In [1]:
```
pip install prophet
```

```
Collecting prophet
  Downloading prophet-1.1.6-py3-none-win_amd64.whl.metadata (3.6 kB)
Collecting cmdstanpy>=1.0.4 (from prophet)
  Downloading cmdstanpy-1.2.5-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: numpy>=1.15.4 in c:\users\sharon\anaconda3\lib\site-
packages (from prophet) (1.26.4)
Requirement already satisfied: matplotlib>=2.0.0 in c:\users\sharon\anaconda3\lib\s
ite-packages (from prophet) (3.8.0)
Requirement already satisfied: pandas>=1.0.4 in c:\users\sharon\anaconda3\lib\site-
packages (from prophet) (2.1.4)
Collecting holidays<1,>=0.25 (from prophet)
  Downloading holidays-0.63-py3-none-any.whl.metadata (26 kB)
Requirement already satisfied: tqdm>=4.36.1 in c:\users\sharon\anaconda3\lib\site-p
ackages (from prophet) (4.65.0)
Collecting importlib-resources (from prophet)
  Downloading importlib_resources-6.4.5-py3-none-any.whl.metadata (4.0 kB)
Collecting stanio<2.0.0,>=0.4.0 (from cmdstanpy>=1.0.4->prophet)
  Downloading stanio-0.5.1-py3-none-any.whl.metadata (1.6 kB)
Requirement already satisfied: python-dateutil in c:\users\sharon\anaconda3\lib\sit
e-packages (from holidays<1,>=0.25->prophet) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sharon\anaconda3\lib\si
te-packages (from matplotlib>=2.0.0->prophet) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\sharon\anaconda3\lib\site-p
ackages (from matplotlib>=2.0.0->prophet) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sharon\anaconda3\lib\s
ite-packages (from matplotlib>=2.0.0->prophet) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sharon\anaconda3\lib\s
ite-packages (from matplotlib>=2.0.0->prophet) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\sharon\anaconda3\lib\sit
e-packages (from matplotlib>=2.0.0->prophet) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\sharon\anaconda3\lib\site-
packages (from matplotlib>=2.0.0->prophet) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sharon\anaconda3\lib\si
te-packages (from matplotlib>=2.0.0->prophet) (3.0.9)
Requirement already satisfied: pytz>=2020.1 in c:\users\sharon\anaconda3\lib\site-p
ackages (from pandas>=1.0.4->prophet) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\sharon\anaconda3\lib\site
-packages (from pandas>=1.0.4->prophet) (2023.3)
Requirement already satisfied: colorama in c:\users\sharon\anaconda3\lib\site-packa
ges (from tqdm>=4.36.1->prophet) (0.4.6)
Requirement already satisfied: six>=1.5 in c:\users\sharon\anaconda3\lib\site-packa
ges (from python-dateutil->holidays<1,>=0.25->prophet) (1.16.0)
Downloading prophet-1.1.6-py3-none-win_amd64.whl (13.3 MB)
   ---------------------------------------- 0.0/13.3 MB ? eta -:--:--
   - -------------------------------------- 0.4/13.3 MB 11.2 MB/s eta 0:00:02
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   --- ------------------------------------ 1.0/13.3 MB 16.4 MB/s eta 0:00:01
   ------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
   ------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
   ------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
```

```
------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
------ --------------------------------- 2.1/13.3 MB 3.0 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
--------- ------------------------------ 3.1/13.3 MB 3.1 MB/s eta 0:00:04
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
------------ --------------------------- 4.2/13.3 MB 3.1 MB/s eta 0:00:03
--------------- ------------------------ 5.2/13.3 MB 3.0 MB/s eta 0:00:03
--------------- ------------------------ 5.2/13.3 MB 3.0 MB/s eta 0:00:03
--------------- ------------------------ 5.2/13.3 MB 3.0 MB/s eta 0:00:03
--------------- ------------------------ 5.2/13.3 MB 3.0 MB/s eta 0:00:03
--------------- ------------------------ 5.2/13.3 MB 3.0 MB/s eta 0:00:03
--------------- ------------------------ 5.2/13.3 MB 3.0 MB/s eta 0:00:03
------------------ --------------------- 6.3/13.3 MB 3.2 MB/s eta 0:00:03
------------------ --------------------- 6.3/13.3 MB 3.2 MB/s eta 0:00:03
------------------ --------------------- 6.3/13.3 MB 3.2 MB/s eta 0:00:03
------------------ --------------------- 6.3/13.3 MB 3.2 MB/s eta 0:00:03
------------------ --------------------- 6.3/13.3 MB 3.2 MB/s eta 0:00:03
------------------ --------------------- 6.3/13.3 MB 3.2 MB/s eta 0:00:03
---------------------- ----------------- 7.3/13.3 MB 3.2 MB/s eta 0:00:02
---------------------- ----------------- 7.3/13.3 MB 3.2 MB/s eta 0:00:02
---------------------- ----------------- 7.3/13.3 MB 3.2 MB/s eta 0:00:02
---------------------- ----------------- 7.3/13.3 MB 3.2 MB/s eta 0:00:02
---------------------- ----------------- 7.3/13.3 MB 3.2 MB/s eta 0:00:02
---------------------- ----------------- 7.3/13.3 MB 3.2 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
------------------------- -------------- 8.4/13.3 MB 3.3 MB/s eta 0:00:02
---------------------------- ----------- 9.4/13.3 MB 3.0 MB/s eta 0:00:02
---------------------------- ----------- 9.4/13.3 MB 3.0 MB/s eta 0:00:02
---------------------------- ----------- 9.4/13.3 MB 3.0 MB/s eta 0:00:02
---------------------------- ----------- 9.4/13.3 MB 3.0 MB/s eta 0:00:02
---------------------------- ----------- 9.4/13.3 MB 3.0 MB/s eta 0:00:02
---------------------------- ----------- 9.4/13.3 MB 3.0 MB/s eta 0:00:02
------------------------------- -------- 10.5/13.3 MB 3.1 MB/s eta 0:00:01
------------------------------- -------- 10.5/13.3 MB 3.1 MB/s eta 0:00:01
------------------------------- -------- 10.5/13.3 MB 3.1 MB/s eta 0:00:01
------------------------------- -------- 10.5/13.3 MB 3.1 MB/s eta 0:00:01
------------------------------- -------- 10.5/13.3 MB 3.1 MB/s eta 0:00:01
------------------------------- -------- 10.5/13.3 MB 3.1 MB/s eta 0:00:01
---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
```

```
   ---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------- ----- 11.5/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------------  13.3/13.3 MB 3.4 MB/s eta 0:00:01
   ---------------------------------------- 13.3/13.3 MB 3.4 MB/s eta 0:00:00
Downloading cmdstanpy-1.2.5-py3-none-any.whl (94 kB)
   ---------------------------------------- 0.0/94.5 kB ? eta -:--:--
   ---------------------------------------- 94.5/94.5 kB ? eta 0:00:00
Downloading holidays-0.63-py3-none-any.whl (1.2 MB)
   ---------------------------------------- 0.0/1.2 MB ? eta -:--:--
   ---------------------------------- ---- 1.0/1.2 MB ? eta -:--:--
   ---------------------------------- ---- 1.0/1.2 MB ? eta -:--:--
   ---------------------------------- ---- 1.0/1.2 MB ? eta -:--:--
   ---------------------------------- ---- 1.0/1.2 MB ? eta -:--:--
   ---------------------------------- ---- 1.0/1.2 MB ? eta -:--:--
   ---------------------------------------- 1.2/1.2 MB 4.7 MB/s eta 0:00:00
Downloading importlib_resources-6.4.5-py3-none-any.whl (36 kB)
Downloading stanio-0.5.1-py3-none-any.whl (8.1 kB)
Installing collected packages: stanio, importlib-resources, holidays, cmdstanpy, pr
ophet
Successfully installed cmdstanpy-1.2.5 holidays-0.63 importlib-resources-6.4.5 prop
het-1.1.6 stanio-0.5.1
Note: you may need to restart the kernel to use updated packages.
```

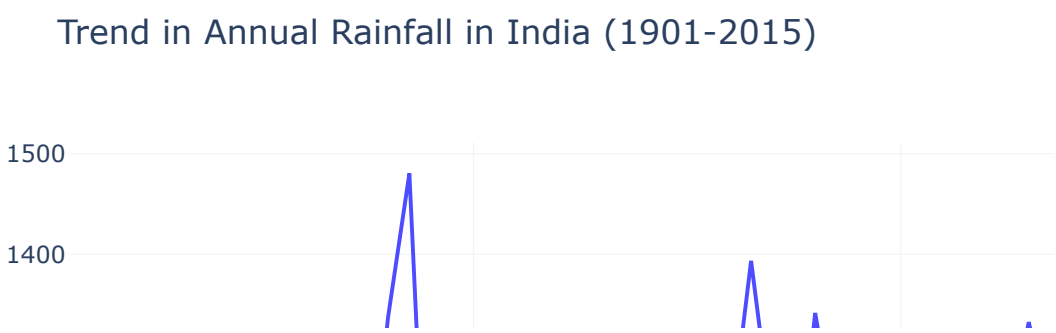
In [2]:
```python
import pandas as pd

# Update the file path to the location of your file
file_path = r"C:\Users\Sharon\Desktop\Docs\India\rainfall_area-wt_India_1901-2015.

# Read the CSV file
rainfall_data = pd.read_csv(file_path)

# Display the first few rows of the dataframe
print(rainfall_data.head())
```

```
  REGION  YEAR   JAN   FEB   MAR   APR   MAY    JUN    JUL    AUG    SEP  \
0  INDIA  1901  34.7  37.7  18.0  39.3  50.8  113.4  242.2  272.9  124.4
1  INDIA  1902   7.4   4.3  19.0  43.5  48.3  108.8  284.0  199.7  201.5
2  INDIA  1903  17.0   8.3  31.3  17.1  59.5  118.3  297.0  270.4  199.1
3  INDIA  1904  14.4   9.6  31.8  33.1  72.4  164.8  261.0  206.4  129.6
4  INDIA  1905  25.3  20.9  42.7  33.7  55.7   93.3  252.8  200.8  178.4

     OCT   NOV   DEC  ANNUAL  Jan-Feb  Mar-May  Jun-Sep  Oct-Dec
0   52.7  38.0   8.3  1032.3     72.4    108.1    752.8     99.0
1   61.5  27.9  24.4  1030.2     11.7    110.8    794.0    113.8
2  117.9  36.9  17.7  1190.5     25.3    107.9    884.8    172.5
3   69.0  11.2  16.3  1019.8     24.0    137.4    761.8     96.6
4   51.4   9.7  10.5   975.3     46.2    132.2    725.4     71.6
```

