The semexamples: R point of view

Ben Kite[1] and Paul Johnson[1]

[1]Center for Research Methods and Data Analysis

2017

Outline

The Repository

- CRMDA created an archive of SEM working examples.
- To the extent possible, the same models are estimated with
  - Mplus
  - R
  - Stata
- The repository can be browsed online: https://gitlab.crmda.ku.edu/crmda/semexample
  - And a snapshot can also be obtained with Git.
- The data folder includes the information that is imported into each of the 3 programs.

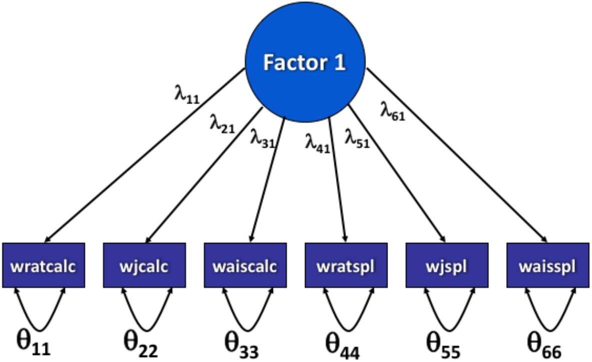
Runable examples

1. Exploratory Factor Analysis (EFA)
2. Confirmatory Factor Analysis (CFA)
3. Measurement Invariance (Multi-group CFA)
4. Multiple Regression
5. Structural Equation Model (SEM)
6. Latent Growth Curve (LGC)
7. Modeling strategy changes for Ordinal Data

Explatory Factor Analysis

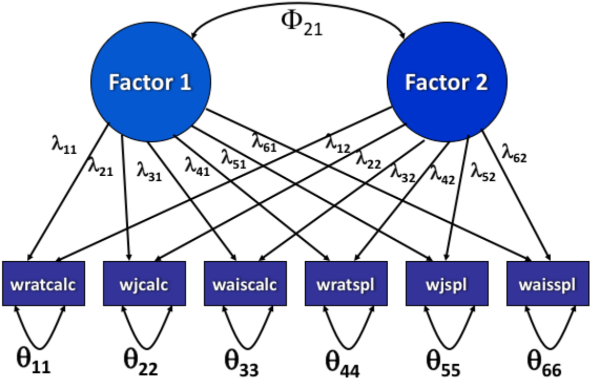
- The Excelsior Springs data. How many latent variables underlie subtest scores (6 items)?
- R's function `factanal` is delivered with the R base package set
- We run EFAs for one and two possible latent factors.

One Factor



Factor 1
λ11
λ21
λ31
λ41
λ51
λ61
wratcalc
wjcalc
waiscalc
wratspl
wjspl
waisspl
θ11
θ22
θ33
θ44
θ55
θ66

Two Factors


Φ_{21}
Factor 1
Factor 2
λ_{11}
λ_{21}
λ_{31}
λ_{41}
λ_{51}
λ_{61}
λ_{12}
λ_{22}
λ_{32}
λ_{42}
λ_{52}
λ_{62}
wratcalc
wjcalc
waiscalc
wratspl
wjspl
waisspl
θ_{11}
θ_{22}
θ_{33}
θ_{44}
θ_{55}
θ_{66}

R code and results

```
head(scalescores)
```

```
  wjcalc  wjspl  wratspl  wratcalc  waiscalc  waisspl
1     29     51       47        49        12       45
2     31     54       57        49        21       47
3     23     45       38        40         9       36
4     31     34       38        48        14       38
5     31     31       30        52        15       38
6     27     42       40        42        12       41
```

```
output1 <- factanal(scalescores, 1, rotation =
    "varimax")
output1
```

R code and results ...

```
Call:
factanal(x = scalescores, factors = 1, rotation = "varimax")

Uniquenesses:
  wjcalc     wjspl   wratspl  wratcalc  waiscalc   waisspl
   0.728     0.093     0.108     0.695     0.749     0.116

Loadings:
         Factor1
wjcalc   0.522
wjspl    0.953
wratspl  0.945
wratcalc 0.552
waiscalc 0.501
waisspl  0.940

               Factor1
SS loadings      3.511
Proportion Var   0.585

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 461.38 on 9 degrees of freedom.
The p-value is 1.06e-93
```

R code and results ...

```
output2 <- factanal(scalescores, 2, rotation =
    "varimax")
output2
```

```
Call:
factanal(x = scalescores, factors = 2, rotation = "varimax")

Uniquenesses:
  wjcalc     wjspl   wratspl  wratcalc  waiscalc   waisspl
   0.184     0.089     0.107     0.096     0.477     0.112

Loadings:
         Factor1  Factor2
wjcalc    0.230    0.873
wjspl     0.907    0.298
wratspl   0.894    0.306
wratcalc  0.248    0.918
waiscalc  0.281    0.667
waisspl   0.896    0.293

                Factor1  Factor2
SS loadings       2.617    2.318
Proportion Var    0.436    0.386
```

