In [1]:

```
#!pip install plotly==5.6.0
```

In [2]:

```
import numpy as np
import pandas as pd
```

In [3]:

```
df = pd.read_excel("gdp_data.xlsx",header=1)  # 246 countries, 57 years (1960-2016)
df.head()
```

Out[3]:

```
       Country  Code    Year  GDP-Growth  GDP-Per-Capita  GDP
0        Aruba   ABW  1960.0         NaN             NaN  NaN
1  Afghanistan   AFG  1960.0         NaN             NaN  NaN
2       Angola   AGO  1960.0         NaN             NaN  NaN
3      Albania   ALB  1960.0         NaN             NaN  NaN
4      Andorra   AND  1960.0         NaN             NaN  NaN
```

In [4]:

```
data = pd.pivot_table(df, index="Country", columns="Year", values="GDP-Growth",fill_value=0
data.head(5).transpose()
```

Out[4]:

Country	Afghanistan	Albania	Algeria	American Samoa	Andorra
Year					
1960.0	0.000000	0.000000	-19.685042	0.000000	0.000000
1961.0	0.000000	0.000000	34.313729	0.000000	0.000000
1962.0	0.000000	0.000000	5.839413	0.000000	0.000000
1963.0	0.000000	0.000000	6.206898	0.000000	0.000000
1964.0	0.000000	0.000000	-4.804971	0.000000	0.000000
1965.0	0.000000	0.000000	9.452963	0.000000	0.000000
1966.0	0.000000	0.000000	10.796239	0.000000	0.000000
1967.0	0.000000	0.000000	8.433280	0.000000	0.000000
1968.0	0.000000	0.000000	8.862657	0.000000	0.000000
1969.0	0.000000	0.000000	-11.331719	0.000000	4.649465
1970.0	0.000000	0.000000	27.423969	0.000000	8.149743
1971.0	0.000000	0.000000	3.813176	0.000000	7.788467
1972.0	0.000000	0.000000	7.494918	0.000000	5.618790
1973.0	0.000000	0.000000	5.045342	0.000000	0.542206
1974.0	0.000000	0.000000	8.386756	0.000000	3.303787
1975.0	0.000000	0.000000	5.258586	0.000000	2.838576
1976.0	0.000000	0.000000	9.214836	0.000000	1.463000
1977.0	0.000000	0.000000	7.477827	0.000000	0.041557
1978.0	0.000000	0.000000	0.790607	0.000000	2.208728
1979.0	0.000000	5.745635	2.999996	0.000000	-0.132475
1980.0	0.000000	2.948597	6.400004	0.000000	1.246461
1981.0	0.000000	1.104938	5.400003	0.000000	1.770118
1982.0	0.000000	-1.251597	5.599997	0.000000	1.784687
1983.0	0.000000	1.780644	3.699997	0.000000	2.321433
1984.0	0.000000	5.637243	0.400001	0.000000	3.253322
1985.0	0.000000	-0.787843	-0.699998	0.000000	5.547122
1986.0	0.000000	-1.420040	-1.000005	0.000000	5.094326
1987.0	0.000000	9.836549	4.400002	0.000000	4.827034
1988.0	0.000000	-9.575640	0.800001	0.000000	3.781388
1989.0	0.000000	-28.002142	-1.200001	0.000000	2.546003
1990.0	0.000000	-7.187111	1.800002	0.000000	0.929212
1991.0	0.000000	9.559412	-2.100001	0.000000	-1.031484
1992.0	0.000000	8.302867	-0.899997	0.000000	2.383188

Country	Afghanistan	Albania	Algeria	American Samoa	Andorra
Year					
1993.0	0.000000	13.322333	3.799995	0.000000	2.757499
1994.0	0.000000	9.099999	4.099998	0.000000	4.649740
1995.0	0.000000	-10.919984	1.100000	0.000000	9.067672
1996.0	0.000000	8.830088	5.100004	0.000000	3.194790
1997.0	0.000000	12.889897	3.200002	0.000000	4.099081
1998.0	0.000000	6.950036	3.819678	0.000000	3.528361
1999.0	0.000000	8.290070	3.008395	0.000000	4.546768
2000.0	0.000000	4.539961	5.609323	0.000000	6.471015
2001.0	8.832278	5.530051	7.201872	0.814111	12.168720
2002.0	1.414118	5.509999	4.301624	0.538358	7.647870
2003.0	11.229715	5.529915	5.907791	-0.401606	7.396983
2004.0	5.357403	5.900084	1.684488	-4.166667	4.536353
2005.0	13.826320	5.979982	3.372875	1.963534	0.040011
2006.0	3.924984	7.499970	2.360135	-2.613480	-8.590004
2007.0	21.390528	3.349994	1.632244	-4.237288	-3.690654
2008.0	14.362441	3.706881	3.634145	0.442478	-5.358826
2009.0	0.426355	2.545405	2.891866	0.293686	-4.646543
2010.0	12.752287	1.417526	3.374769	-4.392387	-1.615218
2011.0	5.600745	1.001988	2.767639	-2.756508	0.351645
2012.0	2.724543	1.770000	3.789121	0.944882	2.277683
2013.0	1.451315	2.230000	3.763467	1.248050	0.842204
2014.0	2.260314	3.350000	3.300000	-2.619414	1.889124
2015.0	2.665292	3.836620	1.600000	-5.379747	1.724022
2016.0	1.030660	4.004413	2.100000	0.000000	1.629345

In [5]:

```
pd.set_option('display.max_rows', 500)
x=data.transpose().describe()
x.transpose()
```

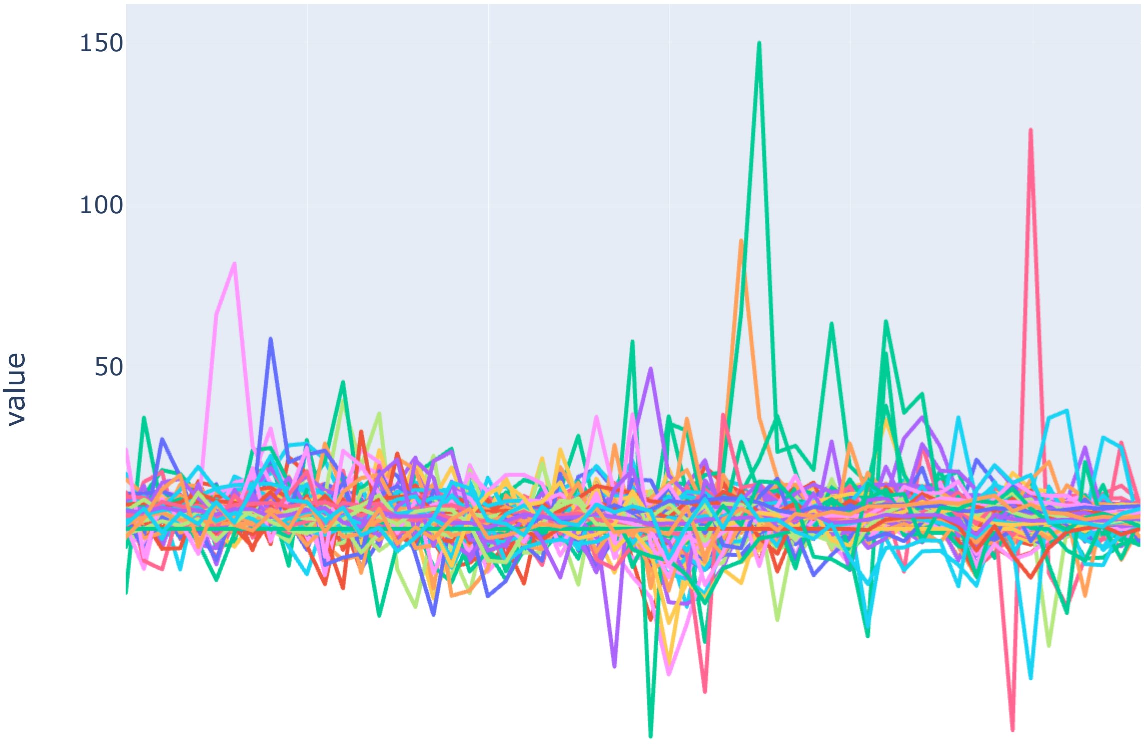
Out[5]:

	count	mean	std	min	25%	50%	75%	max
Country								
Afghanistan	57.0	1.916654	4.457751	0.000000	0.000000	0.000000	1.030660	21.390528
Albania	57.0	1.979943	6.134441	-28.002142	0.000000	1.417526	5.530051	13.322333
Algeria	57.0	4.070332	7.048594	-19.685042	1.684488	3.763467	5.839413	34.313729
American Samoa	57.0	-0.356526	1.371241	-5.379747	0.000000	0.000000	0.000000	1.963534
Andorra	57.0	2.207760	3.565348	-8.590004	0.000000	1.889124	4.536353	12.168720
Angola	57.0	2.533637	6.103642	-23.983417	0.000000	0.858713	4.822626	15.028915
Antigua and Barbuda	57.0	2.767584	4.458639	-12.107536	0.000000	3.032606	5.589741	12.728507
Arab World	57.0	2.856160	4.169774	-9.072763	0.000000	2.455201	4.989372	15.820334

In [6]:

```
import plotly.express as px
df = data.transpose()
fig = px.line(df)
fig.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\plotly\express\_core.py:1222: Per
formanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead.  To get a de-fr
agmented frame, use `newframe = frame.copy()`
  df_output[col_name] = to_unindexed_series(df_input[argument])
```



In [7]:

```
from sklearn.ensemble import IsolationForest
clf = IsolationForest().fit(data)
pred = clf.predict(data)
data['anomaly']=pred
outliers = data.loc[data['anomaly']==-1]
outlier_index = list(outliers.index)
outlier_index
```

Out[7]:

```
['China', 'Equatorial Guinea', 'Iraq', 'Oman']
```

In [8]:

```
from sklearn.neighbors import LocalOutlierFactor
clf = LocalOutlierFactor(n_neighbors=200,novelty=True).fit(data)
pred = clf.predict(data)
data['anomaly']=pred
outliers = data.loc[data['anomaly']==-1]
outlier_index = list(outliers.index)
outlier_index
```

Out[8]:

```
['Bosnia and Herzegovina',
 'Equatorial Guinea',
 'Iraq',
 'Lebanon',
 'Libya',
 'Oman',
 'Rwanda',
 'Saudi Arabia',
 'Timor-Leste']
```

In [9]:

```
from sklearn.svm import OneClassSVM
clf = OneClassSVM().fit(data)
pred = clf.predict(data)
data['anomaly']=pred
outliers = data.loc[data['anomaly']==-1]
outlier_index = list(outliers.index)
len(outlier_index)
#outlier_index
```

Out[9]:

```
124
```

In [ ]:

```

```