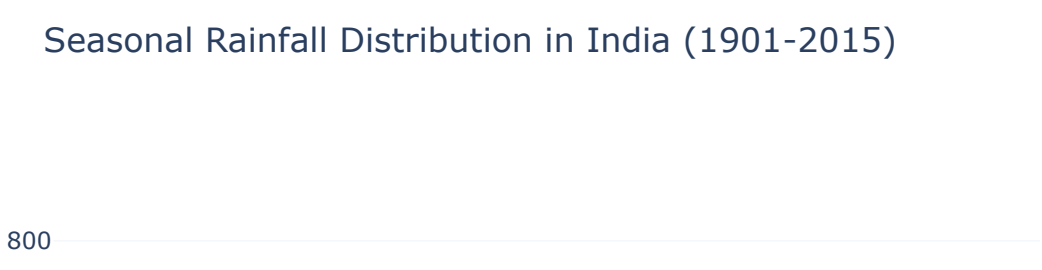
In [3]:
```python
import plotly.graph_objects as go
import plotly.express as px

# analyze trends in annual rainfall over time
annual_rainfall = rainfall_data[['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',
monthly_avg = rainfall_data[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', marke
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_data[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()
```

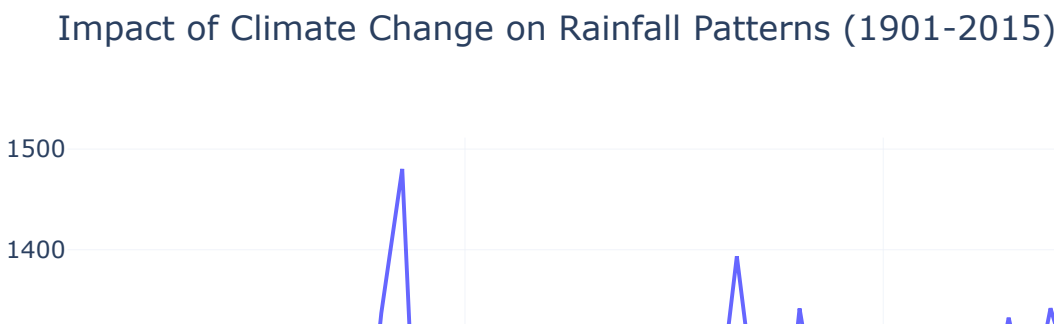



Average Monthly Rainfall in India (1901-2015)
300
250


Seasonal Rainfall Distribution in India (1901-2015)
800

In [4]:

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

fig_climate_change = go.Figure()

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

fig_climate_change.add_trace(go.Scatter(
    x=rainfall_data['YEAR'],
    y=rainfall_data['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()
```



In [5]:

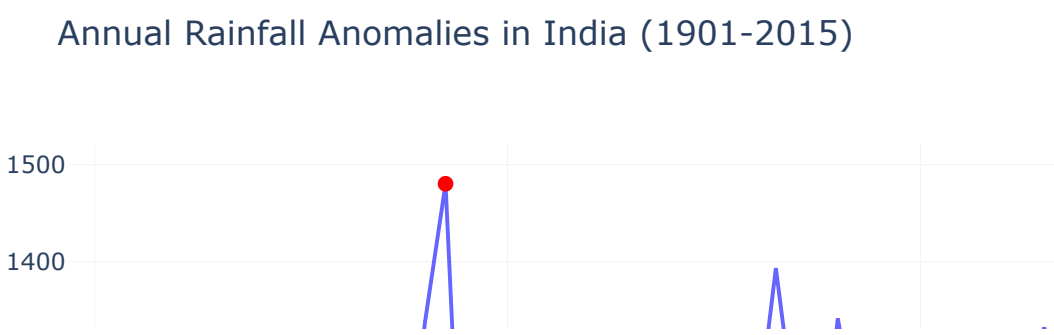
```
from scipy.stats import pearsonr

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

drought_years = rainfall_data[rainfall_data['ANNUAL'] < (mean_rainfall - 1.5 * std
extreme_rainfall_years = rainfall_data[rainfall_data['ANNUAL'] > (mean_rainfall +

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

# 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_
seasonal_correlations_summary = pd.DataFrame.from_dict(seasonal_correlations, orie

drought_years_summary, extreme_rainfall_years_summary, seasonal_correlations_summa
```

Out[5]:
```
(   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)
```

In [6]:
```
from sklearn.ensemble import IsolationForest

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

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

# detect anomalous months based on monthly data
monthly_data = rainfall_data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AU
monthly_anomalies = isolation_forest.fit_predict(monthly_data)

# add anomaly detection results for months
rainfall_data['Monthly_Anomaly'] = monthly_anomalies
monthly_anomalies_df = rainfall_data[rainfall_data['Monthly_Anomaly'] == -1][['YEA

fig_annual_anomalies = go.Figure()

fig_annual_anomalies.add_trace(go.Scatter(
    x=rainfall_data['YEAR'],
    y=rainfall_data['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_data['ANNUAL'].mean(),
    line_dash='dash',
    line_color='green',
    annotation_text='Mean Rainfall',
```

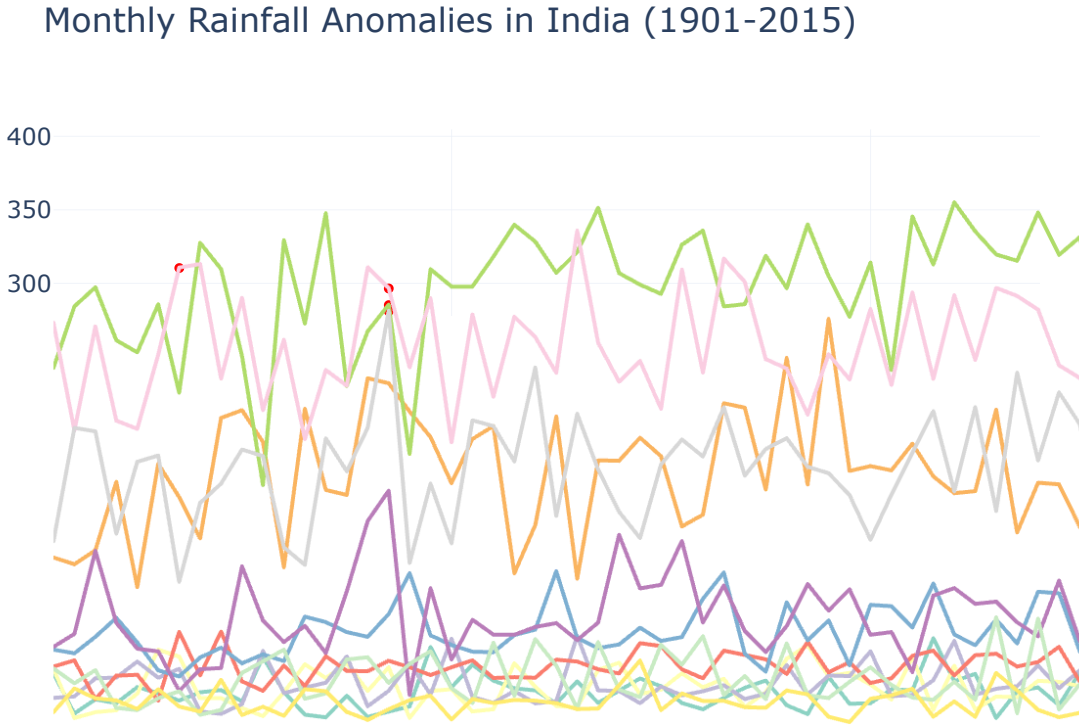
```
    annotation_position='bottom right'
)

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()
```

```
C:\Users\Sharon\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning:

X does not have valid feature names, but IsolationForest was fitted with feature na
mes

C:\Users\Sharon\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning:

X does not have valid feature names, but IsolationForest was fitted with feature na
mes
```



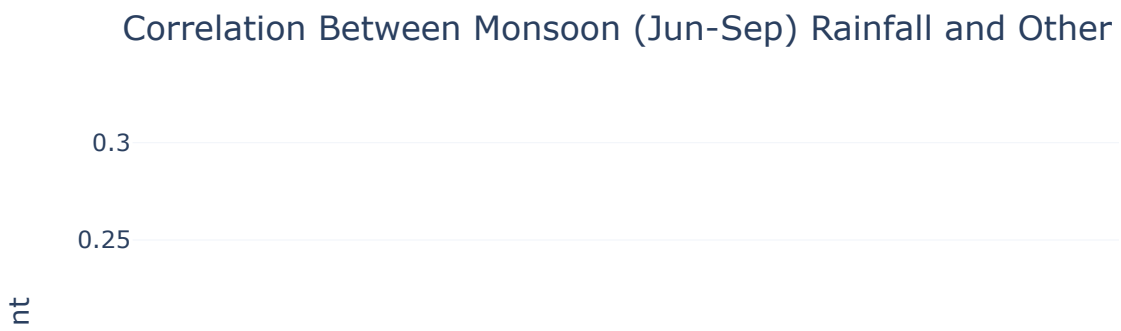
In [7]:
```
# preparing data for monthly anomalies
monthly_anomalies = []
for column in monthly_columns:
    for _, row in monthly_anomalies_df.iterrows():
        monthly_anomalies.append({'Year': row['YEAR'], 'Month': column, 'Rainfall'

monthly_anomalies_df_long = pd.DataFrame(monthly_anomalies)
```

```
fig_monthly_anomalies = px.line(
    rainfall_data,
    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()
```



In [8]:
```
# 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 = {}
```

```python
for season in seasonal_columns:
    if season != monsoon_column:
        corr, _ = pearsonr(rainfall_data[monsoon_column], rainfall_data[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 Coefficien
    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='%{
fig.update_layout(
    template='plotly_white',
    height=500
)

fig.show()
```

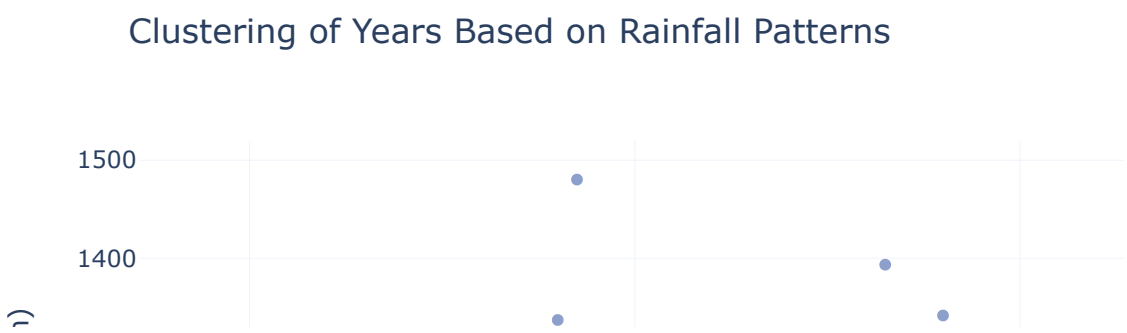


In [9]:

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

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

# perform k-means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
rainfall_data['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_data['Rainfall_Category'] = rainfall_data['Rainfall_Cluster'].map(cluster

fig = px.scatter(
    rainfall_data,
    x='YEAR',
    y='ANNUAL',
    color='Rainfall_Category',
    title='Clustering of Years Based on Rainfall Patterns',
    labels={'YEAR': 'Year', 'ANNUAL': 'Annual Rainfall (mm)', '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()
```

```
C:\Users\Sharon\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureW
arning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value o
f `n_init` explicitly to suppress the warning

C:\Users\Sharon\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWa
rning:

KMeans is known to have a memory leak on Windows with MKL, when there are less chun
ks than available threads. You can avoid it by setting the environment variable OMP
_NUM_THREADS=1.
```

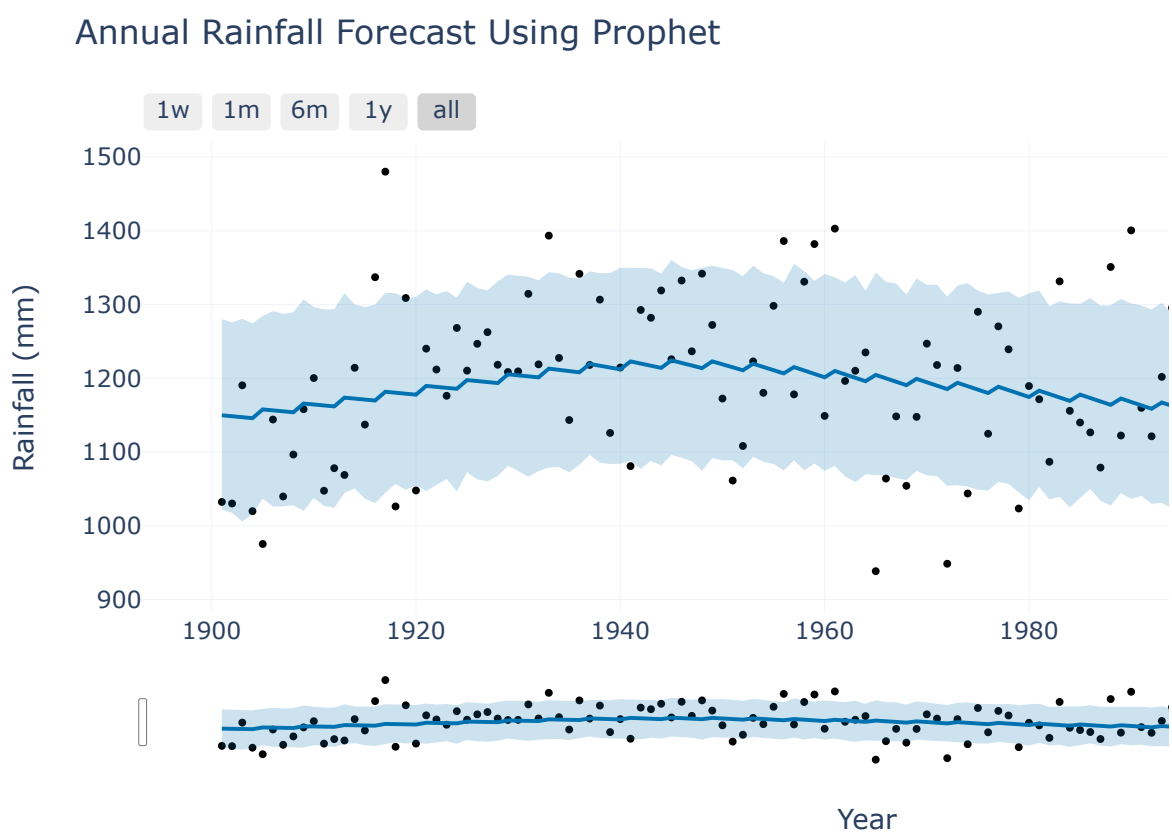


In [10]:

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

# use: pip install prophet

from prophet import Prophet
# 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='Y')
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()
```

```
12:45:08 - cmdstanpy - INFO - Chain [1] start processing
12:45:08 - cmdstanpy - INFO - Chain [1] done processing
C:\Users\Sharon\anaconda3\Lib\site-packages\_plotly_utils\basevalidators.py:106: Fu
tureWarning:

The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version
this will return a Series containing python datetime objects instead of an ndarray.
To retain the old behavior, call `np.array` on the result
```



In [ ]: