Day 5: Bayesian Modelling in R – part 2

Ioannis Ntzoufras

Associate Professor in Statistics

E-mail: ntzoufras@aueb.gr

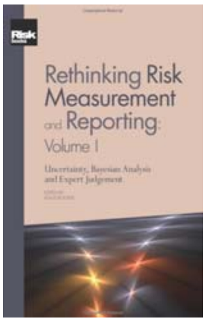
Contents

- **5… The normal linear model**
  - **The conjugate case**
  - **The Gibbs Sampler**
  - **Using the arm package**
  - **Using the MCMCpack**
- **6… A logistic regression example**
- **7… A Poisson regression example**
- **8… Other models that are ready to be used.**

Reference

Part of this presentation is based on the following book chapter

Ntzoufras, I. (2010). Bayesian Analysis of the Normal Regression Model. In *Rethinking Risk Measurement and Reporting: Uncertainty, Bayesian Analysis and Expert Judgement* (K. Böcker, ed.). Risk Books, pp. 69-106.



5…Bayesian Analysis of the Normal Regression Model

Let us consider the multivariate representation of a regression model

$$\boldsymbol{y}|\boldsymbol{\beta},\sigma^2,\boldsymbol{X} \sim N_n(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I}_n)$$

Then the MLEs are given by

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \ \text{ and } \ \widehat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{\beta}})^T(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{\beta}})$$

5…Bayesian Analysis of the Normal Regression Model

Different Priors

1. Conjugate Normal-Inverse Gamma prior
2. Zellner's g-prior (can be considered special case of 1)
3. Jeffrey's improper prior
4. Conditional conjugate prior [Gibbs]
5. Non-conjugate prior e.g. Double exponential leading to Lasso [Metropolis Hastings or other MCMC]

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis

Normal-Inverse Gamma prior

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{X} \sim N_P(\boldsymbol{\mu}_\beta, \boldsymbol{V}\sigma^2) \text{ and } \sigma^2|\boldsymbol{X} \sim \text{IG}(a, b), \quad a, b > 0;$$

Posterior

$$\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X} \sim \text{MSt}_P\left(\widetilde{\boldsymbol{\beta}}, \frac{\text{SS} + 2b}{n + 2a}\widetilde{\boldsymbol{\Sigma}}, n + 2a\right)$$

$$\sigma^2|\boldsymbol{y}, \boldsymbol{X} \sim IG(\widetilde{a}, \widetilde{b})$$

$$\beta_j|\boldsymbol{y}, \boldsymbol{X} \sim \text{MSt}_1\left(\widetilde{\beta}_j, \frac{\text{SS} + 2b}{n + 2a}\widetilde{\Sigma}_{jj}, n + 2a\right)$$

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis

Posterior

$$\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X} \sim \text{MSt}_P\Big(\widetilde{\boldsymbol{\beta}},\ \frac{\text{SS}+2b}{n+2a}\widetilde{\boldsymbol{\Sigma}},\ n+2a\Big) \qquad \sigma^2|\boldsymbol{y},\boldsymbol{X} \sim IG\big(\widetilde{a},\ \widetilde{b}\big)$$

$$\beta_j|\boldsymbol{y},\boldsymbol{X} \sim \text{MSt}_1\Big(\widetilde{\beta}_j,\ \frac{\text{SS}+2b}{n+2a}\widetilde{\Sigma}_{jj},\ n+2a\Big)$$

$$\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\Sigma}}\left(\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{V}^{-1}\boldsymbol{\mu}_\beta\right), \quad \widetilde{\boldsymbol{\Sigma}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{V}^{-1}\right)^{-1}$$

$$\widetilde{a} = \frac{n}{2} + a \ \text{ and } \ \widetilde{b} = \frac{\text{SS}}{2} + b$$

$$\text{with } \ \text{SS} = \boldsymbol{y}^T\boldsymbol{y} - \widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\Sigma}}^{-1}\widetilde{\boldsymbol{\beta}} + \boldsymbol{\mu}_\beta^T\boldsymbol{V}^{-1}\boldsymbol{\mu}_\beta.$$

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis - Posterior Summaries

Model Parameter	Posterior Summaries: Mean	Variance	q-th Quantile
β_j	$\widetilde{\beta}_j$	$\frac{\widetilde{b}}{\widetilde{a}-1}\widetilde{\Sigma}_{jj}$ *	$\widetilde{\beta}_j + t_{2\widetilde{a},q}\sqrt{\frac{\widetilde{b}}{\widetilde{a}}\widetilde{\Sigma}_{jj}}$ **
σ^2	$\frac{\widetilde{b}}{\widetilde{a}-1}$ †	$\left(\frac{\widetilde{b}}{\widetilde{a}-1}\right)^2\frac{1}{\widetilde{a}-2}$ ‡	$1/\Gamma_{\widetilde{a},\widetilde{b};1-q}$
σ	$\widetilde{b}^{1/2}\frac{\Gamma(\widetilde{a}-1/2)}{\Gamma(\widetilde{a})}$ ††	$\frac{\widetilde{b}}{\widetilde{a}-1} - \widetilde{b}\left(\frac{\Gamma(\widetilde{a}-1/2)}{\Gamma(\widetilde{a})}\right)^2$ †	$1/\sqrt{\Gamma_{\widetilde{a},\widetilde{b};1-q}}$

$\widetilde{\beta}$ and $\widetilde{\Sigma}$ are given by (1.15); $\widetilde{a}$ and $\widetilde{b}$ are given by (1.16)

* $\widetilde{\Sigma}_{jk}$ is the jth row and kth column element of matrix $\widetilde{\boldsymbol{\Sigma}}$ given by (1.15)

** $t_{\nu;q}$: q quantile of the Student t distribution with ν degrees of freedom

† for $\widetilde{a} > 1 \Leftrightarrow n > 2-2a$; ‡ for $\widetilde{a} > 2 \Leftrightarrow n > 4-2a$; †† for $\widetilde{a} > 1/2 \Leftrightarrow n > 1-2a$

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis – non informative priors

Low information $\Rightarrow$ V$\rightarrow$large, a=b $\rightarrow$ small

Then V$\rightarrow$0 and

$$\widetilde{\mathbf{\Sigma}} = (\mathbf{X}^T\mathbf{X} + \mathbf{V}^{-1})^{-1} \rightarrow (\mathbf{X}^T\mathbf{X})^{-1}$$

$$\widetilde{\beta} = \widetilde{\mathbf{\Sigma}}(\mathbf{X}^T\mathbf{y} + \mathbf{V}^{-1}\boldsymbol{\mu}_\beta\ ) \rightarrow (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \widehat{\beta}$$

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis – non informative priors

Low information $\Rightarrow$ V$\rightarrow$large, a=b $\rightarrow$ small

Then V$\rightarrow$0 and

$$SS = \mathbf{y}^T\mathbf{y} - \widetilde{\beta}^T\widetilde{\mathbf{\Sigma}}^{-1}\widetilde{\beta} + \mu_\beta^T\mathbf{V}^{-1}\mu_\beta$$
$$\rightarrow \mathbf{y}^T\mathbf{y} - \widehat{\beta}^T(\mathbf{X}^T\mathbf{X})^{-1}\widehat{\beta} = RSS$$

$$E(\sigma^2|\mathbf{y}) = \frac{\widetilde{b}}{\widetilde{a}-1} = \frac{SS/2+b}{n/2+a-1}$$
$$\rightarrow \frac{RSS}{n-2} \approx \sigma^2_{MLE} \text{ for reasonably large n}$$

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis in R

Example – Simple simulated example taken from Dellaportas et al. (2002, *Stats & Comp*)

Sample size n=50,

p=15 covariates from N(0,1)

$$Y_i \sim N(X_{i4} + X_{i5}, (2.5)^2) \text{ for } i = 1, 2, \ldots, 50$$

Dataset EX1.DAT

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis in R

Example – Simple simulated example taken from Dellaportas et al. (2002, *Stats & Comp*)

```
> simex1<-read.table('ex1.dat')
> names(simex1) <- c('y', paste('x',1:15,sep=''))
> simex1[1:5,]
        y      x1      x2      x3      x4      x5      x6      x7      x8      x9     x10     x11     x$
1 -2.4386  0.2532  1.8156 -0.4830 -0.3560 -1.8555  0.9481 -1.3358  0.1841 -0.4689 0.6804 -0.7008 -0.24$
2 -0.7768 -1.2131  0.7096 -0.0402  0.1953  0.1385  0.0058  1.2085  0.2564  1.4045 0.9116 -0.6775 -1.67$
3 -2.3082 -0.4847 -0.1016 -0.6435  0.1701 -0.3215 -0.2552 -2.5718  0.2579 -1.4463 0.1115  0.3261 -0.79$
4 -2.3640 -0.8514  1.5888  1.3791  0.1251 -0.8537 -1.3595 -0.1091 -0.5922 -0.9882 0.4722 -0.0096 -0.71$
5 -5.4150 -1.2730 -0.6017 -0.7611 -0.8256 -0.7499 -0.5043  0.9635 -1.0414 -0.4521 1.0405  1.1622  0.79$
      x14    x15
1  0.9144 0.1809
2 -0.5213 0.8702
3  0.5127 1.4399
4  0.0069 0.9861
5  0.2569 0.7769
> 
```

MLEs

```
> full.model.mle <- lm( y~., data=simex1 )
> summary(full.model.mle)

Call:
lm(formula = y ~ ., data = simex1)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.8620 -1.1113  0.3972  1.3740  4.4287 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.253588   0.440593   0.576  0.56870    
x1          -0.100721   0.404090  -0.249  0.80466    
x2          -0.309495   0.466928  -0.663  0.51191    
x3           0.005422   0.420335   0.013  0.98978    
x4           1.380789   0.395764   3.489  0.00136 ** 
x5           1.811081   0.334335   5.417 4.93e-06 ***
x6           0.463368   0.433084   1.070  0.29219    
x7           0.089440   0.415478   0.215  0.83084    
x8           0.483944   0.514689   0.940  0.35371    
x9          -0.560966   0.437109  -1.283  0.20805    
x10         -0.404871   0.516710  -0.784  0.43873    
x11         -0.415698   0.428122  -0.971  0.33842    
x12         -1.217795   0.573931  -2.122  0.04122 *  
x13         -0.251685   0.464335  -0.542  0.59133    
x14          0.143472   0.529403   0.271  0.78802    
x15         -0.621732   0.369859  -1.681  0.10193    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 2.529 on 34 degrees of freedom
Multiple R-squared: 0.5987,     Adjusted R-squared: 0.4216 
F-statistic: 3.381 on 15 and 34 DF,  p-value: 0.001589
```

$$Y_i \sim N(X_{i4} + X_{i5}, (2.5)^2) \text{ for } i = 1, 2, \ldots, 50$$

13

Conjugate Analysis in R (homemade code) 1/5

```
lm.bayes <- function( y, x, prior.mean=NULL, V=NULL,
    a=0.01, b=0.01, q=0.05, add.constant=TRUE,
    MLE=TRUE, digits=3 ){
    n<-length(y)
    if (add.constant) x<-cbind(1,x)
    p <- ncol(x)
    if (is.null(prior.mean)) prior.mean <- rep(0,p)
    if (is.null(V)) V <- 100*diag(p)
    mu <- prior.mean
    inv.V <- solve(V)
```

