STAT 578 - Final Project

Part 1 - Load in the data and libraries

```r
library("rjags")
```

```
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
```

```r
library("lattice")

setwd("Z:/MSC-DS/2017 - Fall/STAT 578 - Advanced Bayesian Modeling/Final Project")

#load data, initialize starting values and set up a model
project_data <- read.csv(file="stat_578_data.csv")
```

Part 2(a)

- Take a subset of the data. Specifically, 1,000 random sample customers

```r
#Randomly select 1,000 rows of data
set.seed(123)
analysis_data = project_data[sample(nrow(project_data), 1000),]
```

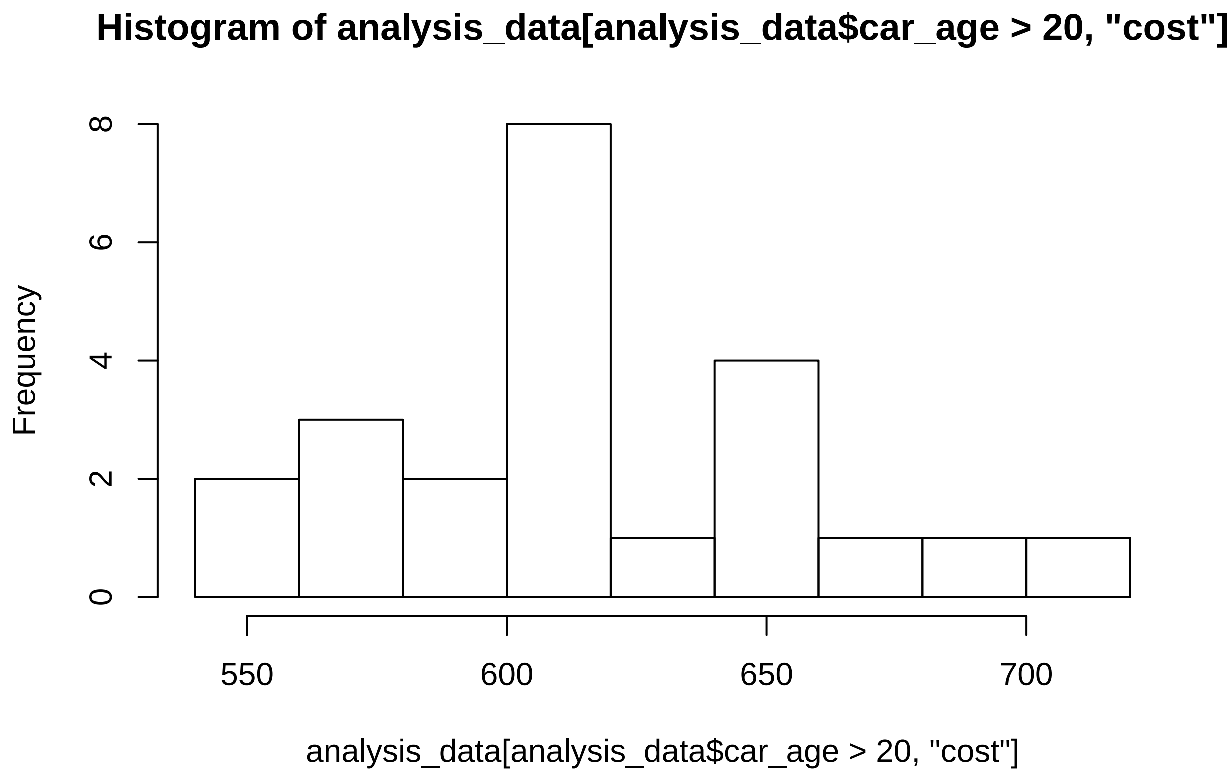
Part 2(b)

- For car_age > 20, change it to 20

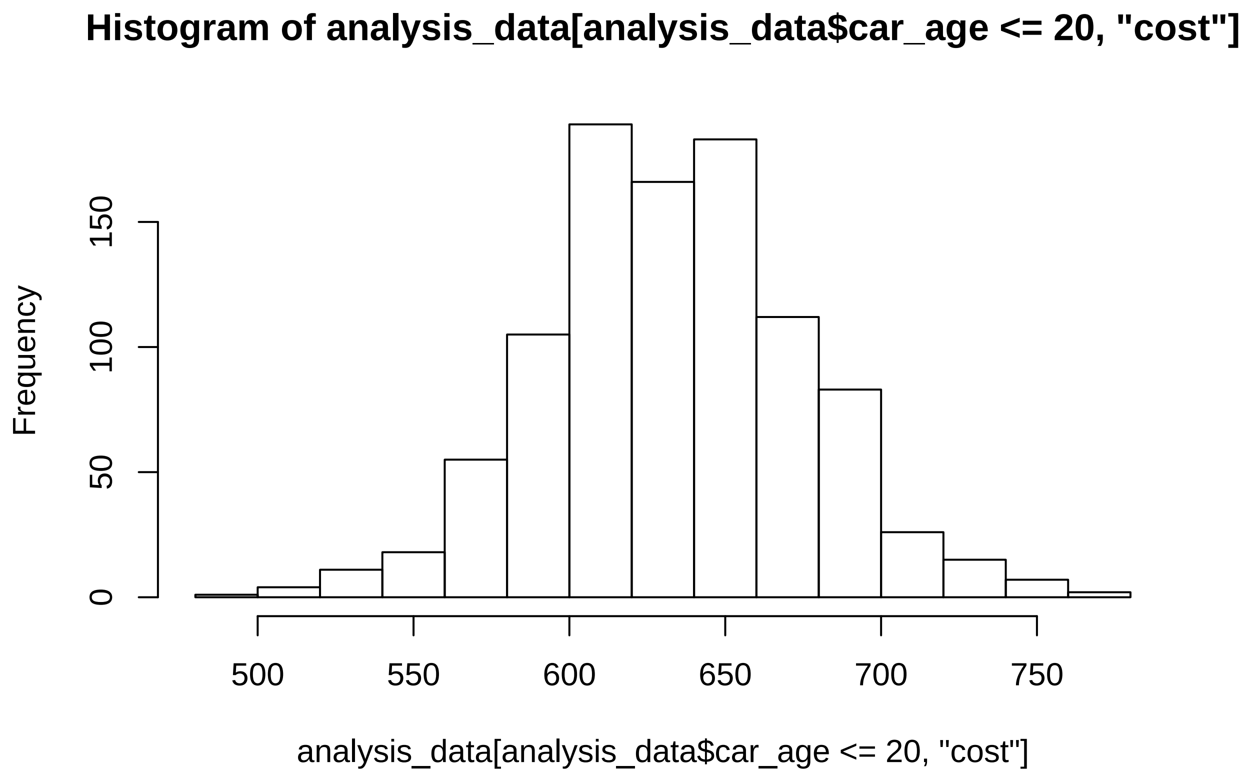
```r
#For car_age > 20, set it to 20 for the three reasons below:
#(1) most cars' useful life is less than 20 years.
#(2) the number of vehicles older than 20 make up about 2.3 % of the overall data
length(analysis_data[analysis_data$car_age > 20, "car_age"])/1000
```

```
## [1] 0.023
```

```r
#(3) the insurance cost for car_age > 20 seems to have the same mean as car_age <= 20
hist(analysis_data[analysis_data$car_age > 20, "cost"])
```




```r
hist(analysis_data[analysis_data$car_age <= 20, "cost"])
```




```
#So, for car_age > 20, set it to 20.
index <- analysis_data$car_age > 20
analysis_data$car_age[analysis_data$car_age>20] <- 20
```

Part 3(a)

-Fit a model for y[i] ~ beta_homeowner*owner[i] + beta_married_couple[i] + beta_age x car_age[i] -Check for convergence of the regression coefficients

```
d1 <- list( cost = analysis_data$cost
            ,owner = analysis_data$homeowner
            ,married = analysis_data$married_couple
            ,age = analysis_data$car_age
           )

inits1 <- list(
              list(beta_homeowner = 1000, beta_married = 1000
                   ,beta_age = 1000, sigmasqinv = 1000000,
                   .RNG.name = "base::Mersenne-Twister", .RNG.seed = 101)

              ,list(beta_homeowner = -1000, beta_married = 1000
                   ,beta_age = 1000, sigmasqinv = 0.0000001,
                   .RNG.name = "base::Mersenne-Twister", .RNG.seed = 103)

              ,list(beta_homeowner = 1000, beta_married = -1000
                   ,beta_age = -1000, sigmasqinv = 1000000,
```

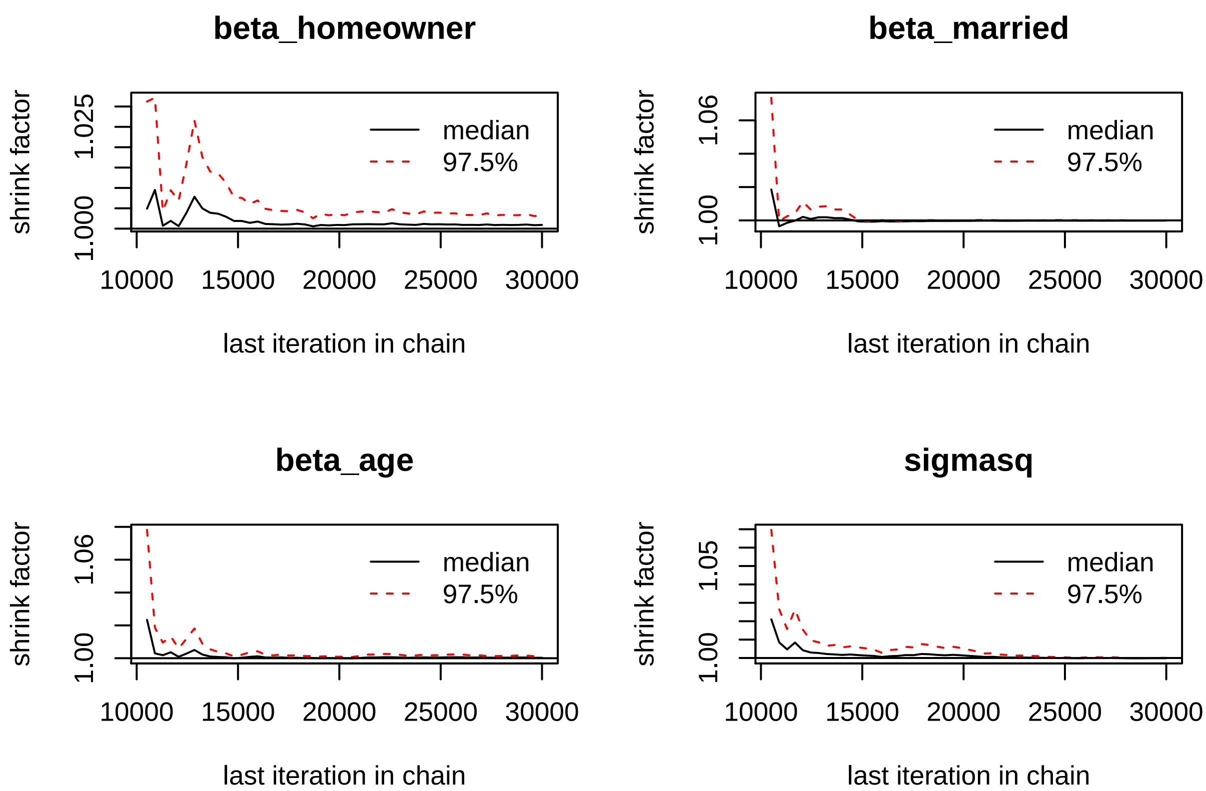
```r
                    .RNG.name = "base::Mersenne-Twister", .RNG.seed = 105)

               ,list(beta_homeowner = -1000, beta_married = -1000
                    ,beta_age = -1000, sigmasqinv = 0.0000001,
                    .RNG.name = "base::Mersenne-Twister", .RNG.seed = 107)
               )


m1 <- jags.model("final_project_cost1.bug",d1,inits1,n.chains=4,n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1000
##    Unobserved stochastic nodes: 1004
##    Total graph size: 5115
##
## Initializing model
```

```r
update(m1, 10000)

x1 <- coda.samples(m1, c("beta_homeowner","beta_married","beta_age","sigmasq","cost_rep"),n.iter=20000,

x1_sub <- x1[,c("beta_homeowner","beta_married","beta_age","sigmasq")]

gelman.diag(x1_sub,autoburnin=FALSE, multivariate=FALSE)
```

```
## Potential scale reduction factors:
##
##                Point est. Upper C.I.
## beta_homeowner          1          1
## beta_married            1          1
## beta_age                1          1
## sigmasq                 1          1
```

```r
gelman.plot(x1_sub,autoburnin=FALSE)
```




```r
effectiveSize(x1_sub)
```

```
## beta_homeowner    beta_married        beta_age         sigmasq
##       8723.118        7826.585        7986.745        7973.757
```

Part 3(b)

- Show summary of beta_homeowner, beta_married, beta_age, and sigmasq

```r
summary(x1_sub)
```

```
##
## Iterations = 10010:30000
## Thinning interval = 10
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                    Mean      SD Naive SE Time-series SE
## beta_homeowner   282.02   17.08  0.19092        0.18325
## beta_married     122.04   23.21  0.25948        0.26245
## beta_age          39.51    1.24  0.01386        0.01387
## sigmasq        90960.88 4087.85 45.70358       45.78405
##
```

```
## 2. Quantiles for each variable:
##
##                     2.5%      25%      50%      75%    97.5%
## beta_homeowner    247.76   270.42   282.05   293.50   315.67
## beta_married       76.08   106.69   122.20   137.65   167.38
## beta_age           37.04    38.66    39.51    40.35    41.91
## sigmasq         83206.25 88191.93 90892.40 93620.13 99331.61
```

Part 3(c)

- Check 95% confidence interval for statistical significance for beta_homeowner, beta_married and beta_age

```
post.samp1 <- as.matrix(x1)

##The 95% confidence interval of beta_homeowner does not include 0
#The mean
mean(post.samp1[,"beta_homeowner"])
```

```
## [1] 282.0235
```

```
#The 95% confidence interval
quantile(post.samp1[,"beta_homeowner"], c(0.025,0.975))
```

```
##     2.5%    97.5%
## 247.7557 315.6677
```

```
##The 95% confidence interval of beta_married does not include 0
#The mean
mean(post.samp1[,"beta_married"])
```

```
## [1] 122.0441
```

```
#The 95% confidence interval
quantile(post.samp1[,"beta_married"], c(0.025,0.975))
```

```
##      2.5%     97.5%
##  76.08077 167.37858
```

```
##The 95% confidence interval of beta_age does not include 0
#The mean
mean(post.samp1[,"beta_age"])
```

```
## [1] 39.5117
```

```
#The 95% confidence interval
quantile(post.samp1[,"beta_age"], c(0.025,0.975))
```

```
##     2.5%    97.5%
## 37.04250 41.91156
```

Part 3(d)

- Check dic for model in Part 3(c)

```
dic.samples(m1,50000)
```

```
## Mean deviance:  14255
## penalty 4
## Penalized deviance: 14259
```

Part 4(a)

-Fit a model for y[i] ~ beta_intercept + beta.age x car_age[i]
-Check for convergence of the regression coefficients

