STAT 460 A5

Steve Hof

16/11/2020

1 (a)

Clearly determine the parameters of interest and their meaning in the context of linear regression

We have three predictor varialbes, x_1, x_2, x_3, which correspond to the advertising spend on Youtube, Facebook and Newspaper, respectively. Our response variable, y, is the sales.

In our regression, $\beta_1, \beta_2, \beta_3$ are the coefficients corresponding to our predictor variables. These are the ratios showing the increase in sales per unit increase in advertising spend. β_0 is our bias (intercept) term, which represents the expected sales when all advertising budgets are zero.

This gives us a regression model:

$$y_i = \beta_{i0} + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \epsilon_i$$

We assume that $\epsilon \sim \text{Normal}(0, \sigma^2)$, so the parameters of interest are $\vec{\beta}$ and σ^2

1 (b)

Write down the full conditionals

The full posterior conditional for β is

$$p(\beta|\sigma^2, y) \propto \exp\left\{\frac{-1}{2\sigma^2}\left(\beta - (X^T X)^{-1} X^T y\right)^T (X^T X)\left(\beta - (X^T X)^{-1} X^T y\right)\right\}$$

which means

$$\beta \sim \text{N}_p\left((X^T X)^{-1} X^T y, \;\; \sigma^2 (X^T X)^{-1}\right)$$

The full posterior conditional for σ^2 is

$$p(\sigma^2|\beta, y) \propto (\sigma^2)^{-\left(1+\frac{n}{2}\right)} \exp\left\{\frac{-1}{2\sigma^2}(y - X\beta)^T (y - X\beta)\right\}$$

which means

$$\sigma^2|\beta, y \sim \text{InverseGamma}\left(\frac{n}{2}, \;\; \frac{(y - X\beta)^T (y - X\beta)}{2}\right)$$

1 (c)

Using R, estimate the parameters using a Gibbs sampler algorithm with number of simulations $25,000$, a burn-in of $5,000$, and thinning of 30.

```
d = marketing
y = d$sales
X = cbind(rep(1, length(y)), d$youtube, d$facebook, d$newspaper)
X.df = as.data.frame(X)

n = length(y) # num rows
k = dim(X)[2] # num columns
NSim = 25000
Sigma2 = matrix(nrow = NSim, ncol = 1)
```

```
model = lm(y ~ ., data = X.df)
BetaLS = solve(t(X) %*% X) %*% t(X) %*%y
resids = model$residuals
s2 = (1 / (n-k)) * sum(resids^2)
Sigma = solve(t(X) %*% X)
```

```
gibbs.sampler = function(NSim, X, y, BetaLS, Sigma, k) {
  Sigma2Gibbs = matrix(nrow = NSim, ncol = 1)
  BetaGibbs = matrix(nrow = NSim, ncol = k)
  Sigma2Gibbs[1] = 1
  BetaGibbs[1, ] = rep(1, k)

  for (ite in 2:NSim){
  aux_sigma = t(y - X %*% BetaGibbs[ite-1,]) %*% (y - X %*% BetaGibbs[ite-1,])
  Sigma2Gibbs[ite] = rinvgamma(1, n/2, rate = aux_sigma/2, scale = 1/rate)
  BetaGibbs[ite,] = mvrnorm(n = 1, mu = BetaLS, Sigma = Sigma2Gibbs[ite]*Sigma)
  }

  # burn and thin
  burnin = 5000
  thin = 30
  Bgibbs = BetaGibbs[seq(from = burnin + 1, to = NSim, by = thin), ]
  S2gibbs = Sigma2Gibbs[seq(from = burnin + 1, to = NSim, by = thin)]
  return(cbind(Bgibbs, S2gibbs))
}

Bgibbs = gibbs.sampler(NSim, X, y, BetaLS, Sigma, k)[, 1:4]
S2gibbs = gibbs.sampler(NSim, X, y, BetaLS, Sigma, k)[, 5]
```

1 (d)

Summarize the distribution of each parameter using meaningful quantiles, the posterior mean, and the posterior standard deviation.

```
# Beta_0 Quantiles
round(quantile(Bgibbs[, 1]), 3)
```

```
##    0%   25%   50%   75%  100%
## 2.444 3.273 3.540 3.793 4.627
```

```r
# Beta_0 Mean
mean(Bgibbs[, 1])
```

```
## [1] 3.530118
```

```r
# Beta_0 Standard Deviation
sd(Bgibbs[, 1])
```

```
## [1] 0.3668748
```

```r
# Beta_1 Quantiles
quantile(Bgibbs[, 2])
```

```
##         0%        25%        50%        75%       100%
## 0.04211237 0.04477292 0.04573321 0.04659946 0.04949097
```

```r
# Beta_1 Mean
mean(Bgibbs[, 2])
```

```
## [1] 0.04569691
```

```r
# Beta_1 Standard Deviation
sd(Bgibbs[, 2])
```

```
## [1] 0.001365336
```

```r
# Beta_2 Quantiles
quantile(Bgibbs[, 3])
```

```
##        0%       25%       50%       75%      100%
## 0.1608320 0.1817342 0.1882087 0.1940664 0.2113774
```

```r
# Beta_2 Mean
mean(Bgibbs[, 3])
```

```
## [1] 0.1880352
```

```r
# Beta_2 Standard Deviation
sd(Bgibbs[, 3])
```

```
## [1] 0.008892952
```

```r
# Beta_3 Quantiles
quantile(Bgibbs[, 4])
```

```
##            0%           25%           50%           75%          100%
## -0.0187882852 -0.0051545445 -0.0007783763  0.0035753529  0.0150288369
```

```r
# Beta_3 Mean
mean(Bgibbs[, 4])
```

```
## [1] -0.0007857683
```

```r
# Beta_3 Standard Deviation
sd(Bgibbs[, 4])
```

```
## [1] 0.006132022
```

```r
# sigma^2 Quantiles
quantile(S2gibbs)
```

```
##       0%      25%      50%      75%     100%
## 2.851639 3.847591 4.091921 4.393705 5.431260
```

```r
# simga^2 Mean
mean(S2gibbs)
```

```
## [1] 4.120401
```

```r
# sigma^2 Standard Deviation
sd(S2gibbs)
```

```
## [1] 0.4121988
```

1 (e)

Based on the quantiles previously computed, construct a 95% credible interval for the paramters; based on them, comment on the significance of the parameters.

```r
ci.betahat0 = ci(Bgibbs[, 1], ci = .95, method = "HDI")
ci.betahat1 = ci(Bgibbs[, 2], ci = .95, method = "HDI")
ci.betahat2 = ci(Bgibbs[, 3], ci = .95, method = "HDI")
ci.betahat3 = ci(Bgibbs[, 4], ci = .95, method = "HDI")
ci.sigma2 = ci(S2gibbs, ci = .95, method = "HDI")

ci.betahat0
```

```
## # Highest Density Interval
##
## 95% HDI
## ------------
## [2.80, 4.18]
```

```r
ci.betahat1
```

```
## # Highest Density Interval
##
## 95% HDI
## ------------
## [0.04, 0.05]
```

```r
ci.betahat2
```

```
## # Highest Density Interval
##
## 95% HDI
## ------------
## [0.17, 0.21]
```

```r
ci.betahat3
```

```
## # Highest Density Interval
##
## 95% HDI
## -------------
## [-0.01, 0.01]
```

```r
ci.sigma2
```

```
## # Highest Density Interval
##
## 95% HDI
```

```
## ------------
## [3.25, 4.87]
```

The 95% credible intervals (HDI) for $\beta_i, i = 0, 1, 2, 3$ and σ^2 are given above. Not that the only interval to contain zero is β_3 which corresponds to the Newspaper advertising. This tells us that β_3 is **not** significant, while the all of the others are.

1 (f)

Run 5 different simulations using the same starting points and check for convergence by computing $\hat{R}$ and trace plots of all 5 MCMC chains for all parameters

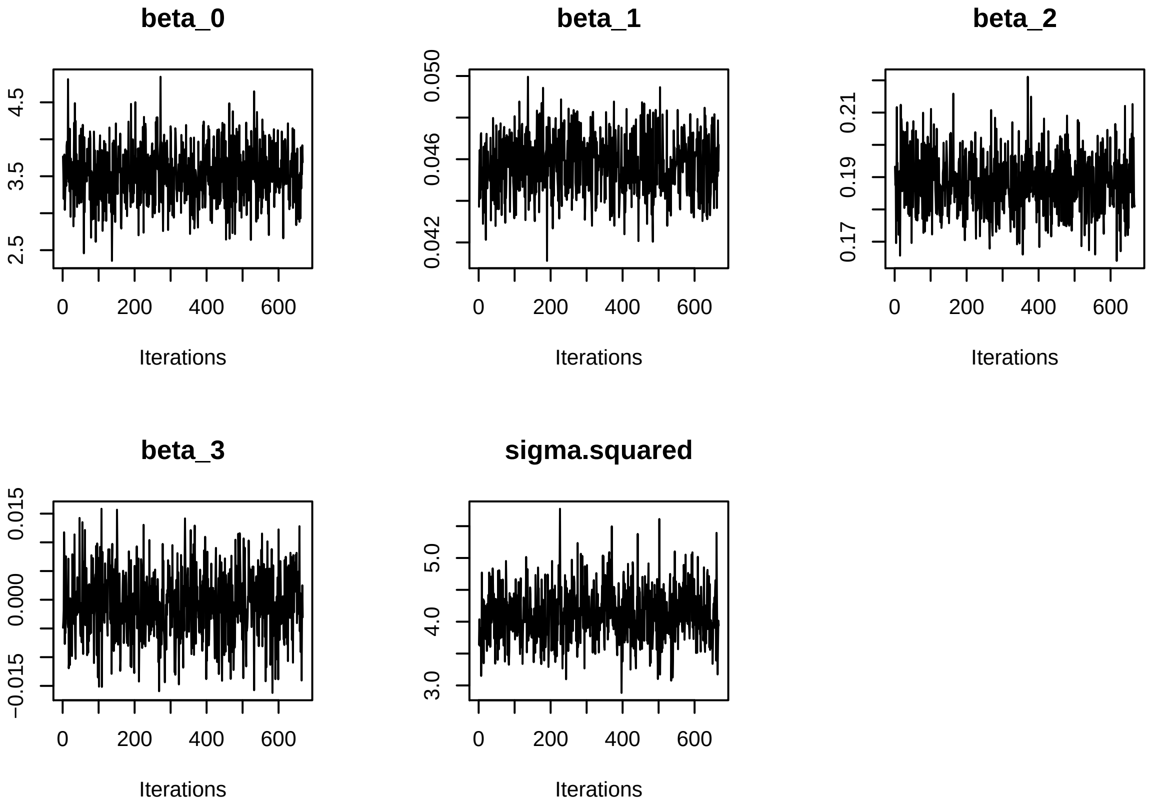
```
calc.rhat = function(final_result, m, nchain) {
  Rhat = NULL
  for (l in 1:5){
    psi_mean = mean(final_result[,l])
    psibar_j = NULL
    auxW = NULL

    for (q in 1:m){
      Subchain = final_result[seq((q - 1) * nchain + 1, q*nchain, 1), l]
      psibar_j[q] = mean(Subchain)
      auxW[q] = (1 / (nchain - 1)) * sum((Subchain - mean(Subchain))^2)
    }

    B = (nchain / (m - 1)) * (sum((psibar_j - psi_mean)^2))
    W = (1 / m) * sum(auxW)
    varPlus = ((nchain - 1) / nchain) * W + (1 / nchain) * B
    Rhat[l] = sqrt(varPlus / W)
  }
  Rhat = round(Rhat, 4)
  return(Rhat)
}
```

Simulation 1

```
resos = gibbs.sampler(NSim, X, y, BetaLS, Sigma, k)
rhat = calc.rhat(final_result = resos, m = 9, nchain = 74)
resos = as.mcmc(resos)
tits = c("beta_0", "beta_1", "beta_2", "beta_3", "sigma.squared")
par(mfrow = c(2,3))
for(pt in 1:5) {
  traceplot(resos[, pt], main = tits[pt])
}
```



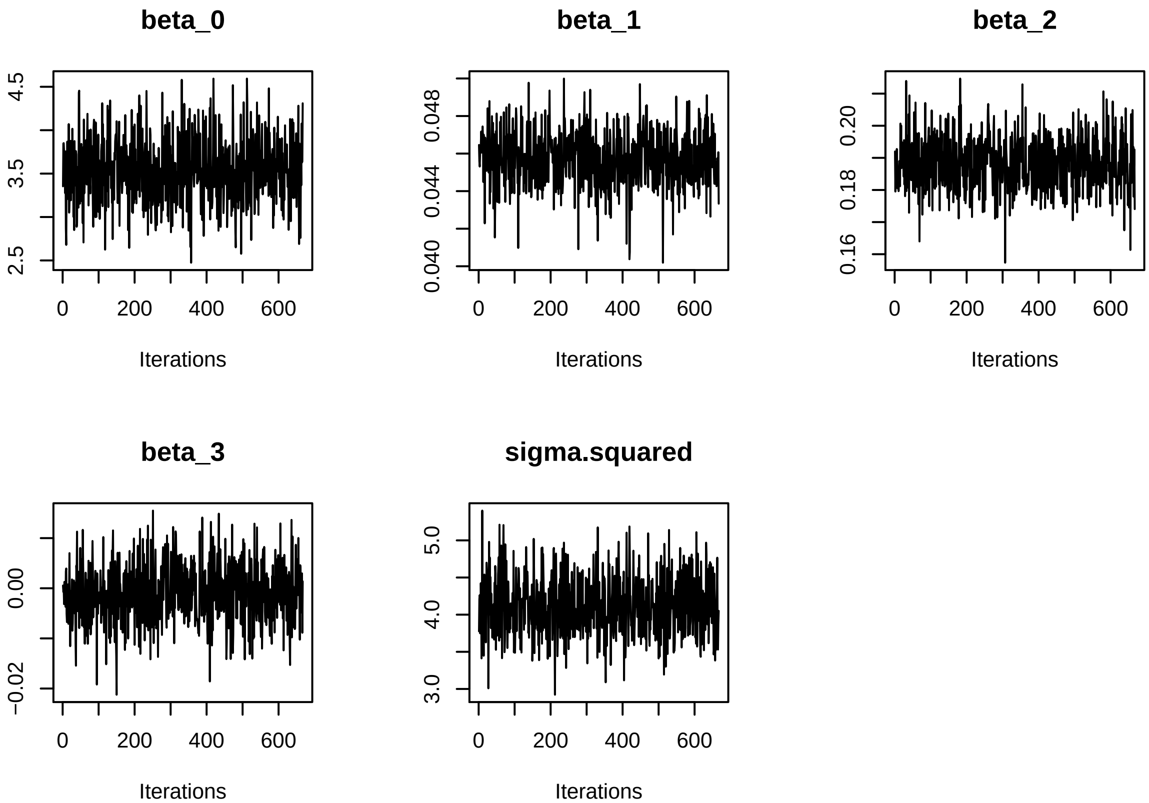
The $\hat{R}$ values:

1.0033, 1.003, 1.001, 0.998, 1.0048

are all close to one. This tells us that we have convergence.

Note: I've suppressed the code for the final 4 simulations so that the marker doesn't need to scroll through so many pages

Simulation 2

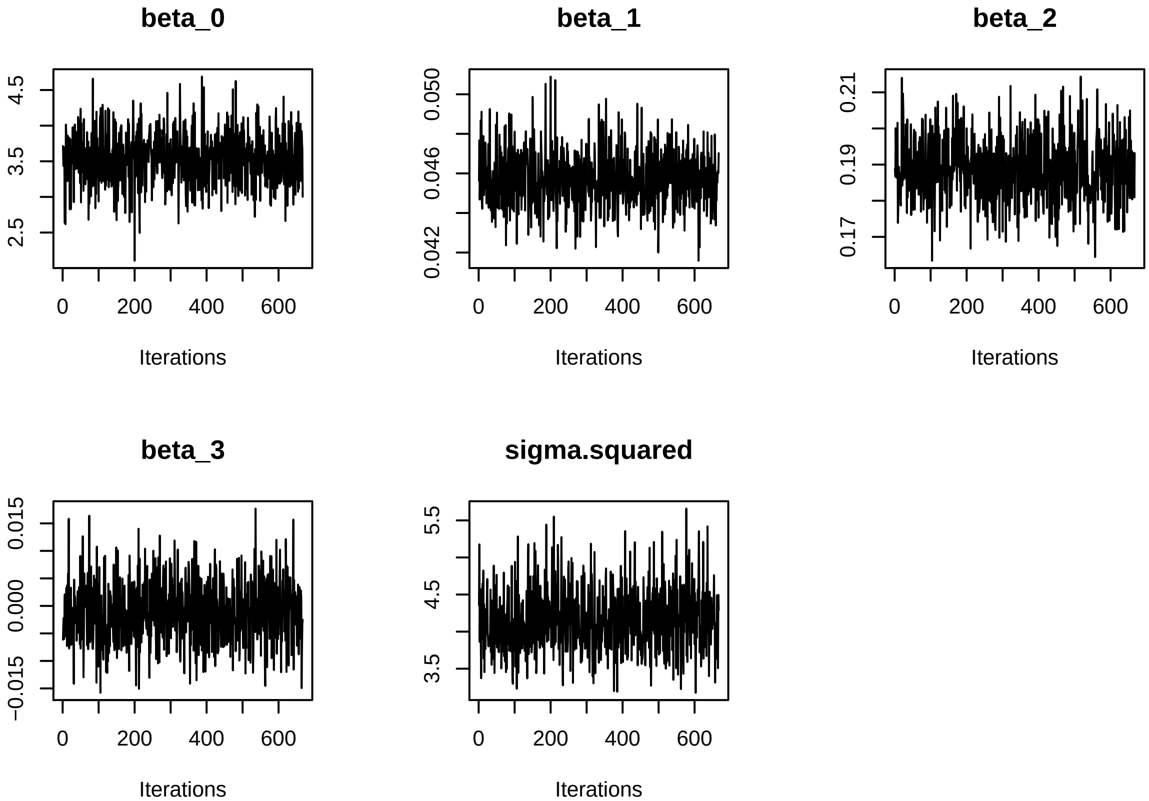


The $\hat{R}$ values:

1.0011, 1.0039, 0.9973, 1.0029, 0.996

are all close to one. This tells us that we have convergence.

Simulation 3

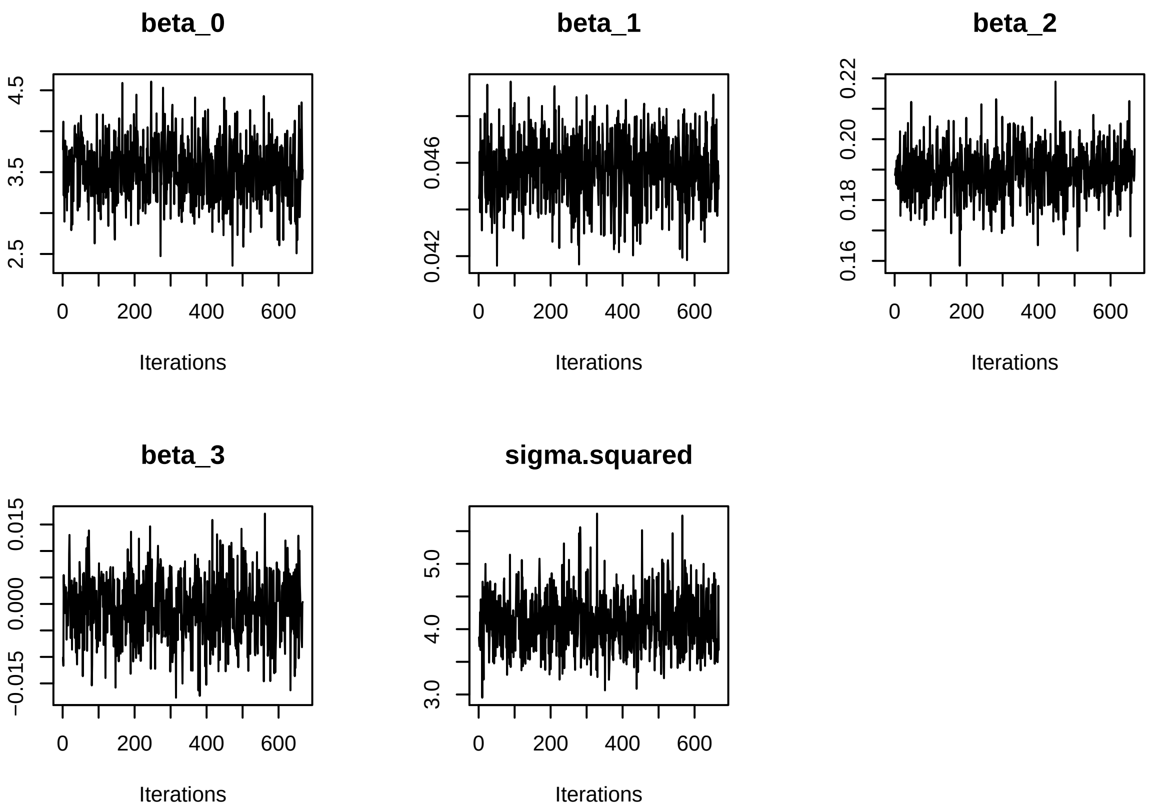


The $\hat{R}$ values:

0.9996, 0.9951, 1.0023, 1.0044, 1.003

are all close to one. This tells us that we have convergence.

Simulation 4

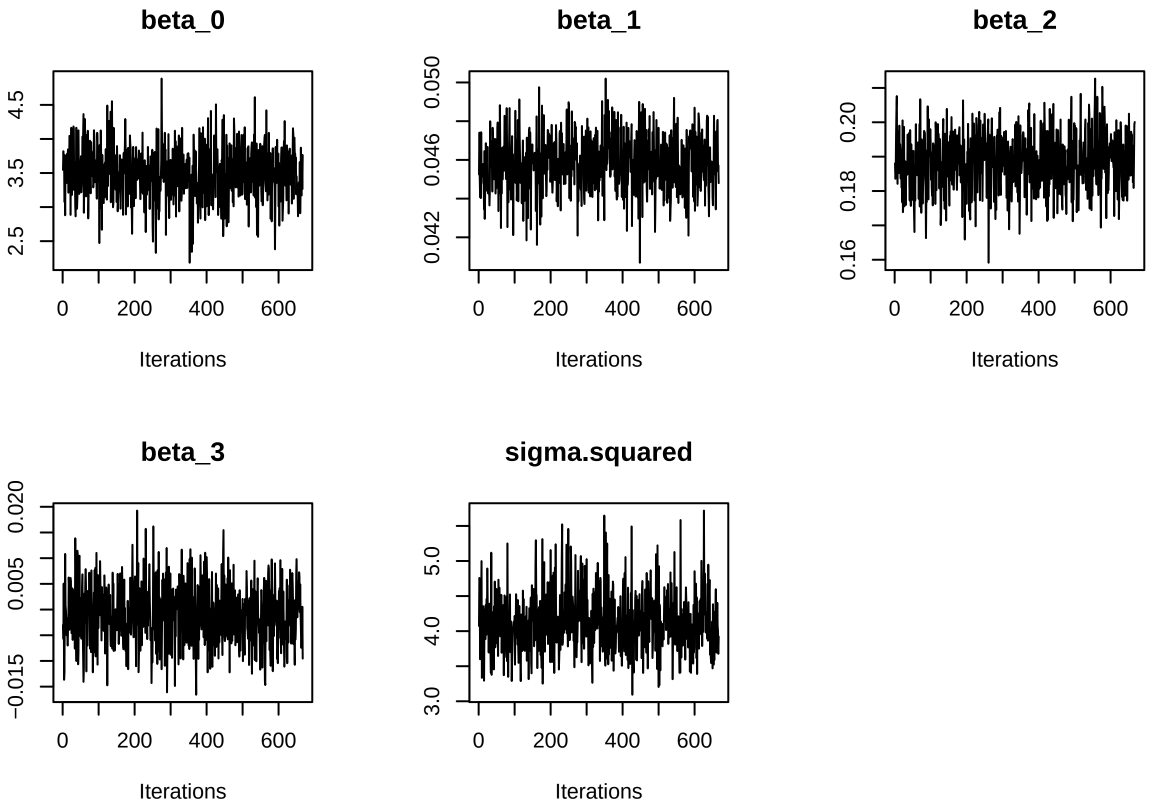


The $\hat{R}$ values:

0.9999, 1.0028, 1.0042, 0.9991, 0.9988

are all close to one. This tells us that we have convergence.

Simulation 5



The $\hat{R}$ values:

0.9994, 0.999, 0.9999, 0.9951, 1.0144

are all close to one. This tells us that we have convergence.