

MLDS: Maximum Likelihood Difference Scaling in R

Kenneth Knoblauch

Inserm U 846

Département Neurosciences Intégratives

Institut Cellule Souche et Cerveau

Bron, France

Laurence T. Maloney

Department of Psychology

Center for Neural Science

New York University

New York, NY USA

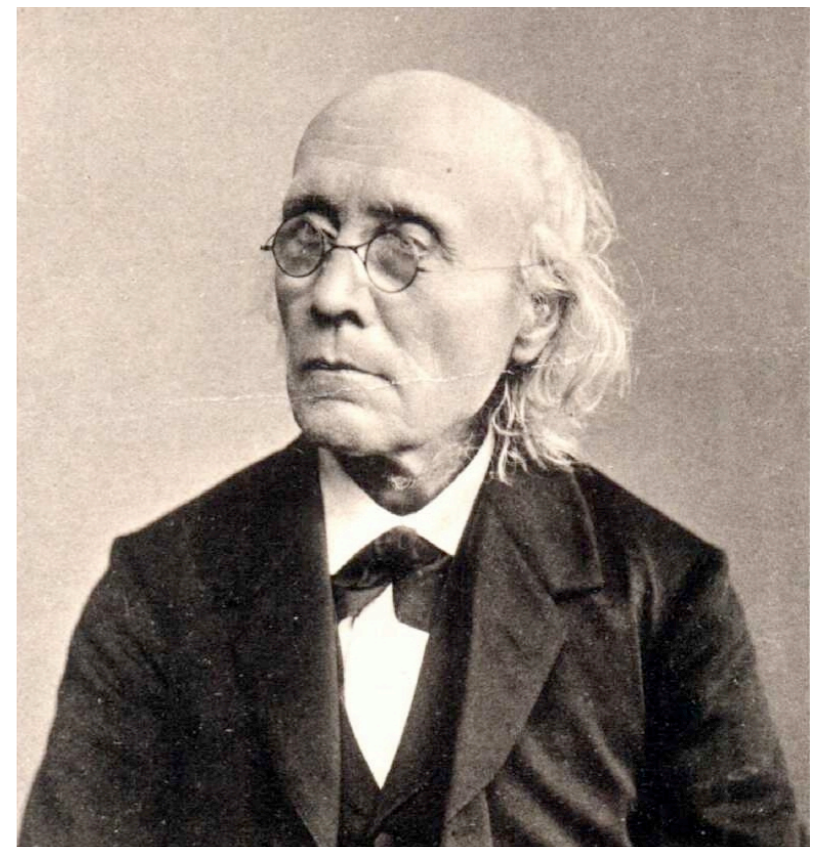
Inserm

Institut national
de la santé et de la recherche médicale

Psychophysics, qu'est-ce que c'est ?

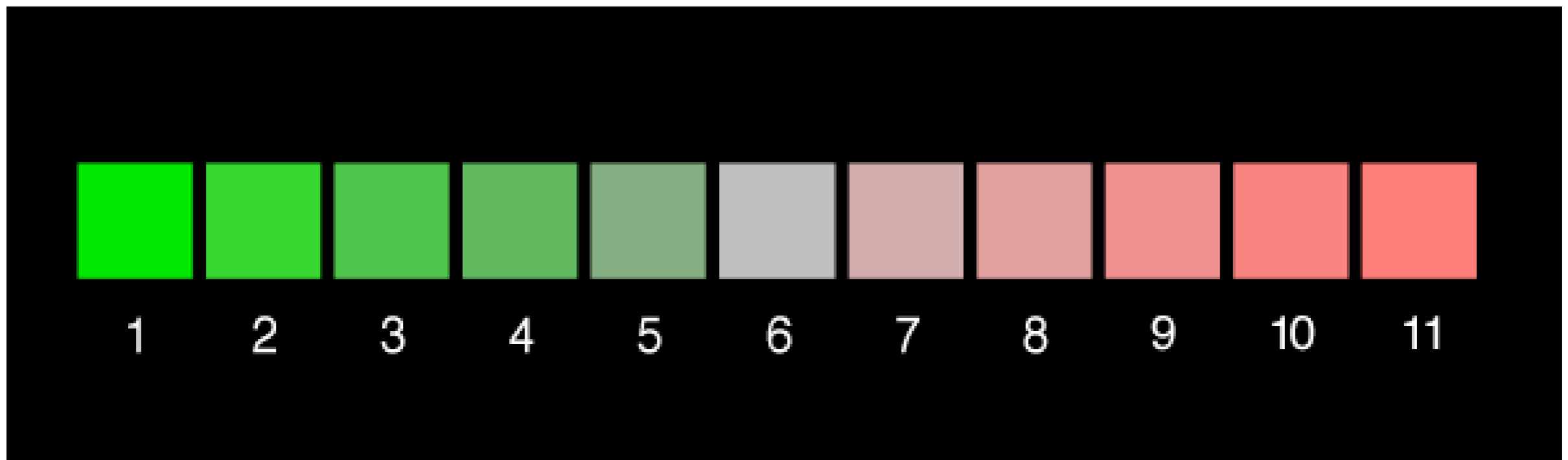
A body of techniques and analytic methods to study the relation between physical stimuli and the organism's (classification) behavior to infer internal states of the organism or their organization.

Gustav Fechner (1801 - 1887)



Difference scaling is a psychophysical procedure
used to estimate a perceptual scale
for stimuli distributed along a physical continuum.

Example: series along a line in tristimulus space,
What are the perceptual distances between samples?



Example: VQ compressed images,

Up to what compression rate can the observer detect no loss of image quality?

1:1



6:1



9:1



12:1



15:1



18:1



21:1



24:1



27:1



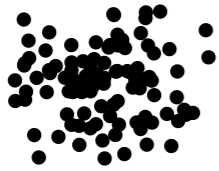
30:1



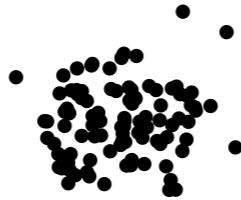
Charrier, Maloney, Cherifi & Knoblauch, submitted

Example: Correlation in scatterplots

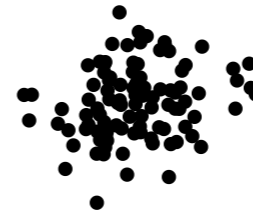
$r = 0$



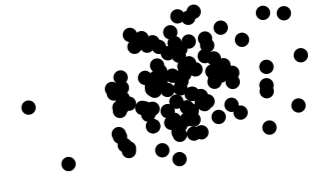
$r = 0.1$



$r = 0.2$



$r = 0.3$



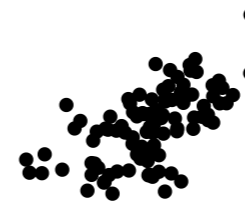
$r = 0.4$



$r = 0.5$



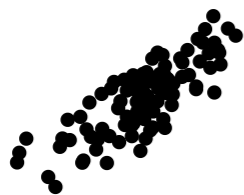
$r = 0.6$



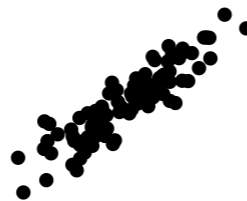
$r = 0.7$



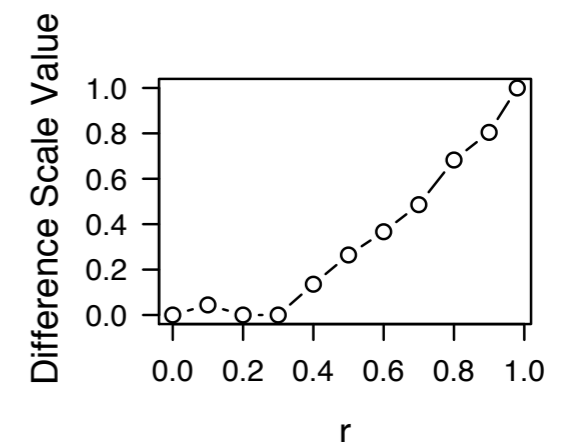
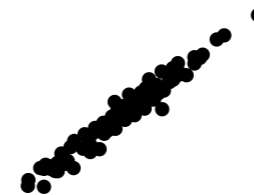
$r = 0.8$



$r = 0.9$



$r = 0.98$



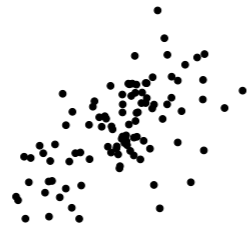
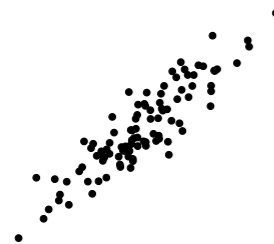
Difference Scaling: Experimental Procedure

From a set of p stimuli, $\{I_1 < I_2 < \dots < I_p\}$,

a random quadruple, $\{I_a, I_b; I_c, I_d\}$,
is chosen (w/out replacement) and presented
to the observer as in this example,



Between which pair (upper/lower) is the
perceived difference greatest?



For a sequence of p stimuli, there are $\binom{p}{4}$ non-overlapping quadruples:
 210 quadruples for $p = 10$
 330 quadruples for $p = 11$, etc.

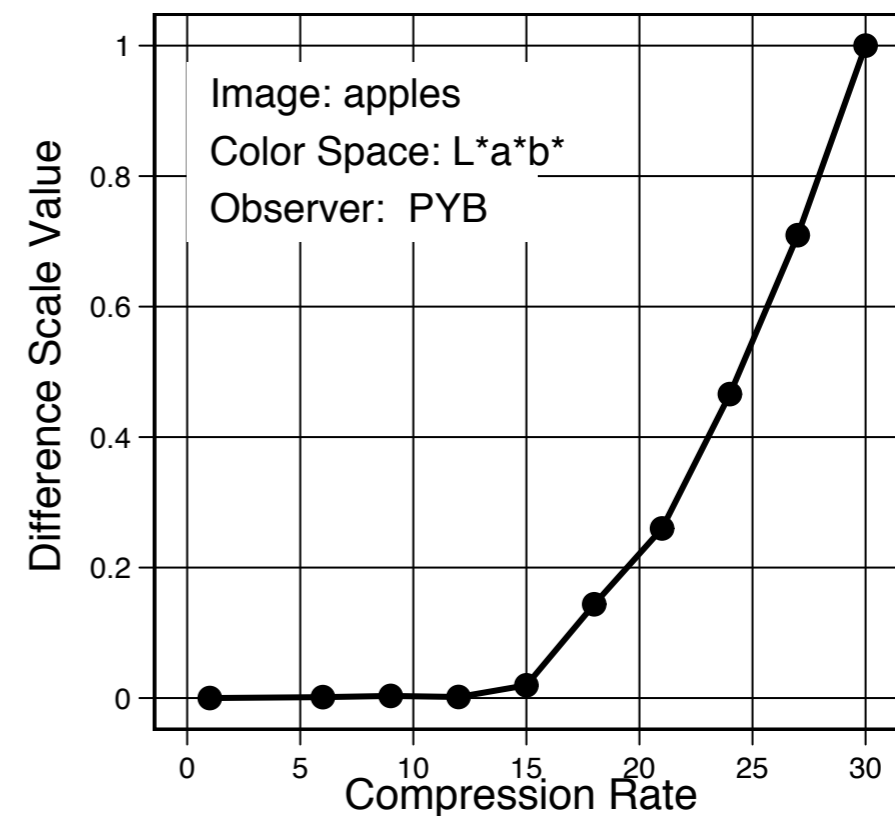
At the termination of an experiment (about 15 minutes), the data are stored in a 5 column data frame. 6 lines of an example are shown here.

	resp	S1	S2	S3	S4
1	0	4	8	2	3
2	1	2	3	6	11
3	1	2	6	7	10
4	0	4	11	1	2
5	0	9	11	7	8
6	0	7	10	1	3

One column for the response indicating whether the upper (1) or lower (0) pair was chosen.
 4 columns for the indices of the stimuli in the sequence, $1 : p$

The aim of the Maximum Likelihood Difference Scaling (MLDS) procedure is to estimate scale values, $(\psi_1, \psi_2, \dots, \psi_p)$, that best capture the observer's judgments of the perceptual difference between the stimuli in each pair.

The MLDS package, available on CRAN, provides tools for performing this analysis in R. An example scale obtained from an observer for the “apples” sequence of VQ compressed images is shown on the right:



The decision model

Given a quadruple, $q = (a, b; c, d)$,
from a single trial, we assume that the observer
chooses the upper pair to be further apart
when

$$\Delta(a, b; c, d) = |\psi_d - \psi_c| - |\psi_b - \psi_a| + \epsilon > 0,$$

where ψ_i are estimated scale values, $\epsilon \sim N(0, \sigma)$,
and σ a scale factor.

Estimation of Scale Values

Maloney and Yang (2003) used a direct method for estimating the maximum likelihood scale values,

$$L(\Psi, \sigma) = \prod_{k=1}^n \Phi \left(\frac{\delta(\mathbf{q}^k)}{\sigma} \right)^{1-R_k} \left(1 - \Phi \left(\frac{\delta(\mathbf{q}^k)}{\sigma} \right) \right)^{R_k}$$

where

$$\Psi = (\psi_2, \psi_3, \dots, \psi_{p-1})$$

$$\delta(\mathbf{q}^k) = |\psi_d - \psi_c| - |\psi_b - \psi_a|$$

Φ is the cumulative standard Gaussian (a probit analysis)

R_k is 0/1 if the judgment is lower/upper

$\psi_1 = 0, \psi_p = 1$ for identifiability,

leaving $p - 1$ parameters to estimate

Estimation of Scale Values

The problem can also be conceptualized as a GLM.

Each level of the stimulus is treated as a covariate in the model matrix, taking on values of 0 or ± 1 in the design matrix,

depending on the presence of the stimulus in a trial and

its weight in the decision variable, with absolute value signs removed.

	resp	S1	S2	S3	S4	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}
1	0	4	8	2	3	0	1	-1	-1	0	0	0	1	0	0	0
2	1	2	3	6	11	0	1	-1	0	0	-1	0	0	0	0	1
3	1	2	6	7	10	0	1	0	0	0	-1	-1	0	0	1	0
4	0	4	11	1	2	1	-1	0	-1	0	0	0	0	0	0	1
5	0	9	11	7	8	0	0	0	0	0	0	1	-1	-1	0	1
6	0	7	10	1	3	1	0	-1	0	0	0	-1	0	0	1	0

For model identifiability, we drop the first column (fixing $\psi_1 = 0$ and $\sigma = 1$).

Estimation of Scale Values

```
> kk.ix <- make.ix.mat(kk)
```

```
> head(kk.ix)
```

	resp	stim.2	stim.3	stim.4	stim.5	stim.6	stim.7	stim.8	stim.9	stim.10	stim.11
1	1	1	0	-1	0	-1	0	1	0	0	0
2	1	0	0	-1	0	-1	0	0	1	0	0
3	1	1	-1	0	0	0	-1	0	1	0	0
4	1	1	0	0	-1	-1	1	0	0	0	0
5	0	0	-1	0	0	-1	1	0	0	0	0
6	0	0	0	0	-1	-1	0	0	1	0	0

$$\eta (E [Y]) = X\beta$$

```
> glm(resp ~ . - 1, family = binomial( "probit" ), data = kk.ix)
```

The MLDS package

The MLDS package provides a modeling function, `mlds()`, that is essentially a wrapper for either `glm()` or `optim()`, and will enable estimation of the perceptual scale values, given a data frame with the previously described structure.

```
mlds(data, stimulus, method = "glm", lnk = "probit",  
      opt.meth = "BFGS", opt.init = NULL,  
      control = glm.control(maxit = 50000, epsilon = 1e-14),  
      ... )
```

It outputs an S3 object of class 'mlds' which can be examined further using several method functions:

