Lecture 4 - Markov Chain Monte Carlo, Stan, and Complex Bayesian Models

Contents

- Today's Learning Objectives
- Loading (R) Packages
- Previously...
- 1. The Big Idea
- 2. The Bikeshare Data
- 3. Bikeshare Likelihood
- 4. Overview of Markov Chain Monte Carlo
- 5. Using **Stan** to Obtain our Posterior Distribution
- 6. Wrapping Up

Today's Learning Objectives

- 1. Illustrate the steps to build a likelihood in a Bayesian design.
- 2. Explore a complex dataset involving categorical, count, and continuous variables.
- 3. Apply a Bayesian model design with the proper random variables (**Normal**, **Gamma**, and **Poisson**).
- 4. Explain the general procedure of Markov Chain Monte Carlo (MCMC).
- 5. Practice computational Bayesian inference via Stan.
- 6. Draw inferential conclusions from our posterior distribution of parameters.

Loading R Packages

```
options(repr.matrix.max.rows = 6, repr.matrix.max.cols = 10)
library(tidyverse)
library(janitor)
library(cowplot)
library(rstan)
library(bayesplot)
library(wesanderson)
```

Previously...

We have explored the Bayesian framework on a conceptual level at the beginning of the course. Then, we got familiar with the Bayes' rule via probability theory. Recall the Bayes' rule as one of **the most powerful components in Bayesian analysis**. By applying this rule, we solved a foundational model called the **Beta-Binomial** in which we combine two random variables via our likelihood and prior distribution. Moreover, we explored the **Gamma-Poisson** model during lab time.

Nonetheless, real-life problems are not theoretically well behaved as the Beta-Binomial and Gamma Poisson cases. Moreover, complex data will require even more diverse distributions to design our Bayesian model. Hence, the analytical application of the Bayes' rule will not be possible for a nice exact posterior distribution.



Let us not worry! Stan will come in handy given this issue via MCMC to obtain an approximate posterior distribution.

1. The Big Idea

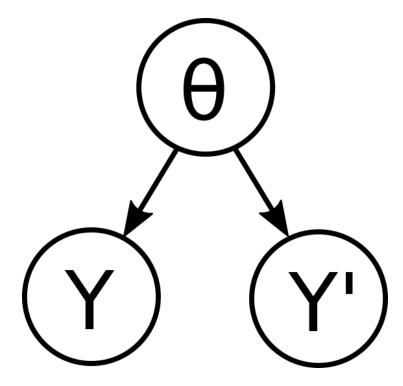
Let us recall the big idea of Bayesian modelling. It involves these steps:

- 1. Question: Pose a scientific question.
- 2. **Design:** Formulate variables and create a probabilistic model for them. Prior knowledge is included here!

- 3. **Infer:** Get **posterior samples** from the conditional distribution of **any variable you want to infer**, given your observed data and prior knowledge.
- 4. **Check:** Make sure the "samples" are *actually* from your posterior (these are model diagnostics, to be covered later!)
- 5. **Analyze:** Use your samples to compute summaries: mean, maximum a posteriori (MAP), variance, posterior credible intervals, etc.

The **scientific question** can involve **inferential** and/or predictive inquiries:

- Inferential: Using observed data Y to infer latent variable θ .
- **Predictive:** Using observed data Y to predict future data Y'.



We already discussed their differences in <u>Lecture 3 - Bayesian Statistics in Action: The Beta-Binomial Model</u> our via some examples. Now, let us think generatively!

1.1. To Design the Model, Think Generatively!

Ultimately, our model is an abstraction of a complex system involving different parameters of interest. Suppose we have d parameters $\theta_1, \ldots, \theta_d$. Mathematically, these d parameters can be put in a vector as follows:

$$\Theta = (\theta_1, \dots, \theta_d)^T$$
.

Note our Bayesian model is a **big joint probability distribution** $P(Y,Y',\Theta)$ over:

- Things we will see. Our observations Y.
- Things we have to infer: Unobserved parameters Θ and predicted observations Y'.

 This is why the posterior distribution of Θ is key!

We already had a first **inferential** approach (analytically speaking!) via the Beta-Binomial model in Lecture 3 - Bayesian Statistics in Action: The Beta-Binomial Model.

Now, it is much more intuitive to design a couple of steps that generate parameters and synthetic data:

- 1. First, generate Θ from the **prior** $P(\Theta)$.
- 2. Then, generate observations Y, Y' given Θ from the **likelihood** $P(Y,Y'|\Theta)$.

Attention

We are basically applying the law of conditioning

$$P(Y, Y', \Theta) = P(Y, Y'|\Theta)P(\Theta).$$

We will start with the **likelihood**. This function involves formulating certain variables. It is the core part of the model design! Then, we can establish the **prior distributions for our parameters of interest** within Θ (the likelihood will be of a big help here).

Important

This whole thing is a story of how our data were created.

So far in lab1, we only used our generative Monte Carlo models to simulate synthetic data given some fixed parameters. Nonetheless, we did not incorporate any observed data. In lab2, we will need to include observed data.

1.2. Constructing the Likelihood

To construct the distribution of the model parameters in Θ conditioned on the observed data Y (i.e., the likelihood), we need to follow these steps:

- **Step 1**: Formulate all data variables **type** (integer, real, categorical, etc.) and **support** (positive, nonnegative, binary, etc.).
- **Step 2**: Figure out which variables we will take as fixed covariates, which will be randomly generated, and how they are related.
- **Step 3**: Pick a distribution with the right support/type/dependencies. The **support** refers to a proper range of plausible values. Usually, the distribution will have some **UNKNOWN PARAMETERS** those will need a prior!

2. The Bikeshare Data

Let us recheck the bikeshare data. This is a complex case in which we have variables of different types.

- **Description:** Hourly record of the number of bike trips in Washington, DC in 2011/2012.
- Source: 2011/12 Capital Bikeshare Data.
- Variables: date, hour, holiday/weekend/workday, weather, temperature, number of trips.

Main statistical inquiries

Suppose we are interested in the following inferential inquiries:

- 1. How are temperature and weather associated with the **mean** number of trips?
- 2. Specifically, does the **mean** number of trips vary by the **type** of weather along with temperature?
- 3. How can we quantify this variation?

Let us start with in-class questions.



Exercise 11

Let us retake **DSCI 562** topics under a frequentist approach; what regression modelling can approach the above inquiries?

- A. A generalized linear model (GLM) such as Multinomial regression.
- B. A GLM such as Ordinal Logistic regression.
- C. A GLM such as Poisson regression.
- D. Ordinary Least-squares (OLS).



Solution to Exercise 11

The above inquiries primarily refer to the number of bike trips as a response, which is of integer type. More specifically, they refer to the **mean** number of trips. Therefore, Poisson regression might be a suitable approach in this context with the natural logarithm of the mean of bike trips as a link function.

Even though OLS might be somehow useful in this case, it is important to stress the fact that this approach is mainly intended for continuous responses, and it is not able to handle **heteroscedasticity** (recall OLS assumes a common variance σ^2 across all observations unlike Poisson regression).

2.1. Data Wrangling

The data frame bikeshare_data contains sixteen different variables. For the sake of our example and inquiries, we will use a small subset of them.

bikeshare_data <- read_csv("../data/bikeshare_hourly.csv", show_col_types = FA bikeshare data

A spec_tbl_df: 17379 × 17

instant	dteday	season	yr	mnth		hum	windspeed	casual	register
<dbl></dbl>	<date></date>	<dbl></dbl>	<dbl></dbl>	<dbl></dbl>		<dbl></dbl>	<dbl></dbl>	<dbl></dbl>	<db< th=""></db<>
1	2011- 01-01	1	0	1		0.81	0	3	
2	2011- 01-01	1	0	1		0.80	0	8	;
3	2011- 01-01	1	0	1		0.80	0	5	:
:	:	:	:	:	٠.	:	:	:	
17377	2012- 12-31	1	1	12		0.60	0.1642	7	{
17378	2012- 12-31	1	1	12		0.56	0.1343	13	4
17379	2012- 12-31	1	1	12		0.65	0.1343	12	;

We quickly wrangle our data, select, and rename the three variables of interest:

- weather: A categorical variable indicating the overall weather situation. It is coded as:
 - 1: Clear, Few clouds, Partly cloudy.
 - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist.
 - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds.
- temp: Temperature in Celsius degrees. The values are multiplied times 41 since they were normalized in the original dataset with the maximum value (you can check the Readme.txt). It is a continuous variable.
- ride_count: Count of total rental bikes, including casual and registered. It is an
 integer-type variable.

Attention

Originally, there was a category [4] (Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog) in weather. Nonetheless, only three rows contained this category. Hence, they were removed, yielding 17376 rows in total to be used.

```
bikeshare_data <- bikeshare_data |>
    select(weathersit, temp, cnt) |>
    mutate(temp = temp * 41) |>
    filter(weathersit < 4)
colnames(bikeshare_data) <- c("weather", "temp", "ride_count")
bikeshare_data</pre>
```

A tibble: 17376 × 3

weather	temp	ride_count		
<dbl></dbl>	<dbl></dbl>	<dbl></dbl>		
1	9.84	16		
1	9.02	40		
1	9.02	32		
:	:	:		
1	10.66	90		
1	10.66	61		
1	10.66	49		

```
nrow(bikeshare_data)
```

17376

2.2. Exploratory Data Analysis

There is a reason why the exploratory data analysis (EDA) is extremely helpful in Bayesian modelling. We might wonder, **how do ride counts behave by weather category?**

Firstly, let us plot histograms of $ride_count$ by category in weather. Note the y-axis is adjusted with $after_stat(density)$ since category 3 is present in fewer rows in $bikeshare_data$

```
bikeshare_data |>
  tabyl(weather) |>
  adorn_pct_formatting()
```

A tabyl: 3×3

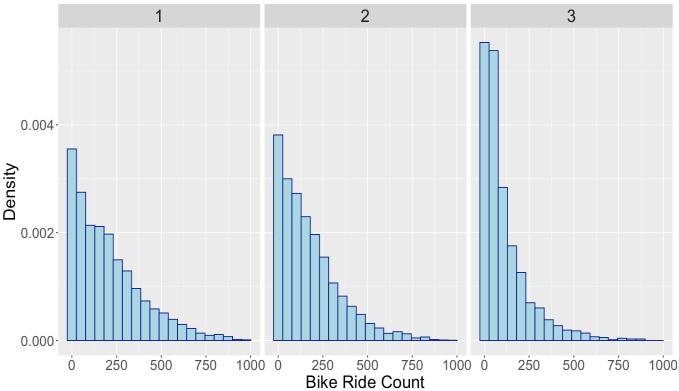
weather	n	percent		
<dbl></dbl>	<dbl></dbl>	<chr></chr>		
1	11413	65.7%		
2	4544	26.2%		
3	1419	8.2%		

```
options(repr.plot.height = 7, repr.plot.width = 12)

bikeshare_data_hist <- ggplot(bikeshare_data, aes(
    x = ride_count, y = after_stat(density)
)) +
    geom_histogram(bins = 20, color = "darkblue", fill = "lightblue") +
    labs(y = "Density", x = "Bike Ride Count", fill = "") +
    ggtitle("Histograms by Weather Category") +
    theme(
        plot.title = element_text(size = 24, face = "bold"),
        axis.text.x = element_text(size = 17, angle = 0),
        axis.text.y = element_text(size = 17, angle = 0),
        axis.title = element_text(size = 21),
        strip.text.x = element_text(size = 21)
) +
    facet_grid(. ~ as.factor(weather))</pre>
```

bikeshare_data_hist

Histograms by Weather Category



Now, how can we plot the relationship between ride_count and temp by weather category? Let us create scatterplots of temp versus ride_count by weather category.

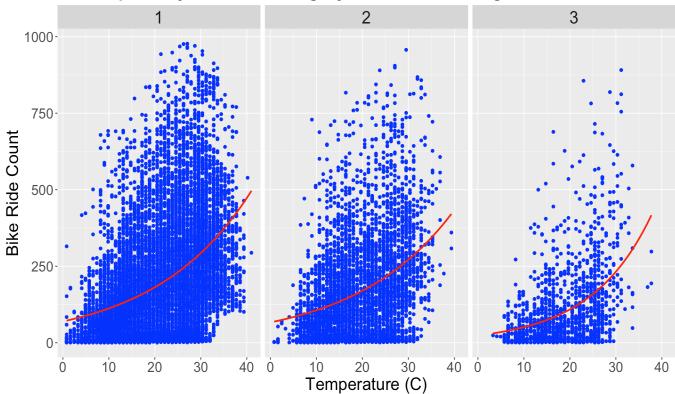
Hint

Identifying patterns with count versus continuous data is not straightforward. How about if we apply what we learned in DSCI 562? Let us fit a simple **frequentist**Poisson regression with a log-link function with ride_count as a response (the insample predictions are plotted as red lines).

```
bikeshare_data_scatterplot <- ggplot(bikeshare_data, aes(</pre>
  x = temp, y = ride_count
)) +
  geom_point(color = "blue") +
  labs(y = "Bike Ride Count", x = "Temperature (C)") +
  geom smooth(
    data = bikeshare_data, aes(temp, ride_count), method = "glm", colour = "re
    formula = "y \sim x",
    se = FALSE, method.args = list(family = poisson)
  ggtitle("Scatterplots by Weather Category with Poisson Regression Fitted Lin
  theme(
    plot.title = element_text(size = 23, face = "bold"),
    axis.text.x = element_text(size = 17, angle = 0),
    axis.text.y = element_text(size = 17, angle = 0),
    axis.title = element text(size = 21),
    strip.text.x = element text(size = 21)
  facet_grid(. ~ as.factor(weather))
```

bikeshare_data_scatterplot

Scatterplots by Weather Category with Poisson Regression Fitted Lines



Based on our previous side-by-side plots (histograms and scatterplots), let us answer the following:



Exercise 12

Graphically speaking, how do bike ride counts behave as we increase the weather category? Recall that a higher category represents harsher weather conditions.

- **A.** When harsher weather conditions are present, the counts of bike rides increase at a faster pace when temperature increases.
- **B.** When **nicer weather** conditions are present, the counts of bike rides **decrease** at a faster pace when temperature increases.
- C. There is not graphical difference on the behaviour of bike rides as temperature increases across weather categories.
- **D.** When **nicer weather** conditions are present, the counts of bike rides **increase at** a faster pace when temperature increases.



Solution to Exercise 12

The correct answer is **D**.

Even though warmer temperatures are graphically associated with larger counts of bike rides, the counts seem to increase at different rates (check the red fitted lines). That said, as the weather gets nicer, the bike counts increase at faster rates.

3. Bikeshare Likelihood

We have already obtained valuable insights with our EDA to proceed with our likelihood construction. Next, we will apply this likelihood to bikeshare_data. It will contain appropriate random variables according to the type of variables present in the dataset.

Note we have the following:

- Weather situation $w_i \in \{1,2,3\}$ (categorical, where 3 indicates the harshest conditions).
- Temperature $t_i \in \mathbb{R}$ (numerical).
- Number of bike rides r_i (non-negative count).

3.1. Modelling the Bike Ride Counts

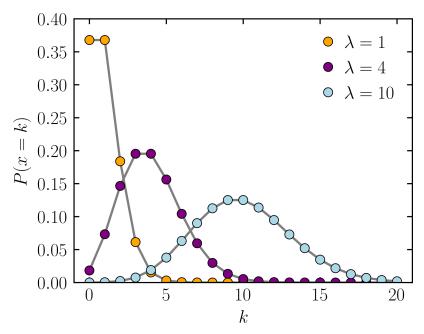
We do not need to **generate** temperature t_i and weather situation w_i for observation i from our model. We are not making a weather forecast model. We are only making inference here! These are covariates that will modify the likelihood of our observations r_i (count of bike rides):

- In regression, you have **inputs** and **output**.
- Inputs are covariates in this case. We expect more rides when it is warmer **OR** weather is nicer (recall our EDA!). Thus, w_i and t_i (inputs) should influence r_i (output).

So our model's observations Y will be the collection of trip counts in bikeshare_data (r_1, r_2, \ldots) . These are the rows in column ride_count.

We start with the rides r_i , which are counts. Hence, a suitable random variable for our Bayesian model is **Poisson distribution** whose mean λ is theoretically equal to its variance.

Poisson Distribution



Source: Poisson distribution. (2025, January 25). In Wikipedia. https://en.wikipedia.org/wiki/Poisson_distribution

Therefore, the **LIKELIHOOD** for bike ride counts is modelled as:

$$R_i \sim \text{Poisson}(\lambda_{w_i}(t_i)).$$

Attention

In the context of this problem, Poisson's mean will be in function of temperature t_i along with the type of weather w_i .

igcap How to define $\lambda_{w_i}(t_i)$?

This is one of the most critical modelling tasks, and we will need to use some distribution theory here, along with a mathematical function. Statistically speaking, the Poisson mean **cannot be negative**, even though it could be continuous!

Furthermore, let us make a **strong assumption** in our modelling. We will assume riders will start taking biking trips at a specific temperature, and **the riding rate will linearly increase with warmer temperature values**.

Mathematically, $\lambda_{w_i}(t_i)$ can be represented as:

$$\lambda_w(t) = \log\{1 + \exp[a_w(t - l_w)]\}.$$
 (7)

The log-transformation is applying the idea we explored in Poisson regression. This also creates some unknown parameters for each $w \in \{1, 2, 3\}$:

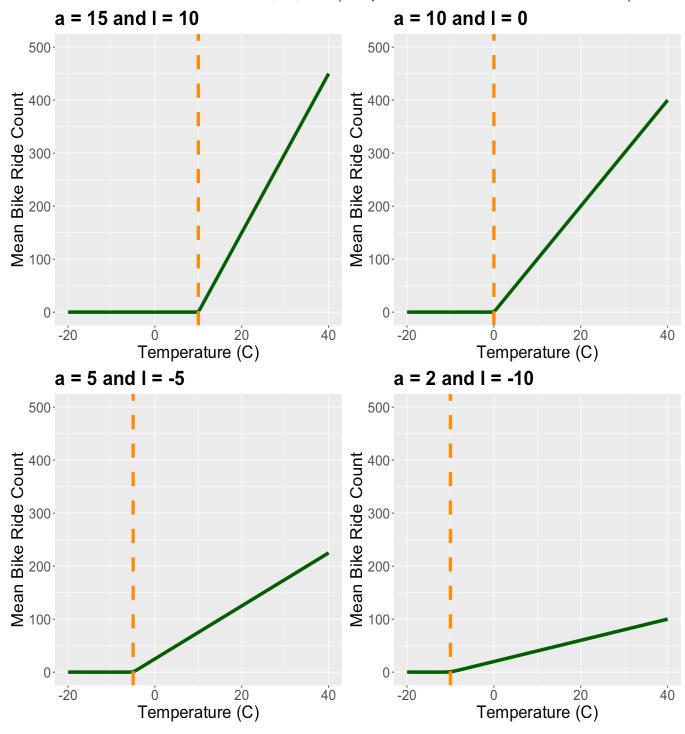
- a_w is the **slope** of mean trips in weather w.
- l_w is the **elbow location** where riders start taking trips in weather w.

It will be easier to visualize function (7) in plots with different values for a_w and l_w .

Suppose that for a weather category w we model the following values:

- $a_w=15$ and $l_w=10$.
- $\bullet \ \ a_w=10 \ {\rm and} \ l_w=0.$
- $a_w = 5$ and $l_w = -5$.
- $a_w = 2$ and $l_w = -10$.

► Show code cell source





Exercise 13

Descriptively, what is the behaviour of our modelling function for $\lambda_w(t)$?



Solution to Exercise 13

We can see that l indicates the temperature at which the mean bike ride count will start increasing. Before l, the mean bike ride count is non-negative, which aligns with the mean of a Poisson random variable. Moreover, a larger value of aindicates a larger rate at which the mean bike ride count increases as t increases.

Note you **do not have** to assign fixed values for a and l in your Bayesian model; they will be parameters of interest in our Bayesian model.

3.2. Modelling the Weather Parameters

We know weather status is categorical, temperature is real, and the number of rides is a non-negative integer count. Moreover, let us assume weather status and temperature are fixed covariates (we are not trying to model them). The counts will be Poisson distributed.

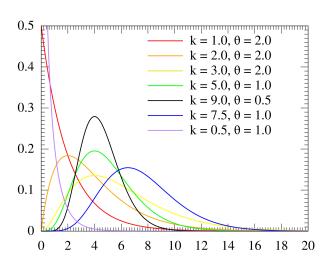
Nonetheless, our previous modelling of the behaviour of $\lambda_{w_i}(t_i)$ yielded six different parameters we need to model! Thus, let us find some suitable distributions for:

- $\bullet \ \ a_w \text{ with } w \in \{1,2,3\}.$
- l_w with $w \in \{1, 2, 3\}$.

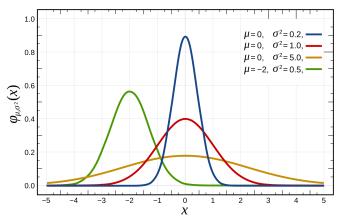
Our previous plots with different values of a_w and l_w shed light on what class of distributions will be suitable in our model:

- For the **CONTINUOUS** slopes a_{w_t} we will assume they will only take on nonnegative values. A handy distribution for this case is the Gamma distribution
- On the other hand, we will allow the **CONTINUOUS** elbows l_w to take on any real value. Thus, the **Normal distribution**.

Gamma Distribution



Normal Distribution



Sources:

Gamma distribution. (2025, January 14). In Wikipedia.

https://en.wikipedia.org/wiki/Gamma_distribution

Normal distribution. (2025, January 26). In Wikipedia.

https://en.wikipedia.org/wiki/Normal_distribution

Therefore, our **prior distributions** for these weather parameters would be:

$$egin{aligned} &l_w \sim \mathcal{N}(\mu, \sigma^2) \quad ext{for} \quad w \in \{1, 2, 3\} \ &a_w \sim \operatorname{Gamma}(k, heta) \quad ext{for} \quad w \in \{1, 2, 3\}. \end{aligned}$$

Now, we have **hyperparameters**! They are μ , σ^2 , k and θ .

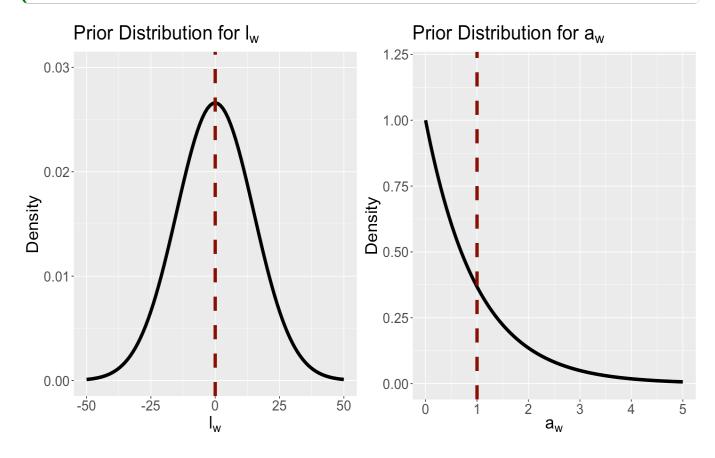
Moreover, we will make them reflect uncertainty:

$$egin{aligned} \mu &= 0 \ \sigma^2 &= 15^2 = 225 \ k &= 1 \ heta &= 1. \end{aligned}$$

The corresponding prior distributions for l_w and a_w are plotted below (the prior means are the vertical dashed red lines). Note we are giving considerable uncertainty to l_w while allowing any real number. On the other hand, we restrict a_w to nonnegative numbers while

allocating larger probabilities to small values (i.e., we are not expecting large rates of change for the mean counts of bike rides).

➤ Show code cell source



Finally, with a dataset containing n observations (i.e., rows in bikeshare_data), for the ith observation $(i=1,\ldots,n)$ with temperature t_i :

$$egin{aligned} ext{likelihood:} & R_i \mid a_{w_i}, l_{w_i} \overset{ ext{ind}}{\sim} \operatorname{Poisson}\left(\lambda_{w_i}(t_i)
ight) \ ext{where} & \lambda_{w_i}(t_i) = \log\{1 + \exp[a_{w_i}(t_i - l_{w_i})]\} \quad ext{and} \quad w_i \in \{1, 2, 3\} \ ext{priors:} & a_{w_i} \sim \operatorname{Gamma}(k = 1, heta = 1) \ & l_{w_i} \sim \mathcal{N}(\mu = 0, \sigma^2 = 15^2). \end{aligned}$$

4. Overview of Markov Chain Monte Carlo

In general, a Bayesian model will have d parameters of interest for which we will need to obtain a **joint posterior distribution**:

$$\Theta = (\theta_1, \dots, \theta_d)^T$$
.

Recall the Bayes' rule:

$$posterior = \frac{prior \times likelihood}{normalizing \ constant} \propto prior \times likelihood.$$

Using Θ as our parameter **vector** of interest, let $f(\Theta)$ be the **multivariate** joint prior. Moreover with our observed data $\mathbf{y} = (y_1, \dots, y_n)^T$, let $\ell(\Theta \mid \mathbf{y})$ be the corresponding **multivariate** likelihood function along with the **multivariate** normalizing constant $f(\mathbf{y})$.

Then, for the **multivariate** joint posterior $f(\Theta \mid \mathbf{y})$, the Bayes' rule becomes:

$$f(\Theta \mid \mathbf{y}) = rac{f(\Theta) imes \ell(\Theta \mid \mathbf{y})}{f(\mathbf{y})} \propto f(\Theta) imes \ell(\Theta \mid \mathbf{y}).$$

Why are we highlighting **multivariate** all the time? It is because we have d parameters in vector Θ .

Suppose we want to obtain an **analytical** form for f(y), then this will be an integral to solve:

$$f(\mathbf{y}) = \int_{ heta_1} \int_{ heta_2} \cdots \int_{ heta_d} f(\Theta) \ell(\Theta \mid \mathbf{y}) d heta_d \, \cdots d heta_1.$$

Solving this multivariate integral looks pretty challenging! Nonetheless, recall we do not need to obtain f(y), but we still need to work around the math in

$$f(\Theta \mid \mathbf{y}) \propto f(\Theta) \times \ell(\Theta \mid \mathbf{y}).$$

Important

Cases such as the Beta-Binomial and Gamma-Poisson are well behaved and **approachable** for parameters π and λ , respectively. Therefore, we can obtain a **closed solution** for the corresponding posterior distribution.

But, what can we do when we have a whole set of parameters in vector Θ ? It will not be possible to find an exact form for $f(\Theta \mid \mathbf{y})$, but it is possible to approximate it via an algorithm!

Let us explore the MCMC algorithm by parts: Markov Chain and Monte Carlo.

4.1. Monte Carlo Algorithm

We will start with the simple Monte Carlo algorithm. Suppose you have a closed analytical form for the posterior distribution $f(\Theta \mid \mathbf{y})$ (e.g., the Beta posterior in the Beta-Binomial model). You can build your **independent** Monte Carlo sample $\{\Theta_1,\ldots,\Theta_N\}$ of size N, from $f(\Theta \mid \mathbf{y})$ by selecting each element Θ_i (i = 1, ..., N) in the following way:

- Step 1. Draw Θ_i from $f(\Theta \mid \mathbf{y})$.
- Step 2. Go there.

This is a random walk!



Caution

Drawing fully independent elements for our sample WILL NOT be possible in MCMC. Moreover, **note that** $n \neq N$.

4.2. Markov Chain

Here comes the MCMC twist to the previous Monte Carlo approach: Markov Chain.

What is a Markov Chain? In terms of our Bayesian modelling for the posterior, it is a sequence $\{\Theta_1,\ldots,\Theta_N\}$ of size N whose elements are **NOT ENTIRELY INDEPENDENT** (hence the word chain!).

What does NOT ENTIRELY INDEPENDENT implicate? It implicates the so-called Markov property:

The next sampled Θ_{i+1} will depend on the current Θ_i but not on previous $\Theta_{i-1},\Theta_{i-2},\dots$

More formally, because of this property, the conditional probability function of Θ_{i+1} can be depicted as:

$$f(\Theta_{i+1} \mid \Theta_i, \mathbf{y}) = f(\Theta_{i+1} \mid \Theta_1, \Theta_2, \dots \Theta_i, \mathbf{y}).$$

Again, there is no free lunch here! Each element in this chain can be drawn from another distribution and not specifically from our target posterior $f(\Theta \mid \mathbf{y})$:

$$f(\Theta_{i+1} \mid \Theta_i, \mathbf{y}) \neq f(\Theta_{i+1} \mid \mathbf{y}).$$

Thus, what is the point of this Markov Chain approach? The answer lies in the MCMC algorithm that approximates $f(\Theta \mid \mathbf{y})$ via our N elements in the chain.

Important

There is more than one MCMC algorithm to do so, and the foundational one is the Metropolis-Hastings algorithm.

4.3. The Metropolis-Hastings Algorithm

This is the algorithm that will allow us to obtain an approximation of the posterior distribution $f(\Theta \mid \mathbf{y})$ via our Markov Chain $\{\Theta_1, \dots, \Theta_N\}$. Hence, with Θ_i in our current iteration $(i=1,2,\dots,N-1)$, the next iteration Θ_{i+1} will be selected through a two-step process.

Recall the Bayes' rule:

$$f(\Theta \mid \mathbf{y}) \propto f(\Theta) imes \ell(\Theta \mid \mathbf{y}).$$

Let the current $\Theta_i = \Theta$. Then, the algorithm's two steps are:

• Step 1. Randomly draw a new location Θ' from a proposed model with PDF $q(\Theta' \mid \Theta)$. This proposed model is what we call a jumping distribution and it could be a **Uniform** or **Normal** which are symmetric.

Important

Another choice of a jumping distribution could be **asymmetric**, such as the **Log-normal**. We can choose this jumping distribution class when there are constraints on our parameters of interest (e.g., they are nonnegative), or we want to sample large values from the chain (recall the Log-normal is a heavy-tailed distribution).

• **Step 2.** Now, let us decide to go or not to go to the proposed Θ' :

Important

Step 2 is divided in two parts:

a. Compute the acceptance probability for Θ' :

$$lpha = \minigg\{1, rac{f(\Theta')\ell(\Theta'\mid \mathbf{y})}{f(\Theta)\ell(\Theta\mid \mathbf{y})} imes rac{q(\Theta\mid \Theta')}{q(\Theta'\mid \Theta)}igg\}.$$

b. Obtain the next location Θ_{i+1} as a Bernoulli trial (**flipping an unfair coin!**) taking on the two possible values:

$$\Theta_{i+1} = egin{cases} \Theta' & ext{with probability } lpha \ \Theta & ext{with probability } 1-lpha. \end{cases}$$

4.4. The Metropolis Algorithm

The previous Metropolis-Hastings algorithm can be simplified when the PDF of the jumping distribution is Uniform such that:

$$q(\Theta' \mid \Theta) = q(\Theta \mid \Theta') = \begin{cases} \frac{1}{2w} & \text{when } \Theta \text{ and } \Theta' \text{ are within } w \text{ units of each other} \\ 0 & \text{otherwise.} \end{cases}$$

This jumping distribution allocates the same probability of switching from Θ to Θ' compared to switching from Θ' to Θ . Note that w is the window parameter. Having said that, the definition of α will change to:

$$lpha = \miniggl\{1, rac{f(\Theta')\ell(\Theta'\mid \mathbf{y})}{f(\Theta)\ell(\Theta\mid \mathbf{y})}iggr\}.$$

4.5. Other MCMC Algorithms

Metropolis-Hastings and its simplified Metropolis version are only a couple of MCMC tools. These two approaches target single parameters (e.g., a probability of success π or a mean λ). However, when we want to deal with d parameters, such as $\Theta = (\theta_1, \dots, \theta_d)^T$, there are more complex tools like Gibbs sampling (used in <u>JAGS</u>, <u>Just Another Gibbs Sampler</u>) or the <u>No-U-Turn sampler (NUTS)</u> which is a Hamiltonian Monte Carlo (HMC) algorithm (implemented in <u>Stan</u>).

4.6. Important Facts to Consider when

sampling() via rstan

The MCMC algorithm has a **learning curve**; thus, the Markov Chain initially starts sampling unreasonable numbers to approximate the posterior $f(\Theta \mid \mathbf{y})$. Therefore, we have a **burn-in stage** at the beginning of <code>sampling()</code> defined by the parameter <code>warmup</code>. This parameter will discard the first <code>warmup</code> iterations for the final simulation output.

Recall that consecutive elements in the chain **are not independent**. Hence, given the Markov property, we could skip draws between our definitive sampled values throughout the simulation to achieve **a certain degree of sampling independence**. This is defined by thin.

Important

We will obtain the following number of **approximate posterior samples** via sampling():

Number of approximate posterior samples $=\frac{\text{iter}-\text{warmup}}{\text{thin}}$.

5. Using Stan to Obtain our Posterior Distribution

We already designed our Bayesian model. Now it is time to code it in Stan. Recall that with a dataset containing n observations (i.e., rows in bikeshare_data) which is our training set size), for the ith observation ($i=1,\ldots,n$) with temperature t_i :

$$ext{likelihood:} \quad R_i \mid a_{w_i}, l_{w_i} \overset{ ext{ind}}{\sim} \operatorname{Poisson}ig(\lambda_{w_i}(t_i)ig) \ ext{where} \quad \lambda_{w_i}(t_i) = \log\{1 + \exp[a_{w_i}(t_i - l_{w_i})]\} \quad ext{and} \quad w_i \in \{1, 2, 3\} \ ext{priors:} \quad a_{w_i} \sim \operatorname{Gamma}(k = 1, \theta = 1) \ l_{w_i} \sim \mathcal{N}(\mu = 0, \sigma^2 = 15^2).$$

5.1. Coding the Model

Recall Stan has a block structure. Therefore, we will use the following blocks:

- data: The **inputs** needed to run the Bayesian model.
- parameters: The parameters to be sampled for our posterior analysis (a_w and l_w for $w \in \{1,2,3\}$). Their approximate posterior distributions will allow us to establish the extent to which temperature and mean bike ride count are related across different categories of weather (this is Bayesian inference!).
- model: Our Bayesian model.

Caution

Stan needs to know every single detail of our data and generative model (object types, sizes, loops, etc.).

Unlike the R markdown, we do not have a Stan cell in Jupyter notebook/lab. Hence, we need to code our bikeshare_model as a string.

```
bikeshare model <- "data {</pre>
int<lower=1> number weather cat;
                                     // integer to indicate how many weather
int<lower=1> n;  // how many rows we have in bikeshare_data
vector[n] temp;
                   // column bikeshare data$temp of real type
int<lower=1> weather[n];
                           // column bikeshare data$weather as integer
                                // column bikeshare data$ride count as intege
int<lower=0> ride count[n];
parameters {
vector<lower=0>[number_weather_cat] a;  // a vector of size number_weather]
vector[number_weather_cat] l;  // a real-type vector of size number_weathe
model {
for (j in 1:number_weather_cat){      // we have three PAIRS of priors, i.e.,
  a[i] \sim gamma(1, 1);
  l[i] \sim normal(0, 15);
for (i in 1:n){     // building our likelihood
  int w index = weather[i];  // auxiliary variable for the ith weather cat
  ride_count[i] ~ poisson(log(1 + exp(a[w_index] * (temp[i] - l[w_index]))));
}
}"
```

5.2. MCMC Simulation

Next, we create a **dictionary** with all the data variables we need to pass into Stan as a list().

```
bikeshare_dictionary <- list(
  number_weather_cat = 3,
  n = nrow(bikeshare_data),
  temp = bikeshare_data$temp,
  weather = as.integer(bikeshare_data$weather),
  ride_count = as.integer(bikeshare_data$ride_count)
)</pre>
```

We have not compiled bikeshare_model. Hence, we cannot use sampling() directly but stan() as in the cell below. Note the MCMC arguments are provided as in sampling():

- **Chains**: **How many times** we want to run the Bayesian simulation.
- <u>iter</u>: The **number of approximate posterior sampled sets of <u>parameters</u>** we want (including **burn-in**).
- warmup: The **burn-in** period.
- thin: Number of samples to **skip** between each draw.
- seed: Reproducibility seed.

```
posterior_bikeshare <- stan(
   model_code = bikeshare_model,
   data = bikeshare_dictionary,
   chains = 1,
   iter = 20000,
   warmup = 2000,
   thin = 40,
   seed = 553,
   verbose = TRUE,
   algorithm = "NUTS"
)</pre>
Show code cell output
```

5.3. Obtaining, Visualizing, and Interpreting the Posterior Results

```
posterior_bikeshare is our one-stop-shop!
```

We obtained a total of

$$\frac{\text{iter} - \text{warmup}}{\text{thin}} = \frac{20000 - 2000}{40} = 450$$

approximate posterior samples to be summarized in our inferential conclusions.

You can obtain the summary() statistics for the object posterior_bikeshare. These statistics are drawn from the approximate joint posterior distribution of the parameters

of interest: a_1 , a_2 , a_3 , l_1 , l_2 , and l_3 . Lower and upper bounds for the **95% credible intervals** are shown in columns 2.5% and 97.5%.

Metrics se_mean (Monte Carlo standard error), n_eff (effective sample size), and Rhat are model diagnostics (to be covered later on at the end of this course).

4 Attention

lp__ refers to the logarithm of the posterior joint density **up to a constant that does not depend on the parameters of interest** (we will not use it in this course).

You can find a useful <u>optional resource</u> if you are more interested in knowing how Stan performs its computation.

round(summary(posterior_bikeshare)\$summary, 2)[-7,]

A matrix: 6×10 of type dbl

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
a[1]	9.46	0	0.01	9.44	9.46	9.47	9.47	9.49	423.09	1
a[2]	8.20	0	0.03	8.16	8.19	8.20	8.22	8.25	427.82	1
a[3]	7.00	0	0.03	6.93	6.98	7.00	7.02	7.06	418.97	1
I[1]	-0.69	0	0.03	-0.75	-0.71	-0.69	-0.68	-0.64	454.05	1
I[2]	-1.90	0	0.06	-2.02	-1.95	-1.91	-1.86	-1.78	480.15	1
I[3]	2.78	0	0.06	2.67	2.75	2.78	2.83	2.90	398.77	1

Before plotting our posterior results, let us recheck the inferential inquiries:

- 1. How are temperature and weather associated with the **mean** number of trips?
- 2. Specifically, does the **mean** number of trips vary by the **type** of weather along with temperature?
- 3. How can we quantify this variation?

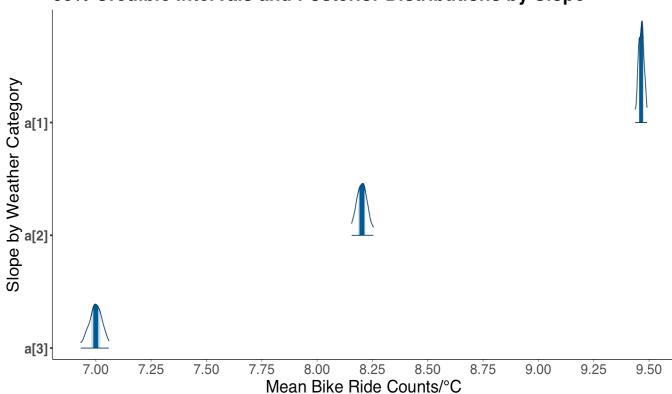
Now, we can also plot the six 95% credible intervals and posterior distributions via mcmc_areas() (from package bayesplot). Based on our summary and credible intervals, we can conclude the following (while addressing the previous inquiries):

- Indeed, we have posterior evidence to state that the rates of change for mean counts of bike rides (a_w) are different by weather category (the corresponding 95% credible intervals do not overlap!). A higher rate of change mean count is associated to a nicer weather.
- The temperature elbows l_w by weather category are also different (the corresponding 95% credible intervals do not overlap!).

```
slope_CIs <- mcmc_areas(posterior_bikeshare, pars = c("a[1]", "a[2]", "a[3]"),
    theme(
    plot.title = element_text(size = 24, face = "bold", family = "sans"),
    axis.text.x = element_text(size = 17, angle = 0, family = "sans"),
    axis.text.y = element_text(size = 17, angle = 0, family = "sans"),
    axis.title = element_text(size = 21, family = "sans")
) +
    ggtitle("95% Credible Intervals and Posterior Distributions by Slope") +
    scale_x_continuous(breaks = seq(7, 10, 0.25)) +
    labs(x = "Mean Bike Ride Counts/°C", y = "Slope by Weather Category")
slope_CIs</pre>
```

```
Scale for x is already present. Adding another scale for x, which will replace the existing scale.
```

95% Credible Intervals and Posterior Distributions by Slope

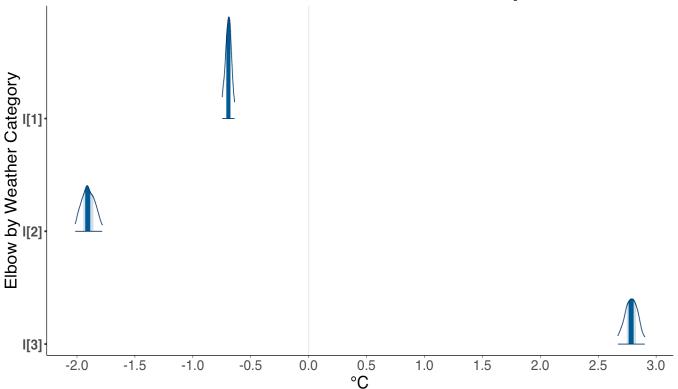


```
elbow_CIs <- mcmc_areas(posterior_bikeshare, pars = c("l[1]", "l[2]", "l[3]"),
    theme(
    plot.title = element_text(size = 24, face = "bold", family = "sans"),
    axis.text.x = element_text(size = 17, angle = 0, family = "sans"),
    axis.text.y = element_text(size = 17, angle = 0, family = "sans"),
    axis.title = element_text(size = 21, family = "sans")
) +
    ggtitle("95% Credible Intervals and Posterior Distributions by Elbow") +
    scale_x_continuous(breaks = seq(-2.5, 3, 0.5)) +
    labs(x = "°C", y = "Elbow by Weather Category")

elbow_CIs</pre>
```

Scale for x is already present. Adding another scale for x, which will replace the existing scale.

95% Credible Intervals and Posterior Distributions by Elbow



We can obtain the **posterior histograms** by parameter via $mcmc_hist()$ from bayesplot. The posterior distributions look fairly symmetric! **Note that parameters** a_w **have positive ranges since we constrain them prior as nonnegative.**

Since we have 450 sampled sets of six parameters each, we can plot 450 times our mean count equation

$$\lambda_w(t) = \log\{1+\exp[a_w(t-l_w)]\} \quad ext{for} \quad w \in \{1,2,3\}.$$

by weather category.

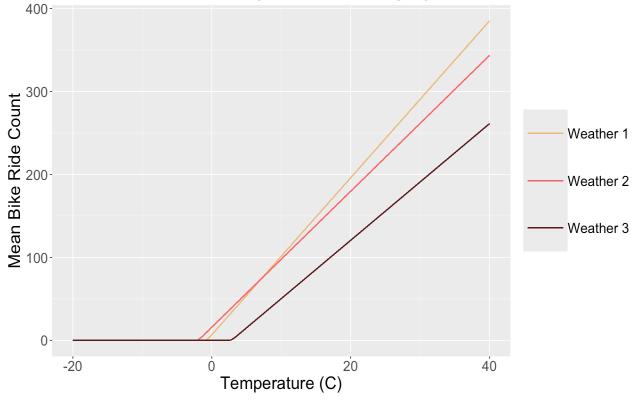
• Attention

Note the low variability across our 450 samples; the lines practically overlap.

```
posterior samples <- as.data.frame(posterior bikeshare)</pre>
plot 450 samples <- ggplot() +</pre>
  \times \lim(-20, 40) +
  theme(
    plot.title = element text(size = 24, face = "bold"),
    axis.text.x = element_text(size = 17, angle = 0),
    axis.text.y = element text(size = 17, angle = 0),
    axis.title = element_text(size = 21),
    legend.text = element text(size = 17, margin = margin(r = 2, unit = "cm"))
    legend.title = element blank(),
    legend.key.size = unit(2, 'cm')
  labs(y = "Mean Bike Ride Count", x = "Temperature (C)") +
  ggtitle("Plots of 450 Samples by Weather Category")
# Plotting 450 lambda functions for w 1
for (i in 1:nrow(posterior samples))
  plot 450 samples <- plot 450 samples +
    geom function(
      fun = function(x) {
        log(1 + exp(posterior samples$`a[1]`[i] *
          (x - posterior samples$`l[1]`[i])))
      },
      aes(colour = "Weather 1"), alpha = 0.4
    )
}
# Plotting 450 lambda functions for w 2
for (i in 1:nrow(posterior_samples))
  plot_450_samples <- plot_450_samples +</pre>
    geom function(
      fun = function(x) {
        log(1 + exp(posterior_samples$`a[2]`[i] *
          (x - posterior samples$`l[2]`[i])))
      },
      aes(colour = "Weather 2"), alpha = 0.4
}
# Plotting 450 lambda functions for w 3
for (i in 1:nrow(posterior samples))
  plot 450 samples <- plot 450 samples +
    geom function(
      fun = function(x)  {
        log(1 + exp(posterior_samples$`a[3]`[i] *
          (x - posterior samples$`l[3]`[i])))
      },
      aes(colour = "Weather 3"), alpha = 0.4
}
```

plot_450_samples + scale_color_manual(values = wes_palette(n = 3, name = "Gran





The harshest weather 3 has the smallest slope and largest elbow. On the other hand, the nicest weather 1 has the largest slope but not the smallest elbow!



Exercise 14

Once we have estimated our Bayesian model, what do you think is the major drawback? Thus, how can we improve this model?



Solution to Exercise 14

Our likelihood involves this equation

$$\lambda_w(t) = \log\{1 + \exp[a_w(t - l_w)]\},\,$$

that is not capped at some maximum temperature. We should assume people stop making bike trips on hot days. However, our model does not assume so. Therefore, we might need to find another mathematical equation to model growth. One possible way is to use an S-shaped logistic growth model.

6. Wrapping Up

- We have built a Bayesian generative model totally from scratch by setting up key mathematical relations between our variables of interest.
- Knowing the suitable class of distributions to use is of particular importance.
- There is more than one possible modelling approach, and all models can be improved!