In [46]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import sklearn
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import scale
import sklearn.metrics as sm
from sklearn import datasets

%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
```

Cargamos los datos de entrada del archivo csv

In [47]:
```
dataframe = pd.read_csv(r"Videojuegos.csv") #Base de datos
dataframe.head()
```

Out[47]:

	Platform	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Rating	Critic
0	Wii	Sports	Nintendo	41.36	28.96	3.77	8.45	82.54	E	
1	Wii	Racing	Nintendo	15.68	12.80	3.79	3.29	35.57	E	
2	Wii	Sports	Nintendo	15.61	10.95	3.28	2.95	32.78	E	
3	DS	Platform	Nintendo	11.28	9.15	6.50	2.88	29.81	E	
4	Wii	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	E	

In [48]:
```
dataframe.describe()
```

Out[48]:

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	ENTEROS
count	7112.000000	7112.000000	7112.000000	7112.000000	7112.000000	7112.000000
mean	0.388567	0.232537	0.062652	0.081347	0.765307	0.457818
std	0.953982	0.680028	0.283475	0.265864	1.936692	1.888293
min	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000
25%	0.060000	0.020000	0.000000	0.010000	0.110000	0.000000
50%	0.150000	0.060000	0.000000	0.020000	0.290000	0.000000
75%	0.390000	0.202500	0.010000	0.070000	0.742500	0.000000
max	41.360000	28.960000	6.500000	10.570000	82.540000	82.000000

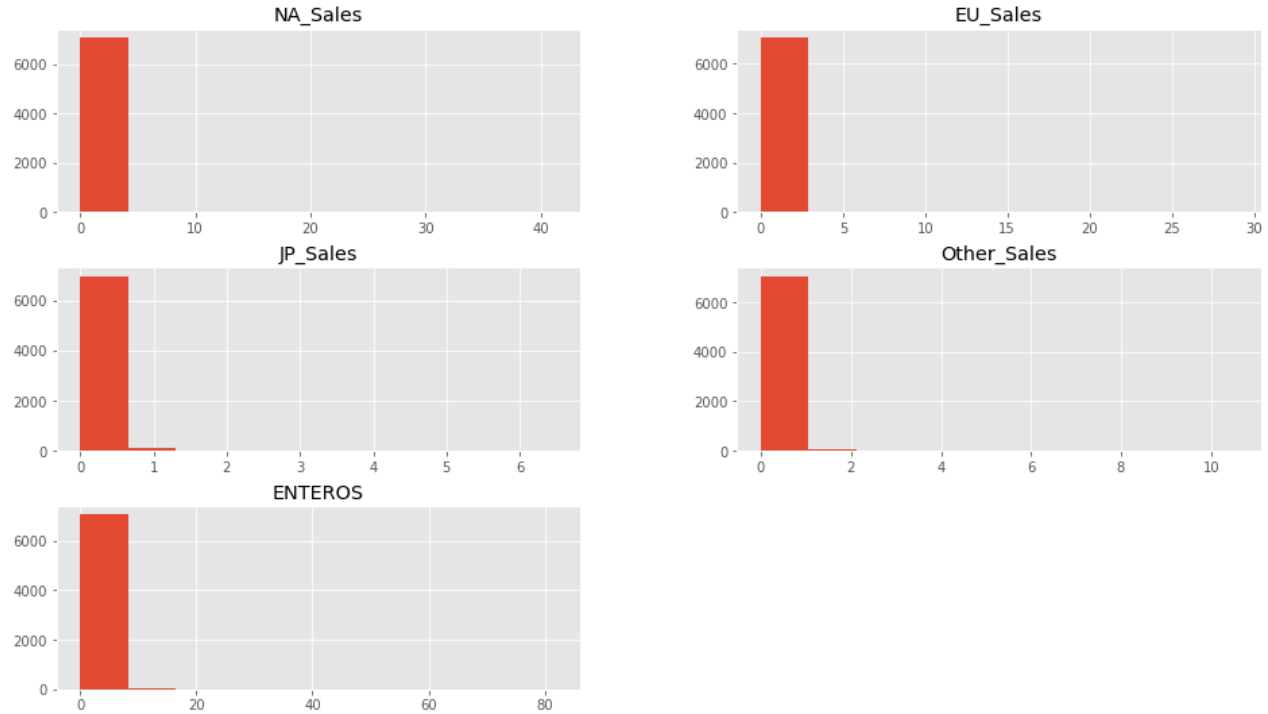
In [49]:
```
#vemos cuantos usuarios hay de cada categoría
print (dataframe.groupby('Platform').size())
```

```
Platform
3DS       161
DC         14
DS        472
GBA       249
GC        363
PC        734
PS        154
PS2      1169
PS3       790
PS4       255
PSP       401
PSV       125
Wii       493
WiiU       89
X         586
X360      888
XOne      169
dtype: int64
```

Las categorias son: 1-actores 2-cantantes 3-modelo 4-TV 5-radio 6-tecnología 7-deportes 8-política 9-escritor

Visualizamos los datos

In [50]:
```
dataframe.drop(['Global_Sales'],1).hist()
plt.show()
```

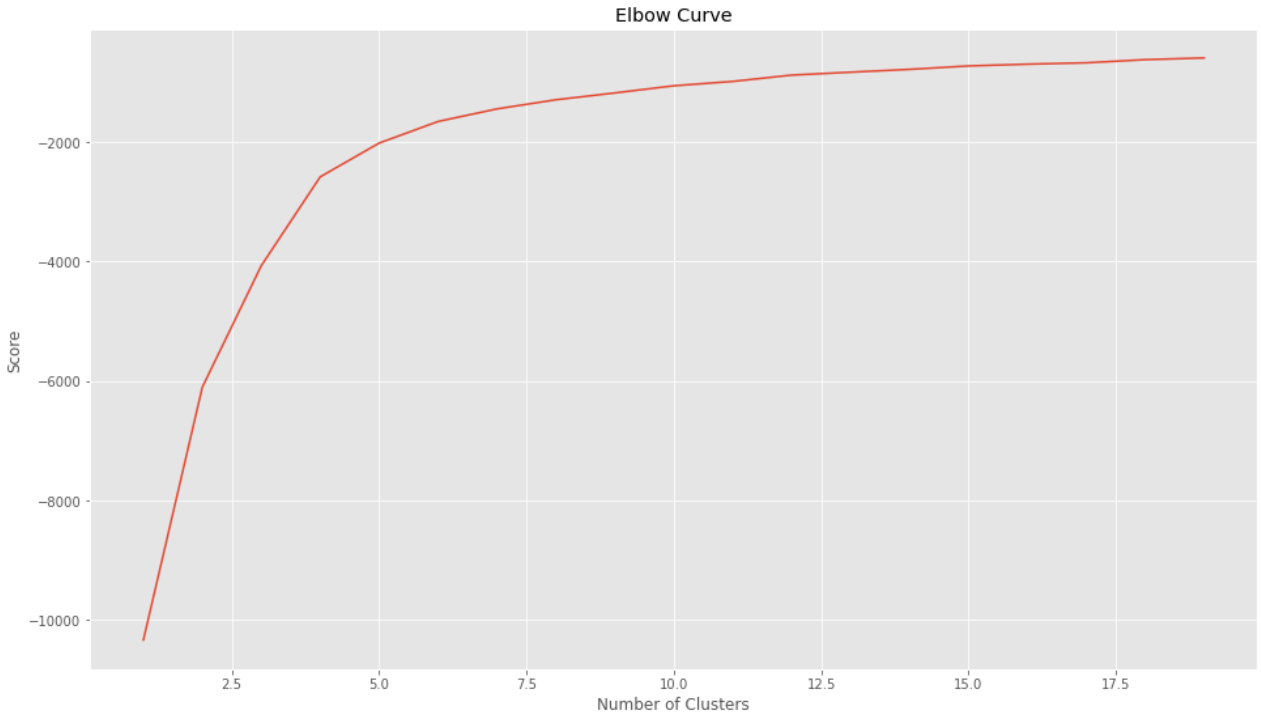


In [54]:
```
#Para el ejericio, solo seleccionamos 3 dimensiones, para poder graficarlo
X = np.array(dataframe[["EU_Sales", "JP_Sales", "NA_Sales"]])
Y = np.array(dataframe['ENTEROS'])
X.shape
```

Out[54]: (7112, 3)

Bucamos el valor de k

In [60]:
```
Nc = range(1, 20)
kmeans = [KMeans(n_clusters=i) for i in Nc]
kmeans
score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]
score
plt.plot(Nc,score)
plt.xlabel('Number of Clusters')
plt.ylabel('Score')
plt.title('Elbow Curve')
plt.show()
```



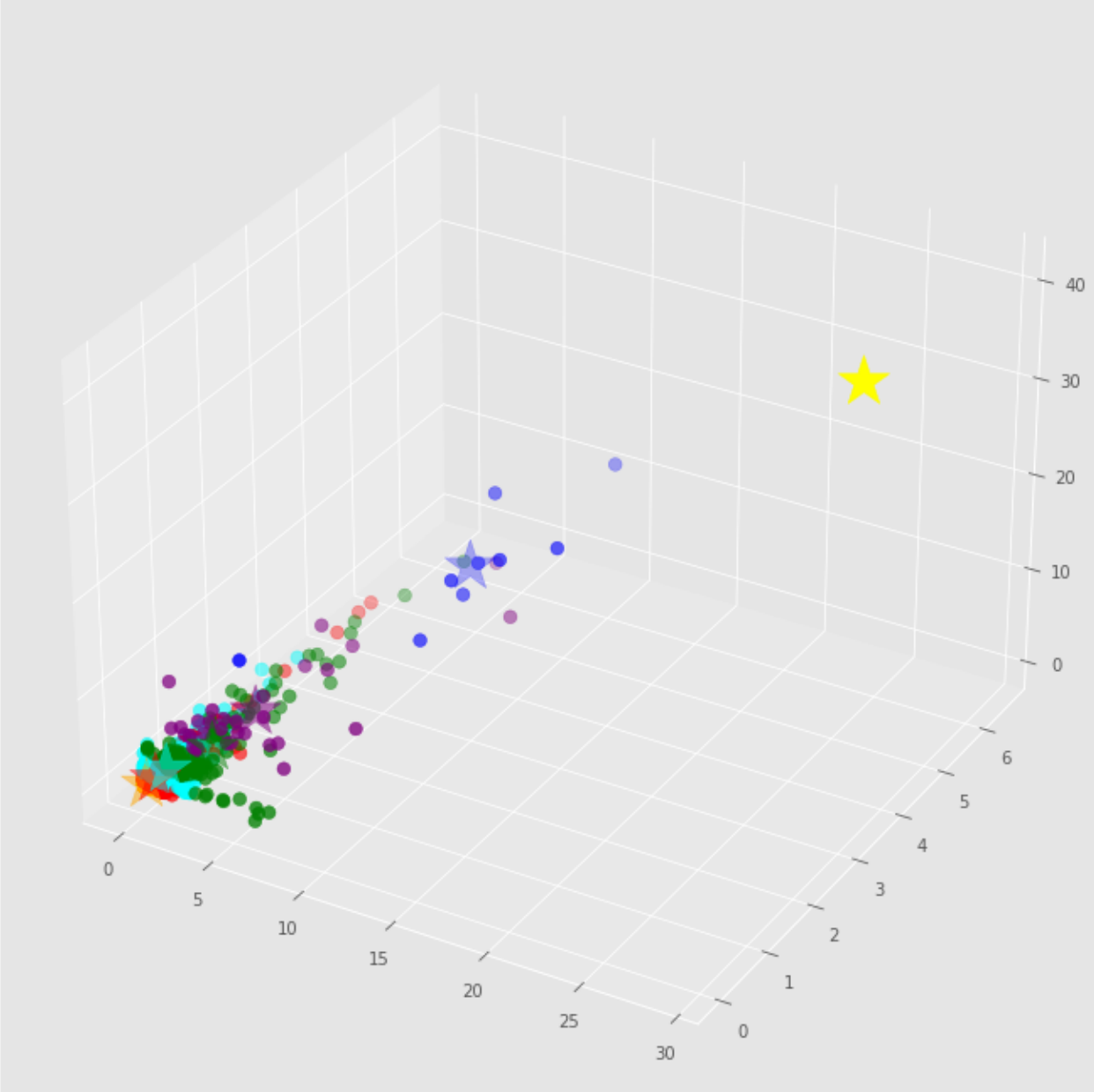
In [69]:
```
# Para el ejercicio, elijo 7 como un buen valor de K. Pero podría ser otro.
kmeans = KMeans(n_clusters=7).fit(X)
centroids = kmeans.cluster_centers_
print(centroids)
```

```
[[3.79141296e-01 9.29117877e-02 6.28212334e-01]
 [2.21756098e+00 5.76422764e-01 2.80707317e+00]
 [8.66000000e+00 3.52222222e+00 1.26300000e+01]
 [8.41777778e-01 1.49111111e-01 1.64488889e+00]
 [2.89600000e+01 3.77000000e+00 4.13600000e+01]
 [4.02062500e+00 8.19062500e-01 6.56625000e+00]
 [6.72370901e-02 2.64417640e-02 1.23665662e-01]]
```

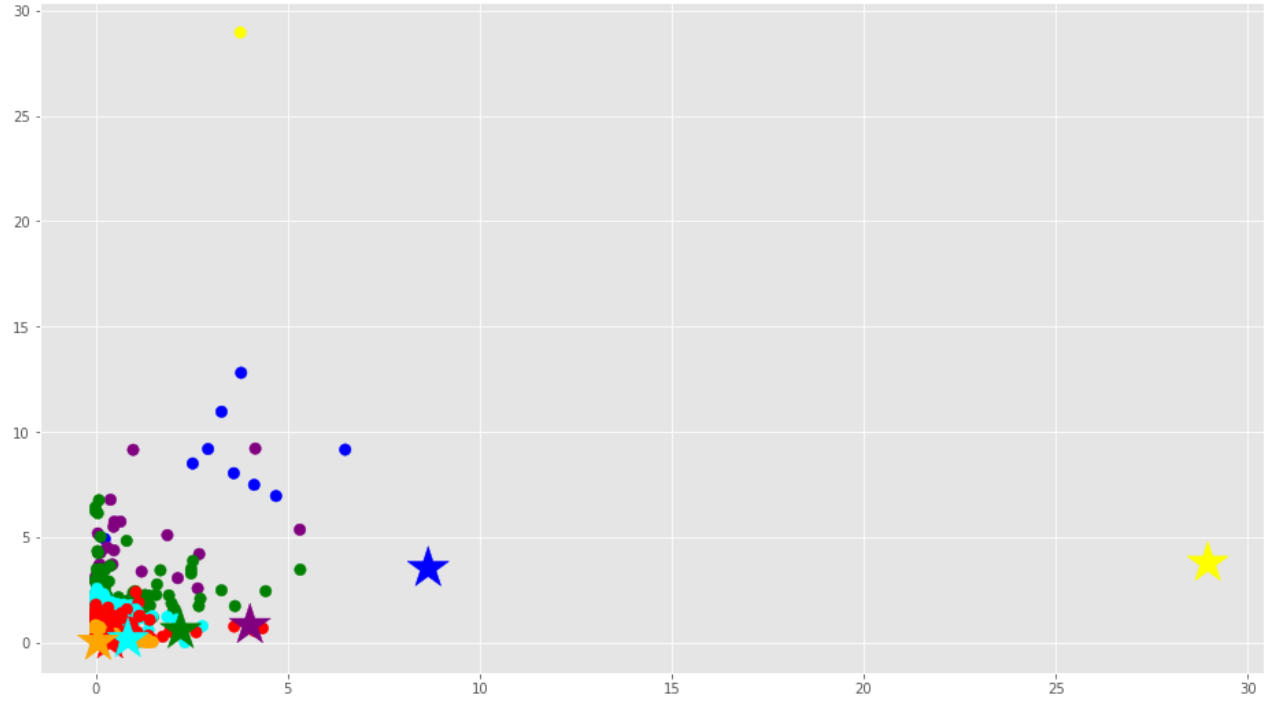
In [70]:
```
# Obtenemos las etiquetas de cada punto de nuestros datos
labels = kmeans.predict(X)
# Obtenemos los centroids
C = kmeans.cluster_centers_
colores=['red','green','blue','cyan','yellow', 'purple', 'orange']
asignar=[]
for row in labels:
    asignar.append(colores[row]);

fig = plt.figure()
ax = Axes3D(fig)
```

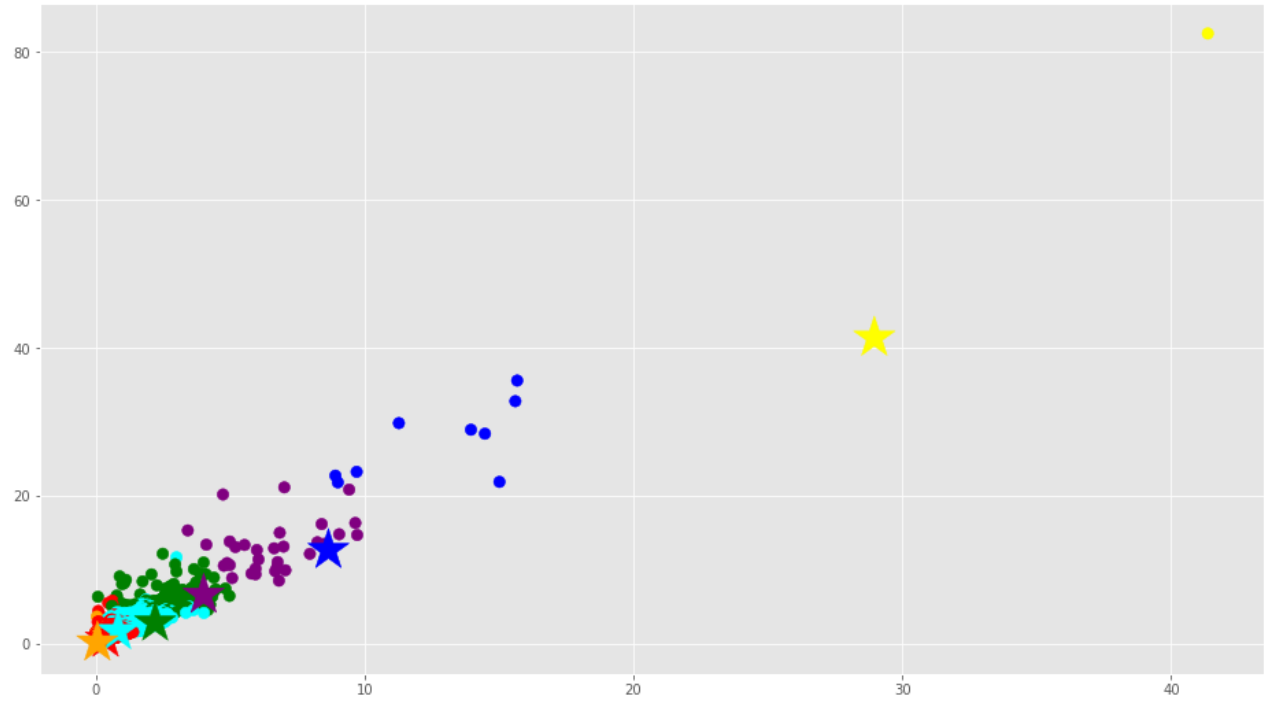
```
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=asignar,s=60)
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='*', c=colores, s=1000);
```



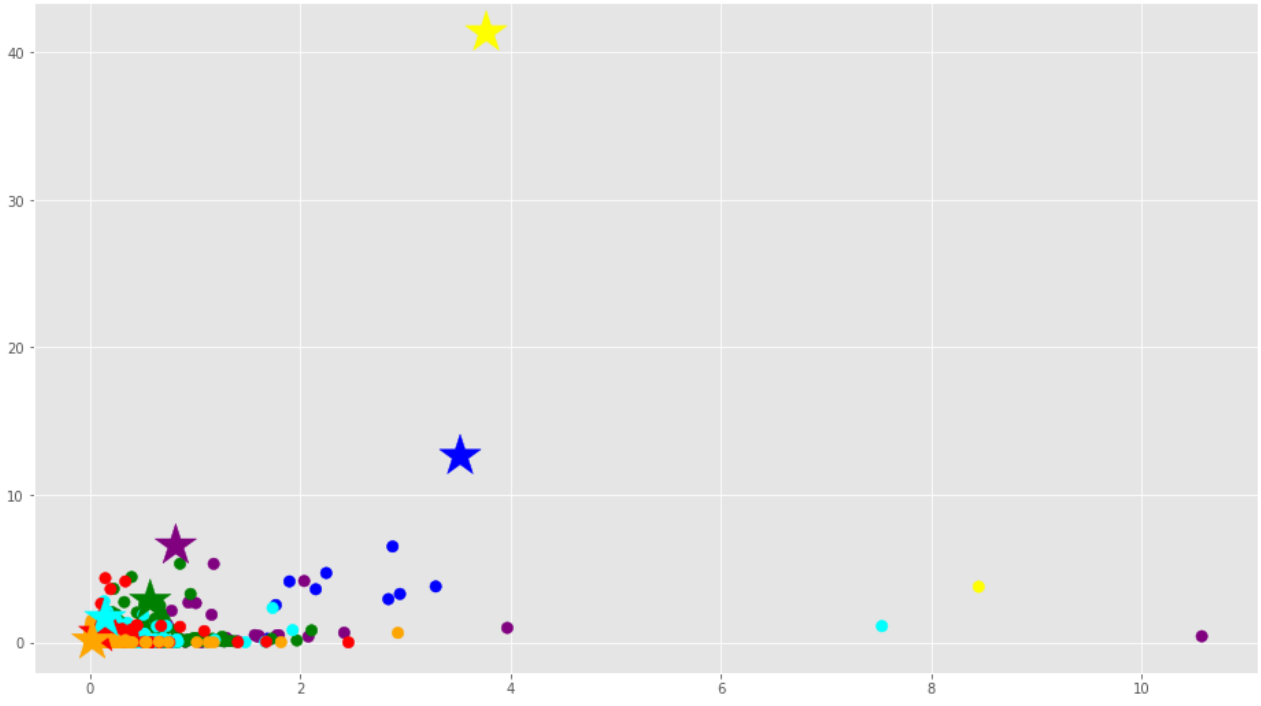
In [71]:
```
# Hacemos una proyección a 2D con los diversos ejes
f1 = dataframe['JP_Sales'].values
f2 = dataframe['EU_Sales'].values
plt.scatter(f1, f2, c=asignar, s=70)
plt.scatter(C[:, 0], C[:, 1], marker='*', c=colores, s=1000)
plt.show()
```



In [72]:
```
# Hacemos una proyección a 2D con los diversos ejes
f1 = dataframe['NA_Sales'].values
f2 = dataframe['Global_Sales'].values
plt.scatter(f1, f2, c=asignar, s=70)
plt.scatter(C[:, 0], C[:, 2], marker='*', c=colores, s=1000)
plt.show()
```



In [73]:
```
f1 = dataframe['Other_Sales'].values
f2 = dataframe['JP_Sales'].values
plt.scatter(f1, f2, c=asignar, s=70)
plt.scatter(C[:, 1], C[:, 2], marker='*', c=colores, s=1000)
plt.show()
```



Evaluando los resultados

In [76]:
```
print (classification_report(y, labels));
```

```
              precision    recall  f1-score   support

           0       0.39      0.09      0.14      5771
           1       0.00      0.00      0.00       757
           2       0.00      0.00      0.00       252
           3       0.31      0.93      0.47       122
           4       0.00      0.00      0.00        63
           5       0.00      0.00      0.00        44
           6       0.00      0.00      0.00        27
           7       0.00      0.00      0.00        17
           8       0.00      0.00      0.00         9
           9       0.00      0.00      0.00         9
          10       0.00      0.00      0.00         8
          11       0.00      0.00      0.00         3
          12       0.00      0.00      0.00         4
          13       0.00      0.00      0.00         7
          14       0.00      0.00      0.00         3
          15       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         2
          20       0.00      0.00      0.00         2
          21       0.00      0.00      0.00         3
          22       0.00      0.00      0.00         1
          23       0.00      0.00      0.00         1
          28       0.00      0.00      0.00         2
          29       0.00      0.00      0.00         1
          32       0.00      0.00      0.00         1
          35       0.00      0.00      0.00         1
          82       0.00      0.00      0.00         1

    accuracy                           0.09      7112
   macro avg       0.03      0.04      0.02      7112
weighted avg       0.32      0.09      0.12      7112
```

¿Crees que estos centros puedan ser representativos de los datos? ¿Por qué?

Si los representan debido a la relacion que tienen entre ellos, aunque si existe una rango amplio entre estos mismos datos, observando la gráfica notamos que su distribución es más densa en la parte inferior izquierda.

¿Cómo obtuviste el valor de k a usar?

El número de cluster identificados por el algoritmo, es representado por k, es un método que se usa para identificar el número de cluster necesarios para el análisis de datos, usamos diferentes funciones, entre ellas el ciclo for, con cierto rango y al final graficamos los resultados.

¿Los centros serían más representativos si usaras un valor más alto? ¿Más bajo?

Los centros tendrían más relevancia si se usara un valor más bajo, debido a la distribución de nuestros datos, en razón de que su aproximación a cero es mayor porque se calcula en escala de millones.

¿Qué distancia tienen los centros entre sí? ¿Hay alguno que este muy cercano a otros?

La distancia se encuentra entre el renago de: 0.00-0.15, demostrando que la distancia entre los centros es pequeña porque la comparación de distancia se realiza entre regiones, no obstante si la comparación fuera global la distancia seria mayor.

¿Qué pasaría con los centros si tuviéramos muchos outliers en el análisis de cajas y bigotes?

Tendria una variación amplia entre distancias existentes sobre los datos, en el análisis de cajas y bigotes los datos no estaría dentro de la región de las cajas, sino fuera de ellas.

¿Qué puedes decir de los datos basándose en los centros?

Que se puede encontrar una relación entre las ventas por regiones y la cantidad de titulos que existen por consola.