Title: Numerical Methods Lecture - 230126

Speakers: Erik Schnetter

Collection: Numerical Methods (2022/2023)

Date: January 26, 2023 - 9:15 AM

URL: https://pirsa.org/23010007

Affine-Invariant Samplers

Dustin Lang
Perimeter Institute for Theoretical Physics

PSI Numerical Methods, 2023-01-26

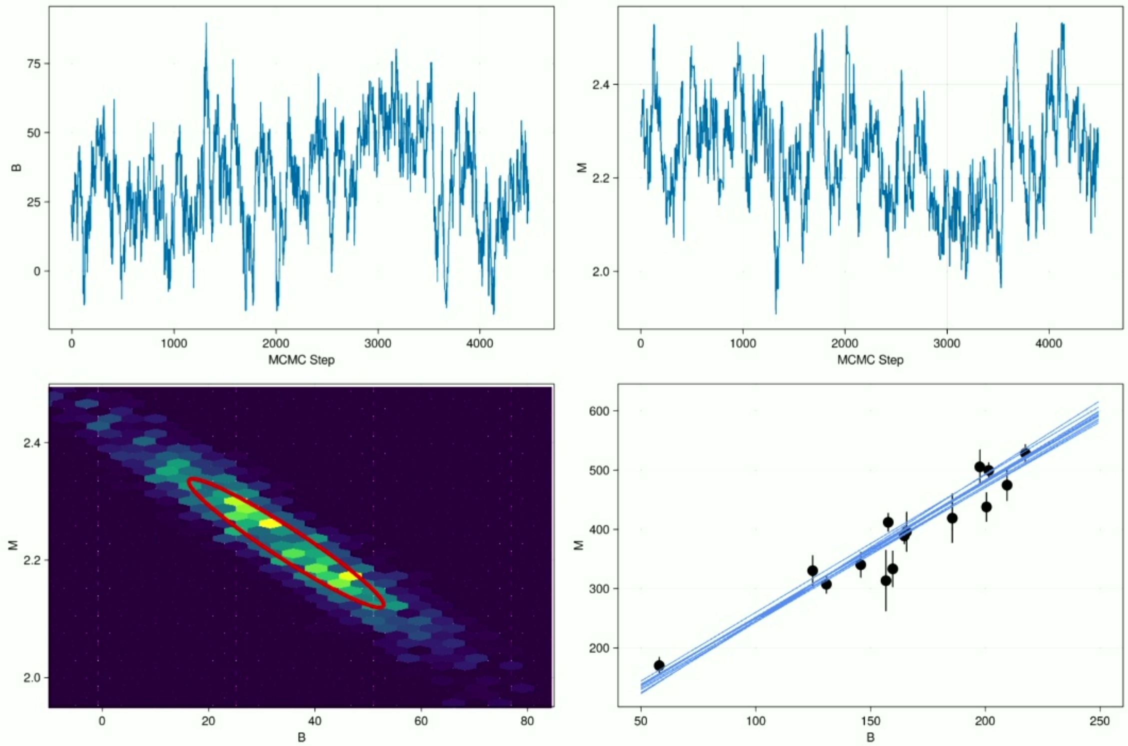
Borrowing heavily from Dan Foreman-Mackey's slides
https://speakerdeck.com/dfm/data-analysis-with-mcmc1
And Jonathan Pritchard's slides
https://www.imperial.ac.uk/media/imperial-college/research-centres-

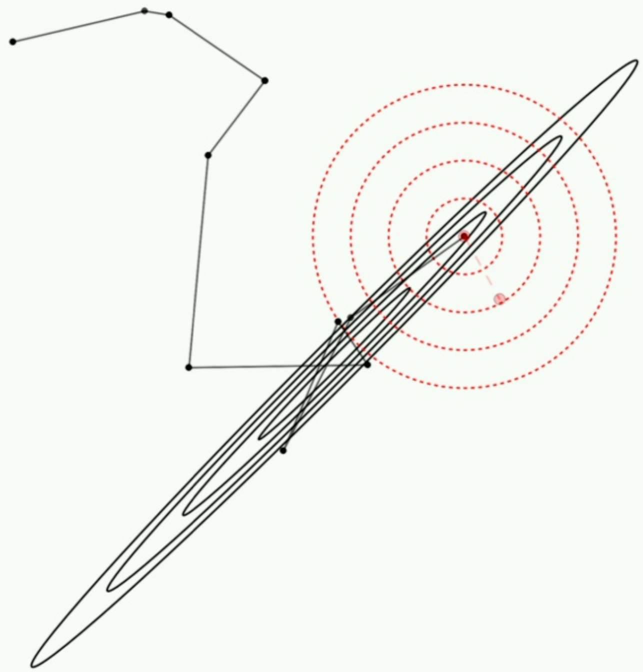
These slides are available at
https://github.com/dstndstn/MCMC-talk

The MCMC Algorithm

```
function mcmc(logprob_func, logprob_args,
              propose_func, propose_args,
              initial_pos, nsteps)
    p = initial_pos
    logprob = logprob_func(p, logprob_args)
    chain = zeros(Float64, (nsteps, length(p)))
    naccept = 0
    for i in 1:nsteps
        # propose a new position in parameter space
        p_new = propose_func(p, propose_args)
        # compute probability at new position
        logprob_new = logprob_func(p_new, logprob_args)
        # decide whether to jump to the new position
        if exp(logprob_new - logprob) > rand()
            p = p_new
            logprob = logprob_new
            naccept += 1
        end
        # save the position
        chain[i,:] = p
    end
    return chain, naccept/nsteps
end
```

