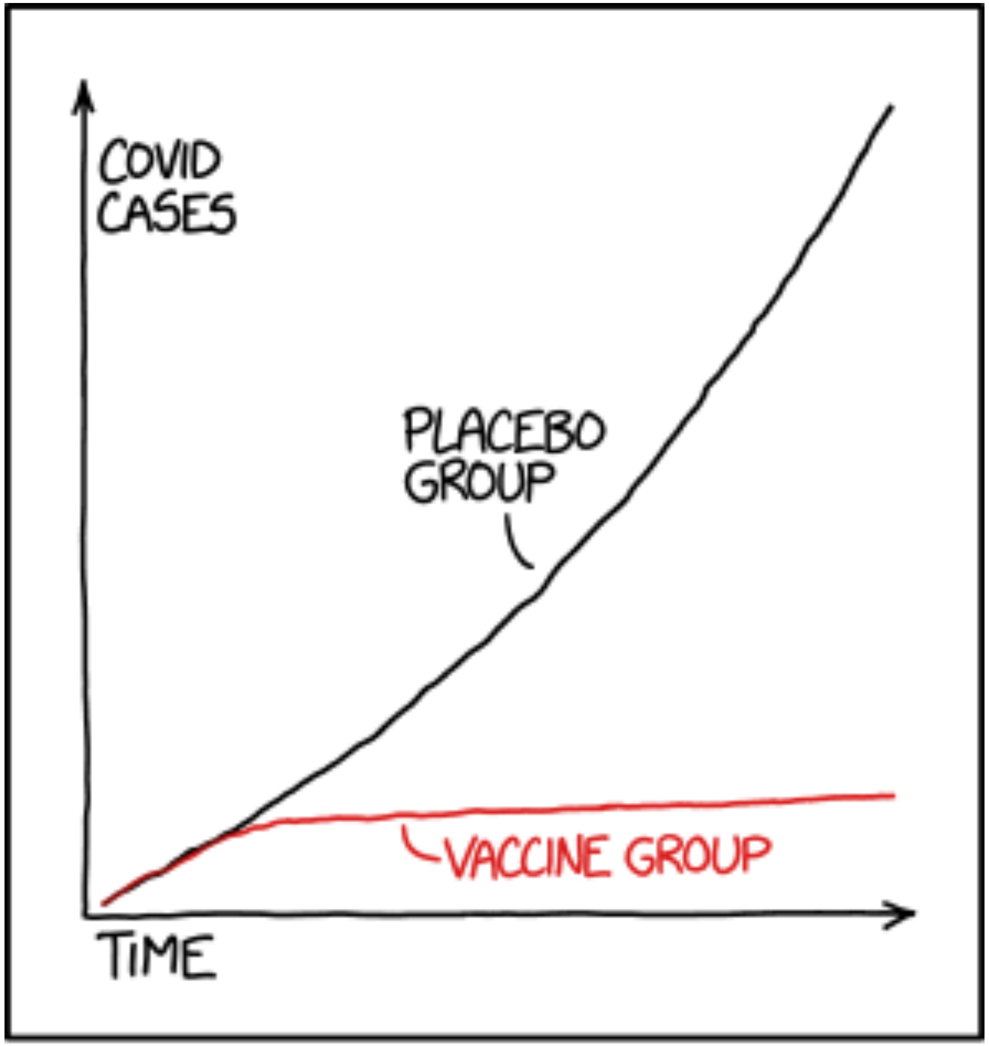
Errors, Fits & MCMC - a practitioner's guide

"An attempt to be practical and efficient"



STATISTICS TIP: ALWAYS TRY TO GET DATA THAT'S GOOD ENOUGH THAT YOU DON'T NEED TO DO STATISTICS ON IT

Corona-Testing

Assume:

- A Corona-test with 99% accuracy and reliability
- You get a positive result
- How high is the chance that you are positive?

Answer: You don't know. Prior information is missing

Example: 1000000 people

incidence rate: 10^{-4}

$\Rightarrow$ 100 positive cases, 99 tested positive

$\Rightarrow$ 999900 negative cases,
989901 tested negative,
9999 tested positive

$\Rightarrow$ p = 99/(99 + 9999) = 0.98%

incidence rate: 10^{-2}

$\Rightarrow$ 10000 positive cases, 9900 tested positive

$\Rightarrow$ 990000 negative cases,
980100 tested negative,
9900 tested positive

$\Rightarrow$ p = 9900/(9900 + 9900) = 50%

Bayes' theorem

$$p(A|B) = \frac{p(B|A)\, p(A)}{p(B)}$$

0.99

10^{-4}

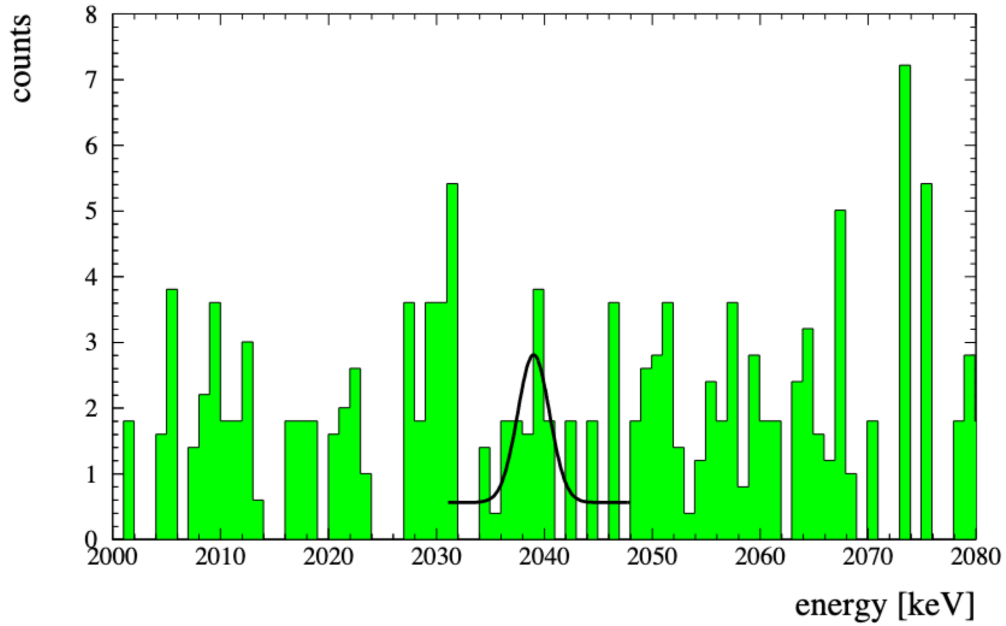
$$p(\text{act pos}|\text{test pos}) = \frac{\mathrm{p}(\text{test pos}|\text{act pos})\, \mathrm{p}(\text{act pos})}{\mathrm{p}(\text{test pos})}$$

$(9999 + 99)/10^6$

Often interpreted as:

- p(A) is our prior knowledge
- new data become available (test, B)
- p(A|B) is our updated knowledge

A ”danger” with (Bayesian) priors



Klapdor-Kleingrothaus et al. 2001

neutrino-less double beta-decay

prior: line position known

.. significance of detection
is around 3 σ ..

Is that believable?

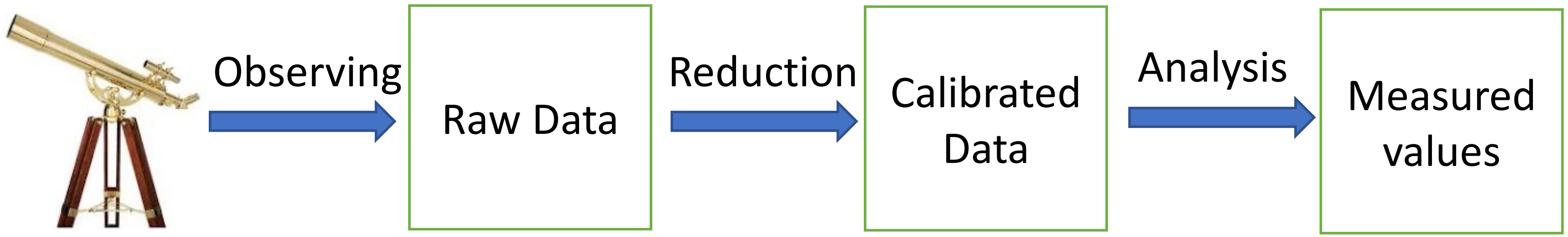
The next generation follow-up experiment GERDA did not find any signal

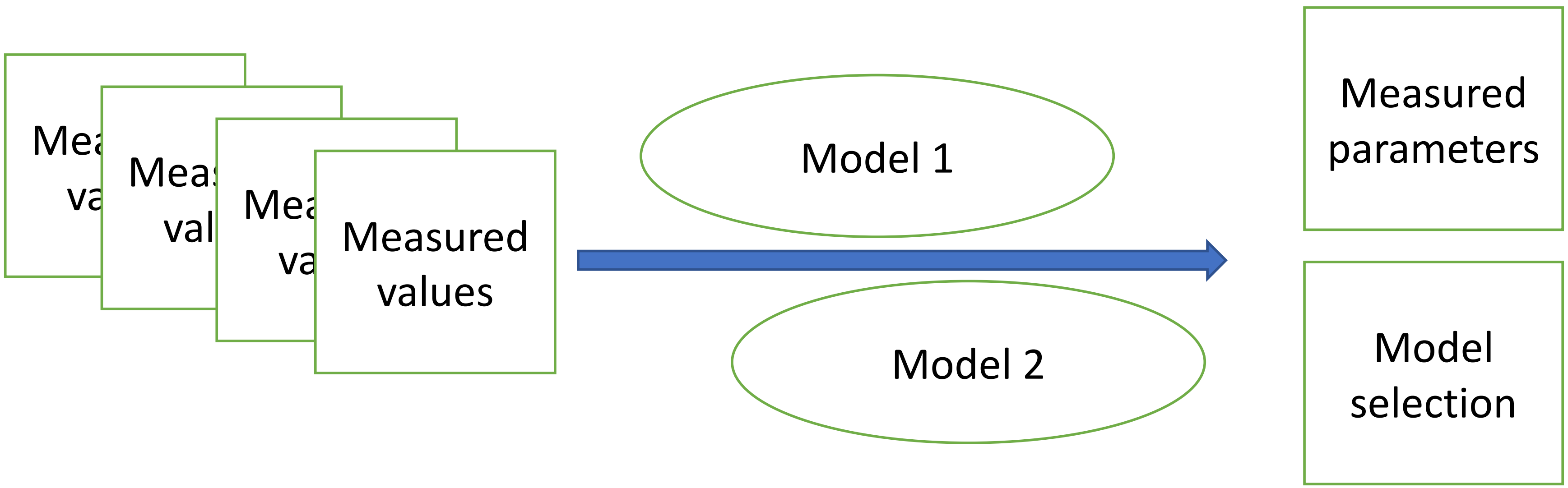
STATISTICS TIP: ALWAYS TRY TO GET DATA THAT'S GOOD ENOUGH THAT YOU DON'T NEED TO DO STATISTICS ON IT

