Bayesian Statistics: a practical introduction

Charles Margossian
(Flatiron Institute)

“Bayesian inference is a flexible and powerful approach to modeling reality, making optimal predictions from data, and quantifying uncertainty in a coherent manner. Thanks to their versatility, Bayesian methods are now widely used in virtually all fields of science, engineering, and beyond.”

—Alexandre Bouchard-Côté, 2025

“The theory of inverse probability is founded upon a principle which is so simple and so general that it may be applied in all cases and all hypotheses.”

—Pierre-Simon Laplace, 1814

Goals:

- Understand what Bayesian analysis is.
- Understand how Bayesian computation is done.
- Use the software `Stan` to fit and analyze models.

🧑‍🏫 About me:

- Research Fellow at the Flatiron Institute, New York 🇺🇸
- Professor of Statistics at the University of British Columbia, Vancouver 🇨🇦
- Core `Stan` developer

Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Model comparisons

Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Model comparison

What is a (Bayesian) model?

$$p(y, \theta) = \underline{p(y \mid \theta)}\ \underline{p(\theta)}$$

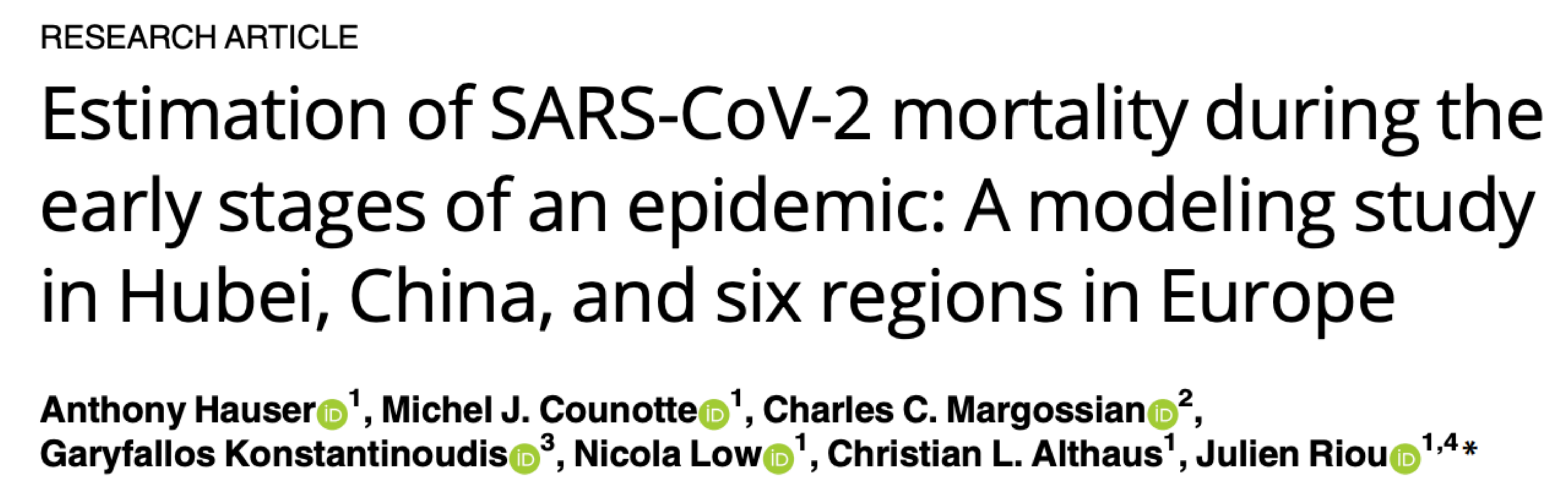
with y observed, θ unknown model parameters.

$p(y \mid \theta)$ is the *likelihood*.

- For a fixed θ, defines a data generating process.

$p(\theta)$ is the *prior*.

- understanding of θ *before* we see the data.
- information from previous analysis, scientific theory, etc.
- regularization tool



RESEARCH ARTICLE

Estimation of SARS-CoV-2 mortality during the early stages of an epidemic: A modeling study in Hubei, China, and six regions in Europe

Anthony Hauser[1], Michel J. Counotte[1], Charles C. Margossian[2], Garyfallos Konstantinoudis[3], Nicola Low[1], Christian L. Althaus[1], Julien Riou[1,4]*

observed *y*:

- reported cases
- hospital deaths

unobserved *θ*:

- transmission rate
- recovery rate
- *f*(*θ*): future cases…

likelihood $p(y \mid \theta)$:

- epidemiological model
- measurement model

prior $p(\theta)$:

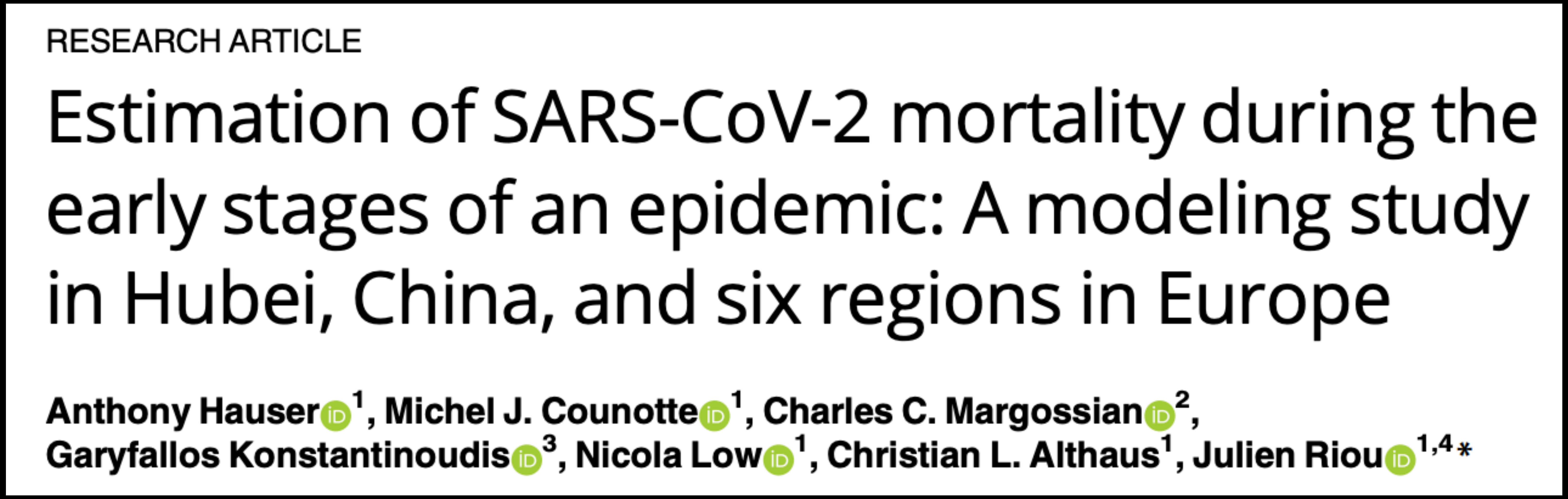
- constraints on interpretable parameters
- meta-analysis for asymptomatic rate

Bayesian inference

Given observations y, want to learn θ.

Proposition: learn a *posterior* distribution.

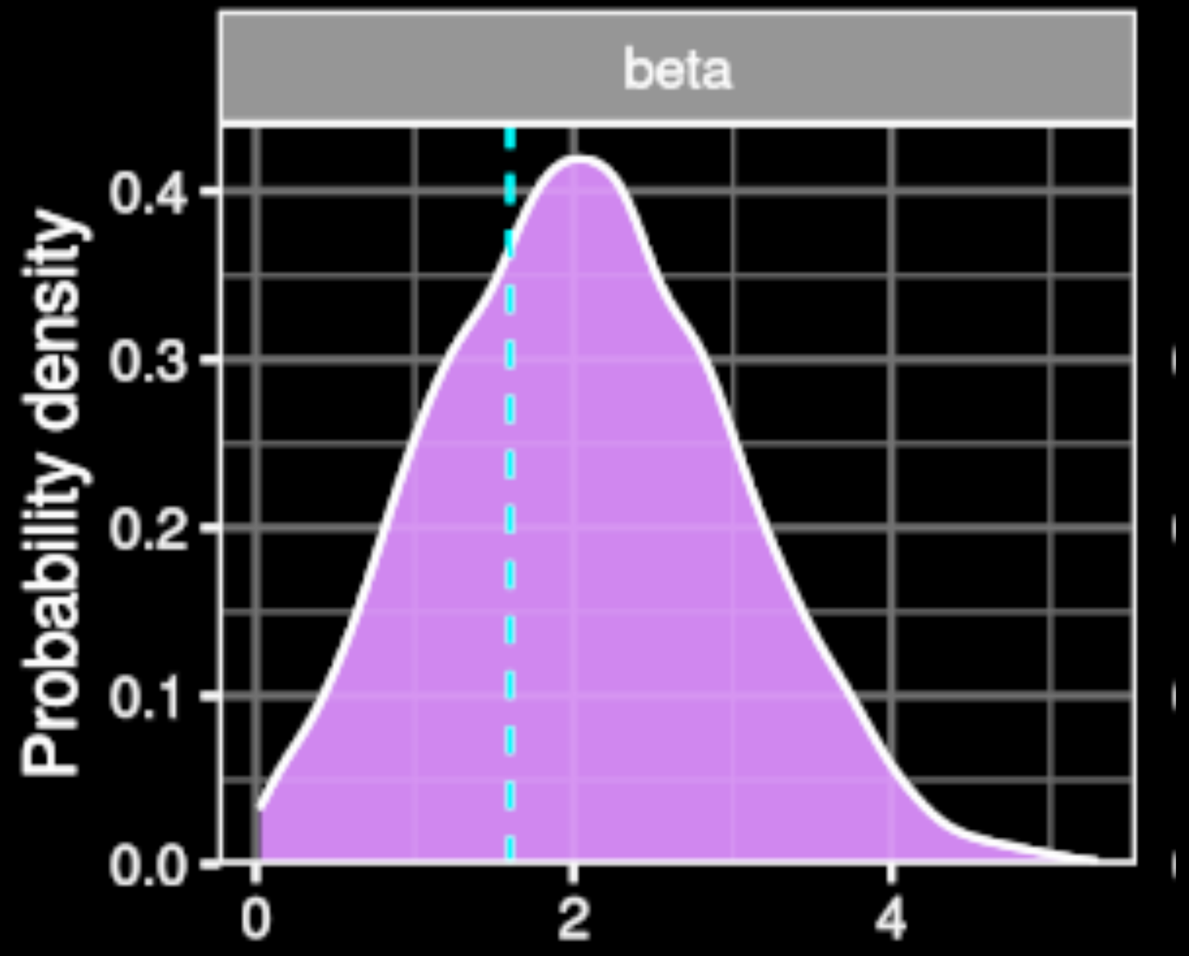
$$\underbrace{p(\theta \mid y)}_{\text{posterior}} = \frac{\overbrace{p(y \mid \theta)}^{\text{likelihood}}\ \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(y)}_{\text{“evidence” (normalizing constant)}}}$$



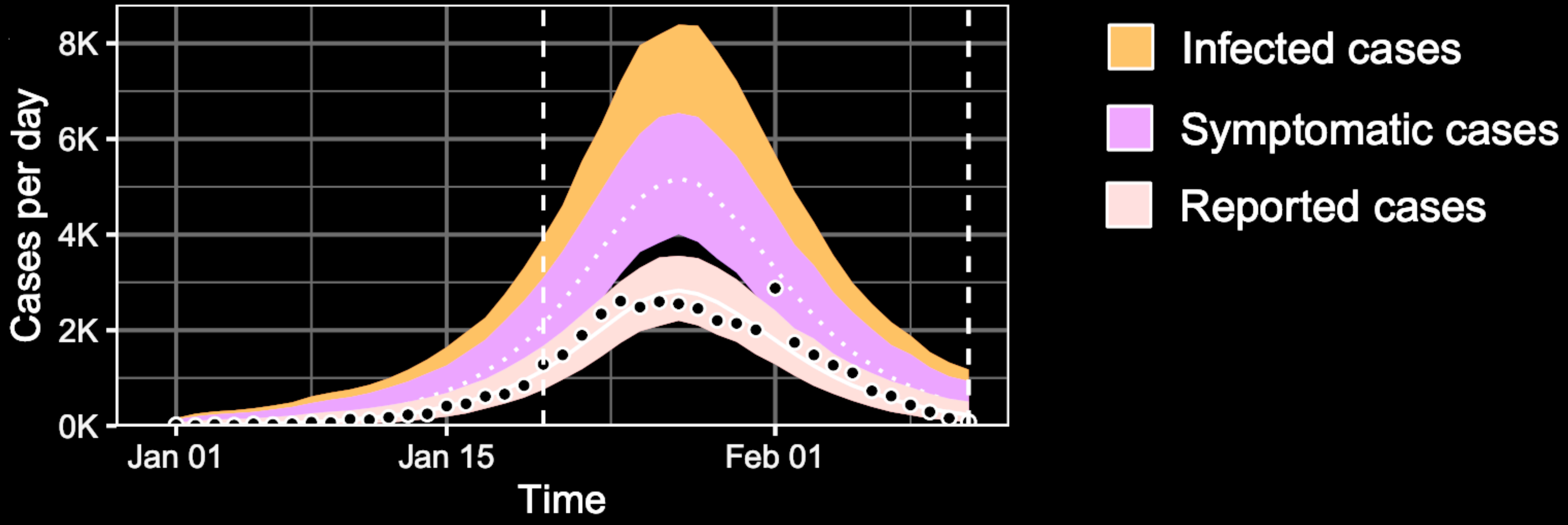
RESEARCH ARTICLE

Estimation of SARS-CoV-2 mortality during the early stages of an epidemic: A modeling study in Hubei, China, and six regions in Europe

Anthony Hauser[1], Michel J. Counotte[1], Charles C. Margossian[2], Garyfallos Konstantinoudis[3], Nicola Low[1], Christian L. Althaus[1], Julien Riou[1,4]*



Posterior of infection rate β



Posterior of infected cases and symptomatic cases

Example: normal-normal model

$p(\theta) = \text{Normal}(\mu, \tau)$

$p(y_i \mid \theta) = \text{Normal}(\theta, \sigma)$

Suppose we have N i.i.d observations, $y_1, \cdots, y_N$.

$$p(\theta \mid y_{1:N}) = \text{Normal}\left(\frac{\mu/\tau^2 + N\bar{y}/\sigma^2}{1/\tau^2 + N/\sigma^2}, \frac{1}{1/\tau^2 + N/\sigma^2}\right)$$

$$p(\theta \mid y_{1:N}) = \text{Normal}\left(\frac{\mu/\tau^2 + N\bar{y}/\sigma^2}{1/\tau^2 + N/\sigma^2}, \frac{1}{1/\tau^2 + N/\sigma^2}\right)$$

🖌️ *Exercise*

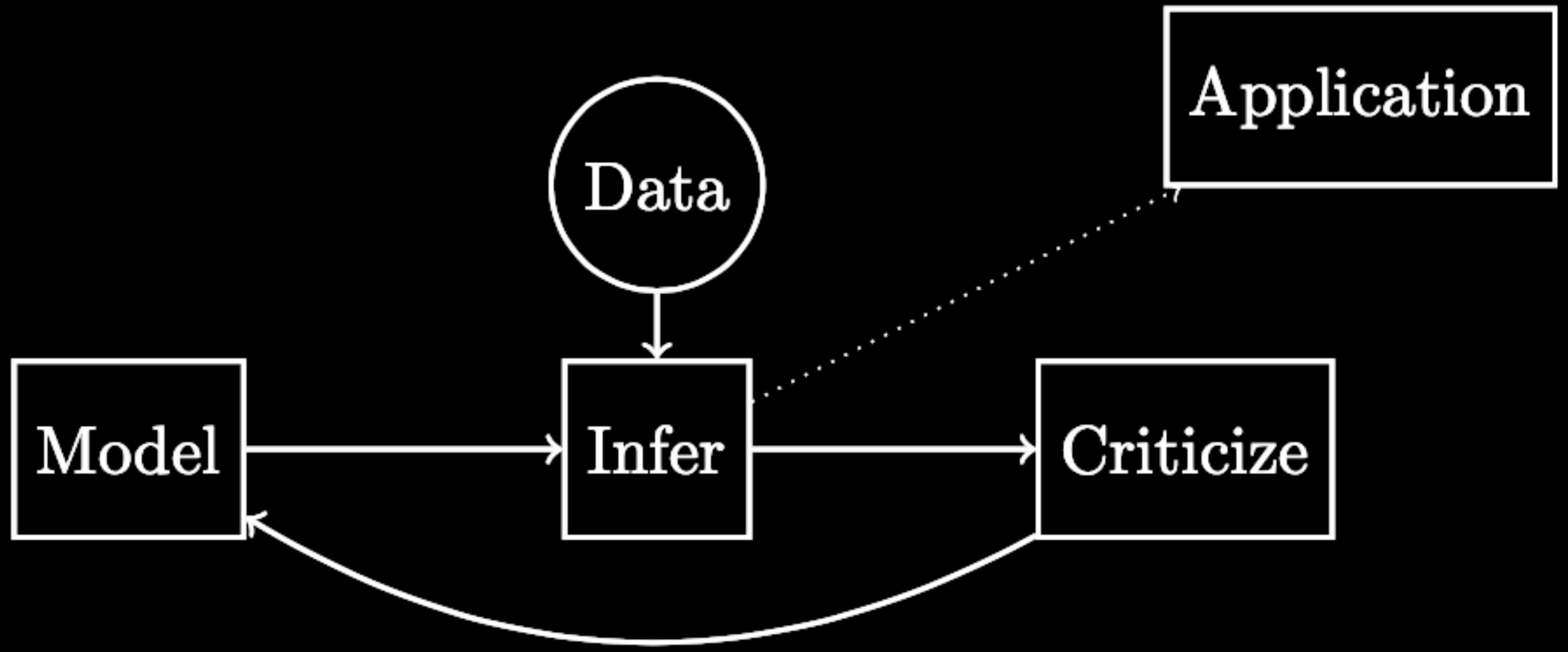
- *Derive the above expression*
- *Show that* $Var(\theta \mid y_{1:N}) \le \tau$ *and* $Var(\theta \mid y_{1:N}) \le \sigma^2/N$.
- *What is the posterior as* $N \to \infty$*?*

Bayesian learning

Suppose we have two independent observations, y_1 and y_2.

$$\begin{aligned} p(\theta \mid y_1, y_2) &\propto p(y_1, y_2 \mid \theta)\ p(\theta) \\ &\propto p(y_1 \mid \theta)\ p(y_2 \mid \theta)\ p(\theta) \\ &\propto p(y_2 \mid \theta)\ p(\theta \mid y_1) \end{aligned}$$

Bayesian workflow



Model

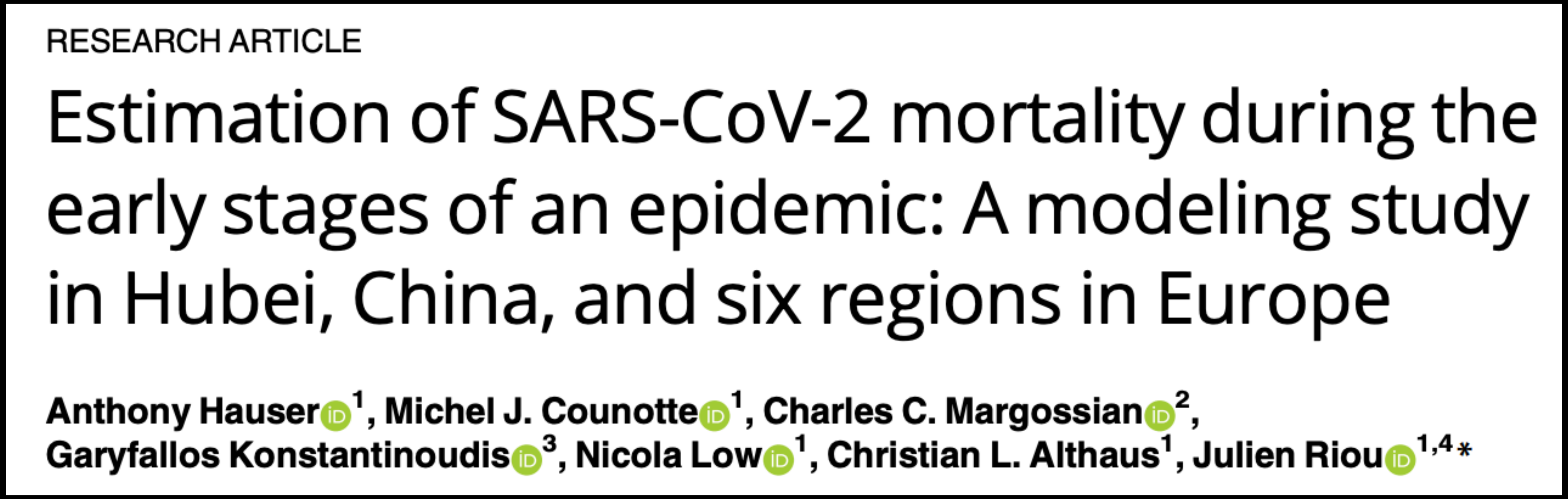
$p(y, \theta) = p(y \mid \theta)\ p(\theta)$

Infer

$p(f(\theta) \mid y)$

Criticize

check inference,
prediction,
cross-validation, etc.



RESEARCH ARTICLE

Estimation of SARS-CoV-2 mortality during the early stages of an epidemic: A modeling study in Hubei, China, and six regions in Europe

Anthony Hauser[1], Michel J. Counotte[1], Charles C. Margossian[2], Garyfallos Konstantinoudis[3], Nicola Low[1], Christian L. Althaus[1], Julien Riou[1,4*]

The published model is the ~15th iteration.

Grinsztajn et al. Bayesian workflow for disease transmission model in Stan, *Statistics in Medicine* (2021)

Gelman et al. Bayesian workflow, *arXiv:2011.01808* (2020)

Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Importance sampling and model comparison

Characterizing the posterior distribution

Expectation values:

$$\mathbb{E}f(\theta) = \int f(\theta)\ p(\theta \mid y)\mathrm{d}\theta$$

Monte Carlo estimator:

$$\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)} \sim p(\theta \mid y)$$

$$\widehat{\mathbb{E}}\, f(\theta) = \frac{1}{N}\sum_{n=1}^{N} f\left(\theta^{(n)}\right)$$

Other summaries: variance, quantiles

How good is our Monte Carlo estimator $\widehat{\mathbb{E}}\, f(\theta)$?

Control expected square error:

$$\mathbb{E}\left[\left(\widehat{\mathbb{E}}\, f(\theta) - \mathbb{E}f(\theta)\right)^2\right] = \underbrace{\left(\mathbb{E}\,\widehat{\mathbb{E}}\, f(\theta) - \mathbb{E}f(\theta)\right)^2}_{\text{Squared bias}} + \underbrace{\text{Var}\left[\widehat{\mathbb{E}}\, f(\theta)\right]}_{\text{variance}}$$

If $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)}$ are i.i.d, the bias is null and $\text{Var}\left[\widehat{\mathbb{E}}\, f(\theta)\right] = \frac{1}{N}\text{Var}\, f(\theta)$.

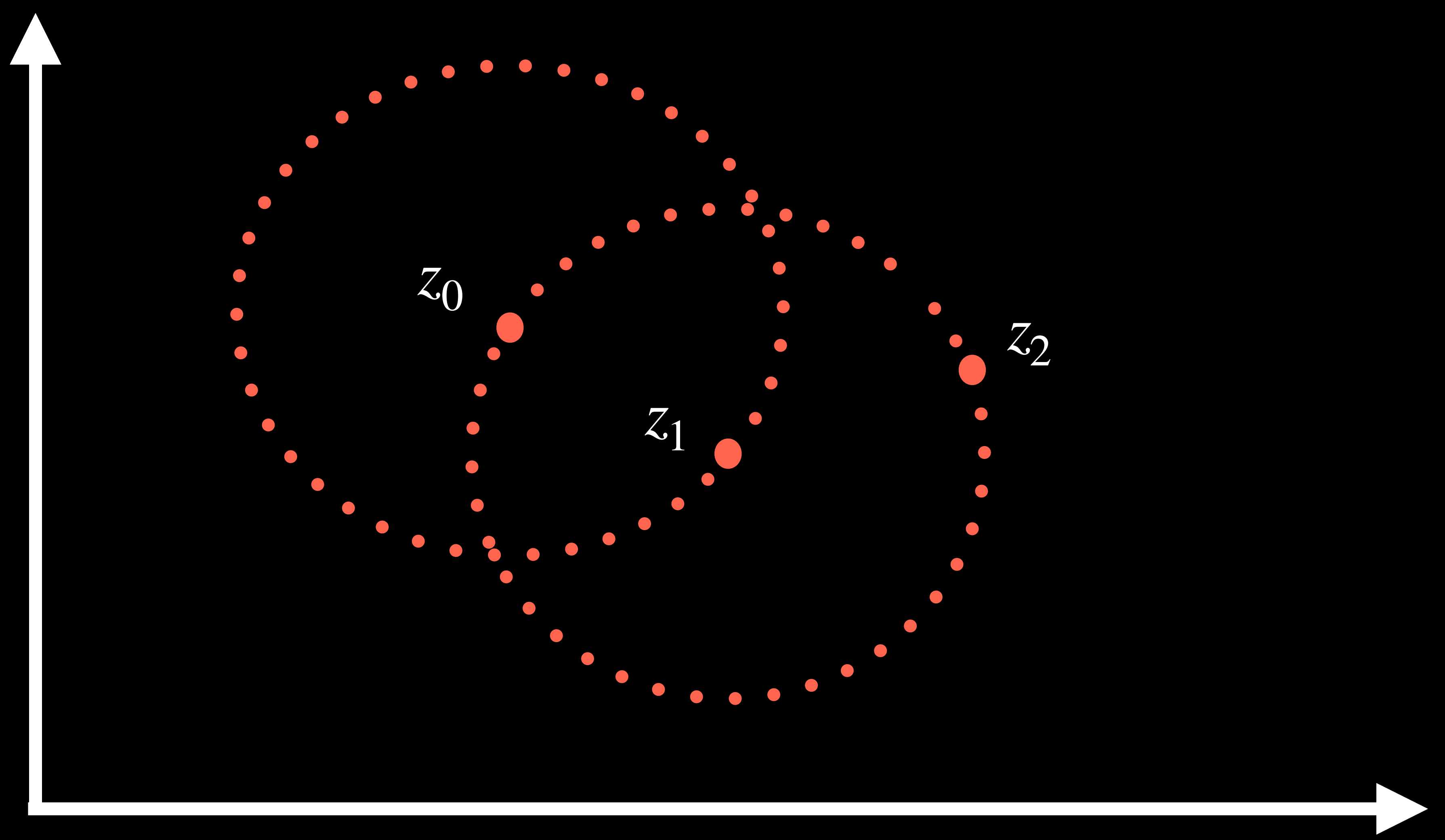
In practice, we cannot generate i.i.d samples from, and so we use Markov chain Monte Carlo.

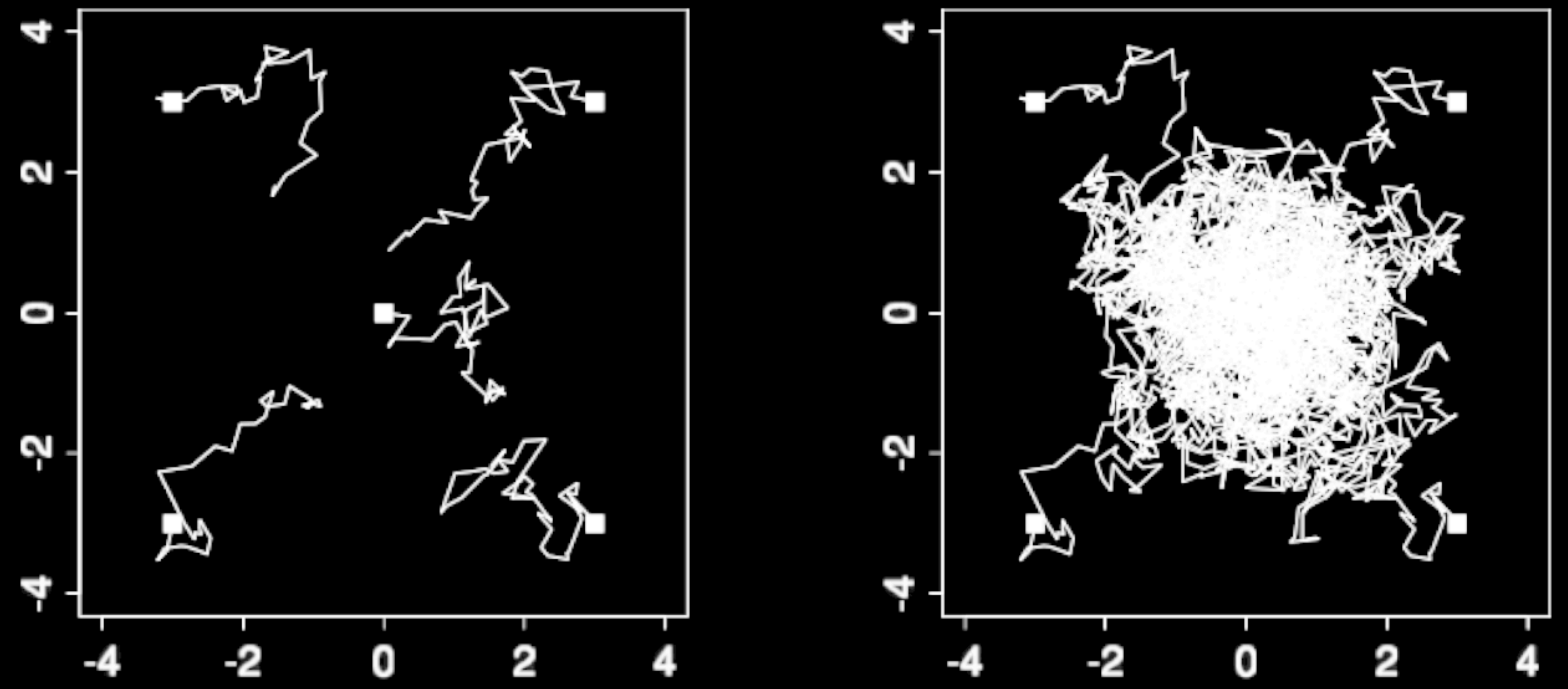
Initialize: $z_0 \sim p_0$

Transition kernel: $\Gamma\left(z^{(i+1)} \mid z^{(i)}\right)$

If we construct Γ carefully

$$\lim_{i\to\infty} z^{(i)} \sim p$$





Metropolis algorithm [Metropolis et al., 1953]

Initialize: $z_0 \sim p_0$

Apply the transition kernel N times:

1. Take a random step from to $\theta^{(i)}$ to propose a new sample $\theta^{(i+1)}$.

2. Accept the proposal with probability

$$\Pr(\text{Accept}) = \min\left(\frac{p(\theta^{(i+1)} \mid y)}{p(\theta^{(i)} \mid y)}, 1\right).$$

Return: $\left(\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)}\right)$**.**

Example: Metropolis algorithm [Metropolis et al., 1953]

Benefits:

1. Only requires evaluating $p(\theta, y) = p(y \mid \theta)\ p(\theta)$
2. Asymptotically, the algorithm samples from $p(\theta \mid y)$.

Drawback:

1. In the finite regime, the samples are biased.
2. The samples are <u>not</u> independent; they are correlated, which increases variance.

Example: Continuous diffusion process

MCMC can be approximated by a Langevin diffusion process [Gelman et al, 1997, Roberts and Rosenthal, 1998].

- Initial distribution: $\pi_0 = \text{Normal}(\mu_0, \sigma_0^2)$
- Target distribution: $\pi = \text{normal}(\mu, \sigma^2)$

Then after time T,

$$\theta^{(T)} \sim \text{normal}[(\mu_0 - \mu)e^{-T} + \mu, (\sigma_0^2 - \sigma^2)e^{-2T} + \sigma^2)]$$

Variance of Monte Carlo estimator

For large N, have a central limit theorem,

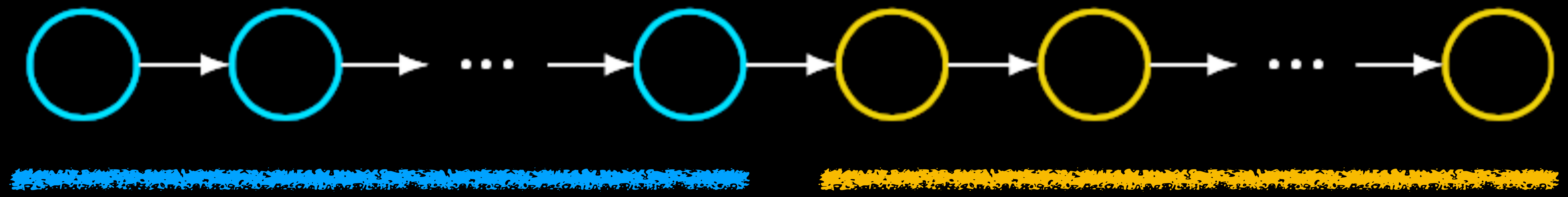
$$\frac{1}{N}\sum_{n} f\left(\theta^{(n)}\right) \approx_{\text{d.}} \text{Normal}\left(\mathbb{E}f(\theta), \frac{\text{Var}f(\theta)}{N_{\text{eff}}}\right),$$

N_{eff} is the effective sample size.

Given autocorrelation ρ_t,

$$N_{\text{eff}} = \frac{N}{1 + 2\sum_{t=1}^{\infty} \rho_t}.$$

Handling the error in MCMC



Warmup: run MCMC and discard samples to make the bias negligable.

Sampling: run MCMC and collect samples to have a large ESS and a low Monte Carlo variance.

Which transition kernel should we use?

Many choices!

Metropolis, Metropolis-Hastings, Gibbs, Hamiltonian Monte Carlo, Langevin diffusion, …

Hamiltonian Monte Carlo

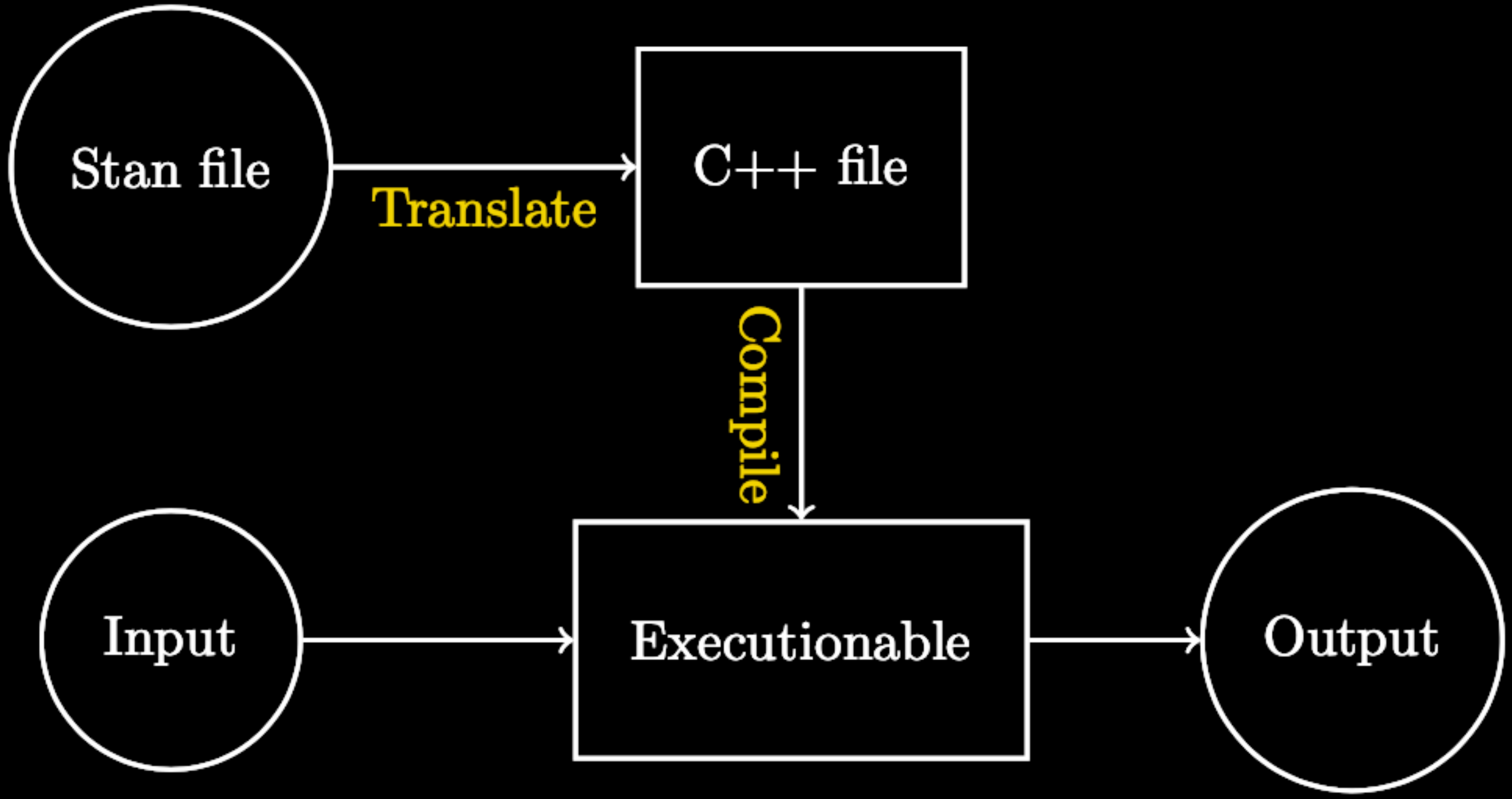
- Scales in high-dimension
- Gradient-based, requires $\nabla_\theta \log p(\theta, y)$
- Difficult to tune!

`Stan` automates the calculations of gradients and provides a self-tuning HMC algorithm.

Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Model comparison

How `Stan` works



- **Stan file:** specifies $p(\theta, y)$.
- **Input:** y, tuning parameters
- **Output:** approx. samples from $p(\theta \mid y)$.
- **Interface:** R, Python, Julia, …

Inference algorithms:

- Hamiltonian Monte Carlo
- No-U Turn sampler
- Laplace approximation
- Variational inference
- …

How we will use Stan

https://stan-playground.flatironinstitute.org/

✅ No need to install Stan on your machine.

❌ Limited functionality: for demo purposes, not full use.

For full Stan capabilities: https://mc-stan.org/

Example: Bayesian linear regression

The data generating process is:

$$p(y \mid \theta) = \mathbf{Normal}(\beta x, \sigma)\,.$$

Goal: estimate $\theta = (\beta, \sigma)$ based on observations (x, y) and prior knowledge on β and σ.

Prior:

$$p(\beta) = \mathbf{Normal}(2{,}1)$$
$$p(\sigma) = \mathbf{Gamma}(1{,}1)$$

Writing the Stan **file**

Stan retains certain C++ features:

- variables need to be declared.
- statement ends with a semi-colon, e.g.

```
real x;
```

The program is divided into blocks:

- **data**: declare the data in the input.
- **parameters**: declare the parameters we want to sample.
- **model**: compute the log joint distribution

Writing the Stan file

```
model {
   target += normal_lpdf(y | beta * x, sigma);

   // or equivalently
   y ~ normal(beta * x, sigma);
}
```

🧑‍💻 Code demo.

Stan playground link:

https://stan-playground.flatironinstitute.org?project=https://gist.github.com/charlesm93/fef6c7960573d3a3d902f64fdd1d2d37

Check the inference

Are the chains still biased by their initialization?

Start each chain at a different location and check they converge to the same distribution:

- trace plots, density plots
- $\widehat{R}$ diagnostic (aim for $\widehat{R} \leq 1.01$).

Is the variance of our Monte Carlo estimator small enough?

- check the ESS (aim for ESS $\geq$ 100).

Check the trained model

Posterior predictive checks

💡 Each time we draw a sample, $\theta^{(i)} = \left(\beta^{(i)}, \sigma^{(i)}\right)$, simulate data

$$y^{(i)}_{\text{pred}} \sim \text{Normal}\left(x\beta^{(i)}, \sigma^{(i)}\right).$$

Want to study the posterior predictive distribution,

$$p(y_{\text{pred}} \mid y) = \int_{\Theta} p(y_{\text{pred}} \mid \theta)\; p(\theta \mid y) \mathrm{d}\theta.$$

To do this, we'll use the `generated quantities` block.

Improving the model

The posterior predictive check suggest our model can be improved with an intercept parameter.

👨‍💻 *Exercise: add an intercept parameter α, then check the inference and the trained model.*

General resources to use `Stan`

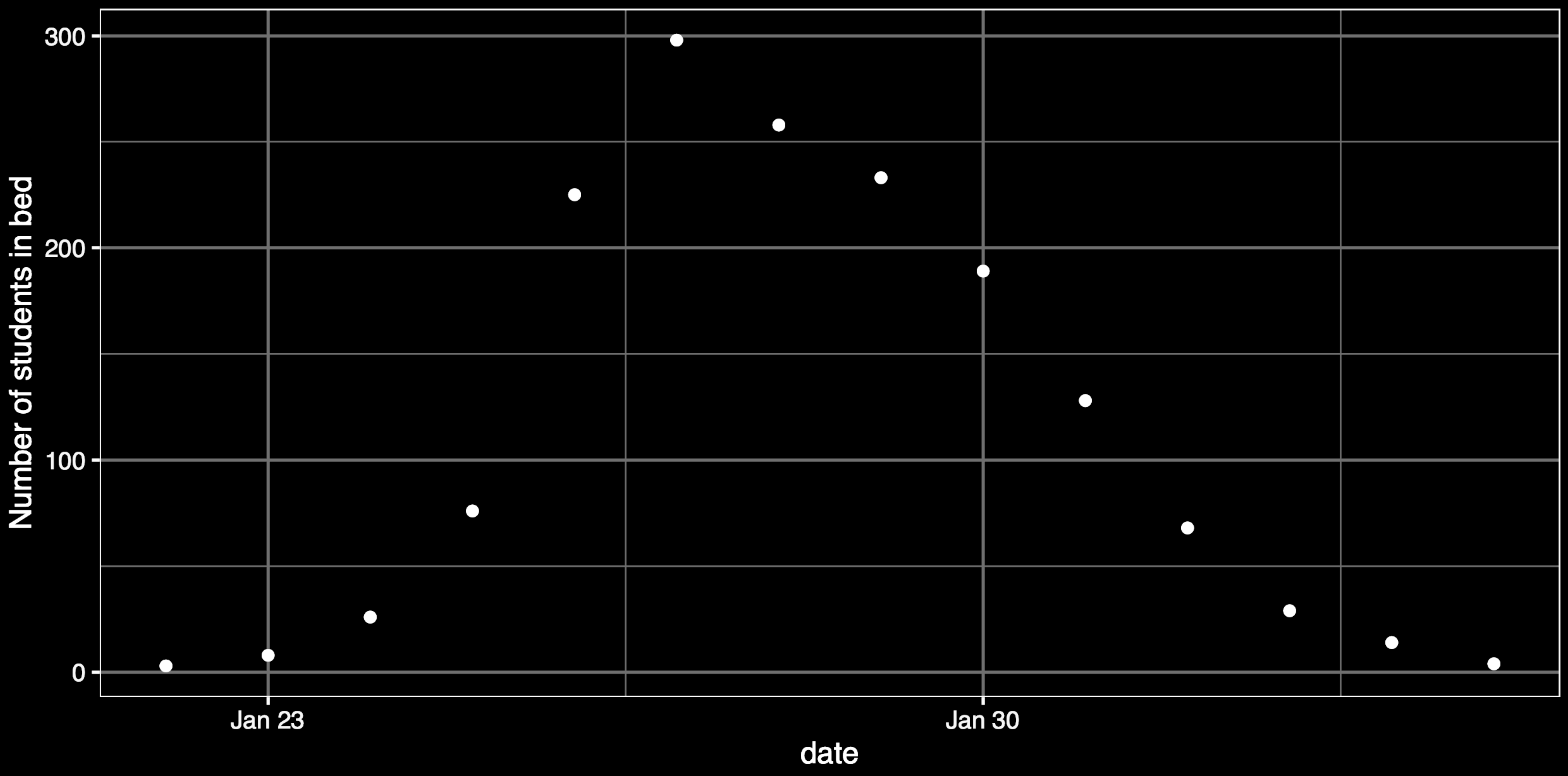
- User's guide (https://mc-stan.org/docs/stan-users-guide/)
- Reference manual (https://mc-stan.org/docs/reference-manual/)
- Functions manual (https://mc-stan.org/docs/functions-reference/)

- Discussion forum (https://discourse.mc-stan.org/)

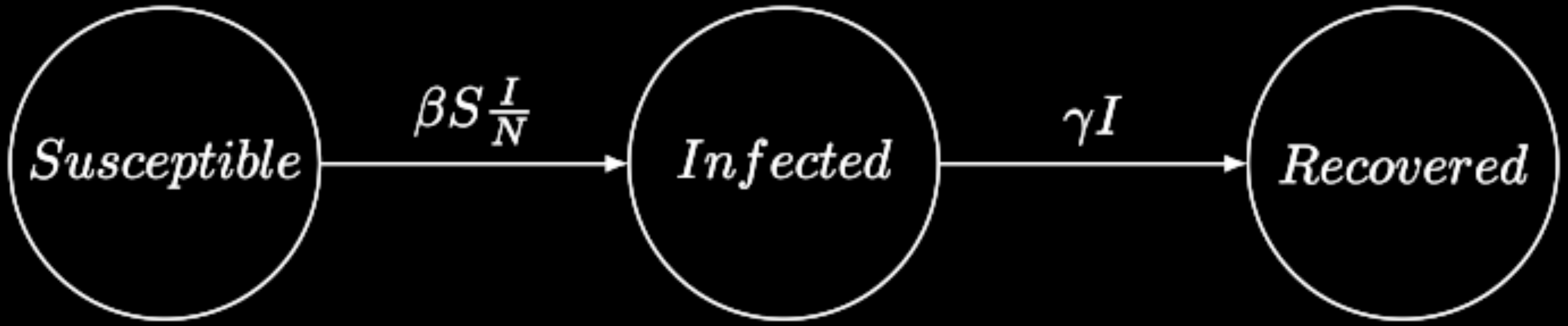
Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Model comparison

1978 influenza outbreak in British boarding school.



