

Universidad Politecnica Salesiana

Nombre: Jéssica Ñauta

Materia: Simulación

Covid-19 infección en Ecuador. Modelos matemáticos y predicciones

Una comparación de modelos, lineal, polinómico, logísticos y exponenciales aplicados a la infección por el virus Covid-19

Se realiza un análisis matemático simple del crecimiento de la infección en Python y dos modelos para comprender mejor la evolución de la infección.

Se crean modelos de series temporales del número total de personas infectadas hasta la fecha (es decir, las personas realmente infectadas más las personas que han sido infectadas). Estos modelos tienen parámetros, que se estimarán por ajuste de curva.

Formulación del Problema:

Determina el objeto de la simulación. Se deben especificar los siguientes elementos:

- Resultados que se esperan del simulador: mediante el modelo logístico obtener una predicción sobre los contagios del covid-19.
- Plan de experimentación: buscar datos de covid en el Ecuador y realizar la simulación.
- Variables de interés: variables que nos ayudaran a realizar el cálculo para obtener una estadística de infectados.
- Tipo de perturbaciones a estudiar: librerías que nos ayudaran a implementar el método de forma sencilla.
- Tratamiento estadístico de los resultados: graficar para obtener un resultado y poderlo visualizar de mejor manera las estadísticas.
- Complejidad del interfaz del simulador: no tiene una interfaz.

Definición del sistema:

El sistema que se simulará debe estar definido perfectamente. Se debe establecer donde estará la frontera de interacción entre el sistema a estudiar y el medioambiente.

```
In [1]: # Importar las librerías para el análisis
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
import matplotlib.pyplot as plt
%matplotlib inline
```

```
from xml.dom import minidom
from datetime import datetime
instanteInicial = datetime.now()
```

Formulación del modelo:

En esta etapa se capturan los aspectos relevantes del sistema real. En este caso vamos a capturar los datos de cada persona para saber cuantas estuvieron infectados, cuantas ya superaron, muertes confirmadas, entre otras, para poder aplicar el método y predecir el futuro sobre esta pandemia.

Colección de datos:

La naturaleza y la cantidad de datos se determinan por la formulación del problema y del modelo. Pueden ser obtenidos de registros históricos, experimentos en laboratorio o mediciones realizadas en el sistema real. En este caso hemos accedido al siguiente enlace para obtener la información: <https://github.com/owid/covid-19-data/tree/master/public/data>

```
In [9]: # Actualizar los datos (URL)
# Datos obtenidos de: https://github.com/owid/covid-19-data/tree/master/public/data
url = 'owid-covid-data.csv'
df = pd.read_csv(url)
df = df.fillna(0)
df
```

```
Out[9]:
```

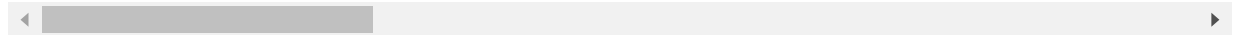
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
0	AFG	Asia	Afghanistan	2019-12-31	0.0	0.0	0.0	
1	AFG	Asia	Afghanistan	2020-01-01	0.0	0.0	0.0	
2	AFG	Asia	Afghanistan	2020-01-02	0.0	0.0	0.0	
3	AFG	Asia	Afghanistan	2020-01-03	0.0	0.0	0.0	
4	AFG	Asia	Afghanistan	2020-01-04	0.0	0.0	0.0	
5	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	0.0	
6	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	0.0	
7	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	0.0	
8	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0	0.0	
9	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0	0.0	
10	AFG	Asia	Afghanistan	2020-01-10	0.0	0.0	0.0	
11	AFG	Asia	Afghanistan	2020-01-11	0.0	0.0	0.0	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
12	AFG	Asia	Afghanistan	2020-01-12	0.0	0.0	0.0	
13	AFG	Asia	Afghanistan	2020-01-13	0.0	0.0	0.0	
14	AFG	Asia	Afghanistan	2020-01-14	0.0	0.0	0.0	
15	AFG	Asia	Afghanistan	2020-01-15	0.0	0.0	0.0	
16	AFG	Asia	Afghanistan	2020-01-16	0.0	0.0	0.0	
17	AFG	Asia	Afghanistan	2020-01-17	0.0	0.0	0.0	
18	AFG	Asia	Afghanistan	2020-01-18	0.0	0.0	0.0	
19	AFG	Asia	Afghanistan	2020-01-19	0.0	0.0	0.0	
20	AFG	Asia	Afghanistan	2020-01-20	0.0	0.0	0.0	
21	AFG	Asia	Afghanistan	2020-01-21	0.0	0.0	0.0	
22	AFG	Asia	Afghanistan	2020-01-22	0.0	0.0	0.0	
23	AFG	Asia	Afghanistan	2020-01-23	0.0	0.0	0.0	
24	AFG	Asia	Afghanistan	2020-01-24	0.0	0.0	0.0	
25	AFG	Asia	Afghanistan	2020-01-25	0.0	0.0	0.0	
26	AFG	Asia	Afghanistan	2020-01-26	0.0	0.0	0.0	
27	AFG	Asia	Afghanistan	2020-01-27	0.0	0.0	0.0	
28	AFG	Asia	Afghanistan	2020-01-28	0.0	0.0	0.0	
29	AFG	Asia	Afghanistan	2020-01-29	0.0	0.0	0.0	
...
56081	0	0	International	2020-10-13	696.0	0.0	0.0	
56082	0	0	International	2020-10-14	696.0	0.0	0.0	
56083	0	0	International	2020-10-15	696.0	0.0	0.0	
56084	0	0	International	2020-10-16	696.0	0.0	0.0	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
56085	0	0	International	2020-10-17	696.0	0.0	0.0	
56086	0	0	International	2020-10-18	696.0	0.0	0.0	
56087	0	0	International	2020-10-19	696.0	0.0	0.0	
56088	0	0	International	2020-10-20	696.0	0.0	0.0	
56089	0	0	International	2020-10-21	696.0	0.0	0.0	
56090	0	0	International	2020-10-22	696.0	0.0	0.0	
56091	0	0	International	2020-10-23	696.0	0.0	0.0	
56092	0	0	International	2020-10-24	696.0	0.0	0.0	
56093	0	0	International	2020-10-25	696.0	0.0	0.0	
56094	0	0	International	2020-10-26	696.0	0.0	0.0	
56095	0	0	International	2020-10-27	696.0	0.0	0.0	
56096	0	0	International	2020-10-28	696.0	0.0	0.0	
56097	0	0	International	2020-10-29	696.0	0.0	0.0	
56098	0	0	International	2020-10-30	696.0	0.0	0.0	
56099	0	0	International	2020-10-31	696.0	0.0	0.0	
56100	0	0	International	2020-11-01	696.0	0.0	0.0	
56101	0	0	International	2020-11-02	696.0	0.0	0.0	
56102	0	0	International	2020-11-03	696.0	0.0	0.0	
56103	0	0	International	2020-11-04	696.0	0.0	0.0	
56104	0	0	International	2020-11-05	696.0	0.0	0.0	
56105	0	0	International	2020-11-06	696.0	0.0	0.0	
56106	0	0	International	2020-11-07	696.0	0.0	0.0	
56107	0	0	International	2020-11-08	696.0	0.0	0.0	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
56108	0	0	International	2020-11-09	696.0	0.0	0.0	
56109	0	0	International	2020-11-10	696.0	0.0	0.0	
56110	0	0	International	2020-11-11	696.0	0.0	0.0	

56111 rows × 49 columns



Implementación del modelo en el ordenador:

Se implementa el modelo a través de un lenguaje de programación/simulación. En este caso usaremos python.

Imprimos los resultados y agregamos el numero del dia

```
In [10]: df = df[df['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuador
df = df.loc[:,['date','total_cases']] #Selecciono las columnas de analisis
# Expresar Las fechas en numero de dias desde el 01 Enero
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01-01", FMT)).days)
df
```

Out[10]:

	date	total_cases
14965	-1	0.0
14966	0	0.0
14967	1	0.0
14968	2	0.0
14969	3	0.0
14970	4	0.0
14971	5	0.0
14972	6	0.0
14973	7	0.0
14974	8	0.0
14975	9	0.0
14976	10	0.0
14977	11	0.0
14978	12	0.0
14979	13	0.0
14980	14	0.0
14981	15	0.0

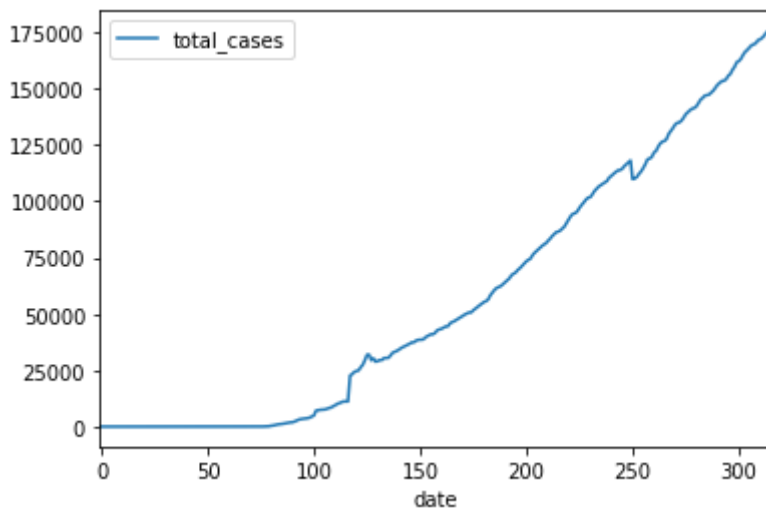
	date	total_cases
14982	16	0.0
14983	17	0.0
14984	18	0.0
14985	19	0.0
14986	20	0.0
14987	21	0.0
14988	22	0.0
14989	23	0.0
14990	24	0.0
14991	25	0.0
14992	26	0.0
14993	27	0.0
14994	28	0.0
...
15252	286	147315.0
15253	287	148171.0
15254	288	149083.0
15255	289	150360.0
15256	290	151659.0
15257	291	152422.0
15258	292	153289.0
15259	293	153423.0
15260	294	154115.0
15261	295	155625.0
15262	296	156451.0
15263	297	158270.0
15264	298	159614.0
15265	299	161635.0
15266	300	162178.0
15267	301	163192.0
15268	302	164908.0
15269	303	166302.0
15270	304	167147.0
15271	305	168192.0
15272	306	169194.0
15273	307	169562.0

	date	total_cases
15274	308	170110.0
15275	309	171433.0
15276	310	171783.0
15277	311	172508.0
15278	312	173486.0
15279	313	174907.0
15280	314	175269.0
15281	315	175711.0

317 rows × 2 columns

```
In [11]: df.plot(x='date', y='total_cases')
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa4e998988>
```



Modelo Lineal

```
In [13]: x = list(df.iloc[:, 0]) # Fecha
y = list(df.iloc[:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)
#pdb.set_trace()
# Veamos los coeficientes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
```

```
Coefficients:
[593.2518536]
Independent term:
-35652.30126823376
```

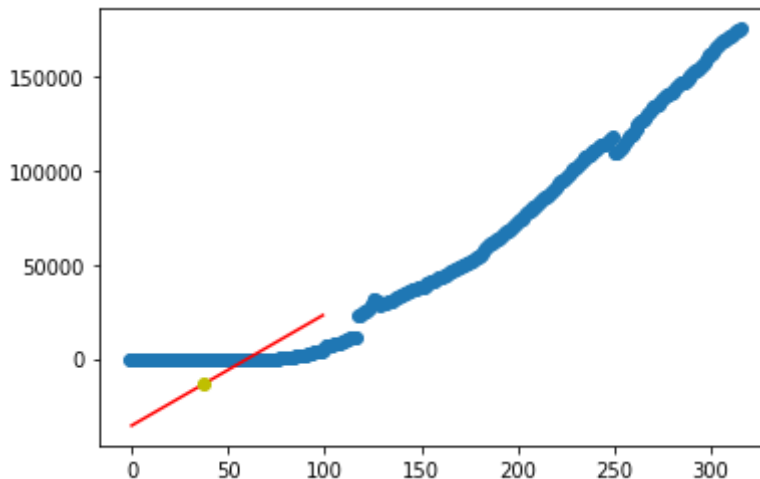
```
In [14]: #Vamos a comprobar:
# Quiero predecir cuántos "Casos" voy a obtener por en el día 100,
```

```
# según nuestro modelo, hacemos:
y_prediccion = regr.predict([[100]])
print(int(y_prediccion))
```

23672

```
In [15]: plt.scatter(x, y)
x_real = np.array(range(0,100))
#print(x_real)
puntos=regr.predict(x_real.reshape(-1, 1))
plt.plot(x_real,puntos, color='red')
plt.plot(37,puntos[37],'oy')
print ('Predicción a 7 días sumando desde el ultimo día en x(30):', puntos[37], 'con')
plt.show()
```

Predicción a 7 días sumando desde el ultimo día en x(30): -13701.98268487637 contagiados



El modelo logístico

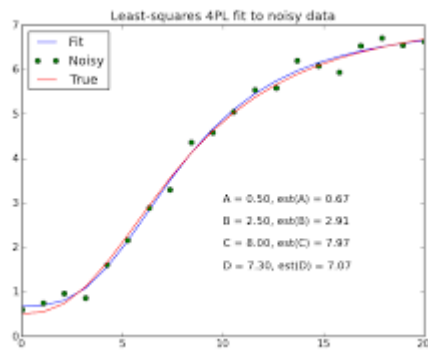
El modelo logístico se ha utilizado ampliamente para describir el crecimiento de una población. Una infección puede describirse como el crecimiento de la población de un agente patógeno, por lo que un modelo logístico parece razonable. La expresión más genérica de una función logística es:

$$f(x, a, b, c) = \frac{c}{1 + e^{-(x-b)/a}}$$

En esta fórmula, tenemos la variable x que es el tiempo y tres parámetros: a, b, c.

- a se refiere a la velocidad de infección
- b es el día en que ocurrieron las infecciones máximas
- c es el número total de personas infectadas registradas al final de la infección

A continuación se puede apreciar un ejemplo de regresión logística



Definamos la función en Python y realicemos el procedimiento de ajuste de curva utilizado para el crecimiento logístico.

```
In [2]: def modelo_logistico(x,a,b):
        return a+b*np.log(x)

exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los paramatros
print(exp_fit)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-2-409fabac0849> in <module>
      2     return a+b*np.log(x)
      3
----> 4 exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los para
      5 print(exp_fit)
NameError: name 'x' is not defined
```

```
In [12]: x = list(df.iloc[:, 0]) # Fecha
y = list(df.iloc[:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)
#pdb.set_trace()
# Veamos los coeficientes obtenidos, En nuestro caso, serán La Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio

Coefficients:
[593.2518536]
Independent term:
-35652.30126823376
```

Graficas

```
In [27]: pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
plt.rcParams['figure.figsize'] = [7, 7]
plt.rc('font', size=14)
# Real data
plt.scatter(x,y,label="Datos Reales",color="red")
# Predicted exponential curve
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x],
plt.legend()
plt.xlabel("Desde el 1 Enero 2020")
plt.ylabel("Total de personas infectadas")
```

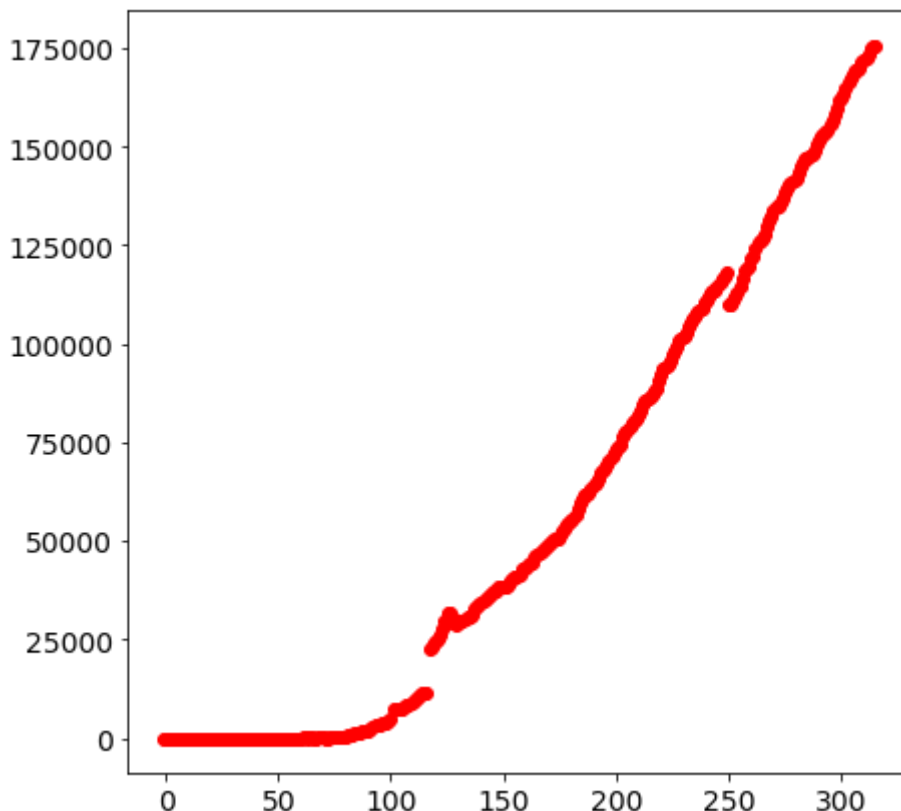
```
plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
plt.show()
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-27-a9b5bfb0e5aa> in <module>
      5 plt.scatter(x,y,label="Datos Reales",color="red")
      6 # Predicted exponential curve
----> 7 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in p
red_x], label="Modelo Logistico" )
      8 plt.legend()
      9 plt.xlabel("Desde el 1 Enero 2020")

<ipython-input-27-a9b5bfb0e5aa> in <listcomp>(.0)
      5 plt.scatter(x,y,label="Datos Reales",color="red")
      6 # Predicted exponential curve
----> 7 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in p
red_x], label="Modelo Logistico" )
      8 plt.legend()
      9 plt.xlabel("Desde el 1 Enero 2020")

<ipython-input-26-63ac4d764725> in modelo_logistico(x, a, b)
      1 def modelo_logistico(x,a,b):
      2     res=0
----> 3     if x[1] == 0:
      4         x=1
      5     else:
```

TypeError: 'int' object is not subscriptable



Modelo exponencial

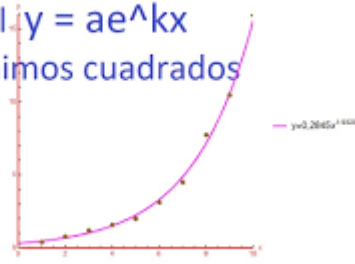
Mientras que el modelo logístico describe un crecimiento de infección que se detendrá en el futuro, el modelo exponencial describe un crecimiento de infección imparables. Por ejemplo, si un paciente infecta a 2 pacientes por día, después de 1 día tendremos 2 infecciones, 4 después

$$f(x, a, b, c) = a \cdot e^{b(x-c)}$$

de 2 días, 8 después de 3 y así sucesivamente.

A continuacion se tiene un ejemplo de regresion exponencial

Curva de ajuste para una función tipo
exponencial $y = ae^{kx}$
usando mínimos cuadrados



In [29]: `# Implementar`

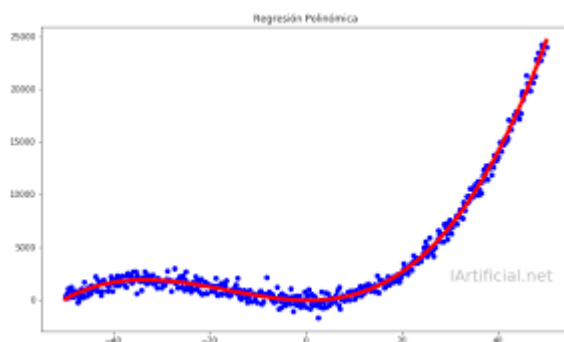
Modelo polinomial

Predicción de una variable de respuesta cuantitativa a partir de una variable predictora cuantitativa, donde la relación se modela como una función polinomial de orden n (esto significa que pueden tener de diferentes exponenciales o grados y se debe ir probando)

Se puede tener una ecuacion con diferentes grados

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n + \varepsilon$$

Ejemplo de una regresion polinomial de grado 4.



In []: `# Implementar`

```
# Se puede implementar modelos adicionales, en caso de ser asi explicar o dar una in
# Se tomara como puntos adicionales al trabajo.
```

Covid en Ecuador

```
In [45]: # Implementar
# Filtrar los datos de Ecuador
df = pd.read_csv('owid-covid-data.csv').fillna(0)
ndf= df.loc[(df['location'] == 'Ecuador') & (df['total_cases'] != 0)]
ndf
```

```
Out[45]:
```

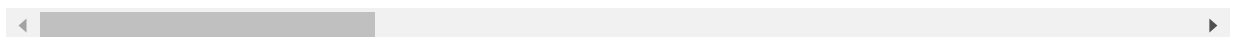
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
15026	ECU	South America	Ecuador	2020-03-01	1.0	1.0	0.143	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
15027	ECU	South America	Ecuador	2020-03-02	6.0	5.0	0.857	
15028	ECU	South America	Ecuador	2020-03-03	7.0	1.0	1.000	
15030	ECU	South America	Ecuador	2020-03-05	10.0	3.0	1.429	
15031	ECU	South America	Ecuador	2020-03-06	13.0	3.0	1.857	
15034	ECU	South America	Ecuador	2020-03-09	14.0	1.0	1.143	
15035	ECU	South America	Ecuador	2020-03-10	15.0	1.0	1.143	
15036	ECU	South America	Ecuador	2020-03-11	17.0	2.0	1.429	
15039	ECU	South America	Ecuador	2020-03-14	23.0	6.0	1.429	
15040	ECU	South America	Ecuador	2020-03-15	28.0	5.0	2.143	
15041	ECU	South America	Ecuador	2020-03-16	37.0	9.0	3.286	
15042	ECU	South America	Ecuador	2020-03-17	58.0	21.0	6.143	
15043	ECU	South America	Ecuador	2020-03-18	111.0	53.0	13.429	
15044	ECU	South America	Ecuador	2020-03-19	168.0	57.0	21.571	
15045	ECU	South America	Ecuador	2020-03-20	199.0	31.0	26.000	
15046	ECU	South America	Ecuador	2020-03-21	426.0	227.0	57.571	
15047	ECU	South America	Ecuador	2020-03-22	532.0	106.0	72.000	
15048	ECU	South America	Ecuador	2020-03-23	789.0	257.0	107.429	1
15049	ECU	South America	Ecuador	2020-03-24	981.0	192.0	131.857	1
15050	ECU	South America	Ecuador	2020-03-25	1082.0	101.0	138.714	2
15051	ECU	South America	Ecuador	2020-03-26	1211.0	129.0	149.000	2
15052	ECU	South America	Ecuador	2020-03-27	1403.0	192.0	172.000	3
15053	ECU	South America	Ecuador	2020-03-28	1627.0	224.0	171.571	4
15054	ECU	South America	Ecuador	2020-03-29	1835.0	208.0	186.143	4

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
15055	ECU	South America	Ecuador	2020-03-30	1890.0	55.0	157.286	5
15056	ECU	South America	Ecuador	2020-03-31	1966.0	76.0	140.714	6
15057	ECU	South America	Ecuador	2020-04-01	2302.0	336.0	174.286	7
15058	ECU	South America	Ecuador	2020-04-02	2758.0	456.0	221.000	12
15059	ECU	South America	Ecuador	2020-04-03	3163.0	405.0	251.429	12
15060	ECU	South America	Ecuador	2020-04-04	3368.0	205.0	248.714	14
...
15252	ECU	South America	Ecuador	2020-10-13	147315.0	282.0	853.714	1221
15253	ECU	South America	Ecuador	2020-10-14	148171.0	856.0	873.571	1223
15254	ECU	South America	Ecuador	2020-10-15	149083.0	912.0	793.143	1226
15255	ECU	South America	Ecuador	2020-10-16	150360.0	1277.0	759.286	1230
15256	ECU	South America	Ecuador	2020-10-17	151659.0	1299.0	830.143	1235
15257	ECU	South America	Ecuador	2020-10-18	152422.0	763.0	799.143	1237
15258	ECU	South America	Ecuador	2020-10-19	153289.0	867.0	893.714	1238
15259	ECU	South America	Ecuador	2020-10-20	153423.0	134.0	872.571	1239
15260	ECU	South America	Ecuador	2020-10-21	154115.0	692.0	849.143	1240
15261	ECU	South America	Ecuador	2020-10-22	155625.0	1510.0	934.571	1245
15262	ECU	South America	Ecuador	2020-10-23	156451.0	826.0	870.143	1250
15263	ECU	South America	Ecuador	2020-10-24	158270.0	1819.0	944.429	1252
15264	ECU	South America	Ecuador	2020-10-25	159614.0	1344.0	1027.429	1254
15265	ECU	South America	Ecuador	2020-10-26	161635.0	2021.0	1192.286	1255
15266	ECU	South America	Ecuador	2020-10-27	162178.0	543.0	1250.714	1257
15267	ECU	South America	Ecuador	2020-10-28	163192.0	1014.0	1296.714	1258

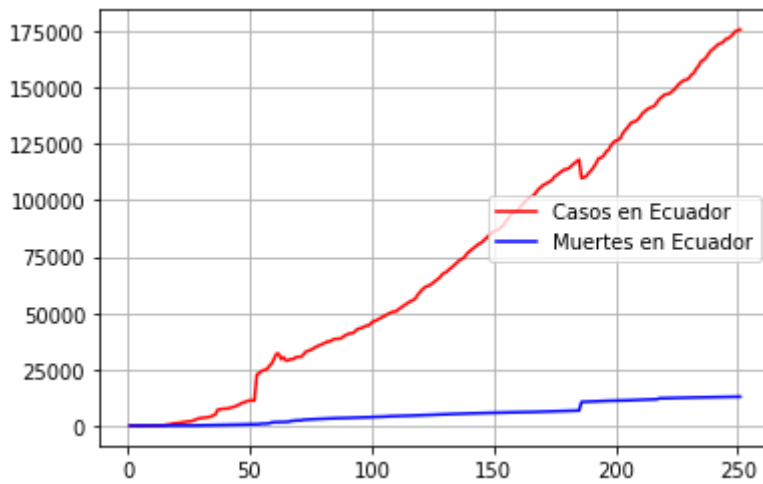
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_dea
15268	ECU	South America	Ecuador	2020-10-29	164908.0	1716.0	1326.143	1260
15269	ECU	South America	Ecuador	2020-10-30	166302.0	1394.0	1407.286	1262
15270	ECU	South America	Ecuador	2020-10-31	167147.0	845.0	1268.143	1263
15271	ECU	South America	Ecuador	2020-11-01	168192.0	1045.0	1225.429	1267
15272	ECU	South America	Ecuador	2020-11-02	169194.0	1002.0	1079.857	1268
15273	ECU	South America	Ecuador	2020-11-03	169562.0	368.0	1054.857	1269
15274	ECU	South America	Ecuador	2020-11-04	170110.0	548.0	988.286	1269
15275	ECU	South America	Ecuador	2020-11-05	171433.0	1323.0	932.143	1270
15276	ECU	South America	Ecuador	2020-11-06	171783.0	350.0	783.000	1273
15277	ECU	South America	Ecuador	2020-11-07	172508.0	725.0	765.857	1276
15278	ECU	South America	Ecuador	2020-11-08	173486.0	978.0	756.286	1281
15279	ECU	South America	Ecuador	2020-11-09	174907.0	1421.0	816.143	1283
15280	ECU	South America	Ecuador	2020-11-10	175269.0	362.0	815.286	1283
15281	ECU	South America	Ecuador	2020-11-11	175711.0	442.0	800.143	1284

251 rows × 49 columns



```
In [49]: # Graficar los casos y muertes por covid en Ecuador
plt.plot(x,y,label='Casos en Ecuador', color='red')
plt.plot(x,y1,label='Muertes en Ecuador', color='blue')
plt.grid(True)
plt.legend(loc=5)
```

Out[49]: <matplotlib.legend.Legend at 0x14d8168a888>



Analisis

Mediante el método de regresión lineal podemos predecir el número de infectados por covid-19 en Ecuador mediante datos encontrados en internet, este método nos ayuda a tener una aproximación para de esta manera poder observar cual es el comportamiento de la pandemia en el futuro.

Conclusiones

Como conclusión tenemos que la regresión lineal es un modelo matemático que nos ayuda a predecir el futuro en este caso sobre la pandemia del covid-19, este modelo es subjetivo por lo que no siempre acertara con veracidad lo que sucederá a futuro, pero si nos dará una idea de lo que sucederá y de esta manera las personas puedan tomar consciencia y prevenir el virus.

Criterio personal (politico, economico y social de la situacion)

En el ámbito político las autoridades de nuestro país deben regir medidas de seguridad, de salud y realizar planificaciones que ayuden al bienestar de la sociedad para de esa manera reducir el contagio y evitar mas muertes, en el ámbito económico esta pandemia a afectado a muchos negocios que algunos tuvieron que cerrar, existe una tasa alta de desempleo por lo que las personas ya no tienen dinero y puede ocasionar enfermedades como la desnutrición y otras, pero también existen personas que han emprendido un negocio vendiendo objetos mediante redes sociales, y en el ámbito social puedo decir que esta pandemia a afectado a todas las personas de nuestro país como al mundo entero, tanto en la salud, ingresos y empleo.

Referencias

- https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infeccion
- <https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>