

## *Supplemental Text 2: R-statistics Commands.*

### **Example 1: Multidimensional Scaling: Generating 2 and 3 Dimensional MDS**

#### **Plots for eHRAF Cultures**

```
Database1 <- read.csv

#insert file-path; in this example Database1 is the .csv file with the ehraf data

Database1 [1,]

library(cluster)

for (i in 2:35){ Database1 [,i] <- factor(Database1 [,i])}

dis <- daisy(Database1[,2:35], metric = "gower") # creates dissimilarity matrix using Gower distances

pl.2 <- cmdscale(dis) # employs multidimensional scaling

plot(pl.2) # shows the plotted graph

identify(pl.2) # enables you to identify individual cases on the plot

attach(Database1)

dim(Database1)

code <- c(1:35)*0

code[Database1[,2]=="Doc"] <- "red"

code[Database1[,2]=="Img"] <- "green" #enables you to colour the categories in column 2

plot(pl.2, col = code) # allows you to plot the colours in terms of the colour coding

identify(pl.2)

library(rgl) # opens the package for 3D real-time rendering system

pl.3 <- cmdscale(dis, k=3) # employs multidimensional scaling

plot3d(pl.3, col = code) #give you the 3-d plot
```

```

plot(pl.2, col = code, main = "Ehraf Cultures", xlab = "PC 1", ylab = "PC 2", sub = "Mode coloured")
legend = c("Doc ", "Img")
codecol <- c("red", "green")
plot(pl.2, col = code, main = " Ehraf Cultures", xlab = "PC 1", ylab = "PC 2", sub = "Mode coloured",
ylim = c(-0.2, 0.4))
legend("topleft", legend = legend, col = codecol, pch = 1)
plot3d(pl.3, col = code,size=5) # plot with colour code, legend and labels the X and Y axis

```

## **Example 2: Principal Component Analysis (PCA): Generating Correlation Circles and Factor Maps for the Archaeological Site-Phases**

```

library(FactoMineR) # Loading package for exploratory data analysis methods to summarize, visualize
and describe datasets—in this case PCA

```

```

Database2 <- read.csv<- read.csv

```

```

#insert file-path; in this example Database2 is the .csv file with the archaeological data

```

```

View(Database2)

```

```

res<-PCA(Database2,quali.sup=1)

```

```

#performs PCA and makes labelling categories into supplementary variables; gives initial plots

```

```

plot(res,choix="var") # re-doing plot—variable factor map and correlation circle

```

```

plot(res,choix="var",shadow=TRUE) # making a clearer plot, if necessary, without shadow

```

```

plot(res,choix="var",shadow=TRUE,select="contrib5")

```

```

# plots with the 5 variables responsible for the max variance

```

```

plot(res,cex=0.8,shadow=TRUE,habillage=1) # give factor map with supplementary categories labelled

```

```

summary(res)

```

### **Example 3: Generalised Linear Modelling (GLM): Generating Percentage Probabilities for the Archaeological Site-Phases**

```
Database3 <- read.csv

#insert file-path; in this example Database3 is the .csv file with the combined archaeology and ehraf data

dim(Database3)

summary(Database3[,2])

Database3R <- Database3[Database3[,2]=="Doc",] #Creates a dataset with just the doctrinal cultures
Database3R2 <- Database3[Database3[,2]=="Img",] #Creates a dataset with just the Imagistic cultures

Database3R <- rbind(Database3R,Database3R2) #binds them both onto one dataset

dim(Database3R)

DI <- c(1:35)*0 + 1 # Creates a vector of ones

DI[Database3R[,2]=="Img"] <- 0 # replace 1 by zero where col 2 of Database3R has Imagistic

PG <- c(1:35)*0
PG[ehneR$Pop.Group==2] <- 1

Database3R <- data.frame(Database3R,DI,PG)

mod1 <- glm(DI ~ PCA1 + PCA2 + PCA3 + PCA4 + PCA5, family = binomial, data = Database3R)

summary(mod1)

#Fits binary GLM using the identified variables note "PCA1 + PCA2 + PCA3 + PCA4 + PCA5" refer to
#whichever set of variables identified following PCA

Generating Receiver Operating Characteristic (ROC) curve

fit <- fitted(mod1) # This gets vector of probabilities of being classified Imagistic
library(ROCR)

pred <- prediction(fit,as.vector(DI))
```

```

perf <- performance(pred,"tpr","fpr")
plot(perf)
Database3U <- Database3[1:99,] # Identifies the unknowns that need classifying
PG <- c(1:99)*0
PG[Database3U$Pop.Group==2] <- 1
Database3U <- data.frame(Database3U,PG)
Unknown <- predict(mod1, newdata = Database3U)
summary(Unknown)

```

### ***Generating the probabilities for each case***

```

ProbU <- exp(Unknown)/(1 + exp(Unknown))
Classification <- c(1:99)*0
Classification[ProbU >= 0.5] <- 1
cbind(Database3U[,1:2], ProbU, Classification) #should get the sites and not the mode cultures
cbind(Database3U[,1:2], round(ProbU, digits = 3), Classification) ) # probability prediction into 3 digits
tab <- cbind(Database3U[,1:2], round(ProbU, digits = 3), Classification)

```

### ***Calculating the standard error***

```

siteprobsN <- data.frame(Site = Database3U$Site.cult,Prob.Site= probs, se.Site = s.e.p)
siteprobsN

```

### ***Calculating the Area Under the Curve (AUC)***

```

lines(c(0,1),c(0,1), lty = 2)
df <- data.frame(fpr=perf@x.values, tpr=perf@y.values)
colnames(df) <- c("x", "y")
nr <- dim(df)[1]
z <- (1 - df[1,2])*(df[1,1]-0)

```

```
for (i in 2:nr){z <- c(z,(1 - df[i,2])*(df[i,1]-df[(i-1),1]))}
```

```
A <- 0.5 - sum(z)
```

```
A
```

### ***Generating a Half Normal Q-Q Plot***

```
mod1 <- glm(cbind(DI, 1-DI) ~ PCA1 + PCA2 + PCA3 + PCA4 + PCA5, family = binomial, data =
```

```
Database3R)
```

```
# PCA1 + PCA2 + PCA3 + PCA4 + PCA5 refer to whichever set of variables identified following PCA
```

```
summary(mod1)
```

```
library("binomTools", lib.loc="C:/Users/Owner/Documents/R/win-library/3.0")# opening the binomTools
```

```
package
```

```
halfnorm(mod1) # plot
```