

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

```
In [1]: import pandas as pd
import numpy as np
from faker import Faker
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

In [2]: # ESTA DATA FUE BAJADA DE http://sinagap.mag.gob.ec/sina/PaginasCGSIN/Rep_Precios_Insumos_Agropecuarios.aspx
df = pd.read_csv('Data.csv', delimiter=',', header=0)

In [3]: # LOS PRECIOS ESTAN FORMATO EJEMPLO "2,50" AL IMPORTAR CON PANDAS DEBEMOS REEMPLAZAR ESE POR . Y LUEGO CONVERTIR EL CAMPO A NUMERICO
df['PRECIO1'] = df['PRECIO1'].str.replace(',', '.', regex=False)
df['PRECIO1'] = pd.to_numeric(df['PRECIO1'], errors='coerce')

In [4]: #HAGO LISTA DE 6 COLUMNAS TIPO MATRIZ PARA INDICAR 0 O 1 APLICA LA OTRA LISTA SI LE ASIGNO SU DESCUENTO
Categorias = [
    df['DESCRIPCION_INSUMO'] == 'FERTILIZANTES',
    df['DESCRIPCION_INSUMO'] == 'FUNGICIDAS',
    df['DESCRIPCION_INSUMO'] == 'HERBICIDAS',
    df['DESCRIPCION_INSUMO'] == 'INSECTICIDAS',
    df['DESCRIPCION_INSUMO'] == 'VETERINARIO'

]
Descuentos = [0.40, 0.35, 0.15, 0.20, 0.43]
#PROCEDEMOS LINEA A LINEA SI LA CONDICION SE CUMPLE ENTRE LAS 2 LISTAS CASO CONTRARIO SERA 0.01
df['COSTO'] = df['PRECIO1'] / (1 + np.select(Categorias, Descuentos, default=0.01))

#AGREGAMOS COLUMNA DE UTILIDAD
df['UTILIDAD'] = df['PRECIO1'] - df['COSTO']

In [5]: # CREAMOS UN DICCIONARIO DE LAS PROVINCIAS CON SU RESPECTIVA REGION
Rergiones = {
    'EL ORO': 'COSTA',
    'ESMERALDAS': 'COSTA',
    'GUAYAS': 'COSTA',
    'LOS RIOS': 'COSTA',
    'MANABÍ': 'COSTA',
    'SANTA ELENA': 'COSTA',
    'GALAPAGOS': 'INSULAR',
    'MORONA SANTIAGO': 'ORIENTE',
    'NAPO': 'ORIENTE',
    'ORELLANA': 'ORIENTE',
    'PASTAZA': 'ORIENTE',
    'SUCUMBIO': 'ORIENTE',
    'ZAMORA CHINCHIPE': 'ORIENTE',
    'AZUAY': 'SIERRA',
    'BOLIVAR': 'SIERRA',
    'CAÑAR': 'SIERRA',
    'CARCHI': 'SIERRA',
    'CHIMBORAZO': 'SIERRA',
    'COTOPAXI': 'SIERRA',
    'IMBABURA': 'SIERRA',
    'LOJA': 'SIERRA',
    'PICHINCHA': 'SIERRA',
    'SANTO DOMINGO DE LOS TSÁCHILAS': 'SIERRA',
    'TUNGURAHUA': 'SIERRA'

}
#PROCEDO A BUSCAR EN LA COLUMNA PROVINCIA Y SI COINCIDE REGRESA LA REGION Q CORRESPONDA
df['REGION'] = df['PROVINCIA'].map(Rergiones).fillna('')

In [6]: # SE CREA NOMBRES ALEATORIOS PARA LOS ZOOTECNISTAS POR REGION
faker = Faker()
ZooTecnista_Costa = 'Dr. ' + faker.last_name() + ' ' + faker.name()
ZooTecnista_Sierra = 'Dr. ' + faker.last_name() + ' ' + faker.name()
ZooTecnista_Oriente = 'Dr. ' + faker.last_name() + ' ' + faker.name()

In [18]: #SE CREA UNA FUNCION QUE RECIBE UN VALOR EL CUAL SEGUN LA REGION VA DEVOLVER LA VARIABLE DEL NOMBRE QUE CORRESPONDA
def Asignar_ZooTecnista(x):
    if x in 'COSTA':
        return ZooTecnista_Costa
    if x in 'SIERRA':
        return ZooTecnista_Sierra
    if x in 'ORIENTE':
        return ZooTecnista_Oriente
    else:
        return ZooTecnista_Costa

df['ZOOTECNISTA'] = df['REGION'].apply(Asignar_ZooTecnista)
#SE GRABA LA DATA COMPLETA
df.to_csv('Data_Completa.csv', sep=',', index=False)
print(df)
```

	FECHA_TOMA	PROVINCIA	\
0	2021-03-05	ESMERALDAS	
1	2021-03-05	ESMERALDAS	
2	2021-03-05	ESMERALDAS	
3	2021-03-05	ESMERALDAS	
4	2021-03-05	ESMERALDAS	
...	
83616	2025-06-05	IMBABURA	
83617	2025-06-05	IMBABURA	
83618	2025-06-05	IMBABURA	
83619	2025-06-05	IMBABURA	
83620	2025-06-05	IMBABURA	

	NOMBRE_PRODUCTO	CONCENTRACION	\
0	UREA	46-0-0	
1	MURIATO DE POTASIO	0-0-60	
2	FOSFATO DIAMONICO	18-46-0	
3	GLIFOSATO	48%	
4	PENDIMETALINA	40%	
...	
83616	CIPERMETRINA	20%	
83617	GLIFOSATO	48%	
83618	8-20-20	8-20-20	
83619	10-30-10	10-30-10	
83620	IVERMECTINA	1%	

	DESCRIPCION_INSUMO	NOMBRE_COMERCIAL	CANTIDAD	UME	\
0	FERTILIZANTES	UREA	50	Kilogramo	
1	FERTILIZANTES	MURIATO DE POTASIO	50	Kilogramo	
2	FERTILIZANTES	FOSFATO DIAMONICO	50	Kilogramo	
3	HERBICIDAS	GLIFOPAC	1	Litro	
4	HERBICIDAS	GRAMILAQ 40 EC	1	Litro	
...	
83616	INSECTICIDAS	CIPERTOX 20	1	Litro	
83617	HERBICIDAS	RONDO	1	Litro	
83618	FERTILIZANTES	8-20-20	50	Kilogramo	
83619	FERTILIZANTES	10-30-10	50	Kilogramo	
83620	VETERINARIO	ALLMECTIN	50	Centimetro cúbico	

	PRECIO1	COSTO	UTILIDAD	REGION	\
0	23.0	16.428571	6.571429	COSTA	
1	22.5	16.071429	6.428571	COSTA	
2	32.0	22.857143	9.142857	COSTA	
3	4.3	3.739130	0.560870	COSTA	
4	9.4	8.173913	1.226087	COSTA	
...	
83616	11.8	9.833333	1.966667	SIERRA	
83617	6.2	5.391304	0.808696	SIERRA	
83618	37.3	26.642857	10.657143	SIERRA	
83619	37.5	26.785714	10.714286	SIERRA	
83620	5.5	3.846154	1.653846	SIERRA	

	ZOOTECNISTA	TRIMESTRE
0	Dr. Harris Dr. Seth Mccarthy Jr.	2021Q1
1	Dr. Harris Dr. Seth Mccarthy Jr.	2021Q1
2	Dr. Harris Dr. Seth Mccarthy Jr.	2021Q1
3	Dr. Harris Dr. Seth Mccarthy Jr.	2021Q1
4	Dr. Harris Dr. Seth Mccarthy Jr.	2021Q1
...
83616	Dr. Alvarado Megan Barr	2025Q2
83617	Dr. Alvarado Megan Barr	2025Q2
83618	Dr. Alvarado Megan Barr	2025Q2
83619	Dr. Alvarado Megan Barr	2025Q2
83620	Dr. Alvarado Megan Barr	2025Q2