Content

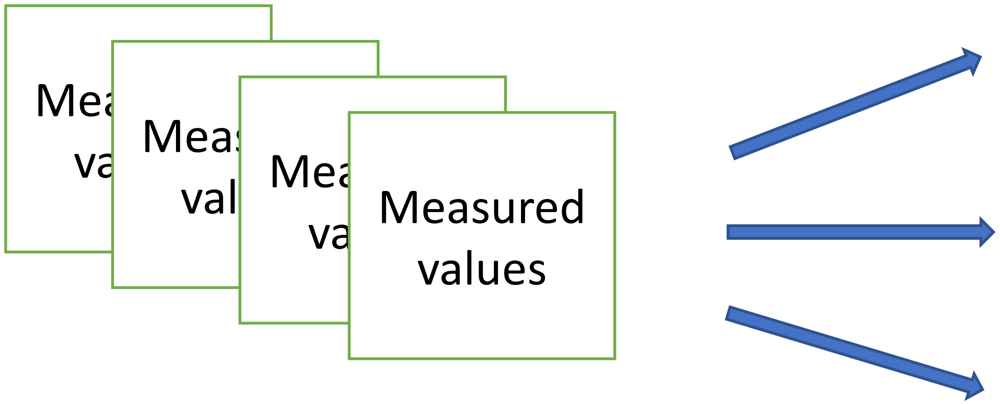
- Basics, error reporting, error propagation
- χ^2, Fitting
- Confidence intervals, covariance matrix
- Goodness of fit, Comparing fits
- Difficulties
- Jack-knife, Bootstrapping
- MCMC
- Literature

A typical chain towards a scientific result





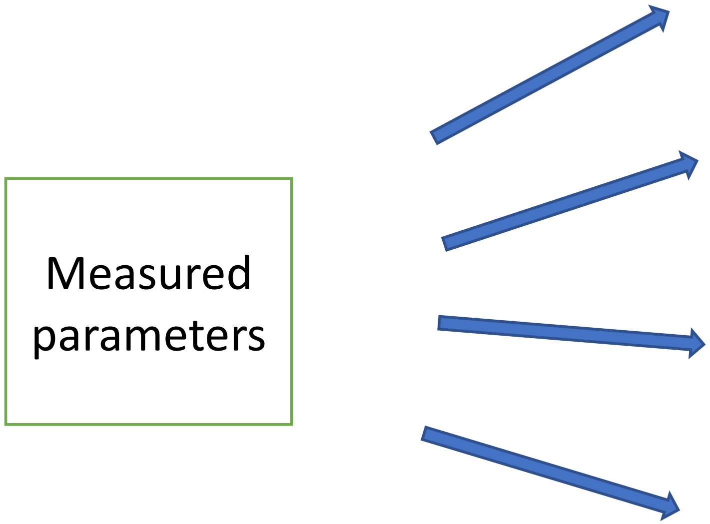
Errors that occur



- **Statistical error:**
  The effect of noise on the data
- **Systematic error:**
  The effect of mis-calibration
- **Mistakes**

- Statistical errors are “straight forward”, sometimes part of pipelines
  - a matter of propagating correctly
- Systematic errors are your job: Physicist’s intuition needed
  - knowledge outside of the current measurement needed to understand how wrong the ruler might be
- It is your right and obligation to check for mistakes (i.e. obvious outliers), and select your data accordingly

Errors of the result



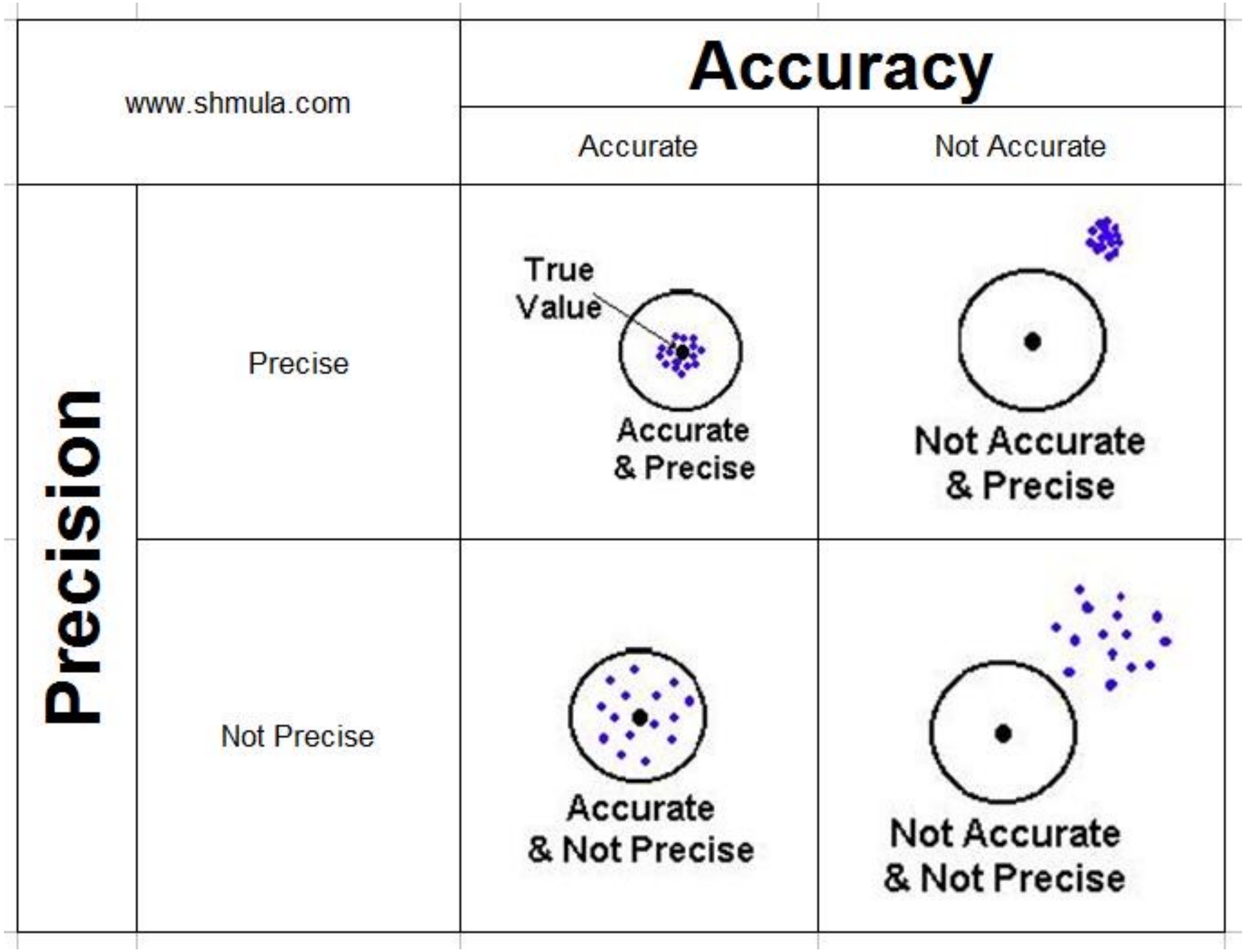
- **Statistical error:**
  The effect of noise on the data
- **Systematic error:**
  The effect of mis-calibration
- **Sampling error:**
  The result might depend on what of your data you use
- **Model error:**
  Your model most likely is a simplified version of reality

What error to report

- Good practice: $R = 100 \pm 5|_{\mathrm{stat}} \pm 8|_{\mathrm{sys}}$

  - Systematics don't necessarily average out, so report them seperately from statistical error
  - Like this, you say
    - how precise the result is (statistical error); and
    - how accurate the result is (systematic error)
  - The sampling error usually can be included in the statistical one
  - The model error usually is part of the systematic one

Precision and Accuracy



A little riddle on averaging

$$v = (v_1 + v_2)/2$$

$$\Delta v = \sqrt{\Delta v_1^2 + \Delta v_2^2}\,/2$$

v_1 = 100 ± 10
v_2 = 100 ± 18

→ v = 100 ± 10.3

Weighted mean !

$$w_i = 1/\Delta v_i^2$$

$$v = \frac{\Sigma w_i v_i}{\Sigma w_i}$$

$$w = \Sigma w_i$$

$$\Delta v = 1/\sqrt{w}$$

v_1 = 100 ± 10
v_2 = 100 ± 18

→ v = 100 ± 8.7

Averaging samples

$$u = \Sigma w_i^2$$

$$s^2 = \frac{w}{w^2 - u} \Sigma w_i (v_i - v)^2$$

100 ± 2

110 ± 1

± 0.6

(weighted)

103.0 ± 8.1

eights the above

$\frac{1}{\;\;- 1} \Sigma (v_i - v)^2$

The authors of paper X show a table of result numbers coming from different ways to analyse their data.

The combined uncertainty should be the average uncertainty.

The authors of paper X show a table of result numbers coming from different galaxies.

The combined uncertainty should be the average uncertainty divided by $\sqrt{n}$.

What ... : *s* or $s/\sqrt{n}$

Combined

- *s* estimates the uncertainty of a si...
  - if your sample is n times reading ...
  - if your sample is from splitting up ...

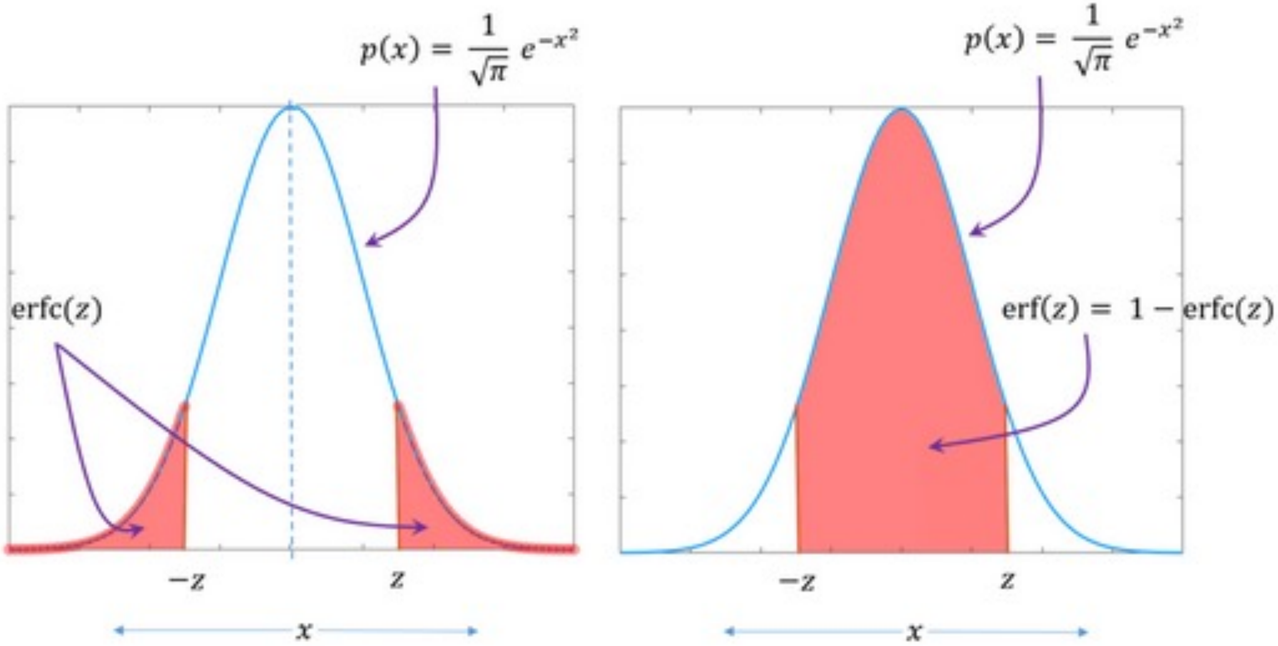
$\sqrt{\Delta v^2 + s^2}$

- $s/\sqrt{n}$ estimates the uncertainty of the mean
  - if your sample is n measurements: use $s/\sqrt{n}$

$\sqrt{\Delta v^2 + s^2/n}$

Gaussians & probabilities

$$\int_{-s}^{s} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} = \mathrm{erf}\,\frac{\mathrm{s}}{\sqrt{2}\sigma}$$



s = 1 σ: p = 68.27%

For an expected value of s=0, and an error of σ,
finding s < 0.67449 σ is **less** likely than finding s > 0.67449 σ.

A measurement of the type
x = 0.012 ± 0.250 is rather unlikely with a chance of occuring of only 3.8%.

Error propagation

$$v = f(v_i)$$

$$\Delta v^2 = \Sigma \left| \frac{\partial f}{\partial v_i} \right|^2 \Delta v_i^2$$

$$v = v_1 + v_2$$
$$\Delta v^2 = \Delta v_1^2 + \Delta v_2^2$$

$$v = v_1 - v_2$$
$$\Delta v^2 = \Delta v_1^2 + \Delta v_2^2$$

$$v = v_1 \times v_2$$
$$\left(\frac{\Delta v}{v}\right)^2 = \left(\frac{\Delta v_1}{v_1}\right)^2 + \left(\frac{\Delta v_2}{v_2}\right)^2$$

$$v = v_1 / v_2$$
$$\left(\frac{\Delta v}{v}\right)^2 = \left(\frac{\Delta v_1}{v_1}\right)^2 + \left(\frac{\Delta v_2}{v_2}\right)^2$$

$$v = v_1^\alpha \times v_2^\beta$$
$$\left(\frac{\Delta v}{v}\right)^2 = \left(\alpha \frac{\Delta v_1}{v_1}\right)^2 + \left(\beta \frac{\Delta v_2}{v_2}\right)^2$$

The χ^2

Data: $(x_i,\, y_i)$ Model: $f(x, p_j)$

An individual data point is $\sigma_i = \dfrac{y_i - f(x_i,\, p_j)}{\Delta y_i}$ away from the model.

How bad is a model?

- The more larger σ_i occur, the worse the model is.
- The badness should be a monotonic function of $|\sigma_i|$
- All points should be treated equal
- Possible choices: $\Sigma|\sigma_i|$ or $\Pi|\sigma_i|$ or $\Sigma\sigma_i^2 = \chi^2$

$$\chi^2 = \Sigma \frac{(y_i - f(x_i,\, p_j))^2}{\Delta y_i^2}$$

The χ^2 is a maximum likelihood estimator for normally distributed data

Probability for a given measurement: $p_i = e^{-\frac{1}{2}\left(\frac{y_i - f(x_i, p_j)}{\Delta y_i}\right)^2}$

Probability for all measurements: $L = \Pi \, p_i$

Maximizing L
= maximizing ln L
= minimizing –ln L

$$\ln L = \sum -\frac{1}{2}\left(\frac{y_i - f(x_i, p_j)}{\Delta y_i}\right)^2$$

$$\chi^2 = \Sigma \frac{(y_i - f(x_i,\, p_j))^2}{\Delta y_i^2}$$

$$L = e^{-\frac{1}{2}\chi^2}$$

Central limit theorem: Why Gaussians are so important

Taylor expansion around maximum of likelyhood:

$$\log \mathcal{L}(\vec{\theta}) \approx \log \mathcal{L}(\vec{\theta}_{\mathrm{max}}) + \frac{1}{2} \left. \frac{\partial^2 \log \mathcal{L}}{\partial \theta_i \partial \theta_j} \right|_{\vec{\theta}_{\mathrm{max}}} (\theta - \theta_{\mathrm{max}})_i (\theta - \theta_{\mathrm{max}})_j$$

Note: No linear terms

$$\mathcal{L}(\vec{\theta}) = e^{\log \mathcal{L}(\vec{\theta})} \text{ is a Gaussian}$$

- ”locally enough, it behaves like that”
- large N makes most distributions over a larger range roughly Gaussian

Fitting = Asking what is least bad model

.. often the language is: what is “the best fit” ..

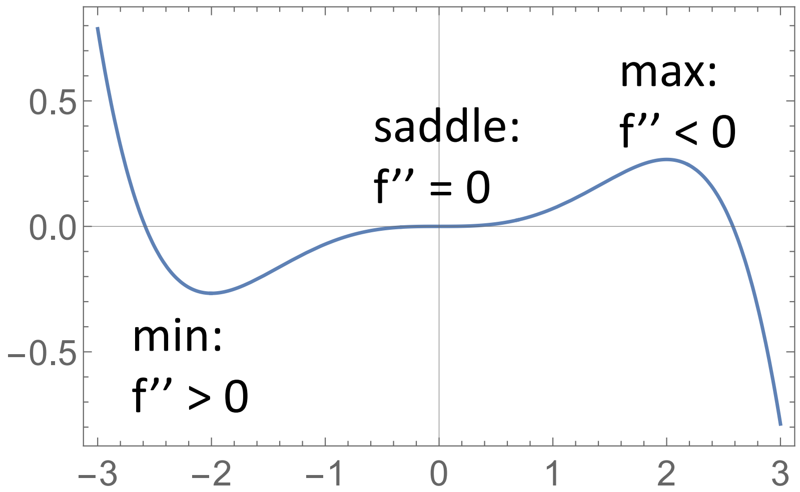
$$min|_{\{p_j\}}\, \chi^2(p_j)$$

Thus: N-dimensional minimization problem

If you are at a minimum, you have:

$$\frac{\partial \chi^2(p_j)}{\partial p_j} = 0$$

$$\frac{\partial^2 \chi^2(p_j)}{\partial p_j^2} > 0$$



How to find the minimum?

.. in an N dimensional space ..

Local minimization

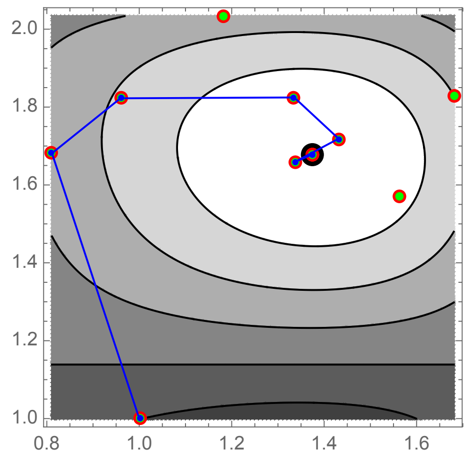
- start at some point
- understand locally the landscape
- estimate where to move to find smaller value
- repeat, until minimum conditions are fulfilled (to a certain numerical accuracy)

- Guaranteed to work
- Example: Newton's method of steepest gradient
- Many, many methods, trying to be efficient in the number of χ^2 evaluations needed

Global minimization

- sample parameter space globally
- run local minimizations
- take the best

- Not guaranteed to work (!)



If you want to fit, you must be able to calculate 1000's of χ^2 in reasonable time

In case you need to do it yourself…

$$\nabla_j = \frac{\chi^2(p_j + \epsilon_j) - \chi^2(p_j - \epsilon_j)}{2\epsilon_j}$$

$${}^{n+1}p_j = {}^{n}p_j - \nabla_j \times \mathrm{step}_j$$

where the difficulty is knowing a suitable N-dimensional ϵ_j and step size

but most likely you have a minimzer…

- Even "Newton" uses also 2nd order derivatives
- Quasi-Newton: be efficient by re-using already calculated χ^2
- Levenberg-Marquardt: for problems which are sums of squares, the 2nd order derivative can be estimated by 1st order derivatives
- Nonlinear conjugate gradient: Clever ways of the above
- Principal axis methods: Not using any derivatives

Linear models

If the model f is linear in the parameters p_j , $\frac{\partial \chi^2(p_j)}{\partial p_j} = 0$ is a linear equation

→ N linear equations for N parameters

→ Matrix inversion

(Simple, yet frequent and useful) example: f(x; a,b) = a + b x

$$\chi^2(a,b) = \sum_{i=1}^{N} \left(\frac{y_i - a - b x_i}{\sigma_i} \right)^2$$

$$0 = \frac{\partial \chi^2}{\partial a} = -2 \sum_{i=1}^{N} \frac{y_i - a - b x_i}{\sigma_i^2}$$

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

$$S \equiv \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \quad S_x \equiv \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \quad S_y \equiv \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2}$$

$$S_{xx} \equiv \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} \quad S_{xy} \equiv \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2}$$

→

$$aS + bS_x = S_y$$
$$aS_x + bS_{xx} = S_{xy}$$

→ $U.\binom{a}{b} = v$ → $\binom{a}{b} = U^{-1}.v$

$$U = \begin{pmatrix} S & S_x \\ S_x & S_{xx} \end{pmatrix}, \; v = \begin{pmatrix} S_y \\ S_{xy} \end{pmatrix}$$

Linear models, Polynomials

data set: $[(x_1, y_1 \pm \sigma_1), (x_2, y_2 \pm \sigma_2), \ldots (x_N, y_N \pm \sigma_N)]$

model: $y = a_0 + a_1 x + a_2 x^2 \ldots + a_m x^m = \sum_{i=0}^{m} a_i x^i$

$$U_{\alpha\beta} = \sum_{i=1}^{N} \frac{x_i^{\alpha+\beta}}{\sigma_i^2}, \qquad v_\alpha = \sum_{i=1}^{N} \frac{y_i\, x_i^\alpha}{\sigma_i^2}.$$

where the α, β run from 0 to m

The parameters and their uncertainties are then:

$$a_\alpha = \sum_{\beta=0}^{m} \left(U^{-1}\right)_{\alpha\beta} v_\beta, \qquad \sigma_\alpha^2 = \left(U^{-1}\right)_{\alpha\alpha},$$

note: σ_α are independent of y_i

Some shortcuts

- For n evenly sampled data points to which a line is fitted, for large n, the errors will be:

  offset: $\frac{2}{\sqrt{n}}$; slope: $\sqrt{\frac{12}{n^3}}$

- Need to fit a Gaussian?
  Take the log of your data and fit a parabola!

  Since the parabola fit is a solved via a matrix inversion, no iterations are needed.
  This is thus stable and fast. Well-suited for any real-time computing

Fitting line to data with errors in x and y

$$\chi^2(a,b) = \sum_{i=1}^{N} \frac{(y_i - a\,x_i - b)^2}{\sigma_{y,i}^2 + a^2\,\sigma_{x,i}^2}$$

Fitting ellipse to data (without errors)

(2-dim) ellipse: $e(\vec{x};\, a, b, \vec{c}, \theta) = (\vec{x} - \vec{c})\,.\,R_\theta\,.\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}.\,R_\theta^T\,.\,(\vec{x} - \vec{c}) = 1$

$$\chi^2 = \sum (e(\vec{x_i};\, a, b, \vec{c}, \theta) - 1)^2$$

Fitting circle to data with errors

use N nuisance parameters t_i :

$x(t) = r\cos(t) + x_0$

$y(t) = r\sin(t) + y_0$

$$\chi^2 = \sum \frac{(x_i - x(t_i))^2}{\Delta x_i^2} + \sum \frac{(y_i - y(t_i))^2}{\Delta y_i^2}$$

and minimze for (r, x_0, y_0, t_1, t_2, …)

Almost always, the question is not only:

- what are the best fit parameters, but also

- how well are they constrained

How to find confidence levels?

$$68.3\% = \mathrm{erf}\left(\frac{1}{\sqrt{2}}\right) = \int_{\mu-\sigma}^{\mu+\sigma} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} dx = \int_{\mu-\sigma}^{\mu+\sigma} p(x;\,\mu,\sigma)dx$$

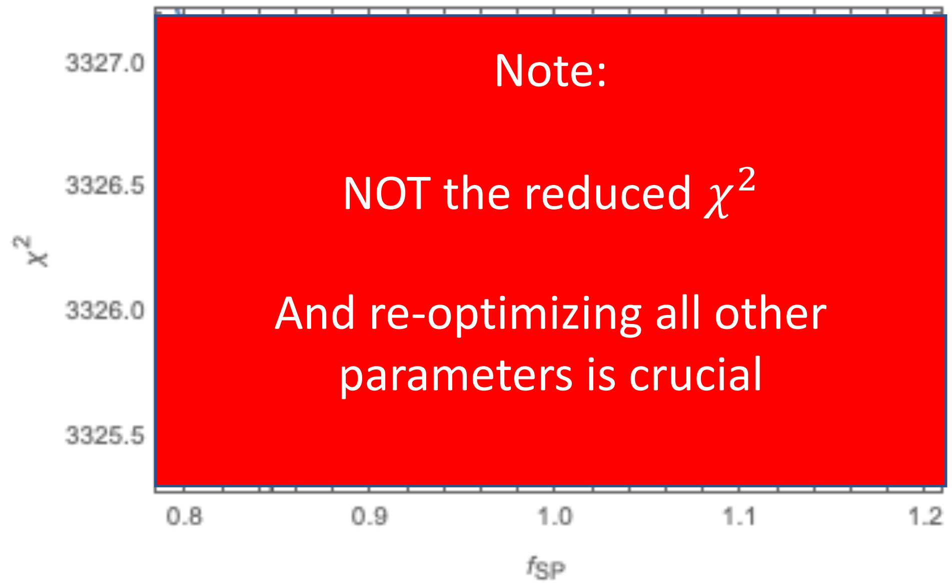
$$p(\mu;\,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}$$

$$p(\mu+\sigma;\,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}}$$

$$L(\mu+\sigma) = L(\mu)e^{-\frac{1}{2}} \rightarrow \chi^2|_\sigma = \chi^2_0 + 1$$

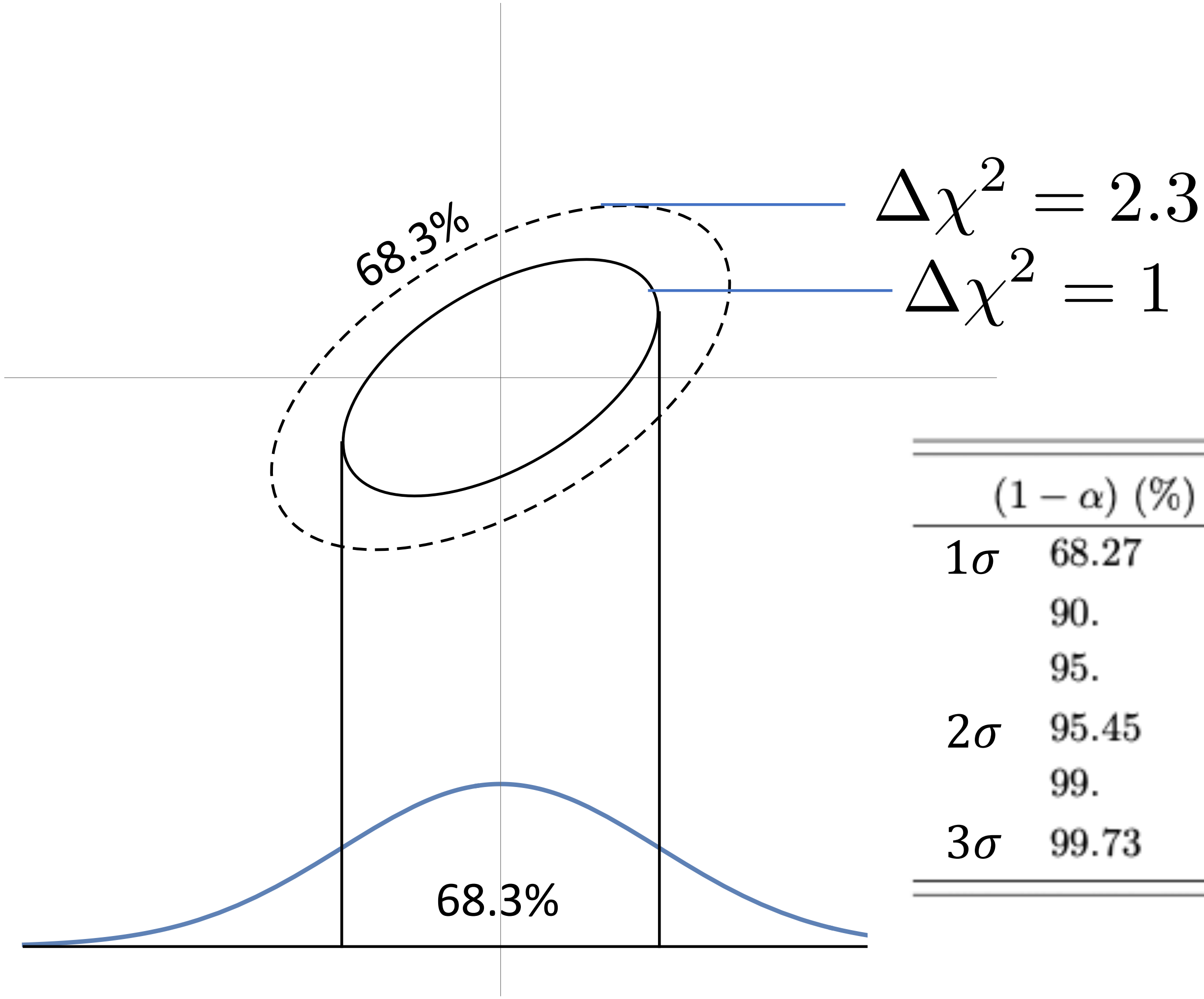
"Find the value of σ such that at $\mu + \sigma$ the χ^2 is larger by 1, **when re-optimizing all other parameters**"

	$(1-\alpha)$ (%)	$m = 1$
1σ	68.27	1.00
	90.	2.71
	95.	3.84
2σ	95.45	4.00
	99.	6.63
3σ	99.73	9.00



This recipe gets impractical for larger dimensions

For Gaussian posterior distributions: Confidence levels are N-dim ellipsoids



	$(1-\alpha)$ (%)	$m = 1$	$m = 2$	$m = 3$
1σ	68.27	1.00	2.30	3.53
	90.	2.71	4.61	6.25
	95.	3.84	5.99	7.82
2σ	95.45	4.00	6.18	8.03
	99.	6.63	9.21	11.34
3σ	99.73	9.00	11.83	14.16

More practical: Using the covariance matrix

$$\chi_0^2 := \chi^2(\{p_0^j\})$$

$$M_{jk} = \frac{\partial^2 \chi^2}{\partial p^j \, \partial p^k}\bigg|_{\{p_0^j\}}$$

$$C_{jk} = (\frac{1}{2} M_{jk})^{-1}$$ "covariance matrix"

$$\Delta p^j = \sqrt{C_{jj}}$$ errors are on the diagonal

Notes:

- M_{jk} and C_{jk} are symmetric and positive definite, so one can invert them
- Parameter correlations are taken into account by the process of inversion

In case you need to do it yourself…

$$M_{jj} = \frac{\chi^2(p_0^j + \epsilon_j) + \chi^2(p_0^j - \epsilon_j) - 2\chi_0^2}{\epsilon_j^2}$$

$$M_{jk} = \frac{\chi^2(p_0^j + \epsilon_j + \epsilon_k) - \chi^2(p_0^j + \epsilon_j - \epsilon_k) - \chi^2(p_0^j - \epsilon_j + \epsilon_k) + \chi^2(p_0^j - \epsilon_j - \epsilon_k)}{4\epsilon_j\epsilon_k}$$

where the difficulty is knowing a suitable N-dimensional ϵ_j

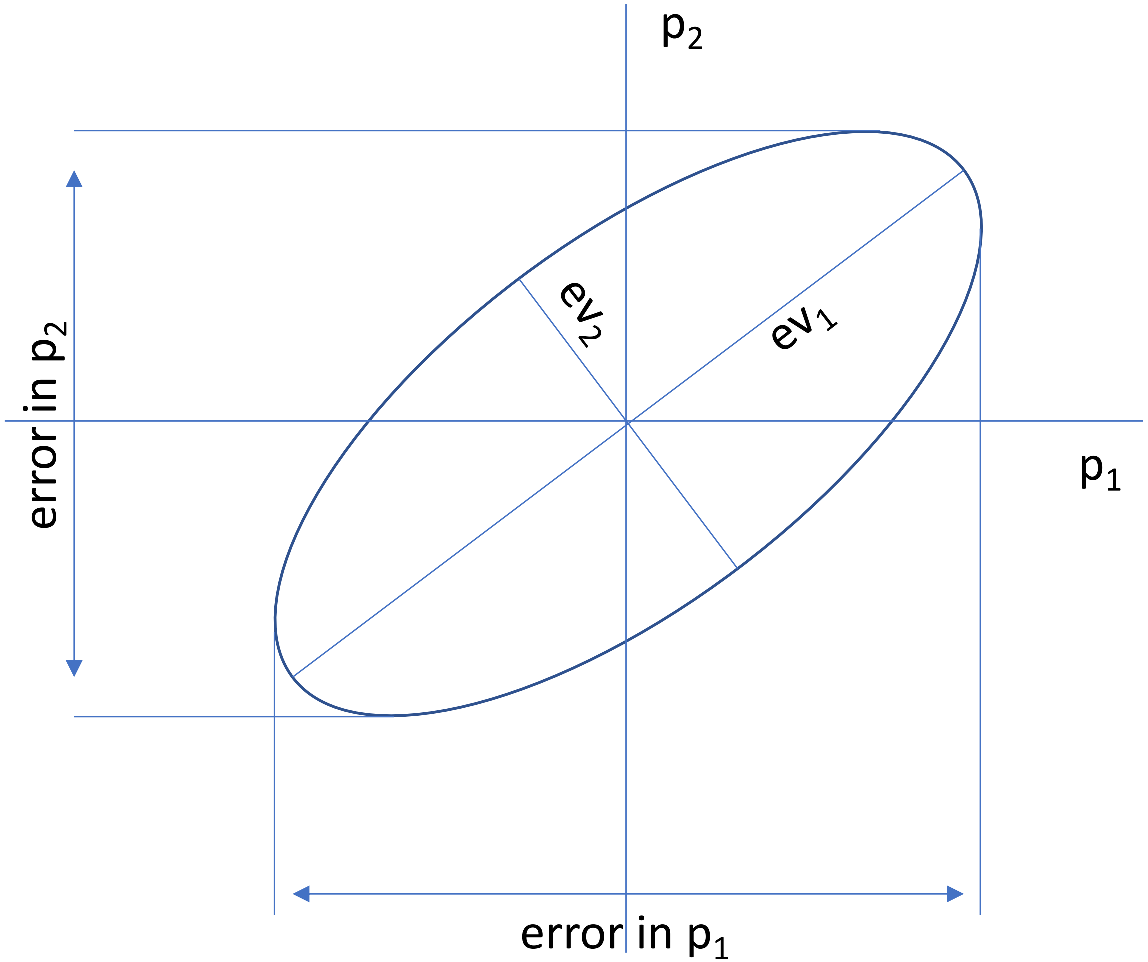
Potential problems:

- You cannot invert because (at least) one Eigenvalue is 0
  ➡ complete parameter degeneracy
- You cannot take the sqrt because (at least) one diagonal element is < 0
  ➡ you are not at a minimum, but rather a saddle point, and you can minimize further

You anyhow want the errors?

