

# MDS Analysis

Danielle Daidone

8/3/23

This script runs multidimensional scaling analyses on any matrix output by `create_FC_similarity_matrices.Rmd`. It outputs 1,2,3,4 and 5 dimensional solutions as text files and produces a stress plot to help you decide on the most appropriate number of dimensions. It also produces plots of the solutions, but note that the solutions are unrotated and thus may not be easily interpretable.

Download and load relevant packages See documentation: `smacof` (for MDS): <https://rdr.io/cran/smacof/man/smacofSym.html> (<https://rdr.io/cran/smacof/man/smacofSym.html>)

```
if(!require(rmarkdown)){install.packages("rmarkdown")}
```

```
## Loading required package: rmarkdown
```

```
library(rmarkdown)
if(!require(smacof)){install.packages("smacof")}
```

```
## Loading required package: smacof
```

```
## Loading required package: plotrix
```

```
## Loading required package: colorspace
```

```
## Loading required package: e1071
```

```
##
## Attaching package: 'smacof'
```

```
## The following object is masked from 'package:base':
##
##   transform
```

```
library(smacof)
```

## PART 1: OPEN AND CHECK DATASET

Set working directory - Change to match your filepath

Open dataset - Change to match your desired *dissimilarity* matrix

```
# Note that the dissimilarity text files are tab delimited files with headers
FCmatrixAll=read.table("Output_Matrix_AllContexts_percent_dis.txt",sep="\t",header=TRUE)
```

Look at the first few rows of the dataset

```
head(FCmatrixAll)
```

```
##          cVccV_L  cVccV_M  cVccV_N  cVccVV_L  cVccVV_M  cVccVV_N  cVcV_L
## cVccV_L  0.0000000  0.7564103  0.6666667  0.8461538  0.8589744  0.8205128  0.7564103
## cVccV_M  0.7564103  0.0000000  0.6666667  0.9615385  0.6025641  0.9487179  0.3974359
## cVccV_N  0.6666667  0.6666667  0.0000000  0.8717949  0.8846154  0.8076923  0.7692308
## cVccVV_L 0.8461538  0.9615385  0.8717949  0.0000000  0.8589744  0.5897436  0.9615385
## cVccVV_M 0.8589744  0.6025641  0.8846154  0.8589744  0.0000000  0.8333333  0.7307692
## cVccVV_N 0.8205128  0.9487179  0.8076923  0.5897436  0.8333333  0.0000000  0.9102564
##          cVcV_M  cVcV_N  cVcVV_L  cVcVV_M  cVcVV_N  cVVccV_L  cVVccV_M
## cVccV_L  0.8076923  0.7692308  0.8461538  0.9102564  0.9102564  0.7692308  0.8589744
## cVccV_M  0.4102564  0.5256410  0.8205128  0.6666667  0.9230769  0.9871795  0.8589744
## cVccV_N  0.7435897  0.7307692  0.9102564  0.9230769  0.9615385  0.7820513  0.7564103
## cVccVV_L 0.9743590  0.9871795  0.6410256  0.8333333  0.6666667  0.8974359  0.8846154
## cVccVV_M 0.7692308  0.8333333  0.7564103  0.5384615  0.8333333  0.9230769  0.7564103
## cVccVV_N 0.9615385  0.9230769  0.6410256  0.8205128  0.5641026  0.8846154  0.9102564
##          cVVccV_N cVVccVV_L cVVccVV_M cVVccVV_N  cVVcV_L  cVVcV_M  cVVcV_N
## cVccV_L  0.7820513  0.8076923  0.8461538  0.8717949  0.8717949  0.8461538  0.8461538
## cVccV_M  0.9487179  0.9102564  0.8589744  0.9102564  0.9743590  0.9102564  0.9487179
## cVccV_N  0.7179487  0.8846154  0.8846154  0.8461538  0.8717949  0.9230769  0.8205128
## cVccVV_L 0.9102564  0.7179487  0.8333333  0.7948718  0.8717949  0.8717949  0.8461538
## cVccVV_M 0.9358974  0.8974359  0.7820513  0.8846154  0.9102564  0.8461538  0.9230769
## cVccVV_N 0.8461538  0.8717949  0.8974359  0.7820513  0.8717949  0.9230769  0.7692308
##          cVVcVV_L  cVVcVV_M  cVVcVV_N
## cVccV_L  0.9487179  0.8974359  0.8717949
## cVccV_M  0.8974359  0.8205128  0.9230769
## cVccV_N  0.8974359  0.7948718  0.8846154
## cVccVV_L 0.7435897  0.8974359  0.8846154
## cVccVV_M 0.9358974  0.8461538  0.8589744
## cVccVV_N 0.9743590  0.9230769  0.7435897
```

Run ordinal multi-dimensional scaling (MDS) - 1 dimensional solution

```
# ndim is the number of dimensions
MDS1Dim = mds(FCmatrixAll, ndim = 1, type = "ordinal")
```

```
# view Kruskal stress
MDS1Dim
```

```
##
## Call:
## mds(delta = FCmatrixAll, ndim = 1, type = "ordinal")
##
## Model: Symmetric SMACOF
## Number of objects: 24
## Stress-1 value: 0.373
## Number of iterations: 8
```

```
# get dim scores
dimscores = MDS1Dim$conf

# write dim scores out to a text file in your working directory
write.table(dimscores, file="MDS_1D.txt", sep="\t", quote=FALSE)
```

## Run ordinal multi-dimensional scaling (MDS) - 2 dimensional solution

```
# ndim is the number of dimensions
MDS2Dim = mds(FCmatrixAll, ndim = 2, type = "ordinal")

# view Kruskal stress
MDS2Dim
```

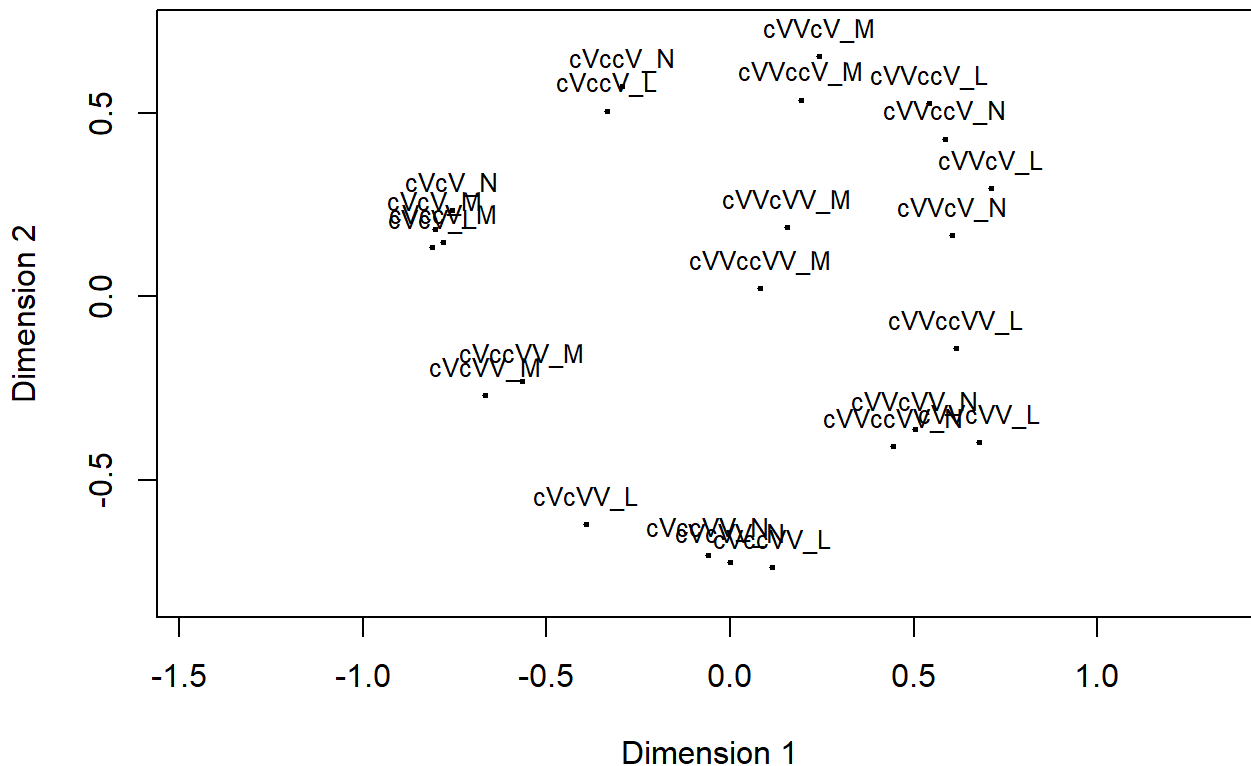
```
##
## Call:
## mds(delta = FCmatrixAll, ndim = 2, type = "ordinal")
##
## Model: Symmetric SMACOF
## Number of objects: 24
## Stress-1 value: 0.183
## Number of iterations: 30
```

```
# get dim scores
dimscores = MDS2Dim$conf

# write dim scores out to a text file in your working directory
write.table(dimscores,file="MDS_2D.txt",sep="\t",quote=FALSE)

#plot
plot(MDS2Dim, plot.type = "confplot", plot.dim = c(1,2))
```

### Configuration Plot



## Run ordinal multi-dimensional scaling (MDS) - 3 dimensional solution

```
# ndim is the number of dimensions
MDS3Dim = mds(FCmatrixAll, ndim = 3, type = "ordinal")
```

```
# view Kruskal stress
MDS3Dim
```

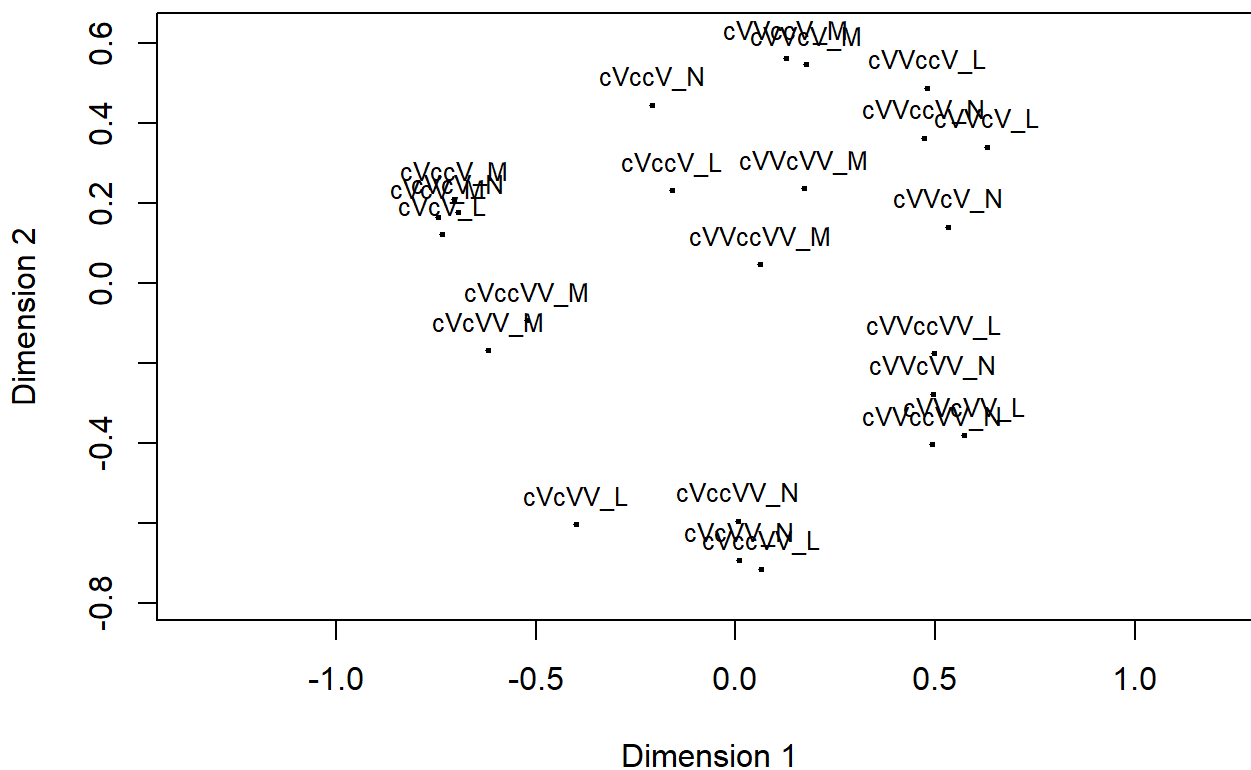
```
##
## Call:
## mds(delta = FCmatrixAll, ndim = 3, type = "ordinal")
##
## Model: Symmetric SMACOF
