```
import pandas as pd


df = pd.read_csv("/content/iris.csv")


df
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

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

```
from sklearn.preprocessing import StandardScaler

features = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']

# Separating out the features
x = df.loc[:, features].values

# Separating out the target
y = df.loc[:,['species']].values

# Standardizing the features
x = StandardScaler().fit_transform(x)


from sklearn.decomposition import PCA


pca = PCA(n_components=2)

principalComponents = pca.fit_transform(x)

principalDf = pd.DataFrame(data = principalComponents
             , columns = ["principal component 1", "principal component 2"])
finalDf = pd.concat([principalDf, df[['species']]], axis = 1)


finalDf
```

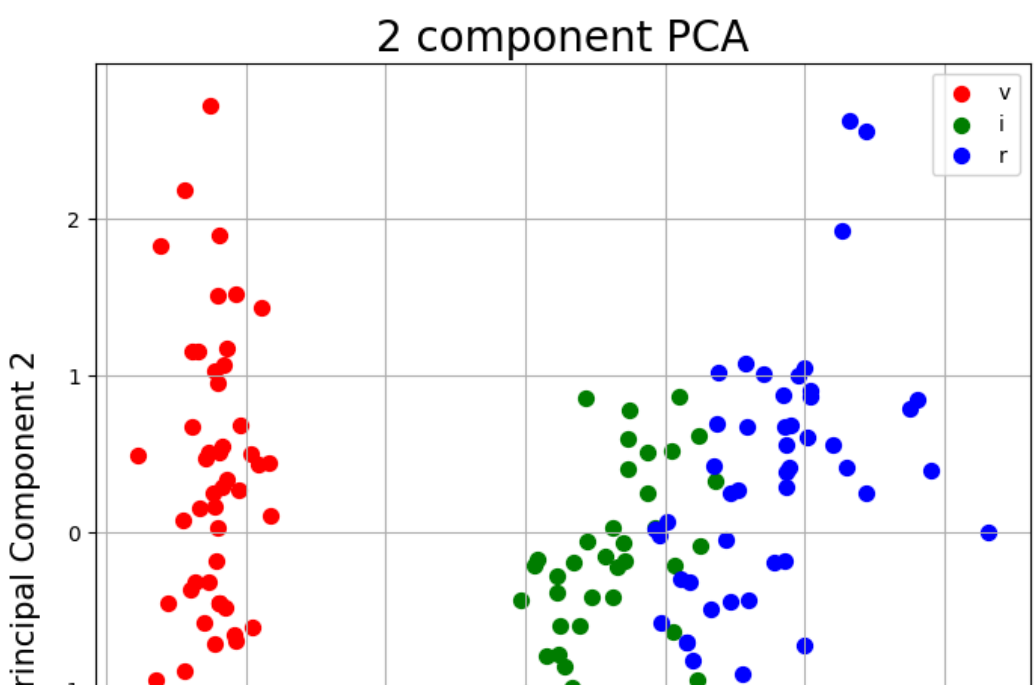
	principal component 1	principal component 2	species
0	-2.264542	0.505704	setosa
1	-2.086426	-0.655405	setosa
2	-2.367950	-0.318477	setosa
3	-2.304197	-0.575368	setosa
4	-2.388777	0.674767	setosa
...	...	...	...
145	1.870522	0.382822	virginica
146	1.558492	-0.905314	virginica
147	1.520845	0.266795	virginica
148	1.376391	1.016362	virginica
149	0.959299	-0.022284	virginica

150 rows × 3 columns

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

```
import matplotlib.pyplot as plt


fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('2 component PCA', fontsize = 20)
species = ['setosa', 'versicolor', 'virginica']
colors = ['r', 'g', 'b']
for species, color in zip(species,colors):
    indicesToKeep = finalDf['species'] == species
    ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1']
               , finalDf.loc[indicesToKeep, 'principal component 2']
               , c = color
               , s = 50)
ax.legend(species)
ax.grid()
```


2 component PCA
v
i
r
2
1
0
rincipal Component 2