

# Taller 1 - corte 2

Modelo de regresión lineal con el uso de matrices

Al ajustar un modelo de regresión lineal múltiple, en particular cuando el número de variable pasa de dos, el conocimiento de la teoría matricial puede facilitar las manipulaciones matemáticas. Supongamos que el experimentador tiene  $x_1, x_2, \dots, x_k$  variables independientes y  $n$  observaciones  $y_1, y_2, \dots, y_n$ .

Donde obtendríamos que los Betas soluciones al sistema de ecuaciones sería:

Se debe utilizar esta ecuación para encontrar los Betas solución de los siguientes set de datos

## puntos a evaluar:

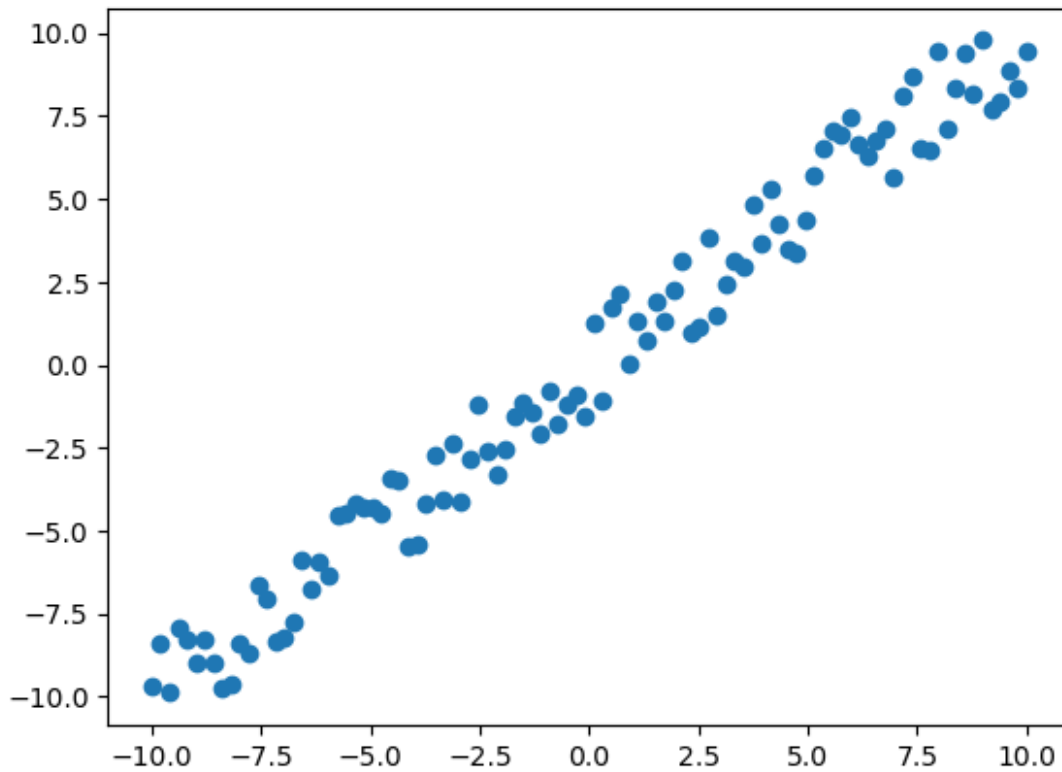
- Aplicar el modelo de regresión lineal matricial
- Encontrar Betas
- Graficar la solución (predicciones) en conjunto con el set de datos

```
import random
import math
import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm
import numpy as np
```

## punto 1

```
data_x = np.linspace(-10,10,100)
data_y = np.array( [ x+(random.random()-0.5)*3 for x in data_x ] )
plt.scatter(data_x, data_y)

<matplotlib.collections.PathCollection at 0x198afda9be0>
```

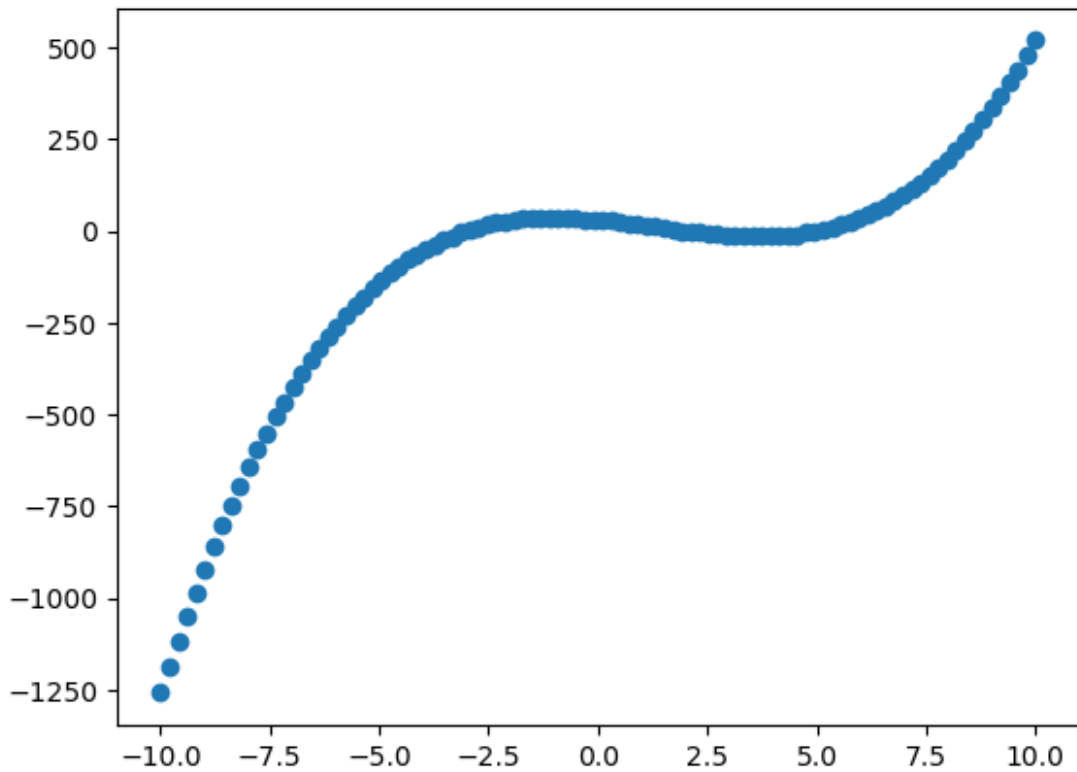


```
# aqui tu implementacion
```

## punto 2

```
data_x = np.linspace(-10,10,100)
data_y = np.array( [ (x-2)*(x-5)*(x+3)+(random.random()-0.5)*3 for x
in data_x ] )
plt.scatter(data_x, data_y)
```

```
<matplotlib.collections.PathCollection at 0x198afdff470>
```

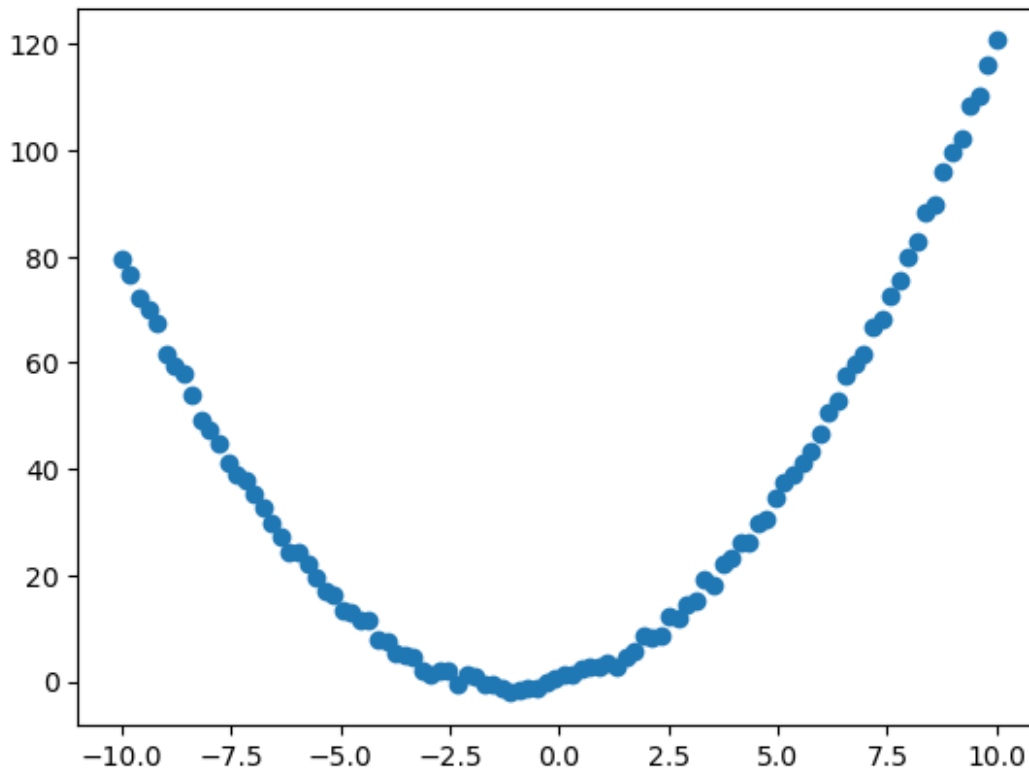


```
# aqui tu implementacion
```

## punto 3

```
data_x = np.linspace(-10,10,100)
data_y = np.array( [ x*(x+2)+(random.random()-0.5)*3 for x in data_x ]
)
plt.scatter(data_x, data_y)

<matplotlib.collections.PathCollection at 0x198afeae450>
```



```
# aqui tu implementacion
```