

# Guide 4 linear regression

September 26, 2018

## 1 Computación II - 2018

### Guía No. 4: regresión lineal

Ingeniería en Estadística - Universidad de Valparaíso

Profesor: Eduardo Jorquera - [eduardo.jorquera@postgrado.uv.cl](mailto:eduardo.jorquera@postgrado.uv.cl)

### 1.1 1. Regresión lineal simple

#### 1.1.1 ¿Qué es una regresión lineal?

Una regresión lineal es la forma estadística de asociar una o más variables a una variable “respuesta”, donde la variable respuesta depende de las demás, es decir, variables explicativas.

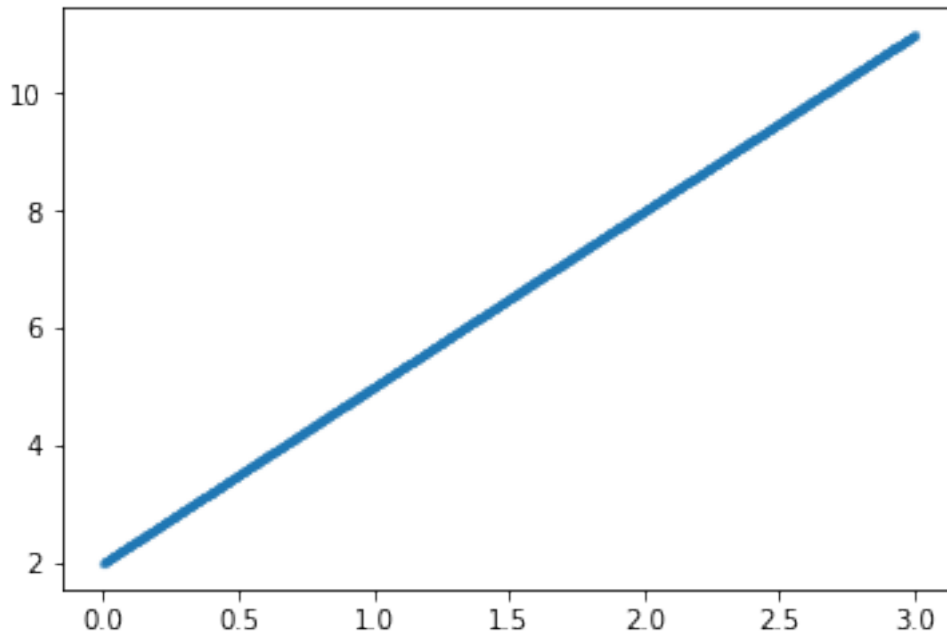
Para empezar, asumamos que entre dos variables, donde una depende de la otra, busca escribir matemáticamente una en función de la otra, a través de una relación lineal. La primera aproximación, podría ser algo así:

$$y_i = \beta_1 + \beta_2 x_i \quad (1)$$

Veamos un gráfico de esto a continuación:

```
In [10]: import numpy as np, matplotlib.pyplot as plt
         a, b = 2, 3
         x=np.linspace(0,3,1000)
         y=a+b*x
         plt.scatter(x, y, s=5, alpha=.5)
```

```
Out[10]: <matplotlib.collections.PathCollection at 0x7f3488c1a4a8>
```



Esto se parece a una recta, ¿no? bueno, dado que una variable aleatoria nunca se comportará de forma determinista, ahora podríamos asumir que hay un error aleatorio con alguna distribución de probabilidad, el cual perturba la recta anteriormente descrita:

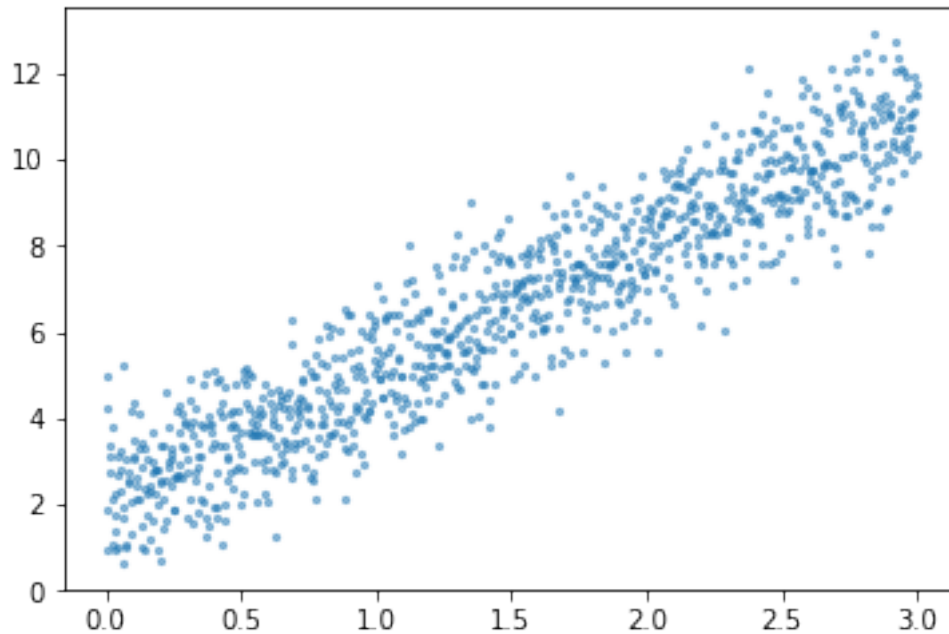
$$y_i = \beta_1 + \beta_2 x_i + \varepsilon \quad (2)$$

¡Perfecto! ahora grafiquémoslo:

- Pregunta: ¿Cuál es la varianza de  $\varepsilon$ ?

```
In [11]: import numpy as np, matplotlib.pyplot as plt
         a, b = 2, 3
         x=np.linspace(0,3,1000)
         y=a+b*x+np.random.randn(1000)
         plt.scatter(x, y, s=5, alpha=.5)
```

```
Out[11]: <matplotlib.collections.PathCollection at 0x7f3488bcbe80>
```



Ahora que tenemos los datos con esa estructura, ¿cómo se puede hacer una predicción? con la función `linear_model` del paquete `sklearn`, veamos el siguiente ejemplo:

```
In [24]: from sklearn import linear_model
         from sklearn.metrics import mean_squared_error, r2_score
         import numpy as np
         import matplotlib.pyplot as plt
```

```
In [25]: x=np.linspace(0,1,1000)
         y=x+.1*np.random.randn(1000)
         x=x.reshape(len(x),1)
         y=y.reshape(len(y),1)
```

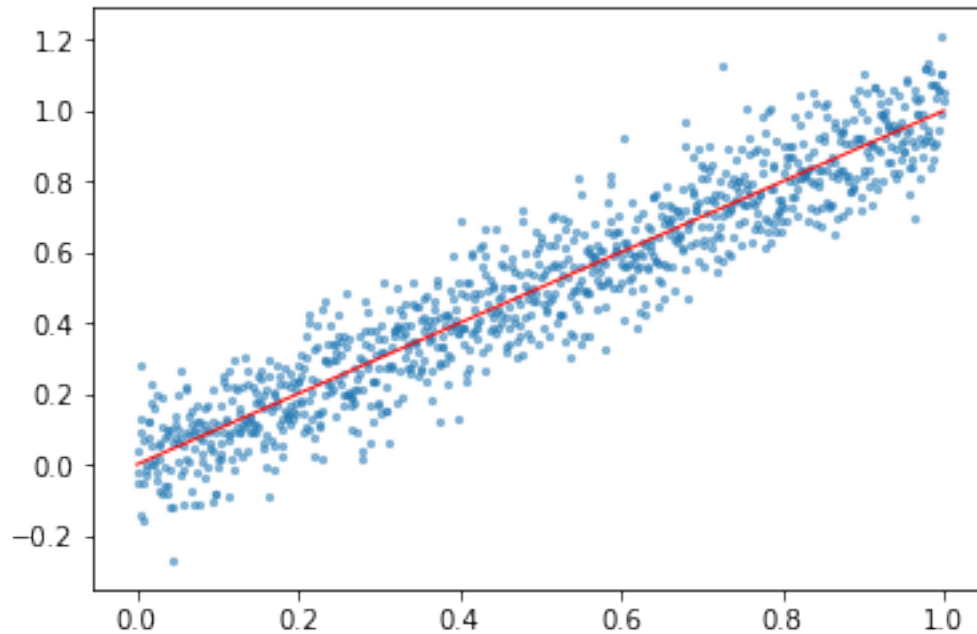
```
In [29]: regr = linear_model.LinearRegression(fit_intercept=True)

         # Train the model using the training sets
         regr.fit(x,y)
```

```
Out[29]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

```
In [30]: plt.scatter(x, y, s=5, alpha=.5)
         plt.plot(x, regr.predict(x), color='red',linewidth=1)
```

```
Out[30]: [<matplotlib.lines.Line2D at 0x7f3482276d30>]
```



Otra forma de hacer la regresión, la cual es útil usando pandas, es usando el paquete statsmodels:

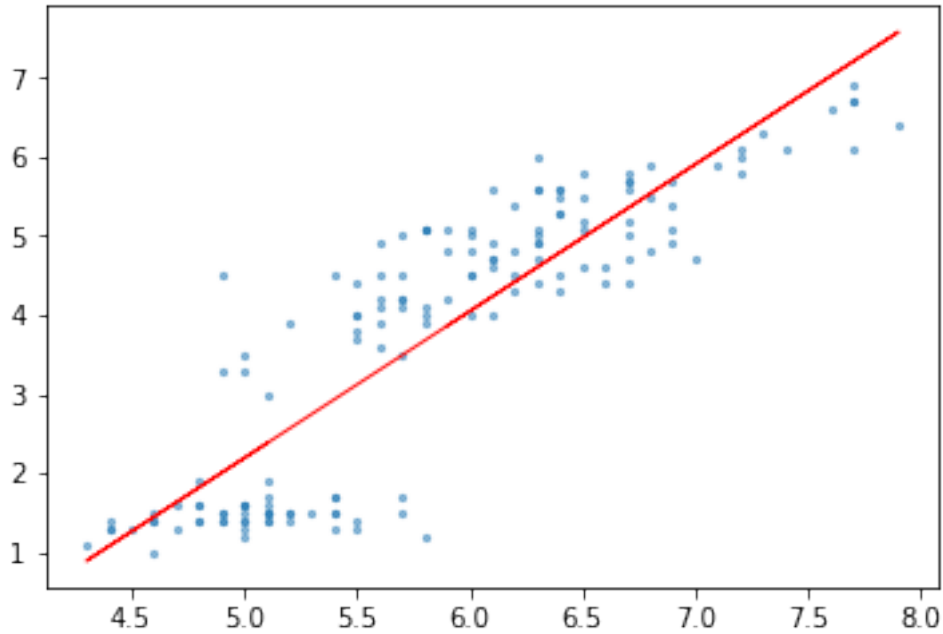
```
In [60]: import pandas as pd
import statsmodels.formula.api as sm

In [74]: import pandas as pd
iris = pd.read_csv("iris.csv")

resultado = sm.ols(formula= "Petal_Length ~ Sepal_Length" , data=iris).fit()
resultado.params
x=iris.Sepal_Length
y=iris.Petal_Length

plt.scatter(x, y, s=5, alpha=.5)
plt.plot(iris.Sepal_Length, resultado.predict(), color='red',linewidth=1)

Out[74]: [<matplotlib.lines.Line2D at 0x7f34821639e8>]
```



## 1.2 1.2 Actividad

- Simule la siguiente ecuación en un objeto de pandas:

$$y_i = \frac{\pi}{3} + 5x_i + \varepsilon \quad (3)$$

donde  $\varepsilon \sim \text{Normal}(0,3)$

- Obtenga los parámetros estimados y haga un gráfico de la estimación.
- ¿Cómo podría estimar algún tipo de error en el modelo?

In [ ]:

## 1.3 1.3 Actividad

- Simule 1.000 datos en el cuadrado  $[0,3] \times [0,3]$ , y simule el siguiente modelo de regresión lineal simple:

$$y_i = e^\pi + e^{\pi+1}x_i^2 + \varepsilon \quad (4)$$

- Similar a la actividad anterior, obtenga los parámetros estimados y haga un gráfico de la estimación.

In [ ]:

## 1.4 1.4 Tarea

- Utilizando el conjunto de datos iris, sólo para la especie setosa ajuste una regresión entre la variable Petal\_Length en función de Sepal\_Width.
- Replique el punto anterior en R.

In [ ]: