

Data Fitting and Time Series Analysis

by Dominic Waithe UKRI Innovation Fellow.

12th December 2019

[IAFIG-RMS - Bioimage Analysis With Python](#)

[Cambridge Bioinformatics Training Centre](#)

TODAYS TALK:

- Data Fitting
- Time series Analysis
- Fluorescence Correlation Spectroscopy
- Fluorescence Recovery After Photobleaching
- Summary

UK Research
and Innovation



Data fitting

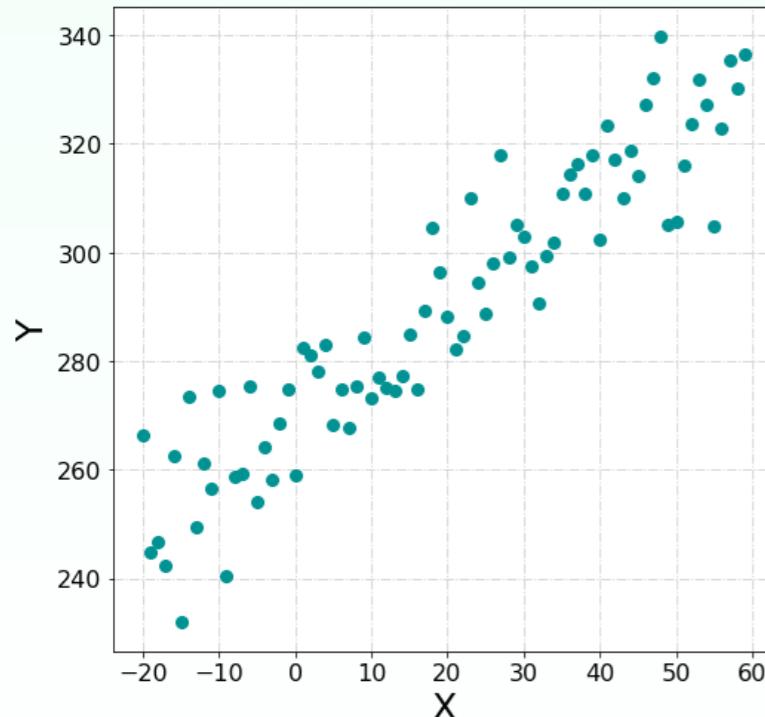
Why do we fit models to data.

“Data fitting is the process of fitting models to data and analyzing the accuracy of the fit. Engineers and scientists use data fitting techniques, including mathematical equations and nonparametric methods, to model acquired data.” - <https://uk.mathworks.com/discovery/data-fitting.html>

- Why do we fit? We fit data so we can make inferences (conclusions) about that data.
- On Friday you will see us fit models to data for machine learning. Here we use the model to help us to quantify and measure our acquired data.

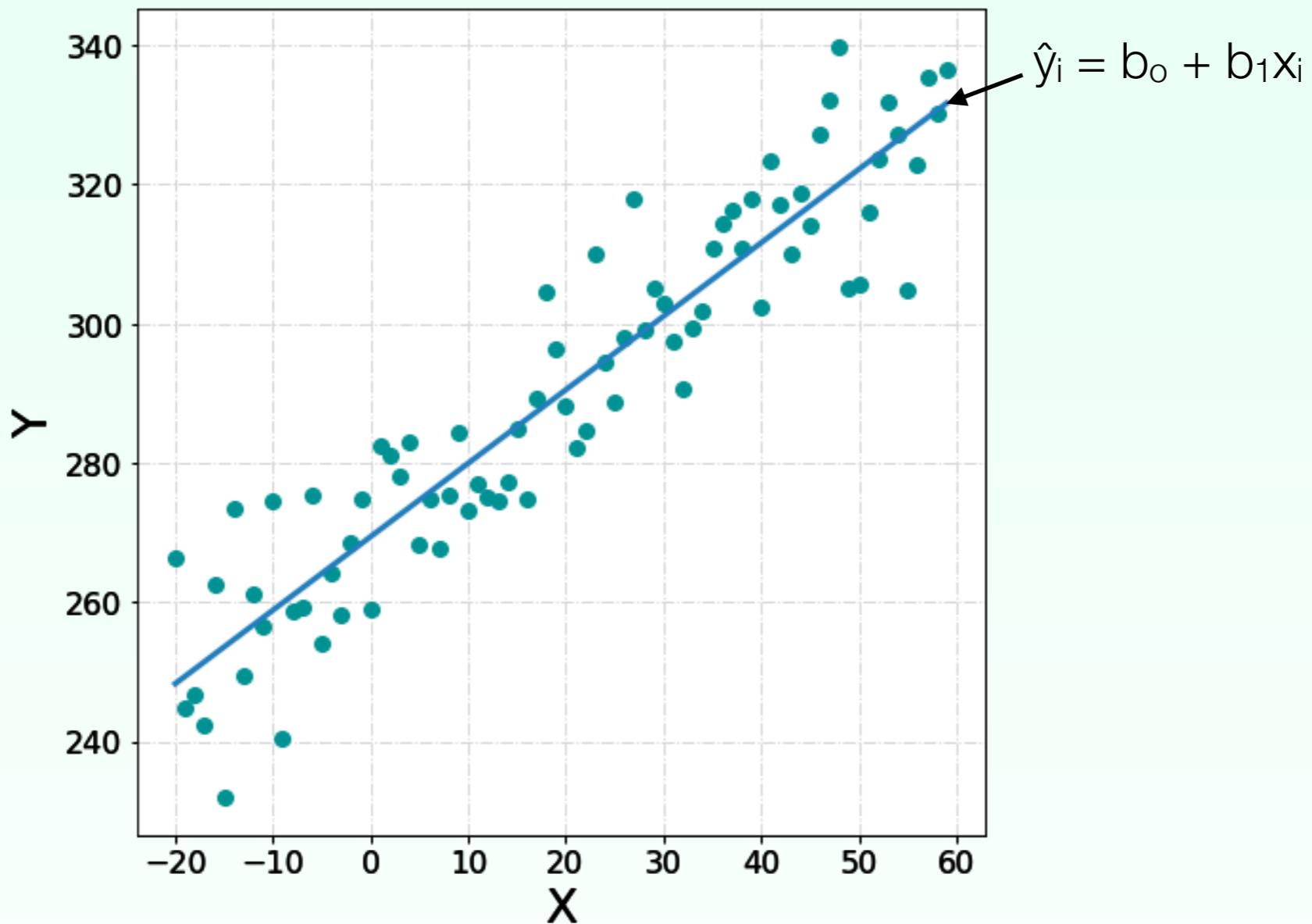
Simple Linear regression

- is a statistical method that allows us to summarise and study relationships between two **continuous** (quantitative variables)
 - One variable often denoted (x), is regarded as the **predictor**, **explanatory** or **independent** variable.
 - The other variable, called (y) is regarded as the **target**, **response**, **outcome** or **dependent** variable



Mastering linear regression,
(i.e. fitting a line to data) is an excellent starting point for mastering data analysis.

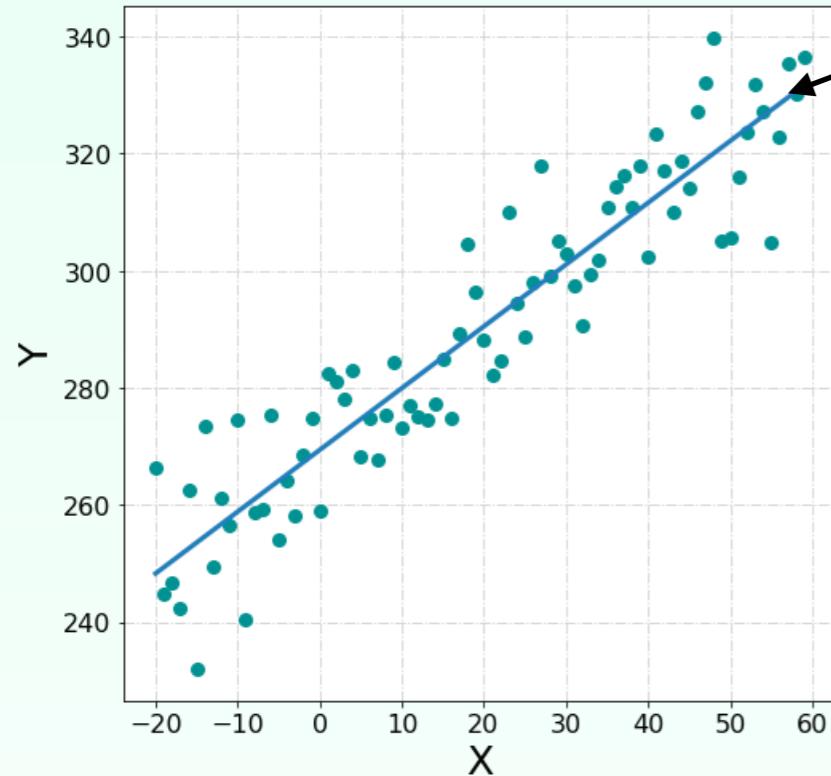
Data fitting



We often want to fit a model to data (e.g. a straight line). We do this so we can get parameters from our model and also to make predictions.

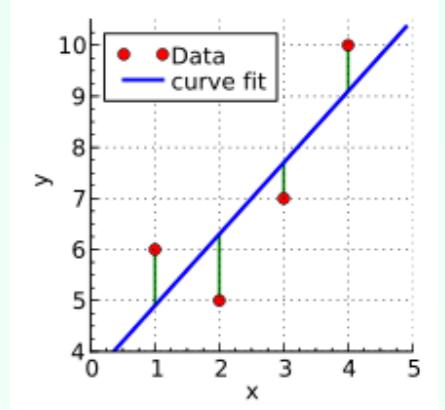
Source:

So how do we fit? Analytical Method



$$b_0 = \frac{\sum y_i - b_1 \sum x_i}{n}$$

$$b_1 = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sum x_i^2 - n \bar{x}^2}$$



With a straight-line and points, this is simple.

We know the math and so we can go straight to the solution.

$$\bar{y} = \frac{\sum y_i}{n}$$
$$\bar{x} = \frac{\sum x_i}{n}$$

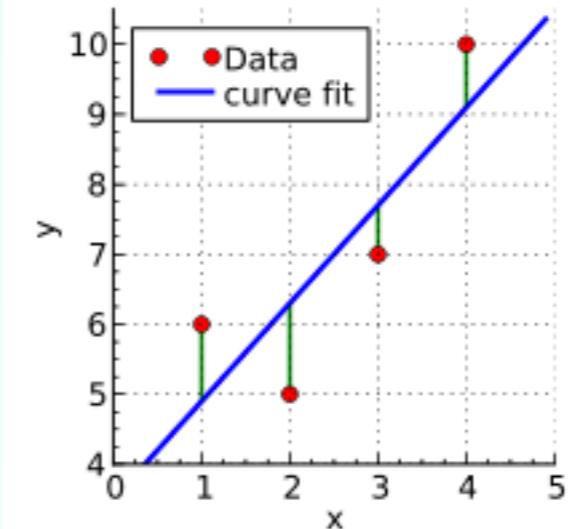
n (number of samples)

This is an analytical solution. An analytical solution to a problem is one that has a "proof": a series of logical steps that can be followed and verified as correct.

An analytical proof:

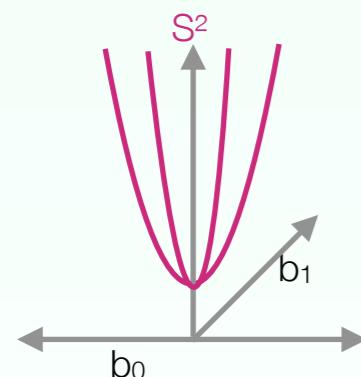
Let's take the following equations. The error for each point can be defined as:

$$\begin{array}{ll} x & y \\ b_0 + 1 \cdot b_1 = 6 & \text{the error is } 6 - (b_0 + 1 \cdot b_1) \\ b_0 + 2 \cdot b_1 = 5 & \text{the error is } 5 - (b_0 + 2 \cdot b_1) \\ b_0 + 3 \cdot b_1 = 7 & \text{the error is } 7 - (b_0 + 3 \cdot b_1) \\ b_0 + 4 \cdot b_1 = 10 & \text{the error is } 10 - (b_0 + 4 \cdot b_1) \end{array}$$



The sum of differences. We want to minimise this, that is the optimum.

$$S^2(b_0, b_1) = (6 - (b_0 + 1 \cdot b_1))^2 + (5 - (b_0 + 2 \cdot b_1))^2 + (7 - (b_0 + 3 \cdot b_1))^2 + (10 - (b_0 + 4 \cdot b_1))^2$$



$$\frac{\delta(S^2)}{\delta b_0} = -2 \sum [y_i - (b_0 + b_1 \cdot x_i)] = 0$$

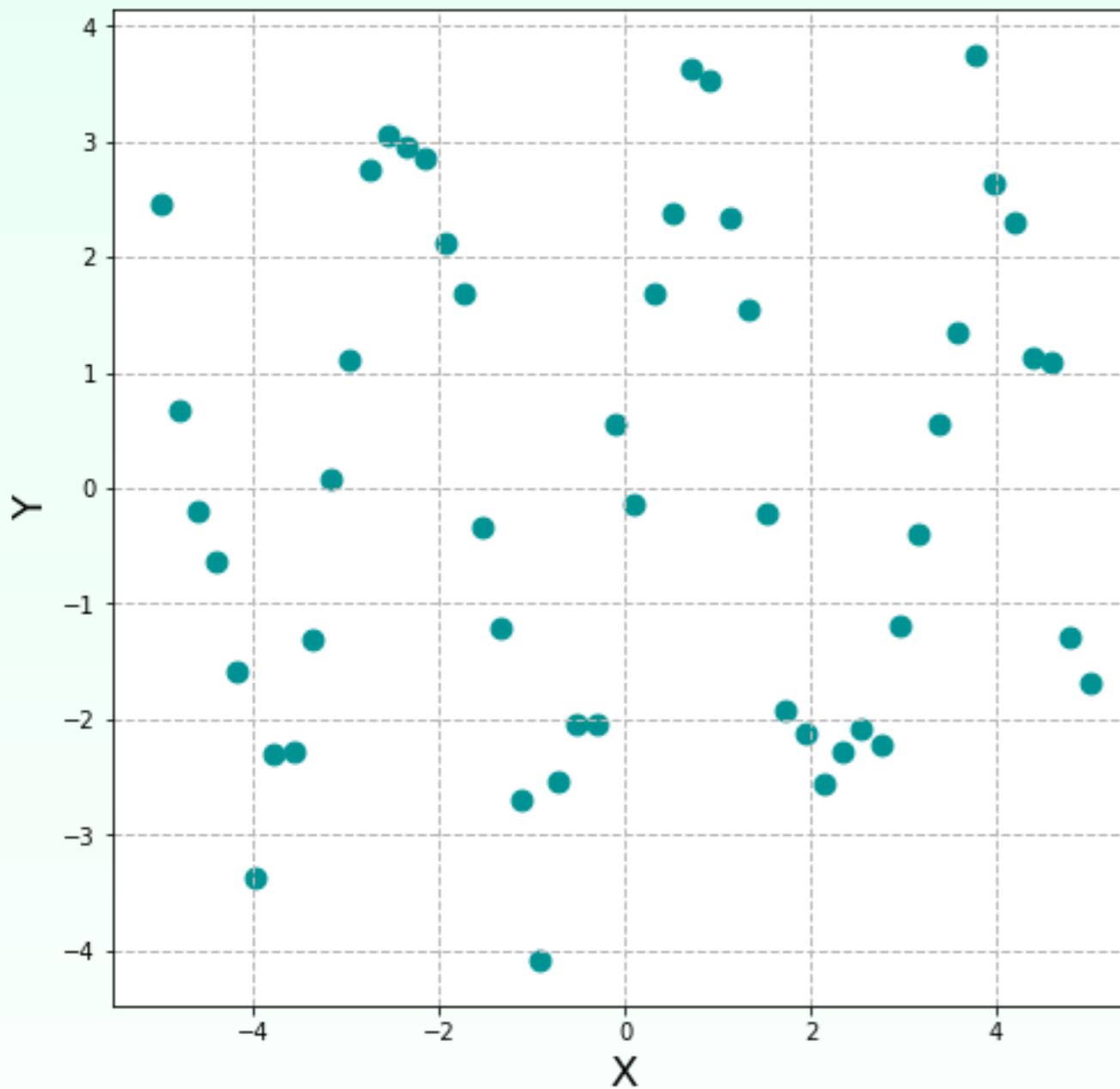
$$\frac{\delta(S^2)}{\delta b_1} = -2 \sum [y_i - (b_0 + b_1 \cdot x_i)] \cdot x_i = 0$$

$$b_1 = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sum x_i^2 - n \bar{x}^2}$$

$$b_0 = \frac{\sum y_i - b_1 \sum x_i}{n}$$

Source: <https://math.stackexchange.com/questions/483339/proof-of-convexity-of-linear-least-squares>

Numerical Optimisation

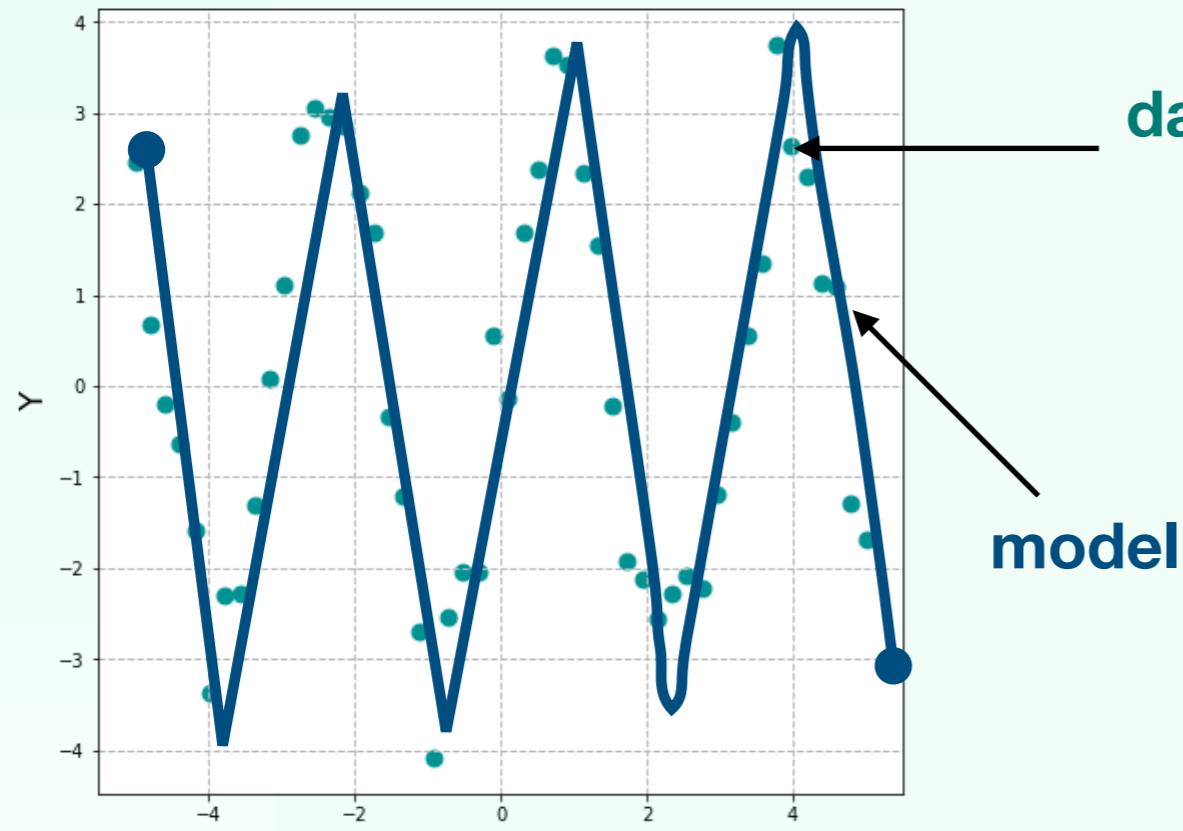


Some distributions are more complicated and so we can't jump to the solution analytically.

We have to do it numerically. We search for the best solution by tweaking the parameters. This works for simpler models too.

Iterative numerical optimisation

With numerical optimisation we try different parameters and evaluate the error (or loss) function. Our goal is to minimise the error between our model and our data.



Error plot

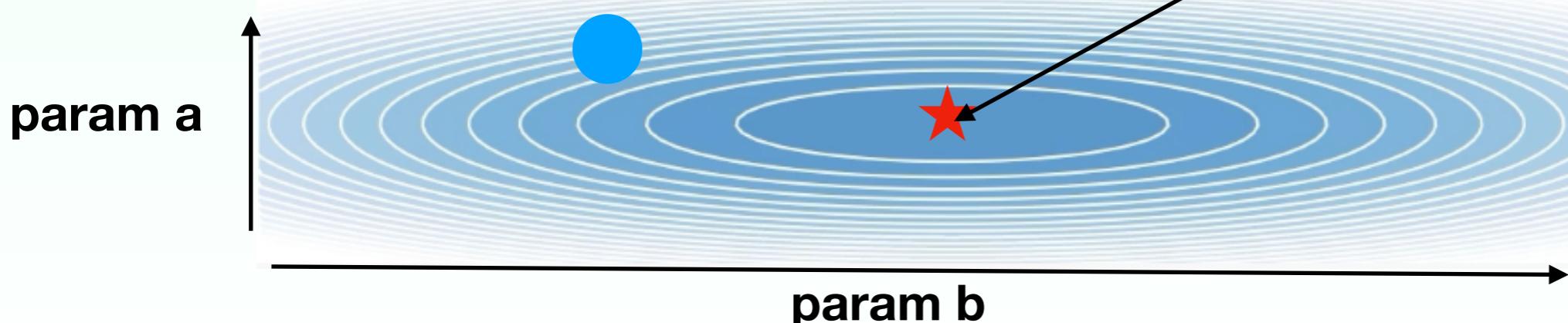
data-points

model

Error, represents the difference between our model and in the input data.

The worse the model fits, the greater the error.

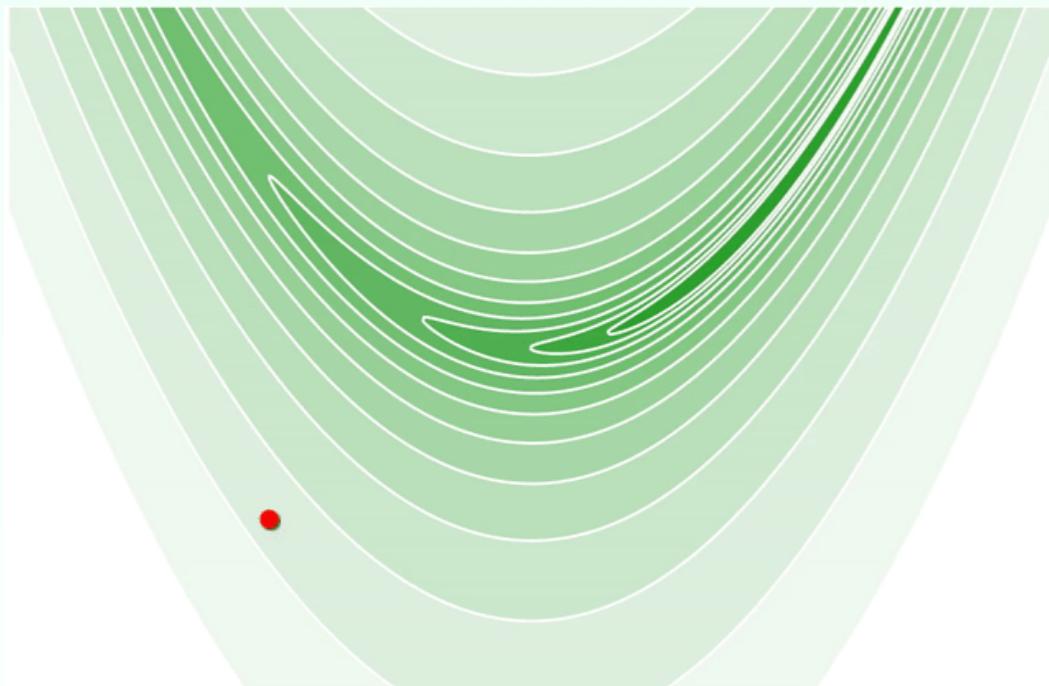
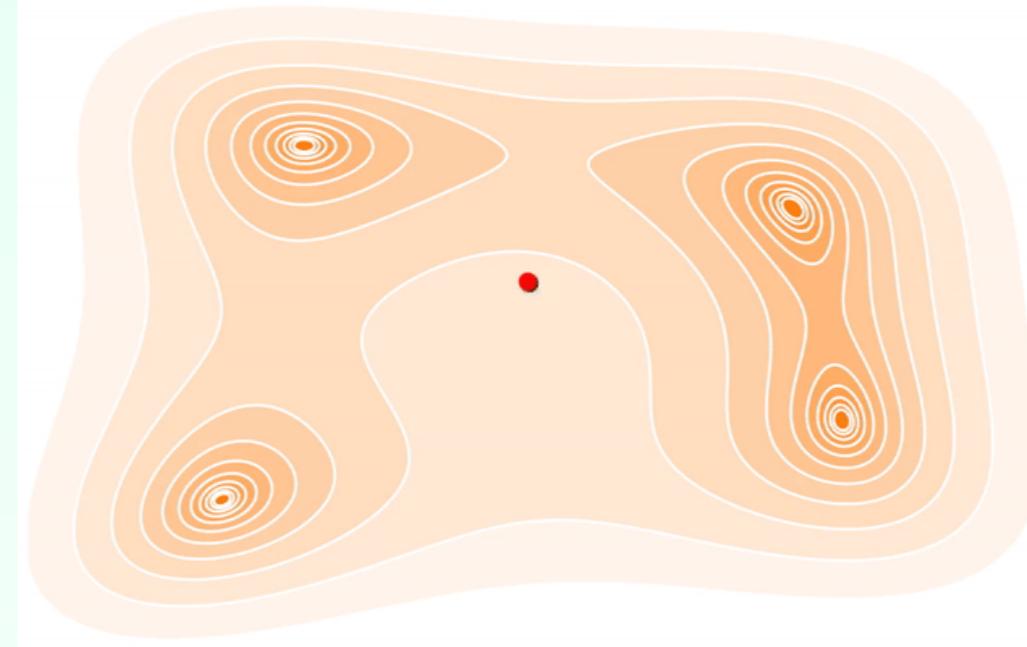
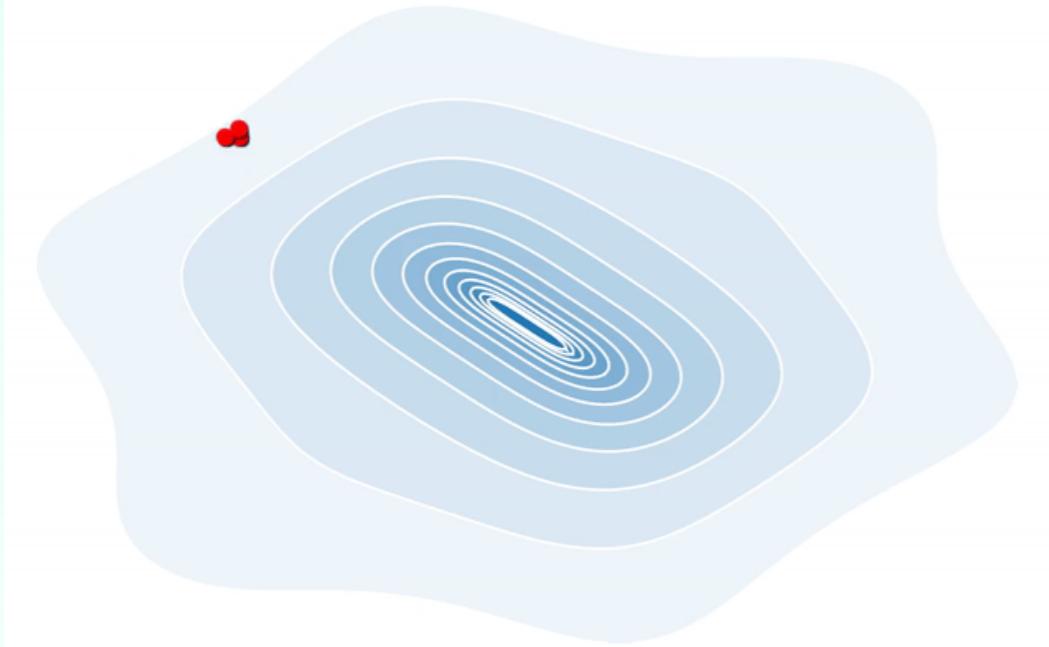
Global minima (lowest error).



We want to tweak the parameters (a and b), and minimise the Sum of Error.

We do it here on a straight-line model for simplicity.

Some loss functions are more complicated.



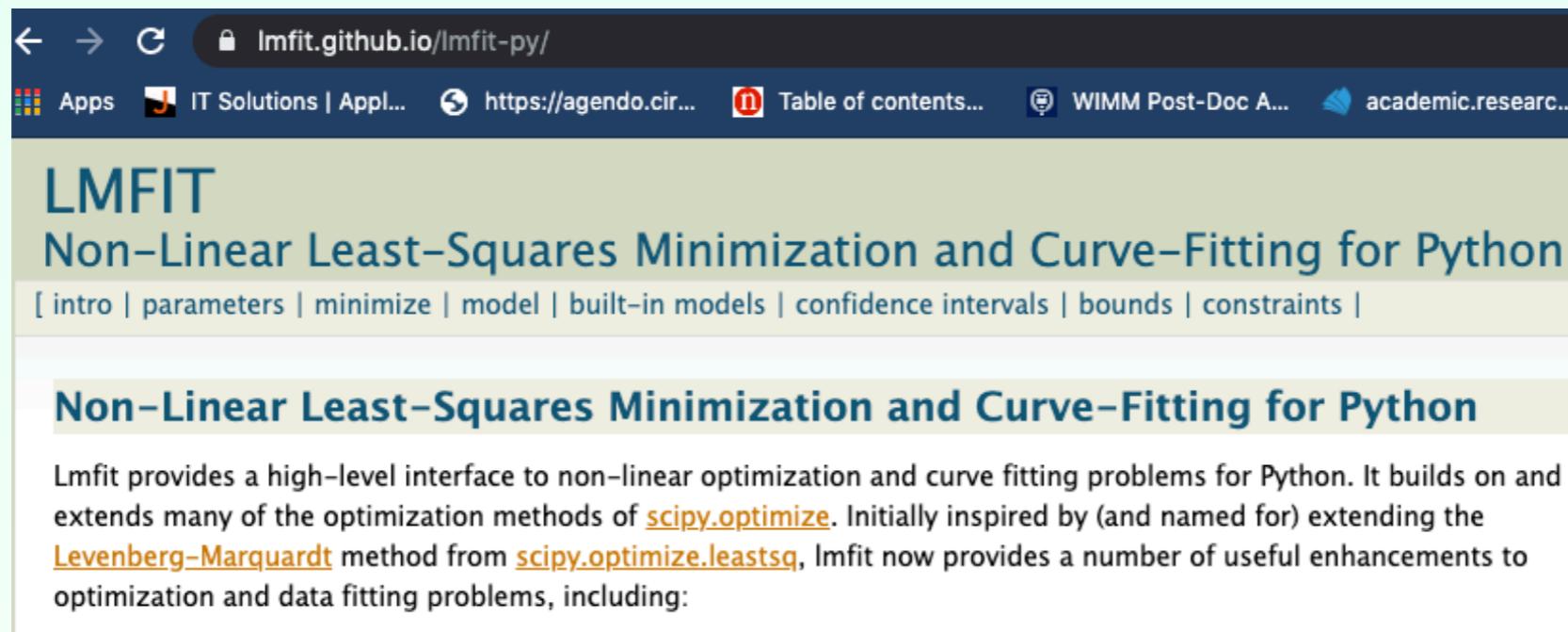
Nonlinear optimization (some famous examples):

- BFGS method
- Gauss–Newton algorithm:
- Levenberg–Marquardt algorithm:
- Nelder–Mead method
- Gradient descent

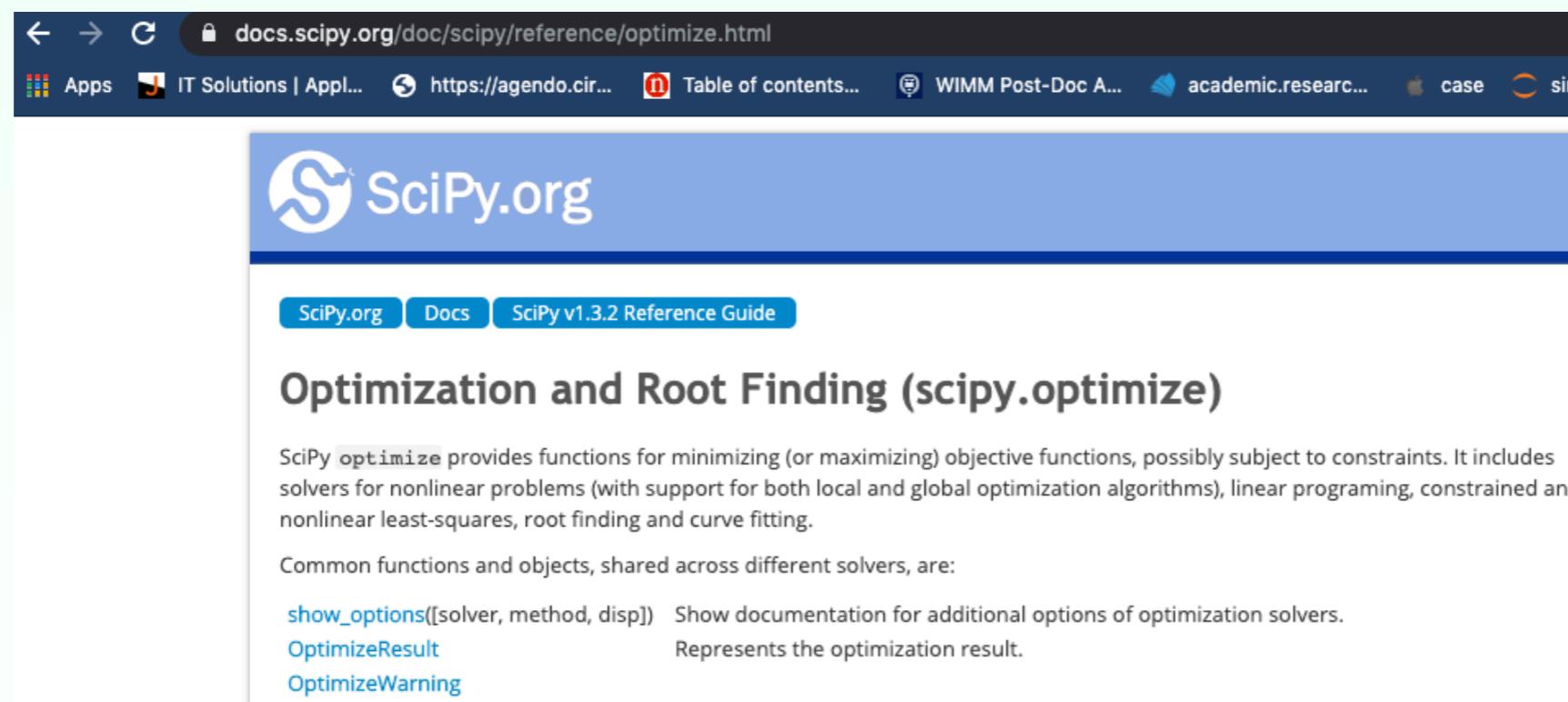
Some loss functions are more complicated and have multiple minima (global minima not guaranteed). Different algorithms exist which can optimise different problems.

Source: https://github.com/benfred/fmin/blob/master/images/nelder_mead.gif https://github.com/benfred/fmin/blob/master/images/gradient_descent.gif https://github.com/benfred/fmin/blob/master/images/conjugate_gradient.gif

Python has several data-fitting/optimization libraries



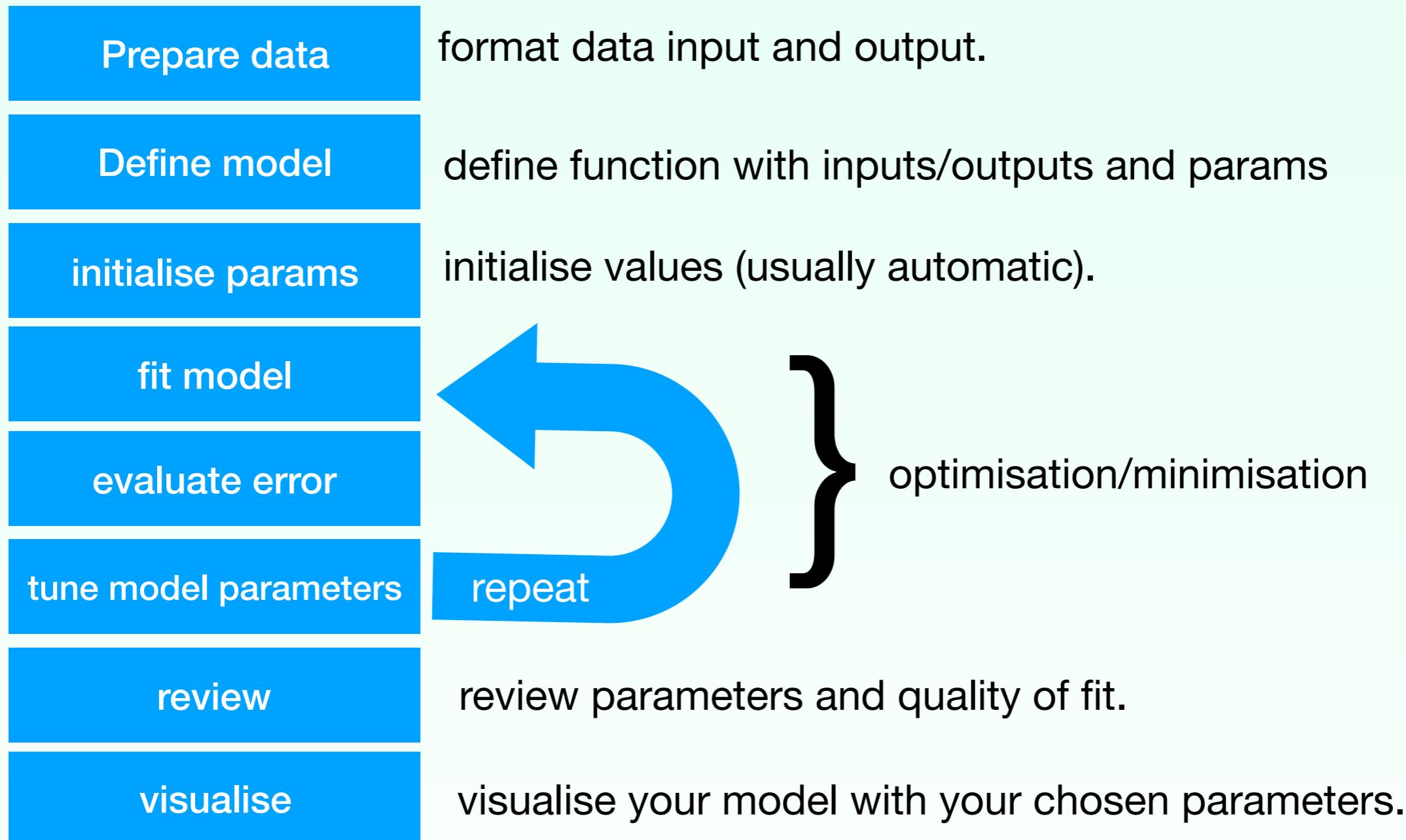
The screenshot shows a web browser displaying the Lmfit library documentation at lmfit.github.io/lmfit-py/. The title is "LMFIT Non-Linear Least-Squares Minimization and Curve-Fitting for Python". Below the title is a navigation bar with links to [intro | parameters | minimize | model | built-in models | confidence intervals | bounds | constraints]. The main content section is titled "Non-Linear Least-Squares Minimization and Curve-Fitting for Python". It describes Lmfit as a high-level interface for non-linear optimization and curve fitting in Python, building on and extending the [scipy.optimize](#) module. It highlights the Levenberg-Marquardt method from [scipy.optimize.leastsq](#) and provides enhancements like bounds and constraints.



