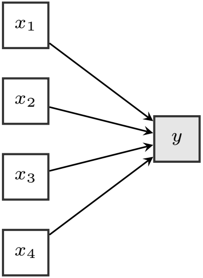
Using the lavaan R package

February 14, 2017

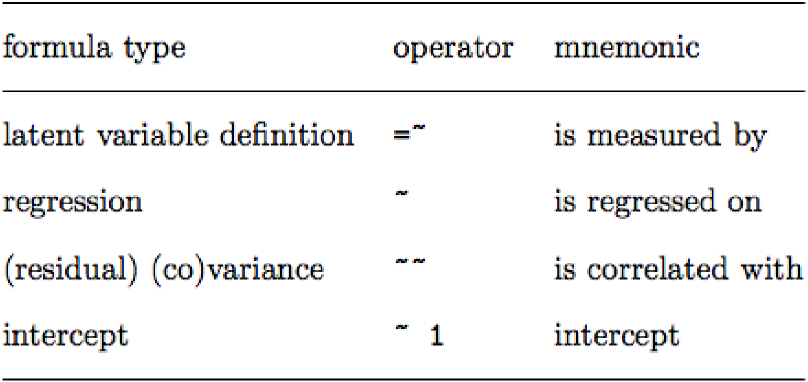
a simple regression analysis in R



```
# read in your data
myData <- read.csv("c:/temp/myData.csv")

# fit model using lm
fit <- lm(formula = y ~ x1 + x2 + x3 + x4,
          data    = myData)

# show results
summary(fit)
```

The standard linear model:

- $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i \quad (i = 1, 2, \ldots, n)$



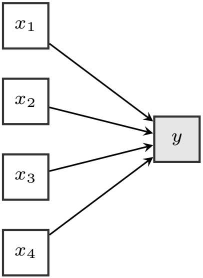
formula type	operator	mnemonic
latent variable definition	=~	is measured by
regression	~	is regressed on
(residual) (co)variance	~~	is correlated with
intercept	~ 1	intercept

lm() output artificial data (N=100)

```
Call:
lm(formula = y ~ x1 + x2 + x3 + x4, data = myData)

Residuals:
     Min       1Q   Median       3Q      Max 
-102.372  -29.458   -3.658   27.275  148.404 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  97.7210     4.7200  20.704   <2e-16 ***
x1            5.7733     0.5238  11.022   <2e-16 ***
x2           -1.3214     0.4917  -2.688   0.0085 ** 
x3            1.1350     0.4575   2.481   0.0149 *  
x4            0.2707     0.4779   0.566   0.5724    
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1 

Residual standard error: 46.74 on 95 degrees of freedom
Multiple R-squared: 0.5911, Adjusted R-squared: 0.5738 
F-statistic: 34.33 on 4 and 95 DF,  p-value: < 2.2e-16 
```

```
# create artificial data
set.seed(1)
x1 <- rnorm(100) * 10; x2 <- rnorm(100) * 10
x3 <- rnorm(100) * 10;  x4 <- rnorm(100) * 10
y <- 100 + 5*x1 + (-2)*x2 + 1*x3 + 0.1*x4 + rnorm(100, sd=40)
myData <- data.frame(y,x1,x2,x3,x4)
```

the lavaan model syntax – a simple regression



```
library(lavaan)
myData <- read.csv("c:/temp/myData.csv")

myModel <- ' y ~ x1 + x2 + x3 + x4 '

# fit model
fit <- sem(model = myModel,
           data  = myData)

# show results
summary(fit)
```

- to ‘see’ the intercept, use either

  ```
  fit <- sem(model=myModel, data=myData, meanstructure=TRUE)
  ```

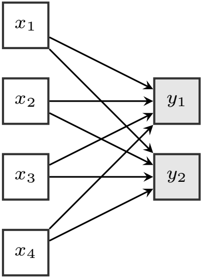
  or include it explicitly in the syntax:

  ```
  myModel <- ' y ~ 1 + x1 + x2 + x3 + x4 '
  ```

output (artificial data, N=100)

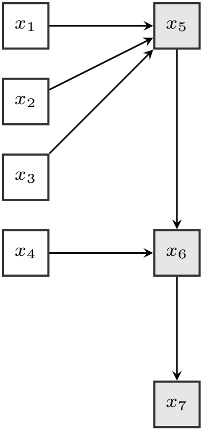
```
lavaan (0.5-13) converged normally after   1 iterations

  Number of observations                           100

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0
  P-value (Chi-square)                           1.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  y ~
    x1                5.773    0.511   11.309    0.000
    x2               -1.321    0.479   -2.757    0.006
    x3                1.135    0.446    2.545    0.011
    x4                0.271    0.466    0.581    0.561

Variances:
    y              2075.100  293.463
```

the lavaan model syntax – multivariate regression



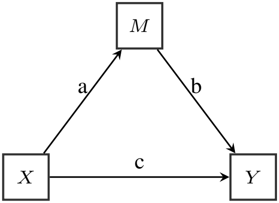
```
myModel <- ' y1 ~ x1 + x2 + x3 + x4
             y2 ~ x1 + x2 + x3 + x4 '
```

the lavaan model syntax – path analysis



```
myModel <- ' x5 ~ x1 + x2 + x3
             x6 ~ x4 + x5
             x7 ~ x6          '
```

the lavaan model syntax – mediation analysis

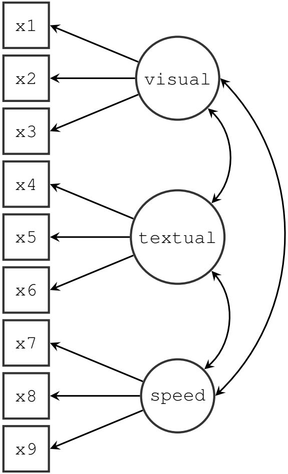


```
model <- '
           Y ~ b*M + c*X
           M ~ a*X

           indirect := a*b
           total    := c + (a*b)
         '

fit <- sem(model,
           data = myData,
           se = "bootstrap")
summary(fit)
```

output

```
...

Parameter estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             1000
  Number of successful bootstrap draws            1000

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  Y ~
    M         (b)     0.597    0.098    6.068    0.000
    X         (c)     2.594    1.210    2.145    0.032
  M ~
    X         (a)     2.739    0.999    2.741    0.006

Variances:
    Y               108.700   17.747
    M               105.408   16.556

Defined parameters:
    indirect          1.636    0.645    2.535    0.011
    total             4.230    1.383    3.059    0.002
```

the lavaan model syntax – using cfa() or sem()

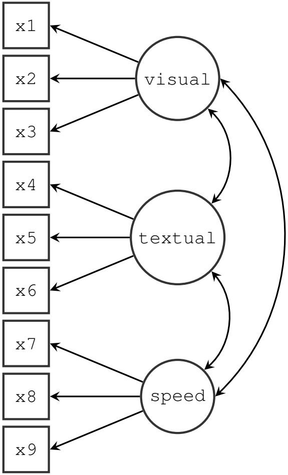


```
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9
            '

fit <- cfa(model = HS.model,
           data  = HolzingerSwineford1939)

summary(fit, fit.measures = TRUE,
             standardized = TRUE)
```

output

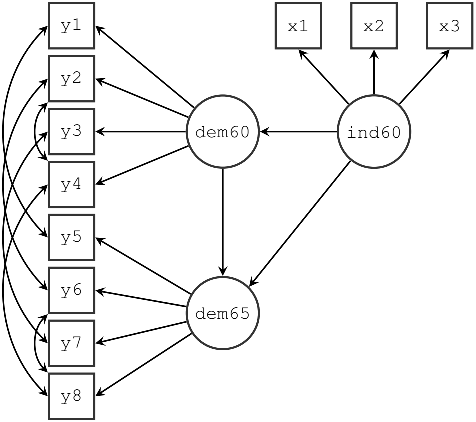
```
lavaan (0.5-13) converged normally after  35 iterations

  Number of observations                           301

  Estimator                                         ML
  Minimum Function Test Statistic               85.306
  Degrees of freedom                                24
  P-value (Chi-square)                           0.000

Model test baseline model:

  Minimum Function Test Statistic              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

  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|)   Std.lv  Std.all
Latent variables:
  visual =˜
    x1                1.000                               0.900    0.772
    x2                0.554    0.100    5.554    0.000    0.498    0.424
    x3                0.729    0.109    6.685    0.000    0.656    0.581
  textual =˜
    x4                1.000                               0.990    0.852
    x5                1.113    0.065   17.014    0.000    1.102    0.855
    x6                0.926    0.055   16.703    0.000    0.917    0.838
  speed =~
    x7                1.000                               0.619    0.570
    x8                1.180    0.165    7.152    0.000    0.731    0.723
    x9                1.082    0.151    7.155    0.000    0.670    0.665

Covariances:
  visual ~~
    textual           0.408    0.074    5.552    0.000    0.459    0.459
    speed             0.262    0.056    4.660    0.000    0.471    0.471
  textual ~~
    speed             0.173    0.049    3.518    0.000    0.283    0.283

Variances:
    x1                0.549    0.114                      0.549    0.404
    x2                1.134    0.102                      1.134    0.821
    x3                0.844    0.091                      0.844    0.662
    x4                0.371    0.048                      0.371    0.275
    x5                0.446    0.058                      0.446    0.269
    x6                0.356    0.043                      0.356    0.298
    x7                0.799    0.081                      0.799    0.676
    x8                0.488    0.074                      0.488    0.477
    x9                0.566    0.071                      0.566    0.558
    visual            0.809    0.145                      1.000    1.000
    textual           0.979    0.112                      1.000    1.000
    speed             0.384    0.086                      1.000    1.000
```

the lavaan model syntax – using lavaan()



```
HS.model <- '
  # latent variables
    visual  =~ 1*x1 + x2 + x3
    textual =~ 1*x4 + x5 + x6
    speed   =~ 1*x7 + x8 + x9

  # factor (co)variances
    visual  ~~ visual; visual  ~~ textual
    visual  ~~ speed;  textual ~~ textual
    textual ~~ speed;  speed   ~~ speed

  # residual variances
    x1 ~~ x1; x2 ~~ x2; x3 ~~ x3
    x4 ~~ x4; x5 ~~ x5; x6 ~~ x6
    x7 ~~ x7; x8 ~~ x8; x9 ~~ x9
'

fit <- lavaan(model = HS.model,
              data  = HolzingerSwineford1939)
```

lavaan model syntax: full sem



```
myModel <- ’

# latent variable definitions
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + a*y2 + b*y3 + c*y4
dem65 =~ y5 + a*y6 + b*y7 + c*y8

# regressions
dem60 ~ ind60
dem65 ~ ind60 + dem60

# residual covariances
y1 ~~ y5
y2 ~~ y4
y2 ~~ y6
y3 ~~ y7
y4 ~~ y8
y6 ~~ y8

,

fit <- sem(model = myModel,
           data  = ...)
```

output

```
lavaan (0.5-13) converged normally after  61 iterations

  Number of observations                            75

  Estimator                                         ML
  Minimum Function Test Statistic               40.179
  Degrees of freedom                                38
  P-value (Chi-square)                           0.374

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  ind60 =~
    x1                1.000
    x2                2.180    0.138   15.751    0.000
    x3                1.818    0.152   11.971    0.000
  dem60 =~
    y1                1.000
    y2        (a)     1.191    0.139    8.551    0.000
    y3        (b)     1.175    0.120    9.755    0.000
    y4        (c)     1.251    0.117   10.712    0.000
  dem65 =~
```

```
    y5                1.000
    y6        (a)     1.191    0.139    8.551    0.000
    y7        (b)     1.175    0.120    9.755    0.000
    y8        (c)     1.251    0.117   10.712    0.000

Regressions:
  dem60 ~
    ind60             1.471    0.392    3.750    0.000
  dem65 ~
    ind60             0.600    0.226    2.660    0.008
    dem60             0.865    0.075   11.554    0.000

Covariances:
  y1 ~~
    y5                0.583    0.356    1.637    0.102
  y2 ~~
    y4                1.440    0.689    2.092    0.036
    y6                2.183    0.737    2.960    0.003
  y3 ~~
    y7                0.712    0.611    1.165    0.244
  y4 ~~
    y8                0.363    0.444    0.817    0.414
  y6 ~~
    y8                1.372    0.577    2.378    0.017

Variances:
    x1                0.081    0.019
    x2                0.120    0.070
```

```
x3                 0.467    0.090
y1                 1.855    0.433
y2                 7.581    1.366
y3                 4.956    0.956
y4                 3.225    0.723
y5                 2.313    0.479
y6                 4.968    0.921
y7                 3.560    0.710
y8                 3.308    0.704
ind60              0.449    0.087
dem60              3.875    0.866
dem65              0.164    0.227
```

shortcut: robust standard errors and scaled test statistic

```
> fit <- cfa(HS.model,
             data = HolzingerSwineford1939,
             estimator = "MLM")
> summary(fit, fit.measures=TRUE)
lavaan (0.5-13) converged normally after  35 iterations

  Number of observations                           301

  Estimator                                         ML      Robust
  Minimum Function Test Statistic               85.306      80.872
  Degrees of freedom                                24          24
  P-value (Chi-square)                           0.000       0.000
  Scaling correction factor                                  1.055
    for the Satorra-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              918.852     789.298
  Degrees of freedom                                36          36
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.931       0.925
  Tucker-Lewis Index (TLI)                       0.896       0.887
...
```

binary and ordered categorical data

```
# binary version of Holzinger & Swineford
HS9 <- HolzingerSwineford1939[,c("x1","x2","x3","x4","x5",
                                 "x6","x7","x8","x9")]
HSbinary <- as.data.frame( lapply(HS9, cut, 2, labels=FALSE) )

# single factor model
model <- ’ visual  =˜ x1 + x2 + x3
           textual =˜ x4 + x5 + x6
           speed   =˜ x7 + x8 + x9 ’

# binary CFA
fit <- cfa(model, data=HSbinary, ordered=names(HSbinary))

# summary
summary(fit, fit.measures=TRUE)
```

output

```
lavaan (0.5-13) converged normally after  36 iterations

  Number of observations                           301

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic               30.918      38.546
  Degrees of freedom                                24          24
  P-value (Chi-square)                           0.156       0.030
  Scaling correction factor                                  0.866
  Shift parameter                                            2.861
    for simple second-order correction (Mplus variant)

Model test baseline model:

  Minimum Function Test Statistic              582.533     469.769
  Degrees of freedom                                36          36
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.987       0.966
  Tucker-Lewis Index (TLI)                       0.981       0.950

Root Mean Square Error of Approximation:

  RMSEA                                          0.031       0.045
  90 Percent Confidence Interval          0.000  0.059       0.014  0.070
  P-value RMSEA <= 0.05                          0.848       0.598

Parameter estimates:

  Information                                   Expected
  Standard Errors                             Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.000
    x2                0.900    0.188    4.788    0.000
    x3                0.939    0.197    4.766    0.000
  textual =~
    x4                1.000
    x5                0.976    0.118    8.241    0.000
    x6                1.078    0.125    8.601    0.000
  speed =~
    x7                1.000
    x8                1.569    0.461    3.403    0.001
    x9                1.449    0.409    3.541    0.000

Covariances:
  visual ~~
    textual           0.303    0.061    4.981    0.000
    speed             0.132    0.049    2.700    0.007
  textual ~~
```

```
    speed             0.076    0.046    1.656    0.098

Intercepts:
    visual            0.000
    textual           0.000
    speed             0.000

Thresholds:
    x1|t1            -0.388    0.074   -5.223    0.000
    x2|t1            -0.054    0.072   -0.748    0.454
    x3|t1             0.318    0.074    4.309    0.000
    x4|t1             0.180    0.073    2.473    0.013
    x5|t1            -0.257    0.073   -3.506    0.000
    x6|t1             1.024    0.088   11.641    0.000
    x7|t1             0.231    0.073    3.162    0.002
    x8|t1             1.128    0.092   12.284    0.000
    x9|t1             0.626    0.078    8.047    0.000

Variances:
    x1                0.592
    x2                0.670
    x3                0.640
    x4                0.303
    x5                0.336
    x6                0.191
    x7                0.778
    x8                0.453
    x9                0.534
    visual              0.408    0.112
    textual             0.697    0.101
    speed               0.222    0.094
```

```
> inspect(fit, "sampstat")
$cov
   x1     x2     x3     x4     x5     x6     x7     x8     x9    
x1  1.000                                                        
x2  0.284  1.000                                                 
x3  0.415  0.389  1.000                                          
x4  0.364  0.328  0.232  1.000                                   
x5  0.319  0.268  0.138  0.688  1.000                            
x6  0.422  0.322  0.206  0.720  0.761  1.000                     
x7 -0.048  0.061  0.041  0.200  0.023 -0.029  1.000              
x8  0.159  0.105  0.439 -0.029 -0.059  0.183  0.464  1.000       
x9  0.165  0.210  0.258  0.146  0.183  0.230  0.335  0.403  1.000

$mean
x1 x2 x3 x4 x5 x6 x7 x8 x9 
 0  0  0  0  0  0  0  0  0 

$th
 x1|t1  x2|t1  x3|t1  x4|t1  x5|t1  x6|t1  x7|t1  x8|t1  x9|t1 
-0.388 -0.054  0.318  0.180 -0.257  1.024  0.231  1.128  0.626 
```