```
pip install bar_chart_race
```

```
Requirement already satisfied: bar_chart_race in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: matplotlib>=3.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/l
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
```

```
pip install https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
```

```
Collecting https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
  Using cached https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
Requirement already satisfied: joblib~=1.1.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3 in /usr/l
Requirement already satisfied: matplotlib>=3.2.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pydantic>=1.8.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: PyYAML>=5.0.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: jinja2>=2.11.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: markupsafe~=2.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: visions[type_image_path]==0.7.4 in /usr/local/lib/python
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: htmlmin>=0.1.12 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: missingno>=0.4.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: phik>=0.11.1 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: tangled-up-in-unicode==0.2.0 in /usr/local/lib/python3.7
Requirement already satisfied: requests>=2.24.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: tqdm>=4.48.2 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: seaborn>=0.10.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: multimethod>=1.4 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from v
Requirement already satisfied: imagehash in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/l
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/d
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.7/di
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.7/dist-packages (fr
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams['animation.embed_limit'] = 2**128
import numpy as np
import bar_chart_race as bcr
import plotly as py
import plotly.express as py
import plotly.graph_objs as go
from plotly.offline import iplot, init_notebook_mode
import datetime as date
import plotly.express as px
from datetime import datetime
power=pd.read_csv("/content/dataset_tk.csv")
long=pd.read_csv("/content/long_data_.csv")
power_copy=power.copy()
power.set_index('Date',inplace=True)


# Importing the given data about power consumption and various libraries.
```

```
power
# How our dataset looks?
```

Date	Punjab	Haryana	Rajasthan	Delhi	UP	Uttarakhand	HP	J&K	Chandigarh
1/1/2019	119.9	130.3	234.1	85.8	313.9	40.7	30.0	52.5	5.0
1/2/2019	121.9	133.5	240.2	85.5	311.8	39.3	30.1	54.1	4.9

```
print(power.shape)
# To check the number of rows and columns of dataset
```

```
(503, 33)
```

	...	...	...	...	...	...	...	...	...

```
power.describe()
# Descriptive Statistics for the dataset
```

	Punjab	Haryana	Rajasthan	Delhi	UP	Uttarakhand	HI
count	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000
mean	141.145527	138.333598	218.443340	83.380716	314.036382	36.157058	26.56819
std	56.977361	38.106593	27.421615	25.915357	66.516960	6.705108	4.80704(
min	56.100000	64.800000	105.800000	41.800000	186.800000	16.800000	11.80000(
25%	104.000000	114.800000	205.800000	63.500000	263.650000	33.800000	25.60000(
50%	118.300000	126.800000	222.900000	72.700000	290.000000	37.000000	28.00000(
75%	162.500000	158.100000	237.600000	105.800000	370.550000	40.350000	29.70000(
max	300.000000	237.200000	278.000000	147.100000	471.800000	53.200000	34.00000(

```
import pandas_profiling
from pandas_profiling import ProfileReport
```

```
profile = ProfileReport(power, title='Pandas Profiling Report', html={'style':{'full_width':F
# To compile a profile report for the given dataset
```

```
profile.to_notebook_iframe()
```

Summarize dataset: 1136/1136 [03:20<00:00, 2.73it/s,

```
profile.to_file(output_file="REPORT.html")
```

