The Simple Double Example

```
> rm(list=ls()); options(scipen=999)
> babydouble =
read.table("http://www.utstat.toronto.edu/~brunner/openSEM/data/Babydouble.data.txt")
> dim(babydouble)

[1] 150   3

> head(babydouble)
     W1    W2     Y
1  9.94 12.24 15.23
2 12.42 11.32 14.55
3 10.43 10.40 12.40
4  9.07  9.85 17.09
5 11.04 11.98 16.83
6 10.40 10.85 15.04
> summary(babydouble)
       W1               W2              Y
 Min.   : 6.190   Min.   : 6.76   Min.   : 3.98
 1st Qu.: 8.932   1st Qu.: 9.11   1st Qu.:10.97
 Median : 9.720   Median :10.05   Median :13.22
 Mean   : 9.809   Mean   :10.06   Mean   :13.10
 3rd Qu.:10.655   3rd Qu.:10.99   3rd Qu.:15.46
 Max.   :12.830   Max.   :13.57   Max.   :21.62
> cor(babydouble)
          W1        W2         Y
W1 1.0000000 0.5748331 0.1714324
W2 0.5748331 1.0000000 0.1791539
Y  0.1714324 0.1791539 1.0000000
>
> # Try ordinary least squares
>
> model1 = lm(Y ~ W1 + W2, data = babydouble); summary(model1)

Call:
lm(formula = Y ~ W1 + W2, data = babydouble)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6793 -2.3881 -0.1321  2.3420  7.6915

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.9695     2.1153   3.768 0.000238 ***
W1            0.2361     0.2282   1.035 0.302538
W2            0.2802     0.2300   1.218 0.225009
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 3.182 on 147 degrees of freedom
Multiple R-squared:  0.03909, Adjusted R-squared:  0.02602
F-statistic:  2.99 on 2 and 147 DF,  p-value: 0.05334
```

```
>
> # install.packages("lavaan", dependencies = TRUE)
> library(lavaan)
This is lavaan 0.5-23.1097
lavaan is BETA software! Please report any bugs.
>
> # The model string
> dmodel1 =  'Y ~ beta1*X         # Latent variable model (even though Y is
observed)
+             X =~ 1*W1 + 1*W2    # Measurement model
+             # Variances (covariances would go here too)
+             X~~phi*X      # Var(X) = phi
+             Y~~psi*Y      # Var(epsilon) = psi
+             W1~~omega1*W1 # Var(e1) = omega1
+             W2~~omega2*W2 # Var(e2) = omega2
+         '
>
> dfit1 = lavaan(dmodel1, data=babydouble)
>
> summary(dfit1)
lavaan (0.5-23.1097) converged normally after  23 iterations

  Number of observations                           150

  Estimator                                         ML
  Minimum Function Test Statistic                0.007
  Degrees of freedom                                 1
  P-value (Chi-square)                           0.933

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  X =~
    W1                1.000
    W2                1.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.707    0.290    2.442    0.015

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.104    0.181    6.104    0.000
   .Y        (psi)    9.775    1.153    8.481    0.000
   .W1      (omg1)    0.834    0.158    5.265    0.000
   .W2      (omg2)    0.800    0.156    5.123    0.000

>
```

```
> parameterEstimates(dfit1)
  lhs op rhs  label   est    se     z pvalue ci.lower ci.upper
1   Y  ~   X  beta1 0.707 0.290 2.442  0.015    0.140    1.275
2   X =~  W1        1.000 0.000    NA     NA    1.000    1.000
3   X =~  W2        1.000 0.000    NA     NA    1.000    1.000
4   X ~~   X    phi 1.104 0.181 6.104  0.000    0.750    1.459
5   Y ~~   Y    psi 9.775 1.153 8.481  0.000    7.516   12.034
6  W1 ~~  W1 omega1 0.834 0.158 5.265  0.000    0.524    1.145
7  W2 ~~  W2 omega2 0.800 0.156 5.123  0.000    0.494    1.105


> parTable(dfit1)
  id lhs op rhs user block group free ustart exo  label plabel start   est    se
1  1   Y  ~   X    1     1     1    1     NA   0  beta1   .p1. 0.000 0.707 0.290
2  2   X =~  W1    1     1     1    0      1   0          .p2. 1.000 1.000 0.000
3  3   X =~  W2    1     1     1    0      1   0          .p3. 1.000 1.000 0.000
4  4   X ~~   X    1     1     1    2     NA   0    phi   .p4. 0.050 1.104 0.181
5  5   Y ~~   Y    1     1     1    3     NA   0    psi   .p5. 5.164 9.775 1.153
6  6  W1 ~~  W1    1     1     1    4     NA   0 omega1   .p6. 0.968 0.834 0.158
7  7  W2 ~~  W2    1     1     1    5     NA   0 omega2   .p7. 0.953 0.800 0.156
> fitted(dfit1) # Sigma(thetahat)
$cov
   W1     W2     Y
W1  1.939
W2  1.104  1.904
Y   0.781  0.781 10.327

$mean
W1 W2  Y
 0  0  0

> logLik(dfit1)
'log Lik.' -878.512 (df=5)

>
> # Fit a restricted model (restricted by H0)
> dfit1r = lavaan(dmodel1, data=babydouble, constraints = 'omega1==omega2')
> anova(dfit1r,dfit1)
Chi Square Difference Test

       Df  AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
dfit1   1 1767 1782.1 0.0071
dfit1r  2 1765 1777.1 0.0262   0.019189       1     0.8898

> # Put multiple constraints on separate lines, like this.
> dfit1r2 = lavaan(dmodel1, data=babydouble, constraints = 'omega1==omega2
+                                                           phi==1')
> anova(dfit1r2,dfit1)
Chi Square Difference Test

        Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
dfit1    1 1767.0 1782.1 0.0071
dfit1r2  3 1763.4 1772.4 0.3868    0.37978       2     0.8271

>
>
> # For Wald tests: Wtest = function(L,Tn,Vn,h=0) # H0: L theta = h
> source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/Wtest.txt")
> LL = cbind(0,0,0,1,-1); LL
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    1   -1
> Wtest(LL,coef(dfit1),vcov(dfit1))
         W         df    p-value
0.01918586 1.00000000 0.88983498
```

```
>
> # Non-linear functions of the parameters with :=
> dmodel1b = 'Y ~ beta1*X         # Latent variable model
+             X =~ 1*W1 + 1*W2    # Measurement model
+             # Variances (covariances would go here too)
+             X~~phi*X      # Var(X) = phi
+             Y~~psi*Y      # Var(epsilon) = psi
+             W1~~omega1*W1 # Var(e1) = omega1
+             W2~~omega2*W2 # Var(e2) = omega2
+             diff := omega1-omega2
+             rel1 := omega1/(omega1+phi)
+             '
> dfit1b = lavaan(dmodel1b, data=babydouble)
> parameterEstimates(dfit1b)
   lhs op                 rhs  label   est    se     z pvalue ci.lower ci.upper
1    Y  ~                   X  beta1 0.707 0.290 2.442  0.015    0.140    1.275
2    X =~                  W1        1.000 0.000    NA     NA    1.000    1.000
3    X =~                  W2        1.000 0.000    NA     NA    1.000    1.000
4    X ~~                   X    phi 1.104 0.181 6.104  0.000    0.750    1.459
5    Y ~~                   Y    psi 9.775 1.153 8.481  0.000    7.516   12.034
6   W1 ~~                  W1 omega1 0.834 0.158 5.265  0.000    0.524    1.145
7   W2 ~~                  W2 omega2 0.800 0.156 5.123  0.000    0.494    1.105
8 diff :=       omega1-omega2   diff 0.035 0.252 0.139  0.890   -0.458    0.528
9 rel1 := omega1/(omega1+phi)   rel1 0.430 0.066 6.540  0.000    0.301    0.559
> sqrt(0.01918586) # Z = sqrt(W)
[1] 0.138513
>
> # Fitting non-identified models
> # Maybe just the first one ...
