Practical 05 SG: Population substructure

Lovro Katalinic and Ivan Almer

Hand-in: 21/12/2020

Resolve the following exercise in groups of two students. Perform the computations and make the graphics that are asked for in the practical below. Take care to give each graph a title, and clearly label x and y axes, and to answer all questions asked. You can write your solution in a word or Latex document and generate a pdf file with your solution. Alternatively, you may generate a solution pdf file with Markdown. You can use R packages **genetics**, **MASS**, **data.table** and others for the computations. Take care to number your answer exactly as in this exercise. Upload your solution in **pdf format** to the web page of the course at raco.fib.upc.edu no later than the hand-in date.

```
## Loading required package: combinat
```

```
##
## Attaching package: 'combinat'
```

```
## The following object is masked from 'package:utils':
##
##     combn
```

```
## Loading required package: gdata
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
## Loading required package: gtools
```

```
## Loading required package: MASS
```

```
## Loading required package: mvtnorm
```

```
##
```

```
## NOTE: THIS PACKAGE IS NOW OBSOLETE.
```

```
##
```

```
##   The R-Genetics project has developed an set of enhanced genetics

##   packages to replace 'genetics'. Please visit the project homepage

##   at http://rgenetics.org for informtion.

##

##
## Attaching package: 'genetics'

## The following objects are masked from 'package:base':
##
##     %in%, as.factor, order

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:gdata':
##
##     first, last
```

1. The file Chr21.dat contains genotype information of a set of individuals of unknown background. Load this data into the R environment with the *fread* 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.

```r
dataset = fread('Chr21.dat', data.table = FALSE)
dataset = dataset[,7:ncol(dataset)]
```

2. (1p) 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.

```r
dists.all = as.matrix(dist(dataset, method = 'manhattan'))

dists = dist(dataset[1:5,], method = 'manhattan')
d = matrix(nrow = 5, ncol = 5)

dists.arr = c(dists)
count = 1

for(i in 1:(nrow(d) - 1)) {
  for(j in (i+1):(ncol(d))) {
    #print(dists.arr[count])
    d[i,j] = dists.arr[count]
    d[j,i] = dists.arr[count]
    count = count + 1
  }
}

d[is.na(d)] = 0.0
d
```

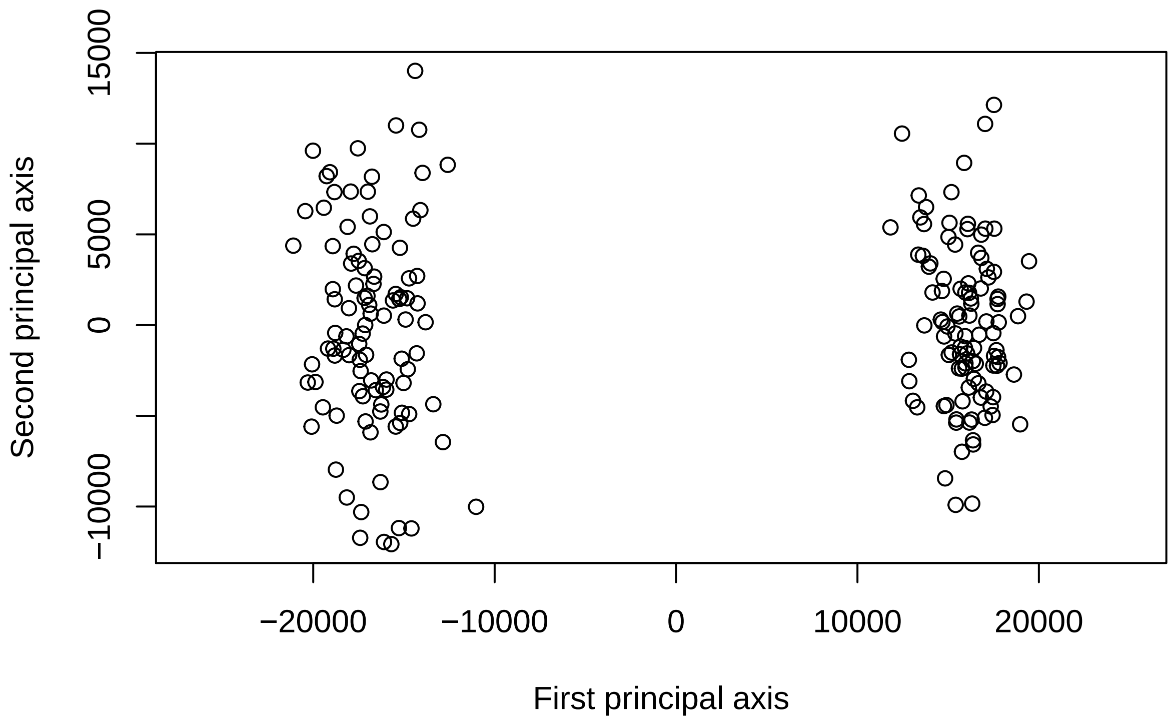
```
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,]     0 53495 55007 58174 53794
## [2,] 53495     0 55372 55995 55699
## [3,] 55007 55372     0 54815 55683
## [4,] 58174 55995 54815     0 59046
## [5,] 53794 55699 55683 59046     0
```

3. (1p) 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?

Manhattan distance is actually the same as Allele sharing distance since distance. Let's see and example: manhattan distance for one variant with values 0 and 2 (let's say AA and BB) is |0-2| = 2 and the allele sharing distance is 2-(number of shared alleles) = 2 - 0 = 2. This can be shown for every combination of variants. Obviously if we scale the Allele sharing distance with number of variants we obtain the results which is smaller than Manhattan distance but they are proportional with the coefficiet $1/K$ where K is the number of variants.

4. (2p) 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.out <- cmdscale(dists.all,k=2,eig=TRUE)

X <- mds.out$points[,1:2]
plot(X[,1],X[,2],asp=1,xlab="First principal axis", ylab="Second principal axis")
```



```
X.mds = X

left.pop = sum(X[,1] < 0)
left.pop
```

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

```
right.pop = sum(X[,1] > 0)
right.pop
```

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

There seem to exist 2 subpopulations i the shown population. In the "left" population there are 99 units and in the "right" population there are 104 units.

5. (1p) 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. (1p) Does a perfect representation of this $n \times n$ distance matrix exist, in n or fewer dimensions? Why so or not?

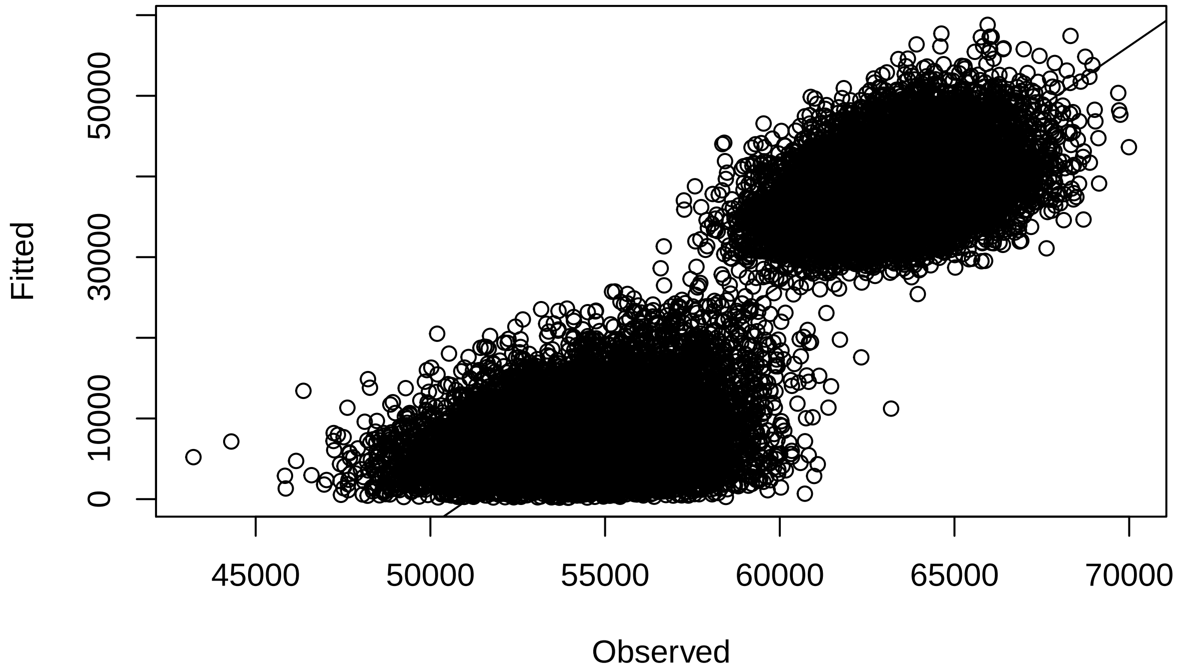
A perfect representation of the distance matrix does not exist because by reducing dimensions we take away the degrees of freedom which enable the original distance matrix to be such as it is.

7. (1p) What is the goodness-of-fit of a two-dimensional approximation to your distance matrix? Explain which criterium you have used.

/

8. (2p) 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.

```
approx.dist = as.matrix(dist(X, method = 'manhattan'))

Dobs.vec <- dists.all[lower.tri(dists.all)]
Dest.vec <- approx.dist[lower.tri(approx.dist)]

plot(Dobs.vec,Dest.vec,xlab="Observed",ylab="Fitted")
model = lm(Dest.vec ~ Dobs.vec)
abline(model)
```



