Bayesian Linear Regression

Ahmed Ali, Alan n. Inglis, Estevão Prado, Bruna Wundervald

Abstract

Bayesian methods are an alternative to standard frequentist methods and as a result have gained popularity. This report will display some of the fundamental ideas in Bayesian modelling and will present both the theory behind Bayesian statistics and some practical examples of Bayesian linear regression. Simulated data and real-world data were used to construct the models using both R code and Python.

1 Introduction

The aim of this report is to investigate and implement Bayesian linear regression models on different datasets. The aim of Bayesian Linear Regression is not to find the single 'best' value of the model parameters, but rather to determine the posterior distribution for the model parameters. This is achieved by using Bayes' theorem. To begin, we take a look at a few basic concepts regarding Bayesian linear regression.

1.1 Introduction to regression:

Regression analysis is a statistical method that allows you to examine the relationship between two or more variables of interest. The overall idea of regression is to examine two things[1]:

(i) Does a set of predictor variables do a good job in predicting an outcome (dependent) variable?

(ii) Which variables in particular are significant predictors of the outcome variable, and in what way do they impact the outcome variables? These regression estimates are used to explain the relationship between one dependant variable and one or more independent variables. A simple from of the linear regression equation, with one dependent variable Y, and one independent variable X, is defined by the formula:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

where;

Y = dependent variable,

β are the weights (or model parameters), where;

β_0 = intercept,

β_1 = slope coefficient (indicating the change of Y on avgerage when X increases by 1 unit),

X = independent variable,

ϵ = is an error term representing random sampling noise (or the effect of the variables not included in the model).

Classical linear regression can then be used to identify, based on the data, the best fitting linear relationship between the inputs and outputs

[1]Statistics Solutions. (2013). What is Linear Regression. http://www.statisticssolutions.com/what-is-linear-regression/

1.2 Bayesian Statistics:

Bayesian statistics is a mathematical method that applies probabilities to statistical problems. It provides people the tools to update their beliefs in the evidence of new data.

Bayes' theorem describes the conditional probability of an event based on data as well as prior information or beliefs about the event or conditions related to the event.

Bayes' theorem can be used to calculate the *posterior probability* (which is the revised probability of an event occurring after taking into consideration new information). The posterior probability is calulated by updating the *prior probability*. The proir probability is the probability of an outcome based on the current knowledge before an experiment.

Bayes' theorem is described by[2]:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

where;

P(A) is the probability of event A occurring, also known as the prior probability.

P(A|B) is the conditional probability of event A occurring, given that B is true. This is the posterior probability due to its variable dependency on B. This assumes that event A is not independent of B.

P(B|A) is the conditional probability of event B occurring given that A is true.

P(B) is the probability of event B occurring.

Bayesian methods can be a useful tool that helps researchers move beyond hunting for statistical significance and instead focus on other aspects of statistical models such as prediction, model fit, data visualization, and uncertainty.

1.3 Bayesian linear regression:

Bayesian linear regression allows a useful mechanism to deal with insufficient data, or poor distributed data. It allows you to put a prior on the coefficients and on the noise so that in the absence of data, the priors can take over. More importantly, you can ask of Bayesian linear regression which parts of it, that fits the data, is it confident about, and which parts are uncertain (perhaps based entirely on the priors). Specifically, you can ask of it[3]:

- What is the estimated linear relation, what is the confidence on that relation, and what is the full posterior distribution on that relation?
- What is the estimated noise and the posterior distribution on that noise?
- What is the estimated gradient and the posterior distribution on that gradient?

In the Bayesian viewpoint, we formulate linear regression using probability distributions. The response, y, is not estimated as a single value, but is assumed to be drawn from a probability distribution. The model for Bayesian Linear Regression, with the response sampled from a normal distribution is given by[4]:

$$y \sim N(\beta^T X, \sigma^2 I)$$

[2]Stuart, A.; Ord, K. (1994), Kendall's Advanced Theory of Statistics: Volume I—Distribution Theory

[3]Auton Technologies.(2018). Bayesian Linear Regression. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/learn-43/lib/idauction2/.g/web/glossary/bayesian.html

[4]Koehrsen W.(2018, April 14). Introduction to Bayesian Linear Regression. https://towardsdatascience.com/introduction-to-bayesian-linear-regression-e66e60791ea7

The output, y, is generated from a normal distribution characterised by a mean and variance. The mean for linear regression is the transpose of the weight matrix multiplied by the predictor matrix. The variance is the square of the standard deviation σ , multiplied by the identity matrix.

Not only is the response generated from a probability distribution, but the model parameters are assumed to come from a distribution as well. The posterior probability of the model parameters is conditional upon the training inputs and outputs[5]:

$$P(\beta|y,X) = \frac{P(y|\beta,X) * P(\beta|X)}{P(y|X)}$$

Here; P(β|y,X) is the posterior probability distribution of the model parameters given the inputs and outputs. This is equal to the likelihood of the data, P(y|β,X), multiplied by the prior probability of the parameters and divided by a normalisation constant:

$$Posterior = \frac{Liklihood * Prior}{Normalization}$$

Here, we can observe the two benefits of Bayesian Linear Regression:

(i) Priors: If we have knowledge (or a guess for what the model parameters should be), we can include them in our model. If we don't have any estimates ahead of time, we can use non-informative priors for the parameters, such as a normal distribution.

(ii) Posterior: The result of performing Bayesian Linear Regression is a distribution of possible model parameters based on the data and the prior. This allows us to quantify our uncertainty about the model. If we have fewer data points, the posterior distribution will be more spread out.

As the amount of data points increases, the likelihood washes out the prior, and in the case of infinite data, the outputs for the parameters converge to the values obtained from ordinary Least Squares (OLS)

1.4 Implementing Bayesian Linear Regression:

To approximate the posterior, we use sampling methods to draw samples from the posterior. The technique of drawing random samples from a distribution to approximate the distribution is one application of Monte Carlo methods.

The basic procedure for implementing Bayesian Linear Regression is:

(i) Specify priors for the model parameter.

(ii) Create a model mapping the training inputs to the training outputs.

(iii) Have a Markov Chain Monte Carlo (MCMC) algorithm draw samples from the posterior distributions for the parameters.

In the following sections, we introduce the mathematical theory behind Bayesian linear regression and some practical implementations of it.

[5]Koehrsen W.(2018, April 14). Introduction to Bayesian Linear Regression. https://towardsdatascience.com/introduction-to-bayesian-linear-regression-e66e60791ea7

2 Mathematical Theory

2.1 Introduction:

Before introducing Bayesian linear regression, in this Section we present the classical linear regression model and parameter estimation via maximum likelihood in order to show the different aspects between the classical and Bayesian approaches.

2.2 Linear regression:

Let $\{Y_i\}_{i=1}^n$ be independent and identically distributed random variables, $\{y_i\}_{i=1}^n$ their observed values and $\{x_{ij}\}_{j=1}^d$ the j-th explanatory variable for i. Consider $\mathbf{y} = (y_1, ..., y_n)^\top$ an $n \times 1$ column vector and $\mathbf{X} = (x_{i1}, ..., x_{id})_{i=1}^n$ an $n \times d$ design matrix containing all variables that may be associated to the response variable $\mathbf{y}$. The classical linear regression model may be written

$$\mathbf{y} = \mathbf{X}\beta + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathrm{N}_n(\mathbf{0}, \sigma^2\mathbf{I}),$$

$\boldsymbol{\beta} = (\beta_0, ..., \beta_{d-1})^\top$ is the $d \times 1$ vector of parameters and $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_n)^\top$ the $n \times 1$ vector of errors. In addition, the likelihood function of $\mathbf{y}$ given $\boldsymbol{\beta}$ and σ^2 is defined as follows

$$\prod_{i=1}^{n} f(Y_i = y_i|\mathbf{X}, \boldsymbol{\beta}, \sigma^2) = f_{\mathrm{y}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\Big\{ -\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\Big\}.$$

In order to make inference, the $\boldsymbol{\beta}$ estimates are obtained by maximizing the likelihood function or its log. For mathematical convenience, once the parameter values that maximize both functions are the same, it is commonly utilized the log-likelihood, which is given by

$$\ln f_{\mathrm{y}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \tag{1}$$

Considering that $\mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y} = (\mathbf{y}^\top\mathbf{X}\boldsymbol{\beta})^\top$ is a scalar, differentiating equation (1) with respect to $\boldsymbol{\beta}$ and solving for $\boldsymbol{\beta}$, we have

$$\begin{aligned}\frac{\partial \ln f_{\mathrm{y}}(\mathbf{y}|\mathbf{X}\boldsymbol{\beta}, \sigma^2)}{\partial\boldsymbol{\beta}} &= \frac{1}{2\sigma^2}2\mathbf{X}^\top\mathbf{y} - 2\mathbf{X}^\top\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{0}\\ \hat{\boldsymbol{\beta}} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y},\end{aligned}$$

the maximum likelihood (ML) estimator. Similarly, the ML estimator for σ^2 is given by

$$\begin{aligned}\frac{\partial \ln f_{\mathrm{y}}(\mathbf{y}|\mathbf{X}\boldsymbol{\beta}, \sigma^2)}{\partial\sigma^2} &= -\frac{n}{2\hat{\sigma}^2} + \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{2(\hat{\sigma}^2)^2} = 0,\\ &= \hat{\sigma}^2 n + (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0,\\ \hat{\sigma}^2 &= \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{n}.\end{aligned}$$

Also, there are some statistical properties such as consistency that are verified for $\boldsymbol{\beta}$. For instance, $\hat{\boldsymbol{\beta}}$ is an asymptotically consistent estimator for $\boldsymbol{\beta}$ if $P(|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}| > a) \to 0$ as n $\to \infty$ for every $a > 0$. For finite samples, it is consistent when $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$. That is,

$$\begin{aligned}\mathbb{E}[\hat{\boldsymbol{\beta}}] &= \mathbb{E}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}],\\ &= \mathbb{E}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon})],\\ &= \mathbb{E}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} + (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\epsilon}],\\ &= \boldsymbol{\beta},\end{aligned}$$

since $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X} = \mathbf{I}$ and $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$. Furthermore, the variance of $\hat{\boldsymbol{\beta}}$ is given by

$$\begin{aligned}\text{Var}[\hat{\boldsymbol{\beta}}] &= \text{Var}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}] \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \text{ Var}[\mathbf{y}]((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top)^\top \\ &= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} \\ &= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}.\end{aligned}$$

In Bayesian linear regression, the parameters $\boldsymbol{\beta}$ and σ^2 are estimated in a different way and usually through stochastic simulation methods. For instance, in the classical context the asymptotic distribution for $\hat{\boldsymbol{\beta}}$ will always be $\text{N}_d(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$, since the Central Limit Theorem holds this result [1, page 244]. On the other hand, in the Bayesian context the distribution of $\hat{\boldsymbol{\beta}}$ will not necessarily be Normal, since it depends on the choice of the prior distribution.

2.3 Bayesian Linear regression

In this Section we present the Bayesian linear regression models. Also, we introduce the Bayesian perspective, prior distributions and estimation methods.

2.3.1 Basics concepts of Bayesian inference

Under the Bayesian perspective, we aim, both through data and subjective information, to draw conclusions about a certain unknown quantity of interest by using probabilistic models. Under the classical point of view, the inference is also made utilizing probabilistic models, but only the information from the data is considered and any other extra information is not incorporated into the decision process.

The inclusion of the subjective information in the inference process is the main point in Bayesian analysis. That is made by inserting a prior distribution that describes all the available knowledge that one can have about the quantity of interest, and its choice may influence the final results depending on how much data is available. That is, the more data, the less the impact of the prior information on the final conclusions [2].

There are many types of prior distributions, such as non-informative, conjugate, improper etc. The non-informative ones are based on the sampling distribution and their idea is to have a default prior distribution when there is no information about the problem at hand. For instance, the Jeffreys prior is non-informative and it is proportional to the Fisher Information, which is the expected value of the second derivative of the log-likelihood function with respect to the parameter of interest [3]. Although the Jeffreys prior is called non-informative, the Fisher Information quantify the variability of the parameter based on the available data. That is, the higher the value of the Fisher Information, the more concave is the log-likelihood, thus evidencing that the data helps to estimate the quantity of interest.

The conjugate priors are a class of distributions that present the same parametric form of the likelihood function and their choice is frequently related to mathematical and computational convenience [3]. As a consequence of conjugacy, the posterior distribution may be obtained analytically and posterior samples are generated straightforwardly. On the other hand, improper priors are distributions that, in their parametric space, do not integrate to 1. For instance, in some cases Jeffreys priors are improper, but the posterior distribution is proper; see Section 3.2 of [2].

Consider $\boldsymbol{\theta} = (\theta_0, ..., \theta_{d-1})^\top$ an unknown vector of parameters that we are interested in estimating and $\mathbf{y} = (y_1, ..., y_n)^\top$ an $n \times 1$ column vector assumed to be a realization of the random variable whose distribution is $p(y_i|\boldsymbol{\theta})$. The likelihood function of the y_i is given by

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{y}) = \prod_{i=1}^{n} p(y_i|\boldsymbol{\theta}). \tag{2}$$

All the information from the observations y_i about $\boldsymbol{\theta}$ is included in (2). The difficulty in estimating $\boldsymbol{\theta}$ becomes an optimization problem of maximizing the likelihood function (or its logarithm). In constrast, under the Bayesian methodology the estimate of $\boldsymbol{\theta}$ is given by the joint posterior distribution, which is defined by the Bayes' theorem,

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\mathcal{L}(\theta|\mathbf{y})p(\boldsymbol{\theta})}{\int_\Theta \mathcal{L}(\theta|\mathbf{y})p(\boldsymbol{\theta})d\boldsymbol{\theta}}, \tag{3}$$

where Θ represents the parametric space of $\boldsymbol{\theta}$ and $p(\boldsymbol{\theta})$ the prior distribution. Equation (3) can also be written as

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto \mathcal{L}(\theta|\mathbf{y})p(\boldsymbol{\theta}), \tag{4}$$

since $\int_\Theta \mathcal{L}(\theta|\mathbf{y})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ is the marginal distribution of $\mathbf{y}$ and does not depend on $\boldsymbol{\theta}$. The posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ provides all the information that one can have about $\boldsymbol{\theta}$. For instance, it is possible to evaluate $p(\boldsymbol{\theta}|\mathbf{y})$ and its mean, median, variance and some other quantities such as quantiles in order to have point and interval estimates. Besides, the posterior distribution frequently has no closed form, thus depending on computational methods to be obtained.

When the posterior distribution is available, one can be interested about the predictive posterior distribution, which is utilized to predict unobserved values of the response outcome, $\tilde{\mathbf{y}}$, and the marginal distribution of $\mathbf{y}$. To obtain these two distributions, the constant that was not considered in (4) is necessary.

$$p(\tilde{\mathbf{y}}|\mathbf{y}) = \int_\Theta p(\tilde{\mathbf{y}}|\boldsymbol{\theta},\mathbf{y})p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}, \qquad \tilde{\mathbf{y}} \sim p(\tilde{\mathbf{y}}|\boldsymbol{\theta},\mathbf{y}),$$
$$p(\mathbf{y}) = \int_\Theta \mathcal{L}(\boldsymbol{\theta}|\mathbf{y})p(\boldsymbol{\theta})d\boldsymbol{\theta}.$$

The advantages of the Bayesian inference when compared to the classical one are that all the information available is considered and its probabilistic interpretation about the estimates is straightforward [4]. The prior knowledge is inserted through the prior distribution and combined with the data, represented by the likelihood function, all inference is carried out based on the posterior distribution. In parallel, the interpretation of the interval estimates does not involve assumptions about replications of the experiment. That is, in the Bayesian context given the interval estimates, $\boldsymbol{\theta}$ belongs to it with $(1-\alpha)\%$ probability.

2.3.2 Linear model: conjugate priors

Here we consider a Bayesian linear model in the form

$$\mathbf{y} = \mathbf{X}\beta + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathrm{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_d),$$

where $\sigma^2 > 0$, $\mathbf{I}_d$ an identity matrix, $\boldsymbol{\beta} = (\beta_0, ..., \beta_{d-1})^\top$ a $d \times 1$ vector, $\mathbf{X}$ an $n \times d$ design matrix and we assume that ϵ_i's are independent. The likelihood function is also

$$f_{\mathrm{y}}(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma^2) = (2\pi\sigma^2)^{-n/2}\exp\Big\{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\Big\}$$

Under the Bayesian point of view, the inference process involves data and prior information. Thus, we assume a $\mathbf{N}_d(\mathbf{m}, \sigma^2\mathbf{V})$, which is a conjugate prior distribution for $\boldsymbol{\beta}|\sigma^2$ as follows

$$f(\boldsymbol{\beta}|\sigma^2,\mathbf{m},\mathbf{V}) = (2\pi\sigma^2)^{-d/2}|\mathbf{V}|^{-d/2}\exp\Big\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\mathbf{m})^\top\mathbf{V}^{-1}(\boldsymbol{\beta}-\mathbf{m})\Big\}.$$

For σ^2, we also set a conjugate prior distribution given by an Inverse Gamma denoted by IG(a, b) in the form of

$$f(\sigma^2|a,b) = \frac{b^a}{\Gamma(a)}(\sigma^2)^{a-1}\exp\Big\{-\frac{b}{\sigma^2}\Big\},$$

where $a > 0$ and $b > 0$. Since we have the likelihood function and the proper priors, we can then find the posterior distribution to make inference on the parameters $\boldsymbol{\beta}$ and σ^2. Using the Bayes' theorem, we have

$$\begin{aligned} f(\boldsymbol{\beta},\sigma^2|\mathbf{y},\mathbf{X}) &= \frac{f_{\mathrm{y}}(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma^2)f(\boldsymbol{\beta}|\sigma^2,\mathbf{m},\mathbf{V})f(\sigma^2|a,b)}{f_{\mathrm{y}}(\mathbf{y})}, \\ &\propto f_{\mathrm{y}}(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma^2)f(\boldsymbol{\beta}|\sigma^2,\mathbf{m},\mathbf{V})f(\sigma^2|a,b), \\ &\propto (\sigma^2)^{-\frac{n}{2}-\frac{d}{2}+a-1}\exp\Big\{-\frac{A}{2\sigma^2}\Big\}, \end{aligned} \tag{5}$$

where

$$\begin{aligned} A &= (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + (\boldsymbol{\beta}-\mathbf{m})^\top\mathbf{V}^{-1}(\boldsymbol{\beta}-\mathbf{m}) + 2b, \\ &= \mathbf{y}^\top\mathbf{y} - \mathbf{y}\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\mathbf{V}^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^\top\mathbf{V}^{-1}\mathbf{m} - \mathbf{m}^\top\mathbf{V}^{-1}\boldsymbol{\beta} + \mathbf{m}^\top\mathbf{V}^{-1}\mathbf{m} + 2b, \\ &= \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X}+\mathbf{V}^{-1})\boldsymbol{\beta} - \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{y}+\mathbf{V}^{-1}\mathbf{m}) + (\mathbf{m}^\top\mathbf{V}^{-1}\mathbf{m} + 2b + \mathbf{y}^\top\mathbf{y}) - (\mathbf{y}^\top\mathbf{X}+\mathbf{m}^\top\mathbf{V}^{-1})\boldsymbol{\beta}. \end{aligned}$$

For convenience, let $\boldsymbol{\Lambda} = (\mathbf{X}^\top\mathbf{X}+\mathbf{V}^{-1})^{-1}$ a $d \times d$ matrix and $\boldsymbol{\mu} = (\mathbf{X}^\top\mathbf{X}+\mathbf{V}^{-1})^{-1}(\mathbf{X}^\top\mathbf{y}+\mathbf{V}^{-1}\mathbf{m})$ a $d \times 1$ vector. Hence,

$$\begin{aligned} A &= \boldsymbol{\beta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\beta} + \mathbf{m}^\top\mathbf{V}^{-1}\mathbf{m} + 2b + \mathbf{y}^\top\mathbf{y}, \\ &= (\boldsymbol{\beta}-\boldsymbol{\mu})^\top\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}) - \boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} + \mathbf{m}^\top\mathbf{V}^{-1}\mathbf{m} + 2b + \mathbf{y}^\top\mathbf{y}. \end{aligned}$$

Finally, the joint posterior distribution for $\boldsymbol{\beta}$ and σ^2 is given by

$$\begin{aligned} f(\boldsymbol{\beta},\sigma^2|\mathbf{y},\mathbf{X}) \propto& f_{\mathrm{y}}(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma^2)f(\boldsymbol{\beta}|\sigma^2,\mathbf{m},\mathbf{V})f(\sigma^2|a,b), \\ \propto& (\sigma^2)^{-\frac{d}{2}}\exp\Big\{-\frac{(\boldsymbol{\beta}-\boldsymbol{\mu})^\top\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})}{2\sigma^2}\Big\} \\ &\times (\sigma^2)^{-\frac{n}{2}+a-1}\exp\Big\{-\frac{\mathbf{m}^\top\mathbf{V}^{-1}\mathbf{m} - \boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} + 2b + \mathbf{y}^\top\mathbf{y}}{2\sigma^2}\Big\}. \end{aligned} \tag{6}$$

Therefore, the equation (6) shows that the posterior distribution $f(\boldsymbol{\beta},\sigma^2|\mathbf{y},\mathbf{X})$ is proportional to the multiplication of kernels of the $\mathbf{N}_n(\boldsymbol{\mu},\sigma^2\boldsymbol{\Lambda})$ and IG $(a^* = -\frac{n}{2}+a,\ b^* = b + \frac{\mathbf{m}^\top\mathbf{V}^{-1}\mathbf{m}-\boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}+\mathbf{y}^\top\mathbf{y}}{2})$. The manipulations presented above are partially available in [2], [4] and [16]. Additionally, [4] shows the normalizing constant and the marginal distribution of $\mathbf{y}$, which are necessary for equation (5).

2.3.3 Linear model: non-conjugate prior

In order to illustrate a Bayesian linear model with non-conjugate prior where Metropolis-Hastings may be helpful, consider $\mathbf{y} \sim \mathbf{N}_n(\mathbf{X}\boldsymbol{\beta},\sigma^2\mathbf{I})$ and assume that σ^2 is known. Also, consider a Laplace distribution as prior for each β_i such as

$$f(\beta_i|m_i,v_i) = (2v_i)^{-d/2}\exp\Big\{-\sum_{i=1}^{d}\frac{|\beta_i-m_i|}{v_i}\Big\},$$

where $\beta_i \in \mathbb{R}$, $m_i \in \mathbb{R}$ and $v_i > 0$. In this case, the posterior distribution is given by

$$f(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}, \sigma^2) \propto f_{\mathrm{y}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) \prod_{i=1}^{d} f(\beta_i|m_i, v_i),$$
$$\propto \exp\Big\{ -\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \sum_{i=1}^{d} \frac{|\beta_i - m_i|}{v_i} \Big\}.$$

The equation above has no closed form of a known p.d.f or p.m.f and MCMC methods can be utilized in order to obtain samples from the posterior distributions of β_i. Although the Gibbs Sampling is an MCMC method, in this case it cannot be used since it is not possible to sample directly from $f(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}, \sigma^2)$. In contrast, Metropolis-Hasting can be applied.

2.3.4 MCMC methods

In order to estimate the unknown quantities in the Bayesian models, Markov Chain Monte Carlo (MCMC) methods are useful tools to sample from those posterior distributions that have no closed form. In this section we briefly introduce two MCMC algorithms commonly used: Gibbs Sampling [5, 6] and Metropolis-Hastings [7, 8].

In the statistical context, Monte Carlo integration is convenient when it is not possible to analytically compute a finite integral such as

$$\int g(\theta)p(\theta)d\theta, \tag{7}$$

where $p(\cdot)$ is a p.d.f or p.m.f and $g(\cdot)$ a integrable function. In short, i.i.d samples are generated from $p(\cdot)$ and evaluated at $g(\cdot)$ and then averaged. For a sufficiently large number of samples, the Strong Law of Large Numbers holds this average converges almost surely to (7) [9]. When it is not possible to directly sample from $p(\cdot)$, Gibbs Sampling and Metropolis-Hastings are alternatives to generate samples $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(K)}$ from $p(\cdot)$ no longer independent (now with a Markovian dependence structure) that evaluated at $t(\cdot)$ and averaged over the samples, also converge almost surely to (7).

The Gibbs Sampling is a stochastic simulation algorithm via Markov Chain utilized when the joint posterior distribution has no closed form but all its conditional distributions do. For instance, consider that $p(\boldsymbol{\theta}|\mathbf{y})$ is the joint posterior distribution and suppose that its conditional distributions may be written as

$$p(\theta_k|\theta_0, ..., \theta_{k-1}, \theta_{k+1}, ..., \theta_{d-1}, \mathbf{y}), \text{ where } k = 0, ..., d-1.$$

The idea is to successively sample from each conditional distribution of θ_k in order to obtain samples from the joint posterior distribution. The Gibbs Sampling is described as following

---

1 **Algorithm** Gibbs Sampling

1. Initialize $t = 1$ and define initial values $\theta_0^{(0)}, \theta_1^{(0)}, ..., \theta_{d-1}^{(0)}$ for the vector $\boldsymbol{\theta} = (\theta_0, \theta_2, ..., \theta_{d-1})^\top$.
2. Sample

$$\begin{aligned}
\theta_0^{(t)} &\sim p(\theta_0|\theta_1^{(t-1)}, \theta_2^{(t-1)}, \theta_3^{(t-1)}, ..., \theta_{d-1}^{(t-1)}, \mathbf{y});\\
\theta_1^{(t)} &\sim p(\theta_1|\theta_2^{(t)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, ..., \theta_{d-1}^{(t-1)}, \mathbf{y});\\
&\vdots\\
\theta_{d-1}^{(t)} &\sim p(\theta_{d-1}|\theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, ..., \theta_{d-2}^{(t)}, \mathbf{y});
\end{aligned}$$

3. Take $t = t + 1$ and return to the step 2 until the desired sample has been obtained for each θ_k.

Similarly to the Gibbs Sampling, Metropolis-Hastings is also a stochastic algorithm that generates samples with Markovian dependence structure from a certain distribution (or kernel). Further, Metropolis-Hastings is more flexible than Gibbs Sampling, since it can be utilized to generate samples from distributions that have or not closed form. When it is possible to generate directly from the conditional or joint distribution, it is appropriate to use Gibbs Sampling, once Metropolis-Hastings has an acceptance and rejection step. But when at least one of the conditional distributions have no closed form, Metropolis might be an alternative. In some cases Metropolis and Gibbs Sampling are combined because some conditionals have closed form and others do not. This hybrid algorithm is called Metropolis-within-Gibbs and was proposed by [10] and [11].

Again, suppose we desire to obtain a sample from the joint posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$. The Metropolis-Hastings is based on a proposal distribution $q(\boldsymbol{\theta}^{(t-1)})$, which generates candidate values $\boldsymbol{\theta}^*$ that are accepted as values from $p(\boldsymbol{\theta}|\mathbf{y})$ with certain probability. In its first version [7], the $q(\boldsymbol{\theta}^{(t-1)})$ is only Normal, where the mean has a Markovian structure and the variance is constant. A more general framework was proposed by [8] in which the proposal distribution can be other than Normal, and since then many adaptive Metropolis algorithms have been introduced, which basically differ by the specification of the covariance matrix of the proposal distribution [12, 13, 14, 15]. Below, the Metropolis algorithm proposed by [8] is described.

---

1 **Algorithm** Metropolis-Hastings

1. Initialize $t = 1$ and define the initial values $\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_d^{(0)}$ for the vector $\boldsymbol{\theta} = (\theta_0, \theta_1, ..., \theta_{d-1})$;
2. Sample $\boldsymbol{\theta}^*$ from the proposal distribution $q(\boldsymbol{\theta}^{(t-1)})$;
   (a) Compute
   $$\alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{p(\boldsymbol{\theta}^*|\mathbf{y})q(\boldsymbol{\theta}^{(t-1)})}{p(\boldsymbol{\theta}^{(t-1)}|\mathbf{y})q(\boldsymbol{\theta}^{(*)})}\right\}, \tag{8}$$
   (b) Compute $u \sim U[0,1]$. If $u < \alpha(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\theta}^*)$, then $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$, otherwise, $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$;
3. Take $t = t + 1$ and return to the step 2 until the desired posterior sample has been obtained.

The choice of the proposal distribution must be based on the support of $\boldsymbol{\theta}$. For example, if $\boldsymbol{\theta} \in \mathbb{R}^d$ it is appropriate to choose a proposal that is also supported on $\mathbb{R}^d$, otherwise the results generated by Metropolis might be invalid.

One important characteristic of the MCMC algorithms is that they generate chains that need to converge. That is, for each component of $\boldsymbol{\theta}$ or $\boldsymbol{\beta}$, a chain of values with Markovian dependence structure is generated and one must verify its convergence to the joint posterior distribution. Due to it, we usually set a warm-up period, which is called burn-in, from which the samples start being considered and all inference is made utilizing only these samples; see [9] for theoretical convergence results of the MCMC methods.

3 Method

3.1 R code

In this section we use R code to implement Bayesian linear regression models on both simulated data and some real-world data, specifically, data containing the salary, years of experience and gender of bank branch managers of a big bank.

To begin, we look at the classical linear model.

The regression problem involves determining the relationship between some response variable, Y, and a set of predictor variables $\mathbf{X} = (X_1, ...X_p)$. Usually, we assume that this relationship can be described by a deterministic function ,f, and some additive random error that follows a Normal distribution centred at 0 and with variance equals to σ^2:

$$Y = f(\mathbf{X}) + \epsilon$$

The predictors, $\mathbf{X}$, are assumed to be observed without error, so they're not considered **random**. We can check that

$$f(\mathbf{X}) = E[Y|\mathbf{X} = \mathbf{x}]$$

meaning that the $f(\mathbf{X})$ describes the expectation over Y when $\mathbf{X}$ is observed. The true regression function is unknown and we have no way of determining its analytic form exactly, even if it exists. What we do is find approximations which are the closest to the truth as possible.

3.2 Basis functions

Assuming that f is made up of a linear combination of basis functions and the correspondent coefficients, it can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{k} \beta_i B_i(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}$$

where $\beta = (\beta_i, ...\beta_k)$ is the set of coefficients corresponding to basis functions $\mathbf{B} = (Bi, ...Bk)$.

3.3 The Classic Linear Model

In this section, the data used for the model was simulated.

```
library(tidyverse) # Always
```

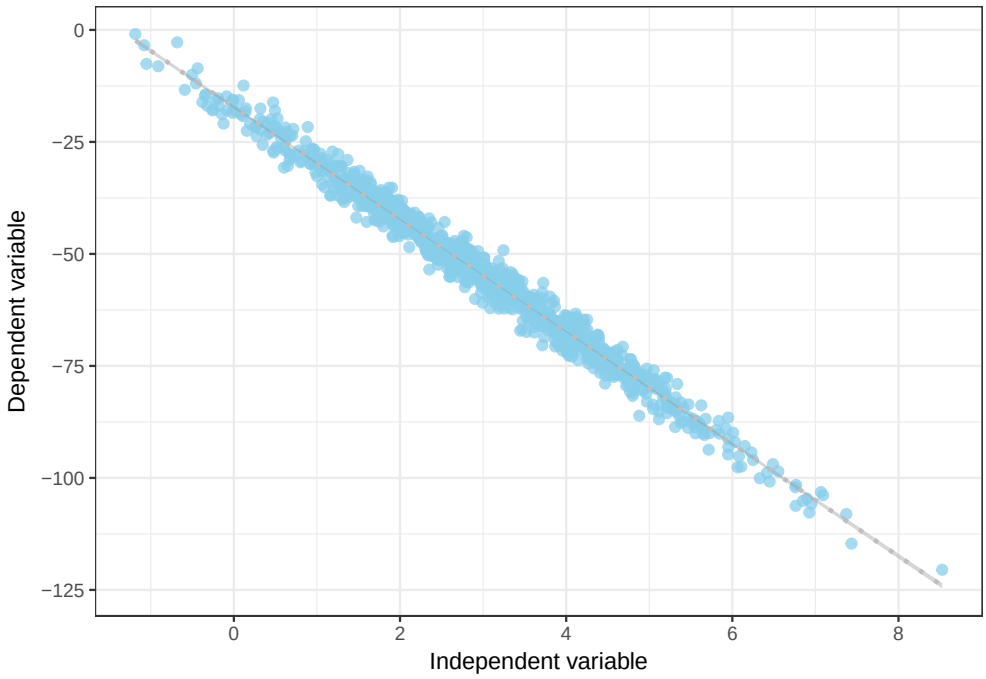
```
## -- Attaching packages ------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ---------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Simulating  some data ----------------------------------
set.seed(2018) # reproducibility

# Simulating the independent variable (arbitrarily
# chosen as ~Normal and the error ~ N(0, sigma^2)
sigma <- 1/rgamma(n = 1, shape = 0.5, rate = 1)
x <- rnorm(n = 1000, mean = 3, sd = 1.5)
e <- rnorm(n = 1000, mean = 0, sd = sqrt(sigma))
V <- matrix(sigma*c(10, 0, 0, 10), ncol = 2, nrow = 2)

# Priors of the parameters - intercept and slope
betas <- MASS::mvrnorm(n = 1, mu = c(0, 0),
                       Sigma = V)


# The regression model
y <- betas[1] + (betas[2] * x) + e

da <- data.frame(y, x)

# Plotting our data
```

```
da %>%
  ggplot(aes(x, y)) +
  geom_point(colour = 'skyblue', size = 2, alpha = 0.75) +
  geom_smooth(method = 'lm', colour = 'grey', linetype = 'dotted') +
  theme_bw() +
  labs(x = 'Independent variable', y = 'Dependent variable')
```




```
# The classical regression model
# With functions:
lm(y ~ x) %>% summary()
```

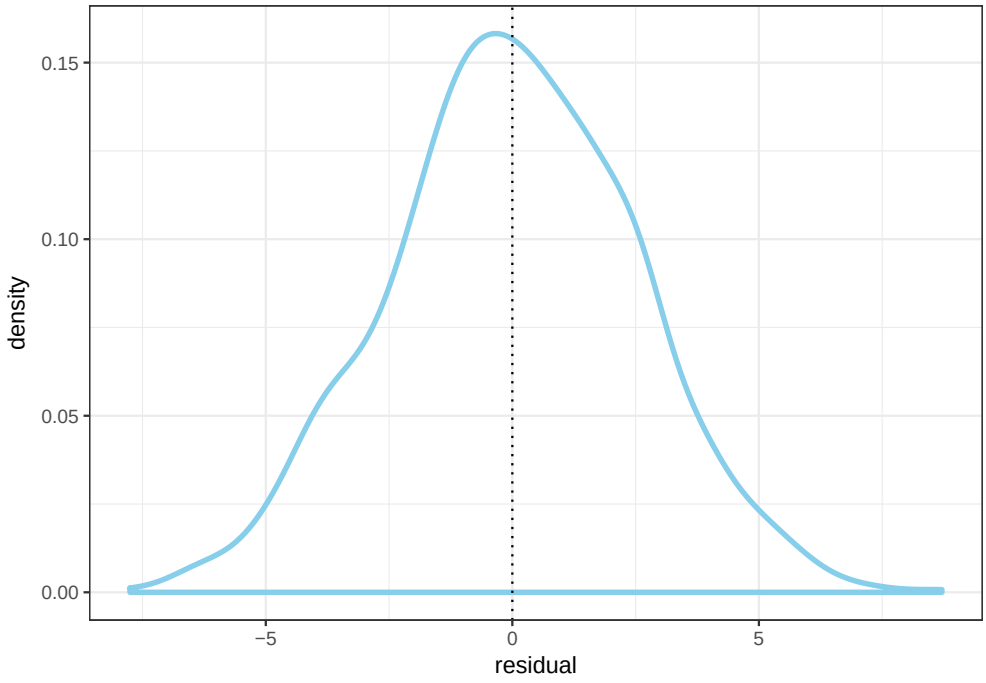
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7544 -1.5793 -0.0066  1.7400  8.7116
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.21746    0.17521  -98.27   <2e-16 ***
## x           -12.53167    0.05199 -241.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.502 on 998 degrees of freedom
## Multiple R-squared:  0.9831, Adjusted R-squared:  0.9831
## F-statistic: 5.811e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
# Vanilla flavour:
mm <- model.matrix( ~ x, data = da)
```

```r
k <-  ncol(mm)
n <- nrow(mm)

# Estimating betas
v <- solve(t(mm) %*% mm)
betas_hat <- c(v %*% t(mm) %*% y)
betas_hat
```

```
## [1] -17.21746 -12.53167
```

```r
# y_hat
y_hat <- mm %*% betas_hat

da$res <- y - y_hat

da %>%
  ggplot(aes(res)) +
  geom_density(colour = 'skyblue', size = 1.2, alpha = 0.75) +
  geom_vline(xintercept = 0, linetype = 'dotted') +
  theme_bw() +
  labs(x = 'residual')
```



The plot above illustrates an approximately normal distribution of residuals produced by a model.

```r
# Residual sum of squares
rss <- sum((y - y_hat)^2)

# Mean squared errors
mse <- mean((y - y_hat)^2)

# Rs - Multiple correlation coefficients
(R <- sum((y_hat - mean(y))^2)/sum((y - mean(y))^2))
```

```
## [1] 0.983115
```

```
(R_adj <- 1 - (1 - R)*((n-1)/(n-k)))
```

```
## [1] 0.9830981
```

```
# Estimating the variance of the parameters
var_betas <- solve(t(mm) %*% mm) * mse
var_betas
```

```
##               (Intercept)            x
## (Intercept)  0.030635895 -0.008110332
## x           -0.008110332  0.002697209
```

```
# t-values
t1 <- betas_hat[2]/sqrt(var_betas[2, 2])
t2 <- betas_hat[1]/sqrt(var_betas[1, 1])
```

3.4 The Bayesian Linear Model

The Bayesian approach consists of: 1. assign prior distributions to all the unknown parameters; 2. write down the likelihood of the data given the parameters; 3. determine the posterior distributions of the parameters given the data using Bayes' Theorem.

3.4.1 1. We find that the conjugate choice of (joint) prior for

β and σ^2 is the normal inverse-gamma (NIG), denoted by $NIG(\mathbf{m}, \mathbf{V}, a, b)$, with probability density function:

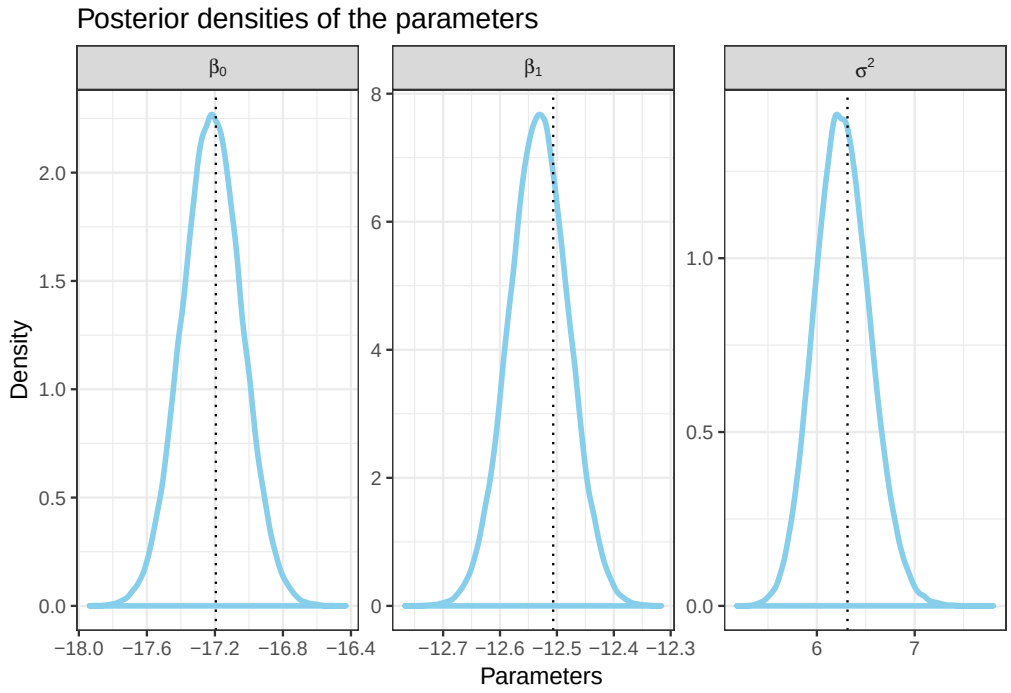
$$p(\beta, \sigma^2) = p(\beta|\sigma^2)p(\sigma^2)$$
$$p(\beta, \sigma^\mathbf{2}) = N(\mathbf{m}, \sigma^2\mathbf{V}) \times IG(a, b)$$

So, the β, given σ^2 are assumed to have a Normal distribution, since its domain goes from $-\infty$ to $+\infty$, and its mean and variance can be adjusted accordingly to the expertise of the one who is building the model. The σ^2 is assumed to have the IG(a, b) distribution as its domain goes from 0 to $+\infty$.

```
# Bayesian model ------------------------------------------------------

# Priors were already assigned as -------------
# Bs ~ Normal(m, V) = Normal(0, sigma^2 * 10)
# Sigma ~ IG(a, b) = IG(0.5, 1)

# Posterior distribution ----------------------
# Betas | Sigma, y ~ N(m*, V*)
# Sigma | y ~ IG(a*, b*), where:
#
# m* = (V^-1 + B'B)^-1 * (V^-1*m + B'Y)
# V* = (V^-1 + B'B)^-1
# a* = a + n/2
# b* = b + (m' V^-1 m + Y'Y -  (m*)'(V*)^-1m*)/2
#---------------------------------------------

v_star <- solve(solve(V) + t(mm) %*% mm)
m_star <- v_star %*% (solve(V) %*% c(0, 0) + t(mm) %*% y)
a_star <- 0.5 + n/2
```

```r
b_star <- 1 + (t(c(0, 0)) %*% solve(V) %*% c(0, 0) +
                 (t(y) %*% y) - t(m_star) %*% solve(v_star) %*% m_star)/2

# Sampling from the posteriors ----------------------------------------------------
sim <- 100000
gamma <- rgamma(sim, shape = a_star,
                rate = b_star)

# For the variance
sigma_sim <- 1/gamma


# Consider that  you have a random variable
# Y ~ Normal(mu, variance), if you multiply it
# by a constant 'c' the variance is multiplied
# by c squared, aka, cY ~ Normal(mu, variance * c^2)

# For the random error, we used that, if Y ~ Normal(0, v*),
# sqrt(sigma) * Y ~ Normal(0, v*sigma), which is our
# target distribution

err <- sqrt(sigma_sim)*MASS::mvrnorm(n = sim, mu = c(0, 0),  v_star)

# consider now that we are adding the random
# error ~ Normal(0, v*sigma) to a constant
# (the estimated mean for the betas), which will lead us to have
# beta ~ Normal(m_star, v*sigma), as we just added a
# constant to the location of the distribution

params <- data.frame(par = c(rep(c(m_star), each = sim) +
                           c(err[,1], err[,2]), sigma_sim))

params$groups <- as.factor(rep(c(1:3), each = sim))
params$groups_label <- factor(params$groups, labels =
                            c('beta[0]', 'beta[1]', 'sigma^2'))


params_prior <- c(betas, sigma)

vline <- function(group){
  geom_vline(data = dplyr::filter(params,
                              groups == group),
             aes(xintercept = params_prior[group]), linetype = 'dotted')
}

params %>%
  ggplot(aes(par)) +
  geom_density(colour = 'skyblue', size = 1.2, alpha = 0.75) +
    1:3 %>% purrr::map(vline) +
  facet_wrap(~groups_label, scales = 'free',
             labeller = label_parsed) +
  theme_bw() +
  labs(x = 'Parameters', y = 'Density',
```

```
       title = 'Posterior densities of the parameters')
```



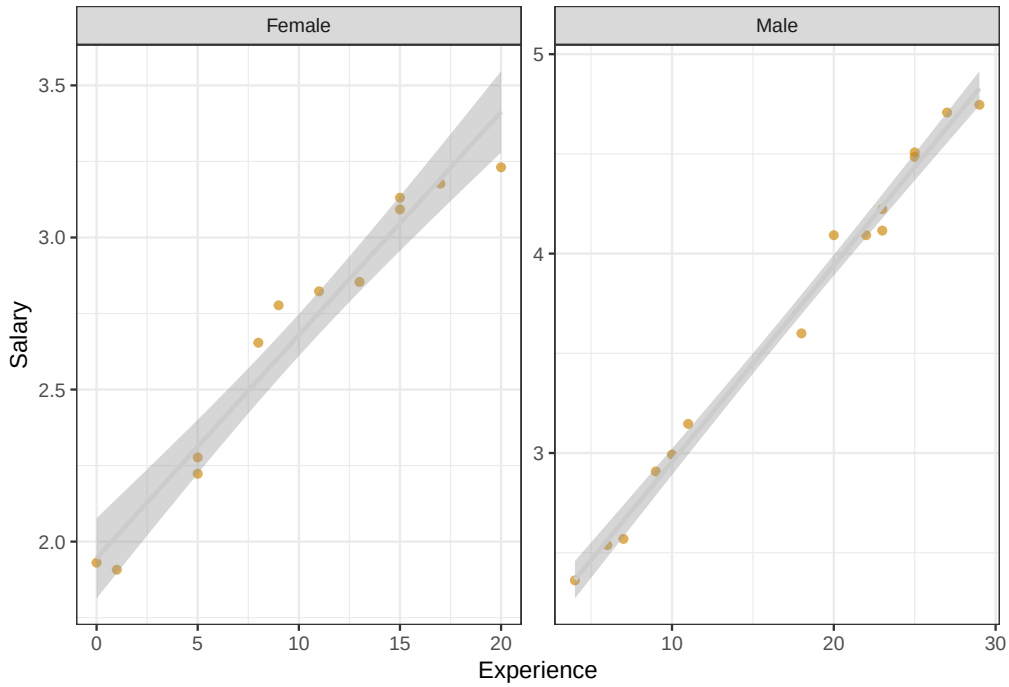
3.5 Real Data

In this section, the data used contains the salary, years of experience and gender of bank branch managers of a large bank.

```
# devtools::install_github('pet-estatistica/labestData')
da <- labestData::CharnetEg7.3 %>%
  mutate(gender = factor(sexo, labels = c('Female', 'Male'))) %>%
  select(-sexo)

head(da) %>% knitr::kable()
```

salario	exp	gender
1.93077	0	Female
3.17692	17	Female
2.27692	5	Female
3.13077	15	Female
2.77692	9	Female
3.09231	15	Female

```
da %>%
  ggplot(aes(y = salario, x = exp)) +
  geom_point(colour = 'orange3', alpha = 0.65, size = 1.5) +
  geom_smooth(method = 'lm', colour = 'grey80') +
  facet_wrap(~gender, scales = 'free') +
  theme_bw() +
  labs(y = 'Salary', x = 'Experience')
```



As can be seen from the above plots, there is a linear relationship between salary and years of experience for both genders. However, for the same years of experience, women earned less than men.

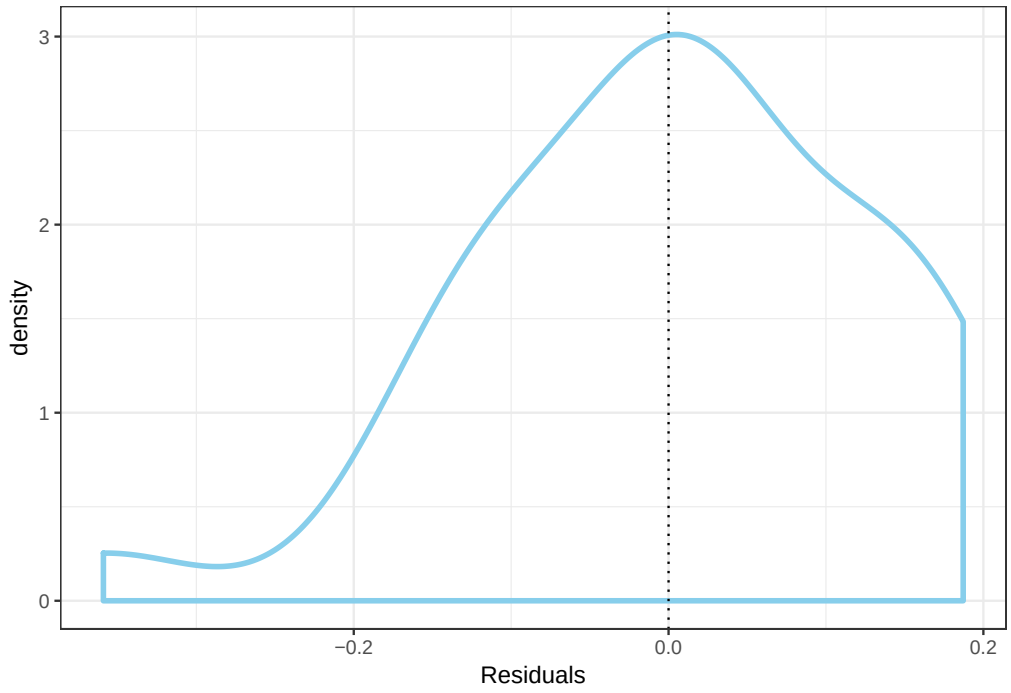
```
# The classic model
lm(salario ~ exp + gender, data = da) %>% summary()
```

```
##
## Call:
## lm(formula = salario ~ exp + gender, data = da)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35905 -0.08887 -0.00299  0.08037  0.18718
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.771484   0.050981  34.748  < 2e-16 ***
## exp         0.090917   0.003436  26.461  < 2e-16 ***
## genderMale  0.330990   0.056801   5.827 5.22e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1314 on 24 degrees of freedom
## Multiple R-squared:  0.9784, Adjusted R-squared:  0.9766
## F-statistic:   543 on 2 and 24 DF,  p-value: < 2.2e-16
```

```
# The classic model by hand ------------------------------------------------
mm <- model.matrix(~ exp + gender, data = da)
k <-  ncol(mm)
n <- nrow(mm)
v <- solve(t(mm) %*% mm)
betas_hat <- v %*% t(mm) %*% da$salario
betas_hat
```

```
##                   [,1]
## (Intercept) 1.77148445
## exp         0.09091695
## genderMale  0.33099028
```

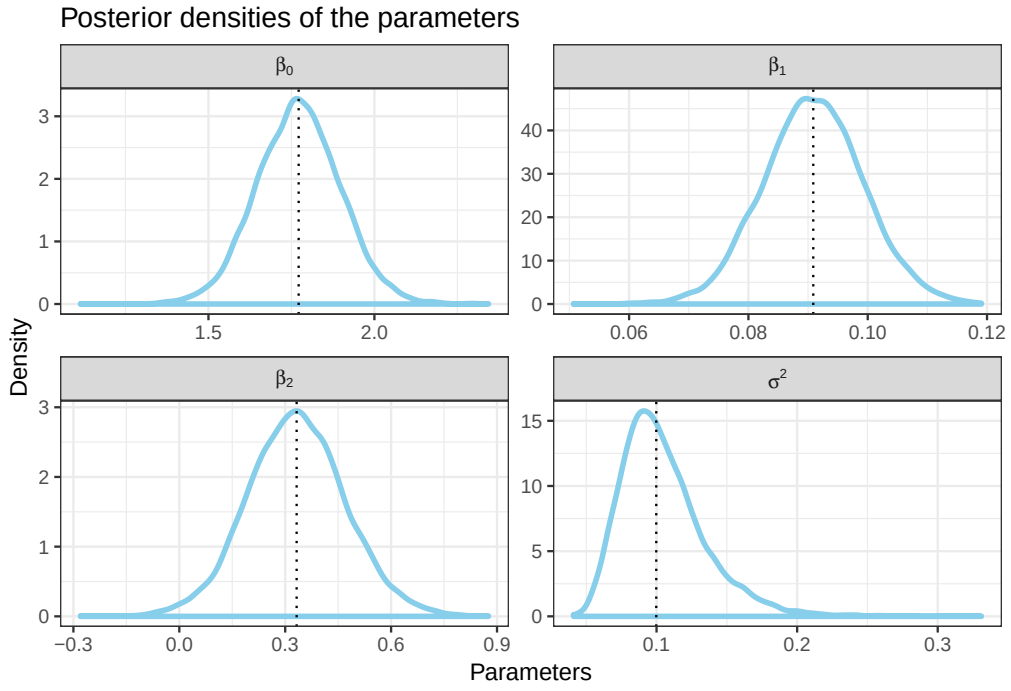
```r
y_hat <- mm %*% betas_hat
da$res <- da$salario - y_hat

da %>%
  ggplot(aes(res)) +
  geom_density(colour = 'skyblue', size = 1.2, alpha = 0.75) +
  geom_vline(xintercept = 0, linetype = 'dotted') +
  theme_bw() +
  labs(x = 'Residuals')
```




```r
# Bayesian model ------------------------------------------------------

# Priors were already assigned as -------------
# Bs ~ Normal(m, V) = Normal(0, sigma^2 * 10)
# Sigma ~ IG(a, b) = IG(0.5, 1)

# Posterior distribution ---------------
# Betas | Sigma, y ~ N(m*, V*)
# Sigma | y ~ IG(a*, b*), where:
#
# m* = (V^-1 + B'B)^-1 * (V^-1*m + B'Y)
# V* = (V^-1 + B'B)^-1
# a* = a + n/2
# b* = b + (m' V^-1 m + Y'Y -  (m*)'(V*)^-1m*)/2
#---------------------------------------

V <-  diag(10, nrow = dim(mm)[2], ncol = dim(mm)[2])
v_star <- solve(solve(V) + t(mm) %*% mm)
```

```r
m_star <- v_star %*% (solve(V) %*% c(0, 0, 0) + t(mm) %*% da$salario)
a_star <- 0.5 + n/2
b_star <- 1 + (t(c(0, 0, 0)) %*% solve(V) %*% c(0, 0, 0) +
                 (t(da$salario) %*% da$salario) -
                 t(m_star) %*% solve(v_star) %*% m_star)/2

# Sampling from the posteriors -------------------------------
sim <- 10000
gamma <- rgamma(sim, shape = a_star, rate = b_star)

# For the variance
sigma_sim <- 1/gamma


# Consider that  you have a random variable
# Y ~ Normal(mu, variance), if you multiply it
# by a constant 'c' the variance is multiplied
# by c squared, aka, cY ~ Normal(mu, variance * c^2)

# For the random error, we used that, if Y ~ Normal(0, v*),
# sqrt(sigma) * Y ~ Normal(0, v*sigma), which is our
# target distribution

err <- sqrt(sigma_sim) * MASS::mvrnorm(n = sim, mu = rep(0, 3), Sigma  = v_star)

# consider now that we are adding the random
# error ~ Normal(0, v*sigma) to a constant
# (the estimated mean for the betas), which will lead us to have
# beta ~ Normal(m_star, v*sigma), as we just added a
# constant to the location of the distribution

params <- data.frame(par = c(rep(c(betas_hat), each = sim) + as.vector(err),
                             sigma_sim))

params$groups <- as.factor(rep(c(1:4), each = sim))
params$groups_label <- factor(params$groups, labels =
                                c('beta[0]', 'beta[1]',
                                  'beta[2]', 'sigma^2'))

# Finding the 'optimal parameters' accordingly to
# a defined loss function - the values for the parameters that
# better represent the uncertainty after observing the data

# Loss for betas: chosen to be the mean squared residual
loss <- function(x, y) {
  mean((y - x)^2)
}

b0 <- optimize(function(x) loss(x, params$par[1:10000]),
               interval = c(-100, 100))$minimum
b1 <- optimize(function(x) loss(x, params$par[10001:20000]),
               interval = c(-100, 100))$minimum
b2 <- optimize(function(x) loss(x, params$par[20001:30000]),
```

```
                interval = c(0, 1))$minimum

# Loss for sigma: chosen to be the mean of the abs(residual)
loss_sigma <- function(x, y) {
  mean(abs(y - x))
}

s <- optimize(function(x) loss_sigma(x, params$par[30001:40000]),
                interval = c(0, 10))$minimum

params_est <- c(b0, b1, b2, s)

vline <- function(group){
  geom_vline(data = dplyr::filter(params,
                                  groups == group),
             aes(xintercept = params_est[group]), linetype = 'dotted')
}

params %>%
  ggplot(aes(par)) +
  geom_density(colour = 'skyblue', size = 1.2, alpha = 0.75) +
  1:4 %>% purrr::map(vline) +
  facet_wrap(~groups_label, scales = 'free',
             labeller = label_parsed) +
  theme_bw() +
  labs(x = 'Parameters', y = 'Density',
       title = 'Posterior densities of the parameters')
```



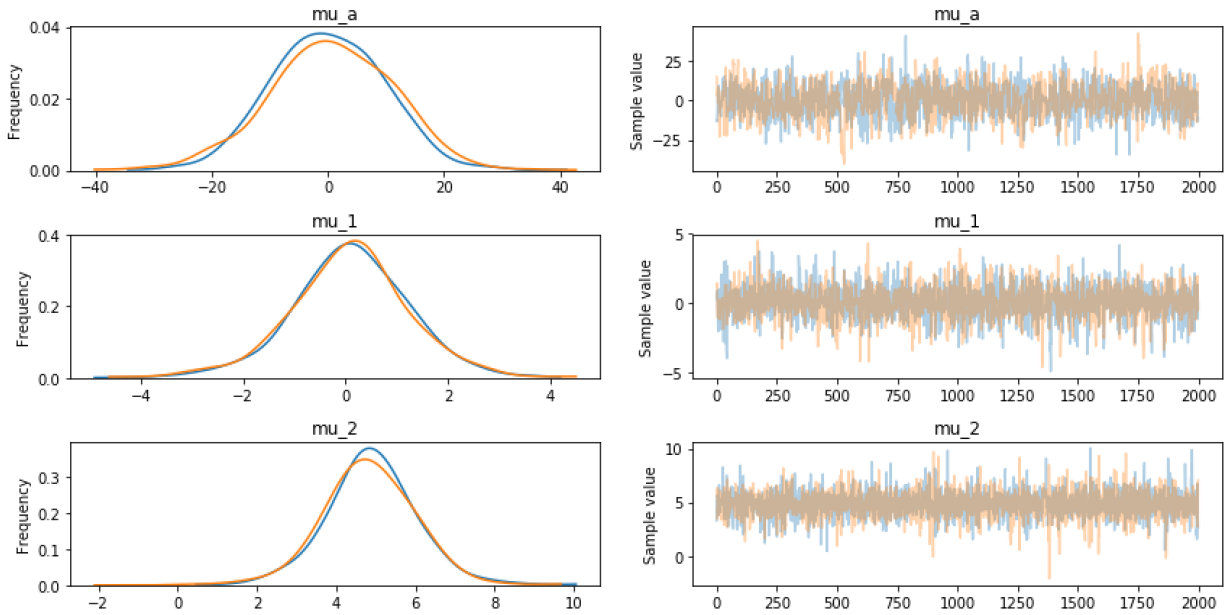
3.6 Python code

In this section python code was used to model Bayesian linear regression applied to bike sharing data. Specifically, temp ~ humidity + season were used from the data as the dependent and independent variables.

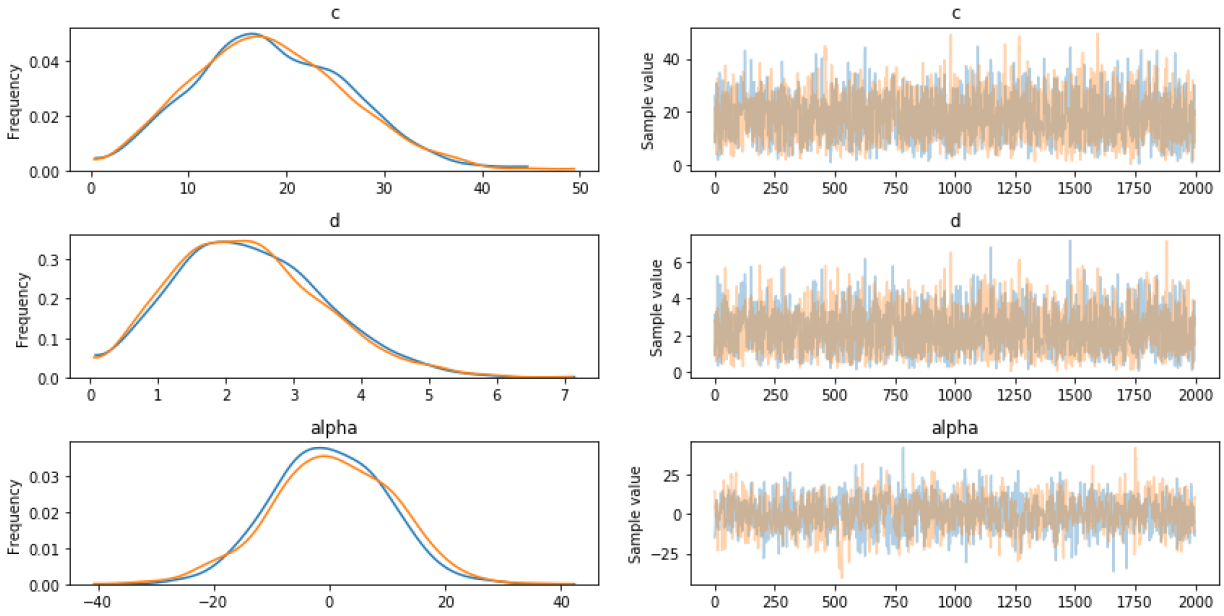
```
# Bayesian Linear Regression Model Applied to Bike Sharing Data

# Import packages to be used

import numpy as np # Basic package
import pandas as pd # Basic package
import os # For working directory
from os import chdir, getcwd
import matplotlib.pyplot as plt # Plotting
!pip install pymc3 # The exclaimation mark forces python to run the command
# pip within the shell. Usually you have to leave the ipython shell in order to
# run the command.
import pymc3 as pm # For Bayesian analysis
from pymc3 import Model, Normal, Gamma, InverseGamma

# Sort out working directory. (Will be useful for us new python users)
# Getting working directory
wd=getcwd()
wd
# set working directory
os.chdir('Insert working directory')

# Import data into python
bike=pd.read_csv('biketest.csv')

# Build model
basic_model=pm.Model()
with basic_model:
    mu_a=Normal('mu_a',mu=0,sd=10)
    mu_1=Normal('mu_1',mu=0,sd=10)
    mu_2=Normal('mu_2',mu=0,sd=10)
    c=Normal('c',10,10)
    d=Normal('d',10,10)
    tau=InverseGamma('tau',c,d)
    tau_a=Gamma('tau_a',10,10)
    tau_1=Gamma('tau_1',10,10)
    tau_2=Gamma('tau_2',10,10)
    sigma_a=1/tau_a
    sigma_1=1/tau_1
    sigma_2=1/tau_2
    sigma=1/tau
    alpha=Normal('alpha',mu=mu_a,sd=sigma_a)
    beta_1=Normal('beta_1',mu=mu_1,sd=sigma_1)
    beta_2=Normal('beta_2',mu=mu_2,sd=sigma_2)
    x_1=bike['humidity']
    x_2=bike['season']
    mu=beta_1*x_1+beta_2*x_2
    Y_obs=Normal('Y_obs',mu=mu,sd=sigma,observed=bike['temp'])
# No-U-Turn Sampler
```

```
    with basic_model:
        trace=pm.sample(2000)
# Trace
pm.traceplot(trace)
pm.plot_posterior(trace) # Histogram
pm.summary(trace)
```

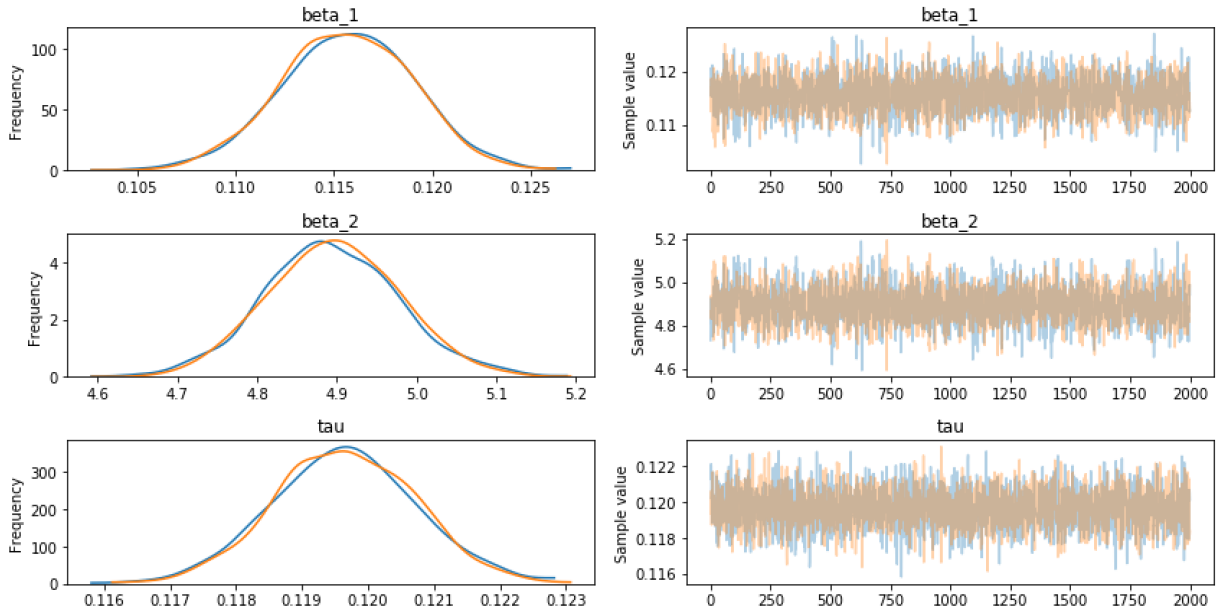
Shown below are the output plots generated from the Python code.



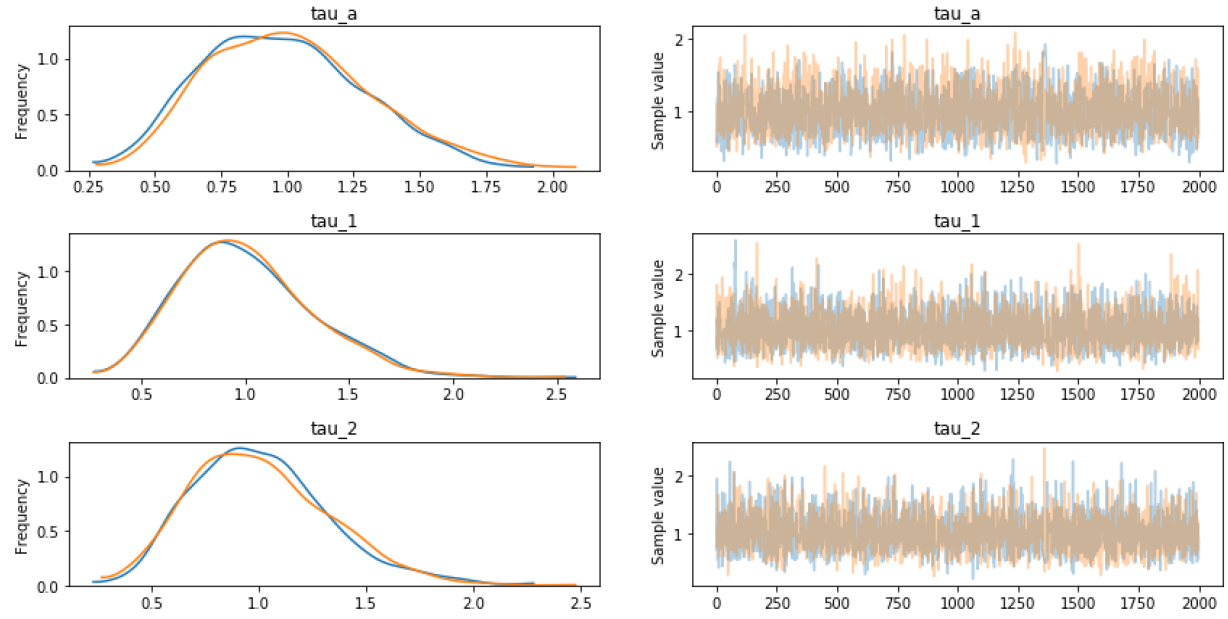
The above plots display the distributions for mu_a, mu_1 and mu_2.



The above plots display the distributions for c, d, and alpha.

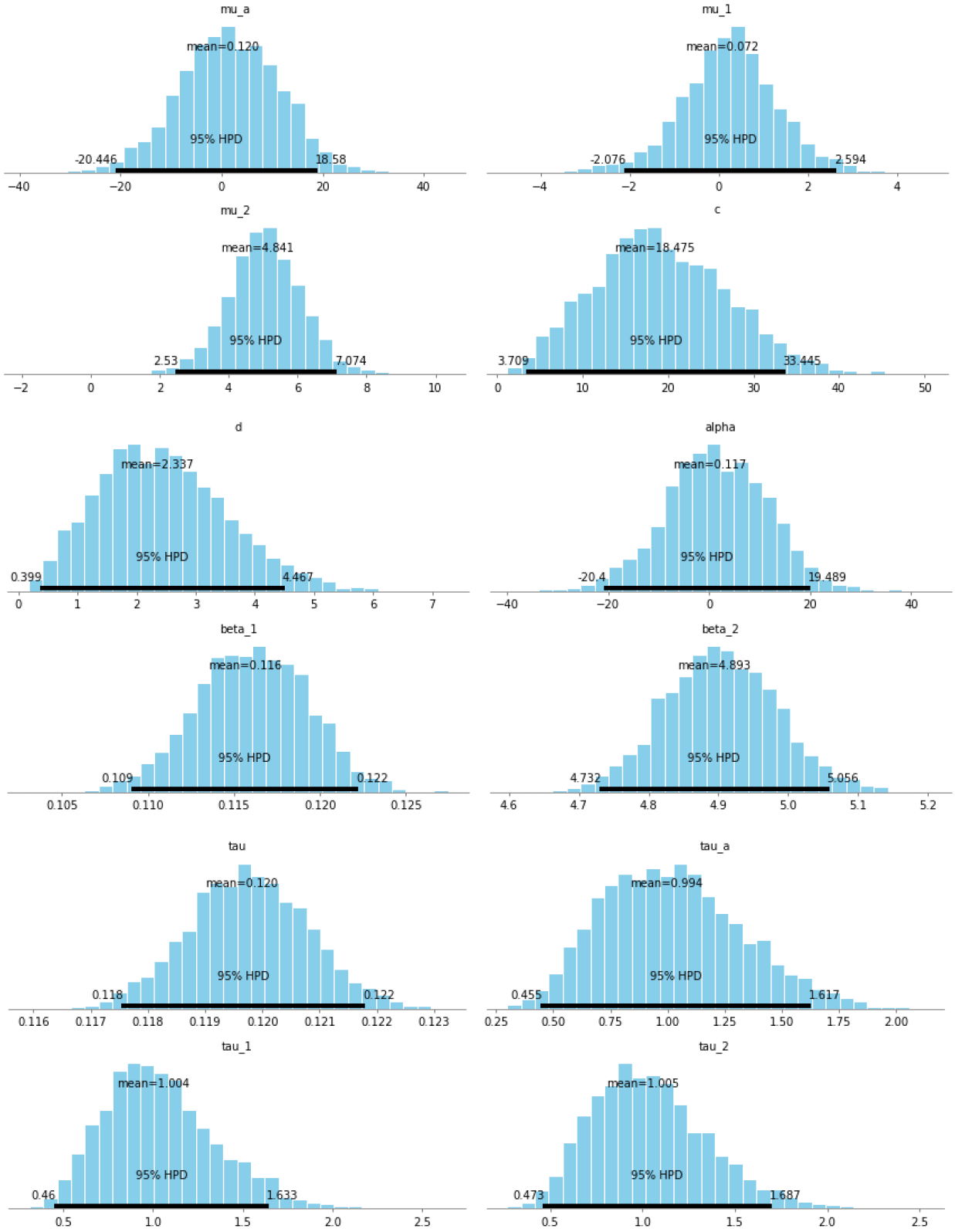


The above plots display the distributions for beta_1, beta_2 and tau.



The above plots display the distributions for tau_a, tau_1 and tau_2.

The following plots display histograms for the posteriors.


mu_a
mean=0.120
95% HPD
-20.446
18.58
mu_1
mean=0.072
95% HPD
-2.076
2.594
mu_2
mean=4.841
95% HPD
2.53
7.074
c
mean=18.475
95% HPD
3.709
33.445
d
mean=2.337
95% HPD
0.399
4.467
alpha
mean=0.117
95% HPD
-20.4
19.489
beta_1
mean=0.116
95% HPD
0.109
0.122
beta_2
mean=4.893
95% HPD
4.732
5.056
tau
mean=0.120
95% HPD
0.118
0.122
tau_a
mean=0.994
95% HPD
0.455
1.617
tau_1
mean=1.004
95% HPD
0.46
1.633
tau_2
mean=1.005
95% HPD
0.473
1.687

References

[1] W. Feller. An Introduction to Probability Theory and Its applications. 1967; John Wiley, Ed.3, vol. 1.

[2] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, D. Rubin. Bayesian Data Analysis. 2014; CRC, Ed.3, Boca Raton.

[3] C. Robert. The Bayesian Choice: from Decision-Theoretic Foundations to Computational Implementation. 2007; Springer, Ed. 2, New York.

[4] A. O'Hagan. Kendall's Advanced Theory of Statistics: Bayesian Inference. 1994; Arnold, Ed. 2B, London.

[5] S. Geman, D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Transactions Pattern Analysis and Machine Intelligence. 1984; 6:721–741.

[6] A. Gelfand, A. Smith. Sampling based approaches to calculating marginal densities. Journal of American Statistical Association. 1990; 85:398–409 Springer, New York.

[7] N. Metropolis, A. Rosenbluth, M. Teller, E. Teller. Equations of state calculations by fast computing machines. Journal of Chemistry and Physics. 1953;1087:1091–21.

[8] W. Hastings. Monte Carlo sampling using Markov chains and their applications. Biometrika. 1970; 57: 97-109.

[9] C. Robert, G. Casella. Monte Carlo Statistical Methods. 2004; Springer, New York.

[10] P. Muller. A generic approach to posterior integration and Gibbs sampling. Technical report, Purdue University, West Lafayette, Indiana.

[11] P. Muller. Alternatives to the Gibbs Sampling scheme. Technical report, Institute of Statistics and Decision Science, Duke University, Durham, North Carolina.

[12] D. Gamerman. Sampling from the posterior distribution in generalized linear mixed models. Statistics and Computing. 1997;57:68–7.

[13] H. Haario, E. Saksman, J. Tamminen. An adaptive Metropolis algorithm. Bernoulli. 2001;223:243–7.

[14] M. Vihola. Robust adaptive Metropolis algorithm with coerced acceptance rate. Statistics and Computing. 2012;997:1008–843.

[15] I. S. Mbalawata, S. Sarkka, M. Vihola, H. Haario. Adaptive Metropolis algorithm using variational Bayesian adaptive Kalman Filter. Computational Statistics and Data Analysis. 2015;101:115–83.

[16] D. Gamerman, H. F. Lopes. Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference. 2006; Chapman and Hall/CRC, Ed. 2, London.