BSG-MDS Practical 3 Statistical Genetics

Biel Caballero and Gerard Gomez

07/11/2023, submission deadline 14/11/2023

Population substructure

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

```
dimensions = dim(chr21)
print(paste0("There are ",dimensions[2]-6," variants"))
```

```
## [1] "There are 138106 variants"
```

```
print(paste0("A total of ", sum(is.na(chr21))/(dimensions[1]*dimensions[2]), " percent of the data is mi
```

```
## [1] "A total of 0 percent of the data is missing"
```

2. Compute the Manhattan distance matrix between the individuals (which is identical to the Minkowsky distance with parameter lambda = 1) using R function dist. Include a submatrix of dimension 5 by 5 with the distances between the first 5 individuals in your report.

```
dist <- dist(chr21[,7:ncol(chr21)],method = "manhattan")
dist <- as.matrix(dist)
dist[1:5,1:5]
```

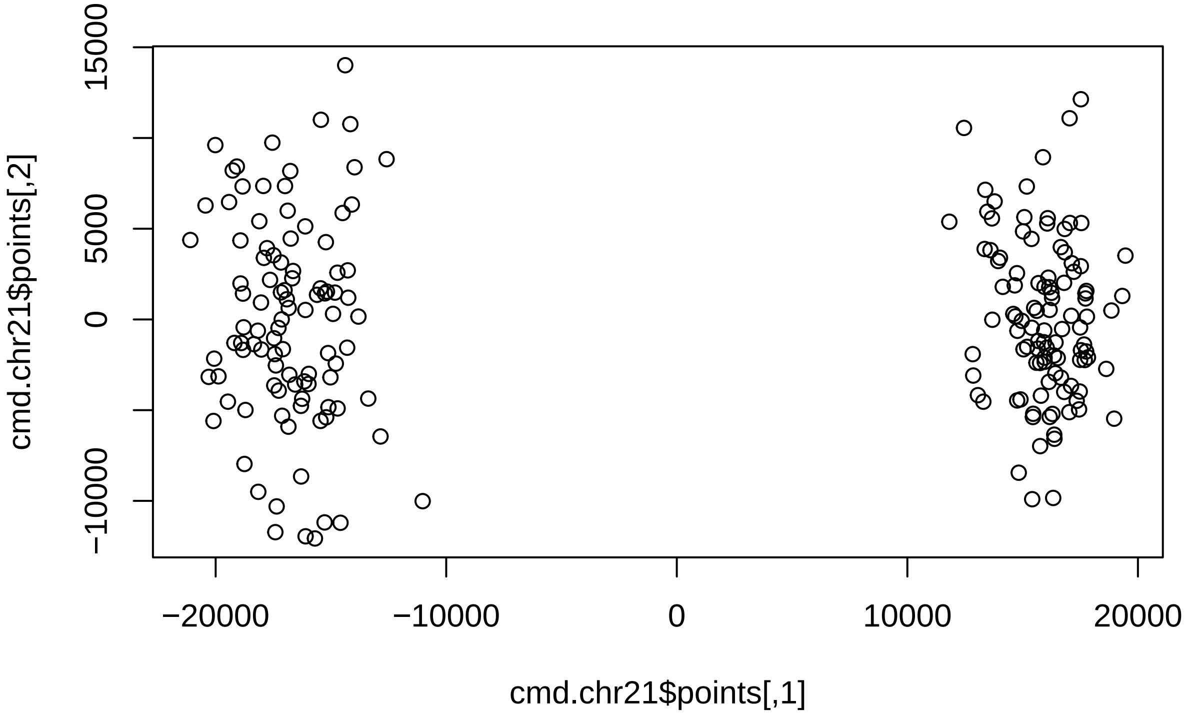
```
##       1     2     3     4     5
## 1     0 53495 55007 58174 53794
## 2 53495     0 55372 55995 55699
## 3 55007 55372     0 54815 55683
## 4 58174 55995 54815     0 59046
## 5 53794 55699 55683 59046     0
```

3. How does the Manhattan distance relate to the allele sharing distance?

The lower the distance the more alleles are shared

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?

```
cmd.chr21 <- cmdscale(dist,k=2,eig = TRUE)
plot(cmd.chr21$points)
```



We can clearly see that the data doesn't come from a homogeneous population. We can see that there are 2 subpopulations.

```r
subpop <- numeric(nrow(data))
subpop[cmd.chr21$points[,1]>0] <- 1
print(paste0("We can say that one subpopulation has a size of ", sum(subpop==1), "and subpopulation 2 ha
```

```
## [1] "We can say that one subpopulation has a size of 104and subpopulation 2 has a size of 99"
```

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

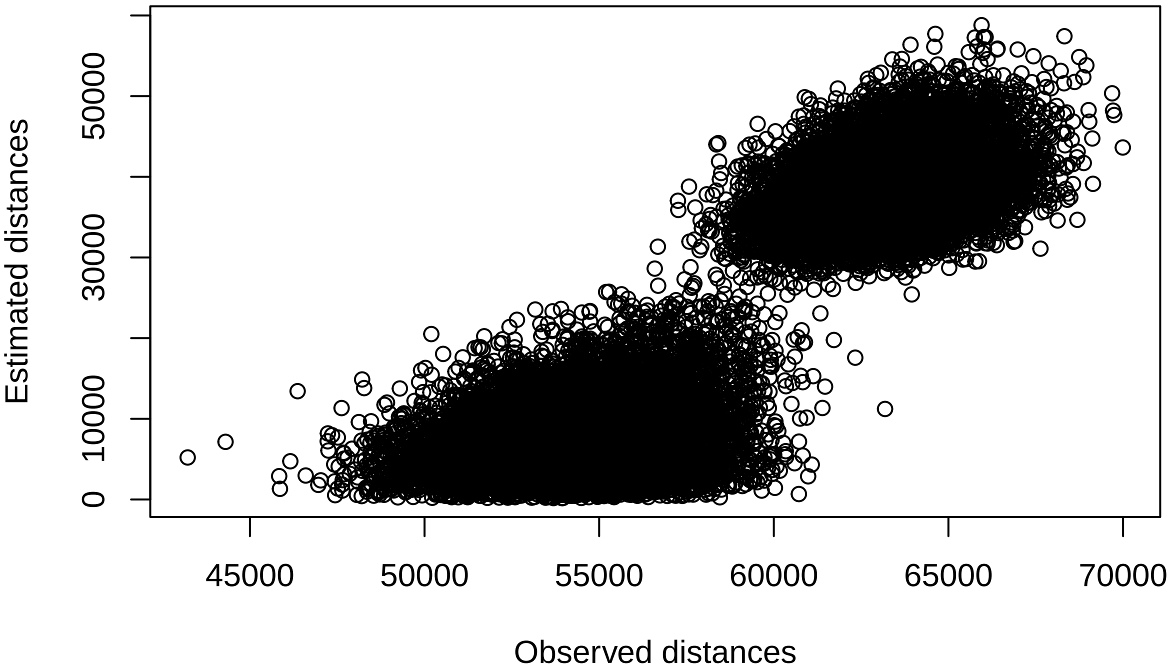
```r
cmd.chr21$GOF
```

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

As we obtain low values of goodness of fit it implies the cmd has a good fit. The criterium used is the STRESS.

6. Make a plot of the estimated distances (according to your two-dimensional 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).

```r
cdm.dist <- as.matrix(dist(cmd.chr21$points,method = "manhattan"))

dist.cmds <- cdm.dist[upper.tri(cdm.dist)]
dist.chr21 <- dist[upper.tri(dist)]

plot(dist.chr21,dist.cmds,xlab="Observed distances",ylab="Estimated distances")
```



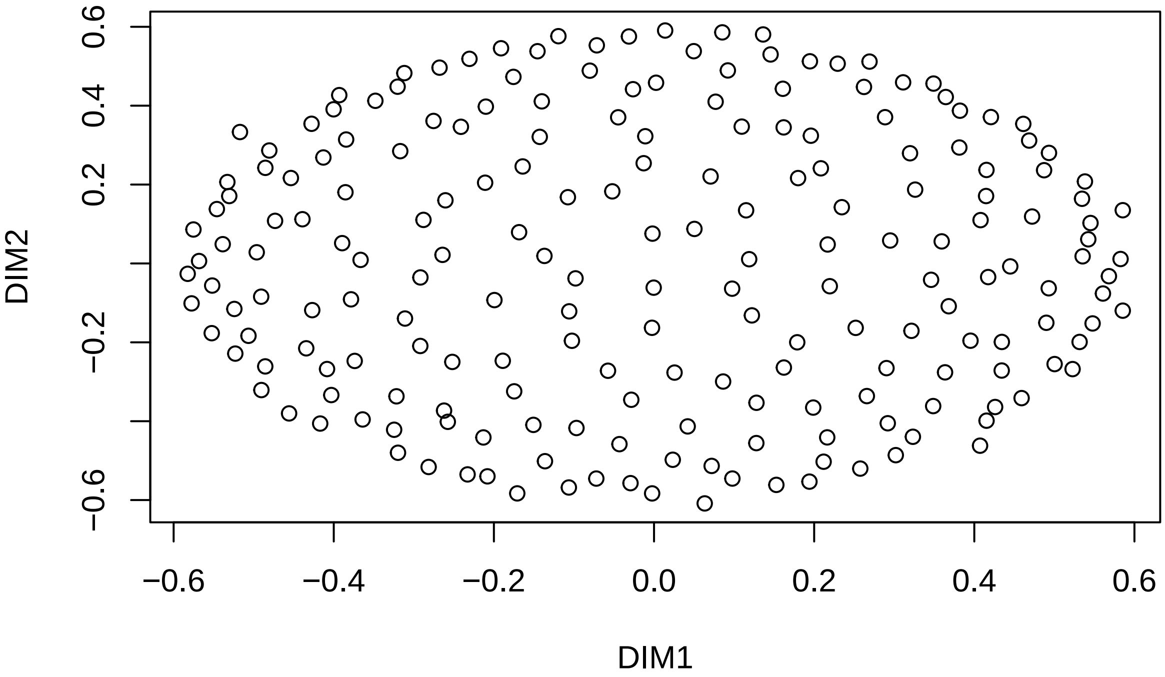
```
mod <- lm(dist.cmds~dist.chr21)
print(paste0("The coefficient of detrmination of the generated linear model is ",summary(mod)$r.squared
```

```
## [1] "The coefficient of detrmination of the generated linear model is 0.824095948197013"
```

Looking at the plot we can see that there are 2 clear clusters. These two clustres are separed mainly by the Estimated distances. This two clusters are generated because close points are going to have a lower distance that highly seppared points. We can see that the observed distances also respect this distances, as data from differnt clusters are always plotted further away, but the estimated makes it more clear.

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?

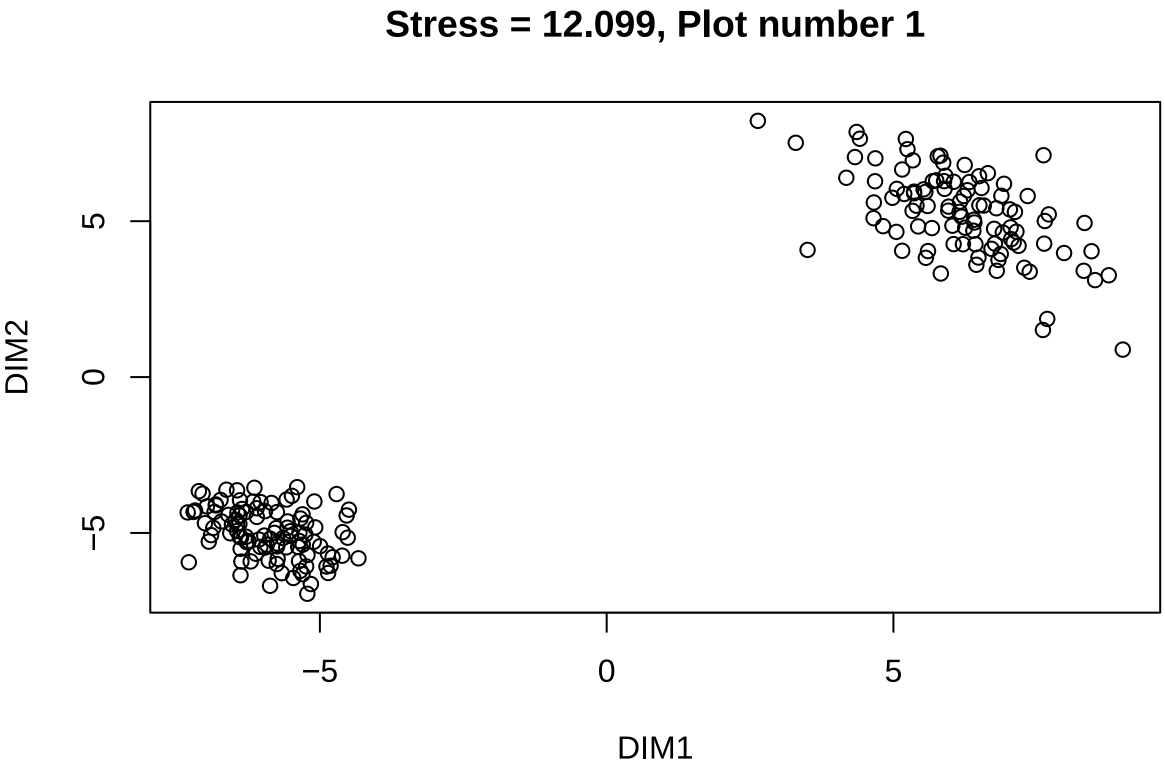
```
set.seed(12345)
m<-2 #as there are two subpopulations
n<-dimensions[1]
init <- scale(matrix(runif(m*n),ncol=m),scale=FALSE)
nonmms<-isoMDS(dist,y=init,k=m, trace = FALSE)

plot(nonmms$points[,1],nonmms$points[,2], xlab = "DIM1", ylab = "DIM2")
```

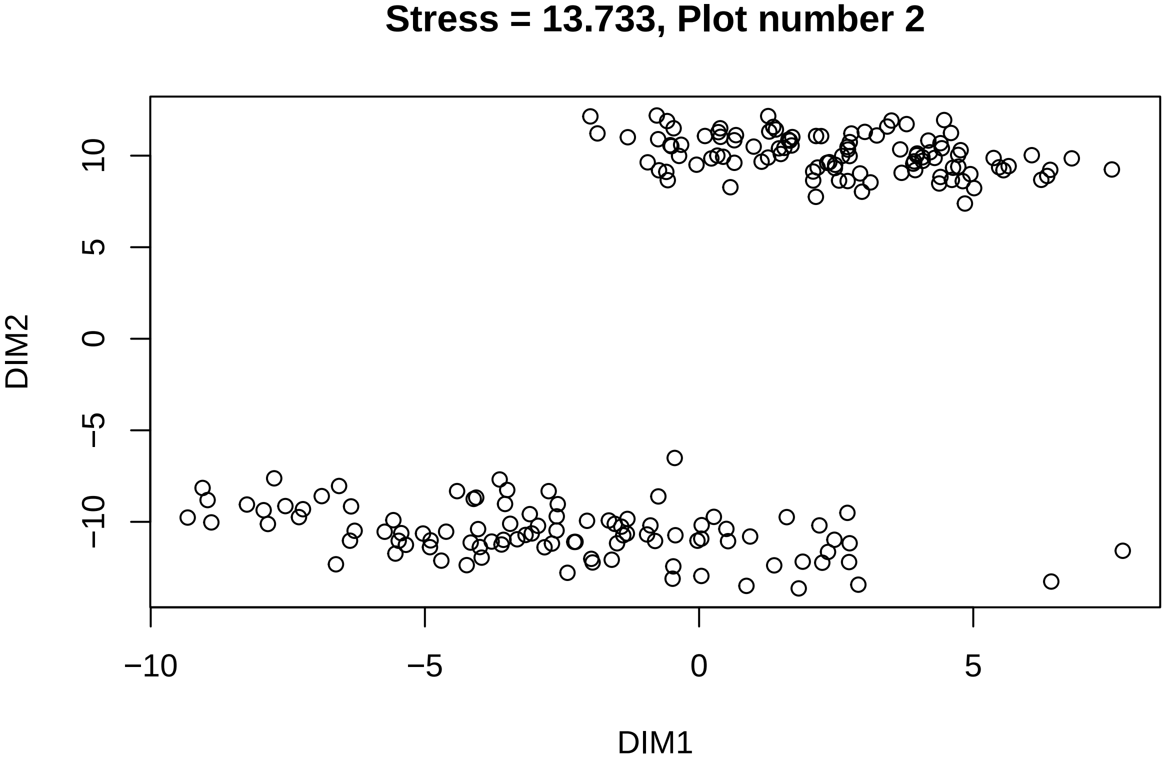


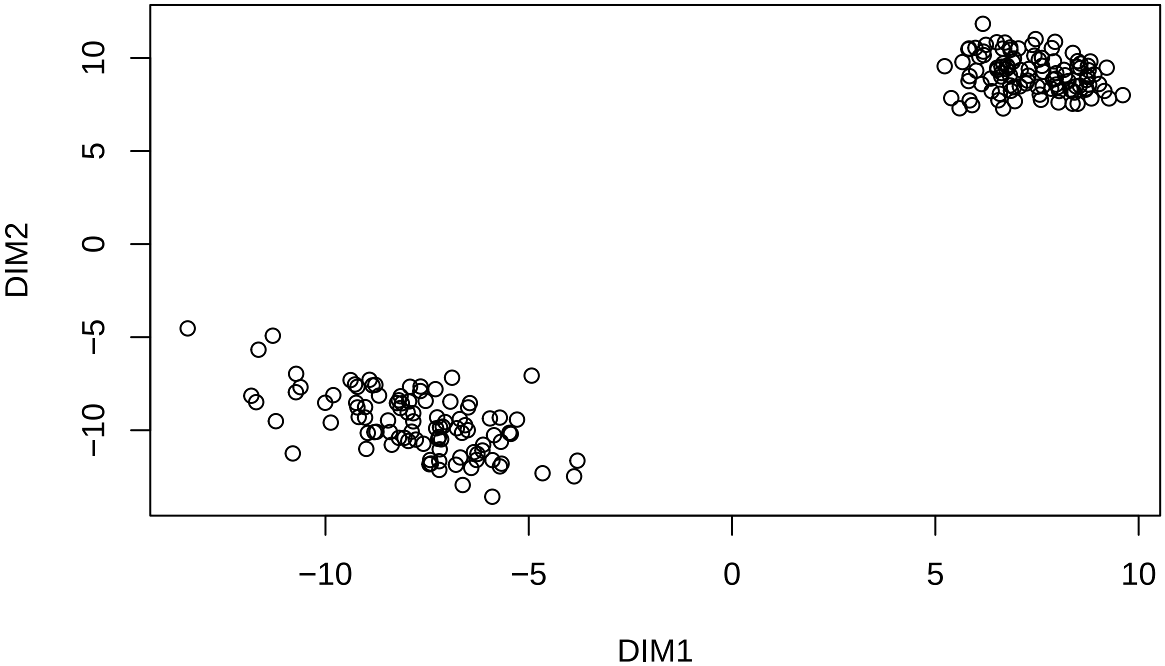
The plot confirms that the data come from one homogeneous population, as there can not be seen any separation of the population.

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?

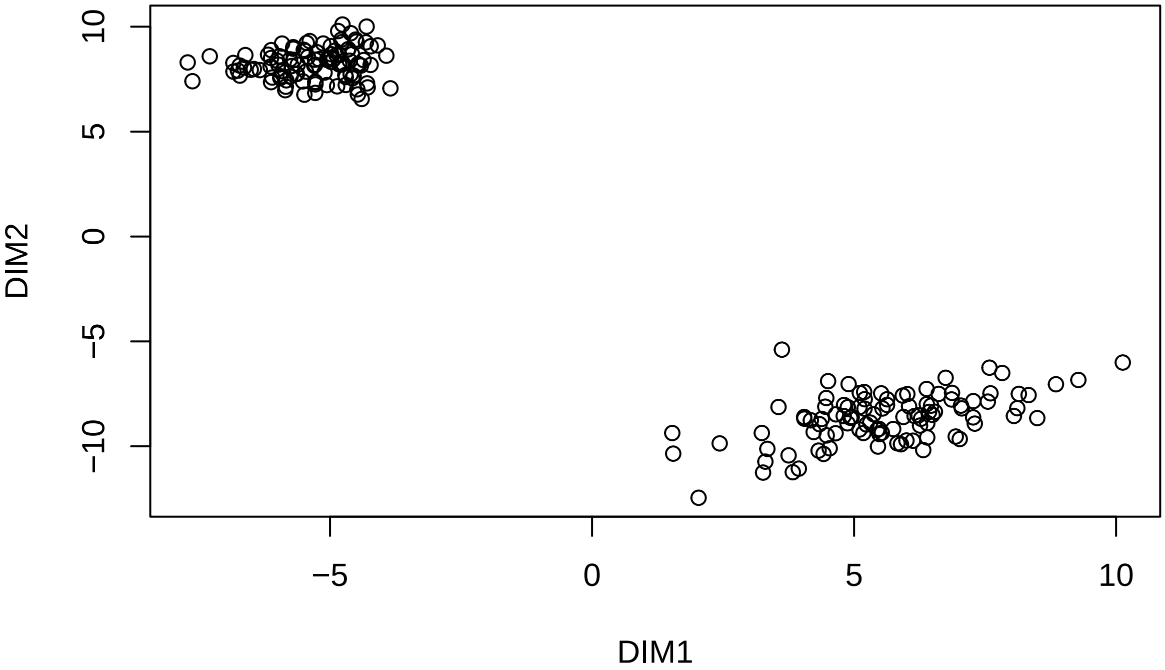
```
for (x in 1:10){
  init <- scale(matrix(runif(m*n),ncol=m),scale=FALSE)
  nonmms<-isoMDS(dist,y=init,k=m, trace = FALSE)
  plot(nonmms$points[,1],nonmms$points[,2], xlab = "DIM1", ylab = "DIM2", main = paste0('Stress = ',rou
}
```


Stress = 12.099, Plot number 1
DIM2
5
0
−5
−5
0
5
DIM1


Stress = 13.733, Plot number 2
DIM2
10
5
0
−5
−10
−10
−5
0
5
DIM1

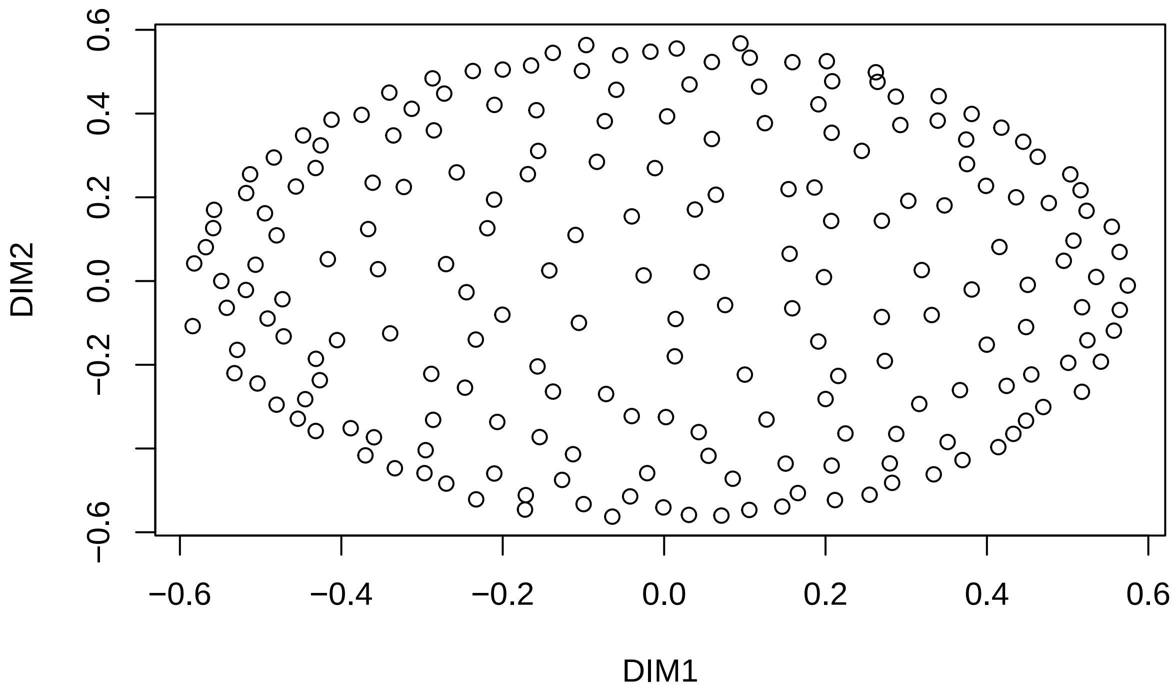


Stress = 11.858, Plot number 3

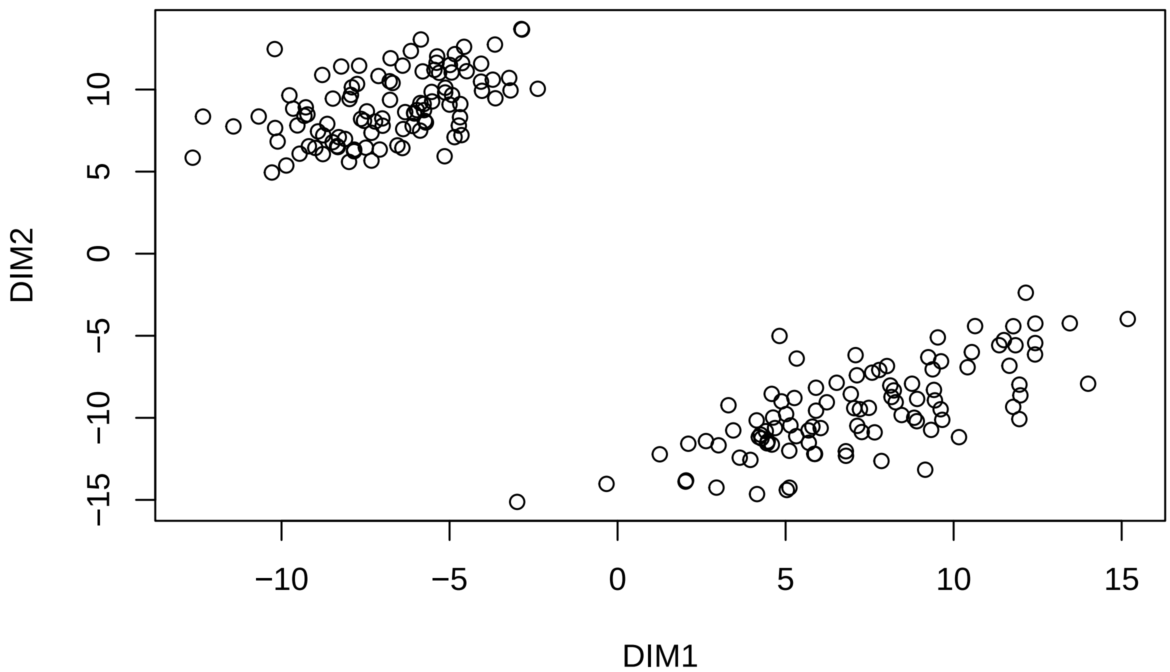


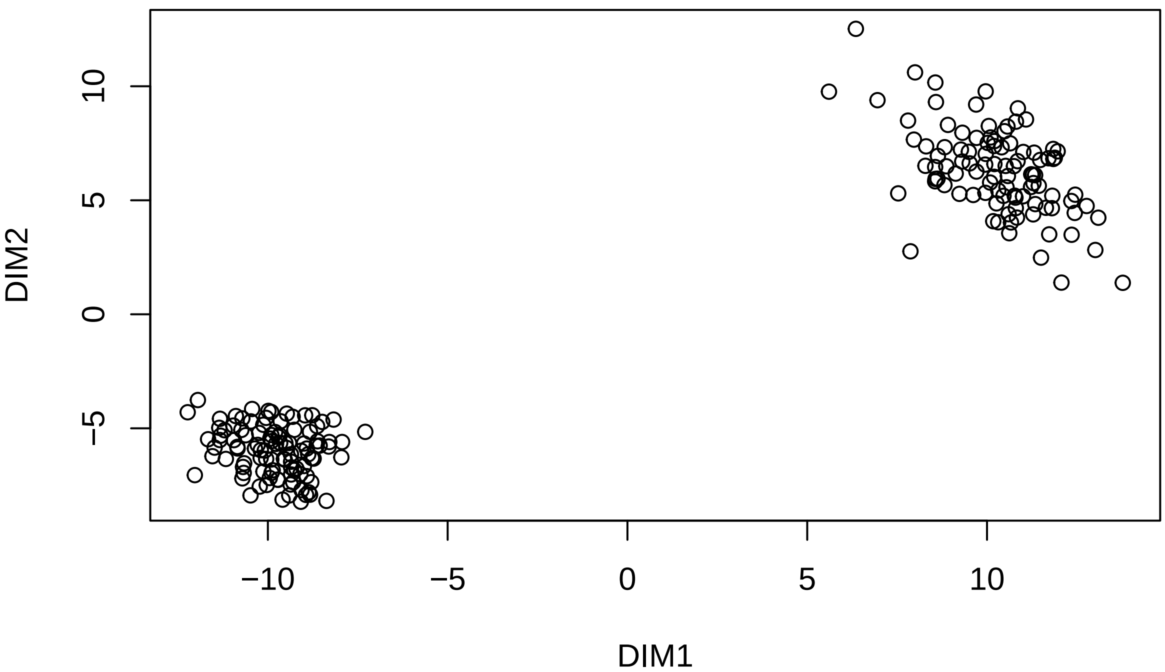
Stress = 11.641, Plot number 4

Stress = 41.687, Plot number 5

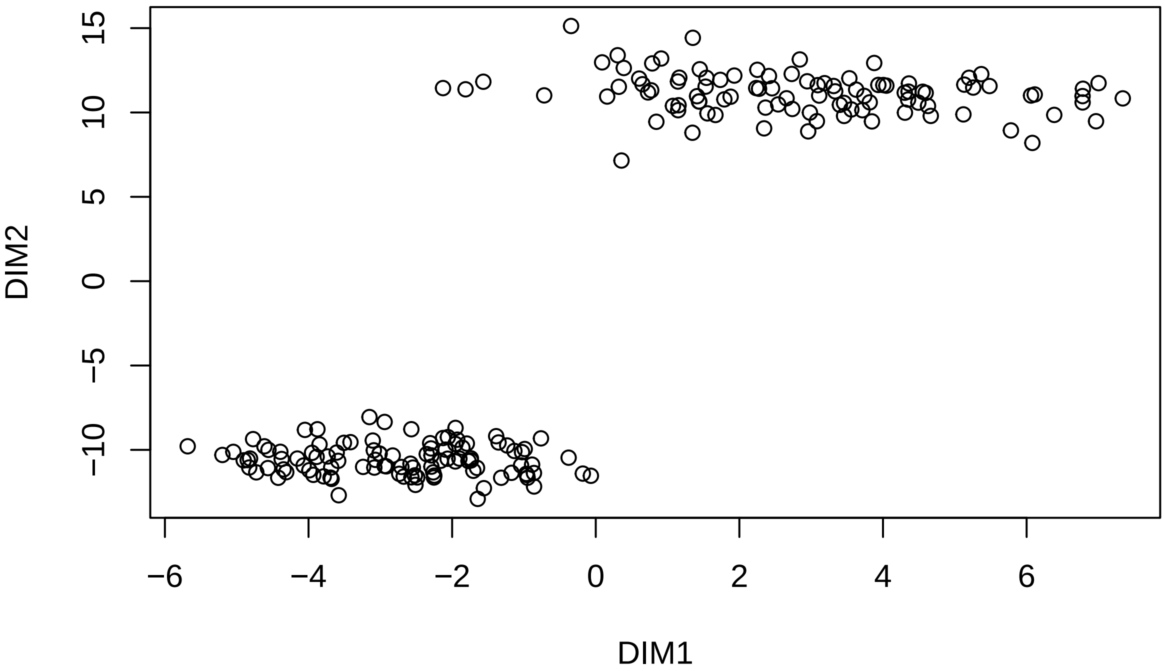


Stress = 14.653, Plot number 6

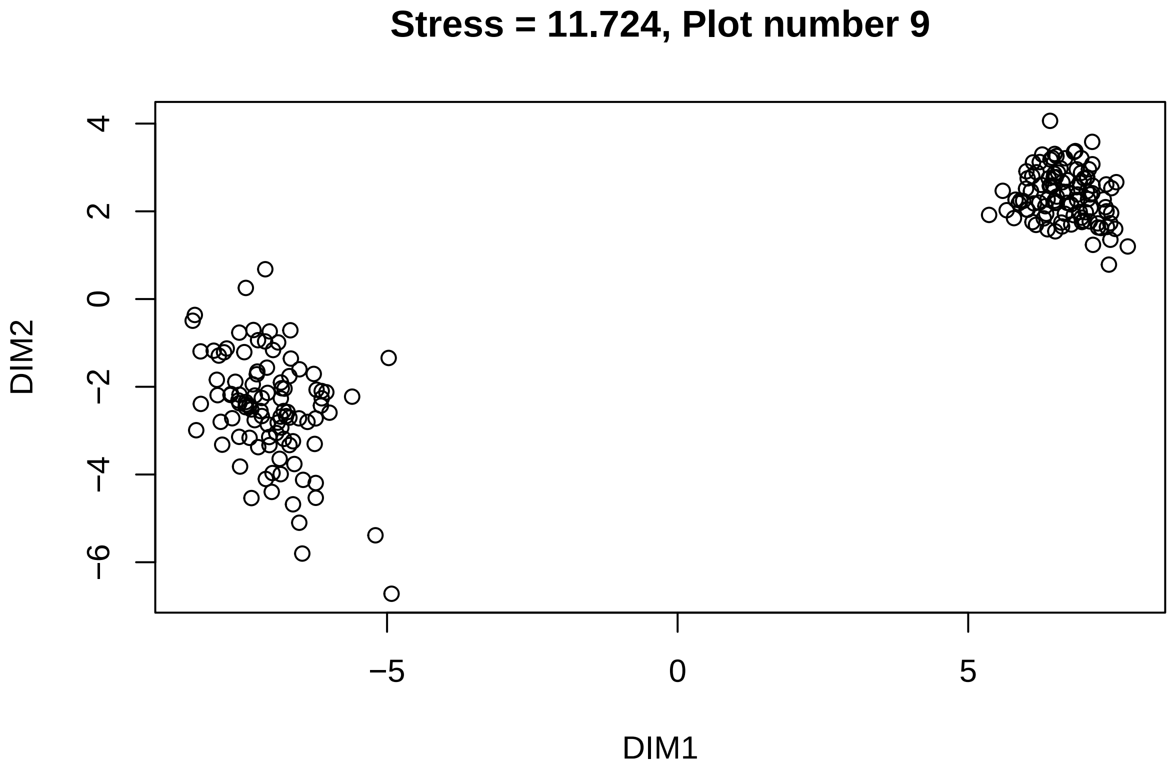


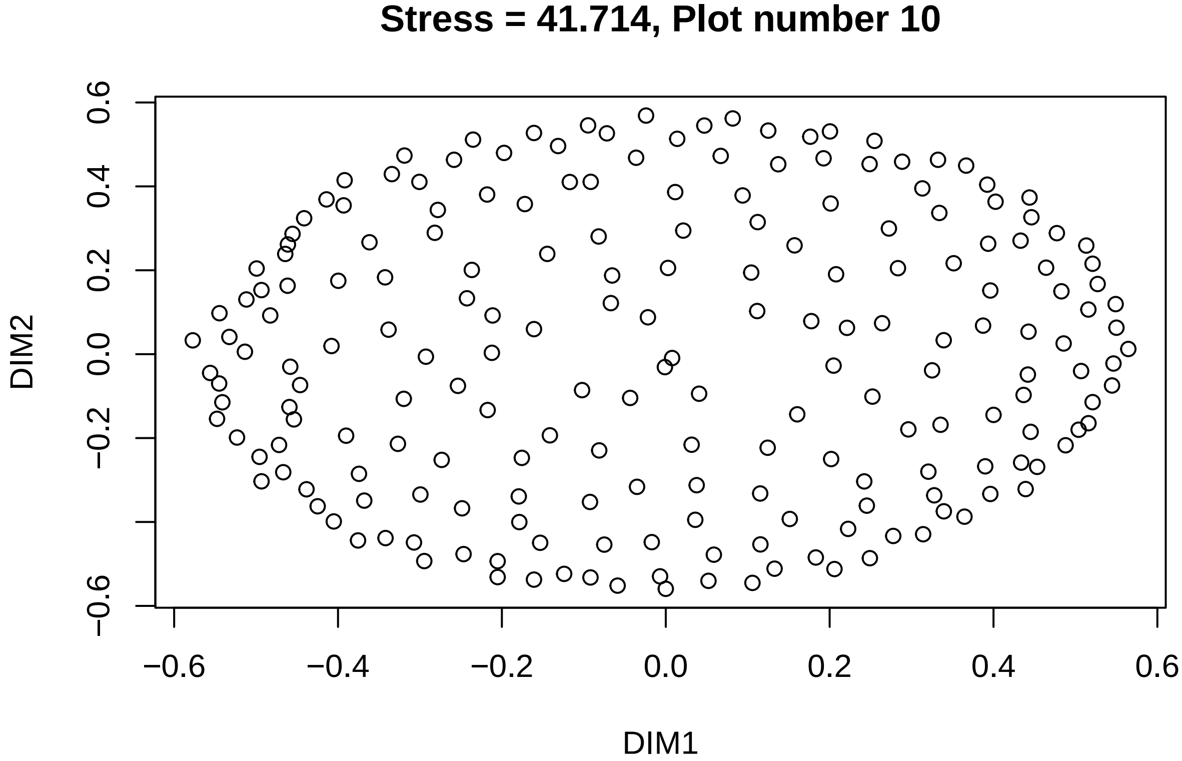


Stress = 11.74, Plot number 7



Stress = 12.076, Plot number 8

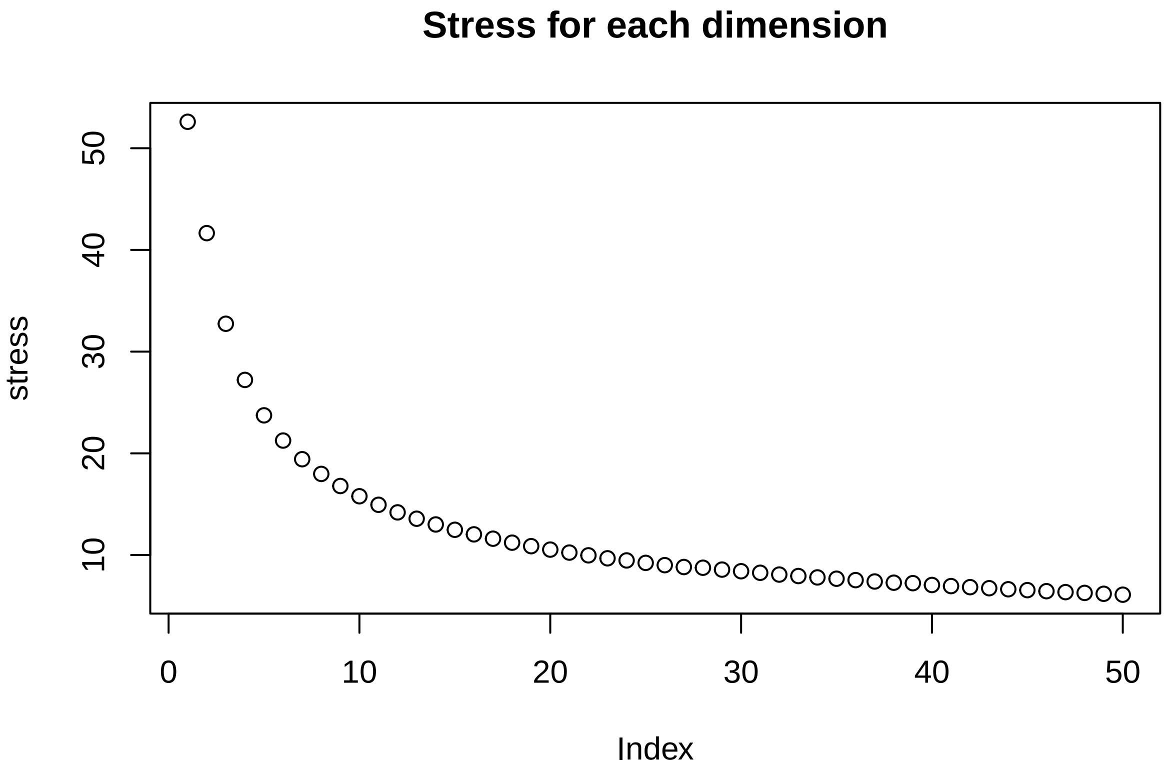




Here we can observe that when the stress is low, there can be seen separation of the population into subpopulations. In conclusion, a low stress value implies that the population is divided into subpopulations.

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.

```
stress <- c()
for (m in 1:50){
  init <- scale(matrix(runif(m*n),ncol=m),scale=FALSE)
  nonmms<-isoMDS(dist,y=init,k=m, trace = FALSE)
  stress <- c(stress, nonmms$stress)
}
plot(stress, main = "Stress for each dimension")
```



```
print(paste0('The number of dimensions necessary to obtain a good representation is',which(stress<10)[1
```

```
## [1] "The number of dimensions necessary to obtain a good representation is22dimensions"
```

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.

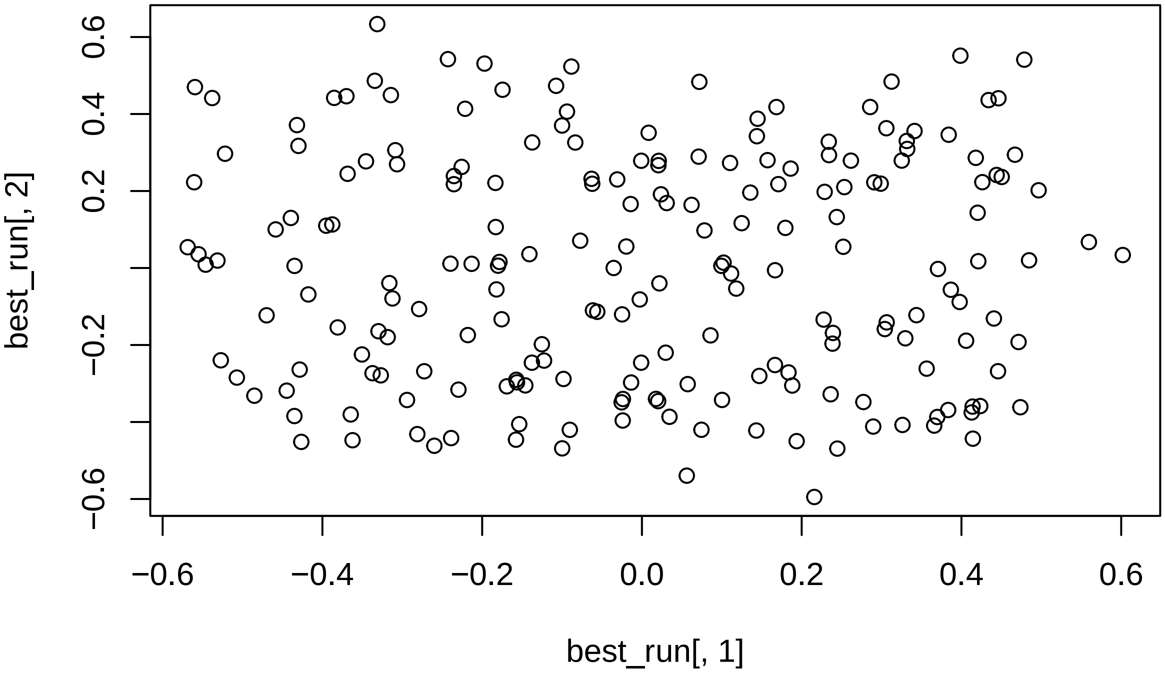
```
stress_100<-c()
points_100<-list()
for (x in 1:100){
  init <- scale(matrix(runif(m*n),ncol=m),scale=FALSE)
  nonmms<-isoMDS(dist,y=init,k=m, trace = FALSE)
  stress_100<-c(stress_100,nonmms$stress)
  points_100[[x]]<-nonmms$points
}
print(paste0('The worst run is run ',which(stress_100==min(stress_100))))
```

```
## [1] "The worst run is run 59"
```

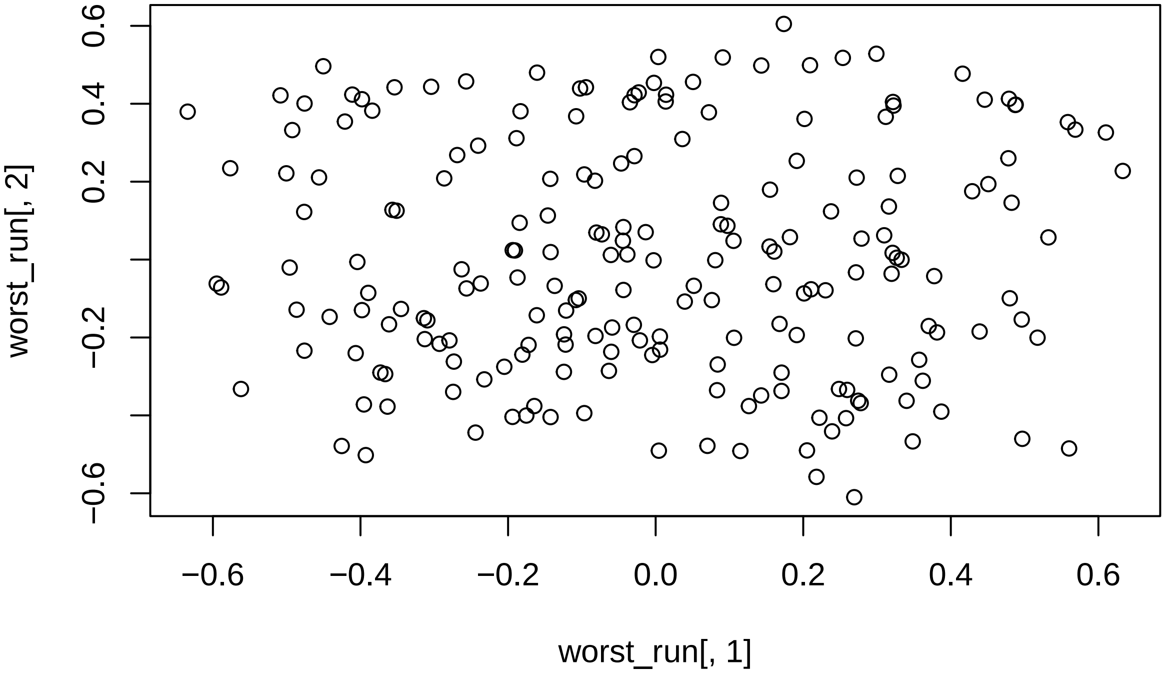
```
print(paste0('The best run is run ',which(stress_100==max(stress_100))))
```

```
## [1] "The best run is run 61"
```

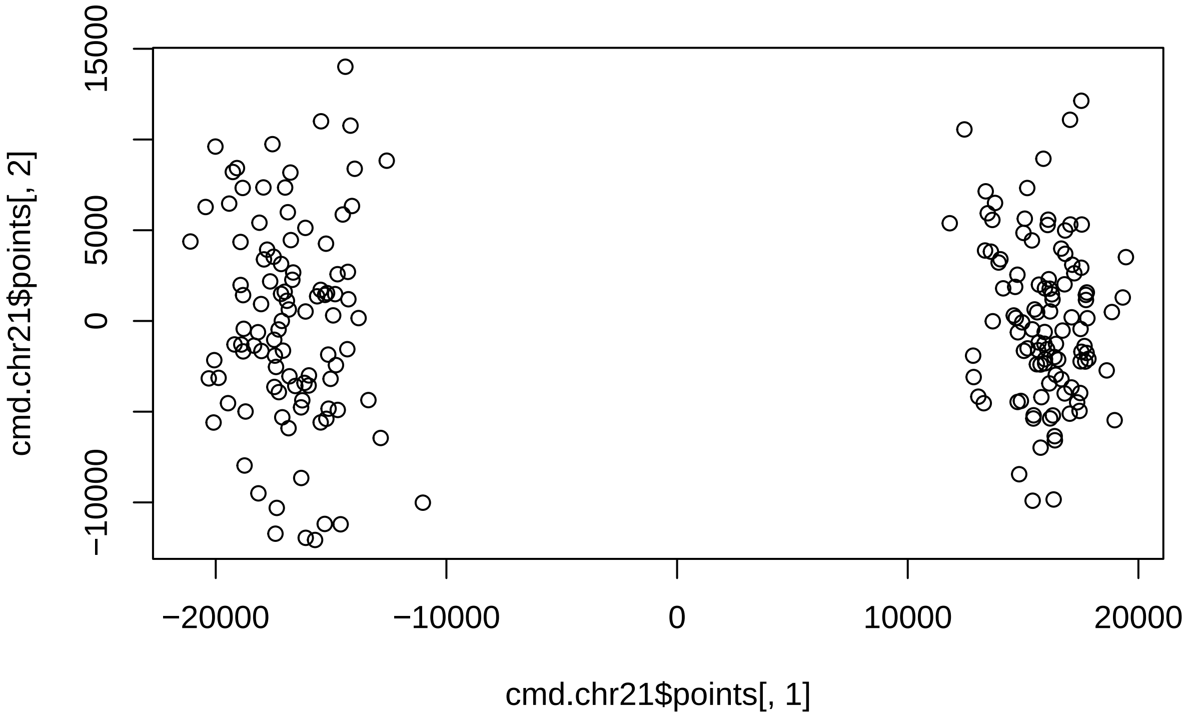
```
best_run<-points_100[[which(stress_100==min(stress_100))]]
worst_run<-points_100[[which(stress_100==max(stress_100))]]
plot(best_run[,1],best_run[,2])
```



```
plot(worst_run[,1],worst_run[,2])
```



```
plot(cmd.chr21$points[,1],cmd.chr21$points[,2])
```



```
print("We can clearly see that the results from the best and the worst run are different from the resul
```

```
## [1] "We can clearly see that the results from the best and the worst run are different from the resul
```

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.

```
corr_matrix<-cbind(cmd.chr21$points,best_run[,1:2])
correlationMat <- cor(corr_matrix)
correlationMat
```

```
##                [,1]          [,2]        [,3]         [,4]
## [1,]  1.000000e+00 -1.434445e-17 0.049206381 -0.065007499
## [2,] -1.434445e-17  1.000000e+00 0.001943080 -0.029889977
## [3,]  4.920638e-02  1.943080e-03 1.000000000  0.002425651
## [4,] -6.500750e-02 -2.988998e-02 0.002425651  1.000000000
```

```
print("From the results of the correlation matrix, we can state that there is no strong correlation (ne
```

```
## [1] "From the results of the correlation matrix, we can state that there is no strong correlation (ne
```