Bioinformatics and Statistical Genetics:Population Substructure

Sara Montese

```
library(genetics)

library(data.table)
library(MASS)
```

Task1

The file SNPChr20.rda contains genotype information of 310 individuals of unknown background. The genotype information concerns 50.000 SNPs on chromosome 20. Load this data into the R environment. The data file contains a matrix Y containing the allele counts (0,1 or 2) for 50.000 SNPs for one of the alleles of each SNP

```
chunk_size <- 10000
raw_data <- fread(file="Chr21.dat", sep = " ", header = TRUE, nThread = chunk_size
)
raw_data_df <- data.frame(raw_data)
```

1

1. How many variants are there in this database? What percentage of the data is missing?

```
variants_df <- raw_data_df[,7:ncol(raw_data_df)]

# Convert values different from 0, 1, or 2 to NA
variants_df[!sapply(variants_df, function(x) x %in% c(0, 1, 2))] <- NA

n <- nrow(variants_df)# individuals
p <- ncol(variants_df) #SNPs
cat("1. How many variants are there in this database? \n")

## 1. How many variants are there in this database?

cat(p)

## 138106

cat("\n")

cat("1. What percentage of the data is missing? \n")

## 1. What percentage of the data is missing?

mis <- 100*sum(is.na(variants_df))/(n*p)
cat(mis)

## 0
```

2

2. Compute the Manhattan distance matrix between the individuals (which is identical to

the Minkowsky distance with parameter λ = 1) using R function dist. Include a submatrix of dimension 5 by 5 with the distances between the first 5 individuals in your report

```
distances <- dist(variants_df, method="manhattan", diag = FALSE)
D <- as.matrix(distances)

print("Manhattan distance matrix between first 5 individuals: ")
```

```
## [1] "Manhattan distance matrix between first 5 individuals: "
```

```
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

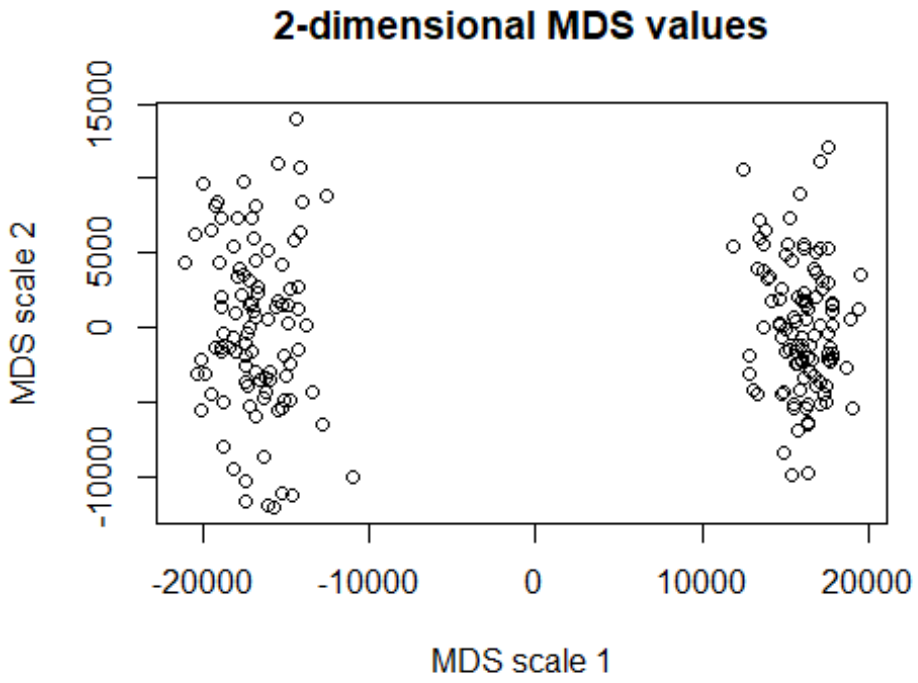
3. (1p) How does the Manhattan distance relate to the allele sharing distance?

When data is coded in (0,1,2) format, the Manhattan distance is equivalent to allele sharing distance. In particularm, the Manhattan distance is directly proportional to the allele sharing distance.

4

4. Apply metric multidimensional scaling (cmdscale) with two dimensions, k = 2, using the Manhattan distance matrix and include the 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 <- cmdscale(D, k=2, eig=TRUE)
points <- mds$points[,1:2]

plot(points[, 1], points[, 2], main="2-dimensional MDS values", xlab="MDS scale 1"
, ylab="MDS scale 2")
```




```
cat("The map shows that there are 2 different population groups. Thus, the data do
es not come from a homogenous population.")
```

```
## The map shows that there are 2 different population groups. Thus, the data does
not come from a homogenous population.
```

```
subpopulation_1 <- points[points[, 1] > 10000]
subpopulation_2 <- points[points[, 1] < 10000]
cat("In the first subpopulation there are ", length(subpopulation_1), "individuals
.\n")
```

```
## In the first subpopulation there are  208 individuals.
```

```
cat("In the second subpopulation there are ", length(subpopulation_2), "individual
s.")
```

```
## In the second subpopulation there are  198 individuals.
```

5

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

SInce we are in the context of Metric MDS, the criterium used to calculate the goodness-of-fit of the k-dimensional approximation of the distance matrix is given:

$(\sum j=1\_to\_k\ (\lambda j)\ )\ /(\sum j=1\_to\_n\ (Ti(\lambda j))$, where λj are the eigenvalues (sorted in decreasing order) of the matrix, and $T\_1(v)=|v|$, and $T2\_(v)=max(v,0)$.

where n is the original dimension. In our case k=2.

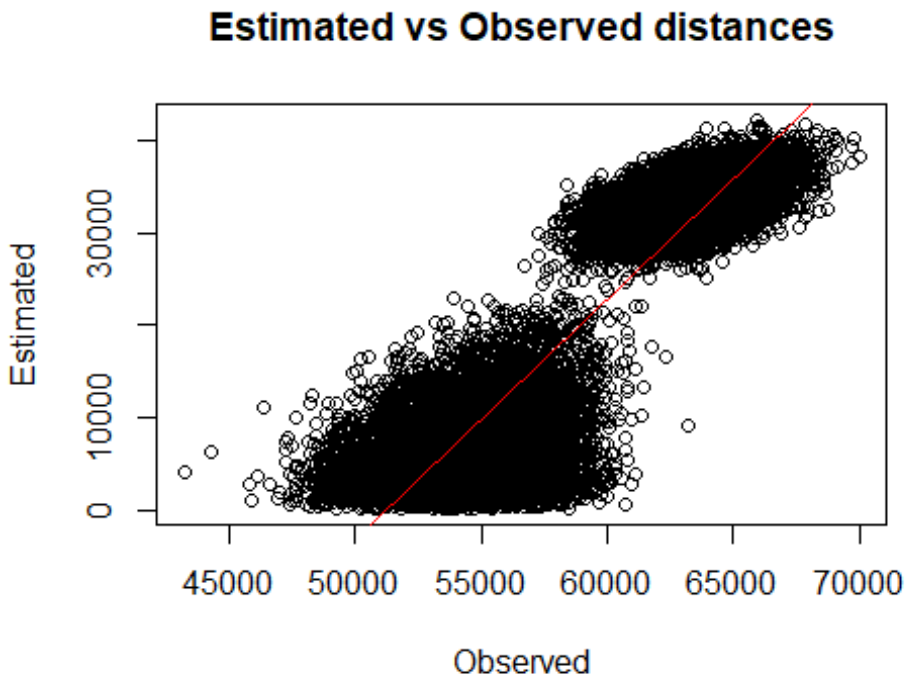
```
gof <- mds$GOF
gof

## [1] 0.1703581 0.1703605
```

6

6.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 (you can use the function lm).

The R-squared value of 0.84284 indicates that approximately 84.28% of the variability in the observed distances can be explained by the estimated distances using the linear relationship shown in the plot. This suggests a positive linear relationship between the estimated and observed distances. It is worth noting that the estimated distances are on a smaller scale than the observed distances. This may suggest that the approximation reduces the distances.

```
estimated_distances <- as.matrix(dist(points))
estimated_distances <- estimated_distances[lower.tri(estimated_distances)]
observed_distances <- D[lower.tri(D)]

plot(observed_distances, estimated_distances, xlab="Observed", ylab="Estimated", m
ain="Estimated vs Observed distances")

regression_model <- lm( estimated_distances ~ observed_distances)

abline(regression_model, col = "red")
```



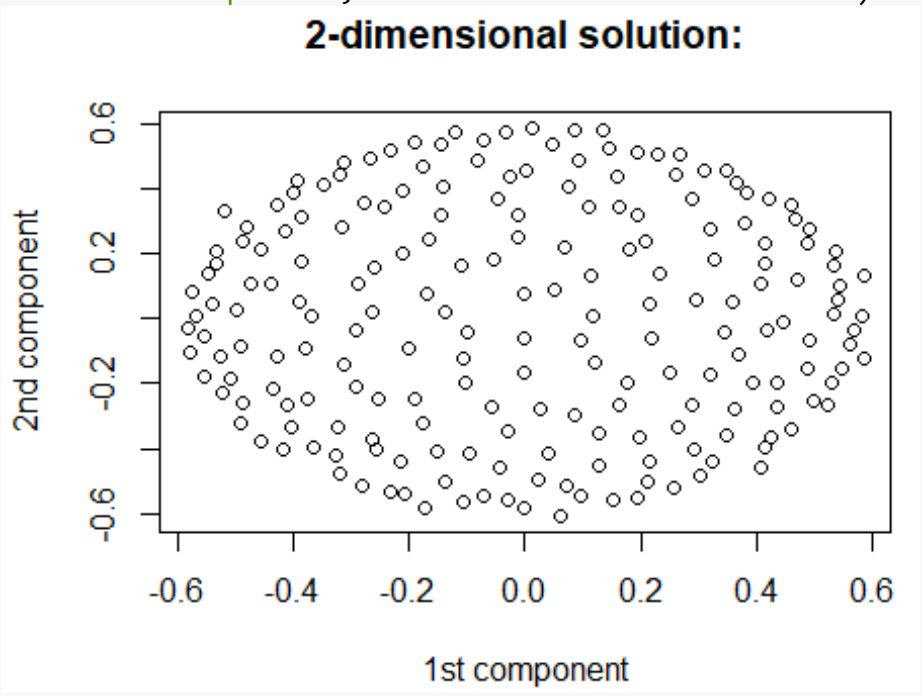
```
#  coefficient of determination (R-squared)
cat("Coefficient of Determination (R-squared): ", summary(regression_model)$r.squa
red, "\n")

## Coefficient of Determination (R-squared):  0.8428413
```

