

```
import pandas as pd

# Load the Iris dataset
iris_df = pd.read_csv('https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv')
```

```
iris_df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5	3.5	1.4	0.2	setosa
1	4	3.0	1.4	0.2	setosa
2	4	3.2	1.3	0.2	setosa
3	4	3.1	1.5	0.2	setosa
4	5	3.6	1.4	0.2	setosa

Next steps:

[Generate code with iris_df](#)

[View recommended plots](#)

```
# Selecting specific columns
selected_columns = iris_df[['sepal_length', 'sepal_width']]
selected_columns
```

	sepal_length	sepal_width
0	5	3.5
1	4	3.0
2	4	3.2
3	4	3.1
4	5	3.6
...
145	6	3.0
146	6	2.5
147	6	3.0
148	6	3.4
149	5	3.0

150 rows × 2 columns

Next steps:

[Generate code with selected_columns](#)

[View recommended plots](#)

```
# Selecting rows based on condition
selected_rows = iris_df[iris_df['sepal_length'] > 5.0]
selected_rows
```




	sepal_length	sepal_width	petal_length	petal_width	species
50	7	3.2	4.7	1.4	versicolor
51	6	3.2	4.5	1.5	versicolor
52	6	3.1	4.9	1.5	versicolor
54	6	2.8	4.6	1.5	versicolor
56	6	3.3	4.7	1.6	versicolor
...
144	6	3.3	5.7	2.5	virginica
145	6	3.0	5.2	2.3	virginica
146	6	2.5	5.0	1.9	virginica
147	6	3.0	5.2	2.0	virginica
148	6	3.4	5.4	2.3	virginica

67 rows × 5 columns

Next steps:

[Generate code with selected_rows](#)[View recommended plots](#)

```
# Sorting data
sorted_data = iris_df.sort_values(by='sepal_length', ascending=False)
sorted_data
```




	sepal_length	sepal_width	petal_length	petal_width	species	
109	7	3.6	6.1	2.5	virginica	
135	7	3.0	6.1	2.3	virginica	
107	7	2.9	6.3	1.8	virginica	
105	7	3.0	6.6	2.1	virginica	
117	7	3.8	6.7	2.2	virginica	
...	
57	4	2.4	3.3	1.0	versicolor	
22	4	3.6	1.0	0.2	setosa	
29	4	3.2	1.6	0.2	setosa	
24	4	3.4	1.9	0.2	setosa	
41	4	2.3	1.3	0.3	setosa	

150 rows × 5 columns

Next steps:

[Generate code with sorted_data](#)[View recommended plots](#)

```
# c) Describing attributes of data, checking data types of each column
# Describe attributes
data_description = iris_df.describe()
data_description
```

	sepal_length	sepal_width	petal_length	petal_width	
count	150.000000	150.000000	150.000000	150.000000	
mean	5.386667	3.054000	3.758667	1.198667	
std	0.841752	0.433594	1.764420	0.763161	
min	4.000000	2.000000	1.000000	0.100000	
25%	5.000000	2.800000	1.600000	0.300000	
50%	5.000000	3.000000	4.350000	1.300000	
75%	6.000000	3.300000	5.100000	1.800000	
max	7.000000	4.400000	6.900000	2.500000	

Next steps:

[Generate code with data_description](#)[View recommended plots](#)

```
# Checking data types of each column
data_types = iris_df.dtypes
data_types
```

```
sepal_length    int64
sepal_width     float64
petal_length     float64
petal_width     float64
species         object
dtype: object
```

```
# d) Counting unique values of data, format of each column, converting variable data type
# Count unique values
unique_species_count = iris_df['species'].nunique()
unique_species_count
```

3

```
# Format of each column
column_formats = iris_df.dtypes
column_formats

sepal_length    int64
sepal_width     float64
petal_length    float64
petal_width     float64
species         object
dtype: object

# Converting variable data type (for example, converting 'sepal_length' to int)
iris_df['sepal_length'] = iris_df['sepal_length'].astype(int)
iris_df['sepal_length']

0      5
1      4
2      4
3      4
4      5
..
145    6
146    6
147    6
148    6
149    5
Name: sepal_length, Length: 150, dtype: int64
```

```
# e) Identifying missing values and filling in the missing values
# Identifying missing values
missing_values = iris_df.isnull().sum()
missing_values

sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Start coding or [generate](#) with AI.

