Weather data analysis project using python

by Dewansh Vishwakarma

Importing Library

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from datetime import datetime
```

Loading the Data

In [2]:
```
weather = pd.read_csv("Weather Data in India from 1901 to 2017.csv")
```

In [3]:
```
weather
```

Out[3]:

	Unnamed: 0	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	O
0	0	1901	17.99	19.43	23.49	26.41	28.28	28.60	27.49	26.98	26.26	25.
1	1	1902	19.00	20.39	24.10	26.54	28.68	28.44	27.29	27.05	25.95	24.
2	2	1903	18.32	19.79	22.46	26.03	27.93	28.41	28.04	26.63	26.34	24.
3	3	1904	17.77	19.39	22.95	26.73	27.83	27.85	26.84	26.73	25.84	24.
4	4	1905	17.40	17.79	21.78	24.84	28.32	28.69	27.67	27.47	26.29	26.
...	...	...	...	...	...	...	...	...	...	...	...	...
112	112	2013	18.88	21.07	24.53	26.97	29.06	28.24	27.50	27.22	26.87	25.
113	113	2014	18.81	20.35	23.34	26.91	28.45	29.42	28.07	27.42	26.61	25.
114	114	2015	19.02	21.23	23.52	26.52	28.82	28.15	28.03	27.64	27.04	25.
115	115	2016	20.92	23.58	26.61	29.56	30.41	29.70	28.18	28.17	27.72	26.
116	116	2017	20.59	23.08	25.58	29.17	30.47	29.44	28.31	28.12	28.11	27.

117 rows × 14 columns

Understanding the data

In [4]:
```
weather.head(5)
```

Out[4]:

	Unnamed: 0	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	0	1901	17.99	19.43	23.49	26.41	28.28	28.60	27.49	26.98	26.26	25.08
1	1	1902	19.00	20.39	24.10	26.54	28.68	28.44	27.29	27.05	25.95	24.37
2	2	1903	18.32	19.79	22.46	26.03	27.93	28.41	28.04	26.63	26.34	24.57
3	3	1904	17.77	19.39	22.95	26.73	27.83	27.85	26.84	26.73	25.84	24.36
4	4	1905	17.40	17.79	21.78	24.84	28.32	28.69	27.67	27.47	26.29	26.16

In [5]:
```
weather.tail()
```

Out[5]:

	Unnamed: 0	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	O
112	112	2013	18.88	21.07	24.53	26.97	29.06	28.24	27.50	27.22	26.87	25.
113	113	2014	18.81	20.35	23.34	26.91	28.45	29.42	28.07	27.42	26.61	25.
114	114	2015	19.02	21.23	23.52	26.52	28.82	28.15	28.03	27.64	27.04	25.
115	115	2016	20.92	23.58	26.61	29.56	30.41	29.70	28.18	28.17	27.72	26.
116	116	2017	20.59	23.08	25.58	29.17	30.47	29.44	28.31	28.12	28.11	27.

In [6]:
```
weather.shape
```

Out[6]:
```
(117, 14)
```

In [7]:
```
weather.index
```

Out[7]:
```
RangeIndex(start=0, stop=117, step=1)
```

In [8]:
```
weather.columns
```

Out[8]:
```
Index(['Unnamed: 0', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC'],
      dtype='object')
```

In [9]:
```
weather.dtypes
```

Out[9]:
```
Unnamed: 0      int64
YEAR            int64
JAN           float64
FEB           float64
MAR           float64
APR           float64
MAY           float64
JUN           float64
JUL           float64
AUG           float64
SEP           float64
OCT           float64
NOV           float64
DEC           float64
dtype: object
```

In [10]:
```
weather.count
```

Out[10]:
```
<bound method DataFrame.count of      Unnamed: 0  YEAR    JAN    FEB    MAR
APR    MAY    JUN    JUL    AUG  \
0              0  1901  17.99  19.43  23.49  26.41  28.28  28.60  27.49  26.98
1              1  1902  19.00  20.39  24.10  26.54  28.68  28.44  27.29  27.05
2              2  1903  18.32  19.79  22.46  26.03  27.93  28.41  28.04  26.63
3              3  1904  17.77  19.39  22.95  26.73  27.83  27.85  26.84  26.73
4              4  1905  17.40  17.79  21.78  24.84  28.32  28.69  27.67  27.47
..           ...   ...    ...    ...    ...    ...    ...    ...    ...    ...
112          112  2013  18.88  21.07  24.53  26.97  29.06  28.24  27.50  27.22
113          113  2014  18.81  20.35  23.34  26.91  28.45  29.42  28.07  27.42
114          114  2015  19.02  21.23  23.52  26.52  28.82  28.15  28.03  27.64
115          115  2016  20.92  23.58  26.61  29.56  30.41  29.70  28.18  28.17
116          116  2017  20.59  23.08  25.58  29.17  30.47  29.44  28.31  28.12

