

Predicting concrete quality



- How accurate the model is?
- Is it better than predicting with random guess?
- Is it possible that the model has overfitted?
- Is model B better than model A?

Outline

- What is cross-validation
 - Leave-one-out cross-validation (`elpd_loo`, `p_loo`)
 - Uncertainty in LOO (SE)
- Fast cross-validation
 - PSIS and diagnostics in `loo` package (Pareto k, `n_eff`, Monte Carlo SE)
- Model comparison and selection (`elpd_diff`, `se`)

Next week

- When is cross-validation applicable?
 - data generating mechanisms and prediction tasks
 - leave-many-out cross-validation
- K -fold cross-validation
- Related methods (WAIC, *IC, BF)
- Model averaging
- Hypothesis testing
- Potential overfitting in model selection

Chapter 7

- 7.1 Measures of predictive accuracy
- 7.2 Information criteria and cross-validation
 - Instead of 7.2, read:
Vehtari, A., Gelman, A., Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*. 27(5):1413–1432. preprint at arxiv.org/abs/1507.04544.
 - See also
<https://users.aalto.fi/~ave/modelselection/CV-FAQ.html>

Next week

- 7.3 Model comparison based on predictive performance
- 7.4 Model comparison using Bayes factors
- 7.5 Continuous model expansion / sensitivity analysis
- 7.5 Example (may be skipped)

Predictive performance

- True predictive performance is found out by using it to make predictions and comparing predictions to true observations
 - external validation

Predictive performance

- True predictive performance is found out by using it to make predictions and comparing predictions to true observations
 - external validation
- Expected predictive performance
 - approximates the external validation

Predictive performance

- We need to choose the utility/cost function
 - more about these in lecture 10
- Application specific utility/cost functions are important
 - eg. money, life years, quality adjusted life years, etc.

Predictive performance

- We need to choose the utility/cost function
 - more about these in lecture 10
- Application specific utility/cost functions are important
 - eg. money, life years, quality adjusted life years, etc.
- If we are interested overall in the goodness of the predictive distribution, or we don't know (yet) the application specific utility, then good information theoretically justified choice is log-score

$$\log p(y^{\text{rep}} \mid y, M),$$

Stan and loo package

Model assessment:

Computed from 4000 by 20 log-likelihood matrix

	Estimate	SE
elpd_loo	-29.5	3.3
p_loo	2.7	1.0

MCSE of elpd_loo is 0.1.

MCSE and ESS estimates assume MCMC draws (r_{eff} in [0.6, 1.2]).

All Pareto k estimates are good ($k < 0.7$).

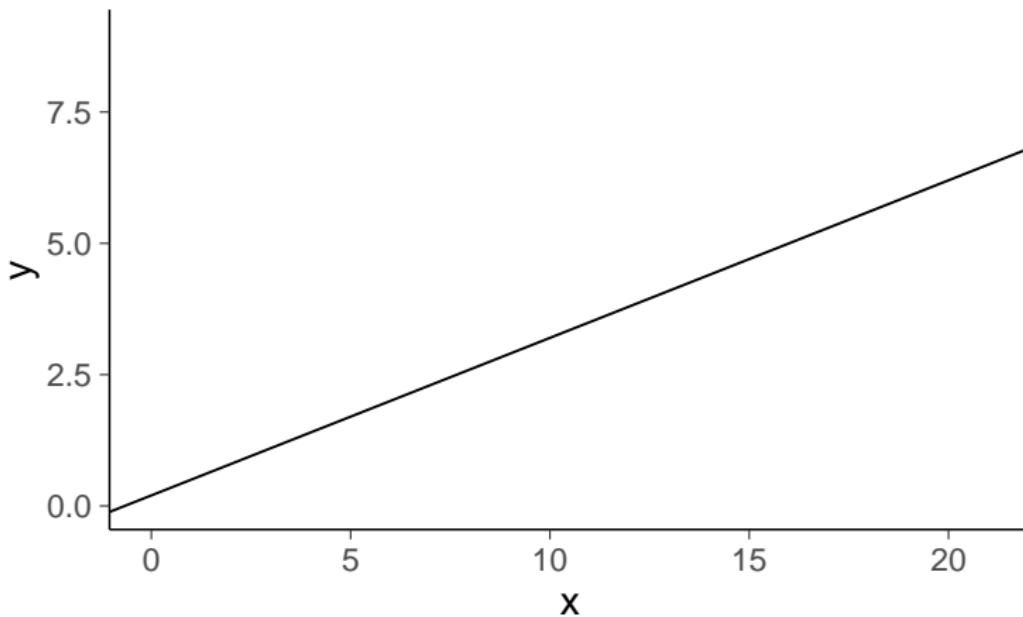
See `help('pareto-k-diagnostic')` for details.

Model comparison:

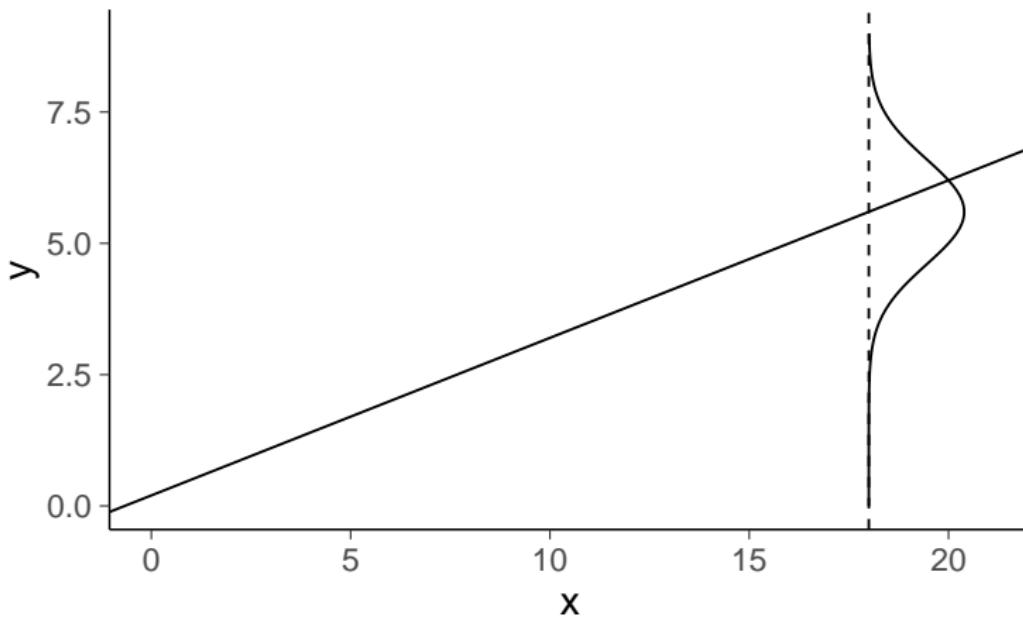
(negative 'elpd_diff' favors 1st model, positive favors 2nd)

elpd_diff	se
-0.2	0.1

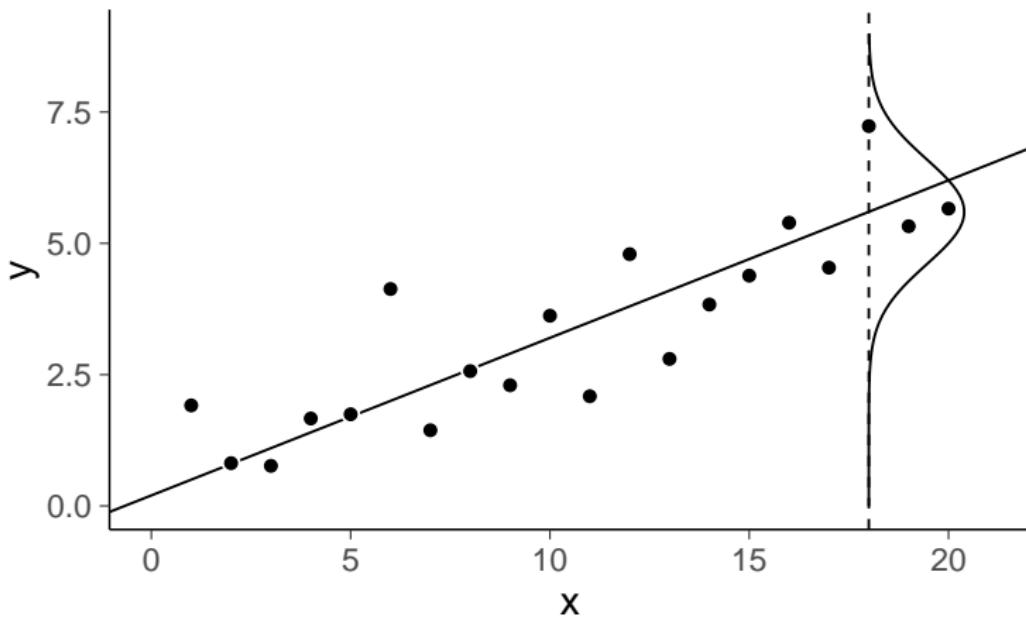
True mean $y = a + bx$



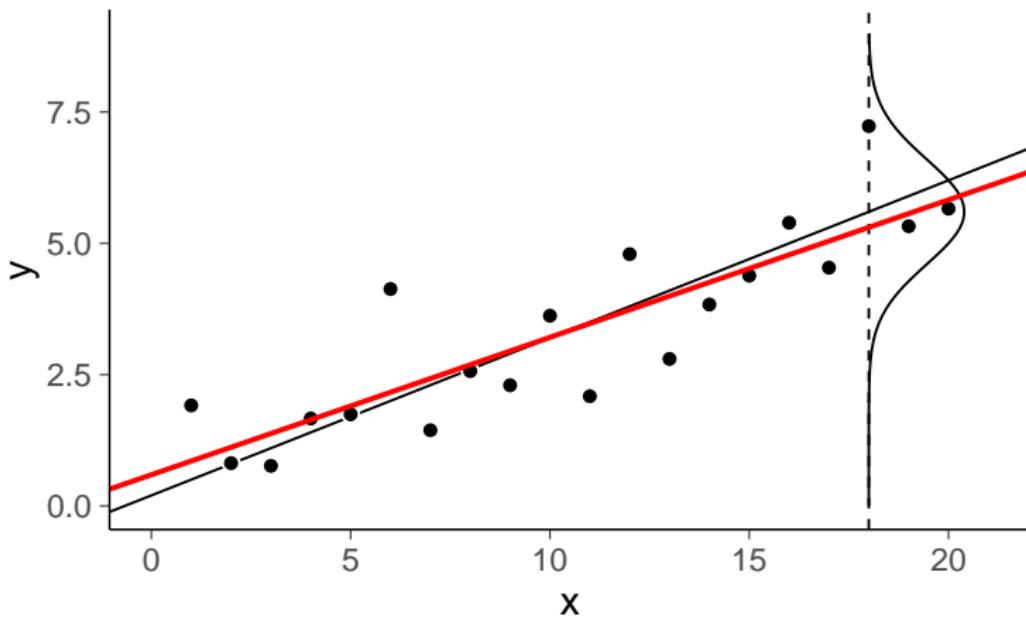
True mean and sigma



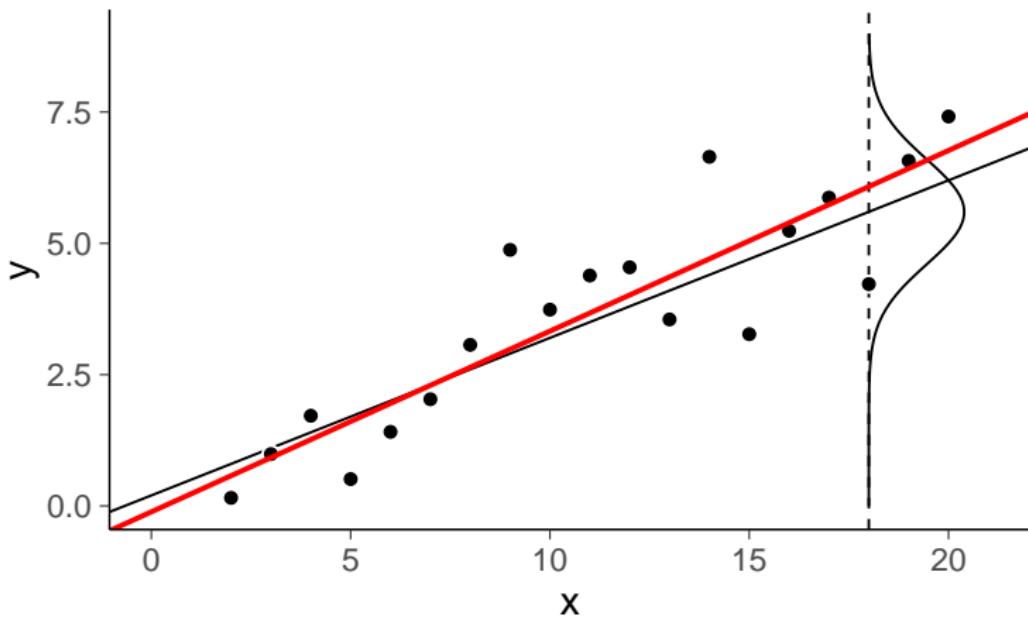
Data



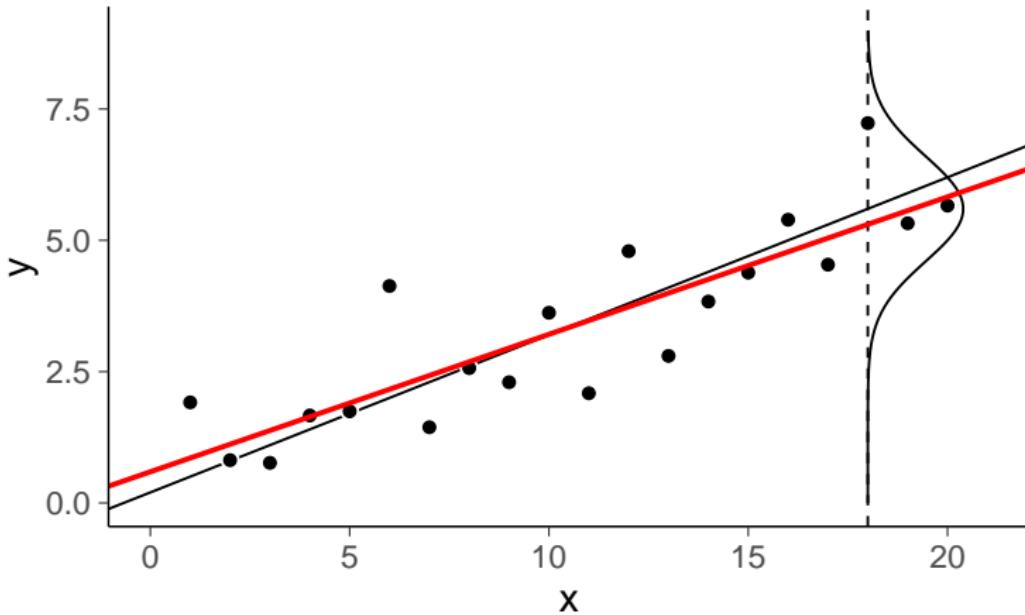
Posterior mean



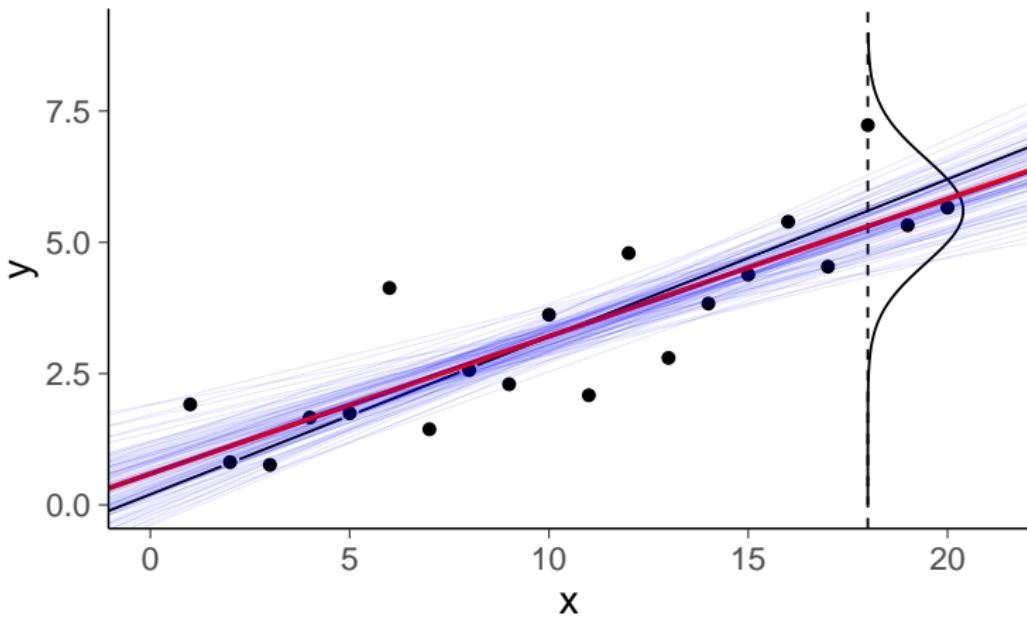
Posterior mean, alternative data realisation



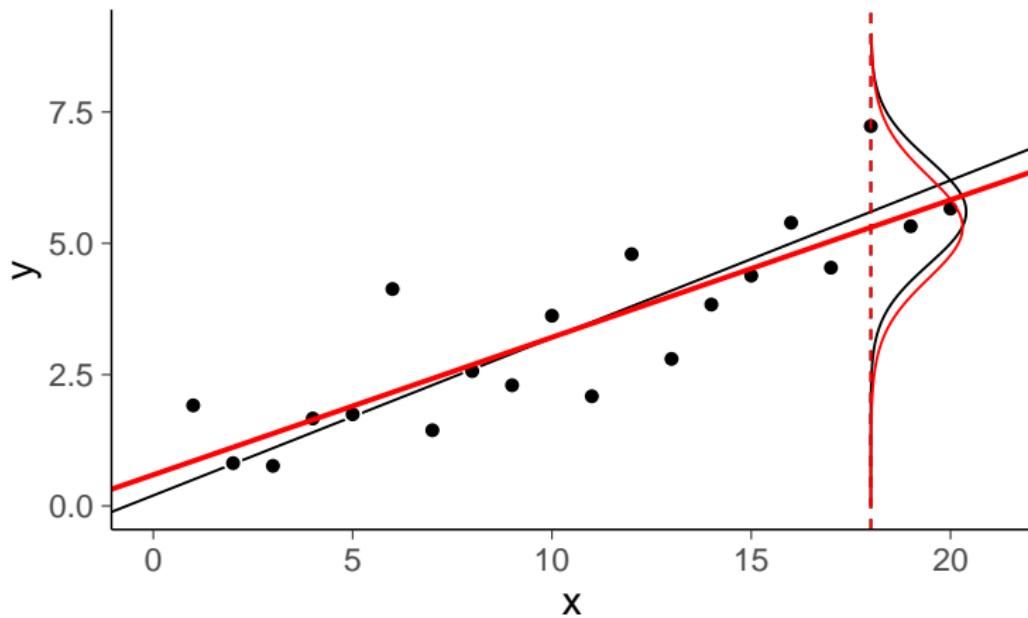
Posterior mean



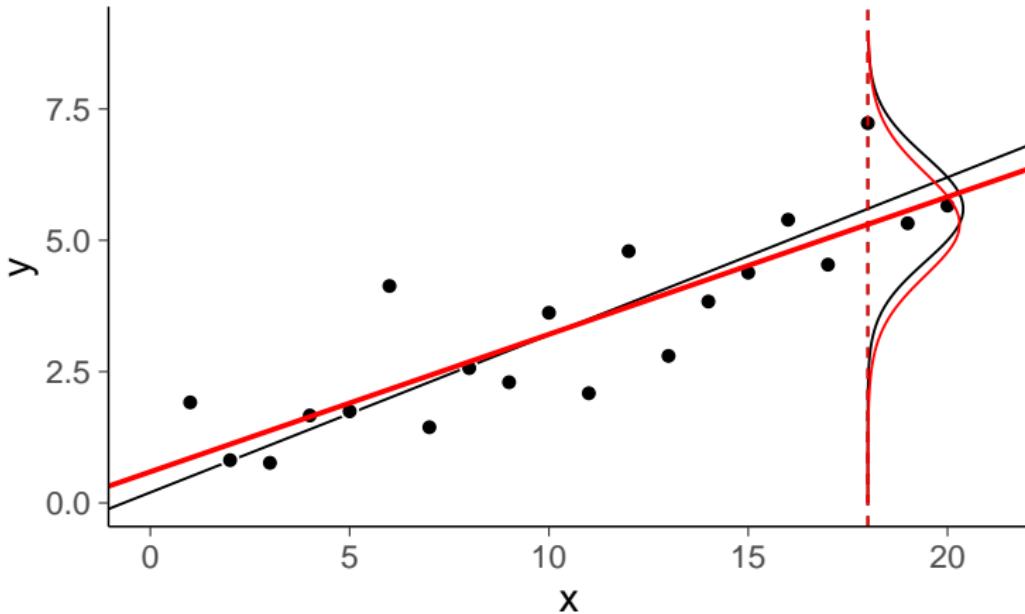
Posterior draws



Posterior predictive distribution

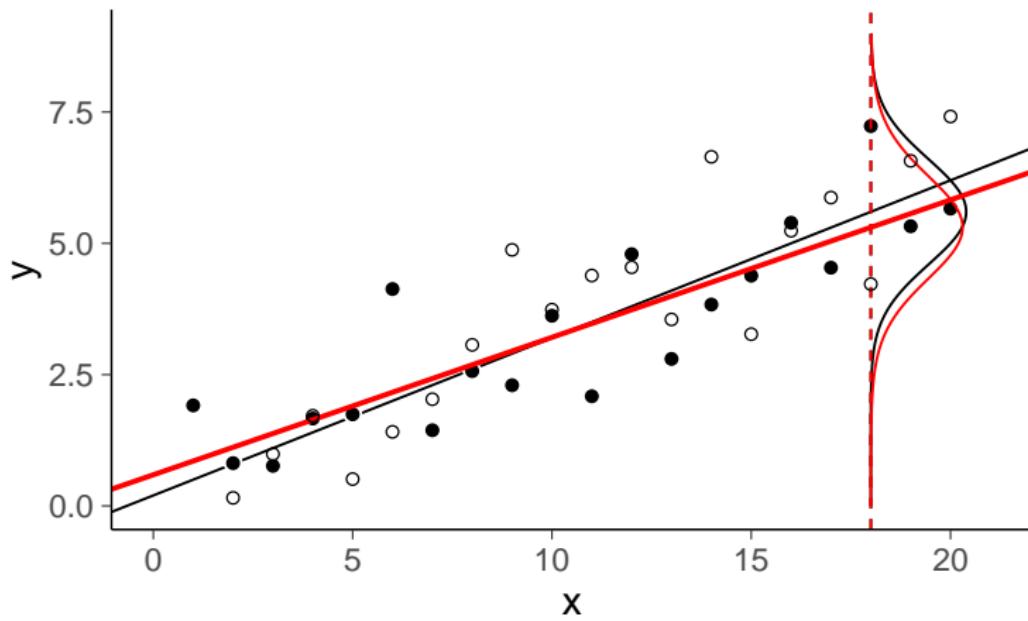


Posterior predictive distribution

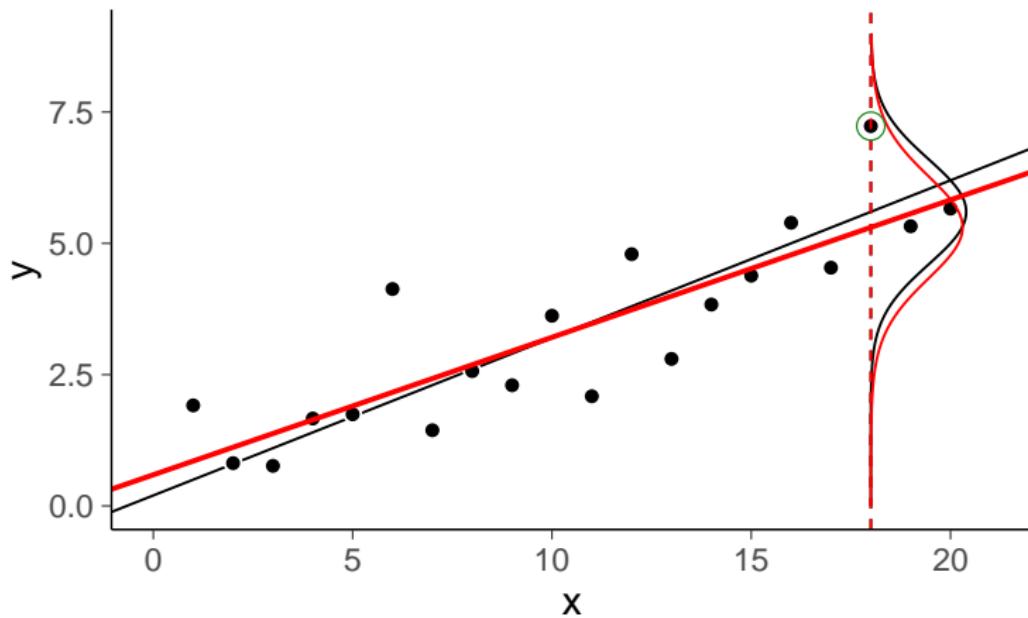


$$p(\tilde{y} \mid \tilde{x} = 18, x, y) = \int p(\tilde{y} \mid \tilde{x} = 18, \theta) p(\theta \mid x, y) d\theta$$

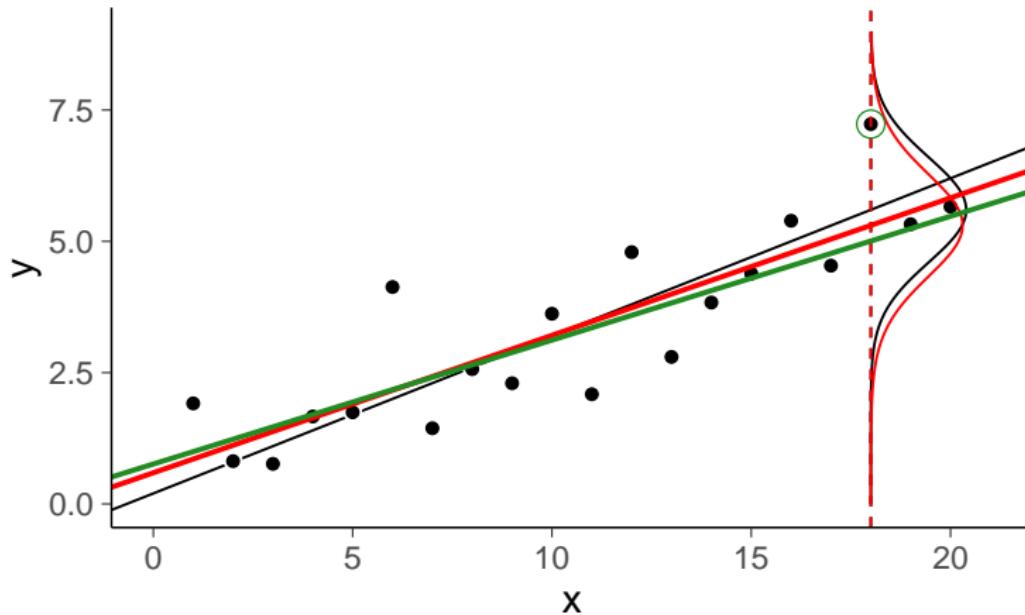
New data



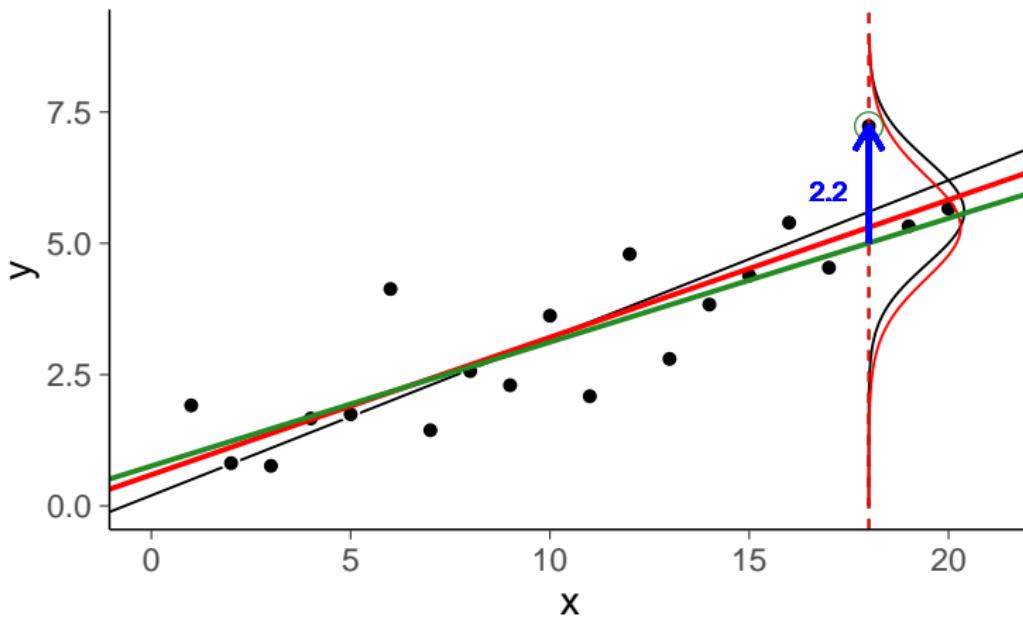
Posterior predictive distribution



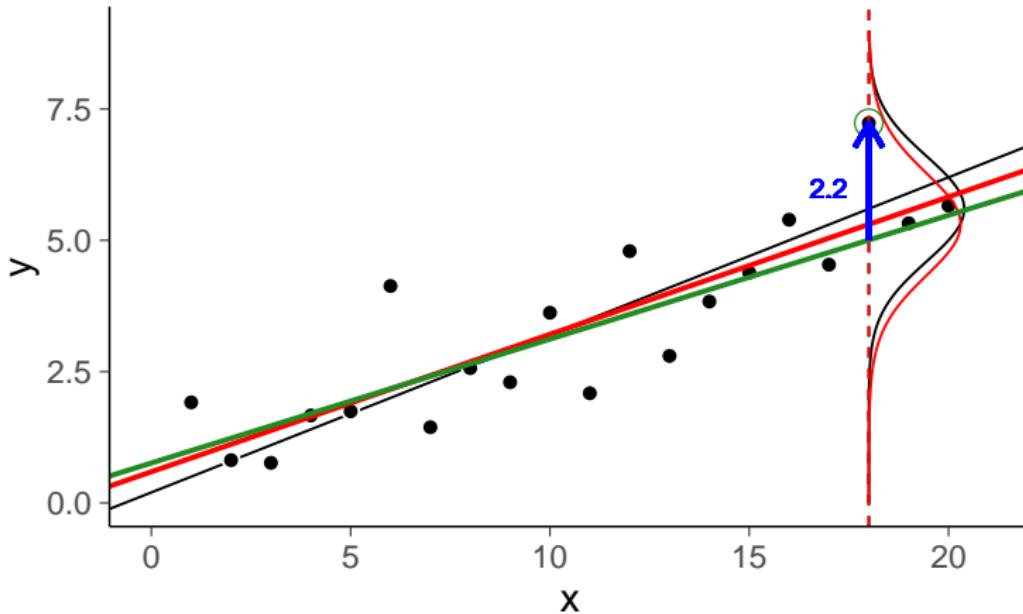
Leave-one-out mean



Leave-one-out residual

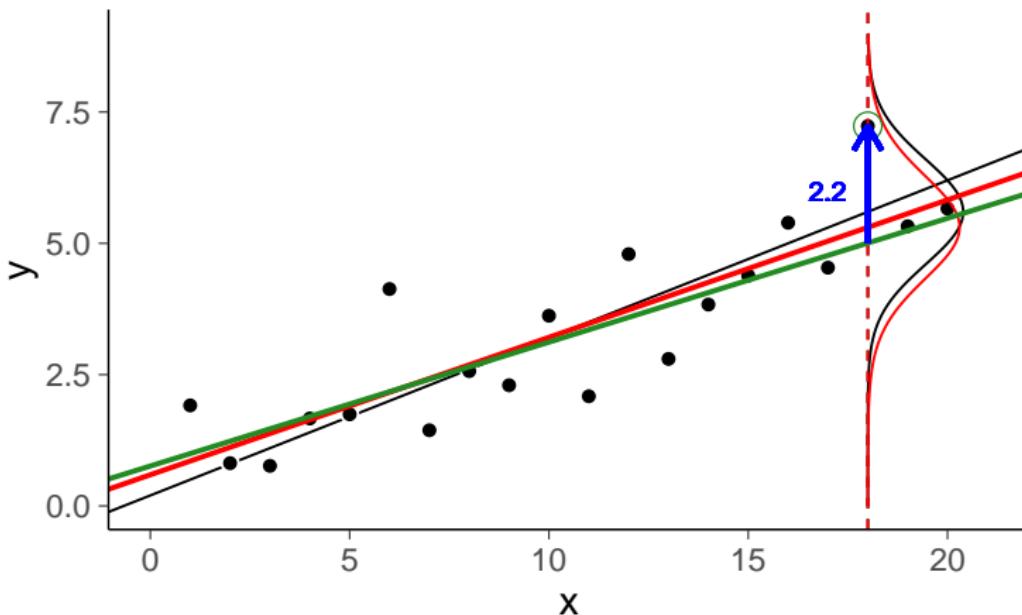


Leave-one-out residual



$$y_{18} - E[p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18})]$$

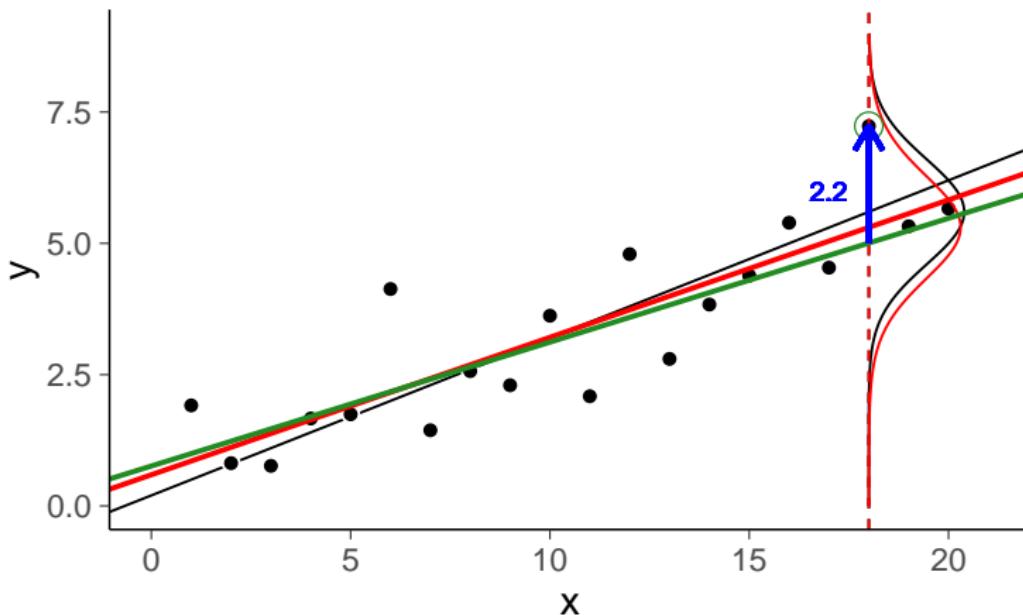
Leave-one-out residual



$$y_{18} - E[p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18})]$$

Can be used to compute, e.g., RMSE, R^2 , 90% error

Leave-one-out residual

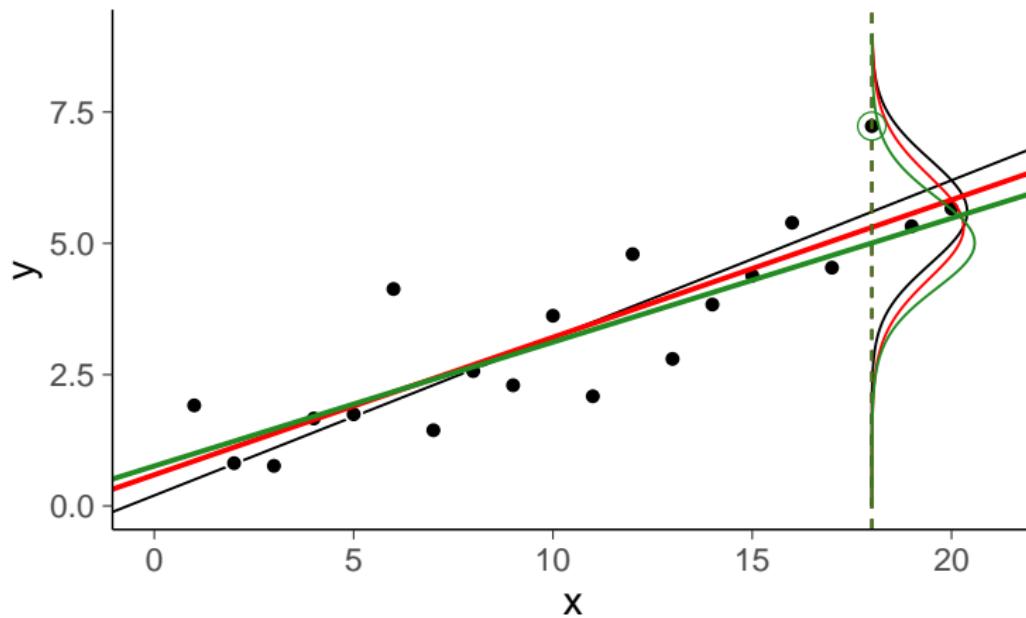


$$y_{18} - E[p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18})]$$

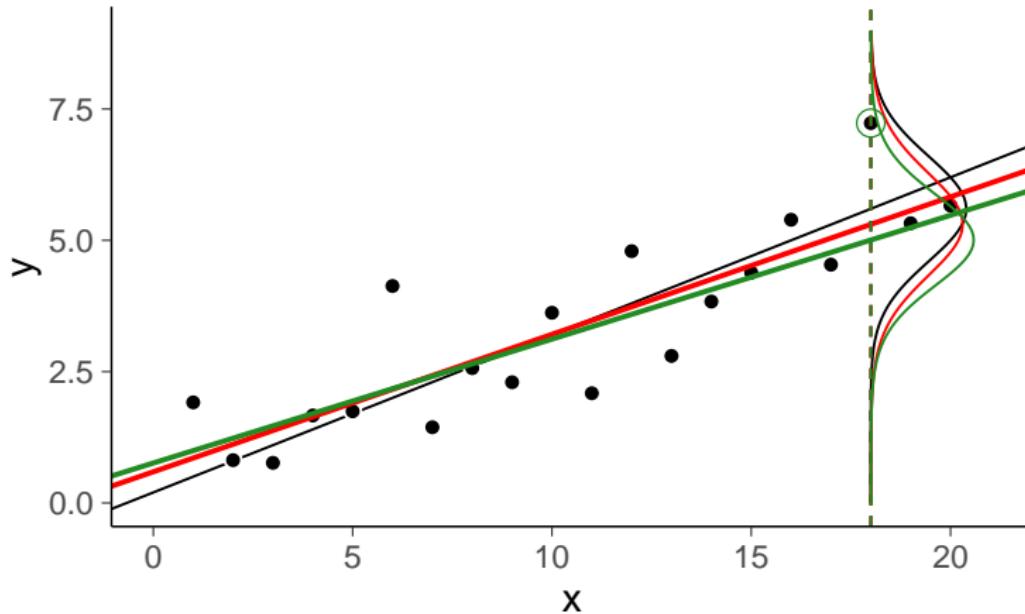
Can be used to compute, e.g., RMSE, R^2 , 90% error

See LOO- R^2 at avehtari.github.io/bayes_R2/bayes_R2.html

Leave-one-out predictive distribution

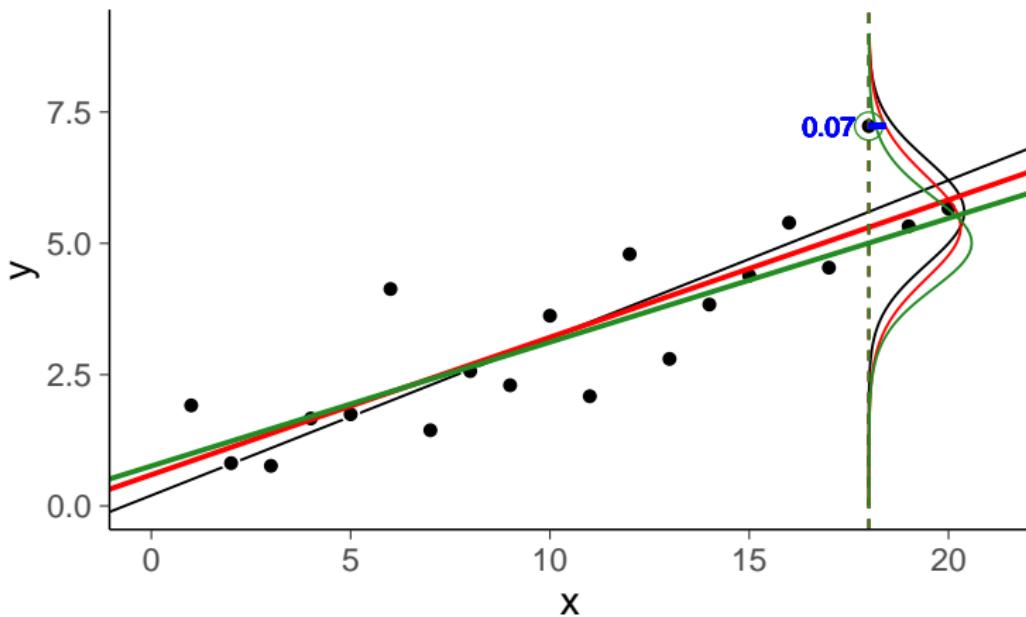


Leave-one-out predictive distribution

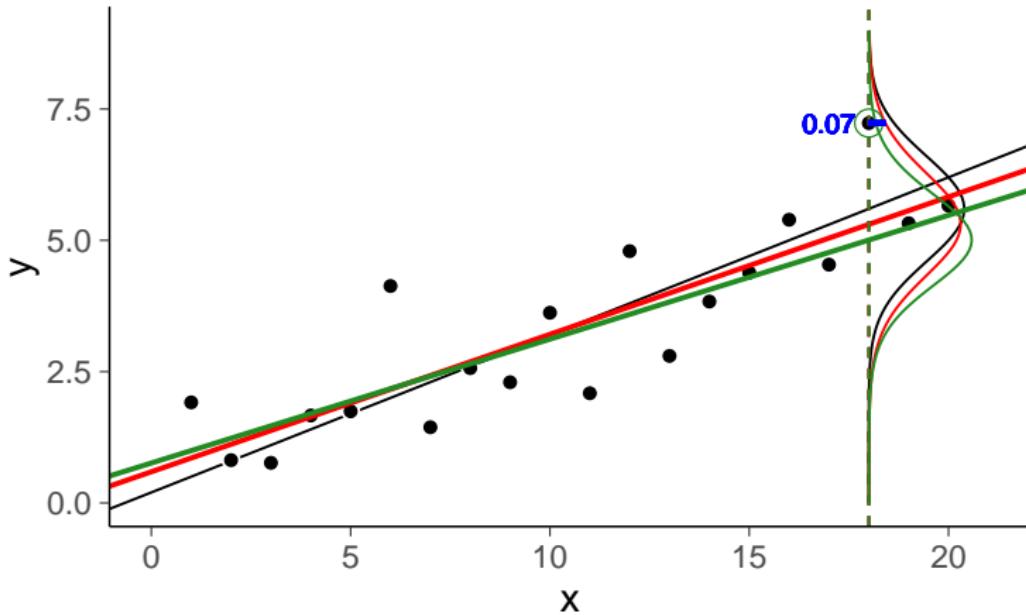


$$p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18}) = \int p(\tilde{y} \mid \tilde{x} = 18, \theta) p(\theta \mid x_{-18}, y_{-18}) d\theta$$

Posterior predictive density

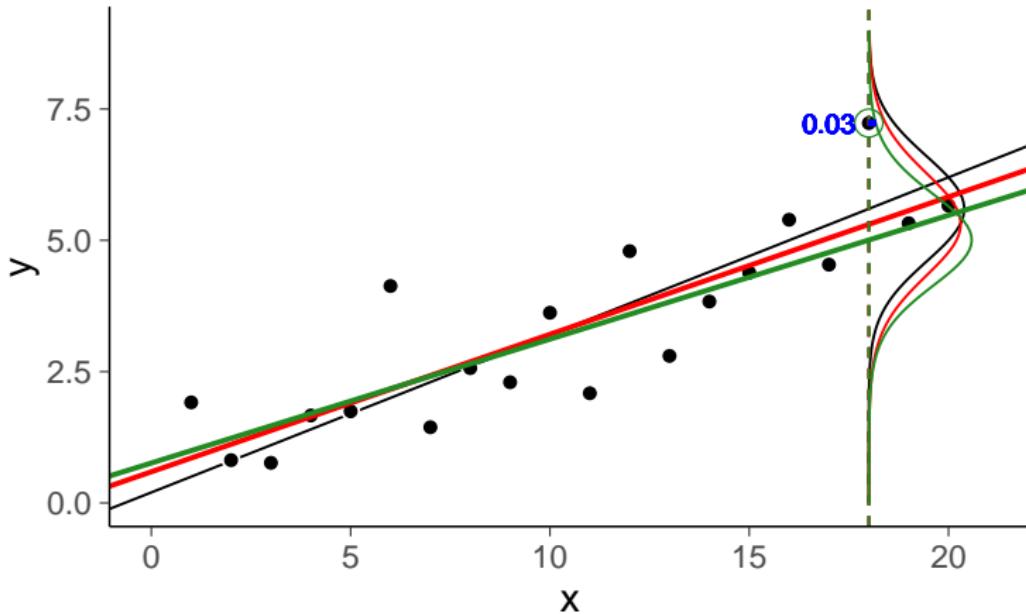


Posterior predictive density



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

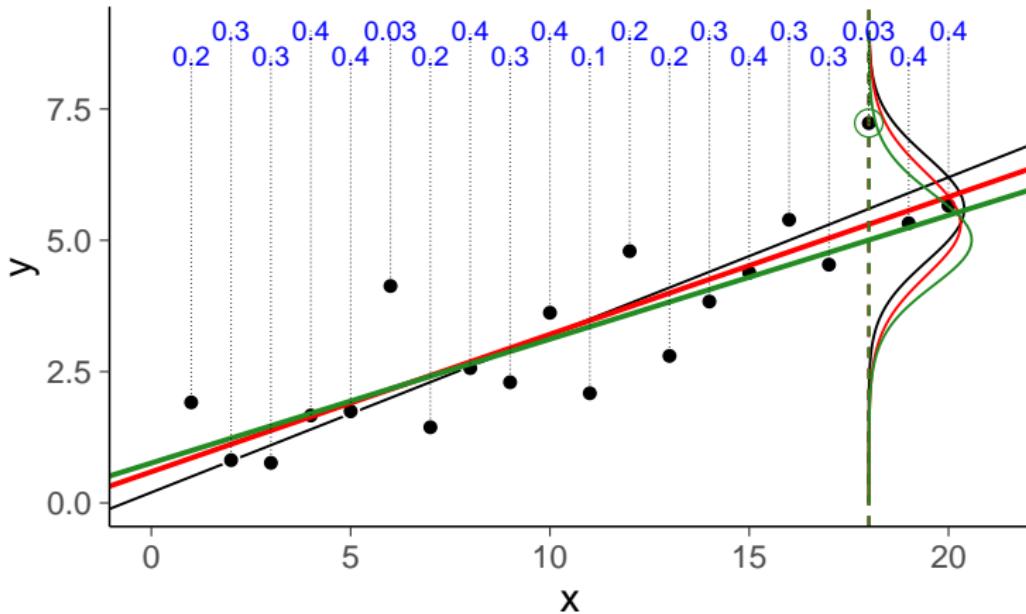
Leave-one-out predictive density



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

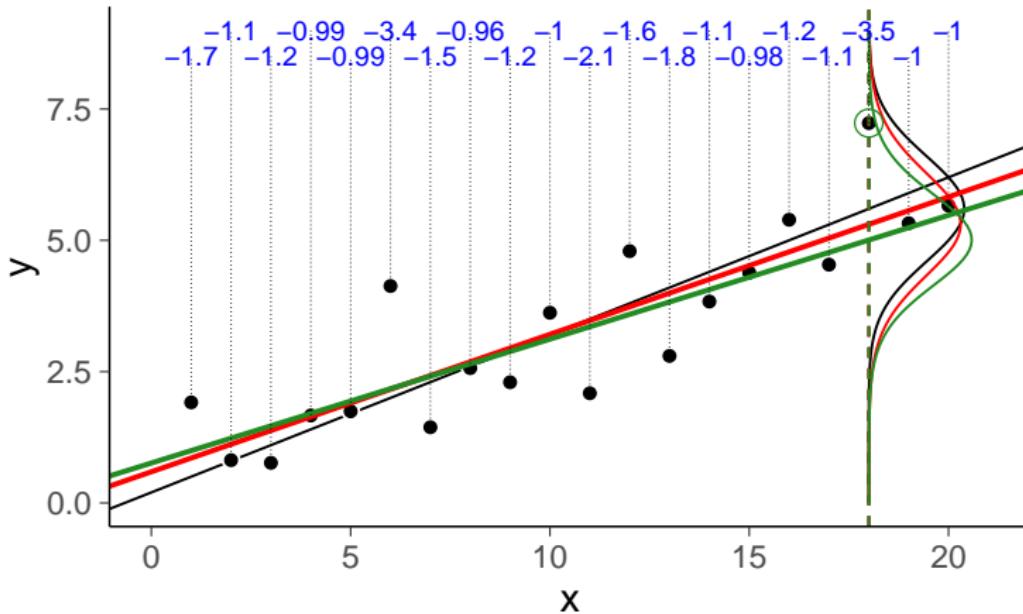
$$p(\tilde{y} = y_{18} \mid \tilde{x} = 18, x_{-18}, y_{-18}) \approx 0.03$$

Leave-one-out predictive densities



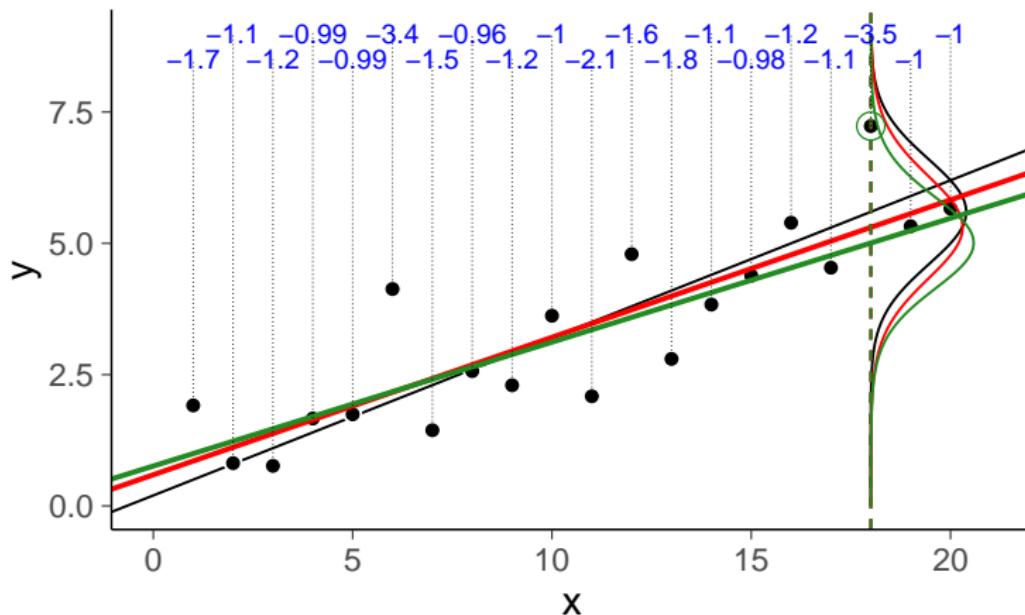
$$p(y_i | x_i, x_{-i}, y_{-i}), \quad i = 1, \dots, 20$$

Leave-one-out log predictive densities



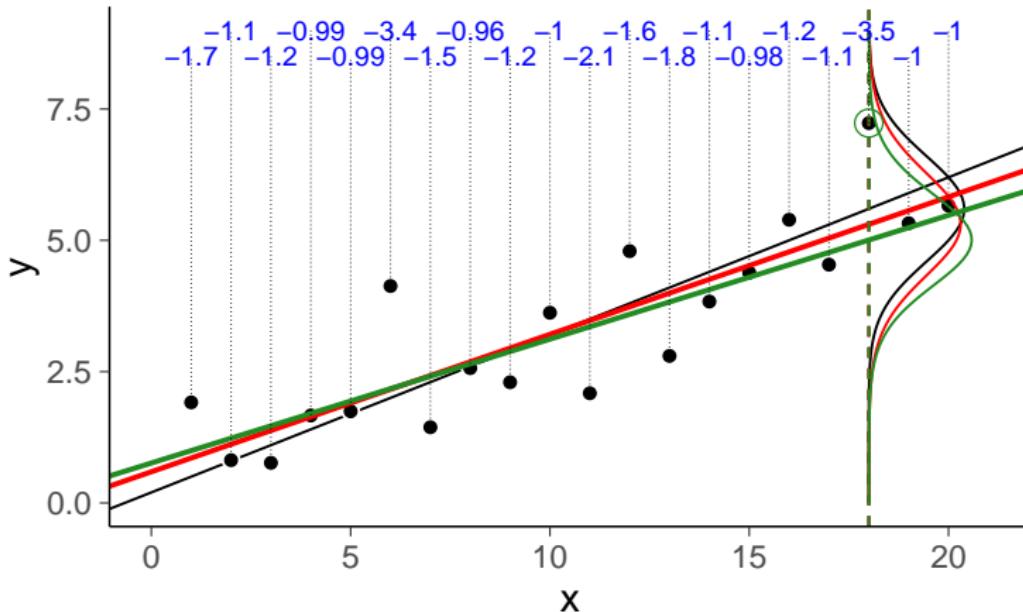
$$\log p(y_i | x_i, x_{-i}, y_{-i}), \quad i = 1, \dots, 20$$

Leave-one-out log predictive densities



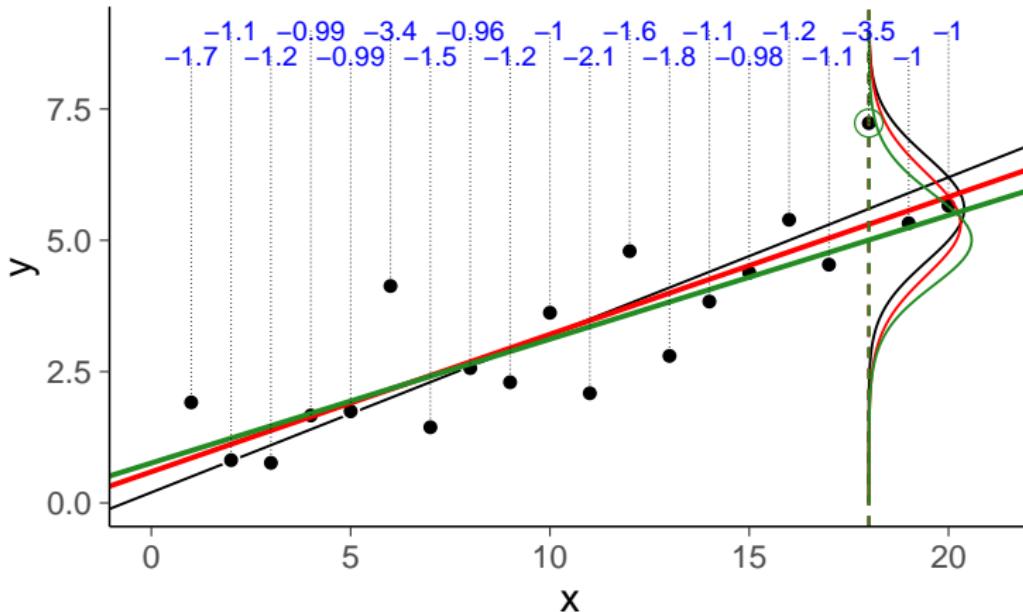
$$\sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

Leave-one-out log predictive densities



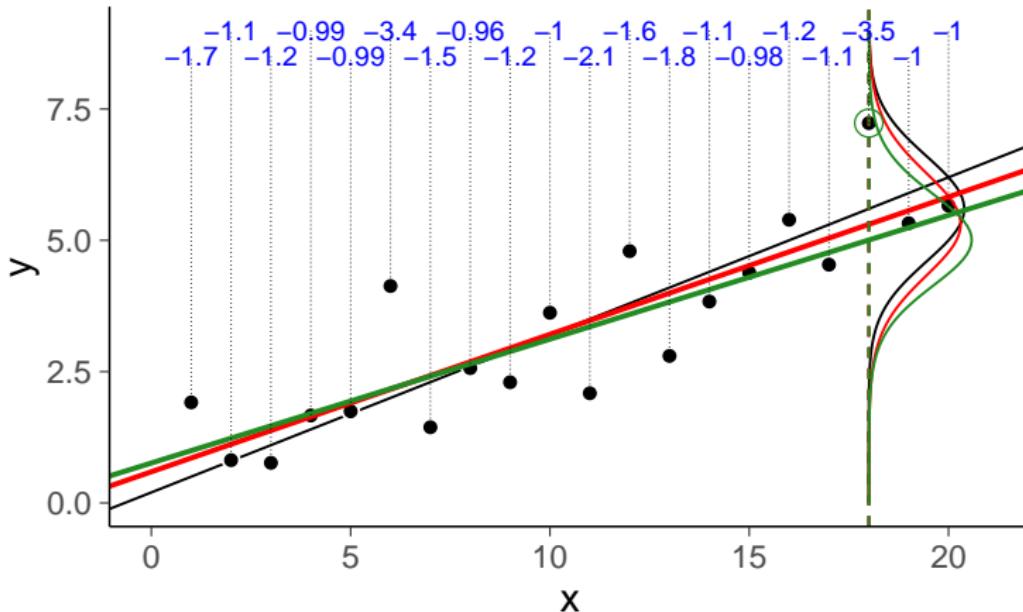
$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

Leave-one-out log predictive densities



$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$
an estimate of log posterior pred. density for new data

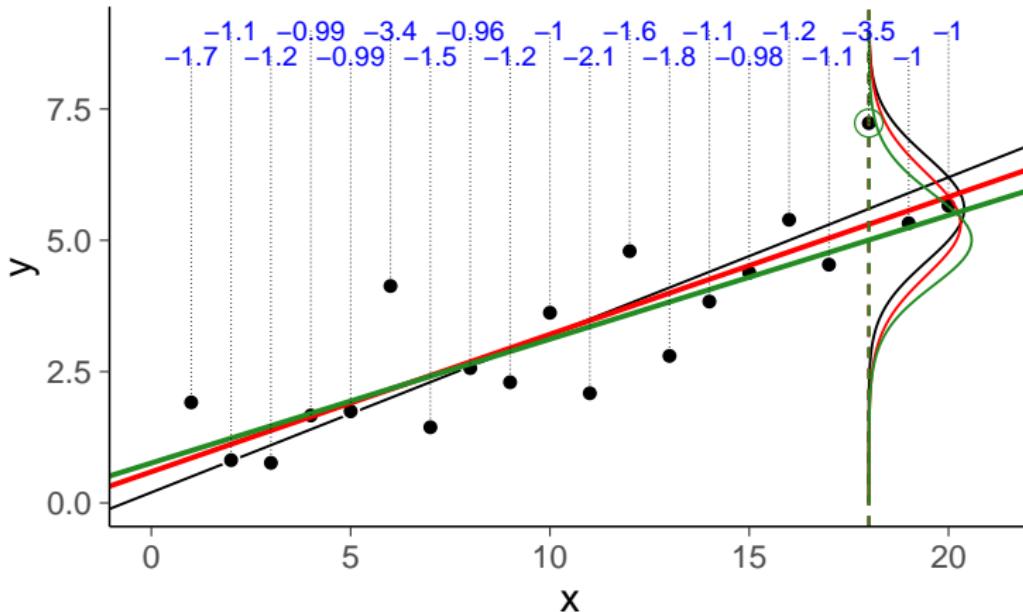
Leave-one-out log predictive densities



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

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

Leave-one-out log predictive densities

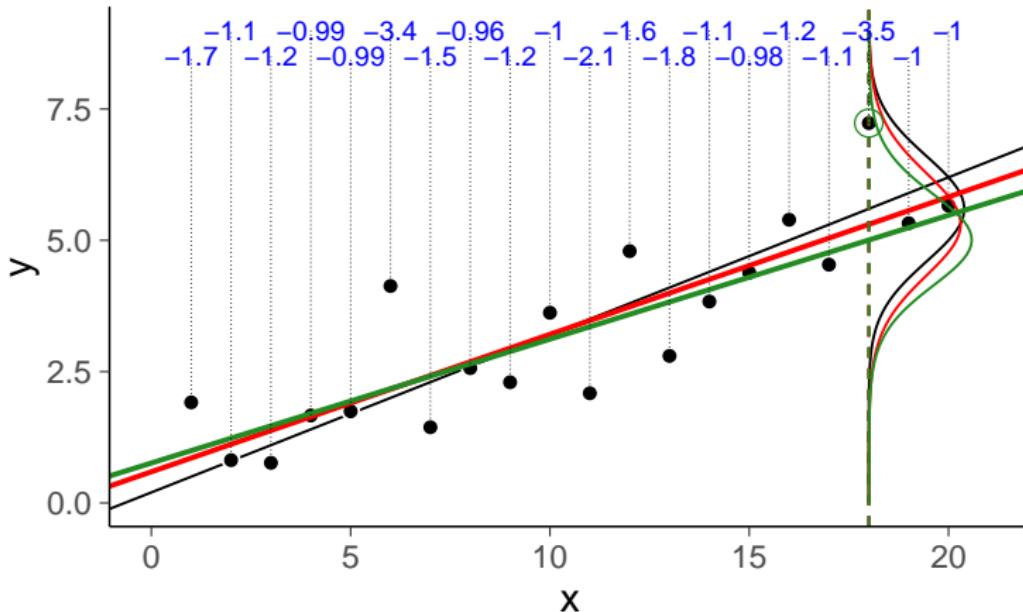


$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

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

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

Leave-one-out log predictive densities

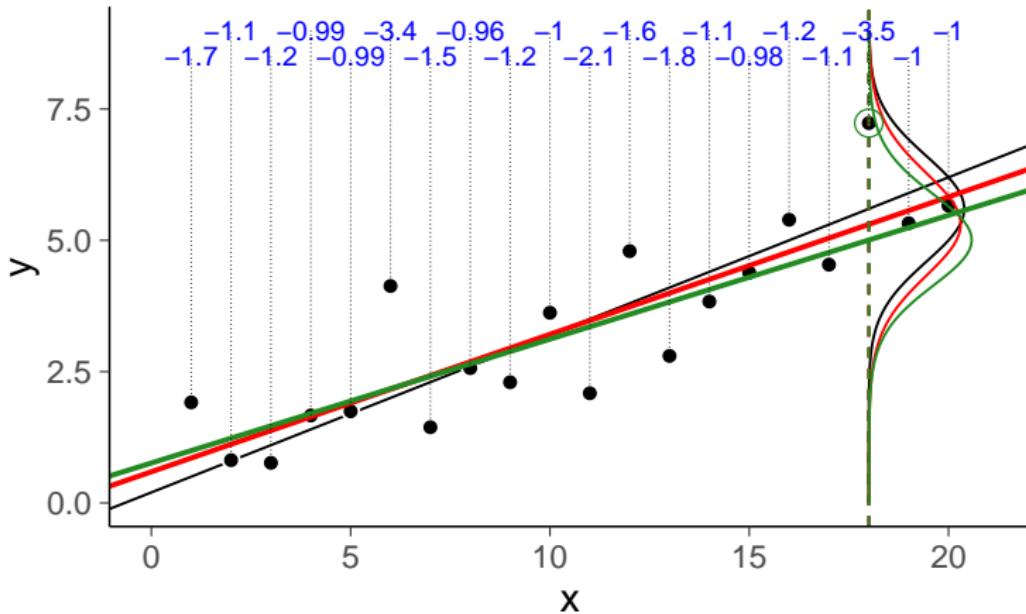


$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

$$p_{\text{loo}} = \text{lpd} - \text{elpd_loo} \approx 2.7$$

asymptotically approaches p in case of regular faithful model

Leave-one-out log predictive densities



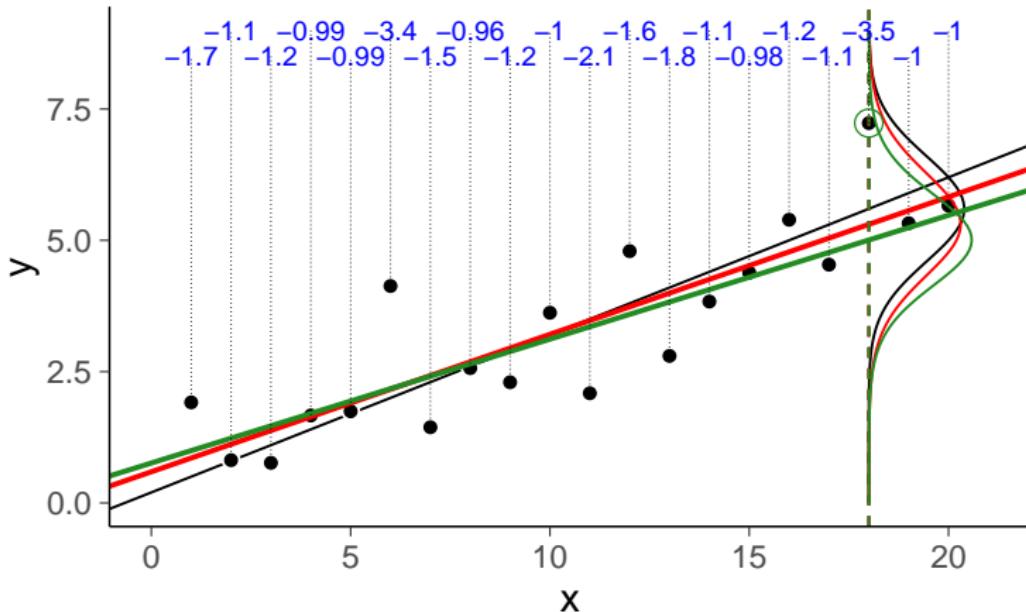
$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$

$$p_{\text{loo}} = \text{lpd} - \text{elpd_loo} \approx 2.7$$

asymptotically approaches p in case of regular faithful model

see Vehtari, Gelman & Gabry (2017a) and Vehtari & Ojanen (2012) for more

Leave-one-out log predictive densities



$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i | x_i, x_{-i}, y_{-i}) \approx -29.5$$
$$\text{SE} = \text{sd}(\log p(y_i | x_i, x_{-i}, y_{-i})) \cdot \sqrt{20} \approx 3.3$$

see Vehtari, Gelman & Gabry (2017a) and Vehtari & Ojanen (2012) for more

loo package

Computed from 4000 by 20 log-likelihood matrix

	Estimate	SE
elpd_loo	-29.5	3.3
p_loo	2.7	1.0

MCSE of elpd_loo is 0.1.

MCSE and ESS estimates assume MCMC draws (r_{eff} in [0.6, 1.0]).

All Pareto k estimates are good ($k < 0.7$).

See `help('pareto-k-diagnostic')` for details.

Helicopter flight time – elpd

Computed from 4000 by 150 log-likelihood matrix.

	Estimate	SE
elpd_loo	-79.2	9.9
p_loo	8.3	1.4

MCSE of elpd_loo is 0.1.

MCSE and ESS estimates assume MCMC draws (r_{eff} in [0.6, 1.2]).

All Pareto k estimates are good ($k < 0.7$).

See `help('pareto-k-diagnostic')` for details.

Helicopter flight time – R^2

R^2 is the proportion of variance explained by the model

```
> bayes_R2(fit) |> round(digits=2)
```

	Estimate	Est.Error	Q2.5	Q97.5
--	----------	-----------	------	-------

R2	0.51	0.04	0.42	0.58
----	------	------	------	------

```
> loo_R2(fit) |> round(digits=2)
```

	Estimate	Est.Error	Q2.5	Q97.5
--	----------	-----------	------	-------

R2	0.47	0.06	0.35	0.57
----	------	------	------	------

Student retention – R^2

R^2 is the proportion of variance explained by the model

```
> bayes_R2(fit6) |> round(digits=2)
```

	Estimate	Est.Error	Q2.5	Q97.5
--	----------	-----------	------	-------

R2	0.98	0	0.97	0.98
----	------	---	------	------

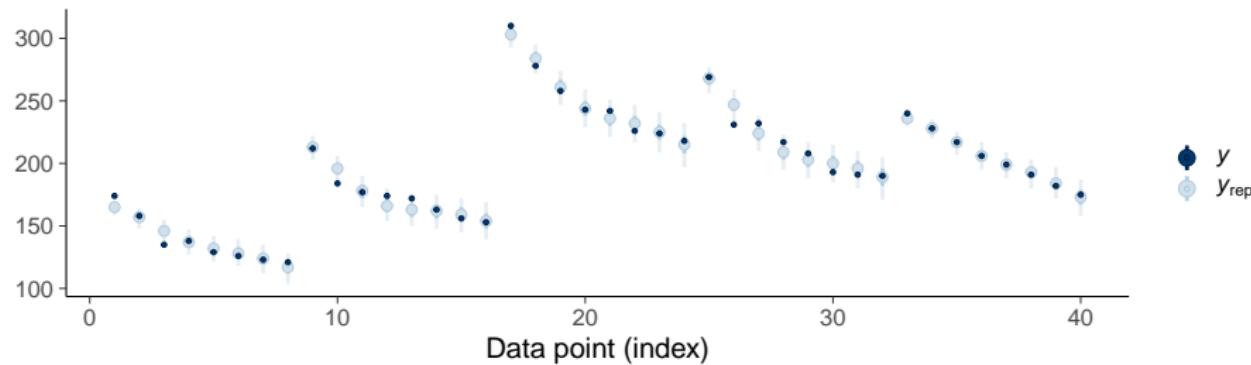
```
> loo_R2(fit6) |> round(digits=2)
```

	Estimate	Est.Error	Q2.5	Q97.5
--	----------	-----------	------	-------

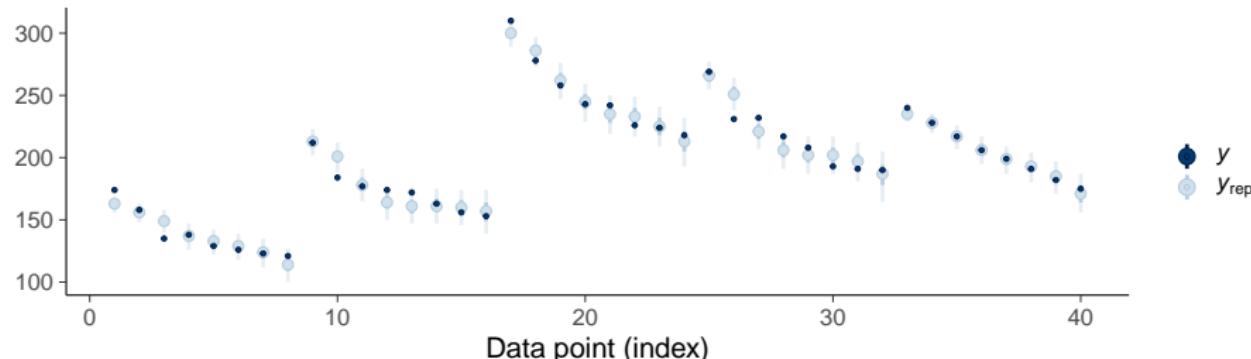
R2	0.97	0.01	0.95	0.98
----	------	------	------	------

Student retention

Posterior predictive intervals



LOO predictive intervals



Student retention – R^2

Latent hierarchical linear vs. latent hierarchical linear + spline

```
> loo_R2(fit4) |> round(digits=2)
```

	Estimate	Est.Error	Q2.5	Q97.5
--	----------	-----------	------	-------

R2	0.92	0.02	0.88	0.95
----	------	------	------	------

```
> loo_R2(fit6) |> round(digits=2)
```

	Estimate	Est.Error	Q2.5	Q97.5
--	----------	-----------	------	-------

R2	0.97	0.01	0.95	0.98
----	------	------	------	------

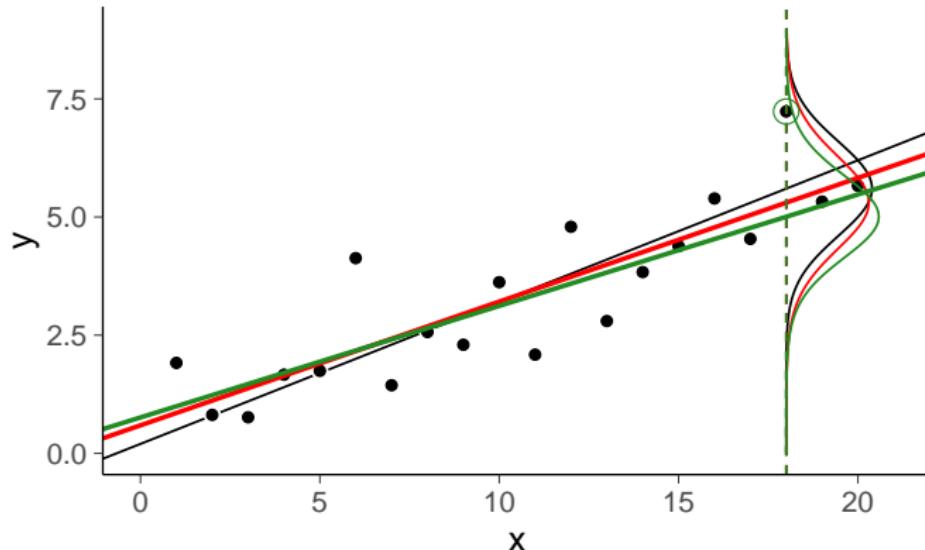
Student retention – elpd (log score)

Latent hierarchical linear vs. latent hierarchical linear + spline

```
> loo_compare(fit4, fit6)
      elpd_diff se_diff
fit6    0.0      0.0
fit4 -43.2     14.4
```

More about this soon

LOO-PIT predictive checking



- LOO probability integral transform (LOO-PIT)

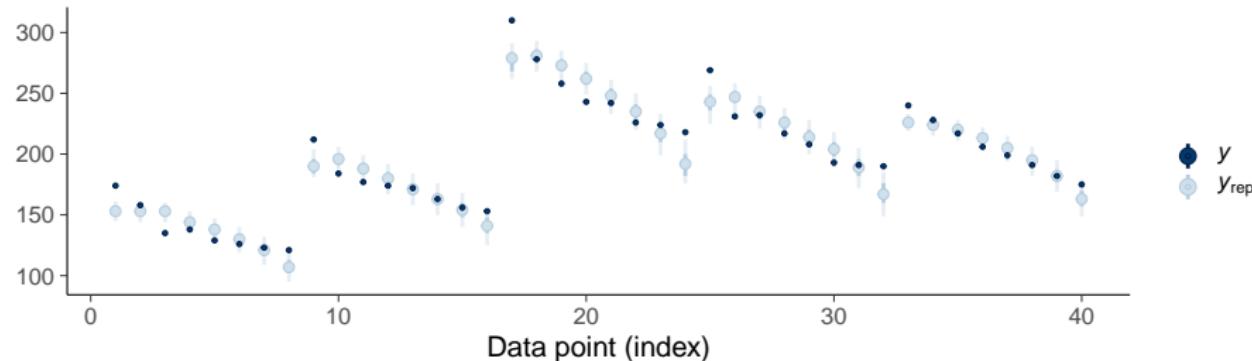
$$p_i = p(y_i^{\text{rep}} \leq y_i | y_{-i})$$

- If $p(\tilde{y}_i | y_{-i})$ is well calibrated, distribution of p_i 's would be uniform between 0 and 1

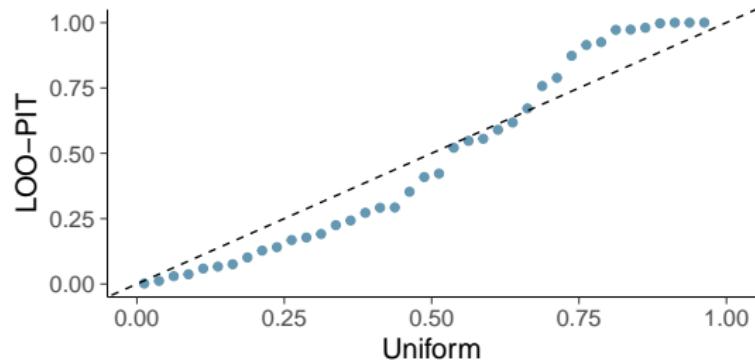
Student retention – LOO-PIT checking

```
pp_check(fit, type = "loo_pit_qq", ndraws=4000)
```

Latent hierarchical linear – LOO predictive intervals



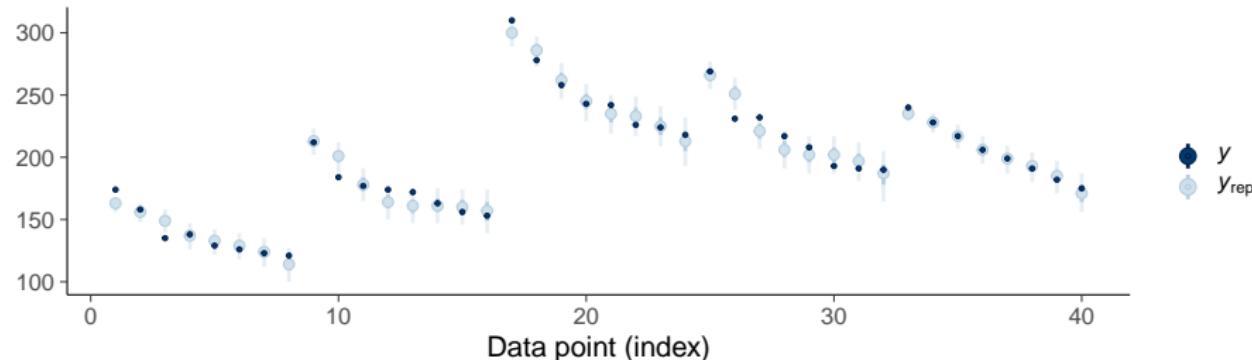
LOO-PIT check



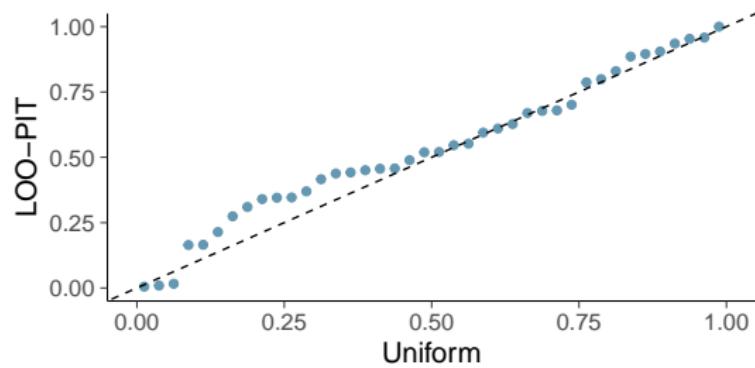
Student retention – LOO-PIT checking

```
pp_check(fit, type = "loo_pit_qq", ndraws=4000)
```

Latent hierarchical linear + spline – LOO predictive intervals/



LOO-PIT check



Brute-force LOO

- Re-run MCMC n times to sample from $p(\theta \mid x_{-i}, y_{-i})$
 - can take a lot of time

Brute-force LOO

- Re-run MCMC n times to sample from $p(\theta \mid x_{-i}, y_{-i})$
 - can take a lot of time
 - or high parallelization
- Cooper, Vehtari, Forbes, Kennedy, and Simpson (2023).
Bayesian cross-validation by parallel Markov chain Monte Carlo. *Statistics and Computing*, **34**:119.
doi:10.1007/s11222-024-10404-w.

Fast cross-validation

- Pareto smoothed importance sampling LOO (PSIS-LOO)
- K -fold cross-validation

see Vehtari, Gelman & Gabry (2017a) and mc-stan.org/loo/

Importance sampling leave-one-out cross-validation

- We want to compute

$$p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$$

Importance sampling leave-one-out cross-validation

- We want to compute

$$p(y_i | x_i, x_{-i}, y_{-i}) = \int p(y_i | x_i, \theta) p(\theta | x_{-i}, y_{-i}) d\theta$$

- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta | x, y)$
- Target distribution is LOO-posterior $p(\theta | x_{-i}, y_{-i})$

Importance sampling leave-one-out cross-validation

- We want to compute

$$p(y_i | x_i, x_{-i}, y_{-i}) = \int p(y_i | x_i, \theta) p(\theta | x_{-i}, y_{-i}) d\theta$$

- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta | x, y)$
- Target distribution is LOO-posterior $p(\theta | x_{-i}, y_{-i})$
- Importance ratio

$$w_i^{(s)} = \frac{p(\theta^{(s)} | x_{-i}, y_{-i})}{p(\theta^{(s)} | x, y)} \propto \frac{1}{p(y_i | x_i, \theta^{(s)})}$$

Importance sampling leave-one-out cross-validation

- We want to compute

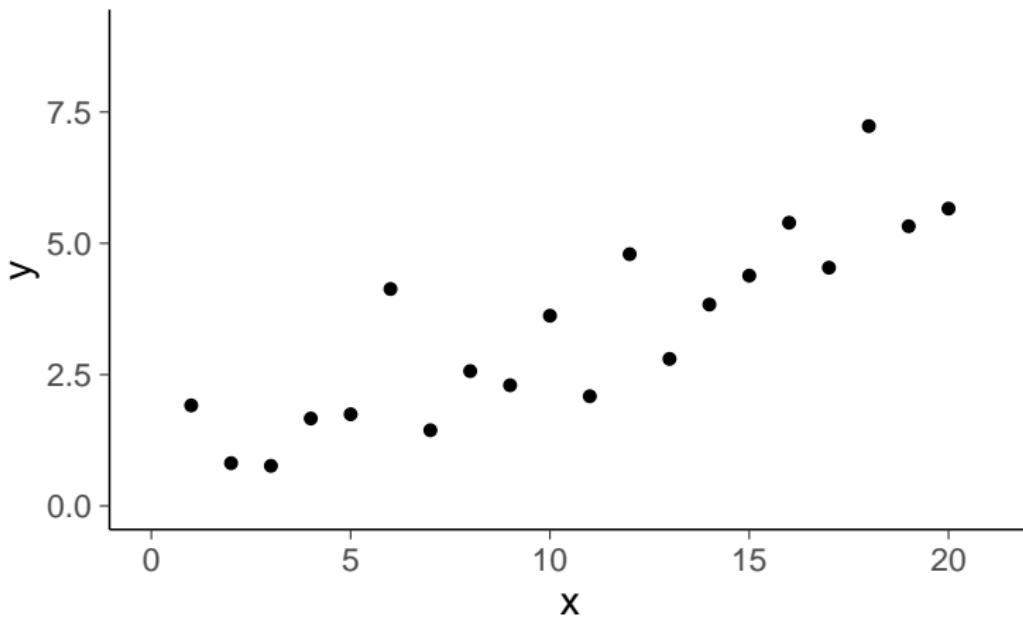
$$p(y_i | x_i, x_{-i}, y_{-i}) = \int p(y_i | x_i, \theta) p(\theta | x_{-i}, y_{-i}) d\theta$$

- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta | x, y)$
- Target distribution is LOO-posterior $p(\theta | x_{-i}, y_{-i})$
- Importance ratio

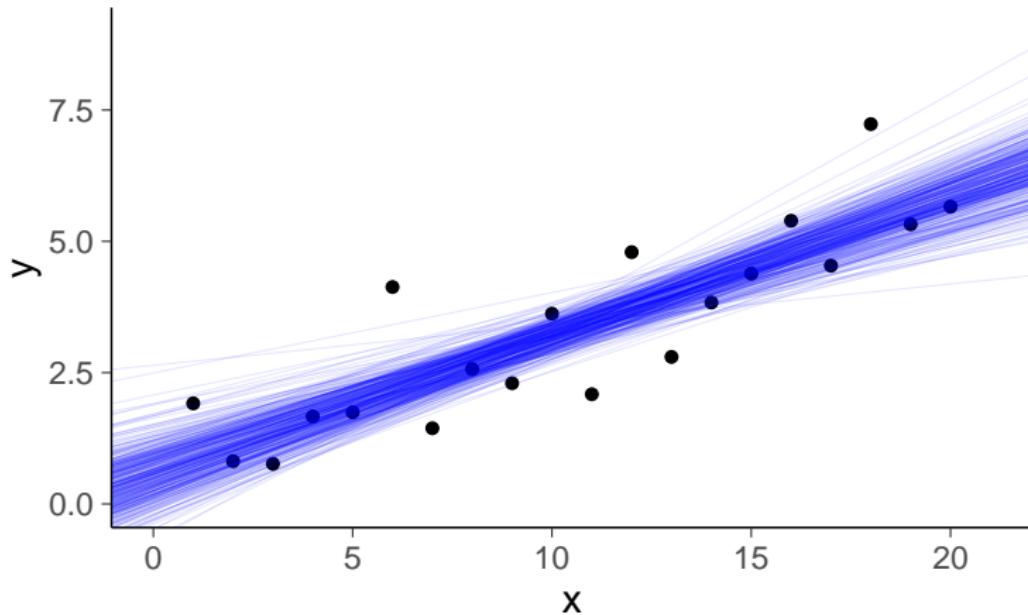
$$w_i^{(s)} = \frac{p(\theta^{(s)} | x_{-i}, y_{-i})}{p(\theta^{(s)} | x, y)} \propto \frac{1}{p(y_i | x_i, \theta^{(s)})}$$

$$\tilde{w}_i^{(s)} = \frac{w_i^{(s)}}{\sum_{s'=1}^S w_i^{(s')}}$$

Data

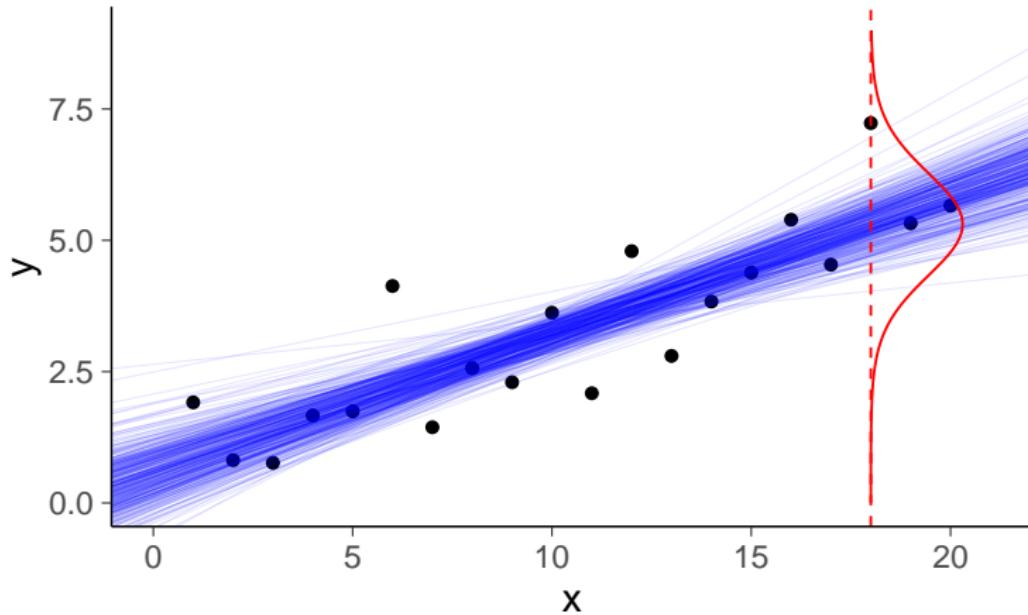


Posterior draws



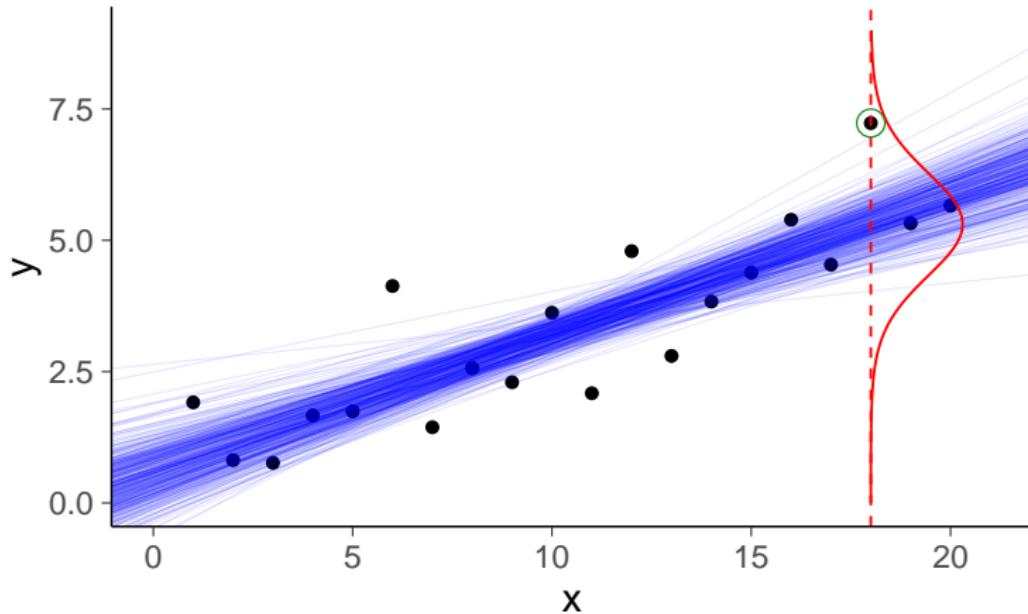
$$\theta^{(s)} \sim p(\theta | x, y)$$

Posterior predictive distribution



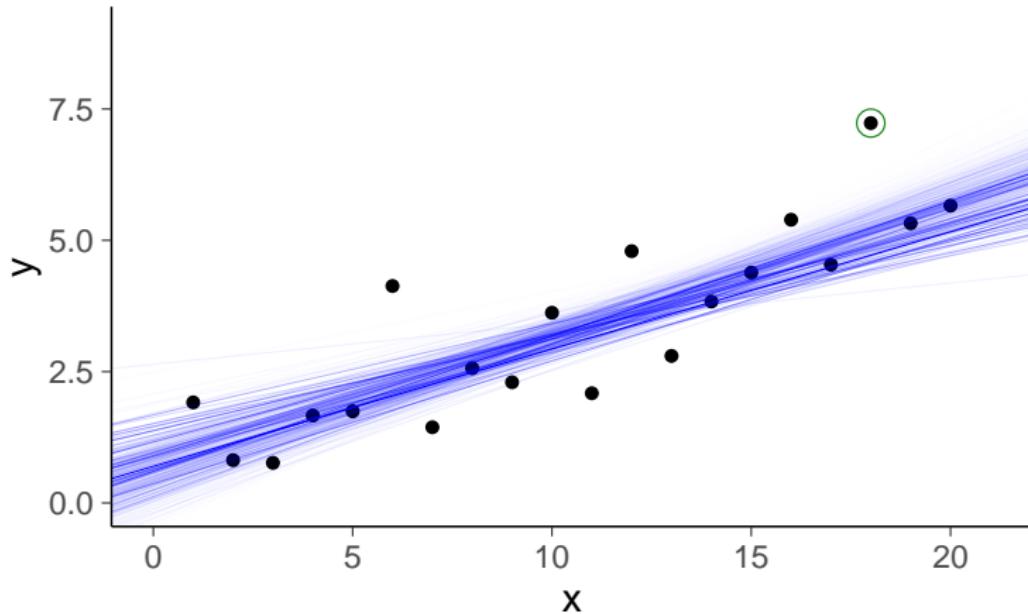
$$\theta^{(s)} \sim p(\theta | x, y), \quad p(\tilde{y} | \tilde{x}, x, y) \approx \frac{1}{S} \sum_{s=1}^S p(\tilde{y} | \tilde{x}, \theta^{(s)})$$

Posterior predictive distribution



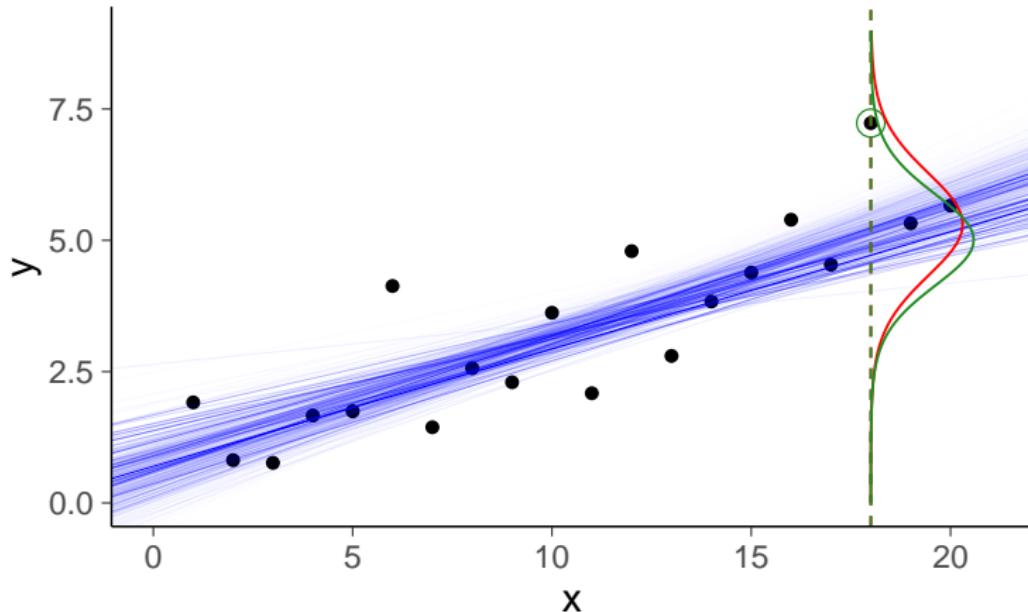
$$\theta^{(s)} \sim p(\theta | x, y), \quad p(\tilde{y} | \tilde{x}, x, y) \approx \frac{1}{S} \sum_{s=1}^S p(\tilde{y} | \tilde{x}, \theta^{(s)})$$

PSIS-LOO weighted draws



$$\theta^{(s)} \sim p(\theta \mid x, y), \quad w_i^{(s)} = p(\theta^{(s)} \mid x_{-i}, y_{-i}) / p(\theta^{(s)} \mid x, y)$$

PSIS-LOO weighted predictive distribution



$$\theta^{(s)} \sim p(\theta | x, y), \quad w_i^{(s)} = p(\theta^{(s)} | x_{-i}, y_{-i}) / p(\theta^{(s)} | x, y)$$

$$p(y_i | x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^S [\tilde{w}_i^{(s)} p(y_i | x_i, \theta^{(s)})]$$

Pareto smoothed importance sampling LOO

- $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal $p(\theta \mid x, y)$ and target $p(\theta \mid x_{-i}, y_{-i})$
- Importance ratio

$$w_i^{(s)} = \frac{p(\theta^{(s)} \mid x_{-i}, y_{-i})}{p(\theta^{(s)} \mid x, y)} \propto \frac{1}{p(y_i \mid x_i, \theta^{(s)})}$$

$$\tilde{w}_i^{(s)} = \frac{w_i^{(s)}}{\sum_{s'=1}^S w_i^{(s')}}$$

$$p(y_i \mid x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^S \left[\tilde{w}_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]$$

Pareto smoothed importance sampling LOO

- $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal $p(\theta \mid x, y)$ and target $p(\theta \mid x_{-i}, y_{-i})$
- Importance ratio

$$w_i^{(s)} = \frac{p(\theta^{(s)} \mid x_{-i}, y_{-i})}{p(\theta^{(s)} \mid x, y)} \propto \frac{1}{p(y_i \mid x_i, \theta^{(s)})}$$

$$\tilde{w}_i^{(s)} = \frac{w_i^{(s)}}{\sum_{s'=1}^S w_i^{(s')}}$$

$$p(y_i \mid x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^S \left[\tilde{w}_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]$$

$$\approx \frac{\sum_{s=1}^S \left[w_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]}{\sum_{s'=1}^S w_i^{(s')}}$$

Pareto smoothed importance sampling LOO

- $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal $p(\theta \mid x, y)$ and target $p(\theta \mid x_{-i}, y_{-i})$
- Importance ratio

$$w_i^{(s)} = \frac{p(\theta^{(s)} \mid x_{-i}, y_{-i})}{p(\theta^{(s)} \mid x, y)} \propto \frac{1}{p(y_i \mid x_i, \theta^{(s)})}$$

$$\tilde{w}_i^{(s)} = \frac{w_i^{(s)}}{\sum_{s'=1}^S w_i^{(s')}}$$

$$p(y_i \mid x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^S \left[\tilde{w}_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]$$

$$\approx \frac{\sum_{s=1}^S \left[w_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]}{\sum_{s'=1}^S w_i^{(s')}}$$

$$\approx \frac{1}{\frac{1}{S} \sum_{s'=1}^S w_i^{(s')}}$$

Pareto smoothed importance sampling LOO

- $p(y_i | x_i, x_{-i}, y_{-i}) = \int p(y_i | x_i, \theta) p(\theta | x_{-i}, y_{-i}) d\theta$
- Proposal $p(\theta | x, y)$ and target $p(\theta | x_{-i}, y_{-i})$
- Importance ratio

$$w_i^{(s)} = \frac{p(\theta^{(s)} | x_{-i}, y_{-i})}{p(\theta^{(s)} | x, y)} \propto \frac{1}{p(y_i | x_i, \theta^{(s)})}$$

$$\tilde{w}_i^{(s)} = \frac{w_i^{(s)}}{\sum_{s'=1}^S w_i^{(s')}}$$

$$p(y_i | x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^S \left[\tilde{w}_i^{(s)} p(y_i | x_i, \theta^{(s)}) \right]$$

$$\approx \frac{\sum_{s=1}^S \left[w_i^{(s)} p(y_i | x_i, \theta^{(s)}) \right]}{\sum_{s'=1}^S w_i^{(s')}}$$

$$\approx \frac{1}{\frac{1}{S} \sum_{s'=1}^S w_i^{(s')}} = \frac{1}{\frac{1}{S} \sum_{s=1}^S \frac{1}{p(y_i | x_i, \theta^{(s)})}}$$

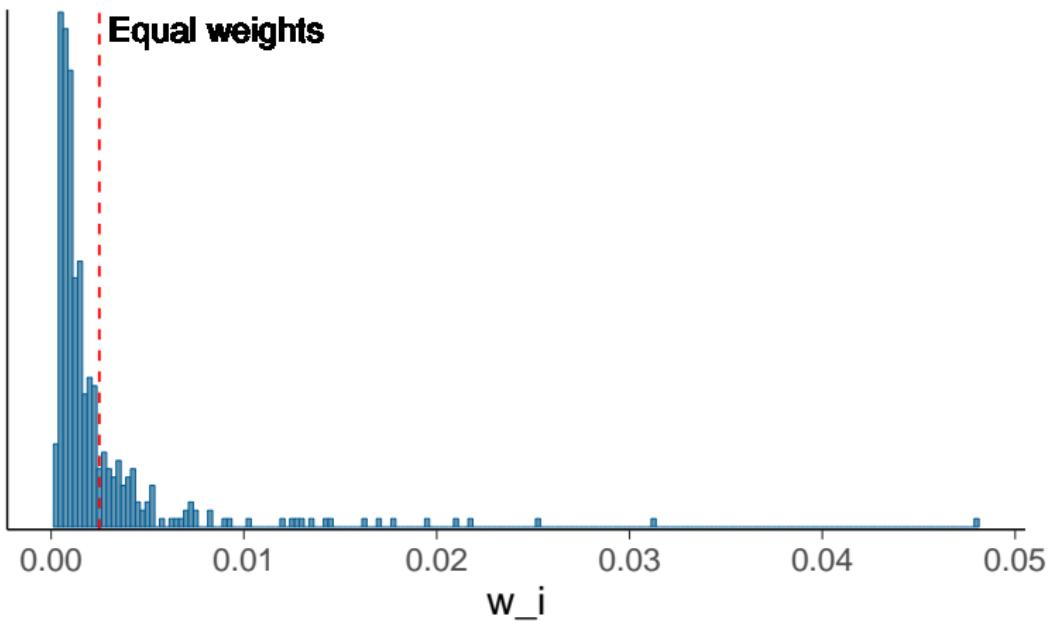
Pareto smoothed importance sampling LOO

- $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$

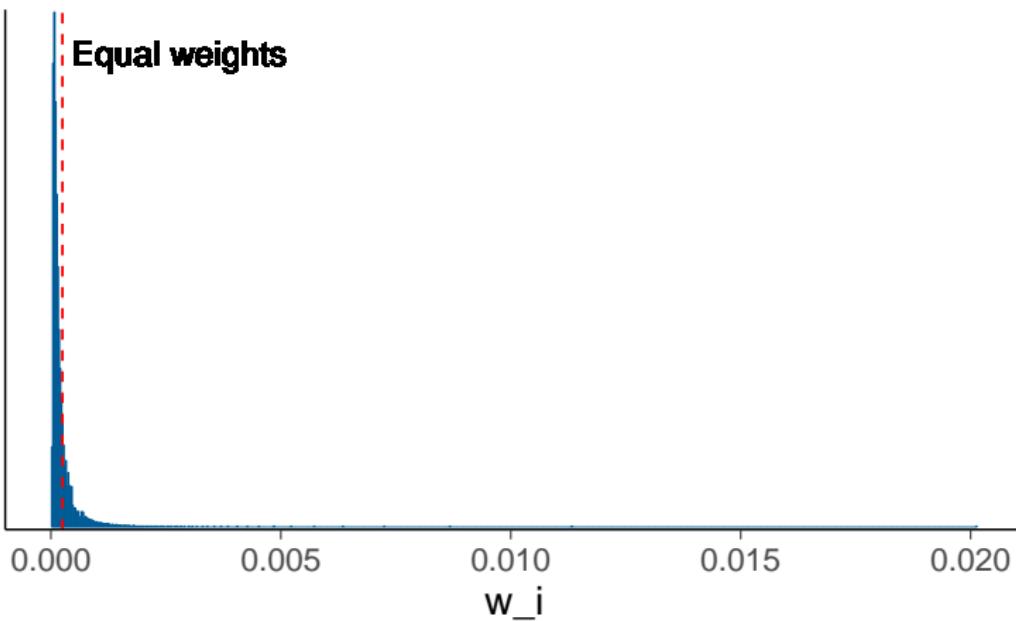
$$\begin{aligned} p(y_i \mid x_i, x_{-i}, y_{-i}) &\approx \sum_{s=1}^S \left[\tilde{w}_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right] \\ &\approx \frac{1}{\frac{1}{S} \sum_{s'=1}^S w_i^{(s')}} \end{aligned}$$

- The variability of importance weights matter
 - Pareto- k diagnostic
 - Pareto smoothed importance sampling LOO (PSIS-LOO)

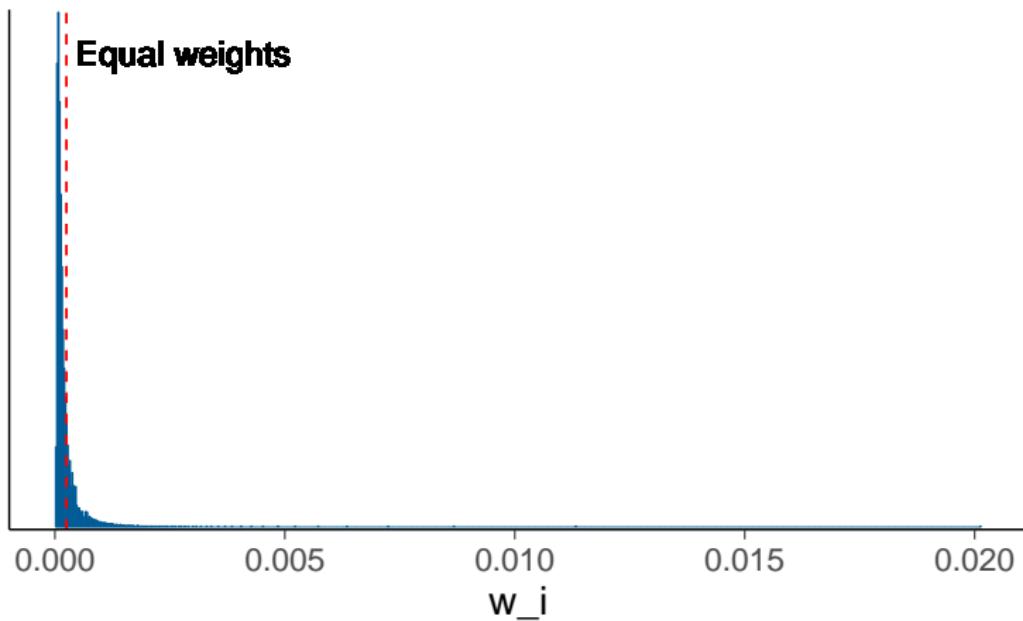
400 importance weights for leave-18th-out



4000 importance weights for leave-18th-out

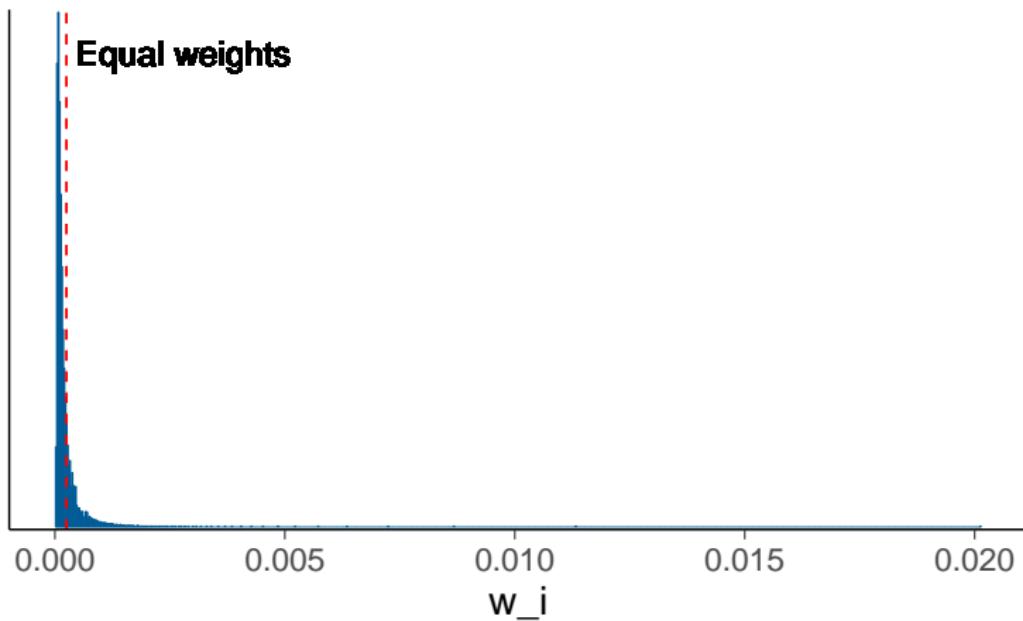


4000 importance weights for leave-18th-out



$$\text{ESS} \approx 1 / \sum_{s=1}^S (\tilde{w}^{(s)})^2 \approx 459$$

4000 importance weights for leave-18th-out



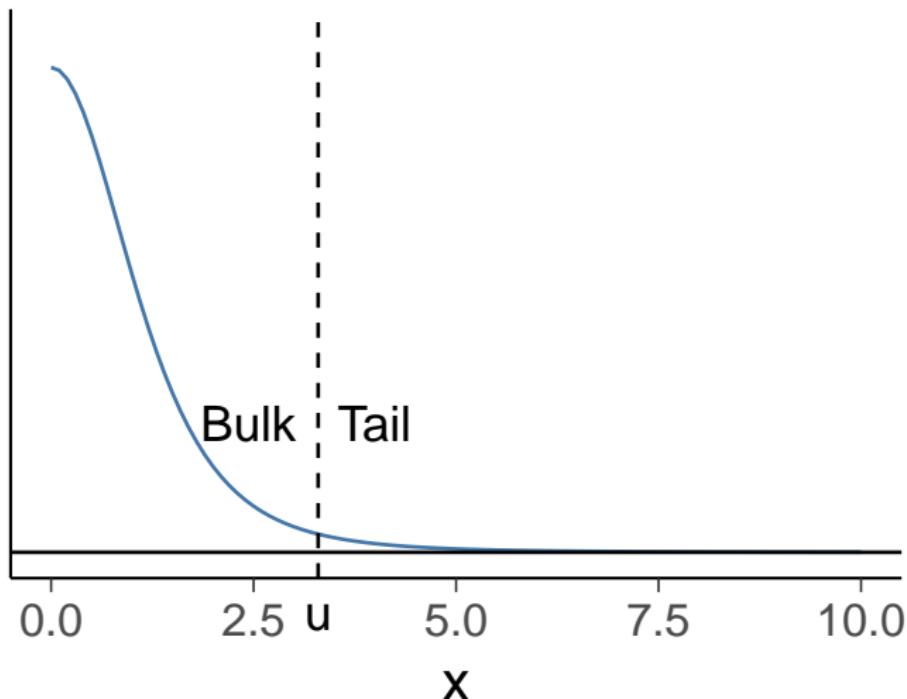
$$\text{ESS} \approx 1 / \sum_{s=1}^S (\tilde{w}^{(s)})^2 \approx 459$$

$$\text{Pareto } \hat{k} \approx 0.52$$

- Pareto \hat{k} estimates the tail shape which determines the convergence rate of PSIS. Less than 0.7 is ok.

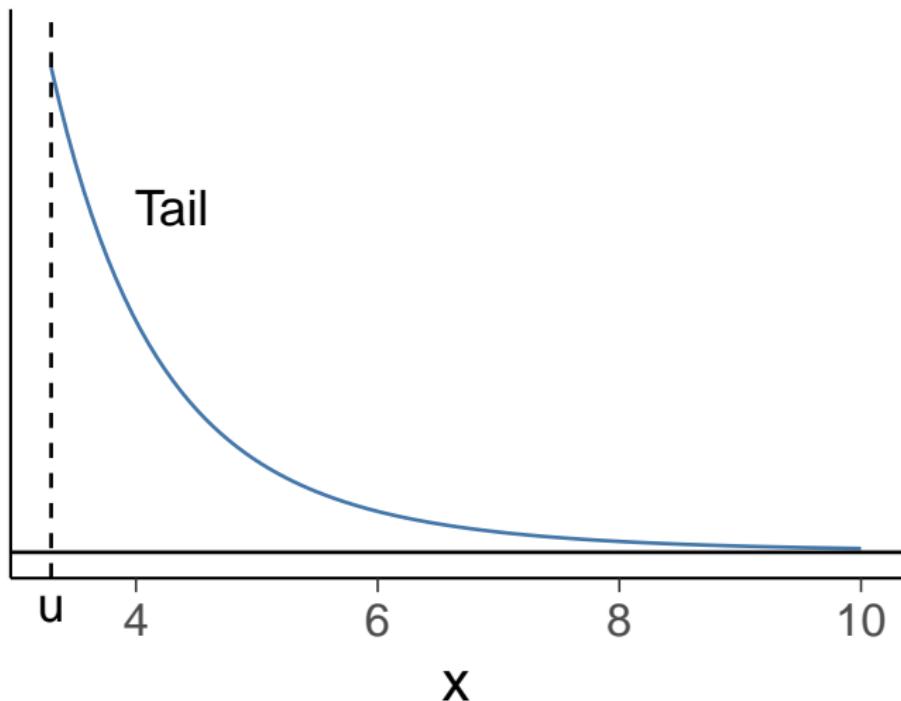
Pareto- \hat{k} diagnostic

Pickands (1975): many distributions have tail ($x > u$) that is well approximated with Generalized Pareto distribution (GPD)



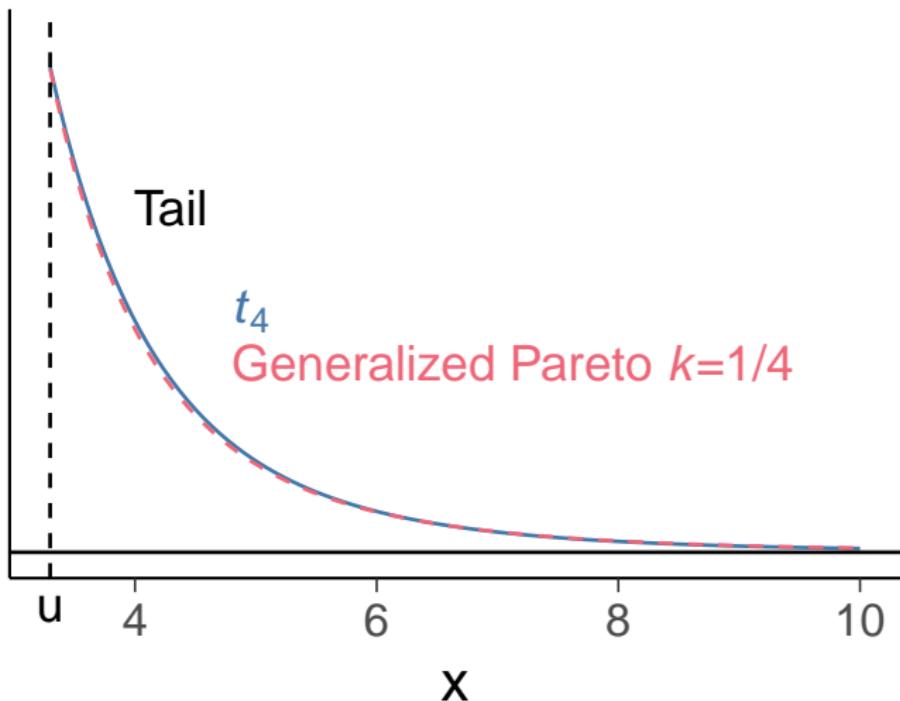
Pareto- \hat{k} diagnostic

Pickands (1975): many distributions have tail ($x > u$) that is well approximated with Generalized Pareto distribution (GPD)



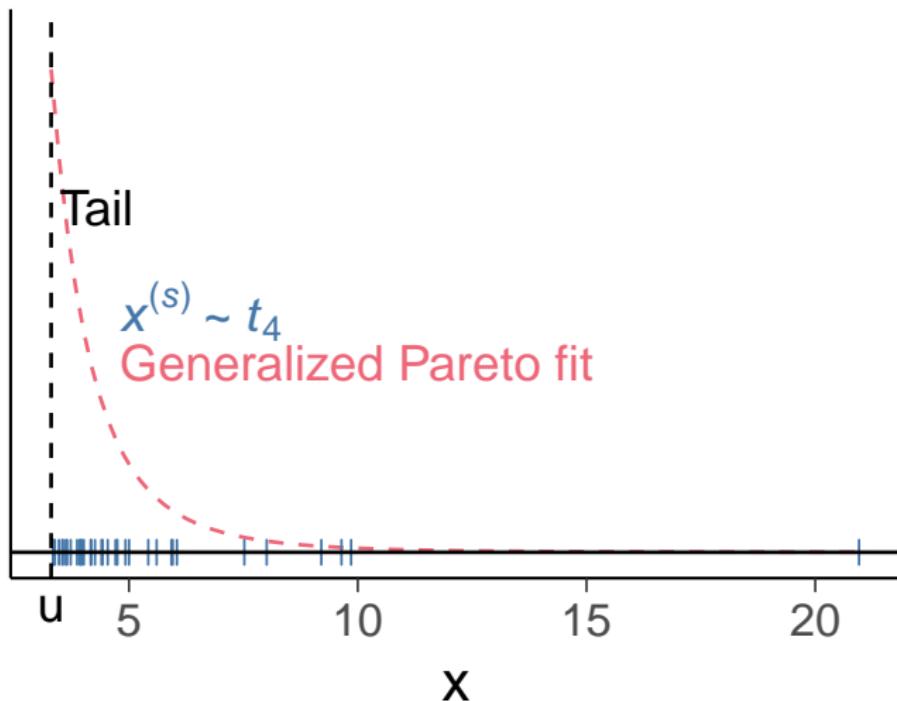
Pareto- \hat{k} diagnostic

Pickands (1975): many distributions have tail ($x > u$) that is well approximated with Generalized Pareto distribution (GPD)

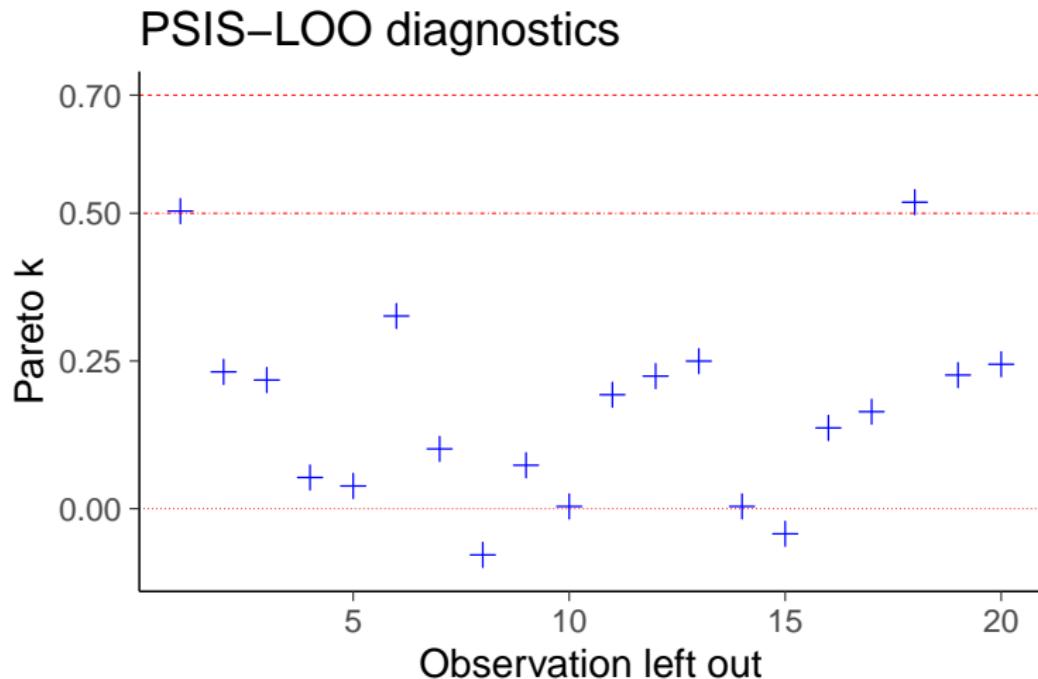


Pareto- \hat{k} diagnostic

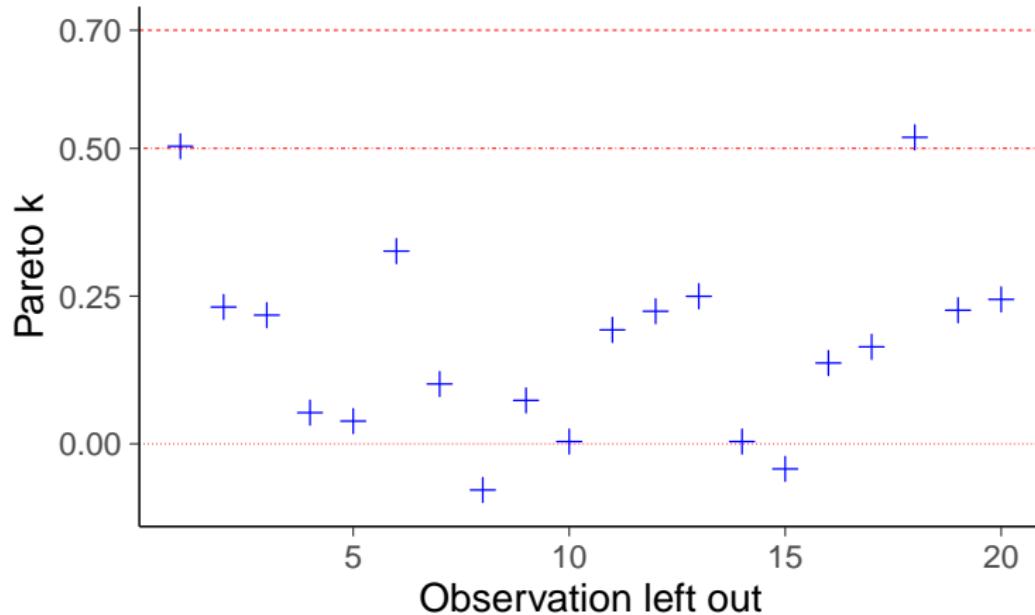
Pickands (1975): many distributions have tail ($x > u$) that is well approximated with Generalized Pareto distribution (GPD)



- Pareto- \hat{k} for each leave-one-out fold indicates reliability of the PSIS-LOO approximation

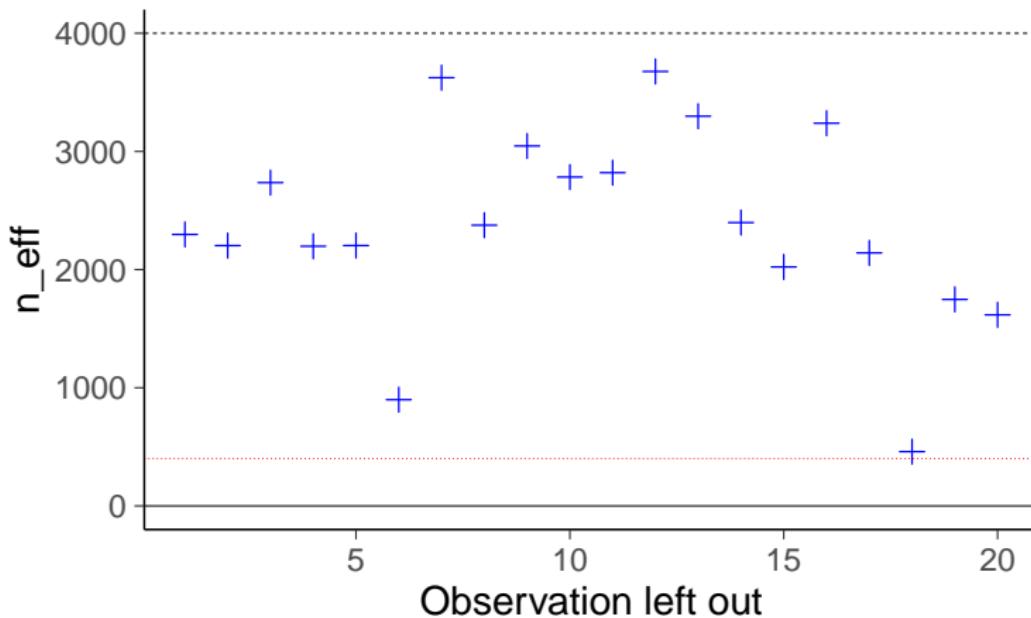


PSIS-LOO diagnostics



All Pareto k estimates are good ($k < 0.7$).

PSIS-LOO diagnostics



All Pareto k estimates are good ($k < 0.7$).

loo package

Computed from 4000 by 20 log-likelihood matrix

	Estimate	SE
elpd_loo	-29.5	3.3
p_loo	2.7	1.0

MCSE of elpd_loo is 0.1.

MCSE and ESS estimates assume MCMC draws (r_{eff} in [0.6, 1.2]).

All Pareto k estimates are good ($k < 0.7$).

See `help('pareto-k-diagnostic')` for details.

see more by Vehtari et al. (2024)

Student retention – loo computation

```
> fit4 <- add_criterion(fit4, 'loo')
Pareto k diagnostic values:
                         Count Pct.    Min. ESS
(-Inf, 0.7]  (good)      32  80.0%   114
(0.7, 1]    (bad)       7  17.5%  <NA>
(1, Inf)   (very bad)  1  2.5%  <NA>
```

Pareto smoothed importance sampling (PSIS)

- Replace the largest weights with ordered statistics of the fitted Pareto distribution
 - equivalent to using model to filter the noise out of the weights

Pareto smoothed importance sampling (PSIS)

- Replace the largest weights with ordered statistics of the fitted Pareto distribution
 - equivalent to using model to filter the noise out of the weights
- Works well if $\hat{k} < 0.7$
- Reduced variability compared to the plain IS
- Reduced bias compared to the truncated IS

Pareto smoothed importance sampling (PSIS)

- Replace the largest weights with ordered statistics of the fitted Pareto distribution
 - equivalent to using model to filter the noise out of the weights
- Works well if $\hat{k} < 0.7$
- Reduced variability compared to the plain IS
- Reduced bias compared to the truncated IS
- Asymptotically consistent under some mild conditions

Pareto- \hat{k} and convergence rate of PSIS

- CLT says that to half the MCSE, need 4 times bigger S

Pareto- \hat{k} and convergence rate of PSIS

- CLT says that to half the MCSE, need 4 times bigger S
- If Pareto- $\hat{k} \approx 0.7$, to half the MCSE, need 10 times bigger S

Pareto- \hat{k} and convergence rate of PSIS

- CLT says that to half the MCSE, need 4 times bigger S
- If Pareto- $\hat{k} \approx 0.7$, to half the MCSE, need 10 times bigger S
- If Pareto- $\hat{k} > 1$, to half the MCSE, nothing helps

Stan code

$$\log(w_i^{(s)}) = \log(1/p(y_i | x_i, \theta^{(s)})) = \text{log_lik}[i]$$

Stan code

$$\log(w_i^{(s)}) = \log(1/p(y_i | x_i, \theta^{(s)})) = \text{log_lik}[i]$$

```
model {
    alpha ~ normal(pmualpha, psalpha);
    beta ~ normal(pmubeta, psbeta);
    y ~ normal(mu, sigma);
}
generated quantities {
    vector[N] log_lik;
    for (i in 1:N)
        log_lik[i] = normal_lpdf(y[i] | mu[i], sigma);
}
```

Stan code

$$\log(w_i^{(s)}) = \log(1/p(y_i | x_i, \theta^{(s)})) = \text{log_lik}[i]$$

```
model {
  alpha ~ normal(pmualpha, psalpha);
  beta ~ normal(pmubeta, psbeta);
  y ~ normal(mu, sigma);
}
generated quantities {
  vector[N] log_lik;
  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu[i], sigma);
}
```

- RStanARM and brms compute `log_lik` by default

`loo()`

- RStan (log_lik in gen. quantities)
`loo(fit)`

loo()

- RStan (log_lik in gen. quantities)
`loo(fit)`
- CmdStanR (log_lik in gen. quantities)
`fit$loo()`

loo()

- RStan (log_lik in gen. quantities)
`loo(fit)`
- CmdStanR (log_lik in gen. quantities)
`fit$loo()`
- RStanARM, brms
`loo(fit)`

loo()

- RStan (log_lik in gen. quantities)
`loo(fit)`
- CmdStanR (log_lik in gen. quantities)
`fit$loo()`
- RStanARM, brms
`loo(fit)`
- brms alternative
`fit <- add_criterion(fit, "loo")`

High Pareto- \hat{k} values

- High Pareto- \hat{k} value indicates the target distribution (LOO posterior) is very different from the proposal distribution (full data posterior)

High Pareto- \hat{k} values

- High Pareto- \hat{k} value indicates the target distribution (LOO posterior) is very different from the proposal distribution (full data posterior)
- This can be caused by
 - well specified, but very flexible model
 - e.g. hierarchical model with one parameter per observation
 - indicated by large p and p_{loo} (e.g. $N/5 < p, p_{\text{loo}} < p$)
 - moment matching or integrated LOO may help

High Pareto- \hat{k} values

- High Pareto- \hat{k} value indicates the target distribution (LOO posterior) is very different from the proposal distribution (full data posterior)
- This can be caused by
 - well specified, but very flexible model
 - e.g. hierarchical model with one parameter per observation
 - indicated by large p and p_{loo} (e.g. $N/5 < p, p_{\text{loo}} < p$)
 - moment matching or integrated LOO may help
 - misspecified model / outliers
 - indicated by $p_{\text{loo}} \ll p$, or $p_{\text{loo}} > p$
 - improve model, check data

High Pareto- \hat{k} values

- High Pareto- \hat{k} value indicates the target distribution (LOO posterior) is very different from the proposal distribution (full data posterior)
- This can be caused by
 - well specified, but very flexible model
 - e.g. hierarchical model with one parameter per observation
 - indicated by large p and p_{loo} (e.g. $N/5 < p, p_{\text{loo}} < p$)
 - moment matching or integrated LOO may help
 - misspecified model / outliers
 - indicated by $p_{\text{loo}} \ll p$, or $p_{\text{loo}} > p$
 - improve model, check data
- See more in CV-FAQ

What if many high Pareto- \hat{k} 's

- `rstan::loo(..., moment_match = TRUE)`
`brms::loo(..., moment_match = TRUE)`
support implicitly adaptive importance sampling with moment matching algorithm by Paananen et al. (2021). See
<http://mc-stan.org/loo/articles/loo2-moment-matching.html>

What if many high Pareto- \hat{k} 's

- `rstan::loo(..., moment_match = TRUE)`
`brms::loo(..., moment_match = TRUE)`
support implicitly adaptive importance sampling with moment matching algorithm by Paananen et al. (2021). See
<http://mc-stan.org/loo/articles/loo2-moment-matching.html>
- `rstanarm::loo(..., k_threshold = 0.7)`
`brms::loo(..., k_threshold = 0.7)`
runs MCMC for the folds with \hat{k} above the threshold

What if many high Pareto- \hat{k} 's

- `rstan::loo(..., moment_match = TRUE)`
`brms::loo(..., moment_match = TRUE)`
support implicitly adaptive importance sampling with moment matching algorithm by Paananen et al. (2021). See
<http://mc-stan.org/loo/articles/loo2-moment-matching.html>
- `rstanarm::loo(..., k_threshold = 0.7)`
`brms::loo(..., k_threshold = 0.7)`
runs MCMC for the folds with \hat{k} above the threshold
- Integrated LOO (for some models)
See <https://users.aalto.fi/~ave/modelselection/roaches.html>

What if many high Pareto- \hat{k} 's

- `rstan::loo(..., moment_match = TRUE)`
`brms::loo(..., moment_match = TRUE)`
support implicitly adaptive importance sampling with moment matching algorithm by Paananen et al. (2021). See
<http://mc-stan.org/loo/articles/loo2-moment-matching.html>
- `rstanarm::loo(..., k_threshold = 0.7)`
`brms::loo(..., k_threshold = 0.7)`
runs MCMC for the folds with \hat{k} above the threshold
- Integrated LOO (for some models)
See <https://users.aalto.fi/~ave/modelselection/roaches.html>
- Use K -fold-CV (more about this later)
`rstanarm::kfold(..., K=10)`
`brms::kfold(..., K=10)`
RStan/CmdStanR vignette
<http://mc-stan.org/loo/articles/loo2-elpd.html>

Student retention – loo computation

PSIS-LOO

```
> fit4 <- add_criterion(fit4, 'loo')
```

Pareto k diagnostic values:

		Count	Pct.	Min. ESS
(-Inf, 0.7]	(good)	32	80.0%	114
(0.7, 1]	(bad)	7	17.5%	<NA>
(1, Inf)	(very bad)	1	2.5%	<NA>

Student retention – loo computation

PSIS-LOO

```
> fit4 <- add_criterion(fit4, 'loo')
Pareto k diagnostic values:
          Count Pct.    Min. ESS
(-Inf, 0.7]  (good)    32   80.0%   114
(0.7, 1]    (bad)      7   17.5%  <NA>
(1, Inf)   (very bad) 1   2.5%  <NA>
```

PSIS-LOO + moment matching + reloo

```
> ...(fit4, 'loo', moment_match=TRUE, reloo=TRUE, overwrite=TRUE)
All Pareto k estimates are good (k < 0.7).
```

Paananen, Piironen, Bürkner, and Vehtari (2021). Implicitly adaptive importance sampling. *Statistics and Computing*, 31, 16.

Cross-validation for model assessment

- CV is good for model assessment when application specific utility/cost functions are used
 - e.g. in concrete quality prediction reported that the absolute error is smaller than X with 90% probability

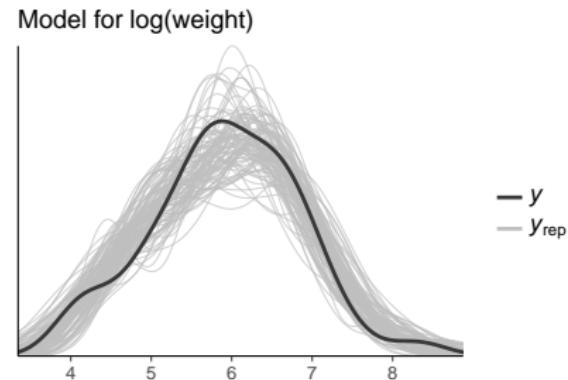
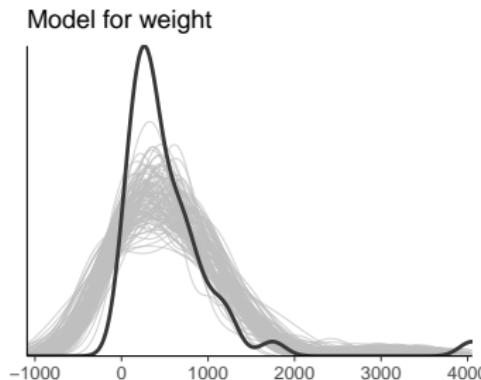
Cross-validation for model assessment

- CV is good for model assessment when application specific utility/cost functions are used
 - e.g. in concrete quality prediction reported that the absolute error is smaller than X with 90% probability
- Also useful in model checking in similar way as posterior predictive checking (PPC)
 - checking calibration of leave-one-out predictive posteriors (`ppc_loo_pit` in `bayesplot`)
 - model misspecification diagnostics (e.g. Pareto- k and `p_loo`)

see demos <https://users.aalto.fi/~ave/casestudies.html>

Sometimes cross-validation is not needed

- Posterior predictive checking is often sufficient

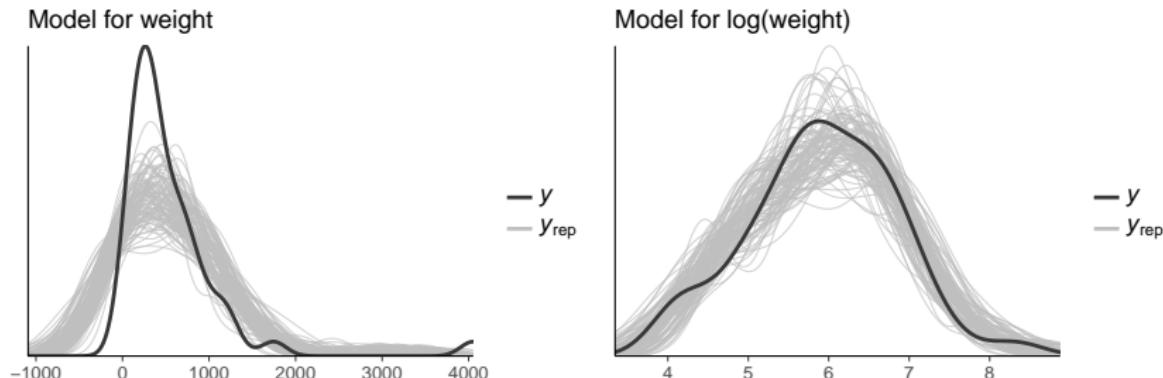


Predicting the yields of mesquite bushes.

Gelman, Hill & Vehtari (2020): Regression and Other Stories, Chapter 11.

Sometimes cross-validation is not needed

- Posterior predictive checking is often sufficient



Predicting the yields of mesquite bushes.

Gelman, Hill & Vehtari (2020): Regression and Other Stories, Chapter 11.

- BDA3, Chapter 6
- Gabry, Simpson, Vehtari, Betancourt, Gelman (2019). Visualization in Bayesian workflow. JRSS A, <https://doi.org/10.1111/rssc.12378>
- mc-stan.org/bayesplot/articles/graphical-ppcs.html

Next week

- Model comparison with LOO-CV
- When is cross-validation applicable?
 - data generating mechanisms and prediction tasks
 - leave-many-out cross-validation
- K -fold cross-validation
- Related methods (WAIC, *IC, BF)
- Model averaging
- Hypothesis testing
- Potential overfitting in model selection