import pandas as pd

# Load the Titanic dataset

url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

df = pd.read\_csv(url)

# Display the first few rows of the dataset

print("Original Titanic DataFrame:")

print(df[['Age', 'Fare']].head())

**Step 2: Z-Score Method**

The Z-score method helps identify how many standard deviations away a data point is from the mean.

python

Copy code

import numpy as np

# Calculate Z-scores for 'Age' and 'Fare'

df['Age\_Z'] = (df['Age'] - df['Age'].mean()) / df['Age'].std()

df['Fare\_Z'] = (df['Fare'] - df['Fare'].mean()) / df['Fare'].std()

# Identify outliers based on Z-score (threshold of 3)

outliers\_z\_age = df[df['Age\_Z'].abs() > 3]

outliers\_z\_fare = df[df['Fare\_Z'].abs() > 3]

print("\nOutliers based on Z-score (Age):")

print(outliers\_z\_age[['Age', 'Age\_Z']])

print("\nOutliers based on Z-score (Fare):")

print(outliers\_z\_fare[['Fare', 'Fare\_Z']])

**Step 3: Interquartile Range (IQR) Method**

The IQR method defines outliers based on the quartiles.

python

Copy code

# Calculate Q1 and Q3 for 'Age' and 'Fare'

Q1\_age = df['Age'].quantile(0.25)

Q3\_age = df['Age'].quantile(0.75)

IQR\_age = Q3\_age - Q1\_age

Q1\_fare = df['Fare'].quantile(0.25)

Q3\_fare = df['Fare'].quantile(0.75)

IQR\_fare = Q3\_fare - Q1\_fare

# Identify outliers for 'Age'

lower\_bound\_age = Q1\_age - 1.5 \* IQR\_age

upper\_bound\_age = Q3\_age + 1.5 \* IQR\_age

outliers\_iqr\_age = df[(df['Age'] < lower\_bound\_age) | (df['Age'] > upper\_bound\_age)]

# Identify outliers for 'Fare'

lower\_bound\_fare = Q1\_fare - 1.5 \* IQR\_fare

upper\_bound\_fare = Q3\_fare + 1.5 \* IQR\_fare

outliers\_iqr\_fare = df[(df['Fare'] < lower\_bound\_fare) | (df['Fare'] > upper\_bound\_fare)]

print("\nOutliers based on IQR (Age):")

print(outliers\_iqr\_age[['Age']])

print("\nOutliers based on IQR (Fare):")

print(outliers\_iqr\_fare[['Fare']])

**Step 4: Visualization Techniques**

**a. Box Plot**

Box plots provide a visual representation of the distribution of data and can highlight outliers.

python

Copy code

import matplotlib.pyplot as plt

import seaborn as sns

# Box plot for 'Age'

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

sns.boxplot(x=df['Age'])

plt.title('Box Plot for Age')

# Box plot for 'Fare'

plt.subplot(1, 2, 2)

sns.boxplot(x=df['Fare'])

plt.title('Box Plot for Fare')

plt.show()

**b. Scatter Plot**

If you want to visualize relationships between features, scatter plots can be useful.

python

Copy code

# Scatter plot for 'Age' vs 'Fare'

plt.figure(figsize=(8, 5))

plt.scatter(df['Age'], df['Fare'], alpha=0.5)

plt.title('Scatter Plot of Age vs Fare')

plt.xlabel('Age')

plt.ylabel('Fare')

plt.grid()

plt.show()

**Summary**

In this example, we demonstrated:

1. **Z-Score Method**: Identifying outliers based on how many standard deviations a point is from the mean for Age and Fare.
2. **IQR Method**: Using the interquartile range to identify outliers for Age and Fare.
3. **Visualization Techniques**: Using box plots and scatter plots to visually inspect outliers.

<https://github.com/codebasics/math-for-machine-learning/blob/main/3_normal_distribution/normal_distribution.ipynb>

## Normal Distribution and Z Score: Math and statistics for data science

In [79]:

**import** pandas **as** pd

**import** seaborn **as** sn

We are going to use heights dataset from kaggle.com. Dataset has heights and weights both but I have removed weights to make it simple

<https://www.kaggle.com/mustafaali96/weight-height>

In [80]:

df **=** pd**.**read\_csv("heights.csv")

df**.**head()

Out[80]:

|  | **gender** | **height** |
| --- | --- | --- |
| **0** | Male | 73.847017 |
| **1** | Male | 68.781904 |
| **2** | Male | 74.110105 |
| **3** | Male | 71.730978 |
| **4** | Male | 69.881796 |

**(1) Outlier detection and removal using Standard Deviation**

In [81]:

df**.**height**.**describe()

Out[81]:

count 10000.000000

mean 66.367560

std 3.847528

min 54.263133

25% 63.505620

50% 66.318070

75% 69.174262

max 78.998742

Name: height, dtype: float64

In [82]:

sn**.**histplot(df**.**height, kde**=True**)

Out[82]:

<AxesSubplot:xlabel='height', ylabel='Count'>

In [83]:

mean **=** df**.**height**.**mean()

mean

Out[83]:

66.367559754866

In [84]:

std\_deviation **=** df**.**height**.**std()

std\_deviation

Out[84]:

3.847528120795573

In [85]:

mean**-**3**\***std\_deviation

Out[85]:

54.824975392479274

In [86]:

mean**+**3**\***std\_deviation

Out[86]:

77.91014411725271

In [88]:

df[(df**.**height **<** 54.82) **|** (df**.**height **>** 77.91)]

Out[88]:

|  | **gender** | **height** |
| --- | --- | --- |
| **994** | Male | 78.095867 |
| **1317** | Male | 78.462053 |
| **2014** | Male | 78.998742 |
| **3285** | Male | 78.528210 |
| **3757** | Male | 78.621374 |
| **6624** | Female | 54.616858 |
| **9285** | Female | 54.263133 |

In [90]:

df\_no\_outlier **=** df[(df**.**height**<**77.91) **&** (df**.**height**>**54.82)]

df\_no\_outlier**.**shape

Out[90]:

(9993, 2)

**(2) Outlier detection and removal using Z Score**

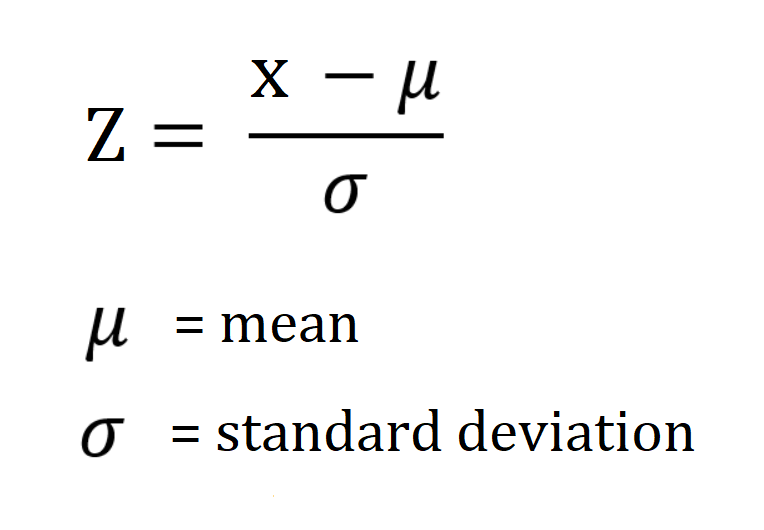
Z score is a way to achieve same thing that we did above in part (1)

Z score indicates how many standard deviation away a data point is.

For example in our case mean is 66.37 and standard deviation is 3.84.

If a value of a data point is 77.91 then Z score for that is 3 because it is 3 standard deviation away (77.91 = 66.37 + 3 \* 3.84)

Calculate the Z Score



Let's add a new column in our dataframe for this Z score

In [91]:

df['zscore'] **=** ( df**.**height **-** df**.**height**.**mean() ) **/** df**.**height**.**std()

df**.**head(5)

Out[91]:

|  | **gender** | **height** | **zscore** |
| --- | --- | --- | --- |
| **0** | Male | 73.847017 | 1.943964 |
| **1** | Male | 68.781904 | 0.627505 |
| **2** | Male | 74.110105 | 2.012343 |
| **3** | Male | 71.730978 | 1.393991 |
| **4** | Male | 69.881796 | 0.913375 |

Above for first record with height 73.84, z score is 1.94. This means 73.84 is 1.94 standard deviation away from mean

In [93]:

df**.**height**.**mean()

Out[93]:

66.367559754866

In [94]:

df**.**height**.**std()

Out[94]:

3.847528120795573

In [92]:

(73.84**-**66.37)**/**3.84

Out[92]:

1.9453124999999998

In [95]:

df[df['zscore']**>**3]

Out[95]:

|  | **gender** | **height** | **zscore** |
| --- | --- | --- | --- |
| **994** | Male | 78.095867 | 3.048271 |
| **1317** | Male | 78.462053 | 3.143445 |
| **2014** | Male | 78.998742 | 3.282934 |
| **3285** | Male | 78.528210 | 3.160640 |
| **3757** | Male | 78.621374 | 3.184854 |

In [96]:

df[df['zscore']**<-**3]

Out[96]:

|  | **gender** | **height** | **zscore** |
| --- | --- | --- | --- |
| **6624** | Female | 54.616858 | -3.054091 |
| **9285** | Female | 54.263133 | -3.146027 |

2. IQR:

## Outlier Detection and Removal Using IQR

In [3]:

**import** pandas **as** pd

df **=** pd**.**read\_csv("heights.csv")

df

Out[3]:

|  | **name** | **height** |
| --- | --- | --- |
| **0** | mohan | 1.2 |
| **1** | maria | 2.3 |
| **2** | sakib | 4.9 |
| **3** | tao | 5.1 |
| **4** | virat | 5.2 |
| **5** | khusbu | 5.4 |
| **6** | dmitry | 5.5 |
| **7** | selena | 5.5 |
| **8** | john | 5.6 |
| **9** | imran | 5.6 |
| **10** | jose | 5.8 |
| **11** | deepika | 5.9 |
| **12** | yoseph | 6.0 |
| **13** | binod | 6.1 |
| **14** | gulshan | 6.2 |
| **15** | johnson | 6.5 |
| **16** | donald | 7.1 |
| **17** | aamir | 14.5 |
| **18** | ken | 23.2 |
| **19** | Liu | 40.2 |

In [4]:

df**.**describe()

Out[4]:

|  | **height** |
| --- | --- |
| **count** | 20.000000 |
| **mean** | 8.390000 |
| **std** | 8.782812 |
| **min** | 1.200000 |
| **25%** | 5.350000 |
| **50%** | 5.700000 |
| **75%** | 6.275000 |
| **max** | 40.200000 |

### Detect outliers using IQR

In [5]:

Q1 **=** df**.**height**.**quantile(0.25)

Q3 **=** df**.**height**.**quantile(0.75)

Q1, Q3

Out[5]:

(5.3500000000000005, 6.275)

In [6]:

IQR **=** Q3 **-** Q1

IQR

Out[6]:

0.9249999999999998

In [7]:

lower\_limit **=** Q1 **-** 1.5**\***IQR

upper\_limit **=** Q3 **+** 1.5**\***IQR

lower\_limit, upper\_limit

Out[7]:

(3.962500000000001, 7.6625)

**Here are the outliers**

In [8]:

df[(df**.**height**<**lower\_limit)**|**(df**.**height**>**upper\_limit)]

Out[8]:

|  | **name** | **height** |
| --- | --- | --- |
| **0** | mohan | 1.2 |
| **1** | maria | 2.3 |
| **17** | aamir | 14.5 |
| **18** | ken | 23.2 |
| **19** | Liu | 40.2 |

### Remove outliers

In [9]:

df\_no\_outlier **=** df[(df**.**height**>**lower\_limit)**&**(df**.**height**<**upper\_limit)]

df\_no\_outlier

Out[9]:

|  | **name** | **height** |
| --- | --- | --- |
| **2** | sakib | 4.9 |
| **3** | tao | 5.1 |
| **4** | virat | 5.2 |
| **5** | khusbu | 5.4 |
| **6** | dmitry | 5.5 |
| **7** | selena | 5.5 |
| **8** | john | 5.6 |
| **9** | imran | 5.6 |
| **10** | jose | 5.8 |
| **11** | deepika | 5.9 |
| **12** | yoseph | 6.0 |
| **13** | binod | 6.1 |
| **14** | gulshan | 6.2 |
| **15** | johnson | 6.5 |
| **16** | donald | 7.1 |

https://github.com/codebasics/py/blob/master/ML/FeatureEngineering/3\_outlier\_IQR/3\_outliers\_iqr.ipynb