Proyecto 2

May 31, 2020

[0]:
```python
# Realizamos las importaciones de paquetes y bibliotecas necesarias
import math
import numpy as np
from decimal import Decimal
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

[0]:
```python
# Almacenamos los datos de los estadísticos, correspondiente a los estados,
# casos y muertes registradas respectivamente
Estado=('Ciudad de México','Estado de México','Baja California','Tabasco',
        'Veracruz','Sinaloa','Puebla','Nuevo León','Jalisco','Sonora',
        'Quintana Roo','Chihuahua','Tamaulipas','Michoaca','Yucatan',
        'Guanajuato','Hidalgo','Guerrero','Chiapas','Morelos','Oaxaca',
        'Coahuila','Aguascalientes','Tlaxcala','Queretaro','San Luis Potosi',
        'Baja Cal. Sur','Campeche','Nayarit','Durango','Zacatecas','Colima')
Casos=(21826,13140,4590,3484,3374,2903,2458,2353,1962,1809,1729,1665,1644,1625,
       1616,1564,1479,1421,1402,1297,1144,1044,972,944,813,786,557,583,503,279,
       274,128)
Muertes=(1945,1393,771,438,444,427,419,92,125,139,323,292,98,149,170,127,234,
         217,106,261,122,72,33,153,96,46,33,58,45,38,32,22)
```

[0]:
```python
datos={
    'Estado':['Ciudad de México','Estado de México','Baja California','Tabasco',
              'Veracruz','Sinaloa','Puebla','Nuevo León','Jalisco','Sonora',
              'Quintana Roo','Chihuahua','Tamaulipas','Michoaca','Yucatan',
              'Guanajuato','Hidalgo','Guerrero','Chiapas','Morelos','Oaxaca',
              'Coahuila','Aguascalientes','Tlaxcala','Queretaro',
              'San Luis Potosi','Baja Cal. Sur','Campeche','Nayarit','Durango',
              'Zacatecas','Colima'],
    'Casos':[21826,13140,4590,3484,3374,2903,2458,2353,1962,1809,1729,1665,1644,
             1625,1616,1564,1479,1421,1402,1297,1144,1044,972,944,813,786,557,
             583,503,279,274,128],
    'Muertes':[1945,1393,771,438,444,427,419,92,125,139,323,292,98,149,170,127,
               234,217,106,261,122,72,33,153,96,46,33,58,45,38,32,22]
}
```

1 A continuación se presentaran los casos y muertes por Covid-19 en cada estado de la Republica Mexicana

por: - Chilpa Navarro Martin Enrique - Espinosa Guadarrama Arturo

---

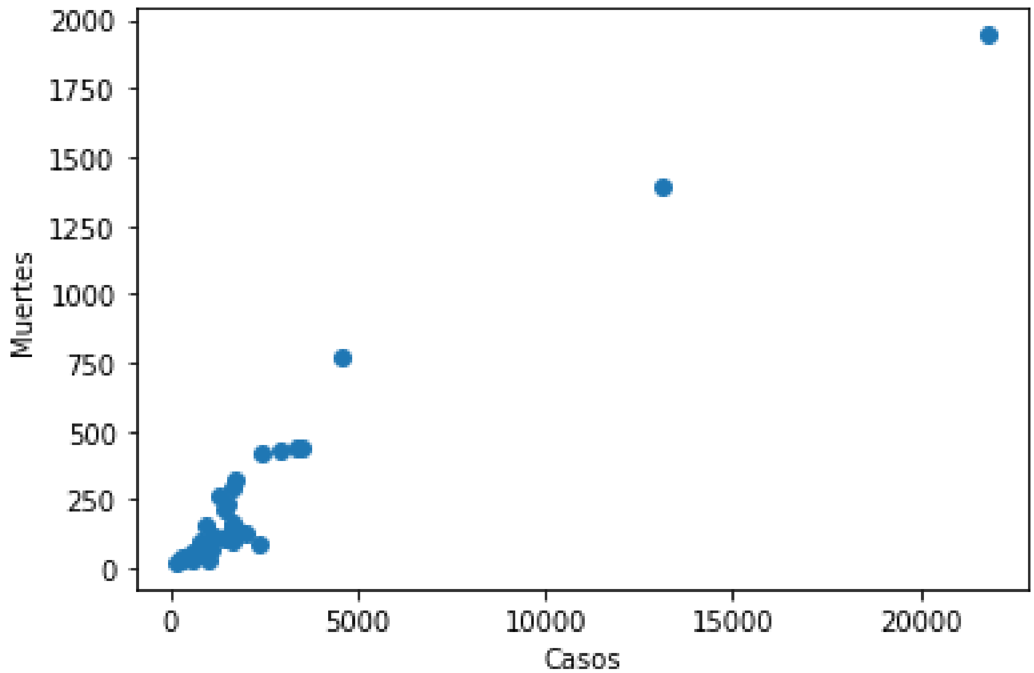
[0]:
```
# Mostramos en una tabla todos los registros estadísticos de cada estado
dataFrameCovid2=pd.DataFrame(datos)
dataFrameCovid2
```

[0]:

	Estado	Casos	Muertes
0	Ciudad de México	21826	1945
1	Estado de México	13140	1393
2	Baja California	4590	771
3	Tabasco	3484	438
4	Veracruz	3374	444
5	Sinaloa	2903	427
6	Puebla	2458	419
7	Nuevo León	2353	92
8	Jalisco	1962	125
9	Sonora	1809	139
10	Quintana Roo	1729	323
11	Chihuahua	1665	292
12	Tamaulipas	1644	98
13	Michoaca	1625	149
14	Yucatan	1616	170
15	Guanajuato	1564	127
16	Hidalgo	1479	234
17	Guerrero	1421	217
18	Chiapas	1402	106
19	Morelos	1297	261
20	Oaxaca	1144	122
21	Coahuila	1044	72
22	Aguascalientes	972	33
23	Tlaxcala	944	153
24	Queretaro	813	96
25	San Luis Potosi	786	46
26	Baja Cal. Sur	557	33
27	Campeche	583	58
28	Nayarit	503	45
29	Durango	279	38
30	Zacatecas	274	32
31	Colima	128	22

[0]:
```
# Realizamos la representación gráfica con un diagrama de dispersión sobre los
# casos vs muertes registradas
plt.scatter(dataFrameCovid2["Casos"],dataFrameCovid2["Muertes"])
plt.xlabel("Casos")
```

```
plt.ylabel("Muertes")
plt.show()
```



La gráfica muestra datos dispersos, sin perder la tendencia lineal que se busca, sin embargo, solo para fines de visualización omitiremos los datos más alejados

[0]:
```
datos={
    'Estado':['Baja California','Tabasco','Veracruz','Sinaloa','Puebla',
              'Nuevo León','Jalisco','Sonora','Quintana Roo','Chihuahua',
              'Tamaulipas','Michoaca','Yucatan','Guanajuato','Hidalgo',
              'Guerrero','Chiapas','Morelos','Oaxaca','Coahuila',
              'Aguascalientes','Tlaxcala','Queretaro','San Luis Potosi',
              'Baja Cal. Sur','Campeche','Nayarit','Durango','Zacatecas',
              'Colima'],
    'Casos':[4590,3484,3374,2903,2458,2353,1962,1809,1729,1665,1644,1625,1616,
             1564,1479,1421,1402,1297,1144,1044,972,944,813,786,557,583,503,279,
             274,128],
    'Muertes':[771,438,444,427,419,92,125,139,323,292,98,149,170,127,234,217,
               106,261,122,72,33,153,96,46,33,58,45,38,32,22]
}
```

[0]:
```
dataFrameCovid=pd.DataFrame(datos)
```

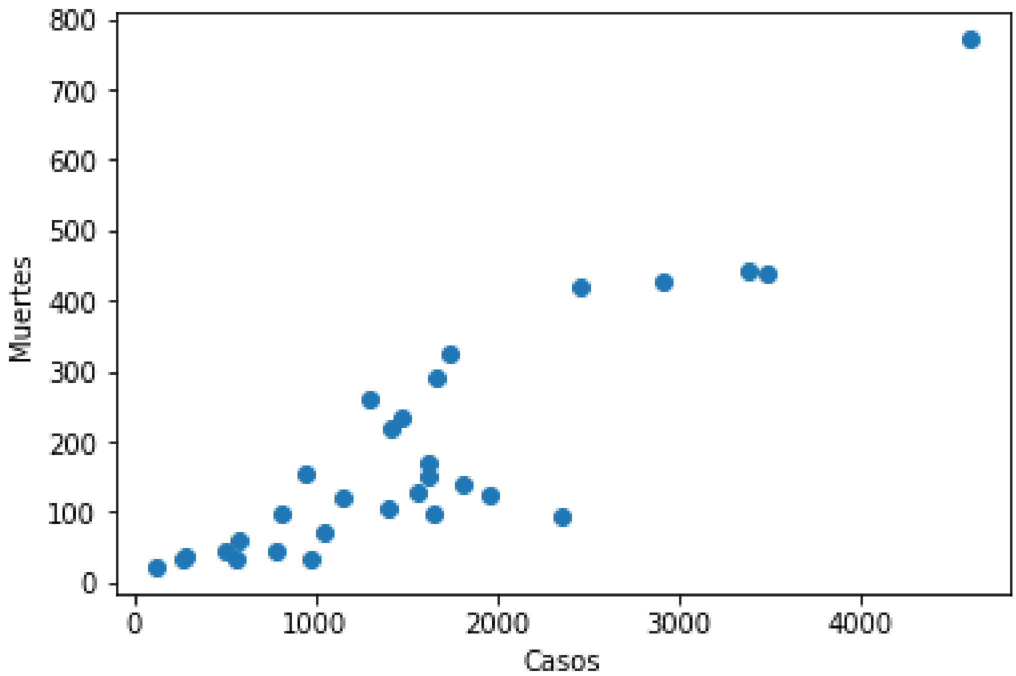
[0]:
```
dataFrameCovid
```

[0]:

	Estado	Casos	Muertes
0	Baja California	4590	771
1	Tabasco	3484	438
2	Veracruz	3374	444
3	Sinaloa	2903	427
4	Puebla	2458	419
5	Nuevo León	2353	92
6	Jalisco	1962	125
7	Sonora	1809	139
8	Quintana Roo	1729	323
9	Chihuahua	1665	292
10	Tamaulipas	1644	98
11	Michoaca	1625	149
12	Yucatan	1616	170
13	Guanajuato	1564	127
14	Hidalgo	1479	234
15	Guerrero	1421	217
16	Chiapas	1402	106
17	Morelos	1297	261
18	Oaxaca	1144	122
19	Coahuila	1044	72
20	Aguascalientes	972	33
21	Tlaxcala	944	153
22	Queretaro	813	96
23	San Luis Potosi	786	46
24	Baja Cal. Sur	557	33
25	Campeche	583	58
26	Nayarit	503	45
27	Durango	279	38
28	Zacatecas	274	32
29	Colima	128	22

[0]:

```
plt.scatter(dataFrameCovid["Casos"],dataFrameCovid["Muertes"])
plt.xlabel("Casos")
plt.ylabel("Muertes")
plt.show()
```



Observamos que la gráfica muestra un comportamiento lineal, por lo que se comprabará a continuación obteniendo el coeficiente de correlación. Los casos correspenderan a la variable independiente "x", por ende el numero de muertes sera la varieble dependiente "y"

El coeficiente de correlación se obtiene de la siguiente manera: $r = \frac{Sxy}{\sqrt{Sxx * Sxy}}$

donde:

$S_(xy) = \sum_{i=1}^{n}(x_i - xmedia)(y_i - ymedia)$

$S_(yy) = \sum_{i=1}^{n}(y_i - ymedia)^2$

$S_(xx) = \sum_{i=1}^{n}(x_i - xmedia)^2$

[0]:
```
xmedia=np.mean(Casos)
ymedia=np.mean(Muertes)
print(" x media=",np.mean(Casos))
print(" y media=",np.mean(Muertes))
```

```
x media= 2542.75
y media= 278.75
```

[0]:
```
xmenos_media=[]
ymenos_media=[]



for i in range(0,len(Casos)):
```

```
    elemento=Casos[i]-xmedia
    xmenos_media.append(elemento)
print("xmenos_media",(xmenos_media))
print(sum(xmenos_media))

for i in range(0,len(Muertes)):
    elemento=Muertes[i]-ymedia
    ymenos_media.append(elemento)
print("ymenos_media:",ymenos_media)
print(sum(ymenos_media))
```

```
xmenos_media [19283.25, 10597.25, 2047.25, 941.25, 831.25, 360.25, -84.75,
-189.75, -580.75, -733.75, -813.75, -877.75, -898.75, -917.75, -926.75, -978.75,
-1063.75, -1121.75, -1140.75, -1245.75, -1398.75, -1498.75, -1570.75, -1598.75,
-1729.75, -1756.75, -1985.75, -1959.75, -2039.75, -2263.75, -2268.75, -2414.75]
0.0
ymenos_media: [1666.25, 1114.25, 492.25, 159.25, 165.25, 148.25, 140.25,
-186.75, -153.75, -139.75, 44.25, 13.25, -180.75, -129.75, -108.75, -151.75,
-44.75, -61.75, -172.75, -17.75, -156.75, -206.75, -245.75, -125.75, -182.75,
-232.75, -245.75, -220.75, -233.75, -240.75, -246.75, -256.75]
0.0
```

[0]:
```
xmenos_cuadrada=[]

for i in range(0,len(Casos)):
    elemento=(xmenos_media[i]*xmenos_media[i])
    xmenos_cuadrada.append(elemento)

#Sxx[31]
#xmenos_media[31]
sxx=sum(xmenos_cuadrada)
print("Sxx:",sxx)
```

```
Sxx 544456212.0
```

[0]:
```
# Realizamos los cálculos necesarios y los almacenamos dentro de dos listas
# respectivamente
ymenos_cuadrada=[]

for i in range(0,len(Casos)):
    elemento=(ymenos_media[i]*ymenos_media[i])
    ymenos_cuadrada.append(elemento)

#Sxx[31]
#xmenos_media[31]
syy=sum(ymenos_cuadrada)
```

```
print("Syy:",syy)
```

```
Syy: 5134702.0
```

[0]:
```
xmen_ymen=[]

for i in range(0,len(Casos)):
    elemento=(xmenos_media[i]*ymenos_media[i])
    xmen_ymen.append(elemento)
#print(Sxy[0])

sxy=sum(xmen_ymen)
print("Sxy:",sxy)
```

```
Sxy: 51285126.0
```

[0]:
```
correlacion=sxy/math.sqrt(syy*sxx)
```

[0]:
```
0.9699566532218736
```

$r = \frac{Sxy}{\sqrt{Sxx*Sxy}} = \frac{51285126.0}{\sqrt{544456212.0*5134702.0}} = .9699566$

Por lo que se puede decir que los datos mantienen una fuerza de asociación lineal muy fuerte, siendo r=1 el valor ideal

Ahora se obtendrá el modelo matematico de la y estimada

$y_(estimada) = \beta_0 + \beta_1 x$

donde:

$\beta_1 = \frac{Sxy}{Sxx}$

$\beta_0 = y(media) - \frac{Sxy}{Sxx} xmedia$

[0]:
```
bo=ymedia-(sxy/sxx)*xmedia
```

$\beta_0 = y(media) - \frac{Sxy}{Sxx} xmedia = 39.2553$

[0]:
```
b1=sxy/sxx
b1
```

[0]:
```
0.09419513428198335
```

$\beta_1 = \frac{Sxy}{Sxx} = .094195$

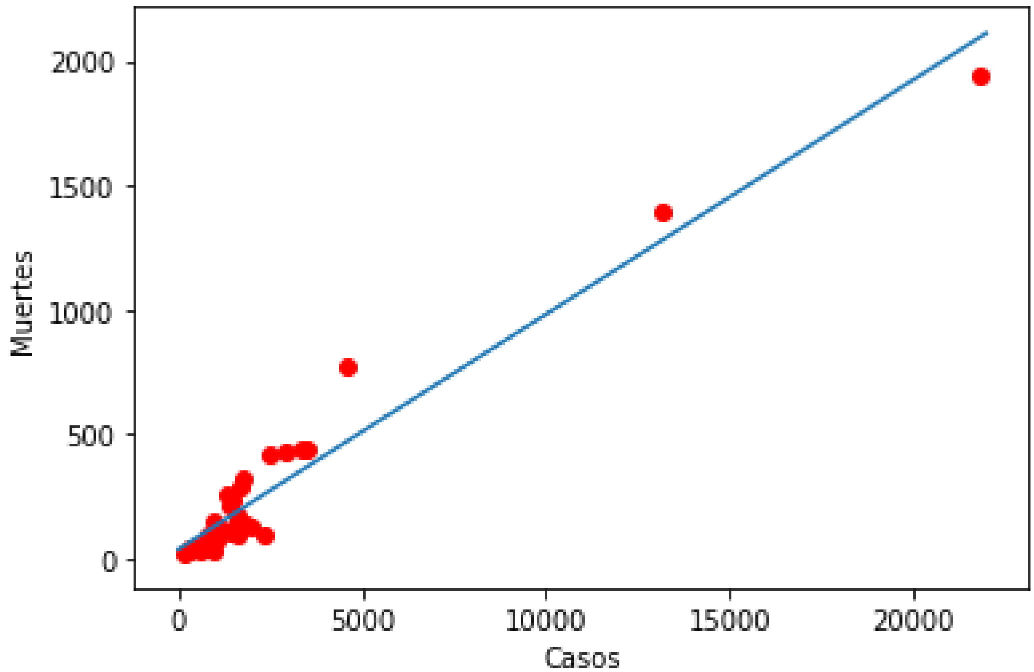
Por lo que se obtiene: $y_(estimada) = 39.2553 + .094195x$

[0]:
```
x=np.arange(0,22000,10)

y=bo+b1*x
```

```
plt.plot(x,y)

plt.scatter(dataFrameCovid2["Casos"],dataFrameCovid2["Muertes"],color="red")
plt.xlabel("Casos")
plt.ylabel("Muertes")
plt.show()
```



1.1 Ahora realizaremos el mismo procemiento pero usando la libreria de Python Sklearn

[0]:
```
X=dataFrameCovid2["Casos"].values.reshape(-1,1)
Y=dataFrameCovid2["Muertes"].values.reshape(-1,1)

linear_regressor=LinearRegression()
linear_regressor.fit(X,Y)
Y_pred=linear_regressor.predict(X)
```

[0]:
```
m=linear_regressor.coef_[0][0]
c=linear_regressor.intercept_[0]
label=r"$Casos= %0.4f*Casos%+0.4f$"%(m,c)
```

[0]:
```
# Realizamos su representación gráfica trazando una recta pendiente sobre la
# gráfica de dispersión ya creada anteriormente
```

```python
plt.scatter(dataFrameCovid2["Casos"],dataFrameCovid2["Muertes"])
plt.plot(X,Y_pred,color="red",label=label)
plt.ylabel("Muertes")
plt.xlabel("Casos")
plt.legend()
plt.show()
```