

STA2201H Methods of Applied Statistics II

Monica Alexander

Week 7: Model checking

Woooooahhhh we're halfway there

Outlook

- ▶ Model checking, comparison
- ▶ Multilevel models
- ▶ Splines etc
- ▶ Sampling/ measurement issues

Assignments

- ▶ A3: out by end of week
- ▶ A4: expect to be writing a mini paper/report

Offsets

$$\text{Deaths} \sim \text{Poisson}(\mu)$$

$$\mu = \lambda \times \text{Population}$$

$$\log(\mu) = \log(\lambda) + \log(\text{Population})$$

$$\log(\lambda) = X\beta$$

λ is a mortality rate, and what we're interested in trying to model

Offsets

$$\text{Deaths} \sim \text{Poisson}(\mu)$$

$$\mu = \lambda \times \text{Expected deaths}$$

$$\log(\mu) = \log(\lambda) + \log(\text{Expected deaths})$$

$$\log(\lambda) = X\beta$$

λ is a mortality ratio (of observed to expected), and what we're interested in trying to model

Note λ is called the standardized mortality ratio.

Overview

- ▶ Prior predictive checks
- ▶ Posterior predictive checks
- ▶ LOO-PSIS
- ▶ (brief) Information criteria
- ▶ (brief) Actually training and testing

Reading

- ▶ BDA chapter 6
- ▶ This lecture inspired by Gabry et al, 2019, “Visualization in the Bayesian Workflow”
- ▶ Extra reading on PSIS: Vehtari, A., Gelman, A., and Gabry, J. 2017. “Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC”.
- ▶ Betancourt, “Towards A Principled Bayesian Workflow”

Phases of the (Bayesian) statistical workflow

1. EDA
2. Model checks based on simulated data and the prior predictive distribution
3. Run the model, computational checks (trace plots etc)
4. Posterior predictive checks, cross-validation checks, other checks of data v predictions

Steps of building a Bayesian model

Model choice: need to decide on

- ▶ Likelihood
- ▶ Functional form of model for $E(Y)$
- ▶ Priors for parameters

Prior predictive distributions

- ▶ If we specify proper priors for all parameters in the model, our model is **generative**
- ▶ Yields a joint prior distribution on the parameters and data, and hence a prior marginal distribution for the data

Prior predictive distribution for new \tilde{y}

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

In practice (in R) we can simulate values of θ from the prior distribution(s), and then simulate from the likelihood to generate values of \tilde{y} , and then look at the resulting distribution.

Prior predictive distributions

Why would we do this? We should be setting our priors so they make sense anyway.

Prior predictive distributions

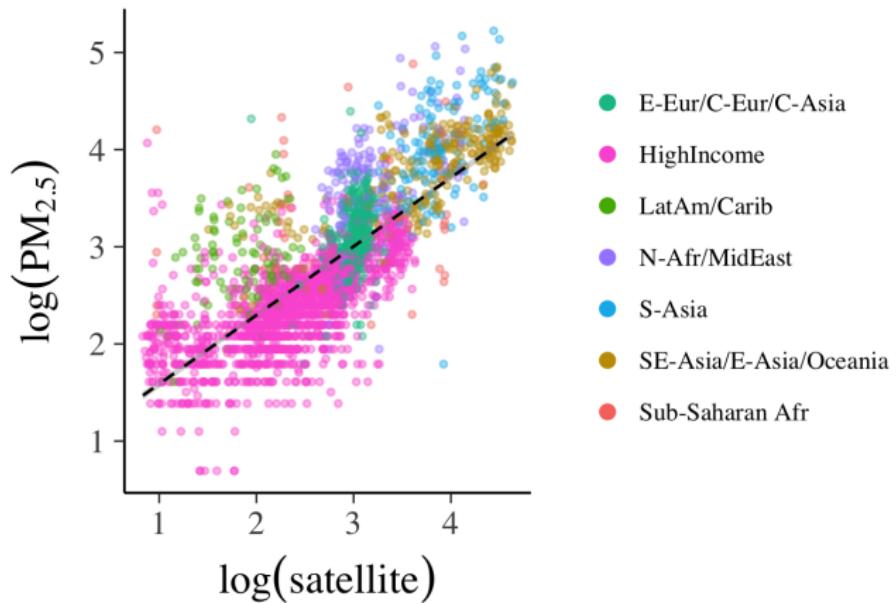
- ▶ Allows you to assess the interplay between the prior and the likelihood
- ▶ Assess whether the priors make sense for a particular problem

We want our priors to be set such that the prior data-generating distribution $p(y)$ could represent any data set that could plausibly be observed.

A *weakly informative joint prior data generating process* has at least some mass around extreme but plausible data sets, but no mass on completely implausible data sets.

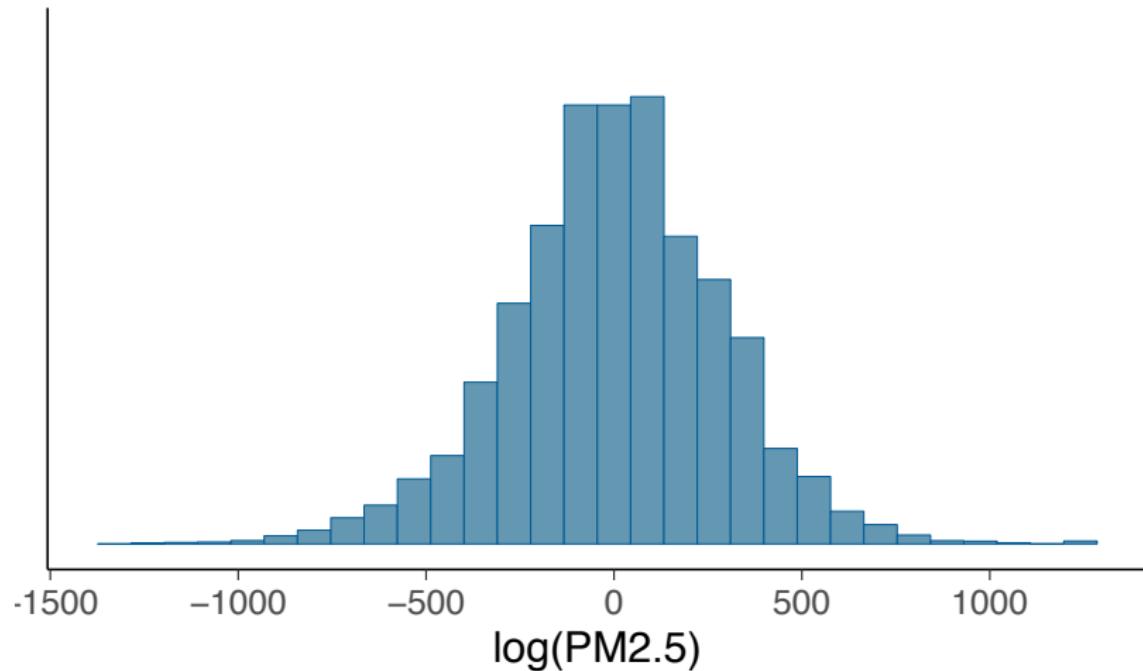
Example: air pollution

- ▶ Example in Gabry et al paper
- ▶ Goal: estimate exposure to $PM_{2.5}$
- ▶ Model relationship between ground measures and satellite data



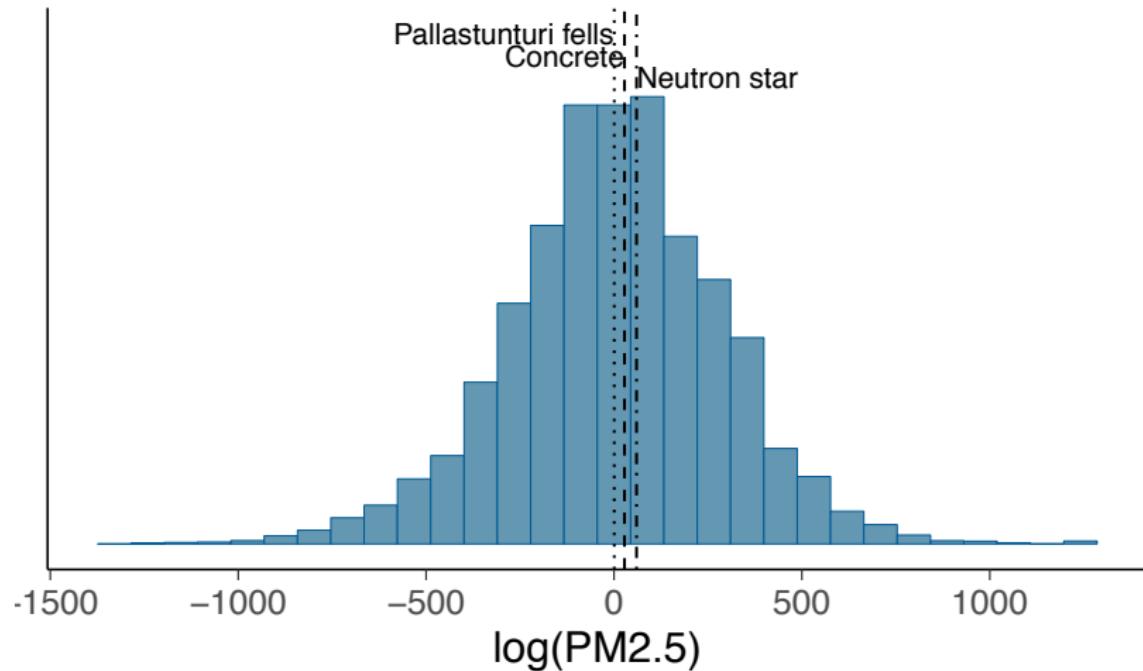
Example: air pollution

Prior predictive distribution with vague prior



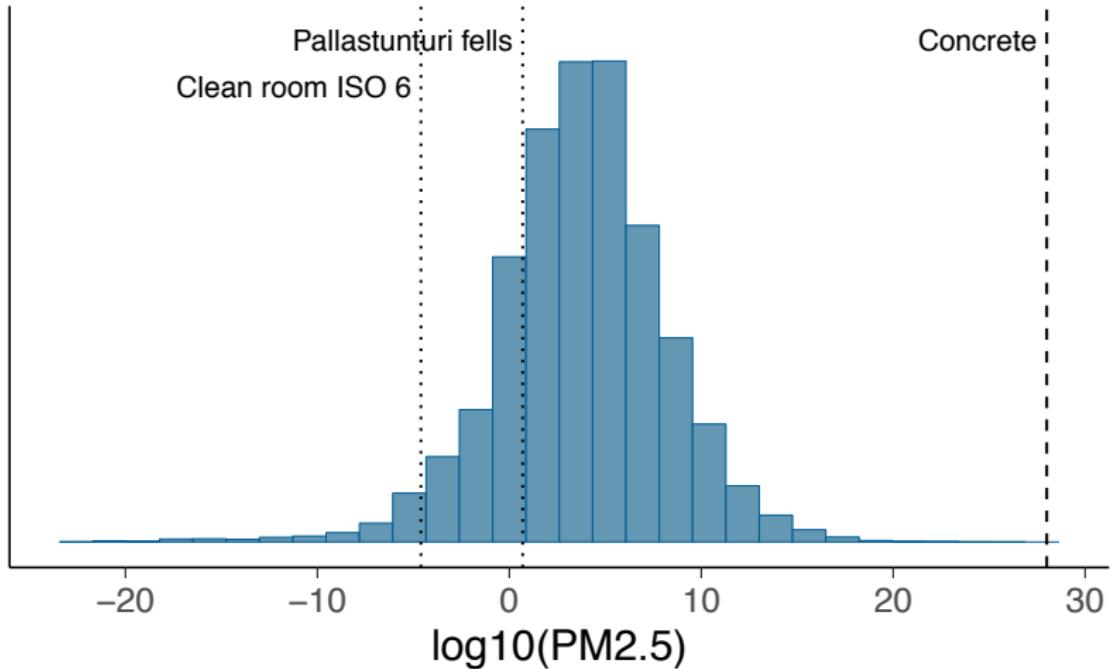
Example: air pollution

Prior predictive distribution with vague prior



Example: air pollution

Prior predictive distribution with weakly informative

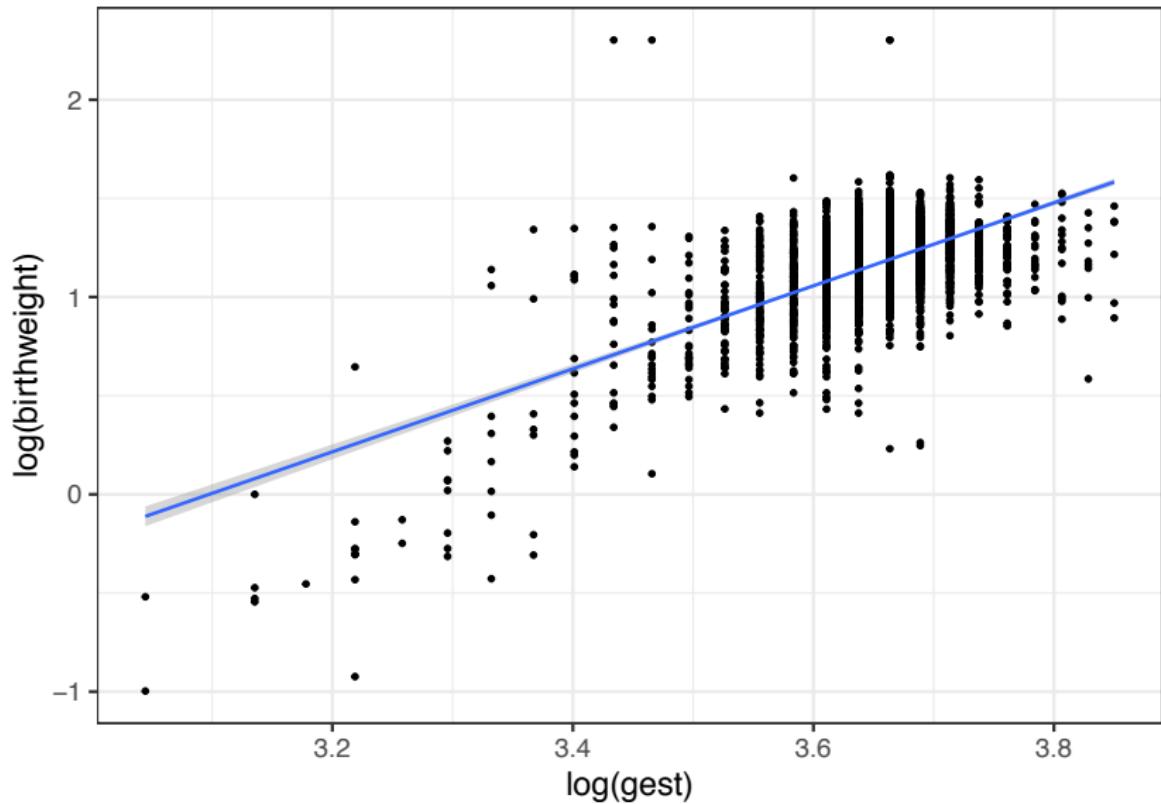


Example: births

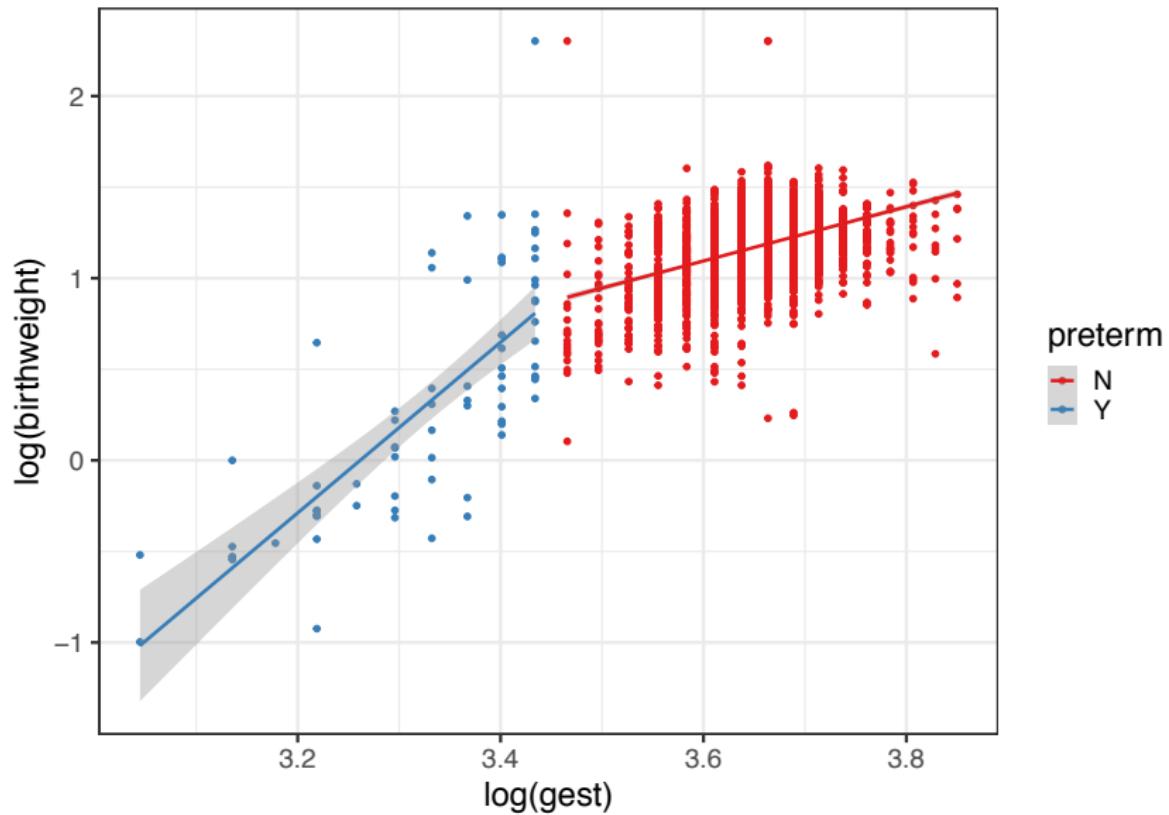
Example: births data

- ▶ Similar set-up to the infant deaths dataset you saw in Assignment 2
- ▶ Looking at all births in 2017
- ▶ Interested in investigating patterns in birth weight
- ▶ Just using .1% sample for now

Weight v gestational age



Weight v gestational age



Priors

Consider a simple model of weight v gestational age (logged)

$$\log(y_i) \sim (\beta_0 + \beta_1 \log(x_i), \sigma^2)$$

Vague priors:

- ▶ β 's $\sim N(0, 100)$
- ▶ $\sigma^2 \sim \text{InvGamma}(1, 100)$

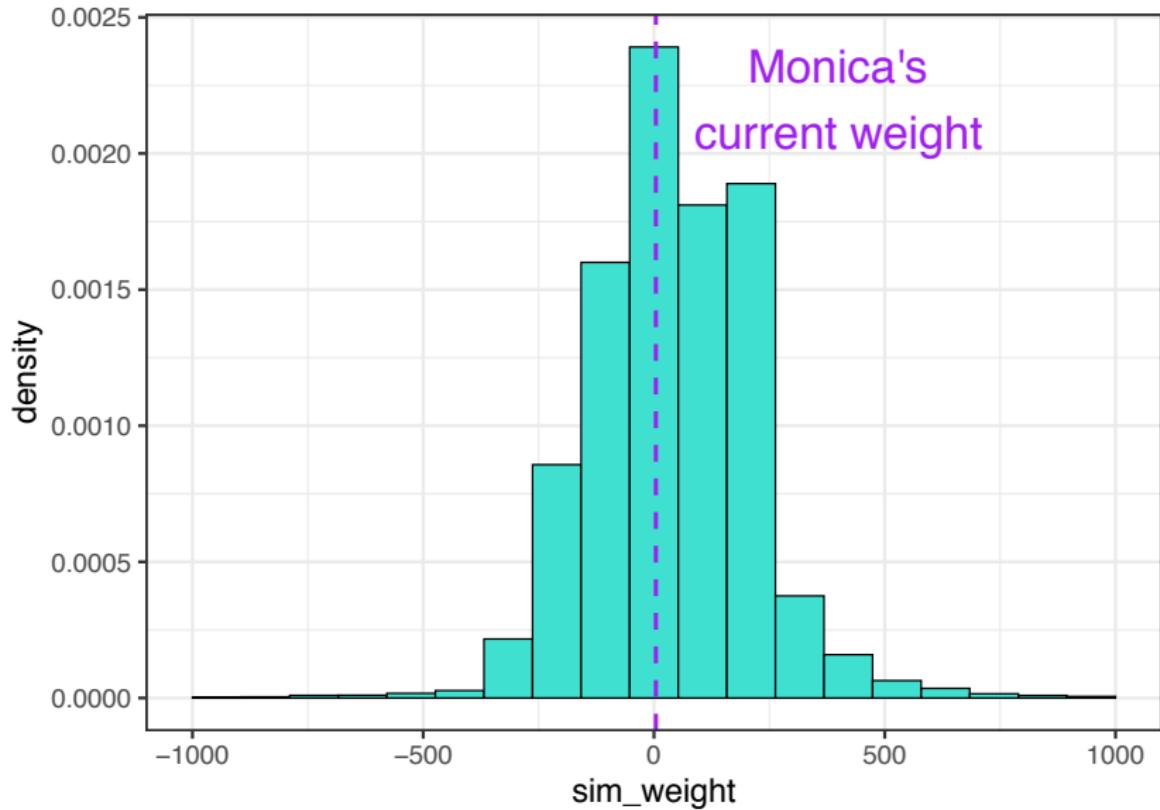
Weakly informative priors:

- ▶ β 's $\sim N(0, 1)$
- ▶ $\sigma \sim \text{Half Normal}(0, 1)$

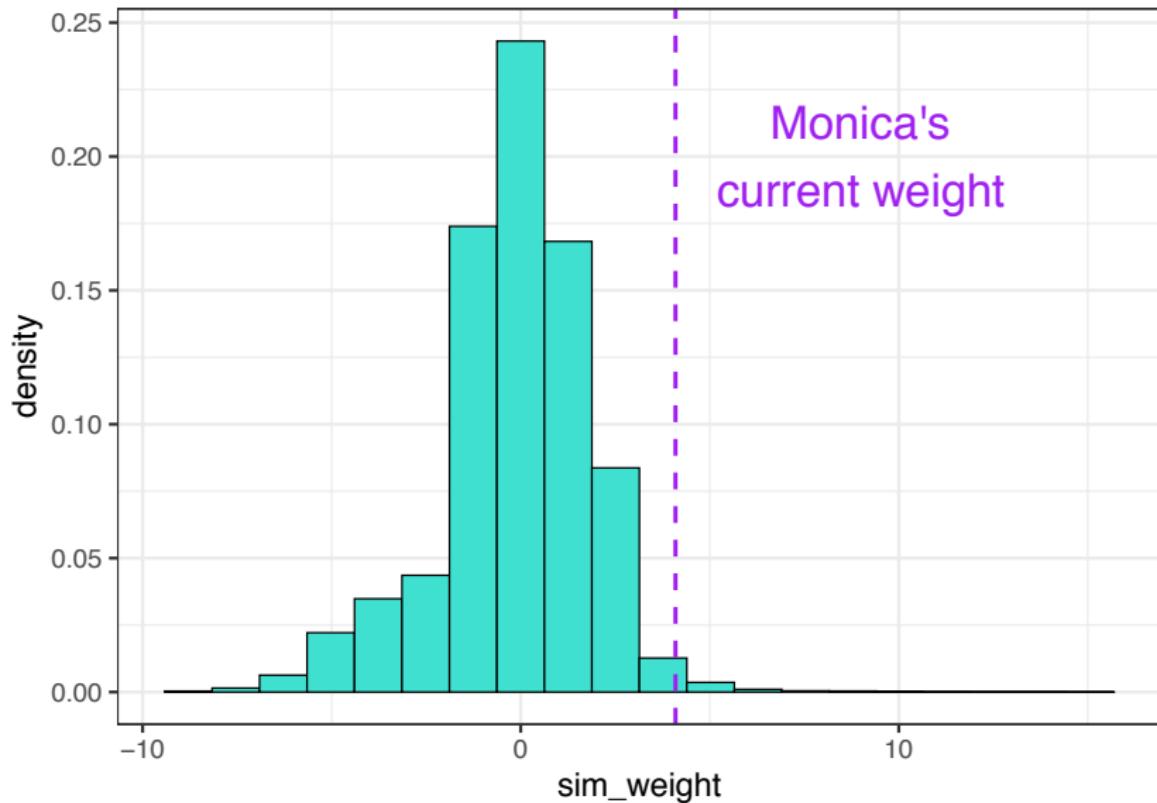
Check these in each case by:

- ▶ Simulating parameters from priors
- ▶ Simulating data y_i from likelihood
- ▶ Plot histogram of implied weights

Vague priors



Weakly informative priors



Posterior predictive checks

Posterior predictive checks

Now we've run a candidate model, how do we explore and check how the model is doing? The idea of **posterior predictive checks** is to compare our observed data to replicated data from the model. If our model is a good fit, we should be able to use it to generate a dataset that resembles the observed data.

Posterior predictive checks

Posterior predictive distribution for new \tilde{y}

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

To obtain samples from this distribution, we need to

- ▶ Get posterior samples of our parameters $\theta^{(s)}$ (MCMC output!)
- ▶ For each posterior sample, we obtain one replicated dataset $\tilde{y}^{(s)}$ by sampling from the likelihood $p(\tilde{y}|\theta^{(s)})$. Can do this in R or within Stan.

Generated quantities in Stan

For each posterior sample, we obtain one replicated dataset $\tilde{y}^{(s)}$ by sampling from the likelihood $p(\tilde{y}|\theta^{(s)})$. Can do this in R or within Stan.

```
model {
    // Log-likelihood
    target += normal_lpdf(log_weight | beta0 + beta1 * log_gest, sigma);

    // Log-priors
    target += normal_lpdf(sigma | 0, 1)
        + normal_lpdf(beta0 | 0, 1)
        + normal_lpdf(beta1 | 0, 1);
}

generated quantities {
    vector[N] log_weight_rep; // replications from posterior predictive dist

    for (n in 1:N) {
        real log_weight_hat_n = beta0 + beta1 * log_gest[n];
        log_weight_rep[n] = normal_rng(log_weight_hat_n, sigma);
    }
}
```

PPCs

Now that we have a set of replicated datasets, can decide on a test statistic $t(\theta)$ that is a summary of interest

- ▶ Things like: median, skew, min, max, proportion under 2500 grams (low birth weight)
- ▶ Calculate test statistic for our data $t(y)$ and for each replicated dataset $t(y_{rep})$ and compare
- ▶ Appropriate test statistic is context dependent
- ▶ Bad test statistics are highly dependent on parameters in the model (e.g. mean, variance)

Example: birthweight

Model 1:

$$\log(y_i) \sim (\beta_0 + \beta_1 \log(x_i), \sigma^2)$$

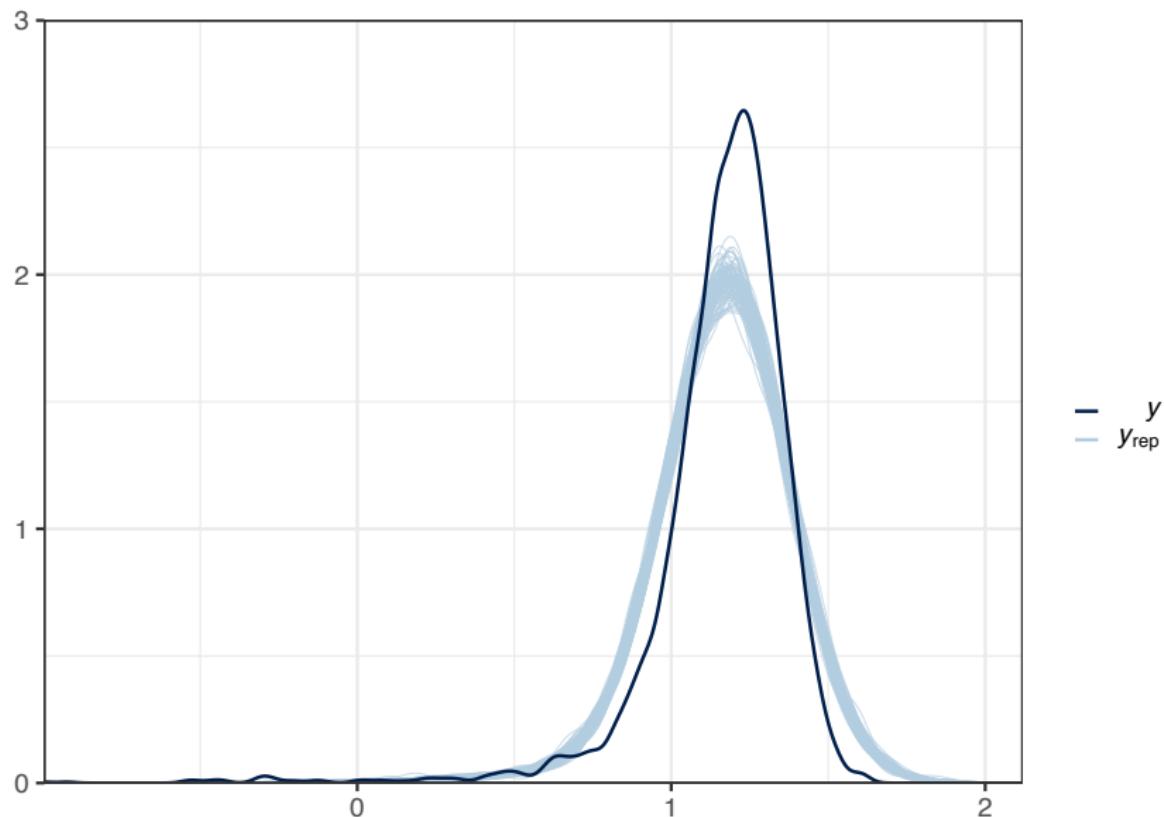
Model 2: preterm + interaction

$$\log(y_i) \sim (\beta_0 + \beta_1 \log(x_i) + \gamma_0 z_i + \gamma_1 \log(x_i) z_i, \sigma^2)$$

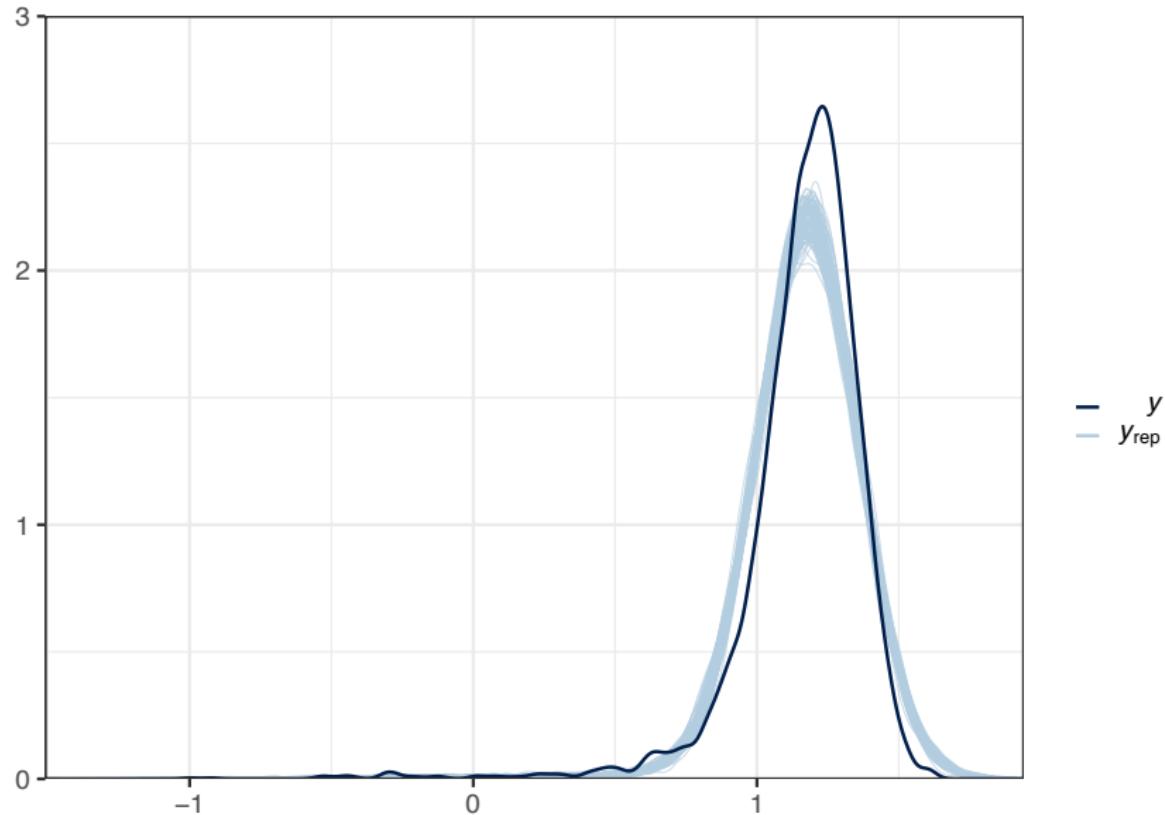
- ▶ y_i is weight in kg
- ▶ x_i is gestational age in weeks
- ▶ z_i is preterm (0 or 1, if gestational age is less than 30 weeks)

PPCs: Compare our dataset with 100 replicates

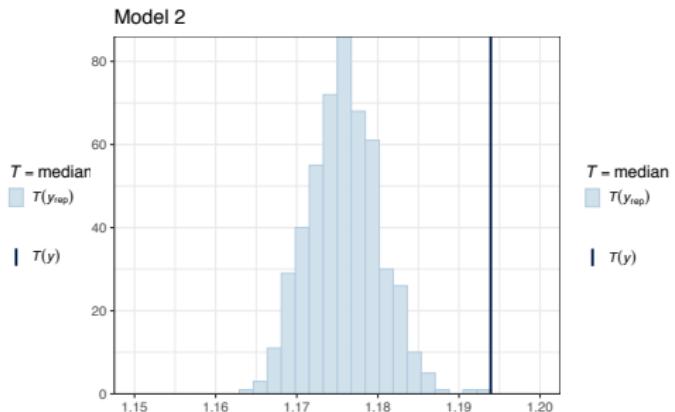
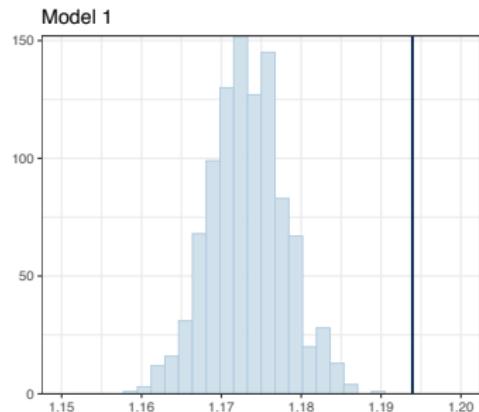
Model 1:



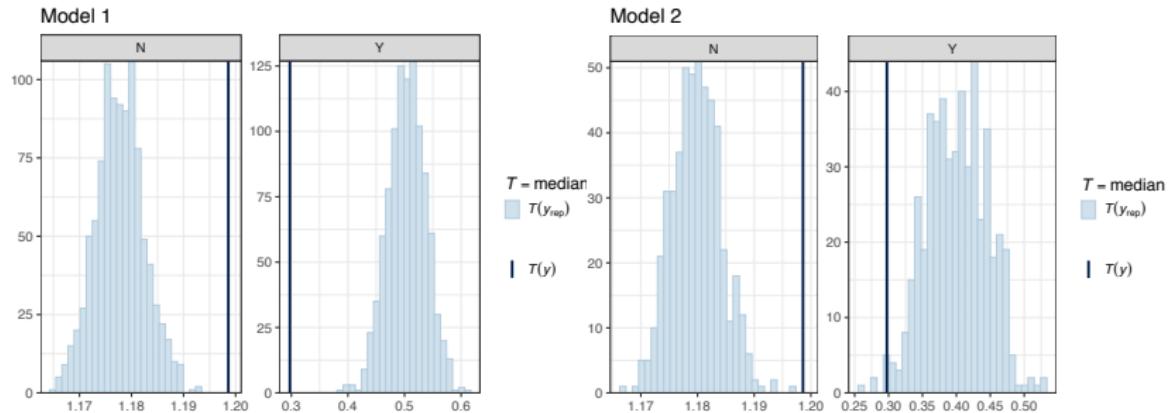
Model 2



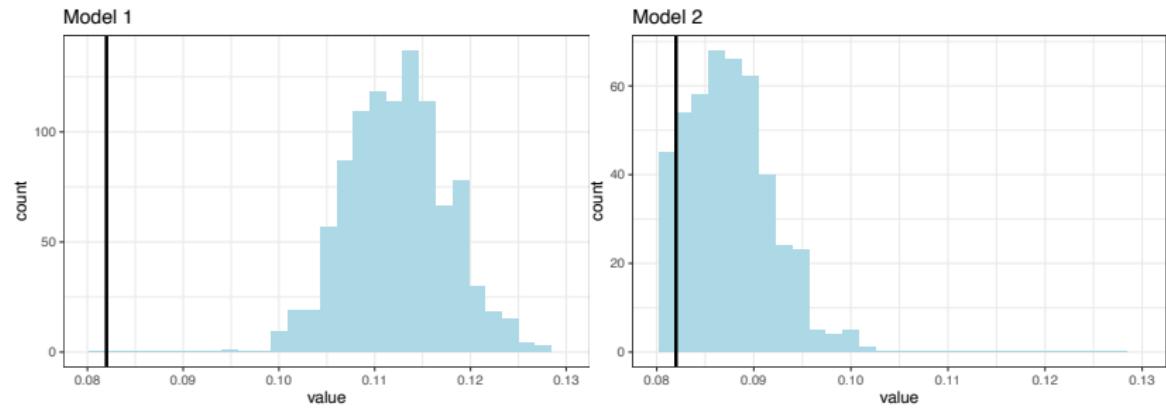
Look at median



Look at median by preterm



Look at proportion low birthweight (< 2.5kg)



LOO-CV

Compare data to itself

- ▶ In PPC we are using the data twice: once to fit model and once to check
- ▶ It would be nice to know how well the model would do fitting to a new dataset
- ▶ I.e. interested in **out of sample** prediction accuracy
- ▶ We usually don't have a new dataset

Idea: split dataset into **train** and **test** datasets (e.g. 80/20% split)

- ▶ Fit the model using training set
- ▶ Check the model using testing set
 - ▶ check based on some metric e.g. MSE

Training and testing over and over

- ▶ An issue with just doing a train/test split once is that the metric is highly dependent on the test set chosen
- ▶ One way to get around this is to split the data into train/test multiple times
- ▶ This is called **cross validation**
- ▶ **Leave-one-out cross validation** (LOO-CV) uses single observations as test sets.

Idea:

- ▶ Use all data except i , i.e. use \mathbf{y}_{-i} to predict y_i .
- ▶ Ideally, we would fit the model to \mathbf{y}_{-i} and get $p(y_i|\mathbf{y}_{-i})$, the **leave-one-out predictive density**.

LOO-CV

Ideally, we would fit the model to \mathbf{y}_{-i} and get $p(y_i|\mathbf{y}_{-i})$, the **leave-one-out predictive density**.

If we had infinite time and resources, we could for $i = 1, \dots, n$

1. Remove the i th data point
2. Rerun the model
3. Calculate $p(y_i|\mathbf{y}_{-i})$

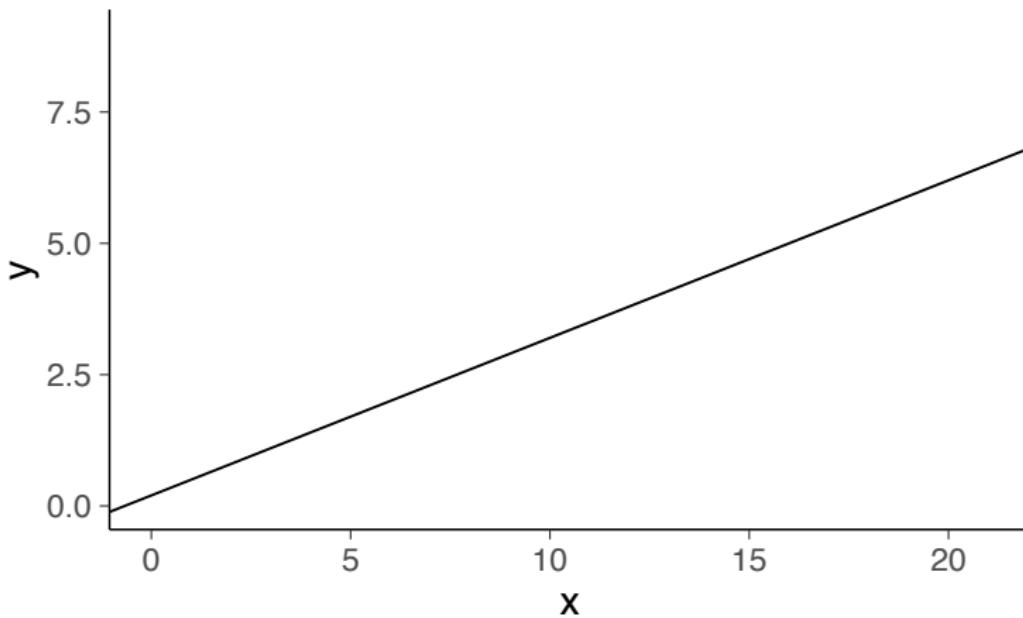
Then calculate the **expected log pointwise predictive density**

$$\text{elpd}_{\text{LOO}} = \sum_{i=1}^n \log p(y_i|\mathbf{y}_{-i})$$

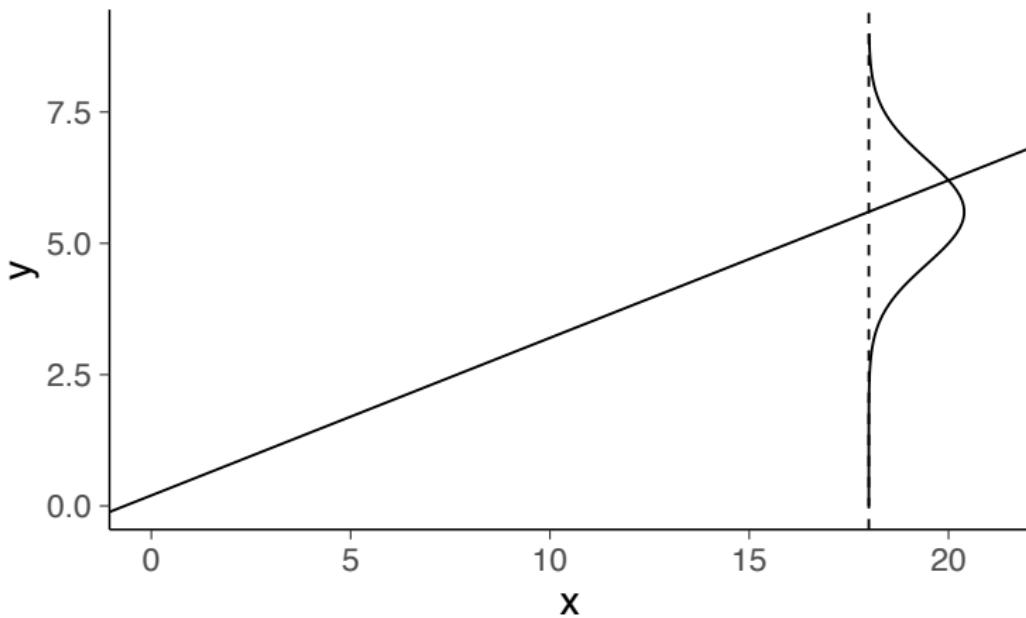
Then can compare elpd_{LOO} across different models, etc. The bigger the better.

Visualizing $p(y_i|\mathbf{y}_{-i})$ (Source)

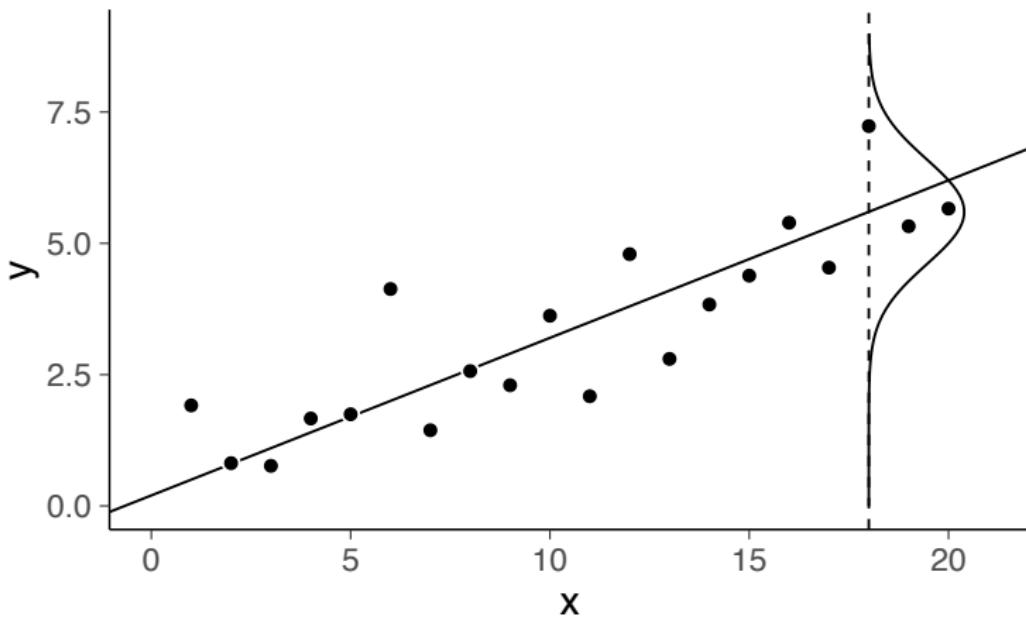
True mean $y = a + bx$



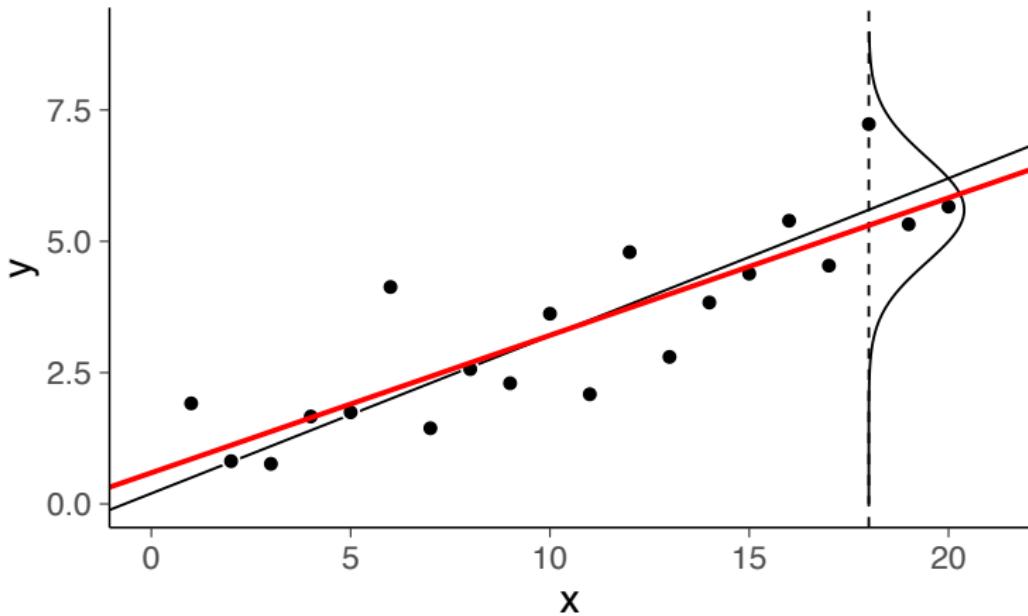
True mean and sigma



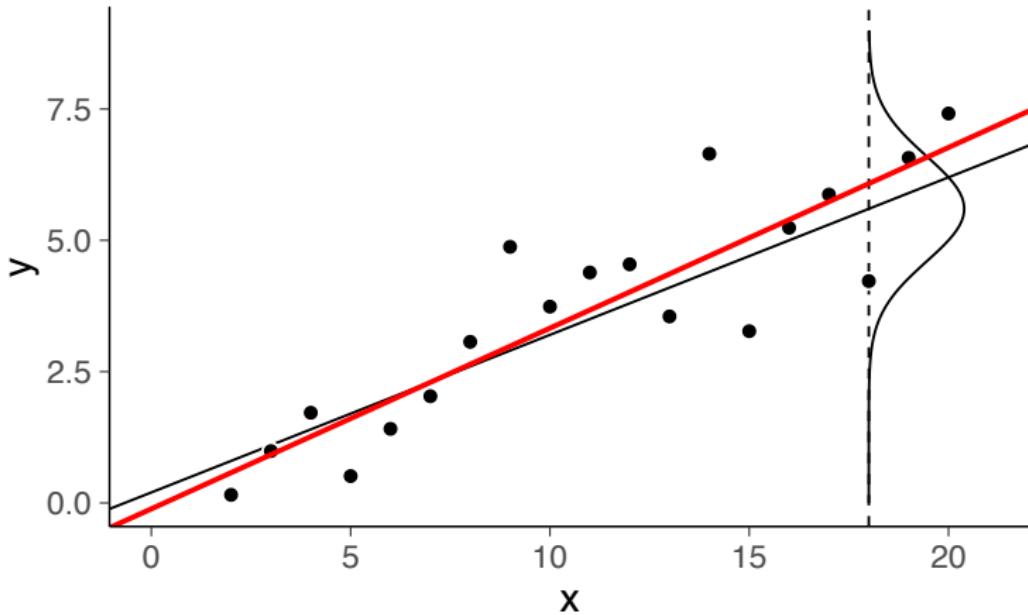
Data



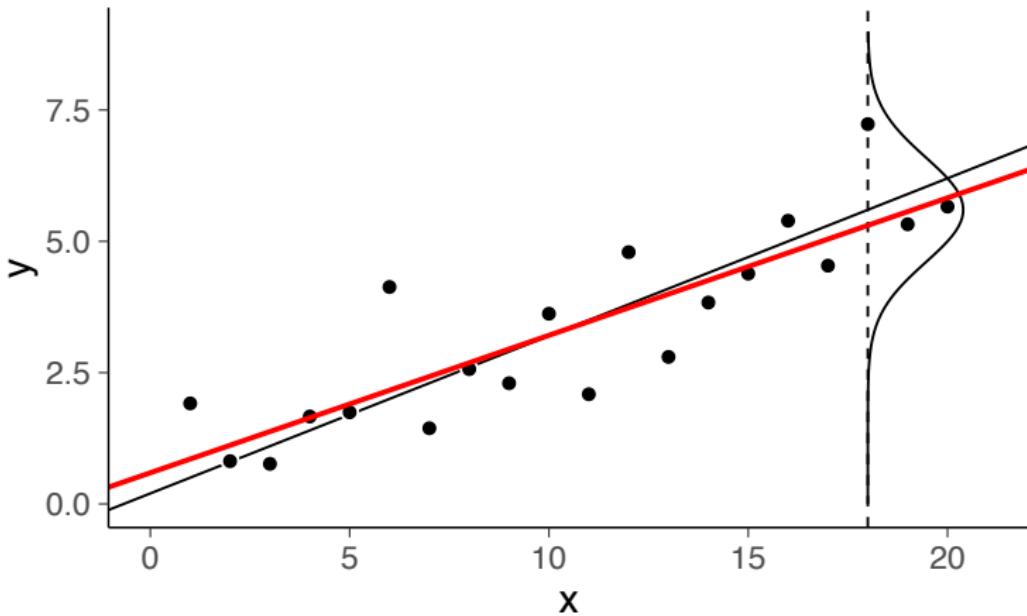
Posterior mean



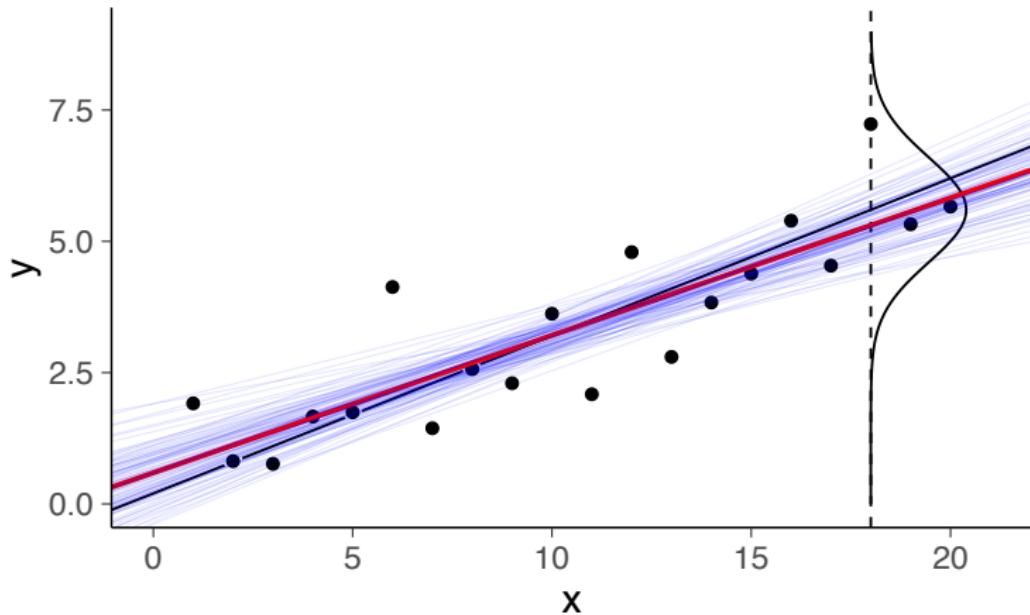
Posterior mean, alternative data realisation



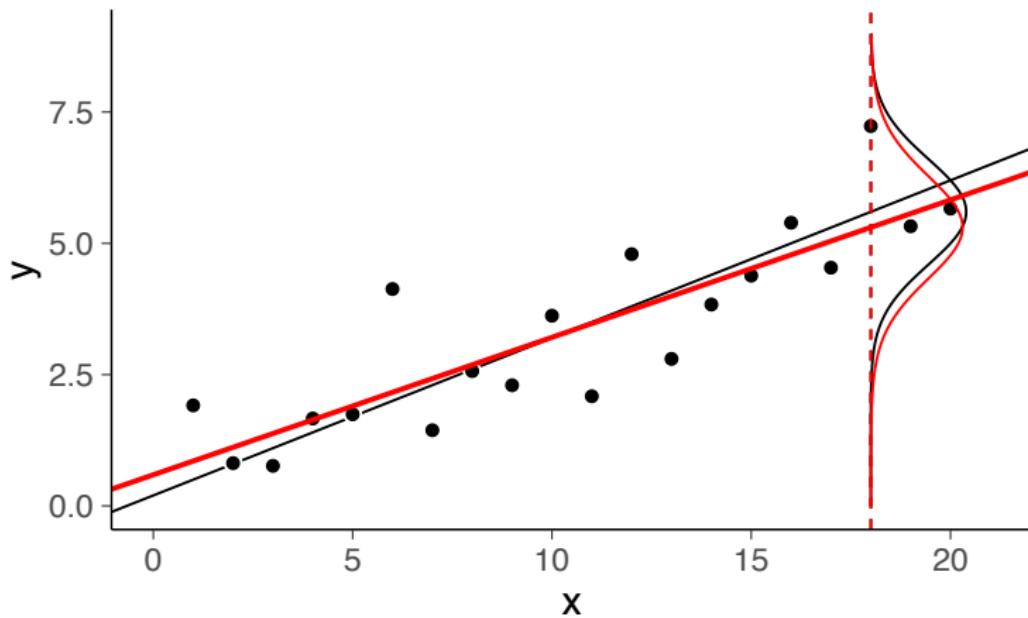
Posterior mean



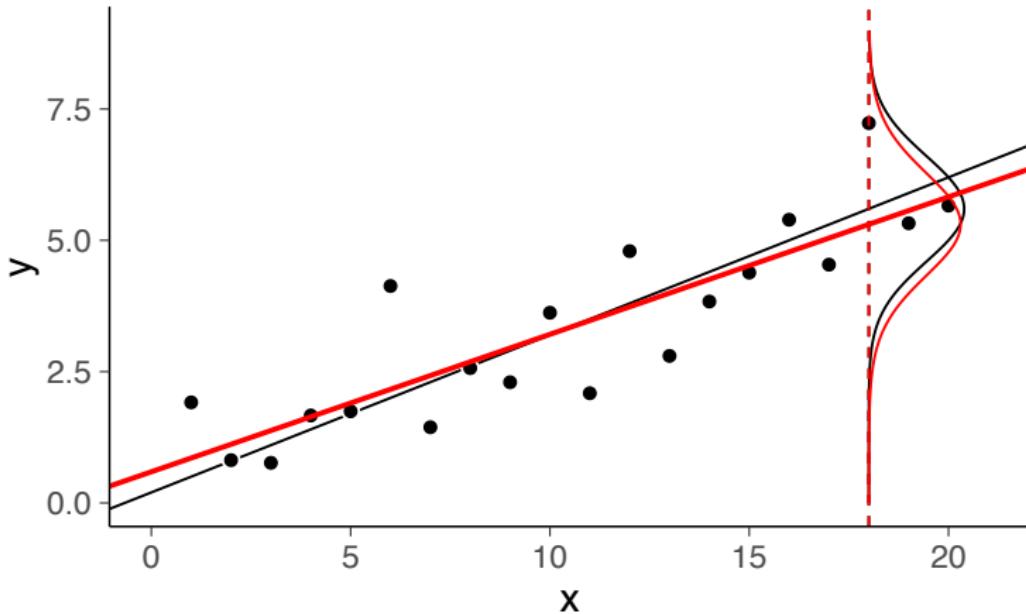
Posterior draws



Posterior predictive distribution

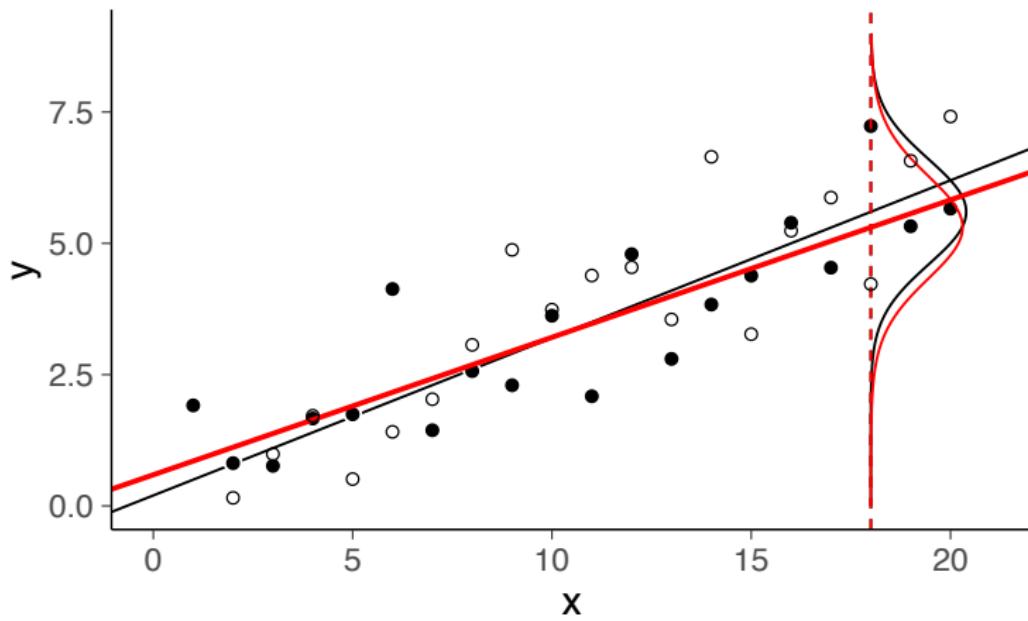


Posterior predictive distribution

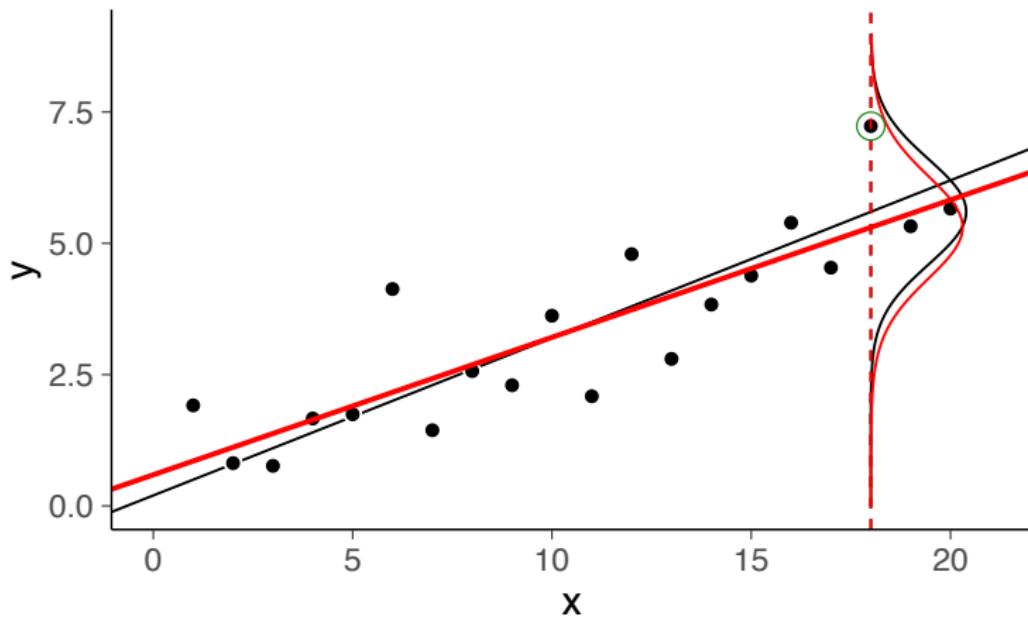


$$p(\tilde{y}|\tilde{x}=18, x, y) = \int p(\tilde{y}|\tilde{x}=18, \theta)p(\theta|x, y)d\theta$$

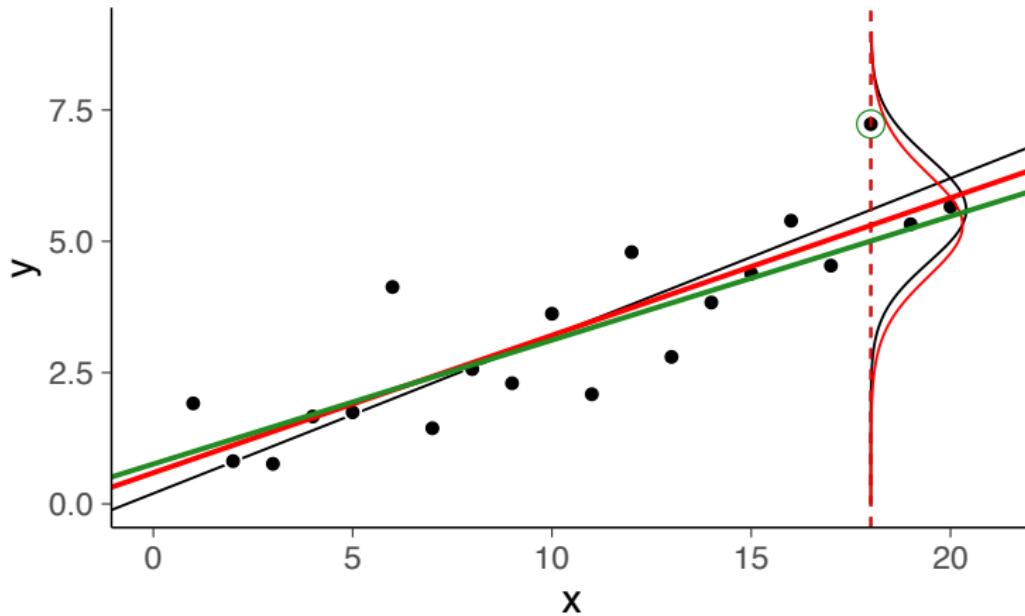
New data



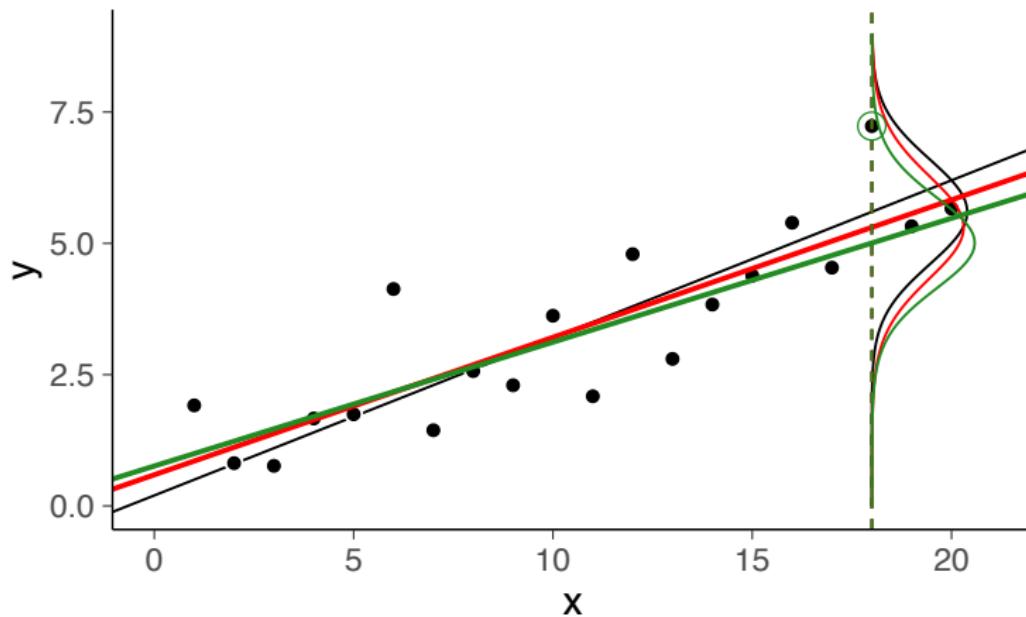
Posterior predictive distribution



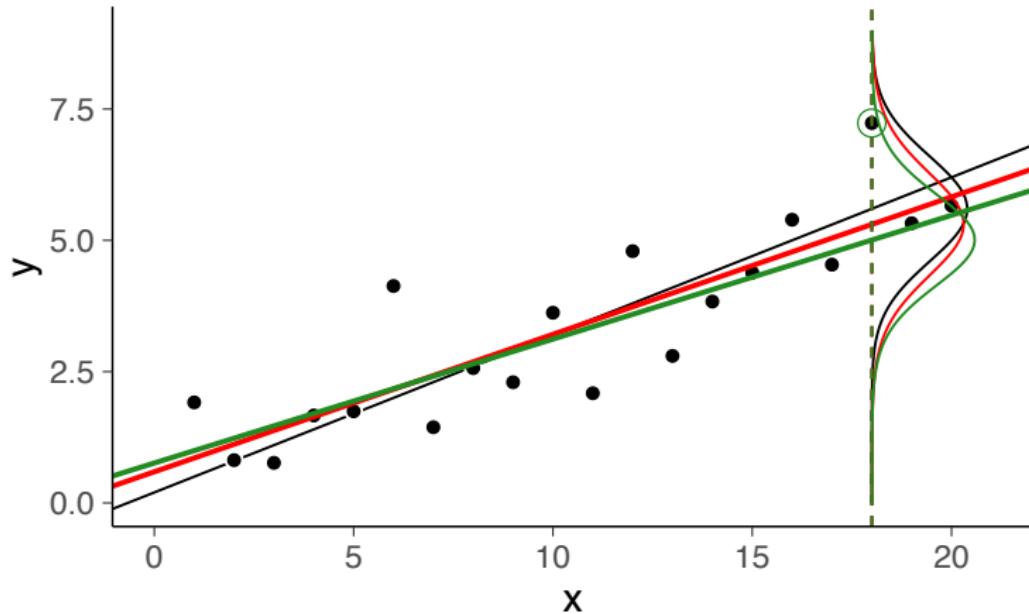
Leave-one-out mean



Leave-one-out predictive distribution

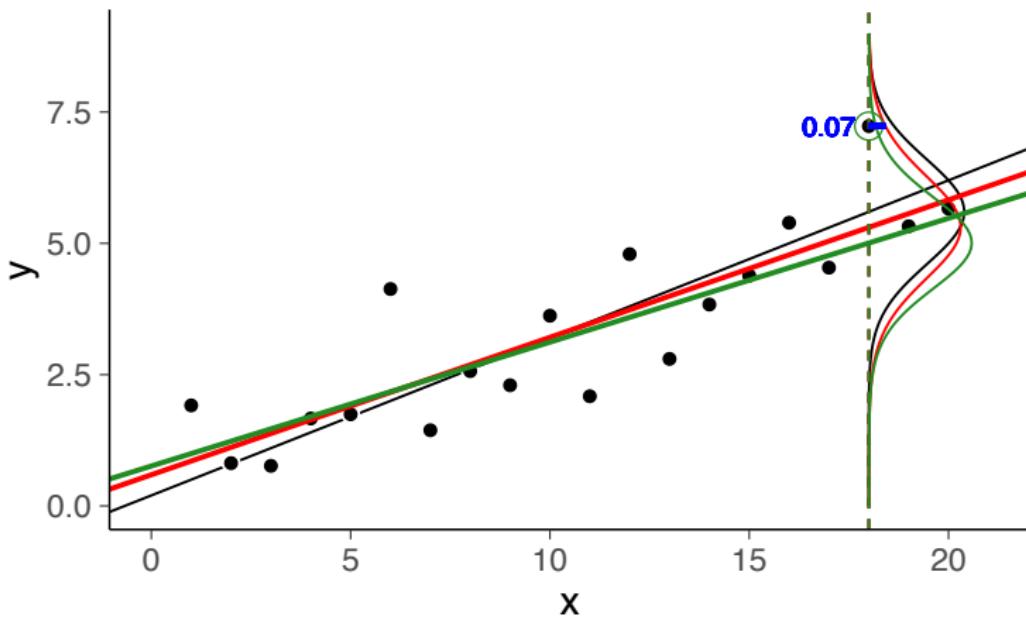


Leave-one-out predictive distribution

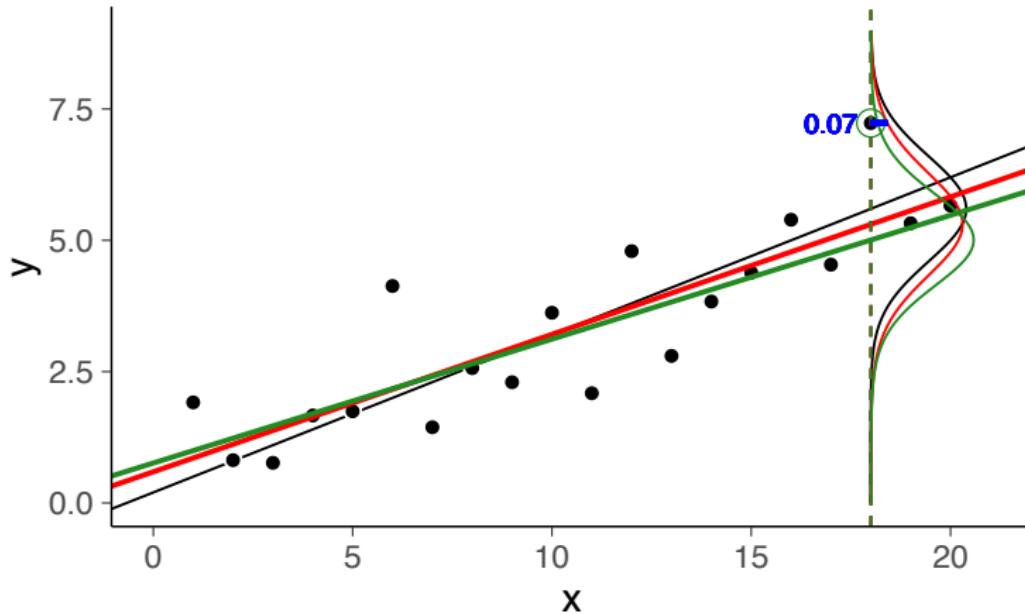


$$p(\tilde{y}|\tilde{x} = 18, x_{-18}, y_{-18}) = \int p(\tilde{y}|\tilde{x} = 18, \theta) p(\theta|x_{-18}, y_{-18}) d\theta$$

Posterior predictive density

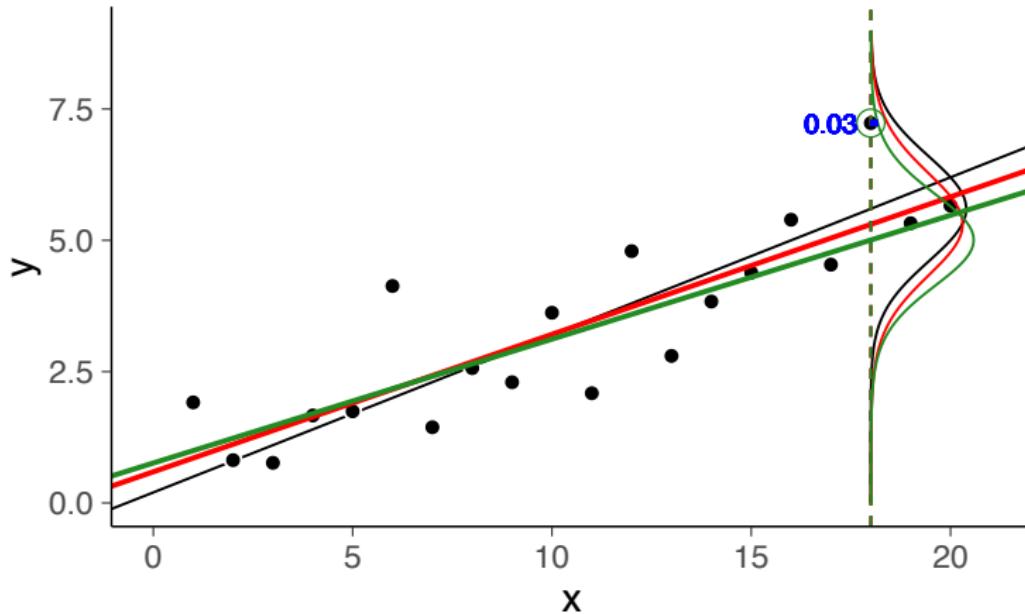


Posterior predictive density



$$p(\tilde{y} = y_{18} | \tilde{x} = 18, x, y) \approx 0.07$$

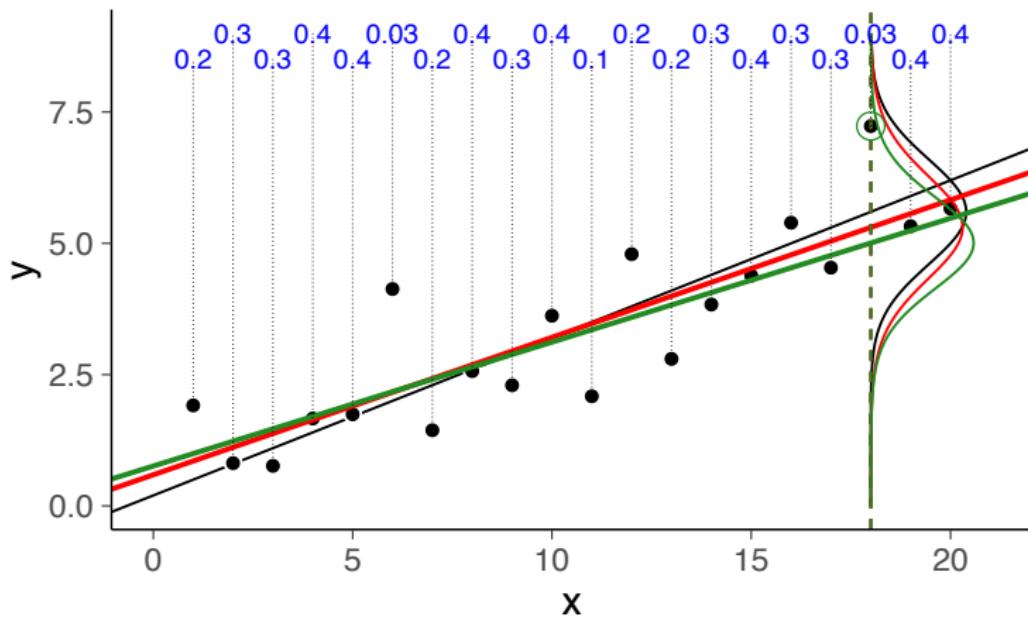
Leave-one-out predictive density



$$p(\tilde{y} = y_{18} | \tilde{x} = 18, x, y) \approx 0.07$$

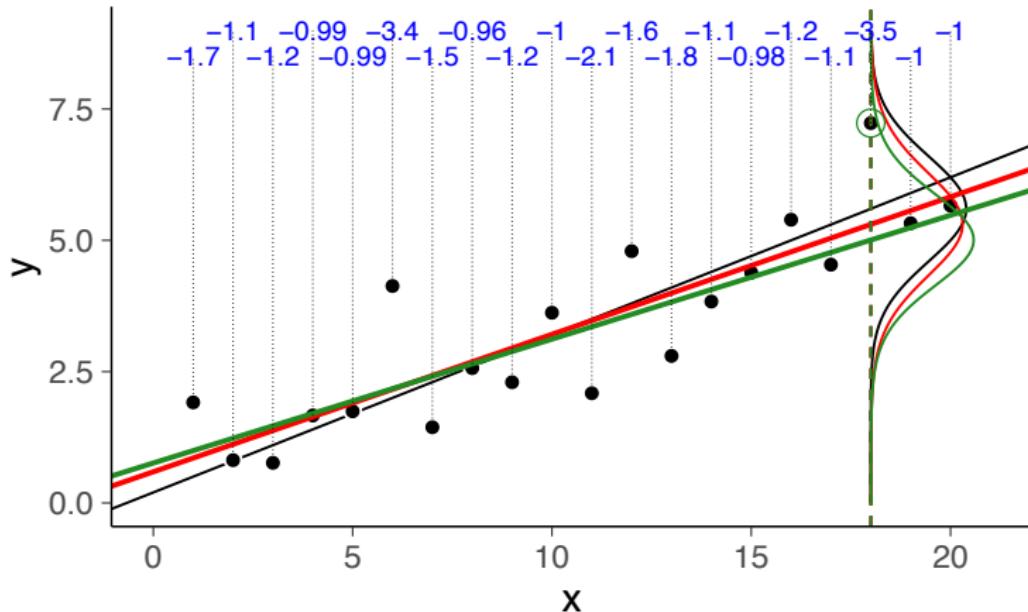
$$p(\tilde{y} = y_{18} | \tilde{x} = 18, x_{-18}, y_{-18}) \approx 0.03$$

Leave-one-out predictive densities



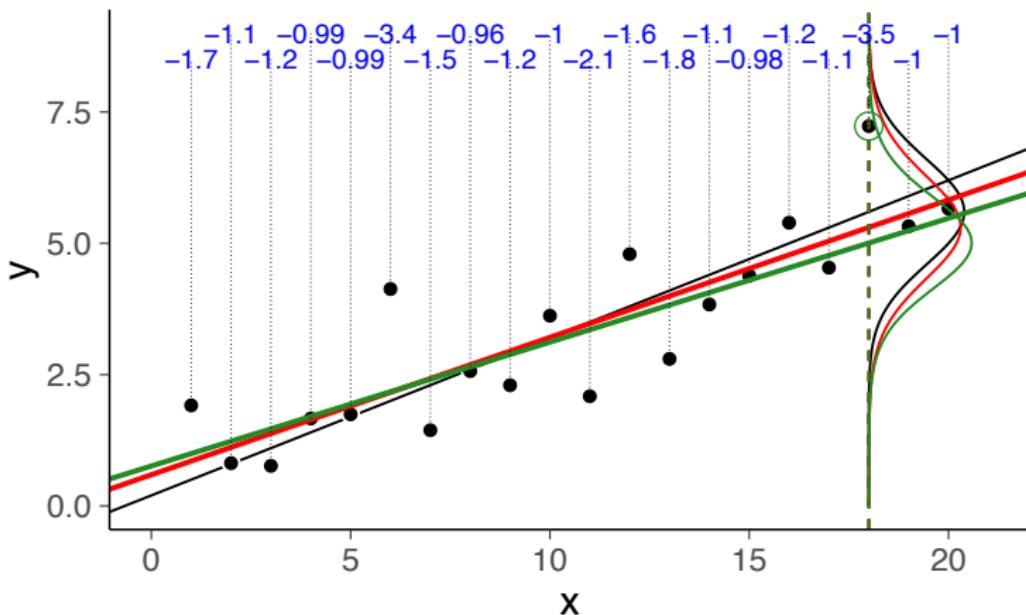
$$p(y_i|x_i, x_{-i}, y_{-i}), \quad i = 1, \dots, 20$$

Leave-one-out log predictive densities



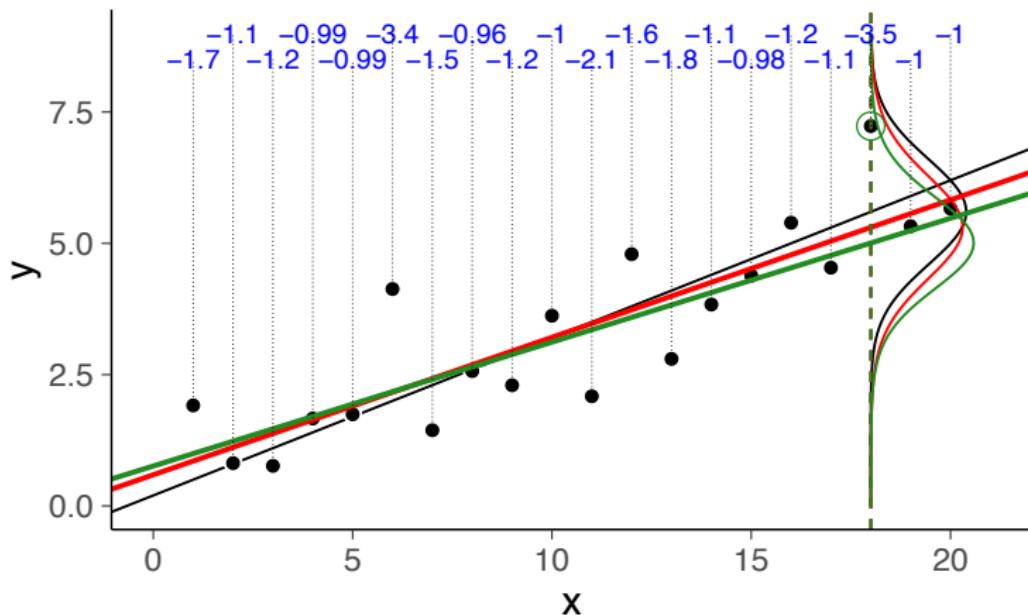
$$\log p(y_i|x_i, x_{-i}, y_{-i}), \quad i = 1, \dots, 20$$

Leave-one-out log predictive densities



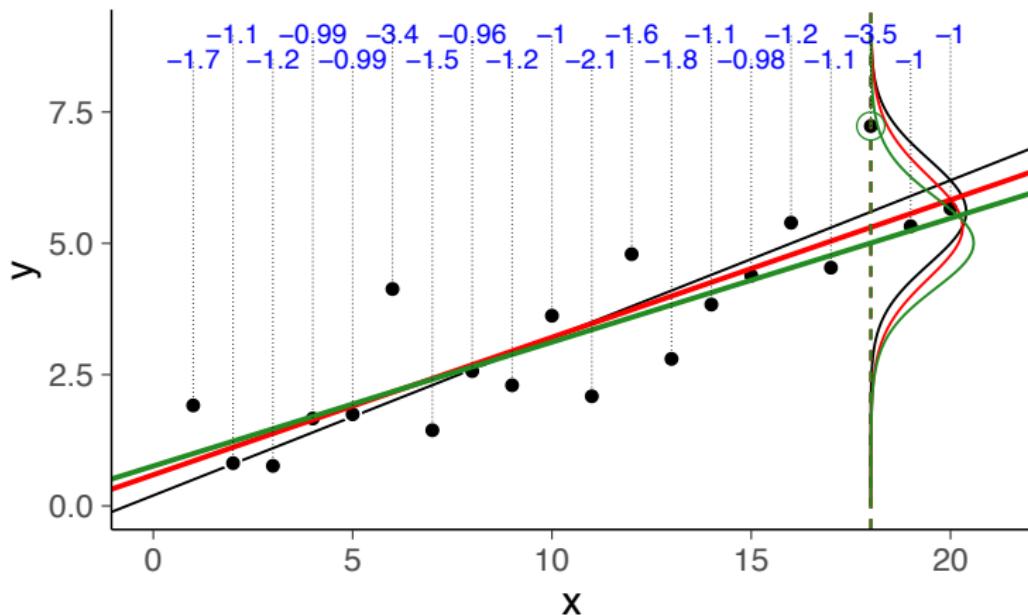
$$\sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

Leave-one-out log predictive densities



$$\text{elpd}_{\text{loo}} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

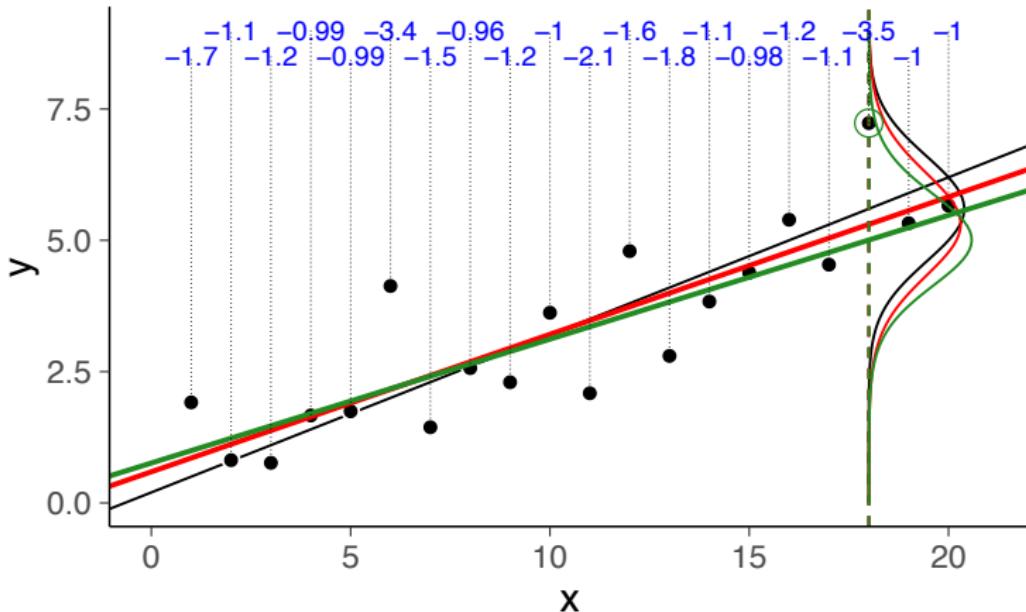
Leave-one-out log predictive densities



$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

unbiased estimate of log posterior pred. density for new data

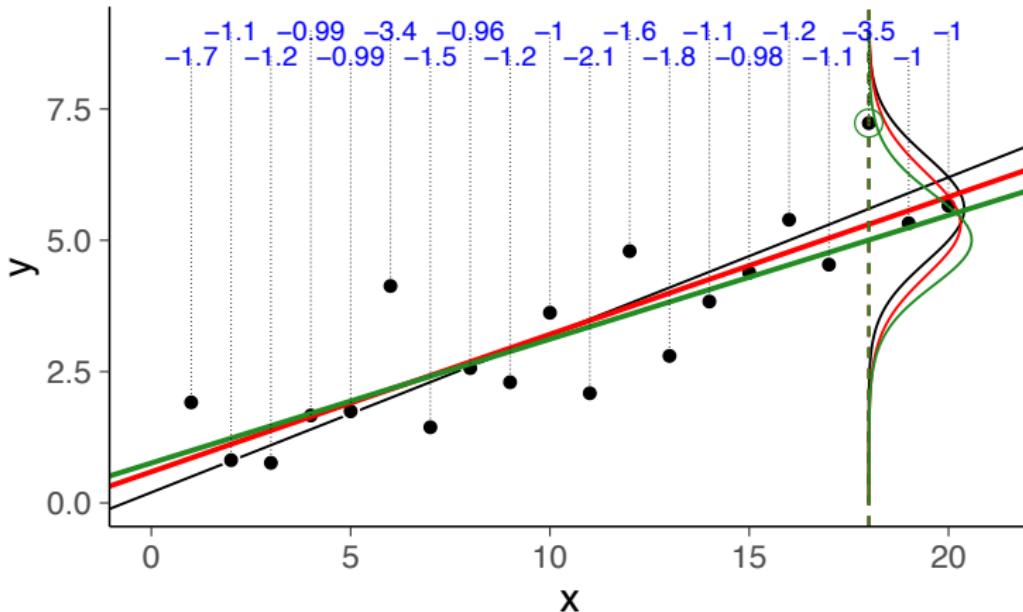
Leave-one-out log predictive densities



$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i|x_i, x_{-i}, y_{-i}) \approx -29.5$$

$$\text{lpd} = \sum_{i=1}^{20} \log p(y_i|x_i, x, y) \approx -26.8$$

Leave-one-out log predictive densities

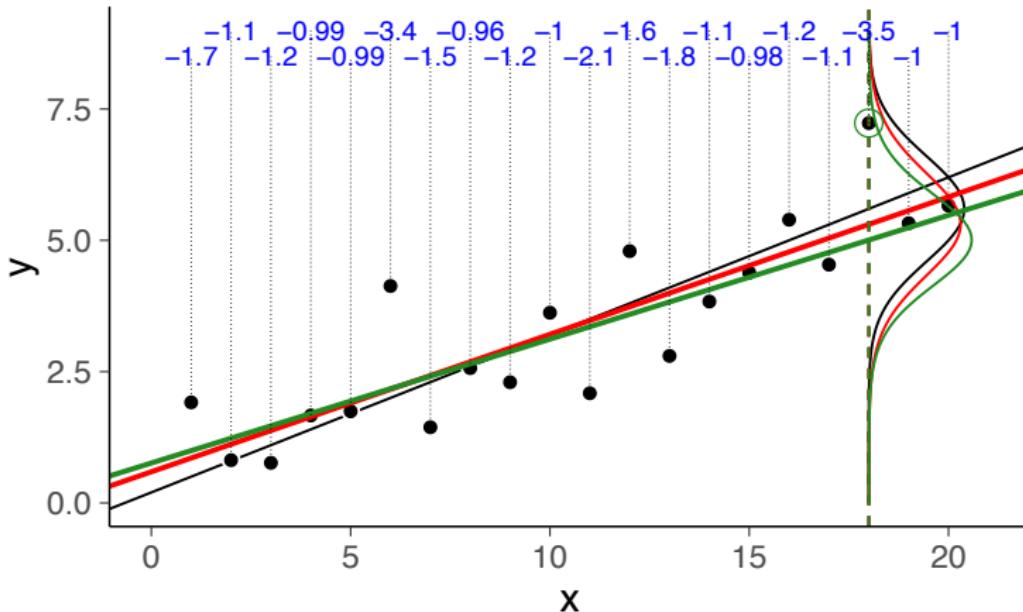


$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i|x_i, x_{-i}, y_{-i}) \approx -29.5$$

$$\text{lpd} = \sum_{i=1}^{20} \log p(y_i|x_i, x, y) \approx -26.8$$

$$\text{p_loo} = \text{lpd} - \text{elpd_loo} \approx 2.7$$

Leave-one-out log predictive densities



$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

$$\text{SE} = \text{sd}(\log p(y_i | x_i, x_{-i}, y_{-i})) \cdot \sqrt{20} \approx 3.3$$

PSIS-LOO

PSIS-LOO

- ▶ But we don't generally have infinite time and resources, so want to estimate $p(y_i|\mathbf{y}_{-i})$.
- ▶ PSIS-LOO refers to using Pareto-smoothed importance sampling to do this

Details in Vehtari et al (2017) but broadly...

This IS bit

Estimating properties of one distribution using samples from another distribution.

We want to know about the LOO predictive distribution

$$p(y_i | \mathbf{y}_{-i}) = \int p(y_i | \theta) p(\theta | \mathbf{y}_{-i}) d\theta = \int p(y_i | \theta) \frac{p(\theta | \mathbf{y}_{-i})}{p(\theta | \mathbf{y})} p(\theta | \mathbf{y}) d\theta$$

We can evaluate above with draws from the full posterior and using importance ratios

$$r_i^s = \frac{1}{p(y_i | \theta^s)} \propto \frac{p(\theta^s | \mathbf{y}_{-i})}{p(\theta^s | \mathbf{y})}$$

which implies

$$p(y_i | \mathbf{y}_{-i}) \approx \frac{\sum_{s=1}^S r_i^s p(y_i | \theta^s)}{\sum_{s=1}^S r_i^s} \approx \frac{1}{\frac{1}{S} \sum_{s=1}^S \frac{1}{p(y_i | \theta^s)}}$$

The PS bit

- ▶ The direct use of the importance ratios/weights can lead to instability because the ratios can have high or infinite variance.
- ▶ To get around this, weights are smoothed by fitting a generalized Pareto distribution to the upper tail (largest 20%) of the importance weights.
- ▶ With new weights w_i^s we can calculate the PSIS estimate of the LOO expected log pointwise predictive density

$$\widehat{\text{elpd}}_{\text{psis-loo}} = \sum_{i=1}^n \log \left(\frac{\sum_{s=1}^S w_i^s p(y_i | \theta^s)}{\sum_{s=1}^S w_i^s} \right)$$

- ▶ Vehtari et al 2017 show that PSIS performs better than other existing methods, and they provide a check to see if the result is reliable or whether you should obtain $p(y_i | \mathbf{y}_{-i})$ by fitting the model to \mathbf{y}_{-i} instead.

An added bonus

The generalized Pareto distribution has the form

$$p(z) = \frac{1}{\sigma} (1 + kz)^{-1/k-1}$$

- ▶ The key parameter here is k , which controls how many moments the tail distribution has
- ▶ If the estimate of k is large, there are more outlying weights, and the LOO predictive distribution for point i is very different from the full predictive distribution
- ▶ So point i is ‘influential’ in some sense.
- ▶ \hat{k} is larger than 0.7 then
 - ▶ point is flagged as influential
 - ▶ PSIS-LOO approximation of $p(y_i | \mathbf{y}_{-i})$ may not be very good, it is recommended to get $p(y_i | \mathbf{y}_{-i})$ directly

PSIS-LOO in R

- ▶ This is what the loo package in R uses
- ▶ In practice, what do we need from the model?

```
model {  
    // Log-likelihood  
    target += normal_lpdf(log_weight | beta0 + beta1 * log_gest, sigma);  
  
    // Log-priors  
    target += normal_lpdf(sigma | 0, 1)  
        + normal_lpdf(beta0 | 0, 1)  
        + normal_lpdf(beta1 | 0, 1);  
}  
generated quantities {  
    vector[N] log_lik;      // pointwise log-likelihood for LOO  
    vector[N] log_weight_rep; // replications from posterior predictive dist  
  
    for (n in 1:N) {  
        real log_weight_hat_n = beta0 + beta1 * log_gest[n];  
        log_lik[n] = normal_lpdf(log_weight[n] | log_weight_hat_n, sigma);  
        log_weight_rep[n] = normal_rng(log_weight_hat_n, sigma);  
    }  
}
```

PSIS-LOO in R

```
library(loo)
loglik1 <- as.matrix(mod1, pars = "log_lik")
loglik2 <- as.matrix(mod2, pars = "log_lik")
loo1 <- loo(loglik1, save_psis = TRUE)
loo2 <- loo(loglik2, save_psis = TRUE)
loo2

##
## Computed from 500 by 3842 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo    1552.8  70.0
## p_loo       14.8   2.3
## looic     -3105.6 139.9
## -----
## Monte Carlo SE of elpd_loo is 0.2.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

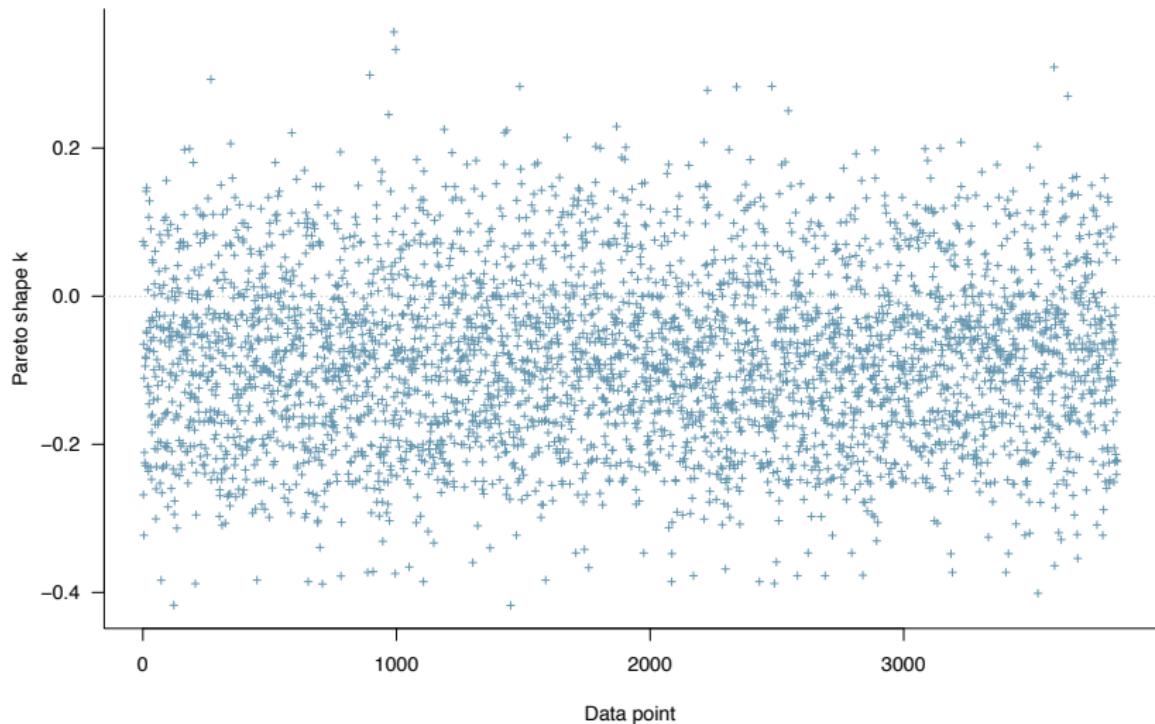
Compare

```
compare(loo1, loo2)
```

```
## elpd_diff      se
##     175.1      36.1
```

Plot k values: Model 2

PSIS diagnostic plot

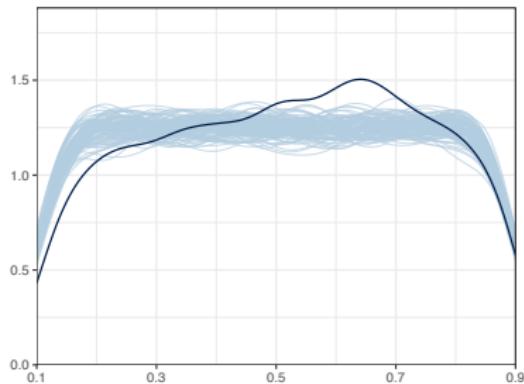


All outliers are extremely preterm.

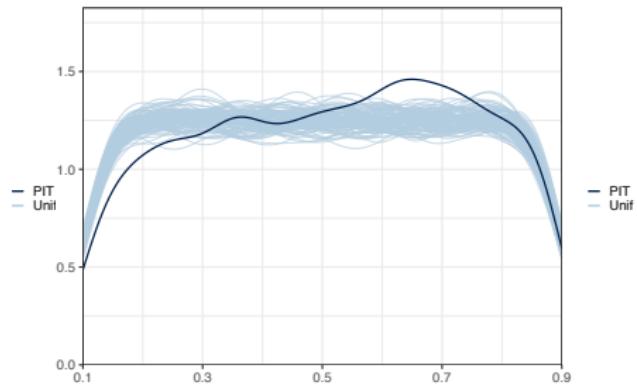
LOO probability integral transform (PIT)

- ▶ LOO-PIT $_i = P(\tilde{y}_i \leq y_i | \mathbf{y}_{-i})$ i.e. where does y_i fall in its LOO predictive distribution
- ▶ If $y_i \sim p(\tilde{y}_i | \mathbf{y}_{-i})$ then LOO-PIT $_i \sim U(0, 1)$
- ▶ So can compare PITs to uniforms

Model 1



Model 2



Briefly: Information Criteria

- ▶ You are probably used to seeing the Akaike Information Criterion to compare models, $AIC = -2 \log(\widehat{p(y|\theta(y))}) + 2 \cdot k$ where k is the number of parameters and $\widehat{\theta(y)}$ is the MLE.
- ▶ Bayesian setting requires estimation of an 'effective number of parameters' to replace k that takes prior info into account
- ▶ E.g. Widely Applicable Information Criterion (WAIC)

$$WAIC = -2\widehat{\text{lpd}} + 2p_{WAIC}$$

Where lpd is the log pointwise predictive density,

$$\text{lpd} = \sum_{i=1}^n \log p(y_i|y) = \sum_{i=1}^n \log \int p(y_i|\theta) p(\theta|y) d\theta$$

estimated as $= \sum_{i=1}^n \log \left(\frac{1}{S} \sum_{s=1}^S p(y_i|\theta^s) \right)$ and the effective number of parameters is

$$p_{\text{waic}} = \sum_{i=1}^n \text{var}_{\text{post}} (\log p(y_i|\theta))$$

Briefly: Information Criteria

- ▶ As for the other criteria, WAIC is the combination of predictive accuracy and a bias adjustment.
- ▶ WAIC aims to approximate -2elpd
- ▶ Asymptotically closest to Bayesian leave-one-out accuracy
- ▶ Veharti et al 2017 shows that PSIS-LOO tends to perform better than WAIC in terms of estimating $\sum_i \log(p(y_i|\mathbf{y}_{-i}))$

Briefly: Actually training and testing

Set-up: Fit the model to a training set and validate predictive accuracy of the model for the left-out test data

- ▶ Bad: how to split, variable results, computationally intensive
- ▶ Good: actual predictions for new data, can be more convincing

May make sense especially with temporal data (neonatal mortality example)

Briefly: Actually training and testing

Measures given by some summary of errors, e.g. relative error

$$e_i = (y_i - \hat{y}_i) / \hat{y}_i$$

- ▶ could look at mean error, median error, mean squared error, median absolute error
- ▶ could also look at coverage of prediction intervals.

Validation measures, model comparison

	Expected	≤ 2005	> 2005
Mean absolute relative error	–	0.05	0.09
80% coverage	≥ 0.80	0.90	0.77
90% coverage	≥ 0.90	0.94	0.84
95% coverage	≥ 0.95	0.96	0.90

(From Alexander and Alkema, 2018)

Summary

What are all these things?

- ▶ $p(\tilde{y})$
- ▶ $p(\tilde{y}|\mathbf{y})$
- ▶ $p(y_i|\mathbf{y})$
- ▶ $p(y_i|\mathbf{y}_{-i})$
- ▶ lpd
- ▶ $\widehat{\text{lpd}}$
- ▶ $\underline{\text{elpd}}_{LOO}$
- ▶ $\underline{\text{elpd}}_{PSIS-LOO}$

Summary

General advice for model checking:

Plot, plot, plot

- ▶ Check your priors
- ▶ PPCs by different groups of interest
- ▶ LOO to not only compare models but also to inform model specification (influential points)

Also not discussed today, but plot

- ▶ Data and model fit
- ▶ residuals, grouped by whatever characteristics of data/predictors you can think of, or a scatter against predictors