Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione

Corso di Laurea Triennale in Ingegneria dei Sistemi Medicali

Bioinformatics and Big Data Analytics

Regression

Eng. Nicola **Altini**, *Ph.D. Student*
Eng. Giacomo Donato **Cascarano**, *Ph.D. Student*
Prof. *Eng*. Vitoantonio **Bevilacqua**, *Ph.D.*

Anno Accademico 2019/2020

Linear Regression

- Linear regression models the relation between a dependent, or response, variable y and one or more independent, or predictor, variables $x_1, \dots, x_n$.

$$y = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n + \epsilon$$

- Simple linear regression considers only one independent variable using the relation:

$$y = \theta_0 + \theta_1 x + \epsilon$$

- θ_0 is the intercept
- θ_1 is the slope
- ϵ is the error term

Matricial form

- Consider a set of m observed values of x and y given by $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \dots, (x^{(n)}, y^{(n)})$. Using the simple linear regression relation, these values form a system of linear equations. Represent these equations in matrix form as:

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ \vdots & \vdots \\ 1 & x^{(m)} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}, \qquad X = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ \vdots & \vdots \\ 1 & x^{(m)} \end{bmatrix}, \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

We can write the matricial relation as:

$$Y = X\theta$$

Where Y is $m \times 1$, X is $m \times 2$, θ is 2×1

Matricial form

- If we have n variables, with m observations for each, we can write:

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots x_n^{(2)} \\ \vdots & \vdots & \vdots \quad \cdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots x_n^{(m)} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}, \qquad X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots x_n^{(2)} \\ \vdots & \vdots & \vdots \quad \cdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots x_n^{(m)} \end{bmatrix}, \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

We can write the matricial relation as:

$$Y = X\theta$$

Where Y is $m\times1$, X is $m\times(n+1)$, θ is $n\times1$.

Example

- In this example we will use the `accident` dataset, which is embedded in your MATLAB environment.

```
%% Dataset loading
load accidents
x = hwydata(:,14); %Population of states
y = hwydata(:,4); %Accidents per state
format long

%% Linear Regression -- no intercept
theta1 = x\y;

%% Linear Regression – with intercept
X = [ones(length(x),1) x];
theta = X\y;
```

We add a column of all ones to `x`!

Correlation Coefficient

- **Pearson's correlation coefficient** is the covariance of the two variables divided by the product of their standard deviations. The form of the definition involves a "product moment", that is, the mean (the first moment about the origin) of the product of the mean-adjusted random variables.
- For a population:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

- For a sample:

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x^{(i)}-\bar{x})(y^{(i)}-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x^{(i)}-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y^{(i)}-\bar{y})^2}}$$

Correlation Coefficient

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x^{(i)} - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y^{(i)} - \bar{y})^2}}$$

$$s_x = \sqrt{\sum_{i=1}^{n}\frac{(x^{(i)} - \bar{x})^2}{n-1}} \quad and \quad s_y = \sqrt{\frac{\sum_{i=1}^{n}(y^{(i)} - \bar{y})^2}{n-1}}$$

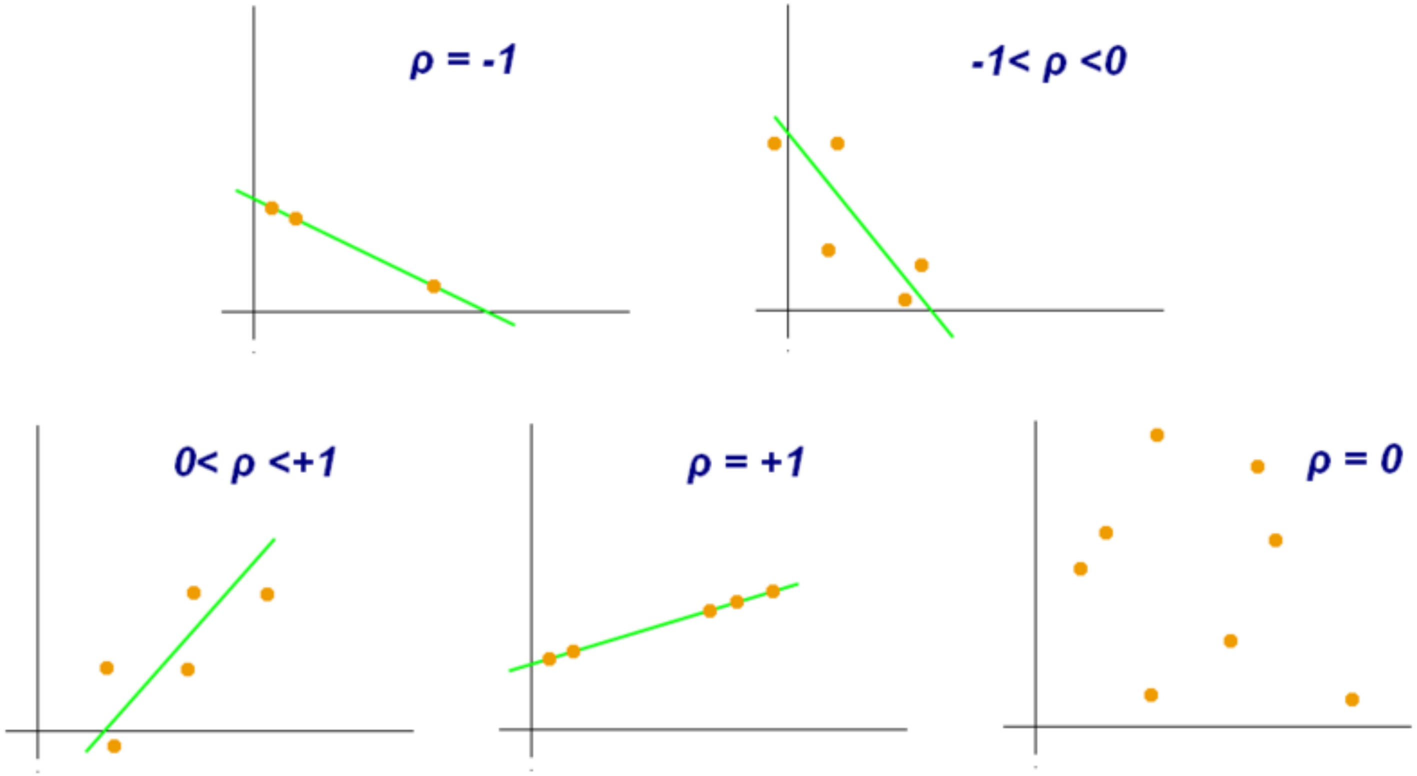
$$r_{x,y} = \frac{\sum_{i=1}^{n}\left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{(n-1)s_x s_y}$$

$$z_x^{(i)} z_y^{(i)} = \frac{\left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{s_x s_y}$$

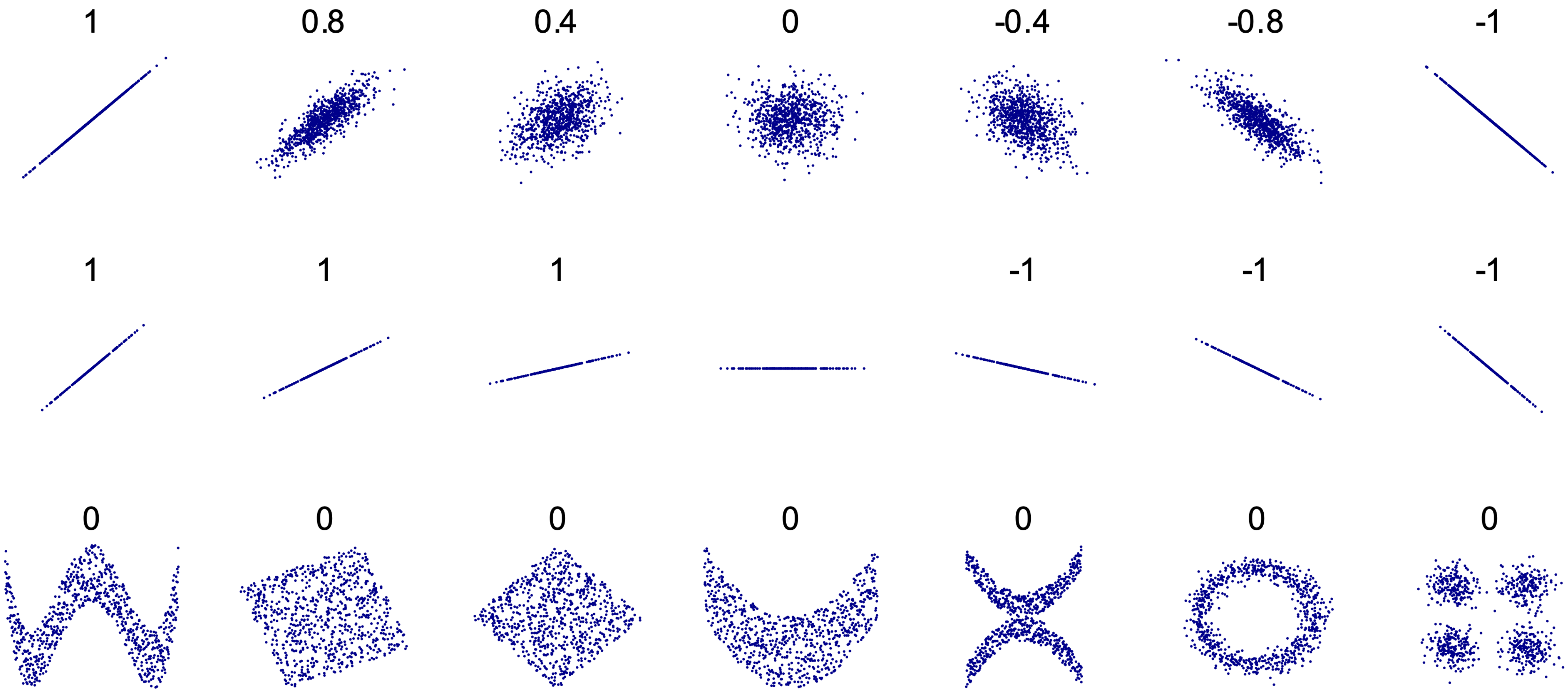
$$r_{x,y} = \frac{\sum_{i=1}^{n} z_x^{(i)} z_y^{(i)}}{n-1}$$

Correlation Coefficient

- A correlation coefficient of $r = +1$ means a perfect positive **linear** relationship between the two variables.
- A correlation coefficient of $r = -1$ means a perfect negative **linear** relationship between the two variables.
- A correlation coefficient of $r = 0$ suggests that there is no **linear** relationship between the two variables.



Correlation Coefficient



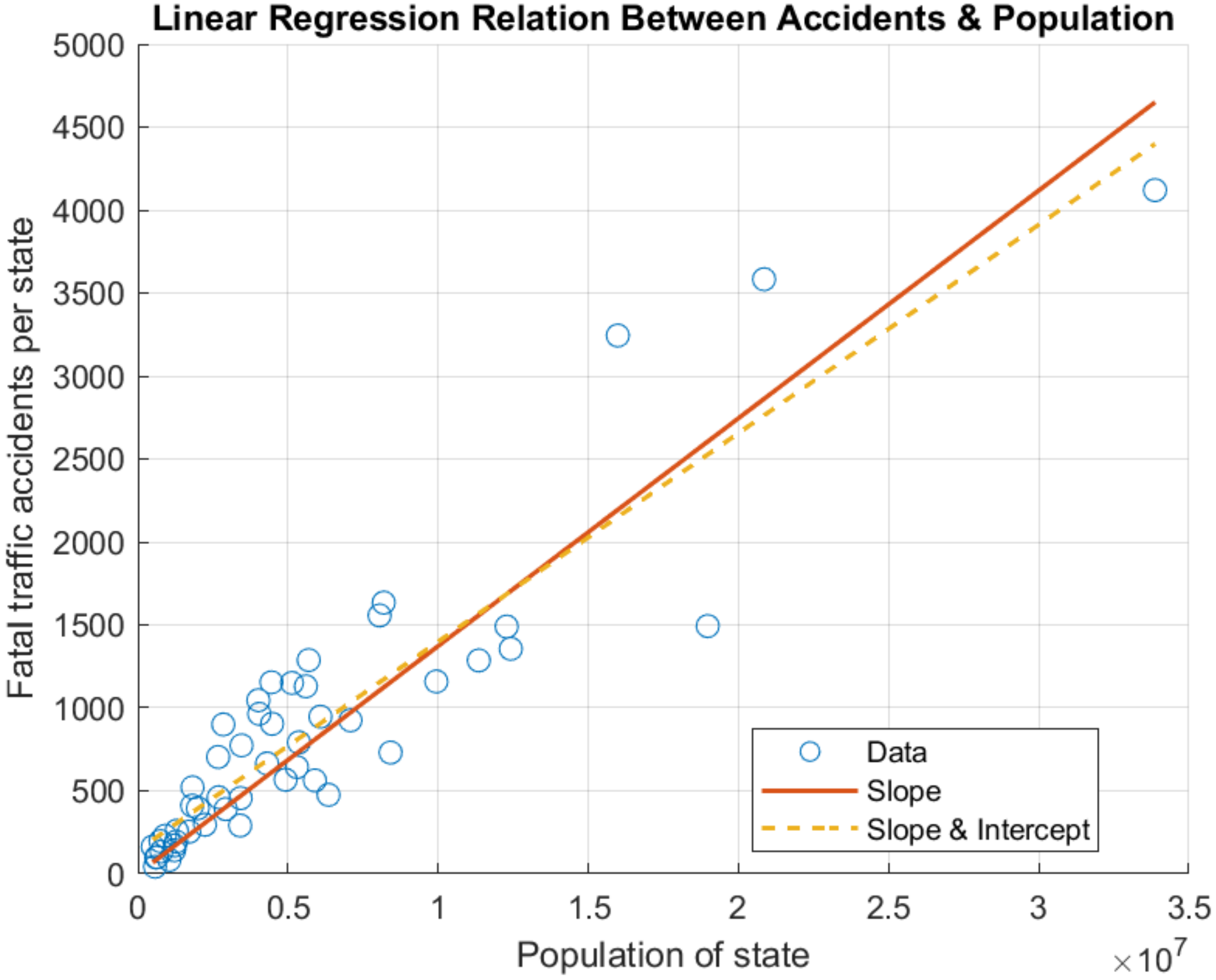
Coefficient of determination

- In statistics, the coefficient of determination, denoted R^2 (R squared), is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
- It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information.
- It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

R^2

From the figure, the two fits look similar. One method to find the better fit is to calculate the coefficient of determination, R^2.

- R^2 is one measure of how well a model can predict the data, and falls between 0 and 1.
- The higher the value of R^2, the better the model is at predicting the data.



R^2

- $y^{(i)}$ are the sampled values, $y^{*(i)}$ are the calculated values, $\overline{y}$ is the mean of y
- Total sum of squares (proportional to variance of data):

$$\sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right)^2$$

- Sum of squares of residuals:

$$\sum_{i=1}^{n} \left(y^{(i)} - y^{*(i)}\right)^2 = \sum_{i=1}^{n} e^{(i)^2}$$

- Then R^2 is the sum of squared residuals over the total sum of squares:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y^{(i)} - y^{*(i)}\right)^2}{\sum_{i=1}^{n} \left(y^{(i)} - \overline{y}\right)^2}$$

```
function Rsq = calcRsq(y, yCalc)
    Rsq = 1 - sum((y - yCalc).^2)/sum((y - mean(y)).^2);
end
```

Polynomial curve fitting

- Polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted $E(y \mid x)$.
- `p = polyfit(x,y,n)` returns the coefficients for a polynomial $p(x)$ of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is $n + 1$.

$$p(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1}$$

Polynomial evaluation

- `y = polyval(p,x)` evaluates the polynomial p at each point in x. The argument p is a vector of length $n+1$ whose elements are the coefficients (in descending powers) of an nth-degree polynomial:

$$p(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1}$$

```
p = [3 2 1];
x = [5 7 9];
y = polyval(p,x)
y = 1×3
    86   162   262
```

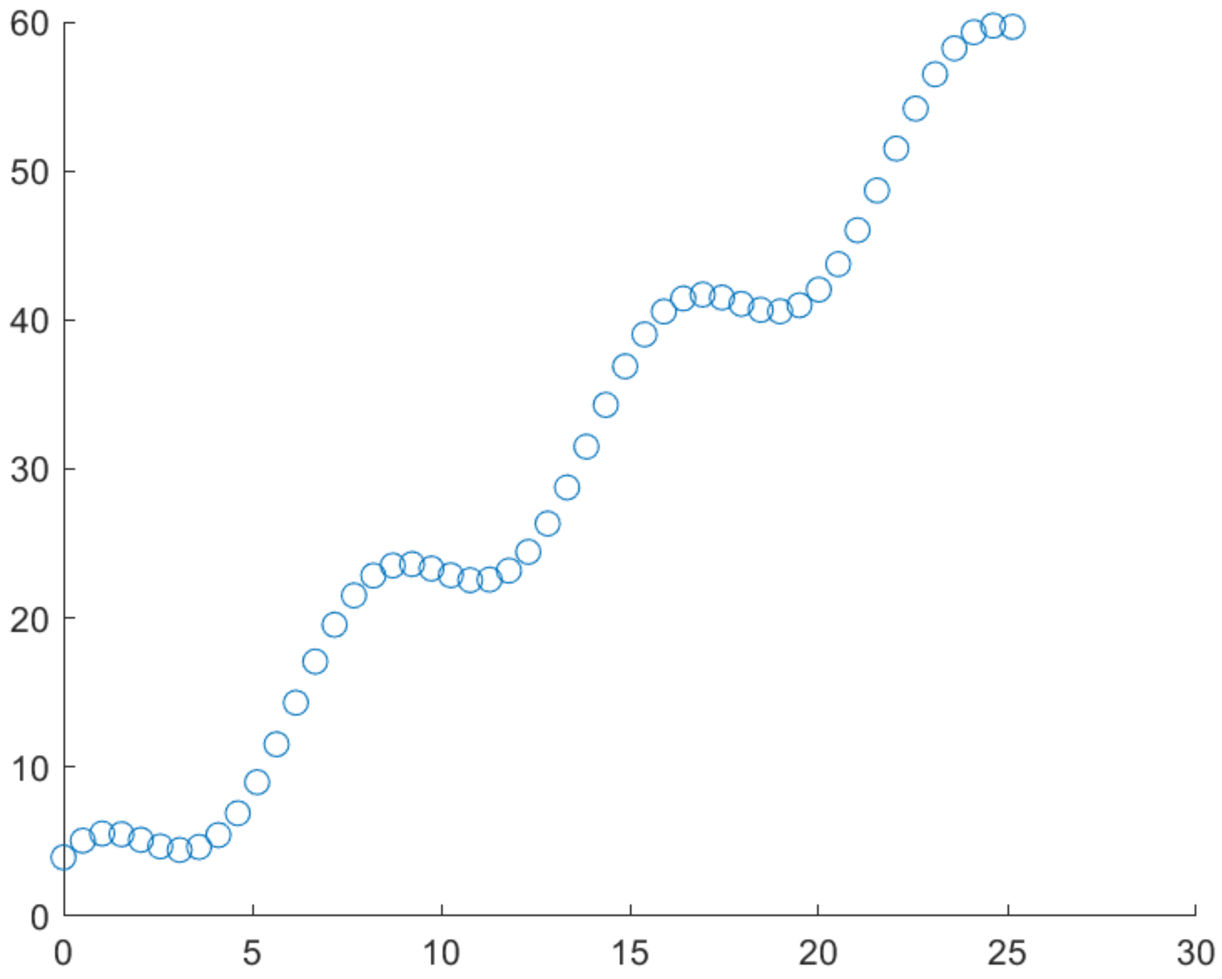
$$p(x) = 3x^2 + 2x + 1$$

$$p(5) = 86$$

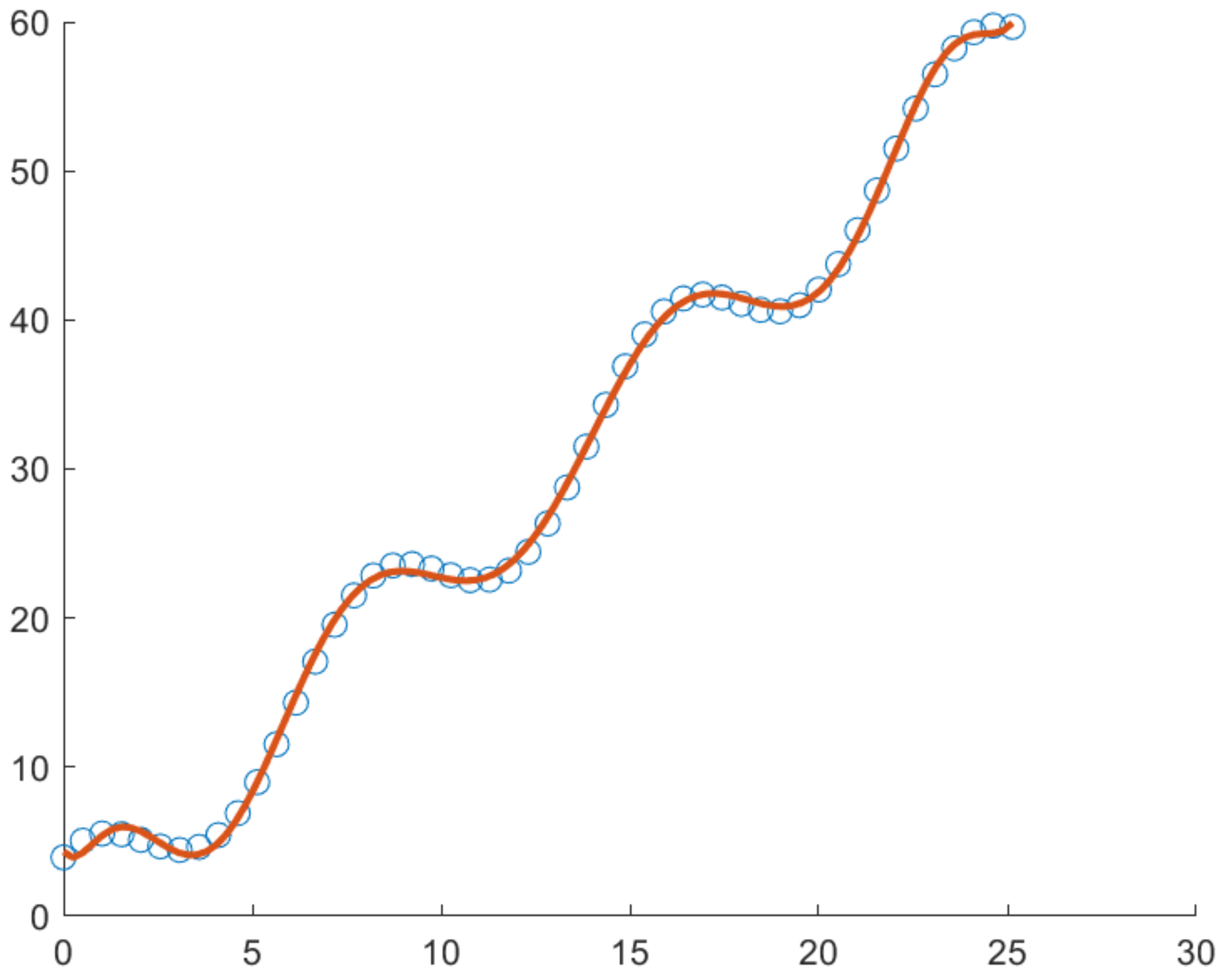
$$p(7) = 162$$

$$p(9) = 262$$

Example - data



Example - fit

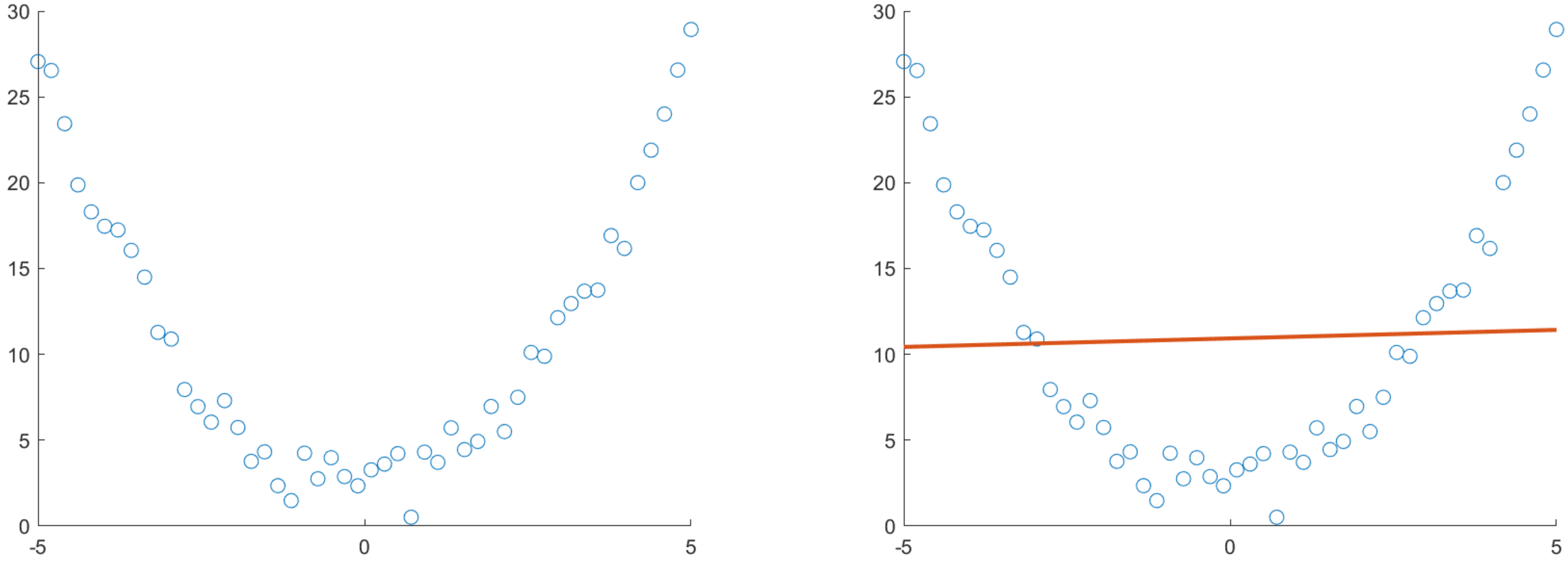


Overfitting and underfitting

- **Overfitting** is "*the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably*". An overfitted model is a statistical model that contains more parameters than can be justified by the data. The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e. the noise) as if that variation represented underlying model structure.
- **Underfitting** occurs when a statistical model cannot adequately capture the underlying structure of the data. An underfitted model is a model where some parameters or terms that would appear in a correctly specified model are missing. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model will tend to have poor predictive performance.

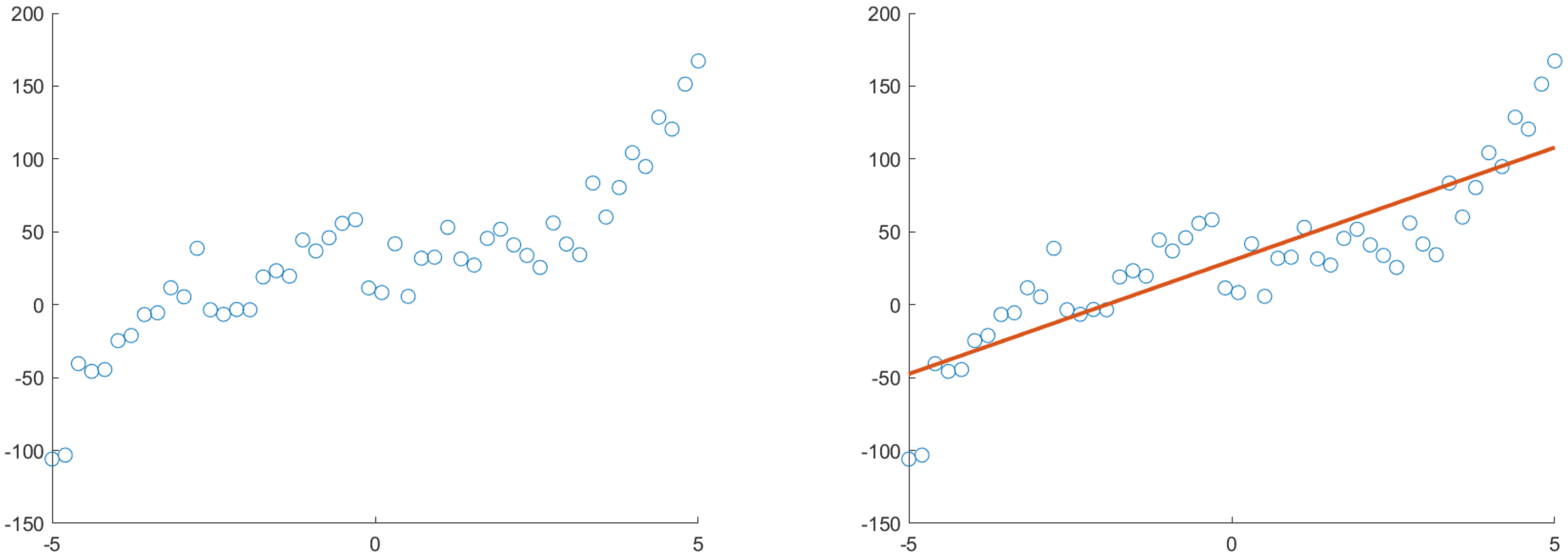
Underfitting

```
p = polyfit(x,y,1);
```



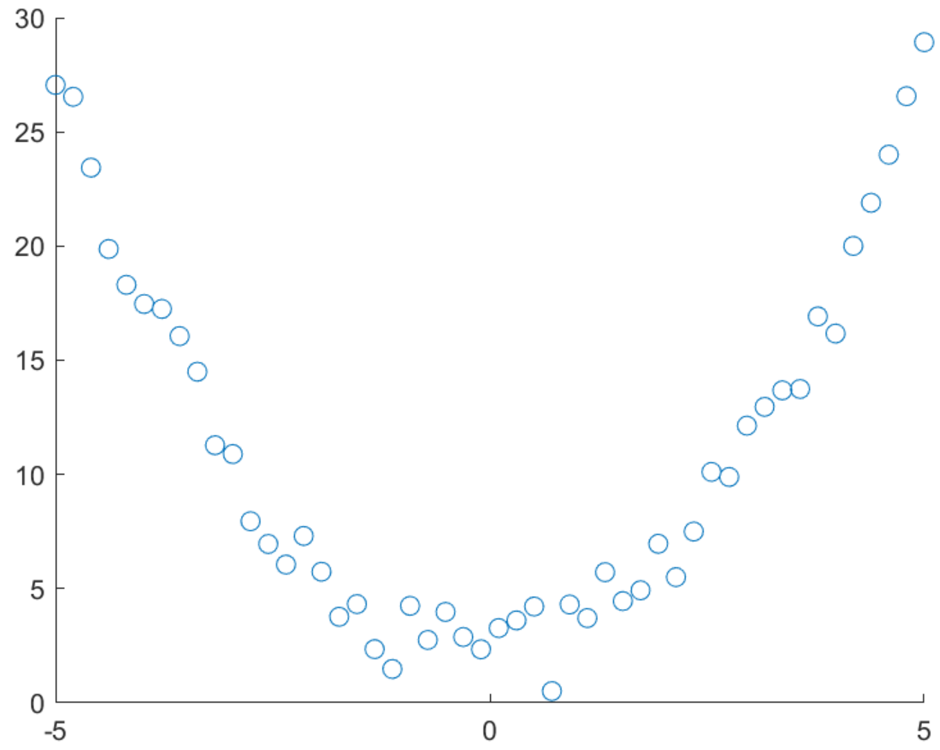
Underfitting

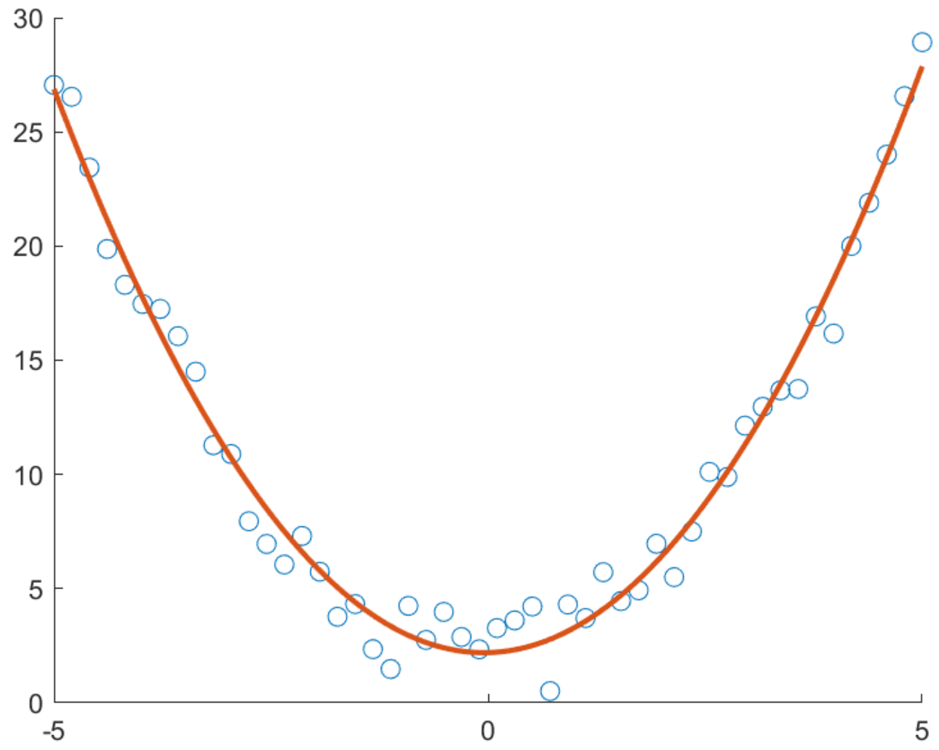
```
p = polyfit(x,y,1);
```



Fitting

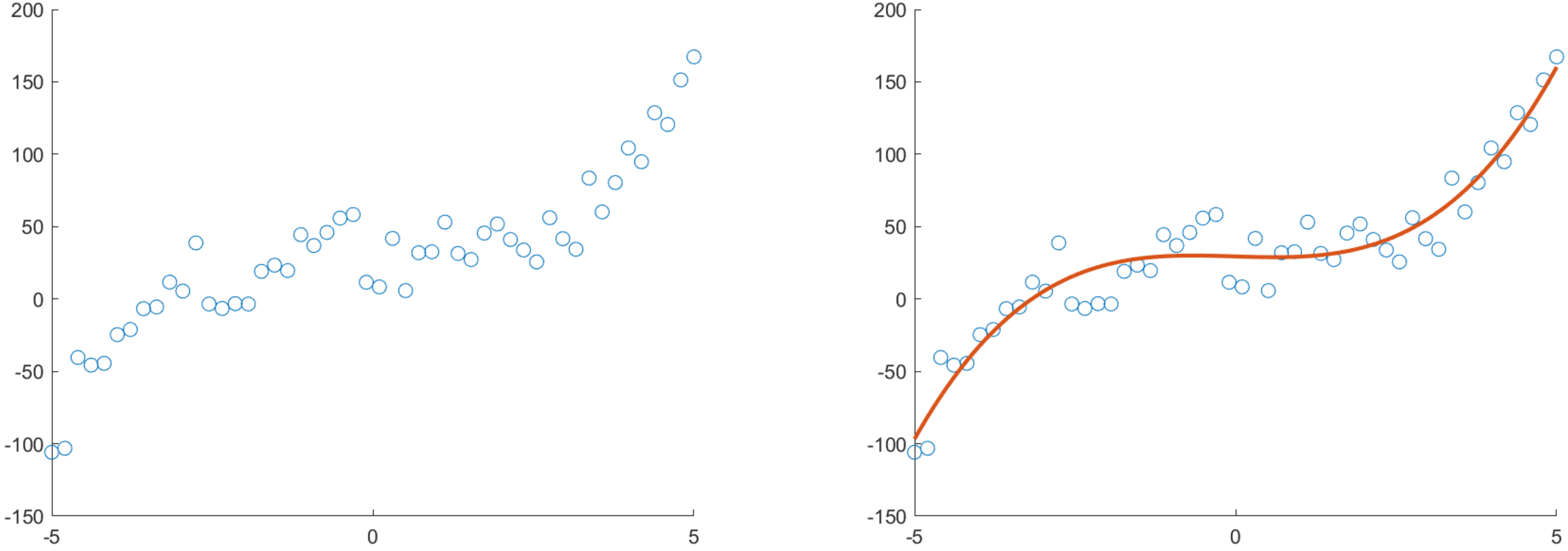
```
p = polyfit(x,y,2);
```





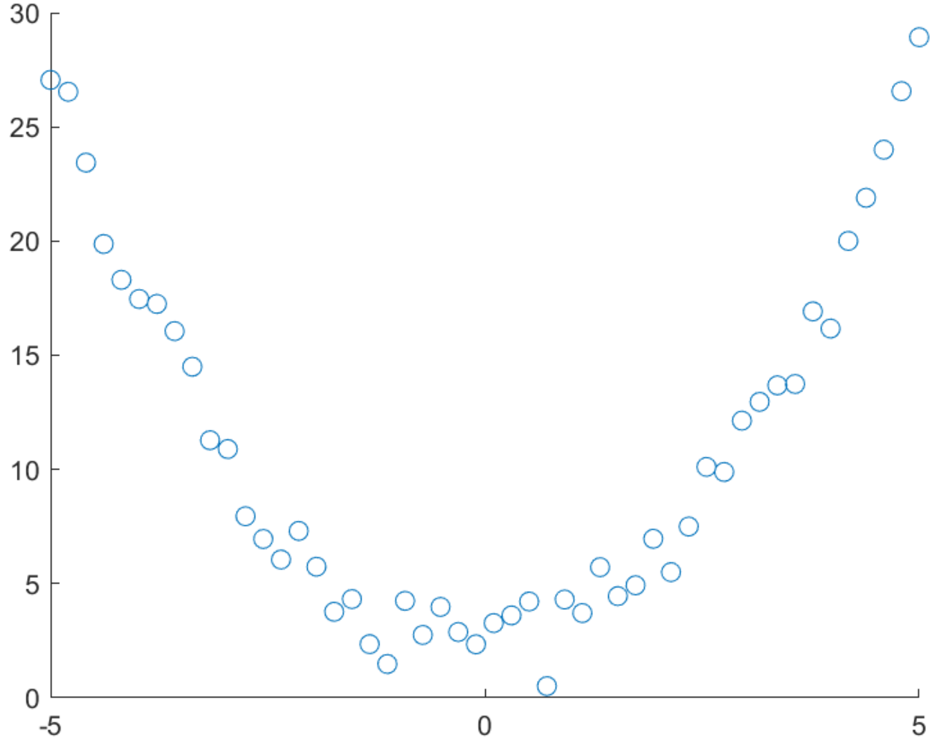
Fitting

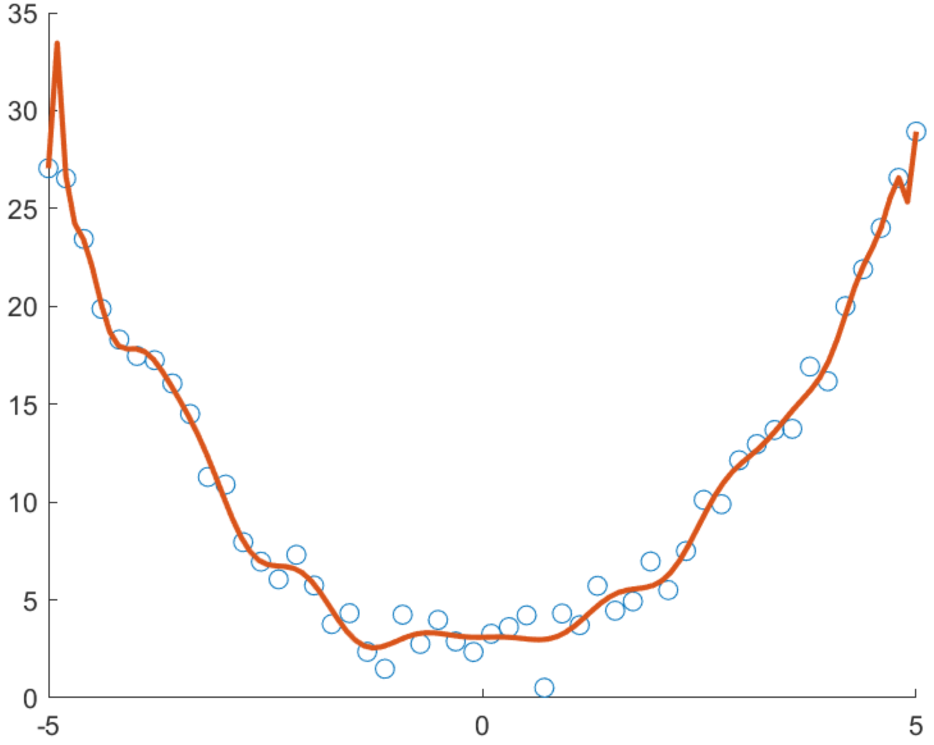
```
p = polyfit(x,y,3);
```



Overfitting

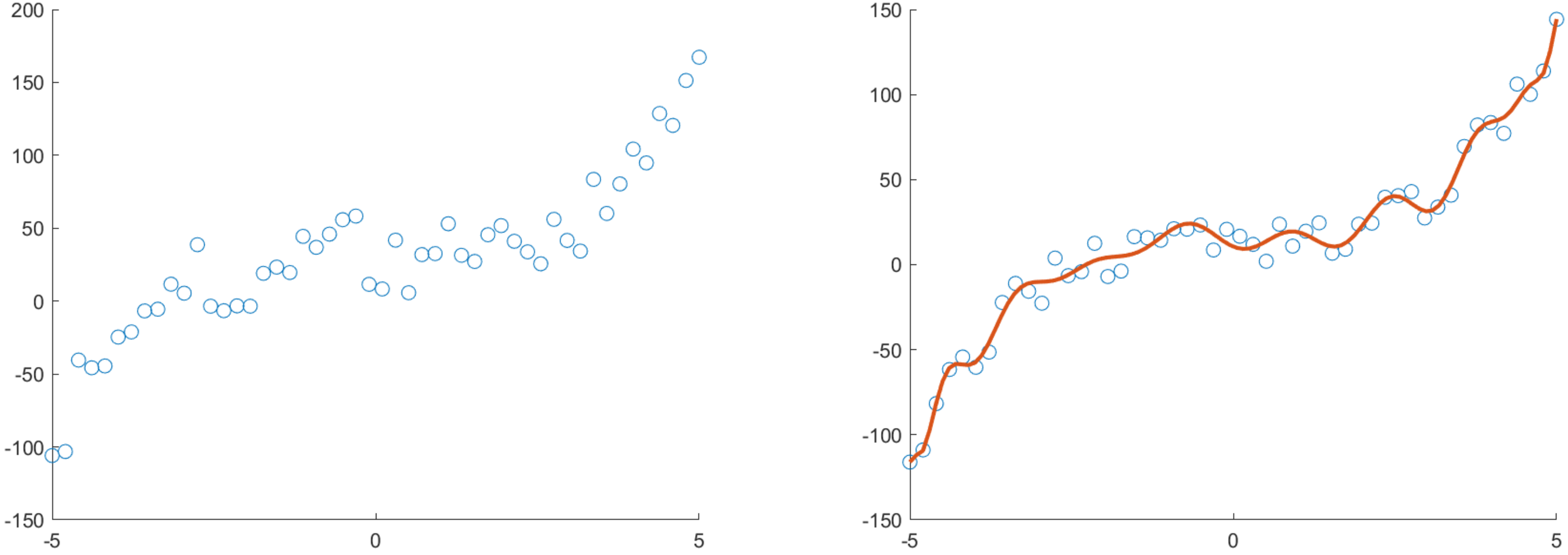
```
p = polyfit(x,y,25);
```





Overfitting

```
p = polyfit(x,y,25);
```



References

- MATLAB Documentation
  - Polynomial curve fitting.
    URL: https://it.mathworks.com/help/matlab/ref/polyfit.html
  - *Polynomial evaluation.*
    URL: https://it.mathworks.com/help/matlab/ref/polyval.html
  - *Linear Regression*.
    URL: https://it.mathworks.com/help/matlab/data_analysis/linear-regression.html