```
import pandas as pd

data = pd.read_csv("C:\\Users\\viven\\Downloads\\Temperature\\
GlobalLandTemperaturesByCountry.csv")

data = data[data["Country"]=="India"].reset_index()

data.head()
```

```
    index          dt  AverageTemperature
AverageTemperatureUncertainty  \
0  243695  1796-01-01              17.044
2.044
1  243696  1796-02-01              19.193
1.359
2  243697  1796-03-01              22.319
2.125
3  243698  1796-04-01              27.233
1.510
4  243699  1796-05-01              30.035
1.338

  Country
0   India
1   India
2   India
3   India
4   India
```

```
data = data.drop(columns=['index', 'Country',
'AverageTemperatureUncertainty'],axis=1)
data = data.drop_duplicates()
data = data.dropna(axis=0)

data['dt'] = pd.to_datetime(data['dt'])
data
```

```
             dt  AverageTemperature
0    1796-01-01              17.044
1    1796-02-01              19.193
2    1796-03-01              22.319
3    1796-04-01              27.233
4    1796-05-01              30.035
...         ...                 ...
2607 2013-04-01              27.981
2608 2013-05-01              31.014
2609 2013-06-01              28.766
2610 2013-07-01              27.012
2611 2013-08-01              26.555

[2508 rows x 2 columns]
```

```
data['Month'] = data['dt'].dt.month_name()
data['Year'] = data['dt'].dt.year

data.head
```

```
<bound method NDFrame.head of              dt  AverageTemperature
Month  Year
0    1796-01-01              17.044   January  1796
1    1796-02-01              19.193  February  1796
2    1796-03-01              22.319     March  1796
3    1796-04-01              27.233     April  1796
4    1796-05-01              30.035       May  1796
...         ...                 ...       ...   ...
2607 2013-04-01              27.981     April  2013
2608 2013-05-01              31.014       May  2013
2609 2013-06-01              28.766      June  2013
2610 2013-07-01              27.012      July  2013
2611 2013-08-01              26.555    August  2013

[2508 rows x 4 columns]>
```

```
data = data.drop(columns=['dt'], axis=1)
data.head
```

```
<bound method NDFrame.head of       AverageTemperature     Month  Year
0                 17.044   January  1796
1                 19.193  February  1796
2                 22.319     March  1796
3                 27.233     April  1796
4                 30.035       May  1796
...                  ...       ...   ...
2607              27.981     April  2013
2608              31.014       May  2013
2609              28.766      June  2013
2610              27.012      July  2013
2611              26.555    August  2013

[2508 rows x 3 columns]>
```

```
data_copy = data.copy()

data = data.pivot(columns='Month', index='Year',
values='AverageTemperature')
data.head
```

```
<bound method NDFrame.head of Month   April  August  December
February  January    July    June   March  \
Year

1796   27.233  26.558    15.485    19.193   17.044  26.800  29.261
22.319
```

```
1797   26.502  26.716    17.102       NaN      NaN  26.891  29.880
22.879
1798      NaN  27.009    16.728    19.525   17.226  27.968  29.369
24.328
1799   27.996  26.674    16.904    17.976   16.648  27.178  29.860
23.080
1800   28.458  26.069    17.548    19.780   17.081  26.494  29.261
22.938
...       ...     ...       ...       ...      ...     ...     ...
...
2009   28.665  27.379    18.631    21.516   18.661  27.611  30.409
25.083
2010   29.814  26.892    17.455    20.764   17.109  27.433  29.908
26.373
2011   27.273  26.533    18.173    20.070   16.478  27.209  28.871
24.595
2012   28.067  26.735    18.622    19.791   16.778  27.699  30.536
24.354
2013   27.981  26.555       NaN    20.243   17.160  27.012  28.766
24.575

Month     May  November  October  September
Year
1796   30.035    20.186   24.031     25.958
1797   29.364    19.323   23.670     26.072
1798      NaN    20.709   23.898     25.973
1799   30.217    20.158   24.597     26.105
1800   29.372    20.625   24.318     24.999
...       ...       ...      ...        ...
2009   30.493    21.457   24.775     27.080
2010   31.169    22.204   25.193     26.296
2011   30.421    21.982   25.061     26.321
2012   30.805    21.162   24.590     26.551
2013   31.014       NaN      NaN        NaN

[211 rows x 12 columns]>
```

```
titles = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December']
titles
```

```
['January',
 'February',
 'March',
 'April',
 'May',
 'June',
 'July',
 'August',
 'September',
 'October',
 'November',
 'December']

data = data[titles]
data.head

<bound method NDFrame.head of Month  January  February   March   April
May    June    July  August  \
Year

1796    17.044    19.193  22.319  27.233  30.035  29.261  26.800
26.558
1797       NaN       NaN  22.879  26.502  29.364  29.880  26.891
26.716
1798    17.226    19.525  24.328     NaN     NaN  29.369  27.968
27.009
1799    16.648    17.976  23.080  27.996  30.217  29.860  27.178
26.674
1800    17.081    19.780  22.938  28.458  29.372  29.261  26.494
26.069
...        ...       ...     ...     ...     ...     ...     ...     .
..
2009    18.661    21.516  25.083  28.665  30.493  30.409  27.611
27.379
2010    17.109    20.764  26.373  29.814  31.169  29.908  27.433
26.892
2011    16.478    20.070  24.595  27.273  30.421  28.871  27.209
26.533
2012    16.778    19.791  24.354  28.067  30.805  30.536  27.699
26.735
2013    17.160    20.243  24.575  27.981  31.014  28.766  27.012
26.555

Month  September  October  November  December
Year
1796      25.958   24.031    20.186    15.485
1797      26.072   23.670    19.323    17.102
1798      25.973   23.898    20.709    16.728
1799      26.105   24.597    20.158    16.904
1800      24.999   24.318    20.625    17.548
...          ...      ...       ...       ...
2009      27.080   24.775    21.457    18.631
2010      26.296   25.193    22.204    17.455
2011      26.321   25.061    21.982    18.173
2012      26.551   24.590    21.162    18.622
2013         NaN      NaN       NaN       NaN

[211 rows x 12 columns]>
```

```
data = data.ffill()
data.head

<bound method NDFrame.head of Month  January  February   March   April
May    June    July  August  \
Year

1796    17.044    19.193  22.319  27.233  30.035  29.261  26.800
26.558
1797    17.044    19.193  22.879  26.502  29.364  29.880  26.891
26.716
1798    17.226    19.525  24.328  26.502  29.364  29.369  27.968
27.009
1799    16.648    17.976  23.080  27.996  30.217  29.860  27.178
26.674
1800    17.081    19.780  22.938  28.458  29.372  29.261  26.494
26.069
...        ...       ...     ...     ...     ...     ...     ...     .
..
2009    18.661    21.516  25.083  28.665  30.493  30.409  27.611
27.379
2010    17.109    20.764  26.373  29.814  31.169  29.908  27.433
26.892
2011    16.478    20.070  24.595  27.273  30.421  28.871  27.209
26.533
2012    16.778    19.791  24.354  28.067  30.805  30.536  27.699
26.735
2013    17.160    20.243  24.575  27.981  31.014  28.766  27.012
26.555

Month  September  October  November  December
Year
1796      25.958   24.031    20.186    15.485
1797      26.072   23.670    19.323    17.102
1798      25.973   23.898    20.709    16.728
1799      26.105   24.597    20.158    16.904
1800      24.999   24.318    20.625    17.548
...          ...      ...       ...       ...
2009      27.080   24.775    21.457    18.631
2010      26.296   25.193    22.204    17.455
2011      26.321   25.061    21.982    18.173
2012      26.551   24.590    21.162    18.622
2013      26.551   24.590    21.162    18.622

[211 rows x 12 columns]>

data['Summer Average'] = data[['April', 'May', 'June']].mean(axis=1)
data['Annual Average'] = data[['January', 'February', 'March',
'April', 'May', 'June', 'July', 'August', 'September', 'October',
'November', 'December']].mean(axis=1)
```

```
data.head

<bound method NDFrame.head of Month  January  February   March   April
May    June    July  August  \
Year

1796    17.044    19.193  22.319  27.233  30.035  29.261  26.800
26.558
1797    17.044    19.193  22.879  26.502  29.364  29.880  26.891
26.716
1798    17.226    19.525  24.328  26.502  29.364  29.369  27.968
27.009
1799    16.648    17.976  23.080  27.996  30.217  29.860  27.178
26.674
1800    17.081    19.780  22.938  28.458  29.372  29.261  26.494
26.069
...        ...       ...     ...     ...     ...     ...     ...     .
..
2009    18.661    21.516  25.083  28.665  30.493  30.409  27.611
27.379
2010    17.109    20.764  26.373  29.814  31.169  29.908  27.433
26.892
2011    16.478    20.070  24.595  27.273  30.421  28.871  27.209
26.533
2012    16.778    19.791  24.354  28.067  30.805  30.536  27.699
26.735
2013    17.160    20.243  24.575  27.981  31.014  28.766  27.012
26.555

Month  September  October  November  December  Summer Average  Annual
Average
Year

1796       25.958   24.031    20.186    15.485       28.843000
23.675250
1797       26.072   23.670    19.323    17.102       28.582000
23.719667
1798       25.973   23.898    20.709    16.728       28.411667
24.049917
1799       26.105   24.597    20.158    16.904       29.357667
23.949417
1800       24.999   24.318    20.625    17.548       29.030333
23.911917
...           ...      ...       ...       ...             ...
...
2009       27.080   24.775    21.457    18.631       29.855667
25.146667
2010       26.296   25.193    22.204    17.455       30.297000
25.050833
```

```
2011       26.321   25.061    21.982    18.173        28.855000
24.415583
2012       26.551   24.590    21.162    18.622        29.802667
24.640833
2013       26.551   24.590    21.162    18.622        29.253667
24.519250

[211 rows x 14 columns]>

import plotly.graph_objects as go
import plotly.express as px

annual_temperature = data[['Annual Average']].reset_index()

fig_annual = go.Figure()
fig_annual.add_trace(go.Scatter(
    x=annual_temperature['Year'],
    y=annual_temperature['Annual Average'],
    mode='lines',
    name='Annual Temperature',
    line=dict(color='blue', width=2),
    opacity=0.7
))
fig_annual.add_trace(go.Scatter(
    x=annual_temperature['Year'],
    y=[annual_temperature['Annual Average'].mean()] *
len(annual_temperature),
    mode='lines',
    name='Mean Temperature',
    line=dict(color='red', dash='dash')
))
fig_annual.update_layout(
    title='Trend in Annual Temperature in India (1796-2013)',
    xaxis_title='Year',
    yaxis_title='Temperature',
    template='plotly_white',
    legend=dict(title="Legend"),
    height=500
)
fig_annual.show()
```

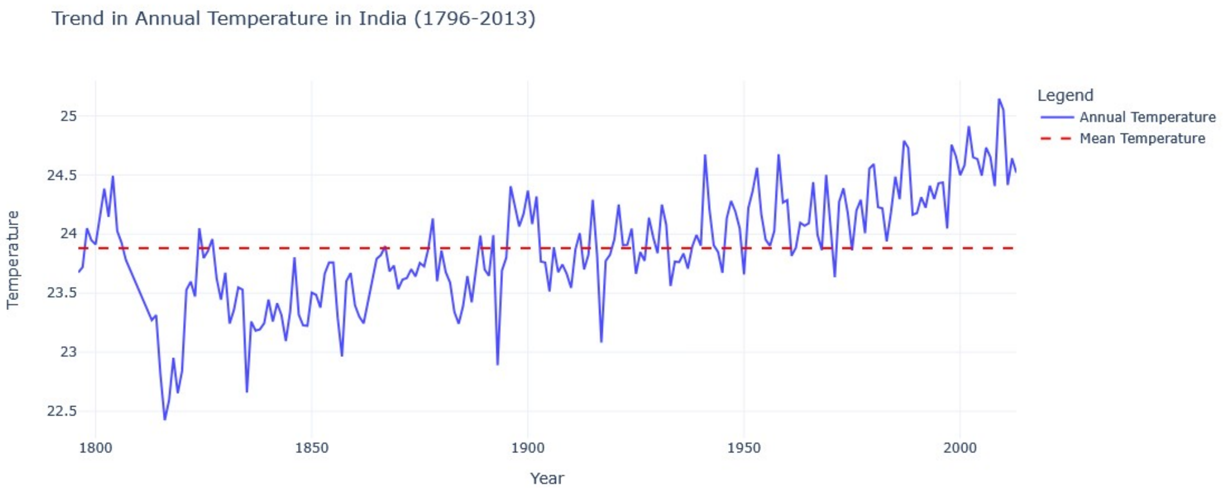
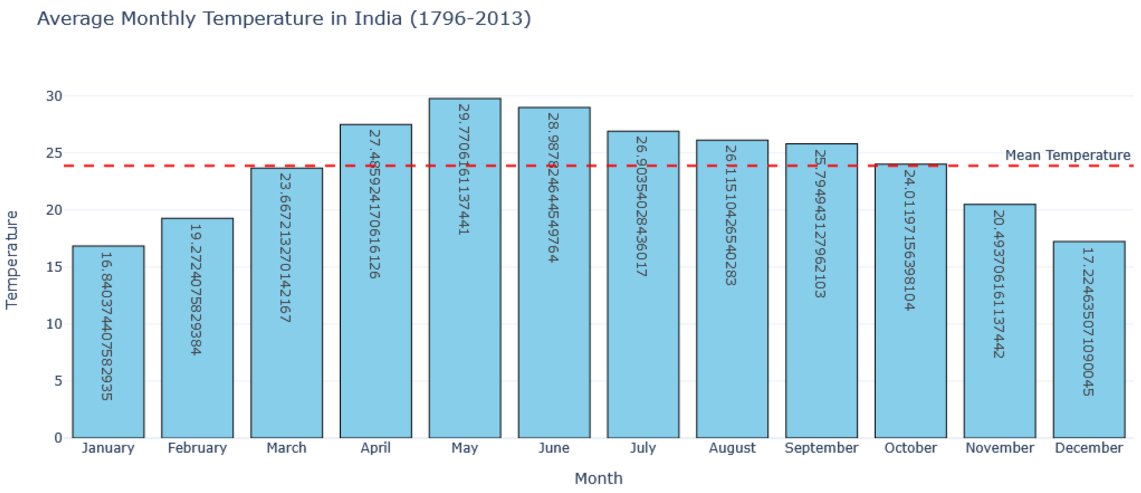


Figure shows year-to-year variability in India’s recorded temperatures, with no apparent long-term upward or downward trend over the century. However a steady rise in temperature oever the year is observed The red dashed line indicates the mean temperature, around which the temperature reorded over the months in a year oscillates. Notable peaks and troughs highlight extreme dry and cold years respectively.

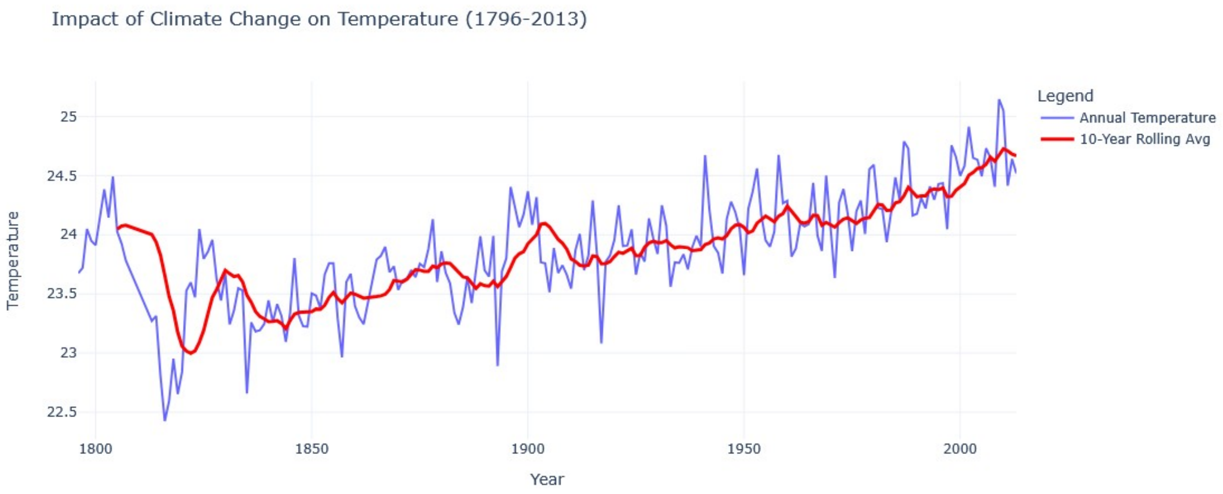
```
monthly_columns = ['January', 'February', 'March', 'April', 'May',
'June', 'July', 'August', 'September', 'October', 'November',
'December']
monthly_avg = data[monthly_columns].mean()

highest_temperature_month = monthly_avg.idxmax()
lowest_temperature_month = monthly_avg.idxmin()

fig_monthly = px.bar(
    x=monthly_avg.index,
    y=monthly_avg.values,
    labels={'x': 'Month', 'y': 'Temperature'},
    title='Average Monthly Temperature in India (1796-2013)',
    text=monthly_avg.values
)
fig_monthly.add_hline(
    y=monthly_avg.mean(),
    line_dash="dash",
    line_color="red",
    annotation_text="Mean Temperature",
    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()
```



Bar chart illustrates a highly uneven variation of recorded temperature across months, with April, May and June recording the highest average temperature months. The red dashed line represents the mean monthly temperature.

From this figure we can safely conclude peak summer seasons and others: April-May-June: Summer July-August-September: Monsoon November-December-January: Winter

```
data['10-Year Rolling Avg'] = data['Annual
Average'].rolling(window=10).mean()

fig_climate_change = go.Figure()

fig_climate_change.add_trace(go.Scatter(
    x=annual_temperature['Year'],
    y=data['Annual Average'],
    mode='lines',
    name='Annual Temperature',
    line=dict(color='blue', width=2),
    opacity=0.6
))

fig_climate_change.add_trace(go.Scatter(
    x=annual_temperature['Year'],
    y=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 Temperature (1796-2013)',
    xaxis_title='Year',
    yaxis_title='Temperature',
    template='plotly_white',
    legend=dict(title="Legend"),
    height=500
)

fig_climate_change.show()
```



This graph shows the annual temperature trends in India (blue line) and a 10-year rolling average (red line) to identify long-term patterns. While annual temperature exhibits significant variability but keeps a slight upward trend, the 10-year rolling average also indicates a slight upward continous trend post-1912, which suggests a possible impact of climate change on rising temperature.

```
data_copy.drop(columns=['Month'], axis=1, inplace=True)

annual_avg_temp = data_copy.groupby('Year')
['AverageTemperature'].mean().reset_index()

annual_avg_temp
```

```
     Year  AverageTemperature
0    1796           23.675250
1    1797           24.839900
2    1798           23.273300
3    1799           23.949417
4    1800           23.911917
..    ...                 ...
206  2009           25.146667
207  2010           25.050833
208  2011           24.415583
209  2012           24.640833
210  2013           25.413250
```

```
[211 rows x 2 columns]
```

Predictive analytics

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

all_years = pd.DataFrame({'Year': range(1796, 2024)})

merged_df = pd.merge(all_years, annual_avg_temp, on='Year',
how='left')

merged_df
```

```
     Year  AverageTemperature
0    1796           23.675250
1    1797           24.839900
2    1798           23.273300
3    1799           23.949417
4    1800           23.911917
..    ...                 ...
223  2019                 NaN
224  2020                 NaN
225  2021                 NaN
226  2022                 NaN
227  2023                 NaN

[228 rows x 2 columns]
```

```
known_data = merged_df.dropna()
missing_data = merged_df[merged_df['AverageTemperature'].isna()]

X_train = known_data[['Year']]
y_train = known_data['AverageTemperature']
X_test = missing_data[['Year']]

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

merged_df.loc[merged_df['AverageTemperature'].isnull(),
'AverageTemperature'] = y_pred

print(merged_df)
```

```
     Year  AverageTemperature
0    1796           23.675250
1    1797           24.839900
```

```
2    1798           23.273300
3    1799           23.949417
4    1800           23.911917
..    ...                 ...
223  2019           24.510931
224  2020           24.516782
225  2021           24.522633
226  2022           24.528485
227  2023           24.534336

[228 rows x 2 columns]
```

```
print(y_pred)
```

```
[23.27628193 23.28213334 23.28798476 23.29383618 23.29968759
23.59810979
 23.60396121 24.48167356 24.48752497 24.49337639 24.4992278
24.50507922
 24.51093063 24.51678205 24.52263347 24.52848488 24.5343363 ]
```