ESEM in R

ESEM with lavaan is very simple. We first specify a model that looks just like the one for EFA in lavaan.

ESEM Model

```
# these are exploratory blocks (you can give them any name), you'll need to specify them
esem_model <- 'efa("block1")*F1 =~ newhabit01 + newhabit02 + newhabit03 + newhabit04 +
                                   newhabit05 + newhabit06 + newhabit07 + newhabit08 +
                                   newhabit09
               efa("block1")*F2 =~ newhabit01 + newhabit02 + newhabit03 + newhabit04 +
                                   newhabit05 + newhabit06 + newhabit07 + newhabit08 +
                                   newhabit09'
```

The difference in the code is here. When fitting the model, we need to use `rotation = geomin` because `geomin` does not assume uncorrelated residuals. Like other oblique estimators, geomin allows latent factors to covary.

Note: In addition to changing the estimator, we should tell lavaan to change the epsilon value so that it overrides its default of 0.001. This is important: Changing epsilon allows to change the size of the cross-loadings or of the factor correlations. The lower the epsilon, the greater the inter-factor correlations, the lower the cross-loadings, and the less accurate the associations between the constructs (Morin et al., in press). To optimize the estimate of the inter-factor correlations, Morin et al. recommend epsilon = .5.

Important: By default, lavaan no longer allows changing the epsilon to values above .01, so the results will differ from those in Mplus and recommended values when using geomin. lavaan directly from GitHub may solve this, as their defaults are up to 1. To find lavaan defaults and how to change values: https://github.com/yrosseel/lavaan/blob/master/R/lav_options.R

We have missing data so we're telling lavaan to use FIML to estimate missing values and we're using MLR as our estimator.

```
esemfit <- cfa(model = esem_model,
                 data = habit,
                 rotation = "geomin",
                 rotation.args = list(geomin.epsilon = 0.5),
                 estimator = "MLR",
                 missing = "FIML")
```

```
lavaan::summary(esemfit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-20.2265 ended normally after 35 iterations
##
```

```
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        37
##   Row rank of the constraints matrix                 2
##
##   Rotation method                       GEOMIN OBLIQUE
##   Geomin epsilon                                   0.5
##   Rotation algorithm (rstarts)                GPA (30)
##   Standardized metric                             TRUE
##   Row weights                                     None
##
##                                                   Used       Total
##   Number of observations                           430         432
##   Number of missing patterns                        40
##
## Model Test User Model:
##                                               Standard      Scaled
##   Test Statistic                                79.099      60.953
##   Degrees of freedom                                19          19
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.298
##     Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                              1554.087    1043.454
##   Degrees of freedom                                36          36
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.489
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.960       0.958
##   Tucker-Lewis Index (TLI)                       0.925       0.921
##
##   Robust Comparative Fit Index (CFI)                         0.965
##   Robust Tucker-Lewis Index (TLI)                            0.933
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -5225.008   -5225.008
##   Scaling correction factor                                  1.231
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -5185.458   -5185.458
##   Scaling correction factor                                  1.254
##       for the MLR correction
##
##   Akaike (AIC)                               10520.015   10520.015
##   Bayesian (BIC)                             10662.248   10662.248
##   Sample-size adjusted Bayesian (SABIC)      10551.178   10551.178
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.086       0.072
```

```
##   90 Percent confidence interval - lower         0.067       0.054
##   90 Percent confidence interval - upper         0.106       0.090
##   P-value H_0: RMSEA <= 0.050                    0.001       0.021
##   P-value H_0: RMSEA >= 0.080                    0.706       0.234
##
##   Robust RMSEA                                               0.092
##   90 Percent confidence interval - lower                     0.062
##   90 Percent confidence interval - upper                     0.122
##   P-value H_0: Robust RMSEA <= 0.050                         0.011
##   P-value H_0: Robust RMSEA >= 0.080                         0.762
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.031       0.031
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 =~ block1
##     newhabit01        0.838    0.092    9.063    0.000    0.838    0.681
##     newhabit02        0.989    0.088   11.272    0.000    0.989    0.769
##     newhabit03        0.656    0.087    7.553    0.000    0.656    0.540
##     newhabit04        0.709    0.116    6.097    0.000    0.709    0.549
##     newhabit05        0.577    0.116    4.957    0.000    0.577    0.489
##     newhabit06        0.419    0.122    3.446    0.001    0.419    0.376
##     newhabit07        0.173    0.095    1.808    0.071    0.173    0.135
##     newhabit08        0.016    0.087    0.180    0.857    0.016    0.011
##     newhabit09        0.131    0.093    1.403    0.160    0.131    0.102
##   F2 =~ block1
##     newhabit01        0.138    0.091    1.520    0.128    0.138    0.112
##     newhabit02        0.167    0.080    2.096    0.036    0.167    0.130
##     newhabit03        0.173    0.086    2.017    0.044    0.173    0.142
##     newhabit04        0.267    0.111    2.396    0.017    0.267    0.207
##     newhabit05        0.331    0.115    2.870    0.004    0.331    0.280
##     newhabit06        0.397    0.127    3.122    0.002    0.397    0.357
##     newhabit07        0.807    0.109    7.436    0.000    0.807    0.634
##     newhabit08        1.021    0.108    9.463    0.000    1.021    0.732
##     newhabit09        0.925    0.115    8.013    0.000    0.925    0.717
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 ~~
##     F2                0.553    0.038   14.510    0.000    0.553    0.553
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .newhabit01        3.405    0.062   55.325    0.000    3.405    2.768
##    .newhabit02        3.230    0.063   51.352    0.000    3.230    2.509
##    .newhabit03        2.907    0.059   49.072    0.000    2.907    2.393
```

```
##    .newhabit04        3.083    0.063   49.088    0.000    3.083    2.388
##    .newhabit05        3.263    0.057   56.876    0.000    3.263    2.764
##    .newhabit06        3.533    0.055   64.663    0.000    3.533    3.175
##    .newhabit07        3.515    0.064   54.618    0.000    3.515    2.759
##    .newhabit08        3.275    0.070   46.550    0.000    3.275    2.347
##    .newhabit09        3.213    0.064   50.490    0.000    3.213    2.492
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .newhabit01        0.664    0.094    7.066    0.000    0.664    0.439
##    .newhabit02        0.468    0.085    5.525    0.000    0.468    0.282
##    .newhabit03        0.891    0.092    9.685    0.000    0.891    0.603
##    .newhabit04        0.885    0.095    9.353    0.000    0.885    0.531
##    .newhabit05        0.740    0.068   10.945    0.000    0.740    0.531
##    .newhabit06        0.721    0.072   10.024    0.000    0.721    0.583
##    .newhabit07        0.788    0.100    7.857    0.000    0.788    0.485
##    .newhabit08        0.886    0.139    6.396    0.000    0.886    0.455
##    .newhabit09        0.655    0.130    5.041    0.000    0.655    0.394
##     F1                1.000                               1.000    1.000
##     F2                1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     newhabit01        0.561
##     newhabit02        0.718
##     newhabit03        0.397
##     newhabit04        0.469
##     newhabit05        0.469
##     newhabit06        0.417
##     newhabit07        0.515
##     newhabit08        0.545
##     newhabit09        0.606
```

Notice that the correlation between the two latent variables is much lower in the ESEM model ($r = .55$) than in the EFA model ($r = .71$), and in the CFA model ($r = .78$) which makes sense given our use of geomin and specification of the value of epsilon. In the context of ESEM, it also makes sense that correlations would be lower in the ESEM model, as not allowing cross-loadings leads to inflation of the correlation between latent factors.

Cross-loadings are fairly small in the ESEM solution. They range from .01-.14 for the first factor (and are all non-significant), and .11-36 for the second factor (all but one are significant). The factors are fairly well-defined in the ESEM solution: the loadings on f1 range from .38-.77 (M = .57), and on f2 range from .63-.73 (M = .69). Model fit is good, but it is poorer than that of the Bifactor CFA model when we consider the TLI and RMSEA (which tax low parsimony).

Bifactor ESEM

To estimate a bifactor ESEM model, just add a general factor to the model with the same specifications as the two specific factors.

```
# these are exploratory blocks (you can give them any name), you'll need to specify them

biesem_model <- 'efa("block1")*F1 =~ newhabit01 + newhabit02 + newhabit03 + newhabit04 +
```

```
                                  newhabit05 + newhabit06 + newhabit07 + newhabit08 +
                                  newhabit09
                 efa("block1")*F2 =~ newhabit01 + newhabit02 + newhabit03 + newhabit04 +
                                  newhabit05 + newhabit06 + newhabit07 + newhabit08 +
                                  newhabit09
                efa("block1")*g =~ newhabit01 + newhabit02 + newhabit03 + newhabit04 +
                                  newhabit05 + newhabit06 + newhabit07 + newhabit08 +
                                  newhabit09'
```

You'll need to change the rotation type to "bigeomin", which is one of the recommended rotations for bifactor ESEM (also what we're using in Mplus demo).

```
biesemfit <- lavaan::cfa(model = biesem_model,
                  data = habit,
                  rotation = "bigeomin",
                  rotation.args = list(geomin.epsilon = .5),
                  estimator = "MLR",
                  missing = "FIML")
```

Compare the output with the one from Mplus. You'll notice that the standardized solution here is very similar to that in Mplus (under STDYX), but not exactly the same. This is likely because Mplus is using BI-GEOMIN orthogonal, in which the specific factors are uncorrelated with the general factor and with each other. Notice that lavaan is using "BIGEOMIN OBLIQUE" and that covariances/correlations are indeed non-zero (even if very small). Setting the factors to be orthogonal does not change lavaan's default to let the LV correlate.

```
lavaan::summary(biesemfit,
        fit.measures = TRUE,
        standardized = TRUE,
        rsquare = TRUE)
```

```
## lavaan 0.6-20.2265 ended normally after 49 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        48
##   Row rank of the constraints matrix                 6
##
##   Rotation method                      BIGEOMIN OBLIQUE
##   Rotation algorithm (rstarts)                GPA (30)
##   Standardized metric                             TRUE
##   Row weights                                     None
##
##                                                   Used       Total
##   Number of observations                           430         432
##   Number of missing patterns                        40
##
## Model Test User Model:
##                                               Standard      Scaled
##   Test Statistic                                34.932      44.017
##   Degrees of freedom                                12          12
##   P-value (Chi-square)                           0.000       0.000
```

```
##   Scaling correction factor                                  0.794
##     Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                            1554.087    1043.454
##   Degrees of freedom                              36          36
##   P-value                                      0.000       0.000
##   Scaling correction factor                                1.489
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                  0.985       0.968
##   Tucker-Lewis Index (TLI)                     0.955       0.905
##
##   Robust Comparative Fit Index (CFI)                       0.997
##   Robust Tucker-Lewis Index (TLI)                          0.991
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -5202.924   -5202.924
##   Scaling correction factor                                1.386
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)    -5185.458   -5185.458
##   Scaling correction factor                                1.254
##       for the MLR correction
##
##   Akaike (AIC)                             10489.848   10489.848
##   Bayesian (BIC)                           10660.527   10660.527
##   Sample-size adjusted Bayesian (SABIC)    10527.244   10527.244
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                        0.067       0.079
##   90 Percent confidence interval - lower       0.042       0.052
##   90 Percent confidence interval - upper       0.093       0.108
##   P-value H_0: RMSEA <= 0.050                  0.128       0.040
##   P-value H_0: RMSEA >= 0.080                  0.217       0.506
##
##   Robust RMSEA                                             0.096
##   90 Percent confidence interval - lower                   0.025
##   90 Percent confidence interval - upper                   0.165
##   P-value H_0: Robust RMSEA <= 0.050                       0.114
##   P-value H_0: Robust RMSEA >= 0.080                       0.703
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                         0.021       0.021
##
## Parameter Estimates:
##
##   Standard errors                           Sandwich
##   Information bread                         Observed
##   Observed information based on              Hessian
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 =~ block1
##     newhabit01        0.853    0.060   14.198    0.000    0.853    0.694
##     newhabit02        1.054    0.044   23.823    0.000    1.054    0.819
##     newhabit03        0.721    0.062   11.626    0.000    0.721    0.594
##     newhabit04        0.856    0.054   15.750    0.000    0.856    0.663
##     newhabit05        0.872    0.127    6.866    0.000    0.872    0.739
##     newhabit06        0.719    0.072   10.008    0.000    0.719    0.646
##     newhabit07        0.843    0.065   12.899    0.000    0.843    0.662
##     newhabit08        0.895    0.068   13.225    0.000    0.895    0.642
##     newhabit09        0.921    0.060   15.260    0.000    0.921    0.714
##   F2 =~ block1
##     newhabit01       -0.273    0.078   -3.498    0.000   -0.273   -0.222
##     newhabit02       -0.498    0.321   -1.550    0.121   -0.498   -0.387
##     newhabit03       -0.158    0.096   -1.651    0.099   -0.158   -0.130
##     newhabit04       -0.187    0.089   -2.098    0.036   -0.187   -0.145
##     newhabit05        0.012    0.233    0.052    0.958    0.012    0.010
##     newhabit06        0.075    0.131    0.571    0.568    0.075    0.067
##     newhabit07        0.318    0.131    2.427    0.015    0.318    0.250
##     newhabit08        0.463    0.213    2.177    0.029    0.463    0.332
##     newhabit09        0.372    0.177    2.099    0.036    0.372    0.289
##   g =~ block1
##     newhabit01        0.038    0.276    0.137    0.891    0.038    0.031
##     newhabit02       -0.208    0.092   -2.251    0.024   -0.208   -0.161
##     newhabit03        0.015    0.254    0.059    0.953    0.015    0.012
##     newhabit04       -0.120    0.211   -0.568    0.570   -0.120   -0.093
##     newhabit05        0.671    0.766    0.876    0.381    0.671    0.568
##     newhabit06        0.205    0.274    0.749    0.454    0.205    0.184
##     newhabit07       -0.098    0.088   -1.120    0.263   -0.098   -0.077
##     newhabit08       -0.175    0.277   -0.631    0.528   -0.175   -0.125
##     newhabit09       -0.210    0.327   -0.643    0.521   -0.210   -0.163
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 ~~
##     F2                0.000    0.000    6.993    0.000    0.000    0.000
##     g                 0.000    0.000    1.323    0.186    0.000    0.000
##   F2 ~~
##     g                -0.052    0.087   -0.601    0.548   -0.052   -0.052
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .newhabit01        3.406    0.061   55.406    0.000    3.406    2.770
##    .newhabit02        3.229    0.063   51.245    0.000    3.229    2.510
##    .newhabit03        2.907    0.059   49.009    0.000    2.907    2.392
##    .newhabit04        3.082    0.063   48.941    0.000    3.082    2.386
##    .newhabit05        3.266    0.058   56.795    0.000    3.266    2.767
##    .newhabit06        3.537    0.055   64.780    0.000    3.537    3.176
##    .newhabit07        3.515    0.064   54.571    0.000    3.515    2.759
##    .newhabit08        3.273    0.070   46.511    0.000    3.273    2.346
##    .newhabit09        3.213    0.064   50.476    0.000    3.213    2.492
##
```

```
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .newhabit01        0.707    0.084    8.444    0.000    0.707    0.468
##    .newhabit02        0.265    0.306    0.866    0.387    0.265    0.160
##    .newhabit03        0.931    0.097    9.609    0.000    0.931    0.631
##    .newhabit04        0.889    0.108    8.225    0.000    0.889    0.533
##    .newhabit05        0.184    1.248    0.147    0.883    0.184    0.132
##    .newhabit06        0.676    0.193    3.503    0.000    0.676    0.546
##    .newhabit07        0.798    0.103    7.787    0.000    0.798    0.492
##    .newhabit08        0.892    0.134    6.680    0.000    0.892    0.458
##    .newhabit09        0.623    0.154    4.045    0.000    0.623    0.375
##     F1                1.000                               1.000    1.000
##     F2                1.000                               1.000    1.000
##     g                 1.000                               1.000    1.000
##
## R-Square:
##                    Estimate
##     newhabit01        0.532
##     newhabit02        0.840
##     newhabit03        0.369
##     newhabit04        0.467
##     newhabit05        0.868
##     newhabit06        0.454
##     newhabit07        0.508
##     newhabit08        0.542
##     newhabit09        0.625
```

Model	df	CFI	TLI	RMSEA	SRMR
CFA: One factor	36	0.90	0.87	0.09	0.05
CFA: Two correlated factors	26	0.96	0.94	0.08	0.04
CFA: One factor plus method factor	24	0.96	0.94	0.08	0.04
CFA: second order	27	0.95	0.93	0.09	0.13
Bifactor	18	0.98	0.97	0.05	0.02
ESEM-geomin	19	0.97	0.93	0.07	0.03

So, is the bifactor model the best fit? Does this mean we can treat this measure as a single score? Bifactor models can accommodate patterns of association in the data that are not accounted for when we use correlated-factor models. For example, in this measure, it would be unreasonable to assume that the last three items are completely unrelated to the first six - there's conceptual overlap between them and other items - but the correlated-factors model does not account for this, resulting in lower fit relative to the bifactor and ESEM models.

This was a simple example, as what we're talking about (wording effects) is likely construct-irrelevant variance, or a methodological artifact. The purpose of this exercise - including both scripts - was to demonstrate the use of these tools to examine measure dimensionality. In this case, to determine whether it is tenable to use this measure as a composite index in an analysis.

Importantly, *how* one treats the structure of a construct depends on more than fit. As you've read on Chapter 18 of the Handbook, better fit does **not** mean that the model is an accurate representation of the measurement structure for everyone (or even for most people in the sample) or an accurate representation of the "true" latent structure itself. This doesn't mean that these models aren't useful, but that they can help understand the sources of item variance and inform whether a measurement structure is tenable and interpretable/valid. These decisions must be made with justifications beyond statistical tests and fit coefficients.