Bahria University,

Karachi Campus

## LAB EXPERIMENT NO.

10

## LIST OF TASKS

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| **1** | An automotive company wants to identify and treat outliers in their "mtcars" dataset. which contains information about various car models. They suspect that certain car models might have outlier values in certain variables and want to analyze and handle them using Python. |
| **2** | An automotive company wants to identify and treat outliers in their "mtcars" dataset. which contains information about various car models. They suspect that certain car models might have outlier values in certain variables and want to analyze and handle them using Python. |

Submitted On:

08 may 2024

(Date: DD/MM/YY)

**TASK#1:** An automotive company wants to identify and treat outliers in their "mtcars" dataset. which contains information about various car models. They suspect that certain car models might have outlier values in certain variables and want to analyze and handle them using Python.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from pydataset import data

mtcars = data('mtcars')

variables\_of\_interest = ['mpg', 'disp', 'hp', 'drat', 'wt', 'qsec']

plt.figure(figsize=(10, 6))

mtcars[variables\_of\_interest].boxplot()

plt.title('Box Plot for Outlier Detection')

plt.xticks(rotation=45)

plt.show()

from scipy.stats import zscore

z\_scores = zscore(mtcars[variables\_of\_interest])

abs\_z\_scores = np.abs(z\_scores)

outliers\_zscore = (abs\_z\_scores > 3).any(axis=1)

print("Outliers detected using Z-score method:")

print(mtcars[outliers\_zscore])

Q1 = mtcars[variables\_of\_interest].quantile(0.25)

Q3 = mtcars[variables\_of\_interest].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

outliers\_iqr = ((mtcars[variables\_of\_interest] < lower\_bound) | (mtcars[variables\_of\_interest] > upper\_bound)).any(axis=1)

print("\nOutliers detected using IQR method:")

print(mtcars[outliers\_iqr])

plt.figure(figsize=(10, 6))

for var in variables\_of\_interest:

    plt.scatter(mtcars.index, mtcars[var], label=var)

threshold\_value = 300

plt.axhline(y=threshold\_value, color='r', linestyle='--', label=f'Threshold: {threshold\_value}')

plt.title('Scatter Plot for Outlier Detection with Threshold')

plt.xlabel('Index')

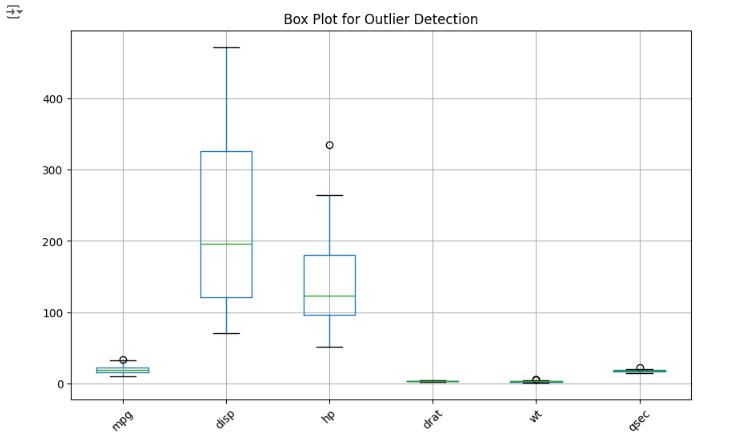
plt.ylabel('Values')

plt.legend()

plt.show()

A screenshot of a computer program

Description automatically generated

A screen shot of a graph

Description automatically generated

**TASK #2:** An automotive company wants to identify and treat outliers in their "mtcars" dataset. which contains information about various car models. They suspect that certain car models might have outlier values in certain variables and want to analyze and handle them using Python.

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

mtcars = sns.load\_dataset('mpg')

def detect\_and\_treat\_outliers\_scatter(df, x\_column, y\_column):

    plt.figure(figsize=(8, 6))

    sns.scatterplot(x=x\_column, y=y\_column, data=df)

    plt.title(f'Scatter plot of {x\_column} vs {y\_column}')

    plt.show()

def detect\_and\_treat\_outliers\_histogram(df, column):

    plt.figure(figsize=(8, 6))

    sns.histplot(df[column], kde=True)

    plt.title(f'Histogram of {column}')

    plt.xlabel(column)

    plt.ylabel('Frequency')

    plt.show()

def detect\_and\_treat\_outliers\_boxplot(df, column):

    plt.figure(figsize=(8, 6))

    sns.boxplot(x=df[column])

    plt.title(f'Boxplot of {column}')

    plt.show()

def detect\_and\_treat\_outliers\_zscore(df, column, threshold=3):

    z\_scores = np.abs((df[column] - df[column].mean()) / df[column].std())

    outliers = df[z\_scores > threshold]

    df\_cleaned = df[z\_scores <= threshold]

    return df\_cleaned, outliers

def detect\_and\_treat\_outliers\_iqr(df, column):

    Q1 = df[column].quantile(0.25)

    Q3 = df[column].quantile(0.75)

    IQR = Q3 - Q1

    lower\_bound = Q1 - 1.5 \* IQR

    upper\_bound = Q3 + 1.5 \* IQR

    outliers = df[(df[column] < lower\_bound) | (df[column] > upper\_bound)]

    df\_cleaned = df[(df[column] >= lower\_bound) & (df[column] <= upper\_bound)]

    return df\_cleaned, outliers

detect\_and\_treat\_outliers\_scatter(mtcars, 'mpg', 'weight')

detect\_and\_treat\_outliers\_histogram(mtcars, 'mpg')

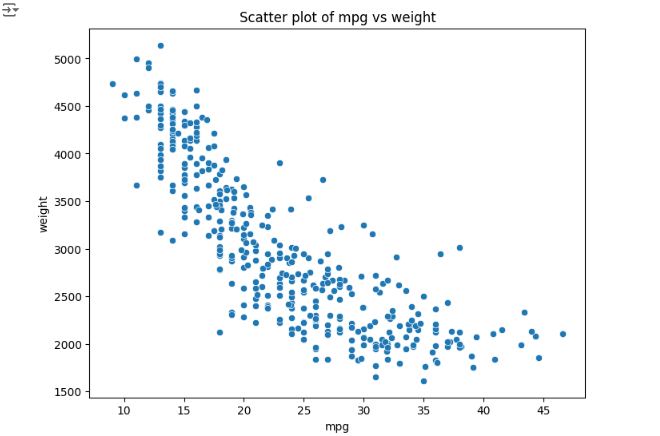
detect\_and\_treat\_outliers\_boxplot(mtcars, 'mpg')

mtcars\_cleaned\_zscore, outliers\_zscore = detect\_and\_treat\_outliers\_zscore(mtcars, 'mpg')

print("Outliers detected using z-score method:\n", outliers\_zscore)

mtcars\_cleaned\_iqr, outliers\_iqr = detect\_and\_treat\_outliers\_iqr(mtcars, 'mpg')

print("Outliers detected using IQR method:\n", outliers\_iqr)

A graph with blue lines

Description automatically generated

A graph with a blue rectangular bar

Description automatically generated with medium confidence

A white background with black text

Description automatically generated