

LinealCovid

February 1, 2021

```
[3]: 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
```

```
[4]: # Actualizar los datos (URL)

#url = 'country_wise_latest.csv'
url = 'covid_19_clean_complete.csv'
df = pd.read_csv(url)
df
```

```
[4]:
```

	Province/State	Country/Region	Lat	Long	Date \
0	NaN	Afghanistan	33.939110	67.709953	2020-01-22
1	NaN	Albania	41.153300	20.168300	2020-01-22
2	NaN	Algeria	28.033900	1.659600	2020-01-22
3	NaN	Andorra	42.506300	1.521800	2020-01-22
4	NaN	Angola	-11.202700	17.873900	2020-01-22
...
49063	NaN	Sao Tome and Principe	0.186400	6.613100	2020-07-27
49064	NaN	Yemen	15.552727	48.516388	2020-07-27
49065	NaN	Comoros	-11.645500	43.333300	2020-07-27
49066	NaN	Tajikistan	38.861000	71.276100	2020-07-27
49067	NaN	Lesotho	-29.610000	28.233600	2020-07-27

	Confirmed	Deaths	Recovered	Active	WHO Region
0	0	0	0	0	Eastern Mediterranean
1	0	0	0	0	Europe
2	0	0	0	0	Africa
3	0	0	0	0	Europe
4	0	0	0	0	Africa
...
49063	865	14	734	117	Africa

49064	1691	483	833	375	Eastern Mediterranean
49065	354	7	328	19	Africa
49066	7235	60	6028	1147	Europe
49067	505	12	128	365	Africa

[49068 rows x 10 columns]

0.1 Listamos la fecha, los contagiados y los recueprados de Ecuador

```
[5]: df = df[df['Country/Region'].isin(['Ecuador'])] #Filtro la Informacion solo
      ↪para Ecuador
df = df.loc[:,['Date','Confirmed', 'Recovered']] #Selecciono las columnas de
      ↪analasis
# 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
```

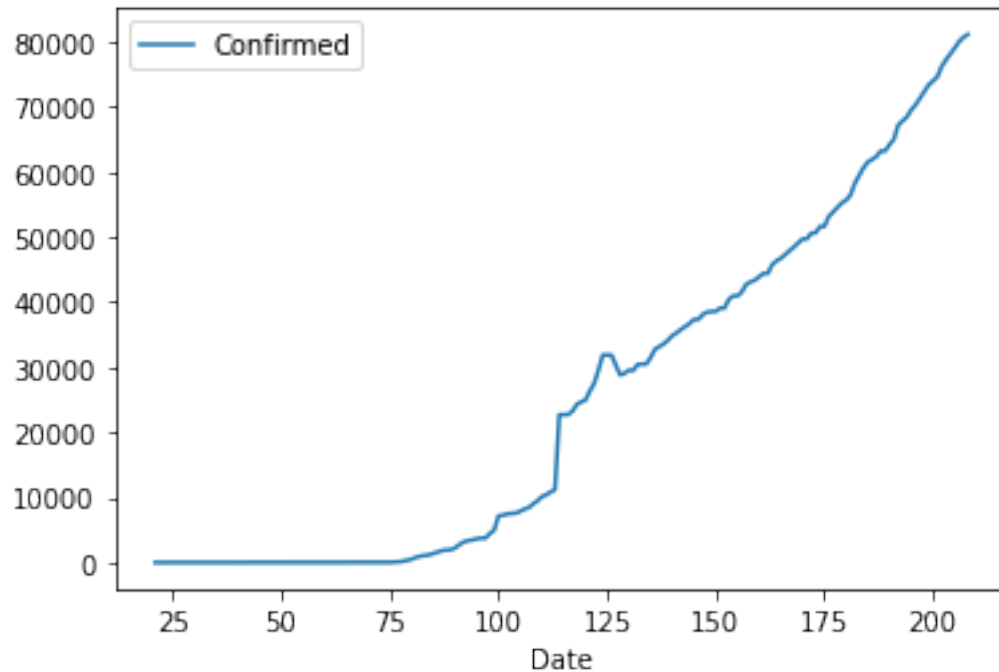
```
[5]:      Date  Confirmed  Recovered
95      21           0           0
356     22           0           0
617     23           0           0
878     24           0           0
1139    25           0           0
...
47858   204       78148       33455
48119   205       79049       34544
48380   206       80036       34544
48641   207       80694       34896
48902   208       81161       34896
```

[188 rows x 3 columns]

1 Se hace el proceso de los contagiados

```
[6]: df.plot(x='Date', y='Confirmed')
```

```
[6]: <AxesSubplot:xlabel='Date'>
```



1.1 Se entrena para la regresión

```
[8]: x = list(df.iloc[:, 0]) # Fecha
y = list(df.iloc[:, 1]) # Numero de casos
z = list(df.iloc[:, 2]) # Numero de recuperados
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()

# Entrenamos nuestros modelos
regr.fit(np.array(x).reshape(-1, 1), y)

#regr.fit(np.array(x).reshape(-1, 1), z)

# 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:

[451.62530905]

Independent term:

-26825.480865331312

1.2 Predecimos

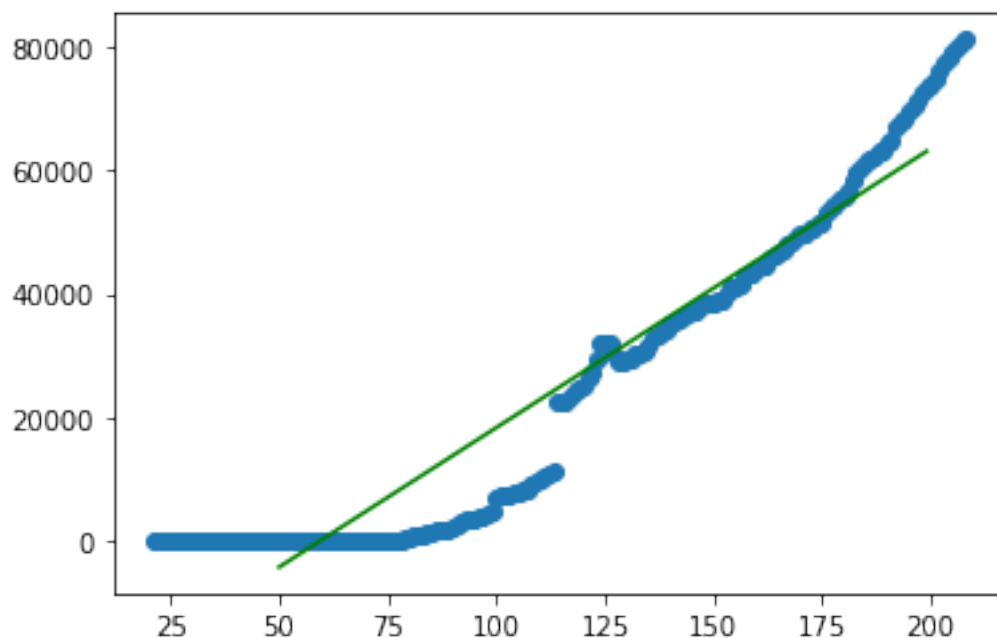
```
[9]: #Vamos a comprobar:
# Quiero predecir cuántos "Casos" voy a obtener por en el día 330,
# según nuestro modelo, hacemos:
y_prediccion = regr.predict([[330]])
print('Se predice que para el mes de Noviembre el número de personas_
→contagiadas será de:')
print(int(y_prediccion))
```

Se predice que para el mes de Noviembre el número de personas contagiadas será de:

122210

```
[11]: #Graficar
plt.scatter(x, y)
x_real = np.array(range(50, 200))
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  99 100 101 102 103
104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157
158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193
194 195 196 197 198 199]
```

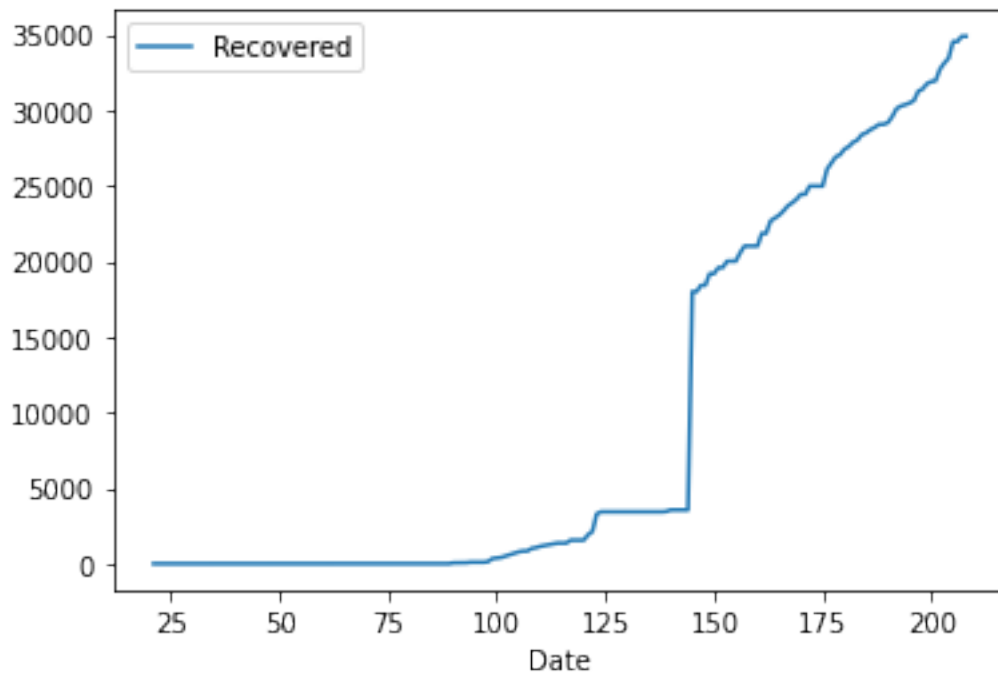


2 Ahora el mismo proceso pero con los recuperados

2.1 Graficamos la ecuación de los recuperados con respecto al tiempo

```
[12]: df.plot(x='Date', y='Recovered')
```

```
[12]: <AxesSubplot:xlabel='Date'>
```



2.2 Se entrena la regresión

```
[14]: x = list(df.iloc[:, 0]) # Fecha
      z = list(df.iloc[:, 2]) # Numero de recuperados
      # Creamos el objeto de Regresión Lineal
      regr = linear_model.LinearRegression()

      regr.fit(np.array(x).reshape(-1, 1), z)

      # 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:

[199.83446299]

Independent term:

-13454.397076432126

2.3 Predecimos cantos recuperados habrá para el día 330

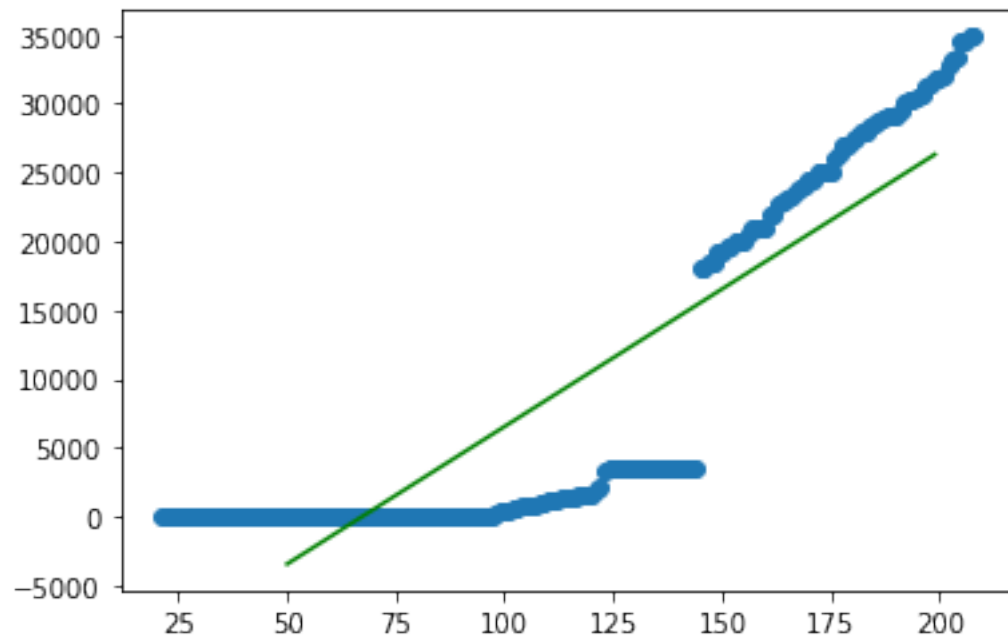
```
[15]: y_prediccion = regr.predict([[330]])
print('Se predice que para el mes de Noviembre el número de personas_
↪recupearadas será de:')
print(int(y_prediccion))
```

Se predice que para el mes de Noviembre el número de personas recupearadas será de:

52490

```
[16]: plt.scatter(x, z)
x_real = np.array(range(50, 200))
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  99 100 101 102 103
104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157
158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193
194 195 196 197 198 199]
```



[]: