Bayesian Regression Analyses

Christoph Völtzke

15 6 2022

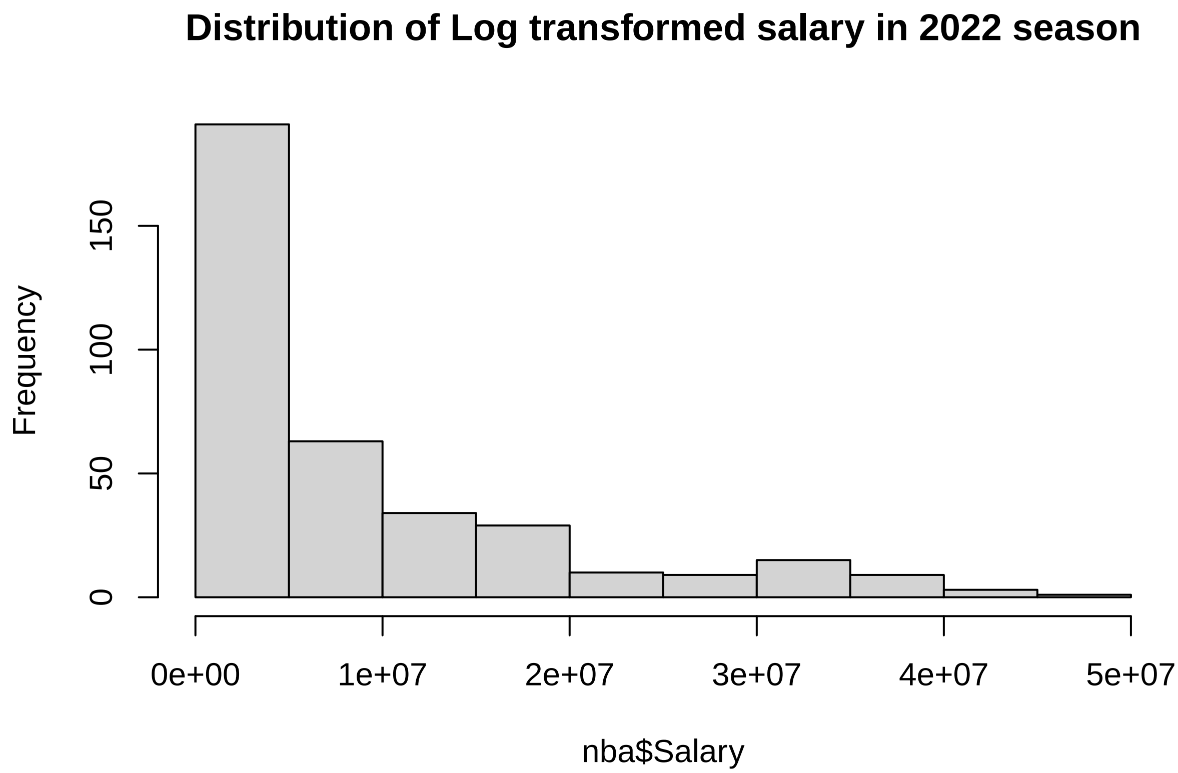
```
library(tidyverse)
library(readr)
library(bain)
library(papaja)
library(tinylabels)
library(kableExtra)
library(RColorBrewer)
```

Introduction to the data

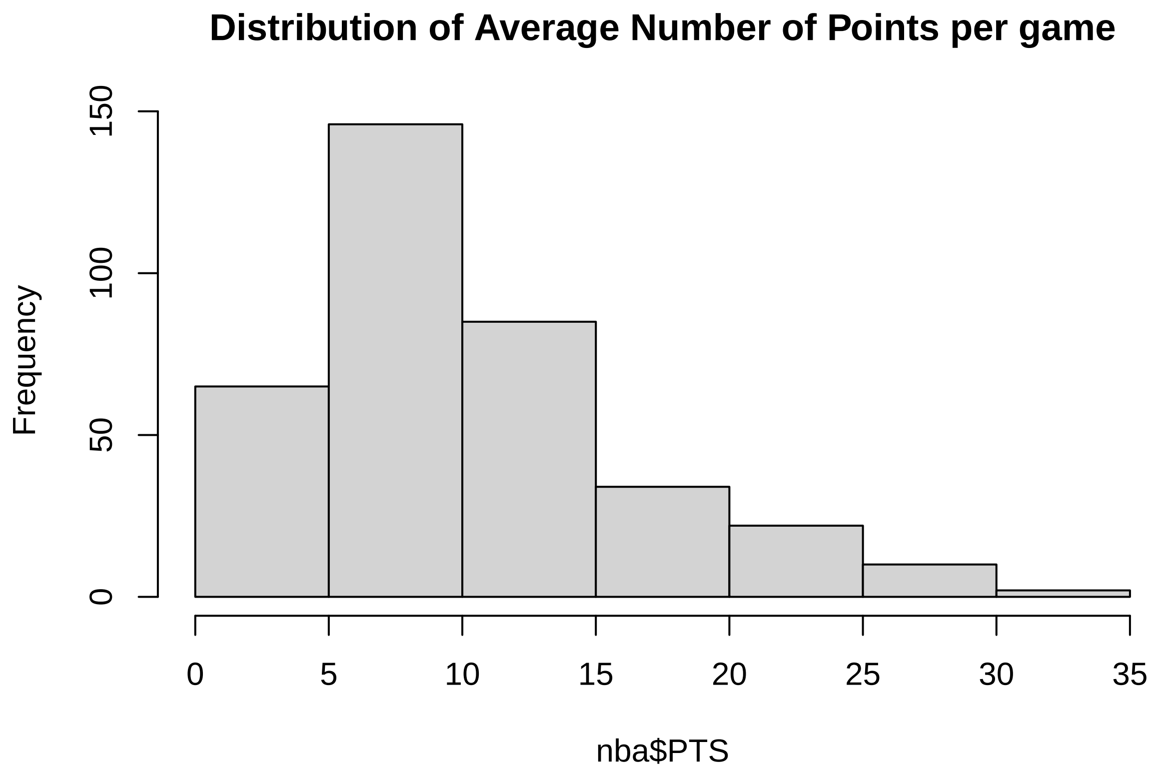
In this file all analyses for different combinations of predictors is conducted. Not all of these analyses are conducted in the original manuscript file.

```
# Load the data - I mention in the text where I got the data from
nba <- read.table("Data/nba.csv", sep=",", header=T)
```

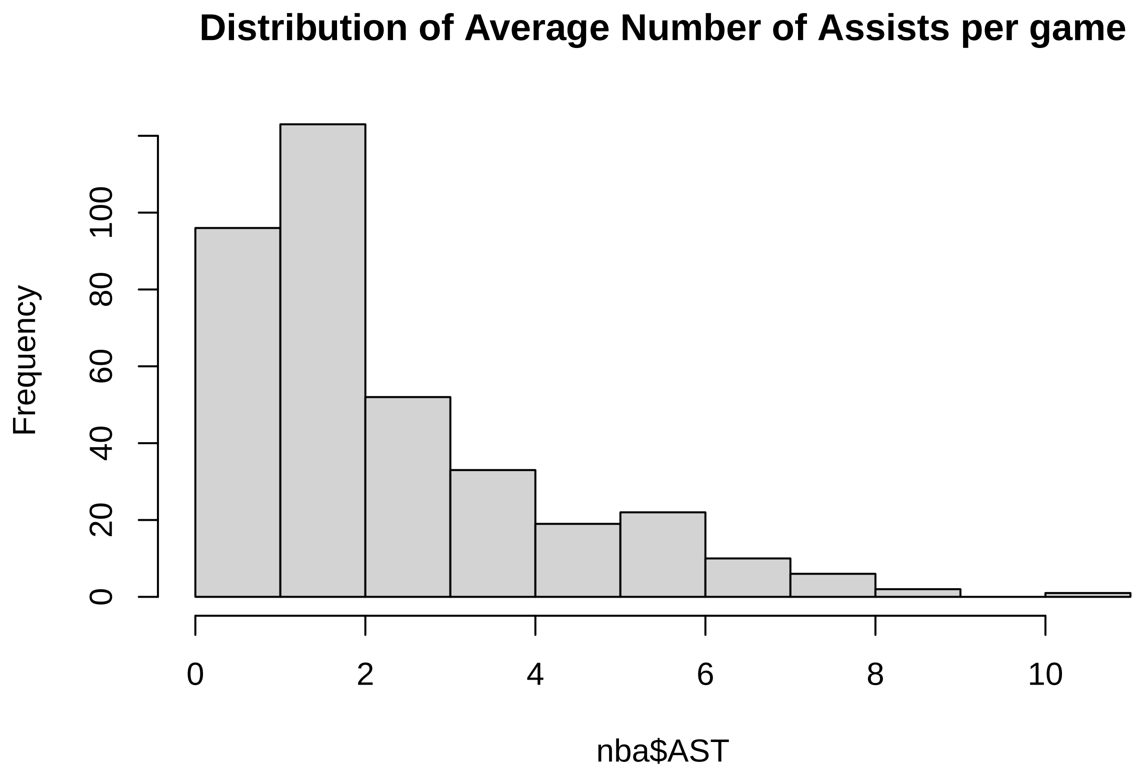
```
hist(nba$Salary, main = "Distribution of Log transformed salary in 2022 season") # this variable is hig
```



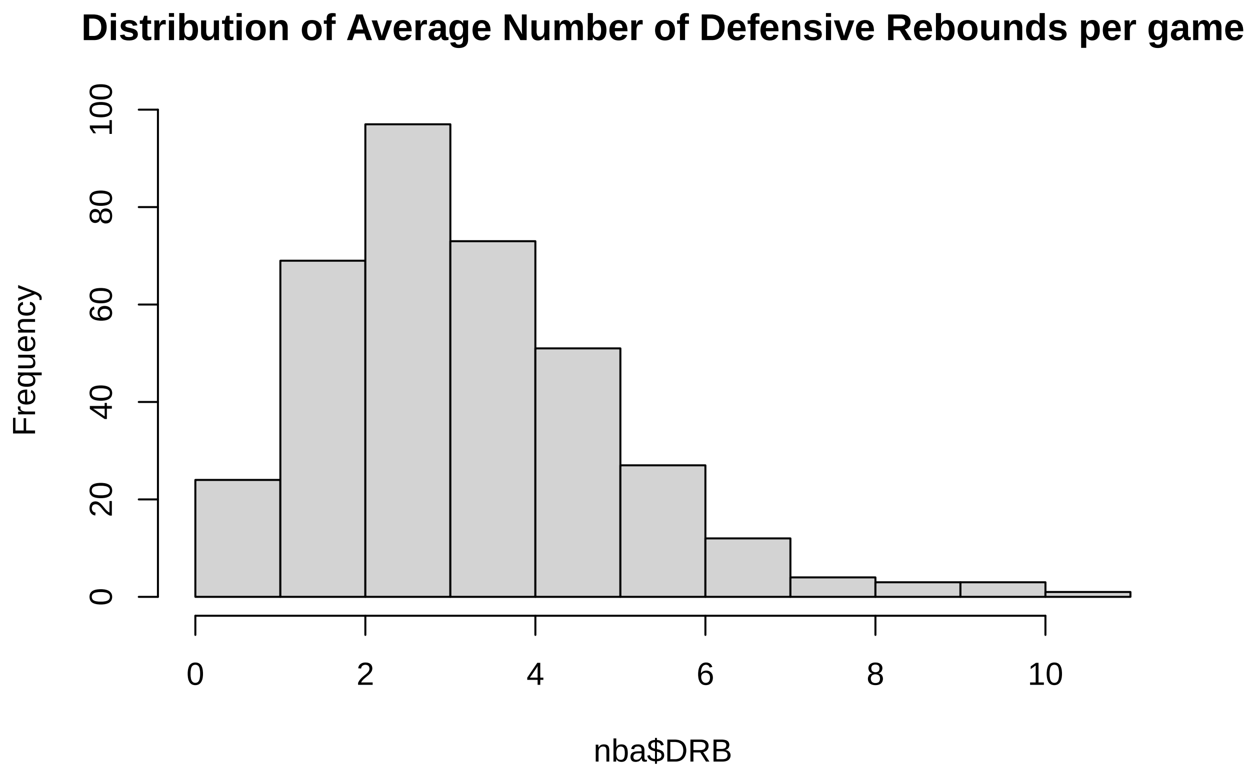

```
hist(nba$PTS, main = "Distribution of Average Number of Points per game")
```



