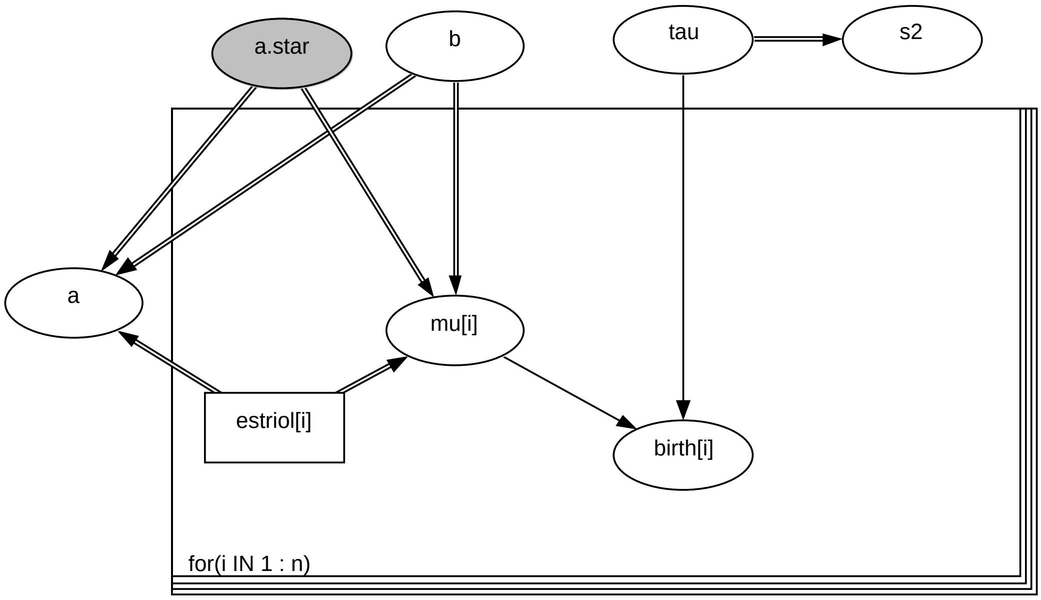
ΠΑΡΑΡΤΗΜΑ Δ1 ($4^{ου}$ ΜΑΘΗΜΑΤΟΣ): ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΛΕΓΧΩΝ ΥΠΟΘΕΣΕΩΝ, ΕΠΙΛΟΓΗΣ ΜΟΝΤΕΛΩΝ ΚΑΙ ΜΕΤΑΒΛΗΤΩΝ

1 ΠΑΡΑΔΕΙΓΜΑ WINBUGS 1: Ένας Απλός Έλεγχος Υπόθεσης (ESTRIOL DATASET)

```
model estriol;
{
#	definition of likelihood function
#
	for (i in 1:n) {
		birth[i]~dnorm( mu[i], tau ); # random component
		mu[i]<-a.star+gamma*b*(estriol[i]-mean(estriol[]));	# systematic component
						#	& link function
	}
#	prior distributions
#
	a.star~dnorm( 0, 1.0E-04 ); # normal prior for a
	b~dnorm( 0, 1.575); # normal prior for b
		gamma~dbern(0.5);
	tau~dgamma( 1.0E-04 , 1.0E-04 ); # gamma prior for precision
	s2<-1/tau;
	a<-a.star-b*mean(estriol[]);
}
list(a.star=0.0, b=0.0, tau=1.0,gamma=1)

list(n=31)
estriol[] birth[]
7	25
9	25
9	25
12	27
14	27
16	27
16	24
14	30
16	30
16	31
17	30
19	31
21	30
24	28
15	32
16	32
17	32
25	32
27	34
15	34
15	34
15	35
16	35
19	34
18	35
17	36
18	37
20	38
22	40
25	39
24	43
END
```

name:	a.star	type:	stochastic	density:	dnorm		
mean	0.0	precision	1.0E-6	lower bound		upper bound	



2 ΠΑΡΑΔΕΙΓΜΑ WINBUGS 2: Διαγνωστικά Τέστ και Συγκρίσεις μοντέλων (ESTRIOL DATASET)

```
model estriol_AIC_BIC;
{
#     definition of likelihood function
#
            pi<-3.14
    for (i in 1:n) {
        birth[i]~dnorm( mu[i], tau );                    # random component
        mu[i]<-a.star+b*(estriol[i]-mean(estriol[]));    # systematic component
                                                         #     & link function
                        birth.pred[i]~dnorm( mu[i], tau )
                        loglike1[i]<- -0.5*log(2*pi)+0.5*log(tau)-0.5*pow( birth[i]-mu[i],2 )*tau
                        loglike1.pred[i]<- -0.5*log(2*pi)+0.5*log(tau)-0.5*pow( birth.pred[i]-mu[i],2 )*tau
                        like1[i]<- exp( loglike1[i] )
#
#                   model m_0
        birth0[i]<-birth[i]
                        birth0[i]~dnorm( mu0[i], tau0 );                # random component
        mu0[i]<-a0;                                      # systematic component
                                                         #     & link function
                        birth0.pred[i]~dnorm( mu0[i], tau0 )
                        loglike0[i]<- -0.5*log(2*pi)+0.5*log(tau0)-0.5*pow( birth0[i]-mu0[i],2 )*tau0
                        loglike0.pred[i]<- -0.5*log(2*pi)+0.5*log(tau0)-0.5*pow( birth0.pred[i]-mu0[i],2 )*tau0
                        like0[i]<- exp( loglike0[i] )
#
#
                        ss1[i] <- pow( birth.pred[i]-birth[i], 2 )
                        ss0[i] <- pow( birth0.pred[i]-birth0[i], 2 )
    }
#     prior distributions for model m1
#
    a.star~dnorm( 0, 1.0E-04 ); # normal prior for a
    b~dnorm( 0, 1.0E-04 ); # normal prior for b
    tau~dgamma( 1.0E-04 , 1.0E-04 ); # gamma prior for precision
    s2<-1/tau;
    a<-a.star-b*mean(estriol[]);
#
#       prior distributions for model m0
    a0~dnorm( 0, 1.0E-04 ); # normal prior for a
    tau0~dgamma( 1.0E-04 , 1.0E-04 ); # gamma prior for precision
#
#       Bayesian versions of LogLikelihood
         L1<-sum( loglike1[] )
         L0<-sum( loglike0[] )
#
#       Bayesian versions of BIC
         BIC1<- -2*L1 + 3*log(n)
         BIC0<- -2*L0 + 2*log(n)
#
#       Bayesian versions of AIC
         AIC1<- -2*L1 + 3*2
         AIC0<- -2*L0 + 2*2
#
#       Lm criterion
         Lm1<- sum( ss1[] )
         Lm0<- sum( ss0[] )
#
#       Mm criterion
        Mm1<-exp( sum(loglike1.pred[]) )
        Mm0<-exp( sum(loglike0.pred[]) )
#
        Mm1.star<-exp( -sum(loglike1.pred[])/n )
        Mm0.star<-exp( -sum(loglike0.pred[])/n )
#
# parallel differences
         DBIC10<- BIC0-BIC1
         DAIC10<- AIC0-AIC1
         diff[1]<-DAIC10
         diff[2]<-DBIC10
         diff[3]<- Lm0-Lm1
         diff[4]<-Mm1-Mm0
```

```
            diff[5]<-Mm0.star-Mm1.star
            PBF<-Mm1/Mm0
            PBFn<-Mm0.star/Mm1.star
#
#           parallel probabilities
            for (i in 1:5){
                        prob[i]<-step(diff[i])
            }
}
```

3 ΠΑΡΑΔΕΙΓΜΑ WINBUGS 3: Διαγνωστικά Τέστ και Συγκρίσεις μοντέλων (LINE DATASET)

```
model{
            pi<-3.14
#
#           Likelihood
            for( i in 1 : N ) {
                        y[i] ~ dnorm(mu[i],tau)
                        mu[i] <- alpha + beta * (x[i] - mean(x[]))
#
#                       residuals
                          resid[i]<-y[i]-mu[i]
                          sresid[i]<-resid[i]*sqrt(tau)
#
#                       predicted values
                         y.pred[i]~dnorm(mu[i],tau)
#
#             predicted standardised residuals
                        sr.pred[i]<-(y[i]-y.pred[i])*sqrt(tau)
#
#           p.smaller
            p.smaller[i]<-step(y[i]-y.pred[i])
#
                        sresid.pred[i]<-(y.pred[i]-mu[i])*sqrt(tau)
                        sresid3[i]<-pow( sresid[i] , 3 )
                        sresid3.pred[i]<-pow( sresid.pred[i] , 3 )
                        }
#
#           Prior distributions
            tau ~ dgamma(0.001,0.001)
            sigma <- 1 / sqrt(tau)
            alpha ~ dnorm(0.0,1.0E-6)
            beta ~ dnorm(0.0,1.0E-6)
#
#
            skew.obs<-mean(sresid3[])
            skew.pred<-mean(sresid3.pred[])
            pval.pred<-step(skew.pred-skew.obs)
  }
```

Data(WITHOUT OUTLIER): list(x = c(1, 2, 3, 4, 5), y= c(1, 3, 3, 3, 5), N = 5)

Data(WITH OUTLIER): list(x = c(1, 2, 3, 4, 5), y= c(1, 10000, 3, 3, 5), N = 5)

Inits: list(alpha = 0, beta = 0, tau = 1)

4 ΠΑΡΑΔΕΙΓΜΑ WINBUGS 4: Πλήρες Μοντέλο για το Antitoxin dataset

```
model {
#
#  model likelihood
for (i in 1:4) {
    r[i]~dbin(p[i],n[i]);
    logit(p[i])<-b[1] +  x[i,2]* b[2]
                      +  x[i,3]* b[3]
                      +  x[i,4]* b[4];  }
# priors and pseudopriors

b[1]~dnorm(0.0, 0.0001)
for (i in 2:4) {  b[i]~dnorm( 0.0 , 0.0001) ; }

}
```

DATA

```
r[] n[] x[,1] x[,2] x[,3] x[,4]
 5 12 1 -1  -1  1
 4 26 1  1  -1 -1
15 20 1 -1   1 -1
 6 21 1  1   1  1
END
```

INITS

```
list(  b=c(1,0,0,0) )
```

5 ΠΑΡΑΔΕΙΓΜΑ WINBUGS 5: Gibbs Variable Selection για το Antitoxin dataset

```
model {
#
#  model likelihood
for (i in 1:4) {
    r[i]~dbin(p[i],n[i]);
    logit(p[i])<-b[1] +  x[i,2]* g[2]* b[2]
                      +  x[i,3]* g[3]* b[3]
                      +  x[i,4]* g[4]* b[4];  }
# priors and pseudopriors
b[1]~dnorm( 0.0, 0.0001 )
for (i in 2:4) {
       tau[i]<-g[i]/8+(1-g[i])/(se[i]*se[i]);
       bpriorm[i]<-mean[i]*(1-g[i]);
  b[i]~dnorm(bpriorm[i],tau[i]); }

       mdl<-g[2]+2*g[3]+3*g[4];
   pmdl[1]<-equals(mdl,0)
   pmdl[2]<-equals(mdl,1)
   pmdl[3]<-equals(mdl,2)
   pmdl[4]<-equals(mdl,3)
   pmdl[5]<-equals(mdl,6)

for (i in 1:4) { g[i]~dbern( pi[i] ) }
pi[1]<-1.0
pi[2]<-0.5*(1-g[4])+g[4]
pi[3]<-0.5*(1-g[4])+g[4]
pi[4]<-0.20
}
```

DATA

```
r[] n[] x[,1] x[,2] x[,3] x[,4]
 5 12 1 -1  -1  1
 4 26 1  1  -1 -1
15 20 1 -1   1 -1
 6 21 1  1   1  1
END
```

PROPOSAL / PSEUDOPRIOR VALUES

```
 mean[]  se[]
-0.4889        0.2823
-0.8919        0.2798
 0.5866        0.2824
-0.1773       0.272
END
```

INITS

```
list( g=c(1,1,1,1),  b=c(1,0,0,0))
```

<u>ΠΑΡΑΡΤΗΜΑ Δ2: ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΔΗΜΟΣΙΕΥΣΕΙΣ ΣΧΕΤΙΚΕΣ ΜΕ BAYESIAN MODEL AND VARIABLE SELECTION</u>

1. Βιβλιογραφία
2. Ntzoufras, I. (2002). Gibbs Variable Selection Using BUGS. *Journal of Statistical Software*, Volume **7**, Issue 7, 1 – 19 .
3. Dellaportas, P., Forster, J.J. and Ntzoufras, I. (2000). *Bayesian Variable Selection Using the Gibbs Sampler*. *Generalized Linear Models: A Bayesian Perspective* (D.K.Dey, S.Ghosh and B. Mallick, eds.). New York: Marcel Dekker, 271 – 286.
4. Katsis, A. and Ntzoufras, I. (2003). Testing Hypotheses for the Distribution of Insurance Claim Counts Using the Gibbs Sampler. Technical Report.
5. Spiegelhalter, D.J., Best, N.G., Carlin, B.P. and van der Linde, A. (2002). Bayesian Measures of Model Complexity and Fit (with discussion). *Journal of the Royal Statistical Society*, **64**, 583 – 639.