Lorenzo Ricciardulli and Ilaria Crippa

2024-07-10

Contents

1 THE DATASET

The data consists of the grades assigned to 10000 students and other information regarding their lifestyle and previous academic results. More specifically, the variables included in the dataset are:

- Hours Studied: The number of hours spent studying on average per day by each student.
- Previous Scores: The scores obtained by students in previous tests.
- Extracurricular Activities: Whether the student participates in extracurricular activities (Yes or No).
- Sleep Hours: The average number of hours of sleep the student had per day.
- Sample Question Papers Practiced: The number of sample question papers the student practiced.
- **Target Variable**:

- **Performance Index**: A measure of the overall performance of each student. The performance index represents the student's academic performance and has been rounded to the nearest integer. The index ranges from 10 to 100, with higher values indicating better performance.

The objectives of this research is to make predictions on the future academic performance of high school students, by regressing their performance index on the other available variables.

2 DATA VISUALIZATION AND CLEANING

2.1 Data loading and cleaning (dummy variable)

We load the dataset and convert the categorical variable "Extracurricular.Activities" into a dummy variable, that takes the value 1 when the student is involved in other activities outside of the school setting and 0 otherwise.

We check the structure of the data.

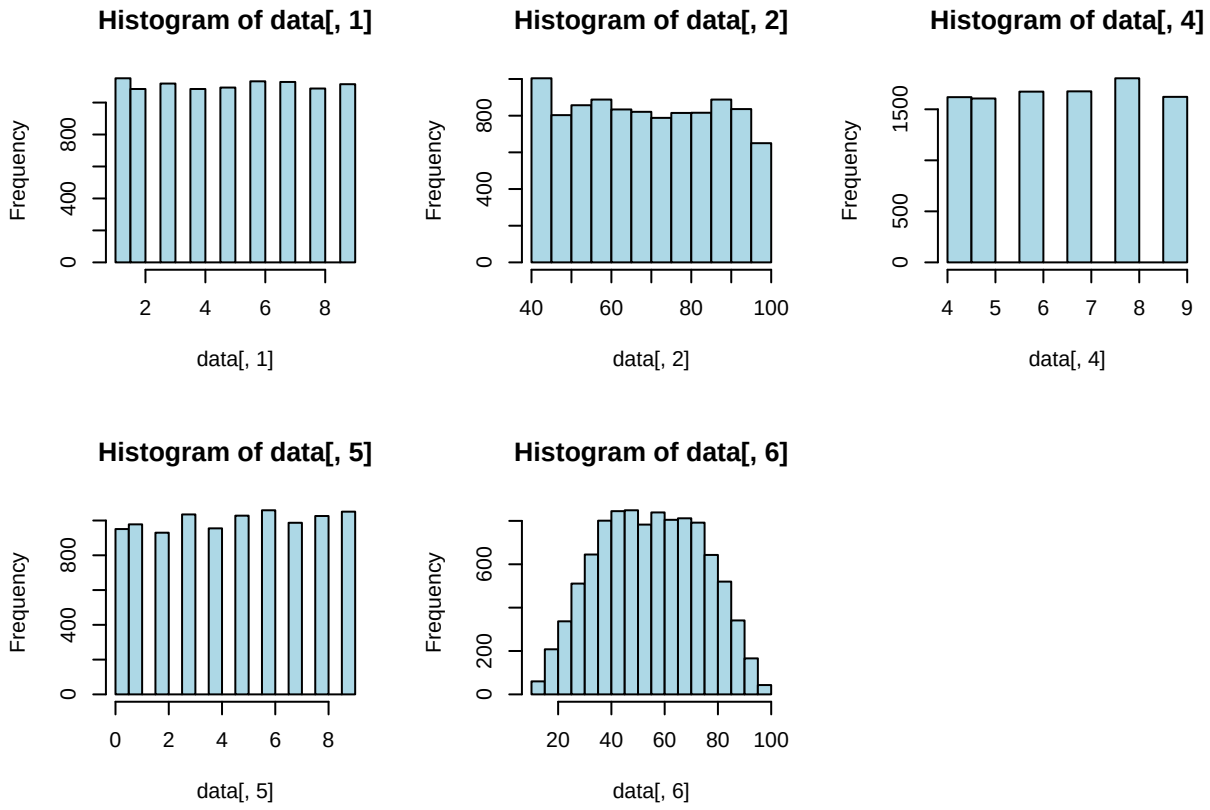
```
data = read.csv("Student_Performance.csv")
data$Extracurricular.Activities = ifelse(data$Extracurricular.Activities == "Yes", 1, 0)
summary(data)
```

```
##  Hours.Studied   Previous.Scores Extracurricular.Activities  Sleep.Hours
##  Min.   :1.000   Min.   :40.00   Min.   :0.0000             Min.   :4.000
##  1st Qu.:3.000   1st Qu.:54.00   1st Qu.:0.0000             1st Qu.:5.000
##  Median :5.000   Median :69.00   Median :0.0000             Median :7.000
##  Mean   :4.993   Mean   :69.45   Mean   :0.4948             Mean   :6.531
##  3rd Qu.:7.000   3rd Qu.:85.00   3rd Qu.:1.0000             3rd Qu.:8.000
##  Max.   :9.000   Max.   :99.00   Max.   :1.0000             Max.   :9.000
##  Sample.Question.Papers.Practiced Performance.Index
##  Min.   :0.000                    Min.   : 10.00
##  1st Qu.:2.000                    1st Qu.: 40.00
##  Median :5.000                    Median : 55.00
##  Mean   :4.583                    Mean   : 55.22
##  3rd Qu.:7.000                    3rd Qu.: 71.00
##  Max.   :9.000                    Max.   :100.00
```

