Instituto Tecnológico y de Estudios Superiores de Monterrey

A8-Series de tiempo no estacionarias. Tendencia

Profesores:

Ivan Mauricio Amaya Contreras

Blanca Rosa Ruiz Hernandez

Antonio Carlos Bento

Frumencio Olivas Alvarez

Hugo Terashima Marín

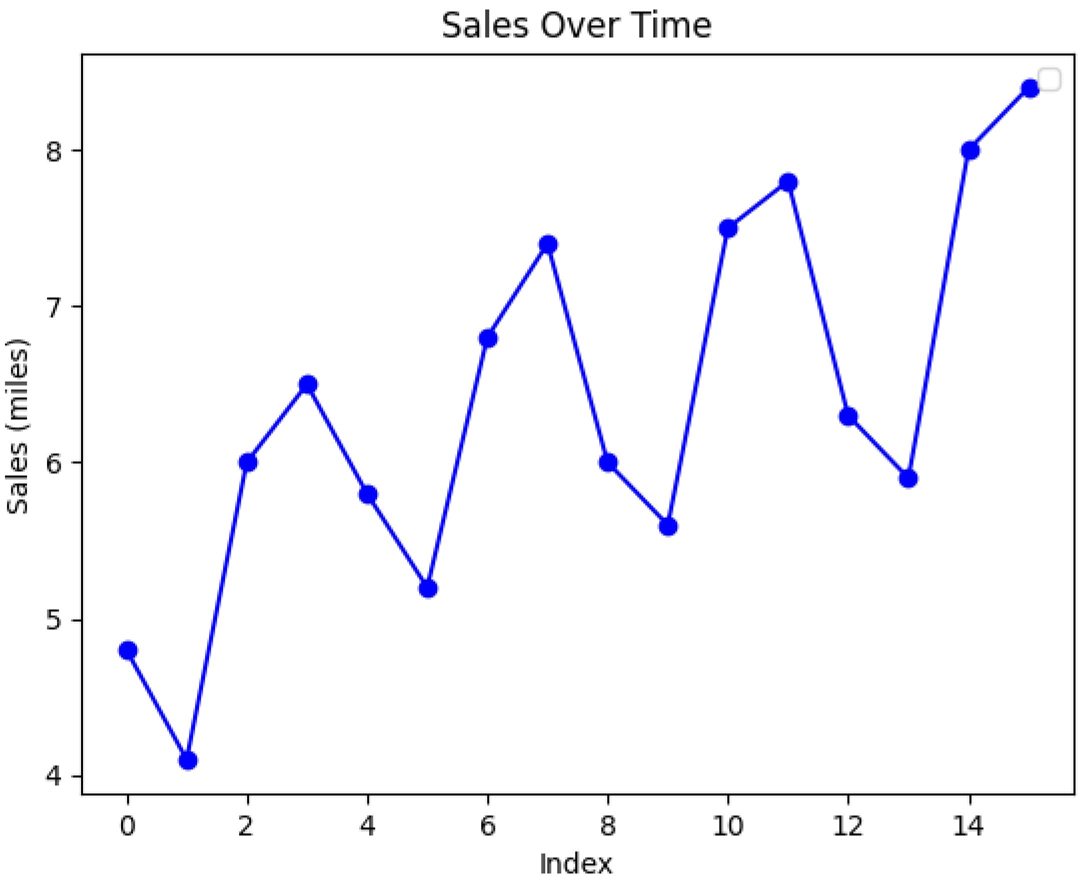
Julian Lawrence Gil Soares - A00832272

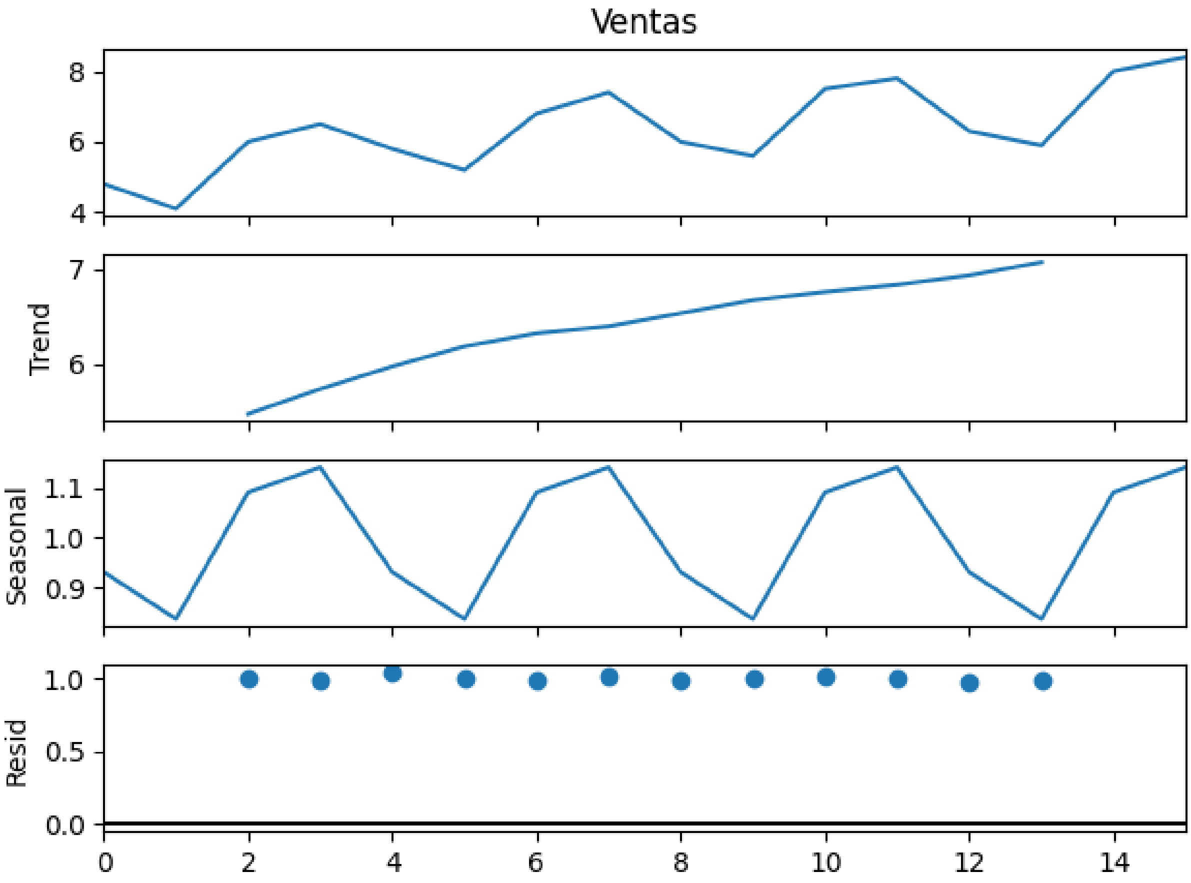
17 de Noviembre de 2023

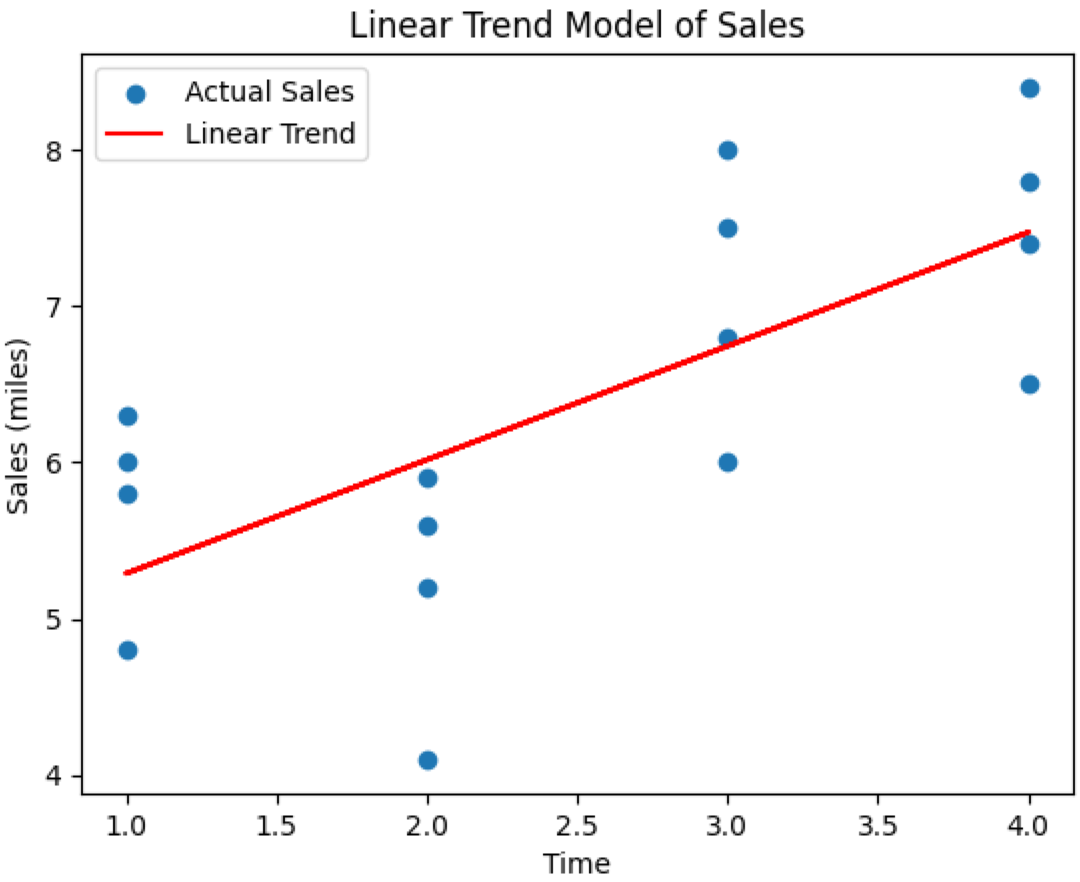
In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm

data = {
    'Año': [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4],
    'Trimestre': [1, 2, 3, 4] * 4,
    'Ventas': [4.8, 4.1, 6.0, 6.5, 5.8, 5.2, 6.8, 7.4, 6.0, 5.6, 7.5, 7.8, 6.3, 5.9
}

df = pd.DataFrame(data)

plt.plot(df.index, df['Ventas'], marker='o', linestyle='-', color='b')
plt.title('Sales Over Time')
plt.xlabel('Index')
plt.ylabel('Sales (miles)')
plt.legend()
plt.show()

result = sm.tsa.seasonal_decompose(df['Ventas'], model='multiplicative', period=4)
result.plot()
plt.show()
df['Time'] = df.groupby('Año').cumcount() + 1

X = sm.add_constant(df['Time'])
model = sm.OLS(df['Ventas'], X).fit()

plt.scatter(df['Time'], df['Ventas'], label='Actual Sales')
plt.plot(df['Time'], model.predict(X), label='Linear Trend', color='red')
plt.title('Linear Trend Model of Sales')
plt.xlabel('Time')
plt.ylabel('Sales (miles)')
plt.legend()
plt.show()

print(model.summary())

next_year_time = df['Time'].max() + 1
next_year_sales = model.predict([1, next_year_time])[0]
print(f'Predicted Sales for the Next Year: {next_year_sales}')

plt.plot(df['Time'], df['Ventas'], label='Actual Sales')
plt.scatter(next_year_time, next_year_sales, color='red', label='Predicted Sales')
plt.title('Actual Sales and Predictions Over Time')
plt.xlabel('Time')
plt.ylabel('Sales (miles)')
plt.legend()
plt.show()
```

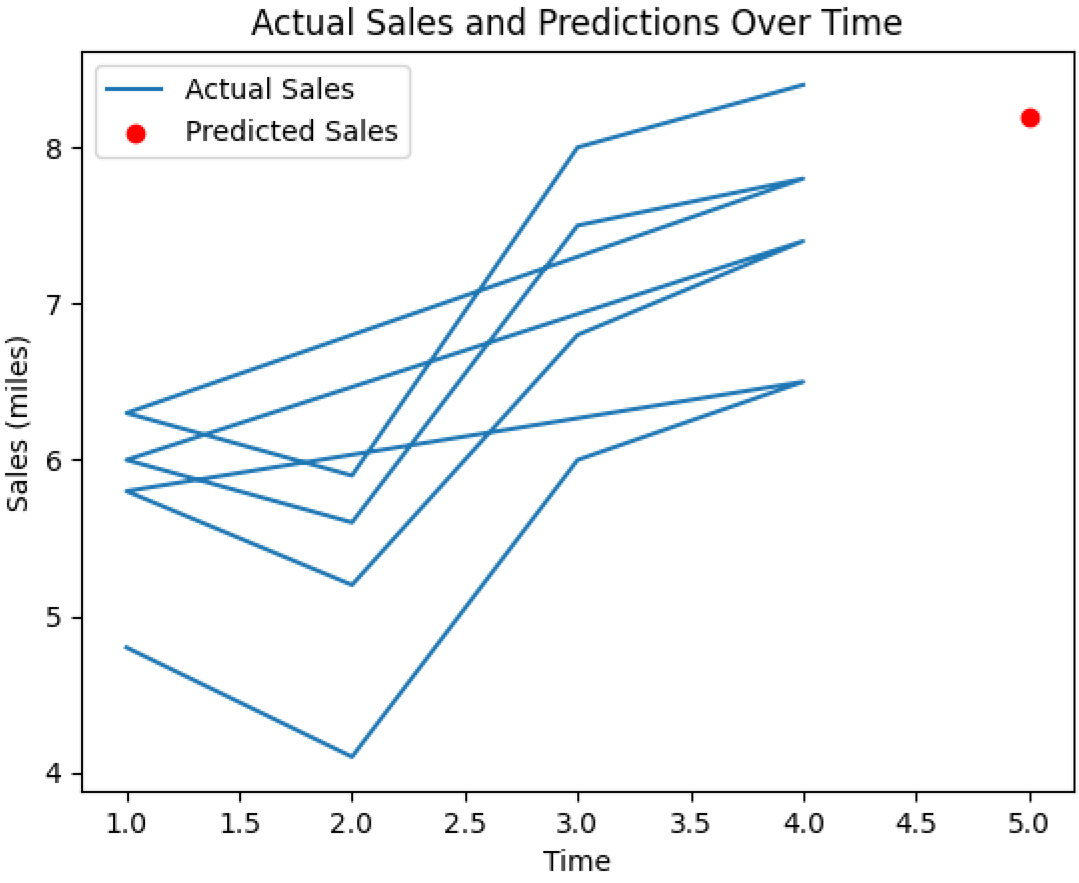
```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that
artists whose label start with an underscore are ignored when legend() is called wit
h no argument.
```


Sales Over Time
Sales (miles)
Index
4
5
6
7
8
0
2
4
6
8
10
12
14


Ventas
4
6
8
Trend
6
7
Seasonal
0.9
1.0
1.1
Resid
0.0
0.5
1.0
0
2
4
6
8
10
12
14



```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=16
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                 Ventas   R-squared:                       0.486
Model:                            OLS   Adj. R-squared:                  0.450
Method:                 Least Squares   F-statistic:                     13.26
Date:                Fri, 17 Nov 2023   Prob (F-statistic):            0.00267
Time:                        06:20:22   Log-Likelihood:                -19.835
No. Observations:                  16   AIC:                             43.67
Df Residuals:                      14   BIC:                             45.21
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.5625      0.547      8.338      0.000       3.389       5.736
Time           0.7275      0.200      3.641      0.003       0.299       1.156
==============================================================================
Omnibus:                        0.927   Durbin-Watson:                   1.375
Prob(Omnibus):                  0.629   Jarque-Bera (JB):                0.697
Skew:                          -0.467   Prob(JB):                        0.706
Kurtosis:                       2.583   Cond. No.                         7.47
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
Predicted Sales for the Next Year: 8.199999999999996
```



In [ ]:

```
import pandas as pd

data = {
    'Trimestre': [1, 2, 3, 4],
    '1': [1690, 940, 2625, 2500],
    '2': [1800, 900, 2900, 2360],
    '3': [1850, 1100, 2930, 2615]
}

df = pd.DataFrame(data)

df['Promedio_Movil'] = df[['1', '2', '3']].mean(axis=1)

df['Promedio_Movil_Centrado'] = df['Promedio_Movil'].rolling(window=2, center=True)

print(df[['Trimestre', '1', '2', '3', 'Promedio_Movil', 'Promedio_Movil_Centrado']]

df['Promedio_Trimestre'] = df[['1', '2', '3']].mean(axis=1)

df['Indice_Estacional'] = df['1'] / df['Promedio_Trimestre']
df['Indice_Estacional_Acumulado'] = df['Indice_Estacional'].cumsum()

max_estacional_trimestre = df.loc[df['Indice_Estacional'].idxmax(), 'Trimestre']

print(f'El mayor índice estacional se obtiene en el trimestre {max_estacional_trime
```

```
   Trimestre     1     2     3  Promedio_Movil  Promedio_Movil_Centrado
0          1  1690  1800  1850     1780.000000                      NaN
1          2   940   900  1100      980.000000              1380.000000
2          3  2625  2900  2930     2818.333333              1899.166667
3          4  2500  2360  2615     2491.666667              2655.000000
El mayor índice estacional se obtiene en el trimestre 4.
```