The screenshot shows a web browser displaying the SciPy.org documentation for the "Optimization and Root Finding (scipy.optimize)" module. The title is "SciPy.org Docs SciPy v1.3.2 Reference Guide". The main content section is titled "Optimization and Root Finding (scipy.optimize)". It describes the `optimize` module, which provides functions for minimizing (or maximizing) objective functions, possibly subject to constraints. It includes solvers for nonlinear problems, linear programming, constrained and nonlinear least-squares, root finding, and curve fitting. It lists common functions and objects shared across different solvers, such as `show_options`, `OptimizeResult`, and `OptimizeWarning`.

The optimisation algorithm iteratively adapts the parameters to reduce the loss/error to get to the global optimum efficiently. There are many choices as to which optimisation and algorithm to choose.

General approach.



All this is done with normal Python code. You will see it in the practical.

Source:

Reviewing the fit.

Table of Fit Results: These values, including the standard Goodness-of-Fit statistics, are all attributes of the [MinimizerResult](#) object returned by [minimize\(\)](#) or [Minimizer.minimize\(\)](#).

Attribute Name	Description / Formula
nfev	number of function evaluations
nvarys	number of variables in fit N_{varys}
ndata	number of data points: N
nfree	degrees of freedom in fit: $N - N_{\text{varys}}$
residual	residual array, returned by the objective function: $\{\text{Resid}_i\}$
chisqr	chi-square: $\chi^2 = \sum_i^N [\text{Resid}_i]^2$
redchi	reduced chi-square: $\chi_v^2 = \chi^2 / (N - N_{\text{varys}})$
aic	Akaike Information Criterion statistic (see below)
bic	Bayesian Information Criterion statistic (see below)
var_names	ordered list of variable parameter names used for init_vals and covar
covar	covariance matrix (with rows/columns using var_names)
init_vals	list of initial values for variable parameters

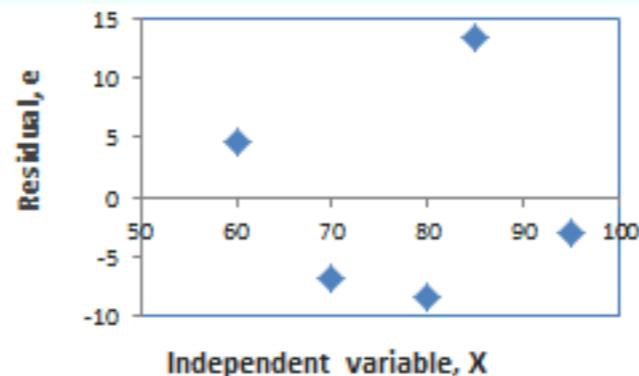
The minimiser returns a number of metrics which can be used to evaluate the quality of the fit.

Source:

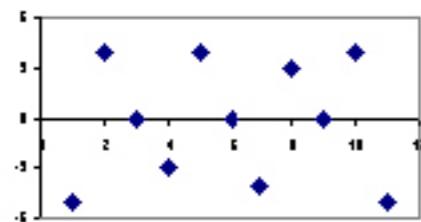
A note about residuals

$$\text{Residual} = \text{Observed value} - \text{Predicted value}$$
$$e = y - \hat{y}$$

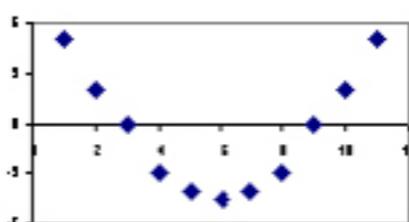
- A residual plot, shows this error for each datapoint x.



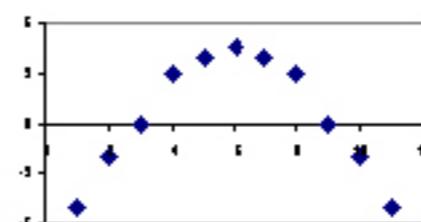
Residuals measure the error between our data and our predicted line.



Random pattern



Non-random: U-shaped



Non-random: Inverted U

If your data, doesn't look randomly distributed, rethink your model.

Source:

Time-series analysis

Time series analysis

A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time.

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data.

Microscopy examples

Mean Squared Displacement (MSD)

Stochastic optical reconstruction microscopy (STORM)

Photo Activation Localisation Microscopy (PALM)

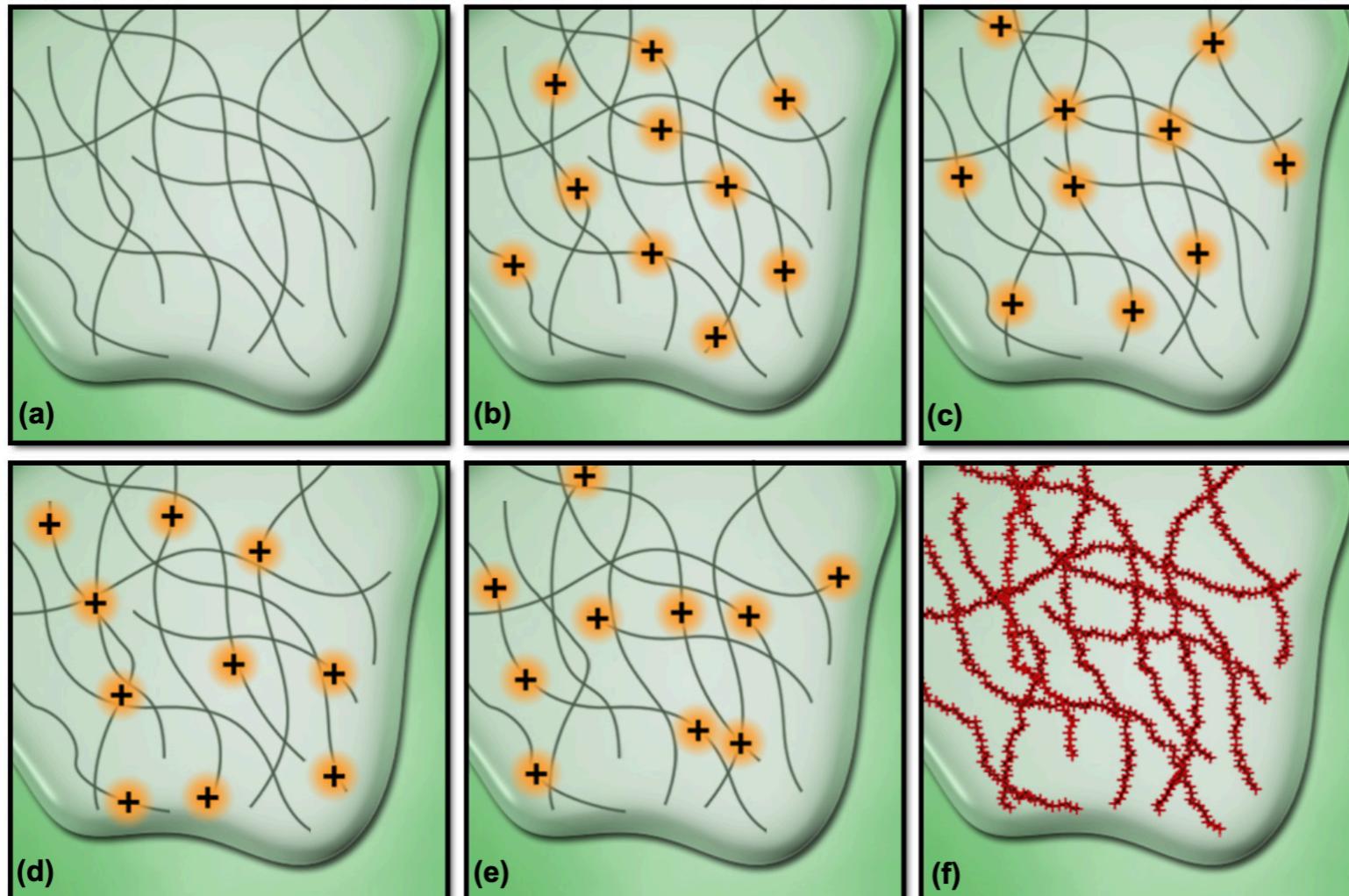
Fluorescence Correlation Spectroscopy (FCS)

Fluorescence Recovery After Photobleaching (FRAP)

Source: https://en.wikipedia.org/wiki/Time_series

STORM imaging

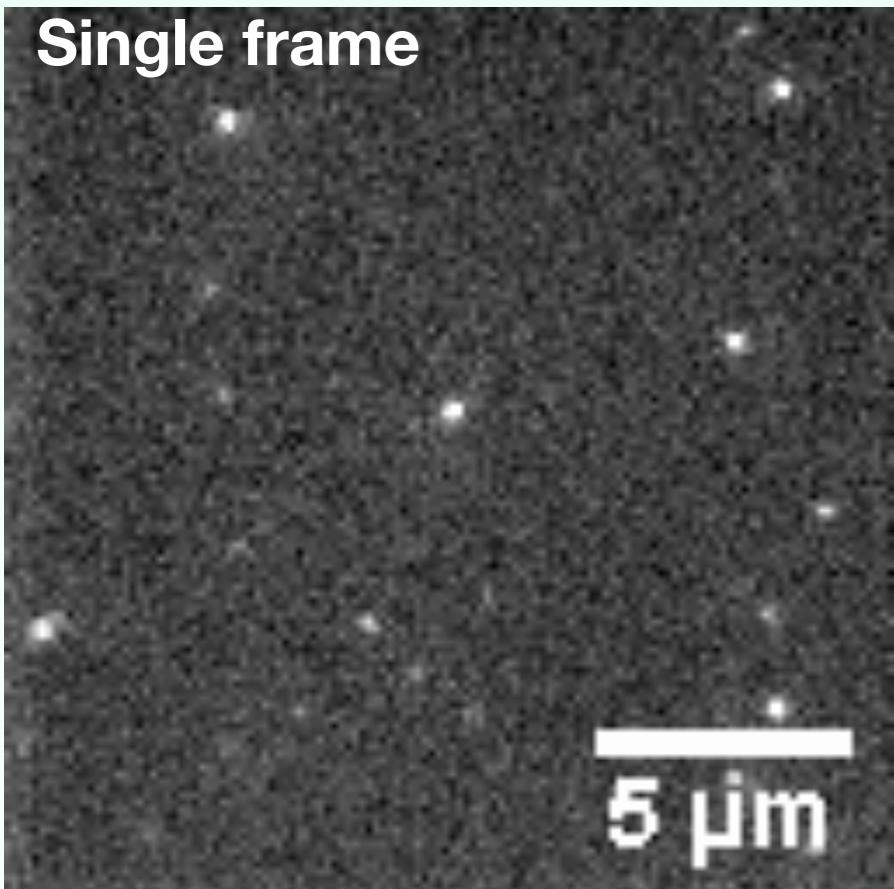
Figure 1 - The Basic Principle of STORM Imaging



We force our fluorophores into a dark state so blinking is transient and occurring sparsely across our sample over time.

Source: <https://www.microscopyu.com/techniques/super-resolution/single-molecule-super-resolution-imaging>

STORM imaging and reconstruction



Multi-frames



Source: <https://en.wikipedia.org/wiki/File:FrameWithSingleMolecules.png> https://en.wikipedia.org/wiki/Super-resolution_microscopy#/media/File:GFP_Superresolution_Christoph_Cremer.JPG <https://www.microscopyu.com/techniques/super-resolution/single-molecule-super-resolution-imaging>

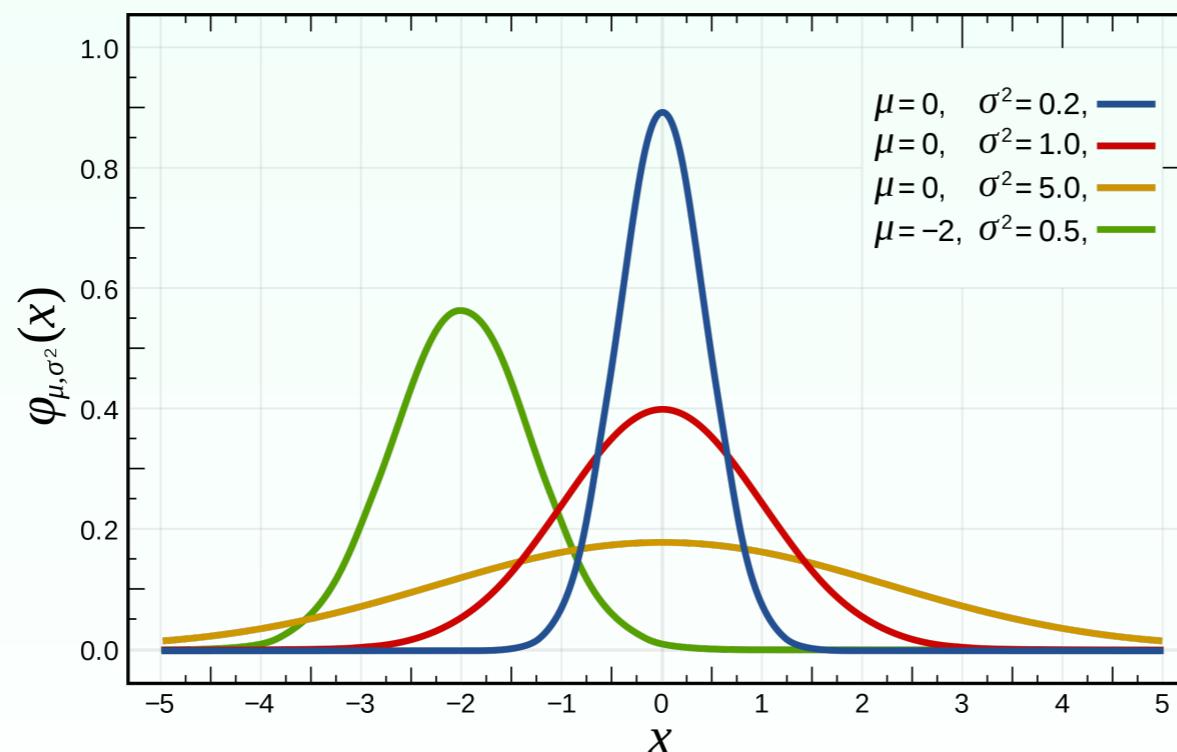
Gaussian (or normal) distribution

The most important distribution you will come across.

Pops up a lot, especially in biology due to central limit theorem.

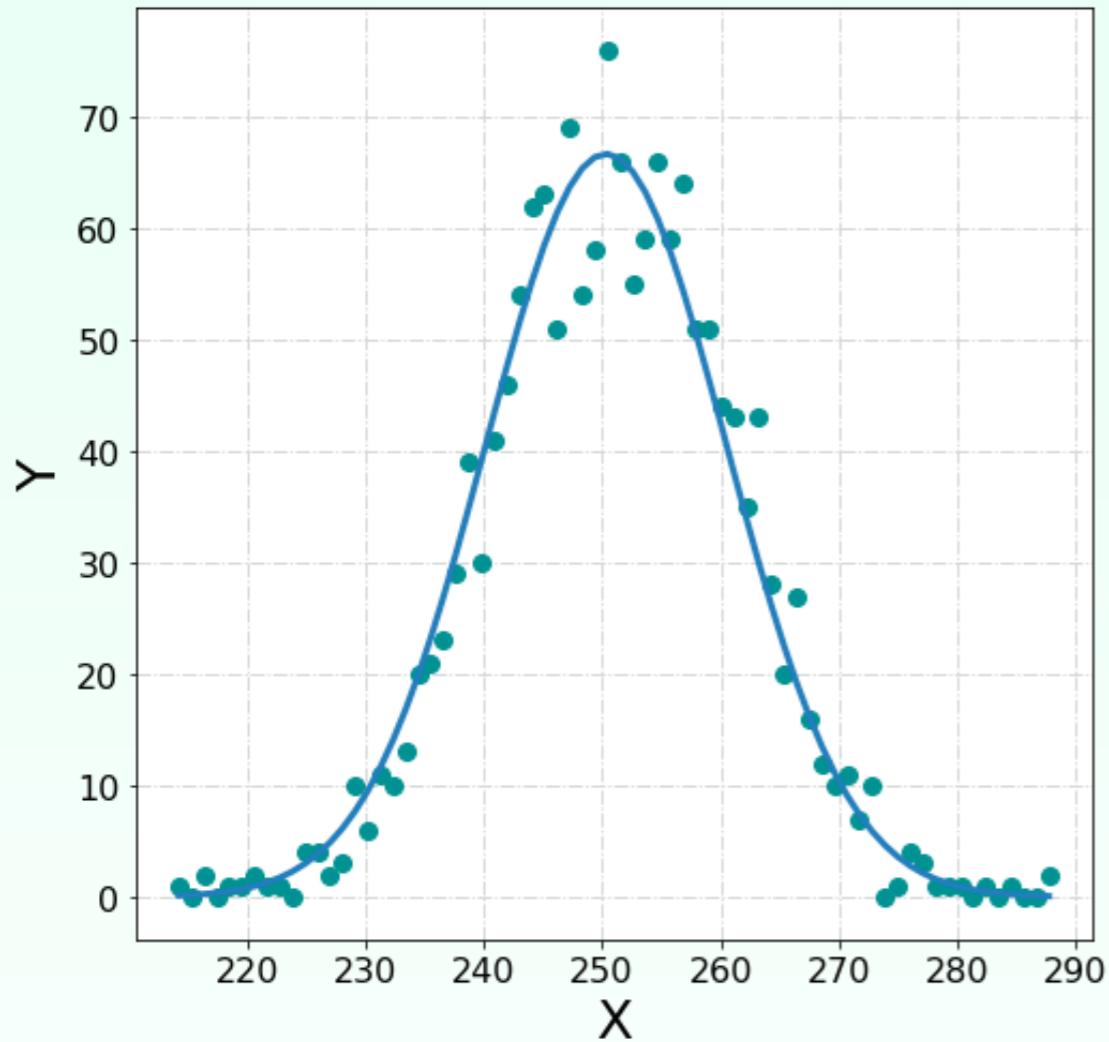
If you sample randomly from any type of distribution many times, the resulting samples will tend toward a Gaussian distribution.

$$y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{(2\sigma^2)}}$$



Source: https://en.wikipedia.org/wiki/Central_limit_theorem

Fitting with Analytical solution?



$$y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{(2\sigma^2)}}$$

parameters

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

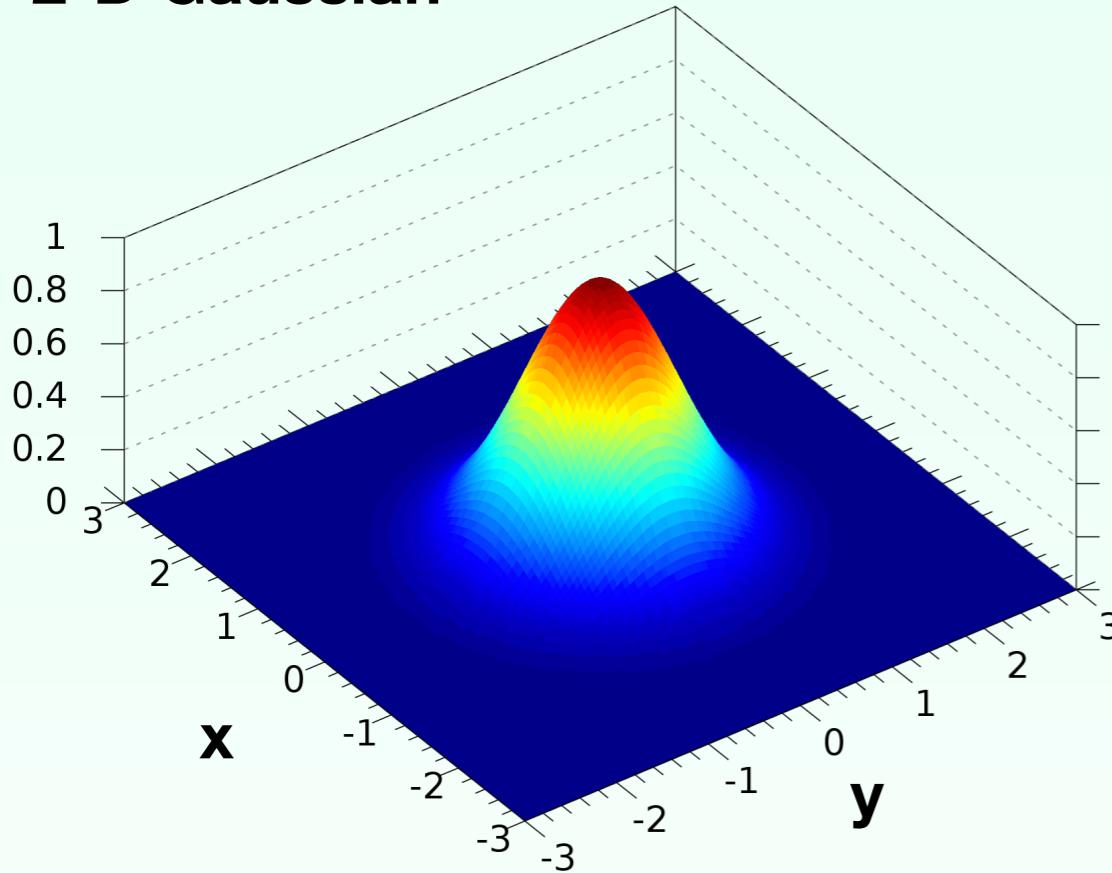
$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Normal or Gaussian distribution. This distribution is also well understood and can be fit analytically. However this does not perform well in the presence of noise. Therefore we tend to fit Gaussians numerically and use the above to estimate.

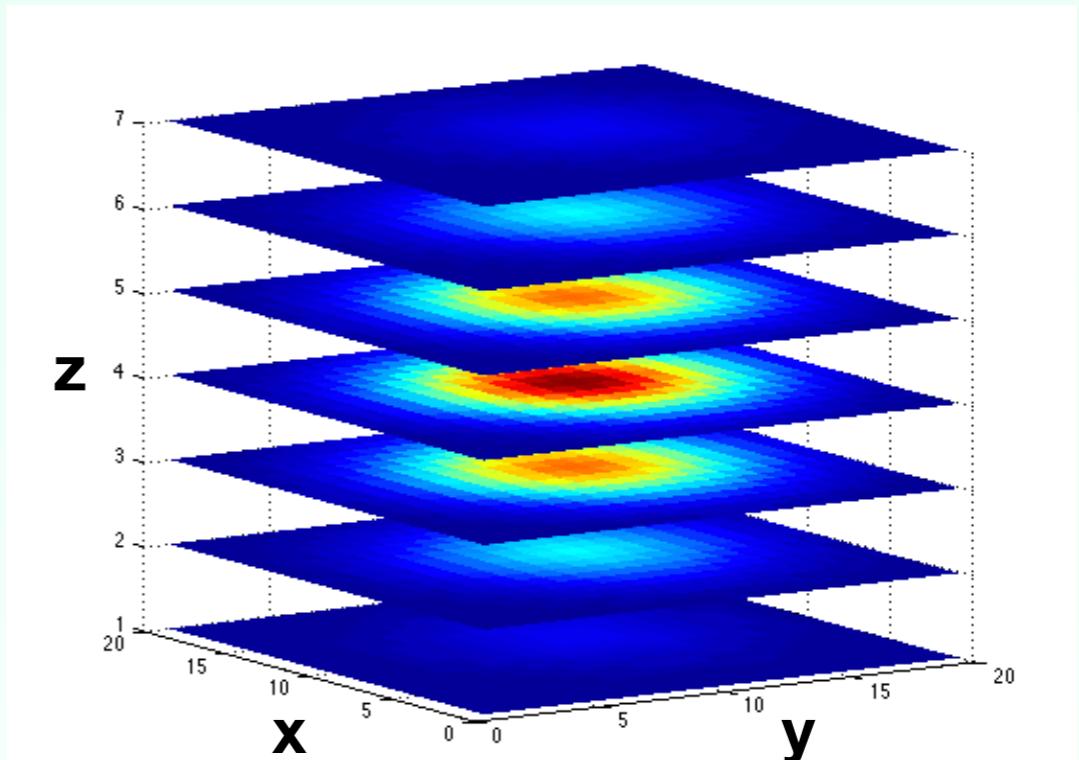
Source:

Multi-variate Gaussians.

2-D Gaussian



3-D Gaussian



$$f(x, y) = A \exp\left(-\left(\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2}\right)\right)$$

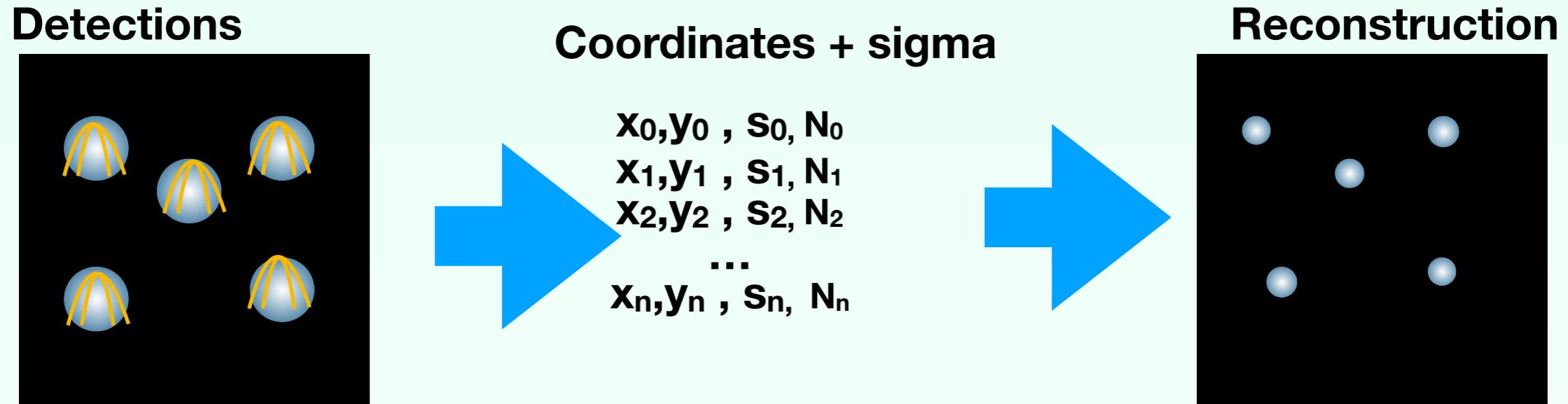
$$f(x, y, z) = A \exp\left(-\left(\frac{x - x_0}{2\sigma_x^2} + \left(\frac{y - y_0}{2\sigma_y^2}\right) + \left(\frac{z - z_0}{2\sigma_z^2}\right)\right)\right)$$

The same techniques for estimating the parameters generalise well to higher dimensions.

The 3-D Gaussian is often used to represent microscopy psf and so is very useful.

Source: <https://physics.stackexchange.com/questions/353414/how-does-a-free-electron-look-like>

Detection of foci, to localisation

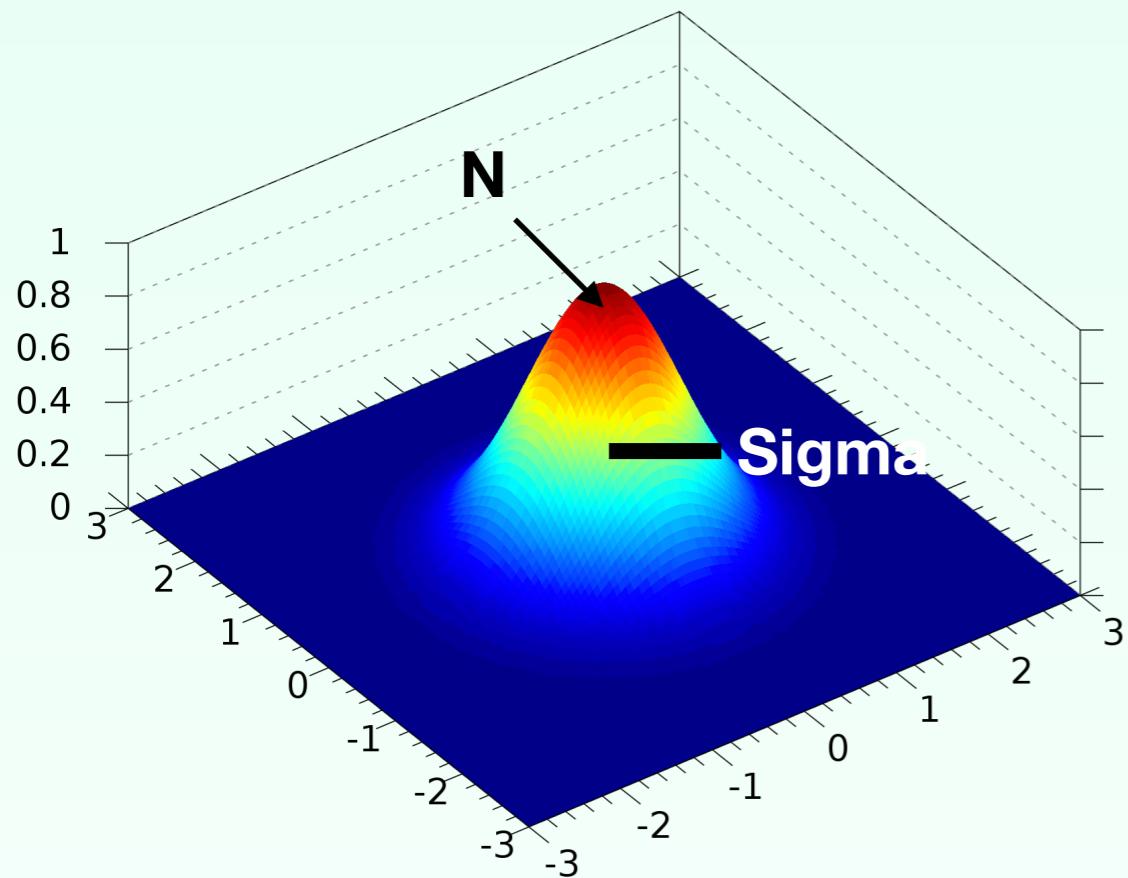


In STORM imaging the basic approach is to fit the fluorescent psfs with 2-D Gaussian distributions using non-linear least squares approach.

You'll see this in the practical.

Source:

Creating the output.



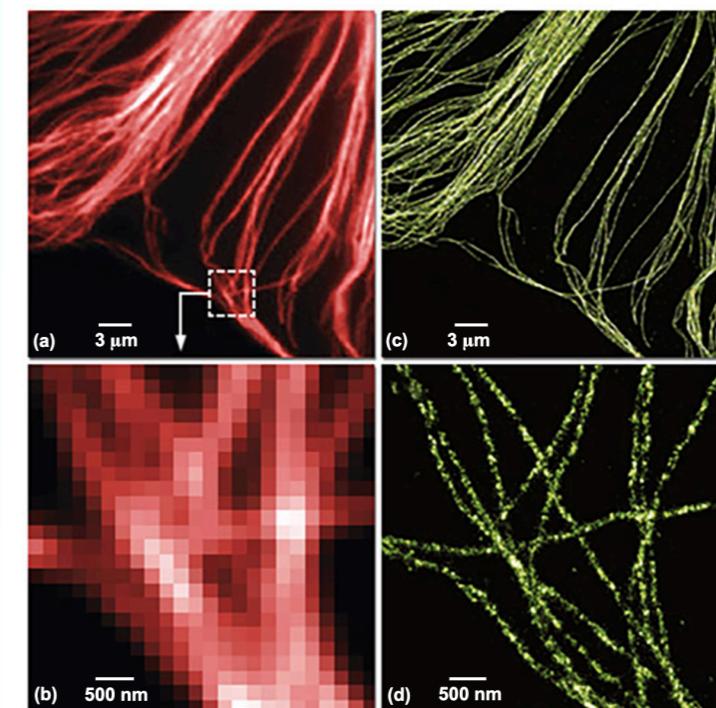
Photon Output

$$\sigma = \sqrt{\frac{s^2}{N} + \frac{(\frac{a^2}{12})}{N} + \frac{4\sqrt{\pi}s^3b^2}{aN^2}}$$

$$\sigma \approx \frac{s}{\sqrt{N}}$$

s is the s.d. of the fit Gaussian. N is the number of photons, \mathbf{a} is the pixel size of the imaging detector, and \mathbf{b} is the standard deviation of the background (including background fluorescence emission combined with detector noise).

The localisation precision of STORM is dependent on the number of photons and can be calculated through the above equation.

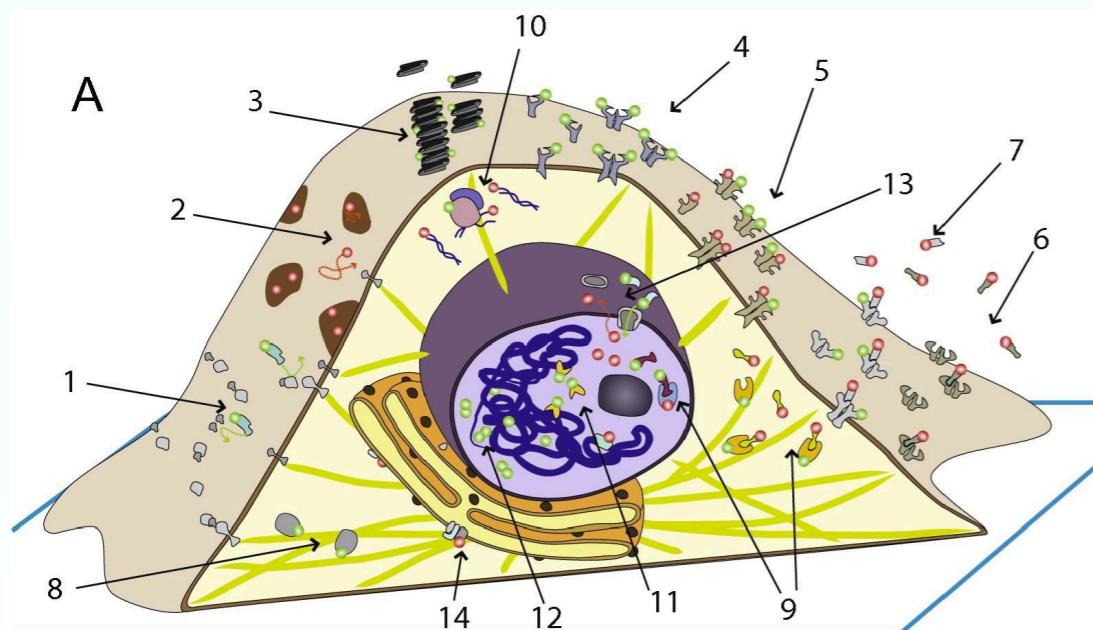
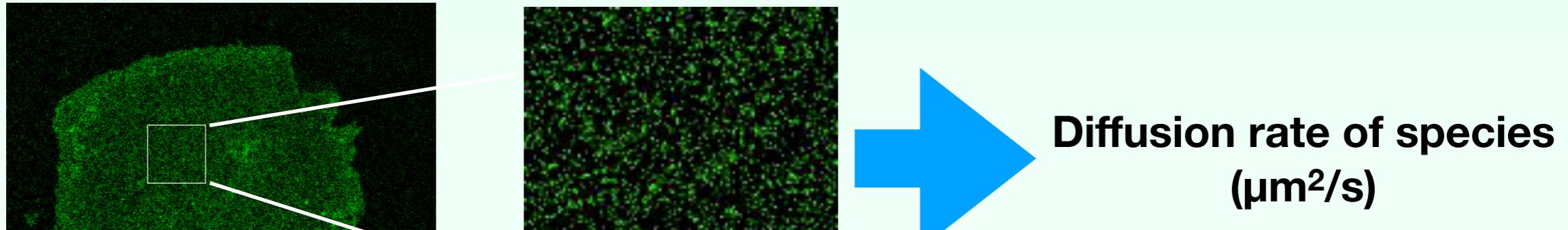


Source: <https://www.microscopyu.com/techniques/super-resolution/single-molecule-super-resolution-imaging>

Fluorescence Correlation Spectroscopy (FCS)

Why do we want to do FCS?

Can be used to quantify highly sensitive dynamic biological processes, often which are too subtle to be visually characterised.



E.g. In biology to elucidate:

- Structural properties of biomolecules
- Interactions between proteins and molecules
- Transport mechanisms.
- Development of Multicellular organisms

Source: Recent applications of fluorescence correlation spectroscopy in live systems
Wilhelm Radek Macháň Thorsten Wohland. Image. Jurkat T-cell eGFP-LCK, box is 7.68 μm

Fluorescence Correlation Spectroscopy

- FCS has been around since the 1970s and has evolved several times.
- These days, FCS is most commonly applied using a confocal microscope. With specialist detection hardware for single photon counting.
- Stimulated Emission Depletion (STED) FCS has been successfully applied to dynamically reduce the effective detection volume and to characterise diffusion characteristics across different length scales.

Key references: The mystery of membrane organization: composition, regulation and physiological relevance of lipid rafts **Erdinc Sezgin, Ilya Levental, Satyajit Mayor, and Christian Eggeling**.

Measuring nanoscale diffusion dynamics in cellular membranes with super-resolution STED–FCS **Erdinc Sezgin, Falk Schneider, Silvia Galiani, Iztok Urbančič, Dominic Waithe, B. Christoffer Lagerholm & Christian Eggeling**.

Cortical actin networks induce spatio-temporal confinement of phospholipids in the plasma membrane – a minimally invasive investigation by STED-FCS **Débora M. Andrade^{1,3,*}, Mathias P. Clausen^{1,2,4,*}, Jan Keller¹, Veronika Mueller¹, Congying Wu⁵, James E. Bear^{5,6}, Stefan W. Hell¹, B. Christoffer Lagerholm^{2,4} & Christian Eggeling¹**,

Source:<https://www.picoquant.com/applications/category/life-science/fluorescence-correlation-spectroscopy-fcs>

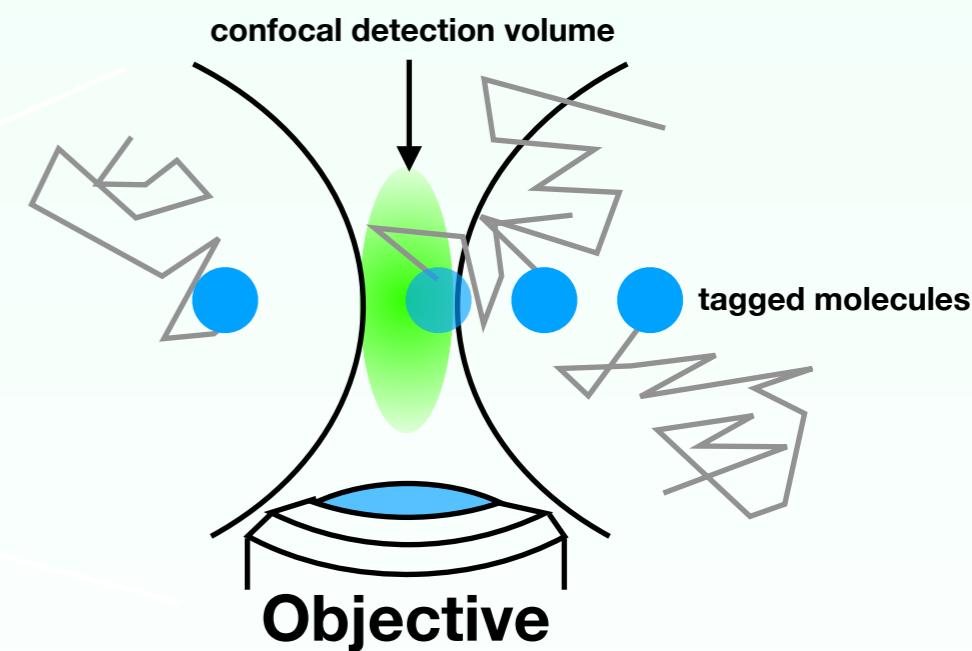
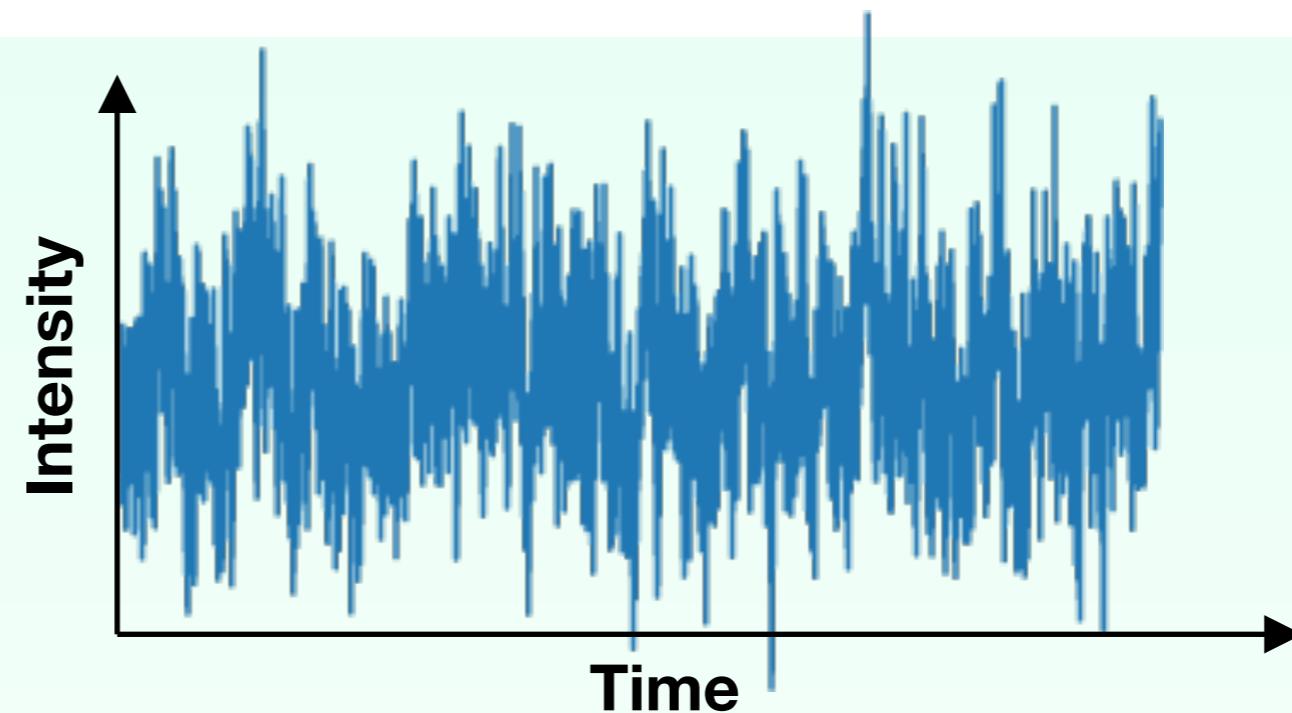
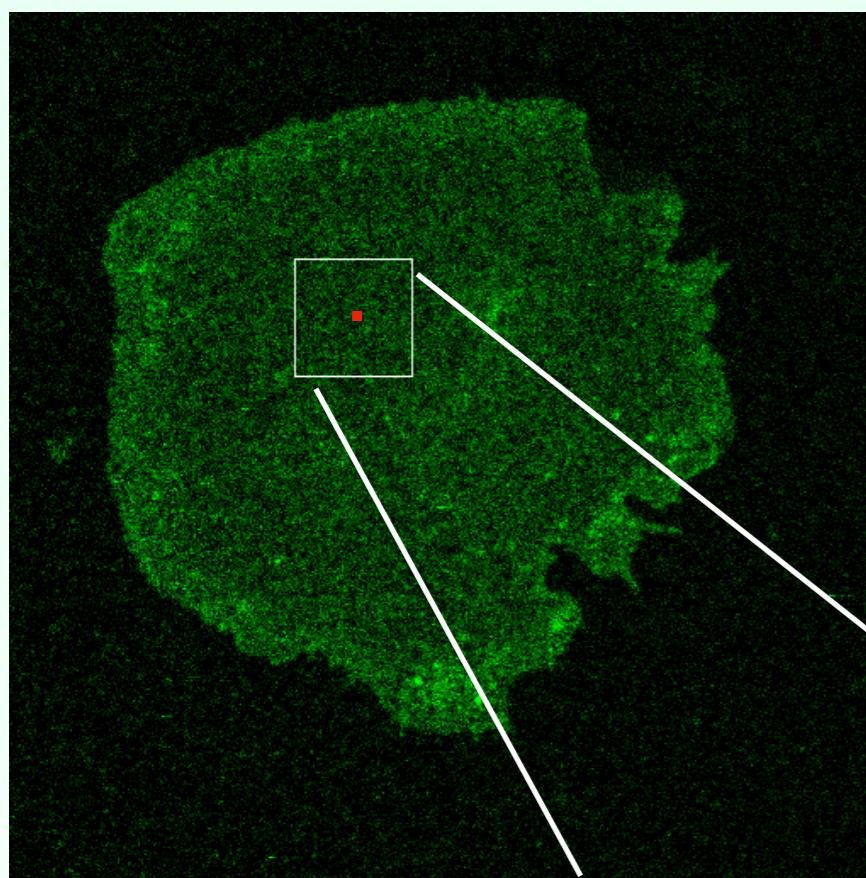
FCS: How it works

Overview

- Fluorescence intensity is measured at high-temporal frequency in a location/s of a sample.
- The intensity in a spot fluctuates with time because there is a flux of fluorescent particles (molecules) in and out of the observation volume.
- Correlation analysis yields strength and duration of the fluctuations.
- Using derived equations we calculate from the correlation function the average number of fluorescent particles in the detection volume and their average transit time through the volume.

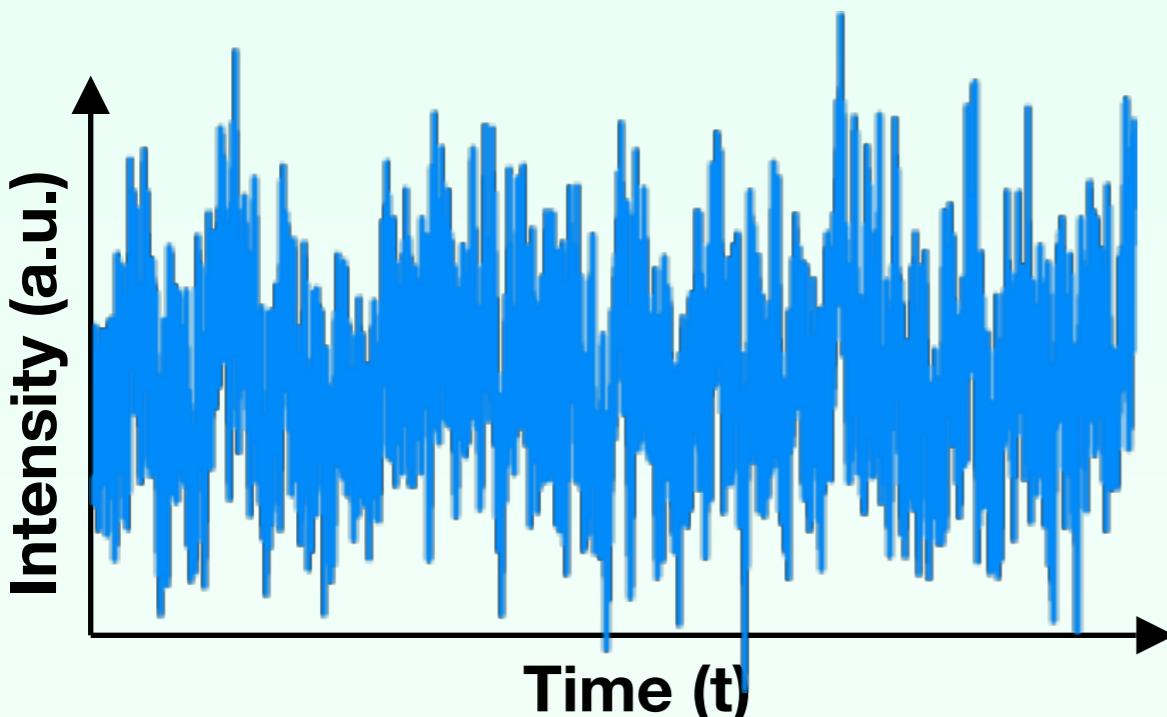
Source:<https://www.picoquant.com/applications/category/life-science/fluorescence-correlation-spectroscopy-fcs>

FCS: - Fluorescence intensity is measured at high-temporal frequency

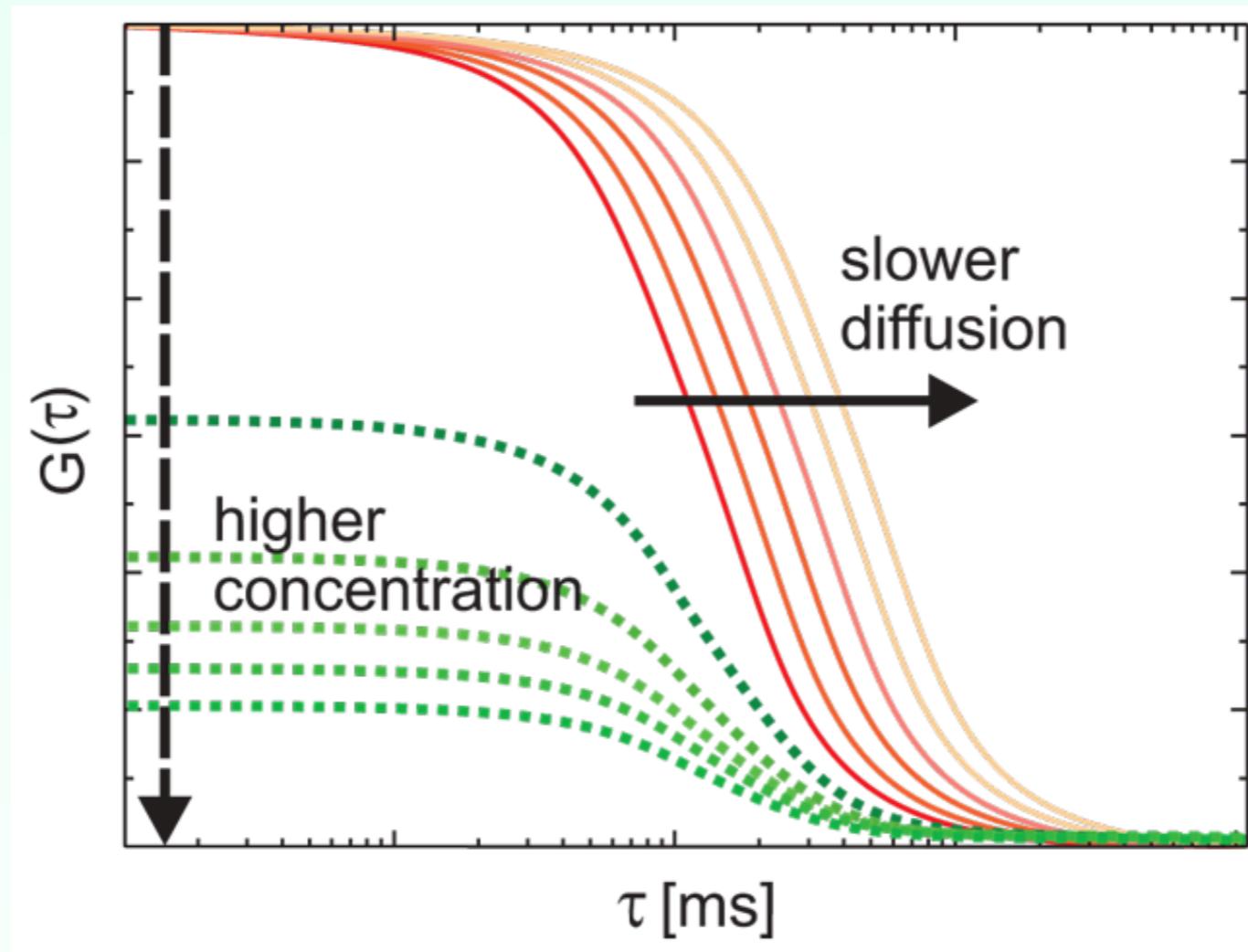


Source: Jurkat T-cell LCK-eGFP. box is 7.68 μm

FCS: Correlation analysis of fluctuations.



