

**Example 50.** A car rental company wants to predict the annual maintenance cost  $y$  (in 100USD/year) of a car using the age  $x$  (in years) of that car (as an explanatory variable). Based on the observations  $(x, y) = (2, 1), (5, 2), (7, 3), (8, 3)$ , predict the cost for a 4.5 year old car (using linear regression).

**Solution.** Once we compute the regression line  $y = a + bx$  (we already did that:  $y = \frac{2}{7} + \frac{5}{14}x$ ), our prediction is  $\frac{2}{7} + \frac{5}{14} \cdot 4.5 = \frac{53}{28} \approx 1.89$ , that is, 189 USD/year.

**Application: multiple linear regression**

In statistics, **linear regression** is an approach for modeling the relationship between a scalar dependent variable and one or more explanatory variables.

The case of one explanatory variable is called simple linear regression.

For more than one explanatory variable, the process is called multiple linear regression.

[http://en.wikipedia.org/wiki/Linear\\_regression](http://en.wikipedia.org/wiki/Linear_regression)

The experimental data might be of the form  $(x_i, y_i, z_i)$ , where now the dependent variable  $z_i$  depends on two explanatory variables  $x_i, y_i$  (instead of just  $x_i$ ).

**Example 51.** Set up a linear system to find values for the parameters  $a, b, c$  such that  $z = a + bx + cy$  best fits some given points  $(x_1, y_1, z_1), (x_2, y_2, z_2), \dots$

**Solution.** The equations  $a + bx_i + cy_i = z_i$  translate into the system:

$$\underbrace{\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ \vdots & \vdots & \vdots \end{bmatrix}}_{\text{design matrix } A} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\text{observation vector } \mathbf{z}} = \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \end{bmatrix}}_{\text{observation vector } \mathbf{z}}$$

Of course, this is usually inconsistent. To find the best possible  $a, b, c$  we compute a least squares solution by solving  $A^T A \begin{bmatrix} a \\ b \\ c \end{bmatrix} = A^T \mathbf{z}$ .

## Application: Fitting data to other curves

We can also fit the experimental data  $(x_i, y_i)$  using other curves.

**Example 52.** Set up a linear system to find values for the parameters  $a, b, c$  that result in the quadratic curve  $y = a + bx + cx^2$  that best fits some given points  $(x_1, y_1), (x_2, y_2), \dots$

**Solution.**  $y_i \approx a + bx_i + cx_i^2$  with parameters  $a, b, c$ .

The equations  $y_i = a + bx_i + cx_i^2$  in matrix form:

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ \vdots & \vdots & \vdots \end{bmatrix}}_{\text{design matrix } A} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix}}_{\text{observation vector } \mathbf{y}}$$

Again, we determine values for  $a, b, c$  by computing a least squares solution to that system.

That is, we need to solve the system  $A^T A \begin{bmatrix} a \\ b \\ c \end{bmatrix} = A^T \mathbf{y}$ .

**Example 53. (homework)** Use Sage to find values for  $a, b, c$  that result in the quadratic curve  $y = a + bx + cx^2$  that best fits the points  $(0, 1), (1, 2), (2, 3), (3, -4), (4, -7), (5, -12)$ .

**Solution.** We first input the points:

```
Sage] points = [[0,1],[1,2],[2,3],[3,-4],[4,-7],[5,-12]]
```

We set up the system described in the previous example, then determine a least-squares solution.

```
Sage] X = matrix([[1,0,0],[1,1,1],[1,2,4],[1,3,9],[1,4,16],[1,5,25]])
```

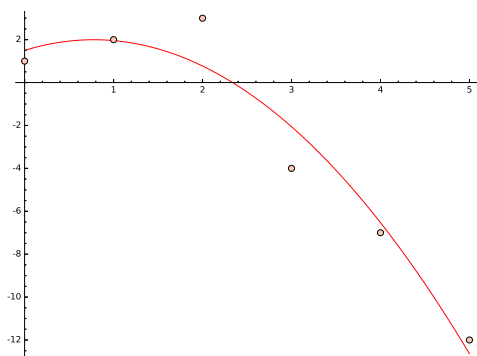
```
Sage] y = vector([1,2,3,-4,-7,-12])
```

```
Sage] (X.transpose()*X).solve_right(X.transpose()*y)
```

$$\left( \frac{3}{2}, \frac{179}{140}, -\frac{23}{28} \right)$$

Hence, the best fitting quadratic curve is  $y = \frac{3}{2} + \frac{179}{140}x - \frac{23}{28}x^2$ . Here's a plot:

```
Sage] scatter_plot(points) + plot(3/2+179/140*x-23/28*x^2,0,5,color='red')
```



**Advanced comment.** If you are comfortable with Python, you can avoid typing out  $X$  and  $y$ :  
[The plot command above now won't work anymore because we are overwriting  $x$  with numbers.]

```
Sage] X = matrix([[1,x,x^2] for x,y in points])
```

```
Sage] y = vector([y for x,y in points])
```