Statistics in Cosmology

Day 2 – Inference

11th TRR33 Winter School in Cosmology
10-16 December 2017, Passo del Tonale - Italy

Emille E. O. Ishida

Laboratoire de Physique de Clermont - Université Clermont-Auvergne
Clermont Ferrand, France

Conversations in a snowing afternoon...

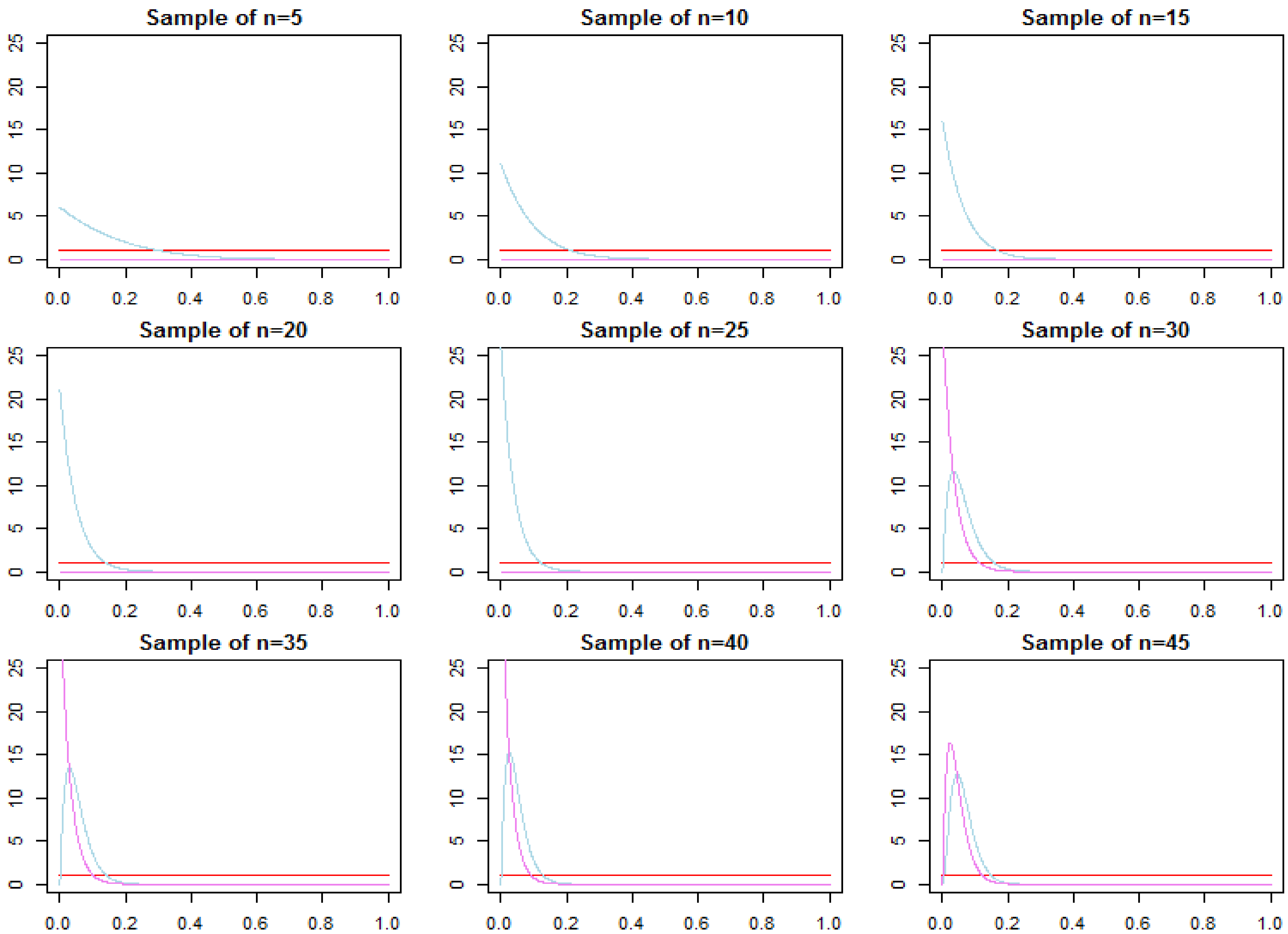
How **NOT** to choose your priors?

- Avoid step functions
- Avoid improper priors

$$P(a, b|\vec{x}, \vec{y}, \sigma) = \frac{\mathcal{L}(\vec{x}, \vec{y}|a, b, \sigma)p(a)p(b)}{\int_a \int_b \mathcal{L}(\vec{x}, \vec{y}|a, b, \sigma)p(a)p(b)dadb}$$

How the prior influence the results?

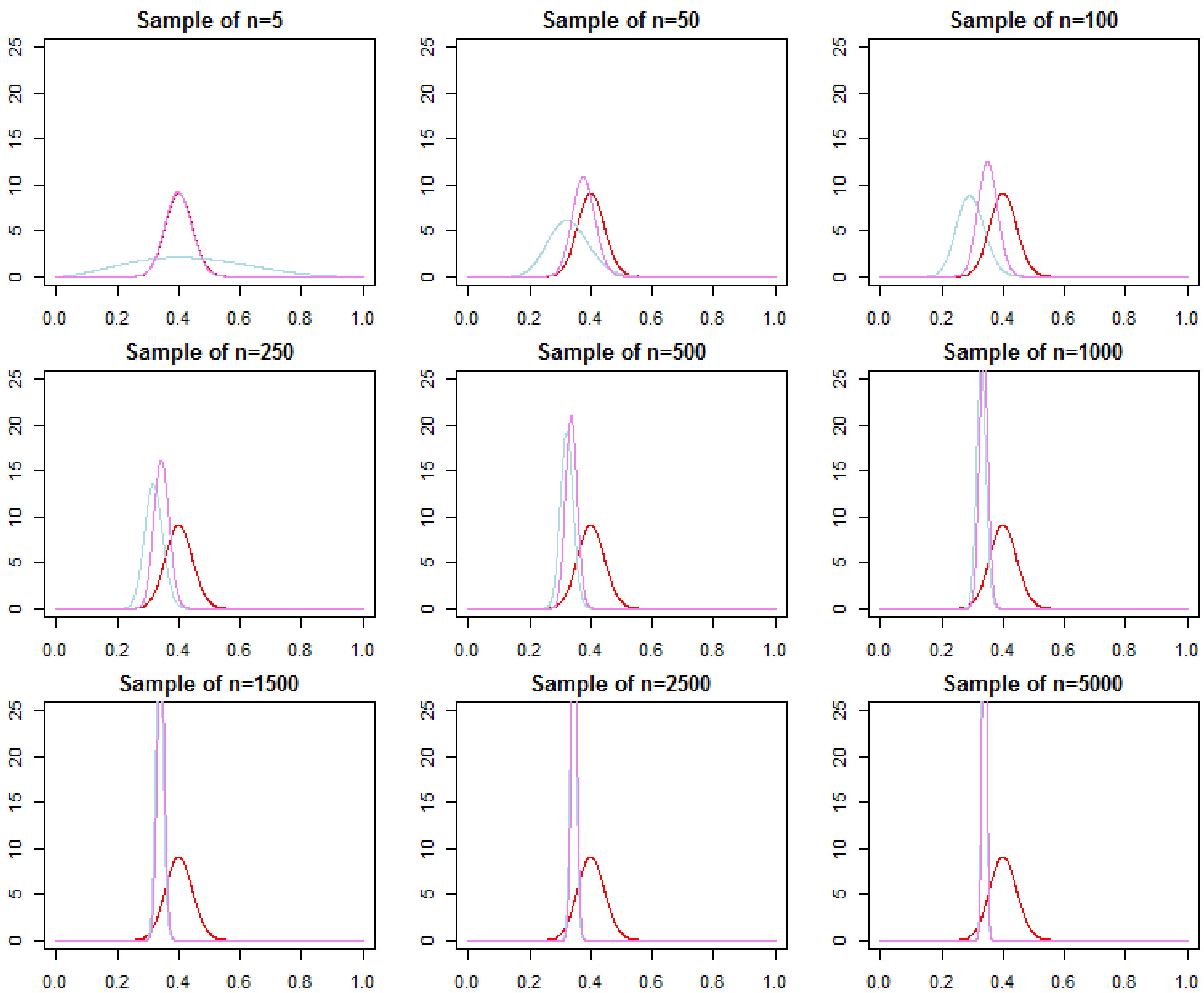
Prior, likelihood, posterior



From StackOverflow

How the prior influence the results?

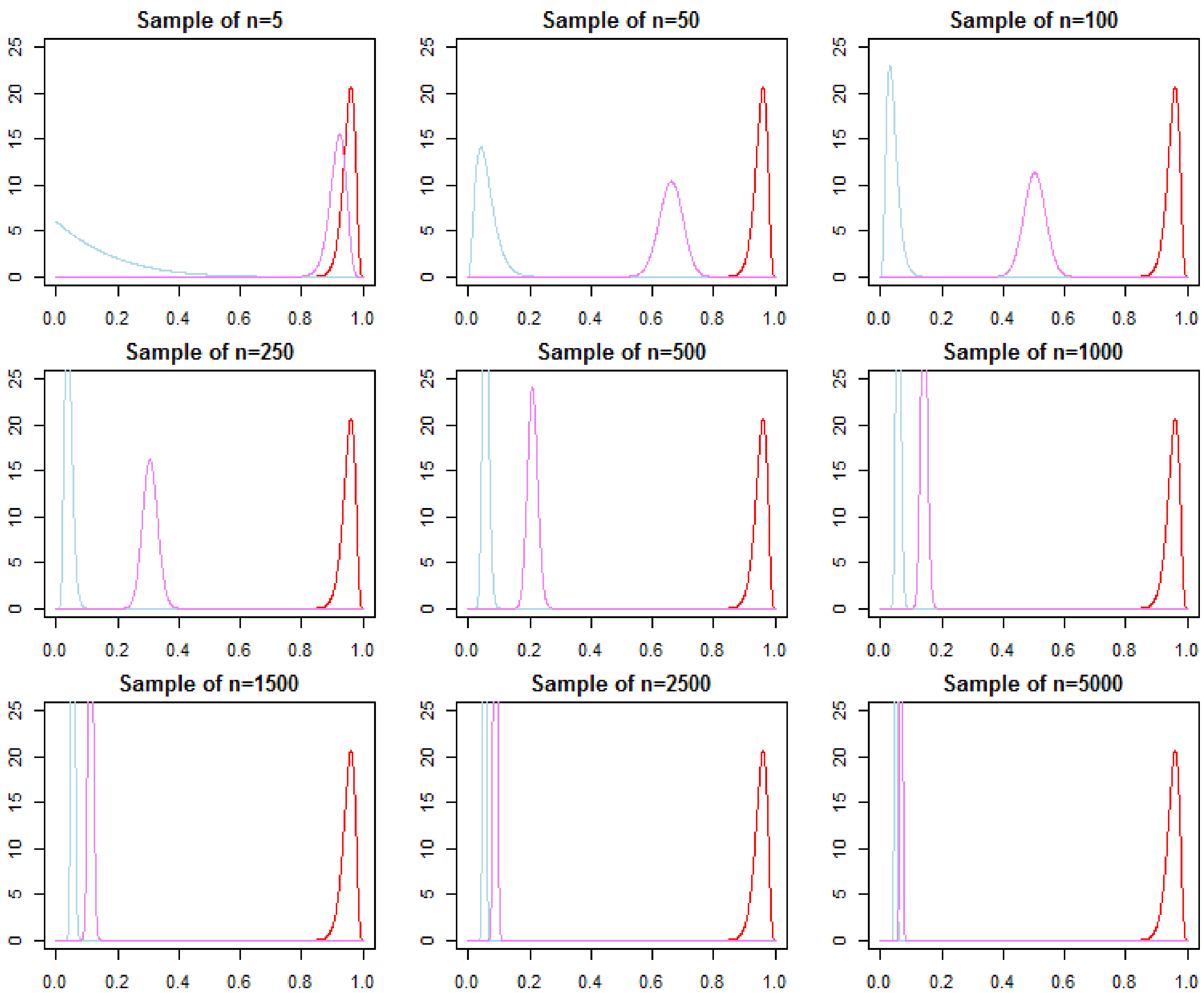
Prior, likelihood, posterior



From StackOverflow

How the prior influence the results?

Prior, likelihood, posterior



From StackOverflow

The most important slide from yesterday

Frequentist:

Frequency

Bayesian:

State of knowledge

Or

Degree of belief

Goals of parameter estimation

(i) Parameter values

(ii) Error estimates on parameters

(iii) Goodness of fit

"Unfortunately, many practitioners of parameter estimation never proceed beyond item (i). They deem a fit acceptable if a graph of data and model "looks good." This approach is known as chi2-by-eye. Luckily, its practitioners get what they deserve."

W. H. Press et al., Numerical Recipes in C, Cambridge University Press, 2002

y = a x + b

Frequentist:

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

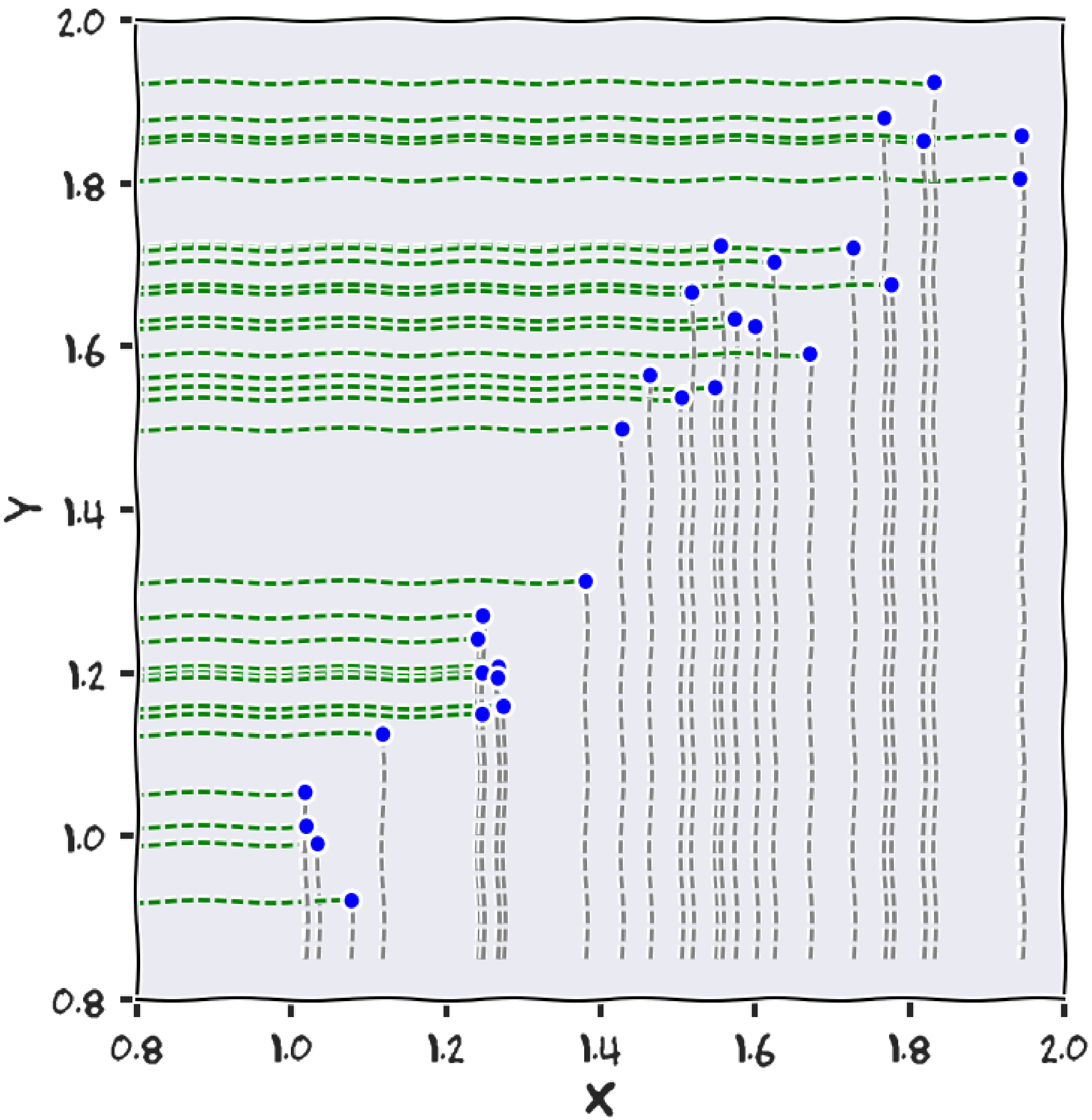
Bayesian:

$$P(a,b|\vec{x},\vec{y},\sigma) = \frac{\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)}{\int_a\int_b \mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)dadb}$$

How to estimate parameter values?

Least square fitting:

Model: $y = ax + b$



$y = f(x;\, a,b)$

Given {a,b}, $\min \left[\sum_{i=1}^{N} (y_i - f(x_i; a, b))^2 \right]$

Given a particular set of parameters, what is the probability that this data set could have occurred?

$$\mathcal{L}(\vec{x}, \vec{y} | a, b, \sigma)$$

Maximum likelihood estimation (MLE):

$$\mathcal{L}(\vec{x}, \vec{y}|a, b, \sigma) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N} \frac{(y_i - (ax_i + b))^2}{\sigma^2}\right]$$

$$\hat{\theta} = \{a, b\} \longrightarrow \min\left[\sum_{i=1}^{N} \frac{(y_i - (ax_i + b))^2}{\sigma^2}\right]$$

Maximum likelihood estimation (MLE):

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

$$\hat{\theta} = \{a,b\} \longrightarrow \min\left[\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

as σ=cte ...

$$\boxed{\hat{\theta} = \{a,b\} \longrightarrow \min\left[\sum_{i=1}^{N}(y_i-f(x_i;a,b))^2\right]}$$

Least square ↔ Maximum Likelihood
if the uncertainties are:

- independent
- normally distributed
- constant standard deviation

Maximum likelihood estimation (MLE):

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

$$\hat{\theta} = \{\hat{a},\hat{b}\} \longleftarrow \max\left[\ln\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)\right]$$

$$\left.\frac{\partial \ln\mathcal{L}}{\partial a}\right|_{\hat{\theta}} = \left.\frac{\partial \ln\mathcal{L}}{\partial b}\right|_{\hat{\theta}} = 0$$

(i) point

Taylor series expansion around maximum:

$$\ln\mathcal{L}(\theta) = \ln\mathcal{L}|_{\hat{\theta}} + (\theta-\hat{\theta})\left.\frac{\partial \ln\mathcal{L}}{\partial\theta}\right|_{\hat{\theta}} + \frac{1}{2}(\theta-\hat{\theta})^2\left.\frac{\partial^2 \ln\mathcal{L}}{\partial\theta^2}\right|_{\hat{\theta}} + \dots$$

Maximum likelihood estimation (MLE):

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

$$\hat{\theta} = \{\hat{a},\hat{b}\} \longleftarrow \max\left[\ln\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)\right]$$

$$\left.\frac{\partial \ln\mathcal{L}}{\partial a}\right|_{\hat{\theta}} = \left.\frac{\partial \ln\mathcal{L}}{\partial b}\right|_{\hat{\theta}} = 0 \longleftarrow \text{(i) point}$$

Taylor series expansion around maximum:

$$\ln\mathcal{L}(\theta) = \ln\mathcal{L}|_{\hat{\theta}} + \cancel{(\theta-\hat{\theta})\left.\frac{\partial \ln\mathcal{L}}{\partial\theta}\right|_{\hat{\theta}}} + \frac{1}{2}(\theta-\hat{\theta})^2\left.\frac{\partial^2 \ln\mathcal{L}}{\partial\theta^2}\right|_{\hat{\theta}} + \ldots$$

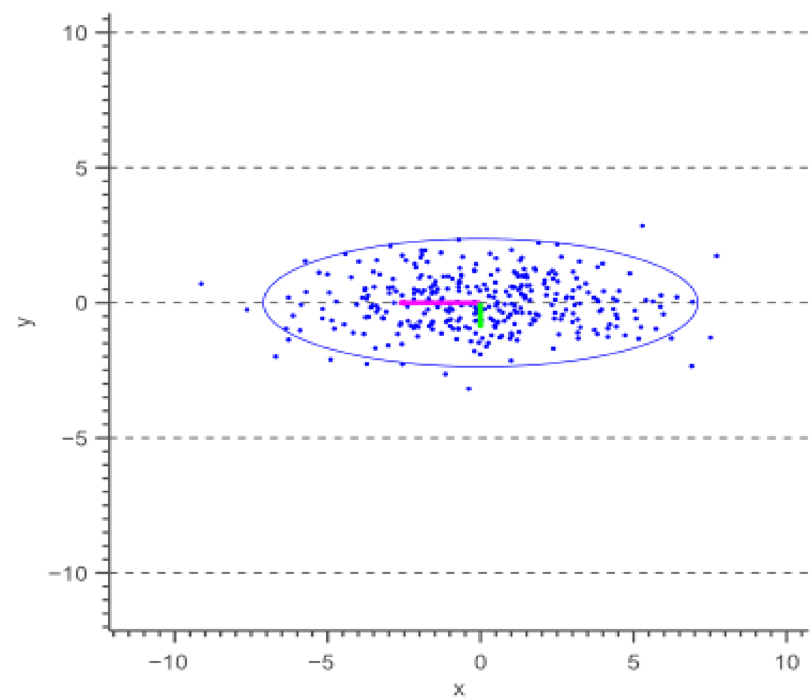
$$\mathcal{L}(\theta) \approx \mathcal{L}(\hat{\theta})\exp\left(-\frac{1}{2}\frac{(\theta-\hat{\theta})^2}{C_{\hat{\theta}}}\right)$$

$$C_{\hat{\theta}} = F^{-1} = \left(-\left.\frac{\partial^2 \ln\mathcal{L}}{\partial\theta^2}\right|_{\hat{\theta}}\right)^{-1}$$

To the extend that:

- $\hat{\theta}$ is an unbiased estimator (N is sufficiently large) and
- this second order approximation is sufficiently accurate:

$$Var(\theta_i) \geq (F_{ii})^{-1}$$



Maximum likelihood estimation (MLE):

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

$$\hat{\theta} = \{\hat{a},\hat{b}\} \longleftarrow \max\left[\ln\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)\right]$$

$$\left.\frac{\partial \ln\mathcal{L}}{\partial a}\right|_{\hat{\theta}} = \left.\frac{\partial \ln\mathcal{L}}{\partial b}\right|_{\hat{\theta}} = 0 \longleftarrow$$ (i) point

Taylor series expansion around maximum:

$$\ln\mathcal{L}(\theta) = \ln\mathcal{L}|_{\hat{\theta}} + \cancel{(\theta-\hat{\theta})\left.\frac{\partial \ln\mathcal{L}}{\partial\theta}\right|_{\hat{\theta}}} + \frac{1}{2}(\theta-\hat{\theta})^2\left.\frac{\partial^2 \ln\mathcal{L}}{\partial\theta^2}\right|_{\hat{\theta}} + \ldots$$

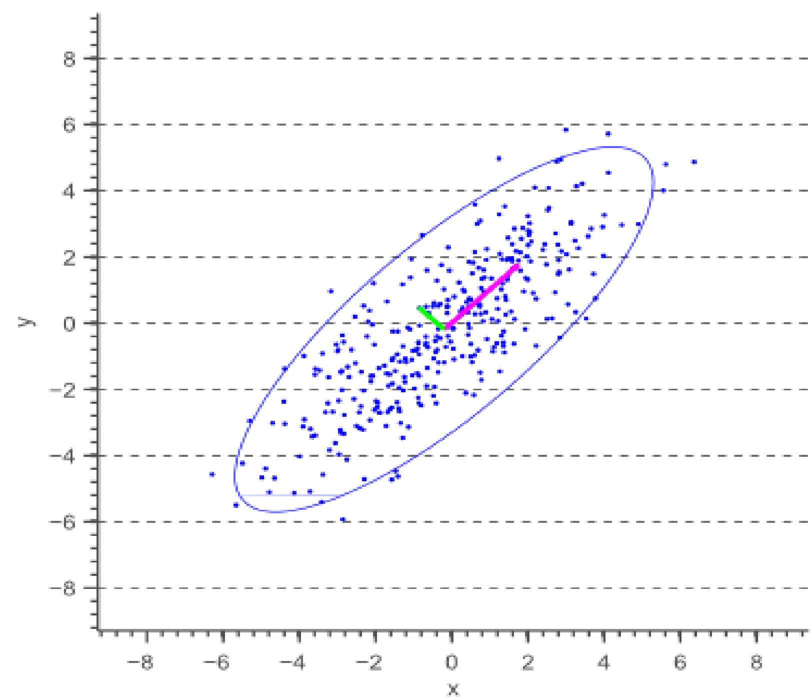
$$\mathcal{L}(\theta) \approx \mathcal{L}(\hat{\theta})\exp\left(-\frac{1}{2}\frac{(\theta-\hat{\theta})^2}{C_{\hat{\theta}}}\right)$$

$$C_{\hat{\theta}} = F^{-1} = \left(-\left.\frac{\partial^2 \ln\mathcal{L}}{\partial\theta^2}\right|_{\hat{\theta}}\right)^{-1}$$

To the extend that:

- $\hat{\theta}$ is an unbiased estimator (N is sufficiently large) and
- this second order approximation is sufficiently accurate:

$$Var(\theta_i) \geq (F_{ii})^{-1}$$



Maximum likelihood estimation (MLE):

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

$$\hat{\theta} = \{\hat{a},\hat{b}\} \longleftarrow \max\left[\ln \mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)\right]$$

$$\left.\frac{\partial \ln \mathcal{L}}{\partial a}\right|_{\hat{\theta}} = \left.\frac{\partial \ln \mathcal{L}}{\partial b}\right|_{\hat{\theta}} = 0 \longleftarrow$$

(i) point

Taylor series expansion around maximum:

$$\ln \mathcal{L}(\theta) = \ln \mathcal{L}|_{\hat{\theta}} + \cancel{(\theta-\hat{\theta})\left.\frac{\partial \ln \mathcal{L}}{\partial \theta}\right|_{\hat{\theta}}} + \frac{1}{2}(\theta-\hat{\theta})^2\left.\frac{\partial^2 \ln \mathcal{L}}{\partial \theta^2}\right|_{\hat{\theta}} + \ldots$$

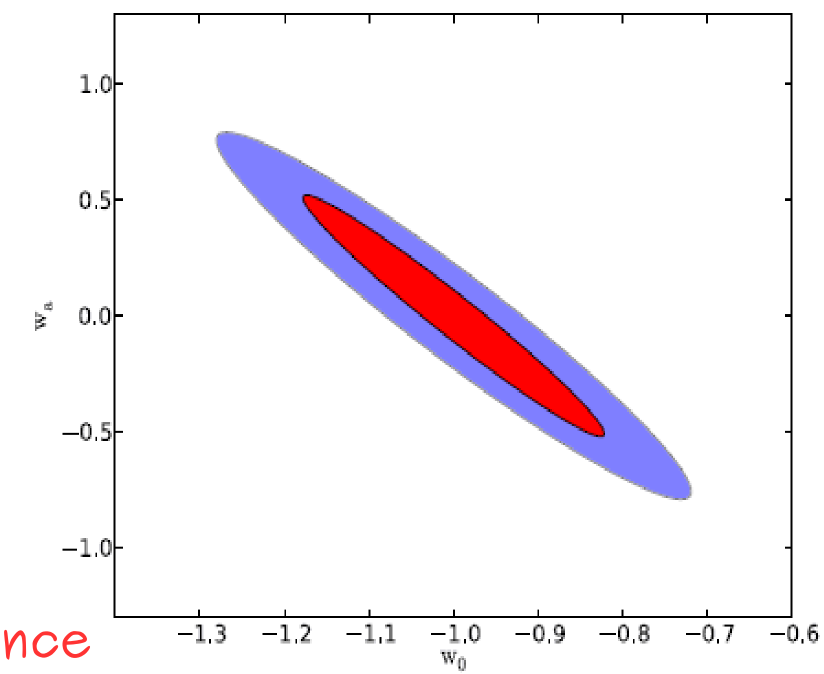
$$\mathcal{L}(\theta) \approx \mathcal{L}(\hat{\theta}) \exp\left(-\frac{1}{2}\frac{(\theta-\hat{\theta})^2}{C_{\hat{\theta}}}\right)$$

$$C_{\hat{\theta}} = F^{-1} = \left(-\left.\frac{\partial^2 \ln \mathcal{L}}{\partial \theta^2}\right|_{\hat{\theta}}\right)^{-1}$$

To the extend that:

- $\hat{\theta}$ is an unbiased estimator (N is sufficiently large) and
- this second order approximation is sufficiently accurate:

$$Var(\theta_i) \geq (F_{ii})^{-1}$$



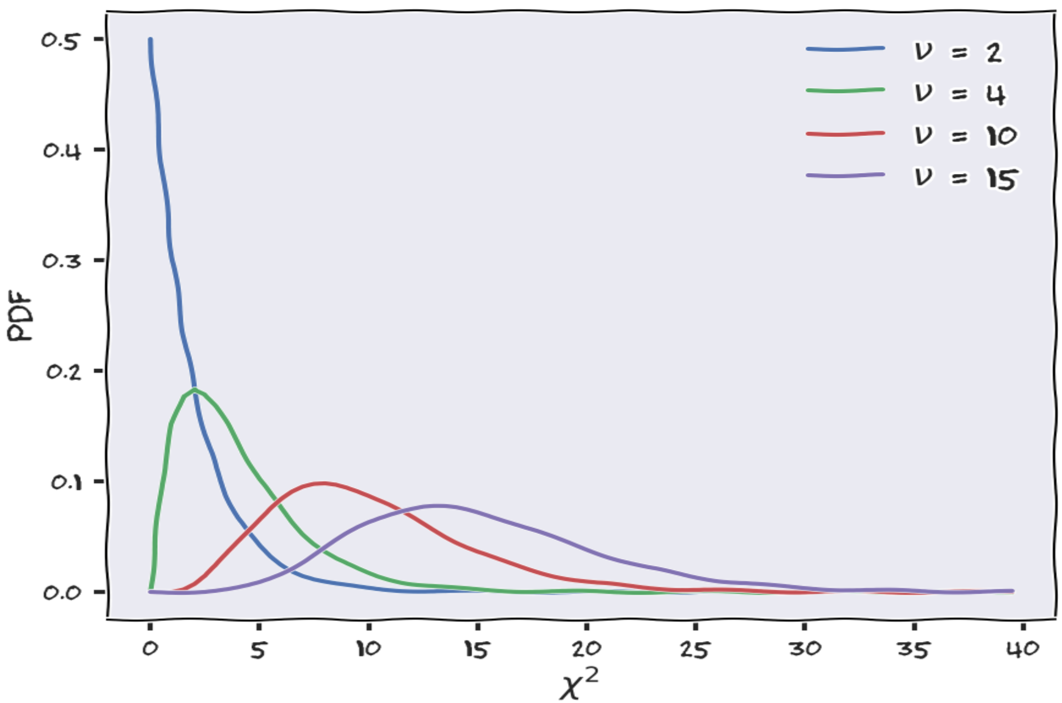
(ii) confidence

The χ^2 statistics: $\mathcal{D} = \{x_1, x_2, ..., x_\nu\}$ $x_i \sim \mathcal{N}(\mu_i, \sigma_i)$

The sum of squares ν independently distributed Gaussian random variables follows a χ^2 distribution with ν degrees of freedom

$$\chi^2_\nu \equiv \sum_{i=1}^{\nu} \frac{(x_i - \mu_i)^2}{\sigma_i^2} \longrightarrow P(\chi^2_\nu) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \exp^{-\chi^2/2} (\chi^2)^{(\nu/2)-1}$$

$$\langle \chi^2 \rangle = \int_0^\infty \chi^2 P(\chi^2_{nu}; \nu) d\chi^2 = \nu$$



Caution!!

$N_{dof} = \nu$ – number of parameters

only holds for linear models

Andrae et al., Dos and don'ts of reduced chi-squared, arXiv:astro-ph/1012.3754

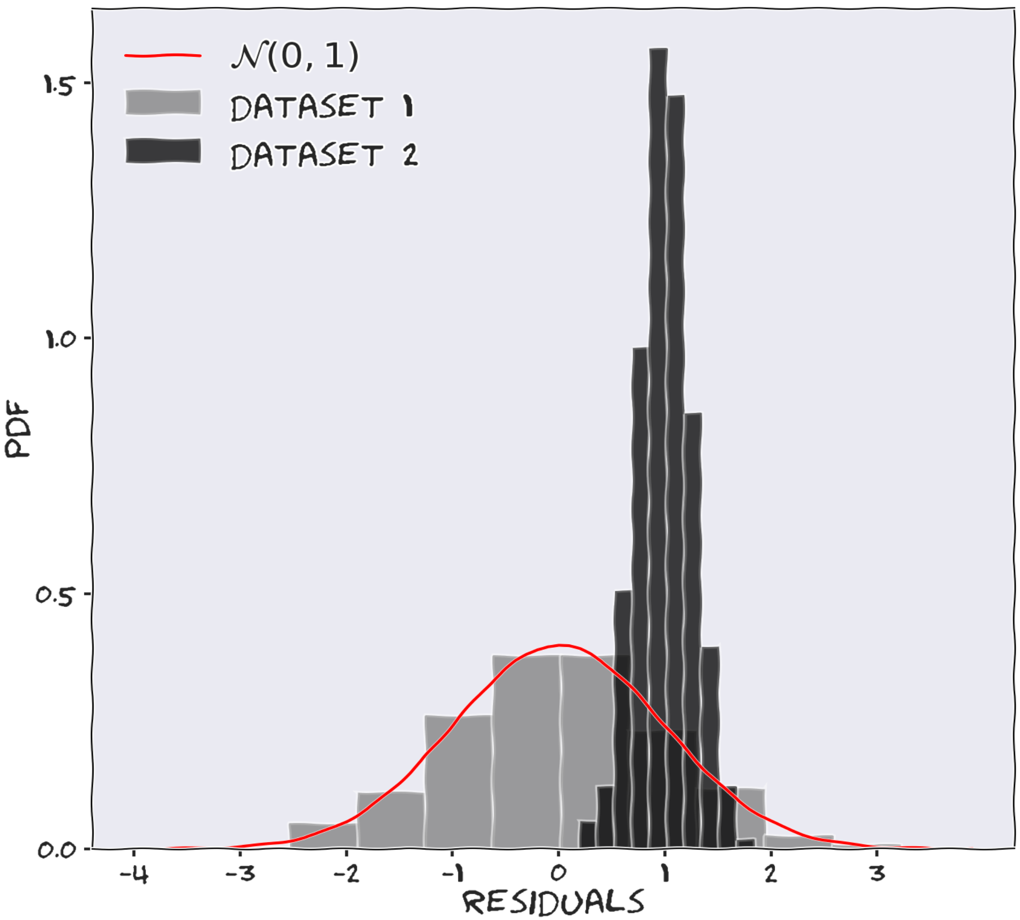
Residuals:

Model: $y = ax + b$

$$r_i = \frac{(y_i - f(x_i; a, b))^2}{\sigma^2}$$

$y = f(x; a,b) + \varepsilon$

$\varepsilon \sim Normal(0, \sigma)$



(iii) goodness of fit

Andrae et al., Dos and don'ts of reduced chi-squared, arXiv:astro-ph/1012.3754

Bayesian Inference:

$$P(a,b|\vec{x},\vec{y},\sigma) = \frac{\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)}{\int_a \int_b \mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)dadb}$$

For now ...

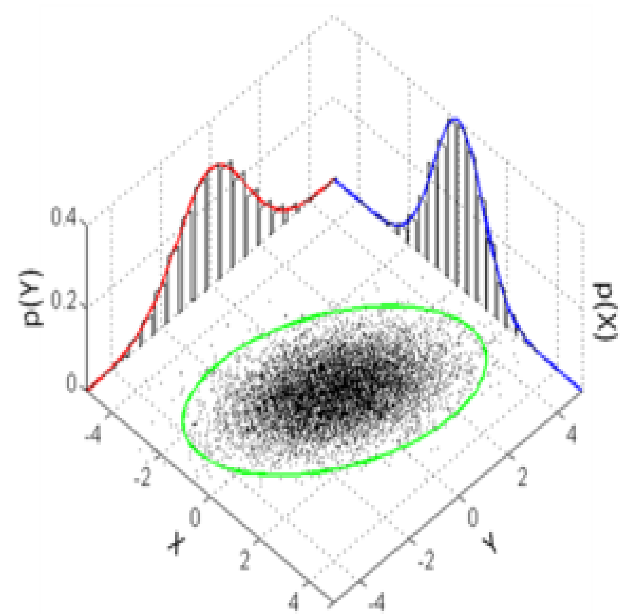
$$P(a,b|\vec{x},\vec{y},\sigma) \propto \mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)$$

$$B_{01} = \frac{E(\mathcal{D}|M_1)}{E(\mathcal{D}|M_2)}$$

(iii) goodness of fit

(i) point

(ii) credible



Always comparative!

y = a x + b

Frequentist:

$$\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-(ax_i+b))^2}{\sigma^2}\right]$$

Bayesian:

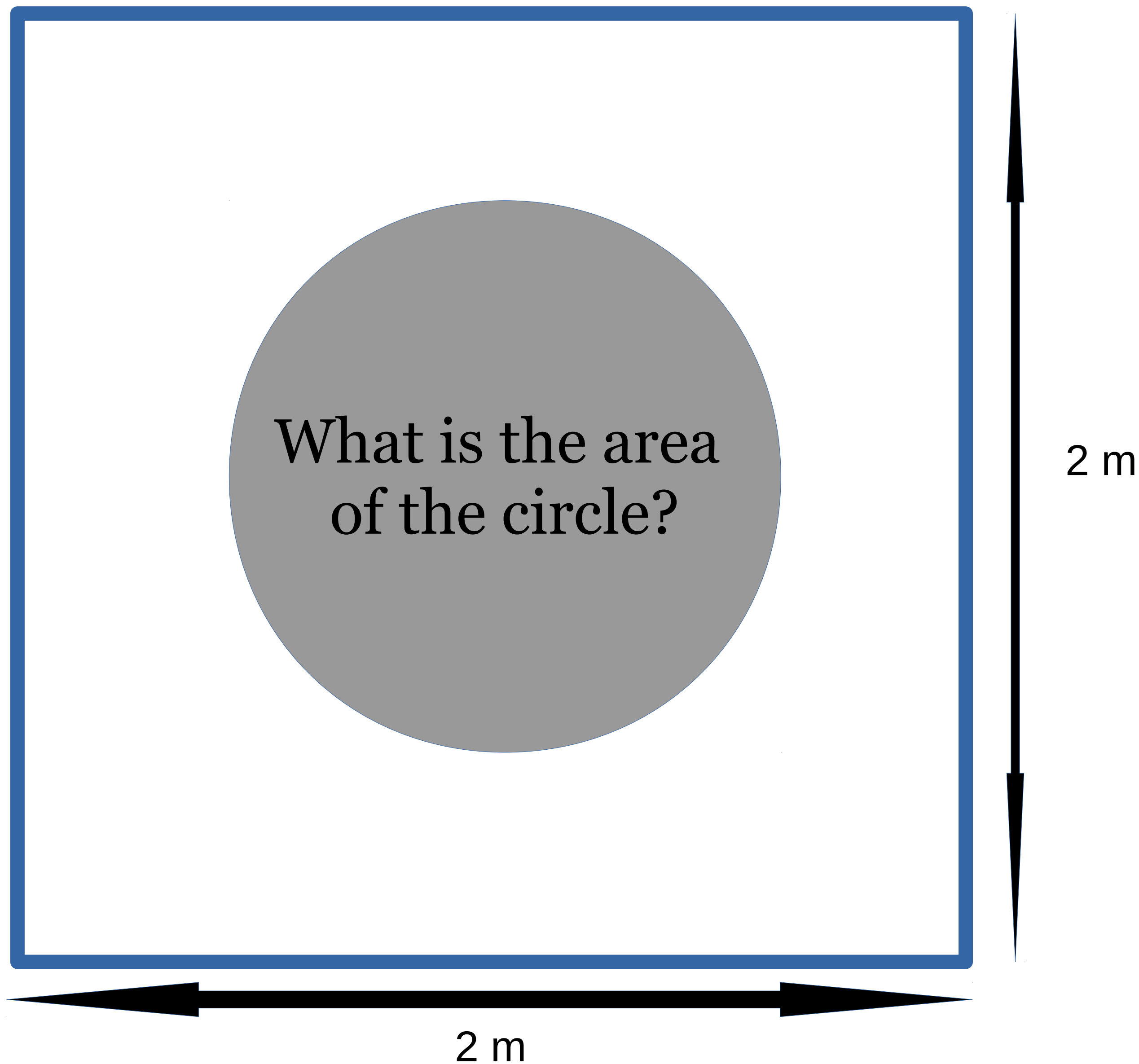
$$P(a,b|\vec{x},\vec{y},\sigma) = \frac{\mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)}{\int_a\int_b \mathcal{L}(\vec{x},\vec{y}|a,b,\sigma)p(a)p(b)dadb}$$

How to estimate parameter values?

Monte Carlo methods

Use randomness as numerical calculation tool

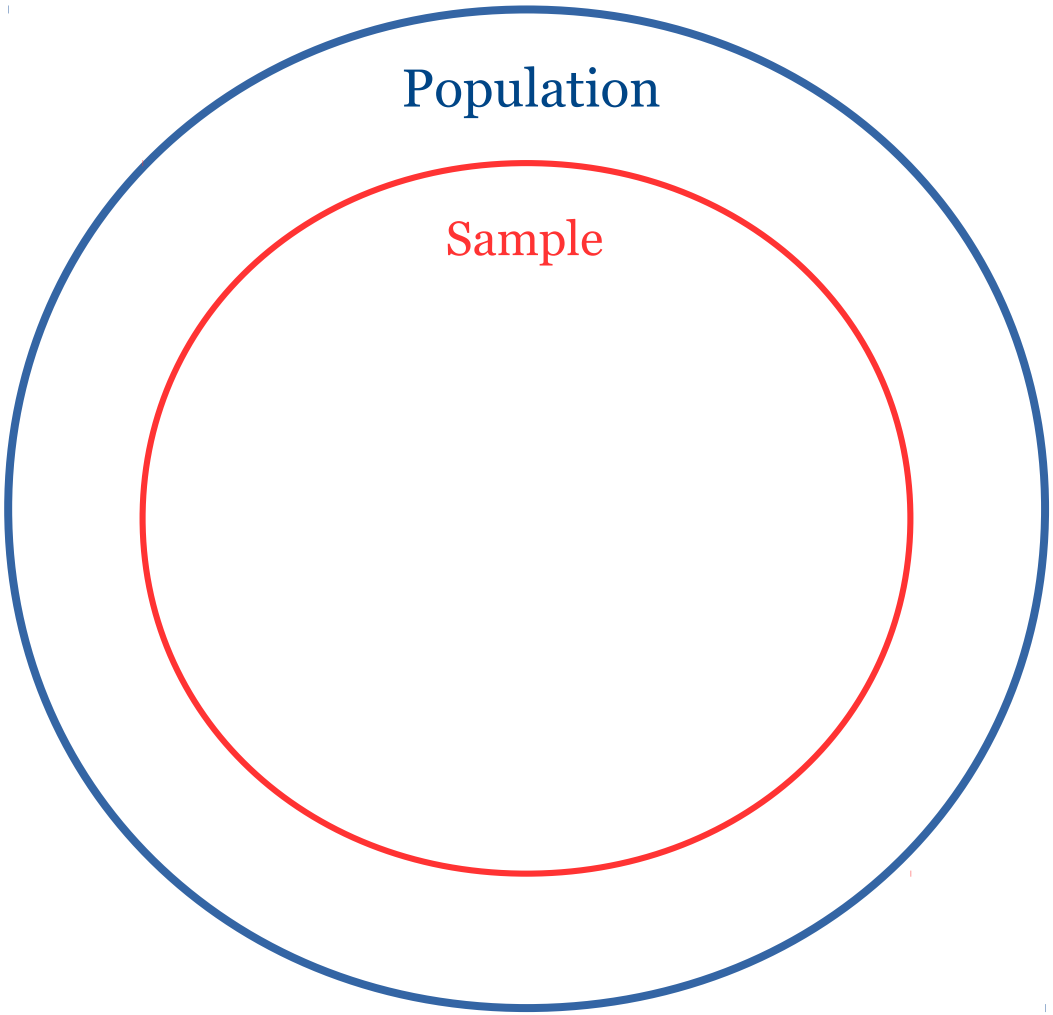
100 M&M's



Monte Carlo methods

Use randomness as numerical calculation tool

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$



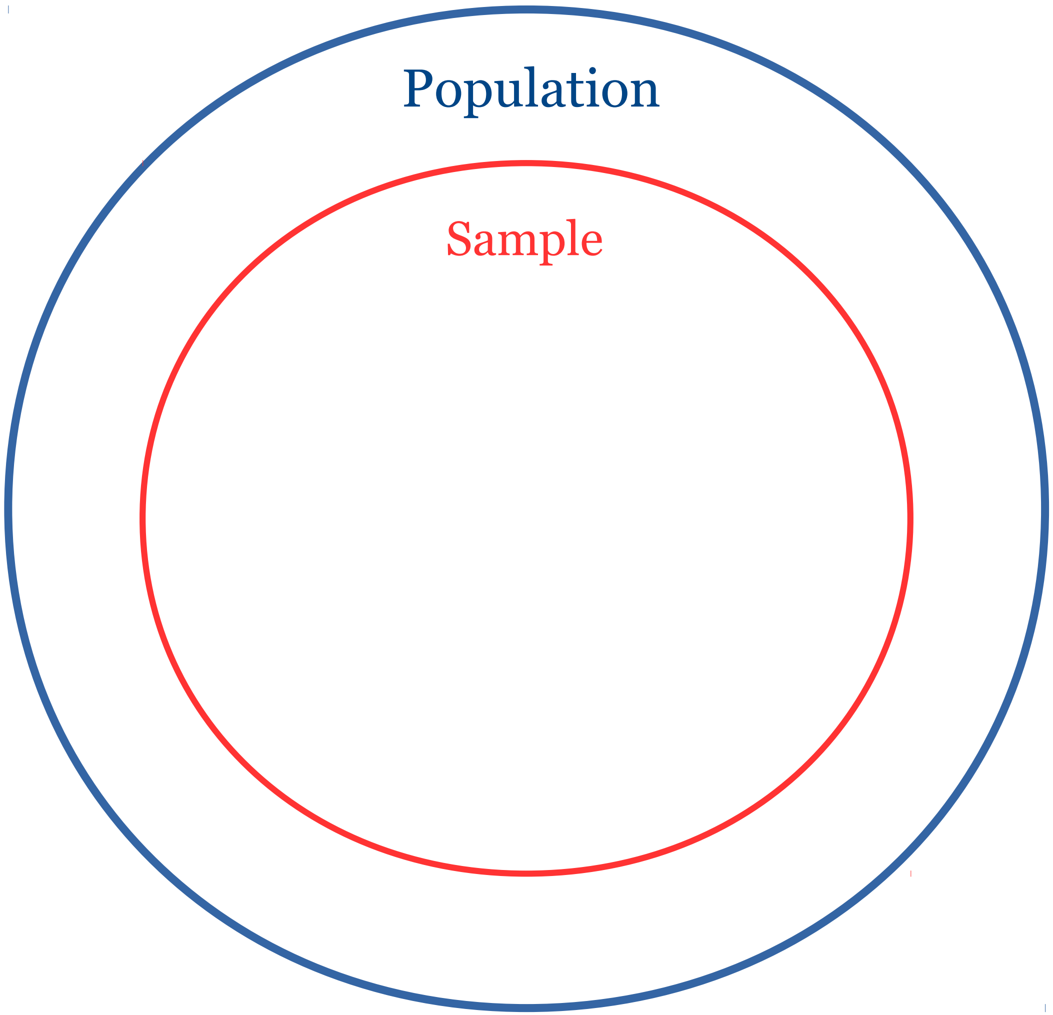
Estimator → *the rule that creates an estimate*

Monte Carlo methods

Use randomness as numerical calculation tool

$$g = f(x)$$

$$\hat{g} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$



Estimator → *the rule that creates an estimate*

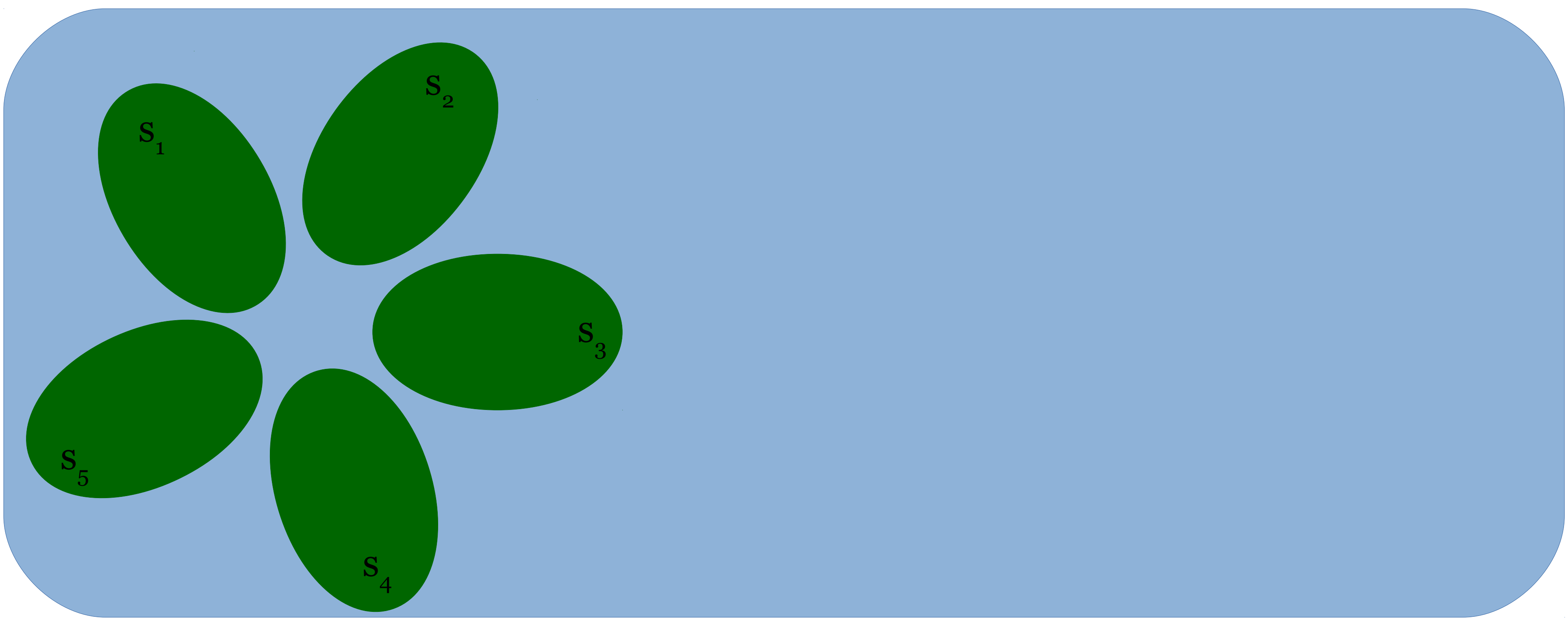
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $P_{trans}(s_i \rightarrow s_{i+1}) = 0.25 * [1 - \delta(s_i - s_{i+1})]$



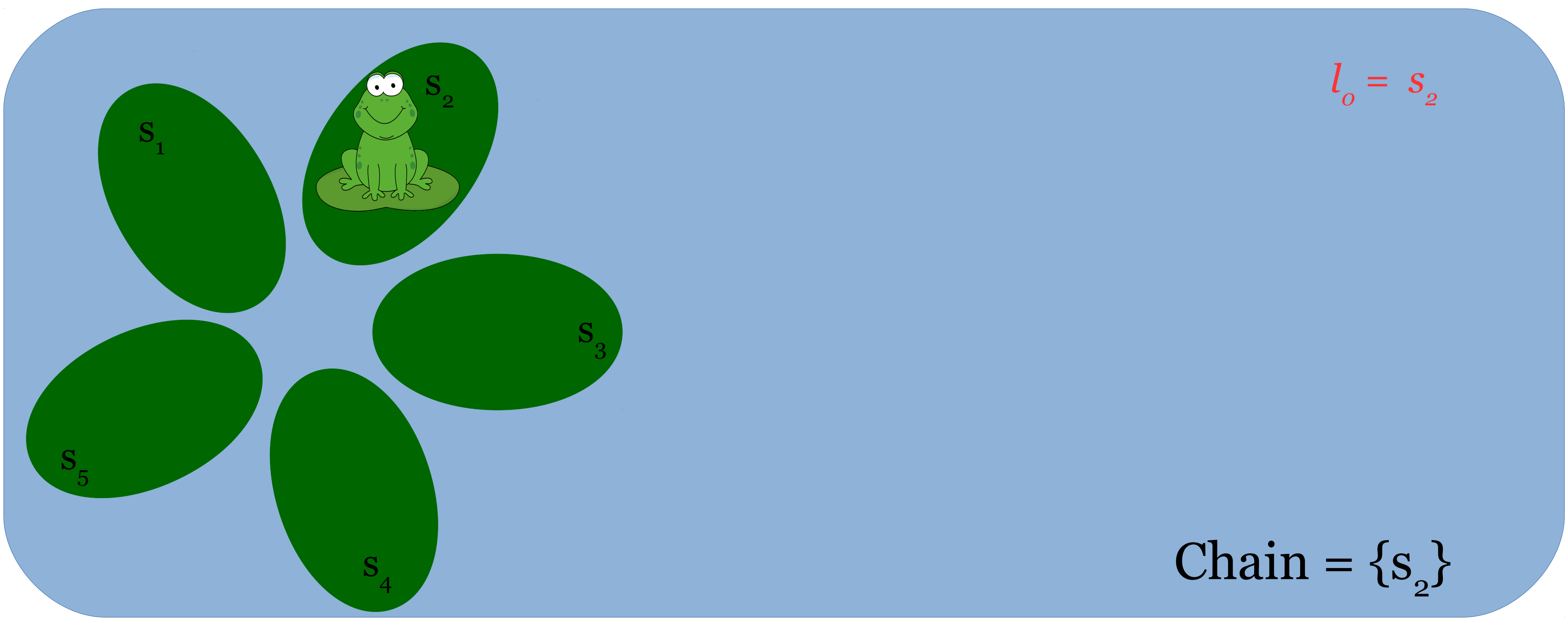
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $P_{trans}(s_i \rightarrow s_{i+1}) = 0.25 * [1 - \delta(s_i - s_{i+1})]$



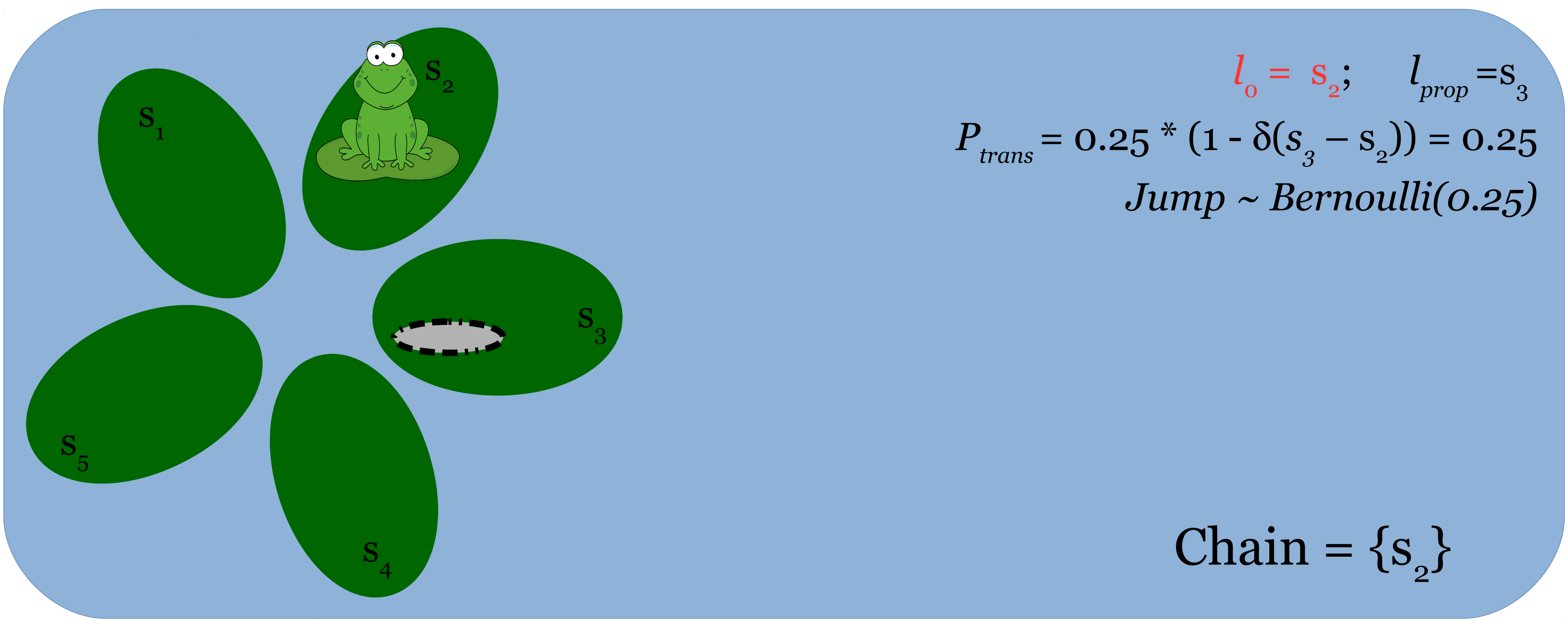
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $P_{trans}(s_i \rightarrow s_{i+1}) = 0.25 * [\,1 - \delta(s_i - s_{i+1})]$



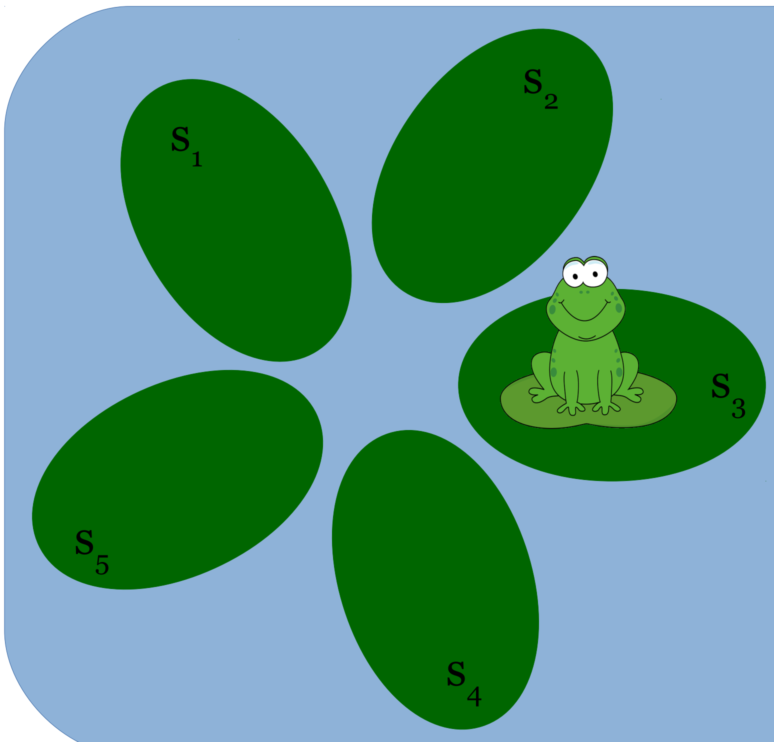
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $T(s_i \rightarrow s_{i+1}) = 0.25 * [\, 1 - \delta(s_i - s_{i+1})]$



$l_o = s_2; \quad l_{prop} = s_3$

$P_{trans} = 0.25 * (1 - \delta(s_3 - s_2)) = 0.25$

$Jump \sim Bernoulli(T) = True \rightarrow l_1 = s_3$

Chain = $\{s_2, s_3\}$

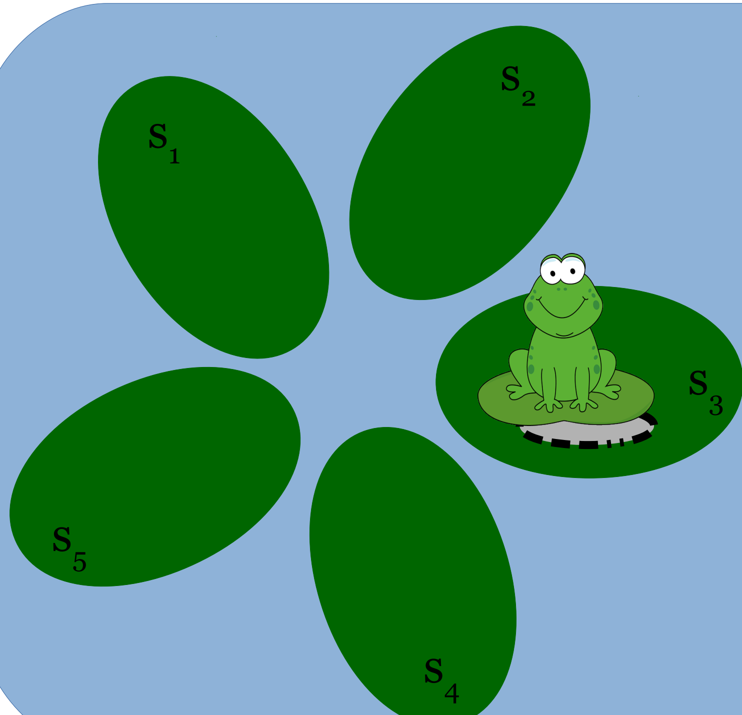
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $T(s_i \rightarrow s_{i+1}) = 0.25 * [\, 1 - \delta(s_i - s_{i+1})]$



$l_0 = s_2; \quad l_{prop} = s_3$

$P_{trans} = 0.25 * (1 - \delta(s_3 - s_2)) = 0.25$

$Jump \sim Bernoulli(0.25) \rightarrow True \rightarrow l_1 = s_3; \; l_{prop} = s_3$

Chain = $\{s_2, s_3, s_3\}$

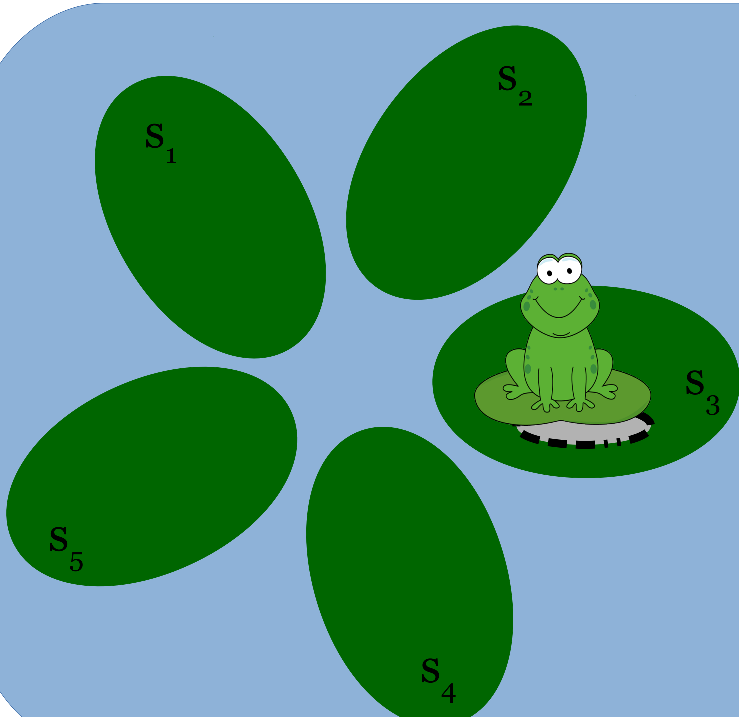
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $T(s_i \rightarrow s_{i+1}) = 0.25 * [\, 1 - \delta(s_i - s_{i+1})]$



$l_0 = s_2; \quad l_{prop} = s_3$

$P_{trans} = 0.25 * (1 - \delta(s_3 - s_2)) = 0.25$

$Jump \sim Bernoulli(0.25) \rightarrow True \rightarrow l_1 = s_3; \; l_{prop} = s_3$

$T = 0.25 * (1 - \delta(s_3 - s_3)) = 0$

$Jump \sim Bernoulli(0) \rightarrow False \rightarrow l_2 = s_3$

Chain = $\{s_2, s_3, s_3\}$

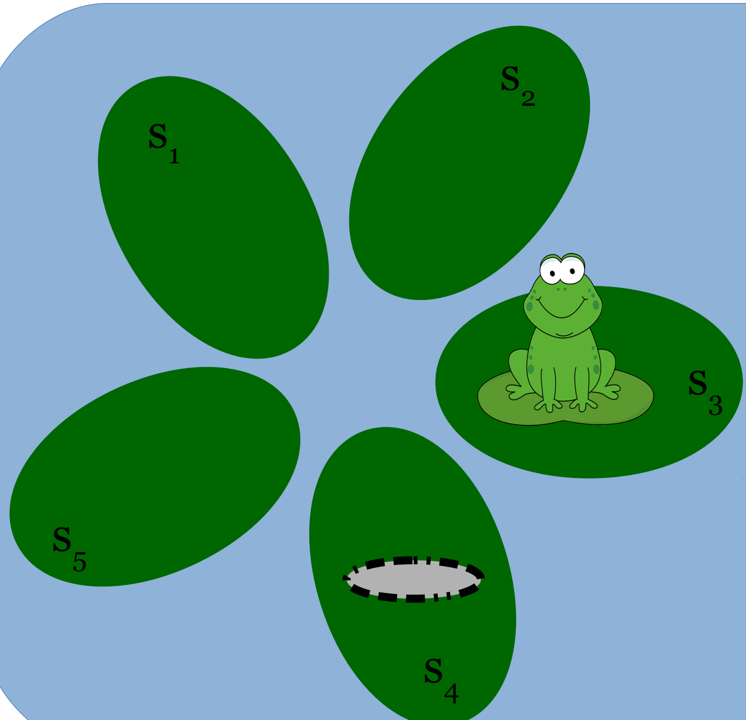
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $T(s_i \rightarrow s_{i+1}) = 0.25 * [\,1 - \delta(s_i - s_{i+1})]$



$l_0 = s_2;\quad l_{prop} = s_3$

$P_{trans} = 0.25 * (1 - \delta(s_3 - s_2)) = 0.25$

$Jump \sim Bernoulli(T) = True \rightarrow l_1 = s_3;\quad l_{prop} = s_3$

$T = 0.25 * (1 - \delta(s_3 - s_3)) = 0$

$Jump \sim Bernoulli(T) = False \rightarrow l_2 = s_3;\quad l_{prop} = s_4$

Chain = $\{s_2, s_3, s_3, s_4\}$

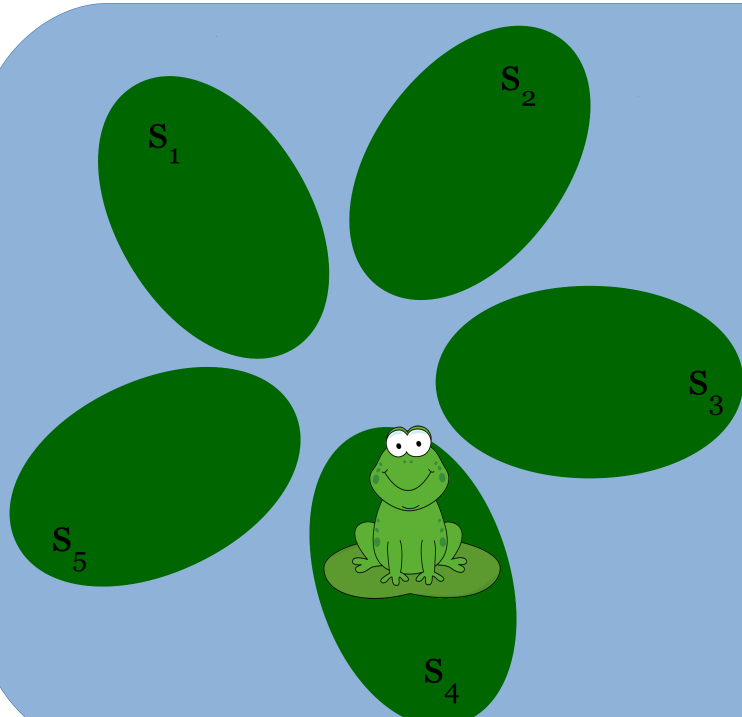
Markov Chain

Making the most of short-term memory

A set of possible states: $S = \{s_1, s_2, s_3, s_4, s_5\}$

Proposal distribution: $P(s) = 0.2$

Transition probability: $T(s_i \rightarrow s_{i+1}) = 0.25 * [\, 1 - \delta(s_i - s_{i+1})]$



$l_0 = s_2; \quad l_{prop} = s_3$

$P_{trans} = 0.25 * (1 - \delta(s_3 - s_2)) = 0.25$

Jump ~ Bernoulli(0.25) = True → $l_1 = s_3; \quad l_{prop} = s_3$

$P_{trans} = 0.25 * (1 - \delta(s_3 - s_3)) = 0$

Jump ~ Bernoulli(0) = False → $l_2 = s_3; \quad l_{prop} = s_4$

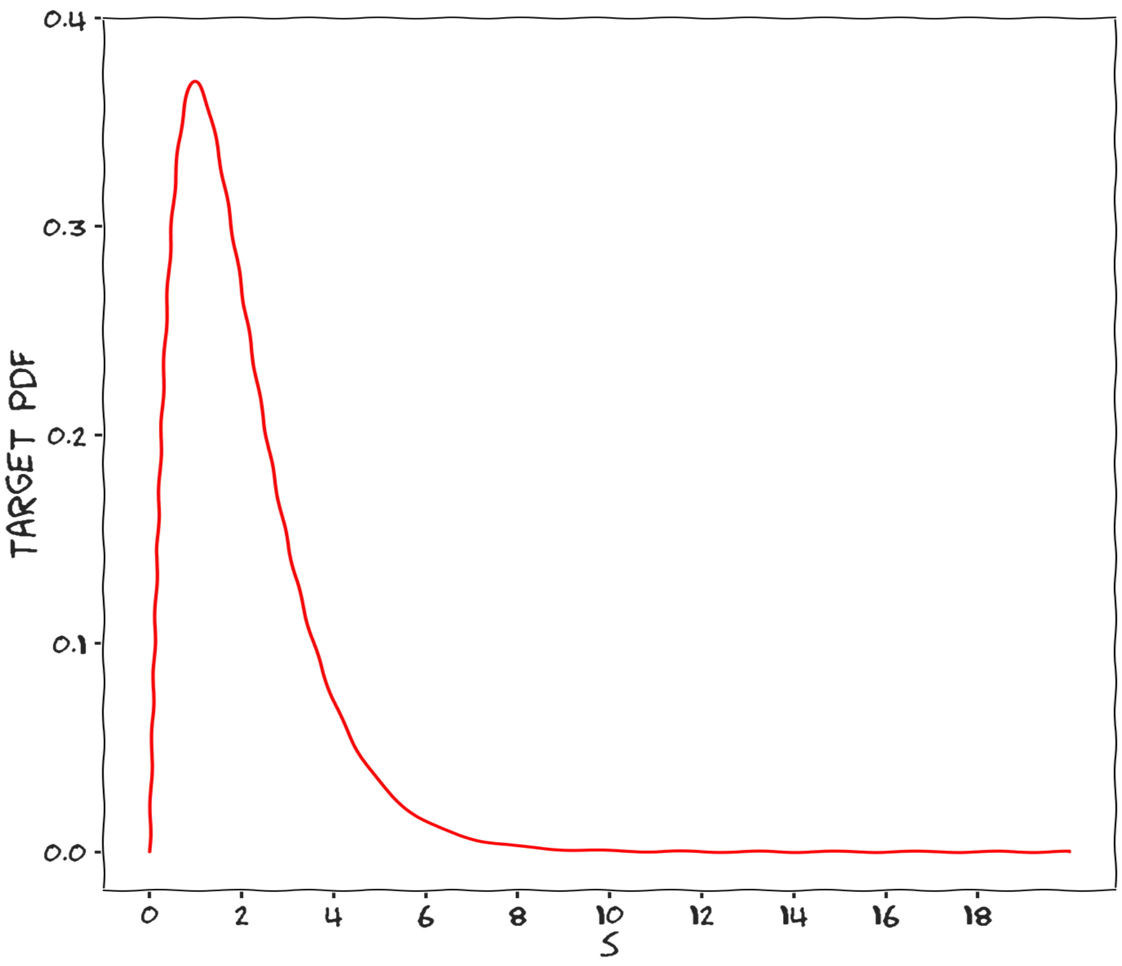
$P_{trans} = 0.25 * (1 - \delta(s_3 - s_4)) = 0.25$

Jump ~ Bernoulli(0.25) = True → $l_3 = s_4$

Chain = $\{s_2, s_3, s_3, s_4\}$

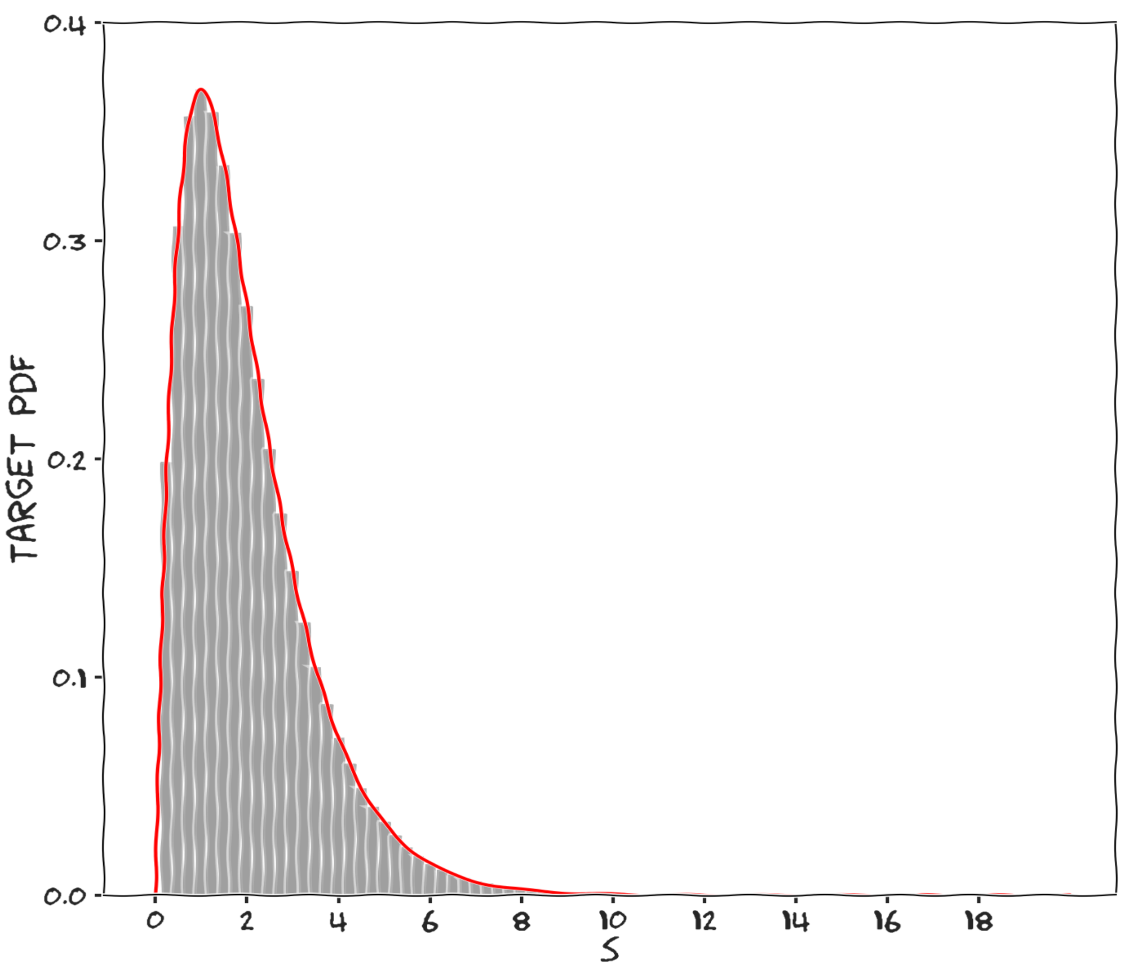
Markov Chain Monte Carlo

Sampling from a distribution



Markov Chain Monte Carlo

Sampling from a distribution

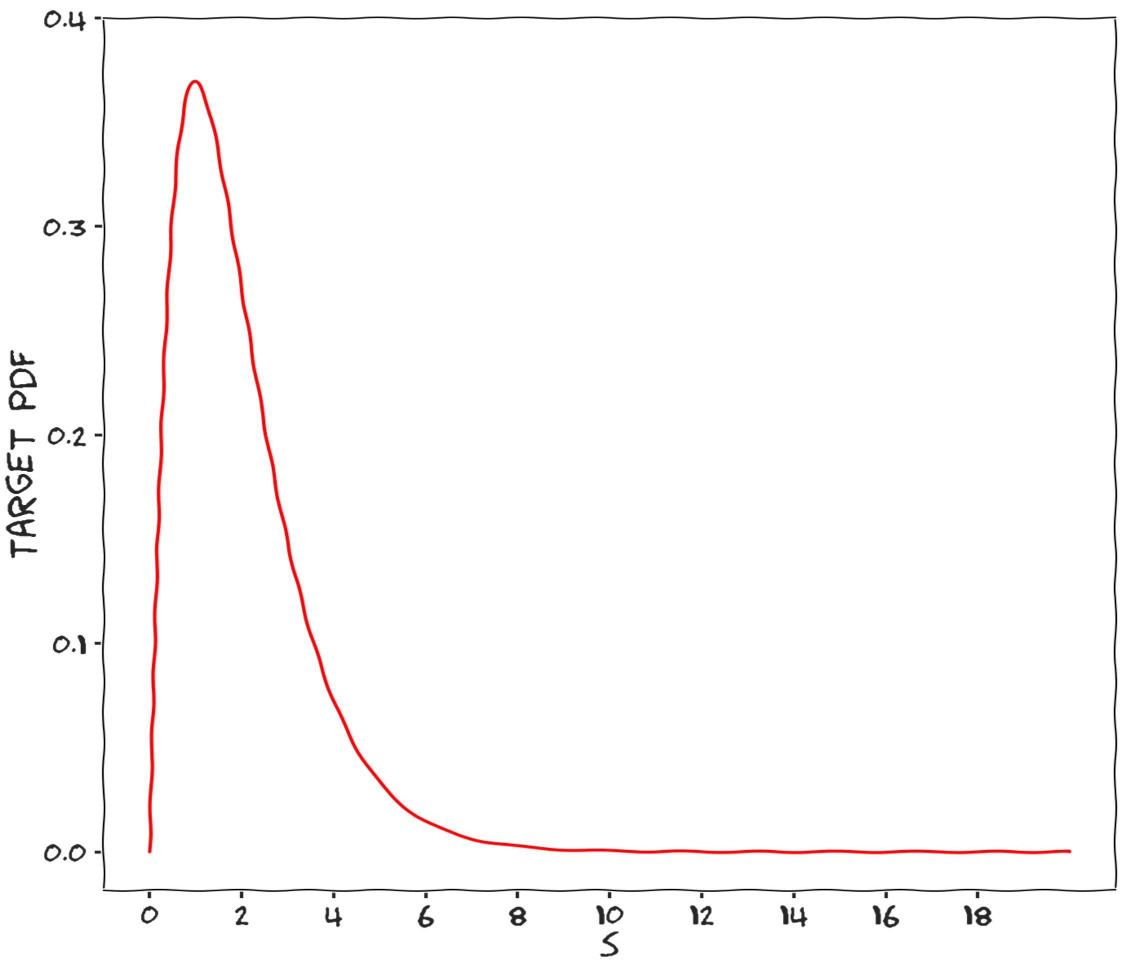


A sample allows the calculation of properties, but ...

How to do that wisely?

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

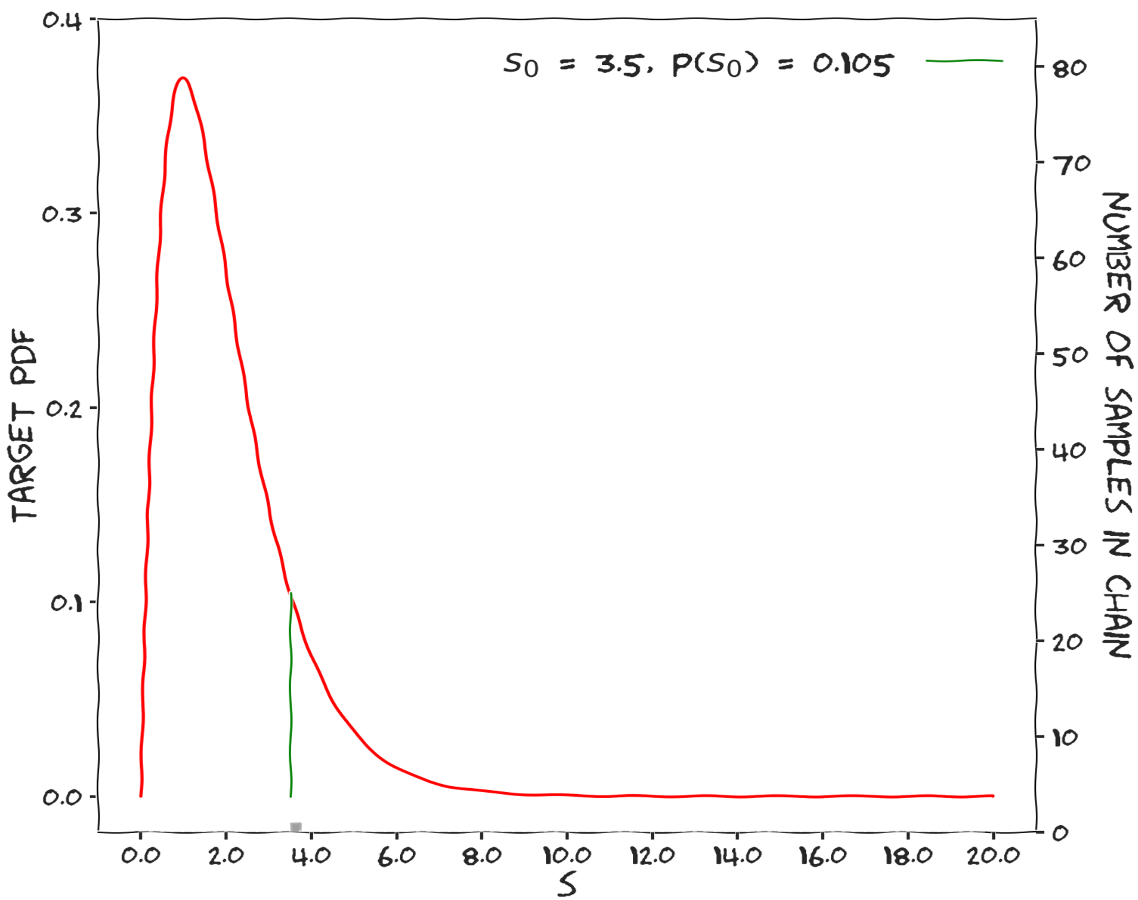
$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

$T = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

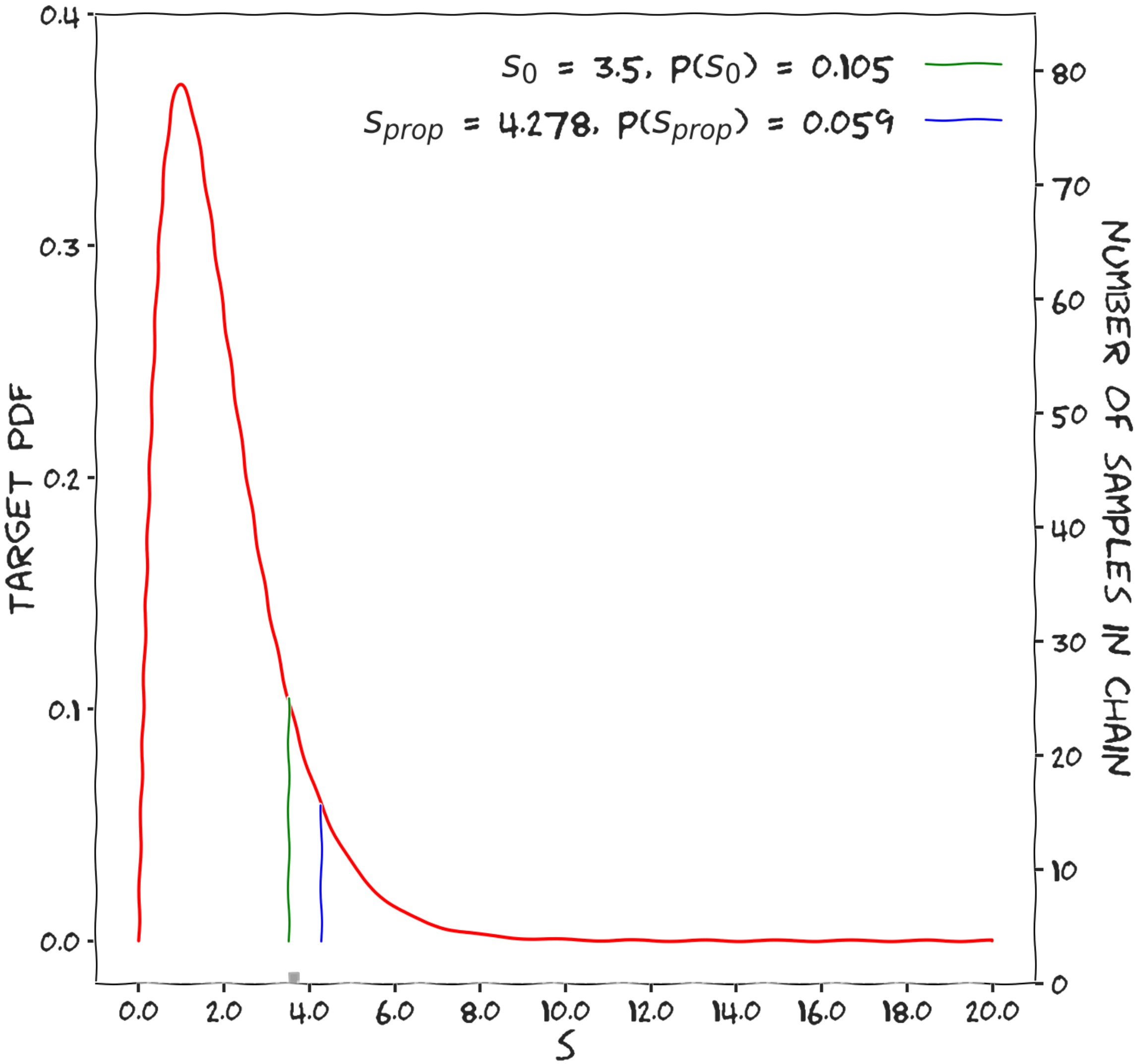
$T = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

Algorithm:

1. given $s_o \rightarrow P_{target}(s_o)$

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

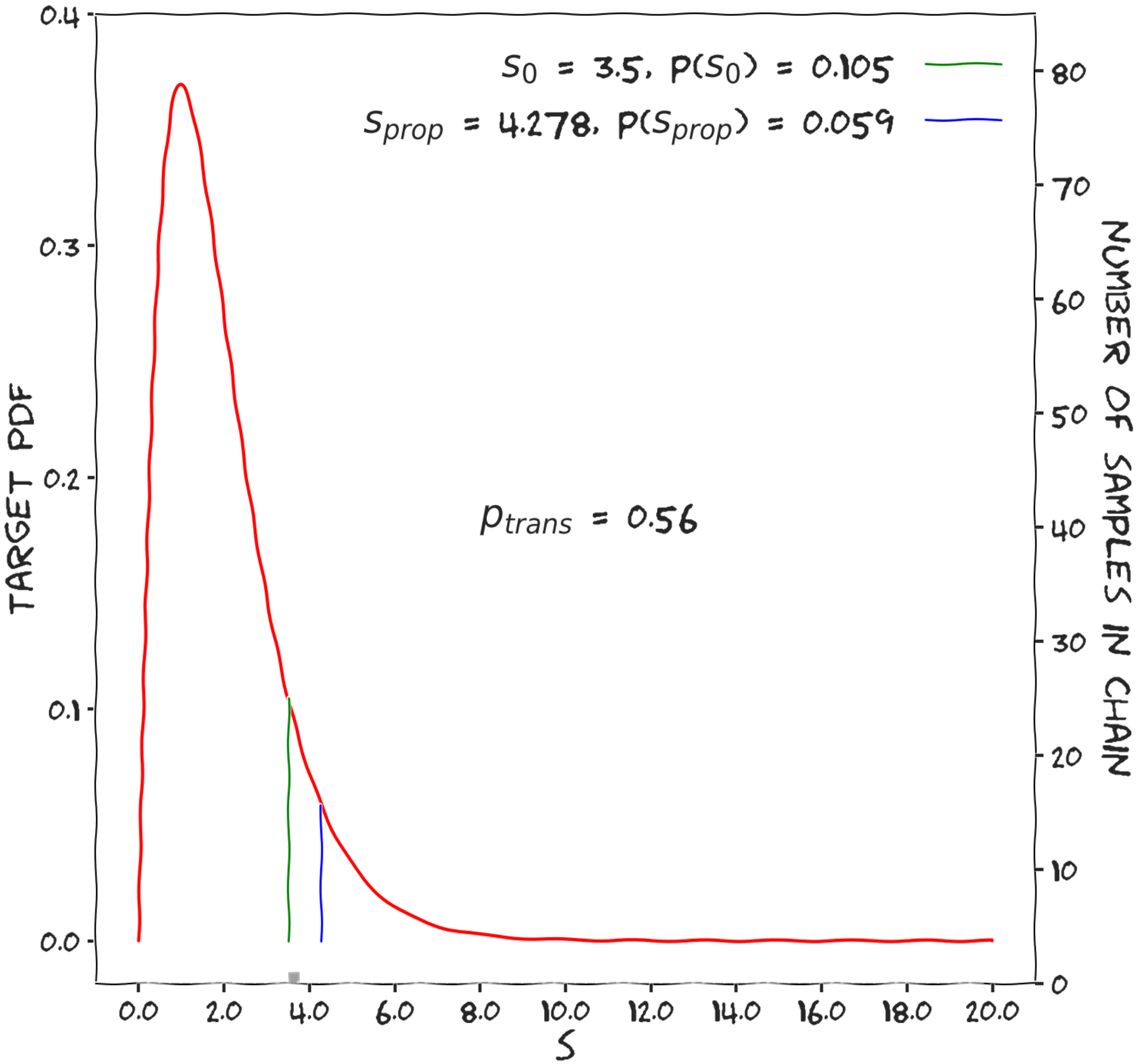
$T = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

Algorithm:

1. given $s_o \rightarrow P_{target}(s_o)$
2. propose a new state, s_{prop} and get $P_{target}(s_{prop})$

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

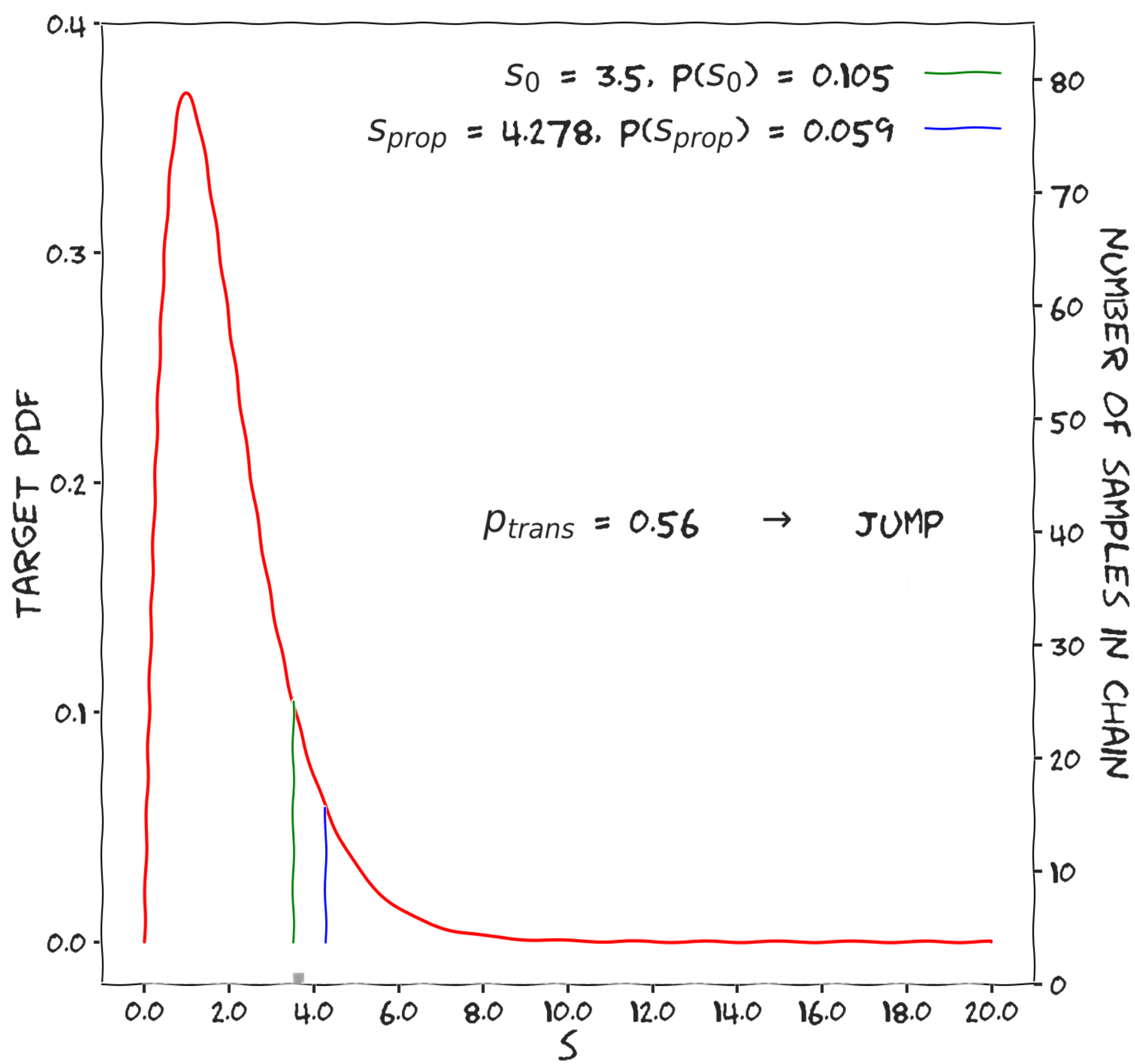
$P_{trans} = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

Algorithm:

1. given $s_o \rightarrow P_{target}(s_o)$
2. propose a new state, s_{prop} and get $P_{target}(s_{prop})$
3. Calculate P_{trans}

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

$P_{trans} = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

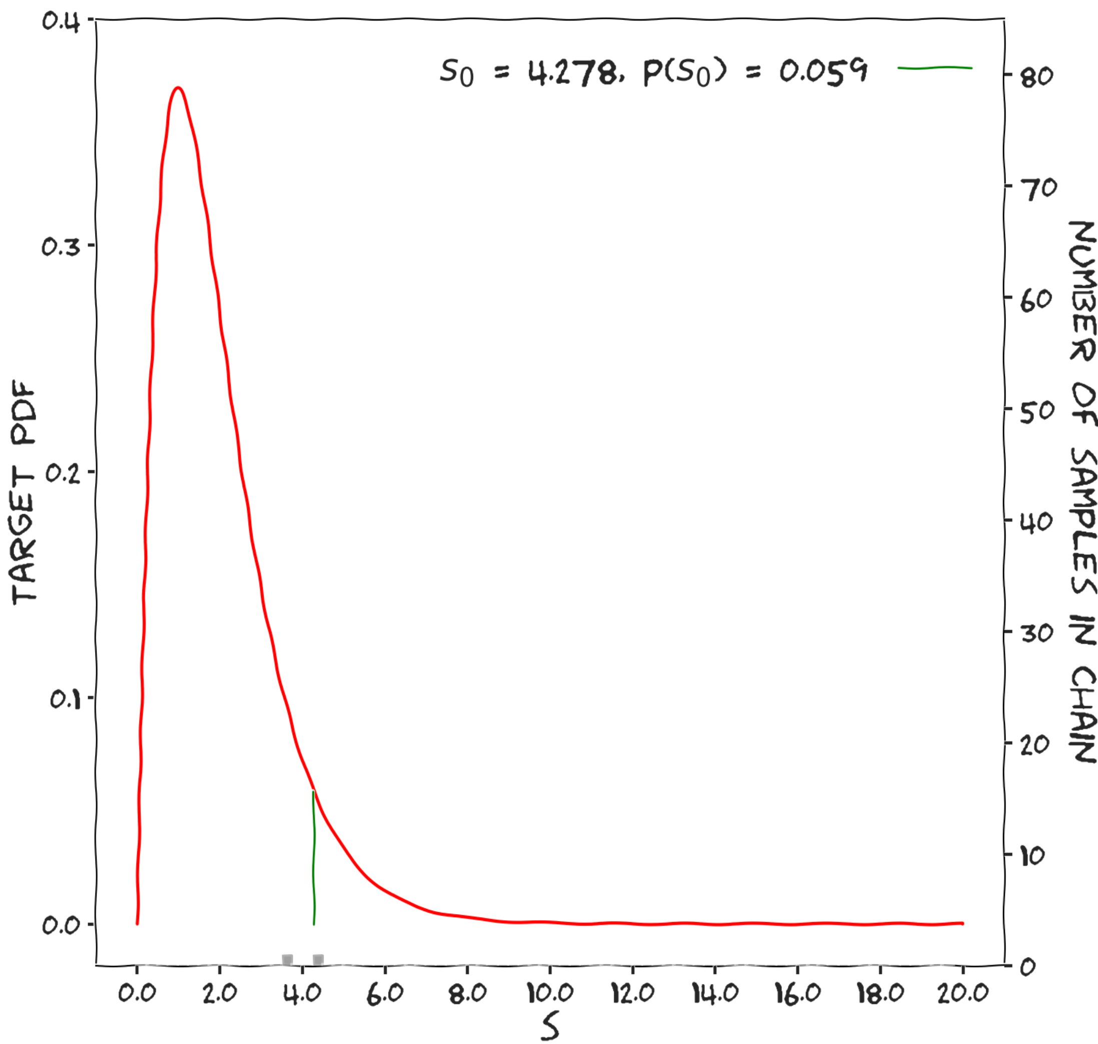
Algorithm:

1. given $s_{now} \rightarrow P_{target}(s_{now})$
2. propose a new state, s_{prop} and get $P_{target}(s_{prop})$
3. Calculate P_{trans}
4. *Jump ~ Bernoulli(P_{trans})*
5. *If Jump:*

   $s_{now} = s_{prop}$
6. *Add s_{now} to chain*
7. *Go to 2*

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

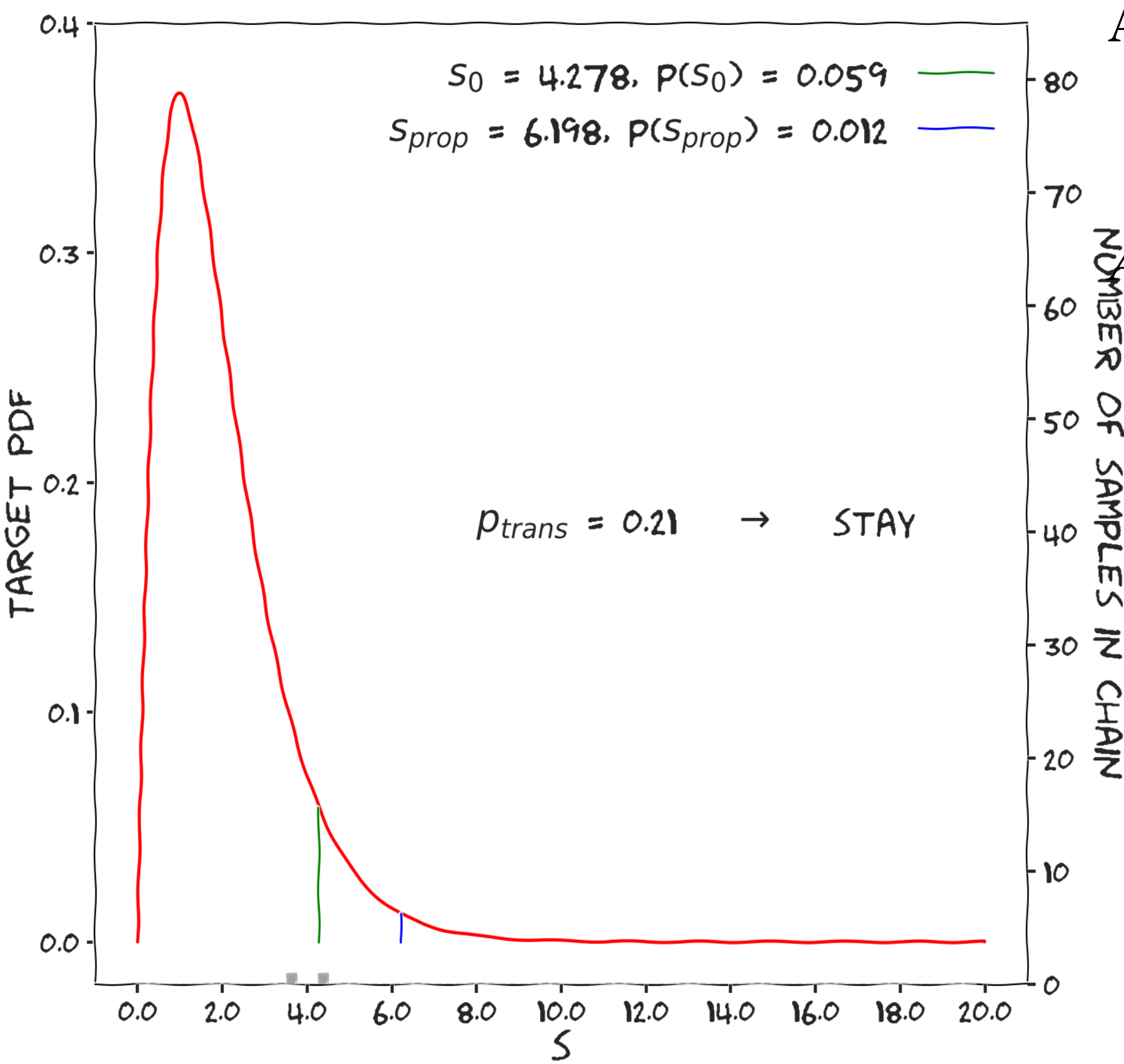
$P_{trans} = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

Algorithm:

1. given $s_{now} \rightarrow P_{target}(s_{now})$
2. propose a new state, s_{prop} and get $P_{target}(s_{prop})$
3. Calculate P_{trans}
4. *Jump ~ Bernoulli(P_{trans})*
5. *If Jump:*
   $s_{now} = s_{prop}$
6. *Add s_{now} to chain*
7. *Go to 2*

Markov Chain Monte Carlo

Sampling from a distribution



Possible states: $0 < s < 20$

A proposal distribution:

Given Δ,

$s_{prop} \sim Uniform(s_{now} - \Delta,\ s_{now} + \Delta)$

A transition probability:

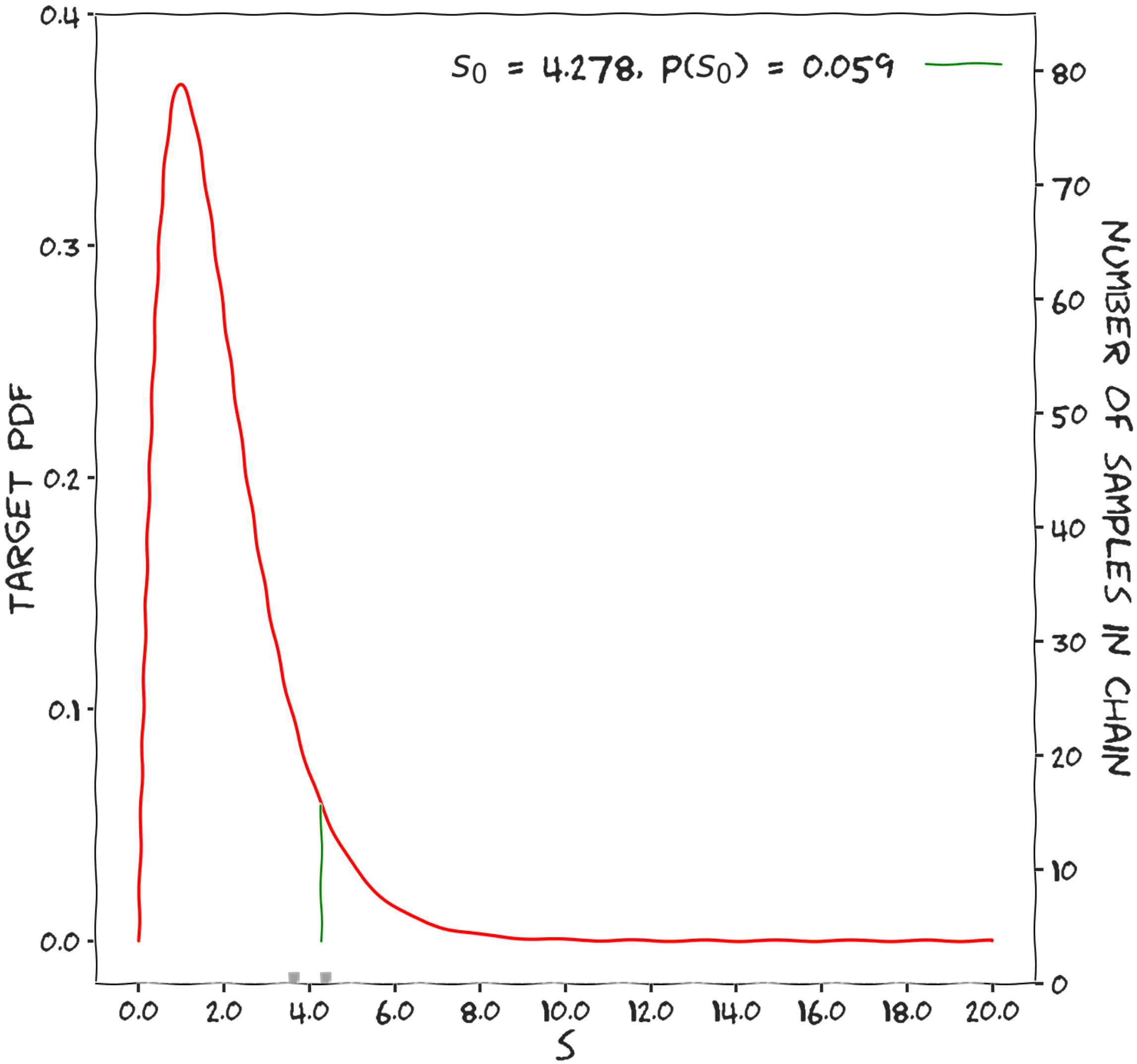
$P_{trans} = min(1,\ P_{target}(s_{prop})/P_{target}(s_{now}))$

Algorithm:

1. given $s_{now} \rightarrow P_{target}(s_{now})$
2. propose a new state, s_{prop} and get $P_{target}(s_{prop})$
3. Calculate P_{trans}
4. *Jump ~ Bernoulli(P_{trans})*
5. *If Jump:*
   $s_{now} = s_{prop}$
6. *Add s_{now} to chain*
7. *Go to 2*

Markov Chain Monte Carlo

Sampling from a distribution



A set of states: $s + P_{target}(s)$

Initial probability:
$P_{ini} \sim Uniform(0, 20)$

<u>Algorithm:</u>

1. from P_{ini} get $s_o \rightarrow P_{target}(s_o)$

2. Given a step size, Δ, propose a new state: $s_{prop} \sim Uniform(s_o - \Delta,\ s_o + \Delta)$

3. Calculate the Hasting ratio:
$H = P_{target}(s_{prop}) \,/\, P_{target}(s_o)$

4. Get the transition probability:
$P_{trans} = min\{1, H\}$

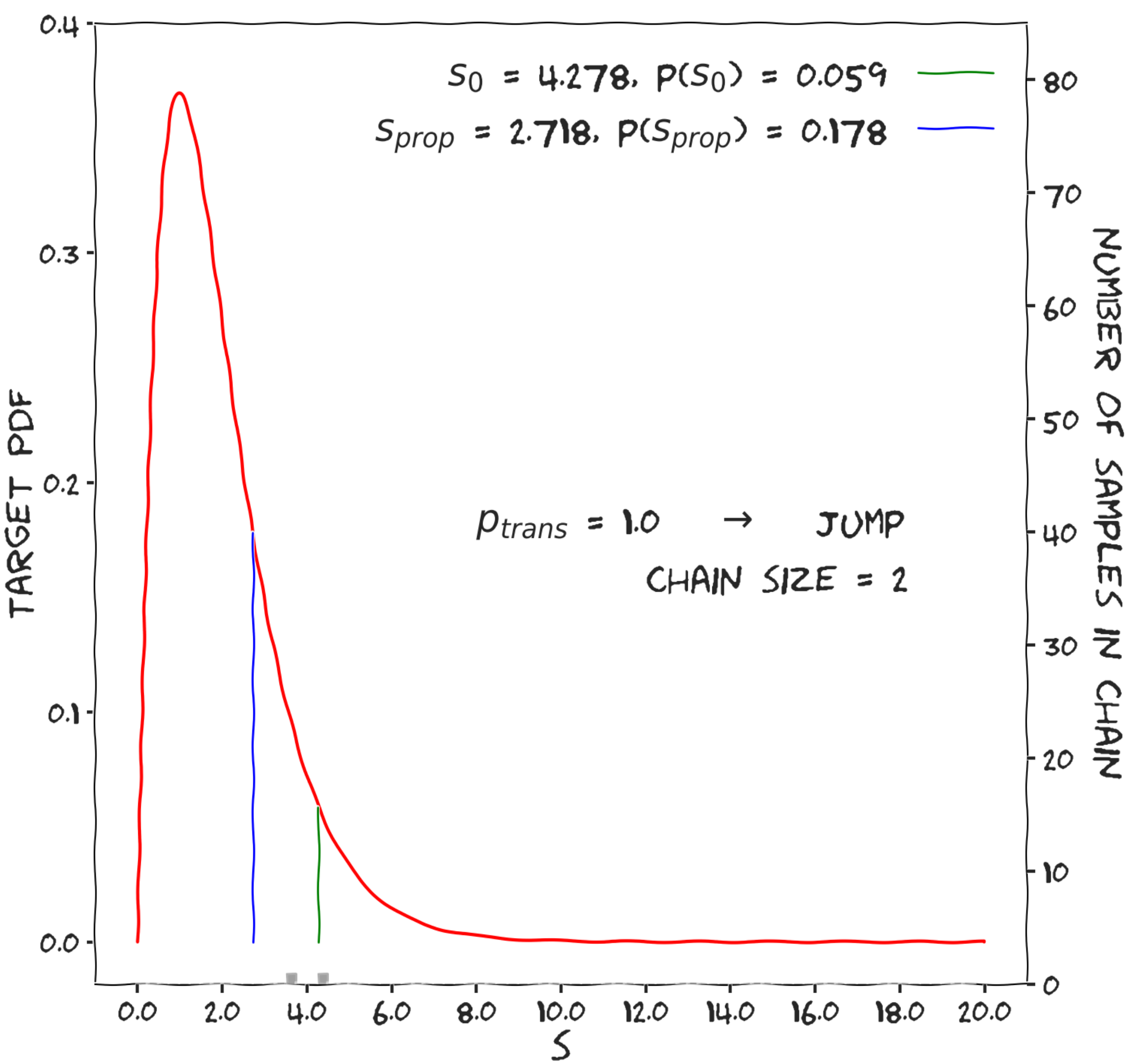
5. Flip a weighted coin:
$Jump = Bernoulli(P_{trans})$

If $Jump == Success$:
Add p_{prop} to the chain; $p_o = p_{prop}$

6. Go to step 2

Markov Chain Monte Carlo

Sampling from a distribution



A set of states: $s + P_{target}(s)$

Initial probability:

$P_{ini} \sim Uniform(0, 20)$

Algorithm:

1. from P_{ini} get $s_o \rightarrow P_{target}(s_o)$

2. Given a step size, Δ, propose a new state: $s_{prop} \sim Uniform(s_o - \Delta,\ s_o + \Delta)$

3. Calculate the Hasting ratio:
$H = P_{target}(s_{prop}) \,/\, P_{target}(s_o)$

4. Get the transition probability:
$P_{trans} = min\{1, H\}$

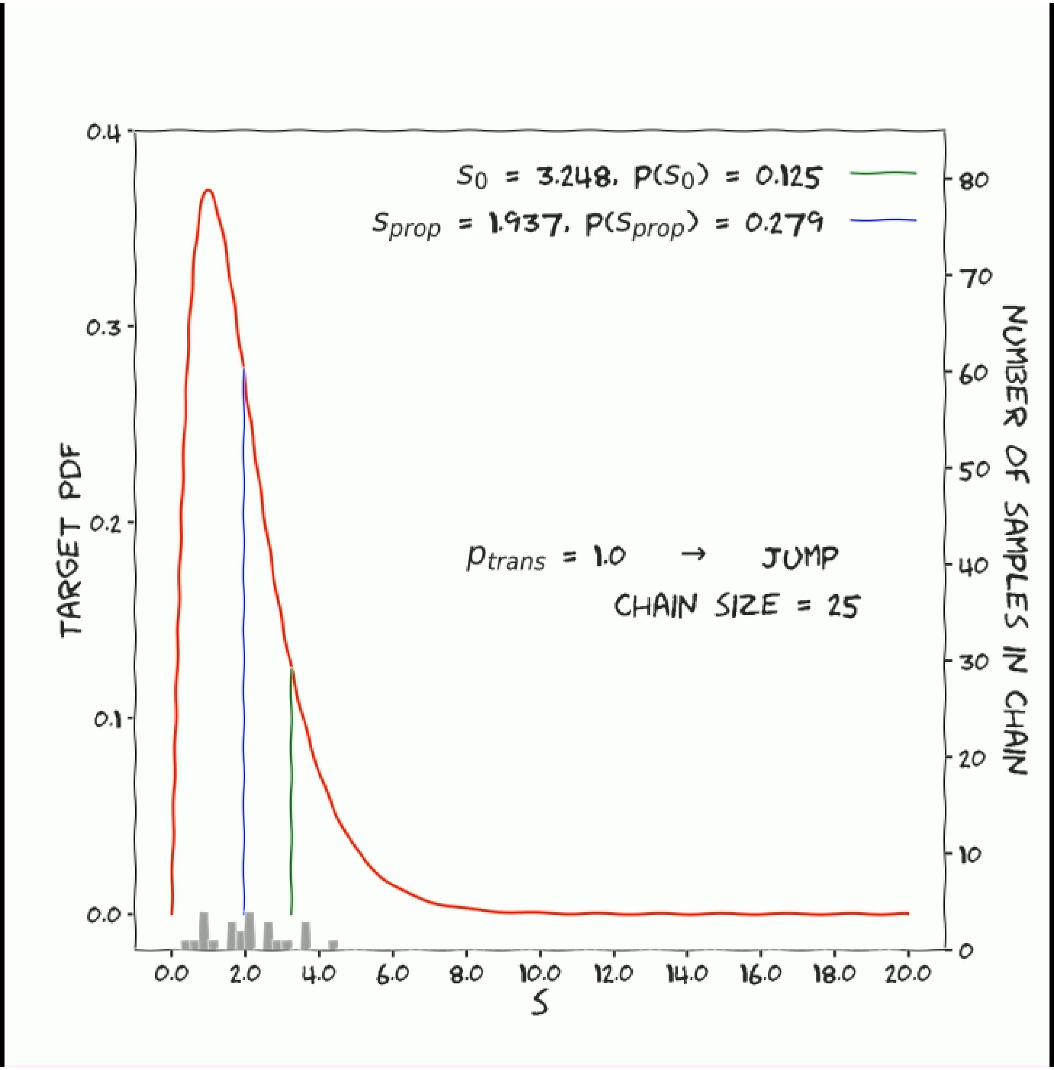
5. Flip a weighted coin:
$Jump = Bernoulli(P_{trans})$

If $Jump == Success$:
Add p_{prop} to the chain; $p_o = p_{prop}$

6. Go to step 2

Markov Chain Monte Carlo

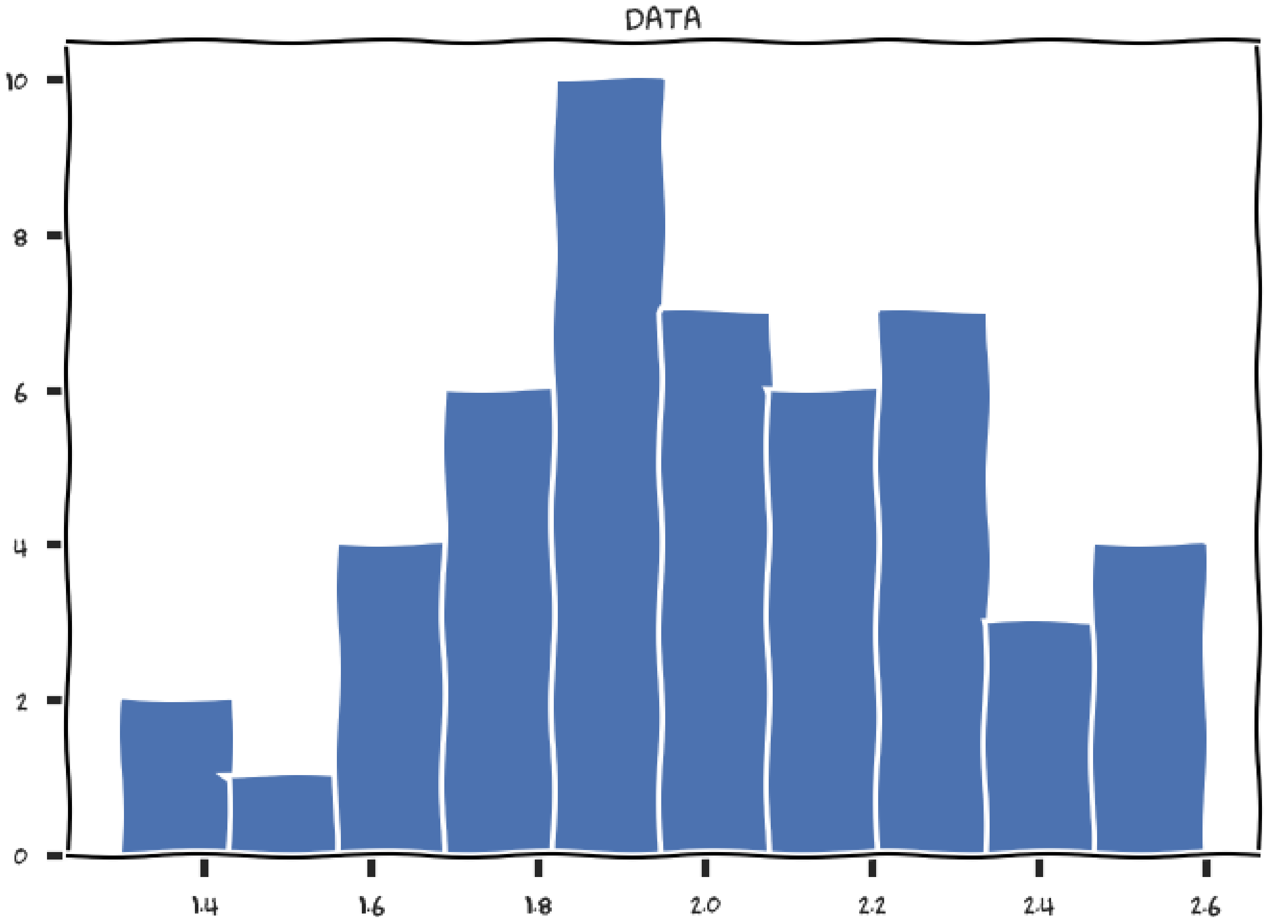
Sampling from a distribution



Markov Chain Monte Carlo

Data

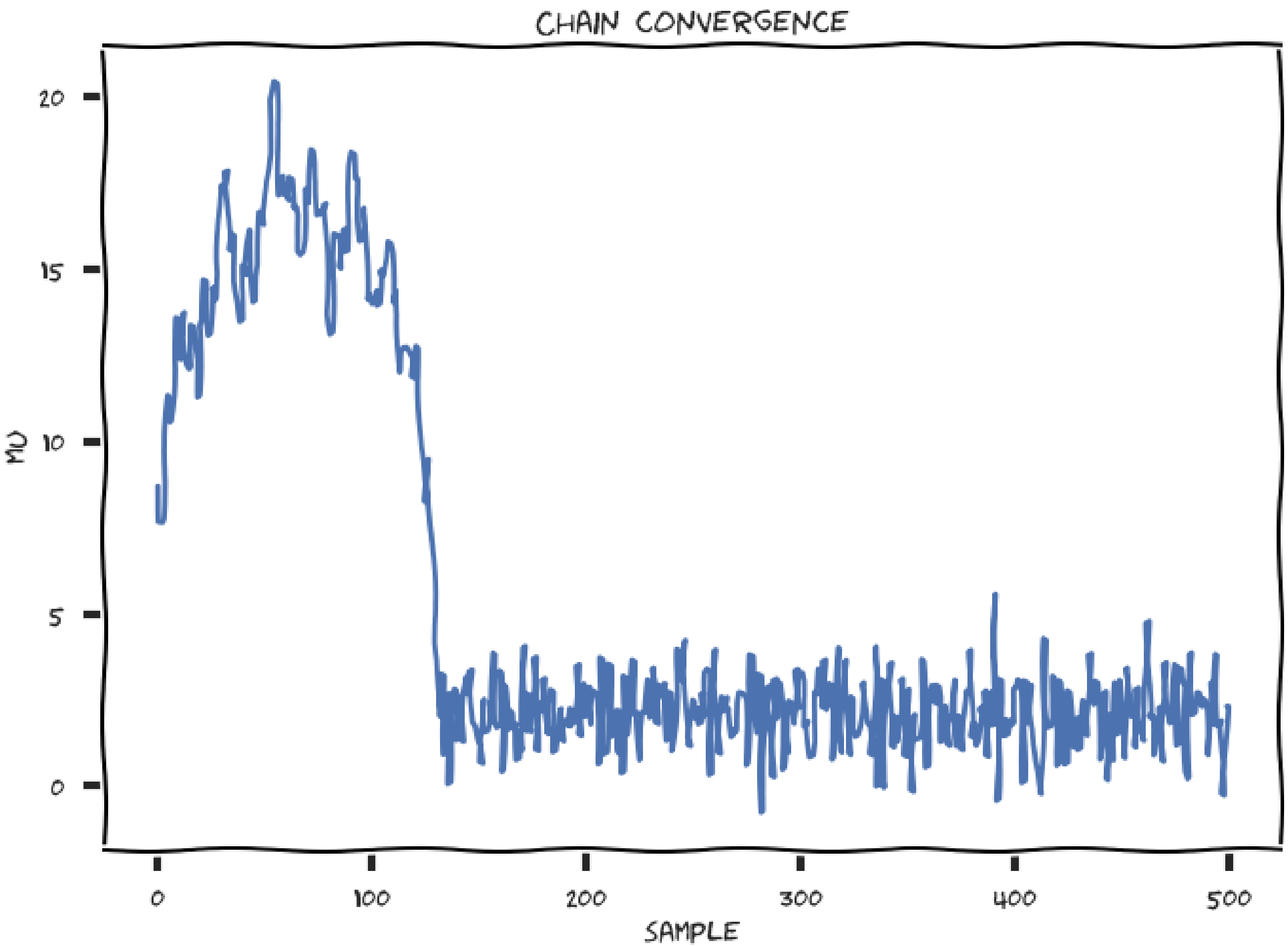
X ~ Normal(mu, σ)



X

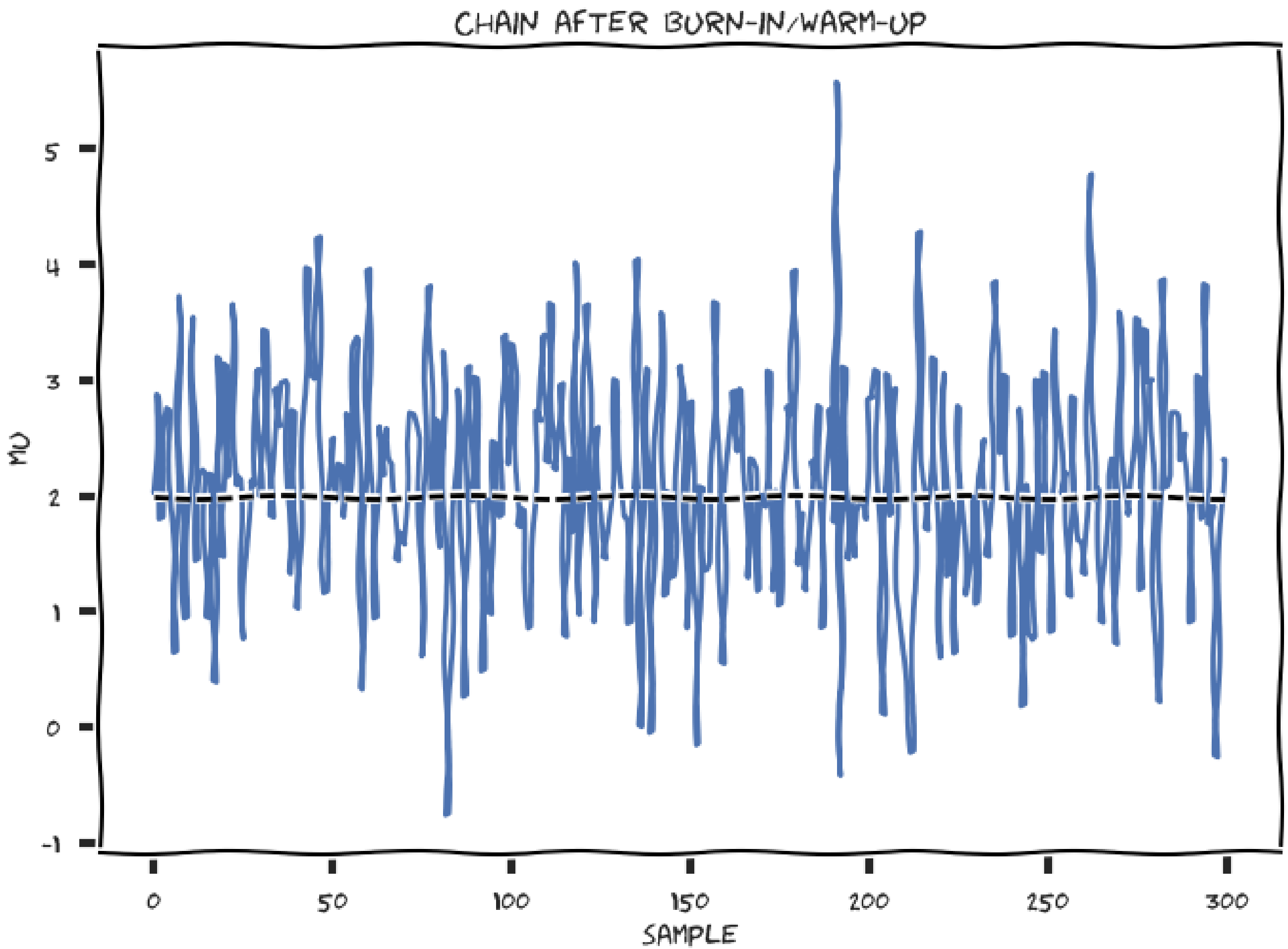
Markov Chain Monte Carlo

Chains



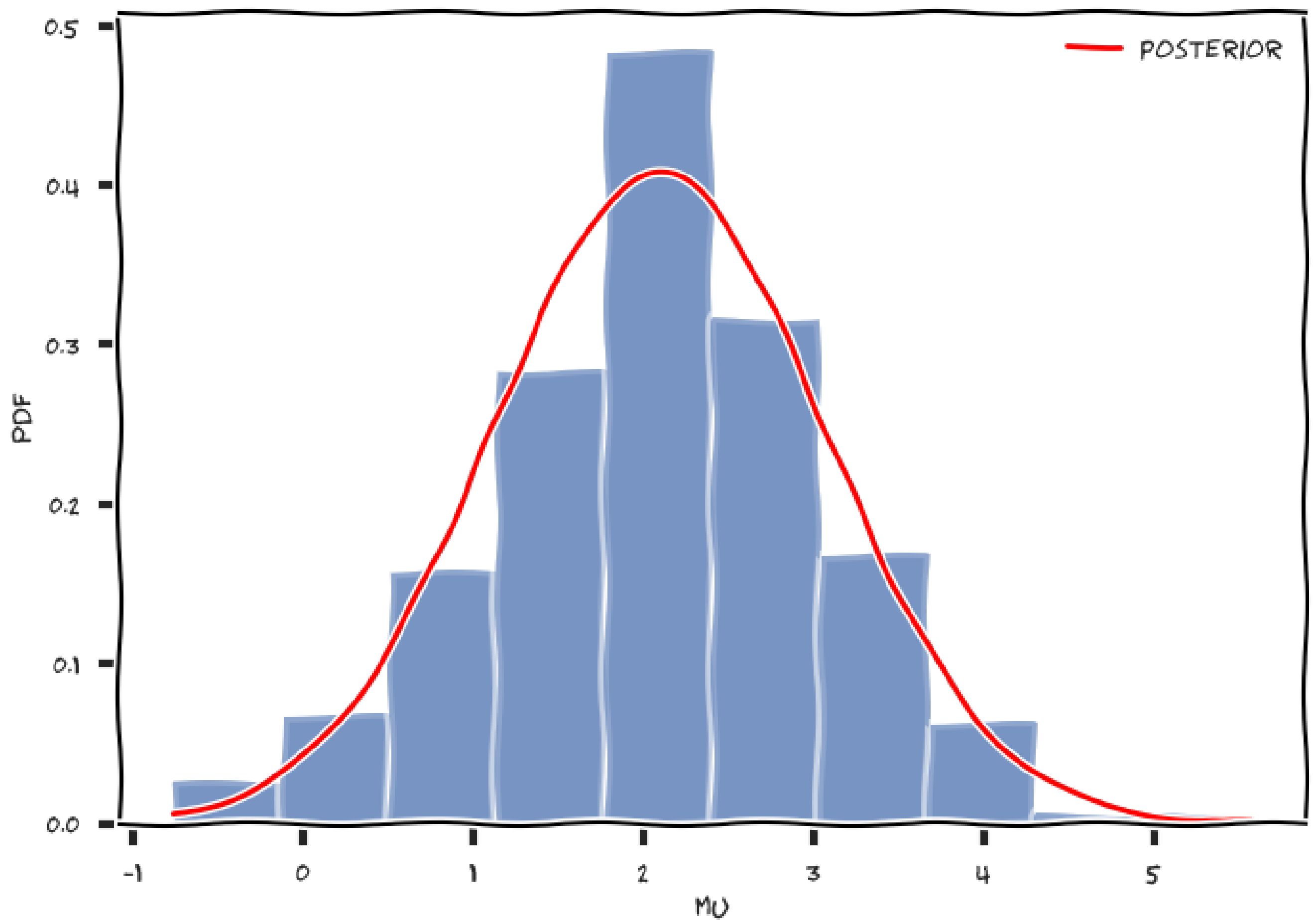
Markov Chain Monte Carlo

Chains after burn-in



Markov Chain Monte Carlo

Posterior



Markov Chain Monte Carlo

Important remarks

MCMC is a numerical technique data allow us to sample from a target distribution

Markov Chain Monte Carlo

Summary

1 – Starting point

2 – Proposal distribution

3 – Transition Probability

Metropolis-Hasting
Gibss Sampling (JAGS)
Hamiltonian Monte Carlo (Stan)

- properties of the Markov Chain
- properties of the Transition Probability
- different implementations

A simple Stan model

```
# Fit
toy_data = {}                          # build data dictionary
toy_data['nobs'] = nobs                # sample size
toy_data['x'] = x1                     # explanatory variable
toy_data['y'] = y                      # response variable

# STAN code
stan_code = """
data {
    int<lower=0> nobs;
    vector[nobs] x;
    vector[nobs] y;
}
parameters {
    real beta0;
    real beta1;
    real<lower=0> sigma;
}
model {
    vector[nobs] mu;
    mu = beta0 + beta1 * x;
    y ~ normal(mu, sigma);             # Likelihood function
}
"""

fit = pystan.stan(model_code=stan_code, data=toy_data, iter=5000, chains=3, verbose=False, n_jobs=3)
```

https://www.bayesianmodelsforastrophysicaldata.com/code-4-3-and-4-4

A simple Stan model

```
# Fit
toy_data = {}                          # build data dictionary
toy_data['nobs'] = nobs                # sample size
toy_data['x'] = x1                     # explanatory variable
toy_data['y'] = y                      # response variable

# STAN code
stan_code = """
data {
   int<lower=0> nobs;
   vector[nobs] x;
   vector[nobs] y;
}
parameters {
   real beta0;
   real beta1;
   real<lower=0> sigma;
}
model {
   vector[nobs] mu;
   mu = beta0 + beta1 * x;
   y ~ normal(mu, sigma);              # Likelihood function
}
"""

fit = pystan.stan(model_code=stan_code, data=toy_data, iter=5000, chains=3, verbose=False, n_jobs=3)
```

Output on screen:

Inference for Stan model: anon_model_2fd44c911bfef7c6ae1d9a3b6094940d.
3 chains, each with iter=5000; warmup=2500; thin=1;
post-warmup draws per chain=2500, total post-warmup draws=7500.

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
beta0	1.99	4.8e-4	0.03	1.93	1.97	1.99	2.01	2.04	3100	1.0
beta1	3.00	8.4e-4	0.05	2.90	2.97	3.00	3.03	3.09	3067	1.0
sigma	0.99	1.5e-4	9.4e-3	0.97	0.98	0.99	1.00	1.01	4105	1.0

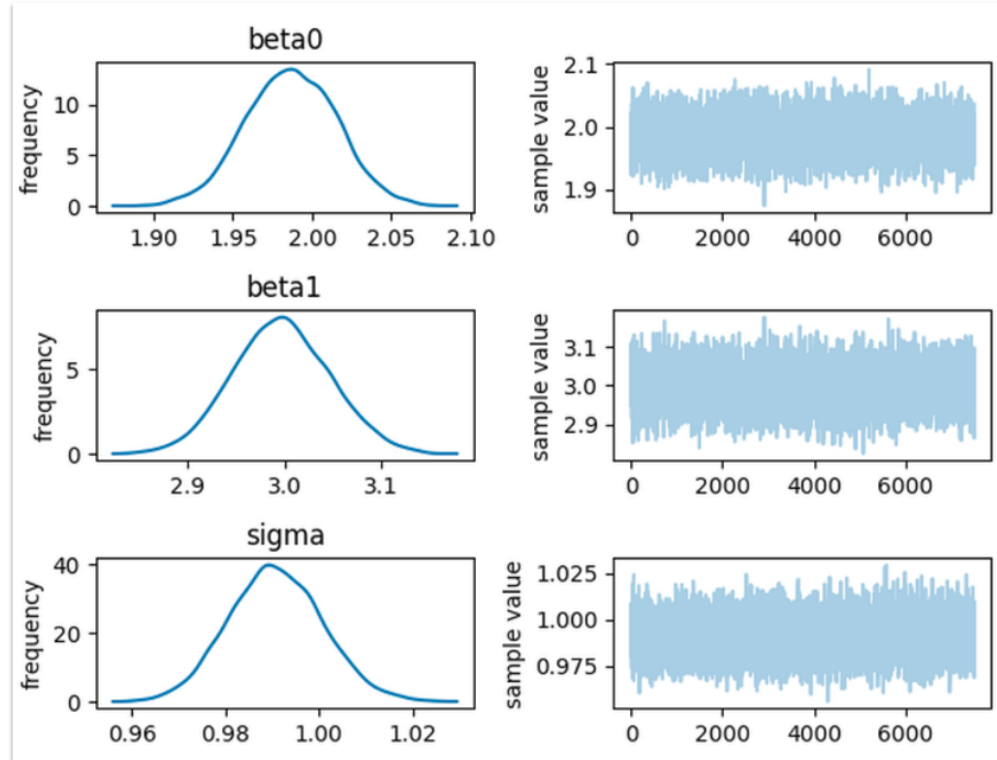
https://www.bayesianmodelsforastrophysicaldata.com/code-4-3-and-4-4

A simple Stan model

```
# Fit
toy_data = {}                        # build data dictionary
toy_data['nobs'] = nobs              # sample size
toy_data['x'] = x1                   # explanatory variable
toy_data['y'] = y                    # response variable

# STAN code
stan_code = """
data {
    int<lower=0> nobs;
    vector[nobs] x;
    vector[nobs] y;
}
parameters {
    real beta0;
    real beta1;
    real<lower=0> sigma;
}
model {
    vector[nobs] mu;
    mu = beta0 + beta1 * x;
    y ~ normal(mu, sigma);           # Likelihood function
}
"""

fit = pystan.stan(model_code=stan_code, data=toy_data, iter=5000, chains=3, verbose=False, n_jobs=3)
```



https://www.bayesianmodelsforastrophysicaldata.com/code-4-3-and-4-4

Check reference list and additional material at

https://github.com/emilleishida/StatisticsInCosmology