Rainfall Trends Analysis

```
[ ] import pandas as pd
    import plotly.graph_objects as go
    import plotly.express as px
    from scipy.stats import pearsonr
    from sklearn.ensemble import IsolationForest
    from sklearn.cluster import KMeans
    from sklearn.preprocessing import StandardScaler
    from prophet import Prophet
```

```
[ ] # import data
    rainfall_df = pd.read_csv('/content/drive/MyDrive/rainfall_area-wt_India_1901-2015.csv')
```

Data Preprocessing

```
[ ] rainfall_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 19 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   REGION    115 non-null    object
 1   YEAR      115 non-null    int64
 2   JAN       115 non-null    float64
 3   FEB       115 non-null    float64
 4   MAR       115 non-null    float64
 5   APR       115 non-null    float64
 6   MAY       115 non-null    float64
 7   JUN       115 non-null    float64
 8   JUL       115 non-null    float64
 9   AUG       115 non-null    float64
 10  SEP       115 non-null    float64
 11  OCT       115 non-null    float64
 12  NOV       115 non-null    float64
 13  DEC       115 non-null    float64
 14  ANNUAL    115 non-null    float64
 15  Jan-Feb   115 non-null    float64
 16  Mar-May   115 non-null    float64
 17  Jun-Sep   115 non-null    float64
 18  Oct-Dec   115 non-null    float64
dtypes: float64(17), int64(1), object(1)
memory usage: 17.2+ KB
```

```python
rainfall_df.describe()
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	1958.000000	19.759130	23.434783	28.254783	38.241739	62.193913	168.360000	291.022609	258.400870	172.473043	75.701739	29.205217	14.980000	1182.034783	43.189565	128.694783
std	33.341666	9.992628	11.512739	12.286408	10.353040	15.673378	35.569654	41.161390	34.975419	36.641234	28.268152	16.101056	8.788761	110.686214	14.476335	22.895134
min	1901.000000	2.700000	2.700000	7.200000	16.100000	32.100000	86.500000	138.900000	191.700000	96.900000	20.000000	3.600000	1.600000	920.800000	11.700000	84.500000
25%	1929.500000	13.000000	13.300000	19.750000	31.600000	51.600000	144.050000	267.350000	233.950000	144.850000	55.600000	17.300000	9.600000	1102.400000	33.800000	112.350000
50%	1958.000000	17.800000	22.500000	25.500000	37.400000	59.500000	165.600000	295.800000	259.300000	173.100000	69.200000	26.100000	14.100000	1190.500000	41.300000	125.100000
75%	1986.500000	24.850000	30.300000	34.400000	43.850000	71.200000	192.050000	318.650000	287.950000	198.300000	92.850000	39.650000	19.000000	1243.550000	51.400000	139.650000
max	2015.000000	58.500000	53.800000	63.300000	69.400000	114.500000	275.500000	383.400000	335.500000	281.000000	158.800000	74.200000	54.400000	1480.300000	86.300000	209.700000

```python
rainfall_df.head()
```

	REGION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	INDIA	1901	34.7	37.7	18.0	39.3	50.8	113.4	242.2	272.9	124.4	52.7	38.0	8.3	1032.3	72.4	108.1	752.8	99.0
1	INDIA	1902	7.4	4.3	19.0	43.5	48.3	108.8	284.0	199.7	201.5	61.5	27.9	24.4	1030.2	11.7	110.8	794.0	113.8
2	INDIA	1903	17.0	8.3	31.3	17.1	59.5	118.3	297.0	270.4	199.1	117.9	36.9	17.7	1190.5	25.3	107.9	884.8	172.5
3	INDIA	1904	14.4	9.6	31.8	33.1	72.4	164.8	261.0	206.4	129.6	69.0	11.2	16.3	1019.8	24.0	137.4	761.8	96.6
4	INDIA	1905	25.3	20.9	42.7	33.7	55.7	93.3	252.8	200.8	178.4	51.4	9.7	10.5	975.3	46.2	132.2	725.4	71.6

```python
# Check for nulls
rainfall_df.isna().sum()
```

	0
REGION	0
YEAR	0
JAN	0
FEB	0
MAR	0
APR	0

