

Chapter 8

Bayesian statistics

Please read Chapter 17 from *Learning Statistics with R* before starting these assignments.

<i>Content</i>	<i>Page</i>
Assignment 8.1: Specifying and interpreting a prior distribution	90
Assignment 8.2: Updating a prior distribution to a posterior distribution	92
Assignment 8.3: Finding the posterior distribution using MCMC sampling	94
Assignment 8.4: Comparing simple models using the Bayes factor	97
Assignment 8.5: Using bridge sampling to calculate the Bayes factor	100
Assignment 8.6: Performing a Bayesian linear regression	102

Chapter 8: Bayesian statistics

Learning objectives of this chapter:

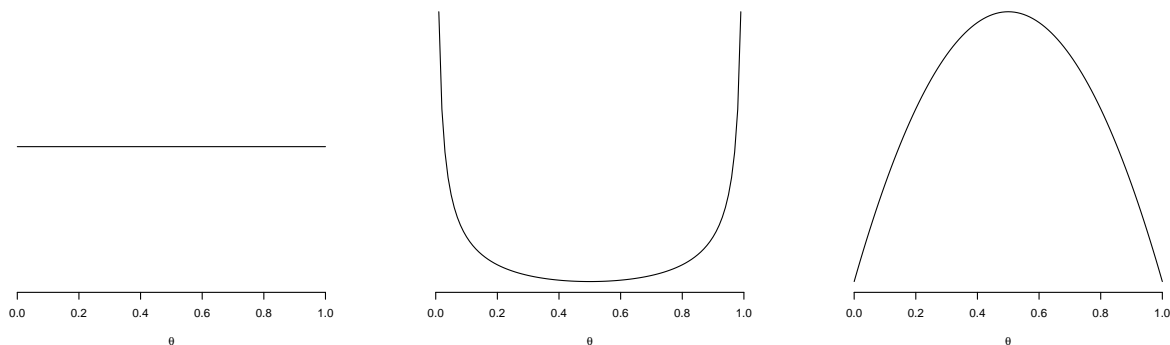
- Specifying and interpreting a prior distribution
- Updating a prior distribution to a posterior distribution
- Quantifying evidence for a model using the Bayes factor

Assignment 8.1: Specifying and interpreting a prior distribution



In this final chapter we move away from the traditional statistical methods that you have learned (**confidence intervals** and **p-values**), and introduce a different methodology; Bayesian statistics. Bayesian statistics expresses the most basic idea in learning, namely, that your prior belief in an event is updated to a posterior belief after observing data. By first specifying your prior beliefs, you allow yourself to learn from the data that you have collected, thereby updating your prior belief to a more informed posterior belief. Bayesian statistics is a powerful tool that can help you to update your existing beliefs in the context of new business information.

In Bayesian inference, your prior beliefs about a **parameter** are expressed through a **probability distribution** . For example, in tossing a coin, the probability of heads occurring may be defined as a **parameter** θ . Your initial probability distribution (i.e., beliefs) on θ is called the **prior distribution** , which can be interpreted quite visually. The figure below shows three possible prior distributions for the probability of heads in a coin toss. The area under the curve represents probability.



- 8.1 a) Interpret the three **prior distributions** above and discuss what information about the probability θ of the coin occurring on heads they incorporate.

Answer 8.1a:

When estimating a proportion (like the probability of heads in a coin toss) you can use the common $Beta(\alpha, \beta)$ distribution as a prior distribution on θ since it is restricted to the range $[0, 1]$. The $Beta(\alpha, \beta)$ distribution has two parameters, α and β , that have an effect on its shape. By playing around with these parameters, you can create the **prior distribution** that reflects your particular beliefs as accurately as possible.

Run the following code in R to create the **prior distribution** shown on the left of the figure above:

```
alpha <- 1
beta <- 1
curve(dbeta(x, alpha, beta), xlab = expression(theta), ylab = '', yaxt = 'n')
```

- 8.1 b) Recreate the middle and right **prior distributions** from the figure by changing the values for **alpha** and **beta** and running the code again. What are the values of α and β for these distributions?

Answer 8.1b:

Middle:

Right

α : _____

α : _____

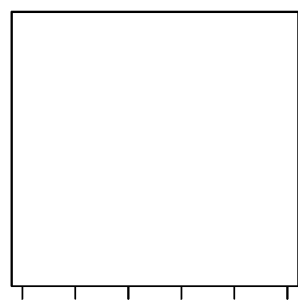
β : _____

β : _____

- 8.1 c) How would you define a **prior distribution** for θ if you believed the coin was definitely biased towards heads? And for one that is biased towards tails? Draw these **prior distributions** below.

Answer 8.1c:

Biased towards heads:



0.0 0.2 0.4 0.6 0.8 1.0

Tails----- θ ----- Heads

Biased towards tails:



0.0 0.2 0.4 0.6 0.8 1.0

Tails----- θ ----- Heads

Assignment 8.2: Updating a prior distribution to a posterior distribution

Suppose you are the manager of a small store and want to estimate what proportion of your customers leaves the store with a feeling of satisfaction. You give out a questionnaire to 40 people in the store and ask them whether they felt satisfied or not. In a previous enquiry done by you, it was already shown that, on average, 60% of your customers feels good about the store, with a **standard deviation** of ten percent. Using Bayesian statistics, you are now going to update the information from your previous enquiry with the information from the new questionnaire.

8.2 a) What **parameter** does θ represent in this scenario?



Hint 8.1: Think about what question you are interested in answering.

Answer 8.2a:

8.2 b) Think of your own α and β parameters for the $Beta(\alpha, \beta)$ **prior distribution** on θ . Take into account the average percentage from your previous enquiry and incorporate this into your **prior distribution**.

Answer 8.2b:

α : _____ β : _____

8.2 c) Use the **curve()** function to create a figure of your **prior distribution** in R.

By updating the $Beta(\alpha, \beta)$ prior distribution with information that you have collected, you create a **posterior distribution**, which contains both the prior information and the information from the sample. Having observed k successes in a sample of n observations, the **prior parameters** α and β are updated to the **posterior parameters** $\alpha = \alpha + k$ and $\beta = \beta + n - k$.

In your sample of $n = 40$ questioned customers, $k = 33$ said they felt satisfied when leaving the store.

8.2 d) Write down the α and β **parameters** of your **posterior distribution**.

Answer 8.2d:

α : _____ β : _____

Run the following code in R to create a figure of the **posterior distribution**. Fill in your own values of **alpha** and **beta** from assignment 8.2b.

```
alpha <- 7
beta <- 5
n <- 40
k <- 33

curve(dbeta(x, alpha + k, beta + n - k),
      xlab = expression(theta), ylab = '', yaxt = 'n')
```

- 8.2 e) Find out the **posterior probability** that more than 60 percent of your customers left the store feeling satisfied, given your prior information and the information from the sample.



Hint 8.2: Use the `qbeta()` function to find the cumulative probability under a beta distribution.

Answer 8.2e:

Probability: _____

- 8.2 f) Find out the **posterior probability** that the percentage of customers that left the store feeling satisfied, given your prior information and the information from the sample, lies between 70 and 90 percent.

Answer 8.2f:

Probability: _____

- 8.2 g) Change your values of `alpha` and `beta` in the code above so that you have a different **prior distribution** and run the code again. Describe how robust the **posterior distribution** is to changes in the prior distribution.

Answer 8.2g:

Assignment 8.3: Finding the posterior distribution using MCMC sampling

In assignment 8.2 you have seen that updating a **prior distribution** to a **posterior distribution** is easy when you are counting successes and failures. In this scenario the **prior distribution** and **posterior distribution** are in the same family, they are both *Beta* distributions. However, many posterior distributions are not so easy to find. In this assignment, you are going to learn how to use a Markov chain Monte Carlo sampling method to approximate the posterior distribution for any combination of **prior distribution** and **likelihoods**.

For this exercise you will be using the Stan coding language with the **rstan** R package. **rstan** performs MCMC sampling using an Hamiltonian Monte Carlo (HMC) algorithm, which is known to have a very high quality in finding the **posterior distribution**. Install and load the package using:

```
install.packages(rstan)
library(rstan)
```

Stan requires very specific commands. It requires that you specify all **data**, all **parameters**, and the complete statistical **model**³.

Copy and run the following code in R to set up the Stan model for the scenario in exercise 8.2b and store it in an object called **modelCode**. The code then compiles the model using the **stan_model()** function and starts sampling using the **sampling()** function. The result of the sampling is stored in the **stanFit** object.



*Hint 8.3: You can change the **theta ~ beta(7, 5)** part of the code to specify your own prior distribution from assignment 8.2b.*

```
modelCode <- '
data {
  int n;
  int k;
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(7, 5);
  k ~ binomial(n, theta);
}
'

# Note: The following line can take a while to execute
compiledModel <- stan_model(model_code = modelCode, model_name = 'model')

stanFit <- sampling(compiledModel, data = list(n = 40, k = 33),
  iter = 5000, warmup = 500, chains = 4)
```

³For a complete introduction to Stan, you can visit <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

- 8.3 a) Extract the samples of the **posterior distribution** on θ from the `stanFit` object using the `extract()` function and store them in an object called `samples`.
- 8.3 b) Create a histogram of the `samples`. Next, run the following code in R to plot the analytical **posterior distribution** over the histogram. Do the samples of the **posterior distribution** resemble the actual **posterior distribution** in any way?

```
curve(dbeta(x, shape1 = 40, shape2 = 12), add = TRUE)
```



Hint 8.4: Be sure to specify a larger value for the `breaks` argument since you have many samples. Also set `probability = TRUE` in the `hist()` function.

Answer 8.3b:

As you have seen in assignment 8.2, the **posterior distribution** you just estimated can be calculated by hand, since it is a *Beta* distribution. However, to represent your prior information (remember: you have conducted a questionnaire that showed you 60 percent of customers left your store satisfied with a **standard deviation** of 10 percent), let's give θ a $Normal(\mu, \sigma)$ distribution as a prior distribution. You use your prior estimate of θ ($p = 0.6$) as the **mean** of the prior distribution, and the **standard deviation** of the sample ($s = 0.10$) as the **standard deviation** of the **prior distribution**.

- 8.3 c) Change the prior distribution in the `modelCode` to the *Normal* distribution described in the text above. Run all appropriate code again.



Hint 8.5: The line `theta ~ beta(7, 5)` specifies the prior distribution.

- 8.3 d) Again, find out the **posterior probability** that more than 60 percent of your customers left the store feeling satisfied, given the prior information and the information from the sample.



Hint 8.6: Do not use the `quantile()` function or the `pbeta()` function.

Answer 8.3d:

Probability: _____

- 8.3 e) Find out the **posterior probability** that the percentage of customers that left the store feeling satisfied, given the prior information and the information from the sample, lies between 70 and 90 percent.

Answer 8.3e:

Probability: _____

- 8.3 f) Do your answers for assignment 8.3d and assignment 8.3e differ substantially from your answers at assignment 8.2e and assignment 8.2f. What does this tell you about the robustness of your outcomes to the choice of **prior distribution**?

Answer 8.3f:

Assignment 8.4: Comparing simple models using the Bayes factor

Bayesian statistics does not use the **p-value** to test hypotheses and yield conclusions in an all-or-none fashion. Instead it uses a continuous measure of evidence, the **Bayes factor**. The **Bayes factor** quantifies the relative predictive performance of two competing hypotheses; the **null hypothesis** H_0 and the **alternative hypothesis** H_1 . The subscript in the **Bayes factor** indicates for which hypothesis support is quantified. BF_{10} indicates the Bayes factor in favor of H_1 over H_0 , whereas BF_{01} indicates the **Bayes factor** in favor of H_0 over H_1 . Larger values of BF_{10} indicate more support for H_1 . A **Bayes factor** can range from 0 to ∞ , and a Bayes factor of 1 indicates that both hypotheses are supported equally well by the data. A $BF_{10} = 10$ indicates that the data is 10 times more likely to occur under H_1 than under H_0 . Given this interpretation, the **Bayes factor** is a continuous measure for the strength of evidence.

Suppose you are a data scientist at a stock trading company. You have written an algorithm that trades stocks automatically. Your algorithm either makes a profitable trade, or a losing trade. You want to make sure that your algorithm actually does something intellectual, and therefore want to disprove the hypothesis that your algorithm makes a profitable trade with a probability other than if it were to trade stocks randomly.

- 8.4 a) Specify the **null hypothesis** H_0 and the **alternative hypothesis** H_1 for the test of a proportion θ against a chance value.

Answer 8.4a:

H_0 : _____

H_1 : _____

- 8.4 b) Specify a *Beta* prior distribution for the **parameter** θ . Let your prior distribution depend on how much confidence you have in your programming skills.



Hint 8.7: Think about what the θ represents in this case.

Answer 8.4b:

$$\theta \sim \text{Beta}(\quad , \quad)$$

You monitor the algorithm for 30 trades, and each time mark a trade as a success (profit) or a failure (loss). After data collection you see that, of the 30 trades, your algorithm made 22 profitable trades.

- 8.4 c) What are the α and β parameters of your **posterior distribution**?

Answer 8.4c:

α : _____

β : _____

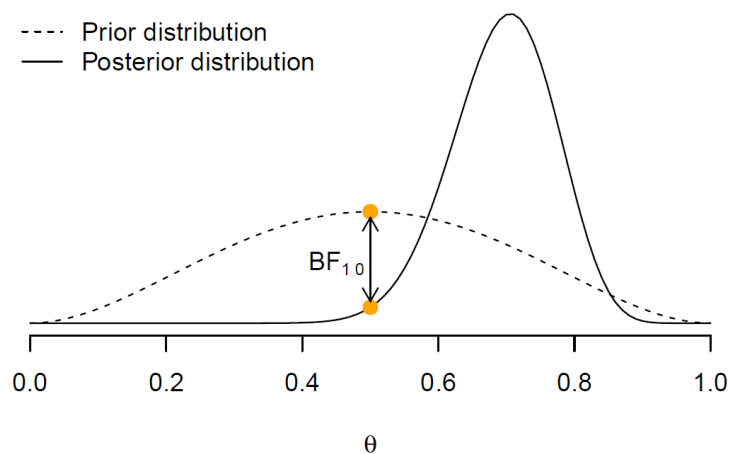
8.4 d) Use the `curve()` function to create a figure showing both the prior and the posterior distribution.



Hint 8.8: The `curve()` function has an argument `add` that controls whether to add a line to an already existing plot.

In this scenario, the **Bayes factor** is intuitively to calculate by only looking at the posterior distribution for H_1 . The **Bayes factor** BF_{10} tells you how much more likely the **alternative hypothesis** H_1 is in comparison with to the point **null hypothesis** H_0 . The **Bayes factor** BF_{10} in favor of the unrestricted **alternative hypothesis** $H_1 : \theta \neq 0.5$ is simply the height of the **posterior distribution** at the point $\theta = 0.5$ divided by the height of the **prior distribution** at the point $\theta = 0.5$.

Please note that this is a trick that only works for **nested** hypotheses and models, in which only one of the **parameters** is fixed to a specific value.



8.4 e) Use the `dbeta()` function to find out the height of your **prior distribution** at the point $\theta = 0.5$ and store it in an object called `heightPrior`.

8.4 f) Use the `dbeta()` function to find out the height of your **posterior distribution** at the point $\theta = 0.5$ and store it in an object called `heightPosterior`.

Run the following code in R to compute the **Bayes factor** in favor of the **alternative hypothesis** H_1 .

```
BF10 <- heightPrior / heightPosterior
```

8.4 g) What is the value of the **Bayes factor** `BF10`? Interpret this **Bayes factor** with respect to the alternative hypothesis H_1 .

Answer 8.4g:

To make interpretation of the **Bayes factor** more simple, the labels in the table below have been proposed. Please note that you should not get too hung up on the specific labels and breakpoints in this table, since what constitutes convincing evidence should be different in every situation.

Bayes factor	Evidence
1 - 3	Anecdotal
3 - 10	Moderate
10 - 30	Strong
30 - 100	Very strong
> 100	Extreme

- 8.4 h) Considering the labels from the table above, are you convinced enough to reject the hypothesis that your algorithm makes a profitable trade with a probability other than if it were to trade stocks randomly?

Answer 8.4h:

- 8.4 i) Re-specify the α and β parameters of the **prior distribution** so that you express a different prior belief. Recalculate the **Bayes factor** by running your code again. What is the new value of **BF10** . How robust is your **Bayes factor** to changes in the prior distribution?

Answer 8.4i:

Assignment 8.5: Using bridge sampling to calculate the Bayes factor

Suppose that you are not convinced by the evidence from assignment 8.4 and want to perform additional testing. This time, however, you want to find out whether your algorithm makes more than 75 percent of the trades with profit. That is, you want to test whether θ is higher or lower than 0.75.

- 8.5 a) Formulate the **models** M_1 (restricts θ to be below 0.75) and M_2 (restricts θ to be higher than 0.75). Choose the restriction (\leq , $=$, or \geq) that applies to each model.

Answer 8.5a:

$M_1: \theta \leq / = / \geq \quad ______$

$M_2: \theta \leq / = / \geq \quad ______$

For these more complicated (e.g., non-nested) models, the computation of the **Bayes factor** becomes more difficult since it involves computing the **marginal likelihoods** of the **models**. To make your life simple, you can use the **bridgesampling** package together with the **rstan** package to let R calculate these values for you.

Run the following code in R to install the **bridgesampling** package. Then load the **bridgesampling** and the **rstan** package.

```
install.packages('bridgesampling')

library(bridgesampling)
library(rstan)
```

Copy and run the following code in R to set up the Stan model for M_1 , the model that restricts θ to be lower than 0.75 and store it in an object called **model1**. The code then compiles the model using the **stan_model()** function.

```
model1code <- '
data {
  int n;
  int k;
}
parameters {
  real<lower=0,upper=0.75> theta;
}
model {
  theta ~ beta(1, 1)T[0, 0.75];
  k ~ binomial(n, theta);
}
'

# Note: The following line can take a while to execute
model1 <- stan_model(model_code = model1code, model_name = 'model1')
```

- 8.5 b) Create the Stan model for M_2 , the model that restricts θ to be higher than 0.75, on the basis of the lines in `model1code`. Name the compiled model `model2`.



Hint 8.9: First find out what elements were added to the code since assignment 8.3.

- 8.5 c) Locate and describe the **prior distributions** for θ in M_1 and M_2 .

Answer 8.5c:

Now you start collecting data again. Suppose that you monitor your algorithm for $n = 156$ more trades, and find that $k = 123$ trades resulted in a profit.

- 8.5 d) Use the `sampling()` function to sample for the **models** M_1 and M_2 using these data. Store the fitted Stan models in objects named `stanFitM1` and `stanFitM2`.
- 8.5 e) Use the `bridge_sampler()` function to calculate the **marginal likelihoods** of the models M_1 and M_2 and store them in `mLike1` and `mLike2`.
- 8.5 f) Use the `bf()` function to calculate the **Bayes factor** BF_{21} in favor of model M_2 over model M_1 and store it in an object called `BF21`.



Hint 8.10: Be sure to insert the two marginal likelihoods into the `bf()` function in the correct order so that the result is BF_{21} .

- 8.5 g) What is the value of `BF21`? Interpret this **Bayes factor** with respect to the model that restricts θ to be higher than 0.75.

Answer 8.5g:

- 8.5 h) What is the strength of evidence associated with this **Bayes factor**? Are you convinced by this evidence?

Answer 8.5h:

Assignment 8.6: Performing a Bayesian linear regression

Suppose you work as an analyst for an insurance firm. Your firm wants to assess which variables can predict the height of of a customer insurance charges. For this assignment you will need the `insurance.csv`⁴ from the online resources. The data set contains a sample of 1338 health insurance customer and their recorded variables. For the current assignment, you can focus on the columns `charges` (the total charges of the customer), `age` (the customer age), `bmi` (the customer BMI score), and `neighbors` (the number of neighbors of the customer) columns.

- 8.6 a) Use the `read.csv()` function (and `setwd()` function if you prefer) to import the data set into an object named `insurance` .
- 8.6 b) Write down the regression equation for a **linear model** where you predict the variable `charges` on the basis of the predictor variables `age` , `bmi` , and `neighbors` .

Answer 8.6b:

charges = _____

Run the following code in R to create a Stan model M_1 that corresponds to this linear model with all **coefficients** .

```
regmodelcode <- '
data {
  int<lower=0> N;
  vector[N] x1;
  vector[N] x2;
  vector[N] x3;
  vector[N] y;
}
parameters {
  real beta0;
  real beta1;
  real beta2;
  real beta3;
  real<lower=0> sigma;
}
model {
  y ~ normal(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3, sigma);
}
'
```

Note: The following line can take a while to execute

```
regressionModel <- stan_model(model_code = regmodelcode, model_name = 'Regression')
```

⁴These data are adapted from the original open-source data set which can be found at: <https://www.kaggle.com/mirichoi0218/insurance/data>.

Note that you do not specify **prior distributions** (under the **model** section) for the β_0 , β_1 , and β_2 parameters. That is because, when you do not specify the prior distributions in **Stan**, the posterior distributions will be estimated solely from the data.

- 8.6 c) Use the **sampling()** function to sample from the **regressionModel** model and store the samples in an object named **regressionSamples**. Next, use the **summary()** function to find out the **posterior means** of the β_0 , β_1 , β_2 , and β_3 coefficients.

Answer 8.6c:

β_0 : _____ β_1 : _____ β_2 : _____ β_3 : _____

- 8.6 d) Use the **lm()** function to fit a regular **linear model** to the data. Use the **summary()** function to find out the estimates of the coefficients when done this way. Do they match the **posterior means** from assignment 8.6b?

Answer 8.6d:

YES / NO

Run the following code in R to create a figure of the posterior distribution of β_1 , the regression coefficient for the variable **age**:

```
plot(density(regressionSamples$beta1), xlab = '', ylab = '',
     yaxt = 'n', bty = 'n', main = expression(beta[1]))
```

- 8.6 e) Also create figures of the posterior distributions for β_0 , β_2 , and β_3 .
- 8.6 f) Find out the probability that the average increase in **charges** as a function of 1 point on the **bmi** scale is higher than 200 dollars.



Hint 8.11: Use the approach that you applied in assignment 8.3d and assignment 8.3e.

Answer 8.6f:

Probability: _____

- 8.6 g) Find out the probability that the value of β_1 is lower than 250.

Answer 8.6g:

Probability: _____

8.6 h) Find out between which values of β_3 90 percent of the **posterior samples** lie.

Answer 8.6h:

Lower bound: _____ Upper bound: _____

As you can see, the posterior distribution for the β_3 coefficient centers around zero, and it's contribution to the accuracy of the prediction can be questioned. To test how much more likely a model is that includes the β_3 predicts the data compared to a model that excludes the coefficient, you can use the procedure that you have learned in assignment 8.5.

8.6 i) Write code for a new Stan model M_2 where you remove the predictor variable **neighbors**.

8.6 j) Using the procedure in assignment 8.5, calculate the **Bayes factor** BF_{12} in favor of M_1 (the model that includes β_3) over M_2 (the model that excludes β_3).

8.6 k) What is the value of BF_{12} . Interpret this **Bayes factor** with respect to the coefficient β_0 . Incorporate the strength of the evidence in your answer.

Answer 8.6k:

Keep in mind that basically all statistical tests are a specific form of regression (like you have learned in chapter 6). This means that you can extend the Stan models that you have seen in this chapter to answer any question that you want to investigate.