

Beta-Logit Transform

Elgin Perry

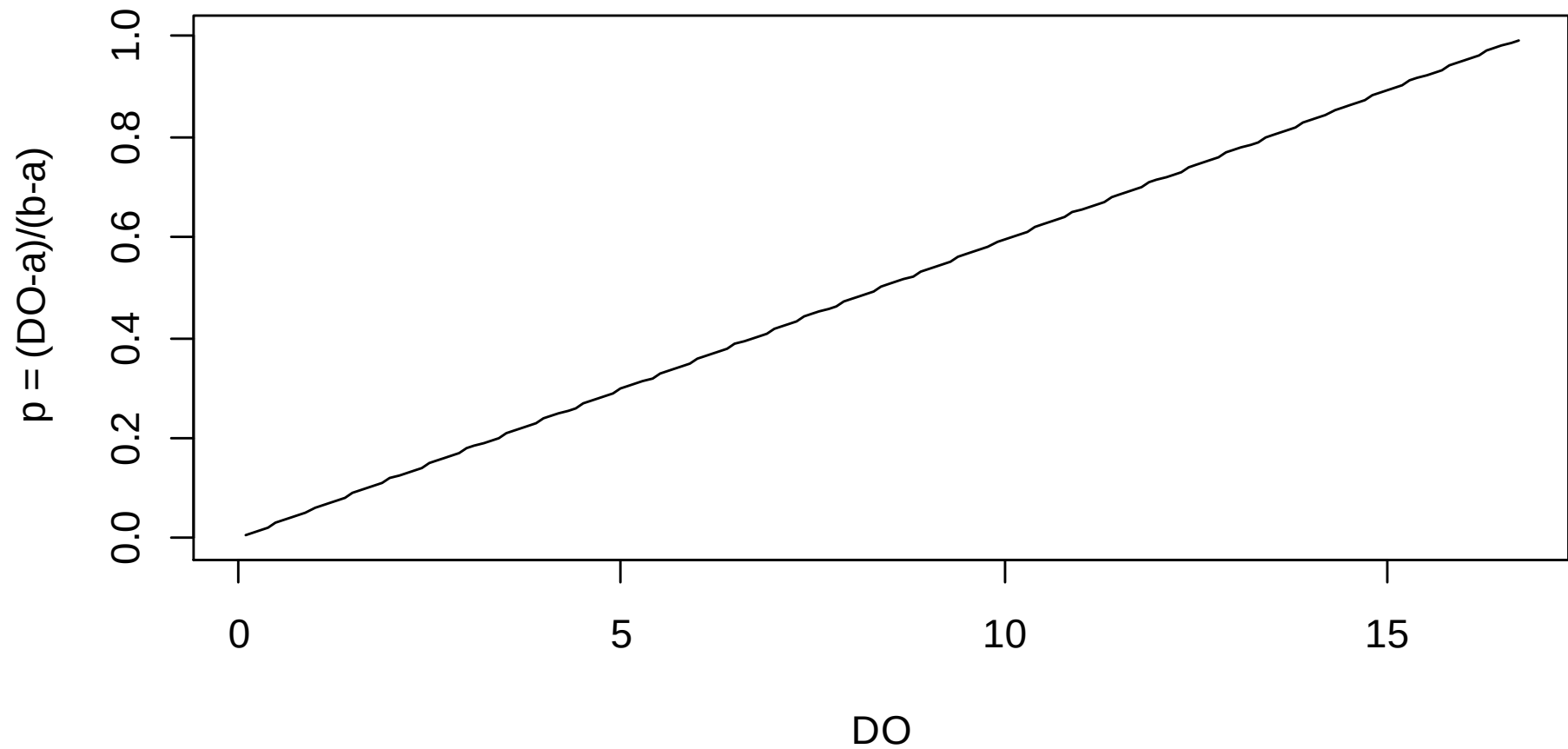
eperry@chesapeake.net

Presentation to BORG 2/27/2023

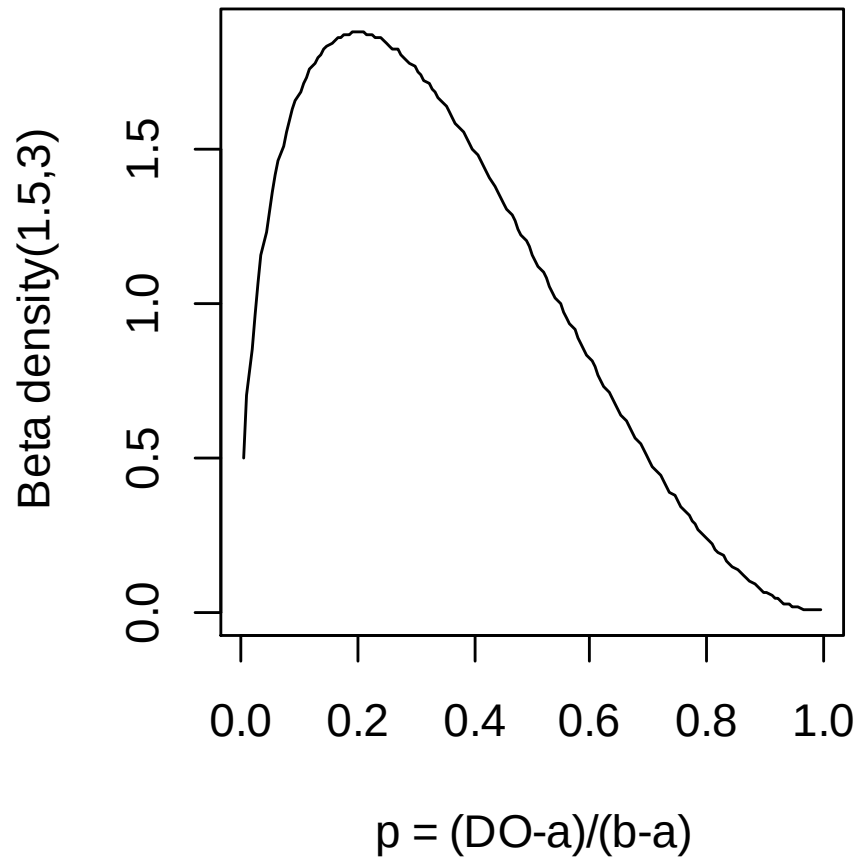
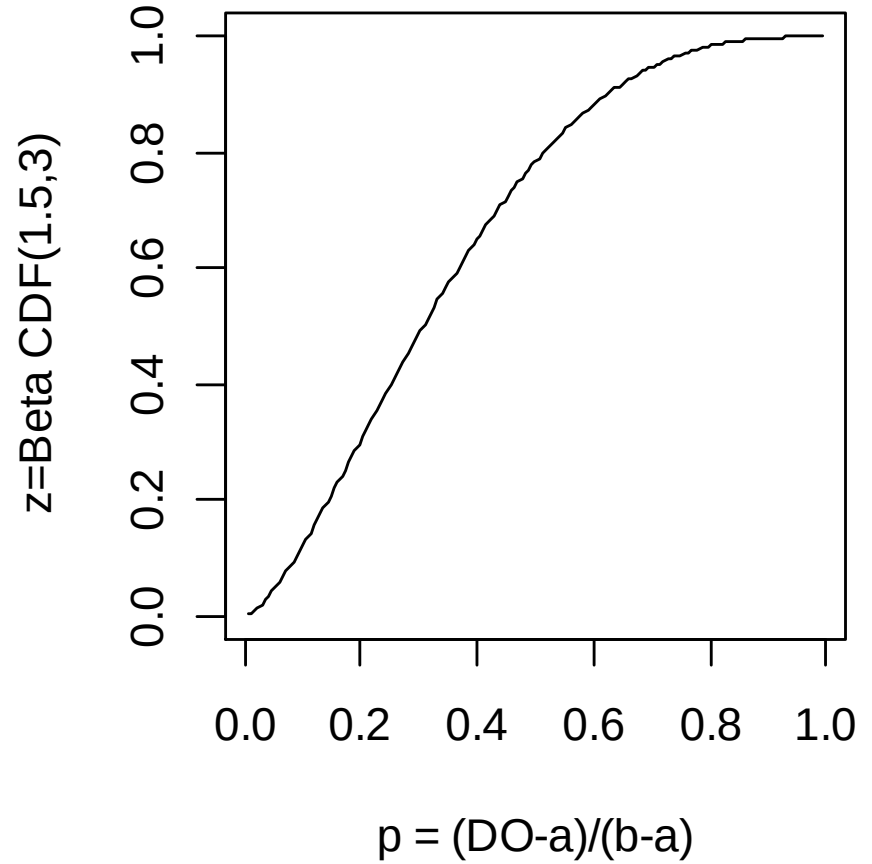
The Beta-Logit transform maps data from a bounded interval (a,b) to an infinite and approximately normal interval.

It can be shown graphically in three steps.

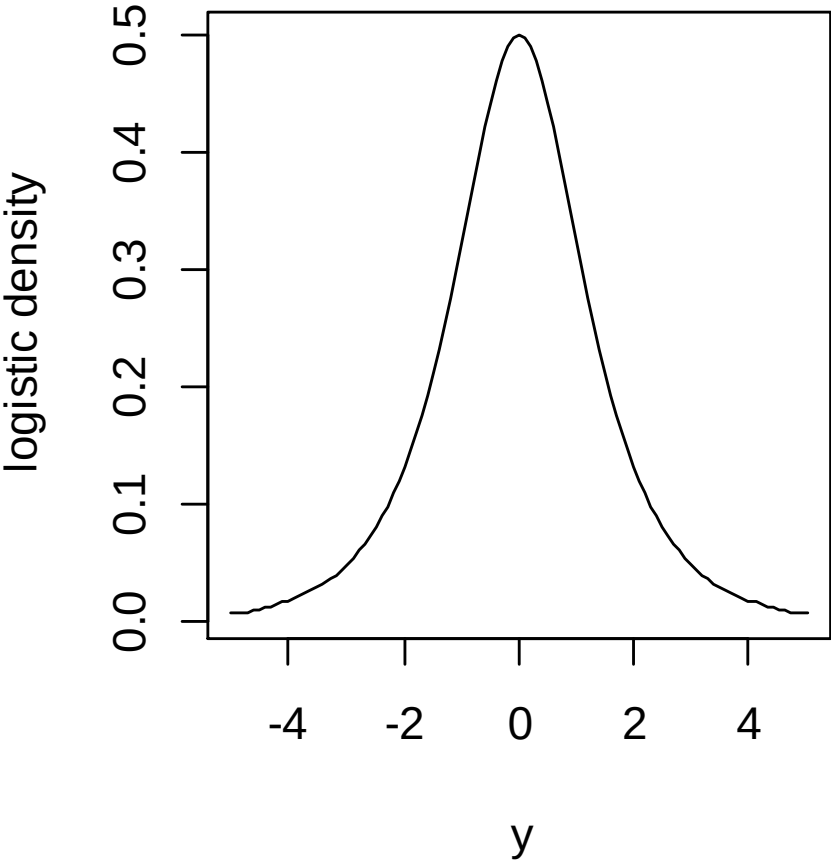
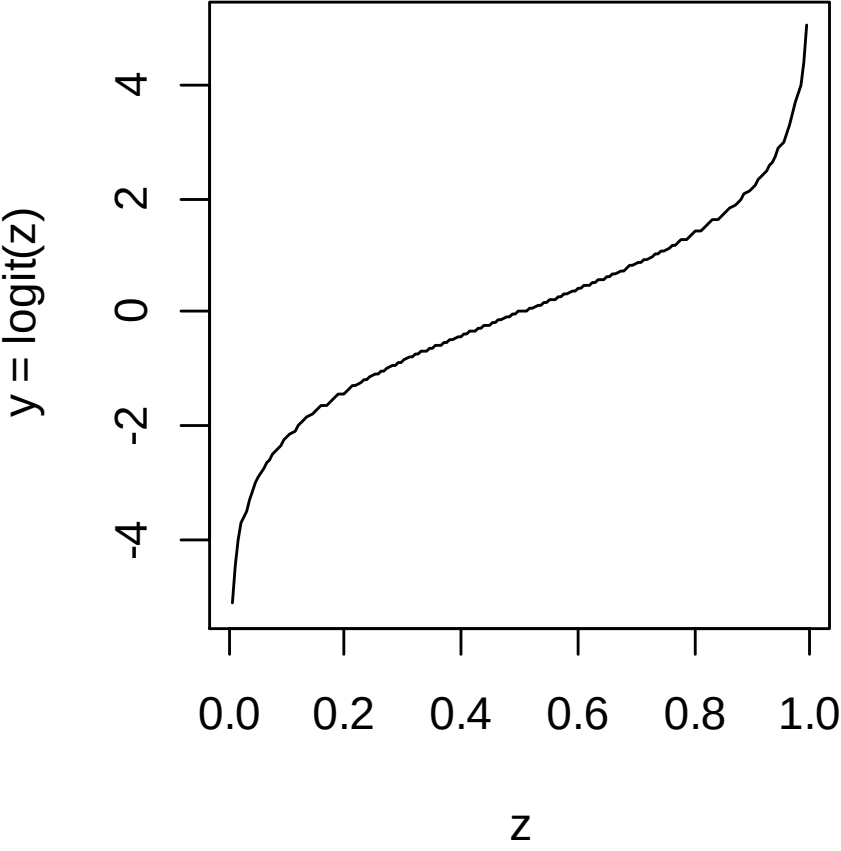
Step 1. Transform DO from (a,b) to p (0,1) as a linear percent.



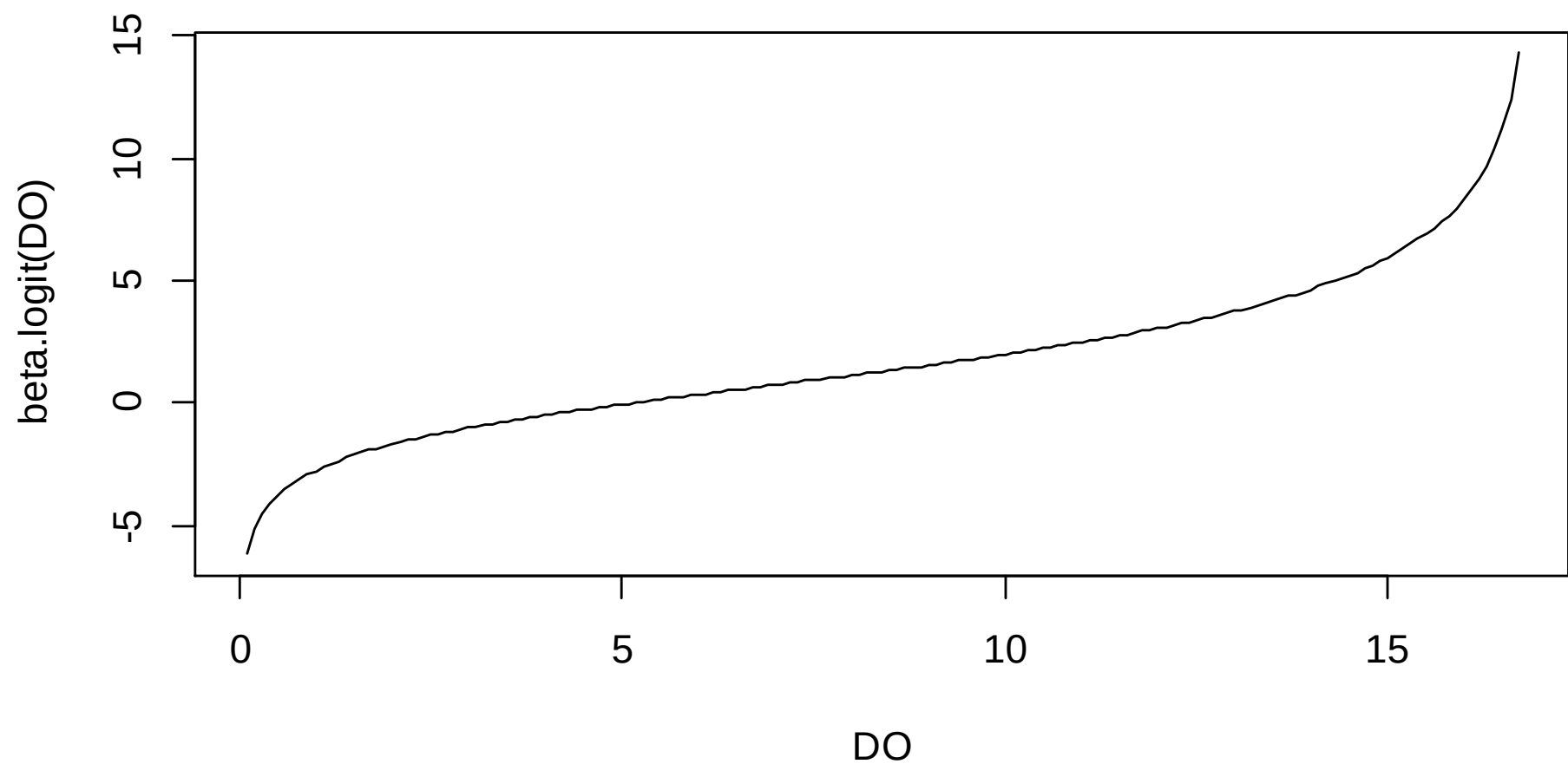
Step 2. Redistribute p using a Beta Distribution to achieve symmetry in the final variable.



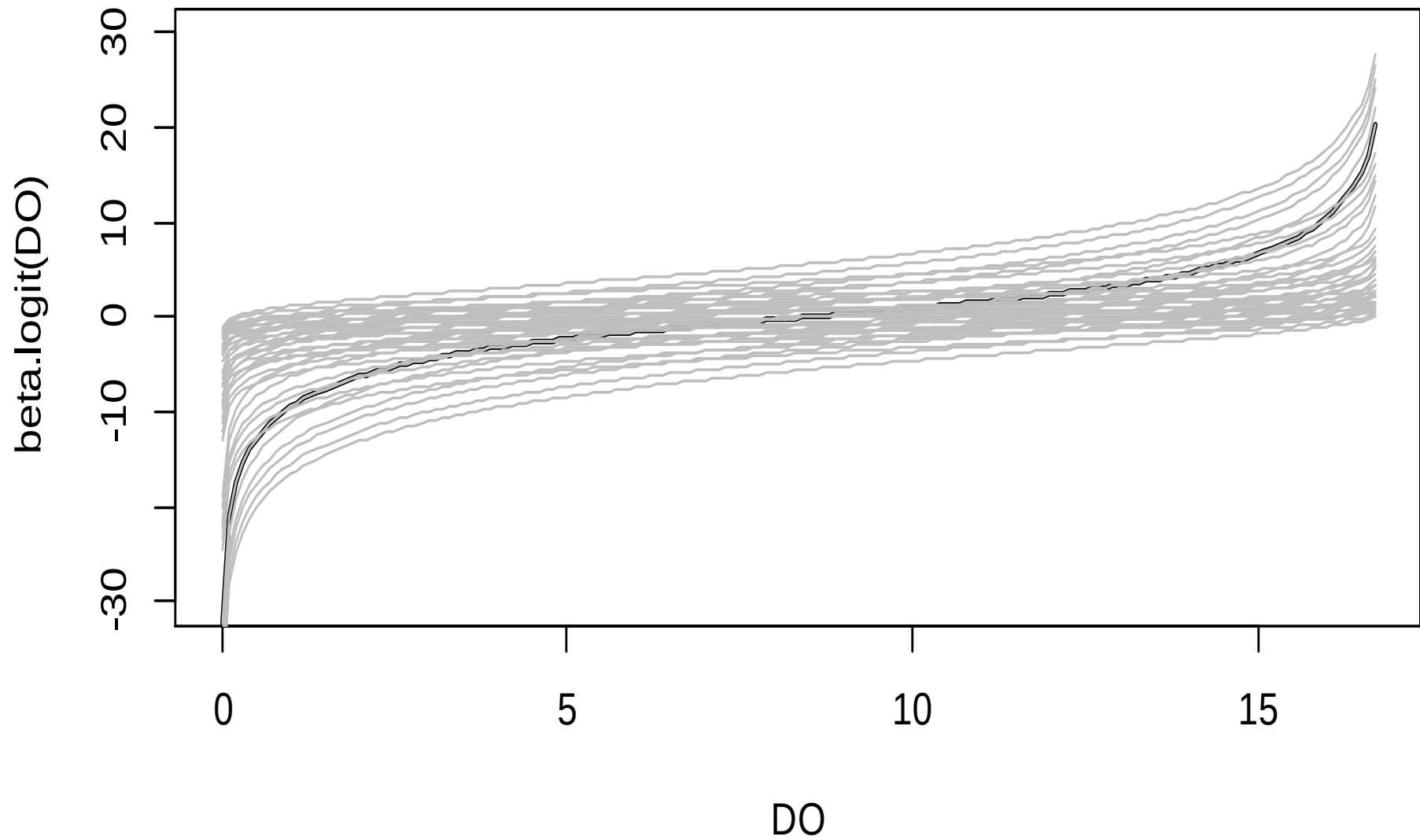
Step 3. Map the output of step 2 (0,1) onto a symmetric and infinite range using a logit transformation.



Putting the three steps together in one function



By “tuning” the beta parameters, a diversity of shapes is possible.



r-functions

`z = beta.lgtTran(x, a, b, lm1, lm2)` Computes Beta-Logit Transform

with arguments:

`a` = lower bound

`b` = upper bound

`lm1` = beta shape parameter

`lm2`=beta shape parameter

`x = lbeta.lgtTran(z, a, b, lm1, lm2)` Computes the Inverse of Beta-Logit Transform

Same arguments

Trial of Beta-Logit Transform

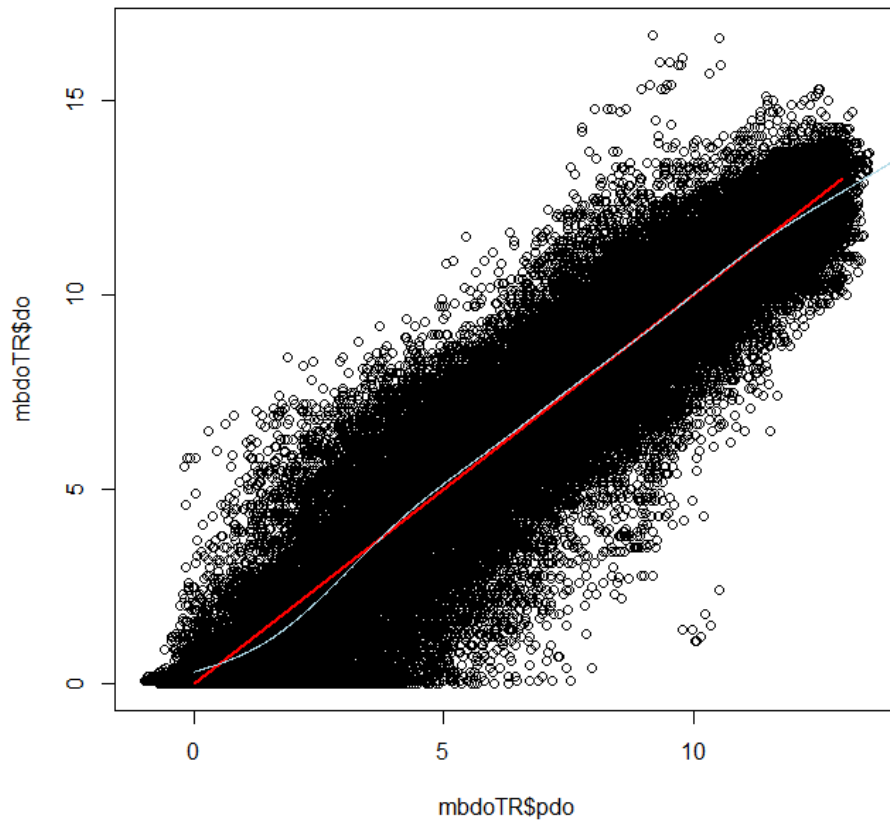
Training data

R-square for original scale DO fit:	0.866
R-square for beta logit transform in original DO scale:	0.8666

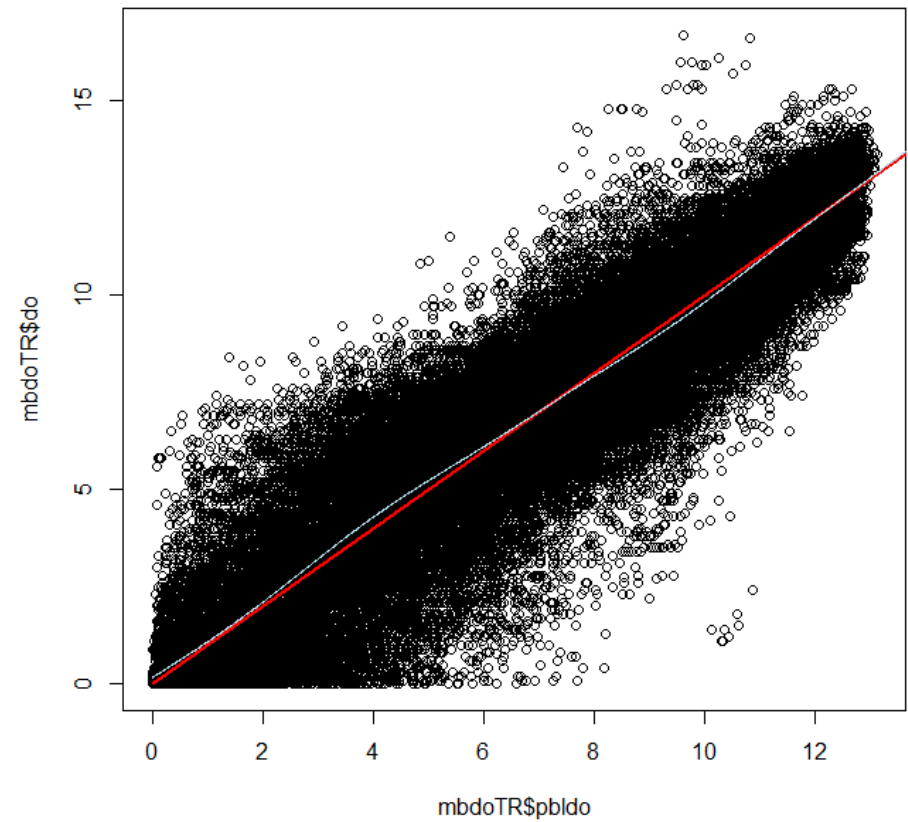
Validation Data

R-square for original scale DO fit:	0.8381
R-square for beta logit transform in original DO scale:	0.8396

Observed vs. Predicted Plots

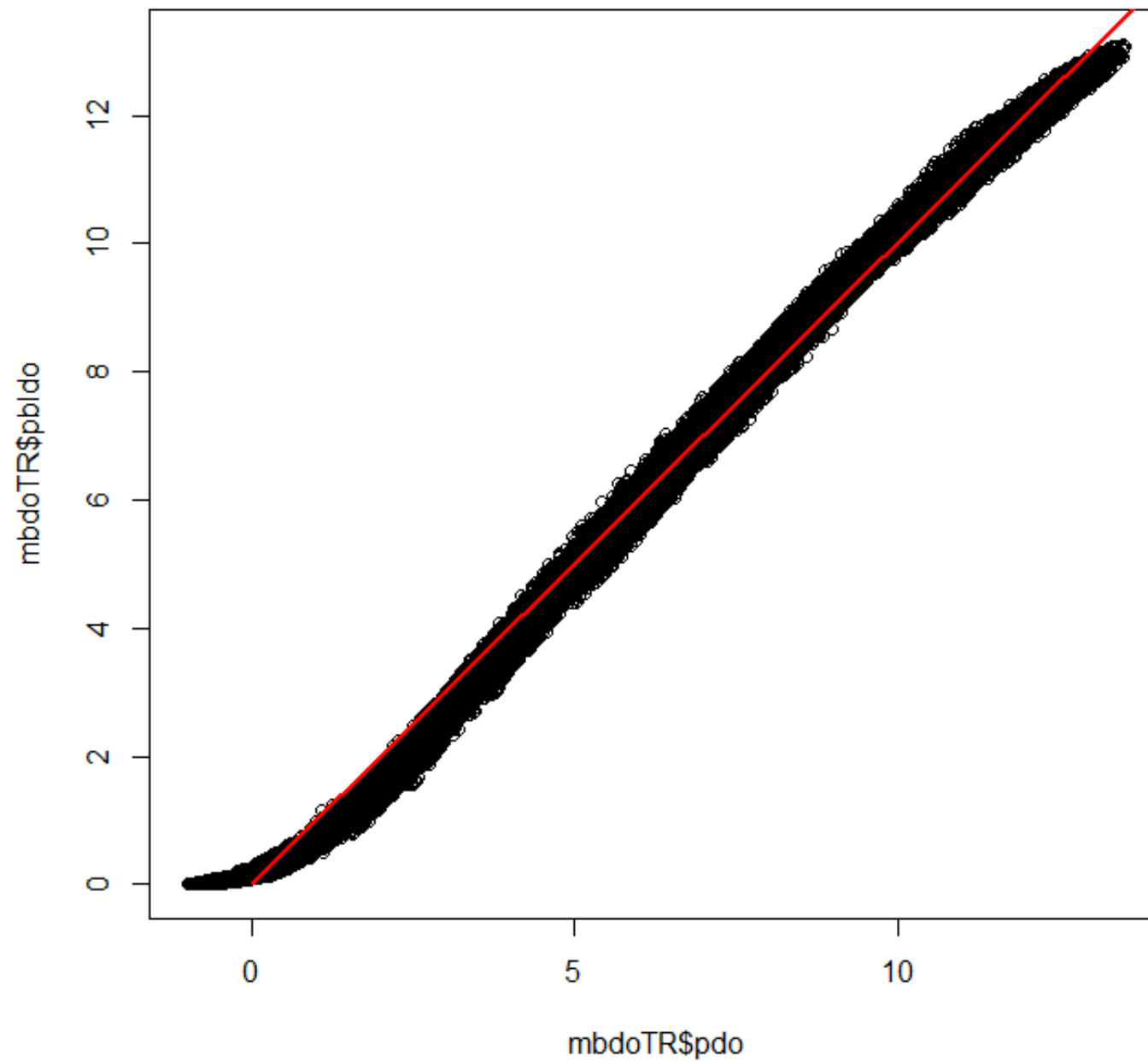


Original Scale



Beta-Logit Transform

Beta-Logit Predictions vs. Original Scale Predictions.



Open questions:

Is this worth the trouble?

Do we need a better theoretical justification for this method?

Peer Review?

Can the beta parameters be estimated by maximum likelihood?

Do we really have an upper bound for DO?

Transformation bias?

Standard Errors?