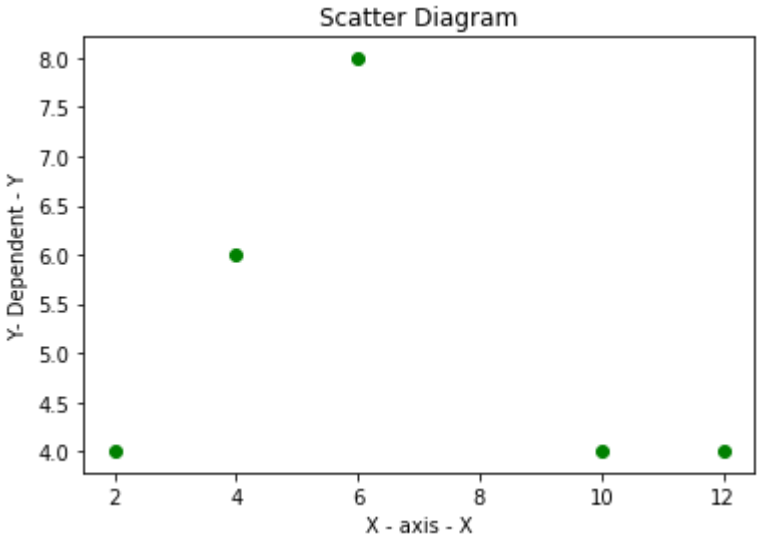
In [1]:
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = [[2,4],[4,6],[6,8],[10,4],[12,4]]
df = pd.DataFrame(data, columns = ['X','Y'])
```

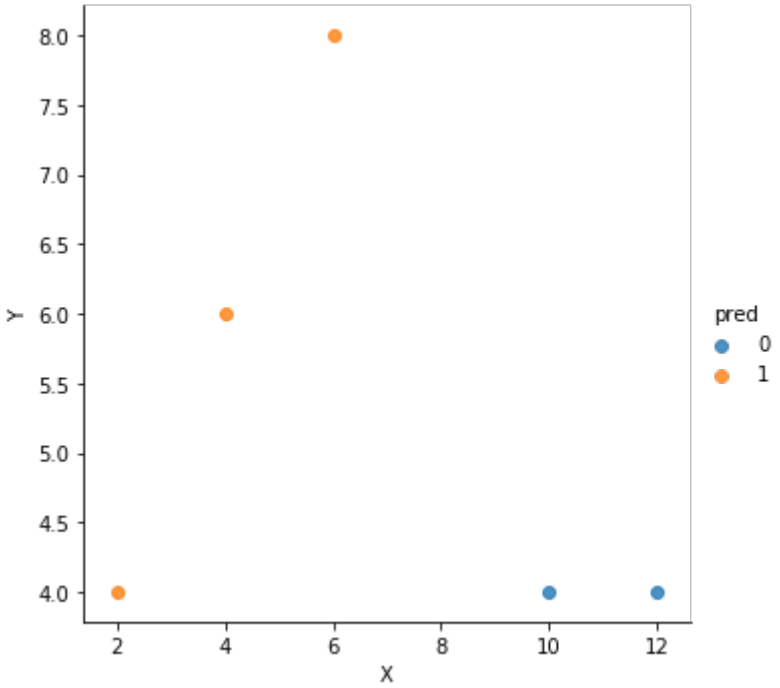
In [3]:
```
plt.scatter(df['X'],df['Y'],color='green')
plt.title('Scatter Diagram')
plt.xlabel('X - axis - X')
plt.ylabel('Y- Dependent - Y')
kmeans = KMeans(n_clusters = 2)
kmeans.fit(df)
```

Out[3]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=3
00,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances=
'auto',
       random_state=None, tol=0.0001, verbose=0)
```



In [4]:
```
df['pred'] = kmeans.predict(df)
sns.lmplot('X','Y',scatter=True,fit_reg=False,data=df,hue='pred')
labels = kmeans.labels_
score=metrics.silhouette_score(df, labels, metric='euclidean')
# Print Silhouette score
print('\nSilhouette score is:',score)
```

```
Silhouette score is: 0.5929095202298239
```



In [5]:
```
import pandas as pd
from sklearn.cluster import KMeans
import seaborn as sns
```

In [6]:
```
df = pd.read_csv("/home/machine/Downloads/iris.csv")
```

In [7]:
```
print(df.head(10))
```

```
    SL   SW   PL   PW      SPECIES
0  4.9  3.0  1.4  0.2  Iris-setosa
1  4.7  3.2  1.3  0.2  Iris-setosa
2  4.6  3.1  1.5  0.2  Iris-setosa
3  5.0  3.6  1.4  0.2  Iris-setosa
4  5.4  3.9  1.7  0.4  Iris-setosa
5  4.6  3.4  1.4  0.3  Iris-setosa
6  5.0  3.4  1.5  0.2  Iris-setosa
7  4.4  2.9  1.4  0.2  Iris-setosa
8  4.9  3.1  1.5  0.1  Iris-setosa
9  5.4  3.7  1.5  0.2  Iris-setosa
```

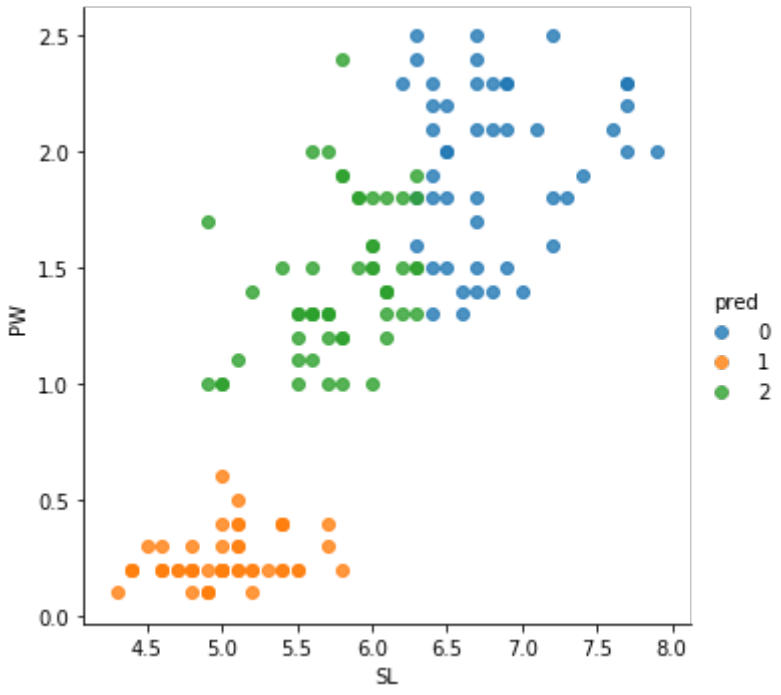
In [8]:
```
kmeans = KMeans(n_clusters = 3)
X=df.values[:,0:2]
kmeans.fit(X)
df['pred'] = kmeans.predict(X)
df.head(10)
```

Out[8]:

	SL	SW	PL	PW	SPECIES	pred
0	4.9	3.0	1.4	0.2	Iris-setosa	1
1	4.7	3.2	1.3	0.2	Iris-setosa	1
2	4.6	3.1	1.5	0.2	Iris-setosa	1
3	5.0	3.6	1.4	0.2	Iris-setosa	1
4	5.4	3.9	1.7	0.4	Iris-setosa	1
5	4.6	3.4	1.4	0.3	Iris-setosa	1
6	5.0	3.4	1.5	0.2	Iris-setosa	1
7	4.4	2.9	1.4	0.2	Iris-setosa	1
8	4.9	3.1	1.5	0.1	Iris-setosa	1
9	5.4	3.7	1.5	0.2	Iris-setosa	1

In [9]:
```
sns.lmplot('SL','PW',scatter=True,fit_reg=False,data=df,hue='pred')
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x7f4e0c4583d0>
```



In [10]:
```
df = pd.read_csv("/home/machine/Downloads/buddymove_holidayiq.csv")
```

In [11]:
```
print(df.head(10))
```

```
   User Id  Sports  Religious  Nature  Theatre  Shopping  Picnic
0   User 1       2         77      79       69        68      95
1   User 2       2         62      76       76        69      68
2   User 3       2         50      97       87        50      75
3   User 4       2         68      77       95        76      61
4   User 5       2         98      54       59        95      86
5   User 6       3         52     109       93        52      76
6   User 7       3         64      85       82        73      69
7   User 8       3         54     107       92        54      76
8   User 9       3         64     108       64        54      93
9  User 10       3         86      76       74        74     103
```

In [12]:
```
df = df.drop(['User Id'],1)
```

In [13]:
```
df
```

Out[13]:

	Sports	Religious	Nature	Theatre	Shopping	Picnic
0	2	77	79	69	68	95
1	2	62	76	76	69	68
2	2	50	97	87	50	75
3	2	68	77	95	76	61
4	2	98	54	59	95	86
...	...	...	...	...	...	...
244	18	139	148	129	129	168
245	22	114	228	104	84	168
246	20	124	178	104	158	174
247	20	133	149	139	144	213
248	20	143	149	139	159	143

249 rows × 6 columns

In [14]:
```
kmeans = KMeans(n_clusters = 3)
```

In [15]:
```
X=df.values[:,0:2]
```

In [16]:
```
kmeans.fit(X)
```

Out[16]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=3
00,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances=
'auto',
       random_state=None, tol=0.0001, verbose=0)
```

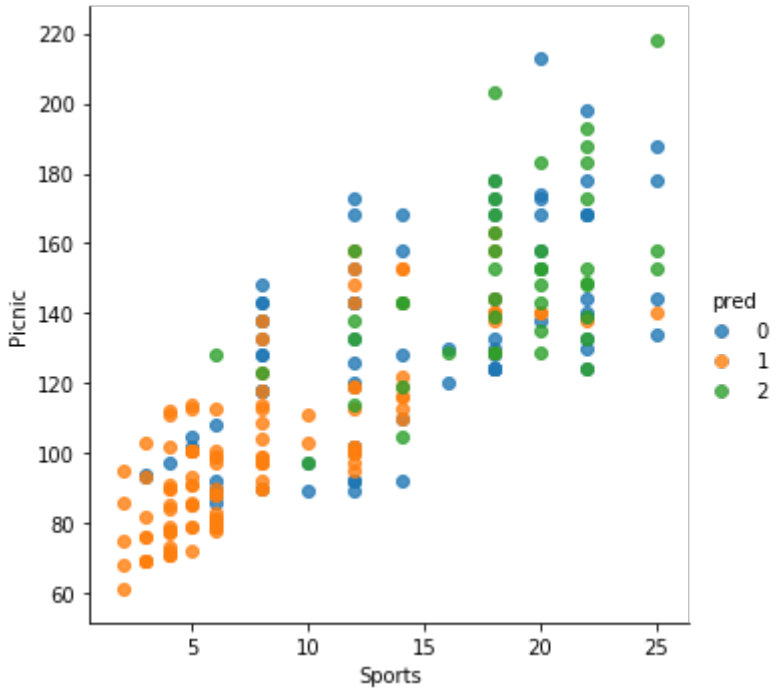
In [17]:
```
df['pred'] = kmeans.predict(X)
df.head(10)
```

Out[17]:

	Sports	Religious	Nature	Theatre	Shopping	Picnic	pred
0	2	77	79	69	68	95	1
1	2	62	76	76	69	68	1
2	2	50	97	87	50	75	1
3	2	68	77	95	76	61	1
4	2	98	54	59	95	86	1
5	3	52	109	93	52	76	1
6	3	64	85	82	73	69	1
7	3	54	107	92	54	76	1
8	3	64	108	64	54	93	1
9	3	86	76	74	74	103	1

In [18]:
```
sns.lmplot('Sports','Picnic',scatter=True,fit_reg=False,data=df,hue='pred')
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x7f4e0b62b790>
```



In [19]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn import datasets
```

In [20]:
```
df = pd.read_csv("/home/machine/Downloads/buddymove_holidayiq.csv")
```

In [21]:
```
print(df.head())
```

```
  User Id  Sports  Religious  Nature  Theatre  Shopping  Picnic
0  User 1       2         77      79       69        68      95
1  User 2       2         62      76       76        69      68
2  User 3       2         50      97       87        50      75
3  User 4       2         68      77       95        76      61
4  User 5       2         98      54       59        95      86
```

In [22]:
```
df = df.drop(['User Id'],1)
```

In [23]:
```
df
```

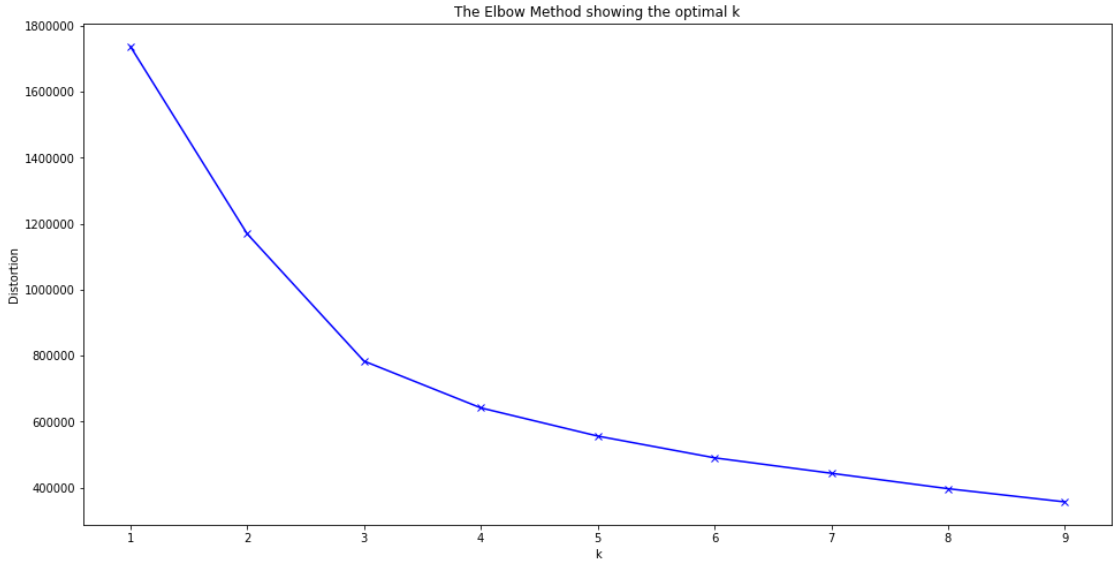
Out[23]:

	Sports	Religious	Nature	Theatre	Shopping	Picnic
0	2	77	79	69	68	95
1	2	62	76	76	69	68
2	2	50	97	87	50	75
3	2	68	77	95	76	61
4	2	98	54	59	95	86
...	...	...	...	...	...	...
244	18	139	148	129	129	168
245	22	114	228	104	84	168
246	20	124	178	104	158	174
247	20	133	149	139	144	213
248	20	143	149	139	159	143

249 rows × 6 columns

In [24]:
```
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(df)
    distortions.append(kmeanModel.inertia_)
```

In [25]:
```
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



In [ ]:
```
```