



Nombre de la Unidad de Aprendizaje: Visualización de datos y desarrollo de escritorio.

Número de Evidencia: Producto Integrador de Aprendizaje.

Título de la Evidencia: Reporte de objetivo de análisis enfocado en la base de datos Shoeland.

Programa educativo: LICENCIATURA EN TECNOLOGIAS DE INFORMACION.

Semestre: 4º

Grupo: 43

Equipo: Los Pythonicos

Nombre del Docente: M.C. José Felipe Ramírez Ramírez.

Nombre(s) del Estudiante:

Nombres	Matriculas
CORONADO ARREGUIN CIRO	2059118
GUGLIELMI MARÍN SAÚL	2179817
MATA CORONADO ORLANDO FABIÁN	2012344
RODRIGUEZ ASTRAIN ALAN JHOSEP	2014292
SOLIS CAVAZOS JULIO ALEXANDER	2118842

Índice

Introducción	3
Opiniones (ANECA 6.4)	5
Descripción del caso	9
Objetivo de analítica	9
PRIMERA PARTE	10
Cambio en el tipo de dato fecha:	10
Generación de una nueva columna	10
SEGUNDA PARTE.....	11
Visualización de la estructura de datos	11
Visualización mediante .info()	11
Visualización mediante .shape	11
Visualización mediante .describe()	12
Visualización mediante len()	12
Visualización mediante .dtypes.....	12
Visualización mediante .head().....	13
Visualización mediante .tail()	13
Visualización completa del DataFrame.....	13
TERCERA PARTE	14
Análisis exploratorio de datos.....	14
4. ¿Cuál es el rendimiento de ventas por vendedor?	20
Gráfico de sectores	22
Gráfica scatter plot	23
Hipótesis basadas en las preguntas y análisis:	24
Anexo	27

Link de libreta colab:	27
Reseñas del curso Diseño y Modelado Profesional de Datos para Analítica	28
2059118 - Coronado Arreguin Ciro:	28
2179817 - Guglielmi Marín Saúl:.....	28
2012344 - Mata Coronado Orlando Fabian:.....	29
2014292 - Rodríguez Astrain Alan Jhosep:.....	30
2118842 - Solís Cavazos Julio Alexander:.....	30

Introducción

En la actualidad, las organizaciones enfrentan un entorno altamente competitivo en el que la toma de decisiones informadas resulta esencial para su crecimiento y sostenibilidad. Uno de los activos más valiosos en este proceso es la información, sin embargo, su utilidad depende en gran medida de la calidad de los datos disponibles. Es por esto que la depuración y estandarización de datos se han convertido en pasos críticos dentro del análisis de datos y la inteligencia empresarial.

La depuración de datos, también conocida como limpieza de datos, consiste en identificar y corregir errores, inconsistencias, duplicados y datos faltantes dentro de un conjunto de información. Este proceso asegura que los análisis posteriores se basen en información precisa y confiable, evitando interpretaciones erróneas y decisiones fallidas. Para las organizaciones, especialmente aquellas que gestionan grandes volúmenes de información, como Shoeland en este caso de estudio, contar con datos depurados permite detectar patrones reales de consumo, evaluar el rendimiento de sus puntos de venta y planificar estrategias eficaces.

Este trabajo tiene como base una base de datos de ventas reales, y demuestra cómo, a través de herramientas de visualización y técnicas de análisis exploratorio, se puede transformar información cruda en conocimiento útil. Todo esto parte, indispensablemente, de un adecuado proceso de limpieza y estructuración de los datos, que garantiza la validez y coherencia del análisis. Así,

se resalta la importancia de la depuración de datos como una práctica clave en la analítica moderna orientada a la toma de decisiones empresariales.

Opiniones (ANECA 6.4)

2059118 - Coronado Arreguin Ciro:

Español:

En este producto integrador de aprendizaje pudimos aplicar diversas herramientas útiles para el análisis de datos, haciendo uso de una libreta de Jupyter en Google Colab. El caso de estudio se centró en una empresa dedicada a la venta de calzado, para la cual utilizamos su base de datos con el fin de examinar la información contenida en sus registros.

A lo largo del análisis, logramos identificar datos clave que pueden aportar valor a la toma de decisiones dentro de la empresa. Entre los objetivos más relevantes destacan la identificación del tipo de calzado con mayor volumen de ventas, los meses con mayor actividad comercial, así como los locales que presentan el mejor rendimiento en ventas. Además, fue posible visualizar tendencias, comportamientos del consumidor y posibles áreas de mejora.

Ingles:

In this integrative learning product, we were able to apply various useful tools for data analysis, using a Jupyter notebook in Google Colab. The case study focused on a company dedicated to selling footwear, for which we used its database in order to examine the information contained in its records. Throughout the analysis, we identified key data that can add value to decision-making within the company. Among the most relevant objectives are the identification of the type of footwear with the highest sales volume, the months with the most commercial activity, as well as the locations that show the best sales performance. Additionally, it was possible to visualize trends, consumer behaviors, and potential areas for improvement.

2179817 - Guglielmi Marín Saúl:

Español:

Después de revisar los datos de ventas por tipo de calzado, por mes y por ciudad, me queda claro que hay patrones bastante marcados. Hay ciertos tipos de calzado que dominan las ventas, lo cual probablemente responde a la temporada, la moda o la necesidad del consumidor. Por ejemplo, si vemos que un calzado específico (como las zapatillas) lidera todos los meses, eso ya habla de una demanda constante y fuerte.

Por otro lado, algunas ciudades claramente venden mucho más que otras, lo que puede estar relacionado con el tamaño del mercado, la ubicación del local o incluso con campañas promocionales locales.

Lo que más me llama la atención es cómo las ventas fluctúan mes a mes, y cómo ciertos tipos de calzado crecen en meses puntuales. Esa variación estacional podría ser clave para ajustar inventario, promociones o lanzamientos. Si yo estuviera tomando decisiones con estos datos, sin duda aprovecharía esa información para hacer campañas dirigidas justo cuando más impacto pueden tener.

Ingles:

After reviewing the sales data by type of footwear, by month, and by city, it is clear to me that there are quite marked patterns. There are certain types of footwear that dominate sales, which probably responds to the season, fashion, or consumer need. For example, if we see that a specific footwear type (like sneakers) leads every month, that already indicates a constant and strong demand. On the other hand, some cities clearly sell much more than others, which may be related to market size, store location, or even local promotional campaigns. What stands out to me the most is how sales fluctuate month by month, and how certain types of footwear grow in specific months. That seasonal variation could be key to adjusting inventory, promotions, or launches. If I were making decisions with this data, I would certainly take advantage of that information to run targeted campaigns just when they can have the most impact.

2012344 - Mata Coronado Orlando Fabian:

Español:

El trabajo muestra el manejo del análisis de datos aplicado a una situación real. A través del caso de Shoeland, se logró identificar información clave como los tipos de calzado más vendidos, los locales con mejores resultados y los meses con mayor volumen de ventas. Se usaron bien las herramientas de visualización y las gráficas fueron claras y útiles para interpretar los resultados.

Además, el proceso de limpieza y estandarización de datos estuvo bien explicado, lo que refuerza la importancia de preparar bien la información antes de analizarla. En general, el trabajo fue completo, bien organizado y logra su objetivo principal: mostrar cómo los datos pueden ayudar a tomar mejores decisiones en una empresa.

Ingles:

The work done shows a handling of data analysis applied to a real situation. Through the case of Shoeland, key information was identified such as the best-selling shoe types, the locations with the best results, and the months with the highest sales volume. The visualization tools were used well, and the graphs were clear and useful for interpreting the results. Moreover, the data cleaning and standardization process was well explained, reinforcing the importance of properly preparing information before analyzing it. Overall, the work was comprehensive, well-organized, and achieves its main objective: to show how data can help make better decisions in a company.

2014292 - Rodríguez Astrain Alan Jhosep:

Español:

En este día pudimos entender de mejor forma como se utilizan las funciones y como convertir las columnas del archivo csv que habíamos exportado al coolab para poder hacer diferente modificación para crear conversiones de los datos también entendí de una mejor forma a cómo usar la librería de matplotlib para poder hacer las graficas

Ingles:

Today we were able to better understand how to use functions and how to convert the columns of the CSV file that we had exported to Colab, in order to make different modifications to create data conversions. I also understood better how to use the matplotlib library to create charts.

2118842 - Solís Cavazos Julio Alexander:

Español:

Antes de realizar el análisis, el caso resulta especialmente relevante porque refleja una situación frecuente en muchas organizaciones: disponer de datos acumulados sin procesar y tener la necesidad de transformarlos en información útil para la toma de decisiones. La base de datos proporciona una oportunidad concreta para aplicar procesos de limpieza, estandarización y visualización, lo cual es esencial en cualquier proyecto de analítica.

La naturaleza del problema invita a reflexionar sobre la importancia de la calidad de los datos y de contar con herramientas adecuadas para explorar tendencias, rendimientos y comportamientos dentro de la empresa. Además, trabajar este tipo de casos fomenta el pensamiento crítico y permite simular escenarios reales donde los datos son la base de decisiones estratégicas.

Desde el punto de vista académico y profesional, este ejercicio representa un buen desafío, ya que requiere habilidades técnicas, capacidad de interpretación y una visión estructurada para llegar a conclusiones claras a partir de datos complejos.

Ingles:

Before conducting the analysis, the case is especially relevant as it reflects a common situation in many organizations: having large volumes of unprocessed data and the need to turn it into actionable insights. The dataset presents a concrete opportunity to apply processes such as data cleaning, standardization, and visualization, which are fundamental in any data analytics project.

The nature of the problem highlights the importance of data quality and having the right tools to explore trends, performance, and internal business behaviors. Moreover, working on this kind of case encourages critical thinking and simulates real-world scenarios where data is the foundation for strategic decision-making.

From both an academic and professional perspective, this exercise is a valuable challenge, as it requires technical skills, interpretation capabilities, and a structured mindset to reach clear conclusions from complex information.

Descripción del caso

Shoeland es una empresa especializada en la comercialización de calzado, la cual opera a través de varios puntos de venta físicos. Su catálogo incluye distintos tipos de calzado, con variaciones en diseño, precio, cantidad y costo. En un entorno de alta competencia dentro del sector minorista, la empresa busca aprovechar el análisis de datos para comprender mejor el comportamiento de sus clientes y el desempeño de sus productos y sucursales.

Para ello, se ha recopilado una base de datos con información histórica de ventas, que incluye detalles como el tipo de calzado vendido, la cantidad de unidades, los precios de venta, el costo por producto, la fecha de la transacción y el local donde se realizó. Esta información representa una fuente valiosa para explorar patrones de consumo, estacionalidad en las ventas, preferencias de los clientes y eficiencia de los distintos puntos de venta.

Al analizar estos datos, la empresa espera identificar oportunidades de mejora en sus operaciones comerciales, aumentar la rentabilidad y tomar decisiones más informadas basadas en evidencia empírica.

Objetivo de analítica

El objetivo principal de este análisis es extraer conocimientos significativos a partir de los datos de ventas registrados por Shoeland. Específicamente, se busca responder a las siguientes preguntas clave:

- ¿Cuáles son los productos más vendidos por categoría?
- ¿Existe una correlación entre el precio y la cantidad vendida?
- ¿Hay alguna tendencia estacional en las ventas a lo largo del tiempo?
- ¿Cuál es el rendimiento de ventas por vendedor?

A través de este proceso de analítica descriptiva, se pretende generar una visión integral del comportamiento comercial de la empresa. Los hallazgos obtenidos permitirán respaldar decisiones estratégicas en áreas como inventario, marketing, distribución y gestión de tiendas, con el objetivo de optimizar recursos, aumentar la satisfacción del cliente y mejorar el posicionamiento competitivo de Shoeland en el mercado.

PRIMERA PARTE

Cambio en el tipo de dato fecha:

```
1 shoe['fecha_venta'] = pd.to_datetime(shoe['fecha_venta'], errors='coerce')
2 print("convertido")
3 shoe.dtypes
```

convertido	object
ticket	int64
item	int64
id_producto	object
producto	object
fecha_venta	datetime64[ns]
cantidad	int64
id_tamaño	object

Se convierte la fecha de tipo “object” a tipo “datetime64” esto a coerde a los trabajos de analítica que se van a realizar a lo largo del proyecto.

Generación de una nueva columna

```
1 def clasificar_cliente(cantidad):
2     if cantidad == 1:
3         return 'Comprador ocasional'
4     elif cantidad in [2, 3]:
5         return 'Comprador frecuente'
6     else:
7         return 'Comprador mayorista'
8
9 |
10 shoe['tipo_cliente'] = shoe['cantidad'].apply(clasificar_cliente)
```

Se crea la función clasificar_cliente con el propósito de que apartir de la columna cantidad se nos muestra si nuestro cliente es de tipo ocasional, frecuente o mayorista.

SEGUNDA PARTE

Visualización de la estructura de datos

Visualización mediante .info()

```
1 shoe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6759 entries, 0 to 6758
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ticket          6759 non-null    int64  
 1   item             6759 non-null    int64  
 2   id_producto     6759 non-null    object  
 3   producto        6759 non-null    object  
 4   fecha_venta     2669 non-null    datetime64[ns]
 5   cantidad         6759 non-null    int64  
 6   id_tamaño       6759 non-null    object  
 7   tamaño           6759 non-null    object  
 8   precio_unitario 6759 non-null    int64  
 9   costo_unitario  6759 non-null    int64  
 10  tipo_calzado   6759 non-null    object  
 11  id_local         6759 non-null    int64  
 12  local            6759 non-null    object  
 13  tipo_local       6759 non-null    object  
 14  id_ciudad        6759 non-null    object  
 15  ciudad           6759 non-null    object  
 16  total_venta      6759 non-null    int64  
 17  total_costo      6759 non-null    int64  
 18  año              6759 non-null    int64  
 19  mes              6759 non-null    object  
dtypes: datetime64[ns](1), int64(9), object(10)
memory usage: 1.0+ MB
```

Visualización mediante .shape

```
1 shoe.shape
```

```
(6759, 20)
```

Visualización mediante .describe()

1 shoe.describe()											
	ticket	item	fecha_venta	cantidad	precio_unitario	costo_unitario	id_local	total_venta	total_costo	año	
count	6759.000000	6759.000000		2669	6759.000000	6759.000000	6759.000000	6759.000000	6759.000000	6759.000000	
mean	52276.246634	1.032697	2020-12-31 20:37:08.250280960	1.010061	1444.123391	410.119692	10.026779	1458.219411	414.206391	2020.545791	
min	10004.000000	1.000000	2020-01-01 00:00:00	1.000000	801.000000	380.000000	1.000000	801.000000	380.000000	2020.000000	
25%	31354.500000	1.000000	2020-07-09 00:00:00	1.000000	1153.000000	388.000000	5.000000	1153.000000	388.000000	2020.000000	
50%	53481.000000	1.000000	2021-01-10 00:00:00	1.000000	1448.000000	396.000000	10.000000	1448.000000	396.000000	2021.000000	
75%	72572.000000	1.000000	2021-07-02 00:00:00	1.000000	1681.000000	424.000000	15.000000	1738.000000	424.000000	2021.000000	
max	98979.000000	4.000000	2021-12-12 00:00:00	4.000000	2102.000000	460.000000	19.000000	6099.000000	1548.000000	2021.000000	
std	23884.878106	0.195304		NaN	0.121227	336.875248	25.479469	5.483600	381.665233	55.323252	0.497936

