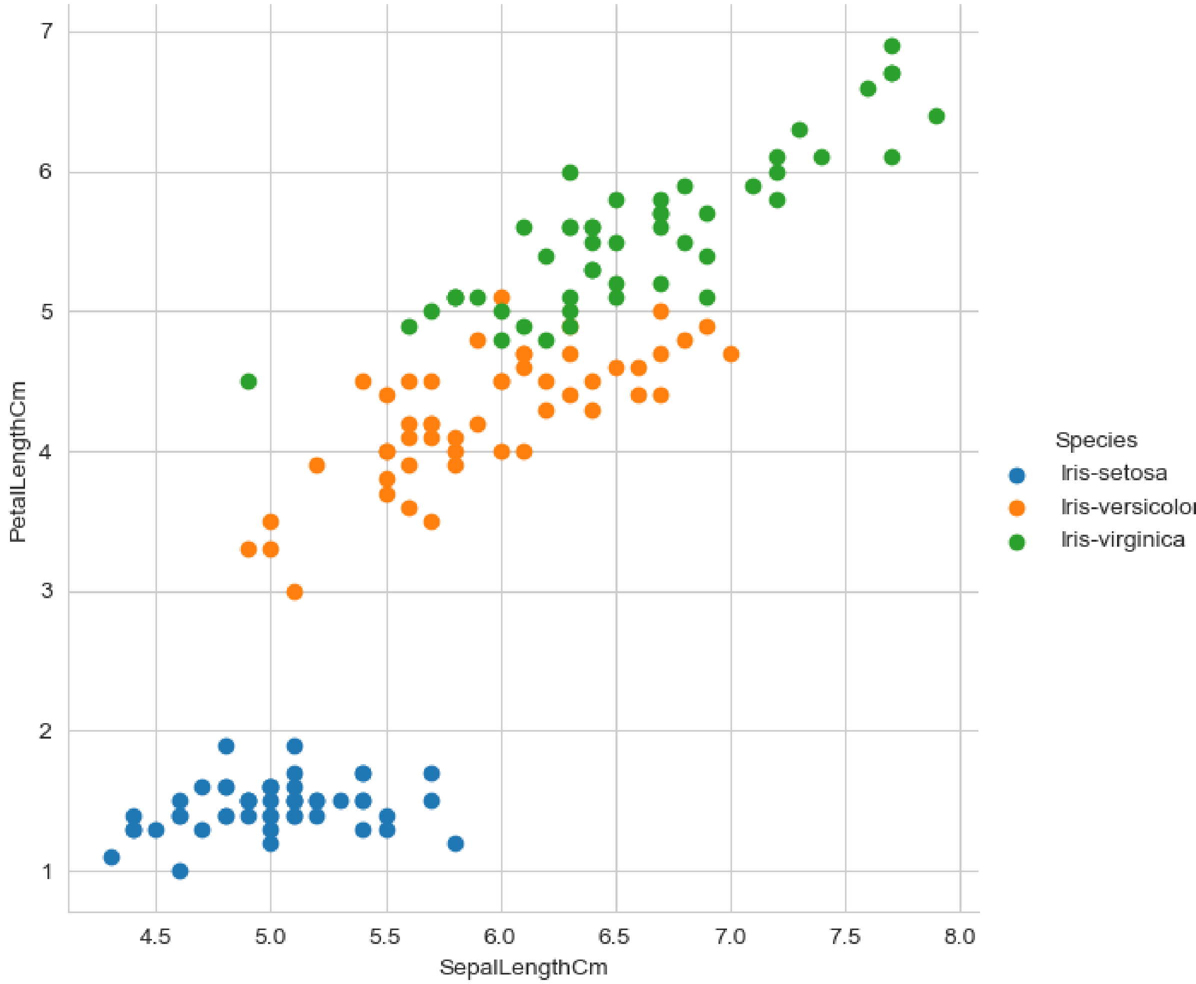
ML LAB 10

Implement Principal Component Analysis for dimensionality reduction in a given business environment and comment on its efficiency and performance.

In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
iris = pd.read_csv("Iris.csv")
df=pd.DataFrame(iris)
df.head()
x=df.drop(['Id','Species'],axis=1)
print(x.head())
y=df.Species
print(y.head())
```

```
   SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
0            5.1           3.5            1.4           0.2
1            4.9           3.0            1.4           0.2
2            4.7           3.2            1.3           0.2
3            4.6           3.1            1.5           0.2
4            5.0           3.6            1.4           0.2
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

In [17]:
```
import seaborn as sns
sns.set_style("whitegrid")
sns.FacetGrid(iris,hue='Species',height=6).map(plt.scatter,'SepalLengthCm','Petal
plt.show()
```



In [35]:
```
from sklearn.preprocessing import StandardScaler
X = StandardScaler().fit_transform(x)
print(X[:5])
type(X)
print(X.shape[0])
```

```
[[-0.90068117  1.03205722 -1.3412724  -1.31297673]
 [-1.14301691 -0.1249576  -1.3412724  -1.31297673]
 [-1.38535265  0.33784833 -1.39813811 -1.31297673]
 [-1.50652052  0.10644536 -1.2844067  -1.31297673]
 [-1.02184904  1.26346019 -1.3412724  -1.31297673]]
150
```

In [36]:
```
X_mean = np.mean(X, axis=0)
print(X_mean)
# cov_mat = np.cov(X)
cov_mat = (X - X_mean).T.dot((X - X_mean)) / (X.shape[0])
print('Covariance matrix \n%s' %cov_mat)
```

```
[-4.73695157e-16 -6.63173220e-16  3.31586610e-16 -2.84217094e-16]
Covariance matrix
[[ 1.         -0.10936925  0.87175416  0.81795363]
 [-0.10936925  1.         -0.4205161  -0.35654409]
 [ 0.87175416 -0.4205161   1.          0.9627571 ]
 [ 0.81795363 -0.35654409  0.9627571   1.        ]]
```

In [37]:
```
eig_vals, eig_vecs = np.linalg.eig(cov_mat)
print('Eigenvectors \n%s' %eig_vecs)
print('\nEigenvalues \n%s' %eig_vals)
```

```
Eigenvectors
[[ 0.52237162 -0.37231836 -0.72101681  0.26199559]
 [-0.26335492 -0.92555649  0.24203288 -0.12413481]
 [ 0.58125401 -0.02109478  0.14089226 -0.80115427]
 [ 0.56561105 -0.06541577  0.6338014   0.52354627]]

Eigenvalues
[2.91081808 0.92122093 0.14735328 0.02060771]
```

In [43]:
```
pc1=X.dot(eig_vecs.T[0])
pc2=X.dot(eig_vecs.T[1])
result = pd.DataFrame(pc1,columns=['PC1'])
result['PC2']=pc2
result['species']=y
result.head()
```

Out[43]:

	PC1	PC2	species
0	-2.264542	-0.505704	Iris-setosa
1	-2.086426	0.655405	Iris-setosa
2	-2.367950	0.318477	Iris-setosa
3	-2.304197	0.575368	Iris-setosa
4	-2.388777	-0.674767	Iris-setosa

In [64]:
```
plt.figure(figsize=(30,10))
sns.FacetGrid(result,hue='species',height=6).map(plt.scatter,'PC1','PC2').add_leg
plt.show()
```

<Figure size 3000x1000 with 0 Axes>