## Number of objects: 24
## Stress-1 value: 0.115
## Number of iterations: 27
```

```
# get dim scores
dimscores = MDS3Dim$conf

# write dim scores out to a text file in your working directory
write.table(dimscores,file="MDS_3D.txt",sep="\t",quote=FALSE)

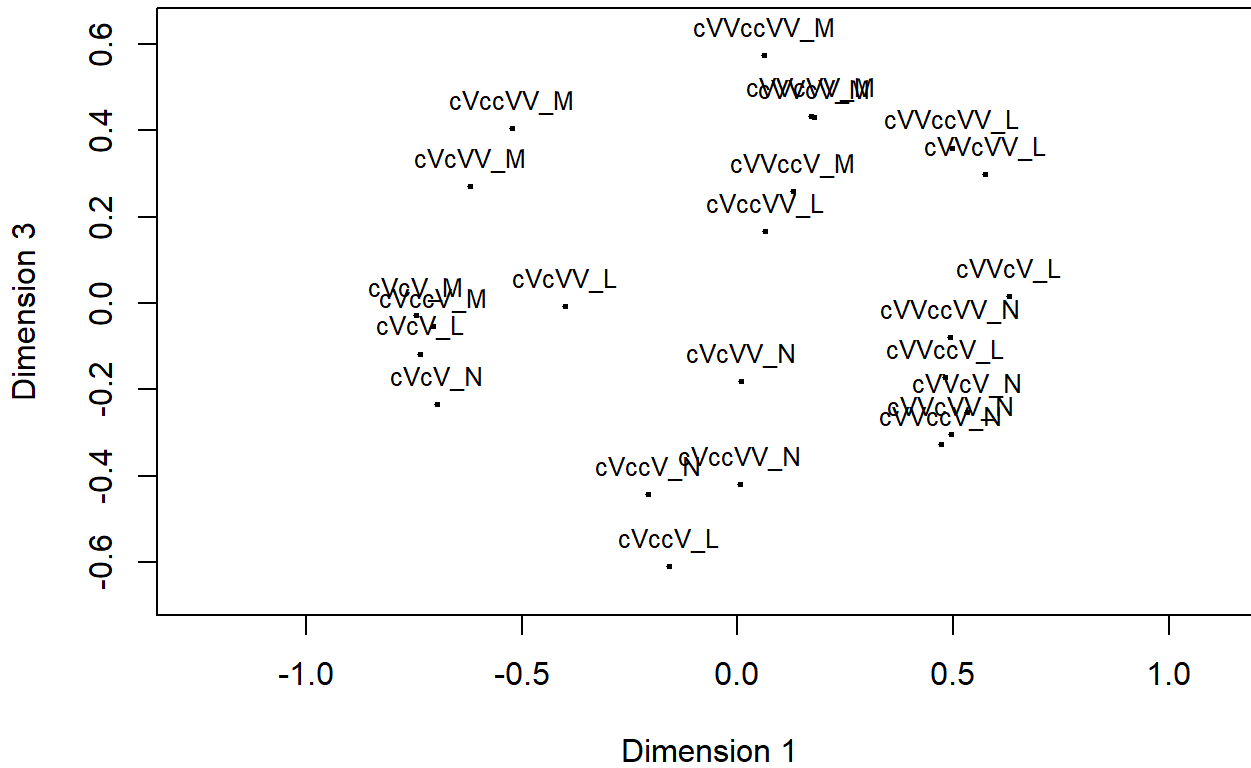
#plot dim 1 by dim 2
plot(MDS3Dim, plot.type = "confplot", plot.dim = c(1,2))
```

### Configuration Plot



```
#plot dim 1 by dim 3 (as if looking into previous plot from the top)
plot(MDS3Dim, plot.type = "confplot", plot.dim = c(1,3))
```

## Configuration Plot



Run ordinal multi-dimensional scaling (MDS) - 4 dimensional solution

```
# ndim is the number of dimensions
MDS4Dim = mds(FCmatrixAll, ndim = 4, type = "ordinal")
```

```
# view Kruskal stress
MDS4Dim
```

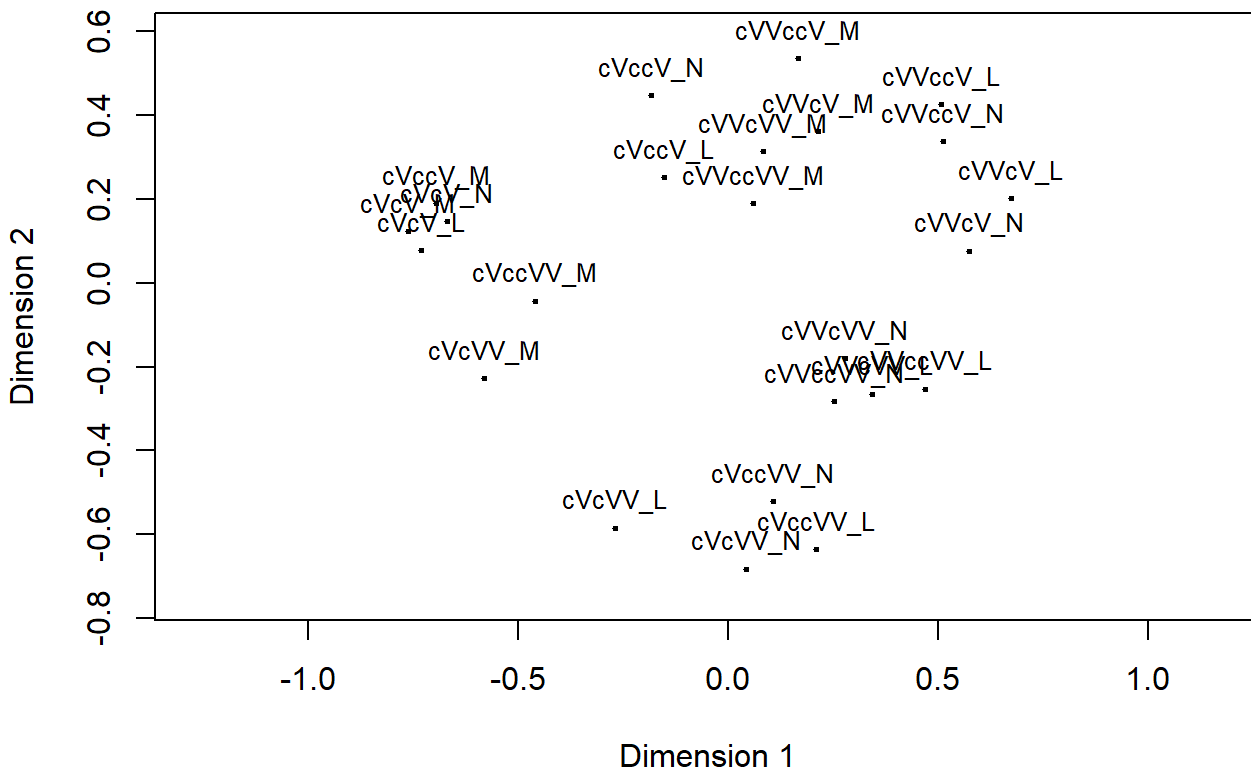
```
##
## Call:
## mds(delta = FCmatrixAll, ndim = 4, type = "ordinal")
##
## Model: Symmetric SMACOF
## Number of objects: 24
## Stress-1 value: 0.081
## Number of iterations: 105
```

```
# get dim scores
dimscores = MDS4Dim$conf

# write dim scores out to a text file in your working directory
write.table(dimscores, file="MDS_4D.txt", sep="\t", quote=FALSE)

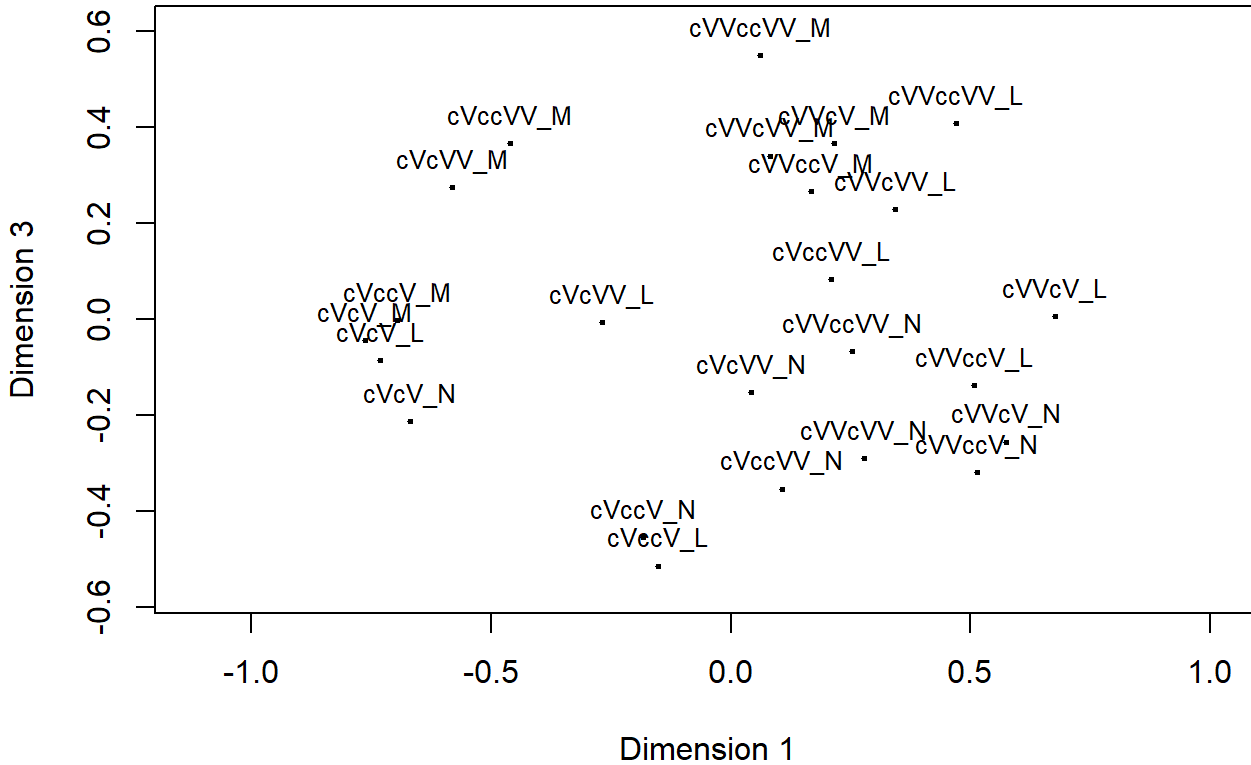
#plot dim 1 by dim 2
plot(MDS4Dim, plot.type = "confplot", plot.dim = c(1,2))
```

### Configuration Plot



```
#plot dim 1 by dim 3 (as if looking into previous plot from the top)  
plot(MDS4Dim, plot.type = "confplot", plot.dim = c(1,3))
```

### Configuration Plot



## Run ordinal multi-dimensional scaling (MDS) - 5 dimensional solution

```
# ndim is the number of dimensions
MDS5Dim = mds(FCmatrixAll, ndim = 5, type = "ordinal")

# view Kruskal stress
MDS5Dim
```

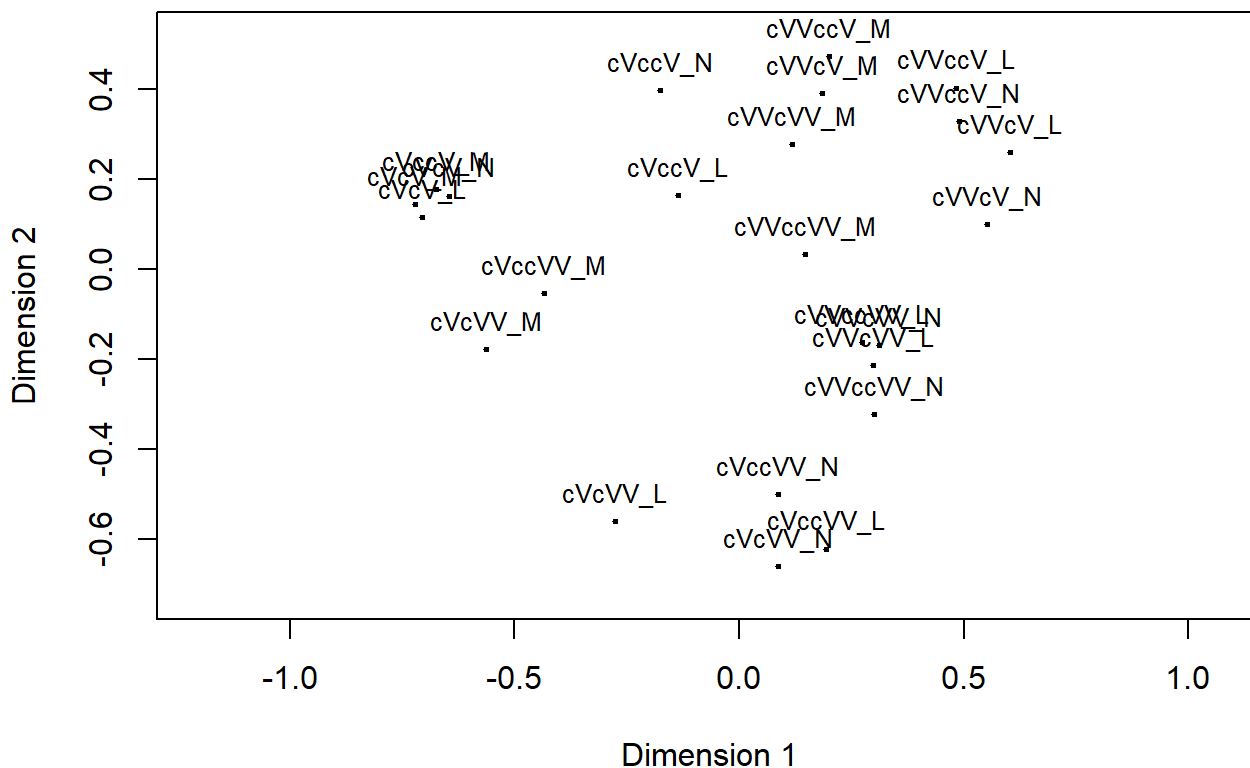
```
##
## Call:
## mds(delta = FCmatrixAll, ndim = 5, type = "ordinal")
##
## Model: Symmetric SMACOF
## Number of objects: 24
## Stress-1 value: 0.056
## Number of iterations: 73
```

```
# get dim scores
dimscores = MDS5Dim$conf

# write dim scores out to a text file in your working directory
write.table(dimscores,file="MDS_5D.txt",sep="\t",quote=FALSE)

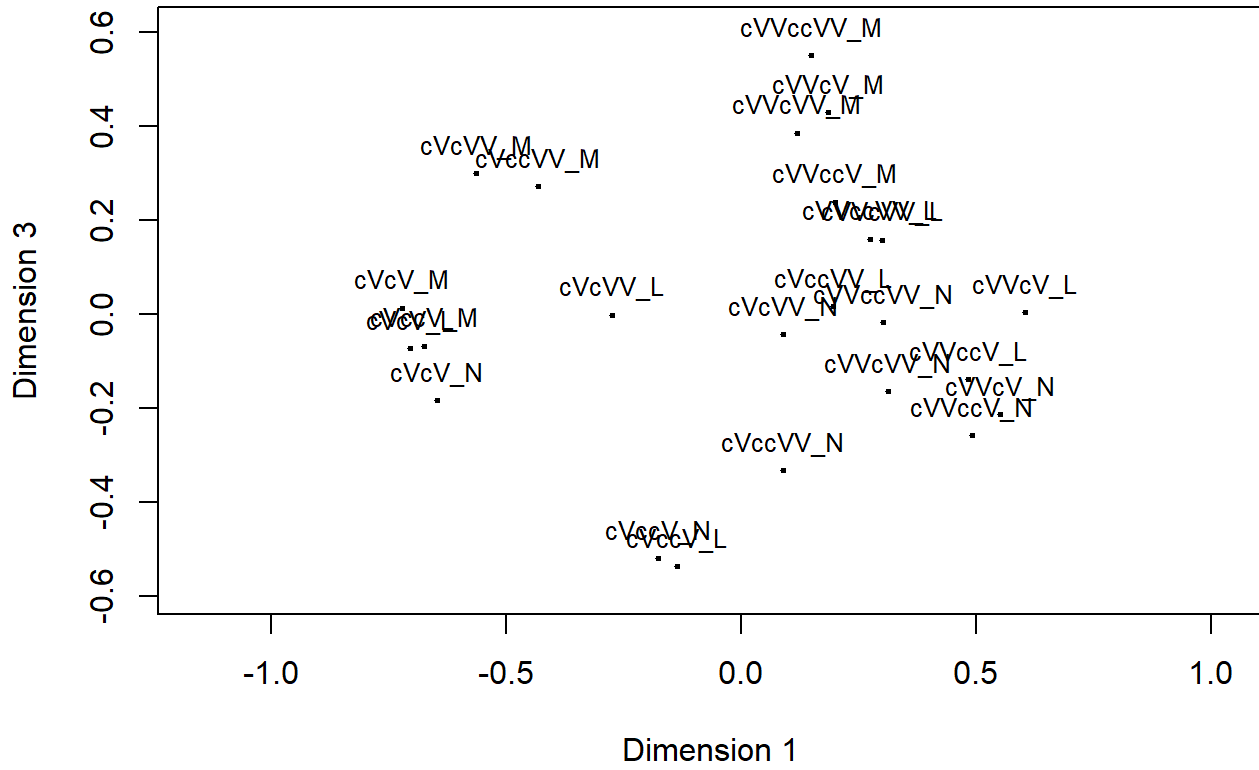
#plot dim 1 by dim 2
plot(MDS5Dim, plot.type = "confplot", plot.dim = c(1,2))
```

### Configuration Plot



```
#plot dim 1 by dim 3 (as if looking into previous plot from the top)
plot(MDS5Dim, plot.type = "confplot", plot.dim = c(1,3))
```

## Configuration Plot



Create stress plot

```
stressvalues = c(MDS1Dim$stress,MDS2Dim$stress,MDS3Dim$stress,MDS4Dim$stress,MDS5Dim$stress)
dimensions = c(1,2,3,4,5)
plot(dimensions, stressvalues)
```



