# Metric/Nonmetric Sammon Mapping

Jan de Leeuw

December 5, 2025

TBD

## Table of contents

**1 Introduction** **2**

**2 Properties** **2**

**3 Algorithm** **3**

**4 Examples** **6**
4.1 Results . . . . . 6
4.2 Timing . . . . . 7
4.3 Plots . . . . . 9

**References** **13**

**Note:** This is a working manuscript which will be expanded/updated frequently. All suggestions for improvement are welcome. All Rmd, tex, html, pdf, R, and C files are in the public domain. Attribution will be appreciated, but is not required. The files can be found at https://github.com/deleeuw/sammon

# 1 Introduction

In engineering and computer science *Sammon Mapping* is a popular multidimensional scaling (MDS) method. The technique was introduced in Sammon Jr (1969). It was originally intended to map points from a higher-dimensional Euclidean space into points in a lower-dimensional Euclidean space by approximating the given higher dimensional distances by best-fitting lower dimensional ones. The *MASS* package for R implemented the *sammon()* function, which generalizes the original idea by allowing the higher-dimensional distances to be replaced by any positive symmetric matrix of dissimilarities. This was again generalized in the packages *stops* (Rusch, Mair, and Hornik (2023)) and *smacofx* (Rusch et al. (2025)) where optimization is over low-dimensional configations and over power transforms of the dissimilarities.

The Sammon loss function is

$$\sigma(X,\Delta)=\frac{1}{\sum\sum_{1\leq i<j\leq n}w_{ij}\delta_{ij}}\sum\sum_{1\leq i<j\leq n}w_{ij}\frac{(\delta_{ij}-d_{ij}(X))^2}{\delta_{ij}}. \tag{1}$$

The metric MDS problem we address in this paper is minimizing (1) over the $n\times p$ *configurations* $X$. In the non-metric case we minimize in addition over the symmetric non-negative and hollow matrices $\Delta:=\{\delta_{ij}\}$, where the $\delta_{ij}$ are monotone with the given dissimilarities.

# 2 Properties

The usual stress loss function in smacof, due originally to Kruskal (1964a) and Kruskal (1964b), is

$$\sigma(X,\Delta)=\frac{\sum\sum_{1\leq i<j\leq n}w_{ij}(\delta_{ij}-d_{ij}(X))^2}{\sum\sum_{1\leq i<j\leq n}w_{ij}\delta_{ij}^2}. \tag{2}$$

If we compare (1) and (2) we see that in Kruskal's stress the sum of the squared residuals is normalized by dividing by the sum of the squared dissimilarities. This means minimum stress is between zero and one, and it also prevents us from considering the trivial zero-residual solution $X=0$ and $\Delta=0$. In Sammon loss each residual is normalized separately, based on the idea that large dissimilarities are somehow less reliable or stable and should be downweighted.

If $\delta_{ij}$ and $d_{ij}(X)$ are close then

$$\sqrt{d_{ij}(X)}-\sqrt{\delta_{ij}}\approx\frac{1}{2}\frac{1}{\sqrt{\delta_{ij}}}(d_{ij}(X)-\delta_{ij}), \tag{3}$$

and thus

$$\sum w_{ij}\frac{(\delta_{ij}-d_{ij}(X))^2}{\delta_{ij}}\approx 4\sum w_{ij}(\sqrt{\delta_{ij}}-\sqrt{d_{ij}(X)})^2 \tag{4}$$

The loss function on the right-hand side of (4), which we will informally call *sqrt-stress*, is the special case of *rStress* with $r=\frac{1}{4}$ (De Leeuw, Groenen, and Mair (2016)).

# 3 Algorithm

In metric (ratio) MDS we minimize over $X$ for fixed $\Delta$. This is a standard metric scaling problem with weights $w_{ij}\delta_{ij}^{-1}$. We use majorization steps to decrease loss. This is identical to the ratio sammon option in smacofx. In the non-metric case we use alternating least squares, i.e. we alternate majorization steps with minimizing over $\Delta$, satisfying the ordinal constraints, for fixed $X$. As far as I know this has not been implemented before, and is somewhat non-standard. First, use the fact that $\sigma(\lambda X,\lambda\Delta)=\sigma(X,\Delta)$

$$\min_X \sigma(X,\Delta)=\min_X \sigma(X,\lambda\Delta), \tag{5}$$

i.e. the problem is homogeneous of degree zero in $\Delta$. Thus we can require without loss of generality that

$$\sum_{1\leq i<j\leq n}\sum w_{ij}\delta_{ij}=1. \tag{6}$$

With this normalization we have

$$\sigma(X,\Delta)=1-2\sum_{1\leq i<j\leq n}\sum w_{ij}d_{ij}(X)+\sum_{1\leq i<j\leq n}\sum w_{ij}\frac{d_{ij}^2(X)}{\delta_{ij}} \tag{7}$$

Thus minimizing $\omega$ for fixed $X$ over non-negative, isotone and normalized $\Delta$ means minimizing the third term on the right. And this means minimizing a separable, differentiable, and strictly convex function over a polyhedral convex set.

To simplify notation we define the extended real valued function $f$ equal to

$$f(x):=\sum_{i=1}^n w_i\frac{y_i}{x_i} \tag{8}$$

if $x_i>0$ for all $i$, and to $+\infty$ otherwise. Thus the effective domain $\text{dom}(f)$ is the interior of the positive orthant. In the context of Sammon Mapping the $y_i$ are squared distances and the $x_i$ are the transformed dissimilarities.

The *Sammon Monotone Regression* or SMR problem is defined as minimization of $f$ over the intersection of the polyhedral convex cone $\mathcal{K}$ defined by $0<x_1\leq\cdots\leq x_n$ and the affine set

$\mathcal{A}$ defined by $w'x = 1$. We assume that all $w_i$ and all $y_i$ are positive, and that the $w_i$ add up to one.

The first and second partials of $f$ are

$$\mathcal{D}_i f(x) = -w_i \frac{y_i}{x_i^2}, \tag{9}$$

$$\mathcal{D}_{ii} f(x) = 2w_i \frac{y_i}{x_i^3}, \tag{10}$$

and $\mathcal{D}_{ik} f(x) = 0$ if $i \neq k$. This shows $f$ is convex and twice differentiable on $\text{dom}(f)$, and thus on the intersection with $\mathcal{K} \cap \mathcal{A}$.

First some preliminary results.

**Lemma 3.1.**

$$\min_{x \in \mathcal{K} \cap \mathcal{A}} f(x) \leq w'y \tag{11}$$

*Proof.* The vector $x$ with $x_i = 1$ for all $i$ is in $\mathcal{K} \cap \mathcal{A}$. □

It follows from Lemma 3.1 that we can add the constraint $f(x) \leq w'y$ to the minimization problem and still have the same minimum and minimizer.

**Lemma 3.2.** *Suppose $y_1 \leq \cdots \leq y_n$. Then*

$$\overline{x}_i = \frac{\sqrt{y_i}}{\sum_{k=1}^n w_k \sqrt{y_k}}. \tag{12}$$

*Proof.* The necessary conditions for a minimum of $f$ on $\mathcal{A}$ are

$$-w_i \frac{y_i}{x_i^2} = \lambda w_i, \tag{13}$$

for all $i$, together with the side condition $w'x = 1$. From (13) it follows that the solution $\overline{x}$ must be proportional to $\sqrt{y}$, where the square root can have either sign. Of the $2^n$ solutions only one is in the effective domain of $f$, the one for which all all square roots are taken with a positive sign. This solution is also in $\mathcal{K}$. Applying the side condition gives (12). □

The next rather trivial lemma deals with the case $n = 1$, in which $w, x$ and $y$ are one-element vectors, which identify with the corresponding scalars.

**Lemma 3.3.** *If $n = 1$ then the minimizer $\overline{x}$ is equal to one and the minimum is $y$.*

*Proof.* $w$ adds up to one, so $w = 1$. Also $wx$ must be one, so $\overline{x} = 1$. □

The following theorem is of prime importance, because it shows the SMR problem can be solved with a variation of the Pool Adjacent Violaters Algorithm (PAVA). For the details on PAVA, see for example De Leeuw, Hornik, and Mair (2009).

**Theorem 3.1.** *Suppose $\overline{x}$ is the optimum solution. If $y_i \geq y_{i+1}$ then $\overline{x}_i = \overline{x}_{i+1}$.*

*Proof.* We show that $y_i \geq y_{i+1}$ and $\overline{x}_i < \overline{x}_{i+1}$ leads to a contradiction. A necessary and sufficient condition for $\overline{x}$ to be the optimum solution is

$$(x - \overline{x})' \mathcal{D} f(\overline{x}) = -\sum_{i=1}^n w_i \frac{y_i}{\overline{x}_i^2}(x_i - \overline{x}_i) = f(\overline{x}) - \sum_{i=1}^n w_i \frac{y_i}{\overline{x}_i^2} x_i \geq 0 \tag{14}$$

for all $x \in \mathcal{K} \cap \mathcal{A}$ (Hiriart-Urruty and Lemaréchal (1993), Theorem 1.1.1, page 293).

Now suppose $\overline{x}_i < \overline{x}_{i+1}$. Then for $\epsilon > 0$ small enough

$$z = \overline{x} + \epsilon \left( \frac{e_i}{w_i} - \frac{e_{i+1}}{w_{i+1}} \right) \tag{15}$$

is also in $\mathcal{K} \cap \mathcal{A}$. In (15) we add a small amount to $\overline{x}_i$ and subtract a small amount from $\overline{x}_{i+1}$, while leaving all other elements of $\overline{x}$ unperturbed. Since

$$\sum_{i=1}^n w_i \frac{y_i}{\overline{x}_i^2} z_i = f(\overline{x}) + \epsilon \left( \frac{y_i}{\overline{x}_i^2} - \frac{y_{i+1}}{\overline{x}_{i+1}^2} \right) \tag{16}$$

we must have

$$\frac{y_i}{\overline{x}_i^2} \leq \frac{y_{i+1}}{\overline{x}_{i+1}^2} \tag{17}$$

But $y_i \geq y_{i+1}$ and $\overline{x}_i < \overline{x}_{i+1}$ implies

$$\frac{y_i}{\overline{x}_i^2} > \frac{y_{i+1}}{\overline{x}_{i+1}^2} \tag{18}$$

and thus $\overline{x}$ cannot be the optimal solution. □

Theorem 3.1 can be used to replace an SMR problem of size $n$ with an SMR problem of size $n - 1$. If $y_i \geq y_{i+1}$ we remove these two $y$-values and put the single value

$$\tilde{y}_i := \frac{w_i y_i + w_{i+1} y_{i+1}}{w_i + w_{i+1}} \tag{19}$$

in their place. This new value gets the weight $\tilde{w}_i := w_i + w_{i+1}$.

**Theorem 3.2.** *Suppose $\hat{x}$ solves the monotone regression problem for the weighted least squares norm*

$$\hat{x} := \underset{x_1 \leq \cdots \leq x_n}{\text{argmin}} \ (x-y)'W(x-y) \tag{20}$$

*then*

$$\overline{x}_i = \frac{\sqrt{\hat{x}_i}}{\sum_{k=1}^n w_k \sqrt{\hat{x}_k}} \tag{21}$$

*solves the SMR problem.*

*Proof.* We use Theorem 3.1 repeatedly until we have an SMR problem of size $r$ for which $\tilde{y}_1 < \cdots < \tilde{y}_r$. We can use the same sequence of pooling adjacent violators as in least squares monotone regression, and we find the same weighted average pooled values in each step. When we have reduced the problem to a strictly increasing $\tilde{y}$ sequence we apply Lemma 3.2 (and if $r = 1$ we apply Lemma 3.3). We then expand each block again to a length equal to the number of averaged elements. □

A small example illustrates our PAVA variation. Start with $y$ equal to $(1, 2, 3, 1, 2, 3, 5, 1)$ and $w$ equal to $(1, 1, 1, 1, 1, 1, 1, 1)$. Merge elements 3 and 4 to get $y = (1, 2, 2, 2, 3, 5, 1)$ and $w = (1, 1, 2, 1, 1, 1, 1)$. Now merge elements 2, 3, and 4 to get $y = (1, 2, 3, 5, 1)$ and $w = (1, 4, 1, 1, 1)$. Merge 4 and 5 to get $y = (1, 2, 3, 3)$ and $w = (1, 4, 1, 2)$ and finally merge 3 and 4 to get $y = (1, 2, 3)$ and $w = (1, 4, 3)$. Thus

$$\overline{x} = \frac{1}{1 + 4\sqrt{2} + 3\sqrt{3}}(1, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{3}, \sqrt{3}, \sqrt{3})$$

Remember that in the context of Sammon Mapping the least squares monotone regression has to be done on the squared distances.

# 4 Examples

## 4.1 Results

Numerical Sammon mapping takes 21 iterations to arrive at stress 0.0222606801. For ordinal we need 31 iterations for stress 0.0006660664. Since the ordinal fit is very good the sqrt-stress of (4) should be close to Sammon stress. It is 0.0006701850.

For the Morse code data numerical Sammon mapping takes 99 iterations to arrive at stress 0.0977737, for ordinal we need 92 iterations for stress 0.0398178. Because the fit is much worse than in the Ekman case we do not expect sqrt-stress to be close to Sammon stress. For the ordinal case final sqrt-stress is 0.0445148.

## 4.2 Timing

The results of a comparison of the R and C implementations of Sammon mapping with microbenchmark (Mersmann (2024)) yields the times in microseconds in the following table.

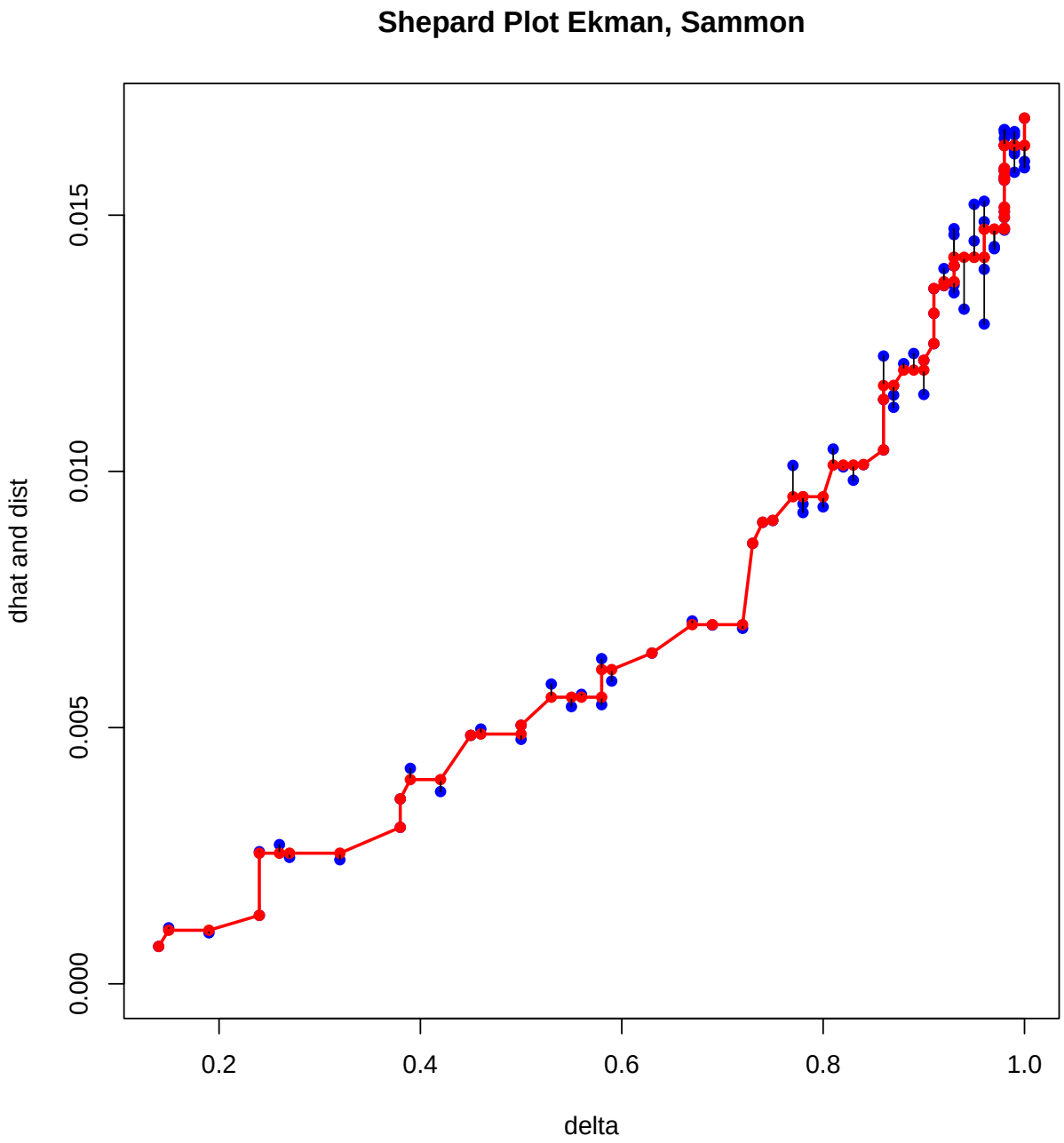
```
Unit: microseconds
                                       expr       min         lq       mean
 smacofSSSammonR(ekmanData, ordinal = FALSE)  1624.584  1694.4685  1958.5417
 smacofSSSammonC(ekmanData, ordinal = FALSE)   298.193   318.1600   338.2229
  smacofSSSammonR(ekmanData, ordinal = TRUE) 10767.953 11144.2510 12111.8748
  smacofSSSammonC(ekmanData, ordinal = TRUE)   525.989   567.1325   672.8375
        smacofSS(ekmanData, ordinal = FALSE)   259.940   276.7295   289.6978
         smacofSS(ekmanData, ordinal = TRUE)   967.764  1037.9355  1142.9222
    median         uq       max neval
  1733.8490  1765.2140 20538.007   100
   332.2025   352.7230   433.083   100
 11342.2810 11505.2150 22628.966   100
   575.9680   590.6050  5144.065   100
   286.2620   298.7055   357.479   100
  1055.1760  1074.0155  9417.905   100

Unit: milliseconds
                                       expr        min         lq       mean
 smacofSSSammonR(morseData, ordinal = FALSE)  38.153288  40.545412  44.831142
 smacofSSSammonC(morseData, ordinal = FALSE)   3.619603   3.844590   3.914574
  smacofSSSammonR(morseData, ordinal = TRUE) 425.866549 447.049281 463.749984
  smacofSSSammonC(morseData, ordinal = TRUE)   4.882649   5.214195   5.317852
        smacofSS(morseData, ordinal = FALSE)   1.564027   1.660111   1.742756
         smacofSS(morseData, ordinal = TRUE)   2.579187   2.700527   2.865947
    median         uq        max neval
  42.262861  43.966227 109.556059   100
   3.901888   3.954204   5.391992   100
 455.434867 467.828306 577.709598   100
   5.331250   5.386026   6.830272   100
   1.683194   1.754390   5.064894   100
   2.770821   2.851673   6.140857   100

Unit: microseconds
                                     expr       min         lq       mean
 smacofSSSammonR(wishData, ordinal = FALSE)  5568.415  5754.2680  6031.8761
```

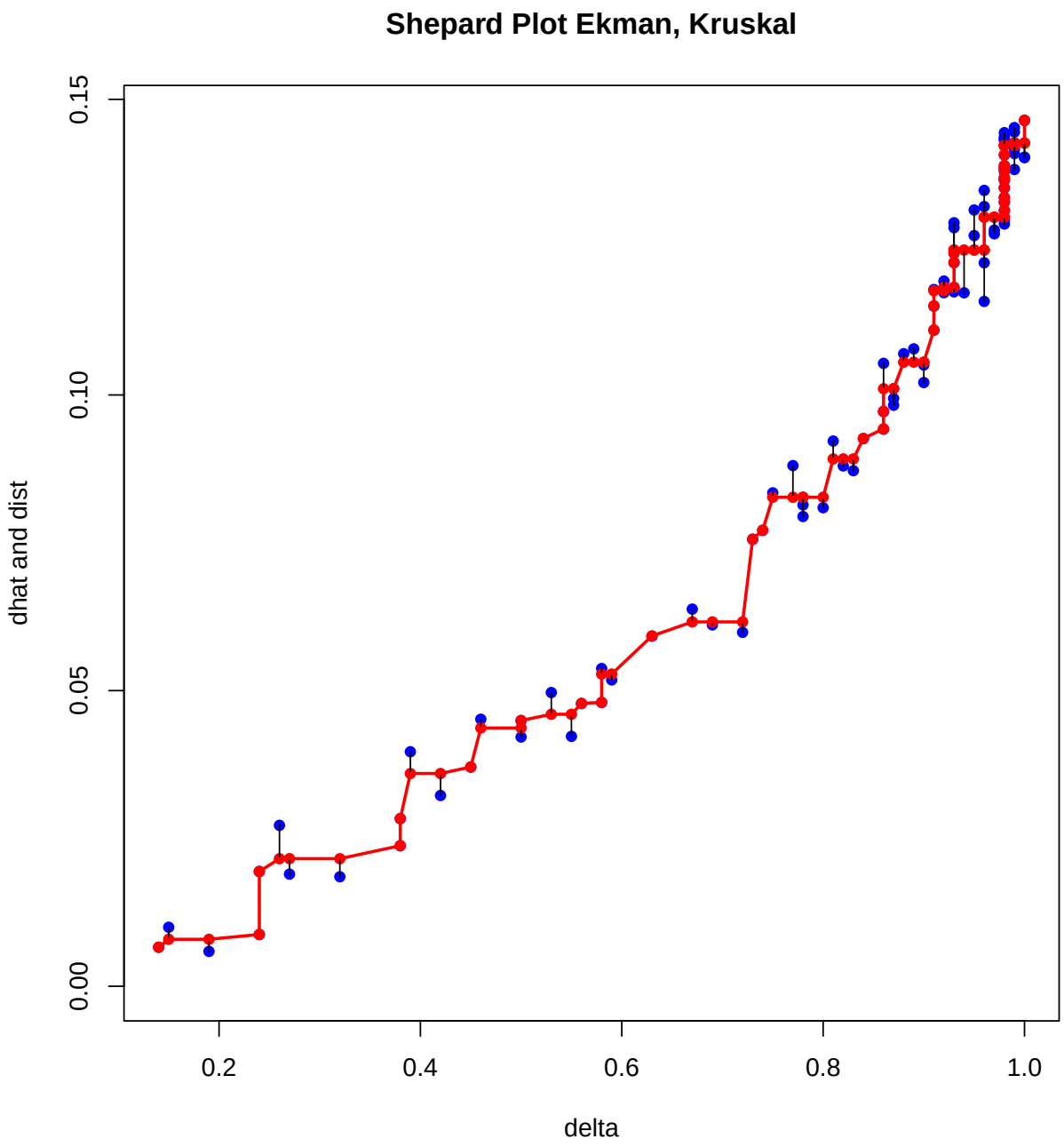
```
 smacofSSSammonC(wishData, ordinal = FALSE)   357.110   386.2405   441.1522
  smacofSSSammonR(wishData, ordinal = TRUE) 15313.254 15889.4475 16975.3760
  smacofSSSammonC(wishData, ordinal = TRUE)   591.302   617.8700   633.4627
        smacofSS(wishData, ordinal = FALSE)   216.275   233.2285   288.7671
         smacofSS(wishData, ordinal = TRUE)  1000.769  1090.3950  1138.7574
    median         uq       max neval
 5875.1360  5975.7500  9804.535   100
  394.5225   423.2020  3961.461   100
16095.9235 18080.9180 27165.411   100
  626.8080   643.2900   743.822   100
  240.2190   255.9425  4485.113   100
 1106.9180  1123.3795  4550.016   100
```

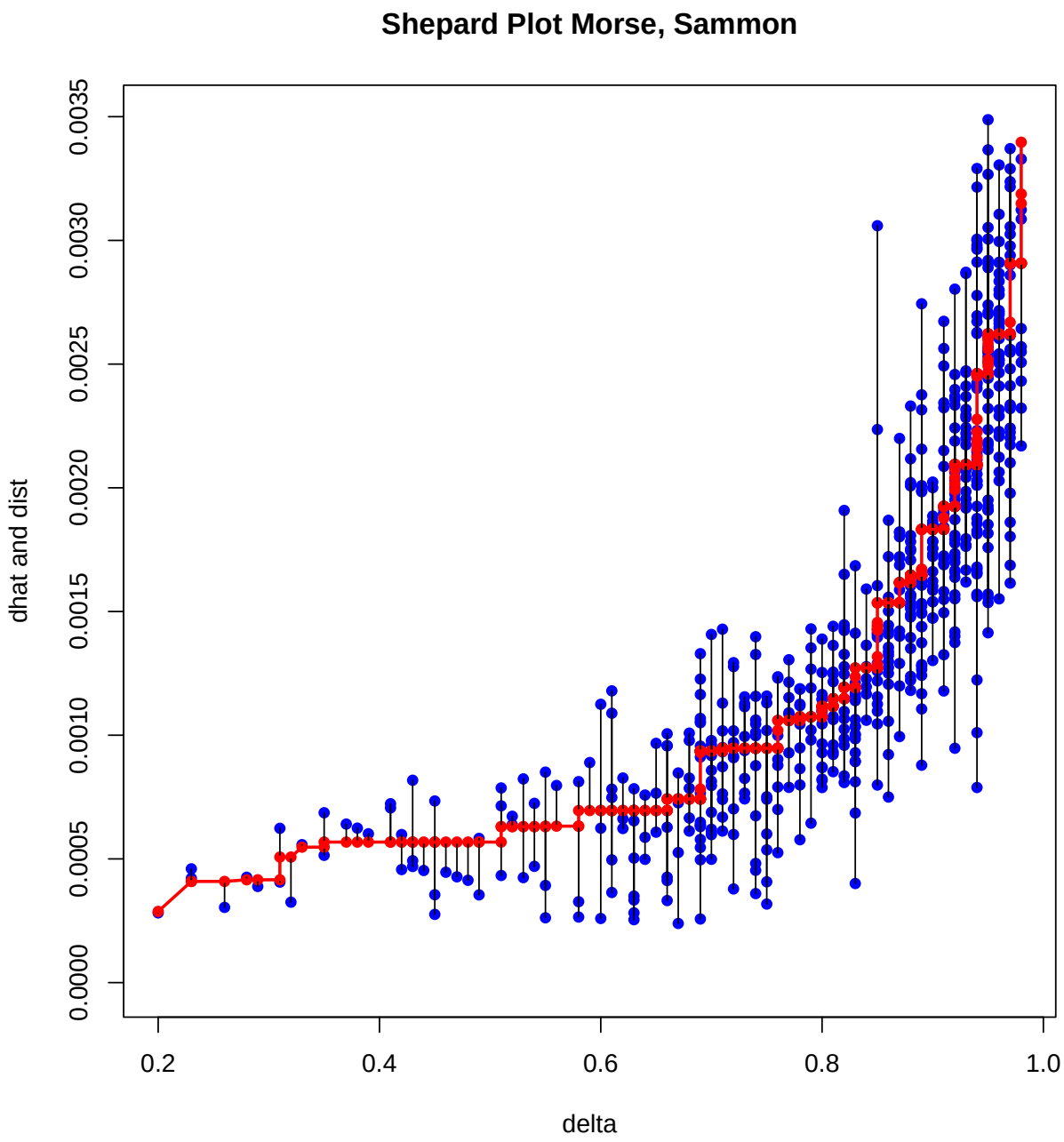
```
Unit: microseconds
                                            expr      min        lq      mean
 smacofSSSammonR(gruijterData, ordinal = FALSE) 8908.193 9296.1760 9823.1437
 smacofSSSammonC(gruijterData, ordinal = FALSE)  334.314  359.2830  369.9188
  smacofSSSammonR(gruijterData, ordinal = TRUE) 8032.146 8218.4295 8853.6597
  smacofSSSammonC(gruijterData, ordinal = TRUE)  412.255  428.9830  448.2731
        smacofSS(gruijterData, ordinal = FALSE)  157.645  166.8290  179.2746
         smacofSS(gruijterData, ordinal = TRUE)  474.616  508.4205  523.1342
   median        uq       max neval
 9493.6525 9695.5160 13150.176   100
  366.7245  378.4505   423.448   100
 8371.9745 8537.5325 21884.611   100
  437.9415  459.9585   583.266   100
  171.9950  185.2380   279.866   100
  517.0715  532.7335   649.317   100
```

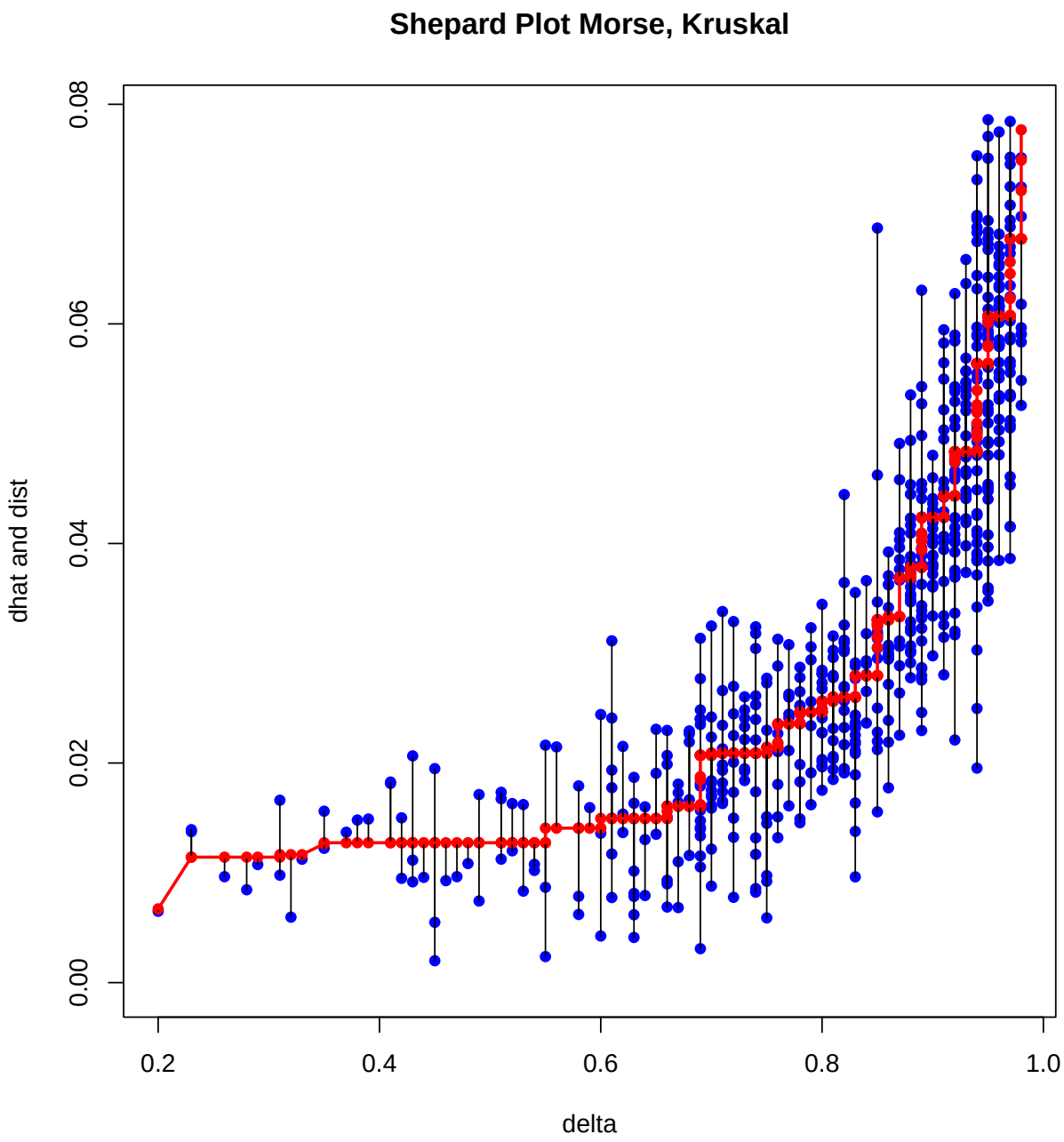
Of course these microbenchmark results depend on the default parameters of the programs (same in R and C), on the speed of my computer (Mac Mini with Apple M4 Pro, 64 GB of RAM, Tahoe 26.2, Apple clang 17.0.0), and on my programming habits (same in R and C).

## 4.3 Plots


Shepard Plot Ekman, Sammon
dhat and dist
0.000
0.005
0.010
0.015
0.2
0.4
0.6
0.8
1.0
delta


Shepard Plot Ekman, Kruskal
dhat and dist
0.00
0.05
0.10
0.15
0.2
0.4
0.6
0.8
1.0
delta


Shepard Plot Morse, Sammon
dhat and dist
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
0.2
0.4
0.6
0.8
1.0
delta


Shepard Plot Morse, Kruskal
dhat and dist
0.00
0.02
0.04
0.06
0.08
0.2
0.4
0.6
0.8
1.0
delta

# References


De Leeuw, Jan, Patrick Groenen, and Patrick Mair. 2016. "Minimizing rStress Using Majorization." 2016. https://jansweb.netlify.app/publication/deleeuw-groenen-mair-e-16-a/deleeuw-groenen-mair-e-16-a.pdf.

De Leeuw, Jan, Kurt Hornik, and Patrick Mair. 2009. "Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods." *Journal of Statistical Software* 32 (5): 1–24.

Hiriart-Urruty, Jean-Baptiste, and Claude Lemaréchal. 1993. *Convex Analysiis and Minimization Algorithms Volume 1: Fundamentals*. Springer.

Kruskal, J. B. 1964a. "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis." *Psychometrika* 29: 1–27.

———. 1964b. "Nonmetric Multidimensional Scaling: a Numerical Method." *Psychometrika* 29: 115–29.

Mersmann, O. 2024. *microbenchmark: Accurate Timing Functions*. https://CRAN.R-project.org/package=microbenchmark.

Rusch, Thomas, Jan de Leeuw, Lisha Chen, and Patrick Mair. 2025. *Smacofx: Flexible Multidimensional Scaling and 'Smacof' Extensions*. https://doi.org/10.32614/CRAN.package.smacofx.

Rusch, Thomas, Patrick Mair, and Kurt Hornik. 2023. "Structure-Based Hyperparameter Selection with Bayesian Optimization in Multidimensional Scaling." *Statistics and Computing* 33 (28): 1–18.

Sammon Jr, John W. 1969. "A Nonlinear Mapping for Data Structure Analysis." *IEEE Transacttions on Computers* C-18 (5).