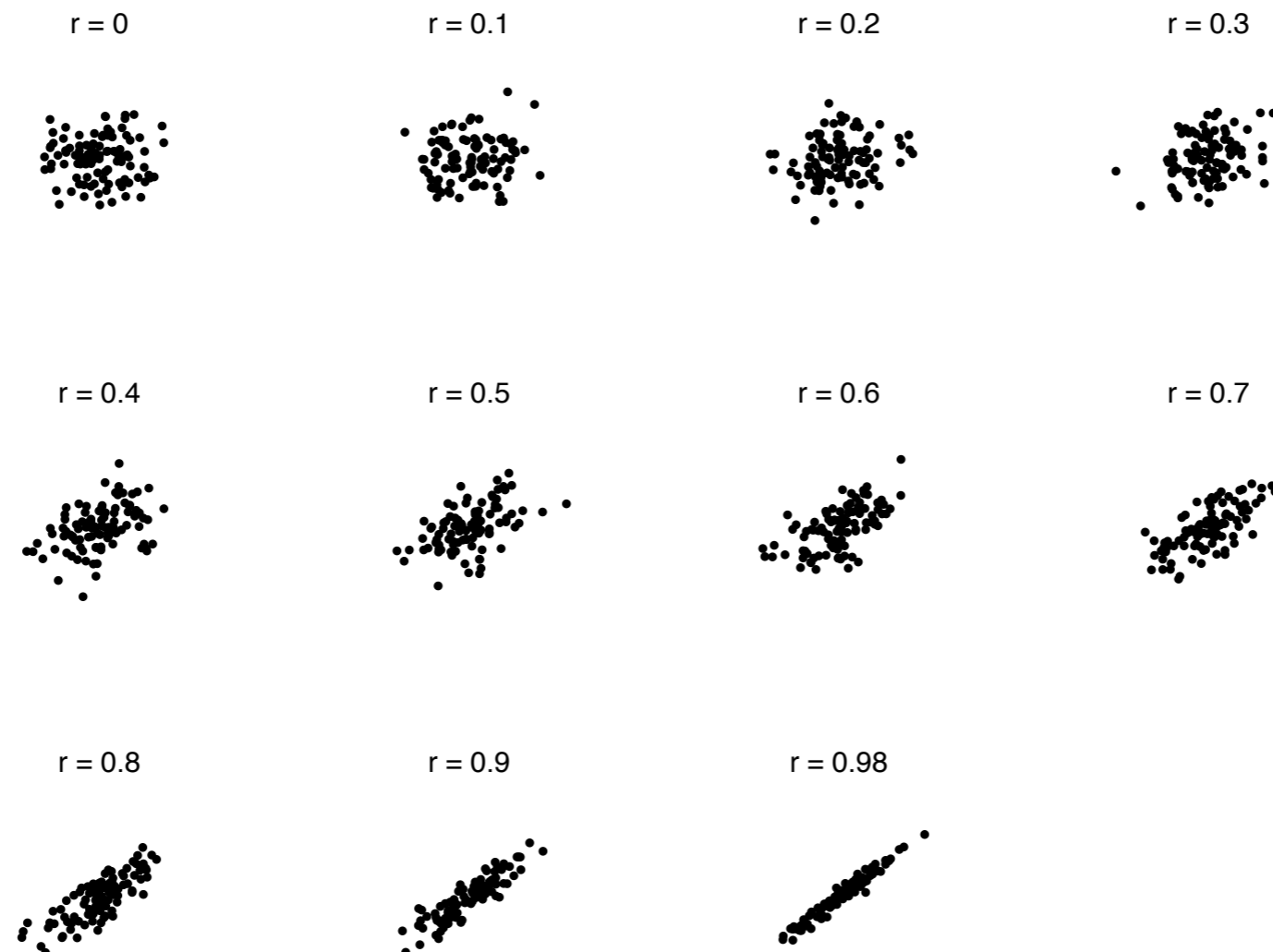
`summary`, `plot`, `predict`, `fitted`, `logLik`, `AIC` and `boot`

As a running example, we consider data sets from an experiment in which one observer judged differences in correlation between scatterplots for 11 levels of correlation:

```
c(seq(0, 0.9, len = 10), 0.98)
```

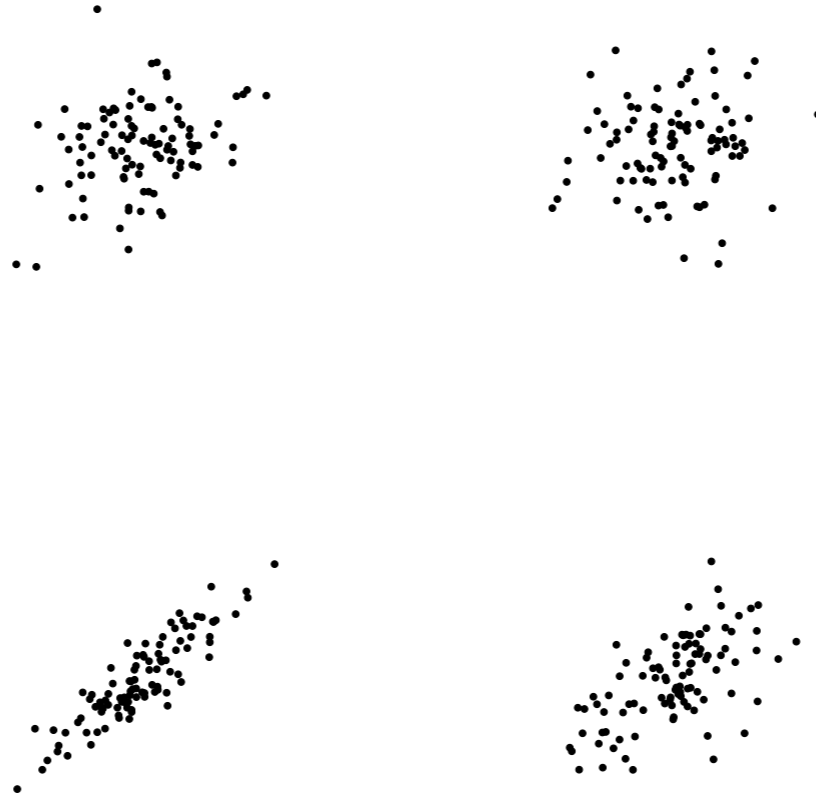
100 points generated with:

```
MASS::mvrnorm(100, mu = c(0, 0), Sigma = matrix(c(r, 0, 0, r), 2, 2))
```



The first author ran himself on 330 trials on 3 separate days, generating data sets `kk1`, `kk2`, `kk3`, available in the package. A typical trial is indicated below.

Between which pair, `lower/upper`, is the difference greatest?



```
> library(MLDS)
> data(kk1)
> data(kk2)
> data(kk3)
```

```
runSampleExperiment("DisplayOneTrial", "DefineMyScale")
```

The data sets have class 'mlds.df' that inherits from 'data.frame'. It differs in including two attributes, "stimulus" and "invord".

```
> str(kk1)
```

```
Classes 'mlds.df' and 'data.frame':  330 obs. of  5 variables:
```

```
$ resp: int  1 0 0 0 1 1 1 1 0 1 ...  
$ S1  : int  2 6 7 6 6 6 1 3 2 3 ...  
$ S2  : int  4 9 9 7 7 9 2 5 5 4 ...  
$ S3  : int  6 1 2 2 1 1 8 10 7 5 ...  
$ S4  : int  8 4 3 5 3 5 9 11 8 10 ...  
- attr(*, "invord")= logi  FALSE TRUE TRUE TRUE TRUE  
TRUE ...  
- attr(*, "stimulus")= num  0.0 0.1 0.2 0.3 0.4 ...
```

stimulus is a numeric vector of the physical stimulus levels

invord is a logical vector indicating whether on each trial the higher scale values were on the bottom or top.

It is sometimes necessary to reorder the pairs so that the higher physical values are after the lower ones.

This is conveniently done with the `SwapOrder()` function, which uses the “invord” attribute, if present.

The `Rbind()` function combines data sets *and their attributes*.

```
> kk <- SwapOrder( Rbind( kk1, kk2, kk3 ) )
```

to produce one large object.

```
> ( kk.mlds <- mlds(kk) )
```

Perceptual Scale:

0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.0000	-0.0454	0.0439	-0.0863	0.5682	1.4234	2.0695	2.6661	3.5527
0.9	0.98							
4.4297	5.5739							

sigma:

```
[1] 1
```

```
> ( kk.mlds2 <- mlds(kk, method = "optim", opt.init = c(seq(0, 1, len = 11), 0.2)) )
```

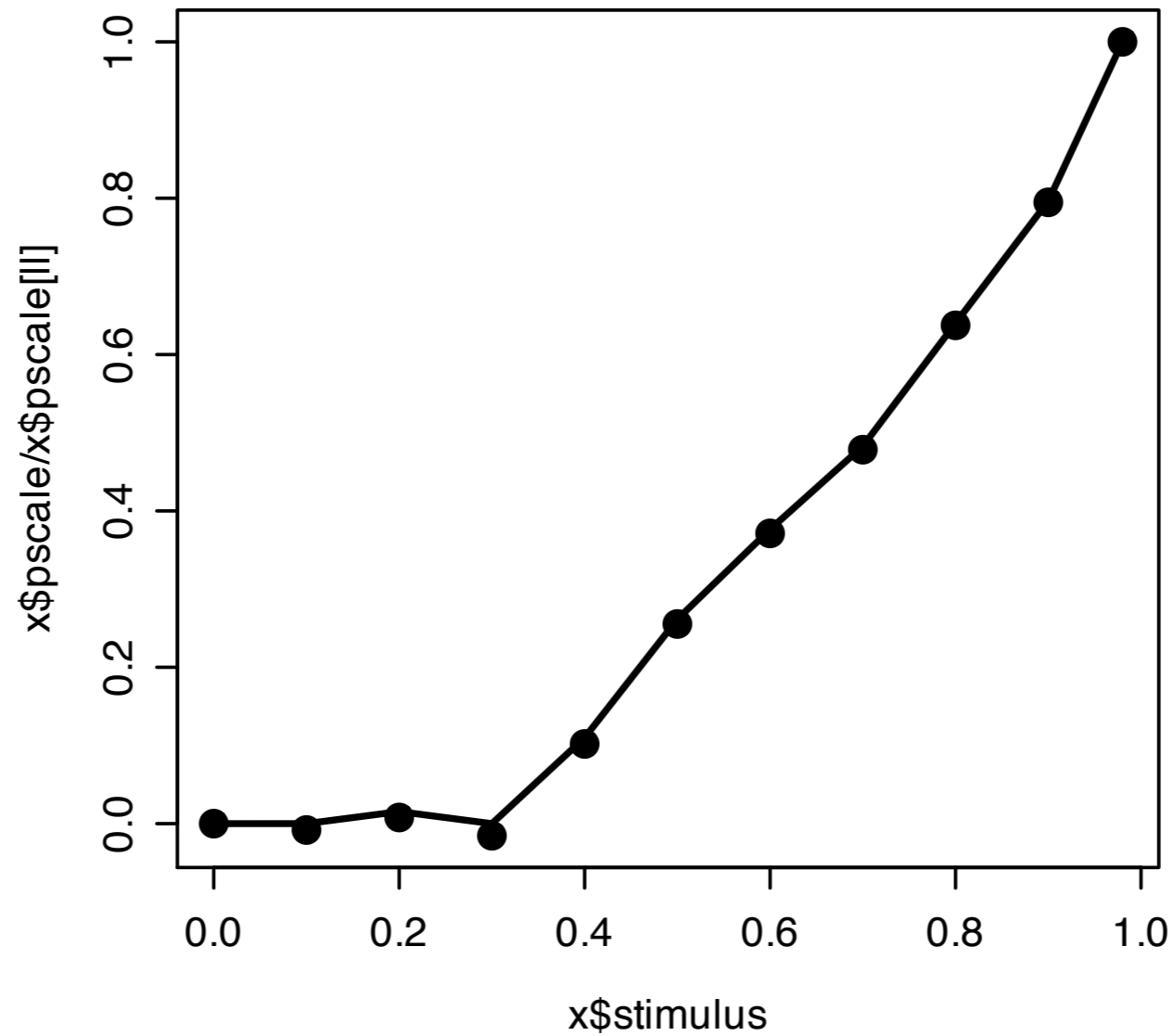
Perceptual Scale:

0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.00e+00	4.70e-05	1.54e-02	1.19e-07	1.10e-01	2.61e-01	3.76e-01	4.83e-01
0.8	0.9	0.98					
6.40e-01	7.96e-01	1.00e+00					

sigma:

```
[1] 0.175
```

- > plot(kk.mlds, standard.scale = TRUE,
 cex = 1.7, pch = 16, col = "black")
- > lines(kk.mlds2, lwd = 2)



Bootstrap Errors on mlds Scale Values

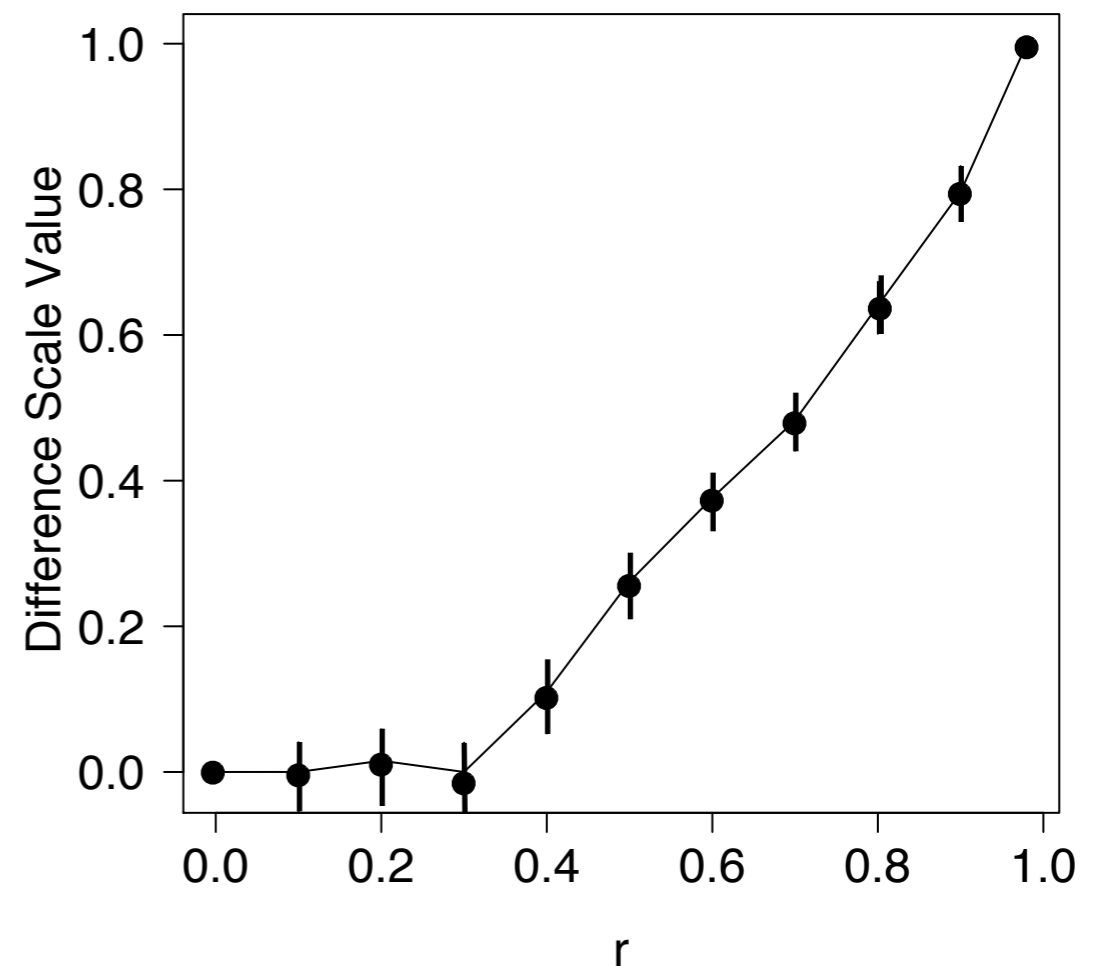
We have included a function for estimating bootstrap standard errors on the scale values

```
boot.mlds(x, nsim, ...)
```

where `x` is of class 'mlds' and `nsim` is the number of bootstrap repetitions.

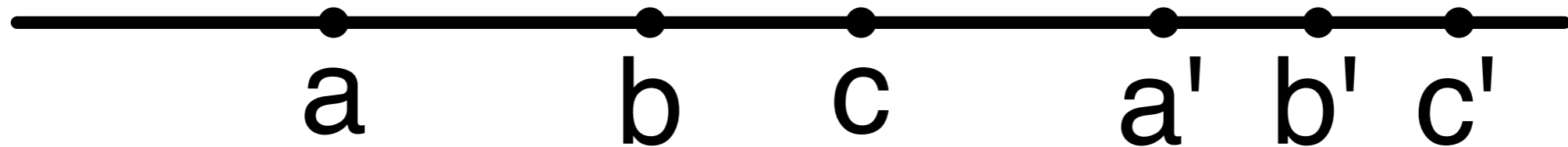
The fitted probabilities are used with `rbinom()` to generate new responses which are fitted using `mlds()` to generate new bootstrap scale values.

These bootstrap scale values are then used to generate the error bars (10 000 replications)



Scale validation

1. *Ordering property*: Observer must be able to reliably order stimuli, (I_1, I_2, \dots, I_p) , in agreement with scale values, $(\psi_1, \psi_2, \dots, \psi_p)$. (usually not formally tested and evidently satisfied).
2. *Six-point property*: Given any two groups of three intervals, (a, b, c) and (a', b', c') ,



if $ab \succ a'b'$ and $bc \succ b'c'$ then $ac \succ a'c'$.

Maloney and Yang (2003) proposed a resampling method for testing the six-point property that we have implemented in MLDS.

Scale validation: Six-point test

We have written a function to estimate the likelihood of the six-point choices from an 'mlds' object and then using the fitted probabilities to generate new responses to be fit, that permit us to compare the observed likelihood with those based on a large number of resamplings.

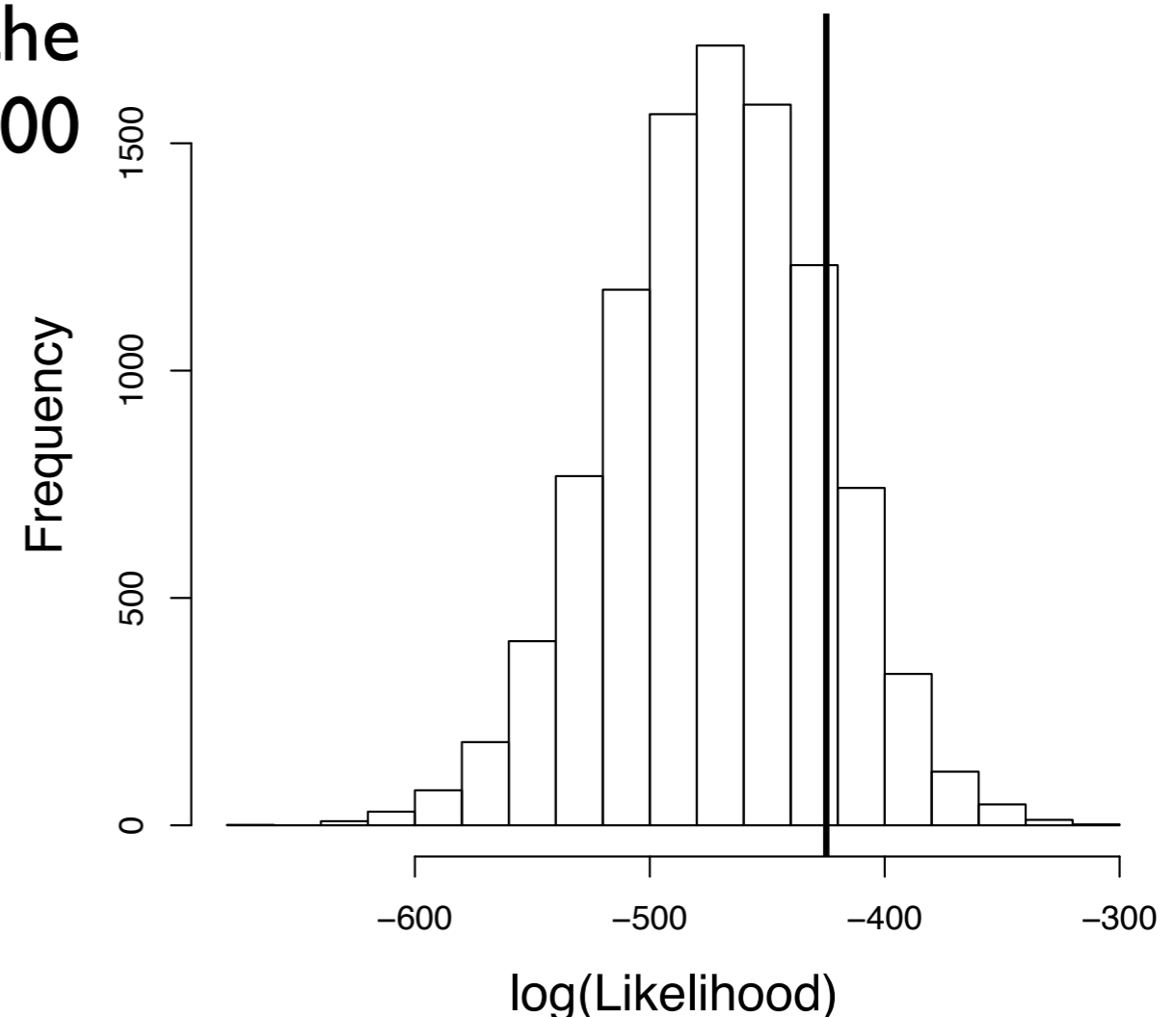
```
kk.6pt <- simu.6pt(kk.mlds, nsim = 10000)
```

returns the six-point likelihood for the observed data as well as the 10 000 resampled likelihoods.

$$L_{6pt} = -425$$

$$p = 0.85$$

Histogram of kk.6pt\$boot.samp



Future Directions

I. Formula interface:

The current fitting procedure requires the estimation of $p - 1$ parameters. The fitted scale may suggest a simpler parametric form with fewer parameters.

```
kk.fun <- mlds.function(~sx^p, p = c(4, 0.2), data = kk)
```

takes a one-sided formula with parameters, p and σ , yielding only two parameters.

```
plot(kk.mlds, standard.scale = TRUE,  
     cex = 1.7, pch = 16, col = "black")
```

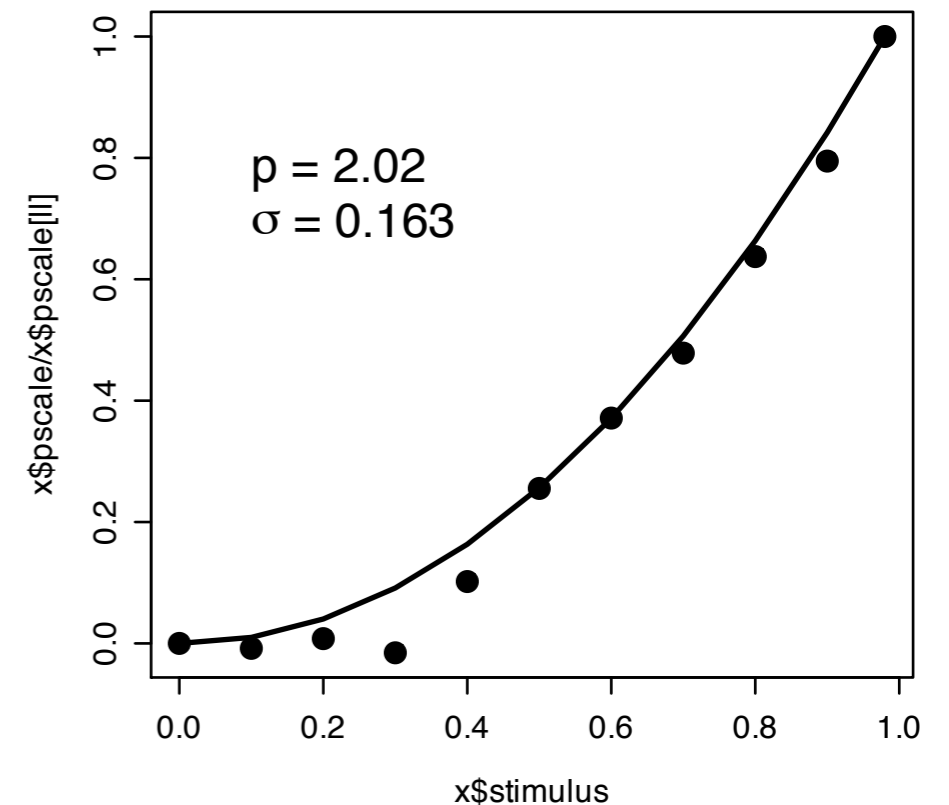
```
lines(kk.fun$stimulus, kk.fun$pscale, lwd = 2)
```

```
> -2 * kk.fun$logLik + 2 * 2
```

```
[1] 656.8796
```

```
> AIC(kk.mlds)
```

```
[1] 632.7912
```



Future Directions

Another example of the formula interface:

```
> kk.fun2 <- mlds.function(~p[1] * (sx + abs(sx - p[2])) - p[1] * p[2],  
  p = c(0.9, 0.3, 0.2), data = kk)
```

```
> kk.fun2$par
```

```
[1] 0.5136073 0.3490738
```

```
> kk.fun2$sigma
```

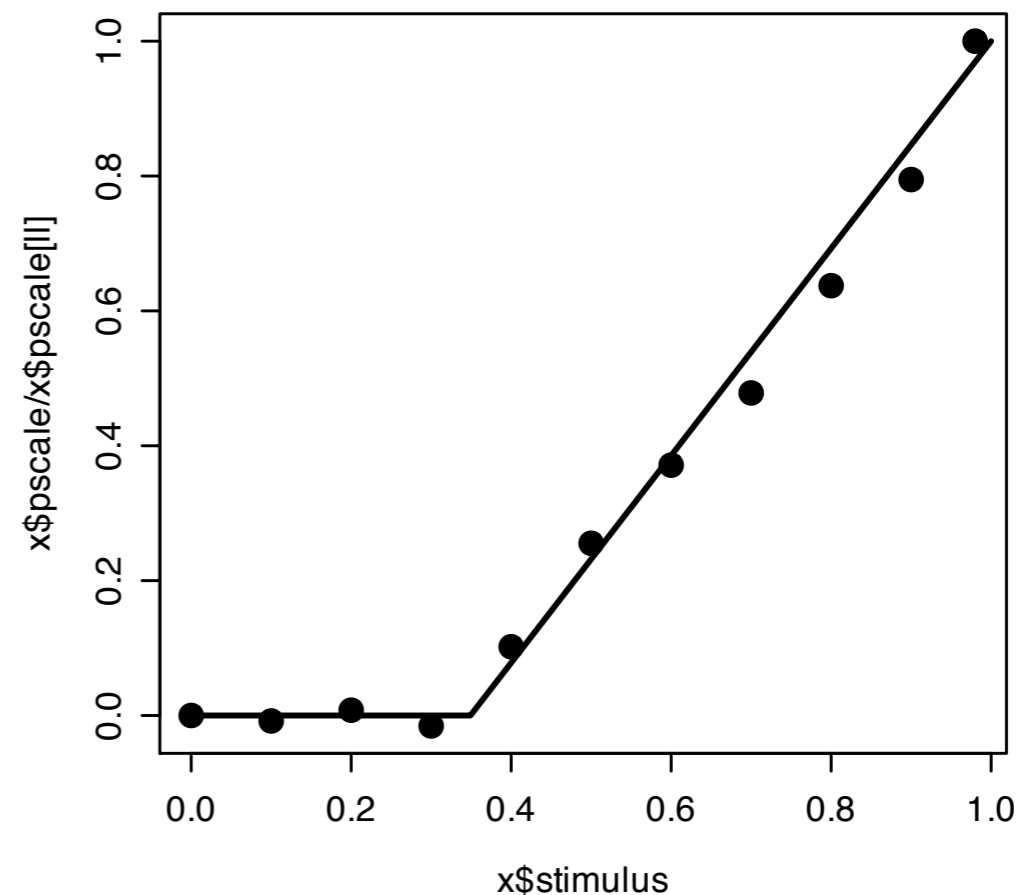
```
[1] 0.1391141
```

```
> AIC(kk.mlds)
```

```
[1] 632.7912
```

```
> -2 * kk.fun2$logLik + 3 * 2
```

```
[1] 658.6734
```



Future Directions

2. Mixed effects models:

We may want to introduce random effects to account for differences in sensitivity between runs or observers.

We have been experimenting with the lme4 package for this.

For example,

```
library(lme4)
Run <- factor(rep(paste("R", 1:3, sep = "")), each = 330))
kk.ix <- make.ix.mat(kk) #generate the data frame for glm
kk.lmer <- lmer(resp ~ . + (1 | Run) - 1, data = kk.ix,
               family = binomial("probit"))
```

> summary(kk.lmer)

Generalized linear mixed model fit using Laplace

Formula: resp ~ . + (1 | Run) - 1

Data: kk.ix

Family: binomial(probit link)

AIC BIC logLik deviance

630.8 684.7 -304.4 608.8

Random effects:

Groups Name Variance Std.Dev.

Run (Intercept) 0.13026 0.36092

number of obs: 990, groups: Run, 3

Estimated scale (compare to 1) 28.98914

Fixed effects:

Estimate Std. Error z value Pr(>|z|)

stim.2 0.05725 0.14189 0.403 0.687

stim.3 0.23181 0.15474 1.498 0.134

stim.4 0.13198 0.16748 0.788 0.431

stim.5 0.84308 0.18613 4.529 5.91e-06 ***

stim.6 1.71516 0.21032 8.155 3.49e-16 ***

stim.7 2.34077 0.23529 9.948 < 2e-16 ***

stim.8 2.87728 0.26501 10.857 < 2e-16 ***

stim.9 3.67165 0.30671 11.971 < 2e-16 ***

stim.10 4.43316 0.35794 12.385 < 2e-16 ***

stim.11 5.39385 0.43525 12.393 < 2e-16 ***

> AIC(kk.mlds)

[1] 632.7912

