In [1]:
```
# loading libraries:
import numpy as np
import pandas as pd
from datetime import datetime
import os
import plotly.graph_objects as go
import plotly.express as px
from IPython.core.display import HTML
from plotly.subplots import make_subplots
```

In [2]:
```
# chanding the working directory:
os.chdir("")
```

In [3]:
```
# Import xlsx file and store each sheet in to a df list in our working d
irectory:
xl_file = pd.ExcelFile('')
dfs = {sheet_name: xl_file.parse(sheet_name)
          for sheet_name in xl_file.sheet_names}
# Data from each sheet can be accessed via key:
keyList = list(dfs.keys())

df = dfs[keyList[0]]
df.head()
```

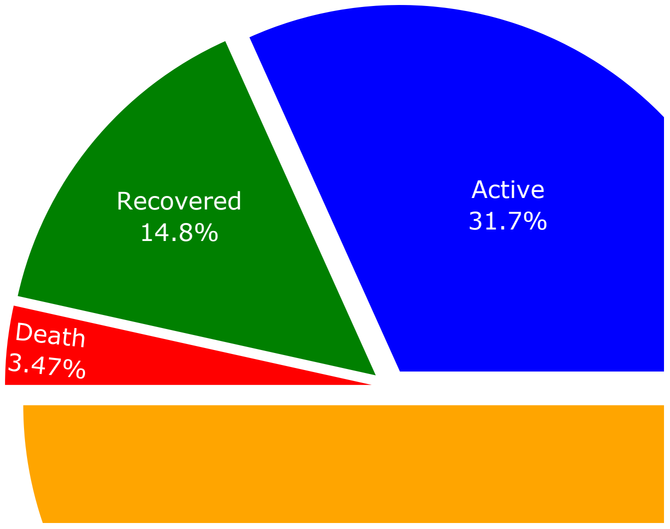
Out[3]:

	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered	Tests	Critical
0	Heilongjiang	Mainland China	4/28/2020 16:30	939	13	586	0	0
1	Shanghai	Mainland China	4/28/2020 16:30	644	7	584	0	0
2	Inner Mongolia	Mainland China	4/28/2020 16:30	199	1	145	0	0
3	Shaanxi	Mainland China	4/28/2020 16:30	306	3	253	0	0
4	Beijing	Mainland China	4/28/2020 16:30	593	9	525	0	0

In [4]:
```
# Data cleaning:
cases = ['Confirmed', 'Deaths', 'Recovered', 'Active']

# replacing Mainland china with just China
df['Country/Region'] = df['Country/Region'].replace('Mainland China', 'C
hina')

# Active Case = confirmed - deaths - recovered
df['Active'] = df['Confirmed'] - df['Deaths'] - df['Recovered']

# filling missing values
df[['Province/State']] = df[['Province/State']].fillna('')
df[cases] = df[cases].fillna(0)

# renaming Country/Region to Country:
df.rename(columns = {'Country/Region':'Country'}, inplace = True)

# printing the latest date of the outbreak and counting how many days of
the outbreak
latestDate=(df['Last Update'][0])
daysOutbreak=(datetime.strptime('04/28/2020', '%m/%d/%Y') - datetime.str
ptime('12/31/2019', '%m/%d/%Y')).days
```

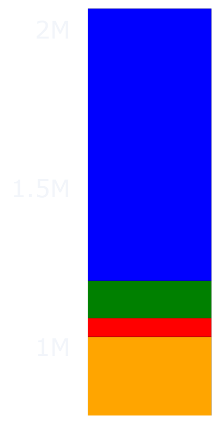
In [5]:
```
# Save numbers into variables to use in the app
confirmedCases=df['Confirmed'].sum()
deathsCases=df['Deaths'].sum()
recoveredCases=df['Recovered'].sum()
activeCases=df['Active'].sum()
```

```
In [6]: # Pie Chart of Covid-19
        data = [['Confirmed', confirmedCases],['Death', deathsCases], ['Recovere
        d', recoveredCases],['Active', activeCases]]
        df2 = pd.DataFrame(data, columns = ['state', 'count'])
        colors = ['orange', 'red', 'green', 'blue']
        fig1 = px.pie(df2,
                     values="count",
                     names="state",
                     template="seaborn")
        fig1.update_traces(rotation=90, pull=0.05, textinfo="percent+label", mar
        ker = dict(colors=colors))
        fig1.layout.template="plotly_dark"
        fig1.show()
```



In [14]:
```
# Top 10 countries Affected by COVID-19
ncov = df.groupby(['Country', 'Last Update'])['Confirmed', 'Deaths', 'Re
covered','Active'].sum()
ncov_all= ncov.reset_index().drop_duplicates(subset=['Country'], keep='l
ast')
ncov_all.reset_index(drop=True, inplace=True)
ncov_all = ncov_all.sort_values(by=['Confirmed'], ascending=False).reset
_index(drop=True)
ncov_all = ncov_all.head(10)

fig3 = go.Figure(data=[
    go.Bar(name='Confirmed',x=ncov_all['Country'].unique(), y=ncov_all[
'Confirmed'], marker_color="orange"),
    go.Bar(name='Deaths', x=ncov_all['Country'].unique(), y=ncov_all['De
aths'],marker_color ="red"),
    go.Bar(name='Recovered', x=ncov_all['Country'].unique(), y=ncov_all[
'Recovered'], marker_color = "green"),
    go.Bar(name='Active', x=ncov_all['Country'].unique(), y=ncov_all['Ac
tive'], marker_color ="blue")

])
# Change the bar mode layout
fig3.layout.update(barmode='stack', yaxis_showgrid=False)
fig3.layout.template="plotly_dark"
fig3.show()
```

```
/Users/karimaidrissi/opt/anaconda3/lib/python3.7/site-packages/ipykerne
l_launcher.py:2: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) w
ill be deprecated, use a list instead.
```



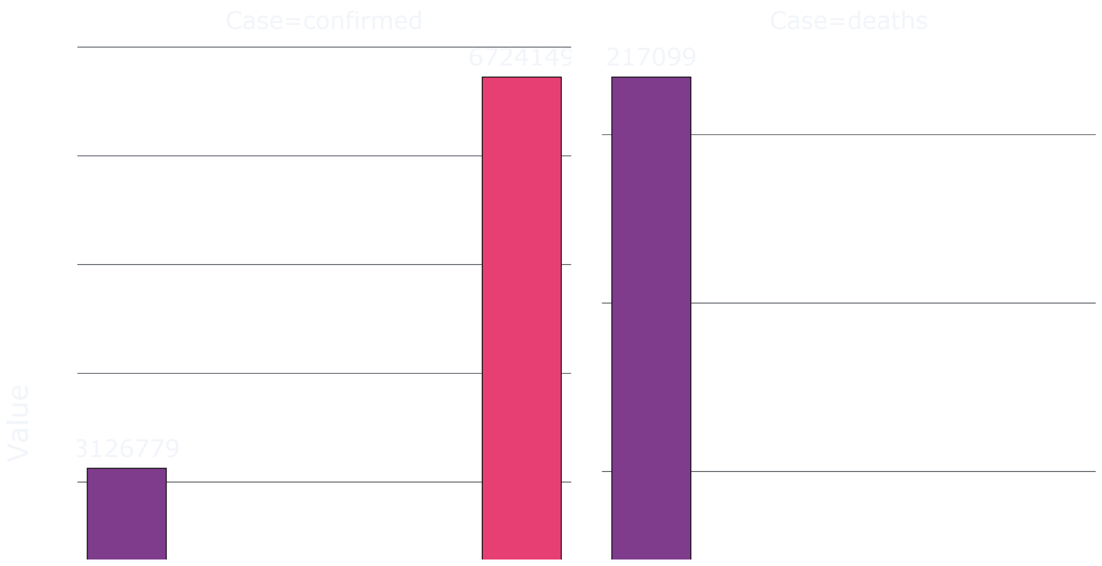
In [16]:

```
# loading the epidemic diseases :
epidemics = pd.DataFrame({
    'epidemic' : ['COVID-19', 'SARS', 'EBOLA', 'MERS', 'H1N1'],
    'start_year' : [2019, 2003, 2014, 2012, 2009],
    'end_year' : [2020, 2004, 2016, 2017, 2010],
    'confirmed' : [df['Confirmed'].sum(), 8096, 28646, 2494, 6724149],
    'deaths' : [df['Deaths'].sum(), 774, 11323, 858, 19654]
})

epidemics['mortality'] = round((epidemics['deaths']/epidemics['confirme
d'])*100, 2)

epidemics.head()
```

Out[16]:

	epidemic	start_year	end_year	confirmed	deaths	mortality
0	COVID-19	2019	2020	3126779	217099	6.94
1	SARS	2003	2004	8096	774	9.56
2	EBOLA	2014	2016	28646	11323	39.53
3	MERS	2012	2017	2494	858	34.40
4	H1N1	2009	2010	6724149	19654	0.29

In [17]:
```
# Comparing COVID-19 with other epidemic diseases:
epi = epidemics.melt(id_vars='epidemic', value_vars=['confirmed', 'death
s', 'mortality'],
                     var_name='Case', value_name='Value')

fig2 = px.bar(epi, x="epidemic", y="Value", color='epidemic', text='Valu
e', facet_col="Case",
             color_discrete_sequence = px.colors.qualitative.Bold)
fig2.update_traces(textposition='outside')
fig2.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')
fig2.update_yaxes(showticklabels=False)
fig2.layout.yaxis2.update(matches=None)
fig2.layout.yaxis3.update(matches=None)
fig2.layout.template="plotly_dark"
fig2.show()
```



In [18]:
```
# loading world coordinate data:
world_coordinate = pd.read_csv("/Users/karimaidrissi/Desktop/DSSA 5103 v
z/world_coordinates 2.csv")
world_coordinate.head()
```

Out[18]:

	Code	Country	latitude	longitude
0	AD	Andorra	42.546245	1.601554
1	AE	United Arab Emirates	23.424076	53.847818
2	AF	Afghanistan	33.939110	67.709953
3	AG	Antigua and Barbuda	17.060816	-61.796428
4	AI	Anguilla	18.220554	-63.068615

In [19]:
```
# counting total cases for each country:
number_of_countries = len(df['Country'].value_counts())
cases = pd.DataFrame(df.groupby('Country')['Confirmed','Deaths','Recover
ed','Active'].sum())
cases['Country'] = cases.index
cases.index = np.arange(1,number_of_countries+1)

total_cases = cases[["Country", "Confirmed","Deaths","Recovered","Activ
e"]]

total_cases.head()
```

```
/Users/karimaidrissi/opt/anaconda3/lib/python3.7/site-packages/ipykerne
l_launcher.py:3: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) w
ill be deprecated, use a list instead.
```

Out[19]:

	Country	Confirmed	Deaths	Recovered	Active
1	Afghanistan	1828	58	228	1542
2	Albania	750	30	431	289
3	Algeria	3649	437	1651	1561
4	Andorra	743	40	385	318
5	Angola	27	2	6	19

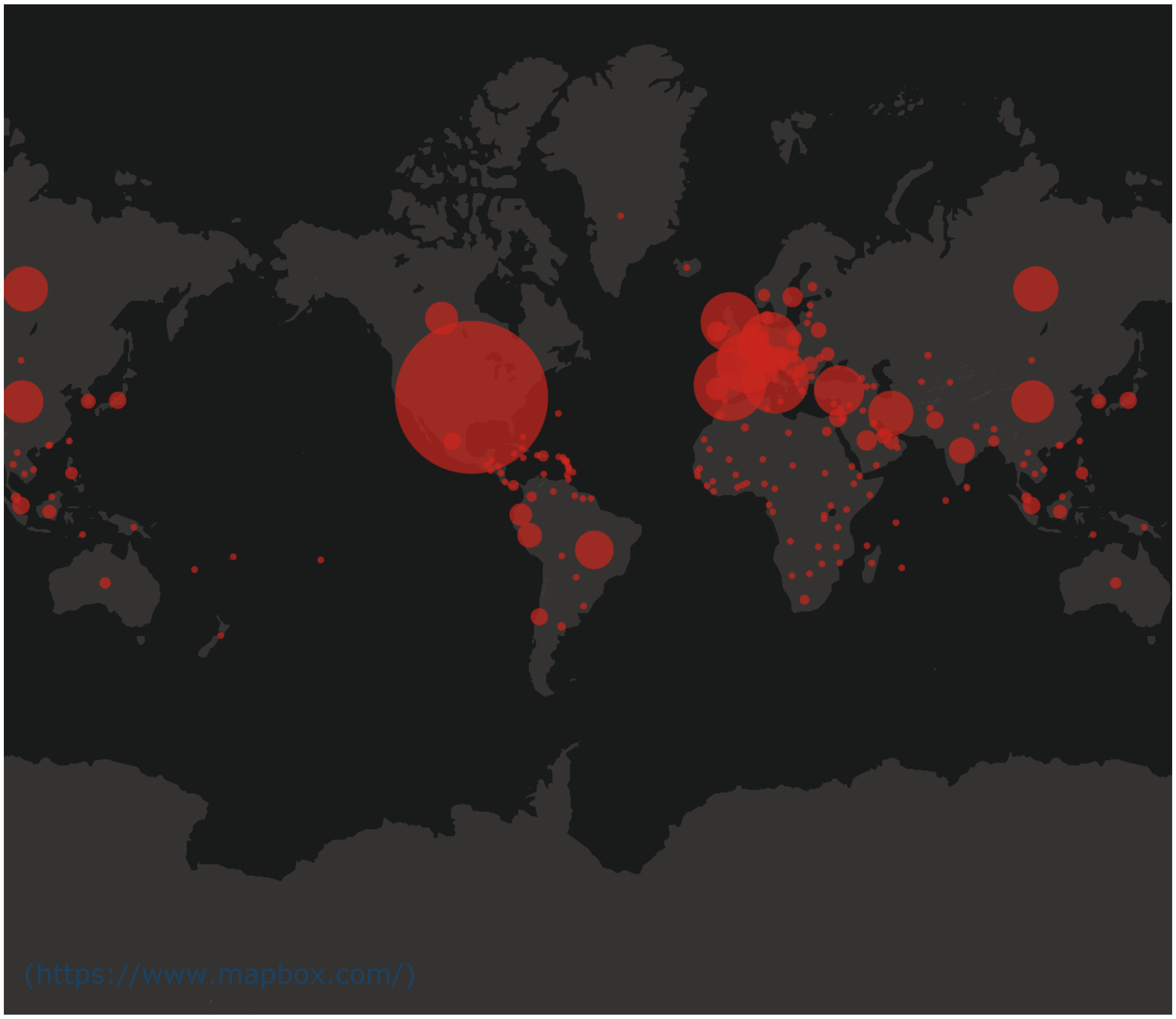
In [24]:
```
# merge the world coordinate data with the total cases on Country column
world_data = pd.merge(world_coordinate,total_cases,on=['Country'])
world_data
```

Out[24]:

	Code	Country	latitude	longitude	Confirmed	Deaths	Recovered	Active
0	AD	Andorra	42.546245	1.601554	743	40	385	318
1	AE	United Arab Emirates	23.424076	53.847818	11380	89	2181	9110
2	AF	Afghanistan	33.939110	67.709953	1828	58	228	1542
3	AG	Antigua and Barbuda	17.060816	-61.796428	24	3	11	10
4	AI	Anguilla	18.220554	-63.068615	3	0	3	0
...	...	...	...	...	...	...	...	...
181	YE	Yemen	15.552727	48.516388	1	0	1	0
182	YT	Mayotte	-12.827500	45.166244	460	4	235	221
183	ZA	South Africa	-30.559482	22.937506	4996	93	2073	2830
184	ZM	Zambia	-13.133897	27.849332	95	3	42	50
185	ZW	Zimbabwe	-19.015438	29.154857	32	4	5	23

186 rows × 8 columns

In [25]:
```
# Geographic Distribution of COVID-19
# load the mapbox key:
mapbox_access_token = "xxxxxxx"
# Generate a list for hover text display
textList=world_data['Country']
fig4 = go.Figure(go.Scattermapbox(
        lat=world_data['latitude'],
        lon=world_data['longitude'],
        mode='markers',
        marker=go.scattermapbox.Marker(
            color='#ca261d',
            size=world_data['Confirmed'].tolist(),
            sizemin=2,
            sizemode='area',
            sizeref=2.*max(world_data['Confirmed'].tolist())/(80.**2),
        ),
        text=textList,
        hovertext=['Comfirmed: {}<br>Recovered: {}<br>Death: {}<br>Activ
e: {}'.format(i, j, k,n) for i, j, k,n in zip(world_data['Confirmed'],

world_data['Recovered'],

world_data['Deaths'],

world_data['Active'])],

        hovertemplate = "<b>%{text}</b><br><br>" +"%{hovertext}<br>" +"<
extra></extra>")
        )

fig4.update_layout(
    plot_bgcolor='#151920',
    paper_bgcolor='#121d1f',
    margin=go.layout.Margin(l=0,r=0,b=0,t=0,pad=5),
    hovermode='closest',
    mapbox=go.layout.Mapbox(
        accesstoken=mapbox_access_token,
        style="dark",
        bearing=0,
        center=go.layout.mapbox.Center(
            lat=43,
            lon=-75
        ),
        pitch=0,
        zoom=1
    )
)

fig4.show()
```



In [26]:
```
# import dash packages:
import dash
import dash_core_components as dcc
import dash_html_components as html
```

In [27]:
```
app = dash.Dash(__name__, assets_folder='./assets/',
                meta_tags=[
                    {"name": "viewport", "content": "width=device-width,
height=device-height, initial-scale=1.0"}
                ]
      )
```

In [28]:
```
app.layout = html.Div(
    children=[
        html.Div(
            id="header",
            children=[
                html.H1(children="Dashboard Tracking the Spread of COVID
-19",
                        style={'textAlign': 'center'}
                       ),
                 html.Div(children='''
                Daily updated of global confirmed, recovered, deaths and
active cases.
                '''),
                html.P(style={'fontWeight':'bold',},
                    children="Last Updated on {}.".format(latestDate))
            ]
        ),
        html.Div(
            id="number-plate",
            style={'marginLeft':'1.5%','marginRight':'.8%','marginBotto
m':'.5%'},
                 children=[
                      html.Div(
                         style={'width':'19%','backgroundColor':'#c2bf3
4','display':'inline-block',
                                'marginRight':'.8%','verticalAlign':'to
p'},
                              children=[
                                  html.H3(style={'textAlign':'center','f
ontWeight':'bold','color':'#090a0a'},
                                              children=[
                                                  html.P(style={'fontS
ize':'2rem','padding':'.5rem'}),
                                                  '{:,d}'.format(daysO
utbreak)
                                                        ]),
                                  html.P(style={'textAlign':'center','fo
ntWeight':'bold','color':'#090a0a','padding':'.1rem'},
                                             children="Days Since the
 Outbreak")
                                   ]),
                     html.Div(
                         style={'width':'19%','backgroundColor':'#d7781
9','display':'inline-block',
                                'marginRight':'.8%','verticalAlign':'to
p'},
                              children=[
                                  html.H3(style={'textAlign':'center','f
ontWeight':'bold','color':'#090a0a'},
                                              children=[
                                                  html.P(style={'fontS
ize':'2rem','padding':'.5rem'}),
                                                  '{:,d}'.format(confi
rmedCases)
                                                        ]),
                                  html.P(style={'textAlign':'center','fo
ntWeight':'bold','color':'#090a0a','padding':'.1rem'},
                                                children="Confirmed Case
s")
                                        ]),
                     html.Div(
                         style={'width':'18%','backgroundColor':'#1a962
2','display':'inline-block',
                                'marginRight':'.8%','verticalAlign':'to
p'},
                              children=[
                                  html.H3(style={'textAlign':'center','f
ontWeight':'bold','color':'#090a0a'},
                                                children=[
                                                    html.P(style={'fontSi
ze':'2rem','padding':'.5rem'}),
                                                    '{:,d}'.format(recove
redCases),
                                                         ]),
                                  html.P(style={'textAlign':'center',
                                                   'fontWeight':'bol
d','color':'#090a0a','padding':'.1rem'},
                                                children="Recovered Case
s")
                                        ]),
                     html.Div(
                         style={'width':'18%','backgroundColor':'#e3050
8','display':'inline-block',
                                'marginRight':'.8%','verticalAlign':'to
p'},
                              children=[
                                  html.H3(style={'textAlign':'center',
                                                    'fontWeight':'bol
d','color':'#090a0a'},
                                                 children=[
                                                     html.P(style={'fontS
ize':'2rem','padding':'.5rem'}),
                                                     '{:,d}'.format(death
sCases),
                                                          ]),
                                  html.P(style={'textAlign':'center',
                                                   'fontWeight':'bol
d','color':'#090a0a','padding':'.1rem'},
                                                children="Death Cases")
                                        ]),
                     html.Div(
                         style={'width':'20%','backgroundColor':'#2219d
7','display':'inline-block',
                                'marginRight':'.8%','verticalAlign':'to
p'},
                              children=[
                                  html.H3(style={'textAlign':'center',
                                                    'fontWeight':'bol
d','color':'#090a0a'},
                                                 children=[
                                                     html.P(style={'fontS
ize':'2rem','padding':'.5rem'}),
                                                     '{:,d}'.format(activ
eCases),
                                                         ]),
                                  html.P(style={'textAlign':'center',
                                                    'fontWeight':'bol
d','color':'#090a0a','padding':'.1rem'},
                                               children="Active Cases")
                                       ]),
                   ]),
        html.Div(
            id='dcc-plot',
            style={'marginLeft':'1.5%','marginRight':'1.5%','marginBotto
m':'.35%','marginTop':'.5%'},
                 children=[
                     html.Div(
                         style={'width':'37%','display':'inline-block',
'marginRight':'.8%','verticalAlign':'top'},
                                children=[
                                    html.H3(style={'textAlign':'center','b
ackgroundColor':'#cbd2d3',
                                                 'color':'#2c2c2c','padd
ing':'1rem','marginBottom':'0'},
                                                 children='Condensed COVID
-19 Cases'),
                                    dcc.Graph(figure=fig1)]),
                     html.Div(
                         style={'width':'60%','display':'inline-block',
'verticalAlign':'top'},
                                children=[
                                    html.H3(style={'textAlign':'center','b
ackgroundColor':'#cbd2d3',
                                                 'color':'#292929','padd
ing':'1rem','marginBottom':'0'},
                                                 children='Comparing COVID
-19 with Other Diseases'),
                                    dcc.Graph(figure=fig2)]),
                     html.Div(
                         style={'width':'100%','display':'inline-block',
'verticalAlign':'top'},
                                children=[
                                    html.H3(style={'textAlign':'center','b
ackgroundColor':'#cbd2d3',
                                                 'color':'#292929','padd
ing':'1rem','marginBottom':'0'},
                                                 children='Top 10 Countrie
s Affected by COVID-19'),
                                    dcc.Graph(figure=fig3)])]),

        html.Div(
            id='dcc-map',
            style={'marginLeft':'1.5%','marginRight':'1.5%','marginBotto
m':'.5%'},
                 children=[
                     html.Div(style={'width':'100%','display':'inline-bl
ock','verticalAlign':'top'},
                                children=[
                                    html.H3(style={'textAlign':'center','b
ackgroundColor':'#cbd2d3',
                                                  'color':'#292929','padd
ing':'1rem','marginBottom':'0'},
                                                children='Geographical Di
stribution of COVID-19'),
                                 dcc.Graph(figure=fig4)]),]),


        html.Div(style={'marginLeft':'1.5%','marginRight':'1.5%'},
                 children=[
                     html.P(style={'textAlign':'center','margin':'auto'
},
                            children=["Data source from ",
                                      html.A('Johns Hopkins University C
SSE', href='https://github.com/CSSEGISandData/COVID-19/tree/master/csse_
covid_19_data'),
                                      " Developed by ",html.A('Karima Ta
jin ', href='https://www.linkedin.com/in/karima-tajin-a42a9892/'),html.A
('Hssaine Zahiri', href = 'https://www.linkedin.com/in/hssaine-zahiri-a4
45487a/')])])

            ])
```

In [ ]:

```
if __name__ == '__main__':
    app.run_server(port=1991)
```

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production
deployment.
   Use a production WSGI server instead.
 * Debug mode: off

 * Running on http://127.0.0.1:1991/ (Press CTRL+C to quit)
127.0.0.1 - - [03/May/2020 12:57:12] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2020 12:57:13] "GET /_dash-dependencies HTTP/1.1"
200 -
127.0.0.1 - - [03/May/2020 12:57:13] "GET /_dash-layout HTTP/1.1" 200 -
```

In [ ]:

In [ ]: