Links to Software for Covariance-Based SEM

Stand-Alone Commercial Products

Amos, https://www.ibm.com/products/structural-equation-modeling-sem

EQS, http://www.mvsoft.com/

LISREL, https://www.ssicentral.com/index.php/product/lisrel

Mplus, https://www.statmodel.com/

Commercial Products for General Statistical Analyses with SEM Procedures

SAS/STAT, https://www.sas.com/en_us/software/stat.html

Stata, https://www.stata.com/

SYSTAT, https://systatsoftware.com/

Freely-Available Tools with Graphical User Interfaces

JASP, https://jasp-stats.org/

Onyx, https://www.statista.com/

Packages for the R Computing Environment

R, https://www.r-project.org/

General Latent Variable Modeling

lavaan, https://cran.r-project.org/package=lavaan

OpenMx, https://CRAN.R-project.org/package=OpenMx

sem, https://CRAN.R-project.org/package=sem

Methods for Continuous, Censored, and Ordinal Outcomes and Graph Analysis

lava, https://CRAN.R-project.org/package=lava

Plot Model Diagrams

semPlot, https://CRAN.R-project.org/package=semPlot

Simulation for Power Analysis and Model Fit Assessment

simsem, https://CRAN.R-project.org/package=simsem

Tools for Testing Measurement Models and Power Analysis

semTools, https://cran.r-project.org/package=semTools

Annotated lavaan and semTools Input for Detailed Example[1]

```
# brydges, ozolnieks, & roberts (2017) final model
# journal of neuropsychology, 11(3), 362-377
# https://doi.org/10.1111/jnp.12096
# observed variables, lowercase
# common factor, sentence case
date()
library(lavaan)
library(semTools)
options("width"=130)
# input the correlations from brydges et al. in lower diagonal form
brydgesLower.cor <- '
 1
 .02 1
 .04  .43  1
-.04  .29  .50  1
-.24 -.11 -.30 -.14 1
-.11 -.11 -.31 -.25 .12 1
-.05 -.04 -.16 -.07 .27 .09 1
-.17 -.20 -.32 -.29 .31 .20 .24 1 '
brydgesFull.cor <- getCov(brydgesLower.cor, names = c("adhd","simple",
```

```
 "choice2","choice4","digits","blocks","span","fluid"))
# add the standard deviations and convert to covariances
brydgesFull.cov <- cor2cov(brydgesFull.cor, sds = c(11.63,
 26.22,71.31,80.18,2.90,1.64,1.04,4.48))
# add the means
brydgesMeans = c(31.66,318.04,505.81,475.13,6.47,9.02,4.83,27.23)
# means are not analyzed here
# display covariances and means
brydgesFull.cov
brydgesMeans
# specify final model
brydges.model <- '
# measurement model
 Speed =~ simple + choice2 + choice4
 Memory =~ span + digits + blocks
# structural model
 Speed ~ a*adhd
 Memory ~ c*adhd
 fluid ~ b*Speed + d*Memory + e*adhd
 Speed ~~ Memory
# indirect effects of adhd on fluid
 ab := a*b
 cd := c*d
# total indirect effect of adhd on fluid
 totind := a*b + c*d
# total effect of adhd on fluid
```

```
 total := e + a*b + c*d '
# fit model to data
model <- sem(brydges.model, sample.cov = brydgesFull.cov,
 sample.nobs = 142)
# display post-analysis results
summary(model, fit.measures = TRUE, standardized = TRUE,
 rsquare = TRUE)
parameterEstimates(model)
standardizedSolution(model)
fitted(model)
residuals(model, type = "raw")
# some results for standardized residuals are invalid
residuals(model, type = "standardized")
residuals(model, type = "normalized")
residuals(model, type = "cor")
# estimate power of chi-square test
# null value is rmsea = 0
# alternative value is .068
# rmsea = SQRT ((26.296 - 16)/(141 * 16)) = .068
# where 26.296 is critical chi-square (16) at .05
findRMSEApower(0, .068, 16, 142, .05, 1)
# determine minimum sample size for power = .90
findRMSEAsamplesize(0, .068, 16, .90, .05, 1)
```

Annotated lavaan and semTools Output for Detailed Example

```
> # fit model to data
> model <- sem(brydges.model, sample.cov = brydgesFull.cov,
```

```
+  sample.nobs = 142)
>
> # display post-analysis results
> summary(model, fit.measures = TRUE, standardized = TRUE,
+  rsquare = TRUE)
lavaan 0.6-6 ended normally after 359 iterations
  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         19
  Number of observations                           142
```

Target model passes the chi-square test of exact fit; that is, model-data discrepancy does not exceed zero (exact fit) at the .05 level.

```
Model Test User Model:
  Test statistic                                14.847
Degrees of freedom                                16
P-value (Chi-square)                           0.536
```

Model chi-square test for baseline model, which assumes all variables are independent; the baseline model fails the exact fit test, which is not a surprise.

```
Model Test Baseline Model:
Test statistic                                159.545
Degrees of freedom                                28
P-value                                        0.000
User Model versus Baseline Model:
```

CFI and TLI compare target model with baseline model,

value of TLI is out-of-bounds (> 1.0), so CFI is reported.

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.015

Likelihood criteria would be more relevant for comparing alternative models fit to the same data, which does not apply in this analysis of a single model.

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-3466.848
Loglikelihood unrestricted model (H1)	-3459.424
Akaike (AIC)	6971.695
Bayesian (BIC)	7027.856
Sample-size adjusted Bayesian (BIC)	6967.738

RMSEA with its 90% confidence interval.

Root Mean Square Error of Approximation:

RMSEA	0.000
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.072

The model passes a test of close fit based on the RMSEA, that is, model-data discrepancy does not exceed the bounds of close fit, where “close fit” means chi-square <= df.

P-value RMSEA <= 0.05	0.822

Standardized Root Mean Square Residual:

SRMR	0.044

Parameter Estimates:

Standard errors	Standard
Information	Expected

Information saturated (h1) model Structured

Unstandardized and standardized parameter estimates with results of significance tests; confidence intervals are listed in a later section.

The completely standardized solution (Std.all) where all variables--factors, indicators, and error terms—are standardized--is reported in the chapter.

In the Std.lv solution, only the factors are standardized.

If all observed variables have meaningful raw score metrics, the solution Std.lv may be preferred over Std.all.

Latent Variables:

		Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
Speed =~							
simple		1.000				12.722	0.487
choice2		4.847	1.049	4.622	0.000	61.667	0.868
choice4		3.665	0.794	4.617	0.000	46.623	0.584
Memory =~							
span		1.000				0.329	0.317
digits		4.533	1.591	2.848	0.004	1.490	0.515
blocks		1.856	0.733	2.532	0.011	0.610	0.373

Regressions:

		Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
Speed ~							
adhd	(a)	0.033	0.103	0.323	0.747	0.003	0.030
Memory ~							
adhd	(c)	-0.010	0.005	-2.216	0.027	-0.032	-0.368

fluid ~

Speed (b) 0.037 0.135 0.272 0.786 0.467 0.105

Memory (d) 9.752 7.491 1.302 0.193 3.205 0.718

adhd (e) 0.035 0.080 0.437 0.662 0.035 0.091

Error covariance for the speed and memory factors.

Covariances:

Estimate Std.Err z-value P(>|z|) Std.lv Std.all

Speed ~~

Memory -2.845 1.159 -2.454 0.014 -0.732 -0.732

Error variances for the indicators and factors.

Variances:

Estimate Std.Err z-value P(>|z|) Std.lv Std.all

.simple 520.791 68.468 7.606 0.000 520.791

0.763

.choice2 1246.433 607.127 2.053 0.040 1246.433

0.247

.choice4 4209.853 615.042 6.845 0.000 4209.853

0.659

.span 0.966 0.122 7.939 0.000 0.966

0.899

.digits 6.132 0.992 6.183 0.000 6.132

0.734

.blocks 2.299 0.300 7.657 0.000 2.299

0.861

.fluid 12.292 3.786 3.247 0.001 12.292

0.617

```
   .Speed            161.706   60.036    2.693    0.007    0.999
0.999
   .Memory             0.093    0.061    1.519    0.129    0.865
0.865
Proportions of explained variance for the indicators
```

Results listed next are reported in Table 3 of the chapter.

```
R-Square:
                   Estimate
    simple            0.237
    choice2           0.753
    choice4           0.341
    span              0.101
    digits            0.266
    blocks            0.139
    fluid             0.383
    Speed             0.001
    Memory            0.135
```

Estimates for indirect, total indirect, and total effects of ADHD on fluid intelligence through speed and memory factors.

```
Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv
Std.all
    ab                0.001    0.006    0.208    0.835    0.001
0.003
    cd               -0.102    0.083   -1.227    0.220   -0.102   -
```

```
0.264
    totind           -0.101    0.078   -1.281    0.200   -0.101   -
0.261
    total            -0.065    0.032   -2.056    0.040   -0.065   -
0.170
```

Listed next are unstandardized estimates with 95% confidence intervals for all parameters and defined effects; these results are reported in Tables 1-3.

```
> parameterEstimates(model)
       lhs op       rhs  label      est      se       z pvalue ci.lower
ci.upper
1    Speed =~    simple           1.000   0.000      NA     NA    1.000
1.000
2    Speed =~   choice2           4.847   1.049   4.622  0.000    2.792
6.903
3    Speed =~   choice4           3.665   0.794   4.617  0.000    2.109
5.220
4   Memory =~      span           1.000   0.000      NA     NA    1.000
1.000
5   Memory =~    digits           4.533   1.591   2.848  0.004    1.414
7.652
6   Memory =~    blocks           1.856   0.733   2.532  0.011    0.419
3.293
7    Speed  ~      adhd      a    0.033   0.103   0.323  0.747   -0.169
0.236
8   Memory  ~      adhd      c   -0.010   0.005  -2.216  0.027   -0.020
```

```
-0.001
9    fluid  ~     Speed      b    0.037   0.135  0.272  0.786   -0.228
0.301
10   fluid  ~    Memory      d    9.752   7.491  1.302  0.193   -4.930
24.433
11   fluid  ~      adhd      e    0.035   0.080  0.437  0.662   -0.122
0.192
12   Speed ~~    Memory          -2.845   1.159 -2.454  0.014   -5.118
-0.573
13  simple ~~    simple         520.791  68.468  7.606  0.000  386.596
654.986
14 choice2 ~~   choice2        1246.433 607.127  2.053  0.040   56.485
2436.380
15 choice4 ~~   choice4        4209.853 615.042  6.845  0.000 3004.392
5415.314
16    span ~~      span           0.966   0.122  7.939  0.000    0.728
1.204
17  digits ~~    digits           6.132   0.992  6.183  0.000    4.188
8.076
18  blocks ~~    blocks           2.299   0.300  7.657  0.000    1.710
2.887
19   fluid ~~     fluid          12.292   3.786  3.247  0.001    4.871
19.713
20   Speed ~~     Speed         161.706  60.036  2.693  0.007   44.037
279.375
21  Memory ~~    Memory           0.093   0.061  1.519  0.129   -0.027
```

```
0.214
22     adhd ~~       adhd          134.304   0.000      NA      NA  134.304
134.304
23       ab :=        a*b      ab     0.001   0.006  0.208  0.835   -0.010
0.013
24       cd :=        c*d      cd    -0.102   0.083 -1.227  0.220   -0.264
0.061
25   totind :=    a*b+c*d totind    -0.101   0.078 -1.281  0.200   -0.254
0.053
26    total := e+a*b+c*d  total    -0.065   0.032 -2.056  0.040   -0.128
-0.003
```

Listed next are standardized estimates with 95% confidence intervals for all parameters and defined effects; these results are reported in Tables 1-3. Some of the absolute lower or upper bounds for results interpreted as estimated correlations or as (1 – squared correlations) in the standardized solution exceed 1.0 and, thus, are invalid. Invalid lower or upper bounds are listed next in boldface, and the corresponding bounds in Tables 1 and 3 are reported as 1.0.

```
> standardizedSolution(model)
       lhs op       rhs est.std    se       z pvalue ci.lower ci.upper
1     Speed =~    simple   0.487 0.078   6.254  0.000    0.334    0.640
2     Speed =~   choice2   0.868 0.070  12.343  0.000    0.730    1.006
3     Speed =~   choice4   0.584 0.073   7.955  0.000    0.440    0.727
```

4	Memory	=~	span	0.317	0.096	3.293	0.001	0.128	0.506
5	Memory	=~	digits	0.515	0.100	5.177	0.000	0.320	0.711
6	Memory	=~	blocks	0.373	0.096	3.900	0.000	0.186	0.561
7	Speed	~	adhd	0.030	0.094	0.324	0.746	-0.154	0.215
8	Memory	~	adhd	-0.368	0.129	-2.856	0.004	-0.620	-0.115
9	fluid	~	Speed	0.105	0.385	0.272	0.786	-0.649	0.858
10	fluid	~	Memory	0.718	0.444	1.618	0.106	-0.152	1.588
11	fluid	~	adhd	0.091	0.208	0.437	0.662	-0.317	0.499
12	Speed	~~	Memory	-0.732	0.155	-4.712	0.000	**-1.037**	-0.428
13	simple	~~	simple	0.763	0.076	10.061	0.000	0.614	0.912
14	choice2	~~	choice2	0.247	0.122	2.023	0.043	0.008	0.486
15	choice4	~~	choice4	0.659	0.086	7.703	0.000	0.492	0.827
16	span	~~	span	0.899	0.061	14.728	0.000	0.780	**1.019**
17	digits	~~	digits	0.734	0.103	7.154	0.000	0.533	0.935
18	blocks	~~	blocks	0.861	0.071	12.048	0.000	0.721	**1.001**
19	fluid	~~	fluid	0.617	0.187	3.305	0.001	0.251	0.983
20	Speed	~~	Speed	0.999	0.006	174.592	0.000	0.988	**1.010**
21	Memory	~~	Memory	0.865	0.095	9.121	0.000	0.679	**1.050**
22	adhd	~~	adhd	1.000	0.000	NA	NA	1.000	1.000
23	ab	:=	a*b	0.003	0.015	0.208	0.835	-0.027	0.033
24	cd	:=	c*d	-0.264	0.214	-1.235	0.217	-0.683	0.155
25	totind	:=	a*b+c*d	-0.261	0.202	-1.290	0.197	-0.657	0.136
26	total	:=	e+a*b+c*d	-0.170	0.081	-2.101	0.036	-0.329	-0.011

The variance-covariance matrix for the observed variables predicted by the model with its unstandardized parameter estimates.

```
> fitted(model)
$cov
         simple   choic2   choic4   span     digits   blocks   fluid
adhd
simple   682.647
choice2  784.549 5049.304
choice4  593.150 2875.121 6383.559
span       -2.892  -14.018  -10.598    1.074
digits    -13.108  -63.539  -48.038    0.490    8.351
blocks     -5.367  -26.017  -19.670    0.200    0.909    2.671
fluid     -22.103 -107.137  -81.000    0.898    4.070    1.667   19.929
adhd        4.486   21.745   16.440   -1.401   -6.351   -2.601   -8.795
134.304
```

Four types of residuals are listed next.

Raw residuals (i.e., covariance residuals) are differences between the observed and predicted variances or covariances for each pair of observed variables.

Because the measured variables do not all have a common metric, it can be difficult to interpret the meaning of covariance residuals.

For example, values of different covariance residuals, such as “13.781” (for the pair simple-choice2) and “4.083” (for the pair simple-digits) are not directly comparable because these three tasks all have different raw score metrics.

```
> residuals(model, type = "raw")
```

```
$type
[1] "raw"
$cov
        simple  choic2  choic4  span    digits  blocks  fluid   adhd
simple    0.000
choice2  13.781   0.001
choice4  12.229 -36.436   0.000
span      1.809   2.236   4.802   0.000
digits    4.803   1.936  15.714   0.319   0.000
blocks    0.671  -9.982 -12.972  -0.048  -0.342   0.000
fluid    -1.225   5.627 -22.436   0.212  -0.071  -0.207   0.000
adhd      1.570  11.195 -53.477   0.801  -1.686   0.517   0.000
0.000
```

Listed next are standardized residuals, or significance tests of the corresponding covariance (raw) residuals where the denominators are the standard errors of those differences and the whole ratio is a normal deviate (z) Some of these results, listed in boldface, are invalid because the calculated value is so large that no meaningful interpretation seems plausible.

This happens because the estimated standard errors for these results are practically zero.

```
> # some results for standardized residuals are invalid
> residuals(model, type = "standardized")
$type
[1] "standardized"
```

```
$cov
       simple     choic2     choic4      span   digits blocks fluid
adhd
simple 723402.200
choice2     0.635 337530.036
choice4     0.130     -0.899 295681.630
span        0.929      0.638      0.867  0.000
digits      0.977      0.282      1.157  1.931  0.000
blocks      0.225     -1.733     -1.479 -0.437 -1.759  0.000
fluid      -0.177      1.249     -1.205  0.971 -0.183 -0.617
0.000
adhd        0.074      0.705     -0.910  1.085 -1.212  0.432
154572529.364 0.000
```

The standard errors (denominators) for normalized residuals are those for the corresponding observed covariance, not for the difference between the observed and predicted covariances It is computationally more difficult to estimate standard errors of difference between a pair of covariances than for the standard error of one of those covariances, especially in small samples.

These results are all valid and, as expected, values of absolute normal deviates for the normalized results are generally smaller than those for the corresponding standardized residuals.

```
> residuals(model, type = "normalized")
$type
[1] "normalized"
```

```
$cov
        simple choic2 choic4 span   digits blocks fluid  adhd
simple   0.000
choice2  0.081  0.000
choice4  0.067 -0.068  0.000
span     0.795  0.357  0.689  0.000
digits   0.754  0.108  0.803  1.226  0.000
blocks   0.186 -0.978 -1.149 -0.337 -0.857  0.000
fluid   -0.123  0.201 -0.720  0.532 -0.063 -0.332  0.000
adhd     0.062  0.162 -0.689  0.795 -0.618  0.327  0.000  0.000
```

Correlation residuals, or differences between sample correlations and those predicted by the model for each pair of measured variables, are listed next.

The designation “cor.bollen” means that the sample covariance matrix and the predicted (fitted) covariance matrix were each converted to a correlation matrix based on their own respective variances (the diagonals) before residuals were calculated.

These results are reported in Table 4.

```
> residuals(model, type = "cor")
$type
[1] "cor.bollen"
$cov
        simple choic2 choic4 span   digits blocks fluid  adhd
simple   0.000
choice2  0.007  0.000
```

```
choice4  0.006 -0.006  0.000
span     0.067  0.030  0.058  0.000
digits   0.064  0.009  0.068  0.107  0.000
blocks   0.016 -0.086 -0.099 -0.028 -0.072  0.000
fluid   -0.011  0.018 -0.063  0.046 -0.005 -0.028  0.000
adhd     0.005  0.014 -0.058  0.067 -0.050  0.027  0.000  0.000
```

Estimated power for the model chi-square test of exact fit based on the population RMSEA under null and alternative hypotheses.

```
> # estimate power of chi-square test
> # null value is rmsea = 0
```

Calculation of population RMSEA for a model chi-square with df = 16 that is significant at the .05 level, which is 26.296.

```
> # alternative value is .068
> # rmsea = SQRT ((26.296 - 16)/(141 * 16)) = .068
> # where 26.296 is critical chi-square (16) at .05
```

Parameters are null RMSEA, alternative RMSEA, df, N, level of alpha, and number of groups.

```
> findRMSEApower(0, .068, 16, 142, .05, 1)
[1] 0.4663349
> # determine minimum sample size for power = .90
```

New parameter is minimum power, .90.

```
> findRMSEAsamplesize(0, .068, 16, .90, .05, 1)
[1] 328
>
```

[1]The output is from the freely available R packages lavaan (Latent Variable Analysis; version 0.6-6; Rosseel, 2012) and semTools (version 0.5-2; Jorgensen et al., 2019). Annotations (comments) in the output, designated by the symbol "#" in the first column, are original.

References

Jorgensen, T. D., Pornprasertmanit, S., Schoemann, A. M., & Rosseel, Y. (2019). *semTools: Useful tools for structural equation modeling (R package 0.5-2)*. https://CRAN.R-project.org/package=semTools

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. https://doi.org/10.18637/jss.v048.i02