Ejercicio Exploratory Data Analysis – Project 1

Exploratory Data Analysis (EDA): Uso de librería pandas y otras librerías

In [ ]:
```
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt

os.chdir('E:\WORK IN PROGRESS\Data Analytics course\parte 2 python\week 19')
```

Leer archivo de venta de videojuegos vgsales.csv.

Lectura del archivo

In [ ]:
```
df = pd.read_csv('vgsales.csv')
```

head, tail, sample, describe

In [ ]:
```
df.head(5)
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	

In [ ]:
```
df.tail(5)
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.0
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.0
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.0
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.0
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.0

In [ ]:
```
df.sample(5)
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	C
12504	12506	Top Gear Hyper-Bike	N64	1999.0	Racing	Kemco	0.05	0.01	0.00	
5054	5056	NCAA Final Four 2000	PS	1999.0	Sports	989 Sports	0.21	0.14	0.00	
6176	6178	Mission: Impossible - Operation Surma	PS2	2003.0	Platform	Atari	0.14	0.11	0.00	
1854	1856	Petz Dogz 2	DS	2007.0	Simulation	Ubisoft	0.47	0.52	0.00	
3844	3846	Mega Man 6	NES	1993.0	Platform	Capcom	0.28	0.07	0.16	

In [ ]:
```
pd.options.display.float_format='{:.2f}'.format
```

In [ ]:
```
df.describe()
```

Out[ ]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	16598.00	16327.00	16598.00	16598.00	16598.00	16598.00	16598.00
mean	8300.61	2006.41	0.26	0.15	0.08	0.05	0.54
std	4791.85	5.83	0.82	0.51	0.31	0.19	1.56
min	1.00	1980.00	0.00	0.00	0.00	0.00	0.01
25%	4151.25	2003.00	0.00	0.00	0.00	0.00	0.06
50%	8300.50	2007.00	0.08	0.02	0.00	0.01	0.17
75%	12449.75	2010.00	0.24	0.11	0.04	0.04	0.47
max	16600.00	2020.00	41.49	29.02	10.22	10.57	82.74

Validación de columnas numéricas para que el describe funcione con todas.

In [ ]:
```
df.info()

# En la siguiente tabla se puede observar que hay cuatro columnas
# que no son numericas (Name, Platform, Genre, Publisher). Como los
# valores de estas variables son tipo capdena (por la informacion
# que expresan) no se pueden cambiar a tipo numerico. Por lo cual,
# se dejan asi.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Rank          16598 non-null  int64  
 1   Name          16598 non-null  object 
 2   Platform      16598 non-null  object 
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object 
 5   Publisher     16540 non-null  object 
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

Totales

In [ ]:
```
df.shape
```

Out[ ]:
```
(16598, 11)
```

1) Total de videojuegos analizados

In [ ]:
```
# Total de video juegos analizados
total_names = df['Name'].unique().__len__()
total_names
```

```
# El rango de filas del dataset es de 16598. Como el total de video juegos
# analizados es 11493 quiere decir que algunos juegos se vendieron en mas
# de 1 ano.
```

Out[ ]: 11493

2) total de géneros analizados

In [ ]:
```
# Total de generos analizados
total_genre = df['Genre'].unique().__len__()
total_genre

# Existen solo 12 tipos de generos diferentes de video juegos.
```

Out[ ]: 12

3) Plataformas analizadas

In [ ]:
```
# Total de Plataformas analizadas
total_platform = df['Platform'].unique().__len__()
total_platform

# Han existido solo 31 tipo de plataformas desde el
# inicio de los videojuegos.
```

Out[ ]: 31

Ventas

1) Total de ventas por año

1.1 Global

In [ ]:
```
df_salesxyears_global=df.groupby('Year')['Global_Sales'].sum().rename('Total_Sales
df_salesxyears_global.sort_values(ascending=False, by='Year')
```

Out[ ]:

	Year	Total_Sales
38	2020	0
37	2017	0
36	2016	71
35	2015	264
34	2014	337
33	2013	368
32	2012	364
31	2011	516
30	2010	600
29	2009	667
28	2008	679
27	2007	611
26	2006	521
25	2005	460
24	2004	419
23	2003	358
22	2002	396
21	2001	331
20	2000	202
19	1999	251
18	1998	256
17	1997	201
16	1996	199
15	1995	88
14	1994	79
13	1993	46
12	1992	76
11	1991	32
10	1990	49
9	1989	73
8	1988	47
7	1987	22
6	1986	37
5	1985	54
4	1984	50
3	1983	17

	Year	Total_Sales
2	1982	29
1	1981	36
0	1980	11

1.2 North America

In [ ]:
```
df_salesxyears_na=df.groupby('Year')['NA_Sales'].sum().rename('NA_Sales').reset_in
df_salesxyears_na.sort_values(ascending=False, by='Year')
```

Out[ ]:

	Year	NA_Sales
38	2020	0
37	2017	0
36	2016	23
35	2015	103
34	2014	132
33	2013	155
32	2012	155
31	2011	241
30	2010	304
29	2009	339
28	2008	351
27	2007	312
26	2006	263
25	2005	243
24	2004	223
23	2003	194
22	2002	216
21	2001	174
20	2000	94
19	1999	126
18	1998	128
17	1997	95
16	1996	87
15	1995	25
14	1994	28
13	1993	15
12	1992	34
11	1991	13
10	1990	25
9	1989	45
8	1988	24
7	1987	8
6	1986	12
5	1985	34
4	1984	33
3	1983	8

	Year	NA_Sales
2	1982	27
1	1981	33
0	1980	11

1.3 European Union

In [ ]:
```
df_salesxyears_eu=df.groupby('Year')['EU_Sales'].sum().rename('EU_Sales').reset_in
df_salesxyears_eu.sort_values(ascending=False, by='Year')
```

Out[ ]:

	Year	EU_Sales
38	2020	0
37	2017	0
36	2016	27
35	2015	98
34	2014	126
33	2013	126
32	2012	119
31	2011	167
30	2010	177
29	2009	192
28	2008	184
27	2007	160
26	2006	129
25	2005	122
24	2004	107
23	2003	104
22	2002	110
21	2001	95
20	2000	53
19	1999	63
18	1998	67
17	1997	48
16	1996	47
15	1995	15
14	1994	15
13	1993	5
12	1992	12
11	1991	4
10	1990	8
9	1989	8
8	1988	7
7	1987	1
6	1986	3
5	1985	5
4	1984	2
3	1983	1

	Year	EU_Sales
2	1982	2
1	1981	2
0	1980	1

1.4 Japan

In [ ]:
```
df_salesxyears_jp=df.groupby('Year')['JP_Sales'].sum().rename('JP_Sales').reset_in
df_salesxyears_jp.sort_values(ascending=False, by='Year')
```

Out[ ]:

	Year	JP_Sales
38	2020	0
37	2017	0
36	2016	14
35	2015	34
34	2014	39
33	2013	48
32	2012	52
31	2011	53
30	2010	59
29	2009	62
28	2008	60
27	2007	60
26	2006	74
25	2005	54
24	2004	42
23	2003	34
22	2002	42
21	2001	40
20	2000	43
19	1999	52
18	1998	50
17	1997	49
16	1996	57
15	1995	46
14	1994	34
13	1993	25
12	1992	29
11	1991	15
10	1990	15
9	1989	18
8	1988	16
7	1987	12
6	1986	20
5	1985	15
4	1984	14
3	1983	8

	Year	JP_Sales
2	1982	0
1	1981	0
0	1980	0

1.5 Others

In [ ]:
```
df_salesxyears_os=df.groupby('Year')['Other_Sales'].sum().rename('Other_Sales').re
df_salesxyears_os.sort_values(ascending=False, by='Year')
```

Out[ ]:

	Year	Other_Sales
38	2020	0
37	2017	0
36	2016	8
35	2015	30
34	2014	40
33	2013	40
32	2012	38
31	2011	54
30	2010	60
29	2009	75
28	2008	82
27	2007	78
26	2006	54
25	2005	41
24	2004	47
23	2003	26
22	2002	27
21	2001	23
20	2000	12
19	1999	10
18	1998	11
17	1997	9
16	1996	8
15	1995	3
14	1994	2
13	1993	1
12	1992	2
11	1991	1
10	1990	1
9	1989	2
8	1988	1
7	1987	0
6	1986	2
5	1985	1
4	1984	1
3	1983	0

	Year	Other_Sales
2	1982	0
1	1981	0
0	1980	0

In [ ]:
```
# Poniendo todas las tablas juntas

result = df_salesxyears_global.merge(df_salesxyears_na,on='Year').merge(df_salesxy
result
```

Out[ ]:

	Year	Total_Sales	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1980	11	11	1	0	0
1	1981	36	33	2	0	0
2	1982	29	27	2	0	0
3	1983	17	8	1	8	0
4	1984	50	33	2	14	1
5	1985	54	34	5	15	1
6	1986	37	12	3	20	2
7	1987	22	8	1	12	0
8	1988	47	24	7	16	1
9	1989	73	45	8	18	2
10	1990	49	25	8	15	1
11	1991	32	13	4	15	1
12	1992	76	34	12	29	2
13	1993	46	15	5	25	1
14	1994	79	28	15	34	2
15	1995	88	25	15	46	3
16	1996	199	87	47	57	8
17	1997	201	95	48	49	9
18	1998	256	128	67	50	11
19	1999	251	126	63	52	10
20	2000	202	94	53	43	12
21	2001	331	174	95	40	23
22	2002	396	216	110	42	27
23	2003	358	194	104	34	26
24	2004	419	223	107	42	47
25	2005	460	243	122	54	41
26	2006	521	263	129	74	54
27	2007	611	312	160	60	78
28	2008	679	351	184	60	82
29	2009	667	339	192	62	75
30	2010	600	304	177	59	60
31	2011	516	241	167	53	54
32	2012	364	155	119	52	38
33	2013	368	155	126	48	40
34	2014	337	132	126	39	40
35	2015	264	103	98	34	30

	Year	Total_Sales	NA_Sales	EU_Sales	JP_Sales	Other_Sales
36	2016	71	23	27	14	8
37	2017	0	0	0	0	0
38	2020	0	0	0	0	0

2) Juegos más vendidos por año

2.1 Global

In [ ]:
```
#Global_sales row: 8

years=list(df['Year'].sort_values().unique())
gamesxyear=[]

for year in years:
    new_df=df.loc[df['Year']==year]
    game=new_df['Name'].loc[new_df['Global_Sales']==new_df['Global_Sales'].max()].
    gamesxyear.append(game)

df_gamesxyear=pd.DataFrame(gamesxyear)
df_gamesxyear.rename(columns={0:'Games_Global'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_global=pd.concat([df_years,df_gamesxyear],axis=1)
df_answer_global.drop(39,inplace=True)
df_answer_global
```

Out[ ]:

	Years	Games_Global
0	1980	Asteroids
1	1981	Pitfall!
2	1982	Pac-Man
3	1983	Baseball
4	1984	Duck Hunt
5	1985	Super Mario Bros.
6	1986	The Legend of Zelda
7	1987	Zelda II: The Adventure of Link
8	1988	Super Mario Bros. 3
9	1989	Tetris
10	1990	Super Mario World
11	1991	The Legend of Zelda: A Link to the Past
12	1992	Super Mario Land 2: 6 Golden Coins
13	1993	Super Mario All-Stars
14	1994	Donkey Kong Country
15	1995	Donkey Kong Country 2: Diddy's Kong Quest
16	1996	Pokemon Red/Pokemon Blue
17	1997	Gran Turismo
18	1998	Pokémon Yellow: Special Pikachu Edition
19	1999	Pokemon Gold/Pokemon Silver
20	2000	Pokémon Crystal Version
21	2001	Gran Turismo 3: A-Spec
22	2002	Grand Theft Auto: Vice City
23	2003	Need for Speed Underground
24	2004	Grand Theft Auto: San Andreas
25	2005	Nintendogs
26	2006	Wii Sports
27	2007	Wii Fit
28	2008	Mario Kart Wii
29	2009	Wii Sports Resort
30	2010	Kinect Adventures!
31	2011	Call of Duty: Modern Warfare 3
32	2012	Call of Duty: Black Ops II
33	2013	Grand Theft Auto V
34	2014	Grand Theft Auto V
35	2015	Call of Duty: Black Ops 3

	Years	Games_Global
36	2016	FIFA 17
37	2017	Phantasy Star Online 2 Episode 4: Deluxe Package
38	2020	Imagine: Makeup Artist

2.2 North America

In [ ]:
```
#NA_sales row: 8

years=list(df['Year'].sort_values().unique())
gamesxyear=[]

for year in years:
    new_df=df.loc[df['Year']==year]
    game=new_df['Name'].loc[new_df['NA_Sales']==new_df['NA_Sales'].max()].values.t
    gamesxyear.append(game)

df_gamesxyear=pd.DataFrame(gamesxyear)
df_gamesxyear.drop([1,2],axis=1,inplace=True)
df_gamesxyear.rename(columns={0:'Games_NA'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_na=pd.concat([df_years,df_gamesxyear],axis=1)
df_answer_na.drop(39,inplace=True)
df_answer_na
```

Out[ ]:

	Years	Games_NA
0	1980	Asteroids
1	1981	Pitfall!
2	1982	Pac-Man
3	1983	Pitfall II: Lost Caverns
4	1984	Duck Hunt
5	1985	Super Mario Bros.
6	1986	The Legend of Zelda
7	1987	Zelda II: The Adventure of Link
8	1988	Super Mario Bros. 3
9	1989	Tetris
10	1990	Super Mario World
11	1991	Sonic the Hedgehog
12	1992	Super Mario Land 2: 6 Golden Coins
13	1993	Super Mario All-Stars
14	1994	Donkey Kong Country
15	1995	Killer Instinct
16	1996	Pokemon Red/Pokemon Blue
17	1997	GoldenEye 007
18	1998	Pokémon Yellow: Special Pikachu Edition
19	1999	Pokemon Gold/Pokemon Silver
20	2000	Tony Hawk's Pro Skater 2
21	2001	Grand Theft Auto III
22	2002	Grand Theft Auto: Vice City
23	2003	Mario Kart: Double Dash!!
24	2004	Grand Theft Auto: San Andreas
25	2005	Mario Kart DS
26	2006	Wii Sports
27	2007	Wii Fit
28	2008	Mario Kart Wii
29	2009	Wii Sports Resort
30	2010	Kinect Adventures!
31	2011	Call of Duty: Modern Warfare 3
32	2012	Call of Duty: Black Ops II
33	2013	Grand Theft Auto V
34	2014	Pokemon Omega Ruby/Pokemon Alpha Sapphire
35	2015	Call of Duty: Black Ops 3

	Years	Games_NA
36	2016	Uncharted 4: A Thief's End
37	2017	Phantasy Star Online 2 Episode 4: Deluxe Package
38	2020	Imagine: Makeup Artist

2.3 European Union

In [ ]:
```
#EU_sales row: 8

years=list(df['Year'].sort_values().unique())
gamesxyear=[]

for year in years:
    new_df=df.loc[df['Year']==year]
    game=new_df['Name'].loc[new_df['EU_Sales']==new_df['EU_Sales'].max()].values.t
    gamesxyear.append(game)

df_gamesxyear=pd.DataFrame(gamesxyear)
df_gamesxyear.drop([1,2],axis=1,inplace=True)
df_gamesxyear.rename(columns={0:'Games_EU'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_eu=pd.concat([df_years,df_gamesxyear],axis=1)
df_answer_eu.drop(39,inplace=True)
df_answer_eu
```

Out[ ]:

	Years	Games_EU
0	1980	Asteroids
1	1981	Pitfall!
2	1982	Pac-Man
3	1983	Mario Bros.
4	1984	Duck Hunt
5	1985	Super Mario Bros.
6	1986	The Legend of Zelda
7	1987	Zelda II: The Adventure of Link
8	1988	Super Mario Bros. 3
9	1989	Super Mario Land
10	1990	Super Mario World
11	1991	The Legend of Zelda: A Link to the Past
12	1992	Super Mario Land 2: 6 Golden Coins
13	1993	Super Mario All-Stars
14	1994	Myst
15	1995	Warcraft II: Tides of Darkness
16	1996	Pokemon Red/Pokemon Blue
17	1997	Gran Turismo
18	1998	Pokémon Yellow: Special Pikachu Edition
19	1999	Pokemon Gold/Pokemon Silver
20	2000	Driver 2
21	2001	Gran Turismo 3: A-Spec
22	2002	Grand Theft Auto: Vice City
23	2003	Need for Speed Underground
24	2004	World of Warcraft
25	2005	Nintendogs
26	2006	Wii Sports
27	2007	Wii Fit
28	2008	Mario Kart Wii
29	2009	Wii Sports Resort
30	2010	Kinect Adventures!
31	2011	Call of Duty: Modern Warfare 3
32	2012	Call of Duty: Black Ops II
33	2013	Grand Theft Auto V
34	2014	Grand Theft Auto V
35	2015	FIFA 16

	Years	Games_EU
36	2016	FIFA 17
37	2017	Phantasy Star Online 2 Episode 4: Deluxe Package
38	2020	Imagine: Makeup Artist

2.4 Japan

In [ ]:
```
#JP_sales row: 8

years=list(df['Year'].sort_values().unique())
gamesxyear=[]

for year in years:
    new_df=df.loc[df['Year']==year]
    game=new_df['Name'].loc[new_df['JP_Sales']==new_df['JP_Sales'].max()].values.t
    gamesxyear.append(game)

df_gamesxyear=pd.DataFrame(gamesxyear)
df_gamesxyear.drop([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24
df_gamesxyear.rename(columns={0:'Games_JP'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_jp=pd.concat([df_years,df_gamesxyear],axis=1)
df_answer_jp.drop(39,inplace=True)
df_answer_jp
```

Out[ ]:

	Years	Games_JP
0	1980	Asteroids
1	1981	Pitfall!
2	1982	Pac-Man
3	1983	Baseball
4	1984	Golf
5	1985	Super Mario Bros.
6	1986	Super Mario Bros.: The Lost Levels
7	1987	Dragon Warrior II
8	1988	Super Mario Bros. 3
9	1989	Tetris
10	1990	Super Mario World
11	1991	Final Fantasy II
12	1992	Super Mario Kart
13	1993	Super Mario All-Stars
14	1994	Donkey Kong Country
15	1995	Dragon Quest VI: Maboroshi no Daichi
16	1996	Pokemon Red/Pokemon Blue
17	1997	Final Fantasy VII
18	1998	Pokémon Yellow: Special Pikachu Edition
19	1999	Pokemon Gold/Pokemon Silver
20	2000	Dragon Quest VII: Warriors of Eden
21	2001	Final Fantasy X
22	2002	Pokemon Ruby/Pokemon Sapphire
23	2003	Final Fantasy X-2
24	2004	Dragon Quest VIII: Journey of the Cursed King
25	2005	Animal Crossing: Wild World
26	2006	New Super Mario Bros.
27	2007	Wii Fit
28	2008	Monster Hunter Freedom Unite
29	2009	New Super Mario Bros. Wii
30	2010	Pokemon Black/Pokemon White
31	2011	Mario Kart 7
32	2012	Animal Crossing: New Leaf
33	2013	Pokemon X/Pokemon Y
34	2014	Yokai Watch 2 Ganso/Honke
35	2015	Monster Hunter X

	Years	Games_JP
36	2016	Yokai Watch 3
37	2017	Phantasy Star Online 2 Episode 4: Deluxe Package
38	2020	Imagine: Makeup Artist

2.5 Others

In [ ]:
```
#Other_sales row: 8

years=list(df['Year'].sort_values().unique())
gamesxyear=[]

for year in years:
    new_df=df.loc[df['Year']==year]
    game=new_df['Name'].loc[new_df['Other_Sales']==new_df['Other_Sales'].max()].va
    gamesxyear.append(game)

df_gamesxyear=pd.DataFrame(gamesxyear)
df_gamesxyear.drop([1,2,3],axis=1,inplace=True)
df_gamesxyear.rename(columns={0:'Games_Other'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_os=pd.concat([df_years,df_gamesxyear],axis=1)
df_answer_os.drop(39,inplace=True)
df_answer_os
```

Out[ ]:

	Years	Games_Other
0	1980	Asteroids
1	1981	Pitfall!
2	1982	Pac-Man
3	1983	Baseball
4	1984	Duck Hunt
5	1985	Super Mario Bros.
6	1986	Dragon Warrior
7	1987	Zelda II: The Adventure of Link
8	1988	Super Mario Bros. 3
9	1989	Tetris
10	1990	Super Mario World
11	1991	The Legend of Zelda: A Link to the Past
12	1992	Super Mario Land 2: 6 Golden Coins
13	1993	Super Mario All-Stars
14	1994	Donkey Kong
15	1995	Warcraft II: Tides of Darkness
16	1996	Pokemon Red/Pokemon Blue
17	1997	Final Fantasy VII
18	1998	Pokémon Yellow: Special Pikachu Edition
19	1999	Pokemon Gold/Pokemon Silver
20	2000	Pokémon Crystal Version
21	2001	Grand Theft Auto III
22	2002	Grand Theft Auto: Vice City
23	2003	Need for Speed Underground
24	2004	Grand Theft Auto: San Andreas
25	2005	Nintendogs
26	2006	Wii Sports
27	2007	Pro Evolution Soccer 2008
28	2008	Mario Kart Wii
29	2009	Wii Sports Resort
30	2010	Gran Turismo 5
31	2011	Call of Duty: Modern Warfare 3
32	2012	Call of Duty: Black Ops II
33	2013	Grand Theft Auto V
34	2014	Grand Theft Auto V
35	2015	Call of Duty: Black Ops 3

	Years	Games_Other
36	2016	FIFA 17
37	2017	Phantasy Star Online 2 Episode 4: Deluxe Package
38	2020	Imagine: Makeup Artist

In [ ]:
```
# Poniendo todas las tablas juntas

result = df_answer_global.merge(df_answer_na,on='Years').merge(df_answer_eu,on='Ye
result
```

Out[ ]:

