Project – COVID-19 Analysis

Problem Statement:

Given data about COVID-19 patients, write code to visualize the impact and analyze the trend of rate of infection and recovery as well as make predictions about the number of cases expected a week in future based on the current trends.

Dataset:

CSV and Excel files containing data about the number of COVID-19 confirmed deaths and recovered patients both around the world and in India.

Guidelines:

- Use pandas to accumulate data from multiple data files.
- Use plotly (visualization library) to create interactive visualizations.
- Use Facebook prophet library to make time series models.
- Visualize the prediction by combining these technologies.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly
```

In [2]:
```
import plotly.io as pio
pio.renderers.default = "notebook+pdf"
```

In [3]:
```
df = pd.read_csv("covid_19_clean_complete _1_.csv")
df
```

Out[3]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Active	WHO Region
0	NaN	Afghanistan	33.939110	67.709953	2020-01-22	0	0	0	0	Eastern Mediterranean
1	NaN	Albania	41.153300	20.168300	2020-01-22	0	0	0	0	Europe
2	NaN	Algeria	28.033900	1.659600	2020-01-22	0	0	0	0	Africa
3	NaN	Andorra	42.506300	1.521800	2020-01-22	0	0	0	0	Europe
4	NaN	Angola	-11.202700	17.873900	2020-01-22	0	0	0	0	Africa
...	...	...	...	...	...	...	...	...	...	...
49063	NaN	Sao Tome and Principe	0.186400	6.613100	2020-07-27	865	14	734	117	Africa
49064	NaN	Yemen	15.552727	48.516388	2020-07-27	1691	483	833	375	Eastern Mediterranean
49065	NaN	Comoros	-11.645500	43.333300	2020-07-27	354	7	328	19	Africa
49066	NaN	Tajikistan	38.861000	71.276100	2020-07-27	7235	60	6028	1147	Europe
49067	NaN	Lesotho	-29.610000	28.233600	2020-07-27	505	12	128	365	Africa

49068 rows × 10 columns

In [4]:
```
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')
```

In [5]:
```
df.drop('Province/State', axis=1, inplace=True)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49068 entries, 0 to 49067
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country/Region  49068 non-null  object
 1   Lat             49068 non-null  float64
 2   Long            49068 non-null  float64
 3   Date            49068 non-null  datetime64[ns]
 4   Confirmed       49068 non-null  int64
 5   Deaths          49068 non-null  int64
 6   Recovered       49068 non-null  int64
 7   Active          49068 non-null  int64
 8   WHO Region      49068 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(4), object(2)
memory usage: 3.4+ MB
```

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

Out[7]:

	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Active	WHO Region
0	Afghanistan	33.939110	67.709953	2020-01-22	0	0	0	0	Eastern Mediterranean
1	Albania	41.153300	20.168300	2020-01-22	0	0	0	0	Europe
2	Algeria	28.033900	1.659600	2020-01-22	0	0	0	0	Africa
3	Andorra	42.506300	1.521800	2020-01-22	0	0	0	0	Europe
4	Angola	-11.202700	17.873900	2020-01-22	0	0	0	0	Africa
...	...	...	...	...	...	...	...	...	...
49063	Sao Tome and Principe	0.186400	6.613100	2020-07-27	865	14	734	117	Africa
49064	Yemen	15.552727	48.516388	2020-07-27	1691	483	833	375	Eastern Mediterranean
49065	Comoros	-11.645500	43.333300	2020-07-27	354	7	328	19	Africa
49066	Tajikistan	38.861000	71.276100	2020-07-27	7235	60	6028	1147	Europe
49067	Lesotho	-29.610000	28.233600	2020-07-27	505	12	128	365	Africa

49068 rows × 9 columns

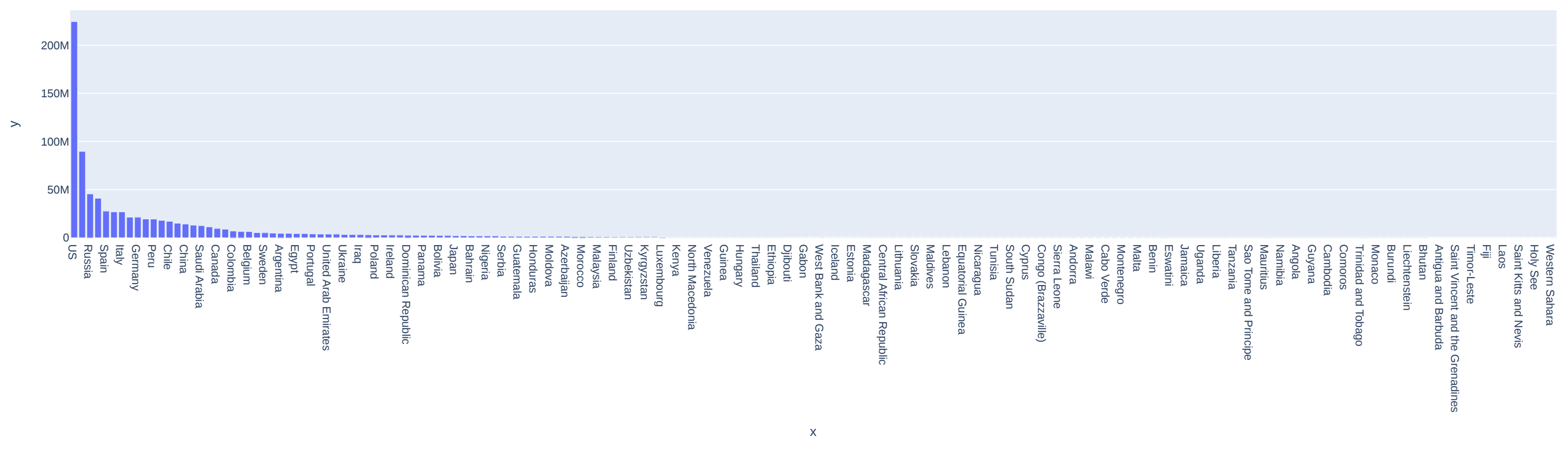
In [8]:
```
df_agg_country_sorted = df.groupby('Country/Region').agg(
    Total_Confirmed=('Confirmed', 'sum'),
    Total_Deaths=('Deaths', 'sum'),
    Total_Recovered=('Recovered', 'sum'),
    Total_Active=('Active', 'sum')).sort_values('Total_Confirmed', ascending=False).reset_index()
df_agg_country_sorted
```

Out[8]:

	Country/Region	Total_Confirmed	Total_Deaths	Total_Recovered	Total_Active
0	US	224345948	11011411	56353416	156981121
1	Brazil	89524967	3938034	54492873	31094060
2	Russia	45408411	619385	25120448	19668578
3	India	40883464	1111831	23783720	15987913
4	Spain	27404045	3033030	15093583	9277432
...	...	...	...	...	...
182	Saint Kitts and Nevis	1772	0	1295	477
183	Greenland	1507	0	1372	135
184	Holy See	1356	0	742	614
185	Papua New Guinea	1185	2	695	488
186	Western Sahara	901	63	648	190

187 rows × 5 columns

In [9]:
```
fig = px.bar(x=df_agg_country_sorted['Country/Region'], y=df_agg_country_sorted['Total_Confirmed'])
fig.show()
```

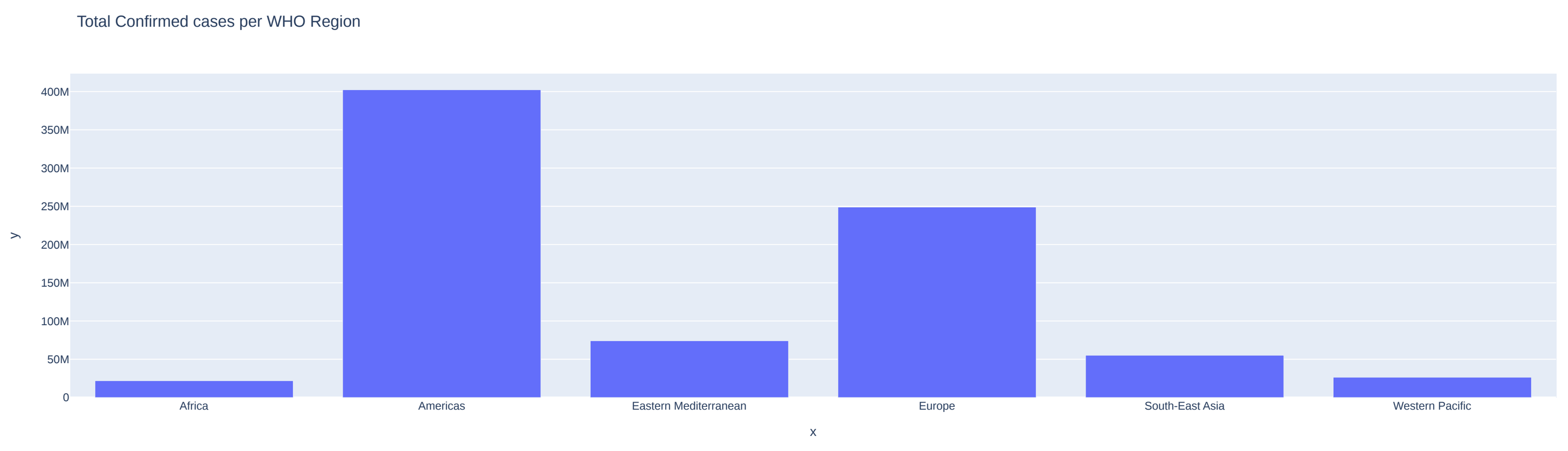


In [10]:
```
df_agg_who = df.groupby('WHO Region').agg(
    Total_Confirmed=('Confirmed', 'sum'),
    Total_Deaths=('Deaths', 'sum'),
    Total_Recovered=('Recovered', 'sum'),
    Total_Active=('Active', 'sum')).reset_index()
df_agg_who
```

Out[10]:

	WHO Region	Total_Confirmed	Total_Deaths	Total_Recovered	Total_Active
0	Africa	21791827	439978	11193730	10158119
1	Americas	402261194	19359292	157069444	225832458
2	Eastern Mediterranean	74082892	1924029	48050703	24108160
3	Europe	248879793	19271040	123202075	106406678
4	South-East Asia	55118365	1458134	30030327	23629904
5	Western Pacific	26374411	932430	18861950	6580031

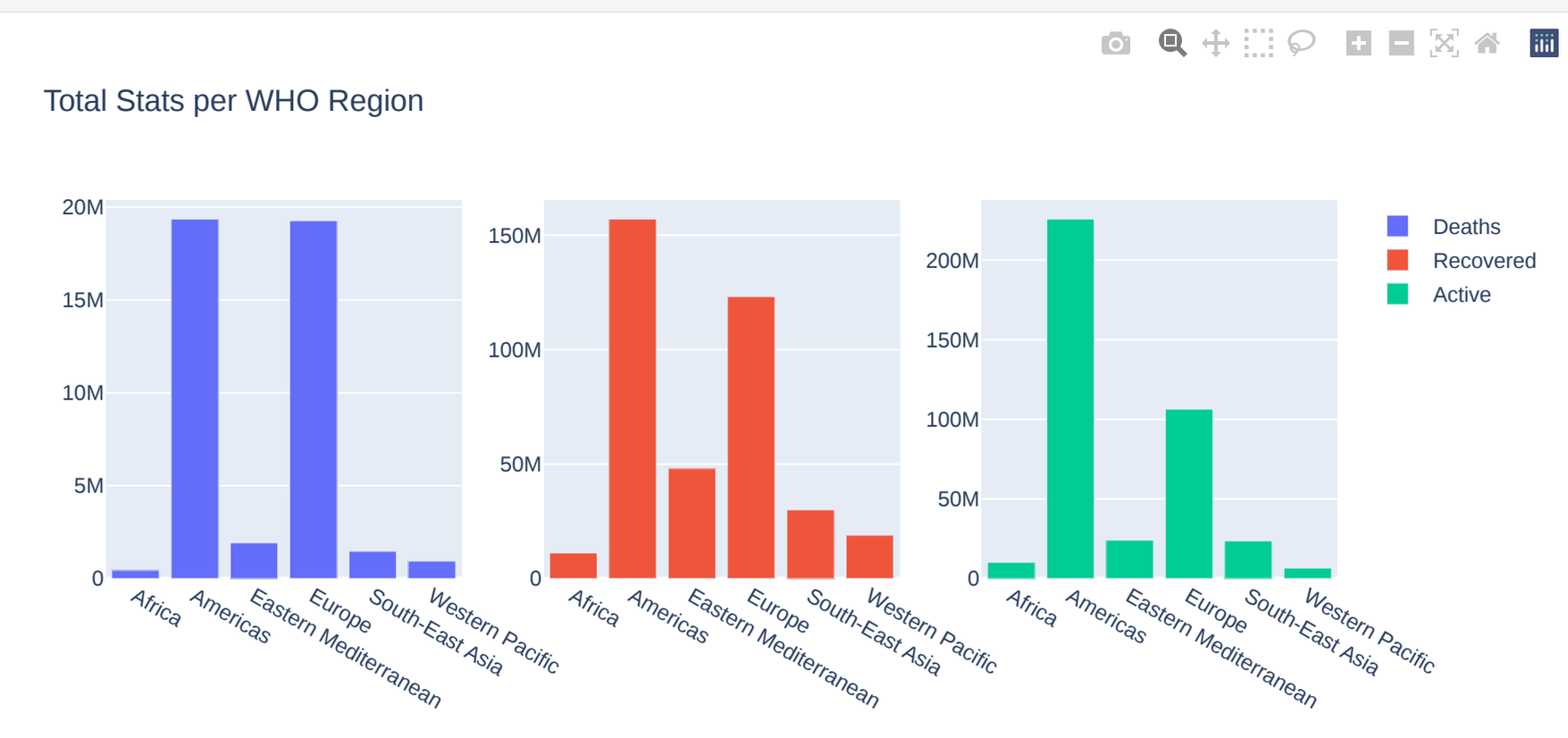
In [11]:
```
fig = px.bar(x=df_agg_who['WHO Region'], y=df_agg_who['Total_Confirmed'], title='Total Confirmed cases per WHO Region')
fig.show()
```



In [12]:
```
fig = make_subplots(rows=1, cols=3)

fig.add_trace(
    go.Bar(x=df_agg_who['WHO Region'], y=df_agg_who['Total_Deaths'], name='Deaths'),
    row=1, col=1
)
fig.add_trace(
    go.Bar(x=df_agg_who['WHO Region'], y=df_agg_who['Total_Recovered'], name='Recovered'),
    row=1, col=2
)
fig.add_trace(
    go.Bar(x=df_agg_who['WHO Region'], y=df_agg_who['Total_Active'], name='Active'),
    row=1, col=3
)


fig.update_layout(height=400, width=900, title_text="Total Stats per WHO Region")
fig.show()
```



In [13]:
```
df_Prophet = pd.DataFrame()
df_Prophet['ds'] = df[['Date']]
df_Prophet['y'] = df[['Confirmed']]
```

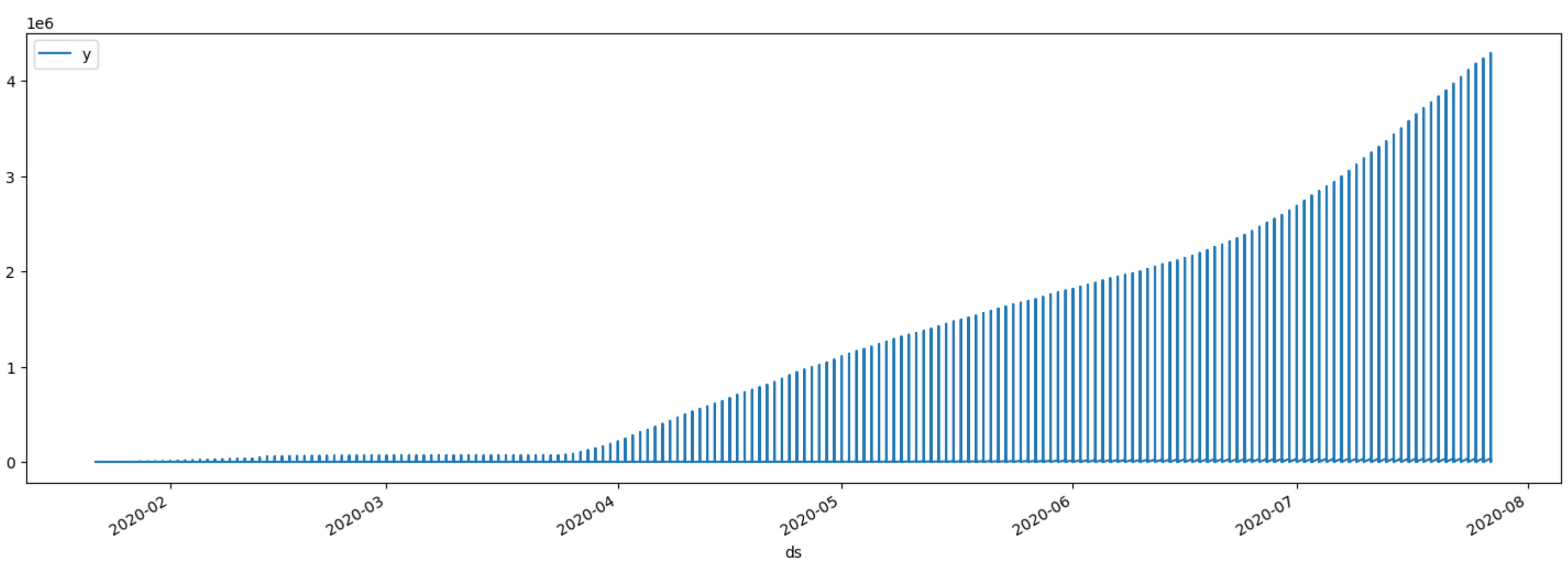
In [14]:
```
df_Prophet.tail()
```

Out[14]:

	ds	y
49063	2020-07-27	865
49064	2020-07-27	1691
49065	2020-07-27	354
49066	2020-07-27	7235
49067	2020-07-27	505

In [15]:
```
df_Prophet.plot(x='ds', y='y', figsize=(18,6))
```

Out[15]: <Axes: xlabel='ds'>



In [16]:
```
m = Prophet()
m.fit(df_Prophet)
```

```
18:48:49 - cmdstanpy - INFO - Chain [1] start processing
18:48:52 - cmdstanpy - INFO - Chain [1] done processing
```

Out[16]: <prophet.forecaster.Prophet at 0x226618818d0>

In [17]:
```
future = m.make_future_dataframe(periods=60)
future.tail()
```

Out[17]:

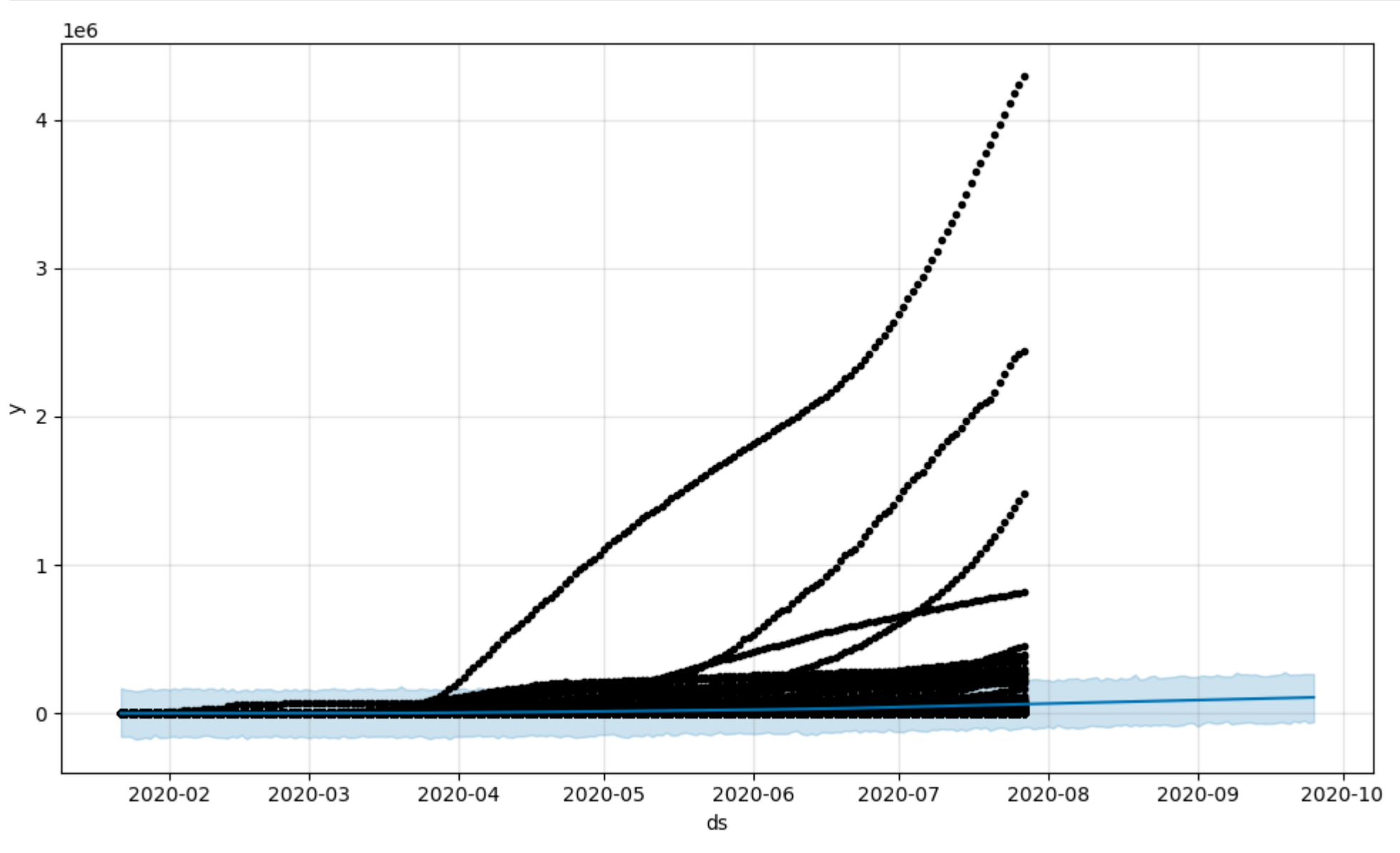
	ds
243	2020-09-21
244	2020-09-22
245	2020-09-23
246	2020-09-24
247	2020-09-25

In [18]:
```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

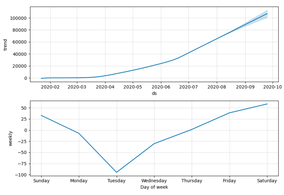
Out[18]:

	ds	yhat	yhat_lower	yhat_upper
243	2020-09-21	104009.084467	-51671.433436	261362.803415
244	2020-09-22	104681.595970	-62774.926124	261272.054124
245	2020-09-23	105506.136445	-47348.394848	265736.526637
246	2020-09-24	106298.264285	-65388.935765	265503.745564
247	2020-09-25	107095.668541	-58267.545211	266258.406036

In [19]:
```
fig1 = m.plot(forecast)
```



In [20]:
```
fig2 = m.plot_components(forecast)
```


trend
100000
80000
60000
40000
20000
0
2020-02
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
ds

weekly
50
25
0
−25
−50
−75
−100
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week

In [ ]: