ASSIGNMENT 2

1 . What are the different ways of creating Data Frame in pandas? Explain with examples.

Different Ways of Creating a DataFrame in Pandas

Pandas provides various ways to create a DataFrame:

a. From a Dictionary of Lists

You can create a DataFrame by passing a dictionary where keys are column names, and values are lists of data.

```
import pandas as pd

data = {

    'Name': ['Alice', 'Bob', 'Charlie'],

    'Age': [25, 30, 35],

    'City': ['New York', 'Los Angeles', 'Chicago']

}

df = pd.DataFrame(data)

print(df)
```

b. From a List of Dictionaries

A DataFrame can also be created from a list of dictionaries, where each dictionary represents a row.

```
data = [

    {'Name': 'Alice', 'Age': 25, 'City': 'New York'},

    {'Name': 'Bob', 'Age': 30, 'City': 'Los Angeles'},

    {'Name': 'Charlie', 'Age': 35, 'City': 'Chicago'}

]

df = pd.DataFrame(data)

print(df)
```

c. From a List of Lists with Column Names

You can use a list of lists and specify column names.

```
data = [

    ['Alice', 25, 'New York'],

    ['Bob', 30, 'Los Angeles'],

    ['Charlie', 35, 'Chicago']

]
```

```
df = pd.DataFrame(data, columns=['Name', 'Age', 'City'])
print(df)
```

d. From a Dictionary of Series

A DataFrame can also be created from a dictionary of Pandas Series objects.

```
data = {
    'Name': pd.Series(['Alice', 'Bob', 'Charlie']),
    'Age': pd.Series([25, 30, 35]),
    'City': pd.Series(['New York', 'Los Angeles', 'Chicago'])
}
df = pd.DataFrame(data)
print(df)
```

e. From Existing CSV or Excel File

If you have data stored in a file, you can read it directly into a DataFrame.

```
df = pd.read_csv('file.csv')  # Read from CSV file
df = pd.read_excel('file.xlsx')  # Read from Excel file
print(df)
```

2. How will you add a column to the Existing Data Frames in Panda?

To add a column to an existing DataFrame in Pandas, you can follow these approaches:

1. Using a constant value

If you want to add a new column with a constant value for all rows:

```
import pandas as pd
# Example DataFrame
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
# Add a new column 'C' with the constant value 10
df['C'] = 10
print(df)
```

2. Using an existing column or a computation

You can create a new column based on the values of other columns or through a computation:

```
# Add a new column 'C' as the sum of columns 'A' and 'B'
```

```
df['C'] = df['A'] + df['B']

print(df)
```

3. Using a list or array

If you have a list or array of values to assign to the new column (ensure the length matches the number of rows in the DataFrame) :

```
# Add a new column 'C' with specific values from a list

df['C'] = [7, 8, 9]

print(df)
```

4. Using assign() method

The assign() method is another way to add a column. It returns a new DataFrame, leaving the original DataFrame unchanged:

```
# Add a new column 'C' using the assign method

df = df.assign(C=[7, 8, 9])

print(df)
```

5. Using insert() method

You can specify the position to insert the column in the DataFrame:

```
# Add a new column 'C' at the second position

df.insert(1, 'C', [7, 8, 9])

print(df)
```

Each of these methods will allow you to add a new column to your existing DataFrame.