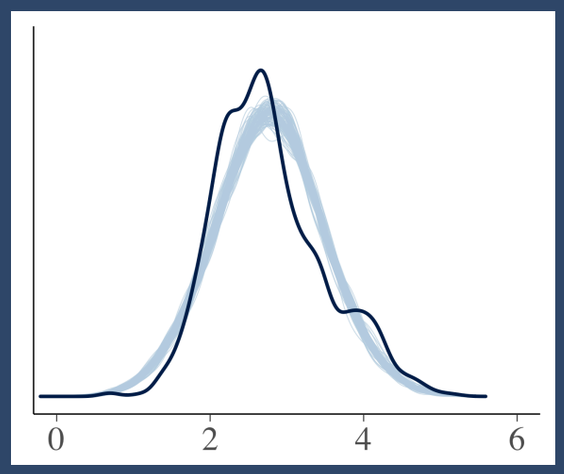
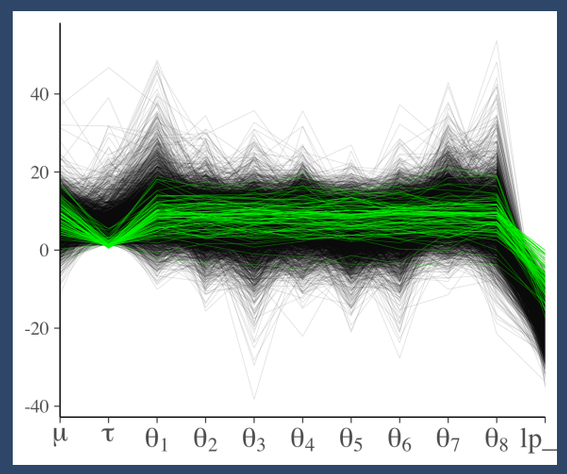
Dipartimento di
Scienze Economiche, Aziendali,
Matematiche e Statistiche “Bruno de Finetti“

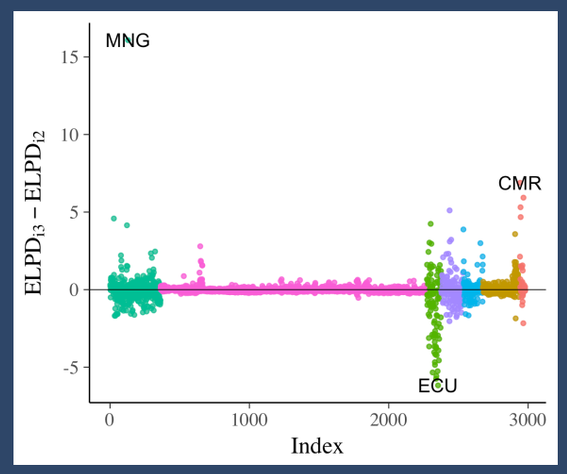
Jonah Sol Gabry
Current Visiting Professor at DEAMS
Applied Statistics Center, Columbia University
Stan Development Team
https://jgabry.github.io/

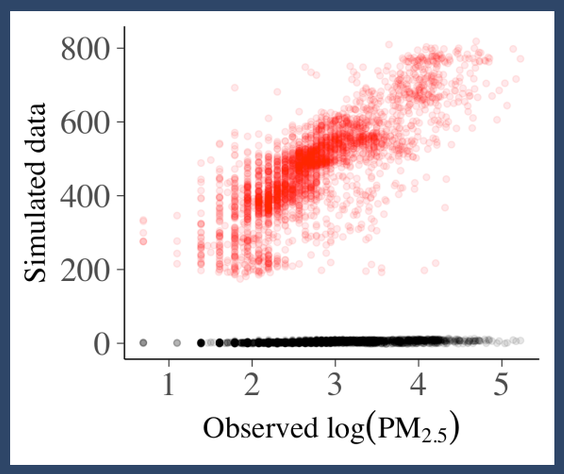
SEMINAR

Bayesian Workflow and the Software That Shapes It



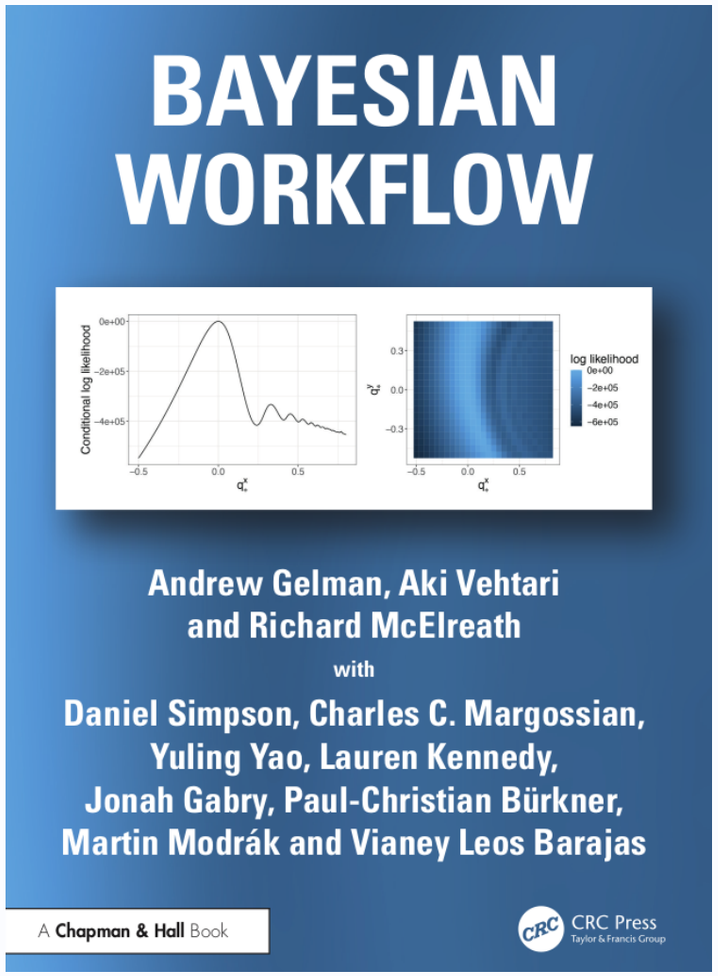






Stan

Software Ecosystem for Modern Bayesian Inference



To be published in June (we hope)

PDF will be free!

- mc-stan.org
- avehtari.github.io/Bayesian-Workflow
- jgabry.github.io

Bayesian inference

A pragmatic perspective

de Finetti —?→ Gelman

subjectivism
(degree of belief of an agent)

pragmatism
(conditional uncertainty inside an approximate model)

- Priors don’t have to be “subjective beliefs”
- Priors provide relevant information, context, regularization
- I don’t have to *believe* θ is gamma distributed, it’s useful
- “*Trust it but don’t die for it*” - L. Egidi

Pragmatic Bayes

A general and versatile framework for learning from data

- Incorporate scientifically relevant information not in the data
- Probability distributions instead of point estimates
- Interpretability
- Flexibility in modeling
- Hierarchical structure, missing data, small data, measurement error, latent variables, regularization, prediction, etc.
- Uncertainty propagation and principled decision making under uncertainty
- And many more

A Bayesian modeler

commits to an a priori *joint* distribution

Data
(observed)

Parameters
(unobserved)

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

$$\propto \quad p(y \mid \theta)\, p(\theta)$$

Generative models

simulate, simulate, simulate

- If we disallow improper priors, then Bayesian modeling is ***generative***

- Generative models can be run forwards or backwards: to generate data or to process data and produce estimates

- For Bayesian workflow a very useful property is that we have a simple way to simulate from *p(y)*:

$$\theta^\star \sim p(\theta)$$

$$y^\star \sim p(y|\theta^\star)$$

$$\Leftrightarrow$$

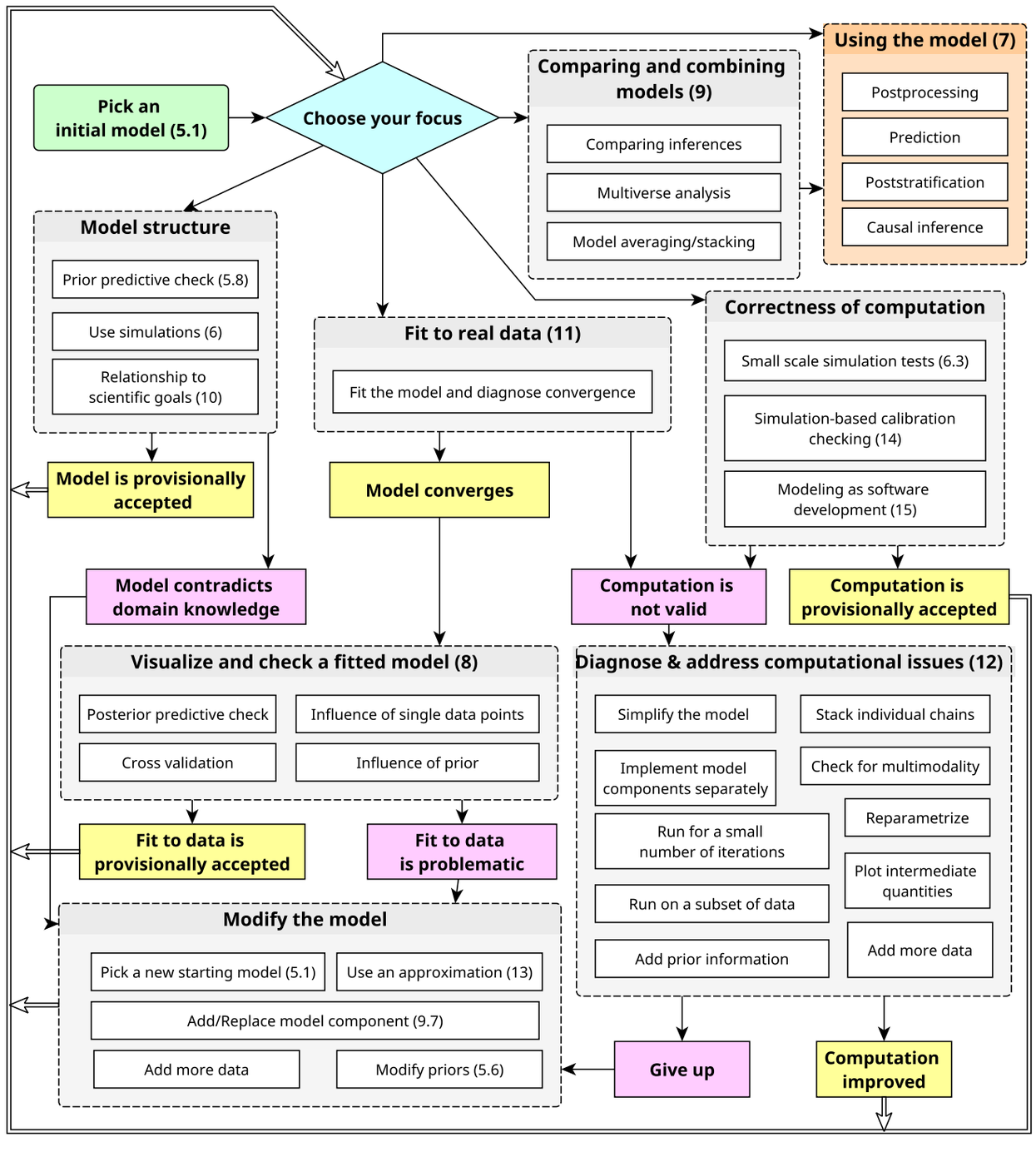
$$y^\star \sim p(y)$$

What is Bayesian Workflow?

Cos'è il flusso di lavoro bayesiano?

Workflow

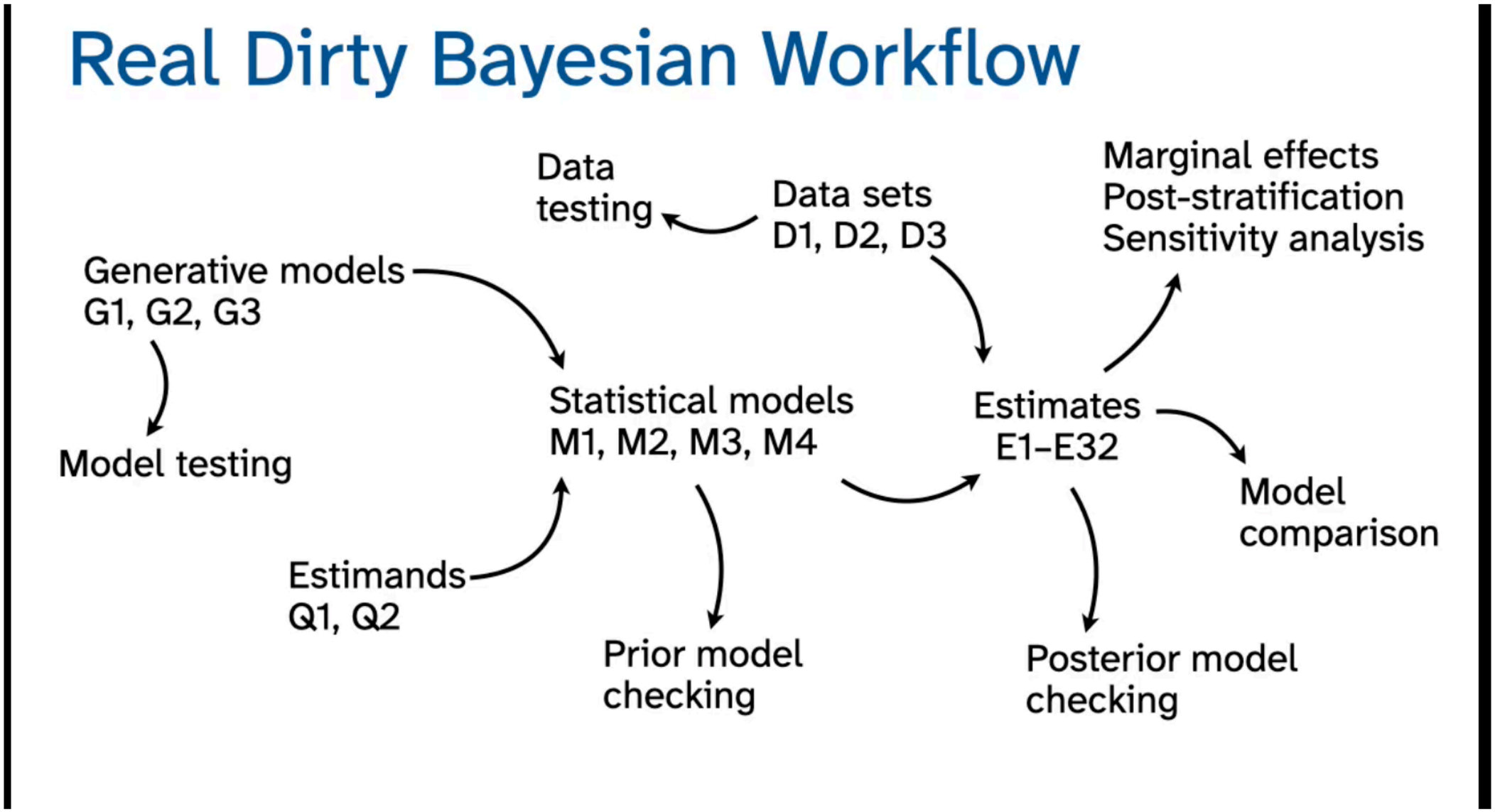
Bayesian data analysis



Source: Bayesian Workflow book

Workflow

Bayesian data analysis



Source: Richard McElreath

Bayesian Workflow

What I work on every day

- I spend most of my time trying to make this easier for you and everyone who wants to use Bayesian methods in practice

- The design choices we make in software shape the workflows of applied Bayesians everywhere!

- Sometimes we realize these choices were not the most convenient or useful for practitioners, so we want your feedback! (if I have time I will say more about this at the end)

Workflow (simplified)

Bayesian data analysis

- Exploratory data analysis
- *Prior* predictive checking
- Model fitting and algorithm diagnostics
- *Posterior* predictive checking
- Model comparison (e.g., via cross-validation)
- Use the models for something in the real world!

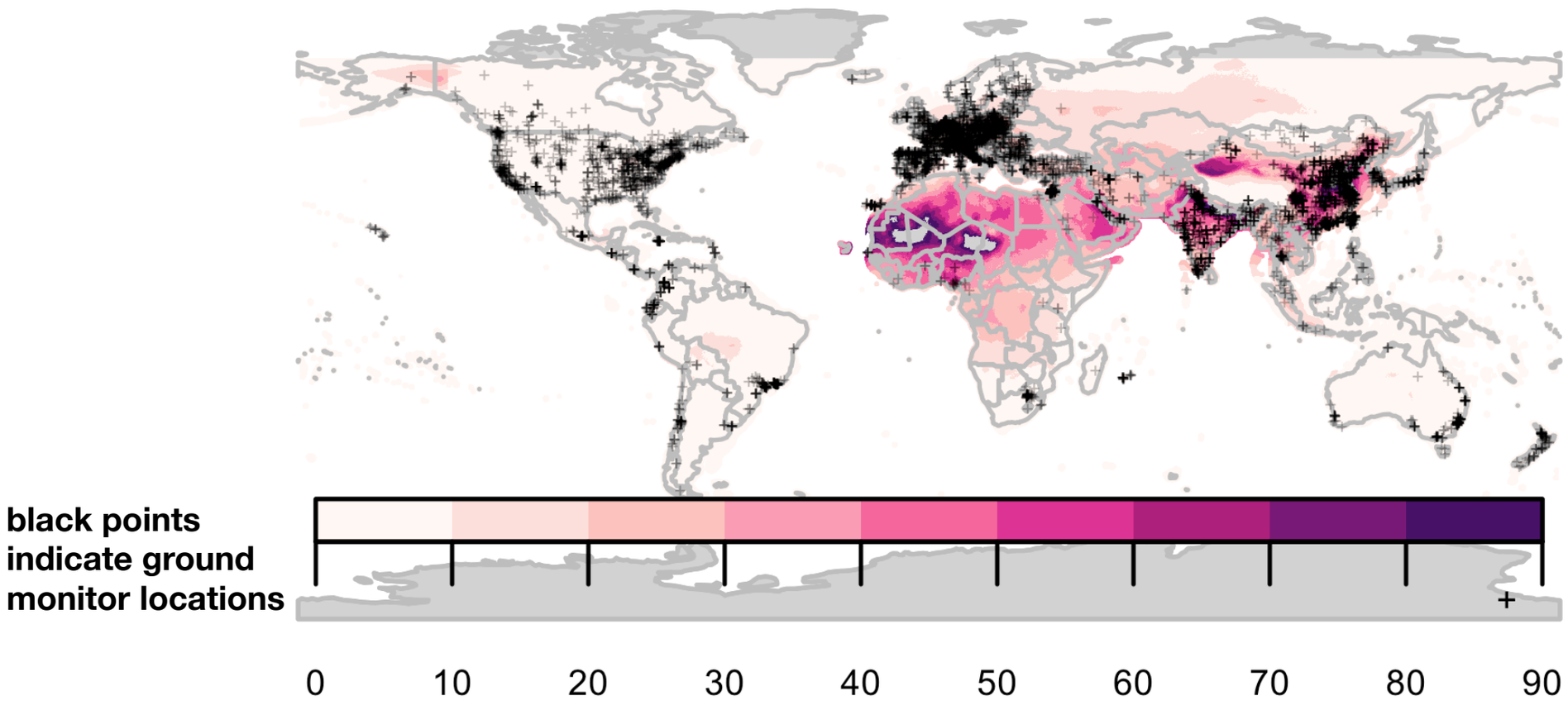
Obligatory question in 2026:
How much of the workflow can/will be automated by AI agents?

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019).
Visualization in Bayesian workflow.
J. R. Stat. Soc. A, 182: 389-402
https://doi.org/10.1111/rssa.12378 | github.com/jgabry/bayes-vis-paper

Example

Goal Estimate global PM2.5 concentration

Problem Most data from noisy satellite measurements (ground monitor network provides sparse, heterogeneous coverage)



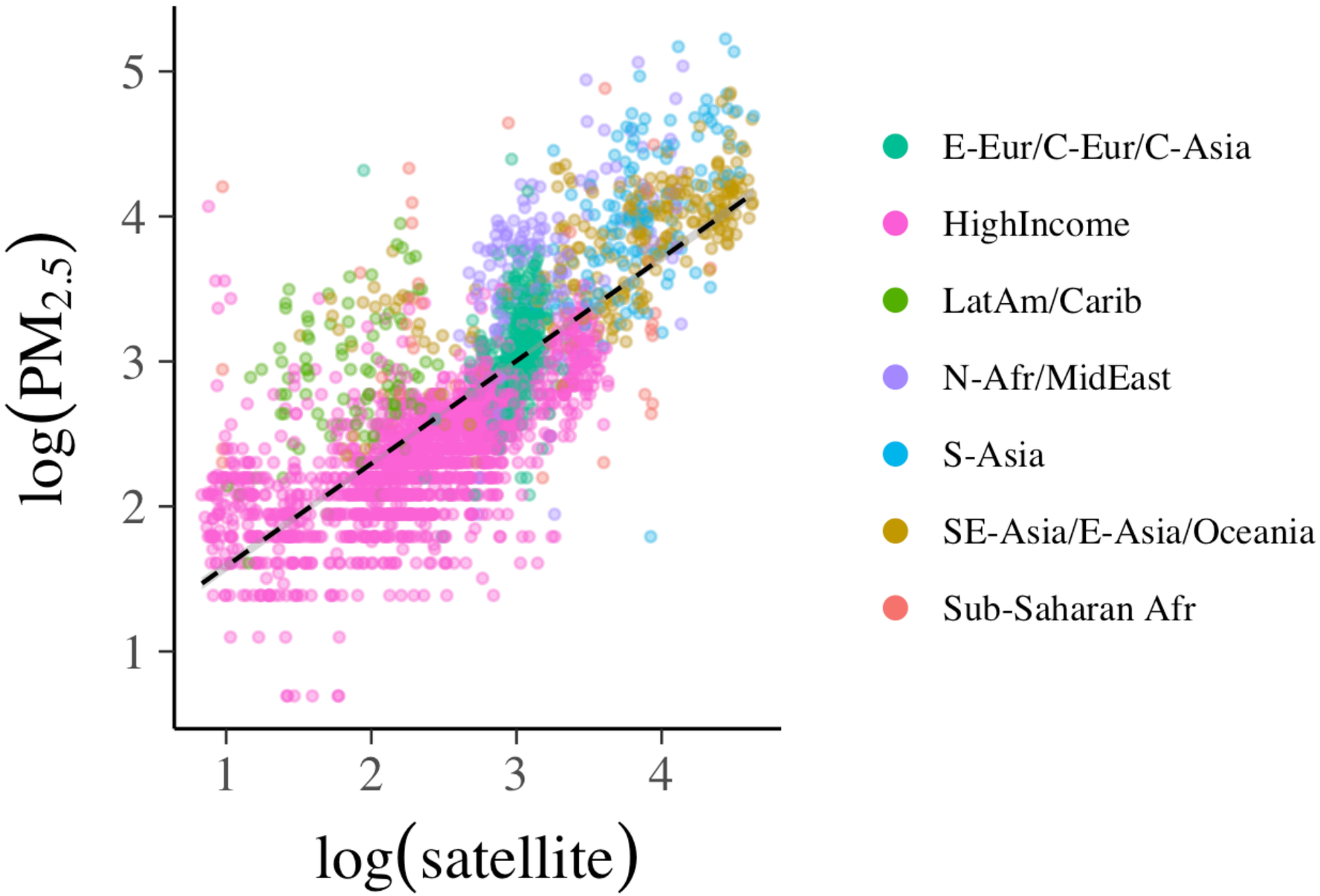
Satellite estimates of PM2.5 and ground monitor locations

Exploratory Data Analysis

Building a network of models

Exploratory data analysis

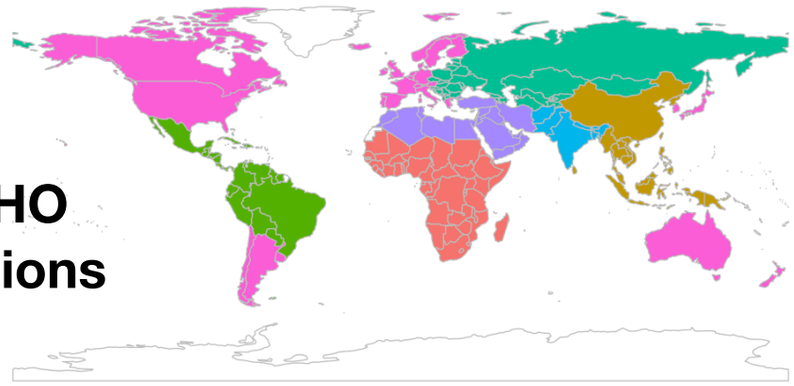
building a network of models



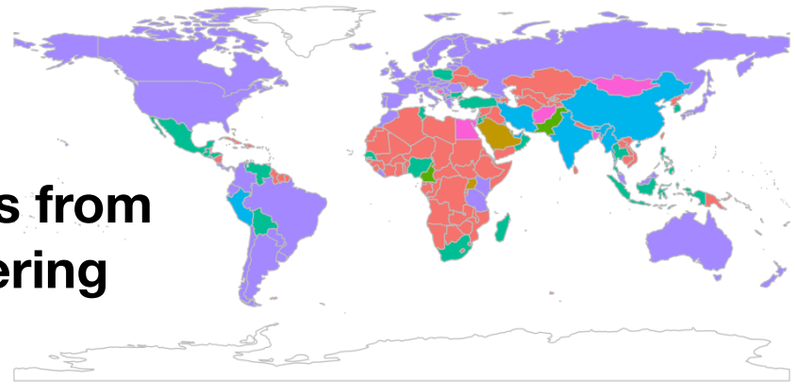
Exploratory data analysis

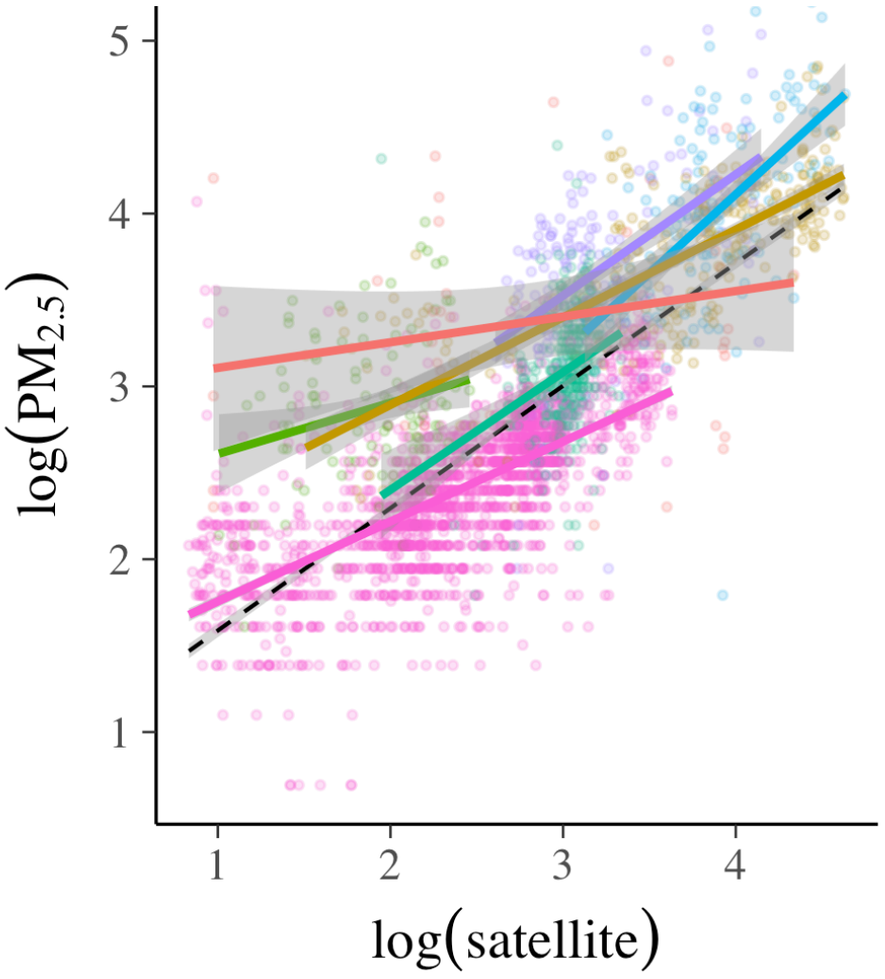
building a network of models

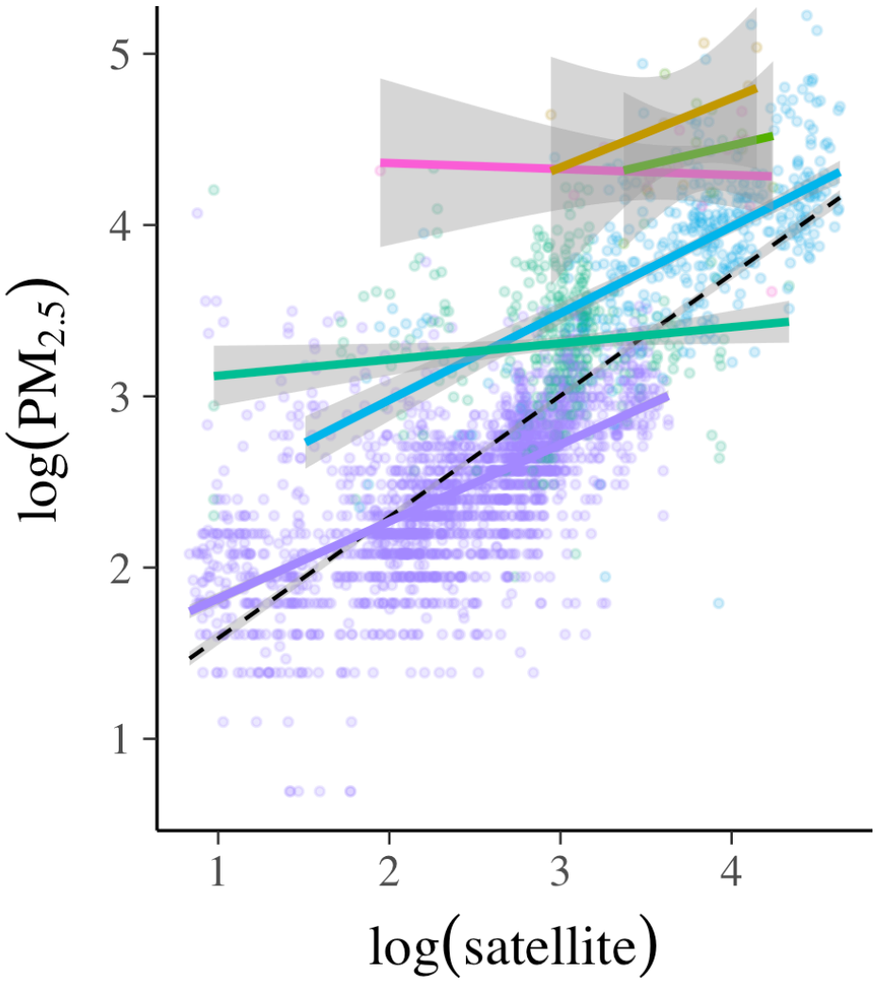
WHO Regions



Regions from clustering







Exploratory data analysis

building a network of models

For measurements $n = 1, \ldots, N$

and regions $j = 1, \ldots, J$

Model 1

$$\log\left(\mathrm{PM}_{2.5,nj}\right) \sim N(\alpha + \beta \log\left(\mathrm{sat}_{nj}\right), \sigma)$$

Exploratory data analysis

building a network of models

For measurements $n = 1, \ldots, N$

and regions $j = 1, \ldots, J$

Models 2 and 3

$$\log\left(\mathrm{PM}_{2.5,nj}\right) \sim N(\mu_{nj}, \sigma)$$

$$\mu_{nj} = \boxed{\alpha_0 + \alpha_j} + \boxed{(\beta_0 + \beta_j)} \log\left(\mathrm{sat}_{nj}\right)$$

$$\alpha_j \sim N(0, \tau_\alpha) \quad \beta_j \sim N(0, \tau_\beta)$$

Prior predictive checks

Fake data can be almost as valuable as real data

Prior predictive checking

fake data is almost as useful as real data

- Used to understand the role of the prior in the data generating process

- Even people who don't have strong feelings about a specific prior know what their observed data *shouldn't* look like. For example: **PM 2.5 levels aren't *so* bad that we're all dead!**

1. Sample from your (proper) prior

2. Use either observed x or reasonable values for x

3. Sample from the **prior predictive distribution** (generate data) given the x values and that particular random draw from the prior

4. Repeat previous steps many times

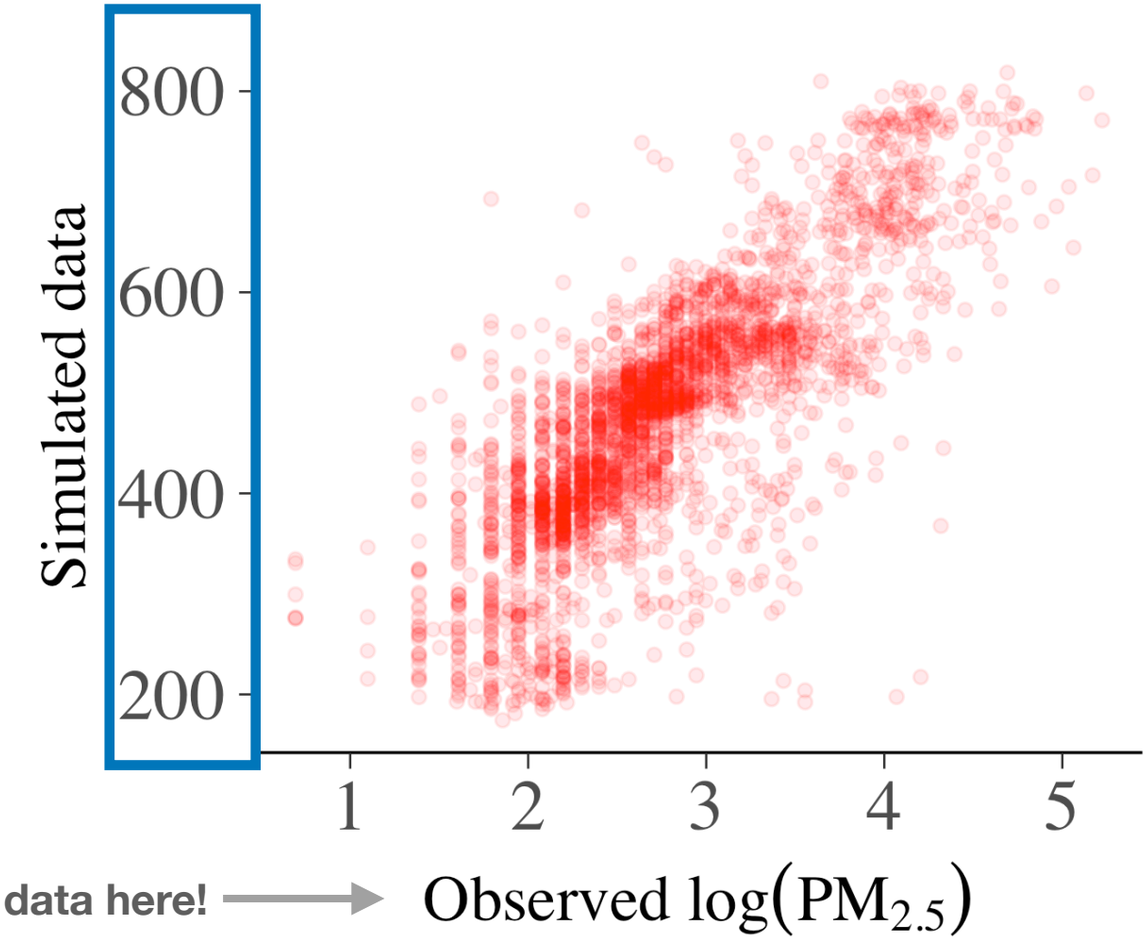
5. Compute summaries / visualize

Prior predictive checking

fake data is almost as useful as real data

What do vague/non-informative priors imply about the data our model can generate?

$$\alpha_0 \sim N(0, 100)$$
$$\beta_0 \sim N(0, 100)$$
$$\tau_\alpha^2 \sim \text{InvGamma}(1, 100)$$
$$\tau_\beta^2 \sim \text{InvGamma}(1, 100)$$

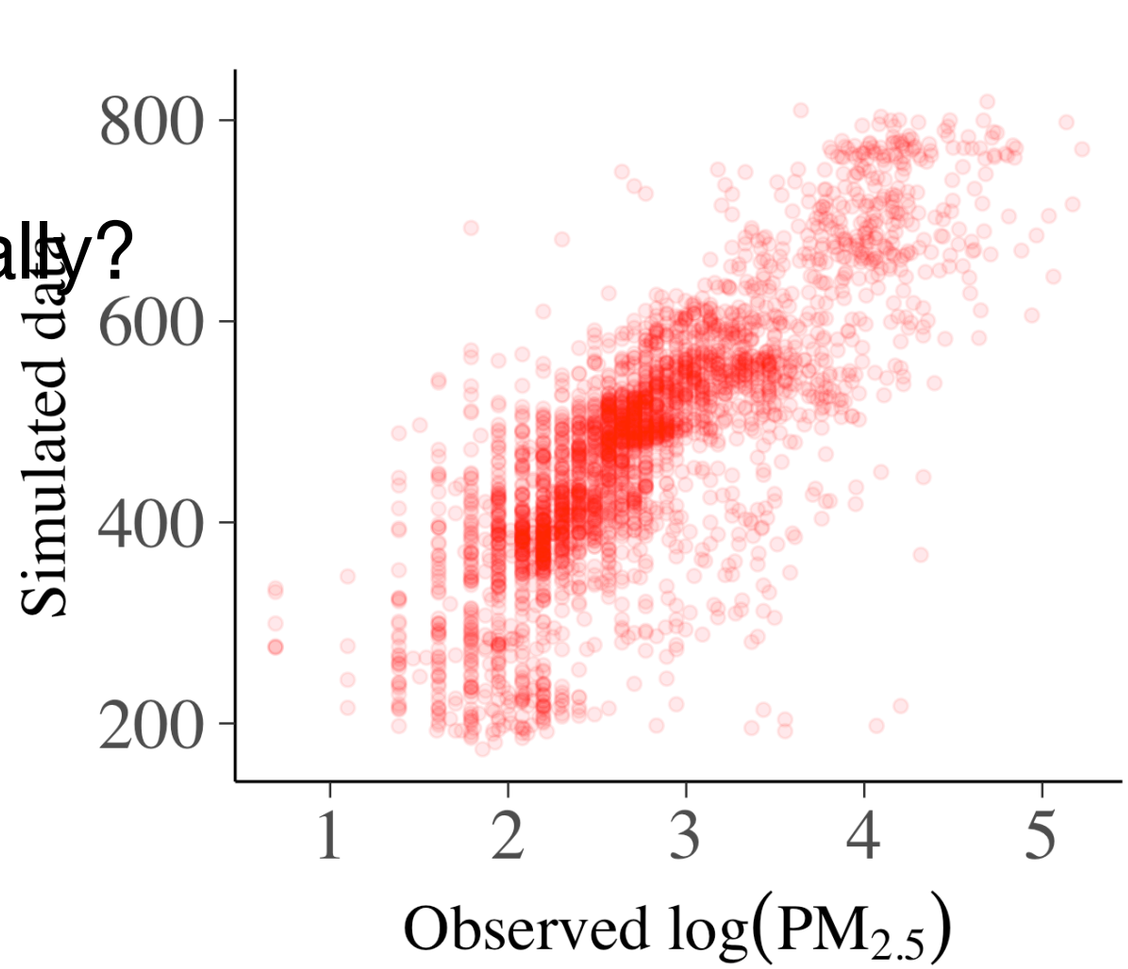


Note: a domain expert should *not* need to look at the real data here!

Prior predictive checking

fake data is almost as useful as real data

- The prior model is **two orders of magnitude** off the real data
- Two orders of magnitude **on the log scale!**
- What does this mean practically?
- The data will have to overcome the prior…

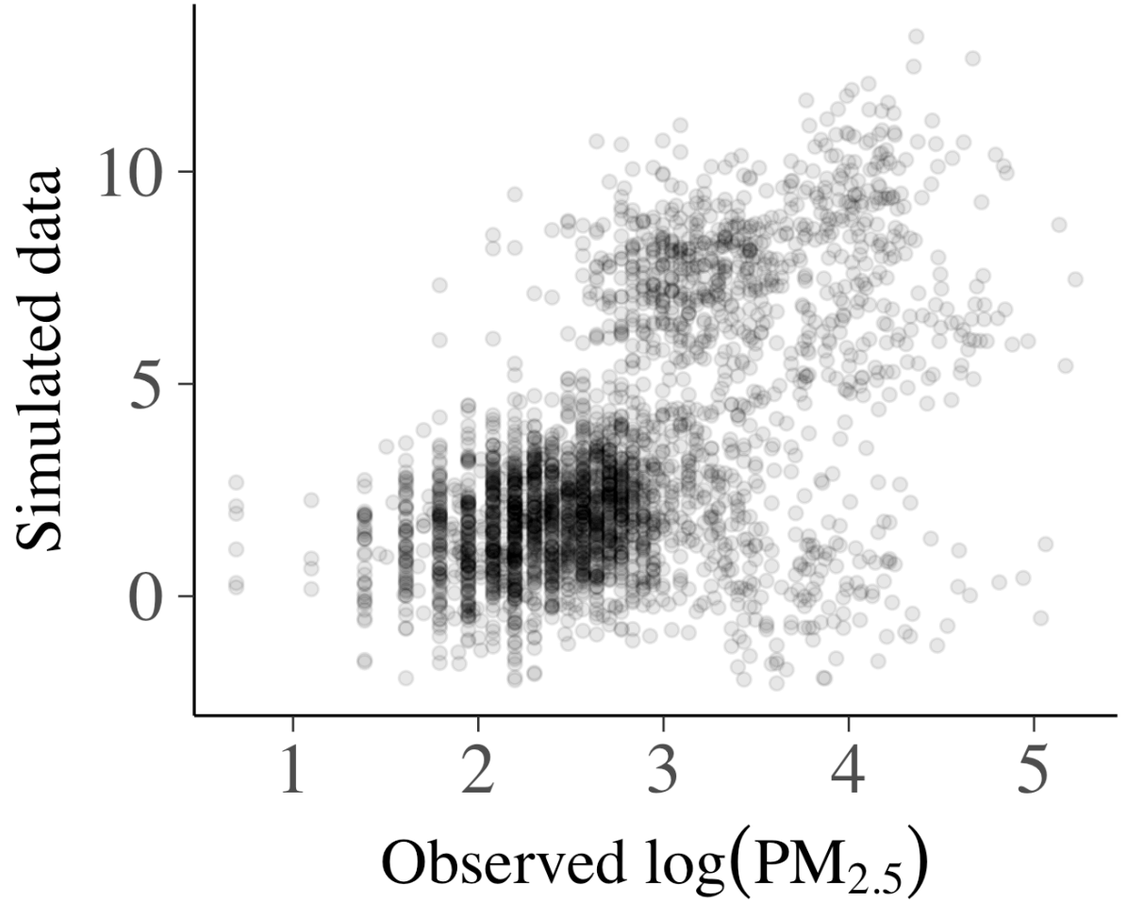


Prior predictive checking

fake data is almost as useful as real data

What are better priors for the global intercept and slope and the hierarchical scale parameters?

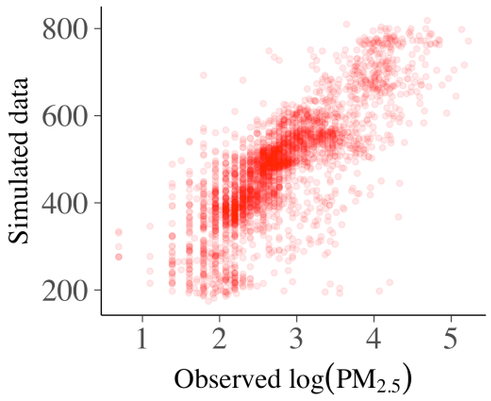
$$\alpha_0 \sim N(0, 1)$$
$$\beta_0 \sim N(1, 1)$$
$$\tau_\alpha \sim N_+(0, 1)$$
$$\tau_\beta \sim N_+(0, 1)$$



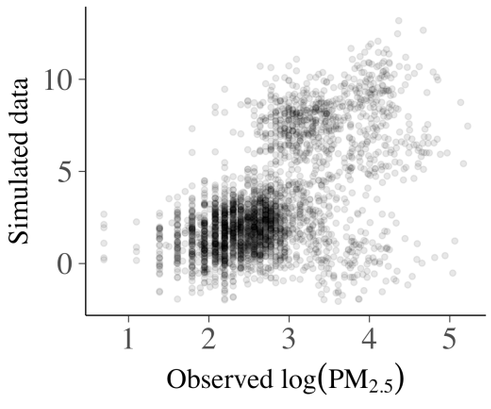
Prior predictive checking

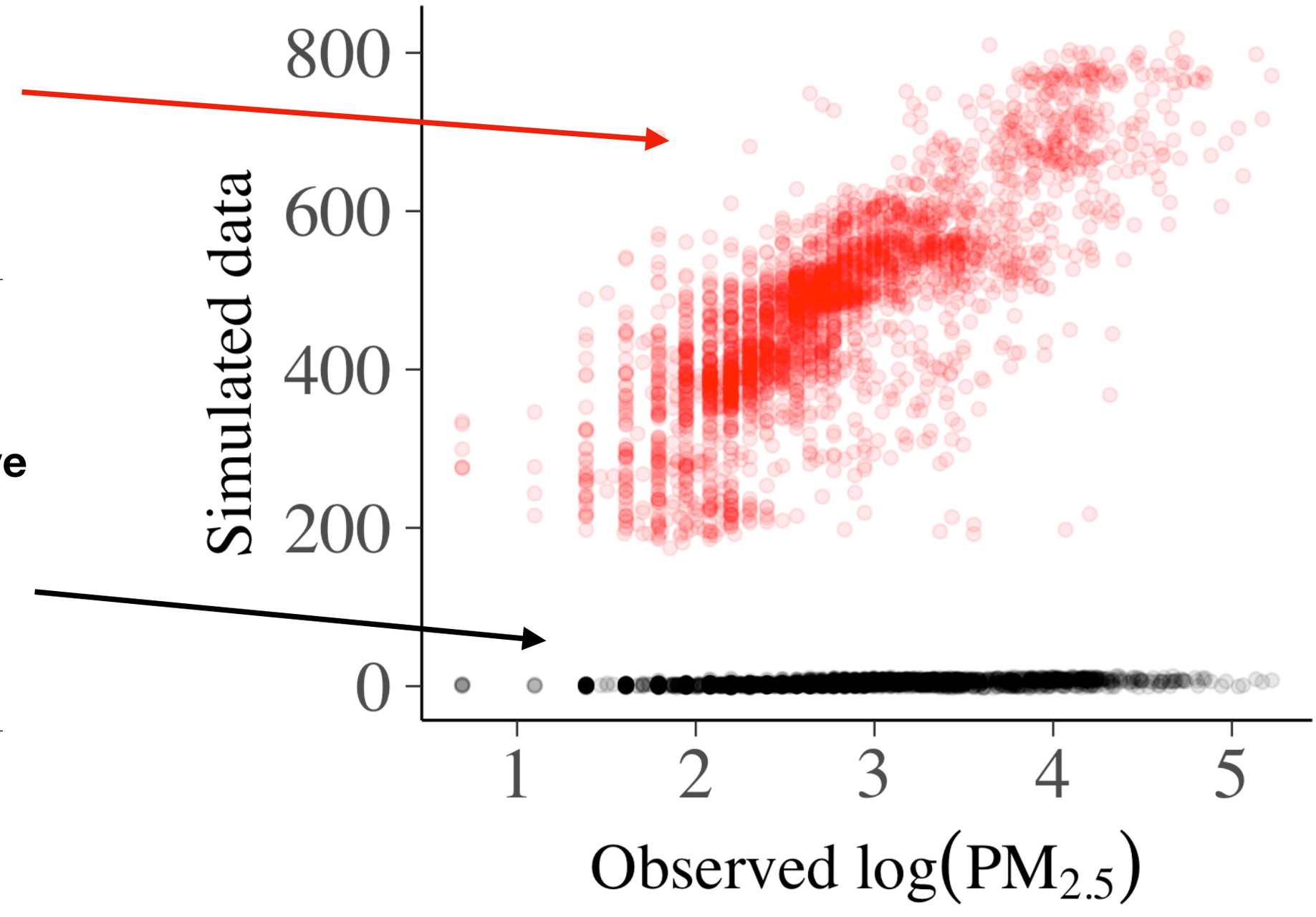
fake data is almost as useful as real data

Non-informative



Weakly informative





What is the problem with “vague” priors?

- If we use an *improper* prior, then we do not specify a joint model for our data and parameters
- More importantly, we do not specify a data generating mechanism ***p(y)***
- By construction, these priors do not regularize inferences, which quite often exaggerates “effect” sizes
- Proper but vague/diffuse is better than improper but is still often problematic. It’s easy to do better!

Prior predictive checking

software packages

bayesplot | mc-stan.org/bayesplot

rstanarm | mc-stan.org/rstanarm

brms | paulbuerkner.com/brms

Model fitting

MCMC and approximate methods

Model fitting

RStan, cmdstanr, rstanarm, brms, …

```
stan_glm(y ~ x, prior = …)

stan_glmer(y ~ x + (1+x|g), prior = …)

stan_gamm4(y ~ s(x), prior = …)
```

and more…

Algorithms

- MCMC (HMC, NUTS)
- Variational (ADVI, Pathfinder)
- Optimization (MLE, MAP)

```
data {
  int<lower=1> N;         // number of observations
  vector[N] log_sat;      // log of satellite measurements
  vector[N] log_pm;       // log of ground PM_2.5 measurements
}
parameters {
  real beta0;             // global intercept
  real beta1;             // global slope
  real<lower=0> sigma;    // error sd for Gaussian likelihood
}
model {
  log_pm ~ normal(beta0 + beta1 * log_sat, sigma);
  beta0 ~ normal(0, 1);
  beta1 ~ normal(1, 1);
  sigma ~ exponential(1);
}
generated quantities {
  vector[N] log_lik;       // pointwise log-likelihood for LOOCV
  vector[N] log_pm_preds; // posterior predictive dist
  for (n in 1:N) {
    real log_pm_hat_n = beta0 + beta1 * log_sat[n];
    log_lik[n] = normal_lpdf(log_pm[n] | log_pm_hat_n, sigma);
    log_pm_preds[n] = normal_rng(log_pm_hat_n, sigma);
  }
}
```

Model fitting

software packages

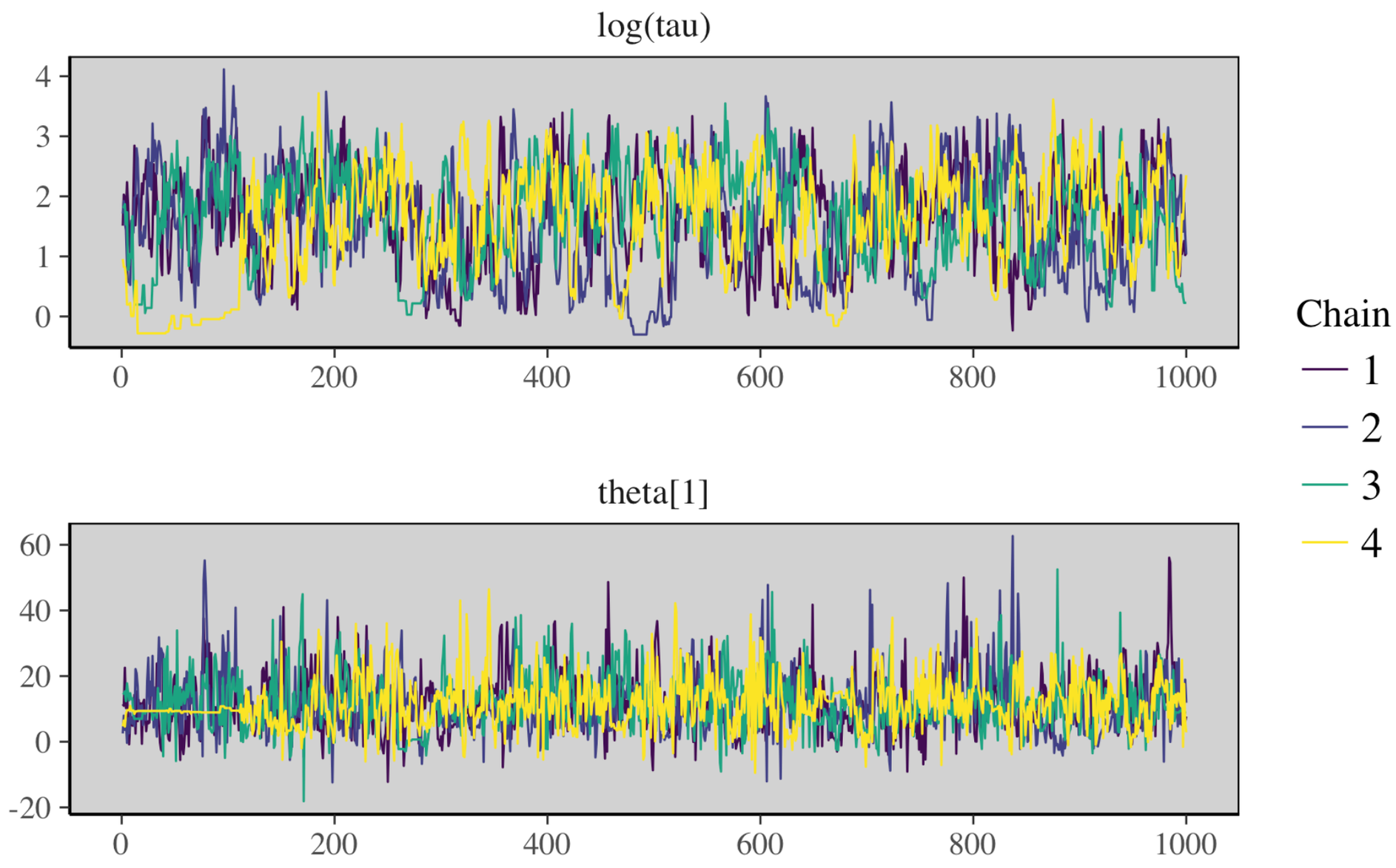
RStan	mc-stan.org/rstan
cmdstanr	mc-stan.org/cmdstanr
rstanarm	mc-stan.org/rstanarm
brms	paulbuerkner.com/brms/

MCMC diagnostics

Beyond trace plots

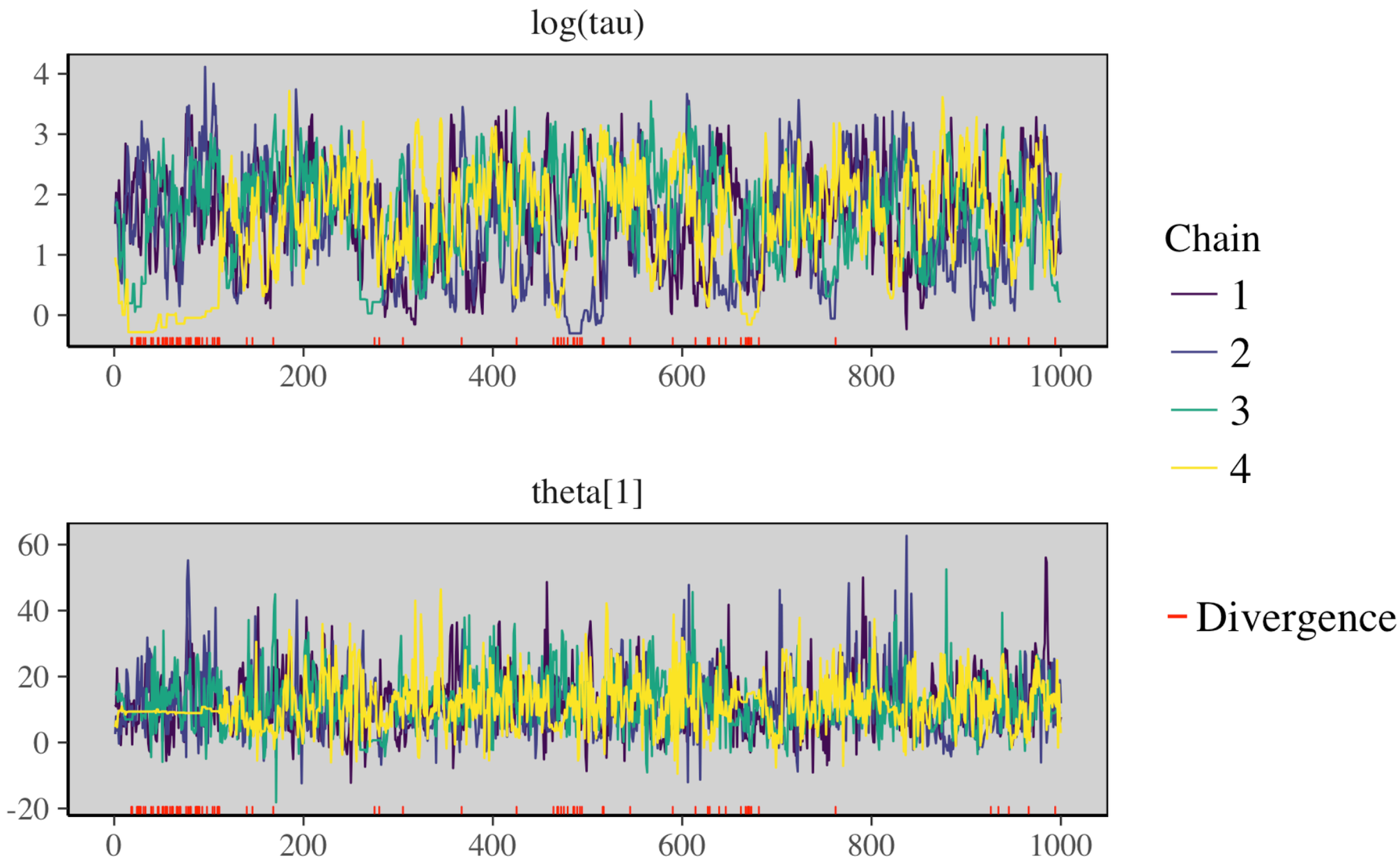
MCMC diagnostics

beyond trace plots



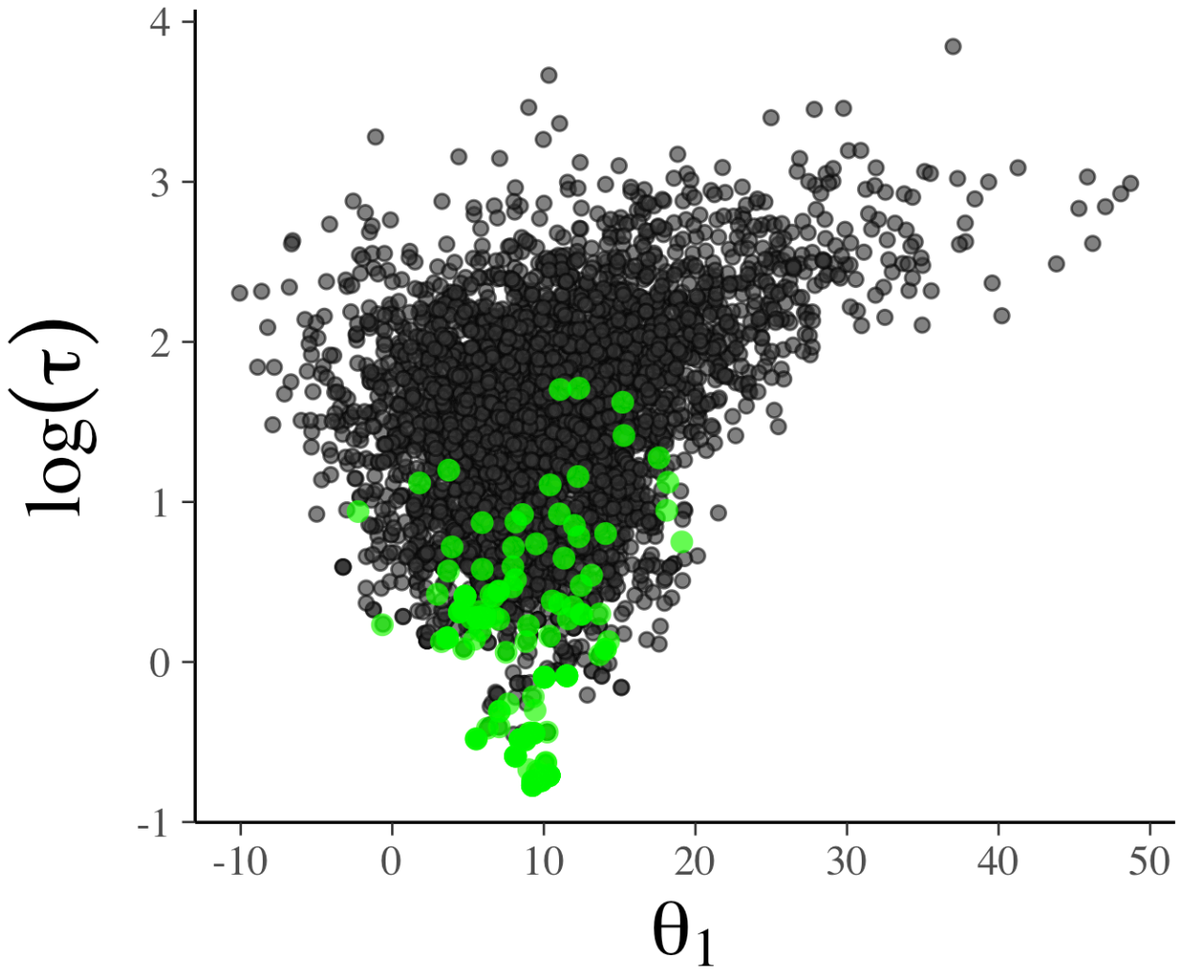
MCMC diagnostics

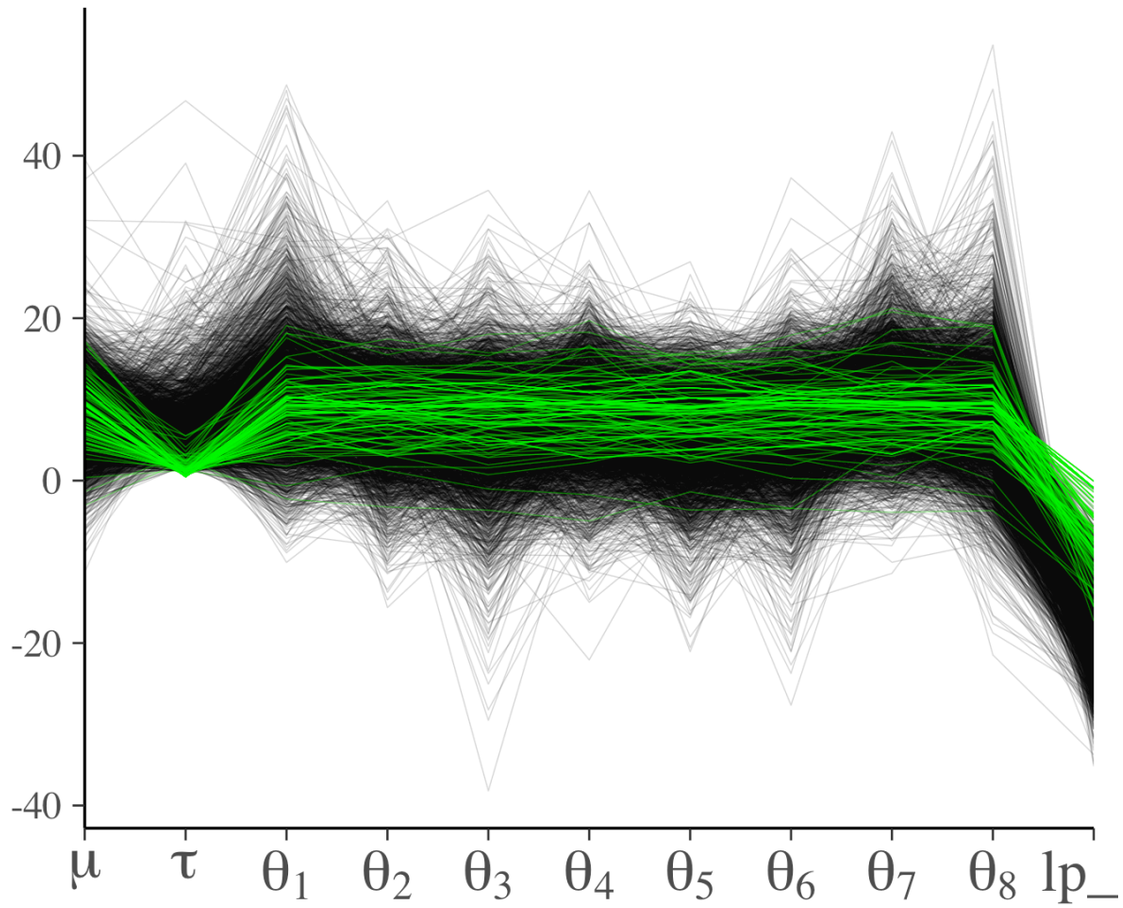
beyond trace plots



MCMC diagnostics

beyond trace plots





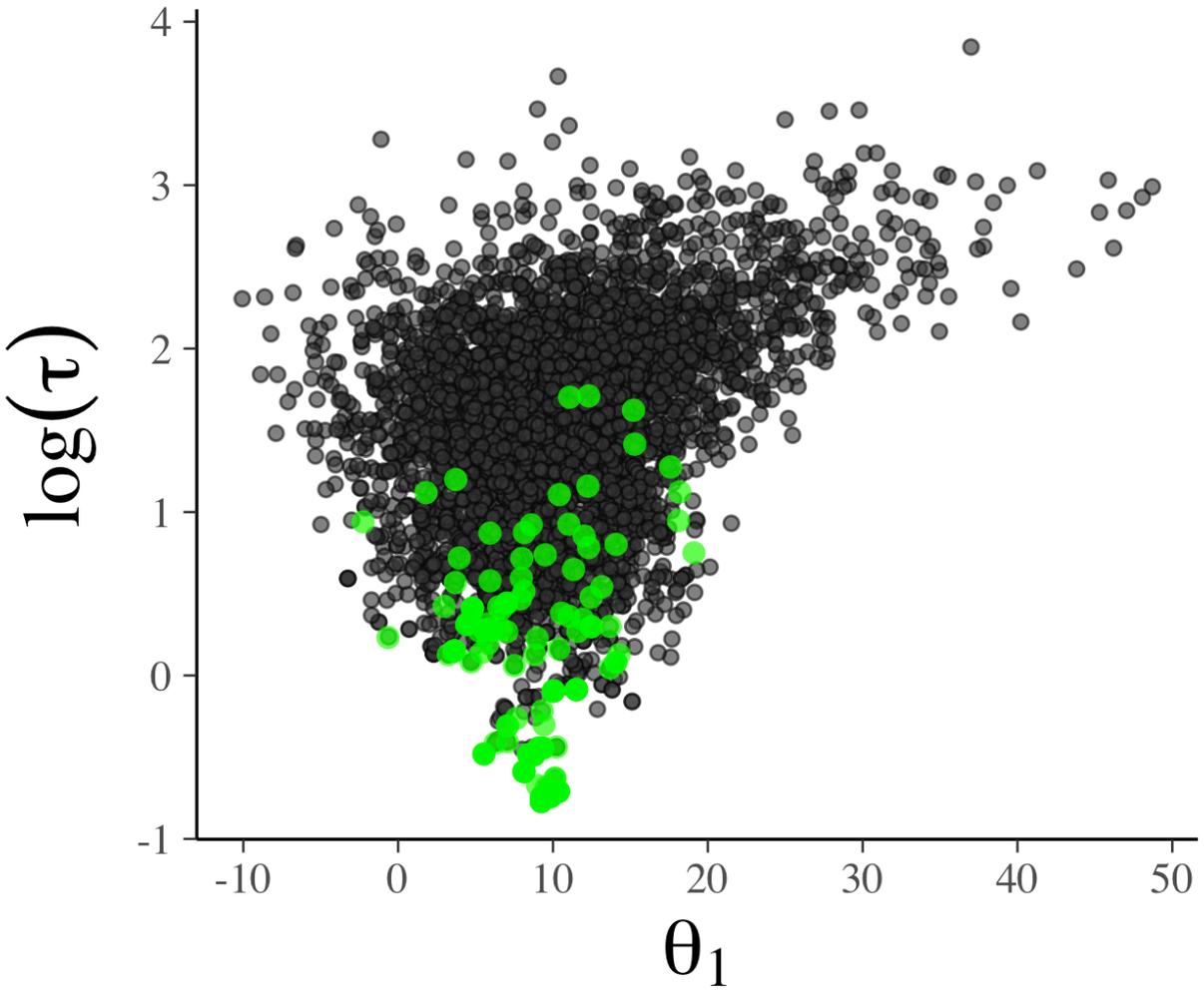
Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019).
Visualization in Bayesian workflow.
J. R. Stat. Soc. A, 182: 389-402
https://doi.org/10.1111/rssa.12378 | github.com/jgabry/bayes-vis-paper

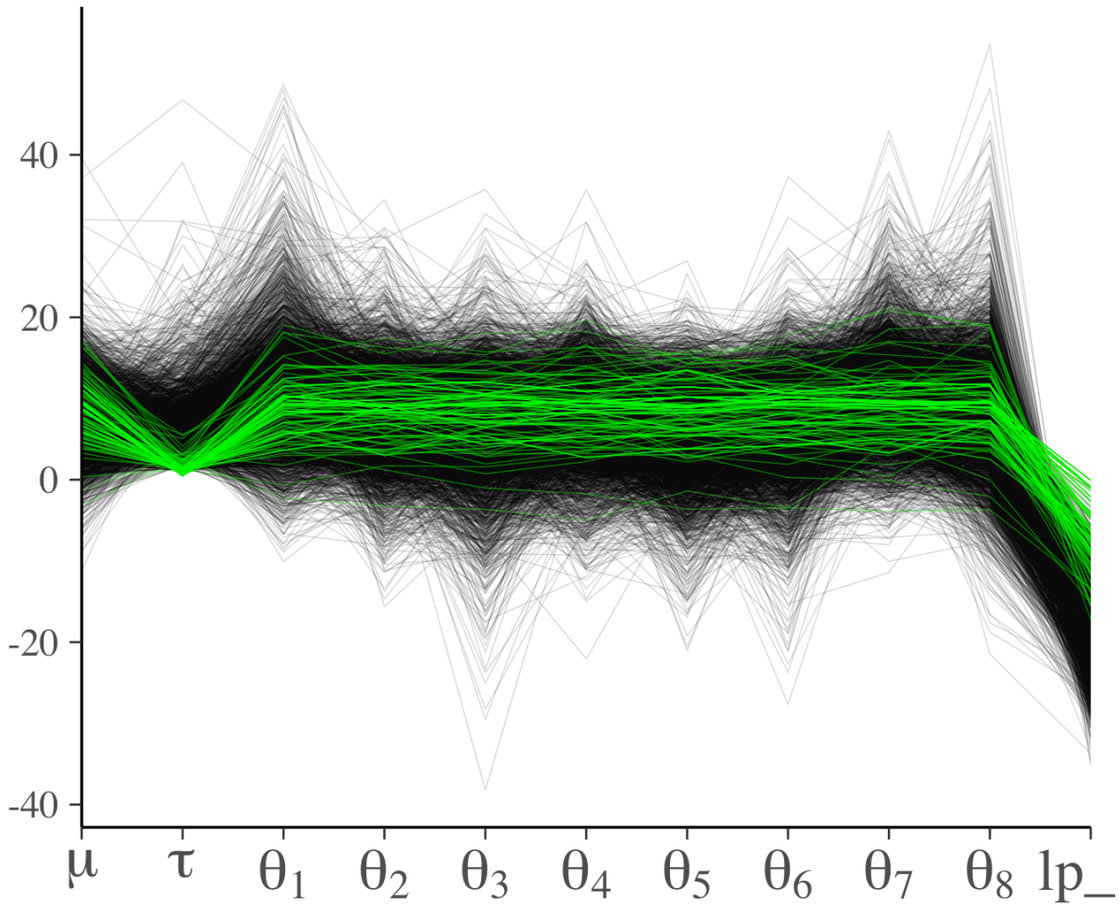
Betancourt, M. (2017).
A conceptual introduction to Hamiltonian Monte Carlo.
arXiv preprint:
arxiv.org/abs/1701.02434

MCMC diagnostics

beyond trace plots

Pathological geometry (divergences cluster)

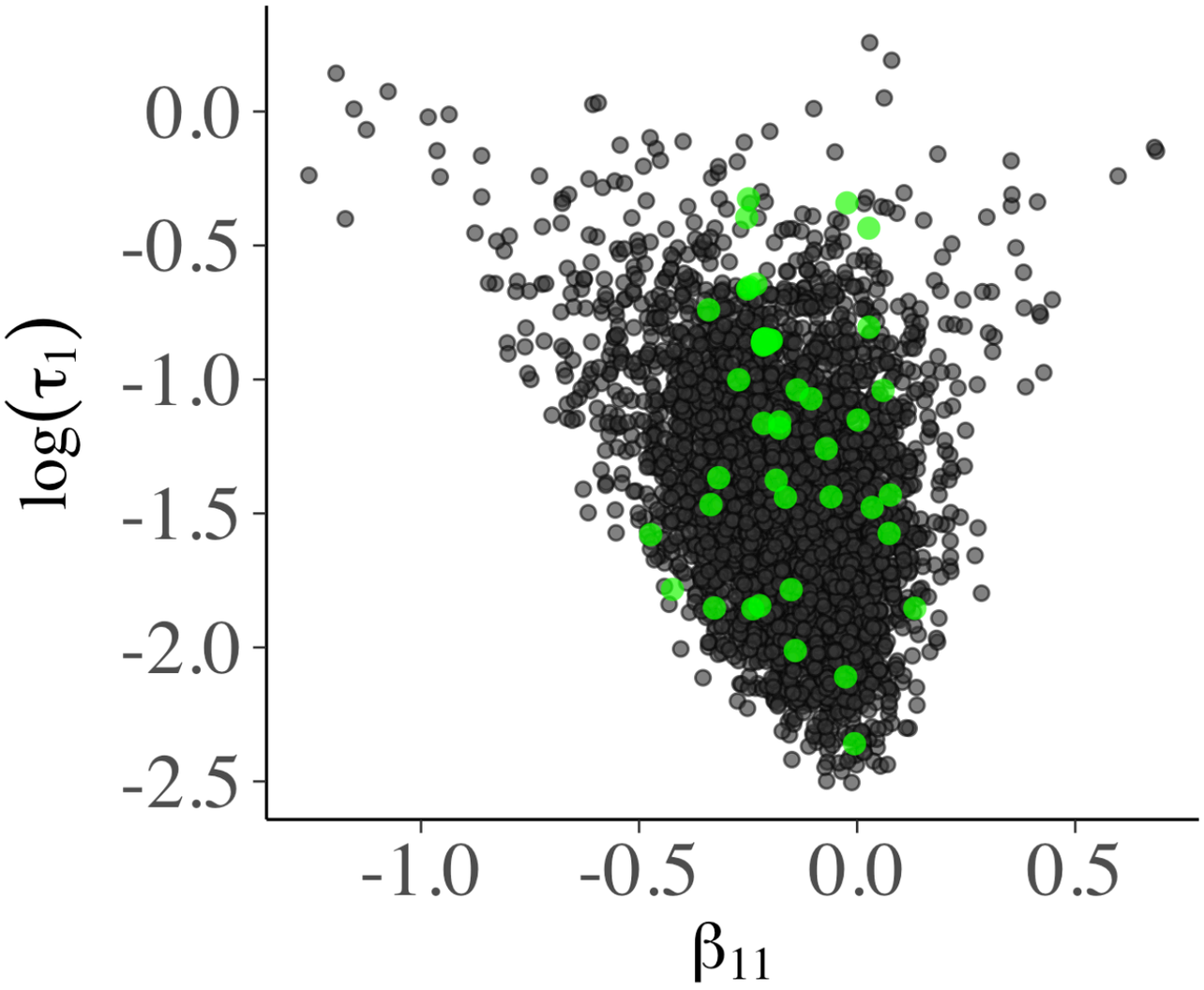


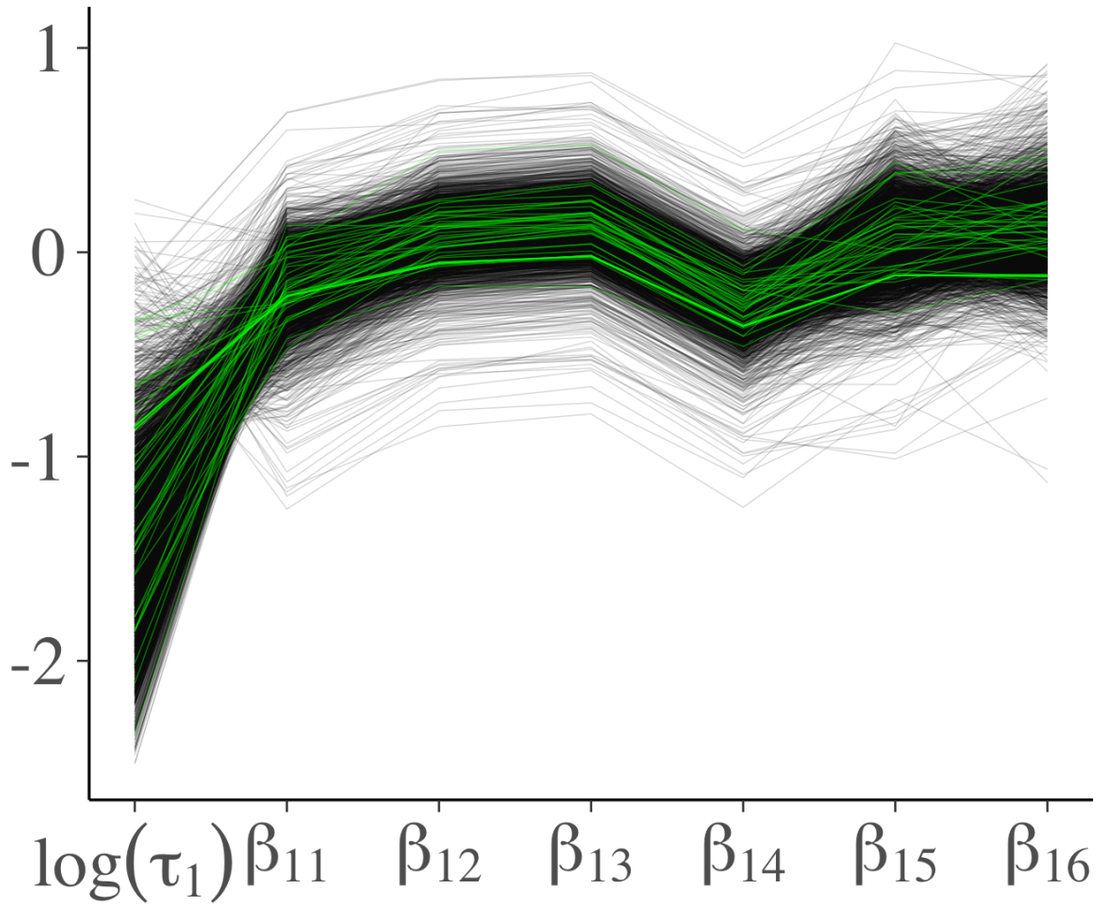


MCMC diagnostics

beyond trace plots

“False positives” (divergences show no pattern)





MCMC diagnostics

numerical checks

```
Warning: 49 of 4000 (1.0%) transitions ended with a divergence.
See https://mc-stan.org/misc/warnings for details.
```

```
$num_divergent
[1]  6 15 18 10

$num_max_treedepth
[1] 0 0 0 0

$ebfmi
[1] 0.4311010 0.4950326 0.5036877 0.4511304
```

	variable	rhat	ess_bulk	ess_tail
	<chr>	<dbl>	<dbl>	<dbl>
1	sigma	1.00	4690.	2643.
2	alpha0	1.00	1562.	2100.
3	beta0	1.00	1272.	1514.
4	alpha[1]	1.00	1836.	2464.
5	alpha[2]	1.00	1567.	2099.
6	alpha[3]	1.00	1597.	2072.
7	alpha[4]	1.00	1569.	2169.
8	alpha[5]	1.00	2576.	2507.
9	alpha[6]	1.00	2394.	2368.
10	beta[1]	1.00	1637.	1858.
11	beta[2]	1.00	1306.	1514.
12	beta[3]	1.00	1273.	1403.
13	beta[4]	1.01	1338.	1703.
14	beta[5]	1.00	2490.	2273.
15	beta[6]	1.00	2368.	2098.
16	tau_alpha	1.00	1680.	2098.
17	tau_beta	1.00	1479.	1472.

MCMC diagnostics

software packages

bayesplot | mc-stan.org/bayesplot

shinystan mc-stan.org/shinystan

posterior mc-stan.org/posterior

Posterior distributions

Visualizing and summarizing (functions of) parameter estimates

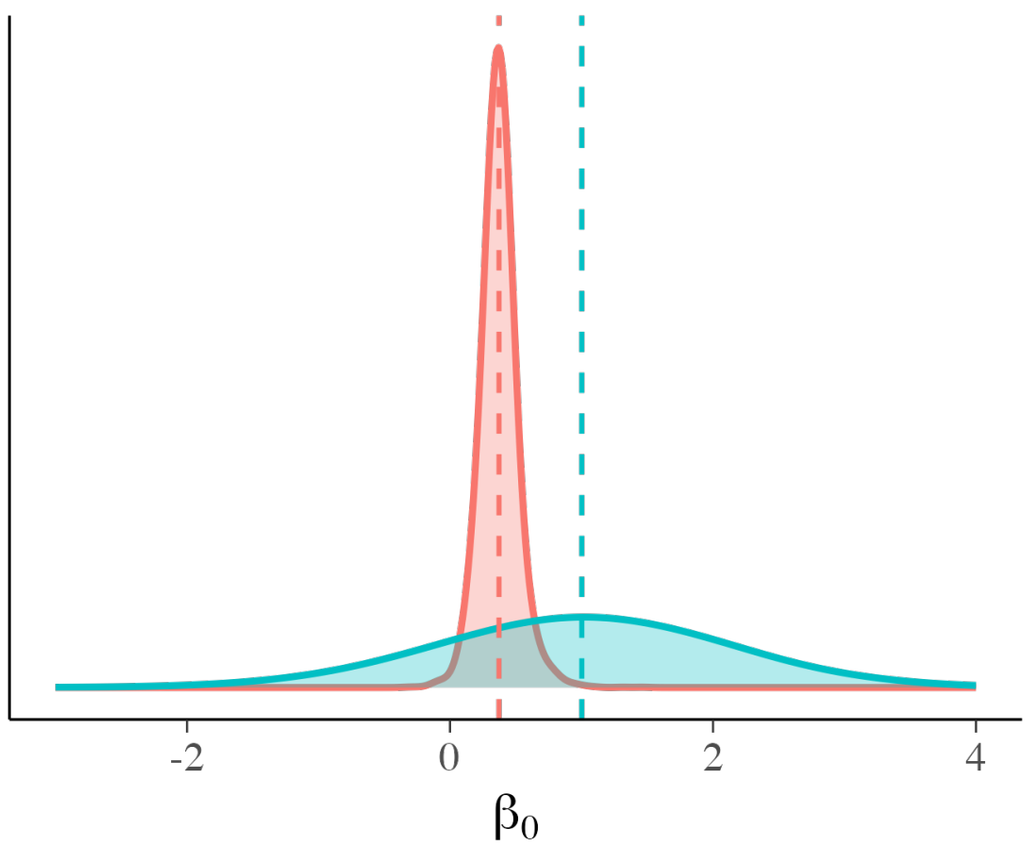
Posterior distributions

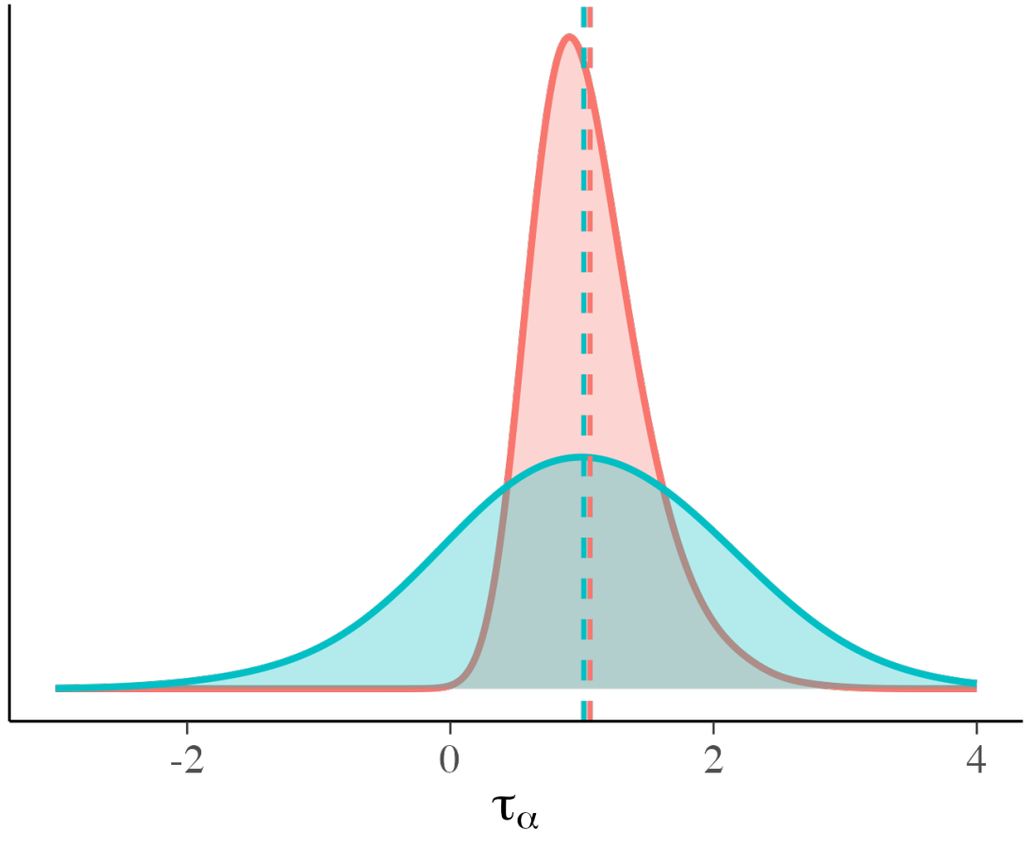
numerical summaries

	variable	mean	median	sd	mad	q5	q95
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	sigma	0.353	0.353	0.00459	0.00457	0.345	0.361
2	alpha0	2.16	2.20	0.478	0.444	1.32	2.87
3	beta0	0.372	0.369	0.136	0.108	0.165	0.594
4	alpha[1]	1.40	1.34	0.677	0.646	0.407	2.60
5	alpha[2]	-0.789	-0.830	0.478	0.441	-1.50	0.0514
6	alpha[3]	-0.177	-0.221	0.483	0.440	-0.896	0.666
7	alpha[4]	0.822	0.786	0.490	0.460	0.0928	1.68
8	alpha[5]	0.621	0.560	0.778	0.722	-0.555	1.98
9	alpha[6]	0.691	0.651	0.746	0.694	-0.439	1.97
10	beta[1]	-0.171	-0.158	0.173	0.153	-0.465	0.0819
11	beta[2]	0.0766	0.0798	0.136	0.108	-0.143	0.285
12	beta[3]	0.129	0.131	0.137	0.112	-0.0890	0.337
13	beta[4]	-0.263	-0.257	0.138	0.115	-0.495	-0.0548
14	beta[5]	0.0547	0.0547	0.193	0.168	-0.258	0.357
15	beta[6]	0.102	0.0988	0.185	0.166	-0.191	0.411
16	tau_alpha	1.06	0.990	0.389	0.354	0.564	1.81
17	tau_beta	0.272	0.236	0.143	0.102	0.123	0.538

Posterior distributions

compare priors and posteriors





- It's interesting to see these distributions change, however: **parameters don't exist** and are (often) hard to interpret!

- The real power of Bayesian inference comes from the posterior *predictive* distribution of **observable quantities**, (where our model makes contact with the real world)

Posterior distributions

software packages

bayesplot | mc-stan.org/bayesplot

shinystan | mc-stan.org/shinystan

posterior | mc-stan.org/posterior

Posterior predictive checks

Visual (and numerical) model evaluation

Posterior predictive checking

assessing model misfitting and overfitting

The *posterior predictive distribution* is the average data generation process over the posterior uncertainty

$$p(\tilde{y}|y) = \int p(\tilde{y}|\theta)\, p(\theta|y)\, d\theta$$

Posterior predictive checking

same process as before but using posterior not prior

- Used to understand the fit of the model
- Misfitting and overfitting both manifest as **tension between measurements and predictive distributions**
- Graphical posterior predictive checks (PPCs) visually compare the observed data (or functions of the data) to simulations from the model

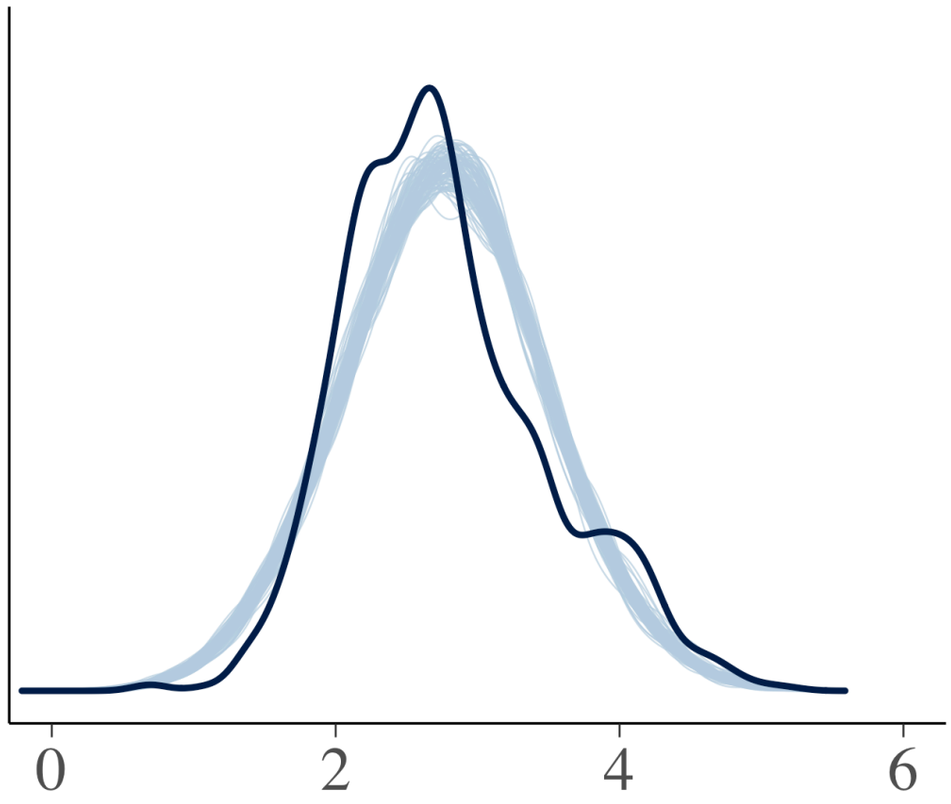
1. Sample from your posterior
2. Use observed values of x
3. Sample from the **posterior predictive distribution** (generate data) given the x values and that particular random draw from the posterior
4. Repeat previous steps many times
5. Compute summaries / visualize

Posterior predictive checking

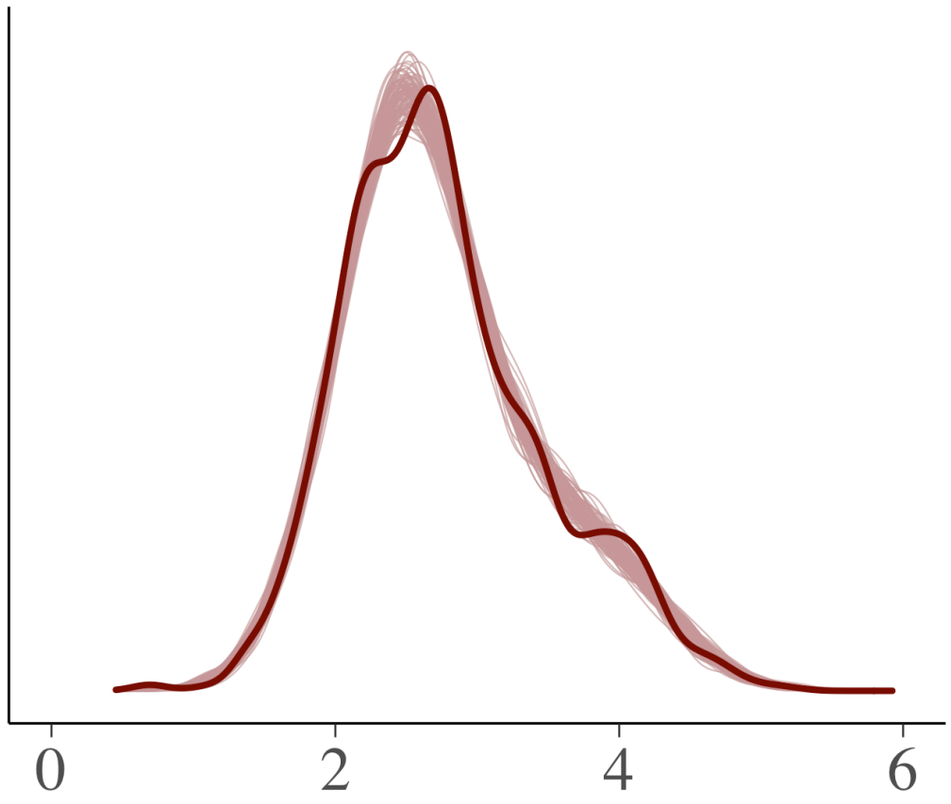
visual model evaluation

Observed data vs posterior predictive simulations

Model 1 (single level)



Model 3 (multilevel)

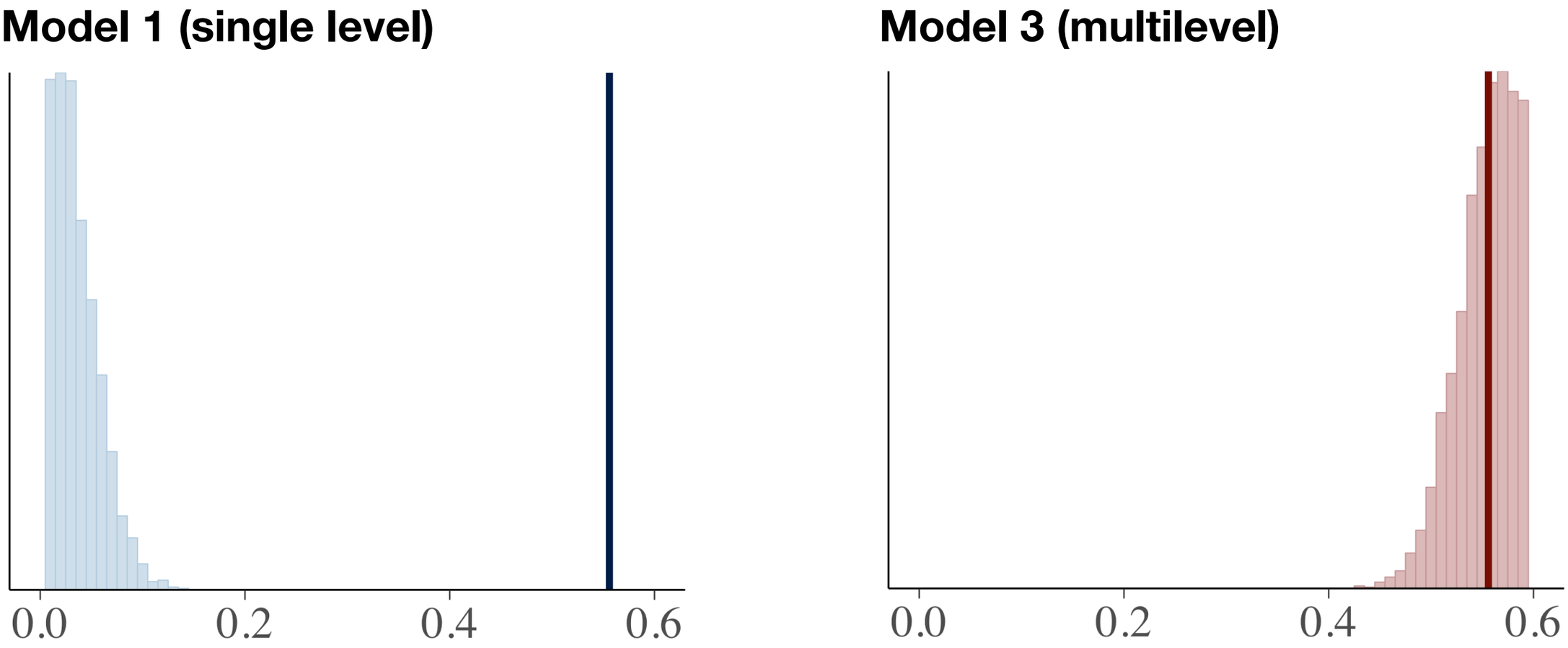


Posterior predictive checking

visual model evaluation

Observed vs posterior predictive statistics

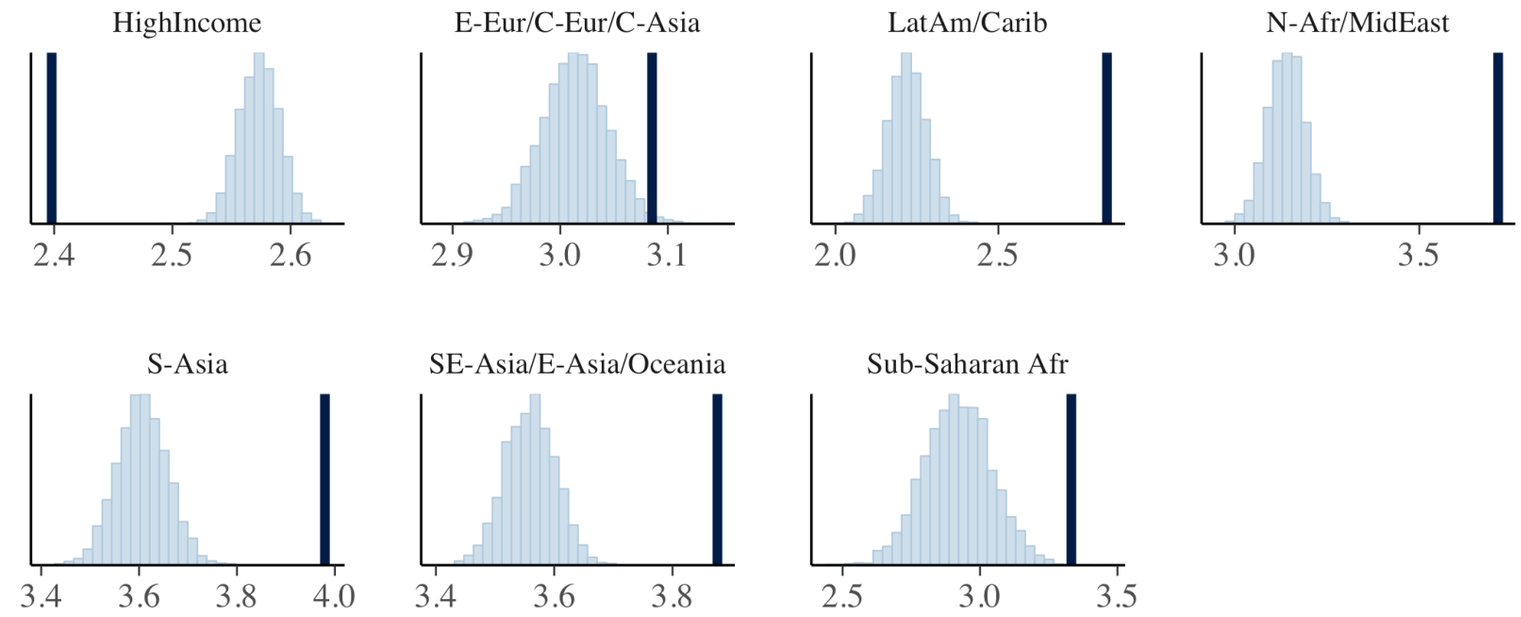
Best to use statistics orthogonal to model parameters otherwise calibration may be an issue



Posterior predictive checking

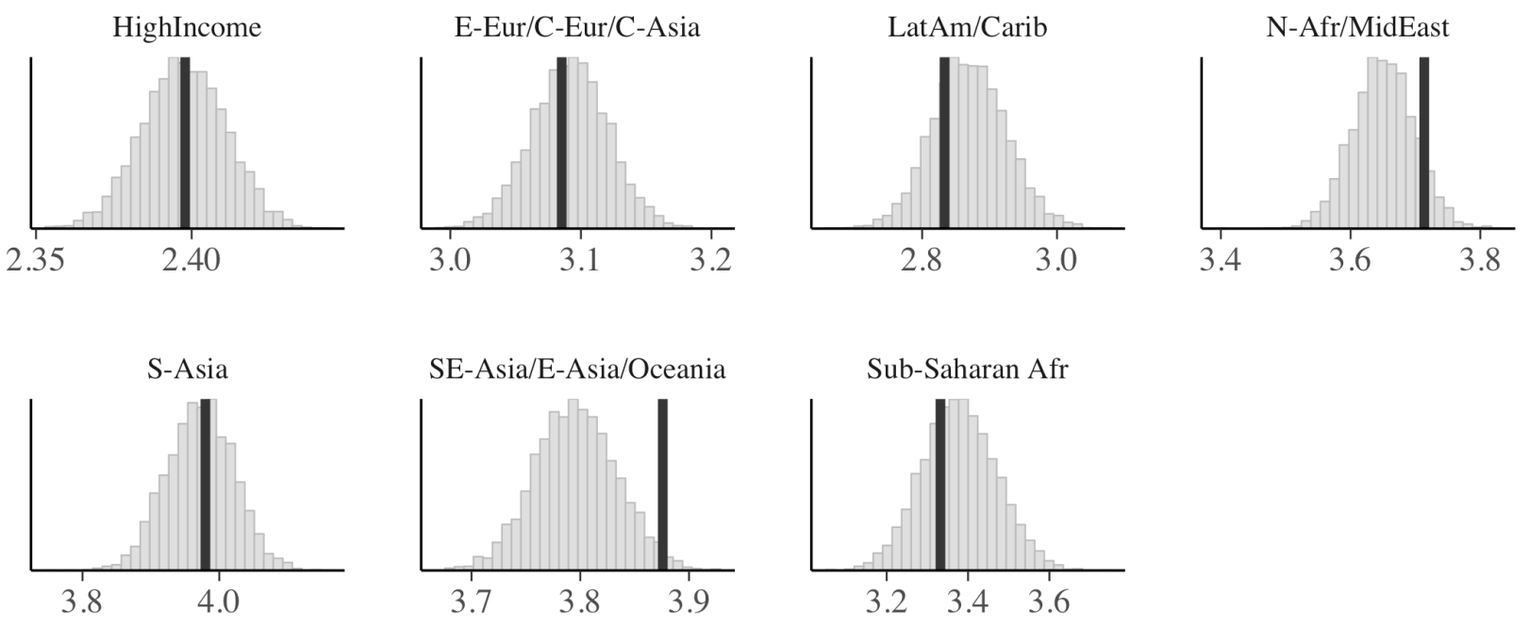
visual model evaluation

Model 1 (single level)



$$T(y) = \text{med}(y|\text{region})$$

Model 2 (multilevel)



Posterior predictive checking

software packages

`bayesplot`	mc-stan.org/bayesplot
`loo`	mc-stan.org/loo
`rstanarm`	mc-stan.org/rstanarm
`brms`	paulbuerkner.com/brms

Model comparison

Pointwise predictive comparisons & LOO-CV

Model comparison

more than just computing a single number

- Many people think of model comparison as simply computing some statistic (e.g., [ABD]IC, WAIC, LOOIC/ELPD), but point-wise comparisons are also useful

- Visual PPCs can also identify unusual/influential (outliers, high leverage) data points

- We like using cross-validated leave-one-out predictive distributions

$$p(y_i|y_{-i})$$

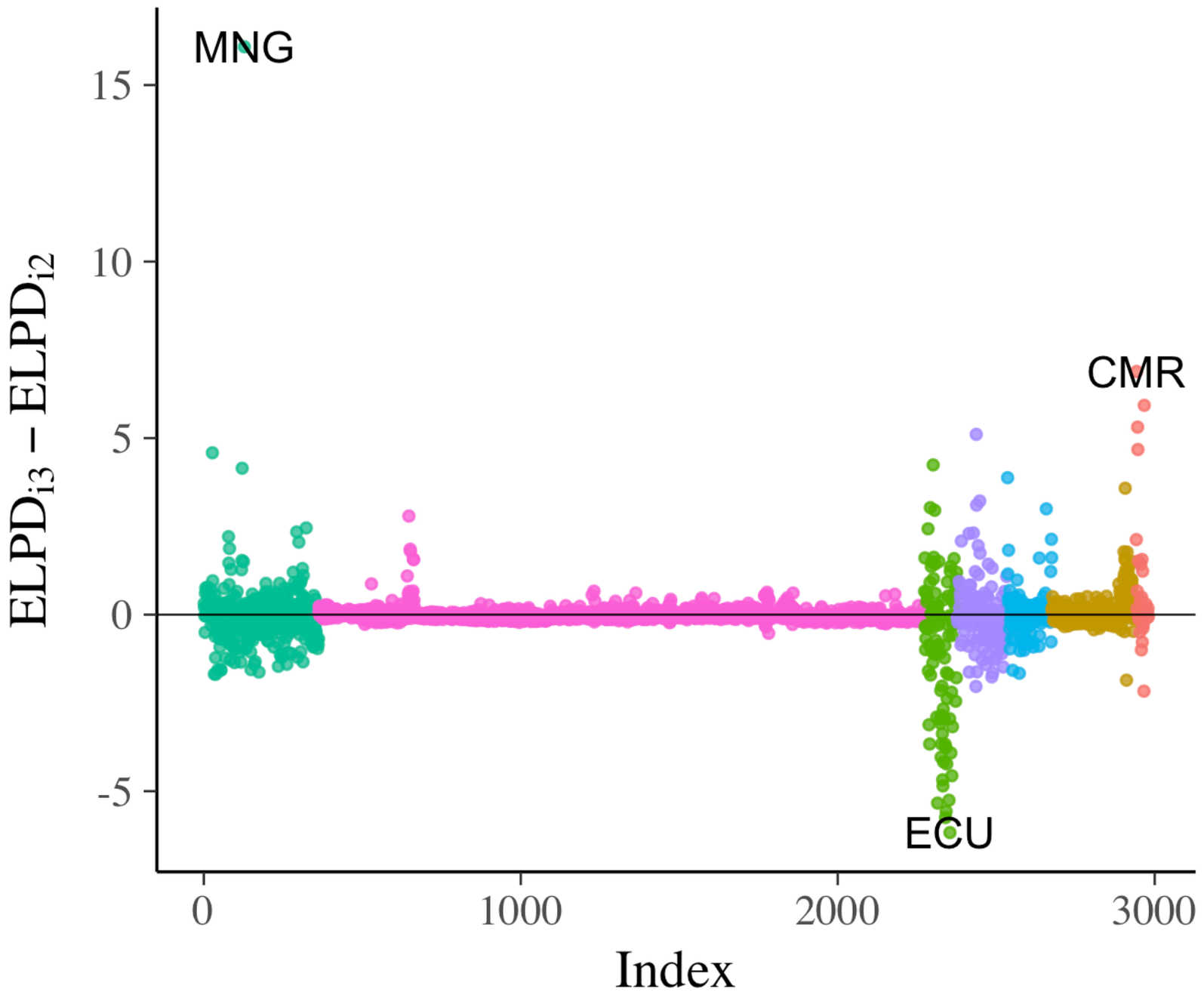
- Which model best predicts the data that are left out?

Vehtari, A., Gelman, A., and Gabry, J. (2017).
Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC.
Statistics and Computing. 27(5), 1413–1432.
https://doi.org/10.1007/s11222-016-9696-4

Bürkner, P., Gabry, J., and Vehtari,A. (2019).
Approximate leave-future-out cross-validation for Bayesian time series models
Journal of Statistical Computation and Simulation, 90(14), 2499–2523
https://doi.org/10.1080/00949655.2020.1783262

Model comparison

pointwise predictive comparisons



Model comparison

Efficient approximate LOO-CV

- How do we compute LOO-CV without fitting the model *N* times?
- Fit once, then use Pareto smoothed importance sampling (PSIS-LOO)
- Has finite variance property of truncated IS
- And less bias (replace largest weights with order stats of generalized Pareto)
- Assumes posterior not *highly* sensitive to leaving out single observations
- Asymptotically equivalent to WAIC
- Advantage: PSIS-LOO CV more robust + has diagnostics (check assumptions)

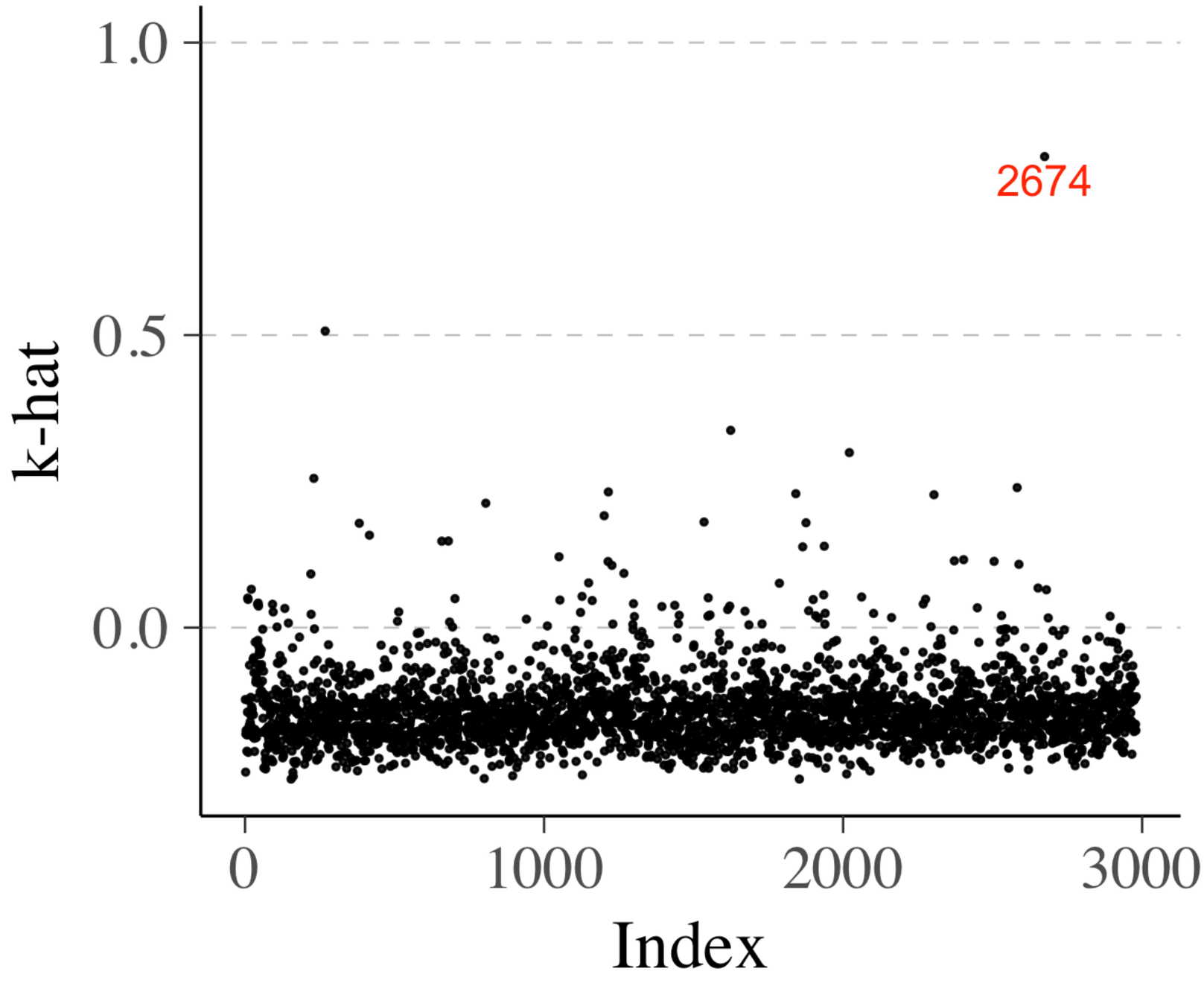
Vehtari, A., Gelman, A., and Gabry, J. (2017).
Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC.
Statistics and Computing. 27(5), 1413–1432.
doi: 10.1007/s11222-016-9696-4

Vehtari, A., Simpson, D., Gelman, A., Yau, Y., and Gabry, J. (2024).
Pareto Smoothed Importance Sampling.
Journal of Machine Learning Research. 25(72):1–58, 2024
http://jmlr.org/papers/v25/19-556.html

Model comparison

Diagnostics

Shape parameter of generalized Pareto distribution & influential observations



Model comparison

software packages

`loo` | mc-stan.org/loo

`projpred` | mc-stan.org/projpred

`bayesplot` | mc-stan.org/bayesplot

Other Software

in R, Python, Julia, and other languages

Other packages

in R, Python, Julia, and other languages

- I focused on R packages written by myself and my collaborators
- But there are many more great packages available in R, Python, Julia and other languages
- And you can make your own! (e.g., using `rstantools` or `instantiate`)

`rstantools` | mc-stan.org/rstantools

mc-stan.org/tools

Thank you!

Questions?