in the data?

R code

```
nfactors(simp4$observed)
```

```
Number of factors
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
    n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximimum of 0.83  with  1  factors
VSS complexity 2 achieves a maximimum of 0.87  with  2  factors
The Velicer MAP achieves a minimum of 0.02  with  1  factors
Empirical BIC achieves a minimum of  -176.84  with  3  factors
Sample Size adjusted BIC achieves a minimum of  -61.46  with  4  factors

Statistics by number of factors
   vss1 vss2   map dof   chisq    prob sqresid  fit RMSEA  BIC SABIC complex  eChisq    SRM
1  0.83 0.00 0.022  54 6.2e+02 6.2e-97     5.4 0.83 0.103  246   417     1.0 6.1e+02 6.8e-0
2  0.56 0.87 0.024  43 2.2e+02 2.5e-26     4.0 0.87 0.065  -73    64     1.5 1.5e+02 3.4e-0
3  0.44 0.78 0.035  33 8.7e+01 8.2e-07     3.5 0.89 0.041 -141   -36     1.8 5.1e+01 2.0e-0
4  0.38 0.62 0.052  24 2.8e+01 2.6e-01     3.2 0.90 0.013 -138   -61     2.0 1.6e+01 1.1e-0
5  0.36 0.61 0.079  16 9.5e+00 8.9e-01     2.8 0.91 0.000 -101   -50     2.1 5.3e+00 6.4e-0
6  0.36 0.63 0.112   9 3.8e+00 9.2e-01     2.6 0.92 0.000  -58   -30     2.1 2.1e+00 3.9e-0
7  0.39 0.67 0.162   3 7.4e-01 8.6e-01     2.5 0.92 0.000  -20   -10     2.1 4.0e-01 1.7e-0
8  0.37 0.59 0.202  -2 9.8e-06      NA     2.4 0.92    NA   NA    NA     2.4 5.0e-06 6.2e-0
9  0.35 0.62 0.287  -6 3.3e-06      NA     2.6 0.92    NA   NA    NA     2.5 1.9e-06 3.8e-0
10 0.40 0.61 0.461  -9 6.2e-08      NA     2.5 0.92    NA   NA    NA     2.3 3.6e-08 5.2e-0
11 0.40 0.61 1.000 -11 2.1e-10      NA     2.4 0.92    NA   NA    NA     2.3 1.4e-10 3.2e-0
12 0.40 0.61    NA -12 2.0e-10      NA     2.4 0.92    NA   NA    NA     2.3 1.4e-10 3.2e-0
```

Extract 4 factors (based upon theory?)

R code

```
f4 <- fa(simp4$observed,4)
f4
```

```
Factor Analysis using method =  minres
Call: fa(r = simp4$observed, nfactors = 4)
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1   MR3   MR2   MR4   h2   u2 com
x1   0.05  0.70 -0.03  0.06 0.59 0.41 1.0
x2  -0.04  0.75  0.01  0.03 0.56 0.44 1.0
x3   0.11  0.56  0.06 -0.09 0.38 0.62 1.2
x4   0.77  0.07  0.02 -0.02 0.67 0.33 1.0
x5   0.66  0.03  0.04  0.01 0.52 0.48 1.0
x6   0.56 -0.08 -0.04  0.13 0.34 0.66 1.1
x7  -0.03  0.07  0.05  0.74 0.63 0.37 1.0
x8   0.09  0.00  0.00  0.60 0.45 0.55 1.0
x9   0.16 -0.01  0.03  0.45 0.35 0.65 1.3
x10 -0.03  0.02  0.85  0.01 0.73 0.27 1.0
x11  0.11 -0.08  0.47  0.19 0.43 0.57 1.5
x12  0.06  0.00  0.60 -0.05 0.36 0.64 1.0
                       MR1  MR3  MR2  MR4
SS loadings           1.66 1.48 1.46 1.41
Proportion Var        0.14 0.12 0.12 0.12
Cumulative Var        0.14 0.26 0.38 0.50
Proportion Explained  0.28 0.25 0.24 0.23
Cumulative Proportion 0.28 0.52 0.77 1.00
 With factor correlations of
     MR1  MR3  MR2  MR4
MR1 1.00 0.73 0.58 0.74
MR3 0.73 1.00 0.46 0.57
MR2 0.58 0.46 1.00 0.73
MR4 0.74 0.57 0.73 1.00
```