Export report to file: 100% 1/1 [00:00<00:00, 3.00it/s]

```
power.columns
```

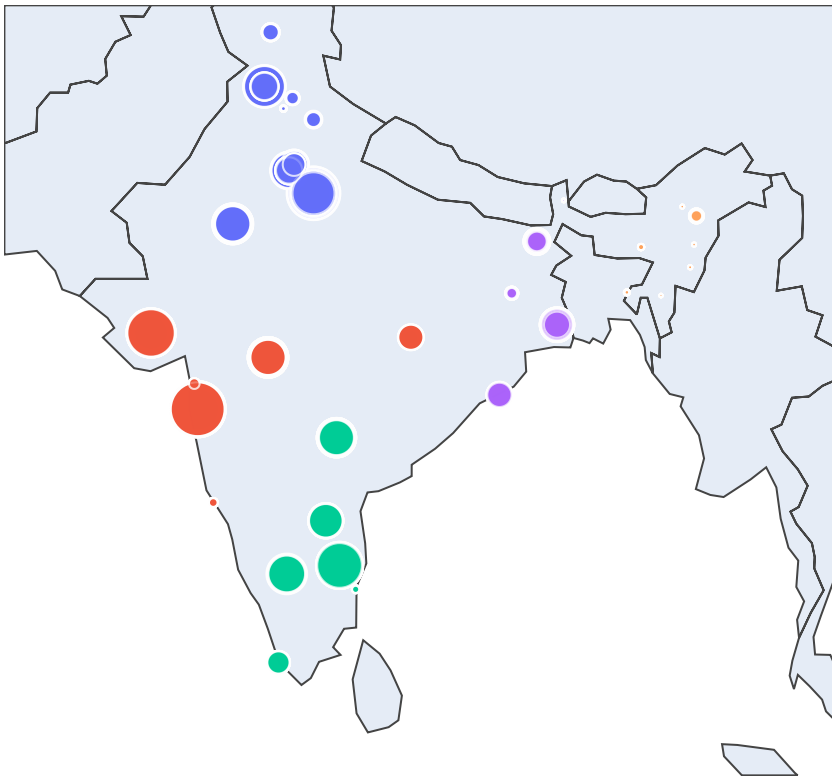
```
Index(['Punjab', 'Haryana', 'Rajasthan', 'Delhi', 'UP', 'Uttarakhand', 'HP',
       'J&K', 'Chandigarh', 'Chhattisgarh', 'Gujarat', 'MP', 'Maharashtra',
       'Goa', 'DNH', 'Andhra Pradesh', 'Telangana', 'Karnataka', 'Kerala',
       'Tamil Nadu', 'Pondy', 'Bihar', 'Jharkhand', 'Odisha', 'West Bengal',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

```
power['Total']=power.sum(axis=1)
power.loc['Total']=power.sum()
Total_comsumes=power.iloc[[-1]]
Total=Total_comsumes.transpose()
Total.drop(Total.index[[-1]], inplace=True)
Total.reset_index(inplace=True)
Total.rename(columns={'index':'State'}, inplace=True)
```

```
fig=go.Figure(go.Pie(labels=Total['State'],
                     values=Total['Total'], title='State Wise Electricity Consumption'))
iplot(fig)
# To display a pie chart diagram for state-wise comsumption during given time period.
```

#Observation-1)Maharashtra state consumed 12.7% of total power which is maximum by any state
2019 & 2020.
2)Gujarat state is second which consumed 9.5% of total power
3)Sikkim consumed the least power of all states ie 0.0379% of total power consumed.

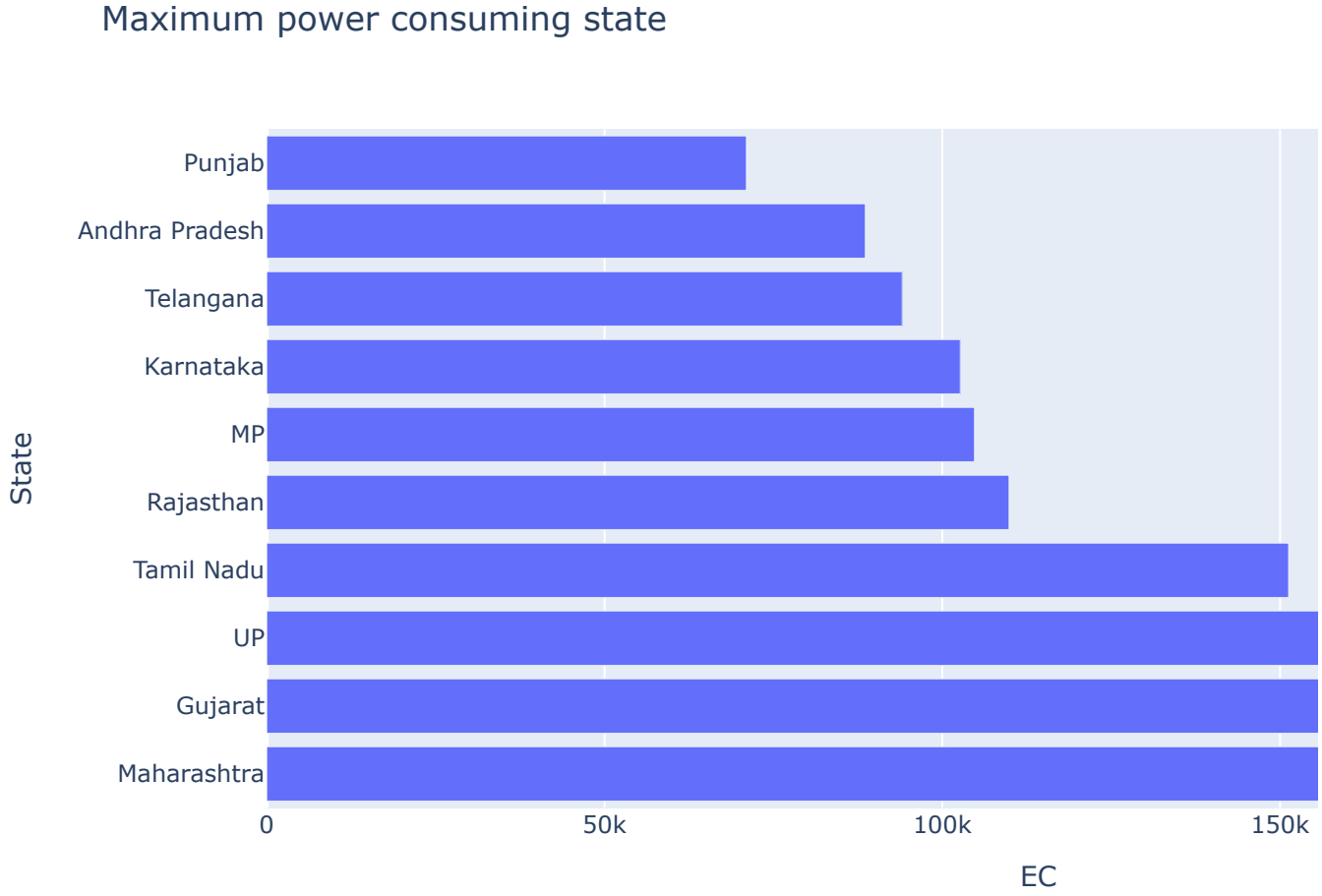
```
fig = px.scatter_geo(long,'latitude','longitude', color="Regions",
                     hover_name="States", size="Usage",
                      scope='asia')
fig.update_geos(lataxis_range=[5,35], lonaxis_range=[65, 100])
fig.show()
```



```
# Observation - Scatter-geo function is used to represent the usage of electricity across dif
  geographies of the country.
  Each colour represent various regions of our country while the spread/density represents th
  UP has consumed the highest electricity in the Northern region
  Maharashtra has consumed the highest electricity in the Western region
  Tamil Nadu has consumed the highest electricity in the  Southern region
  West Bengal has consumed the highest electricity in the Eastern region
  Assam has consumed the highest electricity in the North Eastern region


Top_10_EC_Consume=Total.nlargest(10, 'Total')
Top_10_EC_Consume
# To find the top 10 highest electricity consuming states.
```

	Date	State	Total
12		Maharashtra	217079.8
10		Gujarat	162488.9
4		UP	157960.3
19		Tamil Nadu	151271.5
2		Rajasthan	109877.0
11		MP	104766.4
17		Karnataka	102665.7
16		Telangana	94065.3
15		Andhra Pradesh	88604.4
0		Punjab	70996.2

```
Low_EC_Consumption=Total.nsmallest(10, 'Total')
# To find the 10 least electricity consuming states


layout = go.Layout(title = "Maximum power consuming state", xaxis = {'title':'EC'}, yaxis = {
cons=go.Bar(x=Top_10_EC_Consume['Total'], y=Top_10_EC_Consume['State'], orientation='h' )
fig=go.Figure(data=cons, layout=layout)
iplot(fig)
print(Top_10_EC_Consume)
```

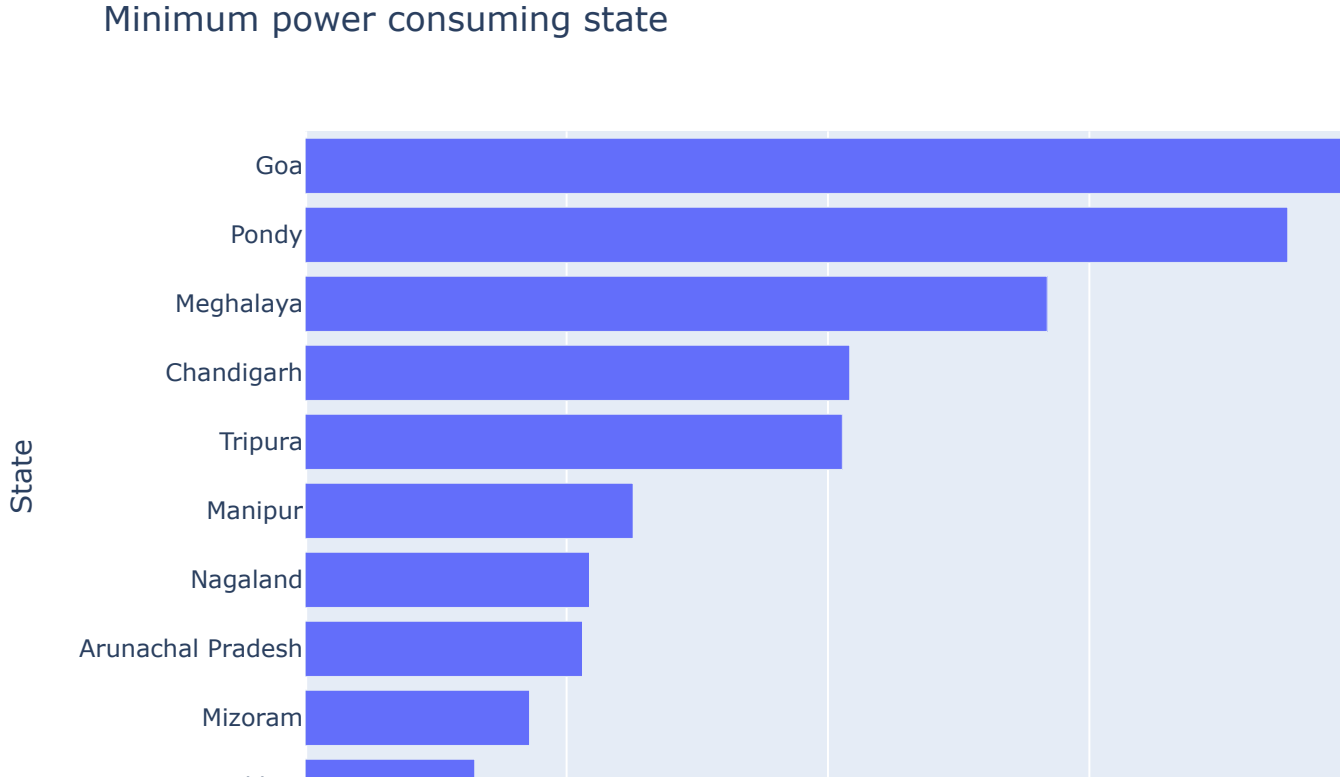


	Date	State	Total
12		Maharashtra	217079.8
10		Gujarat	162488.9
4		UP	157960.3
19		Tamil Nadu	151271.5
2		Rajasthan	109877.0

```
# Observation - This is the rankwise representation of top 10 states who have consumed the hi
  electricity
  In the two year period. Maharashtra with 217079.8 units has consumed highest electricity am
```

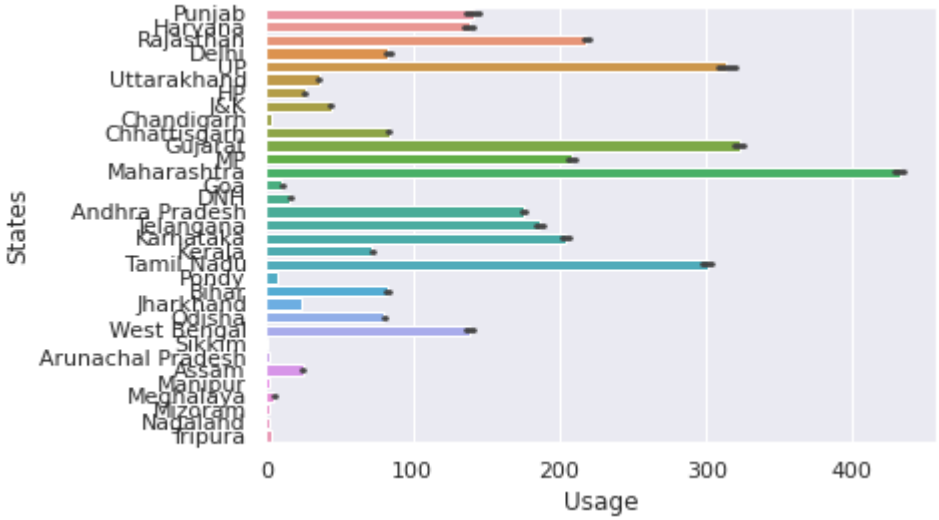
0		Punjab	70996.2

```
Low_EC_Consumption=Total.nsmallest(10, 'Total')
layout = go.Layout(title = "Minimum power consuming state", xaxis = {'title':'EC'}, yaxis = {
d=go.Bar(x=Low_EC_Consumption['Total'],y=Low_EC_Consumption['State'], orientation='h')
fig=go.Figure(data=d, layout=layout)
iplot(fig)
```




```
# Observation - This is the rankwise representation of states who have consumed the least ele
  in the two year period.
  Sikkim has consumed least amount of electricity across two years among all the sates in Inc
```

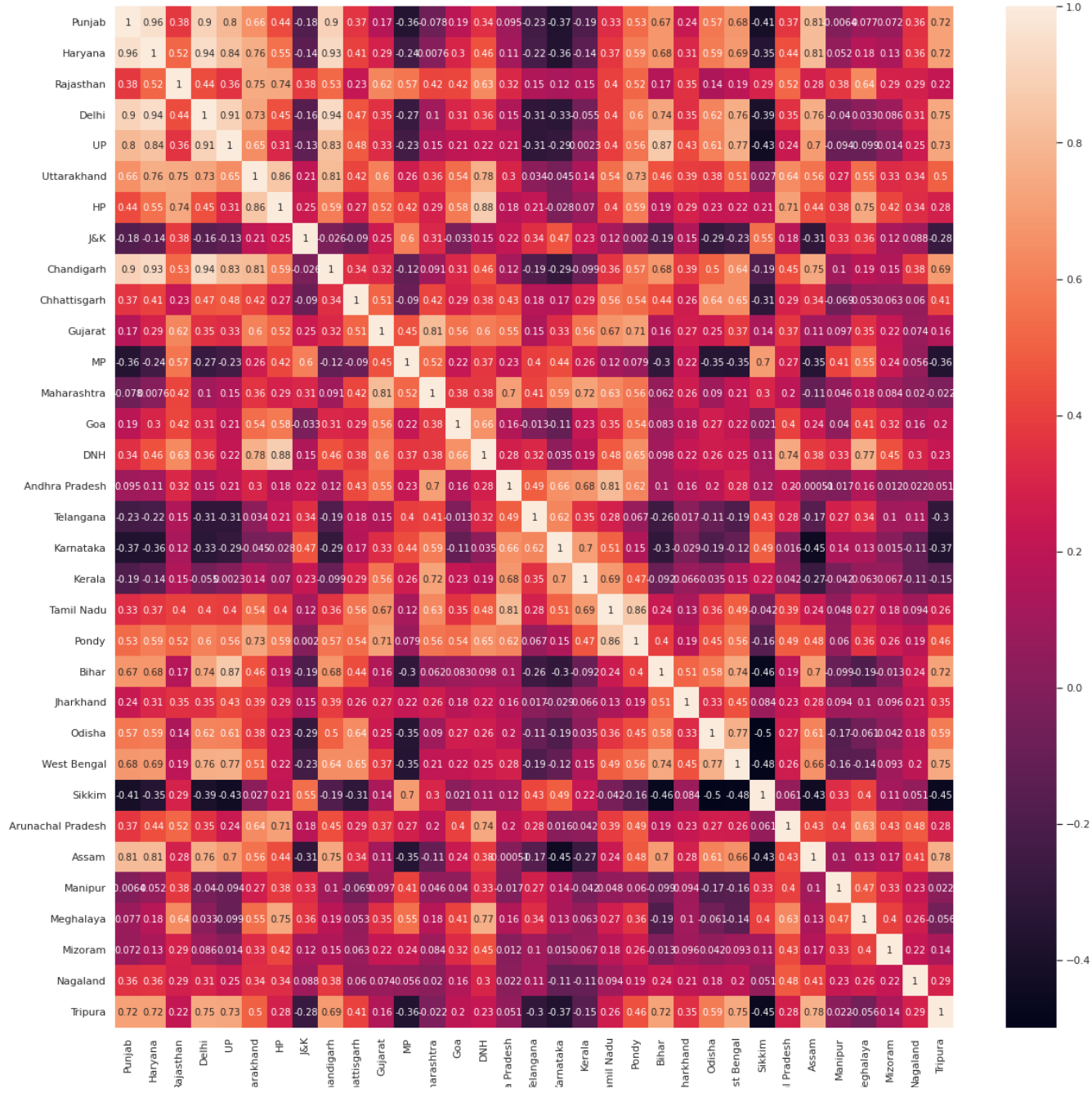
```
sns.barplot(x="Usage", y="States", data=long)
sns.set(rc={'figure.figsize':(50,8.27)})
# Plotting a horizontal bar chart
```




```
# Observation - This is the graphical representation of the electricity consumed by all the s
  given time period and helps in state wise comparison.
```

```
ax=sns.relplot(x="Regions",y="Usage", data=long, kind="line", markers=True)
# Plotting a line graph for region-wise consumption.
```


```
# Observation - It is observed that most units of electricity have been consumed by Western r
  region, northern region, eastern region and north east respectively.


plt.figure(figsize=(20,20))
sns.heatmap(power.corr(), annot= True)
plt.show()
# Plotting a heat map showing correlation
```




```
bcr.bar_chart_race(power, figsize=(7, 4),period_length =500,filename = None, title='power usa
# Generating video representing running bar chart of state wise consumption.
```

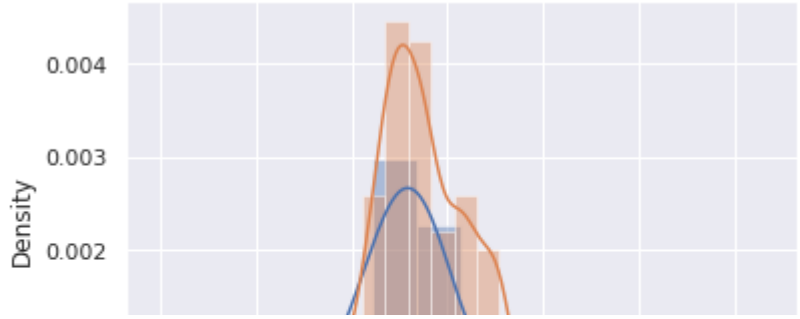
Double-click (or enter) to edit

```
power['NR'] = power['Punjab']+ power['Haryana']+ power['Rajasthan']+ power['Delhi']+power['UP
power['WR'] = power['Chhattisgarh']+power['Gujarat']+power['MP']+power['Maharashtra']+power['
power['SR'] = power['Andhra Pradesh']+power['Telangana']+power['Karnataka']+power['Kerala']+p
power['ER'] = power['Bihar']+power['Jharkhand']+ power['Odisha']+power['West Bengal']+power['
power['NER'] =power['Arunachal Pradesh']+power['Assam']+power['Manipur']+power['Meghalaya']+p
# Alloting states to their respective regions of the country
```

```
df_new = pd.DataFrame({"Northern Region": power["NR"].values,
                       "Southern Region": power["SR"].values,
                       "Eastern Region": power["ER"].values,
                       "Western Region": power["WR"].values,
                       "North Eastern Region": power["NER"].values},index=power.index)
```

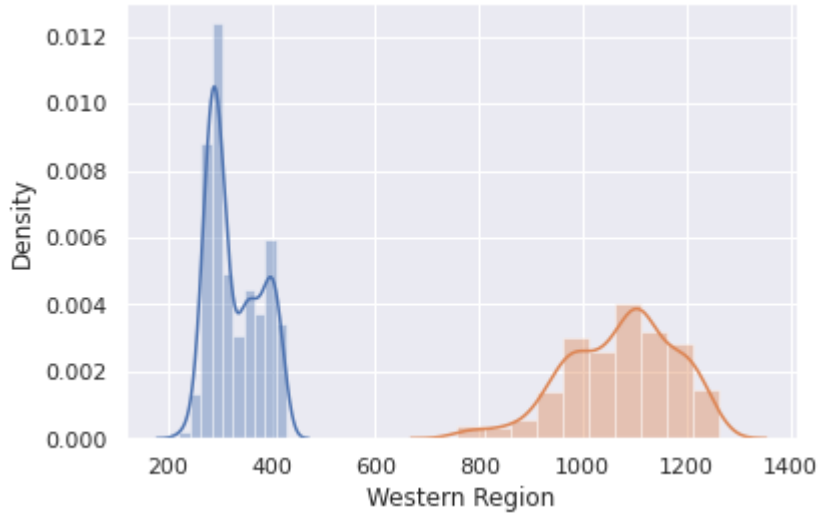
```
sns.distplot(df_new["Northern Region"], bins=10)
sns.distplot(df_new["Southern Region"], bins=10)
# Distribution to show comparison of month wise consumption of northern and southern region
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please ada

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please ada

<matplotlib.axes._subplots.AxesSubplot at 0x7f49a710d050>
```




```
sns.distplot(df_new["Eastern Region"], bins=10)
sns.distplot(df_new["Western Region"], bins=10)
 #Distribution to show comparison of month wise consumption of western and eastern region
```

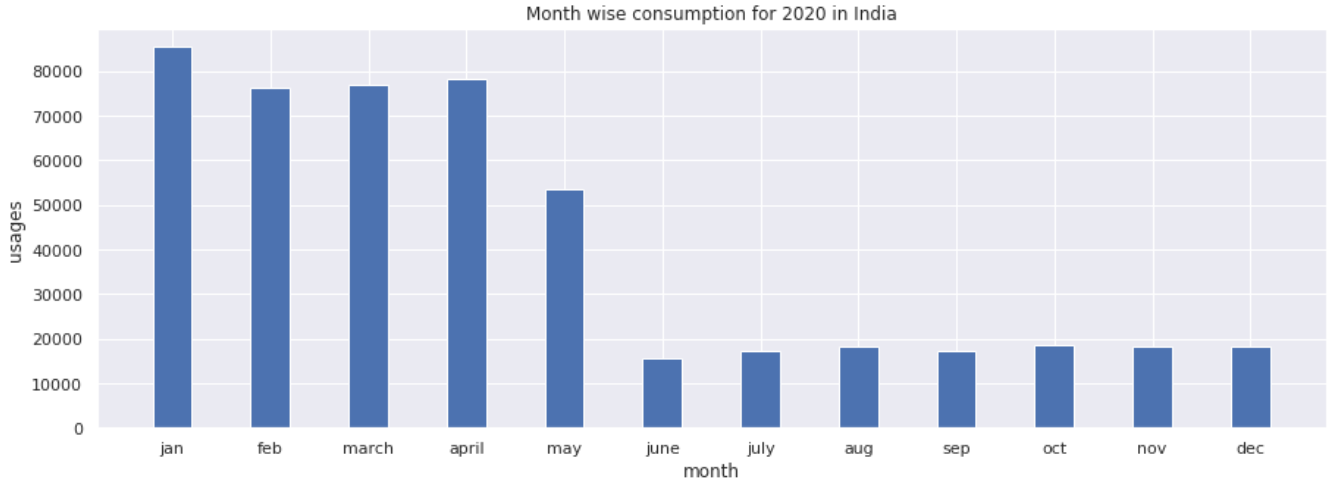
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please ada

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please ada

<matplotlib.axes._subplots.AxesSubplot at 0x7f49a69b2950>
```




```
data=pd.read_csv("/content/dataset_tk1.csv")
data.rename(columns={"Unnamed: 0":"Date"},inplace=True)
data['Date']=pd.to_datetime(data["Date"],dayfirst=True)
```

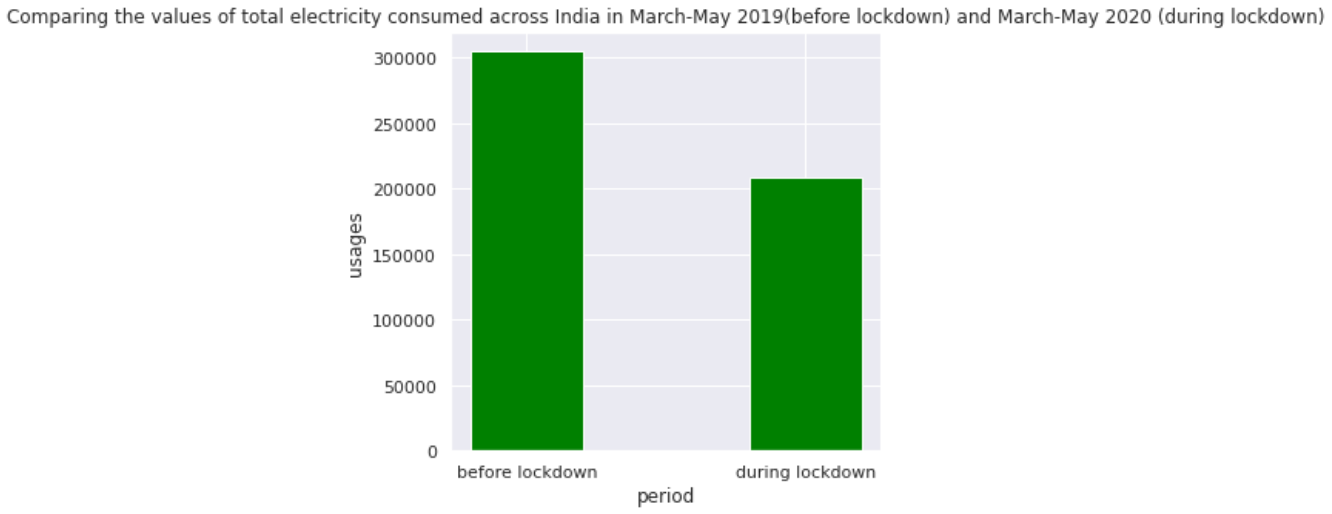
```
data["year"]=data["Date"].dt.year
data["month"]=data["Date"].dt.month
data["day"]=data["Date"].dt.day
data.drop(["Date"],axis=1,inplace=True)
# Indexing for date column



long=pd.read_csv("/content/long_data_.csv")
long['Dates']=pd.to_datetime(long["Dates"],dayfirst=True)
long["year"]=long["Dates"].dt.year
long["month"]=long["Dates"].dt.month
long["day"]=long["Dates"].dt.day
long.drop(["latitude","longitude","Dates"],axis=1,inplace=True)


dict2019=dict()
for i in range(1,13):
    m=long[(long["month"]==i)&(long["year"]==2019)]
    n=m["Usage"].sum()
    dict2019.update({i:n})
dict2020=dict()
for i in range(1,13):
    m=long[(long["month"]==i)&(long["year"]==2020)]
    n=m["Usage"].sum()
    dict2020.update({i:n})



month=["jan","feb",'march',"april","may","june","july","aug","sep","oct","nov","dec"]


list1=list(dict2019.values())
list2=list(dict2020.values())


plt.figure(figsize = (15, 5))
plt.bar(month,list1,width = 0.4)

plt.ylabel("usages")
plt.xlabel("month")
plt.title("Month wise consumption for 2019 in India")
plt.show()
# Plotting bar graph for month wise consumption across India in 2019
```


```
plt.figure(figsize = (15, 5))
plt.bar(month,list2,width = 0.4)

plt.ylabel("usages")
plt.xlabel("month")
plt.title("Month wise consumption for 2020 in India")
plt.show()
# Plotting bar graph for month wise consumption across India in 2020
```




```
# Observation - Studying the graphs of month-wise consumption of electricity in 2019 & 2020,
its observed that the consumption level dropped significantly from April 2020.
  -The overall electricity consumption in the country decreased during the initial period of
  with dramatic reductions in consumption of services and industry only partially offset by
  higher residential use.


num1=float((list1[2]+list1[3]+list1[4]))
num2=float((list2[2]+list2[3]+list2[4]))
list3=[]
list4=[]
```

```
list4.append(num1)
list4.append(num2)
list3.append('before lockdown')
list3.append('during lockdown')

plt.figure(figsize = (5, 5))
plt.bar(list3, list4, color ='green',
        width = 0.4)

plt.xlabel("period")
plt.ylabel("usages")
plt.title("Comparing the values of total electricity consumed across India in March-May 2019(
plt.show()
```




```
# Observation - The comparison of consumption patterns in the period March-May for 2019 and 2
  approximately 31.47% reduction in total consumption across India.
```

✓ 0s completed at 6:49 PM