Little Path Example with R

```
> # Littlepath.R
>
> ###################### Simulation of the data ###############################
> # N <- 10000 ; Gamma <- 1 ; Beta <- 2 ; Phi <- 3 ; Psi1 <- 4 ; Psi2 <- 5
> # X <- rnorm(N,0,sqrt(Phi))
> # e1 <- rnorm(N,0,sqrt(Psi1)) ; e2 <- rnorm(N,0,sqrt(Psi2))
> # Y1 <- Gamma*X + e1  ;  Y2 <- Beta*Y1 + e2
> # datta <- cbind(X,Y1,Y2)
> # write(t(datta),"littlepath.dat.txt",ncolumns=3)
> ############################################################################
>
> info <- read.table("littlepath.dat.txt") ; n <- dim(info)[1]
> MLE <- var(info) * (n-1)/n
> print(MLE)
         V1        V2        V3
V1 3.050319  3.025360  6.085787
V2 3.025360  6.875649 13.844049
V3 6.085787 13.844049 32.782936
> print(var(info))
         V1        V2        V3
V1 3.050624  3.025663  6.086395
V2 3.025663  6.876336 13.845434
V3 6.086395 13.845434 32.786214

>
> ###########  Define Functions ##########
> LLMVN <- function(Sig,Sighat,nn)
+ # Minus Log Likelihood for MVN. A function of Sigma, with mu=Xbar
+ # This is general. Usually call it with another function.
+     {
+     pp <- dim(Sig)[1]
+     LLMVN <- nn/2 * ( pp*log(2*pi) + log(det(Sig)) +
+                       sum(diag(Sighat%*%solve(Sig))) )
+     LLMVN
+     } # End of function LLMVN
>
> path1 <- function(theta,Sighat,nn) # - log likelihood for restricted model (5-D)
+       {
+       gamma <- theta[1] ; beta <- theta[2] ; phi <- theta[3]
+       psi1 <- theta[4] ; psi2 <- theta[5]
+       pp <- dim(Sighat)[1] ; Sig <- matrix(nrow=pp,ncol=pp)
+       # Load matrix Sig with functions of theta values
+       Sig[1,1] <- phi ; Sig[1,2] <- gamma*phi ; Sig[2,1] <- Sig[1,2]
+       Sig[1,3] <- gamma*beta*phi ; Sig[3,1] <- Sig[1,3]
+       Sig[2,2] <- gamma^2*phi + psi1
+       Sig[2,3] <- beta*gamma^2*phi + beta*psi1 ; Sig[3,2] <- Sig[2,3]
+       Sig[3,3] <- beta^2*Sig[2,2] + psi2
+       path1 <- LLMVN(Sig,Sighat,nn)
+       path1
+       } # End of function path1
>
>
```

```
> path2 <- function(theta,Sighat,nn) # - log likelihood for full model (6-D)
+       {
+       gamma1 <- theta[1]; gamma2 <- theta[2] ; beta <- theta[3]
+       phi <- theta[4]; psi1 <- theta[5] ; psi2 <- theta[6]
+       pp <- dim(Sighat)[1] ; Sig <- matrix(nrow=pp,ncol=pp)
+       # Load matrix Sig with functions of theta values
+       Sig[1,1] <- phi ; Sig[1,2] <- gamma1*phi ; Sig[2,1] <- Sig[1,2]
+       Sig[1,3] <- (beta*gamma1+gamma2)*phi ; Sig[3,1] <- Sig[1,3]
+       Sig[2,2] <- gamma1^2*phi + psi1
+       Sig[2,3] <- beta*phi*gamma1^2 +  phi*gamma1*gamma2+ beta*psi1
+       Sig[3,2] <- Sig[2,3]
+       Sig[3,3] <- beta^2*Sig[2,2] + 2*beta*phi*gamma1*gamma2 +
+                   gamma2^2*phi + psi2
+       path2 <- LLMVN(Sig,Sighat,nn)
+       path2
+       } # End of function path2
>
> # Fit the restricted model
> # Starting values
> X <- info[,1] ; Y1 <- info[,2] ; Y2 <- info[,3]
> X <- X-mean(X) ; Y1 <- Y1-mean(Y1) ; Y2 <- Y2-mean(Y2) # Center the vars
> reg1 <- lm(Y1~-1+X) ; reg1 # No intercept

Call:
lm(formula = Y1 ~ -1 + X)

Coefficients:
     X
0.9918

> reg1$coefficients
        X
0.9918177
> anova(reg1) ; anova(reg1)[2,3]
Analysis of Variance Table

Response: Y1
            Df Sum Sq Mean Sq F value    Pr(>F)
X            1  30006   30006  7742.6 < 2.2e-16 ***
Residuals 9999  38750       4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
[1] 3.87543
> reg2 <- lm(Y2~-1+Y1)
>
> s1 <- reg1$coefficients
> s2 <- reg2$coefficients
> s3 <- var(X)
> s4 <- anova(reg1)[2,3]
> s5 <- anova(reg2)[2,3]
> startvals1 <- c(s1,s2,s3,s4,s5) ; startvals1
        X        Y1
0.9918177 2.0134899 3.0506238 3.8754303 4.9085737
>
```

```
> redmod <- nlm(path1,startvals1,hessian=T,Sighat=MLE,nn=n)
> print(redmod)
$minimum
[1] 62871.59

$estimate
[1] 0.9918173 2.0134886 3.0503247 3.8749992 4.9081323

$gradient
[1]  0.0004438334 -0.0039026961  0.0040001581 -0.0138515485  0.0107920830

$hessian
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  7.871793e+03  3.613608e-04  7.155918e-04 -1.000796e-01  4.447287e-04
[2,]  3.613608e-04  1.400869e+04 -2.369326e-04 -1.865088e-04 -2.835295e-01
[3,]  7.155918e-04 -2.369326e-04  5.371587e+02  6.155630e-05  1.457972e-04
[4,] -1.000796e-01 -1.865088e-04  6.155630e-05  3.328610e+02  3.825624e-05
[5,]  4.447287e-04 -2.835295e-01  1.457972e-04  3.825624e-05  2.074699e+02

$code
[1] 1

$iterations
[1] 4

> AsCovMat <- solve(redmod$hessian) ; print(AsCovMat)
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.270359e-04 -3.281957e-12 -1.692389e-10  3.819524e-08 -2.723231e-10
[2,] -3.281957e-12  7.138429e-05  3.146006e-11  3.998587e-11  9.755414e-08
[3,] -1.692389e-10  3.146006e-11  1.861647e-03 -3.443269e-10 -1.308208e-09
[4,]  3.819524e-08  3.998587e-11 -3.443269e-10  3.004257e-03 -5.539945e-10
[5,] -2.723231e-10  9.755414e-08 -1.308208e-09 -5.539945e-10  4.819976e-03

> # Z-test for gamma (Wald)
> Z <- redmod$estimate[1]/sqrt(AsCovMat[1,1]) ; Z
[1] 87.99713

>
> # Fit the full model
> # Starting values
> reg1 <- lm(Y1~-1+X) # The same as before
> reg2 <- lm(Y2~-1+Y1+X)
> s1 <- reg1$coefficients         # gamma1
> s2 <- reg2$coefficients[2]      # gamma2
> s3 <- reg2$coefficients[1]      # beta
> s4 <- var(X)                    # phi
> s5 <- anova(reg1)[2,3]          # psi1
> s6 <- anova(reg2)[3,3]          # psi2
> startvals2 <- c(s1,s2,s3,s4,s5,s6) ; startvals2
           X            X           Y1
 0.991817737 -0.003342162  2.014960486  3.050623803  3.875430267  4.909045407
>
```

```
> fullmod <- nlm(path2,startvals2,hessian=T,Sighat=MLE,nn=n)
> print(fullmod)
$minimum
[1] 62871.57

$estimate
[1]  0.99181723 -0.00334113  2.01495899  3.05031846  3.87503462  4.90806472

$gradient
[1] -5.820766e-05  3.055902e-04 -4.694262e-04  6.726576e-04 -2.050393e-03
[6]  7.427072e-04

$hessian
              [,1]          [,2]          [,3]         [,4]          [,5]
[1,]  7.871720e+03 -1.455192e-03 -3.610971e-04 4.770622e-04 -1.012053e-01
[2,] -1.455192e-03  6.214911e+03  6.164059e+03 7.155932e-04 -3.755299e-04
[3,] -3.610971e-04  6.164059e+03  1.400888e+04 1.183801e-04 -1.863710e-04
[4,]  4.770622e-04  7.155932e-04  1.183801e-04 5.371632e+02  0.000000e+00
[5,] -1.012053e-01 -3.755299e-04 -1.863710e-04 0.000000e+00  3.328486e+02
[6,]  0.000000e+00 -6.255936e-02 -2.845036e-01 1.457995e-04 -3.825642e-05
              [,6]
[1,]  0.000000e+00
[2,] -6.255936e-02
[3,] -2.845036e-01
[4,]  1.457995e-04
[5,] -3.825642e-05
[6,]  2.074797e+02

$code
[1] 1

$iterations
[1] 6

>
> # Compare to the - log likelihood at Sigma=Sigmahat
> right <- n/2 * ( 3*log(2*pi) + log(det(MLE)) + 3 ) ; right
[1] 62871.57
>
> # This requires discussion!
>
> # Test for gamma2 is also a goodness of fit test
> G <- 2 * (redmod$minimum-fullmod$minimum) ; print(G)
[1] 0.03912594
>
```