**What are Outliers?**

* In statistics, any observations or data points that deviate significantly and do not conform with the rest of the observation or data points in a dataset are called outliers. Outliers are extreme values in a feature or dataset. For example, if you have a dataset with a feature height. The majority of the values in this feature range between 4.5−6.54.5−6.5 feet, but there is one value with 10 feet. This value would be considered an outlier, as it is not only an extreme value but an impossible height as well.
* Outliers are also called **aberrations**, **abnormal points**, **anomalies**, etc. It is essential to detect and handle outliers in a dataset as it can have a significant impact on many statistical methods, such as mean, variance, etc., and the performance of the ML models. It can lead to misleading, inconsistent, and inaccurate results if they are not properly accounted for.

**Types of Outliers**

Based on their characteristics, outliers or anomalies can be divided into three categories, as mentioned below:

**Global Outliers**

It is also called point anomaly. Any observations or data points are considered as global outliers if they deviate significantly from the rest of the observations or data points in a dataset. For example, if you are collecting observations of temperatures in a city, then a value of 100 degrees would be considered an outlier, as it is an extreme as well as impossible temperature value for a city.

**Contextual Outliers**

Any data points or observations are considered as contextual outliers if their value significantly deviates from the rest of the data points in a particular context. It means that the same values may not be considered an outlier in a different context. For example, if you have observations of temperatures in a city, then a value of 40 degrees would be considered an outlier in winter, but the same value might be part of the normal observations in summer.

**Collective Outliers**

Any group of observations or data points within a data set is considered collective outliers if these observations as a collection deviate significantly from the entire data set. It means that these values, individually without collection with other data points, are not considered as either contextual or global outliers.

For example, in an Intrusion Detection System, a **DOS (denial-of-service)** package from one computer to another may be considered normal behavior. However, if several computers are receiving DOS packages at the same time, then this may be considered anomalous behavior, and as a collection of data points, they can be considered collective outliers.

**How to Identify Outliers**

Outliers can be identified through various methods, each with its own advantages and drawbacks. Here are some commonly used techniques:

**1. Statistical Methods**

**Standard Deviation:** If the data follows a Gaussian or Gaussian-like distribution, you can use standard deviation to identify outliers. In a normal distribution, most data falls within three standard deviations of the mean. Any data point that lies more than three standard deviations from the mean can be considered an outlier.

**Percentile Method**

* The percentile method identifies outliers in a dataset by comparing each observation to the rest of the data using percentiles.
* In this method, We first define the upper and lower bounds of a dataset using the desired percentiles. For example, we may use the 5th and 95th percentile for a dataset's lower and upper bounds, respectively. Any observations or data points that reside beyond and outside of these bounds can be considered outliers.
* This method is simple and useful for identifying outliers in symmetrical and normal distributions.

**Interquartile Range (IQR) Method**

* The **interquartile range (IQR)** is a measure of the dispersion of a dataset. It is calculated by taking the difference between the upper and lower quartiles of the dataset. Quartiles are values that divide a dataset into four equal parts or quarters. The **upper quartile (Q3)** is the value greater than or equal to 75% of the other values in the dataset, and the lower quartile (Q1) is the value greater than or equal to 25% of the other values. The IQR is calculated by subtracting Q1 from Q3.
* Using IQR, we can define a dataset's upper and lower bounds. The upper bound is defined as **Q3 + 1.5\*IQR**, and the lower bound is defined as **Q1 - 1.5\*IQR**. Any observations or data points that reside beyond and outside these bounds can be considered outliers.

**Z-Score Method**

* For a given value, the respective z-score represents its distance in terms of the standard deviation. For example, a z-score of 2 represents that the data point is 2 standard deviations away from the mean. To detect the outliers using the **z-score**, we can define the lower and upper bounds of the dataset. The **upper bound** is defined as z = 3, and the **lower bound** is defined as z = -3. This means any value more than 3 standard deviations away from the mean will be considered an outlier.

**2. Visualization Tools**

**Box and Whisker Plots (Box Plots):** A Box Plot is a graphical representation of statistical data based on a five-number summary (minimum, first quartile, median, third quartile, and maximum). In a box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here whiskers are drawn from both ends of the box to indicate variability outside the upper and lower quartiles, hence the plot is also termed as the box-and-whisker plot and box-and-whisker diagram. Outliers in box plots are indicated as individual points that are located above or below the whiskers.

**Scatter Plots:** Scatter plots are used for visualizing two-dimensional graphics that uses dots to represent the values obtained for two different variables — one plotted along the x-axis and the other plotted along the y-axis. Scatter plots are used when you want to show the relationship between two variables. Scatter plots are sometimes called correlation plots because they show how two variables are correlated.

**How to Handle Outliers**

After identifying the outliers, the next step is to handle them appropriately. The method used to handle outliers depends on the nature of the data and the reason for the outliers’ existence.

**Deletion:** Outliers can be deleted from the dataset. This is often appropriate for outliers caused by errors in data collection or recording.

**Imputation:** For outliers that are expected as part of the distribution of the data, imputation can be used. This involves replacing the outlier with a new value, such as the mean or median of the data.

**Capping:** Outliers can also be capped at a particular maximum and/or minimum value. This method retains the “outlierness” of the data point but reduces the impact on statistical analysis.

**Transforming:** This involves applying a mathematical operation such as log, square root, or inverse to the data. This can reduce the impact of the outliers.

* **Remove outliers:**  
  In some cases, it may be appropriate to simply remove the observations that contain outliers. This can be particularly useful if you have a large number of observations and the outliers are not true representatives of the underlying population.
* **Transform outliers:**  
  The impact of outliers can be reduced or eliminated by transforming the feature. For example, a log transformation of a feature can reduce the skewness in the data, reducing the impact of outliers.
* **Impute outliers:**  
  In this case, outliers are simply considered as missing values. You can employ various imputation techniques for missing values, such as mean, median, mode, nearest neighbor, etc., to impute the values for outliers.
* **Use robust statistical methods:**  
  Some of the statistical methods are less sensitive to outliers and can provide more reliable results when outliers are present in the data. For example, we can use median and IQR for the statistical analysis as they are not affected by the outlier’s presence. This way we can minimize the impact of outliers in statistical analysis.

Let’s see below how you can implement these methods to detect outliers using Python.

import pandas as pd

import numpy as np

data = {'value': [1, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 40, 56, 73, 176]}

df = pd.DataFrame(data)

*# Outlier Detection using percentile*

lower = df['value'].quantile(0.05)

upper = df['value'].quantile(0.95)

outliers\_percentile = df[((df['value'] < lower) | (df['value'] > upper))]

*# Outlier Detection using IQR*

Q1 = df['value'].quantile(0.25)

Q3 = df['value'].quantile(0.75)

IQR = Q3-Q1

outliers\_iqr = df[((df['value'] < (Q1-1.5\*IQR)) | (df['value'] > (Q3+1.5\*IQR)))]

*# Outlier detection using Z-Score*

mean = np.mean(df['value'])

std = np.std(df['value'])

df['ZScore'] = (df['value'] - mean)/std

outliers\_zscore = df[abs(df['ZScore']) >3]

**How Should the Outliers Be Handled?**

import pandas as pd

import numpy as np

data = {'value': [1, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 40, 56, 73, 176]}

df = pd.DataFrame(data)

*# Outlier Detection using IQR*

Q1 = df['value'].quantile(0.25)

Q3 = df['value'].quantile(0.75)

IQR = Q3-Q1

*# Removal of Outliers*

df\_remove\_outliers = df[((df['value'] >= (Q1-1.5\*IQR)) | (df['value'] <= (Q3+1.5\*IQR)))]

*# Transform outliers using log transformation*

df['value'] = np.log10(df['value'])

*# Impute outliers - using a median technique*

median\_to\_impute = df[((df['value'] >= (Q1-1.5\*IQR)) | (df['value'] <= (Q3+1.5\*IQR)))]['value'].median()

df.loc[((df['value'] < (Q1-1.5\*IQR)) | (df['value'] >(Q3+1.5\*IQR))), 'value'] = median\_to\_impute

* **Outliers** are observations in a dataset that differ significantly from the rest of the data points. They can occur due to errors or anomalies in the data collection process, or they may be legitimate observations that are simply rare or extreme.
* Outliers can sometimes lead to misleading, inconsistent, and inaccurate results if they are not properly accounted for. Therefore, it is essential to identify and deal with outliers to obtain accurate and meaningful results from the data analysis.
* There are several ways to detect outliers, such as the percentile method, IQR method, and z-score method. Outliers, once detected, can be handled in several ways, such as removal, transformation, imputation, etc.

**Conclusion**

In conclusion, outliers, while challenging, can be effectively managed by following the right identification and handling techniques. Through statistical methods, data visualization tools, and appropriate handling methods, you can ensure that your data is robust and reliable, leading to more accurate statistical analyses and better-performing machine learning models. Always remember, understanding your data is the key to deciding how to handle outliers, and not all outliers are bad or need to be removed. It’s all about understanding the nature and source of these anomalies. This comprehensive guide provides a strong foundation for identifying and handling outliers in your data.

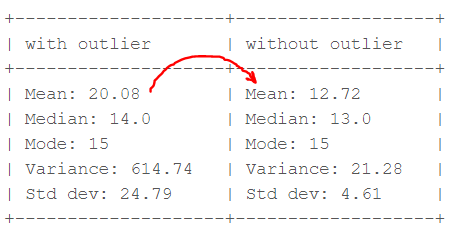
In statistics, we have three measures of central tendency namely Mean, Median, and Mode. They help us describe the data.

* Mean is the accurate measure to describe the data when we do not have any outliers present.
* Median is used if there is an outlier in the dataset.
* Mode is used if there is an outlier AND about ½ or more of the data is the same.

Mean’ is the only measure of central tendency that is affected by the outlier treatment which in turn impacts Standard deviation.

Example

Consider a small dataset, sample= [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]. By looking at it, one can quickly say ‘101’ is an outlier that is much larger than the other values.

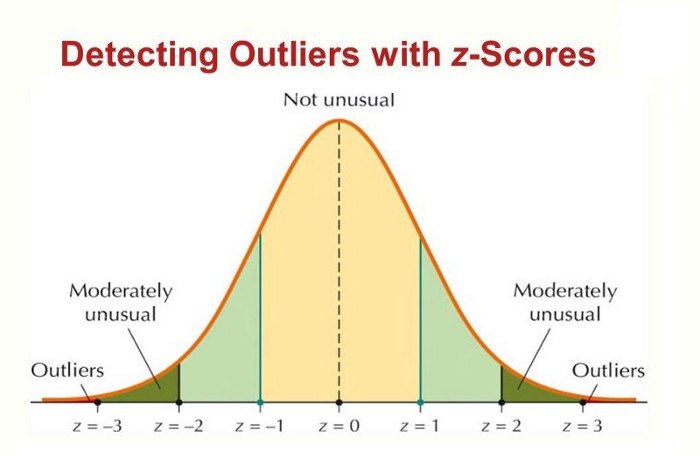
Computation with and without outlier (Image by author)

From the above calculations, we can clearly say the Mean is more affected than the Median.

Python code for boxplot is:

**Detecting Outliers using the Z-scores**

**Criteria:**any data point whose Z-score falls out of 3rd standard deviation is an outlier treatment.

Detecting Outliers with Z-scores

**Steps**

* loop through all the data points and compute the Z-score using the formula (Xi-mean)/std.
* define a threshold value of 3 and mark the datapoints whose absolute value of Z-score is greater than the threshold as outliers.

**import** numpy **as** np

outliers = []

**def** **detect\_outliers\_zscore**(data):

thres = 3

mean = np.mean(data)

std = np.std(data)

# print(mean, std)

**for** i **in** data:

z\_score = (i-mean)/std

**if** (np.abs(z\_score) > thres):

outliers.append(i)

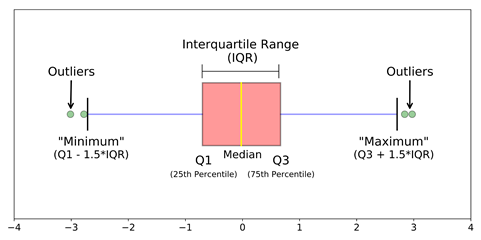
**return** outliers# Driver code

sample\_outliers = detect\_outliers\_zscore(sample)

print("Outliers from Z-scores method: ", sample\_outliers)

The above code outputs: **Outliers from Z-scores method: [101]**

**Detecting Outliers using the Inter Quantile Range(IQR)**

IQR to detect Outliners

**Criteria:** data points that lie 1.5 times of IQR above Q3 and below Q1 are outliers. This shows in detail about outlier treatment in Python.

