R Notebook

```
library(car)
```

```
## Loading required package: carData
```

```
library(MASS)
library(plotrix)
```

Exercise 2

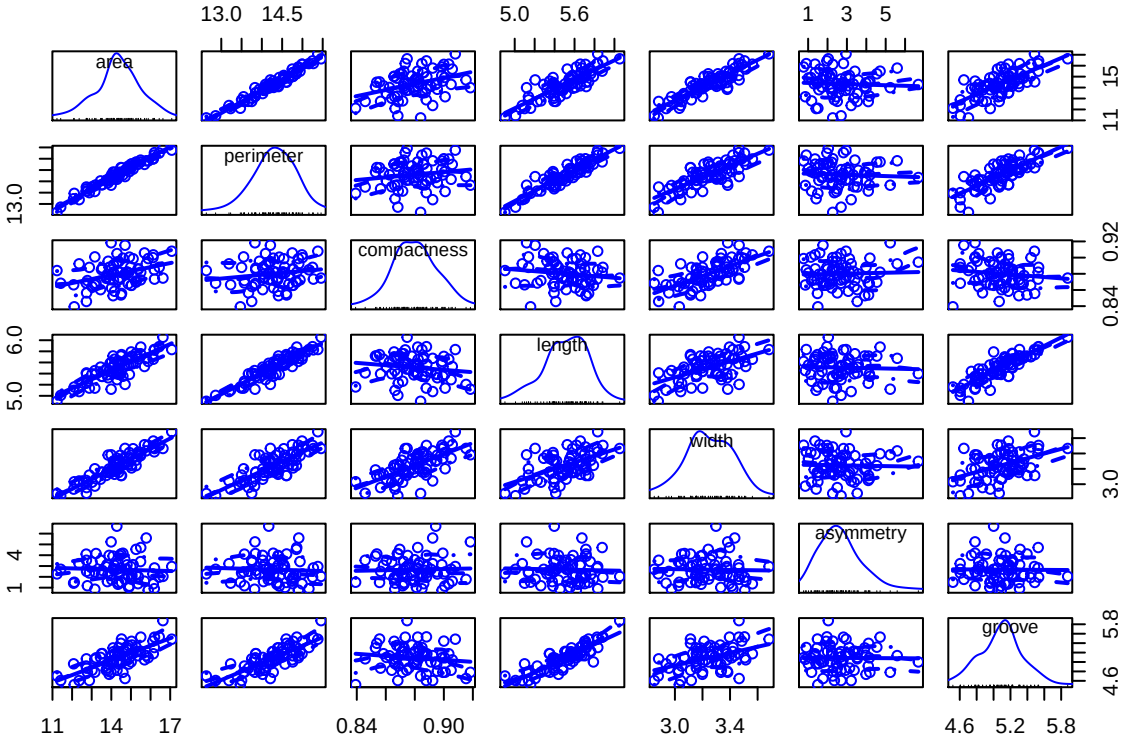
a)

```
ker <- read.table("http://www-eio.upc.es/~jan/Data/MVA/kernels.dat", header=T)
head(ker)
```

```
##    area perimeter compactness length width asymmetry groove
## 1 15.26     14.84      0.8710  5.763 3.312     2.221  5.220
## 2 14.88     14.57      0.8811  5.554 3.333     1.018  4.956
## 3 14.29     14.09      0.9050  5.291 3.337     2.699  4.825
## 4 13.84     13.94      0.8955  5.324 3.379     2.259  4.805
## 5 16.14     14.99      0.9034  5.658 3.562     1.355  5.175
## 6 14.38     14.21      0.8951  5.386 3.312     2.462  4.956
```

b) Perimeter and area are the most correlated.

```
scatterplotMatrix(ker)
```



```
cor(ker)
```

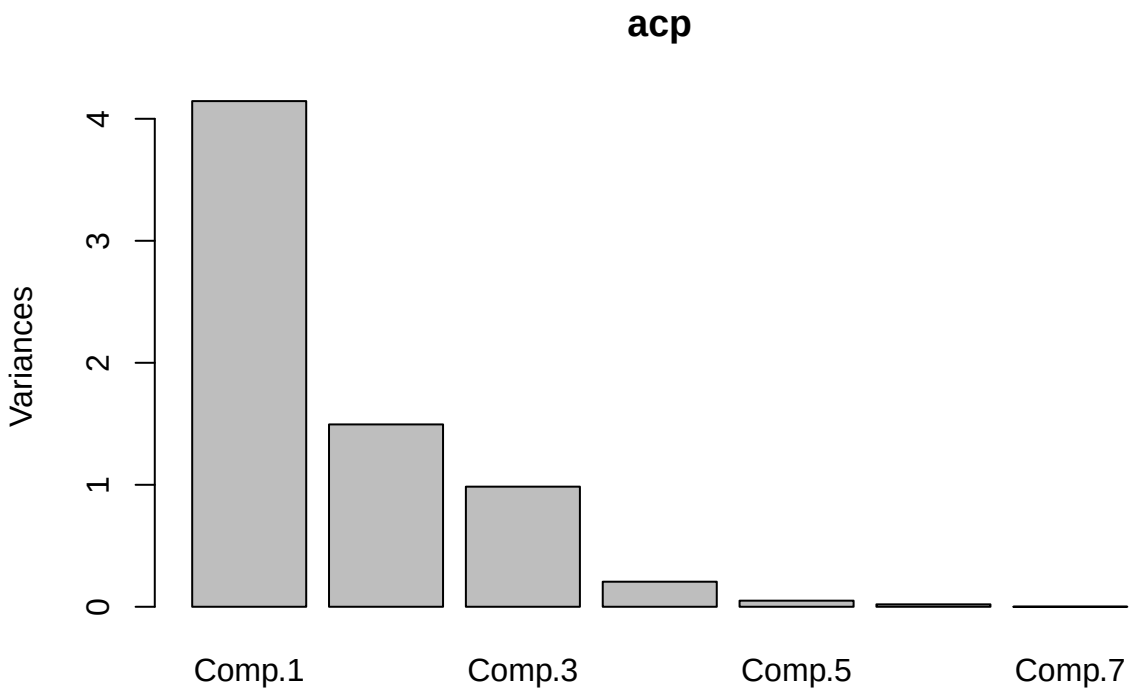
```
##                   area   perimeter compactness      length       width
## area         1.00000000  0.97643665  0.37103733  0.83477809  0.90006617
## perimeter    0.97643665  1.00000000  0.16492283  0.92120227  0.80235953
## compactness  0.37103733  0.16492283  1.00000000 -0.14630391  0.66657308
## length       0.83477809  0.92120227 -0.14630391  1.00000000  0.55056053
## width        0.90006617  0.80235953  0.66657308  0.55056053  1.00000000
## asymmetry   -0.05048194 -0.05394128  0.03695775 -0.03668859 -0.02667164
## groove       0.72095279  0.79367796 -0.13107635  0.86615879  0.44711056
##                asymmetry      groove
## area        -0.05048194  0.72095279
## perimeter   -0.05394128  0.79367796
## compactness  0.03695775 -0.13107635
## length      -0.03668859  0.86615879
## width       -0.02667164  0.44711056
## asymmetry    1.00000000 -0.01101627
## groove      -0.01101627  1.00000000
```

c) Seems that 2 components are enough.

```
ker2 <- scale(ker)
acp <- princomp(ker2)
summary(acp)
```

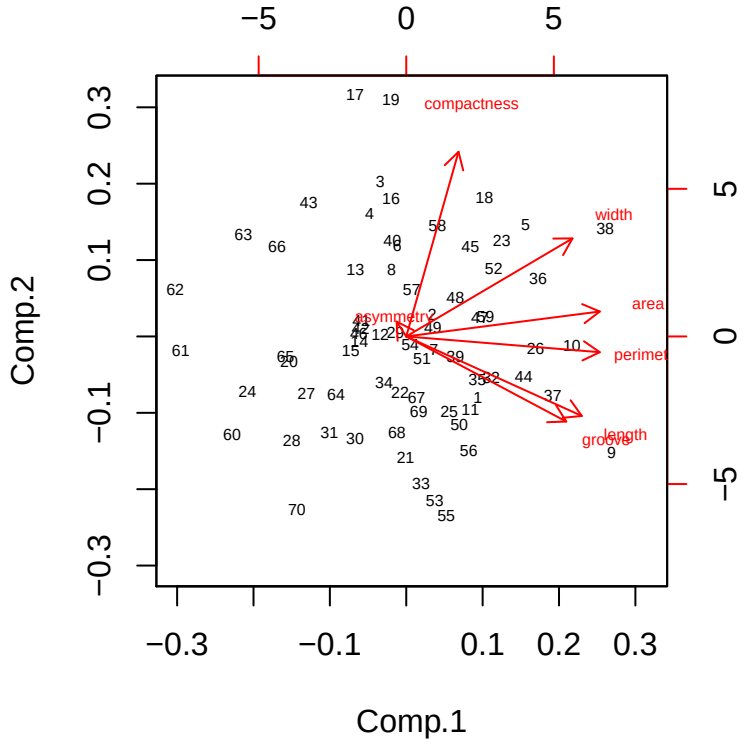
```
## Importance of components:
##                            Comp.1    Comp.2    Comp.3     Comp.4
## Standard deviation     2.0359084 1.2225556 0.9922068 0.45322847
## Proportion of Variance 0.6007135 0.2166148 0.1426774 0.02977044
## Cumulative Proportion  0.6007135 0.8173283 0.9600057 0.98977618
##                              Comp.5      Comp.6       Comp.7
## Standard deviation     0.223638552 0.141269676 2.393861e-02
## Proportion of Variance 0.007248435 0.002892336 8.305173e-05
## Cumulative Proportion  0.997024612 0.999916948 1.000000e+00
```

```
screeplot(acp)
```



d) The second component is principally afected by the compactness. While the first one is almost a mean, except for compactness. It can be seen as the second component almost being compactness and the first the rest.

```
biplot(acp, cex=0.5)
```



e) Length and groove, it isn't the same as the one given by the correlation matrix

f) It is greater in all components

```
apply(acp$scores, 2, sd)
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
## 2.05060830 1.23138284 0.99937082 0.45650092 0.22525329 0.14228969
##     Comp.7
## 0.02411145
```

```
acp$sdev
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
## 2.03590841 1.22255561 0.99220678 0.45322847 0.22363855 0.14126968
##     Comp.7
## 0.02393861
```

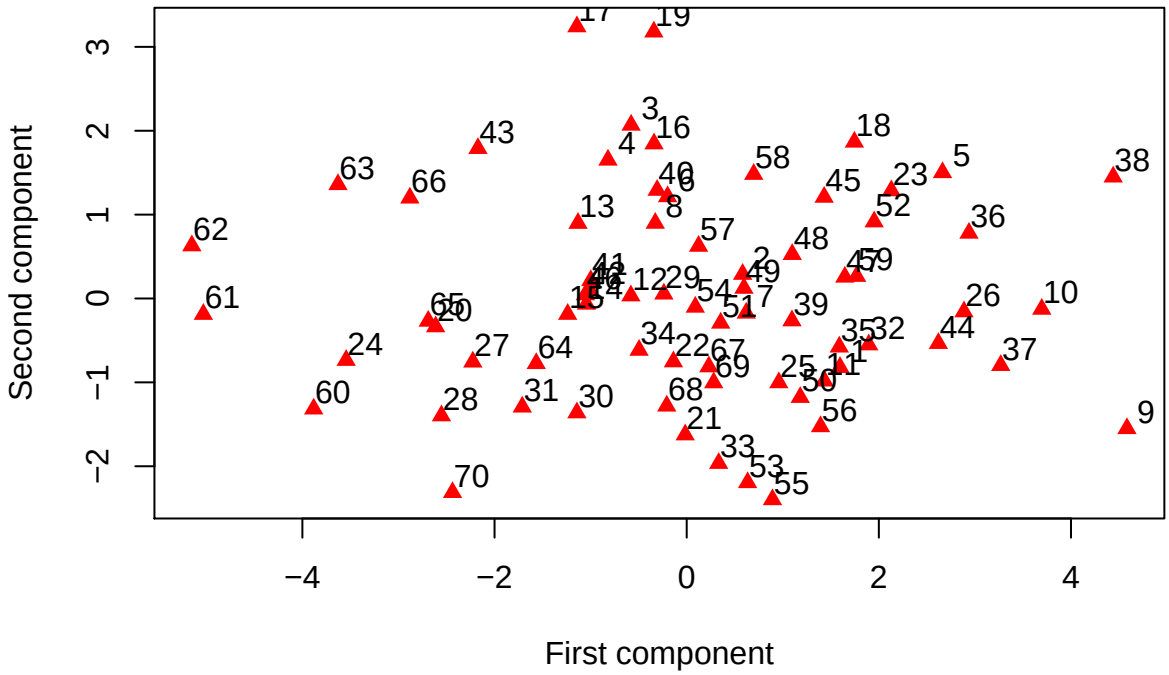
g) It gives the projection matrix.

```
acp$loadings
```

```
##
## Loadings:
##             Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## area         0.481  0.103         0.160         0.456  0.722
## perimeter    0.481                0.270         0.485 -0.675
## compactness  0.130  0.764        -0.389  0.475        -0.149
## length       0.436 -0.329         0.211  0.597 -0.546
## width        0.413  0.406         0.189 -0.608 -0.508
## asymmetry                -0.996
## groove       0.398 -0.352        -0.817 -0.211
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

h) It gives more or less the same result, except for the arrows which aren't plotted.

```
plot(acp$scores[,1], acp$scores[,2], xlab="First component", ylab="Second component", pch=17, col="red")
text(acp$scores[,1]+0.2, acp$scores[,2]+0.2, labels=rownames(ker))
```



i) The correlation between different principal components is almost 0, as expected, i.e. the variables are independent.

```
cor(as.matrix(acp$scores))
```

```
##               Comp.1        Comp.2        Comp.3        Comp.4
## Comp.1  1.000000e+00  1.114776e-16 -2.863031e-18 -6.159107e-16
## Comp.2  1.114776e-16  1.000000e+00  2.092075e-16  1.212260e-15
## Comp.3 -2.863031e-18  2.092075e-16  1.000000e+00 -8.553762e-16
## Comp.4 -6.159107e-16  1.212260e-15 -8.553762e-16  1.000000e+00
## Comp.5 -3.769384e-16  1.906339e-15  1.130203e-15 -2.466619e-15
## Comp.6 -3.362674e-15  5.368211e-16  4.469410e-15  1.947465e-15
## Comp.7 -1.134441e-14 -9.451872e-15 -2.075367e-14  1.411479e-14
##               Comp.5        Comp.6        Comp.7
## Comp.1 -3.769384e-16 -3.362674e-15 -1.134441e-14
## Comp.2  1.906339e-15  5.368211e-16 -9.451872e-15
## Comp.3  1.130203e-15  4.469410e-15 -2.075367e-14
## Comp.4 -2.466619e-15  1.947465e-15  1.411479e-14
## Comp.5  1.000000e+00  8.593496e-16 -4.955240e-14
## Comp.6  8.593496e-16  1.000000e+00  3.260248e-14
## Comp.7 -4.955240e-14  3.260248e-14  1.000000e+00
```

j)

```
X <- scale(ker)
Y = svd(X)
print(Y$d)
```

```
## [1] 17.0336319 10.2286341  8.3013975  3.7919814  1.8710944  1.1819469
## [7]  0.2002848
```

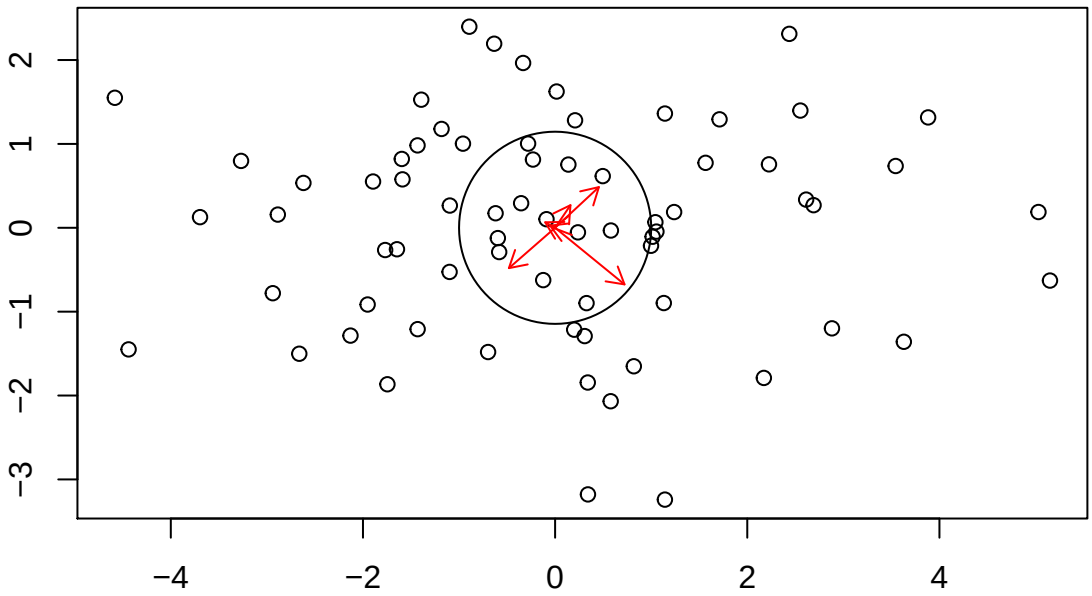
k)

```
pc <- Y$u%*%diag(Y$d)
head(pc)
```

```
##             [,1]       [,2]       [,3]        [,4]         [,5]
## [1,] -1.5957328  0.8206821 -0.3157637  0.46767917 -0.075191143
## [2,] -0.5822900 -0.2893664 -1.4486292  0.64601861  0.018724188
## [3,]  0.5784016 -2.0678495 -0.1152505  0.01427686 -0.071327688
## [4,]  0.8191770 -1.6509794 -0.4794181  0.23054341  0.257216804
## [5,] -2.6628361 -1.5017946 -1.1585181  0.14703854  0.024864032
## [6,]  0.1987344 -1.2149506 -0.2539833 -0.03637763 -0.008173573
##              [,6]        [,7]
## [1,] -0.004573267 -0.00363088
## [2,]  0.060115607 -0.01428786
## [3,]  0.083835223 -0.02380646
## [4,] -0.418473956 -0.03059510
## [5,] -0.009951482  0.02170931
## [6,]  0.051965662 -0.01922961
```

l)

```
plot(pc[,1:2], xlab="", ylab="")
arrows(x0=rep(0,7),y0=rep(0,7),x1=t(Y$v[1,1:7]),y1=t(Y$v[2,1:7]), length = 0.1, col="red")
draw.circle(0,0,radius=1)
```



m) The variable number 6 which corresponds with asymmetry. The reason is that this variable is represented almost totally by the third component.

```
which.min((abs(Y$v[,1])+abs(Y$v[,2]))/sum(abs(Y$v[,1:7])))
```

```
## [1] 6
```

n) It is the sum of the singular values.

```
sum(Y$d)
```

```
## [1] 42.60897
```

o)

```
sum(Y$d[1:3])/sum(Y$d) * 100
```

```
## [1] 83.4652
```

p) Basically represents the asymmetry variable. It is 99.6% asymmetry and very little the others variables.

```
Y$v
```

```
##             [,1]        [,2]         [,3]        [,4]        [,5]
## [1,] -0.48121359 -0.10314023 -0.013445996  0.15966666 -0.05175088
## [2,] -0.48110975  0.06531974 -0.002417561  0.26956705  0.02569809
## [3,] -0.12995166 -0.76431276 -0.016669011 -0.38864994 -0.47471131
## [4,] -0.43645397  0.32907875  0.036453796  0.21133486 -0.59731669
## [5,] -0.41327599 -0.40583935 -0.025186145  0.18881331  0.60811717
## [6,]  0.02398102 -0.06138766  0.996267869  0.05147967  0.01363949
## [7,] -0.39776281  0.35235576  0.070870428 -0.81720765  0.21102338
##              [,6]         [,7]
## [1,]  0.455527757  0.722452520
## [2,]  0.485252943 -0.674880503
## [3,]  0.007776154 -0.148998813
## [4,] -0.546165503  0.011824418
## [5,] -0.508208464 -0.011411920
## [6,]  0.015468398  0.005597953
## [7,] -0.012316848 -0.009830076
```

4

a)

```
dist <- as.matrix(read.table("http://www-eio.upc.es/~jan/data/MVA/GeneticDist.txt"))
dist[1:6,1:6]
```

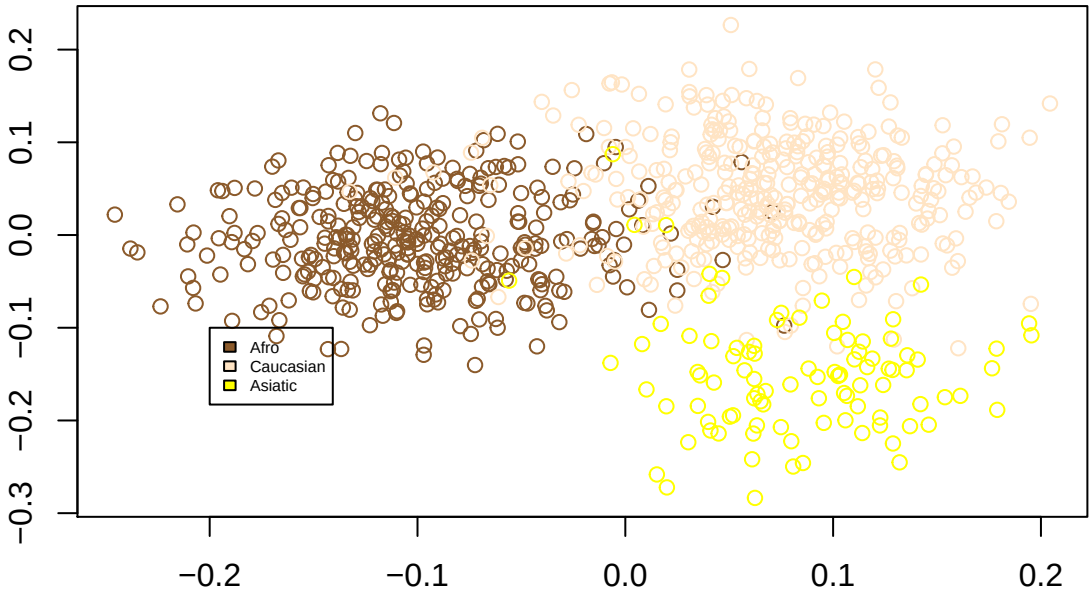
```
##             V1        V2        V3        V4        V5        V6
## [1,] 0.0000000 0.8163265 0.7789474 0.8041237 0.7111111 0.8282828
## [2,] 0.8163265 0.0000000 0.8163265 0.7916667 0.8163265 0.7789474
## [3,] 0.7789474 0.8163265 0.0000000 0.7916667 0.8627451 0.8846154
## [4,] 0.8041237 0.7916667 0.7916667 0.0000000 0.7916667 0.7526882
## [5,] 0.7111111 0.8163265 0.8627451 0.7916667 0.0000000 0.7789474
## [6,] 0.8282828 0.7789474 0.8846154 0.7526882 0.7789474 0.0000000
```

```
raza <- as.factor(c(rep("Afro", 342), rep("Caucasico", 361), rep("Asiatico", 97)))
head(raza)
```

```
## [1] Afro Afro Afro Afro Afro Afro
## Levels: Afro Asiatico Caucasico
```

b)

```
pco <- cmdscale(dist, k = 3, eig=T)
plot(pco$points[,1], pco$points[,2], xlab="", ylab="", col=c(rep("tan4", 342), rep("bisque", 361), rep(
legend(x=-0.2, y=-0.1, legend=c("Afro", "Caucasian", "Asiatic"), fill=c("tan4","bisque","yellow"), cex=
```



c) Yes, if k is equal to the number of genetics variables. Or k = 799 too. Because we need as many dimensions as the data has to represent it. However if the points are classified in a very high dimension but we only have a few points, they can only represent as many independent directions as the number of points minus one.

d) Yes, because the distance matrix is centered to compute the MDS. In this case there is only one as there are no other linear relations among the points.

```
sum(abs(pco$eig)<1E-13)
```

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

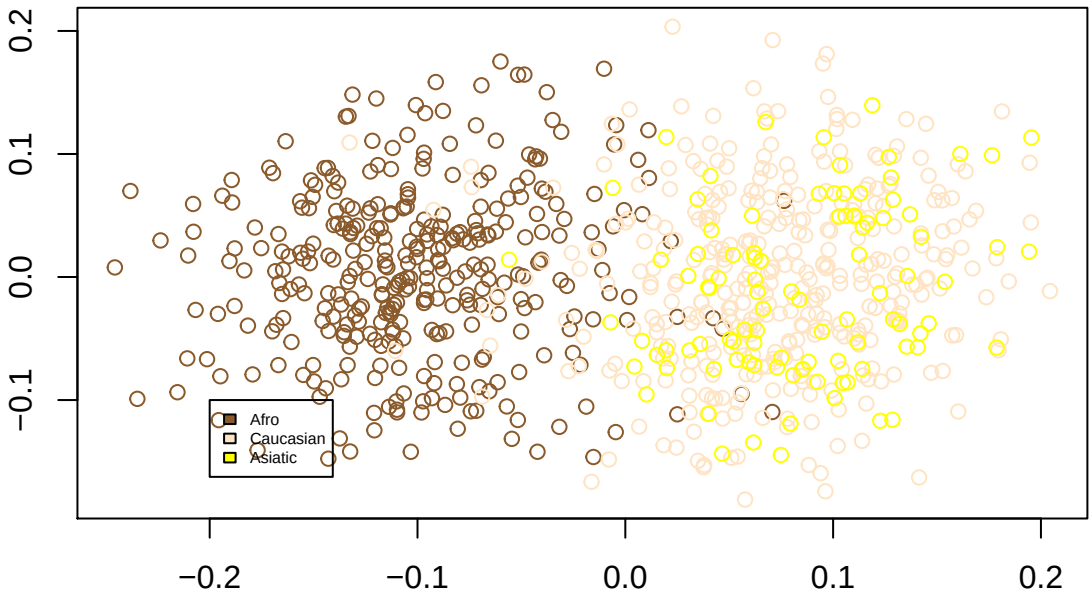
e) It explains only 4% of the variability of the data but since it is genetic data it is quite good.

```
sum(abs(pco$eig[1:2]))/sum(abs(pco$eig))
```

```
## [1] 0.04287947
```

f) It doesn't help, in fact it mixes two races and has a smaller goodness of fit.

```
plot(pco$points[,1], pco$points[,3], xlab="", ylab="", col=c(rep("tan4", 342), rep("bisque", 361), rep(
legend(x=-0.2, y=-0.1, legend=c("Afro", "Caucasian", "Asiatic"), fill=c("tan4","bisque","yellow"), cex=
```

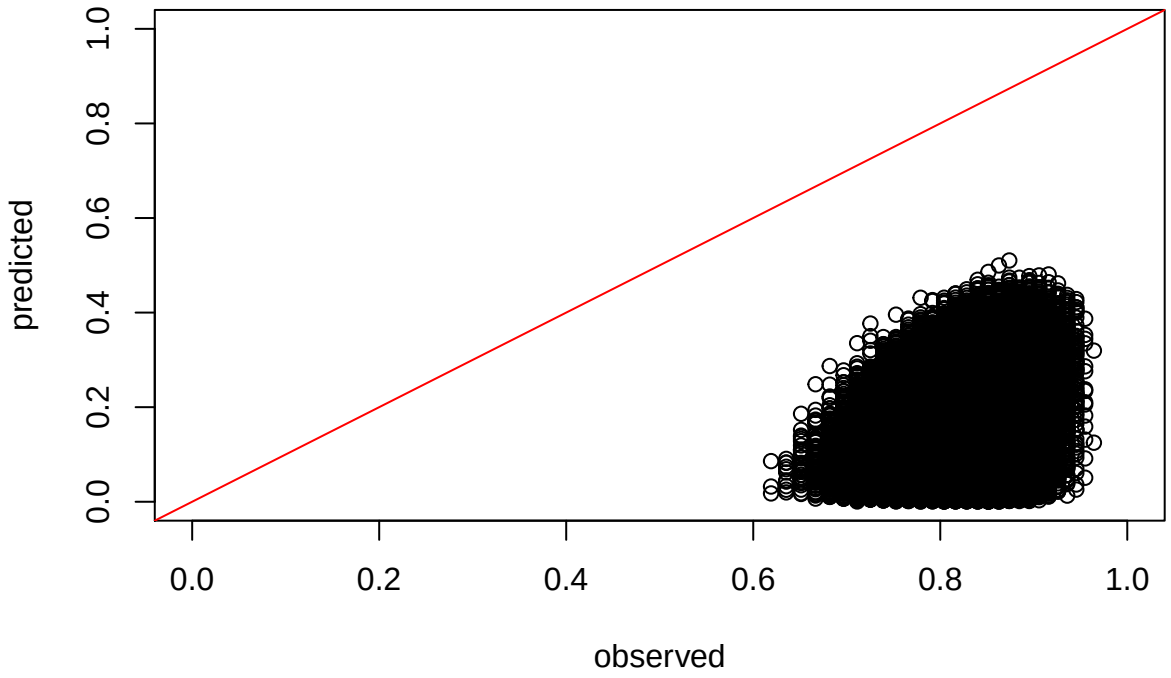



```
sum(abs(pco$eig[c(1,3)]))/sum(abs(pco$eig))
```

```
## [1] 0.0382814
```

g) It has a lot of variance and it doesn't fit the line y=x. They have a much smaller predicted distance than the observed one in general.

```
d2 <- as.matrix(dist(pco$points[,1:2]))
plot(x=dist[lower.tri(dist)], y=d2[lower.tri(d2)], xlab="observed", ylab="predicted", type="p", xlim=c(0
abline(a=0,b=1, col="red")
```



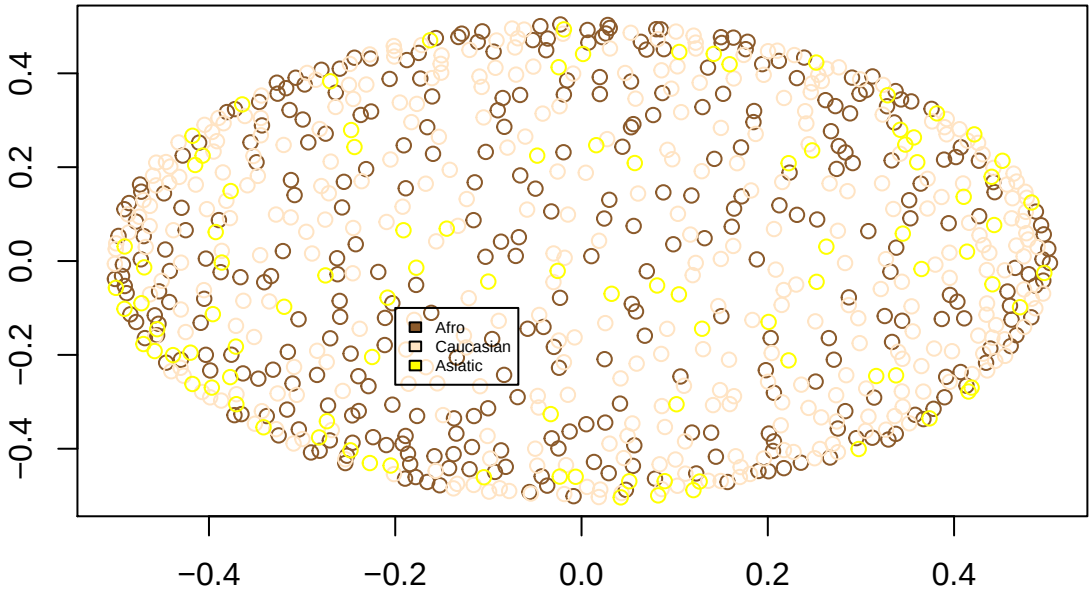
h) The stress is 42.01934 and the three groups are totally mixed.

```
set.seed(1234)
n <- 800
init <- scale(matrix(runif(n*2), ncol=2), scale=F)

nMDS <- isoMDS(dist, k = 2, y = init)
```

```
## initial  value 42.907281
## final  value 42.019343
## converged
```

```
plot(nMDS$points[,1], nMDS$points[,2], xlab="", ylab="", col=c(rep("tan4", 342), rep("bisque", 361), rep
legend(x=-0.2, y=-0.1, legend=c("Afro", "Caucasian", "Asiatic"), fill=c("tan4","bisque","yellow"), cex=0
```




```
nMDS$stress
```

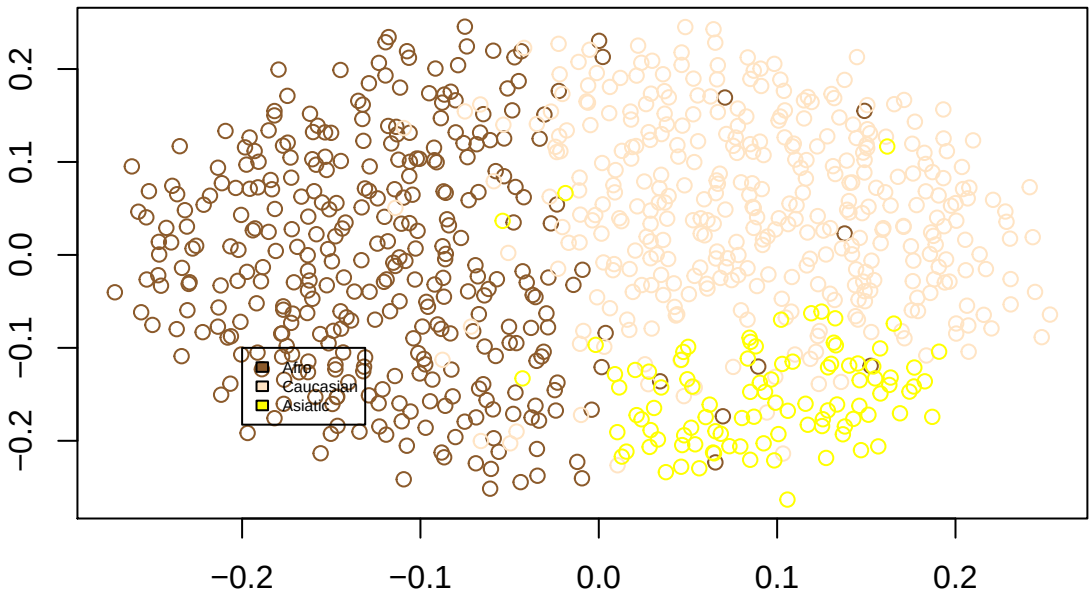
```
## [1] 42.01934
```

i) The stress has decreased and the groups are now distinguishable

```
nMDS <- isoMDS(dist, k = 2)
```

```
## initial  value 43.801421
## iter   5 value 37.967335
## final  value 37.577491
## converged
```

```
plot(nMDS$points[,1], nMDS$points[,2], xlab="", ylab="", col=c(rep("tan4", 342), rep("bisque", 361), rep
legend(x=-0.2, y=-0.1, legend=c("Afro", "Caucasian", "Asiatic"), fill=c("tan4","bisque","yellow"), cex=0
```




```
min_stress <- nMDS$stress
```

j) There wasn't any random configuration that improved the stress of i).

```
for (i in 1:100) {
  print(i)

  init <- scale(matrix(runif(n*2), ncol=2), scale=F)
  nMDS <- isoMDS(dist, k = 2, y = init)
  if (nMDS$stress < min_stress) {
    plot(nMDS$points[,1], nMDS$points[,2], xlab="", ylab="", col=c(rep("tan4", 342), rep("bisque", 361)
    legend(x=-0.2, y=-0.1, legend=c("Afro", "Caucasian", "Asiatic"), fill=c("tan4","bisque","yellow"),
  }
}
```

```
## [1] 1
## initial  value 42.943677
## final  value 42.020332
## converged
## [1] 2
## initial  value 42.783945
## final  value 42.013450
## converged
## [1] 3
## initial  value 42.972085
## final  value 42.020975
## converged
## [1] 4
## initial  value 42.893198
## final  value 42.014596
## converged
## [1] 5
## initial  value 43.049628
## final  value 42.020633
## converged
```

```
## [1] 6
## initial  value 43.096643
## final  value 42.019116
## converged
## [1] 7
## initial  value 42.974863
## final  value 42.020265
## converged
## [1] 8
## initial  value 42.888747
## final  value 42.020159
## converged
## [1] 9
## initial  value 43.100710
## final  value 42.019803
## converged
## [1] 10
## initial  value 42.872473
## final  value 42.018691
## converged
## [1] 11
## initial  value 43.065013
## final  value 42.019269
## converged
## [1] 12
## initial  value 42.926726
## final  value 42.019970
## converged
## [1] 13
## initial  value 42.872070
## final  value 42.020571
## converged
## [1] 14
## initial  value 43.179066
## final  value 42.021541
## converged
## [1] 15
## initial  value 43.030520
## final  value 42.017258
## converged
## [1] 16
## initial  value 42.718971
## final  value 42.019791
## converged
## [1] 17
## initial  value 42.816927
## final  value 42.019161
## converged
## [1] 18
## initial  value 42.930315
## final  value 42.012259
## converged
## [1] 19
## initial  value 42.950602
```

```
## final  value 42.020038
## converged
## [1] 20
## initial  value 42.796021
## final  value 42.017441
## converged
## [1] 21
## initial  value 42.995689
## final  value 42.020847
## converged
## [1] 22
## initial  value 42.700347
## final  value 42.019428
## converged
## [1] 23
## initial  value 42.771959
## final  value 42.018805
## converged
## [1] 24
## initial  value 42.695460
## final  value 42.019645
## converged
## [1] 25
## initial  value 42.854090
## final  value 42.020819
## converged
## [1] 26
## initial  value 42.902630
## final  value 42.018394
## converged
## [1] 27
## initial  value 42.927262
## final  value 42.017357
## converged
## [1] 28
## initial  value 43.082181
## final  value 42.019861
## converged
## [1] 29
## initial  value 42.748571
## final  value 42.019711
## converged
## [1] 30
## initial  value 42.848987
## final  value 42.011103
## converged
## [1] 31
## initial  value 42.761389
## final  value 42.018686
## converged
## [1] 32
## initial  value 42.845818
## final  value 42.014746
## converged
```

```
## [1] 33
## initial  value 42.907216
## final  value 42.017485
## converged
## [1] 34
## initial  value 42.973692
## final  value 42.021426
## converged
## [1] 35
## initial  value 42.867657
## final  value 42.020635
## converged
## [1] 36
## initial  value 42.974371
## final  value 42.015870
## converged
## [1] 37
## initial  value 42.975754
## final  value 42.010351
## converged
## [1] 38
## initial  value 42.779477
## final  value 42.014398
## converged
## [1] 39
## initial  value 42.943967
## final  value 42.019607
## converged
## [1] 40
## initial  value 42.877387
## final  value 42.018894
## converged
## [1] 41
## initial  value 42.811410
## final  value 42.020009
## converged
## [1] 42
## initial  value 42.755024
## final  value 42.019047
## converged
## [1] 43
## initial  value 42.839092
## final  value 42.020557
## converged
## [1] 44
## initial  value 43.097685
## final  value 42.019262
## converged
## [1] 45
## initial  value 42.829735
## final  value 42.013533
## converged
## [1] 46
## initial  value 42.905193
```

```
## final  value 42.015745
## converged
## [1] 47
## initial  value 42.820679
## final  value 42.020580
## converged
## [1] 48
## initial  value 42.962372
## final  value 42.001992
## converged
## [1] 49
## initial  value 42.993833
## final  value 42.019405
## converged
## [1] 50
## initial  value 42.953991
## final  value 42.020406
## converged
## [1] 51
## initial  value 42.767867
## final  value 42.003916
## converged
## [1] 52
## initial  value 42.891856
## final  value 42.019679
## converged
## [1] 53
## initial  value 42.800182
## final  value 42.020288
## converged
## [1] 54
## initial  value 43.040100
## final  value 42.014851
## converged
## [1] 55
## initial  value 42.896944
## final  value 42.017941
## converged
## [1] 56
## initial  value 43.129743
## final  value 42.020385
## converged
## [1] 57
## initial  value 42.943935
## final  value 42.019132
## converged
## [1] 58
## initial  value 42.943630
## final  value 42.020379
## converged
## [1] 59
## initial  value 42.934012
## final  value 42.013462
## converged
```

```
## [1] 60
## initial  value 42.863794
## final  value 42.020089
## converged
## [1] 61
## initial  value 42.868654
## final  value 42.019106
## converged
## [1] 62
## initial  value 42.970900
## final  value 42.021434
## converged
## [1] 63
## initial  value 42.789104
## final  value 42.020662
## converged
## [1] 64
## initial  value 42.771953
## final  value 42.018245
## converged
## [1] 65
## initial  value 42.990417
## final  value 42.019265
## converged
## [1] 66
## initial  value 42.888614
## final  value 42.020373
## converged
## [1] 67
## initial  value 42.996254
## final  value 42.019103
## converged
## [1] 68
## initial  value 42.894285
## final  value 42.013748
## converged
## [1] 69
## initial  value 43.052778
## final  value 42.021041
## converged
## [1] 70
## initial  value 43.051055
## final  value 42.021426
## converged
## [1] 71
## initial  value 42.861060
## final  value 41.995928
## converged
## [1] 72
## initial  value 42.759179
## final  value 42.019768
## converged
## [1] 73
## initial  value 42.934075
```

```
## final  value 42.019698
## converged
## [1] 74
## initial  value 42.942535
## final  value 42.019600
## converged
## [1] 75
## initial  value 43.042221
## final  value 42.020133
## converged
## [1] 76
## initial  value 42.972011
## final  value 42.009422
## converged
## [1] 77
## initial  value 43.039385
## final  value 42.020430
## converged
## [1] 78
## initial  value 42.890008
## final  value 42.019878
## converged
## [1] 79
## initial  value 43.090805
## final  value 42.017256
## converged
## [1] 80
## initial  value 42.715275
## final  value 42.018922
## converged
## [1] 81
## initial  value 43.213033
## final  value 42.021549
## converged
## [1] 82
## initial  value 43.079342
## final  value 42.021134
## converged
## [1] 83
## initial  value 42.886228
## final  value 42.020646
## converged
## [1] 84
## initial  value 42.877294
## final  value 42.017529
## converged
## [1] 85
## initial  value 42.775804
## final  value 42.020842
## converged
## [1] 86
## initial  value 42.746433
## final  value 42.017813
## converged
```

```
## [1] 87
## initial  value 43.112464
## final  value 42.020849
## converged
## [1] 88
## initial  value 42.955994
## final  value 42.011927
## converged
## [1] 89
## initial  value 42.961478
## final  value 42.018424
## converged
## [1] 90
## initial  value 42.933208
## final  value 42.020408
## converged
## [1] 91
## initial  value 43.069875
## final  value 42.021091
## converged
## [1] 92
## initial  value 42.915391
## final  value 42.019763
## converged
## [1] 93
## initial  value 42.778625
## final  value 42.019810
## converged
## [1] 94
## initial  value 42.971306
## final  value 42.021128
## converged
## [1] 95
## initial  value 42.892994
## final  value 42.019242
## converged
## [1] 96
## initial  value 42.921936
## final  value 42.020790
## converged
## [1] 97
## initial  value 42.847356
## final  value 42.014680
## converged
## [1] 98
## initial  value 42.853776
## final  value 42.016011
## converged
## [1] 99
## initial  value 43.065253
## final  value 42.012248
## converged
## [1] 100
## initial  value 42.993168
```

```
## final  value 42.020889
## converged
```