Bayesian and Frequentist statistics

Wine or Beer? Cats or Dogs? Ronaldo or Messi? Pelè or Maradona? Virtus or Fortitudo? Boca or River? Beatles or Rolling Stones? Lakers or Celtics? Panettone or Pandoro?….

Bayes or Frequentist?

Bayesian and Frequentist statistics

- Bayesian: data are fixed, model is repeatable
- Frequentist: model is fixed, data are repeatable

Say $H_0 = (72 \pm 8)$ km/s/Mpc. Then:

Bayesian: the posterior distribution for H_0 has 68% if its integral between 64 and 80 km/s/Mpc. The posterior can be used as a prior on a new application of Bayes' theorem.

Frequentist: Performing the same procedure will cover the real value of H0 within the limits 68% of the time. But how do I repeat the same procedure (generate a new H_0) if I only have one Universe?

Good references:

Bayesian: R. Trotta, “Bayes in the Sky”, https://arxiv.org/pdf/0803.4089.pdf

Frequentist: Feldman & Cousins, “A Unified Approach to the Classical Statistical Analysis of Small Signals”, https://arxiv.org/abs/physics/9711021

Bayesian and Frequentist statistics

- Bayesian:
  - can given probabilities for models
  - depends on both prior and likelihood (of data)
  - currently the dominant method in cosmology
- Frequentist:
  - doesn't give probabilities of models, only of hypotheses
  - doesn't depend on prior, just likelihood
  - currently the dominant method in particle physics

likelihood

prior

D = data

M = model

$$\boxed{\underbrace{P(M|D)}_{\text{posterior}} = \frac{\overbrace{P(D|M)}^{\text{likelihood}}\,\overbrace{P(M)}^{\text{prior}}}{\underbrace{P(D)}_{\text{evidence}}}} \quad \text{(Bayes' theorem)}$$

posterior

evidence

Bayes' Theorem for parameter estimation

Posterior

$$P(p|dM) = \frac{P(d|pM)P(p|M)}{P(d|M)}$$

Likelihood

Prior

$$\propto P(d|pM)P(p|M)$$

Model

Observed data

Parameters

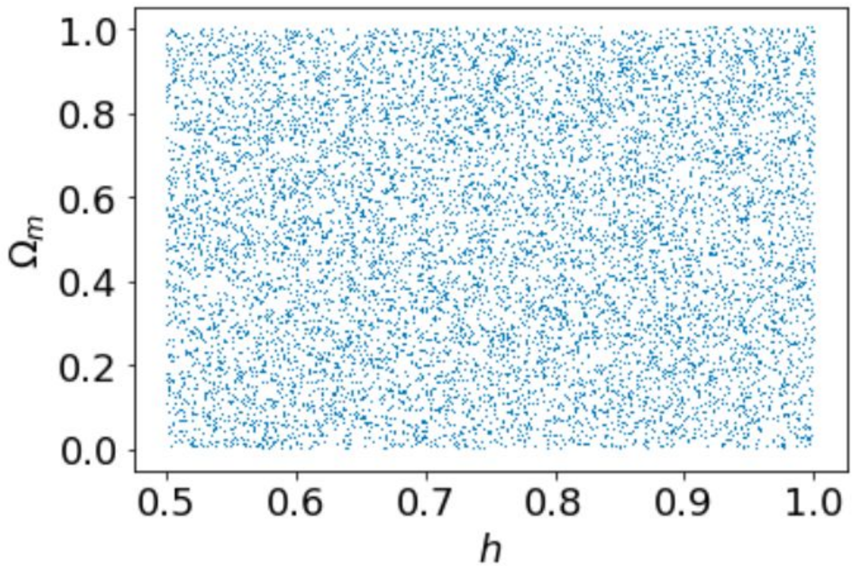
Quantifying Information

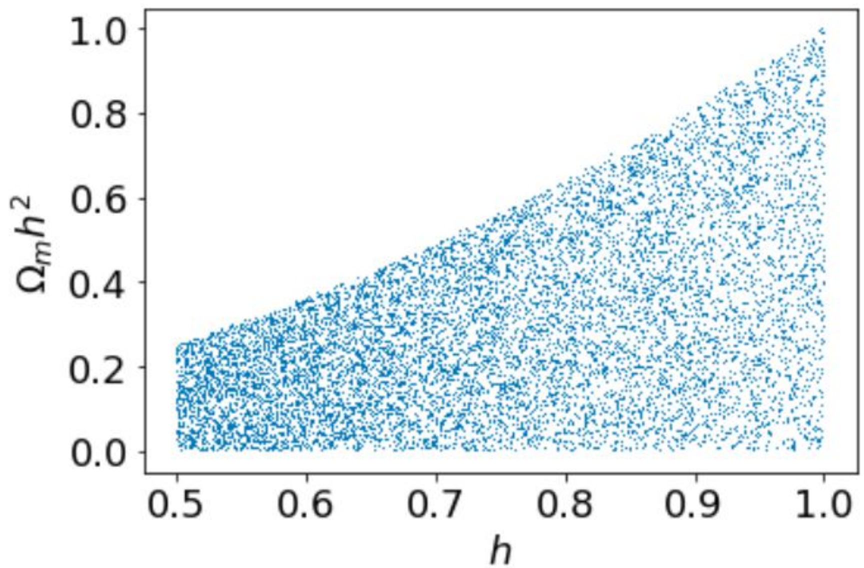
What you know after the experiment (posterior)
= what you knew before (prior)
+ what you learn (likelihood)

Priors

- Priors quantify what you knew about the parameters before you start
  - Theoretical limits, preferences, things that must be true from simpler data
- In regions where your likelihood is zero your prior doesn't matter for parameter estimation, but can for more advanced *model selection*
- It is common practice in cosmology to use uniform priors for most parameters
  - You should think more carefully than that!

Transformed Priors





Jointly uniform priors on Ω_m - h

Implied priors on $\Omega_m h^2$ - h

Bayes' Theorem for parameter estimation

Posterior

$$P(p|dM) = \frac{P(d|pM)P(p|M)}{P(d|M)}$$

Likelihood

Prior

$$\propto P(d|pM)P(p|M)$$

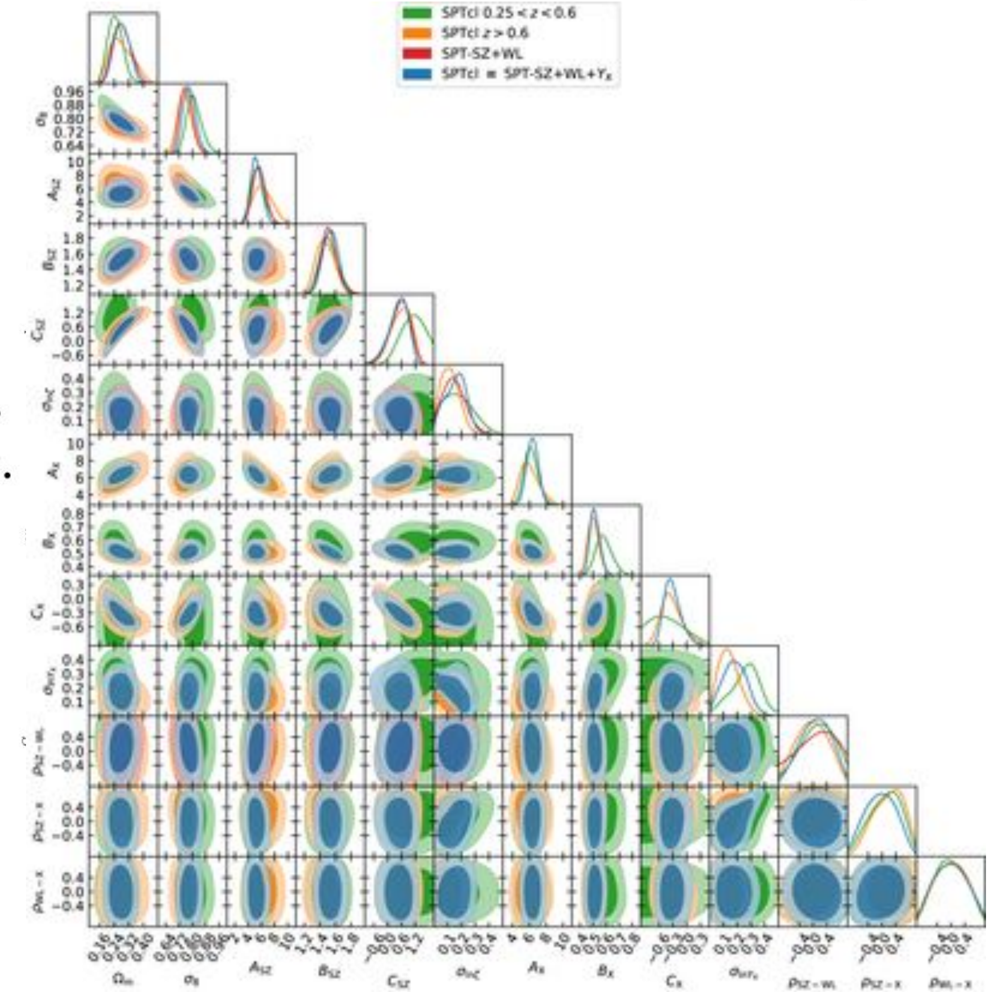
Model

Observed data

Parameters

Markov Chain Monte Carlo (MCMC)

The challenge: map out a posterior in multi-dimensional parameter space.



Example: say there are just 10 parameters.
Lets say calculation takes just 1 second/model.
Say you want a grid with 20 values in each par.
Then

$N = 20^{10} \simeq 10^{13}$

⇒ it would take 300,000 years to do it!

⇒ Totally impossible, ever!!

Amazingly clever, efficient solution to the problem:
Instead of gridding, <u>sample</u>!
"Walk" through the parameter space in a clever way in order to map out the likelihood “banana” just enough.

⇒ MCMC, invented at Los Alamos National Lab in 1950s.

Markov Chain Monte Carlo (MCMC)

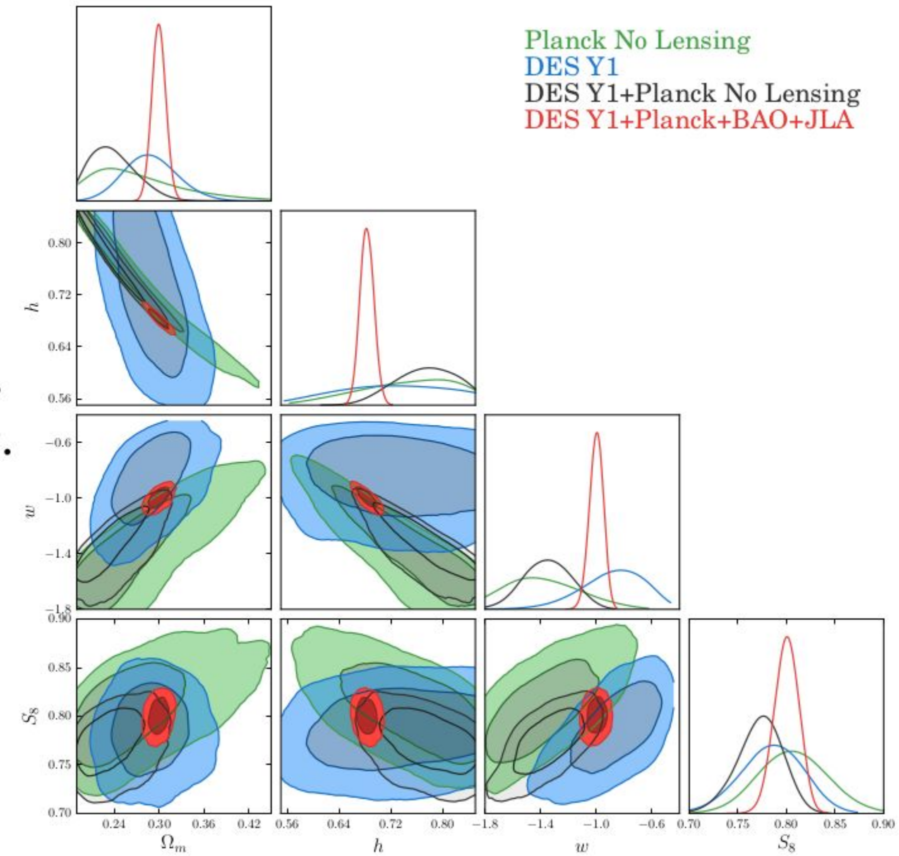
The challenge: map out a posterior in multi-dimensional parameter space.

