

e5c1f0c2-db9f-43a0-99b9-4875c52427e8

July 23, 2024

¡Hola!

Mi nombre es Tonatiuh Cruz. Me complace revisar tu proyecto hoy.

Al identificar cualquier error inicialmente, simplemente los destacaré. Te animo a localizar y abordar los problemas de forma independiente como parte de tu preparación para un rol como data-scientist. En un entorno profesional, tu líder de equipo seguiría un enfoque similar. Si encuentras la tarea desafiante, proporcionaré una pista más específica en la próxima iteración.

Encontrarás mis comentarios a continuación - **por favor no los muevas, modifiques o elimines.**

Puedes encontrar mis comentarios en cajas verdes, amarillas o rojas como esta:

Comentario del revisor

Éxito. Todo está hecho correctamente.

Comentario del revisor

Observaciones. Algunas recomendaciones.

Comentario del revisor

Necesita corrección. El bloque requiere algunas correcciones. El trabajo no puede ser aceptado con comentarios en rojo.

Puedes responderme utilizando esto:

Respuesta del estudiante.

Comentario del revisor

¡Hola! Recuerda que para los ejercicio de análisis de datos es necesario desarrollar la introducción, objetivo y etapas de análisis. Esto es crucial para cualquier proyecto, ya que establece una guía clara sobre los pasos a seguir. Tener estos elementos bien definidos desde el principio nos permite trabajar de manera más organizada y eficiente. En un futuro lo podrías complementar con una tabla de contenido.

Comentario del estudiante: Hola, Tonatiuh. Bueno, lo que sí puse fueron los encabezados en las cajas de Markdown. El índice se puede acceder desde el botón de Tabla de Contenidos de la plataforma.

1 Proyecto Integrado

Trabajo para la tienda online Ice que vende videojuegos por todo el mundo. En los datos delante de mí constan los nombres de juegos, las plataformas, los géneros, las reseñas de usuarios y expertos y los datos históricos sobre ventas. El objetivo del proyecto es identificar patrones que determinen si un juego tiene éxito o no. Esto permitirá detectar proyectos prometedores.

1.1 Inicialización de archivo de datos y estudio de información general

1.1.1 Carga de librerías y datos

```
[1]: #Cargar las librerías
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from scipy import stats as st
import math as mt
```

```
[2]: # Carga los archivos de datos
df_games=pd.read_csv('/datasets/games.csv')
```

Comentario del revisor:

Has realizado un excelente trabajo al importar los datos y las bibliotecas necesarias.

1.1.2 Estudio de información general

```
[3]: # Información general
df_games.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Name                  16713 non-null  object
1   Platform              16715 non-null  object
2   Year_of_Release       16446 non-null  float64
3   Genre                 16713 non-null  object
4   NA_sales              16715 non-null  float64
5   EU_sales              16715 non-null  float64
6   JP_sales              16715 non-null  float64
7   Other_sales           16715 non-null  float64
8   Critic_Score          8137 non-null   float64
9   User_Score            10014 non-null  object
10  Rating                9949 non-null   object
dtypes: float64(6), object(5)
memory usage: 1.4+ MB
```

```
[4]: df_games.sample(50)
```

[4]:

	Name	Platform	\
3665	Mortal Kombat 3	PS	
15111	Kensei: Sacred Fist	PS	
13029	Tom Clancy's HAWX 2	PC	
14620	Phantasy Star Online 2: Episode 3 Deluxe Package	PSV	
13351	Romance of the Three Kingdoms VI: Awakening of...	PS	
2648	Perfect Dark Zero	X360	
13290	Silhouette Mirage	PS	
756	Star Wars: Rogue Squadron	N64	
735	Devil May Cry 3: Dante's Awakening Special Edi...	PS2	
13451	Toukiden Kiwami	PSP	
7477	Pro Yaky? Spirits 2010	PS3	
9120	Pro Evolution Soccer 2015	XOne	
3934	Tiger Woods PGA Tour 13	X360	
13361	Gem Smashers	Wii	
12306	GRID 2	PC	
10168	XXX	GBA	
3327	Wave Race: Blue Storm	GC	
2160	Tom Clancy's Splinter Cell: Chaos Theory	PS2	
11406	Robotron X	PS	
11857	Summon Night	DS	
11966	Disney Sports Skateboarding	GBA	
10032	NBA Jam 2002	GBA	
4839	Sega Superstars Tennis	PS3	
3124	Crafting Mama	DS	
5516	NCAA GameBreaker 2001	PS	
12913	WTA Tour Tennis	PS2	
6012	Harvest Moon: Magical Melody	Wii	
4350	NBA Live 2005	GC	
6149	Operation Flashpoint: Red River	X360	
6621	Amped 2	XB	
13924	Harukanaru Toki no Naka de 4: Aizouban	PSP	
9967	Stolen Song	PS	
8073	Skylanders Imaginators	WiiU	
1858	F-Zero X	N64	
11462	Major League Baseball 2K11	PC	
13372	Chocobo to Mahou no Ehon: Majo to Shoujo to 5-...	DS	
13222	Hummer Badlands	PS2	
6908	Dynasty Warriors 6 Empires	X360	
4526	Max Payne 3	PC	
3234	Tak 2: The Staff of Dreams	PS2	
8736	Major League Baseball 2K11	Wii	
12048	BlazBlue: Chrono Phantasma	PSV	
5840	Mario Party: Star Rush	3DS	
4068	Steel Diver	3DS	
11886	Kengo: Legend of The 9	X360	
5488	GT Advance Championship Racing	GBA	

4347		LEGO Rock Band	DS
9050	World Championship Poker: Howard Lederer - All In		PSP
1252		Football Manager 2012	PC
15858		Kaleidoeve	PSV

	Year_of_Release	Genre	NA_sales	EU_sales	JP_sales	\
3665	1995.0	Fighting	0.29	0.20	0.02	
15111	1997.0	Fighting	0.01	0.01	0.00	
13029	2010.0	Action	0.00	0.04	0.00	
14620	2015.0	Action	0.00	0.00	0.03	
13351	1998.0	Strategy	0.03	0.02	0.00	
2648	2005.0	Shooter	0.66	0.02	0.03	
13290	1998.0	Platform	0.03	0.02	0.00	
756	1998.0	Simulation	1.60	0.46	0.08	
735	2006.0	Action	1.06	0.82	0.05	
13451	2014.0	Action	0.00	0.00	0.05	
7477	2010.0	Sports	0.00	0.00	0.21	
9120	2014.0	Sports	0.03	0.10	0.00	
3934	2012.0	Sports	0.29	0.18	0.00	
13361	2011.0	Platform	0.04	0.00	0.00	
12306	2013.0	Racing	0.00	0.06	0.00	
10168	2002.0	Action	0.08	0.03	0.00	
3327	2001.0	Racing	0.40	0.10	0.09	
2160	2005.0	Action	0.36	0.45	0.00	
11406	1996.0	Shooter	0.05	0.03	0.00	
11857	2008.0	Role-Playing	0.00	0.00	0.07	
11966	2002.0	Sports	0.05	0.02	0.00	
10032	2002.0	Sports	0.08	0.03	0.00	
4839	2008.0	Sports	0.13	0.20	0.00	
3124	2010.0	Simulation	0.39	0.19	0.01	
5516	2000.0	Sports	0.18	0.12	0.00	
12913	2001.0	Sports	0.03	0.02	0.00	
6012	2008.0	Simulation	0.23	0.03	0.00	
4350	2004.0	Sports	0.35	0.09	0.00	
6149	2011.0	Shooter	0.10	0.14	0.01	
6621	2003.0	Sports	0.20	0.04	0.00	
13924	2010.0	Role-Playing	0.00	0.00	0.04	
9967	1998.0	Adventure	0.00	0.00	0.11	
8073	2016.0	Platform	0.09	0.08	0.00	
1858	1998.0	Racing	0.45	0.33	0.29	
11462	2011.0	Sports	0.06	0.01	0.00	
13372	2008.0	Role-Playing	0.00	0.00	0.05	
13222	2006.0	Racing	0.02	0.02	0.00	
6908	2009.0	Action	0.16	0.01	0.04	
4526	2012.0	Shooter	0.16	0.22	0.00	
3234	2004.0	Platform	0.30	0.24	0.00	
8736	2011.0	Sports	0.14	0.00	0.00	

12048	2014.0	Fighting	0.03	0.00	0.03
5840	2016.0	Misc	0.08	0.13	0.08
4068	2011.0	Action	0.31	0.09	0.05
11886	2006.0	Fighting	0.06	0.01	0.00
5488	2001.0	Racing	0.24	0.09	0.00
4347	2009.0	Misc	0.32	0.09	0.00
9050	2006.0	Misc	0.13	0.00	0.00
1252	2011.0	Sports	0.02	1.16	0.00
15858	2015.0	Adventure	0.00	0.00	0.02

	Other_sales	Critic_Score	User_Score	Rating
3665	0.04	NaN	NaN	NaN
15111	0.00	NaN	NaN	NaN
13029	0.01	66.0	3.8	T
14620	0.00	NaN	NaN	NaN
13351	0.00	NaN	NaN	NaN
2648	0.06	81.0	7.3	M
13290	0.00	69.0	7.5	E
756	0.03	NaN	NaN	NaN
735	0.28	NaN	NaN	NaN
13451	0.00	NaN	NaN	NaN
7477	0.00	NaN	NaN	NaN
9120	0.01	79.0	7.3	E
3934	0.04	77.0	5.6	E
13361	0.00	NaN	tbd	E
12306	0.01	80.0	5.7	E
10168	0.00	50.0	5	E
3327	0.02	80.0	8.3	E
2160	0.14	87.0	8.6	M
11406	0.01	NaN	NaN	NaN
11857	0.00	NaN	NaN	NaN
11966	0.00	42.0	tbd	E
10032	0.00	49.0	tbd	E
4839	0.07	67.0	7.4	E10+
3124	0.05	NaN	tbd	E
5516	0.02	69.0	tbd	E
12913	0.01	50.0	tbd	E
6012	0.02	69.0	8.8	E
4350	0.01	83.0	6.7	E
6149	0.03	69.0	6.1	M
6621	0.01	80.0	7.7	E
13924	0.00	NaN	NaN	NaN
9967	0.01	NaN	NaN	NaN
8073	0.02	76.0	tbd	E10+
1858	0.03	NaN	NaN	NaN
11462	0.01	74.0	5.9	E
13372	0.00	NaN	NaN	NaN

13222	0.01	48.0	tbd	E
6908	0.02	59.0	7.3	T
4526	0.05	87.0	7.6	M
3234	0.08	71.0	8	E
8736	0.01	NaN	tbd	E
12048	0.01	NaN	7.8	T
5840	0.02	68.0	6.8	E
4068	0.03	58.0	6.3	E10+
11886	0.01	38.0	4.1	M
5488	0.01	82.0	tbd	E
4347	0.04	80.0	7.5	E
9050	0.01	NaN	NaN	NaN
1252	0.33	84.0	8	E
15858	0.00	NaN	NaN	NaN

1.2 Preparación de datos

1.2.1 Nombres de las columnas

```
[5]: #Reemplazar los nombres de las columnas
df_games.columns=df_games.columns.str.lower()
print(df_games.columns)
```

```
Index(['name', 'platform', 'year_of_release', 'genre', 'na_sales', 'eu_sales',
      'jp_sales', 'other_sales', 'critic_score', 'user_score', 'rating'],
      dtype='object')
```

1.2.2 Valores ausentes

```
[6]: #Análisis de valores ausentes
#-Columna 'name'
print(df_games[df_games['name'].isna()])
print('\n\n\n', df_games.query("platform == 'GEN'"))
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	\
659	NaN	GEN	1993.0	NaN	1.78	0.53	0.00	
14244	NaN	GEN	1993.0	NaN	0.00	0.00	0.03	

	other_sales	critic_score	user_score	rating
659	0.08	NaN	NaN	NaN
14244	0.00	NaN	NaN	NaN

	name	platform	\
145	Sonic the Hedgehog 2	GEN	
257	Sonic the Hedgehog	GEN	
564	Mortal Kombat	GEN	

588	Streets of Rage	GEN
659	NaN	GEN
832	NBA Jam	GEN
960	Sonic & Knuckles	GEN
996	Sonic the Hedgehog 3	GEN
1099	Street Fighter II': Special Champion Edition	GEN
1365	Disney's The Lion King	GEN
1453	Mortal Kombat 3	GEN
1842	NBA Jam Tournament Edition	GEN
6451	Virtua Racing	GEN
7885	Shining Force II	GEN
8893	Super Street Fighter II	GEN
9000	Lunar 2: Eternal Blue(sales, but wrong system)	GEN
9563	Gunstar Heroes	GEN
9787	Ecco the Dolphin	GEN
11563	Yuu Yuu Hakusho: Makyo Toitsusen	GEN
11660	Dragon Slayer: The Legend of Heroes	GEN
11986	Ecco: The Tides of Time	GEN
12098	Street Fighter II': Special Champion Edition (...)	GEN
12264	Streets of Rage 3	GEN
12984	Dynamite Headdy	GEN
13343	Beyond Oasis	GEN
14085	J-League Pro Striker 2	GEN
14132	Castlevania Bloodlines	GEN
14244	NaN	GEN
14488	Puzzle & Action: Tant-R	GEN

	year_of_release	genre	na_sales	eu_sales	jp_sales	\
145	1992.0	Platform	4.47	1.20	0.16	
257	1991.0	Platform	3.03	0.91	0.26	
564	1992.0	Fighting	1.95	0.63	0.00	
588	1990.0	Action	1.86	0.55	0.11	
659	1993.0	NaN	1.78	0.53	0.00	
832	1992.0	Sports	1.75	0.25	0.00	
960	1994.0	Platform	1.24	0.43	0.03	
996	1994.0	Platform	1.02	0.47	0.20	
1099	1992.0	Fighting	1.00	0.30	0.31	
1365	1994.0	Platform	0.97	0.37	0.03	
1453	1994.0	Fighting	1.03	0.27	0.00	
1842	1994.0	Sports	0.95	0.14	0.00	
6451	1994.0	Racing	0.00	0.00	0.26	
7885	1993.0	Strategy	0.00	0.00	0.19	
8893	1993.0	Fighting	0.00	0.00	0.15	
9000	1994.0	Role-Playing	0.00	0.00	0.14	
9563	1992.0	Shooter	0.00	0.00	0.13	
9787	1992.0	Adventure	0.00	0.00	0.12	
11563	1994.0	Fighting	0.00	0.00	0.08	
11660	1994.0	Role-Playing	0.00	0.00	0.08	

11986	1993.0	Adventure	0.00	0.00	0.07
12098	1993.0	Action	0.00	0.00	0.07
12264	1993.0	Action	0.00	0.00	0.07
12984	1993.0	Platform	0.00	0.00	0.05
13343	1993.0	Role-Playing	0.00	0.00	0.05
14085	1994.0	Sports	0.00	0.00	0.04
14132	1994.0	Platform	0.00	0.00	0.04
14244	1993.0	NaN	0.00	0.00	0.03
14488	1994.0	Misc	0.00	0.00	0.03

	other_sales	critic_score	user_score	rating
145	0.19	NaN	NaN	NaN
257	0.13	NaN	NaN	NaN
564	0.09	NaN	NaN	NaN
588	0.08	NaN	NaN	NaN
659	0.08	NaN	NaN	NaN
832	0.05	NaN	NaN	NaN
960	0.12	NaN	NaN	NaN
996	0.07	NaN	NaN	NaN
1099	0.04	NaN	NaN	NaN
1365	0.05	NaN	NaN	NaN
1453	0.04	NaN	NaN	NaN
1842	0.03	NaN	NaN	NaN
6451	0.00	NaN	NaN	NaN
7885	0.00	NaN	NaN	NaN
8893	0.00	NaN	NaN	NaN
9000	0.00	NaN	NaN	NaN
9563	0.00	NaN	NaN	NaN
9787	0.00	NaN	NaN	NaN
11563	0.00	NaN	NaN	NaN
11660	0.00	NaN	NaN	NaN
11986	0.00	NaN	NaN	NaN
12098	0.00	NaN	NaN	NaN
12264	0.00	NaN	NaN	NaN
12984	0.00	NaN	NaN	NaN
13343	0.00	NaN	NaN	NaN
14085	0.00	NaN	NaN	NaN
14132	0.00	NaN	NaN	NaN
14244	0.00	NaN	NaN	NaN
14488	0.00	NaN	NaN	NaN

```
[7]: #Reemplazo de valores ausentes
      #-Columna 'name'
      df_games['name'].fillna('N/A', inplace=True)
      print(df_games[df_games['name'].isna()])
```

Empty DataFrame

Columns: [name, platform, year_of_release, genre, na_sales, eu_sales, jp_sales,


```
other_sales, critic_score, user_score, rating]
Index: []
```

En la columna 'name' solo hubo dos valores ausentes de juegos del año 1993 y de plataforma GEN. No son ausencias relevantes para nuestro objetivo, pero las filas están acompañadas con datos ausentes en otras categorías .

Reemplacé dichos valores ausentes de la columna 'name' por el marcador 'N/A'. Con nombres de películas, juegos u otras obras prefiero este acrónimo a otros marcadores (como 'unknown', 'missing' 'UNKNOWN', etc.) por considerarlo más distintivo y así evitar posibles confusiones.

```
[8]: #Análisis de valores ausentes
      #Columna 'year_of_release'
      df_games[df_games['year_of_release'].isna()].sample(50)
```

```
[8]:
```

	name	platform	year_of_release	\
678	LEGO Indiana Jones: The Original Adventures	Wii	NaN	
16329	Brothers in Arms: Furious 4	X360	NaN	
16079	Football Manager 2007	X360	NaN	
4945	Robert Ludlum's The Bourne Conspiracy	PS3	NaN	
5624	Singularity	X360	NaN	
2522	The Lord of the Rings: War in the North	PS3	NaN	
14449	Fullmetal Alchemist: Brotherhood	PSP	NaN	
3289	Advance Wars: Days of Ruin	DS	NaN	
8364	Sword of the Samurai	PS2	NaN	
15816	Without Warning	XB	NaN	
11957	Yoostar on MTV	X360	NaN	
2108	Suikoden III	PS2	NaN	
6773	Mega Man X Collection	PS2	NaN	
14252	Swords	Wii	NaN	
4635	NHL Slapshot	Wii	NaN	
10764	Luminous Arc 2 (JP sales)	DS	NaN	
5769	Star Wars Jedi Knight II: Jedi Outcast	XB	NaN	
16277	Homeworld Remastered Collection	PC	NaN	
2969	Test Drive Unlimited 2	X360	NaN	
10993	The Daring Game for Girls	Wii	NaN	
9516	Nintendo Puzzle Collection	GC	NaN	
10317	Happy Feet Two	PS3	NaN	
2361	Rhythm Heaven	Wii	NaN	
5857	The Dukes of Hazzard II: Daisy Dukes It Out	PS	NaN	
14141	Major League Baseball 2K8	PSP	NaN	
15675	Cabela's Alaskan Adventure	PS2	NaN	
9119	Our House Party!	Wii	NaN	
10837	Egg Monster Hero	DS	NaN	
7008	Tribes: Aerial Assault	PS2	NaN	
15081	Wii de Asobu: Metroid Prime	Wii	NaN	
7406	DanceDanceRevolution II	Wii	NaN	
6954	Jet X20	PS2	NaN	

609	Space Invaders	2600	NaN
3081	LEGO Harry Potter: Years 5-7	DS	NaN
10686	Happy Feet Two	DS	NaN
16348	Agarest Senki: Re-appearance	PS3	NaN
9855	The Daring Game for Girls	DS	NaN
14369	Majesty 2: The Fantasy Kingdom Sim	X360	NaN
3997	Wet	X360	NaN
13564	RollerCoaster Tycoon	PC	NaN
13289	Reader Rabbit 2nd Grade	Wii	NaN
14325	Beyond the Labyrinth	3DS	NaN
8067	Backbreaker	X360	NaN
627	Rock Band	X360	NaN
13060	Captain America: Super Soldier	DS	NaN
3223	Metal Gear Solid 2: Substance	XB	NaN
14749	Tom and Jerry in War of the Whiskers	XB	NaN
15338	Mario Tennis	3DS	NaN
11925	Port Royale 3	PS3	NaN
15953	PDC World Championship Darts 2008	DS	NaN

	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score \
678	Action	1.51	0.61	0.00	0.21	78.0
16329	Shooter	0.01	0.00	0.00	0.00	NaN
16079	Sports	0.00	0.01	0.00	0.00	NaN
4945	Action	0.18	0.14	0.00	0.06	70.0
5624	Shooter	0.24	0.05	0.00	0.02	76.0
2522	Action	0.25	0.42	0.01	0.13	63.0
14449	Action	0.00	0.00	0.03	0.00	NaN
3289	Strategy	0.43	0.12	0.00	0.05	86.0
8364	Fighting	0.00	0.00	0.17	0.00	NaN
15816	Shooter	0.01	0.00	0.00	0.00	45.0
11957	Misc	0.07	0.00	0.00	0.01	49.0
2108	Role-Playing	0.29	0.23	0.38	0.08	86.0
6773	Misc	0.12	0.09	0.00	0.03	73.0
14252	Fighting	0.03	0.00	0.00	0.00	NaN
4635	Sports	0.39	0.00	0.00	0.02	76.0
10764	Role-Playing	0.00	0.00	0.10	0.00	NaN
5769	Shooter	0.23	0.07	0.00	0.01	81.0
16277	Strategy	0.00	0.01	0.00	0.00	86.0
2969	Racing	0.30	0.31	0.00	0.07	68.0
10993	Adventure	0.09	0.00	0.00	0.01	NaN
9516	Puzzle	0.00	0.00	0.13	0.00	NaN
10317	Action	0.09	0.00	0.00	0.01	NaN
2361	Misc	0.11	0.00	0.77	0.01	NaN
5857	Racing	0.17	0.11	0.00	0.02	53.0
14141	Sports	0.03	0.00	0.00	0.00	63.0
15675	Sports	0.01	0.01	0.00	0.00	NaN
9119	Simulation	0.13	0.00	0.00	0.01	NaN

10837	Role-Playing	0.00	0.00	0.09	0.00	NaN
7008	Shooter	0.11	0.09	0.00	0.03	73.0
15081	Shooter	0.00	0.00	0.02	0.00	NaN
7406	Misc	0.20	0.00	0.00	0.01	NaN
6954	Racing	0.11	0.09	0.00	0.03	NaN
609	Shooter	2.36	0.14	0.00	0.03	NaN
3081	Action	0.34	0.25	0.00	0.07	69.0
10686	Action	0.08	0.01	0.00	0.01	NaN
16348	Role-Playing	0.00	0.00	0.01	0.00	NaN
9855	Adventure	0.11	0.00	0.00	0.01	NaN
14369	Simulation	0.03	0.00	0.00	0.00	NaN
3997	Shooter	0.23	0.21	0.01	0.05	69.0
13564	Strategy	0.02	0.02	0.00	0.01	NaN
13289	Misc	0.04	0.00	0.00	0.00	NaN
14325	Role-Playing	0.00	0.00	0.03	0.00	NaN
8067	Sports	0.17	0.00	0.00	0.01	54.0
627	Misc	1.93	0.33	0.00	0.21	92.0
13060	Action	0.05	0.00	0.00	0.00	NaN
3223	Action	0.38	0.22	0.00	0.03	87.0
14749	Fighting	0.02	0.01	0.00	0.00	NaN
15338	Sports	0.00	0.00	0.02	0.00	NaN
11925	Simulation	0.06	0.00	0.00	0.01	NaN
15953	Sports	0.01	0.00	0.00	0.00	NaN

	user_score	rating
678	6.6	E10+
16329	NaN	M
16079	NaN	NaN
4945	7.6	T
5624	7.7	M
2522	7	M
14449	NaN	NaN
3289	8.7	E10+
8364	NaN	NaN
15816	2	M
11957	tbd	T
2108	7.7	T
6773	8.8	E
14252	tbd	T
4635	8.1	E
10764	NaN	NaN
5769	6.8	T
16277	8.2	E10+
2969	6.4	T
10993	tbd	E
9516	NaN	NaN
10317	NaN	NaN

2361	NaN	NaN
5857	tbd	E
14141	tbd	E
15675	NaN	NaN
9119	NaN	NaN
10837	NaN	E
7008	8.8	T
15081	NaN	NaN
7406	tbd	E10+
6954	NaN	NaN
609	NaN	NaN
3081	tbd	E10+
10686	tbd	E
16348	NaN	NaN
9855	tbd	E
14369	NaN	NaN
3997	7.3	M
13564	NaN	E
13289	tbd	E
14325	NaN	NaN
8067	7.6	E
627	8.2	T
13060	NaN	NaN
3223	8.5	M
14749	NaN	NaN
15338	NaN	NaN
11925	8.9	T
15953	NaN	NaN

Comentario del revisor:

En ocasiones podemos completar los valores faltantes con información recaba mediante una investigación. Es muy común que cuando trabajamos con datos en la vida real, mucho de estos vengan con valores nulos que en ocasiones tendremos que completar con proxys

```
[9]: print(df_games['name'].value_counts())
      print(df_games[df_games['name']=="FIFA 14"])
      print(df_games[df_games['name']=="Ratatouille"])
```

Need for Speed: Most Wanted	12
Ratatouille	9
Madden NFL 07	9
LEGO Marvel Super Heroes	9
FIFA 14	9
..	
Famicom Mini: Wrecking Crew	1
Moto Racer World Tour	1
Mega Man & Bass	1

Neighborhood Games 1
 Cruis'n World 1
 Name: name, Length: 11560, dtype: int64

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	\
126	FIFA 14	PS3	2013.0	Sports	0.78	4.24	0.07	
271	FIFA 14	X360	2013.0	Sports	0.92	2.89	0.01	
469	FIFA 14	PS4	2013.0	Sports	0.61	1.85	0.11	
1753	FIFA 14	XOne	2013.0	Sports	0.41	0.66	0.00	
4699	FIFA 14	PSV	2013.0	Sports	0.08	0.23	0.01	
4809	FIFA 14	PC	2013.0	Sports	0.01	0.36	0.00	
4985	FIFA 14	Wii	2013.0	Sports	0.00	0.36	0.00	
6871	FIFA 14	3DS	2013.0	Sports	0.00	0.22	0.00	
7875	FIFA 14	PSP	2013.0	Sports	0.00	0.15	0.00	

	other_sales	critic_score	user_score	rating
126	1.37	86.0	4.3	E
271	0.40	84.0	4.2	E
469	0.44	87.0	6.3	E
1753	0.09	88.0	5.8	E
4699	0.09	NaN	1.8	E
4809	0.03	87.0	4.6	E
4985	0.02	NaN	4.3	E
6871	0.01	NaN	NaN	NaN
7875	0.04	NaN	3.4	E

	name	platform	year_of_release	genre	na_sales	eu_sales	\
1598	Ratatouille	DS	2007.0	Action	0.49	0.62	
2442	Ratatouille	PS2	2007.0	Action	0.31	0.00	
3069	Ratatouille	PSP	2007.0	Action	0.22	0.27	
3902	Ratatouille	PS3	2007.0	Action	0.09	0.32	
3948	Ratatouille	Wii	2007.0	Action	0.43	0.03	
6398	Ratatouille	X360	2007.0	Action	0.23	0.02	
7681	Ratatouille	GBA	2007.0	Action	0.14	0.05	
9015	Ratatouille	GC	2007.0	Action	0.11	0.03	
14514	Ratatouille	PC	2007.0	Action	0.01	0.01	

	jp_sales	other_sales	critic_score	user_score	rating
1598	0.0	0.14	NaN	NaN	NaN
2442	0.0	0.53	65.0	8	E
3069	0.0	0.16	64.0	7.1	E
3902	0.0	0.10	55.0	4.8	E
3948	0.0	0.04	62.0	6.7	E
6398	0.0	0.02	56.0	tbd	E
7681	0.0	0.00	65.0	tbd	E
9015	0.0	0.00	60.0	5.6	E
14514	0.0	0.00	NaN	7.9	E

En la columna 'year_of_release' hay varios valores ausentes que no obedecen a ningún patrón en particular. Es extraña su presencia asociada en títulos tan populares (como por ejemplo, "Star

Wars Jedi Knight II: Jedi Outcast” o “BiosHock 2”). Puede ser un problema en la disponibilidad de estos valores para ciertos juegos en servidores virtuales o probablemente no se considerará su fecha de estreno en juegos de plataformas distintas a la original del título. Puse a prueba esta segunda hipótesis buscando duplicados en los nombres de los juegos y no se halló relación, más bien se encontró que la fecha de lanzamiento está presente y es la misma indistintamente de la plataforma.

En todo caso, la magnitud del número de valores ausentes no se consideró demasiado significativo.

```
[10]: #Reemplazo de valores ausentes
#-Columna 'year_of_release'
df_games['year_of_release'].fillna(0, inplace=True)
df_games['year_of_release']=df_games['year_of_release'].astype('int')
df_games.info()
print(df_games[df_games['year_of_release']==0])
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 16715 entries, 0 to 16714
```

```
Data columns (total 11 columns):
```

#	Column	Non-Null Count	Dtype
0	name	16715 non-null	object
1	platform	16715 non-null	object
2	year_of_release	16715 non-null	int64
3	genre	16713 non-null	object
4	na_sales	16715 non-null	float64
5	eu_sales	16715 non-null	float64
6	jp_sales	16715 non-null	float64
7	other_sales	16715 non-null	float64
8	critic_score	8137 non-null	float64
9	user_score	10014 non-null	object
10	rating	9949 non-null	object

```
dtypes: float64(5), int64(1), object(5)
```

```
memory usage: 1.4+ MB
```

	name	platform	year_of_release	\
183	Madden NFL 2004	PS2	0	
377	FIFA Soccer 2004	PS2	0	
456	LEGO Batman: The Videogame	Wii	0	
475	wwe Smackdown vs. Raw 2006	PS2	0	
609	Space Invaders	2600	0	
...	
16373	PDC World Championship Darts 2008	PSP	0	
16405	Freaky Flyers	GC	0	
16448	Inversion	PC	0	
16458	Hakuouki: Shinsengumi Kitan	PS3	0	
16522	Virtua Quest	GC	0	

	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score	\
--	-------	----------	----------	----------	-------------	--------------	---

183	Sports	4.26	0.26	0.01	0.71	94.0
377	Sports	0.59	2.36	0.04	0.51	84.0
456	Action	1.80	0.97	0.00	0.29	74.0
475	Fighting	1.57	1.02	0.00	0.41	NaN
609	Shooter	2.36	0.14	0.00	0.03	NaN
...
16373	Sports	0.01	0.00	0.00	0.00	43.0
16405	Racing	0.01	0.00	0.00	0.00	69.0
16448	Shooter	0.01	0.00	0.00	0.00	59.0
16458	Adventure	0.01	0.00	0.00	0.00	NaN
16522	Role-Playing	0.01	0.00	0.00	0.00	55.0

	user_score	rating
183	8.5	E
377	6.4	E
456	7.9	E10+
475	NaN	NaN
609	NaN	NaN
...
16373	tbd	E10+
16405	6.5	T
16448	6.7	M
16458	NaN	NaN
16522	5.5	T

[269 rows x 11 columns]

Los valores ausentes de la columna 'year_of_release' se reemplazaron por valores de 0. Eran solamente 269 valores con este problema, por cuanto su afectación en cálculos ulteriores no se considera significativa. Reemplazar los años ausentes por valores calculados a partir de medidas de tendencia central era una posibilidad, pero opté por no adulterar los datos; más bien descartaría aquellos cuyo año de liberación equivale a 0 para operaciones ulteriores que involucren el año. Adicionalmente el tipo de datos de la columna fue cambiada de flotante a entero porque los años de liberación de una obra son siempre valores discretos.

```
[11]: #Análisis de valores ausentes
#-Columna 'genre'
print(df_games[df_games['genre'].isna()])
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	\
659	N/A	GEN	1993	NaN	1.78	0.53	0.00	
14244	N/A	GEN	1993	NaN	0.00	0.00	0.03	

	other_sales	critic_score	user_score	rating
659	0.08	NaN	NaN	NaN
14244	0.00	NaN	NaN	NaN

```
[12]: #Reemplazo de valores ausentes
#-Columna 'genre'
df_games['genre'].fillna('N/A', inplace=True)
print(df_games[df_games['genre']=='N/A'])
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	\
659	N/A	GEN	1993	N/A	1.78	0.53	0.00	
14244	N/A	GEN	1993	N/A	0.00	0.00	0.03	

	other_sales	critic_score	user_score	rating
659	0.08	NaN	NaN	NaN
14244	0.00	NaN	NaN	NaN

En la columna 'genre' solo hubo dos valores ausentes de juegos del año 1993 y de plataforma GEN. No son ausencias relevantes para nuestro objetivo, pero las filas están acompañadas con datos ausentes en otras categorías.

Reemplacé dichos valores ausentes de la columna 'genre' por el marcador 'N/A', que puede significar "not available".

```
[13]: #Análisis de valores ausentes
#Columnas 'critic_score' y 'user_score'
df_games[(df_games['critic_score'].isna())|(df_games['user_score'].isna())].
↪sample(60)
```

```
[13]:
```

	name	platform	\
8571	Sengoku Basara: Chronicle Heroes	PSP	
6183	Batman: The Brave and the Bold the Videogame	Wii	
361	WWF SmackDown!	PS	
16016	Sid Meier's Civilization Revolution 2+	PSV	
12840	Mega Man 8 Anniversary Collector's Edition	SAT	
5855	Lumines 11	PSP	
15109	Dance Dance Revolution: Hottest Party 5	Wii	
10680	Micro Machines V3	PS	
5676	Goosebumps HorrorLand	DS	
10919	Galactic Taz Ball	DS	
5660	Madden NFL 2002	GBA	
5178	Kamaitachi no Yoru 2	PS2	
12504	Metal Gear Solid: Social Ops	PS2	
5045	Cabela's Dangerous Hunts 2009	Wii	
1894	Imagine: Babysitters	DS	
16502	Aladdin Magic Racer	Wii	
11086	Bleach: Heat the Soul 6	PSP	
11881	SD Gundam: Gashapon Wars	GC	
13711	B.L.U.E.: Legend of Water	PS	
12345	Dynasty Warriors Vs	3DS	
11903	Disney's Donald Duck Advance	GBA	
3732	Let's Draw!	DS	

7040	What's Cooking? Jamie Oliver	DS
13204	George of the Jungle and the Search for the Se...	DS
3263	Super Robot Taisen Gaiden	PS
4124	Jake Power: Fireman	DS
6906	Amazing Adventures: The Forgotten Ruins	DS
1291	Imagine: Master Chef	DS
14910	Harvest Moon: Skytree Village	3DS
4271	Warhawk	PS
4961	Sonic & All-Stars Racing Transformed	PSV
6065	Tomba! 2: The Evil Swine Return	PS
14536	Zoboomafoo: Leapin' Lemurs!	PS
8111	Sing4: The Hits Edition	Wii
8952	Charm Girls Club: Pajama Party	Wii
15568	Transformers: Revenge of the Fallen (XBox 360,...	PC
1440	SaGa Frontier	PS
11974	I Don't Have Many Friends Portable	PSP
9185	Jim Henson's the Muppets: On With the Show!	GBA
3883	Fishing Derby	2600
11743	Pinball Hall of Fame: The Gottlieb Collection	XB
11912	Discovery Kids: Parrot Pals	DS
15398	Command & Conquer 3: Deluxe Edition	PC
30	Pokémon Yellow: Special Pikachu Edition	GB
3018	The Fairly Odd Parents: Breakin Da Rules	PS2
4857	Disney Infinity 3.0	X360
11845	Uprising X	PS
8954	Tamagotchi Collection	DS
6711	Netsu Chu! Pro Yakyuu 2002	PS2
10748	Famicom Mini: Mappy	GBA
5776	Lord of the Rings: Battle for Middle-Earth	X360
7988	Harvest Moon: A Wonderful Life Special Edition	PS2
12697	Cid to Chocobo no Fushigi na Dungeon: Toki Was...	DS
15640	Snow Portable	PSP
7350	Where's Waldo? The Fantastic Journey	DS
5948	Guardian Heroes	SAT
9575	Assault: Retribution	PS
7048	Mega Man Battle Network 5: Team Colonel / Prot...	GBA
4842	Boku no Natsuyasumi	PS
910	Tomb Raider (2013)	X360

	year_of_release	genre	na_sales	eu_sales	jp_sales	\
8571	2011	Action	0.00	0.00	0.16	
6183	2010	Action	0.20	0.05	0.00	
361	2000	Fighting	2.01	1.35	0.06	
16016	2015	Strategy	0.00	0.00	0.02	
12840	1997	Platform	0.00	0.00	0.06	
5855	2006	Puzzle	0.12	0.12	0.00	
15109	2011	Misc	0.00	0.02	0.00	

10680	1997	Racing	0.06	0.04	0.00
5676	2008	Adventure	0.29	0.00	0.00
10919	2010	Platform	0.09	0.00	0.00
5660	2001	Sports	0.23	0.08	0.00
5178	2002	Adventure	0.00	0.00	0.36
12504	2002	Action	0.03	0.02	0.00
5045	2008	Sports	0.35	0.00	0.00
1894	2008	Simulation	0.63	0.35	0.00
16502	2011	Racing	0.01	0.00	0.00
11086	2009	Fighting	0.00	0.00	0.09
11881	2005	Strategy	0.00	0.00	0.07
13711	0	Adventure	0.00	0.00	0.04
12345	2012	Action	0.00	0.00	0.06
11903	2001	Platform	0.05	0.02	0.00
3732	2008	Misc	0.24	0.23	0.00
7040	2008	Simulation	0.20	0.01	0.00
13204	2008	Platform	0.05	0.00	0.00
3263	2001	Strategy	0.00	0.00	0.58
4124	2008	Adventure	0.08	0.35	0.00
6906	2008	Adventure	0.13	0.08	0.00
1291	2007	Simulation	0.39	0.91	0.00
14910	2016	Simulation	0.02	0.00	0.00
4271	1995	Simulation	0.25	0.17	0.00
4961	2012	Racing	0.09	0.20	0.00
6065	1999	Platform	0.16	0.11	0.00
14536	2001	Action	0.02	0.01	0.00
8111	2011	Misc	0.17	0.00	0.00
8952	2009	Misc	0.14	0.00	0.00
15568	2009	Shooter	0.02	0.00	0.00
1440	1997	Role-Playing	0.15	0.04	1.07
11974	2012	Adventure	0.00	0.00	0.07
9185	2003	Action	0.10	0.04	0.00
3883	0	Sports	0.48	0.03	0.00
11743	2004	Misc	0.06	0.02	0.00
11912	2009	Simulation	0.07	0.00	0.00
15398	2008	Strategy	0.00	0.02	0.00
30	1998	Role-Playing	5.89	5.04	3.12
3018	2003	Platform	0.33	0.25	0.00
4857	2015	Action	0.21	0.15	0.00
11845	1998	Action	0.04	0.03	0.00
8954	2011	Misc	0.00	0.00	0.15
6711	2002	Sports	0.00	0.00	0.25
10748	2004	Platform	0.00	0.00	0.09
5776	2006	Strategy	0.28	0.00	0.00
7988	2004	Simulation	0.09	0.07	0.00
12697	2008	Role-Playing	0.00	0.00	0.06
15640	2007	Adventure	0.00	0.00	0.02

7350	2009	Adventure	0.20	0.00	0.00
5948	1995	Role-Playing	0.00	0.00	0.29
9575	1998	Action	0.07	0.05	0.00
7048	2004	Role-Playing	0.16	0.06	0.00
4842	2000	Adventure	0.00	0.00	0.37
910	2013	Action	0.86	0.84	0.01

	other_sales	critic_score	user_score	rating
8571	0.00	NaN	NaN	NaN
6183	0.02	NaN	NaN	NaN
361	0.16	NaN	NaN	NaN
16016	0.00	NaN	NaN	NaN
12840	0.00	NaN	NaN	NaN
5855	0.07	NaN	NaN	NaN
15109	0.00	NaN	NaN	NaN
10680	0.01	NaN	NaN	NaN
5676	0.02	NaN	tbd	E10+
10919	0.01	NaN	NaN	NaN
5660	0.01	NaN	tbd	E
5178	0.00	NaN	NaN	NaN
12504	0.01	NaN	NaN	NaN
5045	0.03	NaN	4.5	T
1894	0.10	NaN	tbd	E
16502	0.00	NaN	NaN	E
11086	0.00	NaN	NaN	NaN
11881	0.00	NaN	NaN	NaN
13711	0.00	NaN	NaN	NaN
12345	0.00	NaN	NaN	NaN
11903	0.00	NaN	NaN	NaN
3732	0.06	NaN	NaN	NaN
7040	0.02	NaN	NaN	NaN
13204	0.00	44.0	NaN	E
3263	0.04	NaN	NaN	NaN
4124	0.05	NaN	tbd	E
6906	0.02	NaN	tbd	E
1291	0.17	NaN	tbd	E
14910	0.00	NaN	7.4	E
4271	0.03	NaN	NaN	NaN
4961	0.09	NaN	NaN	NaN
6065	0.02	NaN	NaN	NaN
14536	0.00	NaN	NaN	NaN
8111	0.01	NaN	tbd	T
8952	0.01	NaN	tbd	E
15568	0.00	NaN	NaN	NaN
1440	0.09	NaN	NaN	NaN
11974	0.00	NaN	NaN	NaN
9185	0.00	NaN	NaN	NaN

3883	0.01	NaN	NaN	NaN
11743	0.00	NaN	NaN	NaN
11912	0.01	NaN	tbd	E
15398	0.00	NaN	NaN	NaN
30	0.59	NaN	NaN	NaN
3018	0.09	NaN	NaN	NaN
4857	0.03	NaN	NaN	NaN
11845	0.00	NaN	NaN	NaN
8954	0.00	NaN	NaN	NaN
6711	0.00	NaN	NaN	NaN
10748	0.00	NaN	NaN	NaN
5776	0.02	NaN	NaN	NaN
7988	0.02	NaN	8.9	E
12697	0.00	NaN	NaN	NaN
15640	0.00	NaN	NaN	NaN
7350	0.02	NaN	8.9	E
5948	0.00	NaN	NaN	NaN
9575	0.01	NaN	NaN	NaN
7048	0.00	NaN	NaN	NaN
4842	0.03	NaN	NaN	NaN
910	0.17	NaN	NaN	NaN

```
[14]: df_games[(df_games['critic_score']==0.0)|(df_games['user_score']=="0.0")]
```

[14]: Empty DataFrame

Columns: [name, platform, year_of_release, genre, na_sales, eu_sales, jp_sales, other_sales, critic_score, user_score, rating]

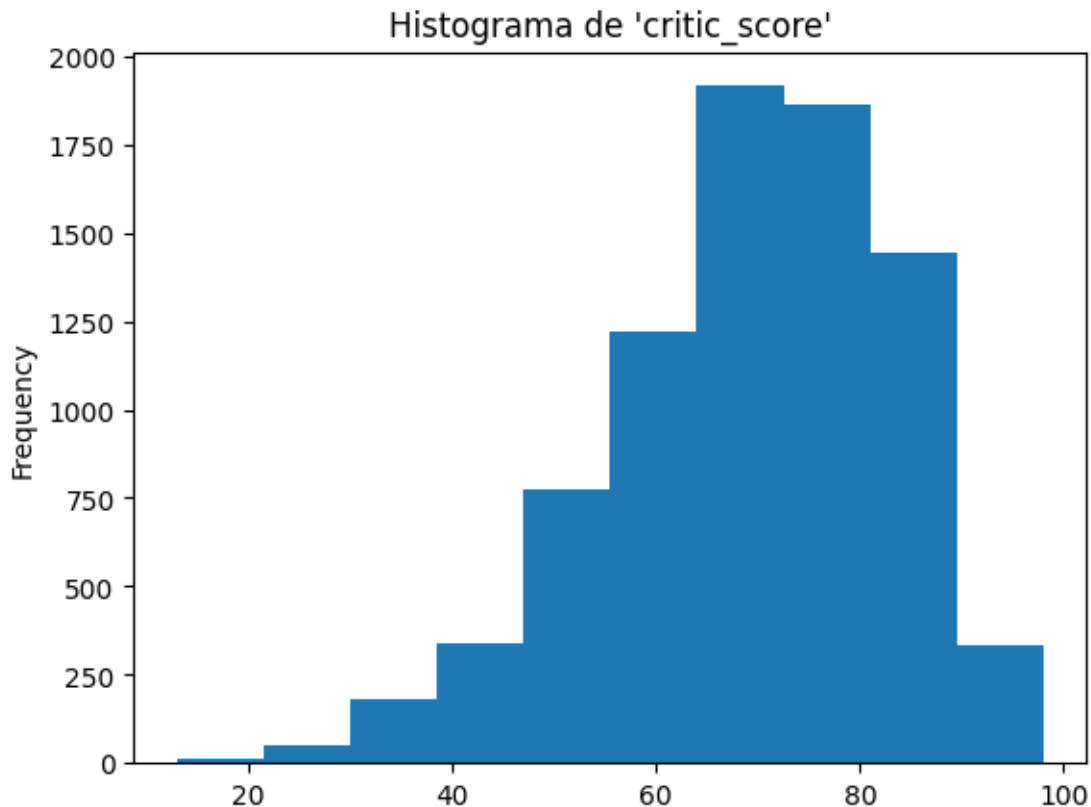
Index: []

```
[15]: #Comprobación de sesgos y valores atípicos en los valores de la columna
      ↪ 'critic_score'
print("Media de 'critic_score':", df_games['critic_score'].mean())
print("Mediana de 'critic_score':", df_games['critic_score'].median())
df_games['critic_score'].plot(kind='hist', title="Histograma de 'critic_score'")
```

Media de 'critic_score': 68.96767850559173

Mediana de 'critic_score': 71.0

```
[15]: <AxesSubplot:title={'center':"Histograma de 'critic_score'"},
      ylabel='Frequency'>
```



```
[16]: #Reemplazo de valores ausentes
      #-Columna 'critic_score'
      df_games['critic_score'].fillna(df_games['critic_score'].median(), inplace=True)
```

Comentario del revisor:

Recuerda que cuando trabajamos con distribuciones sesgadas se recomienda completar los valores nulos con la mediana para no sesgar nuestros resultados

Comentario del estudiante:

Gracias por la corrección. Debí haberlo considerado, sí.

Los valores ausentes de la columnas 'critic_score' y 'user_score' son miles. Casi la mitad de los valores de la primera de estas columnas son ausentes. Esto podría deberse a que algunos juegos lanzados bajo ciertas plataformas no constan en las fuentes de reseñas de críticos y usuarios donde se obtuvieron estos valores. Asimismo, los valores ausentes en estas categorías suelen estar correspondidos con valores ausentes en la columna de 'rating'. Una posible explicación para algunos de estos valores ausentes podría ser la falta de disponibilidad de reseñas para ciertos juegos de los años 80 y 90 (cuando el Internet no era imperante) en las bases de datos, pero se observa esta ausencia para juegos de años más recientes.

Antes de rellenar los valores ausentes se comprobó el sesgo de los valores determinándose en la columna 'critic_score' una asimetría negativa con sesgo a la izquierda. Para normalizar la distribu-

ción y no sesgar los resultados se reemplazaron los valores ausentes por a mediana.

```
[17]: #Análisis de valores ausentes
#-Columna 'user_score'
print(df_games[df_games['user_score']=="0"]) #Revisamos si hay valores reales
↳ de 0 o de 0.0
print(df_games[df_games['user_score']=="0.0"])
print(df_games[df_games['user_score']=="tbd"],"\n\n\n")
print(df_games['user_score'].isna().sum())
```

	name	platform	year_of_release	genre	\
2835	My Little Pony: Pinkie Pie's Party	DS	2008	Adventure	

	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	\
2835	0.66	0.0	0.0	0.06	71.0	0	

rating

2835 E

Empty DataFrame

Columns: [name, platform, year_of_release, genre, na_sales, eu_sales, jp_sales, other_sales, critic_score, user_score, rating]

Index: []

	name	platform	year_of_release	\
119	Zumba Fitness	Wii	2010	
301	Namco Museum: 50th Anniversary	PS2	2005	
520	Zumba Fitness 2	Wii	2011	
645	uDraw Studio	Wii	2010	
657	Frogger's Adventures: Temple of the Frog	GBA	0	
...	
16695	Planet Monsters	GBA	2001	
16697	Bust-A-Move 3000	GC	2003	
16698	Mega Brain Boost	DS	2008	
16704	Plushees	DS	2008	
16706	Men in Black II: Alien Escape	GC	2003	

	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score	\
119	Sports	3.45	2.59	0.0	0.66	71.0	
301	Misc	2.08	1.35	0.0	0.54	61.0	
520	Sports	1.51	1.03	0.0	0.27	71.0	
645	Misc	1.65	0.57	0.0	0.20	71.0	
657	Adventure	2.15	0.18	0.0	0.07	73.0	
...	
16695	Action	0.01	0.00	0.0	0.00	67.0	
16697	Puzzle	0.01	0.00	0.0	0.00	53.0	
16698	Puzzle	0.01	0.00	0.0	0.00	48.0	
16704	Simulation	0.01	0.00	0.0	0.00	71.0	
16706	Shooter	0.01	0.00	0.0	0.00	71.0	

	user_score	rating
119	tbd	E
301	tbd	E10+
520	tbd	T
645	tbd	E
657	tbd	E
...
16695	tbd	E
16697	tbd	E
16698	tbd	E
16704	tbd	E
16706	tbd	T

[2424 rows x 11 columns]

6701

Comentario del revisor:

Recuerda que cuando trabajamos con distribuciones sesgadas se recomienda completar los valores nulos con la mediana para no sesgar nuestros resultados

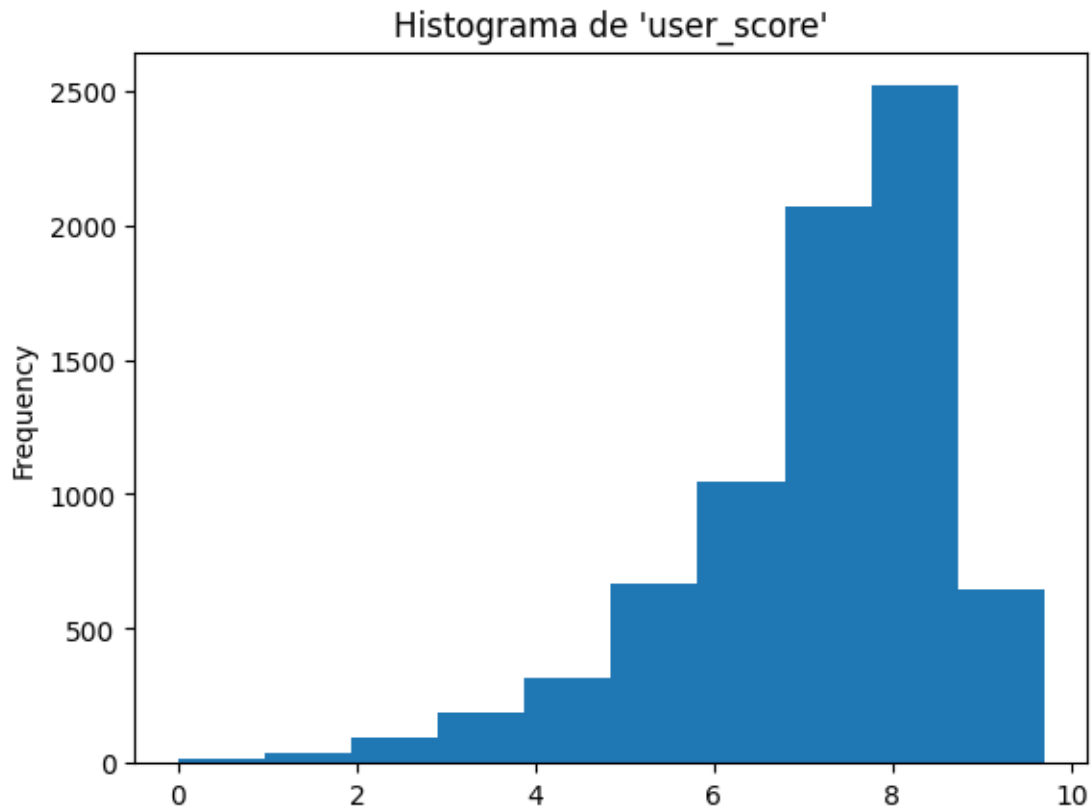
Comentario del estudiante:

Gracias, apliqué la corrección.

```
[18]: #Comprobación de sesgos y valores atípicos en los valores de la columna
      ↪ 'user_score'
df_user_scores=df_games.query("user_score != 'tbd')['user_score']
df_user_scores=df_user_scores.astype('float')
df_user_scores.plot(kind='hist', title="Histograma de 'user_score'")
print("Media de 'user_score':", df_user_scores.mean())
print("Mediana de 'user_score':", df_user_scores.median())
```

Media de 'user_score': 7.125046113306984

Mediana de 'user_score': 7.5



```
[19]: #Reemplazo de valores con la abreviature 'tbd'
      #-Columna 'user_score'
      df_games['user_score']=df_games['user_score'].replace("tbd", df_user_scores.
      ↪median()) #Reemplazo 'tbd' por la mediana
      df_games['user_score'].fillna(df_user_scores.median(), inplace=True) #Reemplazo
      ↪valores ausentes por la mediana
      print(df_games[df_games['user_score']=="tbd"]) #Comprobación
      print(df_games['user_score'].isna().sum()) #Comprobación
```

Empty DataFrame

Columns: [name, platform, year_of_release, genre, na_sales, eu_sales, jp_sales, other_sales, critic_score, user_score, rating]

Index: []

0

```
[20]: #Convertir el tipo de datos de la columna 'user_score' a flotante.
      df_games['user_score']=df_games['user_score'].astype('float')
      df_games.info()
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 16715 entries, 0 to 16714

Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	name	16715 non-null	object
1	platform	16715 non-null	object
2	year_of_release	16715 non-null	int64
3	genre	16715 non-null	object
4	na_sales	16715 non-null	float64
5	eu_sales	16715 non-null	float64
6	jp_sales	16715 non-null	float64
7	other_sales	16715 non-null	float64
8	critic_score	16715 non-null	float64
9	user_score	16715 non-null	float64
10	rating	9949 non-null	object

dtypes: float64(6), int64(1), object(4)

memory usage: 1.4+ MB

Del mismo modo se comprobó el sesgo de los valores en la columna 'user_score' determinándose una asimetría con sesgo a la izquierda. Para normalizar la distribución y no sesgar los resultados se reemplazaron los valores ausentes (6701 celdas) por la mediana de los datos numéricos.

Como los datos de la columna 'user_score' estaban categorizadas como de tipo objeto, se revisó la presencia de la abreviatura tbd (que significa "to be determined") encontrándose 2424 filas con esta abreviatura. Estas filas son equivalentes a valores categóricos ausentes, por lo que para efectos de los cálculos ulteriores se reemplazaron estos valores por la mediana de los valores numéricos. Los valores de la columna 'user_score' finalmente se convirtieron a datos de tipo flotante.

```
[21]: #Reemplazo de valores ausentes
      #-Columna 'rating'
      df_games['rating'].fillna("N/A", inplace=True)
      df_games['rating']=df_games['rating'].replace("tbd", "N/A")
      print(df_games[df_games['rating']=="N/A"])
```

	name	platform	year_of_release	genre	\
1	Super Mario Bros.	NES	1985	Platform	
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	
5	Tetris	GB	1989	Puzzle	
9	Duck Hunt	NES	1984	Shooter	
10	Nintendogs	DS	2005	Simulation	
...	
16710	Samurai Warriors: Sanada Maru	PS3	2016	Action	
16711	LMA Manager 2007	X360	2006	Sports	
16712	Haitaka no Psychedelica	PSV	2016	Adventure	
16713	Spirits & Spells	GBA	2003	Platform	
16714	Winning Post 8 2016	PSV	2016	Simulation	

	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	\
1	29.08	3.58	6.81	0.77	71.0	7.5	
4	11.27	8.89	10.22	1.00	71.0	7.5	

5	23.20	2.26	4.22	0.58	71.0	7.5
9	26.93	0.63	0.28	0.47	71.0	7.5
10	9.05	10.95	1.93	2.74	71.0	7.5
...
16710	0.00	0.00	0.01	0.00	71.0	7.5
16711	0.00	0.01	0.00	0.00	71.0	7.5
16712	0.00	0.00	0.01	0.00	71.0	7.5
16713	0.01	0.00	0.00	0.00	71.0	7.5
16714	0.00	0.00	0.01	0.00	71.0	7.5

	rating
1	N/A
4	N/A
5	N/A
9	N/A
10	N/A
...	...
16710	N/A
16711	N/A
16712	N/A
16713	N/A
16714	N/A

[6766 rows x 11 columns]

Como se mencionó antes, en la columna ‘rating’ se encontraron miles de valores ausentes. Esto podría deberse a que este tipo de calificaciones no constaban en las fuentes virtuales donde se obtuvieron estos datos para algunos juegos.

Reemplacé dichos valores ausentes de la columna ‘rating’ por el marcador ‘N/A’. Asimismo, como los datos de esta columna son de tipo cadena, se constató la presencia de filas la abreviatura tbd, que asimismo son equivalentes a valores categóricos ausentes, por tanto también se reemplazaron con el marcador ‘N/A’.

```
[22]: #Control de filas duplicadas
print(df_games[df_games.duplicated()])
```

Empty DataFrame

Columns: [name, platform, year_of_release, genre, na_sales, eu_sales, jp_sales, other_sales, critic_score, user_score, rating]

Index: []

Comentario del revisor:

Como aprendiste en cursos anteriores, el análisis de registros duplicados es esencial en el análisis de datos para evitar que sesguen nuestros resultados.

No se encontraron filas duplicadas en esta tabla de datos.

```
[23]: #Cálculo de ventas totales de todas las regiones
df_games['total_sales']=df_games['na_sales']+df_games['eu_sales']+df_games['jp_sales']+df_games['other_sales']
print(df_games)
```

	name	platform	year_of_release	genre	\
0	Wii Sports	Wii	2006	Sports	
1	Super Mario Bros.	NES	1985	Platform	
2	Mario Kart Wii	Wii	2008	Racing	
3	Wii Sports Resort	Wii	2009	Sports	
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	
...	
16710	Samurai Warriors: Sanada Maru	PS3	2016	Action	
16711	LMA Manager 2007	X360	2006	Sports	
16712	Haitaka no Psychedelica	PSV	2016	Adventure	
16713	Spirits & Spells	GBA	2003	Platform	
16714	Winning Post 8 2016	PSV	2016	Simulation	

	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	\
0	41.36	28.96	3.77	8.45	76.0	8.0	
1	29.08	3.58	6.81	0.77	71.0	7.5	
2	15.68	12.76	3.79	3.29	82.0	8.3	
3	15.61	10.93	3.28	2.95	80.0	8.0	
4	11.27	8.89	10.22	1.00	71.0	7.5	
...	
16710	0.00	0.00	0.01	0.00	71.0	7.5	
16711	0.00	0.01	0.00	0.00	71.0	7.5	
16712	0.00	0.00	0.01	0.00	71.0	7.5	
16713	0.01	0.00	0.00	0.00	71.0	7.5	
16714	0.00	0.00	0.01	0.00	71.0	7.5	

	rating	total_sales
0	E	82.54
1	N/A	40.24
2	E	35.52
3	E	32.77
4	N/A	31.38
...
16710	N/A	0.01
16711	N/A	0.01
16712	N/A	0.01
16713	N/A	0.01
16714	N/A	0.01

[16715 rows x 12 columns]

Comentario del revisor:

Hola! Muy buen trabajo en la sección, ajustaste los nombres de las columnas a minúsculas con el

uso de la función `str.lower()`, cambiaste el tipo de variable de dos de las variables de la base de datos, consideraste ajustar los valores ausentes de las variables identificadas de score y muy buen trabajo con la suma de todas las ventas.

1.3 Análisis de datos

1.3.1 Juegos en diferentes años

```
[24]: #Conteo de títulos únicos
game_count_unique=df_games.groupby('year_of_release')['name'].nunique()
print(game_count_unique)
game_count_unique_1=game_count_unique.loc[1:]
game_count_unique_1.plot(kind='bar', title="Juegos lanzados por período",
    ↪xlabel='Años', ylabel='Juegos')
```

year_of_release	
0	232
1980	9
1981	46
1982	36
1983	17
1984	14
1985	14
1986	21
1987	16
1988	15
1989	17
1990	16
1991	40
1992	41
1993	60
1994	117
1995	215
1996	258
1997	282
1998	364
1999	326
2000	325
2001	428
2002	631
2003	546
2004	550
2005	657
2006	773
2007	884
2008	1039
2009	1001
2010	900

```

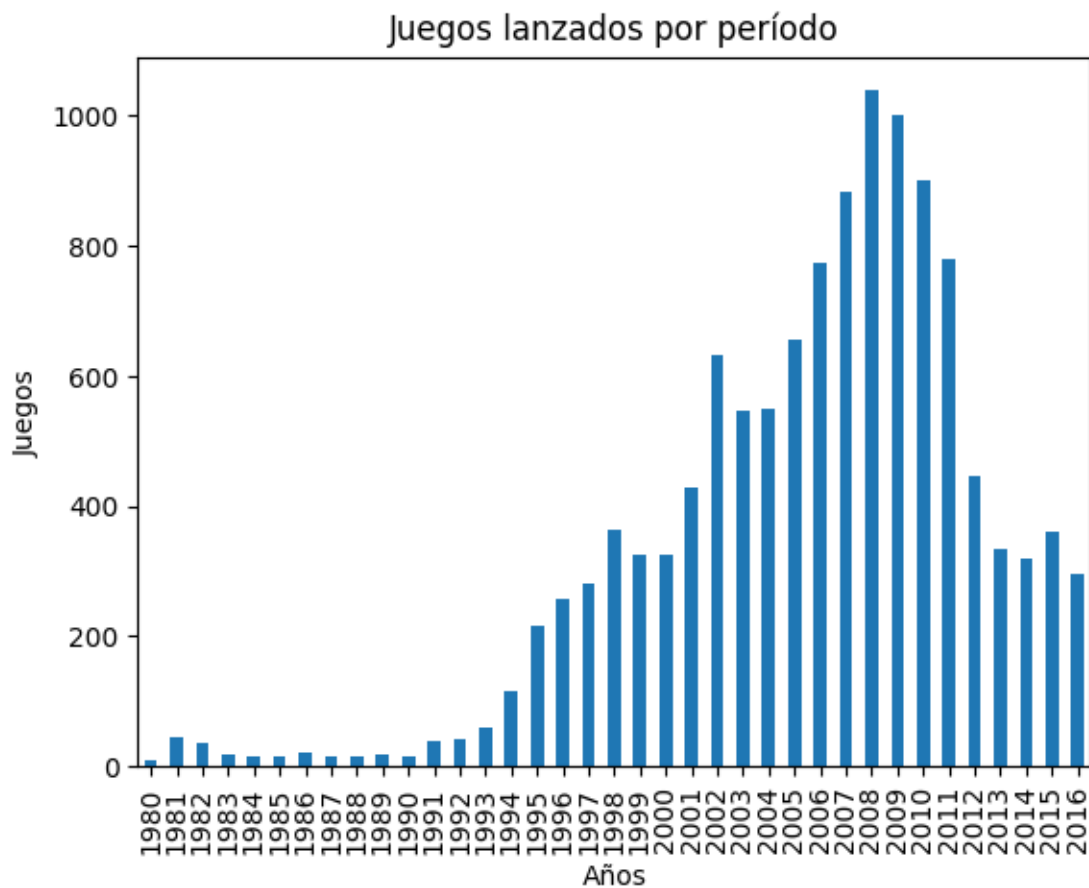
2011      779
2012      446
2013      335
2014      319
2015      362
2016      297
Name: name, dtype: int64

```

```

[24]: <AxesSubplot:title={'center':'Juegos lanzados por período'}, xlabel='Años',
      ylabel='Juegos'>

```



El número de juegos lanzados por año presenta variaciones significativas a lo largo de cada década. El número de juegos registrados en la década de los 80 no supera las dos cifras en cada año. En los años 90 se observa una tendencia creciente de lanzamientos en cada año, tendencia que continúa en los años 2000 hasta alcanzar un pico a finales de esta década. Interesantemente a partir del año 2010 se observa una caída abrupta y muy notable en el número de lanzamientos, mayor al 60% en el número de títulos durante el primer cuarto de esta última década, hasta estabilizarse en los años siguientes hasta el 2016. Esta caída podría deberse a una diversidad en el mercado del videojuego que no incluye a las plataformas convencionales que se consideraron en esta base de datos (como

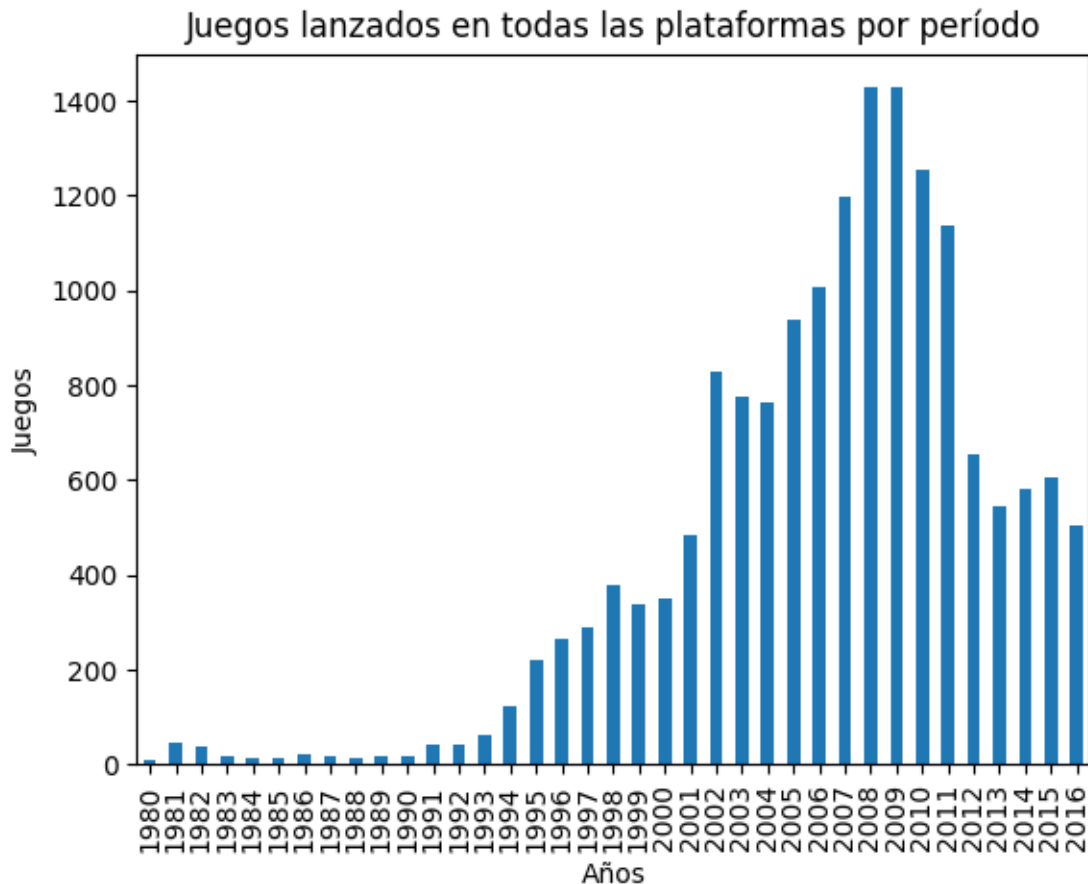
puede ser, por ejemplo, las de dispositivos móviles y tabletas). No obstante, la distribución de este gráfico sólo contempla títulos de juegos únicos, independientemente de las plataformas. Nótese además que se han excluido los juegos cuyo año de lanzamiento no estaba disponible en la tabla.

```
[25]: #Conteo de juegos lanzados en todas las plataformas
game_count=df_games.groupby('year_of_release')['name'].count()
print(game_count)
game_count_1=game_count.loc[1:]
game_count_1.plot(kind='bar', title="Juegos lanzados en todas las plataformas_
↳por período", xlabel='Años', ylabel='Juegos')
```

year_of_release	
0	269
1980	9
1981	46
1982	36
1983	17
1984	14
1985	14
1986	21
1987	16
1988	15
1989	17
1990	16
1991	41
1992	43
1993	62
1994	121
1995	219
1996	263
1997	289
1998	379
1999	338
2000	350
2001	482
2002	829
2003	775
2004	762
2005	939
2006	1006
2007	1197
2008	1427
2009	1426
2010	1255
2011	1136
2012	653
2013	544
2014	581

```
2015    606
2016    502
Name: name, dtype: int64
```

```
[25]: <AxesSubplot:title={'center':'Juegos lanzados en todas las plataformas por
período'}, xlabel='Años', ylabel='Juegos'>
```



En el último gráfico se han incluido los lanzamientos de juegos para todas las plataformas disponibles (así pues, un mismo título puede repetirse según el número de versiones liberadas). Sin embargo se observa que la tendencia es prácticamente la misma.

1.3.2 Ventas según plataforma

```
[26]: #Ventas totales de cada plataforma por año
df_games_a=df_games.query("year_of_release != 0")
pivot_platform=df_games_a.
    ↪pivot_table(index='year_of_release',columns='platform',
    ↪values='total_sales', aggfunc='sum')
pivot_platform.fillna(0.0, inplace=True)
```

```
pivot_platform.loc[:, '2600': 'PC']
```

```
[26]: platform      2600    3D0    3DS    DC      DS      GB      GBA      GC      GEN  \
year_of_release
1980          11.38    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1981          35.68    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1982          28.88    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1983           5.84    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1984           0.27    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1985           0.45    0.00    0.00    0.00     0.02     0.00     0.00     0.00     0.00
1986           0.67    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1987           1.94    0.00    0.00    0.00     0.00     0.00     0.00     0.00     0.00
1988           0.74    0.00    0.00    0.00     0.00     1.43     0.00     0.00     0.00
1989           0.63    0.00    0.00    0.00     0.00    64.97     0.00     0.00     0.00
1990           0.00    0.00    0.00    0.00     0.00     4.89     0.00     0.00     2.60
1991           0.00    0.00    0.00    0.00     0.00     5.57     0.00     0.00     4.33
1992           0.00    0.00    0.00    0.00     0.00    25.49     0.00     0.00    12.64
1993           0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00     3.07
1994           0.00    0.02    0.00    0.00     0.00    12.18     0.00     0.00     8.13
1995           0.00    0.08    0.00    0.00     0.00     3.60     0.00     0.00     0.00
1996           0.00    0.00    0.00    0.00     0.00    36.03     0.00     0.00     0.00
1997           0.00    0.00    0.00    0.00     0.00     6.37     0.00     0.00     0.00
1998           0.00    0.00    0.00    3.38     0.00    26.90     0.00     0.00     0.00
1999           0.00    0.00    0.00    5.16     0.00    38.00     0.00     0.00     0.00
2000           0.00    0.00    0.00    5.99     0.00    19.76     0.07     0.00     0.00
2001           0.00    0.00    0.00    1.07     0.00     9.24    61.53    26.34     0.00
2002           0.00    0.00    0.00    0.29     0.00     0.00    74.16    51.81     0.00
2003           0.00    0.00    0.00    0.00     0.00     0.00    56.67    50.61     0.00
2004           0.00    0.00    0.00    0.00    17.27     0.00    77.91    28.82     0.00
2005           0.00    0.00    0.00    0.00   130.14     0.00    33.86    27.62     0.00
2006           0.00    0.00    0.00    0.00   119.81     0.00     5.28    11.26     0.00
2007           0.00    0.00    0.00    0.02   146.94     0.00     3.40     0.27     0.00
2008           0.00    0.00    0.00    0.04   145.31     0.00     0.00     0.00     0.00
2009           0.00    0.00    0.00    0.00   119.54     0.00     0.00     0.00     0.00
2010           0.00    0.00    0.00    0.00    85.02     0.00     0.00     0.00     0.00
2011           0.00    0.00    63.20    0.00    26.18     0.00     0.00     0.00     0.00
2012           0.00    0.00    51.36    0.00    11.01     0.00     0.00     0.00     0.00
2013           0.00    0.00    56.57    0.00     1.54     0.00     0.00     0.00     0.00
2014           0.00    0.00    43.76    0.00     0.00     0.00     0.00     0.00     0.00
2015           0.00    0.00    27.78    0.00     0.00     0.00     0.00     0.00     0.00
2016           0.00    0.00    15.14    0.00     0.00     0.00     0.00     0.00     0.00

platform      GG      N64      NES      NG      PC
year_of_release
1980          0.00     0.00     0.00     0.00     0.00
1981          0.00     0.00     0.00     0.00     0.00
1982          0.00     0.00     0.00     0.00     0.00
```


1983	0.00	0.00	10.96	0.00	0.00
1984	0.00	0.00	50.08	0.00	0.00
1985	0.00	0.00	53.44	0.00	0.04
1986	0.00	0.00	36.41	0.00	0.00
1987	0.00	0.00	19.76	0.00	0.00
1988	0.00	0.00	45.01	0.00	0.03
1989	0.00	0.00	7.85	0.00	0.00
1990	0.00	0.00	15.73	0.00	0.00
1991	0.00	0.00	6.11	0.00	0.00
1992	0.04	0.00	1.98	0.00	3.03
1993	0.00	0.00	3.61	0.21	0.00
1994	0.00	0.00	0.11	0.80	12.87
1995	0.00	0.00	0.00	0.33	4.22
1996	0.00	34.10	0.00	0.10	10.58
1997	0.00	39.50	0.00	0.00	11.27
1998	0.00	49.24	0.00	0.00	3.26
1999	0.00	57.87	0.00	0.00	4.74
2000	0.00	33.97	0.00	0.00	4.66
2001	0.00	3.25	0.00	0.00	5.47
2002	0.00	0.08	0.00	0.00	8.57
2003	0.00	0.00	0.00	0.00	8.84
2004	0.00	0.00	0.00	0.00	10.39
2005	0.00	0.00	0.00	0.00	4.37
2006	0.00	0.00	0.00	0.00	2.85
2007	0.00	0.00	0.00	0.00	9.28
2008	0.00	0.00	0.00	0.00	12.42
2009	0.00	0.00	0.00	0.00	16.91
2010	0.00	0.00	0.00	0.00	24.28
2011	0.00	0.00	0.00	0.00	35.03
2012	0.00	0.00	0.00	0.00	23.22
2013	0.00	0.00	0.00	0.00	12.38
2014	0.00	0.00	0.00	0.00	13.28
2015	0.00	0.00	0.00	0.00	8.52
2016	0.00	0.00	0.00	0.00	5.25

```
[27]: pivot_platform.loc[:, 'PCFX': 'X0ne']
```

```
[27]: platform      PCFX      PS      PS2      PS3      PS4      PSP      PSV      SAT  \
year_of_release
1980      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1981      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1982      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1983      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1984      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1985      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1986      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
1987      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

1988	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1989	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1990	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1991	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1992	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1993	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1994	0.00	6.03	0.00	0.00	0.00	0.00	0.00	3.65
1995	0.00	35.96	0.00	0.00	0.00	0.00	0.00	11.57
1996	0.03	94.70	0.00	0.00	0.00	0.00	0.00	7.69
1997	0.00	136.17	0.00	0.00	0.00	0.00	0.00	6.77
1998	0.00	169.49	0.00	0.00	0.00	0.00	0.00	3.82
1999	0.00	144.53	0.00	0.00	0.00	0.00	0.00	0.09
2000	0.00	96.37	39.17	0.00	0.00	0.00	0.00	0.00
2001	0.00	35.59	166.43	0.00	0.00	0.00	0.00	0.00
2002	0.00	6.67	205.38	0.00	0.00	0.00	0.00	0.00
2003	0.00	2.07	184.31	0.00	0.00	0.00	0.00	0.00
2004	0.00	0.00	211.81	0.00	0.00	7.06	0.00	0.00
2005	0.00	0.00	160.66	0.00	0.00	43.84	0.00	0.00
2006	0.00	0.00	103.42	20.96	0.00	55.32	0.00	0.00
2007	0.00	0.00	75.99	73.19	0.00	46.93	0.00	0.00
2008	0.00	0.00	53.90	118.52	0.00	34.55	0.00	0.00
2009	0.00	0.00	26.40	130.93	0.00	37.78	0.00	0.00
2010	0.00	0.00	5.64	142.17	0.00	35.04	0.00	0.00
2011	0.00	0.00	0.45	156.78	0.00	17.82	4.63	0.00
2012	0.00	0.00	0.00	107.36	0.00	7.69	16.19	0.00
2013	0.00	0.00	0.00	113.25	25.99	3.14	10.59	0.00
2014	0.00	0.00	0.00	47.76	100.00	0.24	11.90	0.00
2015	0.00	0.00	0.00	16.82	118.90	0.12	6.25	0.00
2016	0.00	0.00	0.00	3.60	69.25	0.00	4.25	0.00

platform	SCD	SNES	TG16	WS	Wii	WiiU	X360	XB	XOne
year_of_release									
1980	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1981	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1982	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1983	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1984	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1985	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1986	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1987	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1988	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1989	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1990	0.00	26.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1991	0.00	16.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1992	0.00	32.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1993	1.50	40.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1994	0.36	35.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00

1995	0.00	32.20	0.16	0.00	0.00	0.00	0.00	0.00	0.00
1996	0.00	15.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1997	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1998	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1999	0.00	0.26	0.00	0.46	0.00	0.00	0.00	0.00	0.00
2000	0.00	0.00	0.00	0.68	0.00	0.00	0.00	0.99	0.00
2001	0.00	0.00	0.00	0.28	0.00	0.00	0.00	22.26	0.00
2002	0.00	0.00	0.00	0.00	0.00	0.00	0.00	48.01	0.00
2003	0.00	0.00	0.00	0.00	0.00	0.00	0.00	55.04	0.00
2004	0.00	0.00	0.00	0.00	0.00	0.00	0.00	65.42	0.00
2005	0.00	0.00	0.00	0.00	0.00	0.00	8.25	49.08	0.00
2006	0.00	0.00	0.00	0.00	137.15	0.00	51.62	10.04	0.00
2007	0.00	0.00	0.00	0.00	152.77	0.00	95.41	0.55	0.00
2008	0.00	0.00	0.00	0.00	171.32	0.00	135.26	0.18	0.00
2009	0.00	0.00	0.00	0.00	206.97	0.00	120.29	0.00	0.00
2010	0.00	0.00	0.00	0.00	127.95	0.00	170.03	0.00	0.00
2011	0.00	0.00	0.00	0.00	59.65	0.00	143.84	0.00	0.00
2012	0.00	0.00	0.00	0.00	21.71	17.56	99.74	0.00	0.00
2013	0.00	0.00	0.00	0.00	8.59	21.65	88.58	0.00	18.96
2014	0.00	0.00	0.00	0.00	3.75	22.03	34.74	0.00	54.07
2015	0.00	0.00	0.00	0.00	1.14	16.35	11.96	0.00	60.14
2016	0.00	0.00	0.00	0.00	0.18	4.60	1.52	0.00	26.15

```
[28]: #Ventas totales para cada plataforma ordenadas de mayor a menor
plat_sales=pivot_platform.sum()
plat_sales.sort_values(ascending=False)
```

```
[28]: platform
PS2      1233.56
X360      961.24
PS3       931.34
Wii       891.18
DS        802.78
PS        727.58
PS4       314.14
GBA       312.88
PSP       289.53
3DS       257.81
PC        255.76
GB        254.43
XB        251.57
NES       251.05
N64       218.01
SNES      200.04
GC        196.73
XOne      159.32
2600       86.48
```

```

WiiU      82.19
PSV       53.81
SAT       33.59
GEN       30.77
DC        15.95
SCD       1.86
NG        1.44
WS        1.42
TG16      0.16
3DO       0.10
GG        0.04
PCFX      0.03
dtype: float64

```

Si consideramos volumen de ventas totales registrado hasta el 2016, las plataformas con los números más altos son: PS2, X360, PS3, Wii, DS y PS; de éstas sólo la X360 y las PS3 estuvieron vigentes en ventas hasta el último año. En base a la tabla se puede apuntar que las plataformas con valores de ventas de decenas de millones hasta el año 2016 son: Nintendo 3DS, PS4 y XOne. Asimismo las plataformas que registraron ventas de al menos un millón son: PC, X360, PS3, WiiU y PSV. En todas las plataformas que continúan vigentes, sin excepción, se observa una tendencia a la baja.

```

[29]: #Tiempo de vida de las plataformas que solían ser populares
top_plat=pivot_platform.loc[:,['PS2', 'Wii', 'DS', 'PS', 'GBA', 'GC', 'GB',
    ↳ 'N64', 'SNES', 'XB', 'NES', 'PSP']]
top_plat_count=top_plat[top_plat>0.00].count()
print(top_plat_count)
life_ex2=top_plat_count.mean()
print("\n\n", "Tiempo de vida promedio de las plataformas con más ventas:
    ↳ ", life_ex2.round(), "años")

```

```

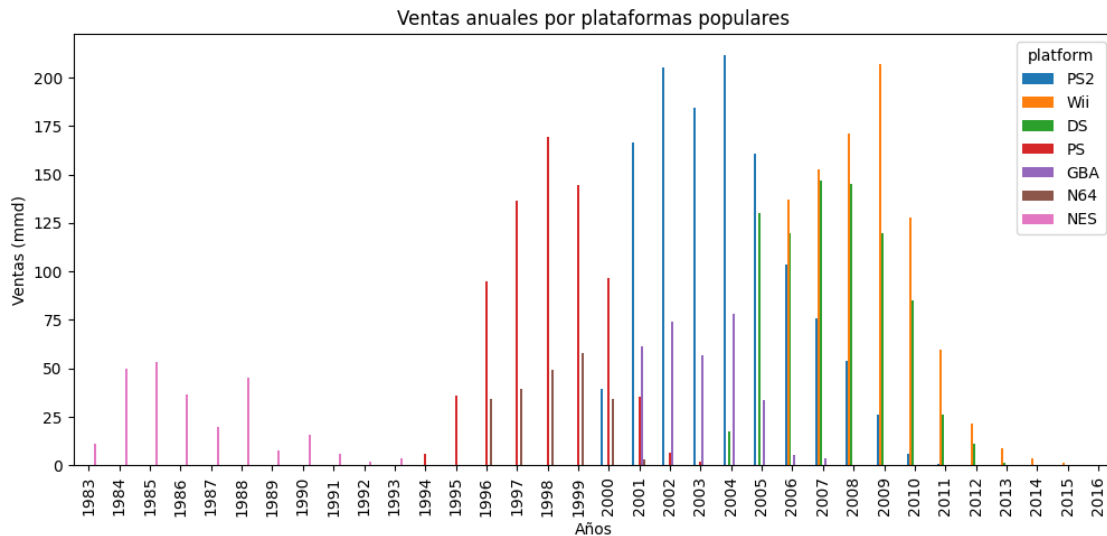
platform
PS2      12
Wii      11
DS       11
PS       10
GBA       8
GC        7
GB       13
N64       7
SNES     10
XB        9
NES      12
PSP      12
dtype: int64

```

Tiempo de vida promedio de las plataformas con más ventas: 10.0 años

```
[30]: #Distribución de ventas de plataformas con las mayores ventas totales basada en
      ↪ los datos anuales
      pivot_platform.loc[1983:,['PS2', 'Wii', 'DS', 'PS', 'GBA', 'N64', 'NES']].
      ↪ plot(kind='bar', title="Ventas anuales por plataformas populares",
      ↪ figsize=[12,5], xlabel='Años', ylabel='Ventas (mmd)')
```

```
[30]: <AxesSubplot:title={'center':'Ventas anuales por plataformas populares'},
      xlabel='Años', ylabel='Ventas (mmd)'
```



Comentario del revisor

Gran trabajo con el desarrollo de la gráfica. Solamente te recomendaría que intentes reducir el periodo dado que el periodo actual puede resultar muy extenso. Recordemos que con base en la información la popularidad la alcanzan en los primeros años. Es por ello que con un periodo largo puede no ser posible analizar de forma clara las plataformas y videojuegos actualizados y podrías estar analizando los desactualizados. Es por ello y dado que buscamos responder qué vender en 2017 te recomendaría graficar un periodo que no supere los 5 años.

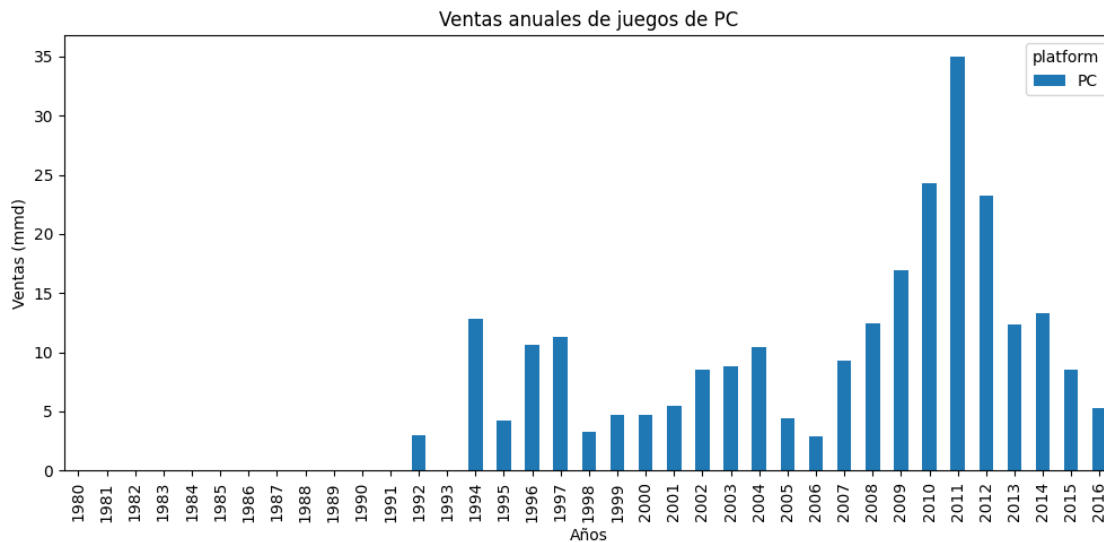
Considerando las plataformas más populares (\geq a 100 millones en ventas totales) y que ahora no tienen ventas se podría decir que en promedio el tiempo de vigencia de una plataforma en el mercado es de alrededor de una década. Una nueva plataforma tardaría alrededor de 5 años en llegar a su cenit de ventas y otros 5 años en decaer hasta desaparecer del mercado.

Es menester apuntar que la PC como plataforma de juegos es un caso aparte ('outlier'), por cuanto para la estimación del tiempo de vida de las plataformas sólo se consideró a las consolas.

1.3.3 Plataformas potencialmente rentables

```
[31]: #Distribución de ventas de la PC de acuerdo a los datos anuales
pivot_platform.loc[:,['PC']].plot(kind='bar', title="Ventas anuales de juegos de PC",
    figsize=[12,5], xlabel='Años', ylabel='Ventas (mmd)')
```

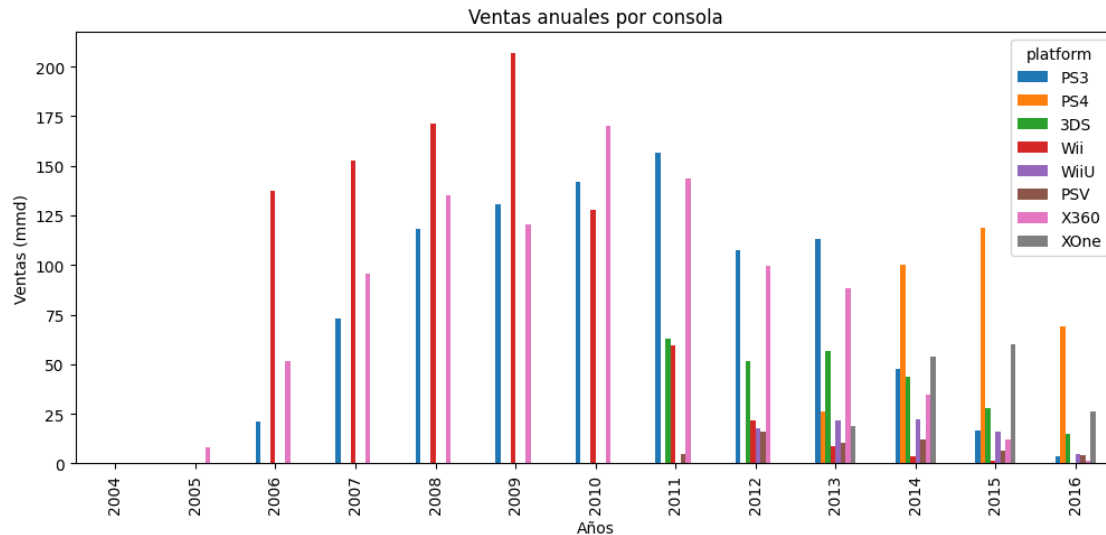
```
[31]: <AxesSubplot:title={'center':'Ventas anuales de juegos de PC'}, xlabel='Años',
    ylabel='Ventas (mmd)'\>
```



Aunque su distribución en el mercado sigue la tendencias globales decrecientes en ventas, la PC se ha mantenido vigente en ventas por más de dos décadas, por lo que se puede suponer que los juegos lanzados en esta plataforma van a seguir manteniendo una tajada de mercado en los años subsiguientes. Incluso su mercado en el último año es más fuerte en ventas que la de consolas recientemente populares como la PS3, PSP, WiiU y X360, ergo se la puede considerar como una plataforma que seguirá siendo rentable.

```
[32]: #Distribución de ventas de consolas potencialmente rentables.
pivot_platform.loc[2004:,['PS3','PS4','3DS','Wii','WiiU','PSV','X360',
    'XOne']].plot(kind='bar', title="Ventas anuales por consola",
    figsize=[12,5], xlabel='Años', ylabel='Ventas (mmd)')
```

```
[32]: <AxesSubplot:title={'center':'Ventas anuales por consola'}, xlabel='Años',
    ylabel='Ventas (mmd)'\>
```



Tal como se ve en el gráfico, todas las plataformas reducen sus ventas en los últimos años. Incluso las consolas más novedosas que están liderando el mercado (como la PS4 y la XOne) parecen sugerir un tiempo de vida más corto que el normal (de 10 años). No obstante, hay varias que seguirán vendiendo millones en el próximo año, a saber: en primer lugar, de lejos la PS4; en segundo y tercero, la XOne y el Nintendo 3DS respectivamente; luego podríamos considerar a la WiiU, la PSV e incluso juegos de PS3, que todavía podrían mantener ventas de al menos 1 millón para el siguiente año. La consola Wii, otrora considerada la más revolucionaria de todas, estaría siendo reemplazada por la WiiU y su retrocompatibilidad; por eso y más aun, considerando el mercado anémico de su último año, sería redundante considerarla.

1.3.4 Datos para modelos

Para estimar ventas para el año 2017 de una plataforma se podrían tomar los datos partiendo desde el cénit de ventas. Por ejemplo, en el caso de la plataforma 3DS se tomarían los datos desde el 2011 para el cálculo.

[33]: *#Ejemplo de modelo de regresión lineal simple para ventas del siguiente año*

```
DS3y=pivot_platform.loc[2011:,'3DS']
DS3x=[1, 2, 3, 4, 5, 6]

n=len(DS3x) #Longitud de datos
Sx=sum(DS3x) #Suma de valores de x
SSx=np.dot(DS3x,DS3x) #Suma de cuadrados de x
Sy=sum(DS3y) #Suma de valores de y
SSxy=np.dot(DS3x,DS3y) #Suma de productos de xy
mx= np.mean(DS3x) #Promedio de x
my= np.mean(DS3y) #Promedio de y
```

```

b=(n*SSxy-Sx*Sy)/(n*SSx-Sx*Sx) #Coeficientes de línea de regresión
a=my-b*mx

print('Sx=',Sx,'SSx=',SSx,'Sy=',Sy,'SSxy=',SSxy,'mx=',mx,'my=',my)
print('a=',a,'b=',b) # Coeficientes de regresión

print('\n','El modelo de regresión es: y =',a,'+', '(' ,b,') ','* x')

```

Sx= 21 SSx= 91 Sy= 257.81000000000006 SSxy= 740.4100000000001 mx= 3.5 my=
 42.96833333333334
 a= 75.35333333333335 b= -9.252857142857145

El modelo de regresión es: $y = 75.35333333333335 + (-9.252857142857145) * x$

```

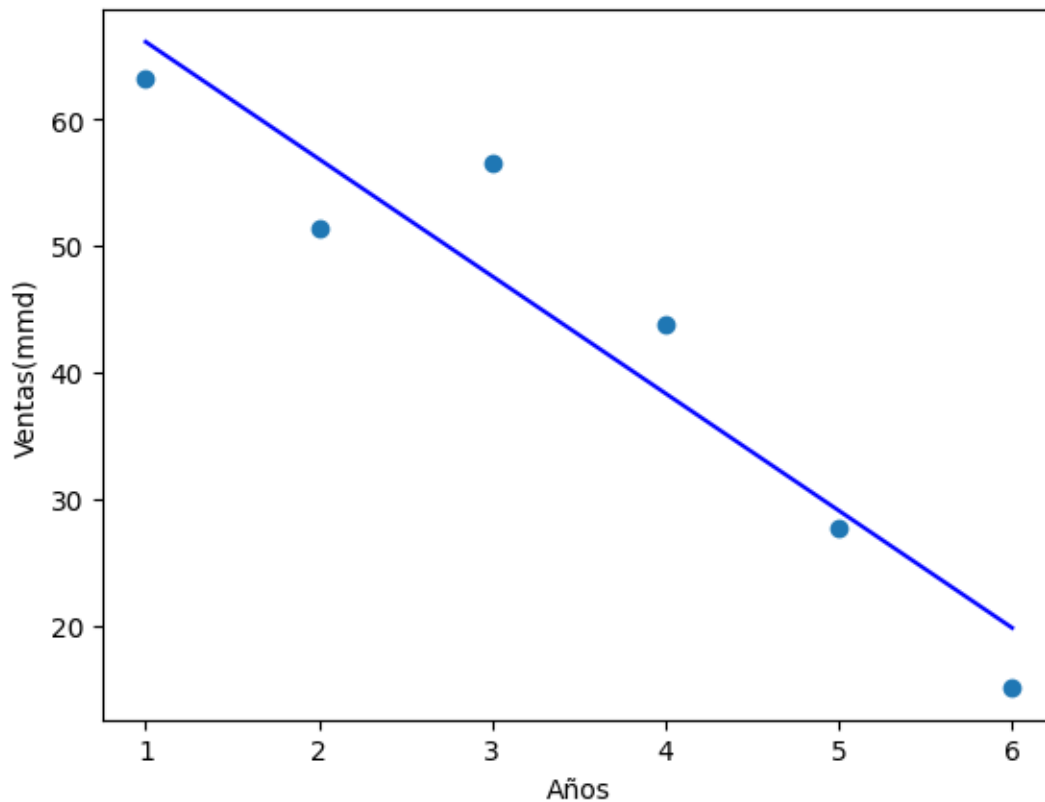
[34]: #Gráfico de ventas totales de plataforma 3DS
plt.scatter(DS3x,DS3y)
plt.plot(DS3x, a+b*np.array(DS3x), '-b')
plt.xlabel('Años')
plt.ylabel('Ventas(mmd)')

```

```

[34]: Text(0, 0.5, 'Ventas(mmd)')

```




```
[35]: #Cálculo de R2
ypred=a+b*np.array(DS3x)
res=DS3y-ypred

SSres=np.dot(res,res) #Suma de cuadrados de los residuos
SSyy=np.dot(DS3y,DS3y)-n*my*my #Suma de cuadrados de y
R2=1-SSres/SSyy #R2, coeficiente de determinación
print(R2)
```

0.8969123842181584

El modelo de regresión se ajustaría bien en el 89.7% de los casos.

```
[36]: #Ejemplo de proyección de ventas
def pred_3DS_2017(year=7):
    result = a+b*year
    return result
print(pred_3DS_2017())
```

10.583333333333334

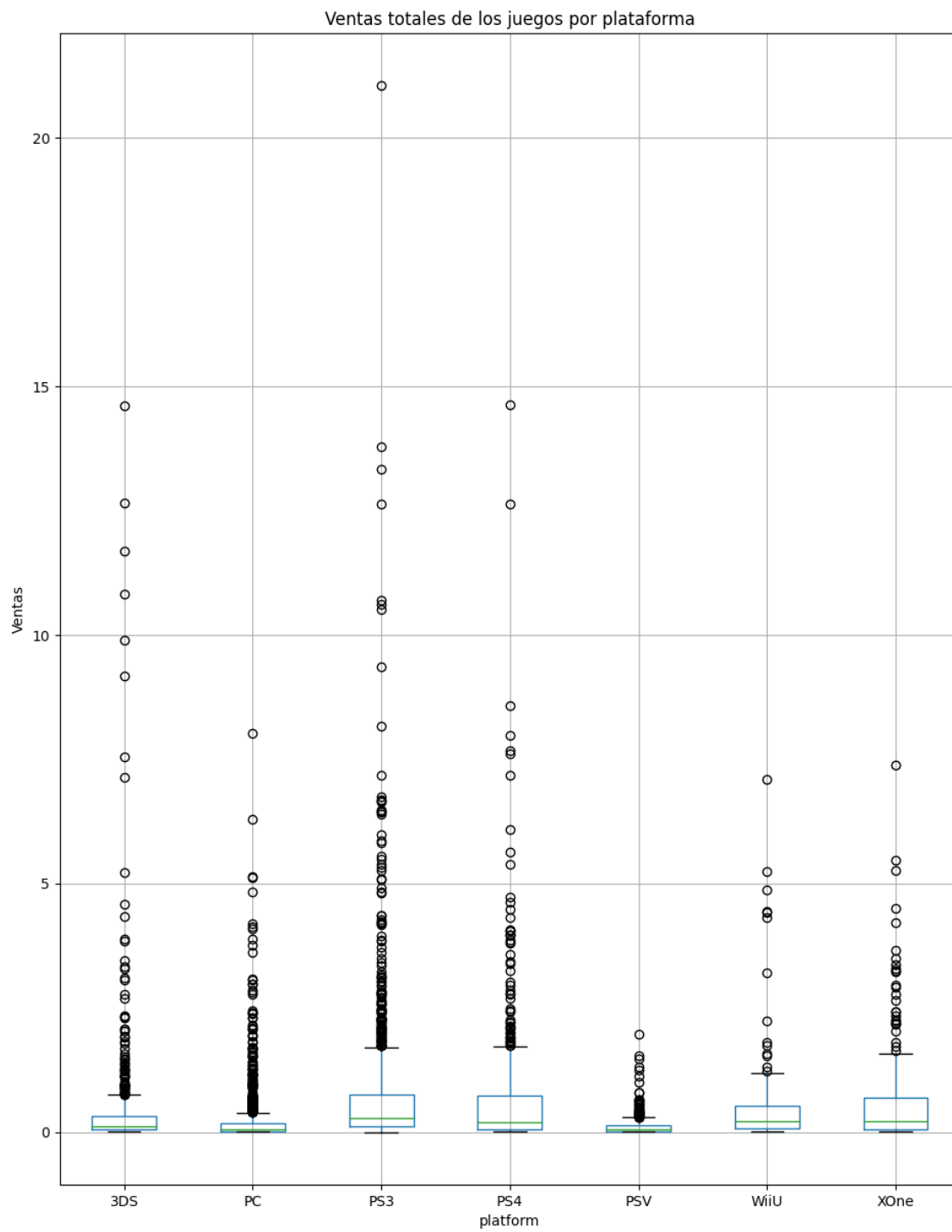
De acuerdo a este modelo para el 2017 se esperarían 10.6 mmd en ventas totales para la plataforma 3DS.

1.3.5 Diferencias en ventas por plataforma

```
[37]: #Diagrama de caja para ventas globales de todos los juegos por plataforma
df_games_b=df_games.query("platform ==_
↳ ['3DS','PC','PS3','PS4','WiiU','PSV','XOne']")
box=df_games_b.boxplot(column="total_sales", by="platform", figsize=(11,15))
box.set_ylabel("Ventas")
box.set_title("Ventas totales de los juegos por plataforma")
```

```
[37]: Text(0.5, 1.0, 'Ventas totales de los juegos por plataforma')
```

Boxplot grouped by platform



Comentario del revisor

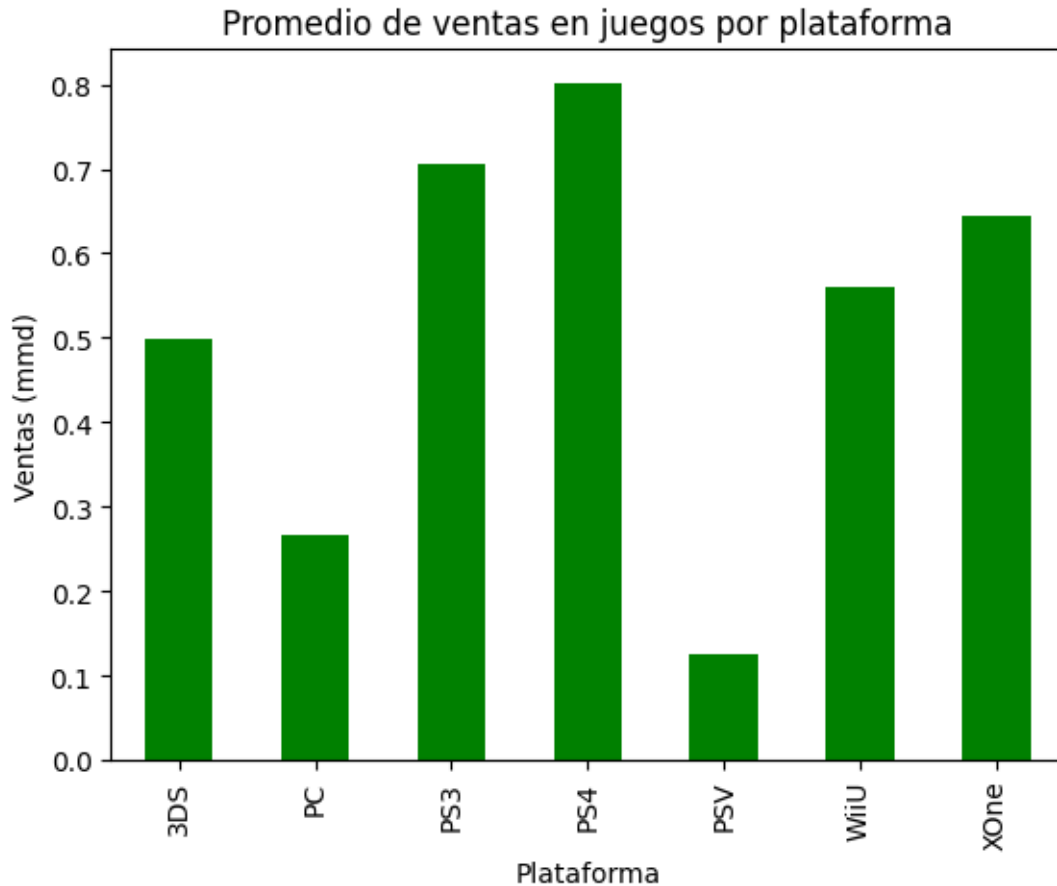
Recuerda que los graficos de box-plot nos ayudan a identificar outliers que posteriormente en el desarrollo de las pruebas de hipótesis nos servirían eliminar para que no puedan sesgar nuestros resultados.

En los diagramas de cajas de estas plataformas se puede apreciar que la mayor parte de juegos no superaron el millón de dólares en ventas. Las medianas en esta distribución de ventas son relativamente similares y son más bien atípicos los juegos que realmente han marcado hitos en el mercado. En esa línea, las consolas con los juegos más revolucionarios en ventas hasta el último año son: la PS3 (en primer lugar, con un juego superando los 20 millones en ventas totales), la PS4, la 3DS y en menor medida, por ser todavía emergentes, la WiiU y la XOne. Es interesante que la tradicional PC siempre ha mantenido una tajada de ventas relativamente modesta en comparación con las consolas. El alcance limitado que se refleja en la PSV se entiende por las limitaciones inherentes a esta plataforma.

Hay que considerar que estos valores atípicos pueden sesgar de forma importante los promedios.

```
[38]: #Promedios de ventas por plataforma
df_games_b_grp=df_games_b.groupby('platform')
df_games_b_grp['total_sales'].mean().plot(kind='bar', title="Promedio de ventas_
    ↪en juegos por plataforma",
                                           xlabel='Plataforma', ylabel='Ventas_
    ↪(mmd)', color='green')
```

```
[38]: <AxesSubplot:title={'center': 'Promedio de ventas en juegos por plataforma'},
      xlabel='Plataforma', ylabel='Ventas (mmd)'>
```



El cálculo de ventas promedio por plataforma me sugiere una relación entre esta medida de tendencia central y el tamaño de las cajas en el diagrama. El promedio de ventas en juego más alto es el de la nueva PS4, seguida de la PS3 y el XOne. Aunque la 3DS lanzó juegos ganadores más exitosos que la XOne y la WiiU, su promedio de ventas globales es más bajo que estas últimas.

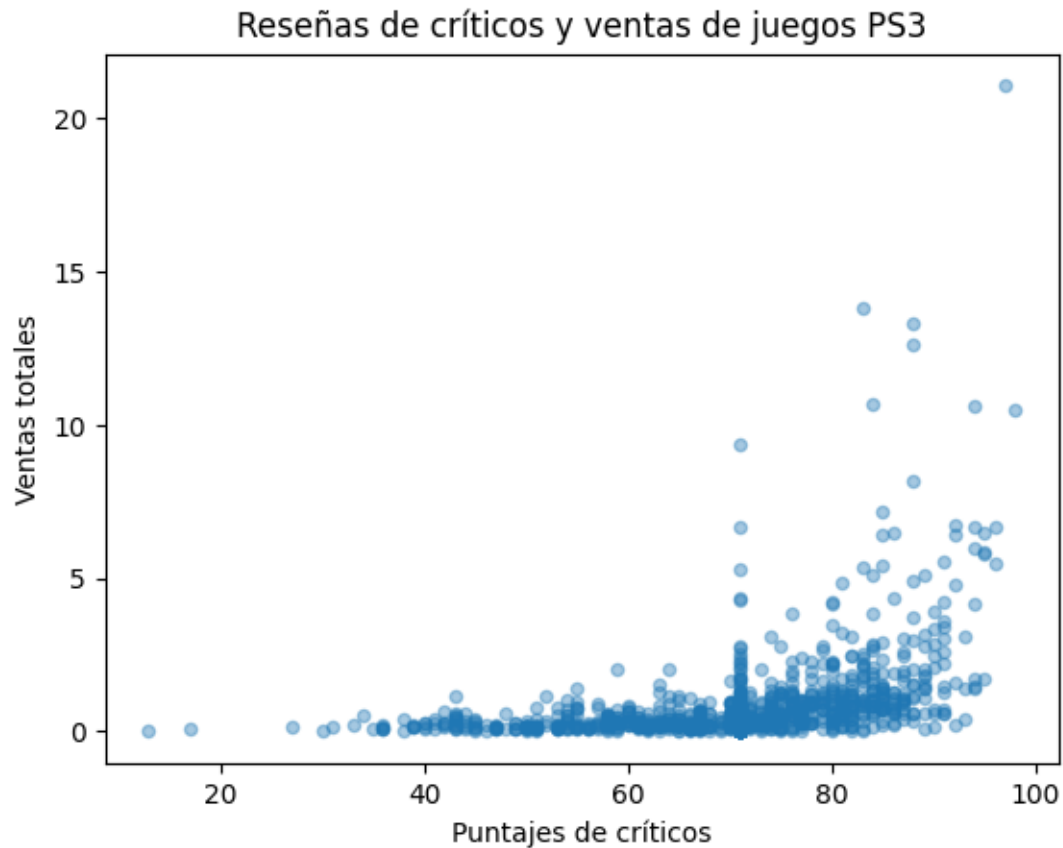
1.3.6 Relación entre reseñas y ventas

Para estudiar si las reseñas de usuarios y críticos afectan las ventas elegí como plataforma popular la PS3.

```
[39]: #Correlación entre reseñas de críticos y ventas
df_games_ps3=df_games.query("platform == 'PS3' and critic_score !=0.0")
    ↪#Descartamos los valores ausentes
df_games_ps3.sort_values('critic_score').plot(kind='scatter', title='Reseñas de
    ↪críticos y ventas de juegos PS3', xlabel='Puntajes de críticos',
    ↪ylabel='Ventas totales',
                                     x='critic_score',
    ↪y='total_sales', alpha=0.4)
```

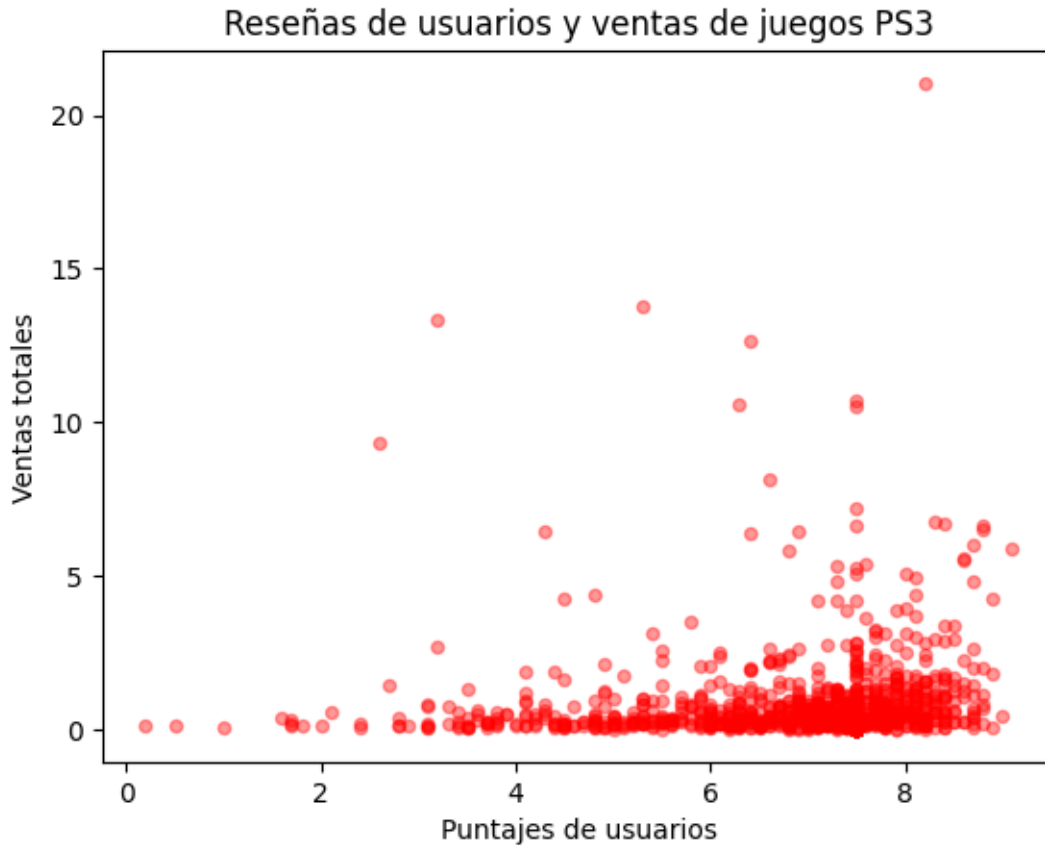
```
print('Correlación entre puntajes de críticos y ventas para juegos de PS3:
↪',df_games_ps3['total_sales'].corr(df_games_ps3['critic_score']))
```

Correlación entre puntajes de críticos y ventas para juegos de PS3:
0.3974129609158125



```
[40]: #Correlación entre reseñas de usuarios y ventas
df_games_ps3=df_games.query("platform == 'PS3' and user_score !=0.0")
↪#Descartamos valores ausentes
df_games_ps3.sort_values('user_score').plot(kind='scatter', title='Reseñas de
↪usuarios y ventas de juegos PS3', xlabel='Puntajes de usuarios',
↪ylabel='Ventas totales',
x='user_score', y='total_sales',
↪color='red', alpha=0.4)
print('Correlación entre reseñas de usuarios y ventas para juegos de PS3:
↪',df_games_ps3['total_sales'].corr(df_games_ps3['user_score']))
```

Correlación entre reseñas de usuarios y ventas para juegos de PS3:
0.041004026504435874



Comentario del revisor [](#)

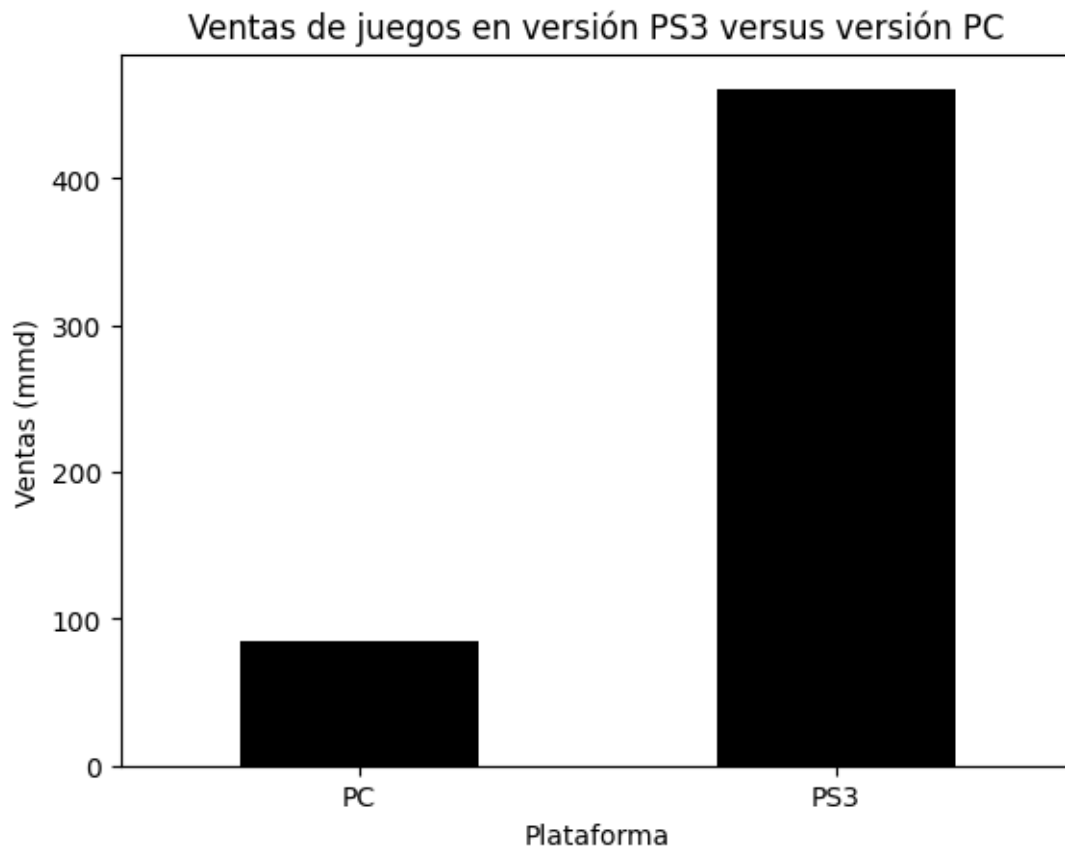
Muy buen trabajo con el análisis de la relación entre las scores y las ventas.

En el caso de los juegos de PS3 se observa una correlación positiva modesta ($r=0.4$) entre los puntajes de reseñas de críticos y las ventas, mientras los puntajes de reseñas de usuarios arrojaron una correlación muy débil, casi nula ($r=0.04$). En base a esto se puede concluir que es la crítica profesional la que tiene más impacto en las ventas.

1.3.7 Ventas de los mismos juegos en diferentes plataformas

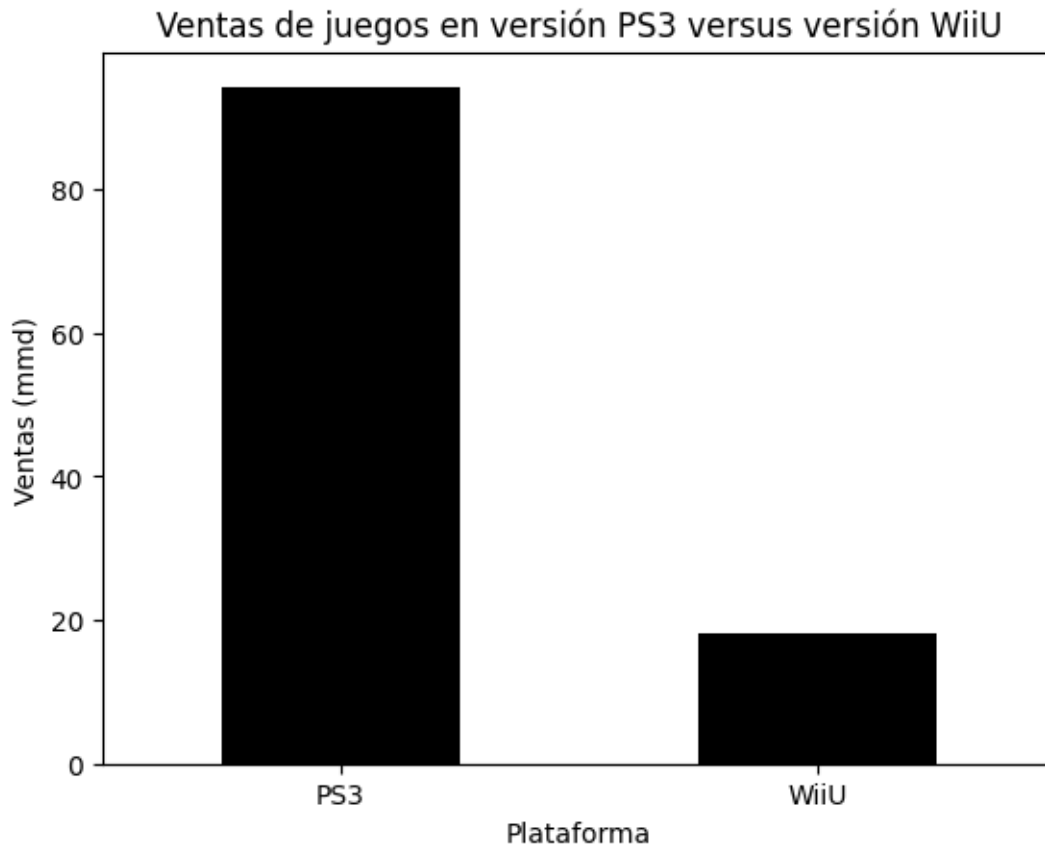
```
[41]: #Comparación de ventas de juegos en versiones de PC y PS3
df_games_plat_pc_ps3=df_games.query("platform=='PC','PS3'")
df_games_plat_1=df_games_plat_pc_ps3.query("name.duplicated(keep=False) ==_
↳True")
grp_plat_pc_ps3=df_games_plat_1.groupby(['platform'])['total_sales'].sum()
grp_plat_pc_ps3.plot(kind='bar', title="Ventas de juegos en versión PS3 versus_
↳versión PC",
xlabel='Plataforma', ylabel='Ventas_
↳(mmd)', color='black',rot=0)
```

```
[41]: <AxesSubplot:title={'center':'Ventas de juegos en versión PS3 versus versión PC'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



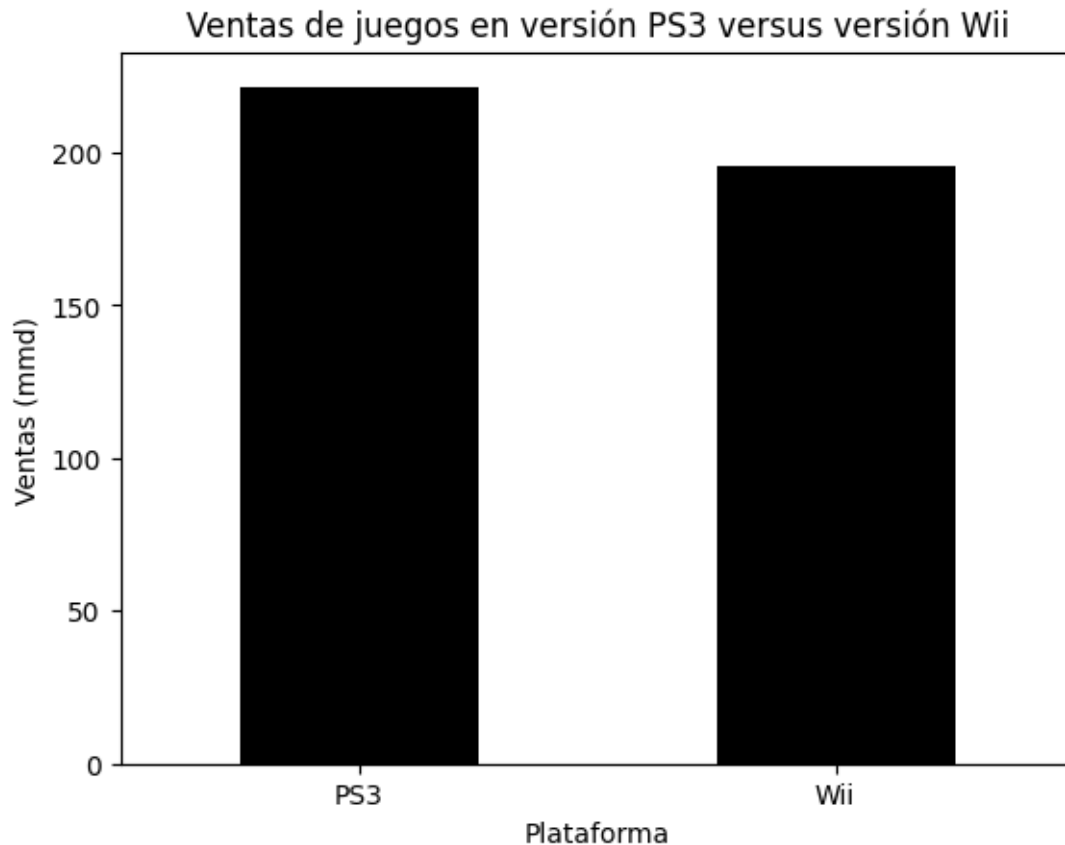
```
[42]: #Comparación de ventas de juegos en versiones de PS3 y WiiU
df_games_plat_wiiu_ps3=df_games.query("platform==['WiiU','PS3']")
df_games_plat_2=df_games_plat_wiiu_ps3.query("name.duplicated(keep=False) ==_
↳True")
df_games_plat_2.groupby(['platform'])['total_sales'].sum().plot(kind='bar',_
↳title="Ventas de juegos en versión PS3 versus versión WiiU",
                                xlabel='Plataforma', ylabel='Ventas_
↳(mmd)', color='black',rot=0)
```

```
[42]: <AxesSubplot:title={'center':'Ventas de juegos en versión PS3 versus versión WiiU'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



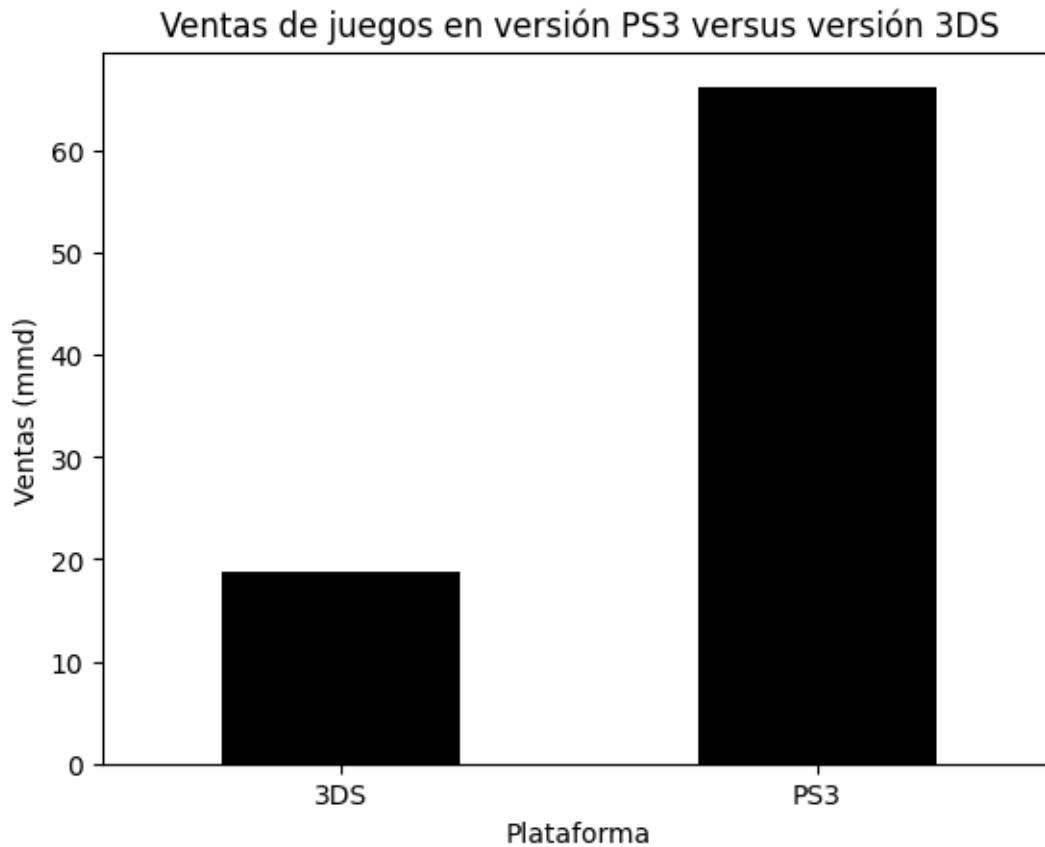
```
[43]: #Comparación de ventas de juegos en versiones de PS3 y Wii
df_games_plat_wii_ps3=df_games.query("platform==['Wii','PS3']")
df_games_plat_3=df_games_plat_wii_ps3.query("name.duplicated(keep=False) ==
↳True")
df_games_plat_3.groupby(['platform'])['total_sales'].sum().plot(kind='bar',
↳title="Ventas de juegos en versión PS3 versus versión Wii",
xlabel='Plataforma', ylabel='Ventas
↳(mmd)', color='black',rot=0)
```

```
[43]: <AxesSubplot:title={'center':'Ventas de juegos en versión PS3 versus versión
Wii'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```

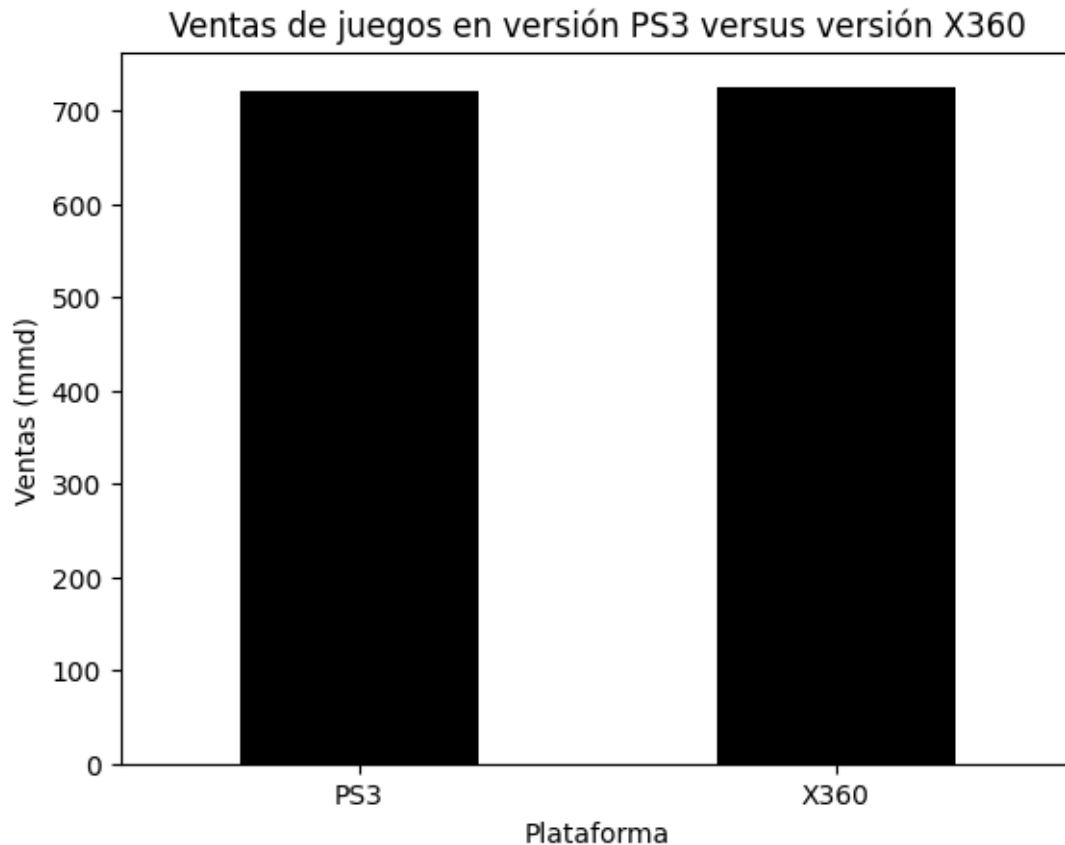
```
[44]: #Comparación de ventas de juegos en versiones de PS3 y 3DS
df_games_plat_3ds_ps3=df_games.query("platform=='3DS','PS3'")
df_games_plat_4=df_games_plat_3ds_ps3.query("name.duplicated(keep=False) ==
↳True")
df_games_plat_4.groupby(['platform'])['total_sales'].sum().plot(kind='bar',
↳title="Ventas de juegos en versión PS3 versus versión 3DS",
xlabel='Plataforma', ylabel='Ventas_
↳(mmd)', color='black',rot=0)
```

```
[44]: <AxesSubplot:title={'center':'Ventas de juegos en versión PS3 versus versión
3DS'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



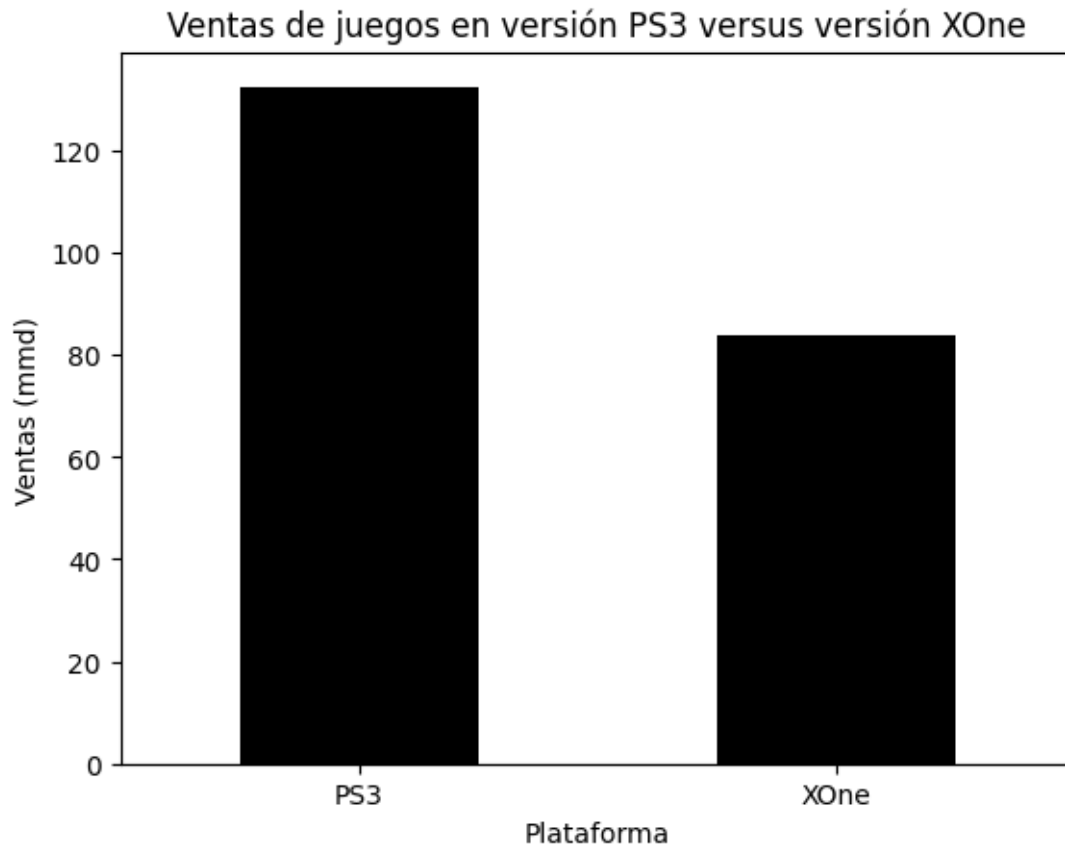
```
[45]: #Comparación de ventas de juegos en versiones de PS3 y X360
df_games_plat_x360_ps3=df_games.query("platform==['X360','PS3']")
df_games_plat_5=df_games_plat_x360_ps3.query("name.duplicated(keep=False) ==
↳True")
df_games_plat_5.groupby(['platform'])['total_sales'].sum().plot(kind='bar',
↳title="Ventas de juegos en versión PS3 versus versión X360",
xlabel='Plataforma', ylabel='Ventas_
↳(mmd)', color='black',rot=0)
```

```
[45]: <AxesSubplot:title={'center':'Ventas de juegos en versión PS3 versus versión
X360'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



```
[46]: #Comparación de ventas de juegos en versiones de PS3 y XOne
df_games_plat_xone_ps3=df_games.query("platform=='XOne','PS3'")
df_games_plat_6=df_games_plat_xone_ps3.query("name.duplicated(keep=False) ==
↳True")
df_games_plat_6.groupby(['platform'])['total_sales'].sum().plot(kind='bar',
↳title="Ventas de juegos en versión PS3 versus versión XOne",
xlabel='Plataforma', ylabel='Ventas_
↳(mmd)', color='black',rot=0)
```

```
[46]: <AxesSubplot:title={'center':'Ventas de juegos en versión PS3 versus versión
XOne'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



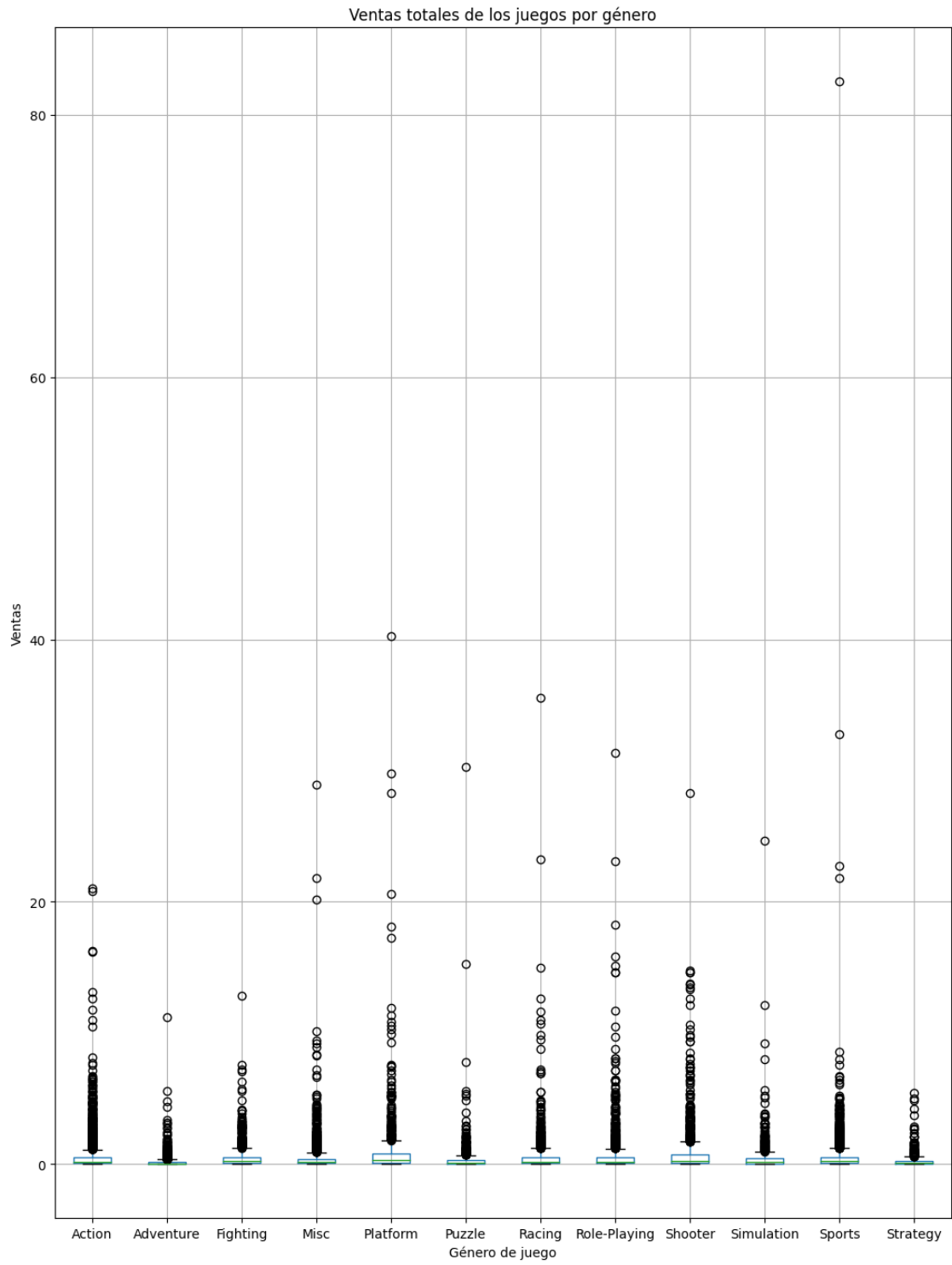
Al comparar las ventas de los mismos juegos en diferentes plataformas de la competencia, en este caso escogiendo el PS3 como plataforma popular, se puede apreciar que el posicionamiento de ventas sigue el mismo patrón que el de las ventas globales de las plataformas (donde se observa que las ventas de la X360 fueron por poco más altas que las del PS3).

1.3.8 Distribución de ventas por género de juego

```
[47]: #Ventas totales por género de juego
df_games_g=df_games.query("genre!='N/A'")
box=df_games_g.boxplot(column="total_sales", by="genre", figsize=(12,17))
box.set_ylabel("Ventas")
box.set_xlabel("Género de juego")
box.set_title("Ventas totales de los juegos por género")
```

```
[47]: Text(0.5, 1.0, 'Ventas totales de los juegos por género')
```

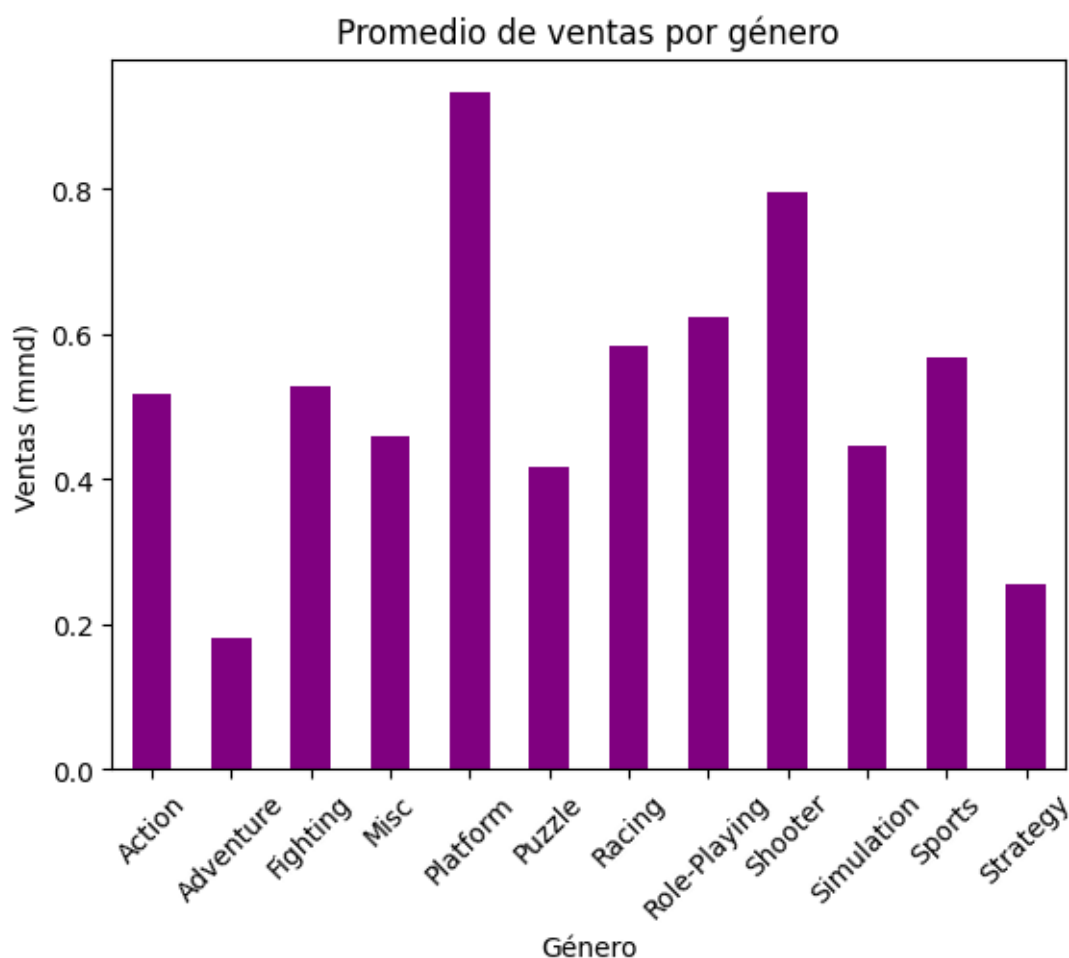
Boxplot grouped by genre



Los diagramas de cajas y bigotes muestran a simple vista medidas de tendencia central relativamente similares en todos los géneros. Lo que hace la diferencia son los juegos más exitosos de cada género. El género 'Sports' muestra por lejos (>80 mmd) el valor atípico más importante en términos de ventas totales. Otros géneros que destacan por productos icónicos son el de 'Platform', 'Racing' y 'Role-Playing'.

```
[48]: #Promedios de ventas por género
df_games_g_grp=df_games_g.groupby('genre')
df_games_g_grp['total_sales'].mean().plot(kind='bar', title="Promedio de ventas_
↳por género",
                                       xlabel='Género', ylabel='Ventas_
↳(mmd)', color='purple', rot=45)
```

```
[48]: <AxesSubplot:title={'center': 'Promedio de ventas por género'}, xlabel='Género',
ylabel='Ventas (mmd)'\>
```

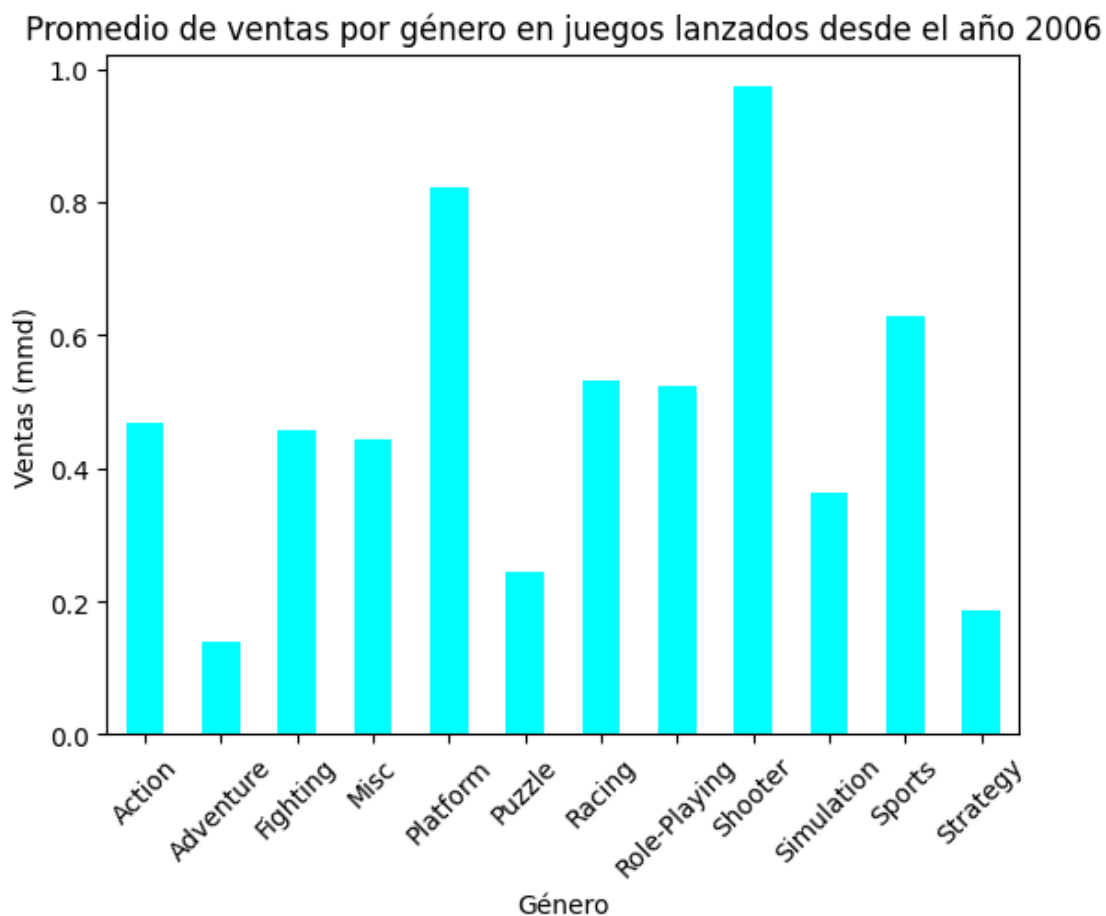


En los promedios de ventas por género de juego destaca el de 'Platform' con un promedio que llegaría casi al millón de dólares, siguiéndole el 'Shooter' (con casi 0.8 mmd) y luego el 'Racing',

‘Role-Playing’ y el de ‘Sports’ con un promedio aproximado de 0.6 mmd. Los géneros menos rentables según este cálculo serían el ‘Adventure’ y el de ‘Strategy’. Sin embargo, para este cálculo se usaron datos de todos los años, por cuanto estos resultados arrojan figuras que podrían resultar contraintuitivas para nuestra época. Para obtener un criterio más acorde con el objeto de este proyecto se podrían descartar los datos de ventas de juegos lanzados hace más de una década.

```
[49]: #Promedios de ventas por género desde los años 2006
df_games_g1=df_games_g.query("year_of_release>=2006")
df_games_g1_grp=df_games_g1.groupby('genre')
df_games_g1_grp['total_sales'].mean().plot(kind='bar', title="Promedio de
    ↪ventas por género en juegos lanzados desde el año 2006",
    xlabel='Género', ylabel='Ventas
    ↪(mmd)', color='cyan', rot=45)
```

```
[49]: <AxesSubplot:title={'center': 'Promedio de ventas por género en juegos lanzados
    desde el año 2006'}, xlabel='Género', ylabel='Ventas (mmd)'\>
```

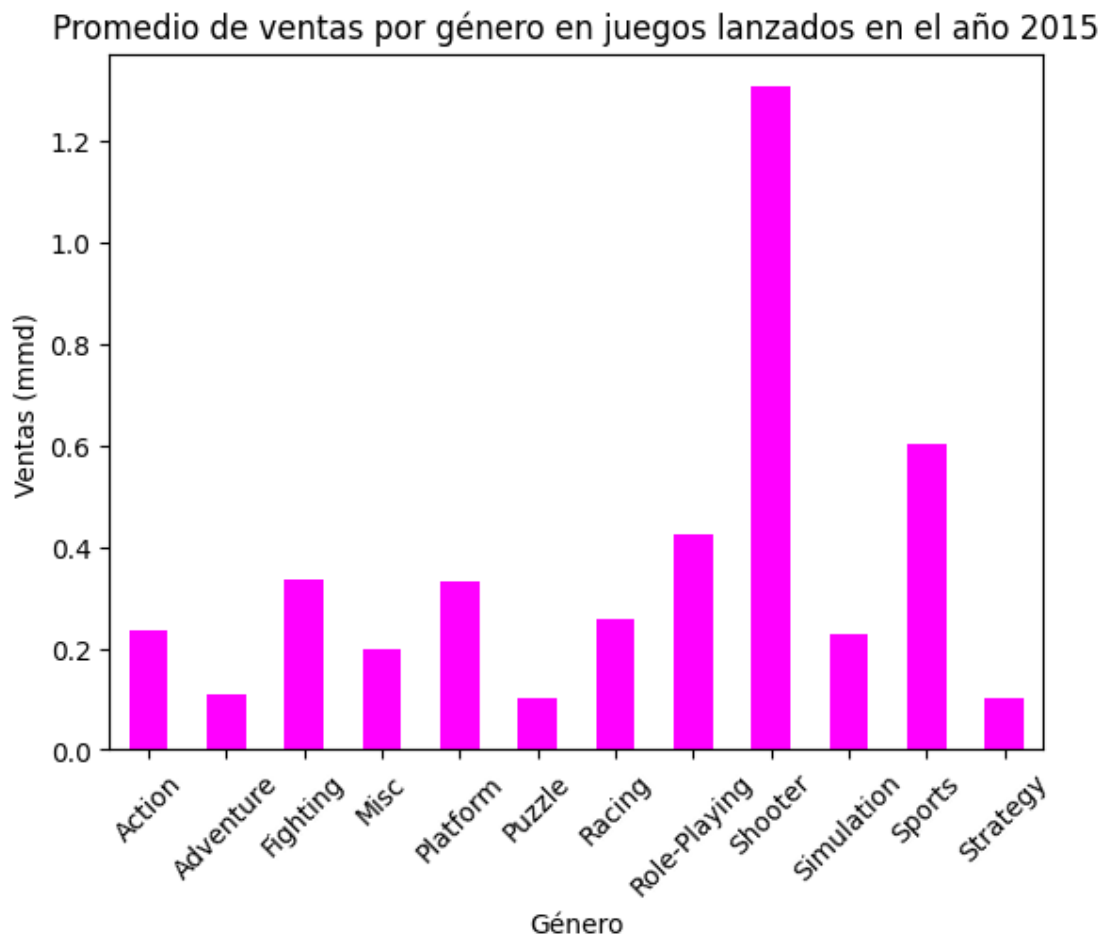


Considerando los juegos lanzados en el presente siglo en cambio vemos que el género con las ventas más altas es el ‘Shooter’ seguido por ‘Platform’ y en tercer lugar ‘Sport’. Las tendencias de los

otros géneros son bastante parecidas al del análisis histórico. Podría deducirse que las preferencias de mercado no han variado demasiado en la última década. Ante este escenario podría reducirse aun más el margen de datos relevantes y considerar las tendencias de ventas de los últimos dos años solamente.

```
[50]: #Promedios de ventas por género desde los años 2015
df_games_g2=df_games_g.query("year_of_release>=2015")
df_games_g2_grp=df_games_g2.groupby('genre')
df_games_g2_grp['total_sales'].mean().plot(kind='bar', title="Promedio de_
↪ventas por género en juegos lanzados en el año 2015",
                                       xlabel='Género', ylabel='Ventas_
↪(mmd)', color='magenta', rot=45)
```

```
[50]: <AxesSubplot:title={'center': 'Promedio de ventas por género en juegos lanzados
en el año 2015'}, xlabel='Género', ylabel='Ventas (mmd)'\>
```



En el último año se observa que el género más importante en promedio de ventas sigue siendo el 'Shooter' (0.8 mmd), pero seguido del género 'Sports', con aproximadamente la mitad del anterior.

Interesantemente, mientras que la preeminencia del ‘Shooter’ se ha mantenido desde la última década, la del género ‘Platform’ en cambio no, pues aquí está desplazado a quinto lugar, a la par con el género ‘Fighting’. No obstante, y siguiendo las tendencias observadas desde hace una década, los géneros con las ventas más bajas siguen siendo el ‘Puzzle’ y el ‘Adventure’.

1.4 Perfil de usuario de cada región

1.4.1 Plataformas principales por región

Para determinar las plataformas con más ventas por región se analizaron los datos desde el año 2006, considerando las plataformas más recientes y un mercado más globalizado. Adicionalmente, para proyectar las ventas para el 2017 se analizaron las plataformas más populares en los últimos dos años, desde el 2015.

```
[51]: #Las cinco plataformas principales de la región NA desde el año 2006
print('Cinco plataformas principales de la región NA desde el 2006\n')
df_games_06=df_games.query("year_of_release>=2006")
na_plat06=df_games_06.groupby('platform')['na_sales'].sum()
print(na_plat06.sort_values(ascending=False).head(5))
#Las cinco plataformas principales de la región NA desde el año 2015
print('\n\nCinco plataformas principales de la región NA desde el 2015')
df_games_15=df_games.query("year_of_release>=2015")
na_plat15=df_games_15.groupby('platform')['na_sales'].sum()
print(na_plat15.sort_values(ascending=False).head(5))
```

Cinco plataformas principales de la región NA desde el 2006

```
platform
X360      588.84
Wii       486.87
PS3       390.13
DS        323.99
PS2       114.89
Name: na_sales, dtype: float64
```

Cinco plataformas principales de la región NA desde el 2015

```
platform
PS4        63.86
XOne       50.30
3DS        10.31
WiiU        8.86
X360        8.05
Name: na_sales, dtype: float64
```

```
[52]: #Las cinco plataformas principales de la región UE desde el año 2006
print('Cinco plataformas principales de la región UE desde el 2006\n')
```

```

eu_plat06=df_games_06.groupby('platform')['eu_sales'].sum()
print(eu_plat06.sort_values(ascending=False).head(5))
#Las cinco plataformas principales de la región UE desde el año 2015
print('\n\nCinco plataformas principales de la región UE desde el 2015')
eu_plat15=df_games_15.groupby('platform')['eu_sales'].sum()
print(eu_plat15.sort_values(ascending=False).head(5))

```

Cinco plataformas principales de la región UE desde el 2006

```

platform
PS3      327.21
X360     267.89
Wii       258.32
DS        142.99
PS4       141.09
Name: eu_sales, dtype: float64

```

Cinco plataformas principales de la región UE desde el 2015

```

platform
PS4       83.54
XOne      28.10
PC         9.10
3DS        7.39
PS3         7.30
Name: eu_sales, dtype: float64

```

```

[53]: #Las cinco plataformas principales de la región JP desde el año 2006
print('Cinco plataformas principales de la región JP desde el 2006\n')
jp_plat06=df_games_06.groupby('platform')['jp_sales'].sum()
print(jp_plat06.sort_values(ascending=False).head(5))
#Las cinco plataformas principales de la región UE desde el año 2015
print('\n\nCinco plataformas principales de la región JP desde el 2015')
jp_plat15=df_games_15.groupby('platform')['jp_sales'].sum()
print(jp_plat15.sort_values(ascending=False).head(5))

```

Cinco plataformas principales de la región JP desde el 2006

```

platform
DS       141.49
3DS      100.62
PS3       79.41
PSP       70.63
Wii       68.28
Name: jp_sales, dtype: float64

```

Cinco plataformas principales de la región JP desde el 2015

platform

3DS 23.55

PS4 12.05

PSV 8.41

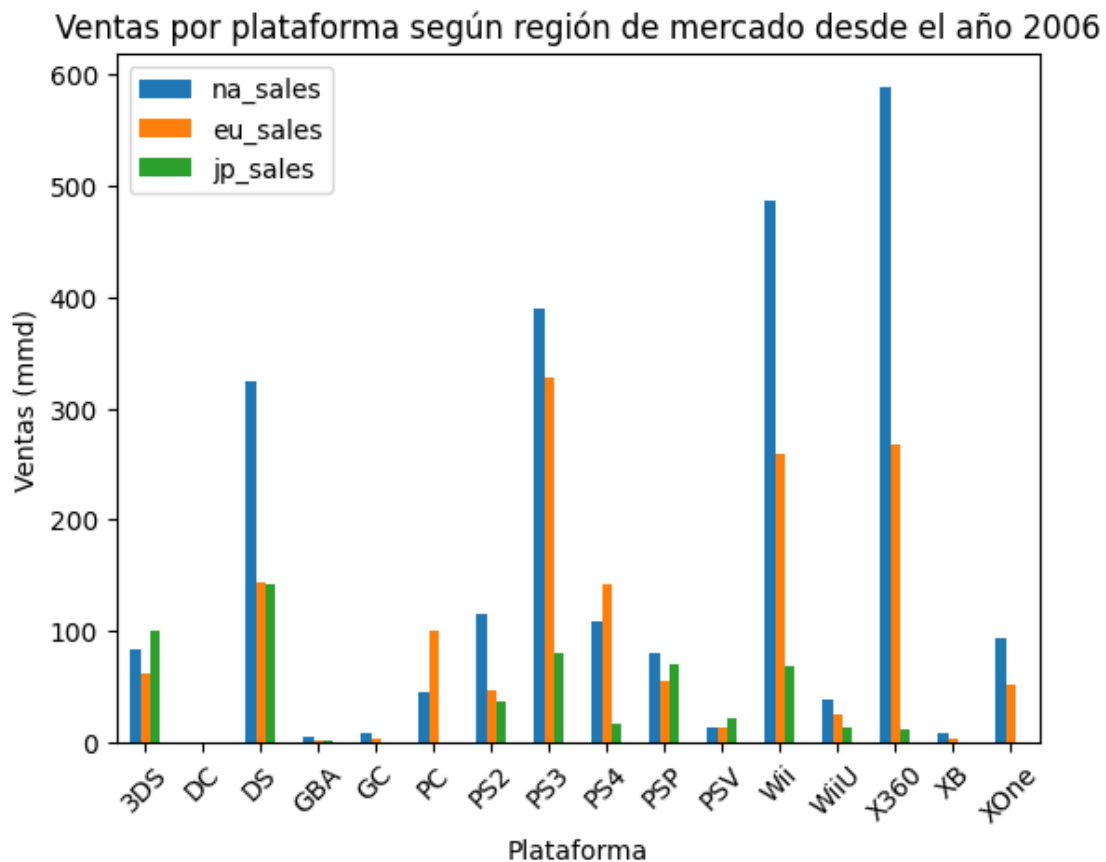
PS3 4.97

WiiU 4.12

Name: jp_sales, dtype: float64

```
[54]: platcon0=pd.concat([na_plat06, eu_plat06], axis='columns')
platcon_t=pd.concat([platcon0, jp_plat06], axis='columns')
platcon_t.plot(kind='bar', title="Ventas por plataforma según región de mercado,
↪desde el año 2006",
xlabel='Plataforma', ylabel='Ventas,
↪(mmd)', rot=45)
```

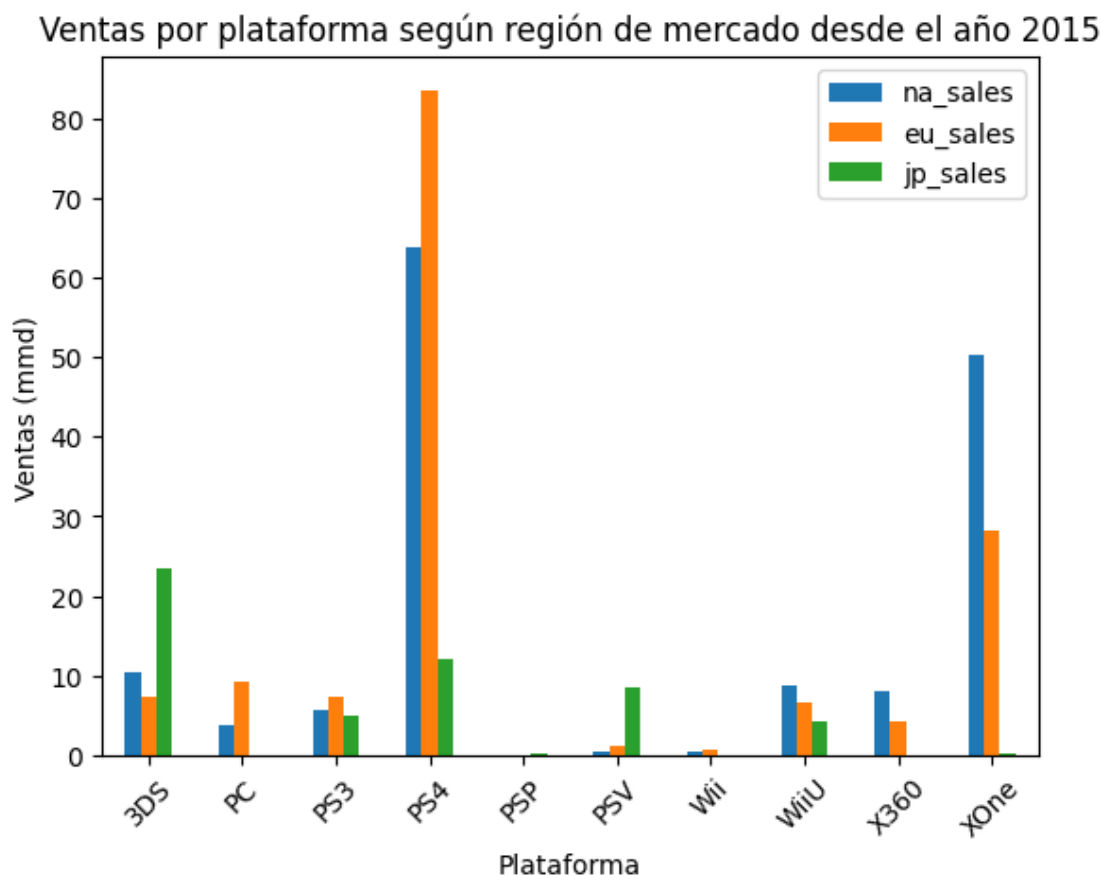
```
[54]: <AxesSubplot:title={'center':'Ventas por plataforma según región de mercado
desde el año 2006'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



Como se aprecia gráficamente, en volumen de ventas el mercado de juegos en NA ha sido el más importante, con la X360 a la cabeza, seguido por la Wii y luego la PS3, DS y PS2. En EU está la PS3 como consola dominante, seguido por la X360, la Wii, DS y PS4. Interesante notar la vigencia en el mercado de los juegos PS2 en NA y la importancia de la PC en EU. La consola portátil preferida en todas las regiones es la Nintendo DS, la que en la región JP ocupa el primer puesto entre todas, seguida de otra portátil, la Nintendo 3DS; asimismo, otra consola portátil relevante en esta región es la PSP, casi a la par con la PS3 en ventas.

```
[55]: platcon=pd.concat([na_plat15, eu_plat15], axis='columns')
platcon1=pd.concat([platcon, jp_plat15], axis='columns')
platcon1.plot(kind='bar', title="Ventas por plataforma según región de mercado,
↳desde el año 2015",
xlabel='Plataforma', ylabel='Ventas_
↳(mmd)', rot=45)
```

```
[55]: <AxesSubplot:title={'center': 'Ventas por plataforma según región de mercado
desde el año 2015'}, xlabel='Plataforma', ylabel='Ventas (mmd)'\>
```



En los últimos dos años se constata la preeminencia de los mercados norteamericano y europeo, además de la importancia de las plataformas más nuevas en estas regiones: PS4, XOne y la

portátil 3DS, que ocupa el primer lugar en el mercado japonés.

1.4.2 Géneros principales por región

```
[56]: #Los cinco géneros principales de la región NA desde el año 2006
print('Cinco géneros principales de la región NA desde el 2006\n')
na_gen06=df_games_g1.groupby('genre')['na_sales'].sum()
print(na_gen06.sort_values(ascending=False).head(5))
#Los cinco géneros principales de la región NA desde el año 2015
print('\n\nCinco géneros principales de la región NA desde el 2015')
na_gen15=df_games_g2.groupby('genre')['na_sales'].sum()
print(na_gen15.sort_values(ascending=False).head(5))
```

Cinco géneros principales de la región NA desde el 2006

```
genre
Action      530.44
Sports      396.64
Shooter     373.13
Misc        285.06
Role-Playing 199.63
Name: na_sales, dtype: float64
```

Cinco géneros principales de la región NA desde el 2015

```
genre
Shooter      48.18
Action       33.68
Sports       26.31
Role-Playing 19.82
Misc         5.58
Name: na_sales, dtype: float64
```

```
[57]: #Los cinco géneros principales de la región EU desde el año 2006
print('Cinco géneros principales de la región EU desde el 2006\n')
eu_gen06=df_games_g1.groupby('genre')['eu_sales'].sum()
print(eu_gen06.sort_values(ascending=False).head(5))
#Los cinco géneros principales de la región EU desde el año 2015
print('\n\nCinco géneros principales de la región EU desde el 2015')
eu_gen15=df_games_g2.groupby('genre')['eu_sales'].sum()
print(eu_gen15.sort_values(ascending=False).head(5))
```

Cinco géneros principales de la región EU desde el 2006

```
genre
Action      350.70
Sports      251.55
```

```

Shooter      238.89
Misc         146.88
Racing       113.39
Name: eu_sales, dtype: float64

```

Cinco géneros principales de la región EU desde el 2015

```

genre
Shooter      40.28
Action       34.82
Sports       27.82
Role-Playing 17.21
Racing        6.52
Name: eu_sales, dtype: float64

```

```

[58]: #Los cinco géneros principales de la región JP desde el año 2006
print('Cinco géneros principales de la región JP desde el 2006\n')
jp_gen06=df_games_g1.groupby('genre')['jp_sales'].sum()
print(jp_gen06.sort_values(ascending=False).head(5))
#Los cinco géneros principales de la región JP desde el año 2015
print('\n\nCinco géneros principales de la región JP desde el 2015')
jp_gen15=df_games_g2.groupby('genre')['jp_sales'].sum()
print(jp_gen15.sort_values(ascending=False).head(5))

```

Cinco géneros principales de la región JP desde el 2006

```

genre
Role-Playing 170.38
Action       102.07
Misc         64.53
Sports       48.97
Platform     35.20
Name: jp_sales, dtype: float64

```

Cinco géneros principales de la región JP desde el 2015

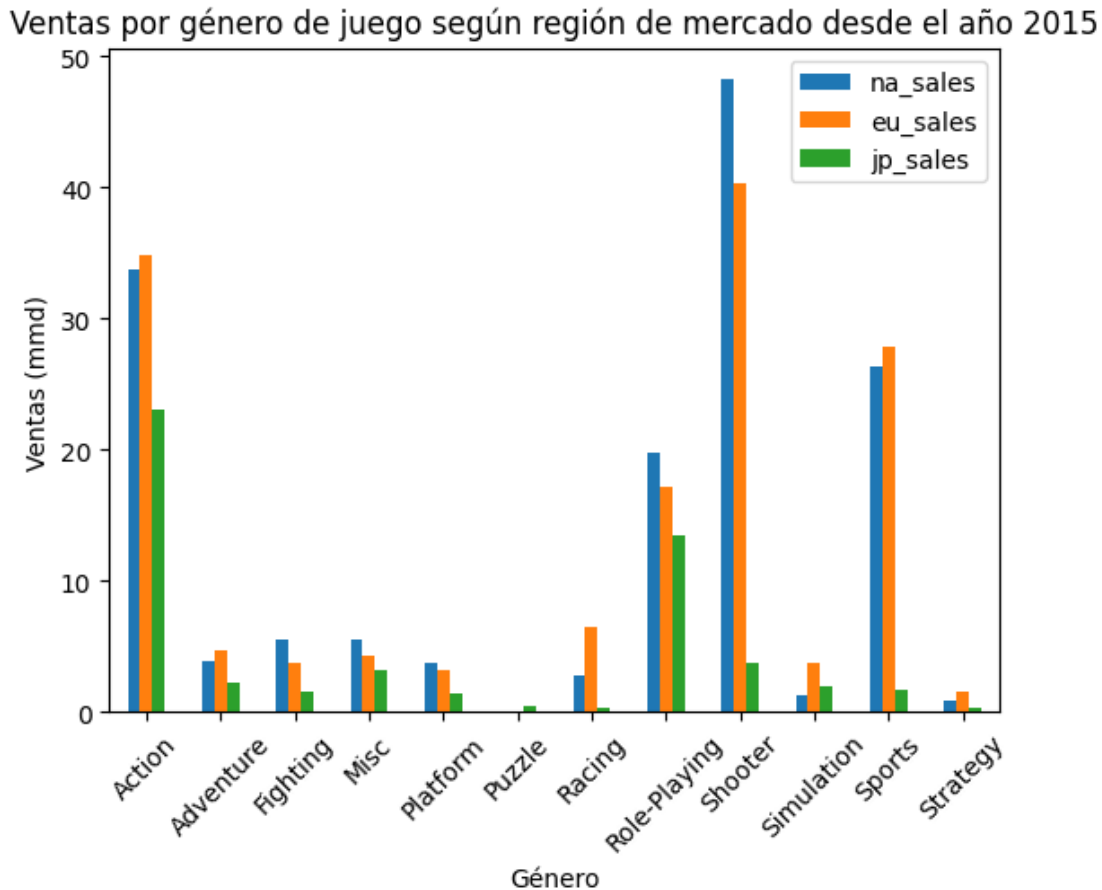
```

genre
Action       23.05
Role-Playing 13.41
Shooter       3.79
Misc          3.19
Adventure     2.22
Name: jp_sales, dtype: float64

```

```
[59]: gencon=pd.concat([na_gen15, eu_gen15], axis='columns')
      gencon1=pd.concat([gencon, jp_gen15], axis='columns')
      gencon1.plot(kind='bar', title="Ventas por género de juego según región de
      ↪mercado desde el año 2015",
                                     xlabel='Género', ylabel='Ventas_
      ↪(mmd)', rot=45)
```

```
[59]: <AxesSubplot:title={'center':'Ventas por género de juego según región de mercado
      desde el año 2015'}, xlabel='Género', ylabel='Ventas (mmd)'\>
```

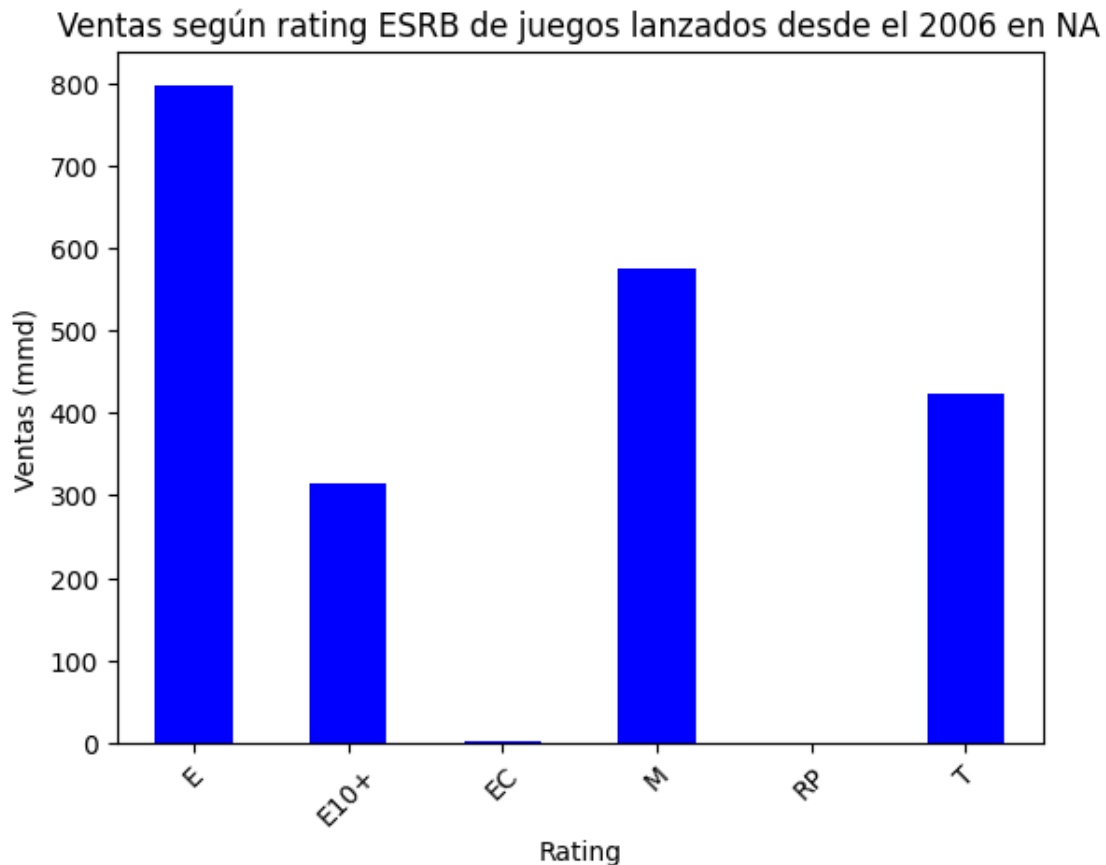


Considerando que la importancia de las plataformas más recientes en el mercado, puse énfasis en el género de juegos de los últimos dos años. En este sentido, los géneros más punteros tanto en NA como en EU son, en este orden, el 'Shooter', 'Action', 'Sports' y 'Role-Playing'. Los géneros más importantes en JP difieren en mucho: allá los géneros con más ventas son el 'Action' y el 'Role-Playing'; el 'Shooter' les sigue, pero muy poder debajo, representando apenas un tercio del segundo en ventas. Las ventas en JP decantan por juegos de un género más cerebral.

1.4.3 Influencia de las clasificaciones ESRB en las ventas

```
[60]: df_games_esrb06=df_games.query("year_of_release>=2006 and rating!='N/A'")
df_games_esrb_grp06=df_games_esrb06.groupby('rating')
df_games_esrb_grp06['na_sales'].sum().plot(kind='bar', title="Ventas según
↳rating ESRB de juegos lanzados desde el 2006 en NA",
                                             xlabel='Rating', ylabel='Ventas
↳(mmd)', color='blue', rot=45)
```

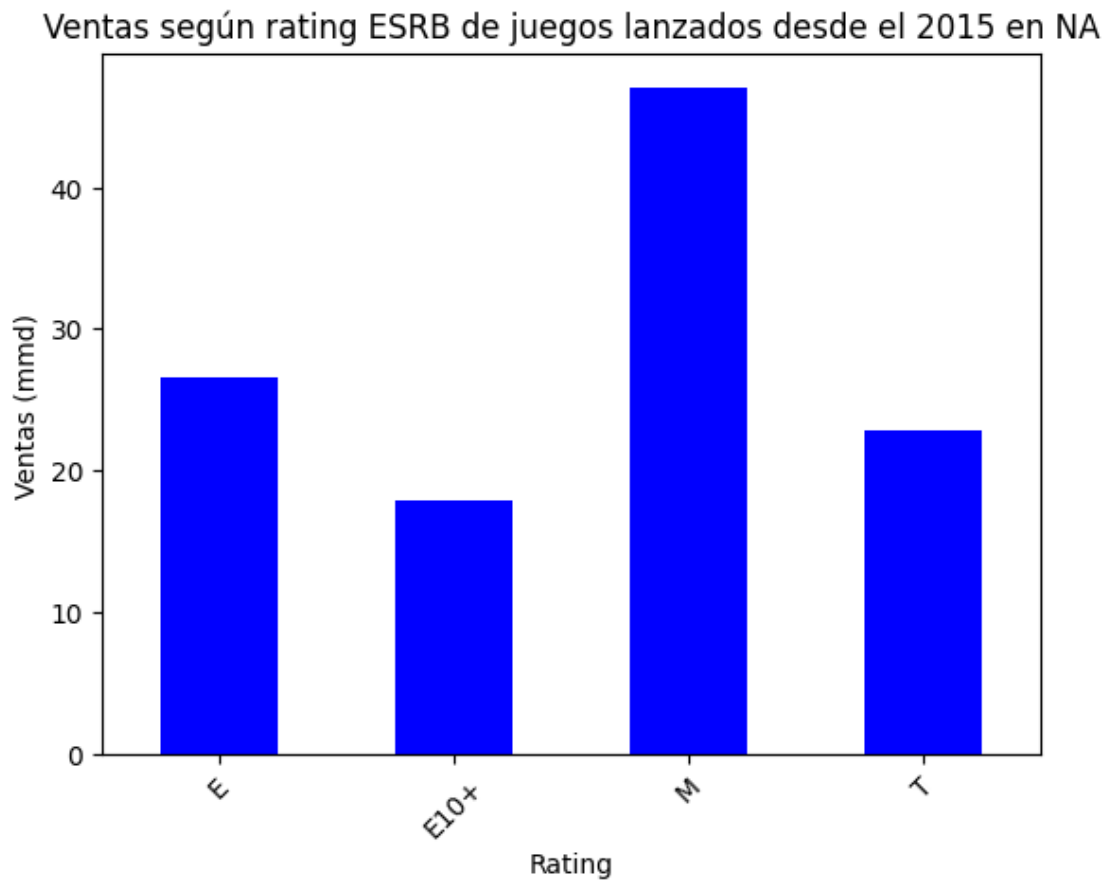
```
[60]: <AxesSubplot:title={'center':'Ventas según rating ESRB de juegos lanzados desde
el 2006 en NA'}, xlabel='Rating', ylabel='Ventas (mmd)'\>
```



```
[61]: df_games_esrb=df_games.query("year_of_release>=2015 and rating!='N/A'")
df_games_esrb_grp=df_games_esrb.groupby('rating')
df_games_esrb_grp['na_sales'].sum().plot(kind='bar', title="Ventas según rating
↳ESRB de juegos lanzados desde el 2015 en NA",
                                             xlabel='Rating', ylabel='Ventas
↳(mmd)', color='blue', rot=45)
```

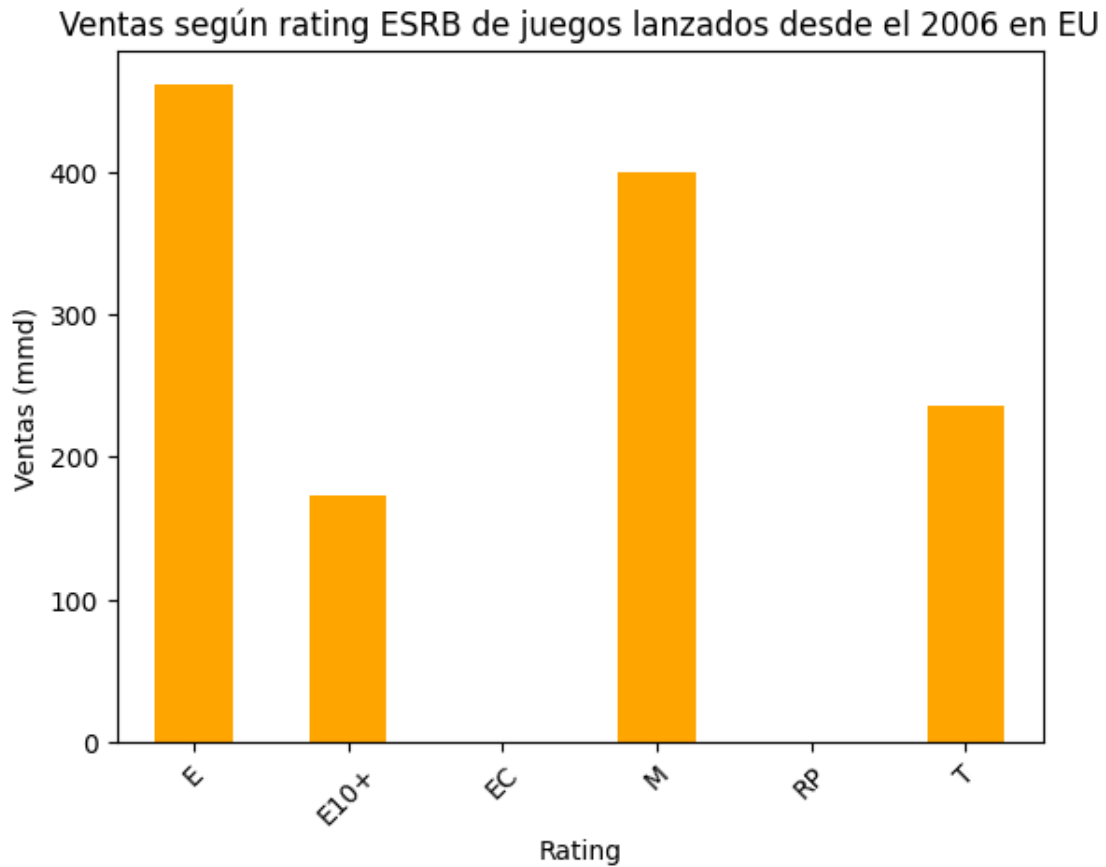


```
[61]: <AxesSubplot:title={'center':'Ventas según rating ESRB de juegos lanzados desde el 2015 en NA'}, xlabel='Rating', ylabel='Ventas (mmd)'\>
```



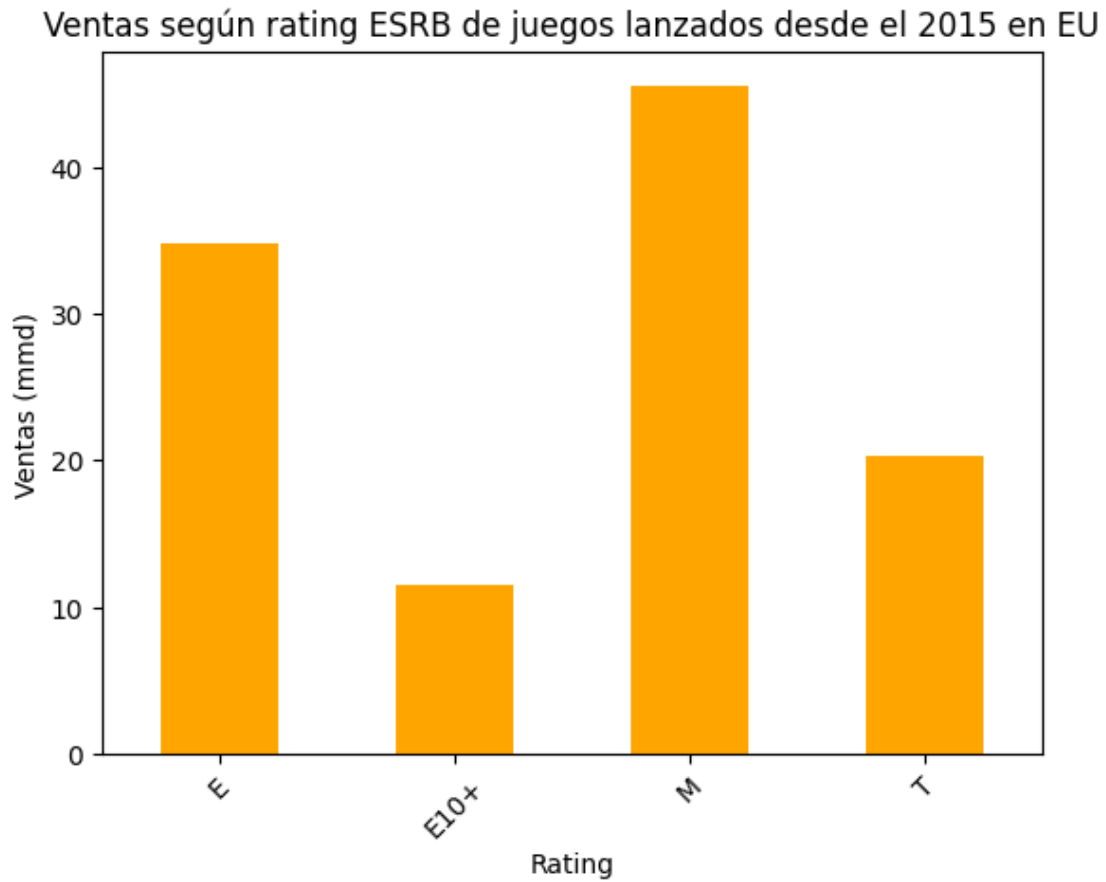
```
[62]: df_games_esrb_grp06['eu_sales'].sum().plot(kind='bar', title="Ventas según rating ESRB de juegos lanzados desde el 2006 en EU",  
        xlabel='Rating', ylabel='Ventas (mmd)', color='orange', rot=45)
```

```
[62]: <AxesSubplot:title={'center':'Ventas según rating ESRB de juegos lanzados desde el 2006 en EU'}, xlabel='Rating', ylabel='Ventas (mmd)'\>
```



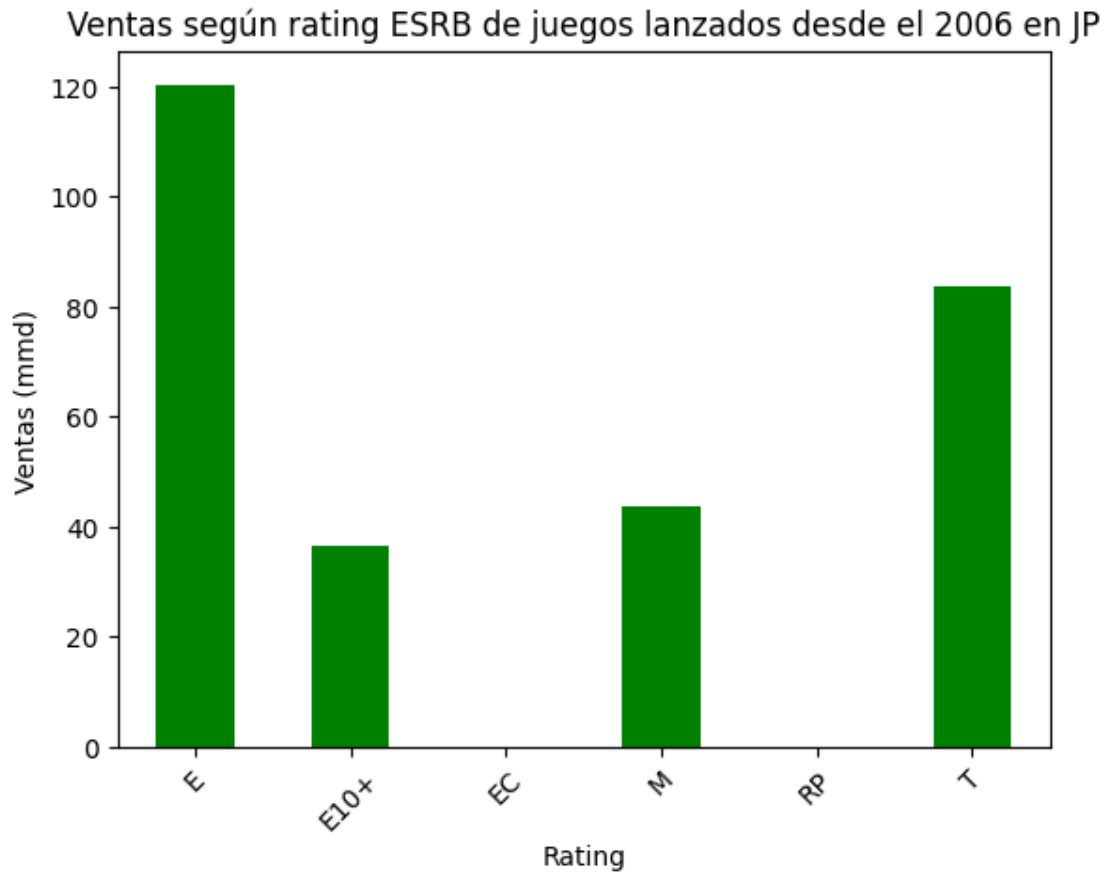
```
[63]: df_games_esrb_grp['eu_sales'].sum().plot(kind='bar', title="Ventas según rating_
↳ ESRB de juegos lanzados desde el 2015 en EU",
                                             xlabel='Rating', ylabel='Ventas_
↳ (mmd)', color='orange', rot=45)
```

```
[63]: <AxesSubplot:title={'center': 'Ventas según rating ESRB de juegos lanzados desde
el 2015 en EU'}, xlabel='Rating', ylabel='Ventas (mmd)'\>
```



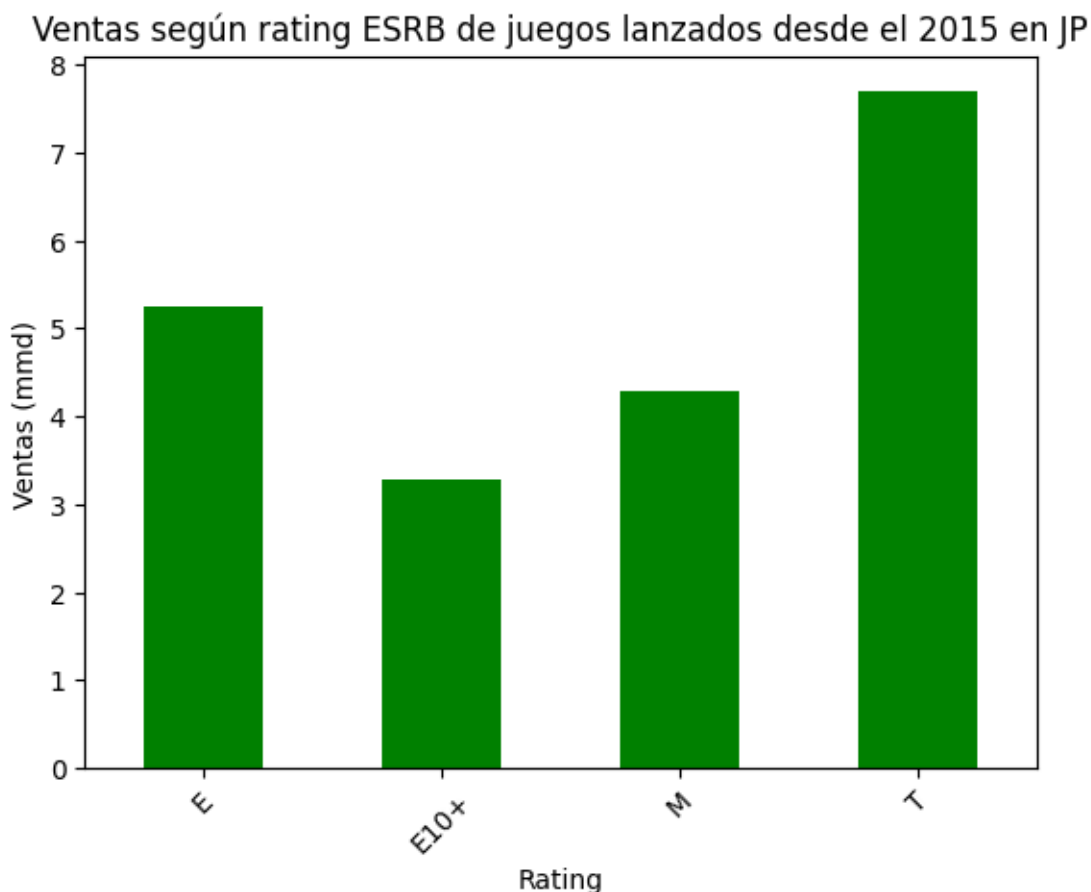
```
[64]: df_games_esrb_grp06['jp_sales'].sum().plot(kind='bar', title="Ventas según
      ↳ rating ESRB de juegos lanzados desde el 2006 en JP",
      xlabel='Rating', ylabel='Ventas
      ↳ (mmd)', color='green', rot=45)
```

```
[64]: <AxesSubplot:title={'center': 'Ventas según rating ESRB de juegos lanzados desde
      el 2006 en JP'}, xlabel='Rating', ylabel='Ventas (mmd)'\>
```



```
[65]: df_games_esrb_grp['jp_sales'].sum().plot(kind='bar', title="Ventas según rating_
↳ ESRB de juegos lanzados desde el 2015 en JP",
                                             xlabel='Rating', ylabel='Ventas_
↳ (mmd)', color='green', rot=45)
```

```
[65]: <AxesSubplot:title={'center': 'Ventas según rating ESRB de juegos lanzados desde
el 2015 en JP'}, xlabel='Rating', ylabel='Ventas (mmd)'\>
```



Las variaciones de mercado en las categorías ESRB en las regiones NA y EU son bastante similares en los últimos años. Pero si se observa las cuotas de mercado desde hace una década se puede notar cierto cambio demográfico. Los juegos para todas las edades (E) dejaron de ser el segmento con más ventas, desplazados por aquellos para audiencias más maduras (M) a lo largo de los últimos años (comprobandose la variación al tomar los datos desde el 2010, 2014 y demás). Este cambio en la audiencia también es patente en la región JP; sin embargo, aquí se observa que la cuota de juegos para adolescentes (T) ya era más relevante que aquellos calificados M. Así pues, si bien estas variaciones pueden definitivamente explicarse por el cambio demográfico, es decir, un público cada vez más adulto, también hay factores culturales a tener en cuenta; en el caso de la región JP, la popularidad del anime, por ejemplo.

Comentario del revisor:

Excelente! Con este análisis por región ayuda a complementar el análisis general anterior y a hacer zoom a los resultados por cada una de las regiones.

1.5 Prueba de hipótesis

1.5.1 Hipótesis 1

La hipótesis de prueba:

Hipótesis nula(H_0): Las calificaciones promedio de los usuarios para las plataformas XOne y PC son las mismas.

Hipótesis alternativa(H_1): Las calificaciones promedio de los usuarios para las plataformas XOne y PC son diferentes estadísticamente.

```
[71]: #Cálculo de varianzas

df_xone=df_games.query("platform == 'XOne')['user_score'] #Filtro la tabla y
↳descarto valores ausentes
print("Varianza de las calificaciones de usuarios para XOne:\n",np.
↳var(df_xone), "\n\n") #Varianza de calificaciones de usuarios de XOne

df_pc=df_games.query("platform == 'PC')['user_score'] #Filtro la tabla y
↳descarto valores ausentes
print("Varianza de las calificaciones de usuarios para PC:\n",np.var(df_pc))
↳#Varianza de calificaciones de usuarios de PC
```

Varianza de las calificaciones de usuarios para XOne:
1.5831195397400384

Varianza de las calificaciones de usuarios para PC:
1.879813729871948

La hipótesis alternativa siempre indica que un parámetro de población es más bajo, más alto o diferente del valor hipotético de la hipótesis nula.

Para probar la hipótesis uso la función `ttest_ind` de la librería `scipy`. En base a lo analizado previamente escojo un valor alfa de 0.05 para asegurar que no haya más de un 5% de probabilidades que las diferencias entre ambas poblaciones se puedan deber al azar. Asimismo se ha constatado en los análisis de datos que las varianzas entre las poblaciones son más bien similares, por tanto el parámetro `equal_var` se expresa como `True`.

```
[72]: # Prueba de las hipótesis
alpha=0.05
results=st.ttest_ind(df_xone, df_pc, equal_var=True)
print('Valor p:', results.pvalue)

if(results.pvalue<alpha):
    print('Rechazamos la H0')
else:
    print('No podemos rechazar la H0')
```

Valor p: 0.00010123955107193582
Rechazamos la H_0

Rechazamos la hipótesis nula y concluimos que las calificaciones promedio de los usuarios para las plataformas XOne y PC son estadísticamente diferentes.

1.5.2 Hipótesis 2

La hipótesis de prueba:

Hipótesis nula(H0): Las calificaciones promedio de los usuarios para los géneros de 'Action' y 'Sports' son las mismas.

Hipótesis alternativa(H1): Las calificaciones promedio de los usuarios para los géneros de 'Actions' y 'Sport' son diferentes estadísticamente.

```
[73]: #Cálculo de varianzas

df_action=df_games.query("genre == 'Action'")['user_score'] #Filtro la tabla y
↳descarto valores ausentes
print("Varianza de las calificaciones de usuarios para el género 'Action':
↳\n",np.var(df_action), "\n\n") #Varianza de calificaciones de usuarios para
↳el género 'Action'
df_sport=df_games.query("genre == 'Sports'")['user_score'] #Filtro la tabla y
↳descarto valores ausentes
print("Varianza de las calificaciones para el género 'Sport':\n",np.
↳var(df_sport)) #Varianza de calificaciones de usuarios para el género 'Sport'
```

```
Varianza de las calificaciones de usuarios para el género 'Action':
1.152365546180358
```

```
Varianza de las calificaciones para el género 'Sport':
1.290948596289858
```

La hipótesis alternativa siempre indica que un parámetro de población es más bajo, más alto o diferente del valor hipotético de la hipótesis nula.

Para probar la hipótesis uso la función `ttest_ind` de la librería `scipy`. En base a lo analizado previamente escojo un valor alfa de 0.05 para asegurar que no haya más de un 5% de probabilidades que las diferencias entre ambas poblaciones se puedan deber al azar. Asimismo se ha constatado en los análisis de datos que las varianzas entre las poblaciones son más bien similares, por tanto el parámetro `equal_var` se expresa como `True`.

```
[74]: # Prueba de las hipótesis
alpha=0.05
results=st.ttest_ind(df_action, df_sport, equal_var=True)
print('Valor p:', results.pvalue)

if(results.pvalue<alpha):
    print('Rechazamos la H0')
else:
    print('No podemos rechazar la H0')
```

```
Valor p: 0.7131417795320426
No podemos rechazar la H0
```

```
[75]: print("Promedio de las calificaciones de usuarios para el género 'Action':  
      ↪\n",df_action.mean(), "\n\n") #Varianza de calificaciones de usuarios para  
      ↪el género 'Action' #Filtro la tabla y descarto valores ausentes  
print("Promedio de las calificaciones para el género 'Sport':\n",df_sport.  
      ↪mean()) #Varianza de calificaciones de usuarios para el género 'Sport'
```

Promedio de las calificaciones de usuarios para el género 'Action':
7.257761947165331

Promedio de las calificaciones para el género 'Sport':
7.246890971039183

No podemos rechazar la hipótesis nula con estos datos, pues las calificaciones promedio entre ambos géneros no es significativa estadísticamente.

Comentario revisor

Para estas pruebas te recomiendo hacer una prueba de Levene para mostrar si las varianzas son iguales y agregarlo dentro de la función. Actualmente lo colocar como `equal_var=True`. Para esto, primero debes calcular las varianzas para cada uno de las plataformas y en un segundo tiempo debes de hacer uso de la siguiente función:

```
levene(xbox_one_data['user_score'], pc_data['user_score'])
```

Adoptaremos un nivel de significancia de 0.05; si el valor p resultante es mayor a 0.05, no podemos rechazar la hipótesis nula, y si es menor a 0.05, rechazamos la hipótesis nula, indicando que las varianzas no son iguales.

Solamente recuerda que la prueba de Levene no es sustituto a la prueba de `st.ttest_ind`, más bien es complemento para saber que colocar dentro del elemento “`equal_var`”. En este caso como rechazamos la hipótesis de varianzas iguales debemos de colocar `False`. Es por eso que para terminar la prueba debes de realizar la prueba de `st.ttest_ind` considerando el resultado de la prueba realizada de Levene.

Comentario del estudiante:

Muchas gracias, lo tendré en cuenta.

1.6 Conclusión general

Los datos de la tabla denotan cambios significativos en la última mitad de la década en el mercado de videojuegos. La disminución en la cantidad de títulos lanzados, los ciclos aparentemente más cortos de las nuevas plataformas y el cambio en las cuotas de mercados según las calificaciones ESRB sugieren tanto cambios demográficos y como probablemente de preferencias en plataformas de entretenimiento digital que no cubre esta tienda (por ejemplo, juegos en tabletas y teléfonos móviles).

No obstante, el mercado seguirá siendo rentable y un particular énfasis en campañas publicitarias para el siguiente año debería ponerse en las plataformas emergentes como el PS4, XOne y la Nintendo 3DS, especialmente para regiones norteamericana y europea, que componen los segmentos más importantes. También debería considerarse la cuota cada vez más alta en el mercado de

audiencias más adultas. Adicionalmente debería tenerse miramiento a los géneros más rentables en los últimos años, como los shooters y los de deportes y su asociación con las plataformas más nuevas. En la región asiática hay que considerar la preferencia por juegos de acción y otros más bien cerebrales. Otro aspecto a considerar podría ser la capitalización de las reseñas profesionales para impulsar los títulos más prometedores.

Comentario revisor

En general creo que hiciste un muy buen trabajo con el proyecto, pudiste limpiar y trabajar las bases de datos de buena manera. Además, el análisis exploratorio de datos fue completo al mostrar resultados relevantes que pueden ser de mucha utilidad para la toma de decisiones y desarrollaste las pruebas de hipótesis de una buena manera. No obstante, recuerda que siempre podemos mejorar y te menciono algunos puntos que debes considerar:

- Verificar que cuando llenamos variables con valores nulos los estamos completando con valores que no sesgan nuestros resultados
- Considerar eliminar registros atípicos que puedan sesgar nuestros resultados.
- Considerar desarrollar un análisis para comprobar los supuestos de la prueba de hipótesis (varianzas iguales)