Handling Outliers

1. What is outliers?

Outliers is anomaly data that out of the range. For example in high school, the age of student has range between 14-18 years old. One event of the data show the age is 23 years old. This event could be an outlier. It might happen because wrong imputation or dataset error.

To handling this event, we may to change the value or delete this event. For fewer dataset it might to change the value into some value (could be mean, mode, median, quantile, or even another value during in range). While for larger dataset the event could be deleted.

2. Could outliers be affected the data?

The answer is yes. Absolutely yes, because it may affect another data. At machine learning process, it would like to interrupt the model and might the error can't be tolerance.

3. How to handling outliers?

At this practice, i'll show you to handling an outlier at World happiness 2018 dataset. You can access the dataset through this link

(https://www.kaggle.com/unsdsn/world-happiness?select=2018.csv)

The dataset contains 9 columns, you can see it below;

```
print(df.head())
```

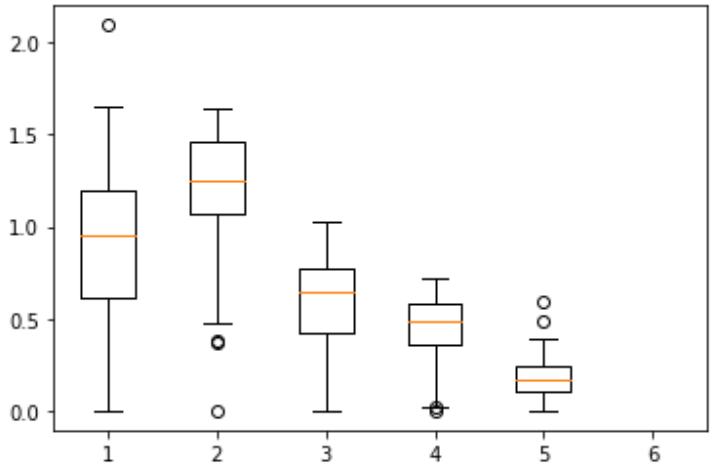
```
   Overall rank Country or region   Score  GDP per capita  Social support  \
0        0.1365             0.1365  0.1365          0.1365          0.1365
1        0.1365             0.1365  0.1365          0.1365          0.1365
2        0.1365             0.1365  0.1365          0.1365          0.1365
3        0.2030              0.203  0.2030          0.2030          0.2030
4        0.1365             0.1365  0.1365          0.1365          0.1365

   Healthy life expectancy  Freedom to make life choices  Generosity  \
0                   0.1365                        0.1365      0.1365
1                   0.1365                        0.1365      0.1365
2                   0.1365                        0.1365      0.1365
3                   0.2030                        0.2030      0.2030
4                   0.1365                        0.1365      0.1365

   Perceptions of corruption
0                     0.1365
1                     0.1365
2                     0.1365
3                     0.2030
4                     0.1365
```

First method, we have to look which the data has outliers, by using boxplot from matplotlib

```
import matplotlib.pyplot as plt
plt.figure()
plt.gca()
data = [df['GDP per capita'], df['Social support'], df['Healthy life expectancy'],
        df['Freedom to make life choices'], df['Generosity'], df['Perceptions of corruption']]
plt.boxplot(data)
plt.show()
```



The result shown that the GDP per capita, Social support, Freedom to make life choices, generosity has an outliers. And the Perception of corruption can't display its boxplot, it may has a NaN value.

First thing we have to handle this GDP per capita. We have to know the statistic of this data such as quantile, iqr, range.

The data could be outlier if this data out of these range;

1. If the data more than *Q3 + 1.5*Iqr*
2. If the data less than *Q1 – 1.5*Iqr*

The Iqr itself is *Q3 – Q1*

```
median = df['GDP per capita'].median()
q3 =  df['GDP per capita'].quantile(0.75)
minn = df['GDP per capita'].min()
maxx = df['GDP per capita'].max()
iqr = df['GDP per capita'].quantile(0.75) - df['GDP per capita'].quantile(0.25)
o3 = df['GDP per capita'].quantile(0.75) + 1.5*iqr
o1 = df['GDP per capita'].quantile(0.25) - 1.5*iqr
print(median)
print(q3)
print(minn)
print(maxx)
print(o1)
print(o3)
```

```
0.9495
1.19775
0.0
2.096
-0.2560000000000002
2.0700000000000003
```

After we have this statistic information, we have to look where the outliers in by knowing the range.

```
print(o1)
df1 = df[df['GDP per capita'] < o1]
df2 = df[df['GDP per capita'] > o3]
print('Less than o1')
print(df1.head())
print('Amount: ', df.shape)
print('')
print('More than o3')
print(df2.head())
```

```
-0.2560000000000002
Less than o1
Empty DataFrame
Columns: [Overall rank, Country or region, Score, GDP per capita, Social support,
choices, Generosity, Perceptions of corruption]
Index: []
Amount:  (156, 9)

More than o3
    Overall rank     Country or region  Score  GDP per capita  Social support  \
19            20  United Arab Emirates  6.774           2.096           0.776

    Healthy life expectancy  Freedom to make life choices  Generosity  \
19                     0.67                         0.284       0.186

    Perceptions of corruption
19                        NaN
```

The result of these code shown, the outliers come from this UAE country that has highest GDP. We can replace this value into *Q3* for safe value.

```
df['GDP per capita'] = df['GDP per capita'].replace(2.096, 1.197)
df3 = df[df['Overall rank'] == 20]
print(df3)
```

```
    Overall rank     Country or region  Score  GDP per capita  Social support  \
19            20  United Arab Emirates  6.774           1.197           0.776

    Healthy life expectancy  Freedom to make life choices  Generosity  \
19                     0.67                         0.284       0.186

    Perceptions of corruption
19                        NaN
```

At this row UAE data, there is NaN value at Perception of corruption column that we have to change the value into mean, median, or, mode. At this time, we have to change this NaN into mean value of its column.

```
import numpy as np
df.iloc[:,-1].fillna(df.iloc[:,-1].mean(), inplace=True)
df3 = df[df['Overall rank'] == 20]
print(df3)
```

```
    Overall rank     Country or region  Score  GDP per capita  Social support
19            20  United Arab Emirates  6.774           1.197           0.776

    Healthy life expectancy  Freedom to make life choices  Generosity  \
19                     0.67                         0.284       0.186

    Perceptions of corruption
19                      0.112
```

After we handle the NaN, we have to hanlde others outliers in another column. This is the example of how to handle the outliers from Perception of corruption.

```
iqr = df.iloc[:,-1].quantile(0.75) - df.iloc[:,-1].quantile(0.25)
o3 = df.iloc[:,-1].quantile(0.75) + 1.5*iqr
q3 = df.iloc[:,-1].quantile(0.75)

outliers = (df.iloc[:,-1] > o3)
df[outliers] = np.nan
df.fillna(q3, inplace=True)
df2 = df[df.iloc[:,-1] > o3]
print(df2)
```

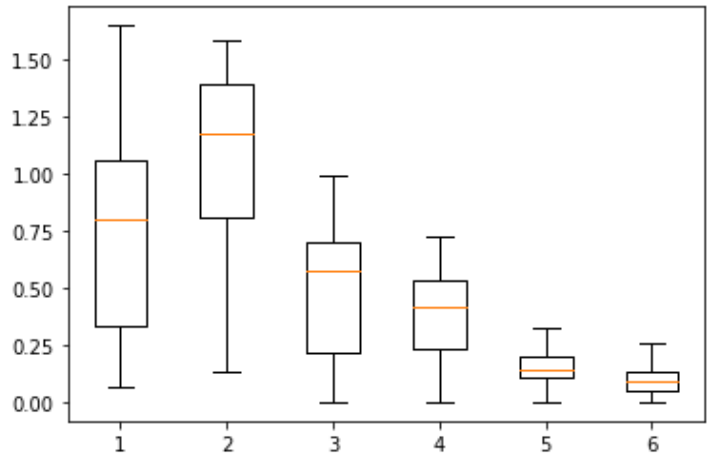
```
Empty DataFrame
Columns: [Overall rank, Country or region, Score, GDP per capita, Social support,
choices, Generosity, Perceptions of corruption]
Index: []
```

At Perceptions of corruption column, the outliers come when the data more than *Q3 + 1.5*Iqr.* So, we have to make a condition statement. After we knew this conditional, the outliers could replace become NaN value. To change the NaN value into another value we can use the .fillna() function. This function could process the NaN value. For replacement the NaN of the outliers, we can use the *Q3* as shown above. After we replace this outliers. The outlier can't find anymore by the result shown that the DataFrame has Empty.

After we handle this perception of corruption column, we have to handle another column. The code as same as like above, but the conditional statement could be different depends on where the oultiers in. And the replacment value depends on the conditional statement, at this practice we use the *Q3* and *Q1* for handling value.

After we handle all column, we have to plot the boxplot to see after outliers handling.

```
import matplotlib.pyplot as plt
plt.figure()
plt.gca()
data = [df['GDP per capita'], df['Social support'], df['Healthy life expectancy'],
        df['Freedom to make life choices'], df['Generosity'], df['Perceptions of corruption']]
plt.boxplot(data)
plt.show()
```



The result show that after we handle the outliers. The graph of boxplot are clean. Thus, this data could be futher process in machine learning