Practical 04 SG: Population substructure

Quim Aguado, Marcel Cases

13-Dec-2021

Ex 1 The file *SNPChr20.rda* contains genotype information of a set of individuals of unknown background. Load this data into the R environment with the load instruction. The first six columns of the data matrix contain identifiers, sex and phenotype and are not needed. The remaining columns contain the allele counts (0, 1 or 2) for over 138.000 SNPs for one of the alleles of each SNP.

```
load(url("http://www-eio.upc.es/~jan/data/bsg/SNPChr20.rda"))
#Y <- Y[,7:ncol(Y)] # where are the first 6 columns as stated¿??¿¿¿?
n <- nrow(Y) # number of individuals
p <- ncol(Y) # number of variants
```

Ex 2 Compute the Manhattan distance matrix between the individuals (this may take a few minutes) using R function dist. Include a submatrix of dimension 5 by 5 with the distances between the first 5 individuals in your report.

```
Dobs <- as.matrix(dist(Y, method = "manhattan"))
Dobs[1:5,1:5]
```

```
##       1     2     3     4     5
## 1     0 18846 19639 21485 19778
## 2 18846     0 20089 20969 21680
## 3 19639 20089     0 21230 20089
## 4 21485 20969 21230     0 20577
## 5 19778 21680 20089 20577     0
```

Ex 3 The Manhattan distance (also known as the taxicab metric) is identical to the Minkowsky distance with parameter lambda = 1. How does the Manhattan distance relate to the allele sharing distance, where the latter is calculated as two minus the number of shared alleles?

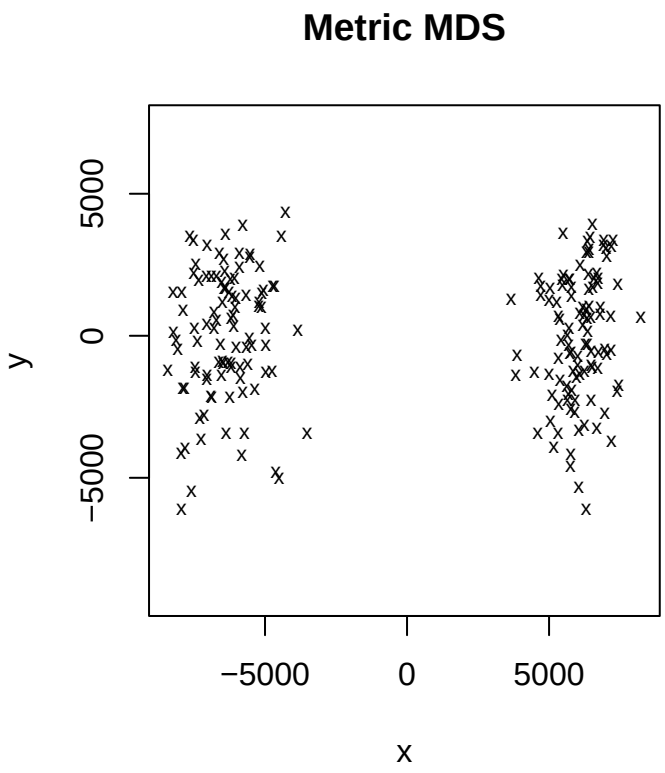
```
as.matrix(dist(Y[1:5,], method = "minkowski", p=1))
```

```
##       1     2     3     4     5
## 1     0 18846 19639 21485 19778
## 2 18846     0 20089 20969 21680
## 3 19639 20089     0 21230 20089
## 4 21485 20969 21230     0 20577
## 5 19778 21680 20089 20577     0
```

Manhattan distance in fact calculates the number of different alleles between two polymorphisms, and ASD does the same from the encoded version (0,1,2) by extracting the difference.

Ex 4 Apply metric multidimensional scaling using the Manhattan distance matrix to obtain a map of the individuals, and include your map in your report. Do you think the data come from one homogeneous human population? If not, how many subpopulations do you think the data might come from, and how many individuals pertain to each suppopulation?

```
mds.fit <- cmdscale(Dobs,eig=TRUE,k=n-27) # k is the max dimension
X <- mds.fit$points[,1:2]
par(pty="s") # square
plot(X[,1],X[,2],xlab="x",ylab="y",main="Metric MDS",type="n",asp=1)
text(X[,1],X[,2],labels="x",cex=.7)
```



There are clearly two different groups of individuals in the dataset (two differentiated clusters).

```
length(mds.fit$points[,1][mds.fit$points[,1]>=0])
```

```
## [1] 104
```

```
length(mds.fit$points[,1][mds.fit$points[,1]<0])
```

```
## [1] 99
```

There are 104 individuals in one subpopulation and 99 in the other.

Ex 5 Report the first 10 eigenvalues of the solution.

```
attributes(mds.fit)
```

```
## $names
## [1] "points" "eig"    "x"      "ac"     "GOF"
```

```
mds.fit$eig[1:10]
```

```
##  [1] 7975174146 1022552407 1007212589  915276321  805632194  788372113
##  [7]  763093810  745029987  724989450  690613576
```

Ex 6 Does a perfect representation of this nxn distance matrix exist, in n or fewer dimensions? Why so or not?

Eigenvalues represent variance of the polymorphisms, and we can represent an nxn distance matrix using n values of obtained eigenvalues.

Ex 7 What is the goodness-of-fit of a two-dimensional approximation to your distance matrix? Explain which criterium you have used.

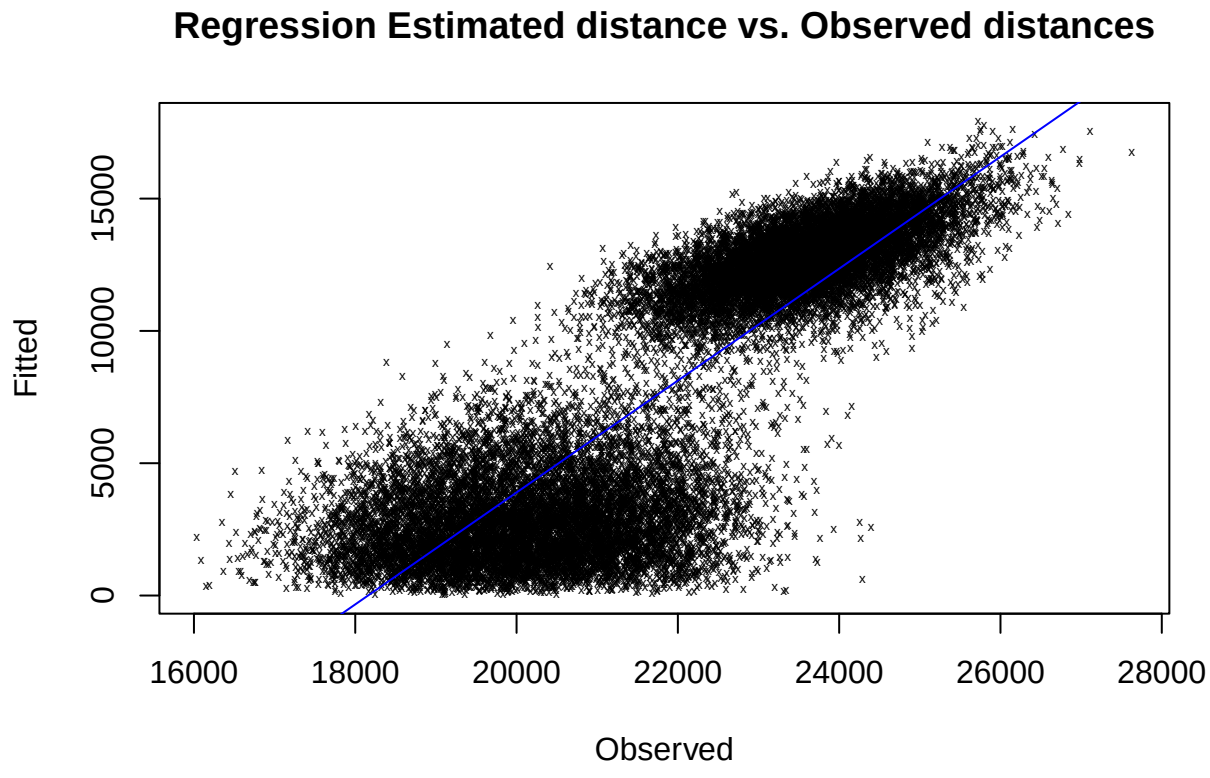
We use the absolute value of lambda to calculate GOF:

```
mds.fit$GOF
```

```
## [1] 0.9875065 1.0000000
```

Ex 8 Make a plot of the estimated distances (according to your map of individuals) versus the observed distances. What do you observe? Regress estimated distances on observed distances and report the coefficient of determination of the regression.

```
#X <- t(Y)
Dest <- as.matrix(dist(X))
Dobs.vec <- Dobs[lower.tri(Dobs)]
Dest.vec <- Dest[lower.tri(Dest)]
plot(Dobs.vec,Dest.vec,xlab="Observed",ylab="Fitted",col="white",main="Regression Estimated distance vs
text(Dobs.vec,Dest.vec,labels="x",cex=.4)
abline(lm(Dest.vec ~ Dobs.vec), col = "blue")
```




```
cor(Dobs.vec,Dest.vec)
```

```
## [1] 0.85579
```

The estimated distances vs. the observed distances from the dataset have a correlation, although not a strong one. We observe how each one of the two subpopulations has a different area in the regression plot.

Ex 9 We now try non-metric multidimensional scaling using the isoMDs instruction. We use a random initial configuration. Make a plot of the two-dimensional solution. Do the results support that the data come from one homogeneous population? Try different runs. What do you observe?

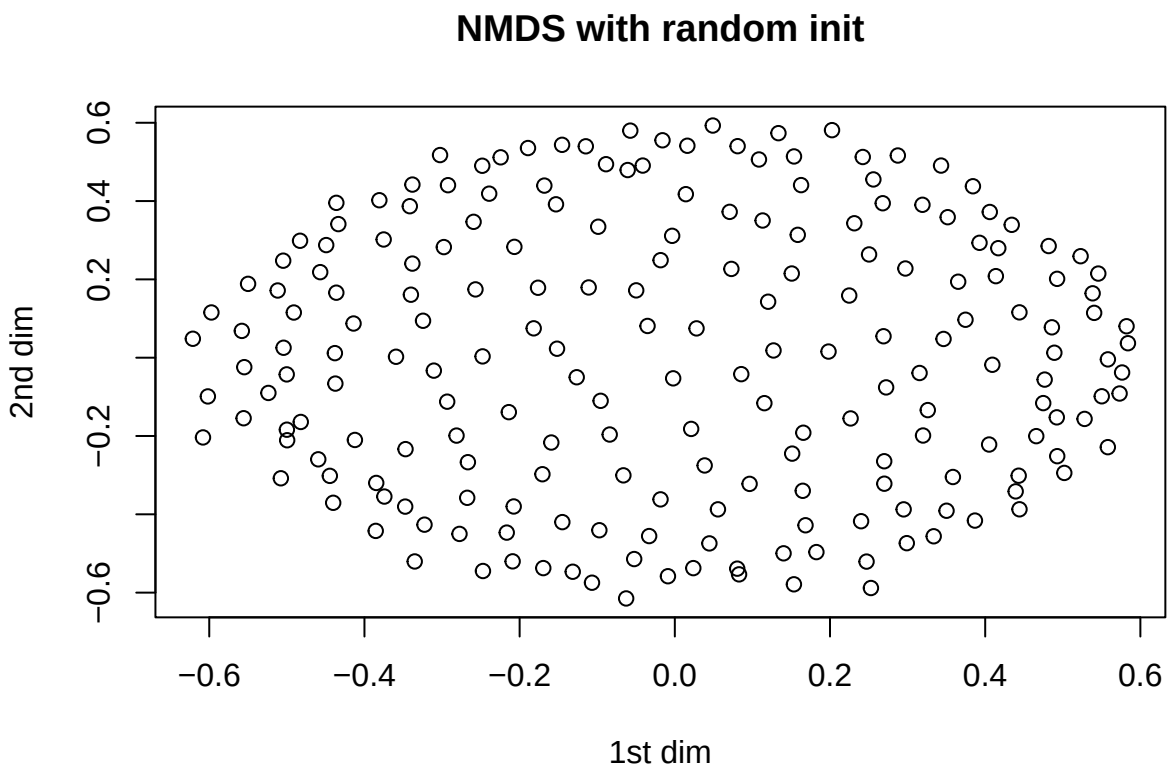
```
set.seed(12345)
k <- 2
init <- scale(matrix(runif(k*n),ncol=k),scale=FALSE)
nmds.fit <- isoMDS(Dobs,k=k,y=init)
```

```
## initial  value 42.999258
## iter   5 value 41.650278
## iter   5 value 41.609917
## iter   5 value 41.609182
## final  value 41.609182
## converged
```

```
nmds.fit$stress
```

```
## [1] 41.60918
```

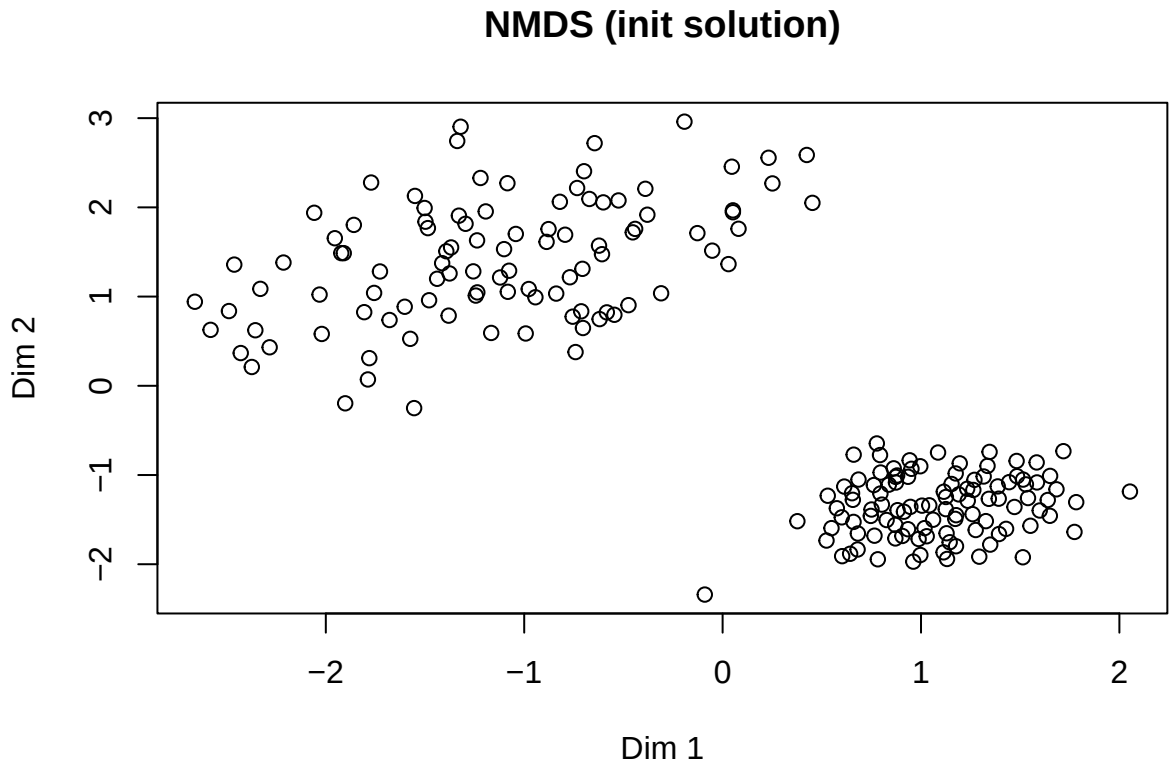
```
Ynmds <- nmds.fit$points
plot(Ynmds, main = "NMDS with random init", xlab = "1st dim", ylab = "2nd dim")
```



Ex 10 Set the seed of the random number generator to 123. Then run isoMDS a hundred times, each time using a different random initial configuration using the instructions above. Save the final stress-value and the coordinates of each run. Report the stress of the best run, and plot the corresponding map.

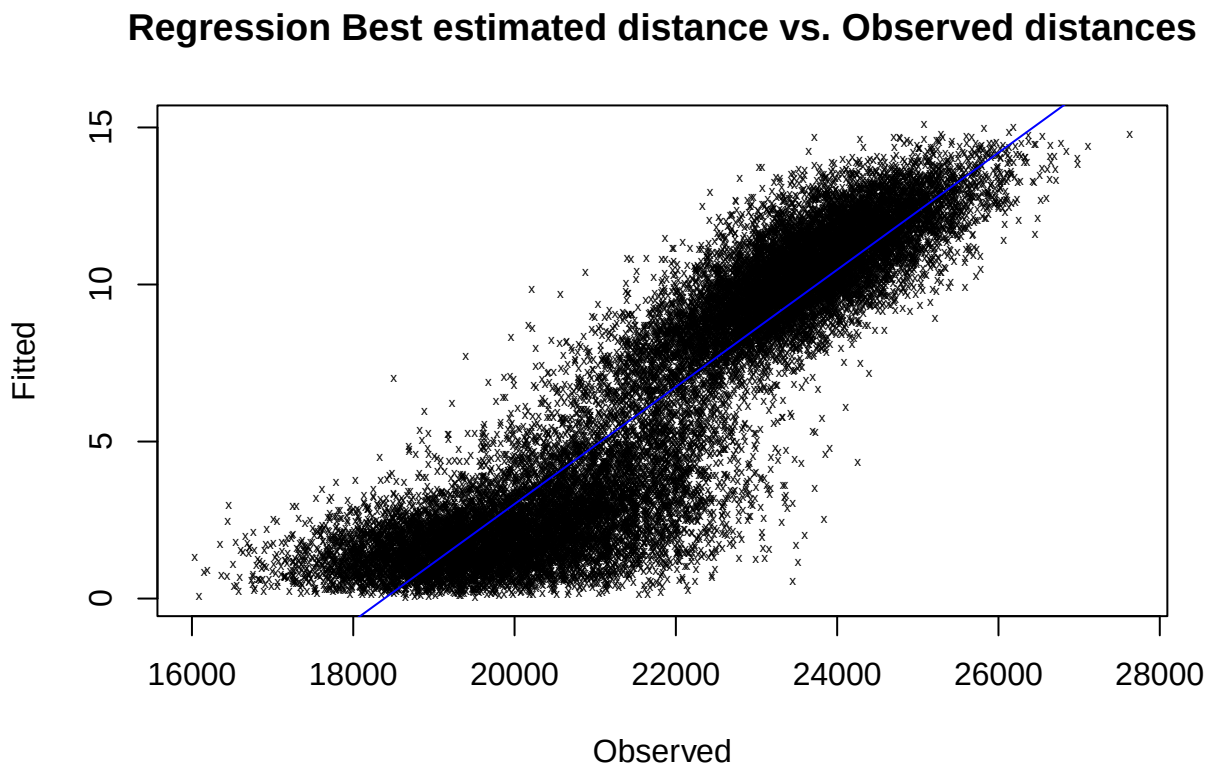
```
set.seed(123)
stress.11 = c()
points.11 = list()
for(i in 1:100){
  init <- scale(matrix(runif(2*n),ncol=2),scale=FALSE)
  nmds.fit<-isoMDS(d=Dobs,y=init,k=2,trace=FALSE)
  stress.11 = append(stress.11,nmds.fit$stress)
  points.11[[i]] = nmds.fit$points
}

plot(points.11[[which.min(stress.11)]], main="NMDS (init solution)", xlab = "Dim 1", ylab = "Dim 2")
```



Ex 11 Make again a plot of the estimated distances (according to your map of individuals of the best run) versus the observed distances, now for the two-dimensional solution of non-metric MDS. Regress estimated distances on observed distances and report the coefficient of determination of the regression.

```
Dest.best <- as.matrix(dist(points.11[[which.min(stress.11)]]))
Dest.best.vec <- Dest.best[lower.tri(Dest.best)]
data1<-data.frame(obs=Dest.best.vec, exp=Dobs.vec)
R = cor(Dobs.vec,Dest.best.vec)
plot(Dobs.vec,Dest.best.vec,xlab="Observed",ylab="Fitted",col="white", main="Regression Best estimated
text(Dobs.vec,Dest.best.vec,labels="x",cex=.4)
abline(lm(Dest.best.vec ~ Dobs.vec), col = "blue")
```

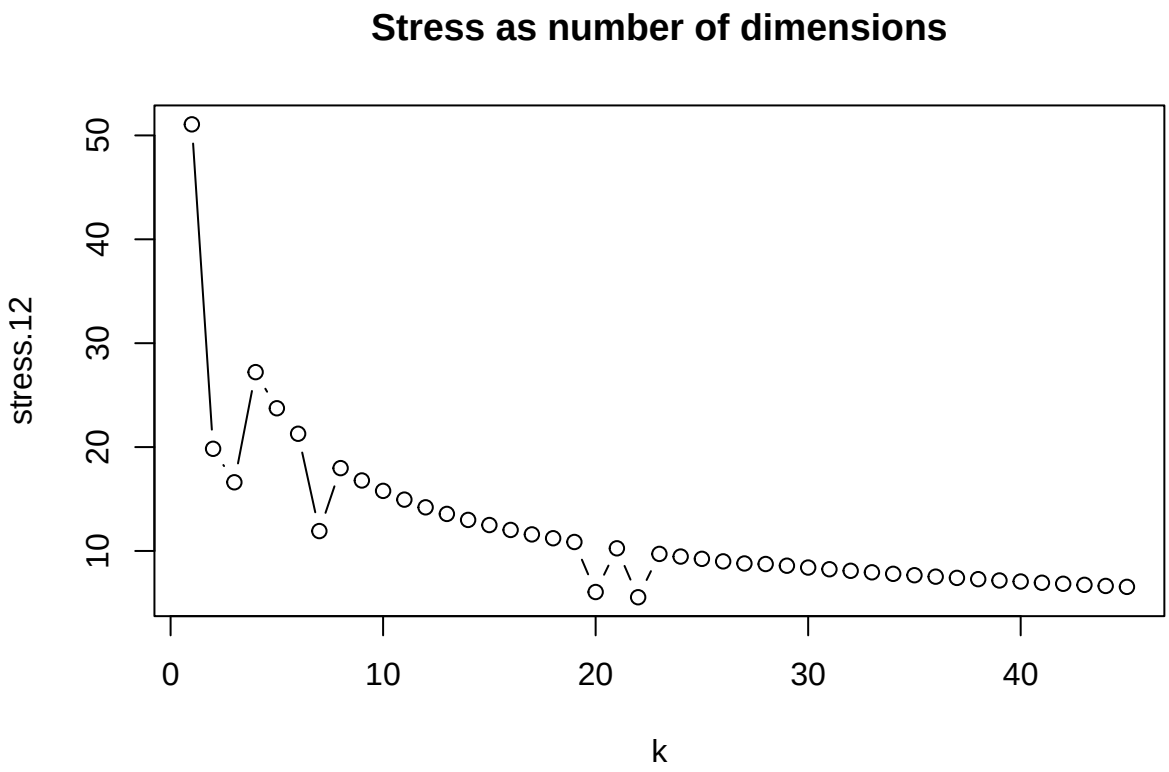



```
cor(Dobs.vec,Dest.best.vec)
```

```
## [1] 0.9101253
```

Ex 12 Compute the stress for a 1, 2, 3, 4, . . . , n-dimensional solution, always using the classical MDS solution as an initial configuration. How many dimensions are necessary to obtain a good representation with a stress below 5? Make a plot of the stress against the number of dimensions

```
set.seed(123)
stress.12 <- c()
coord.12 <- c()
for(k in 1:45) {
  init <- scale(matrix(runif(k*n),ncol=k),scale=FALSE)
  nmds.fit <- isoMDS(Dobs,k=k,y=init,trace=FALSE)
  stress.12 <- c(stress.12,nmds.fit$stress)
}
plot(1:length(stress.12),stress.12,type="b",xlab="k", main="Stress as number of dimensions")
```



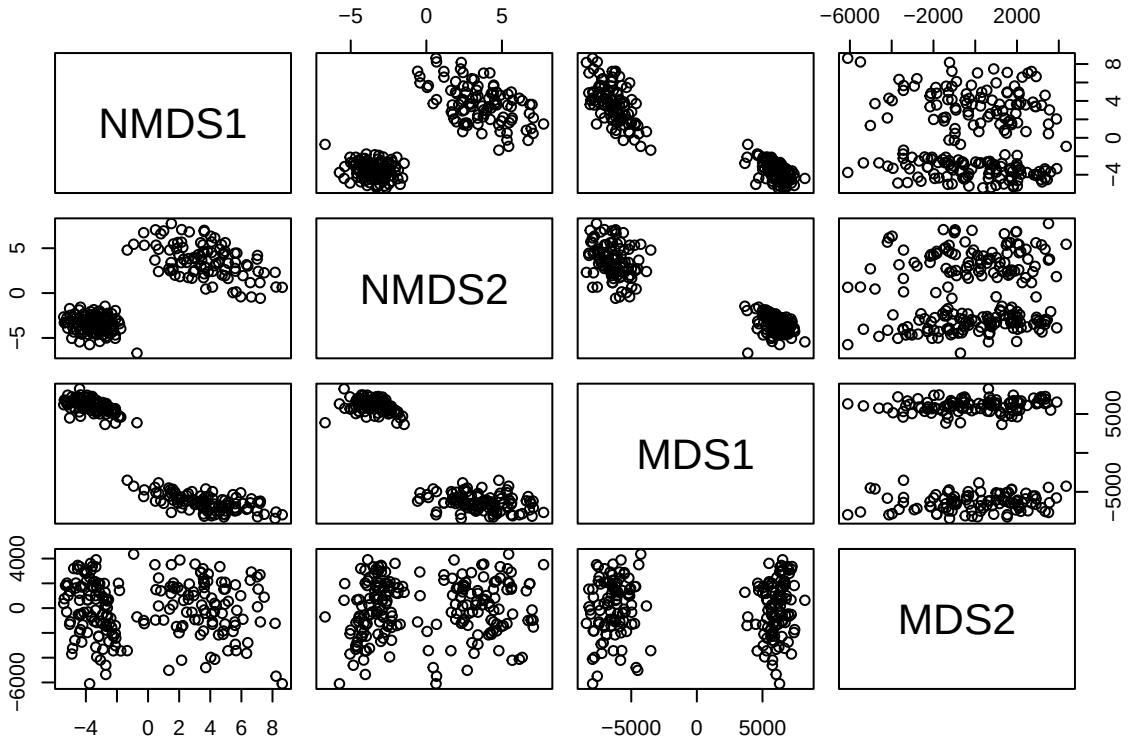
We need at least 20 dimensions to achieve a stress below 5.

Ex 13 Compute the correlation matrix between the first two dimensions of a metric MDS and the two-dimensional solution of your best non-metric MDS. Make a scatterplot matrix of the 4 variables. Comment on your findings.

```
corr.matrix = cbind(points.11[[which.min(stress.11)]], X)
colnames(corr.matrix) = c("NMDS1","NMDS2","MDS1","MDS2")
res = cor(corr.matrix)
round(res,3)
```

```
##        NMDS1  NMDS2   MDS1   MDS2
## NMDS1  1.000  0.770 -0.937 -0.070
## NMDS2  0.770  1.000 -0.928  0.092
## MDS1  -0.937 -0.928  1.000  0.000
## MDS2  -0.070  0.092  0.000  1.000
```

```
pairs(corr.matrix)
```


NMDS1
NMDS2
MDS1
MDS2
−5 0 5
−6000 −2000 2000
−4 0 4 8
−5 0 5
−5000 5000
−6000 0 4000
−4 0 2 4 6 8
−5000 0 5000