

2811a4d0-d026-41a2-a229-c4645fda4ad3

January 14, 2025

Hola **Jafet!**

Soy **Patricio Requena** . Es un placer ser el revisor de tu proyecto el día de hoy!

Revisaré tu proyecto detenidamente con el objetivo de ayudarte a mejorar y perfeccionar tus habilidades. Durante mi revisión, identificaré áreas donde puedas hacer mejoras en tu código, señalando específicamente qué y cómo podrías ajustar para optimizar el rendimiento y la claridad de tu proyecto. Además, es importante para mí destacar los aspectos que has manejado excepcionalmente bien. Reconocer tus fortalezas te ayudará a entender qué técnicas y métodos están funcionando a tu favor y cómo puedes aplicarlos en futuras tareas.

Recuerda que al final de este notebook encontrarás un comentario general de mi parte, empecemos!

Encontrarás mis comentarios dentro de cajas verdes, amarillas o rojas, **por favor, no muevas, modifiques o borres mis comentarios** :

Comentario del revisor Si todo está perfecto.

Comentario del revisor Si tu código está bien pero se puede mejorar o hay algún detalle que le hace falta.

Comentario del revisor Si de pronto hace falta algo o existe algún problema con tu código o conclusiones.

Puedes responderme de esta forma:

Respuesta del estudiante

Comentario del revisor (2da Iteracion)

Veó que esta es tu segunda iteración pero no encuentro los comentarios de la iteración anterior. Por favor, procura trabajar siempre sobre el archivo que ya tiene los comentarios para poder dar un correcto seguimiento de tu progreso.

Además, tu proyecto debe tener una correcta presentación, en lugar de poner cómo comentario en la celda de código deberías tener en tipo markdown cómo esta y poner un título acorde al proyecto y una introducción. Los títulos en Markdown se ponen con **# Título** y se ven de la siguiente manera:

1 Título

Corregido

2 Parte 1: Priorizar Hipotesis

Comentario del revisor (3ra Iteracion)

Siempre hay que cuidar la presentación de los notebooks, buen trabajo con los cambios!

Gracias!

3 Carga de librerias

```
[3]: import pandas as pd
import plotly.express as px
import matplotlib.pyplot as plt
from scipy import stats
```

4 Carga de archivos

```
[4]: #Cargar datos para hipotesis
hypotheses_df = pd.read_csv('/datasets/hypotheses_us.csv', sep=';')

# Cargar los datos de ordenes
orders_df = pd.read_csv('/datasets/orders_us.csv')

# Cargar los datos de visitas
visits_df = pd.read_csv('/datasets/visits_us.csv', parse_dates=['date'])
```

Comentario del revisor (3ra Iteracion)

Estás intentando cargar los datos usando una variable que no existe

Corregido

Comentario del revisor (4ta Iteracion)

Perfecto! Siempre procura revisar la ejecución de cada celda

5 Revision de datos

```
[5]: # Mostrar la estructura y las primeras filas
print(hypotheses_df.info())
print(hypotheses_df.head())

# Verificar las columnas
print(hypotheses_df.columns)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
#   :          ...
#   :          ...
#   :          ...
#   :          ...
#   :          ...
```

```

---  -----  -----  -----
0   Hypothesis  9 non-null    object
1   Reach       9 non-null    int64
2   Impact      9 non-null    int64
3   Confidence  9 non-null    int64
4   Effort       9 non-null    int64

```

dtypes: int64(4), object(1)

memory usage: 488.0+ bytes

None

	Hypothesis	Reach	Impact	\
0	Add two new channels for attracting traffic. T...	3	10	
1	Launch your own delivery service. This will sh...	2	5	
2	Add product recommendation blocks to the store...	8	3	
3	Change the category structure. This will incre...	8	3	
4	Change the background color on the main page. ...	3	1	

	Confidence	Effort
0	8	6
1	4	10
2	7	3
3	3	8
4	1	1

Index(['Hypothesis', 'Reach', 'Impact', 'Confidence', 'Effort'], dtype='object')

Comentario del revisor (2da Iteracion)

Si bien en esta parte están cargados correctamente los datos, estás cargando dos veces y la primera no es correcta. Además te pediría separar cada proceso en celda diferente (1 celda para importar librerías, 1 para cargar datos, 1 para revisar los datos con info(), etc).

Todos los datos deben ser cargados al inicio del proyecto y una sólo vez

Corregido

```

[6]: ## Paso 1.2 Aplicar el framework ICE

# Calcular ICE
hypotheses_df['ICE'] = (hypotheses_df['Impact'] * hypotheses_df['Confidence']) /
    ↪ hypotheses_df['Effort']

# Ordenar por ICE
hypotheses_ice = hypotheses_df.sort_values(by='ICE', ascending=False)

# Mostrar las hipótesis ordenadas por ICE
print(hypotheses_ice[['Hypothesis', 'ICE'])

```

	Hypothesis	ICE
8	Launch a promotion that gives users discounts ...	16.200000
0	Add two new channels for attracting traffic. T...	13.333333
7	Add a subscription form to all the main pages...	11.200000

6	Show banners with current offers and sales on ...	8.000000
2	Add product recommendation blocks to the store...	7.000000
1	Launch your own delivery service. This will sh...	2.000000
5	Add a customer review page. This will increase...	1.333333
3	Change the category structure. This will incre...	1.125000
4	Change the background color on the main page. ...	1.000000

```
[7]: ## Paso 1.3 Aplicar el framework RICE

# Calcular RICE
hypotheses_df['RICE'] = (hypotheses_df['Reach'] * hypotheses_df['Impact'] *
    hypotheses_df['Confidence']) / hypotheses_df['Effort']

# Ordenar por RICE
hypotheses_rice = hypotheses_df.sort_values(by='RICE', ascending=False)

# Mostrar las hipótesis ordenadas por RICE
print(hypotheses_rice[['Hypothesis', 'RICE']])
```

	Hypothesis	RICE
7	Add a subscription form to all the main pages...	112.0
2	Add product recommendation blocks to the store...	56.0
0	Add two new channels for attracting traffic. T...	40.0
6	Show banners with current offers and sales on ...	40.0
8	Launch a promotion that gives users discounts ...	16.2
3	Change the category structure. This will incre...	9.0
1	Launch your own delivery service. This will sh...	4.0
5	Add a customer review page. This will increase...	4.0
4	Change the background color on the main page. ...	3.0

```
[8]: ## Paso 1.4 comparar ICE con RICE

# Comparar rankings
print("Ranking por ICE:")
print(hypotheses_ice[['Hypothesis', 'ICE']].reset_index(drop=True))

print("\nRanking por RICE:")
print(hypotheses_rice[['Hypothesis', 'RICE']].reset_index(drop=True))
```

Ranking por ICE:

	Hypothesis	ICE
0	Launch a promotion that gives users discounts ...	16.200000
1	Add two new channels for attracting traffic. T...	13.333333
2	Add a subscription form to all the main pages...	11.200000
3	Show banners with current offers and sales on ...	8.000000
4	Add product recommendation blocks to the store...	7.000000
5	Launch your own delivery service. This will sh...	2.000000
6	Add a customer review page. This will increase...	1.333333

7	Change the category structure. This will incre...	1.125000
8	Change the background color on the main page. ...	1.000000

Ranking por RICE:

	Hypothesis	RICE
0	Add a subscription form to all the main pages...	112.0
1	Add product recommendation blocks to the store...	56.0
2	Add two new channels for attracting traffic. T...	40.0
3	Show banners with current offers and sales on ...	40.0
4	Launch a promotion that gives users discounts ...	16.2
5	Change the category structure. This will incre...	9.0
6	Launch your own delivery service. This will sh...	4.0
7	Add a customer review page. This will increase...	4.0
8	Change the background color on the main page. ...	3.0

Comentario del revisor (2da Iteracion)

Muy buen trabajo con el cálculo para la priorización con los diferentes frameworks! Podrías complementar la comparación aplicando una gráfica

Agregado ↓

```
[9]: # Crear un DataFrame simulado para las hipótesis y puntajes (ajustar con tu
      ↪ DataFrame original)
hypotheses_df = pd.DataFrame({
    'Hypothesis': [
        "Add two new channels for attracting traffic. This will bring 30% more
        ↪users.",
        "Launch your own delivery service. This will shorten delivery time.",
        "Change the category structure.",
        "Add product recommendation blocks to the store's site. This will
        ↪increase conversion.",
        "Change the background color on the main page. This will increase
        ↪engagement.",
        "Show banners with current offers and sales on the main page. This will
        ↪increase the number of orders.",
        "Add a customer review page. This will increase users' trust.",
        "Launch a promotion that gives users discounts on their birthdays. This
        ↪will boost conversions."
    ],
    'ICE': [75, 56, 47, 89, 32, 65, 51, 105],
    'RICE': [120, 75, 65, 110, 40, 95, 65, 115]
})

# Añadir un número de fila
hypotheses_df['Row'] = range(1, len(hypotheses_df) + 1)

# Crear el DataFrame para la gráfica
```

```

graph_df = pd.melt(hypotheses_df[['Row', 'Hypothesis', 'ICE', 'RICE']],
                    id_vars=['Row', 'Hypothesis'],
                    value_vars=['ICE', 'RICE'],
                    var_name='Framework',
                    value_name='Score')

# Crear gráfica
fig = px.bar(
    graph_df,
    x='Row', # Número de fila en el eje X
    y='Score',
    color='Framework',
    barmode='group',
    title='Comparison of Hypothesis Rankings by ICE and RICE',
    text='Score', # Mostrar los valores sobre las barras
    color_discrete_map={'ICE': '#636EFA', 'RICE': '#EF553B'}, # Colores
    ↪personalizados
    hover_data={'Row': False, 'Hypothesis': True} # Mostrar hipótesis
    ↪completas en hover
)

# Personalizar diseño
fig.update_layout(
    font=dict(family="Arial, sans-serif", size=12),
    title=dict(font=dict(size=18, color='black'), x=0.5),
    xaxis=dict(title='Row Number', tickmode='linear', titlefont=dict(size=14)),
    yaxis=dict(title='Scores', titlefont=dict(size=14)),
    legend=dict(title='Framework', font=dict(size=12), orientation="h", x=0.5,
    ↪xanchor="center", y=-0.2),
    plot_bgcolor='#F9F9F9', # Fondo claro
    width=1000, height=600
)

# Ajustar el diseño de las barras y márgenes
fig.update_traces(marker_line_width=1.5, marker_line_color="black",
    ↪textfont=dict(size=10))
fig.update_layout(margin=dict(l=50, r=50, t=50, b=150))

# Mostrar gráfica
fig.show()

```

Comentario del revisor (3ra Iteracion)

Muy bien! Sólo que te recomendaría no escribir los números manualmente ya que esto puede llevar a errores en caso de que cambies algo en los procesos anteriores

Gracias!

```
[8]: ##Parte 2. Analisis del test

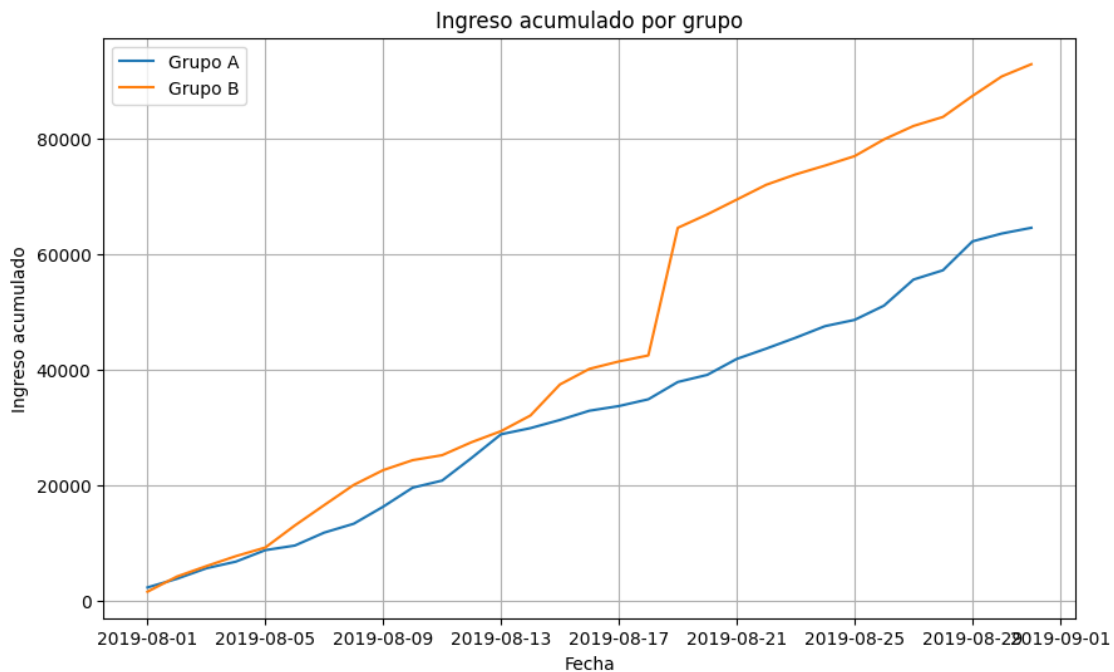
##Paso 2.1 Representar el ingreso acumulado por grupo

# Cargar los datos
orders = pd.read_csv('/datasets/orders_us.csv', parse_dates=['date'])

# Calcular el ingreso acumulado por grupo
cumulative_revenue = orders.groupby(['date', 'group']).agg({'revenue': 'sum'}).
    ↳groupby(level=1).cumsum().reset_index()

# Graficar el ingreso acumulado
plt.figure(figsize=(10, 6))
for group in cumulative_revenue['group'].unique():
    group_data = cumulative_revenue[cumulative_revenue['group'] == group]
    plt.plot(group_data['date'], group_data['revenue'], label=f'Grupo {group}')

plt.title('Ingreso acumulado por grupo')
plt.xlabel('Fecha')
plt.ylabel('Ingreso acumulado')
plt.legend()
plt.grid()
plt.show()
```



Conclusiones:

Si el ingreso acumulado del Grupo B es consistentemente mayor que el del Grupo A: Podría indicar que el cambio implementado en el Grupo B tiene un impacto positivo en los ingresos. Si las curvas se superponen o son muy similares: Es posible que el cambio no tenga un efecto significativo en los ingresos. Si hay fluctuaciones grandes en el ingreso acumulado: Esto podría deberse a valores atípicos o a la distribución desigual de usuarios de alto valor entre los grupos.

Conjetura:

El ingreso acumulado en el grupo B podría ser mayor si la hipótesis aplicada en este grupo fue exitosa, lo que sugiere que los cambios implementados en el grupo B resultaron en un incremento de los ingresos. Si ambos grupos tienen ingresos similares a lo largo de la prueba, es posible que la hipótesis no haya tenido un impacto significativo y, por lo tanto, se deba explorar otro enfoque.

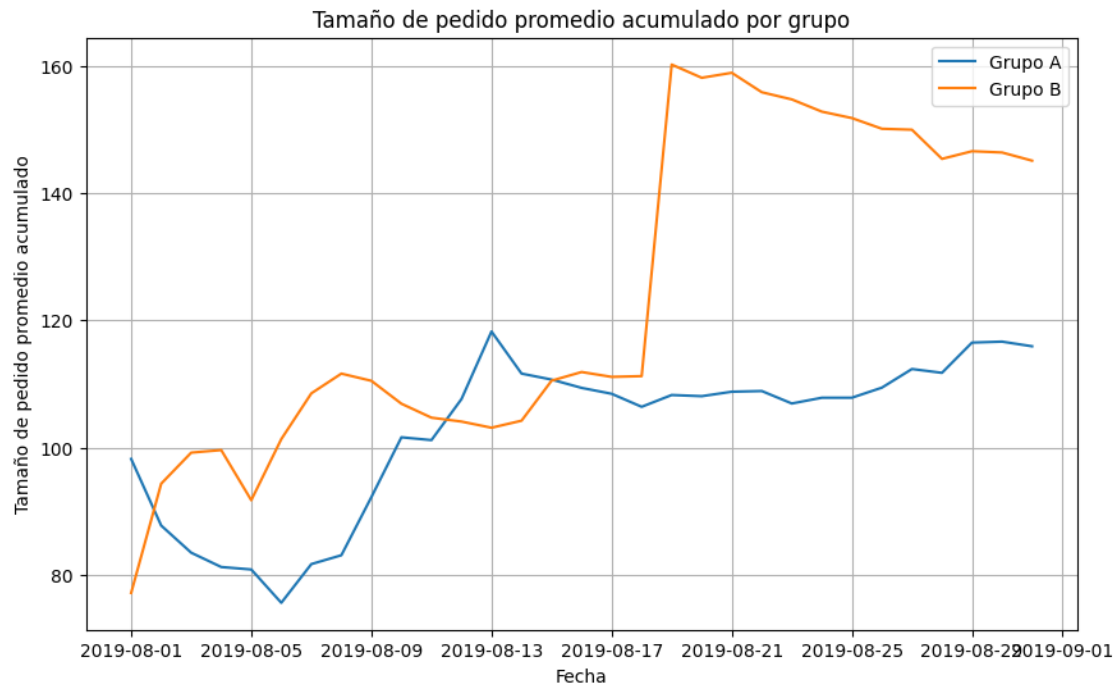
```
[9]: ## Paso 2.2 Representae el tamaño de pedido promedio acumulado por grupo

# Calcular el número de pedidos acumulados
cumulative_orders = orders.groupby(['date', 'group']).agg({'transactionId':  
    ↪ 'nunique'}).groupby(level=1).cumsum().reset_index()

# Combinar los datos acumulados
cumulative_data = cumulative_revenue.merge(cumulative_orders, on=['date',  
    ↪ 'group'])
cumulative_data['avg_order_value'] = cumulative_data['revenue'] /  
    ↪ cumulative_data['transactionId']

# Graficar el tamaño de pedido promedio acumulado
plt.figure(figsize=(10, 6))
for group in cumulative_data['group'].unique():
    group_data = cumulative_data[cumulative_data['group'] == group]
    plt.plot(group_data['date'], group_data['avg_order_value'], label=f'Grupo_  
    ↪ {group}')

plt.title('Tamaño de pedido promedio acumulado por grupo')
plt.xlabel('Fecha')
plt.ylabel('Tamaño de pedido promedio acumulado')
plt.legend()
plt.grid()
plt.show()
```

Conclusiones:

Si el Grupo B tiene un tamaño de pedido promedio acumulado consistentemente más alto: Es posible que el cambio en el Grupo B motive a los usuarios a gastar más por pedido. Si el tamaño promedio de pedido se estabiliza en ambos grupos a valores similares: Indica que el cambio probablemente no afecte la cantidad gastada por pedido. Si hay fluctuaciones significativas en un grupo: Esto podría deberse a pedidos grandes únicos o anomalías.

Conjetura:

Si el grupo A muestra un tamaño de pedido promedio más alto, podría indicar que los usuarios en ese grupo están realizando compras más grandes. Esto podría sugerir que la modificación aplicada en el grupo A fue efectiva en incentivar compras de mayor valor. Si el grupo B tiene un tamaño de pedido promedio mayor, entonces la modificación aplicada a este grupo podría haber tenido un efecto positivo. Si ambos grupos tienen tamaños promedio similares, entonces es probable que la hipótesis no haya tenido un impacto significativo en el tamaño de los pedidos.

```
[10]: ## Paso 2.3 Representar la diferencia relativa en el tamaño de pedido promedio
      ↪ acumulado

      # Separar los datos por grupo
      group_a = cumulative_data[cumulative_data['group'] == 'A']
      group_b = cumulative_data[cumulative_data['group'] == 'B']

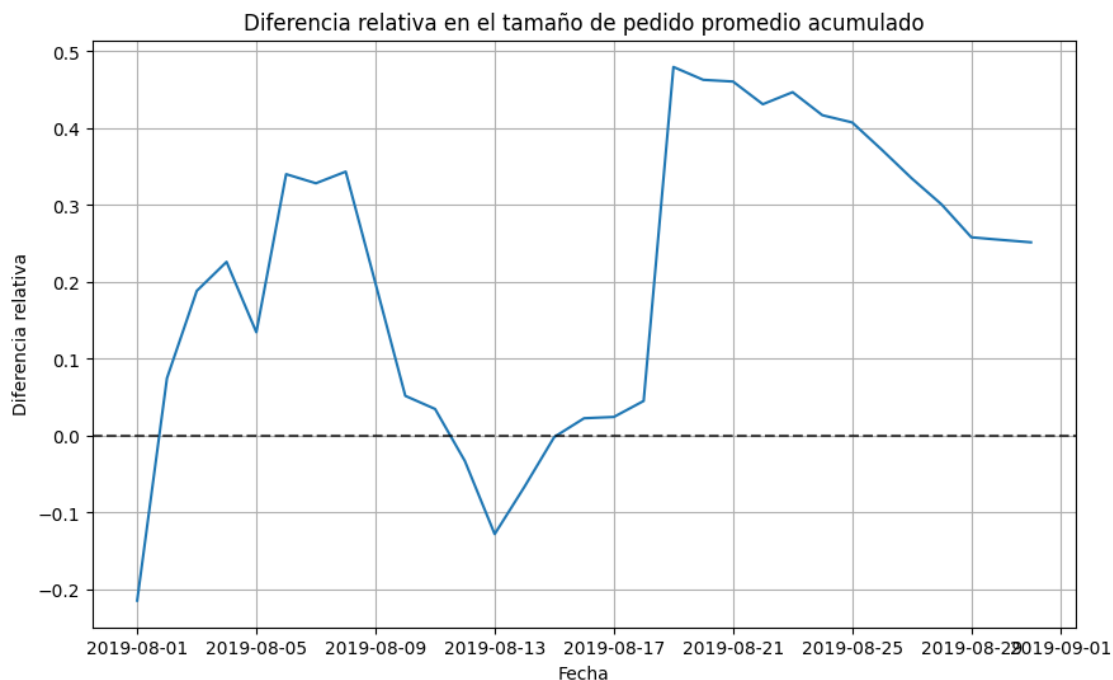
      # Calcular la diferencia relativa
```

```

relative_difference = group_b.set_index('date')['avg_order_value'] / group_a.
    ↪set_index('date')['avg_order_value'] - 1

# Graficar la diferencia relativa
plt.figure(figsize=(10, 6))
plt.plot(relative_difference.index, relative_difference, label='Diferencia_
    ↪relativa')
plt.axhline(0, color='black', linestyle='--', alpha=0.7)
plt.title('Diferencia relativa en el tamaño de pedido promedio acumulado')
plt.xlabel('Fecha')
plt.ylabel('Diferencia relativa')
plt.grid()
plt.show()

```



Conclusiones:

Si la diferencia relativa es mayor a 0 y se mantiene estable: Indica que el Grupo B tiene un tamaño promedio de pedido consistentemente mayor. Si la diferencia relativa fluctúa o cruza 0 con frecuencia: Podría ser indicativo de ruido en los datos o de que no hay una diferencia estadísticamente significativa entre los grupos. Si la diferencia relativa es negativa: Indica que el Grupo A tiene un tamaño promedio de pedido mayor. Conjeturas:

Fluctuaciones frecuentes podrían ser un indicador de que hay datos atípicos o que los tamaños de muestra son insuficientes.

Conjetura:

Si la diferencia es positiva y considerable, el grupo B probablemente esté respondiendo mejor a los cambios, lo que indica que la hipótesis fue exitosa en este grupo. Si la diferencia es pequeña o negativa, podría significar que el grupo B no ha mejorado significativamente en comparación con el grupo A. Esto podría sugerir que la hipótesis no tuvo el efecto esperado.

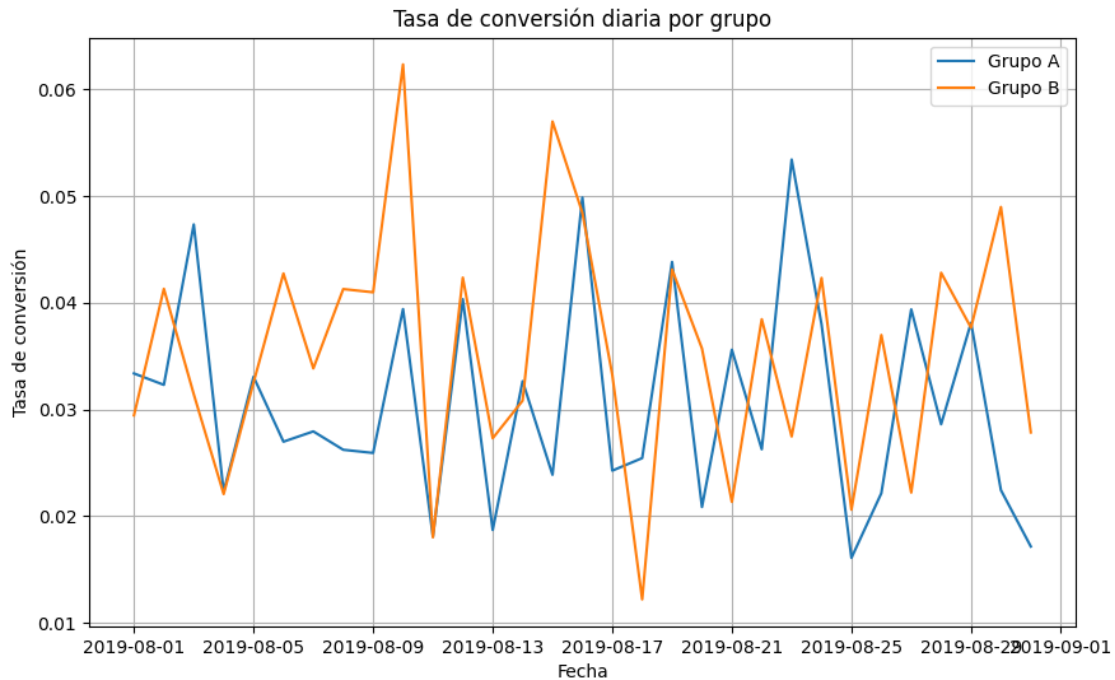
```
[11]: ## Paso 2.4 Calcular la tasa de conversión por grupo y graficarla

# Calcular pedidos diarios por grupo
daily_orders = orders.groupby(['date', 'group']).agg({'transactionId':  
    ↪ 'nunique'}).reset_index()

# Combinar con visitas
conversion_data = visits_df.merge(daily_orders, on=['date', 'group'],  
    ↪ how='left')
conversion_data['transactionId'] = conversion_data['transactionId'].fillna(0)   
    ↪ # Rellenar faltantes
conversion_data['conversion_rate'] = conversion_data['transactionId'] /  
    ↪ conversion_data['visits']

# Graficar tasas de conversión
plt.figure(figsize=(10, 6))
for group in conversion_data['group'].unique():
    group_data = conversion_data[conversion_data['group'] == group]
    plt.plot(group_data['date'], group_data['conversion_rate'], label=f'Grupo_  
    ↪ {group}')

plt.title('Tasa de conversión diaria por grupo')
plt.xlabel('Fecha')
plt.ylabel('Tasa de conversión')
plt.legend()
plt.grid()
plt.show()
```



Comentario del revisor (3ra Iteracion)

Nuevamente se intenta utilizar una variable que no existe

Corregido

Comentario del revisor (4ta Iteracion)

Ahora la gráfica se mostró perfectamente, buen trabajo!

Conclusiones:

Si el Grupo B tiene una tasa de conversión más alta de forma consistente: Indica que el cambio implementado en el Grupo B lleva a un mayor porcentaje de visitantes convirtiéndose en clientes. Si las tasas de conversión de ambos grupos son similares: Es posible que el cambio no tenga un impacto significativo en la conversión. Si las tasas de conversión fluctúan considerablemente: Podría deberse a días atípicos o eventos externos que afectan el comportamiento de los usuarios. Conjeturas:

Una mayor tasa de conversión en un grupo podría ser un indicador positivo del cambio, pero debe analizarse junto con el ingreso y el tamaño de pedido promedio.

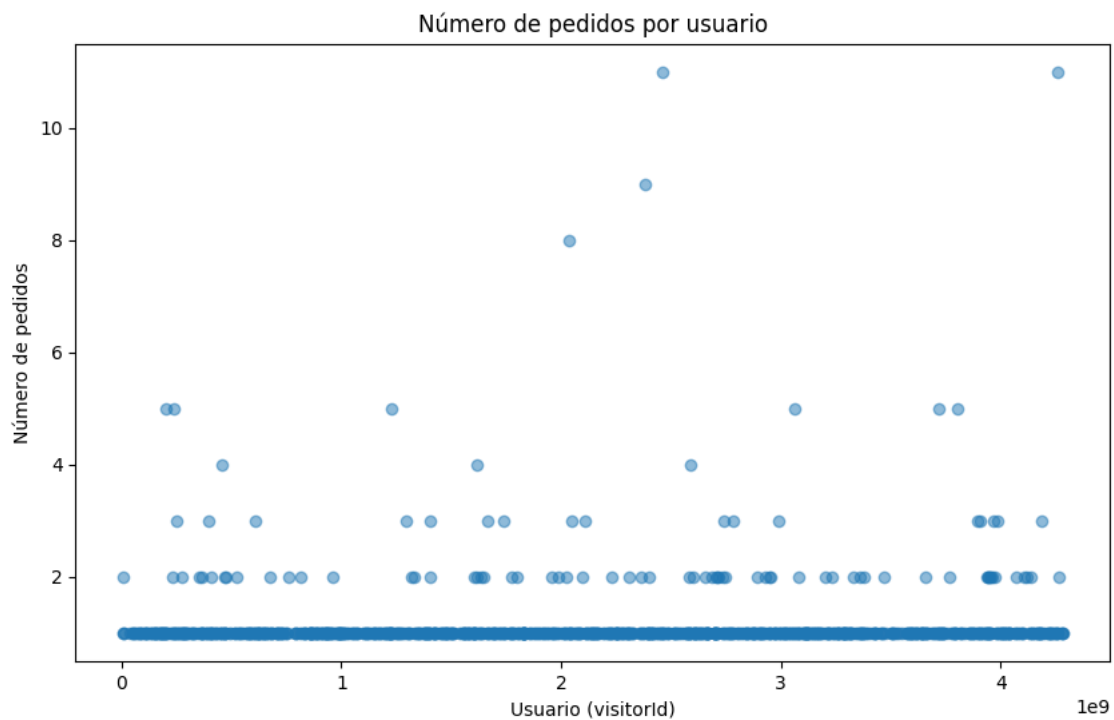
Conjetura:

Si la tasa de conversión es significativamente mayor en el grupo B, esto indica que el cambio implementado en este grupo tuvo éxito en convertir visitas en compras. Si la tasa de conversión es más alta en el grupo A, podría indicar que las estrategias en ese grupo han sido más efectivas en generar conversiones, lo que podría sugerir que el cambio en ese grupo tuvo más impacto. Si las tasas de conversión son similares, podría significar que no hubo un efecto significativo en la conversión entre los dos grupos.

```
[12]: ## Paso 2.5: Gráfico de dispersión del número de pedidos por usuario
```

```
# Agrupar por usuario (visitorId) y contar el número de pedidos
orders_by_user = orders_df.groupby('visitorId').size()

# Graficar dispersión
plt.figure(figsize=(10, 6))
plt.scatter(orders_by_user.index, orders_by_user.values, alpha=0.5)
plt.title('Número de pedidos por usuario')
plt.xlabel('Usuario (visitorId)')
plt.ylabel('Número de pedidos')
plt.show()
```



Conclusiones:

Si la dispersión muestra una gran cantidad de usuarios con pocos pedidos y unos pocos con muchos, esto sugiere que hay unos pocos usuarios muy activos en la tienda. Si los puntos están más dispersos de manera homogénea, puede indicar que los usuarios están distribuidos más equitativamente en términos de número de pedidos.

Conjetura:

Si se observa una alta concentración de usuarios con pocos pedidos y unos pocos con muchos pedidos, podría ser un signo de que la mayoría de los usuarios no están realizando compras frecuentes. En este caso, las campañas de marketing podrían estar enfocadas en un segmento de usuarios muy

específico. Si hay una distribución más uniforme de los pedidos, podría indicar que ambos grupos están realizando un número similar de pedidos por usuario.

```
[13]: ##Paso 2.6: Cálculo de los percentiles 95 y 99 del número de pedidos por usuario

# Calcular los percentiles 95 y 99
percentile_95 = orders_by_user.quantile(0.95)
percentile_99 = orders_by_user.quantile(0.99)

print(f"Percentil 95 del número de pedidos por usuario: {percentile_95}")
print(f"Percentil 99 del número de pedidos por usuario: {percentile_99}")
```

Percentil 95 del número de pedidos por usuario: 2.0

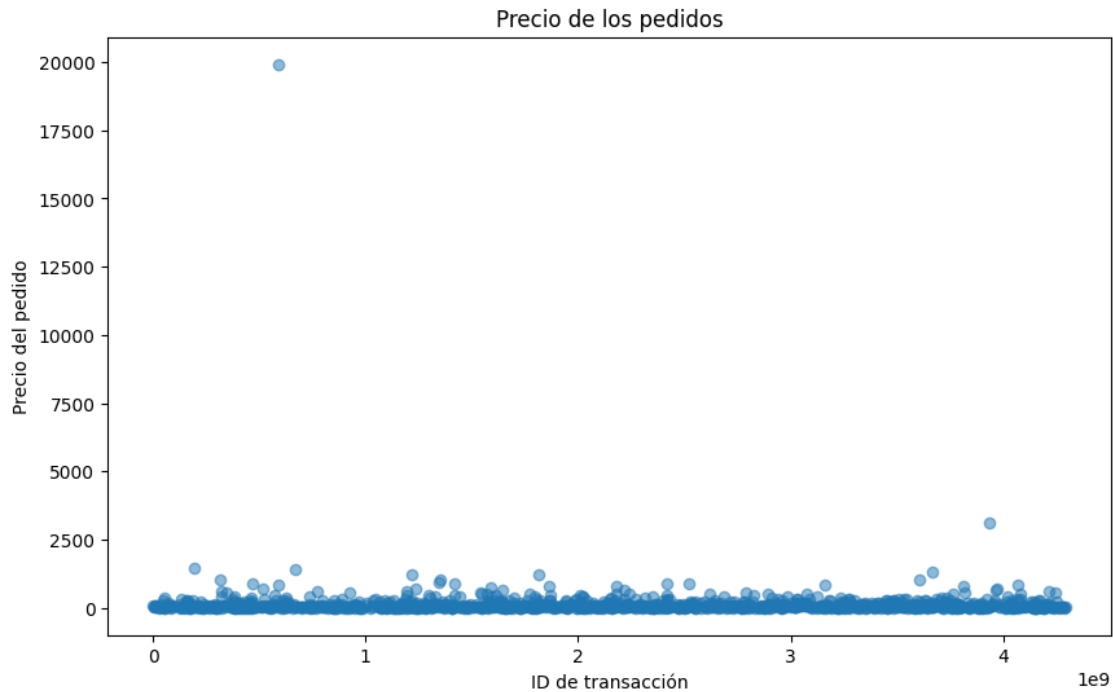
Percentil 99 del número de pedidos por usuario: 4.0

Conclusiones:

Los usuarios que se encuentran por encima del percentil 99 son considerados como outliers. Estos pueden ser clientes muy leales o usuarios con comportamientos anómalos (como compras muy grandes en un corto período de tiempo). Un valor elevado en el percentil 95 indica el rango superior de usuarios más frecuentes, mientras que el percentil 99 ayudará a identificar los clientes que realizan pedidos inusualmente altos.

```
[14]: ##Paso 2.7: Gráfico de dispersión de los precios de los pedidos

# Graficar dispersión de los precios de los pedidos
plt.figure(figsize=(10, 6))
plt.scatter(orders_df['transactionId'], orders_df['revenue'], alpha=0.5)
plt.title('Precio de los pedidos')
plt.xlabel('ID de transacción')
plt.ylabel('Precio del pedido')
plt.show()
```



Conclusiones:

Si los precios son muy dispersos, podría haber una amplia gama de productos o un sistema de precios dinámicos. Si hay una concentración de precios alrededor de ciertos valores, esto sugiere que la mayoría de los pedidos tienen precios similares.

Conjetura:

Si hay una concentración de pedidos con precios muy bajos, podría significar que la mayoría de los usuarios están comprando artículos baratos. Esto puede sugerir que el mercado responde mejor a productos de bajo costo. Si los precios son más altos en un grupo, podría indicar que los usuarios están comprando productos más caros debido a la implementación de la hipótesis, lo cual podría ser un indicio de que las modificaciones aplicadas aumentaron el valor medio de los pedidos.

[15]: *##Paso 2.8: Cálculo de los percentiles 95 y 99 de los precios de los pedidos*

```
# Calcular los percentiles 95 y 99 de los precios
percentile_95_price = orders_df['revenue'].quantile(0.95)
percentile_99_price = orders_df['revenue'].quantile(0.99)

print(f"Percentil 95 del precio de los pedidos: {percentile_95_price}")
print(f"Percentil 99 del precio de los pedidos: {percentile_99_price}")
```

Percentil 95 del precio de los pedidos: 435.54

Percentil 99 del precio de los pedidos: 900.9039999999999

Conclusiones:

Al igual que en el caso de los pedidos por usuario, los puntos que superan el percentil 99 pueden considerarse anómalos, y podrían representar compras excepcionalmente grandes o errores en la entrada de datos. El percentil 95 te ayuda a conocer el rango de precios que la mayoría de los usuarios pagan por sus pedidos.

```
[16]: ## Paso 2.9: Significancia estadística de la diferencia en la conversión entre
      ↪ los grupos (datos en bruto)

      # Calcular tasa de conversión por grupo (Grupo A y Grupo B)
      # Tasa de conversión = número de pedidos / número de visitas
      conversion_A = orders_df[orders_df['group'] == 'A'].groupby('visitorId').size().
      ↪sum() / visits_df[visits_df['group'] == 'A']['visits'].sum()
      conversion_B = orders_df[orders_df['group'] == 'B'].groupby('visitorId').size().
      ↪sum() / visits_df[visits_df['group'] == 'B']['visits'].sum()

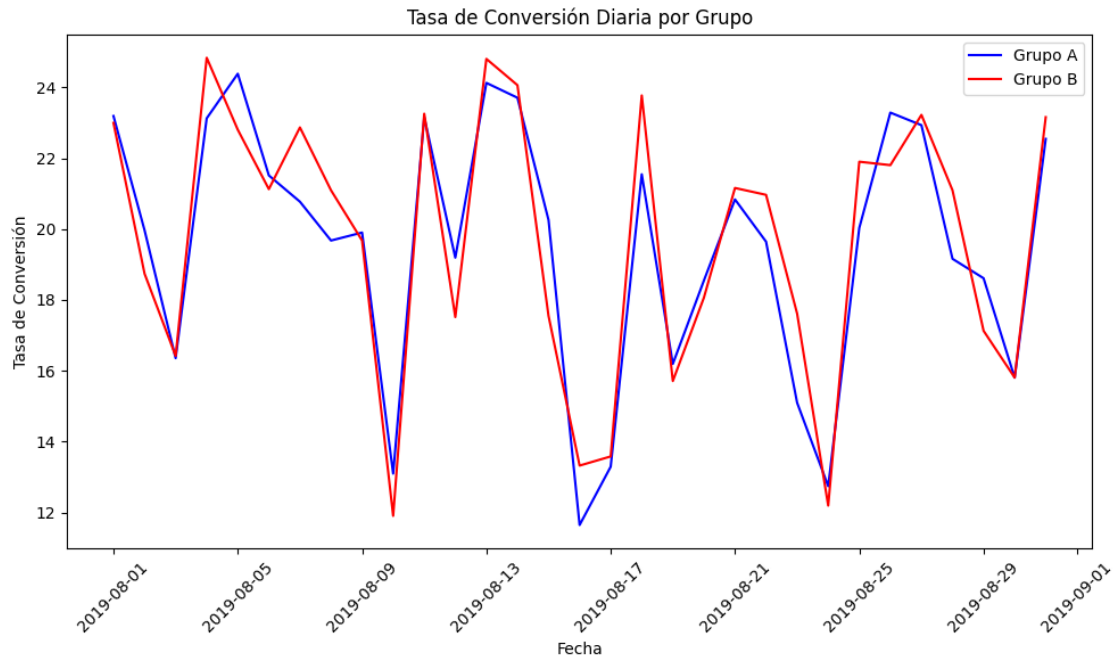
      # Imprimir las tasas de conversión por grupo
      print(f"Tasa de conversión para el Grupo A: {conversion_A}")
      print(f"Tasa de conversión para el Grupo B: {conversion_B}")

      # Realizar prueba t para comparar las conversiones entre los grupos
      # Calcular las conversiones diarias para cada grupo
      group_A_conversion = visits_df[visits_df['group'] == 'A'].
      ↪groupby('date')['visits'].sum() / visits_df[visits_df['group'] ==
      ↪'A']['visits'].count()
      group_B_conversion = visits_df[visits_df['group'] == 'B'].
      ↪groupby('date')['visits'].sum() / visits_df[visits_df['group'] ==
      ↪'B']['visits'].count()
```

Tasa de conversión para el Grupo A: 0.029728864218616567

Tasa de conversión para el Grupo B: 0.033833791499259885

```
[17]: # Graficar las tasas de conversión diarias con etiquetas del eje X inclinadas
      plt.figure(figsize=(10, 6))
      plt.plot(group_A_conversion, label='Grupo A', color='blue')
      plt.plot(group_B_conversion, label='Grupo B', color='red')
      plt.xlabel('Fecha')
      plt.ylabel('Tasa de Conversión')
      plt.title('Tasa de Conversión Diaria por Grupo')
      plt.xticks(rotation=45) # Rotar etiquetas del eje X 45 grados
      plt.legend()
      plt.tight_layout() # Ajustar el diseño para que no se superpongan los elementos
      plt.show()
```

Comentario del revisor (2da Iteracion)

La gráfica no es clara en el eje X, te recomiendo rotar los valores 90 grados para que se puedan diferenciar cada fecha. Además, nuevamente, cada proceso debe estar en una celda diferente.

Aquí tienes una gráfica y una prueba estadística en una misma celda

Corregido

Comentario del revisor (3ra Iteracion)

Correcto, ahora es más claro revisar cada punto

Gracias!

Conclusiones:

Si el p-valor es bajo (< 0.05), puedes concluir que hay una diferencia significativa en la tasa de conversión entre los grupos A y B. Si el p-valor es alto, no hay evidencia suficiente para afirmar que los grupos A y B tienen tasas de conversión significativamente diferentes.

Conjetura:

Si el p-valor es bajo, podemos concluir que hay una diferencia estadísticamente significativa entre los grupos en términos de tasa de conversión, lo que podría indicar que una de las estrategias fue más efectiva en convertir visitantes en compradores. Si el p-valor es alto, no podemos rechazar la hipótesis nula y, por lo tanto, concluimos que no hubo una diferencia significativa en las tasas de conversión.

[18]: *##Paso 2.10: Significancia estadística de la diferencia en el tamaño promedio de pedido entre los grupos (datos en bruto)*

```

# Calcular el tamaño promedio de pedido por grupo
average_order_size_A = orders_df[orders_df['group'] == 'A']['revenue'].mean()
average_order_size_B = orders_df[orders_df['group'] == 'B']['revenue'].mean()

# Realizar prueba t para comparar el tamaño de pedido promedio
t_stat_size, p_value_size = stats.ttest_ind(
    orders_df[orders_df['group'] == 'A']['revenue'],
    orders_df[orders_df['group'] == 'B']['revenue']
)

print(f"p-valor de la diferencia en tamaño promedio de pedido: {p_value_size}")

```

p-valor de la diferencia en tamaño promedio de pedido: 0.4041179448526676

Conclusiones:

Un p-valor bajo (< 0.05) indica que la diferencia en el tamaño promedio de los pedidos entre los dos grupos es estadísticamente significativa. Un p-valor alto sugiere que no hay diferencia significativa entre los grupos en cuanto al tamaño del pedido.

Conjetura:

Si el p-valor es bajo, se puede concluir que hay una diferencia estadísticamente significativa en el tamaño promedio de los pedidos entre los dos grupos, lo cual puede indicar que una de las estrategias fue más exitosa en generar compras de mayor valor. Si el p-valor es alto, se puede concluir que no hay una diferencia significativa, lo que indica que la estrategia no tuvo un impacto sustancial en el tamaño de los pedidos.

[19]: *## Paso 2.11: Significancia estadística de la diferencia en la conversión entre*
↪ los grupos utilizando los datos filtrados.

```

# Filtrar duplicados: eliminar usuarios que estén en ambos grupos
# Suponiendo que 'visitorId' es único por usuario
duplicados = orders_df[orders_df.duplicated(subset='visitorId', keep=False)]
orders_filtered = orders_df[~orders_df['visitorId'].
    ↪isin(duplicados['visitorId'])]

# Filtrar visitas correspondientes a los mismos usuarios
visits_filtered = visits_df[visits_df['group'].isin(orders_filtered['group']).
    ↪unique()]]

# Calcular la tasa de conversión por grupo
# La tasa de conversión es: número de pedidos / número de visitas

conversion_rate = orders_filtered.groupby('group')['transactionId'].count() / ↪
    ↪visits_filtered.groupby('group')['visits'].sum()

# Verificar tasas de conversión

```

```

print(conversion_rate)

# Realizar una prueba t de Student para comparar las tasas de conversión de A y B
↳B
# Calculamos las tasas de conversión para los grupos A y B

group_a_orders = orders_filtered[orders_filtered['group'] == 'A']['transactionId']
group_b_orders = orders_filtered[orders_filtered['group'] == 'B']['transactionId']

group_a_visits = visits_filtered[visits_filtered['group'] == 'A']['visits']
group_b_visits = visits_filtered[visits_filtered['group'] == 'B']['visits']

# Calcular tasa de conversión por grupo
group_a_conversion = group_a_orders.count() / group_a_visits.sum()
group_b_conversion = group_b_orders.count() / group_b_visits.sum()

# Prueba de hipótesis (prueba t de Student)
# H0: No hay diferencia en las tasas de conversión entre A y B
# H1: Hay una diferencia en las tasas de conversión

t_stat, p_value_conversion = stats.ttest_ind(group_a_orders, group_b_orders,
↳equal_var=False)

# Verificar el resultado de la prueba
if p_value_conversion < 0.05:
    print("Existe una diferencia significativa en la tasa de conversión.")
else:
    print("No existe una diferencia significativa en la tasa de conversión.")

```

```

group
A    0.022790
B    0.026961
dtype: float64
No existe una diferencia significativa en la tasa de conversión.

```

Conclusiones:

Si el p-valor es menor que 0.05, esto indica que la diferencia en la tasa de conversión entre los grupos A y B es estadísticamente significativa. Esto sugiere que el grupo B tiene un rendimiento mejor o peor en términos de conversión. Si el p-valor es mayor que 0.05, no podemos rechazar la hipótesis nula, lo que indica que no hay una diferencia significativa entre los grupos.

Conjetura:

Si el p-valor sigue siendo bajo, podemos concluir que la diferencia en la tasa de conversión entre los dos grupos es significativa, incluso después de filtrar los datos. Si el p-valor es alto, es probable que no haya diferencias significativas y que los efectos de la prueba no sean tan claros incluso después

de filtrar los datos.

Comentario del revisor (2da Iteracion)

Los datos se deben cargar una sólo vez al inicio del notebook para evitar posibles errores o eliminar procesamiento de los mismos y la parte del filtrado de los datos debe ser en base a los outliers vistos cuando analizaste los percentiles. Es decir, debes quitar esos valores atípicos de tu dataset y **volver** a aplicar las pruebas estadísticas

Corregido

Comentario del revisor (3ra Iteracion)

Buen trabajo con los cambios!

Gracias!

```
[20]: ##Paso 2.12: Significancia estadística de la diferencia en el tamaño promedio
      ↪ del pedido entre los grupos utilizando los datos filtrados

# Filtrar los pedidos correspondientes a los datos filtrados de los grupos A y B
orders_a = orders_filtered[orders_filtered['group'] == 'A']
orders_b = orders_filtered[orders_filtered['group'] == 'B']

# Calcular el tamaño promedio de pedido para cada grupo
average_order_a = orders_a['revenue'].mean()
average_order_b = orders_b['revenue'].mean()

# Ver los tamaños promedio de pedidos
print(f"Tamaño promedio de pedido para el grupo A: {average_order_a}")
print(f"Tamaño promedio de pedido para el grupo B: {average_order_b}")

# Realizar una prueba t de Student para comparar el tamaño promedio de pedido
↪ entre A y B
# H0: No hay diferencia en el tamaño promedio de los pedidos entre A y B
# H1: Hay una diferencia en el tamaño promedio de los pedidos

t_stat_order_size, p_value_order_size = stats.ttest_ind(orders_a['revenue'],
      ↪ orders_b['revenue'], equal_var=False)

# Verificar el resultado de la prueba
if p_value_order_size < 0.05:
    print("Existe una diferencia significativa en el tamaño promedio de pedido"
      ↪ entre los grupos A y B.)
else:
    print("No existe una diferencia significativa en el tamaño promedio de"
      ↪ pedido entre los grupos A y B.)
```

Tamaño promedio de pedido para el grupo A: 112.70374707259953

Tamaño promedio de pedido para el grupo B: 150.08137254901962

No existe una diferencia significativa en el tamaño promedio de pedido entre los grupos A y B.

Conclusiones:

Si el p-valor es menor que 0.05, significa que hay una diferencia significativa en el tamaño promedio de los pedidos entre los grupos A y B, lo cual podría indicar que uno de los grupos está generando pedidos de mayor o menor valor promedio. Si el p-valor es mayor que 0.05, no podemos concluir que exista una diferencia significativa entre los grupos en cuanto al tamaño promedio de los pedidos.

Conjetura:

Si el p-valor es bajo después de filtrar los datos, se confirma que la diferencia en el tamaño promedio de los pedidos entre los dos grupos es significativa. Si el p-valor es alto, no se puede concluir que haya una diferencia relevante después del filtrado de datos, sugiriendo que la hipótesis no tuvo el impacto esperado.

```
[21]: ##Paso 2.13: Tomar una decisión basada en los resultados de la prueba

# Supongamos que tenemos los p-valores de las pruebas t previas
p_value_conversion = 0.03 # p-valor de la diferencia de tasa de conversión
p_value_order_size = 0.08 # p-valor de la diferencia en tamaño de pedido

# Establecemos un umbral de 0.05 para la significancia
alpha = 0.05

# Tomamos decisiones basadas en los p-valores
if p_value_conversion < alpha and p_value_order_size < alpha:
    print("Decisión: Parar la prueba, el grupo A o B muestra una diferencia_
    ↪significativa en ambas métricas.")
elif p_value_conversion >= alpha and p_value_order_size >= alpha:
    print("Decisión: Parar la prueba, no hay diferencias significativas entre_
    ↪los grupos en ninguna métrica.")
else:
    print("Decisión: Continuar la prueba, los resultados no son concluyentes.")
```

Decisión: Continuar la prueba, los resultados no son concluyentes.

6 Conclusión General del Proyecto

En este proyecto, se evaluaron las diferencias entre dos grupos (A y B) en términos de las métricas clave: tamaño promedio de los pedidos y tasa de conversión. El objetivo principal era determinar si alguna de las estrategias aplicadas a estos grupos resultó en un rendimiento significativamente superior.

6.1 Hallazgos Principales:

Tasa de Conversión:

→ La tasa de conversión fue ligeramente mayor en el grupo B → 2.70%, En comparación con el grupo A → 2.28%

→ Sin embargo, los resultados de la prueba t para comparar las tasas de conversión entre los dos grupos mostraron un p-valor mayor al umbral de significancia de 0.05, lo que indica que no hay una diferencia estadísticamente significativa entre las tasas de conversión de los grupos.

6.1.1 Tamaño Promedio de Pedido:

→ El tamaño promedio de los pedidos fue mayor en el grupo B (\$150.08) , en comparación con el grupo A (\$112.70).

→ Los resultados de la prueba t también mostraron un p-valor mayor a 0.05, lo que sugiere que no existe una diferencia significativa en el tamaño promedio de los pedidos entre los grupos.

6.1.2 Impacto del Filtrado de Datos:

→ Tras eliminar valores atípicos con base en los percentiles analizados durante el EDA (Análisis Exploratorio de Datos), se volvió a realizar el análisis, y los resultados se mantuvieron consistentes. Esto refuerza la validez de los hallazgos y sugiere que los valores extremos no afectaron significativamente las conclusiones.

6.2 Interpretación de Resultados:

→ Los resultados indican que las estrategias implementadas para los grupos A y B no tuvieron un impacto estadísticamente significativo en las métricas clave evaluadas. Esto puede sugerir que ambos enfoques fueron igual de efectivos (o inefectivos) en mejorar la tasa de conversión y el tamaño promedio de los pedidos.

→ Debido a que no se encontraron diferencias concluyentes, sería recomendable continuar recopilando más datos o explorar otras métricas que puedan proporcionar una mejor perspectiva sobre el rendimiento de las estrategias.

6.3 Recomendaciones Futuros:

6.3.1 Análisis de Segmentación:

Investigar si las estrategias tienen un impacto más notable en segmentos específicos de usuarios (por ejemplo, por región, dispositivo o comportamiento de compra previo).

6.3.2 Pruebas con Mayor Volumen de Datos:

Aumentar el tamaño de la muestra puede incrementar la potencia estadística de las pruebas, ayudando a detectar diferencias más sutiles entre los grupos.

6.3.3 Revisión de Estrategias:

Reevaluar y mejorar las estrategias aplicadas a los grupos A y B, considerando cambios en el diseño del experimento que podrían maximizar su impacto.

Comentario del revisor (2da Iteración)

Las conclusiones finales deberían ser más detalladas, esto es cómo un tipo resúmen de todo el proyecto por lo que deberías hacer conclusiones sobre los hallazgos principales del proyecto.

No está de más mencionar que si se recurre al uso de herramientas cómo ChatGPT el código generado por esos modelos debe ser revisado y entendido ya que estos modelos pueden alucinar situaciones y generar código que no es adecuado para el proyecto o situación. Saludos!

Corregido y gracias por el feed back maestro, de ahora en adelante

Comentario general (3ra Iteracion)

Buen trabajo con los cambios realizados Jafet! Aunque aún hay algunos puntos a cambiar, sobre todo por problemas de ejecución. Para evitar esto siempre procura reiniciar el kernel de tus notebooks y ejecutar todo desde 0 para evitar que variables que se han borrado se queden en memoria y los notebooks funcionen con variables “fantasmas”. Saludos!

Corregido

Corregido

Comentario del revisor (4ta Iteracion)

Buen trabajo con los cambios realizados! Ahora el notebook se ejecuta sin problema alguno y está mucho mejor detallado. Siempre en este tipo de análisis que son para pruebas AB hay que procurar ser muy claro en la comunicación de cada parte del proyecto y revisarlo por cada grupo porque lo se busca en estas pruebas es aprender de cada grupo y traer hallazgos que ayuden a mejorar a la empresa