Discover outliers with visualization tools

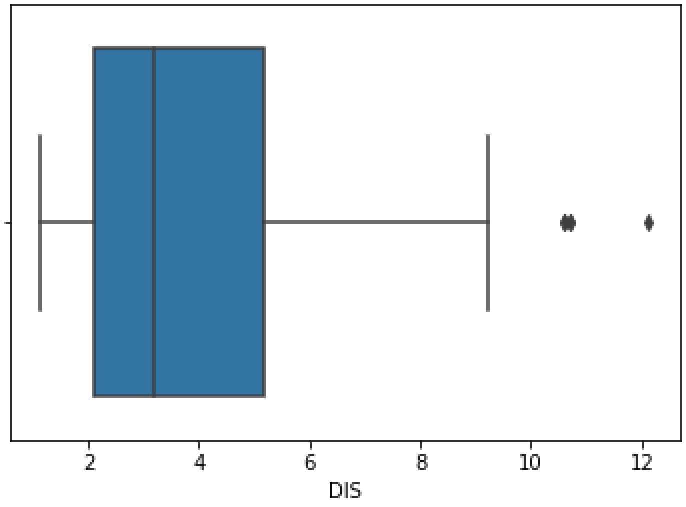
Box plot-

Wikipedia Definition,

> *In descriptive statistics, a* ***box plot*** *is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have* ***lines extending vertically*** *from the boxes (*whiskers*)* ***indicating variability*** *outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram.* ***Outliers*** *may be* ***plotted*** *as* ***individual*** *points.*

Above definition suggests, that if there is an outlier it will plotted as point in boxplot but other population will be grouped together and display as boxes. Let's try and see it ourselves.

```
import seaborn as sns
sns.boxplot(x=boston_df['DIS'])
```



Boxplot — Distance to Employment Center

Above plot shows three points between 10 to 12, these are outliers as there are not included in the box of other observation i.e no where near the quartiles.

Here we analysed Uni-variate outlier i.e. we used DIS column only to check the outlier. But we can do multivariate outlier analysis too. Can we do the multivariate analysis with Box plot? Well it depends, if you have a categorical values then you can use that with any continuous variable and do multivariate outlier analysis. As we do not have categorical value in our Boston Housing dataset, we might need to forget about using box plot for multivariate outlier analysis.

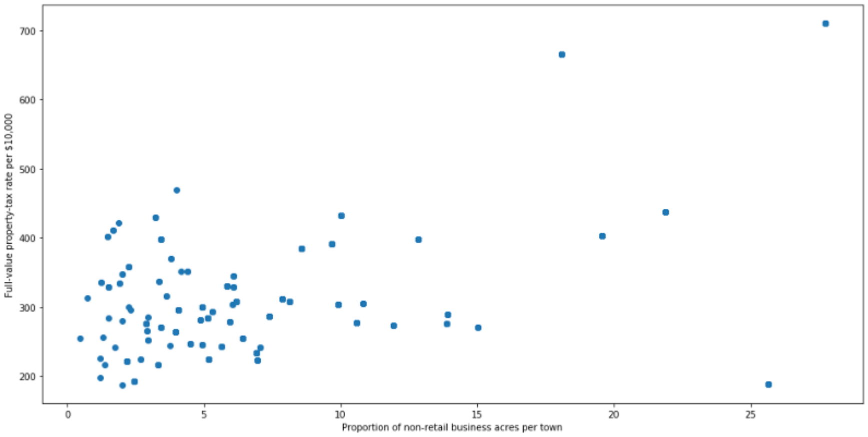
Scatter plot-

Wikipedia Defintion

> *A **scatter plot** , is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. The data are displayed as a **collection of points**, each having the value of **one variable** determining the position on the **horizontal** axis and the value of the **other variable** determining the position on the **vertical** axis.*

As the definition suggests, the scatter plot is the collection of points that shows values for two variables. We can try and draw scatter plot for two variables from our housing dataset.

```
fig, ax = plt.subplots(figsize=(16,8))
ax.scatter(boston_df['INDUS'], boston_df['TAX'])
ax.set_xlabel('Proportion of non-retail business acres per
town')
ax.set_ylabel('Full-value property-tax rate per $10,000')
plt.show()
```



Scatter plot—Proportion of non-retail business acres per town v/s Full value property tax

Looking at the plot above, we can most of data points are lying bottom left side but there are points which are far from the population like top right corner.

Z-Score-

Wikipedia Definition

> *The **Z-score** is the signed number of standard deviations by which the value of an observation or data point is above the mean value of what is being observed or measured.*

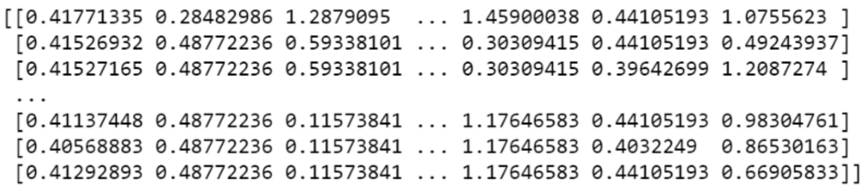
The intuition behind Z-score is to describe any data point by finding their relationship with the Standard Deviation and Mean of the group of data points. Z-score is finding the distribution of data where mean is 0 and standard deviation is 1 i.e. normal distribution.

You must be wondering that, how does this help in identifying the outliers? Well, while calculating the Z-score we re-scale and center the data and look for data points which are too far from zero. These data points which are way too far from zero will be treated as the outliers. In most of the cases a threshold of 3 or -3 is used i.e if the Z-score value is greater than or less than 3 or -3 respectively, that data point will be identified as outliers.

We will use Z-score function defined in scipy library to detect the outliers.

```
from scipy import stats
import numpy as np


z = np.abs(stats.zscore(boston_df))
print(z)
```



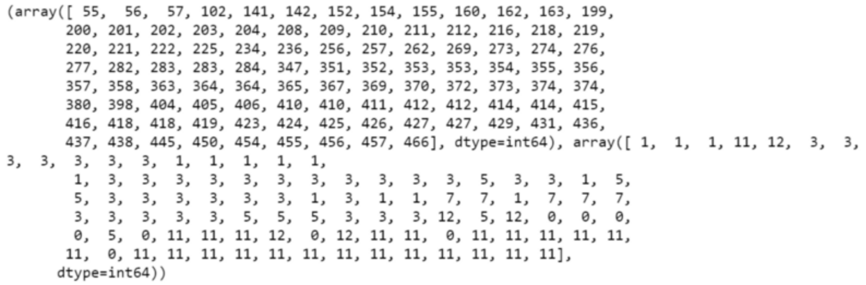
```
[[0.41771335 0.28482986 1.2879095  ... 1.45900038 0.44105193 1.0755623 ]
 [0.41526932 0.48772236 0.59338101 ... 0.30309415 0.44105193 0.49243937]
 [0.41527165 0.48772236 0.59338101 ... 0.30309415 0.39642699 1.2087274 ]
 ...
 [0.41137448 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.98304761]
 [0.40568883 0.48772236 0.11573841 ... 1.17646583 0.4032249  0.86530163]
 [0.41292893 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.66905833]]
```

Z-score of Boston Housing Data

Looking the code and the output above, it is difficult to say which data point is an outlier. Let's try and define a threshold to identify an outlier.

```
threshold = 3
print(np.where(z > 3))
```

This will give a result as below -



```
(array([ 55,  56,  57, 102, 141, 142, 152, 154, 155, 160, 162, 163, 199,
       200, 201, 202, 203, 204, 208, 209, 210, 211, 212, 216, 218, 219,
       220, 221, 222, 225, 234, 236, 256, 257, 262, 269, 273, 274, 276,
       277, 282, 283, 283, 284, 347, 351, 352, 353, 353, 354, 355, 356,
       357, 358, 363, 364, 364, 365, 367, 369, 370, 372, 373, 374, 374,
       380, 398, 404, 405, 406, 410, 410, 411, 412, 412, 414, 414, 415,
       416, 418, 418, 419, 423, 424, 425, 426, 427, 427, 429, 431, 436,
       437, 438, 445, 450, 454, 455, 456, 457, 466], dtype=int64), array([ 1,  1,  1, 11, 12,  3,  3,
3,  3,  3,  3,  3,  1,  1,  1,  1,  1,
        1,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  5,  3,  3,  1,  5,
        5,  3,  3,  3,  3,  3,  3,  1,  3,  1,  1,  7,  7,  1,  7,  7,  7,
        3,  3,  3,  3,  3,  5,  5,  5,  3,  3,  3, 12,  5, 12,  0,  0,  0,
        0,  5,  0, 11, 11, 11, 12,  0, 12, 11, 11,  0, 11, 11, 11, 11, 11,
       11,  0, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11],
      dtype=int64))
```

Data points where Z-scores is greater than 3

Don't be confused by the results. The first array contains the list of row numbers and second array respective column numbers, which mean z[55][1] have a Z-score higher than 3.

```
print(z[55][1])

3.375038763517309
```

So, the data point—55th record on column ZN is an outlier.

IQR score -

Box plot use the IQR method to display data and outliers(shape of the data) but in order to be get a list of identified outlier, we will need to use the mathematical formula and retrieve the outlier data.

Wikipedia Definition

> *The **interquartile range (IQR)**, also called the **midspread** or **middle 50%**, or technically **H-spread**, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, $IQR = Q3 - Q1$.*

> *In other words, the IQR is the first quartile subtracted from the third quartile; these quartiles can be clearly seen on a box plot on the data.*

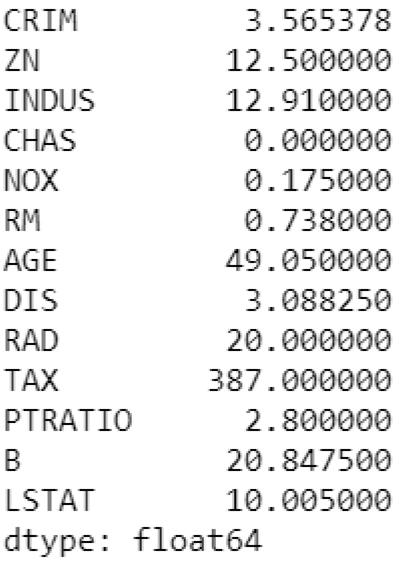
> *It is a measure of the dispersion similar to standard deviation or variance, but is much more robust against outliers.*

IQR is somewhat similar to Z-score in terms of finding the distribution of data and then keeping some threshold to identify the outlier.

Let’s find out we can box plot uses IQR and how we can use it to find the list of outliers as we did using Z-score calculation. First we will calculate IQR,

```
Q1 = boston_df_o1.quantile(0.25)
Q3 = boston_df_o1.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

Here we will get IQR for each column.

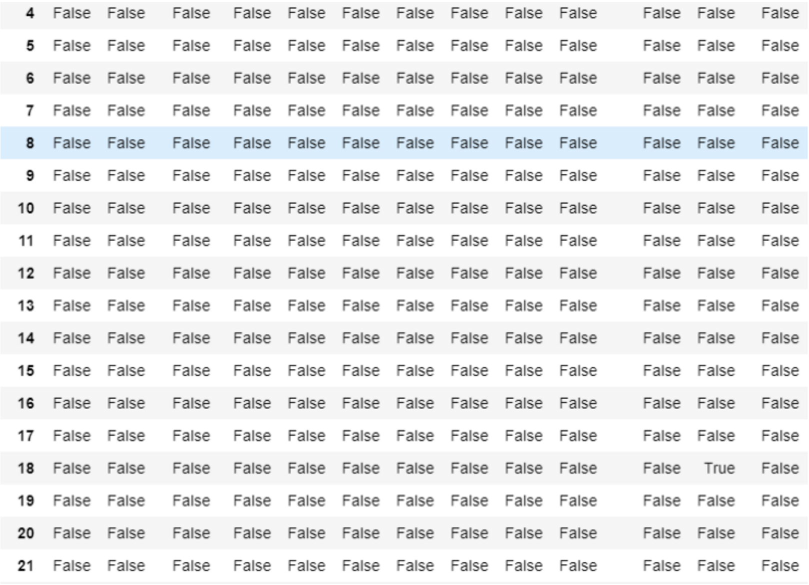


```
CRIM          3.565378
ZN           12.500000
INDUS        12.910000
CHAS          0.000000
NOX           0.175000
RM            0.738000
AGE          49.050000
DIS           3.088250
RAD          20.000000
TAX         387.000000
PTRATIO       2.800000
B            20.847500
LSTAT        10.005000
dtype: float64
```

IQR for each column

As we now have the IQR scores, it’s time to get hold on outliers. The below code will give an output with some true and false values. The data point where we have False that means these values are valid whereas True indicates presence of an outlier.

