## Non-Metric Multidimensional Scaling 59/86

NMDS find a low-dimensional mapping that preserves as best as possible the rank order of the original dissimilarities dij

Solution with minimal stress is sought; a measure of how well the NMDS mapping fits the dij

Stress is sum of squared residuals of monotonic regression between distances in NMDS space, d∗ij & dij

Non-linear regression can cope with non-linear responses in species data

Iterative solution; *convergence is not guaranteed*

Must solve separately different dimensionality solutions

## Non-Metric Multidimensional Scaling 60 / 86

* Use an appropriate dissimilarity metric that gives good gradient separation rankindex(), e.g Bray-Curtis, Jaccard, Kulczynski
* Wisconsin transformation useful; Standardize species to equal maxima, then sites to equal totals wisconsin()
* Use many random starts and look at the fits with lowest stress (try & trymax)
* Only conclude solution reached if lowest stress solutions are similar (Procrustes rotation)
* Fit NMDS for 1, 2, 3, … dimensions; stop after a sudden drop in stress observed in a screeplot
* NMDS solutions can be rotated ; common to rotate to PCs
* Also scale axes in half-change units

## NMDS in vegan 61 / 86

**vegan** implements all these ideas via the metaMDS() wrapper

data(dune)

set.seed(10)

(sol <- metaMDS(dune, trace = FALSE))

##

## Call:

## metaMDS(comm = dune, trace = FALSE)

##

## global Multidimensional Scaling using monoMDS

##

## Data: dune

## Distance: bray

##

## Dimensions: 2

## Stress: 0.1183186

## Stress type 1, weak ties

## Two convergent solutions found after 20 tries

## Scaling: centring, PC rotation, halfchange scaling

## Species: expanded scores based on 'dune'

## NMDS in vegan 62 / 86

If no convergent solutions, continue iterations from previous best solution

(sol <- metaMDS(dune, previous.best = sol, trace = FALSE))

##

## Call:

## metaMDS(comm = dune, trace = FALSE, previous.best = sol)

##

## global Multidimensional Scaling using monoMDS

##

## Data: dune

## Distance: bray

##

## Dimensions: 2

## Stress: 0.1183186

## Stress type 1, weak ties

## Two convergent solutions found after 40 tries

## Scaling: centring, PC rotation, halfchange scaling

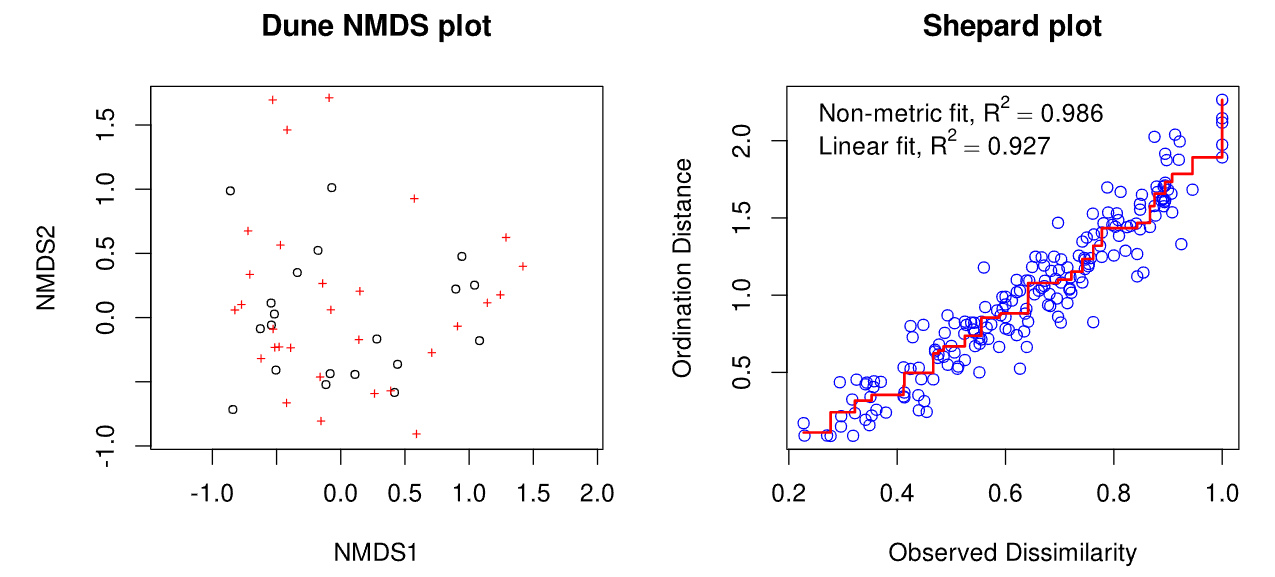
## Species: expanded scores based on 'dune'

## NMDS in vegan 63 / 86

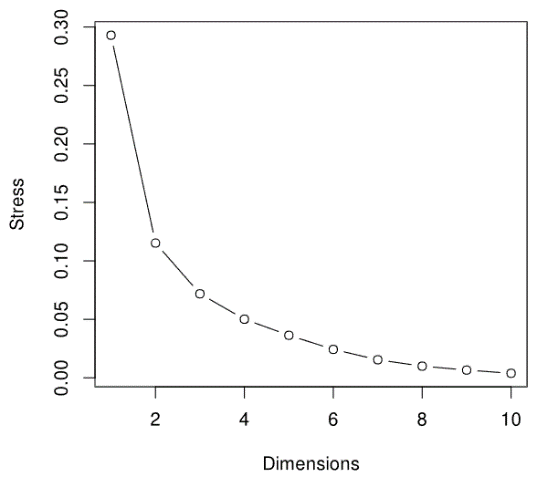
layout(matrix(1:2, ncol = 2))

plot(sol, main = "Dune NMDS plot"); stressplot(sol, main = "Shepard plot")

layout(1)



## Checking dimensionality k 64 / 86

Fit NMDS solutions for a number of k

k\_vec <- 1:10

stress <- numeric(length(k\_vec))

dune\_dij <- metaMDSdist(dune, trace = FALSE)

set.seed(25)

**for**(i **in** seq\_along(k\_vec)) {

sol <- metaMDSiter(dune\_dij, k = i,

trace = FALSE)

stress[i] <- sol$stress

}

Need to use the helper functions to do this right

plot(k\_vec, stress, type = "b", ylab = "Stress",

xlab = "Dimensions")