

In [178]:



```

1  # Importar Las Librerias para el analisis
2  import pandas as pd
3  import numpy as np
4  from datetime import datetime, timedelta
5  from sklearn.metrics import mean_squared_error
6  from scipy.optimize import curve_fit
7  from scipy.optimize import fsolve
8  from sklearn import linear_model
9  import matplotlib.pyplot as plt
10 %matplotlib inline
11
12 url = 'C:/Users/ADMINX/Downloads/Compressed/ecuacovid-master/ecuacovid-master/datos_cru
13
14 df = pd.read_csv(url)
15 df

```

Out[178]:

	fecha	dosis_total	primera_dosis	segunda_dosis
0	21/01/2021	0	0	0
1	22/01/2021	108	108	0
2	27/01/2021	2982	2982	0
3	04/02/2021	6228	6228	0
4	17/02/2021	8190	6228	1962
5	24/02/2021	24492	20784	3708
6	01/03/2021	42114	35886	6228
7	04/03/2021	59316	53088	6228
8	05/03/2021	71148	64920	6228
9	08/03/2021	74472	68244	6228
10	09/03/2021	75258	69030	6228
11	11/03/2021	95915	89687	6228
12	12/03/2021	123176	116948	6228
13	13/03/2021	139359	119222	20137
14	15/03/2021	141191	121054	20137
15	21/03/2021	178970	140765	38205
16	23/03/2021	182261	143614	38647
17	24/03/2021	191179	152526	38653
18	26/03/2021	230770	172413	58357
19	27/03/2021	235000	174642	60358
20	29/03/2021	244866	182329	62537
21	01/04/2021	283106	204902	78204
22	04/04/2021	301069	211720	89349
23	05/04/2021	335093	228504	106589

	fecha	dosis_total	primera_dosis	segunda_dosis
24	06/04/2021	356783	244159	112624
25	08/04/2021	363255	250631	112624
26	14/04/2021	480962	338180	142782
27	15/04/2021	485132	338180	146952
28	16/04/2021	514151	354019	160132
29	17/04/2021	545132	377199	167933
30	18/04/2021	554369	384093	170276
31	19/04/2021	577711	401871	175840
32	20/04/2021	601229	421937	179292
33	21/04/2021	643702	457403	186299
34	22/04/2021	675510	486524	188986
35	23/04/2021	711204	514854	196350
36	24/04/2021	732717	532367	200350
37	25/04/2021	743937	541420	202517
38	26/04/2021	765489	555265	210224
39	27/04/2021	816175	595699	220476
40	28/04/2021	861393	633421	227972
41	29/04/2021	920865	691000	229865
42	30/04/2021	987452	748021	239431
43	01/05/2021	1036794	791822	244972
44	02/05/2021	1067472	821960	245512
45	04/05/2021	1141262	889218	252044
46	05/05/2021	1182085	924539	257546
47	06/05/2021	1215676	953238	262438
48	07/05/2021	1245822	981620	264202

Generar graficas y reportes del total de personas vacunadas.

In [25]:

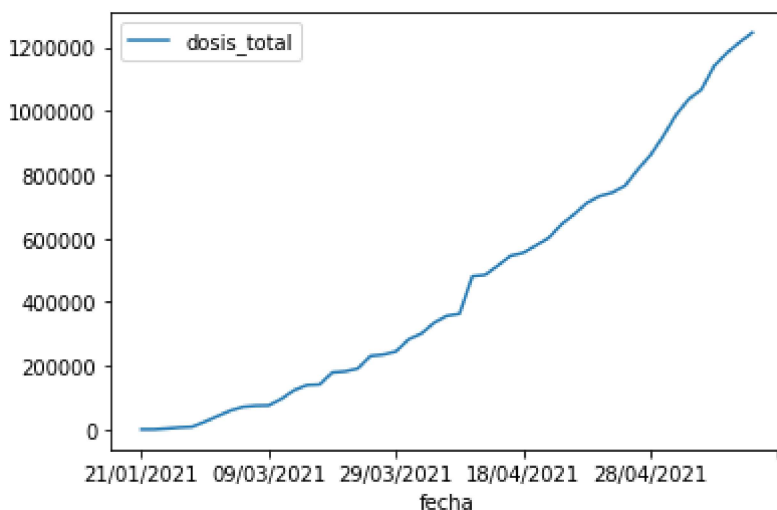
```
1 df_t = pd.read_csv(url,header = None)
2 FMT = '%Y-%m-%d'
3 df_t.columns = ['fecha','dosis_total','primera_dosis', 'segunda_dosis']
4 date = df_t['fecha'][1:]
5 filtro = df_t['dosis_total'][1:] # Filtro los datos que se empezo a tener casos
6 #Obtenemos La mediana
7 media = filtro.mean()
8 mediana = filtro.median()
9 print(mediana)
10 print(media)
11 df.plot(x = 'fecha', y='dosis_total')
12
```

356783.0

2.2101686181263315e+275

Out[25]:

<matplotlib.axes._subplots.AxesSubplot at 0x20c7d25f248>



Generar grafico de pie por fabricante de la vacuna.

In [28]:



```
1 url = 'C:/Users/ADMINX/Downloads/Compressed/ecuacovid-master/ecuacovid-master/datos_cru
2 df = pd.read_csv(url)
3 df
```

Out[28]:

	vaccine	total	arrived_at
0	Pfizer/BioNTech	8190	20/01/2021
1	Pfizer/BioNTech	16380	17/02/2021
2	Pfizer/BioNTech	17550	24/02/2021
3	Pfizer/BioNTech	31590	03/03/2021
4	Sinovac	20000	06/03/2021
5	Pfizer/BioNTech	73710	10/03/2021
6	Oxford/AstraZeneca	84000	17/03/2021
7	Pfizer/BioNTech	62010	17/03/2021
8	Pfizer/BioNTech	65520	24/03/2021
9	Pfizer/BioNTech	66690	31/03/2021
10	Pfizer/BioNTech	53820	05/04/2021
11	Sinovac	300000	07/04/2021
12	Sinovac	700000	10/04/2021
13	Pfizer/BioNTech	53820	14/04/2021
14	Pfizer/BioNTech	54990	21/04/2021
15	Oxford/AstraZeneca	336000	24/04/2021
16	Pfizer/BioNTech	54990	28/04/2021
17	Pfizer/BioNTech	100620	04/05/2021

In [33]:

```

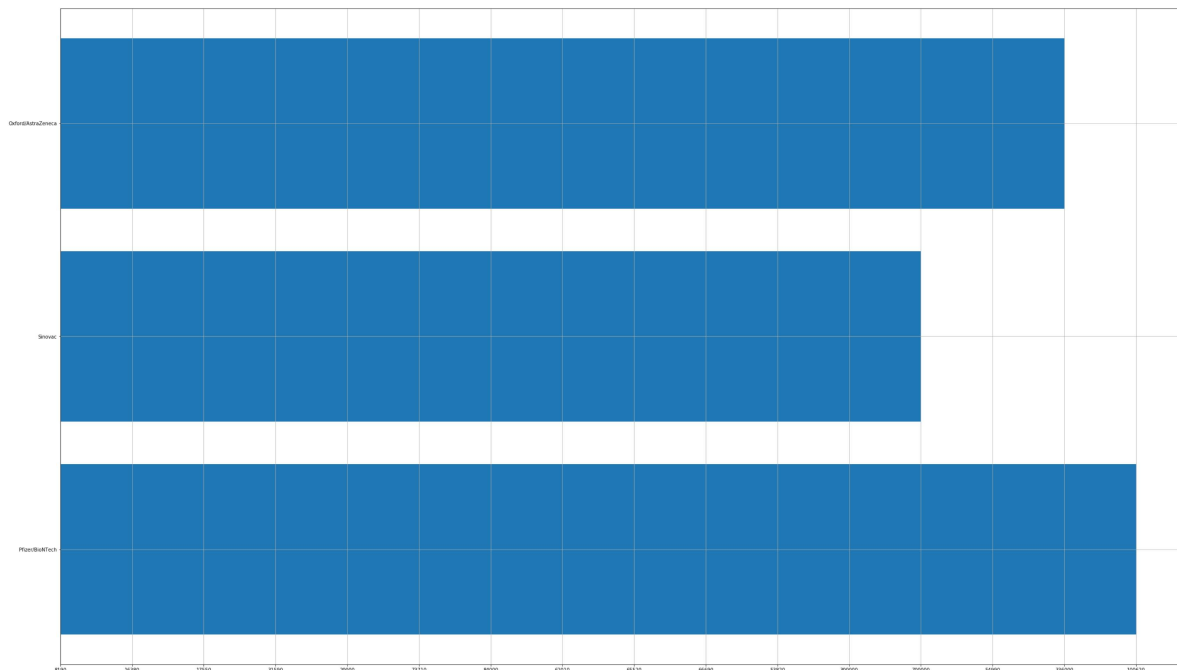
1 df_fa = pd.read_csv(url, header = None)
2 df_fa.columns = ['vaccine', 'total', 'arrived_at']
3 a = np.array(list(set(df_fa['vaccine'][1:])))
4 print(a)
5
6 ox = 0
7 si = 0
8 pf = 0
9 for i, j in zip(df_fa['vaccine'][1:], df_fa['total'][1:]):
10     if i == a[0]:
11         si = si + int(j)
12     elif i == a[1]:
13         pf = pf + int(j)
14     elif i == a[2]:
15         ox = ox + int(j)
16
17 b = np.array([si, pf, ox])
18 print(b)
19 plt.barh(df_fa['vaccine'][1:], df_fa['total'][1:])
20 plt.gcf().set_size_inches(42, 25)
21 plt.grid()
22 plt.show()

```

```

['Oxford/AstraZeneca' 'Sinovac' 'Pfizer/BioNTech']
[ 420000 1020000  659880]

```



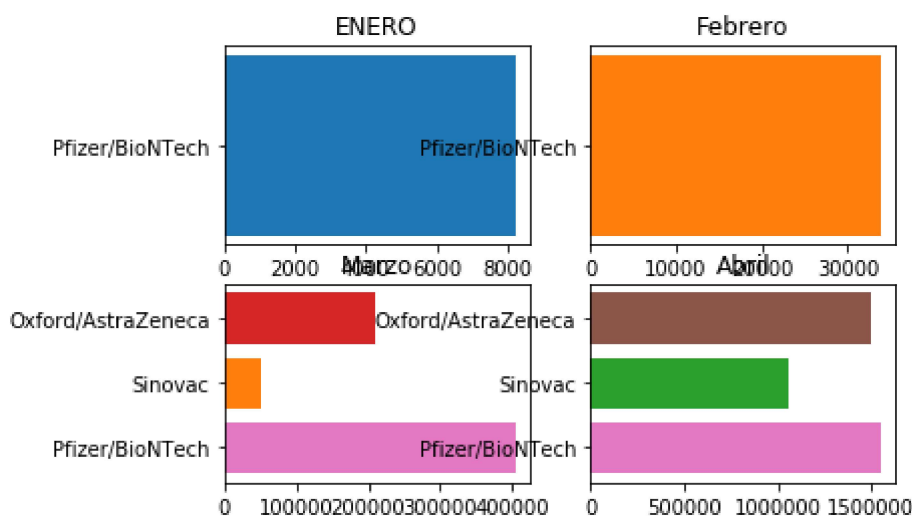
Generar histogramas de vacunas por mes de llega y fabricante.

In [177]:

```

1 from ipykernel import kernelapp as app
2 lista1=[]
3 fig1 = plt.subplot(2, 2, 1)
4 fig2 = plt.subplot(2, 2, 2)
5 fig3 = plt.subplot(2, 2, 3)
6 fig4 = plt.subplot(2, 2, 4)
7 for i in range(1, 32):
8     lista1.append(0)
9 con = 0
10 con1 = 0
11 con2=0
12 con3=0
13 con4=0
14
15 for i, j, k in zip(df_fa['vaccine'][1:], df_fa['total'][1:], df_fa['arrived_at'][1:]):
16     fecha_dt = datetime.strptime(k, '%d/%m/%Y')
17     if((i == a[0] or i == a[1] or i == a[2]) and (fecha_dt.month == 1)):
18         con =con+int(j)
19         fig1.barh(i,con)
20         fig1.set_title('ENERO')
21     elif((i == a[0] or i == a[1] or i == a[2]) and (fecha_dt.month == 2)):
22         con1 =con1+int(j)
23         fig2.barh(i,con1)
24         fig2.set_title('Febrero')
25     elif((i == a[0] or i == a[1] or i == a[2]) and (fecha_dt.month == 3)):
26         con2 =con2+int(j)
27         fig3.barh(i,con2)
28         fig3.set_title('Marzo')
29     elif((i == a[0] or i == a[1] or i == a[2]) and (fecha_dt.month == 4)):
30         con3 =con3+int(j)
31         fig4.barh(i,con3)
32         fig4.set_title('Abril')
33 #     elif((i == a[0] or i == a[1] or i == a[2]) and (fecha_dt.month == 5)):
34 #         con4 =con4+int(j)
35 #         plt.barh(i,con4)
36 #         plt.title('Mayo')

```



Generar un reporte parametrizado que pueda ingresar los datos de las fechas inicio y fin para obtener la información de las graficas vistas en el primer punto.

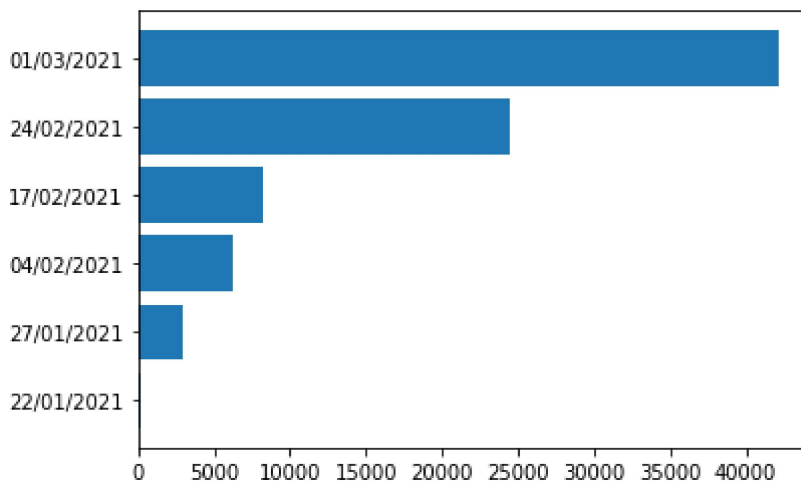
In [197]:

```
1 df_planva = pd.read_csv(url)
2 lisx=[]
3 lisy=[]
4 df_planva.columns = ['fecha', 'dosis_total', 'primera_dosis', 'segunda_dosis']
5 #SE INGRESA LAS FECHAS INICIO Y FIN
6 feini=input("ingrese fecha de inicio")
7 fefin=input("ingrese fecha de fin")
8 for d,i, j, k in zip(df_planva['fecha'][1:],df_planva['dosis_total'][1:], df_planva['pr
9     fecha_dt = datetime.strptime(d, '%d/%m/%Y')
10    if(datetime.strptime(feini,'%d/%m/%Y')<=datetime.strptime(d, '%d/%m/%Y') and datetime
11        lisx.append(d)
12        lisy.append(i)
13 plt.barh(lisx,lisy)
```

ingrese fecha de inicio01/01/2021
ingrese fecha de fin01/03/2021

Out[197]:

<BarContainer object of 6 artists>



Generar un modelo matemático de predicción para regresión lineal, exponencial, polinómico y logarítmico, del procesos de vacunación en base al numero actual de vacunados (1 y 2 dosis) y a la llegada de nuevas vacunas

In [186]:



```
1 url = 'C:/Users/ADMINX/Downloads/Compressed/ecuacovid-master/ecuacovid-master/datos_cru
2
3 #Vamos a comprobar:
4 # según la media y la mediana podemos obtener la tasa de crecimiento y predecir su comp
5 # Cargamos los datos de total de casos
6 df_t = pd.read_csv(url)
7 FMT = '%Y-%m-%d'
8 date = df_t['date']
9 df_t['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020
10 df_t = df_t.loc[:,['date', 'Ecuador']] #Selecciono las columnas de analisis
11 y = list(df_t.iloc[:, 1]) # Total casos
12 x = list(df_t.iloc[:, 0]) # Dias
13 #Realizamos un ejemplo de prediccion
14 prediccion_siguiente = int(y[-1] + mediana)
15 print(prediccion_siguiente)
```

Desarrollar y generar un proceso de comparación con al menos cuatro países (2. Latinoamérica, 1. E.E.U.U./Canada, 1. Europa

In [204]:

```

1 url_v = 'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccin
2 df_vacum = pd.read_csv(url_v, header = None).fillna(0)
3 df_vacum.columns = ['location', 'iso_code', 'date', 'total_vaccinations', 'people_vacci
4 lista_fechas = []
5 lista_tot = []
6 lista_fechasbr = []
7 lista_totbr = []
8 for i, j, k in zip(df_vacum['location'], df_vacum['date'], df_vacum['people_fully_vacci
9 # print(i)
10 if i == 'Ecuador':
11     lista_fechas.append(j)
12     lista_tot.append(k)
13 elif i == 'Peru':
14     lista_fechasbr.append(j)
15     lista_totbr.append(k)
16 dat = np.array(lista_fechas)
17 tot = np.array(lista_tot, dtype='float')
18 x = np.arange(1, len(tot) + 1, 1)
19 dat_br = np.array(lista_fechasbr)
20 tot_br = np.array(lista_totbr, dtype='float')
21 xbr = np.arange(1, len(tot_br) + 1, 1)
22 from sklearn import linear_model
23 print("-----ECUADOR-----")
24 regr = linear_model.LinearRegression()
25 regr.fit(np.array(x).reshape(-1, 1), tot)
26 print('Coefficients: \n', regr.coef_)
27 print('Independent term: \n', regr.intercept_)
28 print("-----PERU-----")
29 regrbr = linear_model.LinearRegression()
30 regrbr.fit(np.array(xbr).reshape(-1, 1), tot_br)
31 print('Coefficients: \n', regrbr.coef_)
32 print('Independent term: \n', regrbr.intercept_)
33 #SE GRAFICA
34 fig, (ax1, ax2) = plt.subplots(1, 2, sharex='col', sharey='row',
35                               gridspec_kw={'hspace': 0, 'wspace': 0}, figsize=(20,20))
36 fig.suptitle('COMPARACION ECUADOR VS. PERU EN VACUNADOS')
37 ax1.scatter(x, tot, color='yellow')
38 ax1.set_title('ECUADOR')
39 x_real = np.array(range(50, 100))
40 ax2.scatter(xbr, tot_br, color='green')
41 ax2.set_title('PERU')
42

```

-----ECUADOR-----

Coefficients:

[2057.478812]

Independent term:

-59189.493800539065

-----PERU-----

Coefficients:

[9357.03840125]

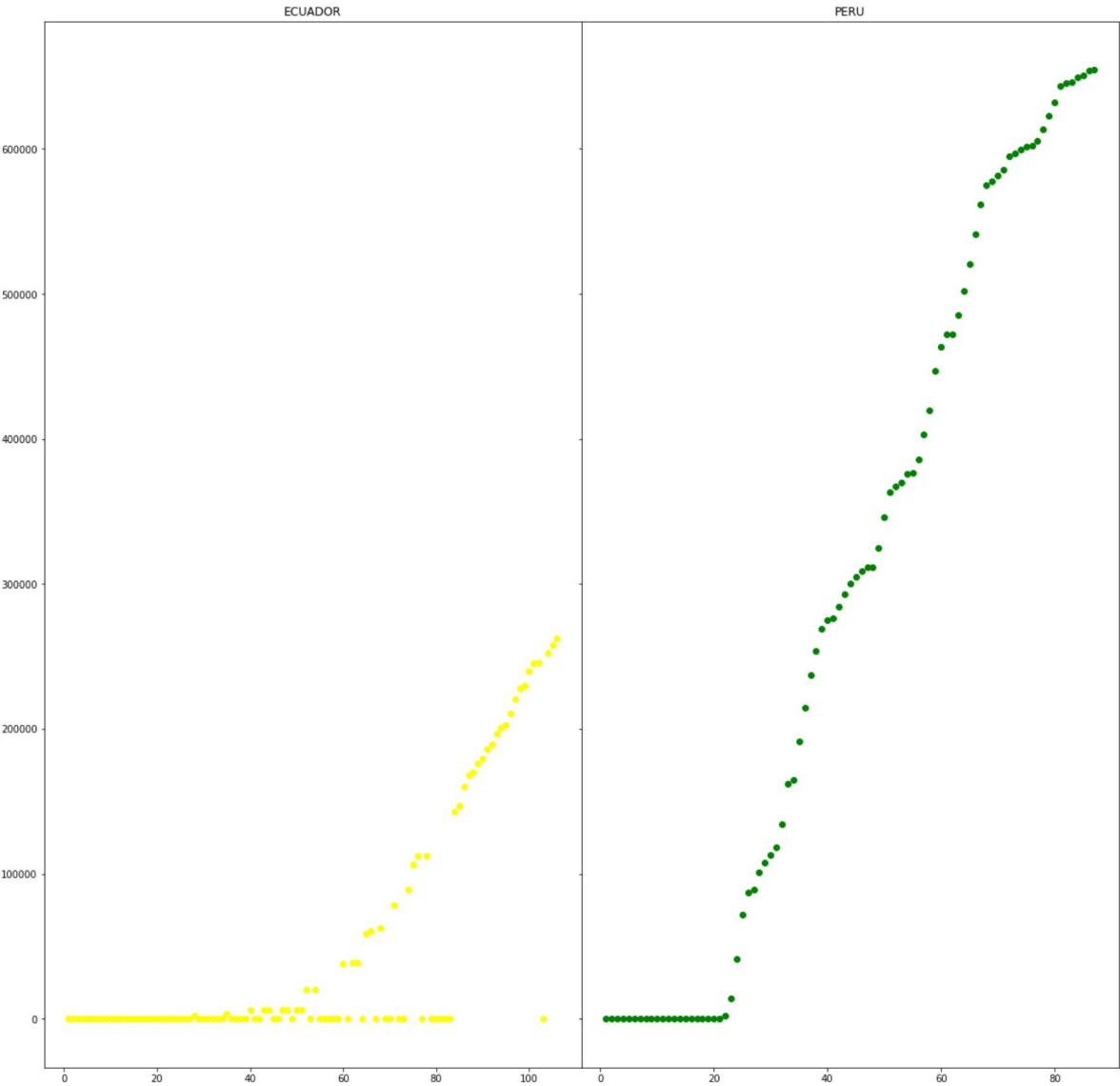
Independent term:

-117858.25287356327

Out[204]:

Text(0.5, 1.0, 'PERU')

COMPARACION ECUADOR VS. PERU EN VACUNADOS



In [205]:

```

1 url_v = 'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccin
2 df_vacum = pd.read_csv(url_v, header = None).fillna(0)
3 df_vacum.columns = ['location', 'iso_code', 'date', 'total_vaccinations', 'people_vacci
4 lista_fechas = []
5 lista_tot = []
6 lista_fechasbr = []
7 lista_totbr = []
8 for i, j, k in zip(df_vacum['location'], df_vacum['date'], df_vacum['people_fully_vacci
9 # print(i)
10 if i == 'Ecuador':
11     lista_fechas.append(j)
12     lista_tot.append(k)
13 elif i == 'Colombia':
14     lista_fechasbr.append(j)
15     lista_totbr.append(k)
16 dat = np.array(lista_fechas)
17 tot = np.array(lista_tot, dtype='float')
18 x = np.arange(1, len(tot) + 1, 1)
19 dat_br = np.array(lista_fechasbr)
20 tot_br = np.array(lista_totbr, dtype='float')
21 xbr = np.arange(1, len(tot_br) + 1, 1)
22 from sklearn import linear_model
23 print("-----ECUADOR-----")
24 regr = linear_model.LinearRegression()
25 regr.fit(np.array(x).reshape(-1, 1), tot)
26 print('Coefficients: \n', regr.coef_)
27 print('Independent term: \n', regr.intercept_)
28 print("-----COLOMBIA-----")
29 regrbr = linear_model.LinearRegression()
30 regrbr.fit(np.array(xbr).reshape(-1, 1), tot_br)
31 print('Coefficients: \n', regrbr.coef_)
32 print('Independent term: \n', regrbr.intercept_)
33 #SE GRAFICA
34 fig, (ax1, ax2) = plt.subplots(1, 2, sharex='col', sharey='row',
35                               gridspec_kw={'hspace': 0, 'wspace': 0}, figsize=(20,20))
36 fig.suptitle('COMPARACION ECUADOR VS. COLOMBIA EN VACUNADOS')
37 ax1.scatter(x, tot, color='yellow')
38 ax1.set_title('ECUADOR')
39 x_real = np.array(range(50, 100))
40 ax2.scatter(xbr, tot_br, color='green')
41 ax2.set_title('COLOMBIA')

```

```

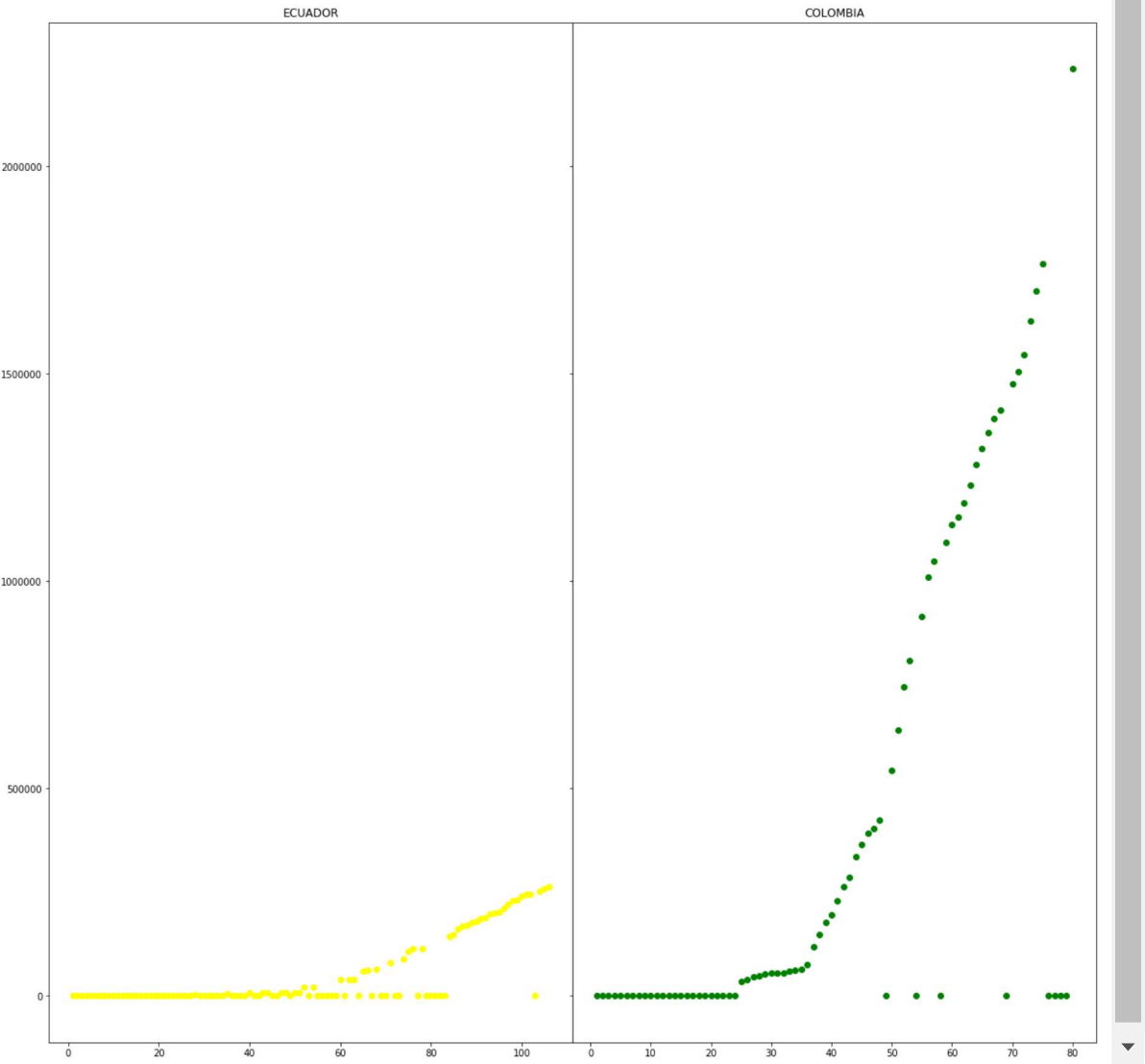
-----ECUADOR-----
Coefficients:
[2057.478812]
Independent term:
-59189.493800539065
-----COLOMBIA-----
Coefficients:
[18072.47528129]
Independent term:
-305725.71139240486

```

Out[205]:

Text(0.5, 1.0, 'COLOMBIA')

COMPARACION ECUADOR VS. COLOMBIA EN VACUNADOS



In [208]:

```

1 url_v = 'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccin
2 df_vacum = pd.read_csv(url_v, header = None).fillna(0)
3 df_vacum.columns = ['location', 'iso_code', 'date', 'total_vaccinations', 'people_vacci
4 lista_fechas = []
5 lista_tot = []
6 lista_fechasbr = []
7 lista_totbr = []
8 for i, j, k in zip(df_vacum['location'], df_vacum['date'], df_vacum['people_fully_vacci
9 # print(i)
10 if i == 'Ecuador':
11     lista_fechas.append(j)
12     lista_tot.append(k)
13 elif i == 'Spain':
14     lista_fechasbr.append(j)
15     lista_totbr.append(k)
16 dat = np.array(lista_fechas)
17 tot = np.array(lista_tot, dtype='float')
18 x = np.arange(1, len(tot) + 1, 1)
19 dat_br = np.array(lista_fechasbr)
20 tot_br = np.array(lista_totbr, dtype='float')
21 xbr = np.arange(1, len(tot_br) + 1, 1)
22 from sklearn import linear_model
23 print("-----ECUADOR-----")
24 regr = linear_model.LinearRegression()
25 regr.fit(np.array(x).reshape(-1, 1), tot)
26 print('Coefficients: \n', regr.coef_)
27 print('Independent term: \n', regr.intercept_)
28 print("-----ESPAÑA-----")
29 regrbr = linear_model.LinearRegression()
30 regrbr.fit(np.array(xbr).reshape(-1, 1), tot_br)
31 print('Coefficients: \n', regrbr.coef_)
32 print('Independent term: \n', regrbr.intercept_)
33 #SE GRAFICA
34 fig, (ax1, ax2) = plt.subplots(1, 2, sharex='col', sharey='row',
35                               gridspec_kw={'hspace': 0, 'wspace': 0}, figsize=(20,20))
36 fig.suptitle('COMPARACION ECUADOR VS. ESPAÑA EN VACUNADOS')
37 ax1.scatter(x, tot, color='yellow')
38 ax1.set_title('ECUADOR')
39 x_real = np.array(range(50, 100))
40 ax2.scatter(xbr, tot_br, color='green')
41 ax2.set_title('ESPAÑA')

```

-----ECUADOR-----

Coefficients:

[2057.478812]

Independent term:

-59189.493800539065

-----ESPAÑA-----

Coefficients:

[31477.07133276]

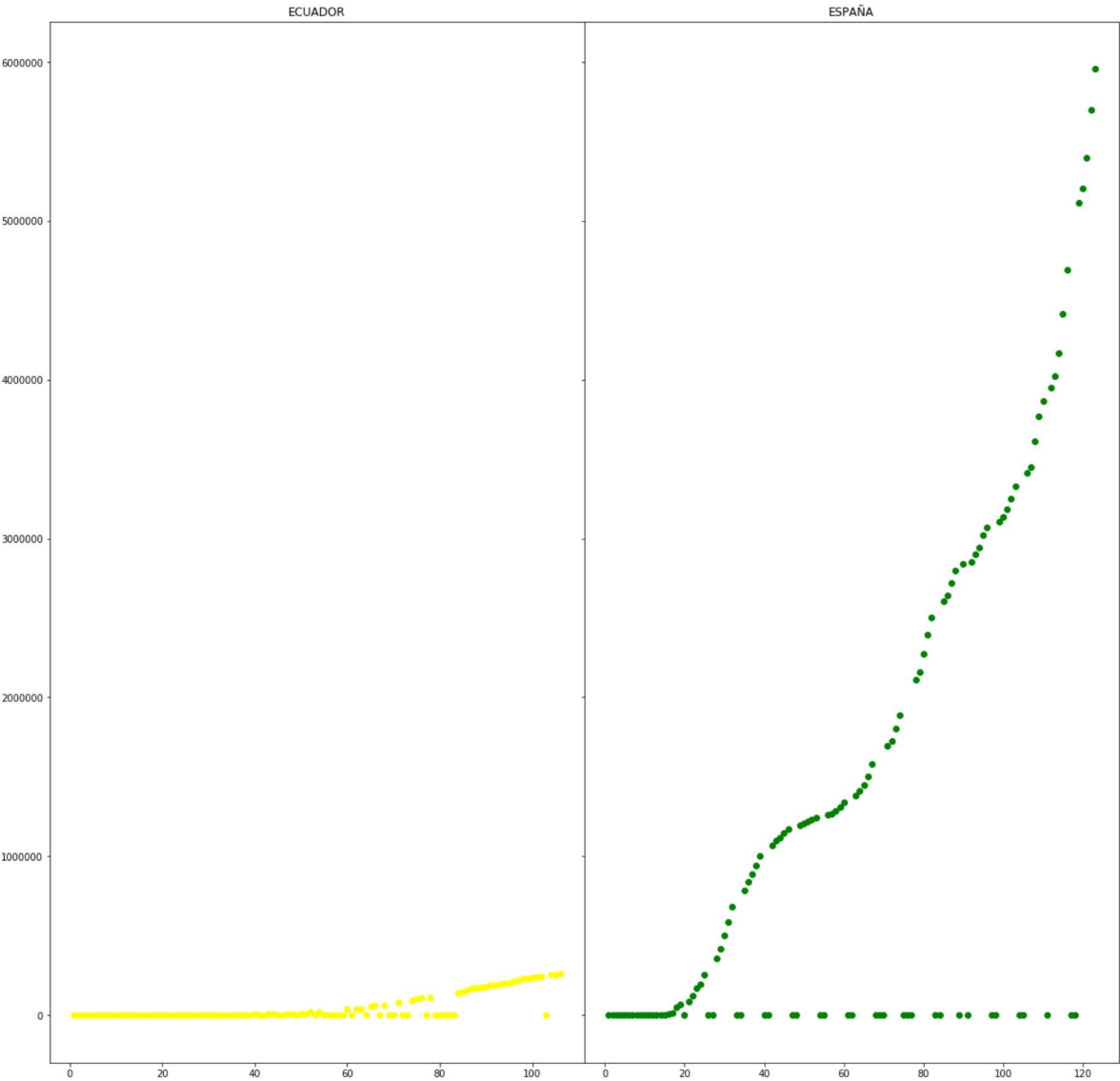
Independent term:

-625140.5852325731

Out[208]:

Text(0.5, 1.0, 'ESPAÑA')

COMPARACION ECUADOR VS. ESPAÑA EN VACUNADOS



In [210]:

```

1 url_v = 'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccin
2 df_vacum = pd.read_csv(url_v, header = None).fillna(0)
3 df_vacum.columns = ['location', 'iso_code', 'date', 'total_vaccinations', 'people_vacci
4 lista_fechas = []
5 lista_tot = []
6 lista_fechasbr = []
7 lista_totbr = []
8 for i, j, k in zip(df_vacum['location'], df_vacum['date'], df_vacum['people_fully_vacci
9 # print(i)
10 if i == 'Ecuador':
11     lista_fechas.append(j)
12     lista_tot.append(k)
13 elif i == 'India':
14     lista_fechasbr.append(j)
15     lista_totbr.append(k)
16 dat = np.array(lista_fechas)
17 tot = np.array(lista_tot, dtype='float')
18 x = np.arange(1, len(tot) + 1, 1)
19 dat_br = np.array(lista_fechasbr)
20 tot_br = np.array(lista_totbr, dtype='float')
21 xbr = np.arange(1, len(tot_br) + 1, 1)
22 from sklearn import linear_model
23 print("-----ECUADOR-----")
24 regr = linear_model.LinearRegression()
25 regr.fit(np.array(x).reshape(-1, 1), tot)
26 print('Coefficients: \n', regr.coef_)
27 print('Independent term: \n', regr.intercept_)
28 print("-----INDIA-----")
29 regrbr = linear_model.LinearRegression()
30 regrbr.fit(np.array(xbr).reshape(-1, 1), tot_br)
31 print('Coefficients: \n', regrbr.coef_)
32 print('Independent term: \n', regrbr.intercept_)
33 #SE GRAFICA
34 fig, (ax1, ax2) = plt.subplots(1, 2, sharex='col', sharey='row',
35                               gridspec_kw={'hspace': 0, 'wspace': 0}, figsize=(20,20))
36 fig.suptitle('COMPARACION ECUADOR VS. INDIA EN VACUNADOS')
37 ax1.scatter(x, tot, color='yellow')
38 ax1.set_title('ECUADOR')
39 x_real = np.array(range(50, 100))
40 ax2.scatter(xbr, tot_br, color='green')
41 ax2.set_title('INDIA')

```

-----ECUADOR-----

Coefficients:

[2057.478812]

Independent term:

-59189.493800539065

-----INDIA-----

Coefficients:

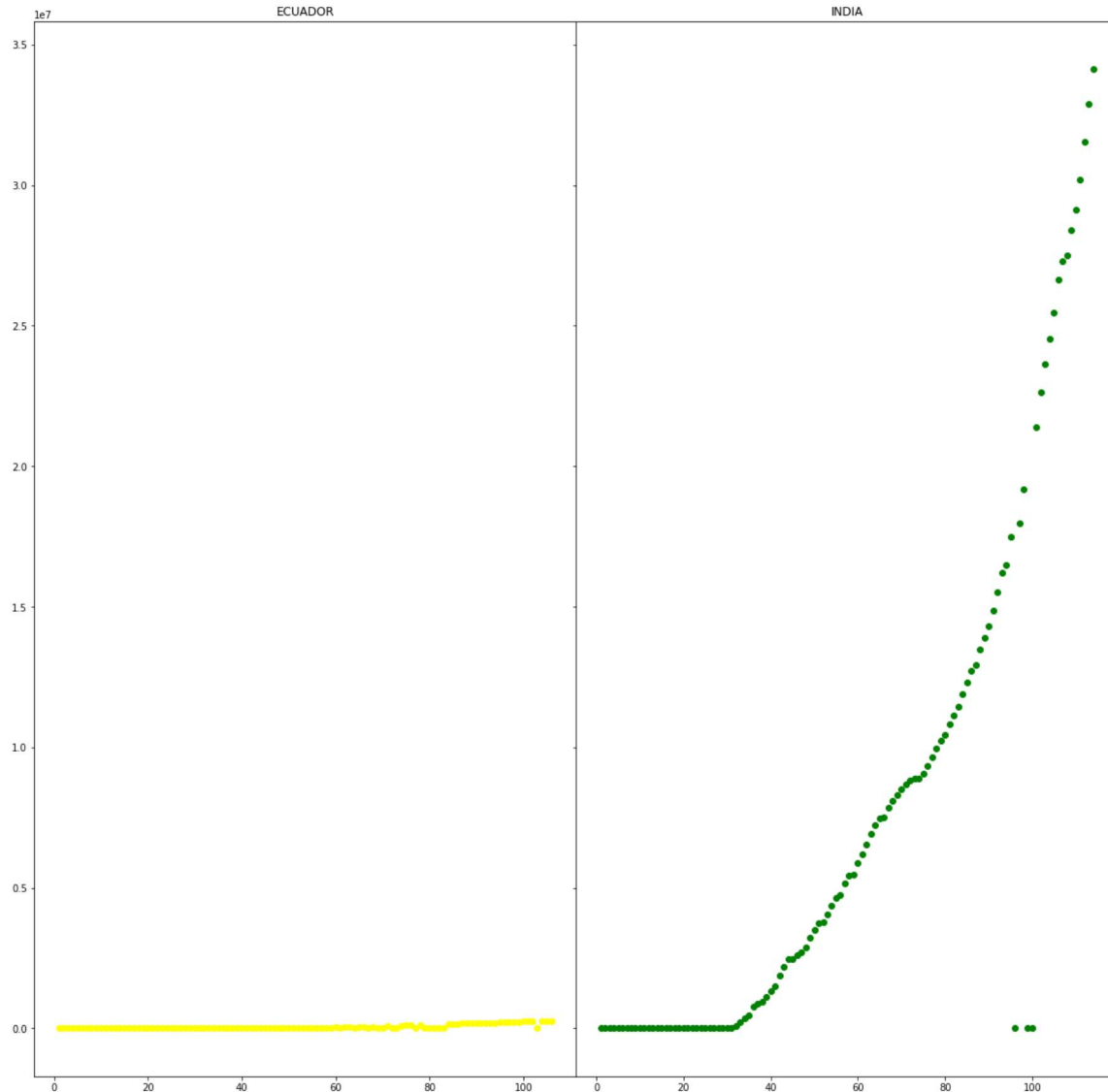
[236482.88382981]

Independent term:

-5920036.346530039

Out[210]:

Text(0.5, 1.0, 'INDIA')



Identificar cual es la fecha tentativa en la que todos los Ecuatorianos podrán ser vacunados con las dos dosis

In []:



1

Cual tiene una mejor predicción.

Ventajas: Aplicar una lectura de manera grafica y prediciendo datos que se obtiene para poder clasificar bien los datos.

Desventajas de los modelos: Usar los datos para poder aplicar un buen sistema de simulacion, es complicado si no se tiene los datos necesarios y la logica de estudio.

Conclusiones: Es bueno poder aplicar predicciones con datos en los cuales importe mucho ya que es horientado a la salud. Recomendaciones:Realizar mas ejemplos como sistemas donde se aplice la lectura de datos reales

In []:



1