

```
In [10]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import datasets
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

iris = datasets.load_iris()

X = iris.data
y = iris.target
```

C:\Users\Youssef\Anaconda3\lib\site-packages\statsmodels\tools_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
import pandas.util.testing as tm

```
In [11]: target_names = iris.target_names
```

```
In [12]: # we can use import pca but we took the pca code in linear Algebra
def mean(x):
    return float(np.sum(x) / len(x))
```

```
In [13]: def std(x):
    m = mean(x)
    l = len(x)
    var = sum([(x - m) ** 2 for x in x]) / (l - 1)
    std = var ** 0.5
    return std
```

```
In [14]: def standardize(M):
    New_M = []
    for i in range(len(M[0])):
        m = mean(M[:, i])
        stddev = std(M[:, i])
        row = []
        for j in range(len(M[0])):
            row.append((M[j, i] - m) / stddev)
        New_M.append(row)
    return np.transpose(New_M)
```

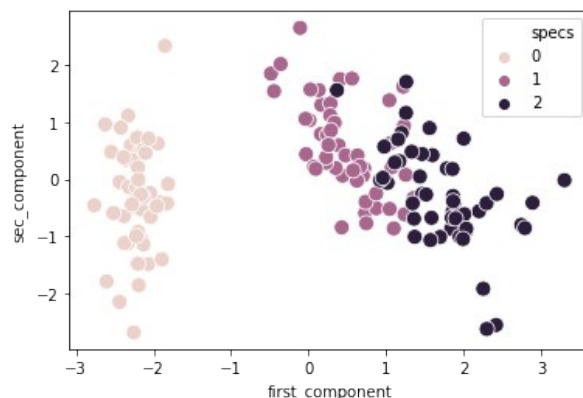
```
In [15]: standardized_col = standardize(X)
```

```
In [16]: C = np.cov(standardized_col.T)
```

```
In [17]: values, vectors = np.linalg.eig(C)
```

```
In [18]: first_comp = standardized_col.dot(vectors.T[0])
sec_comp = standardized_col.dot(vectors.T[1])
df = pd.DataFrame(first_comp, columns=['first_component'])
df['sec_component'] = sec_comp
df['specs'] = y
```

```
In [19]: sns.scatterplot(df["first_component"], df["sec_component"], hue=df['specs'], s=100)
plt.show();
```



```
In [ ]:
```

```
In [ ]:
```

EVO PDF Tools Demo