new;
screen on;
starttime = time;
startdate = date;
print "BVPLN, Clear Lake data set, CORRECTED FOR ON-SITE SAMPLE";
print "USING 1000 RNDN DRAWS; seed 46";
output file = 05_BVPLN_screen3.out reset;
n = 389;
ndraw = 1000;
ecol = 2;

load data[] = 03_DMcleardat1.out;
let vnames = id version q Btrip Pthird inc male age sdum hh WQd wq wBtrip;
print "rows in data =" rows(data);
print "rows should be =" n*13;
data = reshape(data,n,13);
id=data[.,1]; ver=data[.,2]; y1=data[.,3]; y2=data[.,4]; Pthird=data[.,5]; inc=data[.,6]; male=data[.,7]; age=data[.,8]; sdum=data[.,9]; hh=data[.,10]; WQd = data[.,11]; wq = data[.,12]; wq2 = data[.,13];

P = Pthird/100;
I = inc/100000;
male = male/100;
age = age/100;
sdum = sdum/100;
hh = hh/100;
WQd = WQd/10;

library maxlik;
#include maxlik.ext;
maxset;

@ Setting various options within the maxlik procedure

/*_max_GradTol=1e-4; @ Change gradient tolerance for convergence, default=1e-5 @*/
_max_Algorithm=5; @ Change to BHHH algorithm, default=2 (DFP) @
_max_Linesearch=5; @ Change to BHHH line search, default=2 (StepBT) @
/*_max_Covpar=2; @ Change the procedure to calc. std. errors @ */

@ Setting starting values for all of the parameters

@ optvals; ndraw=500 @
stval={ 1.9274,
  2.7953,
  -1.5164,
  -1.2795,
  1.3149,
  0.9482,
  -6.8373,
  6.2050,
  -22.5367,
  1.2310,
  1.0010,
  0.9876,
  2.1461 ];

@ Stores output from Maxlik procedure
@ b   = vector of parameter values
@ logl = log-likelihood value
@ g   = final gradient vector
@ cov  = covariance matrix
@ retcode = return code, see maxlik user's manuel

{b,logl,g,cov,retcode} = maxprt(maxlik(data,0,&BVPLN,stval));

@ Maxlik procedure named "BVPLN", using b vector and data
Any variable defined within the proc must be listed as a "local" var

proc BVPLN(b, data);
local cnst1, cnst2, bp1, bp2, bi1, bi2, bm, ba, basq, bs, bh, s1, s2, rho, demo,
    lambda1, lambda2, r, pn, v, lambdaev, eps1, eps2, a, dum;
local llik;

    cnst1 = b[1];
    cnst2 = b[2];
    bp1 = b[3];
    bp2 = b[4];
    bi1 = b[5];
    bi2 = b[6];
    ba = b[7];
    basq = b[8];
    bs = b[9];
    s1 = b[10];
    s2 = b[11];
    rho = b[12];
    dum = b[13];
    /*rho = 1;*/

demo = ba*age + basq*age^2 + bs*sdum;

    r = 1;
    pn = zeros(n, 1);
rndseed 46;
do while r <= ndraw;
a = sqrt(1-rho^2);
esps1 = rndn(n, 1);
esps2 = rndn(n, 1);
lambda1 = exp(cnst1 + bp1*P + bi1*I + demo)*exp(s1*eps1);

    for each iteration, take (n,1) rndm draws from normal dist, then
calc Poisson density and add togethor the result across all of the ndraw iterations; store in pn
\[
\lambda_2 = \exp(\text{cnst}_2 + \text{dum} \times \text{WQd} + \text{bp}_2 \times \text{P} + \text{bi}_2 \times \text{I} + \text{demo}) \times \exp(s_2 \times \rho \times \eps_1 + s_2 \times a \times \eps_2);
\]

/*@ Use these next three lines of code instead, if assume \(\rho=1\); @
\[
\eps_1 = \text{rndn}(n,1);
\]
\[
\lambda_1 = \exp(\text{cnst}_1 + \text{bp}_1 \times \text{P} + \text{bi}_1 \times \text{I} + \text{demo}) \times \exp(s_1 \times \eps_1);
\]
\[
\lambda_2 = \exp(\text{cnst}_2 + \text{dum} \times \text{WQd} + \text{bp}_2 \times \text{P} + \text{bi}_2 \times \text{I} + \text{demo}) \times \exp(s_2 \times \eps_1);
\]

ﾇ = \(\exp(-\lambda_1) \times \lambda_1^y_1 / y_1!\) .\(\exp(-\lambda_2) \times \lambda_2^y_2 / y_2!\);

\[
r = r + 1;
\]

endo;

\[
\text{calc. } \delta_1 \text{ in Egan&Herregis (2006)}
\]
\[
\lambda_\text{ev} = \exp(\text{cnst}_1 + \text{bp}_1 \times \text{P} + \text{bi}_1 \times \text{I} + \text{demo}) \times \exp(0.5 \times s_1^2);
\]

\[
\text{calc. log of the likelihood given in eqn. 22 of Egan&Herriges (2006)}
\]
\[
\text{llik} = \ln(y_1) - \ln(\lambda_\text{ev}) + \ln(n) / \text{ndraw};
\]

\[
\text{retp or "return procedure" commands maxlik procedure to keep iterating}
\]
\[
\text{using the llik variable until maximization of llik is reached;}
\]
\[
\text{endp command, ends the procedure}
\]

retp (llik);
endp;

\[
\text{print total log-lik value (logl stores the avg log-lik)}
\]
\[
\text{print the name of each parameter}
\]

\[
\text{print "log-lik=\ n*logl;}
\]

print "P01=cnst}_1, P02=cnst}_2, P03=P1, P04=P2, P05=I1, P06=I2,
    P07=male, P08=age, P09=age^2, P10=sdum, P11=hh

P12=s1, P13=s2, P14=rho, P15=dum

@ print b (coefficient values) and cov (covariance matrix) to an output file  @
@@ output file = 05_BVPLN_bcov3.out reset;
print b~cov;
output off;

endtime = time; enddate = date;
start_time = timestr(starttime);
end_time = timestr(endtime);
start_date = datestring(startdate);
end_date = datestring(enddate);
print "Start=\" start_date " \" start_time \"; End=\" end_date " \" end_time \";
timediff = ethsec(startdate,enddate);
time_diff = etstr(timediff);
print "Total Running Time=\" time_diff;