Practical 04 SG: Population substructure

Carlos Moyano & Kleber Reyes

2022-12-12

Population substructure

1. Read data.

```
genotypeData <- fread("Chr21.dat", drop = c(1:6))
dim(genotypeData)
```

```
## [1]    203 138106
```

```
n <- nrow(genotypeData)
max(genotypeData,na.rm=TRUE)
```

```
## [1] 2
```

```
min(genotypeData,na.rm=TRUE)
```

```
## [1] 0
```

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.

```
D <- as.matrix(dist(genotypeData, method = "manhattan"))
```

```
D[1:5,1:5]
```

```
##       1     2     3     4     5
## 1     0 53495 55007 58174 53794
## 2 53495     0 55372 55995 55699
## 3 55007 55372     0 54815 55683
## 4 58174 55995 54815     0 59046
## 5 53794 55699 55683 59046     0
```

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

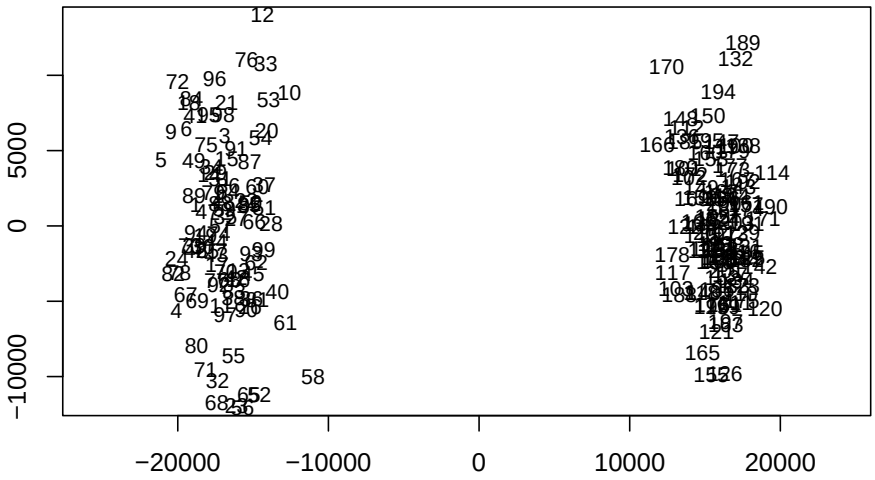
The division of the Manhattan distance by the number of polymorphisms gives the **allele-sharing distance**. This is done for the sake of interpretability, because scaling the Manhattan distance in this case results in the following behaviour: pair of individuals two units apart will then differ by two alleles at all loci.

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 subpopulation?

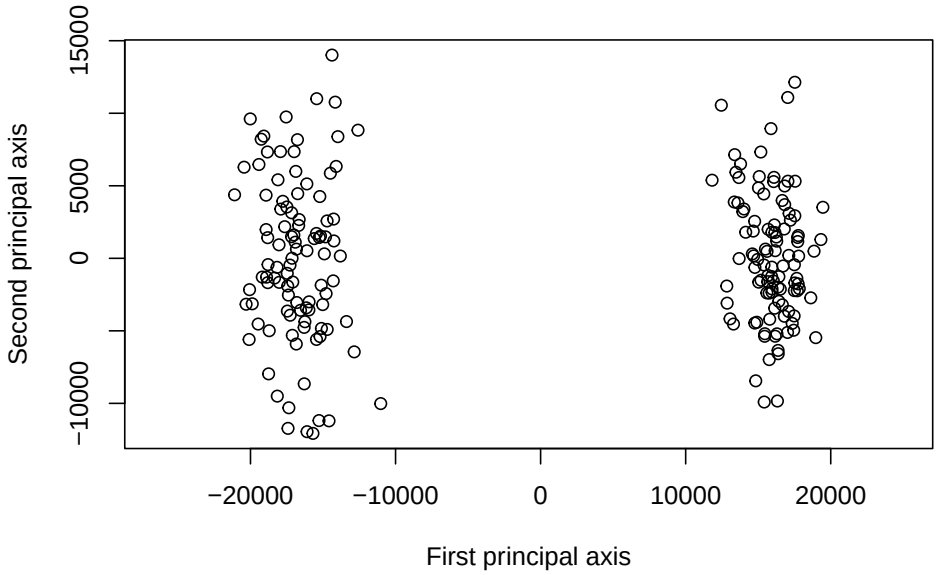
```
mds.out <- cmdscale(D,eig=TRUE,k=2)
mds.out$GOF
```

```
## [1] 0.1703581 0.1703605
```

```
X <- mds.out$points[,1:2]
eqscplot(mds.out$points,type="n")
text(mds.out$point,row.names(genotypeData),cex=.8)
```




```
plot(X[,1],X[,2],asp=1,xlab="First principal axis", ylab="Second principal axis")
```




```
## [1] "There are 99 individuales belonging to the 1st subpopulation (1st PC < 0)"
```

```
## [1] "There are 104 individuales belonging to the 2nd subpopulation (1st PC > 0)"
```

With k=2, the obtained MDS result clearly shows 2 different human subpopulations. The number of individuals belonging to each subpopulation is around 100 (calculated counting the projected points with values in the 1st PC < 0 and with values > 0).

5. Report the first 10 eigenvalues of the solution.

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

```
##  [1] 54920034481  5138177890  4707357775  4643815383  4475557521  4247865984
##  [7]  4207565510  4078798696  3913698113  3895665124
```

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

Since MDS is a *dimensionality reduction* technique having the objective of representing a n x n matrix in **less than** n dimensions (if not there's no dimensionality reduction), of course the answer to this question is affirmative. It's always possible to represent the original n x n matrix perfectly by using the same n dimensions and not performing any kind of dimensionality reduction.

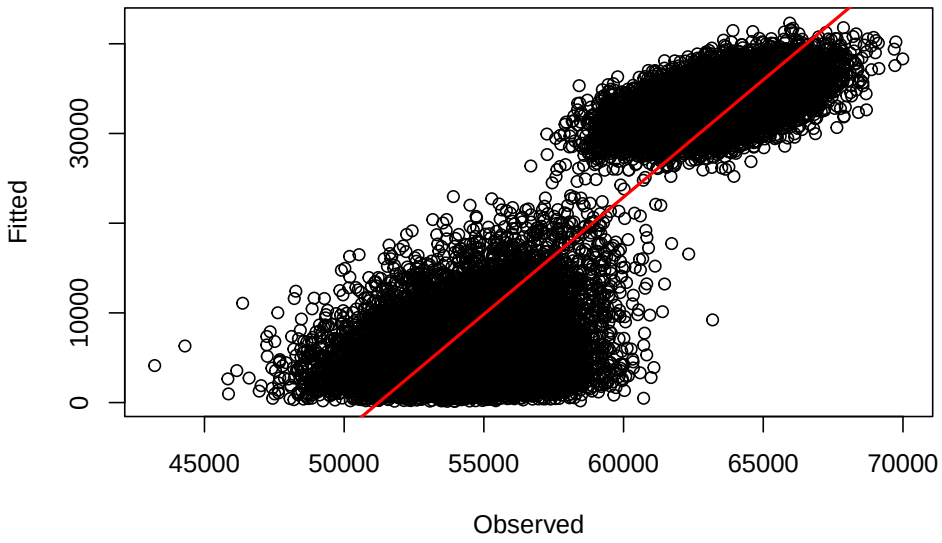
7. What is the goodness-of-fit of a two-dimensional approximation to your distance matrix? Explain which criterium you have used.

If the criterium used is the scatter plot of the estimated distance vs. the original distance, a satisfactory GoF means visualizing a linear relationship in the plot. This means that the original distance can be linearly recovered from the estimated distance.

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.

```
Dest <- as.matrix(dist(X))
Dobs.vec <- D[lower.tri(D)]
Dest.vec <- Dest[lower.tri(Dest)]
model <- lm(Dest.vec~Dobs.vec)
summary(model)
```

```
##
## Call:
## lm(formula = Dest.vec ~ Dobs.vec)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24300.7  -3360.7    429.1   3807.8  25067.4
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.339e+05  4.655e+02  -287.5   <2e-16 ***
## Dobs.vec     2.613e+00  7.880e-03   331.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5551 on 20501 degrees of freedom
## Multiple R-squared:  0.8428, Adjusted R-squared:  0.8428
## F-statistic: 1.099e+05 on 1 and 20501 DF,  p-value: < 2.2e-16
```

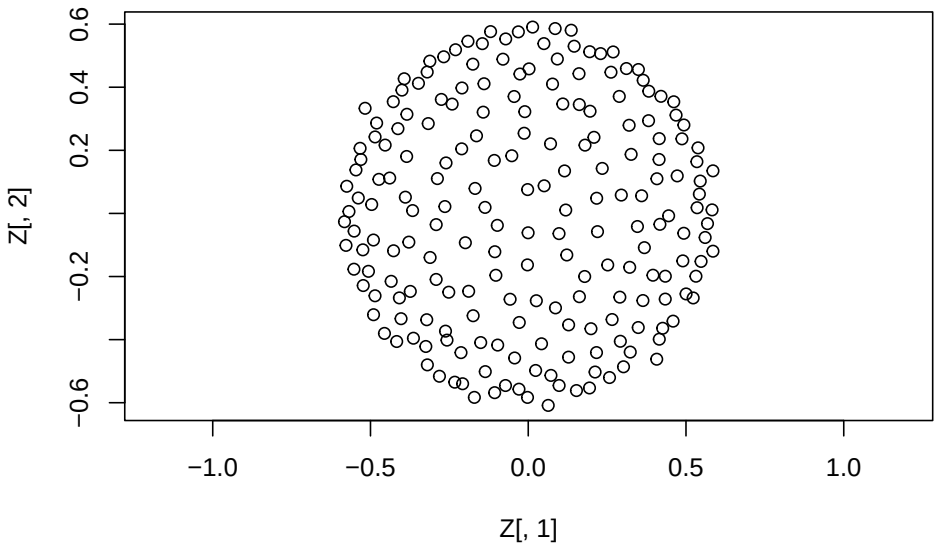


The output obtained depicts two clouds of points that vaguely remind of a straight diagonal line. This lack of definition in the linear relationship between the estimated distance and the original distance indicates that the GoF of our two-dimensional MDS is not very good.

The adjusted R-squared of the regression is 0.843, which is remarkably and indicates that a high percentage of the variability in the estimated distance can be explained using just the original distance as only explanatory variable of a linear regression.

9. We now try non-metric multidimensional scaling using the isoMDs instruction. We use a random initial configuration. or the sake of reproducibility, make this random initial configuration with the instructions. Make a plot of the two-dimensional solution. Do the results support that the data come from one homogeneous population? Try some additional runs of isoMDS with different initial configurations, or eventually using the classical metric solution as the initial solution. What do you observe?

```
set.seed(12345)
k <- 2
yinit <- scale(matrix(runif(k*n),ncol=k),scale=FALSE)
nmds.out <- isoMDS(D,k=k,y=yinit)
s <- nmds.out$stress
Z <- nmds.out$points[,1:2]
plot(Z[,1],Z[,2],asp=1)
```

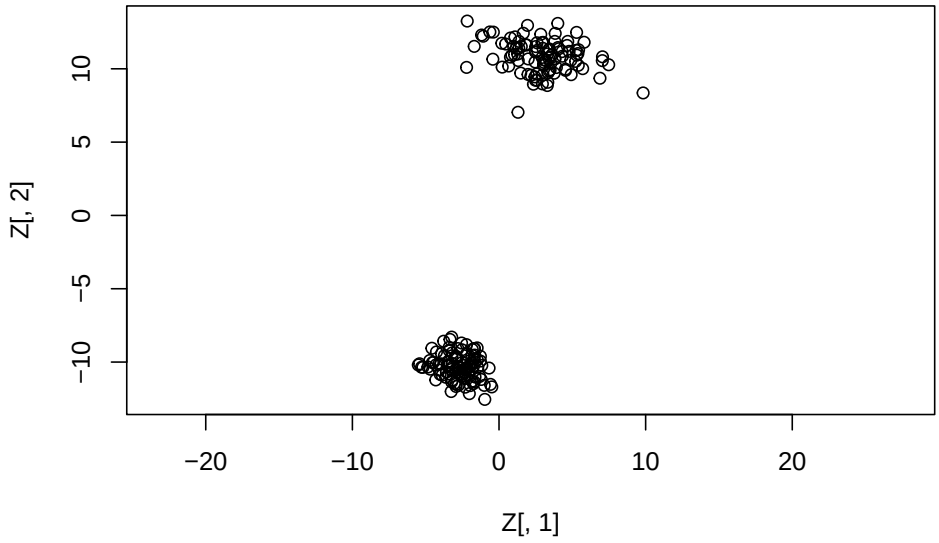



```
s
```

```
## [1] 41.62932
```

In the plot of the non-metric MDS two-dimensional result no more than a single homogeneous population is identifiable. When `isoMDS` is run with different initial configurations (using different seeds/random states), in some cases the circular shape is maintained, but the points seem to be different at first glance, and in others, instead of a single circular point cloud, there are 2 point clouds of different shape (see the example below).

```
set.seed(105)
k <- 2
yinit <- scale(matrix(runif(k*n),ncol=k),scale=FALSE)
nmds.out <- isoMDS(D,k=k,y=yinit)
s <- nmds.out$stress; s
Z <- nmds.out$points[,1:2]
plot(Z[,1],Z[,2],asp=1)
```

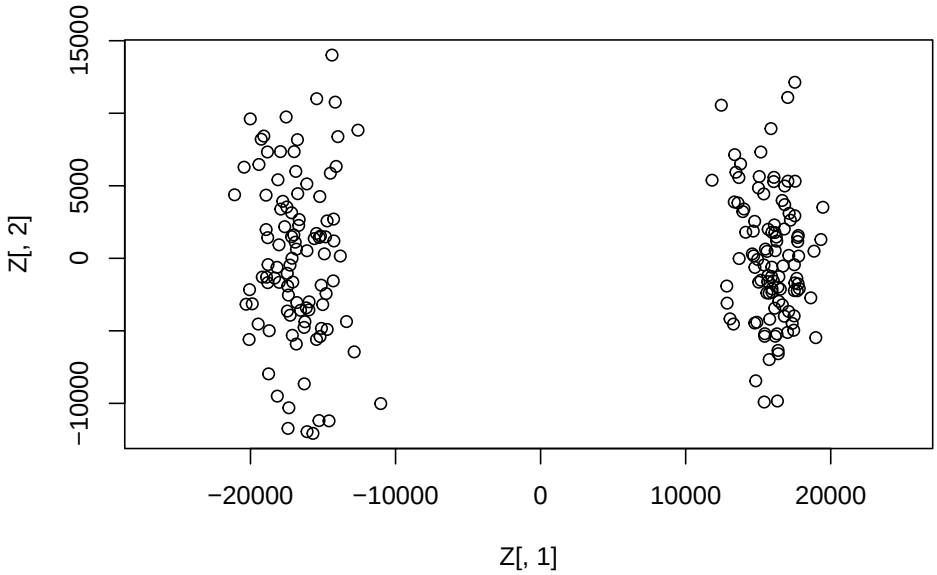



```
s
```

```
## [1] 11.86287
```

With the classical metrical solution (depicted below), the result is exactly the same as the one obtained in question 4.

```
set.seed(2209) # same seed as at the beginning
k <- 2
nmds.out <- isoMDS(D,k=k) # if y is None, classical solution
s <- nmds.out$stress; s
Z <- nmds.out$points[,1:2]
plot(Z[,1],Z[,2],asp=1)
```




```
s
```

```
## [1] 15.18508
```

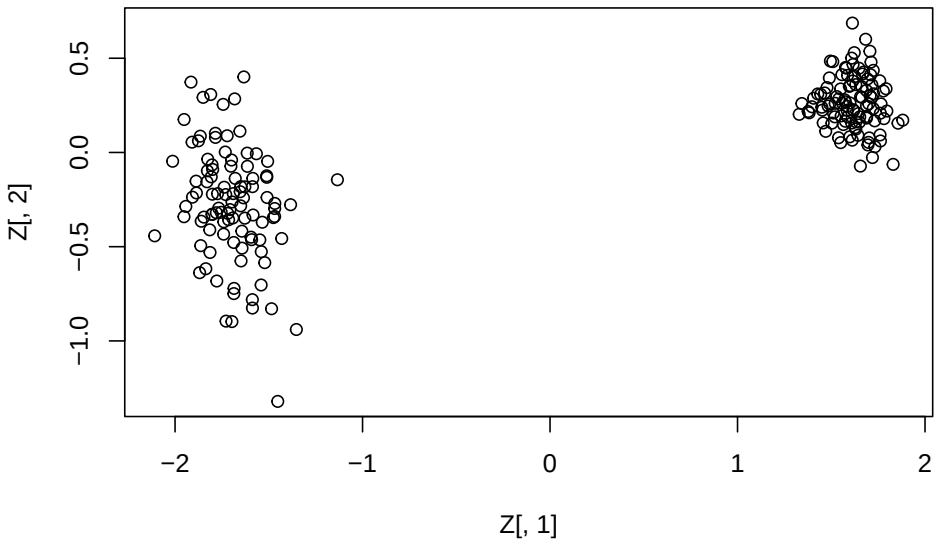
10. Set the seed of the random number generator to 123. Then run isoMDS a hundred times, each time using a different random initial configuration using the instructions above. Save the final stress-value and the coordinates of each run. Report the stress of the best run, and plot the corresponding map.

We have used the runif function in order to have a different initial configuration in each iteration.

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

Our best run is the one with the lowest stress value, once we get the corresponding map we can plot it.

```
## [1] 11.40856
```

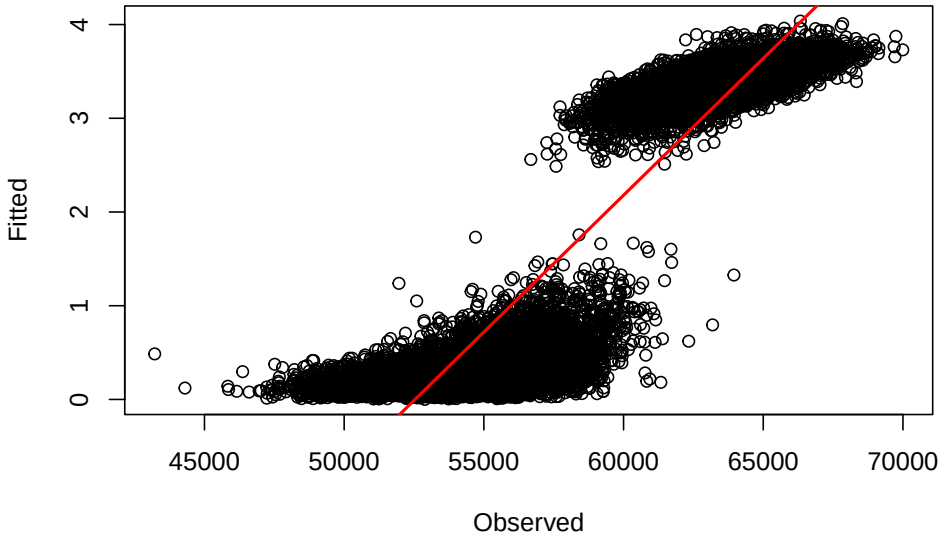


The minimum stress value is 11.40856 that is really close to the obtained with the previous seed 105. Once again, we have 2 point clouds.

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(Z))
Dest.best.vec <- Dest.best[lower.tri(Dest.best)]
model2 <- lm(Dest.best.vec~Dobs.vec)
summary(model2)$r.squared
```

```
## [1] 0.869096
```



This time we can notice a better separation between the 2 point clouds than the obtained in question 8.

The adjusted R-squared of the regression is 0.869, more or less the same as the previous calculation.

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.list2 <- c()
stress.value = 6
k = 1
while(stress.value > 5) {
  init <- scale(matrix(runif(k*n),ncol=k),scale=FALSE)
  nmds.out <- isoMDS(D,k=k,y=init, maxit=100, trace=FALSE)
  stress.value = nmds.out$stress
  stress.list2[k] = stress.value
  k = k +1
}
k-1
```

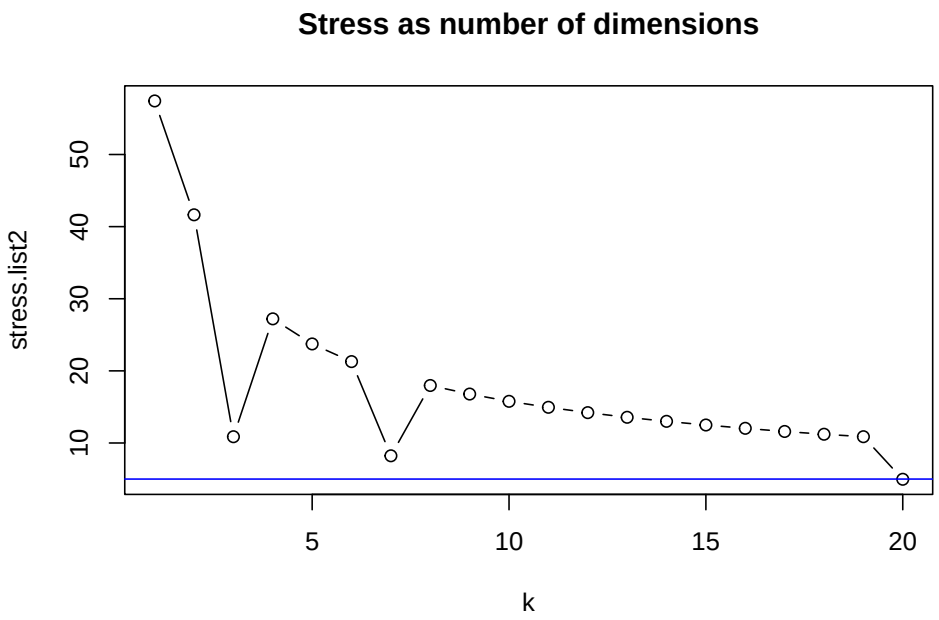
```
## [1] 20
```

```
stress.value
```

```
## [1] 4.971357
```

We need 20 dimensions to obtain a stress value below 5.

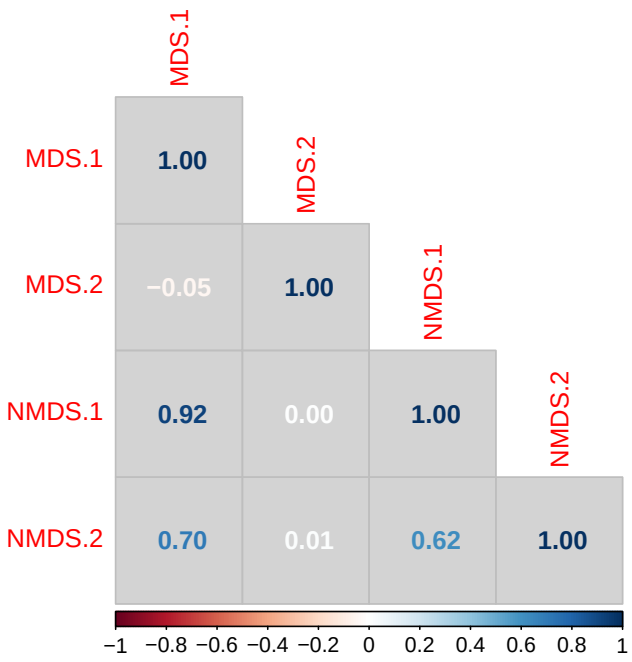
```
plot(1:length(stress.list2), stress.list2,type="b",xlab="k", main="Stress as number of dimensions")
abline(h=5, col="blue")
```



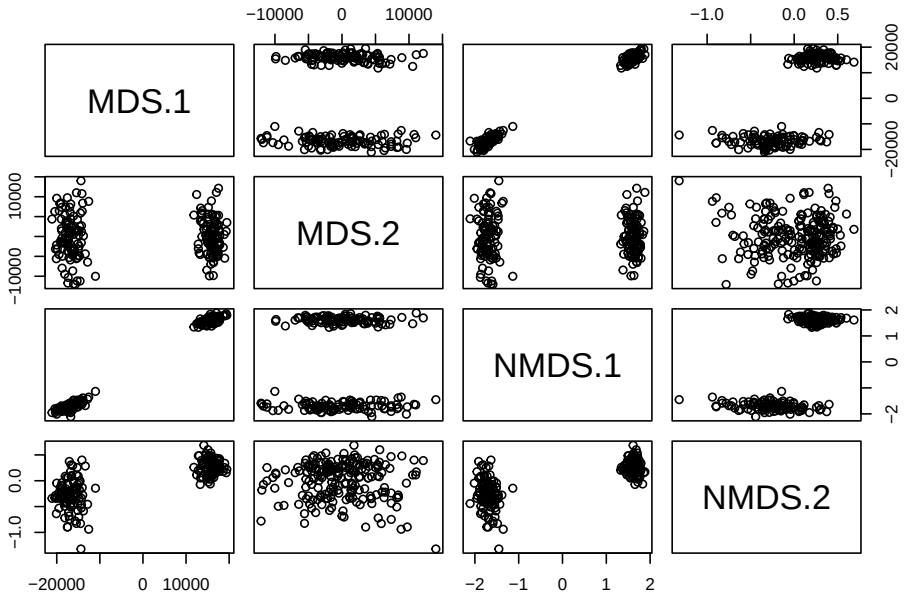
In this plot we can see the decreasing trend of stress as the dimensions increase.

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.

```
matrix = cbind(X,Z)
colnames(matrix) = c("MDS.1","MDS.2","NMDS.1","NMDS.2")
cor.mds = round(cor(matrix, method="spearman"), 2)
corrplot(cor.mds, method = 'number', type = 'lower', bg = "lightgray")
```




```
pairs(matrix)
```



With the info obtained from the correlation matrix and the pairs plot, we can affirm that MDS.1 and NMDS.1 are highly correlated (correlation is almost 1), but the same DOES NOT HAPPEN with MDS.2 and NMDS.2 (correlation is 0.01). This makes sense *partly* since we expect that both techniques find more or less the same solution (the same dimensions) as part of the dimensionality reduction process. It is remarkable to point out that there's a lot of positive correlation between NMDS.2 and NMDS.1 (correlation is 0.62), while this

does not happen between MDS.1 and MDS.2. This behaviour is not desired since we would like to obtain *independent* dimensions, and this seems not to be the case in the output of the best non-metric MDS.