Example: say there are just 10 parameters.
Lets say calculation takes just 1 second/model.
Say you want a grid with 20 values in each par.
Then

$N = 20^{10} \simeq 10^{13}$

$\Rightarrow$ it would take 300,000 years to do it!

$\Rightarrow$ Totally impossible, ever!!



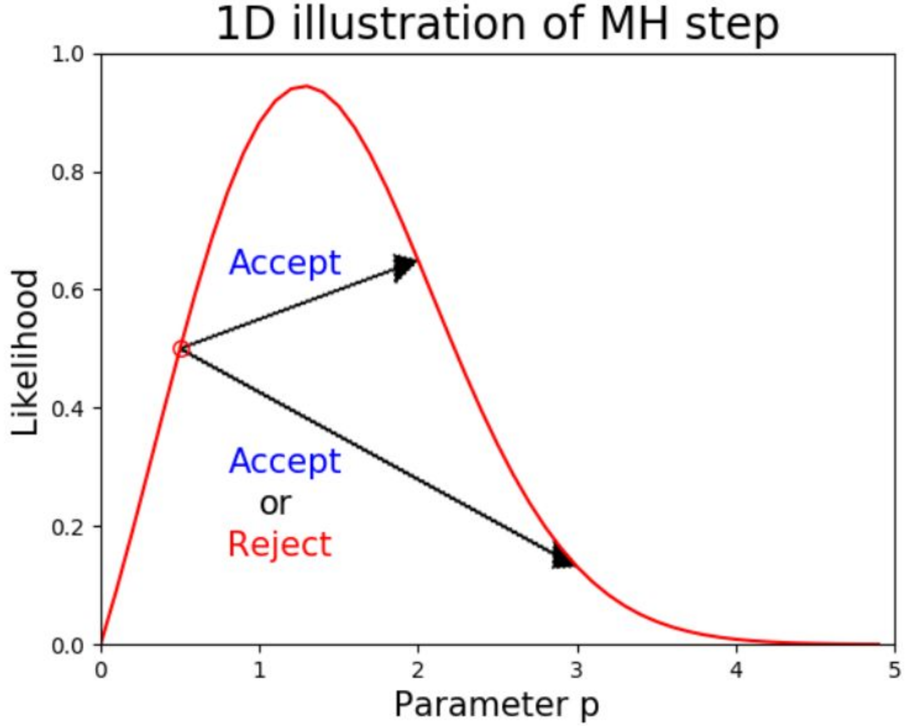
DES Y1 extensions paper (Abbot et al 2019);
the full param-space is 25-dimensional!

Amazingly clever, efficient solution to the problem:
Instead of gridding, <u>sample</u>!
"Walk" through the parameter space in a clever way in order to map out the likelihood "banana" just enough.

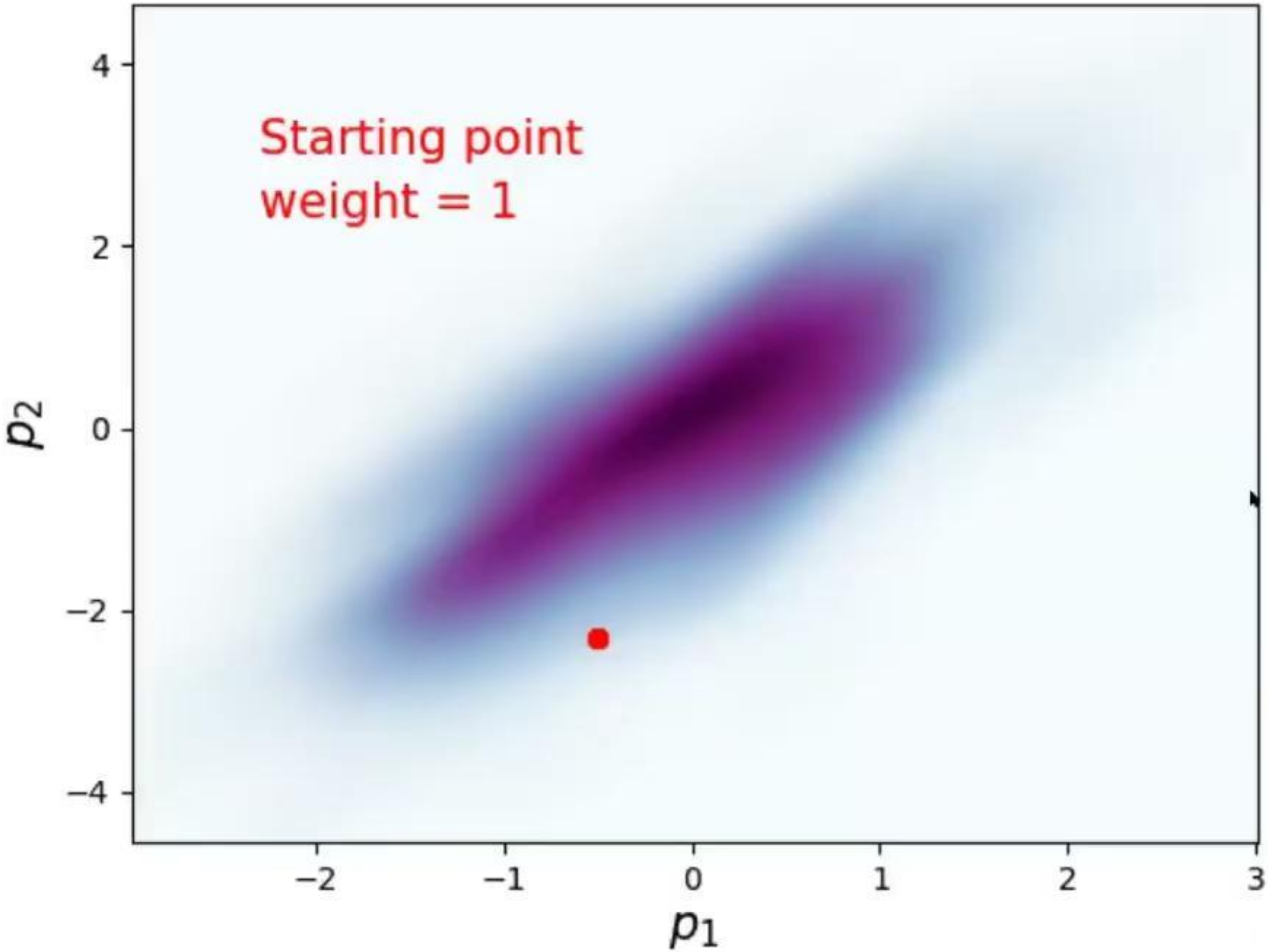
$\Rightarrow$ MCMC, invented at Los Alamos National Lab in 1950s.

MCMC: the Metropolis-Hastings algorithm

- at step t, at some parameters p_t
- propose move to $p_t' = p_t + \Delta p_t$ (randomly draw Δp_t)
- evaluate $r = L(p_t')/L(p_t)$
- MH step:
  - if $r > 1$ accept move
  - if $r < 1$ generate a random number $\alpha \in [0, 1]$
    - if $\alpha < r$, accept move
    - if $\alpha > r$, reject move
- t=t+1

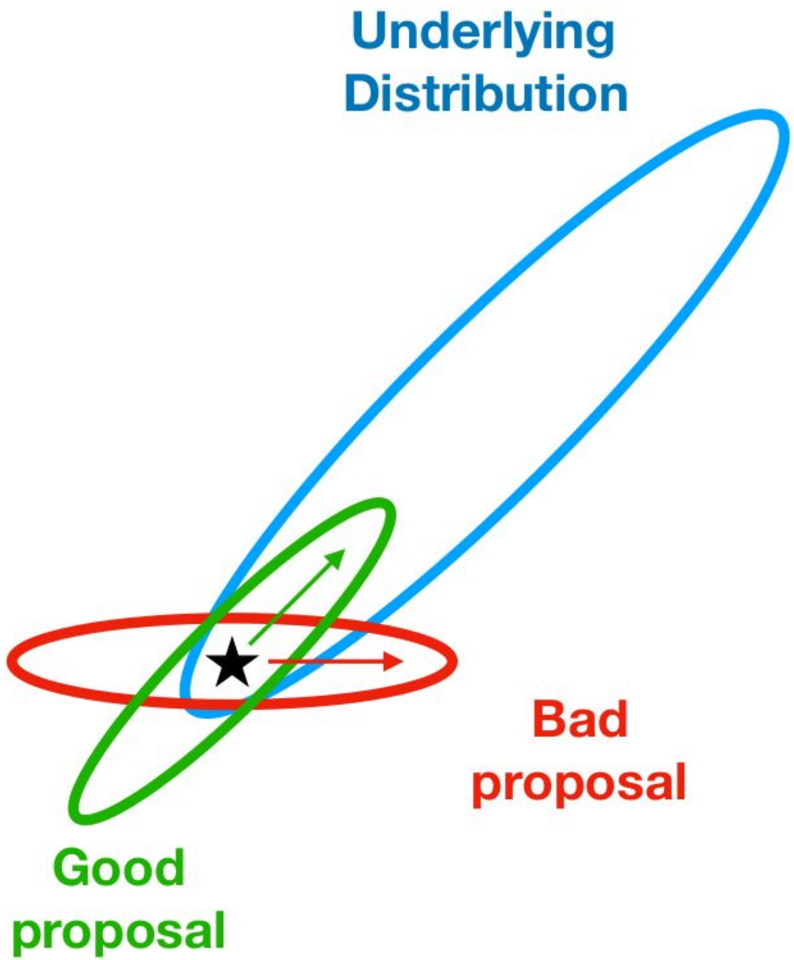


One can prove that,
with this rule,
one asymptotically recovers the
true posterior


Starting point
weight = 1
p_2
4
2
0
−2
−4
−2
−1
0
1
2
3
p_1

Good proposals

- Efficiency of MH depends dramatically on how good the proposal is
- A bad proposal will not converge in any practical length of time
- The ideal proposal matches the shape of the underlying distribution
    - We don't know this, but can look for best approximation

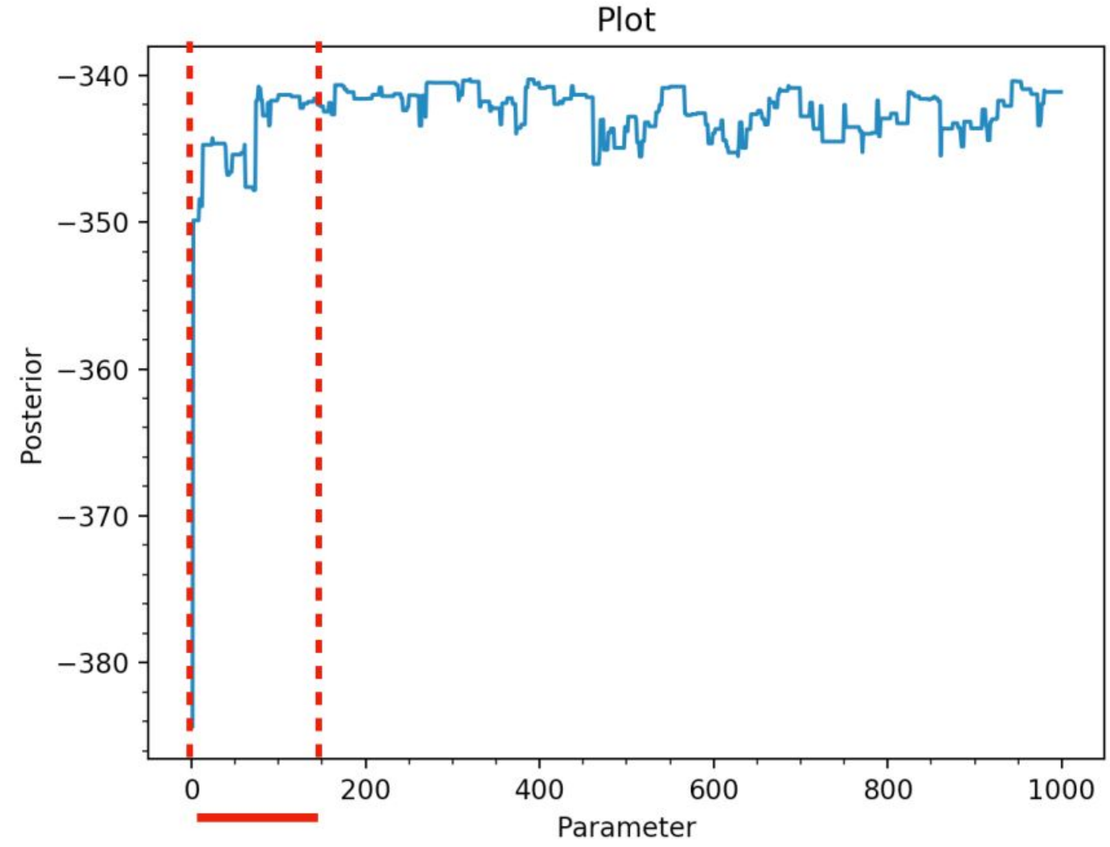


Burn-in

- Unless you're doing a simulation where you know the truth, unlikely to start at the best-fit value

- Will take some iterations to get near this point

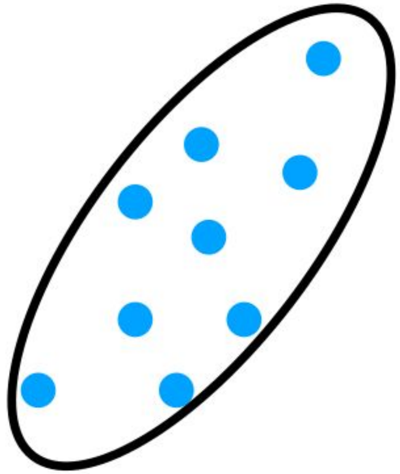
- Need to exclude these



Burn-in - exclude from sampling

Tuning

- One way to get a good proposal is by tuning
    - Run a short initial chain to estimate covariance
    - Use this covariance to initialise the next iteration
- You have to throw away the first chain, and only use samples from when your tuning was finished

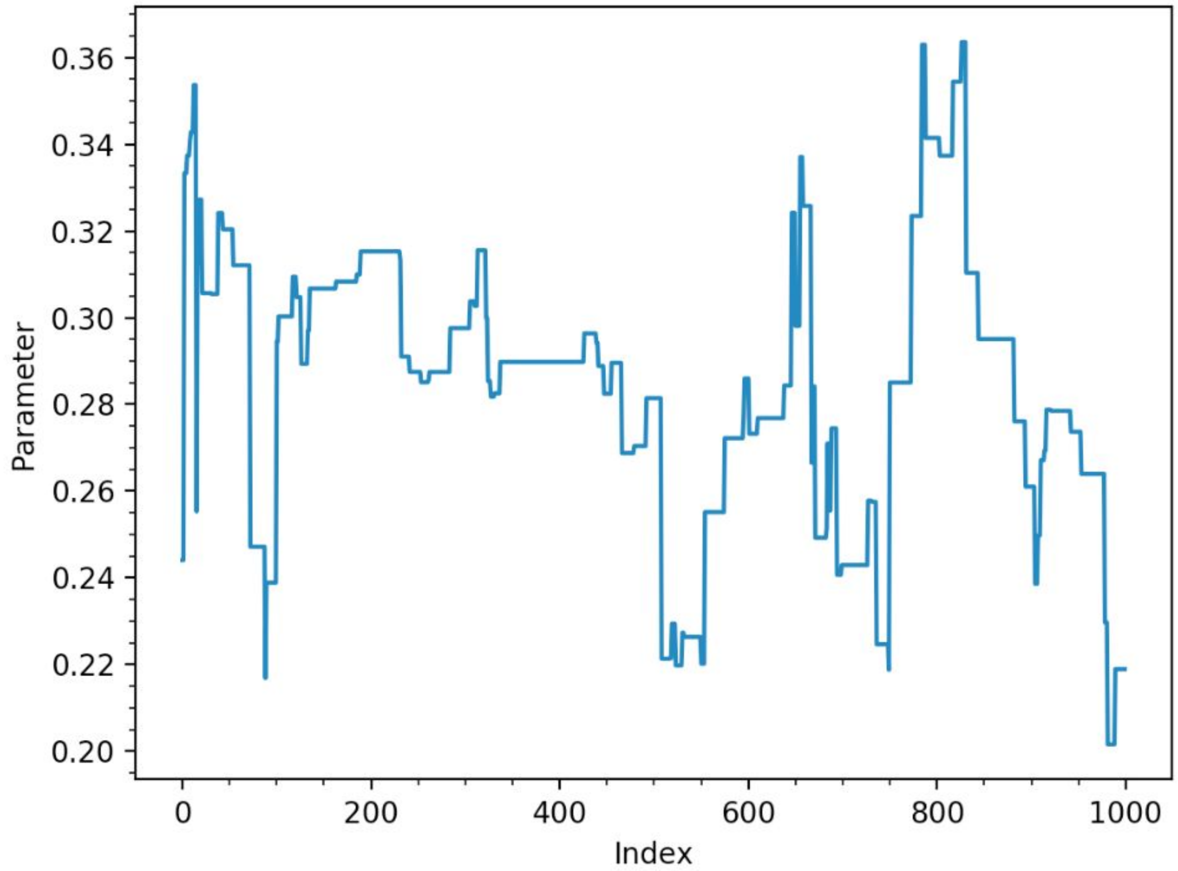


Checking Convergence - Traces

- Good MH chains look like white noise if you plot one parameters values throughout the chain

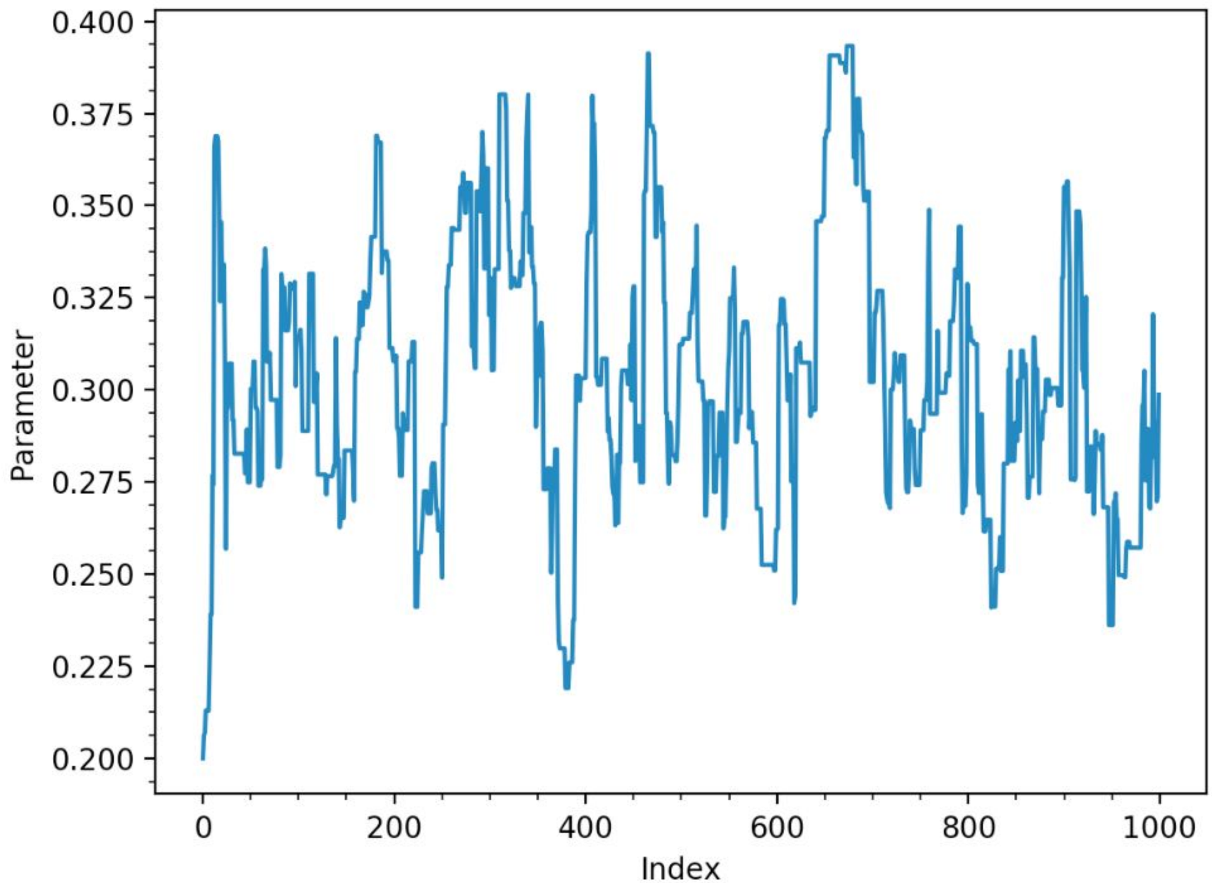
- Other errors can give clues as to issues

Checking Convergence - Traces



Bad - not long enough.
Chain getting stuck for long periods suggests covariance too large.
Acceptance rate too low.

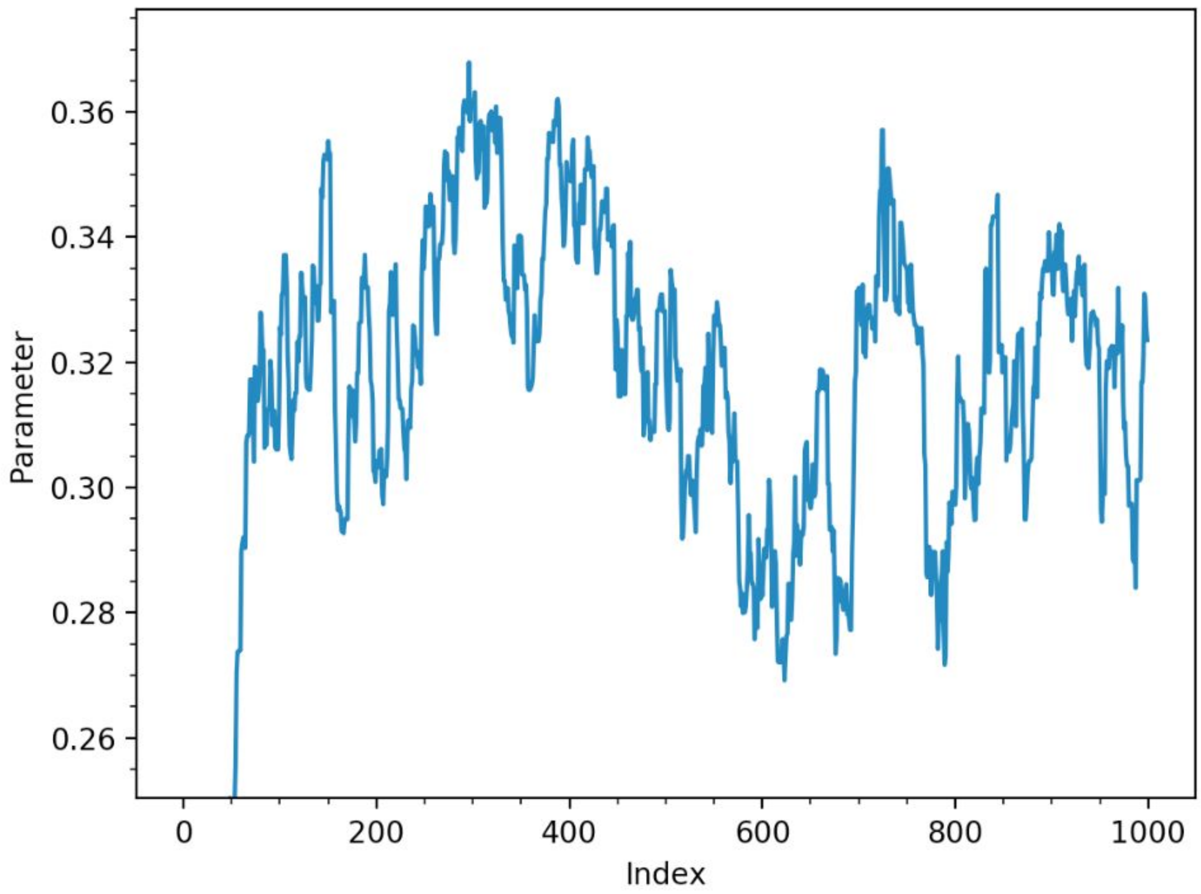
Checking Convergence - Traces



Looks reasonable - could be a bit longer

Quantify with the auto-correlation length

Checking Convergence - Traces



Bad - not long enough.
Chain is random walking, taking long divergences from mean suggests covariance too small.
Acceptance rate too high.

Checking Converge - Gelman-Rubin

- A more formal test for convergence compares chain characteristics
  - between multiple chains or within one chain, split up (if long enough)
- R^2 = (variance of means) / (mean of variances) among the chains
- If R - 1 is small chain is converged
- Typically $|R-1| < 0.01 \dots 0.05$

MCMC: interpreting the output

WEIGHT	P_1	P_2	P_3		P_N
5	0.2	-0.3	0.15	...	2.8
1	-0.7	0.4	0.12	...	3.5
12	0.7	0.1	0.19	...	1.7
...	...	...	...	...	...
...	...	...	...	...	...

(~ MILLION ROWS)

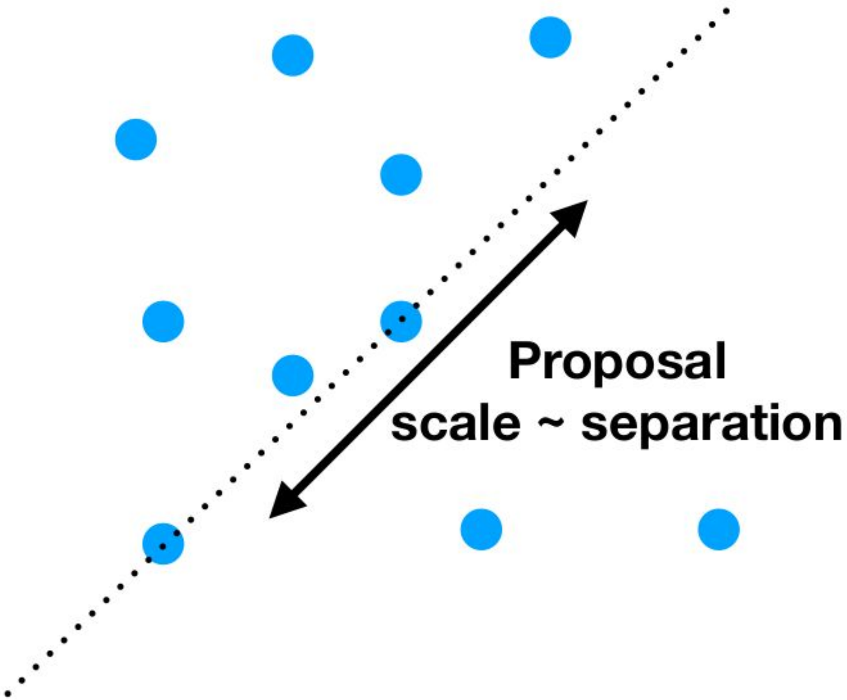
To get the posterior probability,
simply histogram the parameter values vs weights - this is your posterior!

Want to look at posterior in p_3 marginalized over all other parameters?
Simply plot histogram of p_3 values vs weight (eaaasy!)

MCMC is an incredibly clever, powerful set of algorithms without which data-driven cosmology wouldn't have gotten far.

Other Methods: Ensembles

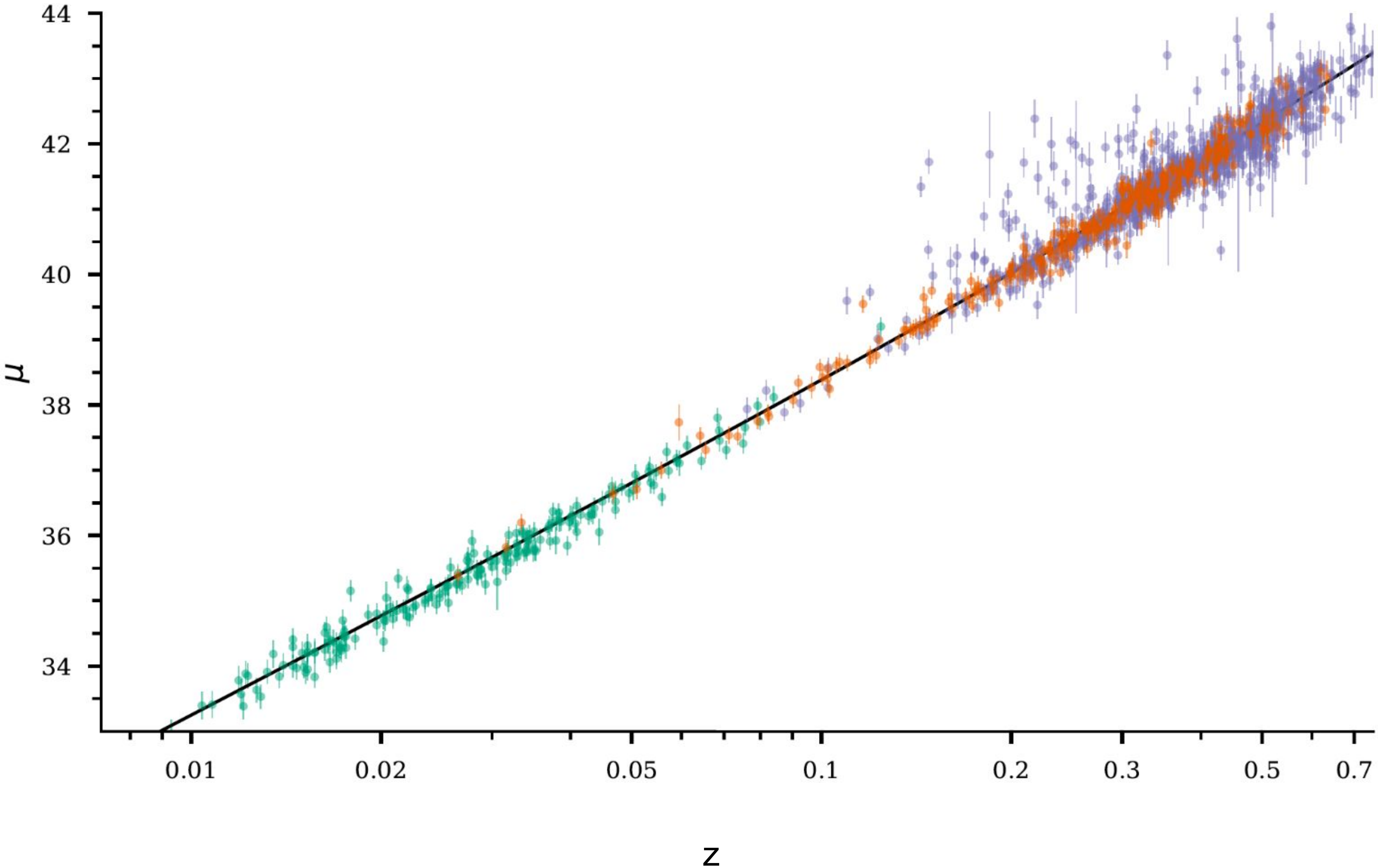
- Many methods use groups of points in parameter space, instead of just one
- These ensembles of points are then updated together
- The most famous is **emcee**, which proposes new points by drawing lines connecting current ones
- Very nice black box implementation, broadly accessible
- Great in moderate dimension but starts to be slow at dim > 25
- Burn-in much longer



https://emcee.readthedocs.io/en/stable/

https://arxiv.org/abs/1202.3665

Supernova measurements



Mean prediction for Supernovae

Supernovae are standard* candles. We observe their apparent magnitude, and our theory prediction for it comes from a distance metric, the luminosity distance:

$$E(z) \equiv \frac{H(z)}{H_0} = \sqrt{\Omega_m(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda}$$

$$D_C(z) = \frac{c}{H_0}\int_0^z \frac{\mathrm{d}z'}{E(z')}$$ *comoving distance*

$$D_M = \frac{c}{H_0}|\Omega_K|^{-1/2}\sin_k\left(|\Omega_K|^{1/2}\frac{H_0}{c}D_C(z)\right)$$ *comoving transverse distance*

$$D_L(z) = (1+z)D_M(z)$$ *luminosity distance*

$$\mu = 5\log_{10}\left(\frac{D_L}{10\mathrm{pc}}\right)$$ *distance modulus*