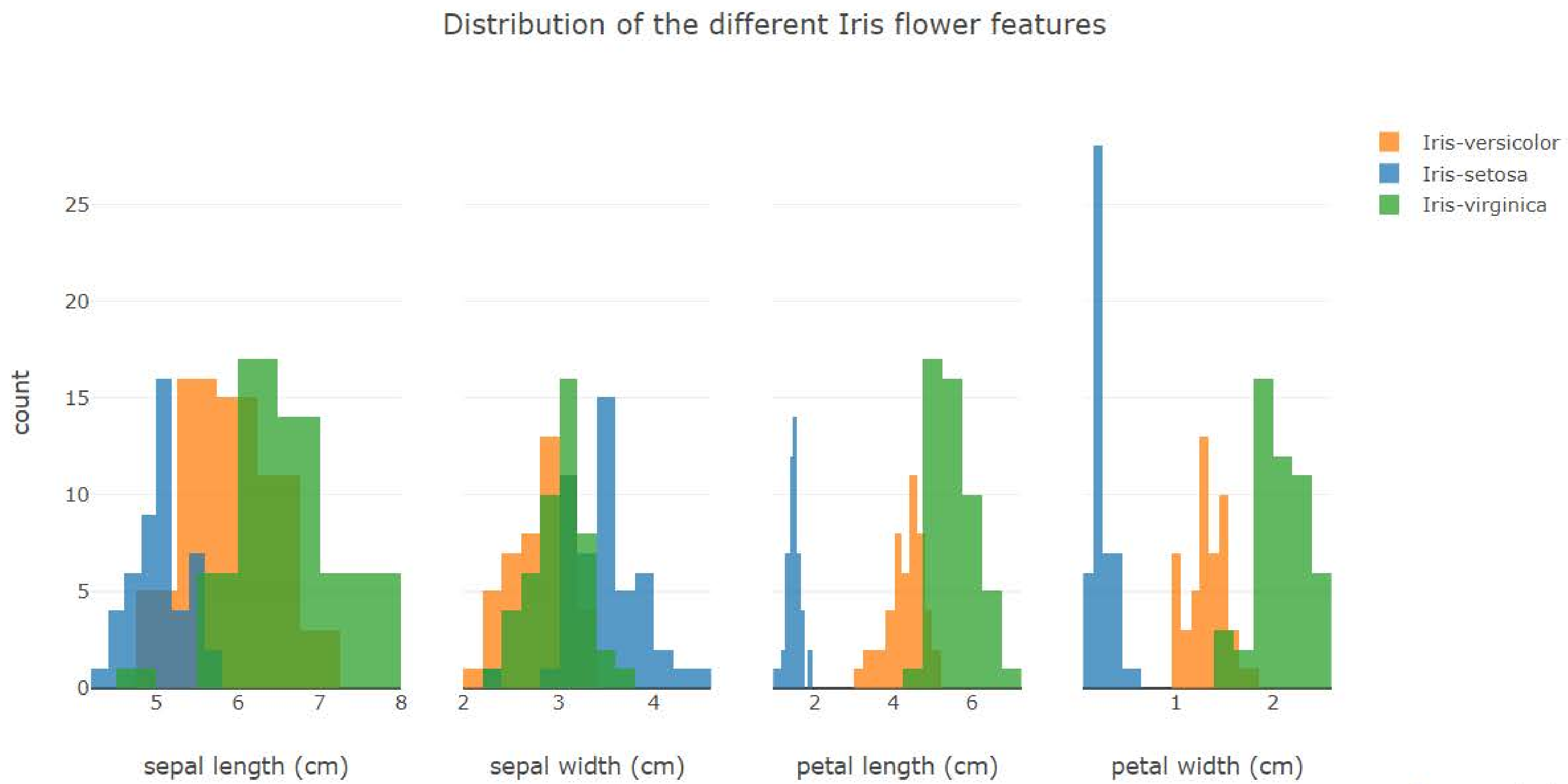
```
fig = go.Figure(data=data, layout=layout)
iplot(fig)
```



Export to plot.ly »

Standardize (scale, transform) the data

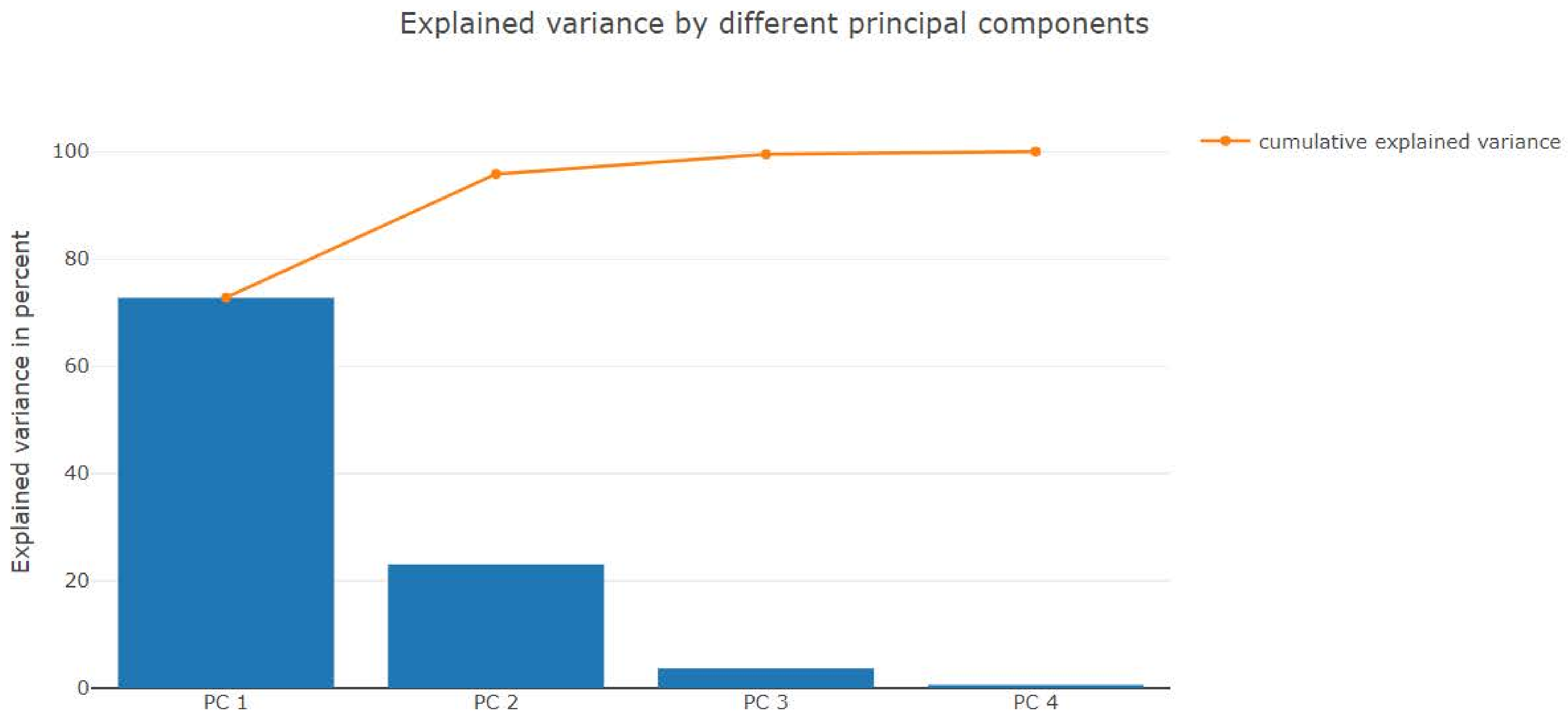
In [47]:

```
from sklearn.preprocessing import StandardScaler
X_std = StandardScaler().fit_transform(X)
```

Perform eigen-decomposition on the covariance matrix

```
        title='Explained variance by different principal components')

fig = Figure(data=data, layout=layout)
iplot(fig)
```



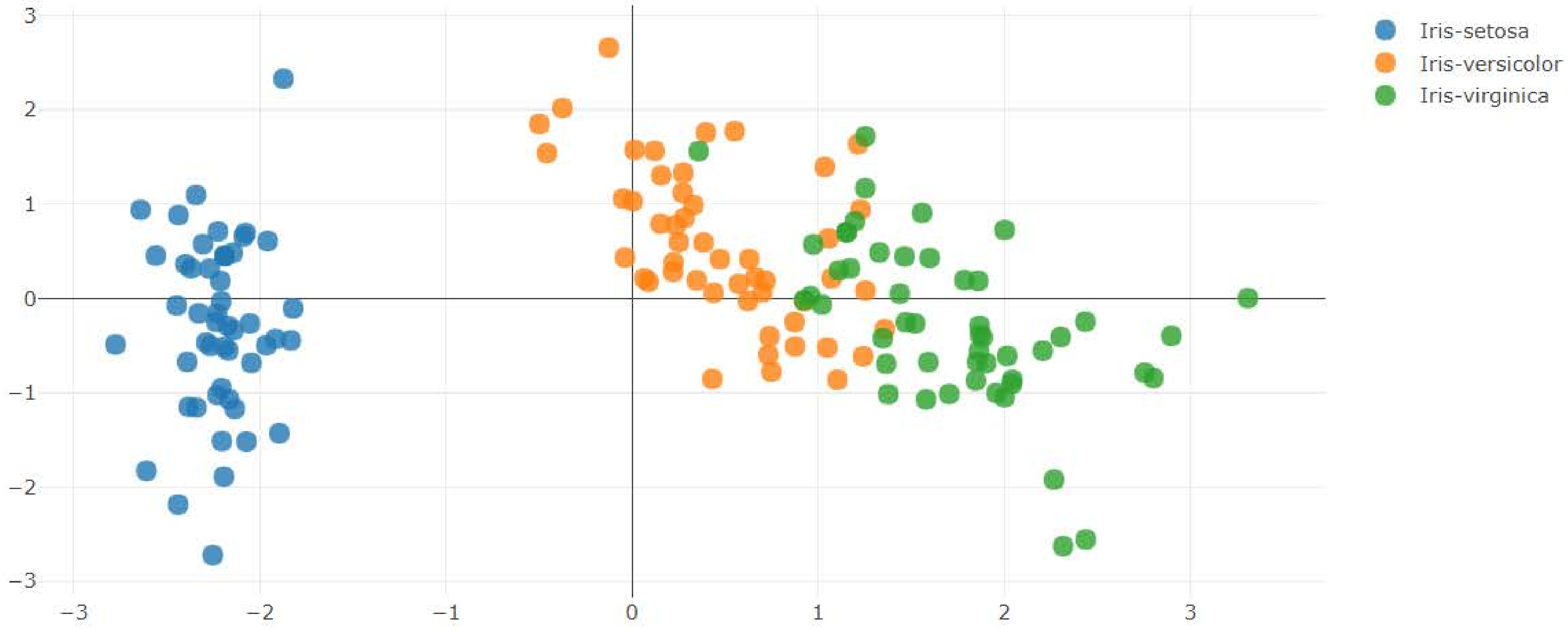
Export to plot.ly »

Now reduce the 4D feature space to 2D feature space by selecting the top 2 eigenvalues and corresponding eigenvectors

In [57]:
```
matrix_w = np.hstack((eig_pairs[0][1].reshape(4,1),
                      eig_pairs[1][1].reshape(4,1)))

print('Matrix W:\n', matrix_w)
```

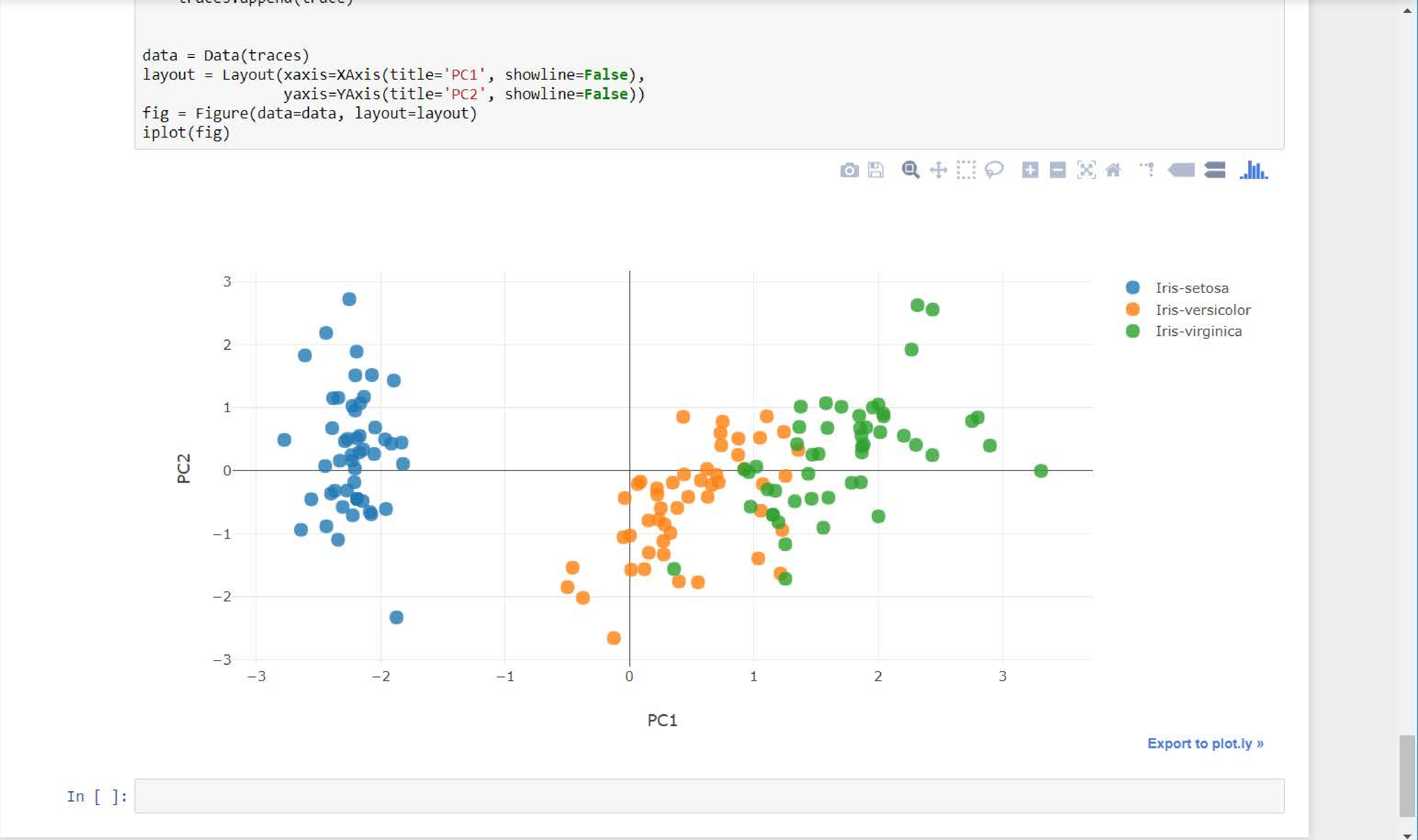
```
fig = Figure(data=data, layout=layout)
iplot(fig)
```



Export to plot.ly »

As an alternative method use sklearn

```
In [60]: from sklearn.decomposition import PCA as sklearnPCA
         sklearn_pca = sklearnPCA(n_components=2)
         Y_sklearn = sklearn_pca.fit_transform(X_std)
```



```
    traces.append(trace)


data = Data(traces)
layout = Layout(xaxis=XAxis(title='PC1', showline=False),
                yaxis=YAxis(title='PC2', showline=False))
fig = Figure(data=data, layout=layout)
iplot(fig)
```