pandas

September 1, 2024

0.1 1. Getting Familiar with Pandas

[1]:
```
import pandas as pd
import numpy as np

# 1. Getting Familiar with Pandas
# Creating DataFrames and Series
data_dict = {'Name': ['AxeKing', 'Hilter', 'CharlieKing'], 'Age': [55, 90, 55]}
df = pd.DataFrame(data_dict)
series = pd.Series([1, 2, 3, 4])
```

[2]:
```
# Selecting data and modifying DataFrames
selected_data = df[df['Age'] > 28]
df['Age'] = df['Age'] + 1  # Incrementing age by 1
```

[10]:
```
# 2. Data Handling with Pandas
# Reading data from a CSV file
df.to_csv('sample_data.csv', index=False)
df_loaded = pd.read_csv('sample_data.csv')
print(df_loaded)
```

```
          Name  Age
0      AxeKing   56
1       Hilter   91
2  CharlieKing   56
```

[11]:
```
# Handling missing data
df_loaded.loc[1, 'Age'] = np.nan  # Introduce a missing value
df_filled = df_loaded.fillna(df_loaded['Age'].mean())  # Fill missing values␣
 ↪with the mean
print(df_filled)
```

```
          Name   Age
0      AxeKing  56.0
1       Hilter  56.0
2  CharlieKing  56.0
```

[12]:
```
# Removing duplicates
df_filled = df_filled.drop_duplicates()
print(df_filled)
```

```
          Name   Age
0      AxeKing  56.0
1       Hilter  56.0
2  CharlieKing  56.0
```

[13]:
```
# Data type conversion
df_filled['Age'] = df_filled['Age'].astype(int)
print(df_filled['Age'])
```

```
0    56
1    56
2    56
Name: Age, dtype: int32
```

[ ]:
```

```

0.2 3. Data Analysis with Pandas

[14]:
```
import pandas as pd
import numpy as np
df=pd.DataFrame(np.arange(15))
# Assuming df is your DataFrame
summary_stats = df.describe()
df1 = pd.DataFrame({'key': ['A', 'B', 'C'], 'value1': [1, 2, 3]})
df2 = pd.DataFrame({'key': ['A', 'B', 'D'], 'value2': [4, 5, 6]})
print(summary_stats)
print(df)
```

```
              0
count  15.000000
mean    7.000000
std     4.472136
min     0.000000
25%     3.500000
50%     7.000000
75%    10.500000
max    14.000000
     0
0    0
1    1
2    2
3    3
4    4
```

```
5    5
6    6
7    7
8    8
9    9
10  10
11  11
12  12
13  13
14  14
```

[15]:
```
mean_value = df[0].mean()
median_value = df[0].median()
std_dev = df[0].std()
print(mean_value)
print(median_value)
print(std_dev)
```

```
7.0
7.0
4.47213595499958
```

[19]:
```
merged_df = pd.merge(df1, df2, on='key', how='inner')
print(merged_df)
```

```
  key  value1  value2
0   A       1       4
1   B       2       5
```

[20]:
```
df1 = pd.DataFrame({'value1': [1, 2, 3]}, index=['A', 'B', 'C'])
df2 = pd.DataFrame({'value2': [4, 5, 6]}, index=['A', 'B', 'D'])
joined_df = df1.join(df2, how='inner')
print(joined_df)
```

```
   value1  value2
A       1       4
B       2       5
```

[21]:
```
concatenated_df = pd.concat([df1, df2], axis=0)  # For row-wise concatenation
print(concatenated_df)
```

```
   value1  value2
A     1.0     NaN
B     2.0     NaN
C     3.0     NaN
A     NaN     4.0
B     NaN     5.0
D     NaN     6.0
```

[ ]:

0.2.1 How Pandas Helps Data Science Professionals

Pandas is a powerful Python library for data manipulation and analysis, offering significant advantages over traditional data structures like lists and dictionaries:

Data Handling and Analysis: - **Efficient Structures:** Pandas' Series (1D) and DataFrame (2D) handle large datasets and complex operations seamlessly. - **Ease of Use:** Simplifies data filtering, aggregation, and transformation with concise code.

Advantages Over Traditional Structures: - **Performance:** Faster vectorized operations compared to looping through lists. - **Missing Data:** Built-in functions for detecting and handling missing values. - **Advanced Manipulation:** Easy merging, joining, and grouping. - **Data Alignment:** Automatic alignment based on indexes.

0.2.2 Real-World Applications of Pandas

Data Cleaning: - **Handling Missing Data:** Use `dropna()` and `fillna()`. - **Data Transformation:** Functions like `apply()`, `map()`, and `replace()`.

Exploratory Data Analysis (EDA): - **Summary Statistics:** Use `describe()` for a quick overview. - **Visualization:** Integrates with Matplotlib and Seaborn for plotting data.

Applications: - **Finance:** Analyzing stock prices and financial records. - **Healthcare:** Managing patient records and clinical data. - **E-commerce:** Analyzing customer behavior and tracking sales.

[ ]:

[ ]: