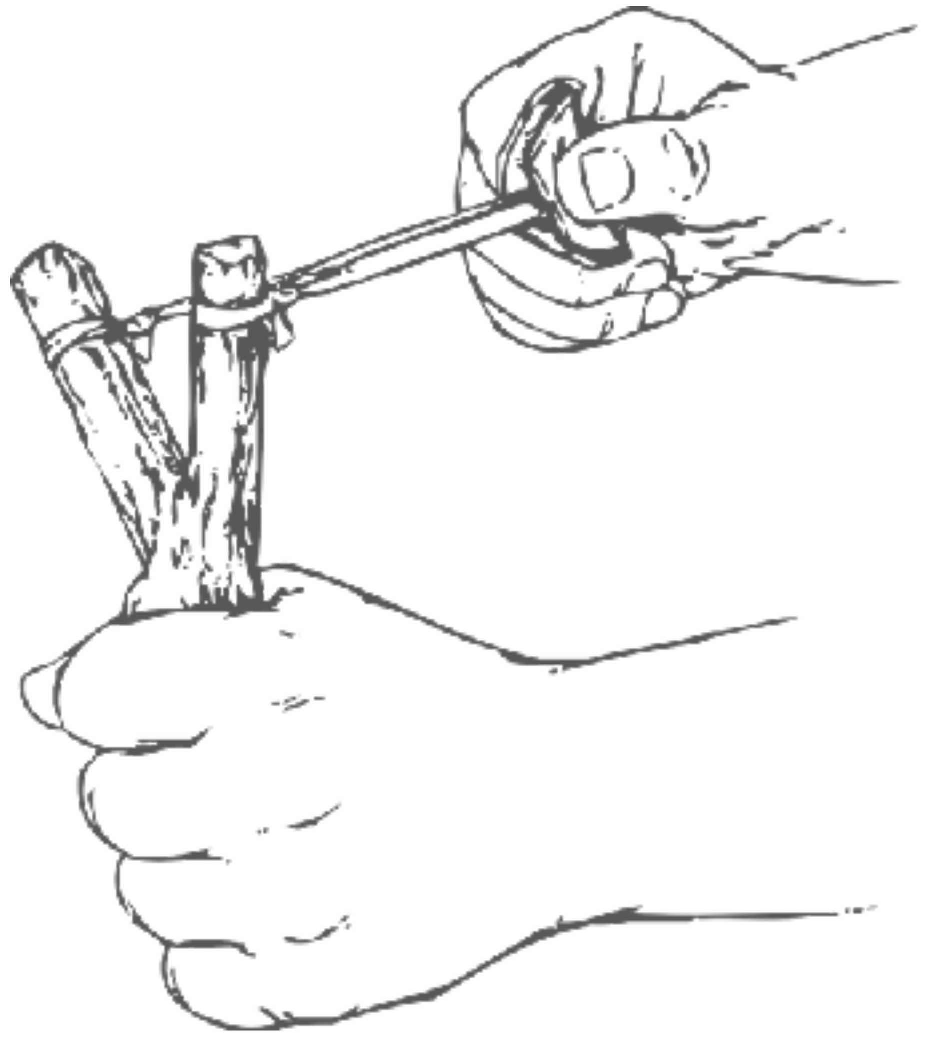
Bayesian data analysis: Theory & practice

Part 4a: Bayesian model comparison

Michael Franke

Main learning goals



1. understand the role of model comparison in statistical inquiry
2. understand & know how to apply common methods
    a. information criteria (AIC)
    b. Bayes factors
    c. cross-validation (LOO)
3. get familiar with methods to compute Bayes factors
    a. Savage-Dickey method
    b. importance & bridge sampling

what is
model comparison
(good for)?

Three pillars of BDA

1. parameter estimation / inference [which parameter values are credible given data and model?]

$$\underbrace{P(\theta \mid D)}_{\text{posterior}} \propto \underbrace{P(\theta)}_{\text{prior}} \times \underbrace{P(D \mid \theta)}_{\text{likelihood}}$$

2. predictions [which future data observations are likely given my model?]

a. prior

$$P(D_{\text{pred}}) = \int P(\theta)\ P(D_{\text{pred}} \mid \theta)\ \mathrm{d}\theta$$

b. posterior

$$P(D_{\text{pred}} \mid D_{\text{obs}}) = \int P(\theta \mid D_{\text{obs}})\ P(D_{\text{pred}} \mid \theta)\ \mathrm{d}\theta$$

3. model comparison [which model of two models is more likely to have generated the data?]

$$\underbrace{\frac{P(M_1 \mid D)}{P(M_2 \mid D)}}_{\text{posterior odds}} = \underbrace{\frac{P(D \mid M_1)}{P(D \mid M_2)}}_{\text{Bayes factor}} \underbrace{\frac{P(M_1)}{P(M_2)}}_{\text{prior odds}}$$

What makes a model ‘good’?

Good explanation

- model M is a good model of data D to the extent that it **explains** D well
- a good explanation of D is a view of the world that makes D less puzzling
  - **the higher $P(D \mid M)$, the better M explains D**

Simplicity / economy / parsimony

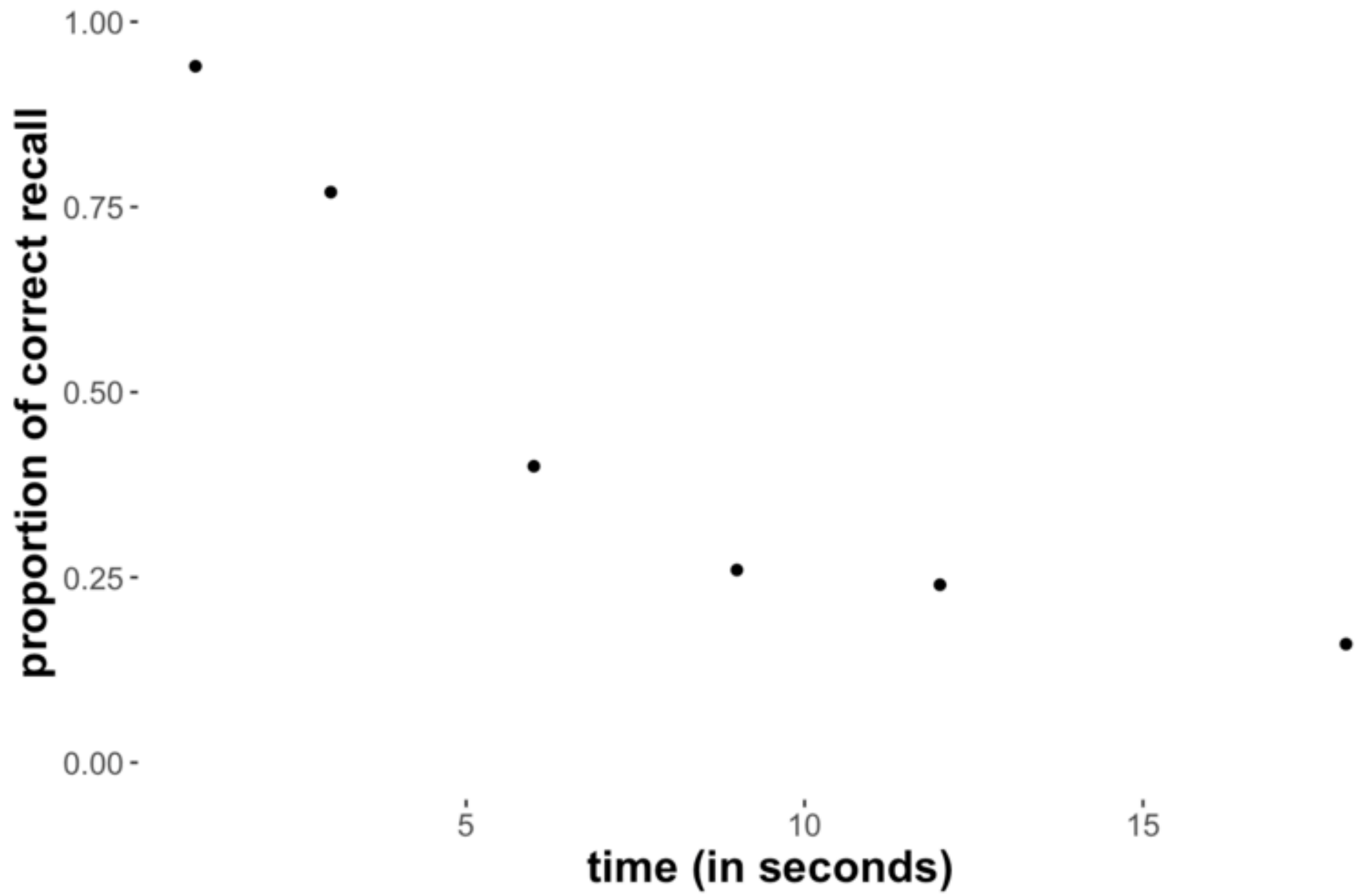
- model M is a good model of data D to the extent that it is **simple**
- we want our explanations to be austere, with few postulates, no magic ingredients and a lean mechanism / functional form
  - **the fewer (powerful) parameters M has, the better**

an information criterion

Forgetting data

- 100 binary measurements (correct / incorrect recall) at different times after memorization

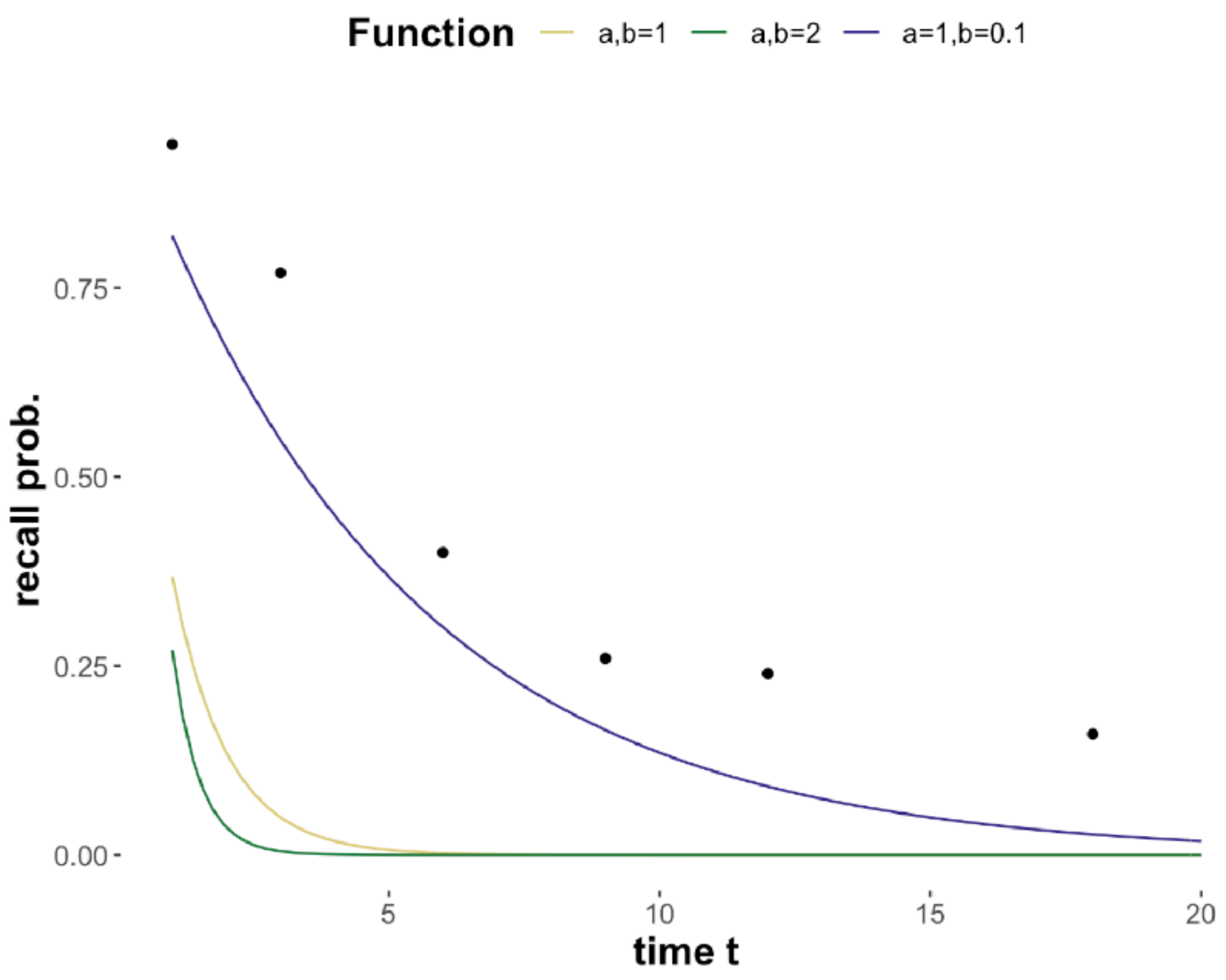
```
# time after memorization (in seconds)
t = c(1, 3, 6, 9, 12, 18)
# proportion (out of 100) of correct recall
y = c(.94, .77, .40, .26, .24, .16)
# number of observed correct recalls (out of 100)
obs = y * 100
```