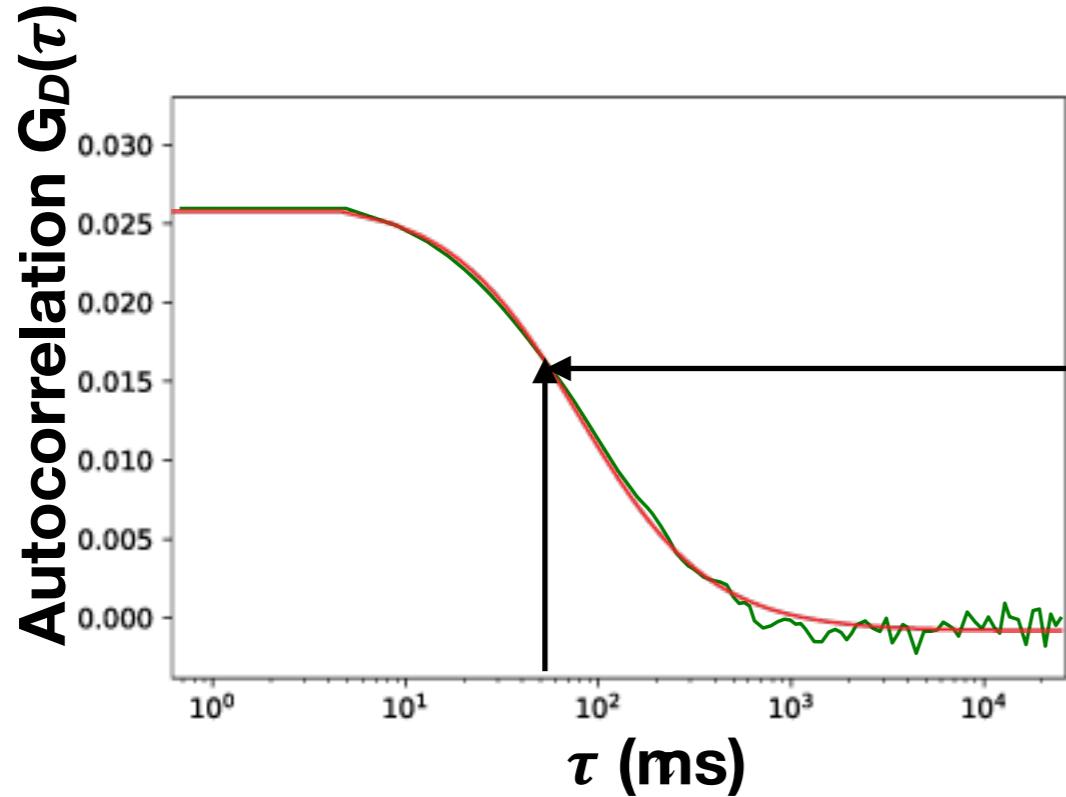
How to interpret a FCS curve.



We collect the transit time and calculate the diffusion time. We can also calculate how many tagged molecules there are present and other diffusion characteristics.

Source: <https://www.picoquant.com/applications/category/life-science/fluorescence-correlation-spectroscopy-fcs>

To Quantify our distribution we fit models



the characteristic decay time
i.e. Transit time τ_D

From this fit we can extract the transit time. This is the amount of time taken on average for a molecule to move across the detection volume.

for 2D-Diffusion (simple case).

$$G_D(\tau) = \frac{1}{N} \left(1 + \frac{\tau}{\tau_D} \right)^{-1}$$

Diffusion = $\omega_{xy}^2 / 4 \cdot \tau_{xyk}$
 $\omega_{xy} = FWHM / \sqrt{2 \cdot \ln(2)}$

Source:

Fitting is an art....there are many models

Diffusion type	Fitting function
3D diffusion	$G(\tau) = \frac{1}{N} \left(1 + \frac{\tau}{\tau_D}\right)^{-1} \frac{1}{\sqrt{1+w_0^2 \frac{\tau}{\tau_D}}}$
2D diffusion	$G(\tau) = \frac{1}{N} \left(1 + \frac{\tau}{\tau_D}\right)^{-1}$
2D diffusion for elliptical Gaussian profile	$G(\tau) = \frac{1}{N} \left(1 + \frac{\tau}{\tau_D}\right)^{-1/2} \frac{1}{\sqrt{1+\frac{\tau}{s^2 \tau_D}}}$
2D diffusion with triplet	$G_{Tr}(\tau) = G(\tau) \left[1 + T(1 - T)^{-1} \exp\left(\frac{-\tau}{\tau_{Tr}}\right) \right]$
2D diffusion with blinking	$G_B(\tau) = G(\tau) \left[1 + \frac{C_{dark}}{C_{bright}} e^{-k_{bl}\tau} \right]$
2D diffusion with two-component	$G_{2C}(\tau) = \frac{1}{N_{tot}} \frac{q_1^2 Y_1 G_1(\tau) + q_2^2 Y_2 G_2(\tau)}{q_1 Y_1 + q_2 Y_2}$

In general, like any fitting operation:

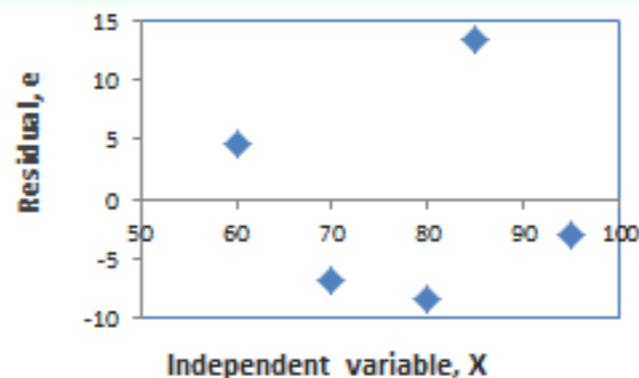
- Use fewest parameters possible.
- Check residuals for systematic failures.

Source: Falk Schneider. Fluorescence Techniques to Study Lipid Dynamics Erdinc Sezgin and Petra Schwille

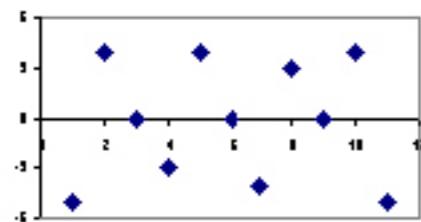
A note about residuals

$$\text{Residual} = \text{Observed value} - \text{Predicted value}$$
$$e = y - \hat{y}$$

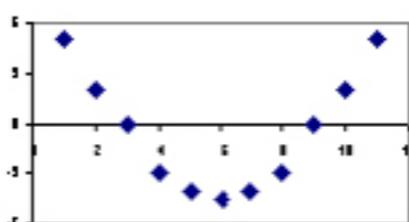
- A residual plot, shows this error for each datapoint x.



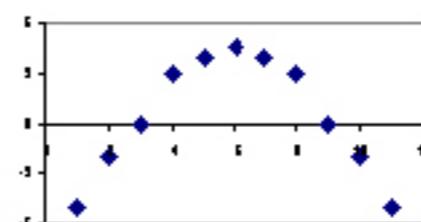
Residuals measure the error between our data and our predicted line.



Random pattern



Non-random: U-shaped

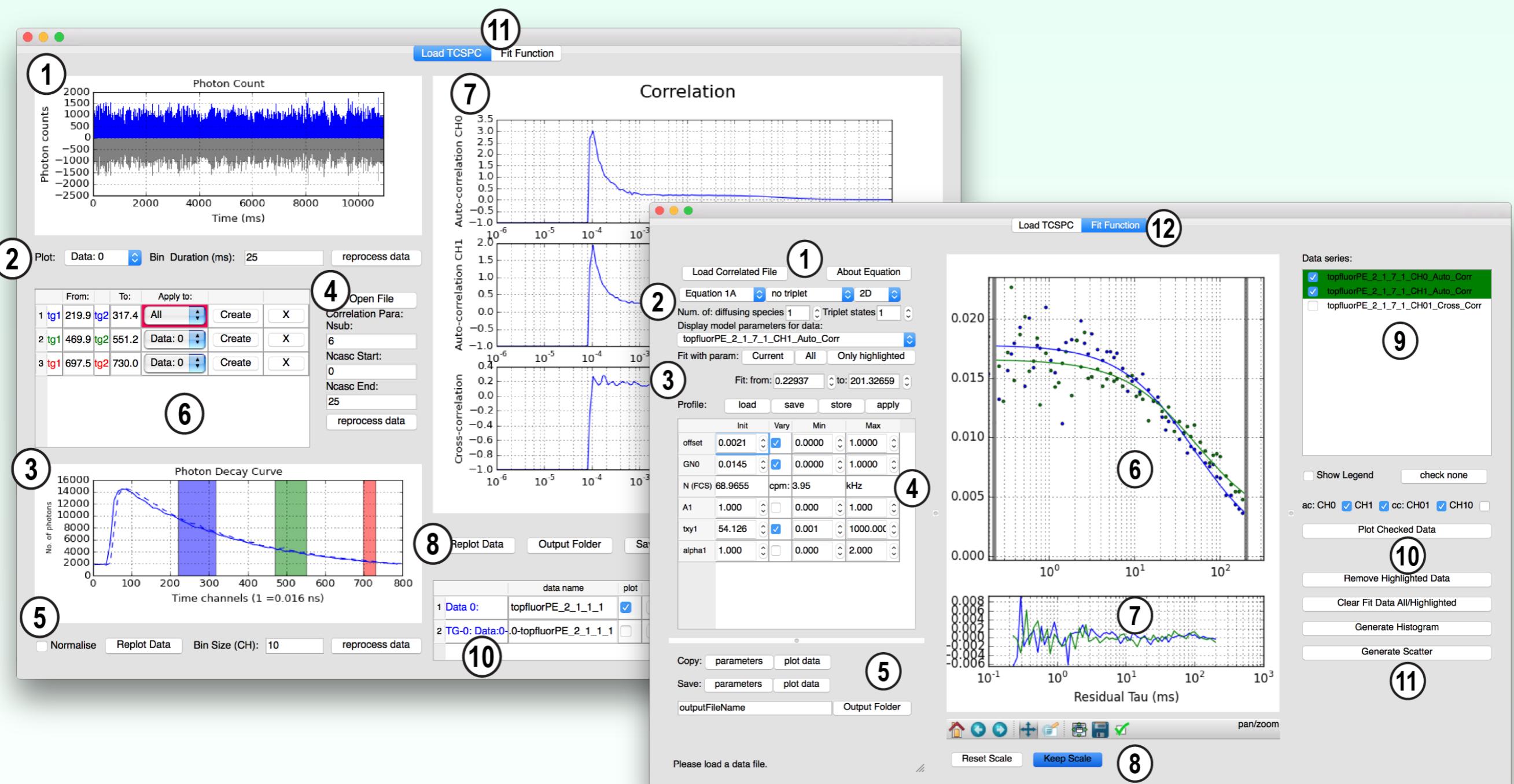


Non-random: Inverted U

If your data, doesn't look randomly distributed, rethink your model.

Source:

FoCuS-point



FoCuS-point software for analysing point FCS TCSPC data.

Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5939892/> FoCuS-point: software for STED fluorescence correlation and time-gated single photon counting
Dominic Waithe, Mathias P. Clausen, Erdinc Sezgin, and Christian Eggeling.

Thanks for your time.

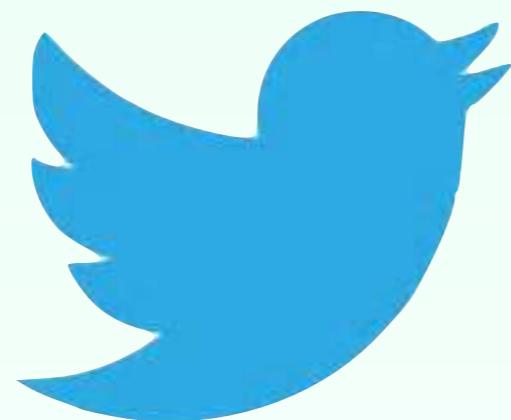
For these slides and more:

<https://github.com/IAFIG-RMS/Bioimage-training>

<https://twitter.com/dwaithe>



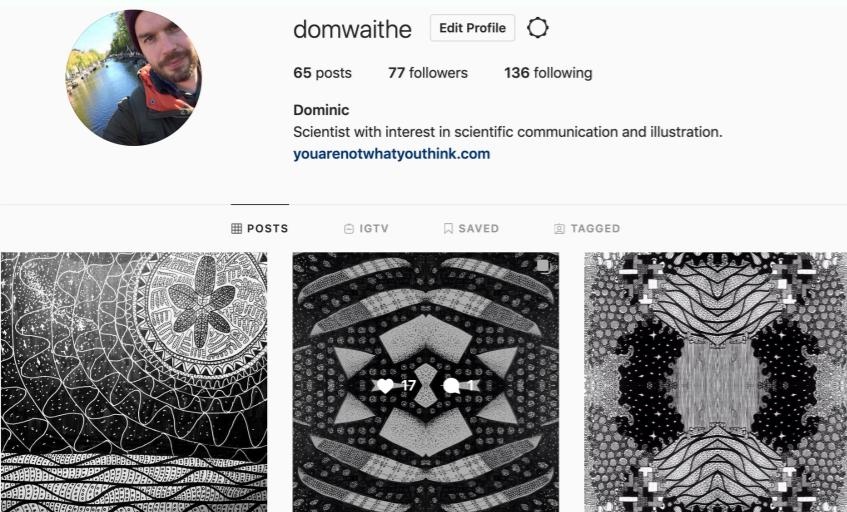
A screenshot of a Twitter profile for 'Dominic Waithe' (@dwaithe). The profile picture shows a man with a beard. The bio reads: 'Scientist with interest in scientific communication and illustration. youarenotwhatyouthink.com'. Statistics below show 401 tweets, 157 following, and 196 followers.



<https://github.com/dwaithe>



<https://instagram.com/dwaithe>



A screenshot of an Instagram profile for 'domwaithe'. The bio reads: 'Scientist with interest in scientific communication and illustration. youarenotwhatyouthink.com'. Statistics show 65 posts, 77 followers, and 136 following. Below the profile are three sample images showing intricate black-and-white micrographs or scientific illustrations.

dominic.waithe@imm.ox.ac.uk



UK Research
and Innovation



Sources