       SEP    OCT    NOV    DEC
0    26.26  25.08  21.73  18.95
1    25.95  24.37  21.33  18.78
2    26.34  24.57  20.96  18.29
3    25.84  24.36  21.07  18.84
4    26.29  26.16  22.07  18.71
..     ...    ...    ...    ...
112  26.87  25.63  22.18  19.69
113  26.61  25.38  22.53  19.50
114  27.04  25.82  22.95  20.21
115  27.72  26.81  23.90  21.89
116  28.11  27.24  23.92  21.47

[117 rows x 14 columns]>
```

In [11]:
```
weather.info
```

Out[11]:
```
<bound method DataFrame.info of      Unnamed: 0  YEAR    JAN    FEB    MAR    A
PR    MAY    JUN    JUL    AUG  \
0             0  1901  17.99  19.43  23.49  26.41  28.28  28.60  27.49  26.98
1             1  1902  19.00  20.39  24.10  26.54  28.68  28.44  27.29  27.05
2             2  1903  18.32  19.79  22.46  26.03  27.93  28.41  28.04  26.63
3             3  1904  17.77  19.39  22.95  26.73  27.83  27.85  26.84  26.73
4             4  1905  17.40  17.79  21.78  24.84  28.32  28.69  27.67  27.47
..          ...   ...    ...    ...    ...    ...    ...    ...    ...    ...
112         112  2013  18.88  21.07  24.53  26.97  29.06  28.24  27.50  27.22
113         113  2014  18.81  20.35  23.34  26.91  28.45  29.42  28.07  27.42
114         114  2015  19.02  21.23  23.52  26.52  28.82  28.15  28.03  27.64
115         115  2016  20.92  23.58  26.61  29.56  30.41  29.70  28.18  28.17
116         116  2017  20.59  23.08  25.58  29.17  30.47  29.44  28.31  28.12

       SEP    OCT    NOV    DEC
0    26.26  25.08  21.73  18.95
1    25.95  24.37  21.33  18.78
2    26.34  24.57  20.96  18.29
3    25.84  24.36  21.07  18.84
4    26.29  26.16  22.07  18.71
..     ...    ...    ...    ...
112  26.87  25.63  22.18  19.69
113  26.61  25.38  22.53  19.50
114  27.04  25.82  22.95  20.21
115  27.72  26.81  23.90  21.89
116  28.11  27.24  23.92  21.47

[117 rows x 14 columns]>
```

In [12]:
```
weather.describe
```

Out[12]:
```
<bound method NDFrame.describe of      Unnamed: 0  YEAR    JAN    FEB    MAR
APR    MAY    JUN    JUL    AUG  \
0             0  1901  17.99  19.43  23.49  26.41  28.28  28.60  27.49  26.98
1             1  1902  19.00  20.39  24.10  26.54  28.68  28.44  27.29  27.05
2             2  1903  18.32  19.79  22.46  26.03  27.93  28.41  28.04  26.63
3             3  1904  17.77  19.39  22.95  26.73  27.83  27.85  26.84  26.73
4             4  1905  17.40  17.79  21.78  24.84  28.32  28.69  27.67  27.47
..          ...   ...    ...    ...    ...    ...    ...    ...    ...    ...
112         112  2013  18.88  21.07  24.53  26.97  29.06  28.24  27.50  27.22
113         113  2014  18.81  20.35  23.34  26.91  28.45  29.42  28.07  27.42
114         114  2015  19.02  21.23  23.52  26.52  28.82  28.15  28.03  27.64
115         115  2016  20.92  23.58  26.61  29.56  30.41  29.70  28.18  28.17
116         116  2017  20.59  23.08  25.58  29.17  30.47  29.44  28.31  28.12

       SEP    OCT    NOV    DEC
0    26.26  25.08  21.73  18.95
1    25.95  24.37  21.33  18.78
2    26.34  24.57  20.96  18.29
3    25.84  24.36  21.07  18.84
4    26.29  26.16  22.07  18.71
..     ...    ...    ...    ...
112  26.87  25.63  22.18  19.69
113  26.61  25.38  22.53  19.50
114  27.04  25.82  22.95  20.21
115  27.72  26.81  23.90  21.89
116  28.11  27.24  23.92  21.47

[117 rows x 14 columns]>
```

Data Cleaning

In [13]:
```
weather1 = pd.melt(weather, id_vars='YEAR', value_vars=weather.columns[1:])
weather1.head()
```

Out[13]:

	YEAR	variable	value
0	1901	JAN	17.99
1	1902	JAN	19.00
2	1903	JAN	18.32
3	1904	JAN	17.77
4	1905	JAN	17.40

In [14]:
```
weather1.rename(columns = {'variable':'Month'}, inplace = True)
weather1.rename(columns = {'value':'Temp'}, inplace = True)
weather1.head()
```

Out[14]:

	YEAR	Month	Temp
0	1901	JAN	17.99
1	1902	JAN	19.00
2	1903	JAN	18.32
3	1904	JAN	17.77
4	1905	JAN	17.40

In [15]:
```
weather1['Date'] = weather1['Month'] + ' ' + weather1['YEAR'].astype(str)
weather1.loc[:,'Date'] = weather1['Date'].apply(lambda x : datetime.strptime(x,
weather1.head()
```

Out[15]:

	YEAR	Month	Temp	Date
0	1901	JAN	17.99	1901-01-01 00:00:00
1	1902	JAN	19.00	1902-01-01 00:00:00
2	1903	JAN	18.32	1903-01-01 00:00:00
3	1904	JAN	17.77	1904-01-01 00:00:00
4	1905	JAN	17.40	1905-01-01 00:00:00

In [16]:
```
weather1['season'] = np.empty(shape= weather1.shape[0], dtype= object)

for i, m in enumerate(weather1['Month']):
    season = None
    if m in ['NOV', 'DEC', 'JAN', 'FEB']:
        season = 'winter'
    elif m in ['MAR', 'APR', 'MAY']:
        season = 'summer'
    elif m in ['JUN', 'JUL', 'AUG', 'SEP', 'OCT']:
        season = 'monsoon'
```

```
    weather1.at[i, 'season'] = season
    i += 1
```

In [17]:
```
weather1.head()
```

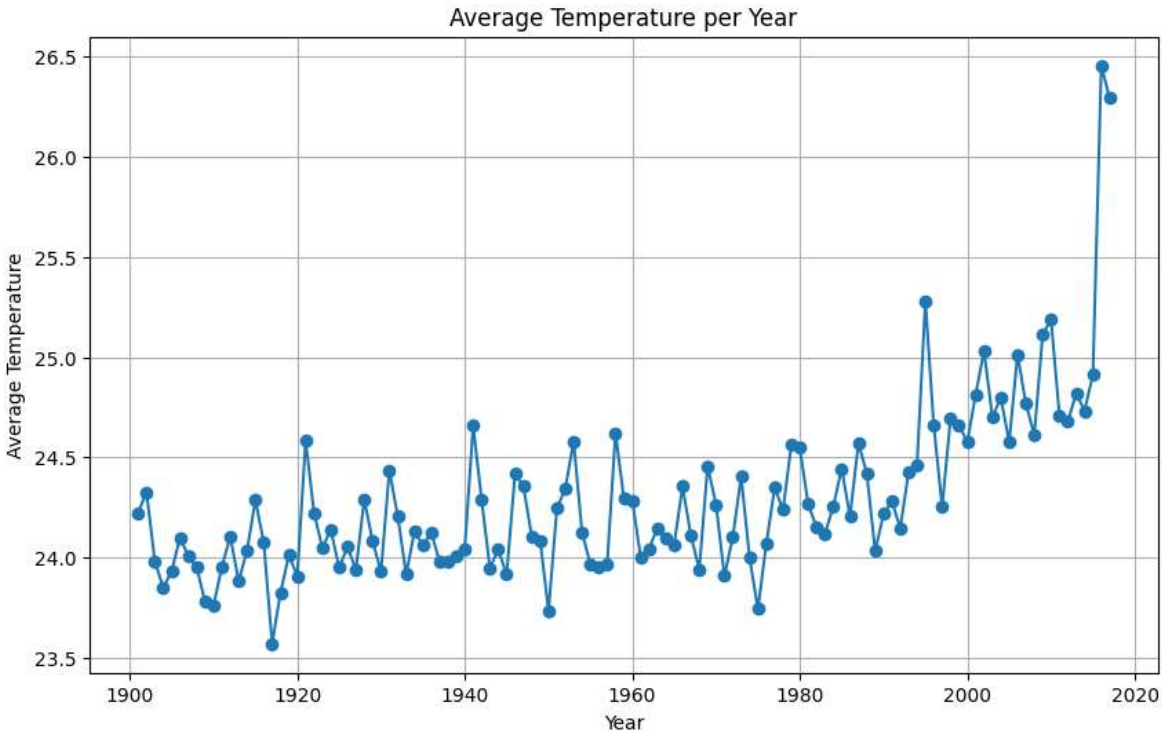
Out[17]:

	YEAR	Month	Temp	Date	season
0	1901	JAN	17.99	1901-01-01 00:00:00	winter
1	1902	JAN	19.00	1902-01-01 00:00:00	winter
2	1903	JAN	18.32	1903-01-01 00:00:00	winter
3	1904	JAN	17.77	1904-01-01 00:00:00	winter
4	1905	JAN	17.40	1905-01-01 00:00:00	winter

Visualization

Q1. How has the average temperature in India changed over the years?

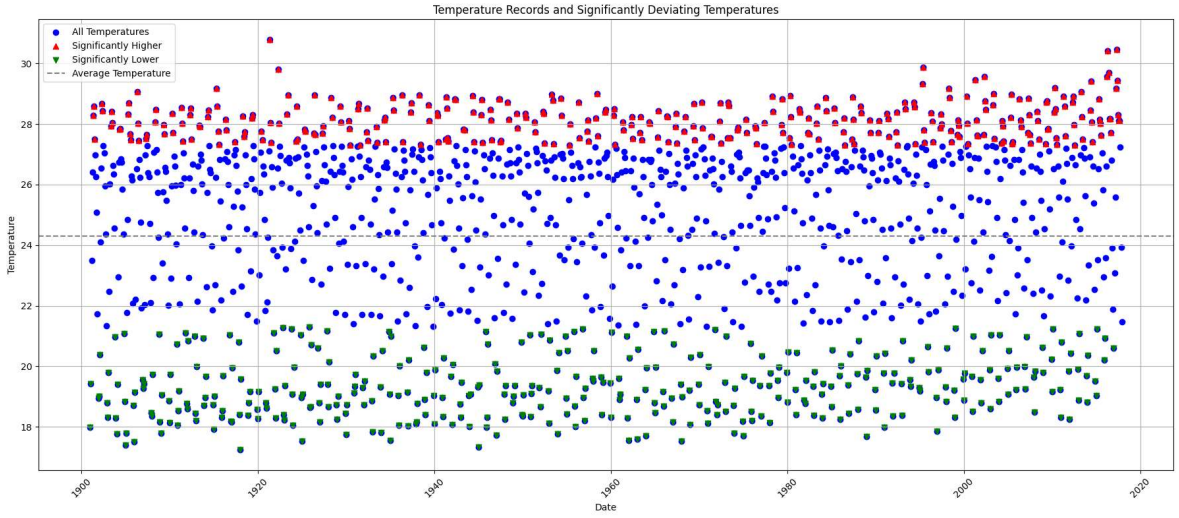
In [18]:
```
average_temp_per_year = weather1.groupby('YEAR')['Temp'].mean()

# line plot
plt.figure(figsize=(10, 6))
plt.plot(average_temp_per_year.index, average_temp_per_year.values, marker='o')
plt.title('Average Temperature per Year')
plt.xlabel('Year')
plt.ylabel('Average Temperature')
plt.grid(True)
plt.show()
```



Q2.Can you identify any instances of extreme heatwaves or cold snaps in the historical data?

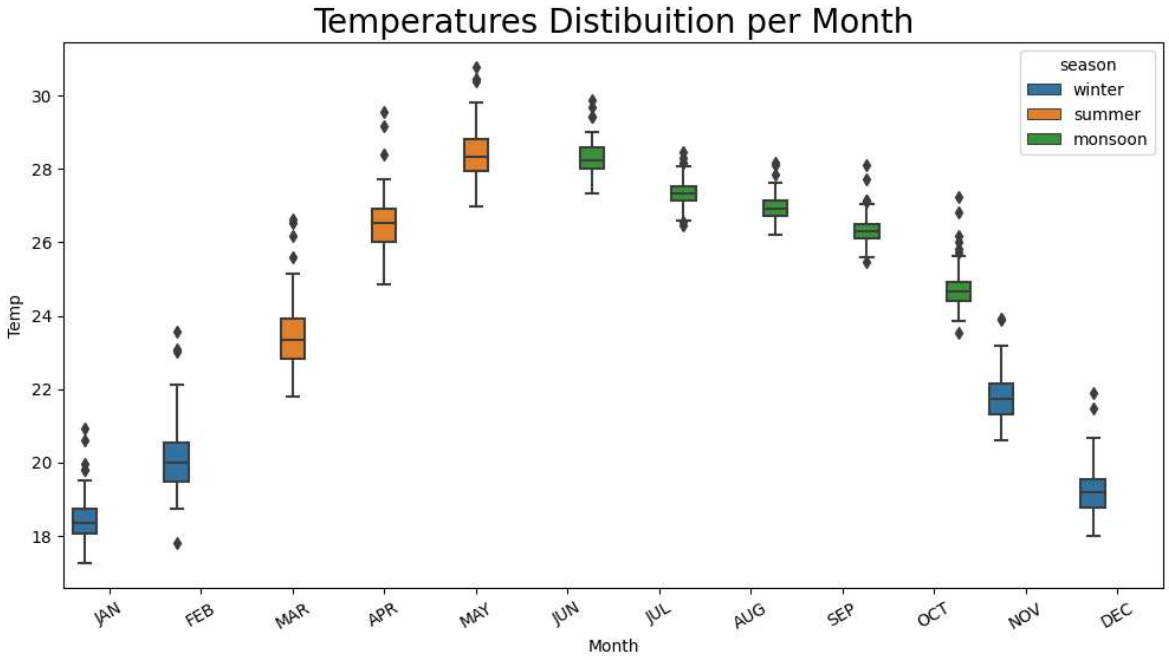
In [19]:
```
average_temp = weather1['Temp'].mean()
threshold = 3

significantly_higher = weather1[weather1['Temp'] > average_temp + threshold]
significantly_lower = weather1[weather1['Temp'] < average_temp - threshold]
```

In [20]:
```
plt.figure(figsize=(18, 8))
plt.scatter(weather1['Date'], weather1['Temp'], label='All Temperatures', color=
plt.scatter(significantly_higher['Date'], significantly_higher['Temp'], label='S
plt.scatter(significantly_lower['Date'], significantly_lower['Temp'], label='Sig

plt.axhline(y=average_temp, color='gray', linestyle='--', label='Average Tempera

plt.title('Temperature Records and Significantly Deviating Temperatures')
plt.xlabel('Date')
plt.ylabel('Temperature')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()
```



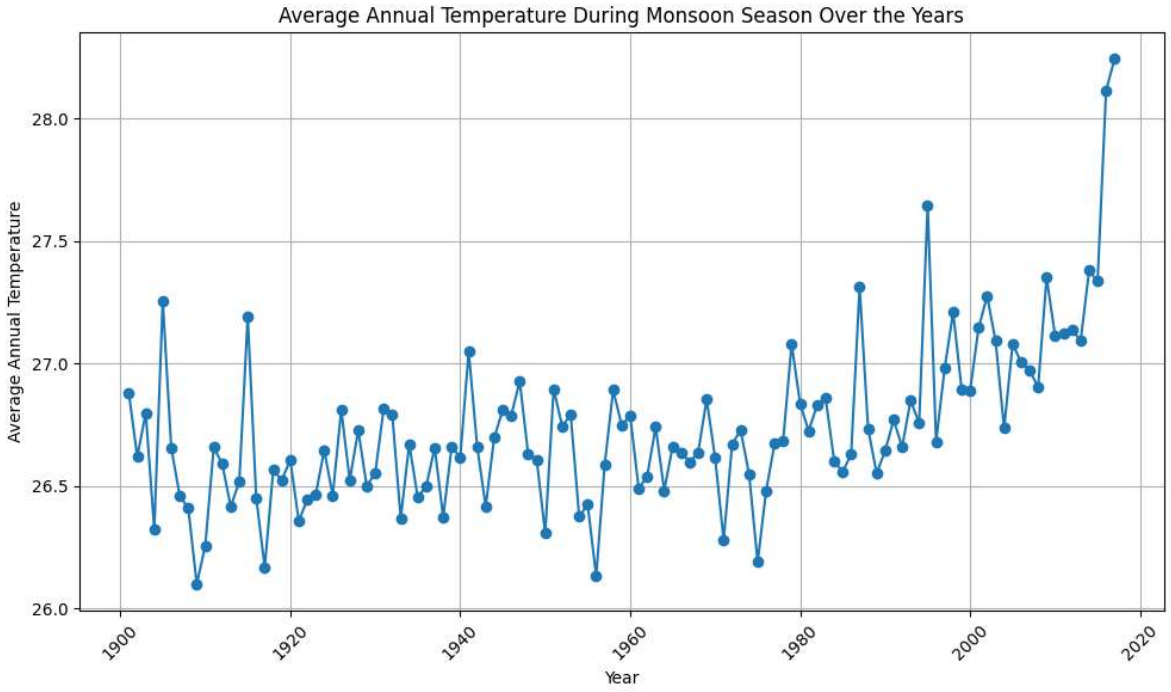
Q3.Can you identify any significant temperature variations or patterns across different months?"

In [21]:
```
plt.figure(figsize= (12, 6))
sns.boxplot(data =weather1, x= 'Month', y='Temp', hue= 'season')
plt.title('Temperatures Distibuition per Month', size= 20)
plt.xticks(rotation= 30)
plt.show()
```



Q4. What is the average annual rainfall pattern in India?

In [22]:
```
# Filter the data for the monsoon season
monsoon_data = weather1[weather1['season'] == 'monsoon']

# Group by year and calculate the average annual temperature for the monsoon sea
average_annual_temperature = monsoon_data.groupby('YEAR')['Temp'].mean()

# Visualize the distribution of temperatures for the monsoon season over the yea
plt.figure(figsize=(10, 6))
plt.plot(average_annual_temperature.index, average_annual_temperature.values, ma
plt.title('Average Annual Temperature During Monsoon Season Over the Years')
plt.xlabel('Year')
plt.ylabel('Average Annual Temperature')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```


Average Annual Temperature During Monsoon Season Over the Years
Average Annual Temperature
28.0
27.5
27.0
26.5
26.0
1900
1920
1940
1960
1980
2000
2020
Year

In [ ]: