

# 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.

```
In [31]: # 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
```

```
In [18]: # 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[18]:
```

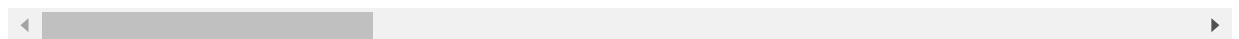
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
0	AFG	Asia	Afghanistan	2019-12-31	0.0	0.0	0.0	
1	AFG	Asia	Afghanistan	2020-01-01	0.0	0.0	0.0	
2	AFG	Asia	Afghanistan	2020-01-02	0.0	0.0	0.0	
3	AFG	Asia	Afghanistan	2020-01-03	0.0	0.0	0.0	
4	AFG	Asia	Afghanistan	2020-01-04	0.0	0.0	0.0	
5	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	0.0	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
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	
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	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
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	
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	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
<b>56102</b>	0	0	International	2020-11-03	696.0	0.0	0.0	
<b>56103</b>	0	0	International	2020-11-04	696.0	0.0	0.0	
<b>56104</b>	0	0	International	2020-11-05	696.0	0.0	0.0	
<b>56105</b>	0	0	International	2020-11-06	696.0	0.0	0.0	
<b>56106</b>	0	0	International	2020-11-07	696.0	0.0	0.0	
<b>56107</b>	0	0	International	2020-11-08	696.0	0.0	0.0	
<b>56108</b>	0	0	International	2020-11-09	696.0	0.0	0.0	
<b>56109</b>	0	0	International	2020-11-10	696.0	0.0	0.0	
<b>56110</b>	0	0	International	2020-11-11	696.0	0.0	0.0	

56111 rows × 49 columns



Imprimos los resultados y agregamos el numero del día

```
In [19]: 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[19]:

	date	total_cases
<b>14965</b>	-1	0.0
<b>14966</b>	0	0.0
<b>14967</b>	1	0.0
<b>14968</b>	2	0.0
<b>14969</b>	3	0.0
<b>14970</b>	4	0.0
<b>14971</b>	5	0.0
<b>14972</b>	6	0.0
<b>14973</b>	7	0.0
<b>14974</b>	8	0.0
<b>14975</b>	9	0.0
<b>14976</b>	10	0.0

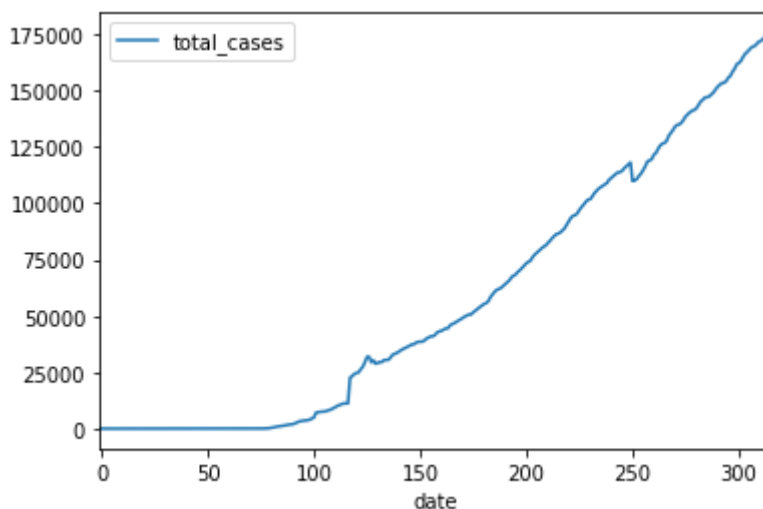
	date	total_cases
<b>14977</b>	11	0.0
<b>14978</b>	12	0.0
<b>14979</b>	13	0.0
<b>14980</b>	14	0.0
<b>14981</b>	15	0.0
<b>14982</b>	16	0.0
<b>14983</b>	17	0.0
<b>14984</b>	18	0.0
<b>14985</b>	19	0.0
<b>14986</b>	20	0.0
<b>14987</b>	21	0.0
<b>14988</b>	22	0.0
<b>14989</b>	23	0.0
<b>14990</b>	24	0.0
<b>14991</b>	25	0.0
<b>14992</b>	26	0.0
<b>14993</b>	27	0.0
<b>14994</b>	28	0.0
...	...	...
<b>15252</b>	286	147315.0
<b>15253</b>	287	148171.0
<b>15254</b>	288	149083.0
<b>15255</b>	289	150360.0
<b>15256</b>	290	151659.0
<b>15257</b>	291	152422.0
<b>15258</b>	292	153289.0
<b>15259</b>	293	153423.0
<b>15260</b>	294	154115.0
<b>15261</b>	295	155625.0
<b>15262</b>	296	156451.0
<b>15263</b>	297	158270.0
<b>15264</b>	298	159614.0
<b>15265</b>	299	161635.0
<b>15266</b>	300	162178.0
<b>15267</b>	301	163192.0
<b>15268</b>	302	164908.0

	date	total_cases
15269	303	166302.0
15270	304	167147.0
15271	305	168192.0
15272	306	169194.0
15273	307	169562.0
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 [4]: df.plot(x='date', y='total_cases')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x14dfaf75908>
```



Ahora podemos analizar los cuatro modelos que tomaré en el examen, que son la función lineal, polinómica, logística y la función exponencial. Cada modelo tiene tres parámetros, que se estimarán mediante un cálculo de ajuste de curva en los datos históricos.

## EL modelo lineal

La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en Machine Learning y en estadística. En su versión más sencilla, lo que haremos es «dibujar una recta» que nos indicará la tendencia de un conjunto de datos continuos.

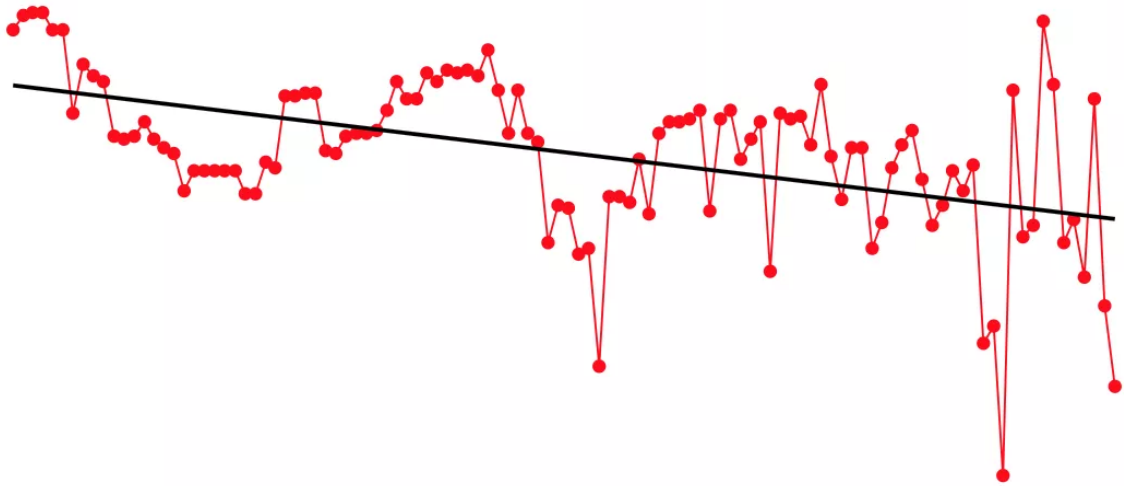
Recordemos rápidamente la fórmula de la recta:

$$Y = mX + b$$

Donde Y es el resultado, X es la variable, m la pendiente (o coeficiente) de la recta y b la constante o también conocida como el «punto de corte con el eje Y» en la gráfica (cuando X=0)

Ejemplo

## The development in Pizza prices in Denmark from 2009 to 2018



Recordemos que los algoritmos de Machine Learning Supervisados, aprenden por sí mismos y - en este caso- a obtener automáticamente esa «recta» que buscamos con la tendencia de predicción. Para hacerlo se mide el error con respecto a los puntos de entrada y el valor «Y» de salida real.

```
In [20]: 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)

# 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
```

De la ecuación de la recta  $y = mX + b$  nuestra pendiente «m» es el coeficiente y el término independiente «b»

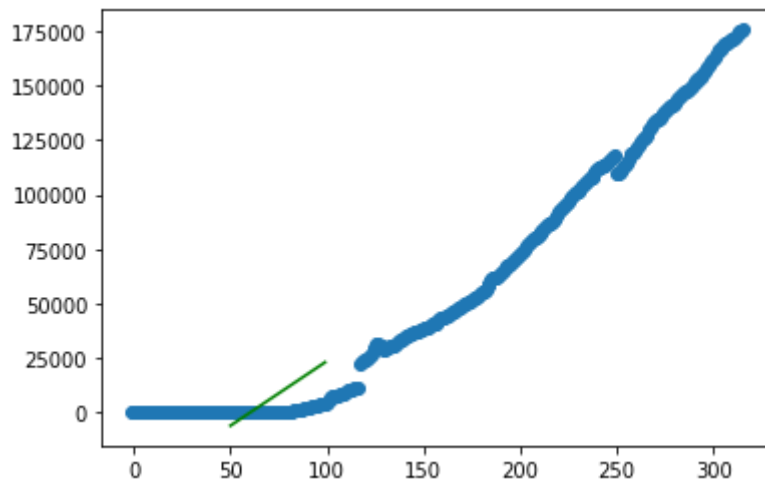
```
In [21]: #Vamos a comprobar:
# Quiero predecir cuántos "Casos" voy a obtener por en el día 100,
# según nuestro modelo, hacemos:
y_prediccion = regr.predict([[100]])
print('Total de casos', int(y_prediccion))
```

```
Total de casos 23672
```

```
In [22]: #Graficar
plt.scatter(x, y)
x_real = np.array(range(50, 100))
```

```
print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='green')
plt.show()
```

```
[50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
 98 99]
```



## El modelo logístico

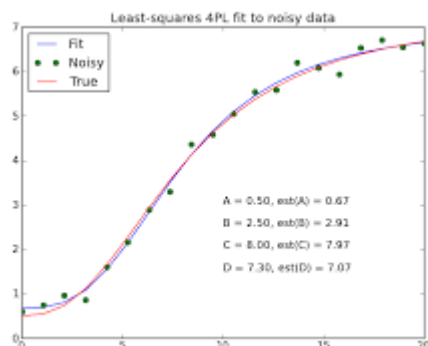
El modelo logístico se ha utilizado ampliamente para describir el crecimiento de una población. Una infección puede describirse como el crecimiento de la población de un agente patógeno, por lo que un modelo logístico parece razonable. La expresión más genérica de una función logística es:

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

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 [26]: def modelo_logistico(x,a,b):
          res=0
```



```

if x[1] == 0:
    x=1
else:
    res= a+b*np.log(x)
return res

```

```

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

```

```

(array([1., 1.]), array([[inf, inf],
                        [inf, inf]]))

```

## Graficas

```

In [27]: pred_x = list(range(min(x),max(x)+50)) # Predecir 50 días 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()

```

```

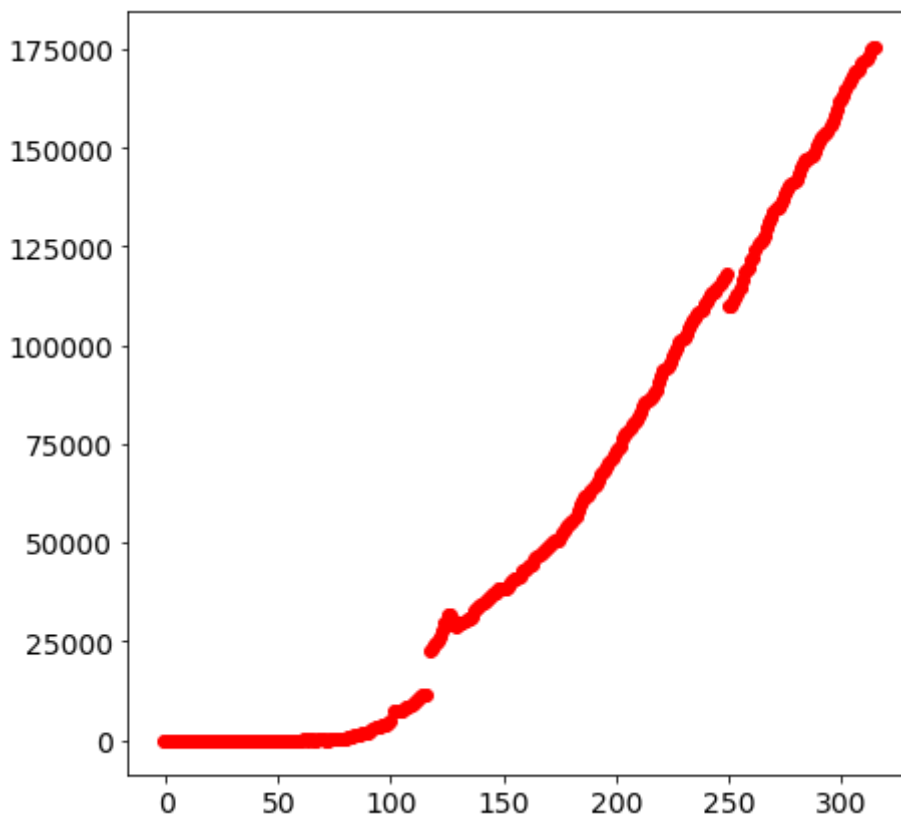
-----
TypeError                                Traceback (most recent call last)
<ipython-input-27-a9b5bfb0e5aa> in <module>
      5 plt.scatter(x,y,label="Datos Reales",color="red")
      6 # Predicted exponential curve
----> 7 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i 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

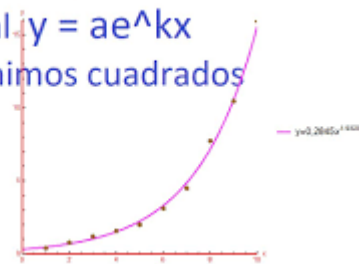
Mientras que el modelo logístico describe un crecimiento de infección que se detendrá en el futuro, el modelo exponencial describe un crecimiento de infección impararable . Por ejemplo, si un paciente infecta a 2 pacientes por día, después de 1 día tendremos 2 infecciones, 4 después

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

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 polinomica 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]:

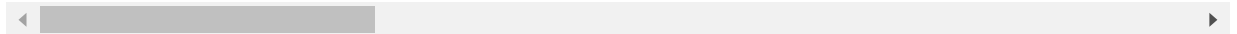
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_dea
<b>15026</b>	ECU	South America	Ecuador	2020-03-01	1.0	1.0	0.143	
<b>15027</b>	ECU	South America	Ecuador	2020-03-02	6.0	5.0	0.857	
<b>15028</b>	ECU	South America	Ecuador	2020-03-03	7.0	1.0	1.000	
<b>15030</b>	ECU	South America	Ecuador	2020-03-05	10.0	3.0	1.429	
<b>15031</b>	ECU	South America	Ecuador	2020-03-06	13.0	3.0	1.857	
<b>15034</b>	ECU	South America	Ecuador	2020-03-09	14.0	1.0	1.143	
<b>15035</b>	ECU	South America	Ecuador	2020-03-10	15.0	1.0	1.143	
<b>15036</b>	ECU	South America	Ecuador	2020-03-11	17.0	2.0	1.429	
<b>15039</b>	ECU	South America	Ecuador	2020-03-14	23.0	6.0	1.429	
<b>15040</b>	ECU	South America	Ecuador	2020-03-15	28.0	5.0	2.143	
<b>15041</b>	ECU	South America	Ecuador	2020-03-16	37.0	9.0	3.286	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
<b>15042</b>	ECU	South America	Ecuador	2020-03-17	58.0	21.0	6.143	
<b>15043</b>	ECU	South America	Ecuador	2020-03-18	111.0	53.0	13.429	
<b>15044</b>	ECU	South America	Ecuador	2020-03-19	168.0	57.0	21.571	
<b>15045</b>	ECU	South America	Ecuador	2020-03-20	199.0	31.0	26.000	
<b>15046</b>	ECU	South America	Ecuador	2020-03-21	426.0	227.0	57.571	
<b>15047</b>	ECU	South America	Ecuador	2020-03-22	532.0	106.0	72.000	
<b>15048</b>	ECU	South America	Ecuador	2020-03-23	789.0	257.0	107.429	1
<b>15049</b>	ECU	South America	Ecuador	2020-03-24	981.0	192.0	131.857	1
<b>15050</b>	ECU	South America	Ecuador	2020-03-25	1082.0	101.0	138.714	2
<b>15051</b>	ECU	South America	Ecuador	2020-03-26	1211.0	129.0	149.000	2
<b>15052</b>	ECU	South America	Ecuador	2020-03-27	1403.0	192.0	172.000	3
<b>15053</b>	ECU	South America	Ecuador	2020-03-28	1627.0	224.0	171.571	4
<b>15054</b>	ECU	South America	Ecuador	2020-03-29	1835.0	208.0	186.143	4
<b>15055</b>	ECU	South America	Ecuador	2020-03-30	1890.0	55.0	157.286	5
<b>15056</b>	ECU	South America	Ecuador	2020-03-31	1966.0	76.0	140.714	6
<b>15057</b>	ECU	South America	Ecuador	2020-04-01	2302.0	336.0	174.286	7
<b>15058</b>	ECU	South America	Ecuador	2020-04-02	2758.0	456.0	221.000	12
<b>15059</b>	ECU	South America	Ecuador	2020-04-03	3163.0	405.0	251.429	12
<b>15060</b>	ECU	South America	Ecuador	2020-04-04	3368.0	205.0	248.714	14
...	...	...	...	...	...	...	...	
<b>15252</b>	ECU	South America	Ecuador	2020-10-13	147315.0	282.0	853.714	1221
<b>15253</b>	ECU	South America	Ecuador	2020-10-14	148171.0	856.0	873.571	1223
<b>15254</b>	ECU	South America	Ecuador	2020-10-15	149083.0	912.0	793.143	1226

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
<b>15255</b>	ECU	South America	Ecuador	2020-10-16	150360.0	1277.0	759.286	1230
<b>15256</b>	ECU	South America	Ecuador	2020-10-17	151659.0	1299.0	830.143	1235
<b>15257</b>	ECU	South America	Ecuador	2020-10-18	152422.0	763.0	799.143	1237
<b>15258</b>	ECU	South America	Ecuador	2020-10-19	153289.0	867.0	893.714	1238
<b>15259</b>	ECU	South America	Ecuador	2020-10-20	153423.0	134.0	872.571	1239
<b>15260</b>	ECU	South America	Ecuador	2020-10-21	154115.0	692.0	849.143	1240
<b>15261</b>	ECU	South America	Ecuador	2020-10-22	155625.0	1510.0	934.571	1245
<b>15262</b>	ECU	South America	Ecuador	2020-10-23	156451.0	826.0	870.143	1250
<b>15263</b>	ECU	South America	Ecuador	2020-10-24	158270.0	1819.0	944.429	1252
<b>15264</b>	ECU	South America	Ecuador	2020-10-25	159614.0	1344.0	1027.429	1254
<b>15265</b>	ECU	South America	Ecuador	2020-10-26	161635.0	2021.0	1192.286	1255
<b>15266</b>	ECU	South America	Ecuador	2020-10-27	162178.0	543.0	1250.714	1257
<b>15267</b>	ECU	South America	Ecuador	2020-10-28	163192.0	1014.0	1296.714	1258
<b>15268</b>	ECU	South America	Ecuador	2020-10-29	164908.0	1716.0	1326.143	1260
<b>15269</b>	ECU	South America	Ecuador	2020-10-30	166302.0	1394.0	1407.286	1262
<b>15270</b>	ECU	South America	Ecuador	2020-10-31	167147.0	845.0	1268.143	1263
<b>15271</b>	ECU	South America	Ecuador	2020-11-01	168192.0	1045.0	1225.429	1267
<b>15272</b>	ECU	South America	Ecuador	2020-11-02	169194.0	1002.0	1079.857	1268
<b>15273</b>	ECU	South America	Ecuador	2020-11-03	169562.0	368.0	1054.857	1269
<b>15274</b>	ECU	South America	Ecuador	2020-11-04	170110.0	548.0	988.286	1269
<b>15275</b>	ECU	South America	Ecuador	2020-11-05	171433.0	1323.0	932.143	1270
<b>15276</b>	ECU	South America	Ecuador	2020-11-06	171783.0	350.0	783.000	1273
<b>15277</b>	ECU	South America	Ecuador	2020-11-07	172508.0	725.0	765.857	1276

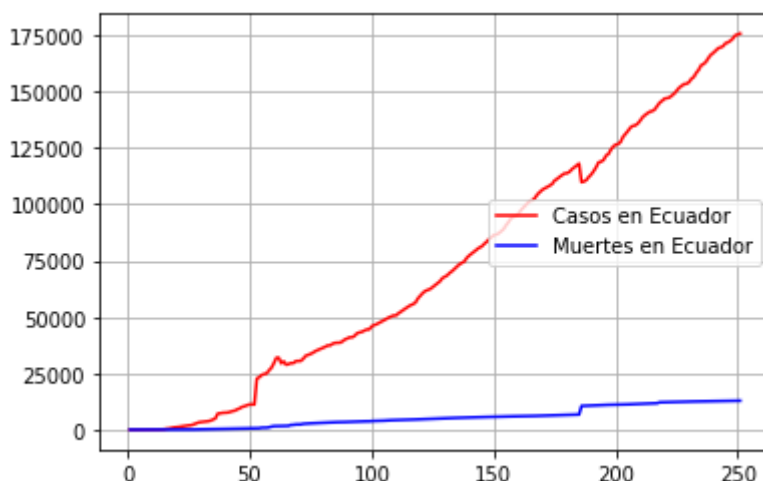
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_dea
<b>15278</b>	ECU	South America	Ecuador	2020-11-08	173486.0	978.0	756.286	1281
<b>15279</b>	ECU	South America	Ecuador	2020-11-09	174907.0	1421.0	816.143	1283
<b>15280</b>	ECU	South America	Ecuador	2020-11-10	175269.0	362.0	815.286	1283
<b>15281</b>	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.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/>

