2.2.3 statsmodels

December 19, 2020

1 Diagnóstico de modelos usando statsmodels

1.1 Problema

Se quiere saber si el valor de una variable independiente (y) se encuentra asociada con el valor de una variable independiente (x).

El modelo se estima utilizando la regresión de mínimos cuadrados ordinarios.

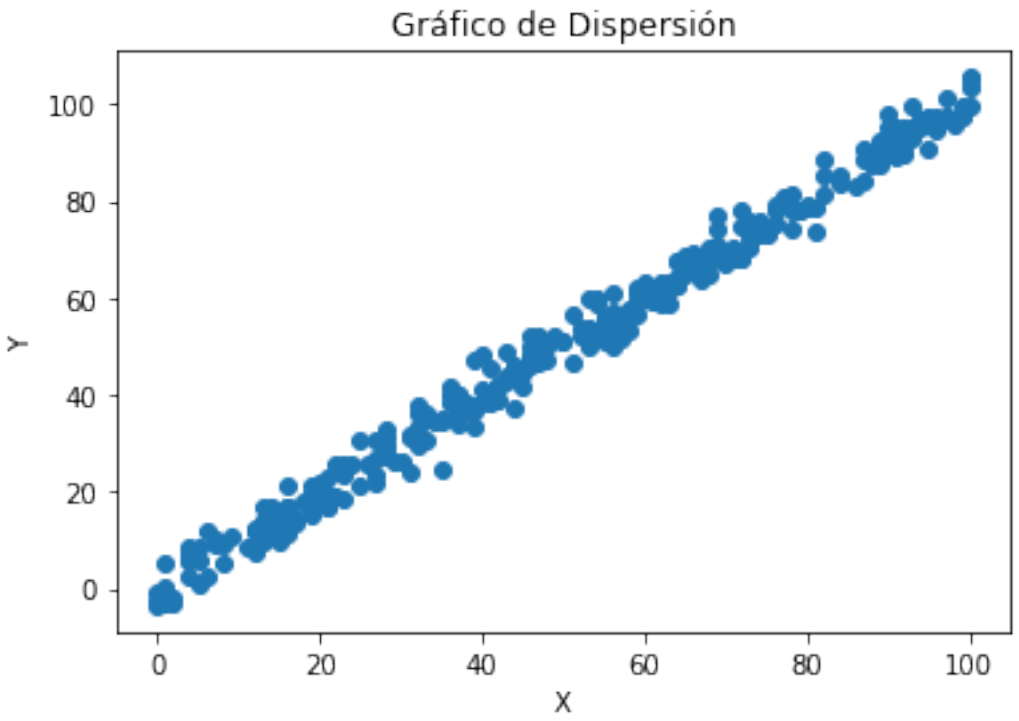
[12]:
```
import pandas as pd
data = pd.read_csv("DataSimpleLinearRegression.csv")
data
```

[12]:

	x	y
0	77	79.775152
1	21	23.177279
2	22	25.609262
3	20	17.857388
4	36	41.849864
..	..	…
295	71	68.545888
296	46	47.334876
297	55	54.090637
298	62	63.297171
299	47	52.459467

[300 rows x 2 columns]

[14]:
```
import matplotlib.pyplot as plt
plt.scatter(data['x'], data['y'])
plt.title('Gráfico de Dispersión')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



1.2 Comenzamos a trabajar

Para ajustar la mayoría de los modelos cubiertos por statsmodels (https://www.statsmodels.org/stable/index.html) , se deben crear 2 matrices de diseño.

- La primera matriz es una matriz de variables endógenas, también conocidas como dependientes o de regresión.
- La segunda matriz es una matriz de variables exógenas, también conocidas como independientes, predictoras o regresoras.

El módulo patsy proporciona una función conveniente para preparar matrices de diseño.

1.3 PASO 1 - Crear las matrices

[15]:
```
from patsy import dmatrices
y, X = dmatrices('y ~ x', data = data, return_type = 'dataframe')
y[:3]
```

[15]:
```
           y
0  79.775152
1  23.177279
2  25.609262
```

1.4 PASO 2 - Ajuste y reducción del modelo

[16]:
```
modelo = sm.OLS(y, X)
resultado = modelo.fit()
print(resultado.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.989
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                 2.709e+04
Date:                Sat, 19 Dec 2020   Prob (F-statistic):          1.33e-294
Time:                        21:23:39   Log-Likelihood:                -757.98
No. Observations:                 300   AIC:                             1520.
Df Residuals:                     298   BIC:                             1527.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.4618      0.360     -1.284      0.200      -1.169       0.246
x              1.0143      0.006    164.598      0.000       1.002       1.026
==============================================================================
Omnibus:                        1.034   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.596   Jarque-Bera (JB):                0.825
Skew:                           0.117   Prob(JB):                        0.662
Kurtosis:                       3.104   Cond. No.                         120.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

1.5 Explicando la tabla anterior

Antes de comenzar se van a considerar algunos conceptos previos

1.5.1 Modelo de regresión ineal

Predicciones de una variable independiente (y) sabiendo los valores de otras variables indepedientes x_1, x_2, … x_n; en caso de que n sea superior a 1 entonces se trata de regresión lineal múltiple.

1.5.2 Ecuación de regresión

$y = w_0 + w_1 x$: donde w_0 es el punto de intersección con el eje Y y w_1 es la pendiente de la recta

1.5.3 Método de los mínimos cuadrados

Se utiliza para calcular los valores de w_0 y w_1 la recta de la regresión lineal que minimiza los residuos.

1.6 Explicando la tabla

En la especificación de un modelo se debe verificar lo siguiente:

- Los parámetros del modelo w_0 y w_1 deben ser estadísticamente diferentes de cero.
- La distribución de los errores debe ser normal.
- La varianza de los errores σ^2 debe ser constante.

El no cumplimiento de estos supuestos indica que pueden haber problemas en la especificación del modelo o en los datos.

1.7 Valores Relevantes

1.7.1 Estadístico F

En la tabla, es de particular interés el estadístico F. La hipótesis nula es que todos los parámetros del modelo (w_0 y w_1) son diferentes de cero, versus la hipótesis alterna en que se indica que al menos uno de ellos no es significativamente de cero.

1.7.2 Coeficiente de determinación (R-squared)

El coeficiente de determinación mide cuanta de la variación de y es explicada por el modelo. Su fórmula es: $^2 = 1 - \frac{\sigma_e^2}{\sigma_y^2}$

Si la varianza de los errores (diferencia entre el valor real y el predecido) σ_e^2 es cero, el modelo explica el 100% de la variable . Si σ_e^2 es igual a la varianza de (σ_y^2) el modelo no explica nada y ^{2} vale cero.

1.7.3 Coeficiente de determinación (Adj. R-squared)

El coeficiente de correlación ajustado ^{2} corrige el valor de ^{2} por la cantidad de variables (igual a 1 para el caso analizado) y la cantidad de datos :

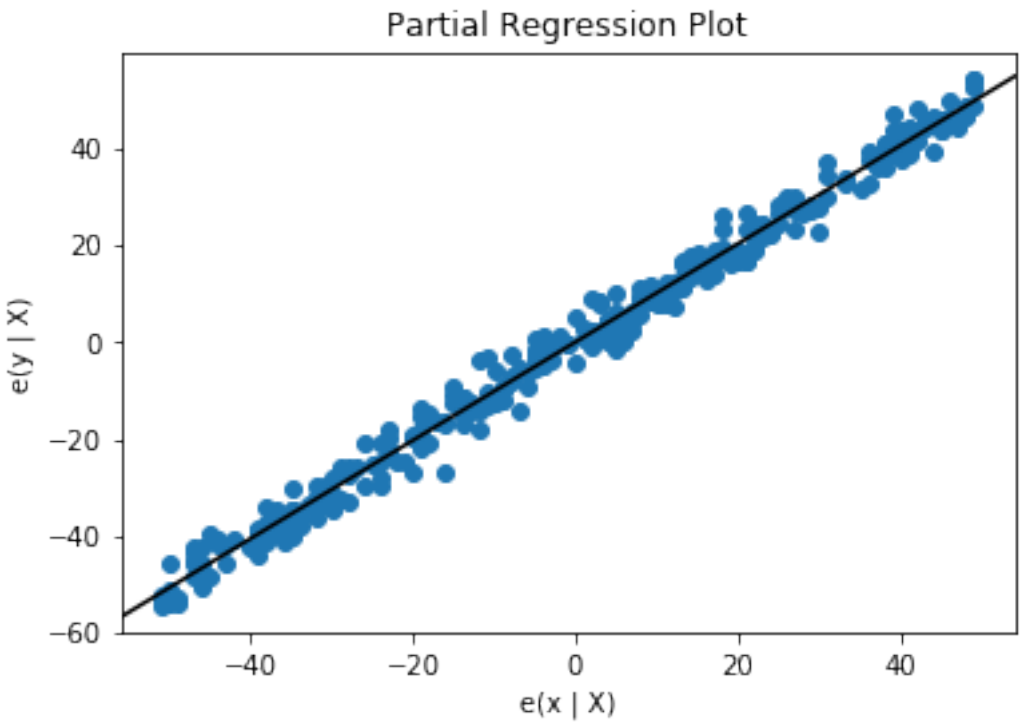
$^2 Ajustado = 1 - \frac{-1}{--1}(1 - ^2)$

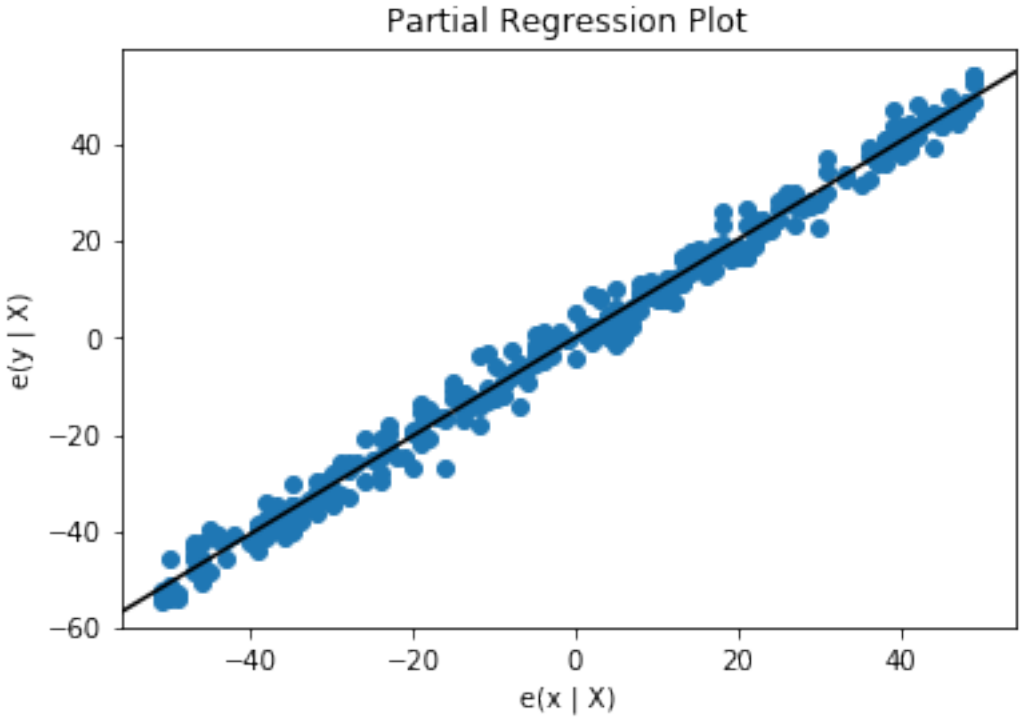
1.8 PASO 3 - Gráficos

Se van a generar algunos gráficos

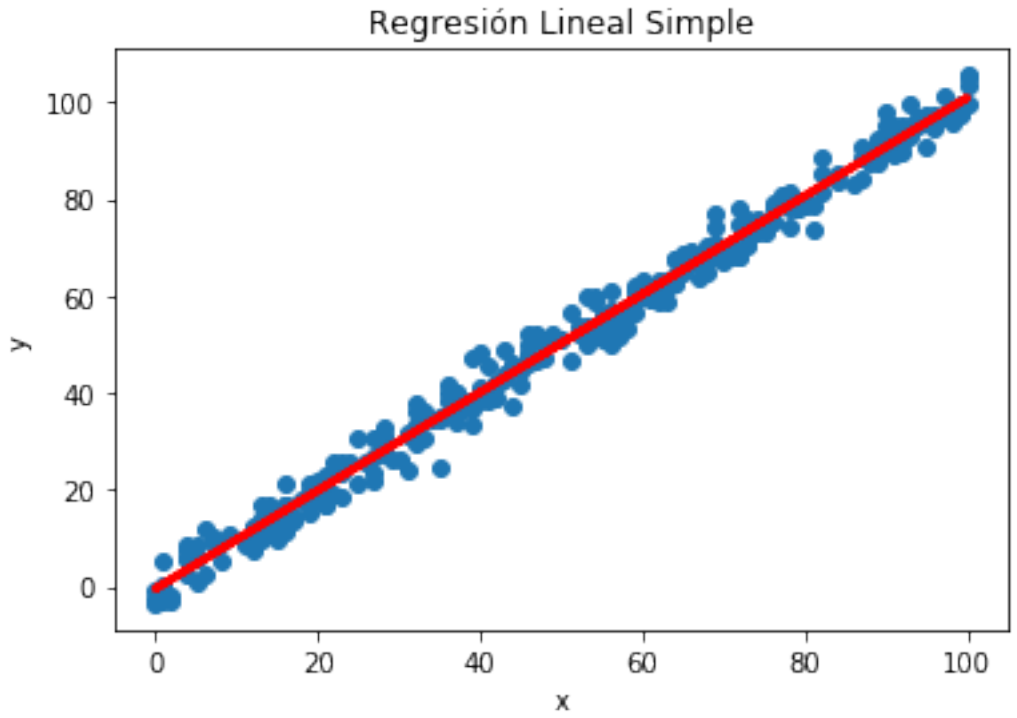
[18]:
```
sm.graphics.plot_partregress('y','x',[], data = data, obs_labels=False)
```

[18]:


Partial Regression Plot
40
20
0
−20
−40
−60
e(y | X)
−40
−20
0
20
40
e(x | X)


Partial Regression Plot
40
20
0
−20
−40
−60
e(y | X)
−40
−20
0
20
40
e(x | X)

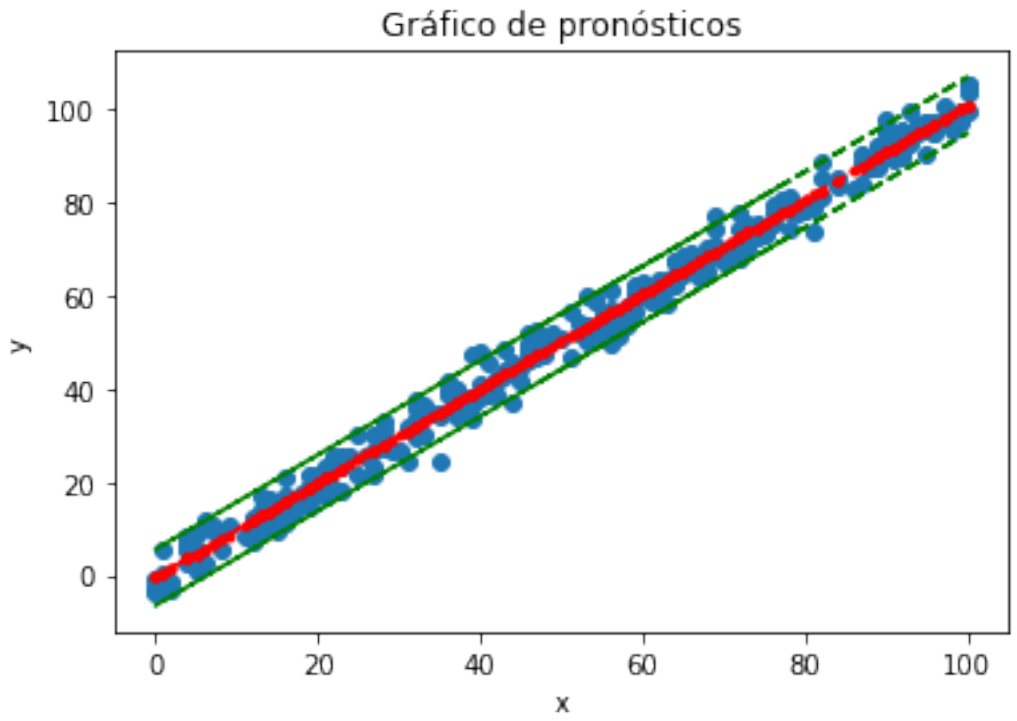
[19]:
```
plt.scatter(data['x'], data['y'])
plt.plot(data['x'], resultado.predict(), color="red", linewidth=3)
plt.title('Regresión Lineal Simple')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



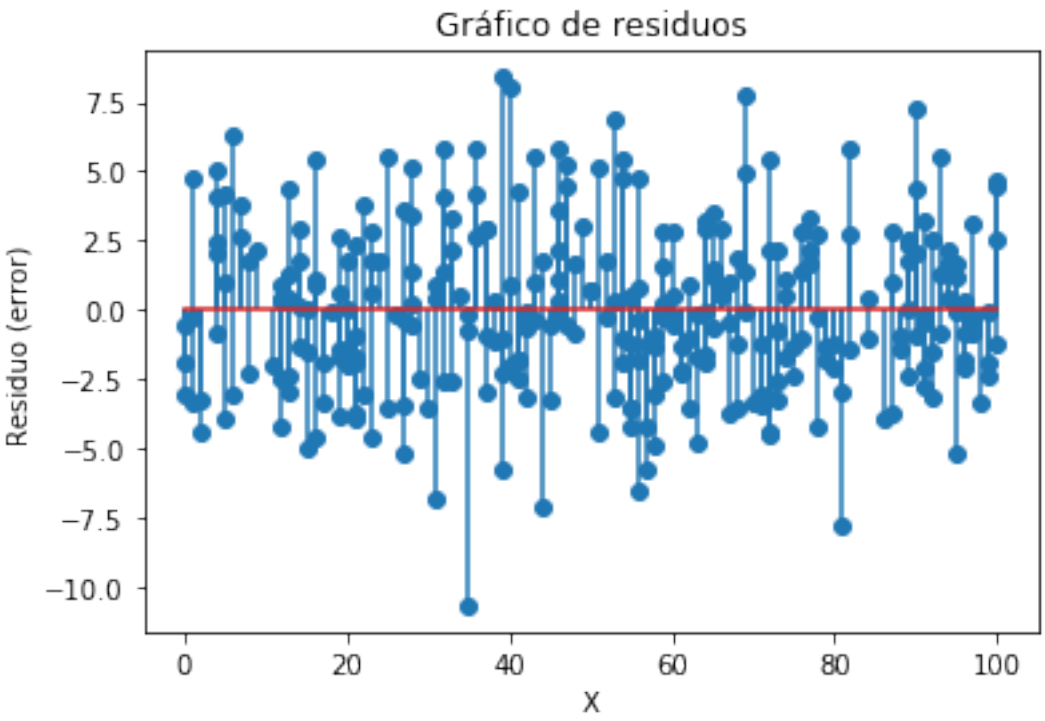
[32]:
```
##
## Gráfico del pronóstico usando statsmodel
##

