```
from google.colab import files
uploaded = files.upload()
```

Choose files Iris.csv

- **Iris.csv**(text/csv) - 5107 bytes, last modified: 24/06/2025 - 100% done

```
Saving Iris.csv to Iris.csv
```

```
import pandas as pd
df = pd.read_csv('Iris.csv')
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.info()       # See column types and missing values
df.describe()   # Summary statistics
df.columns      # Column names
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

```
df.isnull().sum()
```

	0
Id	0
SepalLengthCm	0
SepalWidthCm	0
PetalLengthCm	0
PetalWidthCm	0
Species	0

dtype: int64

```
# Example: Fill missing numeric values with mean
df.fillna(df.mean(numeric_only=True), inplace=True)
```

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Species'] = le.fit_transform(df['Species'])  # Example column
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

# Exclude target column from scaling
X = df.drop('Species', axis=1)
y = df['Species']

X_scaled = scaler.fit_transform(X)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
processed_df = pd.DataFrame(X_scaled, columns=X.columns)
processed_df['Species'] = y.values

processed_df.to_csv('processed_Iris.csv', index=False)
```

```
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('scaler', StandardScaler()),
    # You can add model here too
])
X_transformed = pipeline.fit_transform(X)
```

```
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	0
1	2	4.9	3.0	1.4	0.2	0
2	3	4.7	3.2	1.3	0.2	0
3	4	4.6	3.1	1.5	0.2	0
4	5	5.0	3.6	1.4	0.2	0

Next steps: Generate code with df View recommended plots New interactive sheet