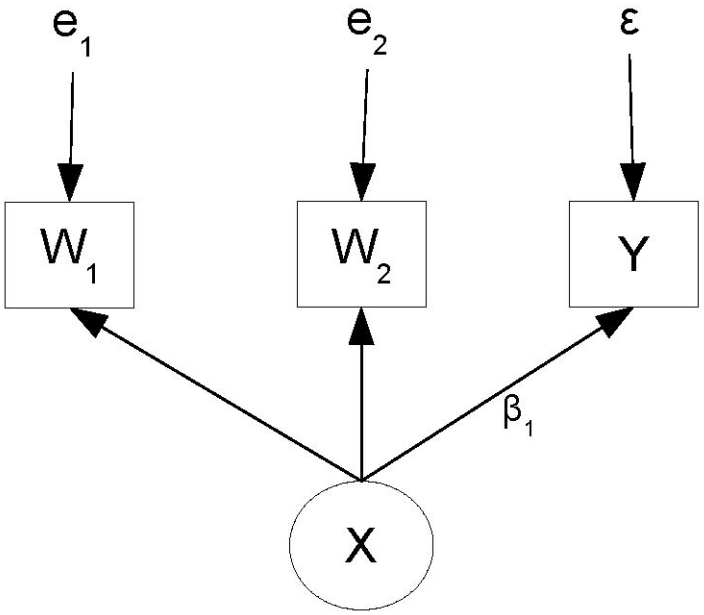
Little Double Measurement Regression Example*




```
> rm(list=ls()); options(scipen=999)
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-11
lavaan is FREE software! Please report any bugs.
>
> babydouble =
read.table("http://www.utstat.toronto.edu/~brunner/openSEM/data/Babydouble.data.txt")
> 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

> dim(babydouble)
[1] 150   3

> 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
```

* Copyright information is on the last page.

```
>
> 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 0.6-11 ended normally after 23 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         5

  Number of observations                           150

Model Test User Model:

  Test statistic                                 0.007
  Degrees of freedom                                 1
  P-value (Chi-square)                           0.933

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

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

> coef(dfit1) # A vector of MLEs
 beta1    phi    psi omega1 omega2
 0.707  1.104  9.775  0.834  0.800

> fitted(dfit1) # Sigma(thetahat) and mu(thetahat)
$cov
    W1     W2     Y
W1  1.939
W2  1.104  1.904
Y   0.781  0.781 10.327

>
> vcov(dfit1)
       beta1  phi    psi    omega1 omega2
beta1   0.084
phi    -0.007  0.033
psi    -0.035  0.002  1.328
omega1  0.003 -0.004 -0.002  0.025
omega2  0.003 -0.005 -0.002 -0.007  0.024

> 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-Squared 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-Squared 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")
> coef(dfit1) # A vector of MLEs
 beta1    phi    psi omega1 omega2
 0.707  1.104  9.775  0.834  0.800

> 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 := phi/(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 := phi/(omega1+phi)   rel1 0.570 0.066 8.657  0.000    0.441    0.699

> sqrt(0.01918586) # Compare Z statistic for H0: omega1=omega2
[1] 0.138513
```

```
> # And one attempt to fit a non-identified model
>
> dmodel0 =  'Y ~ beta1*X    # Latent variable model (even though Y is observed)
+             X =~ 1*W1      # 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
+             '
> dfit0 = lavaan(dmodel0, data=babydouble)

Warning message:
In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING:
    Could not compute standard errors! The information matrix could
    not be inverted. This may be a symptom that the model is not
    identified.

> summary(dfit0)
lavaan 0.6-11 ended normally after 17 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         4

  Number of observations                           150

Model Test User Model:

  Test statistic                                    NA
  Degrees of freedom                                -1
  P-value (Unknown)                                 NA

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

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

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.734       NA

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.044       NA
   .Y        (psi)    9.765       NA
   .W1      (omg1)    0.892       NA
```
