Our plant diet and intro to data visualization in R

Ryoko Oono, UCSB

3/12/2021

This week, we will introduce you to R and some simple commands to upload your data from a different application, such as GoogleSpreadsheet or Excel. We are going to use this opportunity to get to know each other as well! In the shared GoogleSpreadsheet, note the plants you ate in one day. Once everyone has had a chance to report their plant diet, download the spreadsheet as a .csv file.

```
#We have to upload several "packages" which will allow us to run functions written by other people.
knitr::opts_chunk$set(echo = TRUE)
#install.packages(c("FactoMineR", "factoextra", "data.table"))
library(FactoMineR)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-7
```

```
library(here) #I use this package to locate my path
```

```
## here() starts at /Users/ryokooono/Dropbox/R_projects/plantphysiology/FirstWeek
```

```
Class_plant_diet <- read.csv(here("plant_food.csv"), header=TRUE, row.names=1, sep=",")

head(Class_plant_diet) #If you just want to check out the beginning of the data...
```

```
##        Ryoko Jacob Maxi Amy Bob Chris Dana Eric Fiona Gabe Hanna Isaac
## coffee     1     1    1   1   1     1    1    1     1    0     1     0
## sugar      1     1    1   1   1     1    1    1     1    1     1     1
## wheat      1     1    1   1   1     1    1    1     1    1     1     1
## corn       1     1    1   1   1     1    1    1     1    1     1     1
## rice       1     0    0   0   0     0    0    0     0    0     0     0
## walnut     1     0    0   0   1     0    0    1     0    0     0     0
```

Today, we are going to introduce everyone to Principal Component Analysis - a statistical tool for dimension reduction and visualization of multivariate data (a n x m dataset like your .csv file). When the data is binary (yes and no's, 0 and 1's), you can use Multiple Correspondence Analysis.

We will also convert this dataset (n x m) into a dissimilarity matrix and conduct a principal coordinate analysis. A dissimilarity matrix has people (or samples) on both row and column (it is a n x n matrix) where the values in the table are dissimilarity (or distance) indices. We will use the Jaccard Index.

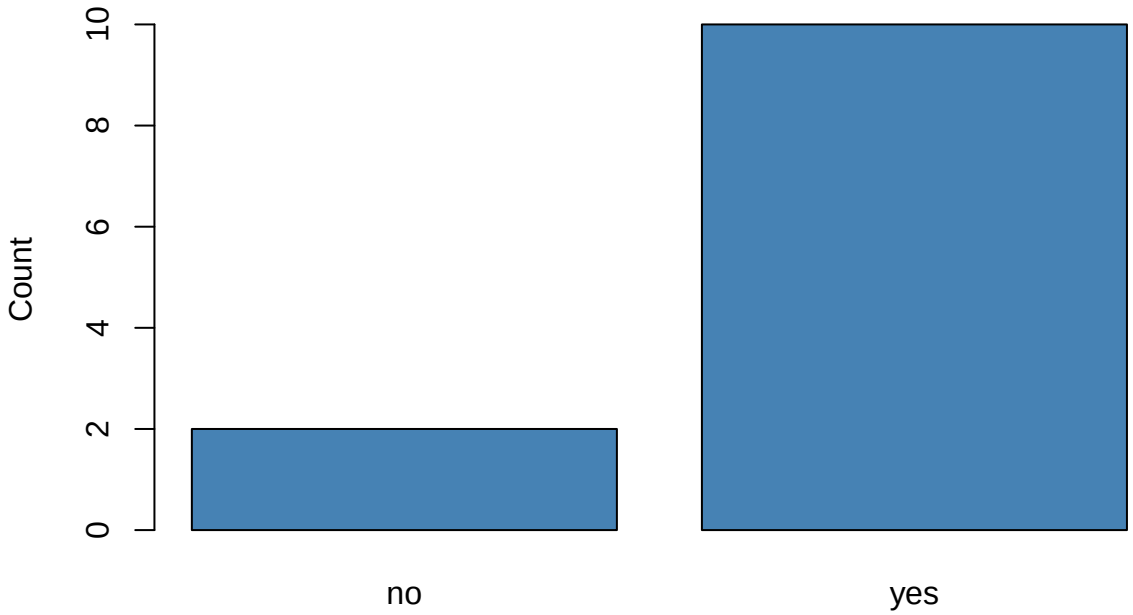
If there's additional data in the .csv file (e.g., metadata), you will have to subset just the data you want to analyze. For example, you may want to make a new dataframe with food <- Class_plant_diet[1:50, 5:40]

