**INDEX**

| **Sr.No. Practical List Date** | **Sign** |
| --- | --- |
| Write a Python program to accept inputs  from users and perform arithmetic  operations24/09/2024  1. |  |
| Write a program to demonstrate relational and  logical operators in Python.01/10/2024  2. |  |
| Write a Python program to demonstrate usage of  08/10/2024  loops. Use both for and while loops to distinguish  between them. [e.g., Reversing the digits of a  3.  number without converting to String]  Demonstrate the use of data structures list, sets,  dictionary15/10/2024  4. |  |
| Import a dataset and perform univariate analysis  22/10/2024  on the numeric columns to analyse the shape of  the data. Write inference of the output. [Python or  5.  R or Excel] |  |
| Demonstrate Hypothesis testing, and ANOVA using  19/11/2024  a dataset [Use Python, R or Excel]  6. |  |
| Demonstrate correlation analysis. Use heatmap  for visualization. Write inferences.26/11/2024  7. |  |
| Import csv or Excel Dataset and demonstrate data  03/12/2024  wrangling, view shape, dimension, column names  8.  of the dataset, ways to select data using column  number, column names, simple and compound  conditional selection , update and modify dataset. |  |
| 10/12/2024  9. Demonstrate group by summary operations and  sorting techniques. |  |

**Practical No.1**

**Aim: Write a Python Program to accept inputs form users and perform arithmetic operations.**

**Code:**

num1 = int(input('Enter First number: '))

num2 = int(input('Enter Second number '))

add = num1 + num2

dif = num1 - num2

mul = num1 \* num2

div = num1 / num2

floor\_div = num1 // num2

power = num1 \*\* num2

modulus = num1 % num2

print('Sum of ',num1 ,'and' ,num2 ,'is :',add)

print('Difference of ',num1 ,'and' ,num2 ,'is :',dif)

print('Product of' ,num1 ,'and' ,num2 ,'is :',mul)

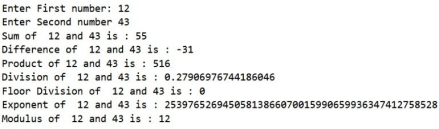
print('Division of ',num1 ,'and' ,num2 ,'is :',div) print('Floor

Division of ',num1 ,'and' ,num2 ,'is :',floor\_div)

print('Exponent of ',num1 ,'and' ,num2 ,'is :',power)

print('Modulus of ',num1 ,'and' ,num2 ,'is :',modulus)

**Output:**

****

**Practical No.2**

**Aim: Write a program to demonstrate relational and logical operators in Python.**

**Code:**

print("Logical Operators")

# logical AND

print(True and True) # True

print(True and False) # False

# logical OR

print(True or False) # True

# logical NOT

print(not True) # False

print("Relational Operators")

a=5

b=6

# equal to operator

print('a == b =', a == b)

# not equal to operator

print('a != b =', a != b)

# greater than operator

print('a > b =', a > b)

# less than operator

print('a < b =', a < b)

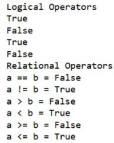
# greater than or equal to operator

print('a >= b =', a >= b)

# less than or equal to operator

print('a <= b =', a <= b)

**Output:**

****

**Practical No.3**

**Aim: Write a Python program to demonstrate usage of loops. Use both for and while loops to distinguish between them. [e.g.**

**Reversing the digits of a number without converting to String.**

**Code:**

**A)Reversing a digits of a number using while loop**

number = int(input("Enter the integer number: "))

revs\_number = 0

while number > 0:

remainder = number % 10

revs\_number = (revs\_number \* 10) + remainder

number = number // 10

print("The reverse number is:", revs\_number)

**Output:**

Enter the integer number: 456

The reverse number is : 654

**B) Reversing of a number using ‘for’ loop.**

# we are taking a number from user as input num = input("Enter the first number:")

# calculate reverse of number

reverse = ''

for i in range(len(num), 0, -1):

reverse += num[i-1]

# print reverse of number

print('The reverse number is =', reverse)

**Output:**

Enter the first number:67567575

The reverse number is = 57576576

**Practical No.4**

**Aim: Demonstrate the use of data structures list, sets, dictionary. Code:**

**a) L**

**ist** #Use of

list subjects

=[]

subjects.append('Python')

subjects.append('ETDS')

print(subjects)

A = [2, 33, 1, 0]

B = [22, 3, 0, -2]

A.append(8)

print(A)

A.sort()

print(A) # [0, 1, 2, 33]

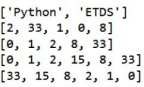
A.insert(3,15)

print(A)

A.sort(reverse=True)

print(A)

**Output:**

****

**b)Sets**

# Creating a set

my\_set = {1, 2, 3, 4, 5}

# Display the set

print("Original Set:", my\_set)

# Adding elements to the set

my\_set.add(6)

my\_set.update([7, 8, 9])

# Display the modified set

print("Modified Set:", my\_set)

# Removing elements from the set

my\_set.remove(3) # Raises an error if the element is not present my\_set.discard(8) # Removes the element if present, otherwise does nothing

# Display the final set

print("Final Set:", my\_set)

# Set operations

set1 = {1, 2, 3, 4, 5}

set2 = {3, 4, 5, 6, 7}

# Union

union\_set = set1.union(set2)

print("Union Set:", union\_set)

# Intersection

intersection\_set = set1.intersection(set2)

print("Intersection Set:", intersection\_set)

# Difference

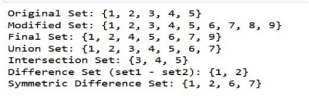
difference\_set = set1.difference(set2)

print("Difference Set (set1 - set2):", difference\_set)

# Symmetric Difference

symmetric\_difference\_set = set1.symmetric\_difference(set2) print("Symmetric Difference Set:", symmetric\_difference\_set)

**Output:**

****

**c) Dictionary**

# Creating a dictionary

student = {

'name': 'Prayag',

'age': 30,

'grades': {'math': 90, 'history': 85, 'english': 95}, 'courses': ['math', 'history', 'english']

}

# Accessing values

print("Student Name:", student['name']) print("Student Age:", student['age'])

# Accessing nested values

print("Math Grade:", student['grades']['math'])

# Modifying values

student['age'] = 21

student['grades']['english'] = 92

# Adding new key-value pairs

student['gender'] = 'Male'

# Displaying the updated dictionary

print("\nUpdated Student Information:") print(student)

# Iterating over keys and values

print("\nIterating over Keys:")

for key in student.keys():

print(key)

print("\nIterating over Values:")

for value in student.values():

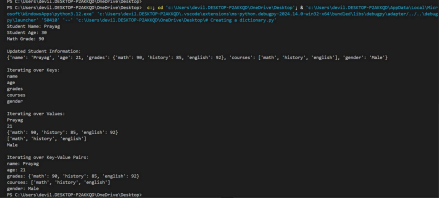
print(value)

print("\nIterating over Key-Value Pairs:")

for key, value in student.items():

print(f"{key}: {value}")

**Output:**

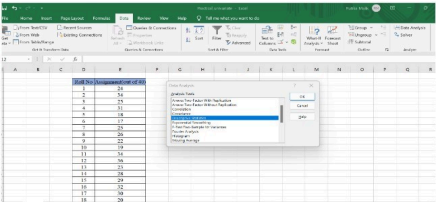
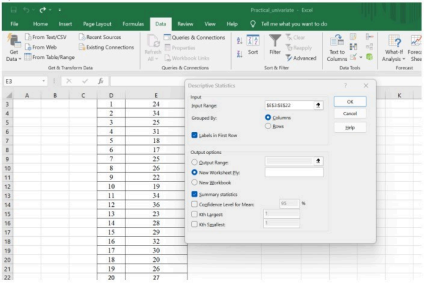
****

**Practical No.5**

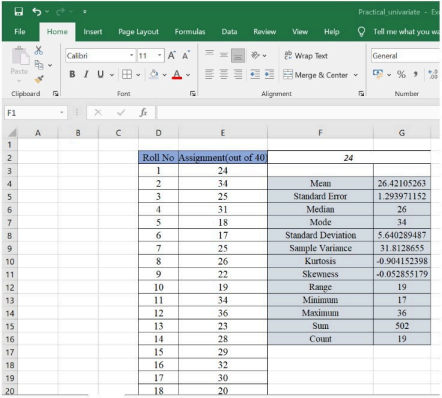
**Aim: Import a dataset and perform univariate analysis on the numeric columns to analyse the shape of the data. Write inference of the output. [Python , R or Excel].**

**Steps:**

a) Import a dataset

b) Click on Data Tab→Data Analysis -→Descriptive Statistics

C)



**Practical No.6**

**Aim: Demonstrate Hypothesis testing, and ANOVA using a dataset [Use Python, R or Excel]**

**Steps:**

The Data Analysis Toolpak in Excel

If you’re analysing data in Excel, then it’s natural to make use of the tools that Microsoft provides for you. One of the less obvious features in Excel is the Data Analysis Toolpak. The Toolpak is an Excel add-in from Microsoft that’s included with Excel, but isn’t turned on. Here’s how to turn it on in the Microsoft Windows operating system.

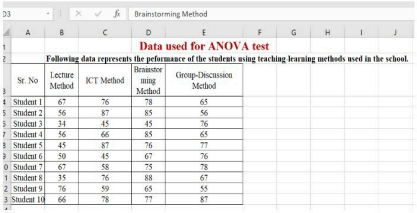
1. Choose **File**, then **Options**

2. In the **Excel Options Window**, choose **Add-ins**

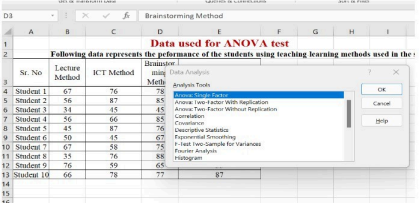
3. Next to **Manage**, select **Excel Add-ins** and click **Go**

4. In the **Add-ins** window, select **Analysis ToolPak** and click **OK**

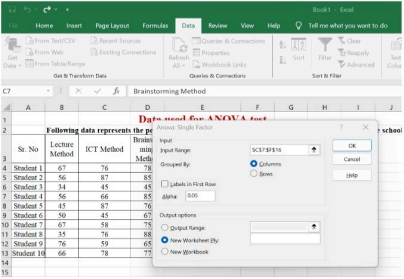
5. Enter data



6. Click on Data Tab and click Data Analysis.



7. Select the range for ANOVA test.



8. Final Interpretation.



**Practical No.7**

**Aim: Demonstrate correlation analysis. Use heatmap for visualization. Write inferences.**

**Code:**

# import modules

import matplotlib.pyplot as mp

import pandas as pd

import seaborn as sb

# import file with data

data = pd.read\_csv("CSV file path")

# prints data that will be plotted

# columns shown here are selected by corr() since

# they are ideal for the plot

print(data.corr())

# plotting correlation heatmap

dataplot = sb.heatmap(data.corr(), cmap="YlGnBu", annot=True)

# displaying heatmap

mp.show()

**Output:**

****

**Practical No.8**

**Aim: Import csv or Excel Dataset and demonstrate data wrangling, view shape, dimension, column names of the dataset, ways to select data using column number, column names, simple and compound conditional selection , update and modify dataset.**

**Code:**

import pandas as pd

# Load the CSV file

df = pd.read\_csv("C:\\Users\\devil.DESKTOP-P2AKKQD\\OneDrive\\Desktop\\ET.DJ\\ET PRACT 8.csv") # Replace with your file path if necessary

# Display the dataset

print(df)

# Display basic dataset info

print("Shape of dataset:", df.shape) # Number of rows and columns

print("Number of dimensions:", df.ndim) # Dimensions of the dataset

print("Column names:", df.columns.tolist()) # List of column names

# Select columns by name

selected\_by\_name = df[["Name", "Age", "Salary"]]

# Select columns by index positions

selected\_by\_index = df.iloc[:, [1, 2, 4]] # Select Name, Age, Salary columns print(selected\_by\_name)

print(selected\_by\_index)

# Filter rows where Age > 25

age\_filter = df[df["Age"] > 25]

print(age\_filter)

# Filter rows where Salary > 50000 AND Department is IT

compound\_filter = df[(df["Salary"] > 50000) & (df["Department"] ==

"IT")] print(compound\_filter)

# Add a new column

df["Bonus"] = df["Salary"] \* 0.1 # 10% bonus of salary

# Update a specific value (e.g., set Salary to 55000 for Bob)

df.loc[df["Name"] == "Bob", "Salary"] = 55000

print(df)

df.to\_csv("modified\_dataset.csv", index=False)

**Output:**

****

****

**Practical No.9**

**Aim: Demonstrate group by summary operations and sorting techniques.**

**Code:**

import pandas as pd

# Creating a sample dataset

data = {

'Team': ['A', 'A', 'B', 'B', 'C', 'C'],

'Player': ['John', 'Mike', 'Sara', 'Anna', 'Tom', 'Jerry'],

'Score': [10, 15, 10, 20, 15, 10],

'Age': [23, 25, 22, 24, 23, 21]

}

df = pd.DataFrame(data)

# Group by 'Team' and calculate summary statistics

grouped = df.groupby('Team').agg({

'Score': ['mean', 'sum', 'max'],

'Age': ['mean', 'min', 'max']

})

print("Group by summary operations:")

print(grouped)

# Sorting the dataset by 'Score' in descending order

sorted\_df = df.sort\_values(by='Score', ascending=False)

print("\nSorted dataset by Score:")

print(sorted\_df)

# Save the dataset to a CSV file

df.to\_csv('sample\_dataset.csv',index=False)

**Output:**

****