```
r2 = summary(model)$r.squared
r2
```

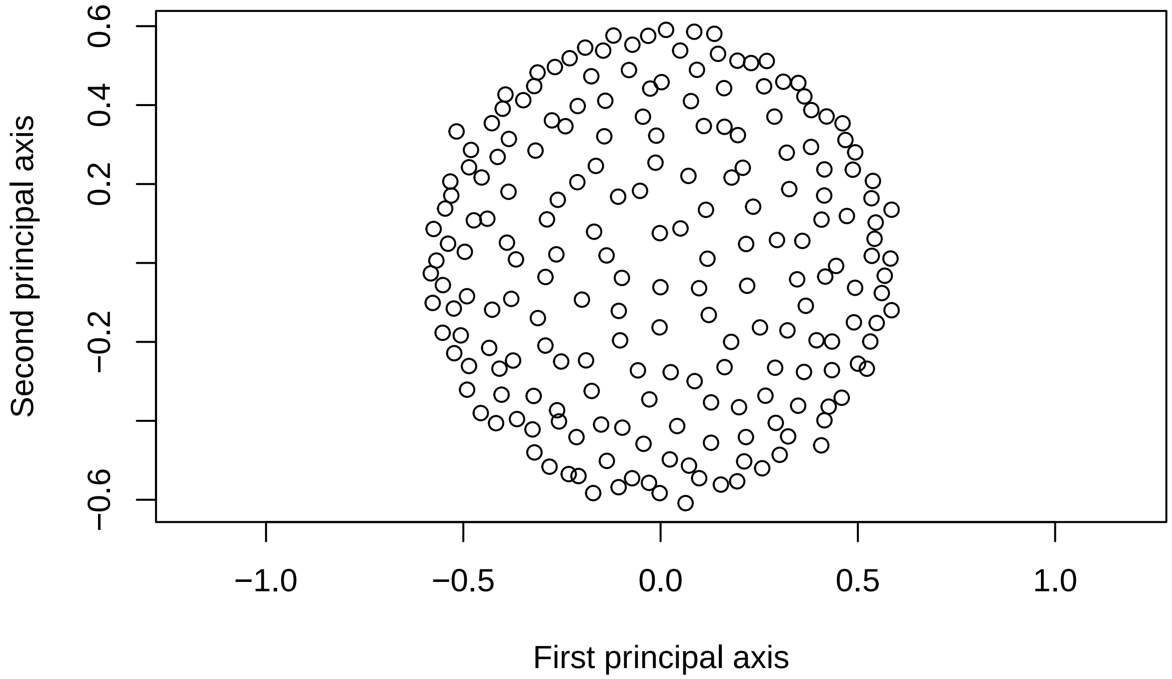
```
## [1] 0.8240959
```

9. (1p) We now try non-metric multidimensional scaling using the `isoMDs` instruction. We use a random initial configuration. For the sake of reproducibility, make this random initial configuration with the instructions:

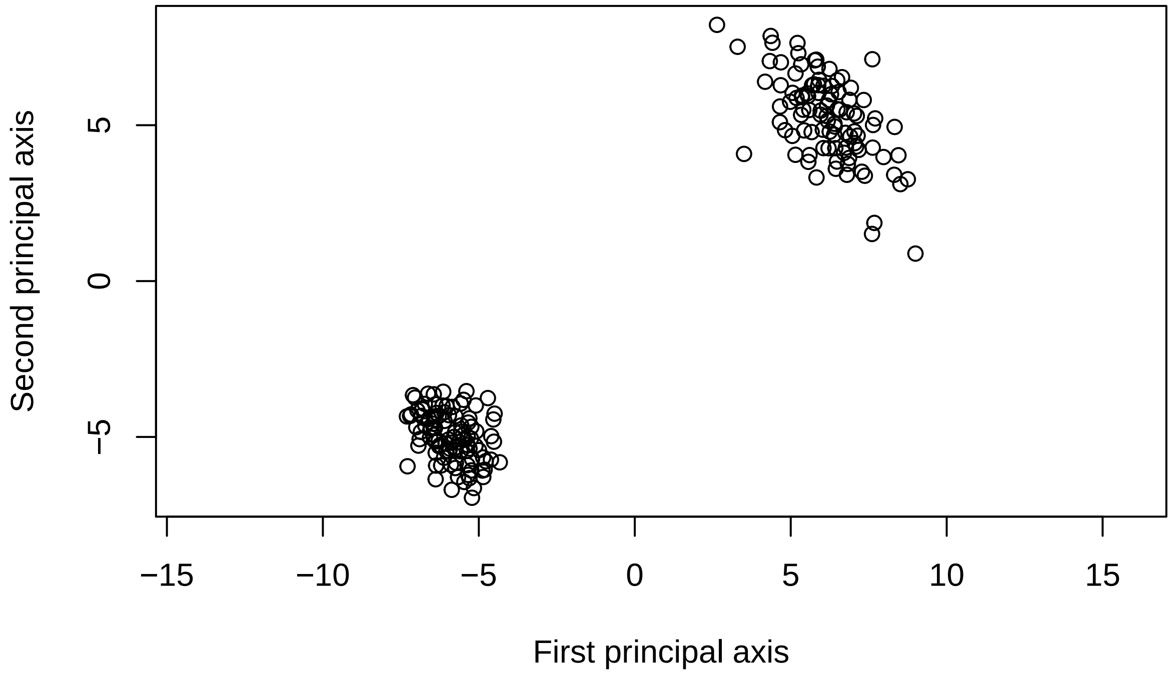
   ```
   set.seed(12345)
   init <- scale(matrix(runif(2*n),ncol=2),scale=FALSE)
   ```

   where n represents the sample size. 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?

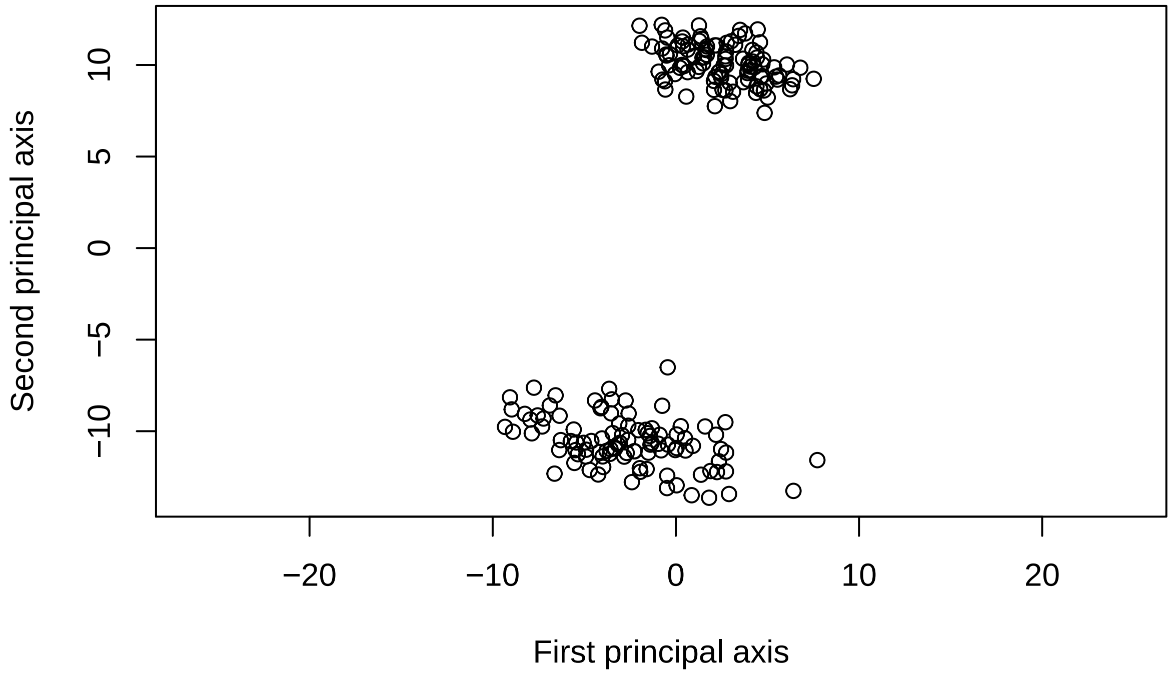
```
n = nrow(dataset)
set.seed(12345)

init <- scale(matrix(runif(2*n),ncol=2),scale=FALSE)
nmds.out <- isoMDS(dists.all,k=2,y=init, trace = FALSE)
X = nmds.out$points
plot(X[,1],X[,2],asp=1,xlab="First principal axis", ylab="Second principal axis")
```



```
init <- scale(matrix(runif(2*n),ncol=2),scale=FALSE)
nmds.out <- isoMDS(dists.all,k=2,y=init, trace = FALSE)
X = nmds.out$points
plot(X[,1],X[,2],asp=1,xlab="First principal axis", ylab="Second principal axis")
```

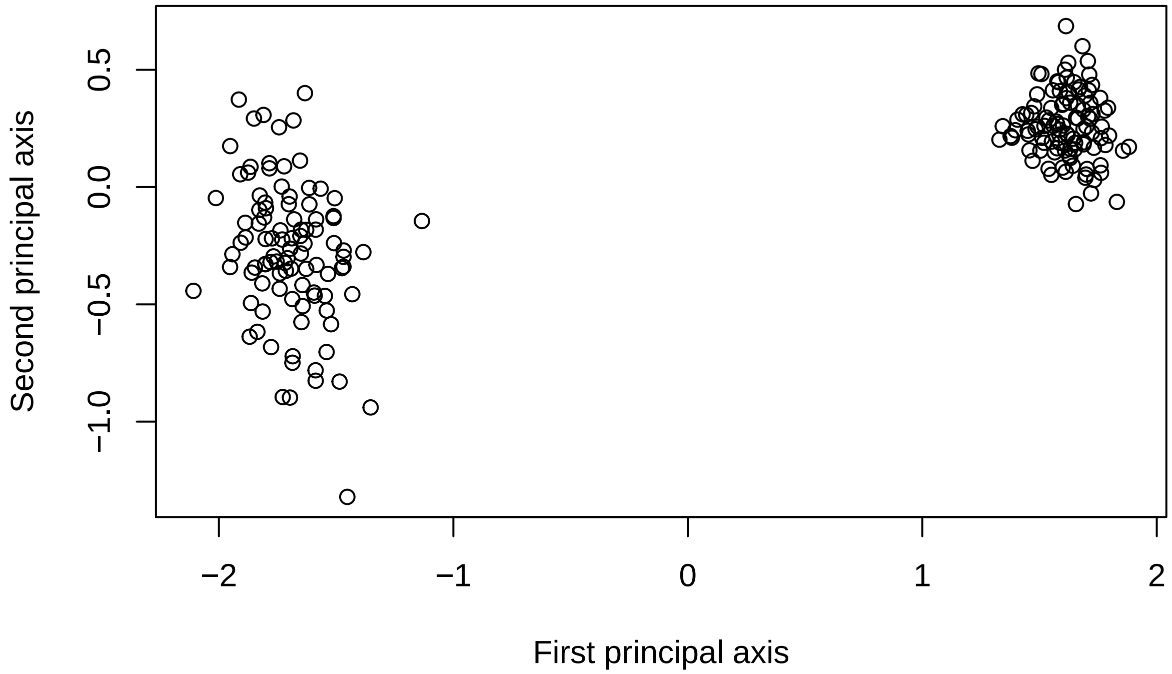


```
init <- scale(matrix(runif(2*n),ncol=2),scale=FALSE)
nmds.out <- isoMDS(dists.all,k=2,y=init, trace = FALSE)
X = nmds.out$points
plot(X[,1],X[,2],asp=1,xlab="First principal axis", ylab="Second principal axis")
```



Initial configuration produces much much different results from the results in the additional runs.

10. (1p) 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.

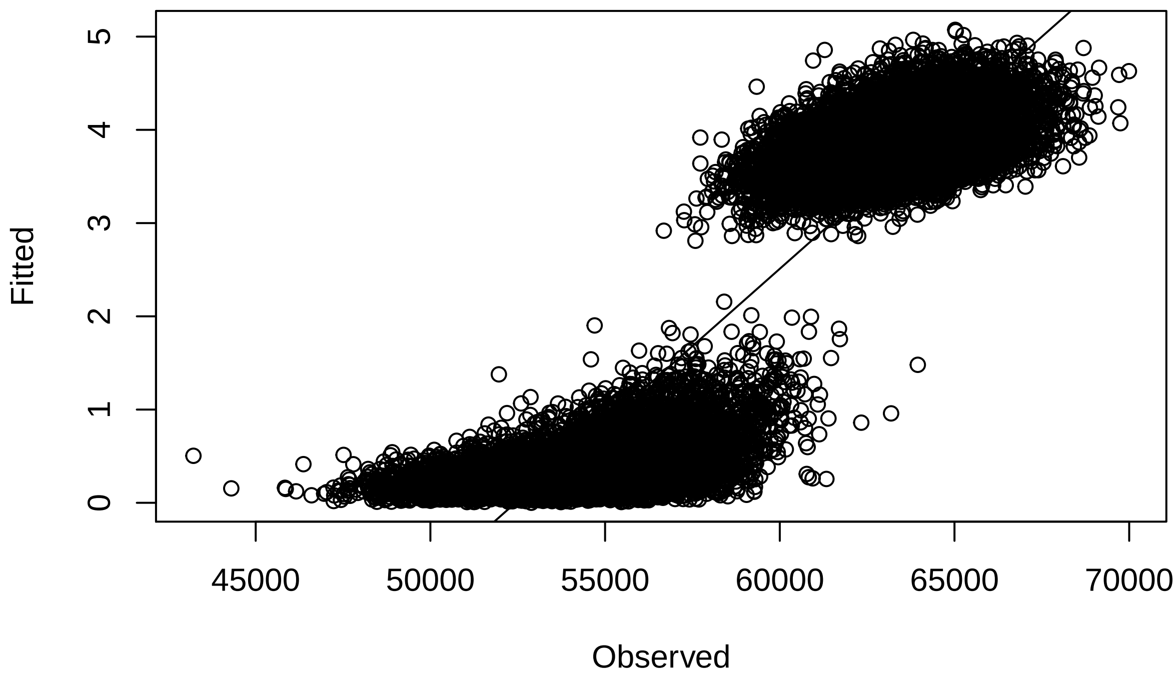


```
## [1] 11.40856
```

The best stress value is 11.41, and the result of dimensionality reduction is visible on the plot.

11. (1p) 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.

```
approx.dist = as.matrix(dist(X.best, method = 'manhattan'))

Dobs.vec <- dists.all[lower.tri(dists.all)]
Dest.vec <- approx.dist[lower.tri(approx.dist)]

plot(Dobs.vec,Dest.vec,xlab="Observed",ylab="Fitted")
model = lm(Dest.vec ~ Dobs.vec)
abline(model)
```



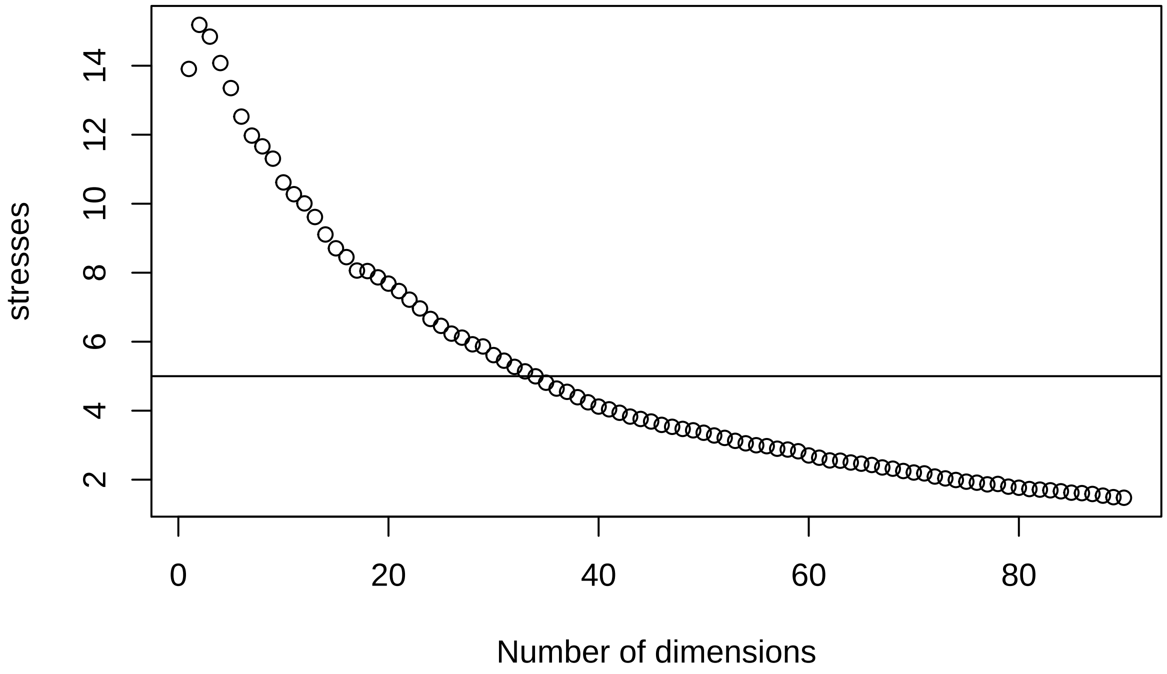
```
r2 = summary(model)$r.squared
r2
```

```
## [1] 0.8611736
```

There are 2 "groups" which can be observed on the plot. Coefficient of determination is 0.86.

12. (1p) Compute the stress for a $1, 2, 3, 4, \ldots, n$-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

```
#stresses = c()
#for(k in 1:90) {
#  nmds.out <- isoMDS(dists.all,k=k, trace = FALSE)
#  stresses = append(stresses, nmds.out$stress)
#}

stresses = read.csv('stresses.csv')[,2]

plot(1:90, stresses, xlab = 'Number of dimensions')
abline(5,0)
```



```
k.min = 1
for(i in 1:length(stresses)) {
  if(stresses[i] < 5) {
    k.min = i
    break;
  }
}

k.min
```

```
## [1] 34
```

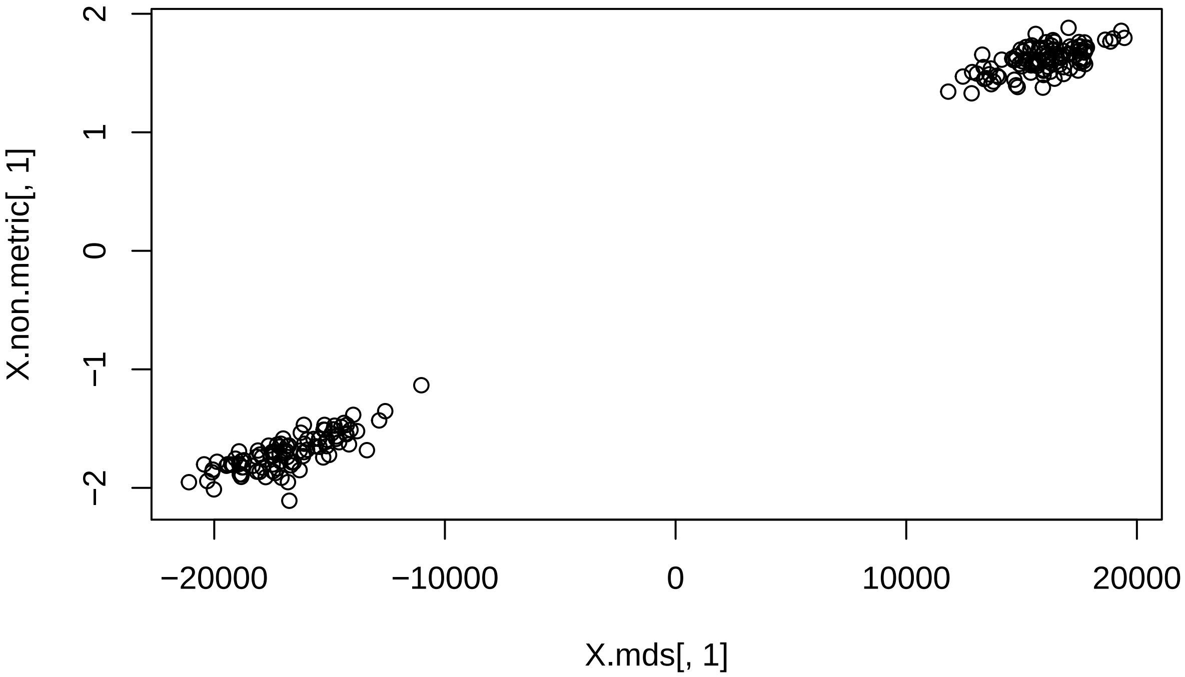
At least 34 dimensions are needed for stress to fall below 5. It is also visible on the plot. We plotted only 90 values because the stress value will only continue to reduce and it started to take a long time to calculate the MDS for large n.

13. (2p) 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.

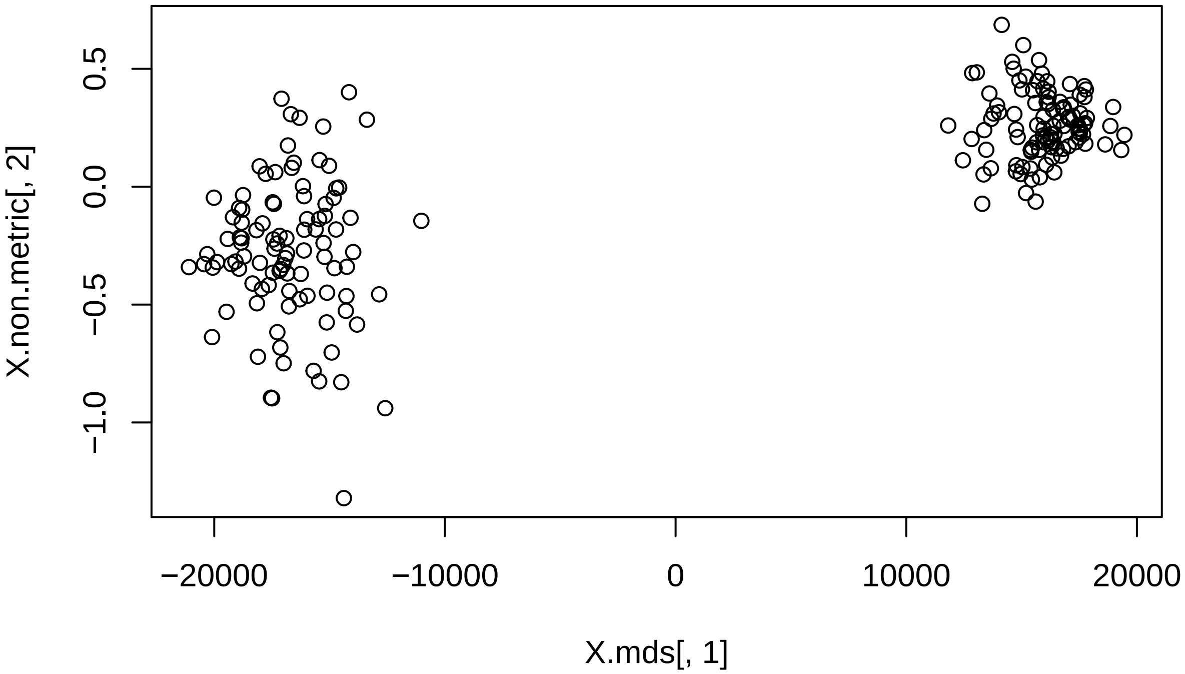
```
X.non.metric = X.best
cor(X.mds, X.non.metric)
```

```
##             [,1]        [,2]
## [1,] 0.99736763  0.75047759
## [2,] 0.01153539 -0.05454761
```

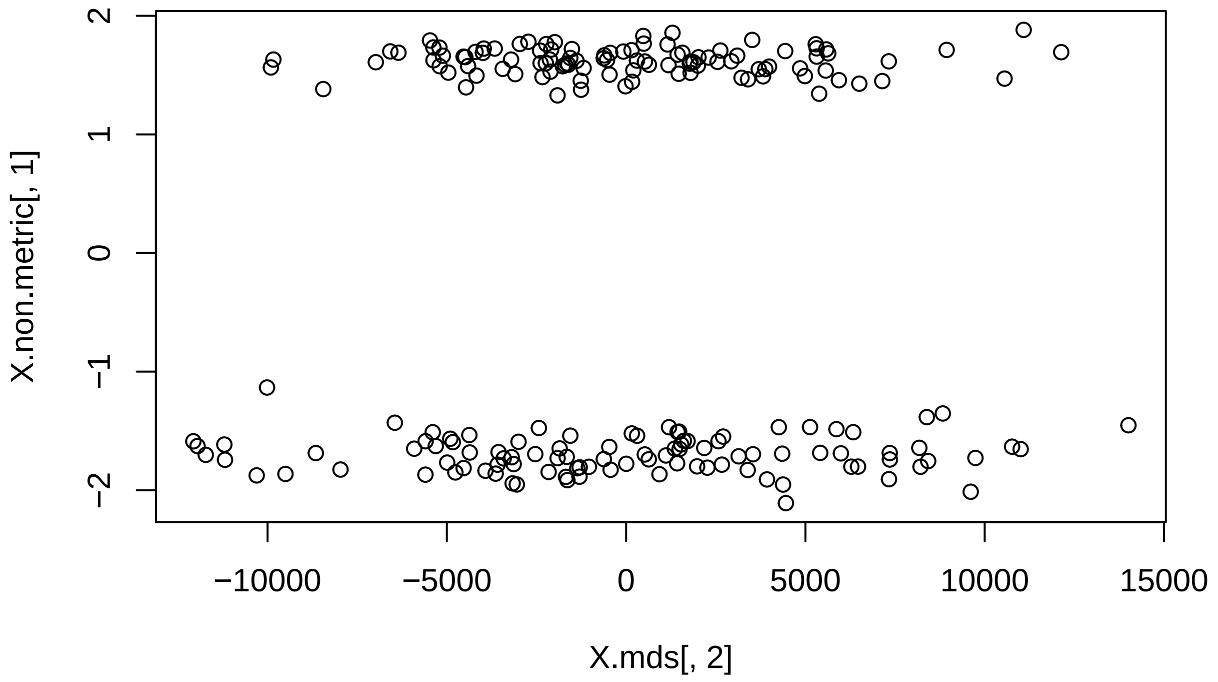
```
plot(X.mds[,1], X.non.metric[,1])
```



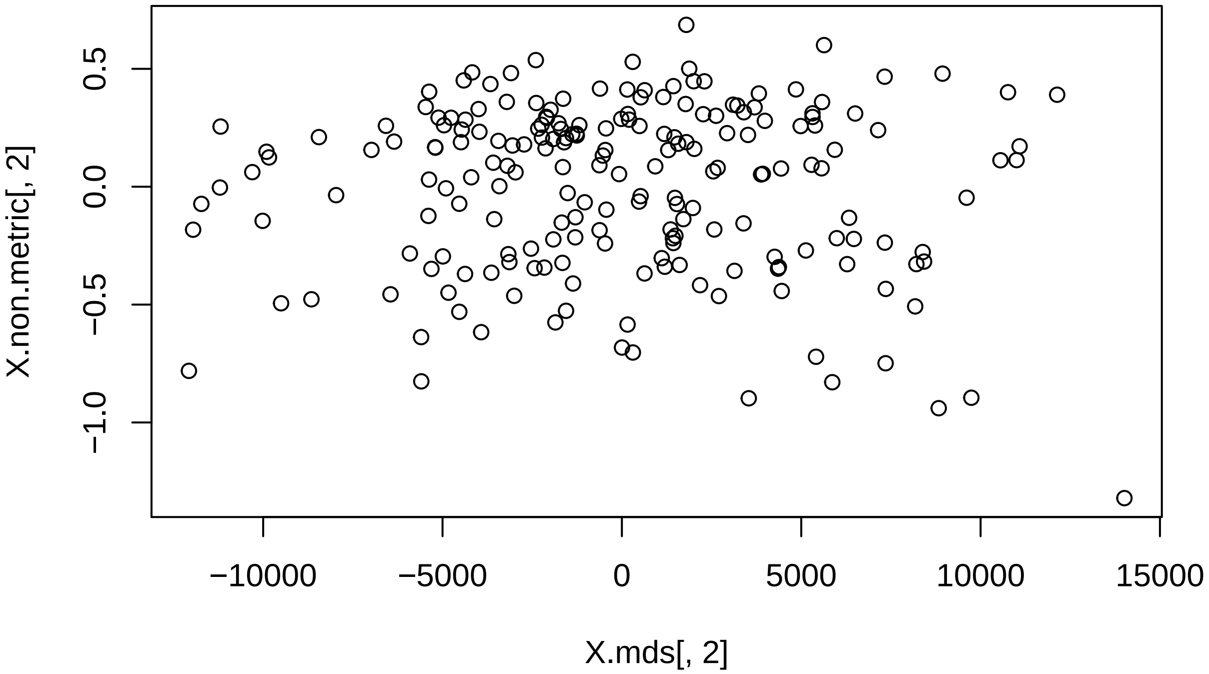
```
plot(X.mds[,1], X.non.metric[,2])
```



```
plot(X.mds[,2], X.non.metric[,1])
```



```
plot(X.mds[,2], X.non.metric[,2])
```



The first dimensions of both matrices (1,1) are highly positively correlated (1st plot) and second dimensions of both matrices (2,2) are completely uncorrelated (4th plot). In other case of (1,2) we can observe a correlation of roughly 0.75 which is a pretty high positive correlation and that is also observable on the plot, since there is some kind of straight line which could be drawn. In the case (2,1) we observe that for each value of metric coordinate there are roughly 2 possible levels which can be achieved (-2 and 2), but there is no correlation between the 2 dimensions.