>
> dmodel2  =  'Y ~ beta1*X         # Latent variable model
+             X =~ lambda1*W1 + lambda2*W2    # Measurement model
+             # Variances (covariances would go here too)
+             X~~phi*X      # Var(X) = phi
+             Y~~psi*Y      # Var(epsilon) = psi
+             W1~~omega1*W1 # Var(e1) = omega1
+             W2~~omega2*W2 # Var(e2) = omega2
+             '
> dfit2 = lavaan(dmodel2, data=babydouble)
Warning message:
In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING: could not compute standard errors!
  lavaan NOTE: this may be a symptom that the model is not identified.
```

```
> summary(dfit2)
lavaan (0.5-23.1097) converged normally after  25 iterations

  Number of observations                           150

  Estimator                                         ML
  Minimum Function Test Statistic                   NA
  Degrees of freedom                                -1

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  X =~
    W1      (lmb1)    1.022       NA
    W2      (lmb2)    1.060       NA

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.736       NA

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.019       NA
   .Y        (psi)    9.776       NA
   .W1      (omg1)    0.871       NA
   .W2      (omg2)    0.761       NA

>
>
>
>
> # dmodel3 passes the parameter count rule, but its parameters are not
identifiable.
> dmodel3  = 'Y ~ beta1*X                     # Latent variable model
+             X =~ lambda1*W1 + lambda2*W2    # Measurement model
+             X~~phi*X      # Var(X) = phi
+             Y~~psi*Y      # Var(epsilon) = psi
+             W1~~omega*W1 # Var(e1) = omega
+             W2~~omega*W2 # Var(e2) = omega
+             '
> dfit3 = lavaan(dmodel3, data=babydouble)
> summary(dfit3)
```

```
lavaan (0.5-23.1097) converged normally after  19 iterations

  Number of observations                           150

  Estimator                                         ML
  Minimum Function Test Statistic                0.014
  Degrees of freedom                                 0
  Minimum Function Value               0.0000466299101

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  X =~
    W1      (lmb1)    1.048    0.089   11.797    0.000
    W2      (lmb2)    1.034    0.089   11.658    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.736    0.275    2.671    0.008

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.019    0.087   11.713    0.000
   .Y        (psi)    9.776    1.153    8.481    0.000
   .W1      (omeg)    0.817    0.094    8.660    0.000
   .W2      (omeg)    0.817    0.094    8.660    0.000
```

Is it really okay?

$$
\begin{aligned}
W_{i,1} &= \lambda_1 X_i + e_{i,1} \\
W_{i,2} &= \lambda_2 X_i + e_{i,2} \\
Y_i &= \beta_1 X_i + \epsilon_i,
\end{aligned}
$$

$$
\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \lambda_1^2\phi + \omega & \lambda_1\lambda_2\phi & \lambda_1\beta_1\phi \\ & \lambda_2^2\phi + \omega & \lambda_2\beta_1\phi \\ & & \beta_1^2\phi + \psi \end{pmatrix}
$$

$\boldsymbol{\theta}_1$	λ_1	λ_2	β_1	ϕ	ω	ψ
$\boldsymbol{\theta}_c$	$c\lambda_1$	$c\lambda_2$	$c\beta_1$	$\frac{\phi}{c^2}$	ω	ψ

```
> V = vcov(dfit3) ; det(V)
[1] 0.00000000000000000000000000000000000000000001801361
> H = solve(V)
Error in solve.default(V) :
  system is computationally singular: reciprocal condition number = 1.95535e-19
>


>
> # Start the search at another minimum, right in the river
>
> c = -2
> thetac = coef(dfit3); thetac
  beta1 lambda1 lambda2     phi     psi   omega   omega
  0.736   1.048   1.034   1.019   9.776   0.817   0.817
> thetac[1] = c*thetac[1]; thetac[2] = c*thetac[2]; thetac[3] = c*thetac[3]
> thetac[4] = thetac[4]/c^2
> cat(thetac)
-1.471502 -2.095291 -2.068575 0.2548175 9.775661 0.816833 0.816833
>
>
> dmodel3b = 'Y ~ beta1*X + start(-1.471502)*X
+             X =~ lambda1*W1 + start(-2.095291)*W1 +
+                  lambda2*W2 + start(-2.068575)*W2
+             # Variances (covariances would go here too)
+             X~~phi*X + start(0.2548175)*X      # Var(X) = phi
+             Y~~psi*Y + start(9.775661)*Y       # Var(epsilon) = psi
+             W1~~omega*W1 + start(0.816833)*W1  # Var(e1) = omega
+             W2~~omega*W2 + start(0.816833)*W2  # Var(e2) = omega
+             '
> dfit3b = lavaan(dmodel3b, data=babydouble)
> show(dfit3b)
lavaan (0.5-23.1097) converged normally after   2 iterations

  Number of observations                           150

  Estimator                                         ML
  Minimum Function Test Statistic                0.014
  Degrees of freedom                                 0
  Minimum Function Value               0.0000466299101
> rbind(thetac,coef(dfit3b))
           beta1   lambda1   lambda2       phi      psi    omega    omega
thetac -1.471502 -2.095291 -2.068575 0.2548175 9.775661 0.816833 0.816833
       -1.471502 -2.095291 -2.068575 0.2548175 9.775661 0.816833 0.816833
> rbind(coef(dfit3),coef(dfit3b))
          beta1   lambda1   lambda2       phi      psi    omega    omega
[1,]  0.7357509  1.047646  1.034288 1.0192699 9.775661 0.816833 0.816833
[2,] -1.4715020 -2.095291 -2.068575 0.2548175 9.775661 0.816833 0.816833
> c( logLik(dfit3), logLik(dfit3b) )
[1] -878.5155 -878.5155
```

```
> # Start the search at another place, close to the river
```

$\boldsymbol{\theta}_1$	λ_1	λ_2	β_1	ϕ	ω	ψ
$\boldsymbol{\theta}_c$	$c\lambda_1$	$c\lambda_2$	$c\beta_1$	$\frac{\phi}{c^2}$	ω	ψ

```
>
> dmodel3c = 'Y ~ beta1*X + start(6)*X
+             X =~ lambda1*W1 + start(8)*W1 +
+                  lambda2*W2  + start(8)*W2
+             # Variances (covariances would go here too)
+             X~~phi*X + start(1/64)*X      # Var(X) = phi
+             Y~~psi*Y       # Var(epsilon) = psi
+             W1~~omega*W1   # Var(e1) = omega
+             W2~~omega*W2   # Var(e2) = omega
+             '
> dfit3c = lavaan(dmodel3c, data=babydouble)
> # show(dfit3c)
> rbind( coef(dfit3), coef(dfit3b), coef(dfit3c) )
          beta1   lambda1   lambda2       phi      psi    omega    omega
[1,]  0.7357509  1.047646  1.034288 1.0192699 9.775661 0.816833 0.816833
[2,] -1.4715020 -2.095291 -2.068575 0.2548175 9.775661 0.816833 0.816833
[3,]  5.7803725  8.230750  8.125805 0.0165135 9.775661 0.816833 0.816833
> c( logLik(dfit3), logLik(dfit3b), logLik(dfit3c) )
[1] -878.5155 -878.5155 -878.5155
>
> parTable(dfit3c)
  id  lhs op  rhs user block group free ustart exo   label plabel start   est    se
1  1    Y  ~    X    1     1     1    1  6.000   0   beta1   .p1. 6.000 5.780 1.895
2  2    X =~   W1    1     1     1    2  8.000   0 lambda1   .p2. 8.000 8.231 0.822
3  3    X =~   W2    1     1     1    3  8.000   0 lambda2   .p3. 8.000 8.126 0.819
4  4    X ~~    X    1     1     1    4  0.016   0     phi   .p4. 0.016 0.017 0.004
5  5    Y ~~    Y    1     1     1    5     NA   0     psi   .p5. 5.164 9.776 1.153
6  6   W1 ~~   W1    1     1     1    6     NA   0   omega   .p6. 0.968 0.817 0.094
7  7   W2 ~~   W2    1     1     1    7     NA   0   omega   .p7. 0.953 0.817 0.094
8  8 .p6. == .p7.    2     0     0    0     NA   0                0.000 0.000 0.000
```
