Covid19 Data Analysis Project using Python

In [41]:
```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
%matplotlib inline

import plotly
import plotly.express as px
import plotly.graph_objects as go #to use plotly in an object-oriented way
#plt.rcParams['figure.figsize']=20,12 #used to increase the size of the plot u
sing pandas
import cufflinks as cf #it binds plotly with pandas together
import plotly.offline as pyo
from plotly.offline import init_notebook_mode,plot,iplot

import folium # used for maps
```

In [2]:
```
# to connect plotly with jupyter notebook
pyo.init_notebook_mode(connected=True)
cf.go_offline()
```

In [3]:
```
df=pd.read_excel(r"E:\Data\covid19ds_project\Covid cases in India.xlsx")
df
```

Out[3]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	1	Andhra Pradesh	12	0	1	0
1	2	Chhattisgarh	6	0	0	0
2	3	Delhi	38	1	6	1
3	4	Gujarat	43	0	0	3
4	5	Haryana	16	14	11	0
5	6	Himachal Pradesh	4	0	0	1
6	7	Karnataka	20	0	3	2
7	8	Kerala	131	7	11	0
8	9	Madhya Pradesh	23	0	0	1
9	10	Maharashtra	144	3	15	4
10	11	Odisha	3	0	0	0
11	12	Puducherry	1	0	0	0
12	13	Punjab	29	0	0	1
13	14	Rajasthan	41	2	3	0
14	15	Tamil Nadu	32	3	1	1
15	16	Telengana	34	11	1	0
16	17	Chandigarh	7	0	0	0
17	18	Jammu and Kashmir	18	0	1	1
18	19	Ladakh	13	0	0	0
19	20	Uttar Pradesh	42	1	11	0
20	21	Uttarakhand	4	0	0	0
21	22	West Bengal	11	0	0	1
22	23	Bihar	7	0	0	1
23	24	Mizoram	1	0	0	0
24	25	Goa	6	0	0	0
25	26	Manipur	1	0	0	0

In [4]:
```
df.drop(columns={'S. No.'},axis=1,inplace=True)
```

In [5]:
```
df
```

Out[5]:

	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	Andhra Pradesh	12	0	1	0
1	Chhattisgarh	6	0	0	0
2	Delhi	38	1	6	1
3	Gujarat	43	0	0	3
4	Haryana	16	14	11	0
5	Himachal Pradesh	4	0	0	1
6	Karnataka	20	0	3	2
7	Kerala	131	7	11	0
8	Madhya Pradesh	23	0	0	1
9	Maharashtra	144	3	15	4
10	Odisha	3	0	0	0
11	Puducherry	1	0	0	0
12	Punjab	29	0	0	1
13	Rajasthan	41	2	3	0
14	Tamil Nadu	32	3	1	1
15	Telengana	34	11	1	0
16	Chandigarh	7	0	0	0
17	Jammu and Kashmir	18	0	1	1
18	Ladakh	13	0	0	0
19	Uttar Pradesh	42	1	11	0
20	Uttarakhand	4	0	0	0
21	West Bengal	11	0	0	1
22	Bihar	7	0	0	1
23	Mizoram	1	0	0	0
24	Goa	6	0	0	0
25	Manipur	1	0	0	0

In [6]:
```
# creating another column including the indian national and foreign national c
itizens in India
df['Total cases'] = df['Total Confirmed cases (Indian National)'] + df['Total
 Confirmed cases ( Foreign National )']
```

In [7]:
```
df
```

Out[7]:

	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total cases
0	Andhra Pradesh	12	0	1	0	12
1	Chhattisgarh	6	0	0	0	6
2	Delhi	38	1	6	1	39
3	Gujarat	43	0	0	3	43
4	Haryana	16	14	11	0	30
5	Himachal Pradesh	4	0	0	1	4
6	Karnataka	20	0	3	2	20
7	Kerala	131	7	11	0	138
8	Madhya Pradesh	23	0	0	1	23
9	Maharashtra	144	3	15	4	147
10	Odisha	3	0	0	0	3
11	Puducherry	1	0	0	0	1
12	Punjab	29	0	0	1	29
13	Rajasthan	41	2	3	0	43
14	Tamil Nadu	32	3	1	1	35
15	Telengana	34	11	1	0	45
16	Chandigarh	7	0	0	0	7
17	Jammu and Kashmir	18	0	1	1	18
18	Ladakh	13	0	0	0	13
19	Uttar Pradesh	42	1	11	0	43
20	Uttarakhand	4	0	0	0	4
21	West Bengal	11	0	0	1	11
22	Bihar	7	0	0	1	7
23	Mizoram	1	0	0	0	1
24	Goa	6	0	0	0	6
25	Manipur	1	0	0	0	1

In [8]:
```
# total no. of cases in india
total_cases_overall=df['Total cases'].sum()
print("Total no. of cases in India:",total_cases_overall)
```

```
Total no. of cases in India: 729
```

In [9]:
```
df['Total active cases']= df['Total cases']-df['Cured'] - df['Death']
df
```

Out[9]:

	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total cases	Total active cases
0	Andhra Pradesh	12	0	1	0	12	11
1	Chhattisgarh	6	0	0	0	6	6
2	Delhi	38	1	6	1	39	32
3	Gujarat	43	0	0	3	43	40
4	Haryana	16	14	11	0	30	19
5	Himachal Pradesh	4	0	0	1	4	3
6	Karnataka	20	0	3	2	20	15
7	Kerala	131	7	11	0	138	127
8	Madhya Pradesh	23	0	0	1	23	22
9	Maharashtra	144	3	15	4	147	128
10	Odisha	3	0	0	0	3	3
11	Puducherry	1	0	0	0	1	1
12	Punjab	29	0	0	1	29	28
13	Rajasthan	41	2	3	0	43	40
14	Tamil Nadu	32	3	1	1	35	33
15	Telengana	34	11	1	0	45	44
16	Chandigarh	7	0	0	0	7	7
17	Jammu and Kashmir	18	0	1	1	18	16
18	Ladakh	13	0	0	0	13	13
19	Uttar Pradesh	42	1	11	0	43	32
20	Uttarakhand	4	0	0	0	4	4
21	West Bengal	11	0	0	1	11	10
22	Bihar	7	0	0	1	7	6
23	Mizoram	1	0	0	0	1	1
24	Goa	6	0	0	0	6	6
25	Manipur	1	0	0	0	1	1

In [10]:
```
total_active_cases_overall = df['Total active cases'].sum()
print("Total no. of active cases in India:",total_active_cases_overall)
```

```
Total no. of active cases in India: 648
```

In [11]:
```
#displaying the numerical data in a more presentable form
df.style.background_gradient(cmap="Reds")
```

Out[11]:

	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total cases	Total active cases
0	Andhra Pradesh	12	0	1	0	12	11
1	Chhattisgarh	6	0	0	0	6	6
2	Delhi	38	1	6	1	39	32
3	Gujarat	43	0	0	3	43	40
4	Haryana	16	14	11	0	30	19
5	Himachal Pradesh	4	0	0	1	4	3
6	Karnataka	20	0	3	2	20	15
7	Kerala	131	7	11	0	138	127
8	Madhya Pradesh	23	0	0	1	23	22
9	Maharashtra	144	3	15	4	147	128
10	Odisha	3	0	0	0	3	3
11	Puducherry	1	0	0	0	1	1
12	Punjab	29	0	0	1	29	28
13	Rajasthan	41	2	3	0	43	40
14	Tamil Nadu	32	3	1	1	35	33
15	Telengana	34	11	1	0	45	44
16	Chandigarh	7	0	0	0	7	7
17	Jammu and Kashmir	18	0	1	1	18	16
18	Ladakh	13	0	0	0	13	13
19	Uttar Pradesh	42	1	11	0	43	32
20	Uttarakhand	4	0	0	0	4	4
21	West Bengal	11	0	0	1	11	10
22	Bihar	7	0	0	1	7	6
23	Mizoram	1	0	0	0	1	1
24	Goa	6	0	0	0	6	6
25	Manipur	1	0	0	0	1	1

In [12]:
```
Total_Active_cases= df.groupby('Name of State / UT')['Total active cases'].sum
().sort_values(ascending=False).to_frame()
```

In [13]:
```
Total_Active_cases
```

Out[13]:

Name of State / UT	Total active cases
Maharashtra	128
Kerala	127
Telengana	44
Rajasthan	40
Gujarat	40
Tamil Nadu	33
Uttar Pradesh	32
Delhi	32
Punjab	28
Madhya Pradesh	22
Haryana	19
Jammu and Kashmir	16
Karnataka	15
Ladakh	13
Andhra Pradesh	11
West Bengal	10
Chandigarh	7
Goa	6
Chhattisgarh	6
Bihar	6
Uttarakhand	4
Himachal Pradesh	3
Odisha	3
Manipur	1
Mizoram	1
Puducherry	1

In [14]:
```
Total_Active_cases.style.background_gradient(cmap='Reds')
```

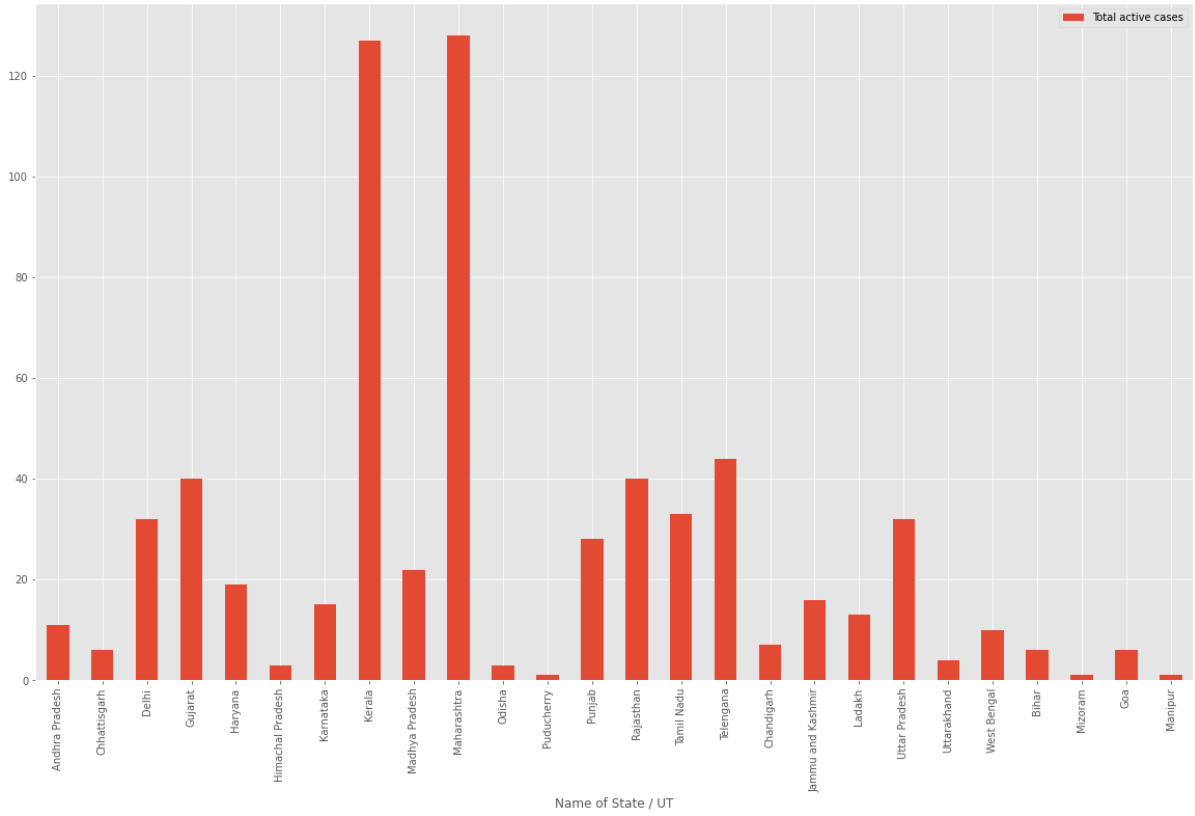
Out[14]:

Name of State / UT	Total active cases
Maharashtra	128
Kerala	127
Telengana	44
Rajasthan	40
Gujarat	40
Tamil Nadu	33
Uttar Pradesh	32
Delhi	32
Punjab	28
Madhya Pradesh	22
Haryana	19
Jammu and Kashmir	16
Karnataka	15
Ladakh	13
Andhra Pradesh	11
West Bengal	10
Chandigarh	7
Goa	6
Chhattisgarh	6
Bihar	6
Uttarakhand	4
Himachal Pradesh	3
Odisha	3
Manipur	1
Mizoram	1
Puducherry	1

Graphical Representation of Data

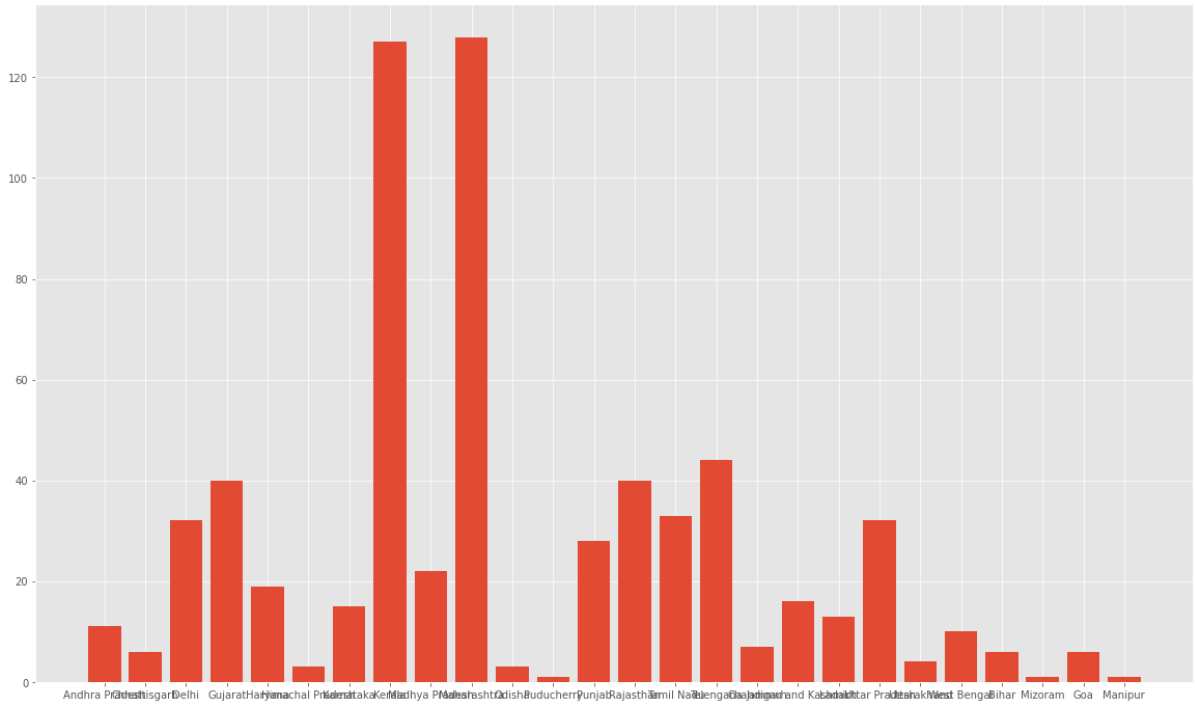
In [17]:
```
# 1. Pandas built in function for data visualization
df.plot(kind='bar',x='Name of State / UT',y='Total active cases',)
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x14fe5f10>

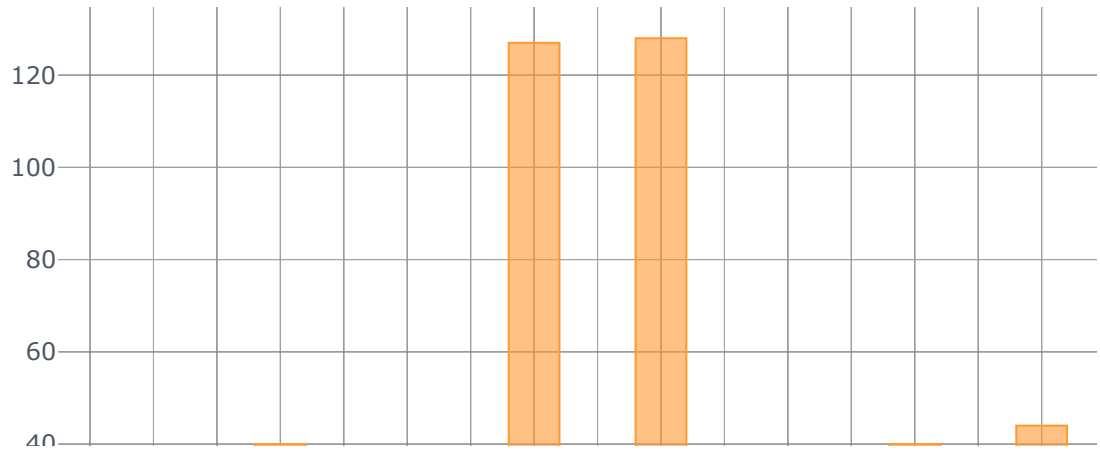


In [18]:
```
# 2. matplotlib visualization
plt.bar(df['Name of State / UT'],df['Total active cases'])
```

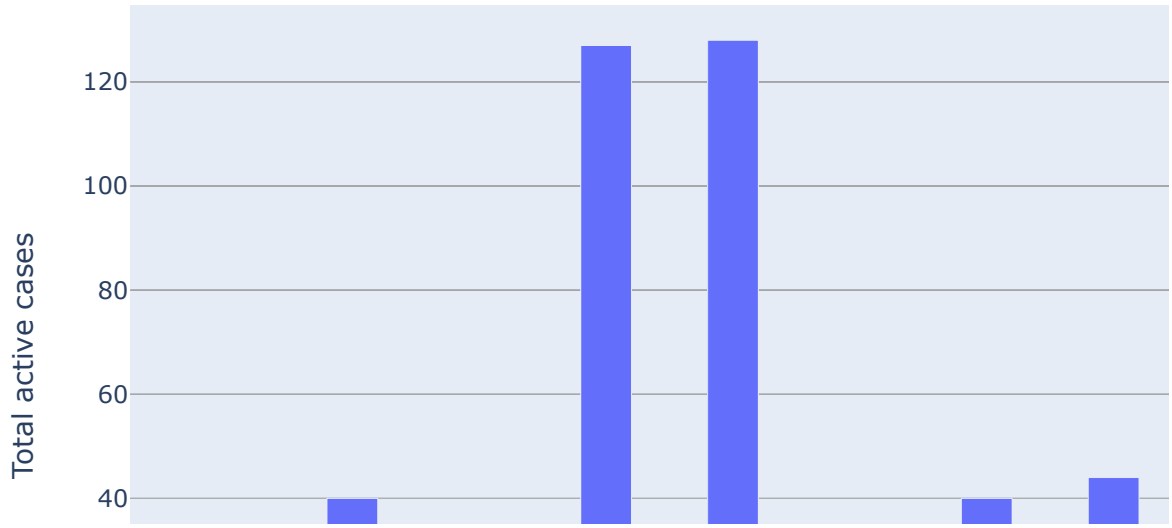
Out[18]: <BarContainer object of 26 artists>



In [20]:
```
# 3.using plotly
df.iplot(kind='bar',x='Name of State / UT',y='Total active cases')
```

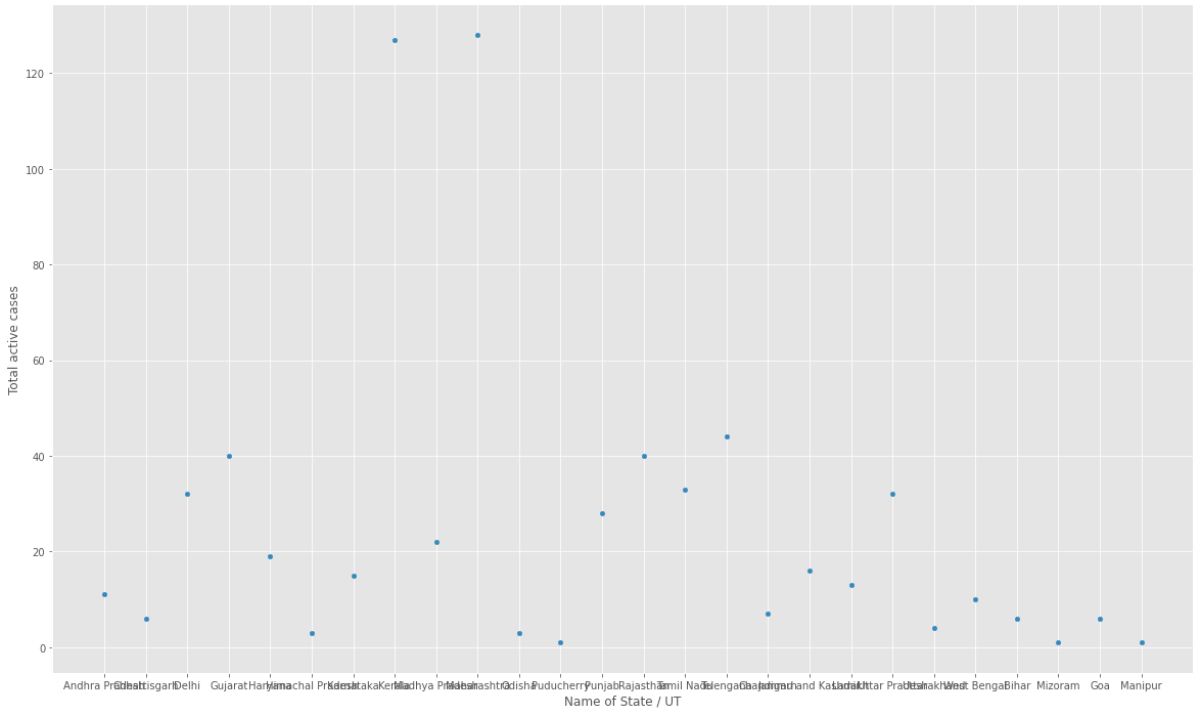


In [21]:
```
# 3. using plotly.express
px.bar(df,x='Name of State / UT',y='Total active cases')
```



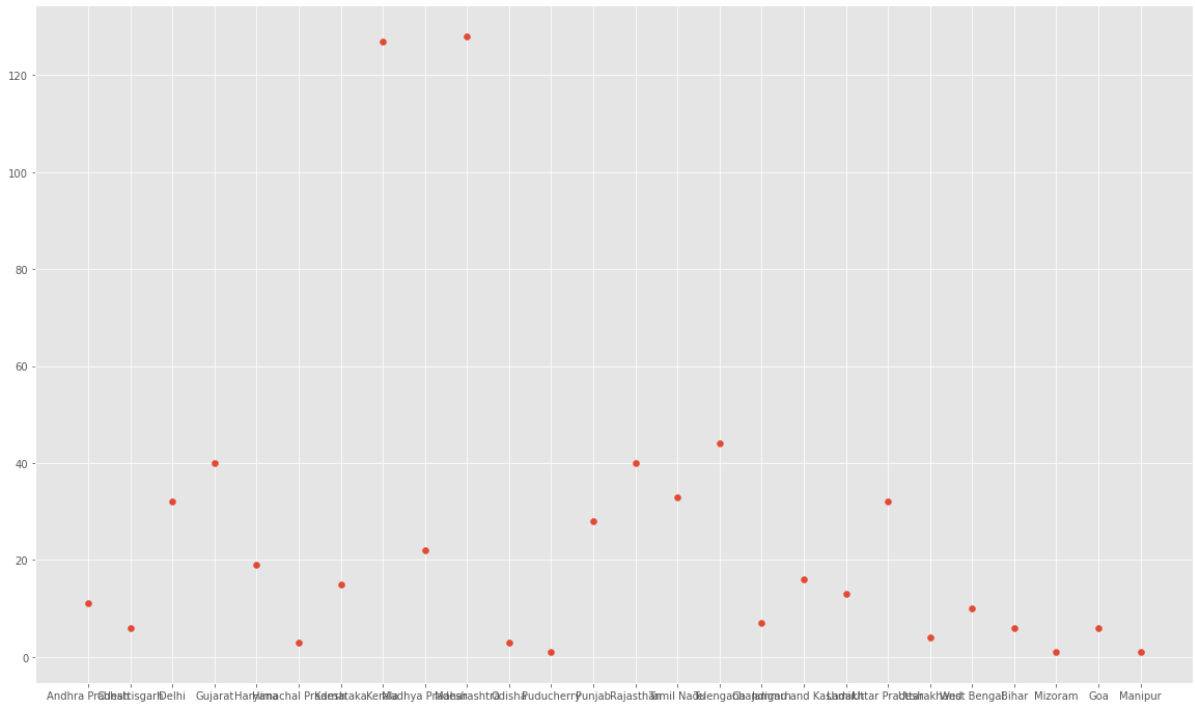
In [22]:
```
# Scatter plot using pandas
df.plot(kind='scatter',x='Name of State / UT',y='Total active cases')
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x17301238>

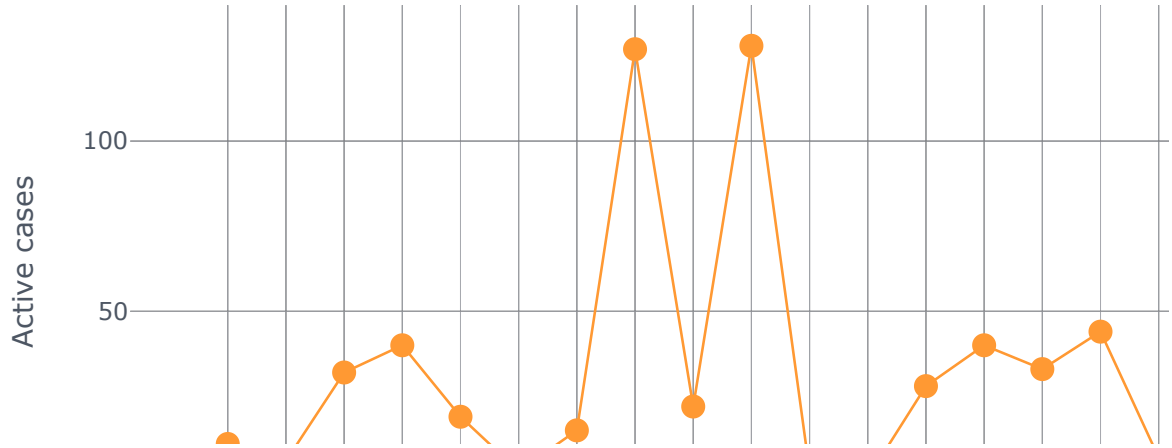


In [30]:
```
# 2. Scatter plot using matplotlib
plt.scatter(df['Name of State / UT'],df['Total active cases'])
```

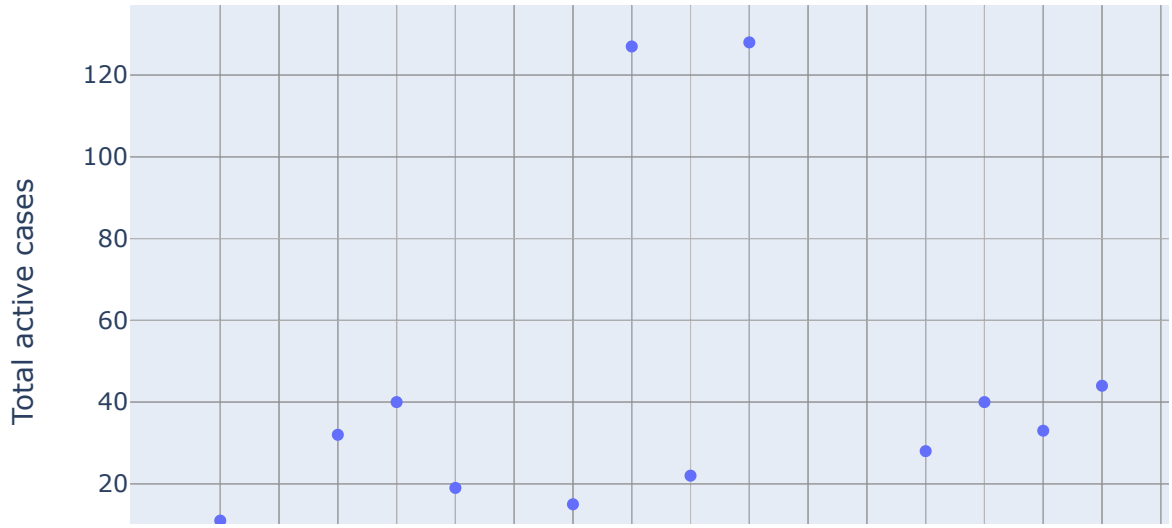
Out[30]: <matplotlib.collections.PathCollection at 0x1839e8c8>



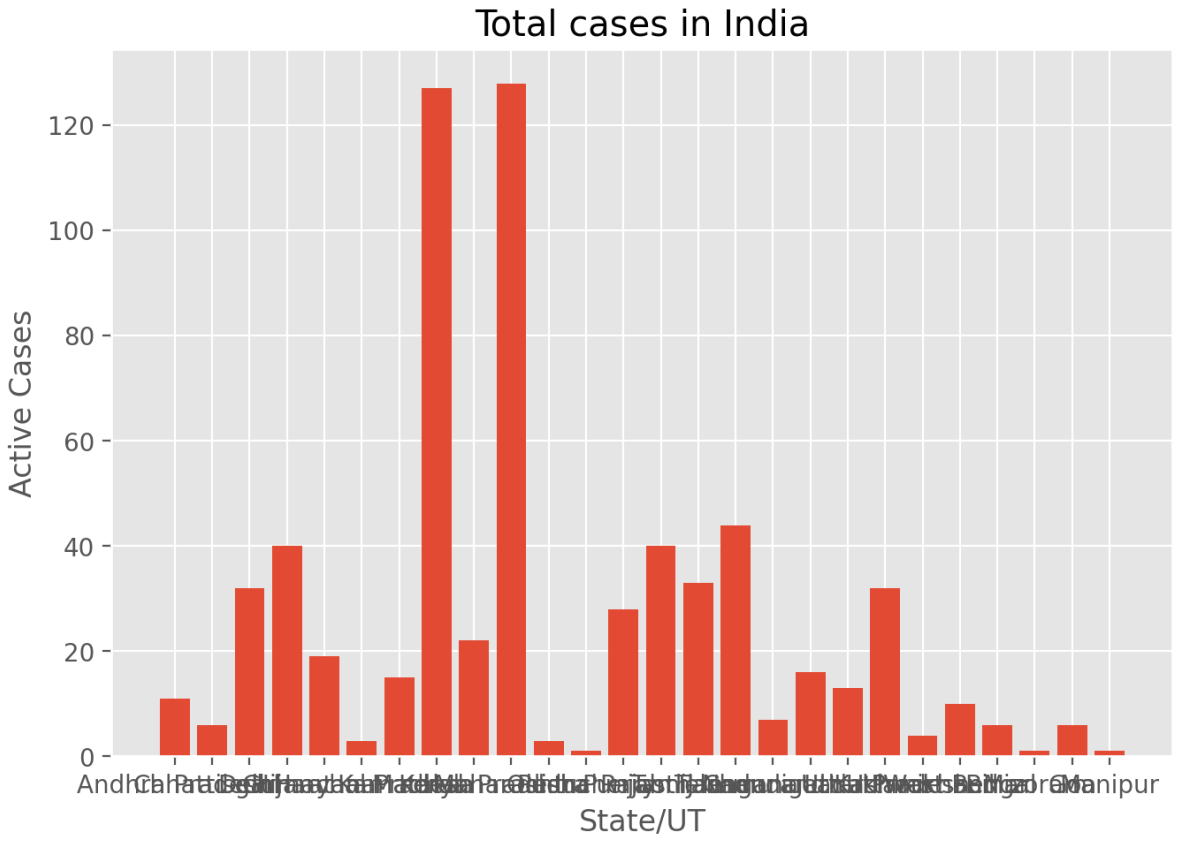
In [25]:
```
# 3. using plotly
df.iplot(kind='scatter',x='Name of State / UT',y='Total active cases',xTitle=
'State/UT',yTitle='Active cases',mode='markers+lines')
```

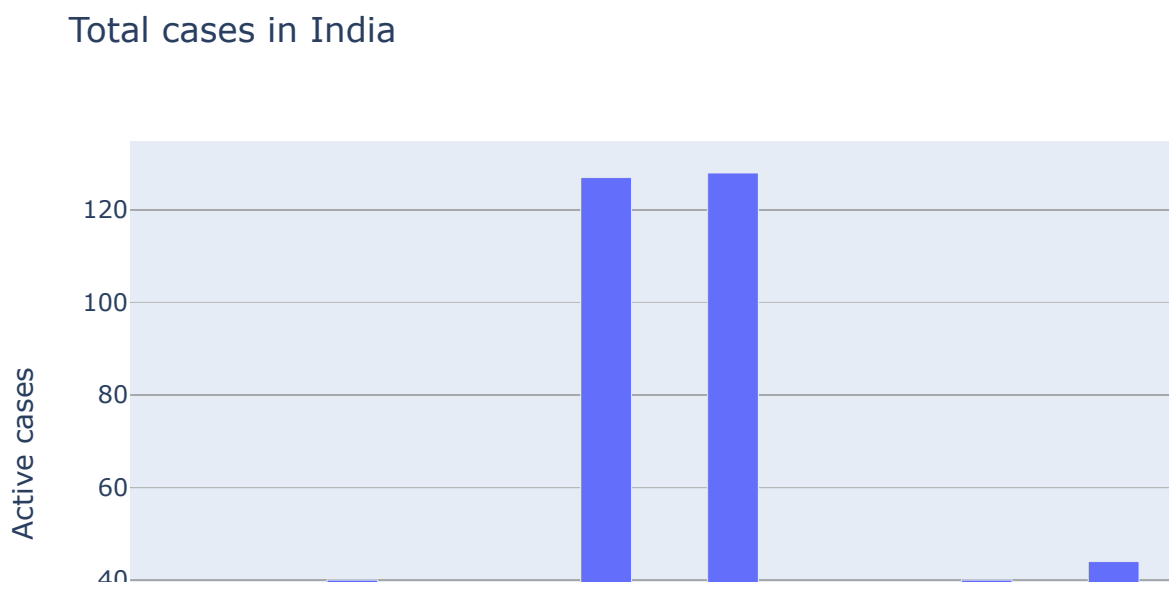


In [34]:
```
# 4. using plotly express
px.scatter(df,x='Name of State / UT',y='Total active cases')
```



In [47]:
```
# object oriented visualization for matplotlib
fig=plt.figure(dpi=200)
axes=fig.add_axes([0,0,1,1])
axes.bar(df['Name of State / UT'],df['Total active cases'])
axes.set_title("Total cases in India")
axes.set_xlabel("State/UT")
axes.set_ylabel("Active Cases")
plt.show()

# plotly
fig=go.Figure()
fig.add_trace(go.Bar(x=df['Name of State / UT'],y=df['Total active cases']))
fig.update_layout(title='Total cases in India',xaxis=dict(title='State/UT'),ya
xis=dict(title='Active cases'))
```


Total cases in India
Active Cases
0
20
40
60
80
100
120
State/UT
Manipur


Total cases in India
Active cases
40
60
80
100
120

In [48]:
```
coordinates=pd.read_excel(r"E:\Data\covid19ds_project\Indian Coordinates.xlsx"
)
```

In [49]:
```
coordinates
```

Out[49]:

	Name of State / UT	Latitude	Longitude
0	Andaman And Nicobar	11.667026	92.735983
1	Andhra Pradesh	14.750429	78.570026
2	Arunachal Pradesh	27.100399	93.616601
3	Assam	26.749981	94.216667
4	Bihar	25.785414	87.479973
5	Chandigarh	30.719997	76.780006
6	Chhattisgarh	22.090420	82.159987
7	Dadra And Nagar Haveli	20.266578	73.016618
8	Delhi	28.669993	77.230004
9	Goa	15.491997	73.818001
10	Haryana	28.450006	77.019991
11	Himachal Pradesh	31.100025	77.166597
12	Union Territory of Jammu and Kashmir	33.450000	76.240000
13	Jharkhand	23.800393	86.419986
14	Karnataka	12.570381	76.919997
15	Kerala	8.900373	76.569993
16	Lakshadweep	10.562573	72.636867
17	Madhya Pradesh	21.300391	76.130019
18	Maharashtra	19.250232	73.160175
19	Manipur	24.799971	93.950017
20	Meghalaya	25.570492	91.880014
21	Mizoram	23.710399	92.720015
22	Nagaland	25.666998	94.116570
23	Orissa	19.820430	85.900017
24	Puducherry	11.934994	79.830000
25	Punjab	31.519974	75.980003
26	Rajasthan	26.449999	74.639981
27	Sikkim	27.333330	88.616647
28	Telengana	18.112400	79.019300
29	Tamil Nadu	12.920386	79.150042
30	Tripura	23.835404	91.279999
31	Uttar Pradesh	27.599981	78.050006
32	Uttarakhand	30.320409	78.050006
33	West Bengal	22.580390	88.329947
34	Union Territory of Ladakh	34.100000	77.340000

In [51]:
```
df_full= pd.merge(coordinates,df,on='Name of State / UT')
df_full
```

Out[51]:

	Name of State / UT	Latitude	Longitude	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total cases	Total active cases
0	Andhra Pradesh	14.750429	78.570026	12	0	1	0	12	11
1	Delhi	28.669993	77.230004	38	1	6	1	39	32
2	Haryana	28.450006	77.019991	16	14	11	0	30	19
3	Karnataka	12.570381	76.919997	20	0	3	2	20	15
4	Kerala	8.900373	76.569993	131	7	11	0	138	127
5	Maharashtra	19.250232	73.160175	144	3	15	4	147	128
6	Punjab	31.519974	75.980003	29	0	0	1	29	28
7	Rajasthan	26.449999	74.639981	41	2	3	0	43	40
8	Telengana	18.112400	79.019300	34	11	1	0	45	44
9	Tamil Nadu	12.920386	79.150042	32	3	1	1	35	33
10	Uttar Pradesh	27.599981	78.050006	42	1	11	0	43	32
11	Uttarakhand	30.320409	78.050006	4	0	0	0	4	4

In [65]:
```
# using folium for maps
map=folium.Map(locations=[20,70],zoom_start=64,tieles='Stamenterrain')

for lat,long,value,name in zip(df_full['Latitude'],df_full['Longitude'],df_full['Total cases'],df_full['Name of State / UT']):
    folium.CircleMarker([lat,long],radius=value*0.8,popup=('<strong>State</strong>: '+str(name).capitalize()+ '<br>''<strong>Total Cases</strong>: ' + str(value) +'<br>'),color='red',fill_color='red',fill_opacity=0.3).add_to(map)
```

In [66]:
```
map
```

Out[66]:

The rise of Coronavirus(Covid19) globally

In [68]:
```
pdc_India=pd.read_excel(r"E:\Data\covid19ds_project\per_day_cases.xlsx",parse_
date=True,sheet_name="India")
pdc_India
```

Out[68]:

	Date	Total Cases	New Cases	Days after surpassing 100 cases
0	2020-01-30	1	1	NaN
1	2020-01-31	1	0	NaN
2	2020-02-01	1	0	NaN
3	2020-02-02	2	1	NaN
4	2020-02-03	3	1	NaN
5	2020-02-04	3	0	NaN
6	2020-02-05	3	0	NaN
7	2020-02-06	3	0	NaN
8	2020-02-07	3	0	NaN
9	2020-02-08	3	0	NaN
10	2020-02-09	3	0	NaN
11	2020-02-10	3	0	NaN
12	2020-02-11	3	0	NaN
13	2020-02-12	3	0	NaN
14	2020-02-13	3	0	NaN
15	2020-02-14	3	0	NaN
16	2020-02-15	3	0	NaN
17	2020-02-16	3	0	NaN
18	2020-02-17	3	0	NaN
19	2020-02-18	3	0	NaN
20	2020-02-19	3	0	NaN
21	2020-02-20	3	0	NaN
22	2020-02-21	3	0	NaN
23	2020-02-22	3	0	NaN
24	2020-02-23	3	0	NaN
25	2020-02-24	3	0	NaN
26	2020-02-25	3	0	NaN
27	2020-02-26	3	0	NaN
28	2020-02-27	3	0	NaN
29	2020-02-28	3	0	NaN
30	2020-02-29	3	0	NaN
31	2020-03-01	3	0	NaN
32	2020-03-02	6	3	NaN
33	2020-03-03	9	3	NaN
34	2020-03-04	28	19	NaN

	Date	Total Cases	New Cases	Days after surpassing 100 cases
35	2020-03-05	30	2	NaN
36	2020-03-06	31	1	NaN
37	2020-03-07	34	3	NaN
38	2020-03-08	39	5	NaN
39	2020-03-09	43	4	NaN
40	2020-03-10	56	13	NaN
41	2020-03-11	62	6	NaN
42	2020-03-12	73	11	NaN
43	2020-03-13	82	9	NaN
44	2020-03-14	102	20	0.0
45	2020-03-15	113	11	1.0
46	2020-03-16	119	6	2.0
47	2020-03-17	142	23	3.0
48	2020-03-18	156	14	4.0
49	2020-03-19	194	38	5.0
50	2020-03-20	244	50	6.0
51	2020-03-21	271	27	7.0

In [69]:
```
pdc_Italy=pd.read_excel(r"E:\Data\covid19ds_project\per_day_cases.xlsx",parse_
date=True,sheet_name="Italy")
pdc_Korea=pd.read_excel(r"E:\Data\covid19ds_project\per_day_cases.xlsx",parse_
date=True,sheet_name="Korea")
pdc_Wuhan=pd.read_excel(r"E:\Data\covid19ds_project\per_day_cases.xlsx",parse_
date=True,sheet_name="Wuhan")
```

In [70]:
```
pdc_Italy.head()
```

Out[70]:

	Date	Total Cases	New Cases	Days after surpassing 100 cases
0	2020-01-31	2	2	NaN
1	2020-02-01	2	0	NaN
2	2020-02-02	2	0	NaN
3	2020-02-03	2	0	NaN
4	2020-02-04	2	0	NaN

In [71]:
```
pdc_Korea.head()
```

Out[71]:

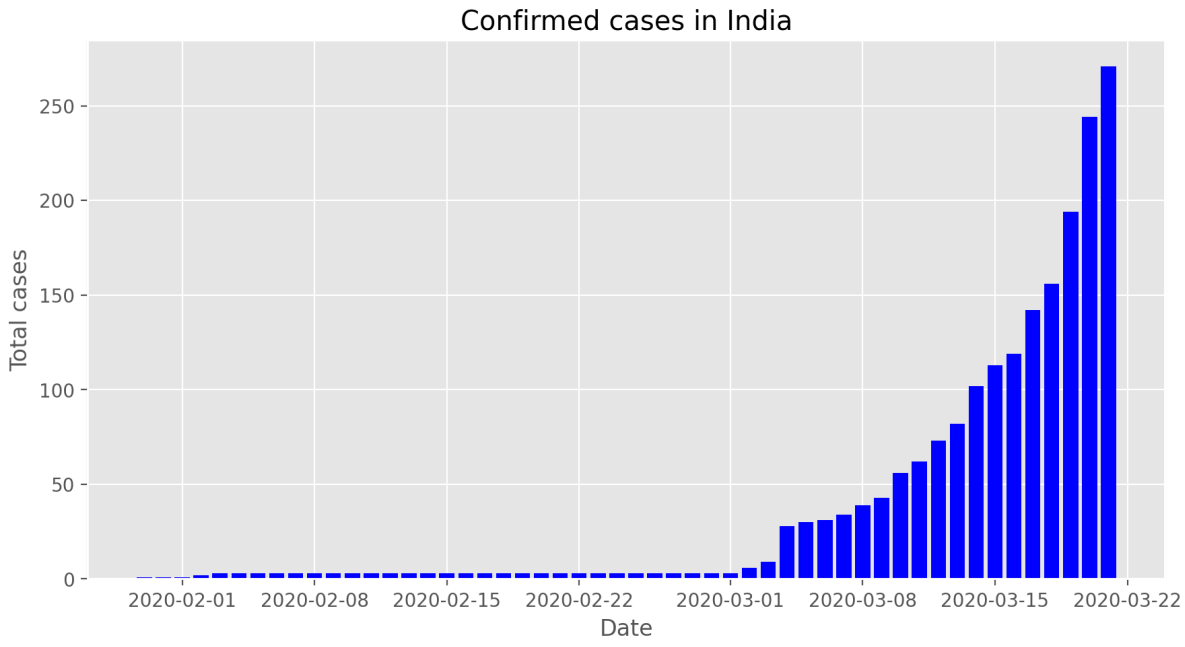
	Date	Total Cases	New Cases	Days after surpassing 100 cases
0	2020-01-20	1	1	NaN
1	2020-01-21	1	0	NaN
2	2020-01-22	1	0	NaN
3	2020-01-23	1	0	NaN
4	2020-01-24	2	1	NaN

In [72]:
```
pdc_Wuhan.head()
```

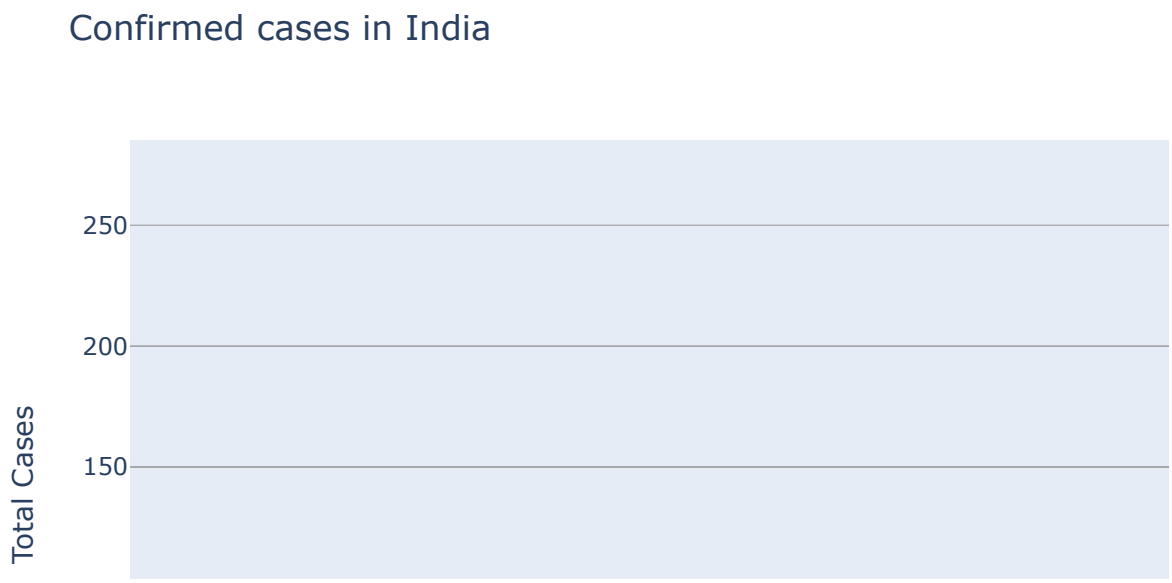
Out[72]:

	Date	Total Cases	New Cases
0	2020-01-22	444	0
1	2020-01-23	444	0
2	2020-01-24	549	105
3	2020-01-25	761	212
4	2020-01-26	1058	297

In [75]:
```
# data visualization using matplotlib
fig=plt.figure(figsize=(10,5),dpi=200)
axes=fig.add_axes([0.1,0.1,0.8,0.8])
axes.bar(pdc_India['Date'],pdc_India['Total Cases'],color='blue')
axes.set_xlabel("Date")
axes.set_ylabel("Total cases")
axes.set_title("Confirmed cases in India")
plt.show()
```



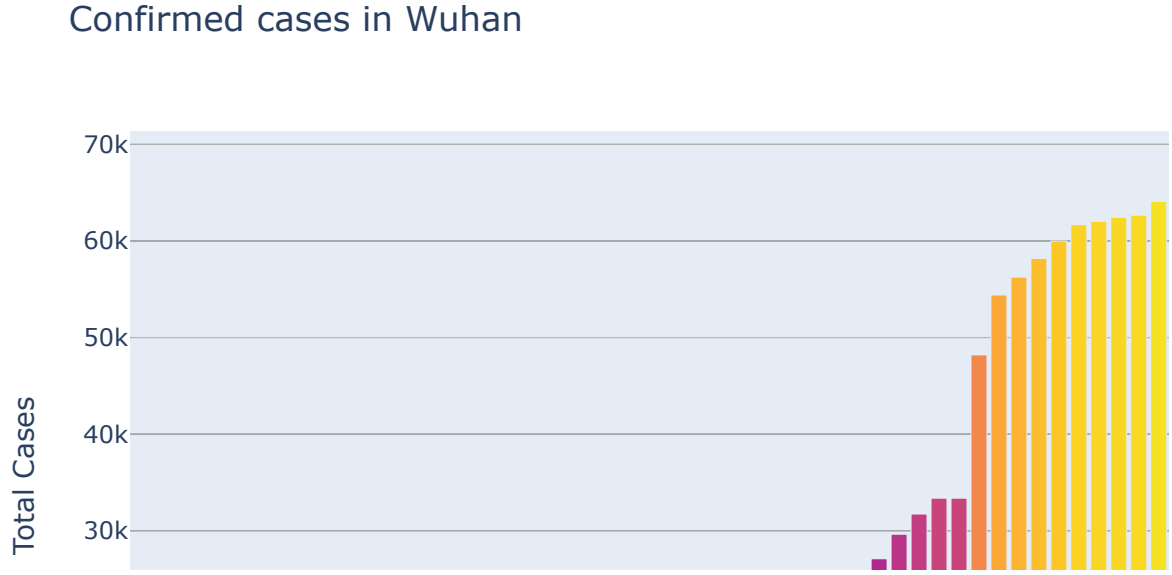
In [78]:
```
# visualization with plotly.express
fig=px.bar(pdc_India,x=pdc_India['Date'],y=pdc_India['Total Cases'],color='Total Cases',title="Confirmed cases in India")
fig.show()
```



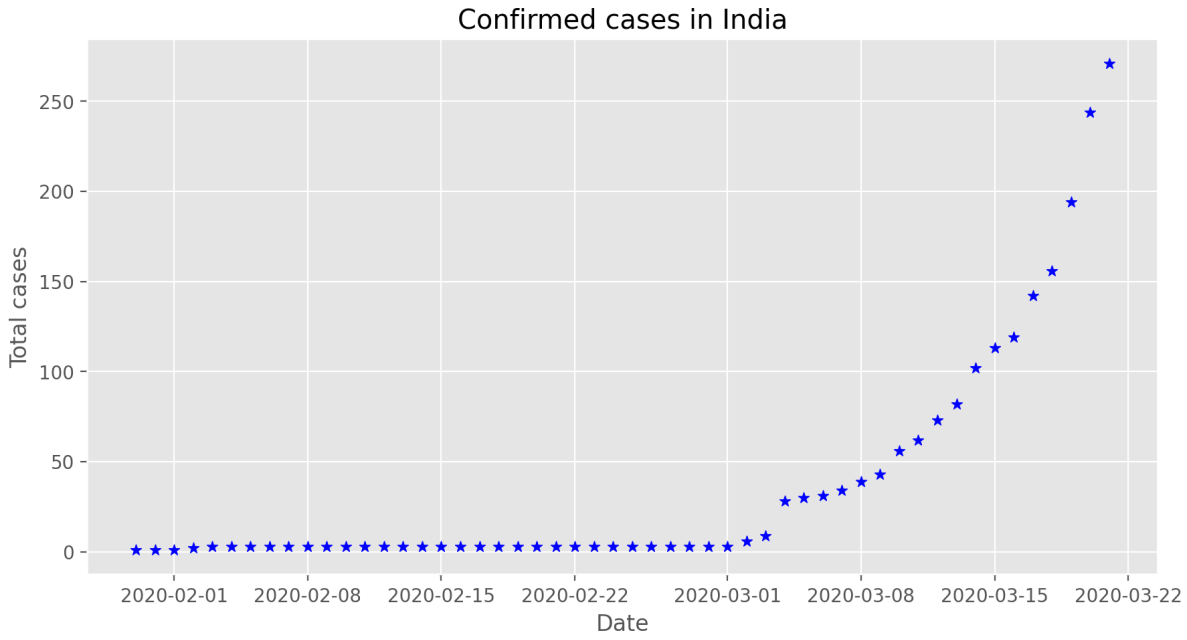
In [80]:
```
# visualization with plotly.express
fig=px.bar(pdc_Italy,x=pdc_Italy['Date'],y=pdc_Italy['Total Cases'],color='Tot
al Cases',title="Confirmed cases in Italy")
fig.show()

# visualization with plotly.express
fig=px.bar(pdc_Korea,x=pdc_Korea['Date'],y=pdc_Korea['Total Cases'],color='Tot
al Cases',title="Confirmed cases in Korea")
fig.show()

# visualization with plotly.express
fig=px.bar(pdc_Wuhan,x=pdc_Wuhan['Date'],y=pdc_Wuhan['Total Cases'],color='Tot
al Cases',title="Confirmed cases in Wuhan")
fig.show()
```


Confirmed cases in Italy
Total Cases
40k
30k
20k

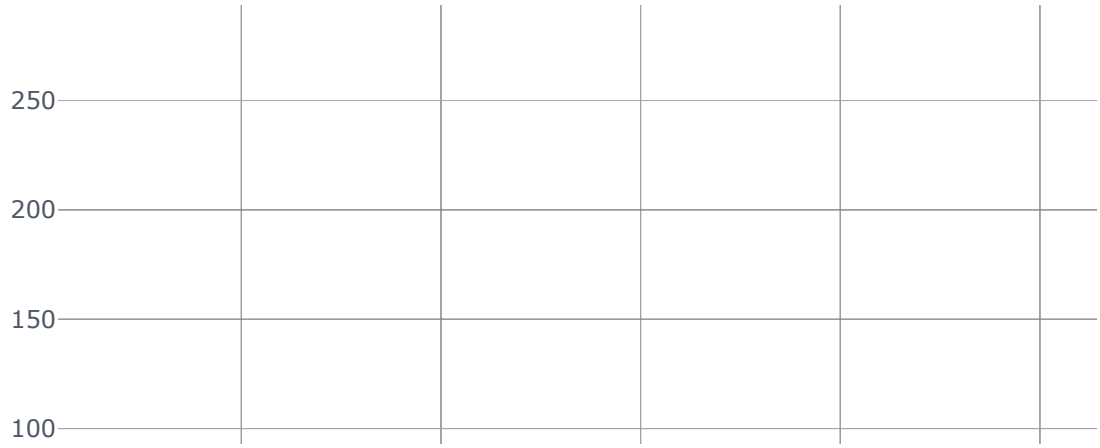

Confirmed cases in Korea
Total Cases
9000
8000
7000
6000
5000
4000


Confirmed cases in Wuhan
Total Cases
70k
60k
50k
40k
30k

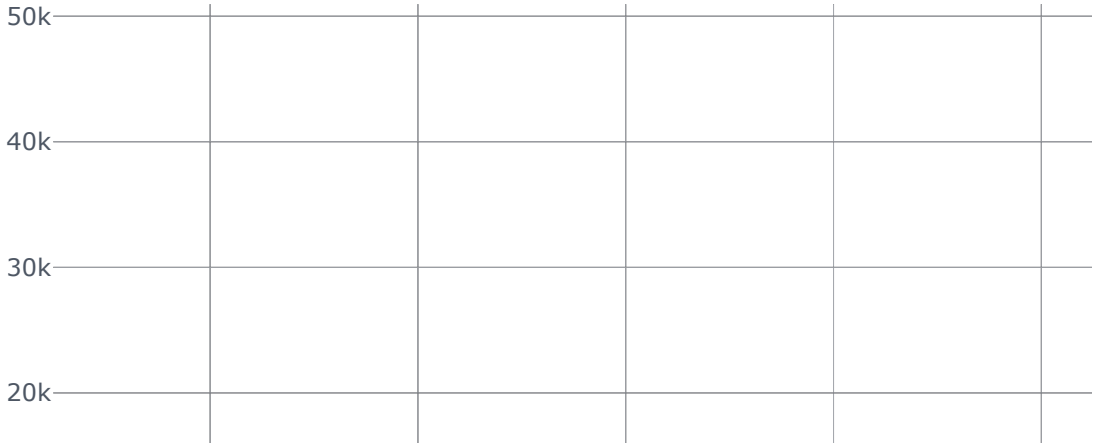
In [85]:
```
# scatter plot visualization of the data using matplotlib
fig=plt.figure(figsize=(10,5),dpi=200)
axes=fig.add_axes([0.1,0.1,0.8,0.8])
axes.scatter(pdc_India['Date'],pdc_India['Total Cases'],color='blue',marker=
'*')
axes.set_xlabel("Date")
axes.set_ylabel("Total cases")
axes.set_title("Confirmed cases in India")
plt.show()
```

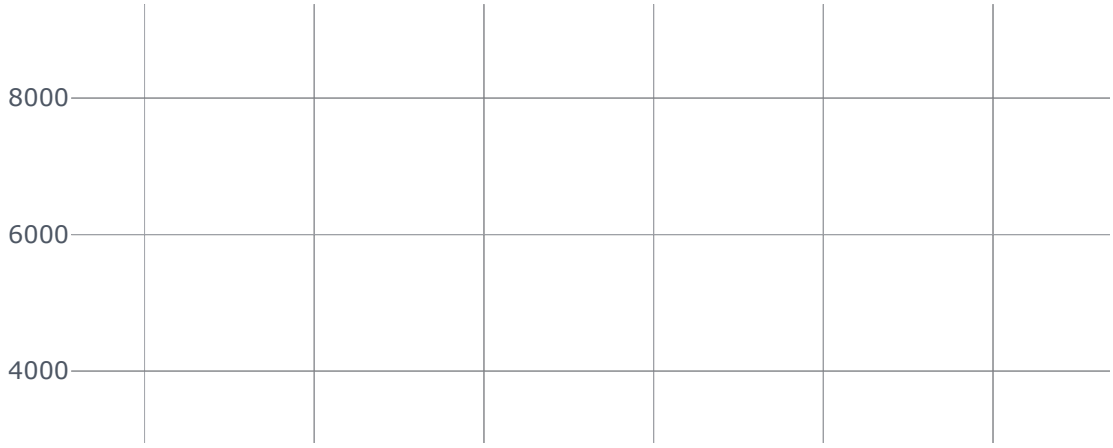


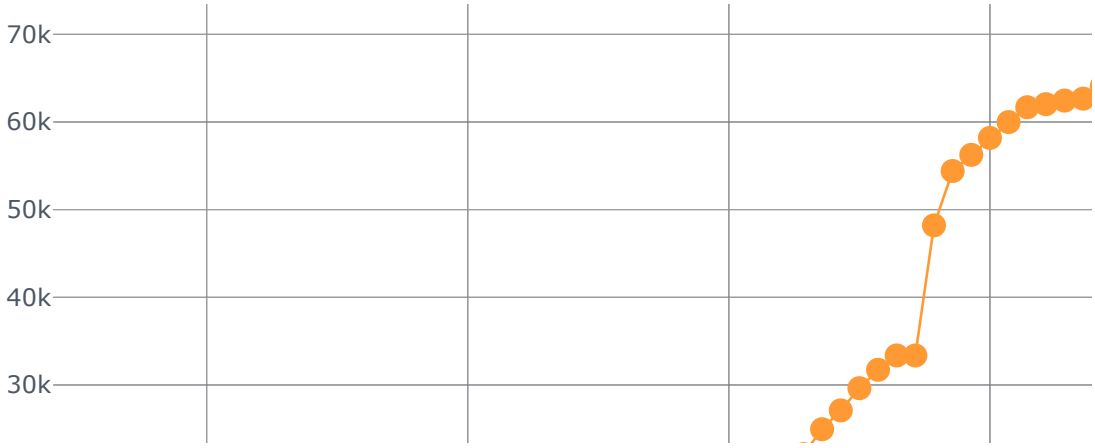
In [86]:
```
# scatter plot visualization using plotly
pdc_India.iplot(kind='scatter',x='Date',y='Total Cases',mode='lines+markers')
```



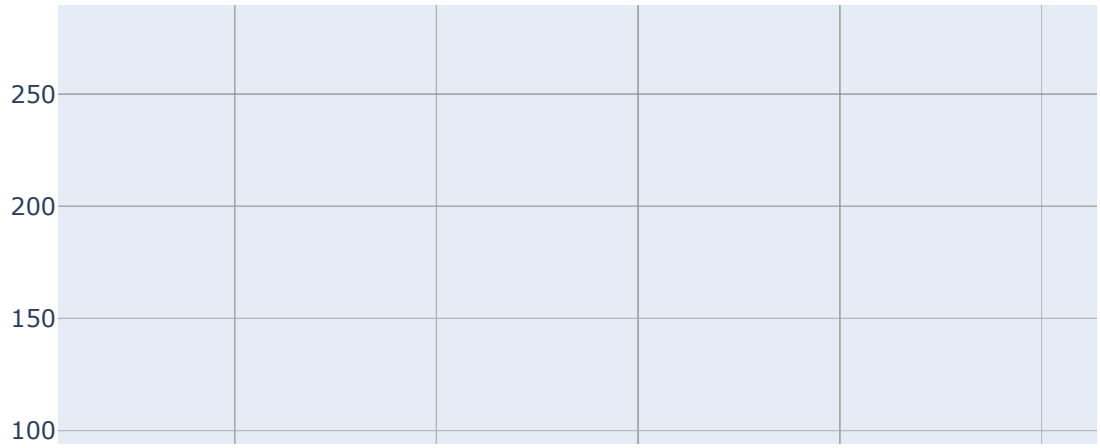
In [89]:
```
# plotly visualization of the data
pdc_Italy.iplot(kind='scatter',x='Date',y='Total Cases',mode='lines+markers')
pdc_Korea.iplot(kind='scatter',x='Date',y='Total Cases',mode='lines+markers')
pdc_Wuhan.iplot(kind='scatter',x='Date',y='Total Cases',mode='lines+markers')
```


50k
40k
30k
20k

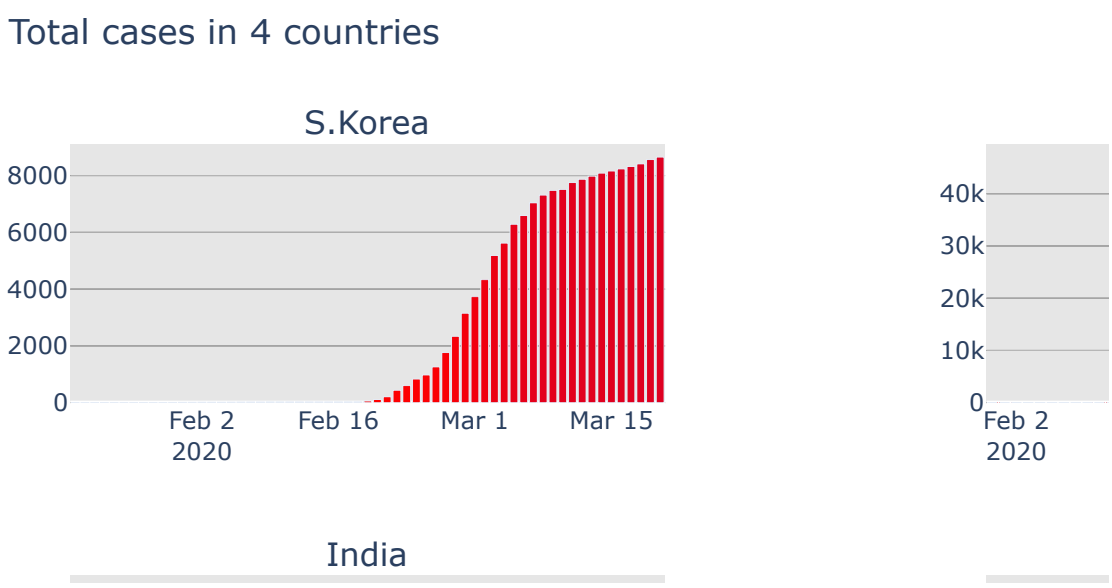

8000
6000
4000


70k
60k
50k
40k
30k

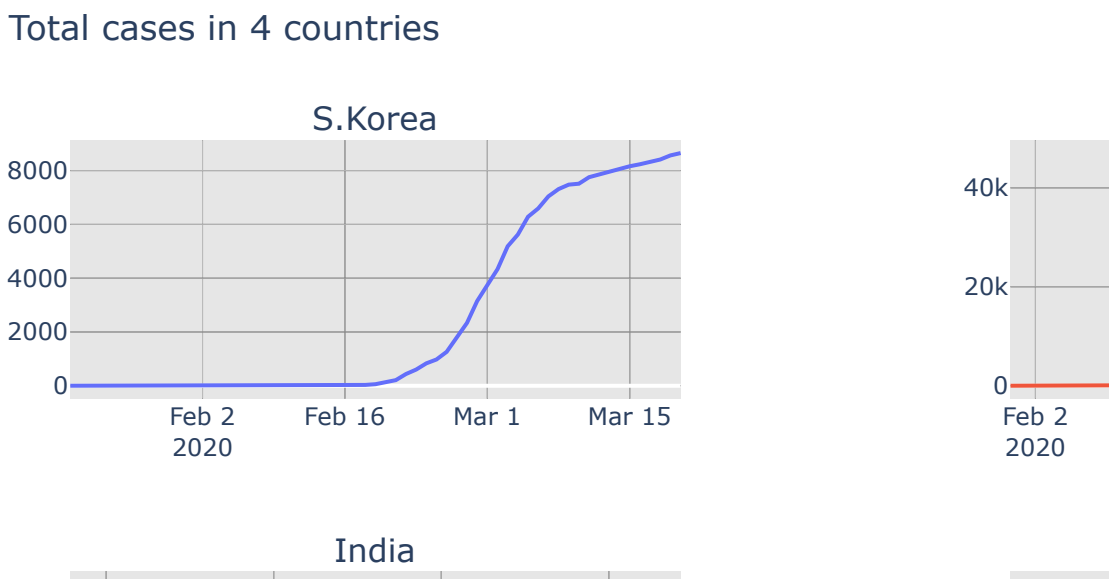
In [90]:
```
# visualization using plotly.graph_objects
fig=go.Figure()
fig.add_trace(go.Scatter(x=pdc_India['Date'],y=pdc_India['Total Cases'],mode=
'lines+markers'))
```



In [98]:
```
# Subplots using bar graph
# In subplots all the graphs can be compared side by side

from plotly.subplots import make_subplots

fig=make_subplots(
    rows=2,cols=2,
    specs=[[{"secondary_y":True},{"secondary_y":True}],[{"secondary_y":True},{
"secondary_y":True}]],
    subplot_titles=("S.Korea","Italy","India","Wuhan")
)

fig.add_trace(go.Bar(x=pdc_Korea['Date'],y=pdc_Korea['Total Cases'],
                     marker=dict(color=pdc_Korea['Total Cases'],coloraxis='col
oraxis')),1,1)

fig.add_trace(go.Bar(x=pdc_Italy['Date'],y=pdc_Italy['Total Cases'],
                     marker=dict(color=pdc_Italy['Total Cases'],coloraxis='col
oraxis')),1,2)

fig.add_trace(go.Bar(x=pdc_India['Date'],y=pdc_India['Total Cases'],
                     marker=dict(color=pdc_India['Total Cases'],coloraxis='col
oraxis')),2,1)

fig.add_trace(go.Bar(x=pdc_Wuhan['Date'],y=pdc_Wuhan['Total Cases'],
                     marker=dict(color=pdc_Wuhan['Total Cases'],coloraxis='col
oraxis')),2,2)

fig.update_layout(coloraxis=dict(colorscale='Bluered_r'),showlegend=False,titl
e_text='Total cases in 4 countries')
fig.update_layout(plot_bgcolor='rgb(230,230,230)')
```


Total cases in 4 countries
S.Korea
8000
6000
4000
2000
0
Feb 2
2020
Feb 16
Mar 1
Mar 15
40k
30k
20k
10k
0
Feb 2
2020
India

In [99]:
```
# subplots using scatter plots

from plotly.subplots import make_subplots

fig=make_subplots(
    rows=2,cols=2,
    specs=[[{"secondary_y":True},{"secondary_y":True}],[{"secondary_y":True},{
"secondary_y":True}]],
    subplot_titles=("S.Korea","Italy","India","Wuhan")
)

fig.add_trace(go.Scatter(x=pdc_Korea['Date'],y=pdc_Korea['Total Cases'],
                         marker=dict(color=pdc_Korea['Total Cases'],coloraxis='col
oraxis')),1,1)

fig.add_trace(go.Scatter(x=pdc_Italy['Date'],y=pdc_Italy['Total Cases'],
                         marker=dict(color=pdc_Italy['Total Cases'],coloraxis='col
oraxis')),1,2)

fig.add_trace(go.Scatter(x=pdc_India['Date'],y=pdc_India['Total Cases'],
                         marker=dict(color=pdc_India['Total Cases'],coloraxis='col
oraxis')),2,1)

fig.add_trace(go.Scatter(x=pdc_Wuhan['Date'],y=pdc_Wuhan['Total Cases'],
                         marker=dict(color=pdc_Wuhan['Total Cases'],coloraxis='col
oraxis')),2,2)

fig.update_layout(coloraxis=dict(colorscale='Bluered_r'),showlegend=False,titl
e_text='Total cases in 4 countries')
fig.update_layout(plot_bgcolor='rgb(230,230,230)')
```



Time Series of covid19

In [106]:
```
data=pd.read_csv("E:\Data\covid19ds_project\covid_19_data.csv",parse_dates=['Last Update'])
data.head()
```

Out[106]:

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovere
0	1	01/22/2020	Anhui	Mainland China	2020-01-22 17:00:00	1.0	0.0	0
1	2	01/22/2020	Beijing	Mainland China	2020-01-22 17:00:00	14.0	0.0	0
2	3	01/22/2020	Chongqing	Mainland China	2020-01-22 17:00:00	6.0	0.0	0
3	4	01/22/2020	Fujian	Mainland China	2020-01-22 17:00:00	1.0	0.0	0
4	5	01/22/2020	Gansu	Mainland China	2020-01-22 17:00:00	0.0	0.0	0

In [107]:
```
data.rename(columns={'ObservationDate':'Date','Country/Region':'Country'},inplace=True)
data.head()
```

Out[107]:

	SNo	Date	Province/State	Country	Last Update	Confirmed	Deaths	Recovered
0	1	01/22/2020	Anhui	Mainland China	2020-01-22 17:00:00	1.0	0.0	0.0
1	2	01/22/2020	Beijing	Mainland China	2020-01-22 17:00:00	14.0	0.0	0.0
2	3	01/22/2020	Chongqing	Mainland China	2020-01-22 17:00:00	6.0	0.0	0.0
3	4	01/22/2020	Fujian	Mainland China	2020-01-22 17:00:00	1.0	0.0	0.0
4	5	01/22/2020	Gansu	Mainland China	2020-01-22 17:00:00	0.0	0.0	0.0

In [112]:
```
data.drop(columns=['SNo'],inplace=True)
data
```

Out[112]:

	Date	Province/State	Country	Last Update	Confirmed	Deaths	Recovered
0	01/22/2020	Anhui	Mainland China	2020-01-22 17:00:00	1.0	0.0	0.0
1	01/22/2020	Beijing	Mainland China	2020-01-22 17:00:00	14.0	0.0	0.0
2	01/22/2020	Chongqing	Mainland China	2020-01-22 17:00:00	6.0	0.0	0.0
3	01/22/2020	Fujian	Mainland China	2020-01-22 17:00:00	1.0	0.0	0.0
4	01/22/2020	Gansu	Mainland China	2020-01-22 17:00:00	0.0	0.0	0.0
...	...	...	...	...	...	...	...
5885	03/15/2020	Gibraltar	UK	2020-03-14 16:33:03	1.0	0.0	1.0
5886	03/15/2020	NaN	Uzbekistan	2020-03-15 18:20:19	1.0	0.0	0.0
5887	03/15/2020	Diamond Princess cruise ship	Australia	2020-03-14 02:33:04	0.0	0.0	0.0
5888	03/15/2020	West Virginia	US	2020-03-10 02:33:04	0.0	0.0	0.0
5889	03/15/2020	NaN	occupied Palestinian territory	2020-03-11 20:53:02	0.0	0.0	0.0

5890 rows × 7 columns

In [113]:
```
data.groupby("Date").sum()
```

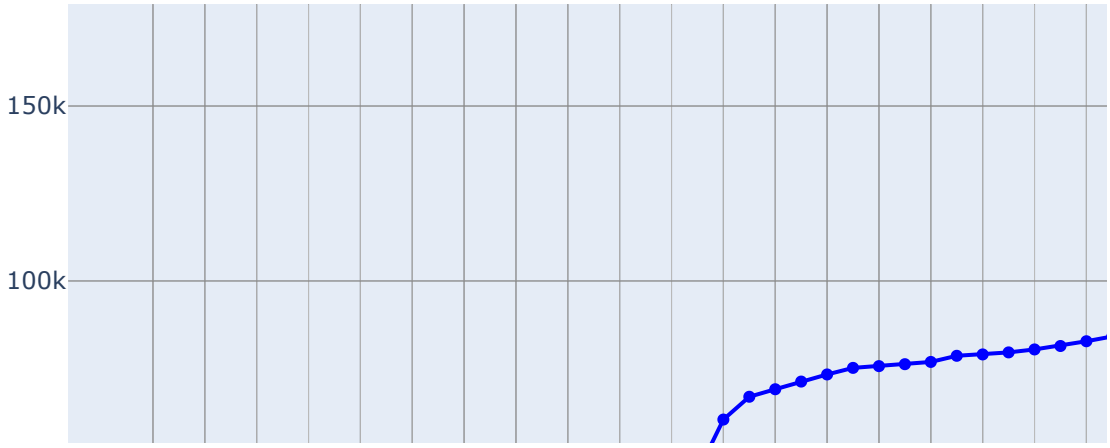
Out[113]:

Date	Confirmed	Deaths	Recovered
01/22/2020	555.0	17.0	28.0
01/23/2020	653.0	18.0	30.0
01/24/2020	941.0	26.0	36.0
01/25/2020	1438.0	42.0	39.0
01/26/2020	2118.0	56.0	52.0
01/27/2020	2927.0	82.0	61.0
01/28/2020	5578.0	131.0	107.0
01/29/2020	6165.0	133.0	126.0
01/30/2020	8235.0	171.0	143.0
01/31/2020	9925.0	213.0	222.0
02/01/2020	12038.0	259.0	284.0
02/02/2020	16787.0	362.0	472.0
02/03/2020	19881.0	426.0	623.0
02/04/2020	23892.0	492.0	852.0
02/05/2020	27636.0	564.0	1124.0
02/06/2020	30818.0	634.0	1487.0
02/07/2020	34392.0	719.0	2011.0
02/08/2020	37121.0	806.0	2616.0
02/09/2020	40151.0	906.0	3244.0
02/10/2020	42763.0	1013.0	3946.0
02/11/2020	44803.0	1113.0	4683.0
02/12/2020	45222.0	1118.0	5150.0
02/13/2020	60370.0	1371.0	6295.0
02/14/2020	66887.0	1523.0	8058.0
02/15/2020	69032.0	1666.0	9395.0
02/16/2020	71226.0	1770.0	10865.0
02/17/2020	73260.0	1868.0	12583.0
02/18/2020	75138.0	2007.0	14352.0
02/19/2020	75641.0	2122.0	16121.0
02/20/2020	76199.0	2247.0	18177.0
02/21/2020	76843.0	2251.0	18890.0
02/22/2020	78599.0	2458.0	22886.0
02/23/2020	78985.0	2469.0	23394.0
02/24/2020	79570.0	2629.0	25227.0

Date	Confirmed	Deaths	Recovered
02/25/2020	80415.0	2708.0	27905.0
02/26/2020	81397.0	2770.0	30384.0
02/27/2020	82756.0	2814.0	33277.0
02/28/2020	84124.0	2872.0	36711.0
02/29/2020	86013.0	2941.0	39782.0
03/01/2020	88371.0	2996.0	42716.0
03/02/2020	90309.0	3085.0	45602.0
03/03/2020	92844.0	3160.0	48229.0
03/04/2020	95124.0	3254.0	51171.0
03/05/2020	97886.0	3348.0	53797.0
03/06/2020	101800.0	3460.0	55866.0
03/07/2020	105836.0	3558.0	58359.0
03/08/2020	109835.0	3803.0	60695.0
03/09/2020	113582.0	3996.0	62512.0
03/10/2020	118582.0	4262.0	64404.0
03/11/2020	125865.0	4615.0	67003.0
03/12/2020	128343.0	4720.0	68324.0
03/13/2020	145193.0	5404.0	70251.0
03/14/2020	156099.0	5819.0	72624.0
03/15/2020	167447.0	6440.0	76034.0

In [117]:
```
confirmed=data.groupby("Date").sum()['Confirmed'].reset_index()
recovered=data.groupby("Date").sum()['Recovered'].reset_index()
deaths=data.groupby("Date").sum()['Deaths'].reset_index()
```

In [121]:
```
fig=go.Figure()
fig.add_trace(go.Scatter(x=confirmed['Date'],y=confirmed['Confirmed'],mode='lines+markers',name='confirmed',line=dict(color='blue',width=2)))

fig.add_trace(go.Scatter(x=recovered['Date'],y=recovered['Recovered'],mode='lines+markers',name='recovered',line=dict(color='red',width=2)))
fig.add_trace(go.Scatter(x=deaths['Date'],y=deaths['Deaths'],mode='lines+markers',name='deaths',line=dict(color='yellow',width=2)))
```



In [125]:
```
df_confirmed=pd.read_csv(r"E:\Data\covid19ds_project\time_series_covid_19_confirmed.csv")
df_confirmed.head()
```

Out[125]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
0	NaN	Thailand	15.0000	101.0000	2	3	5	7	8
1	NaN	Japan	36.0000	138.0000	2	1	2	2	4
2	NaN	Singapore	1.2833	103.8333	0	1	3	3	4
3	NaN	Nepal	28.1667	84.2500	0	0	0	1	1
4	NaN	Malaysia	2.5000	112.5000	0	0	0	3	4

5 rows × 57 columns

In [127]:
```
df_confirmed.rename(columns={'Country/Region':'Country'},inplace=True)
df_confirmed.head()
```

Out[127]:

	Province/State	Country	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/2
0	NaN	Thailand	15.0000	101.0000	2	3	5	7	8	
1	NaN	Japan	36.0000	138.0000	2	1	2	2	4	
2	NaN	Singapore	1.2833	103.8333	0	1	3	3	4	
3	NaN	Nepal	28.1667	84.2500	0	0	0	1	1	
4	NaN	Malaysia	2.5000	112.5000	0	0	0	3	4	

5 rows × 57 columns

In [131]:
```
df_lat_long=pd.merge(data,df_confirmed,on=['Country','Province/State'])
df_lat_long.head()
```

Out[131]:

	Date	Province/State	Country	Last Update	Confirmed	Deaths	Recovered	Lat	Lo
0	01/22/2020	Washington	US	2020-01-22 17:00:00	1.0	0.0	0.0	47.4009	-121.4
1	01/23/2020	Washington	US	2020-01-23 17:00:00	1.0	0.0	0.0	47.4009	-121.4
2	01/24/2020	Washington	US	2020-01-24 17:00:00	1.0	0.0	0.0	47.4009	-121.4
3	01/25/2020	Washington	US	2020-01-25 17:00:00	1.0	0.0	0.0	47.4009	-121.4
4	01/26/2020	Washington	US	2020-01-26 16:00:00	1.0	0.0	0.0	47.4009	-121.4

5 rows × 62 columns

In [136]:
```
fig=px.density_mapbox(df_lat_long,lat='Lat',lon='Long',hover_name="Province/State",hover_data=['Confirmed','Deaths','Recovered'],animation_frame="Date",color_continuous_scale='Portland',radius=7,zoom=0,height=700)
fig.update_layout(title='Worldwide Corona Virus Cases')
fig.update_layout(mapbox_style="open-street-map",mapbox_center_lon=0)
fig.update_layout(margin={'r':0,'t':0,'l':0,'b':0})
```

End of Project