	Years	Games_Global	Games_NA	Games_EU	Games_JP	Games_Other
0	1980	Asteroids	Asteroids	Asteroids	Asteroids	Asteroids
1	1981	Pitfall!	Pitfall!	Pitfall!	Pitfall!	Pitfall!
2	1982	Pac-Man	Pac-Man	Pac-Man	Pac-Man	Pac-Man
3	1983	Baseball	Pitfall II: Lost Caverns	Mario Bros.	Baseball	Baseball
4	1984	Duck Hunt	Duck Hunt	Duck Hunt	Golf	Duck Hunt
5	1985	Super Mario Bros.	Super Mario Bros.	Super Mario Bros.	Super Mario Bros.	Super Mario Bros.
6	1986	The Legend of Zelda	The Legend of Zelda	The Legend of Zelda	Super Mario Bros.: The Lost Levels	Dragon Warrior
7	1987	Zelda II: The Adventure of Link	Zelda II: The Adventure of Link	Zelda II: The Adventure of Link	Dragon Warrior II	Zelda II: The Adventure of Link
8	1988	Super Mario Bros. 3	Super Mario Bros. 3	Super Mario Bros. 3	Super Mario Bros. 3	Super Mario Bros. 3
9	1989	Tetris	Tetris	Super Mario Land	Tetris	Tetris
10	1990	Super Mario World	Super Mario World	Super Mario World	Super Mario World	Super Mario World
11	1991	The Legend of Zelda: A Link to the Past	Sonic the Hedgehog	The Legend of Zelda: A Link to the Past	Final Fantasy II	The Legend of Zelda: A Link to the Past
12	1992	Super Mario Land 2: 6 Golden Coins	Super Mario Land 2: 6 Golden Coins	Super Mario Land 2: 6 Golden Coins	Super Mario Kart	Super Mario Land 2: 6 Golden Coins
13	1993	Super Mario All-Stars	Super Mario All-Stars	Super Mario All-Stars	Super Mario All-Stars	Super Mario All-Stars
14	1994	Donkey Kong Country	Donkey Kong Country	Myst	Donkey Kong Country	Donkey Kong
15	1995	Donkey Kong Country 2: Diddy's Kong Quest	Killer Instinct	Warcraft II: Tides of Darkness	Dragon Quest VI: Maboroshi no Daichi	Warcraft II: Tides of Darkness
16	1996	Pokemon Red/Pokemon Blue	Pokemon Red/Pokemon Blue	Pokemon Red/Pokemon Blue	Pokemon Red/Pokemon Blue	Pokemon Red/Pokemon Blue
17	1997	Gran Turismo	GoldenEye 007	Gran Turismo	Final Fantasy VII	Final Fantasy VII
18	1998	Pokémon Yellow: Special Pikachu Edition	Pokémon Yellow: Special Pikachu Edition	Pokémon Yellow: Special Pikachu Edition	Pokémon Yellow: Special Pikachu Edition	Pokémon Yellow: Special Pikachu Edition
19	1999	Pokemon Gold/Pokemon Silver	Pokemon Gold/Pokemon Silver	Pokemon Gold/Pokemon Silver	Pokemon Gold/Pokemon Silver	Pokemon Gold/Pokemon Silver
20	2000	Pokémon Crystal Version	Tony Hawk's Pro Skater 2	Driver 2	Dragon Quest VII: Warriors of Eden	Pokémon Crystal Version

	Years	Games_Global	Games_NA	Games_EU	Games_JP	Games_Other
21	2001	Gran Turismo 3: A-Spec	Grand Theft Auto III	Gran Turismo 3: A-Spec	Final Fantasy X	Grand Theft Auto III
22	2002	Grand Theft Auto: Vice City	Grand Theft Auto: Vice City	Grand Theft Auto: Vice City	Pokemon Ruby/Pokemon Sapphire	Grand Theft Auto: Vice City
23	2003	Need for Speed Underground	Mario Kart: Double Dash!!	Need for Speed Underground	Final Fantasy X-2	Need for Speed Underground
24	2004	Grand Theft Auto: San Andreas	Grand Theft Auto: San Andreas	World of Warcraft	Dragon Quest VIII: Journey of the Cursed King	Grand Theft Auto: San Andreas
25	2005	Nintendogs	Mario Kart DS	Nintendogs	Animal Crossing: Wild World	Nintendogs
26	2006	Wii Sports	Wii Sports	Wii Sports	New Super Mario Bros.	Wii Sports
27	2007	Wii Fit	Wii Fit	Wii Fit	Wii Fit	Pro Evolution Soccer 2008
28	2008	Mario Kart Wii	Mario Kart Wii	Mario Kart Wii	Monster Hunter Freedom Unite	Mario Kart Wii
29	2009	Wii Sports Resort	Wii Sports Resort	Wii Sports Resort	New Super Mario Bros. Wii	Wii Sports Resort
30	2010	Kinect Adventures!	Kinect Adventures!	Kinect Adventures!	Pokemon Black/Pokemon White	Gran Turismo 5
31	2011	Call of Duty: Modern Warfare 3	Call of Duty: Modern Warfare 3	Call of Duty: Modern Warfare 3	Mario Kart 7	Call of Duty: Modern Warfare 3
32	2012	Call of Duty: Black Ops II	Call of Duty: Black Ops II	Call of Duty: Black Ops II	Animal Crossing: New Leaf	Call of Duty: Black Ops II
33	2013	Grand Theft Auto V	Grand Theft Auto V	Grand Theft Auto V	Pokemon X/Pokemon Y	Grand Theft Auto V
34	2014	Grand Theft Auto V	Pokemon Omega Ruby/Pokemon Alpha Sapphire	Grand Theft Auto V	Yokai Watch 2 Ganso/Honke	Grand Theft Auto V
35	2015	Call of Duty: Black Ops 3	Call of Duty: Black Ops 3	FIFA 16	Monster Hunter X	Call of Duty: Black Ops 3
36	2016	FIFA 17	Uncharted 4: A Thief's End	FIFA 17	Yokai Watch 3	FIFA 17
37	2017	Phantasy Star Online 2 Episode 4: Deluxe Package	Phantasy Star Online 2 Episode 4: Deluxe Package	Phantasy Star Online 2 Episode 4: Deluxe Package	Phantasy Star Online 2 Episode 4: Deluxe Package	Phantasy Star Online 2 Episode 4: Deluxe Package
38	2020	Imagine: Makeup Artist	Imagine: Makeup Artist	Imagine: Makeup Artist	Imagine: Makeup Artist	Imagine: Makeup Artist

3) Plataformas con más ingresos por año

3.1 Global

In [ ]:
```
#Global_sales row: 8

years=list(df['Year'].sort_values().unique())
platformsxyear=[]

for year in years:
    df_helper=df.loc[df['Year']==year]
    df_platforms=df_helper['Platform'].loc[df_helper['Global_Sales']==df_helper['G
    platformsxyear.append(df_platforms)

df_platforms=pd.DataFrame(platformsxyear)
df_platforms.rename(columns={0:'Platfroms_Global'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_global=pd.concat([df_years,df_platforms],axis=1)
df_answer_global.drop(39,inplace=True)

df_answer_global
```

Out[ ]:

	Years	Platfroms_Global
0	1980	2600
1	1981	2600
2	1982	2600
3	1983	NES
4	1984	NES
5	1985	NES
6	1986	NES
7	1987	NES
8	1988	NES
9	1989	GB
10	1990	SNES
11	1991	SNES
12	1992	GB
13	1993	SNES
14	1994	SNES
15	1995	SNES
16	1996	GB
17	1997	PS
18	1998	GB
19	1999	GB
20	2000	GB
21	2001	PS2
22	2002	PS2
23	2003	PS2
24	2004	PS2
25	2005	DS
26	2006	Wii
27	2007	Wii
28	2008	Wii
29	2009	Wii
30	2010	X360
31	2011	X360
32	2012	PS3
33	2013	PS3
34	2014	PS4
35	2015	PS4

	Years	Platfroms_Global
36	2016	PS4
37	2017	PS4
38	2020	DS

3.2 North America

In [ ]:
```
#NA_sales row: 8

years=list(df['Year'].sort_values().unique())
platformsxyear=[]

for year in years:
    df_helper=df.loc[df['Year']==year]
    df_platforms=df_helper['Platform'].loc[df_helper['NA_Sales']==df_helper['NA_Sa
    platformsxyear.append(df_platforms)

df_platforms=pd.DataFrame(platformsxyear)
df_platforms.drop([1,2],axis=1,inplace=True)
df_platforms.rename(columns={0:'Platfroms_NA'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_na=pd.concat([df_years,df_platforms],axis=1)
df_answer_na.drop(39,inplace=True)

df_answer_na
```

Out[ ]:

	Years	Platfroms_NA
0	1980	2600
1	1981	2600
2	1982	2600
3	1983	2600
4	1984	NES
5	1985	NES
6	1986	NES
7	1987	NES
8	1988	NES
9	1989	GB
10	1990	SNES
11	1991	GEN
12	1992	GB
13	1993	SNES
14	1994	SNES
15	1995	SNES
16	1996	GB
17	1997	N64
18	1998	GB
19	1999	GB
20	2000	PS
21	2001	PS2
22	2002	PS2
23	2003	GC
24	2004	PS2
25	2005	DS
26	2006	Wii
27	2007	Wii
28	2008	Wii
29	2009	Wii
30	2010	X360
31	2011	X360
32	2012	X360
33	2013	X360
34	2014	3DS
35	2015	PS4

	Years	Platfroms_NA
36	2016	PS4
37	2017	PS4
38	2020	DS

3.3 European Union

In [ ]:
```
#EU_sales row: 8

years=list(df['Year'].sort_values().unique())
platformsxyear=[]

for year in years:
    df_helper=df.loc[df['Year']==year]
    df_platforms=df_helper['Platform'].loc[df_helper['EU_Sales']==df_helper['EU_Sa
    platformsxyear.append(df_platforms)

df_platforms=pd.DataFrame(platformsxyear)
df_platforms.drop([1,2],axis=1,inplace=True)
df_platforms.rename(columns={0:'Platfroms_EU'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_eu=pd.concat([df_years,df_platforms],axis=1)
df_answer_eu.drop(39,inplace=True)

df_answer_eu
```

Out[ ]:

	Years	Platfroms_EU
0	1980	2600
1	1981	2600
2	1982	2600
3	1983	NES
4	1984	NES
5	1985	NES
6	1986	NES
7	1987	NES
8	1988	NES
9	1989	GB
10	1990	SNES
11	1991	SNES
12	1992	GB
13	1993	SNES
14	1994	PC
15	1995	PC
16	1996	GB
17	1997	PS
18	1998	GB
19	1999	GB
20	2000	PS
21	2001	PS2
22	2002	PS2
23	2003	PS2
24	2004	PC
25	2005	DS
26	2006	Wii
27	2007	Wii
28	2008	Wii
29	2009	Wii
30	2010	X360
31	2011	PS3
32	2012	PS3
33	2013	PS3
34	2014	PS4
35	2015	PS4

	Years	Platfroms_EU
36	2016	PS4
37	2017	PS4
38	2020	DS

3.4 Japan

In [ ]:
```
#JP_sales row: 8

years=list(df['Year'].sort_values().unique())
platformsxyear=[]

for year in years:
    df_helper=df.loc[df['Year']==year]
    df_platforms=df_helper['Platform'].loc[df_helper['JP_Sales']==df_helper['JP_Sa
    platformsxyear.append(df_platforms)

df_platforms=pd.DataFrame(platformsxyear)
df_platforms.drop([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
df_platforms.rename(columns={0:'Platfroms_JP'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_jp=pd.concat([df_years,df_platforms],axis=1)
df_answer_jp.drop(39,inplace=True)

df_answer_jp
```

Out[ ]:

	Years	Platfroms_JP
0	1980	2600
1	1981	2600
2	1982	2600
3	1983	NES
4	1984	NES
5	1985	NES
6	1986	NES
7	1987	NES
8	1988	NES
9	1989	GB
10	1990	SNES
11	1991	SNES
12	1992	SNES
13	1993	SNES
14	1994	SNES
15	1995	SNES
16	1996	GB
17	1997	PS
18	1998	GB
19	1999	GB
20	2000	PS
21	2001	PS2
22	2002	GBA
23	2003	PS2
24	2004	PS2
25	2005	DS
26	2006	DS
27	2007	Wii
28	2008	PSP
29	2009	Wii
30	2010	DS
31	2011	3DS
32	2012	3DS
33	2013	3DS
34	2014	3DS
35	2015	3DS

	Years	Platfroms_JP
36	2016	3DS
37	2017	PS4
38	2020	DS

4.5 Other

In [ ]:
```
#Other_sales row: 8

years=list(df['Year'].sort_values().unique())
platformsxyear=[]

for year in years:
    df_helper=df.loc[df['Year']==year]
    df_platforms=df_helper['Platform'].loc[df_helper['Other_Sales']==df_helper['Ot
    platformsxyear.append(df_platforms)

df_platforms=pd.DataFrame(platformsxyear)
df_platforms.drop([1,2,3],axis=1,inplace=True)
df_platforms.rename(columns={0:'Platfroms_Other'},inplace=True)
df_years=pd.DataFrame(years)
df_years.rename(columns={0:'Years'},inplace=True)
pd.options.display.float_format='{:.0f}'.format
df_answer_os=pd.concat([df_years,df_platforms],axis=1)
df_answer_os.drop(39,inplace=True)

df_answer_os
```

Out[ ]:

	Years	Platfroms_Other
0	1980	2600
1	1981	2600
2	1982	2600
3	1983	NES
4	1984	NES
5	1985	NES
6	1986	NES
7	1987	NES
8	1988	NES
9	1989	GB
10	1990	SNES
11	1991	SNES
12	1992	GB
13	1993	SNES
14	1994	GB
15	1995	PC
16	1996	GB
17	1997	PS
18	1998	GB
19	1999	GB
20	2000	GB
21	2001	PS2
22	2002	PS2
23	2003	PS2
24	2004	PS2
25	2005	DS
26	2006	Wii
27	2007	PS2
28	2008	Wii
29	2009	Wii
30	2010	PS3
31	2011	PS3
32	2012	PS3
33	2013	PS3
34	2014	PS4
35	2015	PS4

	Years	Platfroms_Other
36	2016	PS4
37	2017	PS4
38	2020	DS

In [ ]:
```
# Poniendo todas las tablas juntas

result = df_answer_global.merge(df_answer_na,on='Years').merge(df_answer_eu,on='Ye
result
```

Out[ ]:

	Years	Platfroms_Global	Platfroms_NA	Platfroms_EU	Platfroms_JP	Platfroms_Other
0	1980	2600	2600	2600	2600	2600
1	1981	2600	2600	2600	2600	2600
2	1982	2600	2600	2600	2600	2600
3	1983	NES	2600	NES	NES	NES
4	1984	NES	NES	NES	NES	NES
5	1985	NES	NES	NES	NES	NES
6	1986	NES	NES	NES	NES	NES
7	1987	NES	NES	NES	NES	NES
8	1988	NES	NES	NES	NES	NES
9	1989	GB	GB	GB	GB	GB
10	1990	SNES	SNES	SNES	SNES	SNES
11	1991	SNES	GEN	SNES	SNES	SNES
12	1992	GB	GB	GB	SNES	GB
13	1993	SNES	SNES	SNES	SNES	SNES
14	1994	SNES	SNES	PC	SNES	GB
15	1995	SNES	SNES	PC	SNES	PC
16	1996	GB	GB	GB	GB	GB
17	1997	PS	N64	PS	PS	PS
18	1998	GB	GB	GB	GB	GB
19	1999	GB	GB	GB	GB	GB
20	2000	GB	PS	PS	PS	GB
21	2001	PS2	PS2	PS2	PS2	PS2
22	2002	PS2	PS2	PS2	GBA	PS2
23	2003	PS2	GC	PS2	PS2	PS2
24	2004	PS2	PS2	PC	PS2	PS2
25	2005	DS	DS	DS	DS	DS
26	2006	Wii	Wii	Wii	DS	Wii
27	2007	Wii	Wii	Wii	Wii	PS2
28	2008	Wii	Wii	Wii	PSP	Wii
29	2009	Wii	Wii	Wii	Wii	Wii
30	2010	X360	X360	X360	DS	PS3
31	2011	X360	X360	PS3	3DS	PS3
32	2012	PS3	X360	PS3	3DS	PS3
33	2013	PS3	X360	PS3	3DS	PS3
34	2014	PS4	3DS	PS4	3DS	PS4
35	2015	PS4	PS4	PS4	3DS	PS4

	Years	Platfroms_Global	Platfroms_NA	Platfroms_EU	Platfroms_JP	Platfroms_Other
36	2016	PS4	PS4	PS4	3DS	PS4
37	2017	PS4	PS4	PS4	PS4	PS4
38	2020	DS	DS	DS	DS	DS

4) Gráfico de comparación de ventas entre géneros por año

In [ ]:
```
import plotly.express as px
```

In [ ]:
```
fig=px.bar(df,x='Year',y='Global_Sales',color='Genre',title='Ventas entre géneros
fig.show()
```

Insights: 2 Gráficos adicionales interesantes

In [ ]:
```
df1=df.groupby('Genre')['NA_Sales'].sum().sort_values(ascending=False).reset_index
df2=df.groupby('Genre')['EU_Sales'].sum().sort_values(ascending=False).reset_index
df3=df.groupby('Genre')['JP_Sales'].sum().sort_values(ascending=False).reset_index
df4=df.groupby('Genre')['Other_Sales'].sum().sort_values(ascending=False).reset_in

result = df1.merge(df2,on='Genre').merge(df3,on='Genre').merge(df4,on='Genre')
result
```

Out[ ]:

	Genre	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	Action	877.83	525.00	159.95	187.38
1	Sports	683.35	376.85	135.37	134.97
2	Shooter	582.60	313.27	38.28	102.69
3	Platform	447.05	201.63	130.77	51.59
4	Misc	410.24	215.98	107.76	75.32
5	Racing	359.42	238.39	56.69	77.27
6	Role-Playing	327.28	188.06	352.31	59.61
7	Fighting	223.59	101.32	87.35	36.68
8	Simulation	183.31	113.38	63.70	31.52
9	Puzzle	123.78	50.78	57.31	12.55
10	Adventure	105.80	64.13	52.07	16.81
11	Strategy	68.70	45.34	49.46	11.36

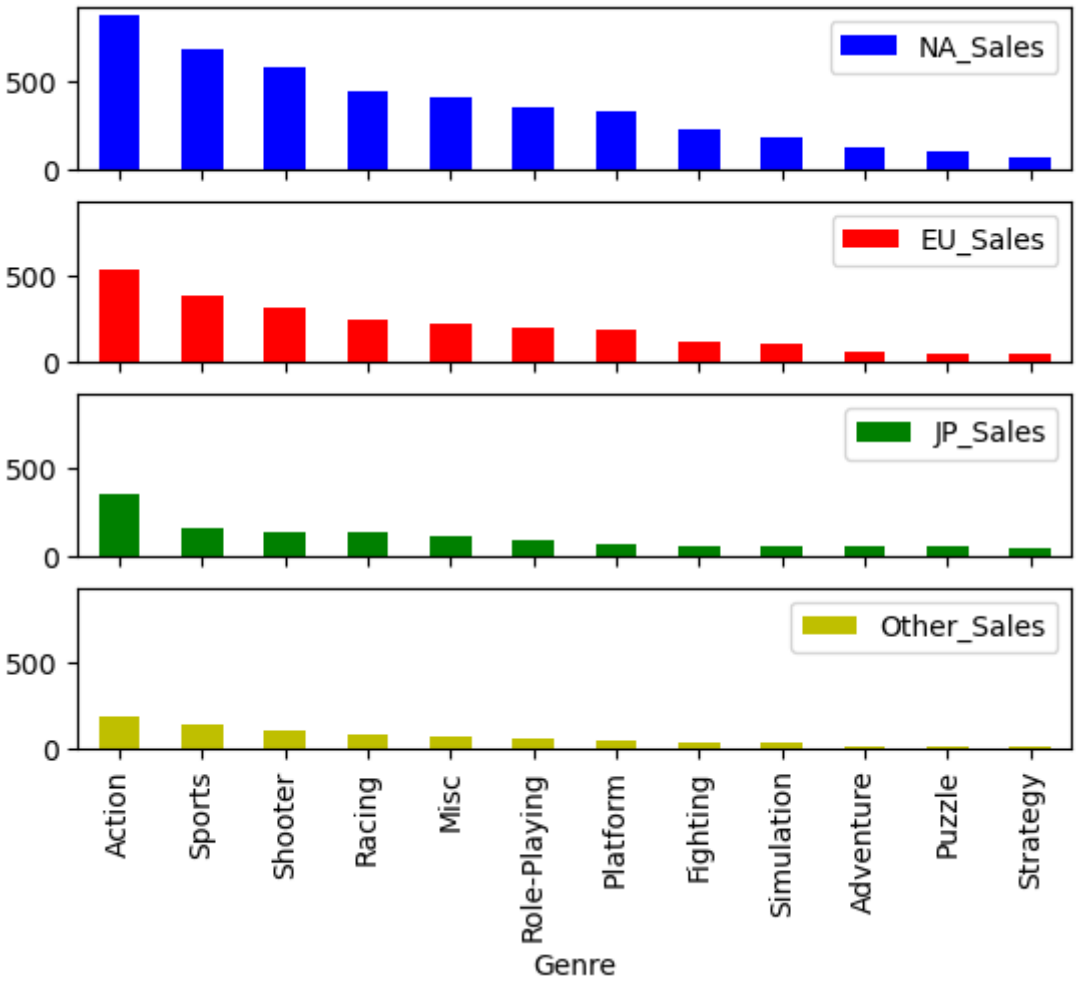
In [ ]:
```
fig,axes=plt.subplots(nrows=4,ncols=1,sharex=True,sharey=True)

df1.plot.bar(x='Genre',ax=axes[0],color='b')
df2.plot.bar(x='Genre',ax=axes[1],color='r')
df3.plot.bar(x='Genre',ax=axes[2],color='g')
df4.plot.bar(x='Genre',ax=axes[3],color='y')

# En este grafico se puede apreciar cuales son los generos
# mas relevantes en este mercado de video juegos y cual es
# la region que tiene una mayor participacion.
# Los generos de Accion, Sport y disparos son los mas
```

```
# importantes en todas las regiones (queria verificar si
# era asi en todos los casos) siendo Norte America aquella
# region en la cual se tiene una mayor demanda por los
# video juegos.
```

Out[ ]: <AxesSubplot: xlabel='Genre'>



In [ ]:
```
total_publisher = df['Publisher'].unique().__len__()
total_publisher
```

Out[ ]: 579

In [ ]:
```
df_publisher=df[df['Year']>2000].groupby('Publisher')['Global_Sales'].sum().sort_v
df5=df_publisher.head(10)
df5
```

Out[ ]:

	Publisher	Global_Sales
0	Nintendo	1150.90
1	Electronic Arts	983.28
2	Activision	662.62
3	Ubisoft	460.44
4	Sony Computer Entertainment	413.86
5	Take-Two Interactive	383.42
6	THQ	304.87
7	Microsoft Game Studios	239.58
8	Sega	218.73
9	Konami Digital Entertainment	214.18

In [ ]:
```
fig= px.pie(df5, names='Publisher',values='Global_Sales', title='Participacion del
fig.show()

# En este grafico se puede apreciar la concentracion del mercado global de videoju
# en los ultimos 20 anos.
# 2 empresas tienen el control de un poco mas del 40% del mercado mientras que el
# restante esta repartido entre 8 empresas. Por otro lado, Aunque si el mercado ma
# importante esta en norteamerica la empresa con mayor participacion es Japones (N
# mientras que la segunda y la tercera son americanas. La Union Europea se hace pr
# solo en la cuarta posicion con Ubisoft (empresa Francesa).
```

In [ ]:
```

```