PH3010 / MSci Skills

Statistical Data Analysis

Lecture 1: The Method of Least Squares

Autumn term 2017

Glen D. Cowan
RHUL Physics

The Statistical Data Analysis Module

This module on Statistical Data Analysis contains two parts:

Curve fitting with the method of least squares (weeks 1, 2)

Introduction to Machine Learning (week 3)

You will be given a number of exercises that should be written up in the form of a mini-project report. The standard rules apply:

Use the LaTeX template from the PH3010 moodle page.

Word limit is 3000, not including appendices.

All code should be submitted as an appendix.

The exercises for the least-squares part of the module are at the end of the script on moodle. Core exercises are numbers 1, 2, 3.

There may be some adjustment of the assigned exercises depending on how fast we are able to get through the material. (Exercise 4 may become optional.)

Outline (part 1)

Today:

Basic ideas of fitting a curve to data

The method of least squares

Finding the fitted parameters

Find the statistical errors of the fitted parameters

Using error propagation

Start of exercises

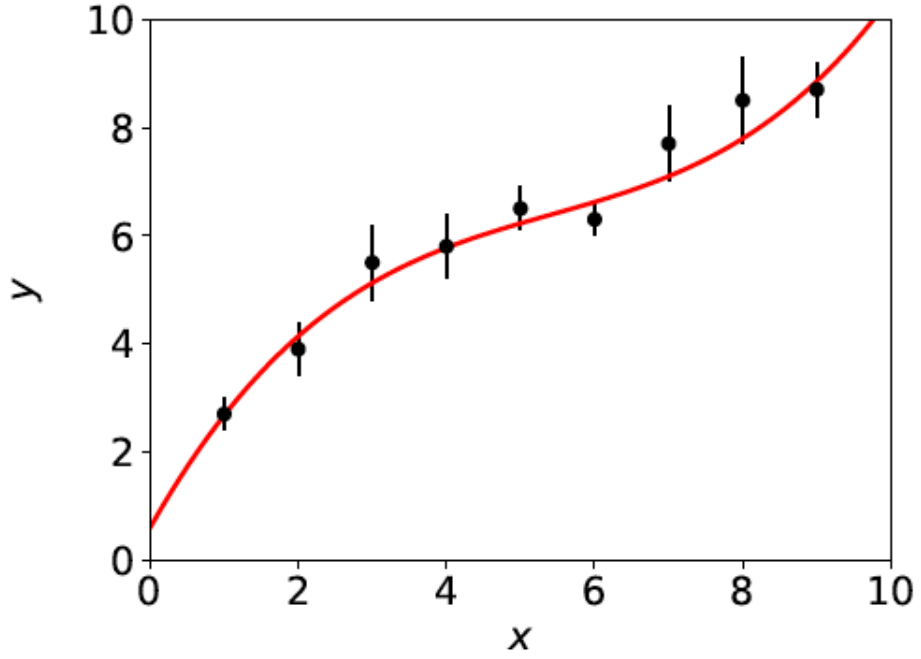
Next week:

goodness-of-fit, fitting correlated data, more exercises

Curve fitting: the basic problem

Suppose we have a set of N measured values y_i, $i = 1,.., N$.

Each y_i has an "error bar" σ_i, and is measured at a value x_i of a control variable x known with negligible uncertainty:

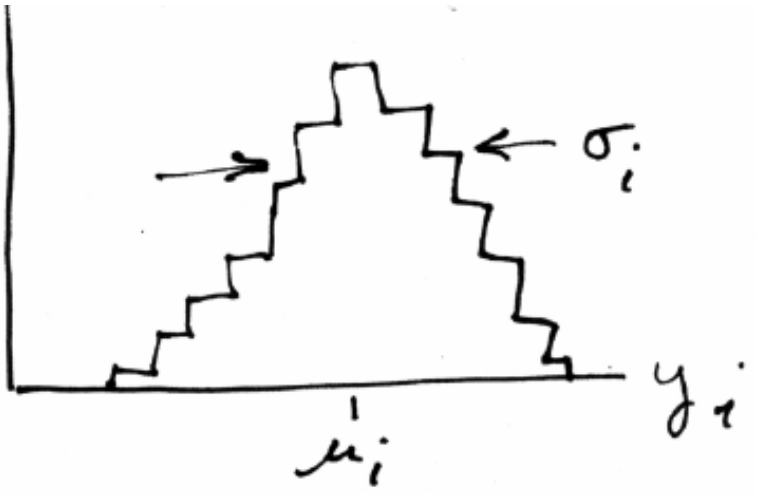


Roughly speaking, the goal is to find a curve that passes "close to" the data points, called "curve fitting".

Measured values → random variables

We will regard the measured y_i as independent observations of random variables (r.v.s).

Idea of an r.v.: imagine making repeated observations of the same y_i, and put these in a histogram:



The distribution of y_i has a mean μ_i and standard deviation σ_i.

We only know the data values y_i from a single measurement, i.e., we do not know the μ_i (goal is to estimate this).

y_i = measured value
x_i = control variable value
μ_i = "true value" (unknown)
σ_i = error bar ← suppose these are known

Fitting a curve

The standard deviation σ_i reflects the reproducibility (statistical error) of the measurement y_i.

If σ_i were to be very small, we can imagine that y_i would we be very close to its mean μ_i, and lie on a smooth curve given by some function of the control variable, i.e., $\mu_i = f(x_i)$.

Goal is to find the function $f(x)$. Here we will assume that we have some hypothesis for its functional form, but that it contains some unknown constants (parameters), e.g., a straight line:

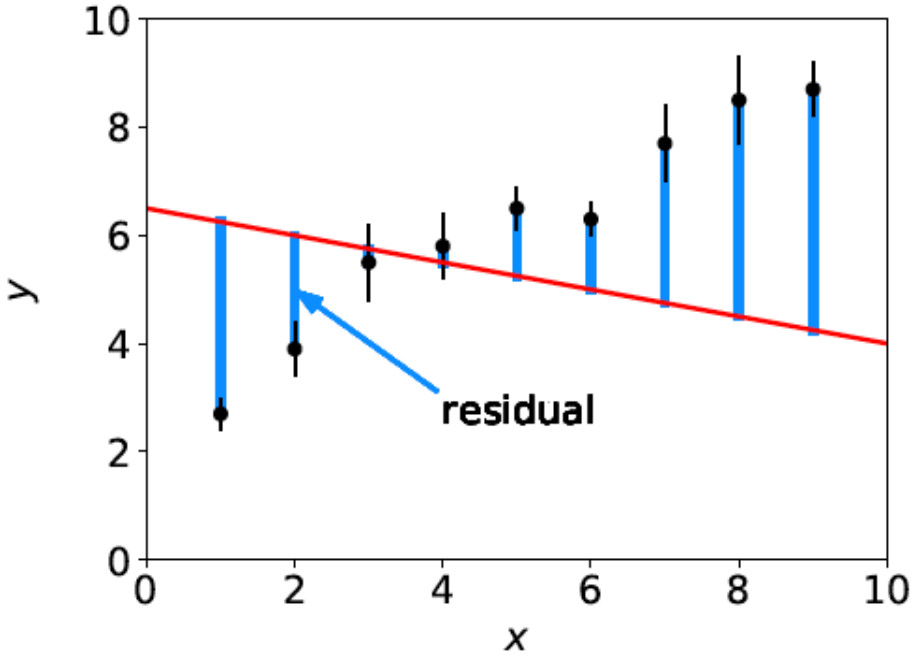
$$f(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$$

vector of parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$

Curve fitting is thus reduced to estimating the parameters.

Least Squares: main idea

Consider fitting a straight line ad suppose we pick an arbitrary point in parameter space (θ_0, θ_1), which gives a certain curve:



Here the curve does not describe the data very well, as can be seen by the large values for the residuals:

residual of i^{th} point $= \; y_i - f(x_i; \boldsymbol{\theta})$

Minimising the residuals

If a measured value y_i has a small σ_i, we want it to be closer to the curve, i.e., measure the distance from point to curve in units of σ_i:

$$\text{normalized residual of } i^{\text{th}} \text{ point} = \frac{y_i - f(x_i;\boldsymbol{\theta})}{\sigma_i}$$

Idea of the method of Least Squares is to choose the parameters that give the minimum of the sum of squared normalized residuals, i.e., we should minimize the "chi-squared":

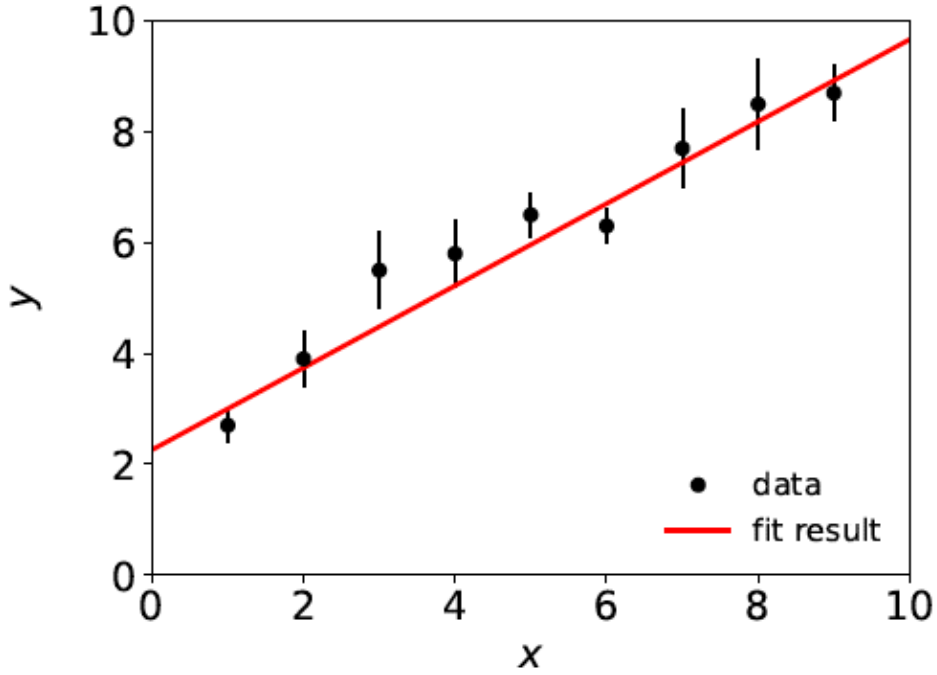
$$\chi^2(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{(y_i - f(x_i;\boldsymbol{\theta}))^2}{\sigma_i^2}$$

Least squares estimators

The values that minimize $\chi^2(\theta)$ are called the least-squares *estimators* for the parameters, written with hats:

$$\hat{\boldsymbol{\theta}} = (\hat{\theta}_0, \hat{\theta}_1)$$

The fitted curve is thus "best" in the least-squares sense:



$$\hat{\theta}_0 = 2.258$$

$$\hat{\theta}_1 = 0.741$$

Comments on LS estimators

We can derive the method of Least Squares from a more general principle called the method of Maximum Likelihood applied to the special case where the y_i are independent and Gaussian distributed:

$$y_i \sim \text{Gauss}(\mu_i, \sigma_i) \ , \qquad \mu_i = f(x_i; \boldsymbol{\theta})$$

It is equally valid to take the minimum of $\chi^2(\boldsymbol{\theta})$ as the definition of the least-squares estimators, and in fact there is no general rule for finding estimators for parameters that are optimal in every relevant sense.

