

Convergence and all that

Christopher Cahill
Associate Director
19 December 2024



Quantitative Fisheries Center
MICHIGAN STATE UNIVERSITY

Numerical computing and convergence



A crash course in maximum likelihood estimation

- We typically seek to find

$$\hat{\theta} = \arg \max \mathcal{L}(\theta \mid Y)$$

A crash course in maximum likelihood estimation

- We typically seek to find

$$\hat{\theta} = \arg \max \mathcal{L}(\theta \mid Y)$$

$$\hat{\theta} = \arg \min -\ln \mathcal{L}(\theta \mid Y)$$

A crash course in maximum likelihood estimation

- We typically seek to find

$$\hat{\theta} = \arg \max \mathcal{L}(\theta \mid Y)$$

$$\hat{\theta} = \arg \min -\ln \mathcal{L}(\theta \mid Y)$$

- Why



remember the sneak turtles

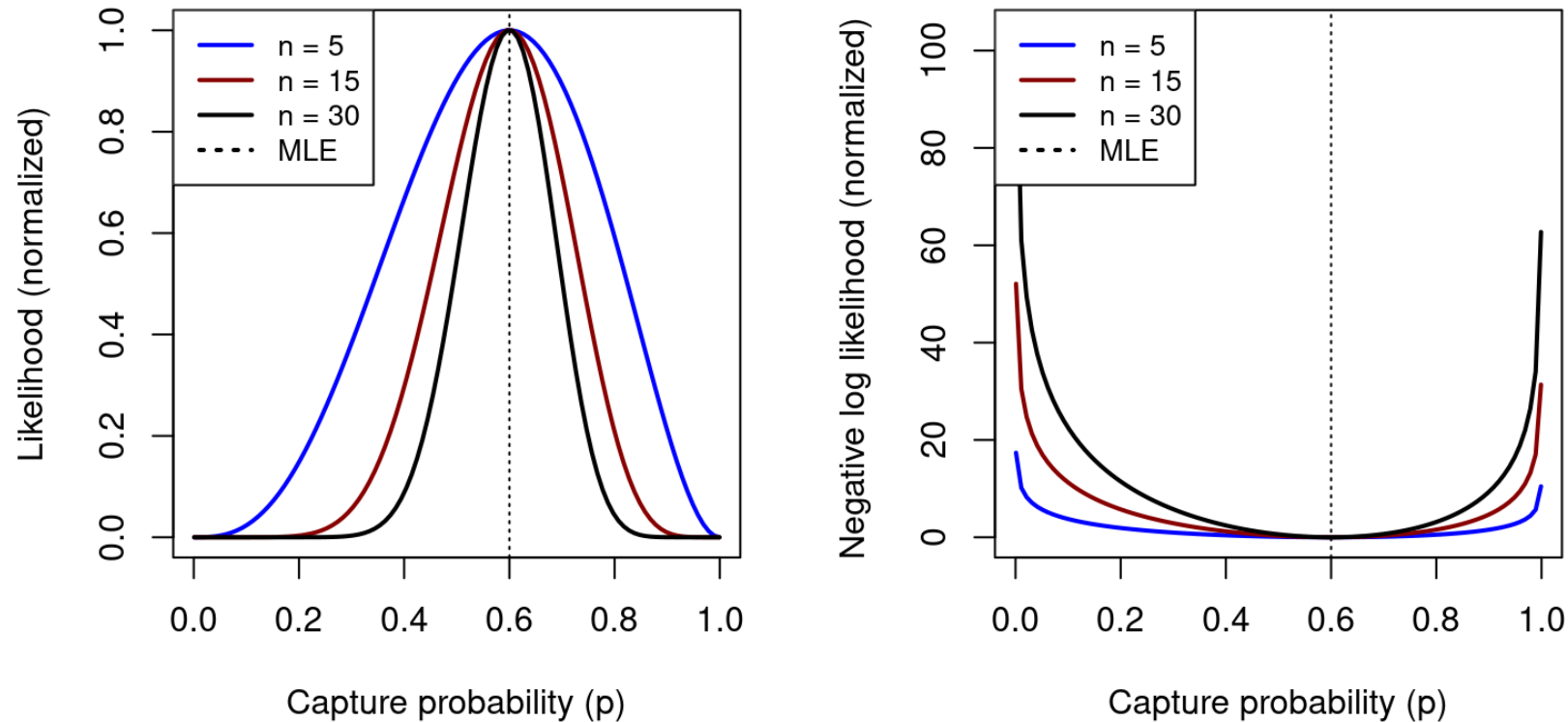


Figure 1. Left panel: Normalized likelihoods for a range of capture probabilities for each of three datasets with differing sample sizes. Right panel: visualizing the normalized negative log-likelihoods corresponding to the likelihoods on the left panel.

Model fitting is (much) more challenging in multi-parameter settings

- High dimensional / many parameter models mean that we must find the minimum of a multi-dimensional surface
- Always need to check that your model is converged
- Convergence can rarely be proven, only disproven
- Because of this there are a number of checks that folks use to evaluate whether a model has converged on a unique solution

The Good, The Bad, and The ugly

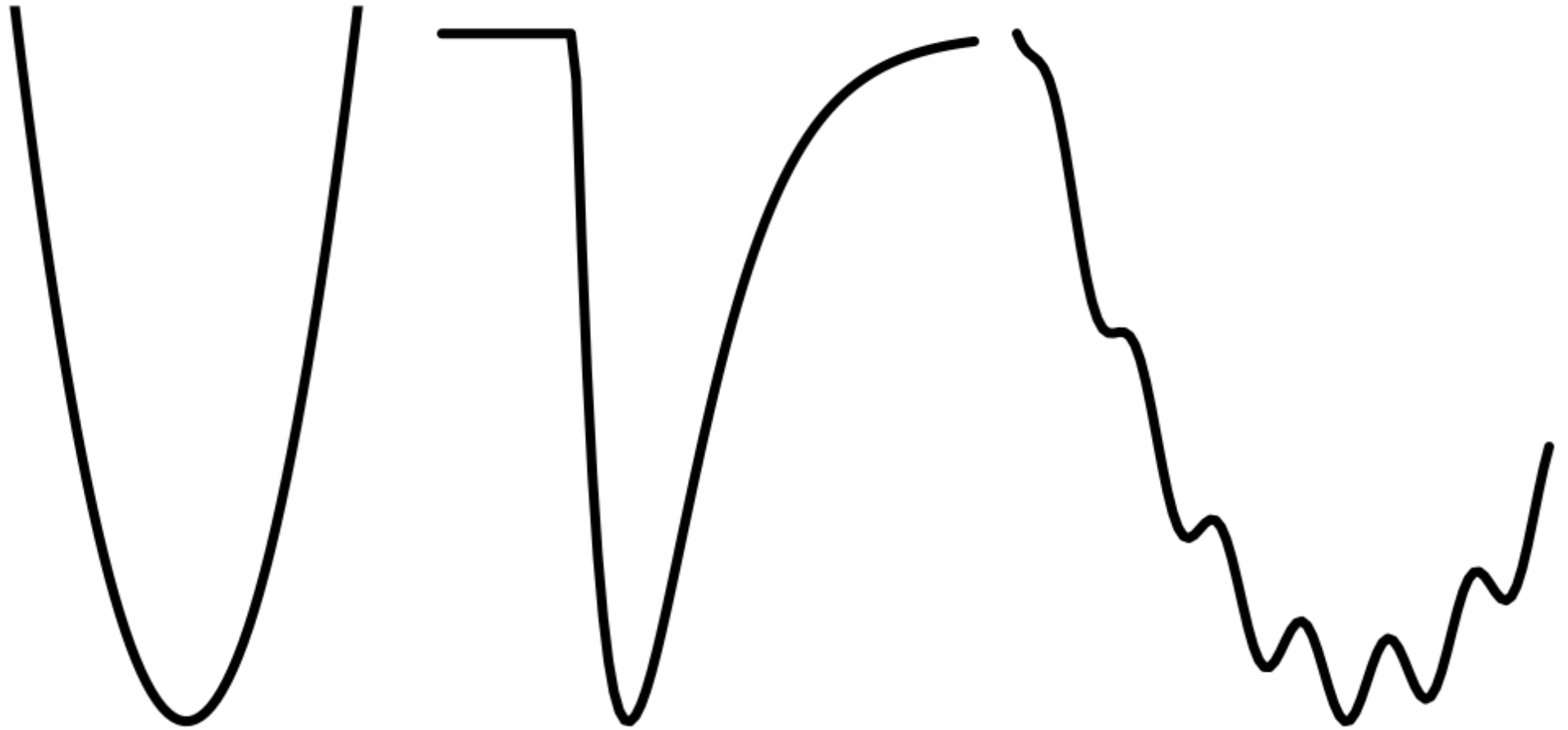


Figure 2. The good, the bad, and the ugly negative log likelihood profiles adapted from Auger-Methe et al. 2021.

An real example of multiple minima

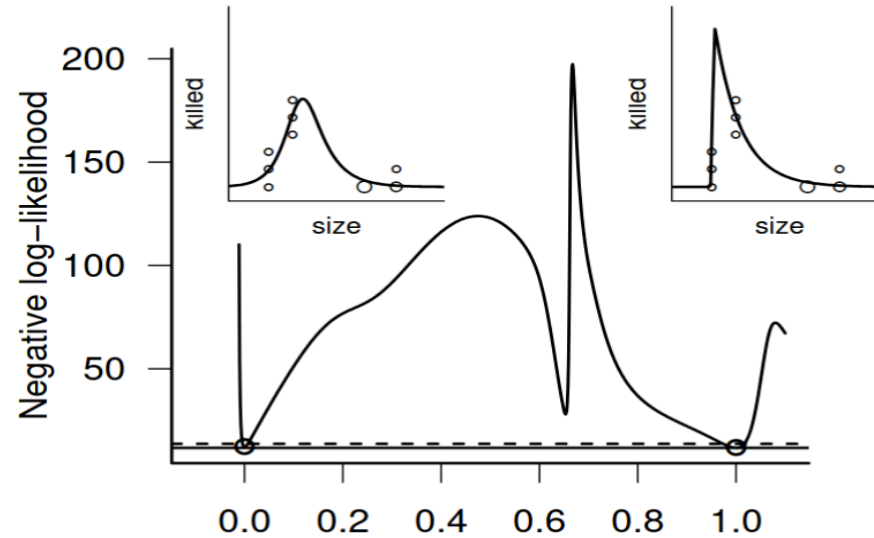


Figure 10: Likelihood slice connecting two negative log-likelihood minima for the modified logistic model of Vonesh and Bolker (2005). The x axis is on an arbitrary scale where $x = 0$ and $x = 1$ represent the locations of the two minima. Subplots show the fits of the curves to the frog predation data for the parameters at each minimum; the right-hand minimum is a slightly better fit ($-L = 11.77$ (right) vs. 12.15 (left)). The horizontal solid and dashed lines show the minimum negative log-likelihood and the 95% confidence cutoff ($-L + \chi^2_1(0.95)/2$). The 95% confidence region includes small regions around both $x = 0$ and $x = 1$.

A minimum toolkit for evaluating convergence in MLE models

- Convergence message from `nlminb()`
- Hessian matrix is positive definite, i.e., it curves upward
- Standard errors are reasonable and not too large
- Absolute values of gradients of parameters are low $< 1e-3$
- Parameters are not estimated near bounds
 - This is critical if you're using an old version of ADMB
- Multiple shooting AKA jitter test
- Likelihood profile

What if we fail convergence tests?

- Must get a model that converges before doing much of anything else
 - e.g., we aren't using “almost maximum likelihood estimation”
 - We certainly do not use the model to inform management

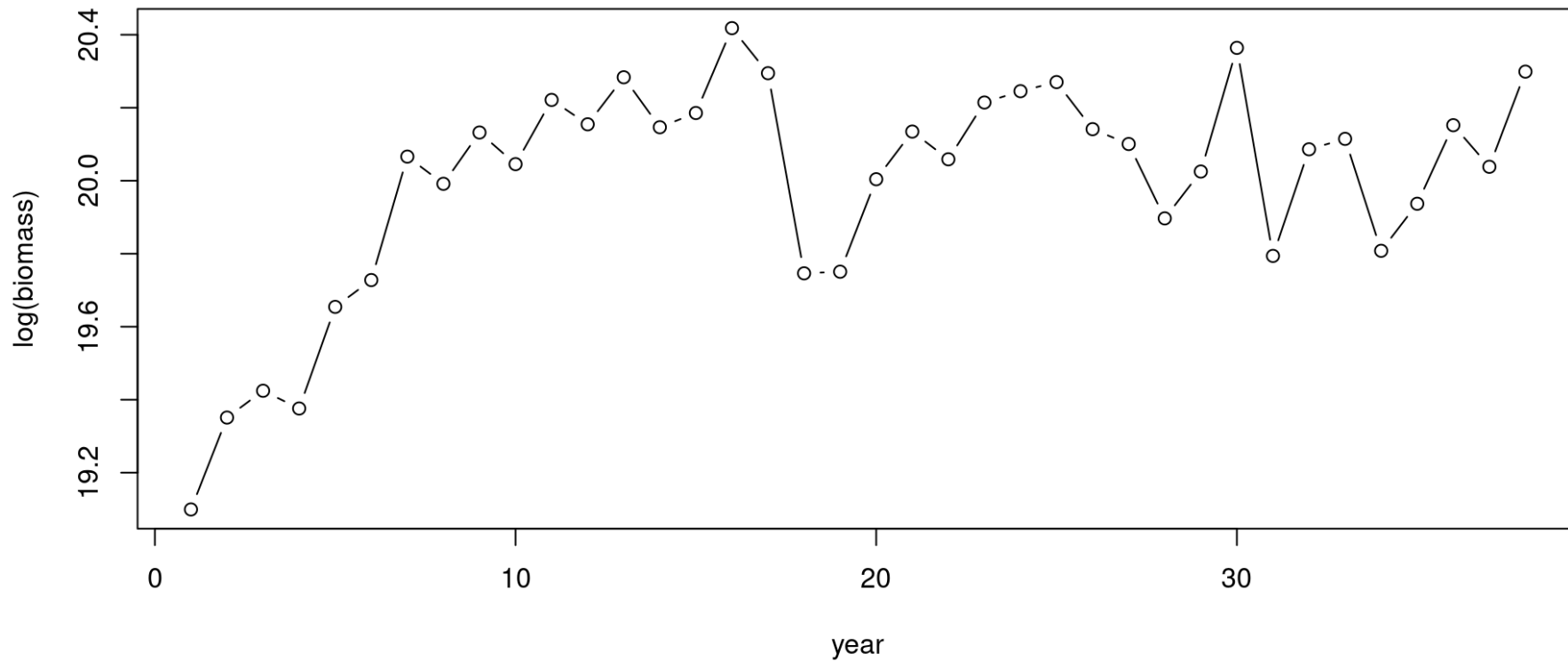
What if we fail convergence tests?



A convergence diagnostic workflow with RTMB

Fit an arbitrary model

```
1 plot(data$log_b_t, type = "b",  
2       xlab = "year", ylab = "log(biomass)"  
3 )
```



Fit an arbitrary model

```
1 obj <- MakeADFun(f, par, random = "log_d_t", silent = TRUE)
2 opt <- nlminb(obj$par, obj$fn, obj$gr)
3 opt
```

```
$par
  log_d0  log_sdp  log_sdo    alpha logit_rho
19.049941 -2.820671 -1.908811  4.664385  1.197744
```

```
$objective
[1] -13.77209
```

```
$convergence
[1] 0
```

```
$iterations
[1] 111
```

```
$evaluations
function gradient
      150       112
```

Can we invert the Hessian matrix of the fixed effects?

```
1 sdr <- sdreport(obj)
2 sdr
```

```
sdreport(.) result
      Estimate Std. Error
log_d0    19.049941  0.1397699
log_sdp   -2.820671  0.7890153
log_sdo   -1.908811  0.2172575
alpha      4.664385  1.0952498
logit_rho  1.197744  0.3072666
Maximum gradient component: 0.0003741121
```


Double check the gradients

```
1 obj$gr(opt$par) # gradients of each parameter
```

```
[1] -6.147670e-07  1.693861e-06  1.143878e-06 -1.044628e-04 -3.741121e-04
```

Double check the gradients

```
1 max(abs(obj$gr(opt$par))) # max gradient
```

```
[1] 0.0003741121
```

```
1 threshold <- 1e-3  
2 max(abs(obj$gr(opt$par))) > threshold # bigger than threshold?
```

```
[1] FALSE
```

Parameters near bounds?

- No (take it on faith for now)

Multiple shooting

- If we assume that we have solved to the MLE, that means that with small deviations from our maximum likelihood estimates `nllminb()` should be able to solve back to the *exact same location*
- This is sometimes called *multiple shooting*

Automate multiple shooting via the jitter test

```
1 opt$par # MLEs
```

log_d0	log_sdp	log_sdo	alpha	logit_rho
19.049941	-2.820671	-1.908811	4.664385	1.197744

```
1 opt$par + rnorm(length(opt$par), sd = 0.1) # jittered MLEs
```

log_d0	log_sdp	log_sdo	alpha	logit_rho
18.968744	-2.919096	-1.953691	4.700522	1.364427

```
1 opt$par + rnorm(length(opt$par), sd = 0.1) # jittered MLEs
```

log_d0	log_sdp	log_sdo	alpha	logit_rho
18.947640	-2.942257	-1.980536	4.604154	1.250253

Automate multiple shooting via the jitter test

```
1 # write a function that jitters your solved MLEs and re-runs
2 do_one <- function() {
3   fit <- nlminb(opt$par + rnorm(length(opt$par), sd = 0.1),
4     obj$fn, obj$gr)
5   fit$par
6 }
7
8 set.seed(333) # cahill is only half evil
9 jit <- replicate(100, do_one()) # 100 iterations
```

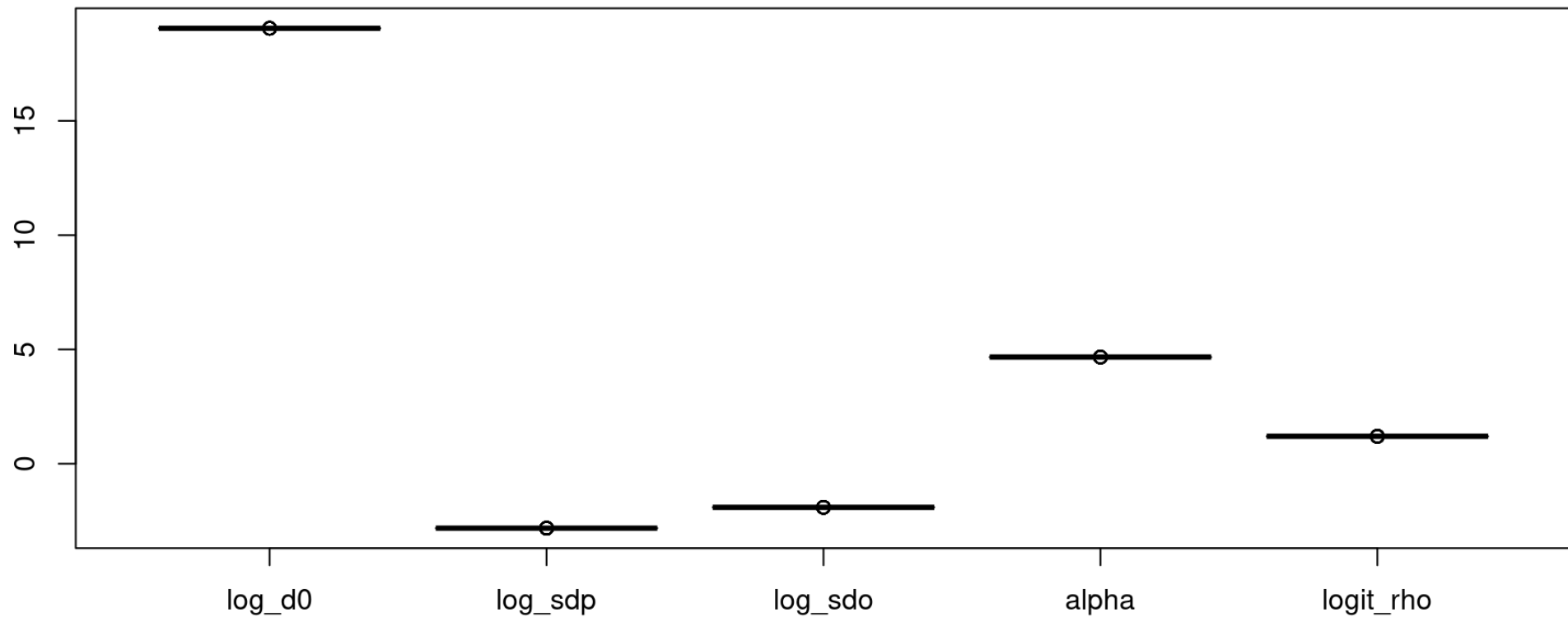
Visualize `jit`

```
1 head(t(jit))
```

	log_d0	log_sdp	log_sdo	alpha	logit_rho
[1,]	19.04994	-2.820672	-1.908811	4.664383	1.197744
[2,]	19.04994	-2.820673	-1.908811	4.664381	1.197745
[3,]	19.04994	-2.820672	-1.908811	4.664384	1.197744
[4,]	19.04994	-2.820672	-1.908811	4.664384	1.197744
[5,]	19.04994	-2.820672	-1.908811	4.664384	1.197744
[6,]	19.04994	-2.820671	-1.908812	4.664386	1.197744

Visualize the jitter test

```
1 boxplot(t(jit))
```



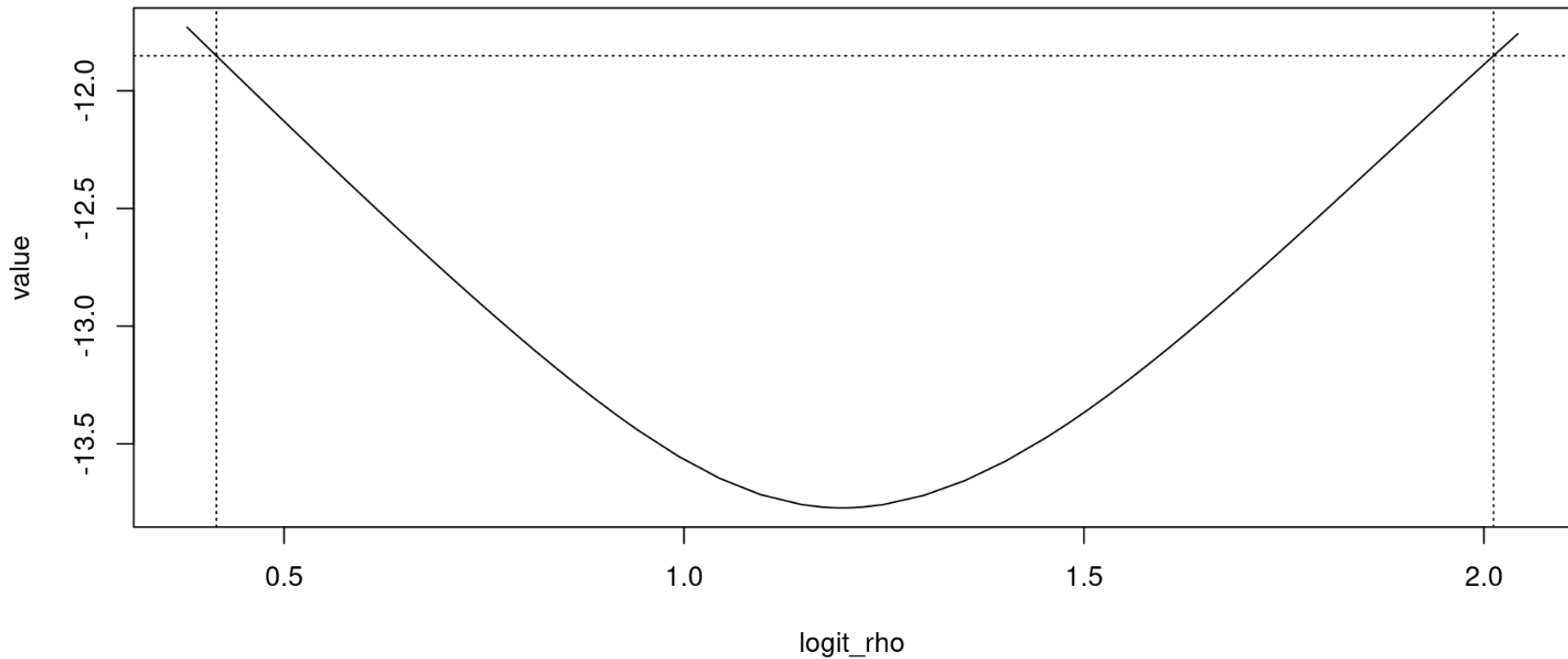
Digging deeper: profile likelihood

```
1 prof <- TMB::tmbprofile(obj, "logit_rho")
```

```
Profile value: -13.77209  
Profile value: -13.77209  
Profile value: -13.77209  
Profile value: -13.77209  
Profile value: -13.77208  
Profile value: -13.77204  
Profile value: -13.77188  
Profile value: -13.77124  
Profile value: -13.76867  
Profile value: -13.7585  
Profile value: -13.71902  
Profile value: -13.65618  
Profile value: -13.5728  
Profile value: -13.47174  
Profile value: -13.41546  
Profile value: -13.35570
```

Digging deeper: profile likelihood

```
1 plot(prof)
```

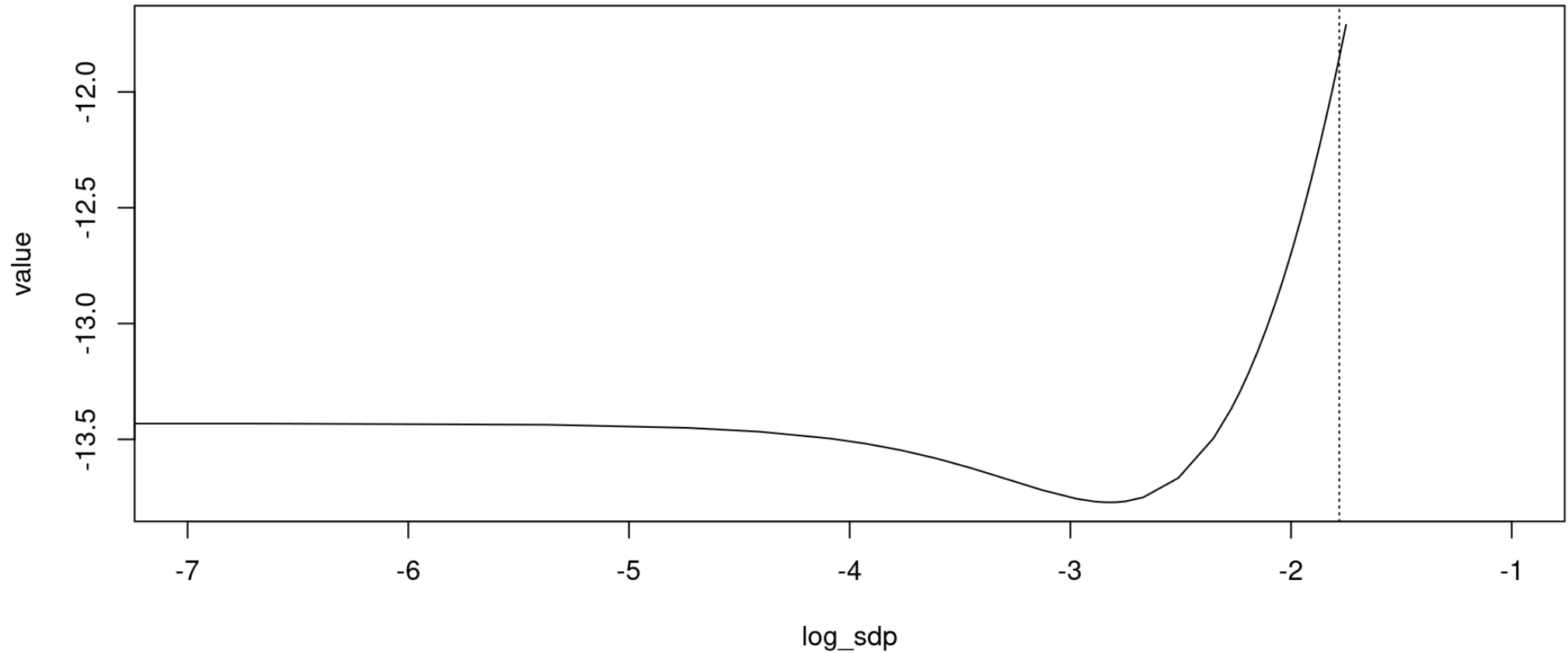


Digging deeper: profile likelihood

```
1 prof <- TMB::tmbprofile(obj, "log_sdp",  
2                           range = c(-7, -1.5), h = 0.01  
3 )
```

```
Profile value: -13.77209  
Profile value: -13.77201  
Profile value: -13.77135  
Profile value: -13.76785  
Profile value: -13.75092  
Profile value: -13.66666  
Profile value: -13.49434  
Profile value: -13.36755  
Profile value: -13.29278  
Profile value: -13.21001  
Profile value: -13.11896  
Profile value: -13.01935  
Profile value: -12.91097  
Profile value: -12.85342  
Profile value: -12.79361  
Profile value: -12.72151
```

Digging deeper: profile likelihood



Wrapping up



WHAT'S THIS?!?! WHAT'S THIS?!?!

imgflip.com

References

- Auger-Méthé, M. et al. 2021. A guide to statespace modeling of ecological time series. *Ecological Monographs* 91(4): e01470. doi:10.1002/ecm.1470.
- Bolker, B. 2006. *Ecological models and data in R*.
- Thorson, J. and K. Kristensen. 2024. *Spatiotemporal models for ecologists*.
- Pawitawn, Y. *In all likelihood: statistical modeling and inference using likelihood*.