Assignment-Lab4

Name – Ningaraddi
SapId- 500110910

Lab 4: Python Library Pandas

Q1. Loading and Inspecting Data:
• Load various data formats (CSV, Excel, JSON) into Pandas DataFrames
.• Explore DataFrame attributes like shape, columns, dtypes, head, tail, info, describe.
• Practice selecting columns and rows using different methods (indexing, slicing, loc, iloc).

Q2. Data Cleaning and Preparation
• Identify missing values using isnull and isna.
• Handle missing values using fillna, dropna, interpolation.
• Apply scaling techniques (min-max, z-score) to numerical columns.
• Create dummy variables for categorical columns.

Q3. Aggregation and Grouping:
• Calculate summary statistics (mean, median, count, etc.) using groupby.
• Create pivot tables for data summarization.
• Combine DataFrames using concat, merge, and join.
• Practice different join types (inner, outer, left, right)

In [1]:
```
import pandas as pd
from sklearn.datasets import load_iris

# Load the iris dataset from sklearn iris
= load_iris()


df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

# Add the target variable to the DataFrame df['species']
= iris.target

# Map numerical target values to actual species names
df['species'] = df['species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})
print(df.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
   3                4.6               3.1               1.5               0.2

   4                5.0               3.6               1.4               0.2

  species  0
setosa  1
setosa
2  setosa
3  setosa
4  setosa
```

In [2]:
```
# Display the shape of the DataFrame
print(df.shape)
```

```
(150, 5)
```

In [3]:
```
#columns print(df.columns)
```

```
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'species'],
dtype='object')
```

In [5]:
```
#data types of each column print(df.dtypes)
```

```
sepal length (cm)    float64 sepal
width (cm)     float64 petal
length (cm)    float64 petal
width (cm)     float64 species
object dtype: object
```

In [6]:
```
#Display the first 5 rows print(df.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
   3                4.6               3.1               1.5               0.2

   4                5.0               3.6               1.4               0.2

  species  0
setosa
1  setosa  2
setosa
3  setosa
```

```
4  setosa
```

In [7]:
```
print(df.tail())
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 145
6.7               3.0               5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
                   149               5.9                3.0               5.1
                   1.8

       species  145
virginica  146
virginica
147  virginica
148  virginica
149  virginica
```

In [8]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149 Data
columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  -
--  ------             --------------  -----   0
sepal length (cm)  150 non-null    float64
1   sepal width (cm)   150 non-null    float64
2   petal length (cm)  150 non-null    float64  3   petal width
    (cm)   150 non-null    float64  4   species            150
    non-null    object dtypes: float64(4), object(1) memory
    usage: 6.0+ KB None
```

In [9]:
```
print(df.describe())
```

```
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000   mean
5.843333         3.057333          3.758000   std           0.828066
0.435866          1.765298   min           4.300000         2.000000
1.000000   25%           5.100000         2.800000          1.600000
50%             5.800000          3.000000           4.350000   75%
6.400000         3.300000          5.100000   max           7.900000
4.400000            6.900000

       petal width (cm)
count        150.000000
mean           1.199333  std
0.762238  min
0.100000  25%
0.300000
50%            1.300000  75%
1.800000  max
2.500000
```

In [10]:

```
import pandas as pd

#dataframe
print("DataFrame:")
print(df.head())

#Indexing

# Selecting a single column by column name print("\nSelecting
single column (by name):") print(df['sepal length
(cm)'].head())

# Selecting multiple columns by column names print("\nSelecting
multiple columns (by names):") print(df[['sepal length (cm)',
'species']].head())

# Selecting a single row by index print("\nSelecting
single row (by index):") print(df.iloc[0])

# Selecting multiple rows by index range
print("\nSelecting multiple rows (by index range):") print(df.iloc[0:3])

# 2.Slicing

# Selecting a range of rows print("\nSelecting
a range of rows:") print(df[5:10])

# Selecting a range of columns using slicing
print("\nSelecting a range of columns (using column index):") print(df.iloc[:,
0:3].head())

# 3.using loc
# Selecting rows and specific columns by labels
print("\nSelecting specific rows and columns (by labels):") print(df.loc[0:2,
['sepal length (cm)', 'species']])

# Conditional selection with `loc`
print("\nConditional selection (species = 'setosa'):") print(df.loc[df['species']
== 'setosa'].head())

# 4.Using iloc
```

```
# Selecting rows and columns by integer-location based indexing
print("\nSelecting specific rows and columns (by index):") print(df.iloc[0:3,
[0, 4]])  # First 3 rows, 1st and 5th columns

# Conditional selection with `iloc` (combination with boolean indexing)
print("\nConditional selection with iloc:") setosa_rows =
df[df['species'] == 'setosa'] print(setosa_rows.iloc[0:3])

# Selecting rows where 'sepal length (cm)' is greater than 5.0
long_sepal_rows = df[df['sepal length (cm)'] > 5.0] print("\nRows
with 'sepal length (cm)' > 5.0:")
print(long_sepal_rows.iloc[0:3])
```

```
DataFrame:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2

4                5.0               3.6                1.4               0.2

  species  0
setosa  1
setosa
2  setosa
3  setosa
4  setosa

Selecting single column (by name):
0    5.1
1    4.9
2    4.7
3    4.6
4    5.0
Name: sepal length (cm), dtype: float64

Selecting multiple columns (by names):
   sepal length (cm) species
0                5.1  setosa
1                4.9  setosa
2                 4.7  setosa
3                 4.6  setosa
4                 5.0  setosa

Selecting single row (by index):
sepal length (cm)       5.1
sepal width (cm)        3.5

petal length (cm)       1.4
petal width (cm)        0.2
species              setosa
Name: 0, dtype: object

Selecting multiple rows (by index range):
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \

0                5.1               3.5                1.4               0.2

1                4.9               3.0                1.4               0.2

2                4.7               3.2                1.3               0.2

  species
0  setosa  1  setosa  2  setosa  Selecting a range of rows:    sepal length (cm)
sepal width (cm)  petal length (cm)  petal width (cm)  \
5                5.4               3.9                1.7               0.4
6                4.6               3.4                1.4               0.3
7                5.0               3.4                1.5               0.2
8                4.4               2.9                1.4               0.2

9                4.9               3.1                1.5               0.1

  species  5
setosa
6  setosa
7  setosa  8  setosa
9  setosa
```

```
Selecting a range of columns (using column index):
   sepal length (cm)  sepal width (cm)  petal length (cm) 0
5.1               3.5                1.4
1                 4.9               3.0                1.4
2                 4.7               3.2                1.3 3
                  4.6               3.1                1.5
4                 5.0               3.6                1.4

Selecting specific rows and columns (by labels):
   sepal length (cm) species
0                 5.1  setosa
1                 4.9  setosa
2                 4.7  setosa

Conditional selection (species = 'setosa'):
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2   1                4.9
3.0                 1.4               0.2
2                 4.7               3.2                1.3               0.2
   3                 4.6               3.1                1.5               0.2

   4                 5.0               3.6                1.4               0.2

  species  0
setosa
1  setosa  2
setosa
3  setosa
4  setosa

Selecting specific rows and columns (by index):
   sepal length (cm) species
0                 5.1  setosa
1                 4.9  setosa
2                 4.7  setosa

Conditional selection with iloc:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                 5.1               3.5                1.4               0.2
   1                 4.9               3.0                1.4               0.2

   2                 4.7               3.2                1.3               0.2

  species  0
setosa
1  setosa
2  setosa

Rows with 'sepal length (cm)' > 5.0:
    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2

5                 5.4               3.9                1.7               0.4

10                5.4               3.7                1.5               0.2

   species  0
setosa
5   setosa
```

10 setosa In

[11]:

```
# Identifying the  missing values using isnull()
print("Missing values (using isnull()):") print(df.isnull())

# Counting the missing values in each column using isnull() print("\nCount
of missing values in each column (using isnull()):")

print(df.isnull().sum())

# Count total missing values in the DataFrame using isnull()
print("\nTotal count of missing values in DataFrame (using isnull()):")
print(df.isnull().sum().sum())

# Identify missing values using isna() print("\nMissing
values (using isna()):") print(df.isna())

# Counting missing values in each column using isna()
print("\nCount of missing values in each column (using isna()):")
print(df.isna().sum())

# Counting total missing values in the DataFrame using isna() print("\nTotal
count of missing values in DataFrame (using isna()):")
print(df.isna().sum().sum())
```

```
Missing values (using isnull()):
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
False             False              False             False
1                 False             False             False             False
2                 False             False             False             False
3                 False             False             False             False
                  4                False             False             False
                  False   ..                 ...               ...
                  ...               ...
145               False             False             False             False
146               False             False             False             False
                  147              False             False             False
                  False
148              False             False             False             False
149              False             False             False             False

     species  0
False
1      False
2      False  3      False
4      False  ..
...
145    False  146
False
147    False
148    False
149    False

[150 rows x 5 columns]

Count of missing values in each column (using isnull()):
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
species              0
dtype: int64

Total count of missing values in DataFrame (using isnull()):
0

Missing values (using isna()):
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
False             False              False             False
1                 False             False             False             False
2                 False             False             False             False
3                 False             False             False             False
                  4                False             False             False
                  False   ..                 ...               ...
                  ...               ...
145               False             False             False             False
146               False             False             False             False
                  147              False             False             False
                  False
148              False             False             False             False
149              False             False             False             False

     species  0
False
1      False
2      False  3      False
4      False  ..
...
145    False  146
False
```

```
147    False
148    False
149    False

[150 rows x 5 columns]

Count of missing values in each column (using isna()):
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
species              0
dtype: int64

Total count of missing values in DataFrame (using isna()):
         0In [12]:
```

```
# Handling the missing values using fillna
print("DataFrame after filling missing values with a specified value:") df_filled
= df.fillna(value={'sepal length (cm)': 0, 'species': 'unknown'})
print(df_filled.head(10))

# Handling the  missing values using dropna
print("\nDataFrame after dropping rows with any missing values:")
df_dropped = df.dropna() print(df_dropped.head(10))

# Handling missing values using interpolation
print("\nDataFrame after interpolating missing values:")
df_interpolated = df.interpolate() print(df_interpolated.head(10))
```

```
DataFrame after filling missing values with a specified value:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
         14.9              3.0               1.4              0.2
         24.7              3.2               1.3              0.2
3                4.6              3.1               1.5              0.2
4                5.0              3.6               1.4              0.2
5                5.4              3.9               1.7              0.4
6                4.6              3.4               1.4              0.3
7                5.0              3.4               1.5              0.2

8                4.4              2.9               1.4              0.2

9                4.9              3.1               1.5              0.1

  species  0
setosa
1  setosa
2  setosa
3  setosa  4  setosa
5  setosa
6  setosa
7  setosa  8  setosa
9  setosa

DataFrame after dropping rows with any missing values:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
1                4.9              3.0               1.4              0.2
2                4.7              3.2               1.3              0.2   3
                 4.6              3.1               1.5              0.2
4                5.0              3.6               1.4              0.2
5                5.4              3.9               1.7              0.4
6                4.6              3.4               1.4              0.3   7
                 5.0              3.4               1.5              0.2
8                4.4              2.9               1.4              0.2

9                4.9              3.1               1.5              0.1

  species  0
setosa
1  setosa
2  setosa  3  setosa
4  setosa
5  setosa
6  setosa  7  setosa
8  setosa
9  setosa

DataFrame after interpolating missing values:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
1                4.9              3.0               1.4              0.2   2
4.7               3.2                1.3               0.2
3                4.6              3.1               1.5              0.2
4                5.0              3.6               1.4              0.2
5                5.4              3.9               1.7              0.4
6                4.6              3.4               1.4              0.3
7                5.0              3.4               1.5              0.2
8                4.4              2.9               1.4              0.2

9                4.9              3.1               1.5              0.1

                 species

0  setosa
```

```
1  setosa  2  setosa
3  setosa
4  setosa
5  setosa  6  setosa
       7setosa
```

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Create instances of the scalers
min_max_scaler = MinMaxScaler() z_score_scaler
= StandardScaler()


# Min-Max Scaling
numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns
df_min_max_scaled = df.copy()
df_min_max_scaled[numerical_cols] = min_max_scaler.fit_transform(df[numerical_cols

print("\nDataFrame after Min-Max Scaling:") print(df_min_max_scaled.head())

# Applying Z-score df_z_score_scaled
= df.copy()
df_z_score_scaled[numerical_cols] = z_score_scaler.fit_transform(df[numerical_cols

print("\nDataFrame after Z-score Standardization:") print(df_z_score_scaled.head())
```

```
       8setosa
       9setosa  In [14]:
```

```
]
```

```
]
```

```
DataFrame after Min-Max Scaling:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
0.222222          0.625000          0.067797          0.041667
1          0.166667          0.416667          0.067797          0.041667
2          0.111111          0.500000          0.050847          0.041667   3
           0.083333          0.458333          0.084746          0.041667   4
           0.194444          0.666667          0.067797          0.041667

  species  0
setosa
1  setosa
2  setosa
```

```
3  setosa
4  setosa

DataFrame after Z-score Standardization:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
-0.900681          1.019004          -1.340227         -1.315444
1          -1.143017         -0.131979          -1.340227         -1.315444
2          -1.385353          0.328414          -1.397064         -1.315444
3          -1.506521          0.098217          -1.283389         -1.315444   4
           -1.021849          1.249201          -1.340227         -1.315444

  species  0
setosa  1
setosa
2  setosa
3  setosa
```

```
# Display the original DataFrame
print("Original DataFrame:") print(df.head())

# Create dummy variables for categorical columns df_dummies
= pd.get_dummies(df)

print("\nDataFrame with Dummy Variables:") print(df_dummies.head())
```

```
4  setosa
```

In [15]:

```
Original DataFrame:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
 1                4.9               3.0                1.4                0.2   2
         4.7               3.2                1.3               0.2   3
         4.6               3.1                1.5               0.2   4
           5.0               3.6                 1.4               0.2

  species  0
setosa
1  setosa
2  setosa  3  setosa
4  setosa

DataFrame with Dummy Variables:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
1                4.9               3.0                1.4                0.2
2                4.7               3.2                1.3                0.2   3
                 4.6               3.1                1.5                0.2   4
                 5.0               3.6                1.4                0.2

   species_setosa  species_versicolor  species_virginica  0
1                   0                   0
         11                  0                  0
         21                  0                  0
         31                  0                  0
```

41 0 0 In [16]:

```
import pandas as pd

# Group by 'species'
grouped = df.groupby('species')

# Calculate summary statistics print("Mean
of each group:") print(grouped.mean())

print("\nMedian of each group:") print(grouped.median())

print("\nCount of each group:") print(grouped.count())

print("\nStandard deviation of each group:") print(grouped.std())

print("\nMinimum value of each group:") print(grouped.min())

print("\nMaximum value of each group:") print(grouped.max())









print("\nSum of each group:") print(grouped.sum())
```

```
Mean of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \

species                                                              setosa
5.006             3.428              1.462   versicolor              5.936
2.770              4.260   virginica               6.588             2.974
5.552

            petal width (cm)  species
setosa                 0.246
versicolor             1.326
virginica              2.026

Median of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \ species
setosa                    5.0               3.4               1.50
versicolor                5.9               2.8               4.35
virginica                 6.5               3.0               5.55

            petal width (cm)  species
setosa                   0.2
versicolor               1.3
virginica                2.0

Count of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \ species
setosa                     50                50                 50
versicolor                 50                50                 50
virginica                  50                50                 50

            petal width (cm)  species
setosa                     50
versicolor                 50
virginica                  50

Standard deviation of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \ species
setosa               0.352490          0.379064           0.173664
versicolor           0.516171          0.313798           0.469911
virginica            0.635880          0.322497           0.551895

            petal width (cm)  species
setosa              0.105386
versicolor          0.197753
virginica           0.274650

Minimum value of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \ species
setosa                    4.3               2.3                1.0
versicolor                4.9               2.0                3.0
virginica                 4.9               2.2                4.5

            petal width (cm)  species
setosa                   0.1
versicolor               1.0
virginica                1.4
Maximum value of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \ species
setosa                    5.8               4.4                1.9

versicolor                7.0               3.4                5.1
virginica                 7.9               3.8                6.9

            petal width (cm)  species
setosa                   0.6
```

```
versicolor               1.8
virginica                2.5

Sum of each group:
            sepal length (cm)  sepal width (cm)  petal length (cm)  \ species
setosa                  250.3             171.4               73.1
versicolor              296.8             138.5              213.0
virginica               329.4             148.7              277.6

            petal width (cm)  species
setosa                  12.3
versicolor              66.3
virginica              101.3
```

In [17]:

```
# Creating  a pivot table to calculate the mean of each numerical column grouped by
pivot_mean = df.pivot_table(index='species', aggfunc='mean')

print("Pivot Table (mean of each numerical column by 'species'):") print(pivot_mean)

# Create a pivot table to calculate the sum of each numerical column grouped by 'sp
pivot_sum = df.pivot_table(index='species', aggfunc='sum')

print("\nPivot Table (sum of each numerical column by 'species'):") print(pivot_sum)

# Example with multiple aggregations: mean and count
pivot_multi = df.pivot_table(index='species', aggfunc={'sepal length (cm)': ['mean'

print("\nPivot Table (multiple aggregations) - Mean and Count of 'sepal length (cm)
print(pivot_multi)
```

```
Pivot Table (mean of each numerical column by 'species'):
            petal length (cm)  petal width (cm)  sepal length (cm)  \ species
setosa                  1.462             0.246              5.006
versicolor              4.260             1.326              5.936
virginica               5.552             2.026              6.588

            sepal width (cm)

species                       setosa
3.428  versicolor             2.770
virginica              2.974

Pivot Table (sum of each numerical column by 'species'):
            petal length (cm)  petal width (cm)  sepal length (cm)  \ species
setosa                   73.1              12.3              250.3
versicolor              213.0              66.3              296.8
virginica               277.6             101.3              329.4

            sepal width (cm)  species
setosa                 171.4

versicolor             138.5
virginica              148.7

Pivot Table (multiple aggregations) - Mean and Count of 'sepal length (cm)', Mean of
'sepal width (cm)':
           sepal length (cm)        sepal width (cm)
count    mean             mean species
setosa                     50  5.006            3.428
versicolor                 50  5.936            2.770
virginica                  50  6.588            2.974
```

In [20]:

```
# Filter for 'setosa' species
df_setosa = df[df['species'] == 'setosa']

# Example DataFrames for concatenation
df_setosa1 = df_setosa.iloc[:25]  # First 25 rows df_setosa2
= df_setosa.iloc[25:]  # Remaining rows

# Concatenate DataFrames along rows (default axis=0) df_concat_rows
= pd.concat([df_setosa1, df_setosa2]) print("Concatenated DataFrame
(along rows):") print(df_concat_rows.head())

# Concatenate DataFrames along columns (axis=1)
df_concat_cols = pd.concat([df_setosa1.reset_index(drop=True), df_setosa2.reset_ind
print("\nConcatenated DataFrame (along columns):") print(df_concat_cols.head())
```

```
Concatenated DataFrame (along rows):
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1                3.5               1.4                0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2   4
                   5.0               3.6                1.4               0.2

  species  0
setosa  1
setosa
2  setosa
3  setosa
4  setosa

Concatenated DataFrame (along columns):
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1                3.5                1.4                0.2   1                4.9
3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.6                3.1                1.5                0.2   4
                 5.0                3.6                1.4                0.2

  species  sepal length (cm)  sepal width (cm)  petal length (cm)  \ 0
setosa                5.0               3.0                1.6
1  setosa                5.0               3.4                1.6
2  setosa                5.2               3.5                1.5
3  setosa                5.2               3.4                1.4   4
   setosa                4.7               3.2                1.6

   petal width (cm) species  0
0.2  setosa
1               0.4  setosa
2               0.2  setosa  3               0.2
                setosa
4               0.2  setosa  In
```

[21]:

```
# Create an additional DataFrame with some extra information df_setosa_extra
= df_setosa.copy()
df_setosa_extra['extra_info'] = ['info' + str(i) for i in range(len(df_setosa_extra

# Merge DataFrames on the index
, df_merged = pd.merge(df_setosa, df_setosa_extra, left_index=True, right_index=True
print("\nMerged DataFrame (on index):") print(df_merged.head())
```

```
Merged DataFrame (on index):
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1                3.5                1.4                0.2
1                4.9                3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.6                3.1                1.5                0.2   4
                 5.0                3.6                1.4                0.2

  species  sepal length (cm)_extra  sepal width (cm)_extra  \ 0
setosa                      5.1                     3.5
1  setosa                      4.9                     3.0
2  setosa                      4.7                     3.2
3  setosa                      4.6                     3.1   4
   setosa                      5.0                     3.6

   petal length (cm)_extra  petal width (cm)_extra species_extra extra_info  0
1.4                    0.2        setosa      info0
 1                    1.4                     0.2        setosa      info1  2
             1.3                     0.2        setosa      info2
     3     1.5                    0.2        setosa      info3
     4     1.4                    0.2        setosa      info4  In [22]:
```

```
# Create another DataFrame to join df_setosa_additional
= pd.DataFrame({
    'additional_col': ['add' + str(i) for i in range(len(df_setosa))] },
index=df_setosa.index)

# Join DataFrames based on index
df_joined = df_setosa.join(df_setosa_additional) print("\nJoined
DataFrame (based on index):") print(df_joined.head())
```

```
Joined DataFrame (based on index):
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1                3.5                1.4                0.2   1                4.9
3.0                1.4                0.2
```

```
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2   4
                 5.0               3.6                1.4               0.2

  species additional_col  0
setosa           add0
1  setosa           add1  2
setosa           add2
     3     setosa           add3
     4     setosa           add4  In [24]:
```

```
# Filter for 'setosa' species
df_setosa = df[df['species'] == 'setosa']

# Create another DataFrame with some additional data df_extra
= pd.DataFrame({
    'species': ['setosa', 'versicolor', 'virginica'],
    'extra_info': ['info1', 'info2', 'info3']
}) # We will merge on the 'species' column, which is present in both

DataFrames

# Inner Join: Only rows with keys in both DataFrames are included
df_inner = pd.merge(df_setosa, df_extra, on='species', how='inner')
print("Inner Join:") print(df_inner.head())

# Left Join: All rows from the left DataFrame are included, with matching rows from


df_left = pd.merge(df_setosa, df_extra, on='species', how='left')
print("\nLeft Join:") print(df_left.head())

# Right Join: All rows from the right DataFrame are included, with matching rows fr
df_right = pd.merge(df_setosa, df_extra, on='species', how='right') print("\nRight
Join:") print(df_right.head())


# Outer Join: All rows from both DataFrames are included, with NaNs where there are
df_outer = pd.merge(df_setosa, df_extra, on='species', how='outer') print("\nOuter
Join:") print(df_outer.head())
```

```
Inner Join:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2   4
                 5.0               3.6                1.4               0.2

  species extra_info  0
setosa      info1  1
setosa      info1
2  setosa      info1
3  setosa      info1
4  setosa      info1

Left Join:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2   1                4.9
3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2   4
                 5.0               3.6                1.4               0.2

  species extra_info  0
setosa      info1
1  setosa      info1  2
setosa      info1
3  setosa      info1
4  setosa      info1

Right Join:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
 1                4.9               3.0                1.4               0.2   2
         4.7               3.2                1.3               0.2   3
         4.6               3.1                1.5               0.2   4
           5.0               3.6                1.4               0.2

  species extra_info  0
setosa      info1
1  setosa      info1
2  setosa      info1  3  setosa      info1
4  setosa      info1

Outer Join:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \ 0
5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2   3
                 4.6               3.1                1.5               0.2   4
                 5.0               3.6                1.4               0.2

  species extra_info  0
setosa      info1
         1 setosa      info1
         2 setosa      info1
         3 setosa      info1
         4 setosa      info1  In [ ]:
```