```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files und

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the
```

```
/kaggle/input/india-power/Powerdata.csv
/kaggle/input/india-power/long_data.csv
```

```
!pip install bar_chart_race
```

```
Collecting bar_chart_race
  Downloading bar_chart_race-0.1.0-py3-none-any.whl (156 kB)
     |████████████████████████████████| 156 kB 926 kB/s
Requirement already satisfied: matplotlib>=3.1 in /opt/conda/lib/python3.7/site-packages
Requirement already satisfied: pandas>=0.24 in /opt/conda/lib/python3.7/site-packages (f
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.7/site-pac
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/lib/python3.7/site-package
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.7/site-packages (
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.7/site-packag
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.7/site-packages
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (f
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packag
Requirement already satisfied: numpy>=1.17 in /opt/conda/lib/python3.7/site-packages (fr
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (f
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-packages (from
Installing collected packages: bar-chart-race
Successfully installed bar-chart-race-0.1.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib notebook
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
```

```
from IPython.display import HTML
import calendar
from plotly.subplots import make_subplots
import bar_chart_race as bcr
```

```
/opt/conda/lib/python3.7/site-packages/geopandas/_compat.py:115: UserWarning: The Shapel
  shapely_geos_version, geos_capi_version_string
```

```
df = pd.read_csv('../input/india-power/Powerdata.csv')
df_long = pd.read_csv('../input/india-power/long_data.csv')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 35 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Date               210 non-null    object 
 1   Punjab             210 non-null    float64
 2   Haryana            210 non-null    float64
 3   Rajasthan          210 non-null    float64
 4   Delhi              210 non-null    float64
 5   UP                 210 non-null    float64
 6   Uttarakhand        210 non-null    float64
 7   HP                 210 non-null    float64
 8   J&K                210 non-null    float64
 9   Chandigarh         210 non-null    float64
 10  Chhattisgarh       210 non-null    float64
 11  Gujarat            210 non-null    float64
 12  MP                 210 non-null    float64
 13  Maharashtra        210 non-null    float64
 14  Goa                210 non-null    float64
 15  DNH                210 non-null    float64
 16  Andhra Pradesh     210 non-null    float64
 17  Telangana          210 non-null    float64
 18  Karnataka          210 non-null    float64
 19  Kerala             210 non-null    float64
 20  Tamil Nadu         210 non-null    float64
 21  Pondy              210 non-null    float64
 22  Bihar              210 non-null    float64
 23  Jharkhand          210 non-null    float64
 24  Odisha             210 non-null    float64
 25  West Bengal        210 non-null    float64
 26  Sikkim             210 non-null    float64
 27  Arunachal Pradesh  210 non-null    float64
 28  Assam              210 non-null    float64
 29  Manipur            210 non-null    float64
 30  Meghalaya          210 non-null    float64
 31  Mizoram            210 non-null    float64
 32  Nagaland           210 non-null    float64
 33  Tripura            210 non-null    float64
 34  ALL INDIA TOTAL    210 non-null    float64
dtypes: float64(34), object(1)
memory usage: 57.5+ KB
```

```
#df['Date'] = pd.to_datetime(df.Date, dayfirst=True)
df_long['Dates'] = pd.to_datetime(df_long.Dates, dayfirst=True)
```

RegionWise Analysis

```
df['NR'] = df['Punjab']+ df['Haryana']+ df['Rajasthan']+ df['Delhi']+df['UP']+df['Uttarakhand

df['WR'] = df['Chhattisgarh']+df['Gujarat']+df['MP']+df['Maharashtra']+df['Goa']+df['DNH']

df['SR'] = df['Andhra Pradesh']+df['Telangana']+df['Karnataka']+df['Kerala']+df['Tamil Nadu']

df['ER'] = df['Bihar']+df['Jharkhand']+ df['Odisha']+df['West Bengal']+df['Sikkim']

df['NER'] =df['Arunachal Pradesh']+df['Assam']+df['Manipur']+df['Meghalaya']+df['Mizoram']+df
```

```
fig = go.Figure()

fig.add_trace(go.Scatter(
    x=df.Date, y=df.NR,
    mode='lines+markers',
    name='Northern region',
    marker=dict(
            color='rgba(300, 50, 50, 0.8)',
            size=5,
            line=dict(
                color='DarkSlateGrey',
                width = 1
                     )
                )
))

fig.add_trace(go.Scatter(
    x=df.Date, y=df.SR,
    mode='lines+markers',
    name='Southern Region',
    marker=dict(
            color='rgba(50, 300, 50, 0.8)',
            size=5,
            line=dict(
                color='DarkSlateGrey',
                width = 1
                     )
                )
))
```

```
fig.add_trace(go.Scatter(
    x=df.Date, y=df.ER,
    mode='lines+markers',
    name='Eastern Region',
    marker=dict(
            color='rgba(50, 50, 300, 0.8)',
            size=5,
            line=dict(
                color='DarkSlateGrey',
                width = 1
                     )
                )
))

fig.add_trace(go.Scatter(
    x=df.Date, y=df.WR,
    mode='lines+markers',
    name='Western Region',
    marker=dict(
            color='rgba(300, 100, 200, 0.8)',
            size=5,
            line=dict(
                color='DarkSlateGrey',
                width = 1
                     )
                )
))

fig.add_trace(go.Scatter(
    x=df.Date, y=df.NER,
    mode='lines+markers',
    name='North-Eastern',
    marker=dict(
            color='rgba(100, 200, 300, 0.8)',
            size=5,
            line=dict(
                color='DarkSlateGrey',
                width = 1
                     )
                )
))


fig.update_xaxes(
    rangeslider_visible=True,
    rangeselector=dict(
        buttons=list([
            dict(count=1, label="1m", step="month", stepmode="backward"),
            dict(count=3, label="3m", step="month", stepmode="backward"),
            dict(count=6, label="6m", step="month", stepmode="backward"),
            dict(step="all")
        ])
    )
)

fig.update_layout(title='Power Consumption in Various Region')
fig.update_layout(width=800,height=500)
fig.show()
```

```
    df1= df[['Date', 'Punjab', 'Haryana', 'Rajasthan', 'Delhi', 'UP',
           'Uttarakhand', 'HP', 'J&K', 'Chandigarh', 'Chhattisgarh', 'Gujarat',
           'MP', 'Maharashtra', 'Goa', 'DNH',
           'Andhra Pradesh', 'Telangana', 'Karnataka', 'Kerala', 'Tamil Nadu',
           'Pondy', 'Bihar', 'Jharkhand', 'Odisha', 'West Bengal', 'Sikkim',
           'Arunachal Pradesh', 'Assam', 'Manipur', 'Meghalaya', 'Mizoram',
           'Nagaland', 'Tripura']]

    df1 = df1.set_index('Date')
    bcr.bar_chart_race(df1, figsize=(4, 3.5),period_length =500,filename = None, title='power usa
         /opt/conda/lib/python3.7/site-packages/bar_chart_race/_make_chart.py:286: UserWarning:

         FixedFormatter should only be used together with FixedLocator

         /opt/conda/lib/python3.7/site-packages/bar_chart_race/_make_chart.py:287: UserWarning:

         FixedFormatter should only be used together with FixedLocator
```

Monthly

```
monthly_df = df_long.groupby([df_long.Dates.dt.year, df_long.Dates.dt.month,df_long.States,df
monthly_df.index = monthly_df.index.set_names(['year', 'month','State','Region','latitude','l
monthly_df = monthly_df.reset_index()
monthly_df['month'] = monthly_df['month'].apply(lambda x: calendar.month_abbr[x])
```

```
monthly_df.head()
```

	year	month	State	Region	latitude	longitude	Usage

```
fig = px.sunburst(monthly_df, path=['Region', 'State','month'], values='Usage',
                  color='Usage',
                  color_continuous_scale='RdBu')
fig.update_layout(title='Click various Regions/States to view power distribution')
fig.update_layout( width=800,height=600)
fig.show()
```

```
fig = px.bar(monthly_df, x="Region", y="Usage",color='State',animation_frame = 'month')
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(title='Region-wise Bar plots')
fig.show()
```

Lockdown

```
df_before = df.iloc[0:150,:]
df_after = df.iloc[151:,]


fig = go.Figure()
fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Gujarat'], name='Gujarat before l
    line=dict(width=2,dash='dot',color='firebrick')
))
fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Maharashtra'], name='Maharashtra
    line=dict(width=2,dash='dot',color='coral')
))

fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['MP'], name='MP before lockdown',f
    line=dict(width=2,dash='dot',color='darkred')
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Gujarat'],name='Gujarat after lockdo
    line=dict(color='firebrick', width=2)
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Maharashtra'],name='Maharashtra afte
    line=dict(color='coral', width=2)
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['MP'],name='MP after lockdown',fill='
    line=dict(color='darkred', width=2)
))

fig.update_layout(title='Power Consumption in top 3 WR states')
fig.update_layout( width=800,height=500)
fig.show()
```

```
    fig = go.Figure()
    fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Karnataka'], name='Karnataka befo
        line=dict(width=2,dash='dot',color='skyblue')
    ))
    fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Tamil Nadu'], name='Tamil Nadu be
        line=dict(width=2,dash='dot',color='lightblue')
    ))

    fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Telangana'], name='Telangana befo
        line=dict(width=2,dash='dot',color='midnightblue')
    ))
```

```
fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Karnataka'],name='Karnataka after lo
    line=dict(color='skyblue', width=2)
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Tamil Nadu'],name='Tamil Nadu after
    line=dict(color='lightblue', width=2)
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Telangana'],name='Telangana after lo
    line=dict(color='midnightblue', width=2)
))

fig.update_layout(title='Power Consumption in top 3 WR states')
fig.update_layout( width=800,height=500)
fig.show()
```

```
fig = go.Figure()
fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Rajasthan'], name='Rajasthan befo
    line=dict(width=2,dash='dot',color='darkviolet')
))

fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['UP'], name='UP before lockdown',f
    line=dict(width=2,dash='dot',color='deeppink')
))


fig.add_trace(go.Scatter( x=df_before['Date'], y=df_before['Haryana'], name='Haryana before l
    line=dict(width=2,dash='dot',color='indigo')
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Rajasthan'],name='Rajasthan after lo
    line=dict(color='darkviolet', width=2)
))

fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['UP'],name='UP after lockdown',fill='
    line=dict(color='deeppink', width=2)
))


fig.add_trace(go.Scatter(x=df_after['Date'], y=df_after['Haryana'],name='Haryana after lockdo
    line=dict(color='indigo', width=2)
))

fig.update_layout(title='Power Consumption in top 3 NR states')
fig.update_layout( width=800,height=500)
fig.show()
```

MAx

```
WR_df = df_long[df_long['Regions']=='WR']
NR_df = df_long[df_long['Regions']=='NR']
SR_df = df_long[df_long['Regions']=='SR']
ER_df = df_long[df_long['Regions']=='ER']
NER_df = df_long[df_long['Regions']=='NER']
```

```
fig= go.Figure(go.Indicator(
    mode = "gauge+number",
    value = WR_df['Usage'].max(),
    title = {'text': "Max Power Usage In WR:Maharashtra 13/05/2020"},
    gauge = {
        'axis': {'range': [None, 500], 'tickwidth': 1},
        'threshold': {
            'line': {'color': "red", 'width': 4},
            'thickness': 0.75,
            'value': 490}}
))

fig.show()
```

```
fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = NR_df['Usage'].max(),
    title = {'text': "Max Power Usage In NR :UP 09/05/2020"},
    gauge = {
        'axis': {'range': [None, 500], 'tickwidth': 1},
        'threshold': {
            'line': {'color': "red", 'width': 4},
            'thickness': 0.75,
            'value': 490}}
))
fig.update_layout(legend_title_text='State   Date::UP')
fig.show()
```

```
fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = SR_df['Usage'].max(),
    title = {'text': "Max Power Usage In SR : Tamil Nadu  01/11/2019"},
    gauge = {
        'axis': {'range': [None, 500], 'tickwidth': 1},
        'threshold': {
            'line': {'color': "red", 'width': 4},
            'thickness': 0.75,
            'value': 490}}
))

fig.show()
```

```
fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = ER_df['Usage'].max(),
    title = {'text': "Max Power Usage In ER: West Bangal 04/05/2020"},
    gauge = {
        'axis': {'range': [None, 500], 'tickwidth': 1},
        'threshold': {
            'line': {'color': "red", 'width': 4},
            'thickness': 0.75,
            'value': 490}}
))
```

```
fig.show()
```

```
fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = NER_df['Usage'].max(),
    title = {'text': "Max Power Usage In NER: Assam 05/05/2020"},
    gauge = {
        'axis': {'range': [None, 500], 'tickwidth': 1},
        'threshold': {
            'line': {'color': "red", 'width': 4},
            'thickness': 0.75,
            'value': 490}}
))

fig.show()
```

Maps

```
df_long = pd.read_csv('../input/india-power/long_data.csv')
df_long.dropna(inplace = True)
```

```
fig = px.scatter_geo(df_long,'latitude','longitude', color="Regions",
                     hover_name="States", size="Usage",
                     animation_frame="Dates", scope='asia')
fig.update_geos(lataxis_range=[5,35], lonaxis_range=[65, 100])
fig.show()
```