install and import packages to the enviroment

In [1]:

```
#install packages required for the enviroment.
#!pip install chart_studio
#!pip install autoviz
#!pip install xlrd
#!pip install -q sklearn

#import packages to the enviroment
import pandas as pd
import numpy as np
import chart_studio.plotly as py
import cufflinks as cf
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from autoviz.AutoViz_Class import AutoViz_Class
import h2o
from h2o.automl import H2OAutoML
import pandas as pd
h2o.init()


import tensorflow as tf

from tensorflow import feature_column
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split


%matplotlib inline

# Make Plotly work in your Jupyter Notebook
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
# Use Plotly locally
cf.go_offline()

print("Packages installed in enviroment successfully")
```

```
Imported AutoViz_Class version: 0.0.81. Call using:
    from autoviz.AutoViz_Class import AutoViz_Class
    AV = AutoViz_Class()
    AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0, verbose=
0,
                           lowess=False,chart_format='svg',max_rows_analy
zed=150000,max_cols_analyzed=30)
Note: verbose=0 or 1 generates charts and displays them in your local Jupy
ter notebook.
      verbose=2 saves plots in your local machine under AutoViz_Plots dire
ctory and does not display charts.
Checking whether there is an H2O instance running at http://localhost:5432
1 (http://localhost:54321) ..... not found.
Attempting to start a local H2O server...
; Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
  Starting server from C:\Users\MSI\anaconda3\Lib\site-packages\h2o\backen
d\bin\h2o.jar
  Ice root: C:\Users\MSI\AppData\Local\Temp\tmph8fl847m
  JVM stdout: C:\Users\MSI\AppData\Local\Temp\tmph8fl847m\h2o_MSI_started_
from_python.out
  JVM stderr: C:\Users\MSI\AppData\Local\Temp\tmph8fl847m\h2o_MSI_started_
```

```
from_python.err
  Server is running at http://127.0.0.1:54321 (http://127.0.0.1:54321)
Connecting to H2O server at http://127.0.0.1:54321 (http://127.0.0.1:5432
1) ... successful.
```

H2O_cluster_uptime:	02 secs
H2O_cluster_timezone:	Europe/Paris
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.32.1.3
H2O_cluster_version_age:	3 days
H2O_cluster_name:	H2O_from_python_MSI_e7bj7w
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	1.766 Gb
H2O_cluster_total_cores:	4
H2O_cluster_allowed_cores:	4
H2O_cluster_status:	accepting new members, healthy
H2O_connection_url:	http://127.0.0.1:54321
H2O_connection_proxy:	{"http": null, "https": null}
H2O_internal_security:	False
H2O_API_Extensions:	Amazon S3, Algos, AutoML, Core V3, TargetEncoder, Core V4
Python_version:	3.8.5 final

```
Packages installed in enviroment successfully
```

Load Dataset and inspect

Now that we have installed and imported the packages that we will use for the enviroment. lets go ahead and load the dataset for inspection and analysis.

In [2]:

```
#load the dataset
df = pd.read_csv("country_wise_latest.csv")

print("'Dataset information'")
print()
df.info()
```

```
'Dataset information'

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Country/Region         187 non-null    object
 1   Confirmed              187 non-null    int64
 2   Deaths                 187 non-null    int64
 3   Recovered              187 non-null    int64
 4   Active                 187 non-null    int64
 5   New cases              187 non-null    int64
 6   New deaths             187 non-null    int64
 7   New recovered          187 non-null    int64
 8   Deaths / 100 Cases     187 non-null    float64
 9   Recovered / 100 Cases  187 non-null    float64
 10  Deaths / 100 Recovered  187 non-null    float64
 11  Confirmed last week    187 non-null    int64
 12  1 week change          187 non-null    int64
 13  1 week % increase      187 non-null    float64
 14  WHO Region             187 non-null    object
dtypes: float64(4), int64(9), object(2)
memory usage: 22.0+ KB
```

In [3]:

```
#columns in the dataset
df.columns
```

Out[3]:

```
Index(['Country/Region', 'Confirmed', 'Deaths', 'Recovered', 'Active',
       'New cases', 'New deaths', 'New recovered', 'Deaths / 100 Cases',
       'Recovered / 100 Cases', 'Deaths / 100 Recovered',
       'Confirmed last week', '1 week change', '1 week % increase',
       'WHO Region'],
      dtype='object')
```

In [4]:

```
#describe the dataset
df.describe()
```

Out[4]:

	Confirmed	Deaths	Recovered	Active	New cases	New deaths	
count	1.870000e+02	187.000000	1.870000e+02	1.870000e+02	187.000000	187.000000	
mean	8.813094e+04	3497.518717	5.063148e+04	3.400194e+04	1222.957219	28.957219	
std	3.833187e+05	14100.002482	1.901882e+05	2.133262e+05	5710.374790	120.037173	4
min	1.000000e+01	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	
25%	1.114000e+03	18.500000	6.265000e+02	1.415000e+02	4.000000	0.000000	
50%	5.059000e+03	108.000000	2.815000e+03	1.600000e+03	49.000000	1.000000	
75%	4.046050e+04	734.000000	2.260600e+04	9.149000e+03	419.500000	6.000000	
max	4.290259e+06	148011.000000	1.846641e+06	2.816444e+06	56336.000000	1076.000000	3

Lets examine the top 5 countries with max confirmed numbers.

In [5]:

```
#max number of confirmed, recovered, deaths and active.
maxConfirmed = df.loc[df["Confirmed"].idxmax()]
maxRecovered = df.loc[df["Recovered"].idxmax()]
maxDeaths = df.loc[df["Deaths"].idxmax()]
maxActive = df.loc[df["Active"].idxmax()]

print("----------Max Confirmed number-------------")
print(maxConfirmed)
print()
print("----------Max Deaths number----------------")
print(maxDeaths)
print()
print("----------Max Recovered number-------------")
print(maxRecovered)
print()
print("----------Max Active number----------------")
print(maxActive)
```

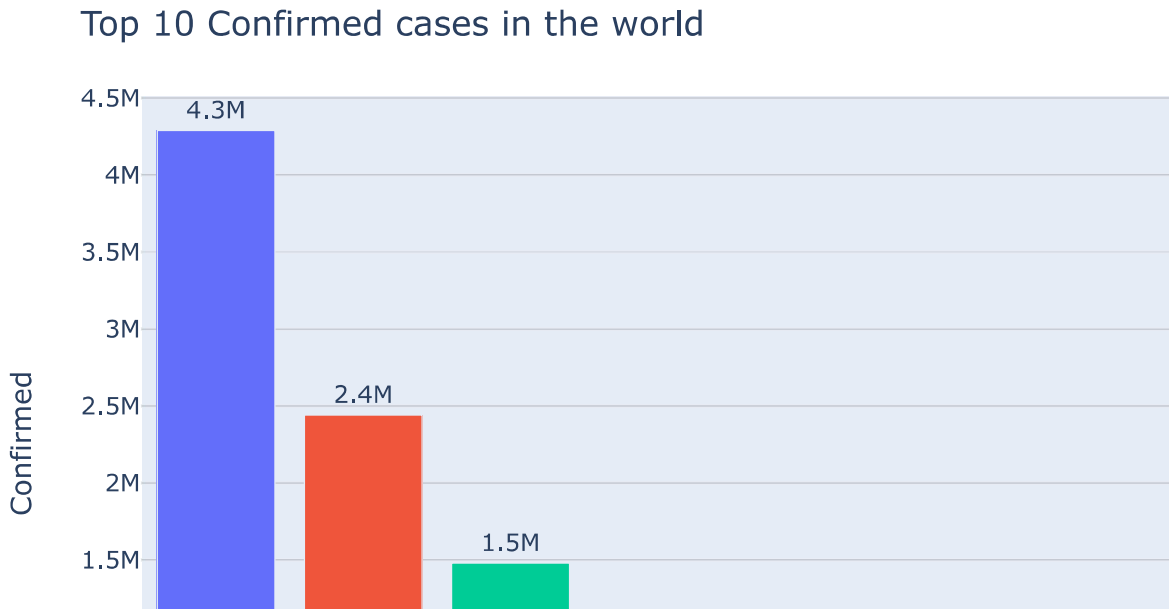
```
----------Max Confirmed number-------------
Country/Region                 US
Confirmed                 4290259
Deaths                     148011
Recovered                 1325804
Active                    2816444
New cases                   56336
New deaths                   1076
New recovered               27941
Deaths / 100 Cases           3.45
Recovered / 100 Cases        30.9
Deaths / 100 Recovered      11.16
Confirmed last week       3834677
1 week change              455582
1 week % increase           11.88
WHO Region               Americas
Name: 173, dtype: object

----------Max Deaths number----------------
Country/Region                 US
Confirmed                 4290259
Deaths                     148011
Recovered                 1325804
Active                    2816444
New cases                   56336
New deaths                   1076
New recovered               27941
Deaths / 100 Cases           3.45
Recovered / 100 Cases        30.9
Deaths / 100 Recovered      11.16
Confirmed last week       3834677
1 week change              455582
1 week % increase           11.88
WHO Region               Americas
Name: 173, dtype: object

----------Max Recovered number-------------
Country/Region              Brazil
Confirmed                 2442375
Deaths                      87618
Recovered                 1846641
```

```
Active                       508116
New cases                     23284
New deaths                      614
New recovered                 33728
Deaths / 100 Cases             3.59
Recovered / 100 Cases         75.61
Deaths / 100 Recovered         4.74
Confirmed last week         2118646
1 week change                323729
1 week % increase             15.28
WHO Region                 Americas
Name: 23, dtype: object

-----------Max Active number-----------------
Country/Region                   US
Confirmed                   4290259
Deaths                       148011
Recovered                   1325804
Active                      2816444
New cases                     56336
New deaths                     1076
New recovered                 27941
Deaths / 100 Cases             3.45
Recovered / 100 Cases          30.9
Deaths / 100 Recovered        11.16
Confirmed last week         3834677
1 week change                455582
1 week % increase             11.88
WHO Region                 Americas
Name: 173, dtype: object
```

Lets visualise the top 10 confirmed countries in descending order

In [6]:

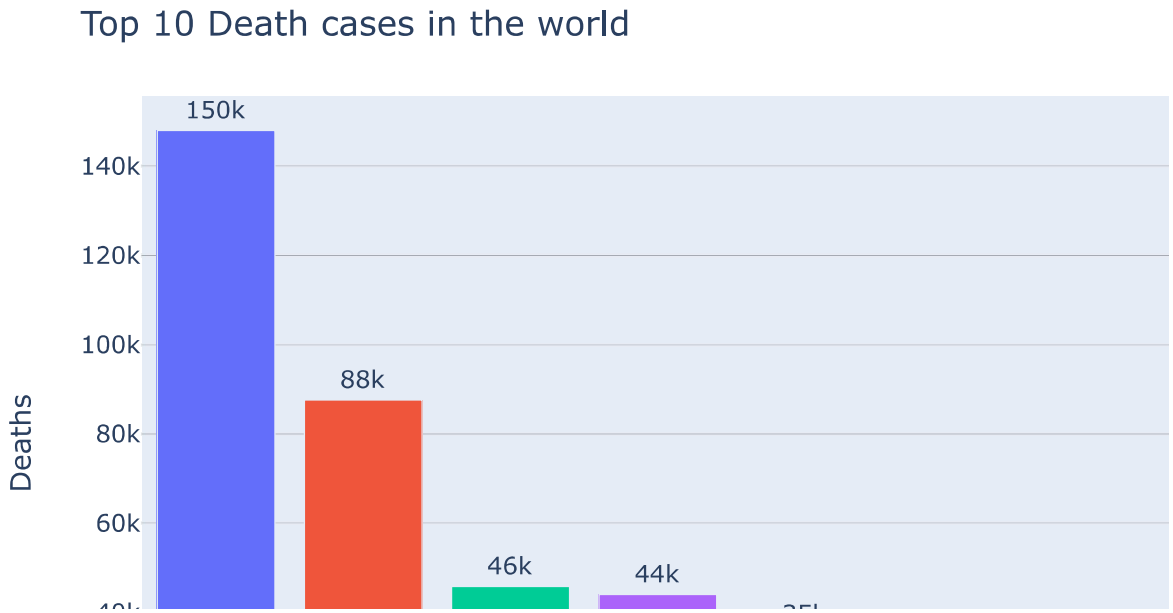
```
#dislay confirmed in first 10 countries
df = df.sort_values("Confirmed", ascending = False)
fig = px.bar(df.head(10), y = "Confirmed", x = "Country/Region",
             text = "Confirmed", color = "Country/Region")
#total values bar with 2 precision values
fig.update_traces(texttemplate = "%{text:.2s}", textposition = "outside")
#set fontsize and uniformText
fig.update_layout(uniformtext_minsize = 8)
#rotate label 45 degrees
fig.update_layout(xaxis_tickangle=-45)
fig.update_layout(legend_title_text='confirmed (Covid-19)')
fig.update_layout(title_text='Top 10 Confirmed cases in the world')
fig

