An introduction to Psychometric Theory with applications in R
Structural Equation Modeling and applied scale construction

William Revelle

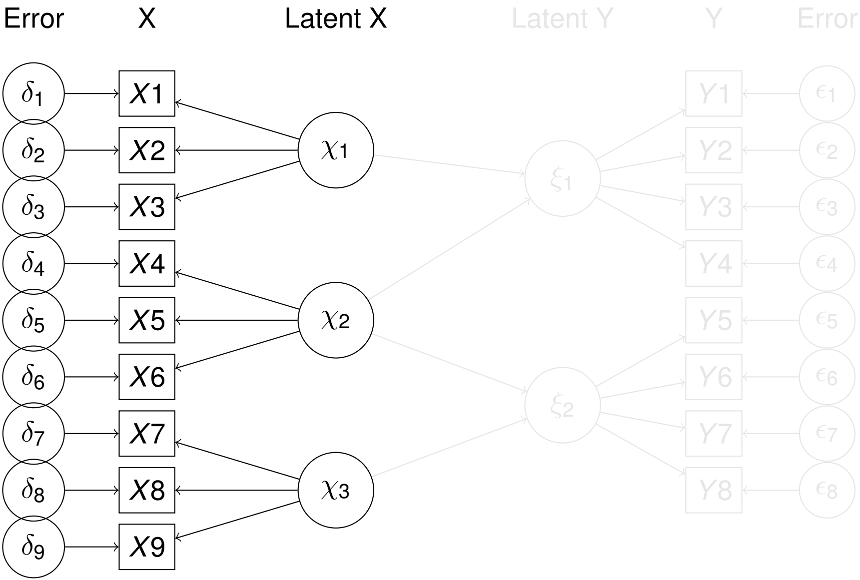
Department of Psychology
Northwestern University
Evanston, Illinois USA

NORTHWESTERN UNIVERSITY

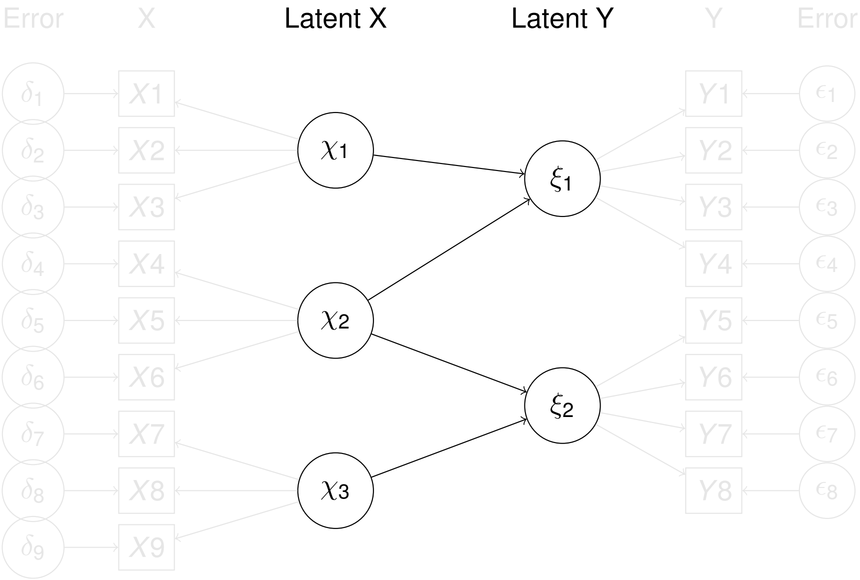
May, 2025

Outline

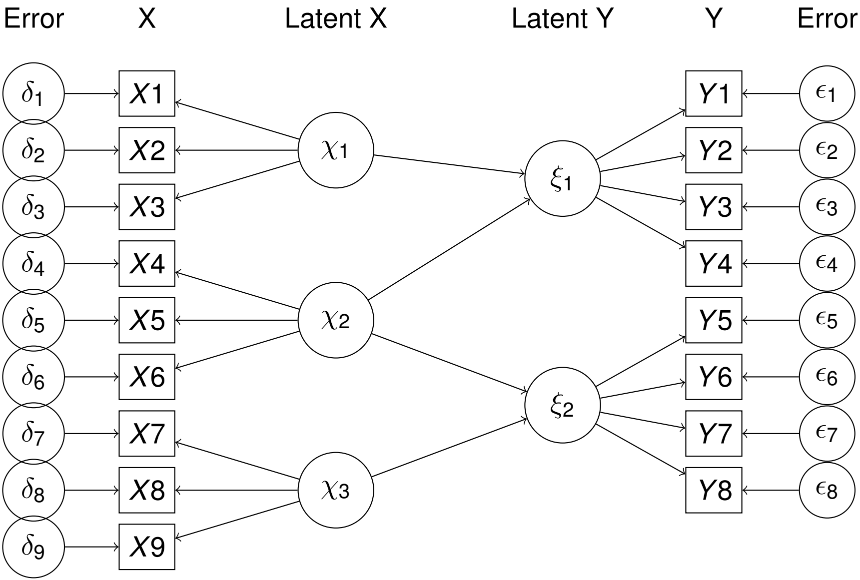
Measurement: A latent variable approach.



Theory


Error
X
Latent X
Latent Y
Y
Error
δ_1
δ_2
δ_3
δ_4
δ_5
δ_6
δ_7
δ_8
δ_9
X1
X2
X3
X4
X5
X6
X7
X8
X9
χ_1
χ_2
χ_3
ξ_1
ξ_2
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
ϵ_1
ϵ_2
ϵ_3
ϵ_4
ϵ_5
ϵ_6
ϵ_7
ϵ_8

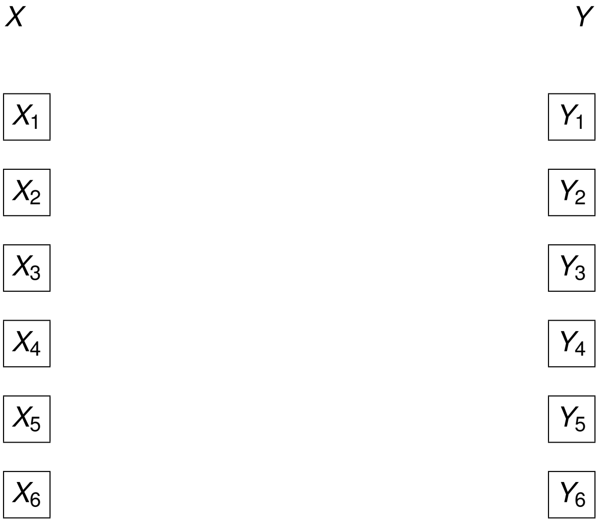
Psychometric Theory: A conceptual Syllabus



Two types of variables, three types of relationships

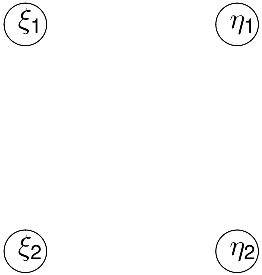
1. Variables
   1.1 Observed Variables (X, Y)
   1.2 Latent Variables (ξ η ϵ ζ)
2. Three kinds of variance/covariances
   2.1 Observed with Observed C_{xy} or σ_{xy}
   2.2 Observed with Latent λ
   2.3 Latent with Latent ϕ
3. Direction
   - Bidirectional (correlation)
   - Directional (regression)

Observed Variables



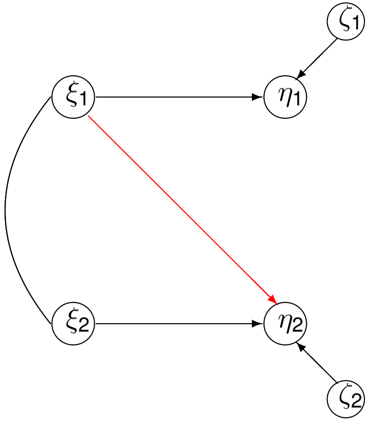
Latent Variables

ξ η

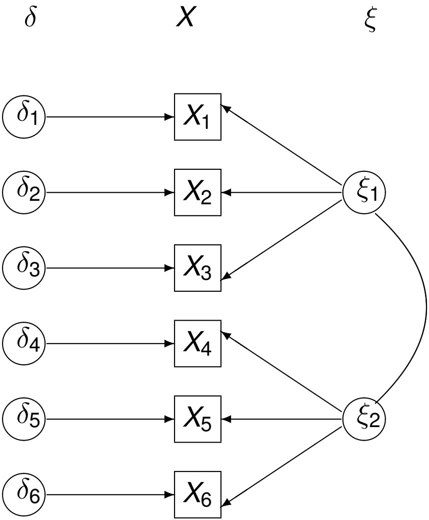


Theory: A regression model of latent variables

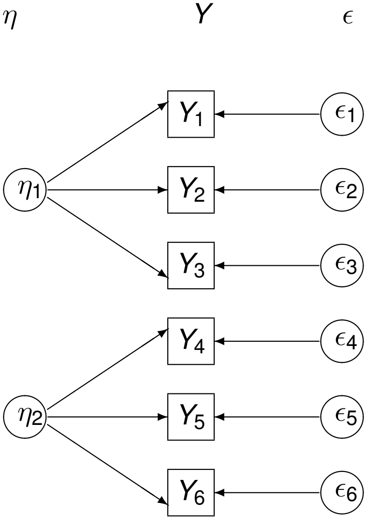
ξ η



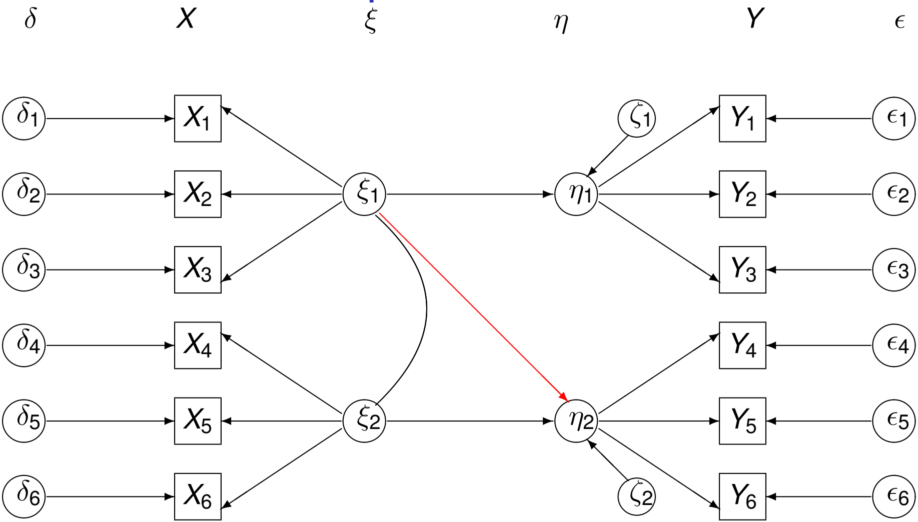
A measurement model for X



A measurement model for Y



A complete structural model

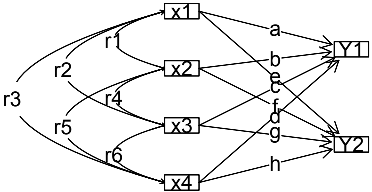


Latent Variable Modeling

1. Requires measuring observed variables
   - Requires defining what is relevant and irrelevant to our theory.
   - Issues in quality of scale information, levels of measurement.
2. Formulating a measurement model of the data: estimating latent constructs
   - Perhaps based upon exploratory and then confirmatory factor analysis, definitely based upon theory.
   - Includes understanding the reliability of the measures.
3. Modeling the structure of the constructs
   - This is a combination of theory and fitting. Do the data fit the theory.
   - Comparison of models. Does one model fit better than alternative models?

The generalized regression model

Generalized regression model



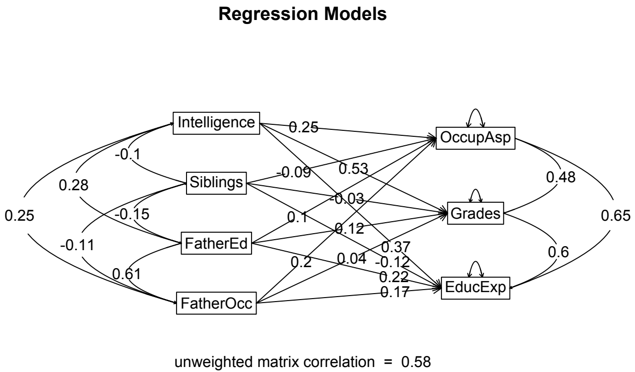
$\hat{Y} = \beta_x x + \epsilon$

The Kerchoff data set of educational outcomes

```
lowerMat(R.kerch)
              Intll Sblng FthrE FthrO Grads EdcEx OccpA
Intelligence   1.00
Siblings      -0.10  1.00
FatherEd       0.28 -0.15  1.00
FatherOcc      0.25 -0.11  0.61  1.00
Grades         0.57 -0.10  0.29  0.25  1.00
EducExp        0.49 -0.21  0.45  0.41  0.60  1.00
OccupAsp       0.34 -0.15  0.30  0.33  0.48  0.65  1.00
```

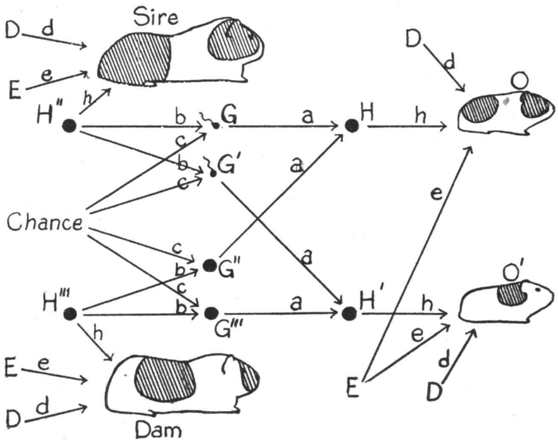
Kerckhoff (1974) (From LISREL manual, available in help page to `mediate`)

The classic regression model

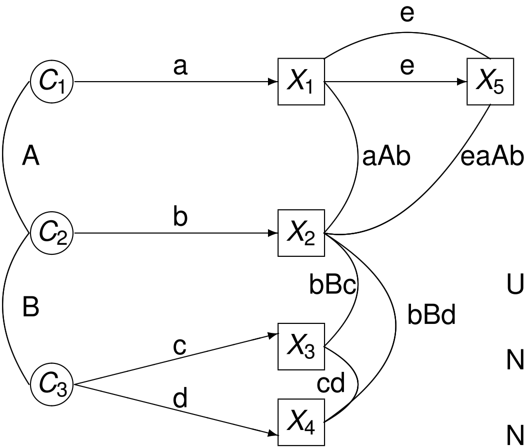


$$\hat{Y} = \beta_x x + \epsilon$$

Wright's Path model of inheritance in the Guinea Pig (Wright, 1921)



The basic rules of path analysis–think genetics



Up ... and over and down ...

No down and up

No double overs

Up ... and down ...

Parents cause children

children do not cause parents

Observed-observed

1. The Kerckhoff (1974) data set is used in the LISREL manual as an example of regression path models
   - "A sample of boys originally studied as ninth graders in 1969 was recontacted in 1974 to obtain information about high school performance and educational attainment."
   - 767 twelfth grade males
   - A study of background, aspiration and educational attainment
2. Variables
   - Intelligence
   - Number of siblings
   - Fathers Education
   - Fathers's occupation
   - Grades
   - Educational expectation
   - Occupational aspiration
3. Correlation matrix is available in the sem (Fox, Nie & Byrnes, 2013) package and the help page to `mediate`

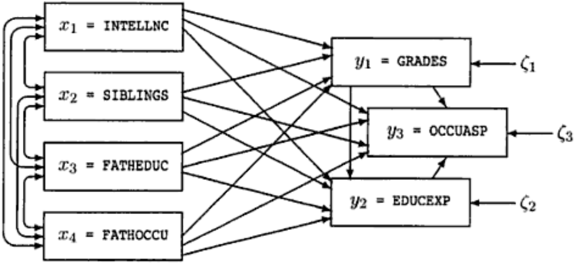
The Kerckhoff correlation matrix (N=767)

```
> R.kerch
             Intelligence Siblings FatherEd FatherOcc Grades EducExp OccupAsp
Intelligence        1.000   -0.100    0.277     0.250  0.572   0.489    0.335
Siblings           -0.100    1.000   -0.152    -0.108 -0.105  -0.213   -0.153
FatherEd            0.277   -0.152    1.000     0.611  0.294   0.446    0.303
FatherOcc           0.250   -0.108    0.611     1.000  0.248   0.410    0.331
Grades              0.572   -0.105    0.294     0.248  1.000   0.597    0.478
EducExp             0.489   -0.213    0.446     0.410  0.597   1.000    0.651
OccupAsp            0.335   -0.153    0.303     0.331  0.478   0.651    1.000
>
```

Conceptual Kerckhoff model

1. Background variables
   - Intelligence
   - Number of siblings
   - Fathers Education
   - Fathers's occupation
2. Intermediate variables
   - Grades
   - Educational expectation
3. Final outcomes
   - Occupational aspiration

The model (From the LISREL manual)



Matrix regression

1. Most regression examples (and functions) use raw data
   - $\hat{Y} = X\beta + \epsilon$
   - $beta = (X'X)^{-1}X'Y$
   - lm(y $\sim$ x , data = data)
2. Regression is just solving the matrix equation
   - $\beta = R^{-1}r_{xy}$
   - mat.regress(R,x,y) (deprecated)
   - `lmCor`(y $\sim$ x, data =R) (recommended)
3. `setCor` will work on raw data as well.

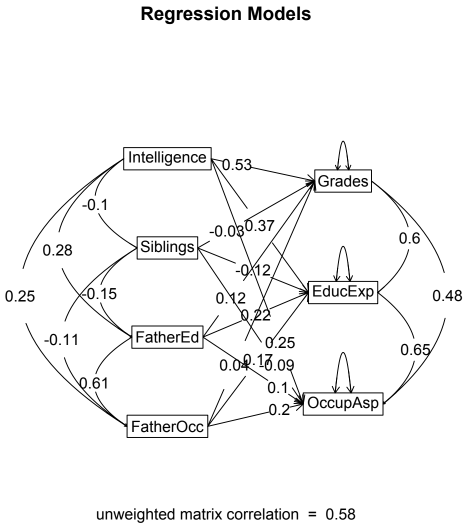
Simple regression 4 predictors, 3 criteria

R code

```
model <- lmCor(Grades + EducExp + OccupAsp ~ Intelligence + Siblings +
                      FatherEd + FatherOcc,data=R.kerch)
summary(model) #to give concise output
```

```
Multiple Regression from matrix input
cx
Beta weights
              Grades EducExp OccupAsp
Intelligence    0.53    0.37     0.25
Siblings       -0.03   -0.12    -0.09
FatherEd        0.12    0.22     0.10
FatherOcc       0.04    0.17     0.20

Multiple R
  Grades  EducExp OccupAsp
    0.59     0.61     0.44
multiple R2
  Grades  EducExp OccupAsp
    0.35     0.38     0.19
```

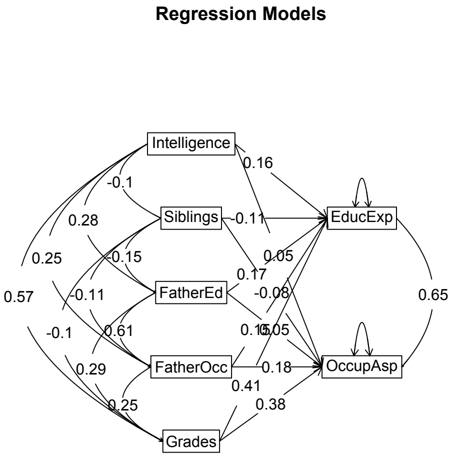
The paths (From lmCor)



More complicated regression

```
> lmCor(y=6:7,x=1:5,data=R.kerch)
#easier for long names

Call: lmtCor(y = 6:7, x = 1:5, data = R.kerch)

Multiple Regression from matrix input

Beta weights
             EducExp  OccupAsp
Intelligence    0.16      0.05
Siblings       -0.11     -0.08
FatherEd        0.17      0.05
FatherOcc       0.15      0.18
Grades          0.41      0.38

Multiple R
 EducExp OccupAsp
    0.70     0.54
multiple R2
 EducExp OccupAsp
    0.48     0.29
```

The paths (From `setCor`)



unweighted matrix correlation = 0.64

Predicting occupational aspirations from the intermediate set

```
lmCor( OccupAsp ~  Grades + EducExp  ,data=R.kerch)

Call: lmtCor(y = OccupAsp ~ Grades + EducExp, data = R.kerch)

Multiple Regression from matrix input

 DV =  OccupAsp
         slope  VIF
Grades     0.14 1.55
EducExp    0.57 1.55

 Multiple Regression
              R    R2   Ruw  R2uw
OccupAsp 0.66 0.44 0.63  0.4
```

Predicting occupational aspirations from the entire set

```
> lmCor(y=7,x=1:6,data=R.kerch)


Call:lmCor(y = 7, x = 1:6, data = R.kerch)

Multiple Regression from matrix input

 DV =  OccupAsp
             slope  VIF
Intelligence -0.04 1.57
Siblings     -0.02 1.05
FatherEd     -0.04 1.73
FatherOcc     0.10 1.66
Grades        0.16 1.85
EducExp       0.55 1.94

 Multiple Regression
            R   R2  Ruw R2uw
OccupAsp 0.67 0.44 0.57 0.32
```

We can also examine mediation using regression (and the `mediate` function).

```
mediate(EducExp ~ FatherOcc + (Intelligence) + (Siblings) +
      (FatherEd),data=R.kerch,n.obs=767)
```

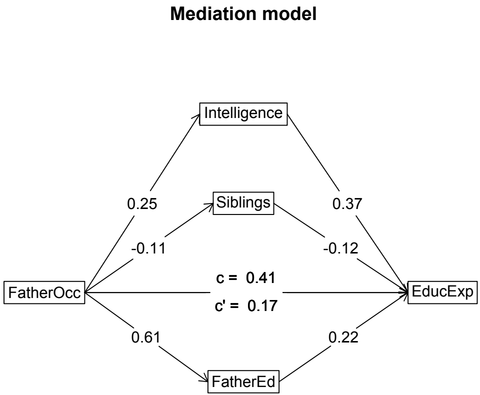
```
> mediate(6,x=4,m=1:3,data=R.kerch,n.obs=767)
TThe replication data matrices were simulated based upon the
specified number of subjects and the observed correlation matrix.

Mediation/Moderation Analysis
Call: mediate(y = EducExp ~ FatherOcc + (Intelligence) + (Siblings) +
    (FatherEd), data = R.kerch, n.obs = 767)

The DV (Y) was  EducExp . The IV (X) was  FatherOcc . The mediating variab
Intelligence Siblings FatherEd .

Total effect(c) of  FatherOcc  on  EducExp  =  0.41   S.E. =  0.03
t  =  12.43  df=  762   with p =  1.9e-32
Direct effect (c')␣of␣␣FatherOcc␣␣on␣␣EducExp␣␣removing␣␣Intelligence␣Sibl
Indirect␣effect␣(ab)␣of␣␣FatherOcc␣␣on␣␣EducExp␣␣through␣␣Intelligence␣Sib
Mean␣bootstrapped␣indirect␣effect␣=␣␣0.26␣␣with␣standard␣error␣=␣␣0.03␣␣Lo
R␣=␣0.61␣R2␣=␣0.38␣␣␣F␣=␣115.03␣on␣4␣and␣762␣DF␣␣␣p-value:␣␣9.89e-77

␣To␣see␣the␣longer␣output,␣specify␣short␣=␣FALSE␣in␣the␣print␣statement␣or
```

R code

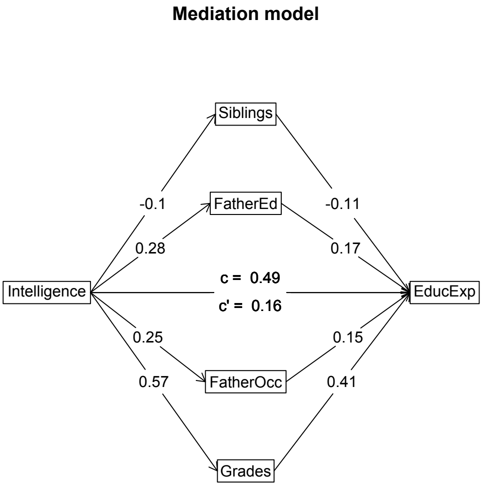
```
summary(mod)
```

```
Call: mediate(y = EducExp ~ FatherOcc + (Intelligence) + (Siblings) +
    (FatherEd), data = R.kerch, n.obs = 767)

 Total effect estimates (c)
          EducExp   se     t  df     Prob
FatherOcc    0.41 0.03 12.43 762 1.92e-32

Direct effect estimates     (c')
             EducExp   se     t  df     Prob
FatherOcc       0.17 0.04  4.63 762 4.28e-06
Intelligence    0.37 0.03 12.45 762 1.64e-32
Siblings       -0.12 0.03 -4.27 762 2.17e-05
FatherEd        0.22 0.04  6.00 762 2.99e-09

R = 0.61 R2 = 0.38   F = 115.03 on 4 and 762 DF   p-value:  9.89e-77

 'a'  effect estimates
             FatherOcc   se     t  df     Prob
Intelligence      0.25 0.04  7.14 765 2.15e-12
Siblings         -0.11 0.04 -3.00 765 2.75e-03
FatherEd          0.61 0.03 21.35 765 1.10e-79

 'b'  effect estimates
             EducExp   se     t  df     Prob
Intelligence    0.37 0.03 12.45 762 1.64e-32
Siblings       -0.12 0.03 -4.27 762 2.17e-05
FatherEd        0.22 0.04  6.00 762 2.99e-09

 'ab'  effect estimates
          EducExp boot   sd lower upper
FatherOcc    0.24 0.22 0.03  0.17  0.27

 'ab' effects estimates for each mediator
          Intelligence boot   sd lower upper
```

The mediated paths (From `mediate`)



An alternative mediated model (From `mediate`)



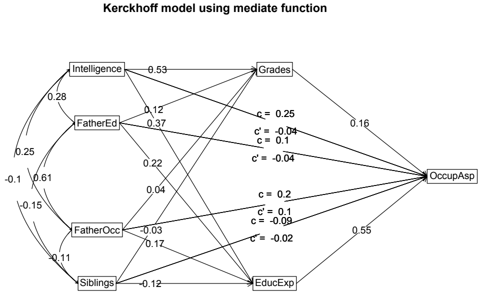
Modeling the Kerchhoff data using mediate

R code

```
model <- mediate(OccupAsp ~ Intelligence + Siblings +
        FatherEd + FatherOcc + (Grades) + (EducExp), data=R.kerch,n.ol
 mediate.diagram(model, main="Kerchhoff model using mediate function")
```

```
Direct effect estimates (traditional regression)    (c')
             OccupAsp   se     t  df     Prob
Intercept        0.00 0.03  0.00 760 1.00e+00
Intelligence    -0.04 0.03 -1.16 760 2.46e-01
Siblings        -0.02 0.03 -0.68 760 4.98e-01
FatherEd        -0.04 0.04 -1.16 760 2.47e-01
FatherOcc        0.10 0.03  2.85 760 4.43e-03
Grades           0.16 0.04  4.29 760 2.06e-05
EducExp          0.55 0.04 14.60 760 1.01e-42

R = 0.67 R2 = 0.44   F = 100.9 on 6 and 760 DF   p-value:  2.95e-93

 Total effect estimates (c)
             OccupAsp   se     t  df     Prob
Intercept        0.00 0.03  0.00 762 1.00e+00
Intelligence     0.25 0.03  7.29 762 7.63e-13
Siblings        -0.09 0.03 -2.78 762 5.60e-03
FatherEd         0.10 0.04  2.36 762 1.84e-02
FatherOcc        0.20 0.04  4.80 762 1.90e-06
```

The full model (without one path)



Can use sem functions (in either sem or lavaan) to estimate the mediation model

1. Treat all variables as observed (fixed)
   - Specify a limited number of paths rather than the full regression model
2. *sem* (Fox et al., 2013)commands
   - either in RAM (path) notation or
   - causal notation
3. *lavaan*(Rosseel, 2012) commands similar to causal notation of sem and the commercial programs Mplus and EQS
   - Output can mimic either MPlus or LISREL
   - Commands can be translated directly from lavaan to MPlus

Kerckhoff-Kenny path analysis (modified from sem help page) to just predict the DVs)

```
model.kerch <- specifyModel()
    Intelligence -> Grades,        gam51
    Siblings -> Grades,            gam52
    FatherEd -> Grades,            gam53
    FatherOcc -> Grades,           gam54
    Intelligence -> EducExp,       gam61
    Siblings -> EducExp,           gam62
    FatherEd -> EducExp,           gam63
    FatherOcc -> EducExp,          gam64
     Intelligence -> OccupAsp,      gam71
    Siblings -> OccupAsp,          gam72
    FatherEd -> OccupAsp,          gam73
    FatherOcc -> OccupAsp,         gam74
  # Grades -> EducExp, beta65
   # Grades -> OccupAsp, beta75
 # EducExp -> OccupAsp, beta76

sem.kerch <- sem(model.kerch, R.kerch, 737, fixed.x=c('Intelligence','Sibl
    'FatherEd','FatherOcc'))
summary(sem.kerch)
```

Fixed path model – not modeling the DV correlations

```
Model Chisquare =  411.72   Df =  3 Pr(>Chisq) = 6.4133e-89
Chisquare (null model) =  1664.3   Df =  21
Goodness-of-fit index =  0.85747
Adjusted goodness-of-fit index =  -0.33031
RMSEA index =  0.43024   90% CI: (0.39572, 0.46581)
Bentler-Bonnett NFI =  0.75262
Tucker-Lewis NNFI =  -0.74103
Bentler CFI =  0.75128
SRMR =  0.099759
AIC =  441.72
AICc =  412.38
BIC =  510.76
CAIC =  388.91

Normalized Residuals
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000   0.000   0.000   0.956   0.000  10.100

R-square for Endogenous Variables
 Grades  EducExp OccupAsp
 0.3490   0.3765   0.1930
```

With path coefficients of

```
Parameter Estimates
              Estimate  Std Error z value  Pr(>|z|)
gam51          0.525902 0.031182  16.86530 8.0987e-64 Grades <--- Intelligence
gam52         -0.029942 0.030149  -0.99314 3.2064e-01 Grades <--- Siblings
gam53          0.118966 0.038259   3.10951 1.8740e-03 Grades <--- FatherEd
gam54          0.040603 0.037785   1.07456 2.8257e-01 Grades <--- FatherOcc
gam61          0.373339 0.030517  12.23376 2.0521e-34 EducExp <--- Intelligence
gam62         -0.123910 0.029506  -4.19954 2.6745e-05 EducExp <--- Siblings
gam63          0.220918 0.037442   5.90022 3.6302e-09 EducExp <--- FatherEd
gam64          0.168302 0.036979   4.55125 5.3328e-06 EducExp <--- FatherOcc
gam71          0.248827 0.034718   7.16718 7.6561e-13 OccupAsp <--- Intelligence
gam72         -0.091653 0.033567  -2.73047 6.3245e-03 OccupAsp <--- Siblings
gam73          0.098869 0.042596   2.32109 2.0282e-02 OccupAsp <--- FatherEd
gam74          0.198486 0.042069   4.71809 2.3807e-06 OccupAsp <--- FatherOcc
V[Grades]      0.650995 0.033935  19.18333 5.1010e-82 Grades <--> Grades
V[EducExp]     0.623511 0.032503  19.18333 5.1010e-82 EducExp <--> EducExp
V[OccupAsp]    0.806964 0.042066  19.18333 5.1010e-82 OccupAsp <--> OccupAsp
```

Note, we are not modeling the DV correlations so the residuals will be large

Residuals from this model

```
> lowerMat(resid(sem.kerch))
             Intll Sblng FthrE FthrO Grads EdcEx OccpA
Intelligence 0.00
Siblings     0.00  0.00
FatherEd     0.00  0.00  0.00
FatherOcc    0.00  0.00  0.00  0.00
Grades       0.00  0.00  0.00  0.00  0.00
EducExp      0.00  0.00  0.00  0.00  0.26  0.00
OccupAsp     0.00  0.00  0.00  0.00  0.25  0.38  0.00
```

fixed sem = regression

Compare the coefficients from this sem with the regression β values

```
round(sem.kerch$coeff,3)
      gam51       gam52       gam53       gam54       gam61       gam62       gam63
       gam64       gam71       gam72       gam73     gam74
      0.526      -0.030       0.119       0.041       0.373      -0.124       0.221
      0.168       0.249      -0.092       0.099       0.198
     V[Grades]  V[EducExp]  V[OccupAsp]
    0.651       0.624       0.807

sc.kk <- setCor(x=c(1:4),y=c(5:7),data=R.kerch)

round(as.vector(sc.kk$coefficients),3)
 [1]  0.526 -0.030  0.119  0.041  0.373 -0.124  0.221  0.168  0.249 -0.092  0.099  0.198
```

More complicated regression

1. Able to model the intercorrelations of the Y variables
   - This treats the Ys as both predictors and predicted
2. Able to let some of the Ys be part of the regression model

Complete Kerckhoff-Kenny path analysis (taken from sem help page)

```
model.kerch1 <- specifyModel()
     Intelligence -> Grades,          gam51
     Siblings -> Grades,              gam52
     FatherEd -> Grades,              gam53
     FatherOcc -> Grades,             gam54
     Intelligence -> EducExp,         gam61
     Siblings -> EducExp,             gam62
     FatherEd -> EducExp,             gam63
     FatherOcc -> EducExp,            gam64
     Grades -> EducExp,               beta65
     Intelligence -> OccupAsp,        gam71
     Siblings -> OccupAsp,            gam72
     FatherEd -> OccupAsp,            gam73
     FatherOcc -> OccupAsp,           gam74
     Grades -> OccupAsp,              beta75
     EducExp -> OccupAsp,             beta76

sem.kerch1 <- sem(model.kerch1, R.kerch, 737, fixed.x=c('Intelligence','S
     'FatherEd','FatherOcc'))
summary(sem.kerch1)
```

sem output

```
 Model Chisquare =  3.2685e-13   Df =  0 Pr(>Chisq) = NA
 Chisquare (null model) =  1664.3   Df =  21
 Goodness-of-fit index =  1
 AIC =  36
 AICc =  0.95265
 BIC =  118.85
 CAIC =  3.2685e-13
 Normalized Residuals
     Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-4.26e-15  -1.35e-15  0.00e+00  -4.17e-16  0.00e+00  1.49e-15
 Parameter Estimates
             Estimate   Std Error z value   Pr(>|z|)
gam51         0.525902 0.031182  16.86530 8.0987e-64 Grades <--- Intelligence
gam52        -0.029942 0.030149  -0.99314 3.2064e-01 Grades <--- Siblings
gam53         0.118966 0.038259   3.10951 1.8740e-03 Grades <--- FatherEd
gam54         0.040603 0.037785   1.07456 2.8257e-01 Grades <--- FatherOcc
gam61         0.160270 0.032710   4.89979 9.5940e-07 EducExp <--- Intelligence
gam62        -0.111779 0.026876  -4.15899 3.1966e-05 EducExp <--- Siblings
gam63         0.172719 0.034306   5.03461 4.7882e-07 EducExp <--- FatherEd
gam64         0.151852 0.033688   4.50758 6.5571e-06 EducExp <--- FatherOcc
beta65        0.405150 0.032838  12.33799 5.6552e-35 EducExp <--- Grades
gam71        -0.039405 0.034500  -1.14215 2.5339e-01 OccupAsp <--- Intelligence
gam72        -0.018825 0.028222  -0.66700 5.0477e-01 OccupAsp <--- Siblings
gam73        -0.041333 0.036216  -1.14126 2.5376e-01 OccupAsp <--- FatherEd
gam74         0.099577 0.035446   2.80924 4.9658e-03 OccupAsp <--- FatherOcc
beta75        0.157912 0.037443   4.21738 2.4716e-05 OccupAsp <--- Grades
beta76        0.549593 0.038260  14.36486 8.5976e-47 OccupAsp <--- EducExp
V[Grades]     0.650995 0.033935  19.18333 5.1010e-82 Grades <--> Grades
V[EducExp]    0.516652 0.026932  19.18333 5.1010e-82 EducExp <--> EducExp
V[OccupAsp]   0.556617 0.029016  19.18333 5.1010e-82 OccupAsp <--> OccupAsp
```

Note that these models give different path coefficients

```
> round(sem.kerch$coeff,3)
       gam51       gam52       gam53       gam54       gam61       gam62       gam63
gam64
        gam71       gam72       gam73
       0.526      -0.030       0.119       0.041       0.373      -0.124       0.221
0.168
       0.249      -0.092       0.099
       gam74   V[Grades]  V[EducExp]  V[OccupAsp]
       0.198       0.651       0.624       0.807
> round(sem.kerch1$coeff,3)
       gam51       gam52       gam53       gam54       gam61       gam62       gam63
gam64
       beta65      gam71       gam72
       0.526      -0.030       0.119       0.041       0.160      -0.112       0.173
0.152
       0.405      -0.039      -0.019
       gam73       gam74       beta75      beta76    V[Grades]  V[EducExp] V[OccupAsp]
      -0.041       0.100       0.158       0.550       0.651       0.517       0.557

> round(sc.kk$coefficients,2)
             Grades EducExp OccupAsp
Intelligence   0.53    0.37     0.25
Siblings      -0.03   -0.12    -0.09
FatherEd       0.12    0.22     0.10
FatherOcc      0.04    0.17     0.20

 round(as.vector(sc.kk$coefficients),3)
 [1]  0.526 -0.030  0.119  0.041  0.373 -0.124  0.221  0.168  0.249 -0.092  0.099  0.198
```

A latent variable structural model

1. Taken from the LISREL User's reference guide
2. Data from Caslyn and Kenny (1977)
   - Self-concept of ability and perceived evaluation of others: Cause or effect of academic achievement
3. Variables
   - self concept
   - parental evaluation
   - teacher evaluation
   - friend evaluation
   - educational aspiration
   - college plans

Caslyn and Kenny data

	self	**parent**	teacher	friend	edu_asp	college
self_concept	1.00	0.73	0.70	0.58	0.46	0.56
parental_**eval**	0.73	1.00	0.68	0.61	0.43	0.52
teacher_**eval**	0.70	0.68	1.00	0.57	0.40	0.48
friend_**eval**	0.58	0.61	0.57	1.00	0.37	0.41
edu_aspir	0.46	0.43	0.40	0.37	1.00	0.72
college_plans	0.56	0.52	0.48	0.41	0.72	1.00

Creating the model using `structure.diagam`

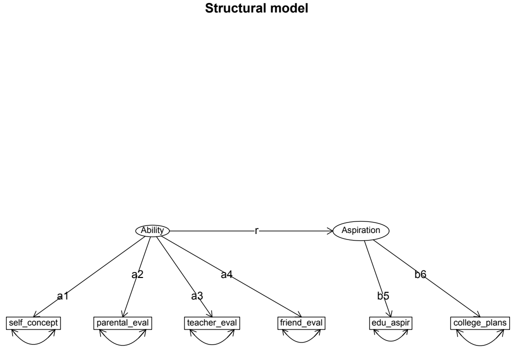
```
fx <- structure.list(6,list(c(1:4),c(5:6)),item.labels = rownames(ability)
                     f.labels=c("Ability","Aspiration"))
mod.edu <- structure.diagram(fx,"r",title="Lisrel␣example␣3.2",
                     errors=TRUE,lr=FALSE,cex=.8)
fx
```

```
fx
                Ability  Aspiration
self_concept    "a1"     "0"
parental_eval   "a2"     "0"
teacher_eval    "a3"     "0"
friend_eval     "a4"     "0"
edu_aspir       "0"      "b5"
college_plans   "0"      "b6"
```

The sem commands are in the mod.edu object

```
mod.edu
      Path                                   Parameter  Value
 [1,] "Ability->self_concept"                "a1"       NA
 [2,] "Ability->parental_eval"               "a2"       NA
 [3,] "Ability->teacher_eval"                "a3"       NA
 [4,] "Ability->friend_eval"                 "a4"       NA
 [5,] "Aspiration->edu_aspir"                "b5"       NA
 [6,] "Aspiration->college_plans"            "b6"       NA
 [7,] "self_concept<->self_concept"          "x1e"      NA
 [8,] "parental_eval<->parental_eval"        "x2e"      NA
 [9,] "teacher_eval<->teacher_eval"          "x3e"      NA
[10,] "friend_eval<->friend_eval"            "x4e"      NA
[11,] "edu_aspir<->edu_aspir"                "x5e"      NA
[12,] "college_plans<->college_plans"        "x6e"      NA
[13,] "Aspiration<->Ability"                 "rF2F1"    NA
[14,] "Ability<->Ability"                    NA         "1"
[15,] "Aspiration<->Aspiration"              NA         "1"
```

A model of the Caslyn-Kenny (1997) data set



```
ability <- as.matrix(ability) #sem requires matrix input
sem.edu <- sem(mod=mod.edu,S=ability,N=556)
summary(sem.edu)
```

```
Model Chisquare =  9.2557   Df =  8 Pr(>Chisq) = 0.32118
 Chisquare (null model) =  1832   Df =  15
 Goodness-of-fit index =  0.99443
 Adjusted goodness-of-fit index =  0.98537
 RMSEA index =  0.016817   90% CI: (NA, 0.054321)
 Bentler-Bonnett NFI =  0.99495
 Tucker-Lewis NNFI =  0.9987
 Bentler CFI =  0.99931
 SRMR =  0.012011
 AIC =  35.256
 AICc =  9.9273
 BIC =  91.426
 CAIC =  -49.31

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.4410 -0.1870  0.0000 -0.0131  0.2110  0.5330

 R-square for Endogenous Variables
 self_concept parental_eval  teacher_eval   friend_eval     edu_aspir college_plans
       0.7451        0.7213        0.6482        0.4834        0.6008        0.8629
```

With parameter estimates

```
 Parameter Estimates
       Estimate Std Error z value Pr(>|z|)
a1     0.86320  0.035145  24.5612 3.2846e-133 self_concept <--- Ability
a2     0.84932  0.035450  23.9582 7.5937e-127 parental_eval <--- Ability
a3     0.80509  0.036405  22.1149 2.2725e-108 teacher_eval <--- Ability
a4     0.69527  0.038634  17.9964  2.0795e-72 friend_eval <--- Ability
b5     0.77508  0.040357  19.2058  3.3077e-82 edu_aspir <--- Aspiration
b6     0.92893  0.039410  23.5712 7.6153e-123 college_plans <--- Aspiration
x1e    0.25488  0.023367  10.9075  1.0617e-27 self_concept <--> self_concep
x2e    0.27865  0.024128  11.5491  7.4600e-31 parental_eval <--> parental_e
x3e    0.35184  0.026919  13.0703  4.8660e-39 teacher_eval <--> teacher_eva
x4e    0.51660  0.034725  14.8768  4.6594e-50 friend_eval <--> friend_eval
x5e    0.39924  0.038196  10.4525  1.4266e-25 edu_aspir <--> edu_aspir
x6e    0.13709  0.043505   3.1511  1.6264e-03 college_plans <--> college_pl
rF2F1  0.66637  0.030954  21.5276 8.5783e-103 Ability <--> Aspiration
```

Three competing models

1. Ability and aspirations are correlated
   - r = .66
2. Ability causes aspirations
   - beta = .89
3. Aspirations cause ability
   - beta = .89

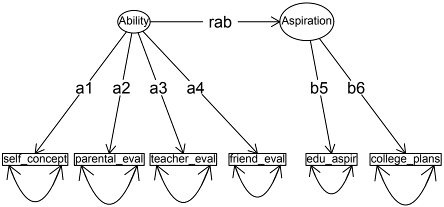
Create the model where ability lead to aspirations

```
phi <- phi.list(2,c(2))
phi
mod.edu <- structure.diagram(fx,phi,title="Aspiration␣leads␣to␣ability",
                        errors=TRUE, lr=FALSE, cex=.7)
```

```
mod.edu
      Path                                   Parameter  Value
 [1,] "Ability->self_concept"                "a1"       NA
 [2,] "Ability->parental_eval"               "a2"       NA
 [3,] "Ability->teacher_eval"                "a3"       NA
 [4,] "Ability->friend_eval"                 "a4"       NA
 [5,] "Aspiration->edu_aspir"                "b5"       NA
 [6,] "Aspiration->college_plans"            "b6"       NA
 [7,] "self_concept<->self_concept"          "x1e"      NA
 [8,] "parental_eval<->parental_eval"        "x2e"      NA
 [9,] "teacher_eval<->teacher_eval"          "x3e"      NA
[10,] "friend_eval<->friend_eval"            "x4e"      NA
[11,] "edu_aspir<->edu_aspir"                "x5e"      NA
[12,] "college_plans<->college_plans"        "x6e"      NA
[13,] "Ability␣->Aspiration"                 "rF1F2"    NA
[14,] "Ability<->Ability"                    NA         "1"
[15,] "Aspiration<->Aspiration"              NA         "1"
```

Ability causes aspiration

Ability leads to Aspiration



Fit statistics are identical

```
> sem.edu <- sem(mod=mod.edu,S=ability,N=556)
> summary(sem.edu)
```

```
summary(sem.edu)

 Model Chisquare =  9.2557   Df =  8 Pr(>Chisq) = 0.32118
 Chisquare (null model) =  1832   Df =  15
 Goodness-of-fit index =  0.99443
 Adjusted goodness-of-fit index =  0.98537
 RMSEA index =  0.016817   90% CI: (NA, 0.054321)
 Bentler-Bonnett NFI =  0.99495
 Tucker-Lewis NNFI =  0.9987
 Bentler CFI =  0.99931
 SRMR =  0.012011
 AIC =  35.256
 AICc =  9.9273
 BIC =  91.426
 CAIC =  -49.31

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.4410 -0.1870  0.0000 -0.0131  0.2110  0.5330

 R-square for Endogenous Variables
 self_concept parental_eval  teacher_eval   friend_eval    Aspiration     edu_aspir college_plans
       0.7451        0.7213        0.6482        0.4834        0.4440        0.6008
0.8629
```

But the paths are different

```
 Parameter Estimates
       Estimate Std Error z value Pr(>|z|)
a1     0.86320  0.035145  24.5612 3.2848e-133 self_concept <--- Ability
a2     0.84932  0.035450  23.9582 7.5919e-127 parental_eval <--- Ability
a3     0.80509  0.036405  22.1149 2.2732e-108 teacher_eval <--- Ability
a4     0.69527  0.038634  17.9964  2.0794e-72 friend_eval <--- Ability
b5     0.57792  0.030630  18.8678  2.0977e-79 edu_aspir <--- Aspiration
b6     0.69263  0.037979  18.2370  2.6257e-74 college_plans <--- Aspiration
x1e    0.25488  0.023367  10.9075  1.0617e-27 self_concept <--> self_concep
x2e    0.27865  0.024128  11.5491  7.4610e-31 parental_eval <--> parental_e
x3e    0.35184  0.026919  13.0703  4.8654e-39 teacher_eval <--> teacher_eva
x4e    0.51660  0.034725  14.8768  4.6595e-50 friend_eval <--> friend_eval
x5e    0.39924  0.038196  10.4525  1.4266e-25 edu_aspir <--> edu_aspir
x6e    0.13709  0.043505   3.1511  1.6264e-03 college_plans <--> college_pl
rF1F2  0.89371  0.074673  11.9683  5.2068e-33 Aspiration <--- Ability
<-------

 Iterations =  30
```

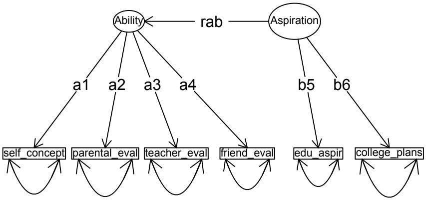
Let aspirations cause ability

```
> phi[1,2] <- phi[2,1]
> phi[2,1] <- "0"
> mod.edu <- structure.diagram(fx,phi,main="Aspiration leads to Ability",e
>mod.edu
```

```
> mod.edu
      Path                                  Parameter Value
 [1,] "Ability->self_concept"               "a1"      NA
 [2,] "Ability->parental_eval"              "a2"      NA
 [3,] "Ability->teacher_eval"               "a3"      NA
 [4,] "Ability->friend_eval"                "a4"      NA
 [5,] "Aspiration->edu_aspir"               "b5"      NA
 [6,] "Aspiration->college_plans"           "b6"      NA
 [7,] "self_concept<->self_concept"         "x1e"     NA
 [8,] "parental_eval<->parental_eval"       "x2e"     NA
 [9,] "teacher_eval<->teacher_eval"         "x3e"     NA
[10,] "friend_eval<->friend_eval"           "x4e"     NA
[11,] "edu_aspir<->edu_aspir"               "x5e"     NA
[12,] "college_plans<->college_plans"       "x6e"     NA
[13,] "Aspiration<-Ability"                 "rF2F1"   NA
[14,] "Ability<->Ability"                   NA        "1"
[15,] "Aspiration<->Aspiration"             NA        "1"
```

Aspiration leads to ability

Aspiration leads to Ability



Fits are still identical

```
> sem.edu <- sem(mod=mod.edu,S=ability,N=556)
> summary(sem.edu)

 Model Chisquare =  9.2557   Df =  8 Pr(>Chisq) = 0.32118
 Chisquare (null model) =  1832   Df =  15
 Goodness-of-fit index =  0.99443
 Adjusted goodness-of-fit index =  0.98537
 RMSEA index =  0.016817   90% CI: (NA, 0.054321)
 Bentler-Bonnett NFI =  0.99495
 Tucker-Lewis NNFI =  0.9987
 Bentler CFI =  0.99931
 SRMR =  0.012011
 AIC =  35.256
 AICc =  9.9273
 BIC =  91.426
 CAIC =  -49.31

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.4410 -0.1870  0.0000 -0.0131  0.2110  0.5330

 R-square for Endogenous Variables
 self_concept parental_eval  teacher_eval   friend_eval     Aspiration
edu_aspir college_plans
```

And the crucial coefficient is different

```
 Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
a1    0.86320  0.035145  24.5612 3.2848e-133 self_concept <--- Ability
a2    0.84932  0.035450  23.9582 7.5919e-127 parental_eval <--- Ability
a3    0.80509  0.036405  22.1149 2.2732e-108 teacher_eval <--- Ability
a4    0.69527  0.038634  17.9964  2.0794e-72 friend_eval <--- Ability
b5    0.57792  0.030630  18.8678  2.0977e-79 edu_aspir <--- Aspiration
b6    0.69263  0.037979  18.2370  2.6257e-74 college_plans <--- Aspiration
x1e   0.25488  0.023367  10.9075  1.0617e-27 self_concept <--> self_concep
x2e   0.27865  0.024128  11.5491  7.4610e-31 parental_eval <--> parental_e
x3e   0.35184  0.026919  13.0703  4.8654e-39 teacher_eval <--> teacher_eva
x4e   0.51660  0.034725  14.8768  4.6595e-50 friend_eval <--> friend_eval
x5e   0.39924  0.038196  10.4525  1.4266e-25 edu_aspir <--> edu_aspir
x6e   0.13709  0.043505   3.1511  1.6264e-03 college_plans <--> college_pl
rF2F1 0.89371  0.074673  11.9683  5.2068e-33 Aspiration <--- Ability
<---------
```

Compare the three models

	correlated	asp	abil
a1	0.86	0.64	0.86
a2	0.85	0.63	0.85
a3	0.81	0.60	0.81
a4	0.70	0.52	0.70
b5	0.78	0.78	0.58
b6	0.93	0.93	0.69
x1e	0.25	0.25	0.25
x2e	0.28	0.28	0.28
x3e	0.35	0.35	0.35
x4e	0.52	0.52	0.52
x5e	0.40	0.40	0.40
x6e	0.14	0.14	0.14
rF2F1	0.67	0.89	0.89

1. Although fits were identical
2. Paths differ as a function of presumed influence
3. Which solution is correct?
4. Is this even possible to answer?

lavaan syntax is perhaps easier (correlated latents)

R code

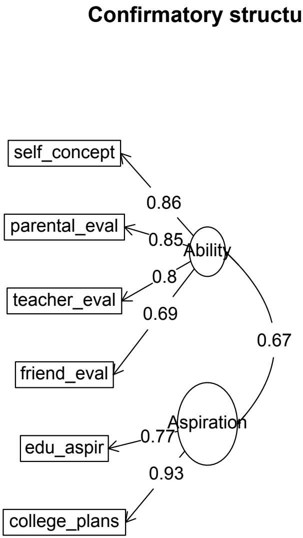
```
model <- "Ability =~ self_concept + parental_eval +
                                 teacher_eval + friend_eval
         Aspiration =~ edu_aspir + college_plans"
fit <- sem(model,sample.cov=casslyn,fixed.x=TRUE,sample.nobs=556,
       std.lv=TRUE,mimic="eqs")   #note that the default is lavaan
```

```
lavaan (0.5-17) converged normally after  22 iterations

  Number of observations                           556

  Estimator                                         ML
  Minimum Function Test Statistic                9.256
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.321

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard
```

lavaan output (continued)

```
                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  Ability =~
    self_concept      0.863    0.035   24.561    0.000
    parental_eval     0.849    0.035   23.958    0.000
    teacher_eval      0.805    0.036   22.115    0.000
    friend_eval       0.695    0.039   17.996    0.000
  Aspiration =~
    edu_aspir         0.775    0.040   19.206    0.000
    college_plans     0.929    0.039   23.571    0.000

Covariances:
  Ability ~~
    Aspiration        0.666    0.031   21.528    0.000

Variances:
    self_concept      0.255    0.023   10.907    0.000
    parental_eval     0.279    0.024   11.549    0.000
    teacher_eval      0.352    0.027   13.070    0.000
    friend_eval       0.517    0.035   14.877    0.000
    edu_aspir         0.399    0.038   10.453    0.000
    college_plans     0.137    0.044    3.151    0.002
    Ability           1.000
    Aspiration        1.000
```

A model of the Caslyn-Kenny (1997) data set using lavaan



lavaan syntax for regressions

R code

```
model <- "Ability =~ self_concept + parental_eval +
                                teacher_eval + friend_eval
         Aspiration =~ edu_aspir + college_plans
         Aspiration ~ Ability"
fit <- sem(model,sample.cov=casslyn,fixed.x=TRUE,sample.nobs=556,
       std.lv=TRUE,mimic="eqs")   #note that the default is lavaan
```

```
   lavaan (0.5-17) converged normally after  22 iterations

     Number of observations                           556

     Estimator                                         ML
     Minimum Function Test Statistic                9.256
     Degrees of freedom                                 8
     P-value (Chi-square)                           0.321

   Parameter estimates:

     Information                                 Expected
     Standard Errors                             Standard

                       Estimate  Std.err  Z-value  P(>|z|)
   Latent variables:
     Ability =~
       self_concept       0.863    0.035   24.561    0.000
```

lavaan regression output, continued

```
                    Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  Ability =~
    self_concept       0.863    0.035   24.561    0.000
    parental_eval      0.849    0.035   23.958    0.000
    teacher_eval       0.805    0.036   22.115    0.000
    friend_eval        0.695    0.039   17.996    0.000
  Aspiration =~
    edu_aspir          0.578    0.031   18.868    0.000
    college_plans      0.693    0.038   18.237    0.000

Regressions:
  Aspiration ~
    Ability            0.894    0.075   11.968    0.000

Variances:
    self_concept       0.255    0.023   10.907    0.000
    parental_eval      0.279    0.024   11.549    0.000
    teacher_eval       0.352    0.027   13.070    0.000
    friend_eval        0.517    0.035   14.877    0.000
    edu_aspir          0.399    0.038   10.453    0.000
    college_plans      0.137    0.044    3.151    0.002
    Ability            1.000
    Aspiration         1.000
```

Implications of arrows

1. Need to fit alternative models
   - Create alternative plausible models
   - Create alternative implausible models (they will fit also).
2. Need to consider alternative representations
   - Try reversing arrows
3. Are there external variables (e.g., time) that allow one to choose between models?
4. Confirmation that a model fits does not confirm theoretical adequacy of the model.

Two fundamentally different types of observed variables

1. Observed variables can be “reflective” of the latent variables. They are “effect indicators”.
   - Variables are caused by the latent variables.
   - Covariation between the variables are explained by the latent variables
2. Observed variables can be “causal indicators” or formative indicators that can directly effect the latent variable
   - Variables cause the “latent” variable
   - Covariation of the the observed variables is not modeled

Formative indicators Bollen (2002)

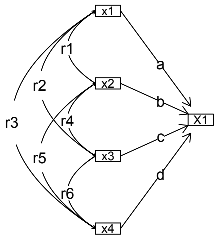
1. The correlational structure of formative indicators is independent of the loadings on a factor.
   - They are not locally independent
2. Examples of formative indicators Time spent in social interaction
   - Time spent with family, time spent with friends, time spent with coworkers.
   - These might in fact be negatively correlated even though total score is important.

Effect (reflective) indicators

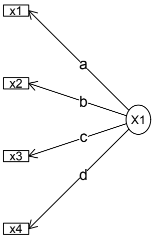
1. Test scores on various quantitative tests as effect indicators of trait
   - Feelings of self worth as effect indicators of self esteem
   - Ability items as indicators of ability
2. Correlational structure is a function of the path coefficients with latent variables
3. Variables are locally independent
   - (uncorrelated with each other when latent variable is partialled out)

Formative vs. Effect variables as regression versus factors

Formative Variables



Effect Variables



Confirmatory Factor Analysis as a way of testing factor models

1. Exploratory factor analysis (EFA) is a way of summarizing the relationships between variables
   - May have goodness of fit of factor model to the covariances but this is unusual
   - But all loadings are modeled
2. Confirmatory factor analysis (CFA) also models the data
   - But limits paths to specified paths, sets other to zero
   - Goodness of fit statistics are always produced
3. Blending CFA with regression is a full Structural Equation Model

Using sim.structural to create models

```
fx <-matrix(c( .9,.8,.6,rep(0,4),.6,.8,-.7),ncol=2)
fy <- matrix(c(.6,.5,.4),ncol=1)
rownames(fx) <- c("V","Q","A","nach","Anx")
rownames(fy)<- c("gpa","Pre","MA")
Phi <-matrix( c(1,0,.7,.0,1,.6,.7,.6,1),ncol=3)
gre.gpa <- sim.structural(fx,Phi,fy,n=1000)
gre.gpa
```

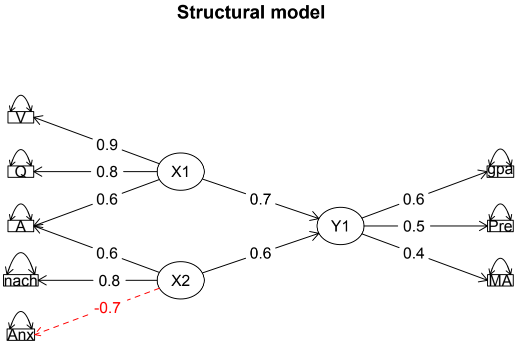
```
Call: sim.structural(fx = fx, Phi = Phi, fy = fy)

 $model (Population correlation matrix)
        V    Q    A nach   Anx   gpa
Pre    MA
V    1.00 0.72  0.54  0.00  0.00  0.38
0.32  0.25
Q    0.72 1.00  0.48  0.00  0.00  0.34
0.28  0.22
A    0.54 0.48  1.00  0.48 -0.42  0.47
0.39  0.31
nach 0.00 0.00  0.48  1.00 -0.56  0.29
0.24  0.19
Anx  0.00 0.00 -0.42 -0.56
1.00 -0.25 -0.21 -0.17
gpa  0.38 0.34  0.47  0.29 -0.25  1.00
0.30  0.24
Pre  0.32 0.28  0.39  0.24 -0.21  0.30
1.00  0.20
MA   0.25 0.22  0.31  0.19 -0.17  0.24
0.20  1.00

$reliability (population reliability)
```

```
> fx
> fy
> Phi
```

```
> fx
      [,1] [,2]
V      0.9  0.0
Q      0.8  0.0
A      0.6  0.6
nach   0.0  0.8
Anx    0.0 -0.7
> Phi
      [,1] [,2] [,3]
[1,]   1.0  0.0  0.7
[2,]   0.0  1.0  0.6
[3,]   0.7  0.6  1.0
> fy
     [,1]
gpa   0.6
Pre   0.5
MA    0.4
>
```

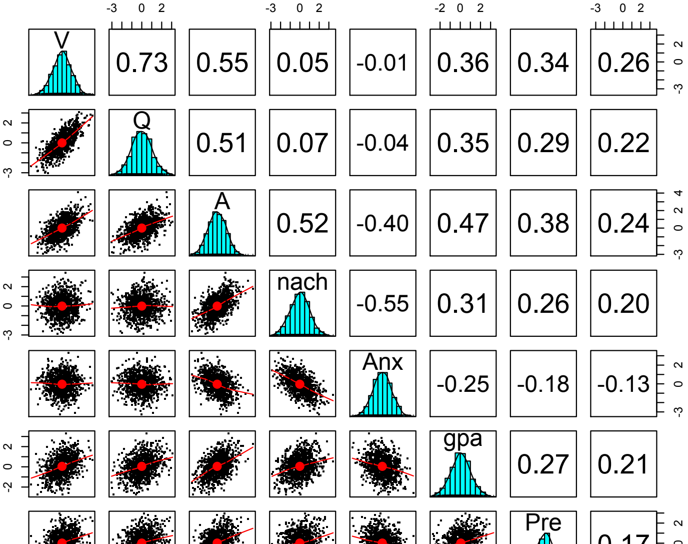
Create a figure (and write sem code for the sem package)

```
mod4 <- structure.diagram(fx,Phi,fy,errors=TRUE,e.size=.3)
```



Show the SPLOM

```
pairs.panels(gre.gpa$observed,pch=".")
```



Show the sem code – This uses RAM path notation for all arrows

```
> mod4

      Path            Parameter Value
 [1,] "X1->V"         "F1V"     NA
 [2,] "X1->Q"         "F1Q"     NA
 [3,] "X1->A"         "F1A"     NA
 [4,] "X2->A"         "F2A"     NA
 [5,] "X2->nach"      "F2nach"  NA
 [6,] "X2->Anx"       "F2Anx"   NA
 [7,] "V<->V"         "x1e"     NA
 [8,] "Q<->Q"         "x2e"     NA
 [9,] "A<->A"         "x3e"     NA
[10,] "nach<->nach"   "x4e"     NA
[11,] "Anx<->Anx"     "x5e"     NA
[12,] "Y1->gpa"       "Fygpa"   NA
[13,] "Y1->Pre"       "FyPre"   NA
[14,] "Y1->MA"        "FyMA"    NA
[15,] "gpa<->gpa"     "y1e"     NA
[16,] "Pre<->Pre"     "y2e"     NA
[17,] "MA<->MA"       "y3e"     NA
[18,] "X2<->X1"       "rF2F1"   NA
[19,] "X1->Y1"        "rX1Y1"   NA
[20,] "X2->Y1"        "rX2Y1"   NA
[21,] "X1<->X1"       NA        "1"
[22,] "X2<->X2"       NA        "1"
[23,] "Y1<->Y1"       NA        "1"
attr(,"class")
[1] "mod"
```

Can we recover the structure using a confirmatory factor model?

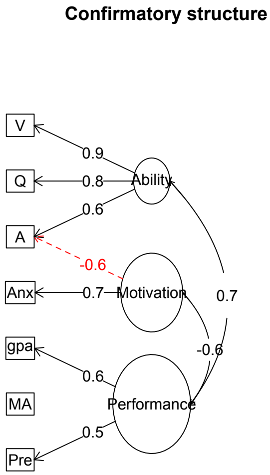
R code

```
model <- 'Ability =~ V + Q + A
              Motivation =~ A + Anx
              Performance =~ gpa + MA + Pre'
  fit <- sem(model,gre.gpa$observed,std.lv=TRUE)
  summary(fit)
  lavaan.diagram(fit)
```

lavaan output

```
lavaan (0.5-17) converged normally after  28 iterations

  Number of observations                          1000

  Estimator                                         ML
  Minimum Function Test Statistic               12.116
  Degrees of freedom                                10
  P-value (Chi-square)                           0.277

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  Ability =~
    V                 0.924    0.029   32.165    0.000
    Q                 0.830    0.029   28.263    0.000
    A                 0.614    0.033   18.540    0.000
  Motivation =~
    A                -0.593    0.038  -15.642    0.000
    Anx               0.660    0.044   14.990    0.000
  Performance =~
    gpa               0.570    0.036   16.001    0.000
    MA                0.341    0.035    9.637    0.000
    Pre               0.483    0.035   13.678    0.000

Covariances:
  Ability ~~
    Motivation       -0.033    0.051   -0.654    0.513
    Performance       0.748    0.039   19.386    0.000
  Motivation ~~
```

Create a figure from the lavaan output)



But the data can also be modeled as full SEM model

R code

```
model <- 'Ability =~ V + Q + A
              Motivation =~ A + Anx
              Performance =~ gpa + MA + Pre
              Performance ~ Ability + Motivation'
  fit <- sem(model,gre.gpa$observed,std.lv=TRUE)
  summary(fit)
  standardizedSolution(fit)
  lavaan.diagram(fit)
```

lavaan solution for the regression model

```
standardizedSolution(fit)
           lhs op         rhs est.std    se       z pvalue
1      Ability =~           V   0.899 0.028  32.165  0.000
2      Ability =~           Q   0.810 0.029  28.263  0.000
3      Ability =~           A   0.593 0.032  18.540  0.000
4   Motivation =~           A  -0.573 0.037 -15.642  0.000
5   Motivation =~         Anx   0.672 0.045  14.990  0.000
6  Performance =~         gpa   0.579 0.198   2.932  0.003
7  Performance =~          MA   0.340 0.117   2.891  0.004
8  Performance =~         Pre   0.484 0.165   2.941  0.003
9  Performance  ~     Ability   0.729 0.252   2.889  0.004
10 Performance  ~  Motivation  -0.575 0.209  -2.749  0.006
11           V ~~           V   0.193 0.026   7.444  0.000
12           Q ~~           Q   0.344 0.025  13.704  0.000
13           A ~~           A   0.297 0.034   8.625  0.000
14         Anx ~~         Anx   0.548 0.053  10.301  0.000
15         gpa ~~         gpa   0.664 0.039  17.001  0.000
16          MA ~~          MA   0.885 0.041  21.435  0.000
17         Pre ~~         Pre   0.766 0.039  19.660  0.000
18     Ability ~~     Ability   1.000    NA      NA     NA
19  Motivation ~~  Motivation   1.000    NA      NA     NA
20 Performance ~~ Performance   0.110    NA      NA     NA
21     Ability ~~  Motivation  -0.033 0.051  -0.654  0.513
```

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

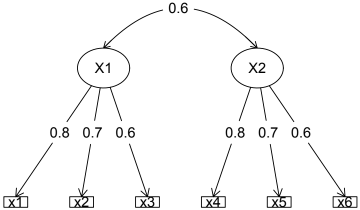
Create some basic data and add in some change

```
> set.seed(42)
 >   fx <- matrix(c(.8,.7,.6,rep(0,6),.8,.7,.6),ncol=2)
>   fx
      [,1] [,2]
[1,]   0.8   0.0
[2,]   0.7   0.0
[3,]   0.6   0.0
[4,]   0.0   0.8
[5,]   0.0   0.7
[6,]   0.0   0.6
> Phi <- matrix(c(1,.6,.6,1),ncol=2)
> Phi
      [,1] [,2]
[1,]   1.0   0.6
[2,]   0.6   1.0

> x.model <- sim(fx=fx,Phi=Phi,mu=c(0,1),n=250)
> x <- x.model$observed
> structure.diagram(fx,Phi,lr=FALSE,e.size=.3,main="A basic two time model")
> describe(x,skew=FALSE)

>  pairs.panels(x,pch=".")


   vars    n mean    sd median trimmed  mad    min
max range   skew kurtosis    se
1      1 250 0.01 0.97     0.06     0.01 0.92 -2.56 2.68
5.24   0.05      0.03 0.06
2      2 250 0.07 1.04     0.05     0.08 0.98 -2.76 3.07
5.83 -0.05     -0.04 0.07
3      3 250 0.00 0.98    -0.03    -0.04 0.97 -2.62 3.90
6.52   0.53      0.79 0.06
4      4 250 0.83 0.95     0.85     0.84 1.01 -2.04 3.64
```

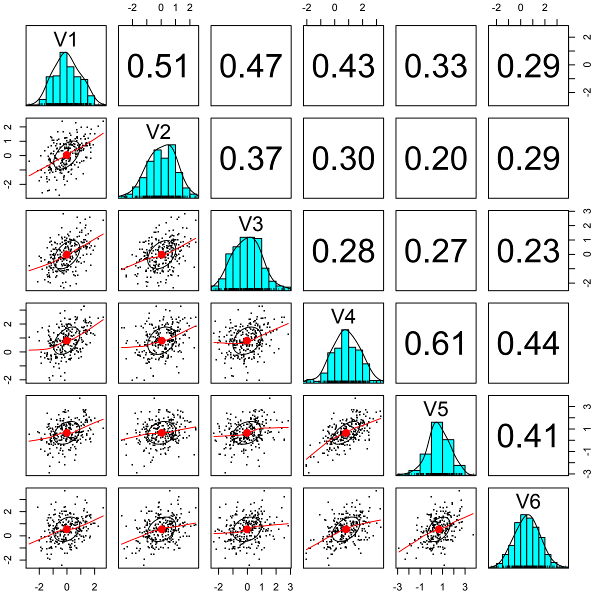
1. Set the random seed
2. Create a one factor structure
3. Put in some change
4. Show the structural model
5. Describe it
6. Show the splom (with small pch)

A basic two occasion trait model

A basic two time model



Splom of 6 basic variables



Multiple models

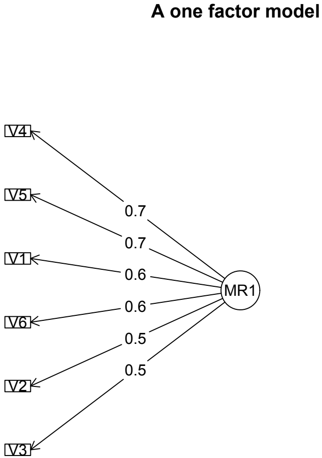
1. Ignore time, are the data congeneric?
   - Are they all measures of the same thing?
   - This is a one factor model
2. Include time, do we recover a correlation across time?
   - Try a two factor model
   - Plot the resulting structure

A one factor model

```
> fa(x)

Factor Analysis using method =  minres
Call: fa(r = x)
Standardized loadings (pattern matrix) based upon correlation matrix
   MR1   h2   u2 com
1 0.76 0.57 0.43   1
2 0.65 0.42 0.58   1
3 0.56 0.31 0.69   1
4 0.64 0.41 0.59   1
5 0.50 0.25 0.75   1
6 0.50 0.25 0.75   1

                MR1
SS loadings    2.21
Proportion Var 0.37

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

```
The degrees of freedom for the null model are
15
    and the objective function was  1.57
                  with Chi Square of  387.63
The degrees of freedom for the model are 9
                   and the objective function was
0.28

The root mean square of the residuals (RMSR) is
0.09
The df corrected root mean square of
                                       the residuals is
0.12

The harmonic number of observations is  250
              with the empirical chi square
62.96  with prob <  3.6e-10
The total number of observations was  250
                    with MLE Chi Square =
67.96  with prob <  3.8e-11

Tucker Lewis Index of factoring reliability =
0.736
RMSEA index =  0.164
and the 90 % confidence intervals are
0.127 0.199
BIC =  18.27
Fit based upon off diagonal values = 0.94
Measures of factor score adequacy
                                                    MR1

Correlation of scores with factors
0.89
```

Fit a one factor model to the data fa.diagram(fa(x),sort=FALSE)



Try a two factor solution

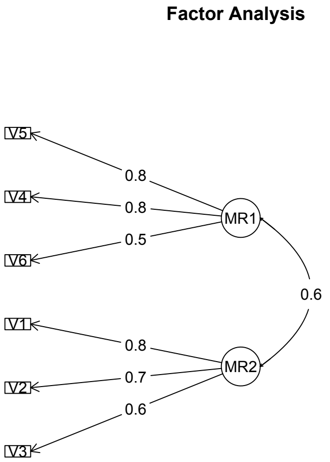
```
> fa(x,2)

Factor Analysis using method =  minres
Call: fa(r = x, nfactors = 2)
Standardized loadings (pattern matrix) based upon correlation matrix
    MR1   MR2   h2   u2 com
1  0.88 -0.01 0.76 0.24 1.0
2  0.66  0.02 0.45 0.55 1.0
3  0.58  0.00 0.33 0.67 1.0
4 -0.01  0.87 0.75 0.25 1.0
5 -0.01  0.60 0.36 0.64 1.0
6  0.10  0.48 0.29 0.71 1.1

                       MR1  MR2
SS loadings           1.58 1.37
Proportion Var        0.26 0.23
Cumulative Var        0.26 0.49
Proportion Explained  0.53 0.47
Cumulative Proportion 0.53 1.00

 With factor correlations of
     MR1  MR2
MR1 1.00 0.58
MR2 0.58 1.00
Measures of factor score adequacy
                                              MR1
MR2
Correlation of scores with factors
0.91 0.90
Multiple R square of scores with factors
0.83 0.81

Test of the hypothesis that 2 factors are sufficier

The degrees of freedom for the null model are
15 and
     the objective function was
1.57 with Chi Square of  387.63
The degrees of freedom for the model are 4
              and the objective function was
0.01

The root mean square of the residuals (RMSR) is
0.02
The df corrected root mean square of the residuals
0.03

The harmonic number of observations is  250
              with the empirical chi square
1.97  with prob <  0.74
The total number of observations was  250
              with MLE Chi Square =  2.51
with prob <  0.64

Tucker Lewis Index of factoring reliability =
1.015
RMSEA index =  0
and the 90 % confidence intervals are
NA 0.077
BIC =  -19.58
Fit based upon off diagonal values = 1
```

Compare the two solutions

```
Factor Analysis using method =  minres
Call: fa(r = x)
Factor Analysis using method =  minres
Call: fa(r = x)
Standardized loadings (pattern matrix) based upon correlation matrix
   MR1   h2   u2 com
1 0.76 0.57 0.43   1
2 0.65 0.42 0.58   1
3 0.56 0.31 0.69   1
4 0.64 0.41 0.59   1
5 0.50 0.25 0.75   1
6 0.50 0.25 0.75   1

                MR1
SS loadings    2.21
Proportion Var 0.37
```

```
Factor Analysis using method =  minres
Factor Analysis using method =  minres
Call: fa(r = x, nfactors = 2)
Standardized loadings (pattern matrix) based upon c
    MR1   MR2   h2   u2 com
1  0.88 -0.01 0.76 0.24 1.0
2  0.66  0.02 0.45 0.55 1.0
3  0.58  0.00 0.33 0.67 1.0
4 -0.01  0.87 0.75 0.25 1.0
5 -0.01  0.60 0.36 0.64 1.0
6  0.10  0.48 0.29 0.71 1.1

                       MR1  MR2
SS loadings           1.58 1.37
Proportion Var        0.26 0.23
Cumulative Var        0.26 0.49
Proportion Explained  0.53 0.47
Cumulative Proportion 0.53 1.00

 With factor correlations of
     MR1  MR2
MR1 1.00 0.58
MR2 0.58 1.00
```

EFA two factor solution



Now try three different sems (using lavaan)

1. Two correlated factors, free loadings
2. Two correlated factors, equal loadings across occasions
3. Two correlated factors, all loadings equal

A simple sem with a standardized solution using lavaan

R code

```
#factor model
mod2f <- 'F1 =~ V1 + V2 + V3
               F2 =~ V4 + V5 + V6
                  #correlation between factors
                 F1 ~~F2'# now fit it and summarize it
fit <- sem(mod2f, data=x,std.lv=TRUE)
summary(fit, fit.measures=TRUE)
standardizedSolution(fit)
```

```
    lhs op rhs est.std    se      z pvalue
1    F1 =~  V1   0.865 0.062 13.921      0
2    F1 =~  V2   0.681 0.063 10.756      0
3    F1 =~  V3   0.580 0.064  9.000      0
4    F2 =~  V4   0.838 0.067 12.599      0
5    F2 =~  V5   0.607 0.066  9.142      0
6    F2 =~  V6   0.558 0.067  8.348      0
7    F1 ~~  F2   0.602 0.061  9.887      0
8    V1 ~~  V1   0.251 0.068  3.713      0
9    V2 ~~  V2   0.536 0.064  8.443      0
10   V3 ~~  V3   0.664 0.068  9.763      0
11   V4 ~~  V4   0.297 0.077  3.885      0
12   V5 ~~  V5   0.631 0.070  9.036      0
13   V6 ~~  V6   0.689 0.072  9.609      0
14   F1 ~~  F1   1.000    NA     NA     NA
15   F2 ~~  F2   1.000    NA     NA     NA
```

lavaan fit statistics

```
lavaan (0.5-17) converged normally after
16 iterations

  Number of observations
250

  Estimator
ML
  Minimum Function Test Statistic
3.986
  Degrees of freedom
8
  P-value (Chi-square)
0.858

Model test baseline model:

  Minimum Function Test Statistic
393.670
  Degrees of freedom
15
  P-value
0.000

User model versus baseline model:

  Comparative Fit Index (CFI)
1.000
  Tucker-Lewis Index (TLI)
1.020

Loglikelihood and Information Criteria:
```

```
Root Mean Square Error of Approximation:

  RMSEA
0.000
  90 Percent Confidence Interval
0.000  0.040
  P-value RMSEA <= 0.05
0.972

Standardized Root Mean Square Residual:

  SRMR
0.017

Parameter estimates:

  Information
Expected
  Standard Errors
Standard


                          Estimate  Std.err
Z-value  P(>|z|)
Latent variables:
  F1 =~
    V1                       0.840    0.060
13.921    0.000
    V2                       0.703    0.065
10.756    0.000
    V3                       0.568    0.063
9.000    0.000
```

Create two new models with some equality constraints

The first sets the loadings for 1-3 equal to those of 4-6, the second says they are all equal

R code

```
mod2fa <- 'F1 =~ a*V1 + b*V2 + c*V3
                   F2 =~ a*V4 + b*V5 + c*V6
                   F1 ~~F2'
 fit2a <- sem(mod2fa,data=x.df,std.lv=TRUE)
 summary(fit2a,fit.measures=TRUE)

mod2fe <- 'F1 =~ a*V1 + a*V2 + a*V3
                   F2 =~ a*V4 + a*V5 + a*V6
                   F1 ~~F2'
 fit2e <- sem(mod2fe,data=x.df,std.lv=TRUE)
 summary(fit2e,fit.measures=TRUE)
```

Equal across time

```
lavaan (0.5-17) converged normally after
15 iterations

  Number of observations
250

   Estimator
ML
  Minimum Function Test Statistic
5.338
  Degrees of freedom
11
  P-value (Chi-square)
0.914

Model test baseline model:

  Minimum Function Test Statistic
393.670
  Degrees of freedom
15
  P-value
0.000

User model versus baseline model:

  Comparative Fit Index (CFI)
1.000
  Tucker-Lewis Index (TLI)
1.020

Loglikelihood and Information Criteria:
```

```
Root Mean Square Error of Approximation:

  RMSEA
0.000
  90 Percent Confidence Interval
0.000   0.025
  P-value RMSEA <= 0.05
0.991

Standardized Root Mean Square Residual:

  SRMR
0.035

Parameter estimates:

   Information
Expected
   Standard Errors
Standard

                         Estimate  Std.err
Z-value  P(>|z|)
Latent variables:
  F1 =~
    V1          (a)       0.817     0.046
17.711     0.000
    V2          (b)       0.662     0.049
13.630     0.000
    V3          (c)       0.549     0.046
11.949     0.000
```

All loadings equal

```
> summary(fit2e, fit.measures=TRUE)
llavaan (0.5-17) converged normally after
12 iterations

  Number of observations
250

  Estimator
ML
  Minimum Function Test Statistic
26.140
  Degrees of freedom
13
  P-value (Chi-square)
0.016

Model test baseline model:

  Minimum Function Test Statistic
393.670
  Degrees of freedom
15
  P-value
0.000

User model versus baseline model:

  Comparative Fit Index (CFI)
0.965
  Tucker-Lewis Index (TLI)
0.960
```

```
Root Mean Square Error of Approximation:

  RMSEA
0.064
  90 Percent Confidence Interval
0.026  0.099
  P-value RMSEA <= 0.05
0.235

Standardized Root Mean Square Residual:

  SRMR
0.084

Parameter estimates:

  Information
Expected
  Standard Errors
Standard

                              Estimate  Std.err
Z-value  P(>|z|)
Latent variables:
  F1 =~
    V1          (a)     0.686    0.033
20.964     0.000
    V2          (a)     0.686    0.033
20.964     0.000
    V3          (a)     0.686    0.033
```

Do the models differ?

These are nested models, and we can compare their χ^2 values.

R code

```
anova(fit,fit2a)
anova(fit2a,2e)
```

```
Chi Square Difference Test

      Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit    8 3838.9 3884.6 3.9861
fit2a 11 3834.2 3869.4 5.3380     1.3519       3     0.7169

hi Square Difference Test

      Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit2a 11 3834.2 3869.4  5.338
fit2e 13 3851.0 3879.2 26.140     20.802       2   3.04e-05 ***
```

Measurement Invariance: Does a test measure the same thing

1. Across groups
    - Different schools
    - Different groups (e.g., ethnicity, age, gender)
2. Across time
    - Is todays' measure the same as next year's measure?
3. Types of invariance
    - Configural: Are the arrows the same
    - Weak invariance: Are the loadings the same across groups
    - Strong invariance: Loadings and intercepts are equal across groups
    - Super strong: Loadings, intercepts and means are equal across groups

Consider the Holzinger Swineford data set

1. 9 ability measures from two schools
   - 145 from Grant-White
   - 156 from Pasteur
2. Are the factor structures the same across schools
   - Although lavaan does this in one call, lets do it part by part
   - over all factor structure
   - factors with schools
   - constrain factors to have the same loadings, etc.

First, some descriptive statistics

describe(HolzingerSwineford1939, skew=FALSE)

	var	n	mean	sd	median	trimmed	mad	min	max	range	se
id	1	301	176.55	105.94	163.00	176.78	140.85	1.00	351.00	350.00	6.11
sex	2	301	1.51	0.50	2.00	1.52	0.00	1.00	2.00	1.00	0.03
ageyr	3	301	13.00	1.05	13.00	12.89	1.48	11.00	16.00	5.00	0.06
agemo	4	301	5.38	3.45	5.00	5.32	4.45	0.00	11.00	11.00	0.20
school*	5	301	1.52	0.50	2.00	1.52	0.00	1.00	2.00	1.00	0.03
grade	6	300	7.48	0.50	7.00	7.47	0.00	7.00	8.00	1.00	0.03
x1	7	301	4.94	1.17	5.00	4.96	1.24	0.67	8.50	7.83	0.07
x2	8	301	6.09	1.18	6.00	6.02	1.11	2.25	9.25	7.00	0.07
x3	9	301	2.25	1.13	2.12	2.20	1.30	0.25	4.50	4.25	0.07
x4	10	301	3.06	1.16	3.00	3.02	0.99	0.00	6.33	6.33	0.07
x5	11	301	4.34	1.29	4.50	4.40	1.48	1.00	7.00	6.00	0.07
x6	12	301	2.19	1.10	2.00	2.09	1.06	0.14	6.14	6.00	0.06
x7	13	301	4.19	1.09	4.09	4.16	1.10	1.30	7.43	6.13	0.06
x8	14	301	5.53	1.01	5.50	5.49	0.96	3.05	10.00	6.95	0.06
x9	15	301	5.37	1.01	5.42	5.37	0.99	2.78	9.25	6.47	0.06

describeBy each group

```
> describeBy(HolzingerSwineford1939,group=HolzingerSwineford1939$school,sk

group: Grant-White
        var   n  mean   sd median trimmed  mad   min   max range
se
sex       2 145  1.50 0.50   2.00    1.50 0.00  1.00  2.00  1.00 0.0
ageyr     3 145 12.72 0.97  13.00   12.67 1.48 11.00 16.00  5.00 0.0
...
grade     6 144  7.45 0.50   7.00    7.44 0.00  7.00  8.00  1.00 0.0
x1        7 145  4.93 1.15   5.00    4.96 1.24  1.83  8.50  6.67 0.1
x2        8 145  6.20 1.11   6.25    6.14 1.11  2.25  9.25  7.00 0.0
...
x8       14 145  5.49 1.05   5.50    5.45 0.89  3.05 10.00  6.95 0.0
x9       15 145  5.33 1.03   5.31    5.33 1.15  3.11  9.25  6.14 0.0
------------------------------------------------------------------------
group: Pasteur
        var   n  mean   sd median trimmed  mad   min   max range
se
sex       2 156  1.53 0.50   2.00    1.53 0.00  1.00  2.00  1.00 0.04
ageyr     3 156 13.25 1.06  13.00   13.15 1.48 12.00 16.00  4.00 0.09
...
grade     6 156  7.50 0.50   7.50    7.50 0.74  7.00  8.00  1.00 0.04
x1        7 156  4.94 1.19   5.00    4.97 1.24  0.67  7.50  6.83 0.09
x2        8 156  5.98 1.23   5.75    5.89 1.11  3.50  9.25  5.75 0.10
```

EFA for both groups

by(HolzingerSwineford1939[,7:15],HolzingerSwineford1939[,5],fa,nfactors

```
HolzingerSwineford1939[, 5]: Grant-White
Factor Analysis using method =  minres
Call: FUN(r = data[x, , drop = FALSE], nfactors = 3
Standardized loadings (pattern matrix) based upon
     MR1   MR2   MR3   h2   u2
x1  0.09  0.06  0.64 0.50 0.50
x2  0.02 -0.03  0.51 0.26 0.74
x3  0.11 -0.03  0.64 0.47 0.53
x4  0.86 -0.04  0.04 0.76 0.24
x5  0.82  0.09 -0.03 0.70 0.30
x6  0.81 -0.04  0.05 0.68 0.32
x7  0.14  0.78 -0.19 0.60 0.40
x8 -0.11  0.79  0.18 0.69 0.31
x9  0.08  0.46  0.40 0.54 0.46

                       MR1  MR2  MR3
SS loadings           2.23 1.53 1.44
Proportion Var        0.25 0.17 0.16
Cumulative Var        0.25 0.42 0.58
Proportion Explained  0.43 0.29 0.28
Cumulative Proportion 0.43 0.72 1.00

 With factor correlations of
     MR1  MR2  MR3
MR1 1.00 0.25 0.41
MR2 0.25 1.00 0.31
MR3 0.41 0.31 1.00
```

```
HolzingerSwineford1939[, 5]: Pasteur
Factor Analysis using method =  minres
Call: FUN(r = data[x, , drop = FALSE], nfactors = 3
Standardized loadings (pattern matrix) based upon c
     MR1   MR2   MR3   h2   u2
x1  0.27  0.59  0.00 0.51 0.49
x2  0.03  0.49 -0.16 0.25 0.75
x3 -0.08  0.73  0.01 0.50 0.50
x4  0.80  0.02  0.06 0.68 0.32
x5  0.92 -0.07 -0.06 0.79 0.21
x6  0.78  0.13  0.06 0.70 0.30
x7  0.06 -0.14  0.71 0.52 0.48
x8 -0.02  0.13  0.60 0.39 0.61
x9 -0.02  0.37  0.40 0.34 0.66

                       MR1  MR2  MR3
SS loadings           2.22 1.36 1.10
Proportion Var        0.25 0.15 0.12
Cumulative Var        0.25 0.40 0.52
Proportion Explained  0.48 0.29 0.23
Cumulative Proportion 0.48 0.77 1.00

 With factor correlations of
     MR1  MR2  MR3
MR1 1.00 0.27 0.26
MR2 0.27 1.00 0.14
MR3 0.26 0.14 1.00
```

How similar are the solutions: factor congruence

Factor congruence is the cosine of the angle between two vectors:

$$Congruence = (diag(X'X))^{-.5}Y'X(diag((Y'Y))^{-.5}$$

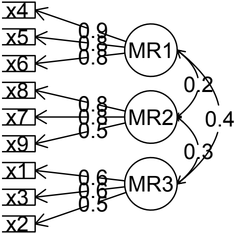
```
> f3.pasteur <- fa(HolzingerSwineford1939[1:156,7:15],3)
> f3.grant <- fa(HolzingerSwineford1939[157:301,7:15],3)
>factor.congruence(f3.pasteur,f3.grant)
```

	MR1	MR2	MR3
MR1	0.97	0.03	0.10
MR2	0.12	0.11	0.99
MR3	0.08	0.97	0.05

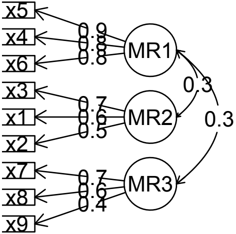
```
cross <- t(y) %*% x
sumsx <- sqrt(1/diag(t(x) %*% x))
sumsy <- sqrt(1/diag(t(y) %*% y))
```

Do they look alike?

Factor Analysis



Factor Analysis



Test if the model fits the combined data

```
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '

 fit <- cfa(HS.model, data=HolzingerSwineford1939, std.lv=TRUE)
summary(fit, fit.measures=TRUE)

lavaan (0.4-14) converged normally after 41 iterations

  Number of observations                           301

  Estimator                                         ML
  Minimum Function Chi-square                   85.306
  Degrees of freedom                                24
  P-value                                        0.000

Chi-square test baseline model:

  Minimum Function Chi-square                  918.852
  Degrees of freedom                                36
  P-value                                        0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.931
  Tucker-Lewis Index (TLI)                       0.896

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -3737.745
  Loglikelihood unrestricted model (H1)      -3695.092
```

With values of

```
Number of free parameters                          21
  Akaike (AIC)                                 7517.490
  Bayesian (BIC)                               7595.339
  Sample-size adjusted Bayesian (BIC)          7528.739

Root Mean Square Error of Approximation:

  RMSEA                                           0.092
  90 Percent Confidence Interval          0.071  0.114
  P-value RMSEA <= 0.05                           0.001

Standardized Root Mean Square Residual:

  SRMR                                            0.065

Parameter estimates:

  Information                                  Expected
  Standard Errors                              Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                0.900    0.081   11.127    0.000
    x2                0.498    0.077    6.429    0.000
    x3                0.656    0.074    8.817    0.000
  textual =~
    x4                0.990    0.057   17.474    0.000
    x5                1.102    0.063   17.576    0.000
    x6                0.917    0.054   17.082    0.000
  speed =~
    x7                0.619    0.070    8.903    0.000
    x8                0.731    0.066   11.090    0.000
```

Now do it for both groups one analysis

```
> fit2 <- cfa(HW.model, data=HolzingerSwineford1939, group="school",std.lv=TRUE)
> summary(fit2)


  Number of observations per group
  Pasteur                                            156
  Grant-White                                        145

  Estimator                                           ML
  Minimum Function Chi-square                    115.851
  Degrees of freedom                                  48
  P-value                                          0.000

Chi-square for each group:

  Pasteur                                         64.309
  Grant-White                                     51.542

Parameter estimates:

  Information                                   Expected
  Standard Errors                               Standard
```

Results for Pasteur

```
Group 1 [Pasteur]:

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.047    0.132    7.934    0.000
    x2                0.412    0.110    3.753    0.000
    x3                0.597    0.108    5.525    0.000
  textual =~
    x4                0.946    0.079   11.927    0.000
    x5                1.119    0.089   12.604    0.000
    x6                0.827    0.068   12.230    0.000
  speed =~
    x7                0.591    0.106    5.557    0.000
    x8                0.665    0.102    6.531    0.000
    x9                0.545    0.097    5.596    0.000

Covariances:
  visual ~~
    textual           0.484    0.086    5.600    0.000
    speed             0.299    0.109    2.755    0.006
  textual ~~
    speed             0.325    0.100    3.256    0.001

Intercepts:
    x1                4.941    0.095   52.249    0.000
    x2                5.984    0.098   60.949    0.000
    x3                2.487    0.093   26.778    0.000
    x4                2.823    0.092   30.689    0.000
    x5                3.995    0.105   38.183    0.000
    x6                1.922    0.079   24.321    0.000
    x7                4.432    0.087   51.181    0.000
    x8                5.563    0.078   71.214    0.000
```

Results for Grant-White

```
Latent variables:
  visual =~
    x1              0.777    0.103    7.525    0.000
    x2              0.572    0.101    5.642    0.000
    x3              0.719    0.093    7.711    0.000
  textual =~
    x4              0.971    0.079   12.355    0.000
    x5              0.961    0.083   11.630    0.000
    x6              0.935    0.081   11.572    0.000
  speed =~
    x7              0.679    0.087    7.819    0.000
    x8              0.833    0.087    9.568    0.000
    x9              0.719    0.086    8.357    0.000

Covariances:
  visual ~~
    textual         0.541    0.085    6.355    0.000
    speed           0.523    0.094    5.562    0.000
  textual ~~
    speed           0.336    0.091    3.674    0.000

Intercepts:
    x1              4.930    0.095   51.696    0.000
    x2              6.200    0.092   67.416    0.000
    x3              1.996    0.086   23.195    0.000
    x4              3.317    0.093   35.625    0.000
    x5              4.712    0.096   48.986    0.000
    x6              2.469    0.094   26.277    0.000
    x7              3.921    0.086   45.819    0.000
    x8              5.488    0.087   63.174    0.000
    x9              5.327    0.085   62.571    0.000
    visual          0.000
    textual         0.000
```

Constrain the loadings to be the same

```
> fit2e <- cfa(HW.model, data=HolzingerSwineford1939, group="school",std.lv=TRUE,group.equal="load
> summary(fit2e)
```

```
lavaan (0.4-14) converged normally after 30 iterations

  Number of observations per group
  Pasteur                                          156
  Grant-White                                      145

  Estimator                                         ML
  Minimum Function Chi-square                  127.834
  Degrees of freedom                                57
  P-value                                        0.000

Chi-square for each group:

  Pasteur                                       71.064
  Grant-White                                   56.770

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard
```

With parameter values of

```
Group 1 [Pasteur]:

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                0.866    0.078   11.149    0.000
    x2                0.523    0.076    6.916    0.000
    x3                0.683    0.071    9.689    0.000
  textual =~
    x4                0.954    0.056   17.002    0.000
    x5                1.033    0.061   17.012    0.000
    x6                0.870    0.052   16.750    0.000
  speed =~
    x7                0.630    0.066    9.500    0.000
    x8                0.752    0.065   11.586    0.000
    x9                0.650    0.064   10.205    0.000

Covariances:
  visual ~~
    textual           0.485    0.087    5.555    0.000
    speed             0.341    0.109    3.126    0.002
  textual ~~
    speed             0.336    0.094    3.590    0.000

Intercepts:
    x1                4.941    0.092   53.661    0.000
    x2                5.984    0.099   60.420    0.000
    x3                2.487    0.093   26.734    0.000
    x4                2.823    0.093   30.400    0.000
    x5                3.995    0.100   39.756    0.000
    x6                1.922    0.081   23.732    0.000
    x7                4.432    0.089   49.903    0.000
```

Now, compare the fit of the two group model to the one with equal parameters

```
> anova(fit2, fit2e)

Chi Square Difference Test

      Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit2  48 7484.4 7706.8 115.85
fit2e 57 7478.4 7667.4 127.83     11.982       9     0.2143
```

But, another way to specify the fit2e model – without making the latent variables standardized

```
fit2e <- cfa(HW.model, data=HolzingerSwineford1939, group="school",group.equal=c("loadings"))
```

```
 Number of observations per group
  Pasteur                                           156
  Grant-White                                       145

  Estimator                                          ML
  Minimum Function Chi-square                   124.044
  Degrees of freedom                                 54
  P-value                                         0.000

Chi-square for each group:
  Pasteur                                        68.825
  Grant-White                                    55.219
Parameter estimates:
  Information                                  Expected
  Standard Errors                              Standard

Group 1 [Pasteur]:
                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.000
    x2                0.599    0.100    5.979    0.000
    x3                0.784    0.108    7.267    0.000
  textual =~
    x4                1.000
    x5                1.083    0.067   16.049    0.000
    x6                0.912    0.058   15.785    0.000
  speed =~
    x7                1.000
```

Test fit by comparing models

```
> anova(fit2, fit2e)

Chi Square Difference Test

      Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit2  48 7484.4 7706.8 115.85
fit2e 54 7480.6 7680.8 124.04     8.1922       6     0.2244
```

Continue this logic, of successive tests with more constraints, but do it automatically

```
mi <- measurementInvariance(HW.model, data=HolzingerSwineford1939, group="school")
```

```
Measurement invariance tests:
Model 1: configural invariance:
   chisq        df    pvalue      cfi     rmsea      bic
 115.851    48.000     0.000    0.923     0.097 7706.822
Model 2: weak invariance (equal loadings):
   chisq        df    pvalue      cfi     rmsea      bic
 124.044    54.000     0.000    0.921     0.093 7680.771
[Model 1 versus model 2]
  delta.chisq      delta.df delta.p.value     delta.cfi
        8.192         6.000         0.224         0.002
Model 3: strong invariance (equal loadings + intercepts):
   chisq        df    pvalue      cfi     rmsea      bic
 164.103    60.000     0.000    0.882     0.107 7686.588
[Model 1 versus model 3]
  delta.chisq      delta.df delta.p.value     delta.cfi
       48.251        12.000         0.000         0.041
[Model 2 versus model 3]
  delta.chisq      delta.df delta.p.value     delta.cfi
       40.059         6.000         0.000         0.038
Model 4: equal loadings + intercepts + means:
   chisq        df    pvalue      cfi     rmsea      bic
 204.605    63.000     0.000    0.840     0.122 7709.969
[Model 1 versus model 4]
  delta.chisq      delta.df delta.p.value     delta.cfi
       88.754        15.000         0.000         0.083
[Model 3 versus model 4]
  delta.chisq      delta.df delta.p.value     delta.cfi
       40.502         3.000         0.000         0.042
>
```

Create a data set with non-invariant factor loadings

```
> set.seed(42)
> fx <- matrix(c(.8,.7,.6,rep(0,6),.6,.7,.8),ncol=2)
> fx
     [,1] [,2]
[1,]  0.8  0.0
[2,]  0.7  0.0
[3,]  0.6  0.0
[4,]  0.0  0.6
[5,]  0.0  0.7
[6,]  0.0  0.8
> Phi <- matrix(c(1,.6,.6,1),ncol=2)
> Phi
> set.seed(42)
> x.model <- sim(fx=fx,Phi=Phi,mu=c(0,1),n=250)
>  x <- x.model$observed
>  structure.diagram(fx,Phi,lr=FALSE,e.size=.3,main="A basic two time mod
>  describe(x,skew=FALSE)
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	se
V1	1	250	0.02	0.99	0.01	0.03	1.02	−2.64	2.76	5.40	0.06
V2	2	250	−0.02	0.96	−0.01	−0.03	0.94	−2.66	3.12	5.78	0.06
V3	3	250	0.02	0.97	−0.06	0.01	0.95	−2.74	2.41	5.15	0.06
V4	4	250	0.61	0.99	0.59	0.65	0.99	−3.26	3.15	6.41	0.06

One factor model

```
> f1n <- fa(x)
> f1n

Factor Analysis using method =  minres
Call: fa(r = x)
Standardized loadings (pattern matrix) based upon correlation matrix
    MR1   h2   u2
V1 0.59 0.35 0.65
V2 0.56 0.31 0.69
V3 0.48 0.23 0.77
V4 0.54 0.29 0.71
V5 0.69 0.48 0.52
V6 0.72 0.52 0.48

                MR1
SS loadings    2.18
Proportion Var 0.36

Test of the hypothesis that 1 factor is sufficient.

The degrees of freedom for the null model are
15
        and the objective function was
1.58
         with Chi Square of  389.85
The degrees of freedom for the model are 9
          and the objective function was
0.33

The root mean square of the residuals (RMSR) is
0.07
The df corrected root mean square of the residuals
0.13
The number of observations was  250
with Chi Square =  79.93  with prob <
1.7e-13

Tucker Lewis Index of factoring reliability =
0.684
RMSEA index =  0.179
and the 90 % confidence intervals are
0.143 0.214
BIC =  30.24
Fit based upon off diagonal values = 0.93
Measures of factor score adequacy
                                                 MR1
Correlation of scores with factors
0.89
Multiple R square of scores with factors
0.79
Minimum correlation of possible factor scores
0.57
```

Two factor model

```
> f2n <- fa(x,2)
> f2n

Factor Analysis using method =  minres
Call: fa(r = x, nfactors = 2)
Standardized loadings (pattern matrix) based upon correlation matrix
     MR1   MR2   h2   u2
V1  -0.01  0.79 0.62 0.38
V2  -0.01  0.73 0.53 0.47
V3   0.15  0.40 0.25 0.75
V4   0.51  0.07 0.30 0.70
V5   0.79 -0.03 0.60 0.40
V6   0.80  0.02 0.65 0.35

                       MR1  MR2
SS loadings           1.58 1.37
Proportion Var        0.26 0.23
Cumulative Var        0.26 0.49
Proportion Explained  0.53 0.47
Cumulative Proportion 0.53 1.00

 With factor correlations of
     MR1  MR2
MR1 1.00 0.54
MR2 0.54 1.00

Correlation of scores with factors
0.90 0.88
Multiple R square of scores with factors
0.80 0.77
Minimum correlation of possible factor scores
```

```
Test of the hypothesis that 2 factors are sufficien

The degrees of freedom for the null model are
15
                and the objective function was
1.58
            with Chi Square of  389.85
The degrees of freedom for the model are 4
                   and the objective function was
0.02

The root mean square of the residuals (RMSR) is
0.01
The df corrected root mean square of the residuals
0.04
The number of observations was  250  with
              Chi Square =  4.7
with prob <  0.32

Tucker Lewis Index of factoring reliability =
0.993
RMSEA index =  0.028
and the 90 % confidence intervals are
NA 0.102
BIC =  -17.39
Fit based upon off diagonal values = 1
Measures of factor score adequacy

                                                   MR
MR2
```

Compare the two solutions

```
Factor Analysis using method =  minres
Call: fa(r = x)
Standardized loadings (pattern matrix)
  based upon correlation matrix
    MR1   h2   u2
V1 0.59 0.35 0.65
V2 0.56 0.31 0.69
V3 0.48 0.23 0.77
V4 0.54 0.29 0.71
V5 0.69 0.48 0.52
V6 0.72 0.52 0.48

                MR1
SS loadings    2.18
Proportion Var 0.36
```

```
Factor Analysis using method =  minres
Call: fa(r = x, nfactors = 2)
Standardized loadings (pattern matrix)
          based upon correlation matrix
     MR1   MR2   h2   u2
V1 -0.01  0.79 0.62 0.38
V2 -0.01  0.73 0.53 0.47
V3  0.15  0.40 0.25 0.75
V4  0.51  0.07 0.30 0.70
V5  0.79 -0.03 0.60 0.40
V6  0.80  0.02 0.65 0.35

                       MR1  MR2
SS loadings           1.58 1.37
Proportion Var        0.26 0.23
Cumulative Var        0.26 0.49
Proportion Explained  0.53 0.47
Cumulative Proportion 0.53 1.00

 With factor correlations of
     MR1  MR2
MR1 1.00 0.54
MR2 0.54 1.00
```

CFA 2 correlated factors

```
> y.df <- data.frame(y)
> fitn <- sem(mod2f,data=y.df,std.lv=TRUE)
> summary(fitn , fit.measures=TRUE)

lavaan (0.4-14) converged normally after 17 iterations
  Number of observations
250
  Estimator
ML
  Minimum Function Chi-square
2.552
  Degrees of freedom
8
  P-value
0.959
Chi-square test baseline model:

  Minimum Function Chi-square
341.309
  Degrees of freedom
15
  P-value
0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)
1.000
  Tucker-Lewis Index (TLI)
1.031

Loglikelihood and Information Criteria:
```

```
Root Mean Square Error of Approximation:

  RMSEA
0.000
  90 Percent Confidence Interval
0.000  0.000
  P-value RMSEA <= 0.05
0.994

Standardized Root Mean Square Residual:

  SRMR
0.014

Parameter estimates:

  Information
Expected
  Standard Errors
Standard

                    Estimate  Std.err
Z-value  P(>|z|)
Latent variables:
  F1 =~
    V1                 0.750    0.063
11.880    0.000
    V2                 0.725    0.065
11.160    0.000
    V3                 0.621    0.066
9.410    0.000
  F2 =~
    V1 ...
```

Are the factors measurement invariant? Well, maybe.

```
> summary(fitn2a, fit.measures=TRUE)

lavaan (0.4-14) converged normally after 14 iterations

  Number of observations
250

  Estimator
ML
  Minimum Function Chi-square
8.093
  Degrees of freedom
11
  P-value
0.705

Chi-square test baseline model:

  Minimum Function Chi-square
341.309
  Degrees of freedom
15
  P-value
0.000

Full model versus baseline model:

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

```
Root Mean Square Error of Approximation:

  RMSEA
0.000
  90 Percent Confidence Interval
0.000  0.051
  P-value RMSEA <= 0.05
0.946

Standardized Root Mean Square Residual:

  SRMR
0.049

Parameter estimates:

  Information
Expected
  Standard Errors
Standard

                         Estimate  Std.err
Z-value  P(>|z|)
Latent variables:
  F1 =~
    V1        (a)        0.686    0.050
13.774    0.000
    V2        (b)        0.675    0.049
13.834    0.000
    V3        (c)        0.657    0.048
13.677    0.000
  F2 =~
```

Create the original data set again

```
> fx <- matrix(c(.8,.7,.6,rep(0,6),.8,.7,.6),ncol=2)
> x.model <- sim(fx=fx,Phi=Phi,mu=c(0,1),n=250)
> x <- x.model$observed
> x.df <- data.frame(x)

> describe(x,skew=FALSE)
   var   n mean   sd median trimmed  mad   min  max range   se
V1   1 250 0.02 0.99  -0.01   -0.01 1.04 -2.38 3.26  5.64 0.06
V2   2 250 0.13 1.07   0.03    0.11 1.08 -2.53 3.65  6.19 0.07
V3   3 250 0.03 1.03   0.05    0.02 0.97 -2.68 3.12  5.80 0.07
V4   4 250 0.87 1.04   0.90    0.85 1.03 -1.63 3.61  5.24 0.07
V5   5 250 0.73 0.97   0.76    0.75 0.90 -2.36 3.29  5.65 0.06
V6   6 250 0.65 1.00   0.67    0.64 1.03 -2.58 3.17  5.75 0.06
```

Model change in the items

```
mod2fc <- 'F1 =~ a*V1 + b* V2 + c*V3
                F2 =~ a* V4 + b*V5 +c* V6
                   #correlation between factors
                  F2 ~  F1 '       #the regression
fit2c <- sem(mod2fc,data=x.df,meanstructure=TRUE)
summary(fit2c , fit.measures=TRUE)
```

Correlated factors

```
lavaan (0.4-14) converged normally after 23 iterations

  Number of observations
250

  Estimator
ML
  Minimum Function Chi-square
10.616
  Degrees of freedom
10
  P-value
0.388

Chi-square test baseline model:

  Minimum Function Chi-square
406.795
  Degrees of freedom
15
  P-value
0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)
```

```
Root Mean Square Error of Approximation:

  RMSEA
0.016
  90 Percent Confidence Interval
0.000  0.071
  P-value RMSEA <= 0.05
0.793

Standardized Root Mean Square Residual:

  SRMR
0.032

Parameter estimates:

  Information
Expected
  Standard Errors
Standard

                    Estimate  Std.err
Z-value  P(>|z|)
Latent variables:
  F1 =~
    V1        (a)     1.000
    V2        (b)     0.925    0.076
12.243    0.000
    V3        (c)     0.816    0.072
11.386    0.000
  F2 =~
    V4        (a)     1.000
```

With the intercepts

```
Intercepts:
    V1                   0.021      0.063
0.332      0.740
    V2                   0.135      0.066
2.033      0.042
    V3                   0.032      0.066
0.482      0.630
    V4                   0.865      0.065
13.294     0.000
    V5                   0.729      0.062
11.696     0.000
    V6                   0.649      0.063
10.369     0.000
    F1                   0.000
    F2                   0.000
```

```
Variances:
    V1                   0.383      0.060
    V2                   0.576      0.068
    V3                   0.670      0.071
    V4                   0.486      0.065
    V5                   0.481      0.061
    V6                   0.596      0.065
    F1                   0.613      0.088
    F2                   0.319      0.063
```

Try fitting a moments model

```
mod2fc <- 'F1 =~ a*V1 + b* V2 + c*V3
                 F2 =~ a* V4 + b*V5 +c* V6
                 means = ~ F1 + F2 + 1*V1 +1*V2+1*V3+1*V4+1*V5+1*V6
                   #correlation between factors

                 F2 ~ F1 '       #the regression
fit2c <- sem(mod2fc, data=x.df, meanstructure=TRUE)
summary(fit2c, fit.measures=TRUE)
```

Using the moments matrix

Modeling change using the moments matrix

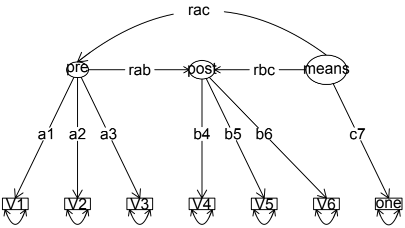
1. McArdle (2009) Latent variable modeling of differences and changes with longitudinal data. Annual Review of Psychology, 60, 577-605
   - Use moments rather than covariances

Create the model to be fit in sem

```
 fxg
     pre   post  means
V1   "a1"  "0"   "0"
V2   "a2"  "0"   "0"
V3   "a3"  "0"   "0"
V4   "0"   "b4"  "0"
V5   "0"   "b5"  "0"
V6   "0"   "b6"  "0"
one  "0"   "0"   "c7"
> phi
   F1     F2   F3
F1 "1"    "0"  "rac"
F2 "rab"  "1"  "rbc"
F3 "0"    "0"  "1"

mod.mom1 <- structure.diagram(fog, phi, errors=TRUE)
```

Modeling the means in a moments matrix

Structural model



the basic path model, with some editing

```
mod.mom1
      Path              Parameter Value
 [1,] "pre->V1"         "a1"      NA
 [2,] "pre->V2"         "a2"      NA
 [3,] "pre->V3"         "a3"      NA
 [4,] "post->V4"        "b4"      NA
 [5,] "post->V5"        "b5"      NA
 [6,] "post->V6"        "b6"      NA
 [7,] "means->one"      "c7"      NA
 [8,] "V1<->V1"         "x1e"     NA
 [9,] "V2<->V2"         "x2e"     NA
[10,] "V3<->V3"         "x3e"     NA
[11,] "V4<->V4"         "x4e"     NA
[12,] "V5<->V5"         "x5e"     NA
[13,] "V6<->V6"         "x6e"     NA
[14,] "one<->one"       NA    "1"        <-- edited
[15,] "pre␣->post"      "rF1F2"   NA
[16,] "means->pre"      "rF3F1"   NA      <- edited
[17,] "means->post"     "rF3F2"   NA     <- edited
[18,] "pre<->pre"       NA        "1"
[19,] "post<->post"     NA        "1"
[20,] "means<->means"   NA        "1"
attr(,"class")
[1] "mod"
```

sem output

```
> sem.mom1 <- sem(mod.mom, MomMat, N=250, raw=TRUE)
> summary(sem.mom1)


Model fit to raw moment matrix.

 Model Chisquare =   12.077    Df =
12
                       Pr(>Chisq) = 0.43954
 AIC =   44.077
 AICc =   14.411
 BIC =   100.42
 CAIC =   -66.181

 Normalized Residuals
   Min.  1st Qu.   Median
Mean  3rd Qu.      Max.
-1.2500  -0.1150   0.0000  -0.0103
0.0972   0.9730
```

```
 Parameter Estimates
      Estimate  Std Error  z value  Pr(>|z|)
a1    0.830111  0.069823  11.8888
1.3536e-32 V1 <--- pre
a2    0.881776  0.076506  11.5256
9.7982e-31 V2 <--- pre
a3    0.698963  0.074824   9.3415
9.5005e-21 V3 <--- pre
b4    0.664838  0.062395  10.6553
1.6468e-26 V4 <--- post
b5    0.554498  0.053510  10.3626
3.6687e-25 V5 <--- post
b6    0.510270  0.051499   9.9084
3.8265e-23 V6 <--- post
c7    1.118034  0.050000
22.3607 9.5054e-111 one <--- means
x1e   0.525613  0.078258   6.7164
1.8623e-11 V1 <--> V1
x2e   0.673405  0.093381   7.2114
5.5387e-13 V2 <--> V2
x3e   0.836504  0.090362   9.2572
2.0983e-20 V3 <--> V3
x4e   0.567286  0.081353   6.9732
3.0990e-12 V4 <--> V4
x5e   0.628041  0.073463   8.5490
1.2412e-17 V5 <--> V5
x6e   0.750522  0.080300   9.3465
9.0592e-21 V6 <--> V6
rF1F2 0.893184  0.142998   6.2461
4.2076e-10 post <--- pre
rF3F1 0.086583  0.073172   1.1833
2.3669e-01 pre <--- means
```

Traits and States and time

1. With just two time points, traits and states are confounded
   - Is the correlation a trait like stability
   - or does the state dissipate slowly?
2. With > 2 time points we can distinguish states and traits
   - States should have an autocorrelation component
   - Traits should be consistent across time
3. Consider the simplex structure of 4 time points
   - Clean within time factor structure
   - Simplex across time points

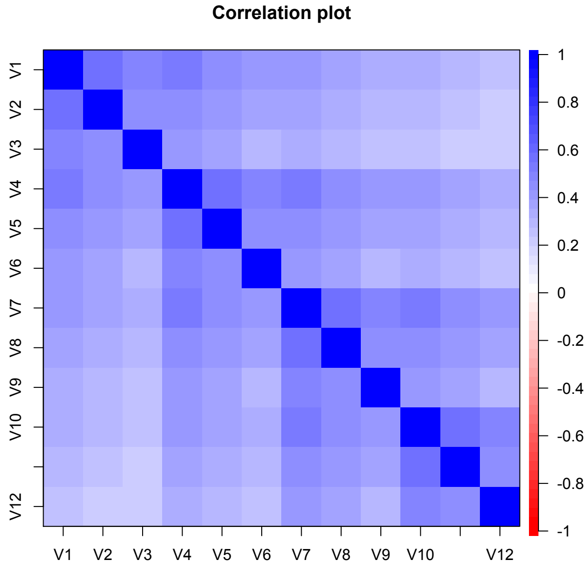
A factor simplex

```
simp <- sim()
```

$model (Population correlation **matrix**)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
V1	1.00	0.56	0.48	0.51	0.45	0.38	0.41	0.36	0.31	0.33	0.29	0.25
V2	0.56	1.00	0.42	0.45	0.39	0.34	0.36	0.31	0.27	0.29	0.25	0.22
V3	0.48	0.42	1.00	0.38	0.34	0.29	0.31	0.27	0.23	0.25	0.22	0.18
V4	0.51	0.45	0.38	1.00	0.56	0.48	0.51	0.45	0.38	0.41	0.36	0.31
V5	0.45	0.39	0.34	0.56	1.00	0.42	0.45	0.39	0.34	0.36	0.31	0.27
V6	0.38	0.34	0.29	0.48	0.42	1.00	0.38	0.34	0.29	0.31	0.27	0.23
V7	0.41	0.36	0.31	0.51	0.45	0.38	1.00	0.56	0.48	0.51	0.45	0.38
V8	0.36	0.31	0.27	0.45	0.39	0.34	0.56	1.00	0.42	0.45	0.39	0.34
V9	0.31	0.27	0.23	0.38	0.34	0.29	0.48	0.42	1.00	0.38	0.34	0.29
V10	0.33	0.29	0.25	0.41	0.36	0.31	0.51	0.45	0.38	1.00	0.56	0.48
V11	0.29	0.25	0.22	0.36	0.31	0.27	0.45	0.39	0.34	0.56	1.00	0.42
V12	0.25	0.22	0.18	0.31	0.27	0.23	0.38	0.34	0.29	0.48	0.42	1.00

A simplex


Correlation plot
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V12
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

Factor structure of a simplex

```
> fsimp <- fa(simp$model)
> fsimp


Factor Analysis using method =  minres
Call: fa(r = simp$model)
Standardized loadings (pattern matrix) based upon correlation matrix
     MR1  h2   u2
V1   0.64 0.41 0.59
V2   0.57 0.33 0.67
V3   0.49 0.24 0.76
V4   0.73 0.54 0.46
V5   0.65 0.42 0.58
V6   0.56 0.31 0.69
V7   0.73 0.54 0.46
V8   0.65 0.42 0.58
V9   0.56 0.31 0.69
V10  0.64 0.41 0.59
V11  0.57 0.33 0.67
V12  0.49 0.24 0.76

                MR1
SS loadings    4.50
Proportion Var 0.38
```

Factors over time

```
> fsimp4 <- fa(simp$model,4)
> fsimp4
```

```
Factor Analysis using method =  minres
Call: fa(r = simp$model, nfactors = 4)
Standardized loadings (pattern matrix)
             based upon correlation matrix
    MR3 MR1 MR2 MR4   h2   u2
V1  0.8 0.0 0.0 0.0 0.64 0.36
V2  0.7 0.0 0.0 0.0 0.49 0.51
V3  0.6 0.0 0.0 0.0 0.36 0.64
V4  0.0 0.0 0.0 0.8 0.64 0.36
V5  0.0 0.0 0.0 0.7 0.49 0.51
V6  0.0 0.0 0.0 0.6 0.36 0.64
V7  0.0 0.8 0.0 0.0 0.64 0.36
V8  0.0 0.7 0.0 0.0 0.49 0.51
V9  0.0 0.6 0.0 0.0 0.36 0.64
V10 0.0 0.0 0.8 0.0 0.64 0.36
V11 0.0 0.0 0.7 0.0 0.49 0.51
V12 0.0 0.0 0.6 0.0 0.36 0.64
```

```
                       MR3  MR1  MR2  MR4
SS loadings           1.49 1.49 1.49 1.49
Proportion Var        0.12 0.12 0.12 0.12
Cumulative Var        0.12 0.25 0.37 0.50
Proportion Explained  0.25 0.25 0.25 0.25
Cumulative Proportion 0.25 0.50 0.75 1.00

 With factor correlations of
     MR3  MR1  MR2  MR4
MR3 1.00 0.64 0.51 0.80
MR1 0.64 1.00 0.80 0.80
MR2 0.51 0.80 1.00 0.64
MR4 0.80 0.80 0.64 1.00
```

Bollen, K. A. (2002). Latent variables in psychology and the social sciences. *Annual Review of Psychology*, *53*, 605–634.

Fox, J., Nie, Z., & Byrnes, J. (2013). *sem: Structural Equation Models*. R package version 3.1-3.

Kerckhoff, A. C. (1974). *Ambition and Attainment: A Study of Four Samples of American Boys.* Washington, D. C.: American Sociological Association.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36.