7

7. We now try a (two-dimensional) non-metric multidimensional scaling using the isoMDs function that you will find in MASS library. We use a random initial configuration and, for the sake of reproducibility, make this random initial configuration with the instructions:

#set.seed(12345) and init <- scale(matrix(runif(m*n),ncol=m),scale=FALSE) where n represents the sample size and m represents the dimensionality of the solution. Make a plot of the two-dimensional solution. Do the results support that the data come from one homogeneous population?

```
set.seed(12345)
m <-2
init <- scale(matrix(runif(m*n),ncol=m),scale = FALSE)
non_param_mds <- isoMDS(D,k=m,y=init)

## initial  value 43.001235
## iter   5 value 41.668593
## iter   5 value 41.629957
## iter   5 value 41.629319
## final  value 41.629319
## converged

plot(non_param_mds$points[,1], non_param_mds$points[,2], xlab = "1st component", y
lab = "2nd component", main="2-dimensional solution:")
```



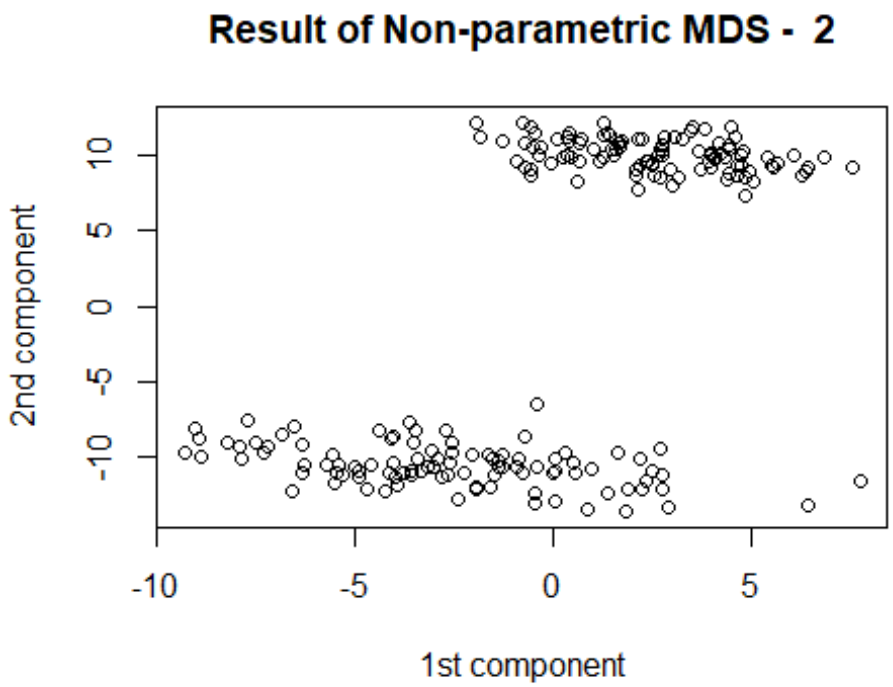
In the context of applying non-parametric Multidimensional Scaling, the absence of distinct clusters in the data suggests that there are no clearly defined or discernible groups. The outcome implies that the data likely comes from an homogeneous population, with no evident patterns of separation among the data points.

8. Try some additional runs of the two-dimensional isoMDS with different initial configurations. Make a plot of the solutions and report the STRESS for each of them. What do you observe?

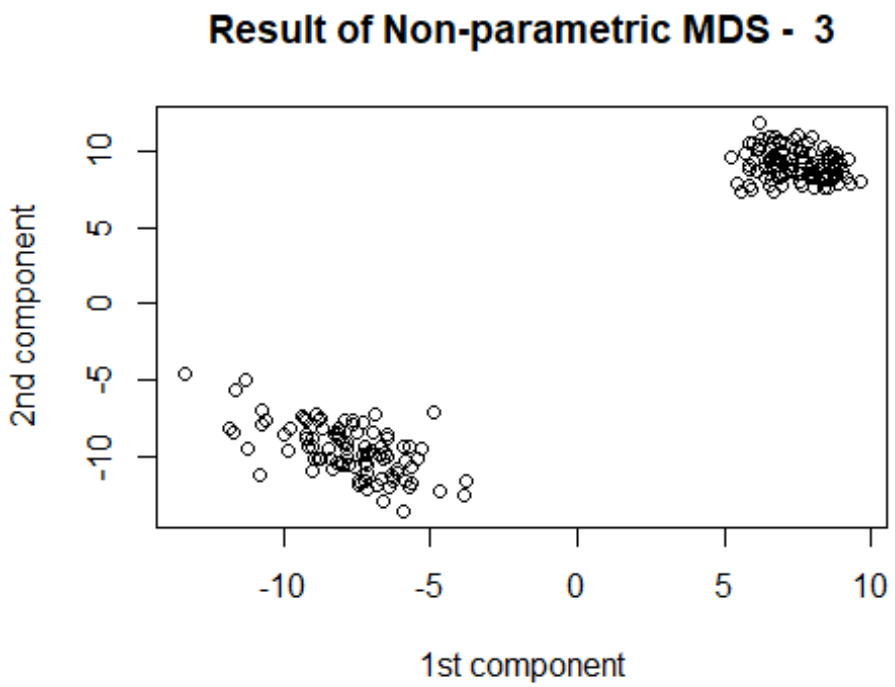
```
n_iter<-10
for (i in seq(1:n_iter)){
  init <- scale(matrix(runif(m*n),ncol=m),scale=FALSE)
  non_param_mds <- isoMDS(D, y=init, k=m)
  stress <- non_param_mds$stress

  plot(non_param_mds$points[,1], non_param_mds$points[,2], xlab = "1st component",
ylab = "2nd component", main=paste("Result of Non-parametric MDS - ", i))
  cat("Stress for ", i,"th iteration: ",stress)
}
```

```
## initial  value 42.943761
## iter   5 value 41.427197
## iter  10 value 40.002996
## iter  15 value 38.060966
## iter  20 value 29.659310
## iter  25 value 21.653732
## iter  30 value 17.463856
## iter  35 value 14.418036
## iter  40 value 13.070793
## iter  45 value 12.426749
## iter  50 value 12.098786
## final  value 12.098786
## stopped after 50 iterations
```

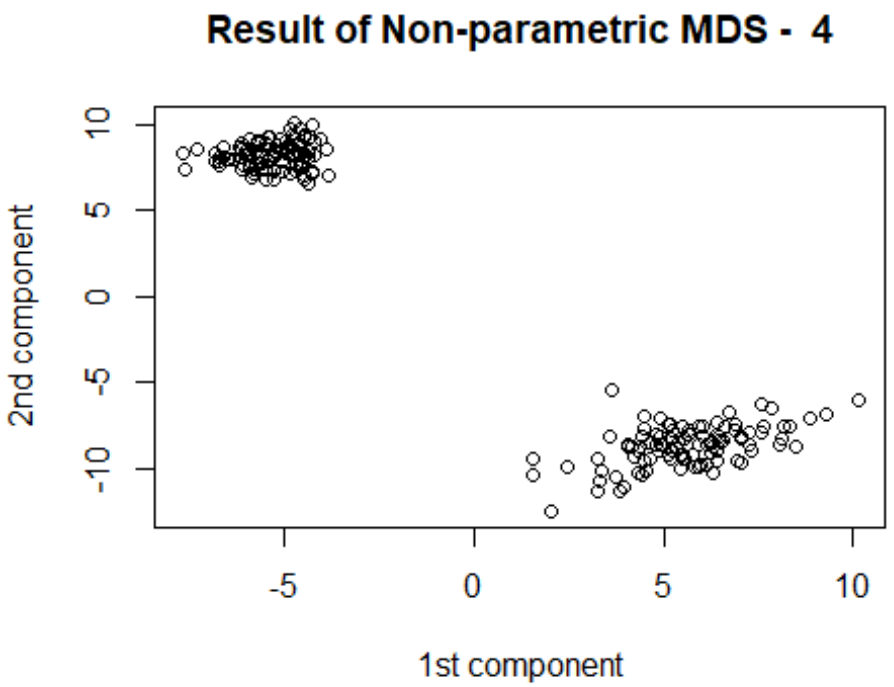

```
## Stress for  1 th iteration:  12.09879initial  value 42.871770
## iter   5 value 41.643057
## iter  10 value 39.862693
## iter  15 value 39.086758
## iter  20 value 35.726119
## iter  25 value 28.329909
## iter  30 value 22.975542
## iter  35 value 19.446347
## iter  40 value 17.493380
## iter  45 value 15.388138
## iter  50 value 13.732536
## final  value 13.732536
## stopped after 50 iterations
```



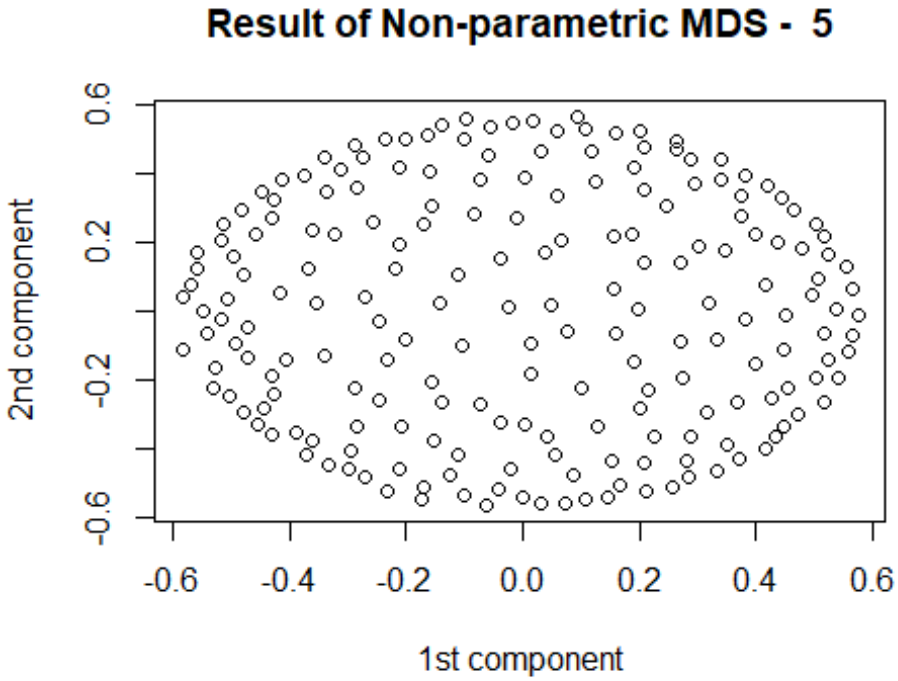

```
## Stress for  2 th iteration:  13.73254initial  value 42.750591
## iter   5 value 41.580957
## iter  10 value 40.434989
## iter  15 value 38.226909
## iter  20 value 29.256158
## iter  25 value 19.959781
## iter  30 value 16.331637
## iter  35 value 13.632103
## iter  40 value 12.522541
## iter  45 value 12.111644
## final  value 11.858405
## converged
```



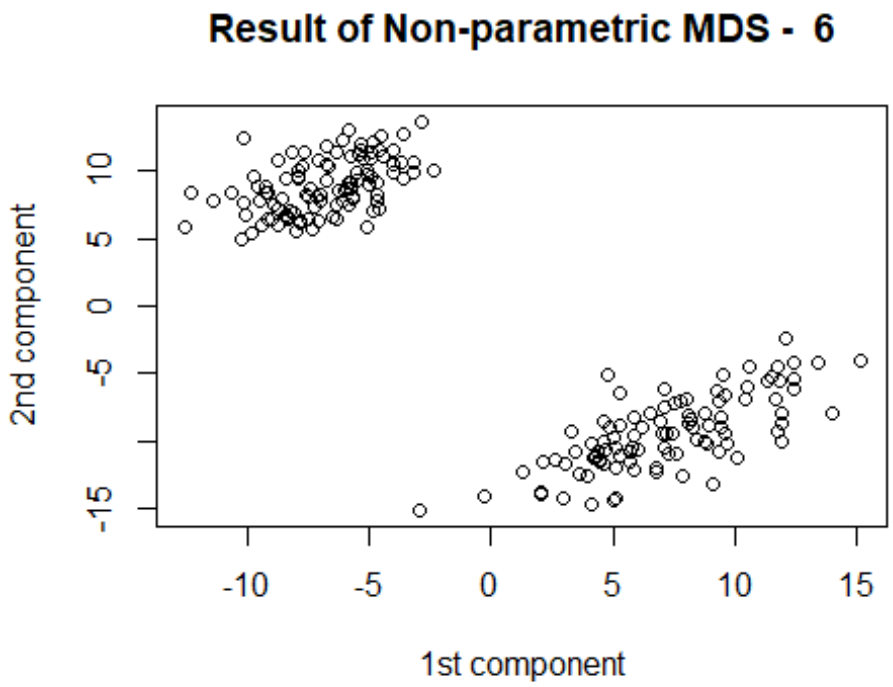

```
## Stress for  3 th iteration:  11.85841initial  value 42.940739
## iter   5 value 41.589356
## iter  10 value 39.480641
## iter  15 value 35.914108
## iter  20 value 25.072993
## iter  25 value 17.130764
## iter  30 value 14.390847
## iter  35 value 12.968520
## iter  40 value 12.026409
## iter  45 value 11.666159
## final  value 11.641099
## converged
```



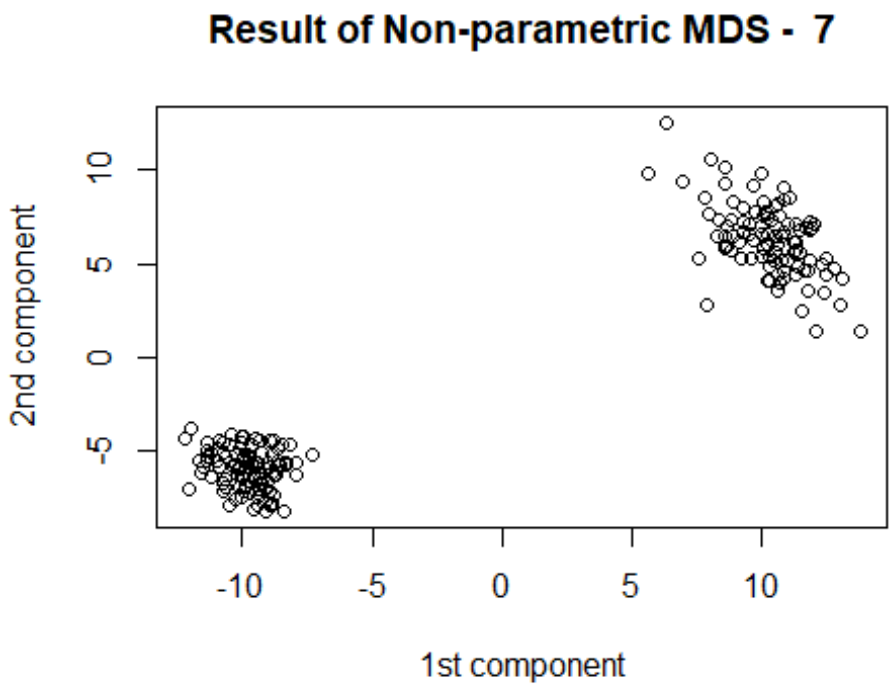

```
## Stress for  4 th iteration:  11.6411initial  value 42.741937
## final  value 41.686800
## converged
```



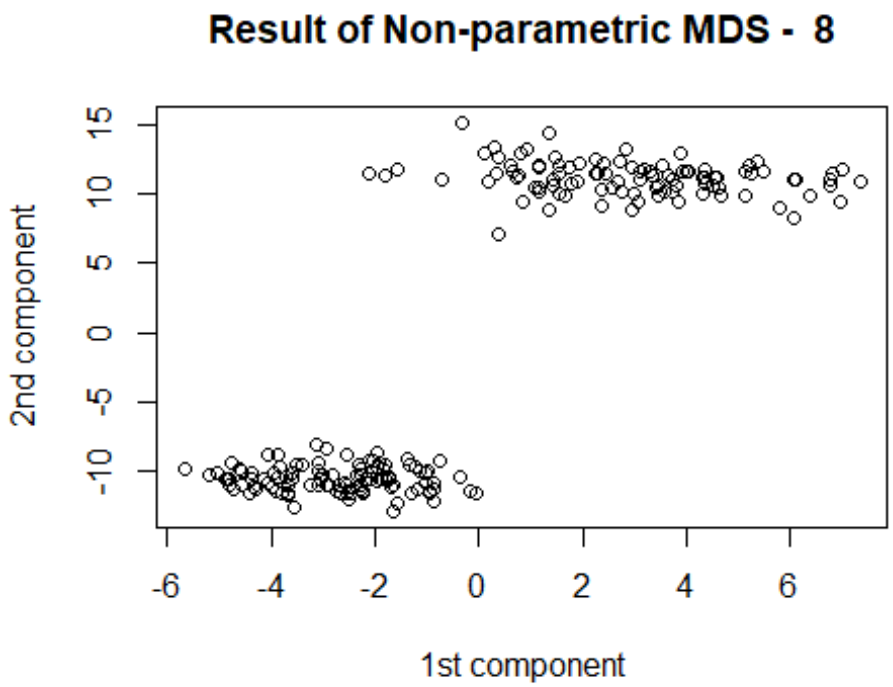
```
## Stress for  5 th iteration:  41.6868initial  value 42.687586
## iter   5 value 41.291632
## iter  10 value 39.485178
## iter  15 value 39.178577
## iter  20 value 38.580936
## iter  25 value 33.047247
## iter  30 value 25.826733
## iter  35 value 21.806066
## iter  40 value 18.522455
## iter  45 value 16.421340
## iter  50 value 14.652942
## final  value 14.652942
## stopped after 50 iterations
```



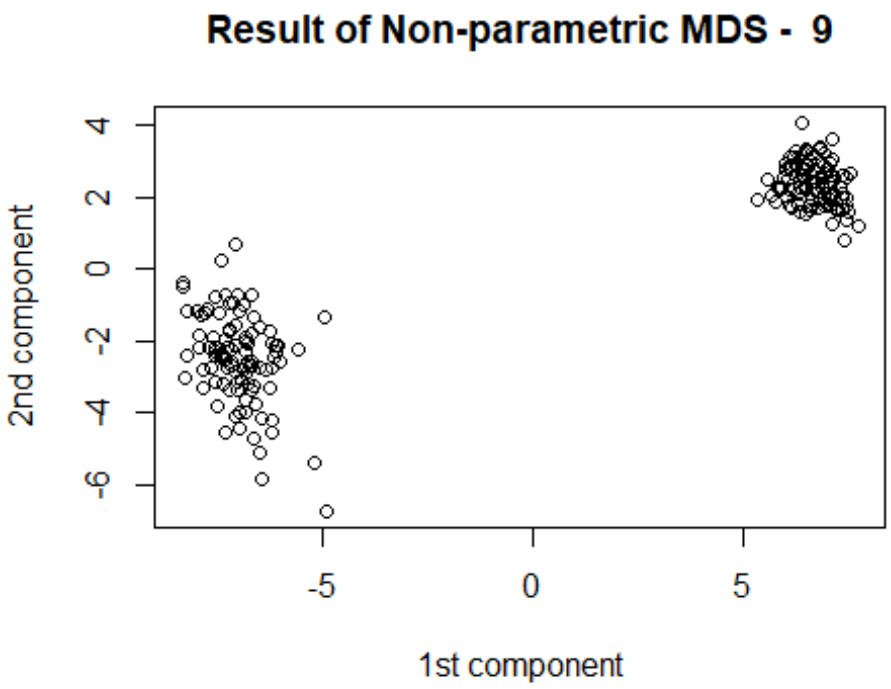

```
## Stress for  6 th iteration:  14.65294initial  value 42.749593
## iter   5 value 41.590515
## iter  10 value 39.943722
## iter  15 value 38.789425
## iter  20 value 31.501041
## iter  25 value 21.182051
## iter  30 value 17.654690
## iter  35 value 13.611497
## iter  40 value 12.643228
## iter  45 value 12.129782
## iter  50 value 11.739946
## final  value 11.739946
## stopped after 50 iterations
```



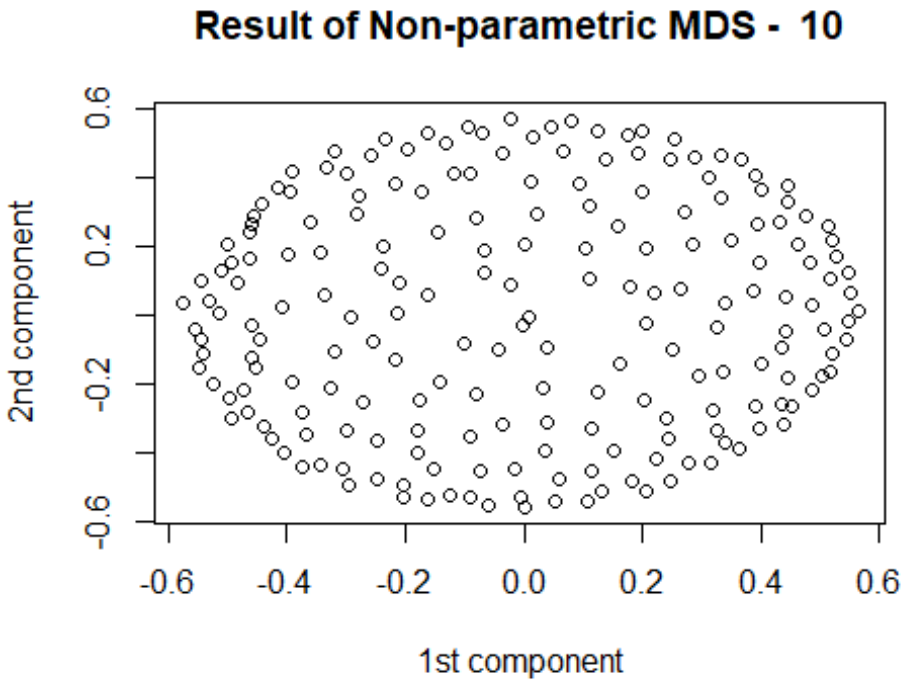

```
## Stress for  7 th iteration:  11.73995initial  value 43.053665
## iter   5 value 41.485155
## iter  10 value 39.959837
## iter  15 value 39.034109
## iter  20 value 32.921849
## iter  25 value 23.583706
## iter  30 value 19.632836
## iter  35 value 16.152837
## iter  40 value 14.599163
## iter  45 value 13.076166
## iter  50 value 12.076263
## final  value 12.076263
## stopped after 50 iterations
```




```
## Stress for  8 th iteration:  12.07626initial  value 43.172234
## iter   5 value 41.629487
## iter  10 value 40.637651
## iter  15 value 28.073191
## iter  20 value 17.935615
## iter  25 value 13.762613
## iter  30 value 12.505731
## iter  35 value 12.051862
## iter  40 value 11.824039
## final  value 11.724180
## converged
```




```
## Stress for  9 th iteration:  11.72418initial  value 42.787973
## final  value 41.714115
## converged
```




```
## Stress for  10 th iteration:  41.71411
```

What we observe is that 2 different patters emerge in the runs using different initial configurations. Therefore, based on the initial configuration we have different outcomes if whether the population is homogeneous or not. It's worth noting that the results with 2 clusters have a lower stress.

9

9. Compute the stress for a 1, 2, 3, . . . , 50-dimensional solution. How many dimensions are necessary to obtain a good representation with a stress below 10? Make a plot of the stress against the number of dimensions.

The number of dimensions necessary to obtain a good representation is 13. Give the non-increasing behaviour of the plot, after this value the stress is getting smaller and smaller.

```
#set.seed(12345)

dimensions <-50
stress_list <- NULL
for(i in 1:dimensions) {
  #init <- scale(matrix(runif(i*n),ncol=i),scale=FALSE)
  stress_list[i] <- isoMDS(D,k=i)$stress
}
```

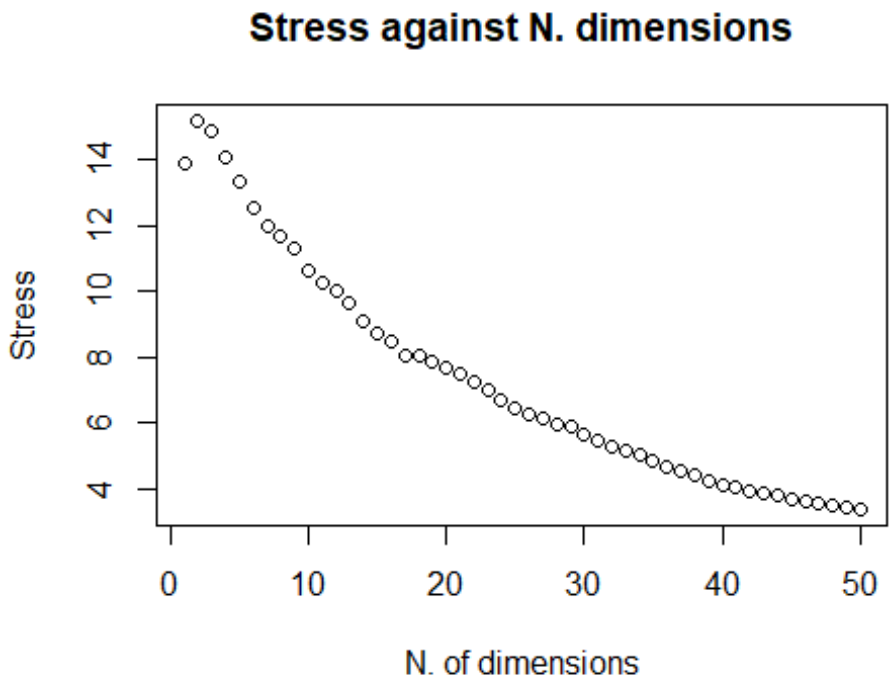
```
## initial  value 13.906634
## final  value 13.906634
## converged
## initial  value 15.185076
## final  value 15.185076
## converged
## initial  value 14.843856
## final  value 14.843856
## converged
## initial  value 14.078522
## final  value 14.078522
## converged
## initial  value 13.351683
## final  value 13.351683
## converged
## initial  value 12.525247
## final  value 12.525247
## converged
## initial  value 11.974448
## final  value 11.974448
## converged
## initial  value 11.662172
## final  value 11.662172
## converged
## initial  value 11.306433
## final  value 11.306433
## converged
## initial  value 10.617137
```

```
## final  value 10.617137
## converged
## initial  value 10.275785
## final  value 10.275785
## converged
## initial  value 10.008590
## final  value 10.008590
## converged
## initial  value 9.614630
## final  value 9.614630
## converged
## initial  value 9.108197
## final  value 9.108197
## converged
## initial  value 8.707142
## final  value 8.707142
## converged
## initial  value 8.451913
## final  value 8.451913
## converged
## initial  value 8.062375
## final  value 8.062375
## converged
## initial  value 8.047934
## final  value 8.047934
## converged
## initial  value 7.864988
## final  value 7.864988
## converged
## initial  value 7.683222
## final  value 7.683222
## converged
## initial  value 7.468312
## final  value 7.468312
## converged
## initial  value 7.218013
## final  value 7.218013
## converged
## initial  value 6.963368
## final  value 6.963368
## converged
## initial  value 6.659217
## final  value 6.659217
## converged
## initial  value 6.458941
## final  value 6.458941
## converged
## initial  value 6.233994
## final  value 6.233994
## converged
## initial  value 6.117082
```

```
## final  value 6.117082
## converged
## initial  value 5.924375
## final  value 5.924375
## converged
## initial  value 5.861823
## final  value 5.861823
## converged
## initial  value 5.610832
## final  value 5.610832
## converged
## initial  value 5.449734
## final  value 5.449734
## converged
## initial  value 5.269790
## final  value 5.269790
## converged
## initial  value 5.139134
## final  value 5.139134
## converged
## initial  value 4.996730
## final  value 4.996729
## converged
## initial  value 4.811151
## final  value 4.811151
## converged
## initial  value 4.640897
## final  value 4.640897
## converged
## initial  value 4.546486
## final  value 4.546486
## converged
## initial  value 4.390268
## final  value 4.390267
## converged
## initial  value 4.242612
## final  value 4.242612
## converged
## initial  value 4.118732
## final  value 4.118732
## converged
## initial  value 4.039682
## final  value 4.039682
## converged
## initial  value 3.939259
## final  value 3.939259
## converged
## initial  value 3.829421
## final  value 3.829421
## converged
## initial  value 3.759687
```

```
## final  value 3.759687
## converged
## initial  value 3.686426
## final  value 3.686426
## converged
## initial  value 3.588795
## final  value 3.588795
## converged
## initial  value 3.532327
## final  value 3.532327
## converged
## initial  value 3.471093
## final  value 3.471093
## converged
## initial  value 3.430112
## final  value 3.430112
## converged
## initial  value 3.360582
## final  value 3.360582
## converged
```

```
plot(1:dimensions, stress_list, xlab="N. of dimensions", ylab="Stress", main="Stre
ss against N. dimensions")
```




```
list_dimensions<-1:50
list_dimensions[stress_list<10] #n. dimensions for stress<10
```

```
##  [1] 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
## [26] 38 39 40 41 42 43 44 45 46 47 48 49 50
```

10

10. Run the two-dimensional isoMDS a hundred times, each time using a different random initial configuration using the instructions above. Report the stress of the best and the worse run, and plot the corresponding maps. Compare your results to the metric MDS and comment on your findings.

Stress of worst run: 42.90679 Stress of best run: 11.43862

```
n_iter <- 100
stress_list <- NULL
non_params_mds_list <- list()

for (i in seq(1:n_iter)) {
  init <- scale(matrix(runif(m * n), ncol = m), scale = FALSE)
  non_param_mds <- isoMDS(D, y = init, k = m)
  stress_list[i] <- non_param_mds$stress
  non_params_mds_list[[i]] <- non_param_mds

  cat("Stress for ", i, "th iteration: ", stress_list[i], "\n")
}
```

```
## initial  value 43.564259
## iter   5 value 41.543462
## iter  10 value 40.032698
## iter  15 value 37.982675
## iter  20 value 30.530080
## iter  25 value 22.857289
## iter  30 value 18.045406
## iter  35 value 14.378411
## iter  40 value 13.640402
## iter  45 value 12.869677
## iter  50 value 11.984643
## final  value 11.984643
## stopped after 50 iterations
## Stress for  1 th iteration:  11.98464
## initial  value 43.154912
## iter   5 value 41.656934
## iter   5 value 41.626314
## iter   5 value 41.623995
## final  value 41.623995
## converged
## Stress for  2 th iteration:  41.62399
## initial  value 42.847441
## final  value 41.647872
## converged
## Stress for  3 th iteration:  41.64787
## initial  value 42.441838
```

```
## final  value 41.668189
## converged
## Stress for  4 th iteration:  41.66819
## initial  value 42.547112
## iter   5 value 41.544287
## iter  10 value 40.156972
## iter  15 value 37.103475
## iter  20 value 25.608132
## iter  25 value 19.171616
## iter  30 value 14.716695
## iter  35 value 12.958316
## iter  40 value 11.994017
## iter  45 value 11.656369
## iter  45 value 11.649638
## iter  45 value 11.645642
## final  value 11.645642
## converged
## Stress for  5 th iteration:  11.64564
## initial  value 43.488921
## iter   5 value 41.573662
## iter  10 value 39.802111
## iter  15 value 38.342548
## iter  20 value 29.262644
## iter  25 value 19.134296
## iter  30 value 16.227120
## iter  35 value 13.250100
## iter  40 value 12.310201
## iter  45 value 12.071257
## final  value 11.734351
## converged
## Stress for  6 th iteration:  11.73435
## initial  value 42.694560
## iter   5 value 41.679947
## iter   5 value 41.674565
## iter   5 value 41.674565
## final  value 41.674565
## converged
## Stress for  7 th iteration:  41.67456
## initial  value 43.234830
## final  value 41.663354
## converged
## Stress for  8 th iteration:  41.66335
## initial  value 42.738511
## iter   5 value 40.729410
## iter  10 value 39.503204
## iter  15 value 37.027112
## iter  20 value 30.883757
## iter  25 value 25.296826
## iter  30 value 20.882295
## iter  35 value 17.378017
## iter  40 value 16.469532
```

```
## iter  45 value 14.126420
## iter  50 value 12.548321
## final  value 12.548321
## stopped after 50 iterations
## Stress for  9 th iteration:  12.54832
## initial  value 42.868715
## iter   5 value 41.518366
## iter  10 value 39.297759
## iter  15 value 31.073124
## iter  20 value 20.430374
## iter  25 value 13.948410
## iter  30 value 12.403626
## iter  35 value 11.893926
## final  value 11.671211
## converged
## Stress for  10 th iteration:  11.67121
## initial  value 43.341356
## final  value 42.197447
## converged
## Stress for  11 th iteration:  42.19745
## initial  value 42.697035
## final  value 41.667837
## converged
## Stress for  12 th iteration:  41.66784
## initial  value 42.794806
## iter   5 value 41.670063
## final  value 41.564516
## converged
## Stress for  13 th iteration:  41.56452
## initial  value 42.818422
## iter   5 value 41.691132
## iter   5 value 41.670627
## iter   5 value 41.670627
## final  value 41.670627
## converged
## Stress for  14 th iteration:  41.67063
## initial  value 42.512275
## iter   5 value 41.401087
## iter  10 value 39.657915
## iter  15 value 37.366792
## iter  20 value 30.578684
## iter  25 value 25.497959
## iter  30 value 19.782798
## iter  35 value 15.544443
## iter  40 value 14.668594
## iter  45 value 13.330450
## iter  50 value 12.714543
## final  value 12.714543
## stopped after 50 iterations
## Stress for  15 th iteration:  12.71454
## initial  value 42.617771
```

```
## iter   5 value 41.592516
## final  value 41.477708
## converged
## Stress for  16 th iteration:  41.47771
## initial  value 42.883990
## final  value 41.684567
## converged
## Stress for  17 th iteration:  41.68457
## initial  value 42.849136
## iter   5 value 41.265947
## iter  10 value 39.684725
## iter  15 value 38.708275
## iter  20 value 31.844894
## iter  25 value 22.903387
## iter  30 value 18.998567
## iter  35 value 14.852591
## iter  40 value 13.415760
## iter  45 value 12.573658
## iter  50 value 12.052875
## final  value 12.052875
## stopped after 50 iterations
## Stress for  18 th iteration:  12.05287
## initial  value 42.586987
## final  value 41.674662
## converged
## Stress for  19 th iteration:  41.67466
## initial  value 42.458387
## final  value 41.692680
## converged
## Stress for  20 th iteration:  41.69268
## initial  value 42.613554
## final  value 41.645665
## converged
## Stress for  21 th iteration:  41.64567
## initial  value 42.901908
## iter   5 value 41.578556
## iter  10 value 39.757976
## iter  15 value 38.602719
## iter  20 value 33.328801
## iter  25 value 26.778851
## iter  30 value 21.754625
## iter  35 value 16.811188
## iter  40 value 15.419023
## iter  45 value 13.385526
## iter  50 value 12.611282
## final  value 12.611282
## stopped after 50 iterations
## Stress for  22 th iteration:  12.61128
## initial  value 42.839946
## final  value 41.684079
## converged
```

```
## Stress for  23 th iteration:  41.68408
## initial  value 42.831669
## iter   5 value 41.522515
## iter  10 value 39.728382
## iter  15 value 36.909406
## iter  20 value 26.785712
## iter  25 value 18.732870
## iter  30 value 14.805752
## iter  35 value 13.131675
## iter  40 value 12.058027
## final  value 11.753963
## converged
## Stress for  24 th iteration:  11.75396
## initial  value 43.558389
## iter   5 value 41.631028
## iter  10 value 39.946889
## iter  15 value 37.271873
## iter  20 value 27.459453
## iter  25 value 17.728723
## iter  30 value 14.477758
## iter  35 value 12.409980
## iter  40 value 11.786684
## iter  45 value 11.654818
## final  value 11.630305
## converged
## Stress for  25 th iteration:  11.63031
## initial  value 42.704618
## final  value 41.672501
## converged
## Stress for  26 th iteration:  41.6725
## initial  value 42.952311
## final  value 42.949535
## converged
## Stress for  27 th iteration:  42.94953
## initial  value 42.741360
## iter   5 value 41.207473
## iter  10 value 39.485467
## iter  15 value 38.993369
## iter  20 value 33.128542
## iter  25 value 24.117494
## iter  30 value 19.541595
## iter  35 value 16.921086
## iter  40 value 15.593074
## iter  45 value 14.796532
## iter  50 value 13.955516
## final  value 13.955516
## stopped after 50 iterations
## Stress for  28 th iteration:  13.95552
## initial  value 43.117495
## iter   5 value 41.664809
## iter   5 value 41.633311
```

```
## iter   5 value 41.632936
## final  value 41.632936
## converged
## Stress for  29 th iteration:  41.63294
## initial  value 43.017313
## final  value 41.678412
## converged
## Stress for  30 th iteration:  41.67841
## initial  value 42.532853
## iter   5 value 41.672848
## iter   5 value 41.653039
## iter   5 value 41.652831
## final  value 41.652831
## converged
## Stress for  31 th iteration:  41.65283
## initial  value 42.917490
## iter   5 value 41.622532
## iter  10 value 39.895693
## iter  15 value 38.377814
## iter  20 value 25.506854
## iter  25 value 18.677537
## iter  30 value 15.931754
## iter  35 value 14.160077
## iter  40 value 12.609629
## iter  45 value 12.263492
## iter  50 value 11.854827
## final  value 11.854827
## stopped after 50 iterations
## Stress for  32 th iteration:  11.85483
## initial  value 43.370288
## final  value 42.075400
## converged
## Stress for  33 th iteration:  42.0754
## initial  value 42.851040
## iter   5 value 41.429750
## iter  10 value 39.548679
## iter  15 value 38.414506
## iter  20 value 27.225152
## iter  25 value 18.284947
## iter  30 value 13.648598
## iter  35 value 12.746153
## iter  40 value 11.812511
## iter  45 value 11.571602
## final  value 11.538144
## converged
## Stress for  34 th iteration:  11.53814
## initial  value 42.874068
## iter   5 value 41.703140
## iter   5 value 41.670168
## iter   5 value 41.670168
## final  value 41.670168
```

```
## converged
## Stress for  35 th iteration:  41.67017
## initial  value 42.907898
## iter   5 value 41.447682
## iter  10 value 39.585102
## iter  15 value 27.391815
## iter  20 value 17.017077
## iter  25 value 13.512567
## iter  30 value 12.327437
## iter  35 value 11.792758
## iter  40 value 11.672541
## final  value 11.646659
## converged
## Stress for  36 th iteration:  11.64666
## initial  value 43.053845
## iter   5 value 41.591973
## iter  10 value 39.648945
## iter  15 value 39.343855
## iter  20 value 37.402563
## iter  25 value 30.088321
## iter  30 value 22.908649
## iter  35 value 19.949935
## iter  40 value 17.523179
## iter  45 value 15.824363
## iter  50 value 13.712416
## final  value 13.712416
## stopped after 50 iterations
## Stress for  37 th iteration:  13.71242
## initial  value 42.751412
## iter   5 value 41.477124
## iter  10 value 39.660142
## iter  15 value 39.232727
## iter  20 value 37.987884
## iter  25 value 27.979224
## iter  30 value 20.834097
## iter  35 value 17.029963
## iter  40 value 15.885244
## iter  45 value 15.084569
## iter  50 value 13.493780
## final  value 13.493780
## stopped after 50 iterations
## Stress for  38 th iteration:  13.49378
## initial  value 42.722107
## iter   5 value 41.399542
## iter  10 value 37.514367
## iter  15 value 22.034550
## iter  20 value 14.773718
## iter  25 value 12.385977
## iter  30 value 11.726294
## iter  35 value 11.592924
## final  value 11.559267
```

```
## converged
## Stress for  39 th iteration:  11.55927
## initial  value 42.599590
## iter   5 value 41.365486
## iter  10 value 39.543764
## iter  15 value 38.581975
## iter  20 value 31.755754
## iter  25 value 23.222836
## iter  30 value 18.393975
## iter  35 value 16.496615
## iter  40 value 15.642166
## iter  45 value 14.751913
## iter  50 value 13.246714
## final  value 13.246714
## stopped after 50 iterations
## Stress for  40 th iteration:  13.24671
## initial  value 42.976891
## iter   5 value 41.582836
## iter  10 value 39.908388
## iter  15 value 39.182371
## iter  20 value 34.144425
## iter  25 value 24.150126
## iter  30 value 18.581022
## iter  35 value 15.872097
## iter  40 value 13.749784
## iter  45 value 12.592987
## iter  50 value 12.123582
## final  value 12.123582
## stopped after 50 iterations
## Stress for  41 th iteration:  12.12358
## initial  value 43.199547
## final  value 42.972384
## converged
## Stress for  42 th iteration:  42.97238
## initial  value 43.070101
## iter   5 value 41.292975
## iter  10 value 39.463295
## iter  15 value 36.406763
## iter  20 value 24.589776
## iter  25 value 17.094436
## iter  30 value 13.871259
## iter  35 value 12.820029
## iter  40 value 12.247055
## iter  45 value 11.974467
## iter  50 value 11.712635
## final  value 11.712635
## stopped after 50 iterations
## Stress for  43 th iteration:  11.71264
## initial  value 42.767876
## final  value 42.646690
## converged
```

```
## Stress for  44 th iteration:  42.64669
## initial  value 43.115372
## iter   5 value 41.511602
## iter  10 value 39.682913
## iter  15 value 38.497668
## iter  20 value 30.000041
## iter  25 value 22.254210
## iter  30 value 18.784033
## iter  35 value 15.081843
## iter  40 value 13.002703
## iter  45 value 12.422926
## iter  50 value 11.924657
## final  value 11.924657
## stopped after 50 iterations
## Stress for  45 th iteration:  11.92466
## initial  value 42.668020
## iter   5 value 41.472070
## iter  10 value 39.505689
## iter  15 value 34.048871
## iter  20 value 24.405753
## iter  25 value 18.382596
## iter  30 value 15.067207
## iter  35 value 13.426011
## iter  40 value 12.125235
## iter  45 value 11.938224
## final  value 11.668064
## converged
## Stress for  46 th iteration:  11.66806
## initial  value 43.095457
## final  value 42.204197
## converged
## Stress for  47 th iteration:  42.2042
## initial  value 43.442766
## iter   5 value 41.695998
## iter   5 value 41.658179
## iter   5 value 41.658179
## final  value 41.658179
## converged
## Stress for  48 th iteration:  41.65818
## initial  value 43.067472
## iter   5 value 41.679600
## iter   5 value 41.665518
## iter   5 value 41.665518
## final  value 41.665518
## converged
## Stress for  49 th iteration:  41.66552
## initial  value 42.764107
## iter   5 value 41.418358
## iter  10 value 39.509786
## iter  15 value 30.743161
## iter  20 value 19.320663
```

```
## iter  25 value 13.896249
## iter  30 value 12.420624
## iter  35 value 11.901089
## final  value 11.687951
## converged
## Stress for  50 th iteration:  11.68795
## initial  value 42.491769
## iter   5 value 41.574626
## iter  10 value 40.778843
## iter  15 value 34.249919
## iter  20 value 23.559608
## iter  25 value 16.256645
## iter  30 value 14.177083
## iter  35 value 12.459387
## iter  40 value 12.164171
## iter  45 value 11.816949
## iter  45 value 11.807874
## iter  45 value 11.801831
## final  value 11.801831
## converged
## Stress for  51 th iteration:  11.80183
## initial  value 42.617879
## final  value 41.683884
## converged
## Stress for  52 th iteration:  41.68388
## initial  value 42.910124
## iter   5 value 41.653123
## iter  10 value 40.914907
## iter  15 value 32.840274
## iter  20 value 21.587321
## iter  25 value 15.285608
## iter  30 value 12.996897
## iter  35 value 12.200033
## iter  40 value 11.903554
## final  value 11.704327
## converged
## Stress for  53 th iteration:  11.70433
## initial  value 43.073653
## final  value 42.020934
## converged
## Stress for  54 th iteration:  42.02093
## initial  value 42.511143
## iter   5 value 41.588905
## iter  10 value 40.826899
## iter  15 value 34.861055
## iter  20 value 24.858867
## iter  25 value 16.895954
## iter  30 value 13.194124
## iter  35 value 12.061268
## iter  40 value 11.680442
## final  value 11.637569
```

```
## converged
## Stress for  55 th iteration:  11.63757
## initial  value 43.131124
## iter   5 value 41.629492
## iter  10 value 40.883322
## iter  15 value 39.074746
## iter  20 value 30.700506
## iter  25 value 19.876139
## iter  30 value 15.904782
## iter  35 value 13.922370
## iter  40 value 12.888413
## iter  45 value 12.347407
## iter  50 value 12.153927
## final  value 12.153927
## stopped after 50 iterations
## Stress for  56 th iteration:  12.15393
## initial  value 42.766132
## iter   5 value 41.627248
## iter  10 value 39.770663
## iter  15 value 37.927155
## iter  20 value 29.248022
## iter  25 value 21.577430
## iter  30 value 17.775906
## iter  35 value 14.518345
## iter  40 value 13.219674
## iter  45 value 12.531649
## final  value 11.767400
## converged
## Stress for  57 th iteration:  11.7674
## initial  value 42.505458
## final  value 41.678323
## converged
## Stress for  58 th iteration:  41.67832
## initial  value 43.050272
## iter   5 value 41.683914
## iter   5 value 41.658027
## iter   5 value 41.657956
## final  value 41.657956
## converged
## Stress for  59 th iteration:  41.65796
## initial  value 42.983531
## iter   5 value 41.541331
## iter  10 value 41.118910
## iter  15 value 39.290779
## iter  20 value 31.755319
## iter  25 value 21.784170
## iter  30 value 17.868773
## iter  35 value 14.555643
## iter  40 value 13.372594
## iter  45 value 12.441419
## iter  50 value 11.871535
```

```
## final  value 11.871535
## stopped after 50 iterations
## Stress for  60 th iteration:  11.87154
## initial  value 42.888072
## iter   5 value 41.643594
## iter  10 value 41.143197
## iter  15 value 39.017200
## iter  20 value 30.746888
## iter  25 value 20.413256
## iter  30 value 15.241800
## iter  35 value 13.058047
## iter  40 value 12.349343
## iter  45 value 11.829650
## iter  50 value 11.645683
## final  value 11.645683
## stopped after 50 iterations
## Stress for  61 th iteration:  11.64568
## initial  value 43.473279
## iter   5 value 41.562818
## iter  10 value 40.321377
## iter  15 value 37.752185
## iter  20 value 25.187587
## iter  25 value 16.462965
## iter  30 value 12.971973
## iter  35 value 12.162608
## iter  40 value 11.698631
## iter  45 value 11.548895
## final  value 11.483701
## converged
## Stress for  62 th iteration:  11.4837
## initial  value 42.955518
## iter   5 value 41.674600
## iter   5 value 41.634536
## iter   5 value 41.629677
## final  value 41.629677
## converged
## Stress for  63 th iteration:  41.62968
## initial  value 42.505686
## final  value 41.672933
## converged
## Stress for  64 th iteration:  41.67293
## initial  value 43.342419
## iter   5 value 41.684078
## iter   5 value 41.676996
## iter   5 value 41.676996
## final  value 41.676996
## converged
## Stress for  65 th iteration:  41.677
## initial  value 43.123031
## iter   5 value 41.675655
## iter   5 value 41.650914
```

```
## iter   5 value 41.650642
## final  value 41.650642
## converged
## Stress for  66 th iteration:  41.65064
## initial  value 42.742927
## iter   5 value 41.463794
## iter  10 value 38.440389
## iter  15 value 23.195298
## iter  20 value 13.436598
## iter  25 value 11.938178
## iter  30 value 11.666857
## iter  35 value 11.556916
## final  value 11.494065
## converged
## Stress for  67 th iteration:  11.49407
## initial  value 42.668951
## iter   5 value 41.657405
## iter  10 value 40.732403
## iter  15 value 39.285083
## iter  20 value 36.393152
## iter  25 value 29.904427
## iter  30 value 20.957130
## iter  35 value 16.962014
## iter  40 value 14.408381
## iter  45 value 13.906032
## iter  50 value 12.202272
## final  value 12.202272
## stopped after 50 iterations
## Stress for  68 th iteration:  12.20227
## initial  value 42.790421
## iter   5 value 41.549618
## iter  10 value 39.806957
## iter  15 value 39.175412
## iter  20 value 35.597693
## iter  25 value 26.877423
## iter  30 value 20.815572
## iter  35 value 17.631127
## iter  40 value 16.271181
## iter  45 value 15.010125
## iter  50 value 13.403397
## final  value 13.403397
## stopped after 50 iterations
## Stress for  69 th iteration:  13.4034
## initial  value 42.813294
## iter   5 value 41.664897
## iter   5 value 41.630099
## iter   5 value 41.628571
## final  value 41.628571
## converged
## Stress for  70 th iteration:  41.62857
## initial  value 42.638796
```

```
## iter   5 value 41.456723
## iter  10 value 39.426700
## iter  15 value 32.352647
## iter  20 value 22.052027
## iter  25 value 15.131286
## iter  30 value 12.225224
## iter  35 value 11.798223
## iter  40 value 11.667291
## final  value 11.633369
## converged
## Stress for  71 th iteration:  11.63337
## initial  value 42.850635
## iter   5 value 41.643887
## iter  10 value 41.348925
## iter  15 value 39.538339
## iter  20 value 32.169509
## iter  25 value 19.497408
## iter  30 value 14.415617
## iter  35 value 12.851811
## iter  40 value 12.168866
## iter  45 value 11.896734
## final  value 11.738899
## converged
## Stress for  72 th iteration:  11.7389
## initial  value 43.516729
## iter   5 value 41.647869
## iter   5 value 41.623119
## iter   5 value 41.623119
## final  value 41.623119
## converged
## Stress for  73 th iteration:  41.62312
## initial  value 43.089146
## final  value 41.706210
## converged
## Stress for  74 th iteration:  41.70621
## initial  value 42.530999
## iter   5 value 41.686357
## iter   5 value 41.671314
## iter   5 value 41.671314
## final  value 41.671314
## converged
## Stress for  75 th iteration:  41.67131
## initial  value 42.786220
## iter   5 value 41.398146
## iter   5 value 41.365893
## iter  10 value 39.745445
## iter  15 value 39.328885
## iter  20 value 38.704478
## iter  25 value 27.424491
## iter  30 value 17.625610
## iter  35 value 13.844263
```

```
## iter  40 value 12.473521
## iter  45 value 11.749001
## iter  50 value 11.611477
## final  value 11.611477
## stopped after 50 iterations
## Stress for  76 th iteration:  11.61148
## initial  value 42.751946
## iter   5 value 41.518733
## iter  10 value 39.559496
## iter  15 value 35.835554
## iter  20 value 25.869862
## iter  25 value 17.666688
## iter  30 value 13.976113
## iter  35 value 12.836939
## iter  40 value 12.234038
## iter  45 value 11.932432
## final  value 11.734118
## converged
## Stress for  77 th iteration:  11.73412
## initial  value 43.242020
## iter   5 value 41.688557
## iter   5 value 41.673446
## iter   5 value 41.673446
## final  value 41.673446
## converged
## Stress for  78 th iteration:  41.67345
## initial  value 42.970185
## iter   5 value 41.316492
## iter  10 value 39.181119
## iter  15 value 32.772307
## iter  20 value 24.137440
## iter  25 value 16.238509
## iter  30 value 13.769250
## iter  35 value 12.388229
## iter  40 value 11.999669
## final  value 11.676714
## converged
## Stress for  79 th iteration:  11.67671
## initial  value 42.830224
## iter   5 value 40.935422
## iter  10 value 39.138860
## iter  15 value 34.372449
## iter  20 value 25.512514
## iter  25 value 16.303621
## iter  30 value 14.706918
## iter  35 value 12.507738
## iter  40 value 12.087496
## iter  45 value 11.905595
## final  value 11.717343
## converged
## Stress for  80 th iteration:  11.71734
```

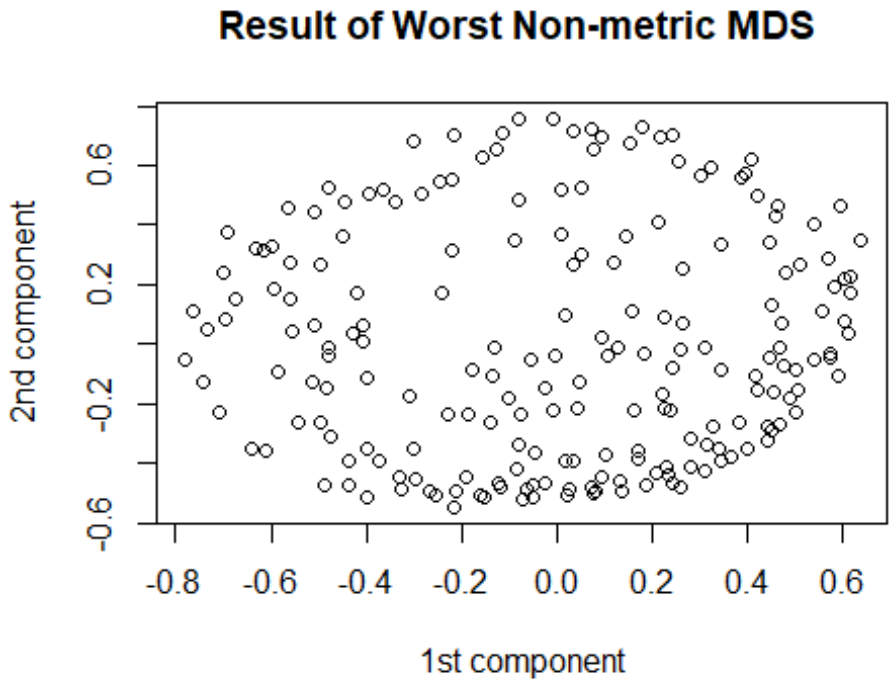
```
## initial  value 42.779989
## iter   5 value 41.542890
## iter  10 value 39.490864
## iter  15 value 37.082882
## iter  20 value 30.202408
## iter  25 value 24.167591
## iter  30 value 20.447899
## iter  35 value 17.211609
## iter  40 value 14.447102
## iter  45 value 13.093839
## iter  50 value 11.876137
## final  value 11.876137
## stopped after 50 iterations
## Stress for  81 th iteration:  11.87614
## initial  value 42.741804
## iter   5 value 41.689965
## iter   5 value 41.681337
## iter   5 value 41.681337
## final  value 41.681337
## converged
## Stress for  82 th iteration:  41.68134
## initial  value 42.764531
## final  value 41.641021
## converged
## Stress for  83 th iteration:  41.64102
## initial  value 42.845060
## iter   5 value 41.501059
## iter  10 value 40.064799
## iter  15 value 39.176944
## iter  20 value 36.708657
## iter  25 value 28.917963
## iter  30 value 23.065923
## iter  35 value 18.572192
## iter  40 value 16.813176
## iter  45 value 14.433141
## iter  50 value 13.348179
## final  value 13.348179
## stopped after 50 iterations
## Stress for  84 th iteration:  13.34818
## initial  value 42.974752
## iter   5 value 41.553990
## iter  10 value 40.319523
## iter  15 value 39.150809
## iter  20 value 35.315126
## iter  25 value 27.431629
## iter  30 value 20.784361
## iter  35 value 16.309524
## iter  40 value 14.966538
## iter  45 value 13.735357
## iter  50 value 12.524443
## final  value 12.524443
```

```
## stopped after 50 iterations
## Stress for  85 th iteration:  12.52444
## initial  value 42.911476
## final  value 41.693425
## converged
## Stress for  86 th iteration:  41.69342
## initial  value 42.555177
## iter   5 value 41.137447
## iter  10 value 39.426768
## iter  15 value 36.008708
## iter  20 value 28.417994
## iter  25 value 20.889961
## iter  30 value 17.880288
## iter  35 value 14.423375
## iter  40 value 13.668810
## iter  45 value 12.485955
## iter  50 value 11.820043
## final  value 11.820043
## stopped after 50 iterations
## Stress for  87 th iteration:  11.82004
## initial  value 42.988933
## final  value 41.667139
## converged
## Stress for  88 th iteration:  41.66714
## initial  value 42.622388
## iter   5 value 41.602089
## iter  10 value 40.095073
## iter  15 value 38.532978
## iter  20 value 29.127832
## iter  25 value 20.706735
## iter  30 value 17.341126
## iter  35 value 14.242974
## iter  40 value 13.146108
## iter  45 value 12.421764
## iter  50 value 11.943074
## final  value 11.943074
## stopped after 50 iterations
## Stress for  89 th iteration:  11.94307
## initial  value 43.062676
## iter   5 value 41.648695
## iter  10 value 41.315186
## iter  15 value 34.236052
## iter  20 value 17.881158
## iter  25 value 13.434041
## iter  30 value 12.222659
## iter  35 value 11.849403
## iter  40 value 11.736610
## final  value 11.679134
## converged
## Stress for  90 th iteration:  11.67913
## initial  value 42.828198
```

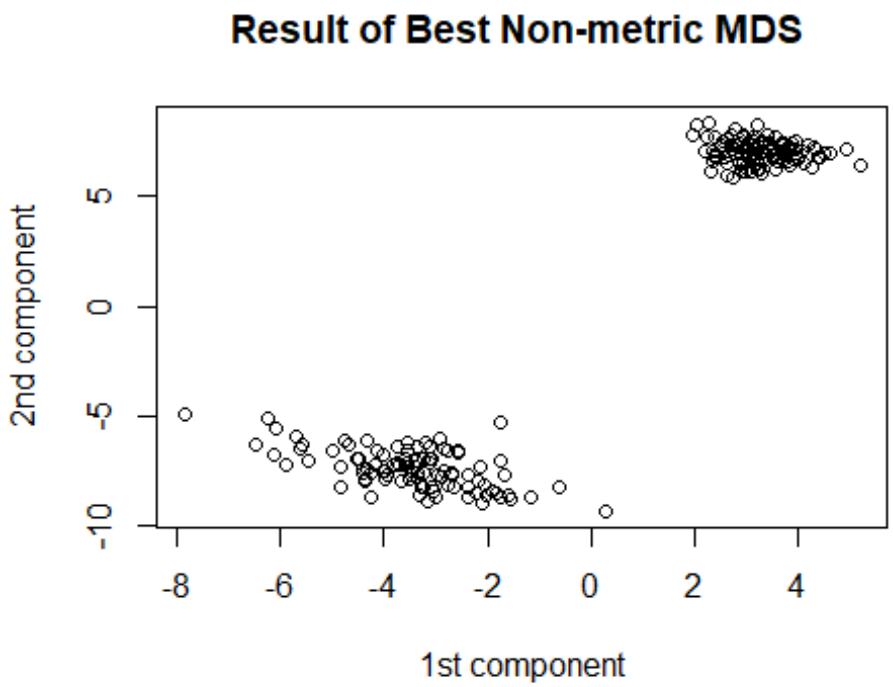
```
## iter   5 value 41.059144
## iter  10 value 39.270591
## iter  15 value 34.926100
## iter  20 value 27.984556
## iter  25 value 22.051742
## iter  30 value 16.977004
## iter  35 value 14.805640
## iter  40 value 13.289459
## iter  45 value 12.120086
## iter  50 value 11.778131
## final  value 11.778131
## stopped after 50 iterations
## Stress for  91 th iteration:  11.77813
## initial  value 42.549231
## final  value 42.031624
## converged
## Stress for  92 th iteration:  42.03162
## initial  value 43.022650
## iter   5 value 41.664095
## iter   5 value 41.623708
## iter   5 value 41.623188
## final  value 41.623188
## converged
## Stress for  93 th iteration:  41.62319
## initial  value 42.750377
## iter   5 value 41.635996
## iter  10 value 41.026187
## iter  15 value 38.988838
## iter  20 value 30.758468
## iter  25 value 22.513293
## iter  30 value 16.530867
## iter  35 value 14.122990
## iter  40 value 12.981354
## iter  45 value 12.458543
## iter  50 value 12.010741
## final  value 12.010741
## stopped after 50 iterations
## Stress for  94 th iteration:  12.01074
## initial  value 42.430652
## iter   5 value 41.227456
## iter  10 value 39.132013
## iter  15 value 29.283460
## iter  20 value 20.333394
## iter  25 value 15.352267
## iter  30 value 13.076396
## iter  35 value 12.096127
## iter  40 value 11.674610
## iter  45 value 11.565337
## final  value 11.526704
## converged
## Stress for  95 th iteration:  11.5267
```

```
## initial  value 42.936950
## iter   5 value 41.655262
## iter   5 value 41.638172
## iter   5 value 41.638099
## final  value 41.638099
## converged
## Stress for  96 th iteration:  41.6381
## initial  value 42.728200
## iter   5 value 41.572028
## iter  10 value 39.639192
## iter  15 value 37.558899
## iter  20 value 24.314829
## iter  25 value 16.342264
## iter  30 value 13.597598
## iter  35 value 12.232831
## iter  40 value 11.880618
## final  value 11.651820
## converged
## Stress for  97 th iteration:  11.65182
## initial  value 43.114456
## iter   5 value 41.515694
## iter  10 value 39.790691
## iter  15 value 37.860198
## iter  20 value 28.663161
## iter  25 value 20.146571
## iter  30 value 15.464908
## iter  35 value 12.005909
## iter  40 value 11.688739
## final  value 11.581079
## converged
## Stress for  98 th iteration:  11.58108
## initial  value 42.698949
## iter   5 value 41.626810
## iter  10 value 40.920406
## iter  15 value 39.551196
## iter  20 value 38.508583
## iter  25 value 25.706182
## iter  30 value 18.515358
## iter  35 value 15.430208
## iter  40 value 13.414929
## iter  45 value 12.628183
## iter  50 value 12.130910
## final  value 12.130910
## stopped after 50 iterations
## Stress for  99 th iteration:  12.13091
## initial  value 42.741704
## final  value 41.680981
## converged
## Stress for  100 th iteration:  41.68098
```

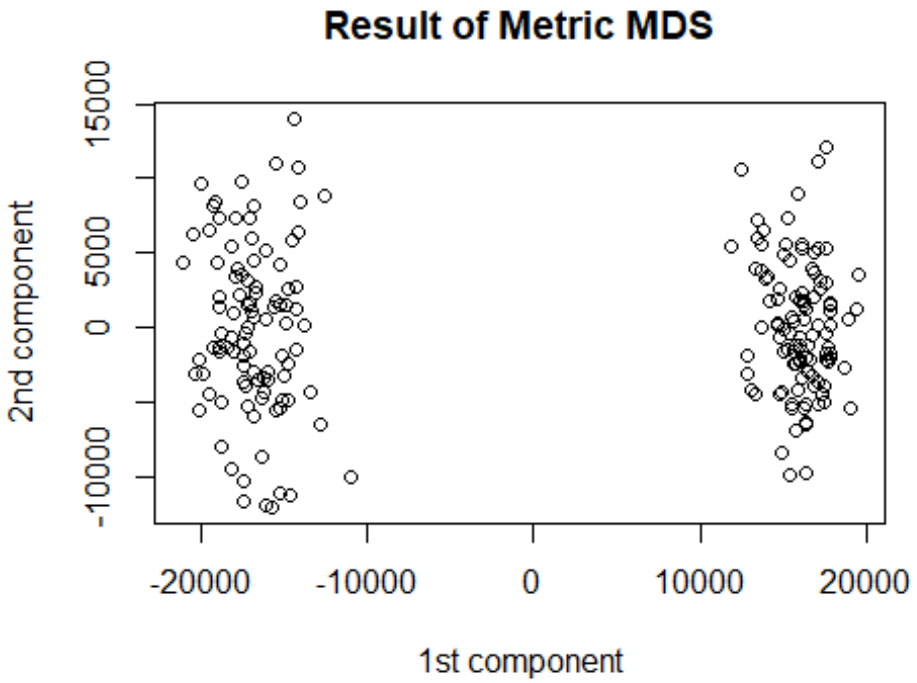
```
list_runs <- 1:n_iter
worst_run <- list_runs[stress_list == max(stress_list)]
best_run <- list_runs[stress_list == min(stress_list)]

cat("Stress of worst run: ", stress_list[worst_run])

## Stress of worst run:  42.97238

cat("Stress of best run: ", stress_list[best_run])

## Stress of best run:  11.4837

# plot worst run representation
plot(non_params_mds_list[[worst_run]]$points[, 1], non_params_mds_list[[worst_run]
]$points[, 2],
     xlab = "1st component", ylab = "2nd component", main = "Result of Worst Non-m
etric MDS ")
```




```
# plot best run representation
plot(non_params_mds_list[[best_run]]$points[, 1], non_params_mds_list[[best_run]]$
points[, 2],
     xlab = "1st component", ylab = "2nd component", main = "Result of Best Non-me
tric MDS ")
```




```
# Metric MDS
metric_mds <- cmdscale(D, k = m)

# Report stress of metric MDS
metric_stress <- sum((D - dist(metric_mds))^2)

## Warning in D - dist(metric_mds): la lunghezza più lunga dell'oggetto non è un
## multiplo della lunghezza più corta dell'oggetto

cat("Stress of Metric MDS: ", metric_stress, "\n")

## Stress of Metric MDS:  7.121805e+13

# Plot again maps for metric MDS for faster comparison
plot(metric_mds[, 1], metric_mds[, 2],
     xlab = "1st component", ylab = "2nd component", main = "Result of Metric MDS"
)
```



Metric and best non-metric mappings are strongly correlated. They both show that the dataset is heterogeneous. In particular, they both show there are 2 clusters well-divided. The main differences are the scale of the components in the different map, and the directions of maximum explained variance in the clusters of the 2 maps.

11

11. Compute the correlation matrix between the first two dimensions of the metric MDS and the two-dimensional solution of your best non-metric MDS. Comment your findings.

```
# create a data frame to store the first two dimensions of metric and non-metric MDS
p <- data.frame(
  pc1_metric = metric_mds[, 1],
  pc2_metric = metric_mds[, 2],
  pc1_n_metric = non_params_mds_list[[best_run]]$points[, 1],
  pc2_n_metric = non_params_mds_list[[best_run]]$points[, 2]
)

#compute the correlation matrix
corr_matrix <- cor(p)

print(corr_matrix)
```

```
##                pc1_metric    pc2_metric pc1_n_metric pc2_n_metric
## pc1_metric    1.000000e+00 -6.702176e-17   0.95472789   0.99566188
## pc2_metric   -6.702176e-17  1.000000e+00  -0.06874971   0.02506743
## pc1_n_metric  9.547279e-01 -6.874971e-02   1.00000000   0.93762408
## pc2_n_metric  9.956619e-01  2.506743e-02   0.93762408   1.00000000
```

```
cat("Correlation between pc1_metric and pc1_n_metric:", corr_matrix[1, 1], "\n")
```

```
## Correlation between pc1_metric and pc1_n_metric: 1
```

```
cat("Correlation between pc1_metric and pc2_n_metric:", corr_matrix[1, 2], "\n")
```

```
## Correlation between pc1_metric and pc2_n_metric: -6.702176e-17
```

```
cat("Correlation between pc2_metric and pc1_n_metric:", corr_matrix[2, 1], "\n")
```

```
## Correlation between pc2_metric and pc1_n_metric: -6.702176e-17
```

```
cat("Correlation between pc2_metric and pc2_n_metric:", corr_matrix[2, 2], "\n")
```

```
## Correlation between pc2_metric and pc2_n_metric: 1
```

Dimensions 1 and 2 of the metric MDS show a strong positive correlation with dimensions 1 and 2 of the non-metric MDS, respectively. The near-zero correlations with the cross-dimensions (pc1_metric vs pc2_n_metric and pc2_metric vs pc1_n_metric) suggest that these dimensions are relatively independent of each other.