



Regresion Total | Lineal

Materia:

Simulación

Docente:

Ing. Diego Quisi

Estudiante:

Ricardo Vinicio Jara Jara

EN BASE AL SIGUIENTE DOCUMENTO ACTUALIZAR LA BASE DE DATOS DEL COVID EN ECUADOR.

```
In [54]: # Importar las librerías para el analisis
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 [55]: # Actualizar los datos (URL)

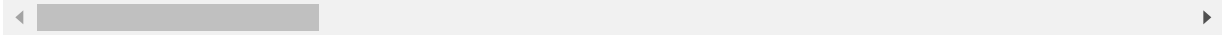
url = 'https://raw.githubusercontent.com/andrab/ecuacovid/master/datos_crudos/ecuacovid.csv'

df = pd.read_csv(url)
df
```

Out[55]:

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	n
0	129	129	0	106	1	
1	206	206	77	178	2	
2	273	273	67	236	2	
3	354	354	81	296	2	
4	762	762	408	651	2	
...
250	624030	603458	4335	48574	8708	
251	629093	608521	5063	48617	8729	
252	632453	611881	3360	48275	8750	
253	636428	615856	3975	47265	8787	
254	641542	620970	5114	45942	8804	

255 rows × 32 columns



Imprimos los resultados y agregamos el numero del dia

```
In [65]: #df = df[df['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuador
df = df.loc[:,['created_at','positivas_pcr']] #Selecciono las columnas de analisis
# Expresar las fechas en numero de dias desde el 01 Enero
FMT = '%d/%m/%Y'
date = df['created_at']
print('hola')
df['created_at'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("01/01/2020", FMT)).days)

df
```

hola

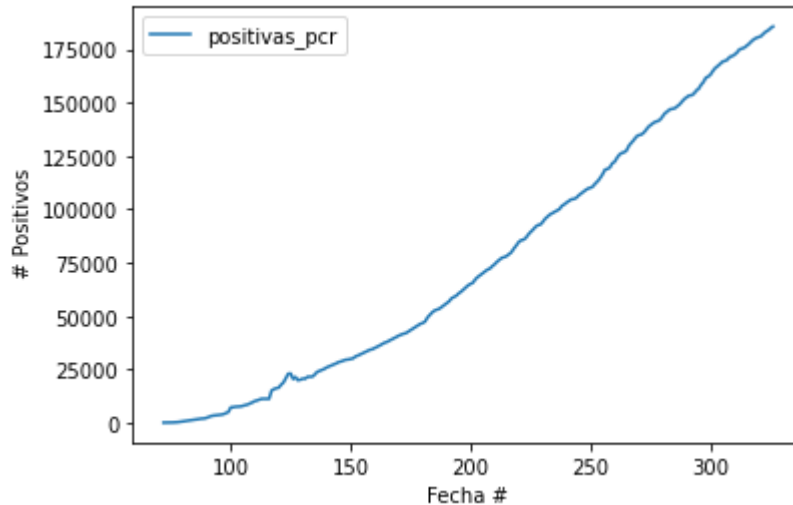
Out[65]:

	created_at	positivas_pcr
0	72	23
1	73	28
2	74	37
3	75	58
4	76	111
...
250	322	182250
251	323	183246
252	324	183840
253	325	184876
254	326	185643

255 rows × 2 columns

```
In [67]: ax = df.plot(x='created_at', y='positivas_pcr')
ax.set_xlabel("Fecha #")
ax.set_ylabel("# Positivos")
```

Out[67]: Text(0, 0.5, ' # Positivos')



```
In [72]: 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:
[778.4157929]
Independent term:
-79569.75847373396
```

Predecir cuantos casos voy a tener en la próximo semana.

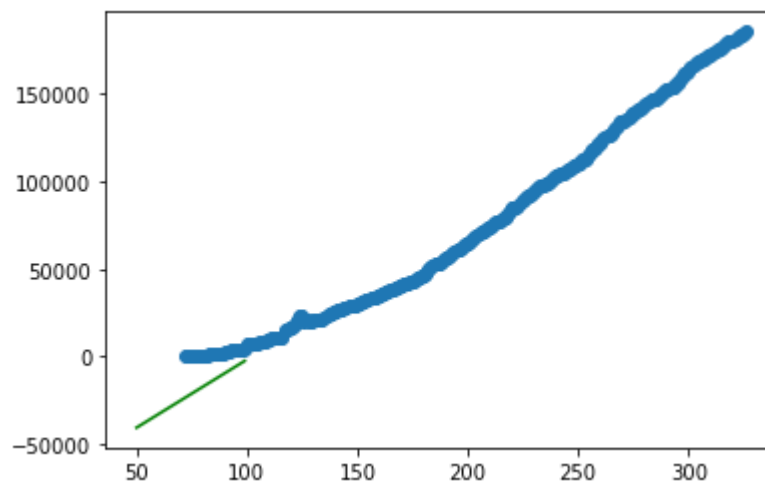
Para esto voy a sacar el ultimo número convertido en fecha y le voy a sumar 7 por los que los datos esta obtenidos en los días de la semana.

```
In [85]: #Vamos a comprobar:
# Quiero predecir cuántos "Casos" voy a obtener por en el día 100,
# según nuestro modelo, hacemos:
next_semana = df['created_at'].to_numpy()[-1]+7
y_prediccion = regr.predict([[next_semana]])
print(int(y_prediccion), 'Predicción de la siguiente semana ')
```

179642 Predicción de la siguiente semana

```
In [86]: #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]



Predecir cuántos casos voy a tener el próximo mes.

Para esto voy a sacar el último número convertido en fecha y le voy a sumar 30 por los que los datos están obtenidos en los días.

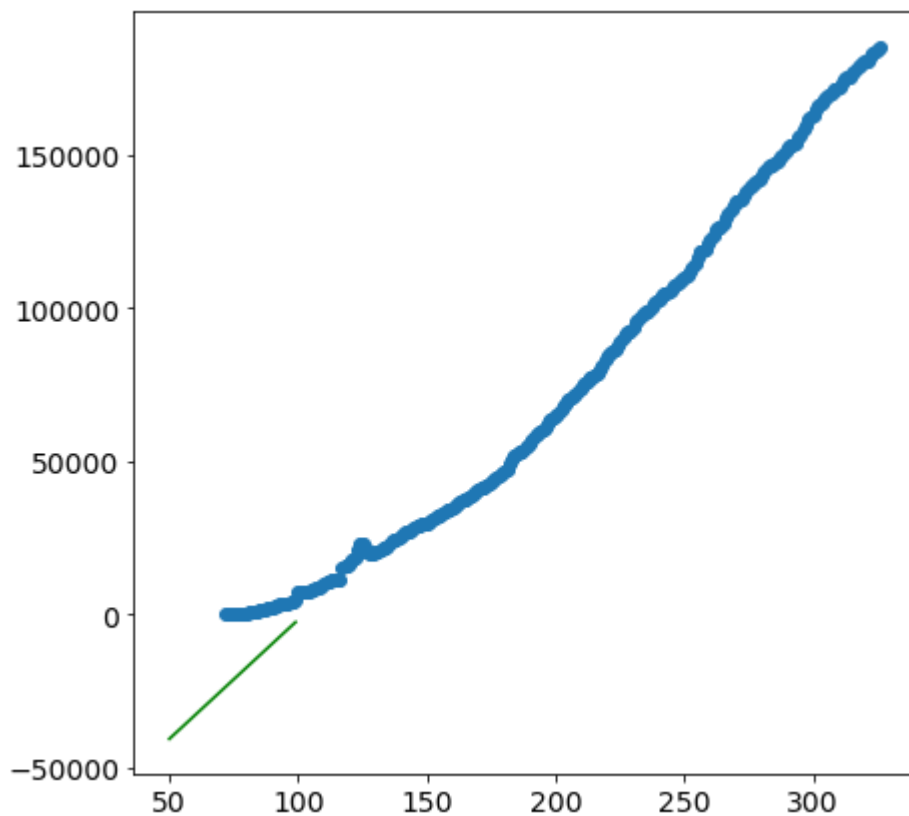
```

In [100]: #Vamos a comprobar:
# Quiero predecir cuántos "Casos" voy a obtener por en el día 100,
# según nuestro modelo, hacemos:
next_semana = df['created_at'].to_numpy()[-1]+30
y_prediccion = regr.predict([[next_semana]])
print(int(y_prediccion), 'Predicción de la siguiente mes ')
#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()

```

197546 Predicción de la siguiente mes

[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

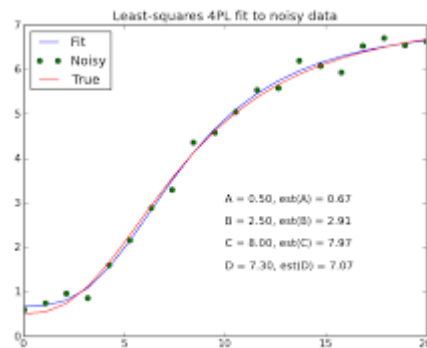
El modelo logístico se ha utilizado ampliamente para describir el crecimiento de una población. Una infección puede describirse como el crecimiento de la población de un agente patógeno, por lo que un modelo logístico parece razonable. La expresión más genérica de una función logística es:

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

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

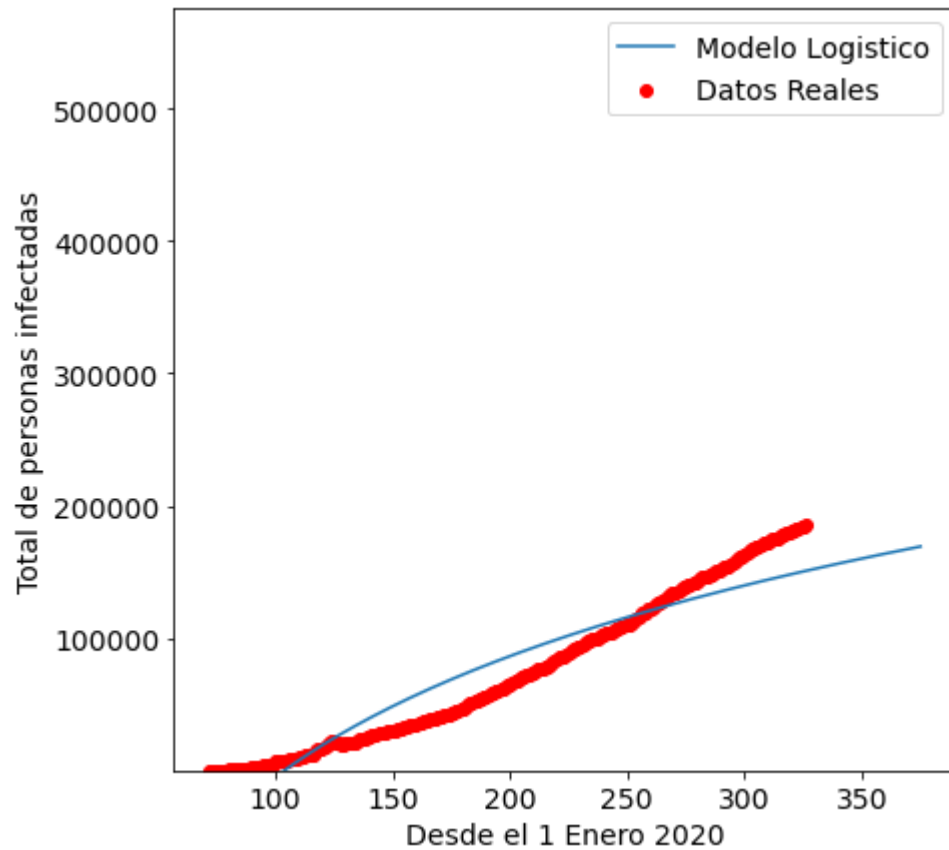


Definamos la función en Python y realicemos el procedimiento de ajuste de curva utilizado para el crecimiento logístico.

```
In [95]: 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 parámetros  
print(exp_fit)  
  
(array([-611486.54809066, 131719.53396173]), array([[ 2.42111223e+08, -4.61406554e+07],  
          [-4.61406554e+07, 8.84891540e+06]]))
```

Graficas

```
In [94]: 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], label="Modelo Logistico" )
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()
```



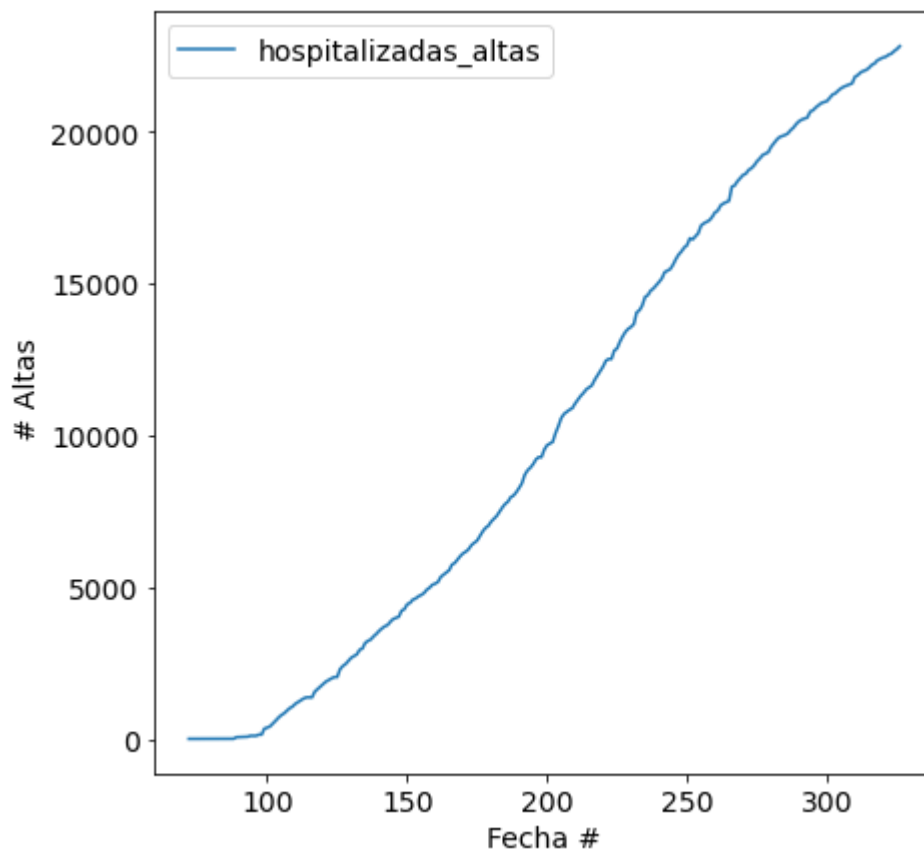
Evaluar Recuperados


```
In [99]: df = pd.read_csv(url)
df = df.loc[:,['created_at','hospitalizadas_altas']] #Selecciono las columnas
de analisis
# Expresar las fechas en numero de dias desde el 01 Enero
FMT = '%d/%m/%Y'
date = df['created_at']
print('hola')
df['created_at'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.s
trptime("01/01/2020", FMT)).days)

ax = df.plot(x='created_at', y='hospitalizadas_altas')
ax.set_xlabel("Fecha #")
ax.set_ylabel("# # Altas")
```

hola

```
Out[99]: Text(0, 0.5, ' # Altas')
```



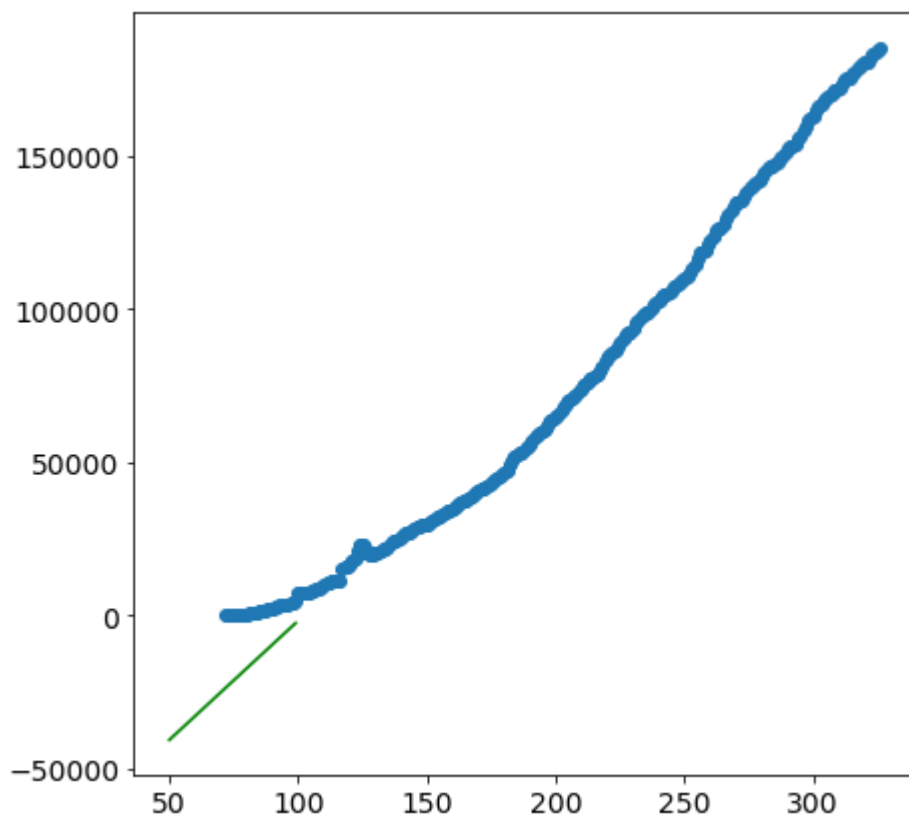
```

In [101]: next_semana = df['created_at'].to_numpy()[-1]+30
y_prediccion = regr.predict([[next_semana]])
print(int(y_prediccion), 'Predicción de la siguiente mes ')
#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()

```

197546 Predicción de la siguiente mes

[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]



Analisis

Con un análisis previo en vista de los datos obtenidos se puede ver que el incremento de los casos es el doble o a veces están triplicando día a día. Por otro lado se ve un crecimiento exponencial aunque no tan prolongado, es decir no existe una concordancia en el crecimiento de los casos positivos por covid recuperación de los casos

Conclusiones

Puedo concluir que la simulación de los datos de regresión podemos evaluar a futuro eso sí me pareció interesante en la parte que la fecha de ser transformada a un número y cuando yo quiero predecir necesito definir este número mayor a los existentes. Sumo la fecha en el que esté pasando a número sumó una cierta cantidad y puedo ver qué datos obtendré en las siguientes fechas o en el siguiente número de días que yo quiero ver

Criterio personal

- politico
En el criterio económico la pandemia perjudico a todo tipo de empresa afectando más a las pequeños y medianos empresarios que recién estaba emprendiendo o que tenían pérdidas en su desarrollo de empresa por otro lado tampoco las personas naturales con ciertas necesidades económicas tuvieron el apoyo por parte del gobierno
- economico
En el criterio político se vio mucho que en Ecuador como no existe un claro gobierno que controlen aspectos en estado crítico. Más bien se ve que estas decisiones son tomadas por empresarios o empresas que compran el gobierno o en que dichas empresas son de funcionarios público. Por lo que se pudo ver claramente que las decisiones en envases de medicamentos para pacientes infectados fueron repartidos en los funcionarios públicos.
- social
En el aspecto el aspecto social se pudo ver que el país estaba dividido ya que gran parte era consciente y respetaba las leyes del confinamiento para evitando que el virus se propague. Por otra parte, existía ciudades del país que les hacían como un quemeimportismo a la situación no respetaban nuestro teniendo así que esta ciudad de sean controladas por las fuerzas públicas del país

Referencias

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