# 1 Interpolation

Let $(x_1, y_1), (x_2, y_2)$ and $(x_3, y_3)$ be points in the real plain $\mathbb{R}^2$. If $x_i \neq x_j$ for $i \neq j$ there is exactly one parabola $f\colon \mathbb{R} \to \mathbb{R}$ (i.e., there are uniquely determined coefficients $a, b, c \in \mathbb{R}$) which satisfies

$$f(x_i) = ax_i^2 + bx_i + c = y_i \qquad \text{for all } i = 1, 2, 3. \tag{1}$$

1. Please implement a function `interpolate(data)` which takes the points $(x_i, y_i)$ for $i = 1, 2, 3$ as arguments and returns the coefficients $a, b, c$, for which (1) holds.

2. Implement a function `parabola(x, coeff)` which evaluates a parabola depending on an argument $x$ and some coefficients $a, b, c$. Then randomly generate some data points $(x_i, y_i)$ for $i = 1, 2, 3$, determine the corresponding coefficients $a, b, c$ with `interpolate(data)` and create a plot which contains the pairs $(x_i, y_i)$ for $i = 1, 2, 3$ and the associated fitted function $f$ from (1) with the determined coefficients.

3. Assume the measuring points $x_1, x_2$ and $x_3$ are fixed and we want to obtain many interpolations for different values of $y_1, y_2, y_3$. What could be changed in your implementation to possibly make it more efficient?

*Hint:* Interpret task (i) as linear system of equations and solve it with `linalg.solve`.

**Solution:**