Python y DataFrames: un match para los modelados

Arossa N., Bonifacino N., Chiattellino M., Herrera C., González A., Hernandez K., Nieves J., Vega K; García J.

Universidad Nacional de Hurlingham

Actividad dirigida a estudiantes del cuarto año del Profesorado Universitario de Matemática.

Objetivos

- Posibilitar el abordaje a actividades de modelado en base a datos reales de libre disponibilidad, evitando reproducir *ejemplos de juguete*.
- Utilizar la programación, vía Python, para explorar el modelado poblacional, aprovechando la eficiencia, potencia y simplicidad que esta provee para la manipulación de datos.
- Experimentar con diferentes modelos de crecimiento en relación con las ecuaciones diferenciales que los caracterizan.

Contexto

Los datos de la población mundial son bien conocidos (e. g. Wikipedia). Utilizamos diferentes modelos para ver cómo se adaptan a la dinámica de la misma. En última instancia exploramos predicciones sobre la evolución de la población mundial a futuro.

Convertir datos a DataFrames es muy sencillo

Se pueden extraer directamente de un sitio web (o archivo .html) o de un .csv o de un .xls, o de muchos otros formatos. En este caso extrajimos una tabla de Wikipedia.

1950 to 2016 [edit]

After World War II, demographic data of some accuracy becomes available f
estimates are often given as grand totals of numbers (typically given by cour
these numbers rounded to the nearest million or the nearest thousand, while

Taking these numbers at face value would be false precision; in spite of bein
be interpreted as accurate to more than three digits at best (estimates by the
differ by about 0.5–1.5%).

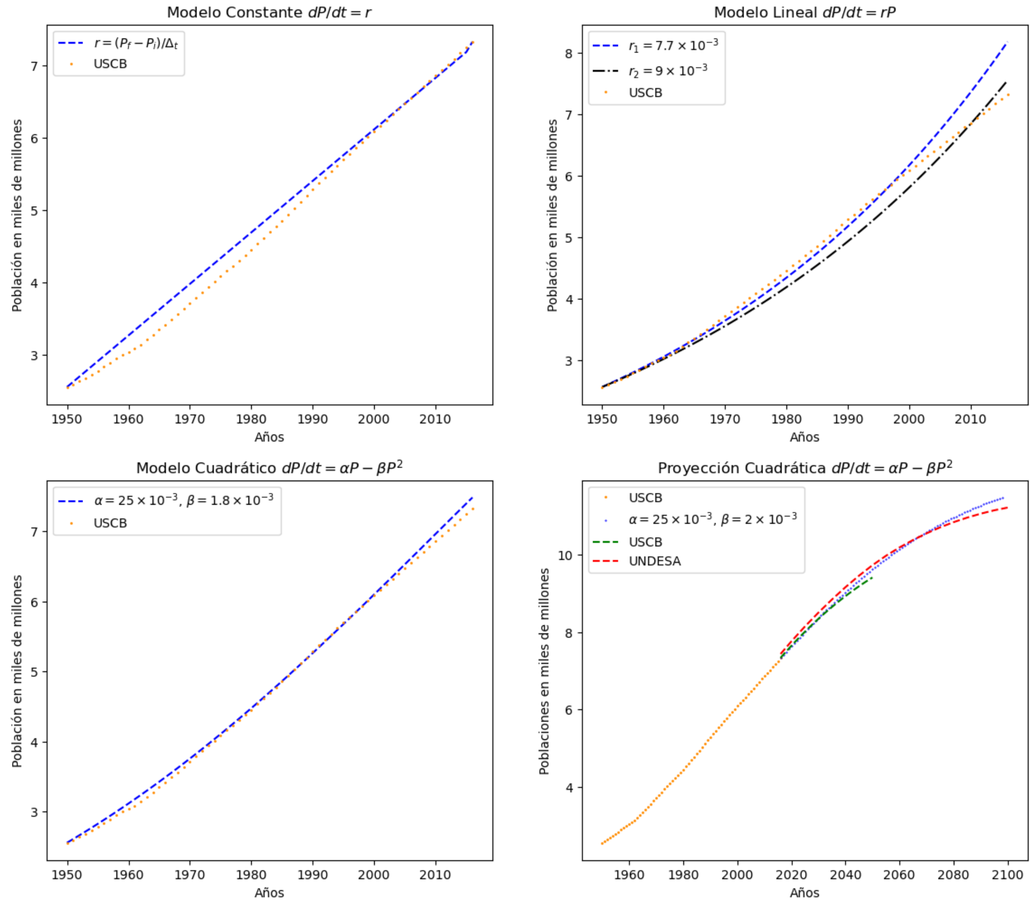
Year	United States Census Bureau (2017)[28]	Population Reference Bureau (1973–2016)[15]	United Nations Department of Economic and Social Affairs (2015)[16]	Maddison (2008)[17]	HYDE (2007)[24]
1950	2,557,628,654	2,516,000,000	2,525,149,000	2,544,000,000	2,527,960,000
1951	2,594,939,877		2,572,850,917	2,571,663,000	
1952	2,636,772,306		2,619,292,068	2,617,949,000	
1953	2,682,053,389		2,665,865,392	2,665,959,000	

```
1 from pandas import read_html
2
3 filename = 'Estimates_World_Pop_Wiki_Donwload.html'
4 tables = read_html(filename, header=0, index_col=0, decimal='M')
```

```
1 tables[2]
```

Year	United States Census Bureau (2017)[28]	Population Reference Bureau (1973–2016) [15]	United Nations Department of Economic and Social Affairs (2015)[16]	Maddison (2008)[17]	HYDE (2007) [24]
1950	2557628654	2.516000e+09	2.525149e+09	2.544000e+09	2.527960e+09
1951	2594939877	NaN	2.572851e+09	2.571663e+09	NaN
1952	2636772306	NaN	2.619292e+09	2.617949e+09	NaN
1953	2682053389	NaN	2.665865e+09	2.665959e+09	NaN

Evolución de la población mundial



Tipos de crecimiento

Constante	Lineal	Cuadrático
$\frac{dP}{dt} = r$	$\frac{dP}{dt} = rP$	$\frac{dP}{dt} = \alpha P - \beta P^2 = rP\left(1 - \frac{P}{K}\right)$

La ecuación logística modela crecimientos demográficos denso-dependientes.

Diferenciales y Euler

Con un dato inicial $P(t_0) = P_0$ y permitiéndonos operar con diferenciales (si Leibniz lo hacía...)

$$\left.\begin{aligned}\frac{dP}{dt} &= f(P,t)\\ \frac{\Delta P}{\Delta t} &\approx f(P_0,t_0)\end{aligned}\right\} \implies \begin{aligned} P_1 - P_0 &\approx f(P_0,t_0)(t_1 - t_0)\\ P1 &\approx P_0 + f(P_0,t_0)(t_1 - t_0)\end{aligned}$$

$$\left.\begin{aligned}t_n &= t_0 + nh\\ P_n &= P_{n-1} + hf(t_{n-1}, P_{n-1})\end{aligned}\right\} = \text{Euler}$$

Un poco de código

```python
import matplotlib.pyplot as plt

def modelo_cuadratico_Euler(p,a,b):
    return a*p-b*p**2

def correr_simulacion_cuadratica(dataframe, func_crecimiento, a, b):

    # Extraemos el año inicial, el año final y la población inicial del DataFrame
    año_inicial       = dataframe.index[0]
    año_final         = dataframe.index[-1]
    poblacion_inicial = dataframe.iloc[0,0]/1e9
    primer_columna = dataframe.iloc[:, 0]/1e9

    # Hacemos una copia de la primer columna de población para rellenar luego
    resultados_euler_cuadratico = primer_columna.copy()

    for t in range(año_inicial, año_final):
        resultados_euler_cuadratico[t+1] = poblacion_inicial + func_crecimiento(poblacion_inicial,a, b)
        poblacion_inicial = resultados_euler_cuadratico[t+1]

    fig  = plt.figure()
    ax   = fig.add_axes([0,0,1,1])
    ax.plot(resultados_euler_cuadratico, '--',  color='blue', label= r'$\alpha = 25\times 10^{-3}$, $\beta = 1.8\times 10^{-3}$ ')

    ax.plot(census,'.',color='orange', label= 'USCB', markersize=2 )
    ax.set_title(r"Modelo Cuadrático $dP/dt = \alpha P - \beta P^2$ ")
    ax.set_xlabel("Años")
    ax.set_ylabel("Población en miles de millones")
    ax.legend()
    plt.show()


    return resultados_euler_cuadratico

correr_simulacion_cuadratica(PM, modelo_cuadratico_Euler, 25/1000, 1.8/1000);
```

Reflexión final

Python es una herramienta poderosa y versátil para poder analizar datos y modelarlos que, combinada con el acceso a un vasto conjunto de DataFrames, archivos .csv y todo otro tipo de datos de libre disponibilidad, posibilita explorar las actividades de modelado con una profundidad contundente. Finalmente, recuperamos un conocimiento estándar en la formación matemática (recta tangente) y le damos un sentido de herramienta que resulta central en la posibilidad de simular el modelo.