Classical statistical inference

Regression and Model fitting

Associated notebook:

[05-MLE_and_regression/Regression_short.ipynb]

Regression and model fitting

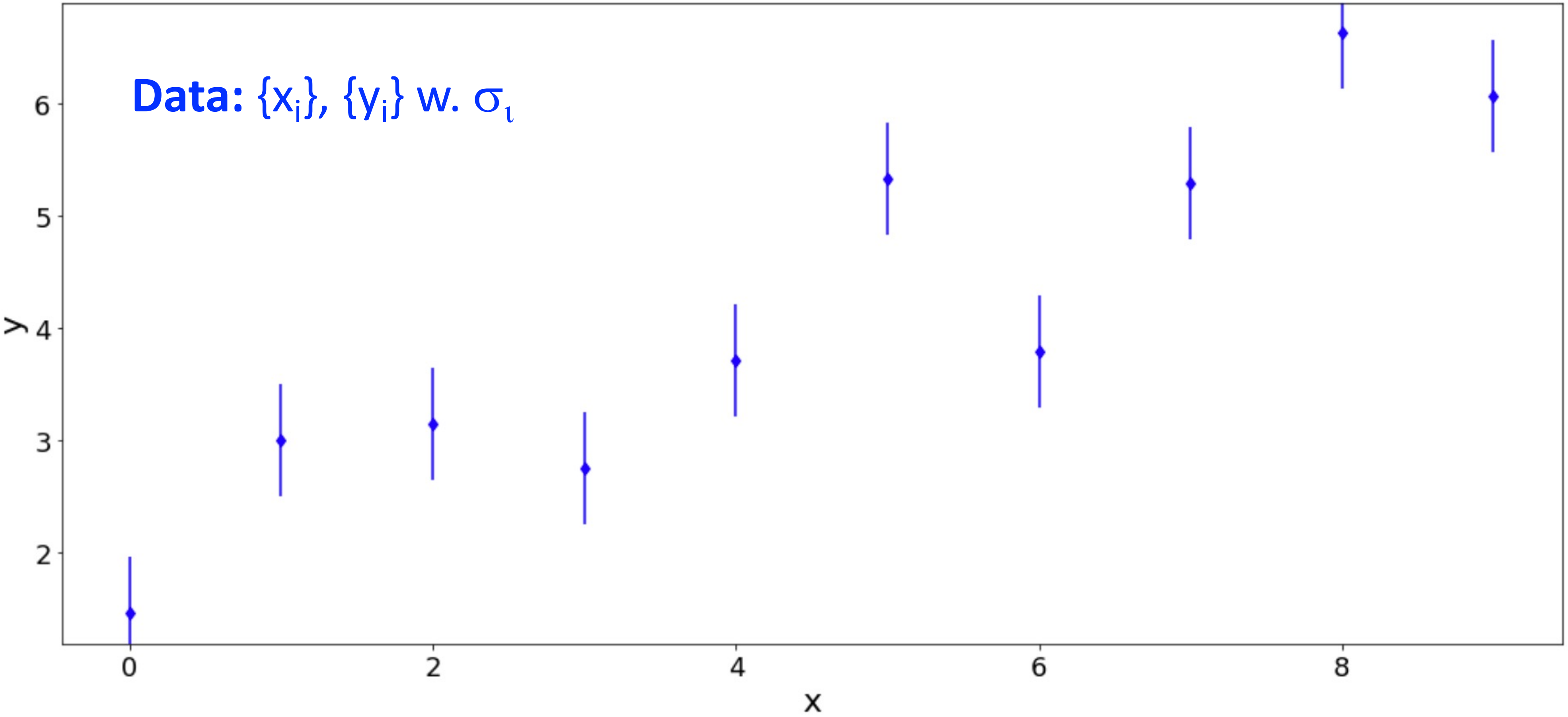
Problem: the quantities of interest are parameters of a model, not the RV that you measure

Examples

Observation	Quantity of interest	Model
Position of a star: **x**(t)	Proper motion (velocity) of the star	V = f(**x**, t, …)
Photometry of an asteroid: mag(t)	P (period of rotation)	mag = f(t, P, …)
Transit of a planet: mag(t)	P (period), e (eccentricity), D (dist to star)	Δ m = f (t, P, e, D, …)
Spectrum of a QSO: F (λ)	M_{BH} (Black hole mass of QSO)	FWHM = f(M_{BH}, L, …)

Regression and model fitting

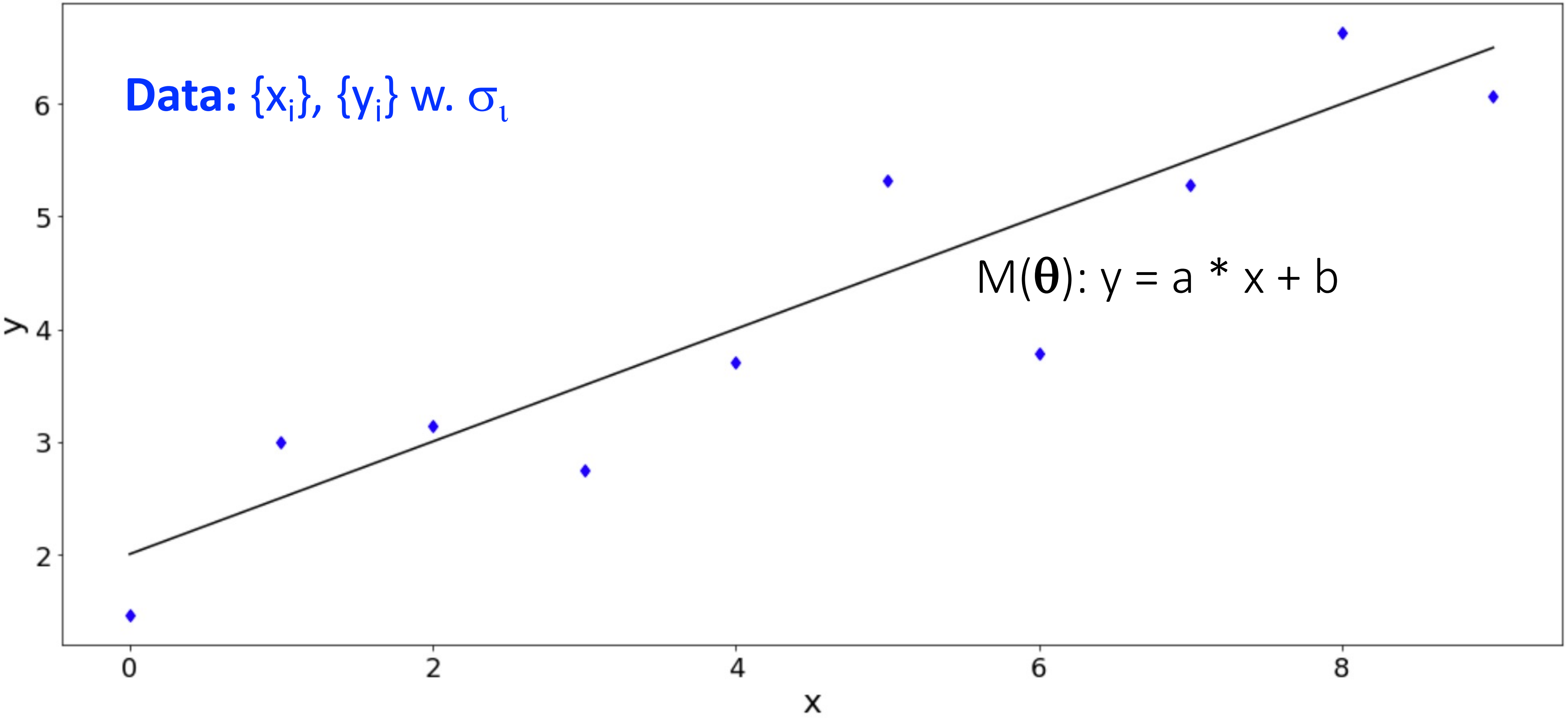
Problem: You measure D ≡ ({y_i}, {x_i})



Model: $M(\theta)$

Regression and model fitting

Problem: You measure D ≡ ($\{y_i\}$, $\{x_i\}$)



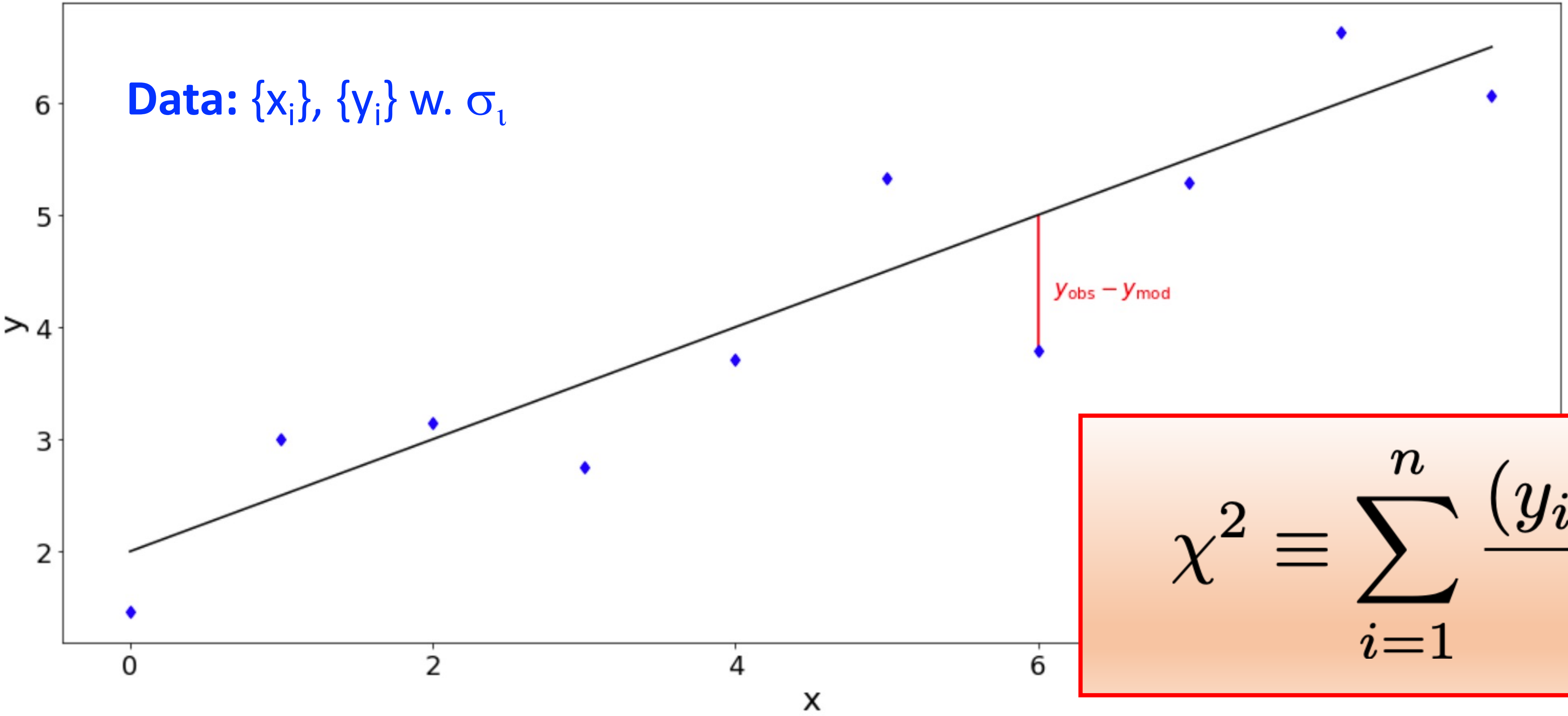
Model: $M(\boldsymbol{\theta})$

y = a * x + b

$\boldsymbol{\theta}$ = a, b

Regression and model fitting

How to find a good model?



Model: M($\boldsymbol{\theta}$)

y = a * x + b
= f(x | $\boldsymbol{\theta}$)

$\boldsymbol{\theta}$ = a, b

Minimize

$$\chi^2 \equiv \sum_{i=1}^{n} \frac{(y_i - y_{i,mod})^2}{\sigma_i^2}$$

Regression and model fitting

$$\chi^2 \equiv \sum_{i=1}^{n} \frac{(y_i - y_{i,mod})^2}{\sigma_i^2}$$

If $\sigma_i = 1$: Least square regression

If $\sigma_i \neq 1$: chi-square regression

The χ^2 is called a **merit** function

When uncertainties between variables are correlated, the χ^2 is expressed:

$$\chi^2 = \sum_{i=1}^{n} \sum_{l=1}^{n} (y_i - y_{i,mod}) F_{i,l} (y_l - y_{l,mod})$$

Where *F* is the Fisher matrix

$$[F]^{-1} = [C] = \begin{bmatrix} \sigma_k^2 & \sigma_{kl} \\ \sigma_{kl} & \sigma_l^2 \end{bmatrix}$$

Regression and model fitting

$$\boxed{\chi^2 \equiv \sum_{i=1}^{n} \frac{(y_i - y_{i,mod})^2}{\sigma_i^2}}$$

If $\sigma_i = 1$: Least square regression

If $\sigma_i \neq 1$: chi-square regression

When uncertainties between variables are correlated, the χ^2 is expressed:

$$\chi^2 = (\vec{d} - \vec{m})^T C^{-1} (\vec{d} - \vec{m})$$

$$[F]^{-1} = [C] = \begin{bmatrix} \sigma_k^2 & \sigma_{kl} \\ \sigma_{kl} & \sigma_l^2 \end{bmatrix}$$

$$\vec{d} \equiv \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} \qquad \vec{m} \equiv \begin{pmatrix} y_{1,\mathrm{mod}} \\ y_{2,\mathrm{mod}} \\ \dots \\ y_{N,\mathrm{mod}} \end{pmatrix}$$

Link between χ^2 and likelihood

$$\boxed{L \; = \; p(D \,|\, M(\boldsymbol{\theta}))}$$

Case of a straight line: $y_i = \theta_0 + \theta_1\, x_i + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma_i)$

For each y_k we have: $$p(y_k \,|\, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\,\pi}} \exp\left[-0.5\left(\frac{y_k - \mu}{\sigma}\right)^2\right]$$

With e.g. $\mu \equiv y_{k,\mathrm{mod}}$ If the model is “correct”

Hence, we have for our data set D:

$$L \;\equiv\; p(\{y_i\} \,|\, \{x_i\}, \boldsymbol{\theta}) = \prod_{i=1}^{N} \frac{1}{\sigma_i\sqrt{2\,\pi}} \exp\left[\left(\frac{-(y_i - (\theta_0 + \theta_1\, x_i))^2}{2\,\sigma_i^2}\right)\right]$$

Link between χ^2 and likelihood

Hence, we have for our data set D:

$$L \equiv p(\{y_i\} \,|\, \{x_i\}, \boldsymbol{\theta}) = \prod_{i=1}^{N} \frac{1}{\sigma_i\sqrt{2\,\pi}} \exp\left[\left(\frac{-(y_i - (\theta_0 + \theta_1\, x_i))^2}{2\,\sigma_i^2}\right)\right]$$

$$\boxed{\ln(L) \propto \sum_{i=1}^{N}\left(\frac{-(y_i - (\theta_0 + \theta_1\, x_i))^2}{2\,\sigma_i^2}\right)}$$

$$\boxed{\chi^2 \equiv \sum_{i=1}^{n} \frac{(y_i - y_{i,mod})^2}{\sigma_i^2}}$$

$$\Rightarrow \quad \boxed{L \propto \exp(-\chi^2/2)}$$

Minimizing χ^2 is equivalent to maximizing L

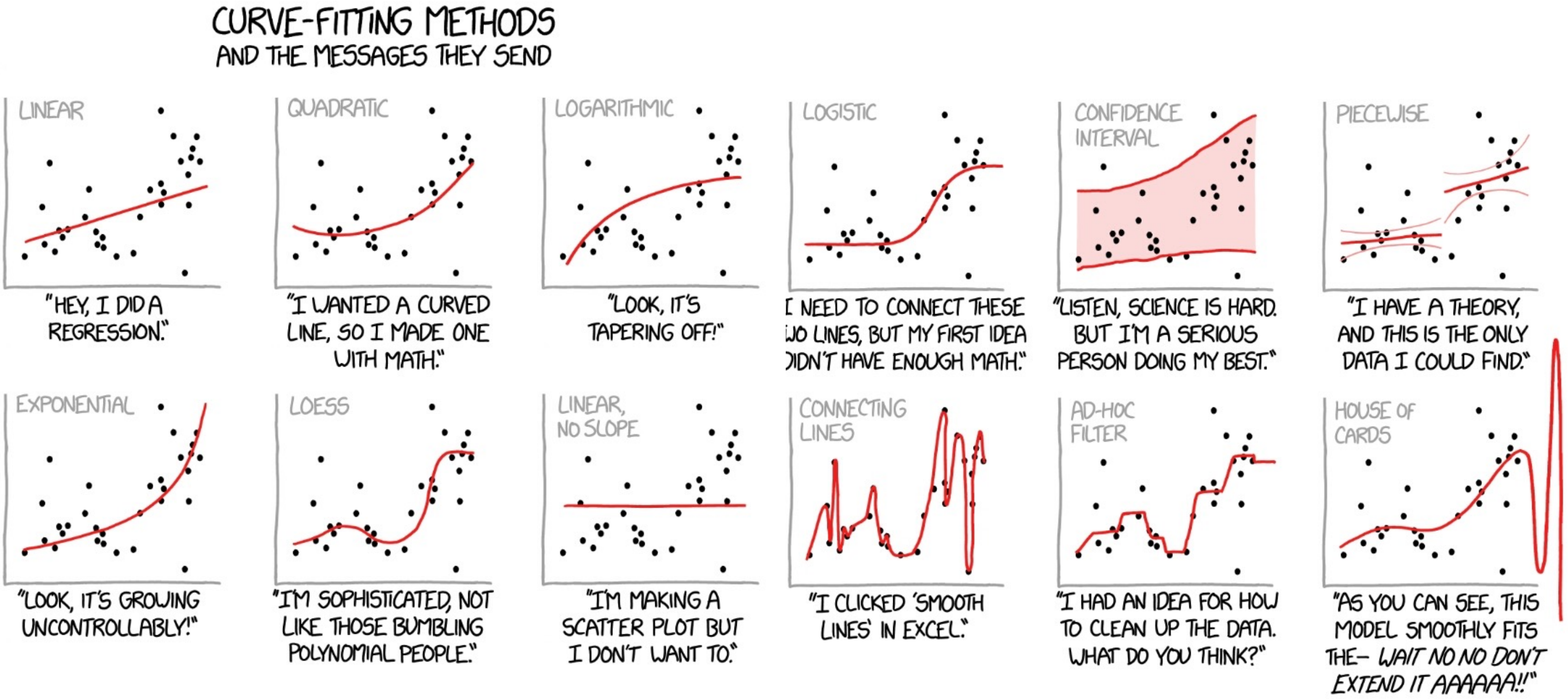
Regression of a straight line in python

Linear model fitting: See Sect. IV.1

Python implementation: `numpy.polyfit(x, y, deg=1, w=1/sigma)`

Go to Sect. IV.1.1 of the Notebook for practical example

Regression and model fitting



https://xkcd.com/2048/

How to choose a suitable regression method ?

The way you will tackle a regression problem may depend on:

- **Linearity:** is the model linear *in its parameters*? $f(x \,|\, \boldsymbol{\theta}) = \sum_{p=1}^{k} \theta_p g_p(x)$

- **Complexity:** large number of parameters increase complexity and covariance matrix on uncertainties

- **Error behaviour:** uncertainties on dependent and independent variable and their correlation.

How to choose a suitable regression method ?

Frequentist: (this lecture)	***Bayes (future lecture):***
Optimization with some merit function	*Sampling* of the likelihood
Search for *best* (fit) model *parameters*	PDF on parameters
Often when *simple* error behaviour	More *complex* error behaviour

Linear vs non linear regression

A model is **linear** if: $$f(x\,|\,\boldsymbol{\theta}) = \sum_{p=1}^{k} \theta_p g_p(x)$$

$g_p(x)$ can be a non linear function of *x* BUT does not depend on any free parameter

In this case, the values of the parameters that yield $\frac{\partial \ln(L)}{\partial \theta_i} = 0$ (max. likelihood) can be found "analytically"

When the model is **not linear** , the minimization of the χ^2 has to be performed *numerically*

Linear vs non linear regression

Linear model fitting: See Sect. IV.1

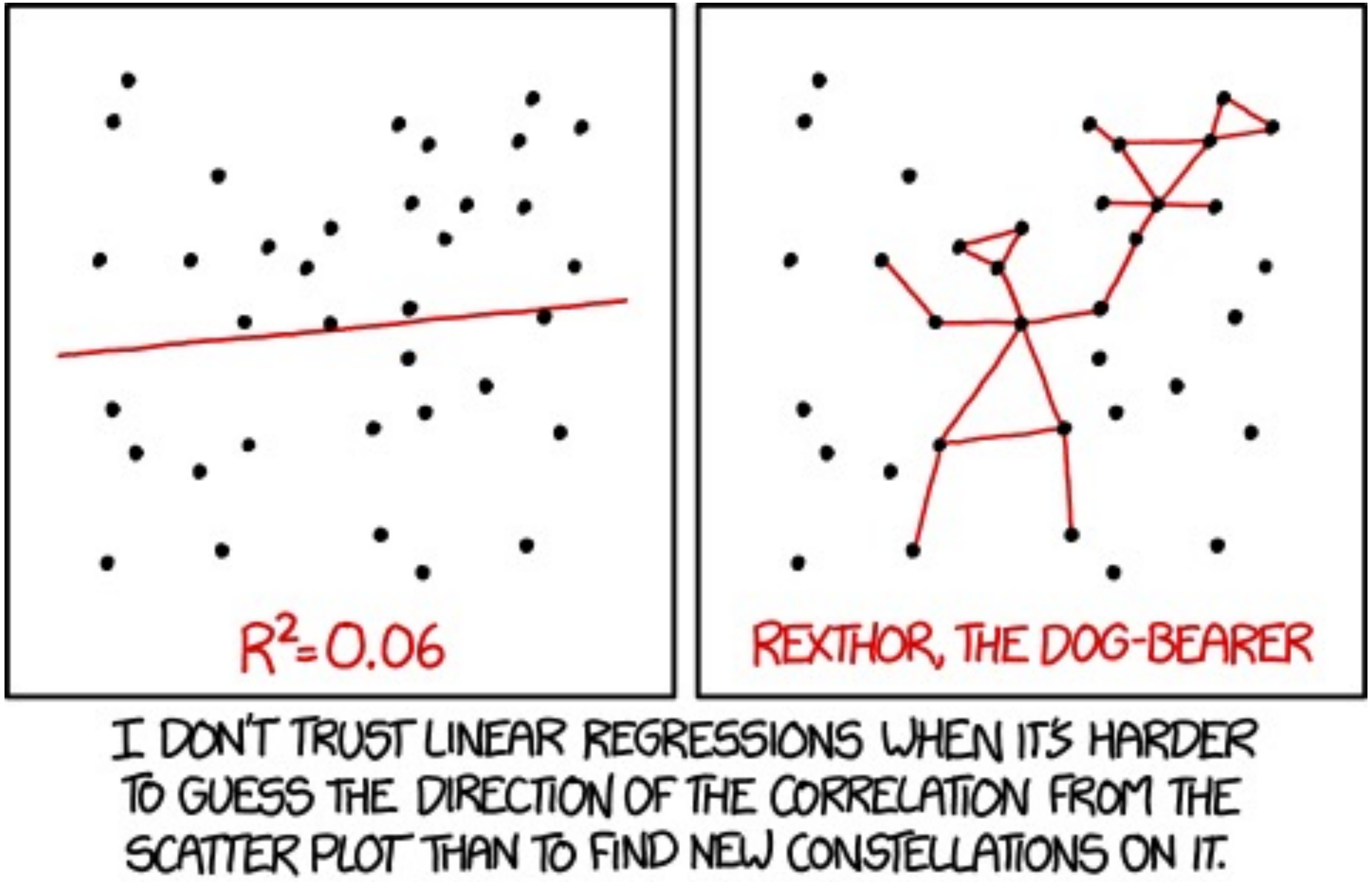
Python implementation: `numpy.polyfit(x, y, deg=1, w=1/sigma)`

NON Linear model fitting: See Sect. IV.3

Python implementation: `scipy.optimize.curvefit()`

Go to Sect. IV.3 of the Notebook for practical example

Quality of the regression



R^2 = Coefficient of determination of Pearson

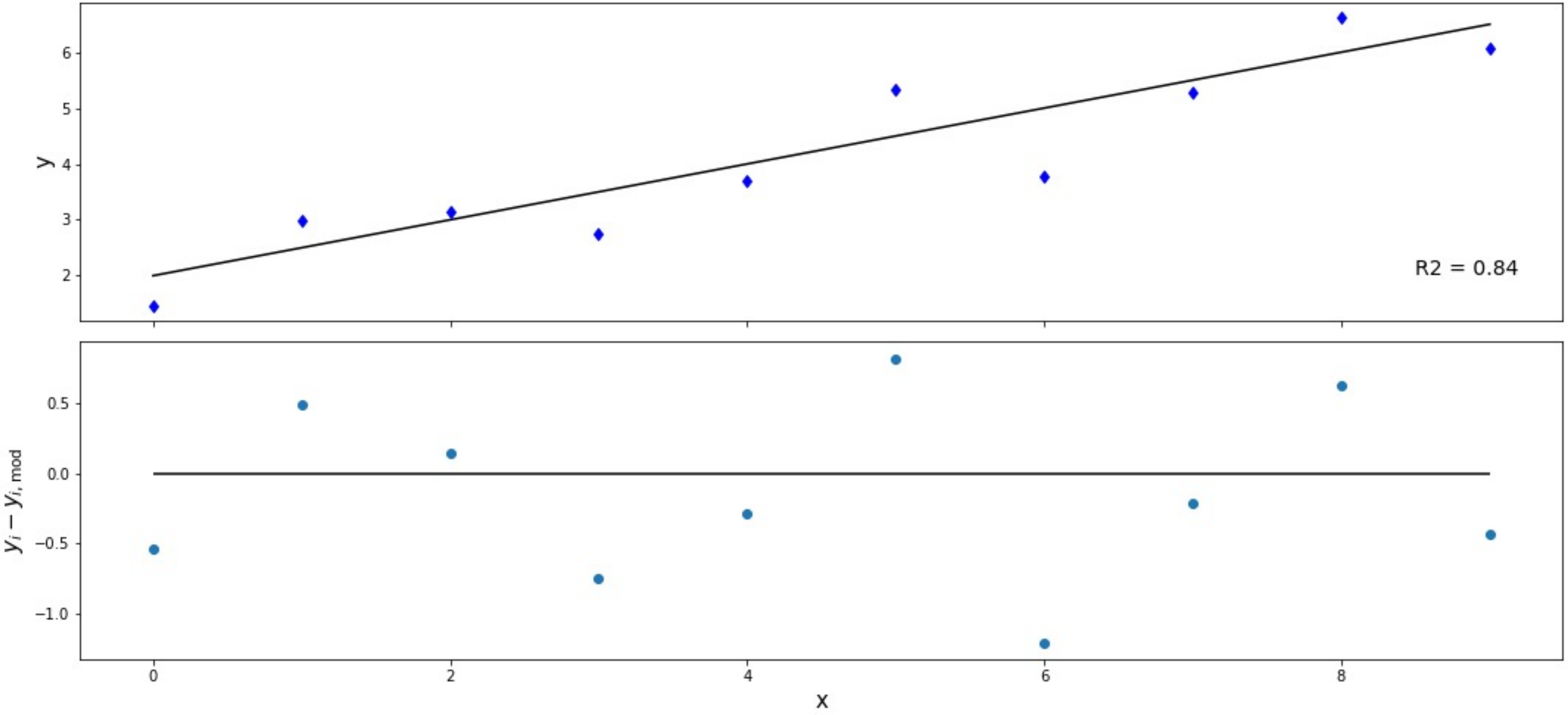
xkcd.com/1725/

Quality of the regression

Coefficient of determination of Pearson

Look at the residuals: $y_i - y_{i,\mathrm{mod}}$

$$R^2 = 1 - \frac{\sum_i (y_i - y_{i,\mathrm{mod}})^2}{\sum_i^n (y_i - \bar{y})^2}$$



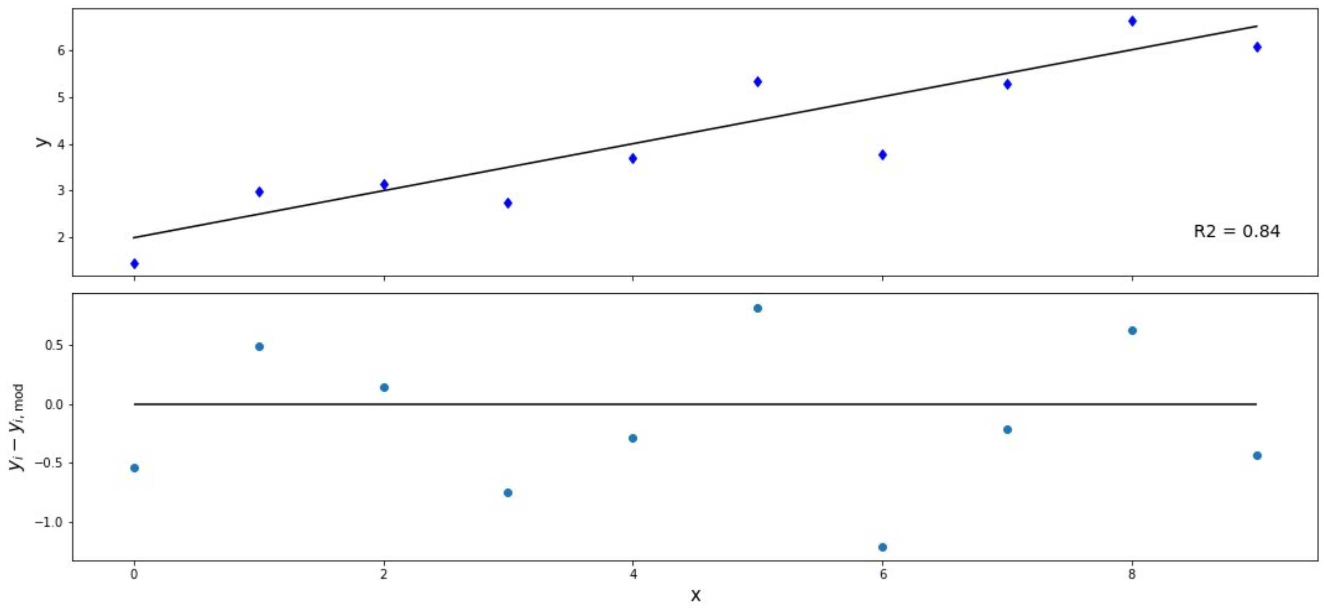
Closer R^2 is to 1, more the model is good at explaining the data (or at making predictions).

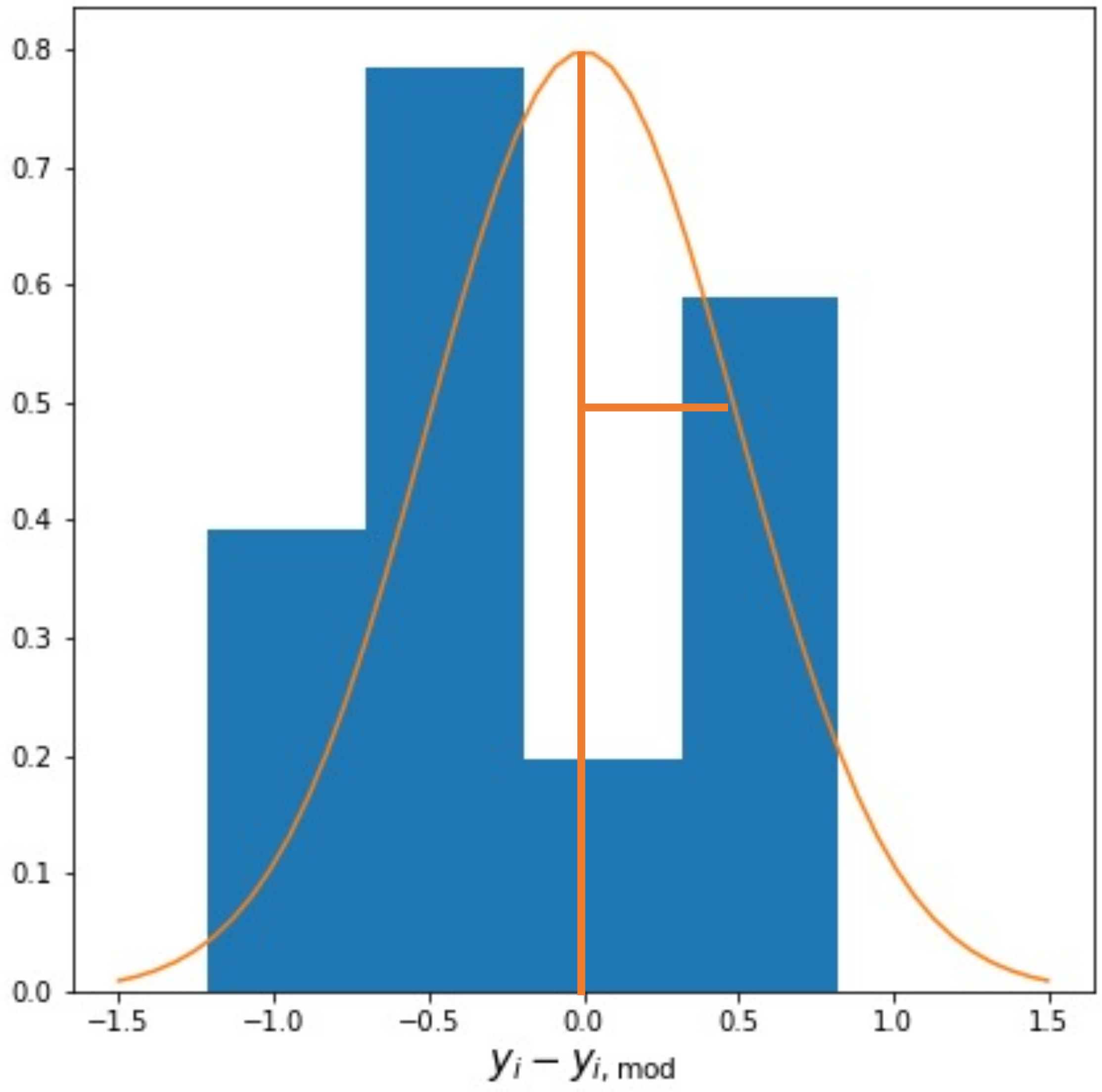
But use with care as it is originally designed for linear models and does not account for uncertainty on data points (cf least-square vs χ^2 regression)

Quality of the regression

Compare variance of residuals to σ^2 on y_i

Look at the residuals: $y_i - y_{i,\mathrm{mod}}$





Quality of the regression

Your χ^2 is a random variable !

$$Q = \sum_{i=1}^{k} z_i^2 \rightarrow \quad p(Q|k) = \frac{1}{(2\,\Gamma(k/2))}(Q/2)^{k/2-1}\exp(-Q/2)$$

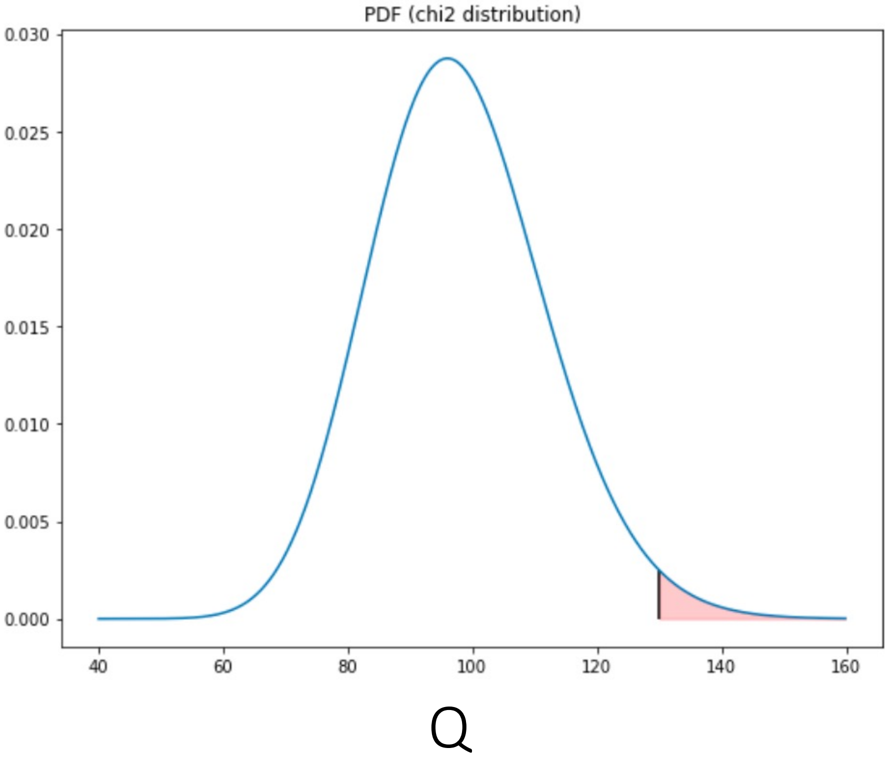
k = **d**egree **o**f **f**reedom = N *points* – n *parameters*

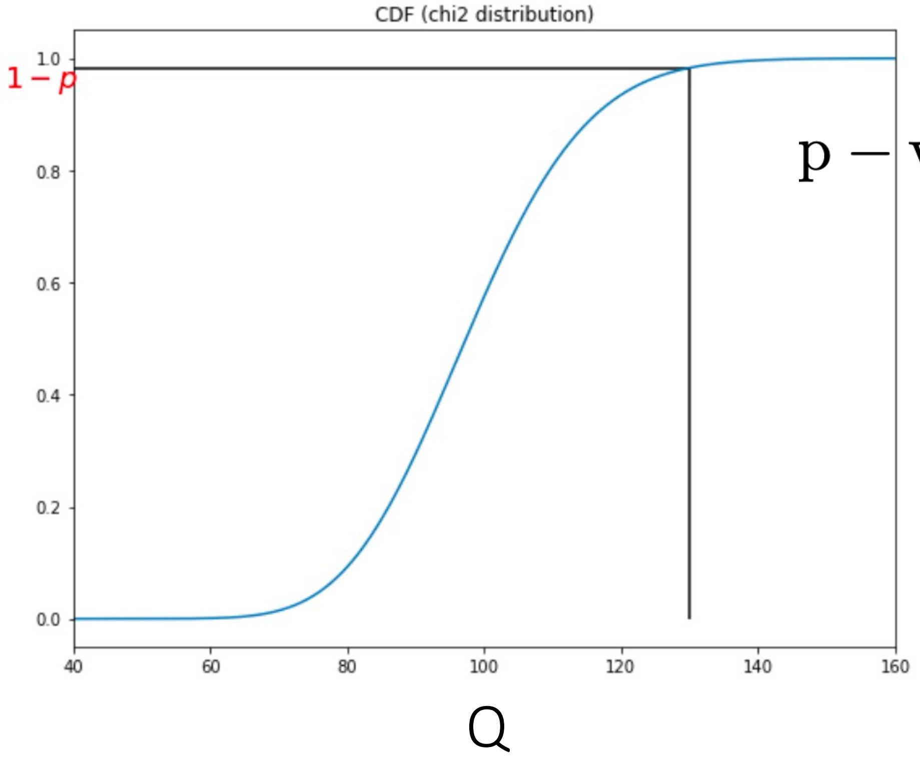
If you fit a model with 2 parameters on a set of 100 points => 98 d.o.f.
Expectation E(χ^2) $= 100 - 2 = 98$
Reduced χ^2 : $\chi^2_{red} = \chi^2_{dof} /$ d.o.f. $\quad\Rightarrow\quad$ Reduced $\chi^2 \equiv 1.$ if good fit

See also the Notebook 03-Basic_statistics_and_proba_concepts/Descriptive_statistics_02.ipynb

Quality of the regression

$$p(Q|k) = \frac{1}{(2\,\Gamma(k/2))}(Q/2)^{k/2-1}\exp(-Q/2)$$





$$\mathrm{p-value} = \mathrm{p}(\mathrm{Q} \geq \chi^2_{\mathrm{obs}})$$

$$= 1 - p(Q \leq \chi^2_{\mathrm{obs}})$$

CDF

Typically:
p-value < 0.05 : ☹️
0.05 < p-value < 1: 😀
p-value close to 1 : 😕

```
1-scipy.stats.chi2.cdf(chi2_data, df= len(data)-nparam)
```

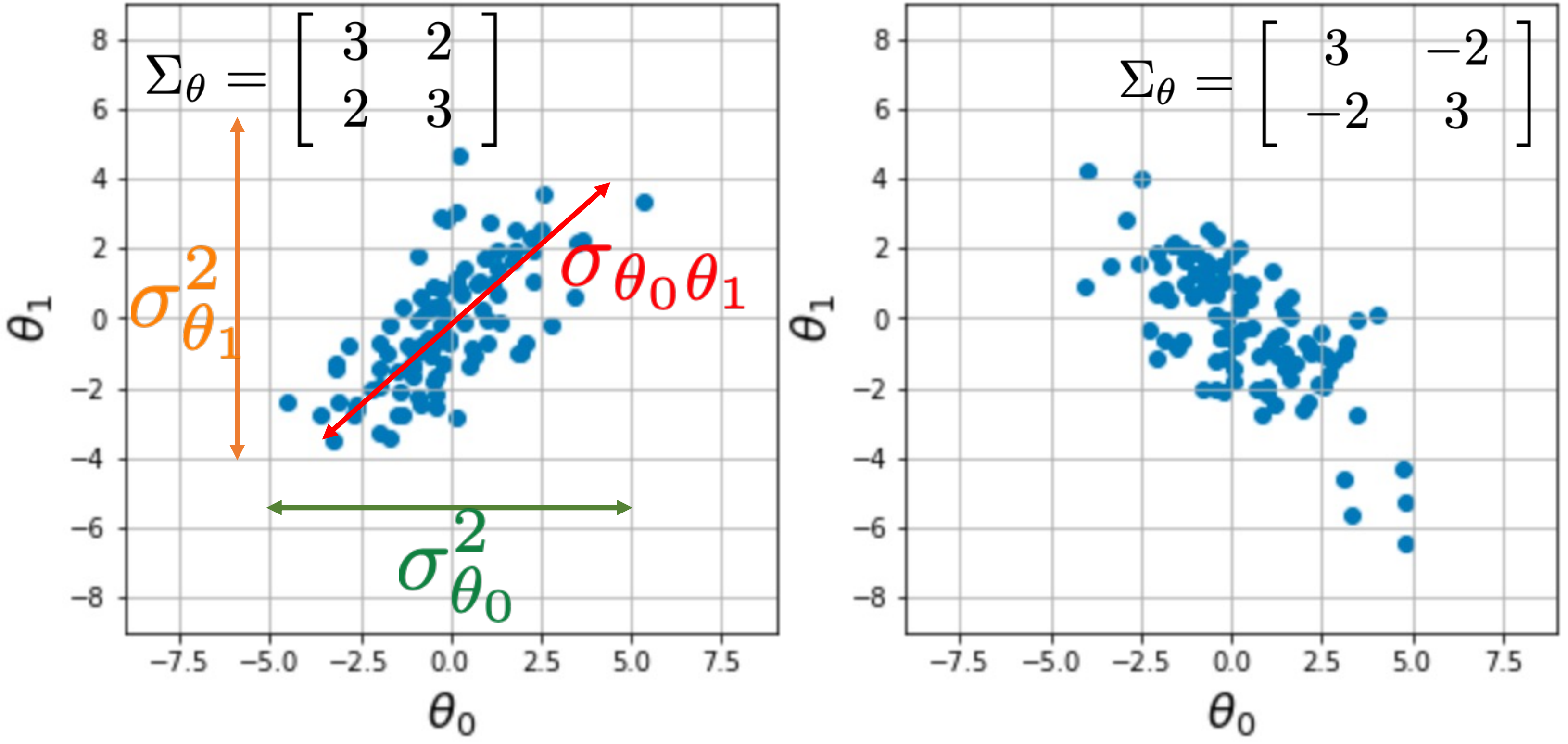
Go to Sect. IV.1.1 of the Notebook for practical example

Uncertainty on the fitted parameters

The python functions return a covariance matrix (Warning : use arg. cov=True)

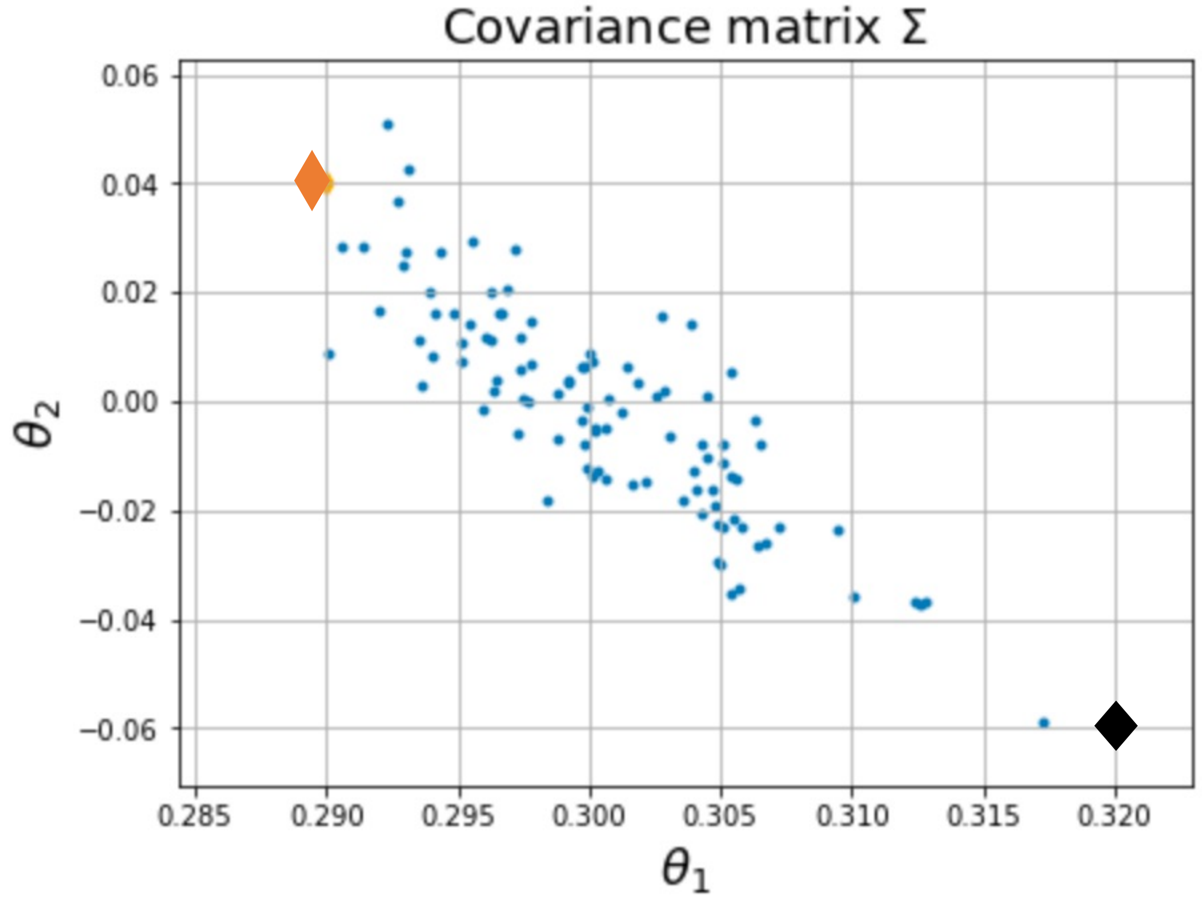
The diagonal elements of the matrix give the variance on the parameters (uncertainty²)

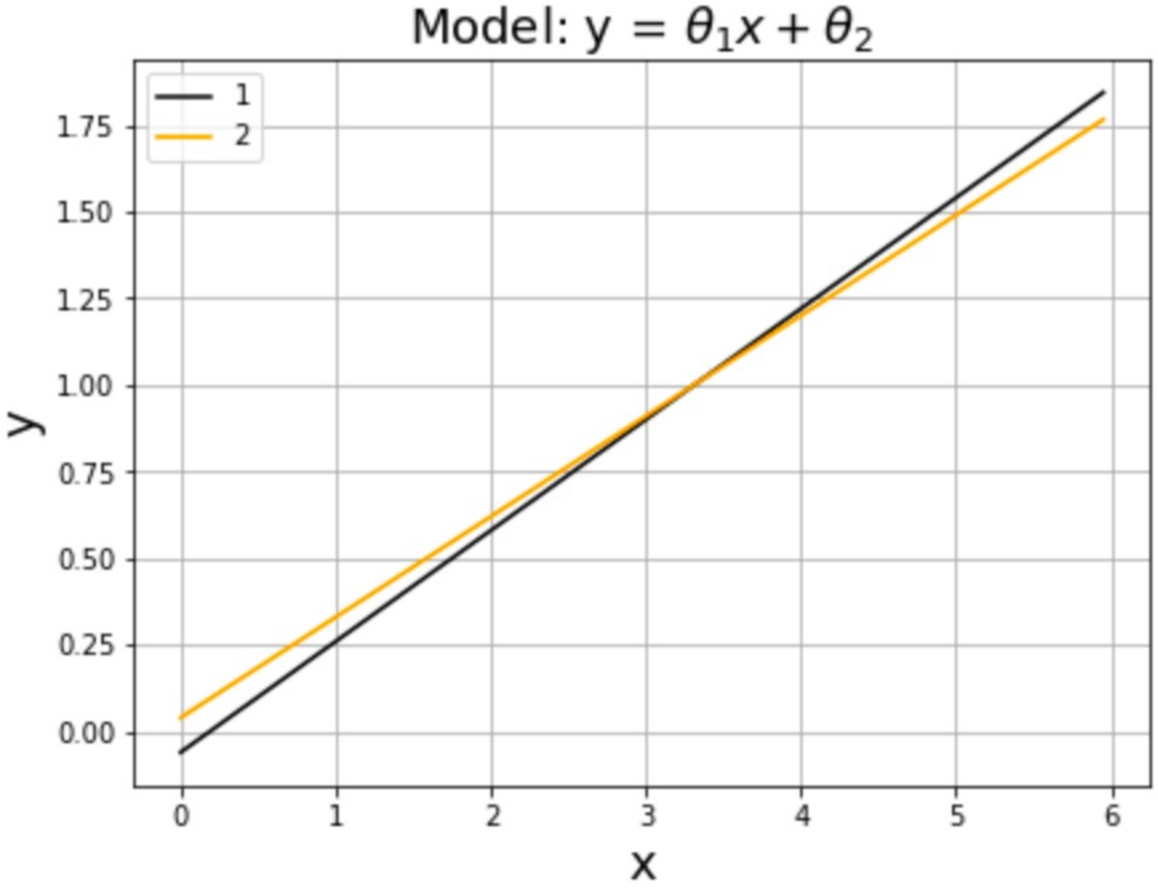
$$\Sigma_\theta = \begin{bmatrix} \sigma^2_{\theta_0} & \sigma_{\theta_0\theta_1} \\ \sigma_{\theta_0\theta_1} & \sigma^2_{\theta_1} \end{bmatrix}$$



Uncertainty on the fitted parameters

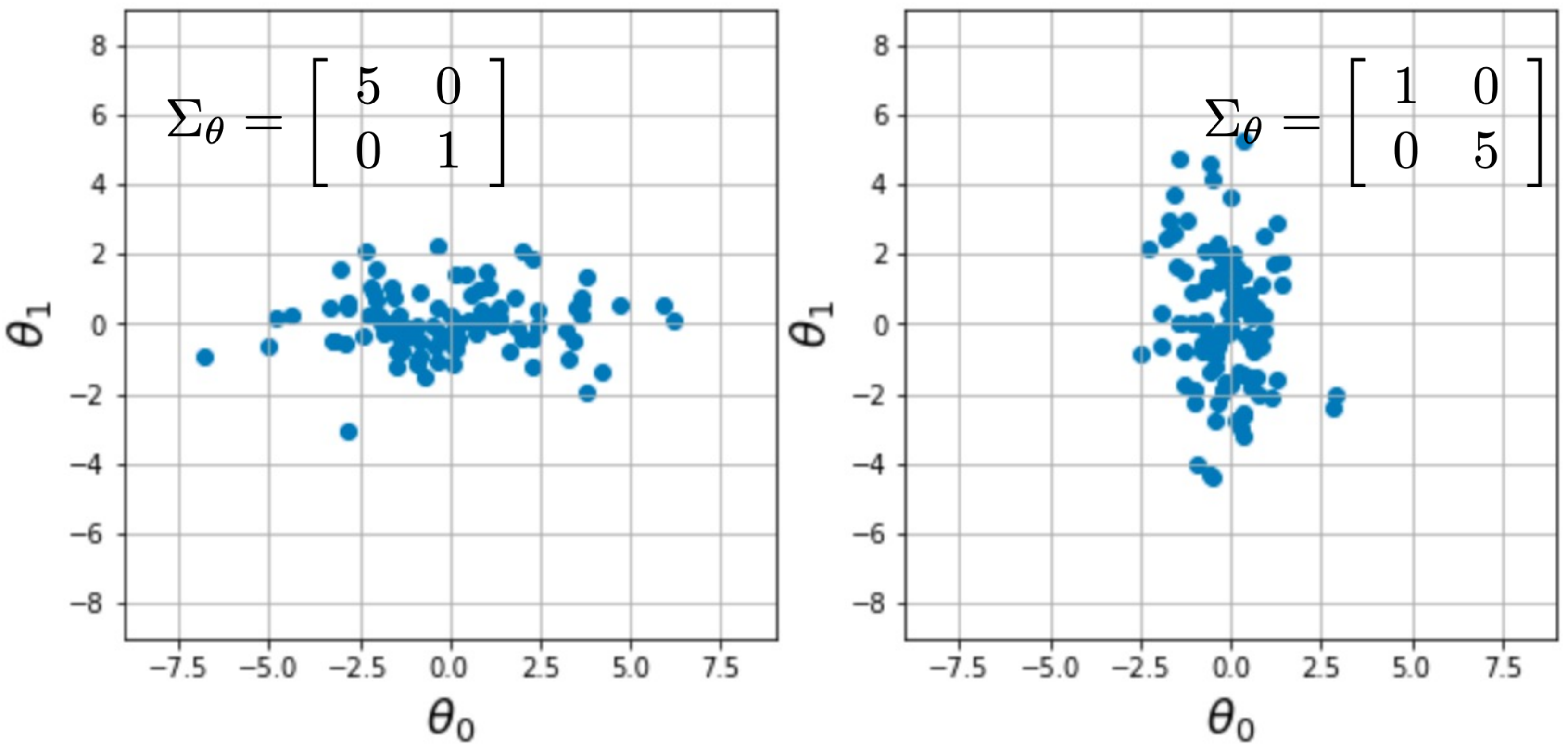
$$\Sigma_\theta = \begin{bmatrix} + & - \\ - & + \end{bmatrix}$$





Uncertainty on the fitted parameters

$$\Sigma_\theta = \begin{bmatrix} \sigma^2_{\theta_0} & \sigma_{\theta_0\theta_1} \\ \sigma_{\theta_0\theta_1} & \sigma^2_{\theta_1} \end{bmatrix}$$



Non linear models and initial conditions

Non linear regression use numerical methods to find the minimum χ^2.
For this reason the results are highly sensitive to initial conditions!

Tips for choosing good initial conditions

- Visualise the prediction of the model for various sets of parameters
- If too many parameters: R andomly sample the parameter space and calculate the χ^2 .
- If the optimization is fast, repeat with different sets of initial conditions