

Diana Cabezas Falcón
 Cristhian Castro Celi
 Joan Cevallos Bedón
 Claudio Arias Piedra

Actividad Grupal Semana 4 – Caso Práctico

- Importe la base de datos a una base en Jupyter Notebook con pandas.

```
df = pd.read_csv('/content/drive/MyDrive/tarea_s4/Walmart(1).csv')
df.head(n=5)
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.900	0	42.310	2.572	211.096	8.106
1	1	12-02-2010	1641957.440	1	38.510	2.548	211.242	8.106
2	1	19-02-2010	1611968.170	0	39.930	2.514	211.289	8.106
3	1	26-02-2010	1409727.590	0	46.630	2.561	211.320	8.106
4	1	05-03-2010	1554806.680	0	46.500	2.625	211.350	8.106

```
df.rename(columns = {
    'Store':'Tienda',
    'Date':'Fecha_Semana_Venta',
    'Weekly_Sales':'Ventas_Semanales',
    'Holiday_Flag':'Es_Semana_Feriado',
    'Temperature':'Temperatura',
    'Fuel_Price':'Precio_Combustible_Region',
    'CPI':'Indice_Precios_Consm',
    'Unemployment':'Tasa_Desempleo'
}, inplace = True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
  0   Tienda          6435 non-null   int64  
  1   Fecha_Semana_Venta 6435 non-null   object  
  2   Ventas_Semanales 6435 non-null   float64 
  3   Es_Semana_Feriado 6435 non-null   int64  
  4   Temperatura      6435 non-null   float64 
  5   Precio_Combustible_Region 6435 non-null   float64 
  6   Indice_Precios_Consm   6435 non-null   float64 
  7   Tasa_Desempleo     6435 non-null   float64 
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

```
## Corregir el tipo de dato para la variable Fecha_Semana_Venta y
## Es_Semana_Feriado
dff['Fecha_Semana_Venta'] = pd.to_datetime(dff['Fecha_Semana_Venta'],
format='%d-%m-%Y')
dff['Es_Semana_Feriado'] = dff['Es_Semana_Feriado'].astype('category')
```

`df.dtypes`

Tienda	int64
Fecha_Semana_Venta	datetime64[ns]
Ventas_Semanales	float64
Es_Semana_Feriado	category
Temperatura	float64
Precio_Combustible_Region	float64
Indice_Precios_Consm	float64
Tasa_Desempleo	float64
dtype: object	

2. Obtenga los descriptivos resúmenes de la base de datos e identifique a las variables numéricas y categóricas. ¿Hay algo que le llame la atención?

`df.describe(include=np.number).round(2)`

	Tienda	Ventas_Semanales	Temperatura	Precio_Combustible_Region	Indice_Precios_Consm	Tasa_Desempleo
count	6435.000	6435.000	6435.000	6435.000	6435.000	6435.000
mean	23.000	1046964.880	60.660	3.360	171.580	8.000
std	12.990	564366.620	18.440	0.460	39.360	1.880
min	1.000	209986.250	-2.060	2.470	126.060	3.880
25%	12.000	553350.100	47.460	2.930	131.740	6.890
50%	23.000	960746.040	62.670	3.440	182.620	7.870
75%	34.000	1420158.660	74.940	3.740	212.740	8.820
max	45.000	3818686.450	100.140	4.470	227.230	14.310

```
var_cuantitativas = df.select_dtypes(include = np.number)
var_cualitativas = df.select_dtypes(exclude = np.number)
```

`var_cuantitativas.columns`

```
Index(['Tienda', 'Ventas_Semanales', 'Temperatura',
       'Precio_Combustible_Region', 'Indice_Precios_Consm', 'Tasa_Desempleo'],
      dtype='object')
```

`var_cualitativas.columns`

```
Index(['Fecha_Semana_Venta', 'Es_Semana_Feriado'], dtype='object')
```

- El método `.describe()` aplicado al DataFrame (df) proporciona estadísticas resumidas de las variables numéricas en el conjunto de datos, como la cuenta, la media, la desviación estándar, los valores mínimos, los percentiles (25%, 50%, 75%), y los valores máximos. Entre los puntos más relevantes se presentan:

- El total de observaciones para todas las variables es de 6435, lo que nos indica un posible indicio de que el conjunto de datos esta completo y no existe datos con errores del tipo NaN o null.
- La variable “Tienda” actúa como un identificador de los locales y se puede deducir (mediante los valores de mínimo y máximo) que se ha registrado un total de 45 diferentes puntos de venta.
- El comportamiento de las desviaciones estándar en las variables evidencia las diferentes unidades de medición y el nivel de dispersión interna con respecto a la media.

- En la variable temperatura observamos un mínimo negativo, un valor esperado por el tipo de medición que se realiza en cada local, además se puede indicar que el rango de esta variable (máximo – mínimo) es muy elevado.
 - El comportamiento de los estadísticos descriptivos sobre la media y mediana puede indicarnos, de forma preliminar, la posible forma de la distribución y su sesgo:
 - o Si la media es mayor a la mediana (y moda) estaríamos ante una distribución asimétrica positivo (sesgo hacia la derecha).
 - o Si la media es menor a la mediana (y moda) estaríamos ante una distribución asimétrica negativa (sesgo hacia la izquierda).
 - Las variables numéricas se identifican utilizando el método `select_dtypes(include=np.number)`, que selecciona las columnas del DataFrame que son de tipo numérico.
 - Las variables categóricas se identifican utilizando el método `select_dtypes(exclude=np.number)`, que selecciona las columnas del DataFrame que no son de tipo numérico.
 - Los resultados que hemos determinado son:
 - o Hay seis variables numéricas identificadas: 'Tienda', 'Ventas_Semanales', 'Temperatura', 'Precio_Combustible_Region', 'Indice_Precios_Consm', 'Tasa_Desempleo'.
 - o Hay dos variables categóricas (o no numéricas) identificadas: 'Fecha_Semana_Venta', 'Es_Semana_Feriado'.
 - Con esto podemos comprender la naturaleza de los datos y prepararlos para nuestros análisis posteriores. Además, se proporcionan algunas estadísticas descriptivas clave que pueden ayudar a identificar patrones o anomalías en los datos, de lo cual no hemos identificado al momento datos que llamen la atención.
3. Evalúe si la base contiene datos perdidos.

```
df_mod = df.copy()
df.isnull().sum()
```

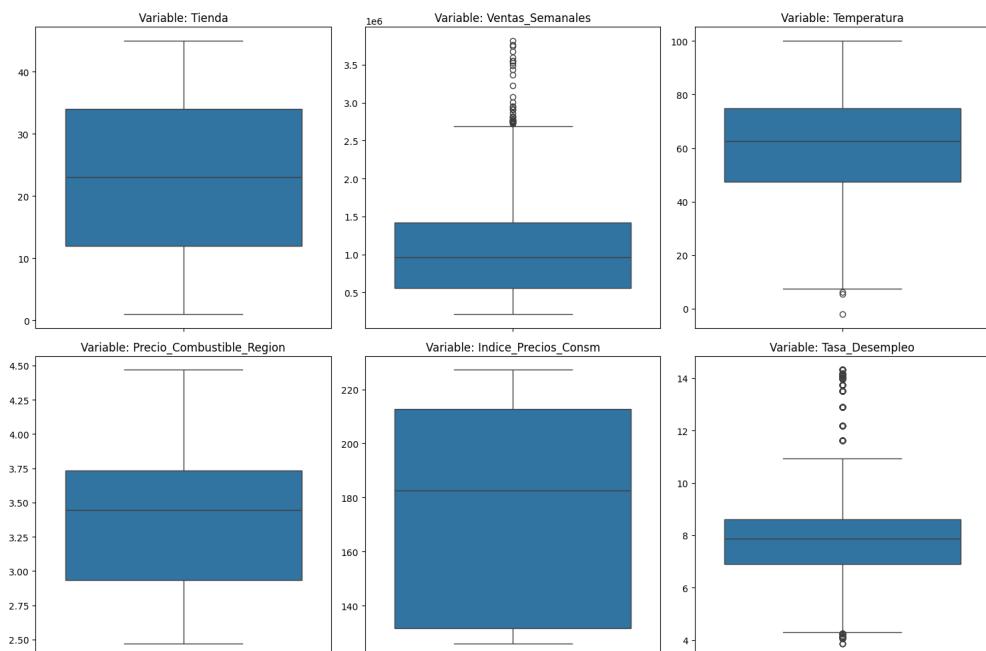
Tienda	0
Fecha_Semana_Venta	0
Ventas_Semanales	0
Es_Semana_Feriado	0
Temperatura	0
Precio_Combustible_Region	0
Indice_Precios_Consm	0
Tasa_Desempleo	0
dtype: int64	

```
perc_na = (df_mod.isnull().sum().sum() / df_mod.shape[0]) * 100
print('El porcentaje de valores nulos total en el dataset es de:
      ,round(perc_na,3),%)'
```

El porcentaje de valores nulos total en el dataset es de: 0.0 %

- Calidad de los Datos: la ausencia de datos nulos sugiere que la recolección de datos fue completa o que ya ha sido preprocesada para manejar los valores nulos. Esto nos da un buen indicador de la calidad de los datos en términos de completitud.
 - Análisis Estadístico y Modelado: se puede proceder con los análisis estadísticos adicionales, como correlaciones o análisis de regresión, sin preocuparse por la distorsión que los valores nulos podrían introducir en los resultados.
 - Representatividad: aunque no hay valores nulos, aún es necesario considerar si existen datos atípicos que se desmarcan de los límites normales y que pueden afectar de manera significativa los resultados futuros.
4. Evalúe si alguna de las variables contiene datos atípicos (outliers). De ser el caso, detalle cuáles y qué método estadístico aplicarán para corregir.

```
fig, axs = plt.subplots(2, 3, figsize=(15, 10))
for i, c in enumerate(var_cuantitativas):
    sns.boxplot(df_mod[c], ax=axs[i//3, i%3]).set_title('Variable: {}'.format(c))
    axs[i//3, i%3].set_ylabel('')
plt.tight_layout()
plt.show()
```



```
#### Tratamiento de los datos atípicos
#### 'Ventas_Semanales'
Q1 = df_mod['Ventas_Semanales'].quantile(0.25)
Q3 = df_mod['Ventas_Semanales'].quantile(0.75)
IQR = Q3 - Q1
df_mod = df_mod[((df_mod['Ventas_Semanales'] < (Q1 - 1.5 * IQR)) | (df_mod['Ventas_Semanales'] > (Q3 + 1.5 * IQR)))]
```

```

print(df_mod.shape,'cambio 1')
### 'Tasa_Desempleo'
Q1 = df_mod['Tasa_Desempleo'].quantile(0.25)
Q3 = df_mod['Tasa_Desempleo'].quantile(0.75)
IQR = Q3 - Q1
df_mod = df_mod[((df_mod['Tasa_Desempleo'] < (Q1 - 1.5 * IQR)) |
|(df_mod['Tasa_Desempleo'] > (Q3 + 1.5 * IQR)))]
print(df_mod.shape,'cambio 2 -> Final de filas y columnas corregido')
  
```

```

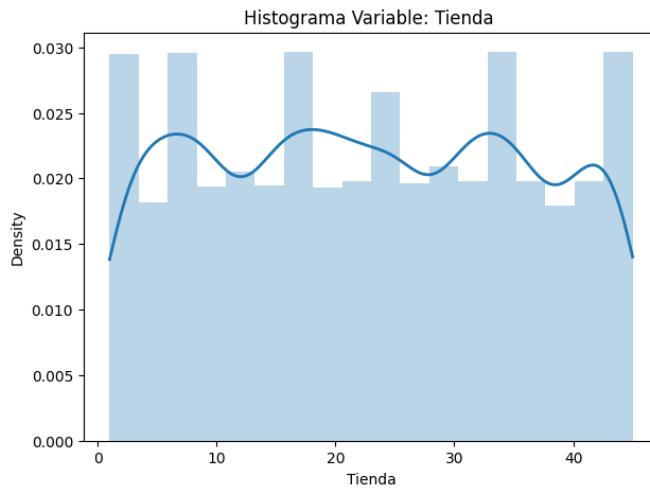
(6401, 8)  cambio 1
(5920, 8)  cambio 2 -> Final de filas y columnas corregido
  
```

- Ventas_Semanales: Esta variable contiene varios valores que se consideran datos atípicos, ya que en el boxplot se extienden significativamente por encima del tercer cuartil (Q3). Estos representan semanas con ventas inusualmente altas.
- Tasa_Desempleo: Esta variable muestra varios valores atípicos, particularmente por debajo del primer cuartil (Q1), indicando períodos con tasas de desempleo inusualmente bajas.
- Para corregir los datos atípicos en estas variables, se aplicó el método IQR:
 - Método IQR: Este método fue utilizado para limpiar los datos atípicos, eliminando cualquier valor que se encuentre fuera del rango de $[Q1 - 1.5 \cdot IQR, Q3 + 1.5 \cdot IQR]$. Esto se lo realizó para mitigar el impacto de los valores extremos en los futuros análisis.

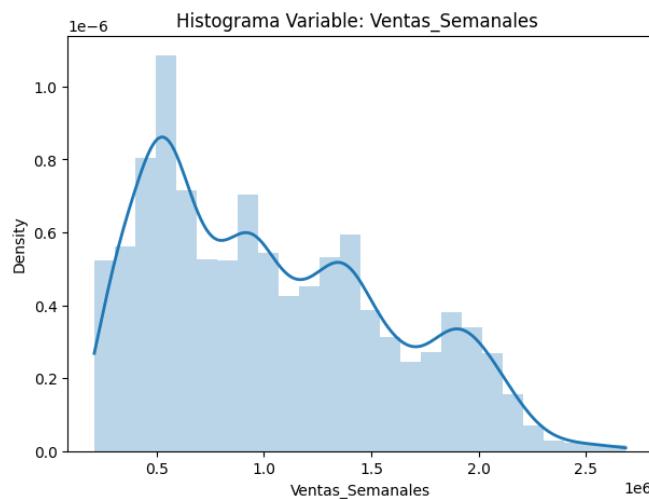
5. Grafique las distribuciones de las variables y a priori comente sobre ellas.

```

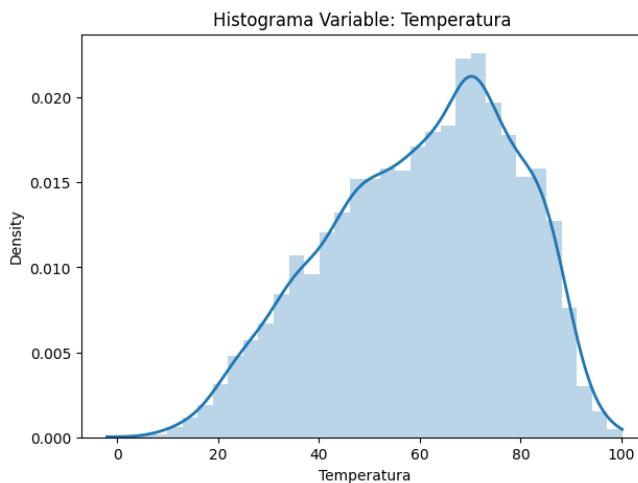
for col in df_mod.columns.drop(['Fecha_Semana_Venta','Es_Semana_Feriado']):
    fig = plt.figure(figsize = (7,5))
    sns.histplot(data = df_mod,
                  x=col,
                  stat='density',
                  kde=True,
                  alpha= 0.3,
                  edgecolor='none',
                  common_norm=False,
                  line_kws={'linewidth':2}).set_title("Histograma      Variable: {}"
                  .format(col))
    plt.show()
  
```



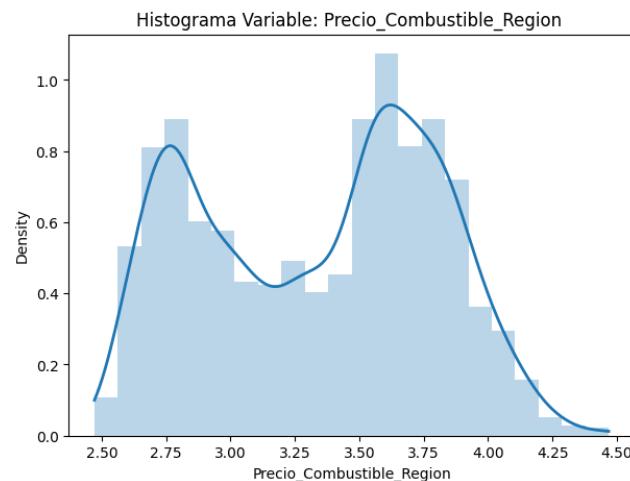
- Tienda: la variable tienda es un identificador de los locales de Walmart registrados en la base de datos, observamos que la muestra es dispareja porque encontramos más registros en ciertas tiendas específicas de la muestra, aunque puede ser un comportamiento normal por las propias características del nivel de observaciones en cada tienda, se debería explorar otras opciones para tratar de igual el número de observaciones.



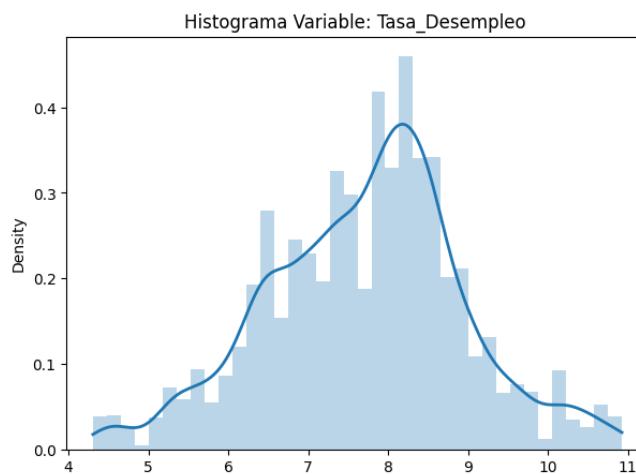
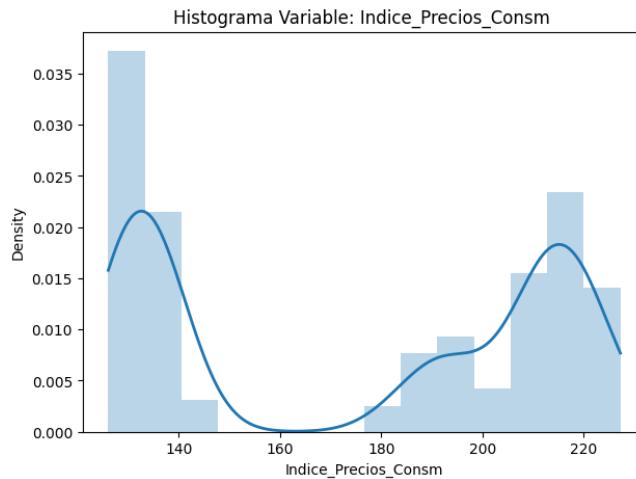
- Variabilidad en Ventas Semanales: las ventas semanales muestran una distribución con cola a la derecha, con la presencia de outliers significativos. Esto indica periodos de alta actividad comercial o eventos específicos que causan picos en las ventas.



- Temperatura: la distribución de la temperatura parece aproximarse a una distribución normal, lo que puede mostrar que la variable podría tener un comportamiento predecible a lo largo del tiempo, con algunas semanas más frías o cálidas que podrían estar asociadas a patrones estacionales.

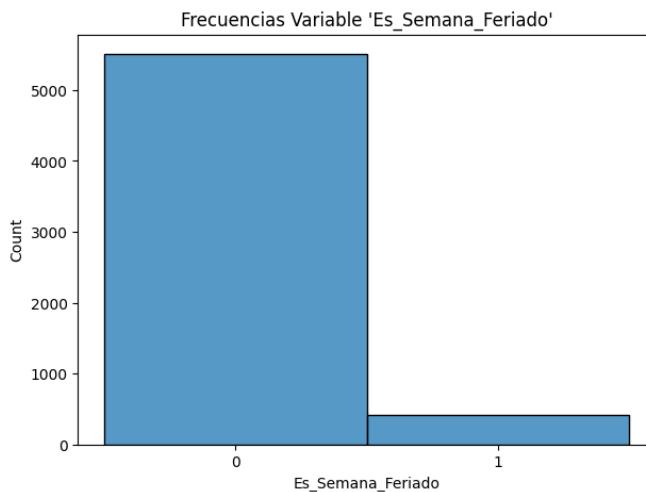


- Precios de Combustible y Índice de Precios al Consumidor: ambas variables muestran distribuciones bimodales, lo que sugiere que podría haber factores subyacentes que crean estos dos grupos distintos, como diferencias regionales o cambios económicos a lo largo del tiempo.



- Tasa de Desempleo: esta variable también muestra una variabilidad significativa, con múltiples picos en su distribución que podrían estar relacionados con eventos económicos o políticas laborales que varían en el tiempo.

```
fig = plt.figure(figsize=(7,5))
sns.histplot(data=df_mod,
             x='Es_Semana_Feriado',
             discrete=True)
plt.xticks(ticks=[0,1])
plt.title("Frecuencias Variable 'Es_Semana_Feriado'")
plt.show()
```



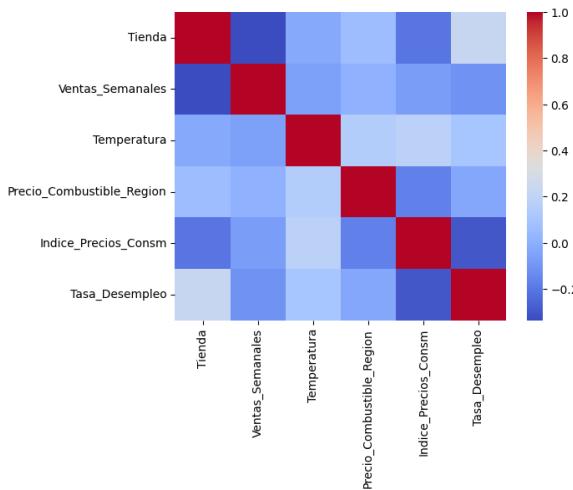
- Feriados: hay una clara desigualdad en la frecuencia de semanas normales versus semanas de feriados, lo que es de esperar dado el calendario típico de feriados. Se debería analizar o estudiar una posible interferencia debido a que no existe un balanceo de clases.

6. Obtenga las correlaciones entre los datos de corte numérico.

```
var_cuantitativas.corr().style.background_gradient(cmap='coolwarm')
```

	Tienda	Ventas_Semanales	Temperatura	Precio_Combustible_Region	Indice_Precios_Consm	Tasa_Desempleo
Tienda	1.000000	-0.335332	-0.022659	0.060023	-0.209492	0.223531
Ventas_Semanales	-0.335332	1.000000	-0.063810	0.009464	-0.072634	-0.106176
Temperatura	-0.022659	-0.063810	1.000000	0.144982	0.176888	0.101158
Precio_Combustible_Region	0.060023	0.009464	0.144982	1.000000	-0.170642	-0.034684
Indice_Precios_Consm	-0.209492	-0.072634	0.176888	-0.170642	1.000000	-0.302020
Tasa_Desempleo	0.223531	-0.106176	0.101158	-0.034684	-0.302020	1.000000

```
sns.heatmap(var_cuantitativas.corr(),cmap='coolwarm')
plt.show()
```



Dentro de la siguiente gráfica podemos ver las distintas variables de corte en numérico en la cual podemos visualizar que la correlación más alta es entre las ventas y número de tiendas, pero en este caso es una correlación negativa es decir entre más número de tiendas hay las ventas semanales baja por tienda. Sin embargo, hay que mencionar que la variable “Tienda” corresponde a un elemento identificador de locales y, por lo tanto, las correlaciones con otras variables deben ser analizadas con precaución. Si bien su medición es numérica, la esencia de esta variable buscar identificar o hacer referencia a un cierto local específico y no al número de locales en total.

La segunda correlación más alta es entre el índice de precios al consumidor y la tasa de desempleo teniendo una correlación negativa de 0.302, haciendo la misma referencia que entre las tiendas y las ventas semanales.

De la misma manera vemos que hay una correlación positiva entre el índice de precios y la temperatura teniendo un 0.176 de correlación entre estas dos variables haciendo referencia a que si sube la temperatura el índice de precios del consumidor va a aumentar.

Y dentro de esta tabla también podemos ver que la correlación más baja entre las distintas variables son las ventas semanales con respecto al precio de combustible que es del 0.009, mostrándonos de esta manera que las ventas semanales y el precio del combustible se encuentran correlacionadas de una forma muy débil.

Es importante recordar que la correlación no implica causalidad, esto quiere decir que no significa que, si una variable cambia, causa directamente un cambio en la otra variable ya que pueden estar involucrados otros factores.

7. Comente que variable escogerán como variable dependiente y que variables introducirán a su modelo.

La variable dependiente es la variable de ventas semanales, inicialmente planteamos el uso de todas las variables disponibles para realizar un modelo inicial completo entre todas las observaciones y luego proceder con las pruebas de los supuestos correspondientes, así se podrá determinar la necesidad o no de utilizar modelos más complejos como los modelos de datos de panel. La presencia de variables como el tiempo y el identificador de tienda proponen, de manera

preliminar, la necesidad de un modelado en datos de panel, aunque aún no se encuentra claro o definido el tipo de efectos a aplicar.

Así que las variables independientes a ser utilizadas de forma preliminar son: si esa semana es feriado, temperatura, precio de combustible por región, índice de precios al consumidor y la tasa de desempleo.

```
# Variables a ser utilizadas (de forma preliminar)
# No se debe considerar 'Tienda', 'Fecha_Semana_Venta' y 'Ventas_Semanales'
df_mod.columns
[Temperatura', 'Precio_Combustible_Region', 'Indice_Precios_Consm',
'Tasa_Desempleo'],
```

8. Indique que tipo de modelación realizarán y porqué.

En primer lugar, para la modelación que vamos a realizar hemos asignamos las columnas de referencia como índice que son fechas de semana de venta y tienda. En segundo lugar, preparamos las variables para un análisis de regresión en el cual definimos la nuestra variable objetivo [ventas semanales] de nuestro DataFrame y luego definimos la matriz de características eliminando la columna de ventas y fecha de venta del DataFrame. Esa matriz contiene las variables predictoras que vamos a utilizar. Agregamos una columna constante a la matriz ya que es necesario para poder estimar el término de intercepción en el modelo de regresión. El tipo de modelación final se plantea utilizar un modelo de datos de panel (si los resultados de los test lo justifican) ya que el objetivo final es el de estimar las ventas semanales de cada una de las tiendas de acuerdo con la variación presente en las variables independientes. En sentido, se puede mencionar que la construcción del data set muestra una configuración adecuada para realizar datos de panel al reportar múltiples observaciones de la misma unidad y grupo durante múltiples períodos de tiempo.

Aquí aplicamos el modelo 1 (preliminar) que es el modelo Pooled OLS (mínimos cuadrados agrupados) a los datos, dandonos como resultado el modelo Pooled OLS ajustado, el cual vamos a usar para analizar los coeficientes estimados, el error estándar, los estadísticos de prueba y otros resultados del modelo.

```
# Asignar las columnas de referencia como indices
df_mod = df_mod.set_index(['Fecha_Semana_Venta','Tienda'])
```

```
# Crear una columna con la Fecha de venta para cada unidad
fechas_venta
df_mod.index.get_level_values('Fecha_Semana_Venta').to_list()
df_mod['Fecha_Venta'] = pd.Categorical(fechas_venta)
```

```
y = df_mod['Ventas_Semanales']
X = df_mod.drop(['Ventas_Semanales','Fecha_Venta'],axis=1)
X = sm.tools.tools.add_constant(X)
```

X.dtypes

const	float64
Es_Semana_Feriado	category
Temperatura	float64
Precio_Combustible_Region	float64
Indice_Precios_Consm	float64
Tasa_Desempleo	float64
dtype: object	

```
modelo1 = PooledOLS(y, X)
resultados_pooled_OLS = modelo1.fit(cov_type='clustered',
cluster_entity=True)
```

```
# Valores para heteroscedasticidad (posterior)
predicciones_pooled_OLS = resultados_pooled_OLS.predict().fitted_values
residuos_pooled_OLS = resultados_pooled_OLS.resids
```

resultados_pooled_OLS

PooledOLS Estimation Summary						
Dep. Variable:	Ventas_Semanales	R-squared:	0.0165			
Estimator:	PooledOLS	R-squared (Between):	0.0125			
No. Observations:	5920	R-squared (Within):	0.0166			
Date:	Fri, Mar 15 2024	R-squared (Overall):	0.0165			
Time:	05:05:44	Log-Likelihood	-8.662e+04			
Cov. Estimator:	Clustered					
		F-statistic:	19.789			
Entities:	143	P-value	0.0000			
Avg Obs:	41.399	Distribution:	F(5,5914)			
Min Obs:	33.000					
Max Obs:	42.000	F-statistic (robust):	234.37			
		P-value	0.0000			
Time periods:	45	Distribution:	F(5,5914)			
Avg Obs:	131.56					
Min Obs:	17.000					
Max Obs:	143.00					
Parameter Estimates						
	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	1.641e+06	4.458e+04	36.824	0.0000	1.554e+06	1.729e+06
Es_Semana_Feriado.1	5.086e+04	3.787e+04	1.3431	0.1793	-2.337e+04	1.251e+05
Temperatura	-380.75	426.50	-0.8927	0.3720	-1216.8	455.34
Precio_Combustible_Region	-1492.9	9496.7	-0.1572	0.8751	-2.011e+04	1.712e+04
Indice_Precios_Consm	-1431.7	73.895	-19.374	0.0000	-1576.5	-1286.8
Tasa_Desempleo	-4.237e+04	2496.8	-16.971	0.0000	-4.727e+04	-3.748e+04

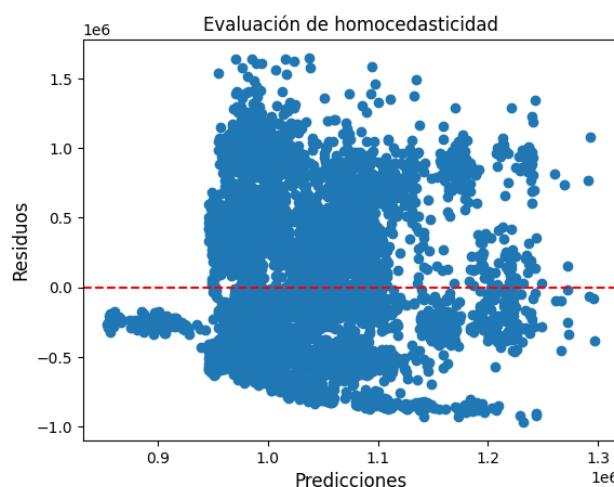
Aquí podemos apreciar los diferentes resultados del modelo en el cual tenemos, 5920 datos observados, tenemos un total de 143 grupos, observados en 45 periodos de tiempo como variable dependiente las ventas semanales, Así mismo podemos ver que el R cuadrado es muy bajo, teniendo como resultado el valor de 1.6% es decir que el modelo o la bondad de ajuste solamente explica el 1.6% de toda la variabilidad en la variable independiente con respecto a la variabilidad de las variables independientes, también debemos tener en cuenta que el R cuadrado no garantiza que el modelo sea válido, tenemos el modelo en su conjunto es estadísticamente significativo por su valor P (para el estadístico F), asimismo podemos ver que el p-valor es diferentes en cada una de las variables, teniendo

como resultado que el índice de precios al consumidor y la tasa de desempleo son significativos sobre nuestra variable dependiente (ventas semanales), de igual manera vemos que Semana Feriado, la temperatura y el precio del combustible no son variables significativas para este modelo que nos encontramos analizando.

El modelo definitivo de datos de panel nos permitirá establecer de manera más precisa el comportamiento de las unidades y grupos en el conjunto de datos con respecto al tiempo y con respecto a las variables predictoras, puesto que el modelo de datos de panel nos permite considerar múltiples parámetros individuales y temporales, teniendo como premisa que desde ahora la variabilidad puede ser tanto temporal como transversal.

9. Verifique los supuestos, de haber escogido el enfoque econométrico.

```
#### Homocedasticidad
# Test gráfico
fig, ax = plt.subplots()
ax.scatter(predicciones_pooled_OLS, residuos_pooled_OLS)
ax.axhline(0, color = 'r', ls = '--')
ax.set_xlabel('Predicciones', fontsize = 12)
ax.set_ylabel('Residuos', fontsize = 12)
ax.set_title('Evaluación de homocedasticidad', fontsize = 12)
plt.show()
```



```
## Test Breusch-Pagan
pooled_OLS_df = pd.concat([df_mod, residuos_pooled_OLS], axis=1)
pooled_OLS_df = pooled_OLS_df.drop(['Fecha_Venta','Ventas_Semanales'],
axis = 1).fillna(0)
X_ = sm.tools.tools.add_constant(df_mod['Ventas_Semanales']).fillna(0)
```

```
breusch_pagan = het_breuscpagan(pooled_OLS_df.residual, X_)
labels = ['LM-Stat', 'LM p-val', 'F-Stat', 'F p-val']
print(dict(zip(labels, breusch_pagan)))
```

```
'F-Stat': 1377.6909605418798, 'F p-val': 2.6891907587646506e-271}
```

La prueba de homocedasticidad en los residuos muestra, de forma gráfica, un comportamiento inusual conforme incrementa las predicciones, evidenciando una forma de cono, característico de un incremento importante en la varianza de los residuos procedentes del modelo, siendo un posible indicador de heterocedasticidad. Para verificar esta suposición, se ha procedido a realizar la prueba de Breusch-Pagan con un nivel de significancia del 5%, de acuerdo con los resultados presentados con un estadístico igual a 1115,91 y un valor p igual a 4.2280e-245, se puede concluir que existen evidencias suficientes para rechazar la hipótesis nula ($p\text{-valor} < \text{Alpha}$) y, por lo tanto, se confirma la presencia de heteroscedasticidad en el modelo. La prueba realizada posee las siguientes características:

- H_0 : La homocedasticidad está presente (los residuos tienen varianza constante).
- H_1 : La heterocedasticidad está presente (los residuos no tienen varianza constante).
- $\text{Alpha} = 0.05$
- Regla de decisión:
- $p > \text{Alpha} = \text{No rechace } H_0$
- $p < \text{Alpha} = \text{Rechace } H_0$

```
durbin_watson = durbin_watson(pooled_OLS_df.residual)
print('El resultado del estadístico Durbin Watson [DW] es de: ',durbin_watson)
```

```
El resultado del estadístico Durbin Watson [DW] es de:  0.08767442835798936
```

El segundo supuesto hace referencia a la no autocorrelación en los residuos, esto debido a su importante influencia en la calidad de los coeficientes y su posterior interpretación con respecto a su fiabilidad. En consecuencia, se ha utilizado el test estadístico de Durbin-Watson (DW) con las siguientes características:

- Regla de decisión:
 - * El valor medio de dos, indicaría que no se ha identificado una autocorrelación,
 - * 0 - 2 significa una autocorrelación positiva (cuanto más cerca de cero, mayor es la correlación), y
 - * 2 - 4 significa autocorrelación negativa (cuanto más cerca de cuatro, mayor es la correlación)

El resultado de la prueba DW aplicado al modelo PooledOLS registra un valor de 0.08767, indicativo de que existe una relación positiva considerable y alta entre los residuos.

Como se ha podido observar, los supuestos requeridos por el modelo PooledOLS no se han cumplido, tanto la heterocedasticidad como la autocorrelación están presentes en los residuos y esto afecta negativamente la capacidad predictiva del modelo, el cual no ha sido capaz de identificar exitosamente las relaciones internas que existen entre las variables (heterogeneidad y endogeneidad). Es por tal motivo que se procederá con una estimación diferente, cuyo enfoque recae en el análisis de las unidades (y grupos) a través del tiempo y considerando las características inobservables con el fin de obtener resultados más robustos y mitigar sesgos.

10. Obtenga el modelo definitivo, prediga los valores y comente el grado de ajuste del modelo. Justifique con métricas su respuesta.

```
# Creación de variables sintéticas
limites = [-2, 15, 25, 100]
categorias = ['frió', 'templado', 'caliente']
def asignar_categoria(temp):
    for i in range(len(limites) - 1):
        if temp >= limites[i] and temp < limites[i + 1]:
            return categorias[i]
    return categorias[-1]
X['temperatura_cod'] = X['Temperatura'].apply(asignar_categoria)
X.drop('Temperatura',axis=1,inplace=True)
```

```
# Crear variables combinadas
for categoria in list(X.temperatura_cod.unique()):
    for feriado in X.Es_Semana_Feriado:
        col_name = f'Es_feriado_{categoria}'
        X[col_name] = (X['temperatura_cod'] == categoria) &
(X['Es_Semana_Feriado'] == 1)
X.drop(['temperatura_cod','Es_Semana_Feriado'],axis=1,inplace=True)
transf = list(X.iloc[:, -3: ].columns)
X[transf] = X[transf].replace({True: 1, False: 0})
```

```
# Mejor combinación con significancia y r^2
X_panel = 
X.drop(['Es_feriado_frió','Es_feriado_templado','Precio_Combustible_Región'],axis=1)
X_panel = sm.tools.tools.add_constant(X_panel)
y_panel = df_mod['Ventas_Semanales']
```

```
##### Modelo con efectos fijos
modelo_fe = PanelOLS(y_panel, X_panel, entity_effects = True)
resultados_fe = modelo_fe.fit()
resultados_fe
```

PanelOLS Estimation Summary						
Dep. Variable:	Ventas_Semanales	R-squared:	0.0177			
Estimator:	PanelOLS	R-squared (Between):	-0.2102			
No. Observations:	5920	R-squared (Within):	0.0177			
Date:	Mon, Mar 18 2024	R-squared (Overall):	0.0124			
Time:	21:34:29	Log-likelihood	-8.655e+04			
Cov. Estimator:	Unadjusted					
		F-statistic:	34.714			
Entities:	143	P-value	0.0000			
Avg Obs:	41.399	Distribution:	F(3,5774)			
Min Obs:	33.000					
Max Obs:	42.000	F-statistic (robust):	34.714			
		P-value	0.0000			
Time periods:	45	Distribution:	F(3,5774)			
Avg Obs:	131.56					
Min Obs:	17.000					
Max Obs:	143.00					
Parameter Estimates						
Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI	
const	1.643e+06	6.277e+04	26.179	0.0000	1.52e+06	1.766e+06
Indice_Precios_Consm	-1497.6	187.39	-7.9921	0.0000	-1865.0	-1130.3
Tasa_Desempleo	-4.613e+04	6069.8	-7.6007	0.0000	-5.803e+04	-3.424e+04
Es_feriado_calleente	2.328e+05	9.476e+04	2.4563	0.0141	4.699e+04	4.185e+05
F-test for Poolability: 0.9055						
P-value: 0.7808						
Distribution: F(142,5774)						
Included effects: Entity						
id: 0x7f91cb6e76a0						

```
#### Modelo con Efectos Aleatorios
modelo_re = RandomEffects(y_panel, X_panel)
resultados_re = modelo_re.fit()
resultados_re
```

RandomEffects Estimation Summary						
Dep. Variable:	Ventas_Semanales	R-squared:	0.0169			
Estimator:	RandomEffects	R-squared (Between):	0.0016			
No. Observations:	5920	R-squared (Within):	0.0172			
Date:	Mon, Mar 18 2024	R-squared (Overall):	0.0169			
Time:	21:34:30	Log-likelihood	-8.662e+04			
Cov. Estimator:	Unadjusted					
		F-statistic:	33.864			
Entities:	143	P-value	0.0000			
Avg Obs:	41.399	Distribution:	F(3,5916)			
Min Obs:	33.000					
Max Obs:	42.000	F-statistic (robust):	33.864			
		P-value	0.0000			
Time periods:	45	Distribution:	F(3,5916)			
Avg Obs:	131.56					
Min Obs:	17.000					
Max Obs:	143.00					
Parameter Estimates						
Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI	
const	1.625e+06	6.179e+04	26.301	0.0000	1.504e+06	1.746e+06
Indice_Precios_Consm	-1478.3	186.85	-7.9114	0.0000	-1844.6	-1112.0
Tasa_Desempleo	-4.303e+04	5866.7	-7.3353	0.0000	-5.453e+04	-3.153e+04
Es_feriado_calleente	8.033e+04	2.951e+04	2.7220	0.0065	2.248e+04	1.382e+05

```
### TEST DE HAUSSMANN (elección modelo)
def hausman(fe, re):
    b = fe.params
    B = re.params
    v_b = fe.cov
    v_B = re.cov
    df = b[np.abs(b) < 1e8].size
    chi2 = np.dot((b - B).T, la.inv(v_b - v_B).dot(b - B))
    pval = stats.chi2.sf(chi2, df)
    return chi2, df, pval
```

```
hausman = hausman(resultados_fe, resultados_re)
print('chi-Squared: ' + str(hausman[0]))
print('degrees of freedom: ' + str(hausman[1]))
print('p-Value:' + str(hausman[2]))
```

```
chi-Squared: 8.076393703402957
degrees of freedom: 4
p-Value: 0.08881954062116044
```

Los modelos presentados se derivan de un proceso de iteración y combinatoria entre las variables predictoras para obtener significancia estadística en todos los coeficientes y la mayor cantidad de r^2 posible (coeficiente de bondad de ajuste). En consecuencia, los modelos de panel con efectos y variables comprenden el conjunto de tres variables estadísticamente significativas ("Indice_Precios_Cons", "Tasa_Desempleo" y "Es_Feriado_Caliente"). Para la selección del modelo final se utilizó la prueba de Hausmann para evaluar la consistencia de los estimadores en efectos fijos y aleatorios bajo el siguiente esquema:

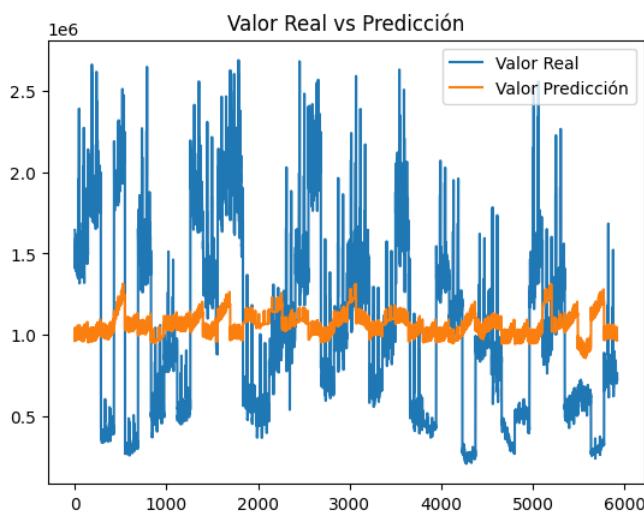
- Ho: El modelo preferido es el de efectos aleatorio
- H1: El modelo preferido es el de efectos fijos

De acuerdo con los resultados presentado en el p-valor (0.0888) se puede concluir que, con un nivel de la significancia del 5%, el modelo más adecuado y eficiente para este conjunto de datos es el modelo de efectos aleatorios.

El modelo de efectos aleatorios ha sido calculado con un total de 5920 observaciones, 143 entidades y 45 periodos con una significancia conjunta estadísticamente importante (p-valor de $F = 0.000$) y coeficientes son estadísticamente significativos al 5%. Sin embargo, al momento de analizar el r^2 se observa que la capacidad de las variables predictores para explicar la variabilidad objetivo es extremadamente baja, con un valor de 0.0169 (1,69%). Esto es indicativo de que posiblemente el conjunto de variables seleccionadas (disponibles en el dataset) no poseen la suficiente capacidad explicativa sobre las "Ventas_Semanales". En este sentido, se ha realizado otros intentos de crear variables sintéticas y realizar un proceso de feature engineering con pocos o nulos resultados en el impacto global del modelo y su ajuste.

11. Grafique a los valores predicho de modelo vs los valores reales. ¿Cómo se ven una vez graficados frente a los valores reales? Argumente su respuesta.

```
real = y_panel.reset_index().iloc[:,2:]
prediccion = prediccion = resultados_re.predict().reset_index().iloc[:,2:]
plt.plot(real, label='Valor Real')
plt.plot(prediccion, label = 'Valor Predicción')
plt.title('Valor Real vs Predicción')
plt.legend()
plt.show()
```



El bajo rendimiento del modelo y de las variables explicativas para explicar la variabilidad de la variable dependiente (medido a través del r^2) derivan en una pobre capacidad predictiva. Como se puede observar en el gráfico, las predicciones generadas del modelo identifican de una forma muy vaga el comportamiento y las fluctuaciones de los valores reales (representado en el r^2 de 1,69%).

12. Concluya sobre su modelo. Para ello, si escogió el enfoque econométrico, interprete coeficientes, por el contrario, si escogió el enfoque de machine learning, determine cuáles son las variables que tienen mayor poder explicativo sobre su variable objetivo.

Para la primera variable “Indice_Precios_Consm” se reporta un coeficiente de -14777.7, lo que nos indica una relación negativa con respecto a las “Ventas_Semanales”. Donde un aumento en el índice de precios al consumidor repercutirá de manera negativa en las ventas semanales en aproximadamente 14777.7 unidades. El valor y signo de esta variable es esperado, pues un aumento de los precios en los bienes reduce la capacidad adquisitiva de las personas, las cuales ahora no podrán comprar el mismo nivel de bienes y servicios.

La segunda variable “Tasa_Desempleo” presenta un valor de -4.26e+04, indicativo de una relación negativa con el nivel de ventas semanales en Walmart. Un incremento en la tasa de desempleo afectaría negativamente las ventas en

4.26e+04 unidades. El valor y signo de esta variable es también esperado debido a que, si la tasa de desempleo aumenta, un mayor número de personas entra en paro y se reduce la cantidad de ingresos que reciben, esto a su vez deriva en una menor capacidad adquisitiva para adquirir bienes (y servicios) como los que ofrece Walmart.

La tercera variable “Es_Feriado_Caliente” presenta un valor de 8.033e+04, lo que nos indica una relación positiva con la variable dependiente. Estos resultados nos permiten comprender que cuando existe feriado y la temperatura es elevada, las ventas en Walmart se verán incrementadas en 8.033e+04 unidades.

En este sentido, se menciona también la necesidad de transformar o directamente buscar un nuevo conjunto de variables que posean un mayor nivel explicativo sobre la variabilidad de las ventas semanales en Amazon con el fin de mejorar el ajuste general del modelo y sus capacidades predictivas. Además, se podrían explorar nuevas opciones de modelación en el campo de Machine Learning mediante algoritmos como XGBoost, Catboost, LightGBM e incluso ANN (redes neuronales) para casos más complejos, recordando que este tipo de modelos poseen diferentes argumentos para controlar su comportamiento en los procesos de aprendizaje.

Intentos previos de combinaciones, transformaciones y selección de variables

Este apartado presenta el código utilizado (y los resultados) para realizar las transformaciones e iteraciones aplicadas a las variables con el fin de obtener el mejor modelo posible considerando aspectos como el r^2 y la significancia de los coeficientes. En un primer momento, se presenta el código correspondiente a la re-codificación y creación de la mayor cantidad de nuevas variables sintéticas. En un segundo momento, se expone el código para aplicar un método de estandarización a las variables. En un tercer momento, se explica el código correspondiente a la transformación logarítmica de las variables en busca de obtener mejores resultados. Todos estos intentos han sido evaluados mediante la creación de una función que realiza de manera automática las diferentes iteraciones entre columnas del dataframe y retorna los mejores modelos tanto para efectos fijos como para efectos variables.

```
X_prueba = df_mod.drop(['Ventas_Semanales','Fecha_Venta'],axis=1)
```

```
# Recodificación variable Temperatura
limites = [-2, 15, 25, 100]
categorias = ['frío', 'templado', 'caliente']
def asignar_categoria(temp):
    for i in range(len(limites) - 1):
        if temp >= limites[i] and temp < limites[i + 1]:
            return categorias[i]
    return categorias[-1]
X_prueba['temperatura_cod'] = X_prueba['Temperatura'].apply(asignar_categoria)
X_prueba.drop('Temperatura',axis=1,inplace=True)
```

```

# Crear variables combinadas
for categoria in list(X_prueba.temperatura_cod.unique()):
    for feriado in X_prueba.Es_Semana_Feriado:
        col_name = f'Es_feriado_{categoria}'
        X_prueba[col_name] = (X_prueba['temperatura_cod'] == categoria) &
(X_prueba['Es_Semana_Feriado'] == 1)
X_prueba.drop(['temperatura_cod','Es_Semana_Feriado'],axis=1,inplace=True)

transf = list(X_prueba.iloc[:, -3:].columns)
X_prueba[transf] = X_prueba[transf].replace({True: 1, False: 0})
X_prueba = sm.tools.tools.add_constant(X_prueba)

# Evitar multicolinealidad
X_prueba_1 = X_prueba.drop(['Es_feriado_frío','Es_feriado_templado'],axis=1) # solo primera combinacion
X_prueba_2 = X_prueba.drop(['Es_feriado_caliente','Es_feriado_templado'],axis=1) # solo primera combinacion
X_prueba_3 = X_prueba.drop(['Es_feriado_caliente','Es_feriado_frío'],axis=1) # solo primera combinacion

import itertools
def encontrar_mejor_modelo(X, Y):
    mejor_modelo_fijo = None
    mejor_modelo_aleatorio = None
    mejor_r2_fijo = 0
    mejor_r2_aleatorio = 0
    mejor_resumen_fijo = None
    mejor_resumen_aleatorio = None
    for i in range(1, len(X.columns) + 1):
        for combo in itertools.combinations(X.columns, i):
            X_temp = X[list(combo)]
            X_temp = sm.tools.tools.add_constant(X_temp)

            # Modelo de efectos fijos
            modelo_fijo = PanelOLS(Y, X_temp, entity_effects=True).fit()
            if modelo_fijo.rsquared > mejor_r2_fijo and all(modelo_fijo.pvalues < 0.05):
                mejor_modelo_fijo = modelo_fijo
                mejor_r2_fijo = modelo_fijo.rsquared
                mejor_resumen_fijo = modelo_fijo

            # Modelo de efectos aleatorios
            modelo_aleatorio = RandomEffects(Y, X_temp).fit()
            if modelo_aleatorio.rsquared > mejor_r2_aleatorio and all(modelo_aleatorio.pvalues < 0.05):
                mejor_modelo_aleatorio = modelo_aleatorio
                mejor_r2_aleatorio = modelo_aleatorio.rsquared
                mejor_resumen_aleatorio = modelo_aleatorio

```

```
return mejor_modelo_fijo, mejor_resumen_fijo, mejor_modelo_aleatorio,
mejor_resumen_aleatorio
```

```
mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio =
encontrar mejor modelo(X prueba 1,y panel)
```

```
print("Mejor modelo de efectos fijos (1):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (1):")
print(resumen_aleatorio)
```

```
Mejor modelo de efectos fijos (1):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0229
Estimator: PanelOLS R-squared (Between): -3.6438
No. Observations: 5920 R-squared (Within): 0.0229
Date: Mon, Mar 18 2024 R-squared (Overall): -0.0595
Time: 21:48:23 Log-likelihood -8.654e+04
Cov. Estimator: Unadjusted F-statistic: 33.802
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(4,5773)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 33.802
P-value 0.0000
Time periods: 45 Distribution: F(4,5773)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 4.776e+05 2.2e+05 2.1709 0.0300 4.633e+04 9.09e+05
Precio_Combustible_Region 3.233e+05 5.851e+04 5.5256 0.0000 2.086e+05 4.38e+05
Indice_Precios_Consm -725.77 233.34 -3.1103 0.0019 -1183.2 -268.32
Tasa_Desempleo -5.245e+04 6161.1 -8.5124 0.0000 -6.452e+04 -4.037e+04
Es_feriado_caliente 2.181e+05 9.455e+04 2.3069 0.0211 3.277e+04 4.035e+05
=====

F-test for Poolability: 1.1247
P-value: 0.1505
Distribution: F(142,5773)

Included effects: Entity
Mejor modelo de efectos aleatorios (1):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0169
Estimator: RandomEffects R-squared (Between): 0.0016
No. Observations: 5920 R-squared (Within): 0.0172
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0169
Time: 21:48:22 Log-likelihood -8.662e+04
Cov. Estimator: Unadjusted F-statistic: 33.864
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(3,5916)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 33.864
P-value 0.0000
Time periods: 45 Distribution: F(3,5916)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 1.625e+06 6.179e+04 26.301 0.0000 1.504e+06 1.746e+06
Indice_Precios_Consm -1478.3 186.85 -7.9114 0.0000 -1844.6 -1112.0
Tasa_Desempleo -4.303e+04 5866.7 -7.3353 0.0000 -5.453e+04 -3.153e+04
Es_feriado_caliente 8.033e+04 2.951e+04 2.7220 0.0065 2.248e+04 1.382e+05
=====
```

mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio = encontrar mejor modelo(X prueba 2,y panel)

```
print("Mejor modelo de efectos fijos (2):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (2):")
print(resumen_aleatorio)
```

```

Mejor modelo de efectos fijos (2):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0220
Estimator: PanelOLS R-squared (Between): -3.7239
No. Observations: 5920 R-squared (Within): 0.0220
Date: Mon, Mar 18 2024 R-squared (Overall): -0.0618
Time: 21:48:37 Log-Likelihood -8.654e+04
Cov. Estimator: Unadjusted F-statistic: 43.263
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(3,5774)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 43.263
P-value 0.0000
Time periods: 45 Distribution: F(3,5774)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 4.677e+05 2.201e+05 2.1253 0.0336 3.629e+04 8.991e+05
Precio_Combustible_Region 3.271e+04 5.851e+04 5.5903 0.0000 2.124e+05 4.418e+05
Indice_Precios_Consm -698.10 233.12 -2.9946 0.0028 -1155.1 -241.09
Tasa_Desempleo -5.167e+04 6154.1 -8.3954 0.0000 -6.373e+04 -3.96e+04
=====

F-test for Poolability: 1.1379
P-value: 0.1281
Distribution: F(142,5774)

Included effects: Entity
Mejor modelo de efectos aleatorios (2):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0157
Estimator: RandomEffects R-squared (Between): -0.0331
No. Observations: 5920 R-squared (Within): 0.0167
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0157
Time: 21:48:34 Log-likelihood -8.662e+04
Cov. Estimator: Unadjusted F-statistic: 47.040
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(2,5917)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 47.040
P-value 0.0000
Time periods: 45 Distribution: F(2,5917)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 1.625e+06 6.183e+04 26.290 0.0000 1.504e+06 1.747e+06
Indice_Precios_Consm -1470.1 186.93 -7.8643 0.0000 -1836.5 -1103.6
Tasa_Desempleo -4.26e+04 5867.7 -7.2603 0.0000 -5.41e+04 -3.11e+04
=====
```

mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio = encontrar mejor modelo(X prueba 3,y panel)

```
print("Mejor modelo de efectos fijos (3):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (3):")
print(resumen_aleatorio)
```

```
Mejor modelo de efectos fijos (3):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0228
Estimator: PanelOLS R-squared (Between): -3.7390
No. Observations: 5920 R-squared (Within): 0.0228
Date: Mon, Mar 18 2024 R-squared (Overall): -0.0613
Time: 21:49:00 Log-likelihood -8.654e+04
Cov. Estimator: Unadjusted
                F-statistic: 33.600
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(4,5773)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 33.600
P-value 0.0000
Time periods: 45 Distribution: F(4,5773)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
      Parameter Std. Err.   T-stat   P-value   Lower CI   Upper CI
=====
const        4.875e+05 2.202e+05  2.2140  0.0269  5.584e+04 9.192e+05
Precio_Combustible_Region 3.246e+05 5.851e+04  5.5486  0.0000  2.099e+05 4.393e+05
Indice_Precios_Consm -721.69   233.32   -3.0932  0.0020  -1179.1   -264.31
Tasa_Desempleo     -5.245e+04  6163.3   -8.5102  0.0000  -6.453e+04 -4.037e+04
Es_feriado_templado -2.066e+05 9.704e+04  -2.1289  0.0333  -3.968e+05 -1.636e+04
=====

F-test for Poolability: 1.1509
P-value: 0.1086
Distribution: F(142,5773)

Included effects: Entity.
Mejor modelo de efectos aleatorios (3):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0157
Estimator: RandomEffects R-squared (Between): -0.0331
No. Observations: 5920 R-squared (Within): 0.0167
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0157
Time: 21:48:55 Log-likelihood -8.662e+04
Cov. Estimator: Unadjusted
                F-statistic: 47.040
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(2,5917)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 47.040
P-value 0.0000
Time periods: 45 Distribution: F(2,5917)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
      Parameter Std. Err.   T-stat   P-value   Lower CI   Upper CI
=====
const        1.625e+06 6.183e+04   26.290  0.0000  1.504e+06 1.747e+06
Indice_Precios_Consm -1470.1   186.93   -7.8643  0.0000  -1836.5   -1103.6
Tasa_Desempleo     -4.26e+04 5867.7   -7.2603  0.0000  -5.41e+04 -3.11e+04
=====
```

```
##### Estandarización
from sklearn.preprocessing import StandardScaler
X_prueba_est = X_prueba.copy()
columnas = ['Precio_Combustible_Region', 'Indice_Precios_Consm',
            'Tasa_Desempleo', 'Es_feriado_caliente', 'Es_feriado_frio',
            'Es_feriado_templado']
scaler = StandardScaler()
X_prueba_est[columnas] = scaler.fit_transform(X_prueba_est[columnas])
```

```
# Evitar multicolinealidad
X_prueba_1_est =
X_prueba_est.drop(['Es_feriado_frio','Es_feriado_templado'],axis=1) # solo primera combinacion
X_prueba_2_est =
X_prueba_est.drop(['Es_feriado_caliente','Es_feriado_templado'],axis=1) # solo primera combinacion
X_prueba_3_est =
X_prueba_est.drop(['Es_feriado_caliente','Es_feriado_frio'],axis=1) # solo primera combinacion
```

```
mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio =
encontrar_mejor_modelo(X_prueba_1_est,y_panel)
```

```
print("Mejor modelo de efectos fijos (1):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (1):")
print(resumen_aleatorio)
```

```

Mejor modelo de efectos fijos (1):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0229
Estimator: PanelOLS R-squared (Between): -3.6438
No. Observations: 5920 R-squared (Within): 0.0229
Date: Mon, Mar 18 2024 R-squared (Overall): -0.0595
Time: 21:49:41 Log-likelihood -8.654e+04
Cov. Estimator: Unadjusted F-statistic: 33.802
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(4,5773)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 33.802
P-value 0.0000
Time periods: 45 Distribution: F(4,5773)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 1.039e+06 7103.2 146.30 0.0000 1.025e+06 1.053e+06
Precio_Combustible_Region 1.481e+05 2.681e+04 5.5256 0.0000 9.557e+04 2.007e+05
Indice_Precios_Consm -2.832e+04 9104.9 -3.1103 0.0019 -4.617e+04 -1.047e+04
Tasa_Desempleo -6.519e+04 7658.6 -8.5124 0.0000 -8.021e+04 -5.018e+04
Es_feriado_caliente 5.26e+04 2.28e+04 2.3069 0.0211 7901.3 9.73e+04
=====

F-test for Poolability: 1.1247
P-value: 0.1505
Distribution: F(142,5773)

Included effects: Entity.
Mejor modelo de efectos aleatorios (1):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0169
Estimator: RandomEffects R-squared (Between): 0.0016
No. Observations: 5920 R-squared (Within): 0.0172
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0169
Time: 21:49:40 Log-likelihood -8.662e+04
Cov. Estimator: Unadjusted F-statistic: 33.864
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(3,5916)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 33.864
P-value 0.0000
Time periods: 45 Distribution: F(3,5916)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 1.039e+06 7113.3 146.09 0.0000 1.025e+06 1.053e+06
Indice_Precios_Consm -5.768e+04 7290.8 -7.9114 0.0000 -7.197e+04 -4.339e+04
Tasa_Desempleo -5.349e+04 7292.6 -7.3353 0.0000 -6.779e+04 -3.92e+04
Es_feriado_caliente 1.937e+04 7116.3 2.7220 0.0065 5420.1 3.332e+04
=====
```

mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio = encontrar mejor modelo(X prueba 2 est,y panel)

```

print("Mejor modelo de efectos fijos (2):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (2):")
print(resumen_aleatorio)

```

```

Mejor modelo de efectos fijos (2):
  PanelOLS Estimation Summary
=====
Dep. Variable:      Ventas_Semanales    R-squared:                  0.0220
Estimator:          PanelOLS            R-squared (Between):     -3.7239
No. Observations:   5920                R-squared (Within):       0.0220
Date:              Mon, Mar 18 2024    R-squared (Overall):      -0.0618
Time:              21:49:48             Log-likelihood:           -8.654e+04
Cov. Estimator:    Unadjusted         F-statistic:                 43.263
Entities:           143                P-value:                   0.0000
Avg Obs:            41.399             Distribution:               F(3,5774)
Min Obs:             33.000
Max Obs:             42.000             F-statistic (robust):    43.263
                                         P-value:                   0.0000
                                         Distribution:               F(3,5774)
Time periods:        45                Avg Obs:                  131.56
                                         Min Obs:                  17.000
                                         Max Obs:                  143.00

Parameter Estimates
=====
      Parameter  Std. Err.    T-stat   P-value   Lower CI   Upper CI
=====
const      1.039e+06    7105.9    146.24  0.0000  1.025e+06  1.053e+06
Precio_Combustible_Region 1.499e+05  2.681e+04   5.5903  0.0000  9.73e+04  2.024e+05
Indice_Precios_Consm -2.724e+04  9096.3    -2.9946  0.0028 -4.507e+04 -9407.3
Tasa_Desempleo -6.422e+04  7649.9    -8.3954  0.0000 -7.922e+04 -4.923e+04
=====

F-test for Poolability: 1.1379
P-value: 0.1281
Distribution: F(142,5774)

Included effects: Entity
Mejor modelo de efectos aleatorios (2):
  RandomEffects Estimation Summary
=====
Dep. Variable:      Ventas_Semanales    R-squared:                  0.0157
Estimator:          RandomEffects       R-squared (Between):     -0.0331
No. Observations:   5920                R-squared (Within):       0.0167
Date:              Mon, Mar 18 2024    R-squared (Overall):      0.0157
Time:              21:49:46             Log-likelihood:           -8.662e+04
Cov. Estimator:    Unadjusted         F-statistic:                 47.040
Entities:           143                P-value:                   0.0000
Avg Obs:            41.399             Distribution:               F(2,5917)
Min Obs:             33.000
Max Obs:             42.000             F-statistic (robust):    47.040
                                         P-value:                   0.0000
                                         Distribution:               F(2,5917)
Time periods:        45                Avg Obs:                  131.56
                                         Min Obs:                  17.000
                                         Max Obs:                  143.00

Parameter Estimates
=====
      Parameter  Std. Err.    T-stat   P-value   Lower CI   Upper CI
=====
const      1.039e+06    7117.2    146.01  0.0000  1.025e+06  1.053e+06
Indice_Precios_Consm -5.736e+04  7293.8    -7.8643  0.0000 -7.166e+04 -4.306e+04
Tasa_Desempleo -5.296e+04  7293.8    -7.2603  0.0000 -6.725e+04 -3.866e+04
=====
```

mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio = encontrar_mejor_modelo(X_prueba_3.est,y_panel)

```

print("Mejor modelo de efectos fijos (3):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (3):")
print(resumen_aleatorio)

```

```

Mejor modelo de efectos fijos (3):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0228
Estimator: PanelOLS R-squared (Between): -3.7390
No. Observations: 5920 R-squared (Within): 0.0228
Date: Mon, Mar 18 2024 R-squared (Overall): -0.0613
Time: 21:50:03 Log-likelihood -8.654e+04
Cov. Estimator: Unadjusted
                F-statistic: 33.600
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(4,5773)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 33.600
Time periods: 45 P-value 0.0000
Avg Obs: 131.56 Distribution: F(4,5773)
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
      Parameter Std. Err. T-stat P-value Lower CI Upper CI
-----
const 1.039e+06 7103.7 146.29 0.0000 1.025e+06 1.053e+06
Precio_Combustible_Region 1.487e+05 2.68e+04 5.5486 0.0000 9.617e+04 2.013e+05
Indice_Precios_Consm -2.816e+04 9103.8 -3.0932 0.0020 -4.601e+04 -1.031e+04
Tasa_Desempleo -6.52e+04 7661.3 -8.5102 0.0000 -8.022e+04 -5.018e+04
Es_feriado_templado -1.671e+04 7850.1 -2.1289 0.0333 -3.21e+04 -1323.3
=====

F-test for Poolability: 1.1509
P-value: 0.1086
Distribution: F(142,5773)

Included effects: Entity
Mejor modelo de efectos aleatorios (3):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0157
Estimator: RandomEffects R-squared (Between): -0.0331
No. Observations: 5920 R-squared (Within): 0.0167
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0157
Time: 21:49:57 Log-likelihood -8.662e+04
Cov. Estimator: Unadjusted
                F-statistic: 47.040
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(2,5917)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 47.040
Time periods: 45 P-value 0.0000
Avg Obs: 131.56 Distribution: F(2,5917)
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
      Parameter Std. Err. T-stat P-value Lower CI Upper CI
-----
const 1.039e+06 7117.2 146.01 0.0000 1.025e+06 1.053e+06
Indice_Precios_Consm -5.736e+04 7293.8 -7.8643 0.0000 -7.166e+04 -4.306e+04
Tasa_Desempleo -5.296e+04 7293.8 -7.2603 0.0000 -6.725e+04 -3.866e+04
=====
```

```

##### Logaritmos
X_prueba_log = X_prueba.copy()
cuant = X_prueba.select_dtypes(include=np.number).iloc[:,1:4]
X_prueba_log[cuant.columns] = cuant.apply(lambda x: np.log(x) if (x > 0).all() else
x)
```

```

# Evitar multicolinealidad
X_prueba_1_log =
X_prueba_log.drop(['Es_feriado_frio','Es_feriado_templado'],axis=1) # solo primera
combinacion
X_prueba_2_log =
X_prueba_log.drop(['Es_feriado_caliente','Es_feriado_templado'],axis=1) # solo
primera combinacion
```

```
X_prueba_3_log
X_prueba_log.drop(['Es_feriado_caliente','Es_feriado_frio'],axis=1) # solo primera combinacion
```

```
mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio =
encontrar_mejor_modelo(X_prueba_1_log,y_panel)
```

```
print("Mejor modelo de efectos fijos (1):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (1):")
print(resumen_aleatorio)
```

```
Mejor modelo de efectos fijos (1):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0148
Estimator: PanelOLS R-squared (Between): -0.1863
No. Observations: 5920 R-squared (Within): 0.0148
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0101
Time: 21:50:26 Log-likelihood: -8.656e+04
Cov. Estimator: Unadjusted
                F-statistic: 28.946
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(3,5774)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 28.946
P-value 0.0000
Time periods: 45 Distribution: F(3,5774)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 2.806e+06 1.976e+05 14.204 0.0000 2.419e+06 3.194e+06
Indice_Precios_Consm -2.309e+05 3.153e+04 -7.3239 0.0000 -2.927e+05 -1.691e+05
Tasa_Desempleo -2.927e+05 4.456e+04 -6.5681 0.0000 -3.801e+05 -2.053e+05
Es_feriado_caliente 2.248e+05 9.489e+04 2.3686 0.0179 3.873e+04 4.108e+05
=====

F-test for Poolability: 0.8972
P-value: 0.8020
Distribution: F(142,5774)

Included effects: Entity
Mejor modelo de efectos aleatorios (1):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0142
Estimator: RandomEffects R-squared (Between): 0.0061
No. Observations: 5920 R-squared (Within): 0.0144
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0142
Time: 21:50:26 Log-likelihood: -8.663e+04
Cov. Estimator: Unadjusted
                F-statistic: 28.357
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(3,5916)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 28.357
P-value 0.0000
Time periods: 45 Distribution: F(3,5916)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
=====
const 2.758e+06 1.966e+05 14.031 0.0000 2.373e+06 3.143e+06
Indice_Precios_Consm -2.281e+05 3.143e+04 -7.2549 0.0000 -2.897e+05 -1.664e+05
Tasa_Desempleo -2.717e+05 4.301e+04 -6.3158 0.0000 -3.56e+05 -1.873e+05
Es_feriado_caliente 7.93e+04 2.955e+04 2.6834 0.0073 2.137e+04 1.372e+05
=====
```

mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio = encontrar mejor modelo(X prueba 2 log,y panel)

```
print("Mejor modelo de efectos fijos (2):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (2):")
print(resumen_aleatorio)
```

```
Mejor modelo de efectos fijos (2):
PanelOLS Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0139
Estimator: PanelOLS R-squared (Between): -0.0311
No. Observations: 5920 R-squared (Within): 0.0139
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0129
Time: 21:50:36 Log-likelihood -8.656e+04
Cov. Estimator: Unadjusted F-statistic: 40.582
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(2,5775)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 40.582
P-value 0.0000
Time periods: 45 Distribution: F(2,5775)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
-----
const 2.792e+06 1.976e+05 14.135 0.0000 2.405e+06 3.18e+06
Indice_Precios_Consm -2.281e+05 3.152e+04 -7.2354 0.0000 -2.898e+05 -1.663e+05
Tasa_Desempleo -2.863e+05 4.45e+04 -6.4333 0.0000 -3.735e+05 -1.99e+05
=====

F-test for Poolability: 0.9076
P-value: 0.7753
Distribution: F(142,5775)

Included effects: Entity
Mejor modelo de efectos aleatorios (2):
RandomEffects Estimation Summary
=====
Dep. Variable: Ventas_Semanales R-squared: 0.0130
Estimator: RandomEffects R-squared (Between): -0.0285
No. Observations: 5920 R-squared (Within): 0.0138
Date: Mon, Mar 18 2024 R-squared (Overall): 0.0130
Time: 21:50:36 Log-likelihood -8.663e+04
Cov. Estimator: Unadjusted F-statistic: 38.895
Entities: 143 P-value 0.0000
Avg Obs: 41.399 Distribution: F(2,5917)
Min Obs: 33.000
Max Obs: 42.000 F-statistic (robust): 38.895
P-value 0.0000
Time periods: 45 Distribution: F(2,5917)
Avg Obs: 131.56
Min Obs: 17.000
Max Obs: 143.00

Parameter Estimates
=====
Parameter Std. Err. T-stat P-value Lower CI Upper CI
-----
const 2.75e+06 1.966e+05 13.986 0.0000 2.365e+06 3.136e+06
Indice_Precios_Consm -2.268e+05 3.145e+04 -7.2133 0.0000 -2.885e+05 -1.652e+05
Tasa_Desempleo -2.686e+05 4.302e+04 -6.2430 0.0000 -3.529e+05 -1.842e+05
=====
```

mejor_modelo_fijo, resumen_fijo, mejor_modelo_aleatorio, resumen_aleatorio = encontrar mejor modelo(X prueba 3 log,y panel)

```
print("Mejor modelo de efectos fijos (3):")
print(resumen_fijo)
print("Mejor modelo de efectos aleatorios (3):")
```

`print(resumen_aleatorio)`

Mejor modelo de efectos fijos (3): PanelOLS Estimation Summary								
Dep. Variable:	Ventas_Semanales	R-squared:	0.0146					
Estimator:	PanelOLS	R-squared (Between):	-0.0526					
No. Observations:	5920	R-squared (Within):	0.0146					
Date:	Mon, Mar 18 2024	R-squared (Overall):	0.0132					
Time:	21:51:09	Log-likelihood	-8.656e+04					
Cov. Estimator:	Unadjusted	F-statistic:	28.604					
Entities:	143	P-value	0.0000					
Avg Obs:	41.399	Distribution:	F(3,5774)					
Min Obs:	33.000							
Max Obs:	42.000	F-statistic (robust):	28.604					
Time periods:	45	P-value	0.0000					
Avg Obs:	131.56	Distribution:	F(3,5774)					
Min Obs:	17.000							
Max Obs:	143.00							
Parameter Estimates								
Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI			
const	2.82e+06	1.979e+05	14.249	0.0000	2.432e+06	3.208e+06		
Indice_Precios_Consm	-2.307e+05	3.153e+04	-7.3169	0.0000	-2.926e+05	-1.689e+05		
Tasa_Desempleo	-2.924e+05	4.458e+04	-6.5600	0.0000	-3.798e+05	-2.05e+05		
Es_feriado_templado	-2.088e+05	9.74e+04	-2.1440	0.0321	-3.998e+05	-1.788e+04		
F-test for Poolability: 0.9233								
P-value: 0.7314								
Distribution: F(142,5774)								
Included effects: Entity								
Mejor modelo de efectos aleatorios (3): RandomEffects Estimation Summary								
Dep. Variable:	Ventas_Semanales	R-squared:	0.0130					
Estimator:	RandomEffects	R-squared (Between):	-0.0285					
No. Observations:	5920	R-squared (Within):	0.0138					
Date:	Mon, Mar 18 2024	R-squared (Overall):	0.0130					
Time:	21:51:07	Log-likelihood	-8.663e+04					
Cov. Estimator:	Unadjusted	F-statistic:	38.895					
Entities:	143	P-value	0.0000					
Avg Obs:	41.399	Distribution:	F(2,5917)					
Min Obs:	33.000							
Max Obs:	42.000	F-statistic (robust):	38.895					
Time periods:	45	P-value	0.0000					
Avg Obs:	131.56	Distribution:	F(2,5917)					
Min Obs:	17.000							
Max Obs:	143.00							
Parameter Estimates								
Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI			
const	2.75e+06	1.966e+05	13.986	0.0000	2.365e+06	3.136e+06		
Indice_Precios_Consm	-2.268e+05	3.145e+04	-7.2133	0.0000	-2.885e+05	-1.652e+05		
Tasa_Desempleo	-2.686e+05	4.302e+04	-6.2430	0.0000	-3.529e+05	-1.842e+05		