Also, convert '1's to "yes" and '0's to "no"

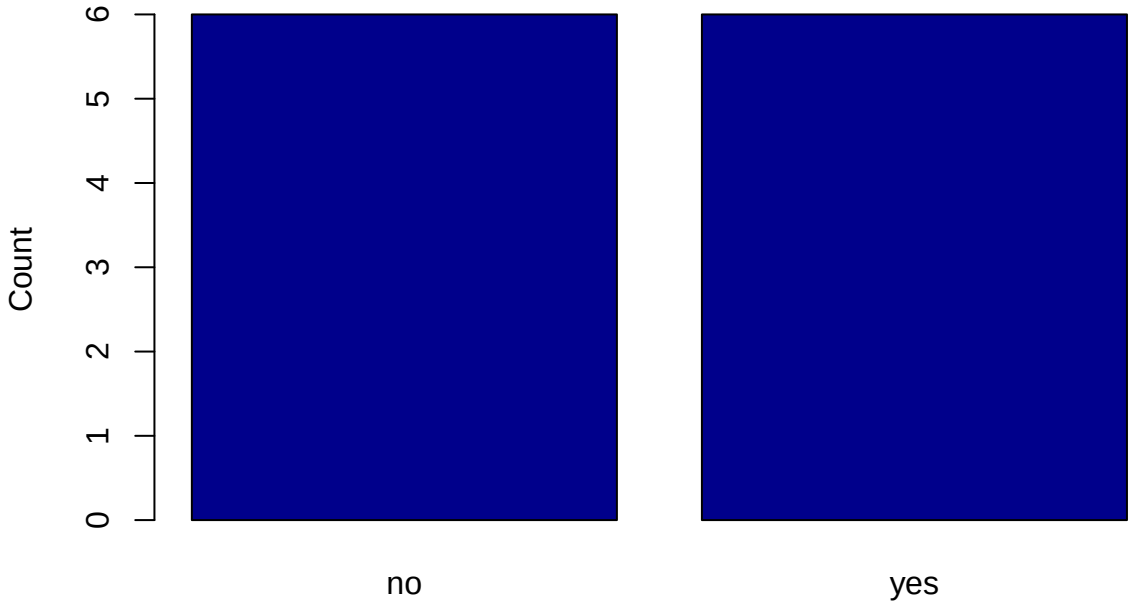
```
Class_data <- as.data.frame(t(Class_plant_diet)) #need to transpose the data frame.

Class_data[Class_data == 1] <- "yes"
Class_data[Class_data == 0] <- "no"

Class_data[] <- lapply(Class_data, factor) #you need to change values into factors rather than characte
summary(Class_data)[,1:4] #See how many people drink coffee, had sugar, etc.
```

```
##  coffee   sugar    wheat     corn
##  no : 2   yes:12   yes:12   yes:12
##  yes:10
```

```
plot(Class_data[,"coffee"], main=colnames(Class_data)["coffee"],
       ylab = "Count", col="steelblue") #check out quickly how many ppl drink coffee
```



