Z-Score technique to handle the outliers

Example1

Here's a Python code using the numpy and scipy libraries to demonstrate the Z-score method for handling outliers:

In [33]:
```
import numpy as np
from scipy import stats

def detect_outliers_zscore(data, threshold=3):
    """
    Detects outliers using Z-score method.

    :param data: List or array-like data points.
    :param threshold: Threshold value for Z-score. Default is 3.
    :return: List of outliers.
    """
    z_scores = np.abs(stats.zscore(data))
    outliers = np.where(z_scores > threshold)
    return np.array(data)[outliers]

# data points
data = [10, 12, 12, 13, 12, 11, 11, 52, 13, 12, 11]

# Detecting outliers
outliers = detect_outliers_zscore(data)

print("Outliers:", outliers)
```

```
Outliers: [52]
```

In the above code:

We first calculate the Z-scores for the data. We then identify outliers as those data points where the absolute Z-score is greater than a threshold (default is 3, but you can modify this based on your needs). We return the outliers.

Example 2

In [37]:
```
import numpy as np
import pandas as pd
import seaborn as sns

# Load the iris dataset from seaborn
df = sns.load_dataset('iris')

# Calculate the Z-scores for the 'sepal_length' column
df['Z_score_sepal_length'] = (df['sepal_length'] - df['sepal_length'].mean()) / df['sepa

# Filter rows in dataframe to exclude data points that are outliers (where |Z-score| > 3
df_no_outliers = df[np.abs(df['Z_score_sepal_length']) <= 3]

print("Original Dataset:")
print(df)

print("\nDataset without Outliers based on sepal_length:")
print(df_no_outliers)
```

```python
# You can drop the 'Z_score_sepal_length' column if you don't need it
df_no_outliers = df_no_outliers.drop(columns=['Z_score_sepal_length'])
print("\nCleaned Dataset:")
print(df_no_outliers)
```

```
Original Dataset:
     sepal_length  sepal_width  petal_length  petal_width    species  \
0             5.1          3.5           1.4          0.2     setosa   
1             4.9          3.0           1.4          0.2     setosa   
2             4.7          3.2           1.3          0.2     setosa   
3             4.6          3.1           1.5          0.2     setosa   
4             5.0          3.6           1.4          0.2     setosa   
..            ...          ...           ...          ...        ...   
145           6.7          3.0           5.2          2.3  virginica   
146           6.3          2.5           5.0          1.9  virginica   
147           6.5          3.0           5.2          2.0  virginica   
148           6.2          3.4           5.4          2.3  virginica   
149           5.9          3.0           5.1          1.8  virginica   

     Z_score_sepal_length  
0               -0.900681  
1               -1.143017  
2               -1.385353  
3               -1.506521  
4               -1.021849  
..                    ...  
145              1.038005  
146              0.553333  
147              0.795669  
148              0.432165  
149              0.068662  

[150 rows x 6 columns]

Dataset without Outliers based on sepal_length:
     sepal_length  sepal_width  petal_length  petal_width    species  \
0             5.1          3.5           1.4          0.2     setosa   
1             4.9          3.0           1.4          0.2     setosa   
2             4.7          3.2           1.3          0.2     setosa   
3             4.6          3.1           1.5          0.2     setosa   
4             5.0          3.6           1.4          0.2     setosa   
..            ...          ...           ...          ...        ...   
145           6.7          3.0           5.2          2.3  virginica   
146           6.3          2.5           5.0          1.9  virginica   
147           6.5          3.0           5.2          2.0  virginica   
148           6.2          3.4           5.4          2.3  virginica   
149           5.9          3.0           5.1          1.8  virginica   

     Z_score_sepal_length  
0               -0.900681  
1               -1.143017  
2               -1.385353  
3               -1.506521  
4               -1.021849  
..                    ...  
145              1.038005  
146              0.553333  
147              0.795669  
148              0.432165  
149              0.068662  

[150 rows x 6 columns]

Cleaned Dataset:
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

In this example, the iris dataset probably doesn't have extreme outliers given its nature, so the filtered dataset might look very similar to the original one. However, this procedure is a generalized approach that can be applied to any dataset.

In [40]:
```
# import sklearn module
import  sklearn
from sklearn import datasets

# Load california dataset
california_dateset = sklearn.datasets.fetch_california_housing(as_frame=True)
```

In [42]:
```
# Loading california dataset for getting outliers using Zscores method
california_dateset = sklearn.datasets.fetch_california_housing(as_frame=True)
df = pd.DataFrame(california_dateset.data)
```

In [44]:
```
numerical_feature = 'MedInc'
```

In [46]:
```
# Calculate the Z-scores for the 'MedInc' column
zscores =  (df[numerical_feature]- df[numerical_feature].mean())/df[numerical_feature].s

# Define the threshold for outlier detection (typically |Z| > 3)
threshold=3

# Filter rows in dataframe to exclude data points that are outliers (where |Z-score| > 3
outliers =  df[np.abs(zscores)> threshold]

# Printing the outliers
print("Outliers detecting using zscore method")
print(outliers)
```

```
Outliers detecting using zscore method
        MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
131    11.6017      18.0  8.335052   1.082474       533.0  2.747423     37.84
409    10.0825      52.0  8.209016   1.024590       658.0  2.696721     37.90
510    11.8603      39.0  7.911111   0.984127       808.0  2.565079     37.82
511    13.4990      42.0  8.928358   1.000000      1018.0  3.038806     37.82
512    12.2138      52.0  9.210227   1.039773      1001.0  2.843750     37.82
...        ...       ...       ...        ...         ...       ...       ...
20376  10.2614      16.0  6.421277   0.919149       578.0  2.459574     34.16
20380  10.1597      16.0  7.606936   1.121387       450.0  2.601156     34.14
20389  10.0595      26.0  8.692308   1.076923       573.0  3.148352     34.19
20426  10.0472      11.0  9.890756   1.159664       415.0  3.487395     34.18
20436  12.5420      10.0  9.873315   1.102426      1179.0  3.177898     34.21

       Longitude
131      -122.19
409      -122.28
```

```
510      -122.22
511      -122.22
512      -122.23
...          ...
20376    -118.86
20380    -118.83
20389    -118.90
20426    -118.69
20436    -118.69

[345 rows x 8 columns]
```

In [ ]:

```
!jupyter nbconvert --to webpdf --allow-chromium-download Week8_Lab1.ipynb
```