Universidad Carlos III de Madrid
Bachelor's in Data Science and Engineering

Notes

Bayesian Data Analytics

Name:
Ricardo Vazquez Alvarez

August 28, 2024

Contents

Review on Probability and Stats

1.1 Random Variable

A Random Variable is just a **mapping** or a **function** that is measurable:

$$X : \Omega \longrightarrow \mathbb{R} \tag{1.1}$$

It is just a function that takes in Ω (**SET OF ALL POSSIBLE VALUES**) and *MAPS* it to a *Real Number*.

Example: Let Ω be the set of possible values. We want to obtain the probability that X has values in the interval $[2, 5]$ = the probability of selecting and element of Ω from $\{\omega \in \Omega : 2 \leq X(\omega) \leq 5\}$

1.2 Types of Random Variables

1.2.1 Cumulative Distribution Function (cdf)

The cdf of a random variable X is the function $F_X(x) : \mathbb{R} \longrightarrow [0, 1]$

$$F_X(x) = P_X((-\infty, x]) = P(X \leq x) \tag{1.2}$$

$$\boxed{F_X(x) = P(X \leq x)} \tag{1.3}$$

- *Therefore the cdf gives you the probability of $X(\omega) \leq x$.*

The **cdf** is the **INTEGRAL of the pdf function** (probability density function = $f_X(t)$):

$$\boxed{F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt} \tag{1.4}$$

1.2.2 Probability Density Function (pdf)

The probability density function give the **likelihood of an outcome**. This means that it shows the **Relative likelihood of the random variable taking on different values in its domain (x)**.

The pdf is simply just the derivative of the cdf:

$$\text{pdf} = \boxed{f_X(x) = \frac{d}{dx} F_X(x)} \tag{1.5}$$

A pdf must adhere to the following ***properties***:

1. The density must always be greater than 0.

$$f_X(x) \geq 0$$

2. The area of the entire pdf must be equal to 1.

$$\int_{-\infty}^{+\infty} f_X(x)\, dx = 1$$

Distributions to remember

2.1 Beta Function

The probability density function **pdf** can be defined as follows:

$$f(\theta|\alpha,\beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} \tag{2.1}$$

Notes on the pdf of the Beta distribution:

- The prior is using the *pdf* as a distribution. So we have to use the $f(\theta|\alpha,\beta)$.
- $B(\alpha,\beta) \neq Beta(\alpha,\beta)$. IT IS NOT THE SAME

Note how in the denominator we have the ***normalization (integration) constant*** which is simply just a constant that equals to the integral. Dividing by this constant makes the integral equal to 1. $\left(\frac{B(\alpha,\beta)}{B(\alpha,\beta)} = 1\right)$:

$$B(\alpha,\beta) = \int_0^1 \theta^{\alpha-1}\cdot(1-\theta)^{\beta-1}\,d\theta \;=\; \frac{\Gamma(\alpha)\cdot\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{2.2}$$

Where if we have to solve the integral then we would have to divide and multiply by the normalization constant in order to make it equal to 1.

Example of normalizing an integral:

Suppose you have the following situation for ***Bayesian Prediction***.

- **A product of a binomial and a Beta function**:

$$f(x|data) = \int f(x|\theta)f(\theta|data)\,d\theta$$

- And the following information:

$$X_{new}|\theta \;\approx\; Bin(12,\theta) = \binom{12}{x}\cdot\theta^x\cdot(1-\theta)^{12-x}$$
$$f(\theta|data) = \frac{\theta^{14-1}\cdot(1-\theta)^{8-1}}{Beta(14,8)}$$

Then this would amount to the following integral:

$$\begin{aligned}
P(X_{new}=x|data) &= \int \binom{12}{x}\cdot\theta^x\cdot(1-\theta)^{12-x}\cdot\frac{\theta^{14-1}\cdot(1-\theta)^{8-1}}{Beta(14,8)}\,d\theta \\
&= \binom{12}{x}\frac{1}{Beta(14,8)}\int_0^1 \theta^{(14+x-1)}(1-\theta)^{(8+12-x-1)}\,d\theta \\
&= \binom{12}{x}\frac{1}{Beta(14,8)}\int_0^1 \frac{Beta(14+x,8+12-x)}{Beta(14+x,8+12-x)}\theta^{(14+x-1)}(1-\theta)^{(8+12-x-1)}\,d\theta \\
&= \binom{12}{x}\frac{Beta(14+x,8+12-x)}{Beta(14,8)}\int_0^1 \frac{1}{Beta(14+x,8+12-x)}\theta^{(14+x-1)}(1-\theta)^{(8+12-x-1)}\,d\theta \\
&= \binom{12}{x}\frac{Beta(14+x,8+12-x)}{Beta(14,8)}\int_0^1 \frac{\theta^{(14+x-1)}(1-\theta)^{(8+12-x-1)}}{Beta(14+x,8+12-x)}\,d\theta \\
&= \boxed{\binom{12}{x}\frac{Beta(14+x,8+12-x)}{Beta(14,8)}}
\end{aligned}$$

Where the **integral turned to 1** since the *normalization constant* was introduced (multiply and divide by it). $(\alpha = 14 + x)$ and $(\beta = 8 + 12 - x)$:

$$\frac{1}{Beta(14 + x\ ,\ 8 + 12 - x)} \tag{2.3}$$

2.2 Poisson Distribution

The probability mass function (**pdf but for discrete**) is defined as follows:

$$f(x|\lambda) = \frac{\lambda^x}{x!} \cdot e^{-\lambda} \tag{2.4}$$

Where the **Expectation** and **Variance** can be found as:

$$E[\lambda] \ = \ Var[\lambda] \ = \ \lambda \tag{2.5}$$

2.3 Gamma Distribution

The probability density function (**pdf**) is:

$$f(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot \lambda^{\alpha-1} e^{-\beta\lambda} \tag{2.6}$$

Where an important note to take into account is the following integral of a gamma function, where one need to find the **normalization constant**:

$$\int_0^\infty \lambda^{\alpha-1} e^{-\lambda(\beta)} = \int_0^\infty \left(\frac{\Gamma(\alpha)}{\beta^\alpha} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \right) \cdot \lambda^{\alpha-1} e^{-\lambda(\beta)} = \frac{\Gamma(\alpha)}{\beta^\alpha} \tag{2.7}$$

2.4 Exponential Distribution

The Exponential distribution is simply just the Gamma distribution with $\alpha = 1$.

$$\text{Exponential}(\lambda) \ = \ \text{Gamma}(1, \lambda) \tag{2.8}$$

Where the **pdf** is as follows:

$$f(\lambda, \alpha, \beta) = \lambda \cdot e^{-\lambda x} \tag{2.9}$$

2.5 Gaussian Distribution

The probability density function (**pdf**) can be shown:

$$f(\theta|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\ e^{-(x-\mu)^2/(2\sigma^2)} \tag{2.10}$$

Given a Gaussian integral, with no normalization constant, we can solve it by **multiplying and dividing by the normalization constant**:

$$\begin{aligned} \int_{-\infty}^{+\infty} e^{-(x-\mu)^2/(2\sigma^2)}\ dx \ &= \ \frac{\sqrt{2\pi\sigma^2}}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{+\infty} e^{-(x-\mu)^2/(2\sigma^2)}\ dx \\ &= \sqrt{2\pi\sigma^2} \cdot \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}\ dx \\ &= \boxed{\sqrt{2\pi\sigma^2}} \end{aligned}$$

Where the **Expectation** is as follows:

$$E[N(\mu, \sigma^2)] = \mu \tag{2.11}$$

And the **Variance** is:

$$\text{Var}[N(\mu, \sigma^2)] = \sigma^2 \tag{2.12}$$

2.6 Binomial Distribution

The Binomial Distribution is usually used as the **likelihood** of our data $f(data|\theta)$ when we have an experiment with **successes** and **total number of trials**.

In general, the binomial distribution has the following probability mass function (pmf):

$$f(\theta) = \binom{n}{s} \cdot \theta^s \cdot (1-\theta)^{n-s} \qquad (\textbf{for s = successes}). \tag{2.13}$$

The experiment usually goes as follows:

Suppose we have an experiment for throwing a coin. It resulted that out of the 20 total times we threw the coin (total number of trials) only 8 times it resulted as heads.

Then we have the following (**successes = heads**):

$$f(\theta) = \binom{20}{8} \cdot \theta^8 \cdot (1-\theta)^{12} \qquad (\textbf{for s = successes}). \tag{2.14}$$

Where the *prior* is usually a **Beta distribution** which, in combination with the binomial (likelihood), produces another Beta distribution for the posterior. This means that the prior is a **conjugate prior** when this happens (posterior is the same distribution as the prior (beta) when we combine the likelihood and prior).

The **Expectation** and **Variance** is given as follows:

$$\mathbf{E}[Bin] = n \cdot \theta \tag{2.15}$$

$$\mathbf{Var}[Bin] = n \cdot p(1-p) \tag{2.16}$$

Chapter 2: Conjugate Distributions

Previously we had been using our **simplest conjugate model** as the *likelihood being a Bernoulli distribution.* And the *prior being a Beta*:

$$\begin{aligned}\text{likelihood} &= \underline{x}|\theta \approx \text{Bernoulli}(\theta) = C_s^n \cdot \theta^k \cdot (1-\theta)^{n-k} \\ \text{prior} &= f(\theta) \propto \theta^{\alpha-1} \cdot (1-\theta)^{\beta-1}\end{aligned}$$

To then produce a posterior of the same distribution as the prior (conjugate prior):

$$\theta|\underline{x} \approx B(\alpha + k, \beta + n - k) \tag{3.1}$$

Therefore we have the equation above which just gives us a quick way of calculating this using the above equation.

Now, the following distributions for the prior will be used (**conjugate prior in this case**):

- Geometric
- Negative Binomial
- Binomial
- Bernoulli

3.1 Beta-Binomial Distribution

$$Pr(X = x|data) = \binom{m}{x} \frac{B(a^* + x, b^* + m - x)}{B(a^*, b^*)} \tag{3.2}$$

Notice how here, we have it as a function of x. So then we dont directly have a number until we get x.

Where $a^* = a + k$ and $b^* = b + n - k$. Which produces the expectation and variance as:

$$E[X|\underline{x}] = \frac{m \cdot a^*}{a^* + b^*} \tag{3.3}$$

$$V[X|\underline{x}] = \frac{m \cdot a^*\ b^*(a^* + b^* + m)}{(a^* + b^*)^2 \cdot (a^* + b^* + 1)} \tag{3.4}$$

3.2 Coin Tossing Revisited

Example:

You have a coin with $Pr(head) = \theta$. Suppose you decide to toss the coin 12 times and observe 9 heads and 3 tails. Suppose we use a Beta(5,5) prior distribution for θ.

1. Obtain the posterior distribution of θ:

The posterior ($f(\theta|\ \underline{x})$) can be calculated as the following:

$$\begin{aligned}\theta|\underline{x} &\propto \text{likelihood} \cdot \text{prior} \\ &\propto f(\underline{x}|\theta) \cdot f(\theta) \\ &\propto \text{Binom}(9,3) \cdot \text{Beta}(5,5) \\ &\propto \text{Beta}(9+5, 3+5) \\ &= \text{Beta}(14, 8)\end{aligned}$$

2. Obtain the predictive distribution, expectation and variance for the number of heads in 10 more tosses.

Now we can apply the formula (m = 10 since there are 10 more tosses, $a^* = a + k = 14, b^* = 8$):

$$\begin{aligned} Pr(X = x|data) &= \binom{m}{x} \frac{B(a^* + x, b^* + m - x)}{B(a^*, b^*)} \\ &= \boxed{\binom{10}{x} \frac{B(14 + x, 8 + 10 - x)}{B(14, 8)}} \end{aligned}$$

- **Expectation:**

$$\begin{aligned} E[X|\underline{x}] &= \frac{m \cdot a^*}{a^* + b^*} \\ &= \boxed{\frac{10 \cdot 14}{14 + 8}} \end{aligned}$$

- **Variance:**

$$\begin{aligned} V[X|\ \underline{x}] &= \frac{m \cdot a^*\ b^*(a^* + b^* + m)}{(a^* + b^*)^2 \cdot (a^* + b^* + 1)} \\ &= \boxed{\frac{10 \cdot (14)(8)(14 + 8 + 10)}{(14 + 8)^2 \cdot (14 + 8 + 1)}} \end{aligned}$$

3. Obtain the predictive distribution for the number of tosses until a head is observed

- Done in R.

3.3 Objective Priors

Objective Priors are simply just those priors that don't skew the posterior. Basically when **we don't have much information about the data**. This means that we can use, for example:

- Uniform prior
- Jeffreys prior
- Reference priors
- ...

We can also consider an **improper prior**. This means the prior that has its integral = infinity.

$$\int_{\mathbb{R}} f(\theta)\ d\theta = \infty\ \neq\ 1$$

This can only be done **if and only if** the *posterior is proper* (it is a conjugate prior).

3.4 Conjugate priors for 'rare events' models

Usually, the Poisson is used to model, for example, **number of arrivals per hour**. We must always check that the data follows a Poisson distribution (before assuming).

This can be done if the mean and variance of our data are the same!

3.4.1 Worksheet 3 Problem 2

Suppose we are interested in estimating the mean arrival of customers to a shop per hour. Assume that the distribution of number of arrivals per hour is constant along time and follows a Poisson distribution. Given that we have observed 27 arrivals in 7 hours, answer the following questions.

(a) Define a gamma prior distribution describing your prior beliefs about expected number of arrivals per hour, λ. Given the observed data, obtain the posterior distribution of this expectation.

In this case our gamma prior distribution will simply be just whatever we believe prior to looking at the data. We are told to put a gamma prior distribution so we just need to assign an a and b.

Gamma Prior:

$$\lambda \approx \lambda^{a-1} \cdot e^{-b\lambda}$$

Whenever we have a rate of something over a given time period n, we use a Poisson Distribution:

$$\underline{X}|\lambda \approx \lambda^{\sum x_i} e^{-n\lambda}$$

Therefore, our posterior is:

$$\lambda|\underline{x} \approx \lambda^{a+\sum x_i - 1} e^{(-\lambda[b+n])}$$

Choosing our prior to be a Gamma(2,1) and our likelihood to be a Poisson(27,7). Then our Posterior is a **Gamma(2+27,7+1)**

(b) Obtain the Posterior mean, median and MAP

In order to obtain the posterior median, we must do the R command:

```
qgamma(0.5, 2+27, 7+1)
```

Then if we want to find a **95% credible interval**, we can use the qgamma in order to find the number where the area of the distribution is equal to a certain number:

Where the 2+27 and the 7+1 is simply just our new coefficients for the posterior (found from the prior and likelihood).

```
# Value at which we have area of 0.025.
a = qgamma(0.025, 2+27, 7+1)
# Value at which we have area of 0.0975.
b = qgamma(0.975, 29, 8)
```

If we want to perform a **Bayesian Hypothesis test** to decide if for the expected number of arrivals (λ) per hour is larger or smaller than 3 then we must do the hypothesis test and find the probability of the Null hypothesis.

$$H_0 : \lambda \leq 3$$
$$H_1 : \lambda > 3$$

Therefore, finding the probability of H_0 occurring, is by doing the following R command:

```r
# Null hypothesis of lambda <= 3
pgamma(lambda = 3, 29, 8)
```

If we want to obtain the **posterior predictive distribution**:

$$\int P(\underline{x}\,|\theta) \cdot P(\theta|\underline{x})$$

Where the first $P(\underline{x}|\theta)$ is done according to the information provided. For example, if we wanted to predict that the next 10 people arrive in an hour, then we would use the Poisson again for n = 1, $\sum x_i = 10$.

3.5 Distributions belonging to the Exponential family

A distribution is a part of the exponential family of distributions if it adheres to this form:

$$f(x|\theta) = g(\theta) \cdot h(x) \cdot \exp(\eta(\theta) \cdot T(x)) \tag{3.5}$$

Where the **conjugate prior** can be obtained as:

$$f(\theta) = g(\theta)^a \cdot \exp(b \cdot \eta(\theta)) \tag{3.6}$$

The Posterior can be obtained as well:

$$f(\theta|\underline{x}) = g(\theta)^{a^*} \cdot \exp(b^* \cdot \eta(\theta)) \tag{3.7}$$

For $a^* = a + n$ and $b^* = b + n \cdot \bar{T}$.

4. Computational Bayesian Analytics (MCMC)

The following are included in this chapter. Also the code belonging to each of them that can be generalized by the same repeated code.

1. **Numerical Integration**.
2. **Importance Sampling.**
3. **Rejection Sampling.**
4. **Monte-Carlo Markov Chain (MCMC).**
5. **Metropolis - Hastings. Random Walk Algorithm.**
6. **Gibbs Sampling.**

4.1 Numerical Integration

```
# a. Use numerical Integration to approximate the normalizing constant and the posterior
# mean. Plot the posterior density.

# Here we are making a probability function for the posterior.
# Where we can integrate over 0 -> 1 to approximate the normalization constant.
# We normalize this by integrating from 0 -> 1, else it would not be normal
tosses <- 12
heads <- 9
tails <- tosses - heads
fprop <- function(theta,heads,tails){
  f <- dexp(theta)*(theta^heads)*((1-theta)^tails)
  return(f)
}
# Integrating for the normalization constant.
c <- integrate(fprop,lower=0,upper=1,heads=heads,tails=tails)$value
c
grid = seq(0,1,0.01)
plot(grid, fprop(grid,heads,tails)/c,type="l")

# Finding the mean of the posterior.
# This is just the -->  integral of the posterior * theta.
# We divide by c since we want the posterior to be normalized.
fmean <- function(theta,heads,tails,c){
  f <- theta * dexp(theta)*(theta^heads)*((1-theta)^tails)/c
  return(f)
}

# Integrating to find the posterior mean.
post.mean <- integrate(fmean,lower=0,upper=1,heads=heads,tails=tails,c)$value
post.mean
```

4.2 Importance Sampling

```
# Importance Sampling
N <- 100000
theta <- runif(N)
w <- fprop(theta,heads,tails)/1
c.est <- mean(w)
c(c,c.est)
# Finding the mean.
mean.est = mean(w*theta)/c.est
c(mean.est,post.mean)

# Sampling Importance Resampling
N2=N
thetapost <- sample(theta,size=N2,replace=TRUE,prob=w)
hist(thetapost,probability=TRUE,xlab=expression(theta),ylab='f',main='')
lines(grid,fprop(grid,heads,tails)/c,lwd=2,col='green')
```

4.3 Rejection Sampling

```
# Rejection Sampling.
M <- max(fprop(theta=grid,heads,tails))
N <- 100000
theta <- runif(N)
pac <- fprop(theta,heads,tails)/(M*1)
u <- runif(N)
thetaac <- theta[u<pac]
Nac <- length(thetaac)
hist(thetaac,probability=TRUE,xlab=expression(theta),ylab='f',main='')
lines(grid,fprop(grid,heads,tails)/c,type='l',lwd=2,col='green')
Nac/N
c(post.mean,mean(thetaac))
```

4.4 Monte-Carlo Markov Chain (MCMC)

4.5 Metropolis - Hastings. Random Walk Algorithm

Note that we can do the log of the fprops of the proposal theta and the current value of the parameter we are trying to sample from the posterior distribution. Note that this is the Random Walk algorithm therefore we can compute the acceptance probability by the following:

$$\alpha(\theta_t, \theta^{'}) = min \left\{ \text{unif}(1) \ , \ \frac{f(\theta^{'}) f(data|\theta^{'}))}{f(\theta_t) f(data|\theta_t)} \right\}$$

$$\text{(taking logs)} \qquad = min \left\{ log(\text{unif}(1)) \ , \ log(\text{proposal}) - log(\text{current value}) \right\}$$

```
burnin = 1000
iters = 10000
totits = burnin+iters
thetapost = rep(NA,iters)
theta = 0.5 # initial value for theta
pac = 0

for (i in 2:totits){

  # We are sampling the proposal theta from the normal distribution
  thetac <- rnorm(1,theta,sd=0.5)

  # If the proposal theta is within 0 and 1.
  # It is a correct value to use.
  if(thetac>0 & thetac<1){

    s# Doing the log to simplify ! LOOK AT THE ABOVE NOTES.
    logal=log(fprop(thetac,heads,tails))
    logal=logal-log(fprop(theta,heads,tails))

    # Randomly obtained value between 0 and 1
    u=runif(1)

    # if the log of u is less than the logal
    # accept this value
    if (log(u)<logal){
      theta=thetac; if (i>burnin){pac=pac+1}
    }
  }
  # If the iterations are after the burnin
  # period then we should include them in the list of thetas.
  if (i>burnin){
    thetapost[i-burnin] <- theta
  }
}
pac = pac / iters
pac
```

4.6 Gibbs Sampling

```
x <- c(254, 249, 252, 252, 249, 249, 250, 247, 251, 252)
hist(x,freq=F)
m=250; c=0.01; a=0.01; b=0.01;
n=length(x)
xbar=mean(x)
s2=var(x)
burnin = 1000
iters = 10000
totits = burnin+iters
mupost = rep(NA,iters)
taupost = rep(NA,iters)
xpred = rep(NA,iters)
mu = 0
for (i in 1:totits){

  tau <- rgamma(1,0.5*(a+n),0.5*(b+(n-1)*s2+n*(mu-xbar)^2))
  mu <- rnorm(1,(c*m+n*tau*xbar)/(c+n*tau),1/sqrt(c+n*tau))


  if (i>burnin){
    mupost[i-burnin] <- mu
    taupost[i-burnin] <- tau
    xpred[i-burnin] <- rnorm(1,mu,1/sqrt(tau))

  }
}
```

5. Bayesian Linear Regression

For Bayesian estimation we do the usual, **likelihood** · **prior** $\propto$ **posterior**.

This time we have to also get priors and hence likelihoods for **both** the beta coefficients (β) and also the variance (σ^2).

$$\beta \sim N(m, V) \tag{5.1}$$

$$\sigma^2 \sim IG(\frac{a}{2}, \frac{b}{2}) \tag{5.2}$$

Therefore, we ultimately get **three distributions**, for sigma, beta, and our intercept.

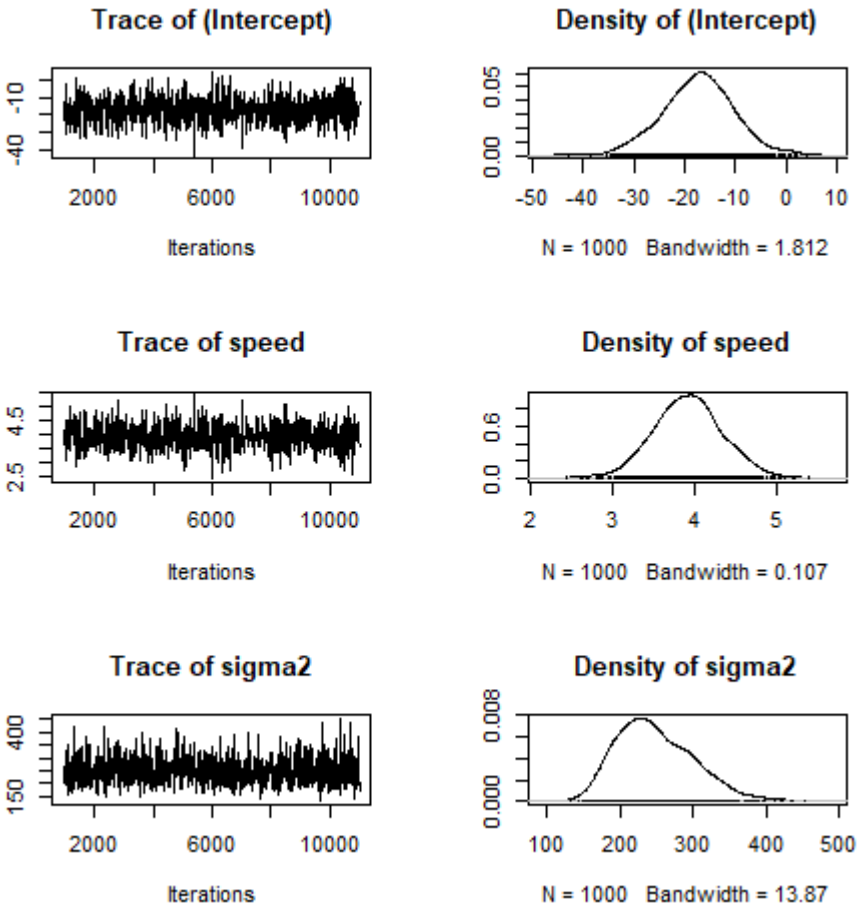
```
# MCMCregress for a bayesian linear regression model using
# MCMC sampling.
bayes.model <- MCMCregress(dist ~ speed, data=cars, thin=10,  burnin= 1000, mcmc = 10000)
summary(bayes.model)

# thin -- only every nth sample is taken (n=thin)!
# burnin -- burnnin iterations. The iterations we remove before we acually include the values.
# (the first 1000 will be removed).
# mcmc -- number of iterations.
```

If we plot the new model we can see the **trace plot** as well as the resulting **posterior** for the estimates.

```
# plotting the densities and trace plots
plot(bayes.model)
```

Something important to know about this plot:

- We don't have point estimates on our betas. But rather a posterior density of values that these coefficients can take (we can sample from this).
- **Trace Plot:** We want to see a large range of values (like is shown in the above figure) so that we know that the sampling done includes a wide range of values.
- The trace plot essentially shows the values that were sampled for each of the variables.
- *If we have a bad trace plot, we may want to increase the number of iterations, burning period, and maybe reduce thinning.*
- If we have very **high auto-correlation** we may need to increase thinning. ***High auto-correlation means our data points are not independent.***



We can also see the 95% quantiles for our variables. It shows where, 95% of the time our values fall.

```
apply(bayes.model, 2,quantile, probs = c(0.025,0.975))
```

5.0.1 R2OpenBugs:

For the R2OpenBugs model we must always do the same thing.

Implement the target variable and the relation with the betas / predictors. Then obtain priors for the betas. Initialize the variables (all the betas). And make the list of data (n, our predictors and the new target variable). Then we need to make the output using the bugs function. Include in this function the data, inits, and parameters to save, as well as the model file number of chains and iterations. **Take into account these considerations!**

- **We CANNOT use = in the R2OpenBugs model, we must use either '<-' or ~ .**
- Using = for initializing variables will produce errors.

```
library(R2OpenBUGS)
lm.bayes <- function(){

  # specifying the model: how the data and parameters are
  # related.
  for( i in 1 : n ) {
    # observed distances are normally distributed.
    dist[i] ~ dnorm(mu[i], phi)
    # mean is a linear function of speed.
    mu[i] <- beta0 + beta1 * speed[i]
  }

  # PRIORS
  beta0 ~ dnorm(0.0, 1.0E-6)
  beta1 ~ dnorm(0.0, 1.0E-6)
  phi ~ dgamma(0.001, 0.001)
  sigma2 <- 1 / phi

  # POSTERIOR PREDICTION.
  dist.new ~ dnorm(mu.new, phi)
  mu.new <- beta0 + beta1 * speed.new
}
```

Listing 1: Simple R2OpenBugs code for linear regression.

```r
# Defining the MODEL: We must include the target variable and its relation to the predictors.
lm.bayes <- function(){
  for( i in 1 : n ) {

    # We are obtaining a distribution for the betas, mu and sigma
    # Target variable is a normal with mu and phi.
    stack.loss[i] ~ dnorm(mu[i], phi)
    # phi is a dgamma

    # Defining the mu as the linear equation.
    mu[i] <- beta0 + beta1 * Air.Flow[i] + beta2* Water.Temp[i] + beta3* Acid.Conc.[i]

  }

  # Priors
  beta0 ~ dnorm(0.0, 1.0E-6)
  beta1 ~ dnorm(0.0, 1.0E-6)
  beta2 ~ dnorm(0.0, 1.0E-6)
  beta3 ~ dnorm(0.0, 1.0E-6)
  phi ~ dgamma(0.001, 0.001)
  sigma2 <- 1 / phi

  # Predictive posteriors.
  stack.loss.new ~ dnorm(mu.new, phi)
  mu.new <- beta0 + beta1 * Air.Flow.new + beta2* Water.Temp.new + beta3* Acid.Conc.new
}

# Getting the size of our target variable.
n = length(stackloss$stack.loss)

# Initializing the data to be used from out dataset.
data <- list(n=n, stack.loss = stackloss$stack.loss, Air.Flow = stackloss$Air.Flow,
            Water.Temp = stackloss$Water.Temp,
            Acid.Conc. = stackloss$Acid.Conc.,
            Air.Flow.new = 60,
            Water.Temp.new = 20, Acid.Conc.new=80)

# Initializing the beta values and phi, and the new stackloss (this is for convergence)
inits <- function(){
  list(beta0 = 1, beta1 = 0,  beta2 = 0, beta3 = 0, phi = 0.5, stackloss.new = 25)
}


# Output of Bugs note the parameters to save uncludes the mu.new and stack.loss.new
output <- bugs(data = data, inits = inits, parameters.to.save = c("beta0", "beta1","beta2","beta3","sigma2",
        "mu.new", "stack.loss.new" ), model.file = lm.bayes, n.chains = 1, n.iter = 10000)

print(output)
```

5.0.2 Bayesian (MCMC) Logistic Regression

In Bayesian Logistic regression we admit that the target variable follows a Bernoulli Distribution:

$$Y \mid p \sim Bernoulli(p)$$

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{j=1}^{k} \beta_j \cdot x_j$$

Therefore, when using R2OpenBugs we must include the bernoulli distribution for the target variable and also the logit for the betas.

The following code shows a simple yet general example that can be followed easily.

```
library(R2OpenBUGS)

logit.bayes <- function(){
  for( i in  1 : n ) {
    # target variable is a bernoulli
    admit[i] ~ dbern(p[i])
    # We then get the logit of the probability with the betas.
    logit(p[i]) <- beta0 + beta1 * gre[i] + beta2 * gpa[i]+ beta3*rank[i]
  }
  # Prior distributions for the betas (no sigma here so no prior for sigma ...)
  beta0 ~ dnorm(0.0, 1.0E-6)
  beta1 ~ dnorm(0.0, 1.0E-6)
  beta2 ~ dnorm(0.0, 1.0E-6)
  beta3 ~ dnorm(0.0, 1.0E-6)
  }
# same getting the length
n=length(x$admit)

# setting up the data from our dataset
data <- list(n=n, admit=x$admit, gre=x$gre, gpa=x$gpa, rank=x$rank)

# initializers
inits <- function(){
  list(beta0 = 1, beta1 = 0, beta2 = 0, beta3 = 0)
}

# output and save just betas.
output <- bugs(data = data, inits = inits, parameters.to.save = c("beta0", "beta1", "beta2", "beta3"),
         model.file = logit.bayes, n.chains = 1, n.burnin=100, n.iter = 1000)
print(output)
```

Listing 2: R2OpenBugs code for a logitic regressor using MCMC.

If we are using a Poisson regression model to find the rate instead of a binary then we simply have to make our target variable be the : **dpois()** *instead of dbern()*.