Susceptible-Infected-Recovered (SIR) model



$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\beta\ \frac{SI}{N}$$

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \beta\ \frac{SI}{N} - \gamma I$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \gamma I$$

β: transmission rate

γ: rate of recovery of infected individual

$T = 1/\gamma$, recovery time

$R_0 = \beta/\gamma$

Which measurement model should we use?

Poisson likelihood parametrized by $\lambda(t) = I(t)$
with $\mathbb{E}y(t) = I(t)$ and $\mathrm{Var}(y(t)) = I(t)$.

Negative binomial likelihood parametrized by $\mu = I(t)$
with $\mathbb{E}y(t) = I(t)$ and $\mathrm{Var}(y(t)) = I(t) + \dfrac{I^2(t)}{\phi}$.

Which prior should we use?

- $p(\beta)$ = Normal$^+$(2,1): insures $\beta > 0$ and $\Pr(\beta < 4) = 0.975$.
- $p(\gamma)$ = Normal$^+$(0.4,0.5): insures $\gamma > 0$ and $\Pr(\gamma < 1) = 0.9$
- $p(\phi^{-1})$ = exponential(5)

90% of the time,
expect patient to
spend less than
1 one day in bed.

🧑‍💻 Code demo.

🧑‍💻 *Exercise: Write and fit an SIR model for the 1978 influenza outbreak:*

Tip: Code for Poisson: x ~ `poisson(lambda)`

Code for Negative Binomial: `x ~ neg_binomial_2(lambda, phi)`

- *Check the standard diagnostics ($\widehat{R}$ and ESS) and examine the density and trace plots. Is the inference reliable?*
- *Optional: what happens if you increase/reduce the length of the chain?*
- *Do the posterior predictive checks: does the model accurately describe the data.*
- *Report the posterior mean and 90% interval for β, γ, $T = 1/\gamma$ and $R_0 = \beta/\gamma$.*

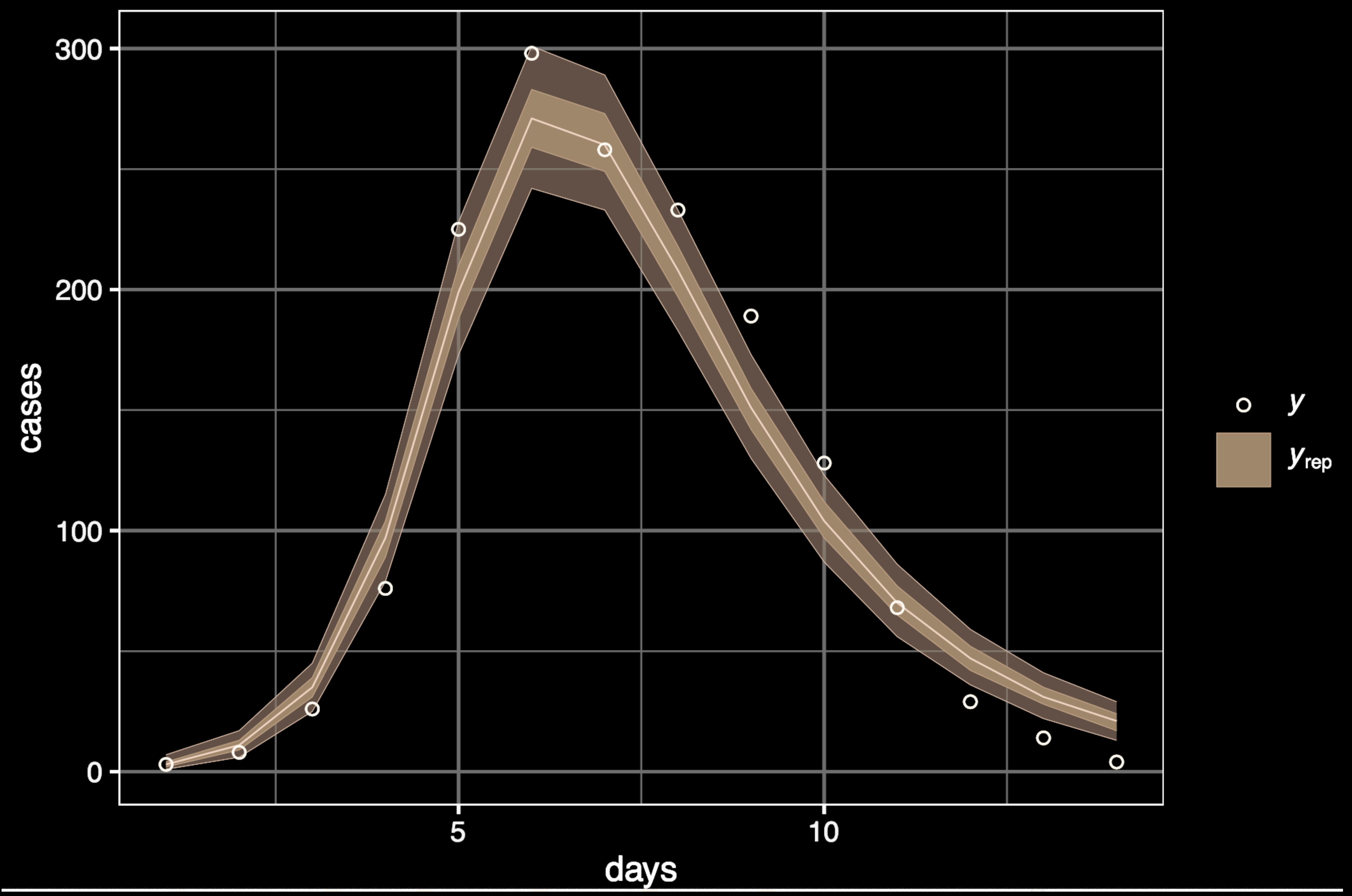
Stan playground link:

https://stan-playground.flatironinstitute.org?project=https://gist.github.com/charlesm93/e29d3a7daaa23569197042357fc96048

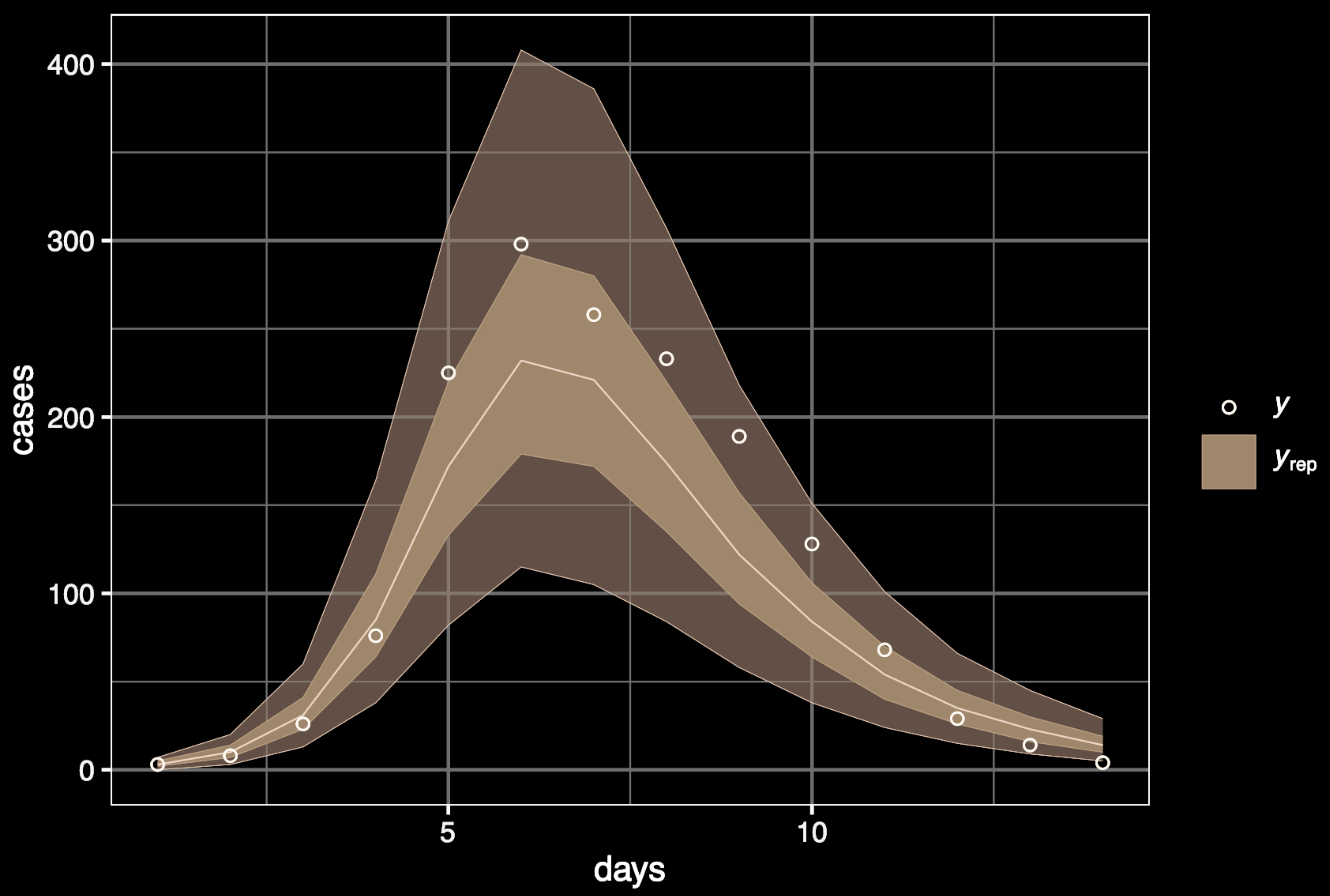
Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Model comparison

Question: for the SIR model, do we get better predictions with the Poisson or the negative binomial likelihood?



Poisson Likelihood



Negative Binomial Likelihood

Question: for the SIR model, do we get better predictions with the Poisson or the negative binomial likelihood?

💡Test model predictions on a validation set:

- Split data into a *training* and *validation* set.

- **Training set:** The data y_{tra} used to learn $p(\theta \mid y_{\text{tra}})$

- **Validation set:** The data y_{val} to "test" model predictions.

Testing predictions

Suppose we have a normal likelihood, with point estimates of the parameters,

$$\text{Normal}(\hat{\mu}(t), \hat{\sigma}).$$

Our best prediction is $\tilde{y}(t) = \mu(t)$.

Then the prediction error is

$$\text{Err} = \left(\hat{\mu}(t) - y_{\text{val}}(t)\right)^2.$$

To account for $\hat{\sigma}$, let's evaluate the *point-estimate log predictive density*,

$$\begin{aligned}\text{p-lpd} &= \log p(y_{\text{val}}(t) \mid \hat{\mu}, \hat{\sigma}) \\ &= const. - \log \hat{\sigma} - \frac{1}{2\hat{\sigma}^2}\left(y_{\mathbf{val}}(t) - \hat{\mu}(t)\right)^2\end{aligned}$$

Testing predictions

Suppose we have a Bernoulli likelihood, with point estimates of the parameters,

$$\text{Bernoulli}(\hat{\pi}(t))\,.$$

Our best prediction is $\tilde{y}(t) = \mathbb{I}(\hat{\pi}(t) > 0.5)\,.$

Then the prediction error is

$$\text{Err} = \mathbb{I}(\tilde{y}(t) = y_{\text{val}}(t)),$$

and the *point-estimate log predictive density*,

$$\begin{aligned}\text{p-lpd} &= \log p(y_{\text{val}}(t) \mid \hat{\pi}(t)) \\ &= y_{\text{val}}(t) \log \hat{\pi}(t) + (1 - y_{\text{val}}(t)) \log(1 - \hat{\pi}(t))\,.\end{aligned}$$

Testing Bayesian predictions

In a Bayesian setting, we don't have a point estimate but a posterior $p(\theta \mid y_{\text{tra}})$.

To be Bayesian, we integrate with respect to the posterior and obtain the *expected log predictive density*,

$$\begin{aligned} \text{elpd} &= \log p(y_{\text{val}}(t) \mid y_{\text{tra}}) \\ &= \log \int_{\Theta} p(y_{\text{val}}(t) \mid \theta)\, p(\theta \mid y_{\text{tra}})\, \mathrm{d}\theta \end{aligned}$$

Testing Bayesian predictions

❓ How do we split the data into a training and a test set?

Proposition: do *leave-one-out cross validation* and compute

$$\mathrm{elpd}_{\mathrm{loo}} = \sum_{i=1}^{N} \log p(y_i \mid y_{-i})$$

Recall

$$p(y_i \mid y_{-i}) = \int_\Theta p(y_i \mid \theta)\, p(\theta \mid y_i)\, \mathrm{d}\theta.$$

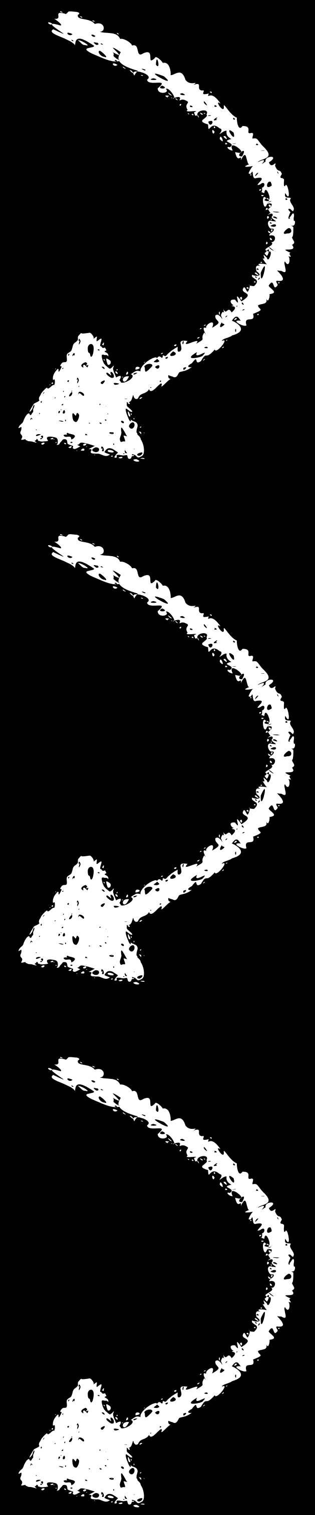
Summary

prediction error based on "best" prediction: $(y_{\text{val}} - \tilde{y})^2$.

point-wise log predictive score: p-lpd = $\log p(y_{\text{val}} \mid \hat{\theta})$

expected log predictive score: elpd = $\log p(y_{\text{val}} \mid y_{\text{tra}})$

$$\text{loo-CV: elpd}_{\text{loo}} = \sum_{i=1}^{N} \log p(y_i \mid y_{-i}) \,.$$



❓ How do we estimate elpd$_{\text{loo}}$ efficiently?

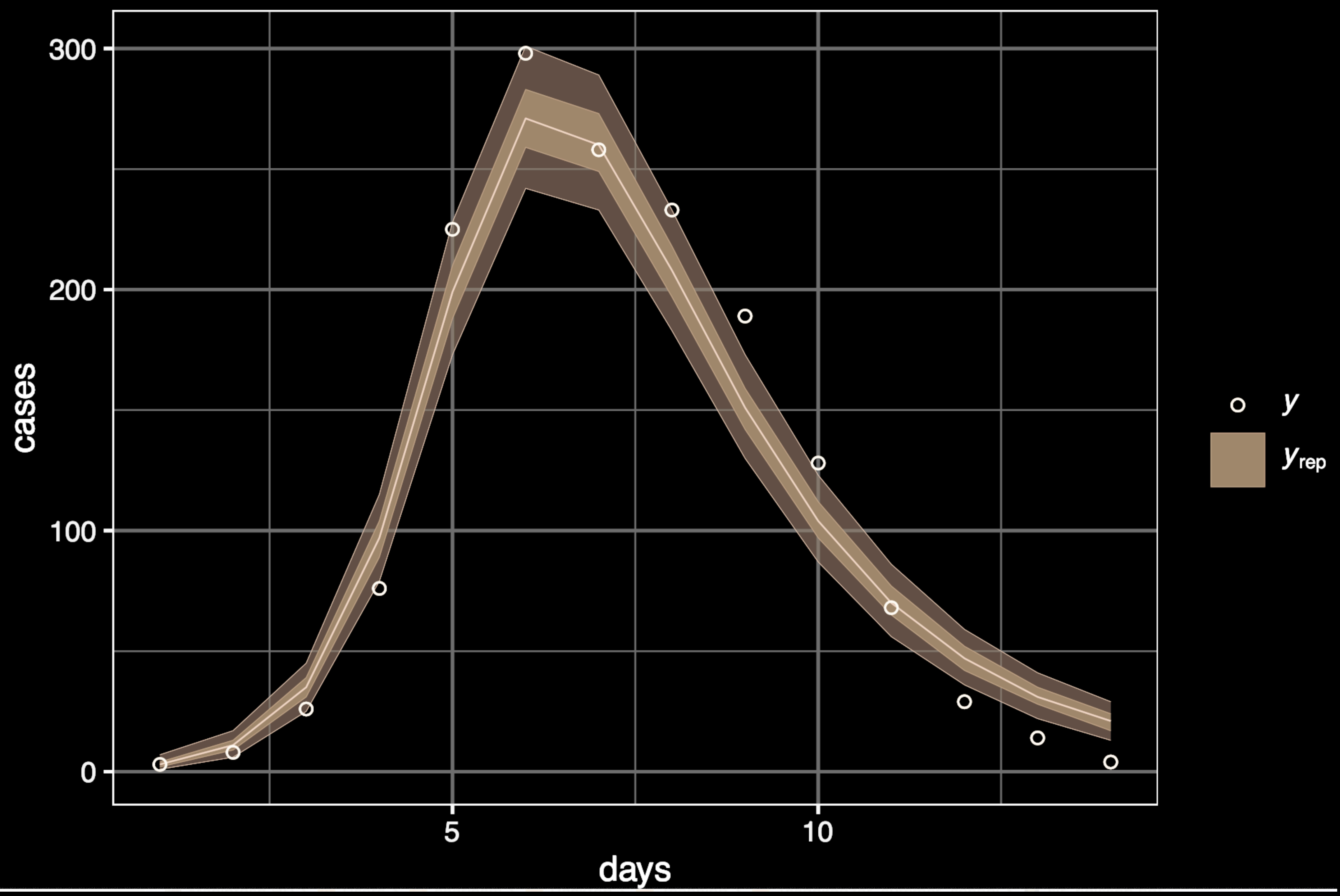
✅ *Pareto-smoothed importance sampling* (PSIS),* using the `R` package `loo`.

❓ Which measurement model is better for the influenza data?

Poison: elp_loo = – 82.5 ± 11
NegBn: elp_loo = – 64.0 ± 5.1

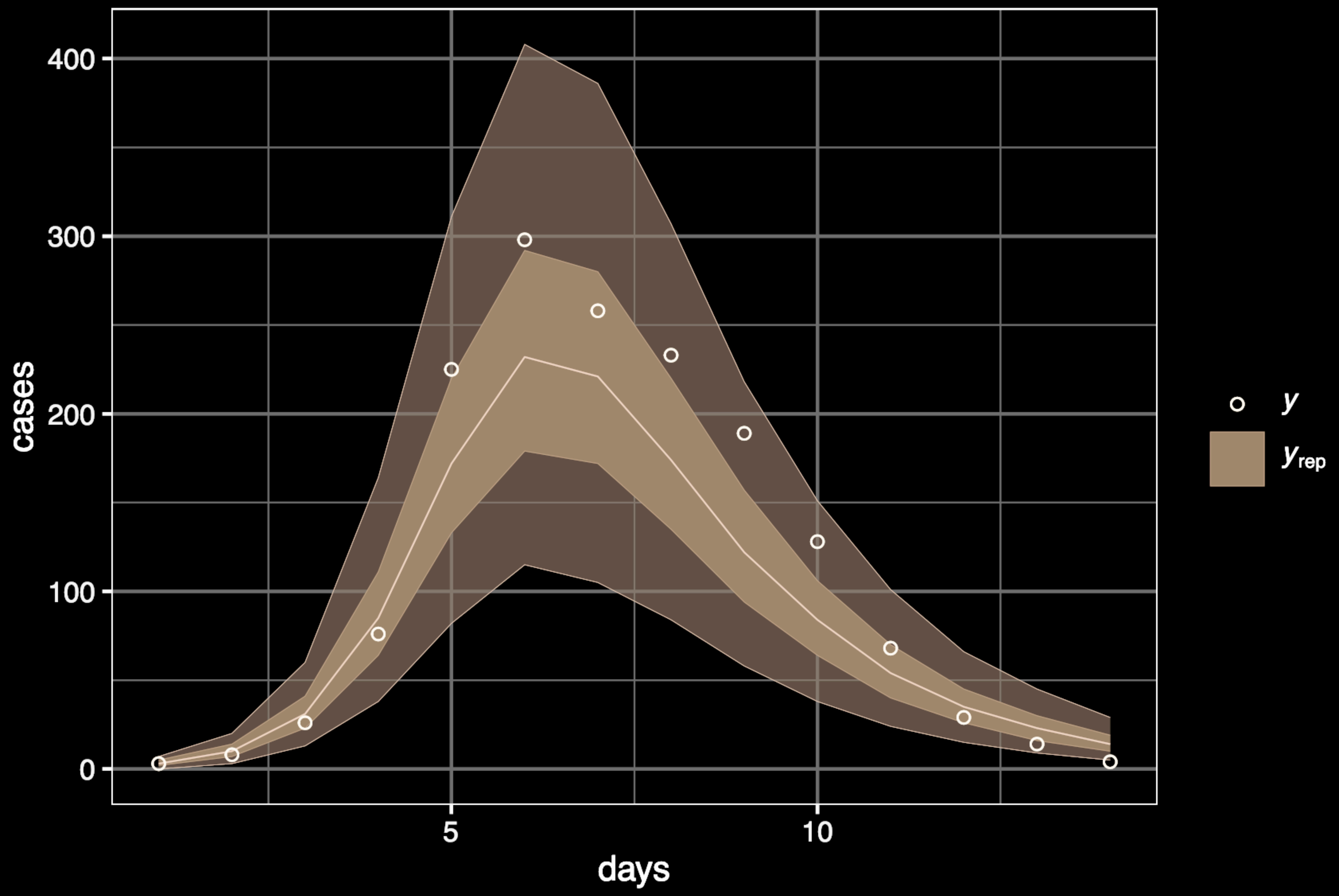
[*] 📄 Vehtari et al. Practical bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing* 2024

Question: for the SIR model, do we get better predictions with the Poisson or the negative binomial likelihood?



Poisson Likelihood

$$\text{elp\_loo} = -82.5 \pm 11$$



Negative Binomial Likelihood

$$\text{elp\_loo} = -64.0 \pm 5.1$$

Outline:

- Basics of Bayesian analysis
- Markov chain Monte Carlo
- Basics of Stan
- Application: Disease transmission model
- Model comparison
- Discussion

What we covered

Bayesian statistics:

- specify model via $p(\theta, y) = p(y \mid \theta)\ p(\theta)$
- estimate unknowns using posterior $p(\theta \mid y)$

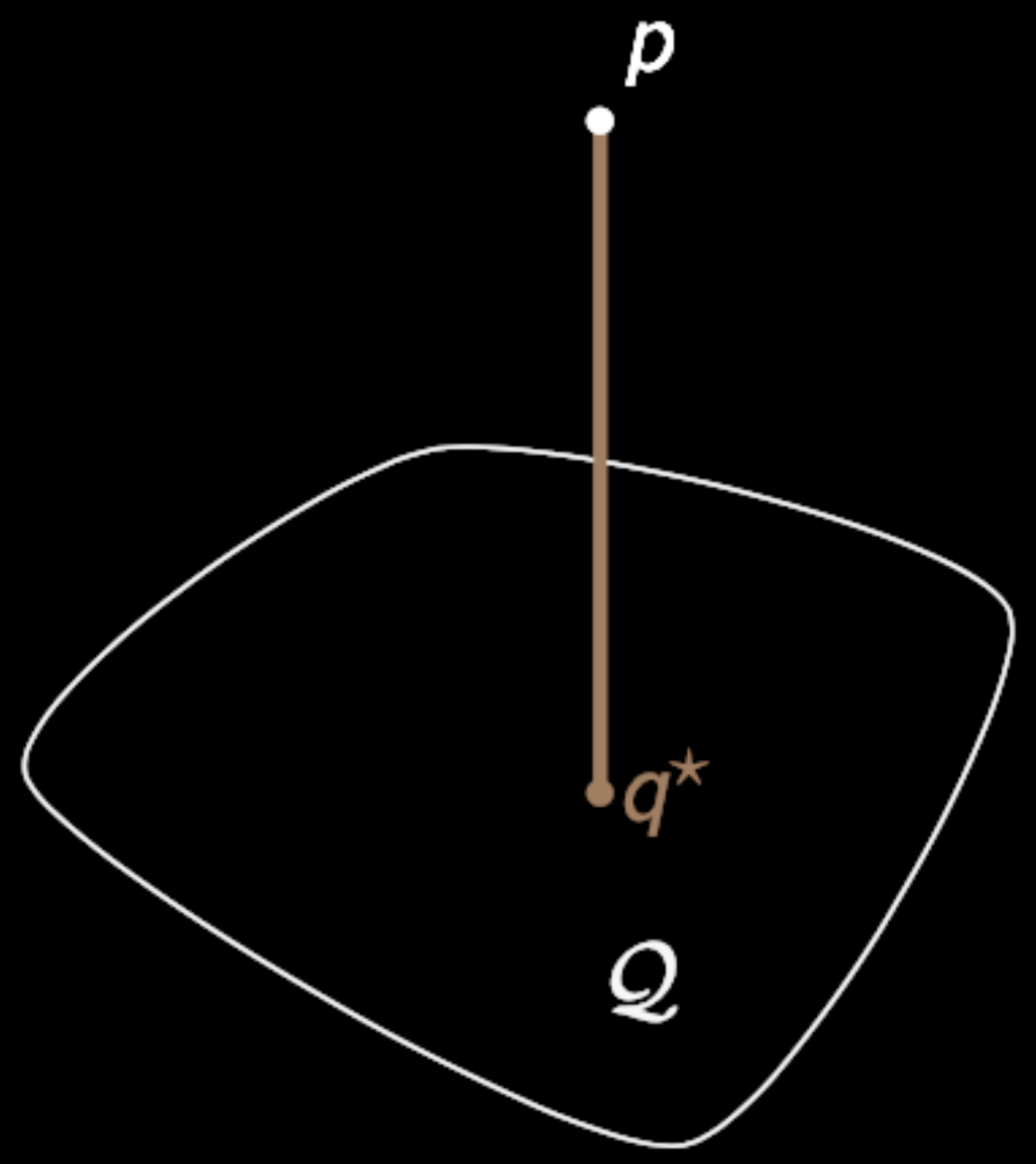
Markov chain Monte Carlo:

- general purpose method to draw from $p(\theta \mid y)$
- computationally expensive!
- efficient implementation in: Stan, PyMC, TensorFlow Prob, …

Bayesian workflow:

- is the inference reliable?
- is the fitted model reliable?
- is our uncertainty well-calibrated?

What we didn’t cover



Modeling techniques:

- prior specification/checking
- hierarchical models: population models, Gaussian processes, spatial models, …

Computation:

- detailed discussion of Hamiltonian Monte Carlo
- Approximate inference, e.g. variational inference
- Efficient algorithms on GPUs
- More ways to check reliability of inference

Where can I learn more?

- 📕 Bayesian Workflow. Gelman et al. *arXiv:2011.01808* (textbook in progress)

- 📄 For how many iterations should we run MCMC? Margossian and Gelman. *Handbook of MCMC 2nd edition (in press)*

- 📄 A conceptual introduction to Hamiltonian Monte Carlo Betancourt. *arXiv:1701.02434*

- 📄 Variational inference: a review for statisticians. Blei et al. *Journal of the American Statistician*

- 📕 *Statistical Rethinking*. McElreath

- 📘 Stan documentation. https://mc-stan.org/docs/

🌍 https://charlesm93.github.io/