data from Myung (2003, Tutorial on Maximum Likelihood Estimation)

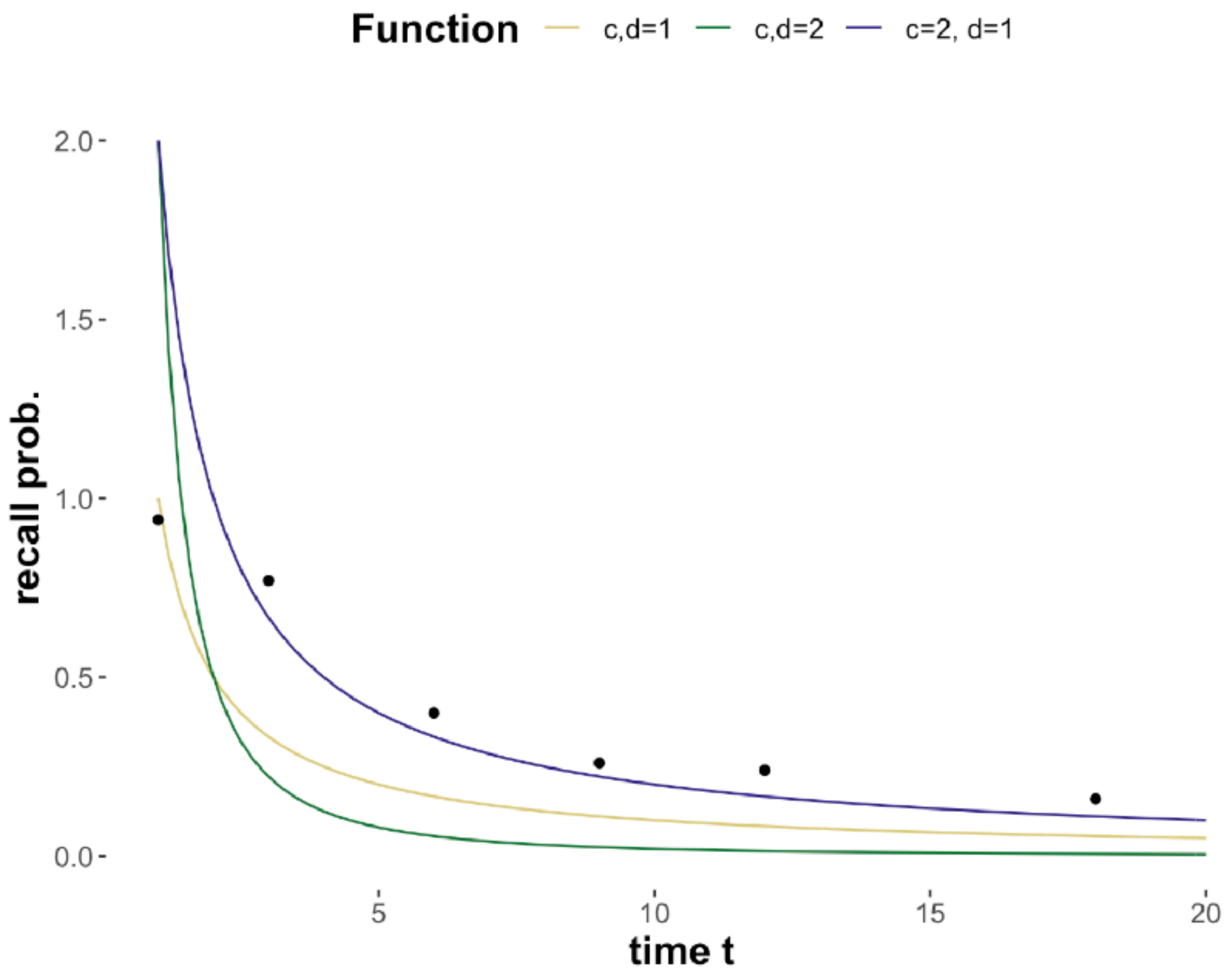
Exponential model

$P(D = \langle k, N \rangle \mid \langle a, b \rangle) = \text{Binom}(k, N, a \exp(-bt))$
with $a, b > 0$



Power model

$P(D = \langle k, N \rangle \mid \langle c, d \rangle) = \text{Binom}(k, N, c\ t^{-d})$
with $c, d > 0$



Akaike information criterion

- M_i is a (frequentist) model with likelihood function $P(D \mid \theta_i, M_i)$
- k free parameters in parameter vector θ_i
- $\hat{\theta}_i = \arg\max_{\theta_i} P(D_{\text{obs}} \mid \theta_i, M_i)$ is the MLE for observed data D_{obs}
- the AIC-score (where lower is better) is defined as:

$$\mathrm{AIC}(M_i, D_{\text{obs}}) = \underbrace{2k}_{\text{[penalty for complexity]}} - 2 \underbrace{\log P(D_{\text{obs}} \mid \hat{\theta}_i, M_i)}_{\text{[how surprising is the data for the best parameter of the model?]}}$$

Computing AIC scores

step 1: compute MLE

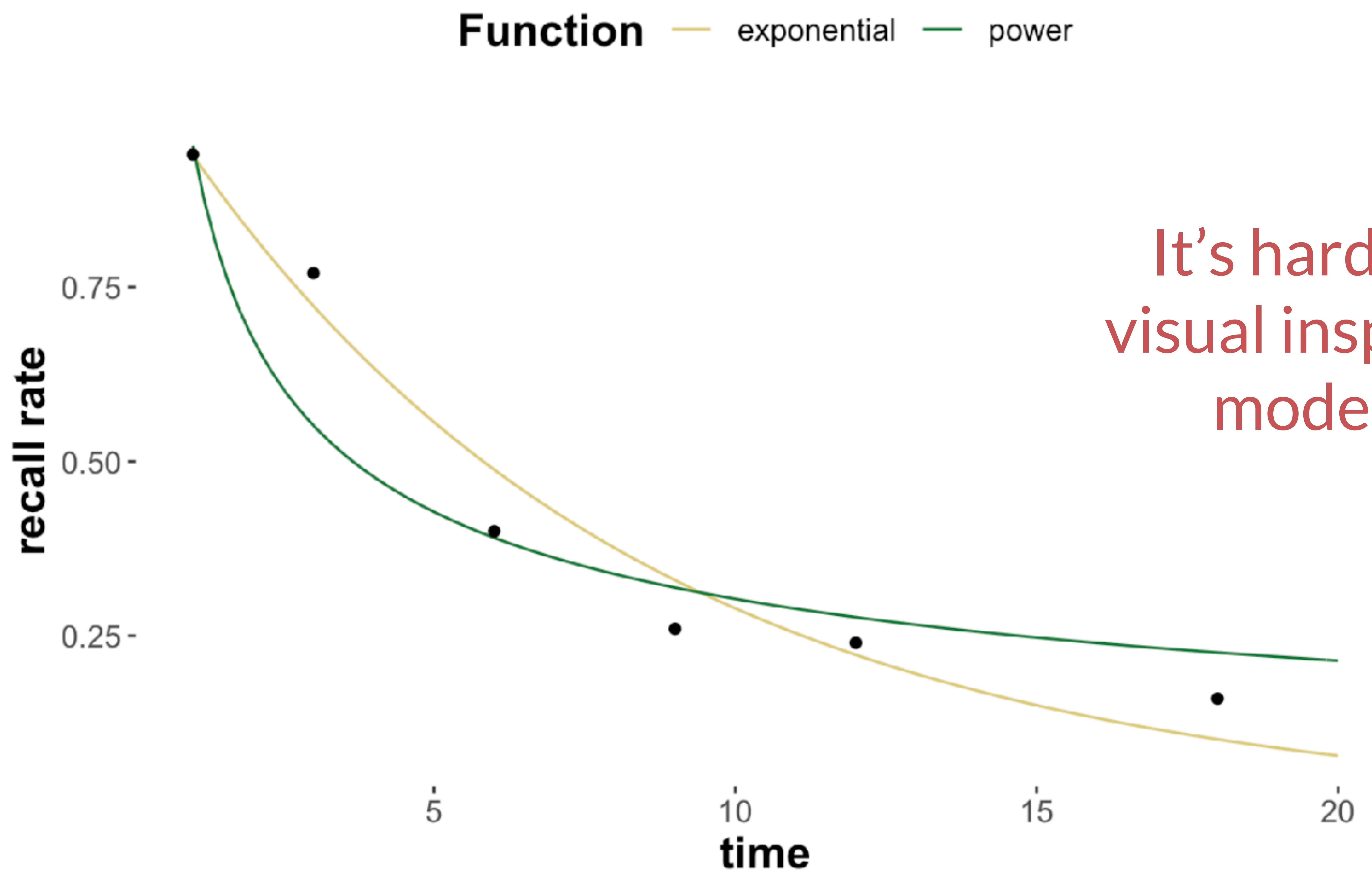
```r
# generic neg-log-LH function (covers both models)
nLL_generic <- function(par, model_name) {
  w1 <- par[1]
  w2 <- par[2]
  # make sure paramters are in acceptable range
  if (w1 < 0 | w2 < 0 | w1 > 20 | w2 > 20) {
    return(NA)
  }
  # calculate predicted recall rates for given parameters
  if (model_name == "exponential") {
    theta <- w1*exp(-w2*t)  # exponential model
  } else {
    theta <- w1*t^(-w2)     # power model
  }
  # avoid edge cases of infinite log-likelihood
  theta[theta <= 0.0] <- 1.0e-4
  theta[theta >= 1.0] <- 1-1.0e-4
  # return negative log-likelihood of data
  - sum(dbinom(x = obs, prob = theta, size = 100, log = T))
}
# negative log likelihood of exponential model
nLL_exp <- function(par) {nLL_generic(par, "exponential")}
# negative log likelihood of power model
nLL_pow <- function(par) {nLL_generic(par, "power")}
```

```r
# getting the best fitting values
bestExpo <- optim(nLL_exp, par = c(1,0.5))
bestPow  <- optim(nLL_pow, par = c(0.5,0.2))
MLEstimates = data.frame(model = rep(c("exponential", "power"), each = 2),
                         parameter = c("a", "b", "c", "d"),
                         value = c(bestExpo$par, bestPow$par))
knitr::kable(MLEstimates)
```

model	parameter	value
exponential	a	1.0701722
exponential	b	0.1308151
power	c	0.9531330
power	d	0.4979154

Inspecting each model's MLE predictions

step 1: compute MLE



It's hard to say from visual inspection which model is better.

Computing AIC scores

step 2: calculate AIC from MLE

```
get_AIC <- function(optim_fit) {
  2 * length(optim_fit$par) + 2 * optim_fit$value
}
AIC_scores <- tibble(
  AIC_exponential = get_AIC(bestExpo),
  AIC_power = get_AIC(bestPow)
)
AIC_scores
```

```
## # A tibble: 1 x 2
##   AIC_exponential AIC_power
##             <dbl>     <dbl>
## 1            41.3      57.5
```

$$\text{AIC}(M_i, D_{\text{obs}}) = 2k - 2\log P(D_{\text{obs}} \mid \hat{\theta}_i, M_i)$$

Exponential model has lower AIC score, so it comes up as "better" under this approach.

Problems with AIC

extending also, with provisos, to other information criteria

- AIC is **not consistent**
  - not guaranteed to select the true data-generating model under incrementally increasing observations
- AIC has a **tendency towards overfitting**
  - selects more complex models over true simpler ones
- AIC has a **crude measure of model complexity**
  - just number of parameters, but not their functional role
  - e.g., do we really want to count *all* random-effect parameters as equal to fixed-effect parameters?

Vanderkerckhove et al. (2015, "Model Comparison and the Principle of Parsimony")

Bayes factors

Bayes factors

measure of belief change from observational evidence

- Bayesian models (with priors):
  - M_1 has prior $P(\theta_1 \mid M_1)$ and likelihood $P(D \mid \theta_1, M_1)$
  - M_2 has prior $P(\theta_2 \mid M_2)$ and likelihood $P(D \mid \theta_2, M_2)$
- Bayes factor is the factor by which the prior odds need to be adjusted by rational belief update after observing D to arrive at posterior odds

$$\underbrace{\frac{P(M_1 \mid D)}{P(M_2 \mid D)}}_{\text{posterior odds}} = \underbrace{\frac{P(D \mid M_1)}{P(D \mid M_2)}}_{\text{Bayes factor}} \underbrace{\frac{P(M_1)}{P(M_2)}}_{\text{prior odds}}$$

Bayes factors

unpacked: ratio of marginal likelihoods

$$\frac{P(D \mid M_1)}{P(D \mid M_2)} = \frac{\int P(\theta_1 \mid M_1)\ P(D \mid \theta_1, M_1)\ \mathrm{d}\theta_1}{\int P(\theta_2 \mid M_2)\ P(D \mid \theta_2, M_2)\ \mathrm{d}\theta_2}$$

- Bayes factors look at **ex ante (a priori) predictions**
- integration over priors → **implicit (severe) punishment for model complexity**
- calculating Bayes factors is **computationally hard** for sophisticated models

Bayes factors

notation & interpretation

$$BF_{12} = \frac{P(D \mid M_1)}{P(D \mid M_2)}$$

read as: "BF in favor of model 1 over model 2"

BF_{12}	**interpretation**
1	irrelevant data
1 - 3	hardly worth ink or breath
3 - 6	anecdotal
6 - 10	now we're talking: substantial
10 - 30	strong
30 - 100	very strong
100 +	decisive (bye, bye M_2!)

How to calculate Bayes factors

calculate marginal likelihood (for each model)

- grid approximation
- Monte Carlo sampling
- importance / bridge sampling

calculate Bayes factor (for a pair of models)

- for nested models:
  - Savage-Dickey method
  - encompassing priors
- transdimensional MCMC (not covered here)

computing marginal likelihoods

- grid approximation
- Monte Carlo sampling
- importance / bridge sampling

Bayesian forgetting models

exponential model

$$P(D = \langle k, N \rangle \mid \langle a, b \rangle, M_{\text{exp}}) = \text{Binom}(k, N, a \exp(-bt))$$
$$P(a \mid M_{\text{exp}}) = \text{Uniform}(a, 0, 1.5)$$
$$P(b \mid M_{\text{exp}}) = \text{Uniform}(b, 0, 1.5)$$

power model

$$P(D = \langle k, N \rangle \mid \langle c, d \rangle, M_{\text{pow}}) = \text{Binom}(k, N, c\, t^{-d})$$
$$P(d \mid M_{\text{pow}}) = \text{Uniform}(c, 0, 1.5)$$
$$P(c \mid M_{\text{pow}}) = \text{Uniform}(d, 0, 1.5)$$

```r
# prior exponential model
priorExp = function(a, b){
  dunif(a, 0, 1.5) * dunif(b, 0, 1.5)
}
# likelihood function exponential model
lhExp = function(a, b){
  theta = a*exp(-b*t)
  theta[theta <= 0.0] = 1.0e-5
  theta[theta >= 1.0] = 1-1.0e-5
  prod(dbinom(x = obs, prob = theta, size = 100))
}


# prior power model
priorPow = function(c, d){
  dunif(c, 0, 1.5) * dunif(d, 0, 1.5)
}
# likelihood function power model
lhPow = function(c, d){
  theta = c*t^(-d)
  theta[theta <= 0.0] = 1.0e-5
  theta[theta >= 1.0] = 1-1.0e-5
  prod(dbinom(x = obs, prob = theta, size = 100))
}
```

Bayes factors from grid approximation

```r
# make sure the functions accept vector input
lhExp = Vectorize(lhExp)
lhPow = Vectorize(lhPow)

# define the step size of the grid
stepsize = 0.01
# calculate the "evidence" aka marginal likelihood
evidence = expand.grid(x = seq(0.005, 1.495, by = stepsize),
                       y = seq(0.005, 1.495, by = stepsize)) %>%
  mutate(lhExp = lhExp(x,y), priExp = 1 / length(x),  # uniform priors!
         lhPow = lhPow(x,y), priPow = 1 / length(x))

paste0("BF in favor of exponential model: ",
          with(evidence, sum(priExp*lhExp)/ sum(priPow*lhPow)) %>% round(2))
```

```
## [1] "BF in favor of exponential model: 1221.39"
```

Reminder: AIC scores

```
## # A tibble: 1 x 2
##   AIC_exponential AIC_power
##             <dbl>     <dbl>
## 1            41.3      57.5
```

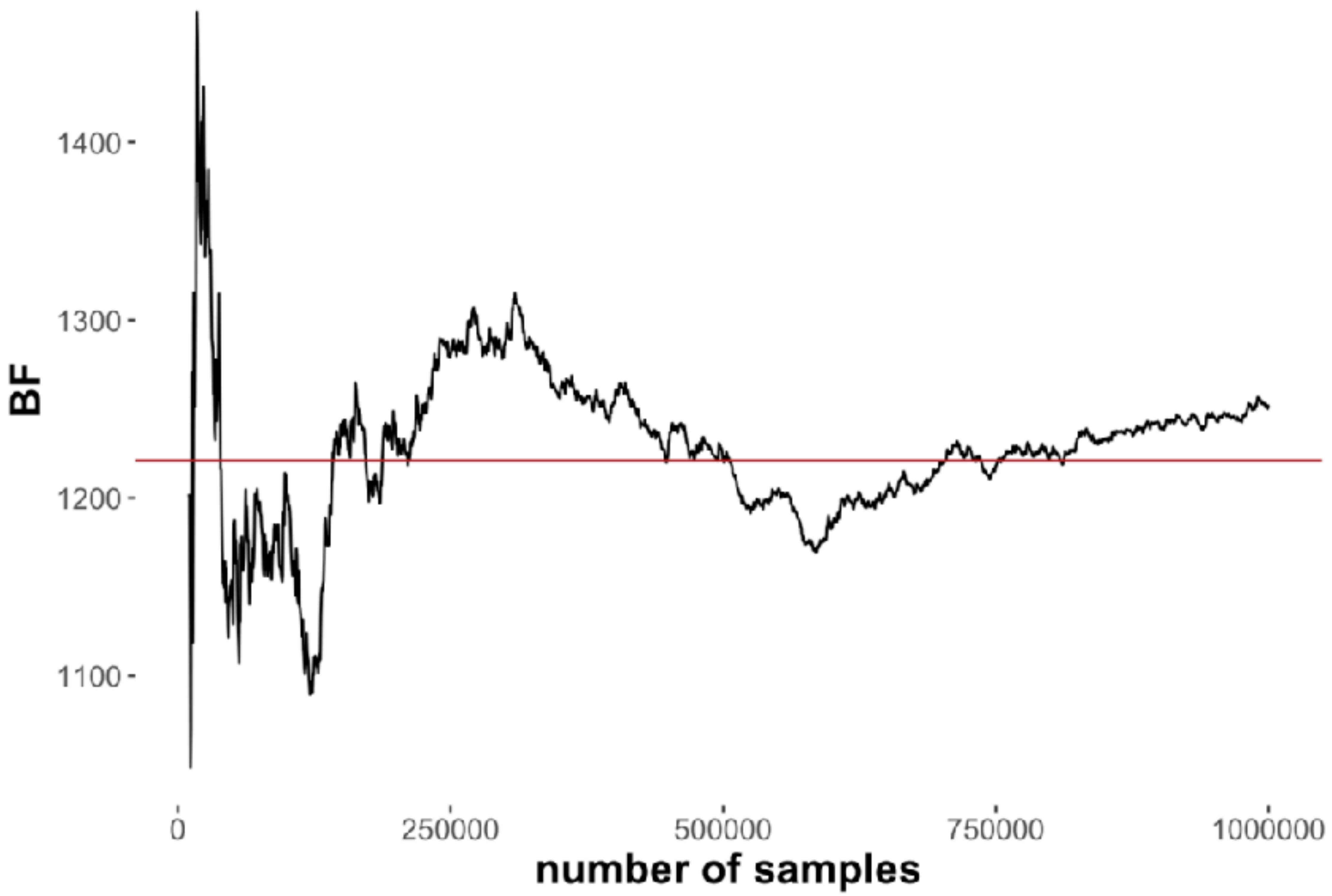
Substantial evidence for the exponential model.

Bayes factors from Monte Carlo simulation

$$P(D, M_i) = \int P(D \mid \theta, M_i)\, P(\theta \mid M_i)\, \mathrm{d}\theta \approx \frac{1}{n} \sum_{\theta_j \sim P(\theta \mid M_i)}^{n} P(D \mid \theta_j, M_i)$$

```r
nSamples = 1000000
a = runif(nSamples, 0, 1.5)
b = runif(nSamples, 0, 1.5)
lhExpVec = lhExp(a,b)
lhPowVec = lhPow(a,b)
paste0("BF in favor of exponential model: ",
            signif(sum(lhExpVec) / sum(lhPowVec)),6)
```

```
## [1] "BF in favor of exponential model: 1250.366"
```



more sampling-based approaches

from naive to brutally efficient

naive Monte Carlo

$$P(D) = \mathbb{E}_{P_{\text{prior}}(\theta)} \left[P(D \mid \theta) \right]$$

generalized harmonic mean sampling

$$P(D) = \left[\mathbb{E}_{P_{\text{posterior}}(\theta|D)} \left[\frac{g_{HM}(\theta)}{P_{\text{prior}}(\theta)\ P(D \mid \theta)} \right] \right]^{-1}$$

importance sampling

$$P(D) = \mathbb{E}_{g_{IS}(\theta)} \left[\frac{P_{\text{prior}}(\theta)\ P(D \mid \theta)}{g_{IS}(\theta)} \right]$$

bridge sampling

$$P(D) = \frac{\mathbb{E}_{g_{\text{proposal}}}(\theta) \left[P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{bridge}}(\theta) \right]}{\mathbb{E}_{P_{\text{posterior}}(\theta|D)} \left[h_{\text{bridge}}(\theta)\ g_{\text{proposal}}(\theta) \right]}$$

generalized harmonic mean sampler

example derivation

$$P(D) = \left[\mathbb{E}_{P_{\text{posterior}}(\theta|D)} \left[\frac{g_{HM}(\theta)}{P_{\text{prior}}(\theta)\ P(D \mid \theta)} \right] \right]^{-1}$$

$$\frac{1}{P(D)} = \frac{P(\theta \mid D)}{P(D \mid \theta)P(\theta)}$$

from Bayes rule

$$= \frac{P(\theta \mid D)}{P(D \mid \theta)P(\theta)} \int g_{HM}(\theta) \mathrm{d}\theta$$

multiply by $1 = \int g_{HM}(\theta)\mathrm{d}\theta$

$$= \int \frac{g_{HM}(\theta)P(\theta \mid D)}{P(D \mid \theta)P(\theta)} \mathrm{d}\theta$$

since $\frac{P(\theta \mid D)}{P(D \mid \theta)P(\theta)}$ is constant (see first line)

$$\approx \frac{1}{n} \sum_{\theta_i \sim P(\theta|D)}^{n} \frac{g_{HM}(\theta_i)}{P(D \mid \theta_i)P(\theta_i)}$$

express as expectation over posterior

bridge sampling

derivation

$$P(D) = \frac{\mathbb{E}_{g_{\text{proposal}}(\theta)}\left[P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{bridge}}(\theta)\right]}{\mathbb{E}_{P_{\text{posterior}}(\theta|D)}\left[h_{\text{bridge}}(\theta)\ g_{\text{proposal}}(\theta)\right]}$$

$$P(D) = P(D)\frac{\int P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{brdg}}(\theta)\ g_{\text{prpsl}}(\theta)\mathrm{d}\ \theta}{\int P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{brdg}}(\theta)\ g_{\text{prpsl}}(\theta)\mathrm{d}\ \theta}$$

multiply by 1

$$= \frac{\int P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{brdg}}(\theta)\ g_{\text{prpsl}}(\theta)\mathrm{d}\ \theta}{\int \frac{P(D \mid \theta)\ P_{\text{prior}}(\theta)}{P(D)}\ h_{\text{brdg}}(\theta)\ g_{\text{prpsl}}(\theta)\mathrm{d}\ \theta}$$

constant $P(D)$ permeates integral

$$= \frac{\int P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{brdg}}(\theta)\ g_{\text{prpsl}}(\theta)\mathrm{d}\ \theta}{\int P(\theta \mid D)\ h_{\text{brdg}}(\theta)\ g_{\text{prpsl}}(\theta)\mathrm{d}\ \theta}$$

Bayes rule

$$= \frac{\mathbb{E}_{g_{\text{proposal}}(\theta)}\left[P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{bridge}}(\theta)\right]}{\mathbb{E}_{P_{\text{posterior}}(\theta|D)}\left[h_{\text{bridge}}(\theta)\ g_{\text{proposal}}(\theta)\right]}$$

express as expectations

bridge sampling

choice of proposal & bridge

$$P(D) = \frac{\mathbb{E}_{g_{\text{proposal}}(\theta)}\left[P(D \mid \theta)\ P_{\text{prior}}(\theta)\ h_{\text{bridge}}(\theta)\right]}{\mathbb{E}_{P_{\text{posterior}}(\theta|D)}\left[h_{\text{bridge}}(\theta)\ g_{\text{proposal}}(\theta)\right]}$$

- proposal function
  - **common choice** (Overstall & Forster 2010): normal distribution whose first two moments match the posterior distribution
    - should resemble the posterior distribution
    - should have sufficient overlap with posterior distribution
- bridge function
  - **optimal choice** (Meng & Wong 1996):

$$h_{\text{bridge}}(\theta) = \left[0.5\ P(D \mid \theta)\ P(\theta) + 0.5\ P(D)\ g_{\text{proposal}}(\theta)\right]$$

  - break circularity (in estimating $P(D)$) by iterative approximation

Gronau et al. (2017, "A tutorial on bridge sampling")

the `bridgesampling` package

example workflow

1. fit models (as usual)

```
fit_n <- brm(
  formula = y ~ x,
  data = data_robust,
  # student prior for slope coefficient
  prior = prior("student_t(1,0,30)", class = "b"),
)
fit_r <- brm(
  formula = y ~ x,
  data = data_robust,
  # student prior for slope coefficient
  prior = prior("student_t(1,0,30)", class = "b"),
  family = student()
)
```

2. update (more samples, include prior)

```
# refit normal model
fit_n_4Bridge <- update(
  fit_n,
  iter = 5e5,
  save_pars = save_pars(all = TRUE)
)
# refit robust model
fit_r_4Bridge <- update(
  fit_r,
  iter = 5e5,
  save_pars = save_pars(all = TRUE)
)
```

3. perform bridge sampling

```
normal_bridge <- bridge_sampler(fit_n_4Bridge, silent = T)
robust_bridge <- bridge_sampler(fit_r_4Bridge, silent = T)
```

4. compute Bayes factor

```
bf_bridge <- bridgesampling::bf(robust_bridge, normal_bridge)
```


Gronau et al. (2020, "bridgesampling: An R Package for Estimating Normalizing Constants") [PCKG]

demo

[calculating a BF with bridge sampling]

cross-validation

ex ante & en route & ex post

marginal likelihoods

prior or posterior predictives?

$$P(D \mid M) = \int P(\theta \mid M)\ P(D \mid \theta, M)\ \mathrm{d}\theta$$

Bayes factors	k-fold cross-validation	LOO	deviance score

prior predictive

posterior predictive

leave-one-out cross-validation

log pointwise density

$$\text{LPD} = \sum_{i=1}^{n} \log P(y_i^{(\text{new})} \mid y) \qquad = \sum_{i=1}^{n} \log \int P(y_i^{(\text{new})} \mid \theta)\ P(\theta \mid y)\ \mathrm{d}\theta$$

$$\approx \sum_{i=1}^{n} \log \left(\frac{1}{S} \sum_{s=1}^{S} P(y_i^{(\text{new})} \mid \theta^s) \right) \qquad \theta^s \sim P(\theta \mid y) \quad \text{(from MCMC)}$$

how (log-)likely is each (new) datum $y_i^{(\text{new})}$ under the posterior predictive distribution given y?

leave-one-out cross-validation

$$\text{LOO} = \sum_{i=1}^{n} \log P(y_i \mid y_{-i}) \qquad = \sum_{i=1}^{n} \log \int P(y_i \mid \theta)\ P(\theta \mid y_{-i})\ \mathrm{d}\theta$$

how (log-)likely is each old datum y_i under the posterior predictive distribution given y_{-i}?

estimated efficiently by **Pareto-smoothed importance sampling**

Pareto-smoothed importance sampling

intuition

$$\text{elpd}_{\text{LOO}} = \sum_{i=1}^{n} \log \int P(y_i \mid \theta)\ P(\theta \mid y_{-i})\ \mathrm{d}\theta$$

Pareto-smoothed importance sampling

$$\text{elpd}_{\text{PSIS-LOO}} \approx \sum_{i=1}^{n} \log \left(\frac{\sum_{s=1}^{S} w_{i,s}\ P(y_i \mid \theta_s)}{\sum_{s=1}^{S} w_{i,s}} \right)$$

θ_s are the posterior samples

$P(y_i \mid \theta_s)$ is the posterior LH of observation i

$w_{i,s}$ are Pareto-smoothed importance weights

Pareto-smoothing

- distribution of naive importance weights can have thick right tails
- therefore, fit Pareto distribution to right tail
- parameter *k* of that fit is indicative of how good the PSIS approximation is

leave-one-out cross-validation

example workflow

```r
fit_n <- brm(
  formula = y ~ x,
  data = data_robust,
  # student prior for slope coefficient
  prior = prior("student_t(1,0,30)", class = "b"),
)
fit_r <- brm(
  formula = y ~ x,
  data = data_robust,
  # student prior for slope coefficient
  prior = prior("student_t(1,0,30)", class = "b"),
  family = student()
)
```

1. fit models (as usual)

```r
loo_comp <- loo_compare(list(normal = loo(fit_n), robust = loo(fit_r)))
loo_comp
```

```
       elpd_diff se_diff
robust     0.0       0.0
normal -131.4      25.9
```

2. compare loo scores with `loo` package

```r
1 - pnorm(-loo_comp[2,1], loo_comp[2,2])
```

```
[1] 0
```

3. test if difference is substantial

method by Ben Lambrecht (2018)

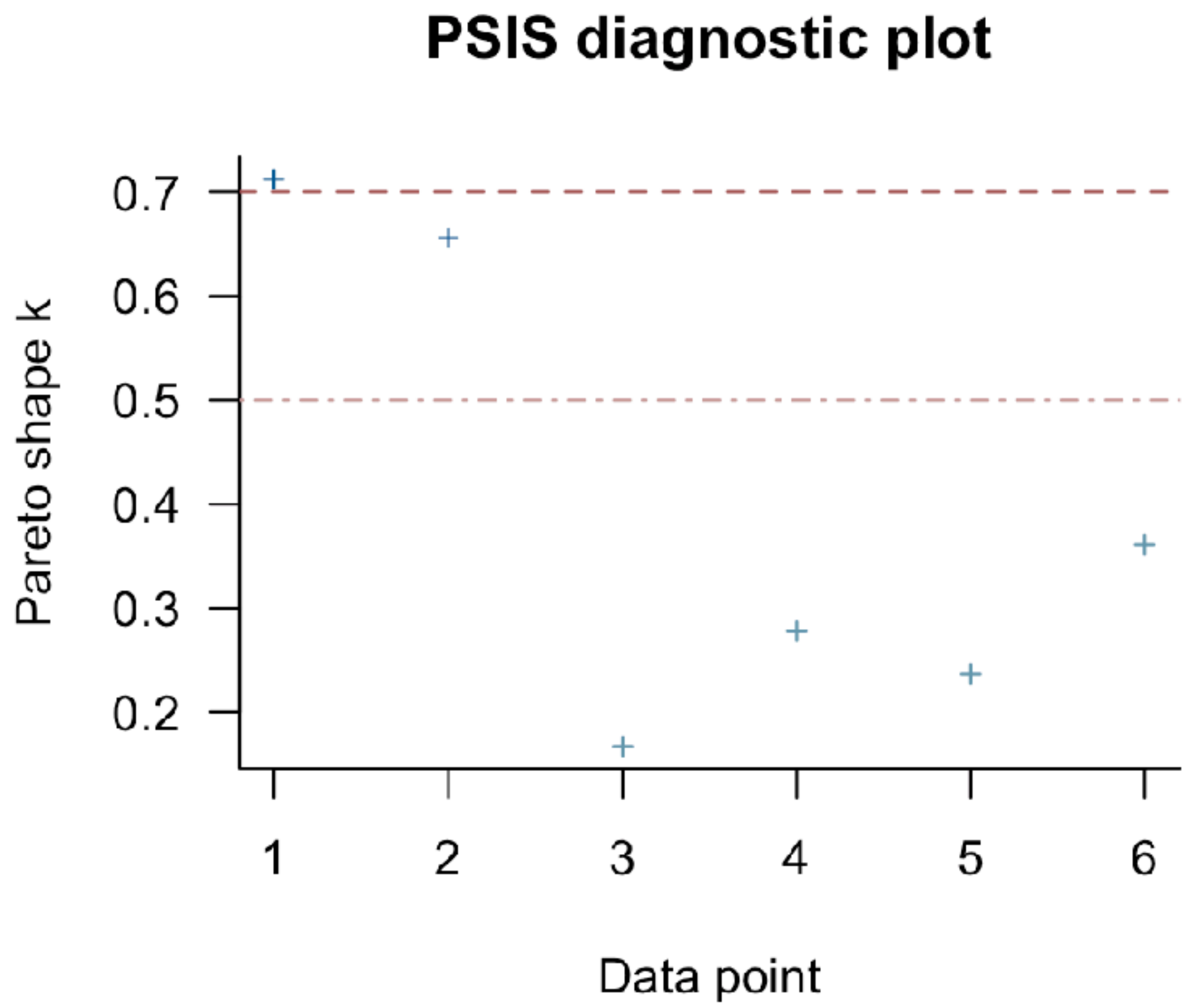
LOO: Pareto-*k* diagnostics

```
> l <- loo(fit_power)
Warning message:
Found 1 observations with a pareto_k > 0.7 in model 'fit_power'. It is recommended to set
'moment_match = TRUE' in order to perform moment matching for problematic observations.
> l

Computed from 16000 by 6 log-likelihood matrix

         Estimate   SE
elpd_loo    -30.6 11.8
p_loo         4.8  2.7
looic        61.3 23.6
------
Monte Carlo SE of elpd_loo is NA.

Pareto k diagnostic values:
                         Count Pct.    Min. n_eff
(-Inf, 0.5]   (good)     4     66.7%   2751
 (0.5, 0.7]   (ok)       1     16.7%   222
   (0.7, 1]   (bad)      1     16.7%   242
   (1, Inf)   (very bad) 0      0.0%   <NA>
See help('pareto-k-diagnostic') for details.
> plot(l)
```



demo

[comparing models with LOO-IC]