```
!pip install pyLDAvis
```

```
Collecting pyLDAvis
  Downloading pyLDAvis-3.4.1-py3-none-any.whl (2.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.6/2.6 MB 10.5 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.24.2 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.11.4)
Requirement already satisfied: pandas>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (2.0.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.3.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (3.1.3)
Requirement already satisfied: numexpr in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (2.9.0)
Collecting funcy (from pyLDAvis)
  Downloading funcy-2.0-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: scikit-learn>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.2.2)
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (4.3.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (67.7.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.0.0->pyLDAvis) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.0.0->pyLDAvis) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.0.0->pyLDAvis) (2024.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->pyLDAvis) (3.4.0)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim->pyLDAvis) (6.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->pyLDAvis) (2.1.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=2.0.0->pyLDAvis) (1.16.0)
Installing collected packages: funcy, pyLDAvis
Successfully installed funcy-2.0 pyLDAvis-3.4.1
```

```
import numpy as np
import pandas as pd
import ast
import matplotlib.pyplot as plt
import seaborn as sns

import pymc as pm
import arviz as az

from gensim import corpora
from gensim.models import LdaModel

import nltk
nltk.download('stopwords')

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS

import pyLDAvis
import pyLDAvis.gensim_models
pyLDAvis.enable_notebook()

import warnings
warnings.filterwarnings('ignore')
```

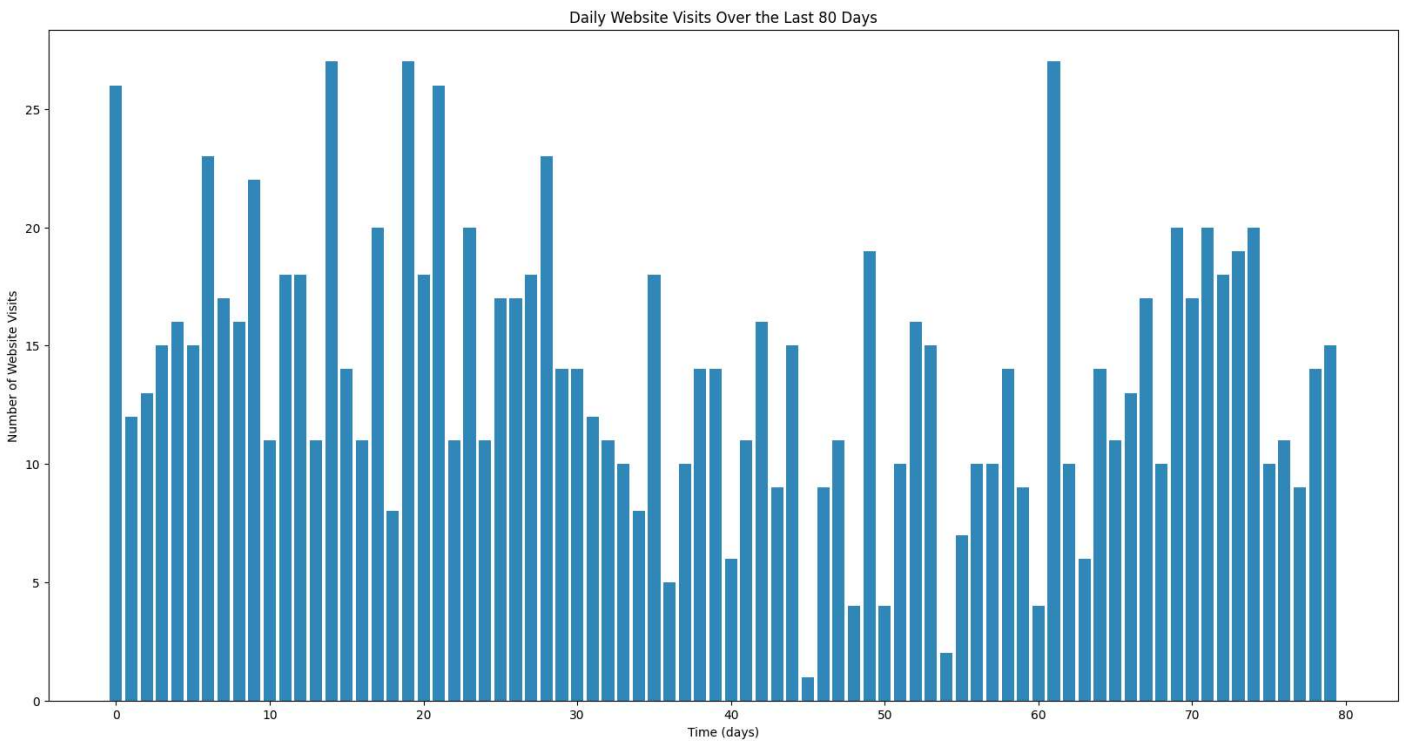
```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Q1

```
# DO NOT MODIFY THIS CODE
website_visits_data = np.loadtxt('https://raw.githubusercontent.com/MIE223-2024/course-datasets/main/website_visits.csv', delimiter=',')
```

(a)

```
# YOUR CODE STARTS HERE
plt.figure(figsize=(20, 10))
plt.bar(np.arange(len(website_visits_data)), website_visits_data, color="#348ABD")
plt.xlabel('Time (days)')
plt.ylabel('Number of Website Visits')
plt.title('Daily Website Visits Over the Last 80 Days')
plt.show()
```



Enter your answer here: The visualization suggests multiple switchpoints, as indicated by the apparent changes in the level of website visits over time, with periods of relative consistency followed by abrupt changes in the frequency of visits.

(b)

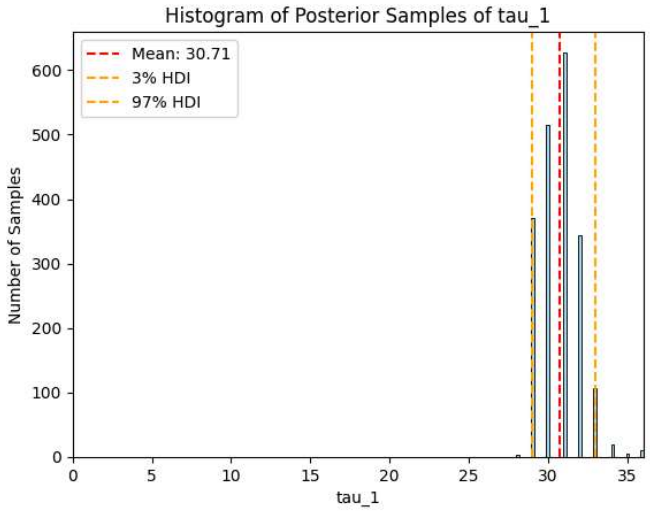
```
# DO NOT MODIFY THIS CODE

param_names = ['tau_1', 'tau_2', 'lambda_1', 'lambda_2', 'lambda_3']

double_switchpoint_model = pm.Model()

with double_switchpoint_model:

  # Define priors
  # YOUR CODE STARTS HERE (FILL IN THE BLANKS)
  alpha = 1.0 / np.mean(website_visits_data)

  tau_1 = pm.DiscreteUniform("tau_1", lower=0, upper=79)
  tau_2 = pm.DiscreteUniform("tau_2", lower=0, upper=79)
  lambda_1 = pm.Exponential("lambda_1", alpha)
  lambda_2 = pm.Exponential("lambda_2", alpha)
  lambda_3 = pm.Exponential("lambda_3", alpha)

  idx = np.arange(len(website_visits_data))
  lambda_ = pm.math.switch(idx < tau_1, lambda_1, pm.math.switch(idx < tau_2, lambda_2, lambda_3))

  # Define likelihood (conditioned on observed data)
  # YOUR CODE STARTS HERE (FILL IN THE BLANK)
  likelihood = pm.Poisson("visits", lambda_, observed=website_visits_data)

  # Sample from posterior
  # YOUR CODE STARTS HERE (FILL IN THE BLANK)
  trace = pm.sample(1000, random_seed=45)
```

```
100.00% [2000/2000 00:04<00:00 Sampling chain 0, 0 divergences]
100.00% [2000/2000 00:05<00:00 Sampling chain 1, 0 divergences]
```

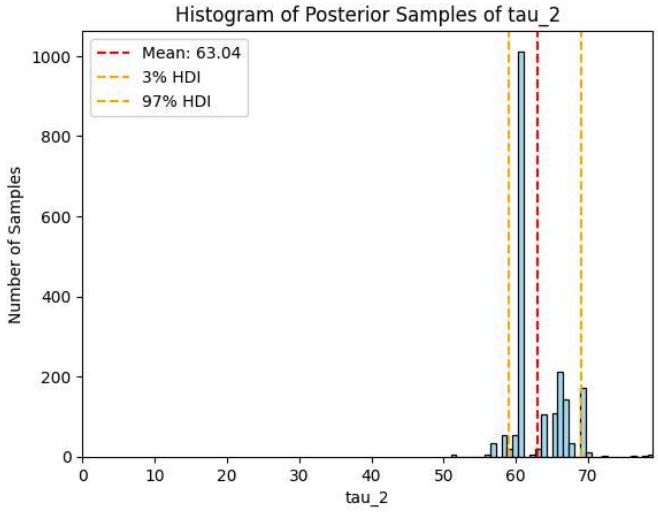
(c)

```
# Function to: (a) plot histogram of posterior samples (b) display mean and 94% HDI of posterior samples

def plot_posterior(param_names, trace):

    # (a) Plot histogram of posterior samples of passed parameter
    # YOUR CODE STARTS HERE
    samples = trace.posterior[param_name].values.flatten()
    ax = sns.histplot(samples, color='skyblue')
    ax.set_xlim(0, samples.max())
    plt.xlabel(param_name)
    plt.ylabel('Number of Samples')
    plt.title(f'Histogram of Posterior Samples of {param_name}')
    mean_sample = np.mean(samples)
    hdi_sample = az.hdi(samples, hdi_prob=0.94)
    plt.axvline(x=mean_sample, color='red', linestyle='--', label=f'Mean: {mean_sample:.2f}')
    plt.axvline(x=hdi_sample[0], color='orange', linestyle='--', label='3% HDI')
    plt.axvline(x=hdi_sample[1], color='orange', linestyle='--', label='97% HDI')
    plt.legend()

    # (b) Display mean and 94% HDI of posterior samples of passed parameter (round to 2 decimal places)
    # YOUR CODE STARTS HERE
    print(f"\nMean of {param_name}: {mean_sample}")
    print(f"94% HDI of {param_name}: [{hdi_sample[0]:.2f}, {hdi_sample[1]:.2f}]")

    return None


# DO NOT MODIFY THIS CODE
for param_name in param_names:
    plot_posterior(param_name, trace)
    plt.show()
```

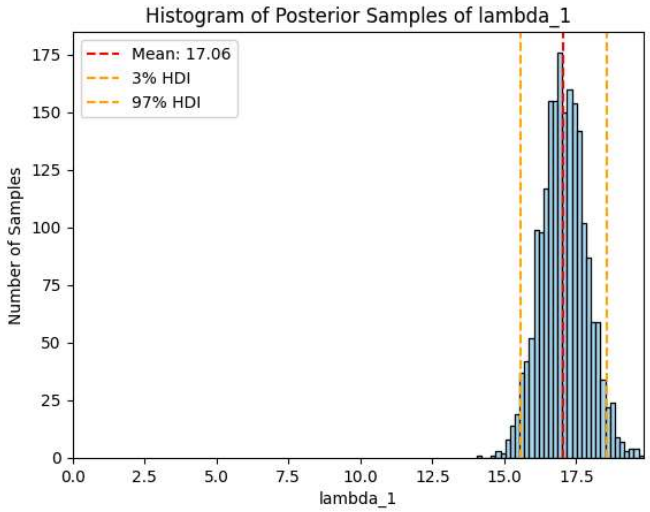
```
Mean of tau_1: 30.7095
94% HDI of tau_1: [29.00, 33.00]
```



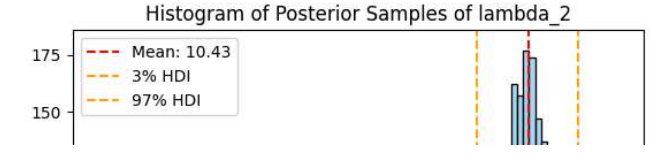

```
Mean of tau_2: 63.0425
94% HDI of tau_2: [59.00, 69.00]
```

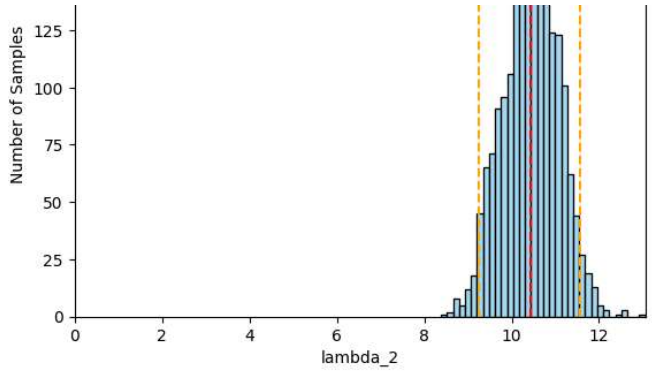



```
Mean of lambda_1: 17.057806481093003
94% HDI of lambda_1: [15.57, 18.54]
```

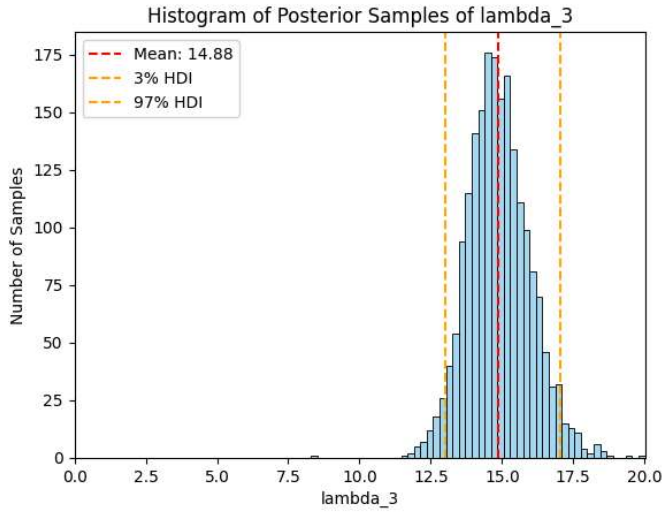



```
Mean of lambda_2: 10.430343240989755
94% HDI of lambda_2: [9.25, 11.56]
```






```
Mean of lambda_3: 14.88168074631954
94% HDI of lambda_3: [12.99, 17.04]
```



Enter answer here: (i) The current model may not be a good fit since the HDI ranges for the switchpoints (tau_1 and tau_2) are narrow and indicate certainty in switchpoint locations that does not reflect the apparent variability and overlap in the count data from the bar graph.

(ii) The histograms of the posterior samples for lambda parameters seem narrowly distributed, which suggests a higher level of certainty about the Poisson rates than what the variability in the bar graph of the observed data might suggest. This could imply that the model is overly confident about the rate parameters and might not be capturing the true underlying process accurately.

(d)

```
# DO NOT MODIFY THIS CODE

double_switchpoint_model_adjusted = pm.Model()

with double_switchpoint_model_adjusted:

  # Define priors
  # YOUR CODE STARTS HERE
  alpha = 1.0 / np.mean(website_visits_data)
  tau_1 = pm.DiscreteUniform("tau_1", lower=0, upper=79)
  tau_2 = pm.DiscreteUniform("tau_2", lower=tau_1, upper=79)  # hyperprior: tau_2 depends on tau_1
  lambda_1 = pm.Exponential("lambda_1", alpha)
  lambda_2 = pm.Exponential("lambda_2", alpha)
  lambda_3 = pm.Exponential("lambda_3", alpha)
  idx = np.arange(len(website_visits_data))
  lambda_ = pm.math.switch(idx < tau_1, lambda_1, pm.math.switch(idx < tau_2, lambda_2, lambda_3))

  # Define likelihood (conditioned on observed data)
  # YOUR CODE STARTS HERE
  likelihood = pm.Poisson('visits', mu=lambda_, observed=website_visits_data)

  # Sample from posterior
  # YOUR CODE STARTS HERE
  trace_adjusted = pm.sample(1000, random_seed=45)
```

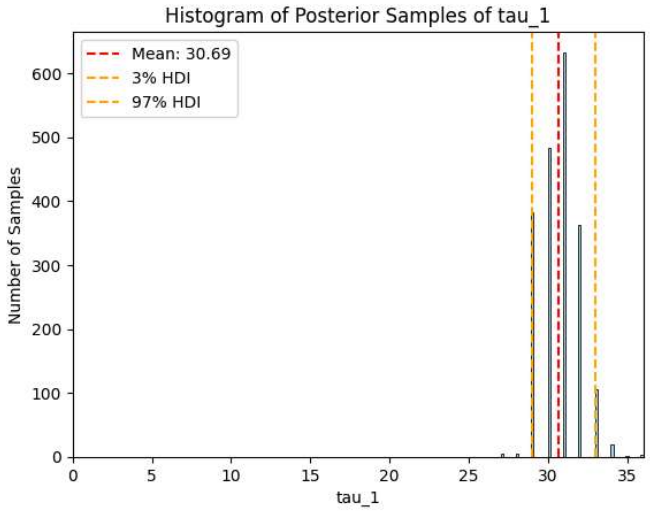
```
100.00% [2000/2000 00:03<00:00 Sampling chain 0, 0 divergences]
100.00% [2000/2000 00:03<00:00 Sampling chain 1, 0 divergences]
```

Enter your answer here: The lower bound of tau_2 is equal to tau_1.

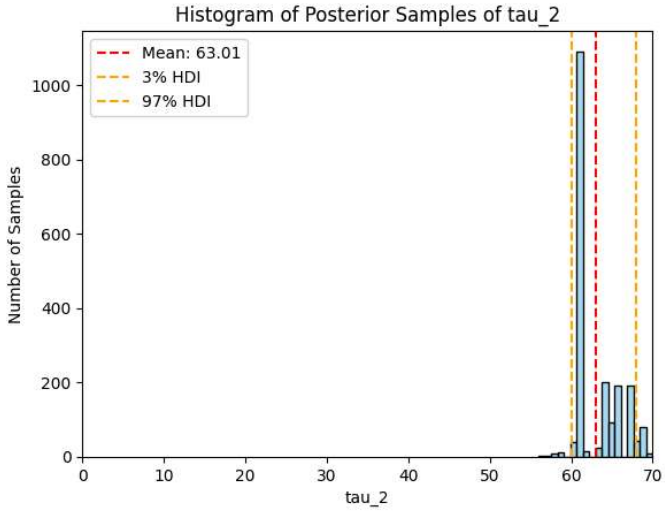
(e)

```
# YOUR CODE GOES HERE
for param_name in param_names:
    plot_posterior(param_name, trace_adjusted)
    plt.show()
```

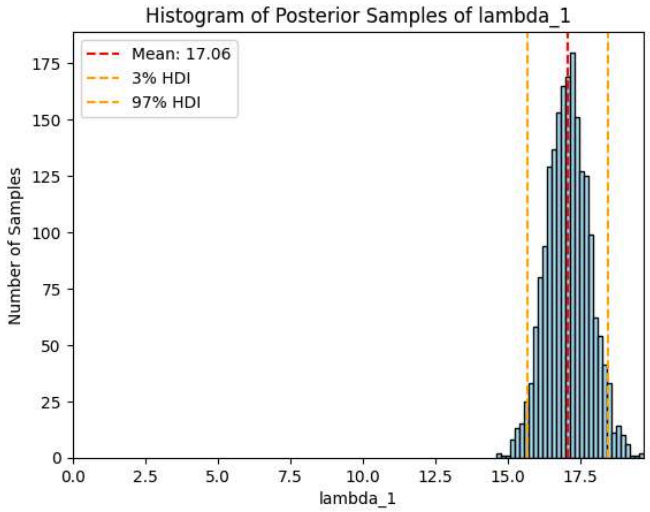
```
Mean of tau_1: 30.687
94% HDI of tau_1: [29.00, 33.00]
```



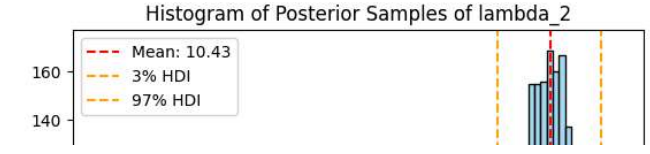

```
Mean of tau_2: 63.0095
94% HDI of tau_2: [60.00, 68.00]
```

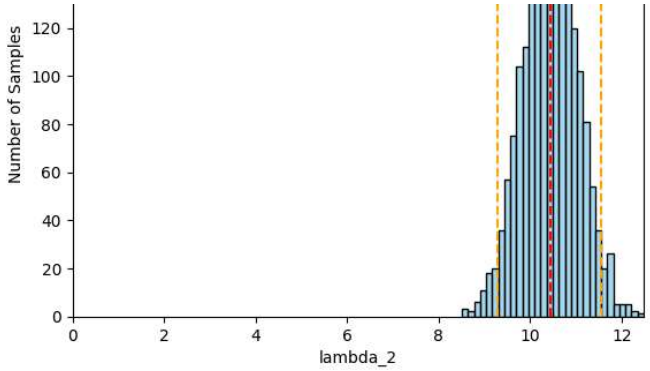



```
Mean of lambda_1: 17.063852209041627
94% HDI of lambda_1: [15.69, 18.45]
```

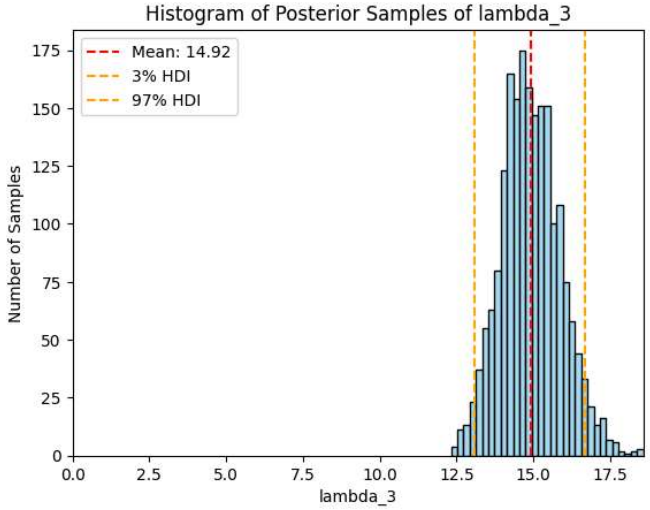



```
Mean of lambda_2: 10.43225281868465
94% HDI of lambda_2: [9.28, 11.54]
```






```
Mean of lambda_3: 14.923616208641993
94% HDI of lambda_3: [13.09, 16.69]
```



Enter answer here: (i) The histograms for tau_1 and tau_2 indicate narrow HDI ranges, suggesting that the model is quite certain about the location of these switchpoints, which may or may not be consistent with the actual data variability.

(ii) Based on the bar graph trends observed earlier, we would expect a wider range of possible values for the lambda parameters, indicating multiple plausible rates of website visits. The narrow distributions here could suggest that the model does not fully capture the possible variability in the visit rates, and the posterior samples might not align with the data as well as they should.

Q2

```
topic_df = pd.read_csv('https://raw.githubusercontent.com/MIE223-2024/course-datasets/main/topic_dataset.csv')
```