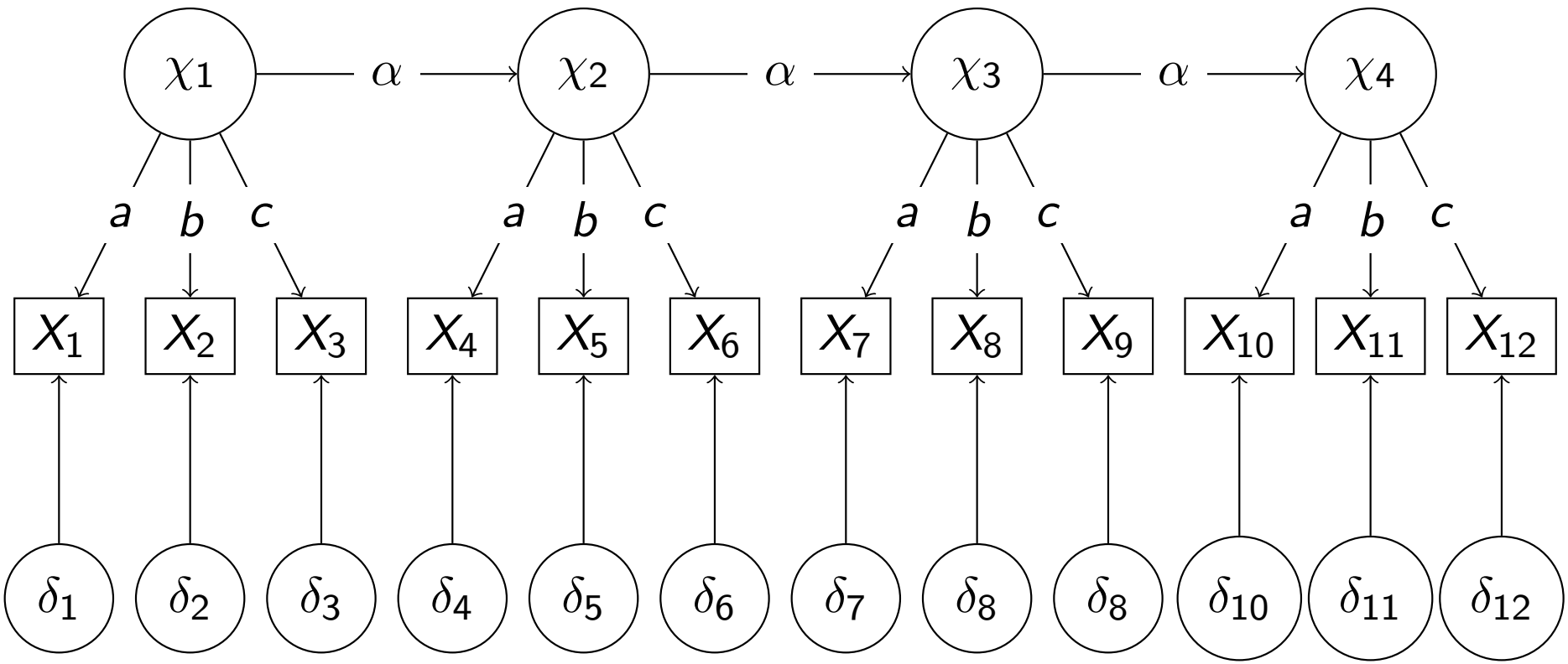
Reorganize to show the temporal structure