```python
# Check for duplicate values
rainfall_df.duplicated().sum()
```

0

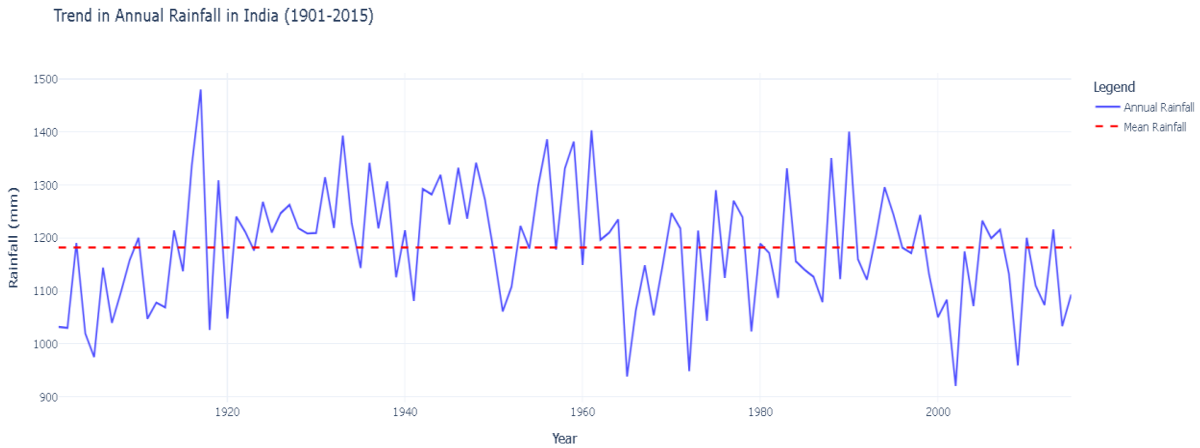
Analyzing Annual Rainfall Trends Over Time To understand the broader trends, we will plot three plots:

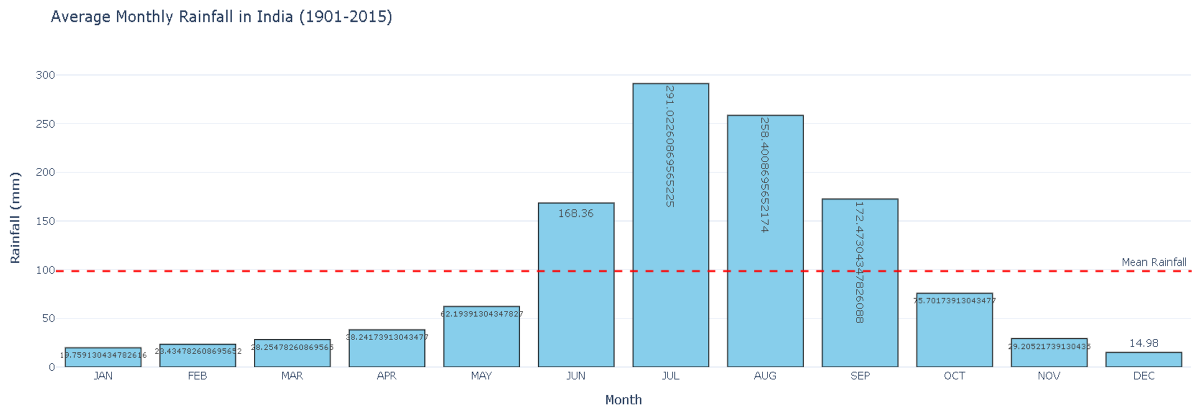
1. We will plot annual rainfall over time and compare it with the overall mean. This helps identify whether there is a noticeable long-term trend, such as an increase or decrease in rainfall over the years.
2. We will calculate the average rainfall for each month across all years. And by plotting a bar chart, we can identify the months with the highest and lowest average rainfall.
3. We will analyze seasonal rainfall by aggregating rainfall data into four seasons: Jan-Feb, Mar-May, Jun-Sep (Monsoon), and Oct-Dec.

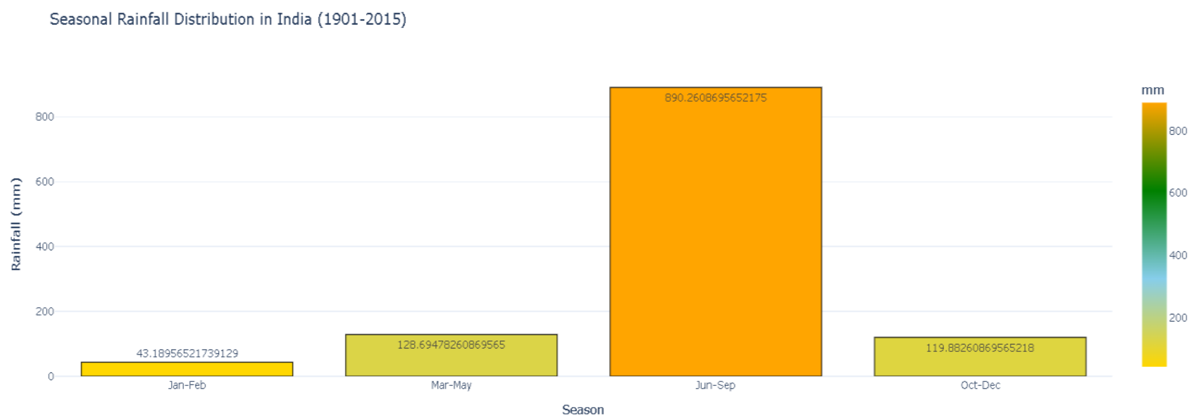
```
# analyze trends in annual rainfall over time
annual_rainfall = rainfall_df[['YEAR', 'ANNUAL']]

fig_annual = go.Figure()
fig_annual.add_trace(go.Scatter(
    x=annual_rainfall['YEAR'],
    y=annual_rainfall['ANNUAL'],
    mode='lines',
    name='Annual Rainfall',
    line=dict(color='blue', width=2),
    opacity=0.7
))
fig_annual.add_trace(go.Scatter(
    x=annual_rainfall['YEAR'],
    y=[annual_rainfall['ANNUAL'].mean()] * len(annual_rainfall),
    mode='lines',
    name='Mean Rainfall',
    line=dict(color='red', dash='dash')
))
fig_annual.update_layout(
    title='Trend in Annual Rainfall in India (1901-2015)',
    xaxis_title='Year',
    yaxis_title='Rainfall (mm)',
    template='plotly_white',
    legend=dict(title="Legend"),
    height=500
)
fig_annual.show()

# identify months with the highest and lowest rainfall on average
monthly_columns = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
```

```
monthly_avg = rainfall_df[monthly_columns].mean()
highest_rainfall_month = monthly_avg.idxmax()
lowest_rainfall_month = monthly_avg.idxmin()

fig_monthly = px.bar(
    x=monthly_avg.index,
    y=monthly_avg.values,
    labels={'x': 'Month', 'y': 'Rainfall (mm)'},
    title='Average Monthly Rainfall in India (1901-2015)',
    text=monthly_avg.values
)
fig_monthly.add_hline(
    y=monthly_avg.mean(),
    line_dash="dash",
    line_color="red",
    annotation_text="Mean Rainfall",
    annotation_position="top right"
)
fig_monthly.update_traces(marker_color='skyblue', marker_line_color='black',
marker_line_width=1)
fig_monthly.update_layout(template='plotly_white', height=500)
fig_monthly.show()

# seasonal rainfall distribution
seasonal_columns = ['Jan-Feb', 'Mar-May', 'Jun-Sep', 'Oct-Dec']
seasonal_avg = rainfall_df[seasonal_columns].mean()

fig_seasonal = px.bar(
    x=seasonal_avg.index,
    y=seasonal_avg.values,
    labels={'x': 'Season', 'y': 'Rainfall (mm)'},
```

```
    title='Seasonal Rainfall Distribution in India (1901-2015)',
    text=seasonal_avg.values,
    color=seasonal_avg.values,
    color_continuous_scale=['gold', 'skyblue', 'green', 'orange']
)

fig_seasonal.update_traces(marker_line_color='black', marker_line_width=1)
fig_seasonal.update_layout(
    template='plotly_white',
    height=500,
    coloraxis_colorbar=dict(title='mm')
)
fig_seasonal.update_layout(template='plotly_white', height=500)
fig_seasonal.show()
```





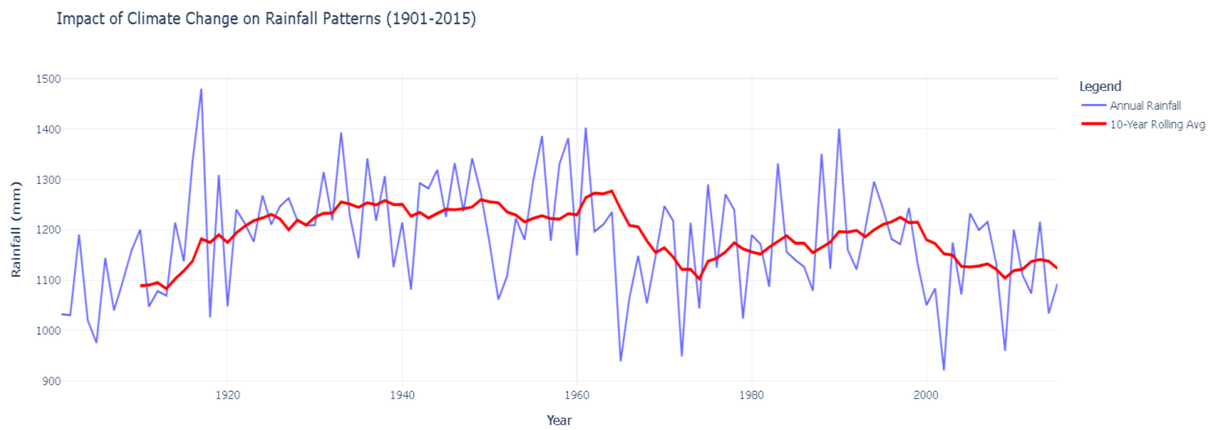


The seasonal distribution highlights the dominance of the monsoon season (June to September), which contributes the bulk of annual rainfall (around 890 mm). In contrast, the other seasons (January-February, March-May, and October-December) contribute significantly less to the annual total, which emphasizes the critical role of the monsoon.

Assessing the Impact of Climate Change in the Rainfall Trends in India Now, we will calculate a 10-year rolling average of annual rainfall to identify long-term trends and smooth out short-term variations. This will help assess the potential impact of climate change on rainfall patterns.

