Task 5

Task 5

Introduction

In this part of the assignment, we are going to work with a morse code signal perception dataset.

The dataset which contains a 36 x 36 matrix of confusion rates for all pairs of signals included in the study. The signals in the subsequent rows of the confusing matrix are the letters of the alphabet A, B, C… in alphabetical order and the numbers 1,2,3,4,5,6,7,8,9,0.

The value in cell (i,j) of the matrix represents the percentage of subjects who indicated that signal i was equal to signal j, when signal j was presented after signal i.

Here, we will search similarities and dissimilarities about the perception of the people which participated in this study.

Multidimensional Scaling will be use.

Data, looks like this

Table 1: extract of confussion matrix

	· -	- · · ·	- · - ·	- · ·	·	· · - ·	– ·	· · · ·
· -	92	4	6	13	3	14	10	13
- · · ·	5	84	37	31	5	28	17	21
- · - ·	4	38	87	17	4	29	13	7
- · ·	8	62	17	88	7	23	40	36
·	6	13	14	6	97	2	4	4
· · - ·	4	51	33	19	2	90	10	29
– ·	9	18	27	38	1	14	90	6
· · · ·	3	45	23	25	9	32	8	87

Methodology

In order to observe dissimilarities and understand them, we are going to work with in this matrix and find a solution, some methods as *randomstress()* and permutation() will be used.

This data is "Tree-way two-mode", that means a that multiple persons rate the similarity between all pairs of objects, in this case, that people rate if two signal are similar or not.

Firstly, and this one is to simplify the analysis, is transform signals into alphabet and numbers. After that is transform data to create a new symmetric dissimilarity matrix.

Secondly, we will Use MDS to get the optimal staled dissimilarities in order to select the better model to use in the analysis based in the goodness of fit and investigate the stability of the selected solution.

And Finally, Interpret the results

Results

Step 1

Here, we use couple of simple steps to help the interpretation of our analysis.

```
#A readable Matrix
alphabet <- c(letters,1:9,0)
translate <- confusion
colnames(translate) <- alphabet
rownames(translate) <- alphabet
```

Step 2

After that, a transformation matrix have to be applied.

```
#Looking a dissimilarity Matrix
#1.Similarity
t.sim <- (translate+t(translate))/2
#2.Dissimilarity
t.dissim <- 100-t.sim
for (i in 1:36){
#3.Diagonal
  t.dissim[i,i]<-0
}
```

Step 3

Using the *smacofSym()* conduct MDS with 2 dimensions in order to "process.." and check different measurement levels, ratio, interval, mspline and ordinal are going to be use, and avoiding to get stuck about the local minimum, will be use configuration of *torgeson* as a classical scaling starting solution.

```
#ratio
t.m1 <- smacofSym(delta=t.dissim, ndim=2, type="ratio", init="torgerson")
#intervalas
t.m2 <- smacofSym(delta=t.dissim, ndim=2, type="interval", init="torgerson")
#mspline
t.m3 <- smacofSym(delta=t.dissim, ndim=2, type="mspline", spline.degree =4,
                spline.intKnots = 4, init="torgerson")
#ordinal
t.m4 <- smacofSym(delta=t.dissim, ndim=2, type="ordinal", init="torgerson")
```

Once obtain the evaluation of *************** we proceed to apply our first judgment.

Table 2: Stress-1 table values

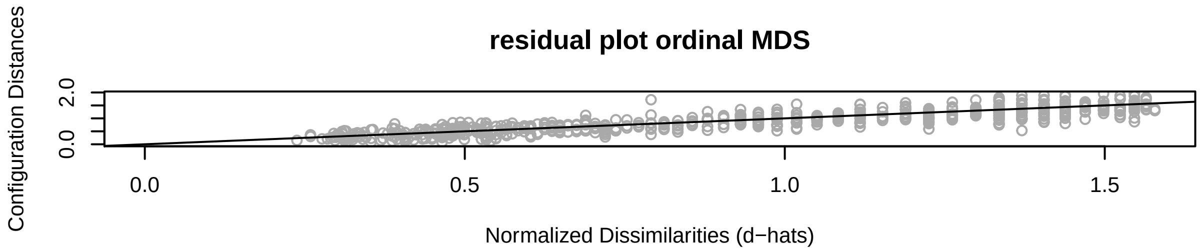
ratio	interval	mspline	ordinal
0.3	0.262	0.205	0.191

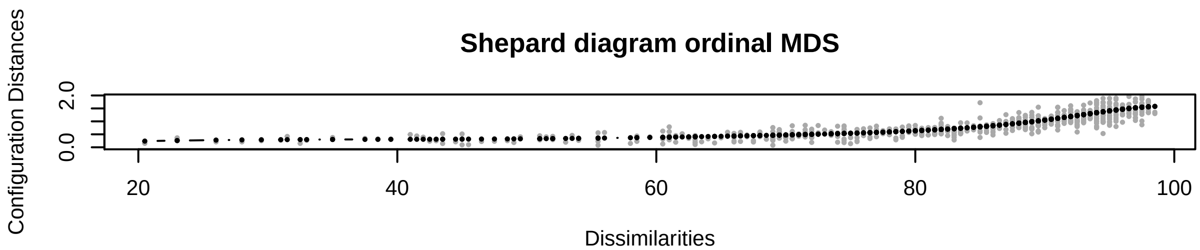
Following this rule

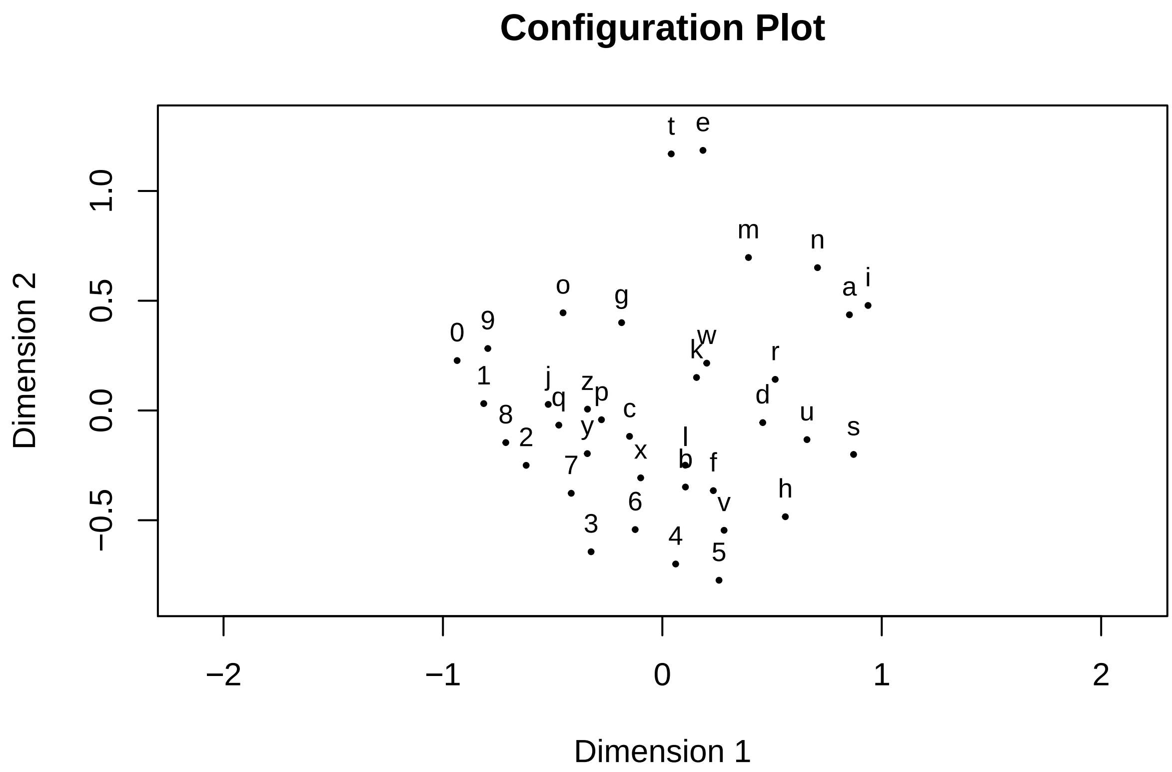
Our first decision is "ordinal" has to be used, because is a better value among those.

Table 3: Rule Of thumb

Perfect	Perfect	Good	Fair	Poor
0	0.025	0.05	0.10	0.20







Step 4

Here, we are going to evaluate the goodness of fit of solutions for those measurement levels using *stress-1* In the previous step, the minor value was the "ordinal", for that reason will be use to evaluate the goodness of fit.

In the process evaluate the goodness of fit of solution, we will observe *stress-1*.

Randomstress function help to understand our parameter which has to be compared with…………********.

Seed is setted in 290685

Table 4: Stress for random data

dsrd
0.3436171

Our first step is use permute test to put our data inside …

```
#permutation test
set.seed(290685)
t.perm.morse<-permtest(t.m4,nrep=500,verbose = FALSE)
```

Graphic information related previous two steps.

```
#plot distribution stress
par(mfrow=c(2,1),pty="s")
hist(rstress,main="stress random data")
hist(t.perm.morse$stressvec, main="stress permuted data")
```

Considering that our number... was... this fit as a good model.

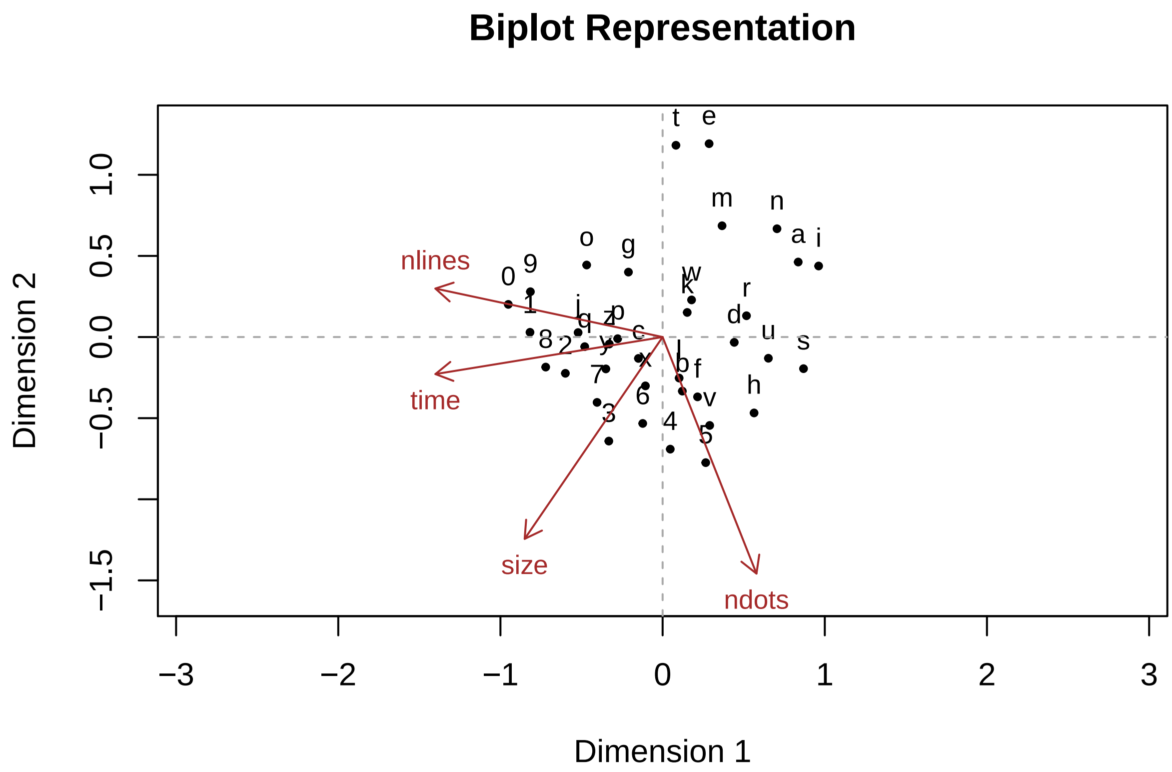
```
#stability of solution using jackknife
jack.morse<-jackmds(t.m3)
plot(jack.morse,xlim=c(-1.2,1.2),ylim=c(-1,1))

boot.morse.p <- bootmds(t.m3, data = dissim, nrep = 500, method.dat = "pearson")
plot(boot.morse.p)
```

Step 5

A external matrix which will help to understand some similarities and dissimilarities is created for this step

```
v_signal <- colnames(confusion)
v_signal <- str_replace_all(v_signal, " ", "")
v_signal <- str_replace_all(v_signal, "·", ".")
alphabet <- c(letters,1:9,0)
another_matrix <- data.frame(v_signal) %>% mutate(alph = alphabet,
         size = nchar(v_signal),
         ndots =  str_count(v_signal, "\\."),
         nlines = str_count(v_signal, "-"),
         time = (str_count(v_signal, "\\.")*0.05)+(str_count(v_signal,"-")*0.15),
         t_s_length = (nchar(v_signal) -1 )*0.05,
         dots_s_length = (str_count(v_signal,"\\.")*0.05),
         lines_s_length = (str_count(v_signal,"-")*0.15))
```

```
#conduct ordinal MDS analysis
t.fit <- mds(t.dissim, type = "ordinal")
#compute MDS biplot: run multivariate linear regression of
#external variables on configuration
biFace <- biplotmds(t.fit, another_matrix[,-c(1,2,7,8,9)])
# project external variables in the MDS solution
plot(biFace, main = "Biplot Representation", vecscale = 0.8,
xlim = c(-1.5, 1.5), vec.conf = list(col = "brown"), pch = 20, cex = 0.7)
```



3. Construct a data set of external variables that describe the signals (e.g. length of the signal, proportion of short beeps in the signal, etc.). Use an *MDS biplot* to project the external variables in the configuration plot of the selected solution, and interpret the results of the analysis.