Next steps

1. How do we find the estimators, i.e., how do we minimize $\chi^2(\boldsymbol{\theta})$?

2. How do we quantify the statistical uncertainty in the estimated parameters that stems from the random fluctuations in the measurements, and how is this information used in an analysis problem, e.g., using error propagation?

3. How do we assess whether the hypothesized functional form $f(x; \boldsymbol{\theta})$ adequately describes the data?

Finding estimators in closed form

For a limited class of problem it is possible to find the estimators in closed form. An important example is when the function $f(x; \boldsymbol{\theta})$ is linear *in the parameters* $\boldsymbol{\theta}$, e.g., a polynomial of order M (note the function does not have to be linear in x):

$$f(x;\boldsymbol{\theta}) = \sum_{n=0}^{M} \theta_n x^n$$

As an example consider a straight line (two parameters):

$$f(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$$

We need to minimize: $$\chi^2(\theta_0,\theta_1) = \sum_{i=1}^{N} \frac{(y_i - \theta_0 - \theta_1 x_i)^2}{\sigma_i^2}$$

Finding estimators in closed form (2)

Set the derivatives of $\chi^2(\boldsymbol{\theta})$ with respect to the parameters equal to zero:

$$\frac{\partial \chi^2}{\partial \theta_0} = \sum_{i=1}^{N} \frac{-2(y_i - \theta_0 - \theta_1 x_i)}{\sigma_i^2} = 0 \,,$$

$$\frac{\partial \chi^2}{\partial \theta_1} = \sum_{i=1}^{N} \frac{-2x_i(y_i - \theta_0 - \theta_1 x_i)}{\sigma_i^2} = 0 \,.$$

Finding estimators in closed form (3)

The equations can be rewritten in matrix form as

$$\begin{pmatrix} \sum_{i=1}^{N} \frac{1}{\sigma_i^2} & \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \\ \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} & \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2} \\ \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2} \end{pmatrix}$$

which has the general form

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix}$$

Read off *a, b, c, d, e, f,* by comparing with eq. above.

Finding estimators in closed form (4)

Recall how to invert a 2×2 matrix:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \qquad A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Apply A^{-1} to both sides of previous eq. to find solution (written with hats, because these are the estimators):

$$\hat{\theta}_0 = \frac{de - bf}{ad - bc},$$

$$\hat{\theta}_1 = \frac{af - ec}{ad - bc}.$$

Comments on solution when $f(x;\boldsymbol{\theta})$ is linear in the parameters

Finding solution requires solving a system of linear equations; can be done with standard matrix methods.

Estimators are linear functions of the y_i. This is true in general for problems of this type with an arbitrary number of parameters.

Even though we could find the solution in closed form, the formulas get a bit complicated.

If the fit function $f(x;\boldsymbol{\theta})$ is not linear in the parameters, it is not always possible to solve for the estimators in closed form.

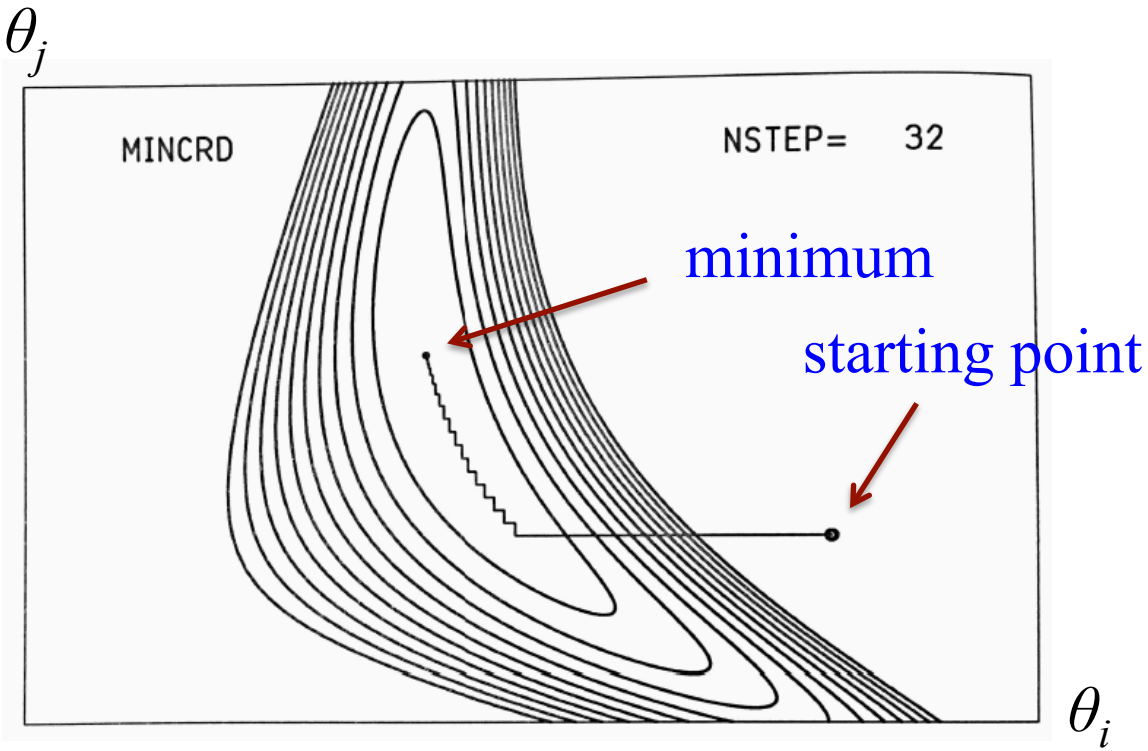
So for many problems we need to find the solution numerically.

Finding LS estimators numerically

Start at a given point in the parameter space and move around according to some strategy to find the point where $\chi^2(\boldsymbol{\theta})$ is a minimum.

For example, alternate minimizing with respect to each component of $\boldsymbol{\theta}$:

Many strategies possible, e.g., steepest descent, conjugate gradients, ... (see Brandt Ch. 10).



Siegmund Brandt, Data Analysis: Statistical and Computational Methods for Scientists and Engineers 4th ed., Springer 2014

Fitting the parameters with Python

The routine routine **`curve_fit`** from **`scipy.optimize`** can find LS estimators numerically. To use it you need:

```
import numpy as np
from scipy.optimize import curve_fit
```

We need to define the fit function $f(x; \theta)$, e.g., a straight line:

```
def func(x, *theta):
    theta0, theta1 = theta
    return theta0 + theta1*x
```

Fitting the parameters with Python (2)

The data values (x_i, y_i, σ_i) need to be in the form of NumPy arrays, e.g,

```
x   = np.array([1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0])
y   = np.array([2.7, 3.9, 5.5, 5.8, 6.5, 6.3, 7.7, 8.5, 8.7])
sig = np.array([0.3, 0.5, 0.7, 0.6, 0.4, 0.3, 0.7, 0.8, 0.5])
```

Start values of the parameters can be specified:

```
p0 = np.array([1.0, 1.0])
```

To find the parameter values that minimize $\chi^2(\theta)$, call **`curve_fit`**:

```
thetaHat, cov = curve_fit(func, x, y, p0, sig, absolute_sigma=True)
```

Returns estimators and covariance matrix as NumPy arrays.

Need **`absolute_sigma=True`** for the fit errors (cov. matrix) to have desired interpretation.

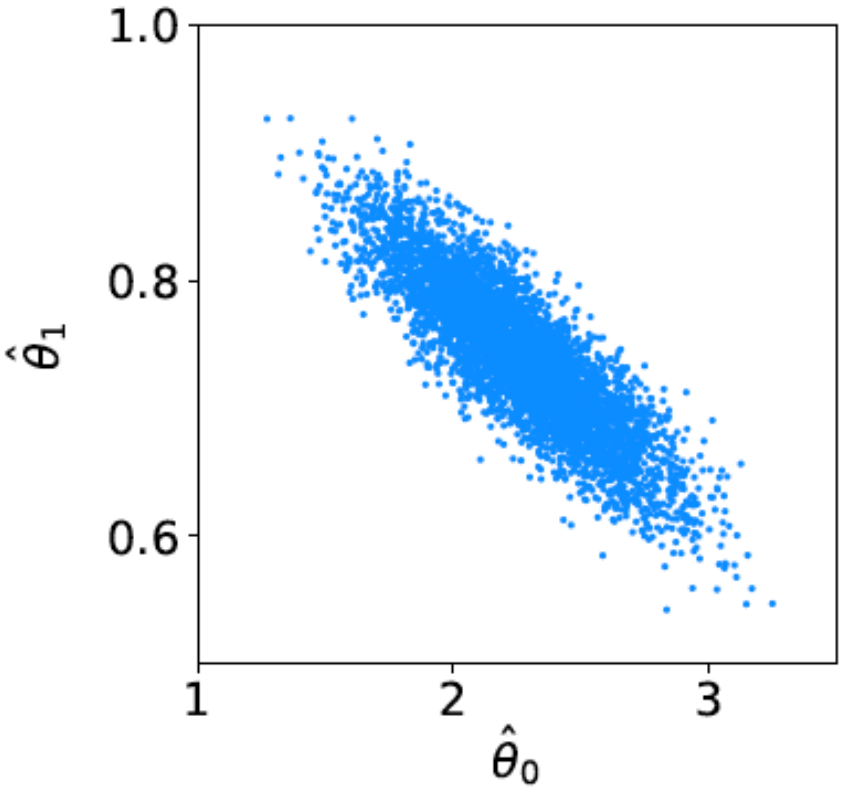
Statistical errors of fitted parameters

The estimators have statistical errors that are due to the random nature of the measured data (the y_i).

If we were to obtain a new independent set of measured values, $y_1,..., y_N$, then these would in general give different values for the estimated parameters.

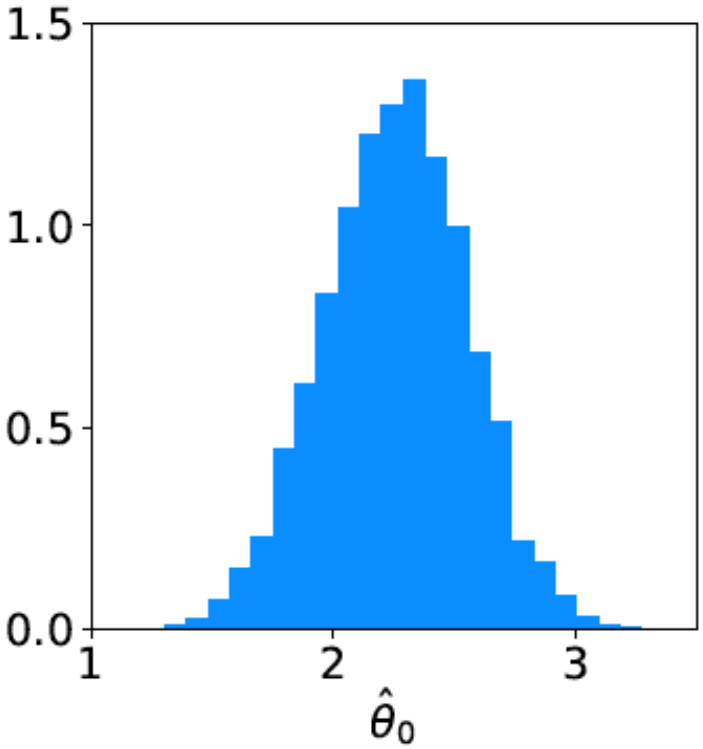
We can simulate the data set $y_1,..., y_N$ many times with the Monte Carlo method.

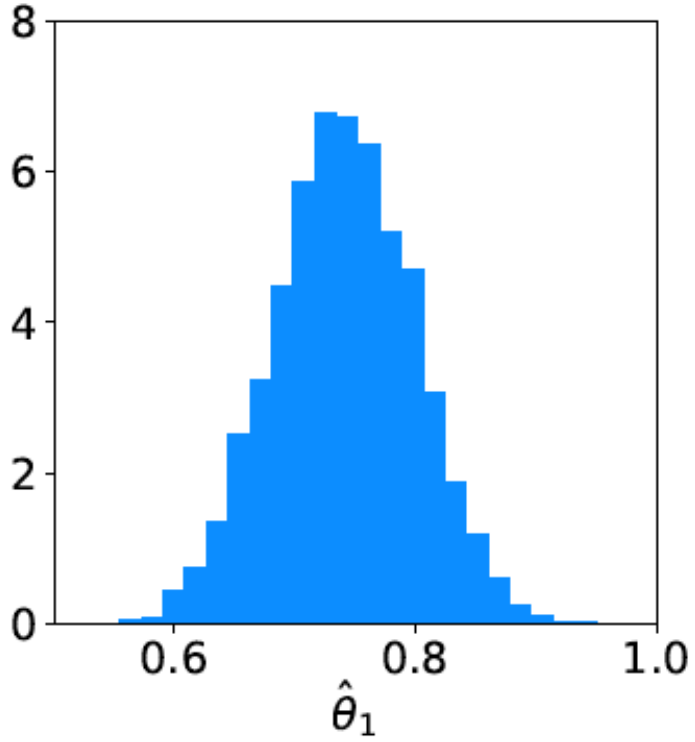
For each set evaluate the estimators for θ_0 and θ_1 from the straight-line fit and enter into a scatter plot:



Statistical errors of fitted parameters (2)

Project points onto the $\hat{\theta}_0$ and $\hat{\theta}_1$ axes:





Each distribution's standard deviation (~width) is used as a measure of the corresponding estimator's statistical error.

(Co)variance, correlation

The scatter plot of $\hat{\theta}_0$ versus $\hat{\theta}_1$ showed that if one estimate came out high, the other tended to be low and vice versa. This indicates that the estimators are (negatively) *correlated*.

To quantify the degree of correlation in any two random variables u and v we define the *covariance*,

$$\mathrm{cov}[u, v] = \langle uv \rangle - \langle u \rangle \langle v \rangle = \langle (u - \langle u \rangle)(v - \langle v \rangle) \rangle$$

The covariance of a variable u with itself is its variance $V[u] = \sigma_u^2$

$$\mathrm{cov}[u, u] = \langle u^2 \rangle - \langle u \rangle^2 = V[u] = \sigma_u^2$$

The square root of the variance = standard deviation σ_u.

Also define dimensionless correlation coefficient (can show $-1 \leq \rho \leq 1$):

$$\rho = \frac{\mathrm{cov}[u, v]}{\sigma_u \sigma_v}$$

Covariance etc. from straight-line fit

From the simulated values shown in the scatter plot, use standard formulae (see RHUL Physics formula book) to obtain the standard deviations and covariance:

$$\begin{aligned} \sigma_{\hat{\theta}_0} &= 0.29 \,, \\ \sigma_{\hat{\theta}_1} &= 0.057 \,, \\ \mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] &= -0.0142 \,, \\ \rho &= -0.86 \,. \end{aligned}$$

Covariance matrix

If we have a set of estimators

$$\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_m)$$

we can find the covariance for each pair and put into a matrix

$$U_{ij} = \mathrm{cov}[\hat{\theta}_i, \hat{\theta}_j]$$

Covariance matrix is square and symmetric.

Diagonal elements are the variances: $U_{ii} = \mathrm{cov}[\hat{\theta}_i, \hat{\theta}_i] = \sigma^2_{\hat{\theta}_i}$

The vector of estimators and their covariance matrix are the two objects returned by the routine **`curve_fit`**:

```
thetaHat, cov = curve_fit(func, x, y, p0, sig, absolute_sigma=True)
```

Covariance from derivatives of $\chi^2(\boldsymbol{\theta})$

It is also possible to obtain the covariance matrix from second derivatives of $\chi^2(\boldsymbol{\theta})$ with respect to the parameters at its minimum.

First find U^{-1},

$$U_{ij}^{-1} = \frac{1}{2}\left(\frac{\partial^2 \chi^2}{\partial\theta_i \partial\theta_j}\right)_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

and then invert to find the covariance matrix U.

This is what **`curve_fit`** does (derivatives computed numerically). Example with straight-line fit gives:

$$U = \begin{pmatrix} 0.08537 & -0.01438 \\ -0.01438 & 0.003275 \end{pmatrix}$$

Using the covariance matrix: error propagation

Suppose we've done a fit for parameters (θ_1,..., θ_m) and obtained estimators and their covariance matrix.

We may then be interested in a given function of the fitted parameters, e.g.,

$$u(\hat{\boldsymbol{\theta}}) = (\hat{\theta}_1\hat{\theta}_2 - 2\hat{\theta}_3)^2$$

What is the standard deviation of the quantity u? That is, how do we *propagate* the statistical errors in the estimated parameters through to u?

Or suppose we have two functions u and v. What is are their standard deviations and what is their covariance cov[u,v]?

The error propagation formulae

By expanding the functions to first order about the parameter estimates, one can show that the covariance is approximately

$$\mathrm{cov}[u, v] \approx \sum_{i,j=1}^{m} \frac{\partial u}{\partial \hat{\theta}_i} \frac{\partial v}{\partial \hat{\theta}_j} U_{ij}$$

and thus the variance for a single function is

$$\sigma_u^2 \approx \sum_{i,j=1}^{m} \frac{\partial u}{\partial \hat{\theta}_i} \frac{\partial u}{\partial \hat{\theta}_j} U_{ij}$$

In the special case where the covariance matrix is diagonal, $U_{ij} = \sigma_i \, \sigma_j \, \delta_{ij}$, we can carry out one of the sums to find

$$\sigma_u^2 \approx \sum_{i=1}^{m} \left(\frac{\partial u}{\partial \hat{\theta}_i} \right)^2 \sigma_{\hat{\theta}_i}^2$$

Comments on error propagation

In general the estimators from a fit *are correlated*, so their full covariance matrix must be used for error propagation.

The approximation of error propagation is that the functions are linear in a region of plus-or-minus one standard deviation about the estimators.

Simple example:

$$u = a\hat{\theta}_1 + b\hat{\theta}_2 \qquad \frac{\partial u}{\partial \hat{\theta}_1} = a \qquad \frac{\partial u}{\partial \hat{\theta}_2} = b$$

$$\begin{aligned} \sigma_u^2 &= a^2 U_{11} + b^2 U_{22} + abU_{12} + baU_{21} \\ &= a^2\sigma_{\hat{\theta}_1}^2 + b^2\sigma_{\hat{\theta}_2}^2 + 2ab\mathrm{cov}[\hat{\theta}_1, \hat{\theta}_2] \end{aligned}$$

Exercise 1: polynomial fit

Exercise 1: Polynomial fit and error analysis

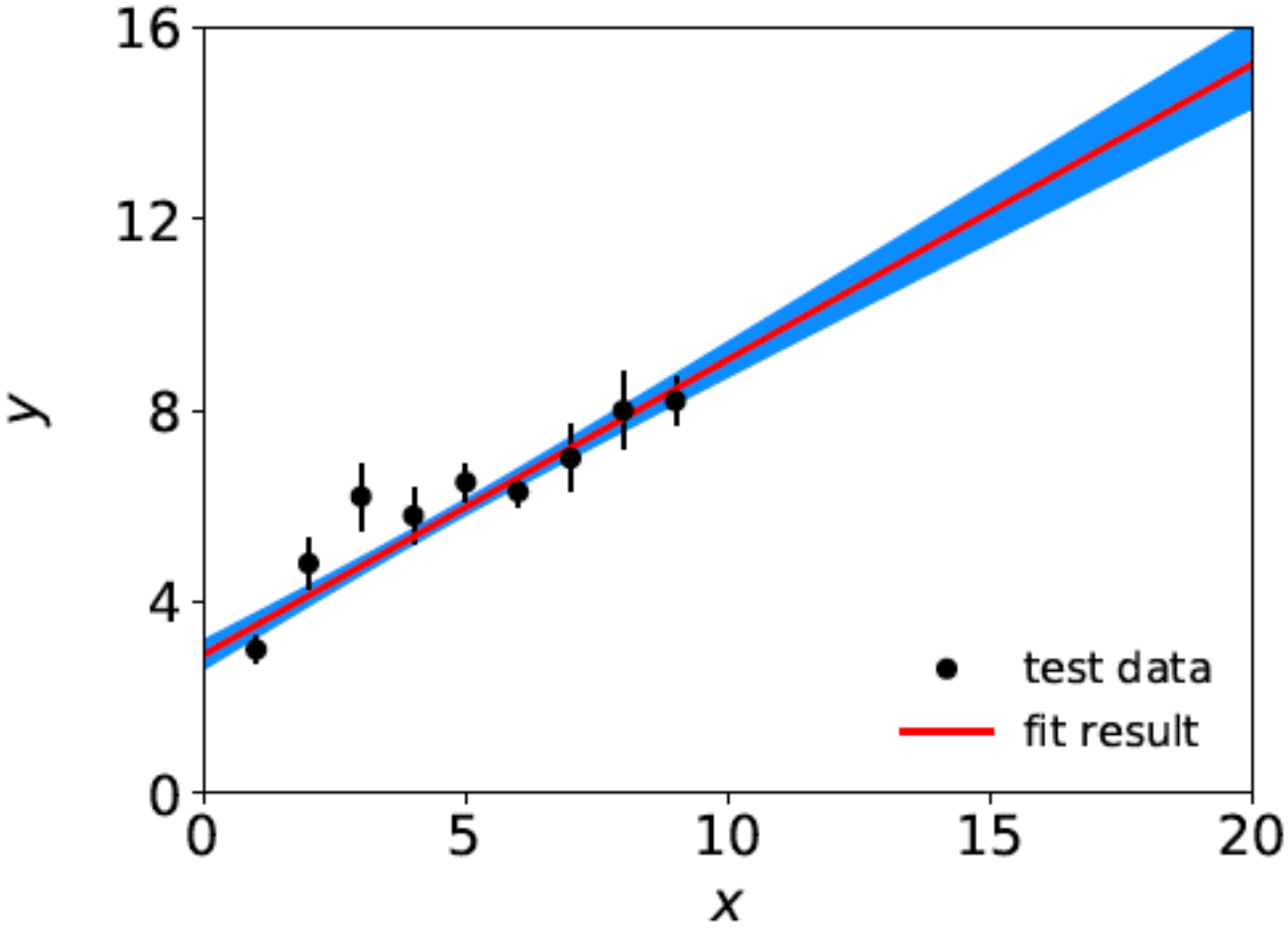
Consider the following set of (x, y, σ) data points:

```
x   = np.array([1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0])
y   = np.array([2.7, 3.9, 5.5, 5.8, 6.5, 6.3, 7.7, 8.5, 8.7])
sig = np.array([0.3, 0.5, 0.7, 0.6, 0.4, 0.3, 0.7, 0.8, 0.5])
```

1(a): Using these data carry out the least-squares fit of an Mth order polynomial, i.e., with $M + 1$ adjustable parameters, for $M = 1, 2, 3$.

1(b): For each fit, use the error propagation formula Eq. (26) to find the standard deviation of the fitted function σ_f as a function of x. Note that to do this you will need to compute the derivatives of $f(x; \hat{\theta})$ with respect to the components of $\hat{\theta}$. Display the fitted curve plus-or-minus one standard deviation as a shaded band, and extend the x axis to at least 20. (The shaded band can be made with the function `matplotlib.fill_between`.) The result for $M = 1$ is shown in Fig. 9. Note how the size of the error band increases when one goes to x values outside the region where data are available; investigate how this behaviour changes as the order of the polynomial is increased.

Polynomial fit: error band



Polynomial fit: error propagation

Consider the difference between the fitted curve values at $x = a$ and $x = b$:

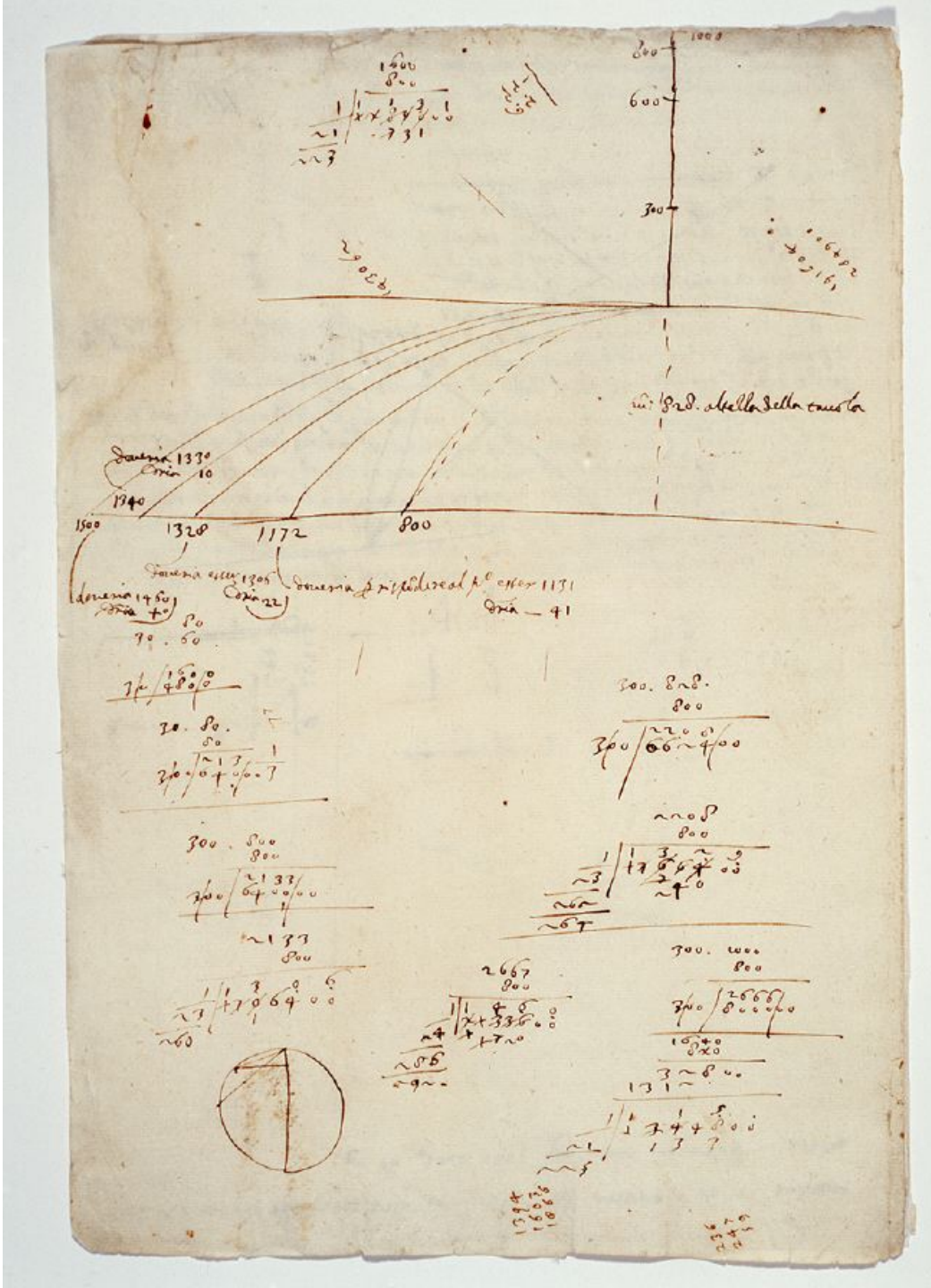
$$\Delta_{ab}(\hat{\boldsymbol{\theta}}) = f(a;\hat{\boldsymbol{\theta}}) - f(b;\hat{\boldsymbol{\theta}})$$

Use error propagation to find the standard deviation of Δ_{ab} (see script).

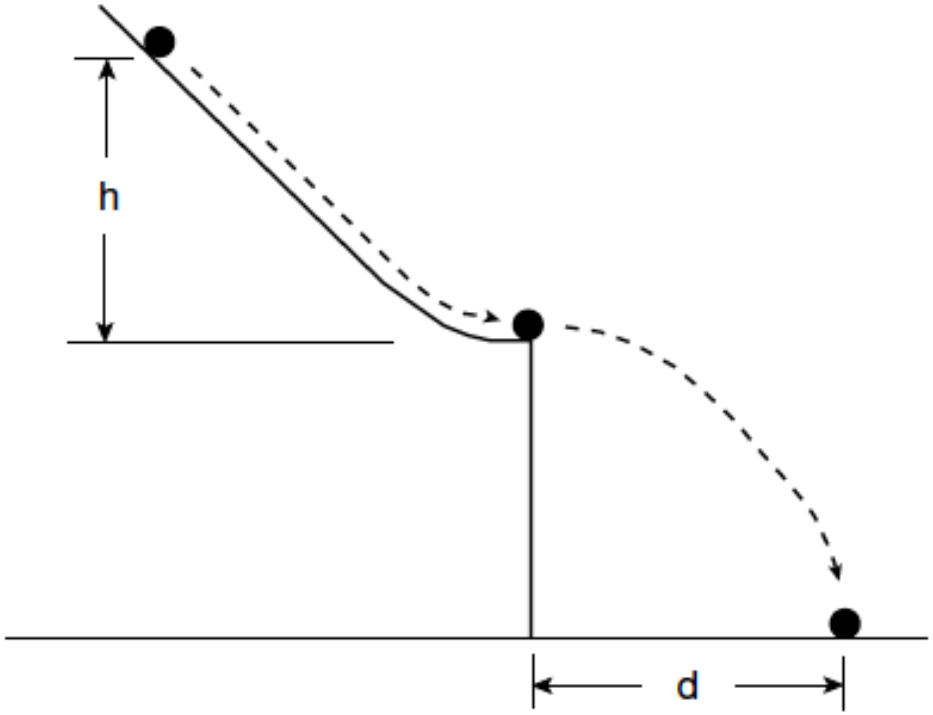
Ball and ramp data from Galileo

Galileo Galilei, Manuscript *f.*116, Biblioteca Nazionale Centrale di Firenze, bncf.firenze.sbn.it

In 1608 Galileo carried out experiments rolling a ball down an inclined ramp to investigate the trajectory of falling objects.



Ball and ramp data from Galileo



Units in punti
(approx. 1 mm)

h	d
1000	1500
828	1340
800	1328
600	1172
300	800

Suppose h is set with negligible uncertainty, and d is measured with an uncertainty $\sigma = 15$ punti.

Analysis of ball and ramp data

What is the correct law that relates d and h?

Try different hypotheses:

$$d = \alpha h$$

$$d = \alpha h + \beta h^2$$

$$d = \alpha h^{\beta}$$

For now, fit the parameters α and β, find their standard deviations and covariance.

Next week we will discuss how to test whether a given hypothesized function is in good or bad agreement with the data.

Extra slides

History

Least Squares fitting also called "regression"

F. Galton, *Regression towards mediocrity in hereditary stature*, The Journal of the Anthropological Institute of Great Britain and Ireland. 15: 246–263 (1886).

Developed earlier by Laplace and Gauss:

C.F. Gauss, *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambentium*, Hamburgi Sumtibus Frid. Perthes et H. Besser Liber II, Sectio II (1809);

C.F. Gauss, *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae*, pars prior (15.2.1821) et pars posterior (2.2.1823), Commentationes Societatis Regiae Scientiarium Gottingensis Recectiores Vol. V (MDCCCXXIII).