In [141…
```
#Andrea González, Juan Pablo Cobos, Xavier Barrera y Olivia Morales
#En esta parte del código isamos diferentes librerías, por lo que las llamamos, y tambi
import pandas as pd
import seaborn as sb
import numpy as np; np.random.seed(0)
import matplotlib.pyplot as plt
data = pd.read_csv ('Videojuegos.csv')
datos = pd.read_csv ('Videojuegos.csv')
from matplotlib import cm
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
#Voy a revisar dimensiones
data.shape
```

Out[141… (7112, 10)

In [2]:
```
#Aquí es para observar las primeras filas de nuestra base de datos
data.head()
```

Out[2]:

	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 [3]:
```
#Aquí podemos observar diferentes características como el máximo, mínimo, desviación es
data.describe()
```

Out[3]:

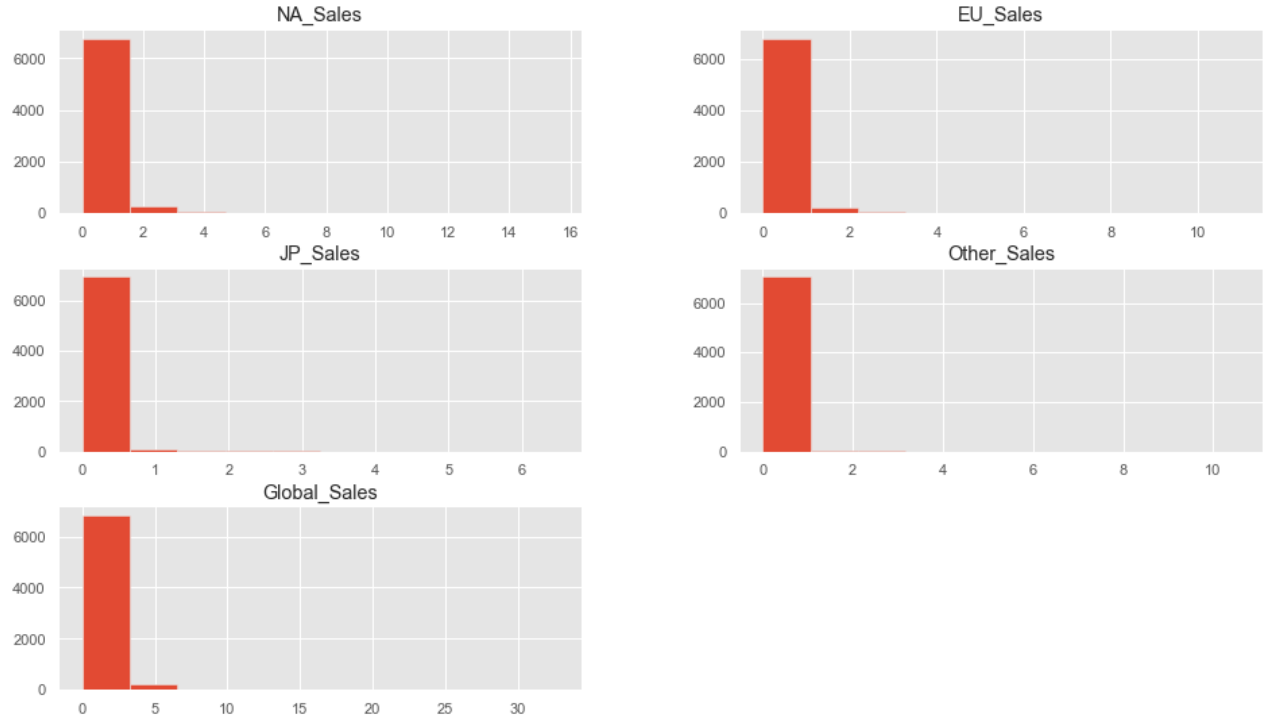
	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	7112.000000	7112.000000	7112.000000	7112.000000	7112.000000
mean	0.388567	0.232537	0.062652	0.081347	0.765307
std	0.953982	0.680028	0.283475	0.265864	1.936692
min	0.000000	0.000000	0.000000	0.000000	0.010000
25%	0.060000	0.020000	0.000000	0.010000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

Puedes ver las estadísticas de todos los campos, ayer vimos de uno en particular escribir texto de mis datos que sean interesantes.

Visualización general

Eliminar etiquetas de filas o columnas

In [217…
```
#Con esta función observamos el historiograma de cada columna de nuestra base de datos
#en diferentes regiones
data.drop ([0,1]).hist()
plt.show()
```



Filtros

In [5]:
```
#En las siguientes opciones encontramos filtros, que nos dan de resultado de diferentes
mas_de_5 = data[data['Global_Sales']>5]
mas_de_5
```

Out[5]:

	Platform	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Rating	Cr
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	
...	...	...	...	...	...	...	...	...	...	
142	DS	Action	Nintendo	1.85	1.80	0.95	0.48	5.08	E	
143	PS3	Sports	Electronic Arts	0.61	3.28	0.06	1.12	5.07	E	
144	PS	Action	Virgin Interactive	2.05	1.16	1.11	0.73	5.05	M	
145	PSP	Action	Take-Two Interactive	1.70	1.99	0.16	1.19	5.03	M	

	Platform	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Rating	Cr
146	PS	Sports	Activision	3.42	1.38	0.02	0.20	5.02	T	

147 rows × 10 columns

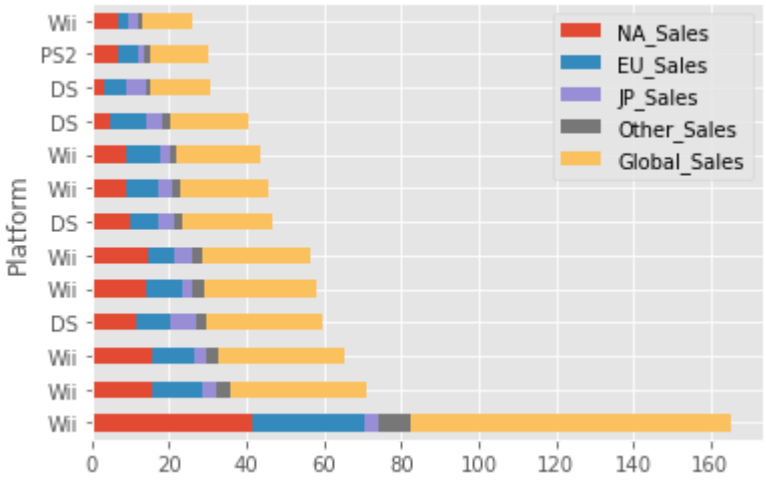
In [20]:
```
doble_filtro = data[(data['Global_Sales'] > 5) & (data['EU_Sales'] > 1) & (data['JP_Sal
& (data['NA_Sales'] > 1)]
doble_filtro
```

Out[20]:

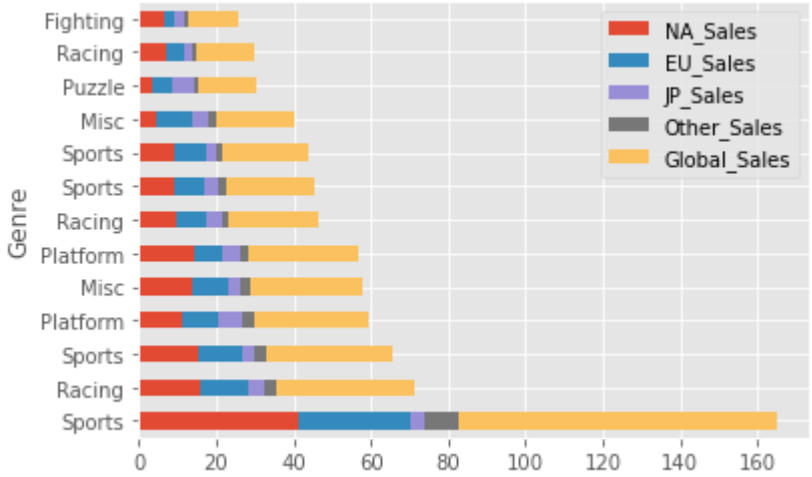
	Platform	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Rating
0	Wii	Sports	Nintendo	41.36	28.96	3.77	8.45	82.54	E
1	Wii	Racing	Nintendo	15.68	12.80	3.79	3.29	35.57	E
2	Wii	Sports	Nintendo	15.61	10.95	3.28	2.95	32.78	E
3	DS	Platform	Nintendo	11.28	9.15	6.50	2.88	29.81	E
4	Wii	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	E
5	Wii	Platform	Nintendo	14.48	6.95	4.70	2.25	28.38	E
6	DS	Racing	Nintendo	9.71	7.48	4.13	1.90	23.22	E
7	Wii	Sports	Nintendo	8.92	8.03	3.60	2.15	22.70	E
9	Wii	Sports	Nintendo	9.01	8.49	2.53	1.77	21.79	E
12	DS	Misc	Nintendo	4.74	9.20	4.16	2.04	20.15	E
15	DS	Puzzle	Nintendo	3.43	5.35	5.32	1.18	15.29	E
16	PS2	Racing	Sony Computer Entertainment	6.85	5.09	1.87	1.16	14.98	E
26	Wii	Fighting	Nintendo	6.64	2.56	2.66	1.01	12.87	T

Visualización

In [22]:
```
doble_filtro.set_index('Platform').plot.barh(stacked=True);
```



In [27]:
```
doble_filtro.set_index('Genre').plot.barh(stacked=True);
```



In [119…
```
df=pd.DataFrame(data['Platform'])
df
```

Out[119…

	Platform
0	Wii
1	Wii
2	Wii
3	DS
4	Wii
…	…
7107	PC
7108	PC
7109	PC
7110	PC
7111	PS4

7112 rows × 1 columns

In [218…
```
d= data['Global_Sales']
d
```

Out[218…
```
0       82.54
1       35.57
2       32.78
3       29.81
4       28.92
        ...
7107     0.01
7108     0.01
7109     0.01
7110     0.01
7111     0.01
Name: Global_Sales, Length: 7112, dtype: float64
```

In [ ]:
```
#En esta sección de código hicimos ciclos for, para hacer una gráfica donde comparemos
```

In [52]:
```
a = "Wii";
k = 0;
for i in df['Platform']:
    if i==a:
        k+=1
print (k)
```

```
493
```

In [59]:
```
b = "WiiU";
z = 0;
for i in df['Platform']:
    if i==b:
        z+=1
print (z)
```

```
89
```

In [60]:
```
c = "DS";
y = 0;
for i in df['Platform']:
    if i==c:
        y+=1
print (y)
```

```
472
```

In [61]:
```
d = "X360";
x = 0;
for i in df['Platform']:
    if i==d:
        x+=1
print (x)
```

```
888
```

In [83]:
```
e = "PS2";
w = 0;
for i in df['Platform']:
    if i==e:
        w+=1
print (w)
```

```
1169
```

In [65]:
```
f = "PS3";
v = 0;
for i in df['Platform']:
    if i==f:
        v+=1
print (v)
```

```
790
```

In [68]:
```
g = "PS4";
u = 0;
for i in df['Platform']:
    if i==g:
        u+=1
print (u)
```

```
255
```

In [70]:
```
h = "3DS";
t = 0;
for i in df['Platform']:
    if i==h:
        t+=1
print (t)
```

```
161
```

In [72]:
```
j = "PS";
s = 0;
for i in df['Platform']:
    if i==j:
        s+=1
print (s)
```

```
154
```

In [87]:
```
f2 = "X";
v2 = 0;
for i in df['Platform']:
    if i==f2:
        v2+=1
print (v2)
```

```
586
```

In [76]:
```
f3 = "PC";
v3 = 0;
for i in df['Platform']:
    if i==f3:
        v3+=1
print (v3)
```

```
734
```

In [78]:
```
f4 = "PSP";
v4 = 0;
for i in df['Platform']:
    if i==f4:
        v4+=1
print (v4)
```

```
401
```

In [80]:
```
e1 = "GC";
w1 = 0;
for i in df['Platform']:
    if i==e1:
        w1+=1
print (w1)
```

```
363
```

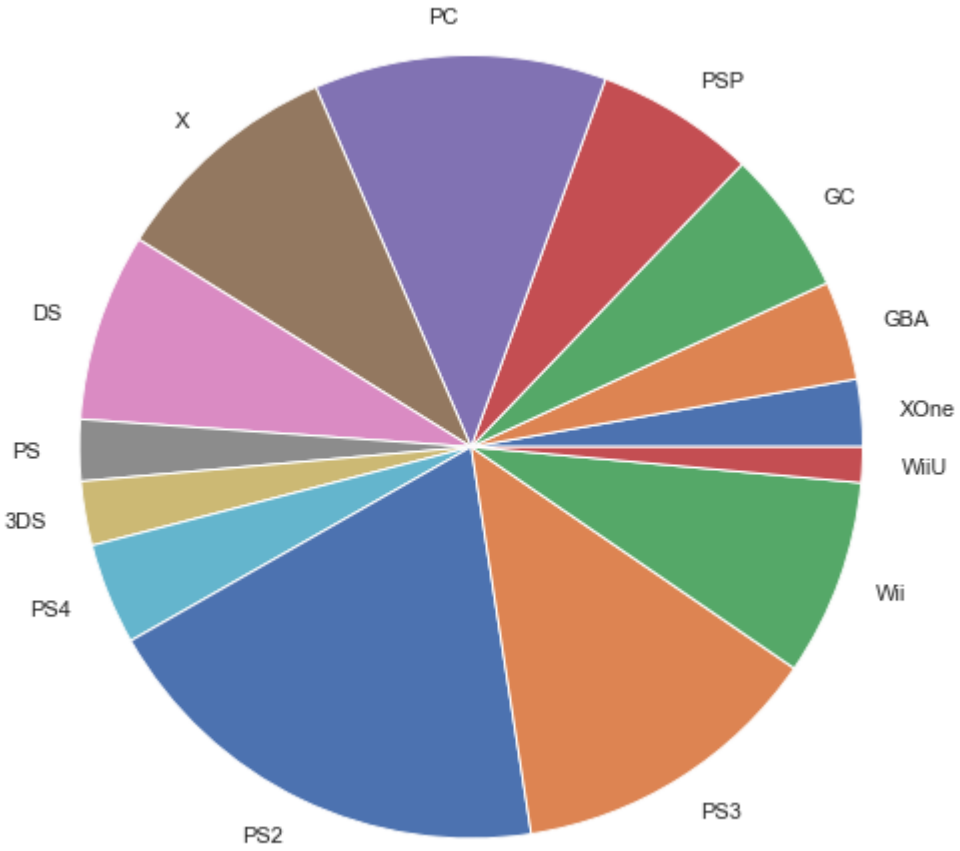
In [81]:
```
e2 = "GBA";
w2 = 0;
for i in df['Platform']:
    if i==e2:
        w2+=1
print (w2)
```

```
249
```

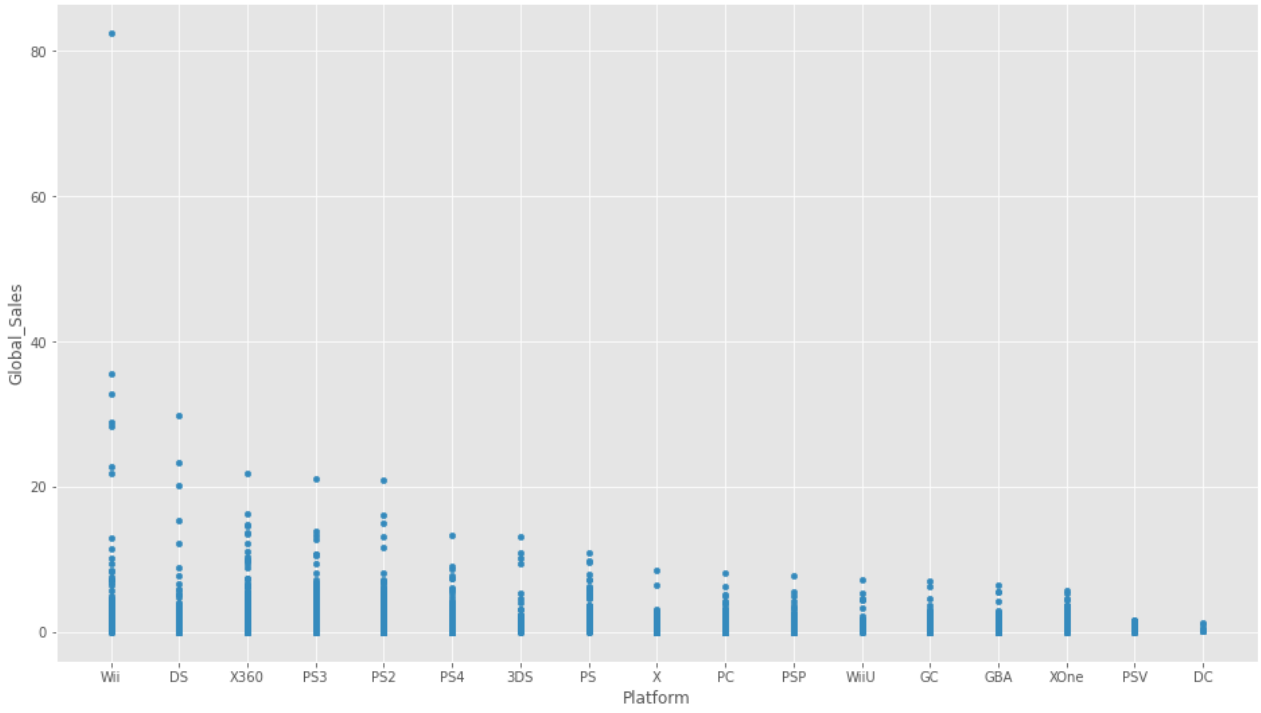
In [84]:
```
e3 = "XOne";
w3 = 0;
for i in df['Platform']:
    if i==e3:
        w3+=1
print (w3)
```

```
169
```

In [98]:
```
#Aquí podemos observar la gráfica de pastel
plataformas = ["XOne", "GBA", "GC", "PSP", "PC", "X", "DS", "PS", "3DS", "PS4", "PS2",
frecuencia = [169, 249, 363, 401, 734, 586, 472, 154, 161, 255, 1169, 790, 493, 89]
plt.rcParams['figure.figsize'] = (16, 9)
plt.pie(frecuencia, labels=plataformas)
plt.show()
```

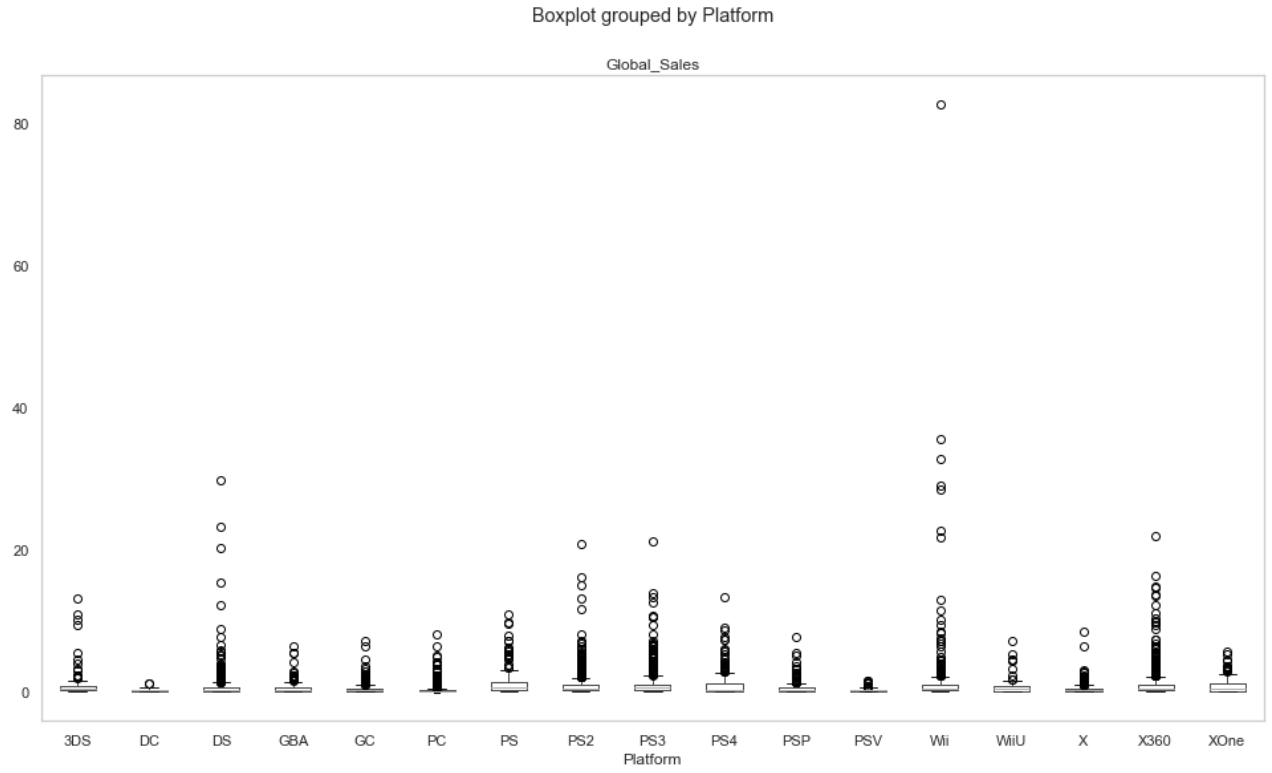


In [94]:
```
#Esta gráfica la usamos para comparar las ventas globales respecto a las plataformas
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
data.plot.scatter(x='Platform', y='Global_Sales');
```

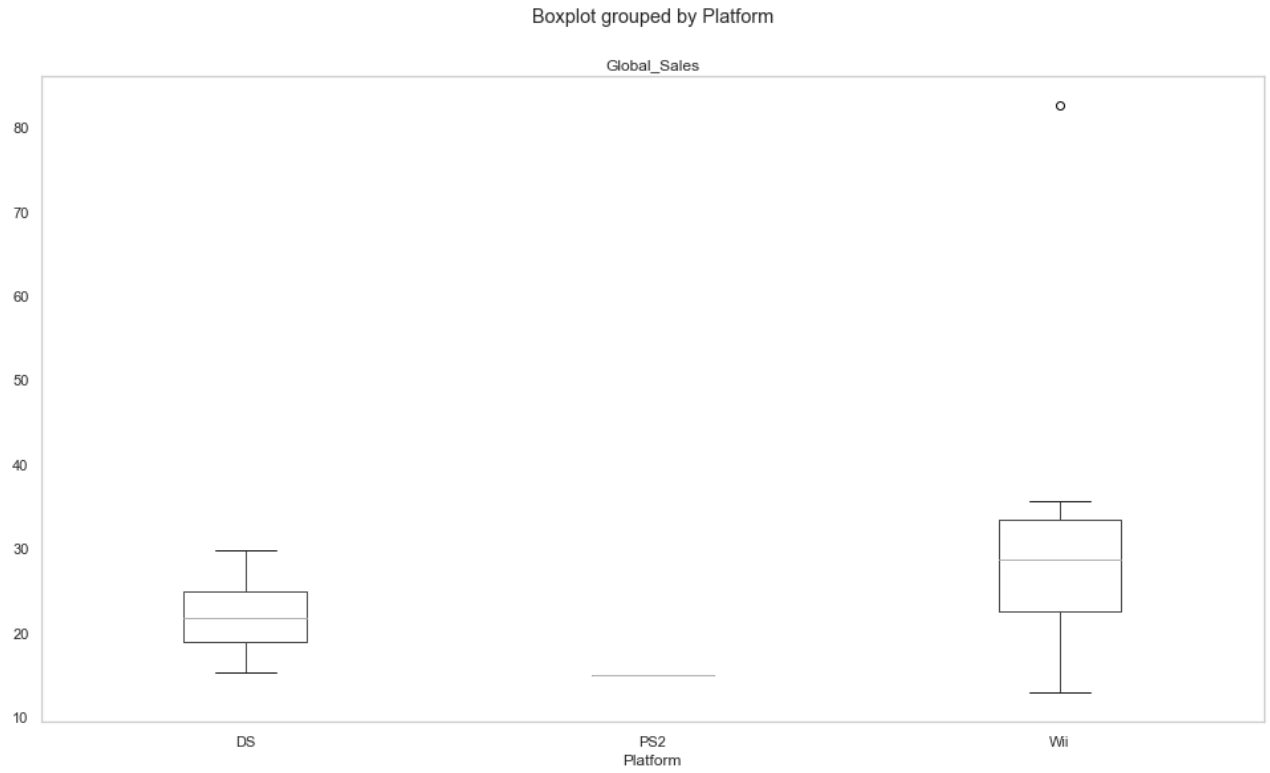


Boxplot para otener un diagrama de cajas y bigotes

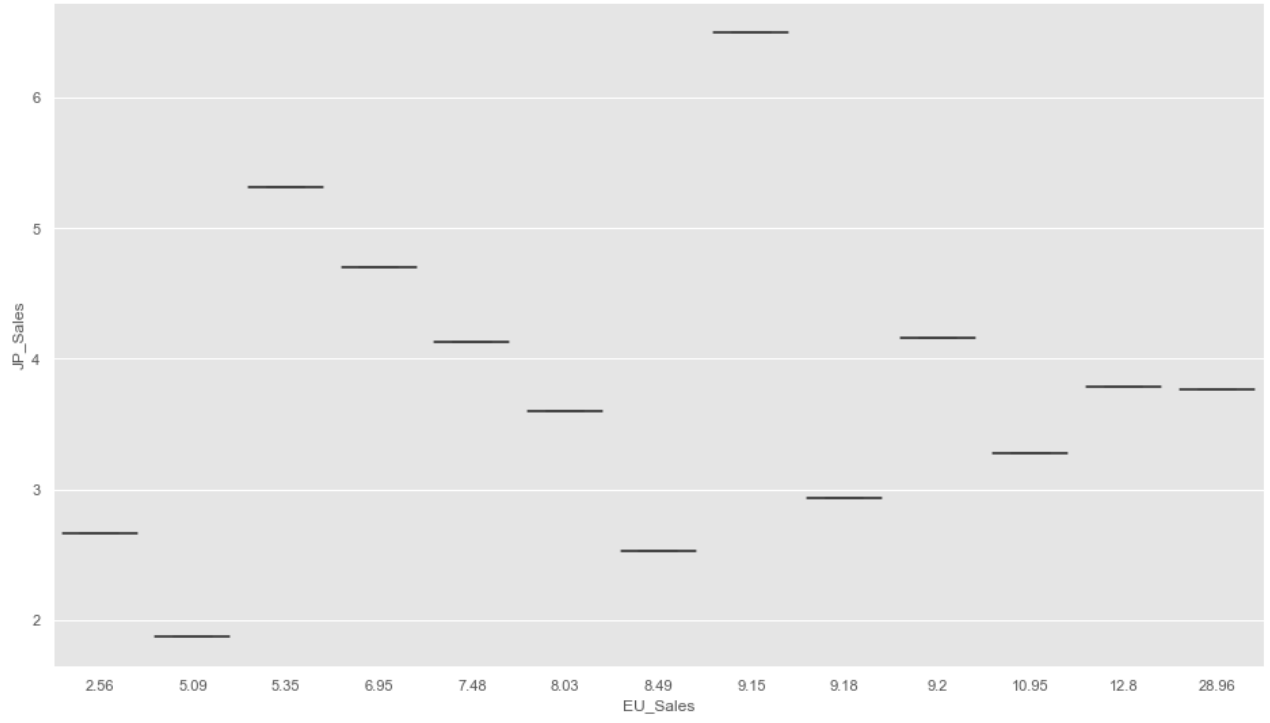
In [111…
```
#El diagrama de cajas sirve para representar una serie de datos a través de cuadriles.
plt.rcParams['figure.figsize'] = (16, 9)
data.boxplot(by ='Platform', column =['Global_Sales'], grid = False);
```



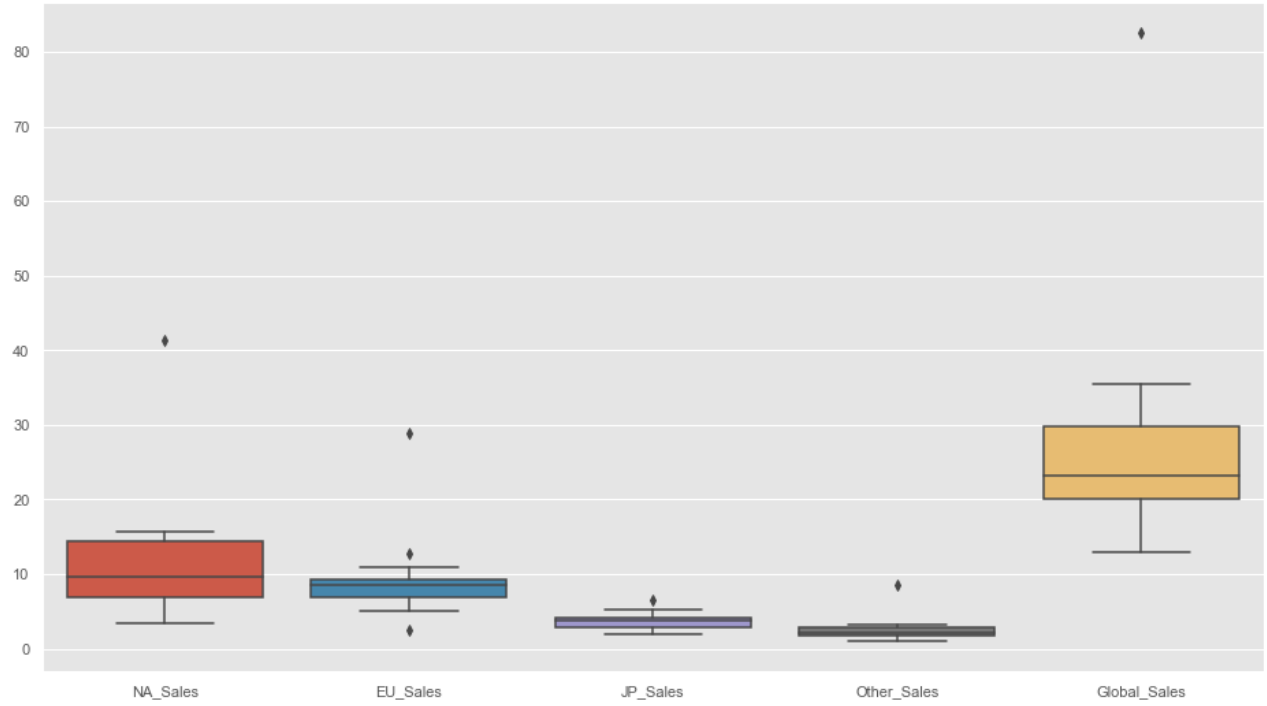
In [112…
```
plt.rcParams['figure.figsize'] = (16, 9)
doble_filtro.boxplot(by ='Platform', column =['Global_Sales'], grid = False);
```



In [168…
```
plt.rcParams['figure.figsize'] = (16, 9)
sb.boxplot(x ="EU_Sales", y ="JP_Sales", data=doble_filtro);
```



In [170…
```
# Boxplot
plt.rcParams['figure.figsize'] = (16, 9)
sb.boxplot(data=doble_filtro);
```



Correlación

In [162…
```
#En la correlación buscamos la dependencia de una respecto a la otra columna
data.corr(method = 'pearson')
```

Out[162…

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
NA_Sales	1.000000	0.838223	0.465930	0.727575	0.954936
EU_Sales	0.838223	1.000000	0.518536	0.718245	0.938461
JP_Sales	0.465930	0.518536	1.000000	0.393503	0.611947
Other_Sales	0.727575	0.718245	0.393503	1.000000	0.805426
Global_Sales	0.954936	0.938461	0.611947	0.805426	1.000000

In [171…
```
data.corr(method = 'kendall')
```

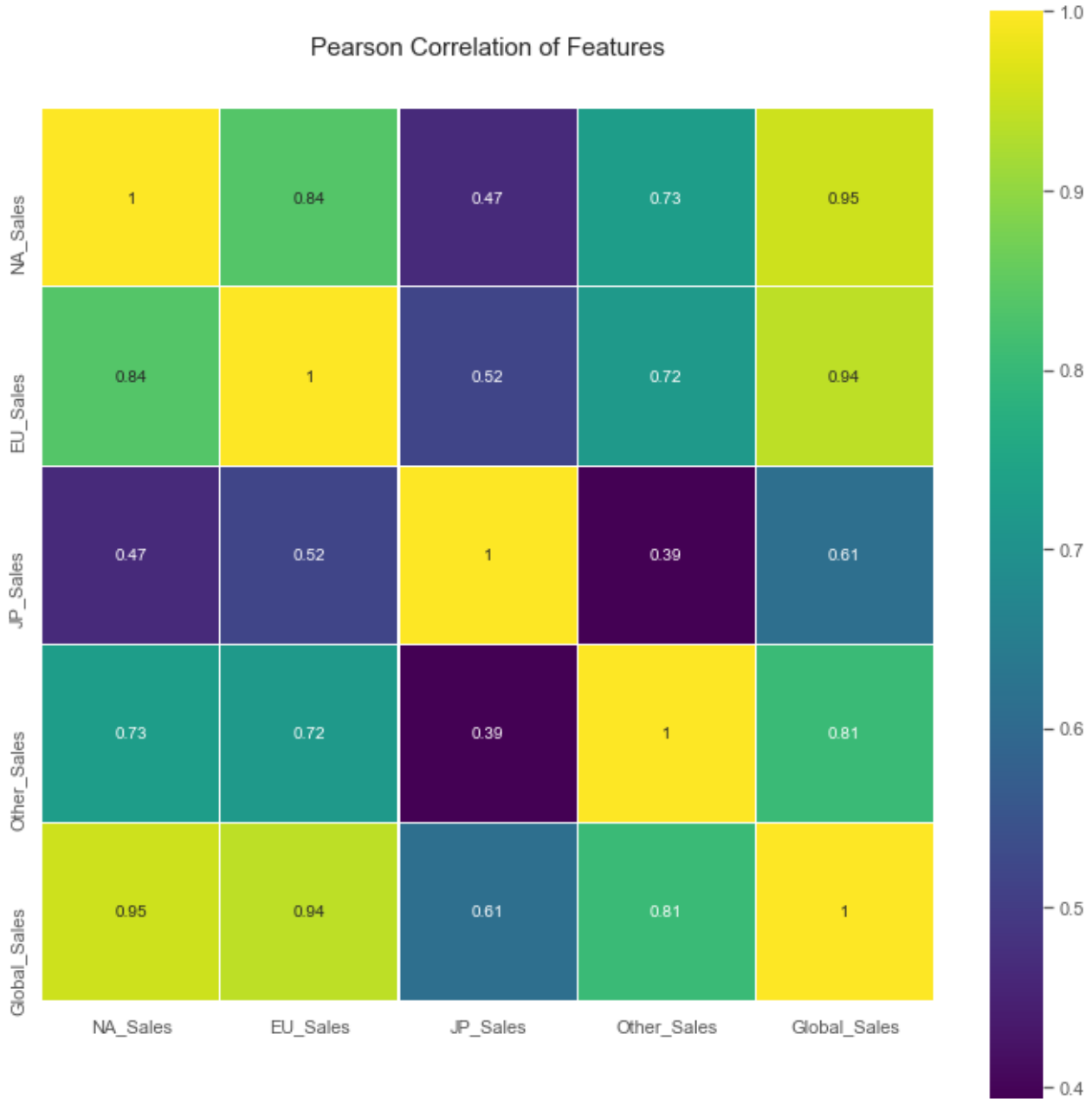
Out[171…

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
NA_Sales	1.000000	0.478510	0.203073	0.639386	0.783117
EU_Sales	0.478510	1.000000	0.198217	0.650148	0.636230
JP_Sales	0.203073	0.198217	1.000000	0.266888	0.319269
Other_Sales	0.639386	0.650148	0.266888	1.000000	0.768032
Global_Sales	0.783117	0.636230	0.319269	0.768032	1.000000

Visualizar mapa de calor

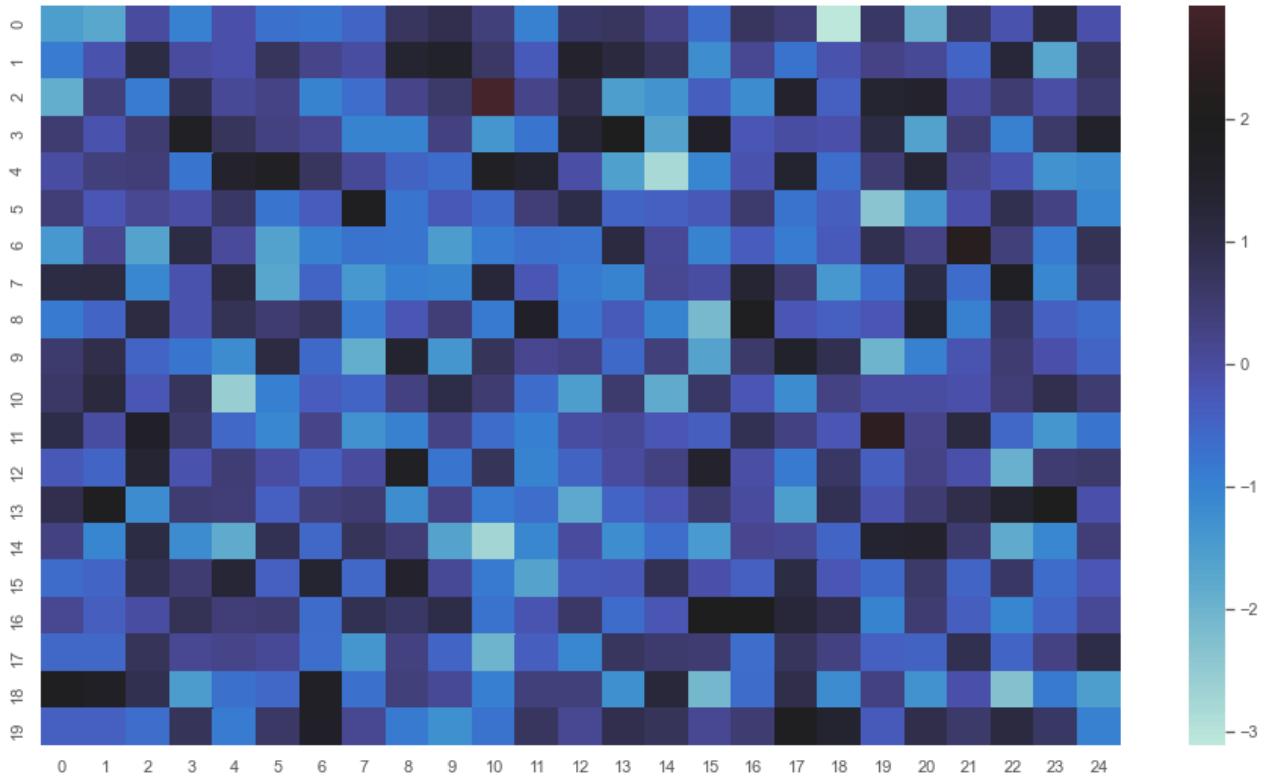
In [205…
```
#Los siguientes mapas de calor nos muestran la correlación en una diferente representac
```

```
colormap = plt.cm.viridis
plt.figure(figsize=(12,12))
plt.title('Pearson Correlation of Features', y = 1.05, size = 15)
sb.heatmap(data.corr(),linewidths=0.1, vmax=1.0, square=True,
           cmap=colormap, linecolor= 'white', annot=True);
```



In [220…

```
normal_data = np.random.randn(20,25)
ax = sb.heatmap(normal_data, center=2)
```



Responde las siguientes preguntas: Para poder responder estas preguntas primero es relevante poder describir nuestra base de datos y así comprender el objetivo detrás de su análisis. La base de datos que escogimos se basa en la venta de videojuegos a nivel mundial, la información de esta base de datos está dividida en categorías como "plataforma" "género" "distribuidora" una sección de ventas a su vez repartidas por regiones en las que se divide la base de datos, "global" corresponde al total de ventas del resto de regiones que son "Norteamérica" "Europa" "Japón" y "otros"; además de otras dos categorías que son el "rating" y la "calificación de la crítica"; todas estas corresponden, por decirlo de una forma simplificada y que se puede entender, a las columnas de nuestra base de datos y las filas son donde se encuentra toda la información según su categoría, contextualizando esto ya podemos responder algunas cuestiones.

● ¿Hay alguna variable que no aporte información? Si, parte de la estadística está en trabajar con valores numéricos, nuestra base de datos cuenta con varios datos no numéricos de los cuales no pudimos sacar mucha información que vaya con nuestro propósito, estas son justamente las variables que no nos aportan información.

● Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué? Es de las variables que mencionamos arriba, como por ejemplo son las categorías de "rating" o "calificación de la crítica" ya que nuestro enfoque no va dirigido a saber cuál es el mejor videojuego o si a la crítica le gusta, no la podemos incluir; además consideramos que no hay demasiada variedad en estas clasificaciones con las que podamos verdaderamente trabajar para sacar conclusiones, por lo tanto nosotros las ignoraremos porque las consideramos como información no relevante a nuestro análisis y no está en las gráficas para esta actividad.

● ¿Existen variables que tengan datos extraños? Extraños como tal, no realmente, checando el resumen elaborado por nuestro programa utilizando Python-pandas podemos ver cómo en realidad

los valores únicos son la gran mayoría de nuestros datos y no tenemos omisiones de información, además en toda la base se respeta el tipo de dato según su categoría.

● Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte? Al comparar las variables nos damos cuenta que los rangos que tenemos son bastante amplios, lo que pasa es que cómo la información que queremos analizar son las ventas y en nuestra base de datos está contabilizada en millones si llegamos a tener números decimales muy muy pequeños que no podemos tratar de la misma forma que los más grandes, por lo que sí nos afecta nuestro análisis; y se puede ver en algunas de las gráficas que generamos para corroborar esto, principalmente en el primer diagrama de caja y la gráfica de dispersión, donde contrastamos las ventas totales de toda nuestra base de datos con la plataforma a la que corresponde el videojuego, la concentración de datos está dada en la parte inferior de la gráfica donde los valores están dados por por números decimales muy pequeños, aún así nuestro análisis está enfocado en el mayor número de ventas por lo tanto fue necesario realizar algunos filtros, en nuestros filtros lo que decidimos hacer es tomar de las ventas globales aquellas que sean mayores a 5 millones y que en cada región solo se tomen en cuenta las ventas mayores a 1 millón.

● ¿Puedes encontrar grupos qué se parezcan? ¿Qué grupos son estos? Una vez generados estos filtros es más fácil para nosotros encontrar grupos que se parezcan, estos grupos están orientados a que sus ventas tengan valores similares o en un rango que nosotros ya establecimos, según esto generamos otro diagrama de caja en la cual ya no nos aparecían tantas plataformas como en el primer caso donde consideramos todos datos sin filtros, una vez poniendo nuestros filtros la gráfica quedó resumida a sólo tres plataformas, de alguna forma esto fue útil también para visualizar de forma más clara las ventas según cada región.