

## Line: Linear Regression

```

model
{
  for(i in 1 : N) {
    Y[i] ~ dnorm(mu[i],tau)
    mu[i] <- alpha + beta * (x[i] - xbar)
  }
  tau ~ dgamma(0.001,0.001) sigma <- 1 / sqrt(tau)
  alpha ~ dnorm(0.0,1.0E-6)
  beta ~ dnorm(0.0,1.0E-6)
  #
  # Model diagnostics
  #
  # Residuals
  for (i in 1:N) {
    resid[i]<-Y[i]-mu[i]          # 1. residuals
    sresid[i]<-resid[i]*sqrt(tau)  # 2. standardised residuals
    Y.rep[i]~dnorm(mu[i],tau)     # generate predictive distributions
    p.smaller[i]<-step(Y[i]-Y.rep[i]) # 3. probability of more extreme obs
    like[i]<-sqrt(tau/(2*3.14))*exp(-0.5*pow(sresid[i],2)) # likelihood
    log.like[i]<-log(like[i])     # log-likelihood
    p.inv[i]<-1/like[i]           # 5.. predicive ordinate
  }
  deviance<- -2*sum(log.like[]) #deviance
  # p.values
  #
  for (i in 1:N){
    m3[i]<-pow(sresid[i],3)
    m4[i]<-pow(sresid[i],4)
    m3.rep[i]<-pow( (Y.rep[i]-mu[i])*sqrt(tau), 3)
    m4.rep[i]<-pow( (Y.rep[i]-mu[i])*sqrt(tau), 4)
  }
  skew.obs<-sum(m3[])/N
  skew.rep<-sum(m3.rep[])/N
  p.skew<-step(skew.rep-skew.obs)

  kur.obs<-sum(m4[])/N
  kur.rep<-sum(m4.rep[])/N
  p.kur<-step(kur.rep-kur.obs)
}

```

## Data

list(x = c(1, 2, 3, 4, 5), Y= c(1, 3, 3, 3, 5), xbar = 3, N = 5)

## Data2

list(x = c(1, 2, 3, 4, 5), Y= c(1, 7, 3, 3, 5), xbar = 3, N = 5)

## Inits

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

## Results

Time for 10000 updates 2s on 120MHz Pentium. A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	2.995	0.595	0.006334	1.989	2.994	3.97	1001	10000
beta	0.7928	0.3951	0.003815	0.08152	0.7938	1.504	1001	10000
sigma	0.9998	0.7714	0.01375	0.4148	0.8221	2.598	1001	10000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
deviance	12.92	3.723	0.07441	8.791	11.93	22.46	1001	10000
p.inv[1]	5.323	17.41	0.2042	1.269	3.039	21.16	1001	10000
p.inv[2]	6.825	22.21	0.2369	1.758	4.096	25.18	1001	10000
p.inv[3]	2.85	2.17	0.04817	1.111	2.266	8.228	1001	10000
p.inv[4]	6.888	19.8	0.1836	1.768	4.128	25.79	1001	10000
p.inv[5]	5.12	13.8	0.1621	1.278	3.042	19.95	1001	10000
p.kur	0.7326	0.4426	0.004581	0.0	1.0	1.0	1001	10000
p.skew	0.4985	0.5	0.004606	0.0	0.0	1.0	1001	10000
p.smaller[1]	0.3562	0.4789	0.004537	0.0	0.0	1.0	1001	10000
p.smaller[2]	0.7994	0.4004	0.004151	0.0	1.0	1.0	1001	10000
p.smaller[3]	0.5021	0.5	0.005143	0.0	1.0	1.0	1001	10000
p.smaller[4]	0.2019	0.4014	0.00353	0.0	0.0	1.0	1001	10000
p.smaller[5]	0.6586	0.4742	0.004697	0.0	1.0	1.0	1001	10000
resid[1]	-0.4012	0.9966	0.00897	-2.217	-0.4036	1.435	1001	10000
resid[2]	0.7974	0.6988	0.00612	-0.4993	0.7955	2.109	1001	10000
resid[3]	-0.003966	0.5609	0.005174	-1.063	-0.006994	1.076	1001	10000
resid[4]	-0.8054	0.6871	0.006957	-2.088	-0.8044	0.4597	1001	10000
resid[5]	0.3932	0.9803	0.01012	-1.423	0.405	2.154	1001	10000
sresid[1]	-0.5083	0.8021	0.00832	-2.065	-0.507	1.075	1001	10000
sresid[2]	1.002	0.6927	0.009614	-0.3197	0.9895	2.39	1001	10000
sresid[3]	-0.004754	0.4444	0.004815	-0.8759	-0.00905	0.8758	1001	10000
sresid[4]	-1.011	0.6907	0.008193	-2.405	-1.003	0.2874	1001	10000
sresid[5]	0.4988	0.807	0.008642	-1.08	0.5031	2.07	1001	10000