```



Top 10 Deaths reported countries in the world

In [7]:

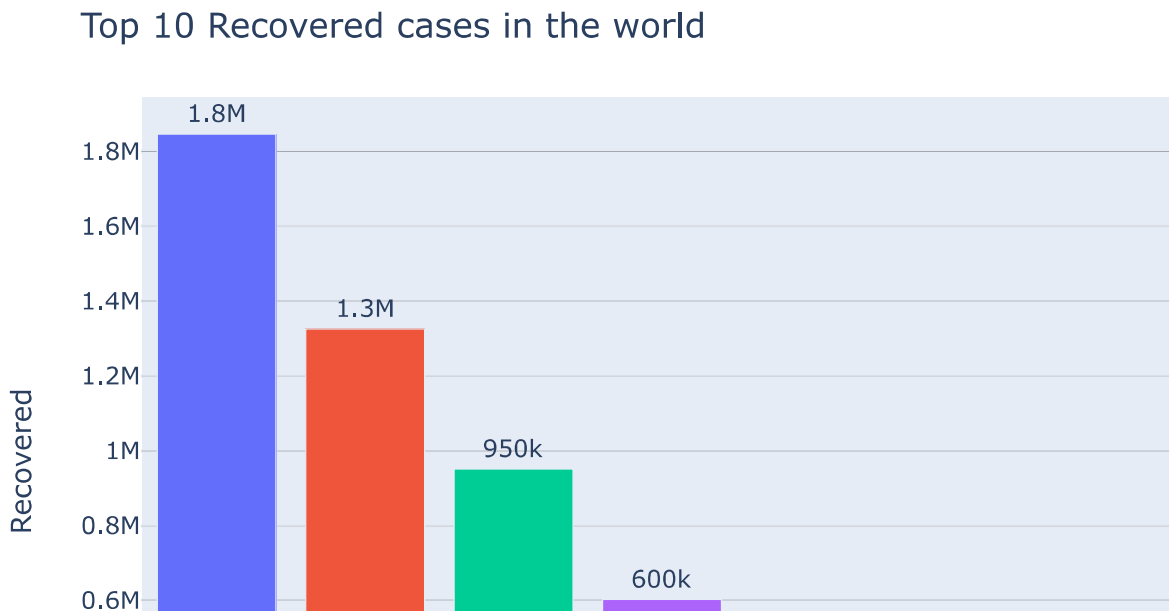
```
#dislay deaths in first 10 countries
df = df.sort_values("Deaths", ascending = False)
fig = px.bar(df.head(10), y = "Deaths", x = "Country/Region",
             text = "Deaths", color = "Country/Region")
#total values bar with 2 precision values
fig.update_traces(texttemplate = "%{text:.2s}", textposition = "outside")
#set fontsize and uniformText
fig.update_layout(uniformtext_minsize = 8)
#rotate label 45 degrees
fig.update_layout(xaxis_tickangle=-45)
fig.update_layout(legend_title_text='Deaths (Covid-19)')
fig.update_layout(title_text='Top 10 Death cases in the world')
fig
```



Top 10 recovered countries in the world

In [8]:

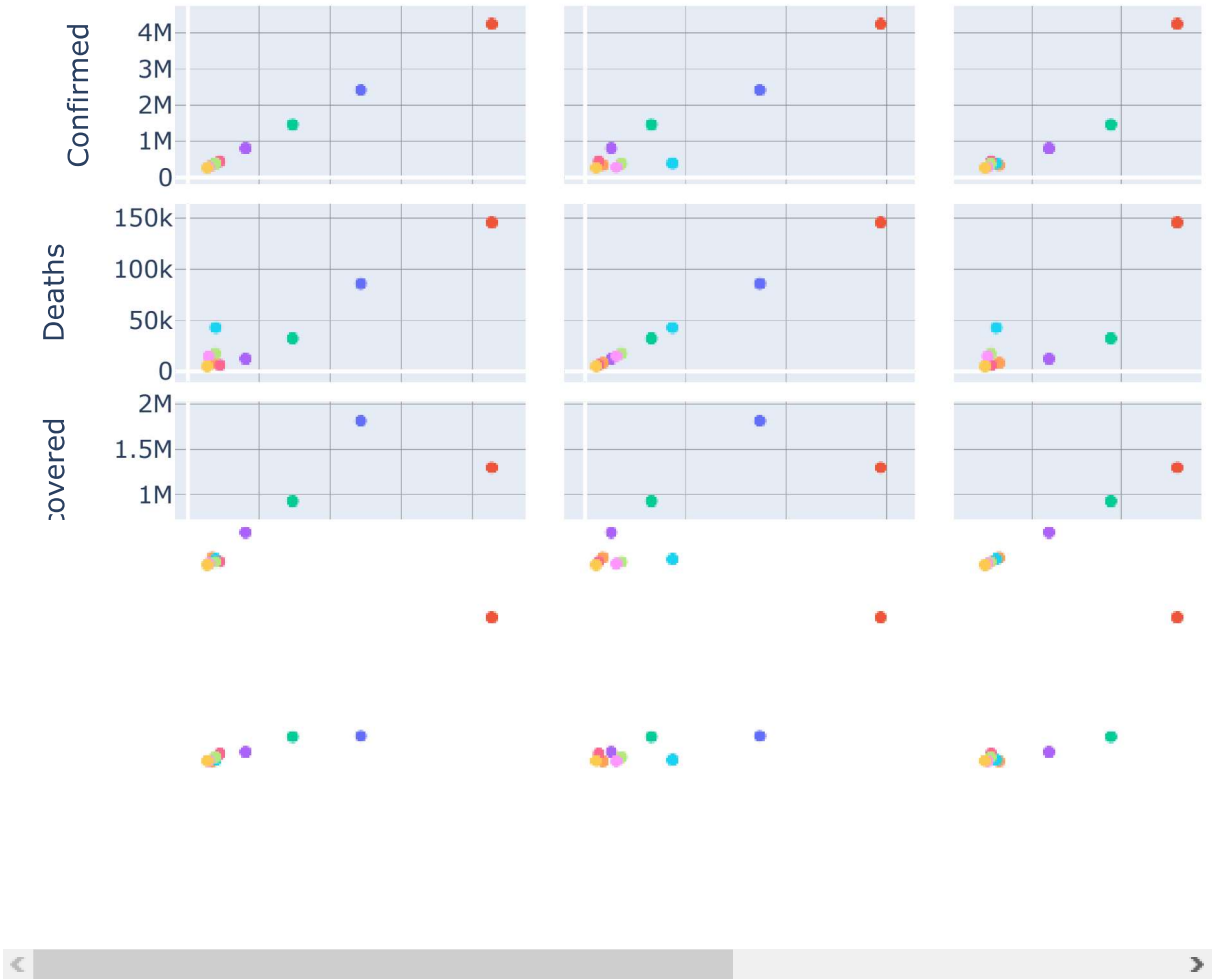
```
#dislay Recovered in first 10 countries
df = df.sort_values("Recovered", ascending = False)
fig = px.bar(df.head(10), y = "Recovered", x = "Country/Region",
             text = "Recovered", color = "Country/Region")
#total values bar with 2 precision values
fig.update_traces(texttemplate = "%{text:.2s}", textposition = "outside")
#set fontsize and uniformText
fig.update_layout(uniformtext_minsize = 8)
#rotate label 45 degrees
fig.update_layout(xaxis_tickangle=-45)
fig.update_layout(legend_title_text='Recovered (Covid-19)')
fig.update_layout(title_text='Top 10 Recovered cases in the world')
fig
```



Scatter matrix plot between the confirmed, deaths, recovered and active cases

In [9]:

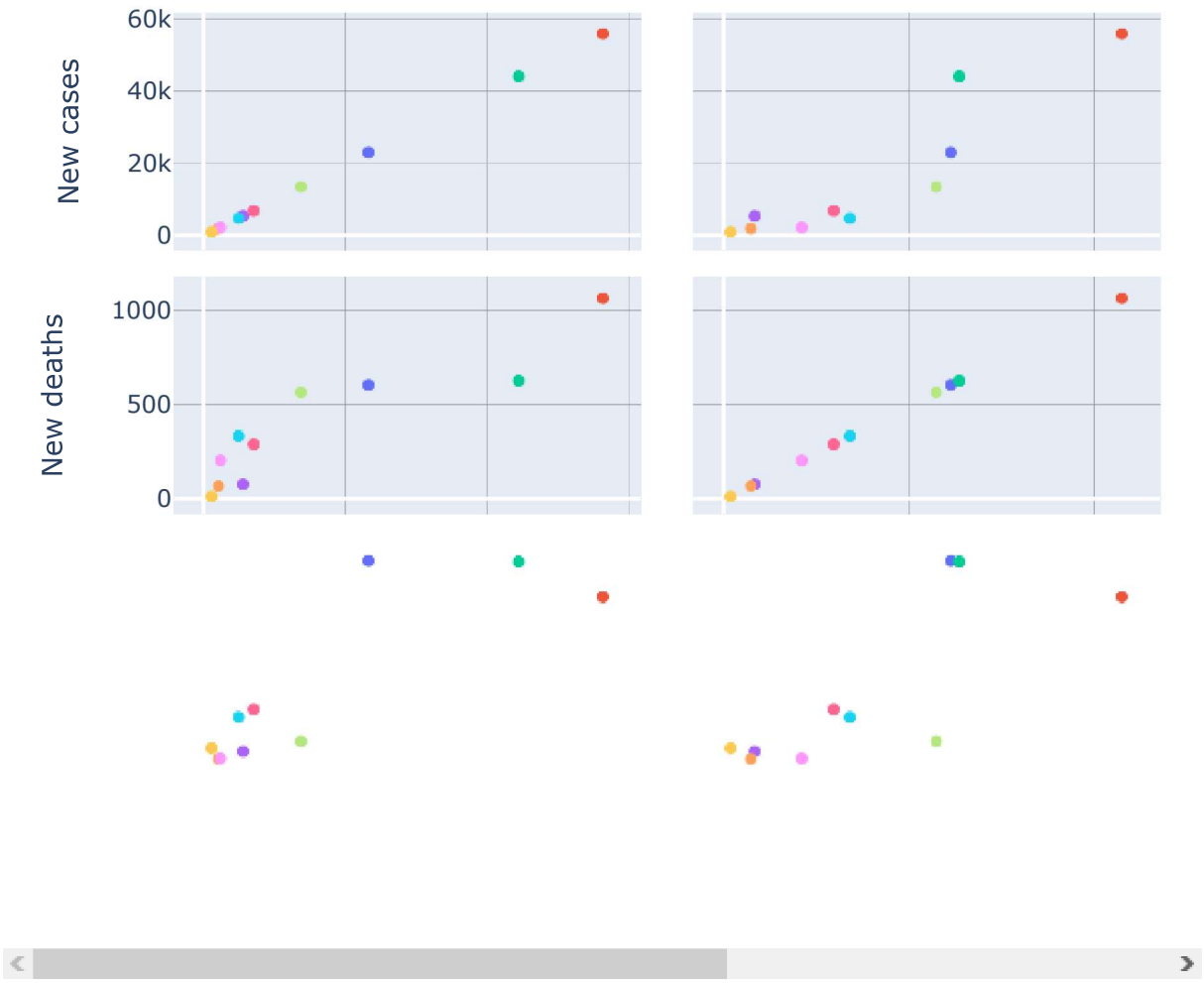
```
#Scatter matrix plot between the confirmed, deaths and recovery category.
fig = px.scatter_matrix(df.head(10), dimensions=["Confirmed","Deaths","Recovered","Acti
fig
```



Scatter matrix of new cases, new deaths and new recovery.

In [10]:

```
#scatter matrix of new cases, new deaths and new recovery.
fig = px.scatter_matrix(df.head(10), dimensions = ["New cases","New deaths","New recove
fig
```



Automatic visualisation of the entire data to find hidden patterns or insight.

In [11]:

```
av = AutoViz_Class()
df = "country_wise_latest.csv"
sep = ","
draftAutoViz = av.AutoViz(
            df,
            sep = ",",
            depVar="",
            dfte = None,
            header = 0,
            verbose = 0,
            lowess = False,
            chart_format = "svg",
            max_cols_analyzed=30,
            max_rows_analyzed=1500000,
            )
```

```
Shape of your Data Set: (187, 15)
############## C L A S S I F Y I N G  V A R I A B L E S  #################
###
Classifying variables in data set...
    Number of Numeric Columns =  4
    Number of Integer-Categorical Columns =  9
    Number of String-Categorical Columns =  1
    Number of Factor-Categorical Columns =  0
    Number of String-Boolean Columns =  0
    Number of Numeric-Boolean Columns =  0
    Number of Discrete String Columns =  0
    Number of NLP String Columns =  0
    Number of Date Time Columns =  0
    Number of ID Columns =  1
    Number of Columns to Delete =  0
    15 Predictors classified...
        This does not include the Target column(s)
        1 variables removed since they were ID or low-information variable
s
```

create a prediction

In [12]:

```
df = pd.read_csv("country_wise_latest.csv")

df = df.sort_values("Confirmed", ascending = False)
df
```

Out[12]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Death / 10 Case
173	US	4290259	148011	1325804	2816444	56336	1076	27941	3.4
23	Brazil	2442375	87618	1846641	508116	23284	614	33728	3.5
79	India	1480073	33408	951166	495499	44457	637	33598	2.2
138	Russia	816680	13334	602249	201097	5607	85	3077	1.6
154	South Africa	452529	7067	274925	170537	7096	298	9848	1.5
...	...	...	...	...	...	...	...	...	
49	Dominica	18	0	18	0	0	0	0	0.0
140	Saint Kitts and Nevis	17	0	15	2	0	0	0	0.0
68	Greenland	14	0	13	1	1	0	0	0.0
75	Holy See	12	0	12	0	0	0	0	0.0
183	Western Sahara	10	1	8	1	0	0	0	10.0

187 rows × 15 columns

In [13]:

```
#split dataset to train and test dataset
x_train, y_train = train_test_split(df, test_size=0.2)
x_test, y_test = train_test_split(df, test_size = 0.2)
print(len(x_train), "x_train Dataset")
print(len(y_train), "y_train Dataset")
print(len(x_test), "x_test Dataset")
print(len(y_test), "y_test Dataset")
```

```
149 x_train Dataset
38 y_train Dataset
149 x_test Dataset
38 y_test Dataset
```

In [14]:

```
x_train = np.array(x_train)
y_train = np.array(y_train)
x_test = np.array(x_test)
y_test = np.array(y_test)

x_train
```

Out[14]:

```
array([['Guinea-Bissau', 1954, 26, ..., 5, 0.26, 'Africa'],
       ['Uzbekistan', 21209, 121, ..., 4060, 23.67, 'Europe'],
       ['Namibia', 1843, 8, ..., 499, 37.13, 'Africa'],
       ...,
       ['Pakistan', 274289, 5842, ..., 8193, 3.08,
        'Eastern Mediterranean'],
       ['Poland', 43402, 1676, ..., 3019, 7.48, 'Europe'],
       ['Czechia', 15516, 373, ..., 1418, 10.06, 'Europe']], dtype=object)
```

In [15]:

```
df = h2o.import_file("country_wise_latest.csv")

df2 = df.sort(1, ascending=False)
df2
```

```
Parse progress: |
100%
```

Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Dea / Ca
US	4.29026e+06	148011	1.3258e+06	2.81644e+06	56336	1076	27941	3
Brazil	2.44238e+06	87618	1.84664e+06	508116	23284	614	33728	3
India	1.48007e+06	33408	951166	495499	44457	637	33598	2
Russia	816680	13334	602249	201097	5607	85	3077	1
South Africa	452529	7067	274925	170537	7096	298	9848	1
Mexico	395489	44022	303810	47657	4973	342	8588	1
Peru	389717	18418	272547	98752	13756	575	4697	4
Chile	347923	9187	319954	18782	2133	75	1859	2
United Kingdom	301708	45844	1437	254427	688	7	3	15
Iran	293606	15912	255144	22550	2434	212	1931	5

Out[15]:

In [16]:

```
train,test,valid = df2.split_frame(ratios=[.7, .15])

x = train.columns
y = "Confirmed"
x.remove(y)



train[y] = train[y].asfactor()
test[y] = test[y].asfactor()

aml = H2OAutoML(max_models=20, seed = 1)
aml.train(x=x,y=y,training_frame=train)
```

```
AutoML progress: |
15:11:24.398: AutoML: XGBoost is not available; skipping it.

[redacted]
15:13:07.440: Skipping training of model GBM_5_AutoML_20210523_151124 due to
exception: water.exceptions.H2OModelBuilderIllegalArgumentException: Illegal
argument(s) for GBM model: GBM_5_AutoML_20210523_151124.  Details: ERRR on f
ield: _min_rows: The dataset size is too small to split for min_rows=100.0:
must have at least 200.0 (weighted) rows, but have only 119.0.


[redacted]
15:19:34.297: StackedEnsemble_BestOfFamily_AutoML_20210523_151124 [StackedEn
semble best (built using top model from each algorithm type)] failed: java.l
ang.RuntimeException: java.lang.ArrayIndexOutOfBoundsException: 591


15:19:36.331: StackedEnsemble_AllModels_AutoML_20210523_151124 [StackedEnsem
ble all (built using all AutoML models)] failed: java.lang.RuntimeException:
java.lang.ArrayIndexOutOfBoundsException: 2361

[redacted]| 100%
```

In [17]:

```
lb = aml.leaderboard
lb.head(rows=lb.nrows)
```

model_id	mean_per_class_error	logloss	rmse
DeepLearning_1_AutoML_20210523_151124	0.991525	6.20505	0.995354
DeepLearning_grid__2_AutoML_20210523_151124_model_2	0.991525	18.1746	0.996047
GBM_1_AutoML_20210523_151124	0.991525	11.5482	0.990043
GBM_grid__1_AutoML_20210523_151124_model_6	0.991525	4.73023	0.985257
GBM_grid__1_AutoML_20210523_151124_model_5	1	5.21482	0.993694
GBM_grid__1_AutoML_20210523_151124_model_2	1	16.8994	0.994507
DeepLearning_grid__2_AutoML_20210523_151124_model_1	1	12.8504	0.999828
GBM_grid__1_AutoML_20210523_151124_model_3	1	4.59093	0.987039
GBM_3_AutoML_20210523_151124	1	4.79748	0.98762
DeepLearning_grid__3_AutoML_20210523_151124_model_2	1	19.461	0.999996
GBM_grid__1_AutoML_20210523_151124_model_1	1	22.1782	0.993019
DeepLearning_grid__3_AutoML_20210523_151124_model_1	1	11.7023	0.999749
DeepLearning_grid__1_AutoML_20210523_151124_model_1	1	12.8702	0.99964
GBM_4_AutoML_20210523_151124	1	4.79189	0.987346
DRF_1_AutoML_20210523_151124	1	34.0008	0.998522
DeepLearning_grid__1_AutoML_20210523_151124_model_2	1	15.1482	0.999956
GLM_1_AutoML_20210523_151124	1	20.0178	0.999825
GBM_2_AutoML_20210523_151124	1	4.79221	0.98755
XRT_1_AutoML_20210523_151124	1	34.0167	0.999229
GBM_grid__1_AutoML_20210523_151124_model_4	1	4.67724	0.987359

Out[17]:

In [18]:

```
test = test.drop("Confirmed")
test
predictions = aml.leader.predict(test)
```

```
deeplearning prediction progress: |
100%
```

In [19]:

```
1 predictions.describe()
```

```
Rows:36
Cols:119
```

	predict	p10	p17	p18	
type	enum	real	real	real	
mins		0.0001635309202609677	0.00661551470655862	0.0002836714388956398	0.000
mean		0.008649500961384735	0.008102108121068645	0.009856949869950662	0.00
maxs		0.010652713094576738	0.013340946614035601	0.014392778655689433	0.01
sigma		0.0018200699434186916	0.00115351036528011	0.0023084134086096297	0.001
zeros		0	0	0	
missing	0	0	0	0	
0	45309	0.0001635309202609677	0.013340946614035601	0.0002836714388956398	0.000
1	18	0.005748725093107088	0.00762718067972786	0.014392778655689433	0.00
2	4548	0.007320875055427739	0.007756135180213147	0.013214322729023403	0.0
3	701	0.010257462572932993	0.0073860848428042815	0.009371200650917172	0.00
4	4599	0.009719109409275868	0.008277350141375952	0.009833570662447414	0.00
5	10621	0.009993711739429931	0.007608825868532975	0.012807158824680637	0.01
6	10621	0.008244337801454116	0.007372844595787201	0.012212652273807578	0.0
7	4599	0.010652713094576738	0.007855823839581484	0.010510809189887647	0.00
8	4548	0.0092108850191359	0.008089744863513727	0.010720320590857613	0.0
9	701	0.009904732945779982	0.007407866049795423	0.009093155854929414	0.01