Downloading From: https://drivers/nikhils/100% df=pd.read_csv("/ldf Rank 0 2061 1 9137	### Platform Vear Genre Publisher NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales Shin Chan Flipa en colores! DS 2007.0 Platform DS 2007.0 Platform DS CS CS CS CS CS CS CS
2 14279 3 8359 4 7109 16647 7925 16648 6279 16649 6977 16650 15422 16651 12919 Zyuden S 16652 rows × 11 column	.hack: Sekai no Mukou ni + Versus PS3 2012.0 Action Namco Bandai Games 1.145709 1.762339 1.493442 0.408693 4.982552 .hack//G.U. Vol.1//Rebirth PS2 2006.0 Role-Playing Namco Bandai Games 2.031986 1.389856 3.228043 0.394830 7.226880 .hack//G.U. Vol.2//Reminisce PS2 2006.0 Role-Playing Namco Bandai Games 2.792725 2.592054 1.440483 1.493442 8.363113 Zumba Fitness Rush X360 2012.0 Sports 505 Games 4.409308 3.167419 4.168474 1.087977 13.053204 Zumba Fitness: World Party Wii 2013.0 Misc Majesco Entertainment 3.033887 2.792725 1.596852 1.493442 8.878837 Zweil! PSP 2008.0 Role-Playing Falcom Corporation 1.087977 0.592445 1.087977 0.394830 3.509168 entai Kyoryuger: Game de Gaburincho!! 3DS 2013.0 Action Namco Bandai Games 1.081046 1.714664 2.004268 0.39
import matplotlib import seaborn as df.shape (16652, 11) # univariate Analy x_val=[0,1,2] y_val=[3,4,6]	sns
plt.plot(x_val,y_val) [<matplotlib.lines 3.5<="" 4.0="" 4.5="" 5.0="" 5.5="" 6.0="" td=""><td>Line2D at 0x7feccacee880>]</td></matplotlib.lines>	Line2D at 0x7feccacee880>]
# plt.plot?	90 39 88 10
Racing 124 Platform 88 Simulation 86 Fighting 84 Strategy 66 Puzzle 56 Name: Genre, dtype x_bar=cat_counts.i y_bar=cat_counts plt.bar(x_bar,y_ba	49 86 67 48 81 82 : int64 index
3000 - 2500 - 2000 - 1500 - 1000 -	iSjiogktvent Bæc i Rigst Eismul atignt Stgate glyszzle
plt.figure(figsize plt.bar(x_bar,y_bar)) <pre></pre>	ar)
2000 - 1500 - 1000 -	
plt.figure(figsize plt.bar(x_bar,y_ba plt.xticks(rotatio plt.show()	e=(12,8)) ar)
2000 - 1500 - 1000 -	
plt.figure(figsize plt.barh(x_bar,y_k plt.xticks(rotatio plt.show()) Puzzle Strategy	par)
Fighting - Simulation - Platform - Racing - Adventure - Shooter - Role-Playing - Misc -	
plt.figure(figsize plt.bar(x_bar,y_ba plt.xticks(rotatio plt.show()	ar,width=0.2)
3000 - 2500 - 2000 -	
1000 - 500 - kktion sports # plt.bar?	Mesc Robertsein Stantes Ratins Autorn Struktur Highting States Partie
<pre>plt.xticks(rotation plt.yticks(rotation plt.xlabel("Genre" plt.ylabel("Count")</pre>	ar,width=0.2,color="orange") on=45,fontsize=12) on=45,fontsize=12) ',fontsize=12)
Solution to the solution of th	
plt.figure(figsize sns.countplot(data	a=df, enre")
3000 - 2500 - 2000 -	
1000 - Shootes Alator	Action to the state of the stat
plt.figure(figsize sns.countplot(data x="Ge order plt.xticks(rotation plt.show())	a=df, enre", r=df["Genre"].value_counts().index)
2000 - 1500 - 1000 -	
plt.figure(figsize sns.countplot(data y="Ge	a=df, enre", r=df["Genre"].value_counts().index)
Sports - Misc - Role-Playing - Shooter - Adventure - Platform - Simulation -	
Fighting - Strategy - Puzzle - # df["Year"].value Glicitation of the strategy - # df["Year"].max()	e_counts()
2020.0 df["Year"].min() 1980.0 plt.hist(df["Year"]) plt.show()	
3000 - 2000 - 1000 - 1980 1985 1990 n, bins, var1=plt.h:	1995 2000 2005 2010 2015 2020 List(df["Year"])
3]: n	1995 2000 2005 2010 2015 2020 ., 92., 449., 1274., 2440., 3921., 5262., 2406.,
bins array([1980., 1984 2016., 2020 plt.hist(df["Year" plt.show()) 8000 - 6000 -	
plt.hist(df["Year" plt.show()	1995 2000 2005 2010 2015 2020 1], bins=[1980, 2000, 2005, 2012, 2015, 2020])
<pre># plt.hist? sns.histplot(df["\])</pre>	1995 2000 2005 2010 2015 2020 //ear"], bins=4)
plt.show() 8000 - 6000 - 2000 - 1980 1985 199	0 1995 2000 2005 2010 2015 2020
plt.hist(df["Year" plt.show()	Year
	1995 2000 2005 2010 2015 2020 (ear"], bins=10)
plt.hist(df["Year" plt.show()	Year
	1995 2000 2005 2010 2015 2020 Year"], shade=True)
0.06 - 2 0.05 - 0.04 - 0.03 - 0.02 - 0.01 - 0.00 1980 19	2000 2010 2020 Year "], kde=True, bins=10)
5000 - 4000 - 1000 -	
df.columns Index(['Rank', 'Name' 'EU_Sales', dtype='object') plt.figure(figsizes sns.boxplot(data=0)	e=(8,5))
plt.figure(figsize sns.boxplot(x=df[plt.show()	10 15 20 25 30 De=(12,8)) Global_Sales'])

plt.show()	10 15 20 25 30 De=(8,5)) If, Dal_Sales')
plt.figure(figsizes sns.violinplot(dat y='Glob plt.show()	e=(8,5)) ca=df, pai_Sales')
plt.show() 30 - 25 - 20 - 30 - 10 - 5 -	
<pre>df.describe()</pre>	Year NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales 6381.000000 16652.000000 16652.000000 16652.000000 16652.000000 16652.000000 2006.390513 2.752314 1.996875 2.499677 1.151829 8.457873 5.863261 1.327002 1.322972 1.164023 1.054813 3.717756 1080.000000 0.140000 0.010000 0.010000 0.0474276 0.340000
mean 8283.409620 std 4794.471477 min 1.000000 25% 4129.750000	5.863261 1.327002 1.322972 1.164023 1.054813 3.717756 1980.000000 0.140000 0.010000 0.000000 -0.474276 0.240000 2003.000000 1.781124 1.087977 1.781124 0.394830 5.580341 2007.000000 2.697415 1.714664 2.480356 0.491870 7.536614
count 16652.000000 1 mean 8283.409620 std 4794.471477 min 1.000000 25% 4129.750000	1980.000000 0.140000 0.010000 0.000000 -0.474276 0.240000 2003.000000 1.781124 1.087977 1.781124 0.394830 5.580341 2007.000000 2.697415 1.714664 2.480356 0.491870 7.536614 2010.000000 3.677290 2.795123 3.176299 1.781124 11.227334