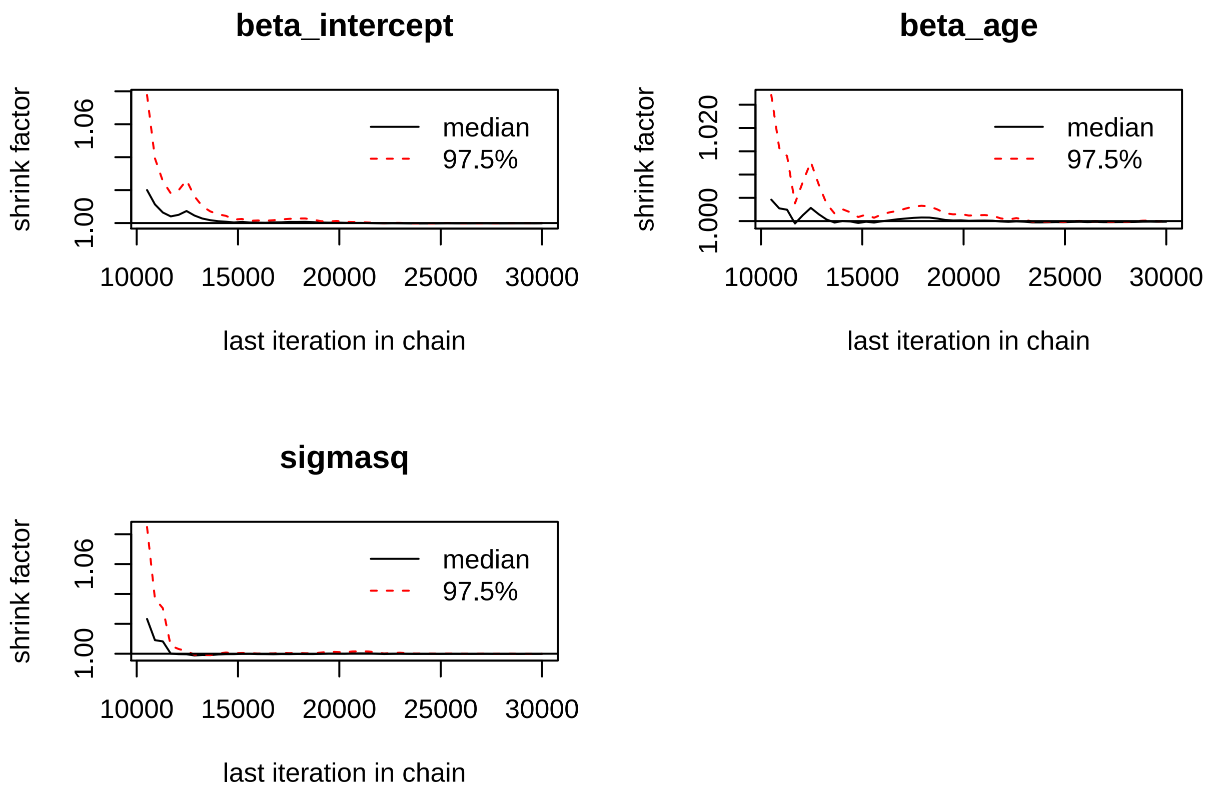
```r
#Coda summary of my results for the monitored parameters


d2 <- list( cost = analysis_data$cost
            ,age = analysis_data$car_age
           )

inits2 <- list(
              list(beta_intercept = 1000
                   ,beta_age = 1000, sigmasqinv = 1000000,
                   .RNG.name = "base::Mersenne-Twister", .RNG.seed = 101)

              ,list(beta_intercept = -1000
                   ,beta_age = 1000, sigmasqinv = 0.0000001,
                   .RNG.name = "base::Mersenne-Twister", .RNG.seed = 103)

              ,list(beta_intercept = 1000
                   ,beta_age = -1000, sigmasqinv = 1000000,
                   .RNG.name = "base::Mersenne-Twister", .RNG.seed = 105)

              ,list(beta_intercept = -1000
                   ,beta_age = -1000, sigmasqinv = 0.0000001,
                   .RNG.name = "base::Mersenne-Twister", .RNG.seed = 107)
              )


m2 <- jags.model("final_project_cost2.bug",d2,inits2,n.chains=4,n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1000
##    Unobserved stochastic nodes: 1003
##    Total graph size: 3050
##
## Initializing model
```

```r
update(m2, 10000)

x2 <- coda.samples(m2, c("beta_intercept","beta_age","sigmasq","cost_rep"),n.iter=20000, thin = 10)

x2_sub <- x2[,c("beta_intercept","beta_age","sigmasq")]

gelman.diag(x2_sub,autoburnin=FALSE, multivariate=FALSE)
```

```
## Potential scale reduction factors:
##
##                 Point est. Upper C.I.
## beta_intercept           1          1
## beta_age                 1          1
## sigmasq                  1          1
```

```r
gelman.plot(x2_sub,autoburnin=FALSE)
```




```r
effectiveSize(x2_sub)
```

```
## beta_intercept       beta_age        sigmasq
##       7889.800       8185.387       8000.000
```

Part 4(b)

- Show summary of beta_intercept, beta_age, and sigmasq

```r
summary(x2_sub)
```

```
##
## Iterations = 10010:30000
## Thinning interval = 10
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
```

```
##    plus standard error of the mean:
##
##                     Mean      SD Naive SE Time-series SE
## beta_intercept  655.898  2.2851 0.025548       0.025732
## beta_age         -2.749  0.2376 0.002656       0.002627
## sigmasq        1552.360 69.2025 0.773708       0.773834
##
## 2. Quantiles for each variable:
##
##                     2.5%      25%     50%      75%    97.5%
## beta_intercept  651.400  654.384  655.91  657.396  660.456
## beta_age         -3.223   -2.907   -2.75   -2.591   -2.273
## sigmasq        1424.534 1504.221 1549.57 1597.922 1693.777
```

Part 4(c)

- Check 95% confidence interval for statistical significance for beta_intercept and beta_age

```r
post.samp2 <- as.matrix(x2)

##The 95% confidence interval of beta_intercept does not include 0
#The mean
mean(post.samp2[,"beta_intercept"])
```

```
## [1] 655.8977
```

```r
#The 95% confidence interval
quantile(post.samp2[,"beta_intercept"], c(0.025,0.975))
```

```
##     2.5%    97.5%
## 651.3997 660.4561
```

```r
##The 95% confidence interval of beta_age does not include 0
#The mean
mean(post.samp2[,"beta_age"])
```

```
## [1] -2.748917
```

```r
#The 95% confidence interval
quantile(post.samp2[,"beta_age"], c(0.025,0.975))
```

```
##      2.5%     97.5%
## -3.223461 -2.272747
```

Part 4(d)

- Check dic for model in Part 4(c)

```r
dic.samples(m2,50000)
```

```
## Mean deviance:  10184
## penalty 3.016
## Penalized deviance: 10187
```

Part 5(a)

-Fit the following loglinear model
num_quotes[i] ~ dpois(lambda[i])
log(lambda[i]) <- logtime + beta_intercept + beta_cost*cost_scaled[i]
-Check for convergence for statistical significance for the regression coefficients

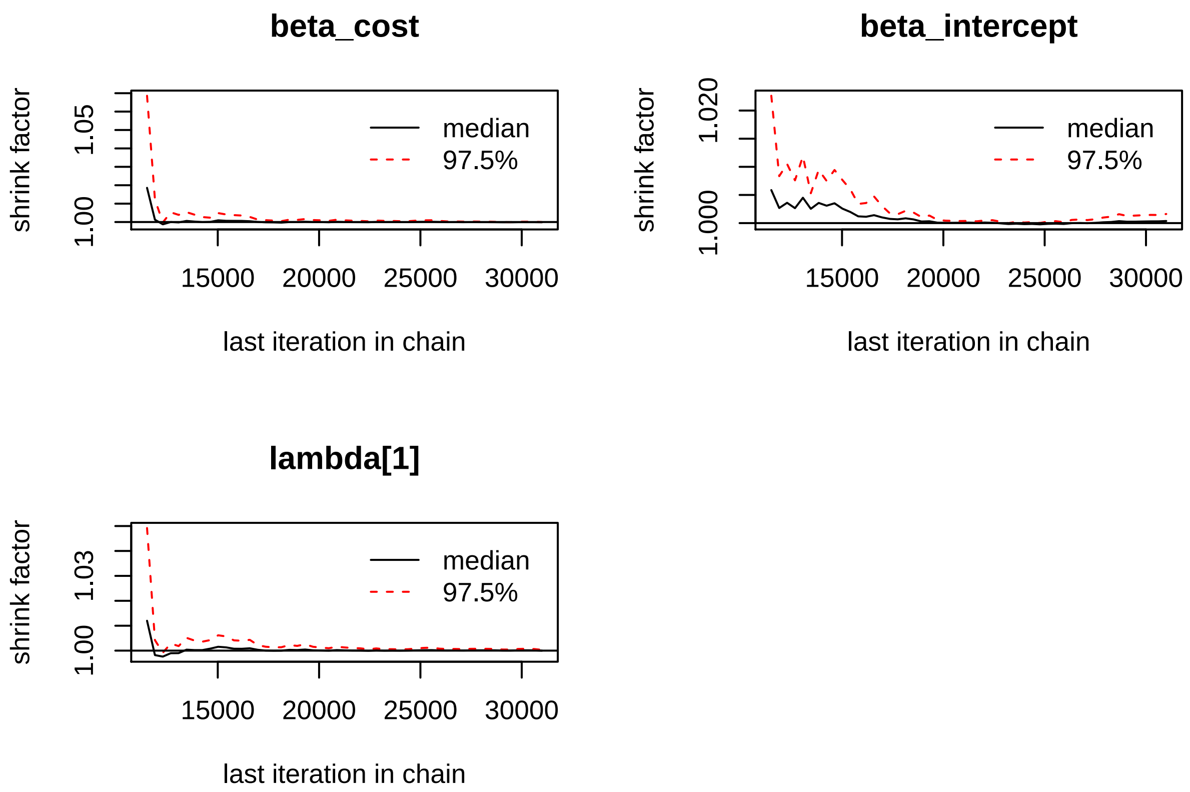
```
d3 <- list (num_quotes = analysis_data$shopping_pt
            ,logtime = log(1)
            ,cost_scaled = as.vector(scale(analysis_data$cost, scale=1*sd(analysis_data$cost)))
            )

inits3 <- list(list(beta_intercept = 100 , beta_cost = 100
               ,.RNG.name = "base::Mersenne-Twister", .RNG.seed = 101)

               ,list(beta_intercept = -100 , beta_cost = 100
               ,.RNG.name = "base::Mersenne-Twister", .RNG.seed = 103)

               ,list(beta_intercept = 100 , beta_cost = -100
               ,.RNG.name = "base::Mersenne-Twister", .RNG.seed = 105)

               ,list(beta_intercept = -100 , beta_cost = -100
               ,.RNG.name = "base::Mersenne-Twister", .RNG.seed = 107)

              )

m3 <- jags.model("final_project_point1.bug", d3, inits3, n.chains=4, n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1000
##    Unobserved stochastic nodes: 1002
##    Total graph size: 3614
##
## Initializing model
```

```
update(m3, 10000)

x3 <- coda.samples(m3, c("beta_intercept","beta_cost", "num_quotes_rep", "lambda"), n.iter=20000, thin =

gelman.diag(x3[,1:3], autoburnin=FALSE)
```

```
## Potential scale reduction factors:
##
##                Point est. Upper C.I.
## beta_cost               1          1
## beta_intercept          1          1
## lambda[1]               1          1
##
## Multivariate psrf
##
## 1
```

```r
gelman.plot(x3[,1:3], autoburnin=FALSE)
```




```r
effectiveSize(x3[,1:3])
```

```
##      beta_cost beta_intercept      lambda[1]
##       7695.369       8755.014       7761.156
```