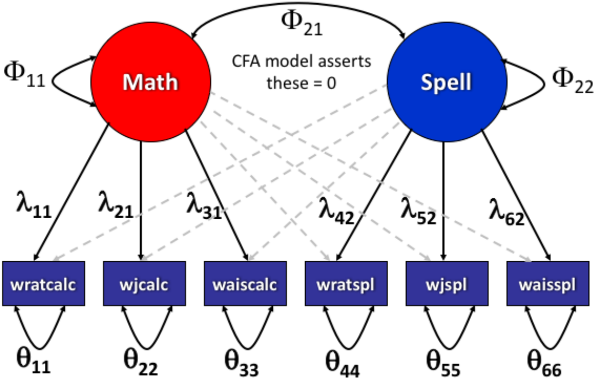
R code and results ...

```
Cumulative Var    0.436    0.823

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 3.8 on 4 degrees of freedom.
The p-value is 0.434
```

Confirmatory Factor Analysis

- EFA allowed connections between each factor and each indicator
- In CFA, we restrict–for theoretical reasons–linkages between factors and indicators
- Math is indicated by:
  - wratcalc, wjcalc, waiscalc
- Spelling is indicated by:
  - wratspl, wjspl, waisspl

Two Factor CFA



R code and results

```
library(lavaan)
CFAmodel <- "
   MATH =~ wratcalc + wjcalc + waiscalc
   SPELL =~ wratspl + wjspl + waisspl
 "
testOutput <- cfa(model = CFAmodel, data = dat,
    std.lv = TRUE, missing = "fiml")
summary(testOutput, fit.measures = TRUE)
```

```
lavaan (0.5-22) converged normally after  73 iterations

  Number of observations                           322

  Number of missing patterns                         4

  Estimator                                         ML
  Minimum Function Test Statistic                9.540
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.299
```

R code and results ...

```
Model test baseline model:

  Minimum Function Test Statistic             1882.335
  Degrees of freedom                                15
  P-value                                        0.000

User model versus baseline model:

  Comparative Fit Index (CFI)                    0.999
  Tucker-Lewis Index (TLI)                       0.998

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -5127.830
  Loglikelihood unrestricted model (H1)      -5123.060

  Number of free parameters                         19
  Akaike (AIC)                               10293.661
  Bayesian (BIC)                             10365.377
  Sample-size adjusted Bayesian (BIC)        10305.112

Root Mean Square Error of Approximation:

  RMSEA                                          0.024
  90 Percent Confidence Interval          0.000  0.073
```

R code and results ...

```
  P-value RMSEA <= 0.05                              0.761

Standardized Root Mean Square Residual:

  SRMR                                               0.024

Parameter Estimates:

  Information                                     Observed
  Standard Errors                                 Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  MATH =~
    wratcalc          6.041    0.276   21.921    0.000
    wjcalc            4.144    0.203   20.370    0.000
    waiscalc          2.410    0.165   14.636    0.000
  SPELL =~
    wratspl           6.532    0.288   22.645    0.000
    wjspl             6.809    0.296   23.025    0.000
    waisspl           6.354    0.283   22.463    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  MATH ~~
```

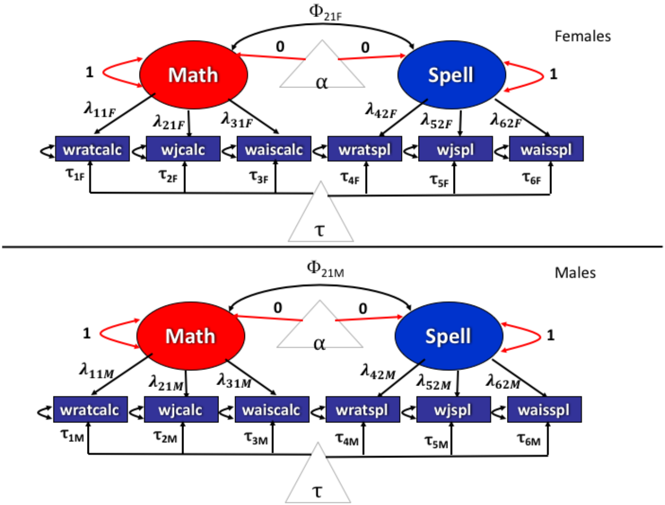
R code and results ...

```
    SPELL            0.553     0.042    13.191     0.000

Intercepts:
                  Estimate   Std.Err   z-value   P(>|z|)
   .wratcalc        38.922     0.355   109.514     0.000
   .wjcalc          23.812     0.255    93.297     0.000
   .waiscalc        11.022     0.186    59.230     0.000
   .wratspl         36.484     0.385    94.751     0.000
   .wjspl           41.674     0.398   104.808     0.000
   .waisspl         37.163     0.376    98.788     0.000
    MATH             0.000
    SPELL            0.000

Variances:
                  Estimate   Std.Err   z-value   P(>|z|)
   .wratcalc         4.179     1.014     4.122     0.000
   .wjcalc           3.769     0.537     7.017     0.000
   .waiscalc         5.304     0.457    11.596     0.000
   .wratspl          5.053     0.612     8.256     0.000
   .wjspl            4.541     0.616     7.369     0.000
   .waisspl          5.156     0.599     8.605     0.000
    MATH             1.000
    SPELL            1.000
```

Multiple Group CFA

- Compare males and females on Math and Spelling

Two group model



Goal: Simplify separate parameter sets

- If the two genders are given completely separate models, there is not much hope of achieving an analytical purpose, which is to compare differences in the latent variable across genders.

Model comparison process

1. Fit the model with all measurement parameters free to vary between groups
2. Fit a model in which measurement parameters are assumed to be the same
3. Conduct a "$\Delta\chi^2$" test to find out if restricting some parameters to be equal between groups caused the model to fit the data poorly.

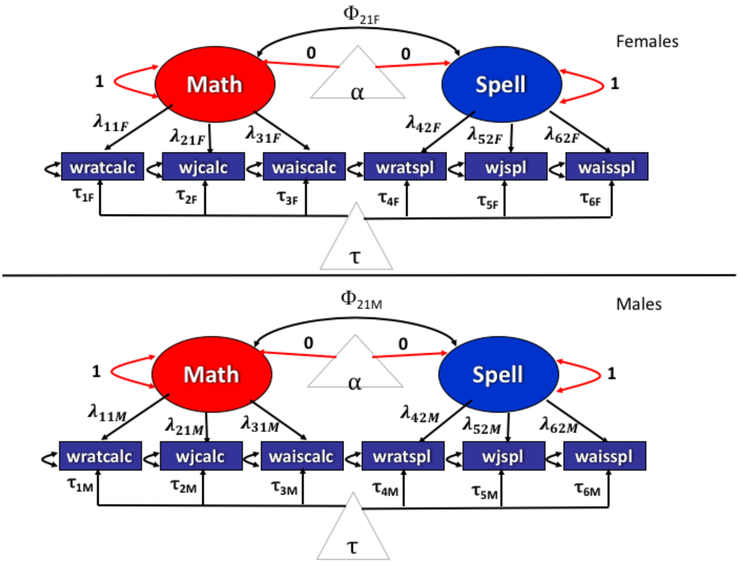
Configural, Metric, Scalar

- Three stages of comparison (models successively drop coefficient differences for the 2 genders)

1. Configural invariance (same model)
2. Metric invariance (factor loadings same for both genders)
3. Scalar invariance (item intercepts are also the same)

Outline

Configural Invariance Model



subscript "g" indicates that the parameters estimated for each group

R code and results

```
configModel <- "
  MATH =~ wratcalc + wjcalc + waiscalc
  SPELL =~ wratspl + wjspl + waisspl
 "
configOutput <- cfa(model = configModel, data =
    dat, group = "female", std.lv = TRUE, missing
    = "fiml")
summary(configOutput)
```

```
lavaan (0.5-22) converged normally after 118 iterations

  Number of observations per group
  1                                                 101
  0                                                 221

  Number of missing patterns per group
  1                                                   4
  0                                                   2

  Estimator                                          ML
```

R code and results ...

```
  Minimum Function Test Statistic               19.215
  Degrees of freedom                                16
  P-value (Chi-square)                           0.258

Chi-square for each group:

  1                                             11.555
  0                                              7.660

Parameter Estimates:

  Information                                 Observed
  Standard Errors                             Standard


Group 1 [1]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  MATH =~
    wratcalc          6.545    0.496   13.187    0.000
    wjcalc            4.215    0.366   11.530    0.000
    waiscalc          2.290    0.276    8.306    0.000
  SPELL =~
    wratspl           7.010    0.547   12.817    0.000
```

R code and results ...

```
    wjspl           6.833    0.518   13.182    0.000
    waisspl         6.638    0.520   12.763    0.000

Covariances:
                 Estimate  Std.Err  z-value  P(>|z|)
  MATH ~~
    SPELL           0.634    0.064    9.914    0.000

Intercepts:
                 Estimate  Std.Err  z-value  P(>|z|)
   .wratcalc       38.267    0.666   57.464    0.000
   .wjcalc         23.603    0.465   50.781    0.000
   .waiscalc       10.266    0.316   32.528    0.000
   .wratspl        37.171    0.735   50.590    0.000
   .wjspl          42.337    0.705   60.073    0.000
   .waisspl        38.030    0.697   54.579    0.000
    MATH            0.000
    SPELL           0.000

Variances:
                 Estimate  Std.Err  z-value  P(>|z|)
   .wratcalc        1.957    1.621    1.208    0.227
   .wjcalc          3.960    0.848    4.671    0.000
   .waiscalc        4.765    0.714    6.678    0.000
   .wratspl         5.307    1.105    4.804    0.000
```

R code and results ...

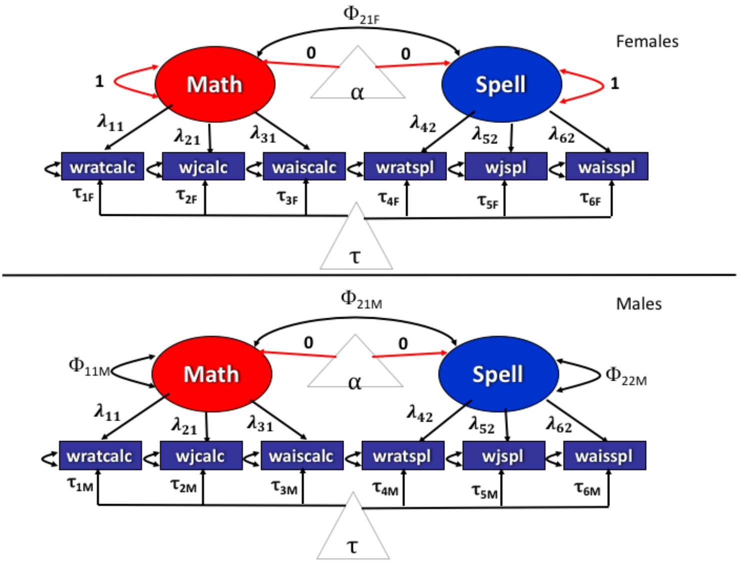
```
    .wjspl            3.474     0.910     3.818     0.000
    .waisspl          4.900     1.007     4.865     0.000
     MATH             1.000
     SPELL            1.000


Group 2 [0]:

Latent Variables:
                   Estimate   Std.Err   z-value   P(>|z|)
  MATH =~
    wratcalc          5.764     0.330    17.492     0.000
    wjcalc            4.123     0.244    16.926     0.000
    waiscalc          2.446     0.200    12.212     0.000
  SPELL =~
    wratspl           6.278     0.337    18.646     0.000
    wjspl             6.788     0.359    18.929     0.000
    waisspl           6.177     0.335    18.441     0.000

Covariances:
                   Estimate   Std.Err   z-value   P(>|z|)
  MATH ~~
    SPELL             0.520     0.053     9.762     0.000

Intercepts:
```

R code and results ...

```
                Estimate  Std.Err  z-value  P(>|z|)
   .wratcalc      39.222    0.417   93.962    0.000
   .wjcalc        23.910    0.305   78.420    0.000
   .waiscalc      11.367    0.226   50.320    0.000
   .wratspl       36.172    0.448   80.762    0.000
   .wjspl         41.371    0.480   86.157    0.000
   .waisspl       36.767    0.444   82.899    0.000
    MATH           0.000
    SPELL          0.000

Variances:
                Estimate  Std.Err  z-value  P(>|z|)
   .wratcalc       5.280    1.232    4.285    0.000
   .wjcalc         3.549    0.664    5.342    0.000
   .waiscalc       5.269    0.557    9.463    0.000
   .wratspl        4.915    0.726    6.772    0.000
   .wjspl          4.881    0.795    6.138    0.000
   .waisspl        5.285    0.736    7.176    0.000
    MATH           1.000
    SPELL          1.000
```

Outline

Metric Invariance Model



R code and results

```
metricModel <- "
   MATH =~ wratcalc + wjcalc + waiscalc
   SPELL =~ wratspl + wjspl + waisspl
   MATH ~~ c(1, NA)*MATH
   SPELL ~~ c(1, NA)*SPELL
 "
metricOutput <- cfa(model = metricModel, data =
    dat, std.lv = TRUE,
                      group = "female",
                         group.equal = "loadings",
                      missing = "fiml")
summary(metricOutput)
```

R code and results ...

```
lavaan (0.5-22) converged normally after  94 iterations

  Number of observations per group
  1                                                 101
  0                                                 221

  Number of missing patterns per group
  1                                                   4
  0                                                   2

  Estimator                                          ML
  Minimum Function Test Statistic                25.871
  Degrees of freedom                                 20
  P-value (Chi-square)                            0.170

Chi-square for each group:

  1                                              15.581
  0                                              10.290

Parameter Estimates:

  Information                                  Observed
  Standard Errors                              Standard
```

R code and results ...

```
Group 1 [1]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  MATH =~
    wratclc (.p1.)    6.359    0.490   12.972    0.000
    wjcalc  (.p2.)    4.354    0.342   12.729    0.000
    waisclc (.p3.)    2.502    0.229   10.909    0.000
  SPELL =~
    wratspl (.p4.)    6.751    0.508   13.295    0.000
    wjspl   (.p5.)    7.020    0.516   13.592    0.000
    waisspl (.p6.)    6.568    0.493   13.326    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  MATH ~~
    SPELL             0.641    0.063   10.164    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wratcalc         38.267    0.656   58.319    0.000
   .wjcalc           23.602    0.474   49.832    0.000
   .waiscalc         10.263    0.330   31.085    0.000
```

R code and results ...

```
   .wratspl          37.172    0.713   52.136    0.000
   .wjspl            42.337    0.721   58.719    0.000
   .waisspl          38.031    0.691   55.052    0.000
    MATH              0.000
    SPELL             0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    MATH              1.000
    SPELL             1.000
   .wratcalc          3.047    1.413    2.157    0.031
   .wjcalc            3.603    0.778    4.632    0.000
   .waiscalc          4.700    0.714    6.586    0.000
   .wratspl           5.686    1.123    5.061    0.000
   .wjspl             3.230    0.929    3.478    0.001
   .waisspl           4.995    1.023    4.883    0.000


Group 2 [0]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  MATH =~
    wratclc  (.p1.)    6.359    0.490   12.972    0.000
    wjcalc   (.p2.)    4.354    0.342   12.729    0.000
```

R code and results ...

```
    waisclc  (.p3.)     2.502    0.229   10.909    0.000
  SPELL =~
    wratspl  (.p4.)     6.751    0.508   13.295    0.000
    wjspl    (.p5.)     7.020    0.516   13.592    0.000
    waisspl  (.p6.)     6.568    0.493   13.326    0.000

Covariances:
                     Estimate  Std.Err  z-value  P(>|z|)
  MATH ~~
    SPELL               0.456    0.089    5.103    0.000

Intercepts:
                     Estimate  Std.Err  z-value  P(>|z|)
   .wratcalc           39.222    0.422   92.880    0.000
   .wjcalc             23.910    0.302   79.192    0.000
   .waiscalc           11.367    0.221   51.462    0.000
   .wratspl            36.172    0.454   79.673    0.000
   .wjspl              41.371    0.472   87.637    0.000
   .waisspl            36.767    0.446   82.522    0.000
    MATH                0.000
    SPELL               0.000

Variances:
                     Estimate  Std.Err  z-value  P(>|z|)
    MATH                0.860    0.156    5.501    0.000
```

R code and results ...

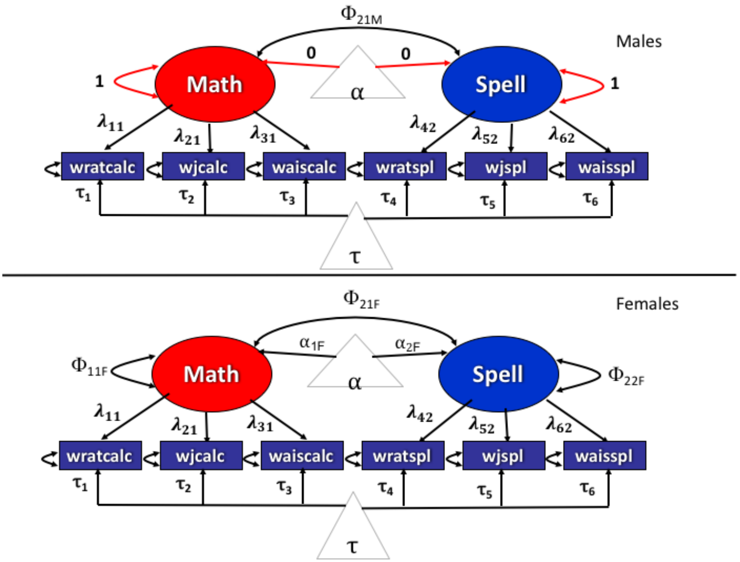
```
   SPELL              0.895    0.158    5.676    0.000
  .wratcalc           4.619    1.205    3.834    0.000
  .wjcalc             3.835    0.654    5.864    0.000
  .waiscalc           5.370    0.564    9.525    0.000
  .wratspl            4.764    0.721    6.607    0.000
  .wjspl              5.154    0.790    6.526    0.000
  .waisspl            5.233    0.732    7.153    0.000
```

```
anova(configOutput, metricOutput)
```

```
Chi Square Difference Test

             Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
configOutput 16 10301 10444 19.215
metricOutput 20 10300 10428 25.871     6.6558       4     0.1552
```

Outline

Scalar Invariance Model



R code and results

```
scalarModel <- "
   MATH =~ wratcalc + wjcalc + waiscalc
   SPELL =~ wratspl + wjspl + waisspl
   MATH ~~ c(1, NA)*MATH
   SPELL ~~ c(1, NA)*SPELL
   MATH ~ c(0, NA)*1
   SPELL ~ c(0, NA)*1
 "
scalarOutput <- cfa(model = scalarModel, data =
    dat, std.lv = TRUE,
                    group = "female",
                    group.equal = c("loadings",
                       "intercepts"),
                    missing = "fiml")
summary(scalarOutput)
```

R code and results ...

```
lavaan (0.5-22) converged normally after  95 iterations

  Number of observations per group
  1                                                  101
  0                                                  221

  Number of missing patterns per group
  1                                                    4
  0                                                    2

  Estimator                                           ML
  Minimum Function Test Statistic                 36.362
  Degrees of freedom                                  24
  P-value (Chi-square)                             0.051

Chi-square for each group:

  1                                               22.546
  0                                               13.815

Parameter Estimates:

  Information                                   Observed
  Standard Errors                               Standard
```

R code and results ...

```
Group 1 [1]:

Latent Variables:
                      Estimate  Std.Err  z-value  P(>|z|)
  MATH =~
    wratclc  (.p1.)      6.381    0.491   12.994    0.000
    wjcalc   (.p2.)      4.340    0.342   12.699    0.000
    waisclc  (.p3.)      2.529    0.232   10.895    0.000
  SPELL =~
    wratspl  (.p4.)      6.751    0.508   13.299    0.000
    wjspl    (.p5.)      7.015    0.516   13.592    0.000
    waisspl  (.p6.)      6.578    0.494   13.329    0.000

Covariances:
                      Estimate  Std.Err  z-value  P(>|z|)
  MATH ~~
    SPELL                0.640    0.063   10.128    0.000

Intercepts:
                      Estimate  Std.Err  z-value  P(>|z|)
    MATH                 0.000
    SPELL                0.000
   .wratclc  (.18.)     38.273    0.653   58.628    0.000
```

R code and results ...

```
    .wjcalc   (.19.)   23.374    0.454    51.465    0.000
    .waisclc  (.20.)   10.751    0.289    37.239    0.000
    .wratspl  (.21.)   37.211    0.692    53.771    0.000
    .wjspl    (.22.)   42.411    0.713    59.443    0.000
    .waisspl  (.23.)   37.869    0.675    56.144    0.000

Variances:
                    Estimate   Std.Err   z-value   P(>|z|)
     MATH              1.000
     SPELL             1.000
    .wratcalc          2.841     1.424     1.995     0.046
    .wjcalc            3.743     0.793     4.723     0.000
    .waiscalc          4.977     0.766     6.496     0.000
    .wratspl           5.681     1.123     5.056     0.000
    .wjspl             3.237     0.930     3.481     0.000
    .waisspl           5.035     1.032     4.880     0.000


Group 2 [0]:

Latent Variables:
                    Estimate   Std.Err   z-value   P(>|z|)
  MATH =~
     wratclc  (.p1.)   6.381    0.491    12.994    0.000
     wjcalc   (.p2.)   4.340    0.342    12.699    0.000
```

R code and results ...

```
    waisclc  (.p3.)     2.529     0.232    10.895     0.000
  SPELL =~
    wratspl  (.p4.)     6.751     0.508    13.299     0.000
    wjspl    (.p5.)     7.015     0.516    13.592     0.000
    waisspl  (.p6.)     6.578     0.494    13.329     0.000

Covariances:
                     Estimate   Std.Err   z-value   P(>|z|)
  MATH ~~
    SPELL               0.454     0.089     5.104     0.000

Intercepts:
                     Estimate   Std.Err   z-value   P(>|z|)
    MATH                0.148     0.121     1.220     0.222
    SPELL              -0.156     0.121    -1.293     0.196
   .wratclc  (.18.)    38.273     0.653    58.628     0.000
   .wjcalc   (.19.)    23.374     0.454    51.465     0.000
   .waisclc  (.20.)    10.751     0.289    37.239     0.000
   .wratspl  (.21.)    37.211     0.692    53.771     0.000
   .wjspl    (.22.)    42.411     0.713    59.443     0.000
   .waisspl  (.23.)    37.869     0.675    56.144     0.000

Variances:
                     Estimate   Std.Err   z-value   P(>|z|)
    MATH                0.857     0.156     5.502     0.000
```

R code and results ...

```
  SPELL                0.894    0.158    5.676    0.000
 .wratcalc             4.518    1.204    3.753    0.000
 .wjcalc               3.911    0.655    5.970    0.000
 .waiscalc             5.431    0.576    9.435    0.000
 .wratspl              4.762    0.721    6.603    0.000
 .wjspl                5.173    0.791    6.541    0.000
 .waisspl              5.235    0.734    7.134    0.000
```

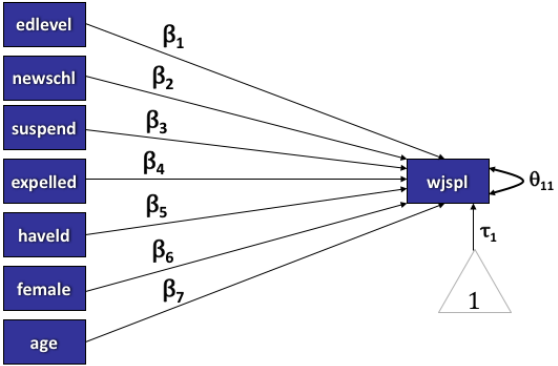
```
anova(metricOutput, scalarOutput)
```

```
Chi Square Difference Test

             Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
metricOutput 20 10300 10428 25.871
scalarOutput 24 10302 10415 36.361     10.491       4    0.03293

metricOutput
scalarOutput *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regression

- The SEM framework includes the ordinary linear regression on observed variables as a special case
- SEM can fit several linear regressions at once
  - possibly allowing correlation between the error terms in the 2 regressions (Seemingly Unrelated Regressions, SUR).

Multiple predictors



R code and results

```
MLRModel <- "
  wjspl ~ edlevel + newschl + suspend + expelled +
      haveld + female + age
  wjspl ~ 1
  "
output <- sem(model = MLRModel, data = dat)
summary(output)
```

```
lavaan (0.5-22) converged normally after  58 iterations

                                                  Used       Total
  Number of observations                           313         322

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Information                                 Expected
```

R code and results ...

```
  Standard Errors                              Standard

Regressions:
                    Estimate  Std.Err  z-value  P(>|z|)
  wjspl ~
    edlevel            1.162    0.324    3.584    0.000
    newschl            0.063    0.747    0.085    0.932
    suspend           -0.052    0.773   -0.067    0.947
    expelled          -2.758    1.098   -2.510    0.012
    haveld            -6.974    0.987   -7.063    0.000
    female             0.720    0.792    0.909    0.363
    age                0.412    0.207    1.994    0.046

Intercepts:
                    Estimate  Std.Err  z-value  P(>|z|)
   .wjspl             21.953    4.414    4.974    0.000

Variances:
                    Estimate  Std.Err  z-value  P(>|z|)
   .wjspl             39.904    3.190   12.510    0.000
```

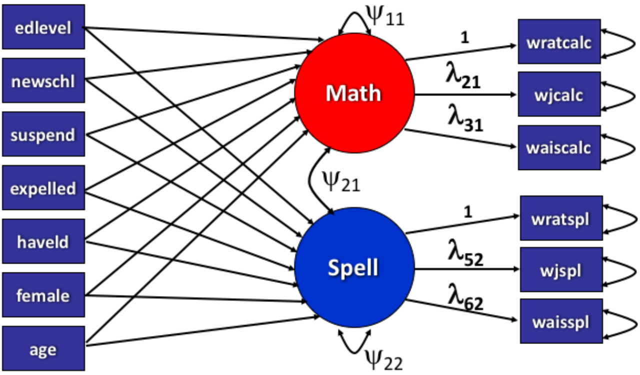
Regression

- Estimate regression coefficients that link together latent (unmeasured) variables.
- This is an example of a LISREL model (measurement and structural component)

Math and Spelling

- Continues with previous data
- Research question: are variations in the latent factors “Math” and “Spell” predictable using the variables
    - edlevel
    - newschl
    - suspend
    - expelled
    - haveld
    - female
    - age

SEM



edlevel
newschl
suspend
expelled
haveld
female
age
ψ11
Math
ψ21
Spell
ψ22
1
λ21
λ31
1
λ52
λ62
wratcalc
wjcalc
waiscalc
wratspl
wjspl
waisspl

R code and results

```
SEMModel <- "
   MATH =~ wratcalc + wjcalc + waiscalc
   SPELL =~ wratspl + wjspl + waisspl
   MATH ~ edlevel + newschl + suspend + expelled
      + haveld + female + age
   SPELL ~ edlevel + newschl + suspend + expelled
      + haveld + female + age
   MATH ~~ SPELL
 "
output <- sem(model = SEMModel, data = dat,
   missing = "fiml")
summary(output)
```

R code and results ...

```
lavaan (0.5-22) converged normally after 196 iterations

  Number of observations                           322

  Number of missing patterns                         9

  Estimator                                         ML
  Minimum Function Test Statistic               53.744
  Degrees of freedom                                36
  P-value (Chi-square)                           0.029

Parameter Estimates:

  Information                                 Observed
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
 MATH =~
    wratcalc          1.000
    wjcalc            0.683    0.027   25.075    0.000
    waiscalc          0.395    0.024   16.229    0.000
 SPELL =~
    wratspl           1.000
```

R code and results ...

```
    wjspl          1.044    0.029   36.398    0.000
    waisspl        0.974    0.028   34.537    0.000

Regressions:
                 Estimate  Std.Err  z-value  P(>|z|)
  MATH ~
    edlevel        1.555    0.265    5.870    0.000
    newschl        0.504    0.640    0.787    0.431
    suspend       -1.767    0.666   -2.652    0.008
    expelled      -0.533    0.943   -0.565    0.572
    haveld        -1.436    0.844   -1.701    0.089
    female        -0.847    0.677   -1.250    0.211
    age            0.634    0.176    3.601    0.000
  SPELL ~
    edlevel        1.143    0.285    4.016    0.000
    newschl       -0.258    0.688   -0.375    0.707
    suspend       -0.008    0.711   -0.012    0.991
    expelled      -2.510    1.016   -2.470    0.014
    haveld        -6.369    0.909   -7.004    0.000
    female         0.970    0.724    1.340    0.180
    age            0.351    0.189    1.860    0.063

Covariances:
                 Estimate  Std.Err  z-value  P(>|z|)
 .MATH ~~
```

R code and results ...