Conjugate Analysis in R (homemade code) 2/5

```
inv.tilde.Sigma <- t(x) %*% x + inv.V
tilde.Sigma <- solve( inv.tilde.Sigma )
M <- t(x) %*% y + inv.V %*% mu
tilde.beta <- as.vector(tilde.Sigma %*% M)
SS <- as.vector(
  t(y) %*% y
  -t(tilde.beta)%*%inv.tilde.Sigma%*%tilde.beta
  + t(mu)%*%inv.V%*%mu )
tilde.a <- 0.5*n + a
tilde.b <- 0.5*SS + b
post.params <- list( beta=tilde.beta,
                     S=tilde.Sigma, a=tilde.a,
                                    b=tilde.b )
```

Conjugate Analysis in R (homemade code) 2/5

$$\widetilde{\Sigma} = \left(X^T X + V^{-1}\right)^{-1}$$

```
inv.tilde.Sigma <- t(x) %*% x + inv.V
tilde.Sigma <- solve( inv.tilde.Sigma )
```

$$\widetilde{\beta} = \widetilde{\Sigma}\left(X^T y + V^{-1}\mu_\beta\right)$$

```
M <- t(x) %*% y + inv.V %*% mu
tilde.beta <- as.vector(tilde.Sigma %*% M)
SS <- as.vector(
  t(y) %*% y
  -t(tilde.beta)%*%inv.tilde.Sigma%*%tilde.beta
  + t(mu)%*%inv.V%*%mu )
tilde.a <- 0.5*n + a
tilde.b <- 0.5*SS + b
```

Conjugate Analysis in R (homemade code) 2/5

```
inv.tilde.Sigma <- t(x) %*% x + inv.V
tilde.Sigma <- solve( inv.tilde.Sigma )
```

$$\mathrm{SS} = \boldsymbol{y}^T\boldsymbol{y} - \widetilde{\boldsymbol{\beta}}^T \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{\beta}} + \boldsymbol{\mu}_\beta^T \boldsymbol{V}^{-1} \boldsymbol{\mu}_\beta$$ %*% M)

```
SS <- as.vector(
  t(y) %*% y
  -t(tilde.beta)%*%inv.tilde.Sigma%*%tilde.beta
  + t(mu)%*%inv.V%*%mu )
tilde.a <- 0.5*n + a
tilde.b <- 0.5*SS + b
post.params <- list( beta=tilde.beta, a=tilde.a,
                                      b=tilde.b )
```

Conjugate Analysis in R (homemade code) 2/5

```
inv.tilde.Sigma <- t(x) %*% x + inv.V
tilde.Sigma <- solve( inv.tilde.Sigma )
M <- t(x) %*% y + inv.V %*% mu
tilde.beta <- as.vector(tilde.Sigma %*% M)
SS <- as.vector(
```

$$\widetilde{a} \quad = \quad \frac{n}{2} + a \;\; \text{and} \;\; \widetilde{b} = \frac{\mathrm{SS}}{2} + b$$ %*%tilde.beta

```
tilde.a <- 0.5*n + a
tilde.b <- 0.5*SS + b
post.params <- list( beta=tilde.beta, a=tilde.a,
                                      b=tilde.b )
```

Conjugate Analysis in R (homemade code) 3/5

```
post.s2 <-list()
post.s2$mean <- tilde.b/(tilde.a-1)
post.s2$mode <- tilde.b/(tilde.a+1)
post.s2$sd   <- post.s2$mean * sqrt(1/(tilde.a-2))
```

Mean= $\frac{\tilde{b}}{\tilde{a}-1}$ † Var= $\left(\frac{\tilde{b}}{\tilde{a}-1}\right)^2 \frac{1}{\tilde{a}-2}$ ‡

Conjugate Analysis in R (homemade code) 4/5

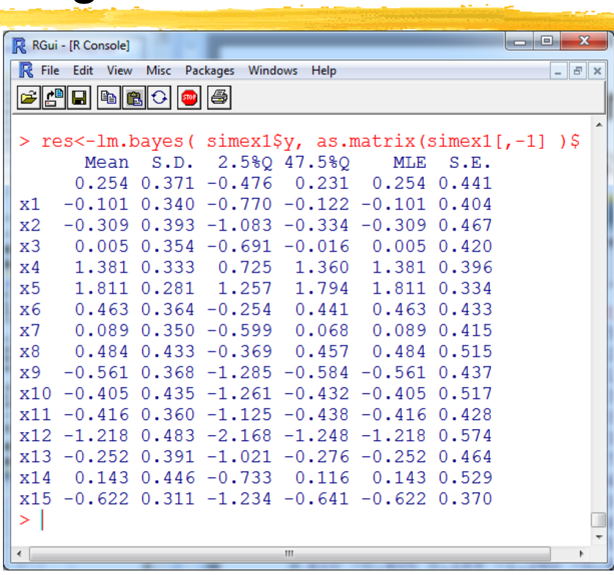
```
post.beta<-matrix( nrow=ncol(x), ncol=4 )
post.beta[,1]<- tilde.beta
tilde.Sigmajj <- diag(tilde.Sigma)
post.beta[,2]<- sqrt( post.s2$mean*tilde.Sigmajj )
se <- sqrt( (tilde.b/tilde.a)*tilde.Sigmajj )
post.beta[,3]<- tilde.beta+qt(q/2,2*tilde.a)*se
post.beta[,4]<- tilde.beta
        + qt((1-q)/2, 2*tilde.a )*se
```

Mean	Variance	q-th Quantile
$\tilde{\beta}_j$	$\frac{\tilde{b}}{\tilde{a}-1}\tilde{\Sigma}_{jj}$ *	$\tilde{\beta}_j + t_{2\tilde{a},q}\sqrt{\frac{\tilde{b}}{\tilde{a}}\tilde{\Sigma}_{jj}}$ **

Conjugate Analysis in R (homemade code) 5/5

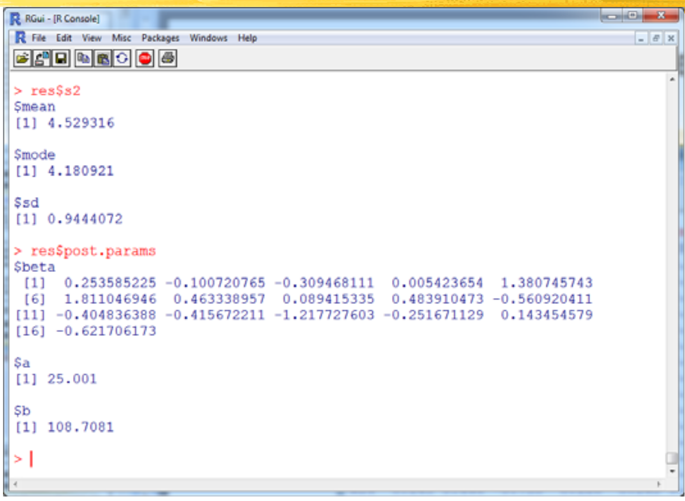
```
    rownames(post.beta) <- colnames(x)
    colnames(post.beta) <- c( 'Mean', 'S.D.',
    paste(round(100*q/2,1),'%Q',sep=''),
    paste(round(100*(1-q)/2,1),'%Q',sep='') )
    if (MLE){
      mles <- lm( y~x-1 )
      post.beta <- cbind(post.beta, as.matrix(
                  summary(full.model.mle)$coef[,1:2]))
      colnames(post.beta)[5:6]<-c('MLE', 'S.E.')
    }
    print (round(post.beta, digits))
    return( list( post.beta=post.beta,
    post.params=post.params) )
}
res<-lm.bayes( simex1$y, as.matrix(simex1[,-1] ) )
```

5…Bayesian Analysis of the Normal Regression Model




```
> res<-lm.bayes( simex1$y, as.matrix(simex1[,-1] ) )$
      Mean  S.D.   2.5%Q 47.5%Q    MLE  S.E.
     0.254 0.371 -0.476  0.231  0.254 0.441
x1  -0.101 0.340 -0.770 -0.122 -0.101 0.404
x2  -0.309 0.393 -1.083 -0.334 -0.309 0.467
x3   0.005 0.354 -0.691 -0.016  0.005 0.420
x4   1.381 0.333  0.725  1.360  1.381 0.396
x5   1.811 0.281  1.257  1.794  1.811 0.334
x6   0.463 0.364 -0.254  0.441  0.463 0.433
x7   0.089 0.350 -0.599  0.068  0.089 0.415
x8   0.484 0.433 -0.369  0.457  0.484 0.515
x9  -0.561 0.368 -1.285 -0.584 -0.561 0.437
x10 -0.405 0.435 -1.261 -0.432 -0.405 0.517
x11 -0.416 0.360 -1.125 -0.438 -0.416 0.428
x12 -1.218 0.483 -2.168 -1.248 -1.218 0.574
x13 -0.252 0.391 -1.021 -0.276 -0.252 0.464
x14  0.143 0.446 -0.733  0.116  0.143 0.529
x15 -0.622 0.311 -1.234 -0.641 -0.622 0.370
> |
```

5…Bayesian Analysis of the Normal Regression Model



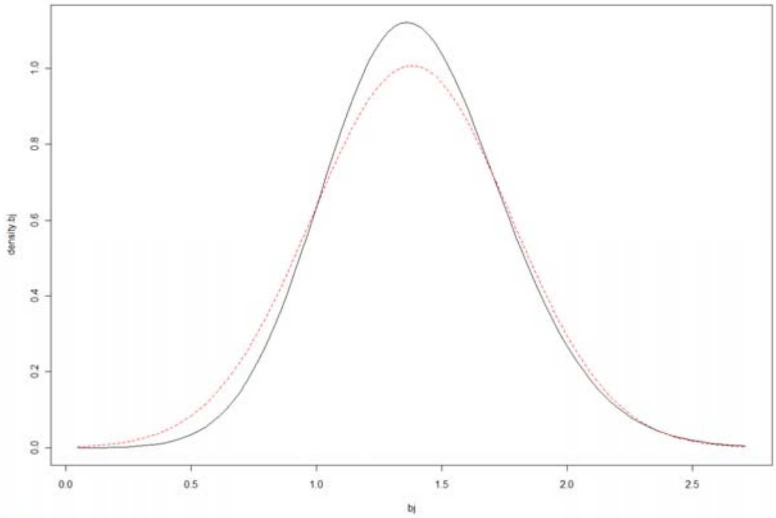
Conjugate Analysis in R (homemade code)

Posterior density plots

```
j<-5
npoints <- 100
b.min<- res$beta[j,1]-4*res$beta[j,2]
b.max<- res$beta[j,1]+4*res$beta[j,2]
bj <- seq( b.min, b.max, length.out=npoints )
a<-res$post.params$a
b<-res$post.params$b
beta<-res$post.params$beta[j]
se <- sqrt( (b/a)* diag(res$post.params$S)[j])
density.bj<- dt( bj/se, 2*a, ncp=beta/se)/se
plot( bj, density.bj, type='l')
mlej <- res$beta[j,5]
sej  <- res$beta[j,6]
lines( bj, dnorm(bj, mlej, sej), col=2, lty=2 )
```

Conjugate Analysis in R (homemade code)

Posterior density plots



Conjugate Analysis in R (homemade code)

Inverse Gamma density function

$$f(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right)$$

```
dinvgamma <-function( x, a, b, log=FALSE ){
    res <- a*log(b)-lgamma(a) -(a+1)*log(x) - b/x
    if (!log) res<-exp(res)
    return(res)
}
```

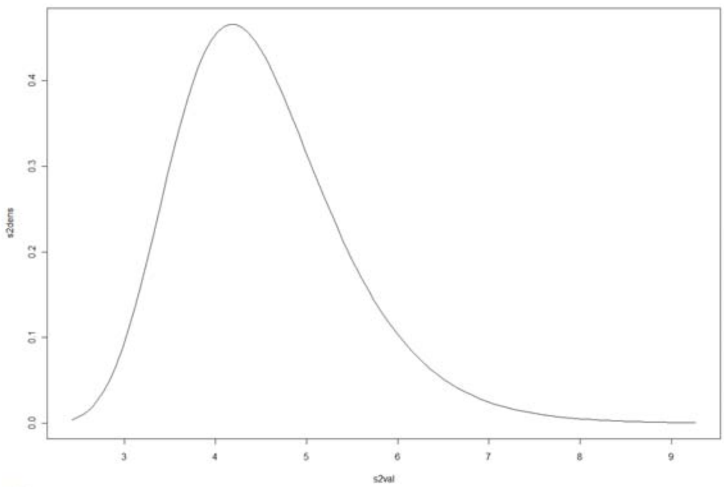
Conjugate Analysis in R (homemade code)

Density plot of σ^2

```
q <- 0.001
a<-res$post.params$a
b<-res$post.params$b
s2.min <- 1/qgamma( (1-q/2), a, b )
s2.max <- 1/qgamma( (q/2), a, b )
npoints<-100
s2val <- seq( s2.min, s2.max, length.out=npoints )
s2dens<- dinvgamma( s2val, a, b )
plot(s2val,s2dens,type='l')
```

Conjugate Analysis in R (homemade code)

Density plot of σ^2



28

Using arm package

The arm package uses a Student t prior distribution and EM to get posterior modes

The function bayesglm is very similar to glm

Here we set large prior variance for betas and the student degrees of freedom to be infinity in order to get a normal prior and comparable results with our previous analysis

The package uses EM to estimate posterior modes

For details see

Gelman, A., Jakulin, A., Pittau, M. G., & Su, Y. (2009). A Weakly Informative Default Prior DistributionFor Logistic And Other Regression Models. *The Annals of Applied Statistics*, 2(4), 1360-1383.

Using arm package

R CODE

```
library(arm)
bayesarm.model<-bayesglm(y~., data=simex1,
    prior.scale=1000, prior.df=Inf, n.iter=3000)
summary(bayesarm.model)
```

Using arm package

```
R Console
> summary(bayesarm.model1)

Call:
bayesglm(formula = y ~ ., data = simex1, prior.scale = 1000,
    prior.df = Inf)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.8620  -1.1113   0.3972   1.3740   4.4287

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.253588   0.363322   0.698   0.4884
x1          -0.100721   0.333222  -0.302   0.7637
x2          -0.309495   0.385038  -0.804   0.4253
x3           0.005422   0.346617   0.016   0.9876
x4           1.380789   0.326356   4.231 9.91e-05 ***
x5           1.811081   0.275700   6.569 2.84e-08 ***
x6           0.463368   0.357130   1.297   0.2004
x7           0.089440   0.342612   0.261   0.7951
x8           0.483944   0.424423   1.140   0.2596
x9          -0.560966   0.360449  -1.556   0.1259
x10         -0.404871   0.426090  -0.950   0.3466
x11         -0.415698   0.353038  -1.177   0.2446
x12         -1.217795   0.473276  -2.573   0.0131 *
x13         -0.251685   0.382900  -0.657   0.5140
x14          0.143472   0.436557   0.329   0.7438
x15         -0.621732   0.304994  -2.039   0.0468 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 4.348116)

    Null deviance: 541.72  on 49  degrees of freedom
Residual deviance: 217.41  on 50  degrees of freedom
AIC: 249.38

Number of Fisher Scoring iterations: 17

> |
```

31

Using arm package

R CODE – arm output

names(bayesarm.model)

```
RGui - [R Console]
File  Edit  View  Misc  Packages  Windows  Help

> names(bayesarm.model)
 [1] "coefficients"      "residuals"         "fitted.values"
 [4] "effects"           "R"                 "rank"
 [7] "qr"                "family"            "linear.predictors"
[10] "deviance"          "aic"               "null.deviance"
[13] "iter"              "weights"           "prior.weights"
[16] "df.residual"       "df.null"           "y"
[19] "converged"         "boundary"          "prior.mean"
[22] "prior.scale"       "prior.df"          "prior.sd"
[25] "dispersion"        "model"             "call"
[28] "formula"           "terms"             "data"
[31] "offset"            "control"           "method"
[34] "contrasts"         "xlevels"
> |
```

32

Using arm package

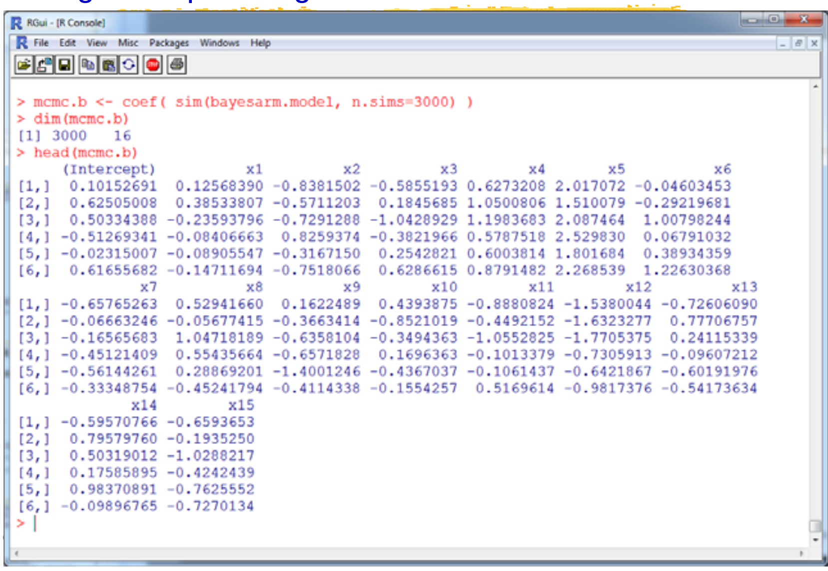
R CODE – Obtaining a posterior sample for beta

```
mle <- lm(y~., data=simex1)
mcmc.b <- coef( sim(mle, n.sims=3000) )
dim(mcmc.b)
head(mcmc.b)
```

IMPORTANT NOTES:

- It is not clear what prior is use to generate samples (we can assume is non-informative since the mle output is used)
- The posterior sample is not necessarily coming from an MCMC algorithm since here the posterior can be obtained analytically under conjugate NIG priors.

Using arm package

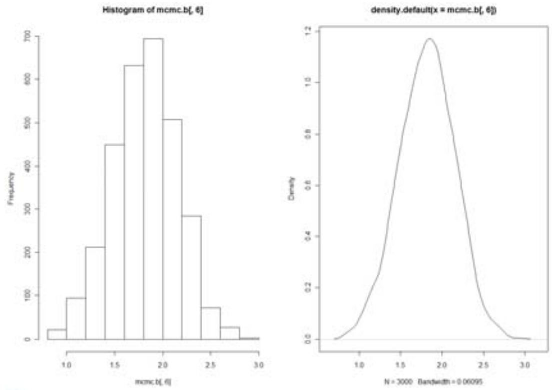


```
> mcmc.b <- coef( sim(bayesarm.model, n.sims=3000) )
> dim(mcmc.b)
[1] 3000   16
> head(mcmc.b)
      (Intercept)          x1         x2         x3        x4       x5          x6
[1,]  0.10152691  0.12568390 -0.8381502 -0.5855193 0.6273208 2.017072 -0.04603453
[2,]  0.62505008  0.38533807 -0.5711203  0.1845685 1.0500806 1.510079 -0.29219681
[3,]  0.50334388 -0.23593796 -0.7291288 -1.0428929 1.1983683 2.087464  1.00798244
[4,] -0.51269341 -0.08406663  0.8259374 -0.3821966 0.5787518 2.529830  0.06791032
[5,] -0.02315007 -0.08905547 -0.3167150  0.2542821 0.6003814 1.801684  0.38934359
[6,]  0.61655682 -0.14711694 -0.7518066  0.6286615 0.8791482 2.268539  1.22630368
              x7          x8         x9        x10        x11        x12         x13
[1,] -0.65765263  0.52941660  0.1622489  0.4393875 -0.8880824 -1.5380044 -0.72606090
[2,] -0.06663246 -0.05677415 -0.3663414 -0.8521019 -0.4492152 -1.6323277  0.77706757
[3,] -0.16565683  1.04718189 -0.6358104 -0.3494363 -1.0552825 -1.7705375  0.24115339
[4,] -0.45121409  0.55435664 -0.6571828  0.1696363 -0.1013379 -0.7305913 -0.09607212
[5,] -0.56144261  0.28869201 -1.4001246 -0.4367037 -0.1061437 -0.6421867 -0.60191976
[6,] -0.33348754 -0.45241794 -0.4114338 -0.1554257  0.5169614 -0.9817376 -0.54173634
             x14        x15
[1,] -0.59570766 -0.6593653
[2,]  0.79579760 -0.1935250
[3,]  0.50319012 -1.0288217
[4,]  0.17585895 -0.4242439
[5,]  0.98370891 -0.7625552
[6,] -0.09896765 -0.7270134
>
```

Using arm package

R CODE – Plotting the posterior for β_5

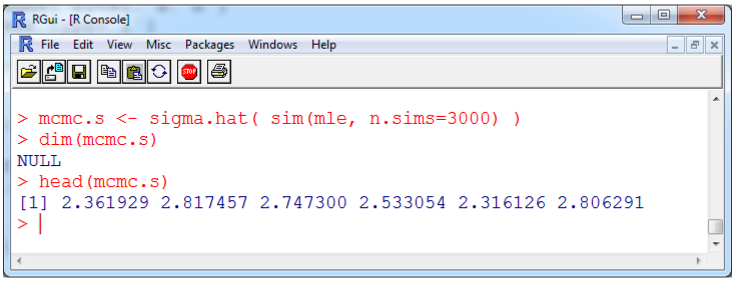
```
par(mfrow=c(1,2))
hist(mcmc.b[,6])
plot(density(mcmc.b[,6]))
```



Using arm package

R CODE – Obtaining a posterior sample for sigma

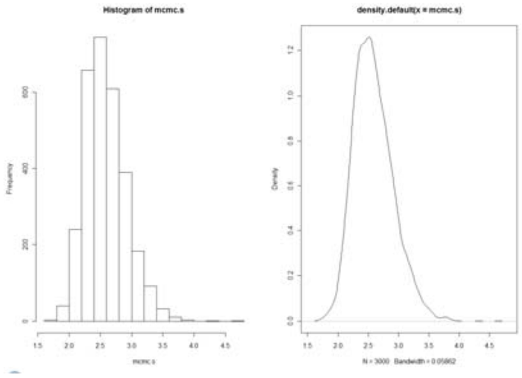
```
mcmc.s <- sigma.hat( sim(mle, n.sims=3000) )
dim(mcmc.s)
head(mcmc.s)
```



Using arm package

R CODE – Plotting the posterior for sigma

```
par(mfrow=c(1,2))
hist(mcmc.s[,6])
plot(density(mcmc.s[,6]))
```

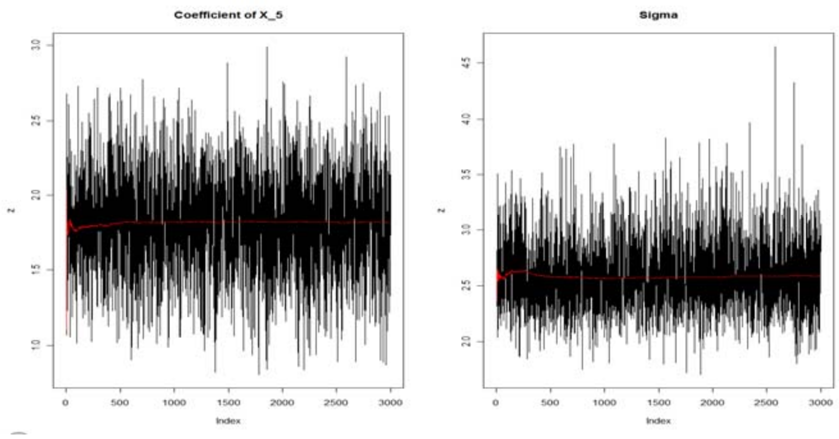


Using arm package

R CODE – Plotting traceplots & ergodic means

```
par(mfrow=c(1,2))
 z<-mcmc.b[,5]
 n<-length(z)
 plot(z, type='l', main='Coefficient of X_5')
 lines(1:n,cumsum(z)/1:n, col=2)
 z<-mcmc.s
 n<-length(z)
 plot(z, type='l', main='Sigma')
 lines(1:n,cumsum(z)/1:n, col=2)
```

Using arm package

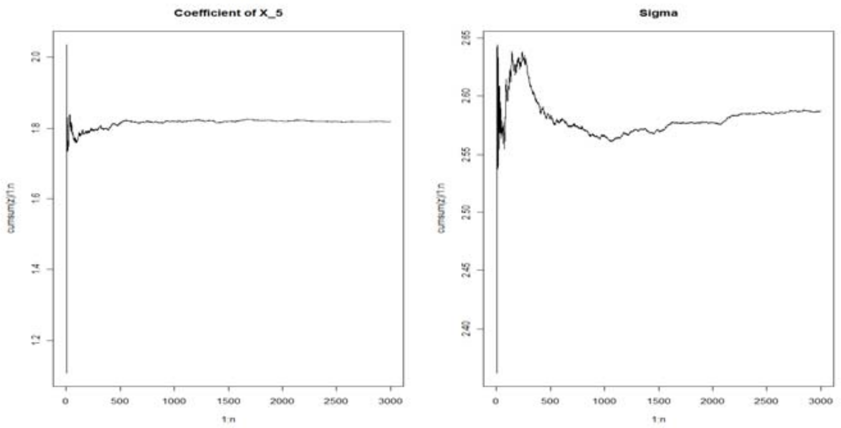


Using arm package

R CODE – Plotting ergodic means (only)

```
par(mfrow=c(1,2))
 z<-mcmc.b[,5]
 n<-length(z)
 plot(1:n,cumsum(z)/1:n, , type='l', main='Coefficient
    of X_5')
 z<-mcmc.s
 n<-length(z)
 plot(1:n,cumsum(z)/1:n, type='l', main='Sigma')
```

Using arm package



MCMCpack package

MCMCpack is very popular and implements several models including glms using MCMC methods

For the normal prior, a conditional conjugate prior is used

For details see

Martin A.D., Quinn K.M. and Park J.H. (2011). MCMCpack: Markov Chain Monte Carlo in R. *Journal of Statistical Software*, vol. 42, issue 9.

R CODE

```
library(MCMCpack)
mcmcpack.model<-MCMCregress(y~., data=simex1,
    b0=0, B0=1/1000, c0=0.001, d0=0.001)
head(mcmcpack.model)
dim(mcmcpack.model)
```

MCMCpack package

```
> head(mcmcpack.model)
Markov Chain Monte Carlo (MCMC) output:
Start = 1001
End = 1007
Thinning interval = 1
      (Intercept)          x1          x2         x3        x4       x5         x6          x7          x8
[1,]   1.17797262 -0.83063333  0.30111450 -0.3212841 0.1972991 1.402432  0.2774340  0.08462749 -0.26753444
[2,]   0.09921117 -1.35533788  0.18509470 -0.2312981 1.8189436 1.261815  0.4512720 -0.04014777  1.30907476
[3,]   1.00845648 -0.04111453  0.11744090 -0.1117612 1.1051260 1.808506  0.7069503  0.30207428 -0.52403009
[4,]   0.28477835  0.06713957 -1.02782216 -0.1059796 1.5199036 1.841074 -0.2593084  0.03107302 -0.00816577
[5,]   0.15036806 -0.40280450 -0.42337278  0.1350658 1.8101356 1.942237  0.7780077 -0.43388895 -0.11208966
[6,]  -0.24162149  0.22264503  0.09121447 -0.4844048 1.2403635 2.118110  0.1979703 -0.44852204  0.80473775
[7,]  -0.51127045 -0.17736447 -0.45392862 -0.9748375 2.0428324 2.101123  0.9892819 -0.79247276  0.92614944
               x9        x10         x11        x12        x13         x14        x15   sigma2
[1,]   0.03912859 -0.6173281  0.08608288 -1.5196355  0.1063027  1.34688583 -0.4572774 7.501590
[2,]  -0.83829712 -0.1094024 -0.76917244 -1.7232133 -0.1284611  0.01336550 -0.6539634 6.722550
[3,]  -0.38593253 -0.8943845  0.28534306 -0.3742197 -0.5547927  0.25822228 -0.7682214 7.443708
[4,]  -1.15054793 -0.3634023 -0.11802495 -1.8073122 -0.2528056  1.65528351 -0.2499832 9.292325
[5,]  -0.17194475 -0.3510138 -0.13794295 -0.7857808 -0.2726802 -0.57532917 -0.3017852 6.088567
[6,]  -1.13721914  0.3786469 -0.58733651 -1.4531209  0.2134558  0.54680696 -0.7019880 5.173546
[7,]  -0.70404940  0.2338510 -0.80354426 -1.4561861 -0.4946237  0.11941803 -0.6983144 5.658403
> dim(mcmcpack.model)
[1] 10000    17
>
```

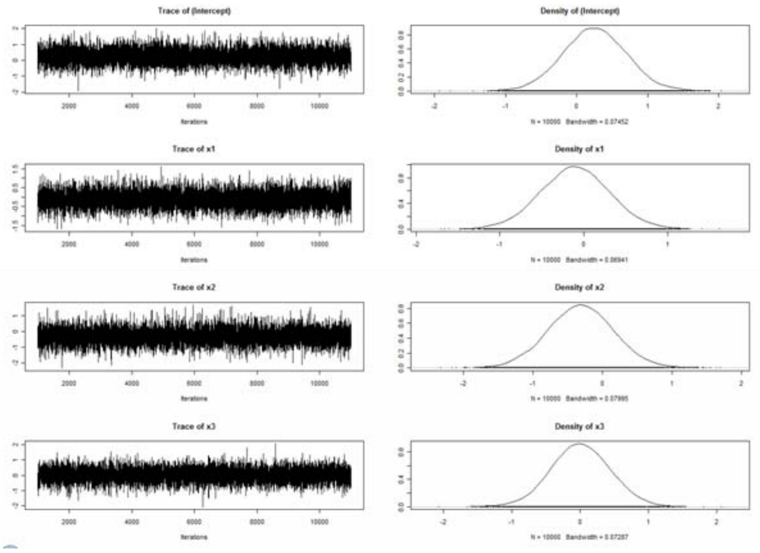

MCMCpack package

```
summary(mcmcpack.model)
plot(mcmcpack.model)
```

```
NULL
> mcmcpack.model<-MCMCregress(y~., data=simex1, b0=0, B0=1/1000, c0=0.01, d0=0.01)
> summary(mcmcpack.model)

Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean     SD Naive SE Time-series SE
(Intercept)  0.25425 0.4480 0.004480       0.004985
x1          -0.10464 0.4155 0.004155       0.003755
x2          -0.31146 0.4814 0.004814       0.004659
x3           0.00990 0.4352 0.004352       0.003723
x4           1.38097 0.4024 0.004024       0.003933
x5           1.80856 0.3446 0.003446       0.003650
x6           0.46255 0.4456 0.004456       0.004040
x7           0.08572 0.4296 0.004296       0.003882
x8           0.48522 0.5322 0.005322       0.004419
x9          -0.55650 0.4460 0.004460       0.005118
x10         -0.39950 0.5317 0.005317       0.004498
x11         -0.41396 0.4450 0.004450       0.004759
x12         -1.21294 0.5944 0.005944       0.005696
x13         -0.25312 0.4841 0.004841       0.004996
x14          0.13534 0.5482 0.005482       0.004837
x15         -0.62337 0.3786 0.003786       0.002964
sigma2       6.79714 1.7812 0.017812       0.028640

2. Quantiles for each variable:

               2.5%      25%       50%      75%   97.5%
(Intercept) -0.6168 -0.04233  0.255338 0.552025 1.14125
x1          -0.9257 -0.38293 -0.105703 0.170669 0.72314
x2          -1.2492 -0.63319 -0.311422 0.004537 0.63659
x3          -0.8465 -0.28044  0.008075 0.300776 0.86809
x4           0.5761  1.11776  1.386492 1.646132 2.16119
x5           1.1267  1.57873  1.813628 2.038423 2.48888
```

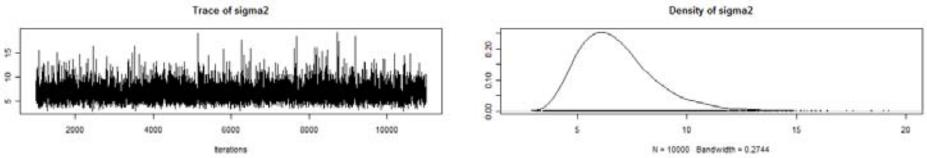
45

MCMCpack package



46

MCMCpack package

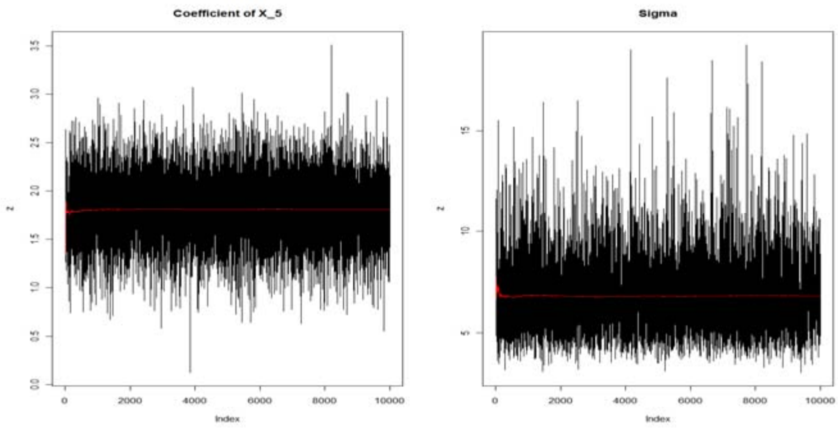


MCMCpack package

R CODE – Plotting traceplots & ergodic means

```
par(mfrow=c(1,2))
 z<-as.vector(mcmcpack.model[,6])
 n<-length(z)
 plot(z, type='l', main='Coefficient of X_5')
 lines(1:n,cumsum(z)/1:n, col=2)
 z<-as.vector(mcmcpack.model[,17])
 n<-length(z)
 plot(z, type='l', main='Sigma')
 lines(1:n,cumsum(z)/1:n, col=2)
```

MCMCpack package

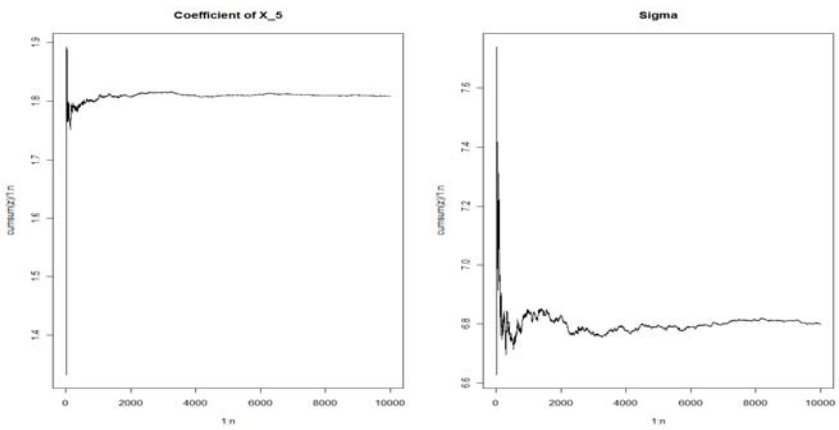


MCMCpack package

R CODE – Plotting ergodic means (only)

```
par(mfrow=c(1,2))
 z<-as.vector(mcmcpack.model[,6])
 n<-length(z)
 plot(1:n,cumsum(z)/1:n, , type='l', main='Coefficient
    of X_5')
 z<-as.vector(mcmcpack.model[,17])
 n<-length(z)
 plot(1:n,cumsum(z)/1:n, type='l', main='Sigma')
```

MCMCpack package



Bayesm package

Bayesm package generally has a variety of functions for generating samples from the posterior

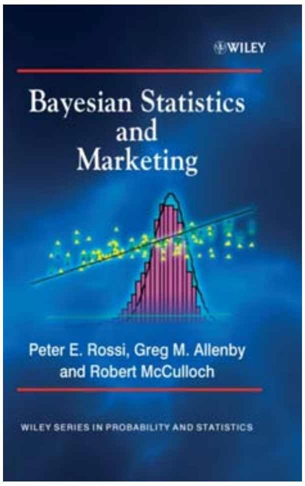
For the normal regression model, we can find

1. runireg which applies a simple IID sampler for regression (conjugate analysis)
2. runiregGibbs which applies a Gibbs Sampler for Regression (conditional conjugate analysis)

Hence posterior inference is made via posterior samples.

Bayesm package

Further details, data and examples can be found in
http://www.perossi.org/home/bsm-1



Bayesm package - runireg

Uses the conjugate prior discussed previously

`runireg` implements an iid sampler to draw from the posterior of a univariate regression with a conjugate prior.

Usage

```
runireg(Data, Prior, Mcmc)
```

Arguments

`Data`	list(y,X)
`Prior`	list(betabar,A, nu, ssq)
`Mcmc`	list(R,keep)

betabar = vector of prior means of beta

A = precision matrix (V^{-1} in our notation)

a=v/2, b=v ssq/2 => mean=1/ssq & var=2/(v*ssq^2)

=> v=2a, ssq=b/a

R=Total number of iterations, keep=Thinning

Bayesm package - runireg

The Code

```
library(bayesm)
simex1.list <- list(y=simex1[,1],
                    X=as.matrix(cbind(1,simex1[,-1])) )
p<-ncol(simex1.list$X)
prior1 <- list( betabar=rep(0,p), A=0.01*diag(p),
                nu=0.02, ssq=1)
mcmc1  <- list( R=1000, keep=1 )
bayesm.model <- runireg(Data=simex1.list, Prior=prior1,
    Mcmc=mcmc1)
```

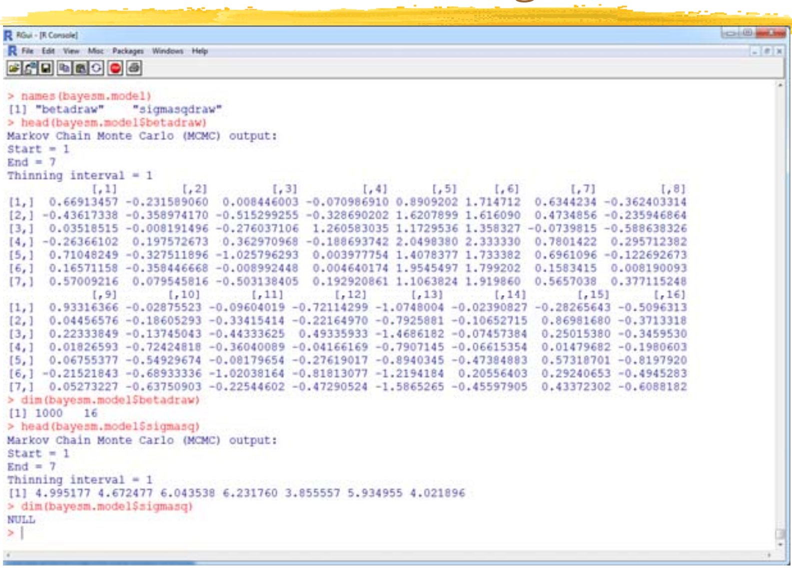
Bayesm package - runireg

```
Starting IID Sampler for Univariate Regression Model
  with  50  observations

Prior Parms:
betabar
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
A
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
 [1,]    1    0    0    0    0    0    0    0    0     0     0     0     0     0     0     0
 [2,]    0    1    0    0    0    0    0    0    0     0     0     0     0     0     0     0
 [3,]    0    0    1    0    0    0    0    0    0     0     0     0     0     0     0     0
 [4,]    0    0    0    1    0    0    0    0    0     0     0     0     0     0     0     0
 [5,]    0    0    0    0    1    0    0    0    0     0     0     0     0     0     0     0
 [6,]    0    0    0    0    0    1    0    0    0     0     0     0     0     0     0     0
 [7,]    0    0    0    0    0    0    1    0    0     0     0     0     0     0     0     0
 [8,]    0    0    0    0    0    0    0    1    0     0     0     0     0     0     0     0
 [9,]    0    0    0    0    0    0    0    0    1     0     0     0     0     0     0     0
[10,]    0    0    0    0    0    0    0    0    0     1     0     0     0     0     0     0
[11,]    0    0    0    0    0    0    0    0    0     0     1     0     0     0     0     0
[12,]    0    0    0    0    0    0    0    0    0     0     0     1     0     0     0     0
[13,]    0    0    0    0    0    0    0    0    0     0     0     0     1     0     0     0
[14,]    0    0    0    0    0    0    0    0    0     0     0     0     0     1     0     0
[15,]    0    0    0    0    0    0    0    0    0     0     0     0     0     0     1     0
[16,]    0    0    0    0    0    0    0    0    0     0     0     0     0     0     0     1
nu =  0.02  ssq=  1

MCMC parms:
R=  1000  keep=  1

IID Iteration (est time to end - min)
  100   ( 0 )
  200   ( 0 )
  300   ( 0 )
  400   ( 0 )
  500   ( 0 )
  600   ( 0 )
  700   ( 0 )
  800   ( 0 )
  900   ( 0 )
  1000  ( 0 )
  Total Time Elapsed:  0.01
```

Bayesm package - runireg

```
names(bayesm.model)
head(bayesm.model$betadraw)
dim(bayesm.model$betadraw)
head(bayesm.model$sigmasq)
dim(bayesm.model$sigmasq)
```

Bayesm package - runireg



Bayesm package - runireg

```
# with some default values
bayesm.model <- runireg(Data=simex1.list,
    Mcmc=list(R=1000))
#
# default values
Prior mean for beta = 0
Prior precision for beta = 0.01 (independent prior)
nu=3?
Ssq=11.01?
Keep=1
```

Bayesm package - runireg

```
bayesm.model <- runireg(Data=simex1.list,
           Prior=prior1, Mcmc=mcmc1)
summary(bayesm.model$betadraw, burnin=100)
```

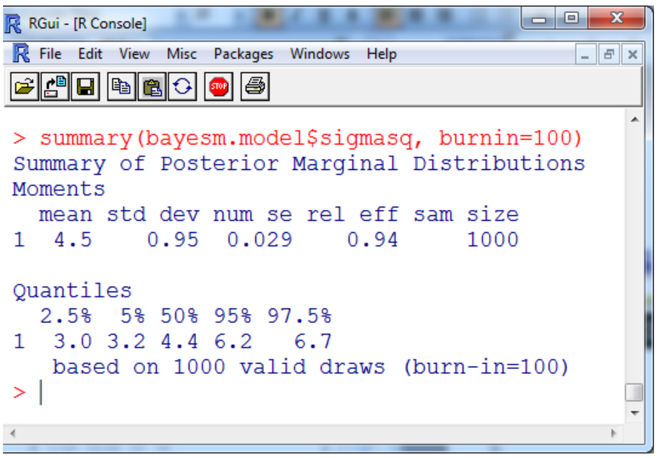
Bayesm package - runireg

```
  Total Time Elapsed:  0.01
> summary(bayesm.model$betadraw, burnin=100)
Summary of Posterior Marginal Distributions
Moments
     mean std dev num se rel eff sam size
1   0.247    0.36 0.0102    0.79     1000
2  -0.104    0.34 0.0109    1.04      500
3  -0.321    0.39 0.0136    1.25      500
4   0.006    0.36 0.0114    1.01      500
5   1.376    0.33 0.0109    1.08      500
6   1.792    0.28 0.0065    0.56     1000
7   0.447    0.35 0.0080    0.51     1000
8   0.098    0.34 0.0089    0.66     1000
9   0.478    0.44 0.0113    0.66     1000
10 -0.534    0.38 0.0109    0.83     1000
11 -0.391    0.42 0.0124    0.86     1000
12 -0.414    0.37 0.0105    0.82     1000
13 -1.210    0.48 0.0139    0.84     1000
14 -0.250    0.40 0.0137    1.20      500
15  0.139    0.44 0.0115    0.69     1000
16 -0.625    0.31 0.0087    0.82     1000

Quantiles
    2.5%    5%    50%    95%  97.5%
1  -0.42 -0.32  0.237  0.857  1.000
2  -0.76 -0.63 -0.106  0.455  0.549
3  -1.10 -0.92 -0.320  0.301  0.454
4  -0.67 -0.57  0.016  0.584  0.726
```

61

Bayesm package - runireg

summary(bayesm.model$sigmasq, burnin=100)



Bayesm package - runireg

R Code – Obtaining posterior summaries (manually)

```
z1<- apply(bayesm.model$betadraw, 2, mean)
z2<- apply(bayesm.model$betadraw, 2, sd)
z3<- c( mean(bayesm.model$sigmasqdraw),
    sd(bayesm.model$sigmasqdraw) )
z <- rbind( cbind(z1,z2), z3)
rownames(z) <- c('Const', paste('X', 1:15, sep=''),
    'sigmasq')
colnames(z) <- c('Mean', 'SD')
round(z,3)
```

Bayesm package - runireg

RGui - [R Console]

File Edit View Misc Packages Windows Help

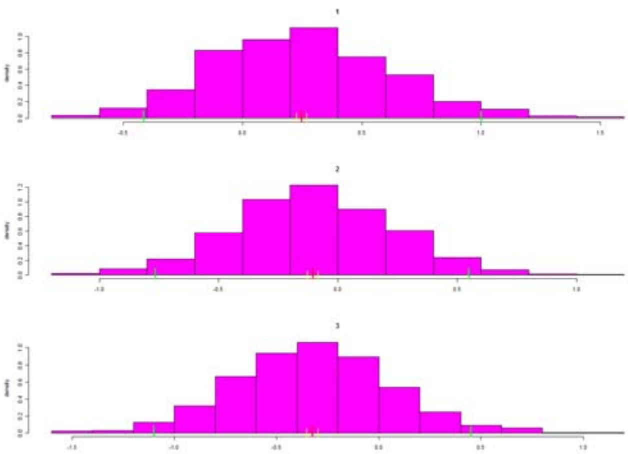
```
> colnames(z) <- c('Mean', 'SD')
> round(z,3)
          Mean    SD
Const    0.242 0.358
X1      -0.096 0.336
X2      -0.324 0.386
X3       0.002 0.354
X4       1.378 0.330
X5       1.797 0.278
X6       0.445 0.353
X7       0.091 0.347
X8       0.470 0.440
X9      -0.536 0.379
X10     -0.385 0.426
X11     -0.414 0.367
X12     -1.214 0.484
X13     -0.246 0.397
X14      0.141 0.441
X15     -0.625 0.309
sigmasq  4.528 0.957
>
```

Bayesm package - runireg

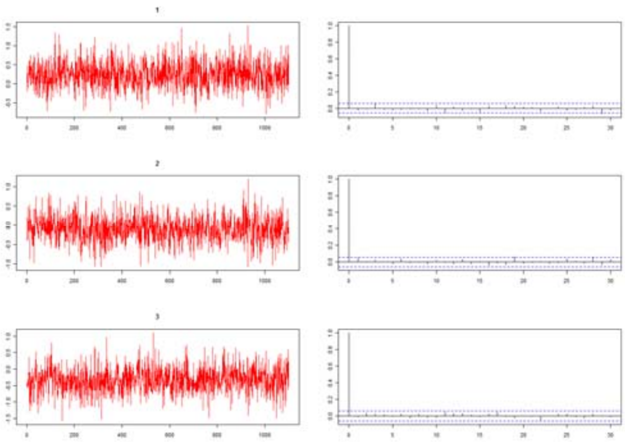
R Code – Obtaining posterior plots

```
plot.bayesm.mat(bayesm.model$betadraw[,1:3])
plot(bayesm.model$betadraw)
```

Bayesm package - runireg



Bayesm package - runireg



Bayesm package - runiregGibbs

Uses the conditional conjugate N-IG prior

Same as in the conjugate prior but β and σ^2 are now independent

```
runiregGibbs(Data, Prior, Mcmc)
```

Arguments

`Data`	list(y,X)
`Prior`	list(betabar,A, nu, ssq)
`Mcmc`	list(sigmasq,R,keep)

Bayesm package - runiregGibbs

The Code

```
library(bayesm)
simex1.list <- list(y=simex1[,1],
    X=as.matrix(cbind(1,simex1[,-1])) )
p<-ncol(simex1.list$X)
prior1 <- list( betabar=rep(0,p), A=0.01*diag(p),
    nu=0.02, ssq=1)
mcmc1  <- list( R=5100, keep=1 )
bayesm.model2 <- runiregGibbs(Data=simex1.list,
    Prior=prior1, Mcmc=mcmc1)
summary(bayesm.model$betadraw)
summary(bayesm.model$sigmasq)
plot(bayesm.model$betadraw)
plot.bayesm.mat(bayesm.model$betadraw[,1:3])
plot(bayesm.model$sigmasq)
```

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis – Zellner's g-prior

Normal-Inverse Gamma prior

$$\beta|\sigma^2, \boldsymbol{X} \sim N_P(\boldsymbol{\mu}_\beta, \boldsymbol{V}\sigma^2) \text{ and } \sigma^2|\boldsymbol{X} \sim \mathrm{IG}(a,b), \quad a,b>0;$$

Z g-prior => $\mu_\beta = \mathbf{0}$ and $\mathbf{V} = g\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$.

The prior is now given by

$$\beta|\sigma^2 \sim N\left(\mathbf{0}, g\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\sigma^2\right)$$

The prior for σ^2 is usually taken as

$$f(\sigma^2) \propto \sigma^{-2} \text{ (or IG as above)}$$

Usually, g=n to represent a Unit information prior (Kass & Wasserman, 1995)

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis

Posterior

$$\beta|\mathbf{y} \sim MSt_p\left(w\widehat{\beta}, w\frac{SS}{n}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right) \text{ with } w = \frac{g}{g+1}$$

$$\sigma^2|\mathbf{y} \sim IG\left(\frac{n}{2}, \frac{SS}{2}\right)$$

$$SS = RSS - \frac{1}{1+g}\beta^T\left(\mathbf{X}^T\mathbf{X}\right)\beta$$

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis – Zellner's g-prior

- Usually, g=n to represent a Unit information prior (Kass & Wasserman, 1995)
- Large values of g $\Rightarrow$ represent ignorance or low information
- w=g/(g+1) is the shrinkage parameter
- For low information priors (i.e. large g) => w$\Rightarrow$1 i.e. the posterior mode convergences to the mle (i.e. the information from the data).

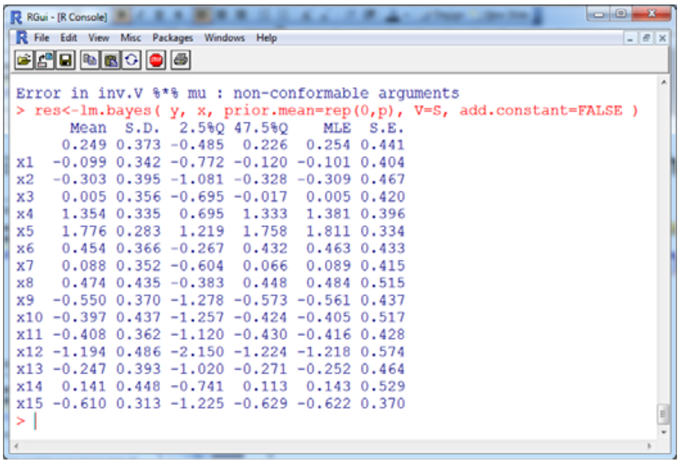
5…Bayesian Analysis of the Normal Regression Model

Zellner's g-prior with bayes.lm

```
y<-simex1$y
x<-cbind( 1, as.matrix(simex1[,-1] ) )
n<-length(y)
p<-ncol(x)

g<-n
S <- g*solve( t(x) %*% x )
res<-lm.bayes( y, x, prior.mean=rep(0,p), V=S,
    add.constant=FALSE )
```

5…Bayesian Analysis of the Normal Regression Model

Conjugate Analysis – Zellner's g-prior

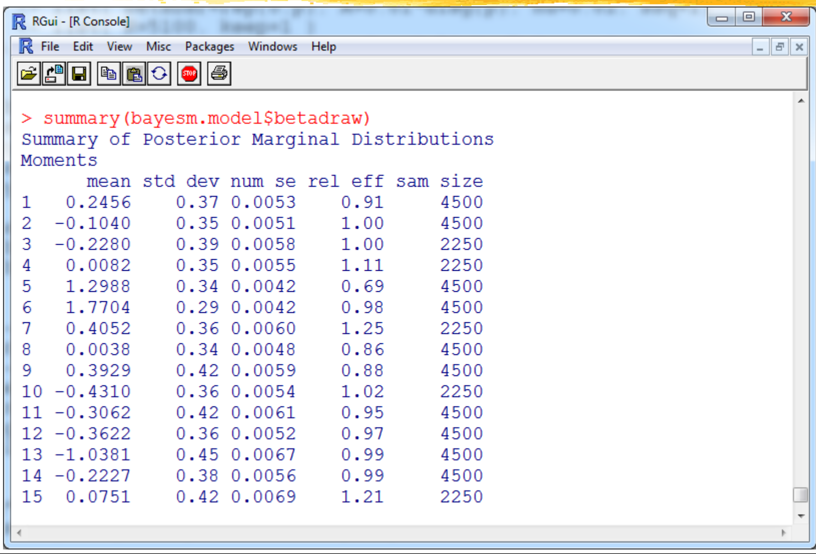


5…Bayesian Analysis of the Normal Regression Model

Zellner’s g-prior with Bayesm

```
library(bayesm)
y<-simex1$y
x<-cbind( 1, as.matrix(simex1[,-1] ) )
n<-length(y)
p<-ncol(x)
g<-n
S <- g*solve( t(x) %*% x )
simex1.list <- list(y=y, X=x )
prior1 <- list( betabar=rep(0,p), A=S, nu=0.02, ssq=1)
mcmc1  <- list( R=2000, keep=1 )
bayesm.model <- runireg(Data=simex1.list, Prior=prior1,
    Mcmc=mcmc1)
summary(bayesm.model$betadraw)
```

5…Bayesian Analysis of the Normal Regression Model




```
> summary(bayesm.model$betadraw)
Summary of Posterior Marginal Distributions
Moments
      mean std dev num se rel eff sam size
1   0.2456    0.37 0.0053    0.91     4500
2  -0.1040    0.35 0.0051    1.00     4500
3  -0.2280    0.39 0.0058    1.00     2250
4   0.0082    0.35 0.0055    1.11     2250
5   1.2988    0.34 0.0042    0.69     4500
6   1.7704    0.29 0.0042    0.98     4500
7   0.4052    0.36 0.0060    1.25     2250
8   0.0038    0.34 0.0048    0.86     4500
9   0.3929    0.42 0.0059    0.88     4500
10 -0.4310    0.36 0.0054    1.02     2250
11 -0.3062    0.42 0.0061    0.95     4500
12 -0.3622    0.36 0.0052    0.97     4500
13 -1.0381    0.45 0.0067    0.99     4500
14 -0.2227    0.38 0.0056    0.99     4500
15  0.0751    0.42 0.0069    1.21     2250
```

5…Bayesian Analysis of the Normal Regression Model

Zellner's g-prior with Bayesm

```
round(
cbind(res$beta, apply( bayesm.model$betadraw,2,mean )
                , apply( bayesm.model$betadraw,2,sd ))
                ,3)
```

5…Bayesian Analysis of the Normal Regression Model

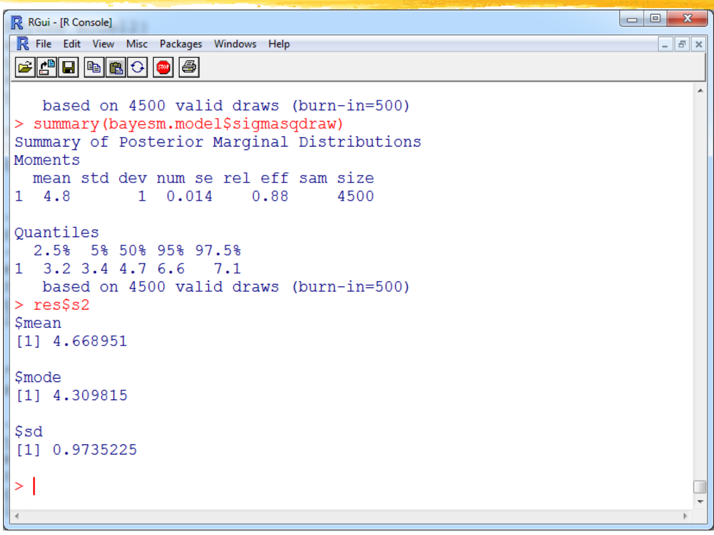
```
>
> round(cbind(res$beta, apply( bayesm.model$betadraw,2,mean )
+               , apply( bayesm.model$betadraw,2,sd )), 3)
      Mean  S.D.  2.5%Q 47.5%Q    MLE  S.E.
     0.249 0.373 -0.485  0.226  0.254 0.441  0.243 0.370
x1  -0.099 0.342 -0.772 -0.120 -0.101 0.404 -0.104 0.346
x2  -0.303 0.395 -1.081 -0.328 -0.309 0.467 -0.231 0.385
x3   0.005 0.356 -0.695 -0.017  0.005 0.420  0.005 0.350
x4   1.354 0.335  0.695  1.333  1.381 0.396  1.297 0.336
x5   1.776 0.283  1.219  1.758  1.811 0.334  1.771 0.286
x6   0.454 0.366 -0.267  0.432  0.463 0.433  0.402 0.357
x7   0.088 0.352 -0.604  0.066  0.089 0.415  0.003 0.344
x8   0.474 0.435 -0.383  0.448  0.484 0.515  0.393 0.419
x9  -0.550 0.370 -1.278 -0.573 -0.561 0.437 -0.434 0.358
x10 -0.397 0.437 -1.257 -0.424 -0.405 0.517 -0.307 0.419
x11 -0.408 0.362 -1.120 -0.430 -0.416 0.428 -0.361 0.357
x12 -1.194 0.486 -2.150 -1.224 -1.218 0.574 -1.040 0.455
x13 -0.247 0.393 -1.020 -0.271 -0.252 0.464 -0.225 0.378
x14  0.141 0.448 -0.741  0.113  0.143 0.529  0.078 0.417
x15 -0.610 0.313 -1.225 -0.629 -0.622 0.370 -0.569 0.317
```

5…Bayesian Analysis of the Normal Regression Model

Zellner's g-prior with Bayesm

```
summary(bayesm.model$sigmasqdraw)
res$s2
```

5…Bayesian Analysis of the Normal Regression Model



80

5…Bayesian Analysis of the Normal Regression Model

Zellner's g-prior with BAS

The g-prior is mainly used for variable selection.

Hence it can be also found in BAS package that we will explore lateron

5…Bayesian Analysis of the Normal Regression Model

Jeffreys prior

Improper non-informative prior $f(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$

Posterior

NIG similar to the one obtained by the conjugate analysis with $\mathbf{V}^{-1}=0$, a=-p/2+1 and b=0

$$\widetilde{\beta} = \widehat{\beta}, \quad SS = RSS, \quad \text{and}$$
$$E(\sigma^2|\mathbf{y}) = RSS/(n-p) = \tilde{\sigma}_u^2$$

5…Bayesian Analysis of the Normal Regression Model

(Approx) Jeffreys prior with bayes.lm

```
y<-simex1$y
x<-cbind( 1, as.matrix(simex1[,-1] ) )
n<-length(y)
p<-ncol(x)

g<- 1000*n
S <- g*solve( t(x) %*% x )
res.j<-lm.bayes( y, x, prior.mean=rep(0,p), V=S,
    add.constant=FALSE, a=(-16/2+1), b=0.001 )
```

5…Bayesian Analysis of the Normal Regression Model

Jeffreys prior with bayes.lm

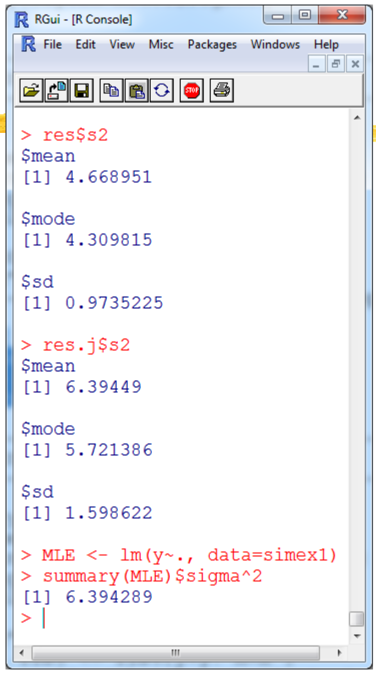
RGui - [R Console]

File Edit View Misc Packages Windows Help

```
> res.j<-lm.bayes( y, x, prior.mean=rep(0,p), V=$
      Mean  S.D.  2.5%Q 47.5%Q    MLE  S.E.
     0.254 0.441 -0.615  0.227  0.254 0.441
x1  -0.101 0.404 -0.897 -0.126 -0.101 0.404
x2  -0.309 0.467 -1.230 -0.338 -0.309 0.467
x3   0.005 0.420 -0.823 -0.020  0.005 0.420
x4   1.381 0.396  0.601  1.356  1.381 0.396
x5   1.811 0.334  1.152  1.791  1.811 0.334
x6   0.463 0.433 -0.390  0.437  0.463 0.433
x7   0.089 0.415 -0.729  0.064  0.089 0.415
x8   0.484 0.515 -0.530  0.452  0.484 0.515
x9  -0.561 0.437 -1.422 -0.588 -0.561 0.437
x10 -0.405 0.517 -1.423 -0.437 -0.405 0.517
x11 -0.416 0.428 -1.260 -0.442 -0.416 0.428
x12 -1.218 0.574 -2.349 -1.253 -1.218 0.574
x13 -0.252 0.464 -1.167 -0.280 -0.252 0.464
x14  0.143 0.529 -0.900  0.111  0.143 0.529
x15 -0.622 0.370 -1.351 -0.644 -0.622 0.370
>
```

Jeffreys prior with bayes.lm

```
#comparison of error variances

res$s2
res.j$s2
MLE <- lm(y~.,
    data=simex1)
summary(MLE)$sigma^2
```



5…Bayesian Analysis of the Normal Regression Model

(Approx) Jeffreys prior with Bayesm

```
library(bayesm)
y<-simex1$y;x<-cbind( 1, as.matrix(simex1[,-1] ) )
n<-length(y);p<-ncol(x);g<-n
S <- solve( t(x) %*% x )/g
simex1.list <- list(y=y, X=x )
prior1 <- list( betabar=rep(0,p), A=S, nu=-14, ssq=0)
mcmc1  <- list( R=5000, keep=1 )
bayesm.model <- runireg(Data=simex1.list, Prior=prior1,
    Mcmc=mcmc1)
summary(bayesm.model$betadraw)
```

5…Bayesian Analysis of the Normal Regression Model

(Approx) Jeffreys prior with Bayesm

… …

```
round(cbind(res.j$beta, apply(
    bayesm.model$betadraw,2,mean )
              , apply( bayesm.model$betadraw,2,sd )), 3)
summary(bayesm.model$sigmasqdraw)
res.j$s2
```

(Approx) Jeffreys prior with Bayesm

```
RGui - [R Console]
File  Edit  View  Misc  Packages  Windows  Help

> round(cbind(res.j$beta, apply( bayesm.model$betadraw,2,mean$
+                  , apply( bayesm.model$betadraw,2,sd )), 3)
      Mean  S.D.  2.5%Q 47.5%Q    MLE  S.E.
     0.254 0.441 -0.615  0.227  0.254 0.441  0.258 0.434
x1  -0.101 0.404 -0.897 -0.126 -0.101 0.404 -0.099 0.407
x2  -0.309 0.467 -1.230 -0.338 -0.309 0.467 -0.318 0.465
x3   0.005 0.420 -0.823 -0.020  0.005 0.420 -0.003 0.413
x4   1.381 0.396  0.601  1.356  1.381 0.396  1.383 0.396
x5   1.811 0.334  1.152  1.791  1.811 0.334  1.813 0.332
x6   0.463 0.433 -0.390  0.437  0.463 0.433  0.456 0.428
x7   0.089 0.415 -0.729  0.064  0.089 0.415  0.085 0.419
x8   0.484 0.515 -0.530  0.452  0.484 0.515  0.480 0.513
x9  -0.561 0.437 -1.422 -0.588 -0.561 0.437 -0.560 0.433
x10 -0.405 0.517 -1.423 -0.437 -0.405 0.517 -0.399 0.524
x11 -0.416 0.428 -1.260 -0.442 -0.416 0.428 -0.417 0.431
x12 -1.218 0.574 -2.349 -1.253 -1.218 0.574 -1.228 0.575
x13 -0.252 0.464 -1.167 -0.280 -0.252 0.464 -0.251 0.470
x14  0.143 0.529 -0.900  0.111  0.143 0.529  0.149 0.535
x15 -0.622 0.370 -1.351 -0.644 -0.622 0.370 -0.607 0.369
>
>
> summary(bayesm.model$sigmasqdraw)
Summary of Posterior Marginal Distributions
Moments
  mean std dev num se rel eff sam size
1  6.4     1.6  0.027     1.3     2250
```

6… Logistic Regression using MCMCpack

Example from Ntzoufras (2009, p. 263)

7.5.2 A simple example

Example 7.3 . Analysis of senility symptoms data using WinBUGS (data were originally analyzed in Example 2.3). In Section 2.3 we have illustrated the Metropolis–Hastings algorithm using a simple example in which 54 elderly people completed a subtest of the Wechsler Adult Intelligence Scale (WAIS) resulting to a discrete score with ranging from 0 to 20. The aim of this study was to identify people with senility symptoms (binary variable) using the WAIS score. Moreover, we were interested in calculating the threshold value of X for which $\pi > 0.5$ to enable us to identify possible patients directly using X.

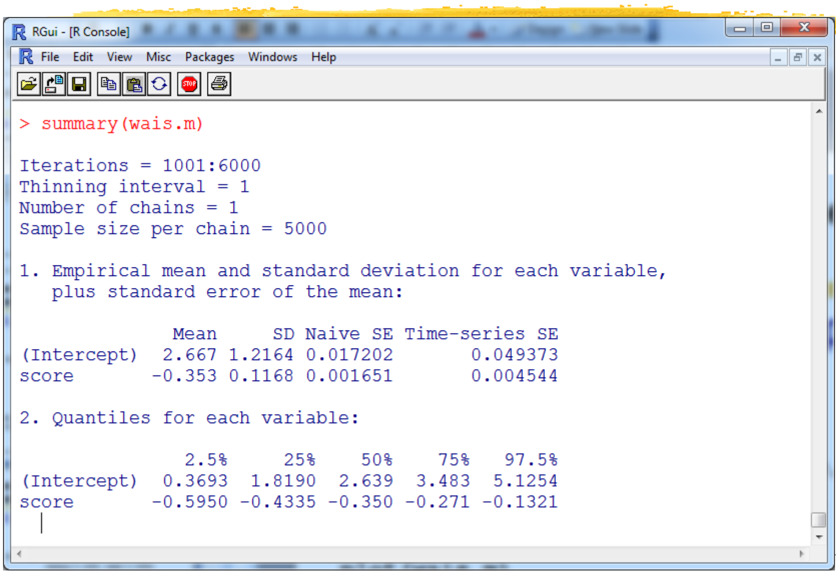
6… Logistic Regression using MCMCpack

A logistic regression example

R code

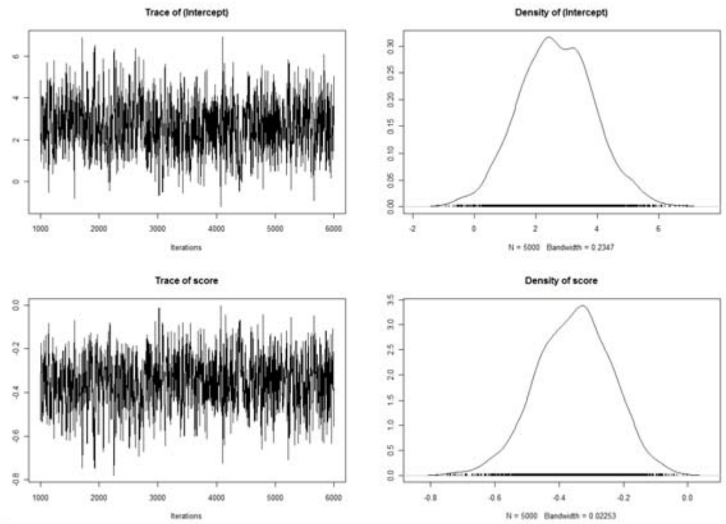
```
wais.m <- MCMClogit(senility~score, data=wais,
                    burnin = 1000, mcmc = 5000,
                    b0=0, B0=0)
                    # b0 prior mean for β
                    # B0 prior precision matrix for β
                    # B0=0 => improper flat prior
summary(wais.m)
plot(wais.m)
```

6… Logistic Regression using MCMCpack



```
> summary(wais.m)

Iterations = 1001:6000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean     SD Naive SE Time-series SE
(Intercept)  2.667 1.2164 0.017202       0.049373
score       -0.353 0.1168 0.001651       0.004544

2. Quantiles for each variable:

               2.5%     25%    50%    75%   97.5%
(Intercept)  0.3693  1.8190  2.639  3.483  5.1254
score       -0.5950 -0.4335 -0.350 -0.271 -0.1321
```

6… Logistic Regression using MCMCpack



92

6… Logistic Regression using MCMCpack

Posterior summaries for OR

```
> summary(exp(wais.m[,2]))

Iterations = 1001:6000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean             SD       Naive SE Time-series SE
      0.707344       0.082102       0.001161       0.003174

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.5515 0.6483 0.7047 0.7626 0.8762
```

93

6… Logistic Regression using MCMCpack

Posterior summaries for P(Y=1|X=0)

```
> summary( 1/(1+exp(-wais.m[,1])) )

Iterations = 1001:6000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean             SD       Naive SE Time-series SE
      0.895935       0.110773       0.001567       0.004582

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.5913 0.8605 0.9333 0.9702 0.9941
```

94

6… Logistic Regression using arm package

```
wais.m2 <- bayesglm (senility~score, data=wais,
    family = binomial,
    prior.mean = 0,
    prior.scale = NULL,
    prior.df = 1,
    prior.mean.for.intercept = 0,
    prior.scale.for.intercept = NULL,
    prior.df.for.intercept = 1)

summary(wais.m2)
```

7… Poisson Regression using MCMCpack

Example from Ntzoufras (2009, p. 245)

7.4.2 A simple Poisson regression example

Example 7.1. Aircraft damage dataset. Here we consider the aircraft damage dataset of Montgomery et al. (2006). The dataset refers to the number of aircraft damages in 30 strike missions during the Vietnam war. Hence it consists of 30 observations and the following four variables:

- `damage`: the number of damaged locations of the aircraft
- `type`: binary variable which indicates the type of plane (0 for A4; 1 for A6)
- `bombload`: the aircraft bomb load in tons
- `airexp`: the total months of aircrew experience

In this example we can use the Poisson distribution to monitor the number of damages after each mission.

7… Poisson Regression using MCMCpack

A Poisson regression example

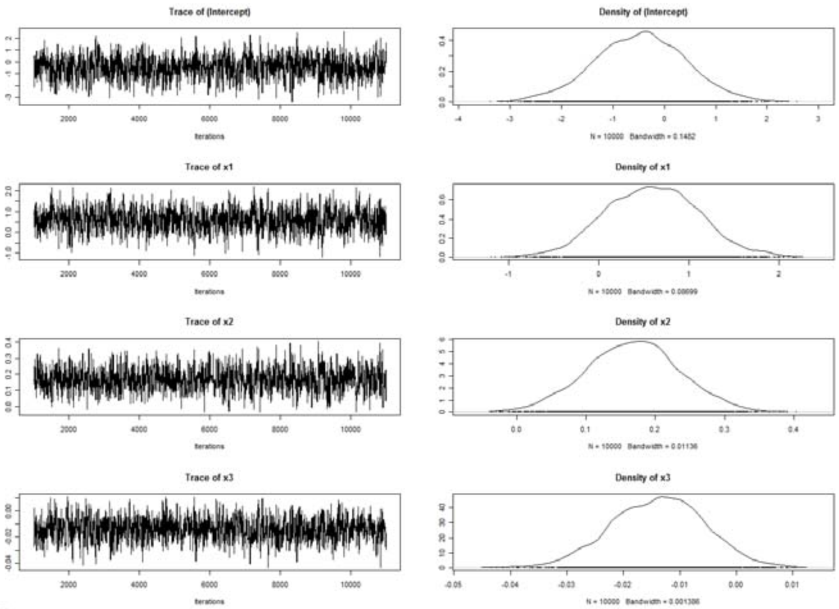
R code

```
air.m <- MCMCpoisson (y~x1+x2+x3 , data=aircraft,
                      burnin = 1000, mcmc = 10000,
                      b0=0, B0=0)
                      # b0 prior mean for β
                      # B0 prior precision matrix for β
                      # B0=0 => improper flat prior
summary(air.m)
plot(air.m)
```

7… Poisson Regression using MCMCpack

```
> summary(air.m)

Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean       SD  Naive SE Time-series SE
(Intercept) -0.46681 0.882043 8.820e-03      0.0330128
x1           0.59112 0.517803 5.178e-03      0.0239782
x2           0.16832 0.067829 6.783e-04      0.0024106
x3          -0.01386 0.008252 8.252e-05      0.0003069

2. Quantiles for each variable:

                2.5%      25%      50%       75%    97.5%
(Intercept) -2.25389 -1.06327 -0.45349  0.142358 1.270406
x1          -0.42339  0.22035  0.58963  0.941490 1.613255
x2           0.03565  0.12188  0.16830  0.212461 0.300761
x3          -0.03029 -0.01951 -0.01353 -0.008139 0.001739
```

7… Poisson Regression using MCMCpack



7… Poisson Regression using MCMCpack

Estimating the posterior of the Odds

```
> summary(exp(air.m))

Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean       SD  Naive SE Time-series SE
(Intercept) 0.9218 0.966162 9.662e-03      0.0349498
x1          2.0655 1.143950 1.144e-02      0.0542637
x2          1.1860 0.080565 8.057e-04      0.0028793
x3          0.9863 0.008135 8.135e-05      0.0003025

2. Quantiles for each variable:

              2.5%    25%    50%    75% 97.5%
(Intercept) 0.1050 0.3453 0.6354 1.1530 3.562
x1          0.6548 1.2465 1.8033 2.5638 5.019
x2          1.0363 1.1296 1.1833 1.2367 1.351
x3          0.9702 0.9807 0.9866 0.9919 1.002
```

Expected value for X=0

% Change of mean

7… Poisson Regression using arm

```
air.m2 <- bayesglm (y~x1+x2+x3 , data=aircraft,
    family = poisson,
    prior.mean = 0,
    prior.scale = NULL,
    prior.df = 1,
    prior.mean.for.intercept = 0,
    prior.scale.for.intercept = NULL,
    prior.df.for.intercept = 1)

summary(air.m2)
```

8… Other models using MCMCpack

A large variety of models is available

- Probit regression for nominal and ordinal data
- Hierarchical models (regression, poisson, logistic, probit)
- Change-point models
- Factor analysis models
- Item response models

and others

[syntax is similar and hence very easy to use]

What else?

- R2WinBUGS
- Variable Selection using BAS