Part (5)(b)

- Show summary of beta_cost and beta_intercept

```r
summary(x3[,1:2])
```

```
##
## Iterations = 11010:31000
## Thinning interval = 10
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                   Mean      SD  Naive SE Time-series SE
## beta_cost      0.02051 0.01221 0.0001365      0.0001394
## beta_intercept 1.91241 0.01225 0.0001370      0.0001316
##
```

```
## 2. Quantiles for each variable:
##
##                     2.5%     25%     50%     75%   97.5%
## beta_cost      -0.003667 0.01228 0.02048 0.02881 0.04439
## beta_intercept  1.888226 1.90432 1.91244 1.92056 1.93647
```

Part (5)(c)

- Check 95% confidence interval for statistical significance for beta_intercept and beta_cost

```r
post.samp3 <- as.matrix(x3)

##The 95% confidence interval of beta_cost does not include 1
quantile(exp(post.samp3[,"beta_intercept"]),c(0.025,0.975))
```

```
##     2.5%    97.5%
## 6.607635 6.934211
```

```r
##The 95% confidence interval of beta_cost includes 1
quantile(exp(post.samp3[,"beta_cost"]),c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.9963401 1.0453881
```

Part (5)(d)

-The p-value for the Chi-square test is 1. This indicates that the variance of the data is smaller than what the Poisson distribution assumes.

```r
post.samp3 <- as.matrix(x3)

lambdas <- post.samp3[,paste("lambda[",1:nrow(analysis_data),"]", sep="")]

num_quotes_srep <- post.samp3[,paste("num_quotes_rep[",1:nrow(analysis_data),"]", sep="")]

Tchi <- numeric(nrow(num_quotes_srep))
Tchirep <- numeric(nrow(num_quotes_srep))

for(s in 1:nrow(num_quotes_srep)) {
  Tchi[s] <- sum((analysis_data$shopping_pt - lambdas[s,])^2/lambdas[s,])
  Tchirep[s] <- sum((num_quotes_srep[s,]-lambdas[s,])^2/lambdas[s,])
}

mean(Tchirep >= Tchi)
```

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

Part 5(e)

- Check dic for model in Part 5

```r
dic.samples(m3,50000)
```

```
## Mean deviance:  4244
## penalty 2.003
## Penalized deviance: 4246
```