pandas-Copy1

September 8, 2024

[2]:
```
import pandas as pd
import numpy as np
```

[3]:
```
# Q1. Create a Pandas Series that contains the following data: 4, 8, 15, 16,␣
 ↪23, and 42. Then, print the series.
z= pd.array([4, 8, 15, 16, 23,42])
pd.Series(z)
```

[3]:
```
0     4
1     8
2    15
3    16
4    23
5    42
dtype: Int64
```

[4]:
```
#Q2.Q2. Create a variable of list type containing 10 elements in it, and apply␣
 ↪pandas.Series function on the
# variable print it.

z=[4, 8, 15, 16, 23,42,15, 16, 23,42]
pd.Series(z)
```

[4]:
```
0     4
1     8
2    15
3    16
4    23
5    42
6    15
7    16
8    23
9    42
dtype: int64
```

[5]:
```
# Q3. Create a Pandas DataFrame that contains the following data:
data_dict = {
```

```
    "Name": ["Alice", "Bob", "Claire"],
    "Age": [25, 30, 27],
    "Gender": ["Female", "Male", "Female"]
}

# Converting to DataFrame
df = pd.DataFrame(data_dict)
df
```

[5]:

	Name	Age	Gender
0	Alice	25	Female
1	Bob	30	Male
2	Claire	27	Female

Q4. What is 'DataFrame' in pandas and how is it different from pandas.series? Explain with an example.

A DataFrame in Pandas is a two-dimensional, size-mutable, and heterogeneous tabular data structure with labeled axes (rows and columns). It is similar to a table in a relational database or an Excel spreadsheet. Each column in a DataFrame is a Pandas Series, and the DataFrame itself is a collection of Series objects.

A Pandas Series, on the other hand, is a one-dimensional array with an associated data label (index). It is similar to a single column in an Excel sheet or a single column in a DataFrame.

Key Differences: Dimensionality:

A Series is one-dimensional (1D), while a DataFrame is two-dimensional (2D). Structure:

A Series contains only one list of data, while a DataFrame contains multiple lists (or columns) of data, where each column is a Series. Use Case:

A Series is used for a single list of values, while a DataFrame is used for multiple lists or a table-like structure with rows and columns.

[6]:
```
# examples:-series
z=[4, 8, 15, 16, 23,42,15, 16, 23,42]
pd.Series(z)
```

[6]:
```
0     4
1     8
2    15
3    16
4    23
5    42
6    15
7    16
8    23
9    42
dtype: int64
```

[7]:
```
# example dataframe:-
data_dict = {
    "Name": ["Alice", "Bob", "Claire"],
    "Age": [25, 30, 27],
    "Gender": ["Female", "Male", "Female"]
}

# Converting to DataFrame
df = pd.DataFrame(data_dict)
df
```

[7]:

	Name	Age	Gender
0	Alice	25	Female
1	Bob	30	Male
2	Claire	27	Female

Q5. What are some common functions you can use to manipulate data in a Pandas DataFrame? Can you give an example of when you might use one of these functions?

ANS:-head() tail() describe() mean() groupby() example:-

[8]:
```
# Example DataFrame of sales
data_dict = {
    "Region": ["East", "West", "East", "West", "North", "South"],
    "Sales": [200, 340, 150, 400, 120, 300]
}
df = pd.DataFrame(data_dict)

# Grouping by Region and calculating the mean sales
df.groupby('Region').mean()
```

[8]:

Region	Sales
East	175.0
North	120.0
South	300.0
West	370.0

Q6. Which of the following is mutable in nature Series, DataFrame, Panel?

In Pandas, the following objects are mutable:

- **Series**: Mutable
- **DataFrame**: Mutable
- **Panel**: Mutable (though Panels have been deprecated as of Pandas 0.25.0)

This means both `Series` and `DataFrame` objects can be modified after they are created (e.g., you can change, add, or delete elements). However, as of the newer versions of Pandas, `Panel` has been replaced with more powerful tools like `MultiIndex` for handling higher-dimensional data.

Q7. Create a DataFrame using multiple Series. Explain with an example.

[1]:
```
import pandas as pd

# Creating individual Series
names = pd.Series(['Alice', 'Bob', 'Charlie'])
ages = pd.Series([25, 30, 22])
cities = pd.Series(['New York', 'Los Angeles', 'Chicago'])

# Creating a DataFrame using the Series
df = pd.DataFrame({
    'Name': names,
    'Age': ages,
    'City': cities
})

# Displaying the DataFrame
print(df)
```

```
      Name  Age         City
0    Alice   25     New York
1      Bob   30  Los Angeles
2  Charlie   22      Chicago
```

[ ]: