modelacion_mexico

November 14, 2025

[11]:

```
# Limpieza de datos

import pandas as pd

df = pd.read_csv('../data/datos_mexico.csv')

df=df.drop(['_id','RENGLON','ENTIDAD'],axis=1)

# Combinacion de columnas

df['Poblacion de jovenes (3-14)']=df['POB_3_5']+df['POB_6_011']+df['POB_012_014']
df=df.drop(['POB_3_5','POB_6_011','POB_012_014'],axis=1)
df['Poblacion de adultos (15-64)']=df['POB_015_29']+df['POB_30_64']
df = df.
 ↪drop(['POB_015_017','POB_015_019','POB_015_29','POB_015_49','POB_018_24','POB_20_24','POB_30_
df = df.rename(columns={'POB_65_MAS':'Poblacion adultos mayores (65+)'})

# Calculo de tasas de cambio por año

df['Cambio jovenes']=df['Poblacion de jovenes (3-14)'].diff().fillna(0)
df['Cambio adultos']=df['Poblacion de adultos (15-64)'].diff().fillna(0)
df['Cambio mayores']=df['Poblacion adultos mayores (65+)'].diff().fillna(0)

print(df)
```

```
    ANIO  Poblacion adultos mayores (65+)  ...  Cambio adultos  Cambio mayores
0   2002                          5399307  ...             0.0             0.0
1   2003                          5576695  ...       1245096.0        177388.0
2   2004                          5761262  ...       1234986.0        184567.0
3   2005                          5953231  ...       1263623.0        191969.0
4   2006                          6155725  ...       1360846.0        202494.0
5   2007                          6368873  ...       1465313.0        213148.0
6   2008                          6585485  ...       1530449.0        216612.0
7   2009                          6808607  ...       1538706.0        223122.0
8   2010                          7046284  ...       1497157.0        237677.0
9   2011                          7302545  ...       1453767.0        256261.0
10  2012                          7569447  ...       1400902.0        266902.0
11  2013                          7840039  ...       1359450.0        270592.0
```

```
12  2014                             8110040  ...     1318887.0       270001.0
13  2015                             8375584  ...     1247533.0       265544.0
14  2017                             8935919  ...     2323830.0       560335.0
15  2018                             9257526  ...     1120796.0       321607.0
16  2019                             9602115  ...     1081726.0       344589.0
17  2020                             9860625  ...      942430.0       258510.0
18  2021                            10027670  ...      826442.0       167045.0
19  2022                            10325860  ...      924775.0       298190.0

[20 rows x 7 columns]
```

[12]:

```
df.to_csv('../data/mexico.csv')
```

[24]:

```
# Modelacion estadistica
import statsmodels.api as sm

# Asegura float en columnas usadas
cols = [
    "Poblacion de adultos (15-64)",
    "Poblacion de jovenes (3-14)",
    "Poblacion adultos mayores (65+)",
    "Cambio jovenes","Cambio adultos","Cambio mayores"
]
df[cols] = df[cols].apply(pd.to_numeric, errors="coerce")
df = df.dropna(subset=cols)

x_J = df[["Poblacion de adultos (15-64)","Poblacion de jovenes (3-14)"]]
y_J = df["Cambio jovenes"]
#x_J = sm.add_constant(x_J,prepend=False)
modelo_J = sm.OLS(y_J, x_J,).fit()
coef_J=modelo_J.params.to_dict()

x_A = df[["Poblacion de jovenes (3-14)","Poblacion de adultos (15-64)"]]
y_A = df["Cambio adultos"]
#x_A = sm.add_constant(x_A,prepend=False)
modelo_A = sm.OLS(y_A, x_A,).fit()
coef_A=modelo_A.params.to_dict()

# Incluye A y E para el modelo de mayores (consistente con dyE = alpha*A - c4*E)
x_E = df[["Poblacion de adultos (15-64)", "Poblacion adultos mayores (65+)"]]
y_E = df["Cambio mayores"]
#x_E = sm.add_constant(x_E,prepend=False)
modelo_E = sm.OLS(y_E, x_E,).fit()
coef_E=modelo_E.params.to_dict()

print(modelo_J.summary())
print(modelo_A.summary())
```

```python
print(modelo_E.summary())
```

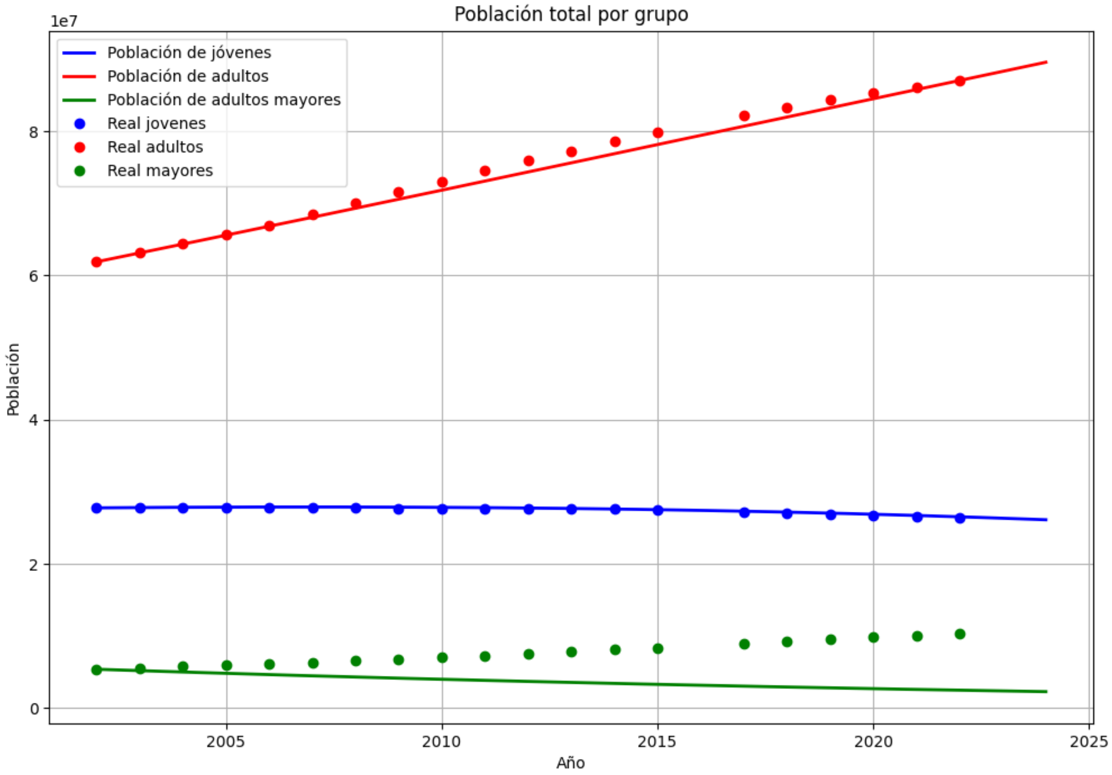
```
                                 OLS Regression Results
================================================================================
=======
Dep. Variable:         Cambio jovenes   R-squared (uncentered):
0.824
Model:                            OLS   Adj. R-squared (uncentered):
0.804
Method:                 Least Squares   F-statistic:
42.08
Date:                Fri, 14 Nov 2025   Prob (F-statistic):
1.64e-07
Time:                        11:16:16   Log-Likelihood:
-244.04
No. Observations:                  20   AIC:
492.1
Df Residuals:                      18   BIC:
494.1
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
================
                                   coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
----------------
Poblacion de adultos (15-64)    -0.0085      0.001     -6.845      0.000
-0.011      -0.006
Poblacion de jovenes (3-14)      0.0207      0.003      6.057      0.000
0.014       0.028
==============================================================================
Omnibus:                       13.709   Durbin-Watson:                   1.332
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               13.941
Skew:                          -1.274   Prob(JB):                     0.000939
Kurtosis:                       6.199   Cond. No.                         25.7
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not
contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                                 OLS Regression Results
================================================================================
=======
Dep. Variable:         Cambio adultos   R-squared (uncentered):
0.903
```

```
Model:                            OLS   Adj. R-squared (uncentered):
0.892
Method:                 Least Squares   F-statistic:
83.69
Date:                Fri, 14 Nov 2025   Prob (F-statistic):
7.67e-10
Time:                        11:16:16   Log-Likelihood:
-286.99
No. Observations:                  20   AIC:
578.0
Df Residuals:                      18   BIC:
580.0
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
================
                                  coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
----------------
Poblacion de jovenes (3-14)     0.0359      0.029      1.225      0.236
-0.026       0.098
Poblacion de adultos (15-64)    0.0036      0.011      0.340      0.738
-0.019       0.026
==============================================================================
Omnibus:                        8.372   Durbin-Watson:                   1.240
Prob(Omnibus):                  0.015   Jarque-Bera (JB):                8.730
Skew:                          -0.531   Prob(JB):                       0.0127
Kurtosis:                       6.057   Cond. No.                         25.7
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not
contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                                 OLS Regression Results
================================================================================
=======
Dep. Variable:         Cambio mayores   R-squared (uncentered):
0.903
Model:                            OLS   Adj. R-squared (uncentered):
0.893
Method:                 Least Squares   F-statistic:
84.09
Date:                Fri, 14 Nov 2025   Prob (F-statistic):
7.38e-10
Time:                        11:16:16   Log-Likelihood:
```

```
-254.84
No. Observations:                  20   AIC:
513.7
Df Residuals:                      18   BIC:
515.7
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
===================
                                      coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
-------------------
Poblacion de adultos (15-64)       -0.0002      0.003     -0.076      0.940
-0.006       0.005
Poblacion adultos mayores (65+)     0.0344      0.026      1.325      0.202
-0.020       0.089
==============================================================================
Omnibus:                       13.437   Durbin-Watson:                   1.473
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               19.057
Skew:                           0.937   Prob(JB):                     7.28e-05
Kurtosis:                       7.400   Cond. No.                         101.
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not
contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

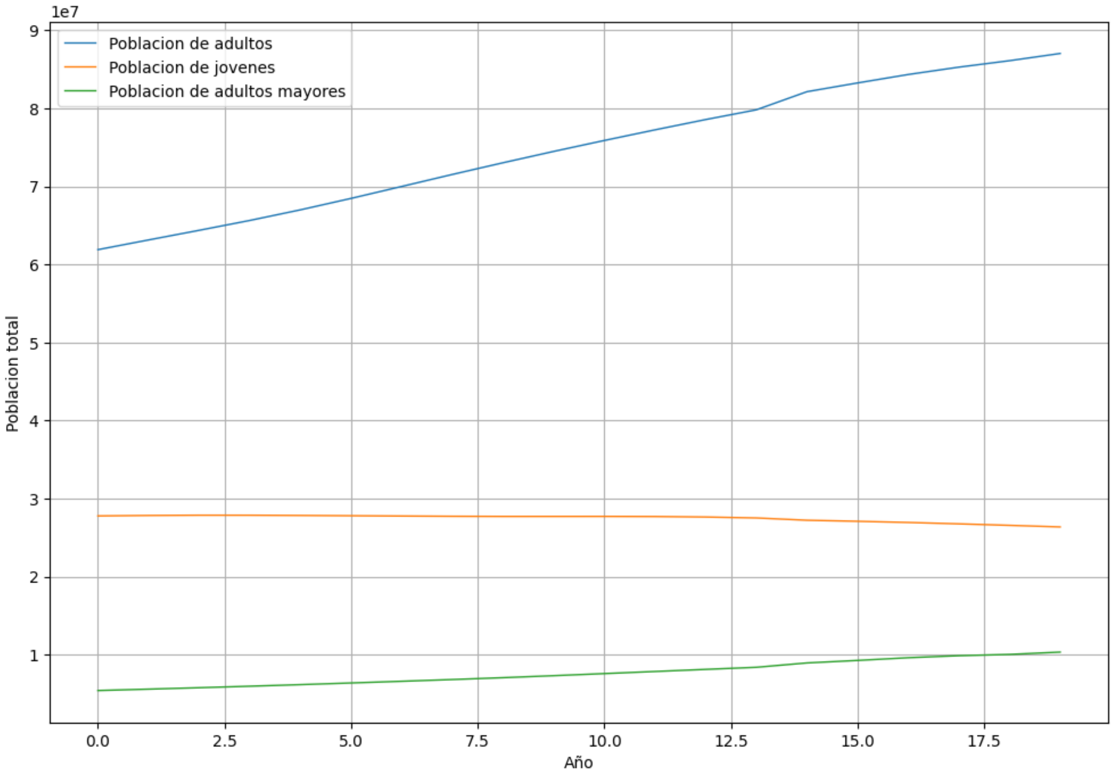
[28]:

```python
# Graficacion

import numpy as np
import matplotlib.pyplot as plt

def sistema_edos(t, y):
    # y = [J, A, E]
    jt, at, et = y

    # COEFICIENTES
    c1 = -0.0085  # tasa de natalidad
    c2 = -0.0207  # tasa de mortalidad jóvenes
    #const1= -2.161e+06
    tj=0.0359
    c3 = -0.0036  # tasa de mortalidad adultos (positiva)
    #const2=-3.146e+07
    ta=-0.0002
    c4 =  0.0344 # tasa de mortalidad mayores
    #const3=-1.404e+06

    # ECUACIONES
    dyJ = c1*at-c2*jt#+const1
    dyA = tj*jt-c3*at#+const2
    dyE = ta*at-c4*et#+const3
    return [dyJ, dyA, dyE]

def RK4(func, y0, t0, tf, h):
    t_values = np.arange(t0, tf + h, h)
    n = len(t_values)
    y_values = np.zeros((n, len(y0)), dtype=float)
    y_values[0] = y0

    for i in range(1, n):
        k1 = np.array(func(t_values[i-1], y_values[i-1]))
        k2 = np.array(func(t_values[i-1] + h/2, y_values[i-1] + h*k1/2))
        k3 = np.array(func(t_values[i-1] + h/2, y_values[i-1] + h*k2/2))
        k4 = np.array(func(t_values[i-1] + h,   y_values[i-1] + h*k3))
        y_values[i] = y_values[i-1] + h*(k1 + 2*k2 + 2*k3 + k4)/6

    return t_values, y_values

# y0 = [J, A, E] en 2002
y0 = [27773497,61886631,5399307]
t0, tf, h = 0, 22, 0.01

t, y = RK4(sistema_edos, y0, t0, tf, h)

years = 2002 + t
fig, ax = plt.subplots(1, 1, figsize=(12, 8))
# Modelo
ax.plot(years, y[:, 0], 'b-', linewidth=2, label='Población de jóvenes')
ax.plot(years, y[:, 1], 'r-', linewidth=2, label='Población de adultos')
ax.plot(years, y[:, 2], 'g-', linewidth=2, label='Población de adultos mayores')
# Reales
years_real=df["ANIO"]
ax.plot(years_real,df["Poblacion de jovenes␣
 ↪(3-14)"],'bo',markersize=6,label="Real jovenes")
ax.plot(years_real,df["Poblacion de adultos␣
 ↪(15-64)"],'ro',markersize=6,label="Real adultos")
ax.plot(years_real,df["Poblacion adultos mayores␣
 ↪(65+)"],'go',markersize=6,label="Real mayores")

ax.set_title('Población total por grupo')
ax.set_xlabel('Año')
ax.set_ylabel('Población')
ax.grid(True)
ax.legend()
plt.show()
```



```
[ ]: fig,ax = plt.subplots(1,1,figsize=(12,8))
     ax.plot(df["Poblacion de adultos (15-64)"],linewidth=1,label="Poblacion de␣
      ↪adultos")
     ax.plot(df["Poblacion de jovenes (3-14)"],linewidth=1,label="Poblacion de␣
      ↪jovenes")
     ax.plot(df["Poblacion adultos mayores (65+)"],linewidth=1,label="Poblacion de␣
      ↪adultos mayores")
     ax.set_xlabel("Año")
     ax.set_ylabel("Poblacion total")
     ax.grid(True)
     ax.legend()
     plt.show
```

```
[ ]: <function matplotlib.pyplot.show(close=None, block=None)>
```


1e7
9
8
7
6
5
4
3
2
1
Poblacion total
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Año
Poblacion de adultos
Poblacion de jovenes
Poblacion de adultos mayores