Assignment5 : Population substructure

Krishna Kalyan

1. The file SNPChr22.rda contains genotype information of 49 individuals of unknown background. The genotype information concerns 136 SNPs on chromosome 22. Load this data into the R environment. The data file contains a matrix Y containing the allele counts (0,1 or 2) for 136 SNPs for one of the alleles of each SNP.

```
load('SNPChr22.rda')
```

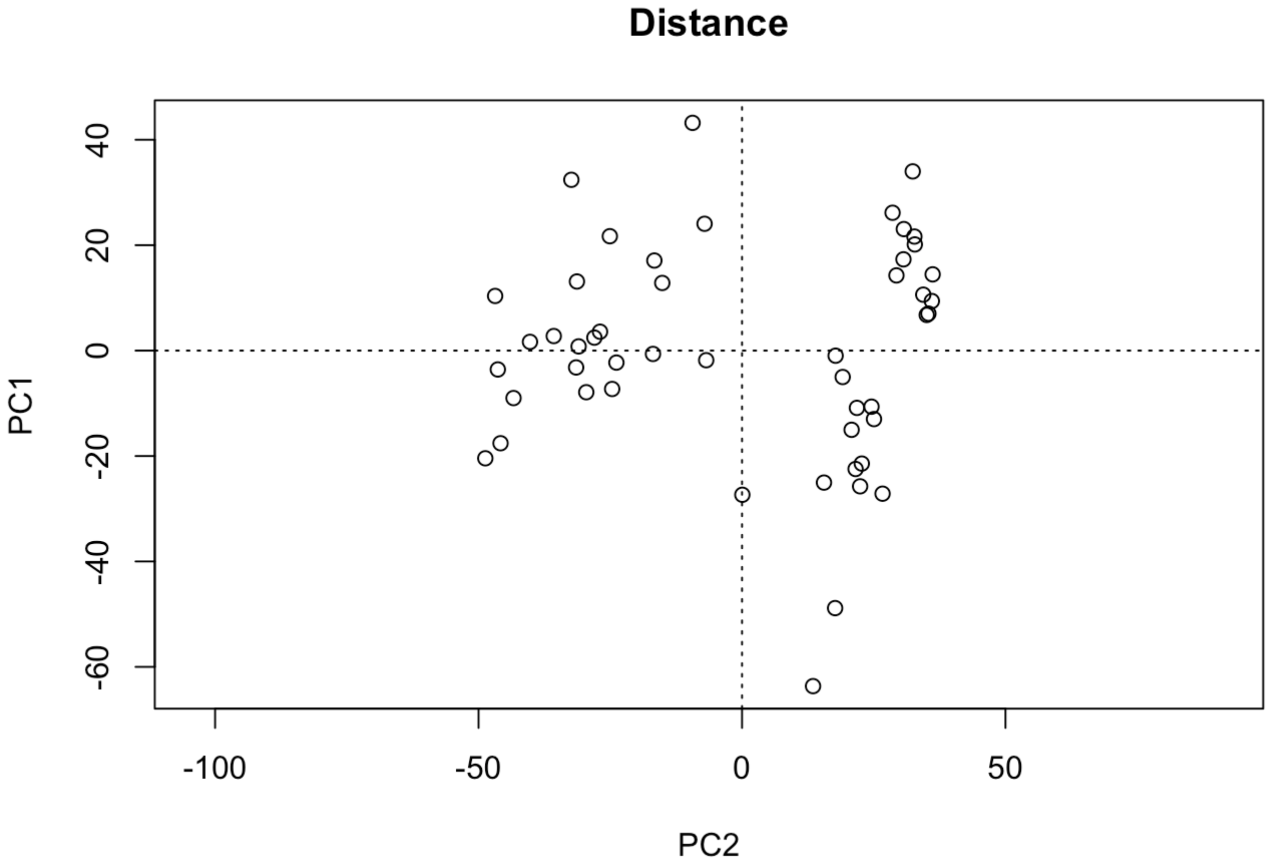
2. Compute the Manhattan distance matrix between the 49 individuals. Include a submatrix of dimension 5 by 5 with the distances between the first 5 individuals in your report.

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

```
##    1  2  3  4  5
## 1  0 80 49 55 65
## 2 80  0 71 77 81
## 3 49 71  0 52 54
## 4 55 77 52  0 62
## 5 65 81 54 62  0
```

3. Use metric multidimensional scaling to obtain a map of the individuals, and include your map in your report. Do you think the data come from one homogeneous population?

```
n = nrow(md)
md_out = cmdscale(md, k = n-1, eig=TRUE)
X = md_out$points[,1:2]
plot(X[,1], X[,2],type="p", ylab="PC1",
     xlab="PC2", main="Distance", asp = 1)
abline(v=0,lty=3)
abline(h=0,lty=3)
```



According to out plot we see two distinct set of popuation. In a homogeneous population all the individuals belong to the same group and are of the same type.

4. Report the eigenvalues of the solution.

```
md_out$eig
```

```
##  [1]  4.151272e+04  2.076796e+04  1.020205e+04  8.114257e+03  7.942764e+03
##  [6]  6.369659e+03  5.378938e+03  4.935698e+03  4.208222e+03  3.895748e+03
## [11]  3.353511e+03  3.030022e+03  2.469254e+03  1.830817e+03  1.695681e+03
## [16]  1.353295e+03  1.232518e+03  1.156127e+03  8.546821e+02  6.821345e+02
## [21]  5.811545e+02  4.845013e+02  2.249721e+02  1.698057e+02  4.795882e+01
## [26]  1.823470e+01 -2.273737e-12 -1.674019e+01 -5.842130e+01 -8.544354e+01
## [31] -2.535464e+02 -2.828951e+02 -3.452678e+02 -3.894639e+02 -4.051893e+02
## [36] -5.065839e+02 -5.378960e+02 -6.817495e+02 -7.244295e+02 -8.140500e+02
## [41] -9.313391e+02 -1.054480e+03 -1.108750e+03 -1.237461e+03 -1.357667e+03
## [46] -1.461376e+03 -1.695883e+03 -1.926193e+03 -2.175898e+03
```

5. Is the distance matrix you have used an Euclidean distance matrix?.

No, the distance matrix used was Manhattan distance.

$$|x_1 - x_2| + |y_1 - y_2|$$

This is the L_1 norm, where as the Euclidean distance is the L_2 norm.

6. What is the goodness-of-fit of a two-dimensional approximation to your distance matrix?

```
md_out$GOF[1]
```

```
## [1] 0.8801121
```

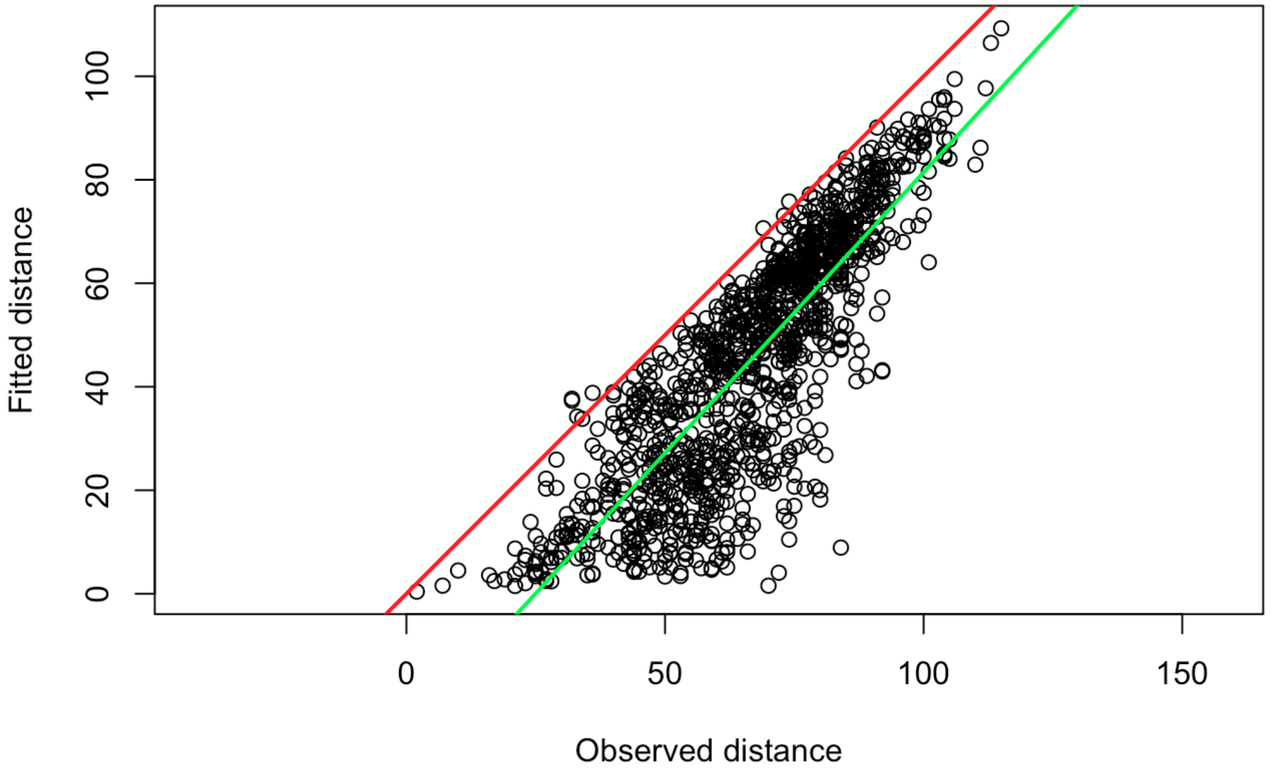
`GOF` is the amount of variance explained by our model. This value is usually represented by out eigenvaules sorted in descending order.

7. Make a plot of the estimated distances (according to your map of individuals) versus the observed distances. Regress estimated distances on observed distances and report the coefficient of determination of the regression.

```
D.fitted = as.matrix(dist(X))
D.obs = md[lower.tri(md)]
D.fit <- D.fitted[lower.tri(D.fitted)]
plot(D.obs,D.fit,asp=1,xlab="Observed distance",ylab="Fitted distance")
abline(0,1,col="red",lw=2)
out.lm <- lm(D.fit~D.obs)
summary(out.lm)$coef
```

```
##               Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) -27.071800 1.40731276 -19.23652  6.730527e-72
## D.obs         1.085097 0.02037819  53.24795 2.153915e-315
```

```
abline(summary(out.lm)$coef[1,1],summary(out.lm)$coef[2,1],col="green",lw=2)
```

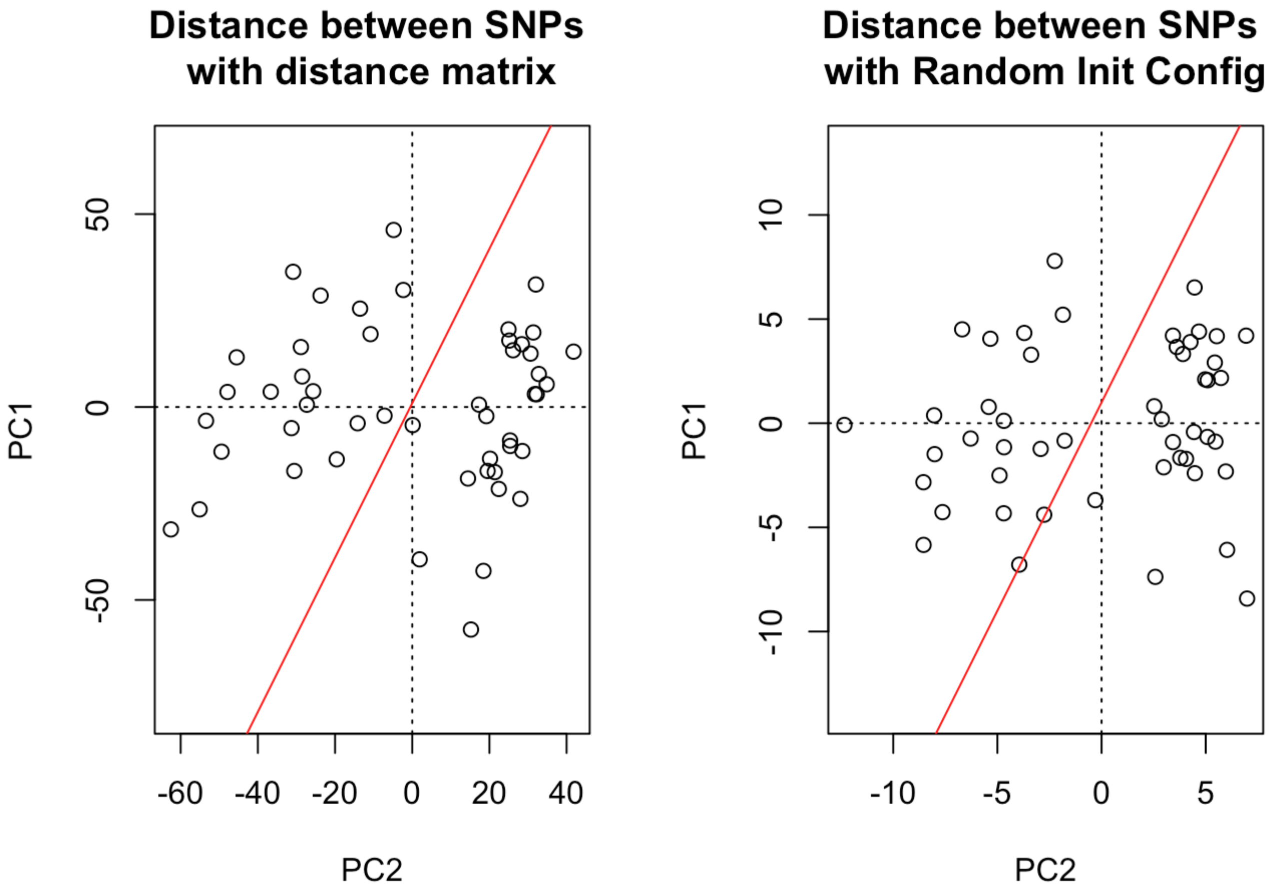


```
(R_sq= summary(out.lm)$adj.r.squared)
```

```
## [1] 0.7069346
```

8. Try now non-metric multidimensional scaling with your distance matrix. Use both a random initial configuration as well as the classical metric solution as an initial solution. Make a plot of the two-dimensional solution. Do the results support that the data come from one homogeneous population?

```
par(mfrow=c(1,2))
out.nmds.1 = isoMDS(md,k=2, trace = F)
X.nmds.1 = out.nmds.1$points
plot(X.nmds.1[,1],X.nmds.1[,2],asp=1, ylab="PC1",
     xlab="PC2",
     main="Distance between SNPs \nwith distance matrix")
abline(v=0,lty=3)
abline(h=0,lty=3)
abline(1,2, col="red")
x.init <- scale(matrix(runif(2*nrow(Y)),ncol=2))
out.nmds.2 <- isoMDS(md,y=x.init)
X.nmds.2 <- out.nmds.2$points
plot(X.nmds.2[,1],X.nmds.2[,2],asp=1, ylab="PC1",
     xlab="PC2",
     main="Distance between SNPs \nwith Random Init Config")
abline(v=0,lty=3)
abline(h=0,lty=3)
abline(1,2, col="red")
```



Distance between SNPs
with distance matrix

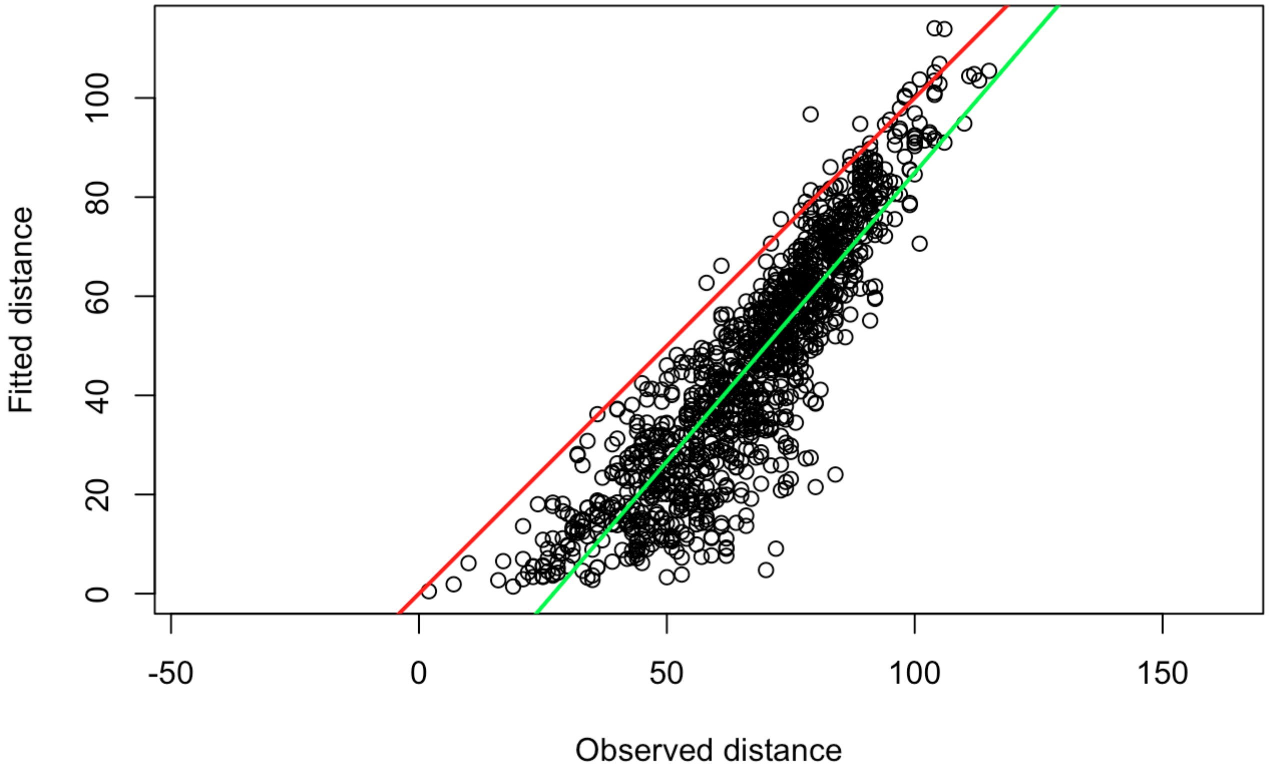
Distance between SNPs
with Random Init Config

Both the plots have 2 divided populations and divided groups.

9. Make again a plot of the estimated distances (according to your map of individuals) 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. Is the fit better or worse than with metric MDS?

```
D.Fitted <- as.matrix(dist(X.nmds.1))
D.fit2 <- D.Fitted[lower.tri(D.Fitted)]

plot(D.obs,D.fit2,asp=1,xlab="Observed distance",ylab="Fitted distance")
abline(0,1,col="red",lw=2)
out.lm <- lm(D.fit2~D.obs)
abline(summary(out.lm)$coef[1,1],summary(out.lm)$coef[2,1],col="green",lw=2)
```



```
summary(out.lm)$coef
```

```
##                Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) -31.601203 1.20788657 -26.16239 3.048145e-119
## D.obs         1.165212 0.01749046  66.61984  0.000000e+00
```

```
(R_sq1= summary(out.lm)$adj.r.squared)
```

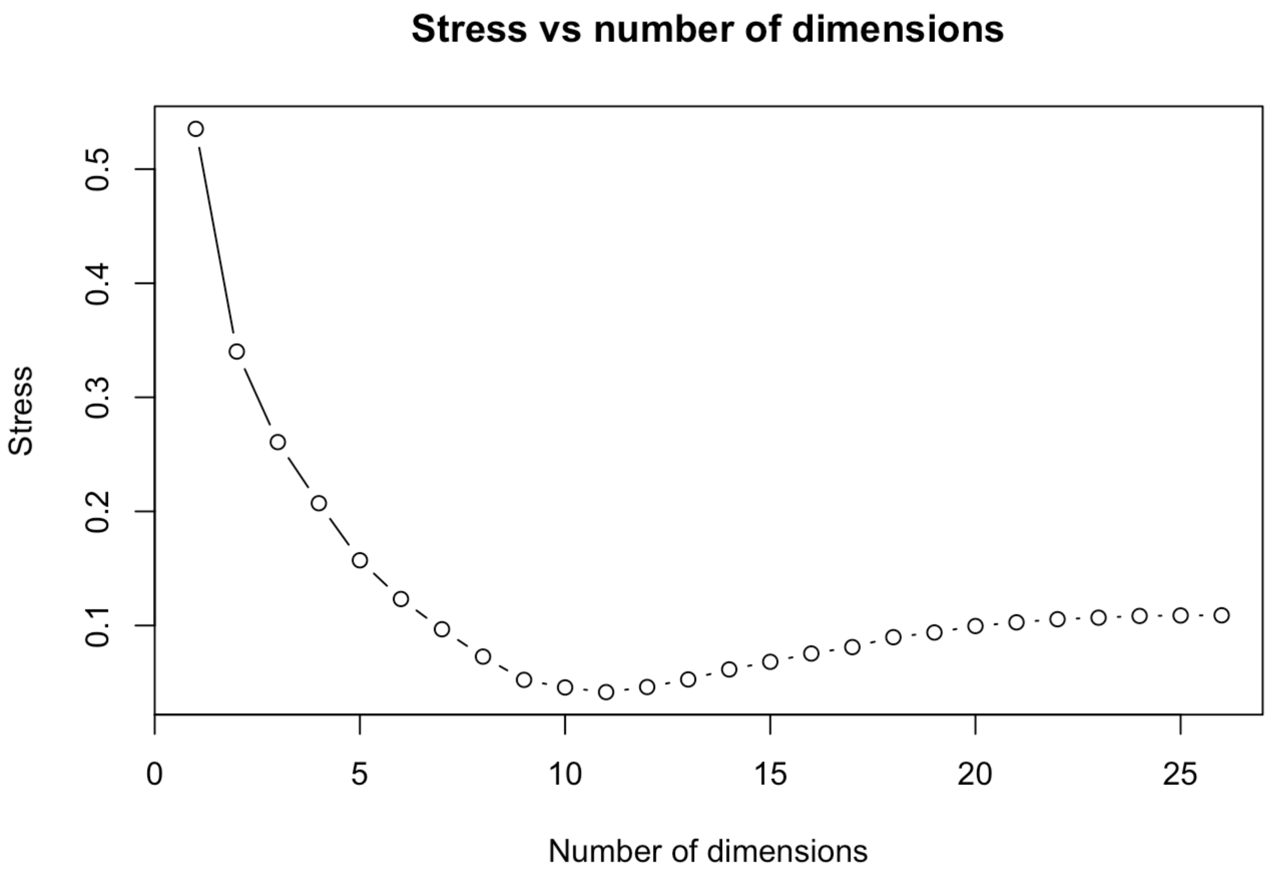
```
## [1] 0.7906348
```

The fit is slightly better as the adjusted R^2 value is `0.79` which is better than `0.70` .

10. 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? Make a plot of the stress against the number of dimensions.

```
get_stress <- function(real.dist, est.dist){
  diag(est.dist) <- 0
  diag(real.dist) <- 0
  stress <- sapply(1:length(real.dist),
                   function (x) (real.dist[x] - est.dist[x])^2)
  stress <- sqrt(sum(stress)/sum((real.dist)^2))
  return(stress)
}
get_nm_mds <- function(k, dist_mat){
  out_nmds <- isoMDS(dist_mat,k=k, trace = F)
 return(as.matrix(dist(out_nmds$points)))
}

stress <- sapply(1:26, function(x)
  get_stress(md, get_nm_mds(x, md)))
plot(1:26, stress, type = 'b', main = 'Stress vs number of dimensions',
     ylab = 'Stress', xlab = 'Number of dimensions')
```



```
which.min(stress)
```

```
## [1] 11
```

Minimum stress is achieved at `11th` dimension. We can say than based on stress level of 0.1 we can choose dimensions between `6 to 11`.

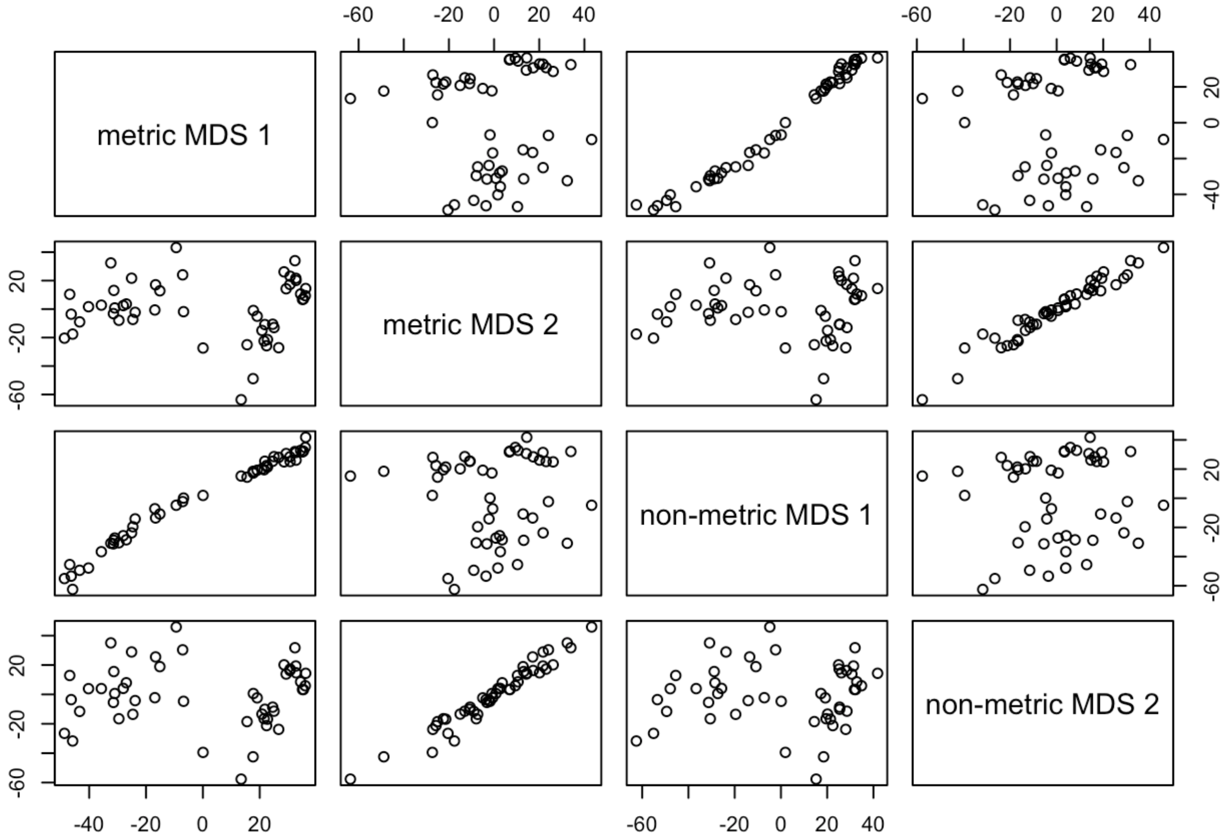
11. Compute the correlation matrix between the first two dimensions of a metric MDS and the two dimensional solution of a non-metric MDS. Make a scatterplot matrix of the 4 variables. Comment on your findings.

```
data_cor <- cor(cbind(X,X.nmds.1))
colnames(data_cor) <- c('metric MDS 1', 'metric MDS 2',
                              'non metric MDS 1', 'non metric MDS 2')
rownames(data_cor) <- c('metric MDS 1', 'metric MDS 2',
                              'non metric MDS 1', 'non metric MDS 2')

kable(data_cor, digits=3)
```

	metric MDS 1	metric MDS 2	non metric MDS 1	non metric MDS 2
metric MDS 1	1.000	0.000	0.988	0.011
metric MDS 2	0.000	1.000	0.011	0.972
non metric MDS 1	0.988	0.011	1.000	0.036
non metric MDS 2	0.011	0.972	0.036	1.000

```
pairs(cbind(X,X.nmds.1),
      labels = c('metric MDS 1', 'metric MDS 2',
          'non-metric MDS 1', 'non-metric MDS 2'))
```



We observe high correlation between the same dimension of each metric, and no correlation / little correlation between the same different dimensions as they are orthogonal to each other and uncorrelated.