R code

```
f4or <- fa.organize(f4,o=c(2,1,4,3))  #order the factors from
f4or                                  # f4  in the specified order
```

```
Factor Analysis using method =  minres
Call: fa(r = simp4$observed, nfactors = 4)
Standardized loadings (pattern matrix) based upon correlation matrix
      MR3   MR1   MR4   MR2   h2   u2 com
x1   0.70  0.05  0.06 -0.03 0.59 0.41 1.0
x2   0.75 -0.04  0.03  0.01 0.56 0.44 1.0
x3   0.56  0.11 -0.09  0.06 0.38 0.62 1.2
x4   0.07  0.77 -0.02  0.02 0.67 0.33 1.0
x5   0.03  0.66  0.01  0.04 0.52 0.48 1.0
x6  -0.08  0.56  0.13 -0.04 0.34 0.66 1.1
x7   0.07 -0.03  0.74  0.05 0.63 0.37 1.0
x8   0.00  0.09  0.60  0.00 0.45 0.55 1.0
x9  -0.01  0.16  0.45  0.03 0.35 0.65 1.3
x10  0.02 -0.03  0.01  0.85 0.73 0.27 1.0
x11 -0.08  0.11  0.19  0.47 0.43 0.57 1.5
x12  0.00  0.06 -0.05  0.60 0.36 0.64 1.0
                       MR3  MR1  MR4  MR2
SS loadings           1.48 1.66 1.41 1.46
Proportion Var        0.12 0.14 0.12 0.12
Cumulative Var        0.12 0.26 0.38 0.50
Proportion Explained  0.25 0.28 0.23 0.24
Cumulative Proportion 0.25 0.52 0.76 1.00
 With factor correlations of
     MR3  MR1  MR4  MR2
MR3 1.00 0.73 0.57 0.46
MR1 0.73 1.00 0.74 0.58
MR4 0.57 0.74 1.00 0.73
MR2 0.46 0.58 0.73 1.00
```

Model change over 4 time points: states with measurement



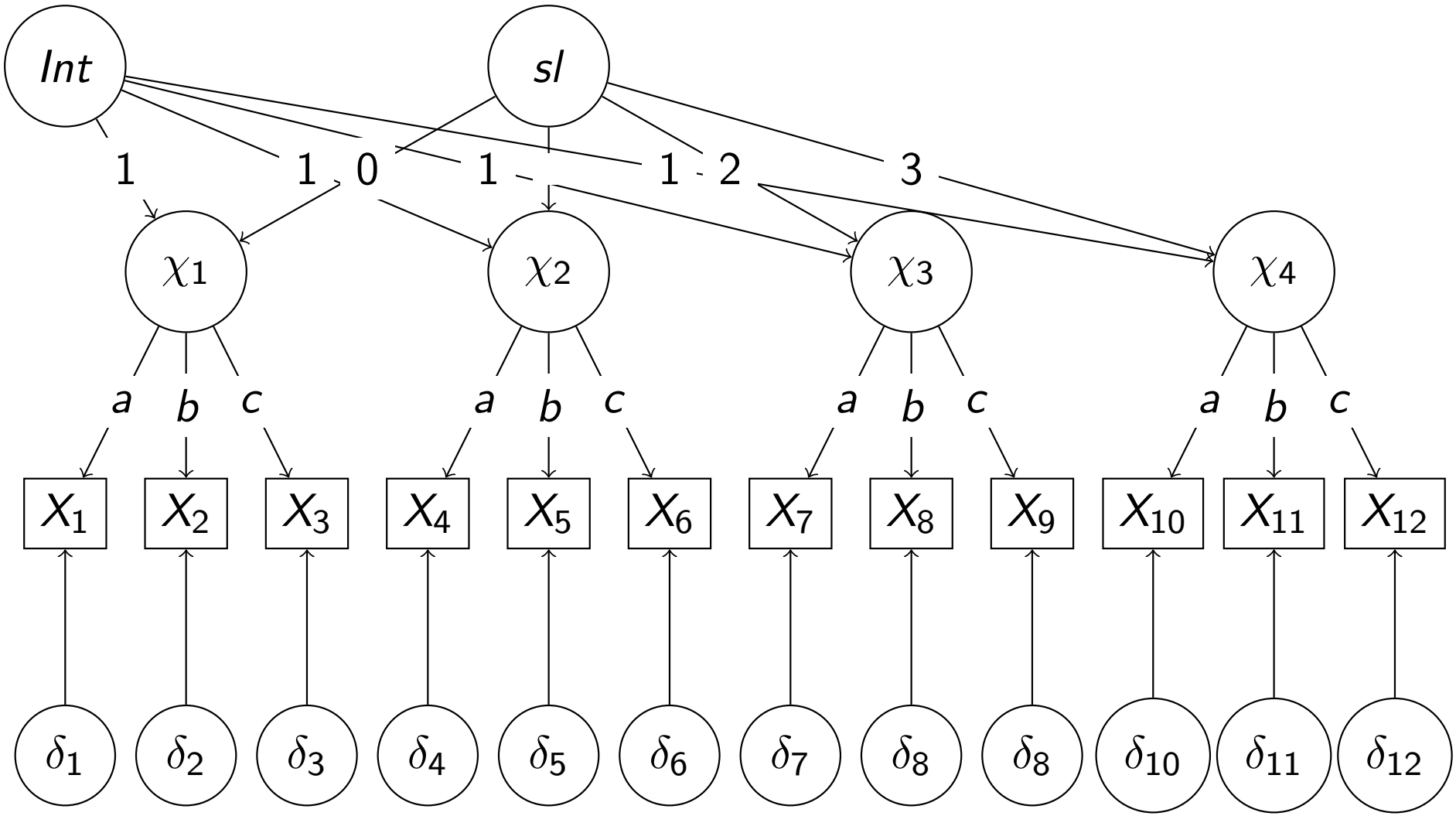
State growth

R code

```
models <- 'L1 =~ a*x1 +b *x2 + c* x3
           L2 =~ a* x4 + b * x5 + c* x6
           L3 =~ a*x7 + b*x8 +c*x9
           L4 =~ a*x10 + b*x11 + c*x12

           L2 ~ alpha* L1
           L3 ~ alpha * L2
           L4 ~ alpha *  L3'

fits <- growth(models,data=simp4$observed)
summary(fits)
```

```
lavaan (0.5-19) converged normally after  32 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               84.661
  Degrees of freedom                                67
  P-value (Chi-square)                           0.071

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
```

Model change over 4 time points: traits with measurement



Now, lets try a lavaan trait growth model

R code

```
model <- 'L1 =~ x1 + x2 + x3
          L2 =~ x4 + x5 + x6
          L3 =~ x7 + x8 +x9
          L4 =~ x10 + x11 + x12
          i =~ 1* L1 + 1 * L2 + 1 * L3 + 1 * L4
          s =~ 0 * L1 + 1 * L2 + 2* L3 + 3* L4'
fit <- growth(model,data=simp4$observed)
summary(fit)
```

```
Warning messages:
1: In lavData(data = data, group = group, group.label = group.label,  :
  lavaan WARNING: data argument has been coerced to a data.frame
2: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING: could not compute standard errors!
  lavaan NOTE: this may be a symptom that the model is not identified.

> summary(fit)
lavaan (0.5-19) converged normally after  42 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic              94.775
  Degrees of freedom                               57
  P-value (Chi-square)                          0.001

Parameter Estimates:
```

Try again with a lavaan growth model with some equality constraints

R code

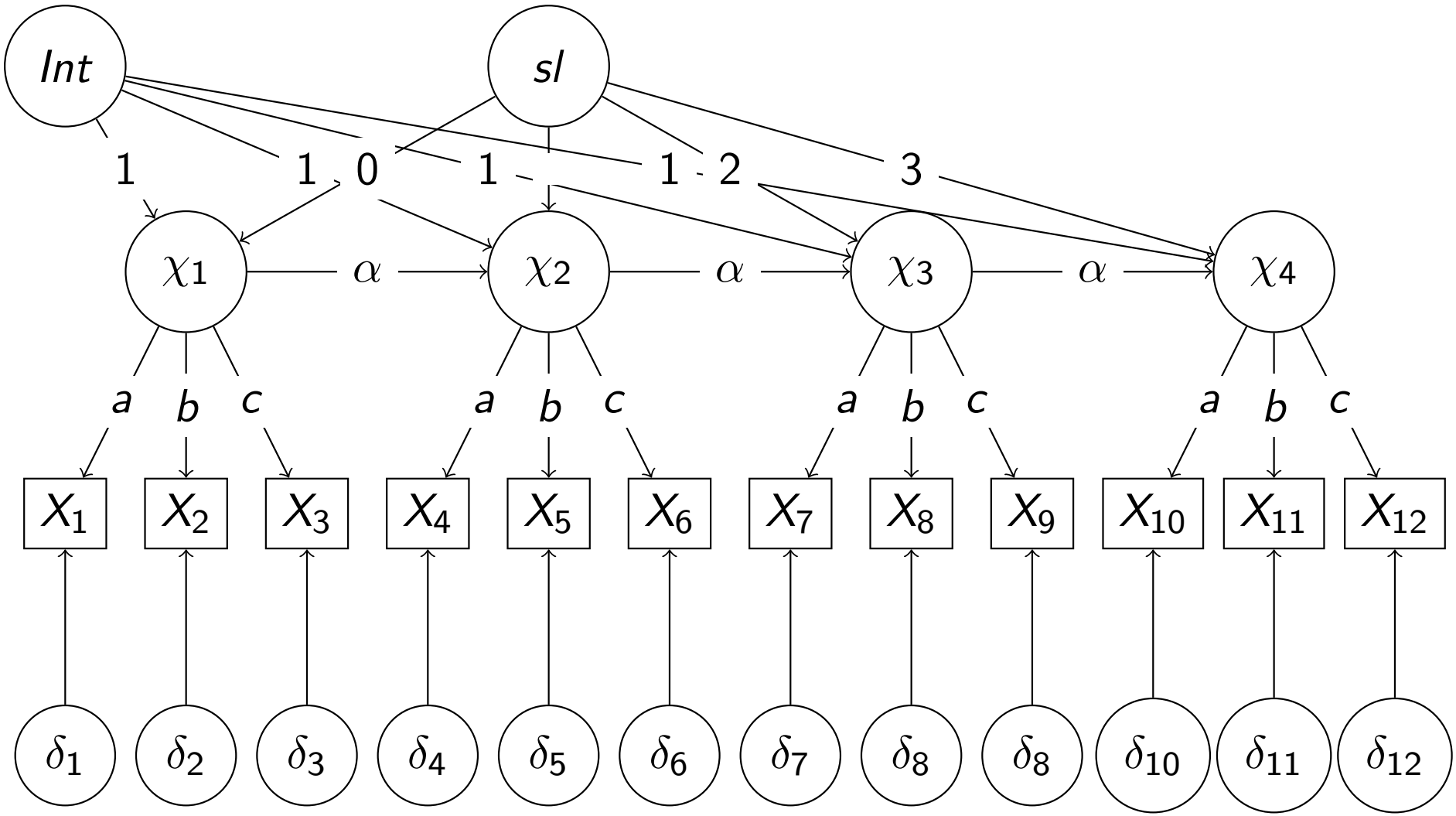
```
modela <- 'L1 =~ a*x1 +b *x2 + c* x3
           L2 =~ a* x4 + b * x5 + c* x6
           L3 =~ a*x7 + b*x8 +c*x9
           L4 =~ a*x10 + b*x11 + c*x12
           i =~ 1* L1 + 1 * L2 + 1 * L3 + 1 * L4
           s =~ 0 * L1 + 1 * L2 + 2* L3 + 3* L4'
fita <- growth(modela,data=simp4$observed)
summary(fita)
```

```
lavaan (0.5-19) converged normally after  40 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               96.397
  Degrees of freedom                                63
  P-value (Chi-square)                           0.004

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  L1 =~
    x1         (a)    1.000
    x2         (b)    0.873    0.005  189.783    0.000
```

lavaan continued

```
Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)
  i ~~
    s                -0.072    0.012   -5.775    0.000

Intercepts:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
    x8                0.000
    x9                0.000
    x10               0.000
    x11               0.000
    x12               0.000
    L1               -0.228    0.015  -15.073    0.000
    L2               -0.051    0.016   -3.125    0.002
    L3                0.184    0.015   12.534    0.000
    L4                0.360    0.010   34.753    0.000
    i                 0.264    0.016   16.046    0.000
    s                 1.396    0.008  170.806    0.000
```

Model change over 4 time points: trait-state model



Try a lavaan growth model with trait and state effects

R code

```
modelb <- 'L1 =~ a*x1 +b *x2 + c* x3
           L2 =~ a* x4 + b * x5 + c* x6
           L3 =~ a*x7 + b*x8 +c*x9
           L4 =~ a*x10 + b*x11 + c*x12
           i =~ 1* L1 + 1 * L2 + 1 * L3 + 1 * L4
           s =~ 0 * L1 + 1 * L2 + 2* L3 + 3* L4
           L2 =~ L1
           L3 =~ L2
           L4 =~ L3'

fitb <- growth(modelb,data=simp4$observed)
summary(fitb)
```

```
  lavaan (0.5-19) converged normally after  65 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               81.949
  Degrees of freedom                                60
  P-value (Chi-square)                           0.031

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard
```

Trait State continued

```
Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  L1 =~
    x1          (a)    1.000
    x2          (b)    0.873    0.005  189.728    0.000
    x3          (c)    0.750    0.005  156.822    0.000
 ...
   L4 =~
    x10         (a)    1.000
    x11         (b)    0.873    0.005  189.728    0.000
    x12         (c)    0.750    0.005  156.822    0.000
  i =~
    L1                 1.000
    L2                 1.000
    L3                 1.000
    L4                 1.000
  s =~
    L1                 0.000
    L2                 1.000
    L3                 2.000
    L4                 3.000
  L2 =~
    L1                 0.532    0.356    1.495    0.135
  L3 =~
    L2                 0.466    0.285    1.634    0.102
  L4 =~
    L3                 0.355    0.250    1.419    0.156
```

Even more trait-state

Intercepts:

	Estimate	Std.Err	Z-value	P(>|z|)
x1	0.000			
x2	0.000			
x3	0.000			
x4	0.000			
x5	0.000			
x6	0.000			
x7	0.000			
x8	0.000			
x9	0.000			
x10	0.000			
x11	0.000			
x12	0.000			
L1	-0.275	0.092	-3.001	0.003
L2	-0.747	0.403	-1.856	0.063
L3	-0.705	0.667	-1.057	0.290
L4	1.183	0.559	2.116	0.034
i	-0.545	0.520	-1.048	0.295
s	1.391	0.044	31.809	0.000

lavaan growth model with trait and state effects: equal state autocorrelations

R code

```
modelc  <- 'L1 =~ a*x1 +b *x2 + c* x3
            L2 =~ a* x4 + b * x5 + c* x6
            L3 =~ a*x7 + b*x8 +c*x9
            L4 =~ a*x10 + b*x11 + c*x12
            i =~ 1* L1 + 1 * L2 + 1 * L3 + 1 * L4
            s =~ 0 * L1 + 1 * L2 + 2* L3 + 3* L4
            L2 =~ alpha* L1
            L3 =~ alpha * L2
            L4 =~ alpha *  L3'

fitc <- growth(modelc,data=simp4$observed)
summary(fitc)
```

```
  lavaan (0.5-19) converged normally after  45 iterations

   Number of observations                          1000

   Estimator                                         ML
   Minimum Function Test Statistic               83.968
   Degrees of freedom                                62
   P-value (Chi-square)                           0.033

 Parameter Estimates:

   Information                                 Expected
```

more output

```
Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  L1 =~
    x1          (a)    1.000
    x2          (b)    0.873    0.005  189.674    0.000
    x3          (c)    0.749    0.005  156.796    0.000
 ...

  L4 =~
    x10         (a)    1.000
    x11         (b)    0.873    0.005  189.674    0.000
    x12         (c)    0.749    0.005  156.796    0.000
  i =~
    L1                 1.000
    L2                 1.000
    L3                 1.000
    L4                 1.000
  s =~
    L1                 0.000
    L2                 1.000
    L3                 2.000
    L4                 3.000
  L2 =~
    L1       (alph)    0.256    0.111    2.312    0.021
  L3 =~
    L2       (alph)    0.256    0.111    2.312    0.021
  L4 =~
    L3       (alph)    0.256    0.111    2.312    0.021
```

Compare these four models

Models is just state change with fixed autocorrelation
Modela was the trait change model
Modelb was trait change with free state autocorrelation
Modelc was trait change with fixed state autocorrelation

R code

```
anova(fits,fita,fitb,fitc)
```

```
Chi Square Difference Test

     Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fitb 60 30024 30171 81.949
fitc 62 30022 30159 83.968     2.0186       2  0.3644668
fita 63 30032 30165 96.397    12.4296       1  0.0004226 ***
fits 67 30012 30125 84.661   -11.7359       4  1.0000000
```

