Stats506: hw2_2

David Li

October 22, 2017

Stats506: Problem 2 Code
Data Used: Professor Henderson's Focal Behavior Data from Problem Set
Submitted by: David Li

```
# Read in the .csv files for focal and name data
interactions = read.csv("~/Desktop/Stats506/Datasets/Stats506_F17_ps2_interactions.csv")
focalnames = read.csv("~/Desktop/Stats506/Datasets/Stats506_F17_ps2_focal_names.csv", header = FALSE)
allnames = read.csv("~/Desktop/Stats506/Datasets/Stats506_F17_ps2_all_names.csv", header = FALSE)
```

```
# We first need to count now many columns we will need to split the behavior recipients into,
#turns out we need 13 since we discovered 12 commas
commas = str_count(interactions$toward, pattern = ",")
maxcommas = max(commas, na.rm = TRUE)
columnsneeded = maxcommas + 1
newcolumns = paste('toward', 1:columnsneeded, sep = "")
allnames$V1 = as.character(allnames$V1)
```

```
# String matching and data cleaning
newtable = interactions %>%
  tidyr::separate(toward, into = newcolumns) %>% # Separate into multiple columns
  mutate_at(vars(toward1:toward13), funs(emptyasna)) # Get rid of blanks
```

```
## Warning: Too few values at 9339 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
## 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
```

```
  for (i in 3:15){ # String matching according to osa algorithm
    a = amatch(newtable[,i], allnames$V1, nomatch = NA_integer_, matchNA = FALSE, method = "osa", useBy
    a = matrix(a)
    newtable[,i] = allnames[a]
  }
# Preparation for counting frequencies, creating a big matrix to store the relationship between focals
interactions[,3:15]= newtable[,3:15] # Overwrite interactions file with corrected names
tabulation = interactions # Duplicate interactions file for utility
tabulation = subset(interactions,!duplicated(subset(interactions, select=c(focal,behavior_cat)))) # Fil
tabulation=arrange(tabulation,focal) # Arrange in Ascending Alphabetical order by Focal name
tabulation[,3:202]=matrix(0) # Add Columns of all focal individuals
colnames(tabulation)[3:201]=allnames$V1 # Replace Columns with meaningful names aka individual names
```

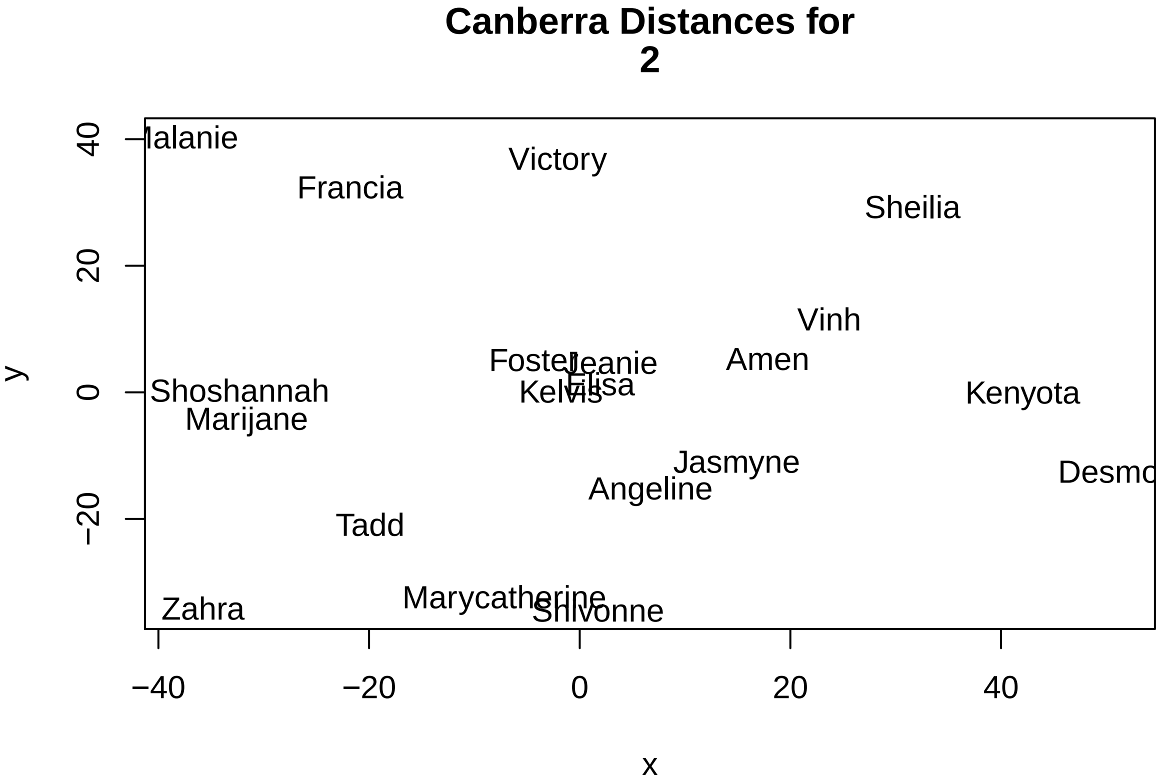
```
# Populating the matrix of frequencies
for(i in 1:140){ # Rows of all pairs of focals and their behavior combinations
  focal=as.character(tabulation[i,1]) # Keep focal name
  behavior_cat=as.character(tabulation[i,2]) # Keep behavior name
  for(j in 1:NROW(interactions)){ # For each column,
    if(interactions[j,1]==focal & interactions[j,2]==behavior_cat){
      for(k in 3:15){
        text=interactions[j,k] # Extract the number of occurences.
        if(is.na(text)){ # Safeguard against exceptions for when not encountering text

        }
        else {
          tabulation[i,text]=tabulation[i,text] + 1
        }
      }
    }
  }
}
tabulation= tabulation[,1:201] # Omitting an unnecessary column
tabulation[c(1:3), c(1:7)] # View the data
```

```
##   focal behavior_cat Jenessa Iban Senora Emilly Kandance
## 1  Amen         play       3    3      6      1        1
## 2  Amen         mate       0    1      0      1        0
## 3  Amen        groom       0    0      0      1        1
```

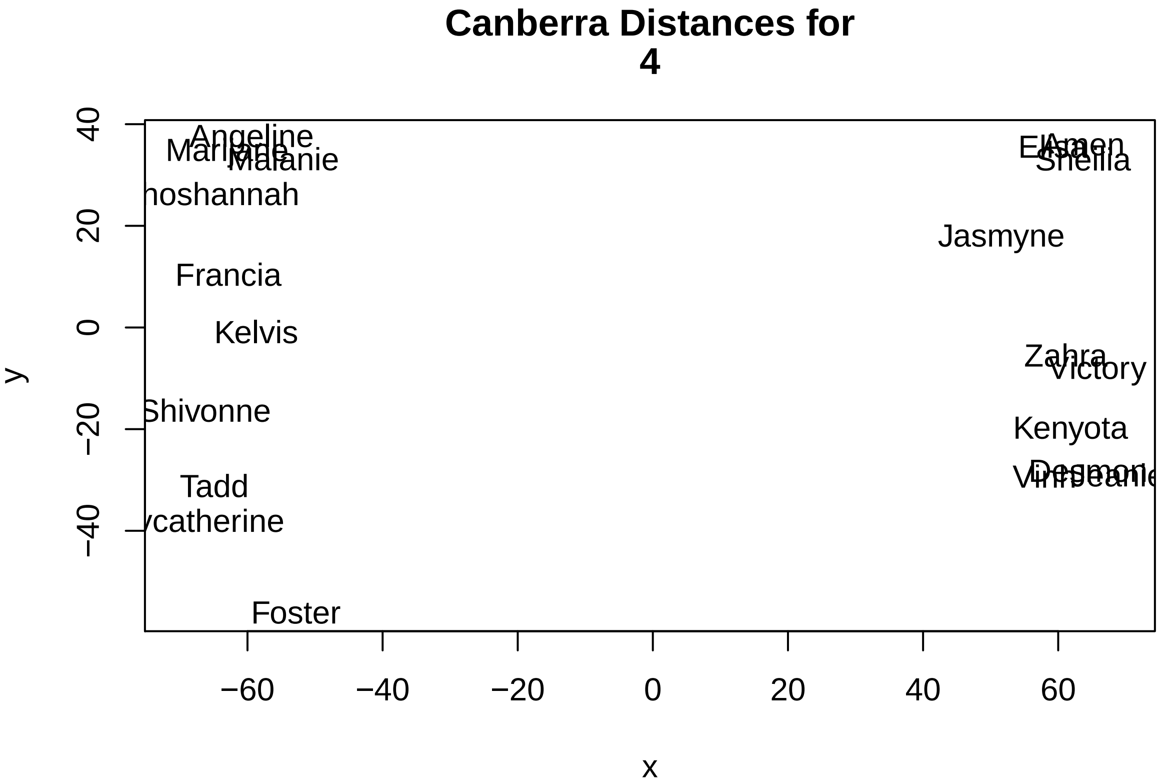
Above smaller chunk of the large [140 x 201] tabulation of focal individuals' behavior towards all of the other individuals based on the data.

```
# Computing canberra distances between individuals and some multi-dimensional displays for the distance
uniqueintertypes = unique(interactions$behavior_cat)
behave_vector = tabulation$behavior_cat
for(j in 1:7){
  interactiontype = tabulation[which(uniqueintertypes[j] == behave_vector),]
  interactiontype = interactiontype[,-c(1:2)]
  rownames(interactiontype) = unique(interactions$focal)
  dist_canberra = dist(interactiontype, method = "canberra")
  cmd = cmdscale(dist_canberra)
  x = cmd[,1]
  y = cmd[,2]
  print(uniqueintertypes[j])
  print(head(dist_canberra))
  plot(x,y,type = 'n', main = c("Canberra Distances for", uniqueintertypes[j]))
  text(x,y,rownames(cmd))
}
```

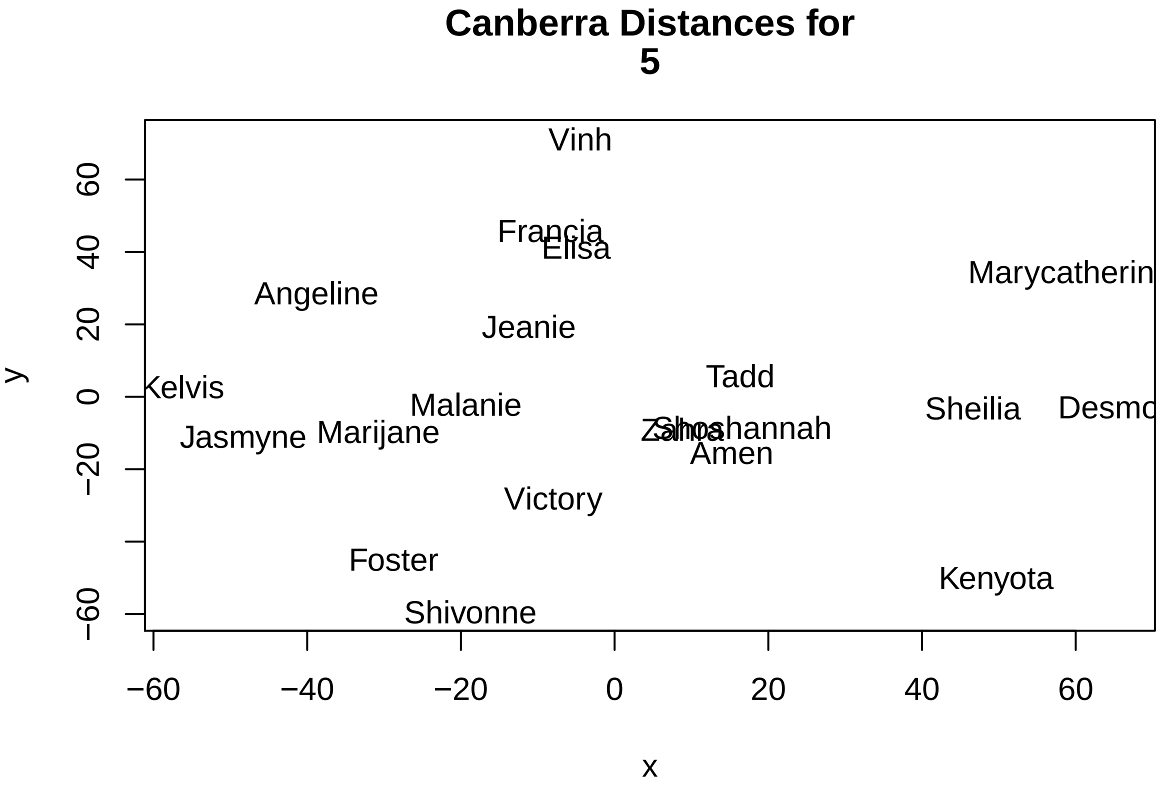
```
## [1] approach
## Levels: aggression approach carry groom mate play share
## [1] 116.9311 114.7883 106.6123 110.5129 122.0841 107.0421
```



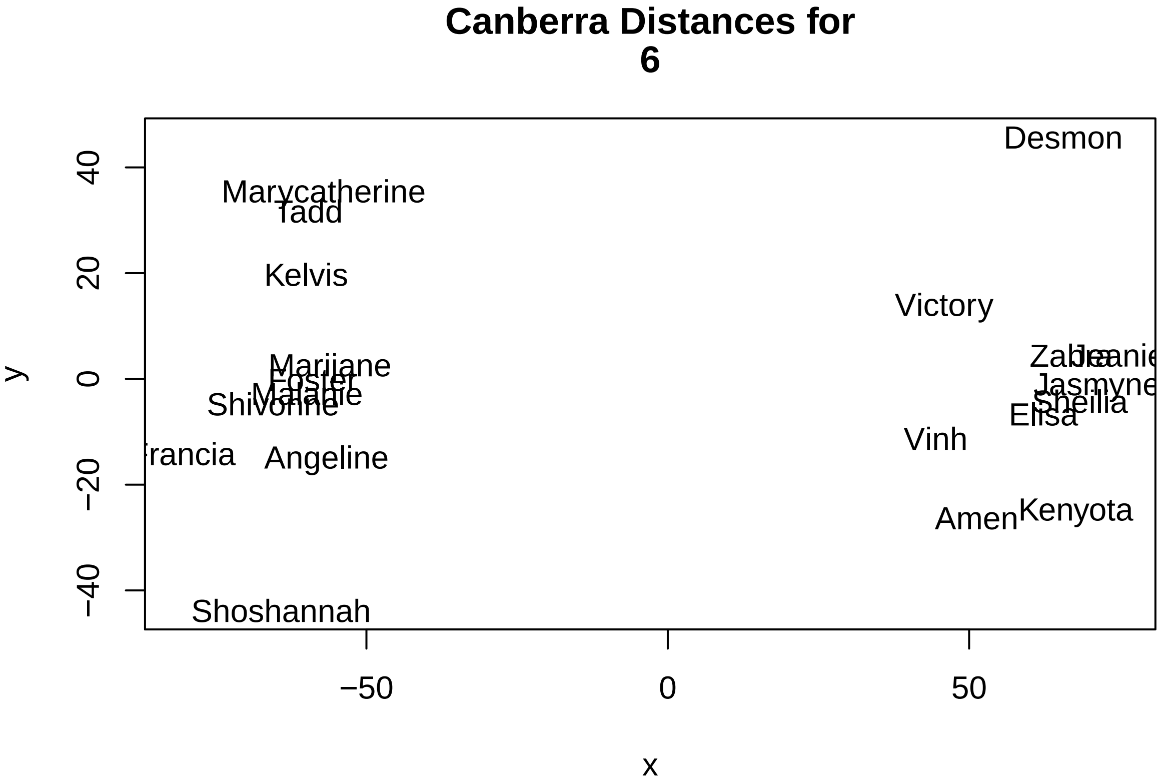
```
## [1] groom
## Levels: aggression approach carry groom mate play share
## [1] 186.8287 189.1728 155.1007 186.9394 193.4722 156.7602
```



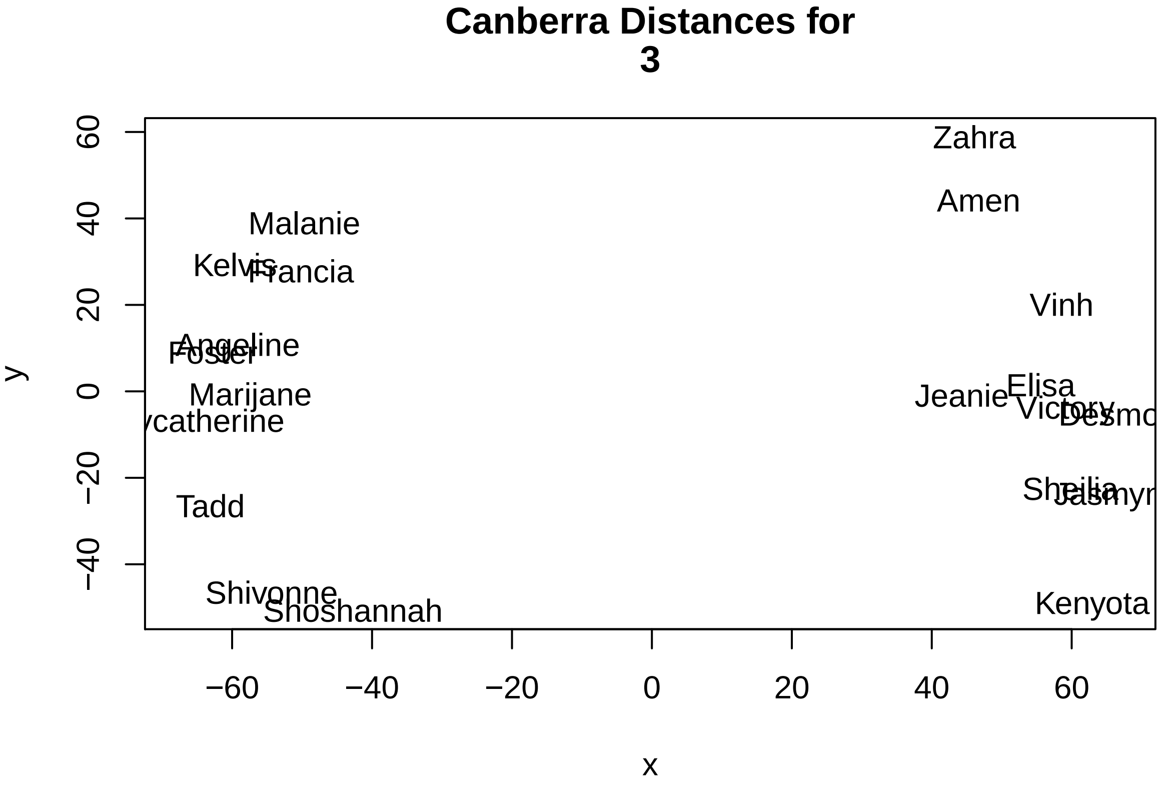
```
## [1] mate
## Levels: aggression approach carry groom mate play share
## [1] 174.2356 157.0340 178.7315 183.7375 172.4667 160.5401
```



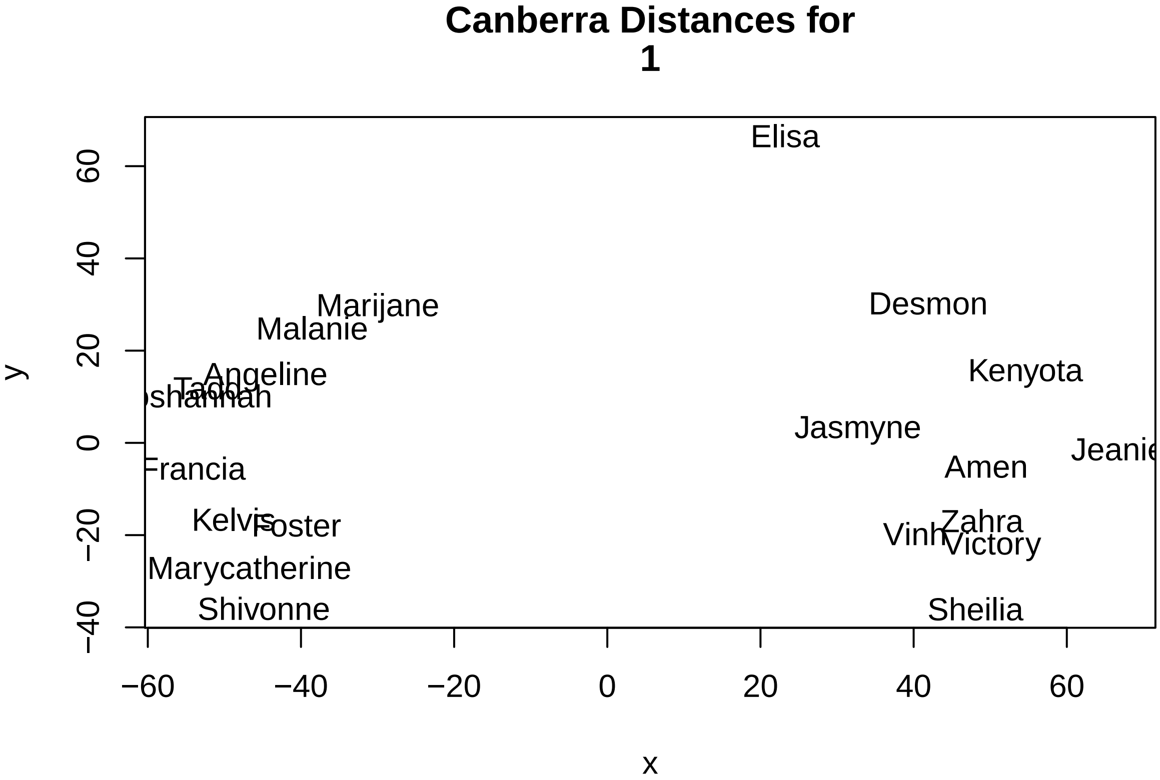
```
## [1] play
## Levels: aggression approach carry groom mate play share
## [1] 175.9169 168.2074 107.0556 170.9545 164.5473 108.4815
```



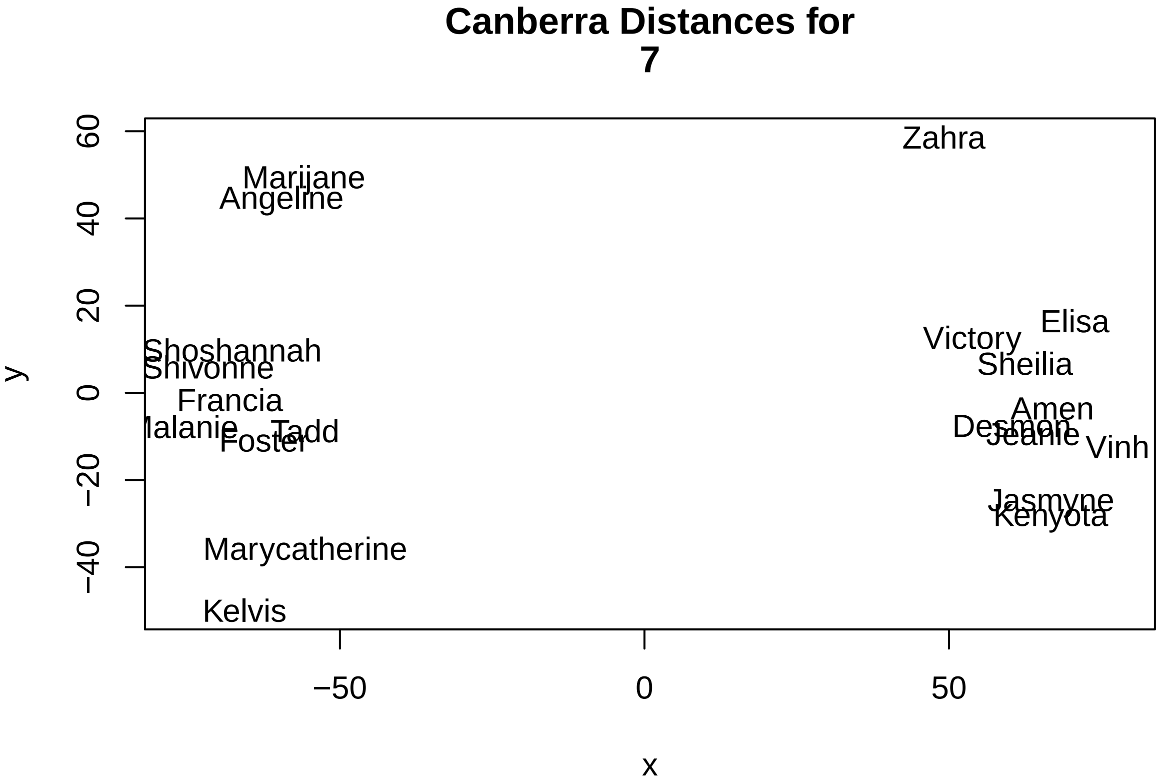
```
## [1] carry
## Levels: aggression approach carry groom mate play share
## [1] 192.9984 183.2063 151.7104 192.0175 188.4336 154.5957
```



```
## [1] aggression
## Levels: aggression approach carry groom mate play share
## [1] 157.5972 154.9825 124.0838 154.9217 149.7633 127.2063
```



```
## [1] share
## Levels: aggression approach carry groom mate play share
## [1] 180.3232 189.4922 134.4663 184.2026 183.3950 133.3752
```



```
## Appendix ##
## Packages Utilized ##
library("dplyr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("ggplot2", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("knitr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("rmarkdown", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("stringdist", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("stringr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("tidyr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")

# Begin Functions Section #
emptyasna <- function(x){ # Turns blanks into NAs
  ifelse(as.character(x) != "", x, NA)
}
# End Functions Section #
```