In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import seaborn as sns
```

In [3]:
```
data = pd.read_csv('4_Iris.csv')
```

In [5]:
```
data
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Speci
0	1	5.1	3.5	1.4	0.2	Ir seto
1	2	4.9	3.0	1.4	0.2	Ir seto
2	3	4.7	3.2	1.3	0.2	Ir seto
3	4	4.6	3.1	1.5	0.2	Ir seto
4	5	5.0	3.6	1.4	0.2	Ir seto
...	...	...	...	...	...	
145	146	6.7	3.0	5.2	2.3	Ir virgini
146	147	6.3	2.5	5.0	1.9	Ir virgini
147	148	6.5	3.0	5.2	2.0	Ir virgini
148	149	6.2	3.4	5.4	2.3	Ir virgini
149	150	5.9	3.0	5.1	1.8	Ir virgini

150 rows × 6 columns

In [6]:
```
data.head()
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [7]:
```
data.tail()
```

Out[7]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Speci
145	146	6.7	3.0	5.2	2.3	Ir virgini
146	147	6.3	2.5	5.0	1.9	Ir virgini
147	148	6.5	3.0	5.2	2.0	Ir virgini
148	149	6.2	3.4	5.4	2.3	Ir virgini
149	150	5.9	3.0	5.1	1.8	Ir virgini

In [8]:
```
len(data)
```

Out[8]:
```
150
```

In [9]:
```
data.shape
```

Out[9]:
```
(150, 6)
```

In [10]:
```
data.columns
```

Out[10]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthC
m',
       'Species'],
      dtype='object')
```

In [13]:
```
for i, col in enumerate(data.columns):
    print(f"Column number {i+1} is {col}")
```

```
Column number 1 is Id
Column number 2 is SepalLengthCm
Column number 3 is SepalWidthCm
Column number 4 is PetalLengthCm
Column number 5 is PetalWidthCm
Column number 6 is Species
```

In [14]:
```
data.dtypes
```

Out[14]:
```
Id                int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [15]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [16]:
```
data.describe()
```

Out[16]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthC
count	150.000000	150.000000	150.000000	150.000000	150.0000
mean	75.500000	5.843333	3.054000	3.758667	1.1986
std	43.445368	0.828066	0.433594	1.764420	0.7631
min	1.000000	4.300000	2.000000	1.000000	0.1000
25%	38.250000	5.100000	2.800000	1.600000	0.3000
50%	75.500000	5.800000	3.000000	4.350000	1.3000
75%	112.750000	6.400000	3.300000	5.100000	1.8000
max	150.000000	7.900000	4.400000	6.900000	2.5000

In [17]:
```
data.isnull()
```

Out[17]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Spec
0	False	False	False	False	False	F
1	False	False	False	False	False	F
2	False	False	False	False	False	F
3	False	False	False	False	False	F
4	False	False	False	False	False	F
...	...	...	...	...	...	
145	False	False	False	False	False	F
146	False	False	False	False	False	F
147	False	False	False	False	False	F
148	False	False	False	False	False	F
149	False	False	False	False	False	F

150 rows × 6 columns

In [18]:
```
data.isnull().sum()
```

Out[18]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [19]:
```
data.drop('Id', axis=1, inplace=True)
data.head()
```

Out[19]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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 [20]:
```
data.isna().sum()
```

Out[20]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [21]:
```
data.head()
```

Out[21]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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 [22]:
```
data['Species'].value_counts()
```

Out[22]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [ ]:
```
# Target data
target_data = data.iloc[:,4]
# Variable_name = datafram.iloc[select rows, select column]
target_data.head()
```

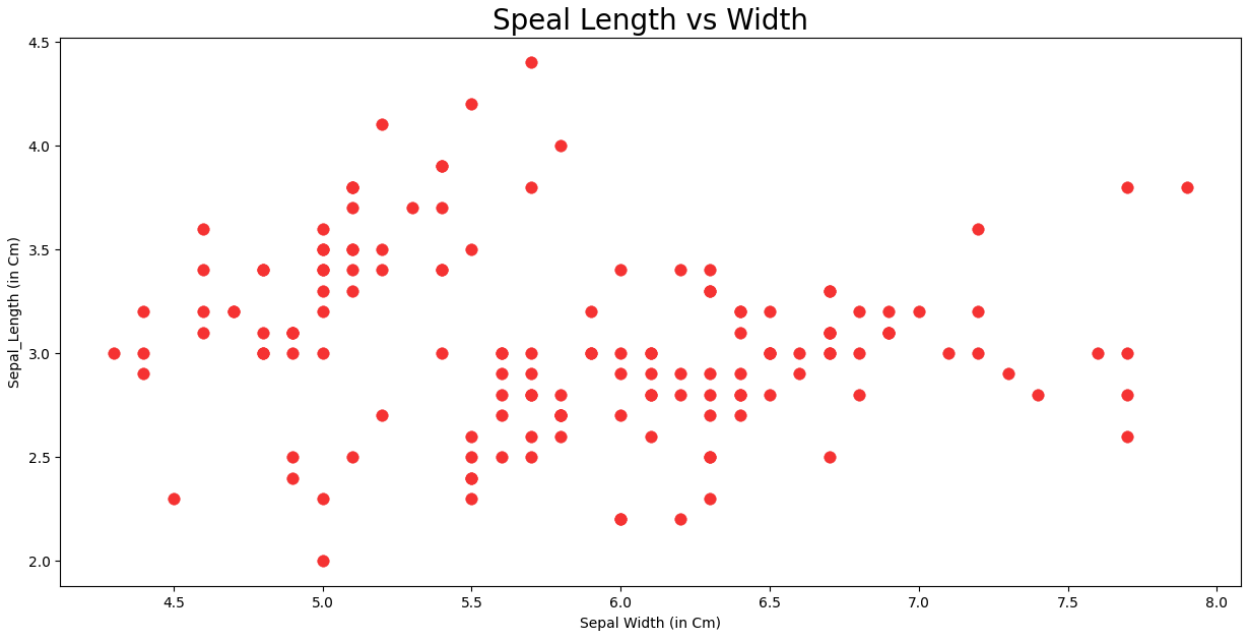
Out[ ]:
```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

In [26]:
```
# Clustering data
clustering_data = data.iloc[:,[0,1,2,3]]
clustering_data.head()
```

Out[26]:

	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

In [27]:
```
fig, ax = plt.subplots(figsize=(15,7))
ax = sns.scatterplot(x=data['SepalLengthCm'], y=data['SepalWidthCm'], s=70, cc
ax.set_xlabel('Sepal Width (in Cm)')
ax.set_ylabel('Sepal_Length (in Cm)')
plt.title('Speal Length vs Width', fontsize=20)
plt.show()
```

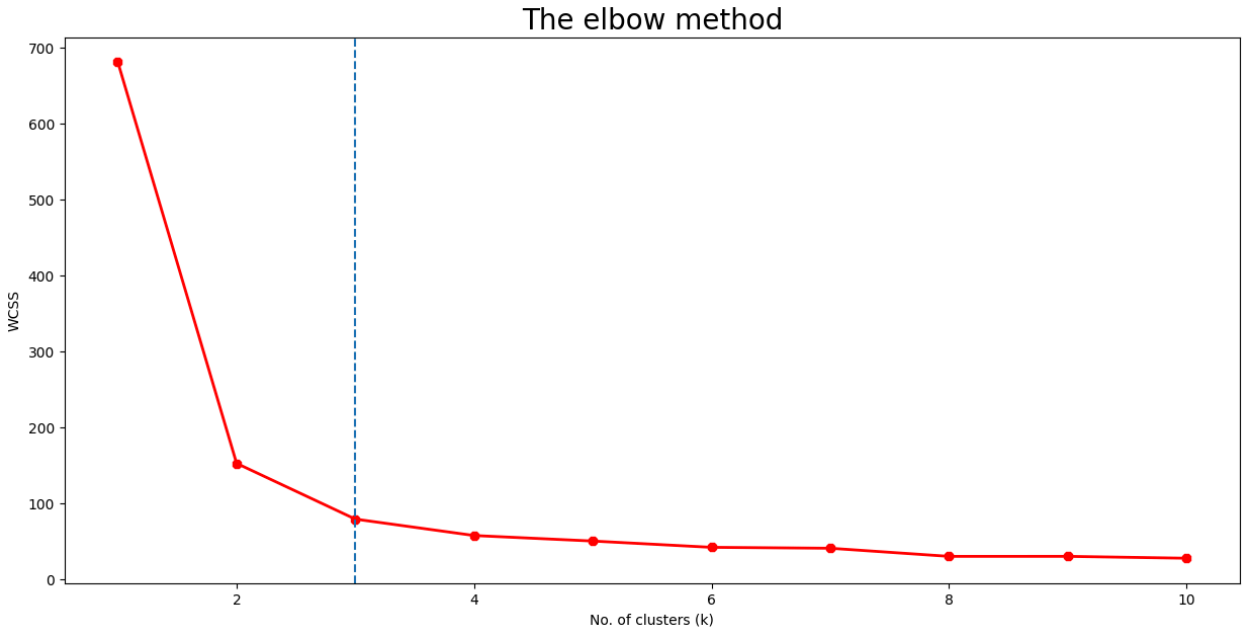


In [28]:
```
#Elbow method
wcss = []
for i in range(1, 11):
    km = KMeans(i)
    km.fit(clustering_data)
    wcss.append(km.inertia_)
np.array(wcss)
```

Out[28]:
```
array([680.8244    , 152.36870648,  78.94506583,  57.34492381,
        50.06165816,  41.79382447,  40.58359608,  29.88769645,
        29.90719002,  27.42902002])
```

In [32]:
```
# Elbow method visualization
fig, ax = plt.subplots(figsize=(15,7))
ax = plt.plot(range(1, 11), wcss, linewidth=2, color='red', marker='8')
```

```
plt.axvline(x=3, ls='--')
plt.ylabel("WCSS")
plt.xlabel("No. of clusters (k)")
plt.title("The elbow method", fontsize=20)
plt.show()
```



In [35]:
```
# clustering
kms = KMeans(n_clusters=3, init = 'k-means++')
kms.fit(clustering_data)
```

Out[35]:

KMeans

Parameters

n_clusters	3
init	'k-means++'
n_init	'auto'
max_iter	300
tol	0.0001
verbose	0
random_state	None
copy_x	True
algorithm	'lloyd'

In [36]:
```
clusters = clustering_data.copy()
clusters['Cluster_Prediction'] = kms.fit_predict(clustering_data)
```

```
clusters.head()
```

Out[36]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Cluster_Predi
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	

In [37]:
```
kms.cluster_centers_
```

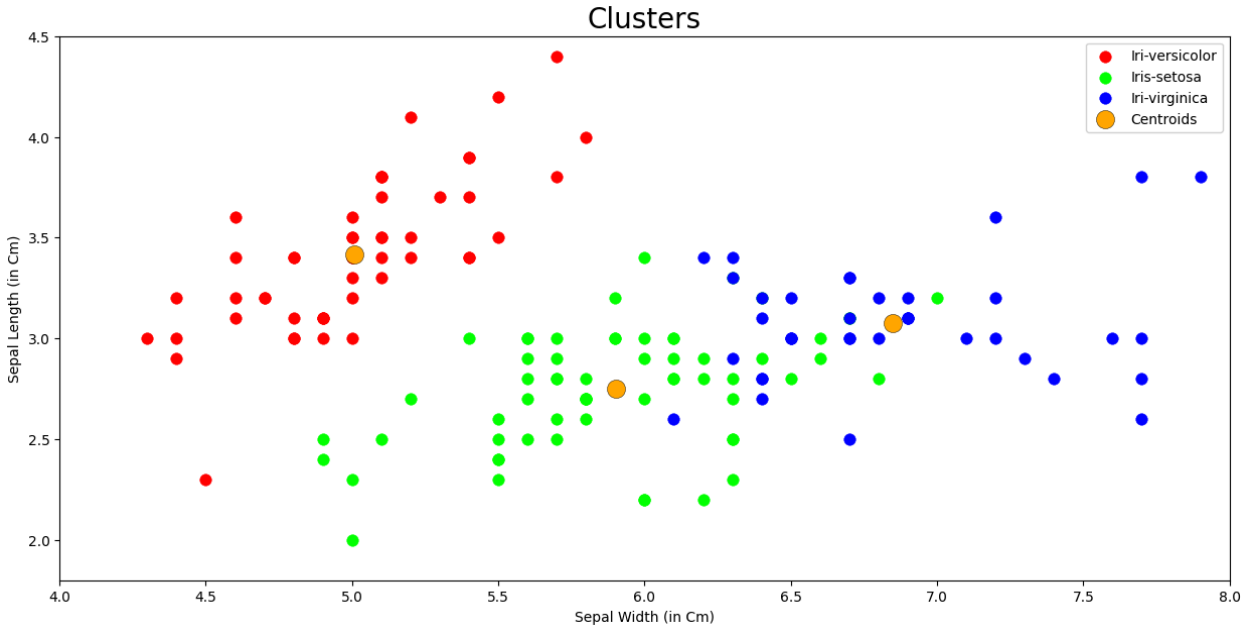
Out[37]:
```
array([[5.006     , 3.418     , 1.464     , 0.244     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

In [48]:
```
fig, ax = plt.subplots(figsize=(15, 7))
plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 0]['SepalLengthCm'],
            y=clusters[clusters['Cluster_Prediction'] == 0]['SepalWidthCm'],
            s=70, edgecolor='red', linewidth=0.3, c='red', label='Iri-versicol

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 1]['SepalLengthCm'],
            y=clusters[clusters['Cluster_Prediction'] == 1]['SepalWidthCm'],
            s=70, edgecolor='lime', linewidth=0.3, c='lime', label='Iris-setos

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 2]['SepalLengthCm'],
            y=clusters[clusters['Cluster_Prediction'] == 2]['SepalWidthCm'],
            s=70, edgecolor='blue', linewidth=0.3, c='blue', label='Iri-virgir

plt.scatter(x=kms.cluster_centers_[:,0], y=kms.cluster_centers_[:, 1], s=170,
plt.legend(loc='upper right')
plt.xlim(4, 8)
plt.ylim(1.8, 4.5)
ax.set_xlabel('Sepal Width (in Cm)')
ax.set_ylabel('Sepal Length (in Cm)')
plt.title("Clusters", fontsize=20)
plt.show()
```


Clusters
Iri-versicolor
Iris-setosa
Iri-virginica
Centroids
Sepal Length (in Cm)
Sepal Width (in Cm)
4.5
4.0
3.5
3.0
2.5
2.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

In [ ]: