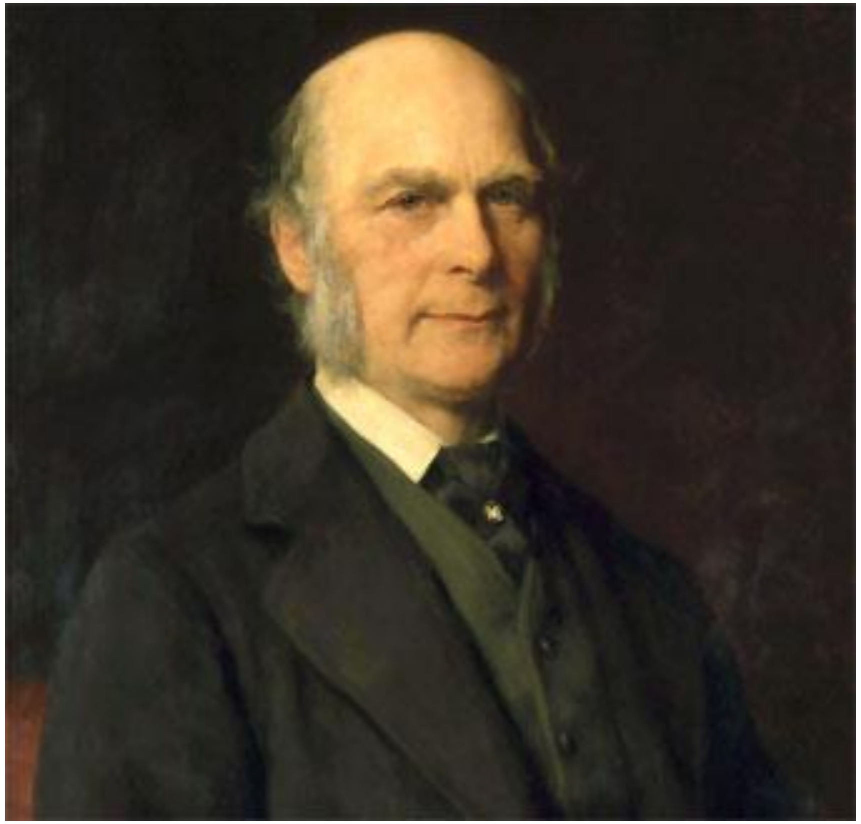
Introducción a la Regresión Lineal

Lectura Sugerida

- Capítulos 2 y 3 del libro "Introduction to Statistical Learning" de Gareth James

Historia

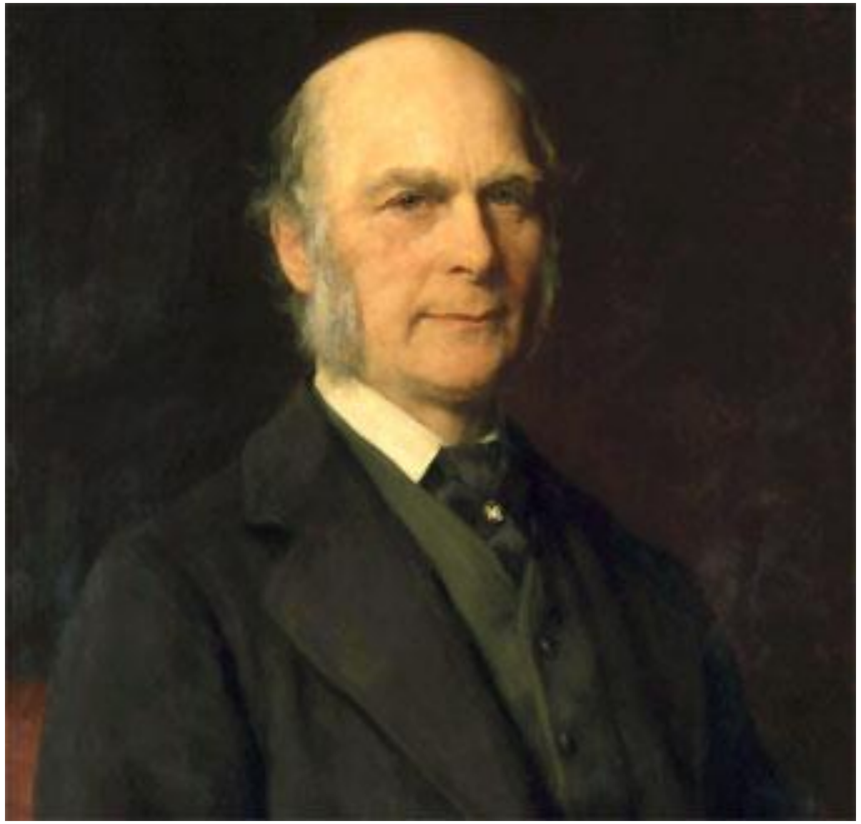
Todo comenzó en el siglo XIX con un tipo llamado Francis Galton. Galton estaba estudiando la relación entre los padres y sus hijos. En particular, investigó la relación entre las alturas de los padres y sus hijos.



Historia

Lo que descubrió fue que el hijo de cualquier hombre tendía a ser más o menos tan alto como su padre.

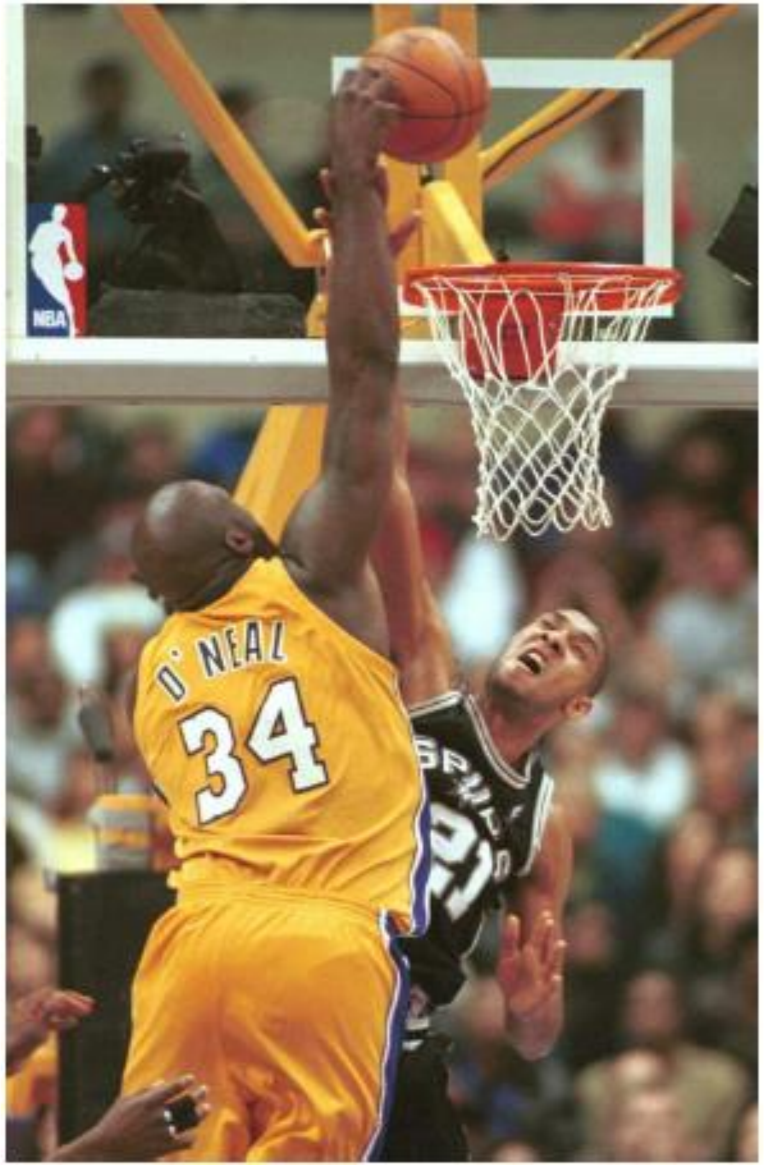
Sin embargo, el descubrimiento de Galton fue que la altura de un hijo tendía a estar más cerca de la estatura promedio general de todas las personas.



Ejemplo

Tomemos a Shaquille O'Neal como ejemplo. Shaq es realmente alto: 7 pies 1 pulgada (2,16 metros).

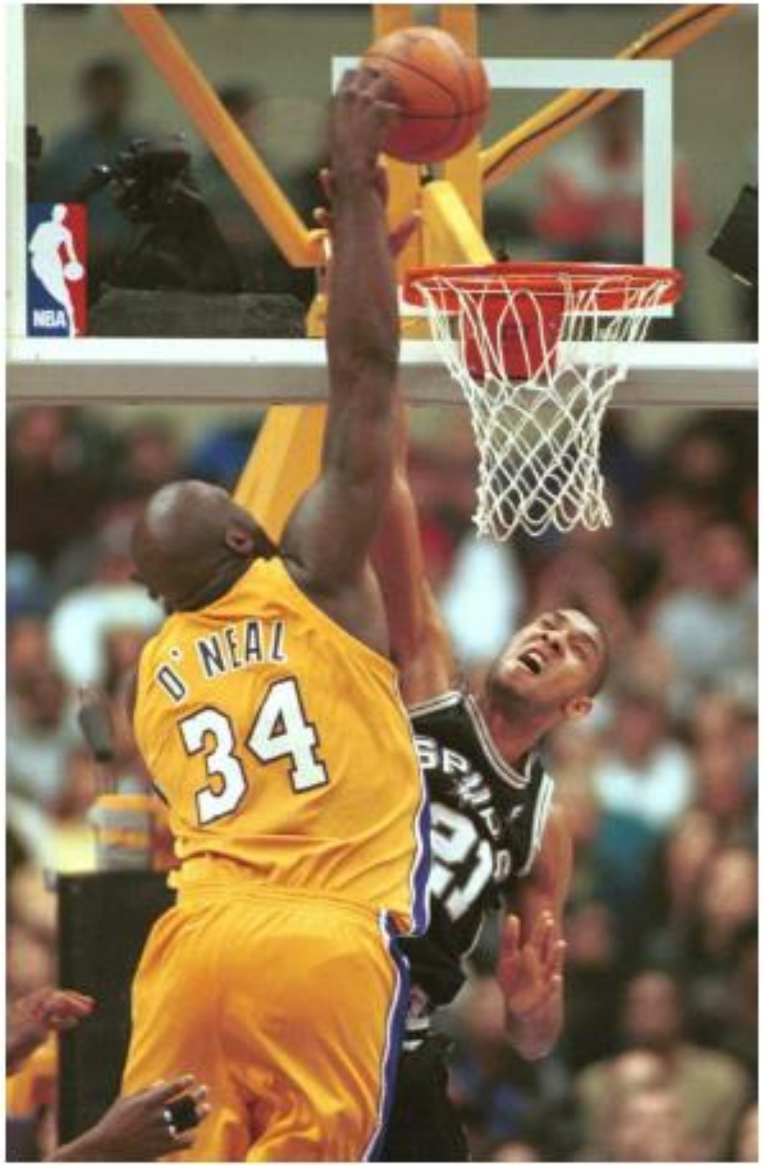
Si Shaq tiene un hijo, es probable que sea bastante alto también. Sin embargo, Shaq es una anomalía, tal que también hay una gran posibilidad de que su hijo no sea tan alto como Shaq



Ejemplo

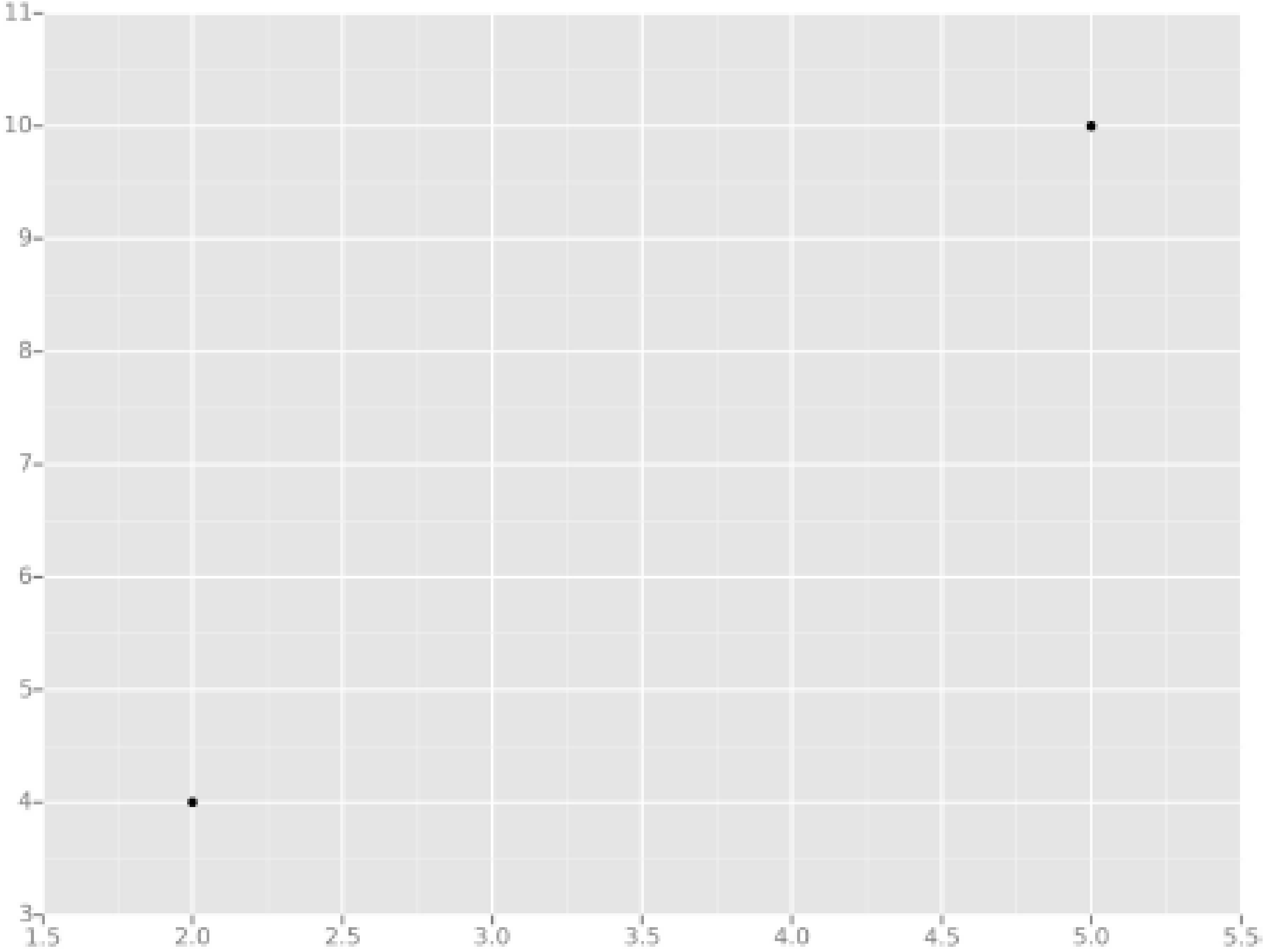
Resulta que este es el caso: el hijo de Shaq es bastante alto, 6 pies 7 pulgadas (2,0 metros), pero no tan alto como su padre.

Galton llamó a este fenómeno regresión, como en "La altura de un hijo de un padre tiende a retroceder (o deriva hacia) la altura media (promedio)".



Ejemplo

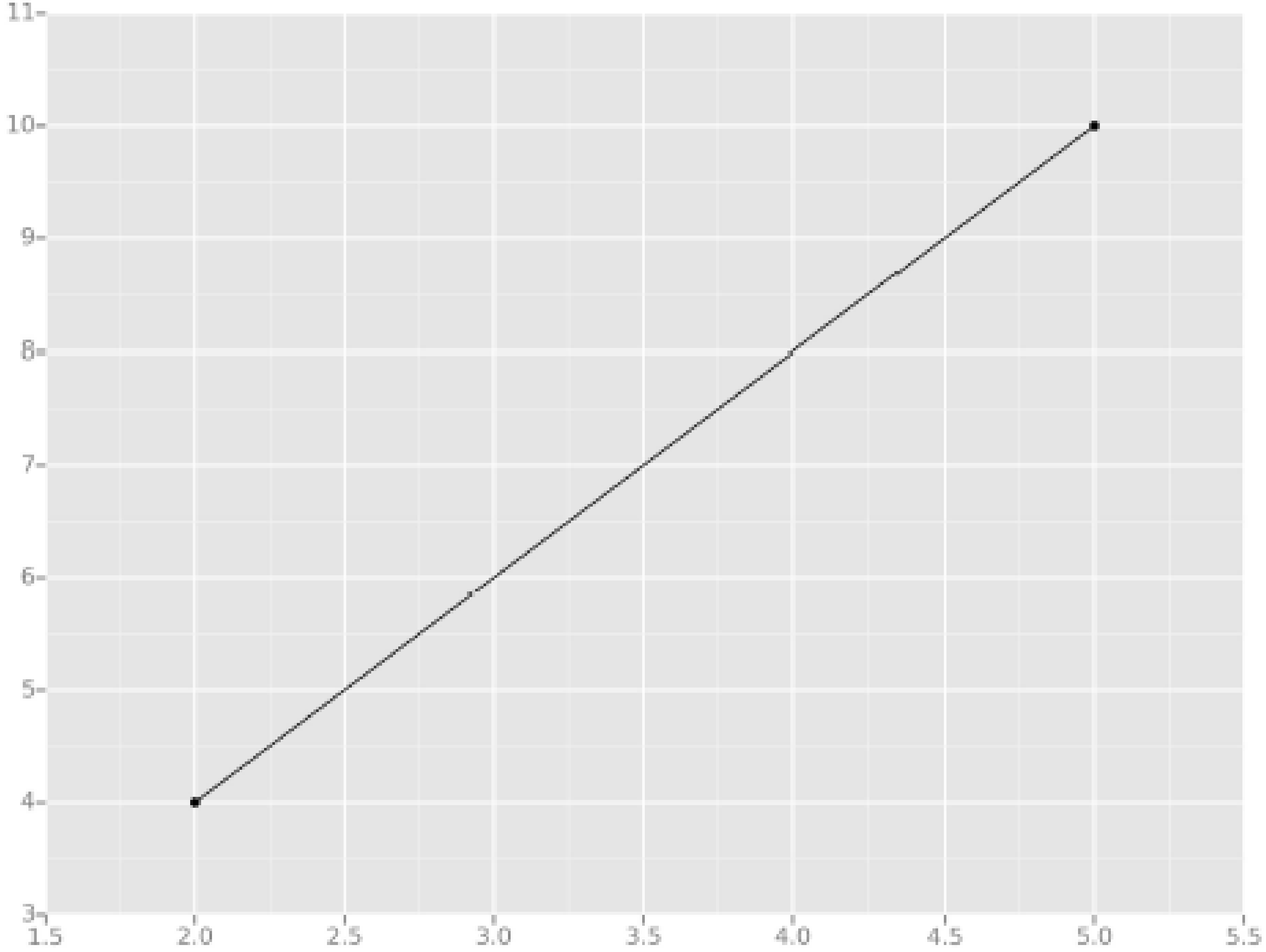
Tomemos el ejemplo más simple posible: calcular una regresión con solo 2 puntos de datos.



Ejemplo

Todo lo que intentamos hacer cuando calculamos nuestra línea de regresión es dibujar una línea lo más cercana posible a cada punto.

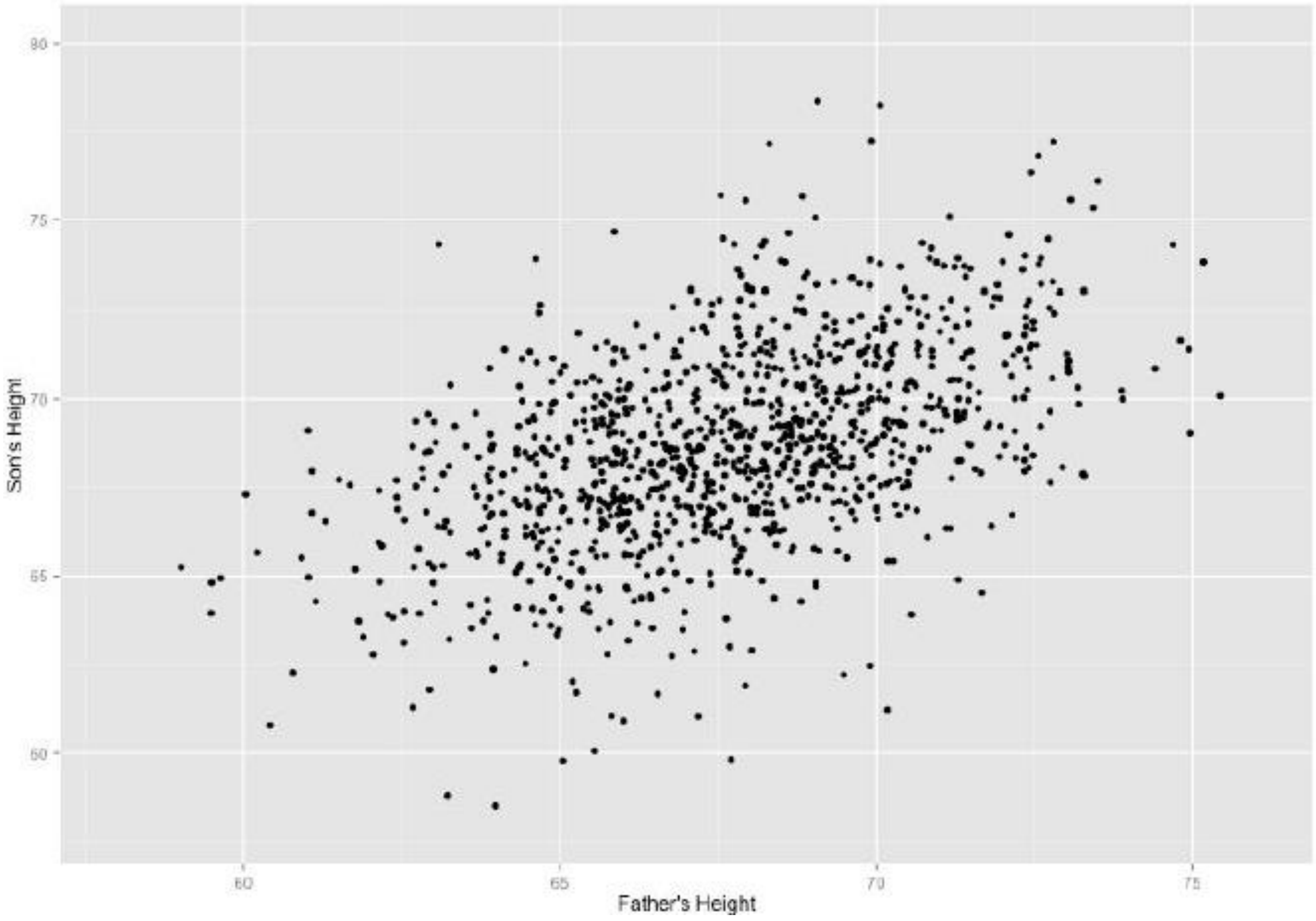
Para la regresión lineal clásica, o el "Método de mínimos cuadrados", solo se mide la cercanía en la dirección "arriba y abajo"



Ejemplo

Ahora, ¿no sería genial si pudiéramos aplicar este mismo concepto a un gráfico con más de dos puntos de datos?

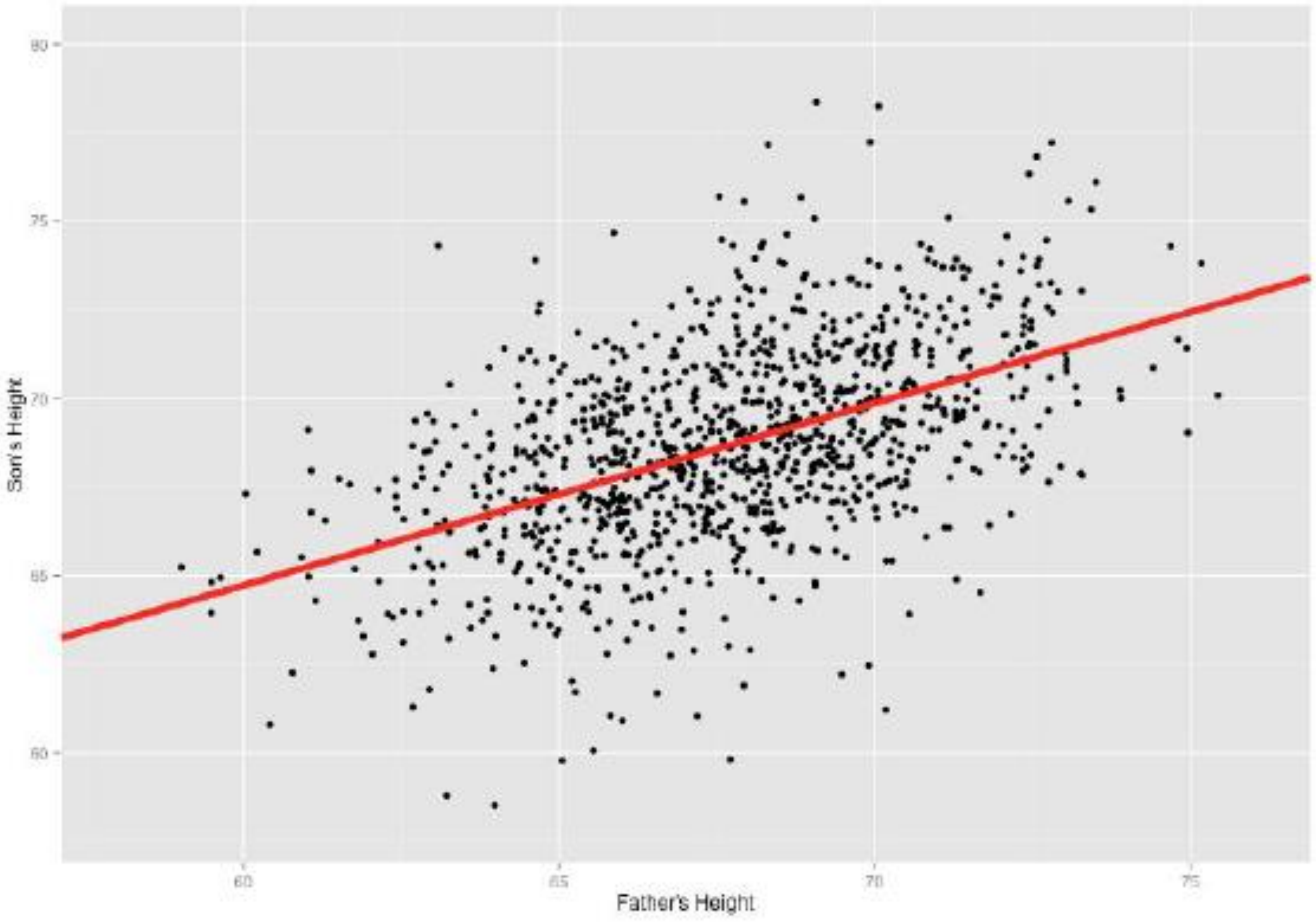
Al hacer esto, podríamos tomar múltiples hombres y las alturas de su hijos y hacer cosas como decirle a un hombre lo alto que esperamos que sea su hijo ... ¡incluso antes de que tenga un hijo!



Ejemplo

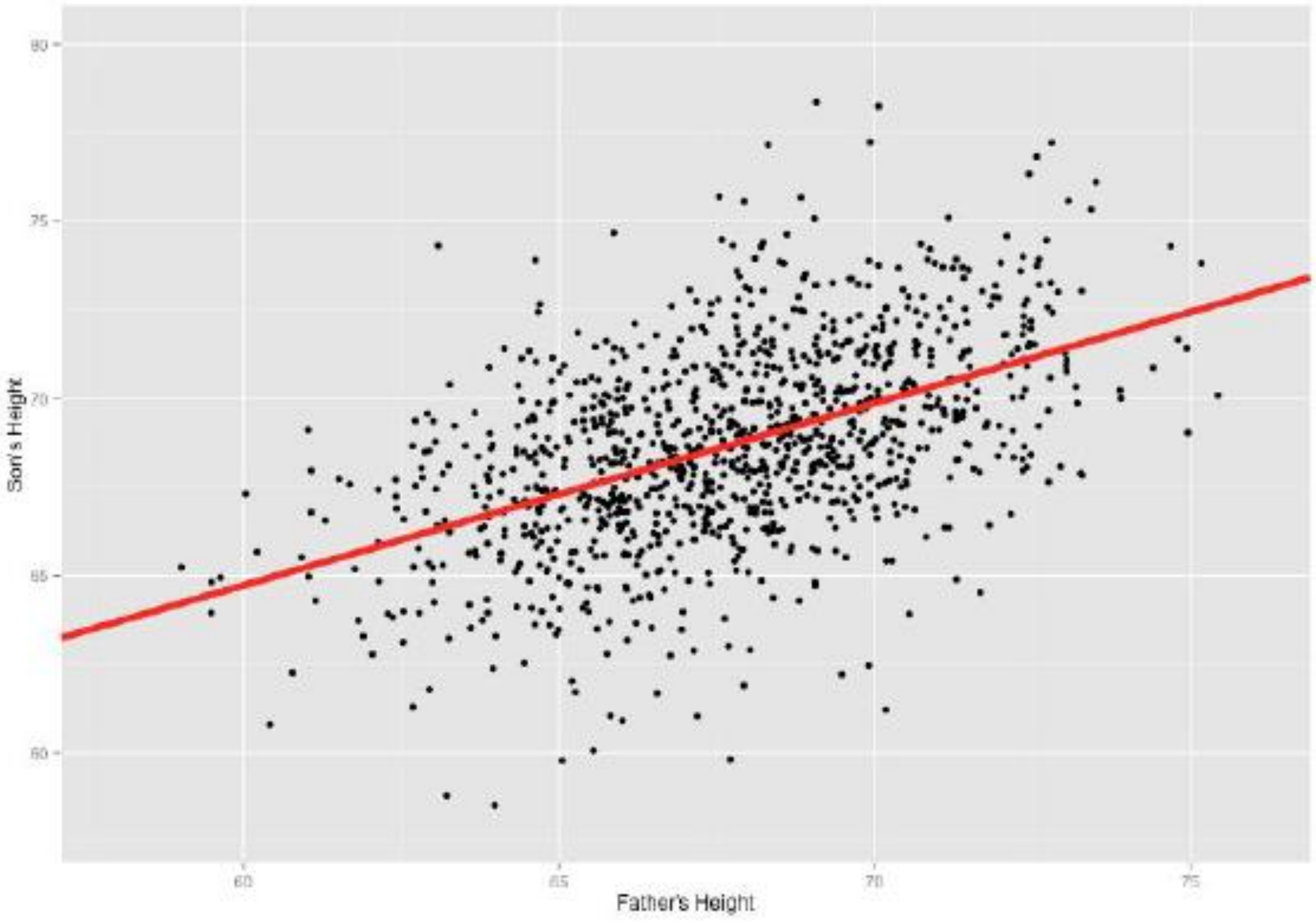
Nuestro objetivo con la regresión lineal es minimizar la distancia vertical entre todos los puntos de datos y nuestra línea.

Entonces, al determinar la mejor línea, intentamos minimizar la distancia entre todos los puntos y su distancia a nuestra línea.



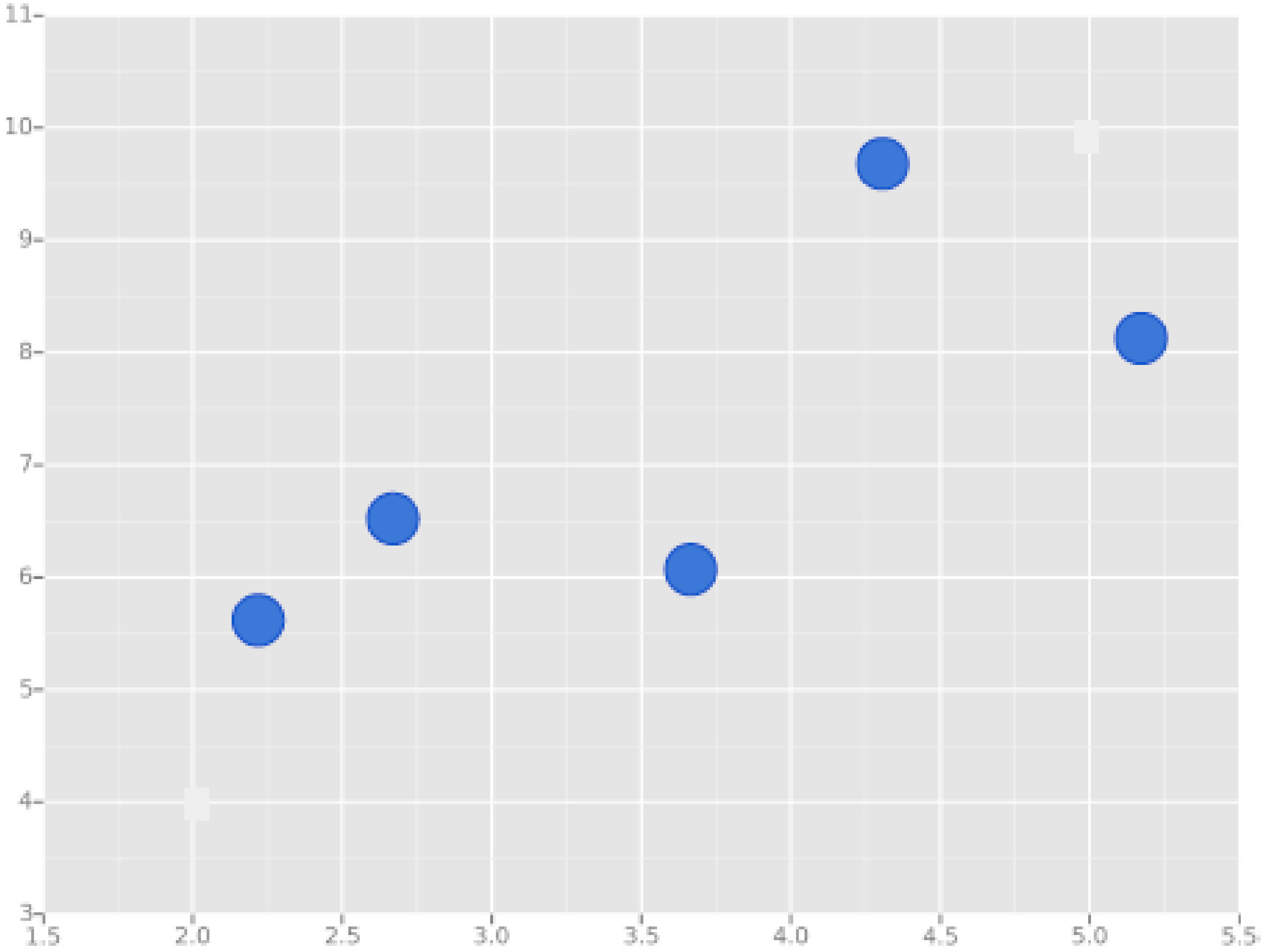
Ejemplo

Hay muchas formas diferentes de minimizar esto (suma de errores al cuadrados, suma de errores absolutos, etc.), pero todos estos métodos tienen el objetivo general de minimizar esta distancia.



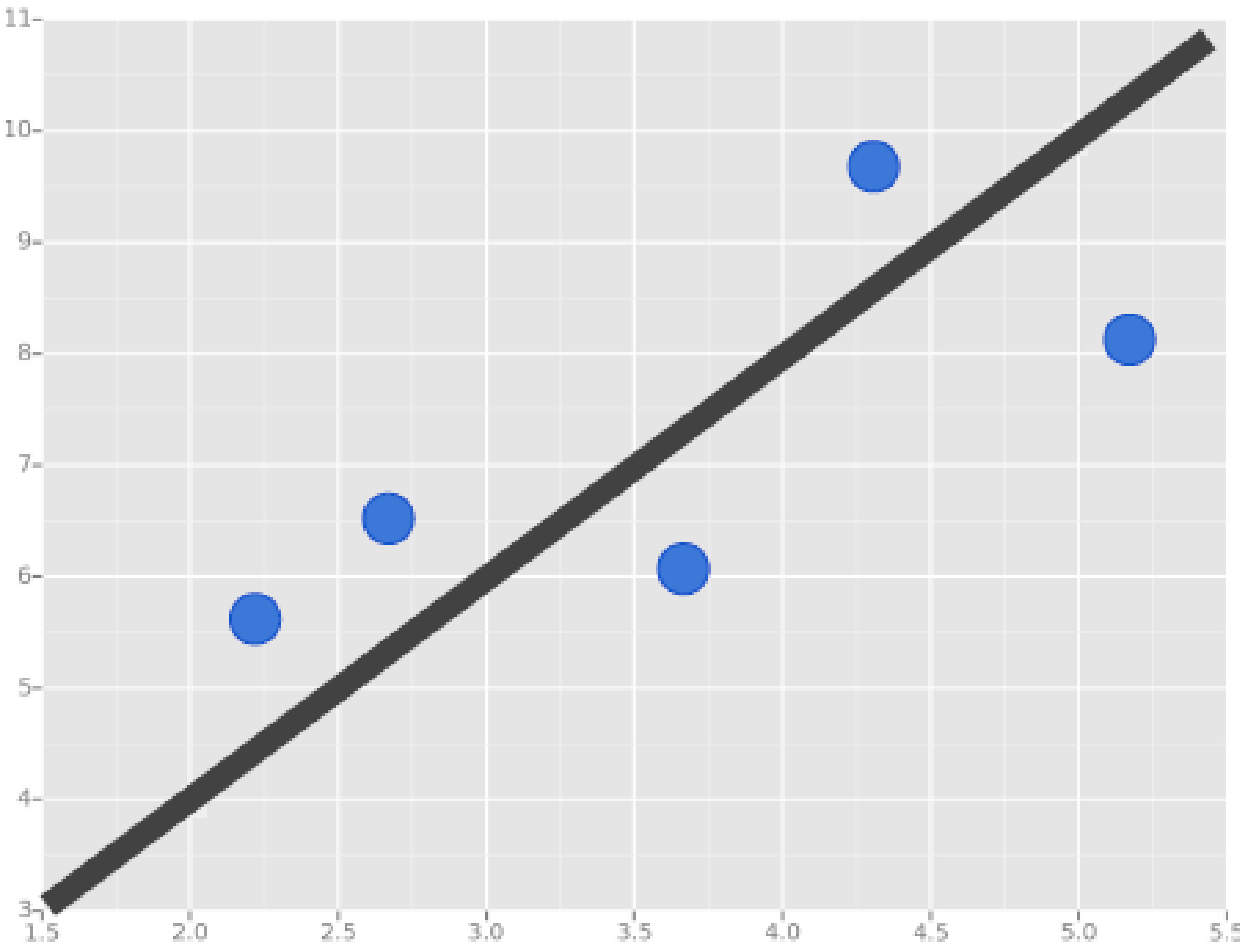
Ejemplo

Por ejemplo, uno de los métodos más populares es el método de mínimos cuadrados. Aquí tenemos puntos de datos azules a lo largo de un eje x e y.



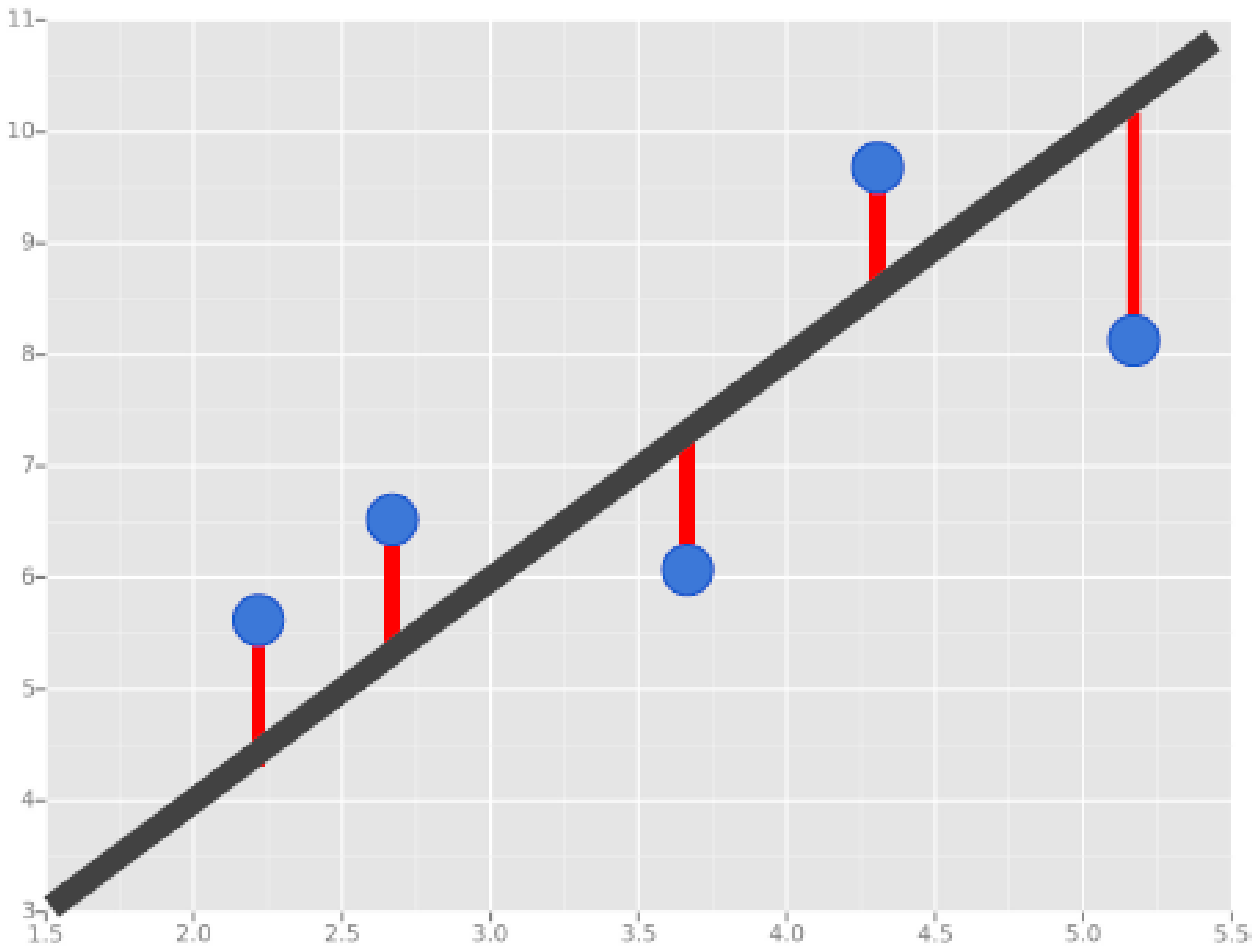
Ejemplo

Ahora queremos ajustar una línea de regresión lineal. La pregunta es, ¿cómo decidimos qué línea es la más adecuada?



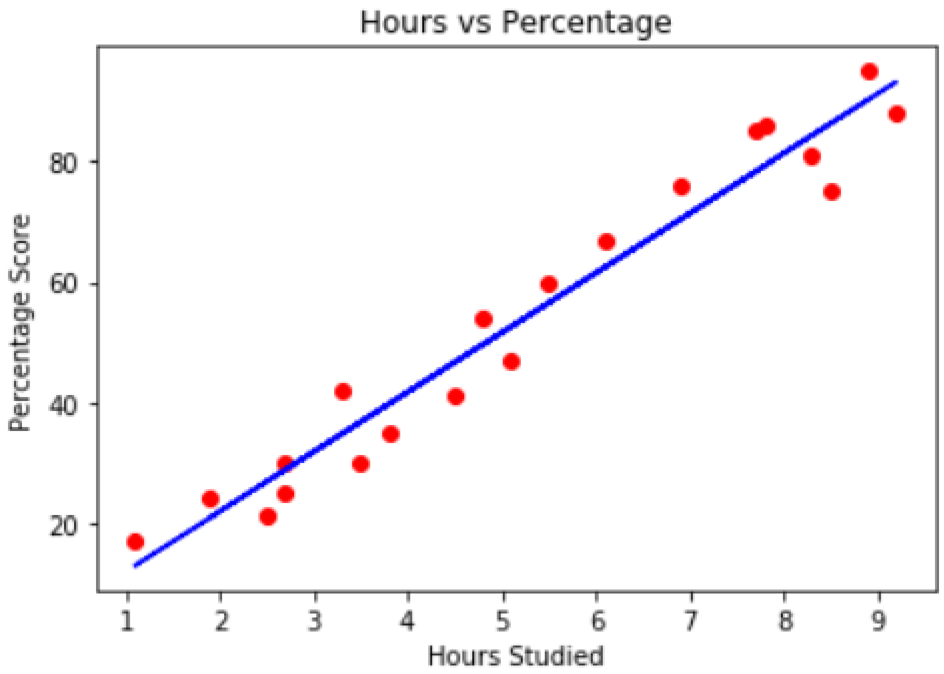
Ejemplo

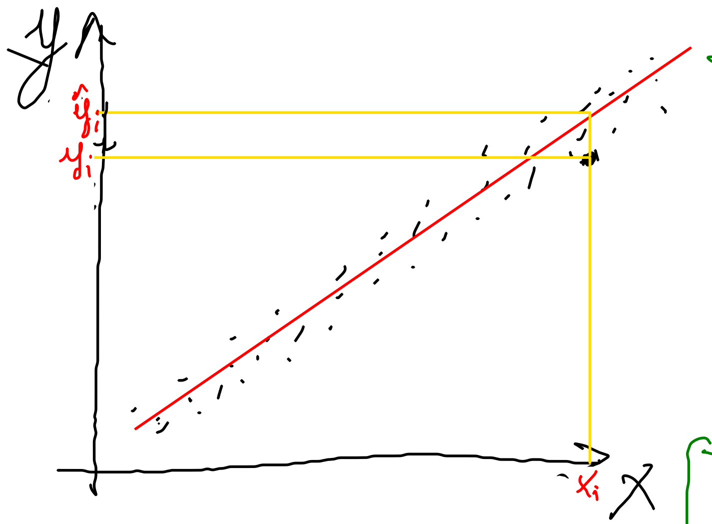
Utilizaremos el método de mínimos cuadrados, que se ajusta minimizando la suma de cuadrados de los residuos. Los residuos para una observación son la diferencia entre la observación (el valor y) y la línea ajustada.



Ejemplo con Python

- Ahora usaremos SciKit-Learn y Python para crear un modelo de regresión lineal. Luego resolverá un ejercicio propuesto y revisaremos las soluciones.





$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

$\hat{y}_i$: el valor predicho para x_i
y_i: el valor verdadero para x_i

Para minimizar las distancias verticales de cada punto a la línea podemos usar varios algoritmos

Permiten encontrar los mejores valores para β_0, β_1 del modelo

X	y
x_1	y_1
x_2	y_2
⋮	⋮
x_n	y_n

Uno de los algoritmos es el Método de Mínimos cuadrados

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

error

$$e_i = y_i - \hat{y}_i$$

Suma residual de los errores al cuadrado

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

$$\boxed{\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}_i)}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

$$\boxed{\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}}$$

donde:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

y = ventas

x = inversión en publicidad en TV

B1=	0.04754
B0=	7.03259

$$\boxed{\hat{y} = 7.03259 + 0.04754x}$$

Interceptor

2102.53058313135000
RSS

RSS = 2102.53058

Tarea: determine el modelo de regresión lineal simple | para x = publicidad en radio

y = ventas

Regresión Lineal Simple

Modelo de Regresión Lineal Simple para:

- x = inversión en publicidad en TV
- y = total de ventas

Importar librerias

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Recuperar los datos y explorarlos

In [5]:

```
datos = pd.read_csv('Advertising.csv',index_col=0)
```

In [9]:

```
datos.head()
```

Out[9]:

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

In [7]:

```
datos.tail()
```

Out[7]:

	TV	Radio	Newspaper	Sales
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

In [10]:

```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200 entries, 1 to 200
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 7.8 KB
```

In [11]:

```
datos.describe()
```

Out[11]:

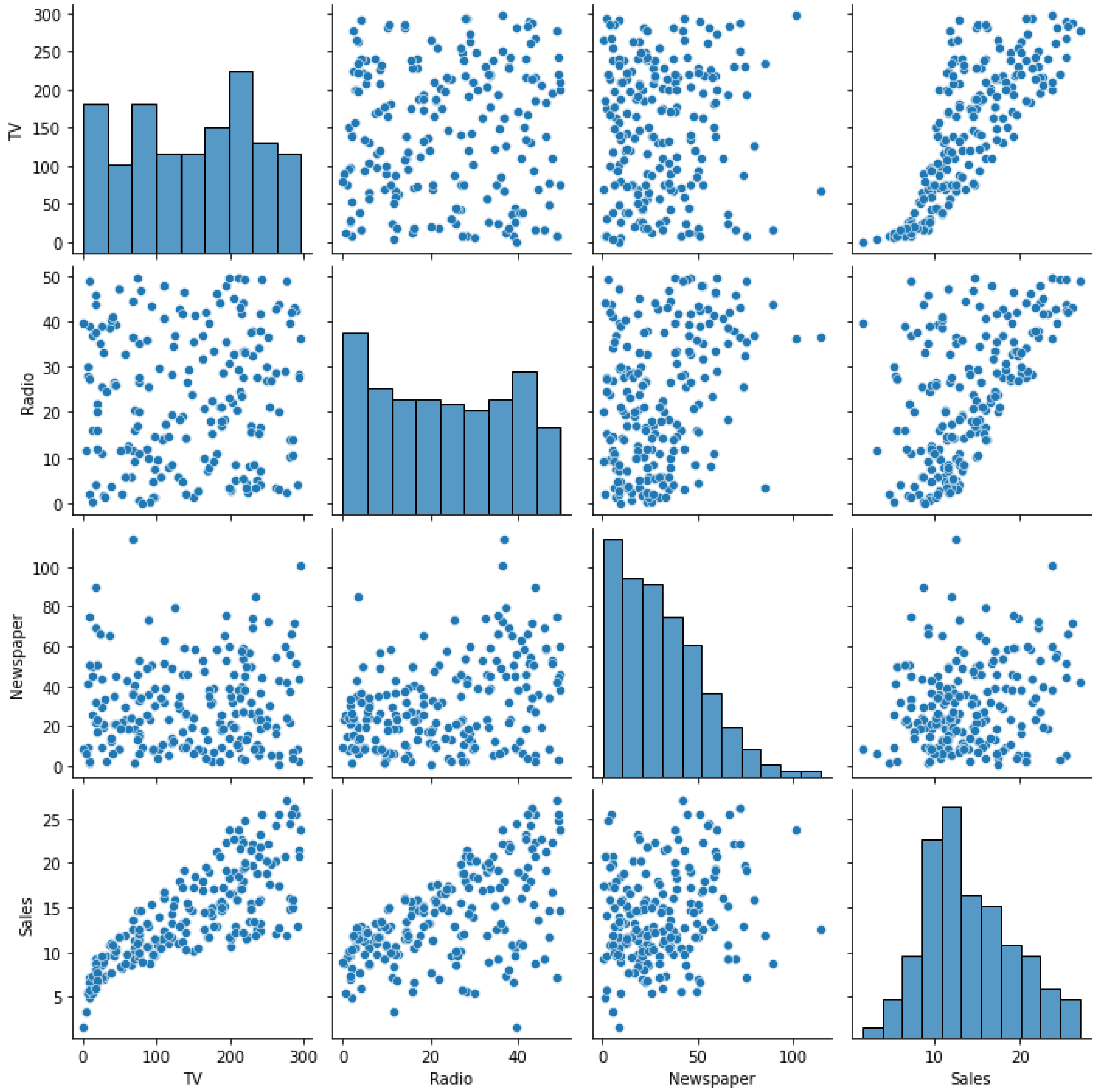
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

In [13]:

```
sns.pairplot(datos)
```

Out[13]:

```
<seaborn.axisgrid.PairGrid at 0x1ab78d19ee0>
```

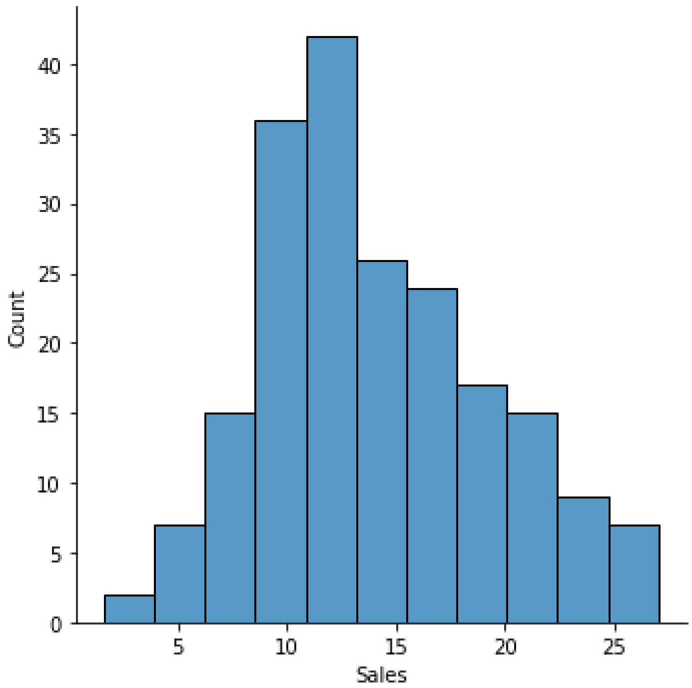


In [14]:

```python
sns.displot(datos['Sales'])
```

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x1ab7a24eee0>
```



In [15]:

```python
datos.corr()
```

Out[15]:

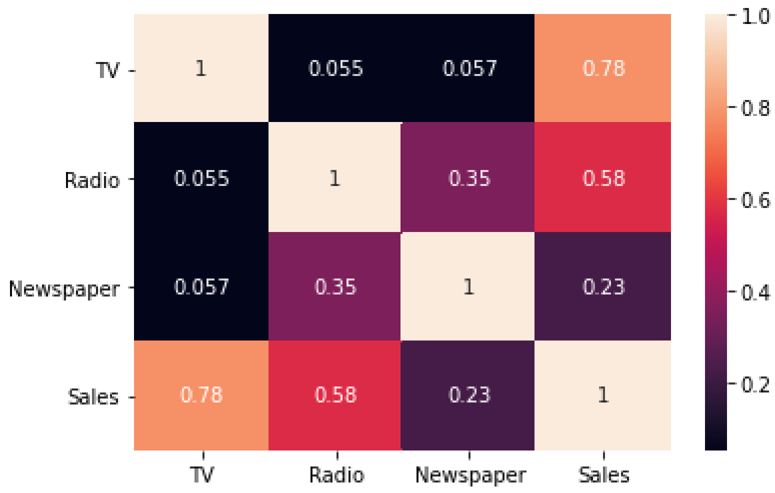
	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Newspaper	0.056648	0.354104	1.000000	0.228299
Sales	0.782224	0.576223	0.228299	1.000000

In [17]:

```
sns.heatmap(datos.corr(),annot=True)
```

Out[17]:

```
<AxesSubplot:>
```



Modelo de Regresión Lineal Simple

In [18]:

```
X = datos[['TV']]
```

In [19]:

```
y = datos['Sales']
```

Regresión Lineal con el método de mínimos cuadrados

In [20]:

```
from sklearn.linear_model import LinearRegression
```

In [21]:

```
modelo = LinearRegression()
```

In [22]:

```
modelo.fit(X,y)
```

Out[22]:

```
LinearRegression()
```

Parámetros obtenidos en el modelo

Muestra el interceptor Beta 0

In [23]:

```python
print(modelo.intercept_)
```

```
7.032593549127693
```

Muestra los otros Beta

In [24]:

```python
print(modelo.coef_)
```

```
[0.04753664]
```

Predicciones

In [25]:

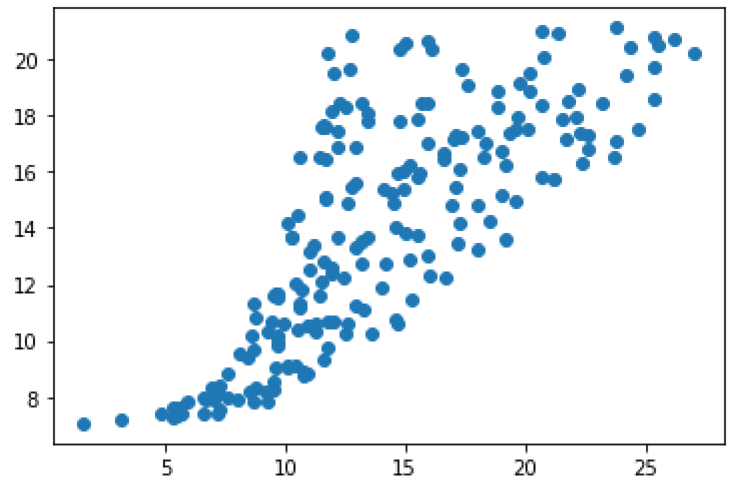
```python
predicciones = modelo.predict(X)
```

In [26]:

```python
plt.scatter(y,predicciones)
```

Out[26]:

```
<matplotlib.collections.PathCollection at 0x1ab7c438e20>
```

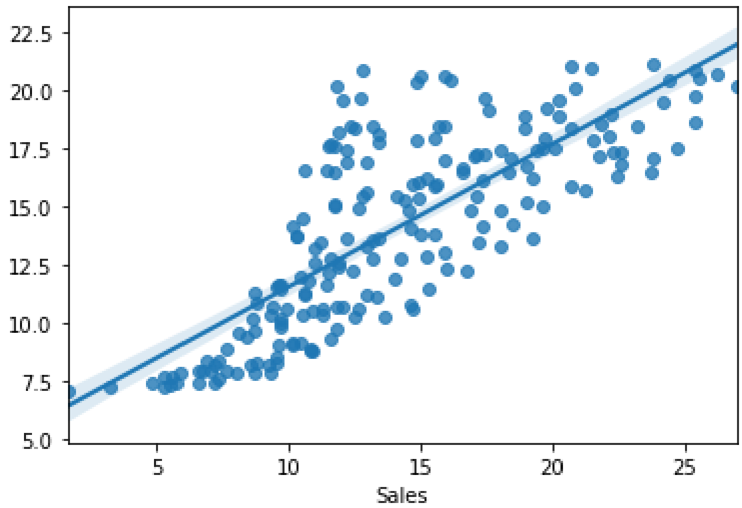


In [27]:

```
sns.regplot(x=y,y=predicciones,data=datos)
```

Out[27]:

```
<AxesSubplot:xlabel='Sales'>
```



Métricas de Evaluación

In [28]:

```
from sklearn import metrics
```

In [29]:

```
MSE = metrics.mean_squared_error(y,predicciones)
```

In [30]:

```
RSS = MSE*200
```

In [31]:

```
print(RSS)
```

```
2102.5305831313512
```