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? (next week)

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)
  - K-fold cross-validation
- When is cross-validation applicable?
  - data generating mechanisms and prediction tasks
  - leave-many-out cross-validation

Next week

- Model comparison and selection (elpd_diff, se)
- Related methods (WAIC, *IC, BF)
- Model averaging
- 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.

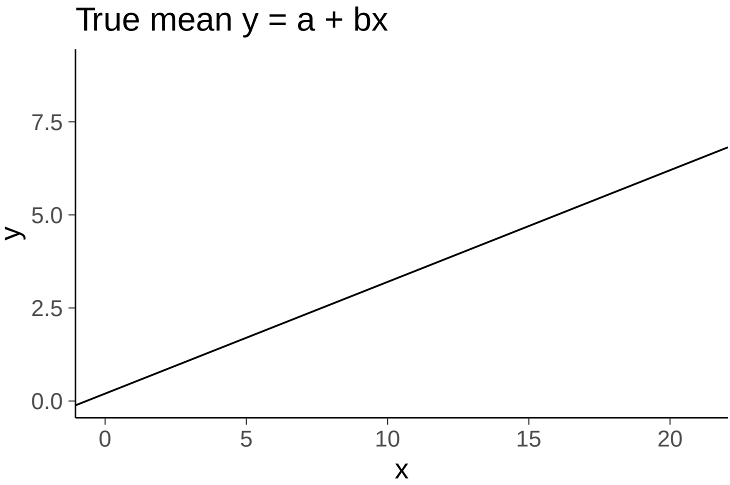
Predictive performance

- 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 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

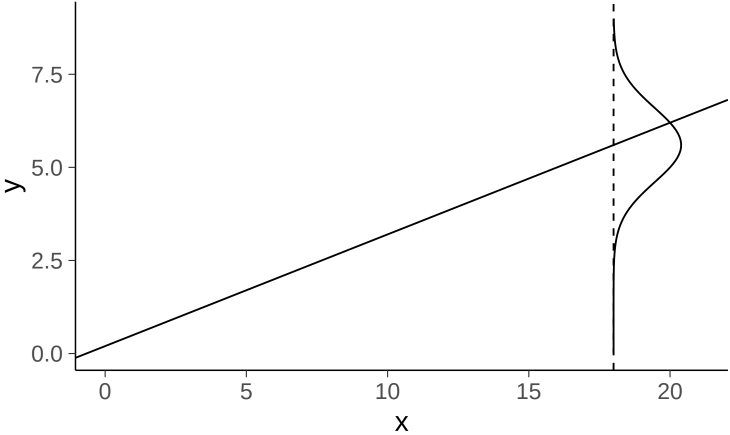
$$\log p(y^{\text{rep}} \mid y, M),$$

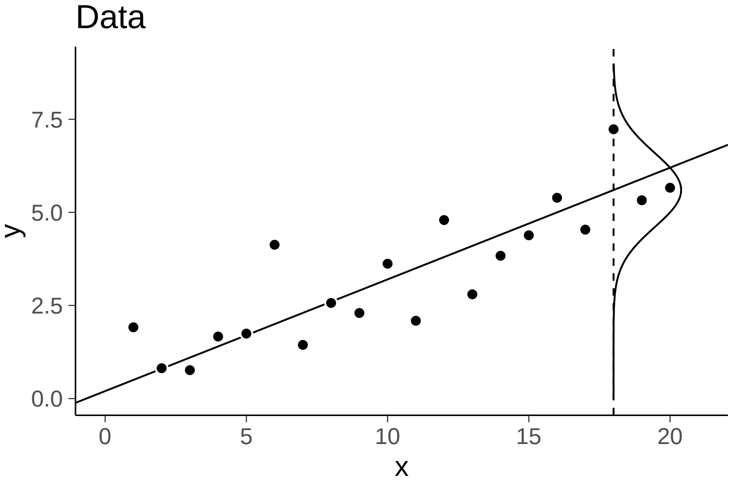
Stan and `loo` package

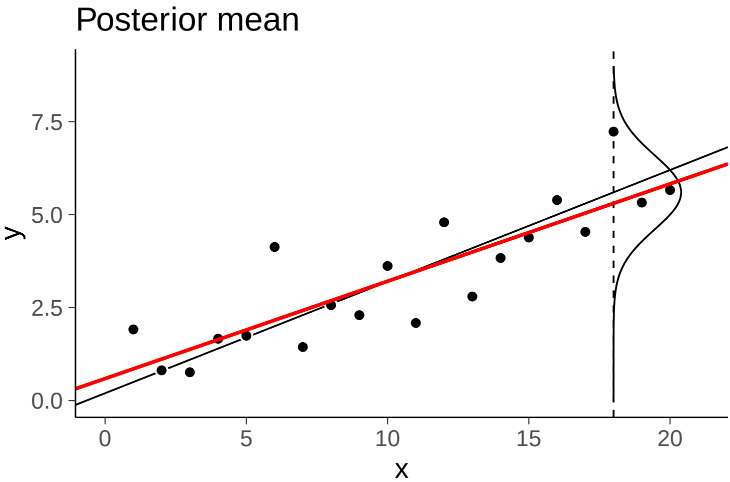
```
 Computed from 4000 by 20 log-likelihood matrix

          Estimate  SE
elpd_loo     -29.5 3.3
p_loo          2.7 1.0
------
Monte Carlo SE of elpd_loo is 0.1.

Pareto k diagnostic values:
                         Count Pct.    Min. n_eff
(-Inf, 0.5]   (good)     18    90.0%   899
 (0.5, 0.7]   (ok)        2    10.0%   459
   (0.7, 1]   (bad)       0     0.0%   <NA>
   (1, Inf)   (very bad)  0     0.0%   <NA>

All Pareto k estimates are ok (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
y
7.5
5.0
2.5
0.0
0
5
10
15
20
x

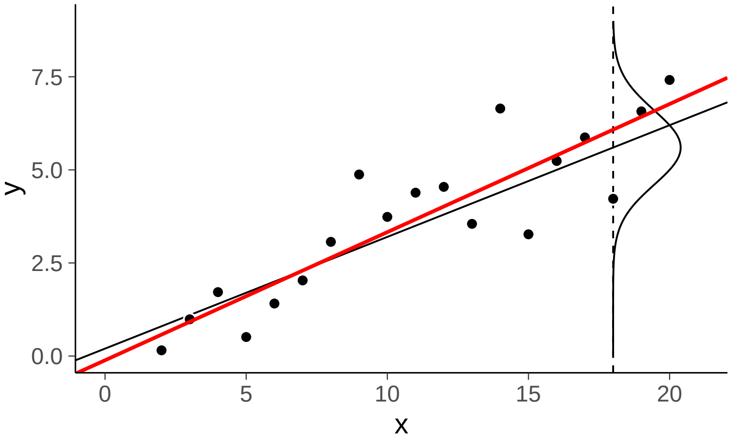
True mean and sigma

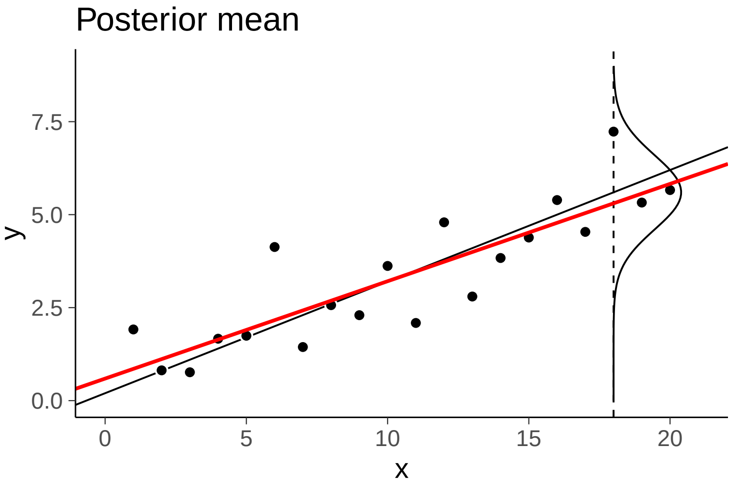


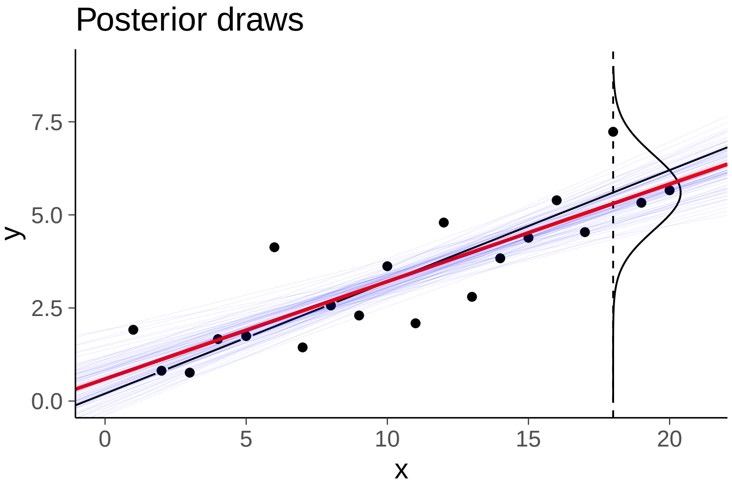

Data
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

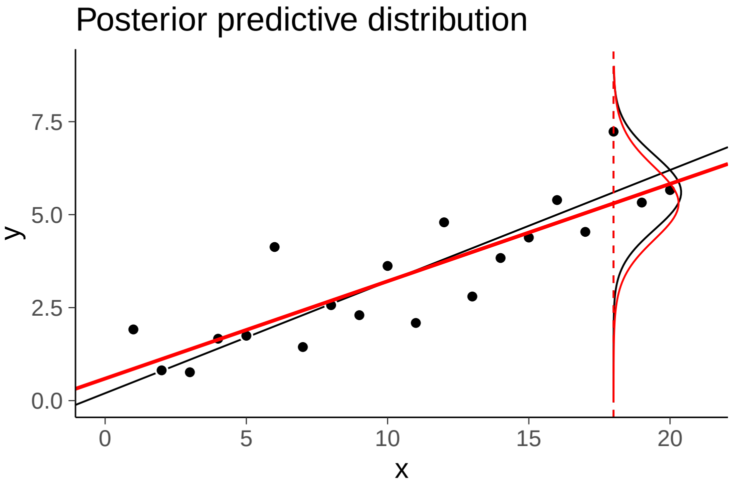

Posterior mean
y
7.5
5.0
2.5
0.0
0
5
10
15
20
x

Posterior mean, alternative data realisation



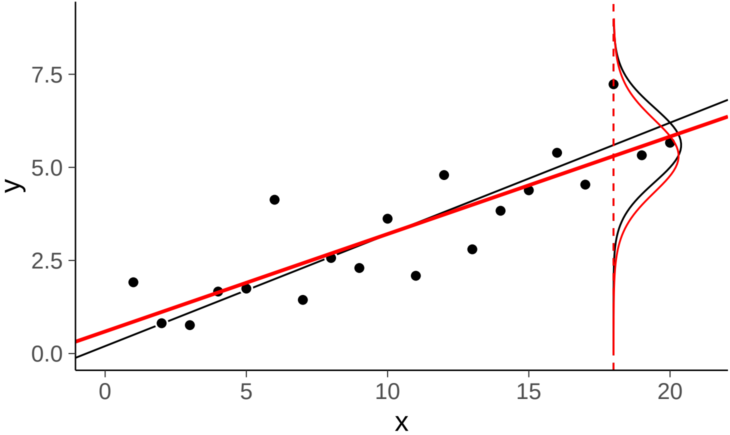

Posterior mean
y
x
0.0
2.5
5.0
7.5
0
5
10
15
20


Posterior draws
y
7.5
5.0
2.5
0.0
0
5
10
15
20
x

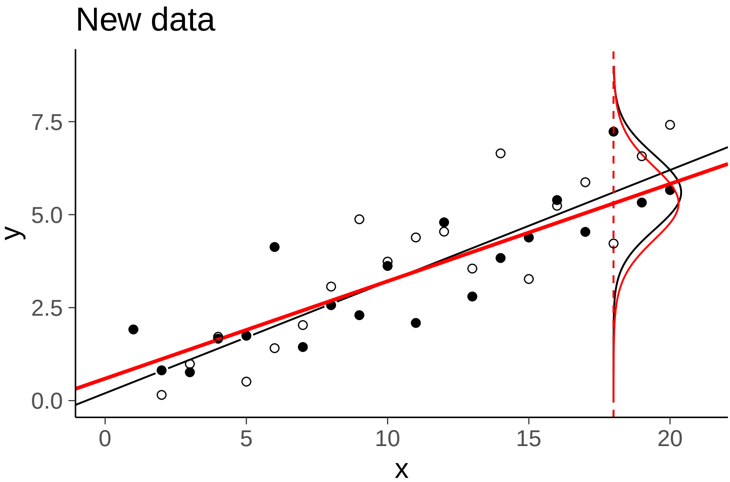


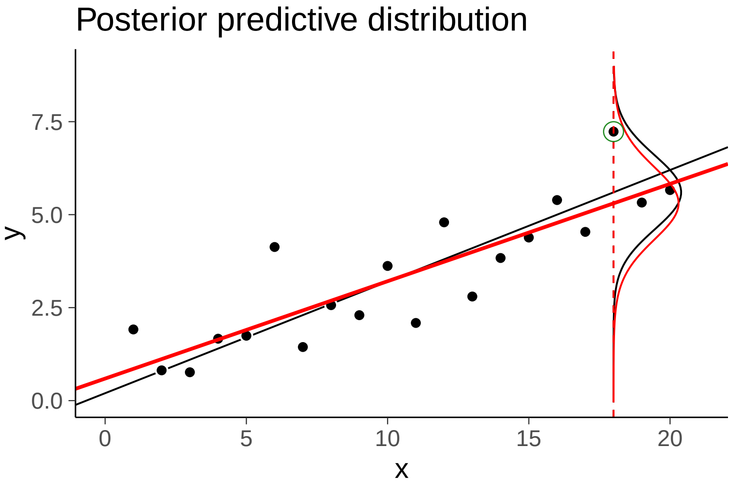
Posterior predictive distribution
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

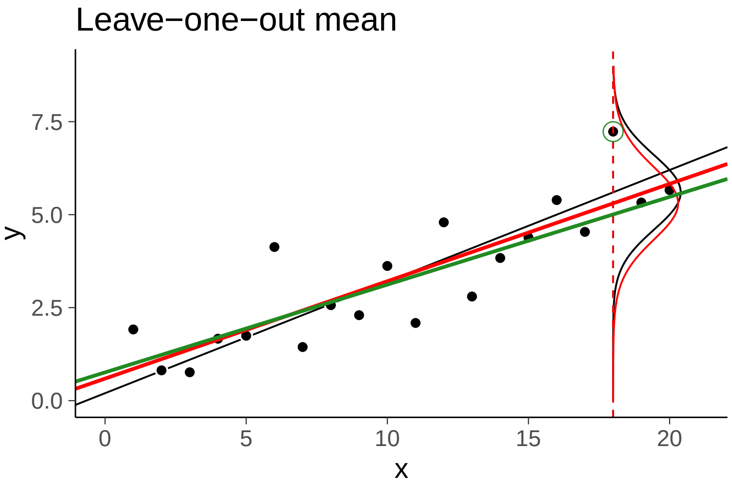
Posterior predictive distribution



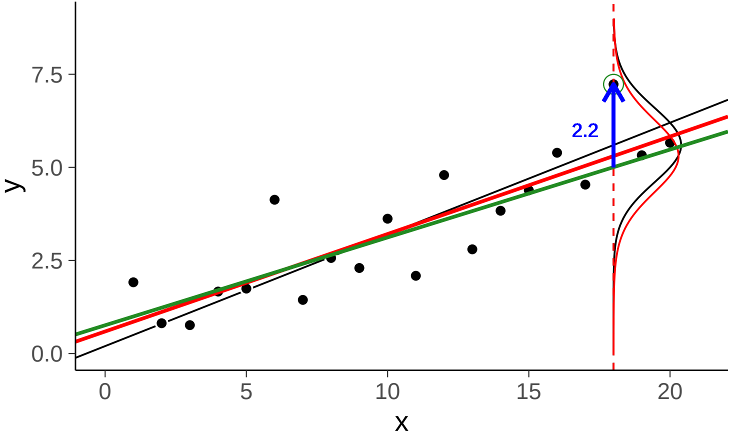
$$p(\tilde{y} \mid \tilde{x} = 18, x, y) = \int p(\tilde{y} \mid \tilde{x} = 18, \theta) p(\theta \mid x, y) d\theta$$


New data
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

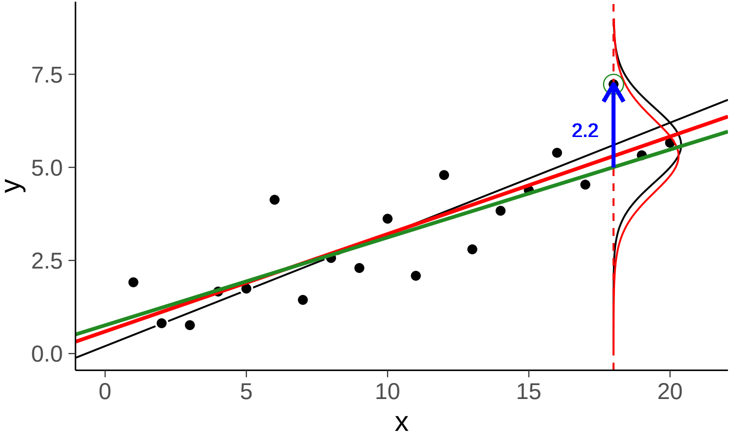

Posterior predictive distribution
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x


Leave−one−out mean
y
7.5
5.0
2.5
0.0
0
5
10
15
20
x

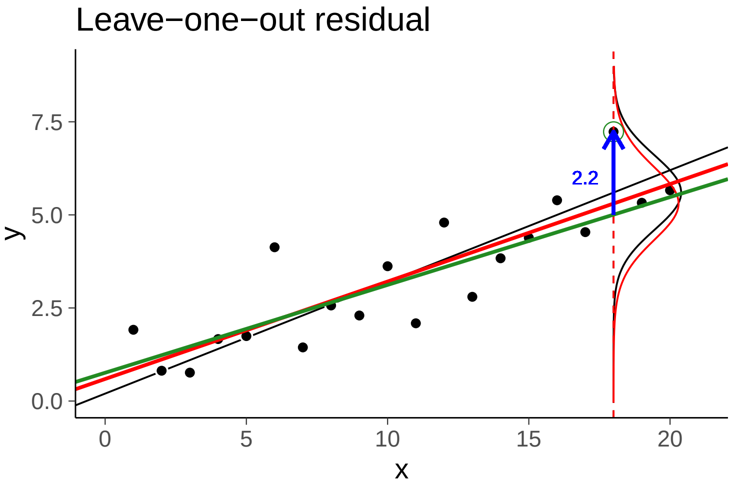
Leave−one−out residual



Leave−one−out residual



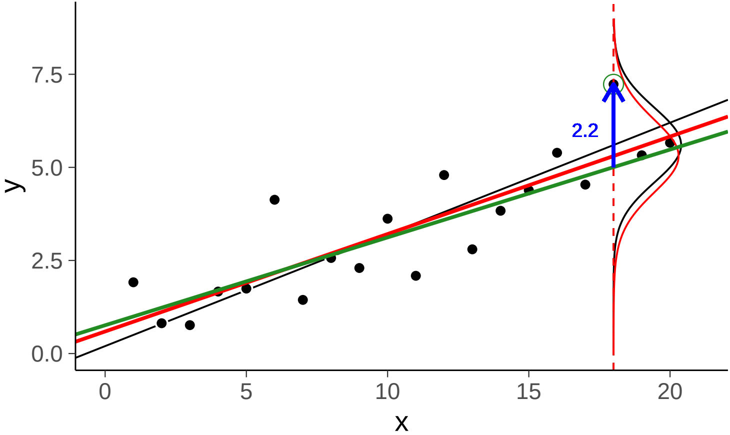
$$y_{18} - E[p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18})]$$



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

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

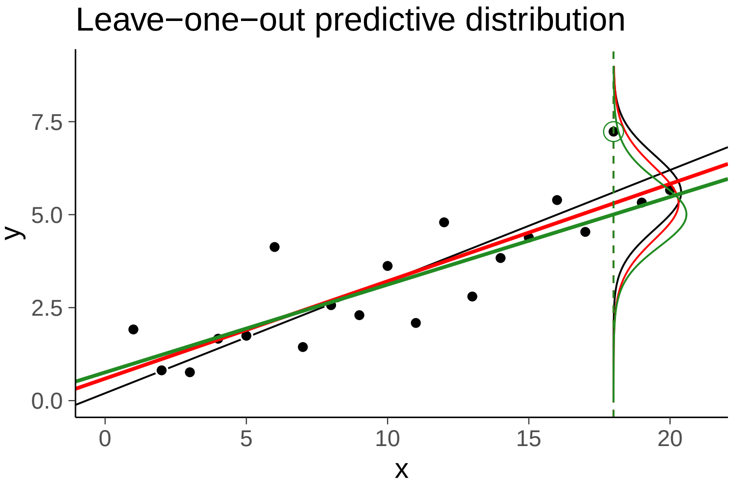
Leave−one−out residual



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

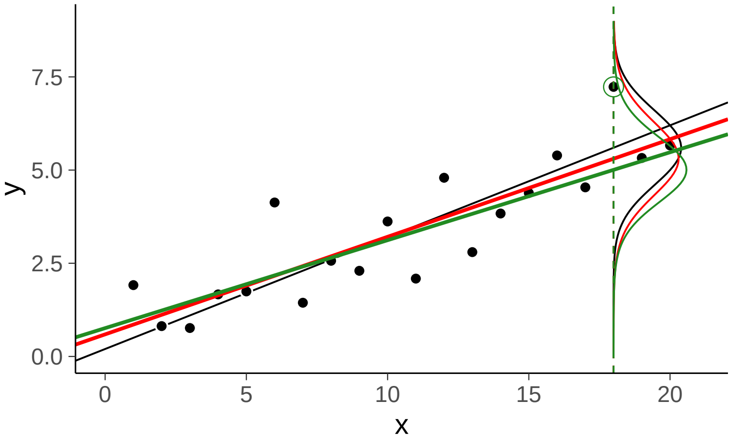
Can be use to compute, e.g., RMSE, R^2, 90% error

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

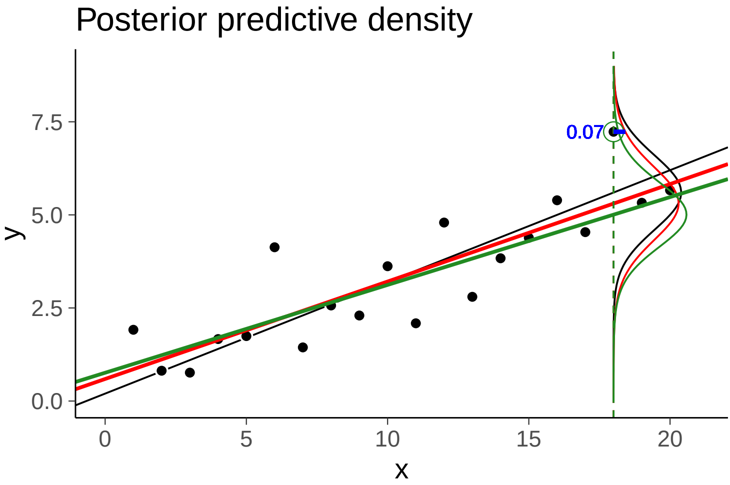


Leave−one−out predictive distribution
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

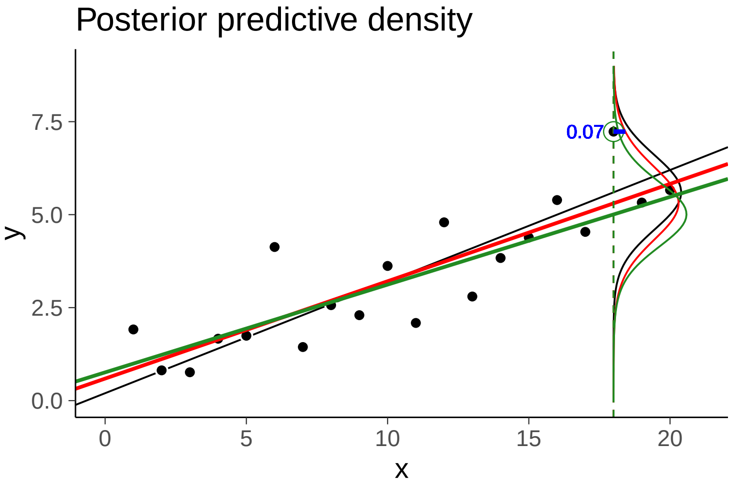
Leave−one−out predictive distribution



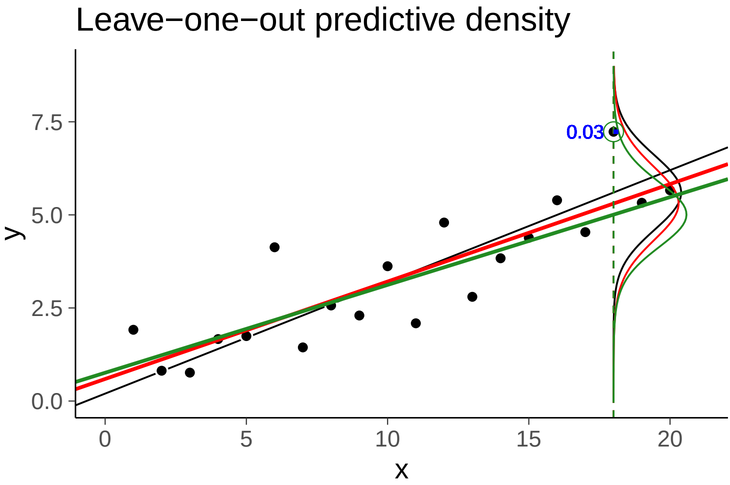
$$p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18}) = \int p(\tilde{y} \mid \tilde{x} = 18, \theta) p(\theta \mid x_{-18}, y_{-18}) d\theta$$



Posterior predictive density
0.07
7.5
5.0
2.5
0.0
0
5
10
15
20
y
x



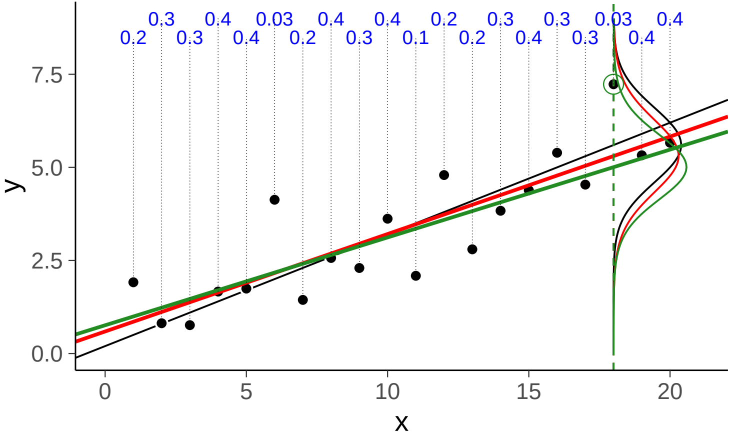
$p(\tilde{y} = y_{18} \mid \tilde{x} = 18, x, y) \approx 0.07$



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

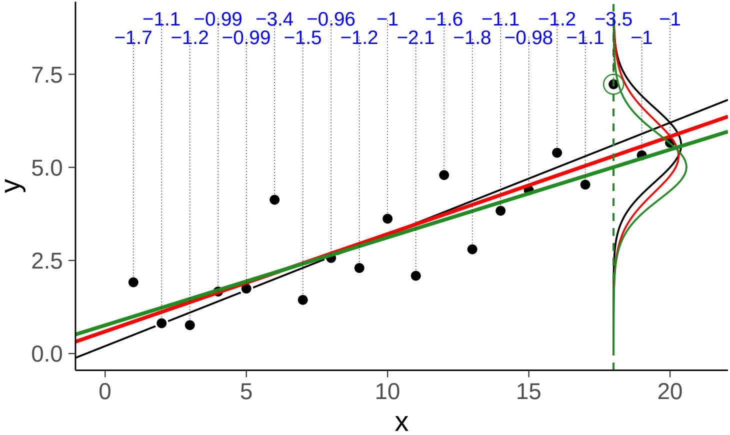
$p(\tilde{y} = y_{18} \mid \tilde{x} = 18, x_{-18}, y_{-18}) \approx 0.03$

Leave−one−out predictive densities



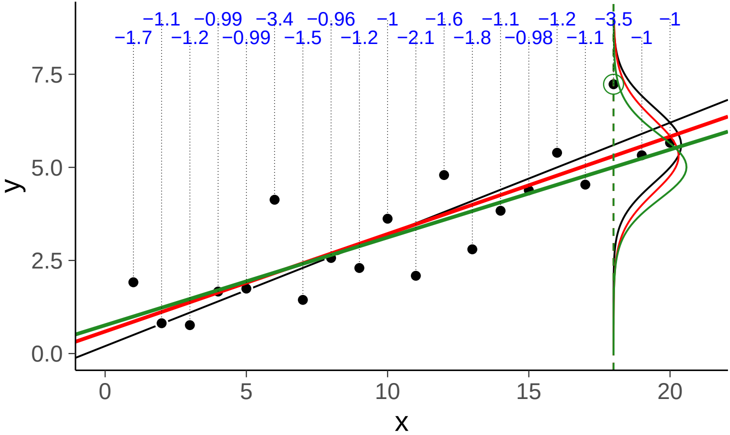
$p(y_i \mid x_i, x_{-i}, y_{-i}), \quad i = 1, \ldots, 20$

Leave−one−out log predictive densities



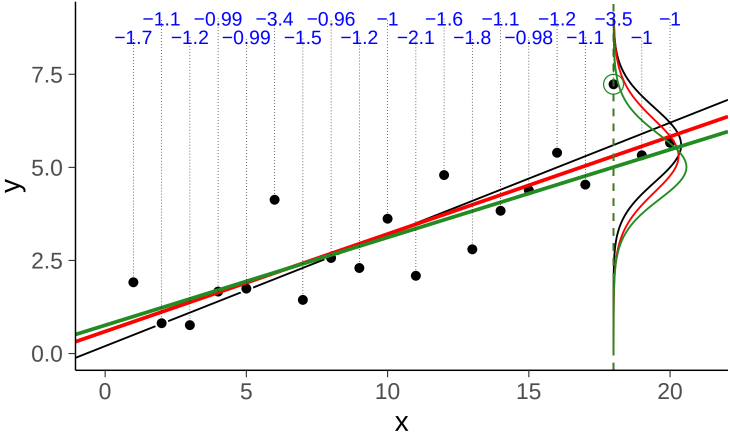
$$\log p(y_i \mid x_i, x_{-i}, y_{-i}), \quad i = 1, \ldots, 20$$

Leave−one−out log predictive densities



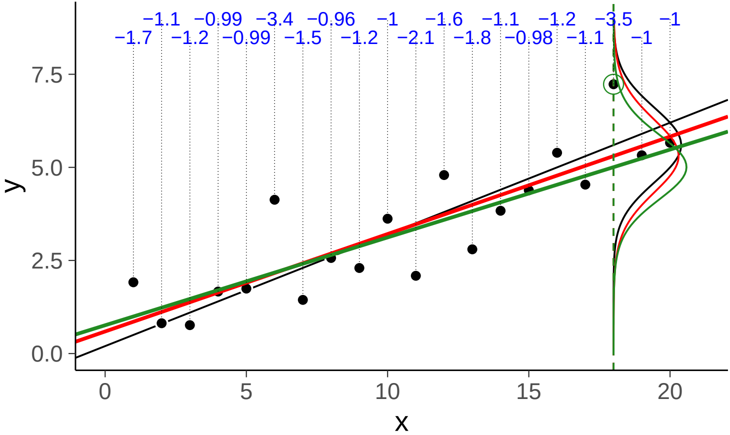
$\sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

Leave−one−out log predictive densities



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

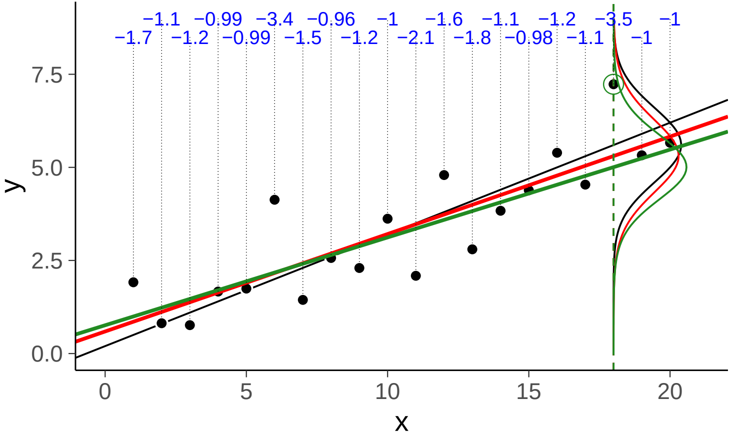
Leave−one−out log predictive densities



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

an estimate of log posterior pred. density for new data

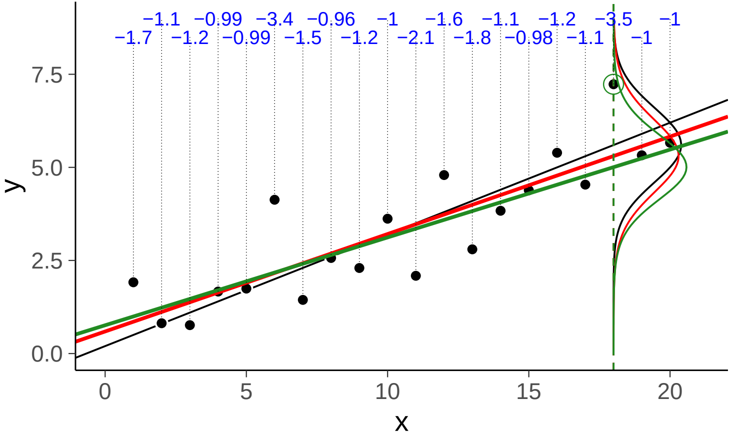
Leave−one−out log predictive densities



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

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

Leave−one−out log predictive densities

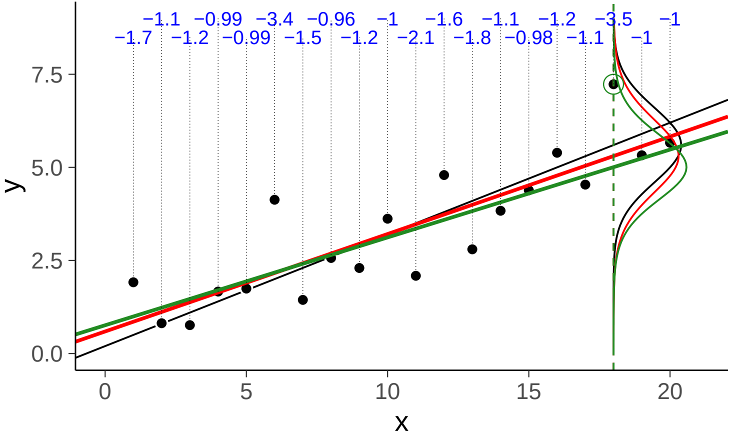


elpd_loo $= \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

p_loo = lpd − elpd_loo ≈ 2.7

Leave−one−out log predictive densities

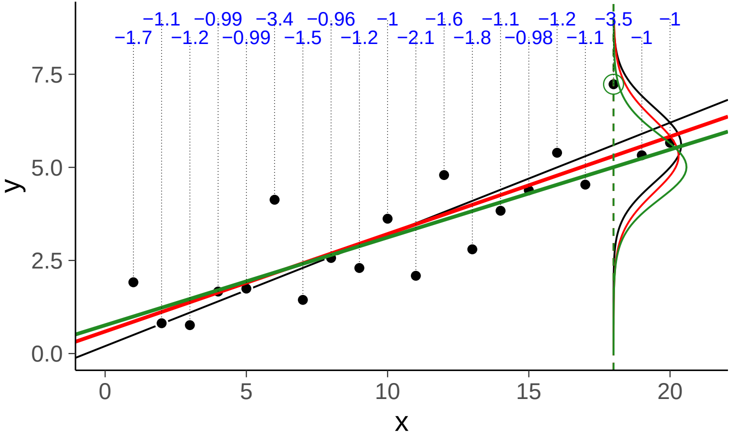


elpd_loo = $\sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

p_loo = lpd − elpd_loo ≈ 2.7

asymptotically approaches p in case of regular faithful model

Leave−one−out log predictive densities



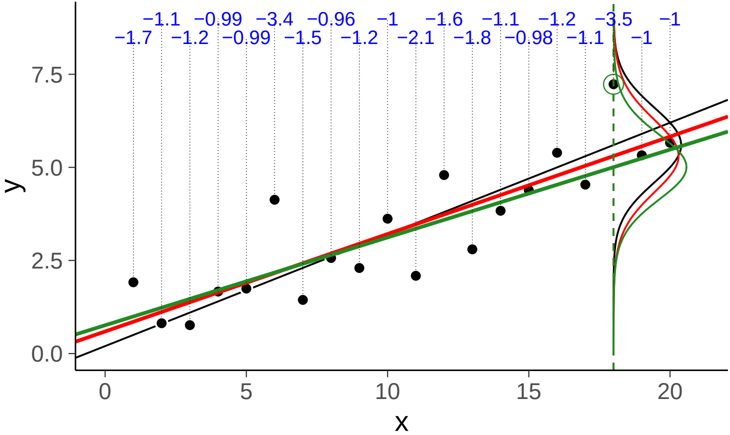
elpd_loo $= \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

p_loo = lpd − elpd_loo ≈ 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



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

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

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

------
Monte Carlo SE of elpd_loo is 0.1.

Pareto k diagnostic values:
                         Count Pct.    Min. n_eff
(-Inf, 0.5]   (good)     18    90.0%   899
 (0.5, 0.7]   (ok)        2    10.0%   459
   (0.7, 1]   (bad)       0     0.0%   <NA>
   (1, Inf)   (very bad)  0     0.0%   <NA>

All Pareto k estimates are ok (k < 0.7).
See help('pareto-k-diagnostic') for details.
```

Helicopter flight time – elpd

```
Computed from 4000 by 145 log-likelihood matrix

         Estimate   SE
elpd_loo    -52.9 10.1
p_loo         9.0  1.3
looic       105.8 20.1
------
Monte Carlo SE of elpd_loo is 0.1.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.
```

Helicopter flight time – R^2

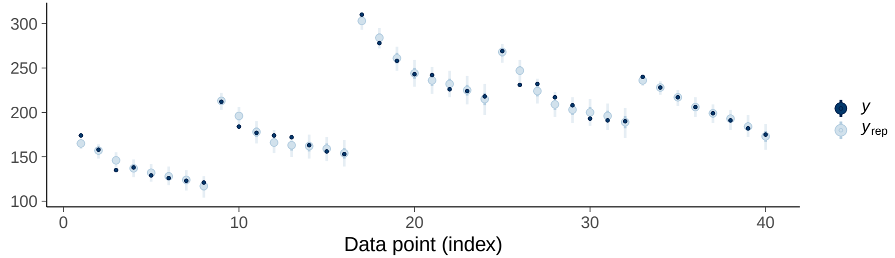
```
> bayes_R2(fit) |> round(digits=2)
   Estimate Est.Error Q2.5 Q97.5
R2     0.41      0.05 0.31   0.5

> loo_R2(fit) |> round(digits=2)
   Estimate Est.Error Q2.5 Q97.5
R2     0.36      0.07 0.22  0.48
```

Student retention – R^2

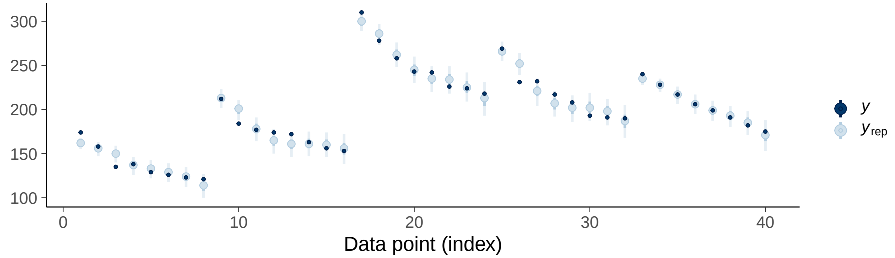
```
> bayes_R2(fit6)|>round(digits=2)
   Estimate Est.Error Q2.5 Q97.5
R2     0.98         0 0.97  0.98

> 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
```

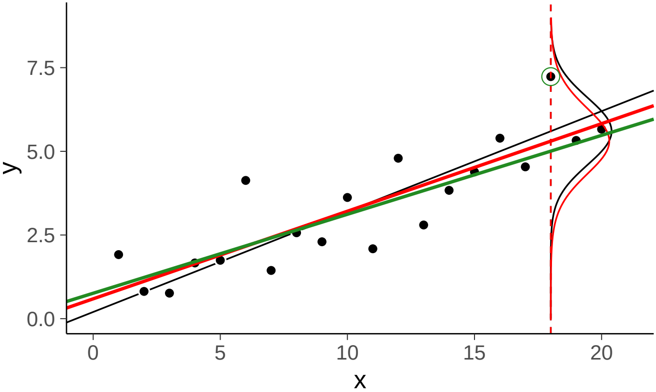
Student retention – elpd (log score)

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
```

Next week more about this

LOO-PIT predictive checking



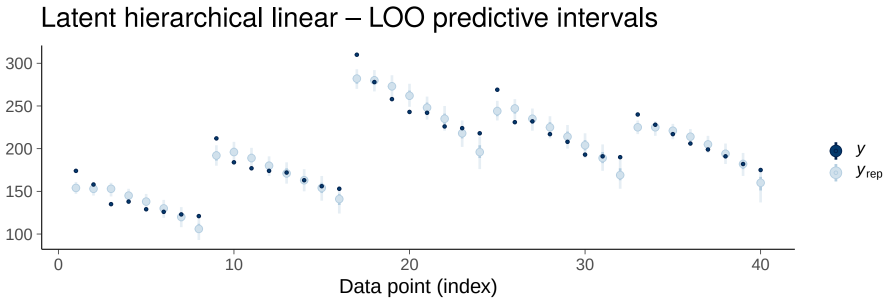
- LOO probability integral transform (LOO-PIT)

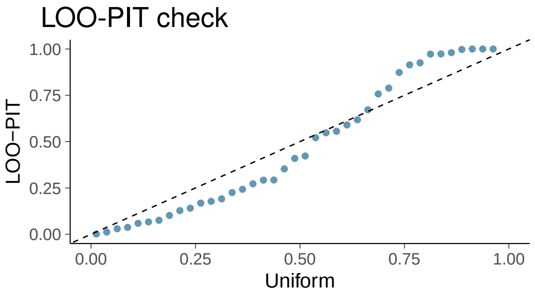
$$p_i = p(y_i^{\text{rep}} \leq y_i | y_{-i})$$

- 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

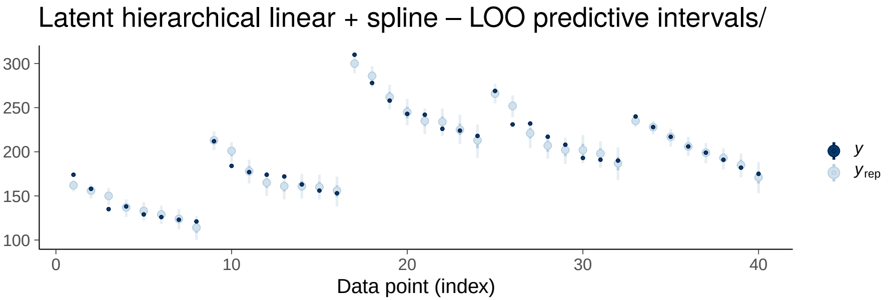
```
pp_check(fit, type = "loo_pit_qq", ndraws=4000)
```

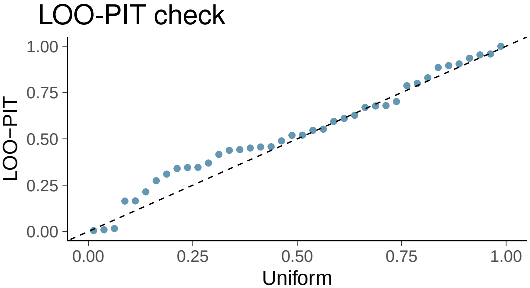




Student retention – LOO-PIT checking

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





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. arXiv:2310.07002.

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
  $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta \mid x, y)$
- Target distribution is LOO-posterior $p(\theta \mid x_{-i}, y_{-i})$

Importance sampling leave-one-out cross-validation

- We want to compute
  $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta \mid x, y)$
- Target distribution is LOO-posterior $p(\theta \mid x_{-i}, y_{-i})$
- Importance ratio

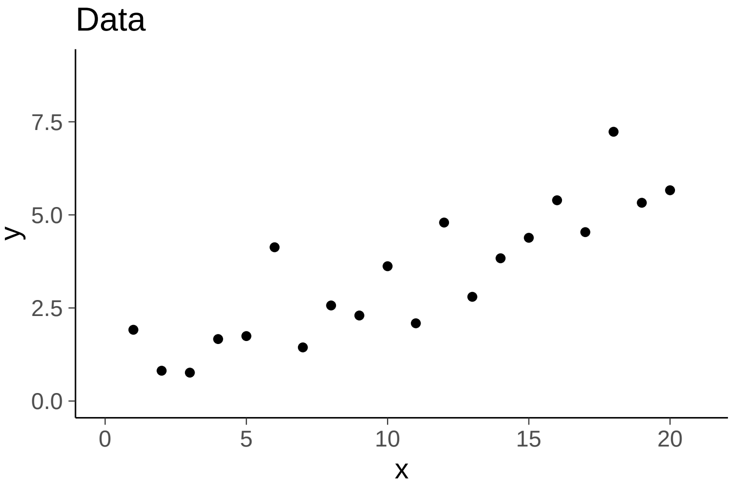
$$w_i^{(s)} = \frac{p(\theta^{(s)} \mid x_{-i}, y_{-i})}{p(\theta^{(s)} \mid x, y)} \propto \frac{1}{p(y_i \mid x_i, \theta^{(s)})}$$

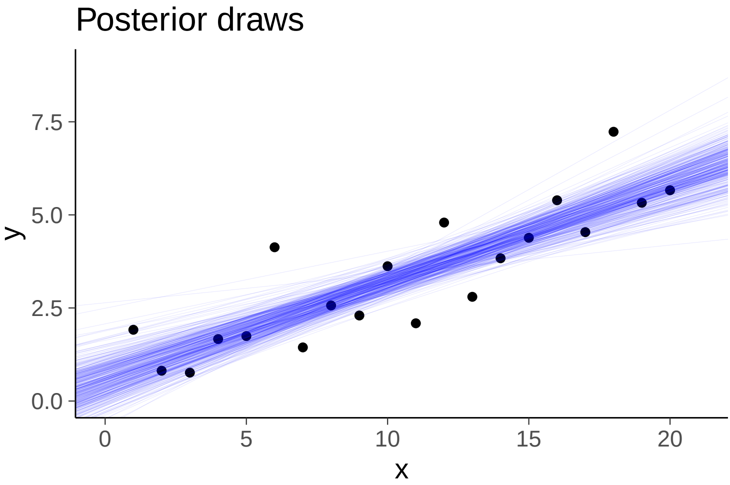
Importance sampling leave-one-out cross-validation

- We want to compute
  $p(y_i \mid x_i, x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta \mid x, y)$
- Target distribution is LOO-posterior $p(\theta \mid x_{-i}, y_{-i})$
- Importance ratio

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

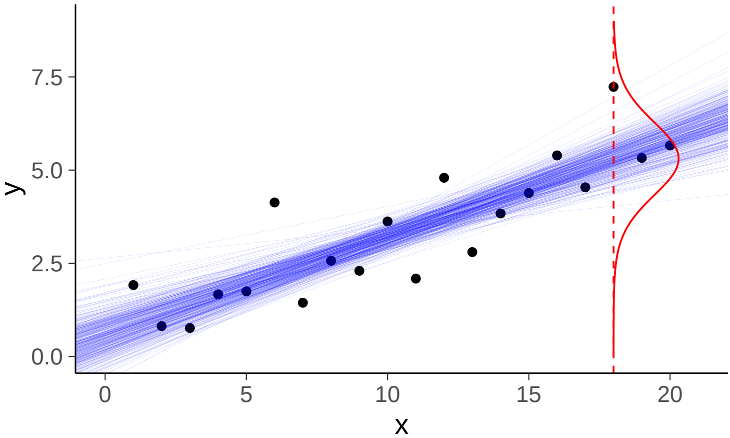
$$\tilde{w}_i^{(s)} = \frac{w_i^{(s)}}{\sum_{s'=1}^{S} w_i^{(s')}}$$


Data
y
7.5
5.0
2.5
0.0
0
5
10
15
20
x



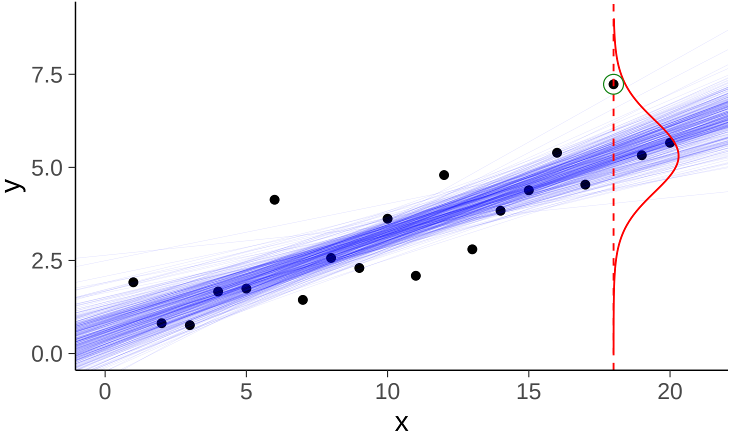
$$\theta^{(s)} \sim p(\theta \mid x, y)$$

Posterior predictive distribution



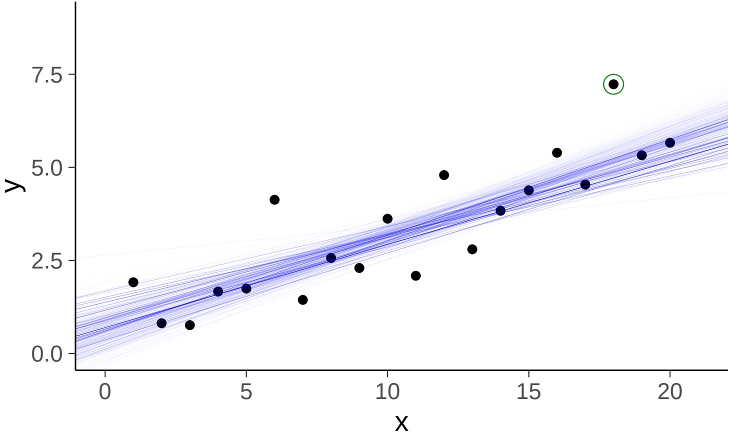
$$\theta^{(s)} \sim p(\theta \mid x, y), \quad p(\tilde{y} \mid \tilde{x}, x, y) \approx \frac{1}{S} \sum_{s=1}^{S} p(\tilde{y} \mid \tilde{x}, \theta^{(s)})$$

Posterior predictive distribution



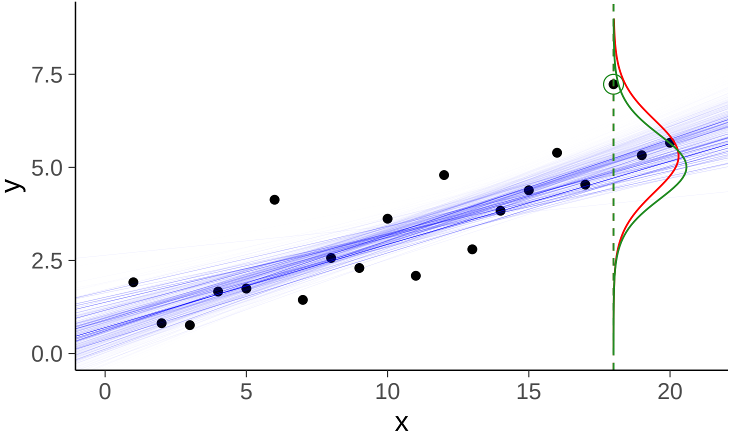
$$\theta^{(s)} \sim p(\theta \mid x, y), \quad p(\tilde{y} \mid \tilde{x}, x, y) \approx \frac{1}{S} \sum_{s=1}^{S} p(\tilde{y} \mid \tilde{x}, \theta^{(s)})$$

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



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

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

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

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

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

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

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

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

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

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

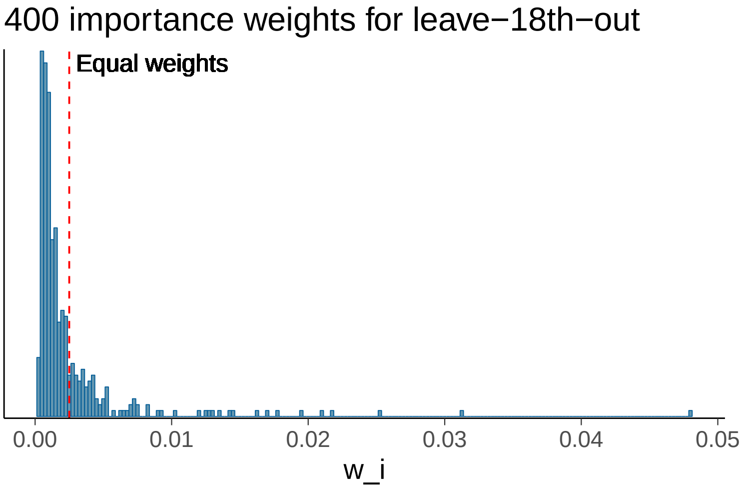
Pareto smoothed importance sampling LOO

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

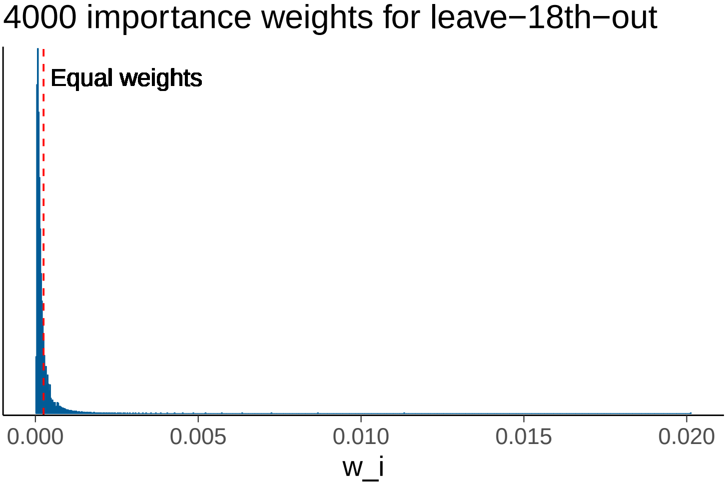
$$p(y_i \mid x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^{S} \left[\tilde{w}_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]$$

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

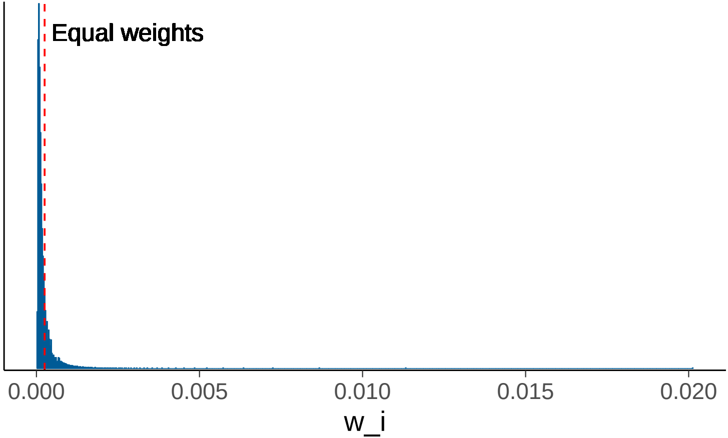
- The variability of importance weights matter
  - Pareto-k diagnostic
  - Pareto smoothed importance sampling LOO (PSIS-LOO)



400 importance weights for leave−18th−out
Equal weights
0.00
0.01
0.02
0.03
0.04
0.05
w_i


4000 importance weights for leave−18th−out
Equal weights
0.000
0.005
0.010
0.015
0.020
w_i

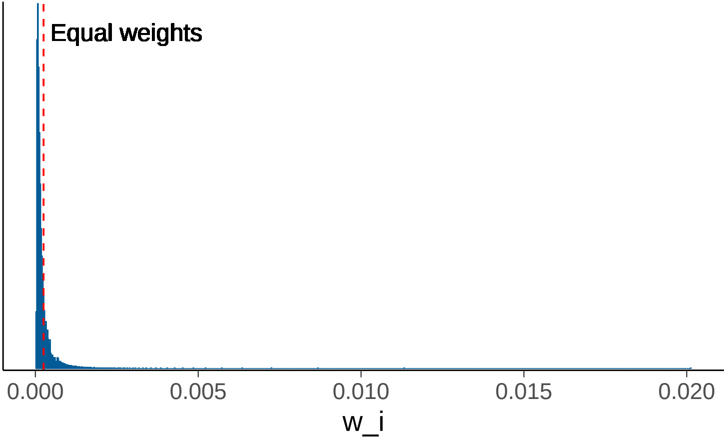
4000 importance weights for leave−18th−out



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

see Vehtari, Gelman & Gabry (2017b)

4000 importance weights for leave−18th−out



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

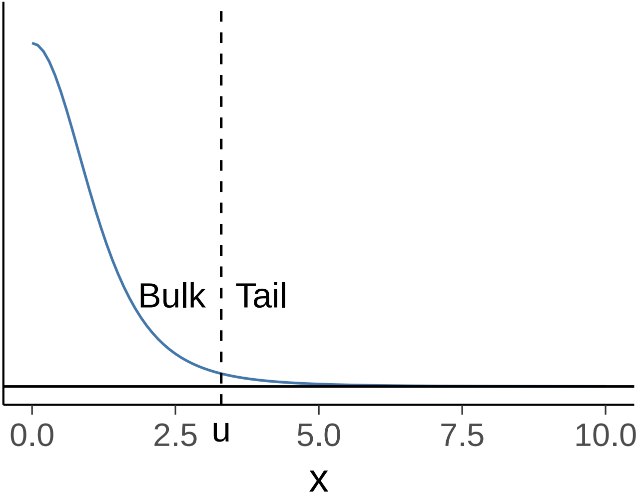
Pareto $\hat{k} \approx$ 0.52

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

see Vehtari, Gelman & Gabry (2017b)

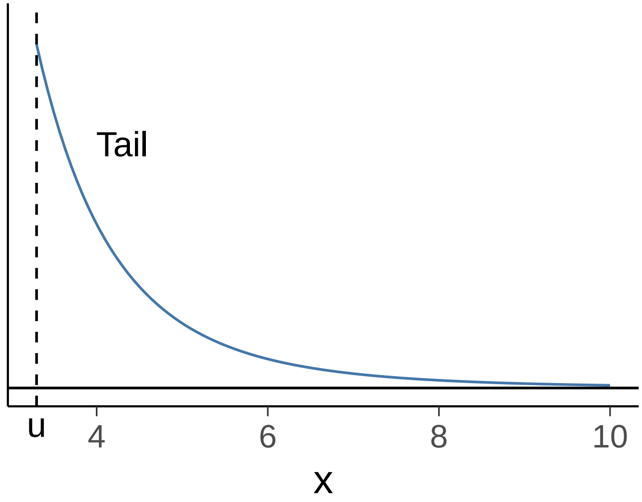
Pareto-$\hat{k}$ diagnostic

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



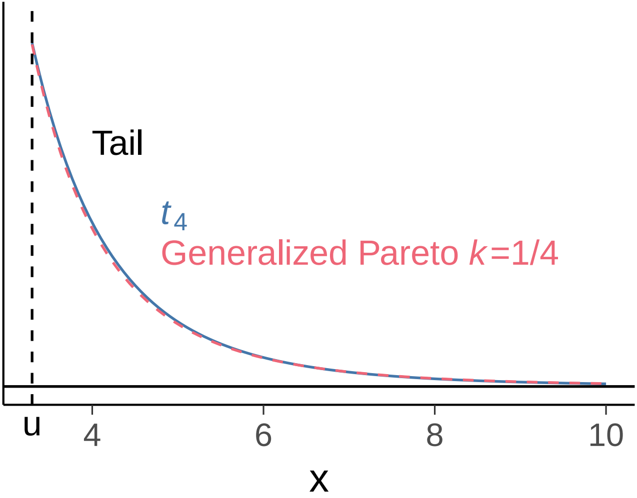
Pareto-$\hat{k}$ diagnostic

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



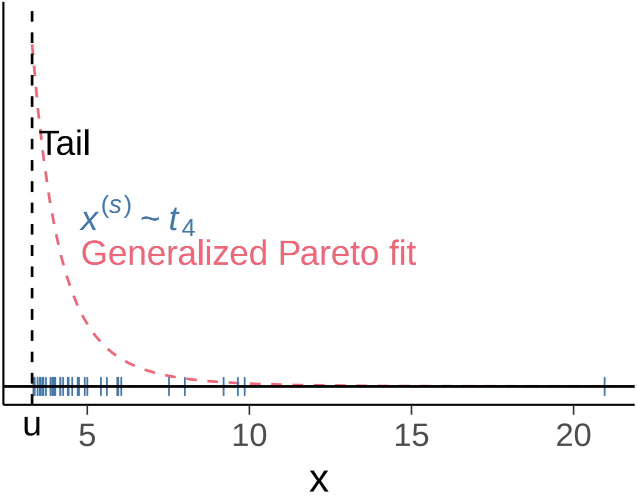
Pareto-$\hat{k}$ diagnostic

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

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

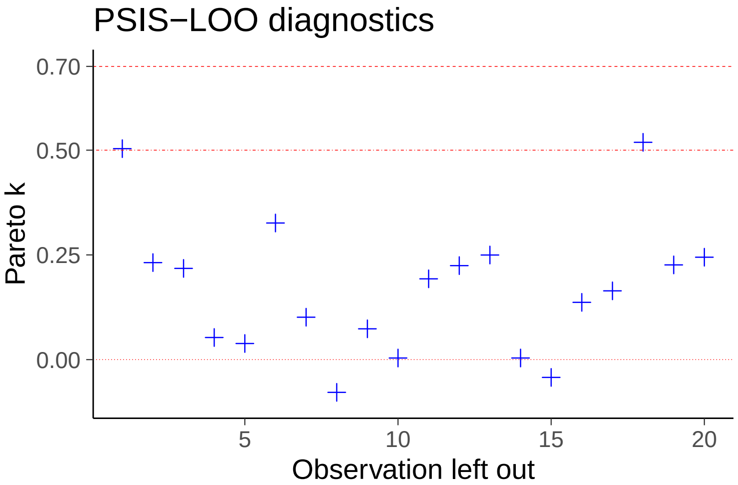
Pareto-$\hat{k}$ and convergence rate of PSIS

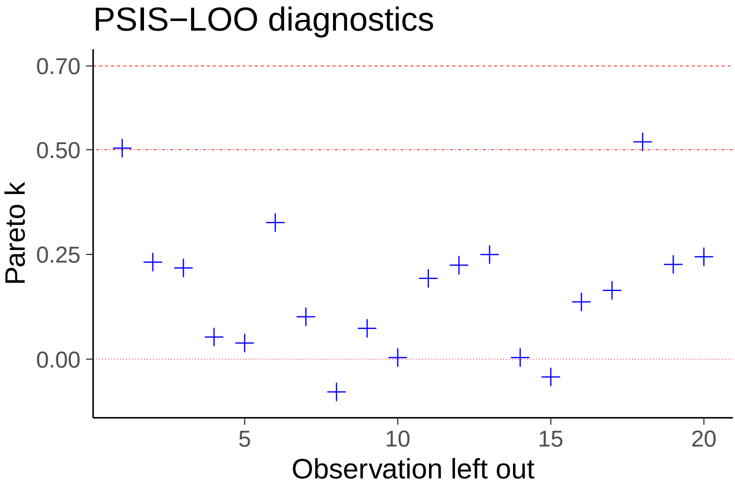
- CLT says that to half the MCSE, need 4 times bigger S
- If Pareto-$\hat{k} \approx 0.7$, to half the MCSE, need 10 times bigger S

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

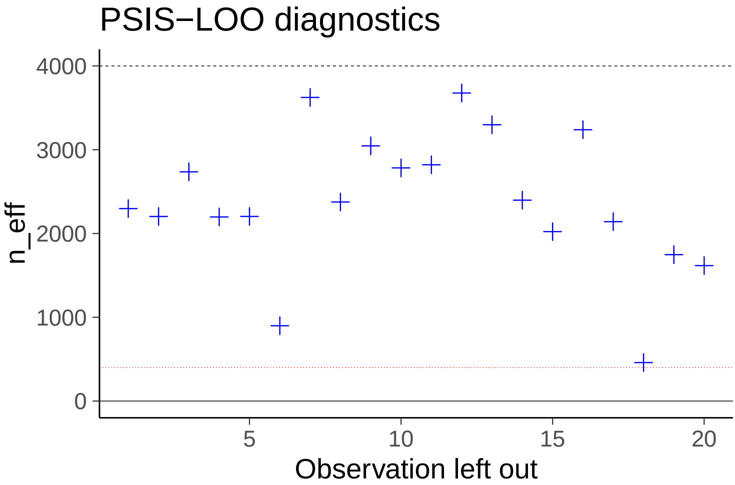
- Pareto-$\hat{k}$ for each leave-one-out fold indicates reliability of the PSIS-LOO approximation





Pareto k diagnostic values:

		Count	Pct.	Min. n_eff
(-Inf, 0.5]	(good)	18	90.0%	899
(0.5, 0.7]	(ok)	2	10.0%	459
(0.7, 1]	(bad)	0	0.0%	<NA>
(1, Inf)	(very bad)	0	0.0%	<NA>



Pareto k diagnostic values:

		Count	Pct.	Min. n_eff
(-Inf, 0.5]	(good)	18	90.0%	899
(0.5, 0.7]	(ok)	2	10.0%	459
(0.7, 1]	(bad)	0	0.0%	<NA>
(1, Inf)	(very bad)	0	0.0%	<NA>

`loo` package

```
 Computed from 4000 by 20 log-likelihood matrix

         Estimate  SE
elpd_loo    -29.5 3.3
p_loo         2.7 1.0

------
Monte Carlo SE of elpd_loo is 0.1.

Pareto k diagnostic values:
                         Count Pct.    Min. n_eff
(-Inf, 0.5]   (good)     18    90.0%   899
 (0.5, 0.7]   (ok)        2    10.0%   459
   (0.7, 1]   (bad)       0     0.0%   <NA>
   (1, Inf)   (very bad)  0     0.0%   <NA>

All Pareto k estimates are ok (k < 0.7).
See help('pareto-k-diagnostic') for details.
```

see more in Vehtari, Gelman & Gabry (2017b)

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

See more in Vehtari, Simpson, Gelman, Yao & Gabry (2021)

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
- Reduced variability compared to the plain IS
- Reduced bias compared to the truncated IS

See more in Vehtari, Simpson, Gelman, Yao & Gabry (2021)

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
- Reduced variability compared to the plain IS
- Reduced bias compared to the truncated IS
- Asymptotically consistent under some mild conditions

See more in Vehtari, Simpson, Gelman, Yao & Gabry (2021)

Stan code

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

Stan code

$$\log(w_i^{(s)}) = \log(1/p(y_i \mid x_i, \theta^{(s)})) = -\texttt{log\_lik[i]}$$

```
...
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

$$\log(w_i^{(s)}) = \log(1/p(y_i \mid x_i, \theta^{(s)})) = -\texttt{log\_lik[i]}$$

```
...
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')`

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

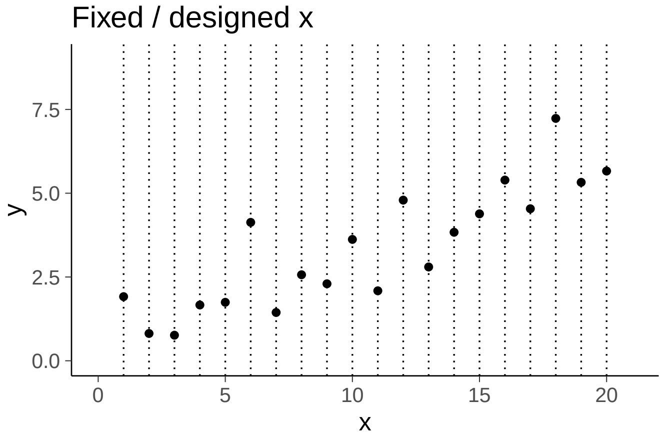
- `rstan::loo(..., moment_match = TRUE)`
  `brms::loo(..., moment_match = TRUE)`
  support implicitly adaptive importance sampling with moment matching algorithm by Paananen et al. (2021). See
  http://mc-stan.org/loo/articles/loo2-moment-matching.html

- `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

Assumptions about the future observations

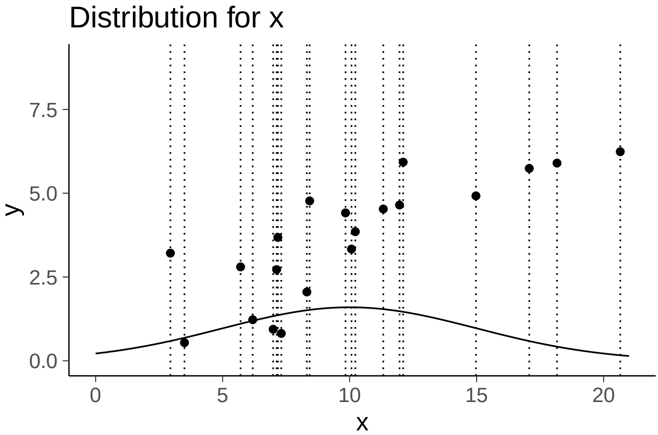


elpd_loo $= \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

LOO is ok for fixed / designed x. SE is uncertainty about $y \mid x$.

Assumptions about the future observations

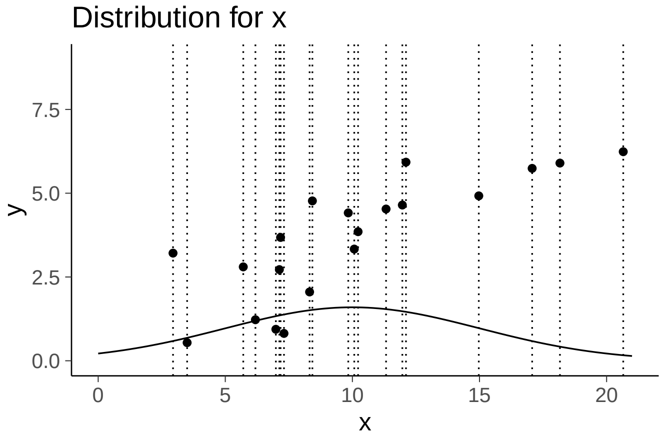


elpd_loo $= \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

LOO is ok for random x. SE is uncertainty about $y \mid x$ and x.

Assumptions about the future observations



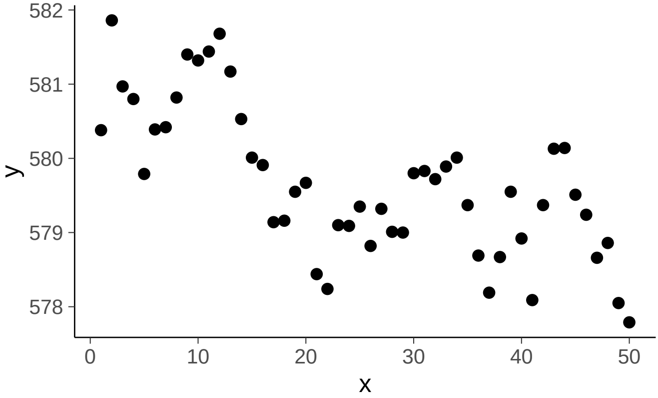
elpd_loo $= \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

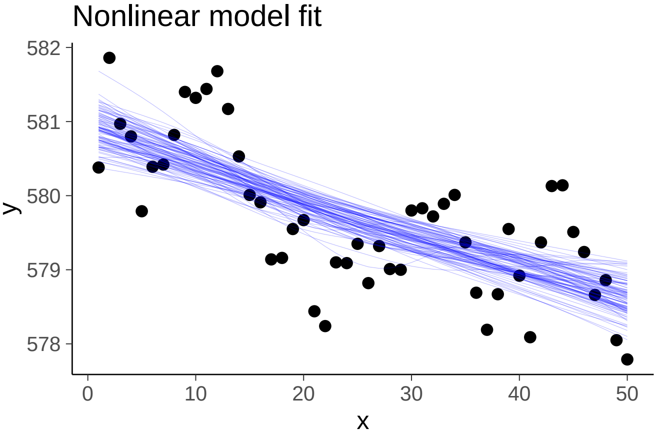
LOO is ok for random x. SE is uncertainty about $y \mid x$ and x.

Covariate shift can be handled with importance weighting or modelling

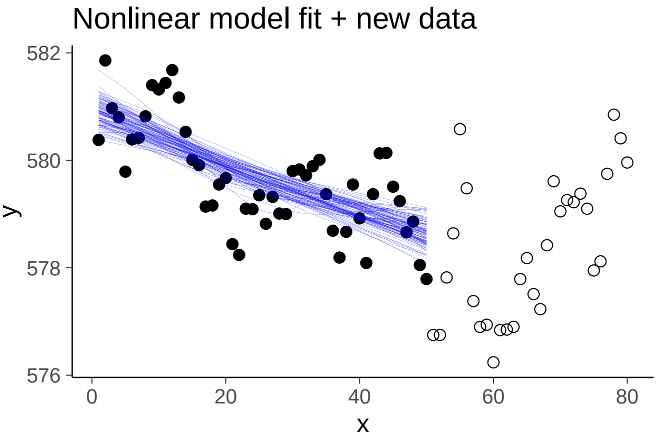
Interpolation vs extrapolation



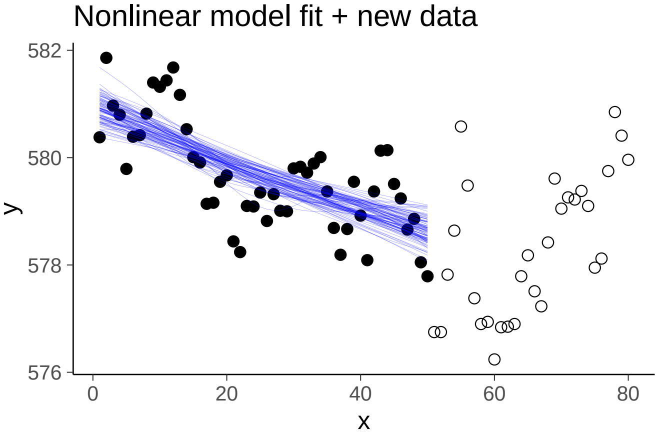
Interpolation vs extrapolation



Interpolation vs extrapolation

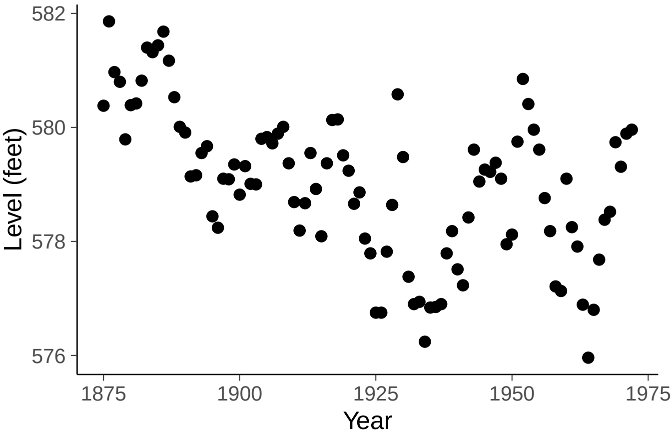


Interpolation vs extrapolation



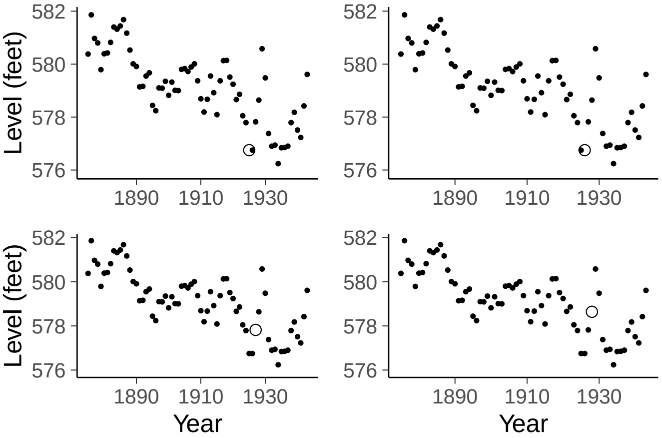
Extrapolation is more difficult

Cross-validation for time series?



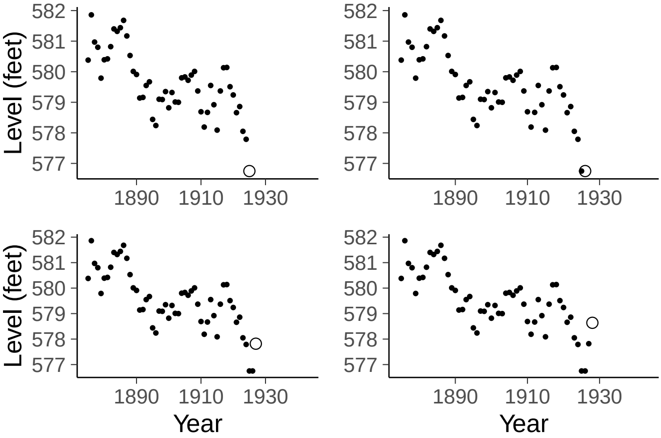
Can LOO or other cross-validation be used with time series?

Cross-validation for time series



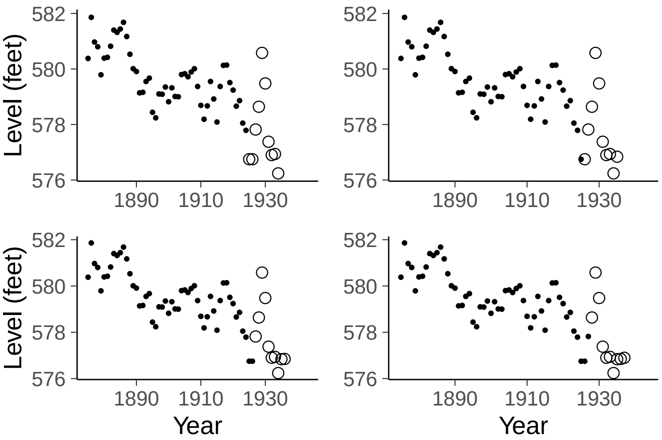
Leave-one-out cross-validation is ok for assessing conditional model

Cross-validation for time series



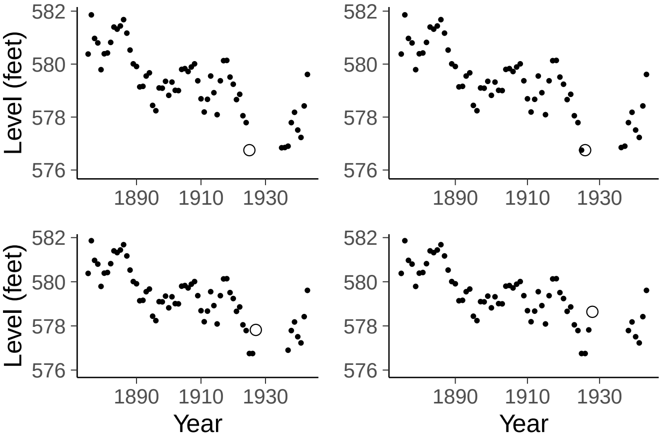
Leave-future-out (LFO) cross-validation is better for predicting future

Cross-validation for time series



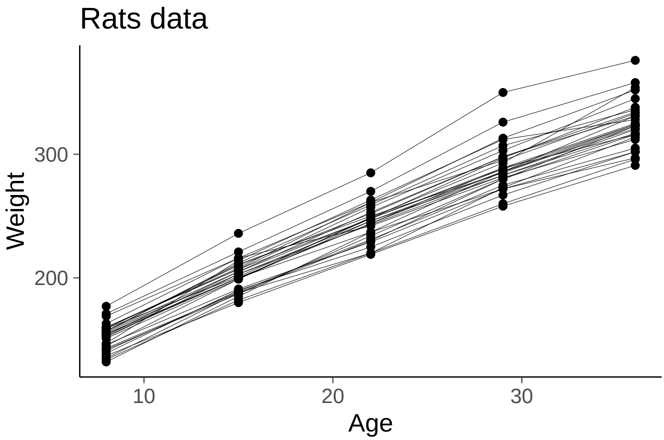
m-step-ahead cross-validation is better for predicting further future

Cross-validation for time series



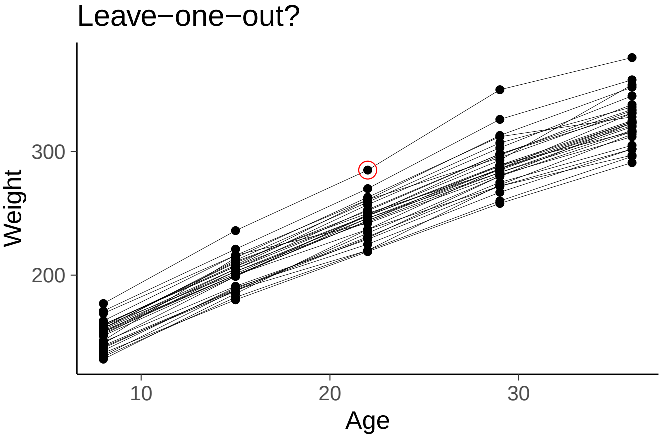
m-step-ahead leave-a-block-out cross-validation

Cross-validation for hierarchical data



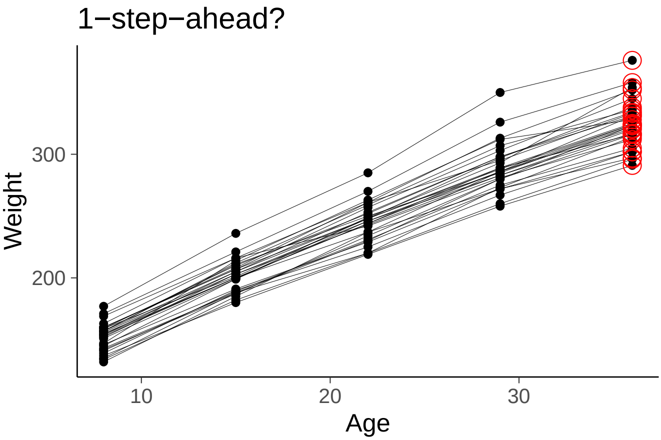
Can LOO or other cross-validation be used with hierarchical data?

Cross-validation for hierarchical data



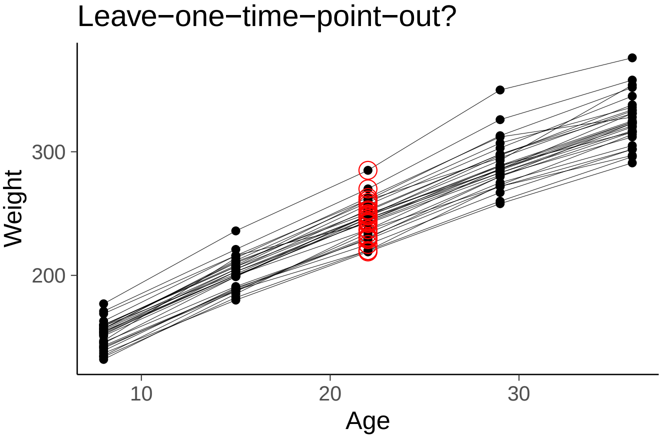
Yes!

Cross-validation for hierarchical data



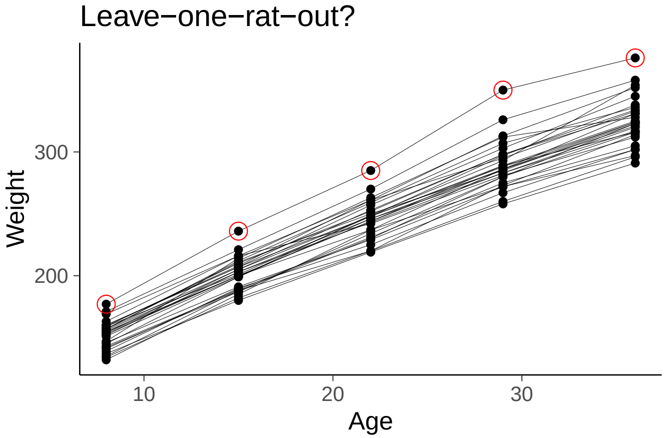
Yes!

Cross-validation for hierarchical data



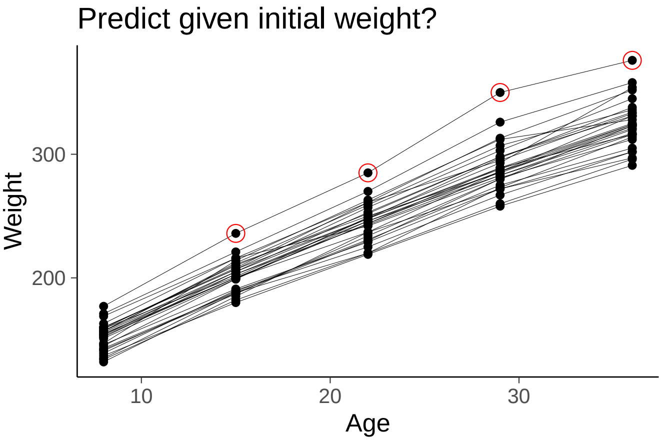
Yes!

Cross-validation for hierarchical data



Yes!

Cross-validation for hierarchical data



Yes!

Summary of data generating mechanisms and prediction tasks

- You have to make some assumptions on data generating mechanism
- Use the knowledge of the prediction task if available
- Cross-validation can be used to analyse different parts, even if there is no clear prediction task

see Vehtari & Ojanen (2012) and CV-FAQ

Pareto smoothed importance sampling CV variants

- PSIS-LOO for hierarchical models
  - leave-one-group out is challenging for PSIS-LOO
  - Stan demo of the challenges and integrated LOO at https://users.aalto.fi/~ave/modelselection/roaches.html
  - see also Merkel, Furr and Rabe-Hesketh (2018)

Pareto smoothed importance sampling CV variants

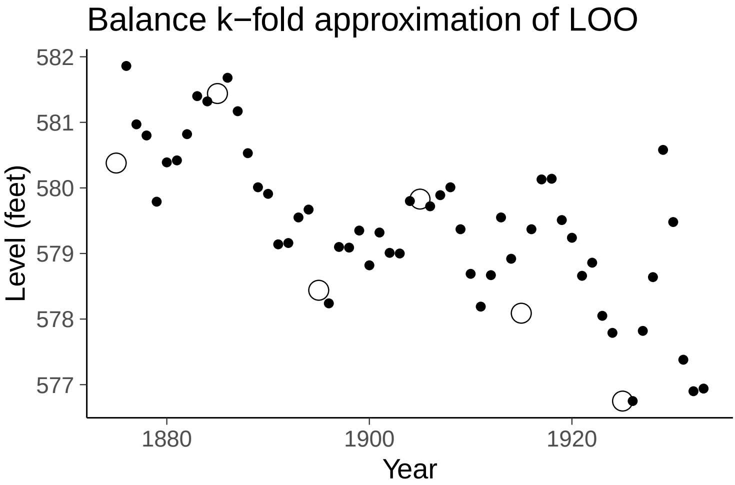
- PSIS-LOO for hierarchical models
  - leave-one-group out is challenging for PSIS-LOO
  - Stan demo of the challenges and integrated LOO at https://users.aalto.fi/~ave/modelselection/roaches.html
  - see also Merkel, Furr and Rabe-Hesketh (2018)
- PSIS-LOO for non-factorized models
  - mc-stan.org/loo/articles/loo2-non-factorizable.html

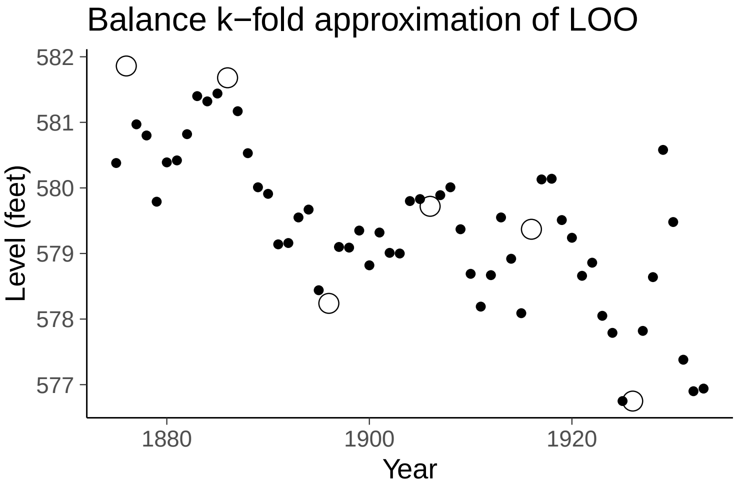
Pareto smoothed importance sampling CV variants

- PSIS-LOO for hierarchical models
  - leave-one-group out is challenging for PSIS-LOO
  - Stan demo of the challenges and integrated LOO at https://users.aalto.fi/~ave/modelselection/roaches.html
  - see also Merkel, Furr and Rabe-Hesketh (2018)
- PSIS-LOO for non-factorized models
  - mc-stan.org/loo/articles/loo2-non-factorizable.html
- PSIS-LOO for time series
  - Approximate leave-future-out cross-validation (LFO-CV) mc-stan.org/loo/articles/loo2-lfo.html

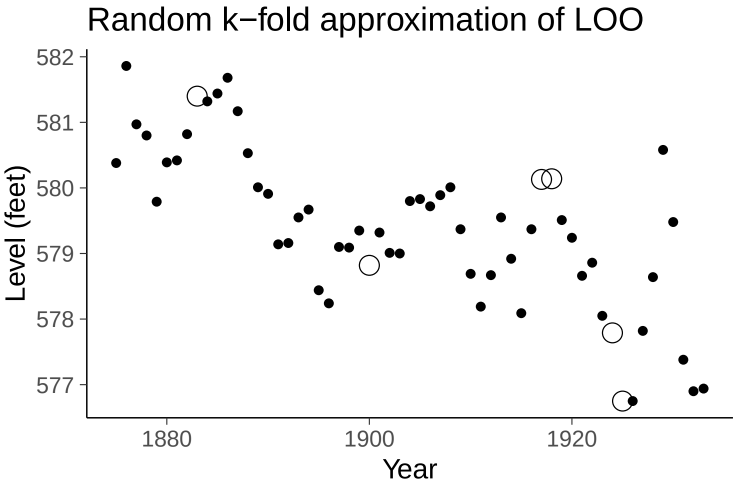
K-fold cross-validation

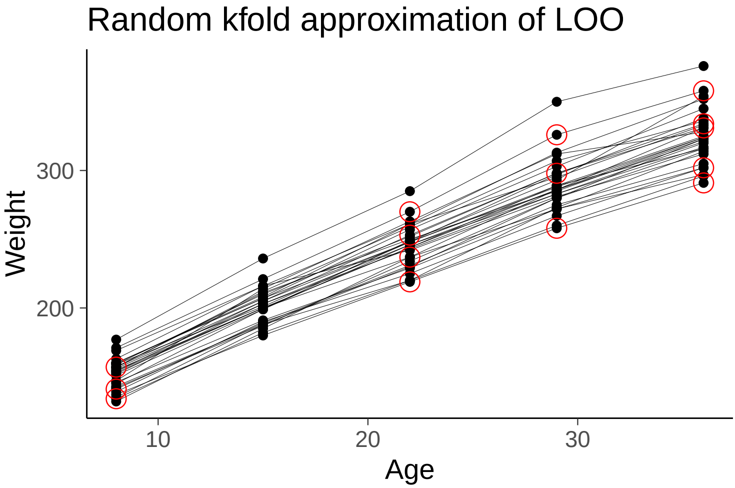
- K-fold cross-validation can approximate LOO
  - the same use cases as with LOO
- K-fold cross-validation can be used for hierarchical models
  - good for leave-one-group-out
- K-fold cross-validation can be used for time series
  - with leave-block-out

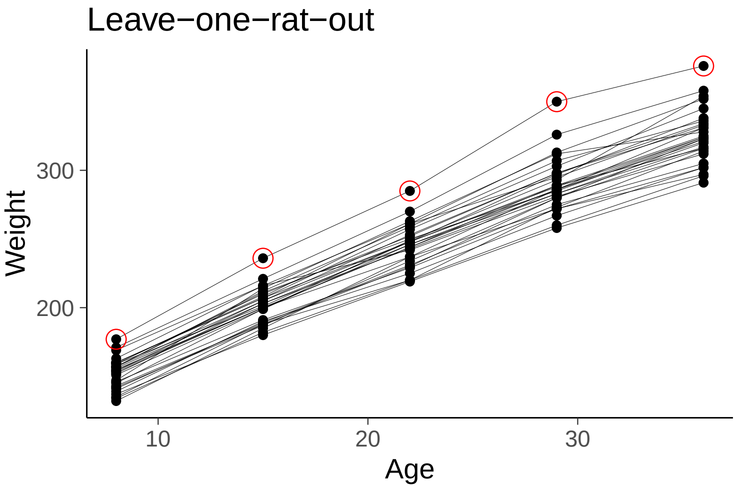

Balance k−fold approximation of LOO
Level (feet)
582
581
580
579
578
577
1880
1900
1920
Year



Balance k−fold approximation of LOO
Level (feet)
582
581
580
579
578
577
1880
1900
1920
Year


Random k−fold approximation of LOO
Level (feet)
582
581
580
579
578
577
1880
1900
1920
Year


Random kfold approximation of LOO
Weight
300
200
10
20
30
Age


Leave−one−rat−out
Weight
300
200
10
20
30
Age

K-fold-CV code

- RStan, CmdStanR
  See vignette http://mc-stan.org/loo/articles/loo2-elpd.html
- RStanARM, brms
  `kfold(fit)`
- Alternative data divisions
  `kfold_split_random()`

  `kfold_split_balanced()`

  `kfold_split_stratified()`

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

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_loo < p`)
    - moment matching or integrated LOO may help

High Pareto-$\hat{k}$ values

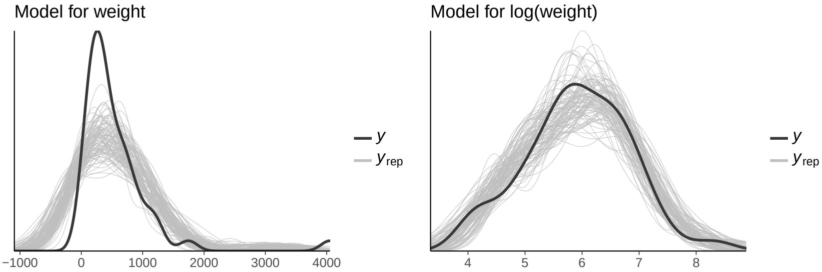
- High Pareto-$\hat{k}$ value indicates the target distribution (LOO posterior) is very different from the proposal distribution (full data posterior)
- This can be caused by
  - well specified, but very flexible model
    - e.g. hierarchical model with one parameter per observation
    - indicated by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

Sometimes cross-validation is not needed

- Posterior predictive checking is often sufficient

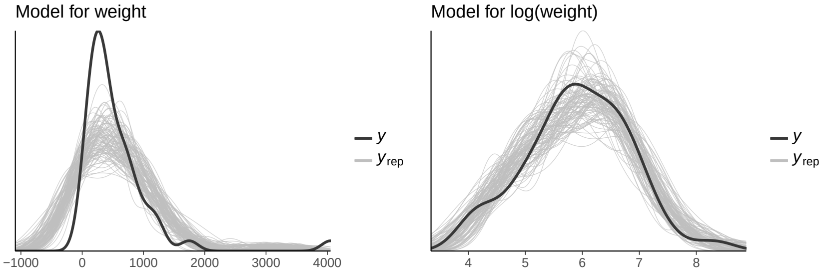


Predicting the yields of mesquite bushes.

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/rssa.12378
- mc-stan.org/bayesplot/articles/graphical-ppcs.html

Model comparison and selection

Next lecture

- Model comparison and selection (elpd_diff, se)
- Related methods (WAIC, *IC, BF)
- Model averaging
- Potential overfitting in model selection