9913_python_exp10

March 15, 2024

[3]:
```
import pandas as pd
```

[6]:
```
data = {'Name': ['John', 'Alice', 'Bob'],
        'Age': [25, 30, 35],
        'City': ['New York', 'Los Angeles', 'Chicago']}

df = pd.DataFrame(data)

print(df)
```

```
    Name  Age         City
0   John   25     New York
1  Alice   30  Los Angeles
2    Bob   35      Chicago
```

[7]:
```
df['Gender'] = ['Male', 'Female', 'Male']

print(df)
```

```
    Name  Age         City  Gender
0   John   25     New York    Male
1  Alice   30  Los Angeles  Female
2    Bob   35      Chicago    Male
```

[8]:
```
# Set column 'Name' as index
df.set_index('Name', inplace=True)

print(df)
```

```
       Age         City  Gender
Name
John    25     New York    Male
Alice   30  Los Angeles  Female
Bob     35      Chicago    Male
```

[9]:
```
# Rename columns
df.rename(columns={'Age': 'Years', 'City': 'Location', 'Gender': 'Sex'},␣
 ↪inplace=True)
```

```
print(df)
```

```
       Years     Location     Sex
Name
John      25     New York    Male
Alice     30  Los Angeles  Female
Bob       35      Chicago    Male
```

[10]:
```
# Filter rows where age is greater than 30
filtered_df = df[df['Years'] > 30]

print(filtered_df)
```

```
      Years Location   Sex
Name
Bob      35  Chicago  Male
```

[11]:
```
# Sort DataFrame based on values in column 'Years'
sorted_df = df.sort_values(by='Years')

print(sorted_df)
```

```
       Years     Location     Sex
Name
John      25     New York    Male
Alice     30  Los Angeles  Female
Bob       35      Chicago    Male
```

[12]:
```
# Create another DataFrame with a common index
other_data = {'Name': ['John', 'Alice'],
              'Income': [50000, 60000]}
other_df = pd.DataFrame(other_data).set_index('Name')

# Merge two DataFrames
merged_df = df.merge(other_df, left_index=True, right_index=True, how='inner')

# Print merged DataFrame
print(merged_df)
```

```
       Years     Location     Sex  Income
Name
John      25     New York    Male   50000
Alice     30  Los Angeles  Female   60000
```

[33]:
```
# Read csv file
diabetes_df = pd.read_csv('https://raw.githubusercontent.com/YBI-Foundation/
 ↪Dataset/main/Diabetes%20Missing%20Data.csv')

# Display first five rows
```

```
print("First five rows:")
print(diabetes_df.head())

# Display last five rows
print("\nLast five rows:")
print(diabetes_df.tail())
```

```
First five rows:
   Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
0         6    148.0          72.0       35.0            NaN  33.6
1         1     85.0          66.0       29.0            NaN  26.6
2         8    183.0          64.0        NaN            NaN  23.3
3         1     89.0          66.0       23.0           94.0  28.1
4         0    137.0          40.0       35.0          168.0  43.1

   Diabetes_Pedigree  Age  Class
0              0.627   50      1
1              0.351   31      0
2              0.672   32      1
3              0.167   21      0
4              2.288   33      1

Last five rows:
     Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
763        10    101.0          76.0       48.0          180.0  32.9
764         2    122.0          70.0       27.0            NaN  36.8
765         5    121.0          72.0       23.0          112.0  26.2
766         1    126.0          60.0        NaN            NaN  30.1
767         1     93.0          70.0       31.0            NaN  30.4

     Diabetes_Pedigree  Age  Class
763              0.171   63      0
764              0.340   27      0
765              0.245   30      0
766              0.349   47      1
767              0.315   23      0
```

[26]:
```
summary_statistics = df.describe()
print("\nSummary statistics:")
print(summary_statistics)
```

```

Summary statistics:
       Years
count    3.0
mean    30.0
std      5.0
min     25.0
```

```
25%     27.5
50%     30.0
75%     32.5
max     35.0
```

[34]:
```
url = "https://raw.githubusercontent.com/YBI-Foundation/Dataset/main/
 ↪Diabetes%20Missing%20Data.csv"
df = pd.read_csv(url)

missing_values = df.isnull().sum()
print("\nMissing values:")
print(missing_values)

# Replace missing values by mean of the column
for column in df.columns:
    if df[column].dtype != 'object':  # Check if column is numeric
        df[column] = df[column].fillna(df[column].mean())

# Verify missing values after replacement
print("\nMissing values after replacement:")
print(df.isnull().sum())

# Display first few rows of the updated DataFrame
print("\nUpdated DataFrame:")
print(df.head())
```

```
Missing values:
Pregnant              0
Glucose               5
Diastolic_BP         35
Skin_Fold           227
Serum_Insulin       374
BMI                  11
Diabetes_Pedigree     0
Age                   0
Class                 0
dtype: int64

Missing values after replacement:
Pregnant            0
Glucose             0
Diastolic_BP        0
Skin_Fold           0
Serum_Insulin       0
BMI                 0
Diabetes_Pedigree   0
Age                 0
```

```
Class               0
dtype: int64

Updated DataFrame:
   Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
0         6    148.0          72.0   35.00000     155.548223  33.6
1         1     85.0          66.0   29.00000     155.548223  26.6
2         8    183.0          64.0   29.15342     155.548223  23.3
3         1     89.0          66.0   23.00000      94.000000  28.1
4         0    137.0          40.0   35.00000     168.000000  43.1

   Diabetes_Pedigree  Age  Class
0              0.627   50      1
1              0.351   31      0
2              0.672   32      1
3              0.167   21      0
4              2.288   33      1
```

[38]:
```
# Create a list of data
data = [45, 78, 92, 35, 68]

# Custom index labels
custom_index = ['1', '2', '3', '4', '5']

# Create a Series with custom index labels
series = pd.Series(data, index=custom_index)

print("Series:")
print(series)

# Calculate sum
sum_value = series.sum()
print("\nSum:", sum_value)

# Calculate mean
mean_value = series.mean()
print("Mean:", mean_value)

# Calculate median
median_value = series.median()
print("Median:", median_value)

# Calculate standard deviation
std_deviation_value = series.std()
print("Standard Deviation:", std_deviation_value)
```

```
Series:
1    45
```

```
2    78
3    92
4    35
5    68
dtype: int64

Sum: 318
Mean: 63.6
Median: 68.0
Standard Deviation: 23.43714999738663
```

[39]:

```
# Create a Series
data = [45, 78, 92, 35, 68]
custom_index = ['1', '2', '3', '4', '5']
series = pd.Series(data, index=custom_index)

# Find maximum value and its index label
max_value = series.max()
max_index = series.idxmax()

# Find minimum value and its index label
min_value = series.min()
min_index = series.idxmin()

# Print maximum and minimum values with their index labels
print("Maximum value:", max_value, "at index label:", max_index)
print("Minimum value:", min_value, "at index label:", min_index)

# Sort the Series
sorted_series = series.sort_values()

# Print sorted Series
print("\nSorted Series:")
print(sorted_series)
```

```
Maximum value: 92 at index label: 3
Minimum value: 35 at index label: 4

Sorted Series:
4    35
1    45
5    68
2    78
3    92
dtype: int64
```