Visualización mediante len()

```
1 len(shoe)
```

6759

Visualización mediante .dtypes

1 shoe.dtypes	
	0
ticket	int64
item	int64
id_producto	object
producto	object
fecha_venta	datetime64[ns]
cantidad	int64
id_tamaño	object
tamaño	object
precio_unitario	int64
costo_unitario	int64
tipo_calzado	object
id_local	int64
local	object
tipo_local	object
id_ciudad	object
ciudad	object

Visualización mediante .head()

1 shoe.head(3)																				
	ticket	item	id_producto	producto	fecha_venta	cantidad	id_tamaño	tamaño	precio_unitario	costo_unitario	...	id_local	local	tipo_local	id_ciudad	ciudad	total_venta			
0	10633	1	C001	COMFORT VERONA	2020-07-01	1	22M	22 Y MEDIO	1311	393	...	11	ARAGON	POPULAR	CMX	CIUDAD DE MÉXICO	1			
1	11031	1	C001	COMFORT VERONA	2020-11-01	1	22M	22 Y MEDIO	1311	393	...	17	CRUZ DEL SUR	POPULAR	PUE	PUEBLA	1			
2	11129	1	C001	COMFORT VERONA	2020-12-01	1	22M	22 Y MEDIO	1311	393	...	16	TLAQUEPAQUE	POPULAR	GDL	GUADALAJARA	1			

3 rows × 21 columns

Visualización mediante .tail()

1 shoe.tail(3)																				
	ticket	item	id_producto	producto	fecha_venta	cantidad	id_tamaño	tamaño	precio_unitario	costo_unitario	...	id_local	local	tipo_local	id_ciudad	ciudad	total_venta			
6756	89944	1	U002	MINIMAL ECHO	NaT	1	30M	30 Y MEDIO	1069	460	...	14	PABELLON	ESTÁNDAR	GDL	GUADALAJARA	1			
6757	98031	1	U002	MINIMAL ECHO	NaT	1	18M	18 Y MEDIO	1069	460	...	15	CENTRO MAGNO	ESTÁNDAR	GDL	GUADALAJARA	1			
6758	90729	1	U002	MINIMAL ECHO	NaT	1	18M	18 Y MEDIO	1069	460	...	19	ANGELOPOLIS	ESTÁNDAR	PUE	PUEBLA	1			

Visualización completa del DataFrame

1 shoe																				
	ticket	item	id_producto	producto	fecha_venta	cantidad	id_tamaño	tamaño	precio_unitario	costo_unitario	...	id_local	local	tipo_local	id_ciudad	ciudad	total_venta			
0	10633	1	C001	COMFORT VERONA	2020-07-01	1	22M	22 Y MEDIO	1311	393	...	11	ARAGON	POPULAR	CMX	CIUDAD DE MÉXICO	1			
1	11031	1	C001	COMFORT VERONA	2020-11-01	1	22M	22 Y MEDIO	1311	393	...	17	CRUZ DEL SUR	POPULAR	PUE	PUEBLA	1			
2	11129	1	C001	COMFORT VERONA	2020-12-01	1	22M	22 Y MEDIO	1311	393	...	16	TLAQUEPAQUE	POPULAR	GDL	GUADALAJARA	1			
3	11794	1	C001	COMFORT VERONA	NaT	1	24N	24 NORMAL	1311	393	...	14	TLAQUEPAQUE	POPULAR	GDL	GUADALAJARA	1			
4	15046	1	C001	COMFORT VERONA	NaT	1	18M	18 Y MEDIO	1311	393	...	14	TLAQUEPAQUE	POPULAR	GDL	GUADALAJARA	1			
...			
6754	89454	1	U002	MINIMAL ECHO	2021-08-12	1	28M	28 Y MEDIO	1069	460	...	14	UNIVERSIDAD	ESTÁNDAR	CMX	CIUDAD DE MÉXICO	1			
6755	89792	1	U002	MINIMAL ECHO	2021-12-12	1	27M	27 Y MEDIO	1069	460	...	15	CENTRO MAGNO	ESTÁNDAR	GDL	GUADALAJARA	1			
6756	89944	1	U002	MINIMAL ECHO	NaT	1	30M	30 Y MEDIO	1069	460	...	14	PABELLON	ESTÁNDAR	GDL	GUADALAJARA	1			
6757	98031	1	U002	MINIMAL ECHO	NaT	1	18M	18 Y MEDIO	1069	460	...	15	CENTRO MAGNO	ESTÁNDAR	GDL	GUADALAJARA	1			
6758	90729	1	U002	MINIMAL ECHO	NaT	1	18M	18 Y MEDIO	1069	460	...	19	ANGELOPOLIS	ESTÁNDAR	PUE	PUEBLA	1			

TERCERA PARTE

Análisis exploratorio de datos

1. ¿Cuáles son los productos más vendidos por categoría?

Este análisis tiene como objetivo identificar las categorías de productos con mayor volumen de ventas. Se agruparán los datos por categoría de producto y se sumará la cantidad vendida en cada categoría. Los resultados se visualizarán en un gráfico de barras para facilitar la comparación entre categorías.

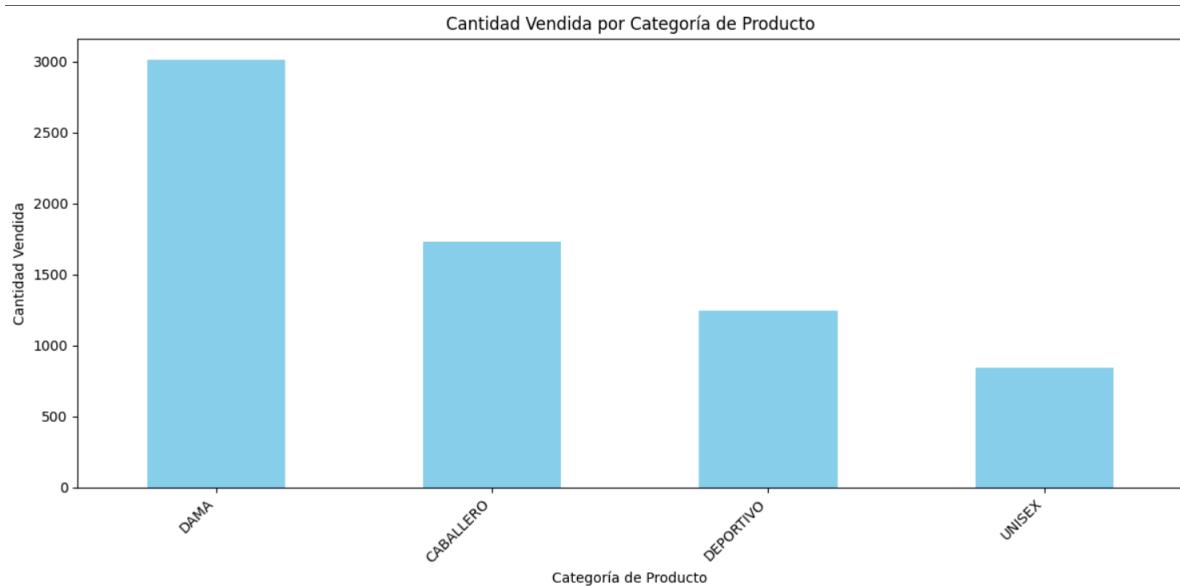
```
try:
    # *** Using the correct column name: 'tipo_calzado' ***
    ventas_por_categoria = shoe.groupby('tipo_calzado')['cantidad'].sum().sort_values(ascending=False)
except KeyError:
    print("Error: The column for product category was not found. Please check the output above to see the correct column names and update the 'groupby' method accordingly.")
    # The 'pass' here means the code continues after the try-except block,
    # but since ventas_por_categoria was not created, the plotting code will be skipped.
    pass
```

Se genera un nuevo DataFrame donde se realiza un groupby a nuestra tabla principal mediante la columna tipo de calzado y se hace una suma de la cantidad de acuerdo con el tipo de calzado y se acomodan de manera descendente

```
# Graficar los resultados (only if ventas_por_categoria was successfully created in the try block)
if 'ventas_por_categoria' in locals():
    plt.figure(figsize=(12, 6))
    ventas_por_categoria.plot(kind='bar', color='skyblue')
    plt.title('Cantidad Vendida por Categoría de Producto')
    plt.xlabel('Categoría de Producto')
    plt.ylabel('Cantidad Vendida')
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

    print("Resultados de la Pregunta 1:")
    print(ventas_por_categoria) # Use print for better formatting of the Series
else:
    print("Plotting skipped because 'ventas_por_categoria' could not be created due to the KeyError.")
```

Se genera una gráfica de barras de nuestro nuevo DataFrame que se realizó en la imagen anterior.

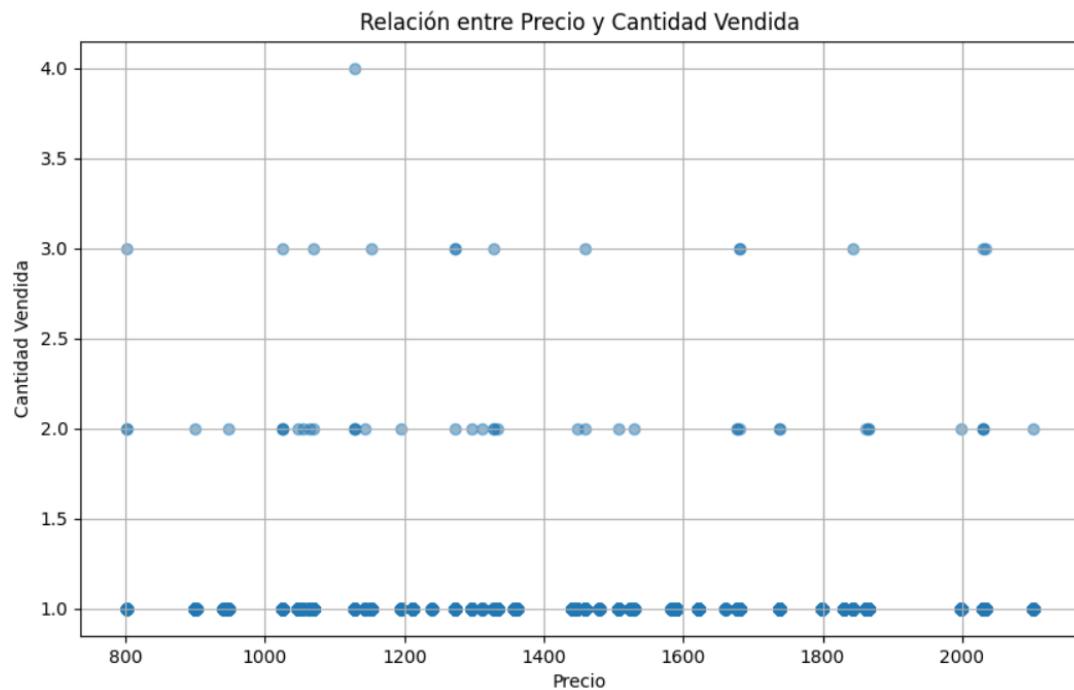


2. ¿Existe una correlación entre el precio y la cantidad vendida?

Este análisis busca determinar si existe una relación lineal entre el precio de un producto y la cantidad de unidades vendidas. Se creará un gráfico de dispersión para visualizar la relación entre el precio y la cantidad vendida. Adicionalmente, se calculará el coeficiente de correlación de Pearson para cuantificar la fuerza y dirección de la relación.

```
#crear un gráfico de dispersión
plt.figure(figsize=(10, 6))
# *** Replaced 'precio' with 'precio_unitario' ***
plt.scatter(shoe['precio_unitario'], shoe['cantidad'], alpha=0.5)
plt.title('Relación entre Precio y Cantidad Vendida')
plt.xlabel('Precio')
plt.ylabel('Cantidad Vendida')
plt.grid(True)
plt.show()
```

Se hace una tabla que nos mostrara la relación entre el precio y la cantidad vendida que nos ayudara a determinar el coeficiente de correlación de una manera mas visual.



```
1 #Calcular el coeficiente de correlación de Pearson
2 correlacion_precio_cantidad = shoe['precio_unitario'].corr(shoe['cantidad'])
3
4 print("\nResultados de la Pregunta 2:")
5 print(f"Coeficiente de correlación entre Precio y Cantidad Vendida: {correlacion_precio_cantidad}")
6
```

```
Resultados de la Pregunta 2:
Coeficiente de correlación entre Precio y Cantidad Vendida: -0.01059976719247022
```

Por ultimo se saca un nuevo DataFrame que por medio de la tabla “precio_unitario” y la columna “cantidad” nos ayudara a determinar el coeficiente aplicando la función de pandas .corr.

3. ¿Hay alguna tendencia estacional en las ventas a lo largo del tiempo?

Este análisis tiene como objetivo identificar patrones de ventas que se repiten en períodos específicos del año (estacionalidad). Se extraerá la información del mes y el año de la columna 'fecha_venta'. Se agruparán los datos por año y mes, y se sumarán las ventas totales. Se visualizarán las ventas mensuales a lo largo del tiempo utilizando un gráfico de línea para observar tendencias y patrones estacionales.

```
1 #extraer el año y el mes de la fecha de venta
2 shoe['año_mes'] = shoe['fecha_venta'].dt.to_period('M')
3
4 ventas_estacionales = shoe.groupby('año_mes').apply(lambda x: (x['precio_unitario'] * x['cantidad']).sum()).reset_index(name='ventas_totales')
5
6 #se convierte año mes
7 ventas_estacionales['año_mes'] = ventas_estacionales['año_mes'].dt.to_timestamp()
8
9 #se ordena por fecha
10 ventas_estacionales = ventas_estacionales.sort_values(by='año_mes')
11
```

Primero se traen las ventas de estación a un nuevo DataFrame.

```
#se visualiza
plt.figure(figsize=(14, 7))
plt.plot(ventas_estacionales['año_mes'], ventas_estacionales['ventas_totales'], marker='o', linestyle='--')
plt.title('Ventas Totales Mensuales a lo largo del Tiempo')
plt.xlabel('Fecha (Año-Mes)')
plt.ylabel('Ventas Totales')
plt.grid(True)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

print("\nResultados de la Pregunta 3:")
print(ventas_estacionales.head())
print("\nObservar el gráfico de línea para identificar patrones o tendencias estacionales.")
```

Se genera un grafico de líneas para poder determinar de manera visual las ventas por estación.

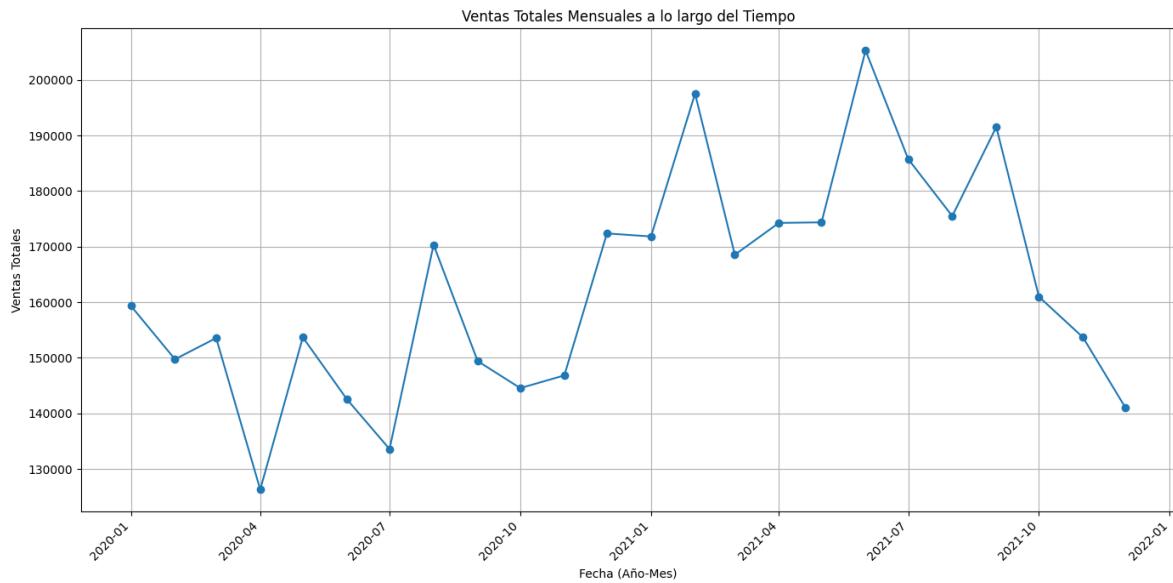


Gráfico que muestra las ventas por estación en el 2020, 2021 y 2022. Donde se observa que donde la venta fue mejor fue en el Varano del 2021 con ventas superiores a las 200 mil.

4. ¿Cuál es el rendimiento de ventas por vendedor?

Este análisis busca evaluar el desempeño de cada vendedor en términos de las ventas que ha generado. Se agruparán los datos por el nombre del vendedor y se calculará la suma del total de ventas (precio * cantidad) para cada uno. Los resultados se presentarán en un gráfico de barras para comparar visualmente el rendimiento de los vendedores.

```
1 local_column_name = 'local'  
2  
3 ventas_por_local = shoe.groupby(local_column_name).apply(lambda x: (x['precio_unitario'] * x['cantidad']).sum()).sort_values(ascending=False)  
4
```

Primero se genera un nuevo DataFrame donde se nos mostrara el rendimiento de cada loca, esto mediante una función lambda que toma como parámetros el precio_unitario y la cantidad y agrupándolos por local.

```
5 #grafica  
6 plt.figure(figsize=(12, 6))  
7  
8 ventas_por_local.plot(kind='bar', color='lightgreen')  
9 plt.title('Rendimiento de Ventas por Local')  
10  
11 plt.xlabel('Local')  
12 plt.ylabel('Ventas Totales')  
13 plt.xticks(rotation=45, ha='right')  
14 plt.tight_layout()  
15 plt.show()  
16  
17 print("\nResultados de la Pregunta 4 (Ventas por Local):")  
18 print(ventas_por_local)  
19 |
```

Se genera un nuevo grafico de barras que nos ayudara a visualizar de mejor manera cuales fueron los locales que tuvieron más ventas.

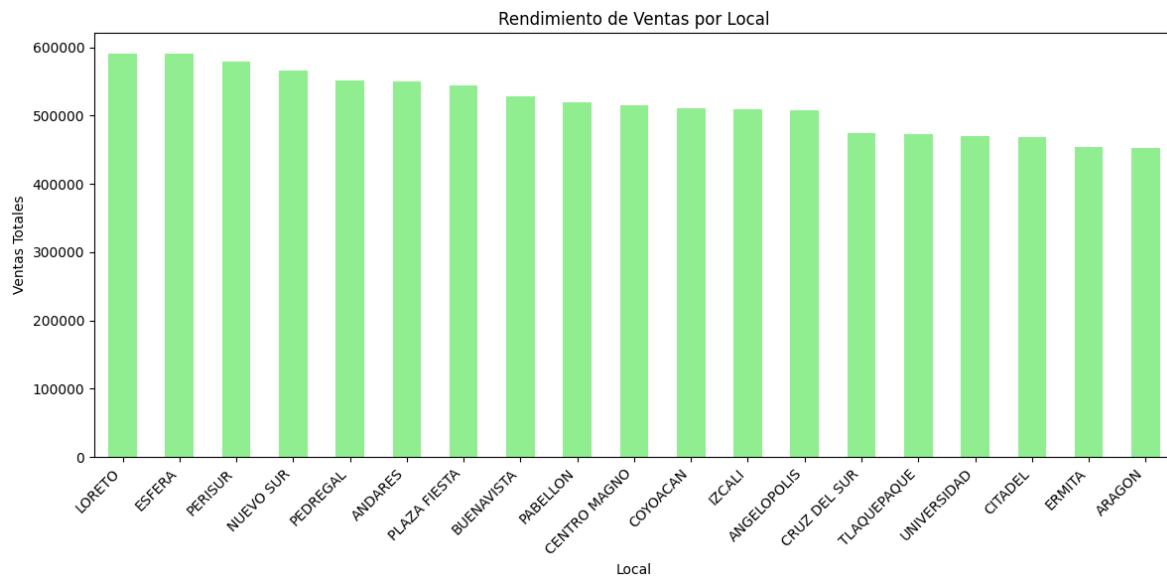
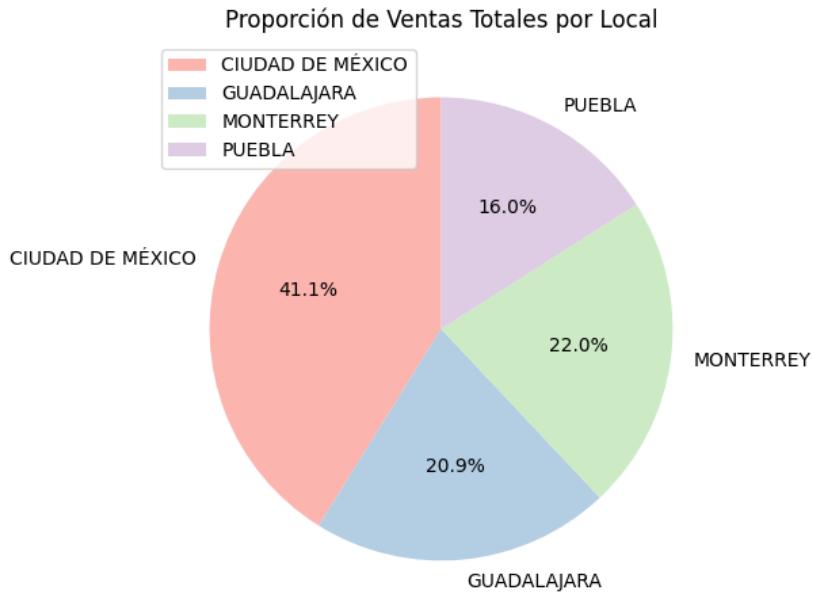


Grafico que nos ayuda a determinar cuales son los locales con mayor numero de ventas, siendo este Loreto con casi 600 mil ventas.

Gráfico de sectores

```
1 if 'ventas_por_local' in locals() and not ventas_por_local.empty:
2     plt.figure(figsize=(8, 8))
3     # Specify the 'total_venta' column as the data for the pie chart
4     ventas_por_local.plot(kind='pie', y='total_venta', autopct='%.1f%%', startangle=90, colors=plt.cm.Pastel1.colors, labels=ventas_por_local['ciudad'])
5     plt.title('Proporción de Ventas Totales por Local')
6     plt.ylabel('')
7     plt.tight_layout()
8     plt.show()
9 else:
0     pass
```

Se genera un gráfico de pastel que nos muestra los locales que mas ventas han tenido pero agrupados por estado.

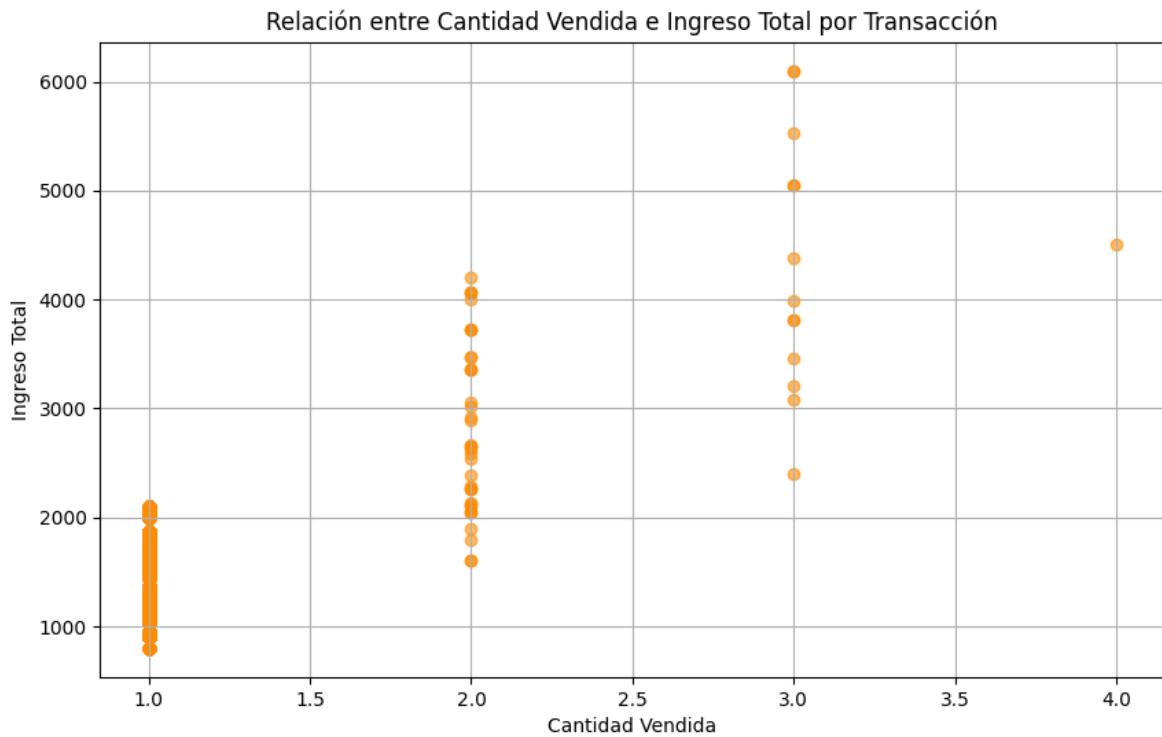


Como se puede apreciar en el grafico destaca el estado de Ciudad de México ya que ahí se concentra la mayor venta por locales.

Gráfica scatter plot.

```
1 data_for_scatter = shoe_filtered if 'shoe_filtered' in locals() and not shoe_filtered.empty else shoe
2
3 if 'cantidad' in data_for_scatter.columns and 'ingreso_total' in data_for_scatter.columns:
4     plt.figure(figsize=(10, 6))
5     plt.scatter(data_for_scatter['cantidad'], data_for_scatter['ingreso_total'], alpha=0.6, color='darkorange')
6     plt.title('Relación entre Cantidad Vendida e Ingreso Total por Transacción')
7     plt.xlabel('Cantidad Vendida')
8     plt.ylabel('Ingreso Total')
9     plt.grid(True)
10    plt.show()
11 else:
12     pass
13
```

Se genera un gráfico que nos muestra la cantidad vendida y el ingreso total por transacción.



Aquí se puede observar el grafico scatter plot que se genero mostando una gran concentración entre la cantidad de 1 venta y mayor dispersión en los precios en la cantidad de 3 ventas, mientras que en 4 observamos una mínima compra alrededor de los 4500 en total.

Hipótesis basadas en las preguntas y análisis:

Basado en los análisis realizados hasta ahora, se pueden plantear las siguientes hipótesis para guiar futuras decisiones:

1. Hipótesis sobre Productos Más Vendidos:

- **Hipótesis:** Existe una alta probabilidad de que las categorías de calzado deportivo y casual (según el análisis de la Pregunta 1) sean las principales fuentes de ingresos y tráfico en la tienda.
- **Implicación:** De ser cierto, la tienda debería priorizar la inversión en inventario y marketing para estas categorías, y considerar la expansión de la oferta en estos segmentos.

2. Hipótesis sobre Correlación Precio-Cantidad:

- **Hipótesis:** La correlación observada (positiva o negativa, según el resultado del análisis de la Pregunta 2) sugiere que la estrategia de precios actual influye significativamente en el volumen de ventas de productos individuales.
- **Implicación:** Si la correlación es negativa, podría indicar que los productos más caros venden menos unidades. Esto podría llevar a explorar estrategias de precios para aumentar las ventas de productos de mayor margen o a enfocar el marketing en resaltar el valor de estos productos. Si la correlación es positiva (menos común, pero posible en nichos), podría indicar que los productos de mayor precio tienen una percepción de mayor calidad o exclusividad, y los clientes están dispuestos a pagar más.

3. Hipótesis sobre Tendencia Estacional:

- **Hipótesis:** El patrón de ventas identificado en la Pregunta 3 (por ejemplo, picos en ciertas estaciones o meses) indica que la demanda de calzado varía significativamente a lo largo del año.

- **Implicación:** La tienda puede optimizar la gestión de inventario, la planificación de campañas de marketing y la asignación de personal basándose en estos patrones estacionales para maximizar las ventas y minimizar el exceso de inventario fuera de temporada alta.
4. **Hipótesis sobre Rendimiento de Ventas por Local (originalmente planteada como por vendedor, pero ajustada al análisis por local):**
- **Hipótesis:** Las diferencias en el rendimiento de ventas entre locales (identificadas en la Pregunta 4) sugieren que hay factores específicos de cada ubicación (tráfico peatonal, demografía de la zona, gestión local, etc.) que impactan directamente en las ventas.
 - **Implicación:** Analizar los factores diferenciadores de los locales con mejor y peor rendimiento podría revelar mejores prácticas o áreas de mejora. Esto podría guiar decisiones sobre la asignación de recursos, capacitación del personal, estrategias de marketing local o incluso la reevaluación de la viabilidad de ciertos locales.

Consideraciones Adicionales para Futuras Hipótesis:

- **Hipótesis sobre la efectividad de descuentos/promociones:** Si hubiera datos sobre promociones, se podría hipotetizar que las ventas aumentan significativamente durante los períodos promocionales, y que ciertos tipos de promociones son más efectivas que otras.
- **Hipótesis sobre la influencia de nuevos productos:** Si se lanzaron nuevos productos, se podría hipotetizar que las ventas totales aumentaron o que las ventas de categorías específicas cambiaron después del lanzamiento.

- **Hipótesis sobre el impacto del marketing digital:** Si se rastrean campañas de marketing digital, se podría hipotetizar que hay una correlación positiva entre la inversión en marketing y el tráfico/ventas online o en tiendas físicas.

Anexo

Link de libreta colab:

<https://colab.research.google.com/drive/1aiXpSJkypAimLVA8bXuYQsNghok0x26j?usp=sharing>

Reseñas del curso Diseño y Modelado Profesional de Datos para Analítica

2059118 - Coronado Arreguin Ciro:

The screenshot shows the Udemy course page for 'Diseño y Modelado Profesional de Datos para Analítica'. On the left, there's a sidebar with course navigation: Overview, Q&A, Notes, and Announcements. The main content area displays a review by 'Ciro Arreguin' with a 5-star rating. The review text is as follows:

Su reseña será pública dentro de 24 horas.

Ciro Arreguin

Este curso es una excelente opción para quienes se están iniciando en el modelado y análisis de datos. Los temas están bien estructurados y se explican de forma clara y detallada en cada clase. Aprendí el significado de muchos conceptos que antes me parecían complejos, y ahora los entiendo con mayor facilidad. Nunca es tarde para aprender algo nuevo, y más a un este tipo de temas tan interesantes que al día de hoy están siendo tan relevantes en la actualidad.

Mostrar menos ^

On the right, there's a preview of the course content structure, showing sections like 'Introducción', 'Fundamentos de análisis de datos', and 'Naturaleza y clasificación de los datos'.

2179817 - Guglielmi Marín Saúl:

The screenshot shows the Udemy course page for 'Modela, documenta e interpreta tus habilidades de analítica'. On the left, there's a sidebar with course navigation: Descripción general, Preguntas y respuestas, and a summary section with 0 calificaciones, 12 estudiantes, and 5,5 horas. The main content area displays a review by 'Saul Guglielmi Marín' with a 5-star rating. The review text is as follows:

Estoy iniciando mi camino profesional en el área de datos, y debo decir que este curso ha sido de gran utilidad para aplicar los conocimientos en proyectos y trabajos reales. Los temas que se abordan me han ayudado a comprender conceptos que antes me resultaban complejos, gracias a la forma clara y didáctica en la que se explican.

Destaco especialmente los ejemplos prácticos, que son sencillos pero muy efectivos para afianzar el aprendizaje. El instructor logra transmitir los contenidos de manera accesible y bien estructurada, lo que facilita mucho el proceso de incorporación de conocimientos técnicos.

Recomiendo este curso tanto a quienes están comenzando en el mundo del modelado de datos como a quienes ya tienen experiencia y desean profundizar o reforzar sus habilidades.

Ver menos ^

On the right, there's a preview of the course content structure, showing sections like 'Presentación del curso', 'Importancia de la modelación en la analítica de datos', and 'Entendiendo los alcances de la analítica de datos'.

2012344 - Mata Coronado Orlando Fabian:

The screenshot shows a browser window on a Windows operating system. The main content is a Udemy course review for 'Diseño y Modelado Profesional de Datos' by Orlando Fabián Mata Coronado. The review is titled 'Thanks for helping our community!' and states: 'Your review will be public within 24 hours.' The review itself is as follows:

Orlando Fabián Mata Coronado
★★★★★
Como licenciado en Tecnologías de la Información, este curso ha sido una herramienta clave para fortalecer mis conocimientos en analítica de datos. Me pareció interesante el cómo se abordó la importancia de la modelación, mostrando su impacto el análisis de los datos en el proceso ETL.
Se destaca también la claridad al explicar el alcance de la analítica y cómo se deberían de aplicar según los objetivos del negocio. La relación entre modelación y el proceso ETL se presenta de forma práctica y útil para proyectos reales.
También gracias a los recursos de interacción proporcionados durante el curso ayudan de gran manera a la comprensión de los temas de una manera práctica, lo cual se aprecia mucho ya que en nuestra área la práctica es de gran importante. Muy recomendable este curso.

Below the review, there is a 'Show less' link. To the right of the review, the course navigation sidebar is visible, showing sections like 'Course content', 'Section 4: Modelación de bases de datos relacionales', 'Section 5: Normalización de bases de datos', and 'Section 6: Diagramación y documentación'. The sidebar also includes an 'AI Assistant' feature and a progress bar.



Orlando Fabián Mata Coronado

★★★★★

Como licenciado en Tecnologías de la Información, este curso ha sido una herramienta clave para fortalecer mis conocimientos en analítica de datos. Me pareció interesante el cómo se abordó la importancia de la modelación, mostrando su impacto el análisis de los datos en el proceso ETL.

Se destaca también la claridad al explicar el alcance de la analítica y cómo se deberían de aplicar según los objetivos del negocio. La relación entre modelación y el proceso ETL se presenta de forma práctica y útil para proyectos reales.

También gracias a los recursos de interacción proporcionados durante el curso ayudan de gran manera a la comprensión de los temas de una manera práctica, lo cual se aprecia mucho ya que en nuestra área la práctica es de gran importante. Muy recomendable este curso.

Show less ^

2014292 - Rodríguez Astrain Alan Jhosep:

Atrás

¿Por qué diste esta calificación?

Excelente, por encima de mis expectativas

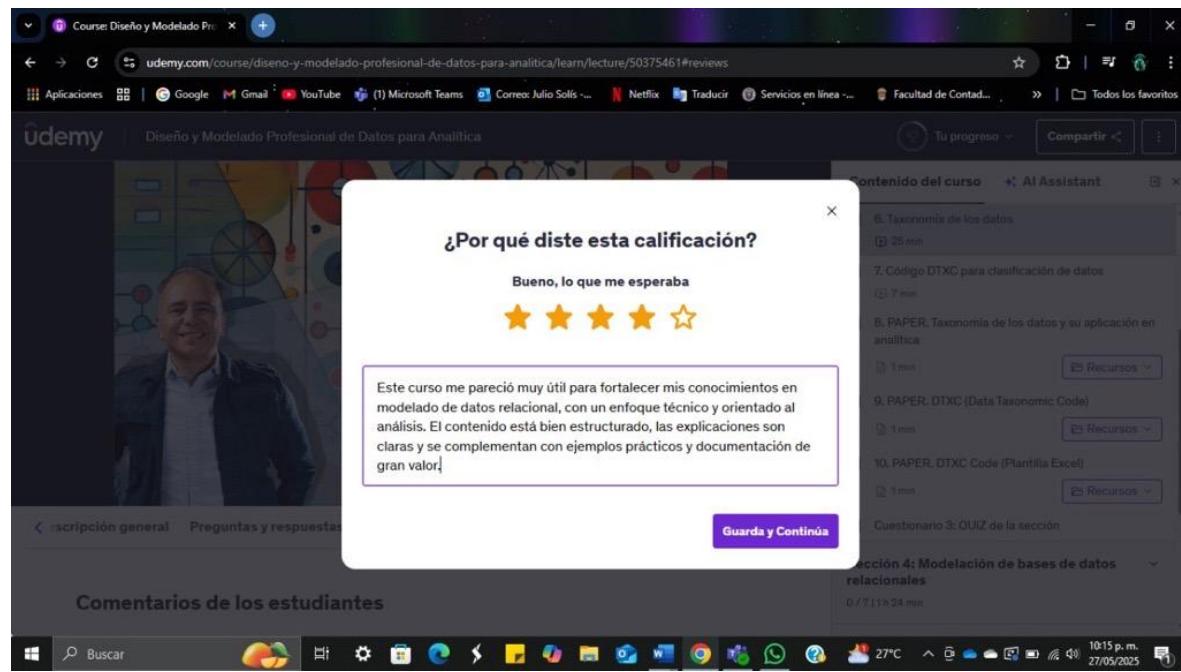
★★★★★

Un excelente curso, bien explicado y los ejercicios prácticos bastante útiles para entender bien cuales son los fundamentos del modelado de datos, sienta muy bien las bases para poder seguir practicando

Guarda y Continúa



2118842 - Solís Cavazos Julio Alexander:



Course: Diseño y Modelado Profesional de Datos para Analítica

udemy | Diseño y Modelado Profesional de Datos para Analítica

¿Por qué diste esta calificación?

Bueno, lo que me esperaba

★★★★★

Este curso me pareció muy útil para fortalecer mis conocimientos en modelado de datos relacional, con un enfoque técnico y orientado al análisis. El contenido está bien estructurado, las explicaciones son claras y se complementan con ejemplos prácticos y documentación de gran valor.

Guarda y Continúa

Comentarios de los estudiantes

Contenido del curso

- 6. Taxonomía de los datos
- 7. Código DTXC para clasificación de datos
- 8. PAPER. Taxonomía de los datos y su aplicación en analíticas
- 9. PAPER. DTXC (Data Taxonomic Code)
- 10. PAPER. DTXC Code (Plantilla Excel)

Cuestionario 3: QUIZ de la sección

Lección 4: Modelación de bases de datos relacionales

0 / 7 | 1h 24 min

10:15 p.m. 27/05/2025

pia

```
1 import pandas as pd
2 import matplotlib.pyplot as plt

1 shoe_url = ('https://raw.githubusercontent.com/AprendaPracticando/AnaliticaPythonR1/refs/heads/main/data/ShoelandDB.csv')
2 shoe = pd.read_csv(shoe_url)

1 shoe.dtypes
```

	0
ticket	int64
item	int64
id_producto	object
producto	object
fecha_venta	object
cantidad	int64
id_tamaño	object
tamaño	object
precio_unitario	int64
costo_unitario	int64
tipo_calzado	object
id_local	int64
local	object
tipo_local	object
id_ciudad	object
ciudad	object
total_venta	int64
total_costo	int64
año	int64
mes	object

dtype: object

▼ Se convierte fecha venta a date time

```
1 shoe['fecha_venta'] = pd.to_datetime(shoe['fecha_venta'], errors='coerce')
2 print("convertido")
3 shoe.dtypes
```

convertido

	0
ticket	int64
item	int64
id_producto	object
producto	object
fecha_venta	datetime64[ns]
cantidad	int64
id_tamaño	object
tamaño	object
precio_unitario	int64
costo_unitario	int64
tipo_calzado	object
id_local	int64
local	object
tipo_local	object
id_ciudad	object
ciudad	object
total_venta	int64
total_costo	int64
año	int64
mes	object

dtype: object

✓ Generación de nueva nueva columna

```

1 def clasificar_cliente(cantidad):
2     if cantidad == 1:
3         return 'Comprador ocasional'
4     elif cantidad in [2, 3]:
5         return 'Comprador frecuente'
6     else:
7         return 'Comprador mayorista'
8
9
10 shoe['tipo_cliente'] = shoe['cantidad'].apply(clasificar_cliente)

```

✓ Visualizacion de los datos

```
1 shoe.dtypes
```

0

ticket	int64
item	int64
id_producto	object
producto	object
fecha_venta	datetime64[ns]
cantidad	int64
id_tamaño	object
tamaño	object
precio_unitario	int64
costo_unitario	int64
tipo_calzado	object
id_local	int64
local	object
tipo_local	object
id_ciudad	object
ciudad	object
total_venta	int64
total_costo	int64
año	int64
mes	object
tipo_cliente	object

dtype: object

1 shoe.head(3)

0

	ticket	item	id_producto	producto	fecha_venta	cantidad	id_tamaño	tamaño	precio_unitario	costo_unitario	...	id_local
0	10633	1	C001	COMFORT VERONA	2020-07-01	1	22M	22 Y MEDIO	1311	393	...	11 A1
1	11031	1	C001	COMFORT VERONA	2020-11-01	1	22M	22 Y MEDIO	1311	393	...	17 CRUZ D
2	11129	1	C001	COMFORT VERONA	2020-12-01	1	22M	22 Y MEDIO	1311	393	...	16 TLAQUE

3 rows × 21 columns

1 shoe.tail(3)

0

	ticket	item	id_producto	producto	fecha_venta	cantidad	id_tamaño	tamaño	precio_unitario	costo_unitario	...	id_local
6756	89944	1	U002	MINIMAL ECHO	NaT	1	30M	30 Y MEDIO	1069	460	...	14 F
6757	98031	1	U002	MINIMAL ECHO	NaT	1	18M	18 Y MEDIO	1069	460	...	15
6758	90729	1	U002	MINIMAL ECHO	NaT	1	18M	18 Y MEDIO	1069	460	...	19 ANG

3 rows × 21 columns

1 shoe.describe()

	ticket	item	fecha_venta	cantidad	precio_unitario	costo_unitario	id_local	total_venta	total_costo
count	6759.000000	6759.000000		2669	6759.000000	6759.000000	6759.000000	6759.000000	6759.000000
mean	52276.246634	1.032697	2020-12-31 20:37:08.250280960	1.010061	1444.123391	410.119692	10.026779	1458.219411	414.206391
min	10004.000000	1.000000	2020-01-01 00:00:00	1.000000	801.000000	380.000000	1.000000	801.000000	380.000000
25%	31354.500000	1.000000	2020-07-09 00:00:00	1.000000	1153.000000	388.000000	5.000000	1153.000000	388.000000
50%	53481.000000	1.000000	2021-01-10 00:00:00	1.000000	1448.000000	396.000000	10.000000	1448.000000	396.000000
75%	72572.000000	1.000000	2021-07-02 00:00:00	1.000000	1681.000000	424.000000	15.000000	1738.000000	424.000000
max	98979.000000	4.000000	2021-12-12 00:00:00	4.000000	2102.000000	460.000000	19.000000	6099.000000	1548.000000

1 shoe.shape

2 (6759, 21)

1 shoe.info()

```
3 <class 'pandas.core.frame.DataFrame'>
RangeIndex: 6759 entries, 0 to 6758
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   ticket      6759 non-null   int64  
 1   item        6759 non-null   int64  
 2   id_producto 6759 non-null   object  
 3   producto    6759 non-null   object  
 4   fecha_venta 2669 non-null   datetime64[ns]
 5   cantidad    6759 non-null   int64  
 6   id_tamaño   6759 non-null   object  
 7   tamaño      6759 non-null   object  
 8   precio_unitario 6759 non-null   int64  
 9   costo_unitario 6759 non-null   int64  
 10  tipo_calzado 6759 non-null   object  
 11  id_local    6759 non-null   int64  
 12  local       6759 non-null   object  
 13  tipo_local  6759 non-null   object  
 14  id_ciudad   6759 non-null   object  
 15  ciudad      6759 non-null   object  
 16  total_venta 6759 non-null   int64  
 17  total_costo 6759 non-null   int64  
 18  año         6759 non-null   int64  
 19  mes         6759 non-null   object  
 20  tipo_cliente 6759 non-null   object  
dtypes: datetime64[ns](1), int64(9), object(11)
memory usage: 1.1+ MB
```

1 shoe.isnull().sum()

```
ticket          0
item            0
id_producto    0
producto        0
fecha_venta    4090
cantidad        0
id_tamaño      0
tamaño          0
precio_unitario 0
costo_unitario 0
tipo_calzado   0
id_local         0
local            0
tipo_local       0
id_ciudad        0
ciudad           0
total_venta     0
total_costo      0
año              0
mes              0
tipo_cliente    0
```

dtype: int64

```
1 len(shoe)
```

```
2 6759
```

```
1 shoe.duplicated().sum()
```

```
2 np.int64(0)
```

```
1 shoe
```

	ticket	item	id_producto	producto	fecha_venta	cantidad	id_tamaño	tamaño	precio_unitario	costo_unitario	...	local
0	10633	1	C001	COMFORT VERONA	2020-07-01	1	22M	22 Y MEDIO	1311	393	...	ARAGON
1	11031	1	C001	COMFORT VERONA	2020-11-01	1	22M	22 Y MEDIO	1311	393	...	CRUZ DEL SUF
2	11129	1	C001	COMFORT VERONA	2020-12-01	1	22M	22 Y MEDIO	1311	393	...	TLAQUEPAQUE
3	11794	1	C001	COMFORT VERONA	Nat	1	24N	24 NORMAL	1311	393	...	TLAQUEPAQUE
4	15046	1	C001	COMFORT VERONA	Nat	1	18M	18 Y MEDIO	1311	393	...	TLAQUEPAQUE
...
6754	89454	1	U002	MINIMAL ECHO	2021-08-12	1	28M	28 Y MEDIO	1069	460	...	UNIVERSIDA
6755	89792	1	U002	MINIMAL ECHO	2021-12-12	1	27M	27 Y MEDIO	1069	460	...	CENTRO MAGN
6756	89944	1	U002	MINIMAL ECHO	Nat	1	30M	30 Y MEDIO	1069	460	...	PABELLON
6757	98031	1	U002	MINIMAL ECHO	Nat	1	18M	18 Y MEDIO	1069	460	...	CENTRO MAGN
6758	90729	1	U002	MINIMAL ECHO	Nat	1	18M	18 Y MEDIO	1069	460	...	ANGELOPOLIS

6759 rows × 22 columns

Preguntas para el análisis

Con el objetivo de comprender mejor el comportamiento de las ventas y optimizar la toma de decisiones en la tienda de zapatos, se plantean una serie de preguntas clave. Estas interrogantes permitirán explorar distintas dimensiones del negocio, como las preferencias de los clientes, el desempeño del equipo de ventas y las posibles tendencias del mercado. A partir de estas preguntas, se desarrollarán análisis apoyados en visualizaciones y estadísticas que faciliten obtener conclusiones útiles y accionables.

▼ Pregunta 1: ¿Cuáles son los productos más vendidos por categoría?

Descripción del análisis para la Pregunta 1: Este análisis tiene como objetivo identificar las categorías de productos con mayor volumen de ventas. Se agruparán los datos por categoría de producto y se sumará la cantidad vendida en cada categoría. Los resultados se visualizarán en un gráfico de barras para facilitar la comparación entre categorías.

```

1 #Agrupar por categoría de producto y sumar la cantidad vendida
2 print("Columns in the shoe DataFrame:", shoe.columns)

Columns in the shoe DataFrame: Index(['ticket', 'item', 'id_producto', 'producto', 'fecha_venta', 'cantidad',
   'id_tamaño', 'tamaño', 'precio_unitario', 'costo_unitario',
   'tipo_calzado', 'id_local', 'local', 'tipo_local', 'id_ciudad',
   'ciudad', 'total_venta', 'total_costo', 'año', 'mes', 'tipo_cliente'],
  dtype='object')

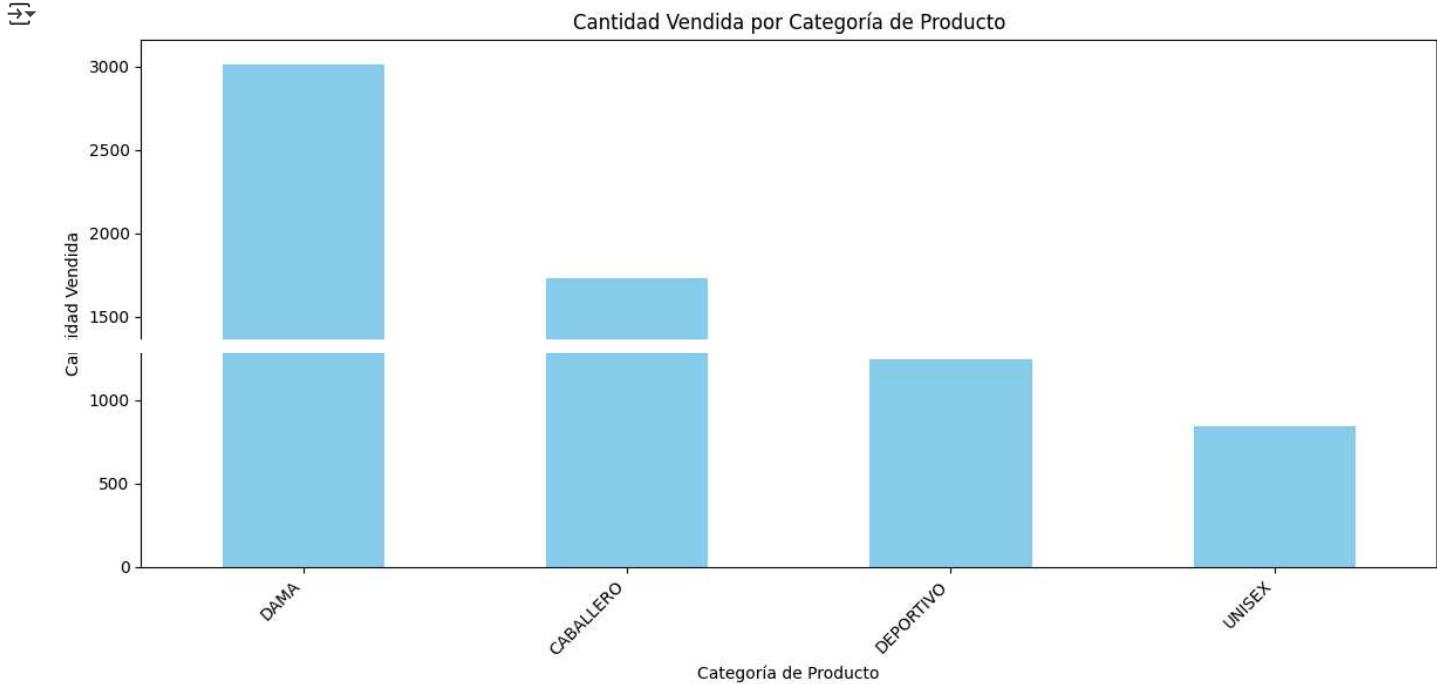
3 try:
4     # *** Using the correct column name: 'tipo_calzado' ***
5     ventas_por_categoria = shoe.groupby('tipo_calzado')['cantidad'].sum().sort_values(ascending=False)
6 except KeyError:
7     print("Error: The column for product category was not found. Please check the output above to see the correct column names and update")
8     # The 'pass' here means the code continues after the try-except block,
9     # but since ventas_por_categoria was not created, the plotting code will be skipped.
10    pass
11
12 # Graficar los resultados (only if ventas_por_categoria was successfully created in the try block)
13 if 'ventas_por_categoria' in locals():
14     plt.figure(figsize=(12, 6))
15     ventas_por_categoria.plot(kind='bar', color='skyblue')
16     plt.title('Cantidad Vendida por Categoría de Producto')

```

```

15     plt.xlabel('Categoria de Producto')
16     plt.ylabel('Cantidad Vendida')
17     plt.xticks(rotation=45, ha='right')
18     plt.tight_layout()
19     plt.show()
20
21     print("Resultados de la Pregunta 1:")
22     print(ventas_por_categoria) # Use print for better formatting of the Series
23 else:
24     print("Plotting skipped because 'ventas_por_categoria' could not be created due to the KeyError.")

```



```

Resultados de la Pregunta 1:
tipo_calzado
DAMA      3011
CABALLERO 1732
DEPORTIVO 1244
UNISEX    840
Name: cantidad, dtype: int64

```

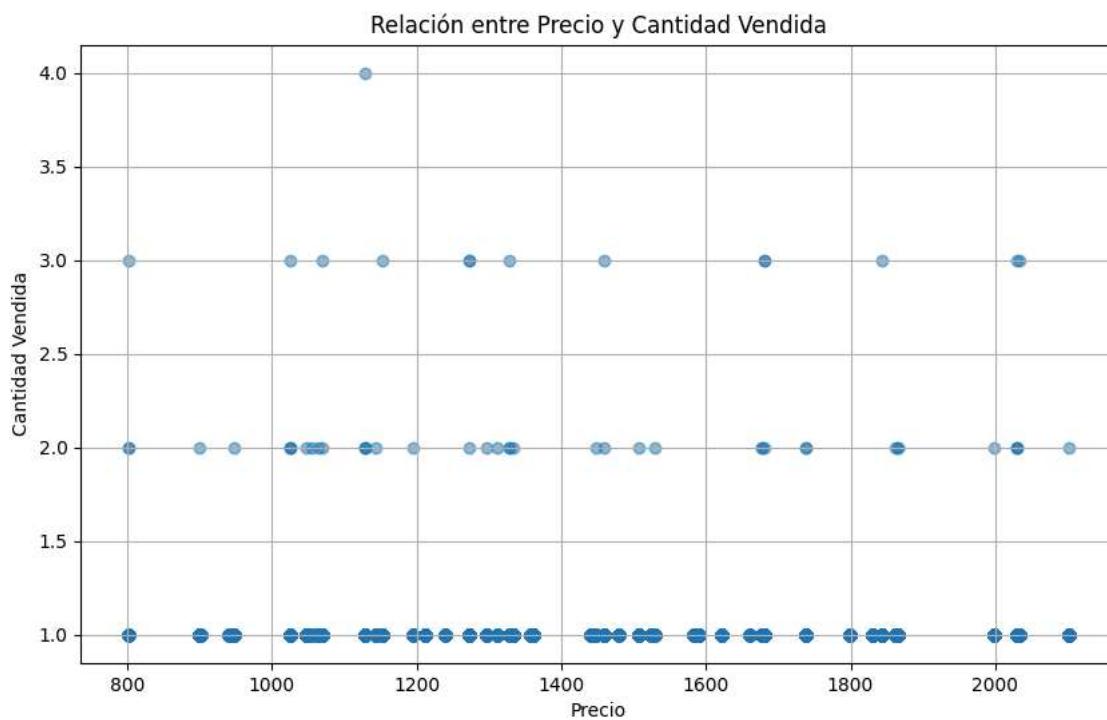
✓ pregunta 2: ¿Existe una correlación entre el precio y la cantidad vendida?

Descripción del análisis para la Pregunta 2: Este análisis busca determinar si existe una relación lineal entre el precio de un producto y la cantidad de unidades vendidas. Se creará un gráfico de dispersión para visualizar la relación entre el precio y la cantidad vendida. Adicionalmente, se calculará el coeficiente de correlación de Pearson para cuantificar la fuerza y dirección de la relación.

```

1 #crear un gráfico de dispersión
2 plt.figure(figsize=(10, 6))
3 # *** Replaced 'precio' with 'precio_unitario' ***
4 plt.scatter(shoe['precio_unitario'], shoe['cantidad'], alpha=0.5)
5 plt.title('Relación entre Precio y Cantidad Vendida')
6 plt.xlabel('Precio')
7 plt.ylabel('Cantidad Vendida')
8 plt.grid(True)
9 plt.show()
10
11

```



```

1 #Calcular el coeficiente de correlación de Pearson
2 correlacion_precio_cantidad = shoe['precio_unitario'].corr(shoe['cantidad'])
3
4 print("\nResultados de la Pregunta 2:")
5 print(f"Coeficiente de correlación entre Precio y Cantidad Vendida: {correlacion_precio_cantidad}")
6

```

Resultados de la Pregunta 2:
Coeficiente de correlación entre Precio y Cantidad Vendida: -0.01059976719247022

✓ Pregunta 3 ¿Hay alguna tendencia estacional en las ventas a lo largo del tiempo?

Descripción del análisis para la Pregunta 3: Este análisis tiene como objetivo identificar patrones de ventas que se repiten en períodos específicos del año (estacionalidad). Se extraerá la información del mes y el año de la columna 'fecha_venta'. Se agruparán los datos por año y mes, y se sumarán las ventas totales. Se visualizarán las ventas mensuales a lo largo del tiempo utilizando un gráfico de línea para observar tendencias y patrones estacionales.

```

1 #extraer el año y el mes de la fecha de venta
2 shoe['año_mes'] = shoe['fecha_venta'].dt.to_period('M')
3
4 ventas_estacionales = shoe.groupby('año_mes').apply(lambda x: (x['precio_unitario'] * x['cantidad']).sum()).reset_index(name='ventas_totales')
5
6 #se convierte año mes
7 ventas_estacionales['año_mes'] = ventas_estacionales['año_mes'].dt.to_timestamp()
8
9 #se ordena por fecha
10 ventas_estacionales = ventas_estacionales.sort_values(by='año_mes')
11

```

<ipython-input-20-17600e85a995>:4: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated.

```

ventas_estacionales = shoe.groupby('año_mes').apply(lambda x: (x['precio_unitario'] * x['cantidad']).sum()).reset_index(name='ventas_totales')

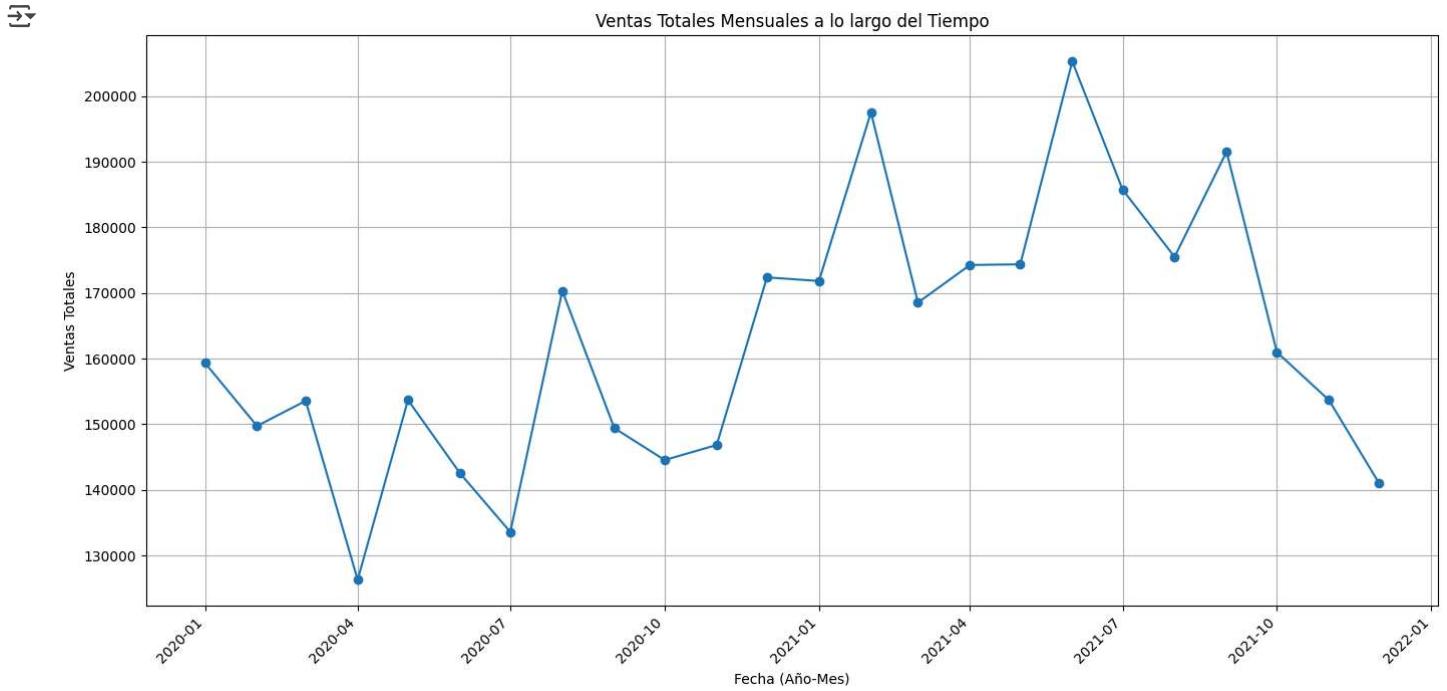
```

1 #se visualiza
2 plt.figure(figsize=(14, 7))
3 plt.plot(ventas_estacionales['año_mes'], ventas_estacionales['ventas_totales'], marker='o', linestyle='--')
4 plt.title('Ventas Totales Mensuales a lo largo del Tiempo')
5 plt.xlabel('Fecha (Año-Mes)')
6 plt.ylabel('Ventas Totales')
7 plt.grid(True)
8 plt.xticks(rotation=45, ha='right')

```

9 plt.tight_layout()
10 plt.show()
11
12 print("\nResultados de la Pregunta 3:")
13 print(ventas_estacionales.head())
14 print("\nObservar el gráfico de línea para identificar patrones o tendencias estacionales.")
15
16

```



Resultados de la Pregunta 3:

año_mes	ventas_totales
2020-01-01	159343
2020-02-01	149742
2020-03-01	153562
2020-04-01	126307
2020-05-01	153689

Observar el gráfico de línea para identificar patrones o tendencias estacionales.

▼ Pregunta 4: ¿Cuál es el rendimiento de ventas por vendedor?

Descripción del análisis para la Pregunta 4: Este análisis busca evaluar el desempeño de cada vendedor en términos de las ventas que ha generado. Se agruparán los datos por el nombre del vendedor y se calculará la suma del total de ventas (precio * cantidad) para cada uno. Los resultados se presentarán en un gráfico de barras para comparar visualmente el rendimiento de los vendedores.

```

1 local_column_name = 'local'
2
3 ventas_por_local = shoe.groupby(local_column_name).apply(lambda x: (x['precio_unitario'] * x['cantidad']).sum()).sort_values(ascending=False)
4
5 #grafica
6 plt.figure(figsize=(12, 6))
7
8 ventas_por_local.plot(kind='bar', color='lightgreen')
9 plt.title('Rendimiento de Ventas por Local')
10
11 plt.xlabel('Local')

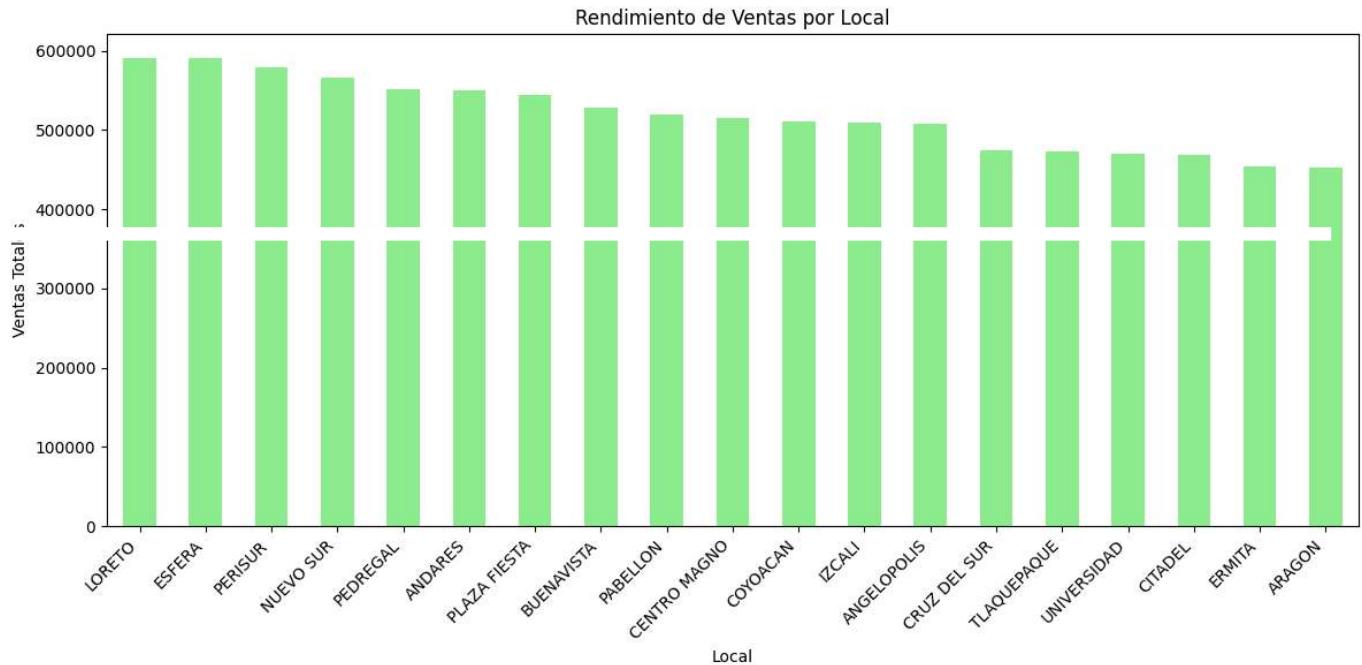
```

```

12 plt.ylabel('Ventas Totales')
13 plt.xticks(rotation=45, ha='right')
14 plt.tight_layout()
15 plt.show()
16
17 print("\nResultados de la Pregunta 4 (Ventas por Local):")
18 print(ventas_por_local)
19

```

→ <ipython-input-22-64681b7f11e0>:3: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is depr ventas_por_local = shoe.groupby(local_column_name).apply(lambda x: (x['precio_unitario'] * x['cantidad']).sum()).sort_values(ascend



Resultados de la Pregunta 4 (Ventas por Local):

```

local
LORETO      591393
ESFERA      590381
PERISUR     579403
NUEVO SUR   566554
PEDREGAL    550911
ANDARES     550209
PLAZA FIESTA 543727
BUENAVISTA   527952
PABELLON    518999
CENTRO MAGNO 515695
COYOACAN    510356
IZCALI       509376
ANGELOPOLIS 507883
CRUZ DEL SUR 474922
TLAQUEPAQUE 472801
UNIVERSIDAD   469643
CITADEL      468875
ERMITA       454357
ARAGON        452668
dtype: int64

```

Hipótesis basadas en las preguntas y análisis:

Basado en los análisis realizados hasta ahora, se pueden plantear las siguientes hipótesis para guiar futuras decisiones:

1. Hipótesis sobre Productos Más Vendidos:

- **Hipótesis:** Existe una alta probabilidad de que las categorías de calzado deportivo y casual (según el análisis de la Pregunta 1) sean las principales fuentes de ingresos y tráfico en la tienda.
- **Implicación:** De ser cierto, la tienda debería priorizar la inversión en inventario y marketing para estas categorías, y considerar la expansión de la oferta en estos segmentos.

2. Hipótesis sobre Correlación Precio-Cantidad:

- **Hipótesis:** La correlación observada (positiva o negativa, según el resultado del análisis de la Pregunta 2) sugiere que la estrategia de precios actual influye significativamente en el volumen de ventas de productos individuales.
- **Implicación:** Si la correlación es negativa, podría indicar que los productos más caros venden menos unidades. Esto podría llevar a explorar estrategias de precios para aumentar las ventas de productos de mayor margen o a enfocar el marketing en resaltar el valor de estos productos. Si la correlación es positiva (menos común, pero posible en nichos), podría indicar que los productos de mayor precio tienen una percepción de mayor calidad o exclusividad, y los clientes están dispuestos a pagar más.

3. Hipótesis sobre Tendencia Estacional:

- **Hipótesis:** El patrón de ventas identificado en la Pregunta 3 (por ejemplo, picos en ciertas estaciones o meses) indica que la demanda de calzado varía significativamente a lo largo del año.
- **Implicación:** La tienda puede optimizar la gestión de inventario, la planificación de campañas de marketing y la asignación de personal basándose en estos patrones estacionales para maximizar las ventas y minimizar el exceso de inventario fuera de temporada alta.

4. Hipótesis sobre Rendimiento de Ventas por Local (originalmente planteada como por vendedor, pero ajustada al análisis por local):

- **Hipótesis:** Las diferencias en el rendimiento de ventas entre locales (identificadas en la Pregunta 4) sugieren que hay factores específicos de cada ubicación (tráfico peatonal, demografía de la zona, gestión local, etc.) que impactan directamente en las ventas.
- **Implicación:** Analizar los factores diferenciadores de los locales con mejor y peor rendimiento podría revelar mejores prácticas o áreas de mejora. Esto podría guiar decisiones sobre la asignación de recursos, capacitación del personal, estrategias de marketing local o incluso la reevaluación de la viabilidad de ciertos locales.

Consideraciones Adicionales para Futuras Hipótesis:

- **Hipótesis sobre la efectividad de descuentos/promociones:** Si hubiera datos sobre promociones, se podría hipotetizar que las ventas aumentan significativamente durante los períodos promocionales, y que ciertos tipos de promociones son más efectivas que otras.
- **Hipótesis sobre la influencia de nuevos productos:** Si se lanzaron nuevos productos, se podría hipotetizar que las ventas totales aumentaron o que las ventas de categorías específicas cambiaron después del lanzamiento.
- **Hipótesis sobre el impacto del marketing digital:** Si se rastrean campañas de marketing digital, se podría hipotetizar que hay una correlación positiva entre la inversión en marketing y el tráfico/ventas online o en tiendas físicas.

Objetivo analítico:

El objetivo de este análisis de datos de ventas de Shoeland es identificar patrones y tendencias clave en el comportamiento de compra de los clientes y el rendimiento de la tienda. A través de la exploración de las categorías de productos más vendidos, la relación entre precio y cantidad, la estacionalidad de las ventas y el rendimiento por local, se busca obtener información accionable que permita optimizar el inventario, refinar las estrategias de precios, mejorar la planificación de campañas y potenciar el rendimiento general de la tienda."

▼ Ingeniería de datos

Filtrado: Considerar solo las ventas mayores a 0 cantidad y precio unitario mayor a 0** Esto ayuda a eliminar posibles entradas de datos erróneos donde la cantidad o el precio es cero o negativo.

```
1 shoe_filtered = shoe[(shoe['cantidad'] > 0) & (shoe['precio_unitario'] > 0)].copy()
2
3 print("\nShape of the original data:", shoe.shape)
4 print("Shape after filtering quantity and price > 0:", shoe_filtered.shape)
5
6
```

Shape of the original data: (6759, 22)
Shape after filtering quantity and price > 0: (6759, 22)

Conversión: Asegura que las columnas numéricas usadas para cálculos sean de tipo numérico. Aunque ya se verificó con dtypes, es buena práctica asegurar, especialmente después de filtros.

```
1 numeric_cols = ['precio_unitario', 'cantidad']
2 for col in numeric_cols:
```

```
3     shoe_filtered[col] = pd.to_numeric(shoe_filtered[col], errors='coerce')
```

✓ Elimina si hay datos nulos

```
1
2 shoe_filtered.dropna(subset=numerical_cols, inplace=True)
3 print("Shape despues de eliminar datos nulos: ", shoe_filtered.shape)
```

→ Shape despues de eliminar datos nulos: (6759, 22)

✓ calcula el ingreso total por fila

```
1 shoe_filtered['ingreso_total'] = shoe_filtered['precio_unitario'] * shoe_filtered['cantidad']
2
3 print("\nDataFrame with new 'ingreso_total' column:")
4 print(shoe_filtered[['precio_unitario', 'cantidad', 'ingreso_total']].head())
5 print("\nData types after creating 'ingreso_total':")
6 print(shoe_filtered.dtypes)
```

→ DataFrame with new 'ingreso_total' column:

	precio_unitario	cantidad	ingreso_total
0	1311	1	1311
1	1311	1	1311
2	1311	1	1311
3	1311	1	1311
4	1311	1	1311

Data types after creating 'ingreso_total':

ticket	int64
item	int64
id_producto	object
producto	object
fecha_venta	datetime64[ns]
cantidad	int64
id_tamaño	object
tamaño	object
precio_unitario	int64
costo_unitario	int64
tipo_calzado	object
id_local	int64
local	object
tipo_local	object
id_ciudad	object
ciudad	object
total_venta	int64
total_costo	int64
año	int64
mes	object
tipo_cliente	object
año_mes	period[M]
ingreso_total	int64
dtype:	object

Ahora podemos usar `shoe_filtered` y la nueva columna `ingreso_total` en los análisis posteriores si es necesario para una mayor precisión o para responder preguntas que involucren ingresos en lugar de solo cantidad vendida.

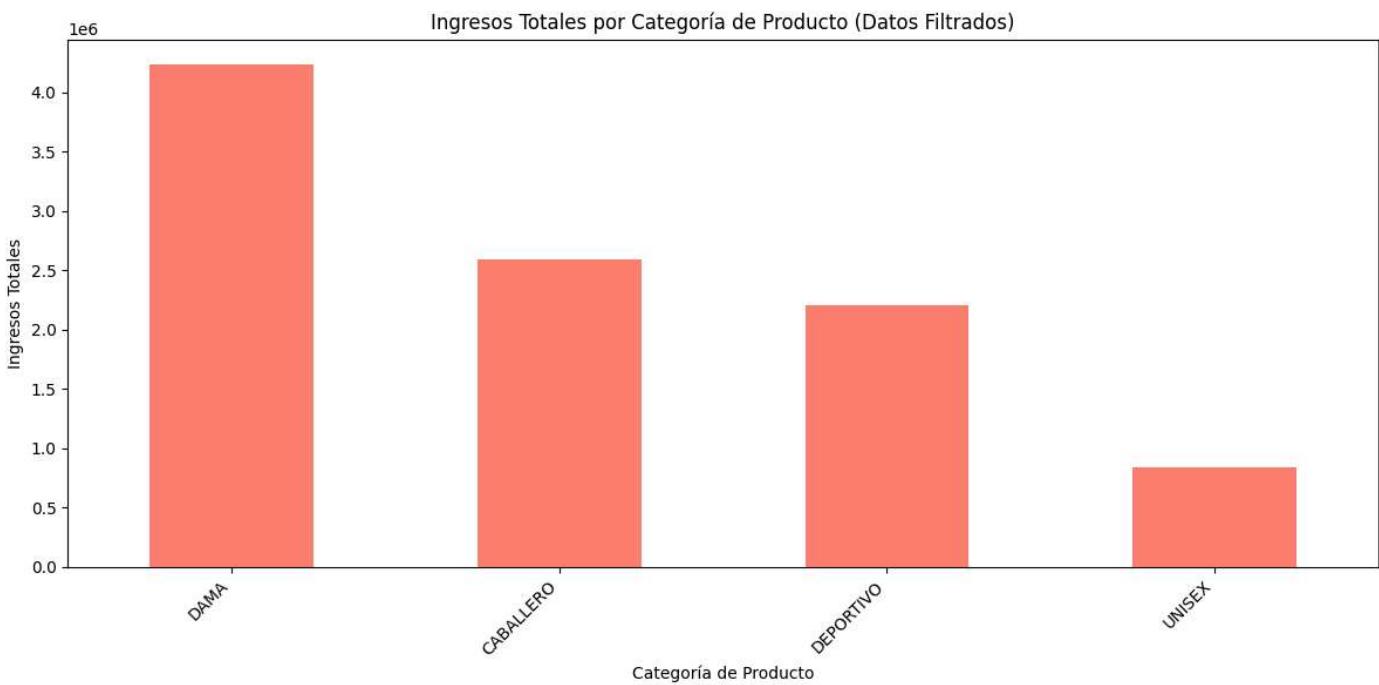
Ejemplo: Recalcular Ventas por Categoría usando `ingreso_total` del DataFrame filtrado

```
1 try:
2
3     ingresos_por_categoria = shoe_filtered.groupby('tipo_calzado')['ingreso_total'].sum().sort_values(ascending=False)
4
5     plt.figure(figsize=(12, 6))
6     ingresos_por_categoria.plot(kind='bar', color='salmon')
7     plt.title('Ingresos Totales por Categoría de Producto (Datos Filtrados)')
8     plt.xlabel('Categoría de Producto')
9     plt.ylabel('Ingresos Totales')
10    plt.xticks(rotation=45, ha='right')
11    plt.tight_layout()
12    plt.show()
13
14    print("\nResultados de Ingresos por Categoría (usando datos filtrados y columna derivada):")
```

```

15     print(ingresos_por_categoria)
16
17 except KeyError:
18     print("\nError recalculating ingresos_por_categoria: Check column names in the filtered DataFrame.")
19

```



Resultados de Ingresos por Categoría (usando datos filtrados y columna derivada):

```

tipo_calzado
DAMA      4232603
CABALLERO 2588934
DEPORTIVO 2200900
UNISEX    833668
Name: ingreso_total, dtype: int64

```

✓ Ejemplo: Recalcular Correlación entre Precio Unitario y Cantidad vendida usando datos filtrados

```

1 if not shoe_filtered.empty:
2     correlacion_precio_cantidad_filtered = shoe_filtered['precio_unitario'].corr(shoe_filtered['cantidad'])
3     print(f"\nCoeficiente de correlación entre Precio Unitario y Cantidad Vendida (Datos Filtrados): {correlacion_precio_cantidad_filtered}")
4 else:
5     print("\nCorrelation calculation skipped as the filtered DataFrame is empty.")
6

```

Coeficiente de correlación entre Precio Unitario y Cantidad Vendida (Datos Filtrados): -0.01059976719247022

✓ Ejemplo: Recalcular Ventas Totales Mensuales usando `ingreso_total` del DataFrame filtrado

```

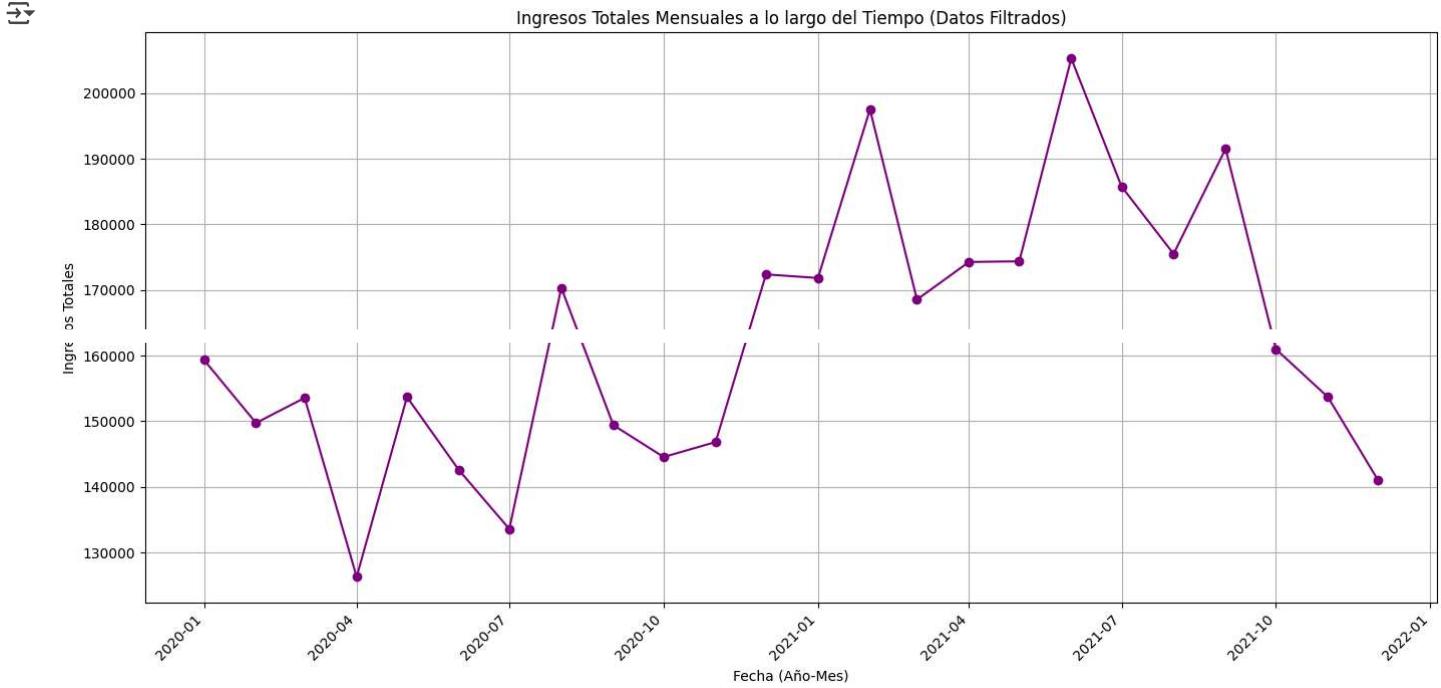
1 if not shoe_filtered.empty:
2     shoe_filtered['año_mes'] = shoe_filtered['fecha_venta'].dt.to_period('M')
3     ingresos_estacionales_filtered = shoe_filtered.groupby('año_mes')['ingreso_total'].sum().reset_index(name='ingresos_totales')
4     ingresos_estacionales_filtered['año_mes'] = ingresos_estacionales_filtered['año_mes'].dt.to_timestamp()
5     ingresos_estacionales_filtered = ingresos_estacionales_filtered.sort_values(by='año_mes')
6
7     # se visualiza
8     plt.figure(figsize=(14, 7))
9     plt.plot(ingresos_estacionales_filtered['año_mes'], ingresos_estacionales_filtered['ingresos_totales'], marker='o', linestyle='-', c
10    plt.title('Ingresos Totales Mensuales a lo largo del Tiempo (Datos Filtrados)')
11    plt.xlabel('Fecha (Año-Mes)')
12    plt.ylabel('Ingresos Totales')
13    plt.grid(True)

```

```

14     plt.xticks(rotation=45, ha='right')
15     plt.tight_layout()
16     plt.show()
17
18     print("\nResultados de Ingresos Totales Mensuales (usando datos filtrados y columna derivada):")
19     print(ingresos_estacionales_filtered.head())
20 else:
21     print("\nIngresos Totales Mensuales calculation and plotting skipped as the filtered DataFrame is empty.")

```



Resultados de Ingresos Totales Mensuales (usando datos filtrados y columna derivada):

año_mes	ingresos_totales
2020-01-01	159343
2020-02-01	149742
2020-03-01	153562
2020-04-01	126307
2020-05-01	153689

✓ Ejemplo: Recalcular Rendimiento de Ventas por Local usando `ingreso_total` del DataFrame filtrado

```

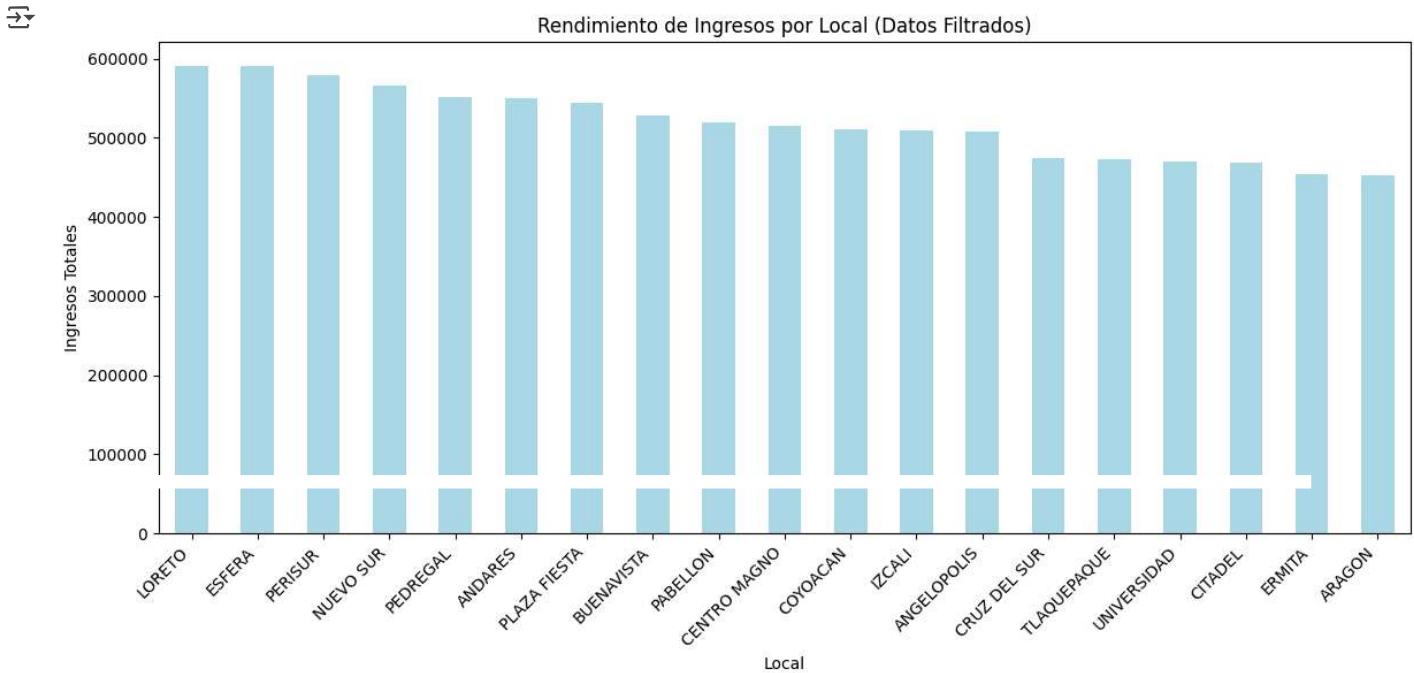
1
2 local_column_name = 'local'
3
4 if local_column_name in shoe_filtered.columns and not shoe_filtered.empty:
5     ingresos_por_local = shoe_filtered.groupby(local_column_name)[['ingreso_total']].sum().sort_values(ascending=False)
6
7     # grafica
8     plt.figure(figsize=(12, 6))
9     ingresos_por_local.plot(kind='bar', color='lightblue')
10    plt.title('Rendimiento de Ingresos por Local (Datos Filtrados)')
11    plt.xlabel('Local')
12    plt.ylabel('Ingresos Totales')
13    plt.xticks(rotation=45, ha='right')
14    plt.tight_layout()
15    plt.show()
16
17    print(f"\nResultados de Ingresos por {local_column_name} (usando datos filtrados y columna derivada):")
18    print(ingresos_por_local)
19 elif local_column_name not in shoe_filtered.columns:

```

```

20     print(f"\nError: The column '{local_column_name}' was not found in the filtered DataFrame. Cannot calculate performance by {local_cc}
21 else:
22     print(f"\nIngresos by {local_column_name} calculation and plotting skipped as the filtered DataFrame is empty.")

```



Resultados de Ingresos por local (usando datos filtrados y columna derivada):

```

local
LORETO      591393
ESFERA      590381
PERISUR     579403
NUEVO SUR   566554
PEDREGAL    550911
ANDARES     558209
PLAZA FIESTA 543727
BUENAVISTA   527952
PABELLON    518999
CENTRO MAGNO 515695
COYOACAN    510356
IZCALI       509376
ANGELOPOLIS 507883
CRUZ DEL SUR 474922
TLAQUEPAQUE 472801
UNIVERSIDAD  469643
CITADEL      468875
ERMITA       454357
ARAGON       452668
Name: ingreso_total, dtype: int64

```

▼ Ventas por mes y año

```

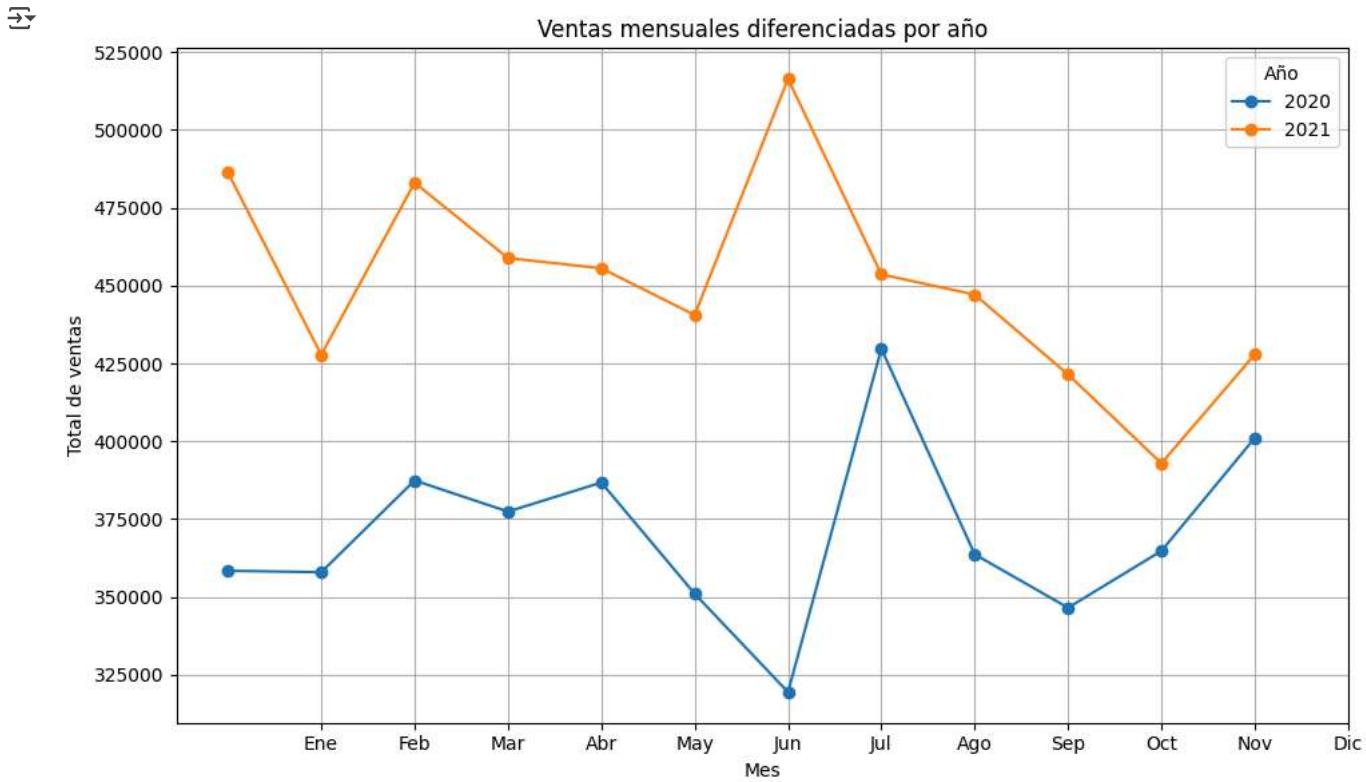
1 # Agrupar por año y mes, y sumar las ventas
2 ventas_por_año_mes = shoe_filtered.groupby(['año', 'mes'])['total_venta'].sum().reset_index()
3
4 # Ordenar por año y mes
5 ventas_por_año_mes = ventas_por_año_mes.sort_values(['año', 'mes'])
6
7 # Crear un gráfico con una línea por cada año
8 plt.figure(figsize=(10, 6))
9
10 # Obtener los años únicos
11 años = ventas_por_año_mes['año'].unique()
12
13 # Etiquetas de los meses
14 meses_labels = ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun',
15                 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic']
16

```

```

17 # Graficar cada año con su propia línea
18 for año in años:
19     datos_año = ventas_por_año_mes[ventas_por_año_mes['año'] == año]
20     plt.plot(
21         datos_año['mes'],
22         datos_año['total_venta'],
23         marker='o',
24         label=str(año)
25     )
26
27 # Configuración del gráfico
28 plt.title('Ventas mensuales diferenciadas por año')
29 plt.xlabel('Mes')
30 plt.ylabel('Total de ventas')
31 plt.xticks(ticks=range(1, 13), labels=meses_labels)
32 plt.legend(title='Año')
33 plt.grid(True)
34 plt.tight_layout()
35 plt.show()

```



▼ Ciudad con ventas más bajas

```

1 # Agrupar por ciudad y sumar ventas
2 ventas_por_local = shoe_filtered.groupby('ciudad')['total_venta'].sum().reset_index()

```

```

1 # Ordenar de menor a mayor
2 ventas_bajas = ventas_por_local.sort_values('total_venta')
3 ventas_bajas
4

```

	ciudad	total_venta
3	PUEBLA	1574198
1	GUADALAJARA	2057704
2	MONTERREY	2169537
0	CIUDAD DE MÉXICO	4054666

▼ Talla de calzado más vendido

```

1 ventas_por_tamaño = shoe_filtered.groupby('tamaño')['cantidad'].sum().reset_index().sort_values('cantidad', ascending=False)
2
3 ventas_por_tamaño.head()
4

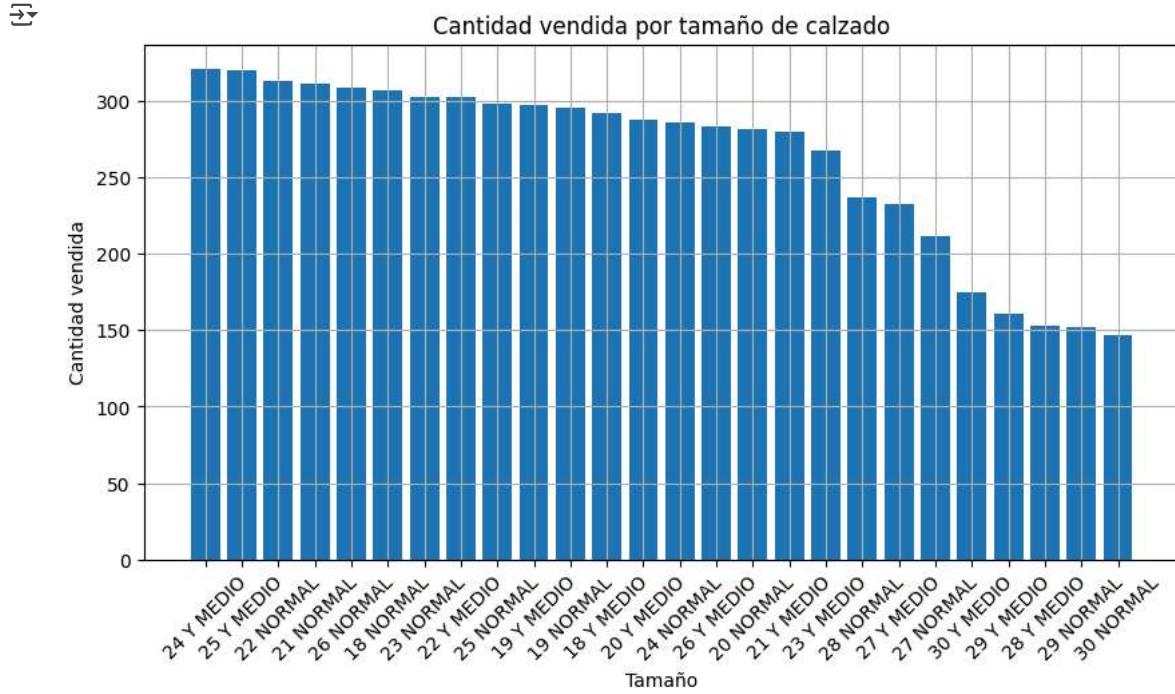
```

	tamaño	cantidad
13	24 Y MEDIO	321
15	25 Y MEDIO	320
8	22 NORMAL	313
6	21 NORMAL	311
16	26 NORMAL	309

```

1 plt.figure(figsize=(10,5))
2 plt.bar(ventas_por_tamaño['tamaño'].astype(str), ventas_por_tamaño['cantidad'])
3 plt.title('Cantidad vendida por tamaño de calzado')
4 plt.xlabel('Tamaño')
5 plt.ylabel('Cantidad vendida')
6 plt.xticks(rotation=45)
7 plt.grid(True)
8 plt.show()

```



▼ Calzados más vendidos en cada ciudad

```

1 # Agrupar por ciudad y tipo de calzado, y sumar las ventas
2 ventas_por_tipo = shoe_filtered.groupby(['ciudad', 'tipo_calzado'])['total_venta'].sum().reset_index()

1 # Ordenar de mayor a menor dentro de cada ciudad
2 ventas_por_tipo = ventas_por_tipo.sort_values(['ciudad', 'total_venta'], ascending=[True, False])
3 ventas_por_tipo.head(10)

```

	ciudad	tipo_calzado	total_venta
1	CIUDAD DE MÉXICO	DAMA	1757855
0	CIUDAD DE MÉXICO	CABALLERO	1071117
2	CIUDAD DE MÉXICO	DEPORTIVO	886547
3	CIUDAD DE MÉXICO	UNISEX	339147
5	GUADALAJARA	DAMA	883848
4	GUADALAJARA	CABALLERO	544523
6	GUADALAJARA	DEPORTIVO	460552
7	GUADALAJARA	UNISEX	168781
9	MONTERREY	DAMA	877880
8	MONTERREY	CABALLERO	572422

✓ Uso de todas las graficas

```

1 if 'tipo_calzado' in shoe.columns and 'marca' in shoe.columns:
2     contingency_table = pd.crosstab(shoe['tipo_calzado'], shoe['marca'])
3     print("Tabla de Contingencia entre Tipo de Calzado y Marca:")
4     print(contingency_table)
5 else:
6     print("Skipping Contingency Table: 'tipo_calzado' or 'marca' column not found.")

```

→ Skipping Contingency Table: 'tipo_calzado' or 'marca' column not found.

✓ i. Tablas de frecuencias

```

1 if 'tipo_calzado' in shoe.columns:
2     frequency_table_tipo = shoe['tipo_calzado'].value_counts().reset_index()
3     frequency_table_tipo.columns = ['Tipo Calzado', 'Frecuencia']
4     print("Tabla de Frecuencias para Tipo de Calzado:")
5     print(frequency_table_tipo)
6 else:
7     pass
8

```

→ Tabla de Frecuencias para Tipo de Calzado:

Tipo Calzado	Frecuencia
0 DAMA	2981
1 CABALLERO	1718
2 DEPORTIVO	1234
3 UNISEX	826

✓ ii. Tablas de valores (Example: Simple view of unique values in 'local')

```

1 print("\n--- ii. Tablas de valores ---")
2 if 'local' in shoe.columns:
3     unique_locals = shoe['local'].unique()
4     print("Valores únicos en la columna 'local':")
5     for local in unique_locals:
6         print(local)
7 else:
8     pass

```

→ --- ii. Tablas de valores ---
 Valores únicos en la columna 'local':
 ARAGON
 CRUZ DEL SUR
 TLAQUEPAQUE
 ERMITA
 ESFERA
 PERISUR
 PEDREGAL

NUEVO SUR
 ANDARES
 LORETO
 BUENAVISTA
 PLAZA FIESTA
 IZCALI
 UNIVERSIDAD
 COYOACAN
 CITADEL
 PABELLON
 CENTRO MAGNO
 ANGELOPOLIS

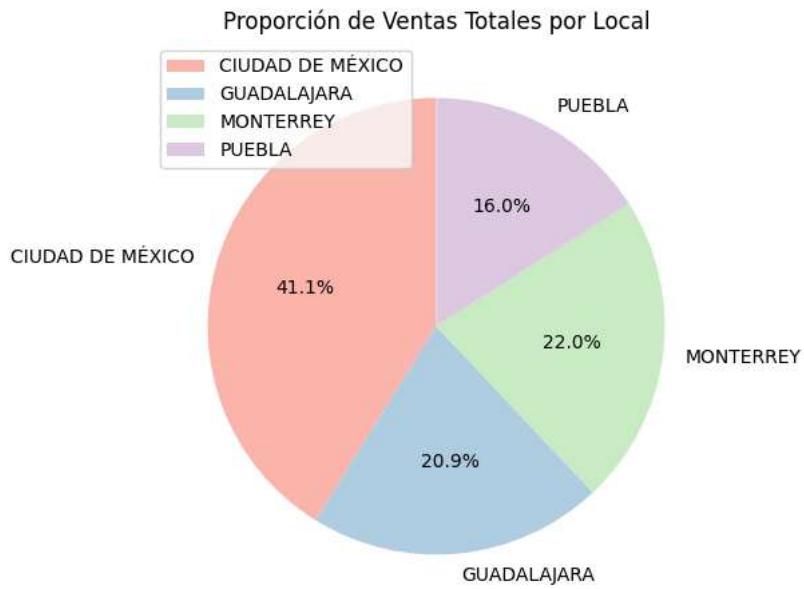
iv. Gráfica de sectores (pastel)

```

1 if 'ventas_por_local' in locals() and not ventas_por_local.empty:
2   plt.figure(figsize=(8, 8))
3   # Specify the 'total_venta' column as the data for the pie chart
4   ventas_por_local.plot(kind='pie', y='total_venta', autopct='%1.1f%%', startangle=90, colors=plt.cm.Pastel1.colors, labels=ventas_por_l
5   plt.title('Proporción de Ventas Totales por Local')
6   plt.ylabel('')
7   plt.tight_layout()
8   plt.show()
9 else:
10  pass
11
12

```

→ <Figure size 800x800 with 0 Axes>



v. Gráfica de barras.

```

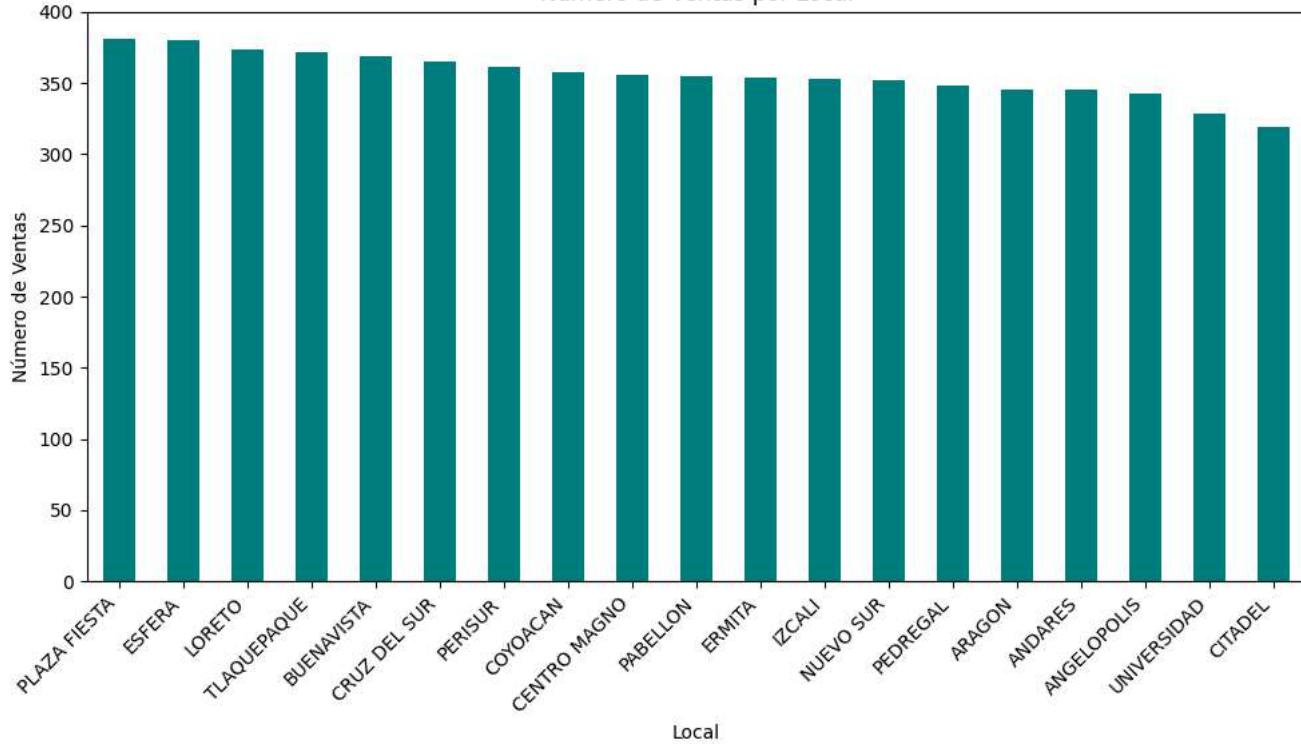
1 print("\n--- v. Gráfica de barras (Example: Count of sales by local) ---")
2 if 'local' in shoe.columns:
3   sales_count_by_local = shoe['local'].value_counts()
4   plt.figure(figsize=(10, 6))
5   sales_count_by_local.plot(kind='bar', color='teal')
6   plt.title('Número de Ventas por Local')
7   plt.xlabel('Local')
8   plt.ylabel('Número de Ventas')
9   plt.xticks(rotation=45, ha='right')
10  plt.tight_layout()
11  plt.show()
12 else:
13  pass
14

```



--- v. Gráfica de barras (Example: Count of sales by local) ---

Número de Ventas por Local



▼ vi. Gráfica de líneas.

```
1 if 'fecha_venta' in shoe.columns and 'cantidad' in shoe.columns:  
2  
3     quantity_over_time = shoe.groupby('fecha_venta')['cantidad'].sum()  
4     plt.figure(figsize=(14, 7))  
5     quantity_over_time.plot(kind='line', marker='o', linestyle='-', color='forestgreen')  
6     plt.title('Cantidad Total Vendida a lo largo del Tiempo')  
7     plt.xlabel('Fecha de Venta')  
8     plt.ylabel('Cantidad Vendida')  
9     plt.grid(True)  
10    plt.tight_layout()  
11    plt.show()
```