

Proyecto 2 de IA

Determinación de parámetros de sistemas de ecuaciones diferenciales ordinarias.

Objetivo

Con la ayuda de algunos datos referentes a la pandemia provocada por el Sars-Cov2, determinar parámetros del modelo SIR con la ayuda del código proporcionado por el profesor.

```
In [168.] #Importamos paqueteria necesaria
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from pandas import DataFrame

In [161.] #Código del profesor
import PDEparams as pde
```

Definición del modelo SIR

El modelo SIR está dado por las siguientes ecuaciones:

$$\frac{dS}{dt} = -bSI$$
$$\frac{dI}{dt} = bSI - cI$$
$$\frac{dR}{dt} = cI,$$

(1)

(2)

(3)

donde S es la variable que representa a las personas susceptibles, I a las personas infectadas y R representa a las personas recuperadas.

```
In [165.] #Ecuación del modelo SIR
def SIR(x, t, b, c):
    I, R, S = x
    return [b*S*I-c*I, c*I, -b*S*I ]
```

Datos utilizados

Los datos que se van a utilizar son los referentes a la pandemia en Chile, obtenidos de <https://www.gob.cl/coronavirus/dtrasoficiales/datos>.

```
In [178.] #Cargar datos
datos_chile = pd.read_csv("data.csv")
len(datos_chile.index)

Out[178.] 299

In [187.] datos_chile.head(40)

Out[187.]
```

	Unnamed: 0	Activos	Recuperados
0	22-Feb	1.000	0.000
1	23-Feb	2.000	0.000
2	24-Feb	2.000	0.000
3	25-Feb	9.000	0.000
4	26-Feb	10.000	0.000
5	27-Feb	6.000	0.000
6	28-Feb	6.000	0.000
7	29-Feb	3.000	0.000
8	01-Mar	11.000	0.000
9	02-Mar	17.000	0.000
10	03-Mar	15.000	0.000
11	04-Mar	53.000	1.000
12	05-Mar	73.000	2.000
13	06-Mar	85.000	3.000
14	07-Mar	90.000	11.000
15	08-Mar	112.000	13.000
16	09-Mar	141.000	17.000
17	10-Mar	203.000	19.000
18	11-Mar	252.000	20.000
19	12-Mar	312.000	30.000
20	13-Mar	418.000	37.000
21	14-Mar	560.000	45.000
22	15-Mar	722.000	54.000
23	16-Mar	973.000	75.000
24	17-Mar	1.207	88.000
25	18-Mar	1.451	101.000
26	19-Mar	1.709	125.000
27	20-Mar	2.028	158.000
28	21-Mar	2.259	222.000
29	22-Mar	2.504	272.000
30	23-Mar	2.809	342.000
31	24-Mar	3.008	455.000
32	25-Mar	3.181	605.000
33	26-Mar	3.341	776.000
34	27-Mar	3.403	1.048
35	28-Mar	3.476	1.295
36	29-Mar	3.494	1.552
37	30-Mar	3.583	1.834
38	31-Mar	3.549	2.184
39	01-Abr	3.691	2.481

Aquí, Activos y Recuperados representan a las personas infectadas y recuperadas de las que habla el modelo SIR, respectivamente.

```
In [188.] #Revisar si hay datos faltantes
datos_chile.loc[datos_chile.Activos.isnull()]

Out[188.]
```

	Unnamed: 0	Activos	Recuperados
--	------------	---------	-------------

```
In [189.] datos_chile.loc[datos_chile.Recuperados.isnull()]

Out[189.]
```

	Unnamed: 0	Activos	Recuperados
--	------------	---------	-------------

Como en un inicio solamente hay personas infectadas y susceptibles a ser infectadas, entonces la condición inicial para este caso, de acuerdo con los datos, es de la forma $(S(0), I(0), R(0)) = (S_0, 1, 0)$, donde $S_0 > 0$.

```
In [190.] def cond_ini_S():
    return 9732569-1
def cond_ini_I():
    return 1
def cond_ini_R():
    return 0

In [191.] #Pasar a miles
df1 = datos_chile.iloc[24:]
df1 = df1.Activos.map(lambda x: x*1000)

In [192.] df2=datos_chile.iloc[24,1]

In [193.] df3=pd.concat([df2, df1], axis=0)

In [194.] df11 = datos_chile.iloc[34:]
df11 = df11.Recuperados.map(lambda x: x*1000)

In [195.] df22=datos_chile.iloc[:,34,2]
df33=pd.concat([df22, df11], axis=0)

In [200.] df=pd.concat([df3,df33],axis=1)
df.head(40)
```

```
Out[200.]
```

	Activos	Recuperados
0	1.0	0.0
1	2.0	0.0
2	2.0	0.0
3	9.0	0.0
4	10.0	0.0
5	6.0	0.0
6	6.0	0.0
7	3.0	0.0
8	11.0	0.0
9	17.0	0.0
10	15.0	0.0
11	53.0	1.0
12	73.0	2.0
13	85.0	3.0
14	90.0	11.0
15	112.0	13.0
16	141.0	17.0
17	203.0	19.0
18	252.0	20.0
19	312.0	30.0
20	418.0	37.0
21	560.0	45.0
22	722.0	54.0
23	973.0	75.0
24	1207.0	88.0
25	1451.0	101.0
26	1709.0	125.0
27	2028.0	158.0
28	2259.0	222.0
29	2504.0	272.0
30	2809.0	342.0
31	3008.0	455.0
32	3181.0	605.0
33	3341.0	776.0
34	3403.0	1048.0
35	3476.0	1295.0
36	3494.0	1552.0
37	3583.0	1834.0
38	3549.0	2184.0
39	3691.0	2481.0

```
In [201.] p = [x for x in range(290)]
t = DataFrame(p, columns=['time'])
#t=t.time.map(lambda x : x*86400)
t
```

```
Out[201.]
```

	time
0	0
1	1
2	2
3	3
4	4
...	...
285	285
286	286
287	287
288	288
289	289

290 rows × 1 columns

```
In [202.] # Base de datos con la que trabajamos
dff=pd.concat([t,df],axis=1)
dff
```

```
Out[202.]
```

	time	Activos	Recuperados
0	0	1.0	0.0
1	1	2.0	0.0
2	2	2.0	0.0
3	3	9.0	0.0
4	4	10.0	0.0
...
285	285	15420.0	531654.0
286	286	14681.0	533369.0
287	287	13610.0	534892.0
288	288	12351.0	536344.0
289	289	10996.0	537768.0

290 rows × 3 columns

Parámetros cuando solamente observamos a los infectados

```
In [203.] dff=dff[df.columns[[0,1]]]
dff
```

```
Out[203.]
```

	time	Activos
0	0	1.0
1	1	2.0
2	2	2.0
3	3	9.0
4	4	10.0
...
285	285	15420.0
286	286	14681.0
287	287	13610.0
288	288	12351.0
289	289	10996.0

290 rows × 2 columns

```
In [286.] modelo1 = pde.PDEmodel(dff, SIR, [cond_ini_I, cond_ini_R, cond_ini_S], bounds=[(973.9,974.1), (908.9,908.2)],
    param_names=[r'$b$', r'$c$'], nvars=3, ndims=0,nreplicates=1, obsidx=[1], outfunc=None)
```

```
In [287.] %time
modelo1.fit()
```

```
Out[287.]
```

	b	c
0	974.020272	908.936356

Wall time: 1.67 s

```
In [288.] modelo1.best_params
```

```
Out[288.]
```

	b	c
0	974.020272	908.936356

```
In [289.] modelo1.best_error
```

```
Out[289.] 334752685833786.1
```

```
In [290.] %time
modelo1.likelihood_profiles()
```

```
Wall time: 2min 46s

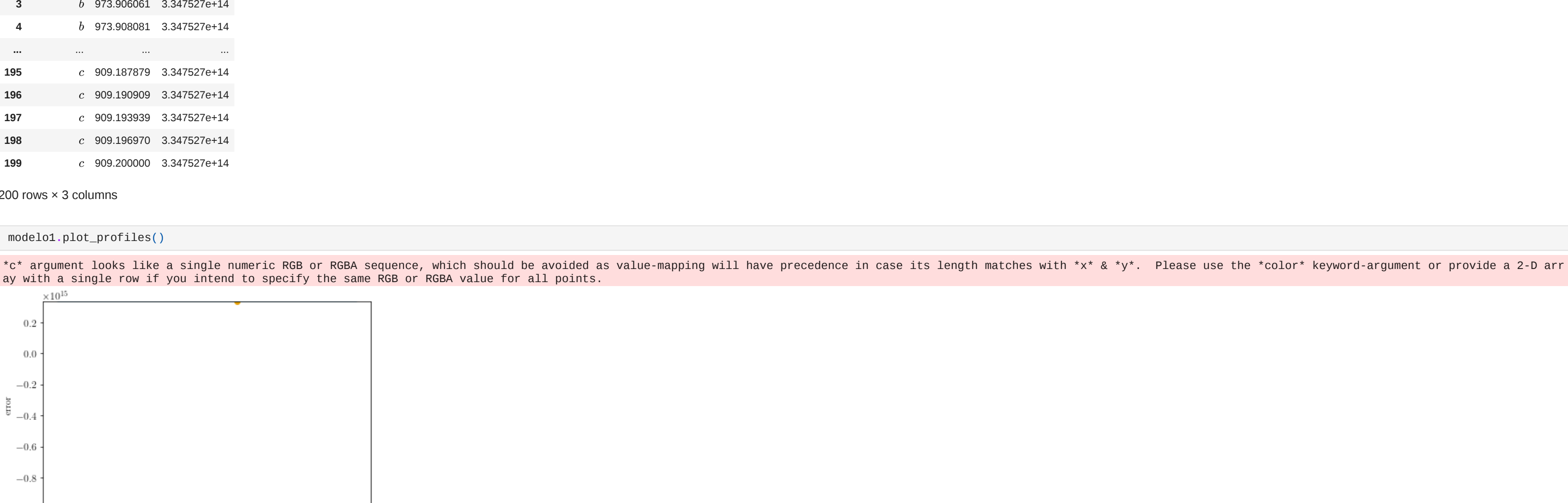
In [291.]
```

	parameter	value	error
0	b	973.900000	3.347527e+14
1	b	973.902020	3.347527e+14
2	b	973.904040	3.347527e+14
3	b	973.906061	3.347527e+14
4	b	973.908081	3.347527e+14
...
195	c	908.187879	3.347527e+14
196	c	908.189909	3.347527e+14
197	c	908.193939	3.347527e+14
198	c	908.196970	3.347527e+14
199	c	908.200000	3.347527e+14

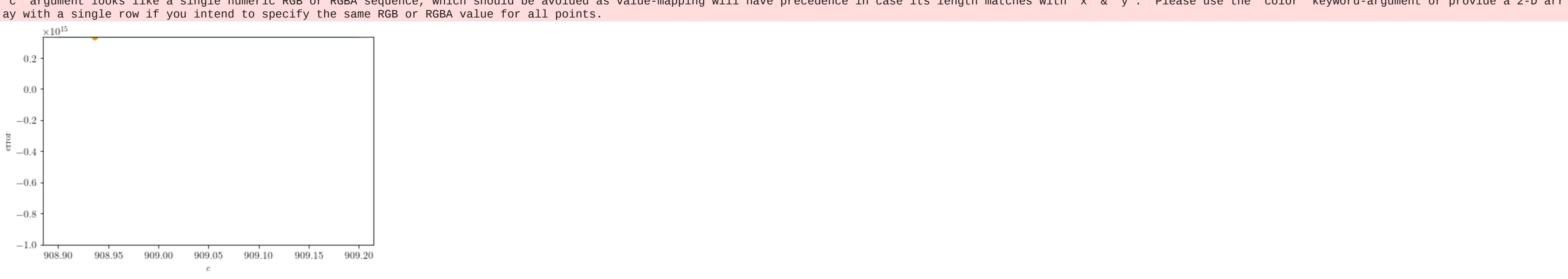
200 rows × 3 columns

```
In [292.] modelo1.plot_profiles()
```

c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provide a 2-D array with a single row if you intend to specify the same RGB or RGBA value for all points.



c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provide a 2-D array with a single row if you intend to specify the same RGB or RGBA value for all points.



Parámetros cuando se observan tanto infectados como recuperados

```
In [243.] modelo2 = pde.PDEmodel(dff, SIR, [cond_ini_I, cond_ini_R, cond_ini_S], bounds=[(0.5,3), (0.000005,0.01)],
    param_names=[r'$b$', r'$c$'], nvars=3, ndims=0,obsidx=[1,2], outfunc=None)
```

```
In [244.] %time
modelo2.fit()
```

```
Out[244.]
```

	b	c
0	0.99109	0.060006

Wall time: 4.1 s

```
In [245.] modelo2.best_params
```

```
Out[245.]
```

	b	c
0	0.99109	0.000006

```
In [246.] modelo2.best_error
```

```
Out[246.] 65635417939.2681
```

```
In [247.] %time
modelo2.likelihood_profiles()
```

```
Wall time: 8min 8s

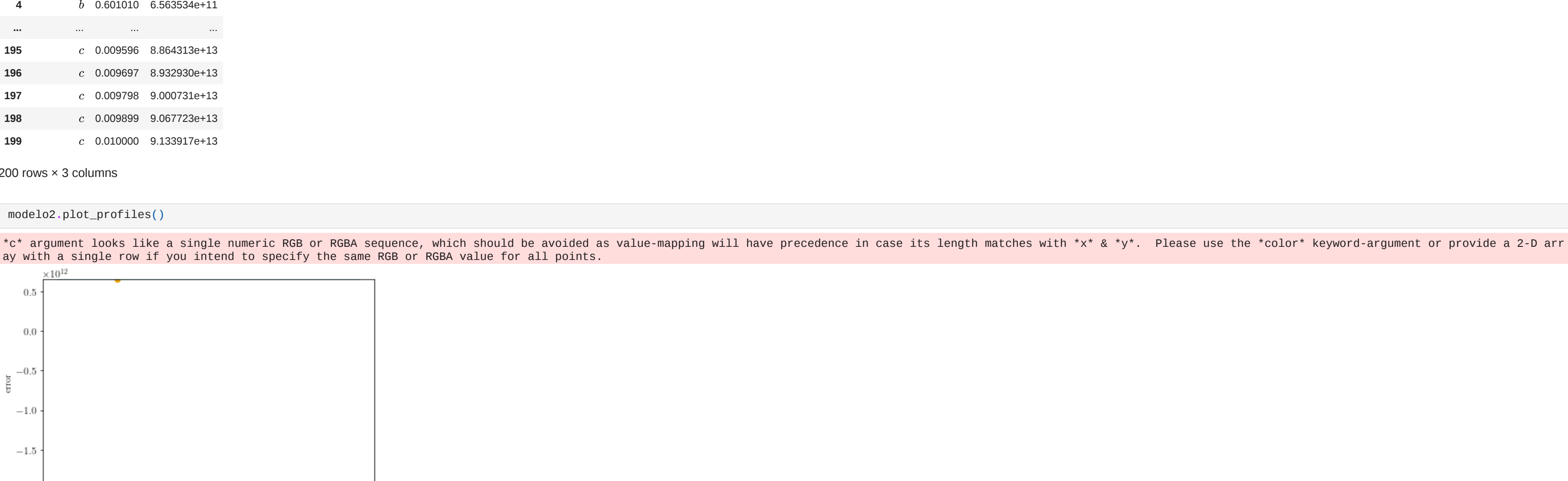
In [248.]
```

	parameter	value	error
0	b	0.500000	6.563534e+11
1	b	0.525293	6.563534e+11
2	b	0.550505	6.563534e+11
3	b	0.575798	6.563535e+11
4	b	0.601010	6.563534e+11
...
195	c	0.000596	8.864313e+13
196	c	0.000697	8.932930e+13
197	c	0.000798	9.000731e+13
198	c	0.000899	9.067723e+13
199	c	0.010000	9.133917e+13

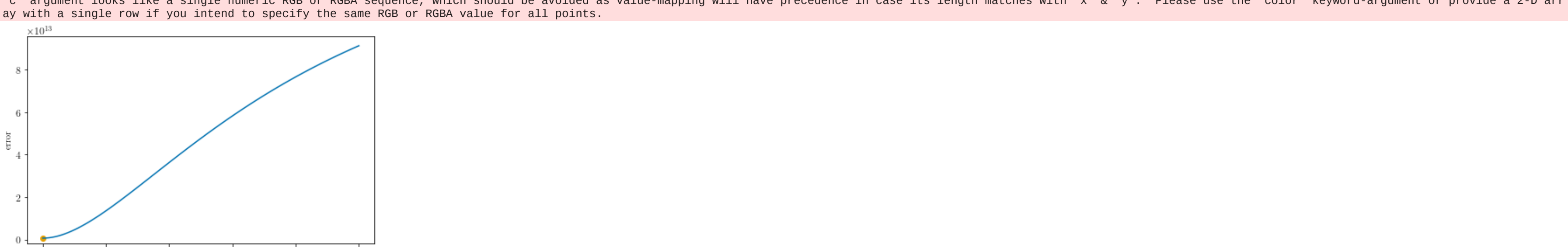
200 rows × 3 columns

```
In [249.] modelo2.plot_profiles()
```

c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provide a 2-D array with a single row if you intend to specify the same RGB or RGBA value for all points.



c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provide a 2-D array with a single row if you intend to specify the same RGB or RGBA value for all points.



```
In [ ]:
```