Assignment 4: K-Means Clustering

Loading Dataset (IRIS.csv)

In [109]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('IRIS.csv')
df.head()
```

Out[109]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [110]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepal_length    150 non-null float64
sepal_width     150 non-null float64
petal_length    150 non-null float64
petal_width     150 non-null float64
species         150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

Changing species datatype to float

In [111]:
```
df['species'] = df['species'].replace({'Iris-setosa': 0.0, 'Iris-versicolor': 1.0, 'Iris-virginica': 2.0})
df.head()
```

Out[111]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

In [112]:
```
df.corr()
```

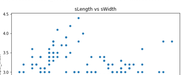
Out[112]:

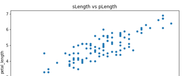
	sepal_length	sepal_width	petal_length	petal_width	species
sepal_length	1.000000	-0.109369	0.871754	0.817954	0.782561
sepal_width	-0.109369	1.000000	-0.420516	-0.356544	-0.419446
petal_length	0.871754	-0.420516	1.000000	0.962757	0.949043
petal_width	0.817954	-0.356544	0.962757	1.000000	0.956464
species	0.782561	-0.419446	0.949043	0.956464	1.000000

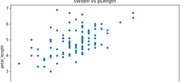
Plotting scatter plots using different dimensions

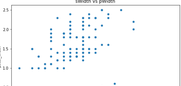
In [113]:
```
df.plot.scatter(0,1,figsize=(8,5),title="sLength vs sWidth")
df.plot.scatter(0,2,figsize=(8,5),title="sLength vs pLength")
df.plot.scatter(0,3,figsize=(8,5),title="sLength vs pWidth")
df.plot.scatter(0,4,figsize=(8,5),title="sLength vs species")
df.plot.scatter(1,2,figsize=(8,5),title="sWidth vs pLength")
df.plot.scatter(1,3,figsize=(8,5),title="sWidth vs pWidth")
df.plot.scatter(1,4,figsize=(8,5),title="sWidth vs species")
df.plot.scatter(2,3,figsize=(8,5),title="pLength vs pWidth")
df.plot.scatter(2,4,figsize=(8,5),title="pLength vs species")
df.plot.scatter(3,4,figsize=(8,5),title="pWidth vs species")
```

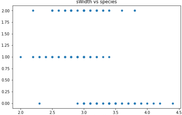
Out[113]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1da5a05e8048>
```

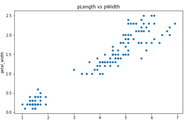


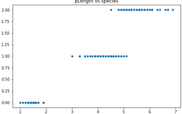


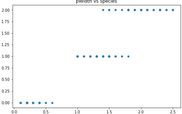












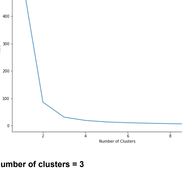
Selecting preferred dimensions

Petal Length vs Petal Width

In [114]:
```
X = df.iloc[:, [2, 3]].values
```

Finding ideal number of clusters using elbow method

In [115]:
```
from sklearn.cluster import KMeans
fig = plt.figure(figsize=(10, 8))
WCSS = []
for i in range(1, 11):
    clf = KMeans(n_clusters=i, init="k-means++", max_iter=300, n_init=10, random_state=0)
    clf.fit(X)
    WCSS.append(clf.inertia_)
plt.plot(range(1, 11), WCSS)
plt.title('Elbow Method')
plt.ylabel("WCSS")
plt.xlabel('Number of Clusters')
plt.show()
```



Number of clusters = 3

In [116]:
```
clf = KMeans(n_clusters=3, init="k-means++", max_iter=300, n_init=10, random_state=0)
y_kmeans = clf.fit_predict(X)
```

In [117]:
```
fig = plt.figure(figsize=(10, 8))
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], color='blue', s=60, label='Cluster 1', edgecolors='black')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], color='yellow', s=60, label='Cluster 2', edgecolors='black')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], color='red', s=60, label='Cluster 3', edgecolors='black')
plt.scatter(clf.cluster_centers_[:, 0], clf.cluster_centers_[:, 1], color='magenta', s=100, label='Centroid',edgecolors='black')
plt.legend()
plt.title('Clusters using KMeans')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.show()
```