- Option 1: Use a Pseudo-Inverse
- Option 2: Add positive numbers to the diagonal, until one can invert

Eigenvalues and Eigenvectors



- The semi-major and –minor axes of the ellipse are the eigenvalues of the covariance matrix
- The angle is $\tan 2\phi = \dfrac{C_{ij}}{C_{ii} - C_{jj}}$
- The eigenvectors give the parameters in which the problem is uncorrelated
- In these parameters, the covariance matrix is diagonal
- Note that the constraints are "better" in these parameters than the 1D-projected ones of the original parameters

Not trusting the error matrix? Propagate the errors yourself!

- Create M new data sets by perturbing each data point
- Assume Gaussian errors on the data, and add/subtract a random Gaussian number with width of the (1 σ) error bar
- Re-fit M times
- Take the width of the resulting parameter distributions as errors

What is the error on the orbital period?

- I have fitted semi-major axis *a* and mass *M*, and want to know the error on the period

$$T = 2\pi\sqrt{\frac{a^3}{GM}}$$

- Standard error propagation:

$$\left(\frac{\Delta T}{T}\right)^2 = \frac{9}{4}\left(\frac{\Delta a}{a}\right)^2 + \frac{1}{4}\left(\frac{\Delta m}{m}\right)^2$$

- Is WRONG! It misses the correlations.

In general going from p_i to q_k:

$$D_{kl} = \sum_{i,j} \frac{\partial q_k}{\partial p_i}\frac{\partial q_l}{\partial p_j} C_{ij}$$

n params p_i
m params q_k
n can be different from m

$$\begin{aligned}\Delta T^2 =& \frac{\partial T}{\partial a}\frac{\partial T}{\partial a} C_{aa} + \frac{\partial T}{\partial m}\frac{\partial T}{\partial m} C_{mm} \\ &+ \frac{\partial T}{\partial a}\frac{\partial T}{\partial m} C_{am} \frac{\partial T}{\partial m}\frac{\partial T}{\partial a} C_{ma} \\ =& \left(\frac{\partial T}{\partial a}\right)^2 (\Delta a)^2 + \left(\frac{\partial T}{\partial m}\right)^2 (\Delta m)^2 \\ &+ 2\frac{\partial T}{\partial a}\frac{\partial T}{\partial m} C_{am}\end{aligned}$$

Correlation coefficient

With the covariance matrix, the correlation coefficient is easily calculated:

$$r_{j,k} = \frac{C_{jk}}{\sigma_j \, \sigma_k}$$

The most frequent case will be that of linear fits.

The above relation also holds in case one fitted a line to data with errors in both axes.

Prediction with uncertainties

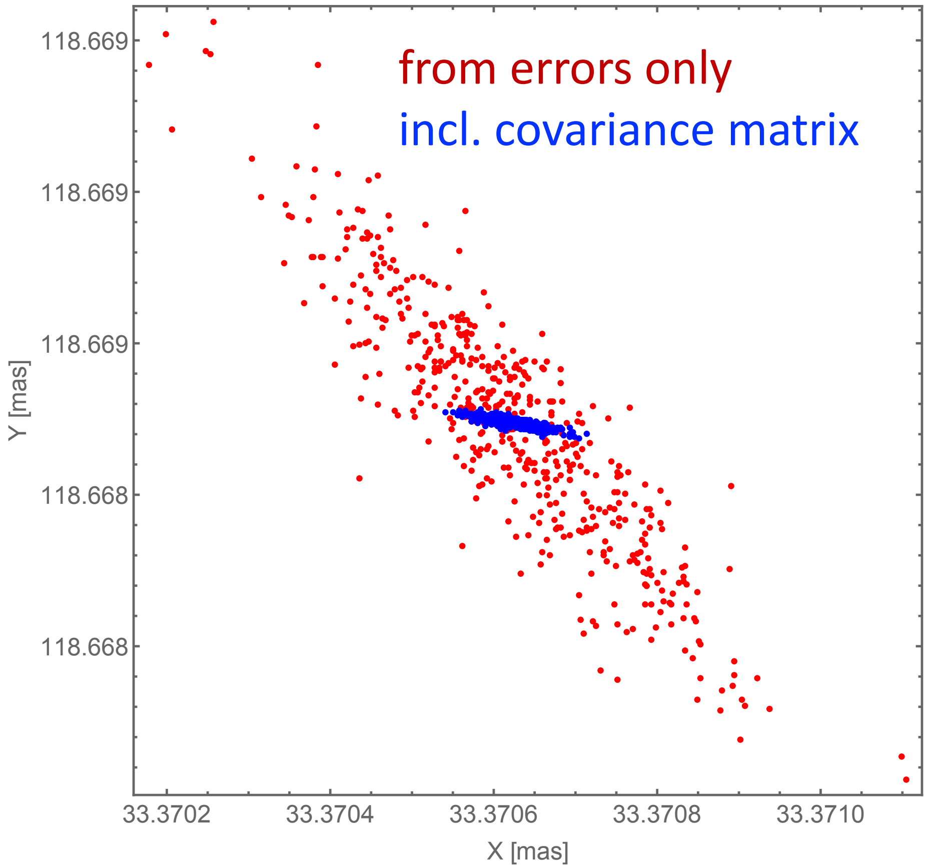
- you have: best fit model + error bars
- it is not sufficient to draw parameters according to the errors
- Need to take into account covariance
- Recipe:
  - Diagonalize covariance matrix
  - draw in the independent parameters

$$r = \mathrm{random}(0,1) \times 2 - 1$$

$$g = \sqrt{2}\,\mathrm{erf}^{-1}(r)$$

  - transform back to original parameters
  - calculate the prediction

S2 position predicted



Error bands around the model

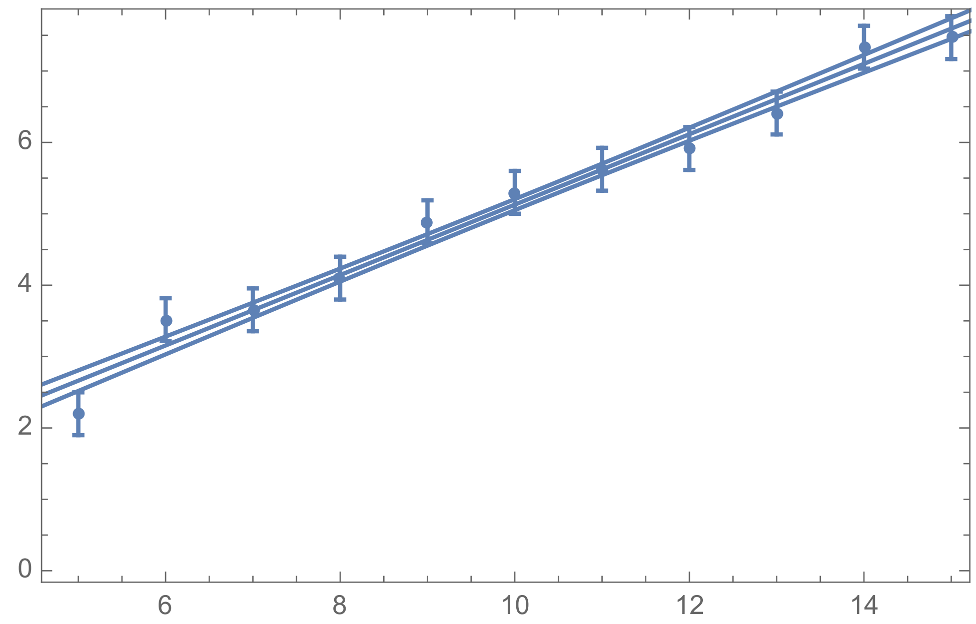
What is the uncertainty at any given point?

$$(\Delta f(x))^2 = \left(\frac{\partial f}{\partial p_j}\bigg|_x \right)^T . C_{jj} . \left(\frac{\partial f}{\partial p_j}\bigg|_x \right)$$

Example: straight line, no correlation:

$$f(x) = a + b\,x$$

$$(\Delta f(x))^2 = \sigma_a^2 + \sigma_b^2\,x^2$$



Or the expensive way:

- at each point x you need, predict with uncertainties, i.e. a small Monte Carlo
- at each point x get the local confidence interval in f
- connect these points

Is a fit good?

- Define normalized residuals: $$r_i = \frac{y_i - f_0(x_i, p_0^j)}{\Delta y_i} \qquad \chi_0^2 = \sum r_i^2$$
- These r_i should be a normal distribution around 0 with width 1
"on average, each data point should be 1 σ away from the best fit"
- Expect thus: $\chi^2 \approx \#\,\mathrm{data}$
- A bit more precisely: $\chi^2 \approx \#\,\mathrm{data} - \#\,\mathrm{parameters}$
- Thus, one defines the "reduced χ^2": $$\chi_r^2 = \chi_0^2/\mathrm{d.o.f.}$$

Note: The value of the reduced χ^2 in absolute terms ("it needs to be 1") is only meaningful if one can trust the error bars of the underlying data.

In practice: Values between 0.1 and 10 might be fine (!)

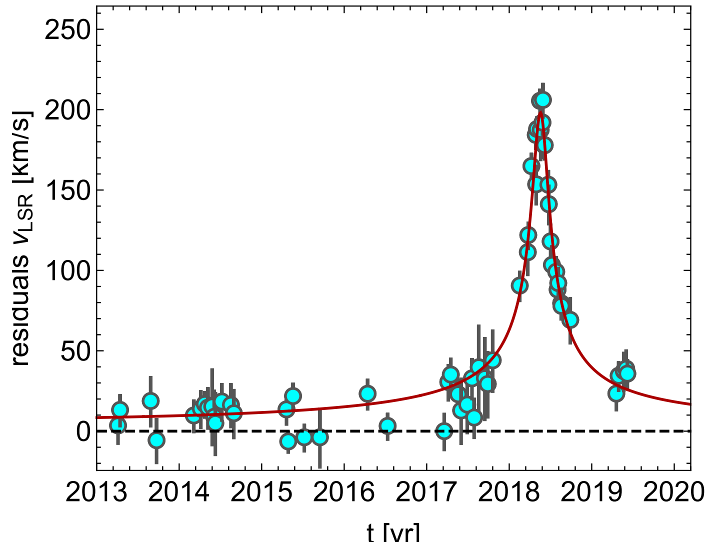
What one ALWAYS should do: Inspect the residuals

Does the fit capture the feature in the model what you think is your signal?
- maybe you have a mistake in the model?
- maybe the fit did not find a proper minimum?

Extreme outliers?
- Could be a hint for a mistake, inspect that data point
- allowed to remove

Is there a tail of outliers?
- Consider outlier robust fitting

Are all points more or less described equally well / bad?
- Consider error rescaling



Error rescaling, or adding constant

Are all points more or less described equally well / bad?

- Consider error rescaling

$$\chi^2 = \Sigma \frac{(y_i - f(x_i,\, p_j))^2}{\Delta y_i^2}$$

Rescaling (1):

force $\chi_r^2 = 1$
by multiplying all errors with $\sqrt{\chi_r^2}$

no need to fit again, all relative weights remain the same and errors scale accordingly

Rescaling (2):

force $\chi_r^2 = 1$
by adding an “error floor”

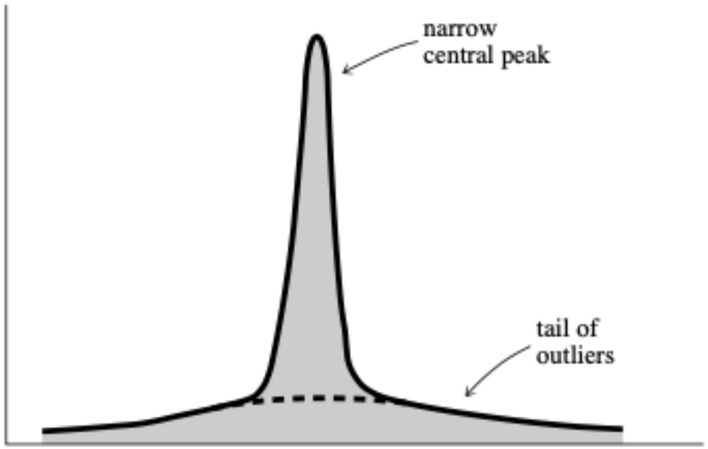
$$\chi^2 = \sum \frac{(y_i - f(x_i, p_j))^2}{\Delta y_i^2 + c^2}$$

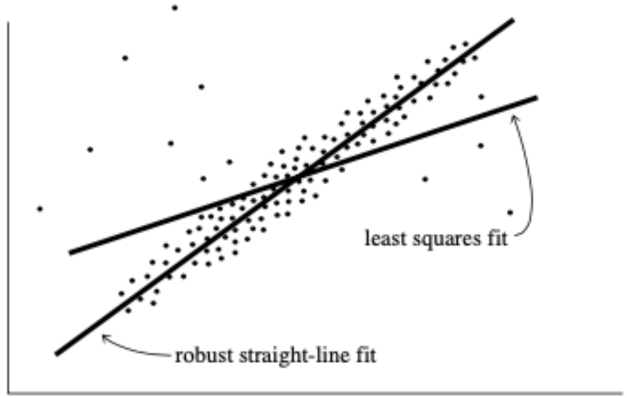
need re-fitting

Outlier robust fitting

Is there a tail of outliers?

- Consider outlier robust fitting





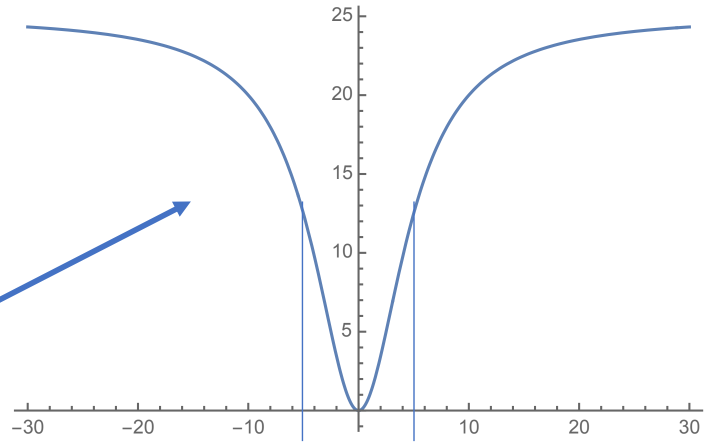
$$r_i = \frac{y_i - f_0(x_i, p_0^j)}{\Delta y_i}$$

$$p(r) = r^2$$

$$\chi^2 = \sum p(r)$$

$$p(r, s) = r^2 \cdot s^2/(r^2 + s^2)$$

$$s \approx 5..10$$



quadratic regime

asymptotic regime

asymptotic regime

Is a certain fit better than another?

Don't judge before having seen the residuals!

1) Simple version: Has the χ_r^2 improved significantly?

What is the uncertainty on the χ_r^2 ?

In the sense of 1σ it is: $\Delta\chi_r^2 \approx \sqrt{2/N}$

2) Information criteria: Bayes IC, Aitken IC

$$\mathrm{BIC} = \chi^2 + \#\mathrm{par} + ln(\#\mathrm{data})$$

$$\mathrm{AIC} = \chi^2 + 2\#\mathrm{par}$$

and look at Δ BIC or Δ AIC

$< 10^0$	Negative (supports M_2)
10^0 to $10^{1/2}$	Barely worth mentioning
$10^{1/2}$ to 10^1	Substantial
10^1 to $10^{3/2}$	Strong
$10^{3/2}$ to 10^2	Very strong
$> 10^2$	Decisive

”Is the additional parameter justified?”

- If model (1) is ”nested” in model (2)

$$\Delta\chi^2 = {}^{(1)}\chi^2 - {}^{(2)}\chi^2$$

$$f = \frac{\Delta\chi^2/\Delta\#\mathrm{par}}{{}^{(2)}\chi^2_r}$$

- Needs CDF of F-distribution

$$1 - p = \int_0^f \mathcal{F}(x;\, \Delta\#\mathrm{par},\, {}^{(2)}\mathrm{d.o.f.})\, dx$$

- The result is significant at level

$$\sigma = \sqrt{2}\, \mathrm{erf}^{-1}(1-p)$$

- “quadratic vs. linear”
- “break or straight line”
- “GR” or “Newton”

“My fit is not working”

- Cause #1: **Bad starting values**
  “you must know the result before”

- Cause #2: **Bad parametrization**

- Cause #2.A: Some parametrizations are more efficient than others:
  - example: eccentricity e, (1-e), ln (1-e)

- Cause #2.B: The parameters have very different ”influencing power”
  - Then it gets hard to minimize the weaker ones

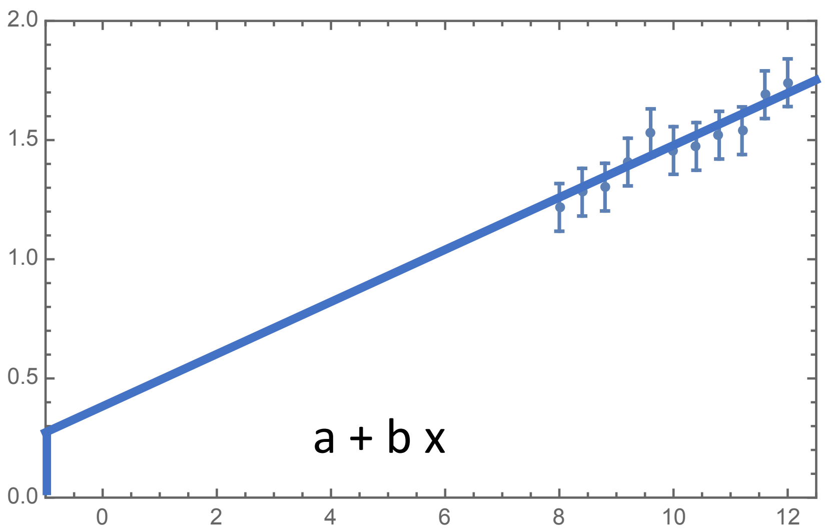
Condition number C:
ratio of largest to smallest Eigenvalue of covariance matrix

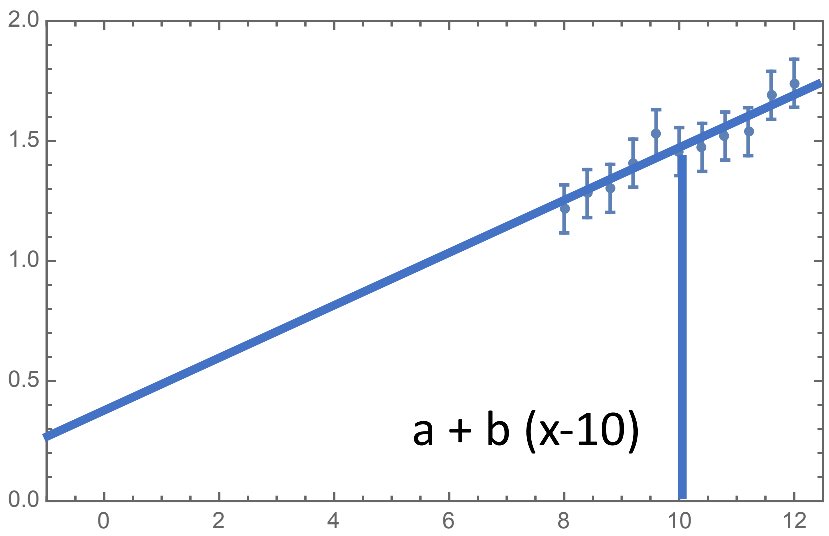
For inverting a matrix (which numerical minimization does at each step)
the condition number should be $C \lesssim 1/\sqrt{p}$ where p is the precision

For calculations with double precision, $C \approx 10^8$ starts to be problematic

Fitting independent parameters is easier

("the minimizer can change one parameter without needing to fiddle with the other")



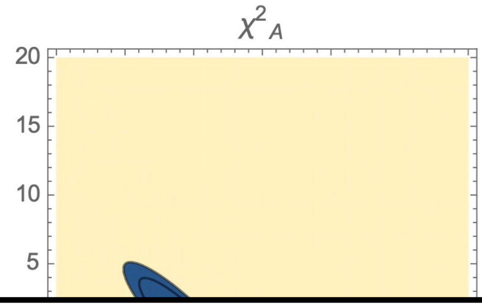


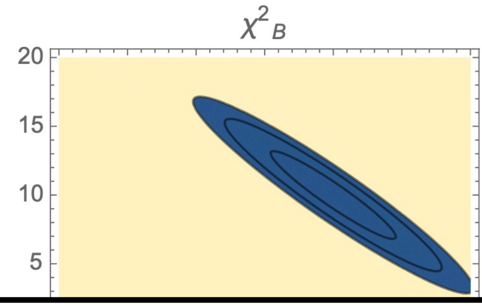
covariance matrix:

$$\begin{pmatrix} 0.0160751 & -0.00158219 \\ -0.00158219 & 0.000158219 \end{pmatrix}$$

$$\begin{pmatrix} 0.000253151 & 0 \\ 0 & 0.000158219 \end{pmatrix}$$

Be careful when combining data - check consistency!





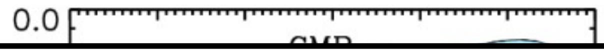
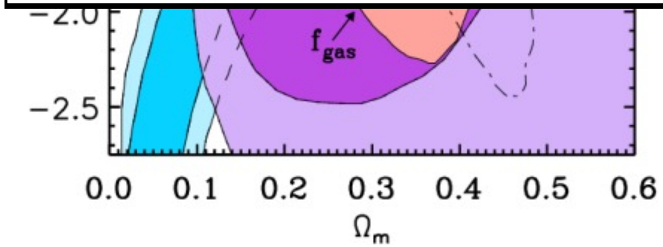
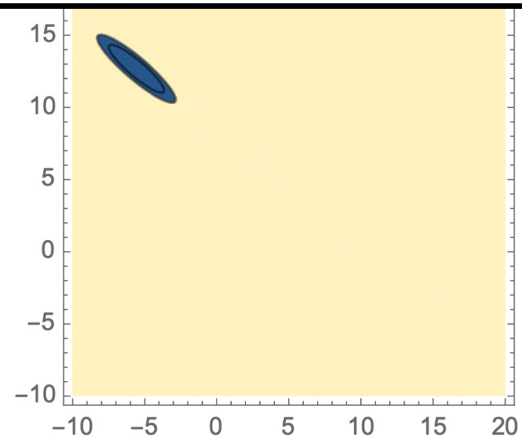


Table 4
Best-fit Black Hole and Orbital Parameters as Derived from the Fit of S0-2 Alone, S0-38 Alone, and the Simultaneous Fit of S0-2 and S0-38

Model Parameter (units)	Best-fit Parameter Values from Orbital Fits[a]		
	S0-2 Only	S0-38 Only	S0-2 and S0-38
Black Hole Properties:			
Distance (kpc)	8.02 ± 0.36 ± 0.04	[6.5, 9.5][b]	7.86 ± 0.14 ± 0.04
Mass (10^6 $M_\odot$)	4.12 ± 0.31 ± 0.04	[2.5, 5.5][b]	4.02 ± 0.16 ± 0.04

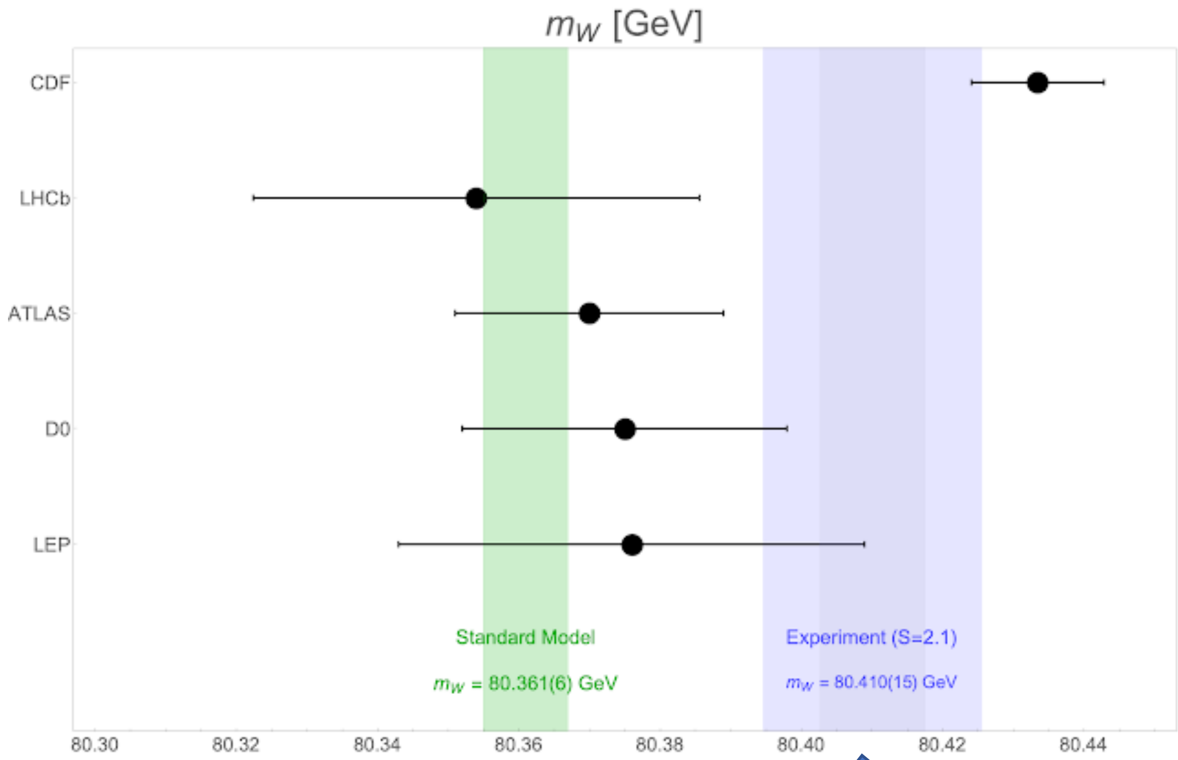


this looks good



severely underestimated errors!

W-Boson mass



Naïve combination:

mW = 80.410(7) GeV

is nonsense. None of the measured values is in the 68% error band

Proposed solution: rescale all errors until χ_r^2 is 1.

Particle Physics Blog: "The question of combining information from incompatible measurements is a delicate one, residing at a boundary between statistics, psychology, and arts."

The sampling error (I) "Jackknife Test"

- You have fitted N data points.
- Form N subsets of data with N – 1 points by leaving out one data point at a time
- Fit each subset and take the sample of best fit parameters $p_j^{(-i)}$

$$^{\text{jack}}\Delta p_j^2 = \frac{N-1}{N} \sum (p_j^{(-i)} - p_{0,j})^2$$

- Needs N times fitting (not so bad), historically first resampling method

The sampling error (II) "Bootstrapping"

- You have fitted N data points.
- Form M >> N sets of data with N points
  by randomly drawing with replacement from your N data points
- repetition of data allowed (and wanted), ~1/e points will be duplicates
- Fit all M sets and take the sample of best fit parameters $p_j^{(-i)}$
- Their distribution can be used the estimate $^{\mathrm{boot}}\Delta p_j$
- Needs around M = 10^4 times fitting, computationally demanding
  - but easy to parallelize

Markov-Chain Monte Carlo

Sampling the parameter space

“You walk through the parameter space and remember all points visited”,
i.e. you build a chain of points

Basic (Metropolis-Hastings) algorithm:

- You are at p_1
- Calculate $\chi^2(p_1)$
- Take a random step to p_2
- Calculate $\chi^2(p_2)$
- if $\chi^2(p_2)$ < $\chi^2(p_1)$ your next start point is p_2
- if $\chi^2(p_2)$ > $\chi^2(p_1)$ draw a random variable 0 < r < 1 (uniformly)

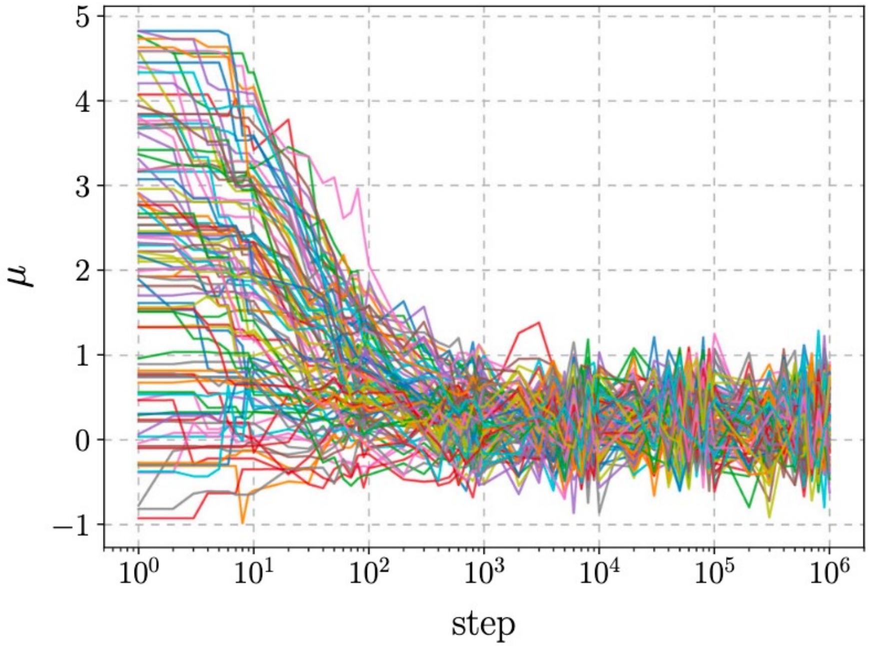
  if $r < e^{(\chi_1^2 - \chi_2^2)/2}$ your next start point is p_2, otherwise p_1

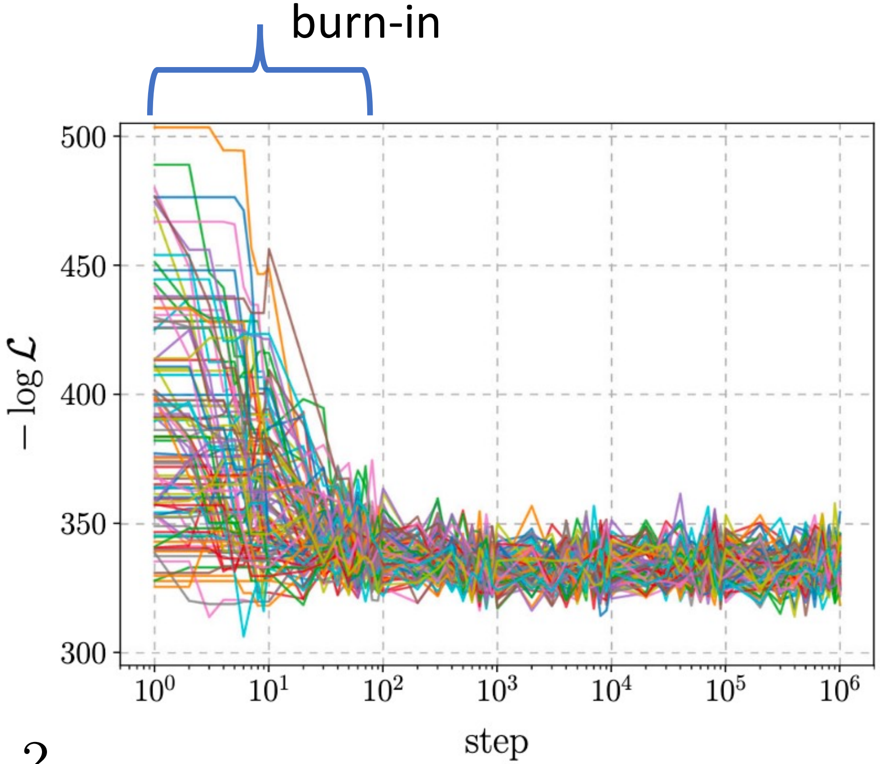
- Better work in independent variables
- Found by diagonizing covariance matrix

MCMC in practice

- many variants, for example several "walkers"
  - this can be parallelized, a chain not

- In the beginning, such a chain will move towards minimum
  - either start already at minimum (if interested in errors)
  - or throw away "burn-in" phase





- You need around 10^5 evaluations of χ^2

Do not confuse starting values and priors

- **Starting values**: Initial estimates. The MCMC should be independent of these, one cuts away the initial “burn-in phase”.
- **Priors**: Additional information that should be taken into account in addition to what the new data tell
- Using priors means, the result will be a mixture of these priors and the new information (Bayes theorem)
- One is **allowed to “play” with the starting values**, one is **not allowed to “play” with the priors**
- Flat priors with hard boundaries are sometimes used to limit the fit range. Make sure not to introduce a bias on the result

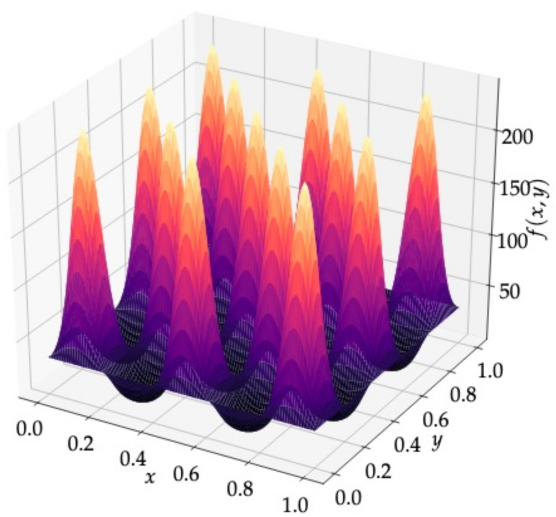
MCMC – some considerations

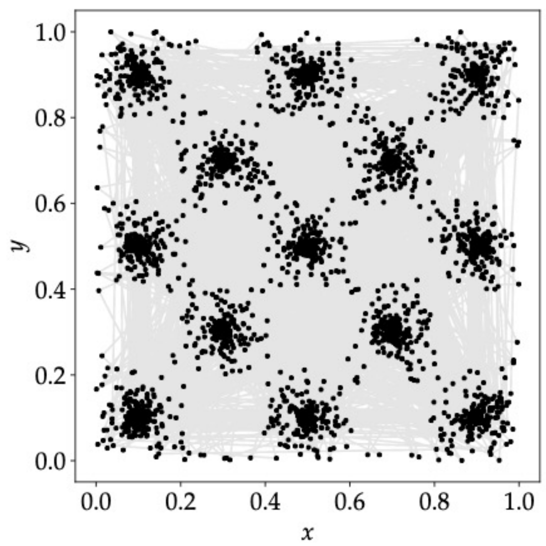
- MCMC is useful to get to know the structure of the parameter space
  - is it well-behaved?
  - multiple minima?

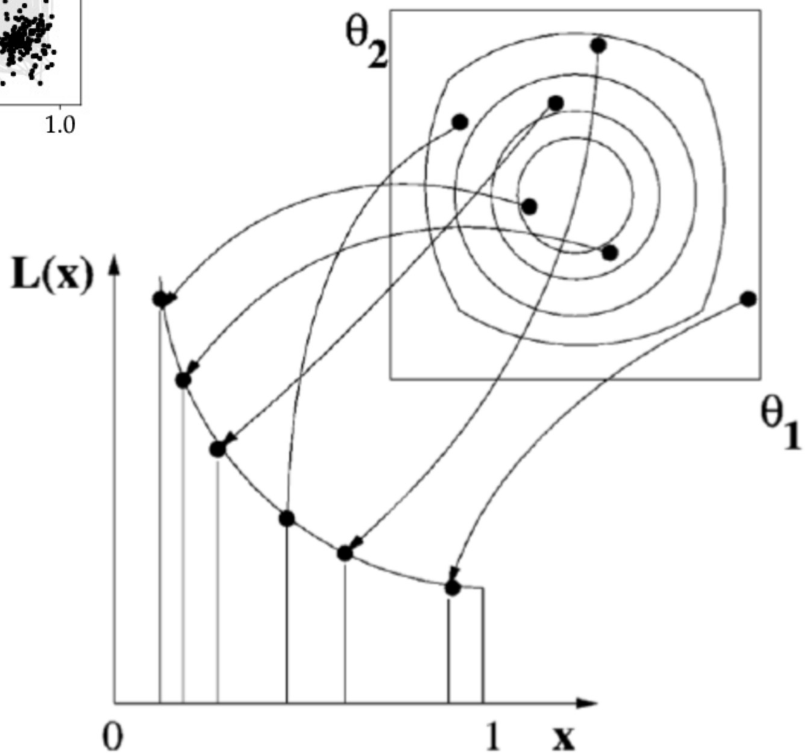
- MCMC yields error estimates from the widths of the parameter distribution
  - for well-behaved, Gaussian problems it much less efficient than the error matrix

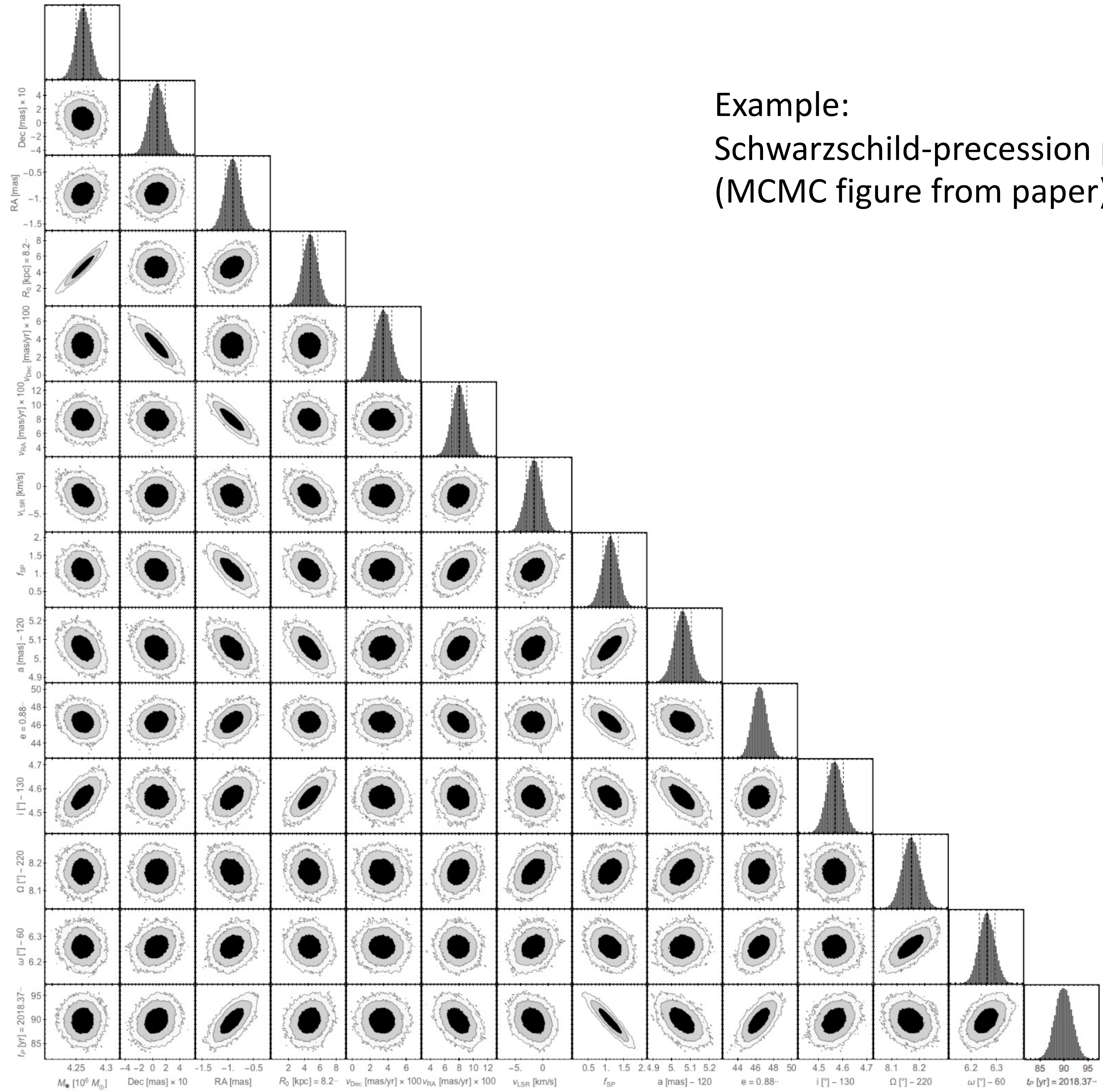
- MCMC is inefficient for finding the minimum (“not a fit”)
  - but can be crucial to show that the minimum found is the global one

MCMC – nested sampling



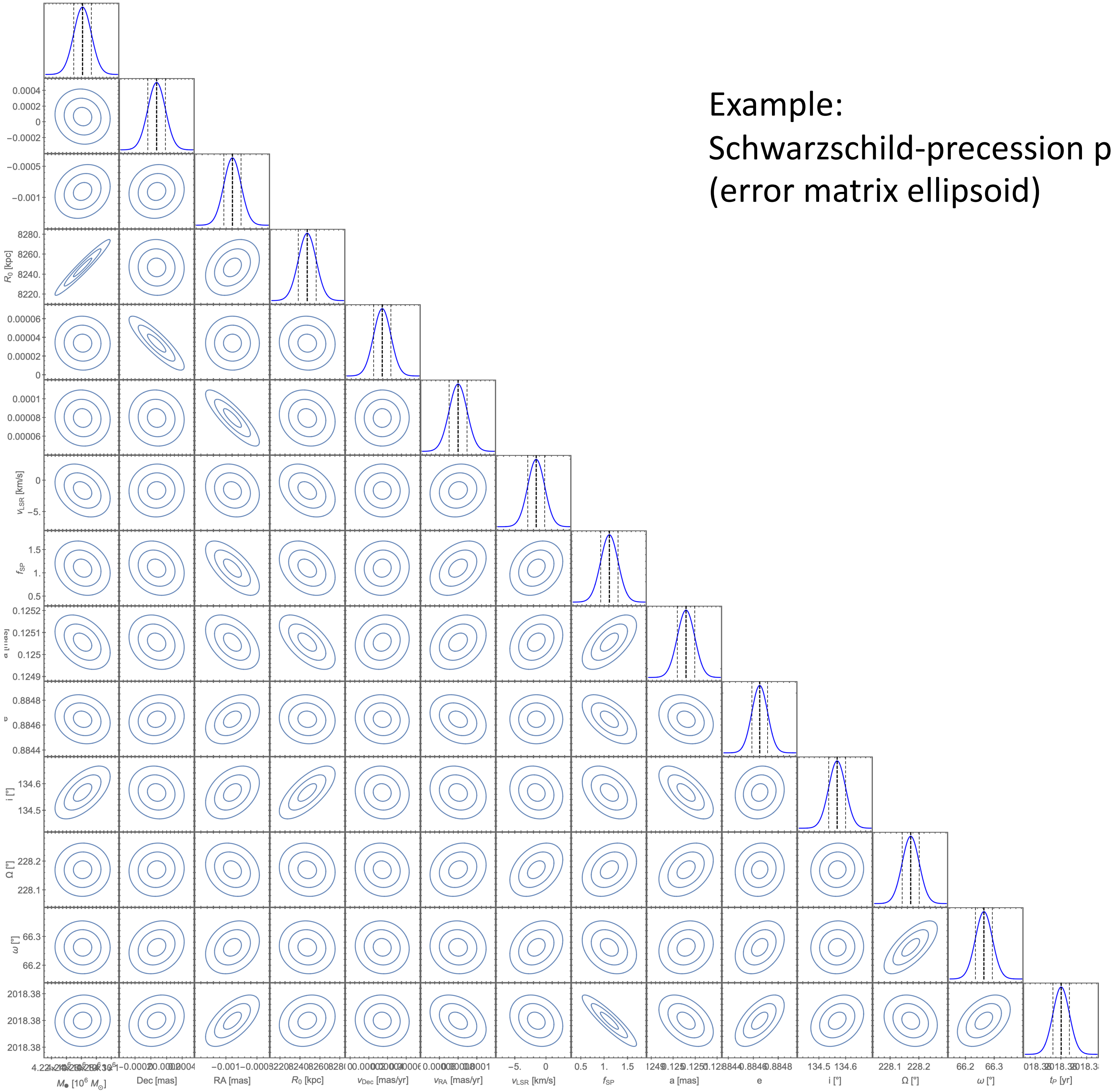






Example:
Schwarzschild-precession paper
(MCMC figure from paper)

Example:
Schwarzschild-precession paper
(error matrix ellipsoid)



Error propagation with MCMC is simple

- You don't need to know the derivatives of the transformation q <-> p

$$D_{kl} = \sum_{i,j} \frac{\partial q_k}{\partial p_i} \frac{\partial q_l}{\partial p_j} C_{ij}$$

- Only needed:
  transformation function q(p)
- Your chain with N samples is $\{p_i\}_N$
- Calculate q_k (p_i) for each of the N samples
- Your posterior is simply $\{q_k\}_N$
- So the standard error on q_k is stddev $\{q_k\}_N$

If you want to …

N data points, M parameters

	computing demand	parallel?
• Find a minimum with a fit:	$10^{3\,..\,4}$ χ^2 evaluations	(no)
• Calculate error matrix:	M x (M-1) χ^2 eval.	(no)
• Run a MCMC chain	$10^{5\,..\,6}$ χ^2 evaluations	yes
• Do a jack-knife	N fits, N x $10^{3\,..\,4}$ χ^2	yes
• Do a bootstrap	10^4 fits, i.e. $10^{7\,..\,8}$ χ^2	yes