

# Universidad Politecnica Salesiana

Mombre Jessica Ñauta.

Asignatura: Simulación.

## Covid-19 infección en Ecuador. Modelos probabilísticos

Implementacion de un modelo probabilistico de infección por el virus Covid-19

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

Se crea 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 probabilidad.

```
In [1]: # Importar Las Librerias 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 [2]: # Actualizar Los datos (URL)
url = 'https://covid.ourworldindata.org/data/ecdc/new_cases.csv'
df = pd.read_csv(url)
df
```

```
Out[2]:
```

	date	World	Afghanistan	Albania	Algeria	Andorra	Angola	Anguilla	Antigua and Barbuda	Argentina
0	2019-12-31	27	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
1	2020-01-01	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
2	2020-01-02	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
3	2020-01-03	17	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
4	2020-01-04	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
5	2020-01-05	15	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN

	date	World	Afghanistan	Albania	Algeria	Andorra	Angola	Anguilla	Antigua and Barbuda	Argentina
6	2020-01-06	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
7	2020-01-07	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
8	2020-01-08	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
9	2020-01-09	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
10	2020-01-10	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
11	2020-01-11	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
12	2020-01-12	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
13	2020-01-13	1	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
14	2020-01-14	1	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
15	2020-01-15	1	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
16	2020-01-16	0	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
17	2020-01-17	5	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
18	2020-01-18	17	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
19	2020-01-19	136	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
20	2020-01-20	20	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
21	2020-01-21	153	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
22	2020-01-22	142	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
23	2020-01-23	97	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
24	2020-01-24	266	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
25	2020-01-25	453	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
26	2020-01-26	673	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
27	2020-01-27	797	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN

	date	World	Afghanistan	Albania	Algeria	Andorra	Angola	Anguilla	Antigua and Barbuda	Argentina
<b>28</b>	2020-01-28	1767	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
<b>29</b>	2020-01-29	1480	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
<b>305</b>	2020-10-31	542158	157.0	319.0	319.0	98.0	195.0	0.0	3.0	13955.0
<b>306</b>	2020-11-01	457999	76.0	241.0	291.0	91.0	289.0	0.0	1.0	9745.0
<b>307</b>	2020-11-02	487232	132.0	327.0	330.0	69.0	247.0	0.0	0.0	6609.0
<b>308</b>	2020-11-03	470494	95.0	321.0	302.0	63.0	230.0	0.0	0.0	9598.0
<b>309</b>	2020-11-04	504999	86.0	381.0	405.0	22.0	193.0	0.0	2.0	12145.0
<b>310</b>	2020-11-05	569891	121.0	396.0	548.0	135.0	349.0	0.0	0.0	10652.0
<b>311</b>	2020-11-06	612387	40.0	421.0	642.0	90.0	236.0	0.0	0.0	11100.0
<b>312</b>	2020-11-07	606021	58.0	489.0	631.0	0.0	289.0	0.0	0.0	11786.0
<b>313</b>	2020-11-08	593274	126.0	495.0	581.0	184.0	121.0	0.0	1.0	8037.0
<b>314</b>	2020-11-09	528837	80.0	501.0	670.0	64.0	112.0	0.0	0.0	5331.0
<b>315</b>	2020-11-10	502288	224.0	525.0	642.0	54.0	98.0	0.0	0.0	8317.0
<b>316</b>	2020-11-11	566935	0.0	563.0	753.0	40.0	247.0	0.0	0.0	11976.0
<b>317</b>	2020-11-12	623511	146.0	507.0	811.0	90.0	136.0	0.0	0.0	10880.0
<b>318</b>	2020-11-13	647499	360.0	410.0	851.0	49.0	137.0	0.0	0.0	11162.0
<b>319</b>	2020-11-14	635829	66.0	490.0	867.0	109.0	104.0	0.0	2.0	11859.0
<b>320</b>	2020-11-15	594470	205.0	532.0	844.0	0.0	171.0	0.0	1.0	8468.0
<b>321</b>	2020-11-16	514538	163.0	597.0	860.0	147.0	146.0	0.0	0.0	5645.0
<b>322</b>	2020-11-17	544481	65.0	602.0	910.0	42.0	77.0	0.0	0.0	7895.0
<b>323</b>	2020-11-18	582979	383.0	694.0	1002.0	37.0	164.0	0.0	0.0	10621.0

	date	World	Afghanistan	Albania	Algeria	Andorra	Angola	Anguilla	Antigua and Barbuda	Argentina
<b>324</b>	2020-11-19	625770	0.0	711.0	1038.0	67.0	203.0	0.0	5.0	10332.0
<b>325</b>	2020-11-20	640970	282.0	786.0	1023.0	48.0	104.0	0.0	0.0	10110.0
<b>326</b>	2020-11-21	679758	232.0	836.0	1103.0	76.0	212.0	0.0	0.0	9592.0
<b>327</b>	2020-11-22	572043	154.0	737.0	1019.0	65.0	279.0	0.0	0.0	7143.0
<b>328</b>	2020-11-23	518858	252.0	565.0	1088.0	49.0	80.0	1.0	0.0	4181.0
<b>329</b>	2020-11-24	533527	246.0	795.0	1005.0	48.0	0.0	0.0	0.0	3998.0
<b>330</b>	2020-11-25	561704	185.0	744.0	1133.0	47.0	141.0	0.0	1.0	7434.0
<b>331</b>	2020-11-26	649666	200.0	644.0	1025.0	77.0	108.0	0.0	0.0	8593.0
<b>332</b>	2020-11-27	549900	0.0	656.0	1085.0	106.0	79.0	0.0	1.0	9043.0
<b>333</b>	2020-11-28	616139	214.0	645.0	NaN	76.0	NaN	0.0	0.0	7846.0
<b>334</b>	2020-11-29	552296	228.0	545.0	2102.0	60.0	266.0	0.0	0.0	6098.0

335 rows × 216 columns



Imprimos los resultados y agregamos el numero del día

```
In [3]: df = df.loc[:,['date','Ecuador']] #Selecciono las columnas de analisis
# Expresar las fechas en numero de días desde el 01 Enero
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-11-01", FMT)).days)
df
```

Out[3]:

	date	Ecuador
<b>0</b>	-1	0.0
<b>1</b>	0	0.0
<b>2</b>	1	0.0
<b>3</b>	2	0.0
<b>4</b>	3	0.0
<b>5</b>	4	0.0
<b>6</b>	5	0.0
<b>7</b>	6	0.0
<b>8</b>	7	0.0

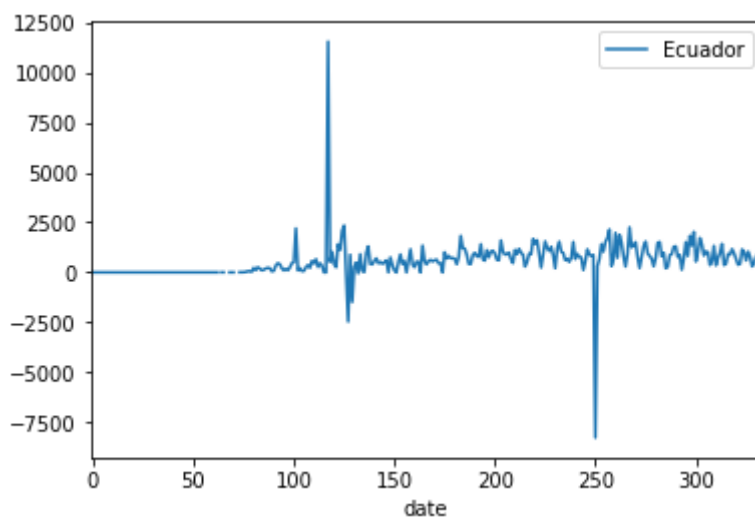
	date	Ecuador
<b>9</b>	8	0.0
<b>10</b>	9	0.0
<b>11</b>	10	0.0
<b>12</b>	11	0.0
<b>13</b>	12	0.0
<b>14</b>	13	0.0
<b>15</b>	14	0.0
<b>16</b>	15	0.0
<b>17</b>	16	0.0
<b>18</b>	17	0.0
<b>19</b>	18	0.0
<b>20</b>	19	0.0
<b>21</b>	20	0.0
<b>22</b>	21	0.0
<b>23</b>	22	0.0
<b>24</b>	23	0.0
<b>25</b>	24	0.0
<b>26</b>	25	0.0
<b>27</b>	26	0.0
<b>28</b>	27	0.0
<b>29</b>	28	0.0
...	...	...
<b>305</b>	304	845.0
<b>306</b>	305	1045.0
<b>307</b>	306	1002.0
<b>308</b>	307	368.0
<b>309</b>	308	548.0
<b>310</b>	309	1323.0
<b>311</b>	310	350.0
<b>312</b>	311	725.0
<b>313</b>	312	978.0
<b>314</b>	313	1421.0
<b>315</b>	314	362.0
<b>316</b>	315	442.0
<b>317</b>	316	919.0
<b>318</b>	317	883.0

	date	Ecuador
319	318	1161.0
320	319	953.0
321	320	668.0
322	321	381.0
323	322	428.0
324	323	1146.0
325	324	996.0
326	325	594.0
327	326	1036.0
328	327	767.0
329	328	301.0
330	329	492.0
331	330	794.0
332	331	908.0
333	332	1396.0
334	333	1375.0

335 rows × 2 columns

```
In [4]: df.plot(x='date', y='Ecuador')
```

```
Out[4]: <AxesSubplot:xlabel='date'>
```



Ahora podemos analizar un modelo probabilístico para el examen.

## El modelo basado en probabilidad

Para realizar una estimación del factor de crecimiento de los casos de Covid 19 en Ecuador calculamos la mediana, con esto obtenemos el valor medio de crecimiento de un conjunto de

datos, con esto podemos obtener un factor de crecimiento o tasa de crecimiento de los nuevos casos.

```
In [5]: filtro = df["Ecuador"][61:] # Filtro Los datos que se empezo a tener casos
#Obtenemos La mediana
media = filtro.mean()
mediana = filtro.median()
print("La mediana es: ", mediana)
print("La media es:", media)
```

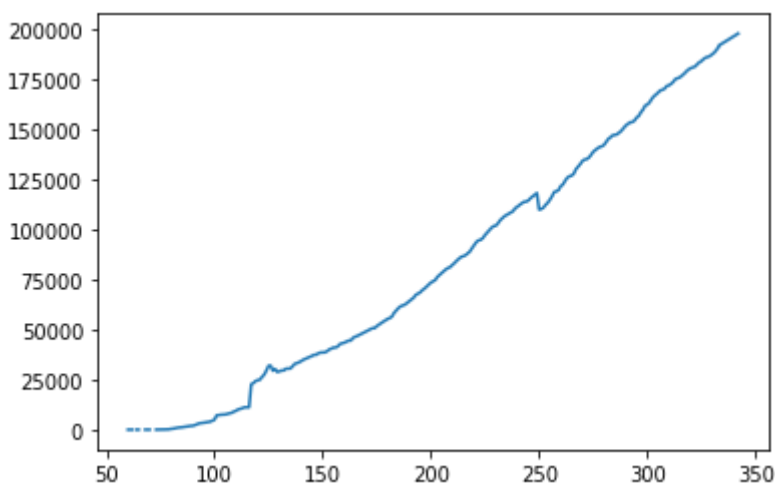
La mediana es: 670.0  
La media es: 709.6988847583643

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

```
In [6]: #Vamos a comprobar:
# según la media y la mediana podemos obtener la tasa de crecimiento y predecir su c
# Cargamos Los datos de total de casos
url = 'https://covid.ourworldindata.org/data/ecdc/total_cases.csv'
df_t = pd.read_csv(url)
FMT = '%Y-%m-%d'
date = df_t['date']
df_t['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2
df_t = df_t.loc[:, ['date', 'Ecuador']] #Selecciono las columnas de analisis
y = list(df_t.iloc[:, 1]) # Total casos
x = list(df_t.iloc[:, 0]) # Dias
#Realizamos un ejemplo de prediccion
prediccion_siguiente = int(y[-1] + mediana)
print(prediccion_siguiente)
```

191579

```
In [7]: # Quiero predecir cuántos "Casos" voy a obtener de aqui a 10 dias.
for i in range(x[-1], x[-1]+10):
    x.append(i)
    y.append(int(y[-1] + mediana))
valorProbabilistico=y[319]
plt.plot(x[61:], y[61:])
plt.show()
```



## Practica

1. Comparar el modelo de predicion matematico vs probabilidad.

2. Generar el SIR en base al modelo de probabilidad y obtener beta y gamma con una semana de prediccion.
3. Retroceder un semana y comparar el modelo matematico vs probabilidad vs reales. Solo cargan los datos para generar los modelos menos 7 días.

Puntos extras: Investiga sobre la correlacion de variables y aplicar el calculo en base a los datos del Ecuador.

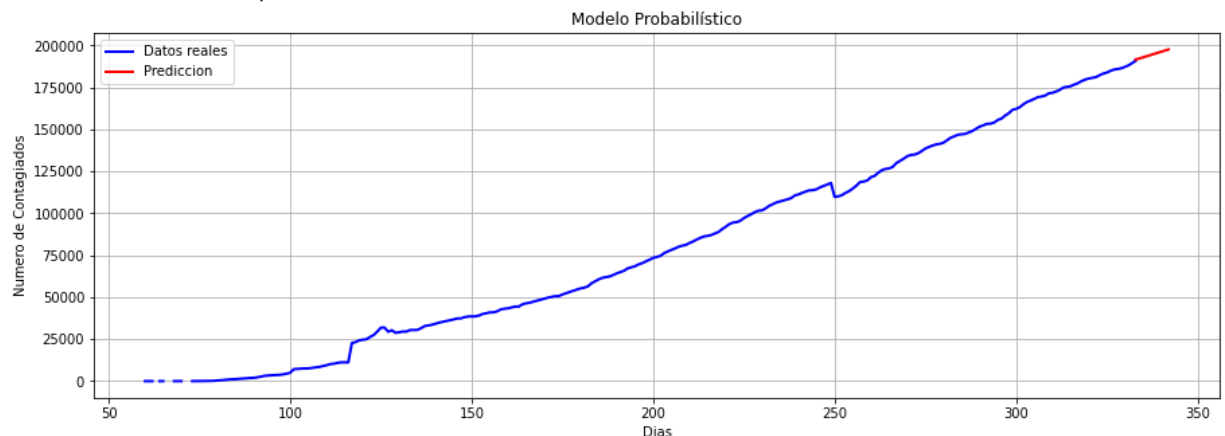
# 1. Comparar el modelo de prediccion matematico vs probabilidad

## Modelo Probabilistico

```
In [8]: # Implementar
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(x[:len(x)-9], y[:len(x)-9], 'b', alpha=1, lw = 2, label = 'Datos reales')
ax.plot(x[len(x)-10:], y[len(x)-10:], 'r', alpha=1, lw = 2, label = 'Prediccion')
ax.set_xlabel('Dias')
ax.set_ylabel('Numero de Contagiados')
ax.set_title("Modelo Probabilístico")
ax.legend()
ax.grid()
x_matematico = x[:]
y_matematico = y[:]

for i in range(x[-1], x[-1]+7):
    x.append(i)
    y.append(int(y[-1] + mediana))
print('Los casos en los proximos días serán: ', y[-1]+7,)
```

Los casos en los proximos días serán: 202306



## Modelo Polinomial

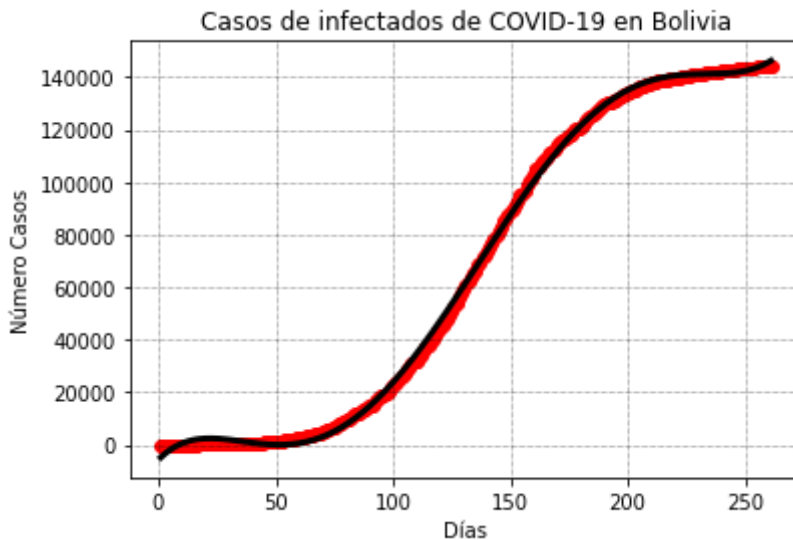
```
In [10]: #Implementar
df = pd.read_csv('owid-covid-data.csv').fillna(0)
ndf= df.loc[(df['location'] == 'Bolivia') & (df['total_cases'] != 0)]
ndf1=ndf[['date', 'total_cases']]
x=np.arange(1,len(ndf1)+1,1, dtype='float')
y=np.array(ndf1.values[:,1], dtype='float')
fun_pol = np.poly1d(np.polyfit(x, y, 5))
# print(fun_pol)
# print()
print('Los casos en los próximos 7 días serán: ', round(fun_pol(len(x)+7),5))
y_pred=fun_pol(x)
```



```
#Propiedades
plt.title('Casos de infectados de COVID-19 en Bolivia')
plt.xlabel('Días')
plt.ylabel('Número Casos')
plt.grid(color='black', linestyle='dotted', linewidth=0.5)

plt.scatter(x,y,label="Casos de Datos Reales",color="red")
plt.plot(x, y_pred, c='black',lw=3)
plt.show()
```

Los casos en los próximos 7 días serán: 151420.4857



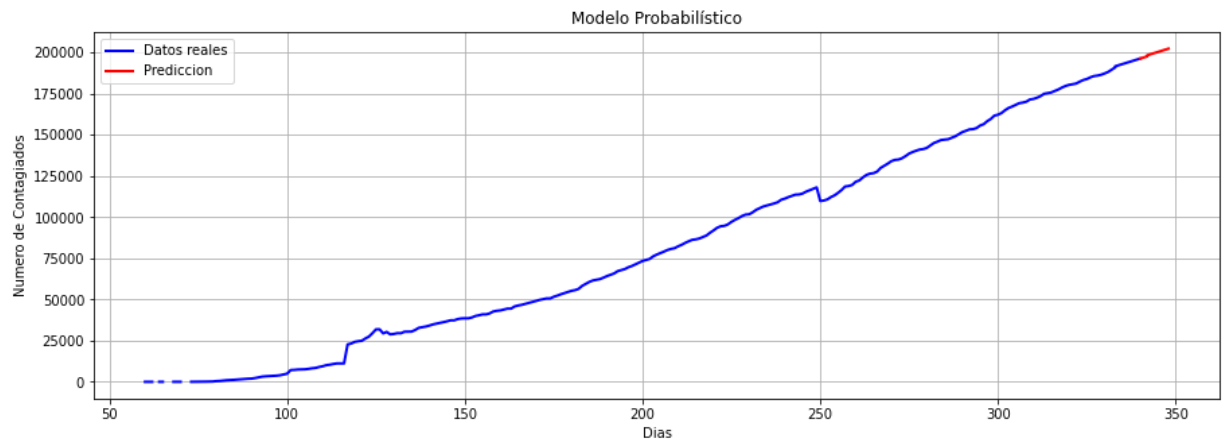
**3. Retroceder un semana y comparar el modelo matematico vs probabilidad vs reales. Solo cargan los datos para generar los modelos menos 7 dias.**

### Modelo Probabilistico

```
In [9]: # Implementar
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(x[:len(x)-9], y[:len(x)-9], 'b', alpha=1, lw = 2, label = 'Datos reales')
ax.plot(x[len(x)-10:], y[len(x)-10:], 'r', alpha=1, lw = 2, label = 'Prediccion')
ax.set_xlabel('Dias')
ax.set_ylabel('Numero de Contagiados')
ax.set_title("Modelo Probabilístico")
ax.legend()
ax.grid()
x_matematico = x[:]
y_matematico = y[:]

for i in range(x[-1], x[-1]-7):
    x.append(i)
    y.append(int(y[-1] + mediana))
print('Los casos en la semana pasada fueron: ', y[-1]-7,)
```

Los casos en la semana pasada fueron: 202292



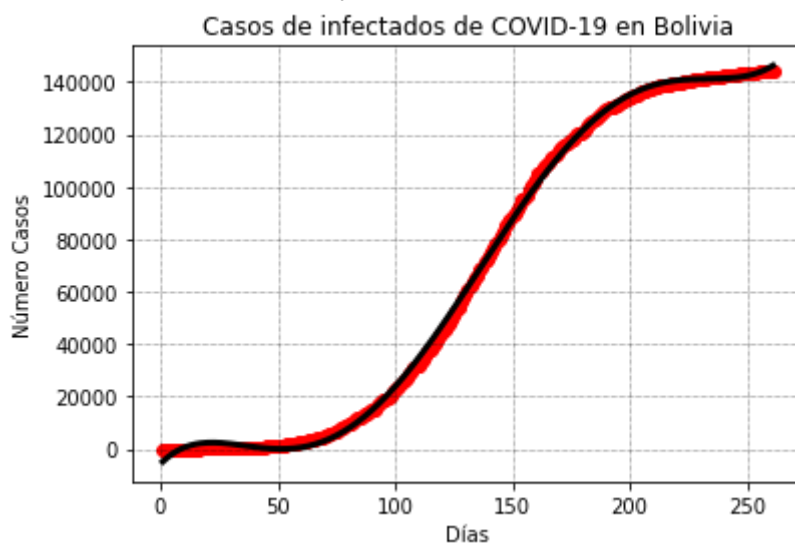
## Modelo Polinomial

```
In [11]: #Implementar
df = pd.read_csv('owid-covid-data.csv').fillna(0)
ndf= df.loc[(df['location'] == 'Bolivia') & (df['total_cases'] != 0)]
ndf1=ndf[['date', 'total_cases']]
x=np.arange(1,len(ndf1)+1,1, dtype='float')
y=np.array(ndf1.values[:,1], dtype='float')
fun_pol = np.poly1d(np.polyfit(x, y, 5))
# print(fun_pol)
# print()
print('Los casos en la semana pasada fueron: ', round(fun_pol(len(x)-7),5))
y_pred=fun_pol(x)

#Propiedades
plt.title('Casos de infectados de COVID-19 en Bolivia')
plt.xlabel('Días')
plt.ylabel('Número Casos')
plt.grid(color='black', linestyle='dotted', linewidth=0.5)

plt.scatter(x,y,label="Casos de Datos Reales",color="red")
plt.plot(x, y_pred, c='black',lw=3)
plt.show()
```

Los casos en la semana pasada fueron: 143378.67185



## Analisis

Mediante los métodos matemáticos y probabilísticos 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 los modelos matematicos y probabilistico 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\\_infe](https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infe)
- <https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>

