MATH 4225
Foundation of Big Data and Learning
Project

Mustafa Batin EFE
23501154

December 21, 2023

1 Introduction

In this project, the task is to work with the data obtained from the data.gov.hk. The goal is to develop a PGM and its corresponding generative model.

Firstly, we will load and pre-process the necessary datasets. We will then perform exploratory data analysis and visualization to gain initial insights into the data.

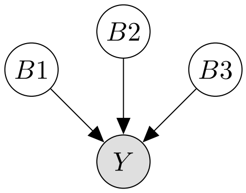
Then, we will conduct a correlation analysis to identify any relationships between the number of taxi accidents and the number of visitors, such as mainland visitors, Hong Kong residents, and tourists.

After that, we will use MCMC(Markov Chain Monte Carlo) and Gibbs Sampling techniques to perform an analysis and examine the posterior distributions of the parameters.

Finally, we will use VI (Variational Inference to examine the posterior distributions of the parameters.

2 Probabilistic Graphical Model

- Used the `pygraphviz` library to create a Bayesian network graph.
- The code renders the graph as a file and displays the image.



3 Data Sources and Preprocessing

3.1 Data Sources

The data used for this study was obtained from the following sources:

3.1.1 Daily Passenger Traffic Data

- Source: Hong Kong Immigration Department
- Dataset: Statistics on Daily Passenger Traffic
- Resource URL: https://data.gov.hk/en-data/dataset/hk-immd-set5-statistics-daily-passenger-traffic/resource/e06a2a45-fe05-4eb4-9302-237d74343d52

3.1.2 Accident Data

- Source: Hong Kong Transport Department
- Dataset: Monthly Traffic and Transport Digest
- Resource URL: https://data.gov.hk/en-data/dataset/hk-td-tis$_1 0 - monthly - traffic - and - transport - digest$

3.1.3 Maximum Taxi Involvement Data

- Source: Hong Kong Transport Department
- Dataset: Monthly Traffic and Transport Digest (Section 7: Road Traffic Accident Statistics - Table 7.2)
- Resource URL: https://data.gov.hk/en-data/dataset/hk-td-tis$_1 0 - monthly - traffic - and - transport - digest/resource/3e1c47ab - fed4 - 4e69 - bd22 - e2523bca3266$

3.2 Preprocessing

3.2.1 Filtering Airport Arrival Data

The code filters the daily passenger traffic data to include only rows where the control point is "Airport" and the movement is "Arrival." This is done using the following code snippet:

```
airport_arrival_rows = df[(df['Control␣Point'] == 'Airport')
     & (df['Arrival␣/␣Departure'] == 'Arrival')]
airport_arrival_df = pd.DataFrame(airport_arrival_rows)
```

The filtered data is stored in the `airport_arrival_df` DataFrame.

3.2.2 Converting Date Columns

The code converts the "Date" column in the `airport_arrival_df` DataFrame to the datetime format using the `pd.to_datetime()` function. This is done using the following code snippet:

```
airport_arrival_df['Date'] = pd.to_datetime(
    airport_arrival_df['Date'], format='%d-%m-%Y')
```

3.2.3 Grouping and Calculating Averages

The code groups the airport arrival data by month and calculates the average arrivals for each month. This is done separately for Mainland Visitors, Tourist Visitors, and Hong Kong Residents. The code snippet for calculating the average arrivals for tourists is shown below:

```
avg_arrivals_by_month_tourists = airport_arrival_df.groupby(
    ’Month’)[’Tourists’].mean().reset_index()
```

Similar code snippets are used to calculate the average arrivals for Mainland Visitors and Hong Kong Residents.

3.2.4 Filtering Taxi Involvement Data

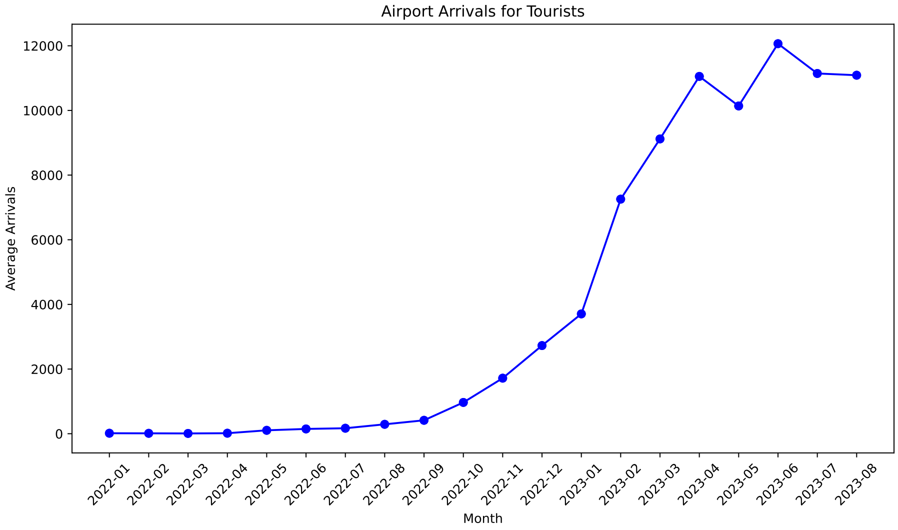
The code filters the taxi involvement data to include only rows where the vehicle class code is 3. This is done using the following code snippet:

```
filtered_data = data[data[’VEHICLE_CLASS_CODE’] == 3]
filtered_df_taxi = filtered_data[filtered_data[’YR_MTH’] > ’
    2021-12-01’]
```

The filtered data is stored in the `filtered_df_taxi` DataFrame.

4 Exploratory Data Analysis

4.1 Airport Arrivals for Tourists

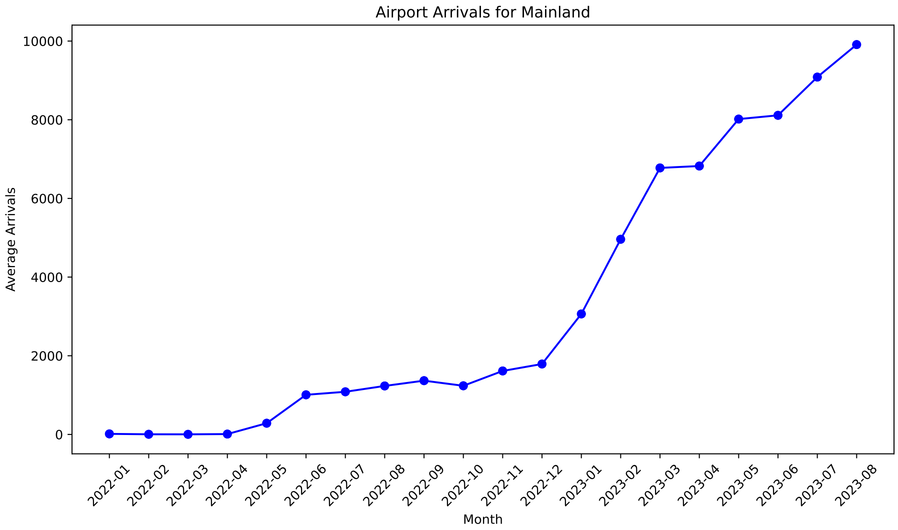


Description: This plot represents the average monthly airport arrivals for tourists.

Visualization: The scatter plot and line plot show the trend and variation in average arrivals for tourists each month.

Insights: By analyzing this plot, you can understand the patterns and fluctuations in tourist arrivals over the specified time period (from January 2022 to August 2023).

4.2 Airport Arrivals for Mainland Visitors

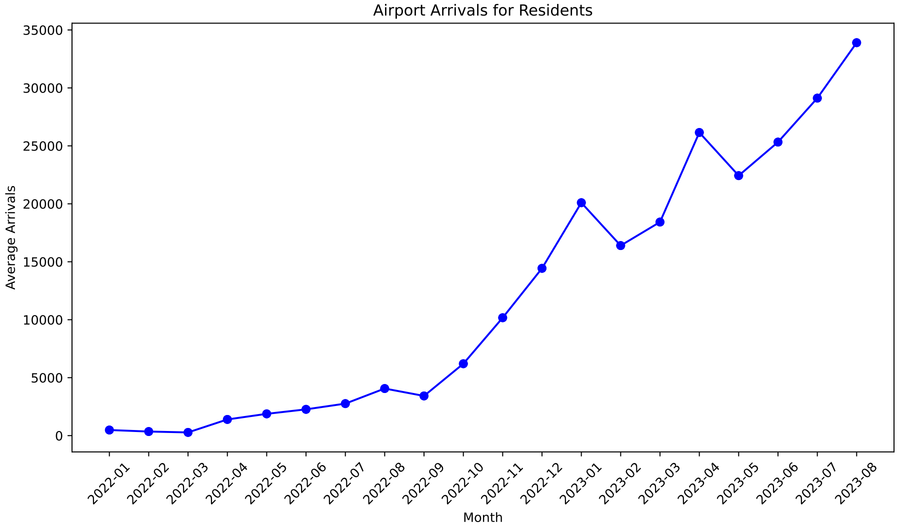


Description: This plot represents the average monthly airport arrivals for Mainland visitors.

Visualization: Similar to the tourists' plot, it uses scatter and line plots to depict the average arrivals for Mainland visitors each month.

Insights: Examining this plot allows you to gain insights into the trends and variations in arrivals of Mainland visitors over the specified time period.

4.3 Airport Arrivals for Hong Kong Residents

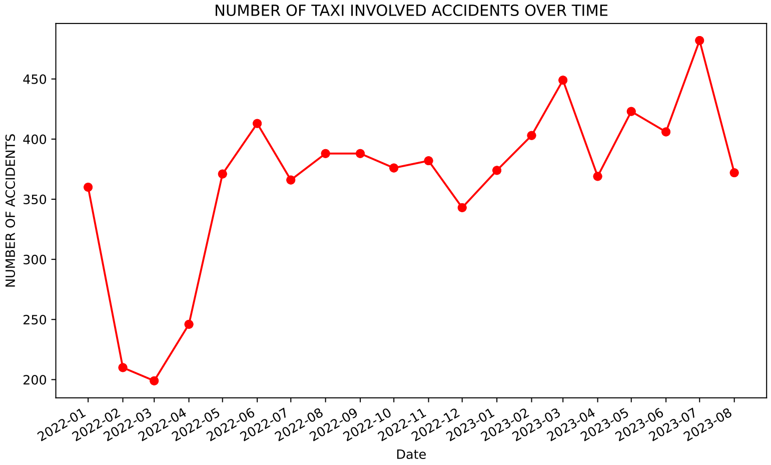


Description: This plot represents the average monthly airport arrivals for Hong Kong residents.

Visualization: It utilizes scatter and line plots to illustrate the average arrivals for Hong Kong residents on a monthly basis.

Insights: Analyzing this plot helps in understanding how the airport arrivals for Hong Kong residents have been changing over the specified time period.

4.4 Number of Taxi Involved in Accidents

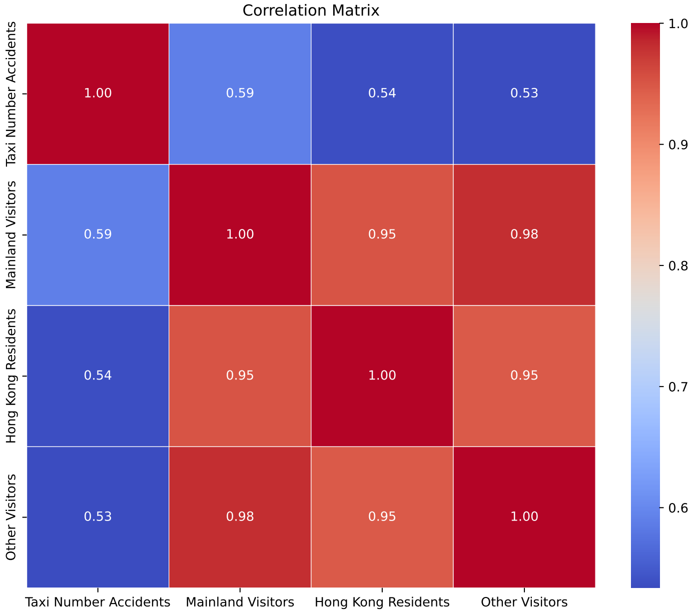


Description: This plot represents the number of taxi-involved accidents over time.

Visualization: It utilizes both scatter and line plots with red dots to illustrate the monthly variation in the number of taxi-involved accidents.

Insights: Analyzing this plot helps in understanding how the frequency of taxi-involved accidents has been changing over the specified time period, with a focus on months after December 2021.

5 Correlation Analysis



Taxi Number Involved in Accident vs. Mainland Visitors (0.588797):

There is a moderate positive correlation (0.588797) between the number of taxi accidents (‘Taxi Number’) and the number of Mainland Visitors. This suggests that as the number of Mainland Visitors increases, the number of taxi accidents tends to increase, but the correlation is not extremely strong.

Taxi Number Involved in Accident vs. Hong Kong Residents (0.536850):

There is a moderate positive correlation (0.536850) between the number of taxi accidents and the number of Hong Kong Residents. This implies that there is a tendency for an increase in taxi accidents as the number of Hong Kong Residents increases.

Taxi Number Involved in Accident vs. Tourists (0.533777):

There is a moderate positive correlation (0.533777) between the number of taxi accidents and the number of Tourists. This suggests a similar pattern as observed with Mainland Visitors and Hong Kong Residents.

Mainland Visitors vs. Hong Kong Residents (0.952173):

There is a very strong positive correlation (0.952173) between the number of Mainland Visitors and Hong Kong Residents. This indicates a close relationship where an increase in Mainland Visitors is highly correlated with an increase in Hong Kong Residents.

Mainland Visitors vs. Tourists (0.980271):

There is a very strong positive correlation (0.980271) between the number of Mainland Visitors and Tourists. This implies a close relationship where an increase in Mainland Visitors is highly correlated with an increase in Tourists.

Hong Kong Residents vs. Tourists (0.945693):

There is a very strong positive correlation (0.945693) between the number of Hong Kong Residents and Tourists. This indicates a close relationship where an increase in Hong Kong Residents is highly correlated with an increase in Tourists.

Overall, the correlation matrix provides insights into the relationships between taxi accidents and the number of visitors from different backgrounds. The strong correlations among the visitor categories suggest a potential common factor influencing their numbers.

6 MCMC and Gibbs Sampling

For simplicity, let's denote:

$$\begin{aligned}
y &: \text{Response variable (e.g., accidents)}\\
X_1 &: \text{Predictor variable 1 (e.g., mainland\_visitors)}\\
X_2 &: \text{Predictor variable 2 (e.g., tourists)}\\
X_3 &: \text{Predictor variable 3 (e.g., locals)}\\
B_1 &: \text{Coefficient for } X_1\\
B_2 &: \text{Coefficient for } X_2\\
B_3 &: \text{Coefficient for } X_3\\
\sigma_{\text{sq}} &: \text{Variance}
\end{aligned}$$

The likelihood function for a linear regression model is assumed to follow a normal distribution:

$$P(y|B_1, B_2, B_3, \sigma_{\text{sq}}) = \frac{1}{\sqrt{2\pi\sigma_{\text{sq}}}} \exp\left(-\frac{(y - (B_1X_1 + B_2X_2 + B_3X_3))^2}{2\sigma_{\text{sq}}}\right)$$

The prior distributions for the coefficients are assumed to be normal distributions:

$$\begin{aligned}
P(B_1) &= \frac{1}{\sqrt{2\pi\sigma_{B_1}^2}} \exp\left(-\frac{B_1^2}{2\sigma_{B_1}^2}\right)\\
P(B_2) &= \frac{1}{\sqrt{2\pi\sigma_{B_2}^2}} \exp\left(-\frac{B_2^2}{2\sigma_{B_2}^2}\right)\\
P(B_3) &= \frac{1}{\sqrt{2\pi\sigma_{B_3}^2}} \exp\left(-\frac{B_3^2}{2\sigma_{B_3}^2}\right)
\end{aligned}$$

The posterior distributions for the coefficients are then proportional to the product of the likelihood and the priors:

$$\begin{aligned}
P(B_1|y, X_1, X_2, X_3) &\propto P(y|B_1, B_2, B_3, \sigma_{\text{sq}}) \times P(B_1)\\
P(B_2|y, X_1, X_2, X_3) &\propto P(y|B_1, B_2, B_3, \sigma_{\text{sq}}) \times P(B_2)\\
P(B_3|y, X_1, X_2, X_3) &\propto P(y|B_1, B_2, B_3, \sigma_{\text{sq}}) \times P(B_3)
\end{aligned}$$

Then, we use Metropolis-Hastings algorithm to sample.

Note: Normal distribution is used as the proposal distribution for each parameter.

1. **Initialization of Parameters:**

```
    B1 = 0  # Mainland Visitors
    B2 = 0  # Tourists
    B3 = 0  # Residents
```

   The parameters $B1$, $B2$, and $B3$ are initialized to zero. These parameters represent coefficients associated with Mainland Visitors, Tourists, and Residents, respectively.

2. **Setting Up Iterations and Data:**

```
    N_iter = 1000
    samples = np.zeros((N_iter, 3))
    accidents = filtered_df_taxi[’NO_VEHICLE’]
    mainland_visitors = avg_arrivals_by_month_mainland[’
       Mainland␣Visitors’]
    tourists = avg_arrivals_by_month_tourists[’Tourists’
       ]
    locals = avg_arrivals_by_month_residents[’Hong␣Kong␣
       Residents’]
    N = len(accidents)
```

   The number of iterations (N_{iter}) is set to 1000, and placeholders are created to store samples. Relevant data such as accidents, mainland visitors, tourists, and locals are loaded from dataframes.

3. **Markov Chain Monte Carlo (MCMC) Sampling:**

```
    for i in range(N_iter):
        # Sampling B1
        # ... (update B1 based on sampled values)

        # Sampling B2
        # ... (update B2 based on sampled values)

        # Sampling B3
        # ... (update B3 based on sampled values)

        samples[i, :] = [B1, B2, B3]
```

   The MCMC loop iteratively updates parameters ($B1$, $B2$, $B3$) based on sampled values using normal distribution. The update formulas involve calculations related to data.

4. **Burn-In Period and Posterior Distributions:**

```
    burn_in = int(N_iter * 0.1)
    posterior_B1 = samples[burn_in:, 0]
    posterior_B2 = samples[burn_in:, 1]
    posterior_B3 = samples[burn_in:, 2]
```

A burn-in period is defined (discarding initial samples), and posterior distributions of parameters ($B1$, $B2$, $B3$) are extracted from MCMC samples.

6.1 Results

Parameter: $B1$ **Posterior**

- Effect on Accidents (1.0 Positive to 0.0 → Negative) = 0.45222222222222225
- Mean: -1.654735237092731e-05
- Median: -5.1664493283096606e-05
- Standard deviation: 0.0004238580034395113
- Interquartile range: 0.0006459896115224824

Parameter: $B2$ **Posterior**

- Effect on Accidents (1.0 Positive to 0.0 → Negative) = 0.5044444444444445
- Mean: 5.62337158564161e-06
- Median: 6.334585317296868e-06
- Standard deviation: 0.00026982959376700607
- Interquartile range: 0.0003754271059697741

Parameter: $B3$ **Posterior**

- Effect on Accidents (1.0 Positive to 0.0 → Negative) = 0.5766666666666667
- Mean: 2.7350132733917572e-06
- Median: 1.1096929669754101e-05
- Standard deviation: 6.405233217248062e-05
- Interquartile range: 6.901099954962191e-05

Parameter: $B1$

- HPDI: [-0.00072511, 0.00076627]
- $B1$ lies within the 95% HPDI.

Parameter: $B2$

- HPDI: [-0.00045037, 0.00051407]
- $B2$ lies within the 95% HPDI.

Parameter: $B3$

- HPDI: [-0.00011837, 0.00013067]
- $B3$ lies within the 95% HPDI.

6.2 Outcome

6.2.1 Parameter: B1 Posterior

- Effect on Accidents (1.0 Positive to 0.0 → Negative): 0.452
- Mean: -1.6547×10^{-5}
- Median: -5.1664×10^{-5}
- Standard deviation: 0.0004239
- Interquartile range: 0.000646

6.2.2 Parameter: B2 Posterior

- Effect on Accidents (1.0 Positive to 0.0 → Negative): 0.504
- Mean: 5.6234×10^{-6}
- Median: 6.3346×10^{-6}
- Standard deviation: 0.0002698
- Interquartile range: 0.0003754

6.2.3 Parameter: B3 Posterior

- Effect on Accidents (1.0 Positive to 0.0 → Negative): 0.577
- Mean: 2.7350×10^{-6}
- Median: 1.1097×10^{-5}
- Standard deviation: 6.4052×10^{-5}
- Interquartile range: 6.9011×10^{-5}

6.2.4 Parameter: B1

- HPDI: $[-0.00072511, 0.00076627]$
- B1 lies within the 95% HPDI.

6.2.5 Parameter: B2

- HPDI: $[-0.00045037, 0.00051407]$
- B2 lies within the 95% HPDI.

6.2.6 Parameter: B3

- HPDI: $[-0.00011837, 0.00013067]$
- B3 lies within the 95% HPDI.

7 MCMC and VI

Likelihood Function:

$$P(\mathbf{y} \mid B_1, B_2, B_3, \sigma_{\text{sq}}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{\text{sq}}^2}} \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2\sigma_{\text{sq}}^2}\right)$$

where:

$\mathbf{y}$ is the vector of observed response variables,
$\hat{y}_i$ is the predicted value for the i-th observation,
σ_{sq} is the standard deviation of the likelihood.

Prior Distributions:

Note: B_1, B_2, and B_3 are assumed to be normal distributions.

$$P(B_1) = \frac{1}{\sqrt{2\pi\sigma_{B_1}^2}} \exp\left(-\frac{B_1^2}{2\sigma_{B_1}^2}\right)$$

$$P(B_2) = \frac{1}{\sqrt{2\pi\sigma_{B_2}^2}} \exp\left(-\frac{B_2^2}{2\sigma_{B_2}^2}\right)$$

$$P(B_3) = \frac{1}{\sqrt{2\pi\sigma_{B_3}^2}} \exp\left(-\frac{B_3^2}{2\sigma_{B_3}^2}\right)$$

where:

$\sigma_{B_1}^2, \sigma_{B_2}^2,$ and $\sigma_{B_3}^2$ are the variances of the priors for $B_1, B_2,$ and B_3 respectively.

Posterior Distributions:

$$P(B_1 \mid \mathbf{y}, X_1, X_2, X_3) \propto P(\mathbf{y} \mid B_1, B_2, B_3, \sigma_{\text{sq}}) \times P(B_1)$$
$$P(B_2 \mid \mathbf{y}, X_1, X_2, X_3) \propto P(\mathbf{y} \mid B_1, B_2, B_3, \sigma_{\text{sq}}) \times P(B_2)$$
$$P(B_3 \mid \mathbf{y}, X_1, X_2, X_3) \propto P(\mathbf{y} \mid B_1, B_2, B_3, \sigma_{\text{sq}}) \times P(B_3)$$

Optimization:

The code uses the Adam optimizer to minimize the negative log joint probability, which is equivalent to maximizing the log joint probability.

Variational Inference:

The final step involves using variational inference to obtain samples from the posterior distribution.

1. **Data Loading:**

```
accidents = filtered_df_taxi['NO_VEHICLE']
mainland_visitors = avg_arrivals_by_month_mainland['
    Mainland␣Visitors']
tourists = avg_arrivals_by_month_tourists['Tourists']
locals = avg_arrivals_by_month_residents['Hong␣Kong␣
    Residents']
```

Listing 1: Data Loading

This part loads the necessary data: `accidents`, `mainland_visitors`, `tourists`, and `locals` from specific columns of DataFrames (`filtered_df_taxi`, `avg_arrivals_by_month_mainland`, `avg_arrivals_by_month_tourists`, `avg_arrivals_by_month_reside`

2. **TensorFlow Probability Setup:**

```
tfd = tfp.distributions
```

Listing 2: TensorFlow Probability Setup

This line creates a shorthand reference (`tfd`) for the `tensorflow_probability.distributions` module.

3. **Variable Initialization:**

```
B1 = tf.Variable(0.0, dtype=tf.float32)
B2 = tf.Variable(0.0, dtype=tf.float32)
B3 = tf.Variable(0.0, dtype=tf.float32)
```

Listing 3: Variable Initialization

Three TensorFlow variables (`B1`, `B2`, and `B3`) are initialized as trainable parameters with an initial value of `0.0` and data type `tf.float32`.

4. **Log Joint Probability Function:**

```
def log_joint_prob(B1, B2, B3):
    # ... (Normalization of data)
    # ... (Model using normalized data)
    # ... (Calculation of log likelihood)
    # ... (Priors for B1, B2, B3)
    return (prior_B1.log_prob(B1) + prior_B2.log_prob(B2) + prior_B3.log_prob(B3) + log_likelihood)
```

Listing 4: Log Joint Probability Function

This function calculates the log joint probability of the Bayesian model, including the log likelihood and prior terms for the parameters `B1`, `B2`, and `B3`.

5. **Loss Function:**

```
@tf.function
def loss():
    return -log_joint_prob(B1, B2, B3)
```

Listing 5: Loss Function

The loss function is defined as the negation of the log joint probability. The `@tf.function` decorator is used to convert the function into a TensorFlow graph for optimization.

6. **Optimizer Setup:**

```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.01)
```

Listing 6: Optimizer Setup

An Adam optimizer with a learning rate of `0.01` is initialized.

7. **Training Loop:**

```
N_iter = 1000
for i in range(N_iter):
    # ... (Gradient Tape to compute gradients)
    # ... (Optimizer applies gradients to update parameters)
```

Listing 7: Training Loop

A training loop runs for `N_iter` iterations. Inside the loop, a gradient tape is used to compute gradients of the loss with respect to the parameters, and the optimizer applies these gradients to update the parameters.

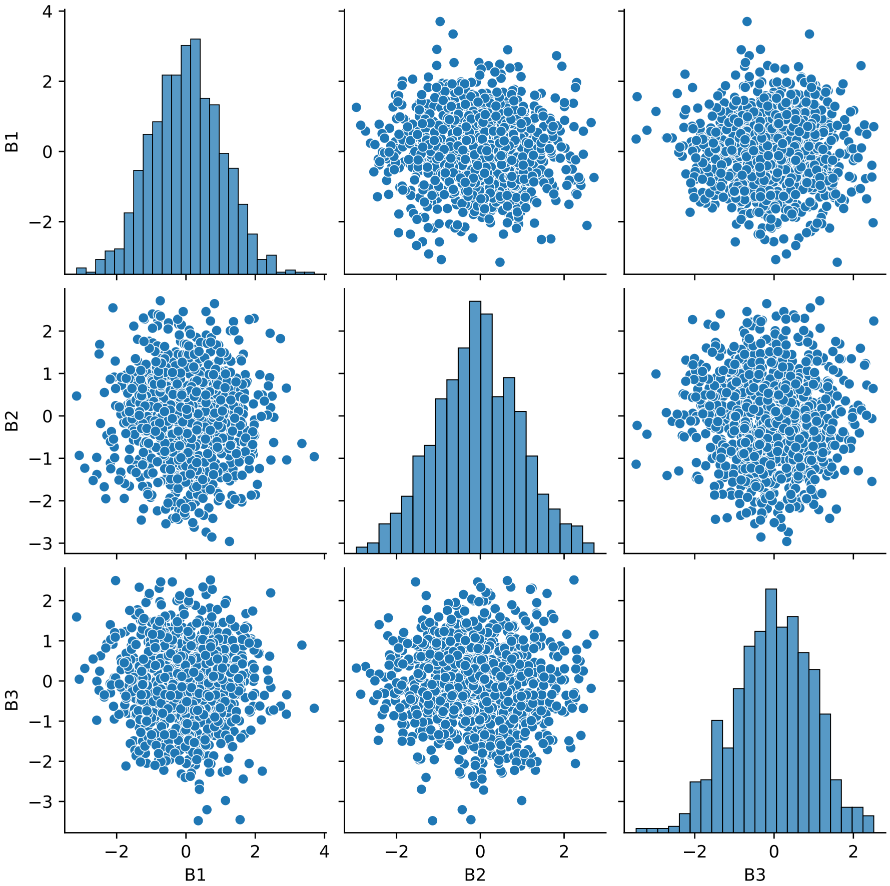
8. **Posterior Distribution Sampling:**

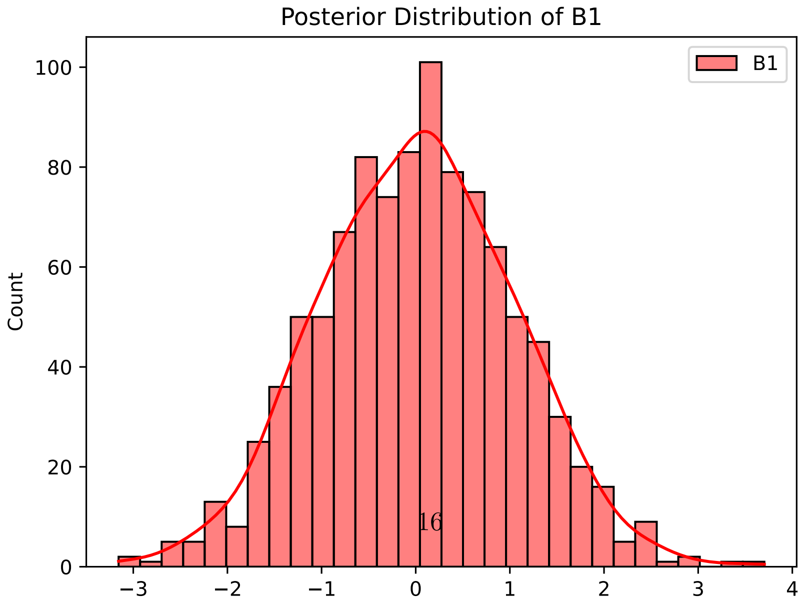
```
variational_distribution = tfd.MultivariateNormalDiag(loc=[B1, B2, B3], scale_diag=[1.0, 1.0, 1.0])
posterior_samples = variational_distribution.sample(N_iter)
```

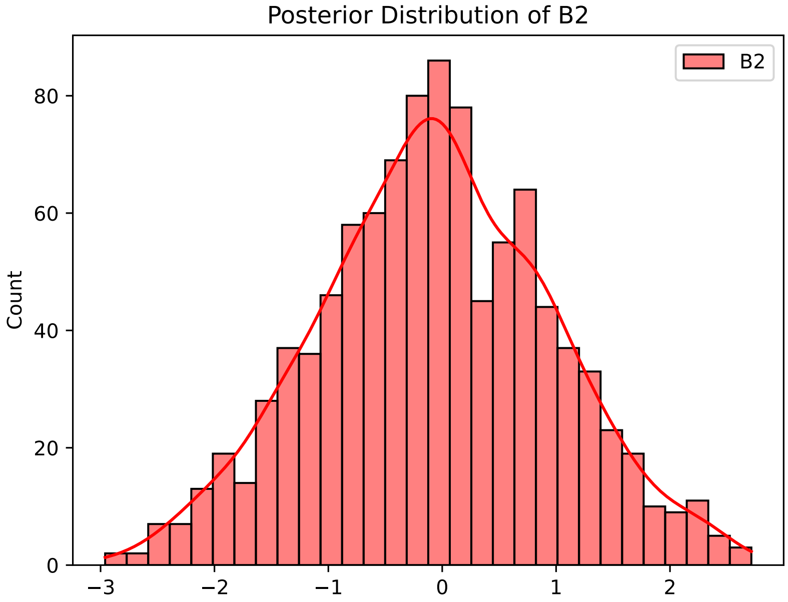
Listing 8: Posterior Distribution Sampling

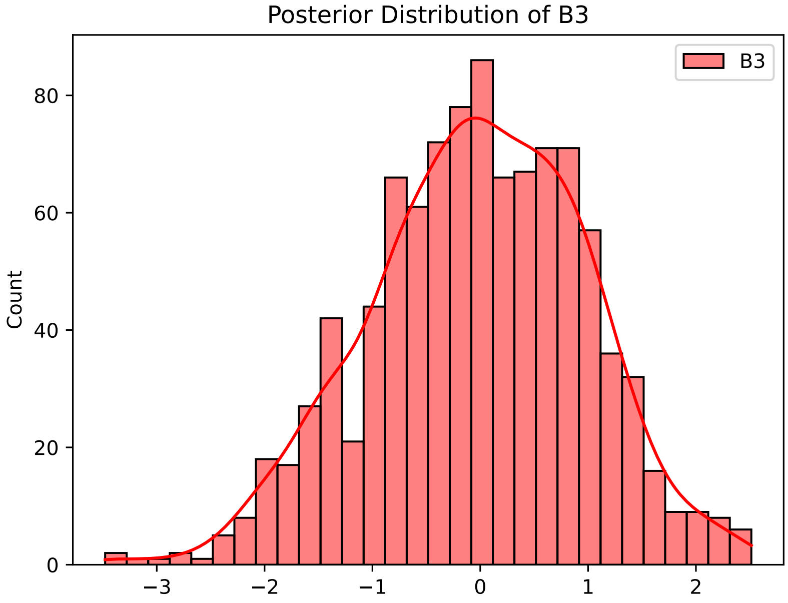
A multivariate normal distribution is defined based on the learned parameters, and samples from this distribution are drawn. These samples represent the posterior distribution of the parameters.

7.1 Results


B1
B2
B3
4
2
0
−2
−3
−1
1
−2
0
2
4
B1
−2
0
2
B2
−2
0
2
B3


Posterior Distribution of B1
B1
Count
100
80
60
40
20
0
−3
−2
−1
0
1
2
3
4


Posterior Distribution of B2
B2
Count
0
20
40
60
80
−3
−2
−1
0
1
2


Posterior Distribution of B3
B3
Count
0
20
40
60
80
−3
−2
−1
0
1
2

7.2 Outcome

Mean:

The mean values represent the central tendency of your posterior distribution for each parameter.

$$B1 : 0.06775$$
$$B2 : -0.03778$$
$$B3 : 0.01665$$

Standard Deviation:

The standard deviation provides a measure of the spread or uncertainty in your parameter estimates.

$$B1 : 0.97111$$
$$B2 : 0.99863$$
$$B3 : 0.97493$$

95% Credible Interval:

The 95% credible interval indicates the range in which you are 95% confident that the true parameter value lies.

$$\text{For } B1 : [-1.85397, 1.98181]$$
$$\text{For } B2 : [-1.97905, 2.08504]$$
$$\text{For } B3 : [-1.86814, 1.93454]$$

Interpretation:

It seems that the mean values are close to zero for each parameter. It shows that the predictors may not have a strong impact on the response variable. However, the wide credible intervals indicate significant uncertainty in these estimates.