2.2 Data visualization (histograms and boxplots)

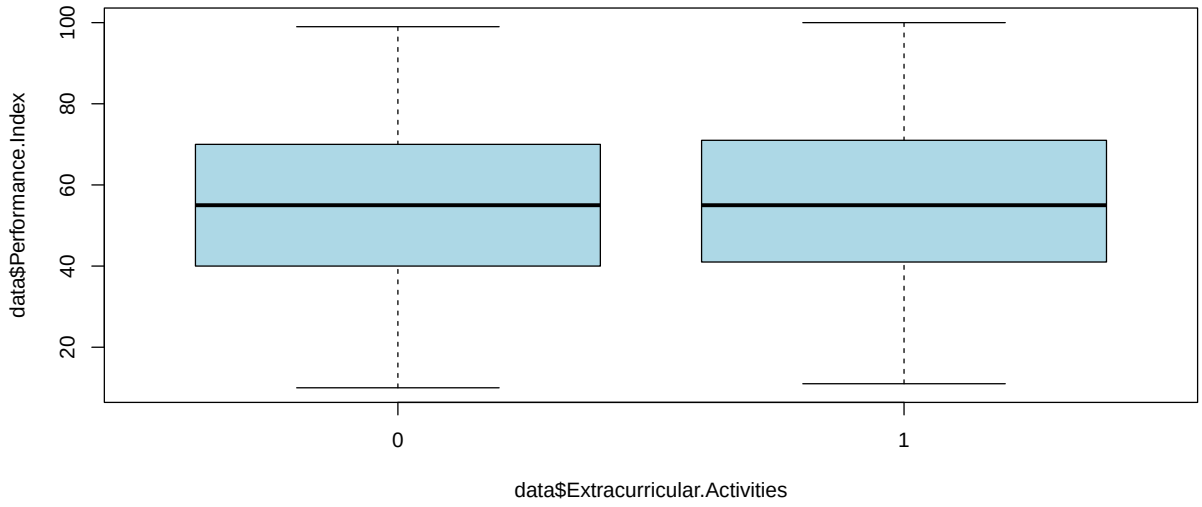
We plot the distributions of the variables.

```
par(mfrow = c(2,3))
hist(data[,1], col='lightblue')
hist(data[,2], col='lightblue')
hist(data[,4], col='lightblue')
hist(data[,5], col='lightblue')
hist(data[,6], col='lightblue')
```



The following boxplot shows the relationship between the response variable and the categorical variable "Extracurricular Activities".

```
boxplot(data$Performance.Index ~ data$Extracurricular.Activities, col='lightblue')
```



3 REGRESSION

We consider all the covariates present in the dataset to explain the response and assume a Bayesian normal linear regression model with conjugate priors. The prior distributions are: a Gamma distribution for the precision of the dependent variable, , with hyper-parameters a/2 and b/2; a multivariate normal for the conditional distribution of given , with hyper-parameters $_0$ for the mean and 1/ * $\mathbf{V}^{-1}$ for the variance covariance matrix. We assume non-informative priors.

The update of the hyper-parameters is computed as follows:

$a_n = a + n$, $b_n = b + \mathbf{y}^T \mathbf{y} + {}_0{}^T\mathbf{V}_0$ - $\mathbf{tilde}^T$ $(\mathbf{V} + \mathbf{X}^T\mathbf{X})$ **tilde** for .

tilde $= (\mathbf{V} + \mathbf{X}^T\mathbf{X})^{-1}$ $(\mathbf{V}_0 + \mathbf{X}^T\mathbf{y})$ and **Vtilde** $= \mathbf{V} + \mathbf{X}^T\mathbf{X}$

3.1 Samples of the joint posterior distribution of beta and tau

3.1.1 Function for the Monte Carlo sampling scheme

In order to obtain a sequence of draws from the posterior joint distribution of (, ^{2}), it is necessary to implement a Monte Carlo sampling scheme, where for each iteration we sample a value of from its posterior distribution and then plug it in the full conditional distribution of .

The output of the following R function is a list with two elements: an (S = numbers of iterations, p) matrix, in which each row is a draw from the posterior of = (beta_1, ..., beta_p) and an (S,1) matrix, in which each row is a draw from the posterior of ^{2} = $^{-1}$.

```
## Posterior inference for the linear model (conjugate priors)

posterior_linear_conjugate = function(y, X, beta.0, V, a, b, S){

  n = nrow(X)
  p = ncol(X)

  ## Compute posterior hyperparameters

  V.n    = V + t(X)%*%X
  beta.n = solve(V + t(X)%*%X)%*%(V%*%beta.0 + t(X)%*%y)

  a.n = a + n
  b.n = b + t(y)%*%y + beta.0%*%V%*%beta.0 - t(beta.n)%*%(V + t(X)%*%X)%*%beta.n


  beta_post   = matrix(NA, S, p)
  sigma2_post = matrix(NA, S, 1)


  ## Monte Carlo

  for(s in 1:S){

    ## (1) Sample sigma2

    tau  = rgamma(1, a.n/2, b.n/2)
    sigma2 = 1/tau

    ## (2) Sample beta conditionally on tau
```

```
    beta = c(rmvnorm(1, beta.n, solve(V.n)/tau))

    ## Store sampled draws

    beta_post[s,]   = beta
    sigma2_post[s,] = sigma2

  }

  return(posterior = list(beta_post   = beta_post,
                          sigma2_post = sigma2_post))


}
```

3.1.2 Data preparation for function

As stated above, we set the values of the hyper-parameters for non-informative priors. The number of iterations is set to 5000.

```
y = data$Performance.Index ## response variable
data_X = data[,-6]
X = model.matrix(y ~., data_X)
p = ncol(X)
beta.0 = rep(0,p)
V      = diag(0.1, p)
a      = 0.01
b      = 0.01
S = 5000
```

3.1.3 Sampling from the joint posterior distribution

The function defined above is now applied to our data. The empirical means and quantiles of each parameter from the joint posterior distribution are provided below.

```
set.seed(1234)
out = posterior_linear_conjugate(y = y, X = X, beta.0, V, a, b, S = S)
str(out)
```

```
## List of 2
##  $ beta_post  : num [1:5000, 1:6] -34 -34.1 -34.1 -34.1 -34.3 ...
##  $ sigma2_post: num [1:5000, 1] 4.24 4.11 4.26 4.19 4.1 ...
```

```
beta_post = out$beta_post
sigma2_post = out$sigma2_post

beta_post_mcmc = as.mcmc(beta_post)
sigma2_post_mcmc = as.mcmc(sigma2_post)

summary(beta_post_mcmc)
```

```
##
## Iterations = 1:5000
## 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
## [1,] -34.0597 0.125798 1.779e-03      1.779e-03
## [2,]   2.8525 0.007848 1.110e-04      1.060e-04
## [3,]   1.0184 0.001163 1.644e-05      1.644e-05
## [4,]   0.6124 0.040286 5.697e-04      5.697e-04
## [5,]   0.4797 0.011788 1.667e-04      1.633e-04
## [6,]   0.1935 0.007084 1.002e-04      1.002e-04
##
## 2. Quantiles for each variable:
##
##          2.5%      25%      50%      75%    97.5%
## var1 -34.3011 -34.1459 -34.0605 -33.9751 -33.8106
## var2   2.8372   2.8471   2.8526   2.8578   2.8677
## var3   1.0161   1.0176   1.0183   1.0192   1.0206
## var4   0.5337   0.5856   0.6120   0.6395   0.6904
## var5   0.4569   0.4717   0.4795   0.4876   0.5032
## var6   0.1794   0.1887   0.1935   0.1982   0.2074
```

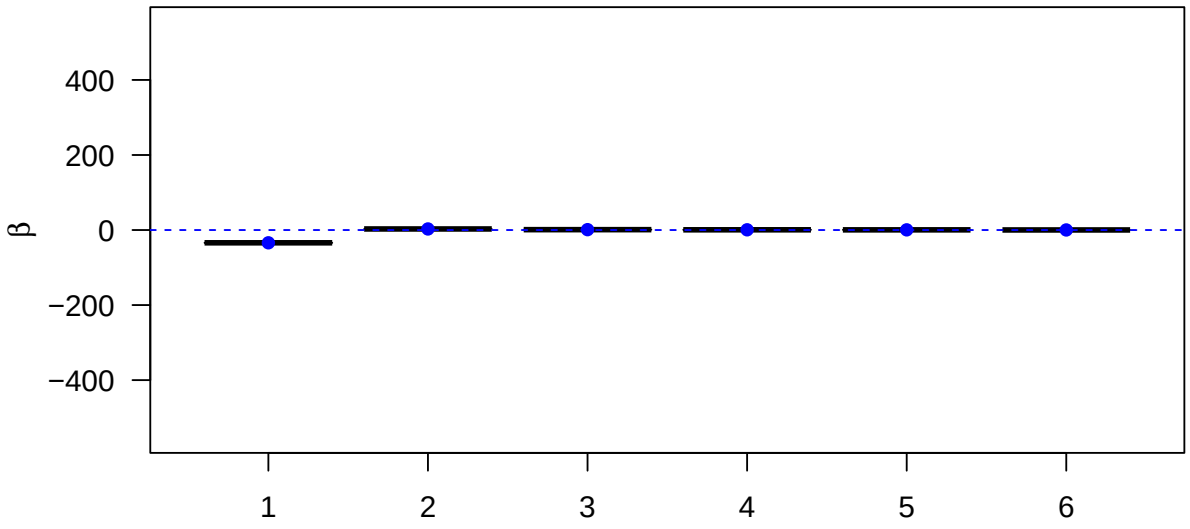
```
summary(sigma2_post_mcmc)
```

```
##
## Iterations = 1:5000
## 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
##      4.1648854      0.0600638      0.0008494      0.0008872
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75% 97.5%
## 4.049 4.124 4.165 4.206 4.281
```

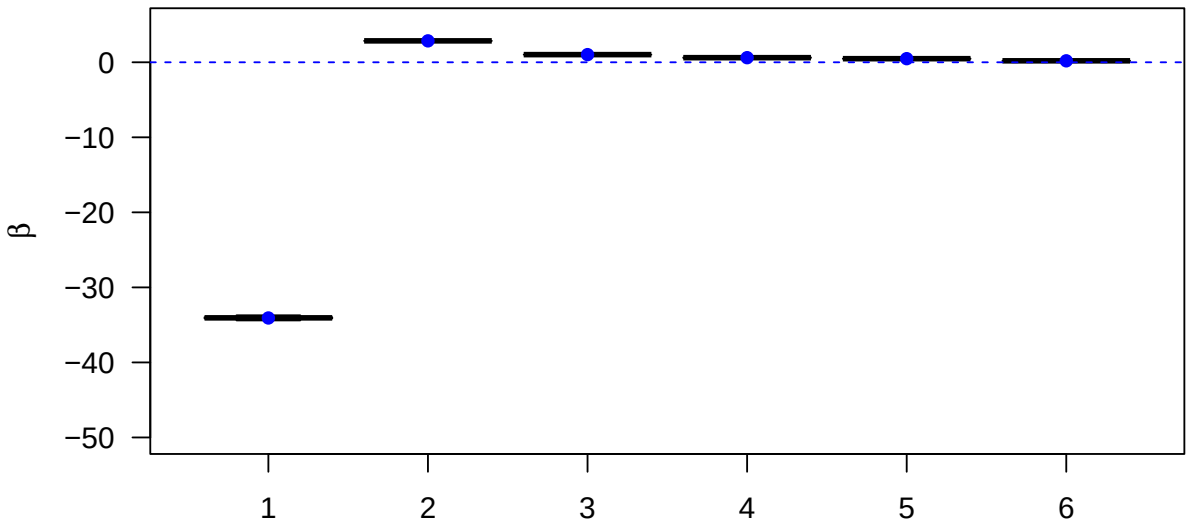
3.2 Boxplots of the MCMC samples from the posterior of beta

The boxplots of the posterior are plotted below. The MLE of is shown as the blue dots on the boxplots. As expected, since non-informative priors have been adopted, the posterior expectation of **tilde** tends to the OLS estimate: ***tilde*** $\rightarrow$ ***hat***.

```
par(mar = c(3,5,1,1))

boxplot(out$beta_post, outline = F, ylim = c(-550,550), ylab = expression(beta), las = 1)
abline(h = 0, col = "blue", lty = "dashed")
out_lm = summary(lm(y ~ X - 1))
points(out_lm$coefficients[,1], col = "blue", pch = 16)
```




```
par(mar = c(3,5,1,1))

boxplot(out$beta_post, outline = F, ylim = c(-50,5), ylab = expression(beta), las = 1)
abline(h = 0, col = "blue", lty = "dashed")
out_lm = summary(lm(y ~ X - 1))
points(out_lm$coefficients[,1], col = "blue", pch = 16)
```

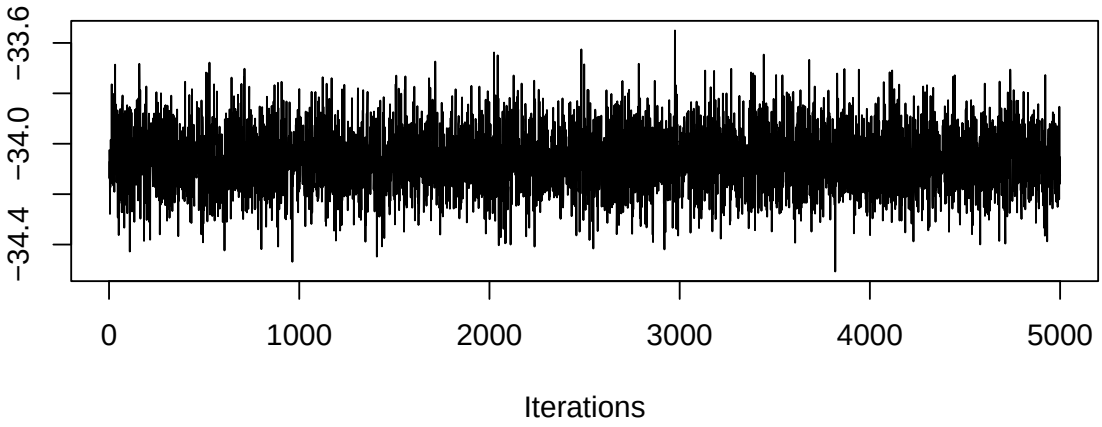


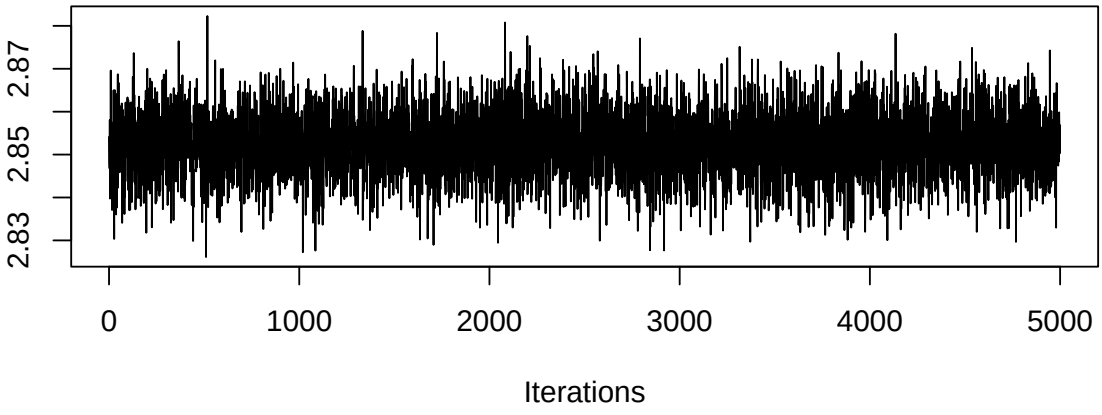
4 DIAGNOSTICS

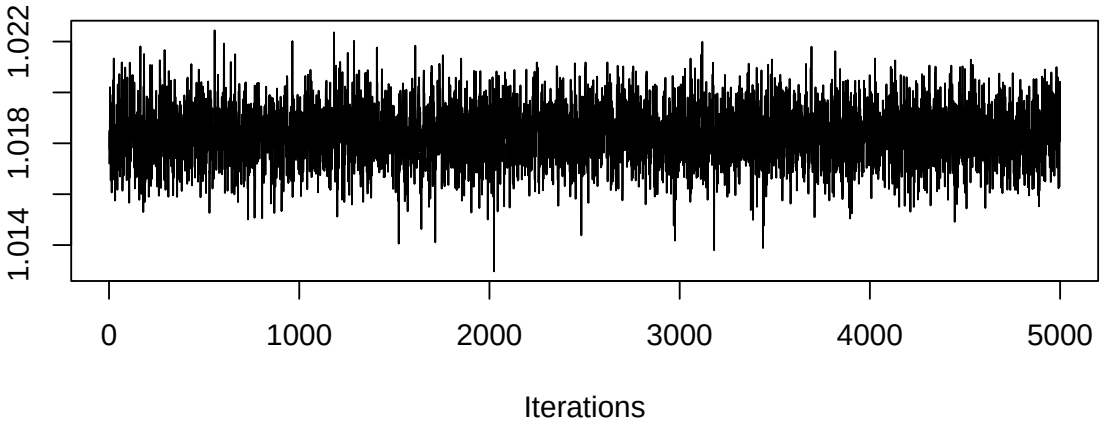
4.1 Trace plots

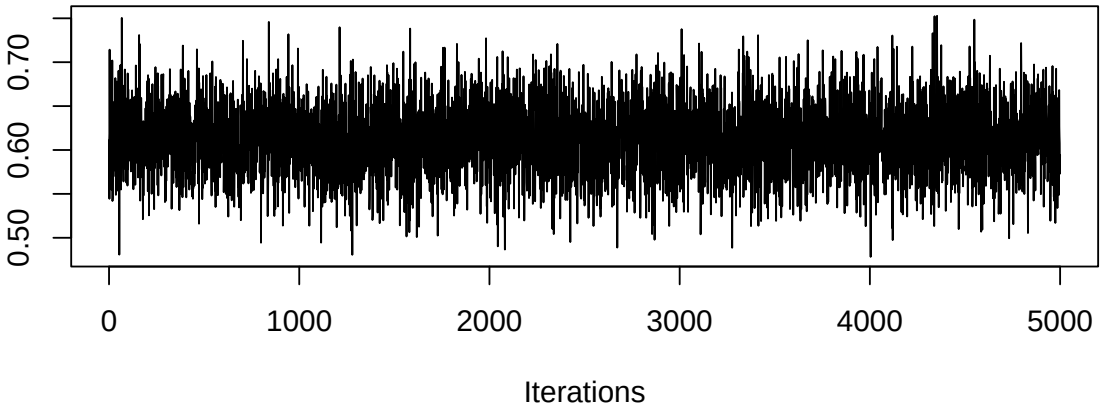
The convergence of the MCMC outputs is verified with a visual inspection. The plots show that all the chains converge. The chains are concentrated within a region of high posterior probability, centred around a mode of p(| , **y**) for the coefficients and of p(^{2} | **y**) for the variance. Additionally, no trends are observed.

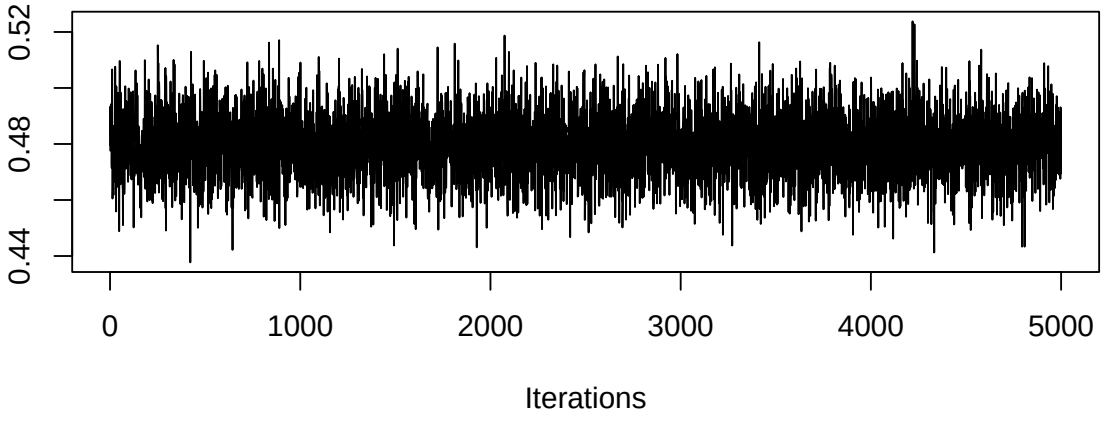
```r
beta_post_mcmc   = as.mcmc(out$beta_post)
sigma2_post_mcmc = as.mcmc(out$sigma2_post)

traceplot(beta_post_mcmc)
```

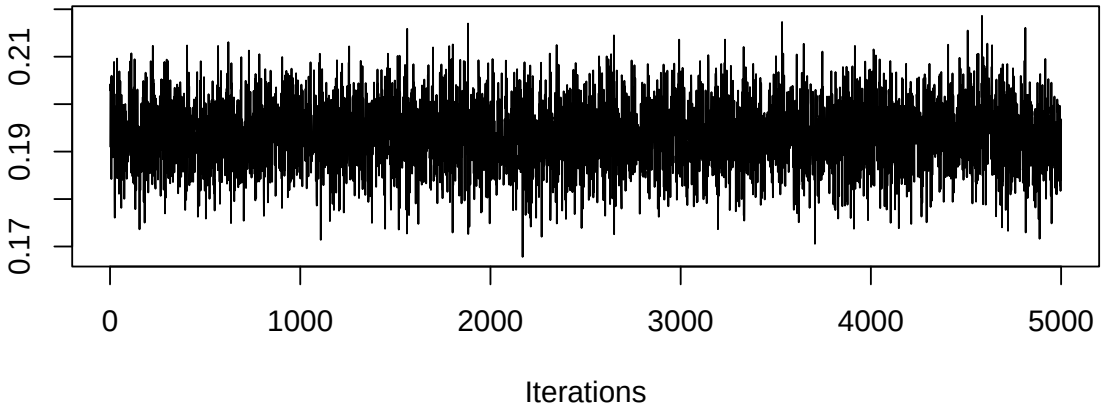




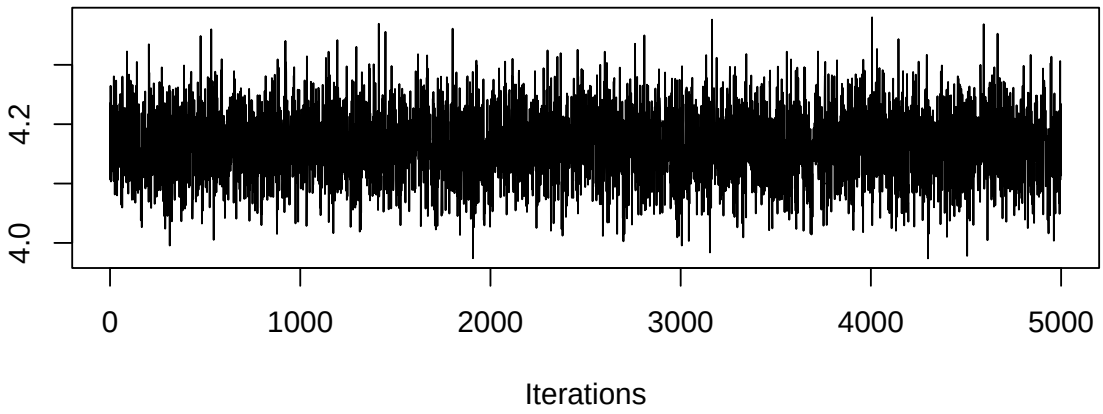

1.014
1.018
1.022
0
1000
2000
3000
4000
5000
Iterations


0.50
0.60
0.70
0
1000
2000
3000
4000
5000
Iterations





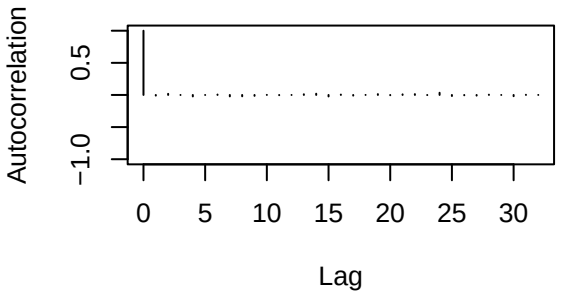

```
traceplot(sigma2_post_mcmc)
```

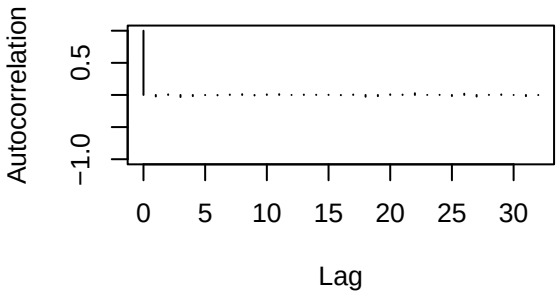


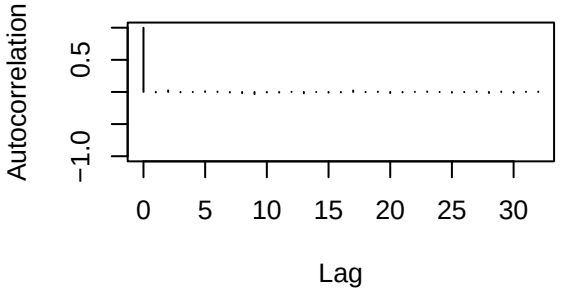
4.2 Autocorrelation tests

No dependencies can be observed in the chains, when inspecting the autocorrelation plots. Autocorrelation represents the degree of similarity between a given portion of a chain and a lagged version of itself over successive intervals.

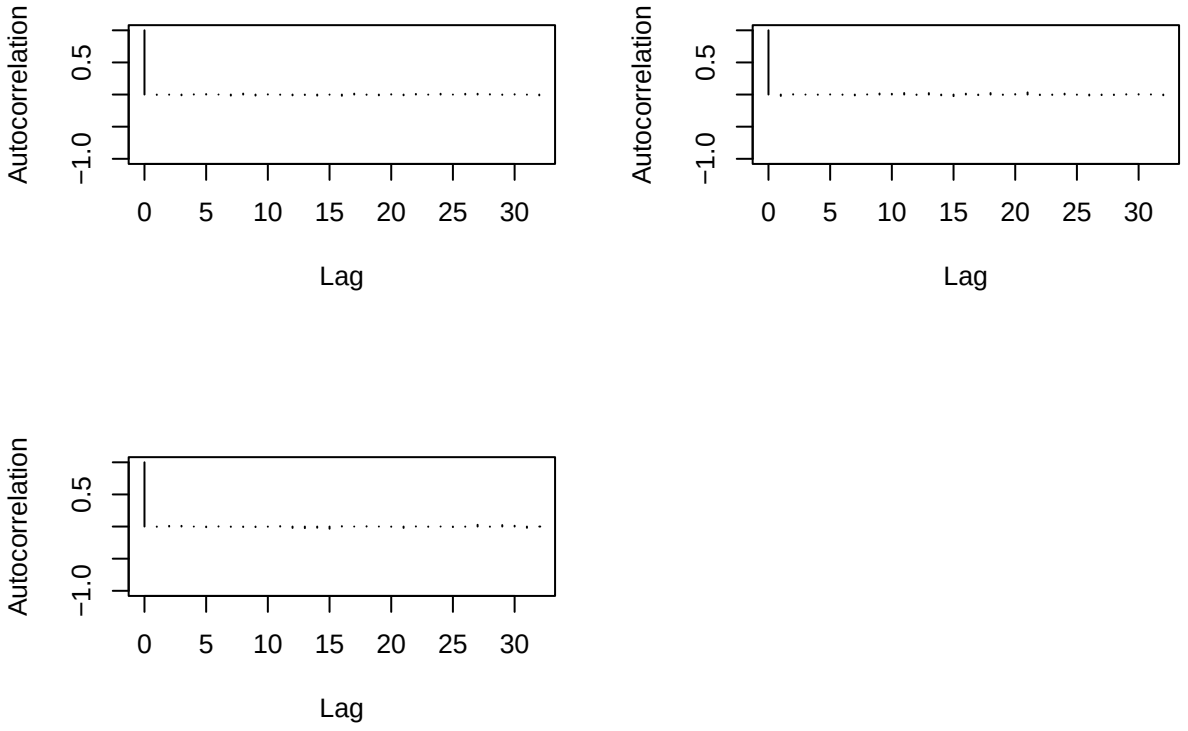
```
coda::autocorr.plot(beta_post_mcmc[, 1:3])
```



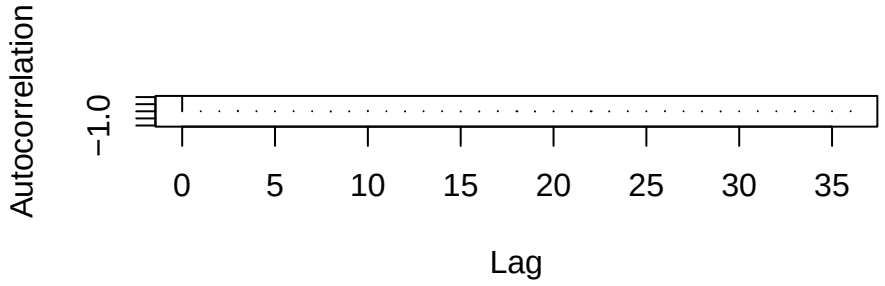




```
coda::autocorr.plot(beta_post_mcmc[, 4:6])
```




```
coda::autocorr.plot(sigma2_post_mcmc)
```



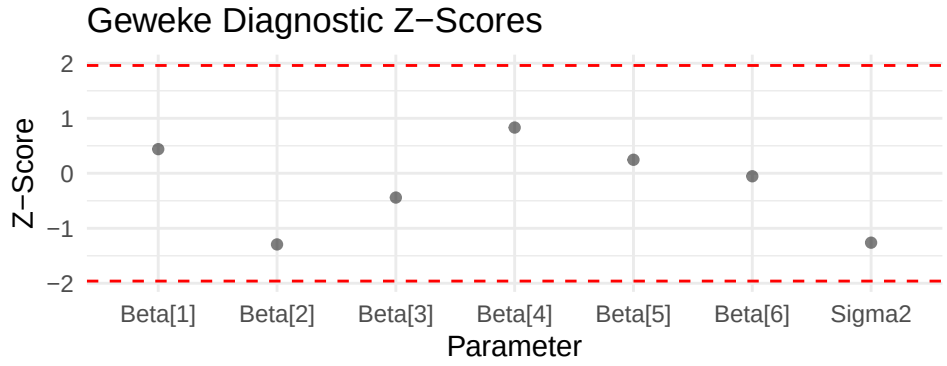
4.3 Geweke test

Another method to check for stationarity in the MCMC is the Geweke test, which verifies whether two "windows" of the chain (in this case the first 10% and last 50% of the total draws) have sample means that are almost equal. If the absolute value of the Z-statistic is large, the hypothesis of stationarity is rejected.

The plot below shows that each Z-score is within the acceptance regions.

```
geweke_beta <- apply(beta_post_mcmc, 2, function(x) geweke.diag(mcmc(x))$z)
geweke_sigma2 <- geweke.diag(mcmc(sigma2_post_mcmc))$z
```

```r
df_beta <- data.frame(Parameter = paste0("Beta[", 1:6, "]"), Z_Score = geweke_beta)
df_sigma2 <- data.frame(Parameter = "Sigma2", Z_Score = geweke_sigma2)

df_combined <- rbind(df_beta, df_sigma2)

ggplot(df_combined, aes(x = Parameter, y = Z_Score)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = c(-1.96, 1.96), linetype = "dashed", color = "red") +
  labs(title = "Geweke Diagnostic Z-Scores",
       x = "Parameter",
       y = "Z-Score") +
  theme_minimal()
```



5 MODEL / VARIABLE SELECTION

In order to perform a model selection, the marginal likelihood function is computed. The chosen prior belief on each model was specified as a uniform prior: $p(M_k) = 1/K$, where M_k refers to each model belonging to the model space $\mathbf{M} = \{M_1,..., M_K\}$ and $K = 2^5$.

5.1 Function for marginal likelihood

```r
marg_like_regr = function(y, X, beta.0, V, a, b){

  X = as.matrix(X)
  V = as.matrix(V)

  n = nrow(X)
  p = ncol(X)

  ## posterior hyperparameters

  V.n    = V + t(X)%*%X
  beta.n = solve(V + t(X)%*%X)%*%(V%*%beta.0 + t(X)%*%y)

  a.n = a + n
  b.n = b + t(y)%*%y + beta.0%*%V%*%beta.0 - t(beta.n)%*%(V + t(X)%*%X)%*%beta.n

  ## log-marginal likelihood
```

```
  m = -n/2*log(2*pi) + a/2*log(b/2) - a.n/2*log(b.n/2) +
        lgamma(a.n/2) - lgamma(a/2) +
          log(det(V))/2 - log(det(V.n))/2

  return(m)

}
```

5.2 Model selection with uniform prior

The result of the model selection shows that the model with the highest posterior probability is the one containing every covariate. Therefore, there is no possible improvement to be made on the previous regression.

```
##all the possible distinct models with 5 predictors

set = 2:p

1:length(set)
```

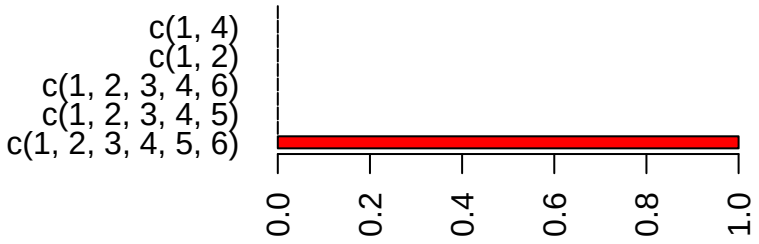
```
## [1] 1 2 3 4 5
```

```
comb = unlist(lapply(1:length(set), combinat::combn, x = set, simplify = FALSE), recursive = FALSE)

## We add the intercept and also include the model with the intercept only

all.comb = mapply(append, 1, comb, SIMPLIFY = FALSE)

all.models.index = append(1, all.comb)

## Compute marginal likelihoods

all.log.marg.like = c()

K = length(all.models.index)

beta.0 = rep(0, p)
V      = diag(0.01, p)
a      = 0.01
b      = 0.01

k = 1

for(k in 1:K){

  set.k = all.models.index[[k]]

  all.log.marg.like[k] = marg_like_regr(y, X[,set.k], beta.0[set.k], V[set.k,set.k], a, b)

}

## Obs: we assume the uniform prior on M_k

all.marg.like = exp(all.log.marg.like)

post.model.probs = all.marg.like/sum(all.marg.like)
```

```r
const = max(all.log.marg.like)
all.marg.like = exp(all.log.marg.like - const)
post.model.probs = all.marg.like / sum(all.marg.like)

names(post.model.probs) = all.models.index

head(sort(post.model.probs, decreasing = TRUE))
```

```
## c(1, 2, 3, 4, 5, 6)    c(1, 2, 3, 5, 6)    c(1, 2, 3, 4, 5)       c(1, 2, 3, 5)
##        1.000000e+00        1.555270e-46       6.682263e-153       1.532609e-197
##    c(1, 2, 3, 4, 6)                   1
##       2.226704e-319        0.000000e+00
```

```r
par(mar=c(5,12,1,1))
barplot(sort(post.model.probs, decreasing = TRUE)[1:10], horiz = TRUE, las = 2, xlim = c(0,1), col = ra
```

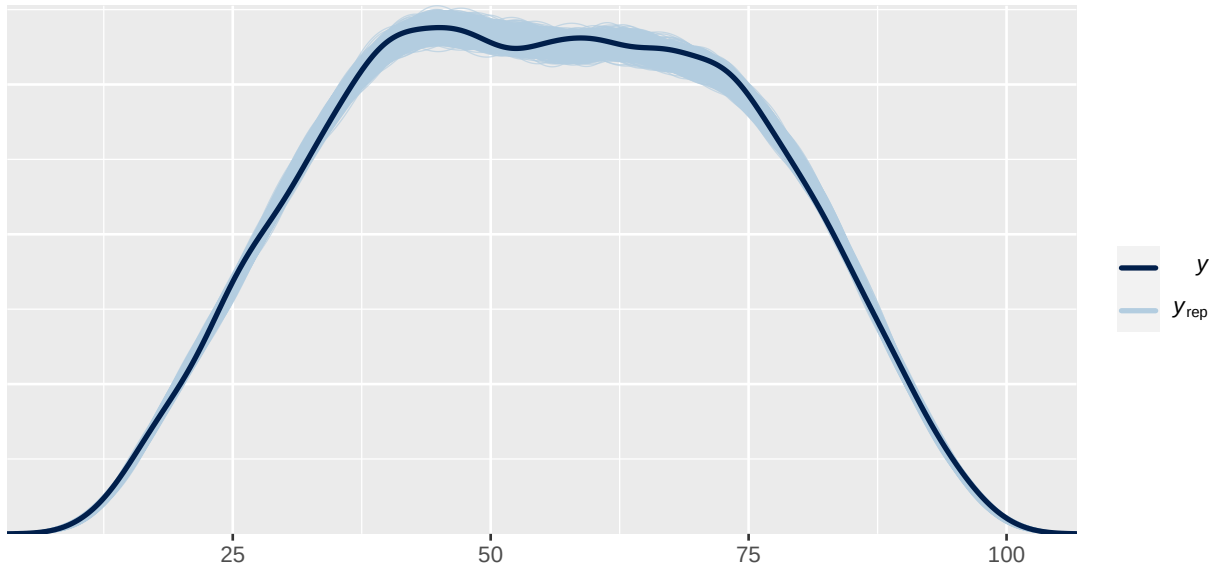


6 PREDICTIONS

In the following section, our posterior draws of and ^{2} were used to predict the dependent variable for every row in our dataset. The plot below compares the predicted values (in blue) with the observed values of the response (in black).

```r
post_beta_jags = as.matrix(beta_post_mcmc)
post_tau_jags = as.matrix(sigma2_post_mcmc)

n_samples <- nrow(beta_post_mcmc)

n_obs <- length(y)

posterior_predictive <- matrix(0, n_samples, n_obs)

for (i in 1:n_samples) {
  y_pred_mean <-   X %*% post_beta_jags[i, ]
  y_pred_sd <- sqrt(as.vector(post_tau_jags[i]))
  posterior_predictive[i, ] <- rnorm(n_obs, y_pred_mean, y_pred_sd)
}

y_matrix <- matrix(y, ncol = length(y))

color_scheme_set("blue")
```

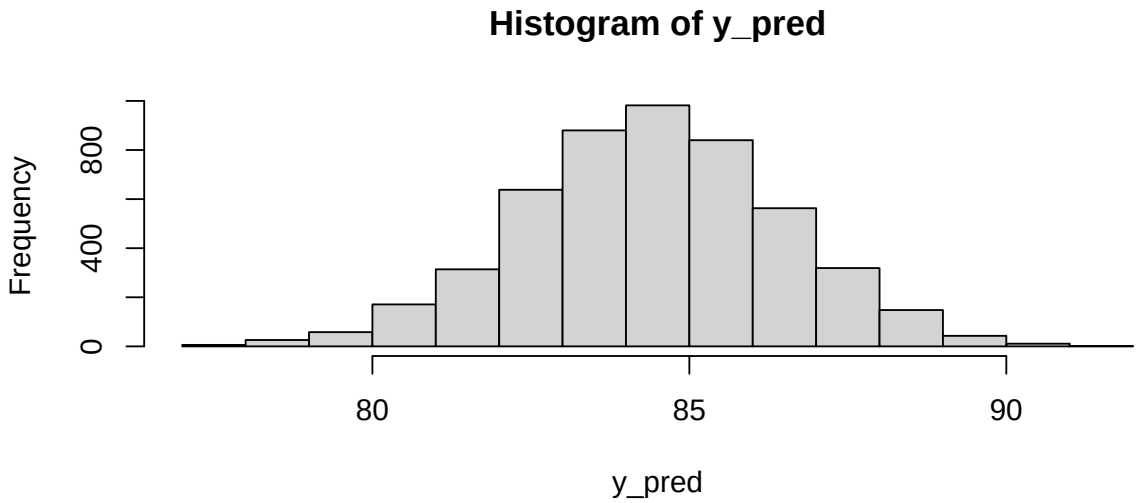
```r
ppc_dens_overlay(y, posterior_predictive)
```



6.1 Predictions for new observation

A prediction is made with new values of the covariates: 8 hours of study per day on average, average previous score equal to 90, extracurricular activities, 7 hours of sleep on average and no sample questions practiced. The posterior expected value of the predicted performance index is 84.4.

```r
x_new = c(1,8,90,1,7,0)

mu = beta_post_mcmc%*%x_new # S values of mu

y_pred = rnorm(S, mu, sqrt(sigma2_post_mcmc))

hist(y_pred)
```



```r
mean(y_pred)
```

```
## [1] 84.40487
```