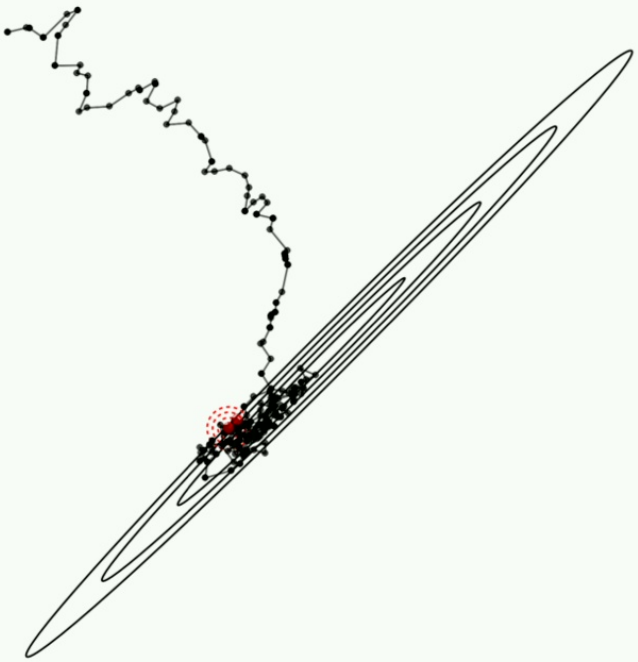
MCMC for model parameter inference





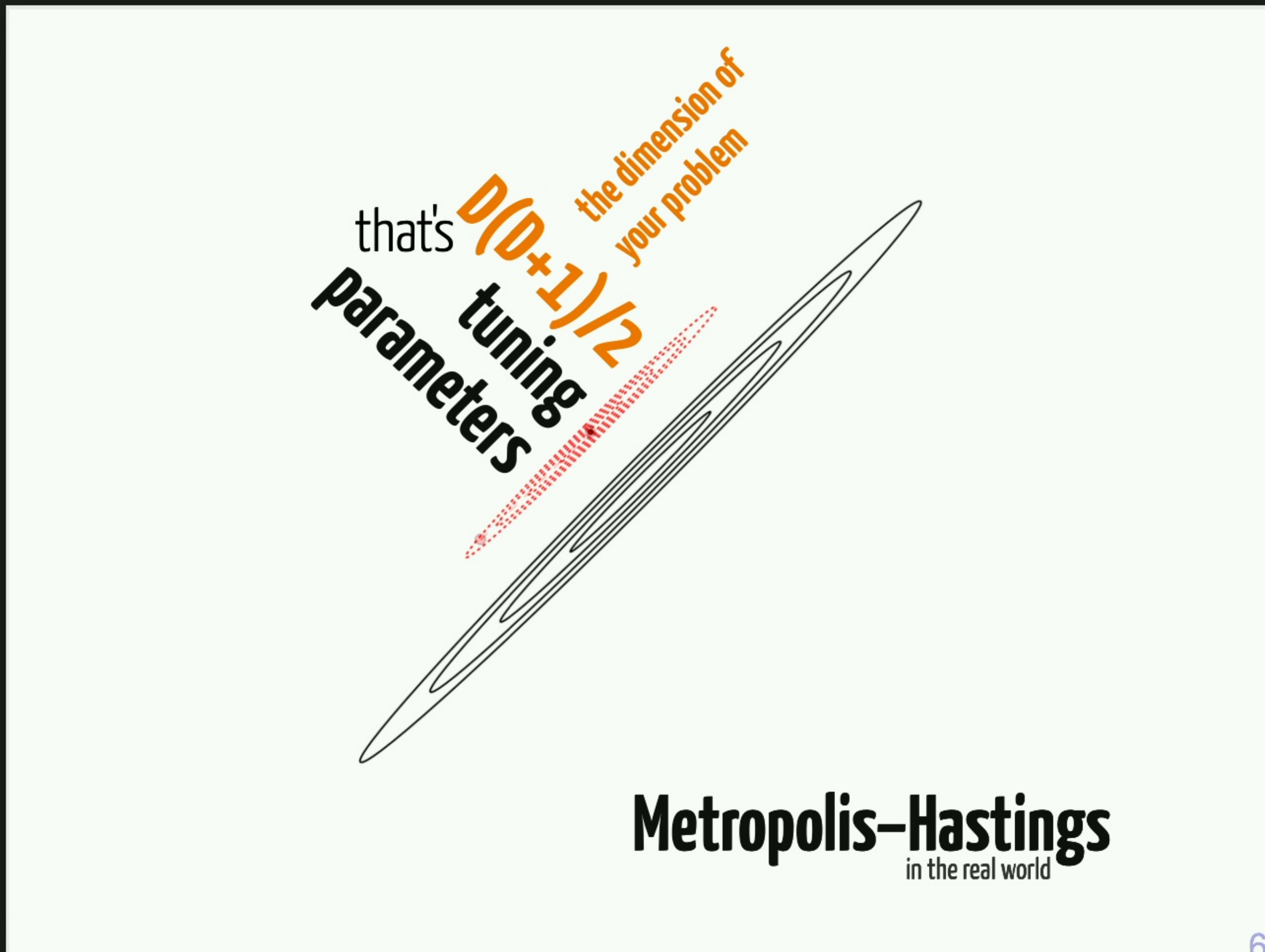
Metropolis–Hastings
in the real world

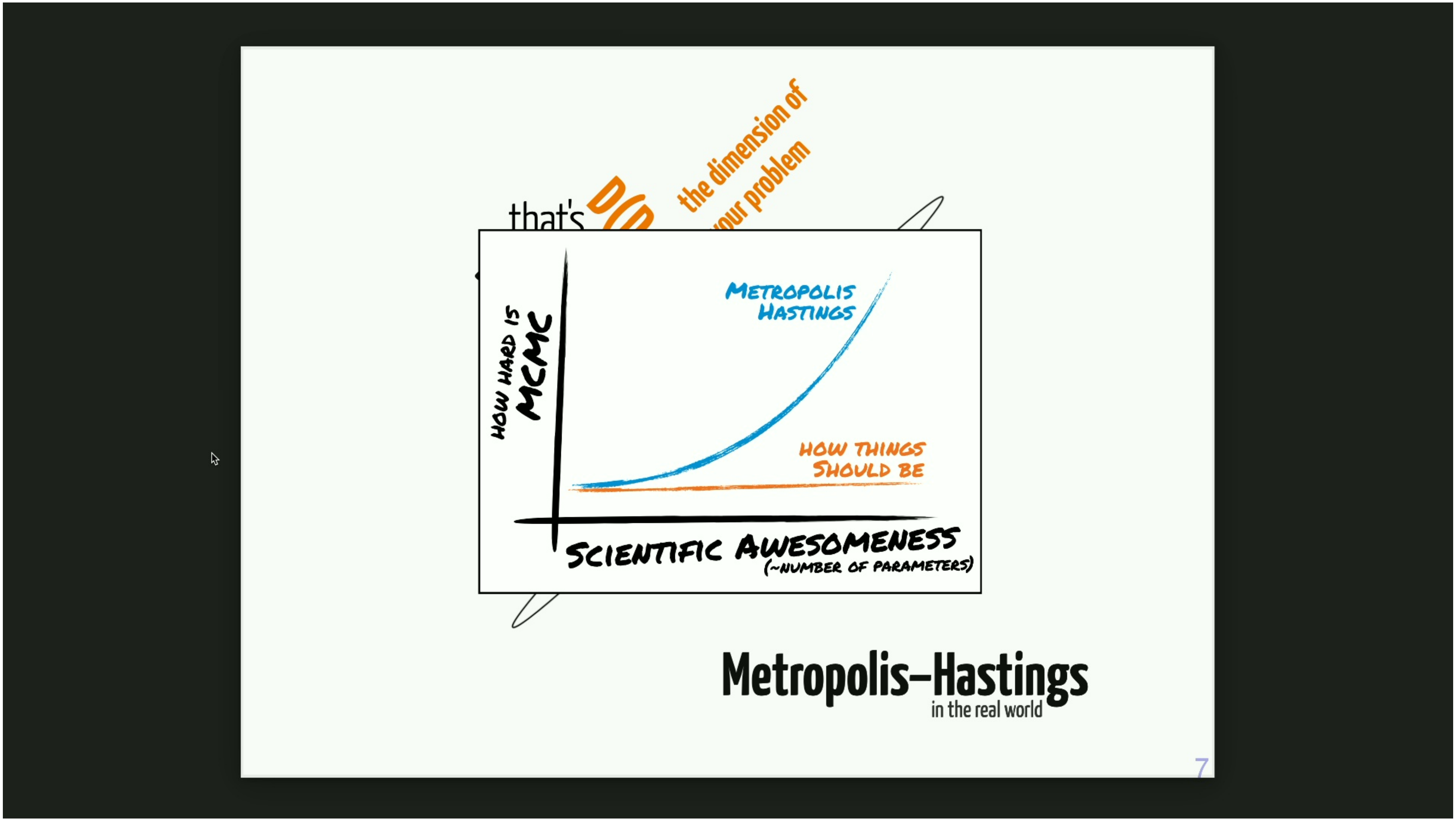
4



Metropolis–Hastings
in the real world

5


that's
D(D+1)/2
the dimension of
your problem
tuning
parameters
Metropolis–Hastings
in the real world
6


that's
the dimension of
METROPOLIS HASTINGS
HOW HARD IS MCMC
HOW THINGS SHOULD BE
SCIENTIFIC AWESOMENESS
(~NUMBER OF PARAMETERS)
Metropolis–Hastings
in the real world
7

Jonathan Goodman

Jonathan Weare

"Ensemble samplers with affine invariance"

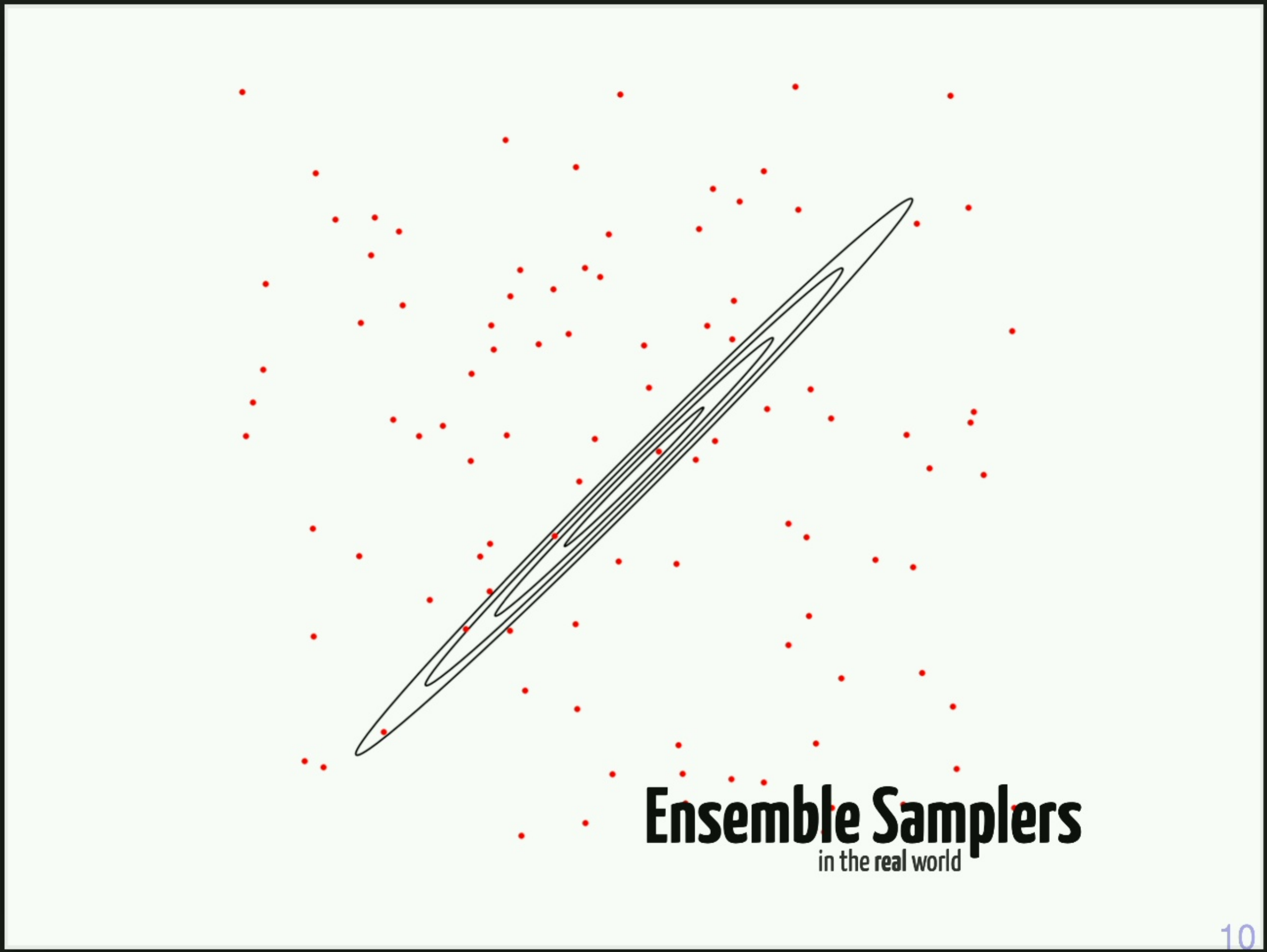
(dfm.io/mcmc-gw10)

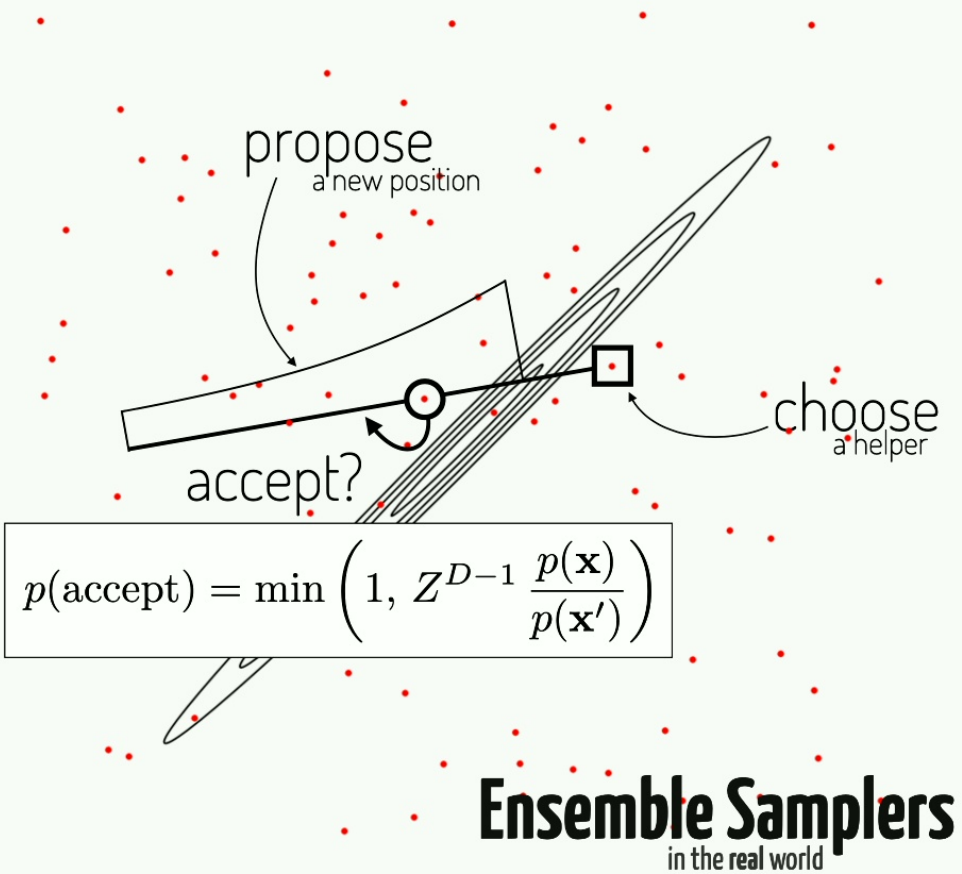
introducing **emcee** the MCMC Hammer

arxiv.org/abs/1202.3665

dan.iel.fm/emcee

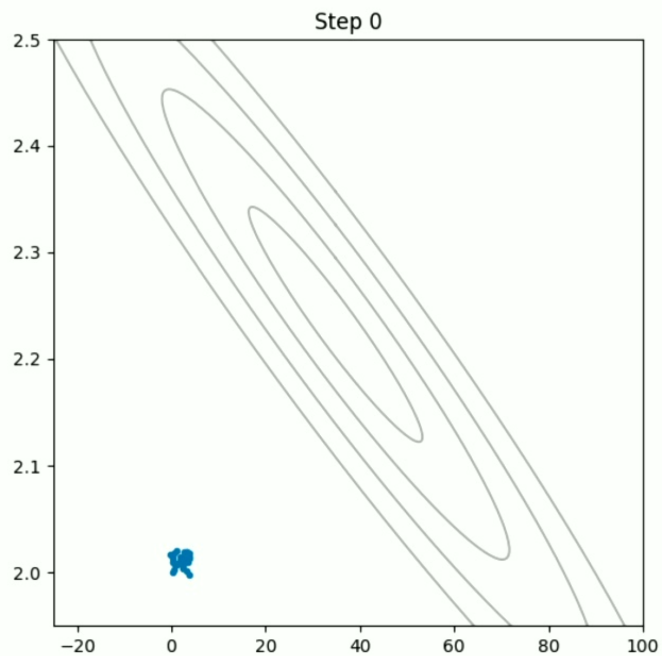
9


Ensemble Samplers
in the real world
10



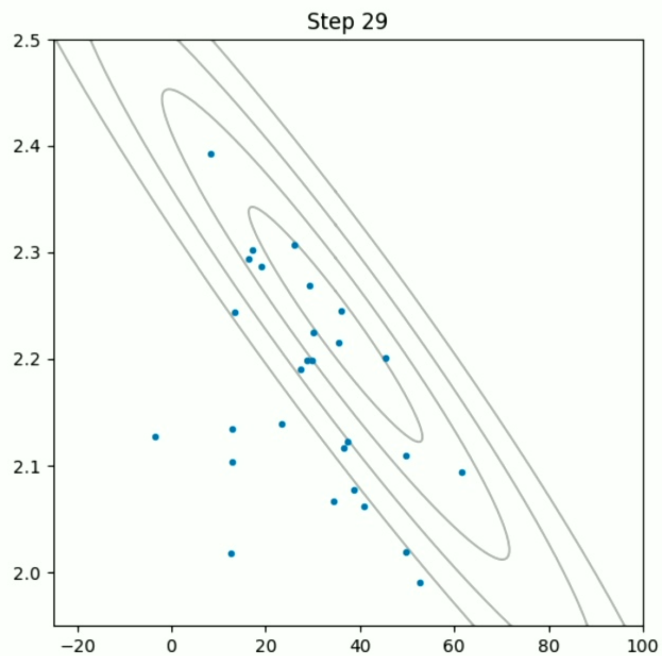
$$p(\text{accept}) = \min\left(1,\ Z^{D-1}\,\frac{p(\mathbf{x})}{p(\mathbf{x}')}\right)$$

Emcee demo

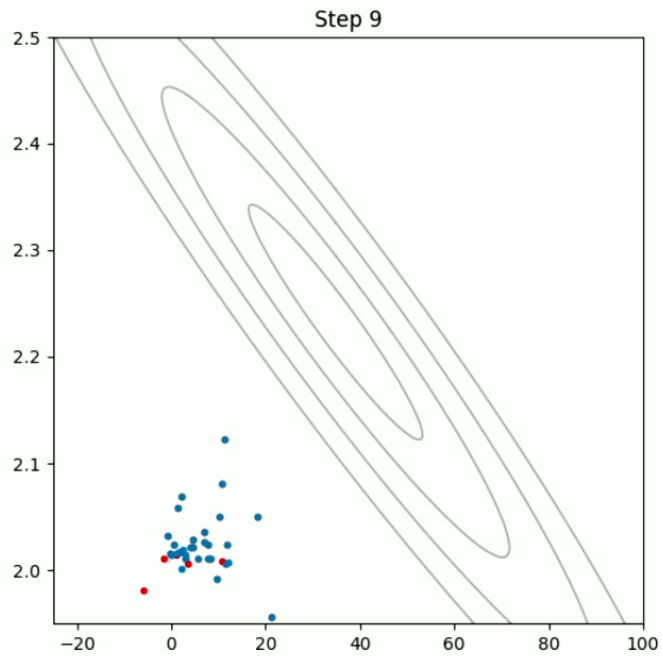


16

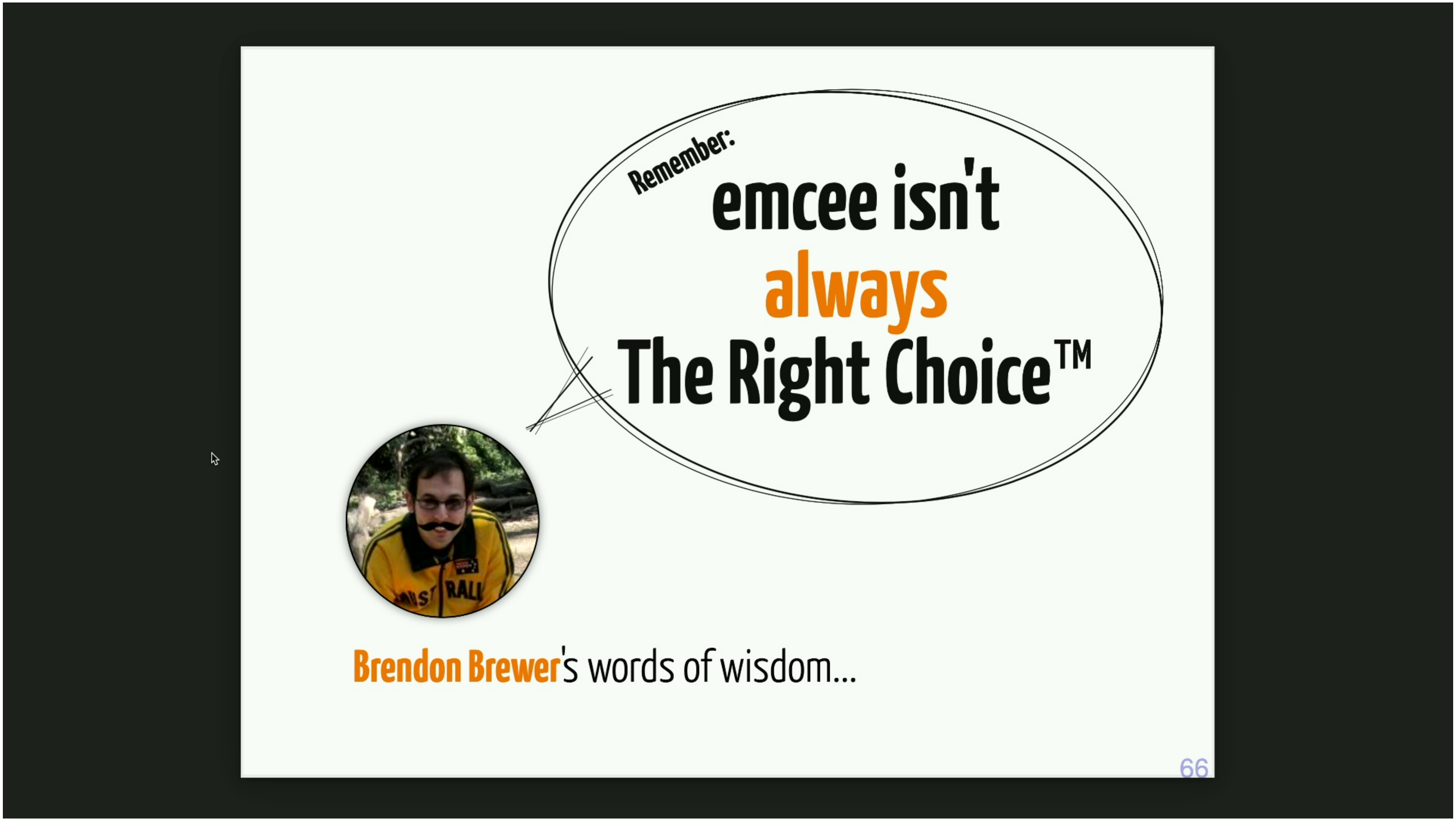
Emcee demo



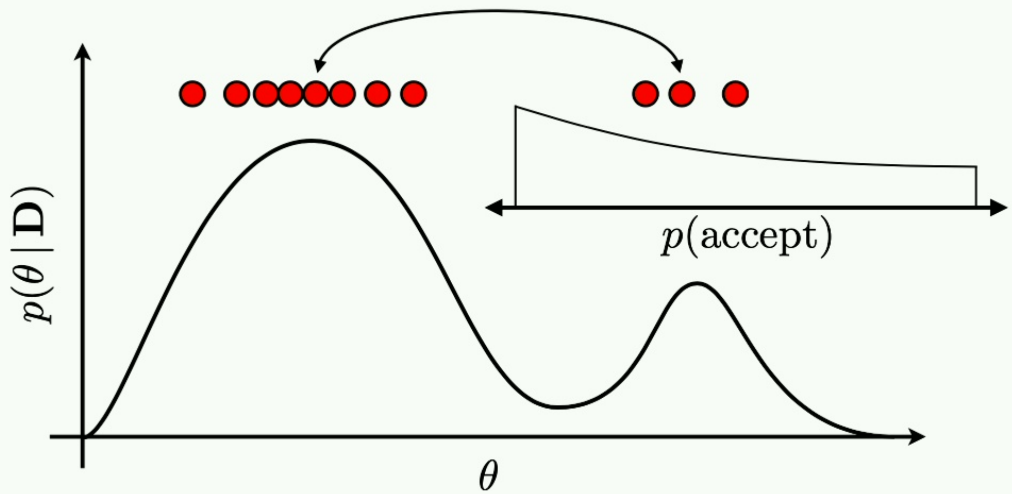
Emcee demo



65


Remember:
emcee isn't
always
The Right Choice™
Brendon Brewer's words of wisdom...
66

what about multimodal densities?



wh… es?

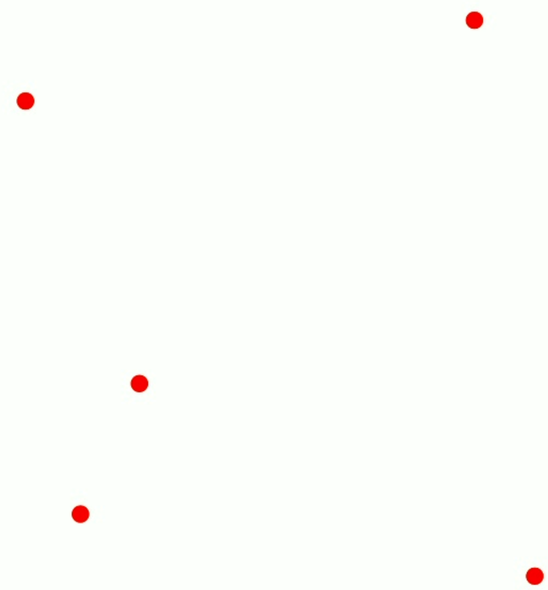
$p(\theta\,|\,\mathbf{D})$

θ

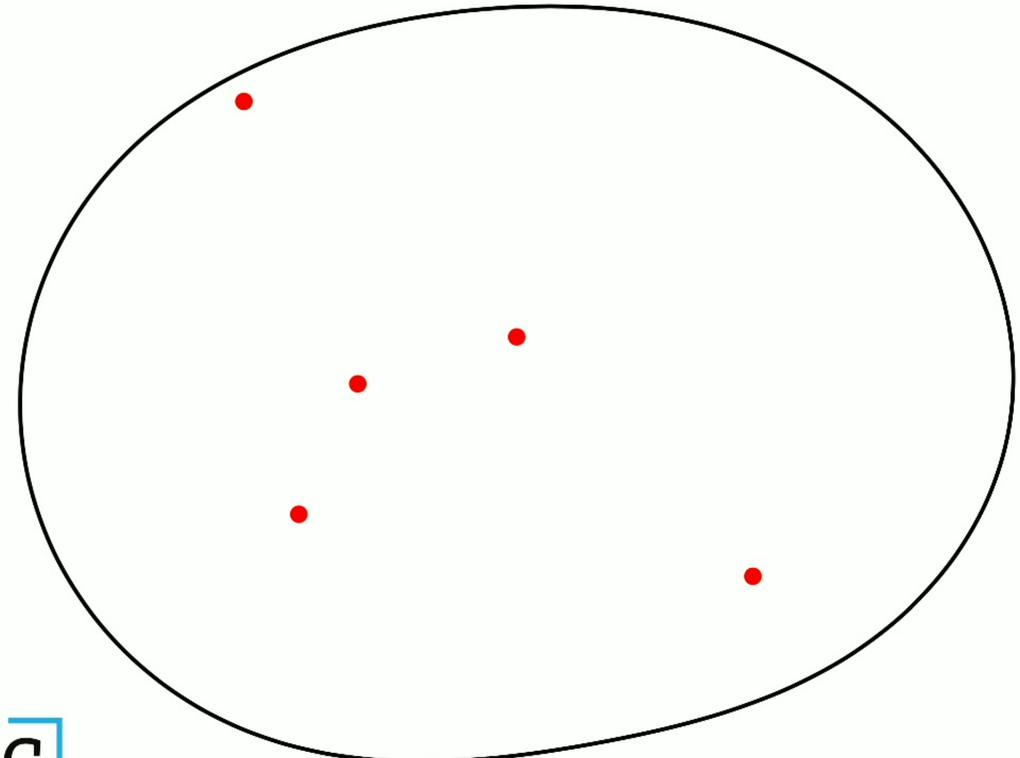
Nested Sampling

- An alternative to MCMC (and emcee)
- (Originally built for integrating the *evidence*)
- Idea: successively resample points, demanding they be above a minimum likelihood level
- Think of slowly raising the water level to find terrain contours

five live points chosen uniformly from prior



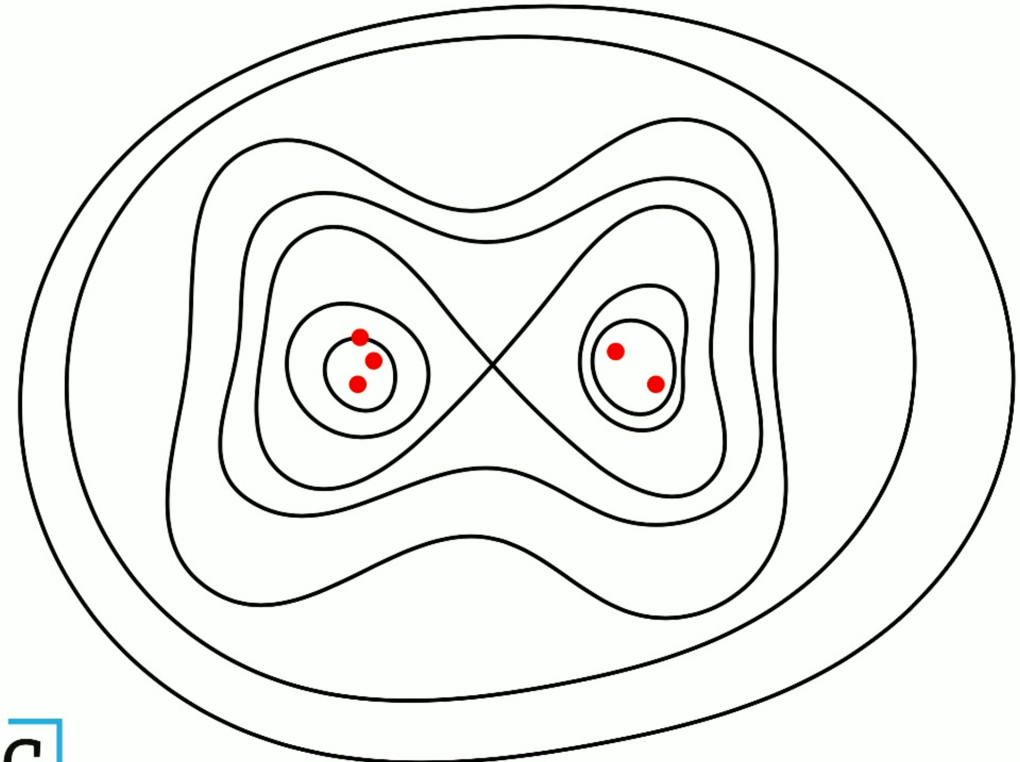
Select a new point uniformly sampled subject to requirement $L(X_{new})>L(X_{old})$



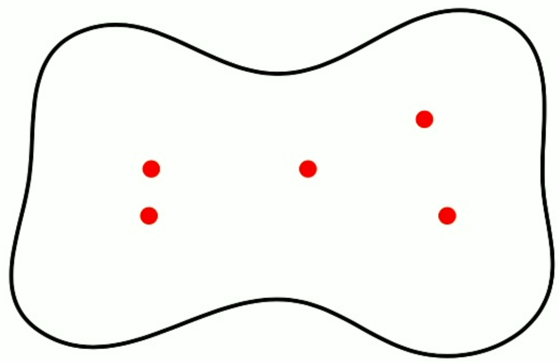
ICIC

76

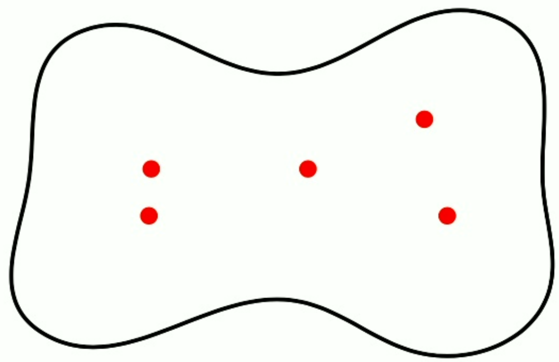
Successive points map likelihood in ordered way



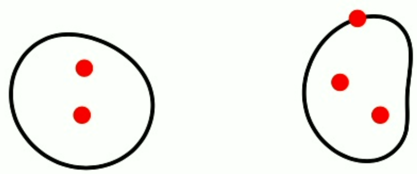
ICIC



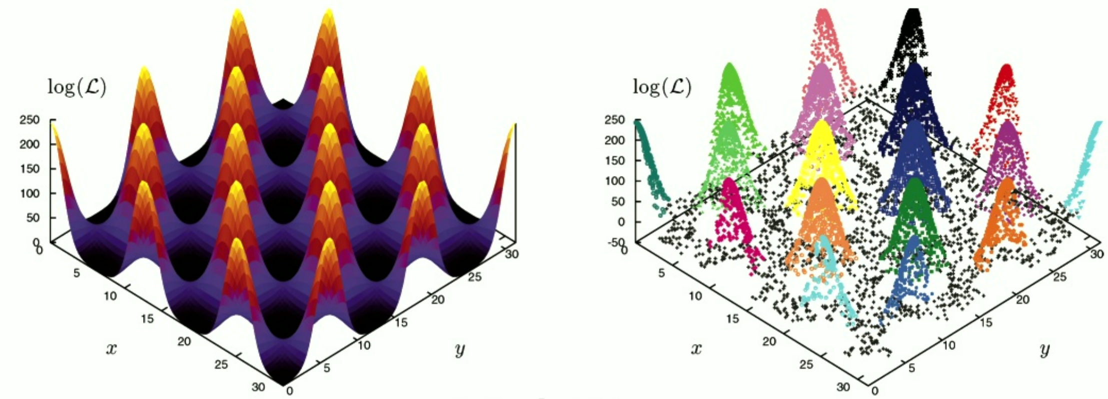
85



85

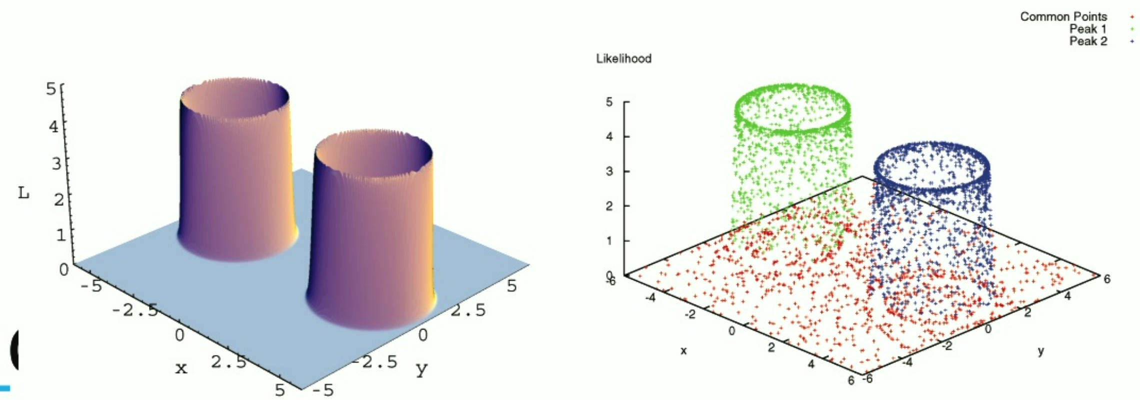


89



(a) (b)

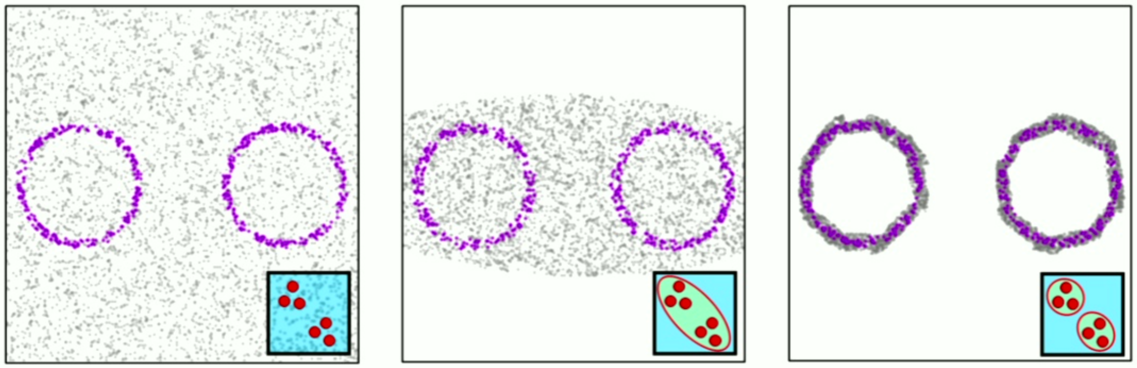
MultiNest



(a) (b)

94

Dynamic Nested Sampling



Unit Cube (no bound) | Single Ellipsoid | Multiple Ellipsoids

95

dynesty
stable

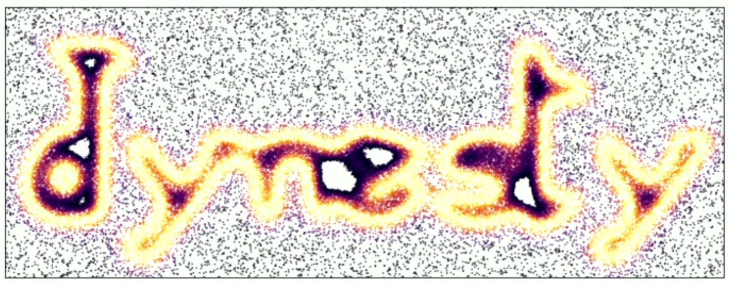
Search docs

- Crash Course
- Background
- Getting Started
- Dynamic Nested Sampling with dynesty
- Nested Sampling Errors
- Examples
- FAQ
- References and Acknowledgements
- API

Read the Docs v: stable

Docs » dynesty Edit on GitHub

dynesty



`dynesty` is a Pure Python, MIT-licensed Dynamic Nested Sampling package for estimating Bayesian posteriors and evidences. See Crash Course and Getting Started for more information. The latest development version can be found here.

The release paper describing dynesty 1.0 can be found here.

As a multi-purpose sampler, `dynesty` is designed to perform "reasonably well" across a large array of problems but is not optimized for any single one. In particular, please take caution when applying `dynesty` to estimate Bayesian posteriors and evidences for large-dimensional (>30 dimensions or so) problems.

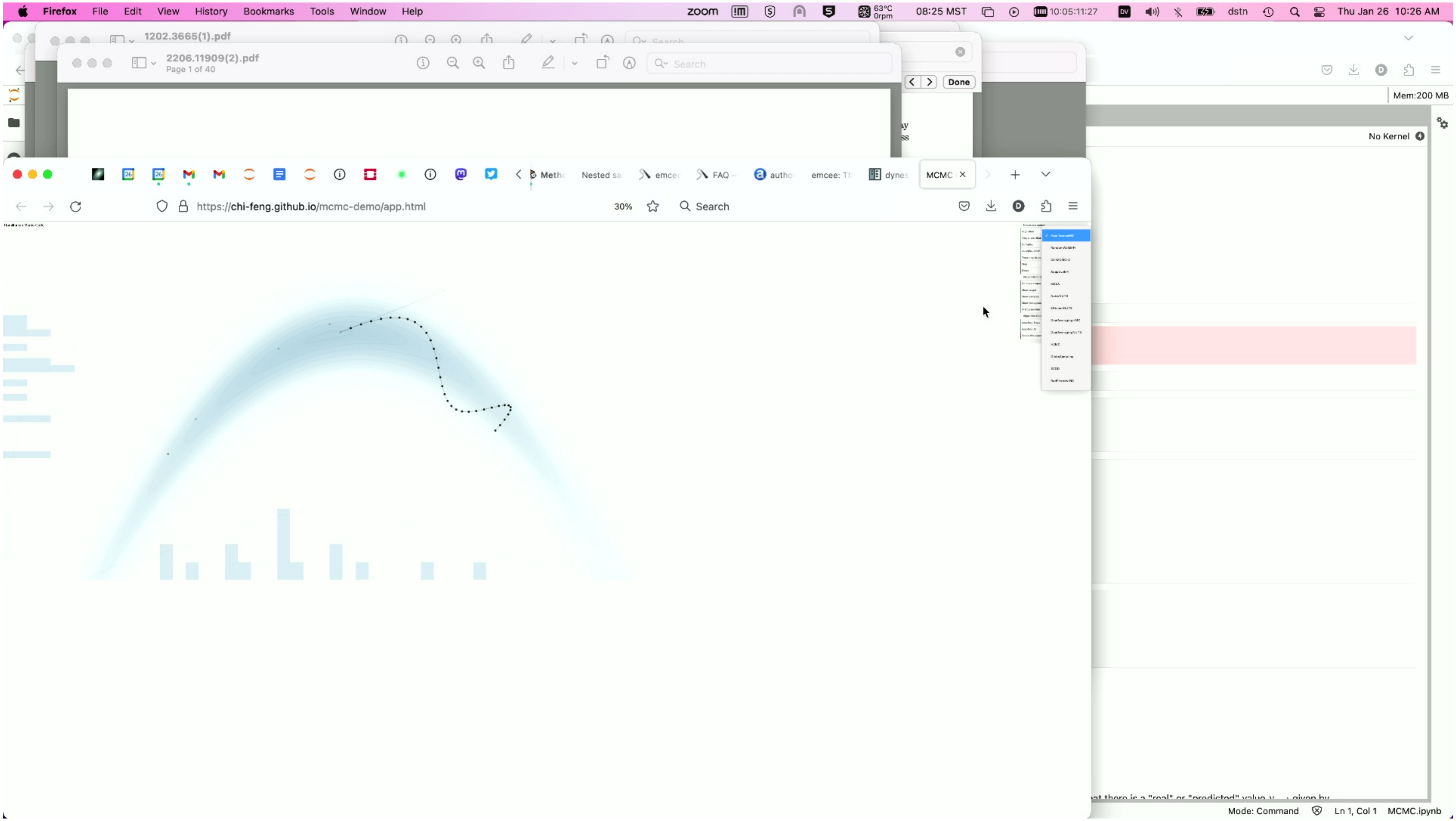
Installation

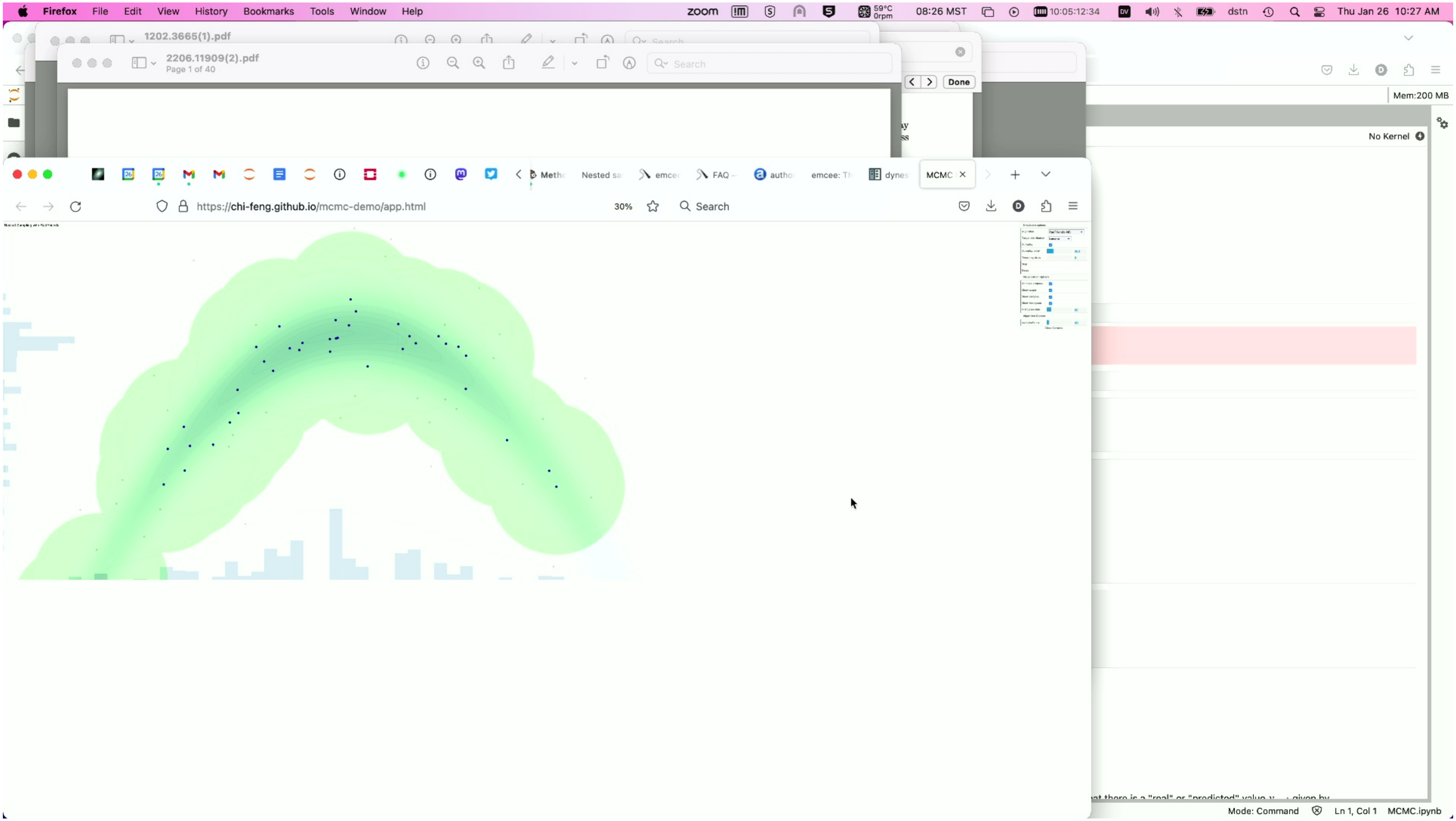
`dynesty` is compatible wit Python 3.6+. It requires `numpy` (for arithmetic), `scipy` (for special

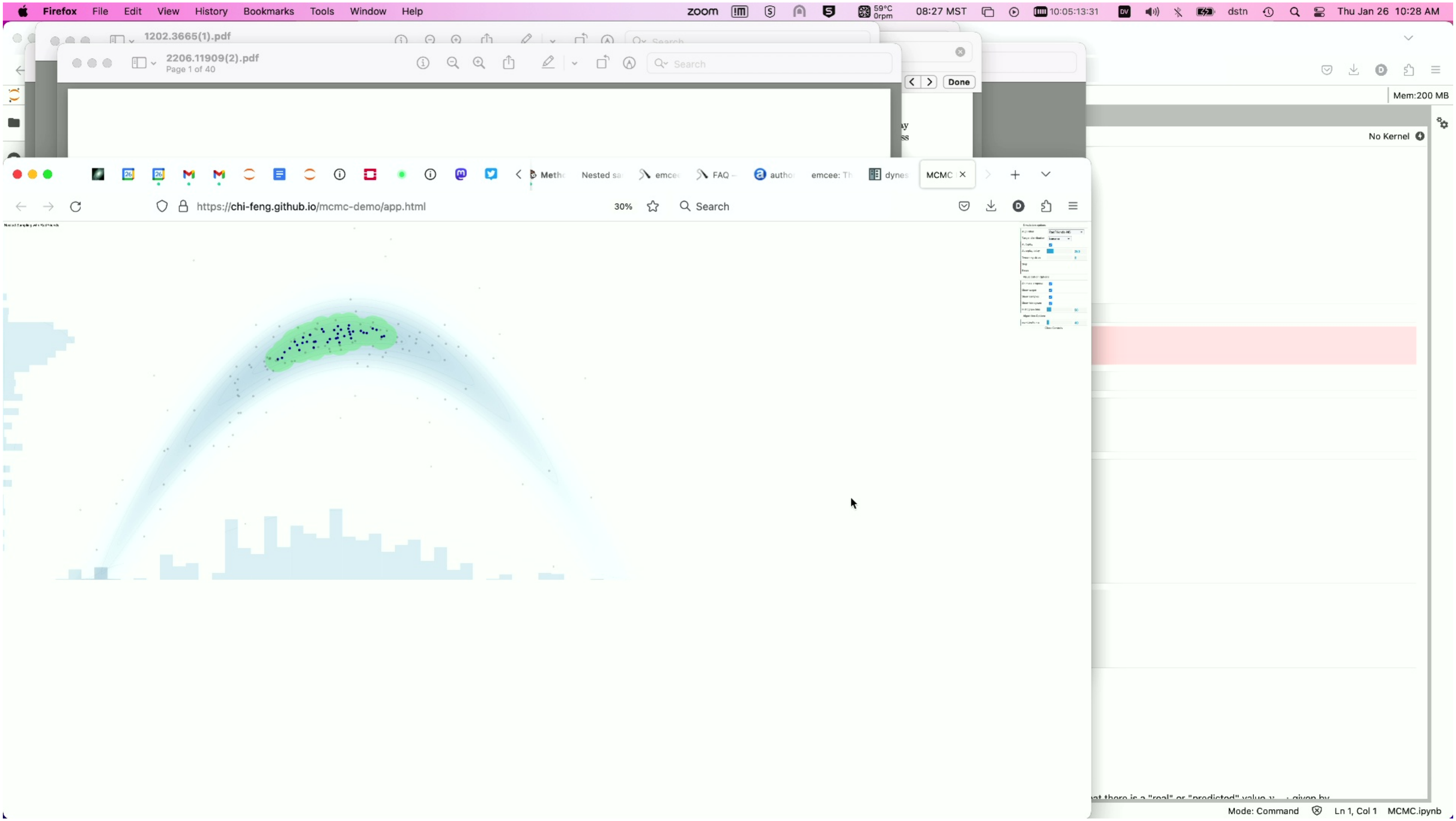
Real. Fake. Data.
Safe Synthetic Data Modeled After Yours
Scalable Fake Data
Learn More
Ad by EthicalAds

Dynamic Nested Sampling

https://dynesty.readthedocs.io/en/stable/
https://chi-feng.github.io/mcmc-demo/app.html


Firefox
File
Edit
View
History
Bookmarks
Tools
Window
Help
zoom
08:25 MST
10:05:11:27
dstn
Thu Jan 26 10:26 AM
1202.3665(1).pdf
2206.11909(2).pdf
Page 1 of 40
Search
Done
Mem:200 MB
No Kernel
MCMC
https://chi-feng.github.io/mcmc-demo/app.html
30%
Search
Mode: Command
Ln 1, Col 1
MCMC.ipynb


Firefox File Edit View History Bookmarks Tools Window Help
1202.3665(1).pdf
2206.11909(2).pdf
Page 1 of 40
https://chi-feng.github.io/mcmc-demo/app.html
Mem:200 MB
No Kernel
Mode: Command
Ln 1, Col 1
MCMC.ipynb


Firefox File Edit View History Bookmarks Tools Window Help
zoom
08:27 MST
Thu Jan 26 10:28 AM
1202.3665(1).pdf
2206.11909(2).pdf
Page 1 of 40
Done
Mem:200 MB
No Kernel
Nested sa
FAQ
MCMC
https://chi-feng.github.io/mcmc-demo/app.html
30%
Search
Mode: Command
Ln 1, Col 1
MCMC.ipynb

Dynamic Nested Sampling

https://dynesty.readthedocs.io/en/stable/
https://chi-feng.github.io/mcmc-demo/app.html

Emcee demo