```
print(boston_df_o1 < (Q1 - 1.5 * IQR)) |(boston_df_o1 > (Q3 
+ 1.5 * IQR))
```



4	False	False	False	False	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False	False	False	False	True	False
19	False	False	False	False	False	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False	False	False	False	False	False

Detecting outlier with IQR

Now that we know how to detect the outliers, it is important to understand if they needs to be removed or corrected. In the next section we will consider a few methods of removing the outliers and if required imputing new values.

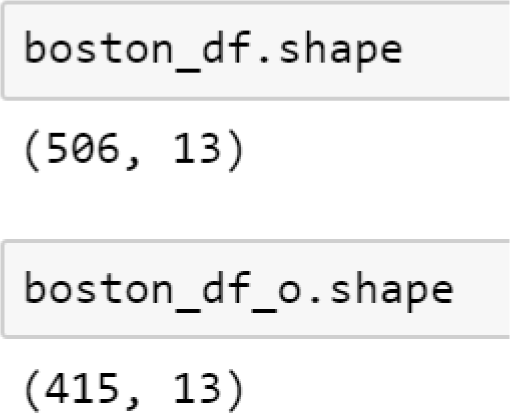
Working with Outliers: Correcting, Removing

During data analysis when you detect the outlier one of most difficult decision could be how one should deal with the outlier. Should they remove them or correct them? Before we talk about this, we will have a look at few methods of removing the outliers.

Z-Score

In the previous section, we saw how one can detect the outlier using Z-score but now we want to remove or filter the outliers and get the clean data. This can be done with just one line code as we have already calculated the Z-score.

```
boston_df_o = boston_df_o[(z < 3).all(axis=1)]
```



```
boston_df.shape

(506, 13)

boston_df_o.shape

(415, 13)
```

With and without outlier size of the dataset

So, above code removed around 90+ rows from the dataset i.e. outliers have been removed.

IQR Score -

Just like Z-score we can use previously calculated IQR score to filter out the outliers by keeping only valid values.

```
boston_df_out = boston_df_o1[~((boston_df_o1 < (Q1 - 1.5 * 
IQR)) |(boston_df_o1 > (Q3 + 1.5 * IQR))).any(axis=1)]


boston_df_out.shape
```

The above code will remove the outliers from the dataset.

There are multiple ways to detect and remove the outliers but the methods, we have used for this exercise, are widely used and easy to understand.

Whether an outlier should be removed or not. Every data analyst/data scientist might get these thoughts once in every problem they are working on. I have found some good explanations -

> *https://www.researchgate.net/post/When_is_it_justifiable_to_exclude_outlier_data_points_from_statistical_analyses*

> *https://www.researchgate.net/post/Which_is_the_best_method_for_removing_outliers_in_a_data_set*

> *https://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/*

To summarize their explanation- bad data, wrong calculation, these can be identified as Outliers and should be dropped but at the same time you might want to correct them too, as they change the level of data i.e. mean which cause issues when you model your data. For ex- 5 people get salary of 10K, 20K, 30K, 40K and 50K and suddenly one of the person start getting salary of 100K. Consider this situation as, you are the employer, the new salary update might be seen as biased and you might need to increase other employee's salary too, to keep the balance. So, there can be multiple reasons you want to understand and correct the outliers.

Summary

Throughout this exercise we saw how in data analysis phase one can encounter with some unusual data i.e outlier. We learned about techniques which can be used to detect and remove those outliers. But there was a question raised about assuring if it is okay to remove the outliers. To answer those questions we have found further readings(this links are mentioned in the previous section). Hope this post helped the readers in knowing Outliers.

Note- For this exercise, below tools and libaries were used.

Framework- Jupyter Notebook, **Language-** Python, **Libraries**- sklearn library, Numpy, Panda and Scipy, **Plot Lib-** Seaborn and Matplot.

Refernces

1. Boston Dataset
2. Github Repo
3. KDNuggets outliers
4. Detect outliers