[83621 rows x 14 columns]

```
In [8]: #CONVERTIMOS EL CAMPO TIPO OBJECT A DATE
df['FECHA_TOMA'] = pd.to_datetime(df['FECHA_TOMA'], errors='coerce')

In [9]: # SE PROCEDE CON LA FILTRACION DE LA DATA EN UN NUEVO DATAFRAME Y POR PERIODOS ESTE CASO MENSUAL
df_DataGrafico1 = df[df['NOMBRE_COMERCIAL'] == 'UREA'].copy()
df_DataGrafico1['PERIODO'] = df_DataGrafico1['FECHA_TOMA'].dt.to_period('M')

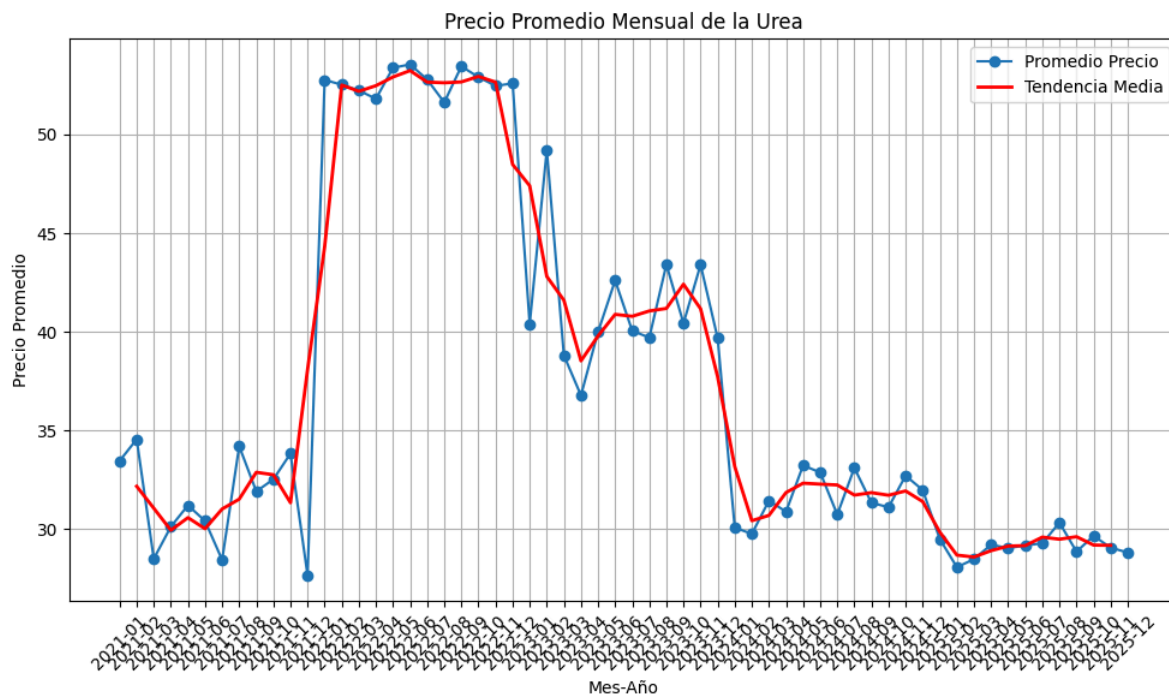
In [10]: promedios = df_DataGrafico1.groupby('PERIODO')['PRECIO1'].mean().reset_index()
promedios['PRECIO_MEDIA'] = promedios['PRECIO1'].rolling(window=3, center=True).mean()

In [11]: promedios['PERIODO'] = promedios['PERIODO'].astype(str)

In [12]: plt.figure(figsize=(10, 6))

plt.plot(promedios['PERIODO'], promedios['PRECIO1'], label='Promedio Precio', marker='o', linestyle='-')
plt.plot(promedios['PERIODO'], promedios['PRECIO_MEDIA'], label='Tendencia Media', color='red', linewidth=2)
plt.xticks(rotation=45)
plt.title(f'Precio Promedio Mensual de la Urea')
plt.xlabel('Mes-Año')
plt.ylabel('Precio Promedio')

plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



```
In [13]: df_DataGrafico2 = df[(df['NOMBRE_COMERCIAL'] == 'UREA') & (df['FECHA_TOMA'].dt.year == 2024)].copy()
Promedios_Region = df_DataGrafico2.groupby('REGION')['PRECIO1'].mean()
Promedios1 = df_DataGrafico1.groupby('REGION')[['PRECIO1', 'COSTO']].mean().reset_index()
```

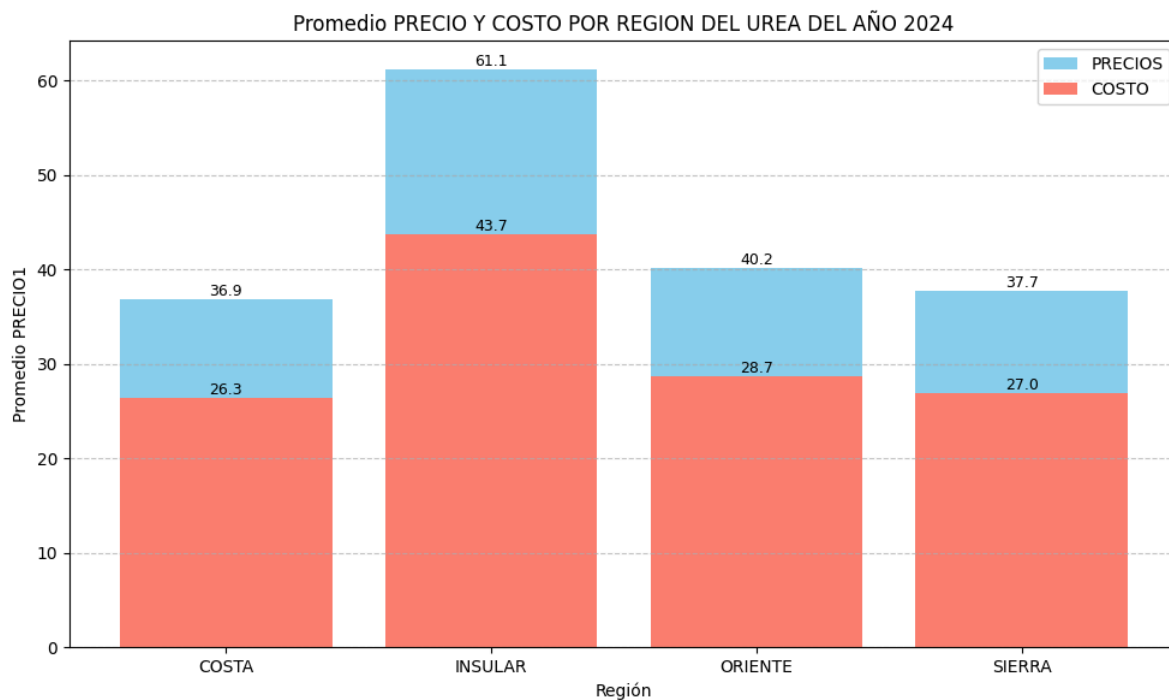
```
In [14]: plt.figure(figsize=(10, 6))
bars1 = plt.bar(Promedios1['REGION'], Promedios1['PRECIO1'], color='skyblue', label='PRECIOS')
bars2 = plt.bar(Promedios1['REGION'], Promedios1['COSTO'], color='salmon', label='COSTO')

plt.xlabel('Región')
plt.ylabel('Promedio PRECIO1')
plt.title('Promedio PRECIO Y COSTO POR REGION DEL UREA DEL AÑO 2024')

plt.grid(axis='y', linestyle='--', alpha=0.7)

# PONER ETIQUETAS VALORES
for bars in [bars1, bars2]:
    for bar in bars:
        height = bar.get_height()
        plt.text(
            bar.get_x() + bar.get_width() / 2,
            height,
            f'{height:.1f}',
            ha='center', va='bottom', fontsize=9
        )

plt.tight_layout()
plt.legend()
plt.show()
```



In []:

```
In [15]: # EN LA DATA INICIAL PONEMOS UN CAMPO MAS PARA EL TRIMESTRE
df['TRIMESTRE'] = df['FECHA_TOMA'].dt.to_period('Q')
# AGRUPAMOS POR TRIMESTRE y CATEGORIA EN UNA NUEVA DATA
df_Trimestre = df.groupby(['TRIMESTRE', 'DESCRIPCION_INSUMO'])['PRECIO1'].mean().reset_index()
df_Trimestre['TRIMESTRE_RANGO'] = df_Trimestre['TRIMESTRE'].dt.to_timestamp()
```

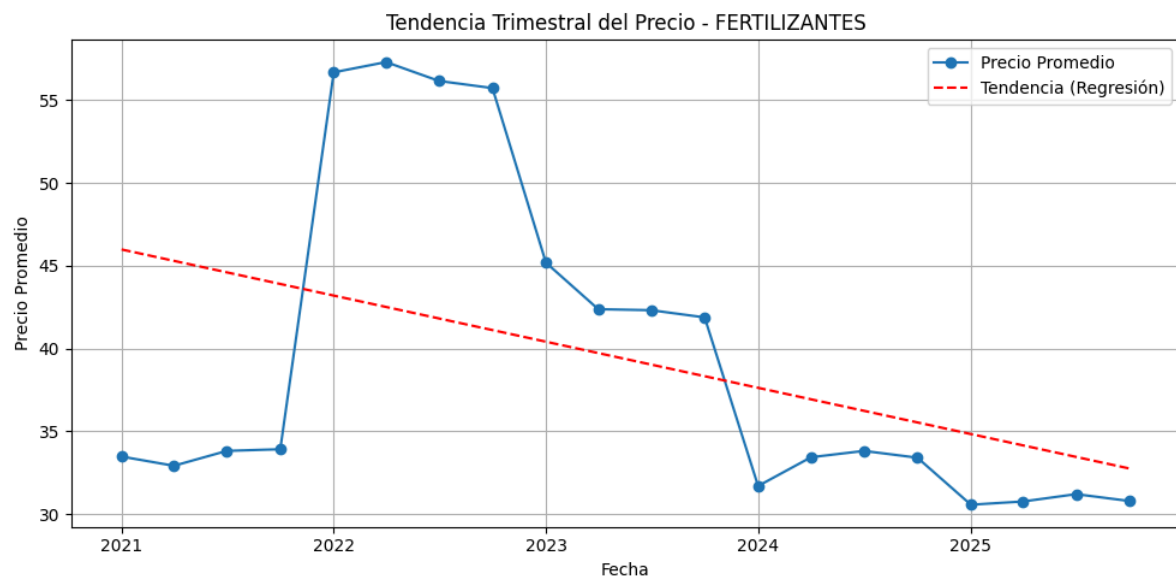
```
In [16]: #CREAMOS UN NUEVO DATAFRAME FILTRANDO SOLO LA CATEGORIA QUE SE REQUIERE
CATEGORIA_FILTRAR = 'FERTILIZANTES'
df_cat = df_Trimestre[df_Trimestre['DESCRIPCION_INSUMO'] == CATEGORIA_FILTRAR].copy()

# SE HACE CONVERSION DE LAS FECHAS A NUMEROS ORDINARIOS PARA USARLOS EN LA REGRESION
X = df_cat['TRIMESTRE_RANGO'].map(pd.Timestamp.toordinal).values.reshape(-1, 1)
y = df_cat['PRECIO1'].values

# MODELO REGRESION LINEAL
modelo = LinearRegression()
modelo.fit(X, y)

# PREDICCION DE VALORES
y_pred = modelo.predict(X)
```

```
In [17]: plt.figure(figsize=(10, 5))
plt.plot(df_cat['TRIMESTRE_RANGO'], y, marker='o', label='Precio Promedio')
plt.plot(df_cat['TRIMESTRE_RANGO'], y_pred, color='red', linestyle='--', label='Tendencia (Regresión)')
plt.title(f'Tendencia Trimestral del Precio - {CATEGORIA_FILTRAR}')
plt.xlabel('Fecha')
plt.ylabel('Precio Promedio')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



In []: