Chapter 7

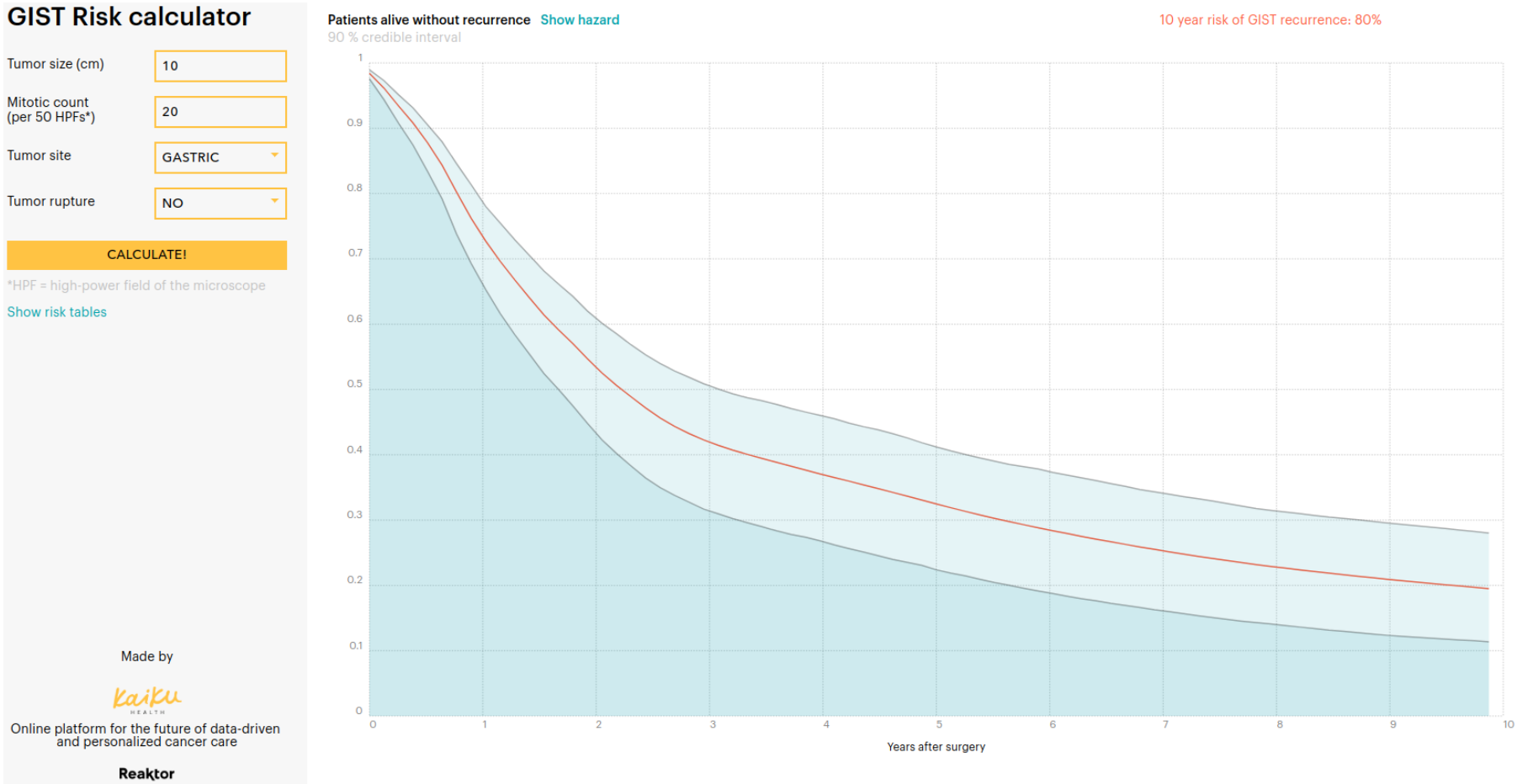
- 7.1 Measures of predictive accuracy
- 7.2 Information criteria and cross-validation
  - Instead of 7.2, read:
    Vehtari, A., Gelman, A., Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and Computing. 27(5):1413–1432. arXiv preprint.
- 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)

Model assessment, selection and inference after selection

- Extra material at https://avehtari.github.io/modelselection/
  - Videos, Slides, Notebooks, References
- CV-FAQ https://avehtari.github.io/modelselection/CV-FAQ.html

Predicting concrete quality

Predicting cancer recurrence



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
- Application specific utility/cost functions are important
  - eg. money, life years, quality adjusted life years, etc.

Predictive performance

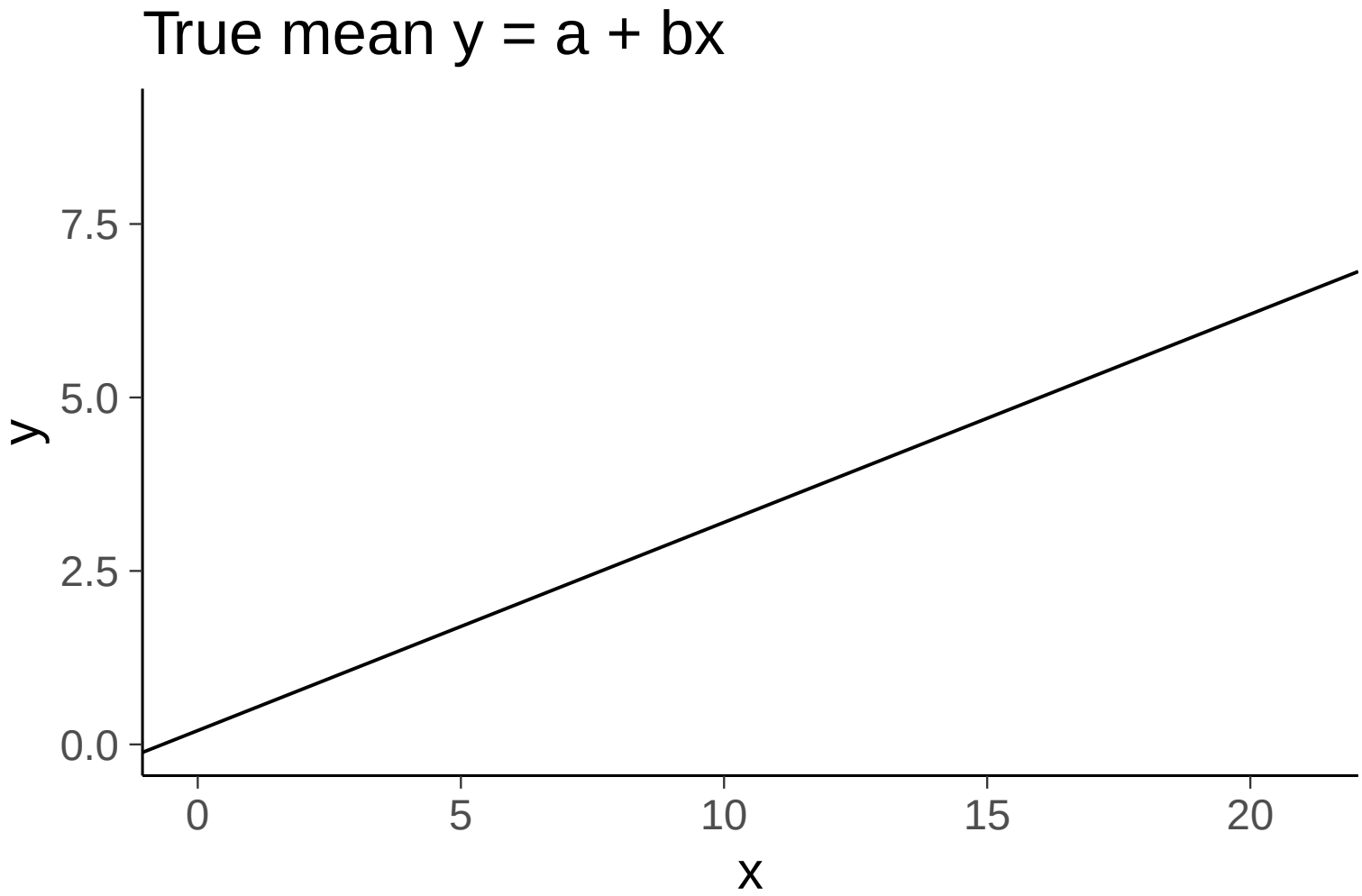
- We need to choose the utility/cost function
- Application specific utility/cost functions are important
  - eg. money, life years, quality adjusted life years, etc.
- If are interested overall in the goodness of the predictive distribution, or we don't know (yet) the application specific utility, then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

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

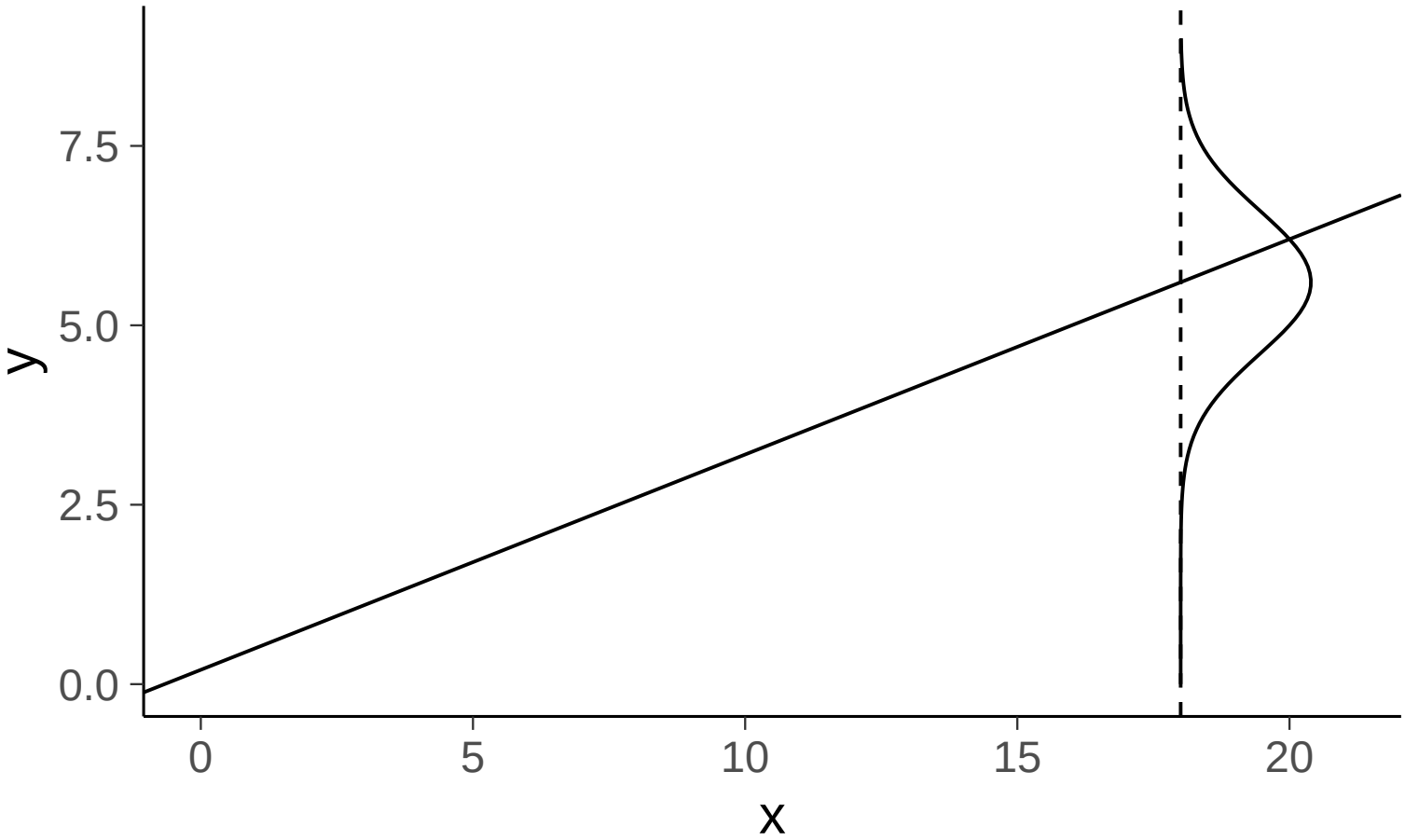
- What is cross-validation
  - Leave-one-out cross-validation (elpd_loo, p_loo)
  - Uncertainty in LOO (SE)
- When is cross-validation applicable?
  - data generating mechanisms and prediction tasks
  - leave-many-out cross-validation
- Fast cross-validation
  - PSIS and diagnostics in loo package (Pareto k, n_eff, Monte Carlo SE)
  - K-fold cross-validation
- Related methods (WAIC, *IC, BF)
- Model comparison and selection (elpd_diff, se)
- Model averaging with Bayesian stacking

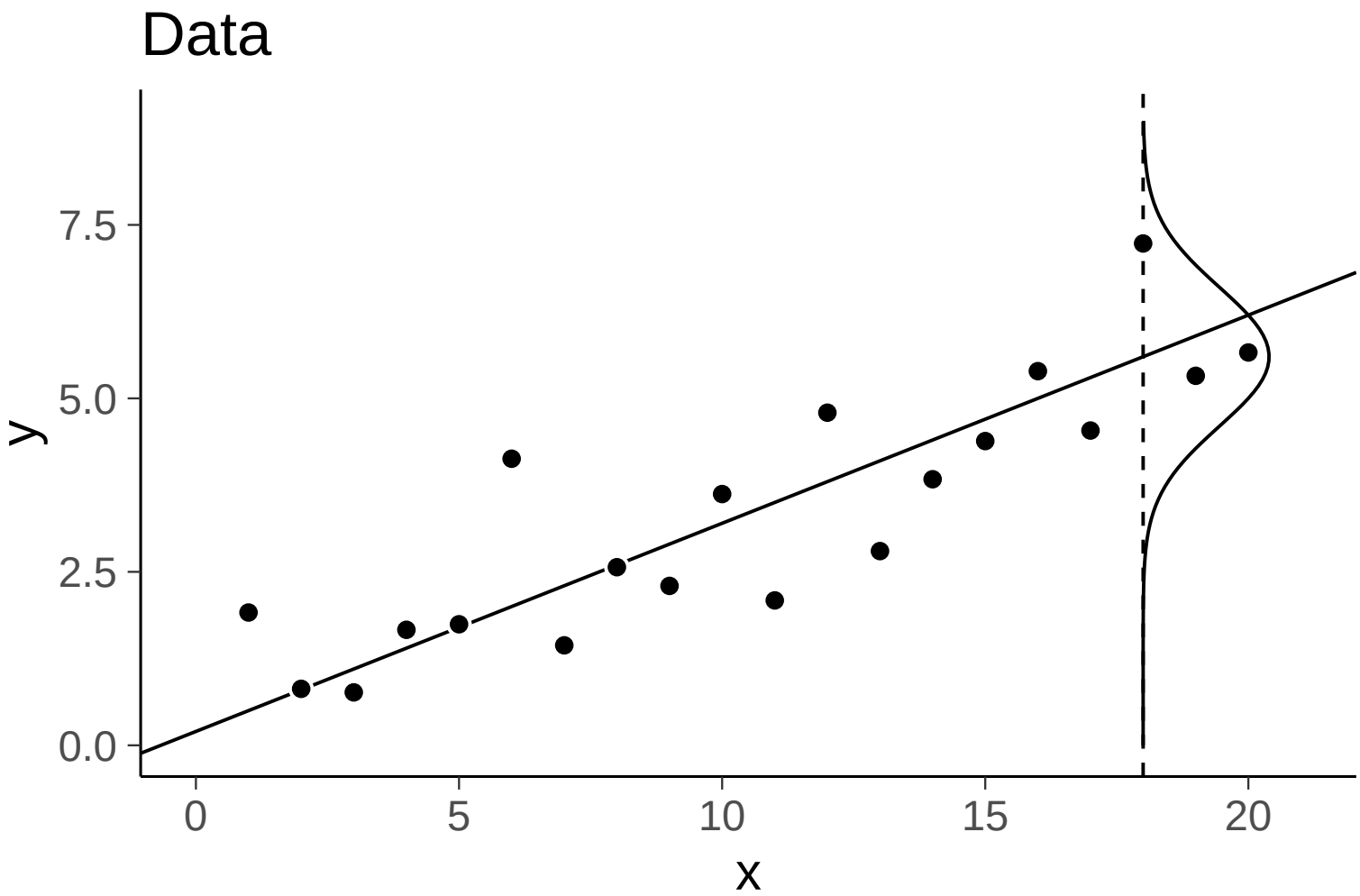
Stan and `loo` package

```
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
```

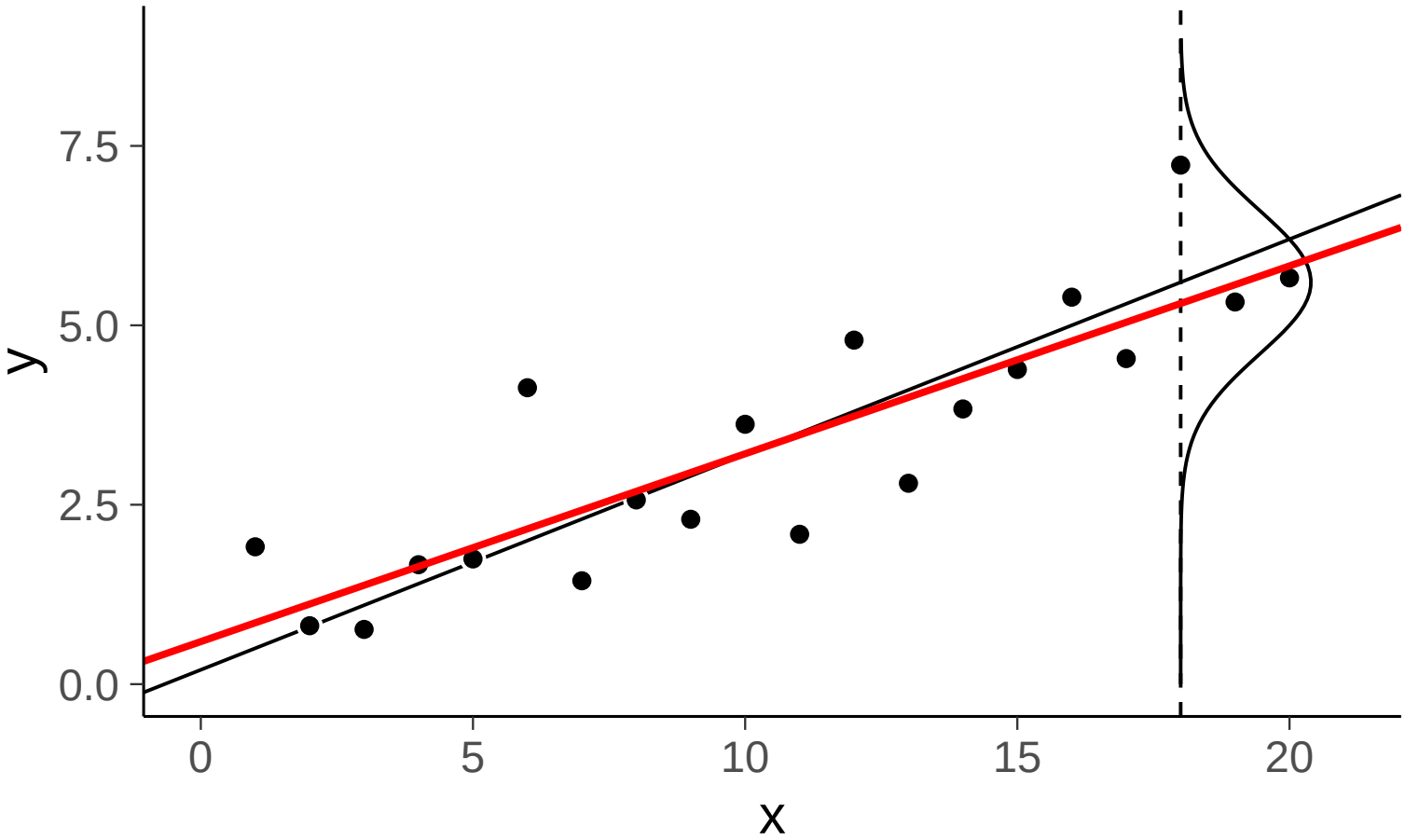

True mean y = a + bx
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

True mean and sigma

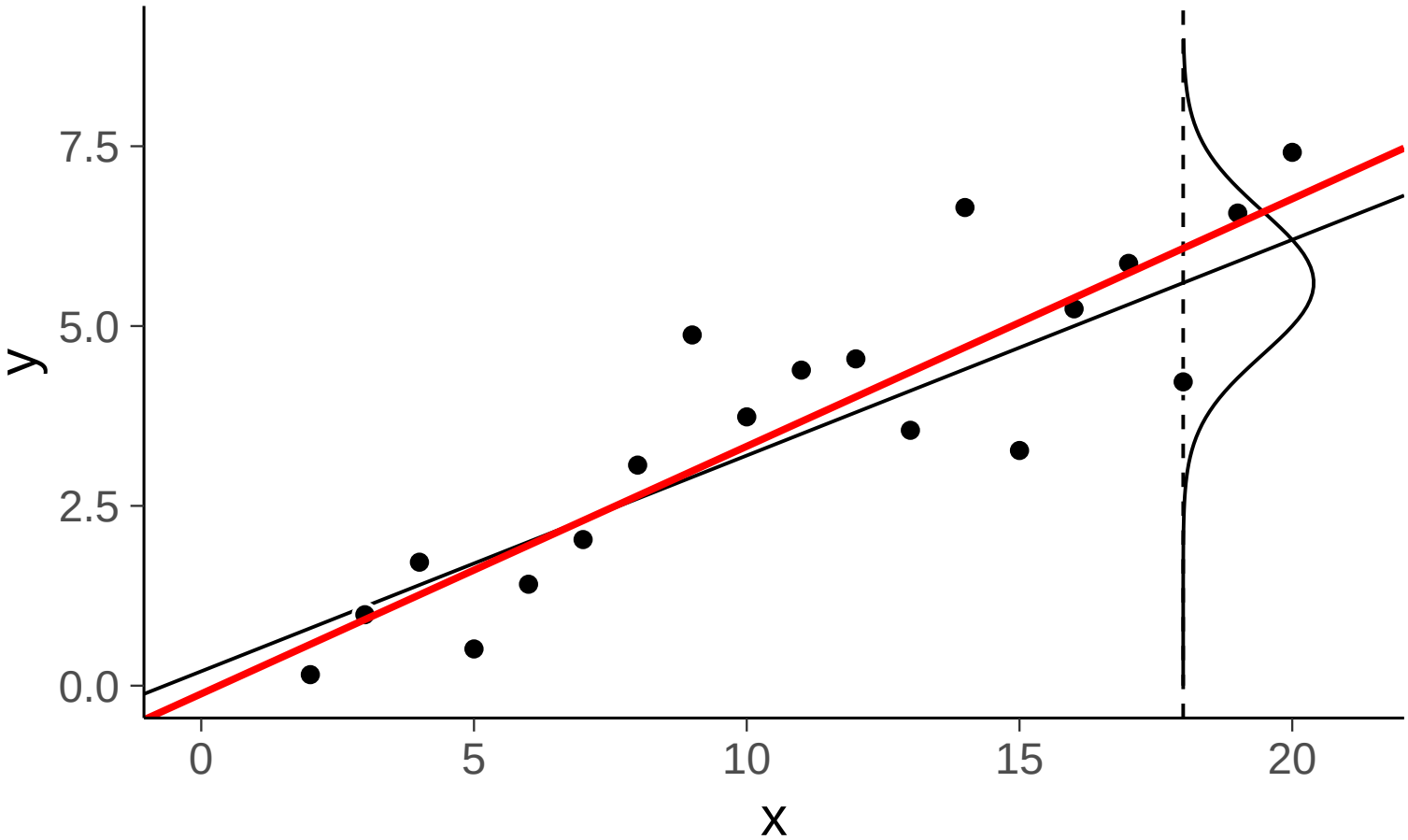



Data
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

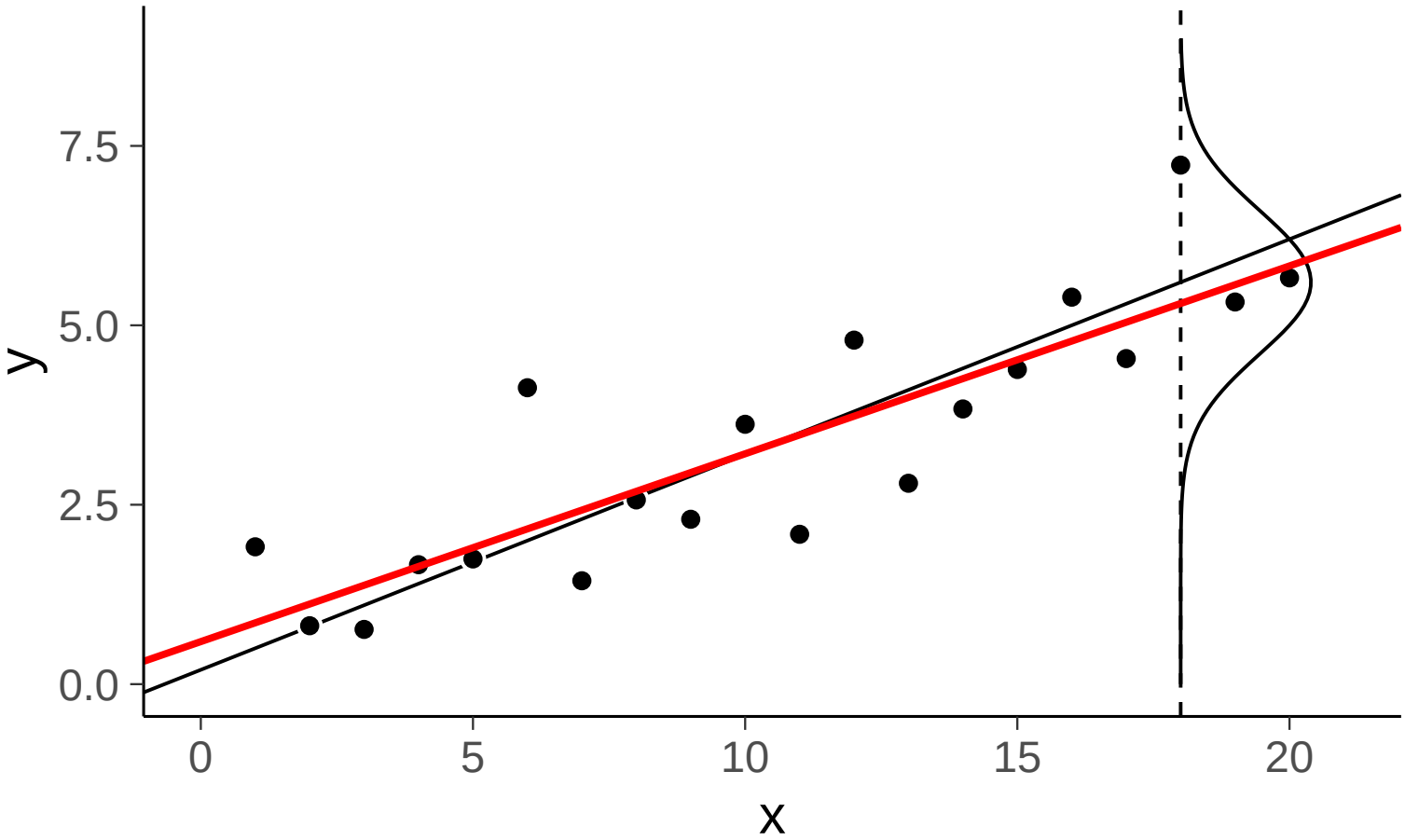
Posterior mean



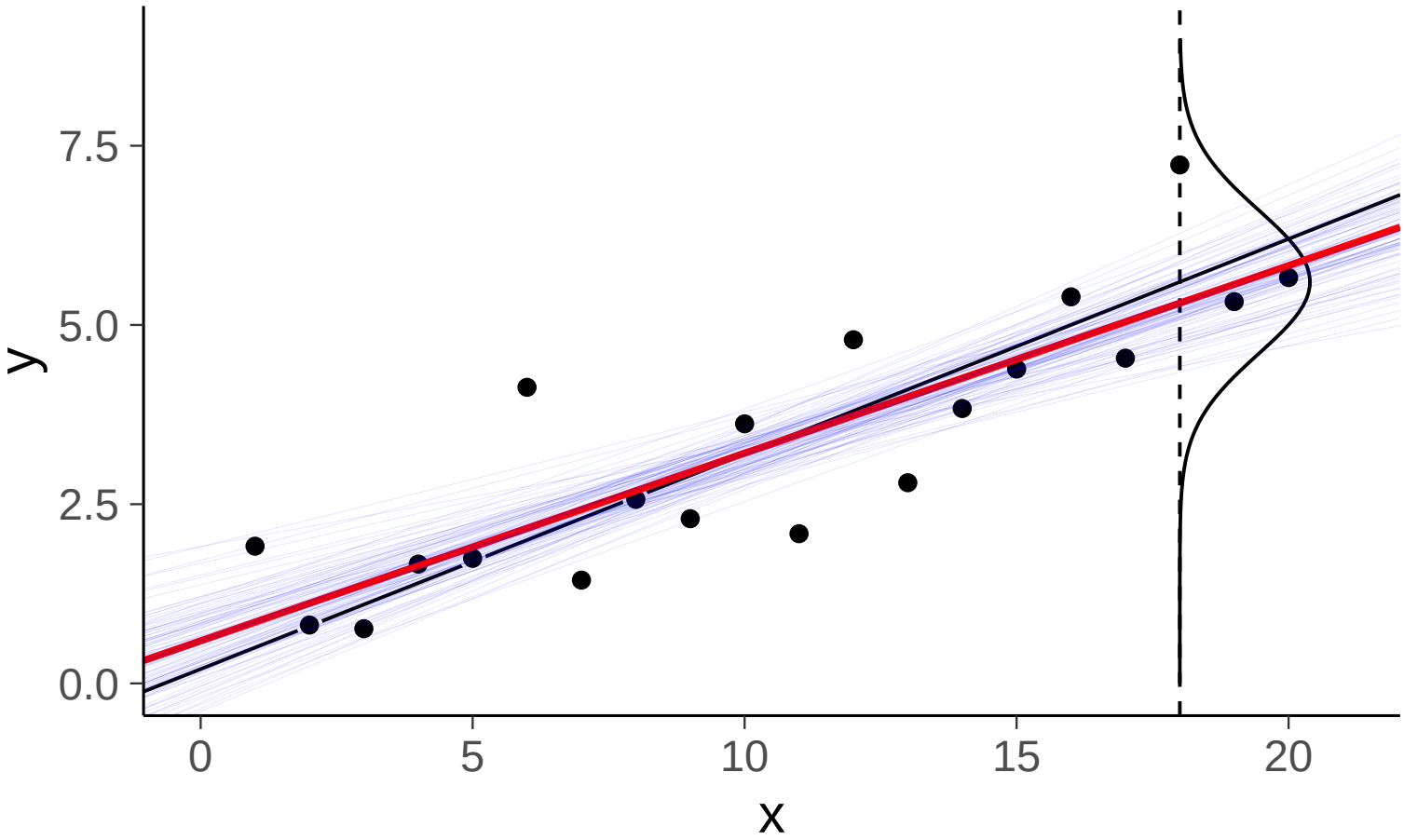
Posterior mean, alternative data realisation



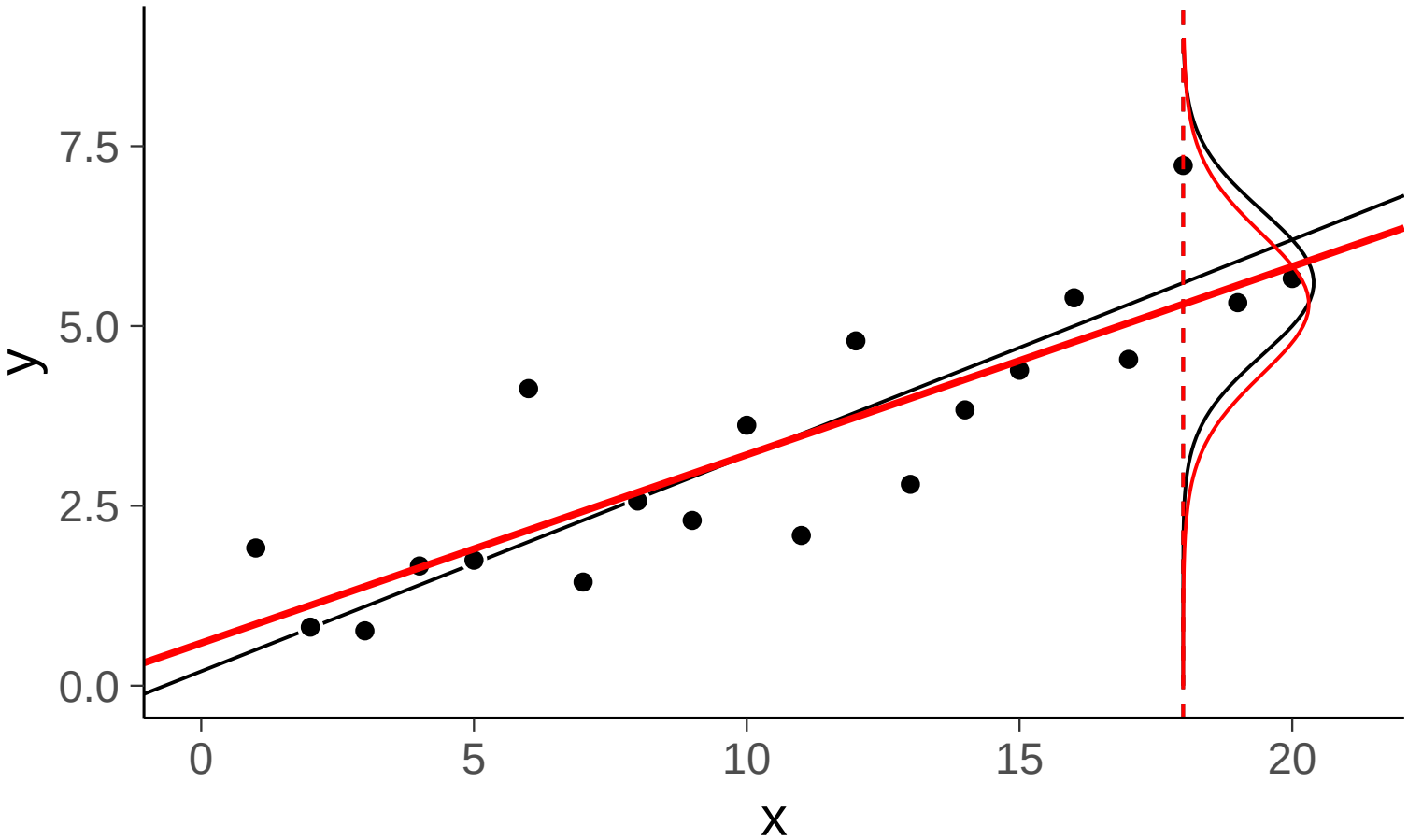
Posterior mean



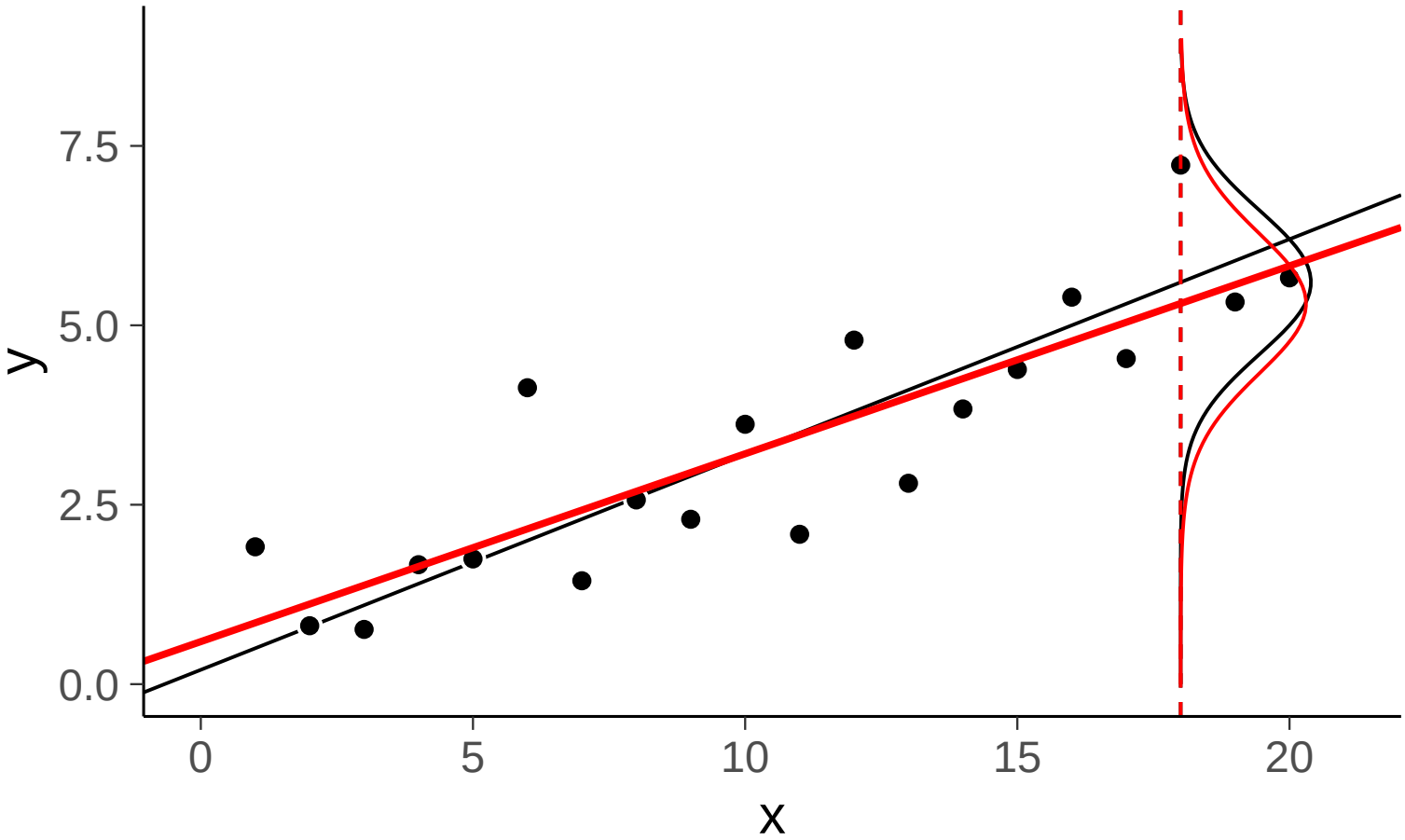
Posterior draws



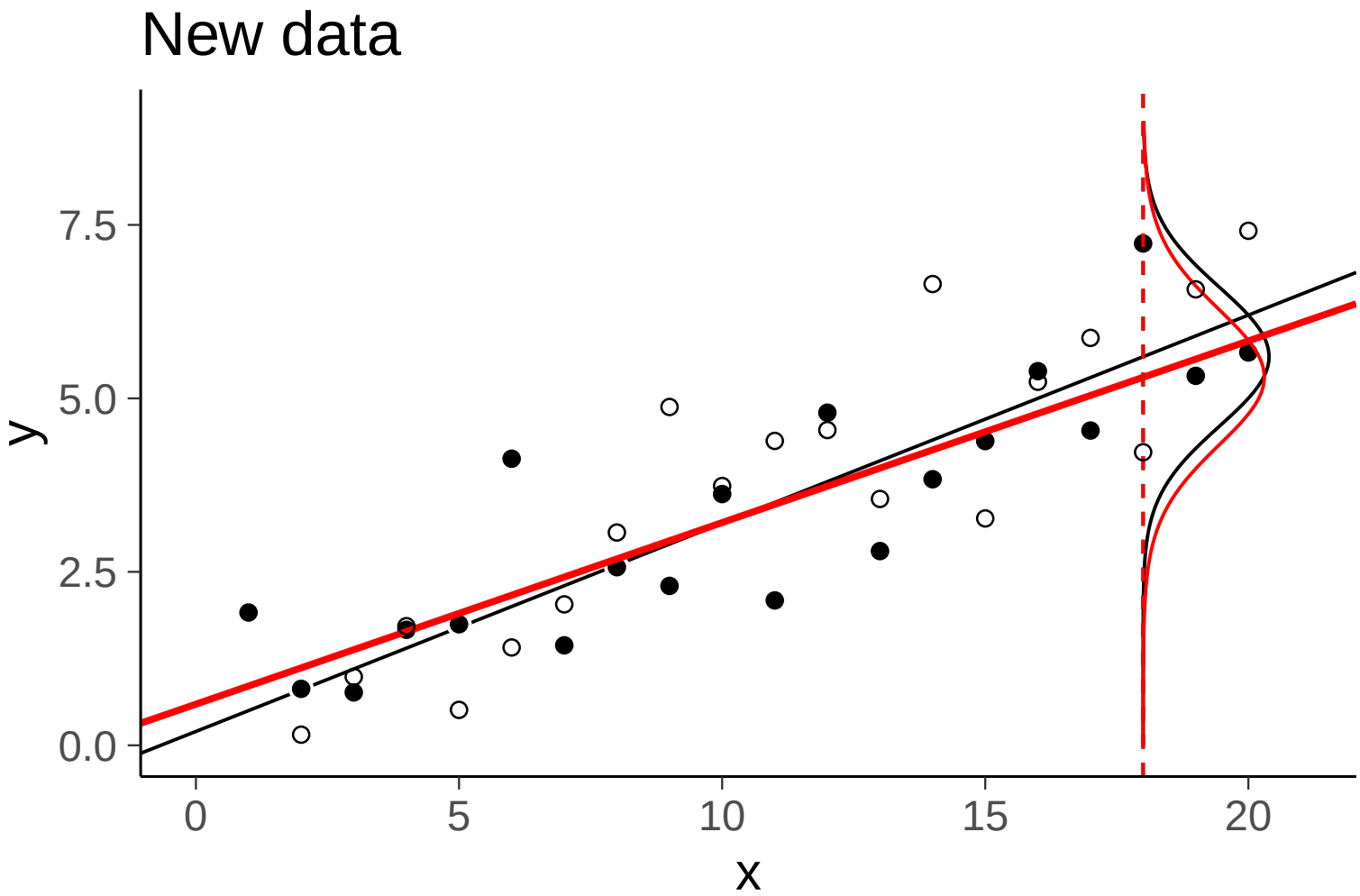
Posterior predictive distribution



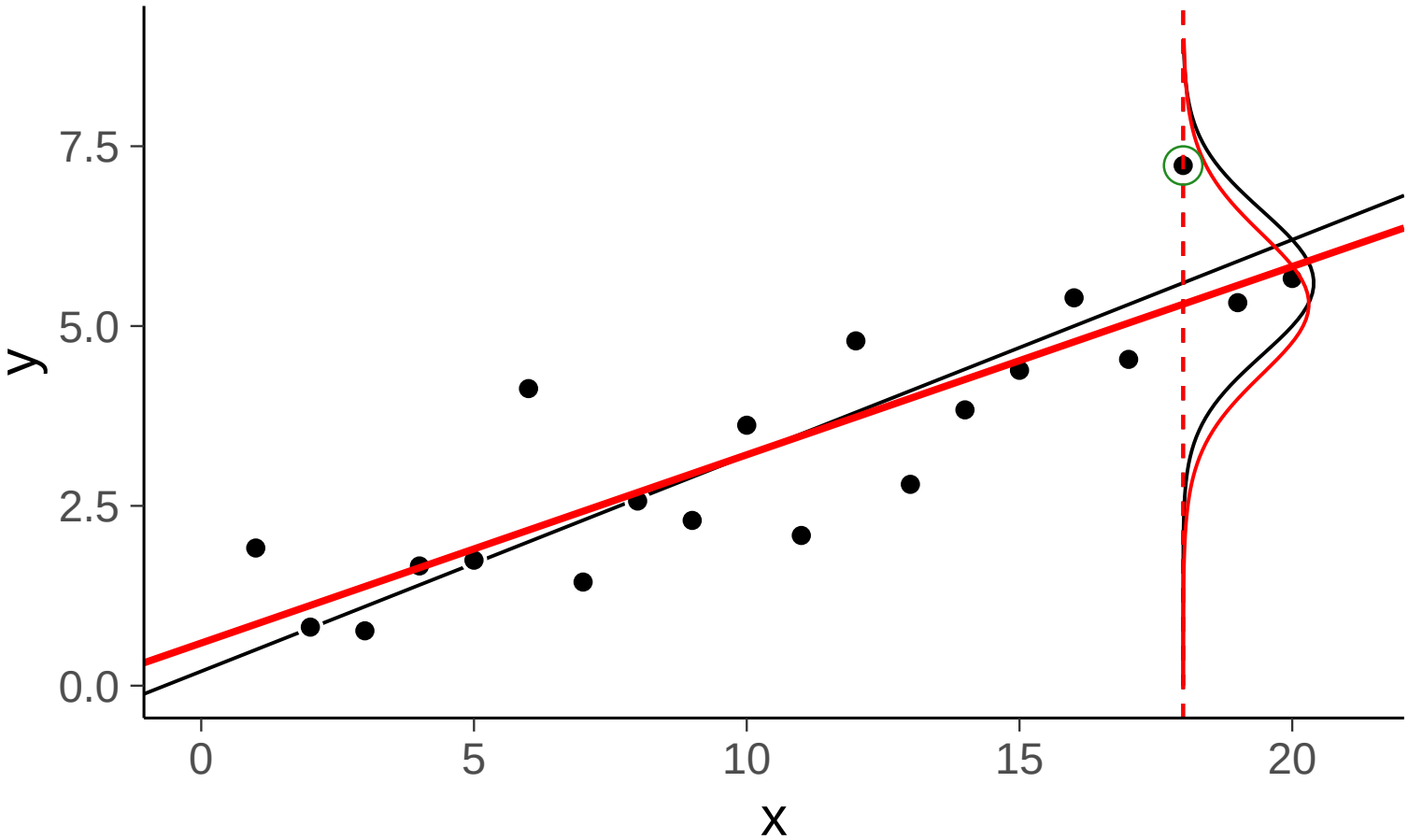
Posterior predictive distribution



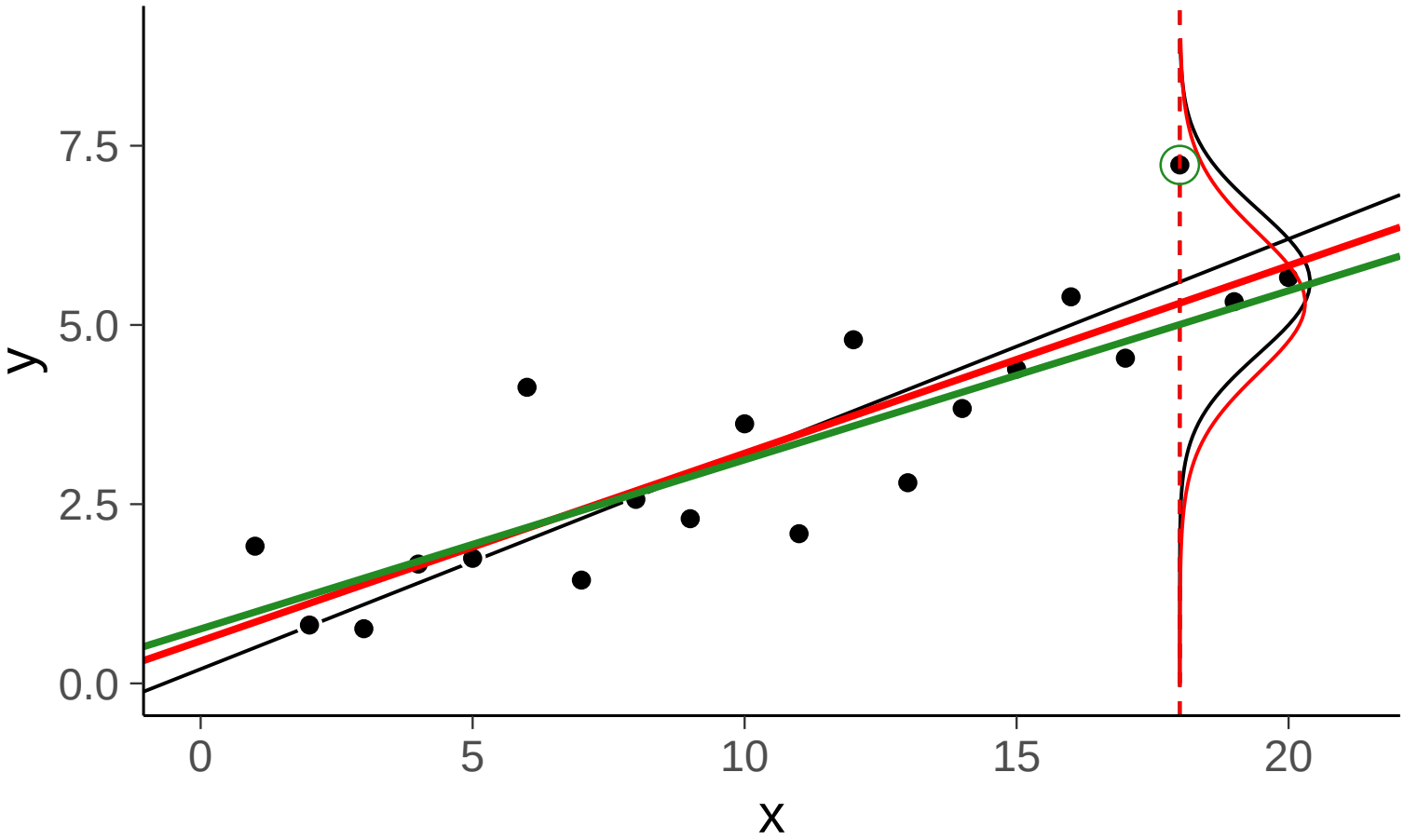
$$p(\tilde{y} \mid \tilde{x} = 18, x, y) = \int p(\tilde{y} \mid \tilde{x} = 18, \theta) p(\theta \mid x, y) d\theta$$


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

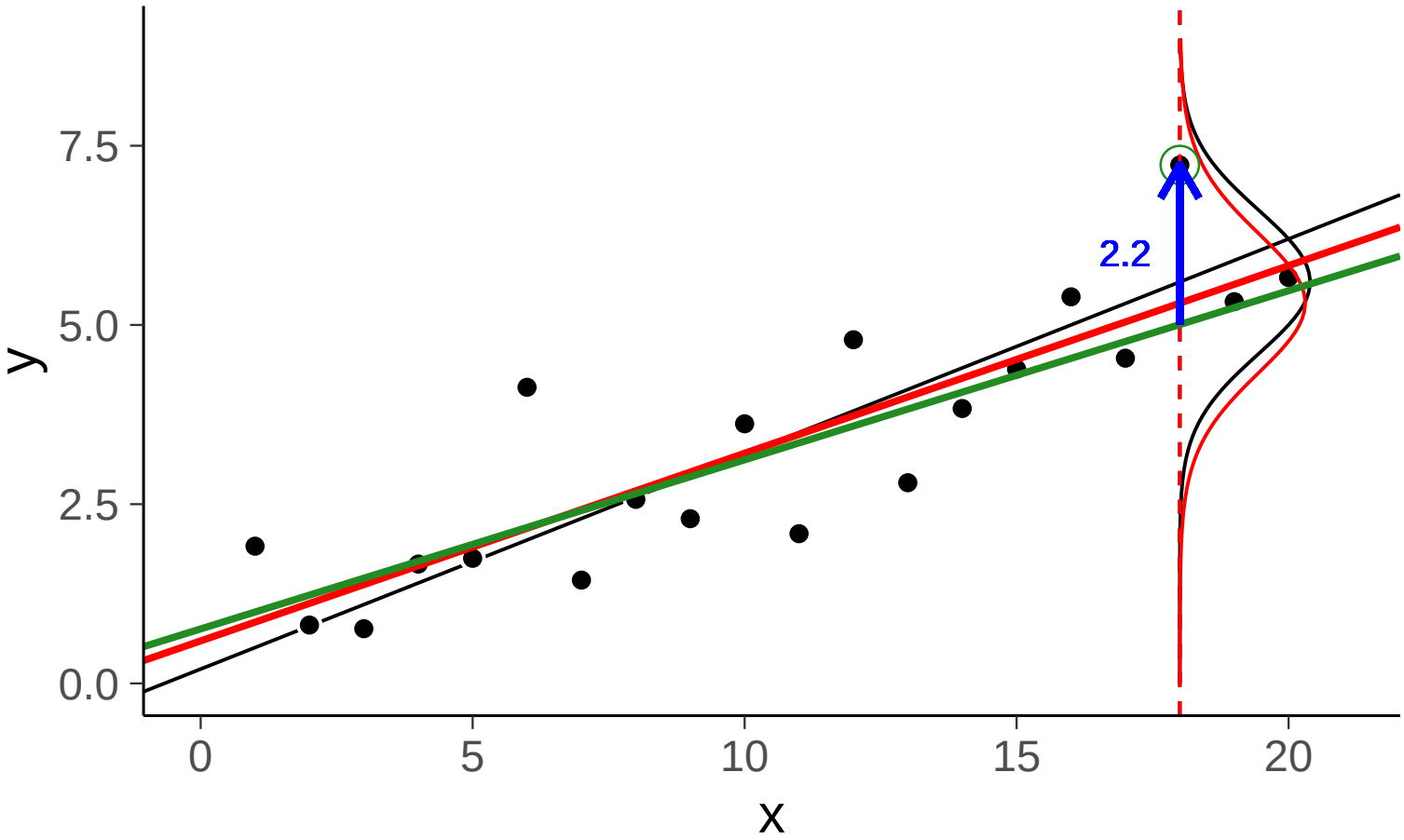
Posterior predictive distribution



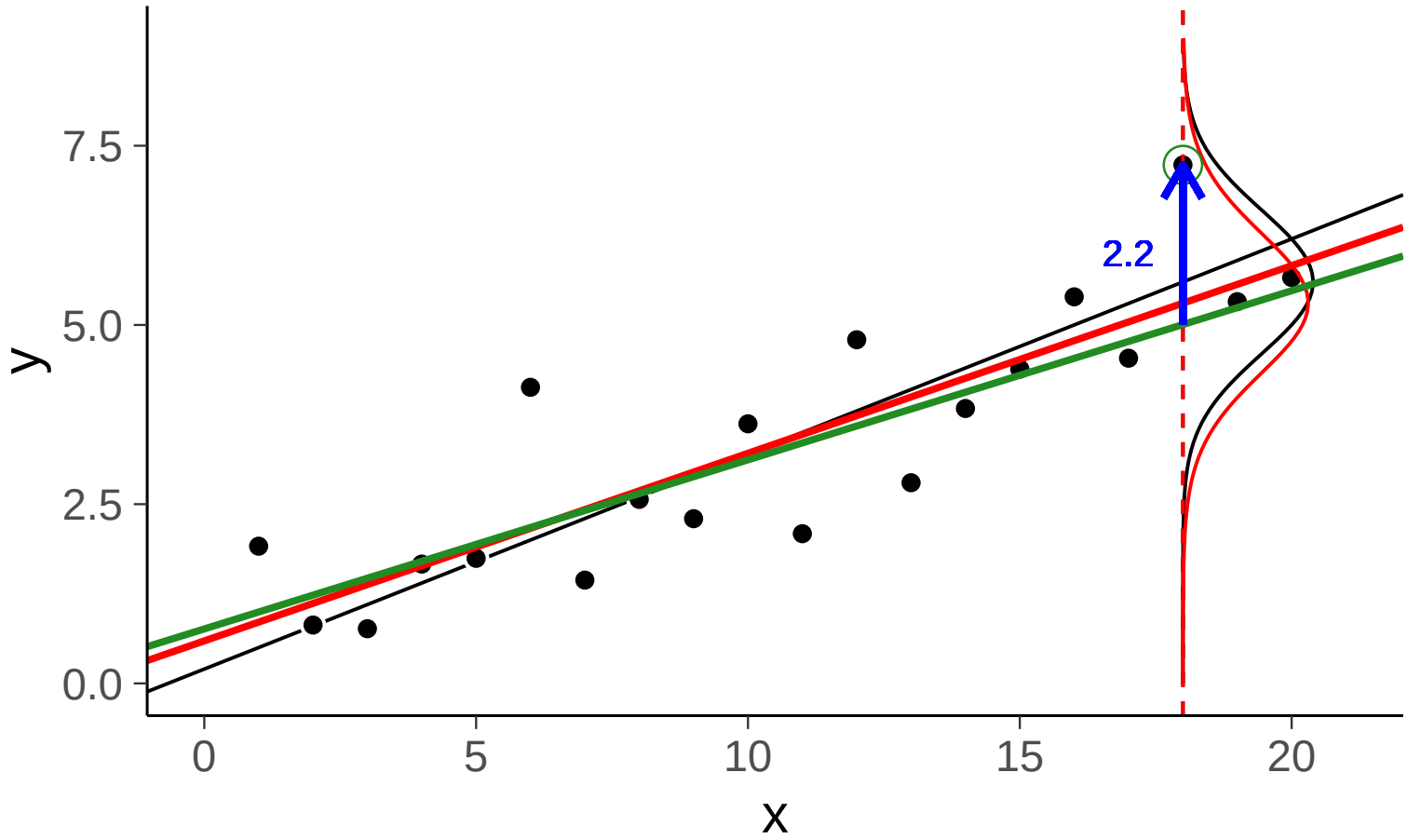
Leave−one−out mean



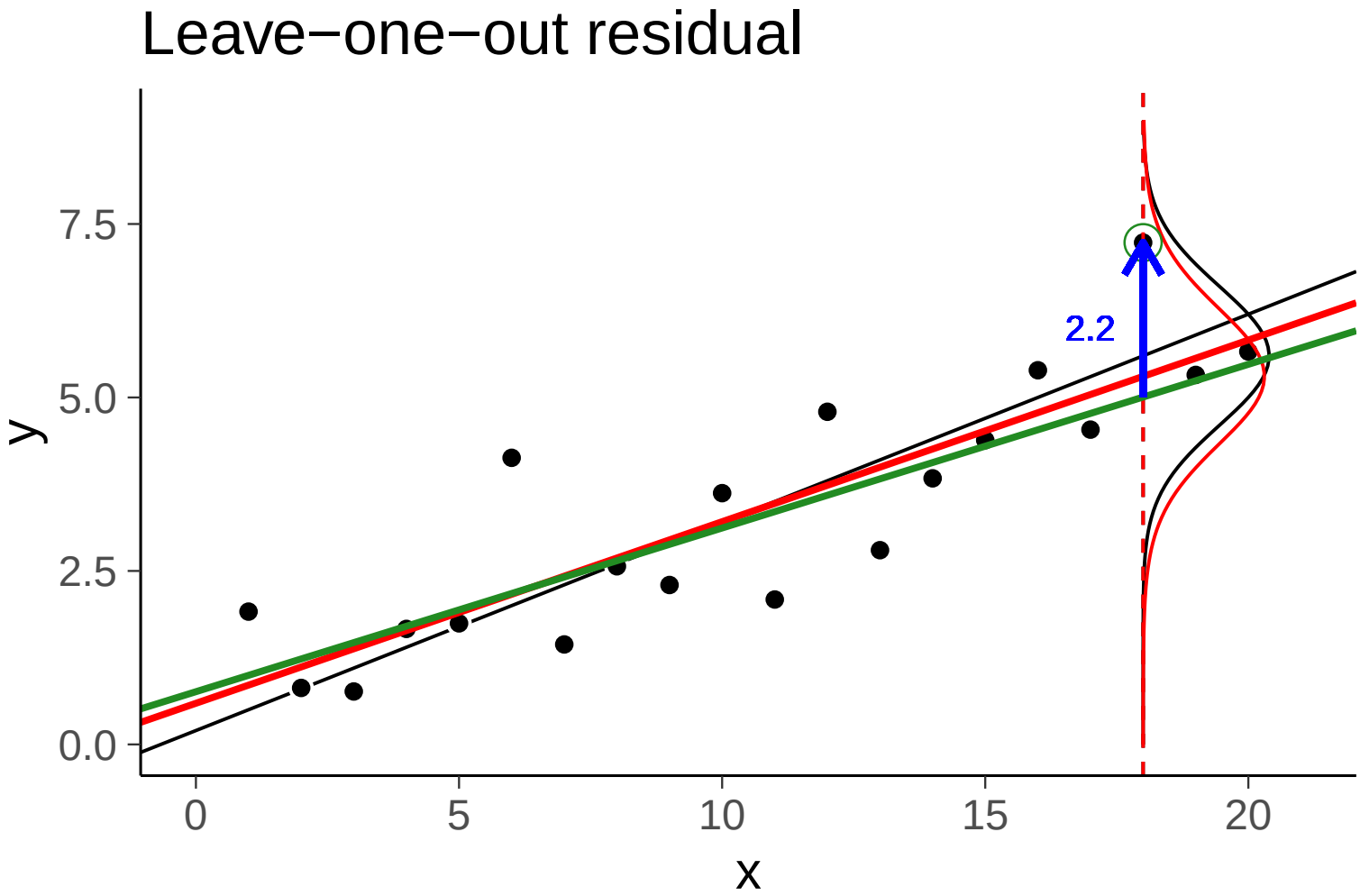
Leave−one−out residual



Leave−one−out residual

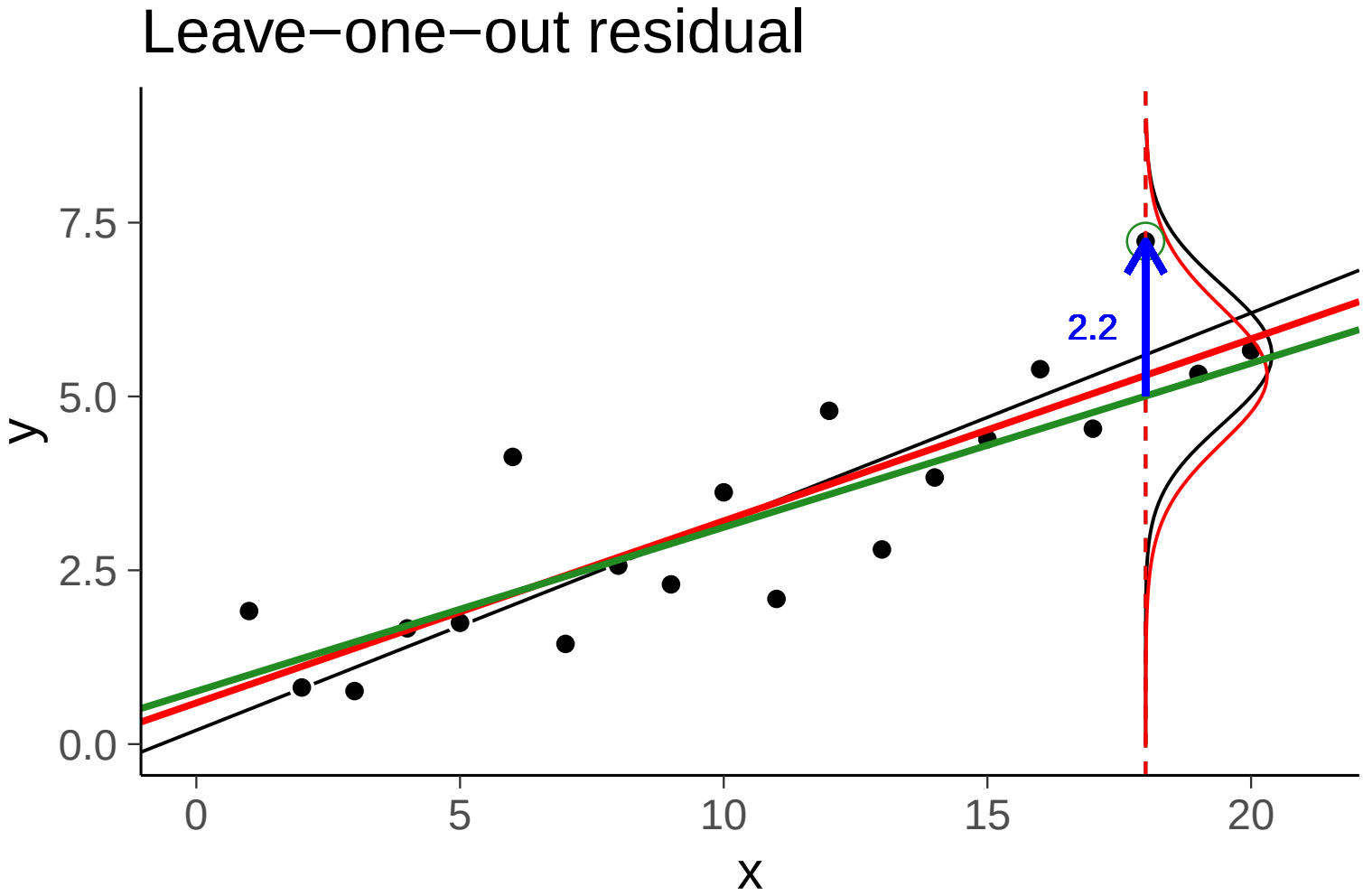


$y_{18} - E[p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18})]$



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

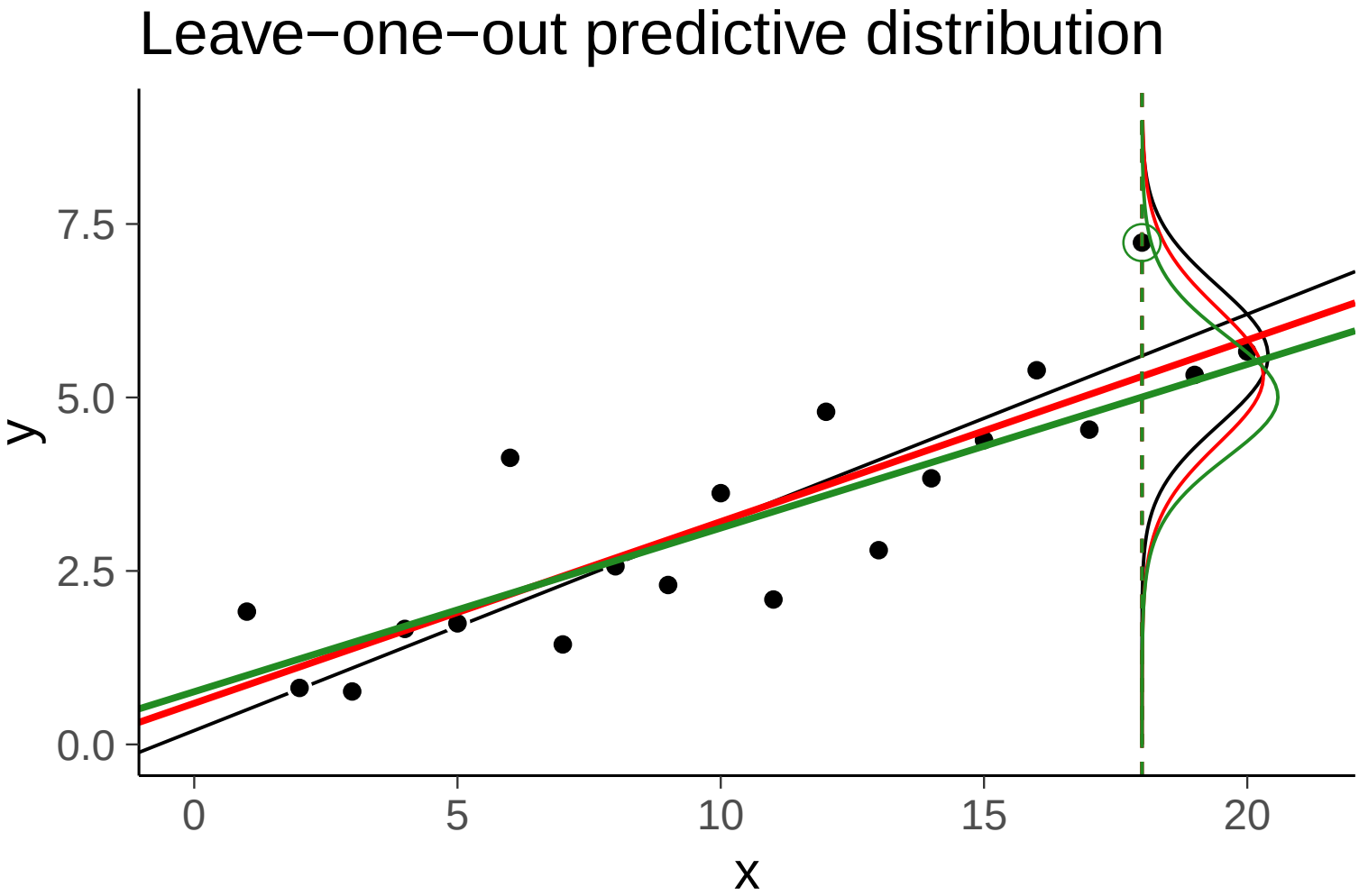
Can be use to compute, e.g., RMSE, R^2, 90% error



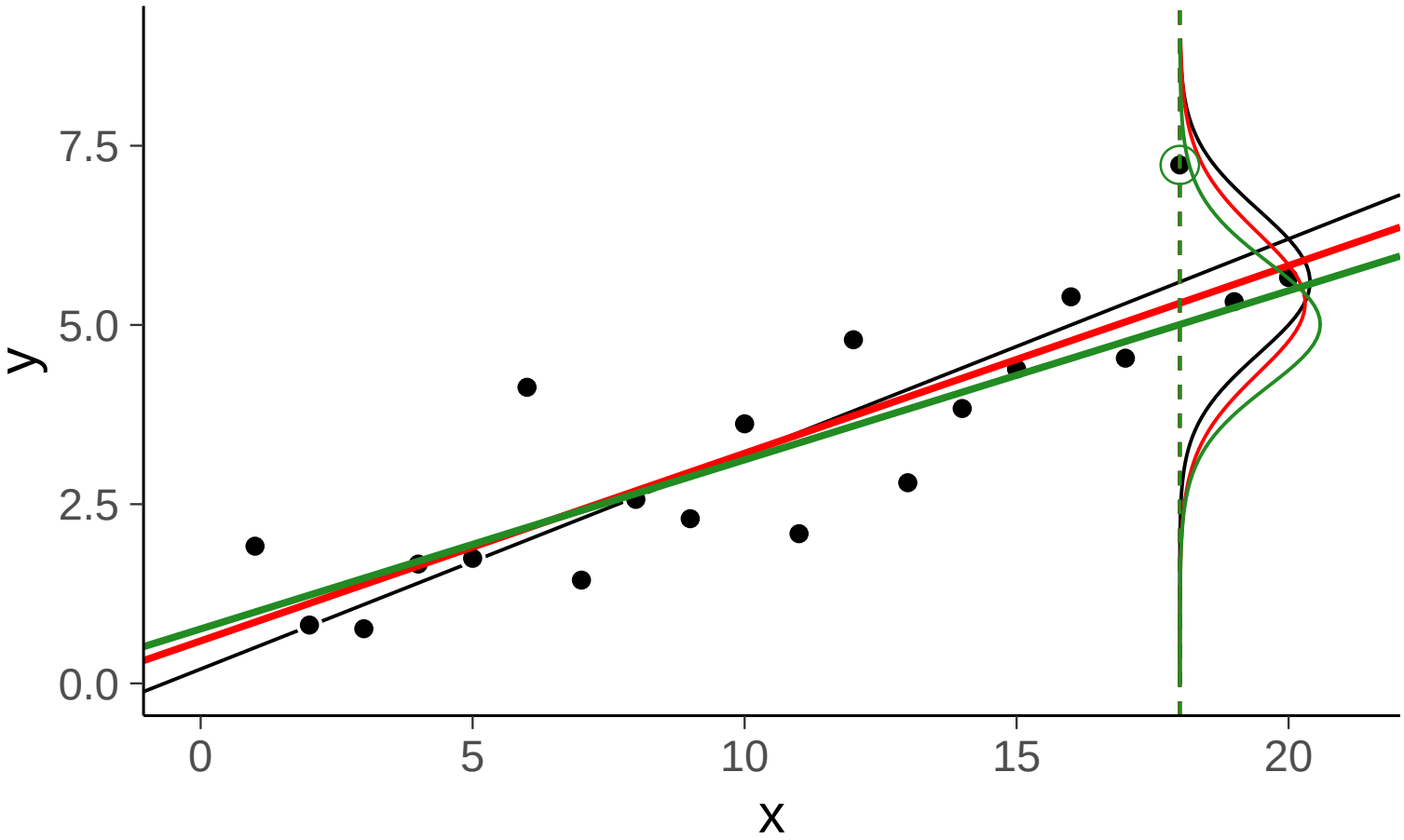
$y_{18} - E[p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18})]$

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

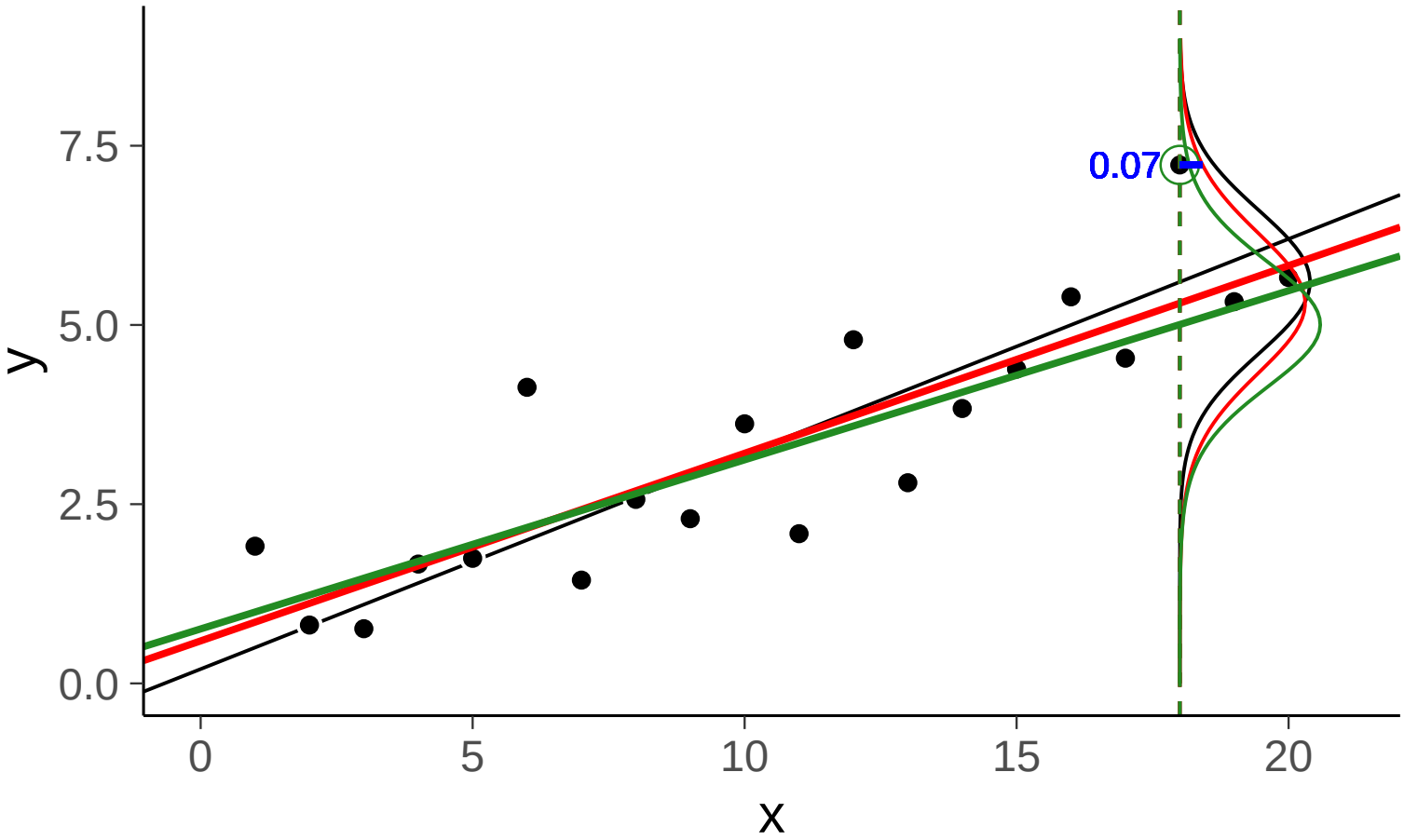

Leave−one−out predictive distribution
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

Leave−one−out predictive distribution

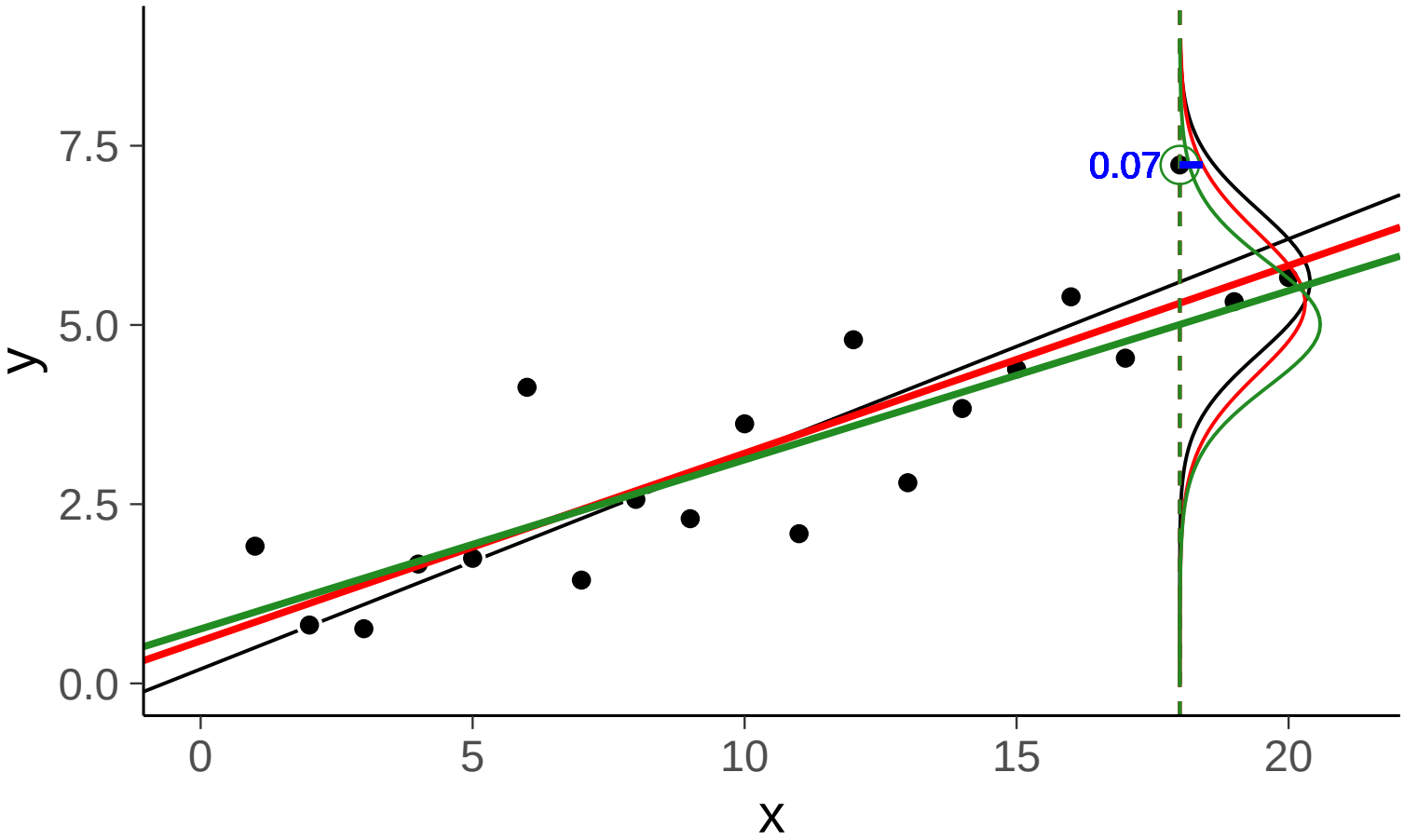


$$p(\tilde{y} \mid \tilde{x} = 18, x_{-18}, y_{-18}) = \int p(\tilde{y} \mid \tilde{x} = 18, \theta) p(\theta \mid x_{-18}, y_{-18}) d\theta$$

Posterior predictive density

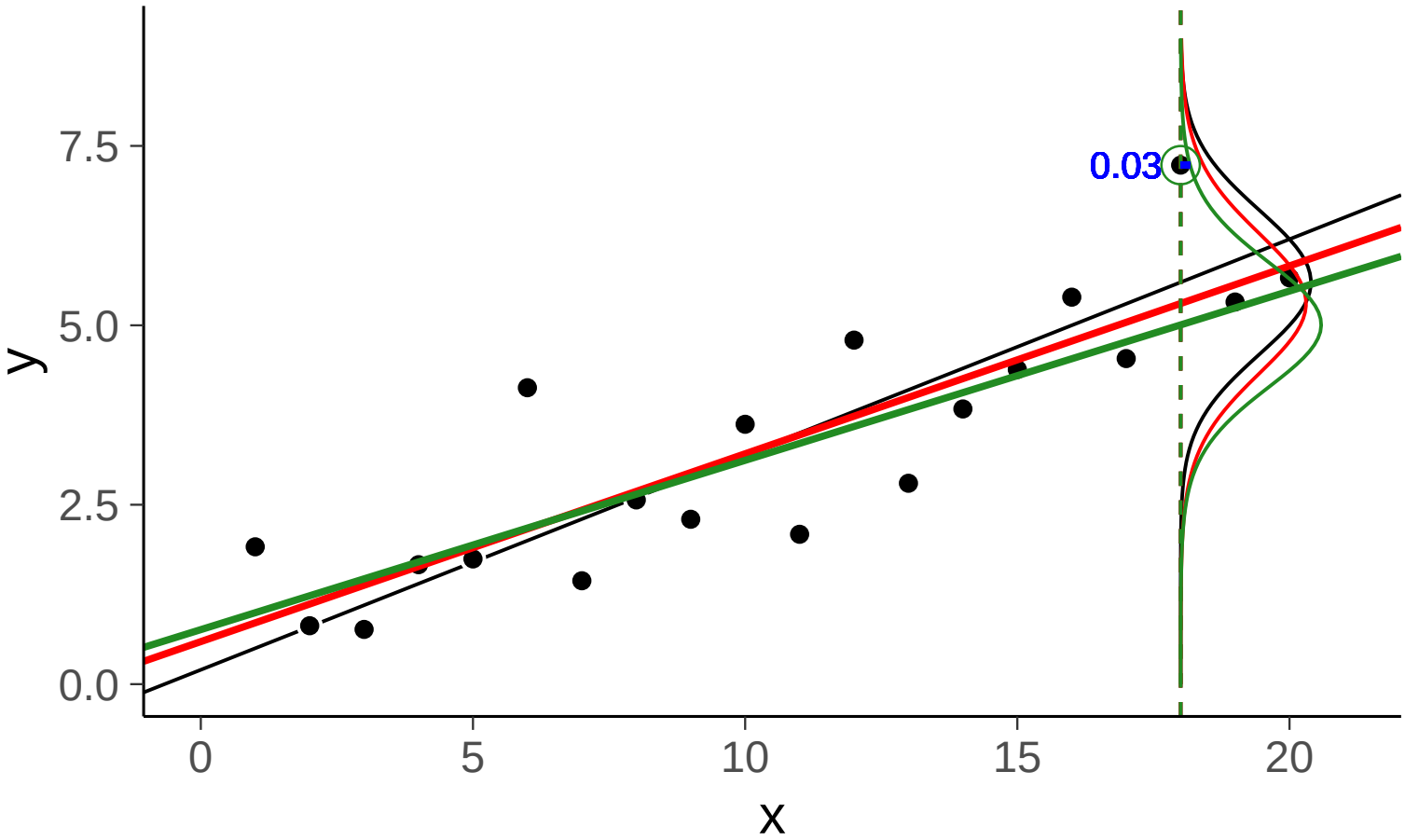


Posterior predictive density



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

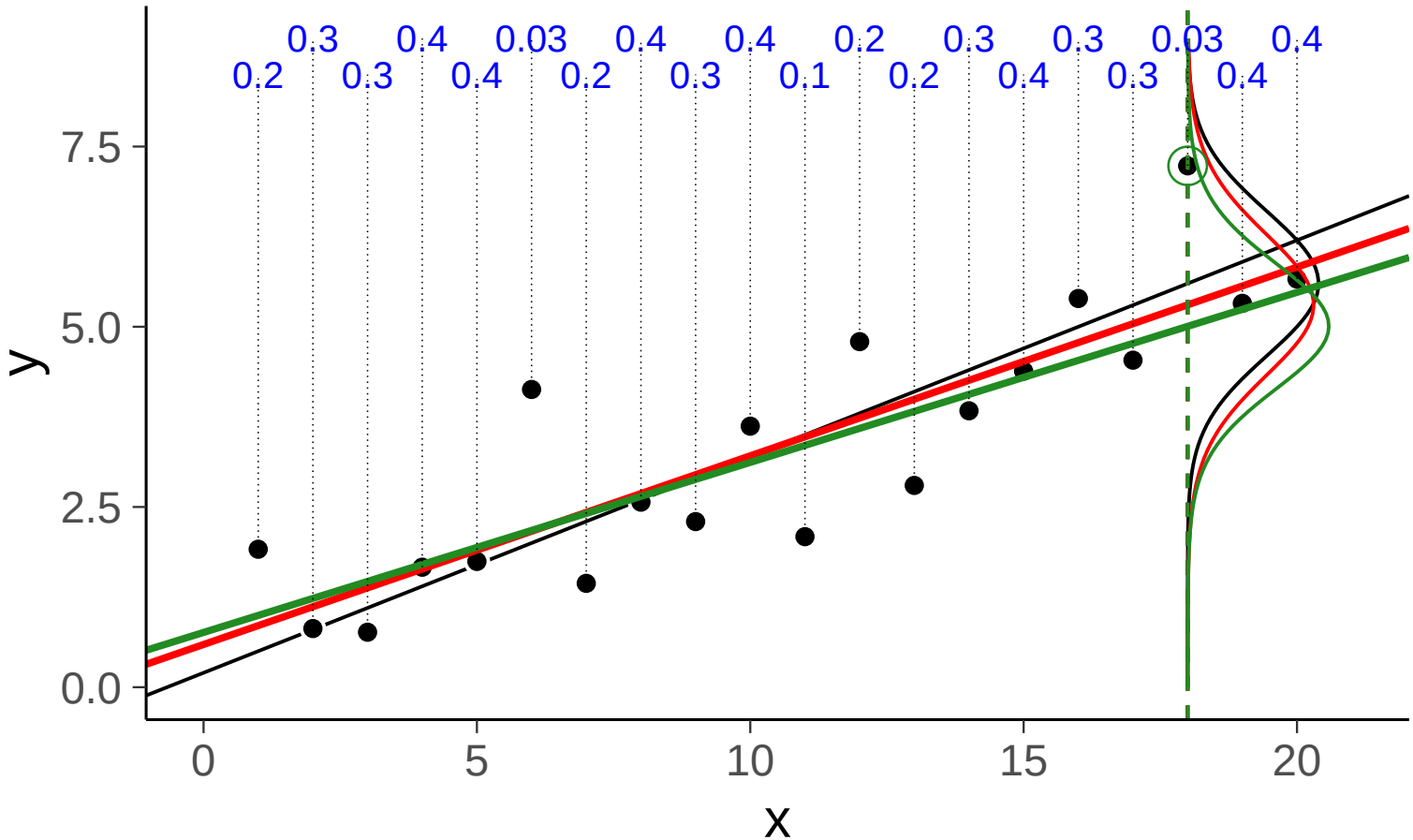
Leave−one−out predictive density



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

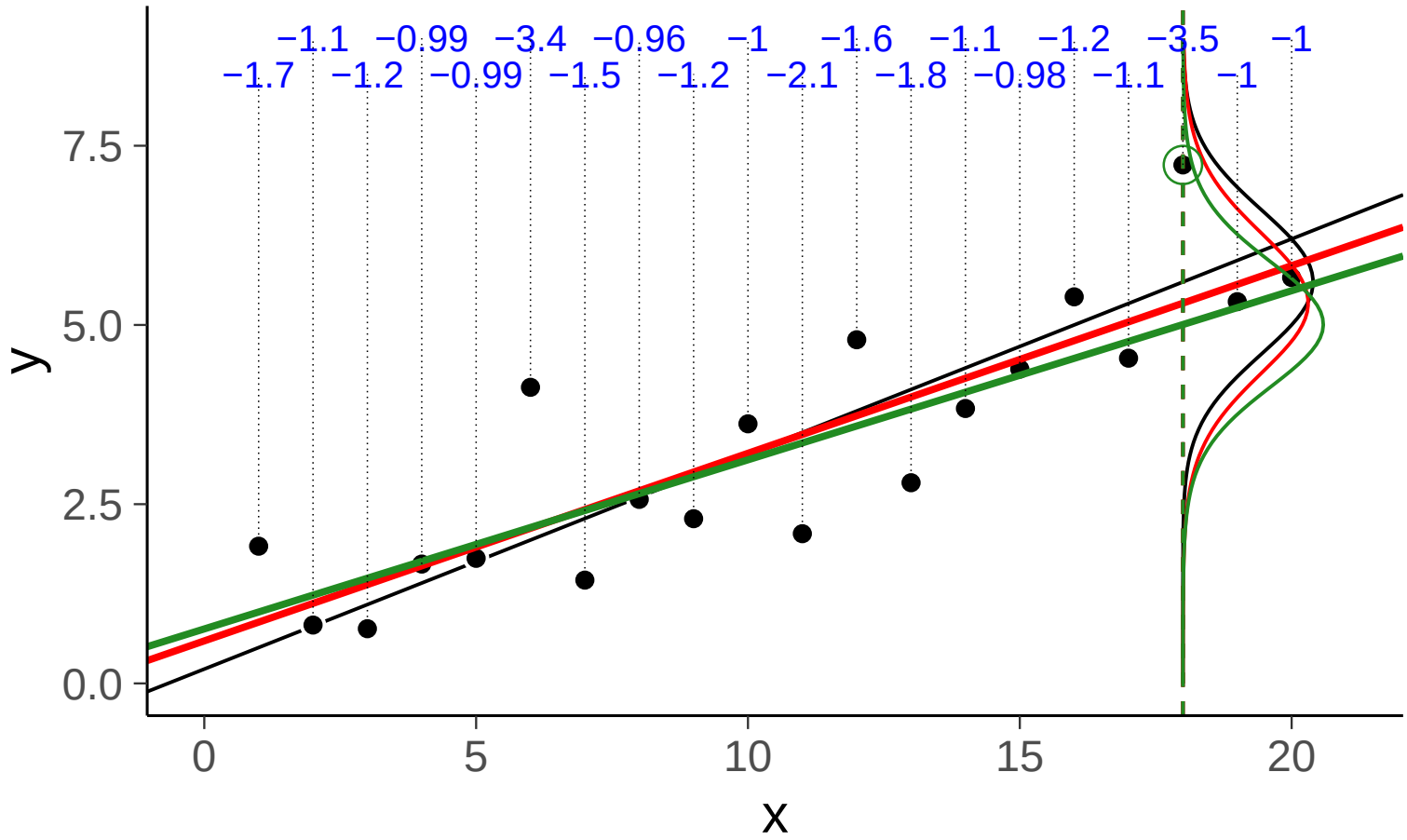
$p(\tilde{y} = y_{18} \mid \tilde{x} = 18, x_{-18}, y_{-18}) \approx 0.03$

Leave−one−out predictive densities



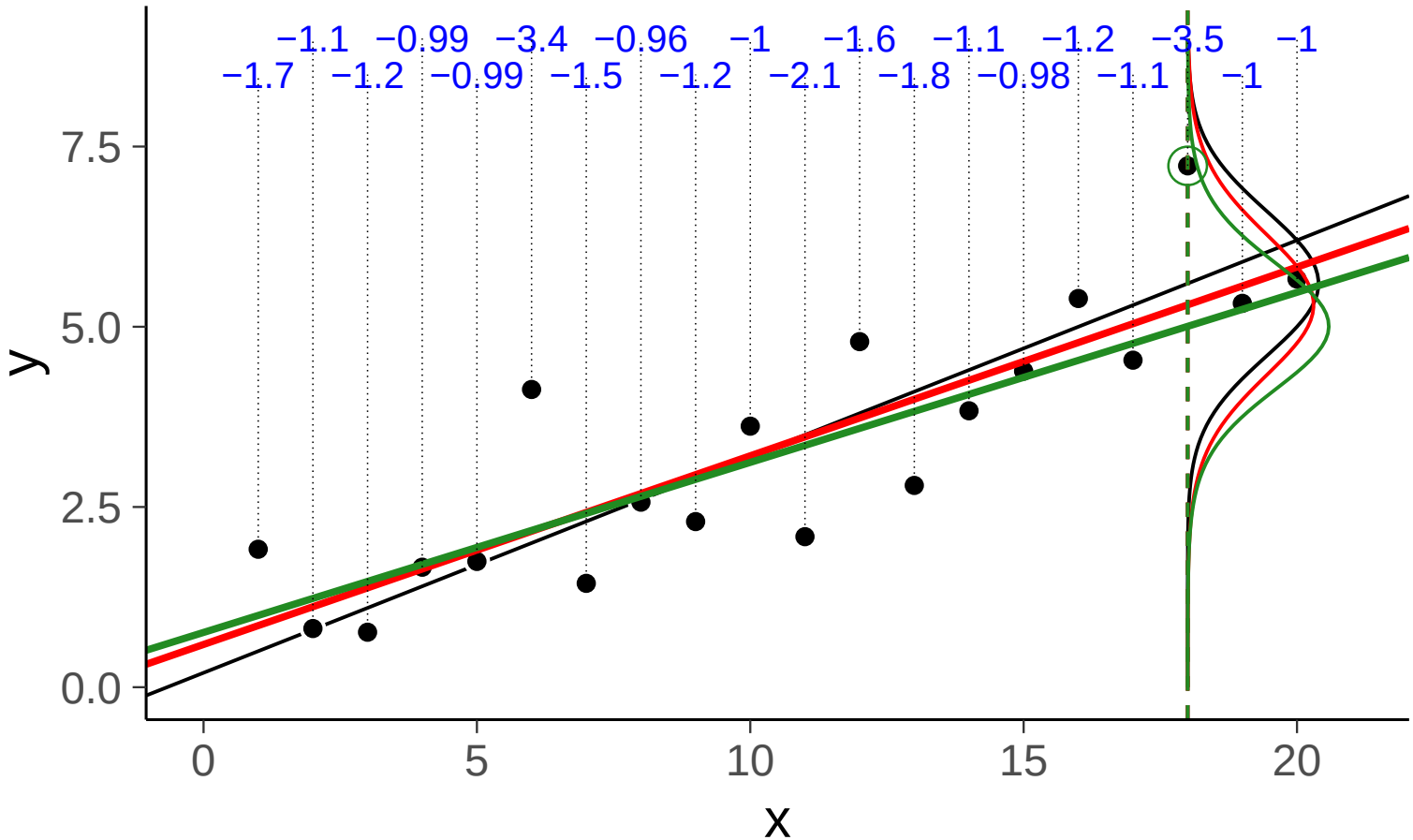
$p(y_i \mid x_i, x_{-i}, y_{-i}), \quad i = 1, \ldots, 20$

Leave−one−out log predictive densities



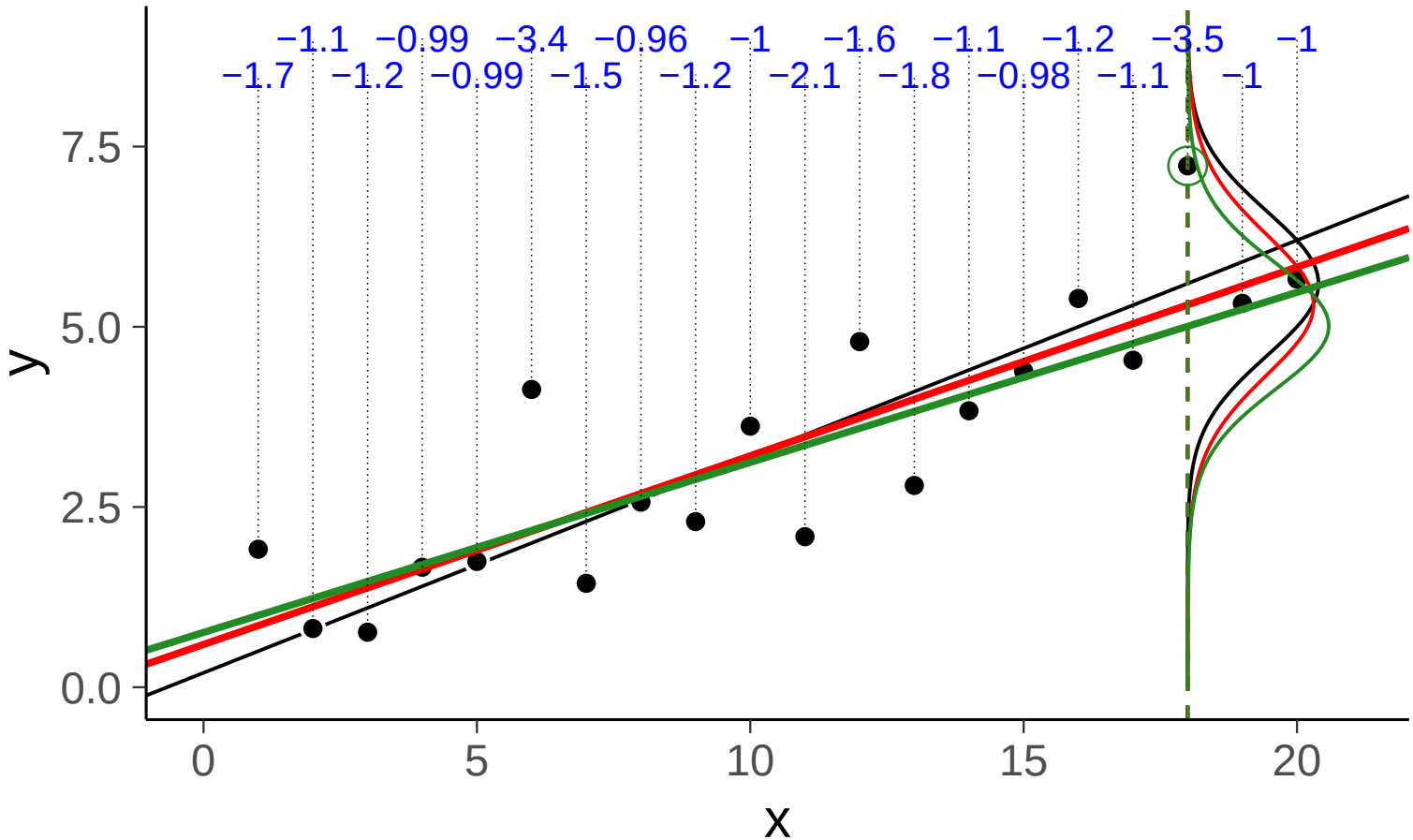
$\log p(y_i \mid x_i, x_{-i}, y_{-i}), \quad i = 1, \ldots, 20$

Leave−one−out log predictive densities



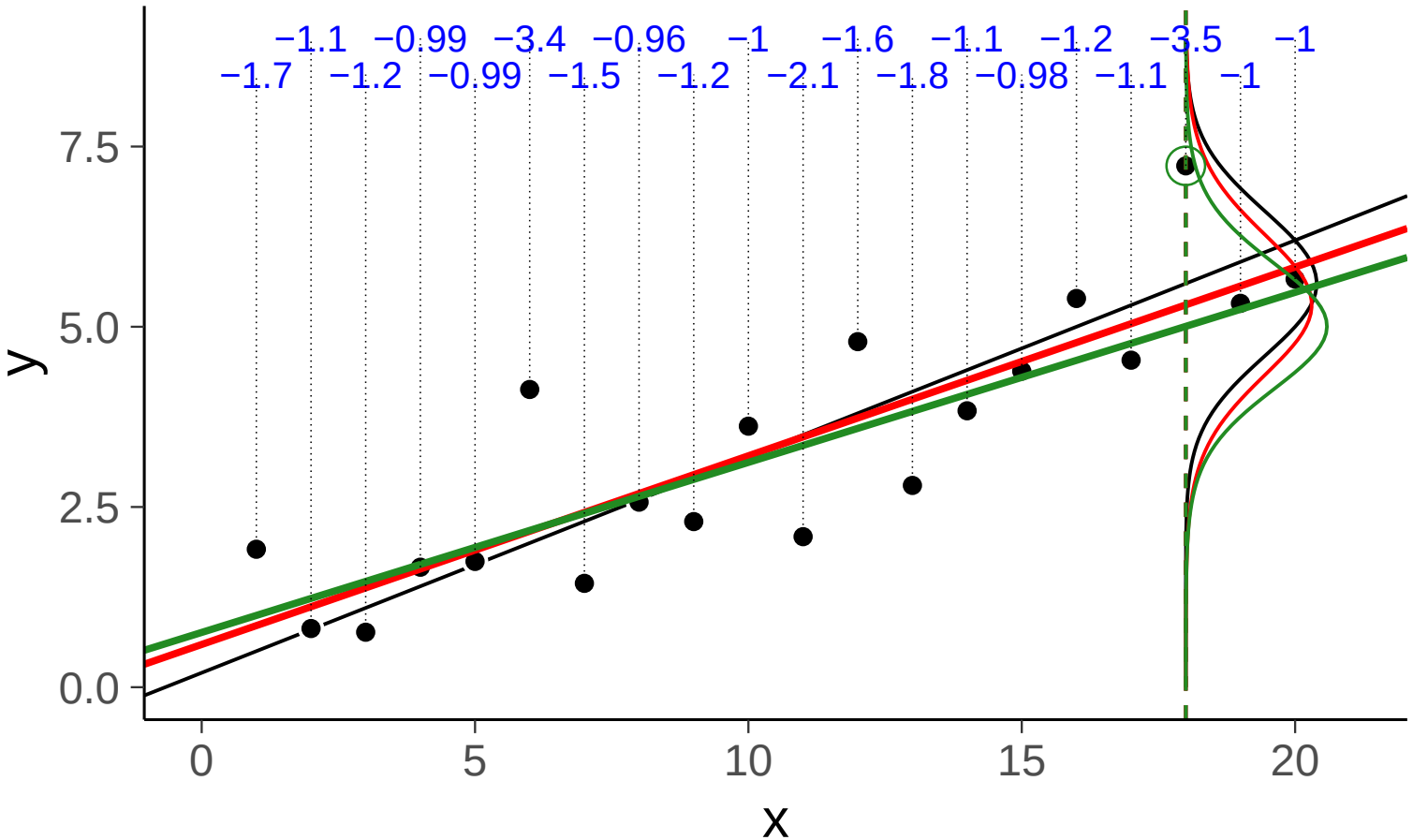
$\sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

Leave−one−out log predictive densities



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

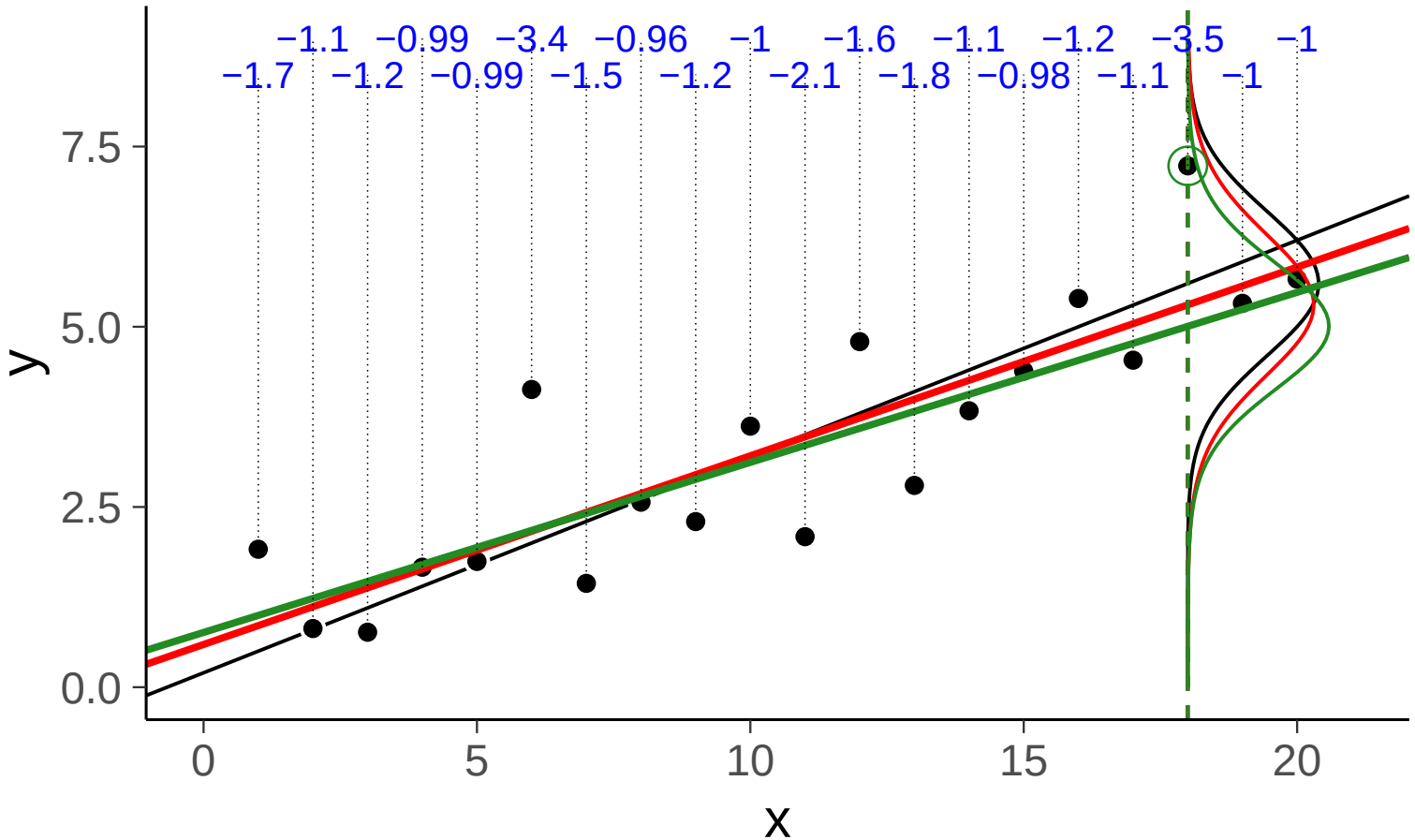
Leave−one−out log predictive densities



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

an estimate of log posterior pred. density for new data

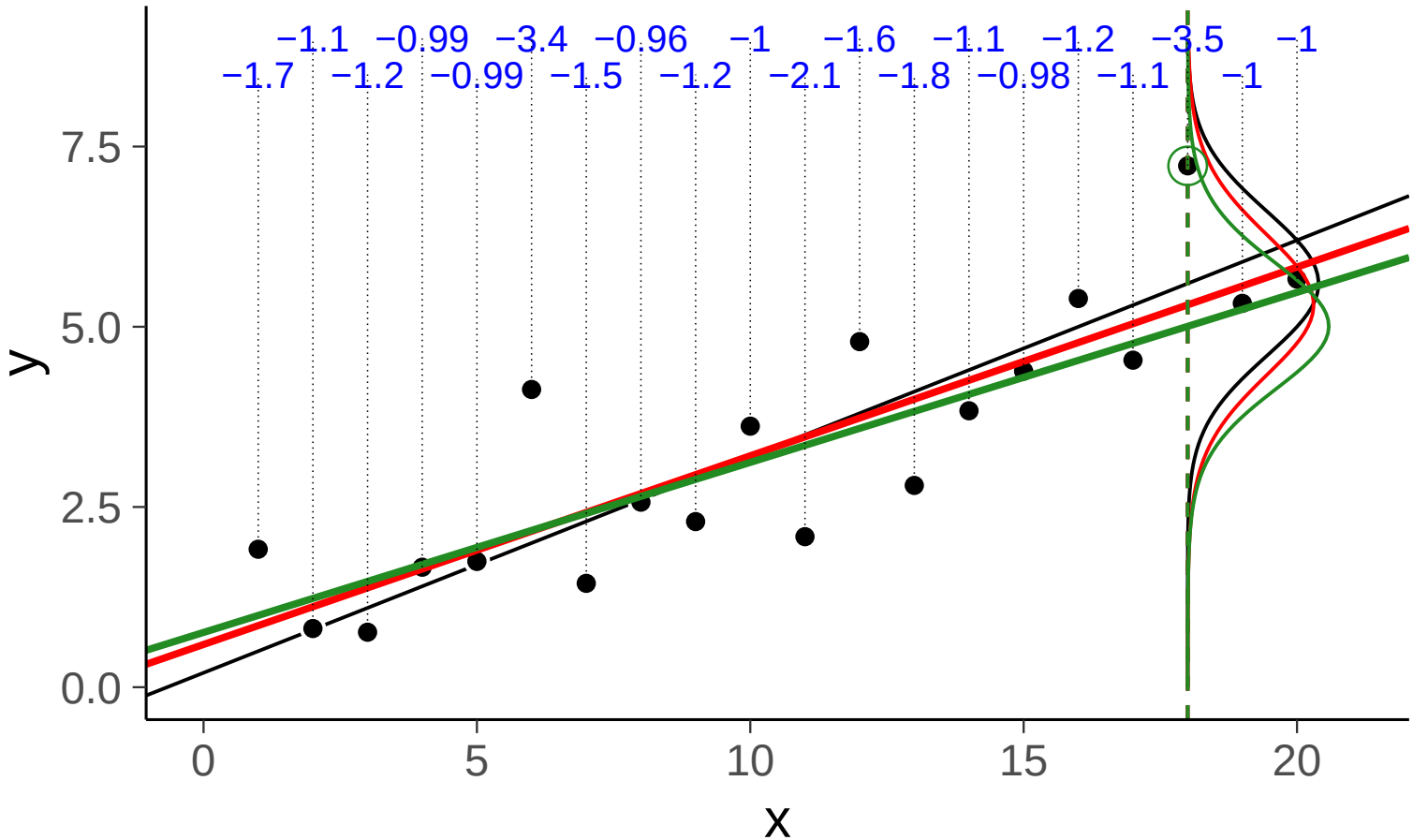
Leave−one−out log predictive densities



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

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

Leave−one−out log predictive densities

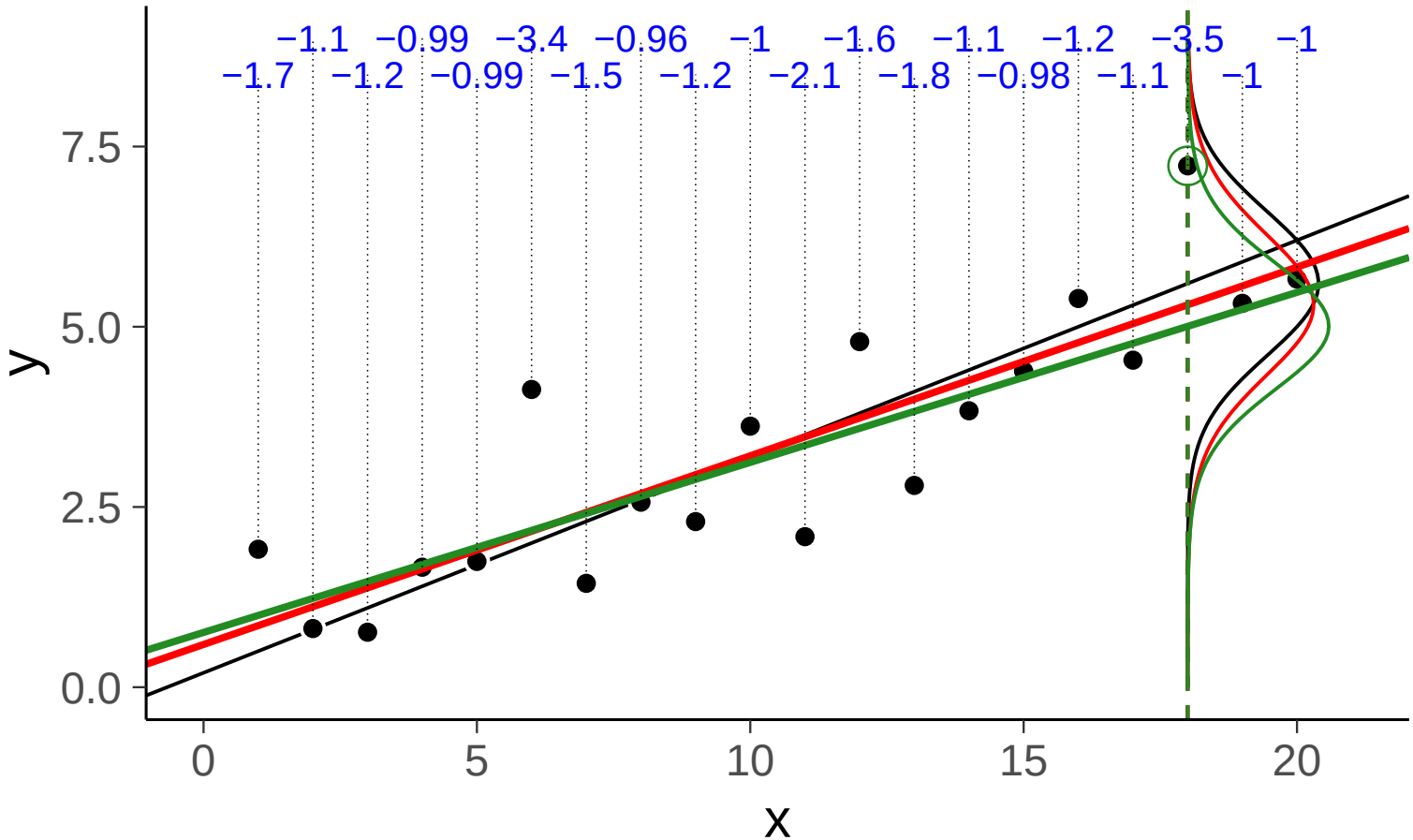


$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

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

Leave−one−out log predictive densities

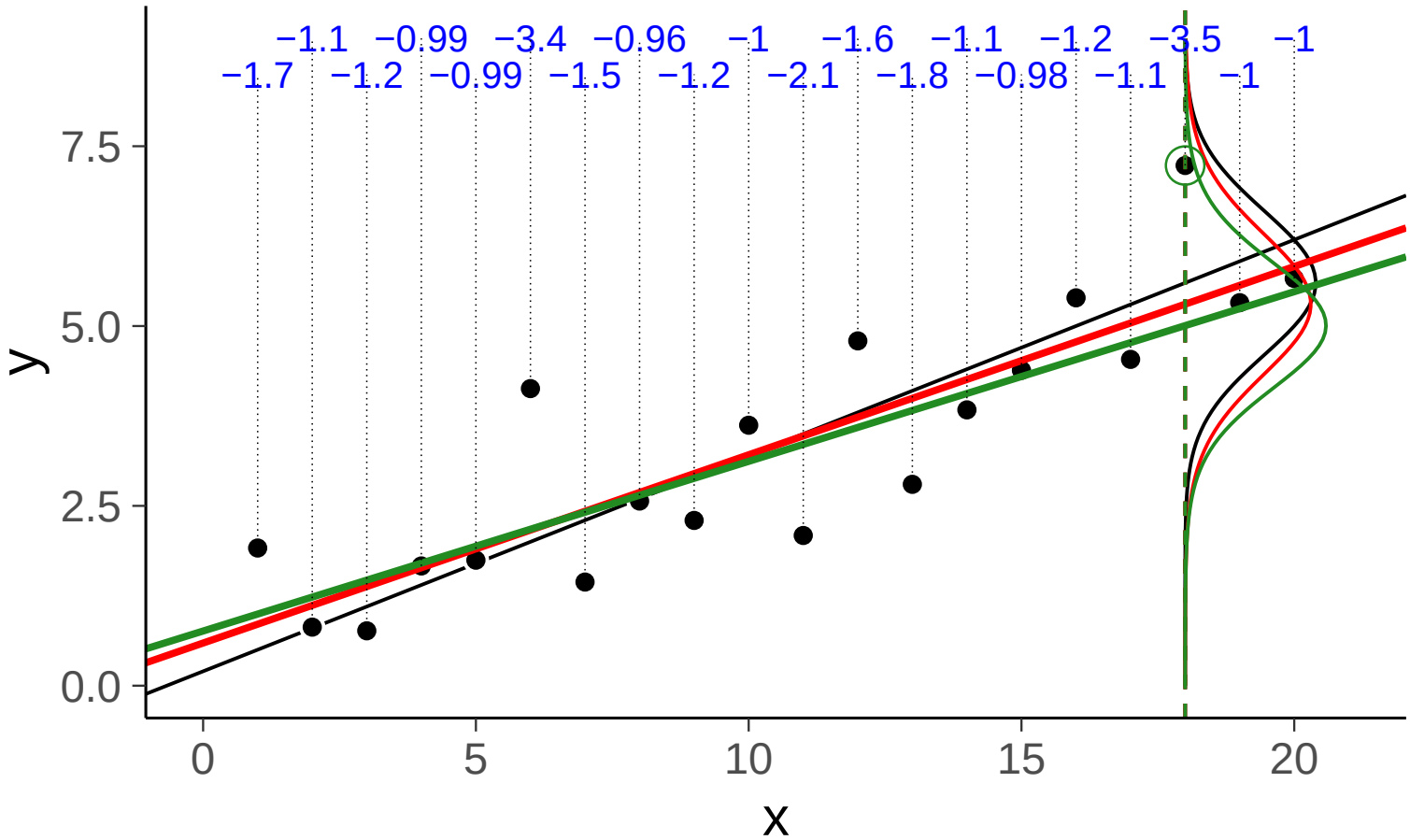


$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

asymptotically approaches p in case of regular faithful model

Leave−one−out log predictive densities



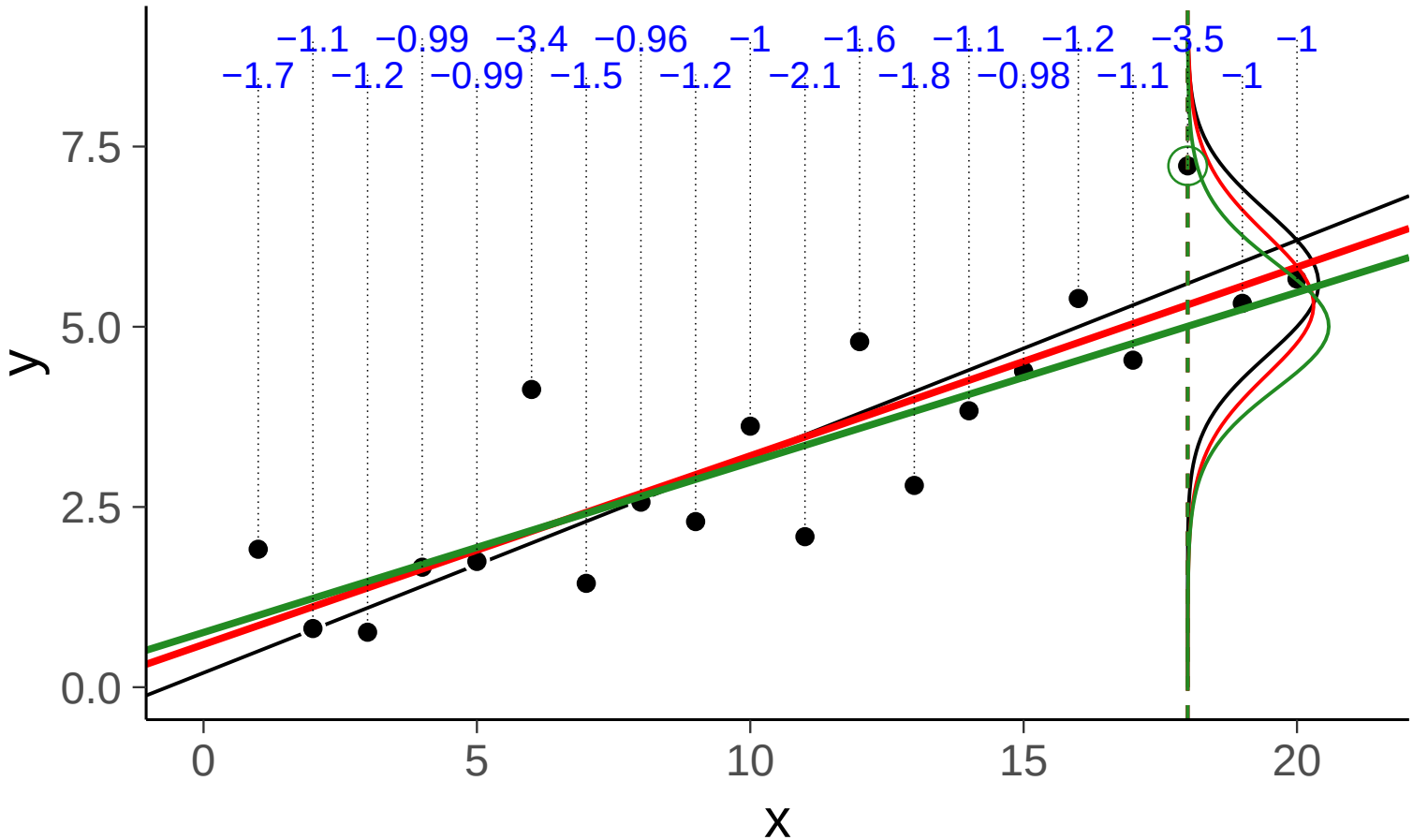
$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$

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

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

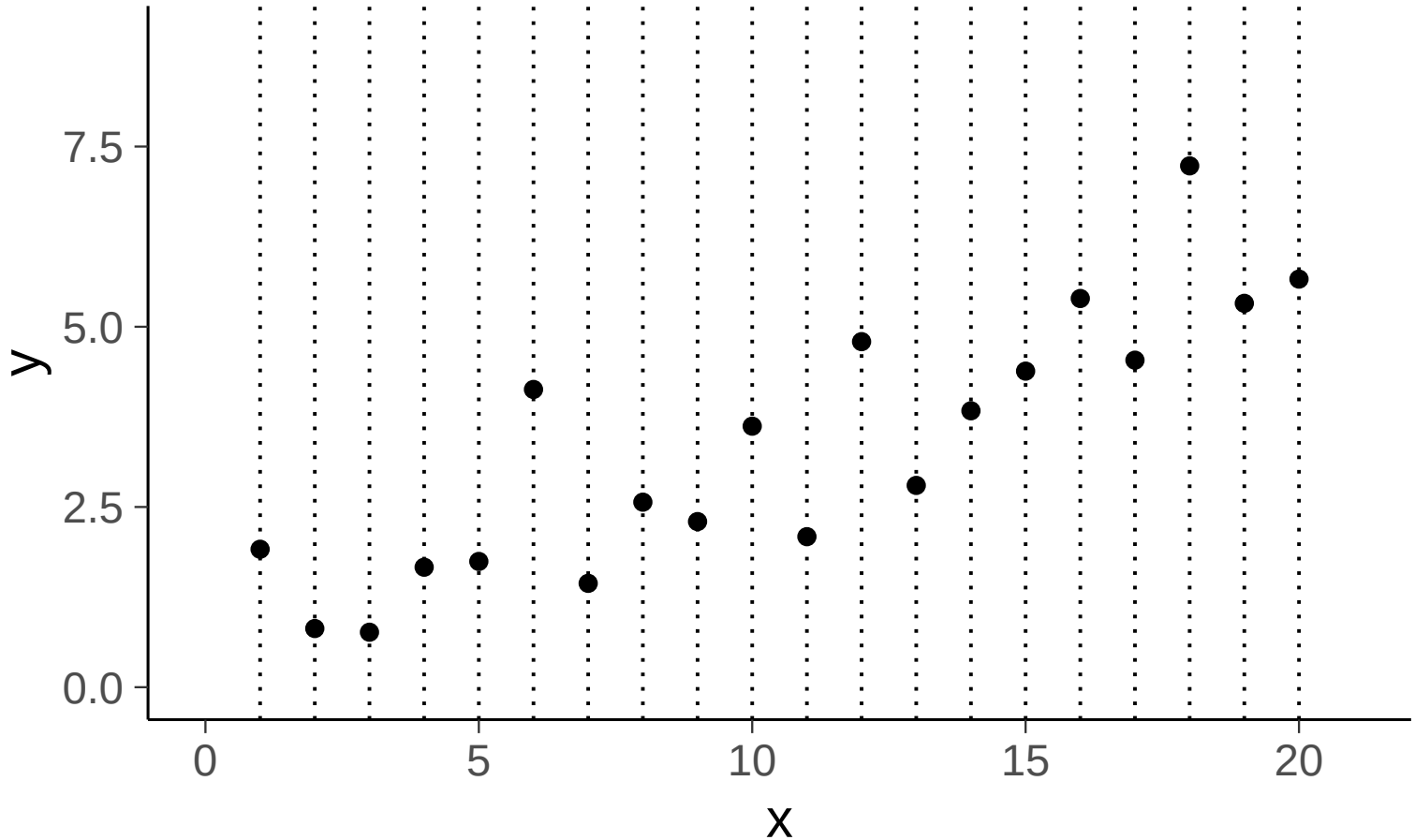


$$\text{elpd_loo} = \sum_{i=1}^{20} \log p(y_i \mid x_i, x_{-i}, y_{-i}) \approx -29.5$$

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

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

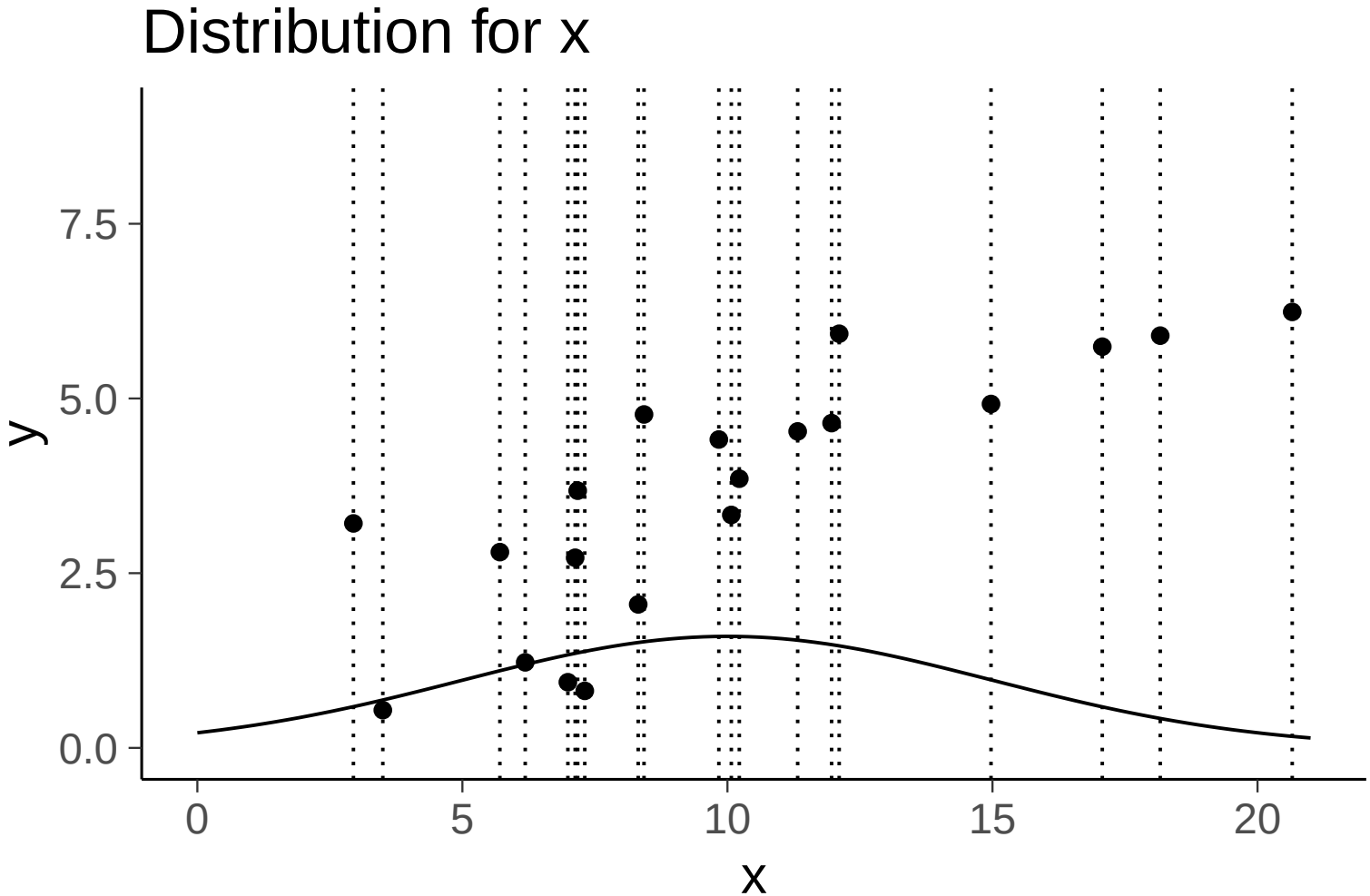
Fixed / designed x



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

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

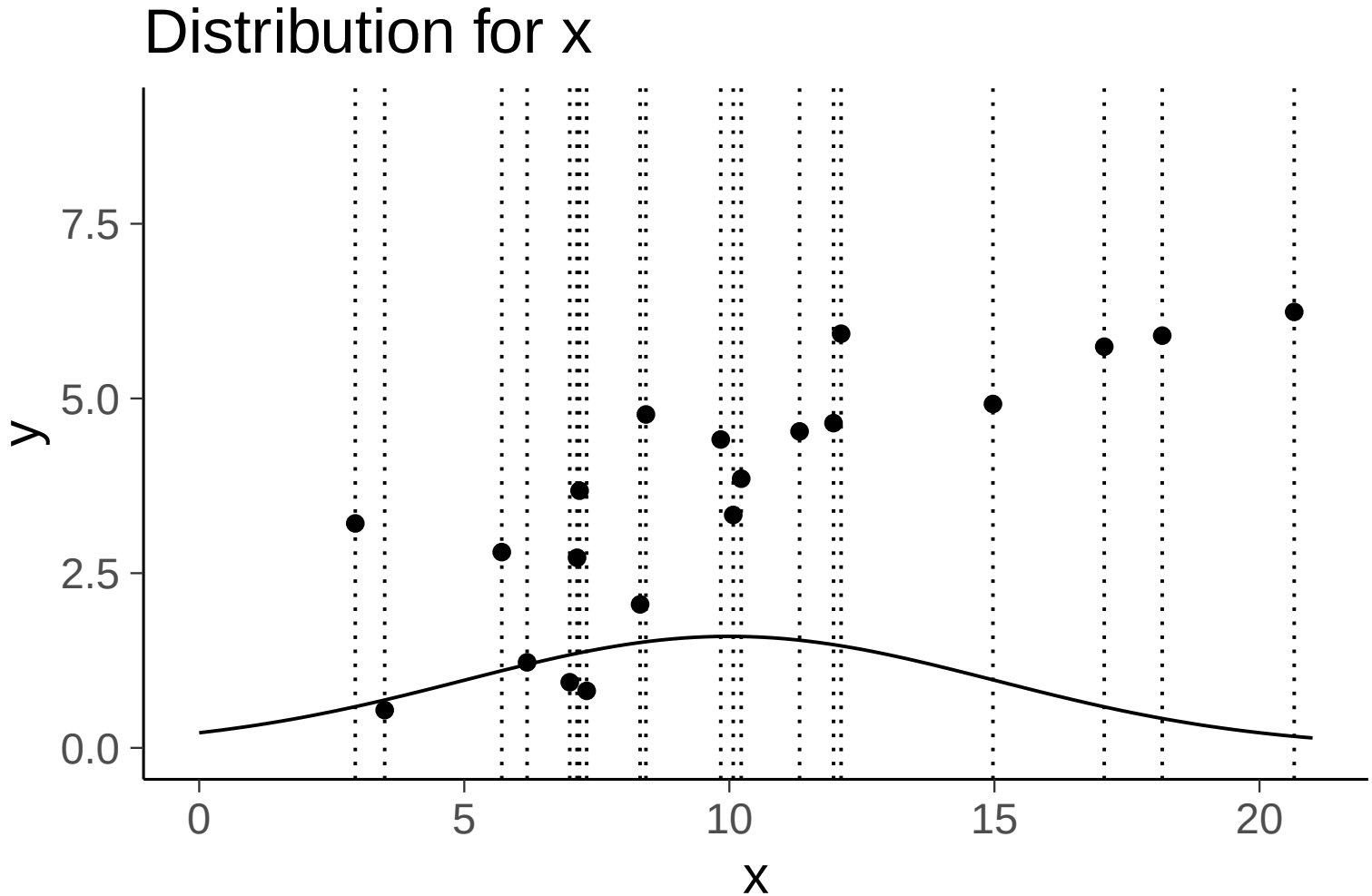
LOO is ok for fixed / designed x. SE is uncertainty about $y \mid x$.



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

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

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



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

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

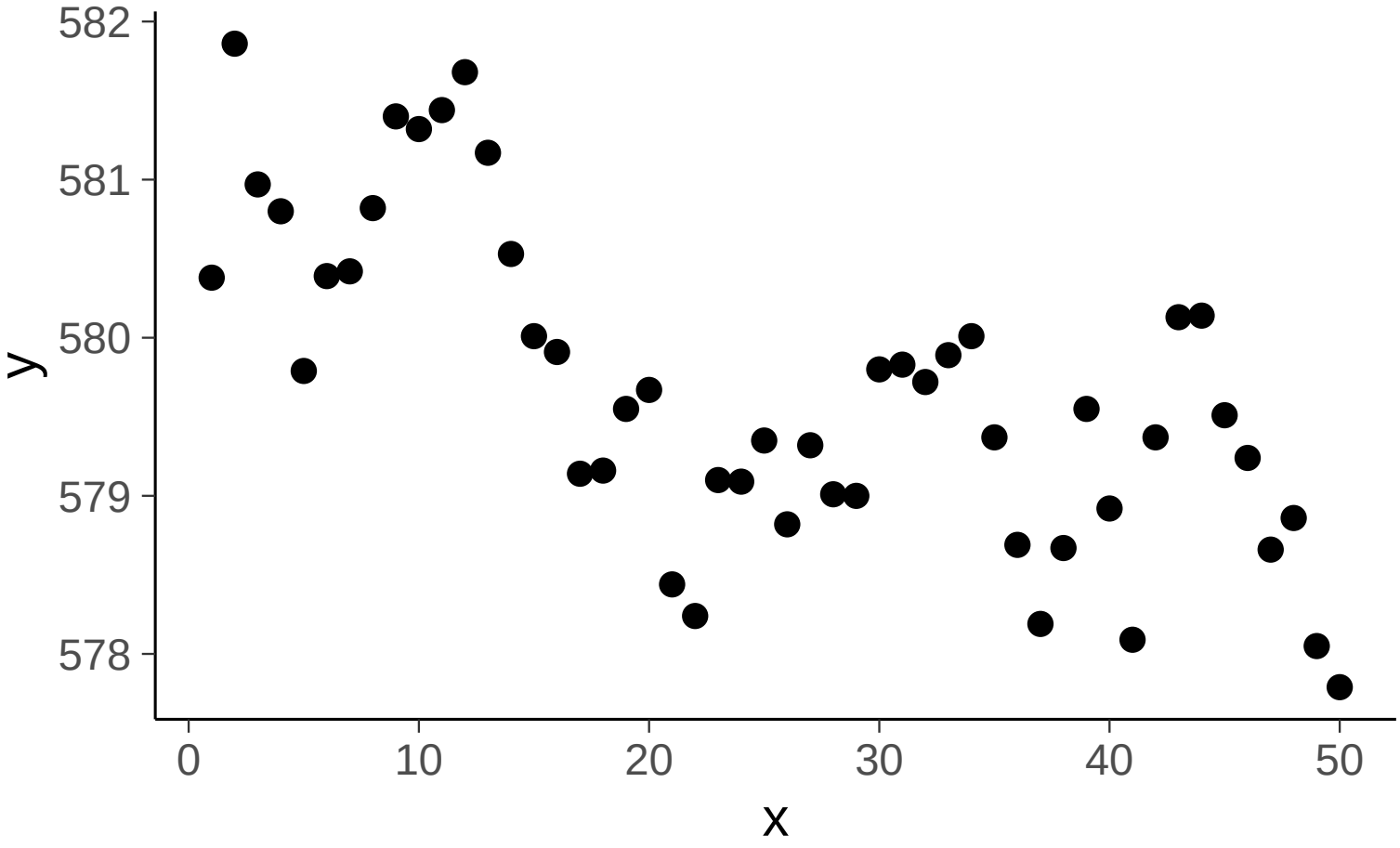
LOO is ok for random x. SE is uncertainty about $y \mid x$ and x.

Covariate shift can be handled with importance weighting or modelling

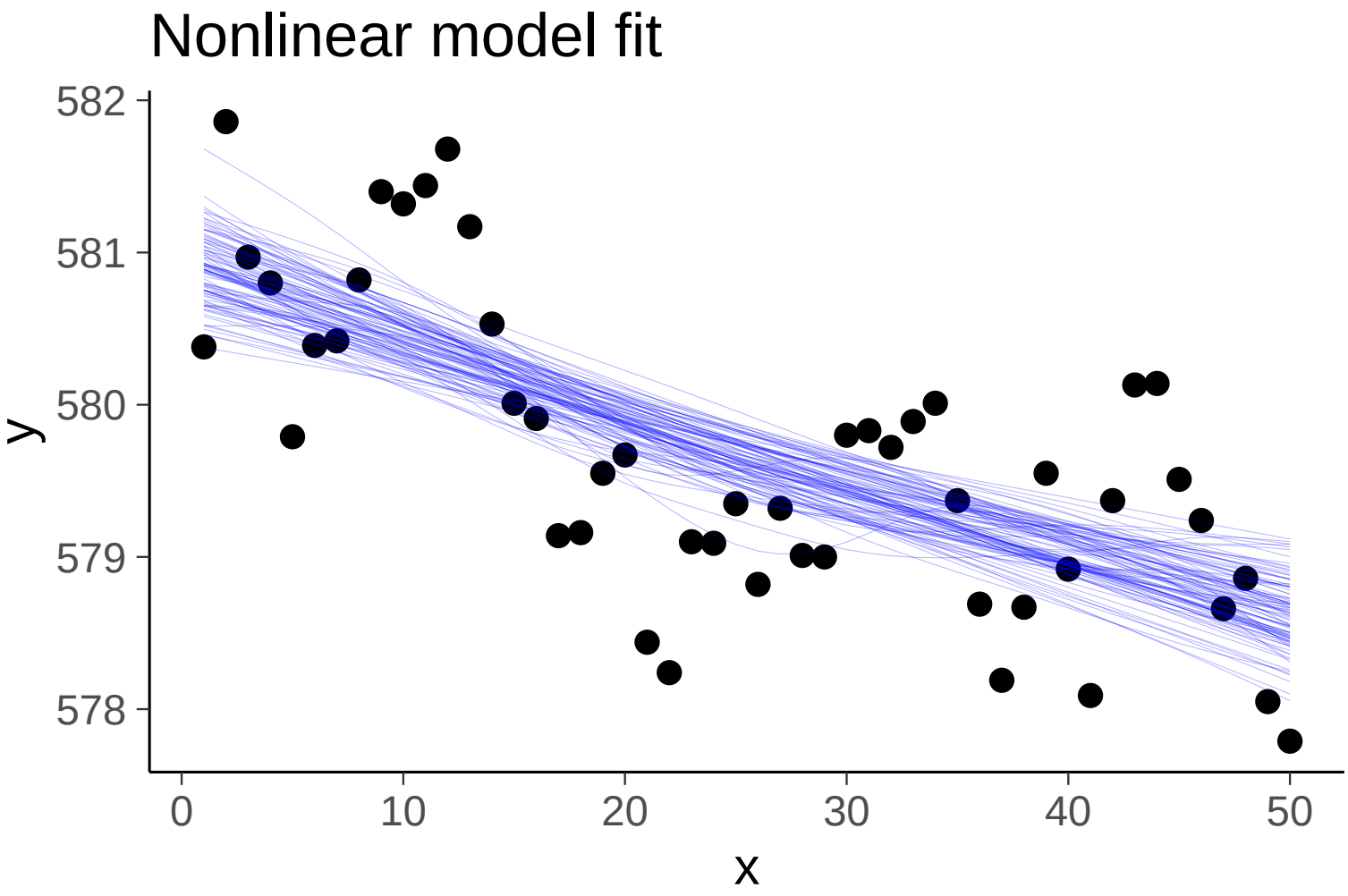
loo package

```
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.
```

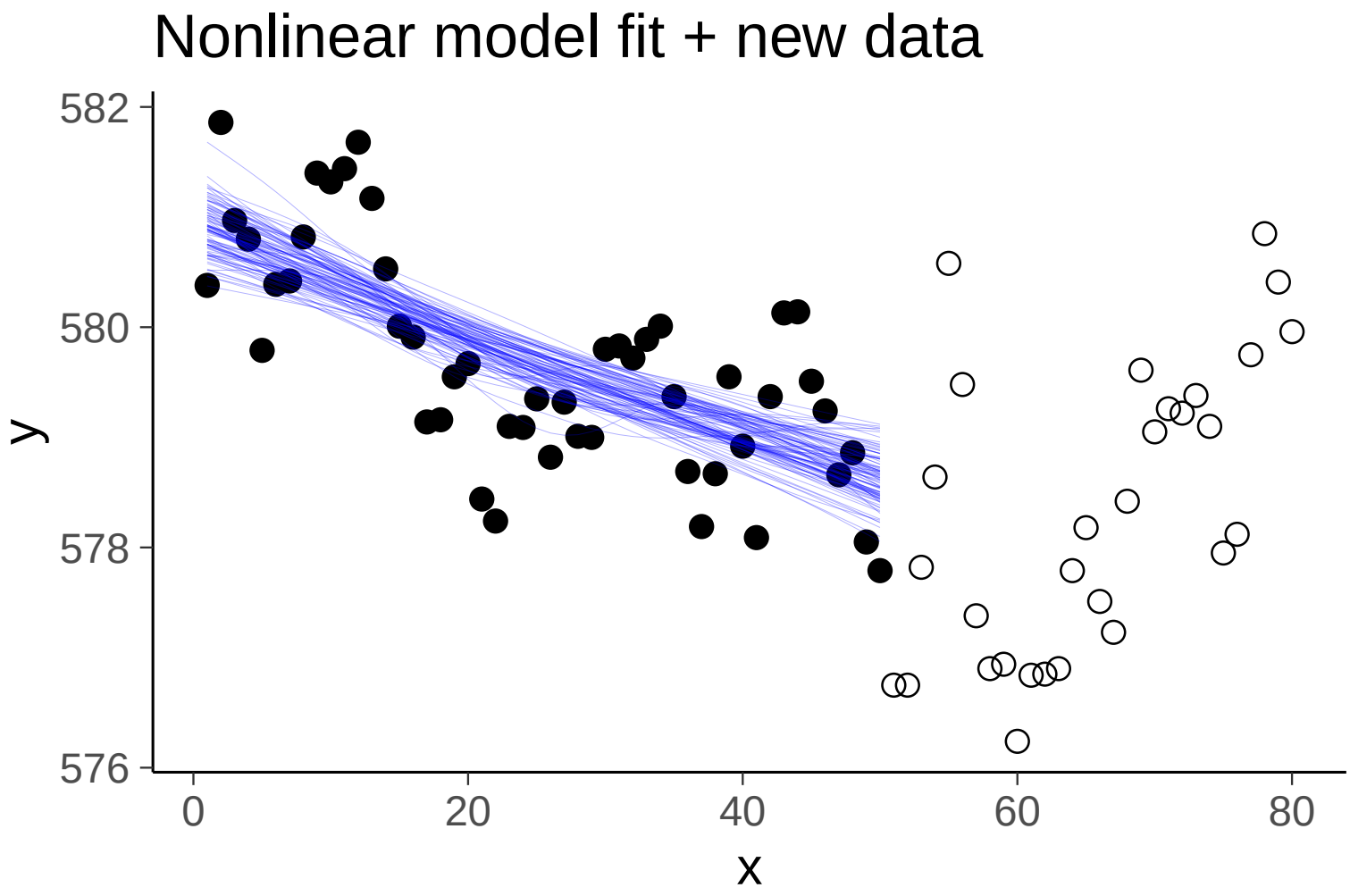
Interpolation vs extrapolation



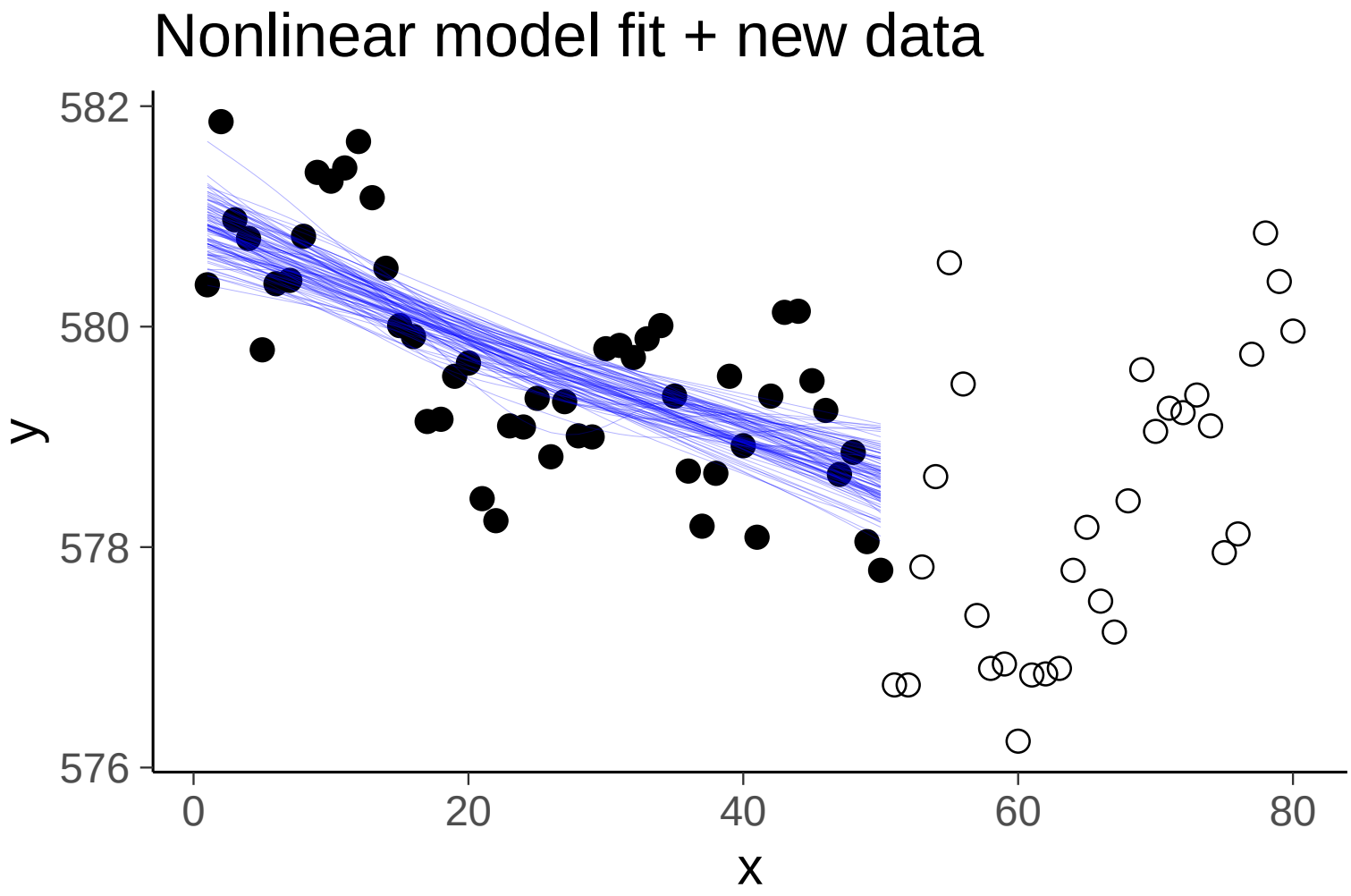
Interpolation vs extrapolation



Interpolation vs extrapolation

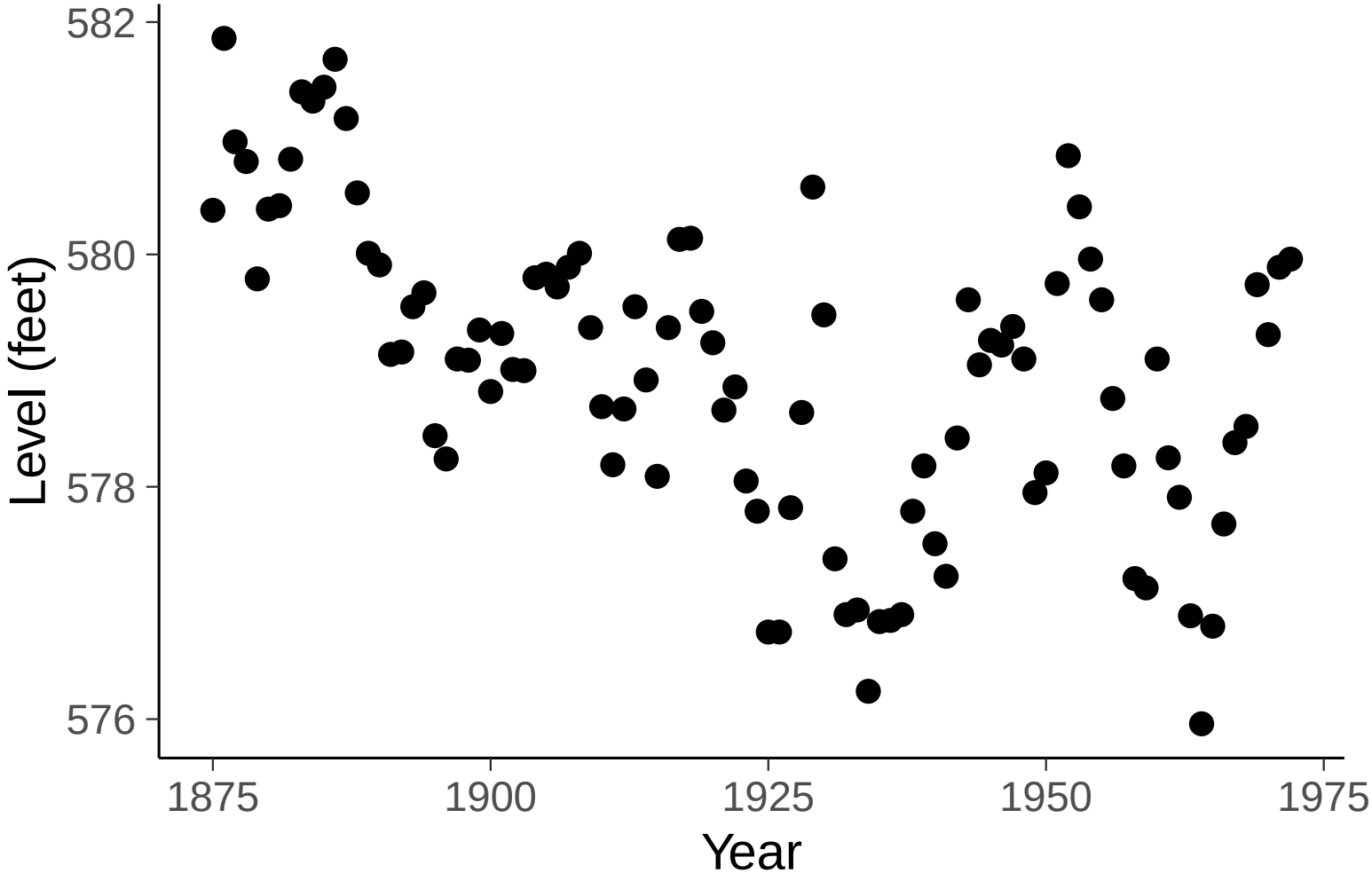


Interpolation vs extrapolation



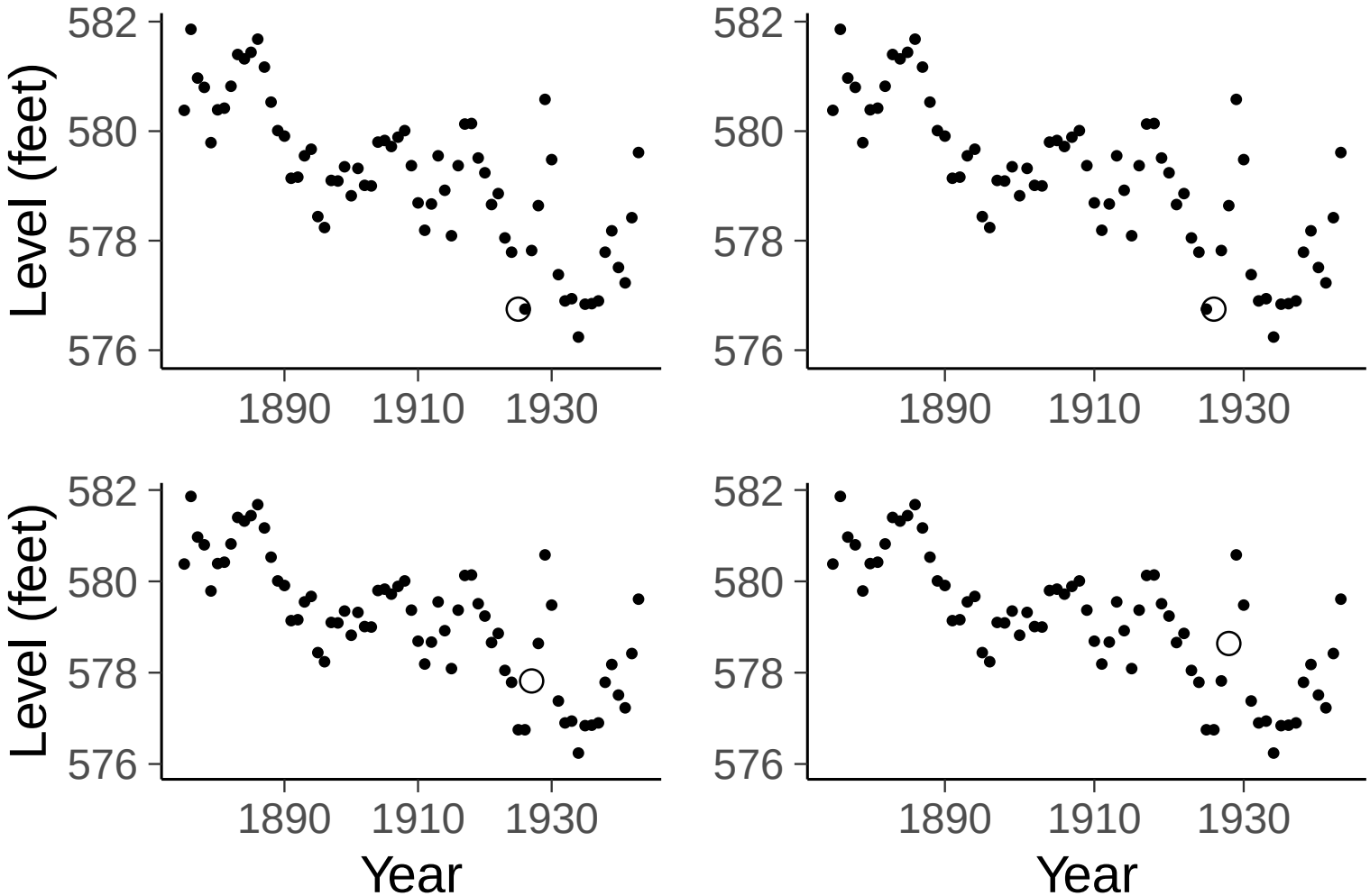
Extrapolation is more difficult

Cross-validation for time series?



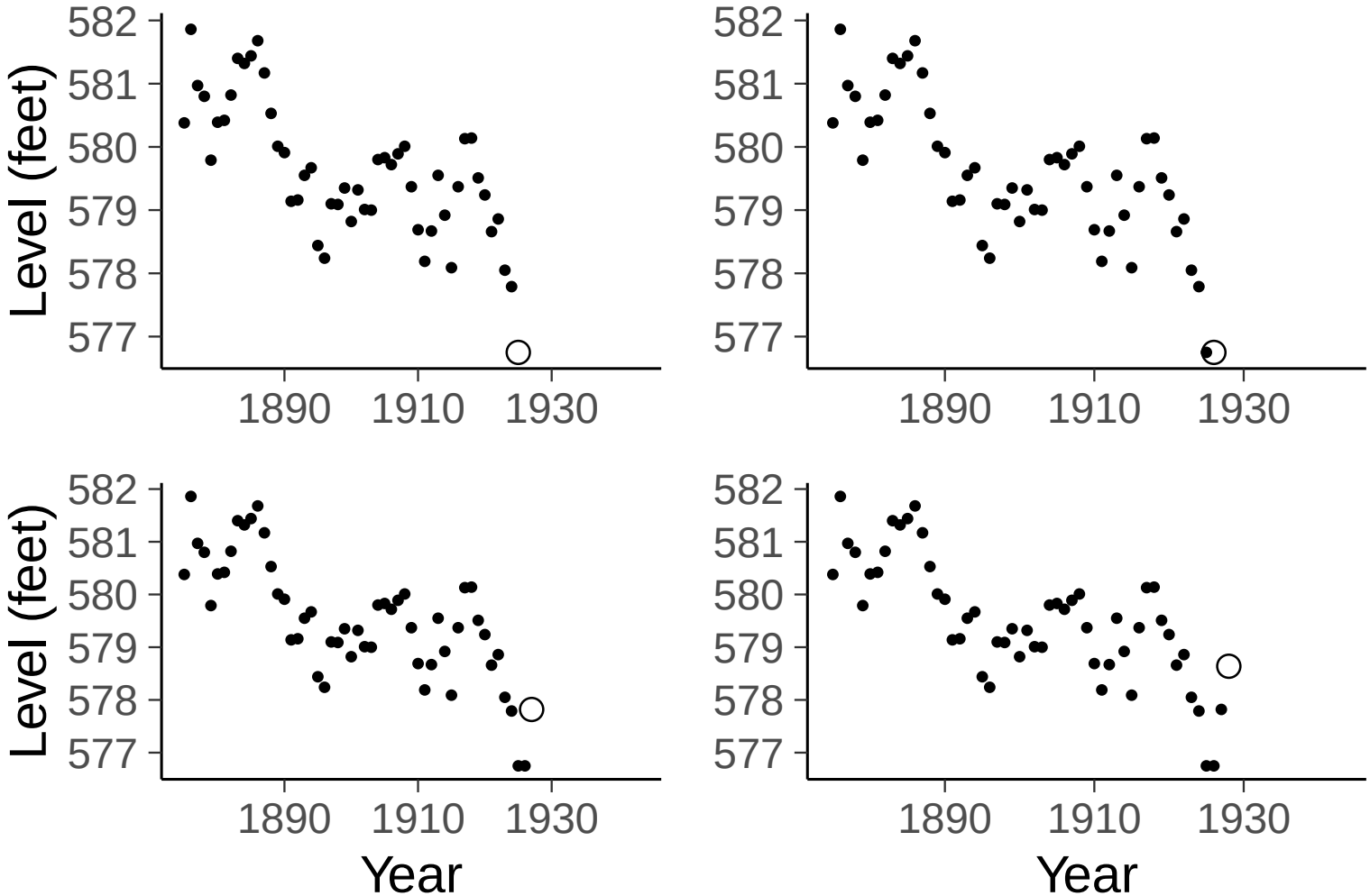
Can LOO or other cross-validation be used with time series?

Cross-validation for time series



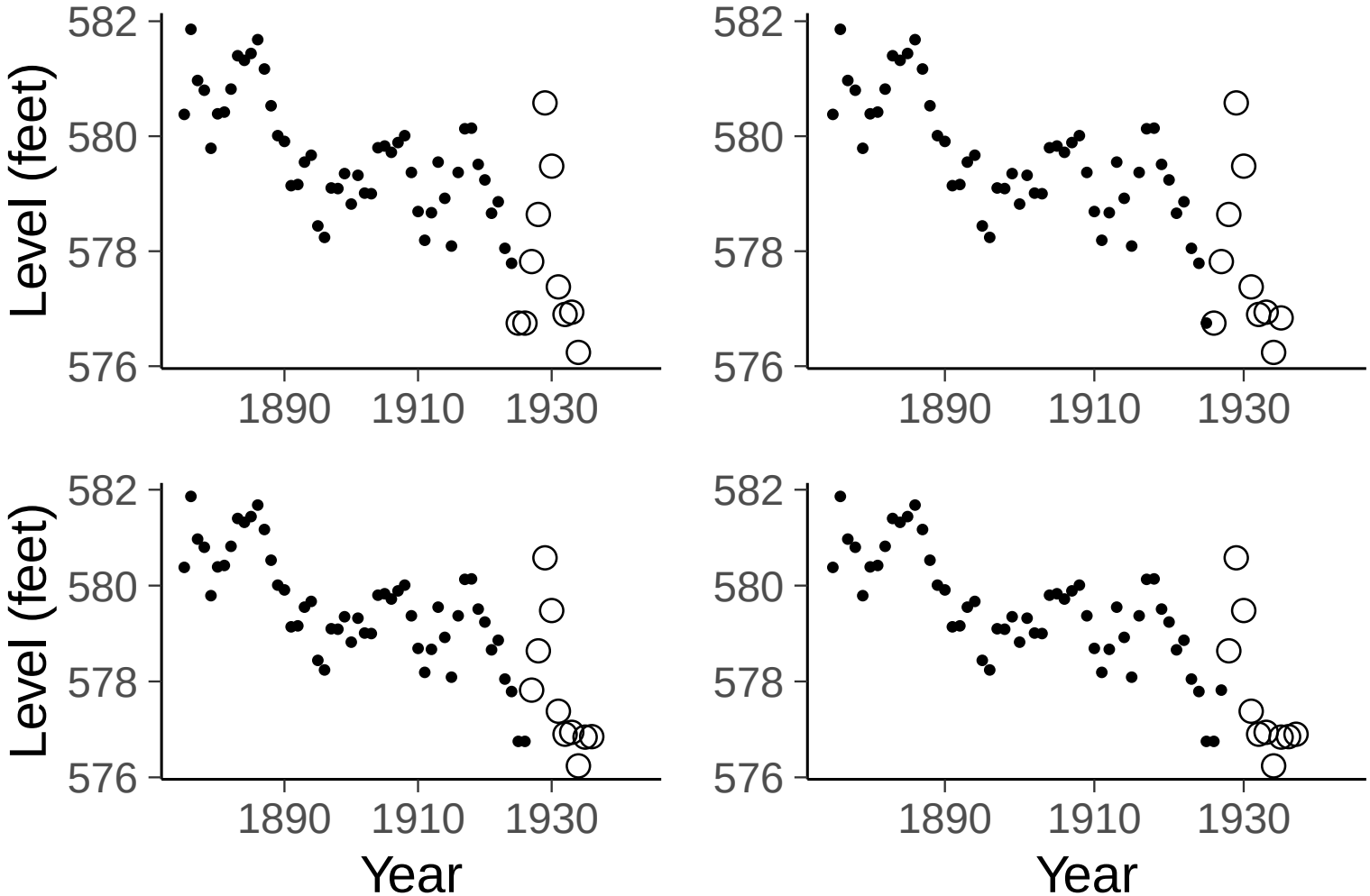
Leave-one-out cross-validation is ok for assessing conditional model

Cross-validation for time series



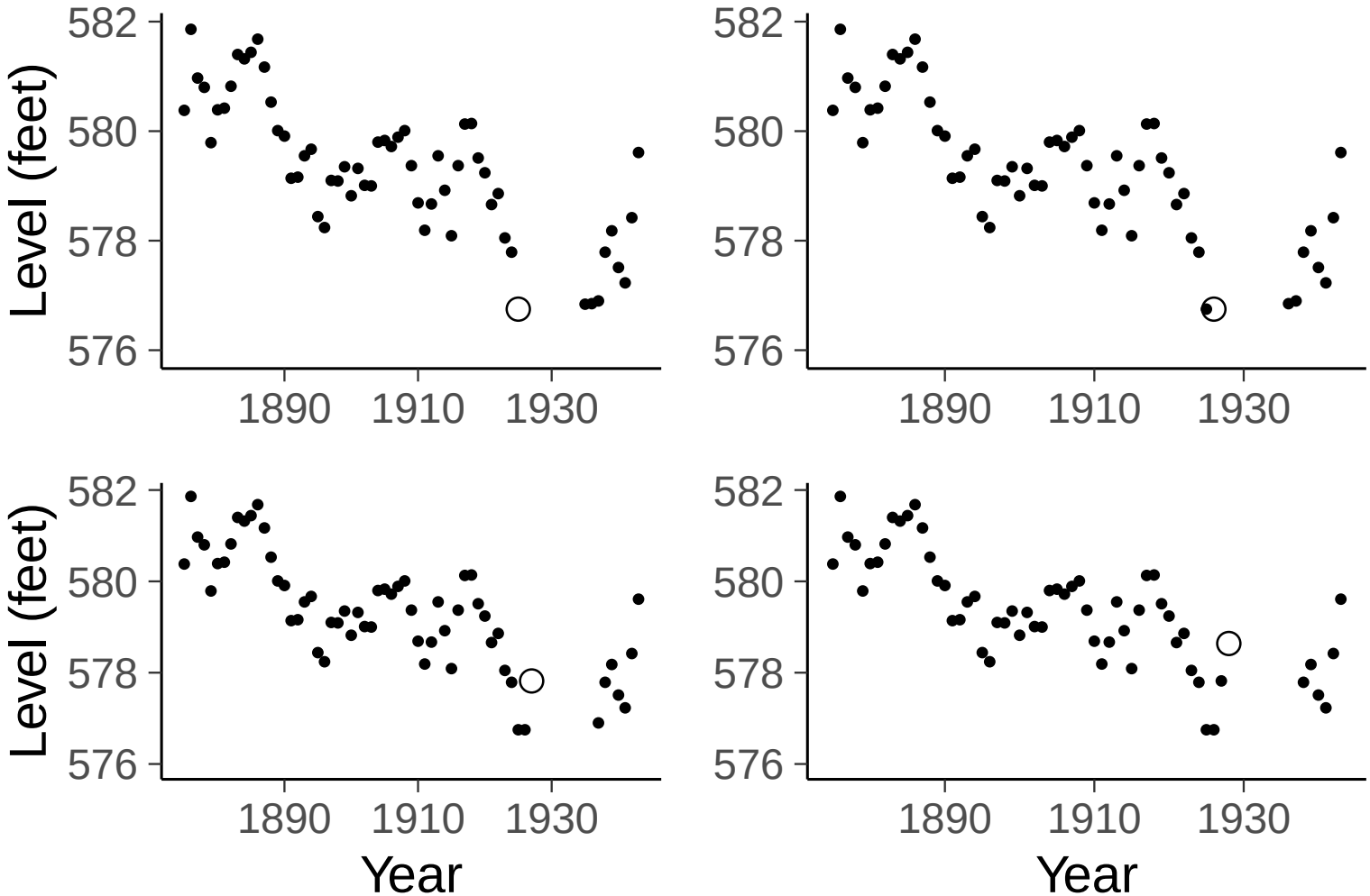
Leave-future-out cross-validation is better for predicting future

Cross-validation for time series



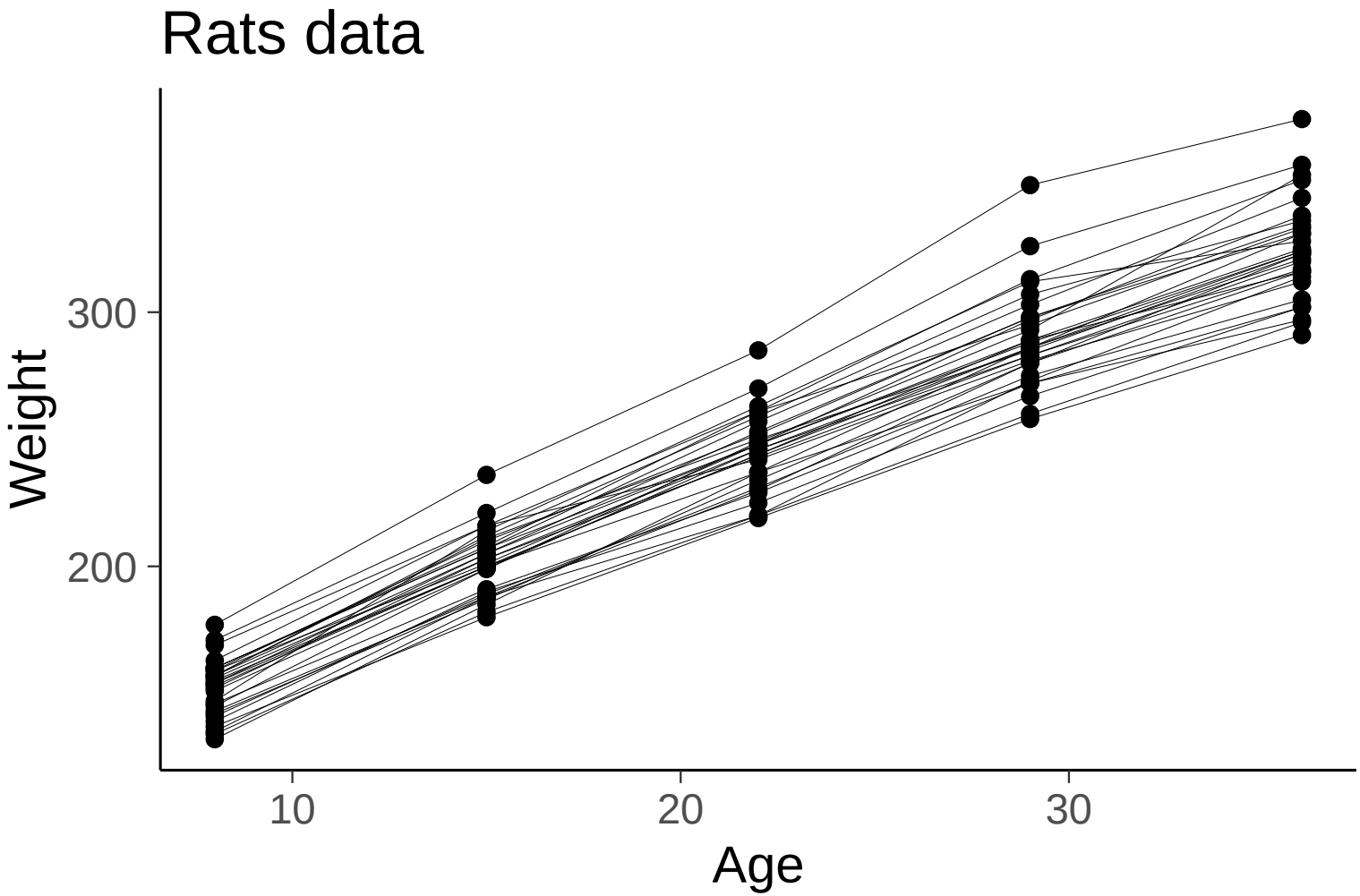
m-step-ahead cross-validation is better for predicting further future

Cross-validation for time series



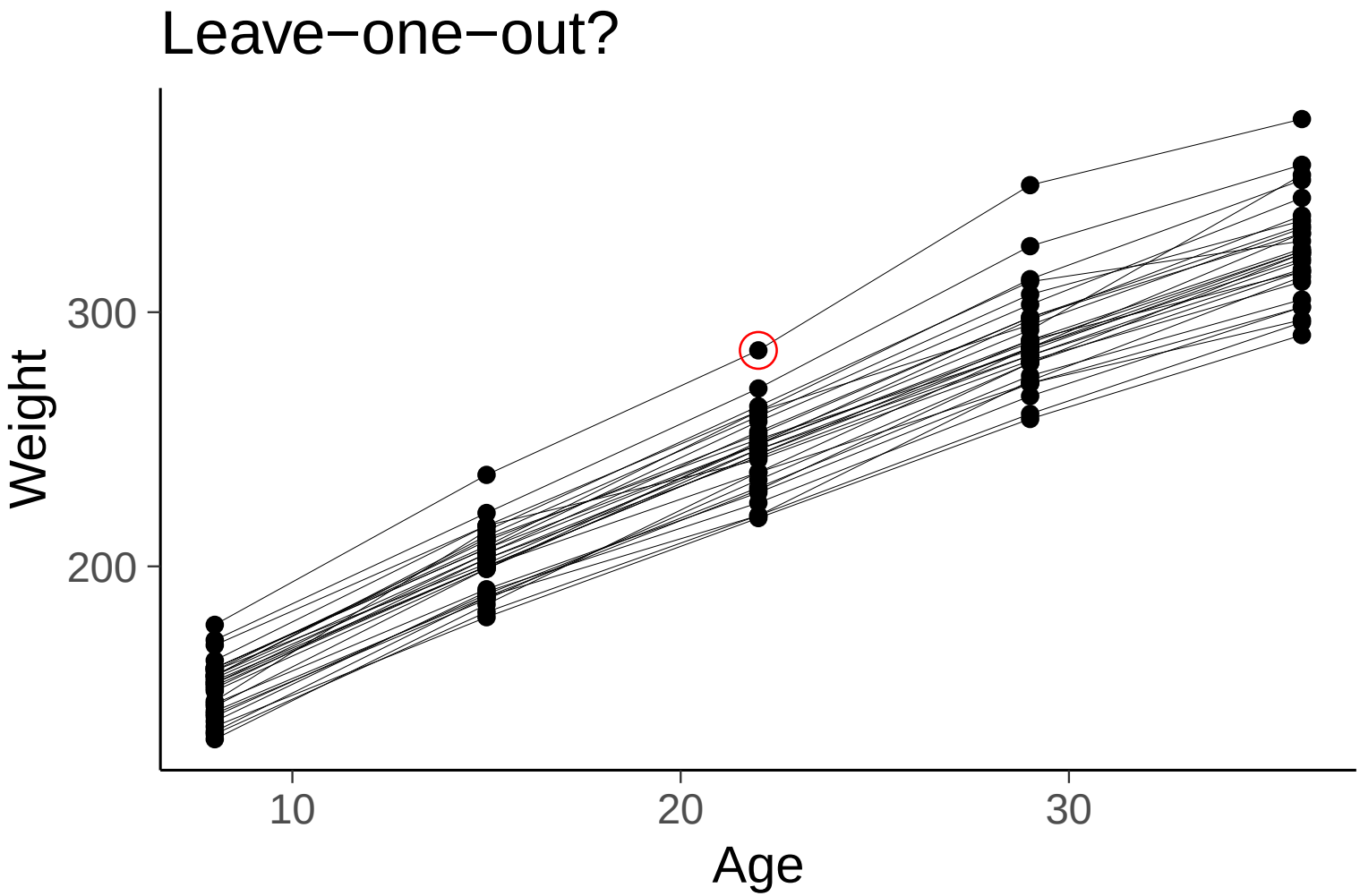
m-step-ahead leave-a-block-out cross-validation

Cross-validation for hierarchical data



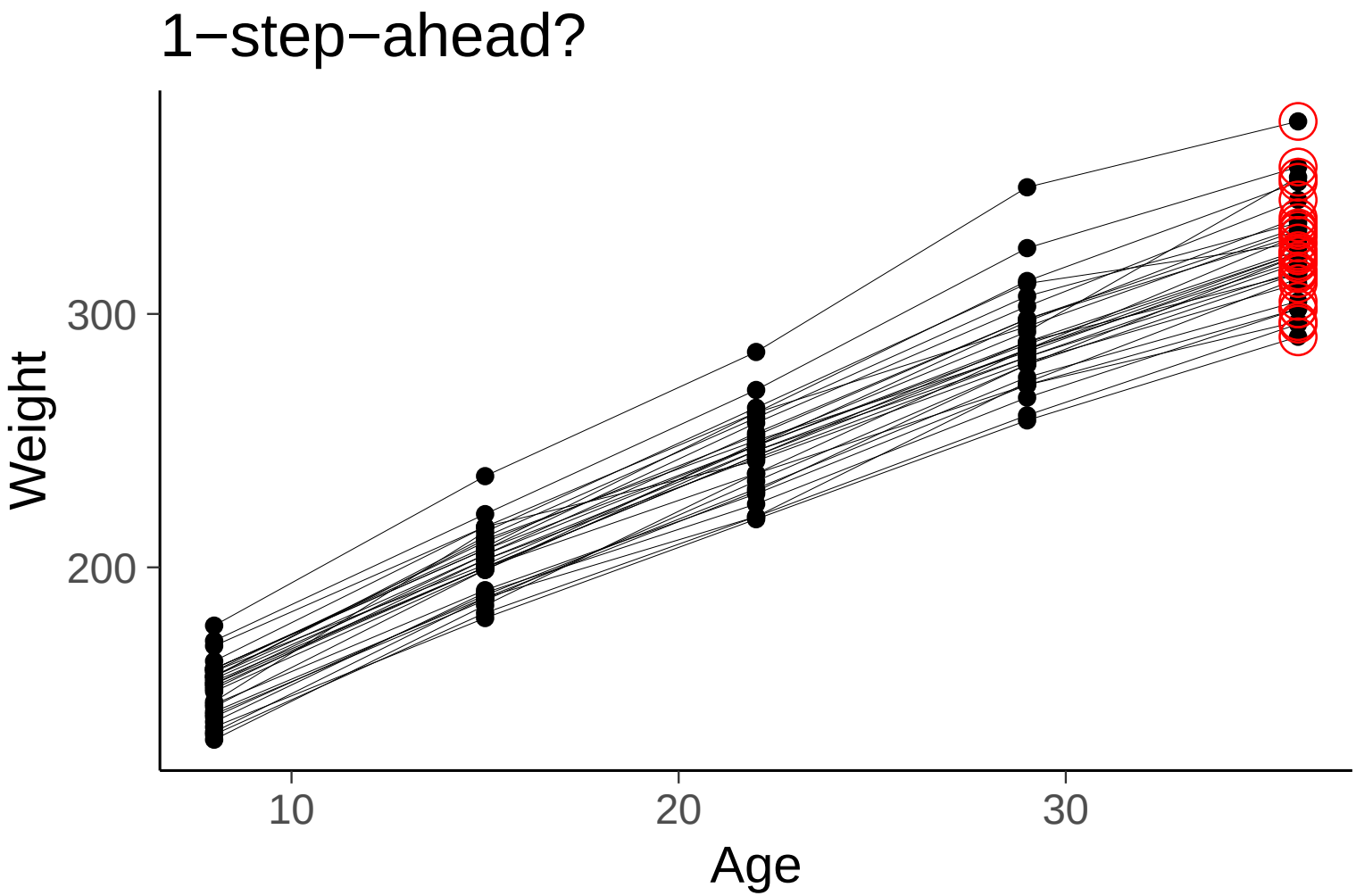
Can LOO or other cross-validation be used with hierarchical data?

Cross-validation for hierarchical data



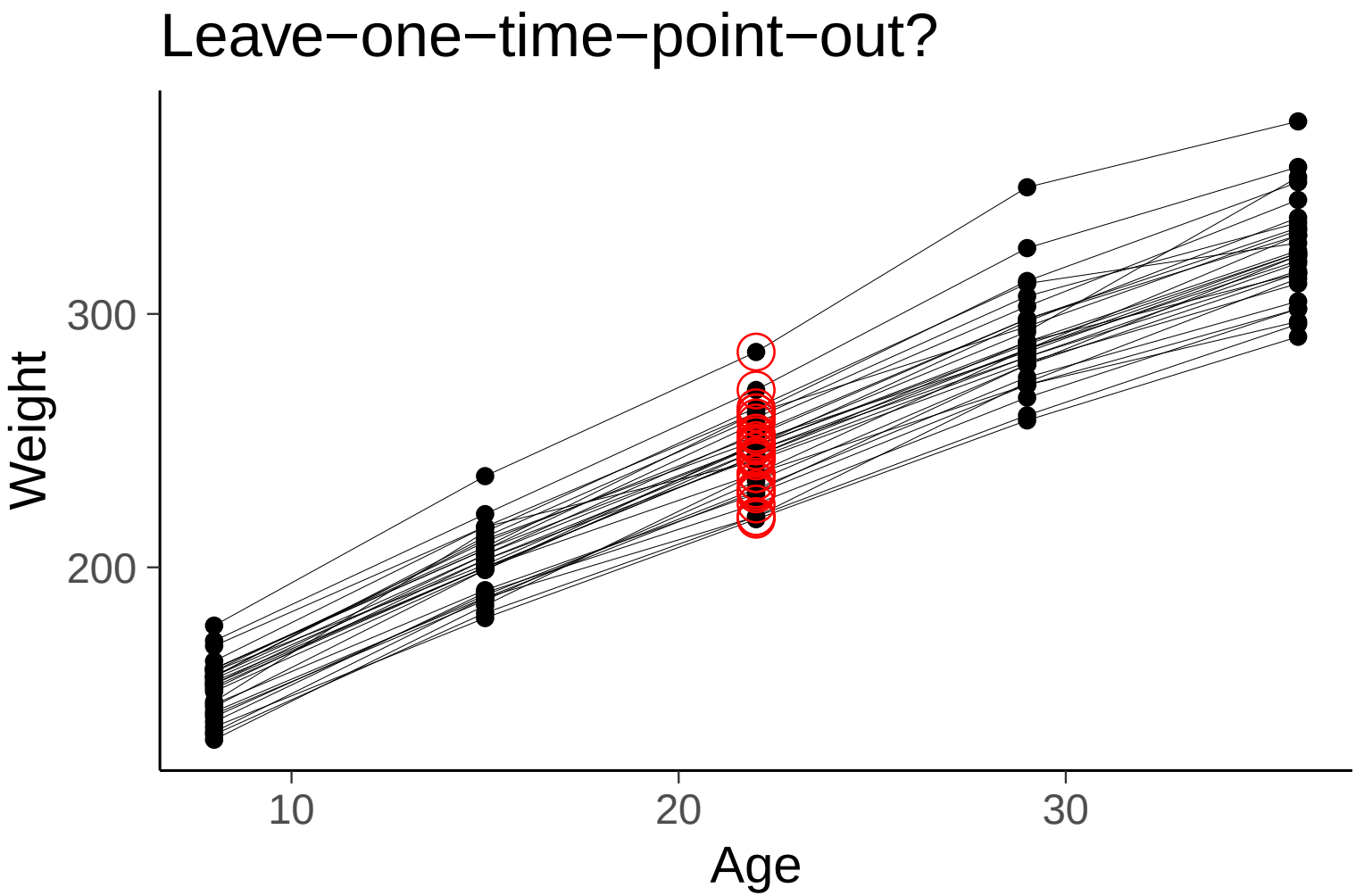
Yes!

Cross-validation for hierarchical data



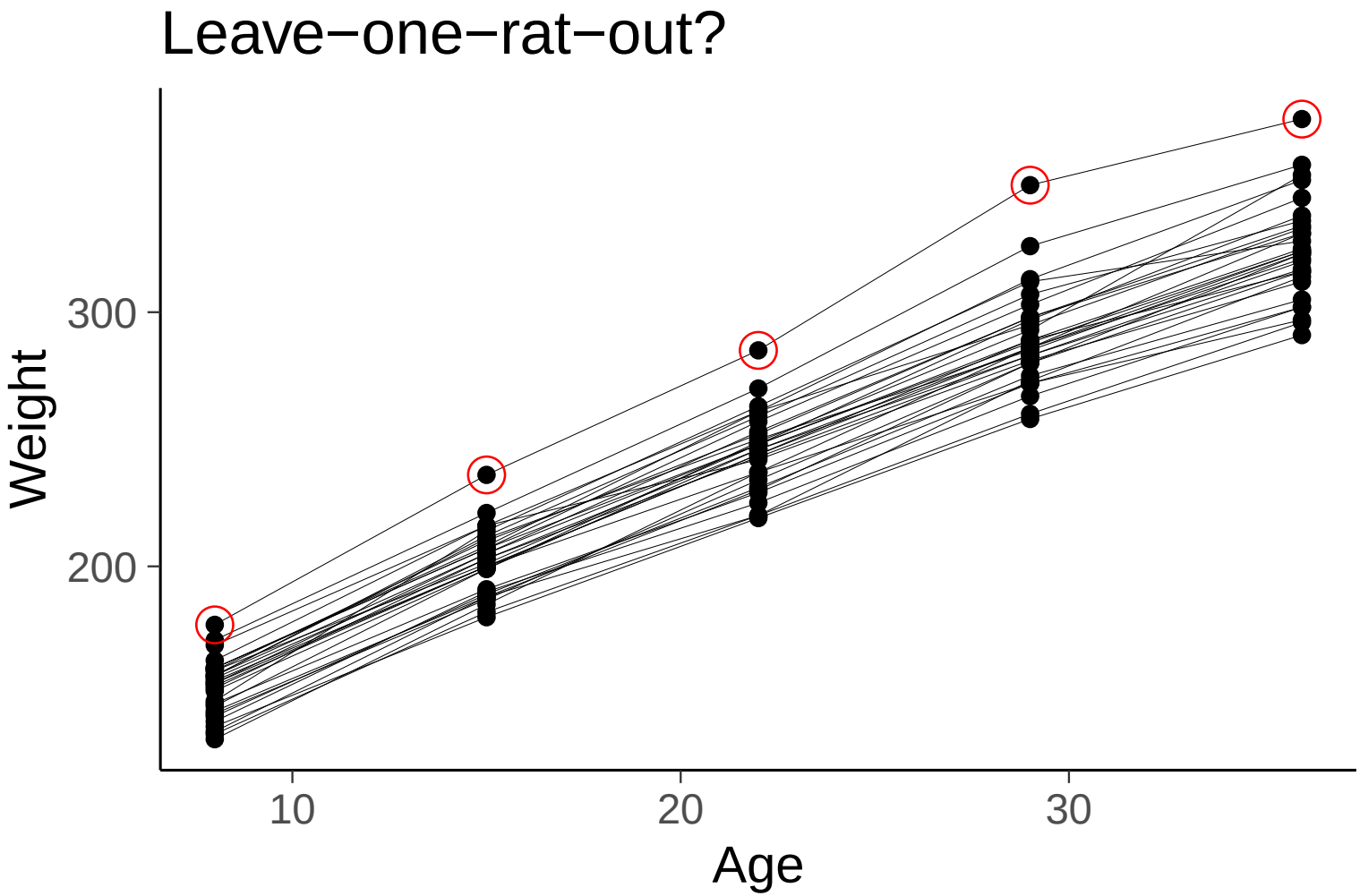
Yes!

Cross-validation for hierarchical data



Yes!

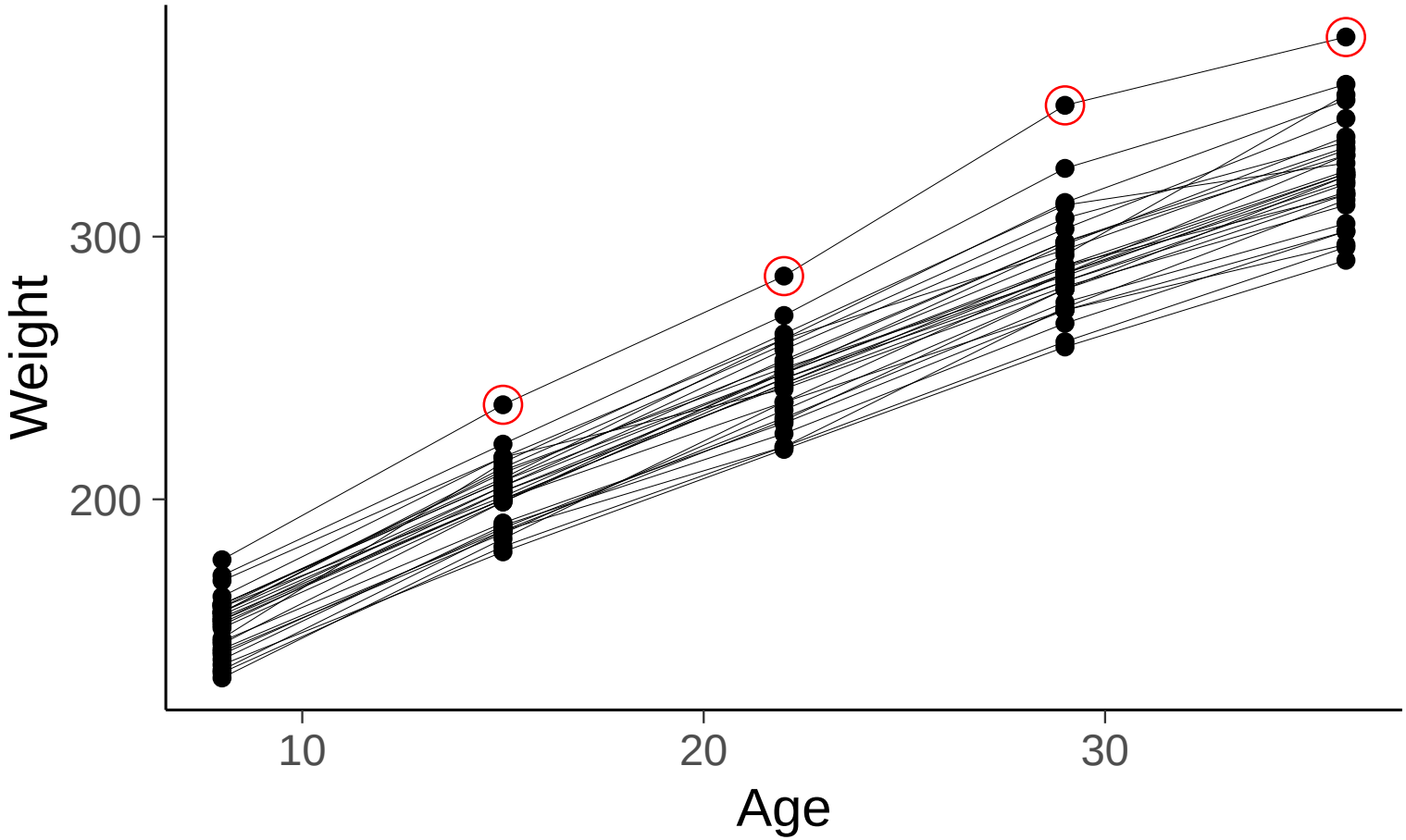
Cross-validation for hierarchical data



Yes!

Cross-validation for hierarchical data

Predict given initial weight?



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

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

Importance sampling leave-one-out cross-validation

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

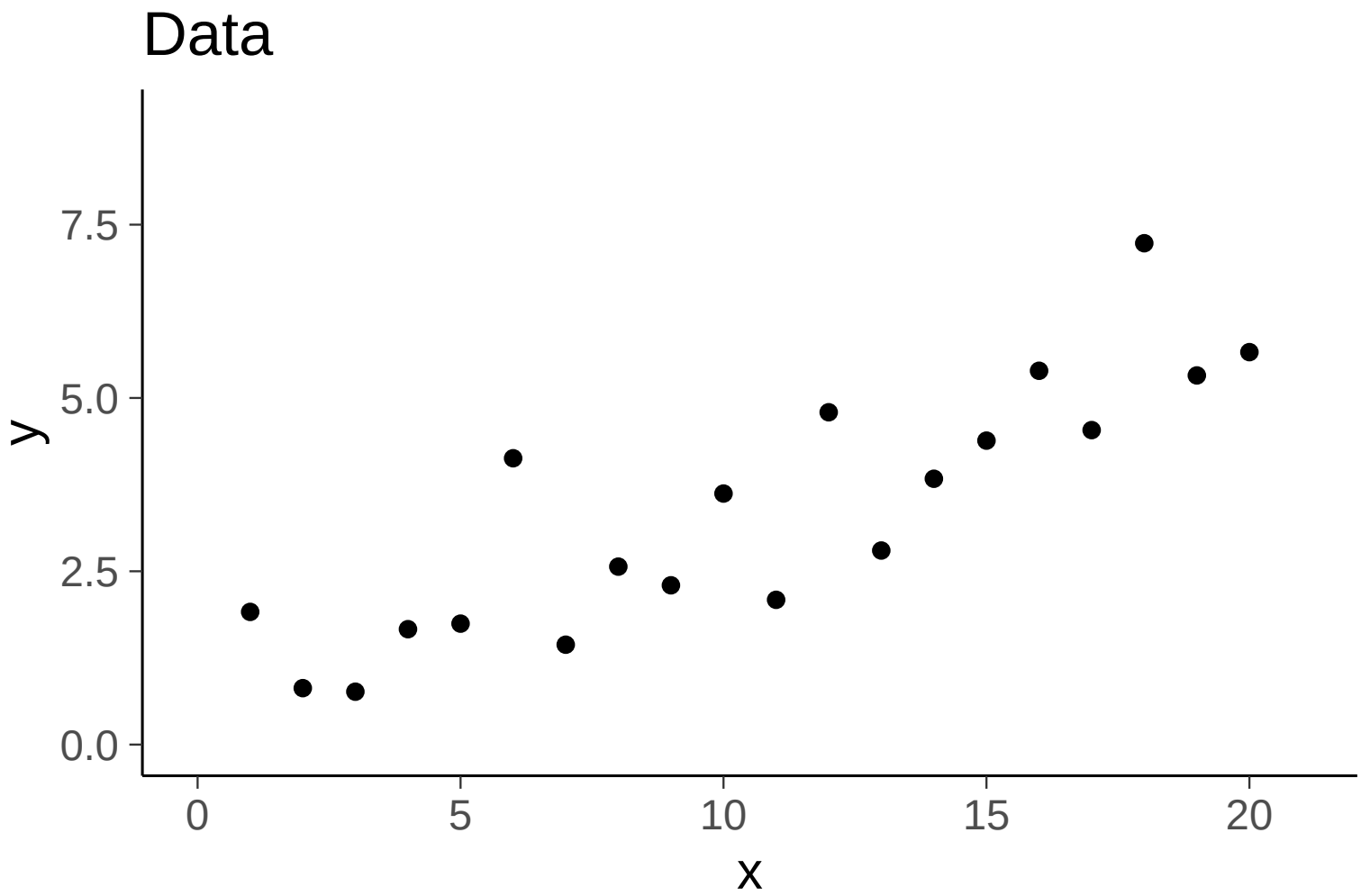
$$w_i^{(s)} = \frac{p(\theta^{(s)} \mid x_{-i}, y_{-i})}{p(\theta^{(s)} \mid x, y)} \propto \frac{1}{p(y_i \mid \theta^{(s)})}$$

Importance sampling leave-one-out cross-validation

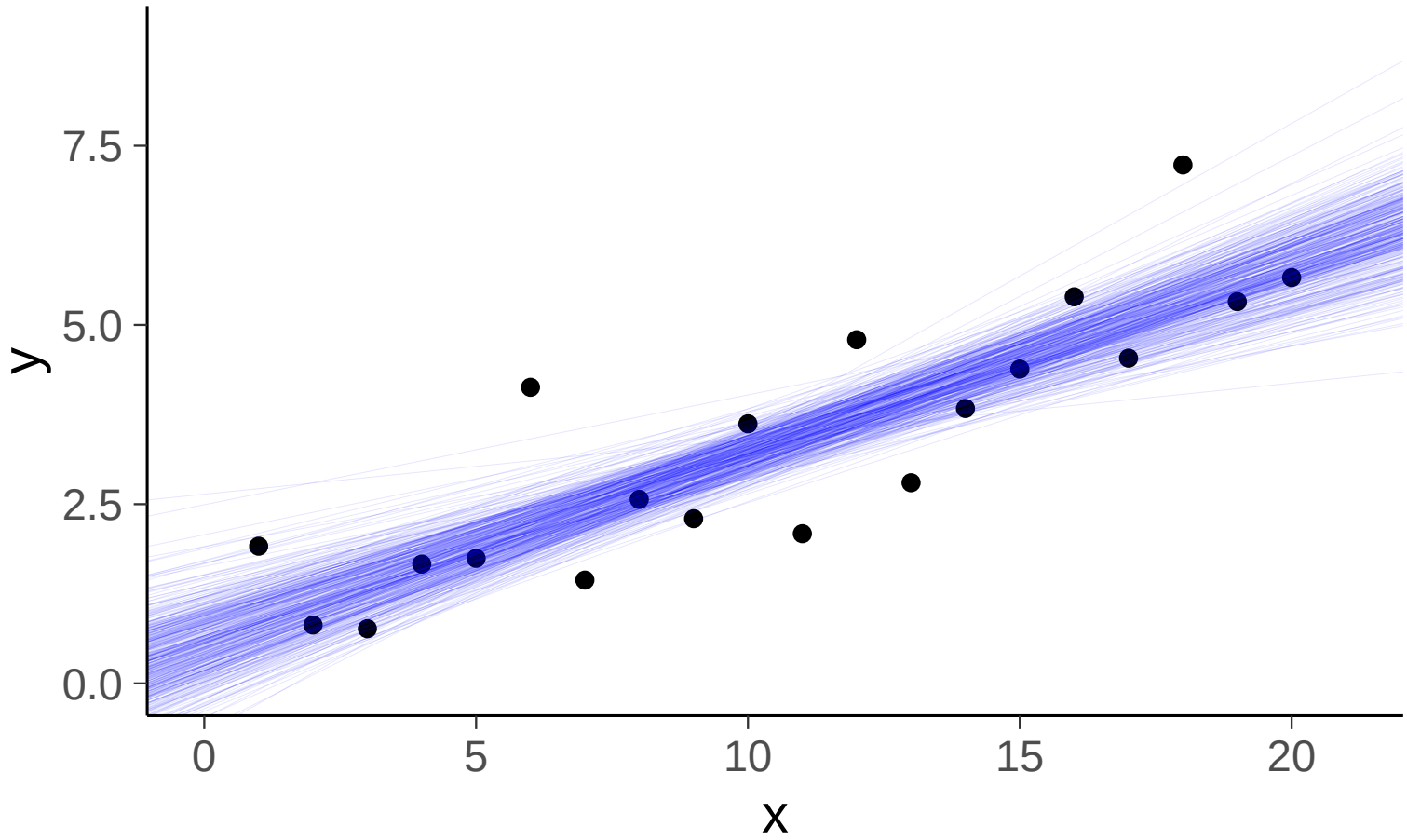
- We want to compute
  $p(y_i \mid x_{-i}, y_{-i}) = \int p(y_i \mid x_i, \theta) p(\theta \mid x_{-i}, y_{-i}) d\theta$
- Proposal distribution is full posterior $\theta^{(s)} \sim p(\theta \mid x, y)$
- Target distribution is LOO-posterior $p(\theta \mid x_{-i}, y_{-i})$
- Importance ratio

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

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

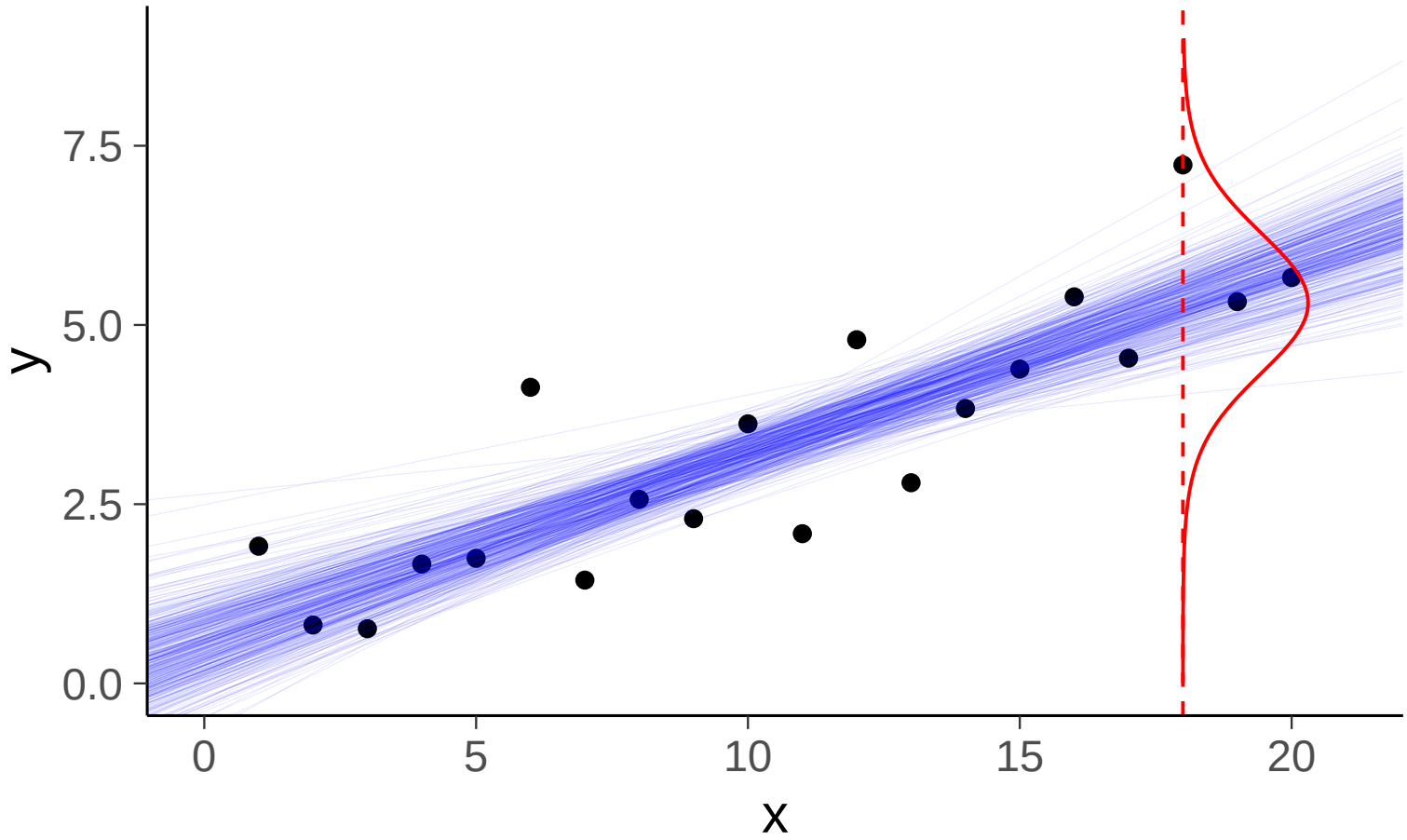

Data
7.5
5.0
2.5
0.0
y
0
5
10
15
20
x

Posterior draws



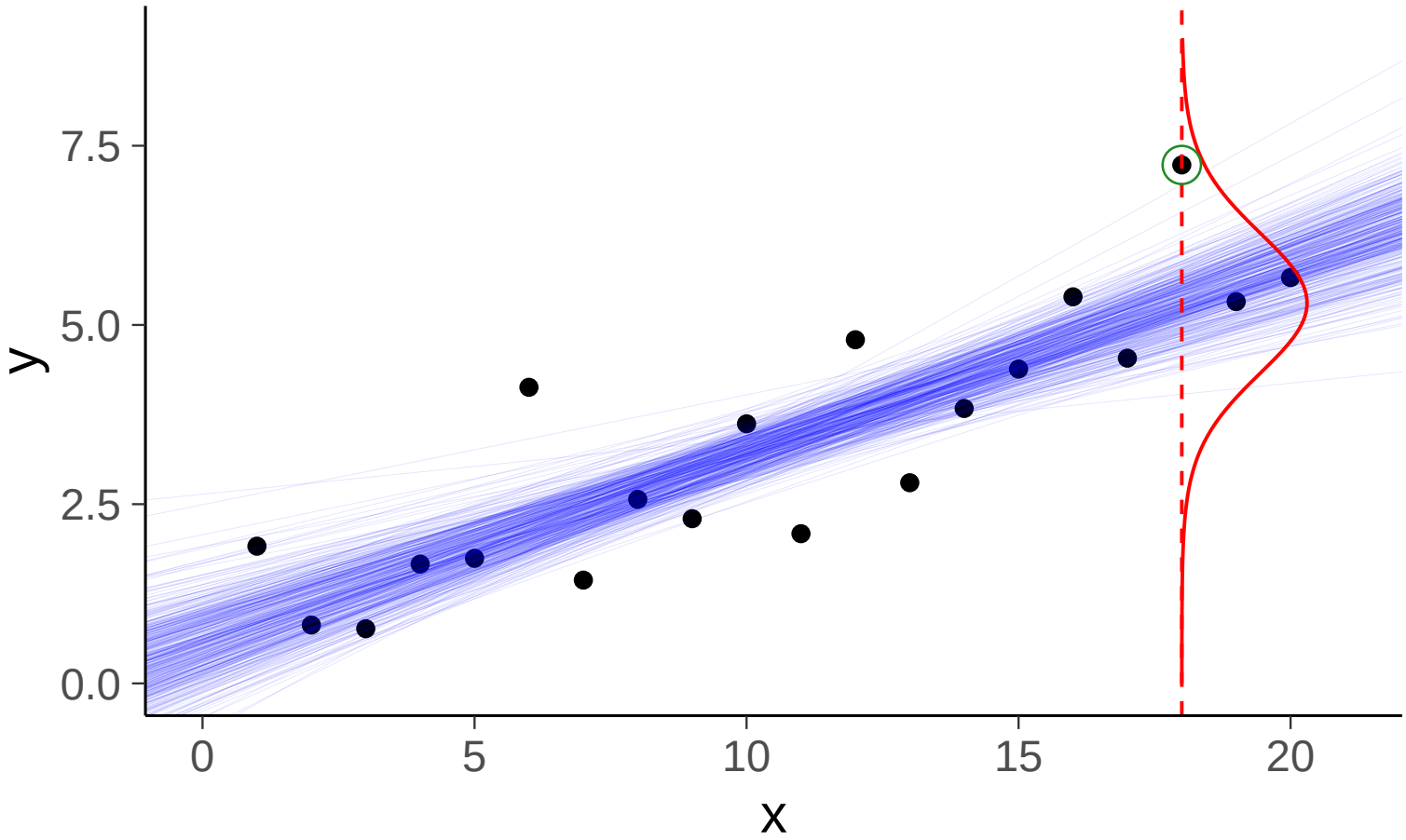
$$\theta^{(s)} \sim p(\theta \mid x, y)$$

Posterior predictive distribution



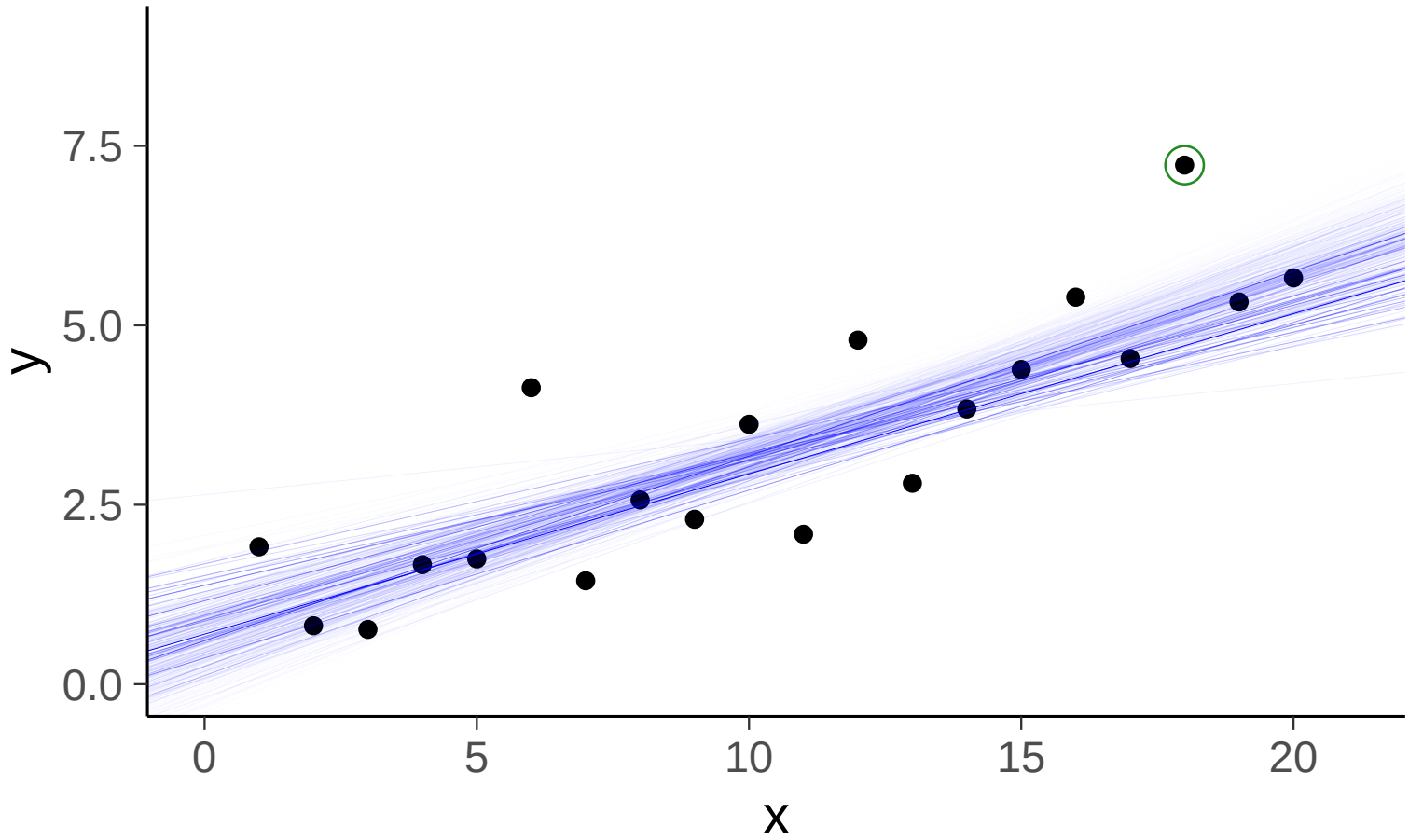
$$\theta^{(s)} \sim p(\theta \mid x, y), \quad p(\tilde{y} \mid \tilde{x}, x, y) \approx \frac{1}{S} \sum_{s=1}^{S} p(\tilde{y} \mid \tilde{x}, \theta^{(s)})$$

Posterior predictive distribution



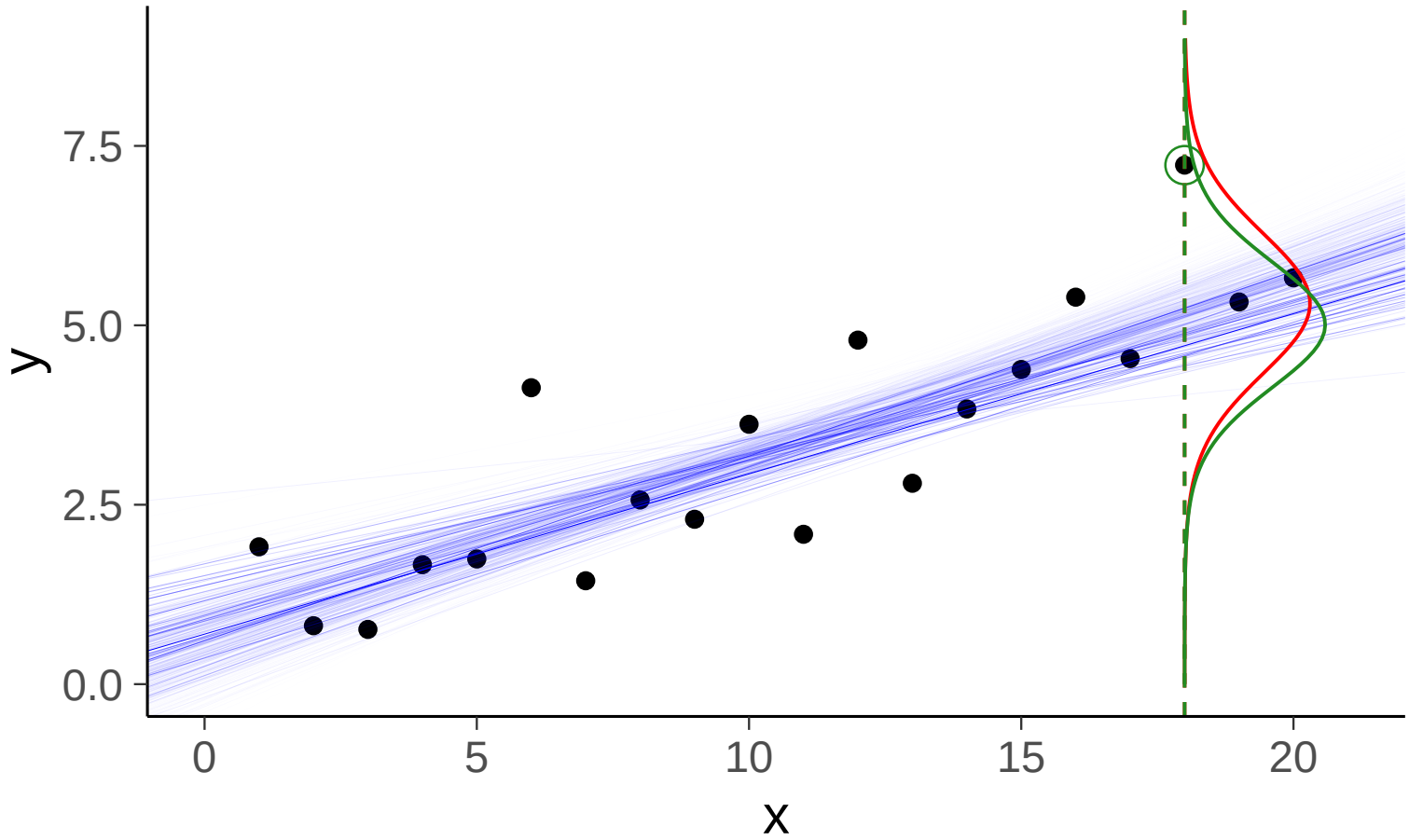
$$\theta^{(s)} \sim p(\theta \mid x, y), \quad p(\tilde{y} \mid \tilde{x}, x, y) \approx \frac{1}{S} \sum_{s=1}^{S} p(\tilde{y} \mid \tilde{x}, \theta^{(s)})$$

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



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

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

Pareto smoothed importance sampling LOO

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

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

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

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

Pareto smoothed importance sampling LOO

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

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

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

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

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

Pareto smoothed importance sampling LOO

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

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

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

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

Pareto smoothed importance sampling LOO

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

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

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

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

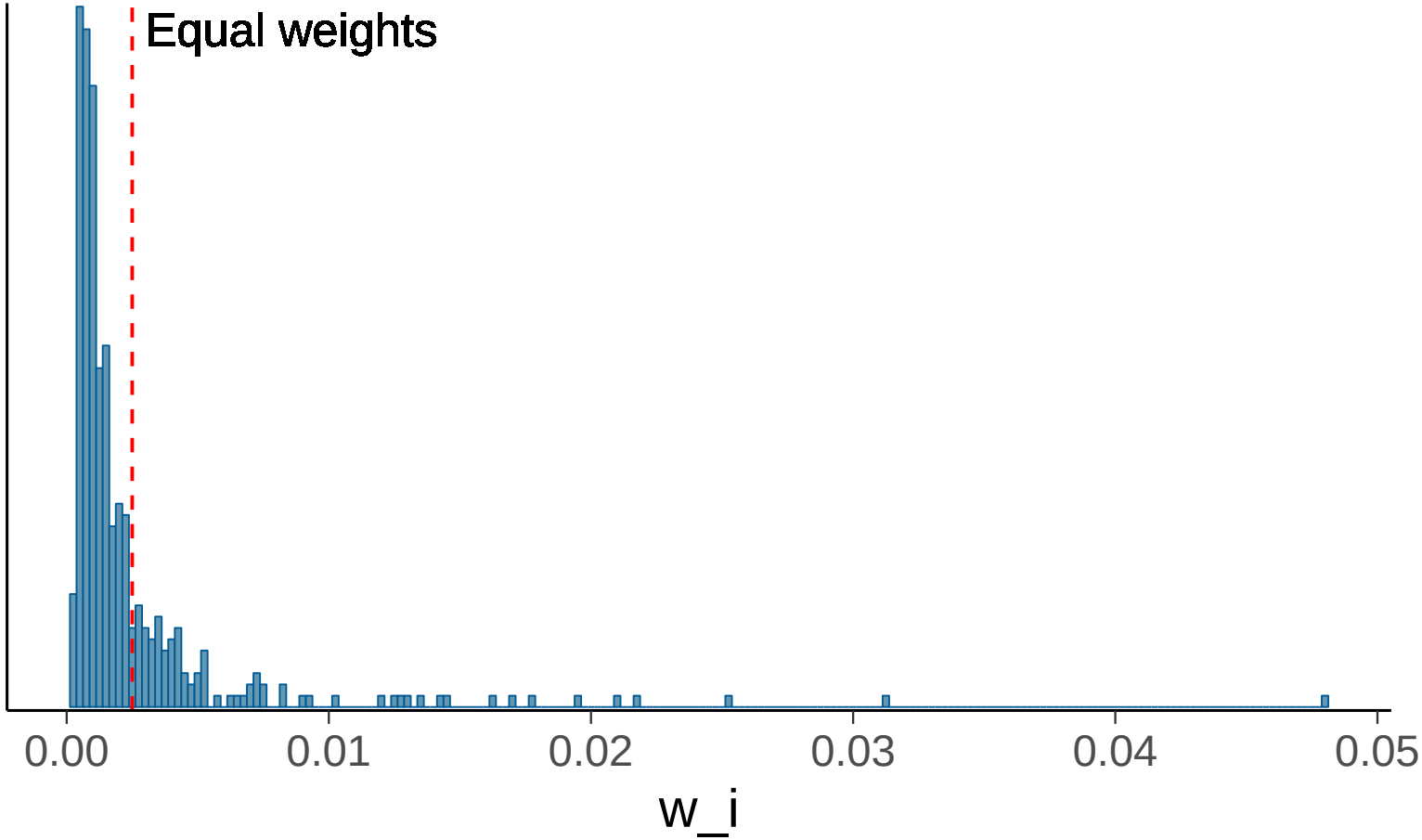
Pareto smoothed importance sampling LOO

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

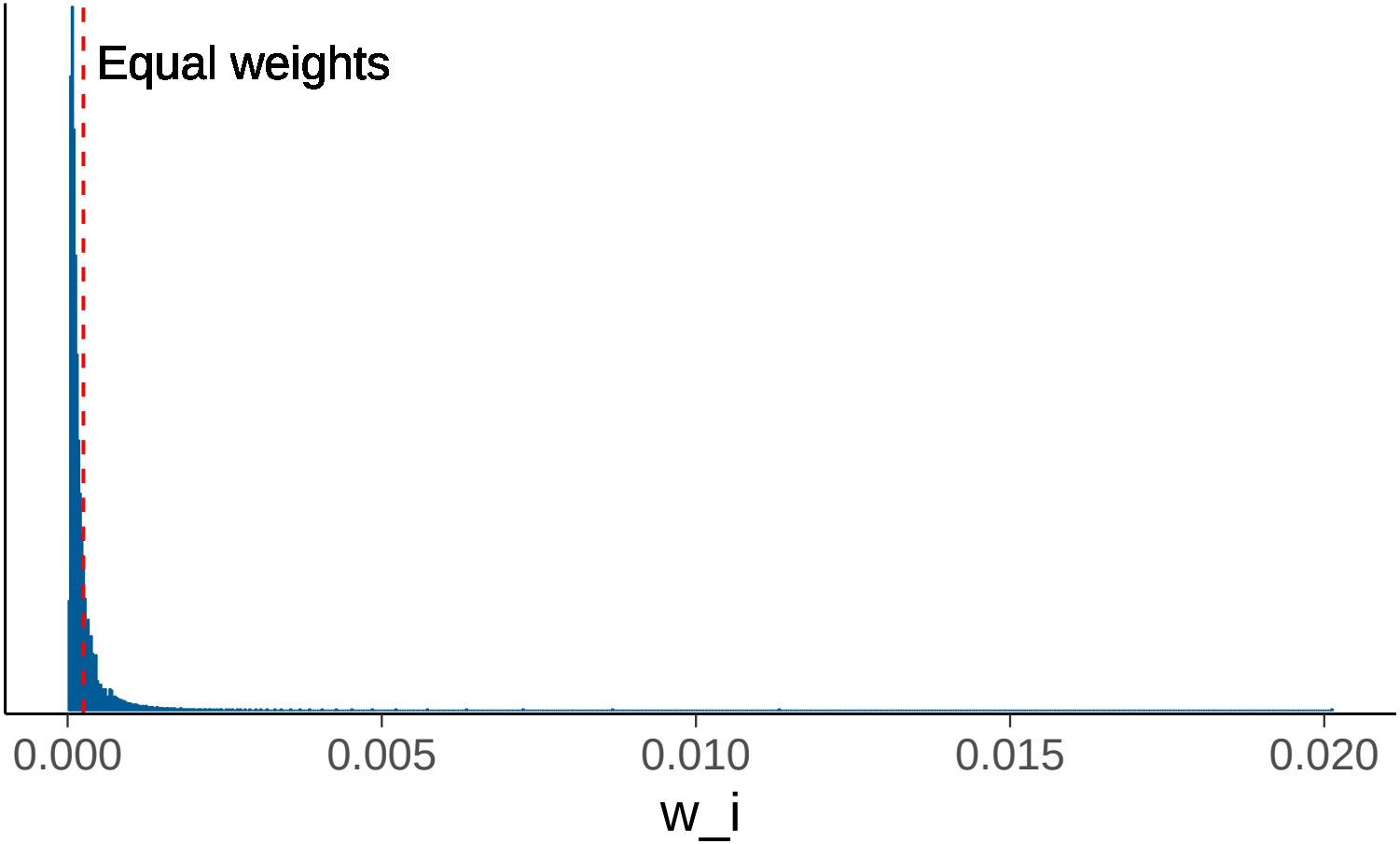
$$p(y_i \mid x_i, x_{-i}, y_{-i}) \approx \sum_{s=1}^{S} \left[\tilde{w}_i^{(s)} p(y_i \mid x_i, \theta^{(s)}) \right]$$
$$\approx \frac{1}{\frac{1}{S} \sum_{s'=1}^{S} w_i^{(s')}}$$

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

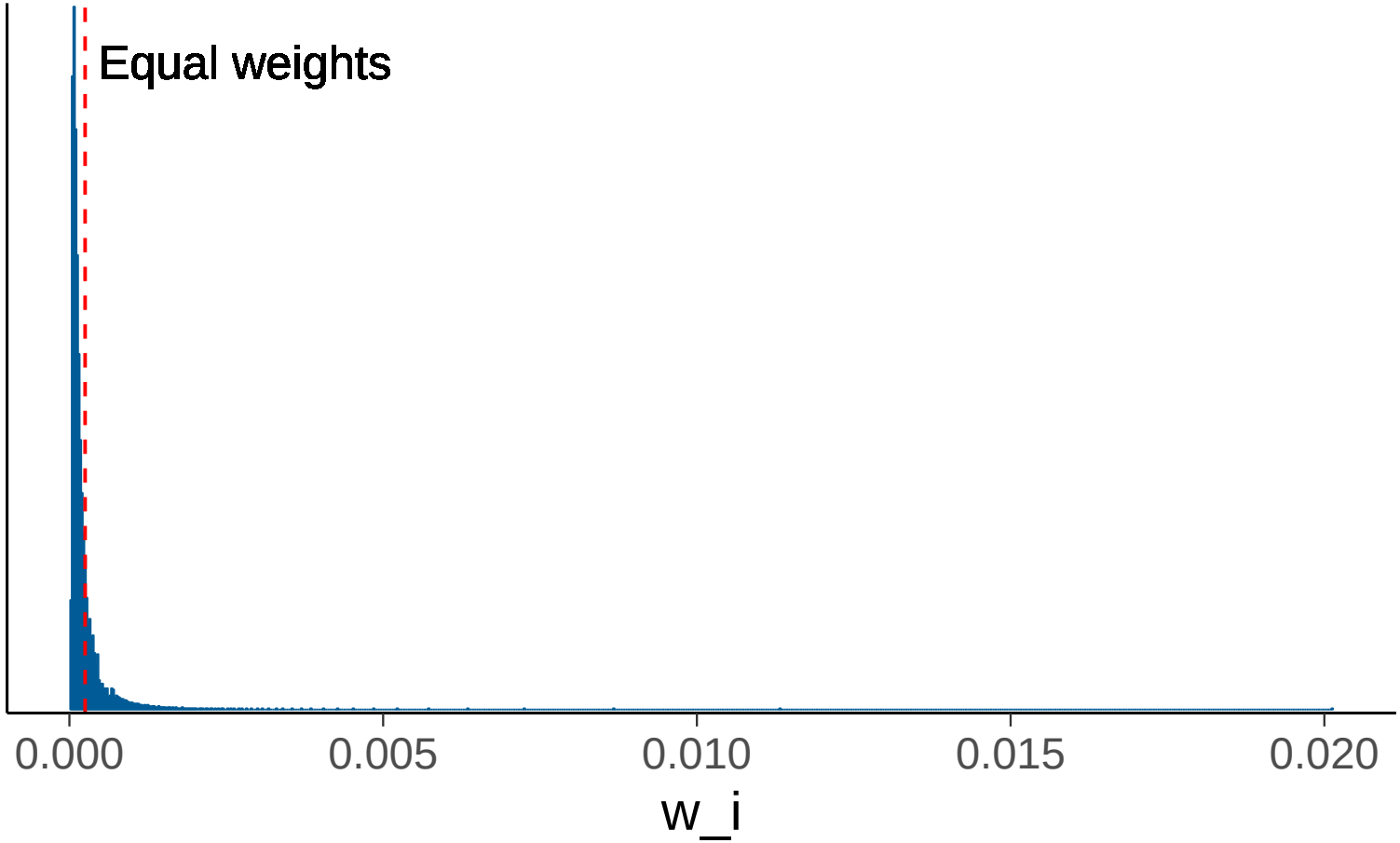
400 importance weights for leave−18th−out



4000 importance weights for leave−18th−out



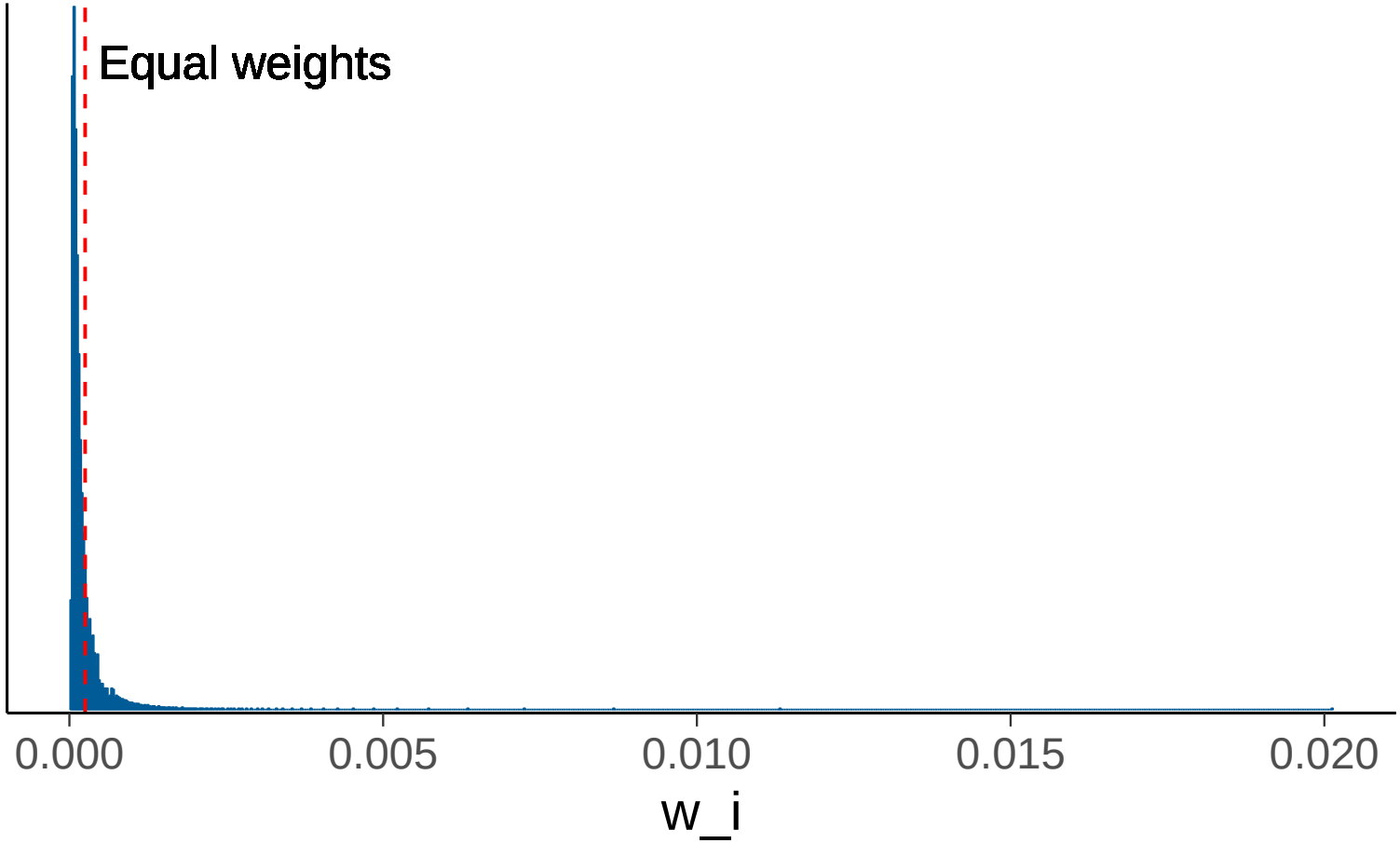
4000 importance weights for leave−18th−out



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

see Vehtari, Gelman & Gabry (2017b)

4000 importance weights for leave−18th−out

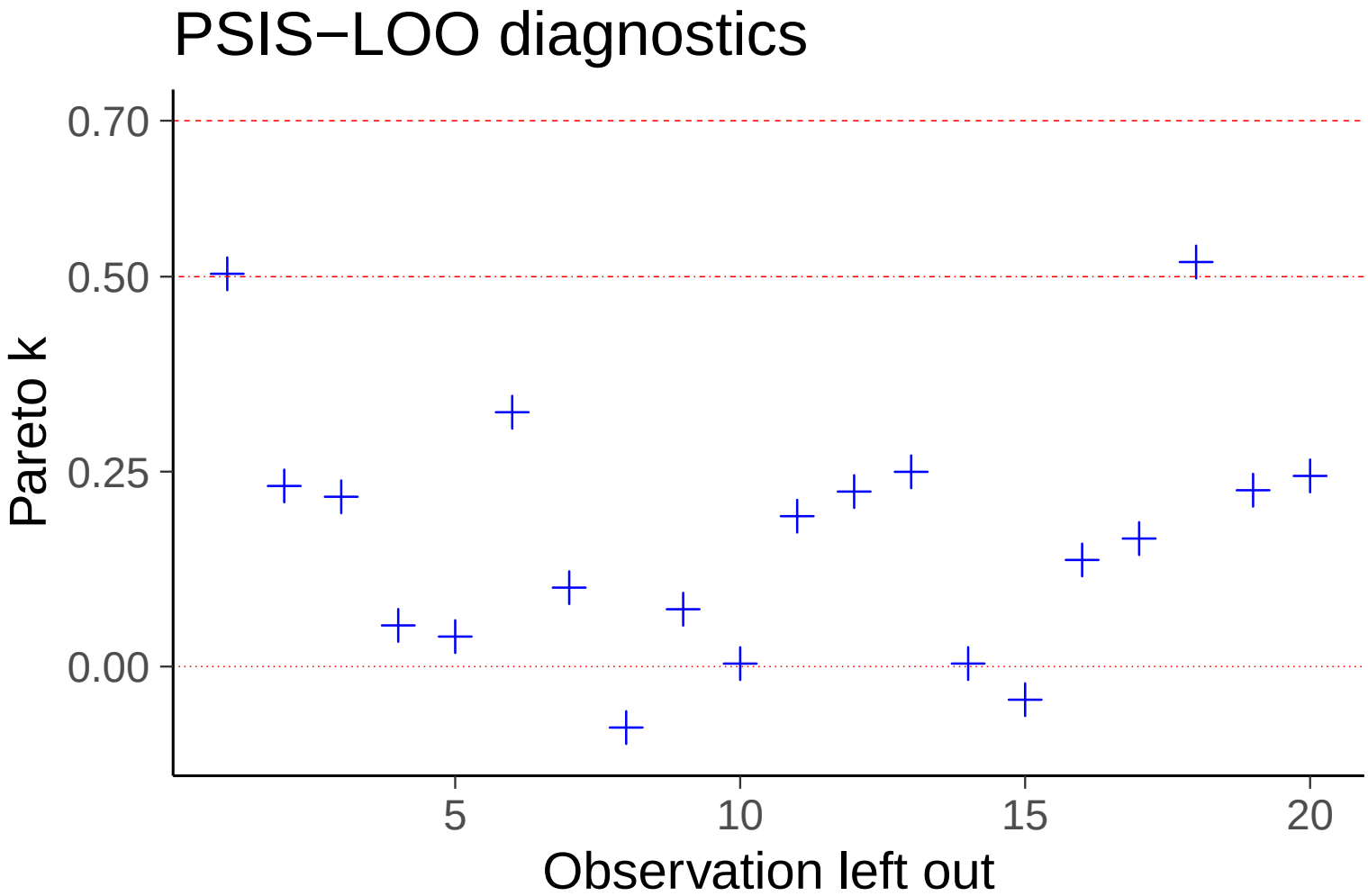


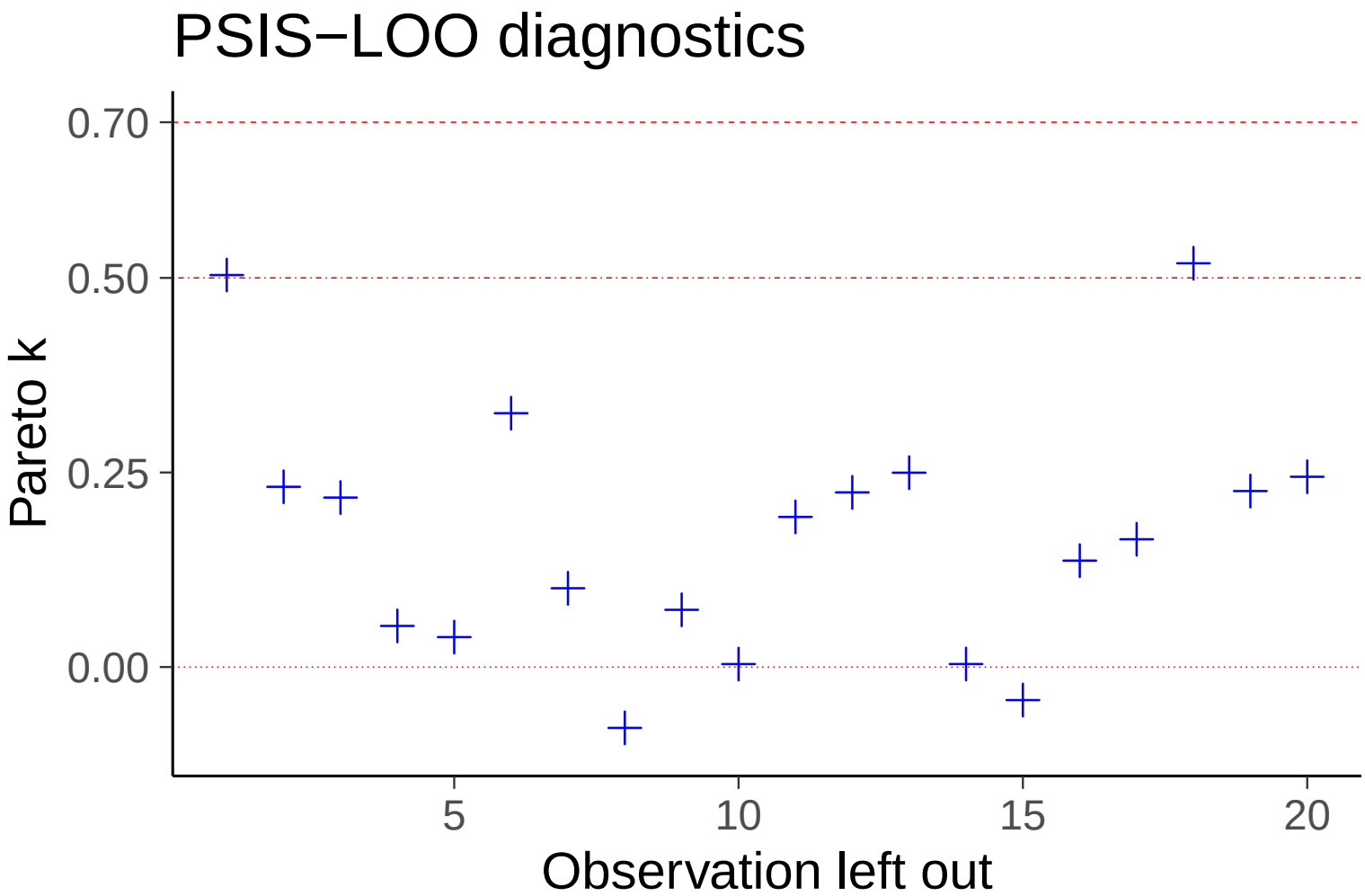
ESS $\approx 1/\sum_{s=1}^{S}(\tilde{w}^{(s)})^2 \approx 459$

Pareto $\hat{k} \approx 0.52$

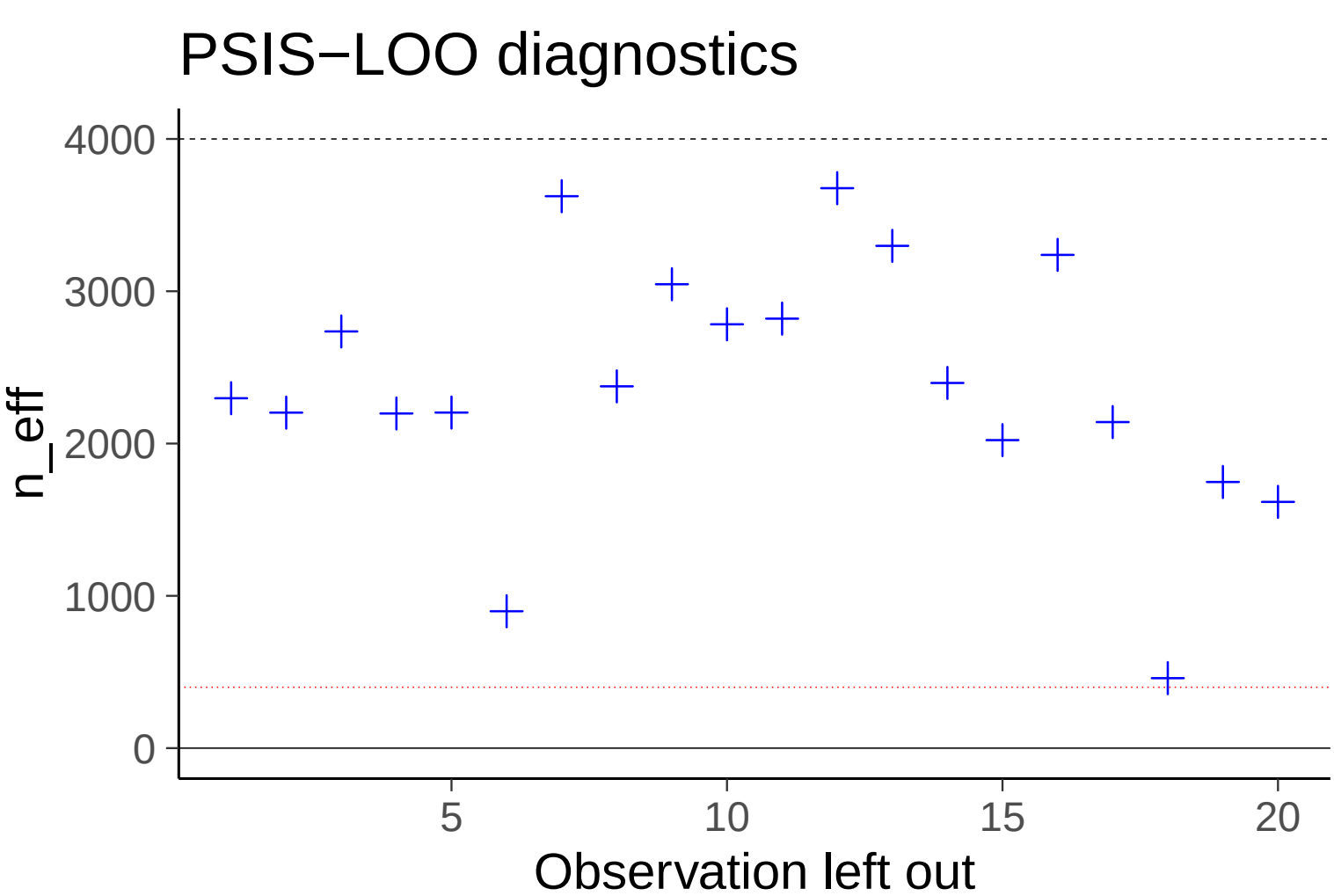
- Pareto $\hat{k}$ estimates the tail shape which determines the convergence rate of PSIS. Less than 0.7 is ok.

see Vehtari, Gelman & Gabry (2017b)


PSIS−LOO diagnostics
0.70
0.50
0.25
0.00
Pareto k
5
10
15
20
Observation left out




```
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

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

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), RStanARM, brms `loo(fit)`

loo()

- RStan (`log_lik` in gen. quantities), RStanARM, brms
  `loo(fit)`
- CmdStanR (eventually will work as others)

```
loocmd <- function(fit, ...)  {
 loo(fit$draws("log_lik"),
r_eff=relative_eff(fit$draws("log_lik"))),
...
}
loocmd(fit)
```

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 = TRUE)`
  `brms::loo(..., k_threshold = TRUE)`
  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 = TRUE)`
  `brms::loo(..., k_threshold = TRUE)`
  runs MCMC for the folds with $\hat{k}$ above the threshold
- Use K-fold-CV (more about this soon)

Pareto smoothed importance sampling CV variants

- PSIS-LOO for hierarchical models
  - leave-one-group out is challenging for PSIS-LOO
  - see Merkel, Furr and Rabe-Hesketh (2018) for an approach using quadrature integration
  - Stan demo of the challenges and integrated LOO at https://avehtari.github.io/modelselection/roaches.html

Pareto smoothed importance sampling CV variants

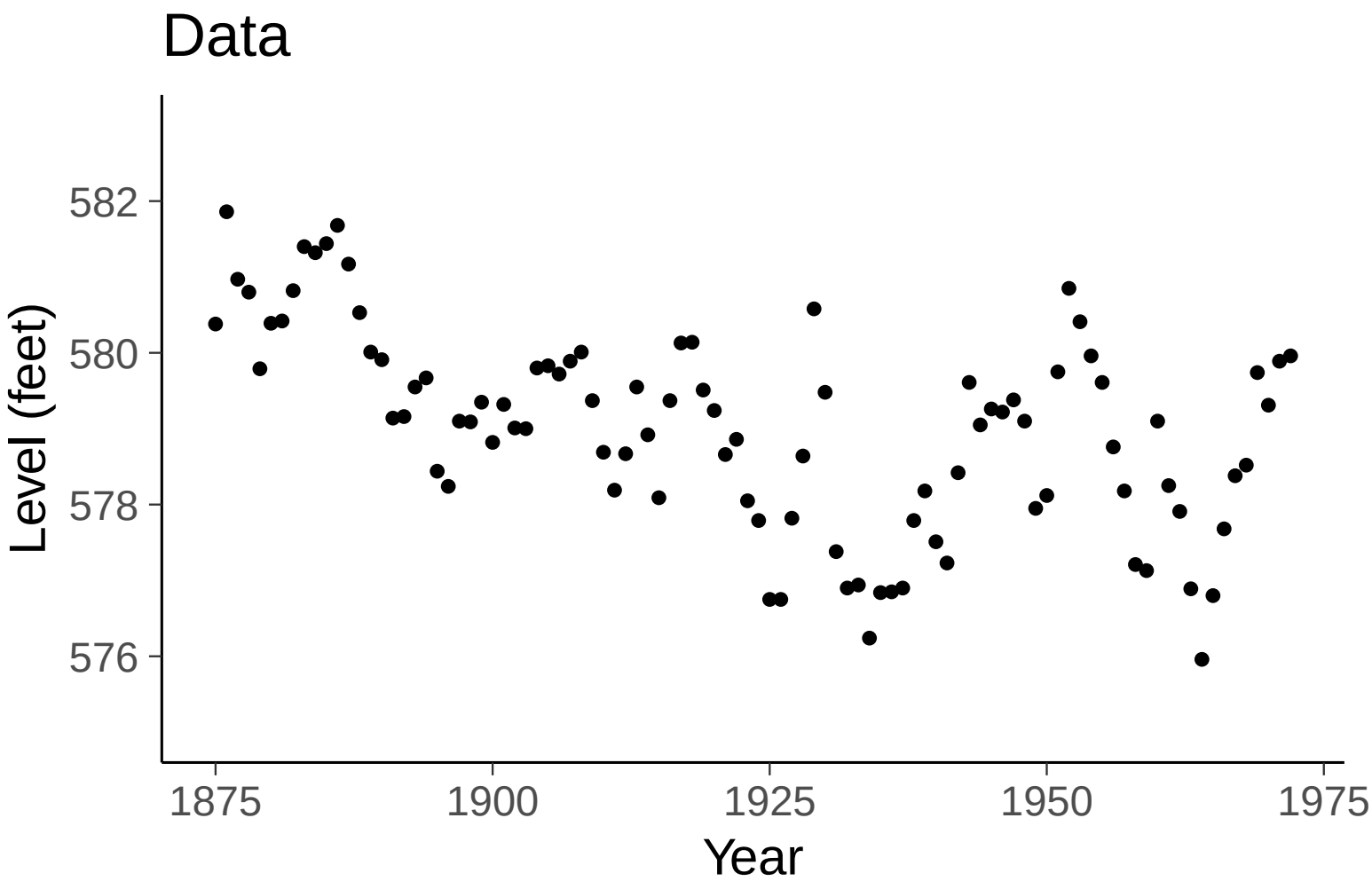
- PSIS-LOO for hierarchical models
  - leave-one-group out is challenging for PSIS-LOO
  - see Merkel, Furr and Rabe-Hesketh (2018) for an approach using quadrature integration
  - Stan demo of the challenges and integrated LOO at https://avehtari.github.io/modelselection/roaches.html
- PSIS-LOO for non-factorizable models
  - mc-stan.org/loo/articles/loo2-non-factorizable.html

Pareto smoothed importance sampling CV variants

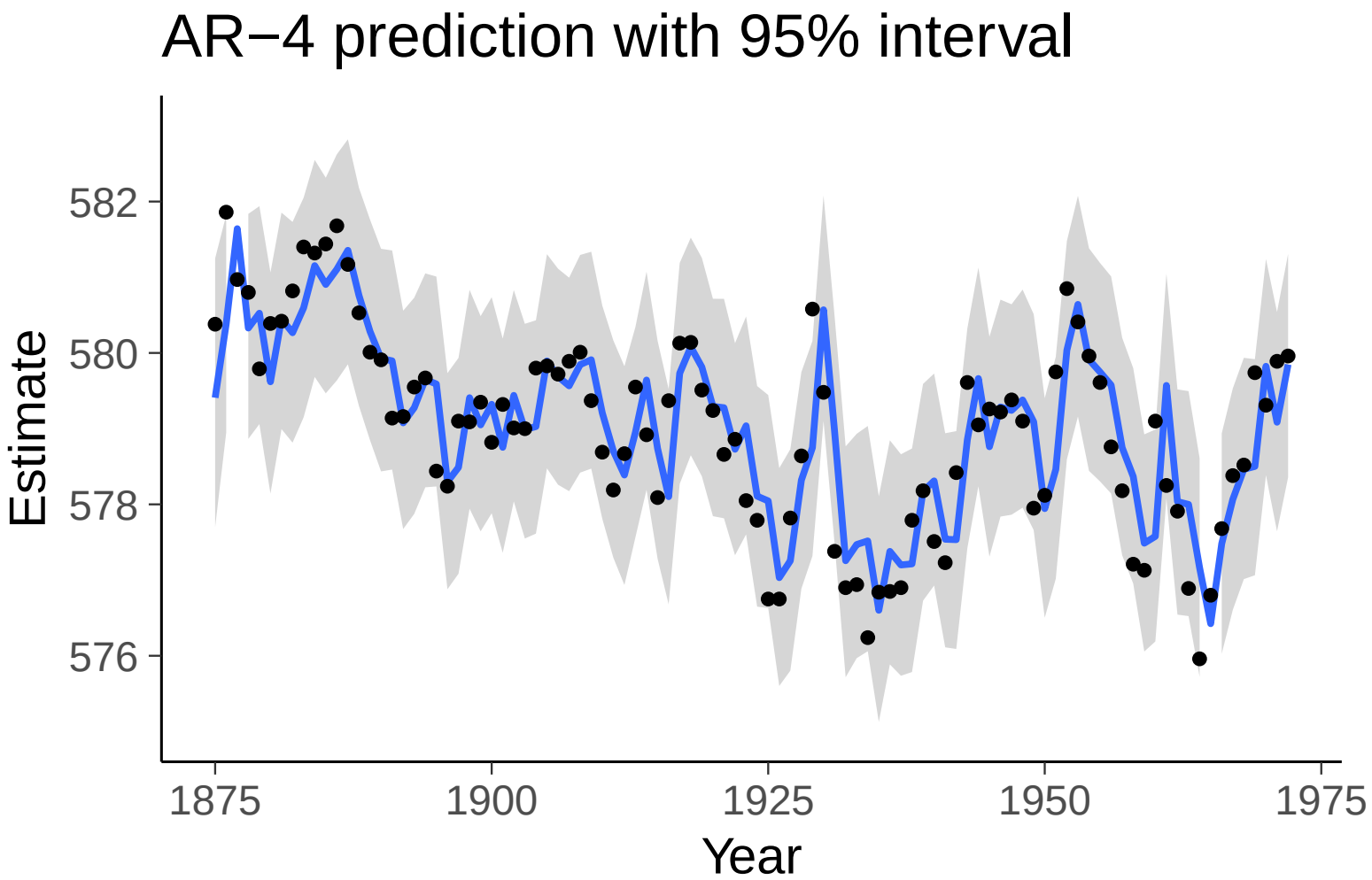
- PSIS-LOO for hierarchical models
  - leave-one-group out is challenging for PSIS-LOO
  - see Merkel, Furr and Rabe-Hesketh (2018) for an approach using quadrature integration
  - Stan demo of the challenges and integrated LOO at https://avehtari.github.io/modelselection/roaches.html
- PSIS-LOO for non-factorizable 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

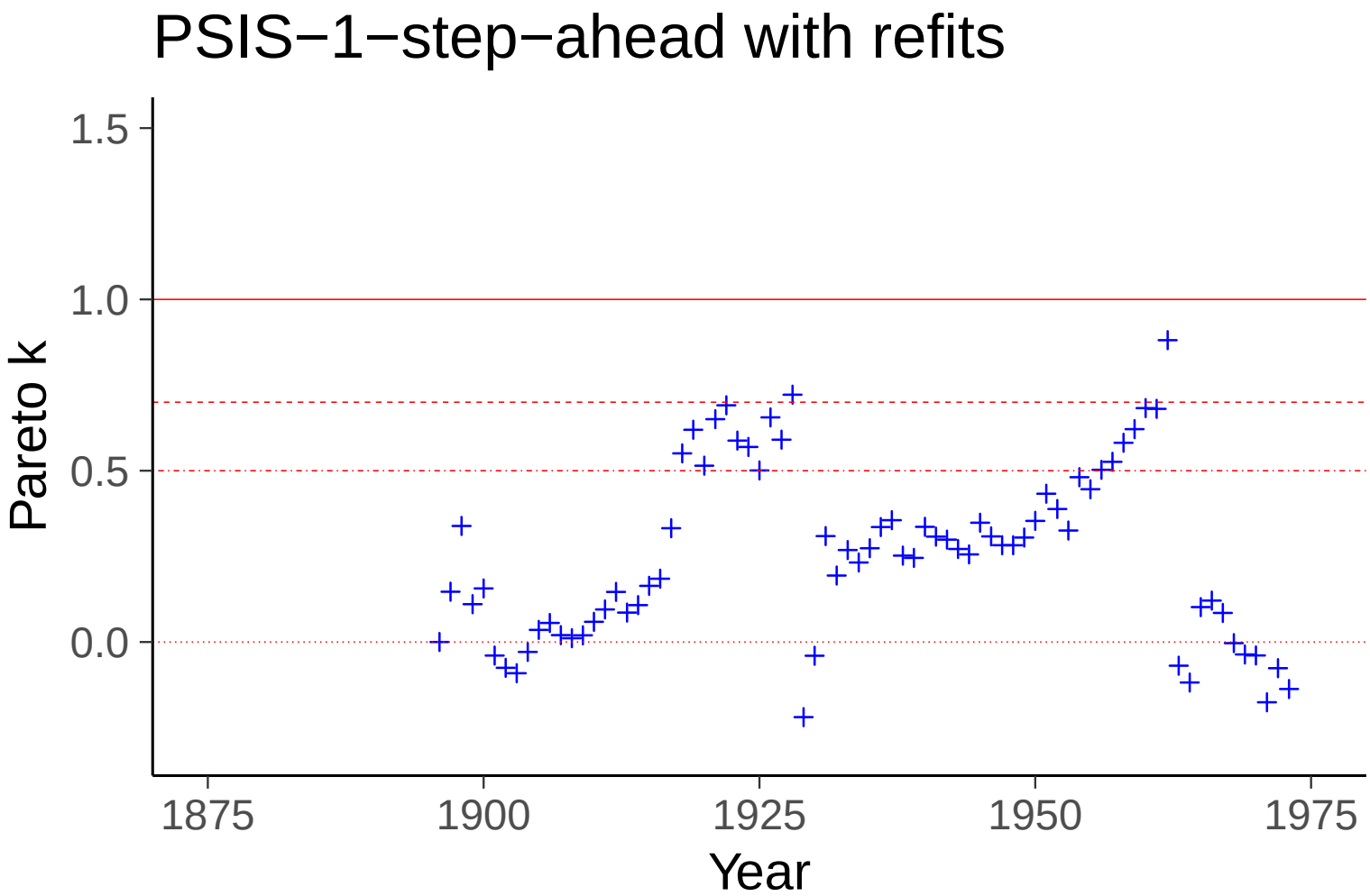
LFO-CV



LFO-CV


AR−4 prediction with 95% interval
Estimate
582
580
578
576
1875
1900
1925
1950
1975
Year

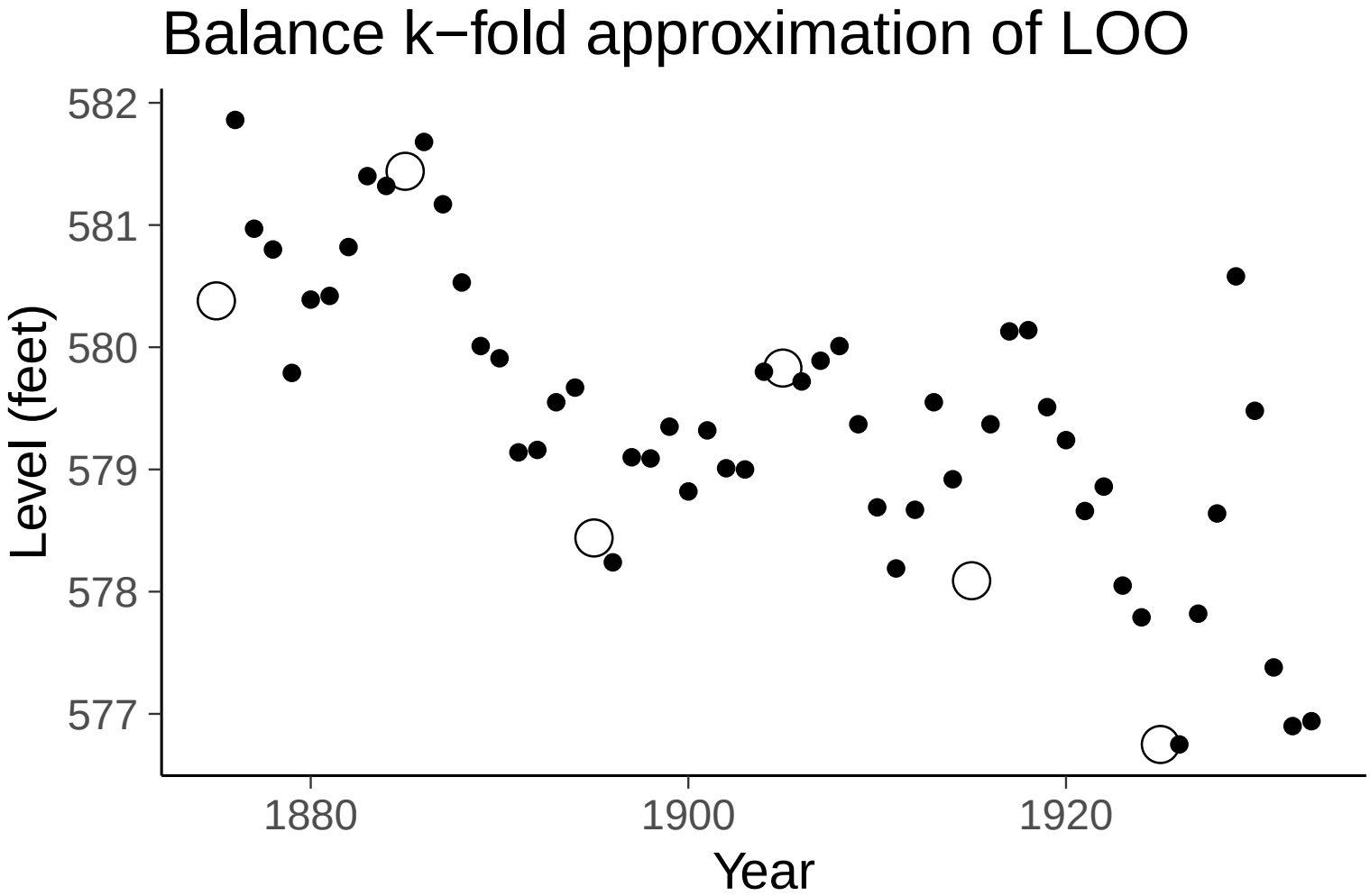
LFO-CV

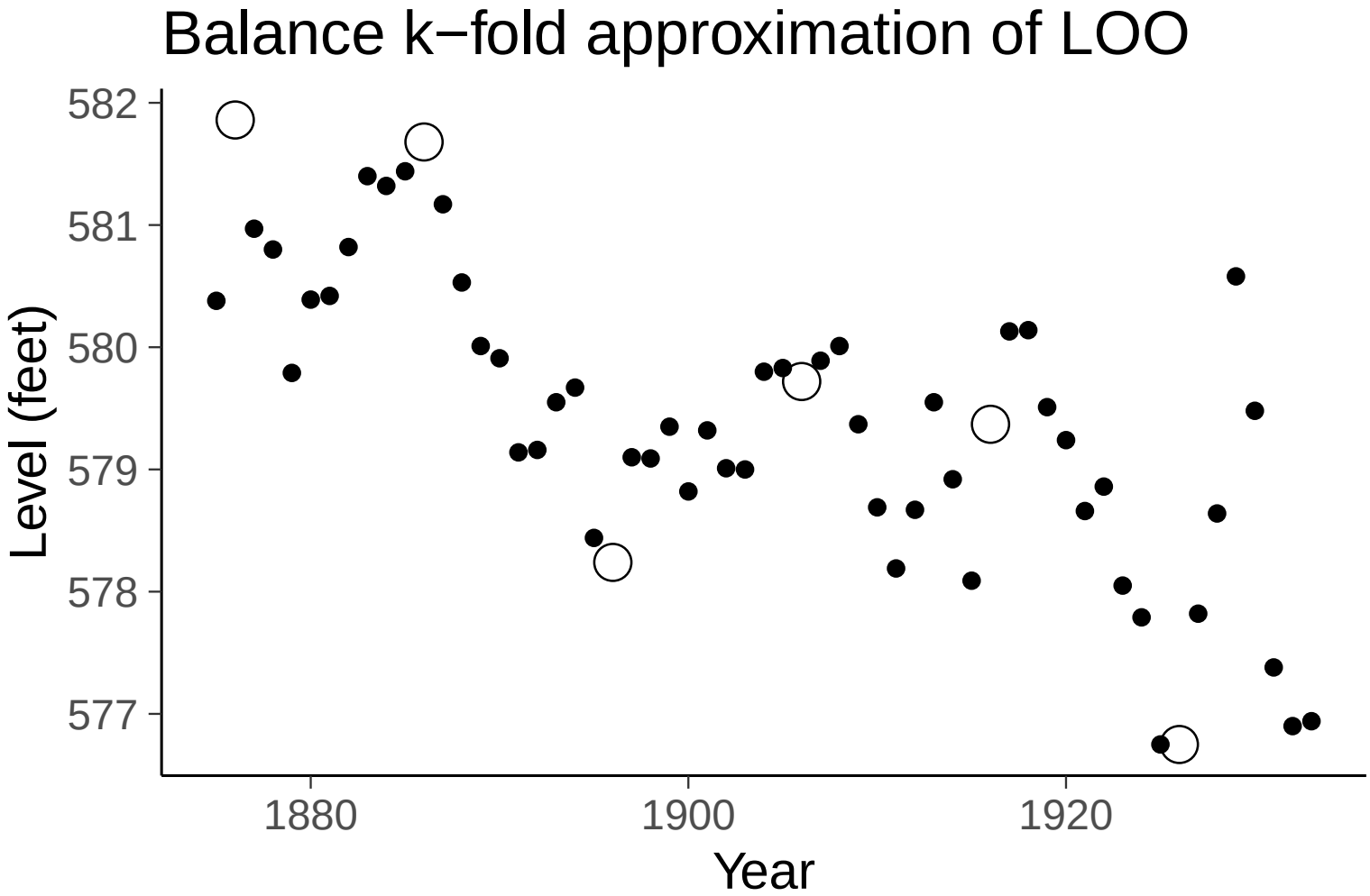


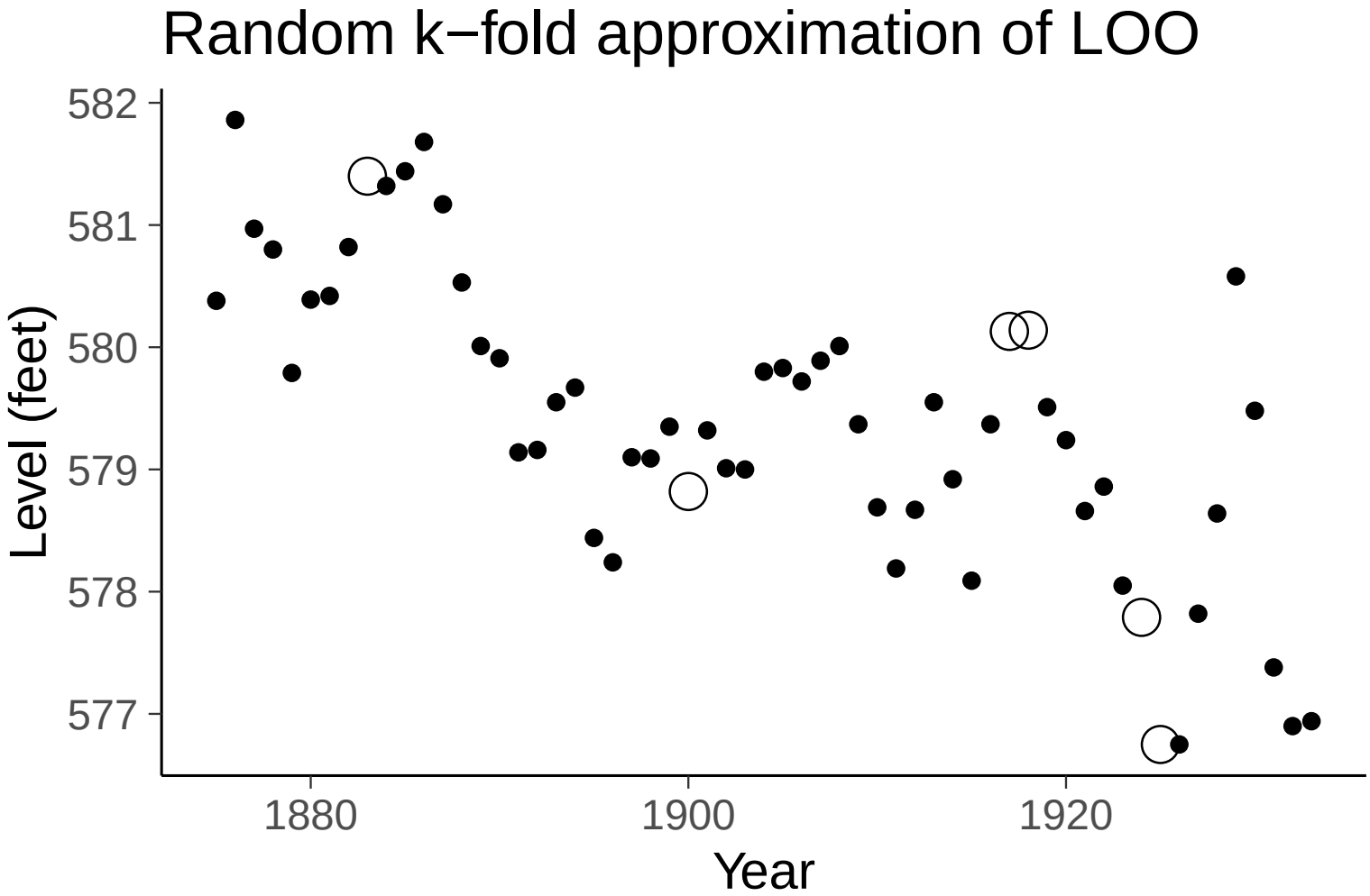
mc-stan.org/loo/articles/loo2-lfo.html

K-fold cross-validation

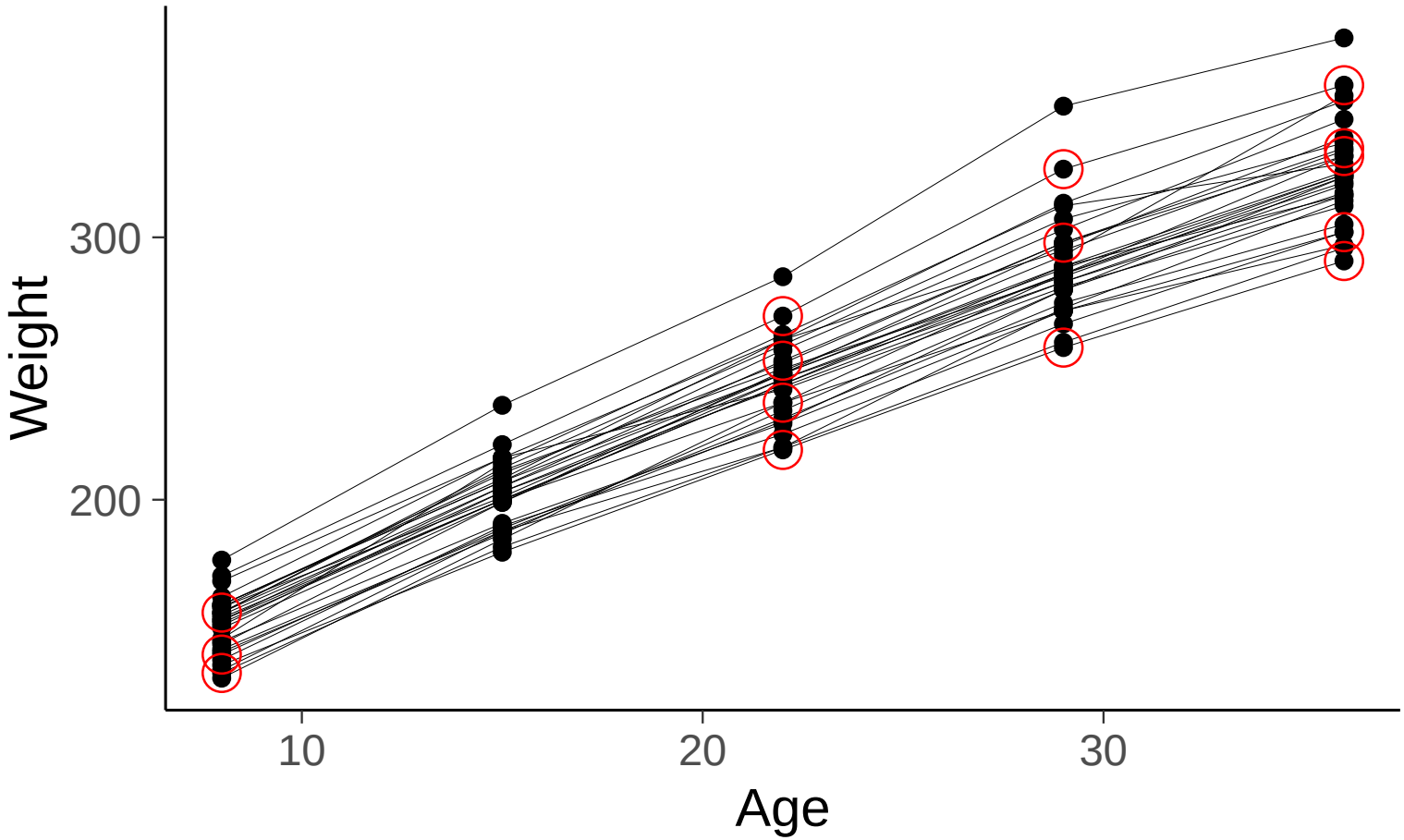
- K-fold cross-validation can approximate LOO
  - the same use cases as with LOO
- K-fold cross-validation can be used for hierarchical models
  - good for leave-one-group-out
- K-fold cross-validation can be used for time series
  - with leave-block-out

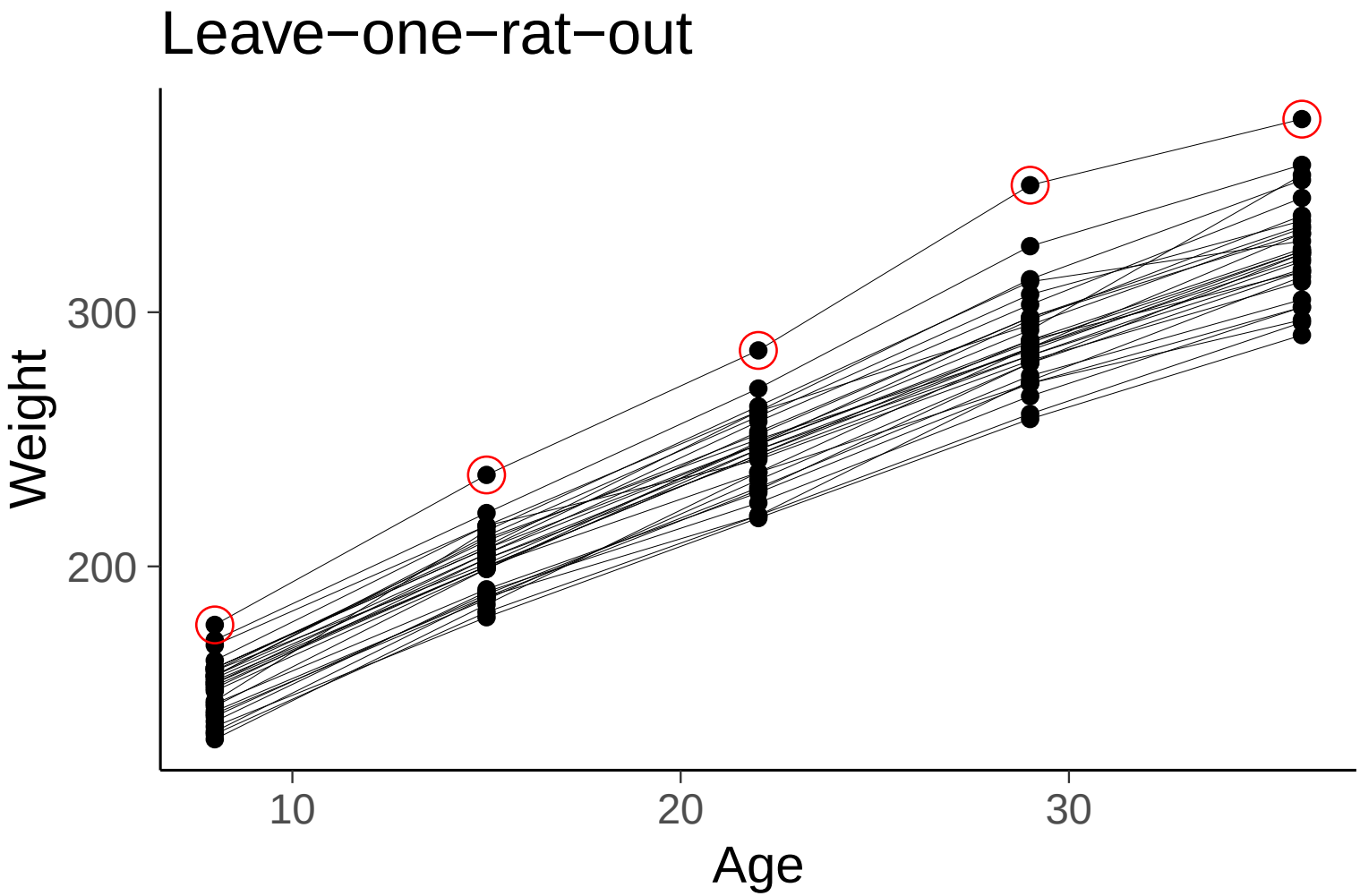

Balance k−fold approximation of LOO
582
581
580
579
578
577
Level (feet)
1880
1900
1920
Year


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


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

Random kfold approximation of LOO




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()`

WAIC vs PSIS-LOO

see Vehtari, Gelman & Gabry (2017a)

WAIC vs PSIS-LOO

- WAIC has same assumptions as LOO

see Vehtari, Gelman & Gabry (2017a)

WAIC vs PSIS-LOO

- WAIC has same assumptions as LOO
- PSIS-LOO is more accurate

see Vehtari, Gelman & Gabry (2017a)

WAIC vs PSIS-LOO

- WAIC has same assumptions as LOO
- PSIS-LOO is more accurate
- PSIS-LOO has much better diagnostics

see Vehtari, Gelman & Gabry (2017a)

WAIC vs PSIS-LOO

- WAIC has same assumptions as LOO
- PSIS-LOO is more accurate
- PSIS-LOO has much better diagnostics
- LOO makes the prediction assumption more clear, which helps if K-fold-CV is needed instead

see Vehtari, Gelman & Gabry (2017a)

WAIC vs PSIS-LOO

- WAIC has same assumptions as LOO
- PSIS-LOO is more accurate
- PSIS-LOO has much better diagnostics
- LOO makes the prediction assumption more clear, which helps if K-fold-CV is needed instead
- Multiplying by -2 doesn't give any benefit (Watanabe didn't multiply by -2)

see Vehtari, Gelman & Gabry (2017a)

*IC

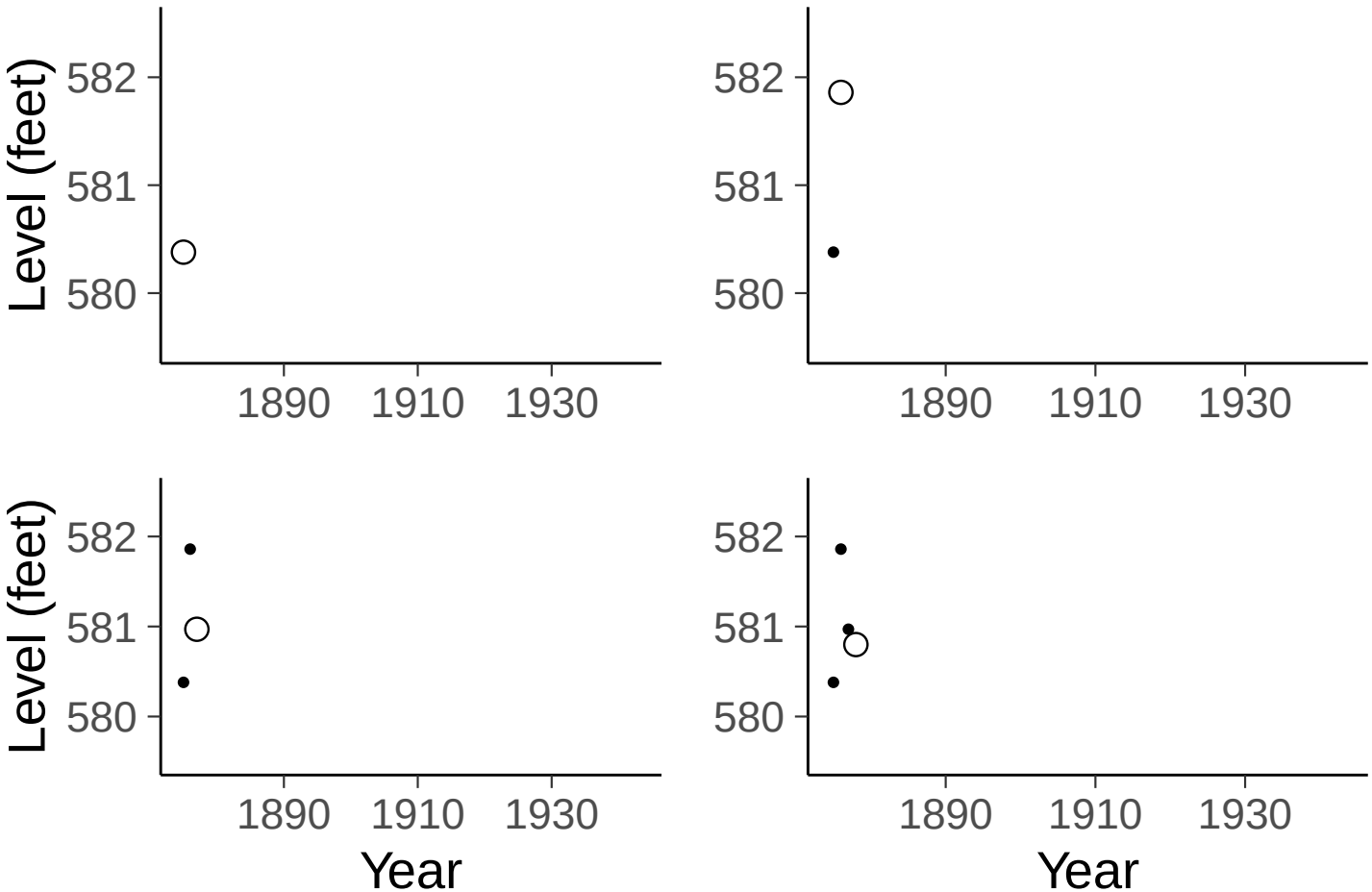
- AIC uses maximum likelihood estimate for prediction
- DIC uses posterior mean for prediction
- BIC is an approximation for marginal likelihood
- TIC, NIC, RIC, PIC, BPIC, QIC, AICc, ...

Marginal likelihood / Bayes factor

- Like leave-future-out 1-step-ahead cross-validation but starting with 0 observations

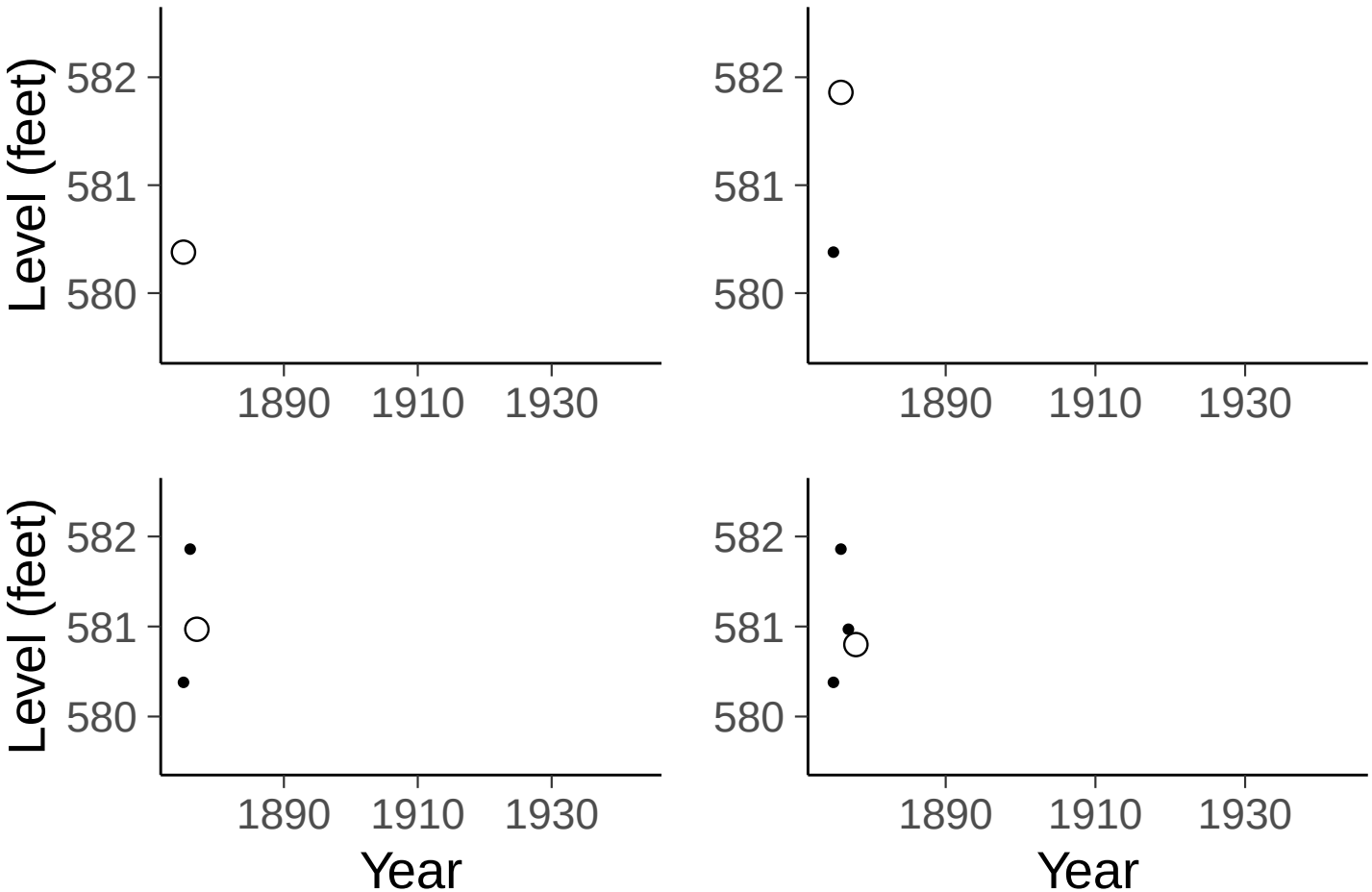
Marginal likelihood / Bayes factor

- Like leave-future-out 1-step-ahead cross-validation but starting with 0 observations



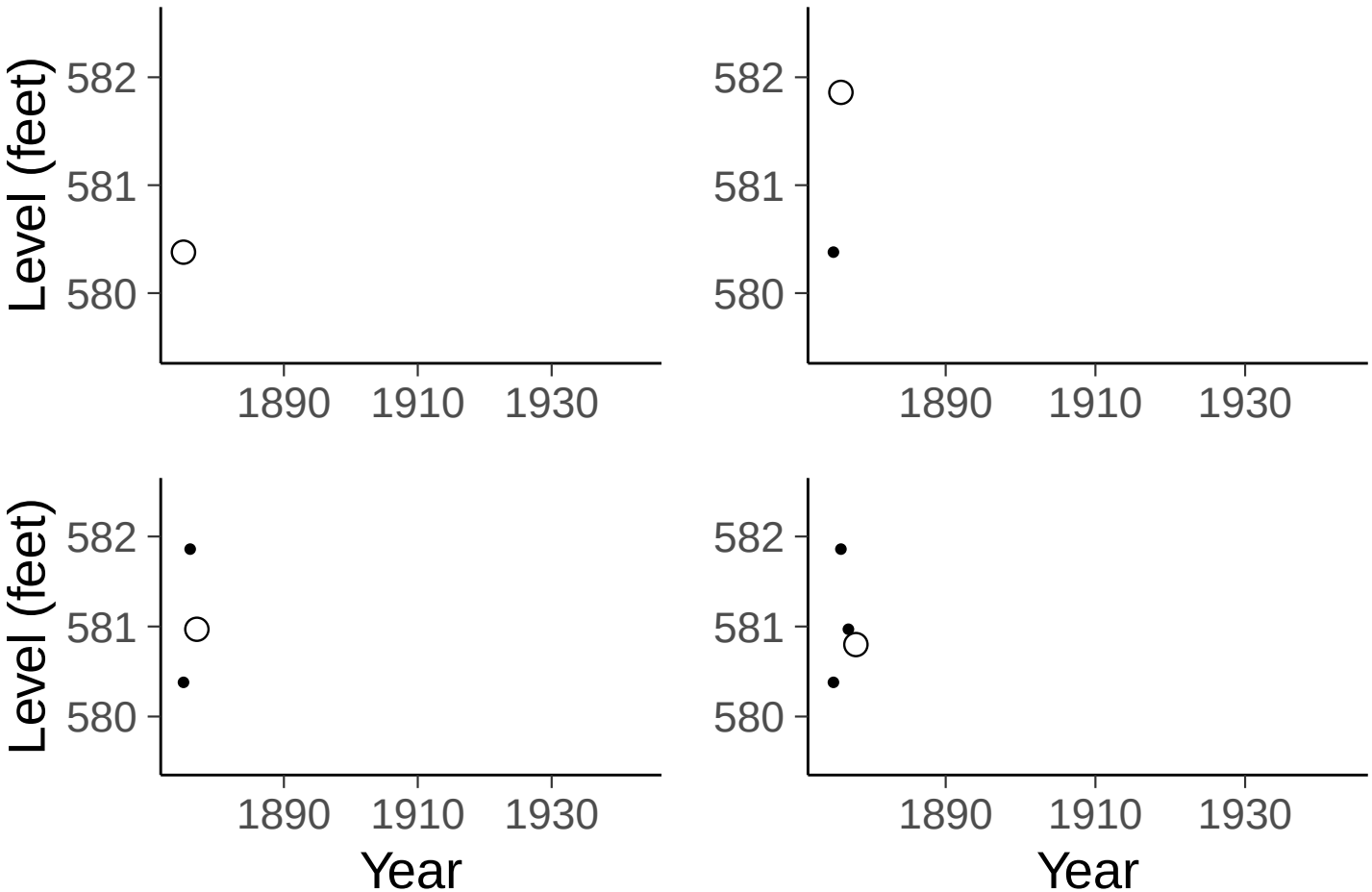
Marginal likelihood / Bayes factor

- Like leave-future-out 1-step-ahead cross-validation but starting with 0 observations
  - which makes it very sensitive to prior



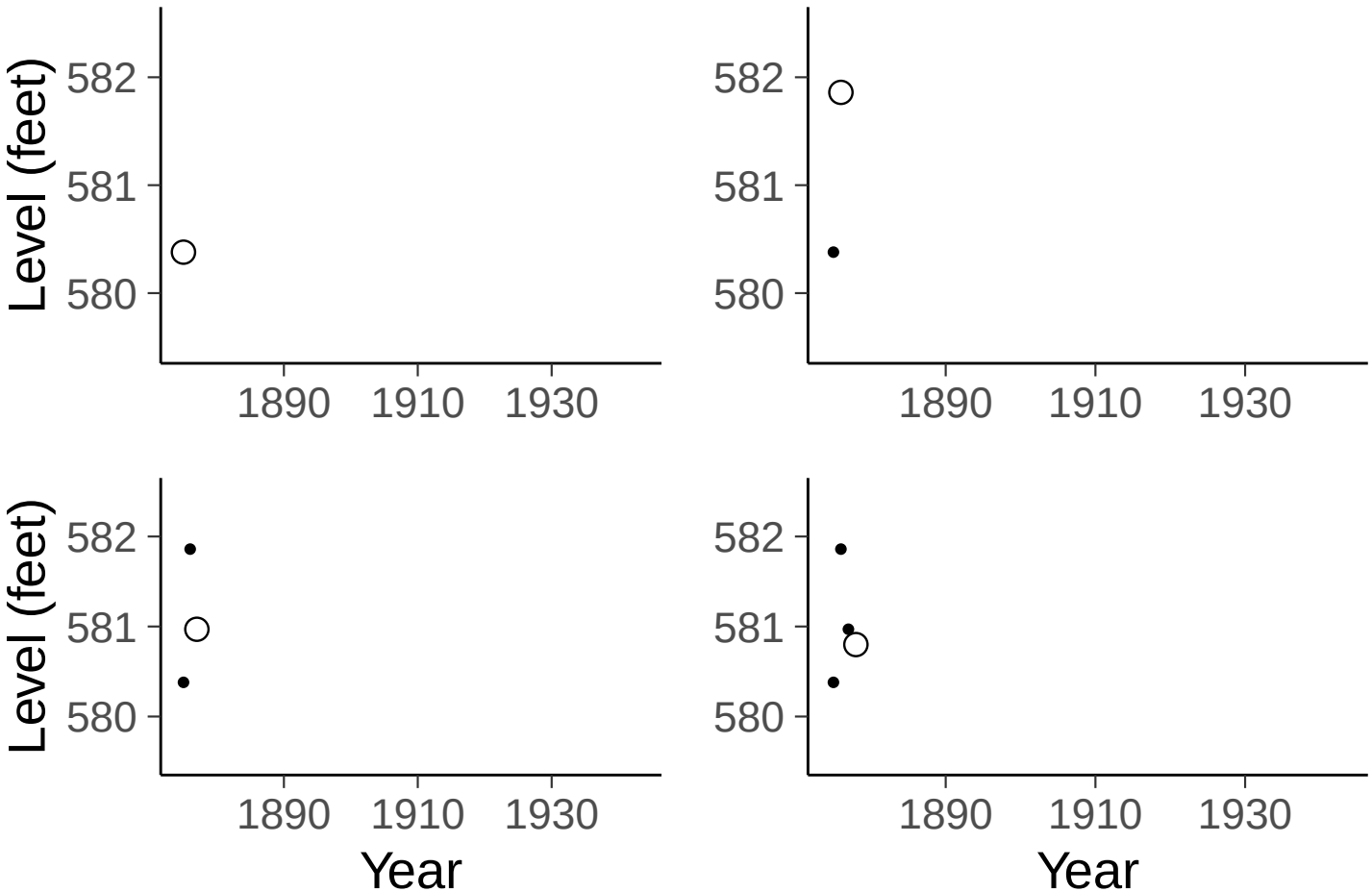
Marginal likelihood / Bayes factor

- Like leave-future-out 1-step-ahead cross-validation but starting with 0 observations
  - which makes it very sensitive to prior and
  - unstable in case of misspecified models



Marginal likelihood / Bayes factor

- Like leave-future-out 1-step-ahead cross-validation but starting with 0 observations
  - which makes it very sensitive to prior and
  - unstable in case of misspecified models also asymptotically



Cross-validation for model assessment

- CV is good for model assessment when application specific utility/cost functions are used
  - e.g. 90% absolute error

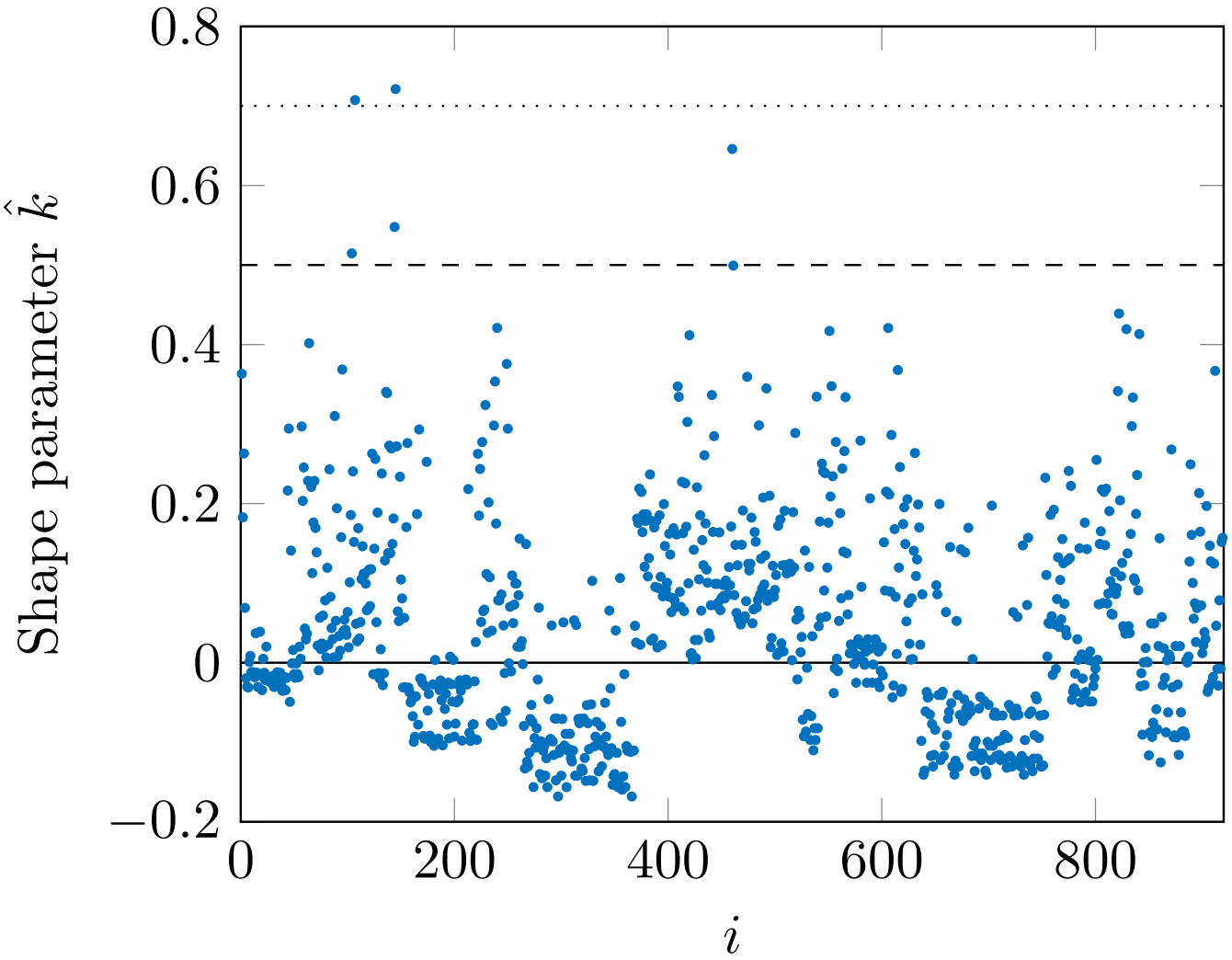
Cross-validation for model assessment

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

see demos avehtari.github.io/modelselection/

Radon example

PSIS-LOO diagnostics

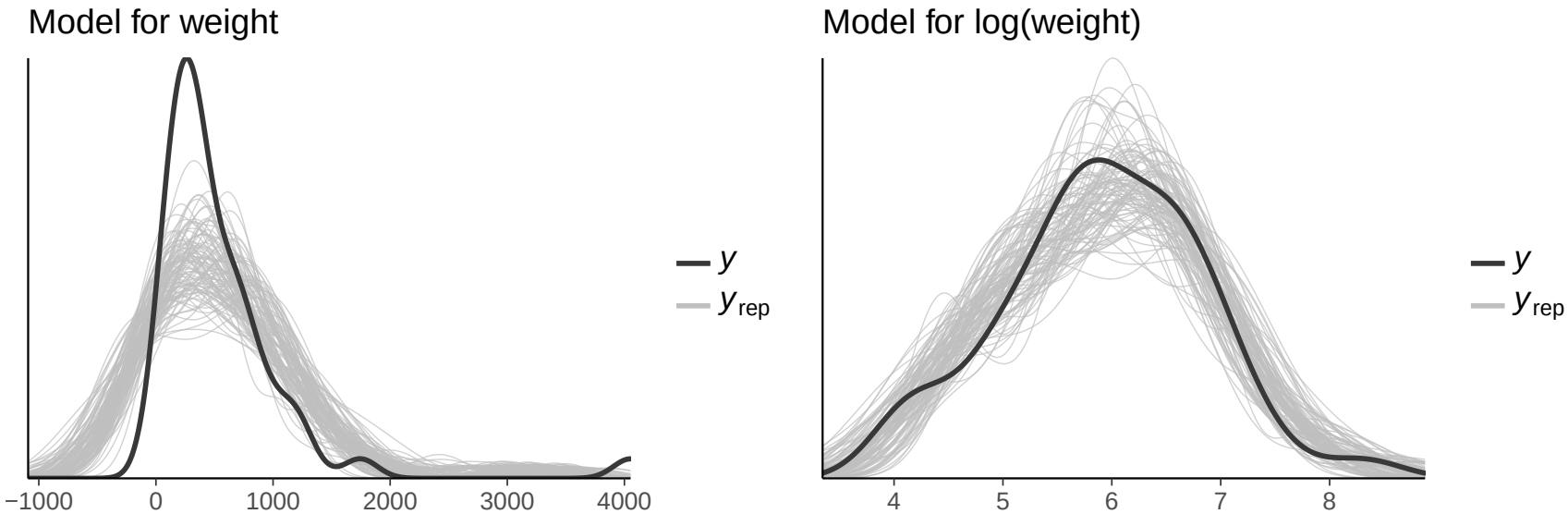


see Vehtari, Gelman & Gabry (2017a)

Sometimes cross-validation is not needed

Sometimes cross-validation is not needed

- Posterior predictive checking is often sufficient

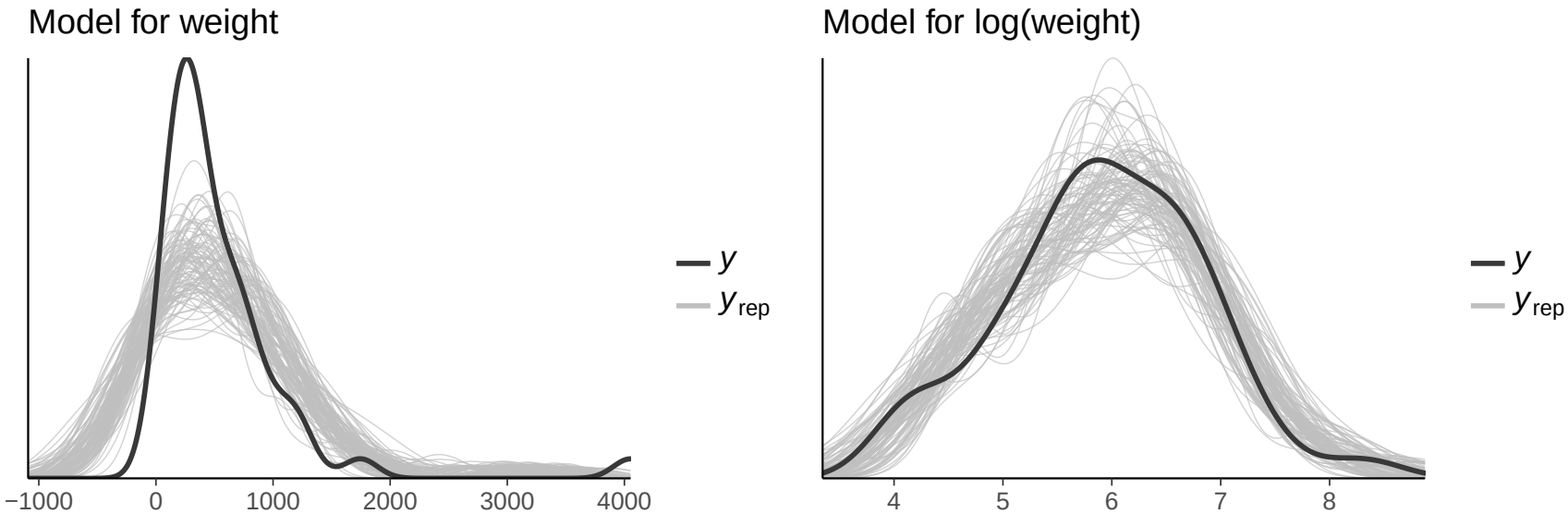


Predicting the yields of mesquite bushes.

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.

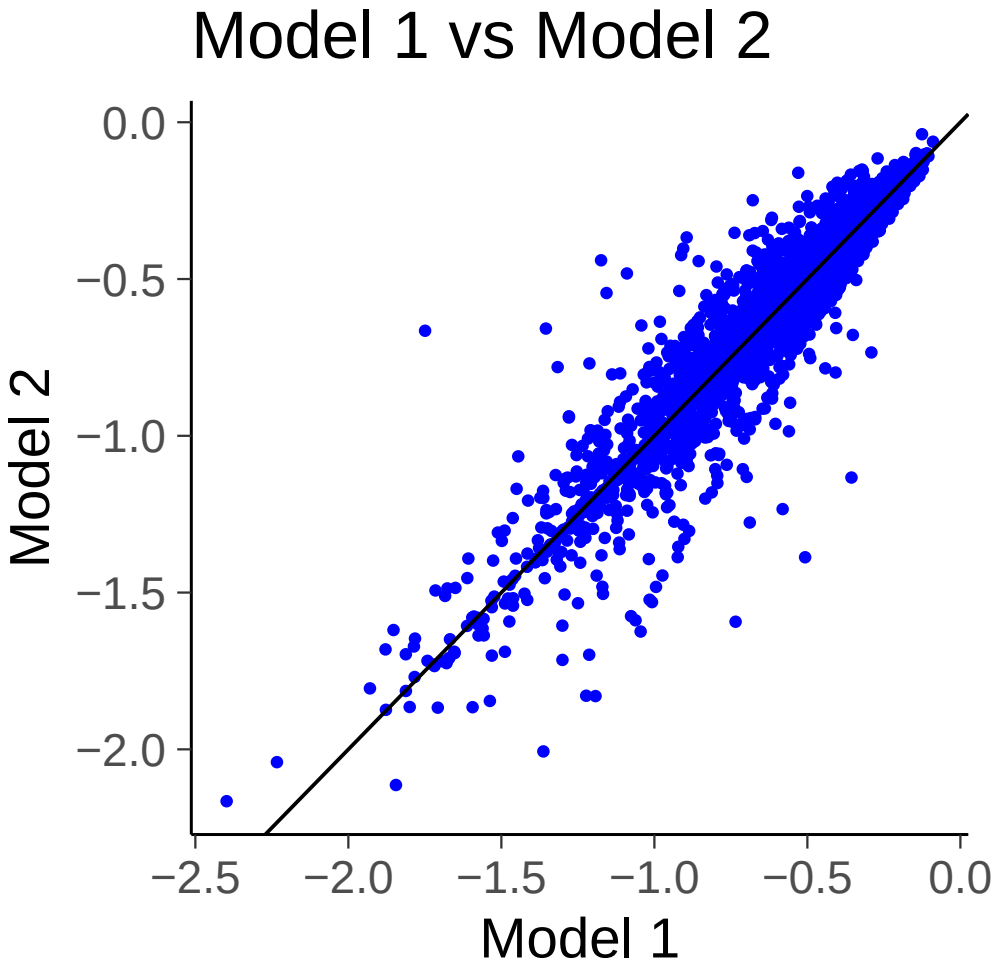
- BDA3, Chapter 6
- Gabry, Simpson, Vehtari, Betancourt, Gelman (2019). Visualization in Bayesian workflow. JRSS A, https://doi.org/10.1111/rssa.12378
- mc-stan.org/bayesplot/articles/graphical-ppcs.html

Arsenic well example – Model comparison

- Logistic regression for predicting probability of switching well with high arsenic level in rural Bangladesh
  - Model 1:
    log(arsenic) + distance
  - Model 2:
    log(arsenic) + distance + education level

Gelman, Hill & Vehtari (2020): Regression and Other Stories, Chapter 13.

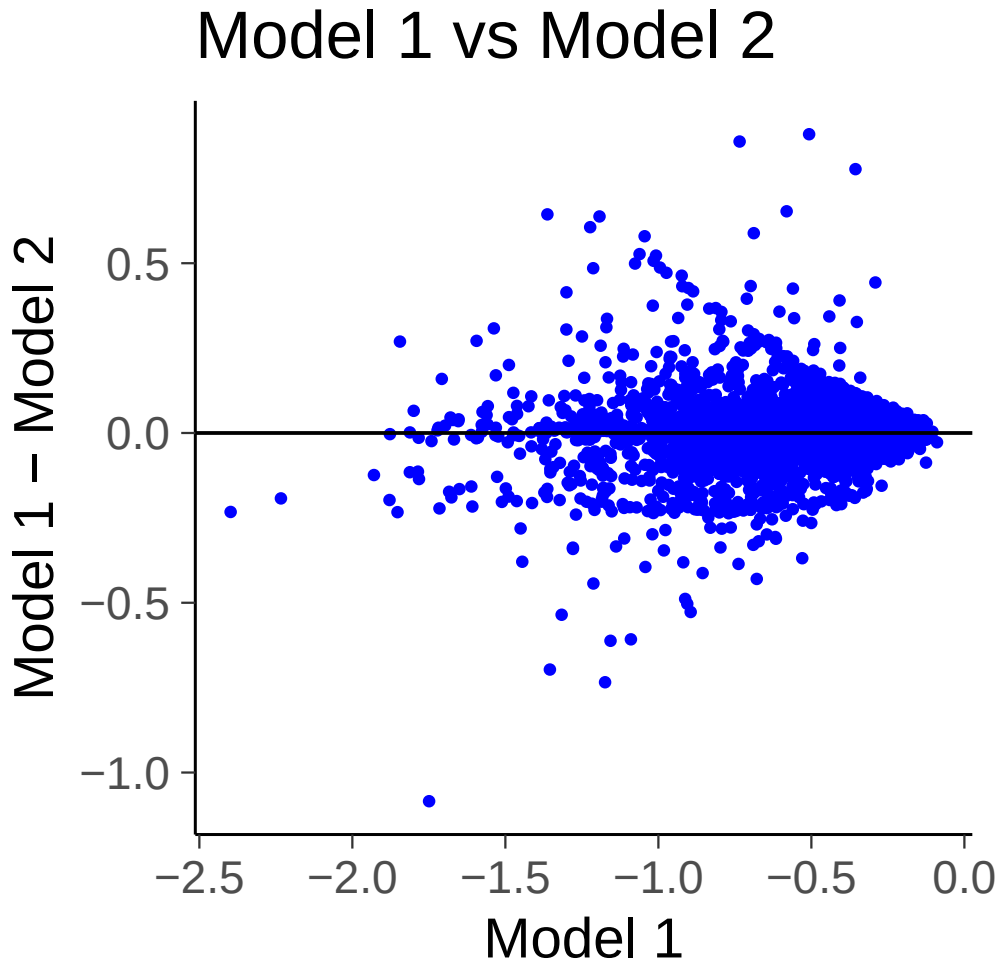
Arsenic well example – Model comparison



Model 1: $\widehat{\text{elpd}}_{\text{LOO}}\left(\text{M}_a \mid y^{\text{obs}}\right) \approx$ -1952, SE=16

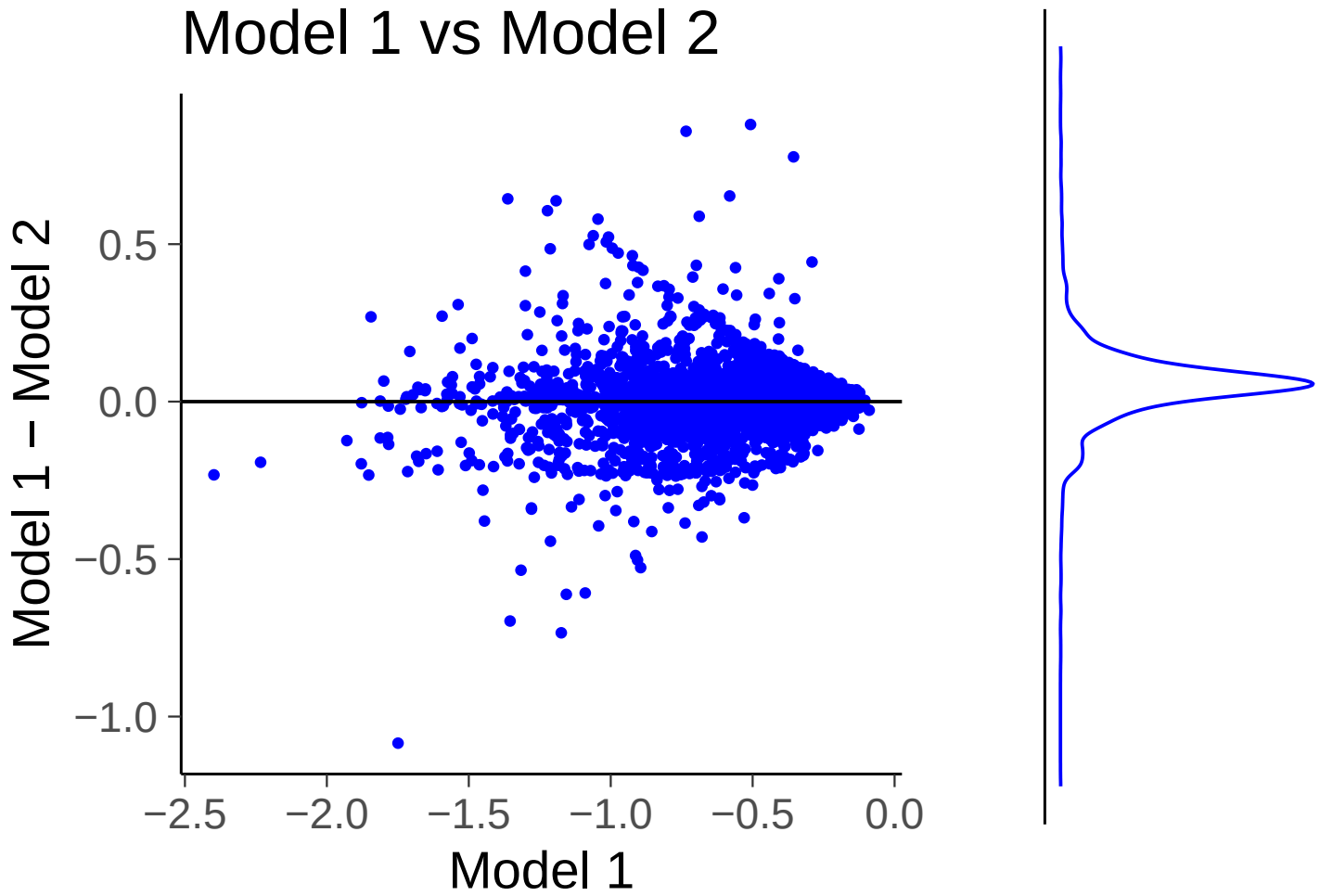
Model 2: $\widehat{\text{elpd}}_{\text{LOO}}\left(\text{M}_b \mid y^{\text{obs}}\right) \approx$ -1938, SE=17

Arsenic well example – Model comparison



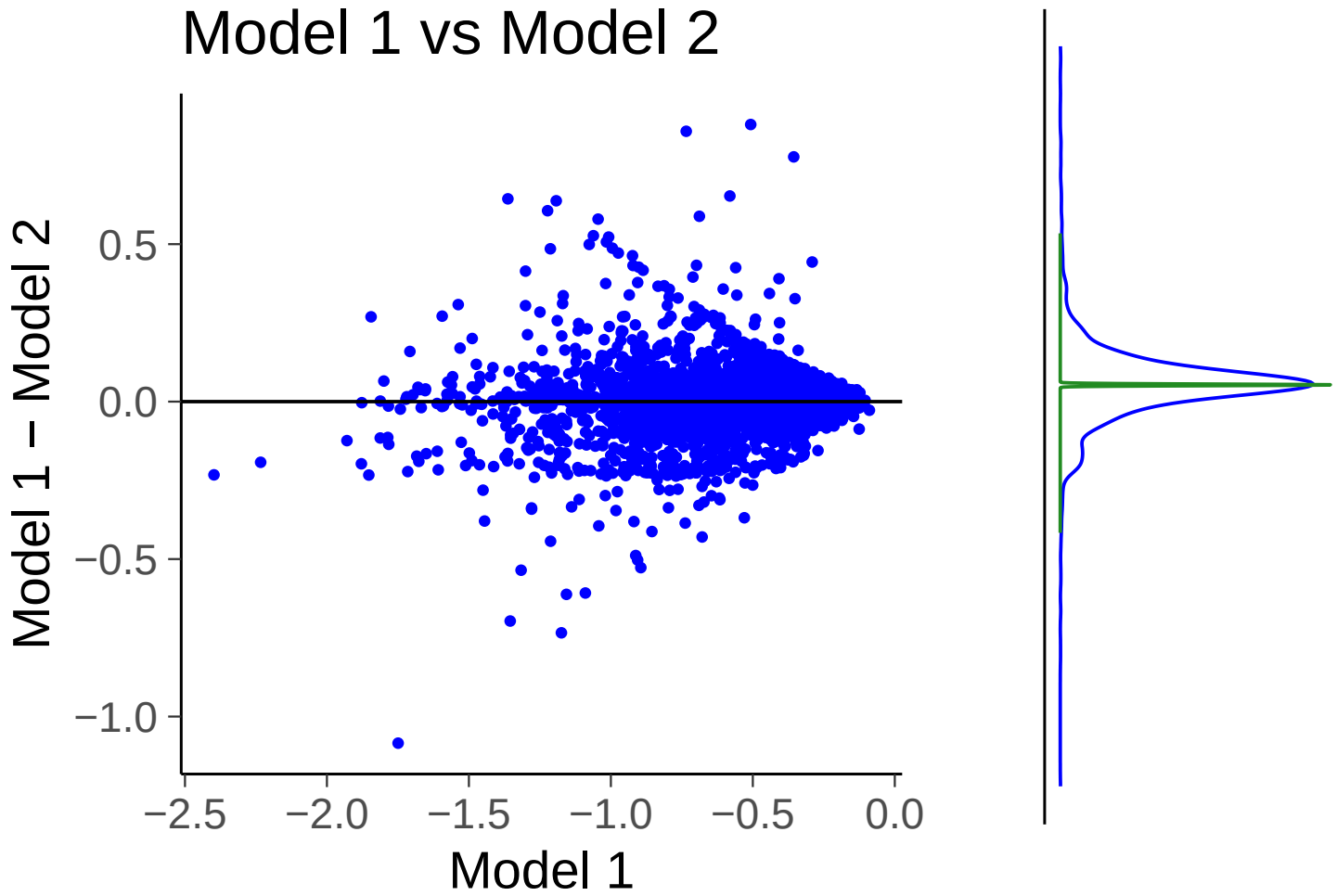
Difference: $\widehat{\mathrm{elpd}}_{\mathrm{LOO}}\left(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}}\right) \approx$ -14.4, SE = 6.1

Arsenic well example – Model comparison



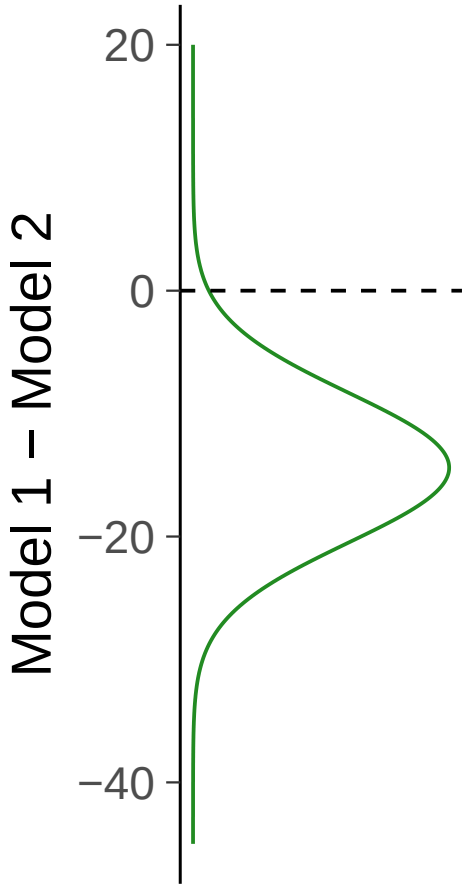
Difference: $\widehat{\mathrm{elpd}}_{\mathrm{LOO}}\left(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}}\right) \approx$ -14.4, SE = 6.1

Arsenic well example – Model comparison



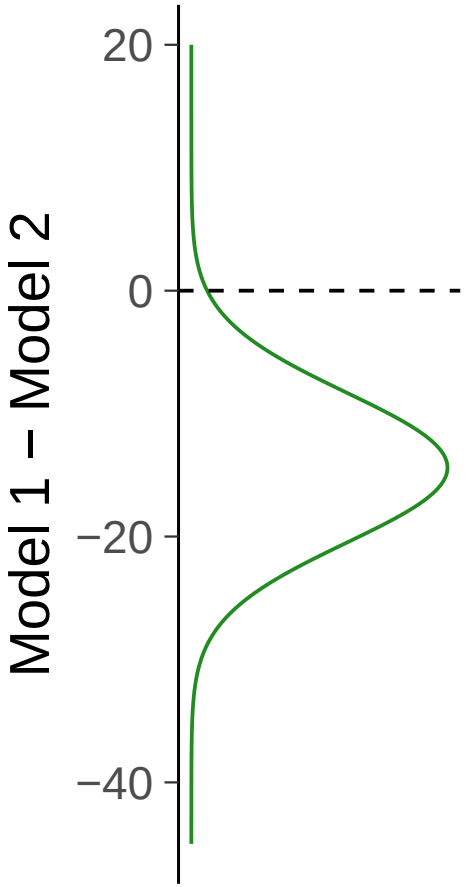
Difference: $\widehat{\mathrm{elpd}}_{\mathrm{LOO}}\left(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}}\right) \approx$ -14.4, SE = 6.1

Arsenic well example – Model comparison



Difference: $\widehat{\mathrm{elpd}}_{\mathrm{LOO}}\left(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}}\right) \approx$ -14.4, SE = 6.1

Arsenic well example – Model comparison




```
> loo_compare(model1, model2)
       elpd_diff se_diff
model2    0.0       0.0
model1  −14.4       6.1
```

see Vehtari, Gelman & Gabry (2017a)

8 schools – Model comparison

```
> loo_compare(pooled, hierarchical)
             elpd_diff se_diff
pooled          0.0       0.0
hierarchical   -0.3       0.7
```

No difference between pooled and hierarchical for predicting the future observations for a new school (exchangeble with the schools in the data).

Poisson vs Hurdle-Poisson example

	elpd_diff	se_diff
Hurdle−Poisson	0.0	0.0
Poisson	−215.9	22.1

Clear difference (which was also obvious in posterior predictive checks)

LOO difference uncertainty estimate reliability

1. The models make very similar predictions

2. The models are misspecified with outliers in the data

3. The number of observations is small

LOO difference uncertainty estimate reliability

1. The models make very similar predictions
   - if $|\widehat{\text{elpd}}_{\text{LOO}}(\text{M}_a, \text{M}_b \mid y^{\text{obs}})| < 4$, SE is not reliable, but the difference is small anyway
   - selecting a “wrong” model has small cost
   - in nested case the skewness favors the simpler model
2. The models are misspecified with outliers in the data
3. The number of observations is small

LOO difference uncertainty estimate reliability

1. The models make very similar predictions
   - if $|\widehat{\mathrm{elpd}}_{\mathrm{LOO}}(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}})| < 4$, SE is not reliable, but the difference is small anyway
   - selecting a “wrong” model has small cost
   - in nested case the skewness favors the simpler model
2. The models are misspecified with outliers in the data
   - in nested case the bias favors the simpler model
   - model checking and model extension to avoid misspecified models (Bayesian workflow)
3. The number of observations is small

LOO difference uncertainty estimate reliability

1. The models make very similar predictions
   - if $|\widehat{\mathrm{elpd}}_{\mathrm{LOO}}(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}})| < 4$, SE is not reliable, but the difference is small anyway
   - selecting a "wrong" model has small cost
   - in nested case the skewness favors the simpler model
2. The models are misspecified with outliers in the data
   - in nested case the bias favors the simpler model
   - model checking and model extension to avoid misspecified models (Bayesian workflow)
3. The number of observations is small
   - in nested case the skewness favors the simpler model
   - any inference with small n is difficult
   - if $|\widehat{\mathrm{elpd}}_{\mathrm{LOO}}(\mathrm{M}_a, \mathrm{M}_b \mid y^{\mathrm{obs}})| > 4$, model is well specified, and $n > 100$ then the normal approximation is good

Sometimes cross-validation is not needed

- In nested case, often easier and more accurate to analyse posterior distribution of more complex model directly avehtari.github.io/modelselection/betablockers.html
  - instead of comparing
    Model 1: $y \sim \text{normal}(\alpha, \sigma)$
    vs
    Model 2: $y \sim \text{normal}(\alpha + \beta x, \sigma)$
    look at the posterior of β directly

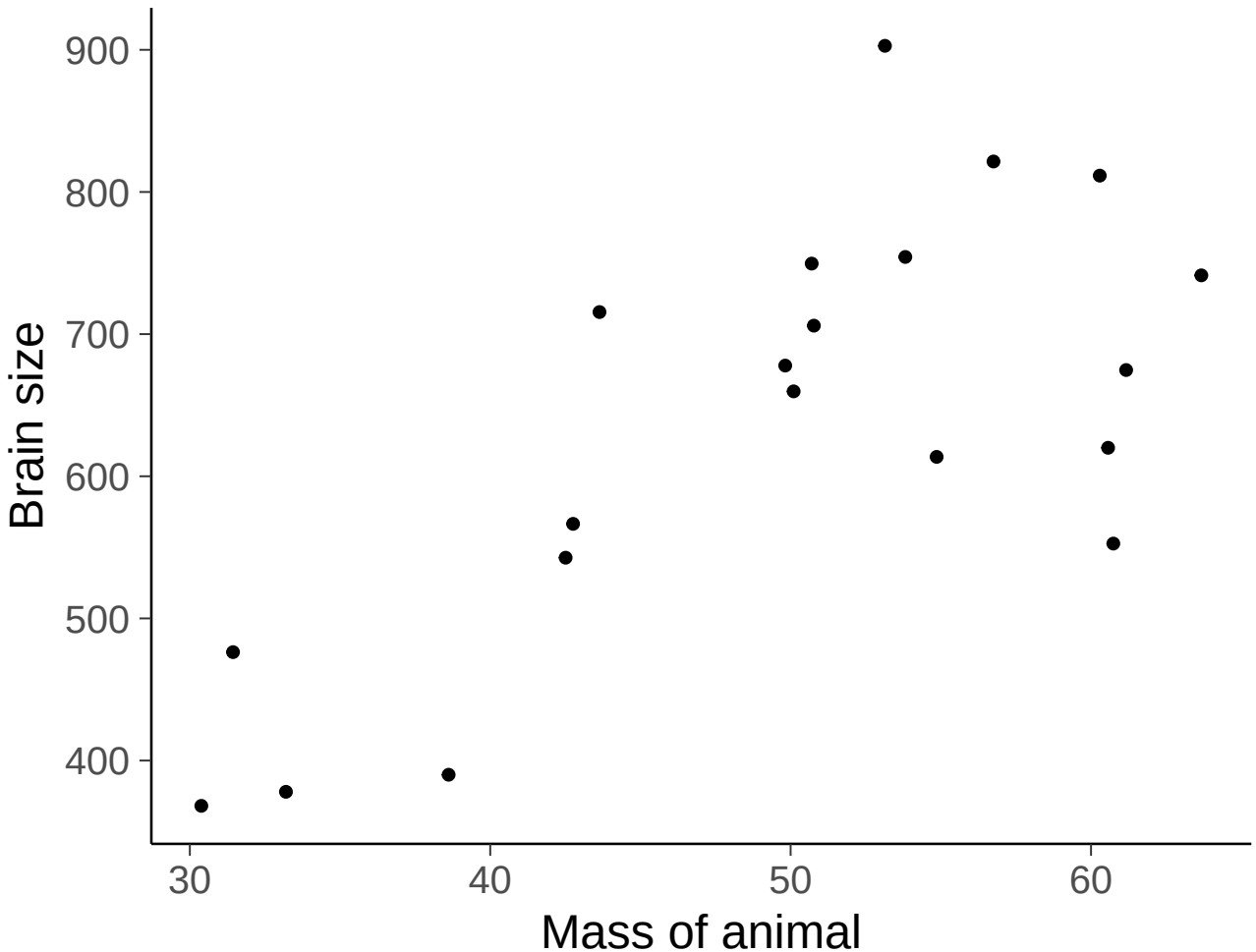
Model selection needed to avoid overfitting?

- Classic example is polynomial model with increasing number of components
  - overfits also with Bayesian inference and weak priors

Model selection needed to avoid overfitting?

- Classic example is polynomial model with increasing number of components
  - overfits also with Bayesian inference and weak priors

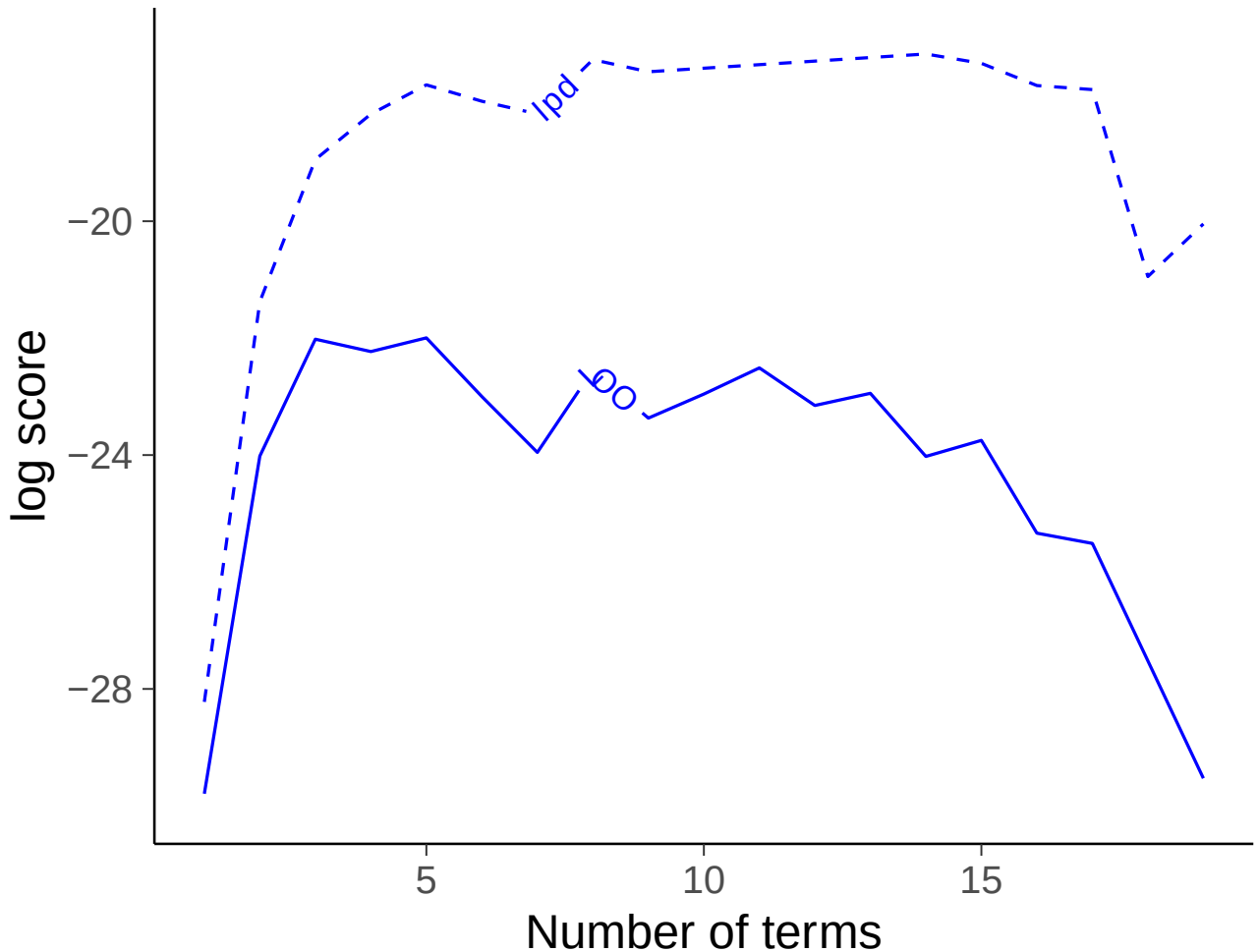
Simulated data by Richard McElreath



Model selection needed to avoid overfitting?

- Classic example is polynomial model with increasing number of components
  - overfits also with Bayesian inference and weak priors

Polynomial basis functions



Model selection needed to avoid overfitting?

- Gaussian process can be used as a prior on function space
  - GP can be approximated with basis functions

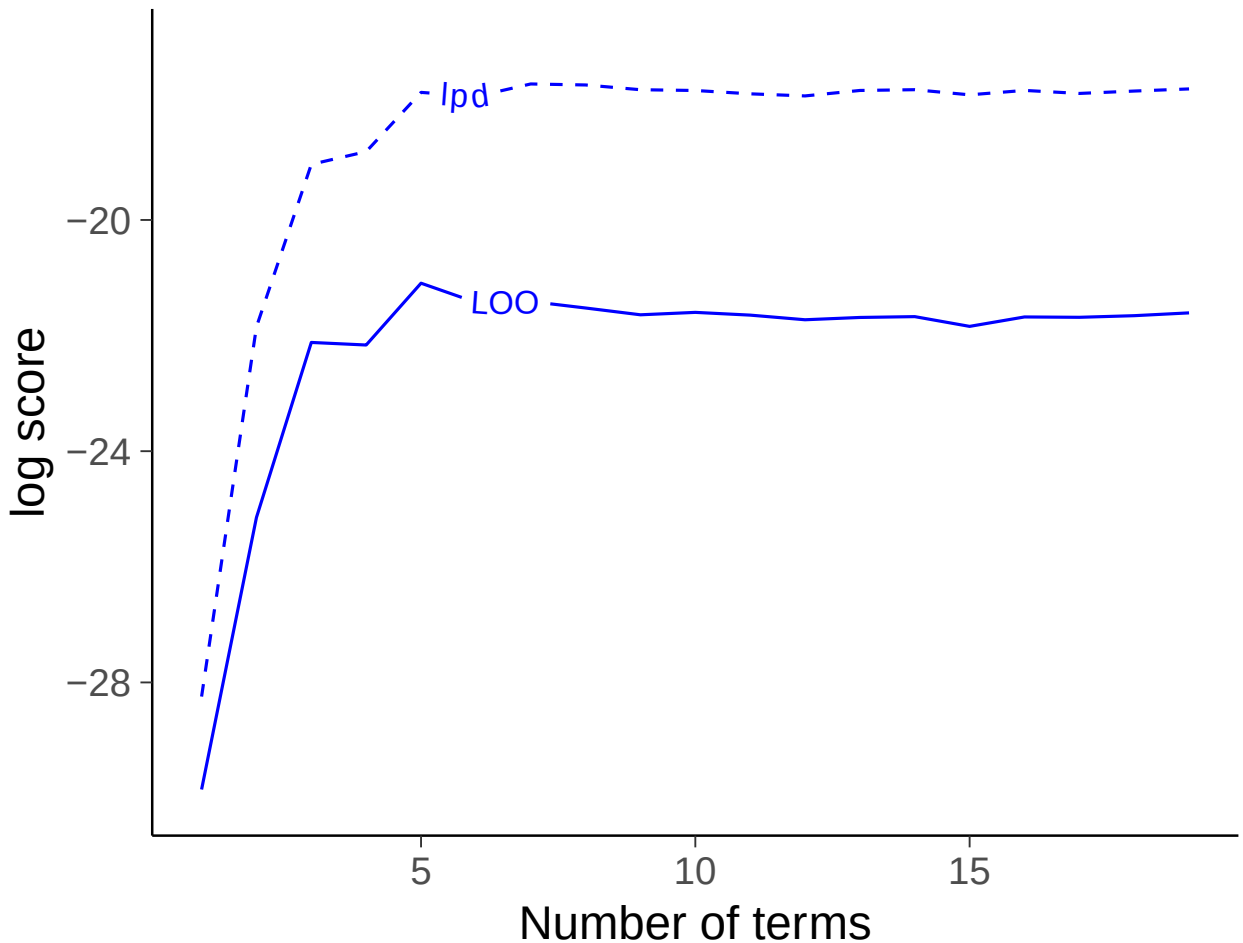
Model selection needed to avoid overfitting?

- Gaussian process can be used as a prior on function space
  - GP can be approximated with basis functions
  - more basis functions makes the approximation more accurate, but doesn't inflate the prior on function space

Model is not needed to avoid overfitting

- Gaussian process can be used as a prior on function space
  - GP can be approximated with basis functions
  - more basis functions makes the approximation more accurate, but doesn't inflate the prior on function space

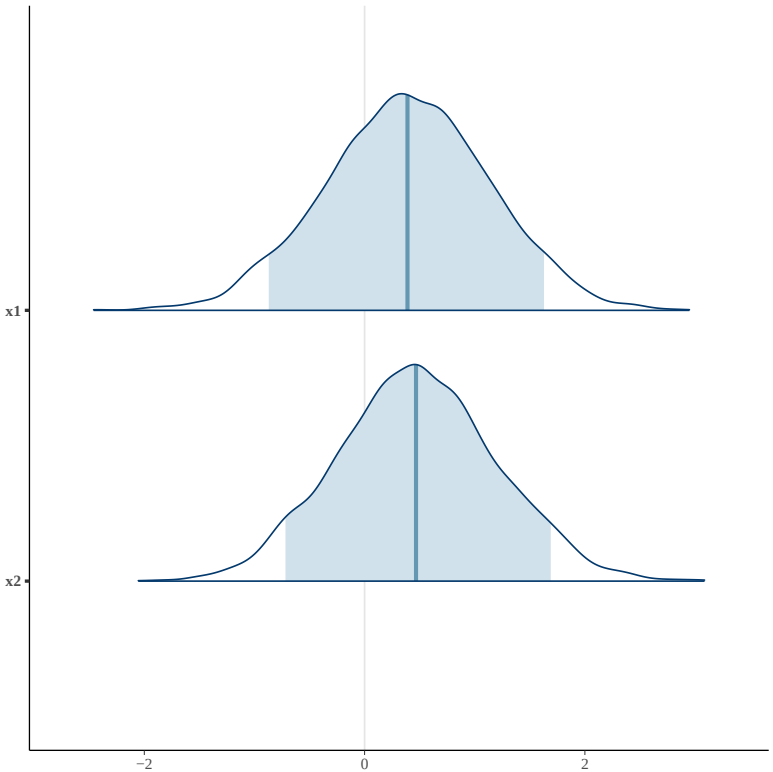
Gaussian process basis functions



Model is not needed to avoid overfitting

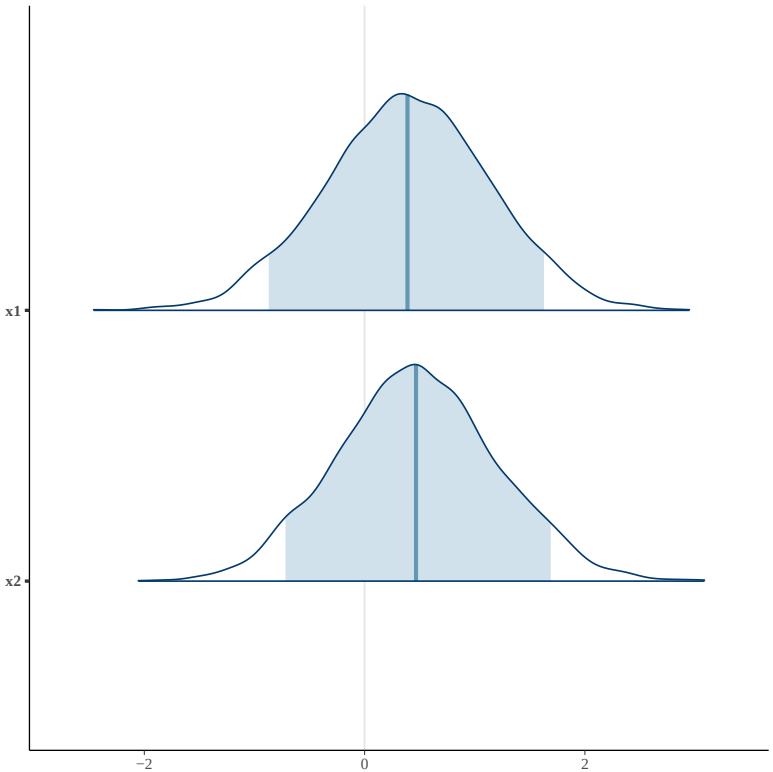
- No overfiitng when using good priors that keep the prior on the predictive space approximately constant when more components are added, e.g.
  - Gaussian procesees
  - (regularized) Horseshoe for sparsity
  - R2-D2 and R2-D2-M2 for prior on R^2

Sometimes predictive model comparison can be useful

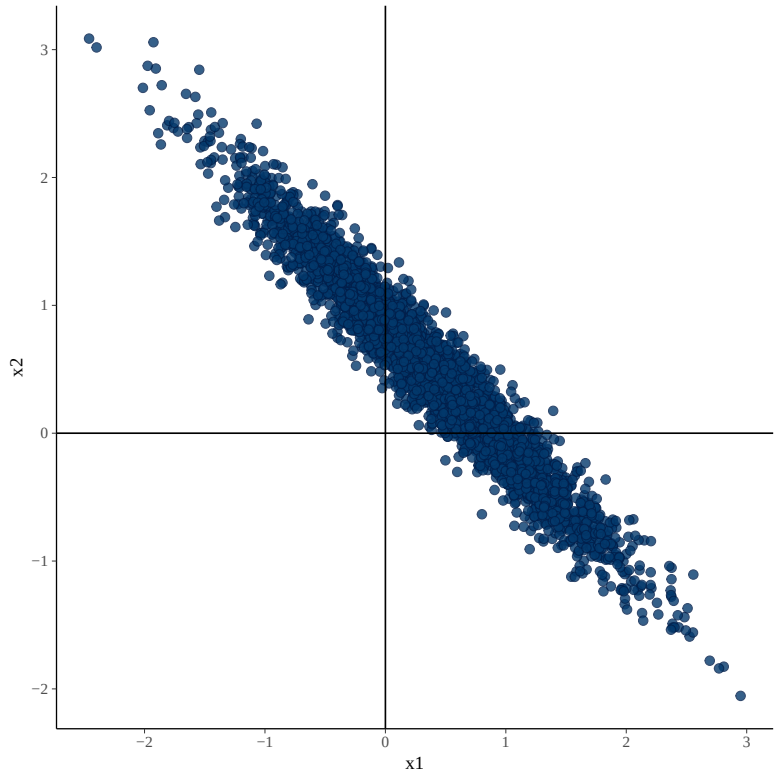


Marginal posterior intervals

Sometimes predictive model comparison can be useful



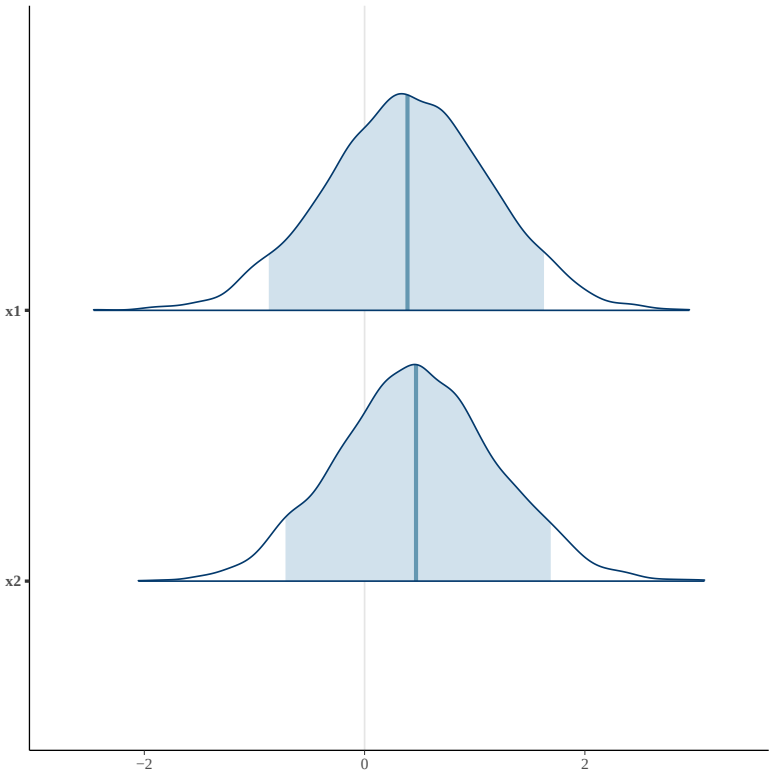
Marginal posterior intervals



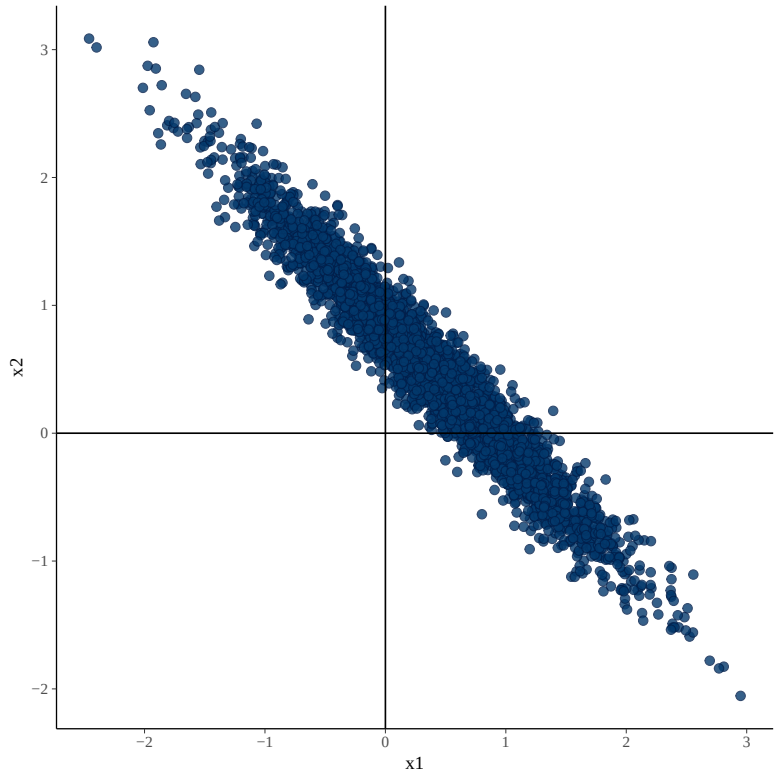
Joint posterior density

rstanarm + bayesplot

Sometimes predictive model comparison can be useful



Marginal posterior intervals



Joint posterior density

rstanarm + bayesplot

see also Collinear demo

What if one is not clearly better than others?

What if one is not clearly better than others?

- Continuous expansion including all models?
  - and then analyse the posterior distribution directly
    avehtari.github.io/modelselection/betablockers.html
  - sparse priors like regularized horseshoe prior instead of variable selection
    video, refs and demos at avehtari.github.io/modelselection/

What if one is not clearly better than others?

- Continuous expansion including all models?
  - and then analyse the posterior distribution directly
    avehtari.github.io/modelselection/betablockers.html
  - sparse priors like regularized horseshoe prior instead of variable selection
    video, refs and demos at avehtari.github.io/modelselection/
- Model averaging with BMA or Bayesian stacking?
  mc-stan.org/loo/articles/loo2-example.html

What if one is not clearly better than others?

- Continuous expansion including all models?
  - and then analyse the posterior distribution directly
    avehtari.github.io/modelselection/betablockers.html
  - sparse priors like regularized horseshoe prior instead of variable selection
    video, refs and demos at avehtari.github.io/modelselection/
- Model averaging with BMA or Bayesian stacking?
  mc-stan.org/loo/articles/loo2-example.html
- In a nested case choose simpler if assuming some cost for extra parts?
  andrewgelman.com/2018/07/26/
  parsimonious-principle-vs-integration-uncertainties/

What if one is not clearly better than others?

- Continuous expansion including all models?
  - and then analyse the posterior distribution directly
    avehtari.github.io/modelselection/betablockers.html
  - sparse priors like regularized horseshoe prior instead of variable selection
    video, refs and demos at avehtari.github.io/modelselection/
- Model averaging with BMA or Bayesian stacking?
  mc-stan.org/loo/articles/loo2-example.html
- In a nested case choose simpler if assuming some cost for extra parts?
  andrewgelman.com/2018/07/26/
  parsimonious-principle-vs-integration-uncertainties/
- In a nested case choose more complex if you want to take into account all the uncertainties.
  andrewgelman.com/2018/07/26/
  parsimonious-principle-vs-integration-uncertainties/

Model averaging

- Prefer continuous model expansion

Model averaging

- Prefer continuous model expansion
- If needed integrate over the model space = model averaging

Model averaging

- Prefer continuous model expansion
- If needed integrate over the model space = model averaging
- Bayesian stacking may work better than BMA in case of misspecified models or small data
  - Yao, Vehtari, Simpson, & Gelman (2018)

Cross-validation and model selection

- Cross-validation can be used for model selection if
  - small number of models
  - the difference between models is clear

Cross-validation and model selection

- Cross-validation can be used for model selection if
  - small number of models
  - the difference between models is clear
- Be careful if using cross-validation to choose from a large set of models
  - selection process can lead to severe overfitting

Cross-validation and model selection

- Cross-validation can be used for model selection if
  - small number of models
  - the difference between models is clear
- Be careful if using cross-validation to choose from a large set of models
  - selection process can lead to severe overfitting
- Overfitting in selection process is not unique for cross-validation

Selection induced bias and overfitting

- Selection induced bias in cross-validation
  - same data is used to assess the performance and make the selection
  - the selected model fits more to the data
  - the CV estimate for the selected model is biased
  - recognized already, e.g., by Stone (1974)

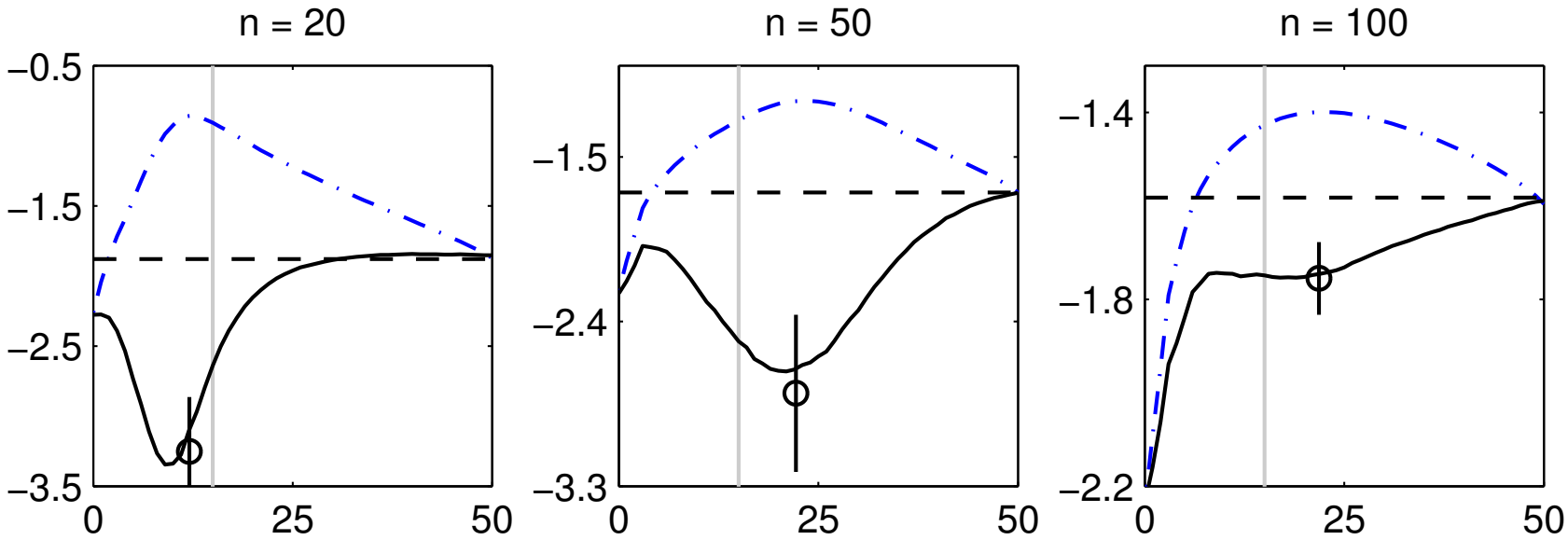
Selection induced bias and overfitting

- Selection induced bias in cross-validation
  - same data is used to assess the performance and make the selection
  - the selected model fits more to the data
  - the CV estimate for the selected model is biased
  - recognized already, e.g., by Stone (1974)
- Performance of the selection process itself can be assessed using two level cross-validation, but it does not help choosing better models

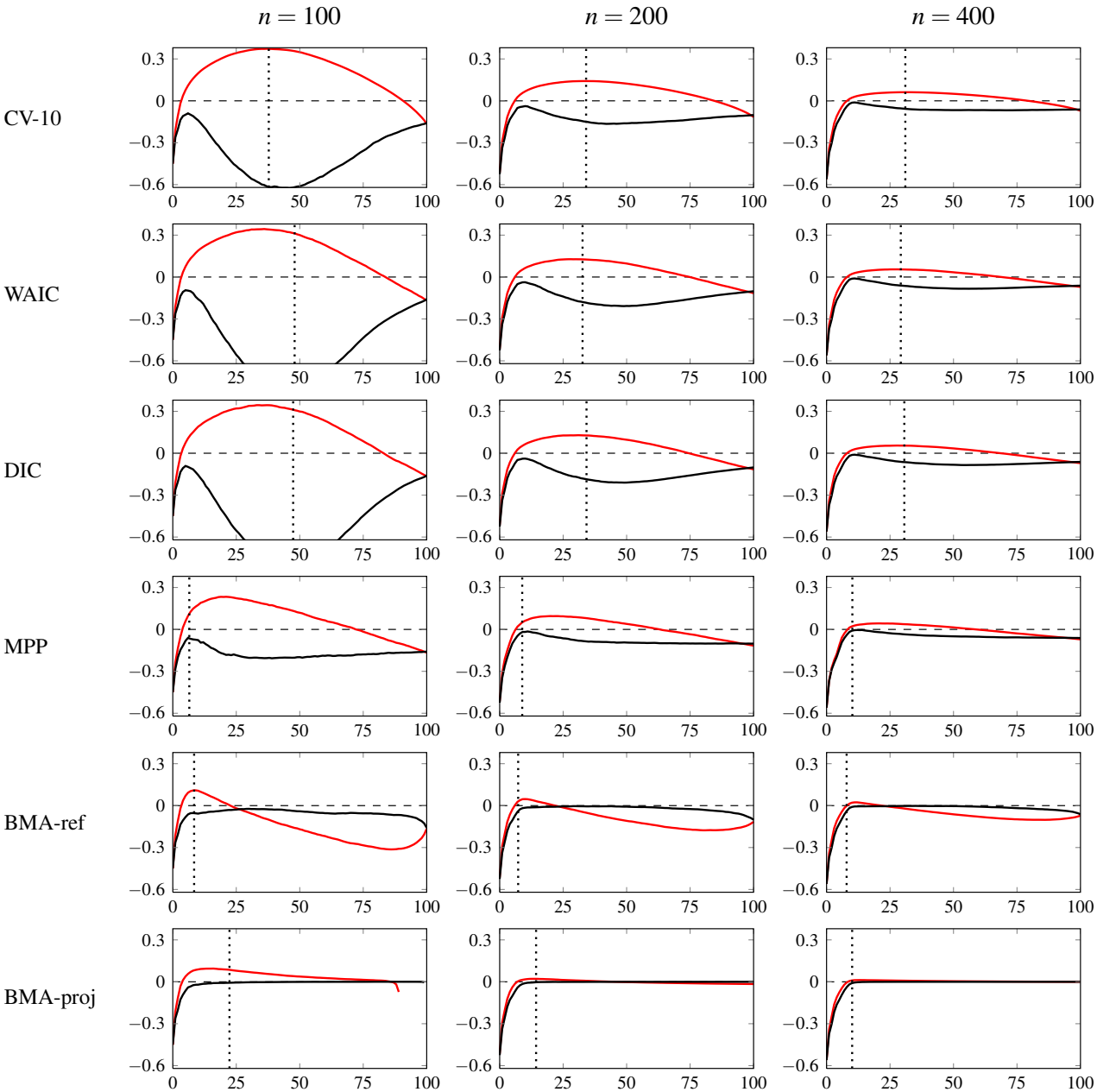
Selection induced bias and overfitting

- Selection induced bias in cross-validation
  - same data is used to assess the performance and make the selection
  - the selected model fits more to the data
  - the CV estimate for the selected model is biased
  - recognized already, e.g., by Stone (1974)
- Performance of the selection process itself can be assessed using two level cross-validation, but it does not help choosing better models
- Bigger problem if there is a large number of models as in covariate selection

Selection induced bias in variable selection



Selection induced bias in variable selection



Piironen & Vehtari (2017)

Take-home messages

- It's good to think predictions of observables, because observables are the only ones we can observe
- Cross-validation can simulate predicting and observing new data
- Cross-validation is good if you don't trust your model
- Different variants of cross-validation are useful in different scenarios
- Cross-validation has high variance, and **if** you trust your model you can beat cross-validation in accuracy

Take-home messages

- It's good to think predictions of observables, because observables are the only ones we can observe
- Cross-validation can simulate predicting and observing new data
- Cross-validation is good if you don't trust your model
- Different variants of cross-validation are useful in different scenarios
- Cross-validation has high variance, and **if** you trust your model you can beat cross-validation in accuracy

Take-home messages

- It's good to think predictions of observables, because observables are the only ones we can observe
- Cross-validation can simulate predicting and observing new data
- Cross-validation is good if you don't trust your model
- Different variants of cross-validation are useful in different scenarios
- Cross-validation has high variance, and **if** you trust your model you can beat cross-validation in accuracy

Take-home messages

- It's good to think predictions of observables, because observables are the only ones we can observe
- Cross-validation can simulate predicting and observing new data
- Cross-validation is good if you don't trust your model
- Different variants of cross-validation are useful in different scenarios
- Cross-validation has high variance, and **if** you trust your model you can beat cross-validation in accuracy

Take-home messages

- It's good to think predictions of observables, because observables are the only ones we can observe
- Cross-validation can simulate predicting and observing new data
- Cross-validation is good if you don't trust your model
- Different variants of cross-validation are useful in different scenarios
- Cross-validation has high variance, and **if** you trust your model you can beat cross-validation in accuracy