**Steps**

* Sort the dataset in ascending order
* calculate the 1st and 3rd quartiles(Q1, Q3)
* compute IQR=Q3-Q1
* compute lower bound = (Q1–1.5\*IQR), upper bound = (Q3+1.5\*IQR)
* loop through the values of the dataset and check for those who fall below the lower bound and above the upper bound and mark them as outlier treatment in python

Python Code

outliers = []

def detect\_outliers\_iqr(data):

data = sorted(data)

q1 = np.percentile(data, 25)

q3 = np.percentile(data, 75)

# print(q1, q3)

IQR = q3-q1

lwr\_bound = q1-(1.5\*IQR)

upr\_bound = q3+(1.5\*IQR)

# print(lwr\_bound, upr\_bound)

**for** i in data:

**if** (i<lwr\_bound **or** i>upr\_bound):

outliers.append(i)

**return** outliers# Driver code

sample\_outliers = detect\_outliers\_iqr(sample)

**print**("Outliers from IQR method: ", sample\_outliers)

The above code outputs: **Outliers from IQR method: [101]**

How to Handle Outliers?

Till now we learned about detecting the outliers handling. The main question is how to deal with outliers?  
  
Below are some of the methods of treating the outliers:

Step 1: Trimming/Remove the outliers

In this technique, we remove the outliers from the dataset. Although it is not a good practice to follow.  
  
Python code to delete the outlier treatment and copy the rest of the elements to another array.

# Trimming for **i** in sample\_outliers: a = np.delete(sample, np.where(sample==i)) print(a) # print(len(sample), len(a))

The outlier ‘101’ is deleted and the rest of the data points are copied to another array ‘a’.

Step 2: Quantile Based Flooring and Capping

In this technique, the outlier is capped at a certain value above the 90th percentile value or floored at a factor below the 10th percentile value. Python code to delete the outlier and copy the rest of the elements to another array.

# Computing 10th, 90th percentiles and replacing the outlier treatment in python

tenth\_percentile = np.percentile(sample, 10)

ninetieth\_percentile = np.percentile(sample, 90)

# print(tenth\_percentile, ninetieth\_percentile)b =

np.where(sample<tenth\_percentile, tenth\_percentile, sample)

b = np.where(b>ninetieth\_percentile, ninetieth\_percentile, b)

# print("Sample:", sample)

print("New array:",b)

The above code outputs: **New array:**[15, 20.7, 18, 7.2, 13, 16, 11, 20.7, 7.2, 15, 10, 9]

The data points that are lesser than the 10th percentile are replaced with the 10th percentile value and the data points that are greater than the 90th percentile are replaced with 90th percentile value.

Step 3: Mean/Median Imputation

As the mean value is highly influenced by the outlier treatment, it is advised to replace the outliers with the median value.

**Python Code:**

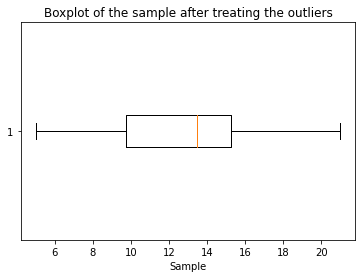
median = np.median(sample)# Replace with median **for** i in sample\_outliers: c = np.where(sample==i, 14, sample) print("Sample: ", sample) print("New array: ",c) # print(x.dtype)

Step 5: Visualizing the Data after Treating the Outlier

plt.boxplot(c, vert=False)

plt.title("Boxplot of the sample after treating the outliers")

plt.xlabel("Sample")

Data after treating Outliner

Conclusion

In conclusion, identifying and addressing [outliers handling](https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/) is paramount in data analysis. These data anomalies can skew results, leading to inaccurate insights and decisions. By employing robust detection techniques and thoughtful treatment strategies, we can enhance the integrity of our analyses and unlock hidden patterns within our data. How to Handle Outlier treatment in python, in this article once understood and managed, become valuable sources of information, ultimately contributing to more informed and reliable decision-making processes.

Hope you understand how to deal with outliers effectively. Knowing how to treat outliers is crucial for accurate data analysis. Proper handling of outliers can significantly improve your results and insights.

Key Takeaways:

* Learning techniques to detect outliers: boxplots, Z-score method, interquartile range (IQR) method
* Strategies to handle outliers: trimming/removing, quantile-based flooring and capping, mean/median imputation
* Visualizing and evaluating the data after treating outliers for improved analysis and decision-making