from statsmodels.sandbox.regression.predstd import wls_prediction_std

mean_pred, lower, upper = wls_prediction_std(resultado)
X = data['x']
plt.plot(X, y, 'o', label="data")
plt.plot(X, resultado.fittedvalues, 'r--.', label="OLS")
plt.plot(X, upper, 'g--')
plt.plot(X, lower, 'g--');
plt.title('Gráfico de pronósticos')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



[35]:
```
## Gráfico de residuos
plt.stem(data['x'], resultado.resid, use_line_collection=True);
plt.title("Gráfico de residuos")
plt.xlabel("X")
plt.ylabel("Residuo (error)")
plt.show()
```



[34]:
```
## Valores predecidos
resultado.predict()
```

[34]:
```
array([ 77.64201157,  20.83923168,  21.85356704,  19.82489633,
        36.05426201,  14.75321955,  62.42698124,  95.90004796,
        19.82489633,   4.609866  ,   3.59553065,  18.81056097,
        96.91438332,  62.42698124,  36.05426201,  14.75321955,
        65.46998731,  13.7388842 ,  87.78536512,  69.52732873,
        89.81403583,  51.26929234,  89.81403583,  26.92524381,
        97.92871867,  58.36963982,  79.67068228,  20.83923168,
        93.87137725,  26.92524381,  99.95738938,  30.98258524,
        33.01125595,  80.68501764,  27.93957917,  47.21195092,
        53.29796305,  69.52732873,  27.93957917,  33.01125595,
        91.84270654,  71.55599944,  50.25495698,  76.62767622,
         3.59553065,  37.06859737,  70.54166408,  68.51299337,
        40.11160343,  35.03992666,  94.88571261,  88.79970048,
        52.28362769,  30.98258524,  59.38397518,  -0.46181077,
        39.09726808,  64.45565195,  69.52732873,  57.35530447,
        12.72454884,  72.57033479,  76.62767622,  61.41264589,
        82.71368835,  17.79622562,  41.12593879,  50.25495698,
        55.32663376,  12.72454884,  46.19761556,  12.72454884,
        79.67068228,  53.29796305,  14.75321955,  27.93957917,
        81.69935299,  69.52732873,  52.28362769,  84.74235906,
```

```
68.51299337,  26.92524381,  56.34096911,  48.22628627,
40.11160343,  39.09726808,  82.71368835, 100.97172474,
59.38397518,  43.1546095 ,  67.49865802,  38.08293272,
63.4413166 ,  91.84270654,  60.39831053,  13.7388842 ,
20.83923168,  87.78536512,  73.58467015,  31.99692059,
 1.56685994,  82.71368835,  18.81056097,  74.59900551,
42.14027414,  11.71021349,   0.55252458,  90.82837119,
89.81403583,  -0.46181077,  41.12593879,  15.76755491,
94.88571261,  97.92871867,  66.48432266,  23.88223775,
16.78189026,  90.82837119,  12.72454884,  -0.46181077,
64.45565195,  96.91438332,  98.94305403,  11.71021349,
41.12593879,  47.21195092,  78.65634693,  19.82489633,
89.81403583,  28.95391452,  64.45565195,  75.61334086,
11.71021349,  24.8965731 ,  27.93957917,  29.96824988,
65.46998731,  59.38397518,  64.45565195,  53.29796305,
71.55599944,  97.92871867,  73.58467015,   8.66720742,
11.71021349,  63.4413166 ,  99.95738938,  60.39831053,
35.03992666,   1.56685994,  60.39831053,  31.99692059,
94.88571261,  84.74235906,  63.4413166 ,  21.85356704,
81.69935299,  93.87137725,  33.01125595,   6.63853671,
42.14027414,  46.19761556,  54.3122984 ,  15.76755491,
49.24062163,  43.1546095 ,  95.90004796,  66.48432266,
20.83923168,  35.03992666,  80.68501764,  37.06859737,
54.3122984 ,  56.34096911,   0.55252458,  31.99692059,
58.36963982,  31.99692059,  46.19761556,  72.57033479,
16.78189026,  97.92871867,  93.87137725,  91.84270654,
37.06859737,   3.59553065,  54.3122984 ,  51.26929234,
26.92524381,  46.19761556,  92.8570419 ,  73.58467015,
77.64201157,  91.84270654,  61.41264589,  99.95738938,
 3.59553065,  72.57033479,  18.81056097,  57.35530447,
78.65634693,  25.91090846,  74.59900551,  90.82837119,
66.48432266,  12.72454884,  40.11160343,  77.64201157,
67.49865802,  75.61334086,  22.86790239,  45.18328021,
59.38397518,  44.16894485,  22.86790239,  55.32663376,
55.32663376,  95.90004796,  11.71021349,   3.59553065,
 6.63853671, 100.97172474,  48.22628627,  42.14027414,
96.91438332,  39.09726808, 100.97172474,  87.78536512,
13.7388842 ,  13.7388842 ,  37.06859737,   4.609866  ,
88.79970048,  91.84270654,  65.46998731,  74.59900551,
56.34096911,  15.76755491,   4.609866  ,  27.93957917,
92.8570419 ,  46.19761556,  54.3122984 ,  39.09726808,
44.16894485,  30.98258524,  68.51299337,  86.77102977,
90.82837119,  38.08293272,  20.83923168,  95.90004796,
56.34096911,  60.39831053,  65.46998731,  78.65634693,
89.81403583,   5.62420136,  67.49865802,  36.05426201,
15.76755491, 100.97172474,  45.18328021,  73.58467015,
57.35530447,  19.82489633,  76.62767622,  34.0255913 ,
```

```
       55.32663376,  72.57033479,  55.32663376,   7.65287207,
       56.34096911,  72.57033479,  58.36963982,   5.62420136,
       96.91438332,  22.86790239,  58.36963982,  22.86790239,
       18.81056097,  24.8965731 ,  64.45565195,  20.83923168,
       59.38397518,  18.81056097,  15.76755491,  42.14027414,
       43.1546095 ,  61.41264589,  92.8570419 ,  10.69587813,
       41.12593879,   0.55252458,   7.65287207,  71.55599944,
       46.19761556,  55.32663376,  62.42698124,  47.21195092])
```

[ ]: