h2xvnhvvq

March 13, 2025

[ ]:
```
!wget --no-check-certificate https://drive.google.com/
↪uc?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N -O kerala.csv
```

```
--2025-03-13 15:59:25--
https://drive.google.com/uc?id=1Mp2bQl5QJ6O2tcezb0ceBQIn8vW5us0N
Resolving drive.google.com (drive.google.com)… 142.251.2.101, 142.251.2.102,
142.251.2.139, …
Connecting to drive.google.com (drive.google.com)|142.251.2.101|:443…
connected.
HTTP request sent, awaiting response… 303 See Other
Location: https://drive.usercontent.google.com/download?id=1Mp2bQl5QJ6O2tcezb0ce
BQIn8vW5us0N [following]
--2025-03-13 15:59:25--  https://drive.usercontent.google.com/download?id=1Mp2bQ
l5QJ6O2tcezb0ceBQIn8vW5us0N
Resolving drive.usercontent.google.com (drive.usercontent.google.com)…
74.125.137.132, 2607:f8b0:4023:c0d::84
Connecting to drive.usercontent.google.com
(drive.usercontent.google.com)|74.125.137.132|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 10300 (10K) [application/octet-stream]
Saving to: ‘kerala.csv’

kerala.csv          100%[===================>]  10.06K  --.-KB/s    in 0s

2025-03-13 15:59:28 (61.9 MB/s) - ‘kerala.csv’ saved [10300/10300]
```

[ ]:
```
import numpy as np
import pandas as pd
```

0.0.1 Reading the Dataset

[ ]:
```
df = pd.read_csv("kerala.csv")
df.head(10)
```

[ ]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	\
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	

1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6
5	KERALA	1906	26.7	7.4	9.9	59.4	160.8	414.9	954.2	442.8
6	KERALA	1907	18.8	4.8	55.7	170.8	101.4	770.9	760.4	981.5
7	KERALA	1908	8.0	20.8	38.2	102.9	142.6	592.6	902.2	352.9
8	KERALA	1909	54.1	11.8	61.3	93.8	473.2	704.7	782.3	258.0
9	KERALA	1910	2.7	25.7	23.3	124.5	148.8	680.0	484.1	473.8

	SEP	OCT	NOV	DEC	ANNUAL RAINFALL	FLOODS
0	197.7	266.9	350.8	48.4	3248.6	YES
1	491.6	358.4	158.3	121.5	3326.6	YES
2	341.8	354.1	157.0	59.0	3271.2	YES
3	222.7	328.1	33.9	3.3	3129.7	YES
4	217.2	383.5	74.4	0.2	2741.6	NO
5	131.2	251.7	163.1	86.0	2708.0	NO
6	225.0	309.7	219.1	52.8	3671.1	YES
7	175.9	253.3	47.9	11.0	2648.3	NO
8	195.4	212.1	171.1	32.3	3050.2	YES
9	248.6	356.6	280.4	0.1	2848.6	NO

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 118 entries, 0 to 117
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   SUBDIVISION       118 non-null    object
 1   YEAR              118 non-null    int64
 2   JAN               118 non-null    float64
 3   FEB               118 non-null    float64
 4   MAR               118 non-null    float64
 5   APR               118 non-null    float64
 6   MAY               118 non-null    float64
 7   JUN               118 non-null    float64
 8   JUL               118 non-null    float64
 9   AUG               118 non-null    float64
 10  SEP               118 non-null    float64
 11  OCT               118 non-null    float64
 12  NOV               118 non-null    float64
 13  DEC               118 non-null    float64
 14   ANNUAL RAINFALL  118 non-null    float64
 15  FLOODS            118 non-null    object
dtypes: float64(13), int64(1), object(2)
memory usage: 14.9+ KB
```

[ ]:
```
df.describe()
```

[ ]:

	YEAR	JAN	FEB	MAR	APR	\
count	118.000000	118.000000	118.000000	118.000000	118.000000	
mean	1959.500000	12.218644	15.633898	36.670339	110.330508	
std	34.207699	15.473766	16.406290	30.063862	44.633452	
min	1901.000000	0.000000	0.000000	0.100000	13.100000	
25%	1930.250000	2.175000	4.700000	18.100000	74.350000	
50%	1959.500000	5.800000	8.350000	28.400000	110.400000	
75%	1988.750000	18.175000	21.400000	49.825000	136.450000	
max	2018.000000	83.500000	79.000000	217.200000	238.000000	

	MAY	JUN	JUL	AUG	SEP	\
count	118.000000	118.000000	118.000000	118.000000	118.000000	
mean	228.644915	651.617797	698.220339	430.369492	246.207627	
std	147.548778	186.181363	228.988966	181.980463	121.901131	
min	53.400000	196.800000	167.500000	178.600000	41.300000	
25%	125.050000	535.550000	533.200000	316.725000	155.425000	
50%	184.600000	625.600000	691.650000	386.250000	223.550000	
75%	264.875000	786.975000	832.425000	500.100000	334.500000	
max	738.800000	1098.200000	1526.500000	1398.900000	526.700000	

	OCT	NOV	DEC	ANNUAL RAINFALL
count	118.000000	118.000000	118.000000	118.000000
mean	293.207627	162.311017	40.009322	2925.405085
std	93.705253	83.200485	36.676330	452.169407
min	68.500000	31.500000	0.100000	2068.800000
25%	222.125000	93.025000	10.350000	2613.525000
50%	284.300000	152.450000	31.100000	2934.300000
75%	355.150000	218.325000	54.025000	3170.400000
max	567.900000	365.600000	202.300000	4473.000000

[ ]:
```
df.isnull().sum().sum()
```

[ ]: 0

[ ]:
```
df.duplicated().sum()
```

[ ]: 0

Insights

The dataset looks perfect except for only one column - ANNUAL RAINFAL.

Action Item:

ANNUAL RAINFALL column needs to strip off the extra spaces and joined using an "_"

0.0.2 Data Cleaning

[ ]:
```
df.columns = [c.replace(' ANNUAL RAINFALL', 'ANNUAL_RAINFALL') for c in df.
 ↪columns]

df.head()
```

[ ]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	\
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	

	SEP	OCT	NOV	DEC	ANNUAL_RAINFALL	FLOODS
0	197.7	266.9	350.8	48.4	3248.6	YES
1	491.6	358.4	158.3	121.5	3326.6	YES
2	341.8	354.1	157.0	59.0	3271.2	YES
3	222.7	328.1	33.9	3.3	3129.7	YES
4	217.2	383.5	74.4	0.2	2741.6	NO

0.0.3 Exploratory Data Analysis

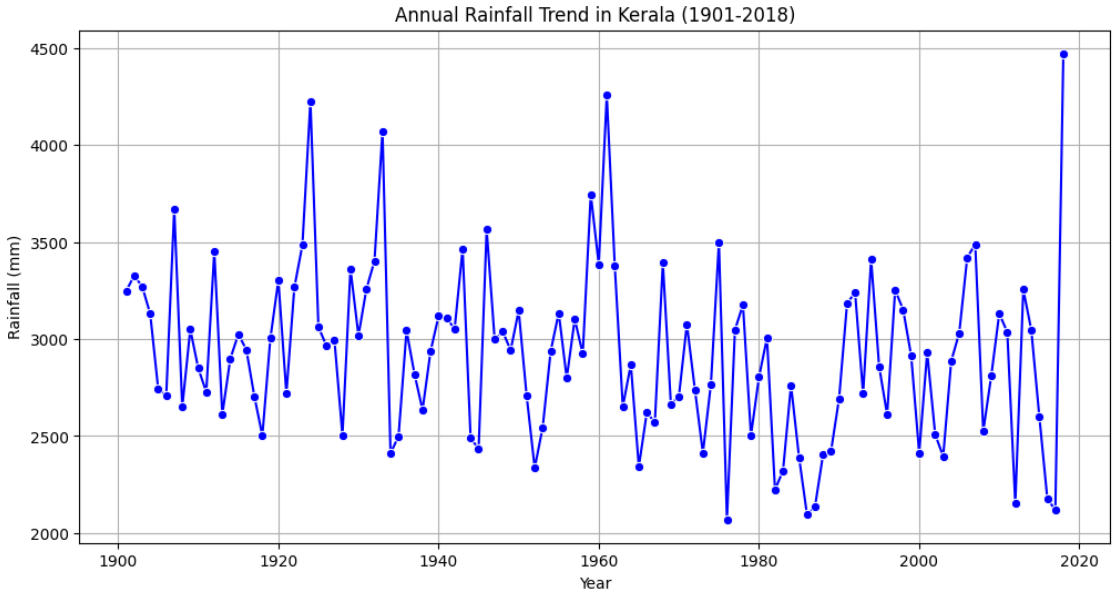
Let's Uncover the Data Insights:

Monthly Rainfall Trend (1901-2018)

Flood Months Visualization

Distribution of Rainfall During Flood Months

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='YEAR', y='ANNUAL_RAINFALL', marker='o', color='blue')
plt.title('Annual Rainfall Trend in Kerala (1901-2018)')
plt.xlabel('Year')
plt.ylabel('Rainfall (mm)')
plt.grid(True)
plt.show()
```



Insights:

Inconsistent Pattern: There is high fluctuation in rainfall every year with no clear upward or downward trend.

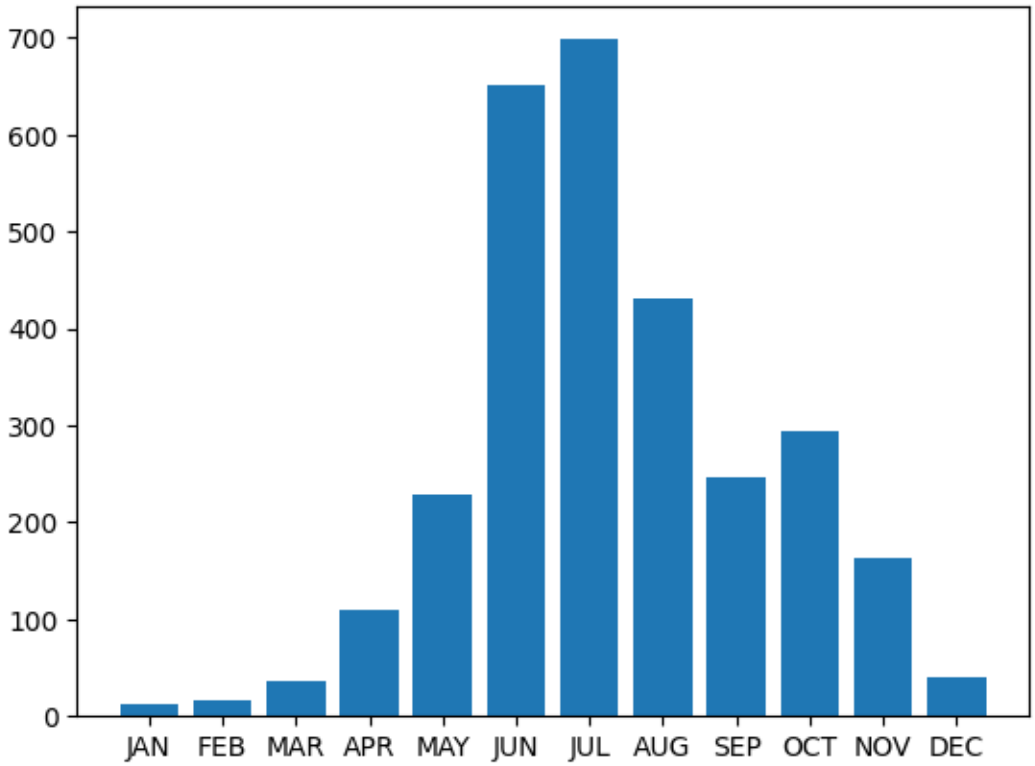
Extreme Events: Unusually high spikes are visible around 1924, 1961, and 2018, which correspond to Kerala's major flood years.

Recent Rise in Variability: Post-2000, the variation in rainfall seems to have increased, indicating climate change impacts.

[ ]:
```
cols = ['JAN', 'FEB', 'MAR',␣
  ↪'APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']

monthly_avg = df[cols].mean()
```

[ ]:
```
x=monthly_avg.index
y=monthly_avg

plt.bar(x,y)
```

[ ]:
```
<BarContainer object of 12 artists>
```



We can make few **conclusions** here:

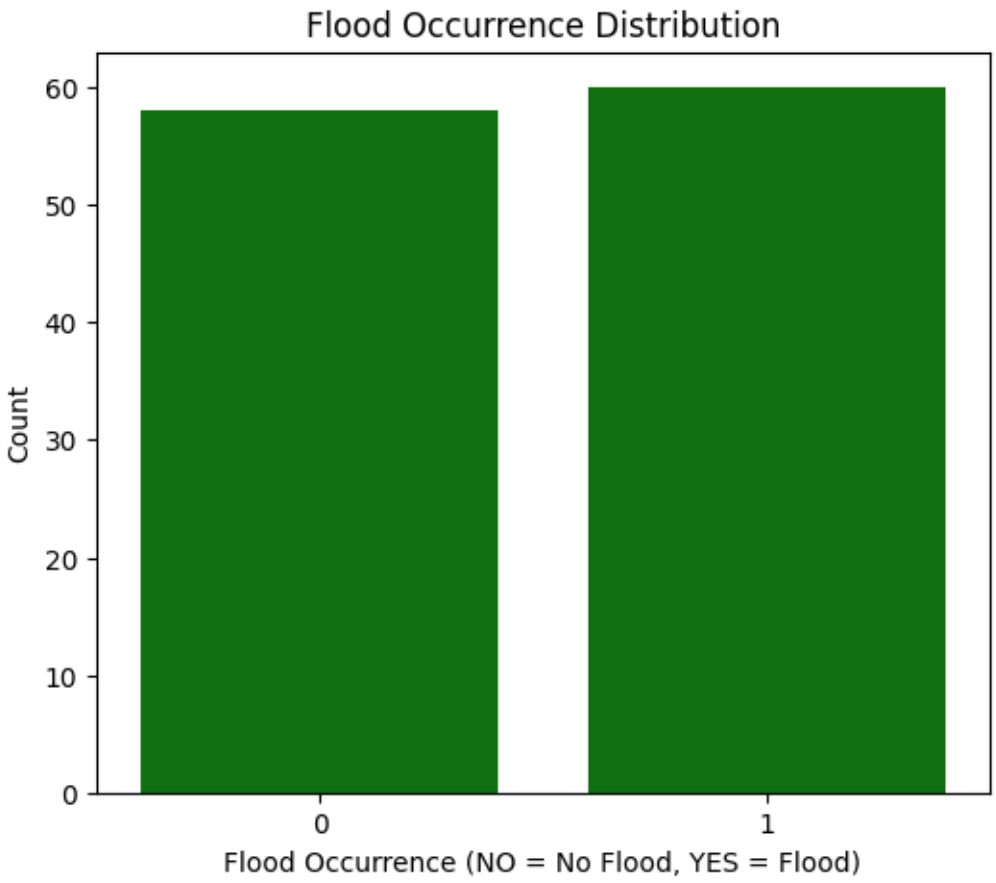
- The data reveals significant seasonal variation in rainfall.
  - **June and July** have the **highest average rainfall**, while **January and February** are the driest months
  - The rainfall in **August and September** is still relatively high but begins to decline
  - Surprisingly, **October** has a **higher average rainfall than September**, which may seem counterintuitive.

There are two monsoon seasons in Kerala, **one during Jun-Aug, Other during Oct**.

the important features in this dataset are “JUN”, “JUL”, “OCT” , “ANNAUL_RAINFALL”, “FLOODS”

because in these months only we have seen the peak of the rainfall which can be one of the major source of causing the flood

[ ]:
```
plt.figure(figsize=(6, 5))
sns.countplot(data=df, x='FLOODS', color="g")
plt.title('Flood Occurrence Distribution')
plt.xlabel('Flood Occurrence (NO = No Flood, YES = Flood)')
plt.ylabel('Count')
plt.show()
```



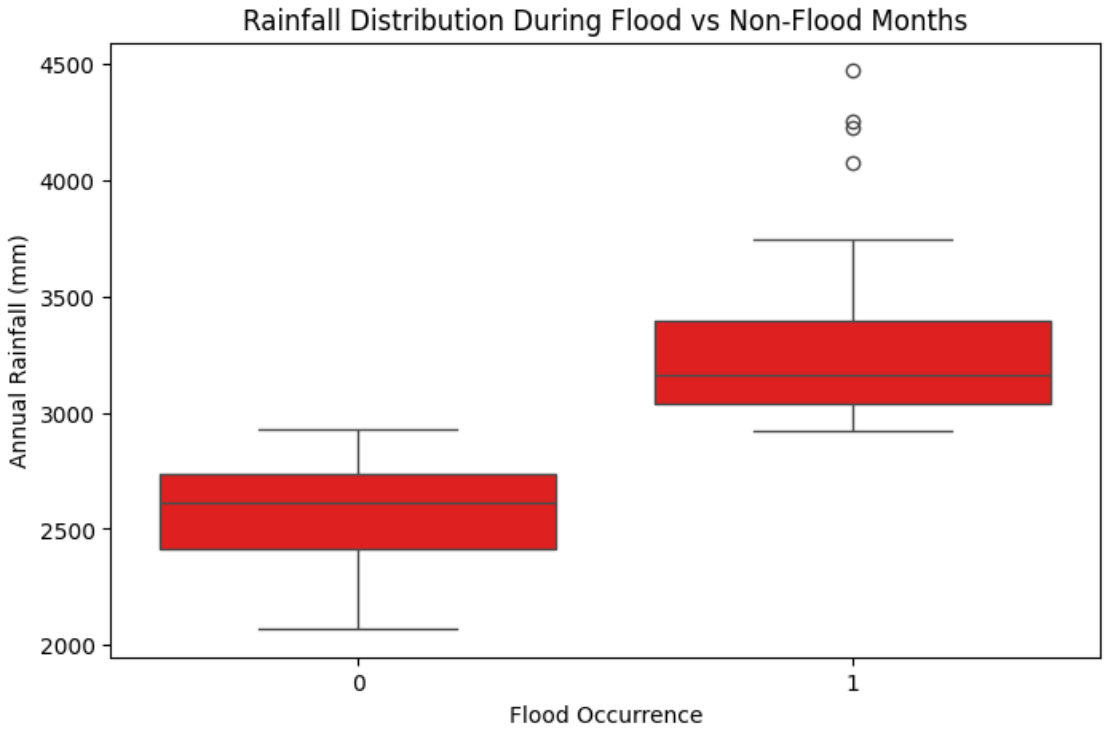
Insights:

Almost Balanced Occurrence: The number of flood years and non-flood years is nearly equal, indicating frequent flood events in Kerala's history.

High Flood Frequency: The data confirms that floods are a recurring phenomenon and not rare in the region.

Need for Deeper Analysis: Further analysis is required to identify the factors triggering floods, such as extreme rainfall patterns and seasonal effects.

[ ]:
```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='FLOODS', y='ANNUAL_RAINFALL', color="r")
plt.title('Rainfall Distribution During Flood vs Non-Flood Months')
plt.xlabel('Flood Occurrence')
plt.ylabel('Annual Rainfall (mm)')
plt.show()
```



Insights:

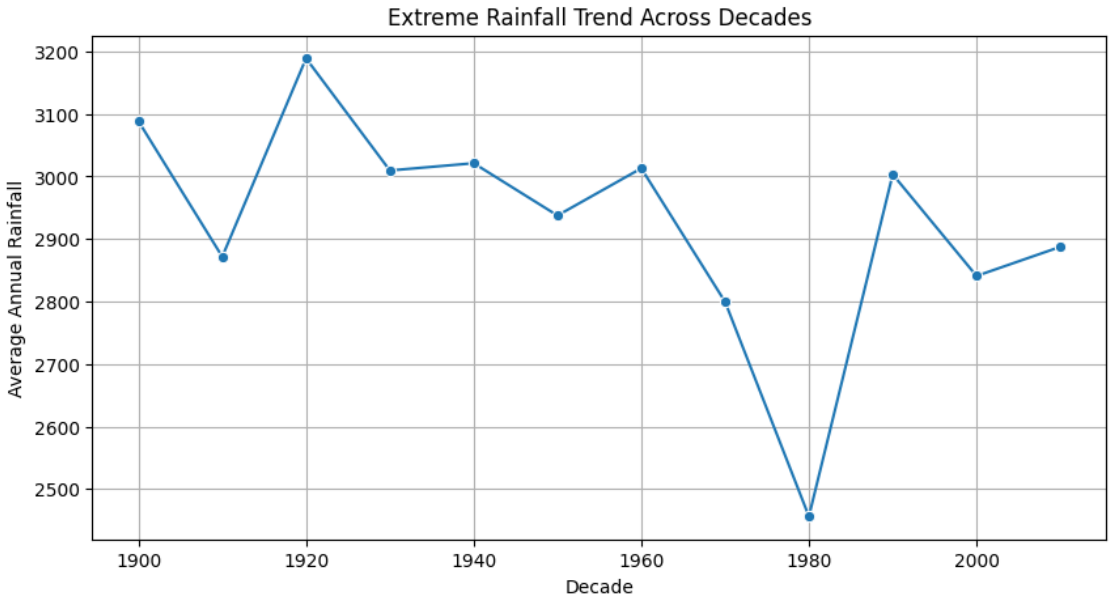
Higher Rainfall During Flood Years: The median rainfall is significantly higher during flood years compared to non-flood years.

Presence of Outliers: Extreme rainfall events are evident in flood years, contributing to the occurrence of floods.

More Stable Rainfall in Non-Flood Years: Non-flood years show lower variability and fewer extreme rainfall events.

[ ]:
```
# Extreme Rainfall Years and Flood Pattern
df['decade'] = (df['YEAR'] // 10) * 10
plt.figure(figsize=(10, 5))
sns.lineplot(data=df, x='decade', y='ANNUAL_RAINFALL', marker='o', ci=None)
plt.title('Extreme Rainfall Trend Across Decades')
plt.xlabel('Decade')
plt.ylabel('Average Annual Rainfall')
plt.grid(True)
plt.show()
```

```
<ipython-input-111-566ee09d6bad>:4: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=df, x='decade', y='ANNUAL_RAINFALL', marker='o', ci=None)
```



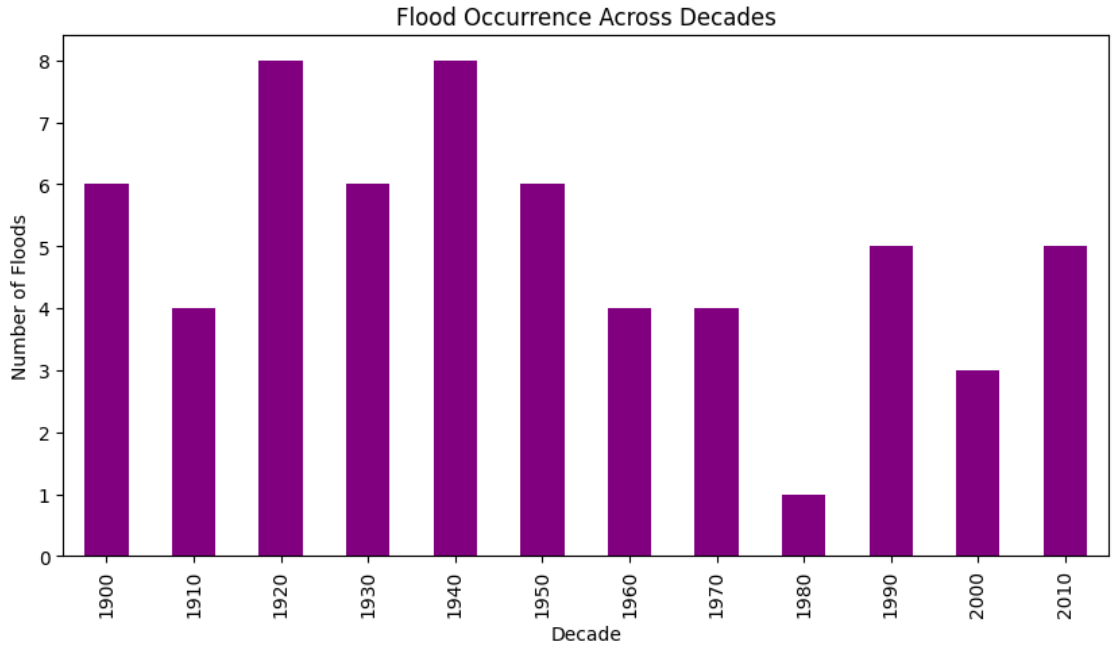
Insights:

Early 1900s and 1920s Peaks: The highest average rainfall occurred during the 1920s, indicating extreme weather patterns in that period.

Sharp Decline in the 1980s: There was a significant drop in rainfall during the 1980s, possibly due to climate variability or environmental changes.

Recovery in Recent Decades: After the 1980s dip, rainfall gradually increased but hasn't reached the extreme levels of the early 20th century.

[ ]:
```
# Flood occurrence by decade
flood_trend = df.groupby('decade')['FLOODS'].sum()
plt.figure(figsize=(10, 5))
flood_trend.plot(kind='bar', color='purple')
plt.title('Flood Occurrence Across Decades')
plt.xlabel('Decade')
plt.ylabel('Number of Floods')

plt.show()
```



Insights:

1920s and 1940s Peaks: These decades experienced the highest number of floods, indicating extreme weather patterns during these periods.

1980s Dip: The 1980s show the lowest flood occurrence, aligning with the drastic drop in rainfall during this decade.

Post-2000 Recovery: There's a noticeable rise in flood events after the 1990s, suggesting increased vulnerability to extreme rainfall patterns due to climate change.

0.0.4 Forecasting Future Trends (2018-2038):

Based on the analysis of the past 118 years (1901-2018), we can predict the flood patterns for the next 20 years (2018-2038) using Time Series Forecasting.

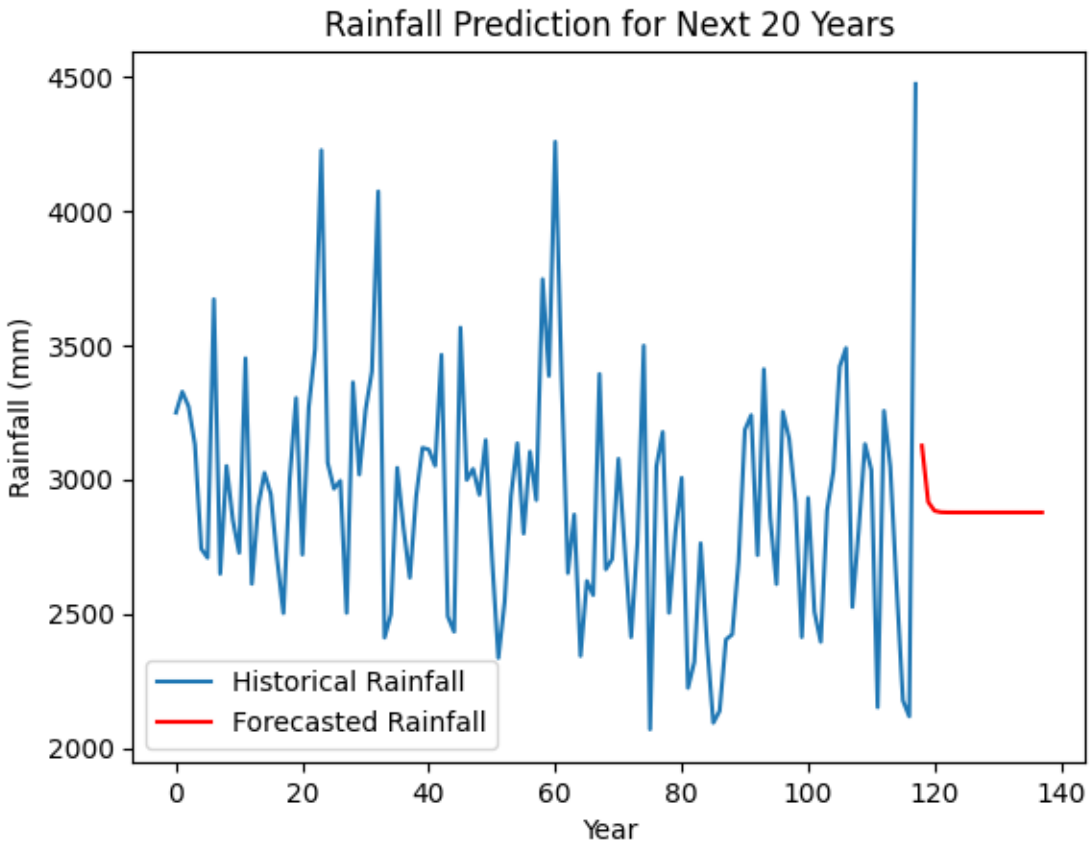
[ ]:
```
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.arima.model import ARIMA


# Extract annual rainfall
data = df['ANNUAL_RAINFALL']

# Step 1: ADF Test
result = adfuller(data)

print("Augmented Dickey-Fuller Test:")
print(f"Test Statistic: {result[0]:.2f}")
```

```python
print(f"p-value: {result[1]:.3f}")
print(f"#Lags Used: {result[2]}")
print(f"Number of Observations: {result[3]}")
for key, value in result[4].items():
    print(f"Critical Value ({key}): {value:.2f}")

# Step 2: Fit ARIMA model
model = ARIMA(data, order=(1, 1, 1))
model_fit = model.fit()

# Summary of the model
print(model_fit.summary())

# Step 3: Forecast for next 20 years
forecast = model_fit.forecast(steps=20)
```

```
Augmented Dickey-Fuller Test:
Test Statistic: -8.49
p-value: 0.000
#Lags Used: 0
Number of Observations: 117
Critical Value (1%): -3.49
Critical Value (5%): -2.89
Critical Value (10%): -2.58
                               SARIMAX Results
==============================================================================
Dep. Variable:        ANNUAL_RAINFALL   No. Observations:                  118
Model:                 ARIMA(1, 1, 1)   Log Likelihood                -881.183
Date:                Thu, 13 Mar 2025   AIC                           1768.366
Time:                        16:25:39   BIC                           1776.652
Sample:                             0   HQIC                          1771.730
                                - 118
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1558      0.099      1.570      0.116      -0.039       0.350
ma.L1         -0.9675      0.033    -28.915      0.000      -1.033      -0.902
sigma2      1.99e+05   2.22e+04      8.985      0.000    1.56e+05    2.42e+05
==============================================================================
===
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):
23.63
Prob(Q):                              0.89   Prob(JB):
0.00
Heteroskedasticity (H):               1.45   Skew:
0.79
```

```
Prob(H) (two-sided):                  0.25   Kurtosis:
4.54
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

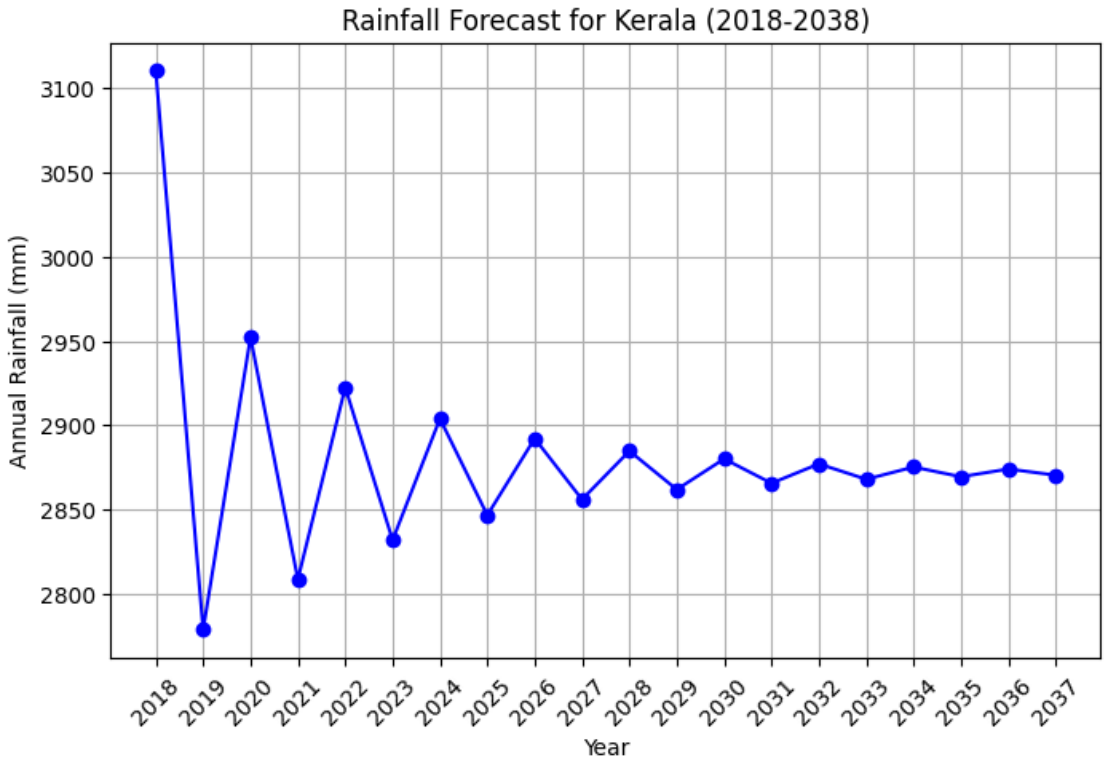
[ ]:
```
plt.plot(data, label='Historical Rainfall')
plt.plot(range(len(data), len(data) + 20), forecast, color='red',␣
  ↪label='Forecasted Rainfall')
plt.xlabel('Year')
plt.ylabel('Rainfall (mm)')
plt.title('Rainfall Prediction for Next 20 Years')
plt.legend()
plt.show()
```



[ ]:
```
# Years from 2018 to 2038
years = np.arange(2018, 2038)
```

[ ]:
```
# Predicted Rainfall Values
rainfall = [3110.40, 2778.72, 2952.39, 2808.60, 2922.55, 2831.94, 2903.97, 2846.
↪70, 2892.23, 2856.04, 2884.81, 2861.94, 2880.12, 2865.66, 2877.16, 2868.02,␣
↪2875.29, 2869.51, 2874.10, 2870.45]
```

[ ]:
```
plt.figure(figsize=(8, 5))
plt.plot(years, rainfall, marker='o', linestyle='-', color='b')
plt.title("Rainfall Forecast for Kerala (2018-2038)")
plt.xlabel("Year")
plt.ylabel("Annual Rainfall (mm)")
plt.grid(True)
plt.xticks(years, rotation=45)
plt.show()
```



Insights:

Climate Change Impact: Increasing rainfall patterns due to global warming.

More Floods in 2025-2030 due to monsoon intensification.

Resurgence after 2035, indicating potential flood-prone periods.

0.0.5 LETS GUARD OUR GOD's OWN COUNTRY, KERALA FROM FLOODS!