```
   .SPELL          14.951    1.993    7.504    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
    .wratcalc      10.349    3.676    2.815    0.005
    .wjcalc         4.288    2.585    1.659    0.097
    .waiscalc      -0.278    1.585   -0.175    0.861
    .wratspl       18.069    3.941    4.585    0.000
    .wjspl         22.455    4.110    5.463    0.000
    .waisspl       19.219    3.840    5.004    0.000
   .MATH            0.000
   .SPELL           0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    .wratcalc       3.963    0.971    4.083    0.000
    .wjcalc         3.814    0.526    7.246    0.000
    .waiscalc       5.372    0.460   11.678    0.000
    .wratspl        5.131    0.609    8.426    0.000
    .wjspl          4.526    0.609    7.430    0.000
    .waisspl        5.095    0.595    8.570    0.000
   .MATH           27.269    2.540   10.736    0.000
   .SPELL          32.493    2.908   11.174    0.000
```

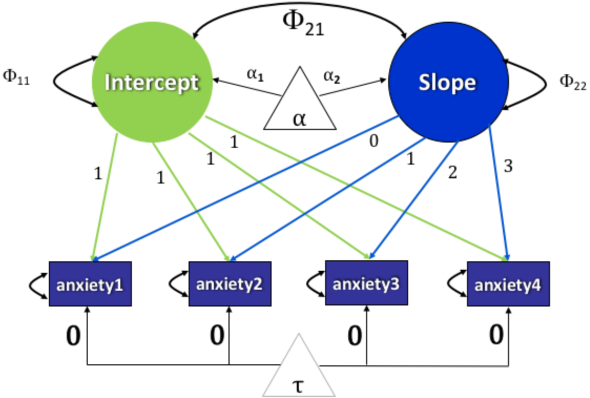
Latent Growth Curve Modeling

- Repeated measurements allow us to think about change in a latent variable over time
- If we can predictably measure the latent variable, then we can assess its linkage to other outcomes (both observed and latent).
- However, in this example we are mostly interested in the possibility that
  - some people are more anxious than others at the outset of the study, and they remain more (or less) anxious across time
  - as time passes, some people become progressively more and more (or less and less) anxious

Measurements

- The data set is named “anxiety.dat”
- participants anxiety level at four different time points were measured
- Do anxiety levels change over time?
- Initial anxiety (intercept) and change in anxiety (slope) can be modeled as latent variables.

Latent Growth Curve Graph



R code and results

```
dat <- read.table("../data/anxiety.dat", header =
    F)
names(dat) <- c("a1", "a2", "a3", "a4")
model <- "
    intercept =~ 1*a1 + 1*a2 + 1*a3 + 1*a4
    slope =~ 0*a1 + 1*a2 + 2*a3 + 3*a4
    a1 ~ 0*1
    a2 ~ 0*1
    a3 ~ 0*1
    a4 ~ 0*1
    intercept ~ 1
    slope ~ 1
  "
output <- sem(model, data = dat)
summary(output)
```

R code and results ...

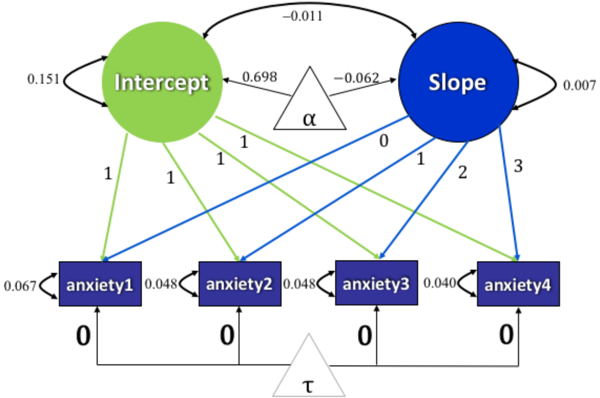
```
lavaan (0.5-22) converged normally after  45 iterations

  Number of observations                           485

  Estimator                                         ML
  Minimum Function Test Statistic               27.288
  Degrees of freedom                                 5
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  intercept =~
    a1                1.000
    a2                1.000
    a3                1.000
    a4                1.000
  slope =~
    a1                0.000
    a2                1.000
```

R code and results ...

```
    a3              2.000
    a4              3.000

Covariances:
                 Estimate  Std.Err  z-value  P(>|z|)
  intercept ~~
    slope          -0.011    0.003   -3.472    0.001

Intercepts:
                 Estimate  Std.Err  z-value  P(>|z|)
   .a1              0.000
   .a2              0.000
   .a3              0.000
   .a4              0.000
    intercept       0.698    0.020   35.050    0.000
    slope          -0.062    0.006  -10.513    0.000

Variances:
                 Estimate  Std.Err  z-value  P(>|z|)
   .a1              0.067    0.007    8.910    0.000
   .a2              0.048    0.004   11.119    0.000
   .a3              0.048    0.004   11.383    0.000
   .a4              0.040    0.006    6.658    0.000
    intercept       0.151    0.013   11.871    0.000
    slope           0.007    0.001    4.790    0.000
```

R code and results ...

Latent Growth Results



SEM Examples

- More examples will be created in future

Session

```
sessionInfo()
```

```
R version 3.3.3 (2017-03-06)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 17.04

locale:
 [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8           LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8       LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8          LC_NAME=C
 [9] LC_ADDRESS=C                  LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  base

other attached packages:
[1] lavaan_0.5-22

loaded via a namespace (and not attached):
[1] MASS_7.3-45     tools_3.3.3     mnormt_1.5-5    pbivnorm_0.6.0
[5] methods_3.3.3   stats4_3.3.3    quadprog_1.5-5
```