Create four factor-state trait data set

R code

```
set.seed(42)
#explicitly stating some of the default values
simp4 <- sim(mu=c(0,2,4,6),n=1000,alpha=.3,lambda=.6)
colnames(simp4$observed) <- paste0("x",1:12)
describe(simp4$observed)
```

```
     vars    n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
x1      1 1000  0.02 1.03   0.00    0.02 0.96 -3.97 3.75  7.72 -0.01     0.63 0.03
x2      2 1000  0.03 0.97   0.05    0.03 1.01 -3.02 3.51  6.52  0.02     0.16 0.03
x3      3 1000 -0.01 1.02   0.00   -0.01 1.03 -3.02 3.54  6.57  0.03    -0.03 0.03
x4      4 1000  1.62 1.01   1.61    1.62 1.09 -2.41 4.48  6.89 -0.07    -0.14 0.03
x5      5 1000  1.40 1.02   1.44    1.40 0.98 -1.95 4.74  6.69 -0.03     0.24 0.03
x6      6 1000  1.25 1.01   1.28    1.26 1.01 -1.86 3.98  5.85 -0.10    -0.14 0.03
x7      7 1000  3.24 0.98   3.23    3.25 1.00  0.21 6.51  6.30 -0.04    -0.11 0.03
x8      8 1000  2.80 0.99   2.74    2.80 0.93 -1.29 6.33  7.62  0.00     0.30 0.03
x9      9 1000  2.39 1.01   2.39    2.38 1.03 -0.32 5.43  5.75  0.11    -0.14 0.03
x10    10 1000  4.82 0.98   4.82    4.82 0.96  1.66 8.54  6.88  0.01     0.13 0.03
x11    11 1000  4.19 1.01   4.21    4.18 1.05  0.99 7.13  6.14  0.06    -0.13 0.03
x12    12 1000  3.63 1.01   3.64    3.64 1.01 -0.15 6.78  6.93 -0.04     0.02 0.03
```

With correlations very similar to our prior model

R code

```
lowerCor(simp4$observed)
```

```
     x1   x2   x3   x4   x5   x6   x7   x8   x9   x10  x11  x12
x1   1.00
x2   0.56 1.00
x3   0.45 0.40 1.00
x4   0.44 0.41 0.33 1.00
x5   0.36 0.35 0.27 0.59 1.00
x6   0.28 0.25 0.25 0.43 0.42 1.00
x7   0.27 0.21 0.16 0.41 0.35 0.29 1.00
x8   0.24 0.18 0.19 0.36 0.31 0.24 0.53 1.00
x9   0.19 0.20 0.16 0.32 0.33 0.23 0.50 0.44 1.00
x10  0.31 0.24 0.19 0.30 0.27 0.18 0.41 0.36 0.34 1.00
x11  0.26 0.19 0.19 0.31 0.24 0.18 0.38 0.34 0.32 0.56 1.00
x12  0.25 0.19 0.16 0.24 0.16 0.13 0.28 0.22 0.23 0.48 0.42 1.00
```

Try a lavaan fit

R code

```
modelb <- 'L1 =~ a*x1 +b *x2 + c* x3
           L2 =~ a* x4 + b * x5 + c* x6
           L3 =~ a*x7 + b*x8 +c*x9
           L4 =~ a*x10 + b*x11 + c*x12
           i =~ 1* L1 + 1 * L2 + 1 * L3 + 1 * L4
           s =~ 0 * L1 + 1 * L2 + 2* L3 + 3* L4
           L2 =~ alpha *  L1
           L3 =~alpha *  L2
           L4 =~alpha *  L3'

fitb <- growth(modelb,data=simp4$observed)
summary(fitb)
```

```
Warning messages:
1: In lavData(data = data, group = group, group.label = group.label,  :
  lavaan WARNING: data argument has been coerced to a data.frame
2: In lav_object_post_check(lavobject) :
  lavaan WARNING: some estimated variances are negative
3: In lav_object_post_check(lavobject) :
  lavaan WARNING: covariance matrix of latent variables is not positive definite; use inspe
>
```

Struct Equ Modeling. Author manuscript; available in PMC 2014 Dec 8. Published in final edited form as: Struct Equ Modeling. 2012; 19(4): 651■682. doi: 10.1080/10705511.2012.713275