Vishwas mishra
21BCE0959

```python
import pandas as pd
import numpy as np

# Task 1: Create a pandas dataframe
data = {
    'Feature1': np.random.rand(10),
    'Feature2': np.random.randint(1, 100, 10),
    'Feature3': np.random.choice(['A', 'B', 'C', 'D'], 10),
    'Feature4': np.random.randn(10),
    'Feature5': np.random.uniform(0, 1, 10)
}

df = pd.DataFrame(data)
df
```

```
   Feature1  Feature2 Feature3  Feature4  Feature5
0  0.739493        39        B -2.209081  0.933679
1  0.055776        90        D  0.283296  0.116273
2  0.585586        72        A  0.240210  0.952893
3  0.065296        86        B -0.097260  0.575096
4  0.119688        72        A  0.860834  0.052219
5  0.640211        27        B  0.909632  0.428420
6  0.483359        10        A -1.350128  0.578097
7  0.063325        20        C  0.288137  0.420774
8  0.887235        99        C  1.739494  0.195495
```

```python
# Task 2: Check the info of 'df'
print("Task 2:")
print(df.info())
```

```
Task 2:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
 0   Feature1  10 non-null     float64
 1   Feature2  10 non-null     int64
 2   Feature3  10 non-null     object
 3   Feature4  10 non-null     float64
 4   Feature5  10 non-null     float64
dtypes: float64(3), int64(1), object(1)
memory usage: 528.0+ bytes
None
```

```python
# Task 3: Check the descriptive statistics of 'df'
print("\nTask 3:")
print(df.describe())
```

```
Task 3:
        Feature1    Feature2    Feature4    Feature5
count  10.000000   10.000000   10.000000   10.000000
mean    0.428140   60.100000   -0.065065    0.509968
std     0.320513   32.814123    1.211821    0.329315
min     0.055776   10.000000   -2.209081    0.052219
25%     0.078894   30.000000   -1.011156    0.251814
50%     0.534472   72.000000    0.261753    0.501758
75%     0.641123   86.000000    0.717660    0.779574
max     0.887235   99.000000    1.739494    0.952893
```

```python
# Task 4: Check the 4th index observation with 'loc' slicing operator
print("\nTask 4:")
print(df.loc[3])   # Note: Indexing is 0-based, so the 4th index is 3
```

```
Task 4:
Feature1     0.065296
Feature2           86
Feature3            B
Feature4     -0.09726
Feature5     0.575096
Name: 3, dtype: object
```

```python
# Task 5: Check the null values in 'df'
print("\nTask 5:")
print(df.isnull().sum())
```

```
Task 5:
Feature1    0
Feature2    0
Feature3    0
Feature4    0
Feature5    0
dtype: int64
```