**PROYECTO SPRINT 6**

**Descripción del proyecto**

Trabajas para la tienda online Ice que vende videojuegos por todo el mundo. Las reseñas de usuarios y expertos, los géneros, las plataformas (por ejemplo, Xbox o PlayStation) y los datos históricos sobre las ventas de juegos están disponibles en fuentes abiertas. Tienes que identificar patrones que determinen si un juego tiene éxito o no. Esto te permitirá detectar proyectos prometedores y planificar campañas publicitarias.

Delante de ti hay datos que se remontan a 2016. Imaginemos que es diciembre de 2016 y estás planeando una campaña para 2017.

Lo importante es adquirir experiencia de trabajo con datos. Realmente no importa si estás pronosticando las ventas de 2017 en función de los datos de 2016 o las ventas de 2027 en función de los datos de 2026.

El dataset contiene una columna "rating" que almacena la clasificación ESRB de cada juego. El Entertainment Software Rating Board (la Junta de clasificación de software de entretenimiento) evalúa el contenido de un juego y asigna una clasificación de edad como Adolescente o Adulto.

Instrucciones para completar el proyecto

Paso 1.Abre el archivo de datos y estudia la información general

Ruta de archivo:

/datasets/games.csv . Descarga el dataset

Paso 2. Prepara los datos

Reemplaza los nombres de las columnas (ponlos en minúsculas).

Convierte los datos en los tipos necesarios.

Describe las columnas en las que los tipos de datos han sido cambiados y explica por qué.

Si es necesario, elige la manera de tratar los valores ausentes:

Explica por qué rellenaste los valores ausentes como lo hiciste o por qué decidiste dejarlos en blanco.

¿Por qué crees que los valores están ausentes? Brinda explicaciones posibles.

Presta atención a la abreviatura TBD: significa "to be determined" (a determinar). Especifica cómo piensas manejar estos casos.

Calcula las ventas totales (la suma de las ventas en todas las regiones) para cada juego y coloca estos valores en una columna separada.

Paso 3. Analiza los datos

Mira cuántos juegos fueron lanzados en diferentes años. ¿Son significativos los datos de cada período?

Observa cómo varían las ventas de una plataforma a otra. Elige las plataformas con las mayores ventas totales y construye una distribución basada en los datos de cada año. Busca las plataformas que solían ser populares pero que ahora no tienen ventas. ¿Cuánto tardan generalmente las nuevas plataformas en aparecer y las antiguas en desaparecer?

Determina para qué período debes tomar datos. Para hacerlo mira tus respuestas a las preguntas anteriores. Los datos deberían permitirte construir un modelo para 2017.

Trabaja solo con los datos que consideras relevantes. Ignora los datos de años anteriores.

¿Qué plataformas son líderes en ventas? ¿Cuáles crecen y cuáles se reducen? Elige varias plataformas potencialmente rentables.

Crea un diagrama de caja para las ventas globales de todos los juegos, desglosados por plataforma. ¿Son significativas las diferencias en las ventas? ¿Qué sucede con las ventas promedio en varias plataformas? Describe tus hallazgos.

Mira cómo las reseñas de usuarios y profesionales afectan las ventas de una plataforma popular (tu elección). Crea un gráfico de dispersión y calcula la correlación entre las reseñas y las ventas. Saca conclusiones.

Teniendo en cuenta tus conclusiones compara las ventas de los mismos juegos en otras plataformas.

Echa un vistazo a la distribución general de los juegos por género. ¿Qué se puede decir de los géneros más rentables? ¿Puedes generalizar acerca de los géneros con ventas altas y bajas?

Paso 4. Crea un perfil de usuario para cada región

Para cada región (NA, UE, JP) determina:

Las cinco plataformas principales. Describe las variaciones en sus cuotas de mercado de una región a otra.

Los cinco géneros principales. Explica la diferencia.

Si las clasificaciones de ESRB afectan a las ventas en regiones individuales.

Paso 5. Prueba las siguientes hipótesis:

— Las calificaciones promedio de los usuarios para las plataformas Xbox One y PC son las mismas.

— Las calificaciones promedio de los usuarios para los géneros de Acción y Deportes son diferentes.

Establece tu mismo el valor de umbral alfa.

Explica:

— Cómo formulaste las hipótesis nula y alternativa.

— Qué criterio utilizaste para probar las hipótesis y por qué.

Paso 6. Escribe una conclusión general

Formato: Completa la tarea en Jupyter Notebook. Inserta el código de programación en las celdas code y las explicaciones de texto en las celdas markdown. Aplica formato y agrega encabezados.

Descripción de datos

— Name (Nombre)

— Platform (Plataforma)

— Year\_of\_Release (Año de lanzamiento)

— Genre (Género)

— NA\_sales (ventas en Norteamérica en millones de dólares estadounidenses)

— EU\_sales (ventas en Europa en millones de dólares estadounidenses)

— JP\_sales (ventas en Japón en millones de dólares estadounidenses)

— Other\_sales (ventas en otros países en millones de dólares estadounidenses)

— Critic\_Score (máximo de 100)

— User\_Score (máximo de 10)

— Rating (ESRB)

Es posible que los datos de 2016 estén incompletos.

¿Cómo será evaluado mi proyecto?

Lee atentamente estos criterios de evaluación de proyectos antes de empezar a trabajar.

Esto es lo que buscan los revisores de proyecto cuando evalúan tu proyecto:

¿Cómo describirías los problemas identificados en los datos?

¿Cómo se prepara un dataset para el análisis?

¿Cómo creas gráficos de distribución y cómo los explicas?

¿Cómo calculas la desviación estándar y varianza?

¿Formulas las hipótesis alternativas y nulas?

¿Qué métodos aplicas a la hora de probarlos?

¿Explicas los resultados de tus pruebas de hipótesis?

¿Sigues la estructura del proyecto y mantienes tu código ordenado y comprensible?

¿A qué conclusiones llegas?

¿Has dejado comentarios claros y relevantes en cada paso?

Todo lo que necesitas para completar este proyecto se encuentra en las hojas informativas y los resúmenes de los capítulos anteriores.

//////////////////////////////////////////////////////////////////////////////////////////////////

**Desarrollo**

# importar librerías

import pandas as pd

from matplotlib import pyplot as plt

from scipy import stats as st

import numpy as np

import math

import seaborn as sns

from scipy.stats import levene

# Carga los archivos de datos en diferentes DataFrames

df=pd.read\_csv('/datasets/games.csv')

print(df.head()) # Vemos una muestra del df

print('///////////////////')

df.info() # Imprimimos la información general del df

#Reemplaza los nombres de las columnas, ponlos en minúsculas

newcol=[] #creo lista de nuevos nombres

for i in df.columns: #itero en cada una de las columnas

minu=i.lower() #convierto a minusculas

noes=minu.strip() #elimino espacios por si acaso

newcol.append(noes) #agrego a la lista creada

df.columns=newcol #reemplazo nombres con los de la lista

print(df.columns)

#Convierte los datos en los tipos necesarios.

#debido al error could not convert string to float: 'tbd' reemplazamos por Nan

# Reemplazamos 'tbd' con NaN en la columna user score

df['user\_score'] = df['user\_score'].replace('tbd', np.nan)

# Conviertimos 'user\_score' a tipo flotante

df['user\_score'] = df['user\_score'].astype('float')

# Conviertimos el año de lanzamiento a tipo entero

#debido al error cannot convert to 'int64'-dtype NumPy array with missing values. convertimos a Int64

df['year\_of\_release'] = df['year\_of\_release'].astype('Int64')

# Conviertimos la puntuación de la crítica a tipo flotante

#(no se cómo califique la crítica, si debía ser entero solo es cambiar a Int64)

df['critic\_score'] = df['critic\_score'].astype('float')

#comprobamos si realizó los cambios correctamente

df.info()

print(df.head())

Sobre los valores ausentes

Con el metodo info podemos notar que las columnas que cuentan con valores ausentes son las siguientes:

name y genre- en estas dos columnas solo hay dos valores ausentes en cada una, probablemente se trató de algún error de informarción que faltó colocar nada más, ya que no es posible que se sepa información de ventas y ratings sin antes saber de qué juego se trata.

year\_of\_release.- quizás solo son fechas de lanzamiento que no se registron.

critic\_score y user\_score.- probablemente son juegos que no fueron reseñados debido a la poca popularidad o chance se tratan de juegos antiguos que llegan a carecer de estas puntuaciones. Además de que no todos los juegos van a ser de interés de la mayoría de usuarios o de la crítica en general.

rating.- puede ser la misma situación que critic score y user score solo que aunado a eso la calificación ESRB parece ser muy especial como para que la tengan todos los juegos.

A medida que avance en el proyecto, iré lidiando con los valores ausentes conforme estos me vayan presentando conflictos, mientras tanto, los voy a mantener en el df, al final del proyecto en la conclusión desgloso más sobre este asunto.

#Calculo de ventas totales

df['total\_sales']= df['eu\_sales']+df['jp\_sales']+df['other\_sales']+df['na\_sales']

print(df.head())

#Muchos juegos fueron lanzados en diferentes años y para comprobar si son significativos los

#datos de cada período realizaremos un análisis estadístico

#veremos el valor máximo

maxy=df['year\_of\_release'].max()

print(f'valor máximo {maxy} ')

#veremos el valor máximo

miny=df['year\_of\_release'].min()

print(f'valor mínimo {miny} ')

#veremos el valor de la media

meany=df['year\_of\_release'].mean()

print(f'valor media {meany} ')

#veremos el valor de la mediana

mediany=df['year\_of\_release'].median()

print(f'valor mediana {mediany} ')

Como podemos observar los datos sobre las fechas de lanzamiento indican que a pesar de contar con un rango de los datos algo amplio, considerando una diferencia de más de 30 años, realmento los valores no pueden considerarse atípicos ya que la media y la mediana tienen solo una diferencia de un año, dejando en claro que tienen una distribución simétrica por lo que se puede decir que los datos de cada periodo si son significativos

Comentario del revisor (1ra Iteración)

Puedes complementar estos valores con una figura mostrando las ventas año a año, de esta forma podremos tener claridad sobre cómo ha cambiado esta variable en el tiempo

#Observación, cómo varían las ventas de una plataforma a otra.

#agrupamos los datos por plataforma

plat= df.groupby('platform')

#realizamos la suma de las ventas totales y organizamos de mayor a menor

platform\_sales=plat['total\_sales'].sum().sort\_values(ascending=False)

#tomamos las 5 primeras plataformas de la lista

top\_plat = platform\_sales.head()

print("Top de plataformas más exitosas:", top\_plat)

#filtramos el df para incluir solo las plataformas principales

top\_df = df[df['platform'].isin(top\_plat.index.tolist())]

#agrupamos por plataforma y año, y calcula las ventas totales

platform\_yearly = top\_df.groupby(['platform', 'year\_of\_release'])

#calculamos las ventas totales

platform\_yearly\_sales=platform\_yearly['total\_sales'].sum().reset\_index()

#debido a que las plataformas más populares son de los años 2000 para adelante, ajustaremos

platform\_yearly\_sales = platform\_yearly\_sales[platform\_yearly\_sales['year\_of\_release'] > 2000]

#Para visualizar la ascendencia y decadencia realizaremos un grafico para cada plataforma

#nos apoyaremos de seaborn lineplot ya que con el por medio de hue podemos realizar una linea

#diferente para cada plataforma

sns.lineplot(data=platform\_yearly\_sales, x='year\_of\_release', y='total\_sales', hue='platform', marker='o')

plt.title('Ventas anuales de las plataformas más exitosas')

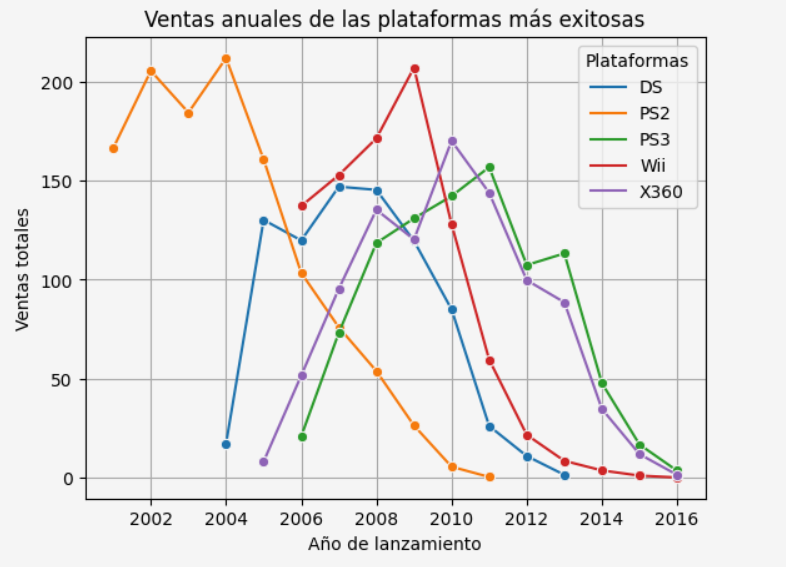
plt.xlabel('Año de lanzamiento')

plt.ylabel('Ventas totales')

plt.legend(title='Plataformas')

plt.grid(True)

plt.show()



El cálculo de la suma de las ventas en los datos agrupados por plataformas nos indica que las consolas con más ventas en el df son en orden descendente son la ps2, el xbox 360, el ps3, la wii y la nintendo DS, en general todas tuvieron un ciclo de vida bastante similar. Como podemos visualizar en la gráfica las plataformas a partir de su año de lanzamiento crecen sus ventas hasta llegar a un pico (que aparece entre su cuarto y sexto año) y a partir de ese máximo valor empiezan a decrecer sus ventas con el tiempo, claramente esto ocurre porque a medida que pasa el tiempo, las plataformas de van volviendo obsoletas, aparecen nuevas consolas, dejan de desarrollarle juegos y dejan de producir más hasta ya dejar de tener ventas, en general parece que hay un período de tempo en el que la aparición de nuevas consolas se da cada año y de ahí se detiene hasta llegar al siguiente periodo, mientras tanto las plataformas están disponibles en el mercado por lo menos unos diez años hasta su desaparición.

#Creación de un diagrama de caja para las ventas globales de todos los juegos desglosados por plataforma.

#nuevamente nos apoyamos de sns para graficar las plataformas contra las ventas de nuestro df

sns.boxplot(x='platform', y='total\_sales', data=df)

plt.title('Distribución de ventas por juego por plataforma')

plt.xlabel('Plataforma')

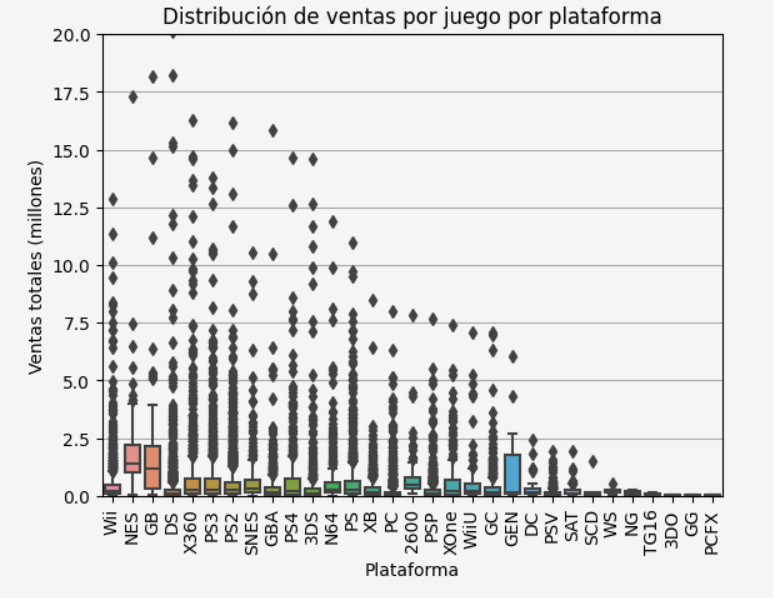
plt.ylabel('Ventas totales (millones)')

plt.grid(True, axis='y')

plt.xticks(rotation=90)

plt.ylim(0,20) # Establece el límite a 20 millones en las ventas

plt.show()



Como podemos observar realmente la información de las ventas por plataforma nos proporciona datos con diferencias que no son tan significativas en las ventas, o al menos en la mayoría ya que hay consolas que presentan valores atípicos muy notorios como por ejemplo la Wii que llegó a tener un juego que generó más de 80 millones o la NES que tuvo un juego que generó más de 40 millones (razón principal por la que se ajustó el limite del eje 'y' y visualizar mejor los diagramas ), fuera de ello, la mayoría de las consolas tienen sus cajas de tamaños similares y en alturas similares, por lo que sus medias no llegan a variar tanto visualmente hablando, ya que se ve que en cuanto a las ventas promedio, la mayoría de los juegos no superan los 2 millones de ventas.

Comentario del revisor (1ra Iteración)

Como sugerencia, cuando tengas gráficas *box-plot* con cajas muy "estrechas" debido a la presencia de numerosos datos atípicos, considera incluir una versión adicional usando escala logarítmica o ajustando el rango del eje Y. Esto facilitará la comparación y el análisis de los datos

#veremos cómo las reseñas de usuarios y profesionales afectan las ventas de una plataforma popular

#en este caso, la wii

#primero filtraremos los datos para la consola

df\_wii = df[df['platform'] == 'Wii']

#graficaremos la relación entre la puntuación de usuarios y las ventas totales

#dado que el maximo en userscore es 10 y en criticscore es 100, haremos dos graficos diferentes

#para una mejor visualización de la relación haremos un gráfico de dispersión

#gráfico de ventas contra puntuación de usuarios

plt.figure(figsize=(10, 6))

sns.scatterplot(x='user\_score', y='total\_sales', data=df\_wii)

plt.title('Ventas vs puntuación de usuarios para Wii')

plt.grid(True)

plt.ylim(0,15) # Establece el límite a 15 millones en las ventas

plt.show()

#gráfico de ventas contra puntuación de la crítica

plt.figure(figsize=(10, 6))

sns.scatterplot(x='critic\_score', y='total\_sales', data=df\_wii)

plt.title('Ventas vs puntuación de crítica para Wii')

plt.ylim(0,15) # Establece el límite a 15 millones en las ventas

plt.grid(True)

plt.show()

# Calculamos la correlación de las reseñas de usuarios con las ventas

#nos apoyamos del método corr()

corr\_users = df\_wii['total\_sales'].corr(df\_wii['user\_score'])

print(f"Correlación entre ventas y reseña de usuarios: {corr\_users}")

# Calcula la correlación de las reseñas de la crítica con las ventas

corr\_critics = df\_wii['total\_sales'].corr(df\_wii['critic\_score'])

print(f"Correlación entre ventas y reseña de críticos: {corr\_critics}").



Gráfico, Gráfico de dispersão

O conteúdo gerado por IA pode estar incorreto.

Realmente los valores de correlación son demasiado pequeños, la que existe entre ventas y reseñas de usuarios es de apenas 0.11 y la que hay entre ventas y reseñas de críticos es de solo 0.17, esa cercanía a cero solo nos indica que no hay verdaderamente una correlación, y por más que en las gráficas se pueda observar que a medida que aumenta la puntuación de la crítica aumenta la cantidad de juegos, solo nos dice que la consola se enfocó en hacer juegos de buena calidad que llegaban a muy buenas puntuaciones (esto se nota más en el gráfico que compara con la puntuación de los usuarios), sin embargo, la altura de las ventas se mantiene casi de forma regular, la mayoría de las ventas están entre 0 y 2 millones, y si bien las mayores ventas están ubicadas en puntuaciones muy altas, no es indicativo de nada porque hay juegos de puntuaciones arriba de 80 que de igual manera no superan esas 2 millones de ventas.

#comparemos las ventas de los mismos juegos de la wii en otras plataformas.

# juntamos los juegos de la wii con los de otras plataformas en un merge sobre los nombres

# agrregamos los suffixes para distinguir columnas

comparacion = df\_wii.merge( df, on='name', suffixes=('\_wii', '\_other') )

# Filtramo para que la columna plataform\_other no tenga juegos de la wii

comparacion = comparacion[comparacion['platform\_other'] != 'Wii']

print(comparacion[['name', 'platform\_wii', 'total\_sales\_wii', 'total\_sales\_other', 'platform\_other']].head())

#graficamos una disperción entre las ventas de wii y las ventas de otras plataformas

comparacion.plot(x='total\_sales\_wii', y='total\_sales\_other', kind='scatter')

plt.xlabel("Ventas en wii")

plt.ylabel("Ventas en otras plataformas")

plt.xlim(0,5) # Establece el límite a 5 millones en las ventas de wii

plt.ylim(0,5) # Establece el límite a 5 millones en las ventas de otras

plt.title("Comparación de ventas")

plt.axline((0,0), slope=1, color='red', linestyle='--') # Línea de similitud

plt.show()

#calculamos correlación y numero de ventas superiores en otras plataformas y en la wii

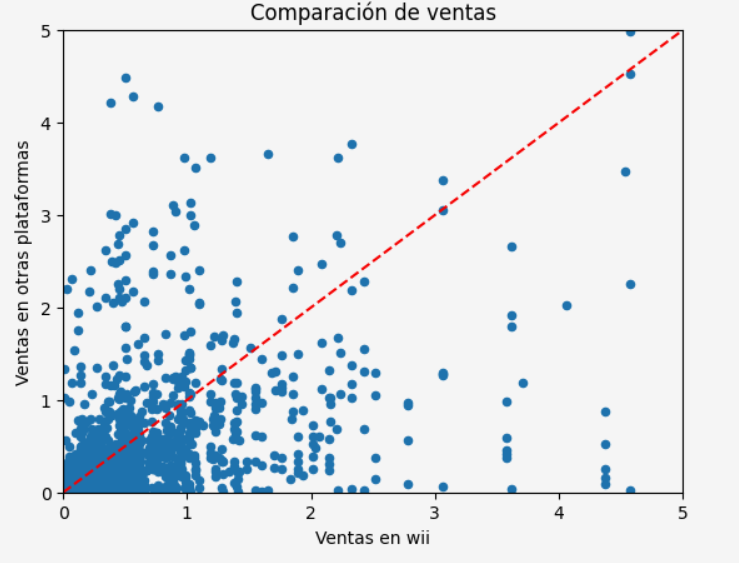
print(comparacion['total\_sales\_wii'].corr(comparacion['total\_sales\_other']))

other\_superior = (comparacion['total\_sales\_other'] > comparacion['total\_sales\_wii']).sum()

wii\_superior= (comparacion['total\_sales\_other'] < comparacion['total\_sales\_wii']).sum()

print("Juegos con más ventas en otras plataformas:", other\_superior )

print("Juegos con más ventas en Wii:", wii\_superior)



Con esto podemos notar que hay algo de variabilidad, ya que podemoms notar que la mayoría de los juegos tienen ventas bastante bajas porque hay una concentración de los puntos cerca de 0–1.25 millones y son pocos los juegos que llegan a tener puntos en posiciones muy arriba o muy a la derecha favoreciendo a una plataforma diferente a la wii o a la misma wii.

Pero en cuanto a la similitud vemos que la gráfica no puede ayudarnos por simple vista a saber si hay más juegos que se encuentran arriba de la línea roja o debajo, sin embargo el conteo nos indica que el wii tiene más juegos con más ventas que las demás plataformas. Además de que su correlación de 0.31 está más cerca del 0 que del 1 por lo que es muy pequeña realmente no hay una.

#distribución general de los juegos por genero

#agrupamos por genero y sumamos el total de las ventas, organizamos en orden descendente

df\_games=df.groupby('genre')['total\_sales'].sum().sort\_values(ascending=False)

#realizamos un gráfico de barras para observar cómo se distinguen las ventas por genero

df\_games.plot(kind='bar', x='genre', y='total\_sales')

plt.xlabel('Género')

plt.ylabel('Ventas totales')

plt.title('Ventas totales por género')

plt.show()

Gráfico, Gráfico de barras

O conteúdo gerado por IA pode estar incorreto.

Con esto podemos observar que los juegos con mejores ventas son los de género de acción, deportes y disparos, dada a su alta popularidad llegaron a superar hasta los 1000 millones dejando en claro que son los generos predilectos para los juegos más rentables en ventas globales, en contraste con aquellos con menos ventas que son los de estrategia, aventura y rompecabezas que parecen llegar a lo mucho a los 250 millones, la diferencia es tan clara que podríamos generalizar sobre el éxito que puede llegar a tener el desarrollo de un juego dependiendo de su género.

#creación de un perfil de usuario para cada región

#Región Norteamérica

#Obtención de sus cinco plataformas principales

platforms\_na = df.groupby('platform')['na\_sales'].sum()

top\_platforms\_na= platforms\_na.sort\_values(ascending=False).head()

print(top\_platforms\_na)

print('/////////////')

#Obtención de sus cinco géneros principales

genre\_na = df.groupby('genre')['na\_sales'].sum()

top\_genre\_na= genre\_na.sort\_values(ascending=False).head()

print(top\_genre\_na)

#Veremos como las clasificaciones de ESRB afectan a las ventas

esrb\_na = df.groupby('rating')['na\_sales']

top\_esrb\_na=esrb\_na.sum().sort\_values(ascending=False)

#Realizaremos un gráfico de barras

top\_esrb\_na.plot(kind='bar', x='rating', y='na\_sales')

plt.xlabel("Clasificación ESRB")

plt.ylabel("Ventas en NA ")

plt.title("Impacto de ESRB en ventas en Norteamérica")

plt.show()

Gráfico, Gráfico de barras, Histograma

O conteúdo gerado por IA pode estar incorreto.

#Región Europa

#Obtención de sus cinco plataformas principales

platforms\_eu = df.groupby('platform')['eu\_sales'].sum()

top\_platforms\_eu= platforms\_eu.sort\_values(ascending=False).head()

print(top\_platforms\_eu)

print('/////////////')

#Obtención de sus cinco géneros principales

genre\_eu = df.groupby('genre')['eu\_sales'].sum()

top\_genre\_eu= genre\_eu.sort\_values(ascending=False).head()

print(top\_genre\_eu)

#Veremos como las clasificaciones de ESRB afectan a las ventas

esrb\_eu = df.groupby('rating')['eu\_sales']

top\_esrb\_eu=esrb\_eu.sum().sort\_values(ascending=False)

#Realizaremos un gráfico de barras

top\_esrb\_eu.plot(kind='bar', x='rating', y='eu\_sales')

plt.xlabel("Clasificación ESRB")

plt.ylabel("Ventas en EU ")

plt.title("Impacto de ESRB en ventas en Europa")

plt.show()

Gráfico, Histograma

O conteúdo gerado por IA pode estar incorreto.

#Región Japón

#Obtención de sus cinco plataformas principales

platforms\_jp = df.groupby('platform')['jp\_sales'].sum()

top\_platforms\_jp= platforms\_jp.sort\_values(ascending=False).head()

print(top\_platforms\_jp)

print('/////////////')

#Obtención de sus cinco géneros principales

genre\_jp = df.groupby('genre')['jp\_sales'].sum()

top\_genre\_jp= genre\_jp.sort\_values(ascending=False).head()

print(top\_genre\_jp)

#Veremos como las clasificaciones de ESRB afectan a las ventas

esrb\_jp = df.groupby('rating')['jp\_sales']

top\_esrb\_jp=esrb\_jp.sum().sort\_values(ascending=False)

#Realizaremos un gráfico de barras

top\_esrb\_jp.plot(kind='bar', x='rating', y='jp\_sales')

plt.xlabel("Clasificación ESRB")

plt.ylabel("Ventas en Japón")

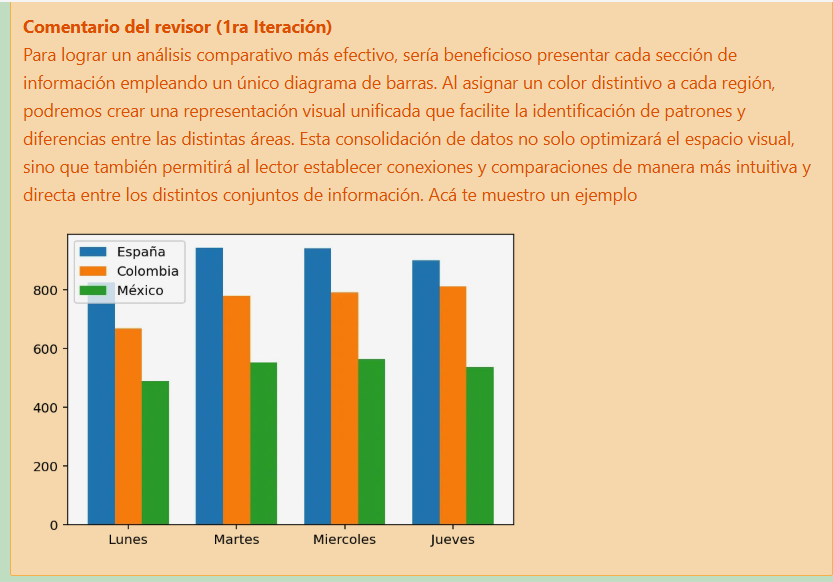
plt.title("Impacto de ESRB en ventas en Japón")

plt.show()

Gráfico, Histograma

O conteúdo gerado por IA pode estar incorreto.

Podemos ver que para cada región ya sea NA, UE o JP las cinco plataformas principales cuentan con la PS2 en su s listas confirmando la popularidad mundial de esta misma, de ahí consolas Wii, PS, DS, PS3, X360 . En cuanto los géneros los cinco géneros principales de cada región tienen en común la presencia de los generos de acción, deportes y misc (teniendo los dos primeros en el top 3 para las tres regiones), manteniendo un poco más de similitudes en las preferencias. Gracias a las gráficas de barras podemos determinar también que las clasificaciones de ESRB afectan a las ventas en las tres regiones y de una manera bastante similar ya que las clasificaciones E,T y M son las tres principales clasificaciones que tienen más ventas sobre todas las demás, en su contraste están las clasificaciones AO,EC y RP que tienen las ventas más bajas y en algunos casos ni siquiera llegan a presencias ventas en absoluto. Con esto podemos concluir que las ventas por región son afectadas cuando se hablan de cuestiones de consolas, cuando son cuestiones de géneros de los juegos hay pocas variaciones en los tipos de genéros en las listas de tops por región por lo que son levemente afectados, sin embargo cuando son por cuestiones de clasificaciones ESRB la región no importa, el comportamiento es muy similar.



#primera prueba de hipótesis:

#Las calificaciones promedio de los usuarios para las plataformas Xbox One y PC son las mismas.

#tomamos las calificaciones de los usuarios en el filtrado por plataformas y borramos valores ausentes

xbox\_userscore = df[df['platform'] == 'XOne']['user\_score'].dropna()

pc\_userscore = df[df['platform'] == 'PC']['user\_score'].dropna()

# Realizamos prueba de Levene

stat\_levene, p\_levene = levene(xbox\_userscore, pc\_userscore)

print("Prueba de Levene p-value:", p\_levene)

# Prueba t para dos muestras independientes

alpha = 0.05

results = st.ttest\_ind(xbox\_userscore, pc\_userscore, equal\_var=True)

# imprimimos resultado pvalue

print('valor-p: ', results.pvalue)

# rechaza la hipótesis solo si la media muestral es significativamente menor que el valor propuesto

if (results.pvalue < alpha) :

print("Rechazamos la hipótesis nula")

else:

print("No podemos rechazar la hipótesis nula")

#segunda prueba de hipótesis:

#Las calificaciones promedio de los usuarios para los géneros de Acción y Deportes son diferentes.

#tomamos las calificaciones de los usuarios en el filtrado por géneros y borramos valores ausentes

action\_userscore = df[df['genre'] == 'Action']['user\_score'].dropna()

sport\_userscore = df[df['genre'] == 'Sports']['user\_score'].dropna()

# Realizamos prueba de Levene

stat\_levene, p\_levene = levene(action\_userscore, sport\_userscore)

print("Prueba de Levene - p-value:", p\_levene)

# Prueba t para dos muestras independientes

alpha = 0.05

results = st.ttest\_ind(action\_userscore, sport\_userscore , equal\_var=False)

# imprimimos resultado pvalue

print('valor-p: ', results.pvalue)

# rechaza la hipótesis solo si la media muestral es significativamente menor que el valor propuesto

if (results.pvalue < alpha) :

print("Rechazamos la hipótesis nula")

else:

print("No podemos rechazar la hipótesis nula")

El criterio para realizar las pruebas t para dos muestras independientes asumiendo que las varianzas son iguales se dió en la primera prueba de hipótesis mientras que en la segunda fue con la evidencia de varianzas diferentes dado que, las pruebas de Levene en ambos casos dieron un valor p mayor y menor al umbral establecido respectivamente.

Para la primera prueba de hipótesis el resultado de un valor p de 1.3896435533548819e-05 indica por su tan pequeña magnitud casi de cero que se rechaza la hipótesis nula, incluso con un valor umbral de 0.01 se sigue rechazando la hipótesis nula confirmando así que realmente las calificaciones promedio de los usuarios para las plataformas Xbox One y PC no son las mismas.

Para la segunda prueba de hipótesis el resultado de un valor p de 0.11483818791498286 indica por su alto valor que no se puede se rechazar la hipótesis nula, incluso con un valor umbral de 0.1 se sigue manteniendo que no se puede rechazar la hipótesis nula confirmando así que realmente no hay evidencia estadísticamente significativa de que las calificaciones promedio de los usuarios para los géneros de Acción y Deportes sean diferentes.

**Conclusión genral**

Una de las cosas que más me llegaron a costar del proyecto fue el pensar en la forma de lidiar con los valores ausentes del DataFrame que en este caso si habían bastantes, ya que no importa la forma en que lo hagas, todas tienen sus consecuencias, si eliminas las filas con valores NaN, podrías perder una gran cantidad de datos que sí tienen información valiosa en otras columnas y si rellenas los NaN con un valor como ceros, la media o la mediana, se podría distorsionar la distribución de los datos. Y no existe una solución única que funcione para todos los casos. Por eso al final tomé la decisión de considerar como la mejor estrategia no eliminarlos ni rellenarlos de manera indiscriminada, más que nada entender por qué están ausentes y no tomar acción hasta que viera casos en los que me generaran problemas (más que nada porque la mayoría de las funciones ignoran los NaN por defecto) que eventualmente si aparecieron, tal como me pasó en las pruebas de hipótesis, por lo que todo fue cuestión de intentar hacer las conversiones/cálculos y si ese intento marcaba un error o resultados dudosos, entender por qué ocurrió y corregir la causa. El análisis de los datos nos llevó a notar que a lo largo de los años que nos presenta el conjunto de datos sobre videojuegos llega a presentar bastante valores atípicos, sin embargo al final los comportamientos demuestran que el aumento de la calidad de los juegos trae consigo un aumento en el interés de las personsas para querer jugarlos, eso se refleja en el aumento de las ventas cuando un videojuego es de un género más complejo y con más diversidad, o cuando un juego cuenta con mejores valoraciones, o cuando con el paso del tiempo, aparecen nuevas y mejores consolas que cumplen un ciclo de vida donde sus ventas van creciendo, llegando a un pico y bajando hasta ya no generar más.