```
# calculating rolling averages to assess climate change impact
rainfall_df['10-Year Rolling Avg'] = rainfall_df['ANNUAL'].rolling(window=10).mean()

fig_climate_change = go.Figure()

fig_climate_change.add_trace(go.Scatter(
    x=rainfall_df['YEAR'],
    y=rainfall_df['ANNUAL'],
    mode='lines',
    name='Annual Rainfall',
    line=dict(color='blue', width=2),
    opacity=0.6
))

fig_climate_change.add_trace(go.Scatter(
    x=rainfall_df['YEAR'],
    y=rainfall_df['10-Year Rolling Avg'],
```

```
    mode='lines',
    name='10-Year Rolling Avg',
    line=dict(color='red', width=3)
))

fig_climate_change.update_layout(
    title='Impact of Climate Change on Rainfall Patterns (1901-2015)',
    xaxis_title='Year',
    yaxis_title='Rainfall (mm)',
    template='plotly_white',
    legend=dict(title="Legend"),
    height=500
)

fig_climate_change.show()
```



This graph shows the annual rainfall trends in India (blue line) and a 10-year rolling average (red line) to identify long-term patterns. While annual rainfall exhibits significant variability, the 10-year rolling average indicates a slight downward trend post-1960, which suggests a possible impact of climate change on rainfall distribution. Periods of higher averages in the early 20th-century contrast with more consistent but lower averages in recent decades.

Now, using statistical thresholds (1.5 standard deviations below or above the mean), let's identify years with extreme or deficient rainfall. This will help detect drought years and periods of excessive rainfall.

```
[ ] # identifying drought and extreme rainfall years
    mean_rainfall = rainfall_df['ANNUAL'].mean()
    std_dev_rainfall = rainfall_df['ANNUAL'].std()

    drought_years = rainfall_df[rainfall_df['ANNUAL'] < (mean_rainfall - 1.5 * std_dev_rainfall)]
    extreme_rainfall_years = rainfall_df[rainfall_df['ANNUAL'] > (mean_rainfall + 1.5 * std_dev_rainfall)]

    # correlating seasonal rainfall with annual rainfall totals
    seasonal_columns = ['Jan-Feb', 'Mar-May', 'Jun-Sep', 'Oct-Dec']
    seasonal_correlations = {
        season: pearsonr(rainfall_df[season], rainfall_df['ANNUAL'])[0] for season in seasonal_columns
    }

    # displaying results for drought/extreme years and correlations
    drought_years_summary = drought_years[['YEAR', 'ANNUAL']].reset_index(drop=True)
    extreme_rainfall_years_summary = extreme_rainfall_years[['YEAR', 'ANNUAL']].reset_index(drop=True)
    seasonal_correlations_summary = pd.DataFrame.from_dict(seasonal_correlations, orient='index', columns=['Correlation'])

    drought_years_summary, extreme_rainfall_years_summary, seasonal_correlations_summary
```

```
(   YEAR  ANNUAL
 0  1905   975.3
 1  1965   938.4
 2  1972   948.5
 3  2002   920.8
 4  2009   959.3,
    YEAR  ANNUAL
 0  1917  1480.3
 1  1933  1393.5
 2  1956  1386.2
 3  1959  1382.1
 4  1961  1403.0
 5  1988  1351.0
 6  1990  1400.6,
          Correlation
 Jan-Feb     0.228913
 Mar-May     0.313057
 Jun-Sep     0.930027
 Oct-Dec     0.531648)
```

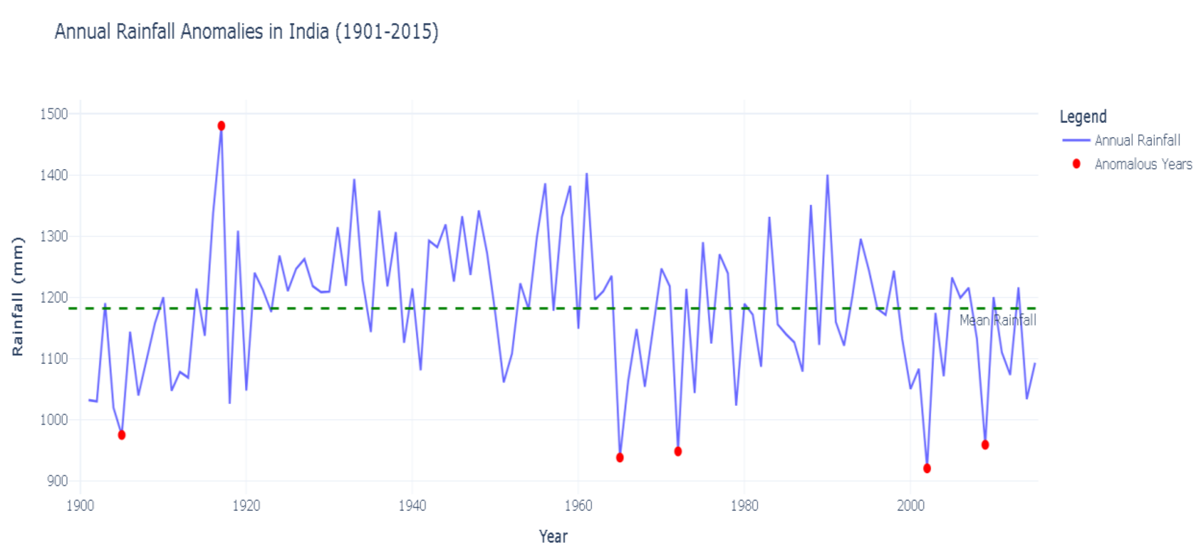
The analysis identifies five significant drought years (e.g., 2002 and 2009) and seven extreme rainfall years (e.g., 1917 and 1990) based on deviations from the mean annual rainfall. Seasonal rainfall correlations with annual totals reveal that the monsoon season (June-Septembe has the strongest correlation (0.93), which indicates it predominantly drives annual rainfall patterns. In contrast, other seasons like January-February (0.23) and March-May (0.31) have weaker correlations, which emphasizes the critical role of the monsoon in India's overall rainfall dynamics.

Detecting Anomalies in the Rainfall Trends in India

Now, using an Isolation Forest algorithm, we will identify anomalies in both annual and monthly rainfall. This will highlight specific years or months with unusual rainfall patterns. Let's start with identifying anomalies in annual rainfall.

```
# detect anomalous rainfall years based on annual data
isolation_forest = IsolationForest(contamination=0.05, random_state=42)
rainfall_df['Annual_Anomaly'] = isolation_forest.fit_predict(rainfall_df[['ANNUAL']])

# identify anomalies in annual rainfall
annual_anomalies = rainfall_df[rainfall_df['Annual_Anomaly'] == -1]

# detect anomalous months based on monthly data
monthly_data = rainfall_df[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV',
'DEC']]
monthly_anomalies = isolation_forest.fit_predict(monthly_data)

# add anomaly detection results for months
rainfall_df['Monthly_Anomaly'] = monthly_anomalies
monthly_anomalies_df = rainfall_df[rainfall_df['Monthly_Anomaly'] == -1][['YEAR'] +
monthly_columns]

fig_annual_anomalies = go.Figure()

fig_annual_anomalies.add_trace(go.Scatter(
    x=rainfall_df['YEAR'],
    y=rainfall_df['ANNUAL'],
    mode='lines',
    name='Annual Rainfall',
    line=dict(color='blue', width=2),
    opacity=0.6
))

fig_annual_anomalies.add_trace(go.Scatter(
    x=annual_anomalies['YEAR'],
    y=annual_anomalies['ANNUAL'],
```

```
    mode='markers',
    name='Anomalous Years',
    marker=dict(color='red', size=8, symbol='circle')
))

fig_annual_anomalies.add_hline(
    y=rainfall_df['ANNUAL'].mean(),
    line_dash='dash',
    line_color='green',
    annotation_text='Mean Rainfall',
    annotation_position='bottom right'
)
fig_annual_anomalies.update_layout(
    title='Annual Rainfall Anomalies in India (1901-2015)',
    xaxis_title='Year',
    yaxis_title='Rainfall (mm)',
    template='plotly_white',
    legend=dict(title="Legend"),
    height=500
)
fig_annual_anomalies.show()
```

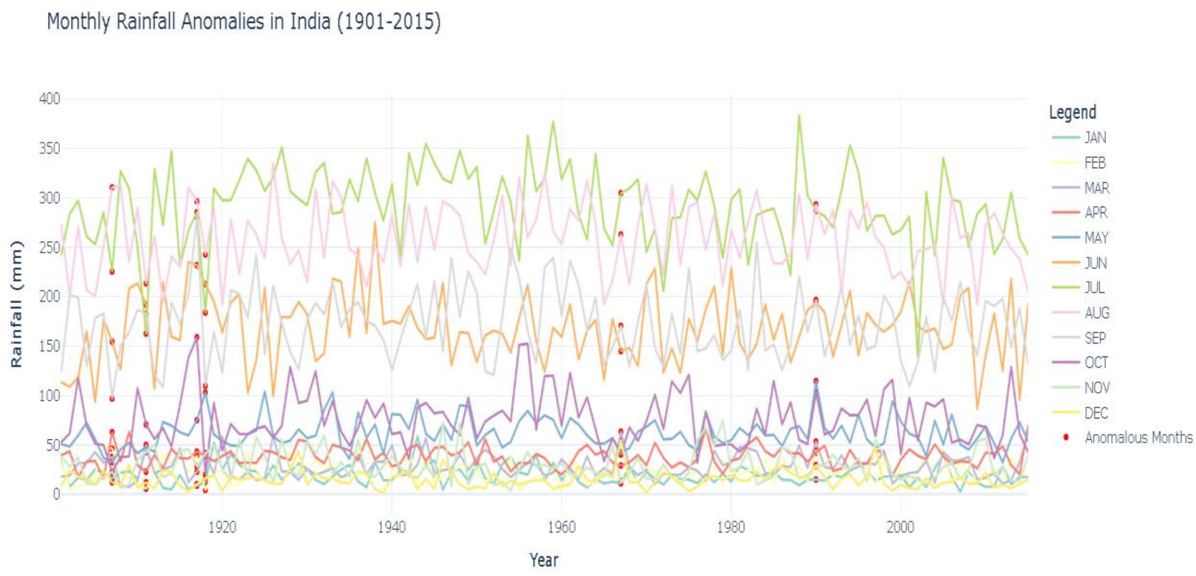


This graph highlights years with significant rainfall anomalies, where annual rainfall deviated substantially from the mean. Drought years (e.g., 1905, 1965, 2002) and extreme rainfall years (e.g., 1917, 1961) are marked as red points, which showcase outliers in rainfall patterns. While most years cluster around the mean (green dashed line), the anomalies emphasize the variability in India's rainfall, driven by factors like monsoonal fluctuations and climate events. This underscores the need for monitoring and preparedness for extreme weather events.

Now, let's identify anomalies in monthly rainfall.

```
# preparing data for monthly anomalies
monthly_anomalies = []
for column in monthly_columns:
    for _, row in monthly_anomalies_df.iterrows():
        monthly_anomalies.append({'Year': row['YEAR'], 'Month': column, 'Rainfall': row[column]})

monthly_anomalies_df_long = pd.DataFrame(monthly_anomalies)

fig_monthly_anomalies = px.line(
    rainfall_df,
    x='YEAR',
    y=monthly_columns,
    labels={'YEAR': 'Year', 'value': 'Rainfall (mm)', 'variable': 'Month'},
    title='Monthly Rainfall Anomalies in India (1901-2015)',
    color_discrete_sequence=px.colors.qualitative.Set3
)

fig_monthly_anomalies.add_trace(go.Scatter(
    x=monthly_anomalies_df_long['Year'],
    y=monthly_anomalies_df_long['Rainfall'],
    mode='markers',
    name='Anomalous Months',
    marker=dict(color='red', size=5, symbol='circle')
))
```

```
fig_monthly_anomalies.update_layout(
    template='plotly_white',
    legend=dict(title="Legend"),
    height=500
)
fig_monthly_anomalies.show()
```

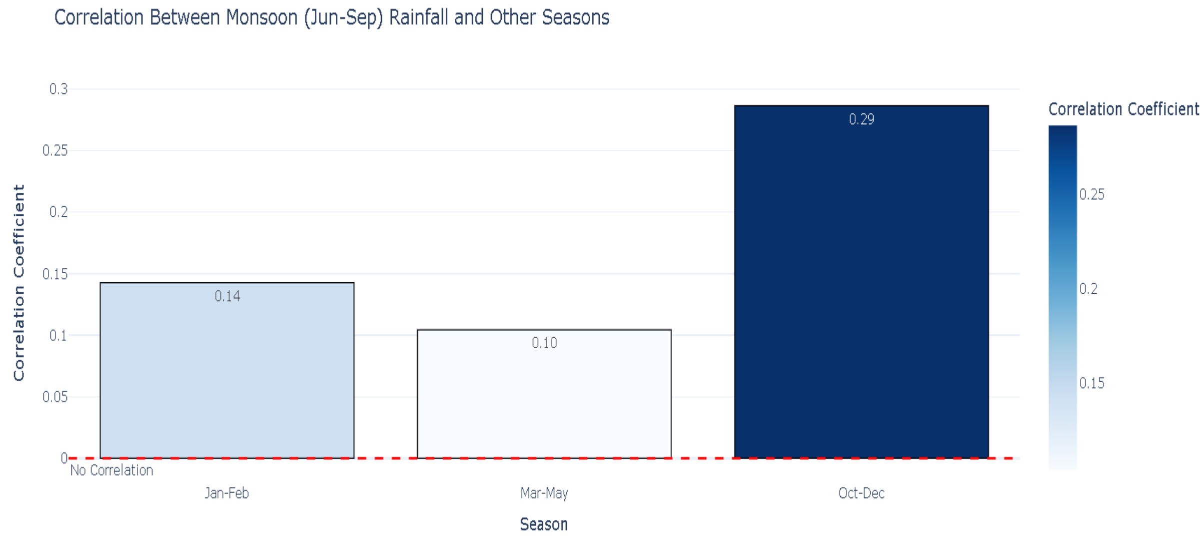


The variability is most pronounced during the monsoon months (June to September), which reflects the critical role of these months in India's rainfall dynamics. Anomalies in non-monsoon months, while less frequent, highlight periods of unusual weather patterns, potentially linked to climate variability or regional disturbances. This graph underscores the uneven distribution and high dependence on monsoonal rainfall for India's water resources.

Correlating Seasonal Rainfall with Annual Totals Now, we will calculate the correlation coefficients between seasonal rainfall and annual rainfall totals to understand how much each season contributes to the overall yearly rainfall.

```
# correlation analysis between monsoon (Jun-Sep) rainfall and other seasons
seasonal_columns = ['Jan-Feb', 'Mar-May', 'Jun-Sep', 'Oct-Dec']
monsoon_column = 'Jun-Sep'
relationships = {}

for season in seasonal_columns:
    if season != monsoon_column:
        corr, _ = pearsonr(rainfall_df[monsoon_column], rainfall_df[season])
        relationships[season] = corr
```

```
correlation_data = pd.DataFrame({
   'Season': list(relationships.keys()),
   'Correlation Coefficient': list(relationships.values())
})

fig = px.bar(
   correlation_data,
   x='Season',
   y='Correlation Coefficient',
   title='Correlation Between Monsoon (Jun-Sep) Rainfall and Other Seasons',
   labels={'Season': 'Season', 'Correlation Coefficient': 'Correlation Coefficient'},
   text='Correlation Coefficient',
   color='Correlation Coefficient',
   color_continuous_scale='Blues'
)

fig.add_hline(
   y=0,
   line_dash="dash",
   line_color="red",
   annotation_text="No Correlation",
   annotation_position="bottom left"
)
fig.update_traces(marker_line_color='black', marker_line_width=1, texttemplate='%{text:.2f}')
fig.update_layout(
   template='plotly_white',
   height=500
)
fig.show()
```

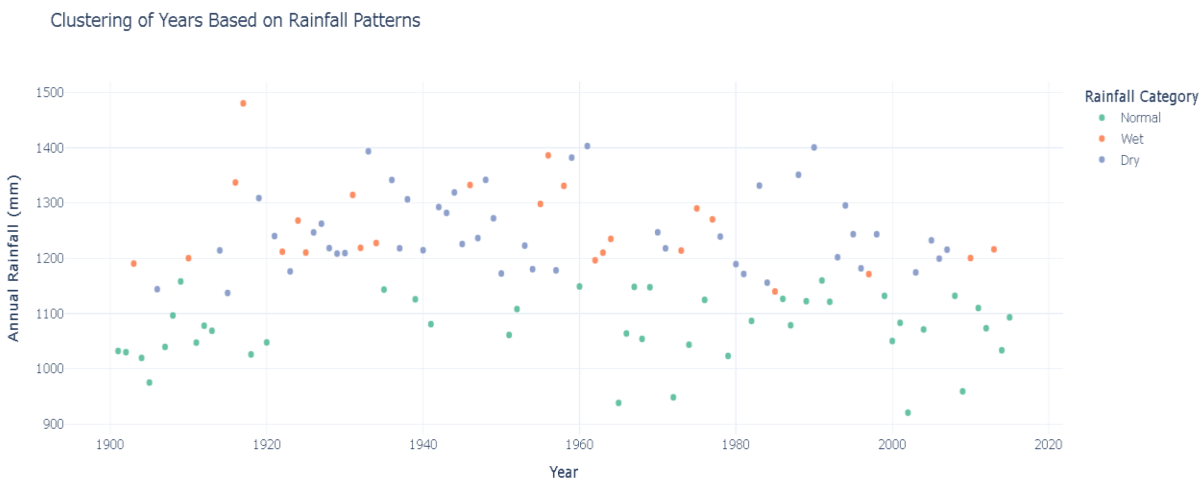


This graph shows the correlation between monsoon rainfall and rainfall during other seasons. The October-December season has the highest correlation (0.29), which suggests a moderate relationship, possibly due to the post-monsoon retreat rains. The January-February (0.14) and March-May (0.10) seasons exhibit weaker correlations, which indicate minimal dependence on monsoon rainfall. This highlights the dominance of monsoonal patterns as an independent driver of India's annual rainfall, with limited spillover effects on other seasons.

Grouping Years Based on Rainfall Patterns Now, by applying k-means clustering, we will group years into three categories: Dry, Normal, and Wet, based on rainfall patterns.

```
# prepare data for clustering
rainfall_features = rainfall_df[['Jan-Feb', 'Mar-May', 'Jun-Sep', 'Oct-Dec', 'ANNUAL']]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(rainfall_features)

# perform k-means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
rainfall_df['Rainfall_Cluster'] = kmeans.fit_predict(scaled_features)

# map cluster labels to categories (e.g., Dry, Normal, Wet)
cluster_labels = {0: 'Dry', 1: 'Normal', 2: 'Wet'}
rainfall_df['Rainfall_Category'] = rainfall_df['Rainfall_Cluster'].map(cluster_labels)

fig = px.scatter(
   rainfall_df,
   x='YEAR',
```

```
    y='ANNUAL',
    color='Rainfall_Category',
    title='Clustering of Years Based on Rainfall Patterns',
    labels={'YEAR': 'Year', 'ANNUAL': 'Annual Rainfall (mm)', 'Rainfall_Category': 'Rainfall Category'},
    color_discrete_sequence=px.colors.qualitative.Set2,
    hover_data={'Rainfall_Cluster': True, 'Rainfall_Category': True}
)
fig.update_layout(
    template='plotly_white',
    legend_title='Rainfall Category',
    height=500
)
fig.show()
```

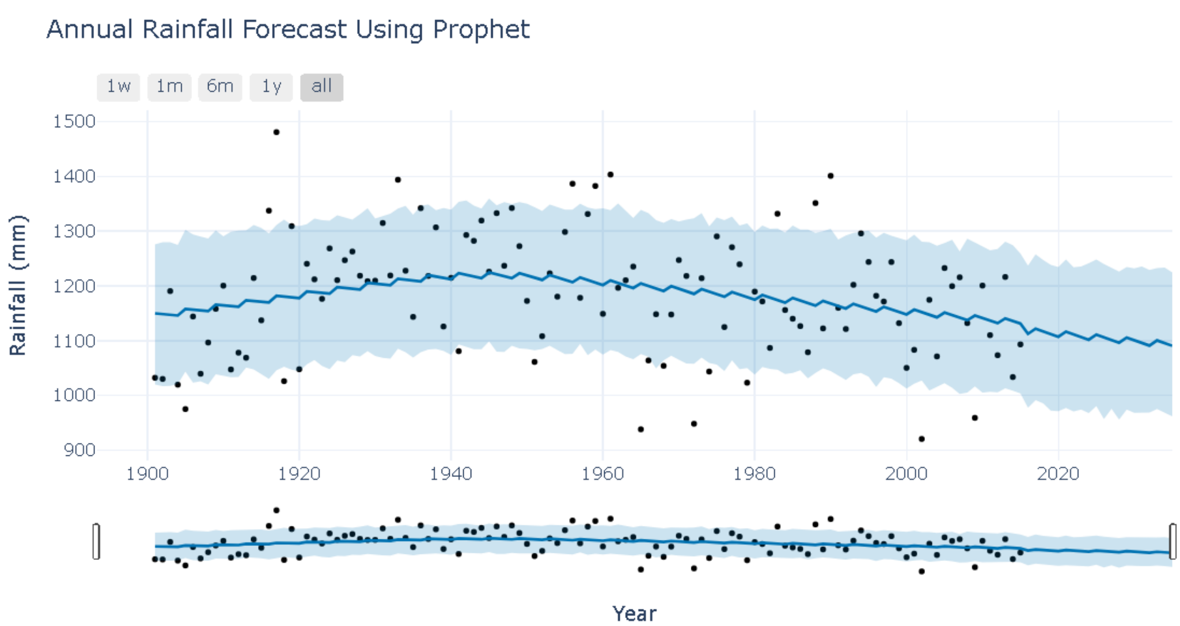


The clusters reveal that most years fall into the Normal category, while Wet years (above-normal rainfall) are sporadically distributed throughout the timeline, with a concentration in the early and mid-20th century. Dry years (below-normal rainfall) are more frequent in the latter half of the timeline, which indicates a potential shift in rainfall patterns over time. This clustering emphasizes the variability and potential long-term changes in India's rainfall dynamics.

Forecasting Future Rainfall Finally, we will use the Prophet library to forecast annual rainfall for the next 20 years.

```
rainfall_df['DATE'] = pd.to_datetime(rainfall_df['YEAR'], format='%Y')
annual_rainfall_ts = rainfall_df.set_index('DATE')['ANNUAL']

# Prepare the data for Prophet
prophet_data = annual_rainfall_ts.reset_index()
```

```
prophet_data.columns = ['ds', 'y']

from prophet.plot import plot_plotly, plot_components_plotly

prophet_model = Prophet()
prophet_model.fit(prophet_data)

# create a future dataframe for the next 20 years
future = prophet_model.make_future_dataframe(periods=20, freq='YE')
forecast = prophet_model.predict(future)
fig_forecast = plot_plotly(prophet_model, forecast)
fig_forecast.update_layout(
    title='Annual Rainfall Forecast Using Prophet',
    xaxis_title='Year',
    yaxis_title='Rainfall (mm)',
    template='plotly_white',
    height=500
)
fig_forecast.show()
```



The blue line represents the model's forecast trend, while the shaded area indicates the confidence interval. The trend reveals a slight decline in annual rainfall over time, with notable year-to-year variability (black dots representing actual data points). The model captures the variability well but highlights that future rainfall may continue to slightly decrease, which emphasizes the need for adaptive strategies to manage potential water resource challenges.

Conclusion

The analysis of India's rainfall trends and patterns from 1901 to 2015 reveals significant variability in annual and seasonal rainfall, with the monsoon season (June-September) being the dominant contributor. Anomalous years of extreme drought and wetness highlight the unpredictability of rainfall, while clustering shows a shift towards more dry years in recent decades. Correlations indicate the limited dependency of non-monsoon seasons on monsoon rainfall. A time-series forecast using Prophet suggests a slight declining trend in annual rainfall, which emphasises the need for long-term water resource planning and adaptation to changing climate patterns.