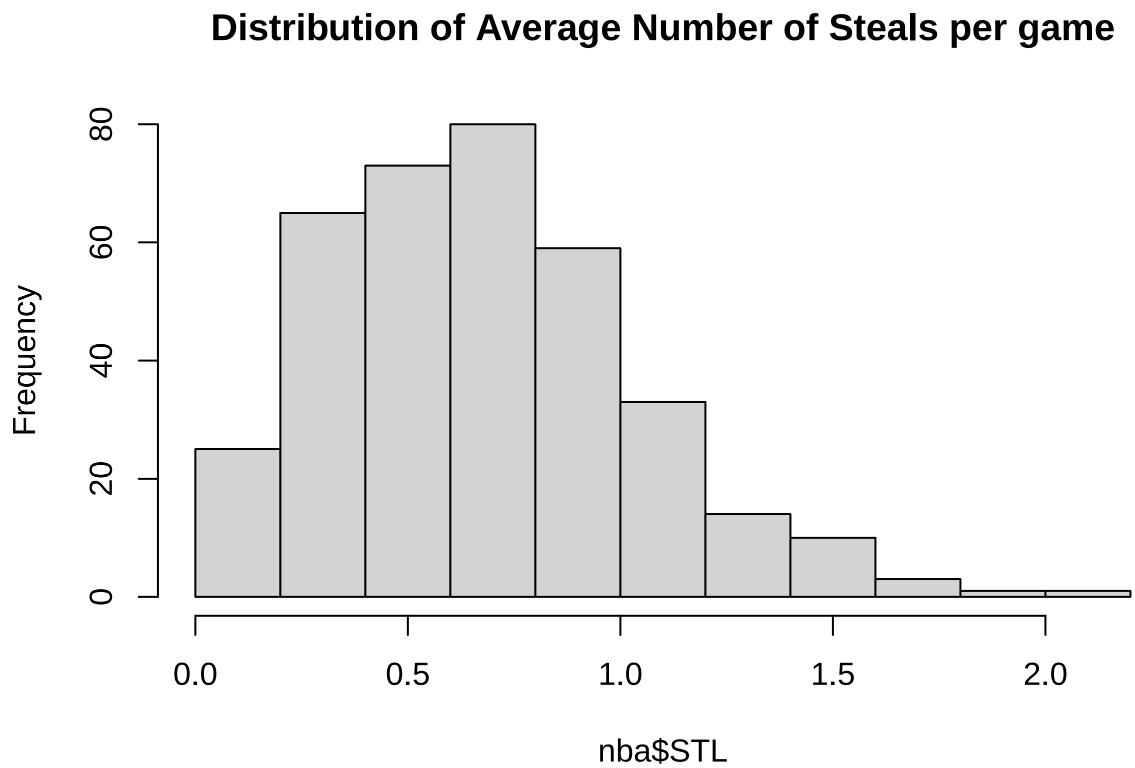

```
hist(nba$AST, main = "Distribution of Average Number of Assists per game")
```




```
hist(nba$DRB, main = "Distribution of Average Number of Defensive Rebounds per game")
```




```
hist(nba$STL, main = "Distribution of Average Number of Steals per game") # all other variables seem to
```



1 - specify prior distributions for your parameters

```
set.seed(976) # set a seed for reproducability
# initial values are randomly chosen between 1 and 8 and are right now for 3 chains -  I also included 
init2_1<- matrix(runif(4, min=1, max=8))
init2_2 <- matrix(runif(4, min=1, max=8))
init2_3 <- matrix(runif(4, min=1, max=8))
init2 <- rbind(matrix(init2_1,1,4),matrix(init2_2,1,4),matrix(init2_3,1,4))

init4_1 <- matrix(runif(6, min=1, max=8))
init4_2 <- matrix(runif(6, min=1, max=8))
init4_3 <- matrix(runif(6, min=1, max=8))
init4 <- rbind(matrix(init4_1,1,6),matrix(init4_2,1,6),matrix(init4_3,1,6))

# uninformative priors for normal distribution - For now I set them to be uninformative as I don't have
sigma02 <- c(1000,1000,1000); mu02 <- c(0,0,0)
sigma04 <- c(1000,1000,1000,1000,1000); mu04 <- c(0,0,0,0,0)

# uninformative priors for inverse gamma distribution, these to values are specified before and don't n
alpha0 <- 0.001; beta0 <- 0.001

# specifying the data sets including the predictors of interest - As I am interested in the effect of d
x2 <- cbind(nba$DRB,nba$AST)
x2.1 <- cbind(nba$PTS,nba$AST)
```

```
x2.2 <- cbind(nba$DRB,nba$STL)
x2.3 <- cbind(nba$PTS,nba$DRB)
x4 <- cbind(nba$PTS,nba$AST,nba$DRB,nba$STL)

y <- log(nba$Salary)
```

2./3. Gibbs sampler & Metropolis-Hastings step

```
# For full functions see Full_code.Rmd or the seperate functions
source("Functions/bayesian_regression.R")
```

4. - Assess convergence of the model

```
# For full functions see Full_code.Rmd or the seperate functions
source("Functions/Convergence.R")
```

5. - Check a model assumption with PPP

```
# For full functions see Full_code.Rmd or the seperate functions
source("Functions/Posterior_predictive_p_value.R")
```

6. - Obtain parameter estimates, credible intervals

```
# the output includes the parameter estimates for each sampled parameter as well as the credible interv
twopred <- bayesian_reg(y, x2, init2, sigma02, mu02, 10000, 40000, 3, 976)
```

```
##             Mean  S.E. MC-error   2.5% Median  97.5% Acceptance Burn-in
## Intercept 14.057 0.092  0.00031 13.876 14.057 14.236       1.00   30000
## Beta 1     0.245 0.026  0.00009  0.194  0.245  0.296       0.86   30000
## Beta 2     0.269 0.025  0.00008  0.221  0.269  0.317       1.00   30000
## Variance   0.622 0.046  0.00015  0.538  0.620  0.720       1.00   30000
##           Iterations
## Intercept      90000
## Beta 1         90000
## Beta 2         90000
## Variance       90000
```

```
twopred_0 <- bayesian_reg(y, x2.1, init2, sigma02, mu02, 10000, 40000, 3, 976)
```

```
##             Mean  S.E. MC-error   2.5% Median  97.5% Acceptance Burn-in
## Intercept 14.126 0.077  0.00026 13.974 14.127 14.276      1.000   30000
## Beta 1     0.105 0.009  0.00003  0.088  0.105  0.122      0.581   30000
## Beta 2     0.109 0.030  0.00010  0.051  0.109  0.167      1.000   30000
## Variance   0.557 0.042  0.00014  0.481  0.555  0.644      1.000   30000
```

```
##           Iterations
## Intercept      90000
## Beta 1         90000
## Beta 2         90000
## Variance       90000
```

```
twopred_D <- bayesian_reg(y, x2.2, init2, sigma02, mu02, 10000, 40000, 3, 976)
```

```
##             Mean  S.E. MC-error   2.5% Median  97.5% Acceptance Burn-in
## Intercept 13.805 0.111  0.00037 13.585 13.805 14.021      1.000   30000
## Beta 1     0.264 0.027  0.00009  0.210  0.264  0.317      0.867   30000
## Beta 2     1.119 0.129  0.00043  0.867  1.119  1.373      1.000   30000
## Variance   0.688 0.051  0.00017  0.595  0.686  0.796      1.000   30000
##           Iterations
## Intercept      90000
## Beta 1         90000
## Beta 2         90000
## Variance       90000
```

```
twopred_OD <- bayesian_reg(y, x2.3, init2, sigma02, mu02, 10000, 40000, 3, 976)
```

```
##             Mean  S.E. MC-error   2.5% Median  97.5% Acceptance Burn-in
## Intercept 13.949 0.087  0.00029 13.776 13.949 14.119      1.000   30000
## Beta 1     0.106 0.008  0.00003  0.091  0.106  0.121      0.576   30000
## Beta 2     0.128 0.028  0.00009  0.073  0.128  0.182      1.000   30000
## Variance   0.545 0.041  0.00014  0.471  0.543  0.630      1.000   30000
##           Iterations
## Intercept      90000
## Beta 1         90000
## Beta 2         90000
## Variance       90000
```

```
fourpred <- bayesian_reg(y, x4, init4, sigma04, mu04, 10000, 40000, 3, 976)
```

```
##             Mean  S.E. MC-error   2.5% Median  97.5% Acceptance Burn-in
## Intercept 13.802 0.098  0.00033 13.612 13.802 13.993      1.000   30000
## Beta 1     0.079 0.010  0.00003  0.060  0.079  0.098      0.566   30000
## Beta 2     0.076 0.033  0.00011  0.011  0.076  0.141      1.000   30000
## Beta 3     0.127 0.027  0.00009  0.074  0.127  0.181      1.000   30000
## Beta 4     0.352 0.144  0.00048  0.071  0.351  0.635      1.000   30000
## Variance   0.515 0.039  0.00013  0.445  0.513  0.596      1.000   30000
##           Iterations
## Intercept      90000
## Beta 1         90000
## Beta 2         90000
## Beta 3         90000
## Beta 4         90000
## Variance       90000
```

7. - a.) Compare multiple model by means of DIC

```
# Calculating the DIC for all models of relevance based on the output of the bayesian_reg function
twopred$DIC
```

```
## [1] 863.4717
```

```
twopred_O$DIC
```

```
## [1] 822.7534
```

```
twopred_D$DIC
```

```
## [1] 899.8959
```

```
twopred_OD$DIC
```

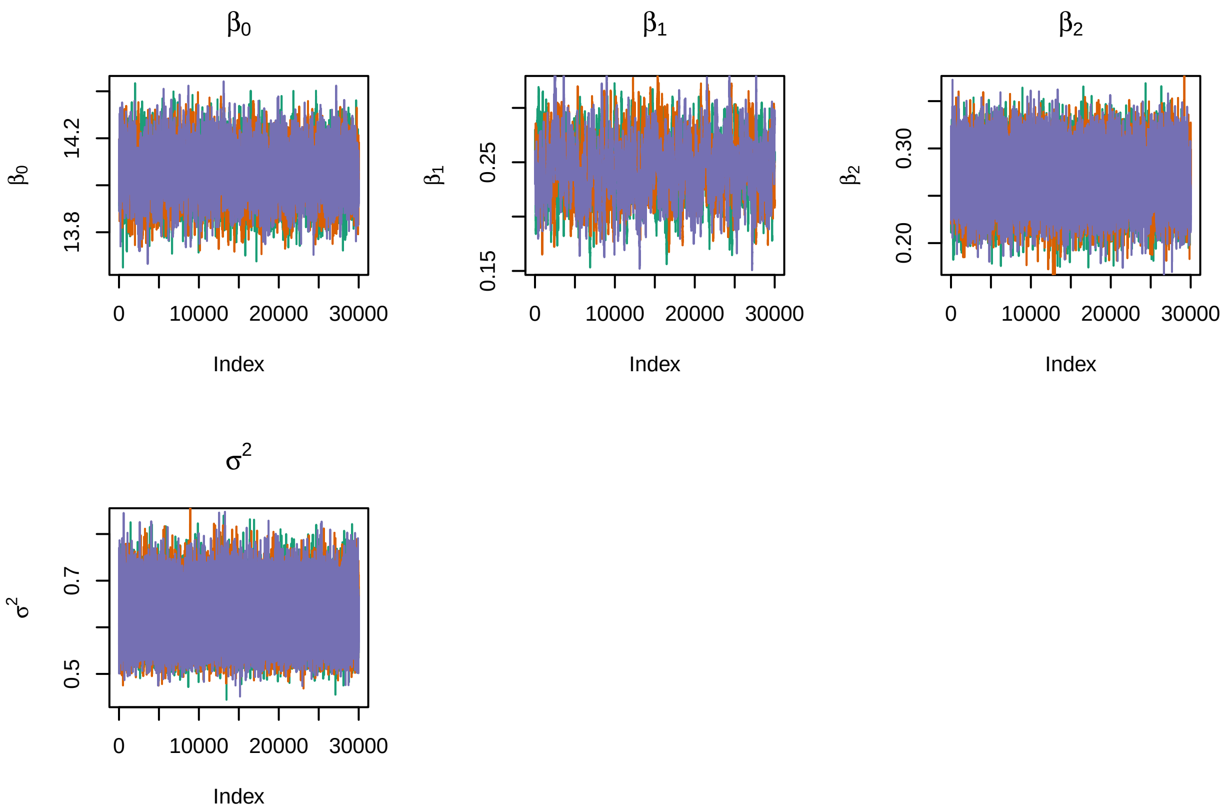
```
## [1] 815.1085
```

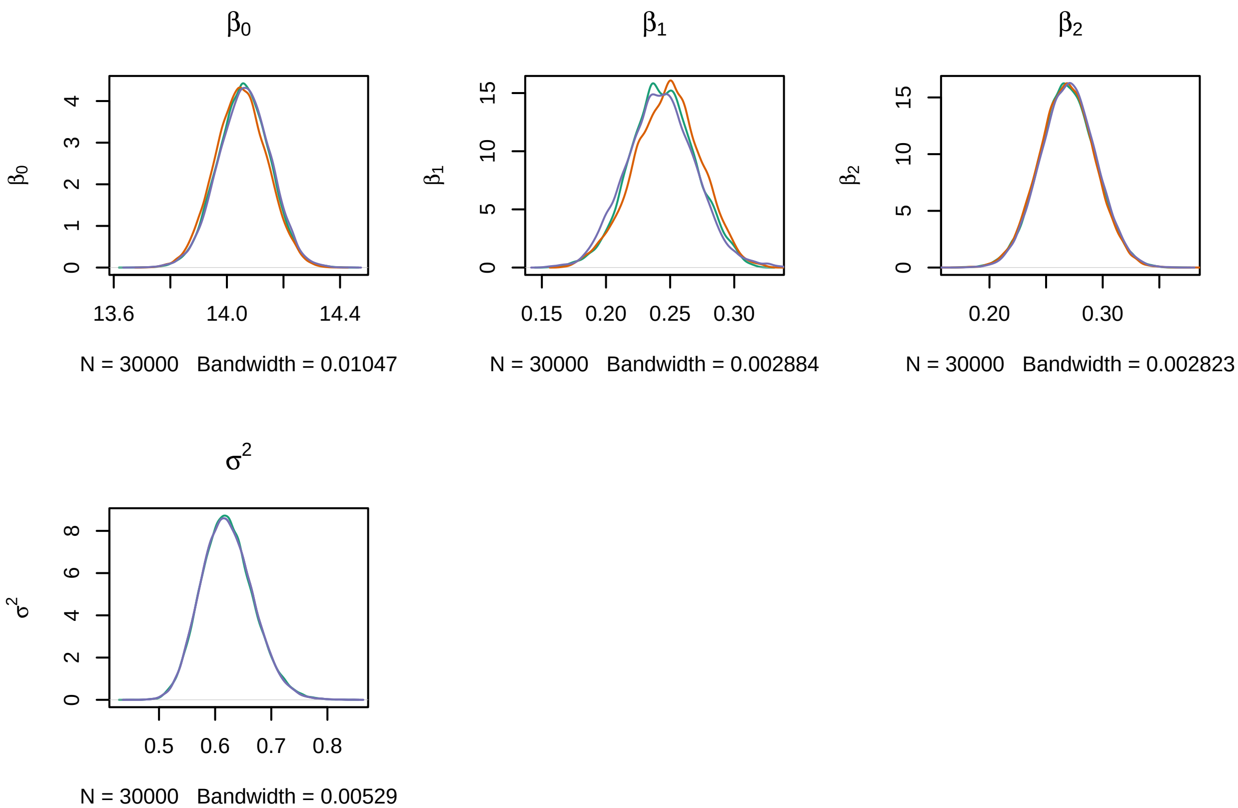
```
fourpred$DIC
```

```
## [1] 796.4155
```

7- b.) Convergence

```
convergencecheck(twopred$sampled_values_chains,50,976)
```

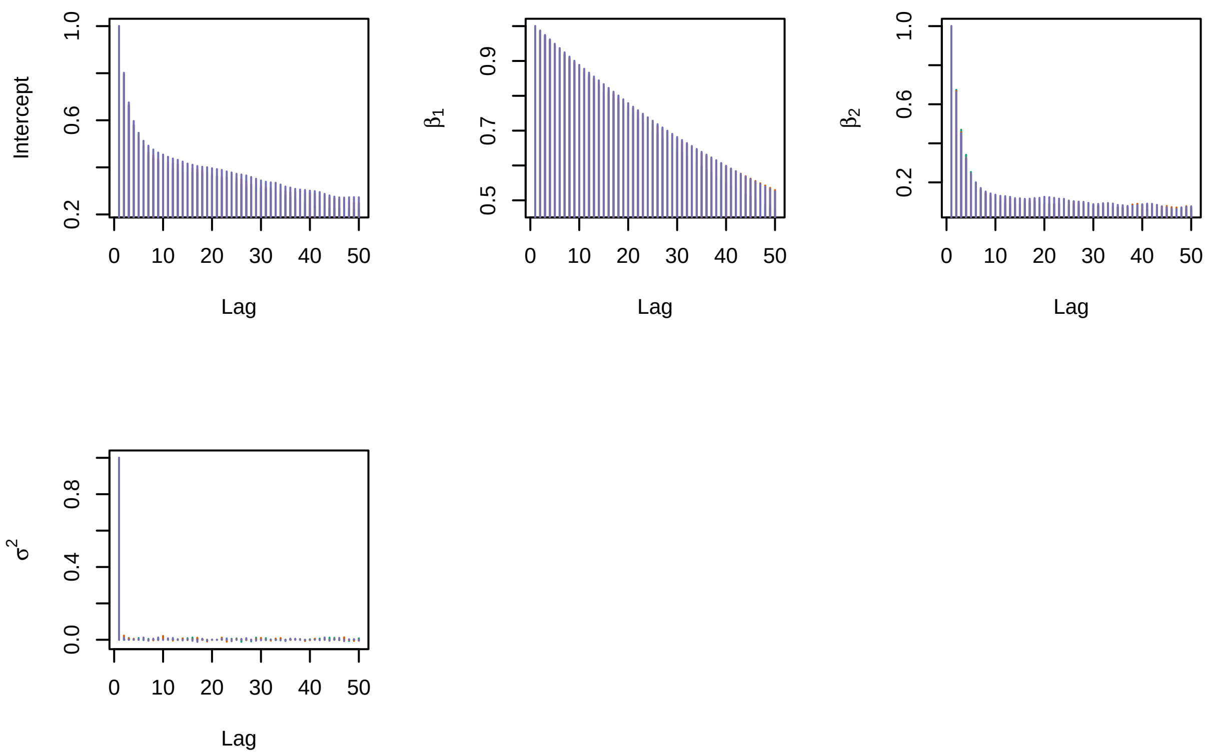

β0
β0
14.2
13.8
0
10000
20000
30000
Index
β1
β1
0.25
0.15
0
10000
20000
30000
Index
β2
β2
0.30
0.20
0
10000
20000
30000
Index
σ2
σ2
0.7
0.5
0
10000
20000
30000
Index

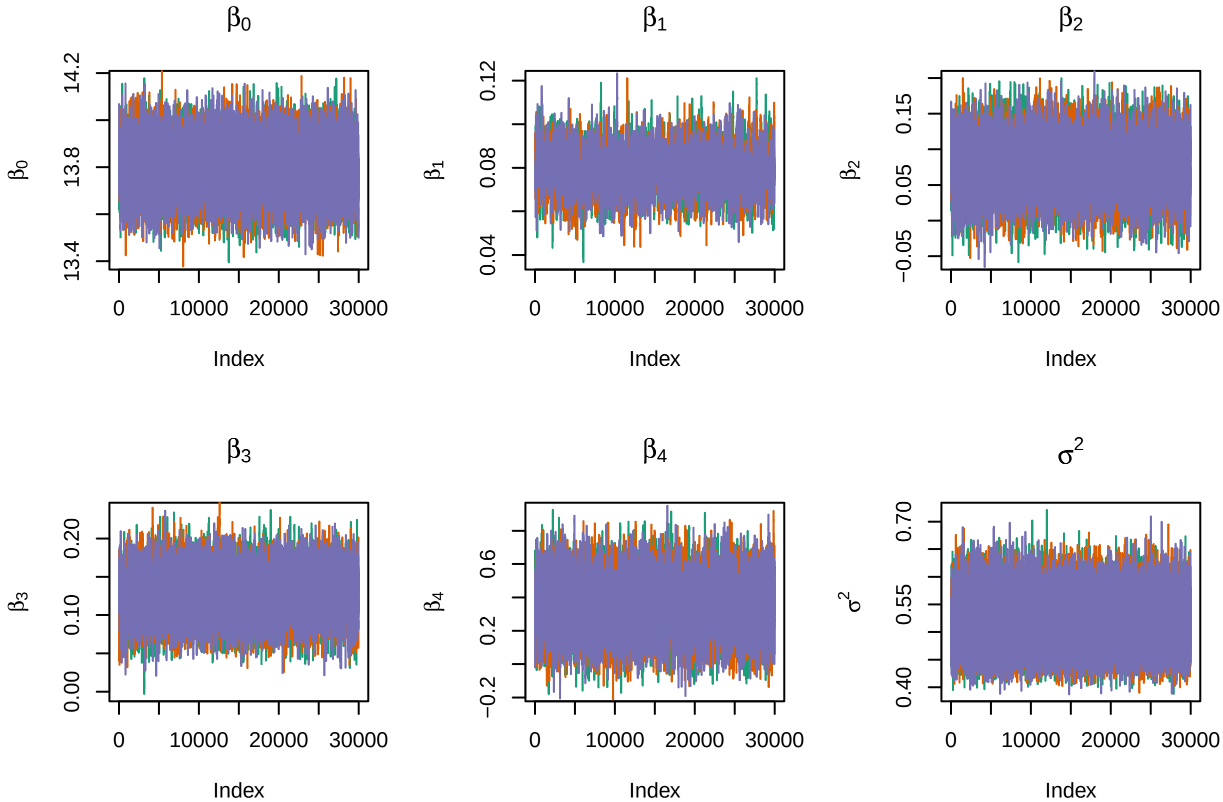


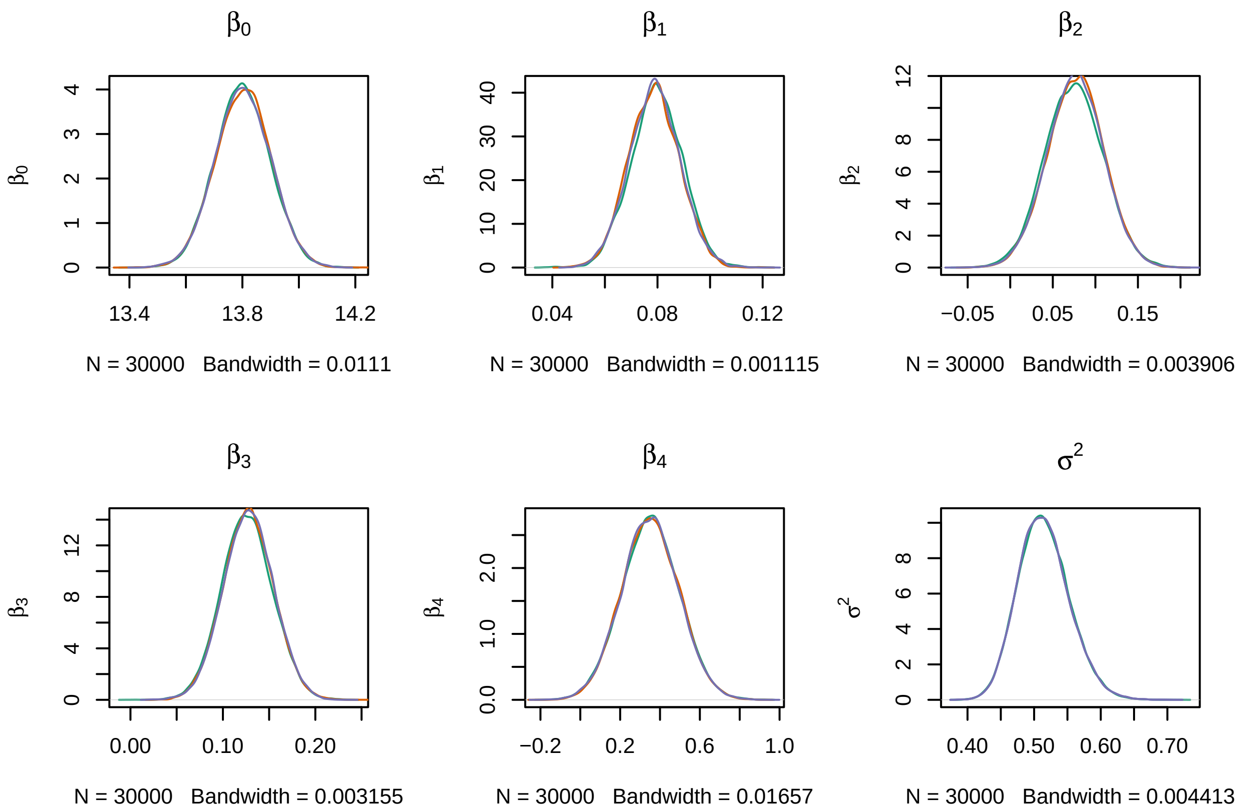

```
## $traceplots
## NULL
##
## $densityplots
## NULL
##
## $autocorrelation
## NULL
##
## $acceptance_chains
## $acceptance_chains[[1]]
##       [,1]
## [1,] 1.000
## [2,] 0.862
## [3,] 1.000
## [4,] 1.000
##
## $acceptance_chains[[2]]
##       [,1]
## [1,] 1.000
## [2,] 0.862
## [3,] 1.000
## [4,] 1.000
##
## $acceptance_chains[[3]]
##       [,1]
```

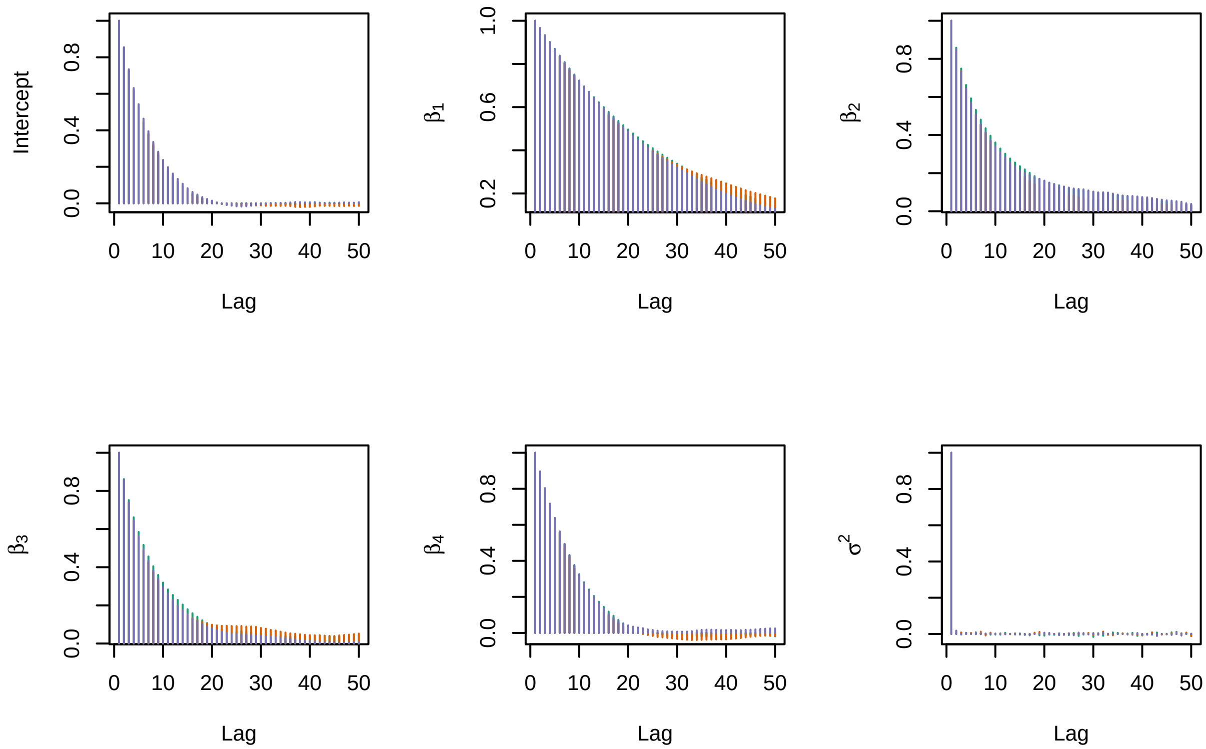
```
## [1,] 1.000
## [2,] 0.857
## [3,] 1.000
## [4,] 1.000
```

```
convergencecheck(fourpred$sampled_values_chains,50,976)
```




β0
β1
β2
β3
β4
σ2
Index
13.4
13.8
14.2
0.04
0.08
0.12
−0.05
0.05
0.15
0.00
0.10
0.20
−0.2
0.2
0.6
0.40
0.55
0.70
0
10000
20000
30000


β_0
β_1
β_2
β_3
β_4
σ^2
N = 30000 Bandwidth = 0.0111
N = 30000 Bandwidth = 0.001115
N = 30000 Bandwidth = 0.003906
N = 30000 Bandwidth = 0.003155
N = 30000 Bandwidth = 0.01657
N = 30000 Bandwidth = 0.004413




```
## $traceplots
## NULL
##
## $densityplots
## NULL
##
## $autocorrelation
## NULL
##
## $acceptance_chains
## $acceptance_chains[[1]]
##       [,1]
## [1,] 1.000
## [2,] 0.566
## [3,] 1.000
## [4,] 1.000
## [5,] 1.000
## [6,] 1.000
##
## $acceptance_chains[[2]]
##       [,1]
## [1,] 1.000
## [2,] 0.564
## [3,] 1.000
## [4,] 1.000
## [5,] 1.000
```

```
## [6,] 1.000
##
## $acceptance_chains[[3]]
##       [,1]
## [1,] 1.000
## [2,] 0.568
## [3,] 1.000
## [4,] 1.000
## [5,] 1.000
## [6,] 1.000
```

7. - c.) Bayes factor

```
# Models to be tested
OLS4 <- lm(log(Salary) ~ PTS + AST + DRB + STL, data = nba)
OLS2 <- lm(log(Salary) ~ AST + DRB, data = nba)

set.seed(976)
# Using the bain package to obtain the bayes factor
# Bayes factor 1
bf1 <- bain(OLS4, hypothesis = "PTS > 0 & DRB > 0 & AST > 0 & STL > 0", standardize = TRUE); print(bf1)
```

```
## Bayesian informative hypothesis testing for an object of class lm (continuous predictors):
##
##    Fit   Com   BF.u    BF.c      PMPa  PMPb  PMPc
## H1 0.982 0.002 638.074 36368.059 1.000 0.998 1.000
## Hu                                     0.002
## Hc 0.018 0.998 0.018                         0.000
##
## Hypotheses:
##   H1: PTS>0&DRB>0&AST>0&STL>0
##
## Note: BF.u denotes the Bayes factor of the hypothesis at hand versus the unconstrained hypothesis Hu
```

```
# Bayes factor 2
bf2 <- bain(OLS4, hypothesis = "PTS > DRB > AST> STL", standardize = TRUE); print(bf2)
```

```
## Bayesian informative hypothesis testing for an object of class lm (continuous predictors):
##
##    Fit   Com   BF.u   BF.c   PMPa  PMPb  PMPc
## H1 0.430 0.031 13.956 23.718 1.000 0.933 0.960
## Hu                                  0.067
## Hc 0.570 0.969 0.588                      0.040
##
## Hypotheses:
##   H1: PTS>DRB>AST>STL
##
## Note: BF.u denotes the Bayes factor of the hypothesis at hand versus the unconstrained hypothesis Hu
```

```r
bf3 <- bain(OLS2, hypothesis = "AST> DRB", standardize = TRUE); print(bf3)
```

```
## Bayesian informative hypothesis testing for an object of class lm (continuous predictors):
##
##    Fit   Com   BF.u  BF.c  PMPa  PMPb  PMPc
## H1 0.815 0.500 1.630 4.398 1.000 0.620 0.815
## Hu                               0.380
## Hc 0.185 0.500 0.370                   0.185
##
## Hypotheses:
##   H1: AST>DRB
##
## Note: BF.u denotes the Bayes factor of the hypothesis at hand versus the unconstrained hypothesis Hu
```

```r
bf1
```

```
## Bayesian informative hypothesis testing for an object of class lm (continuous predictors):
##
##    Fit   Com   BF.u    BF.c      PMPa  PMPb  PMPc
## H1 0.982 0.002 638.074 36368.059 1.000 0.998 1.000
## Hu                                     0.002
## Hc 0.018 0.998 0.018                         0.000
##
## Hypotheses:
##   H1: PTS>0&DRB>0&AST>0&STL>0
##
## Note: BF.u denotes the Bayes factor of the hypothesis at hand versus the unconstrained hypothesis Hu
```

```r
bf2
```

```
## Bayesian informative hypothesis testing for an object of class lm (continuous predictors):
##
##    Fit   Com   BF.u   BF.c   PMPa  PMPb  PMPc
## H1 0.430 0.031 13.956 23.718 1.000 0.933 0.960
## Hu                                 0.067
## Hc 0.570 0.969 0.588                     0.040
##
## Hypotheses:
##   H1: PTS>DRB>AST>STL
##
## Note: BF.u denotes the Bayes factor of the hypothesis at hand versus the unconstrained hypothesis Hu
```

```r
bf3
```

```
## Bayesian informative hypothesis testing for an object of class lm (continuous predictors):
##
##    Fit   Com   BF.u  BF.c  PMPa  PMPb  PMPc
## H1 0.815 0.500 1.630 4.398 1.000 0.620 0.815
## Hu                               0.380
## Hc 0.185 0.500 0.370                   0.185
##
```

```
## Hypotheses:
##   H1: AST>DRB
##
## Note: BF.u denotes the Bayes factor of the hypothesis at hand versus the unconstrained hypothesis Hu
```

7. - d.) PPP

```
ppp(fourpred$sampled_values,y,x4,3000,976)
```

```
## $Homoscedactisty
## [1] 0.572
##
## $Linearity
## [1] 0.518
```

```
ppp(twopred$sampled_values,y,x2,3000,976)
```

```
## $Homoscedactisty
## [1] 0.497
##
## $Linearity
## [1] 0.508
```