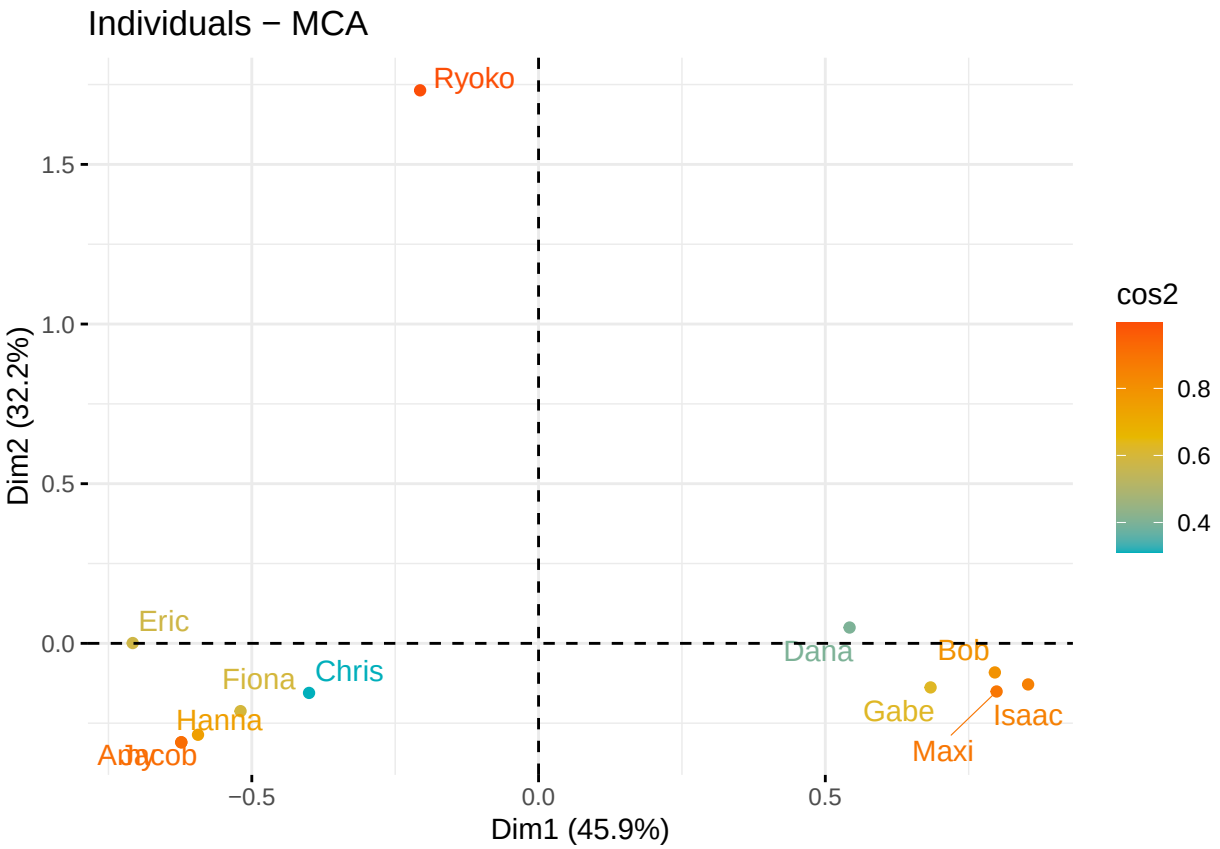

```
plot(Class_data[,"tea"], main=colnames(Class_data)["coffee"],
       ylab = "Count", col="darkblue") #check out quickly how many ppl drink coffee
```



Now let's make a biplot, which is just a plot with 2 axes (x and y). We first have to conduct multiple correspondence analysis, which is simply done by the function MCA. What's really happening is, MCA is a function that calculates the correspondence values of the individuals (all of you) as well as the variables (the plant types). It's like principla component analysis, for those of you who are familiar with PCA, but it works on categorical variables (not continuous ones).

MCA is an eigen analysis working on the correspondence matrix. It decomposes this correspondence data into Eigenvectors and Eigenvalues. One will rotate (eigenvectors) and stretch (eigenvalues) the data points in n-dimensions to represent them as accurately in two dimensional space.

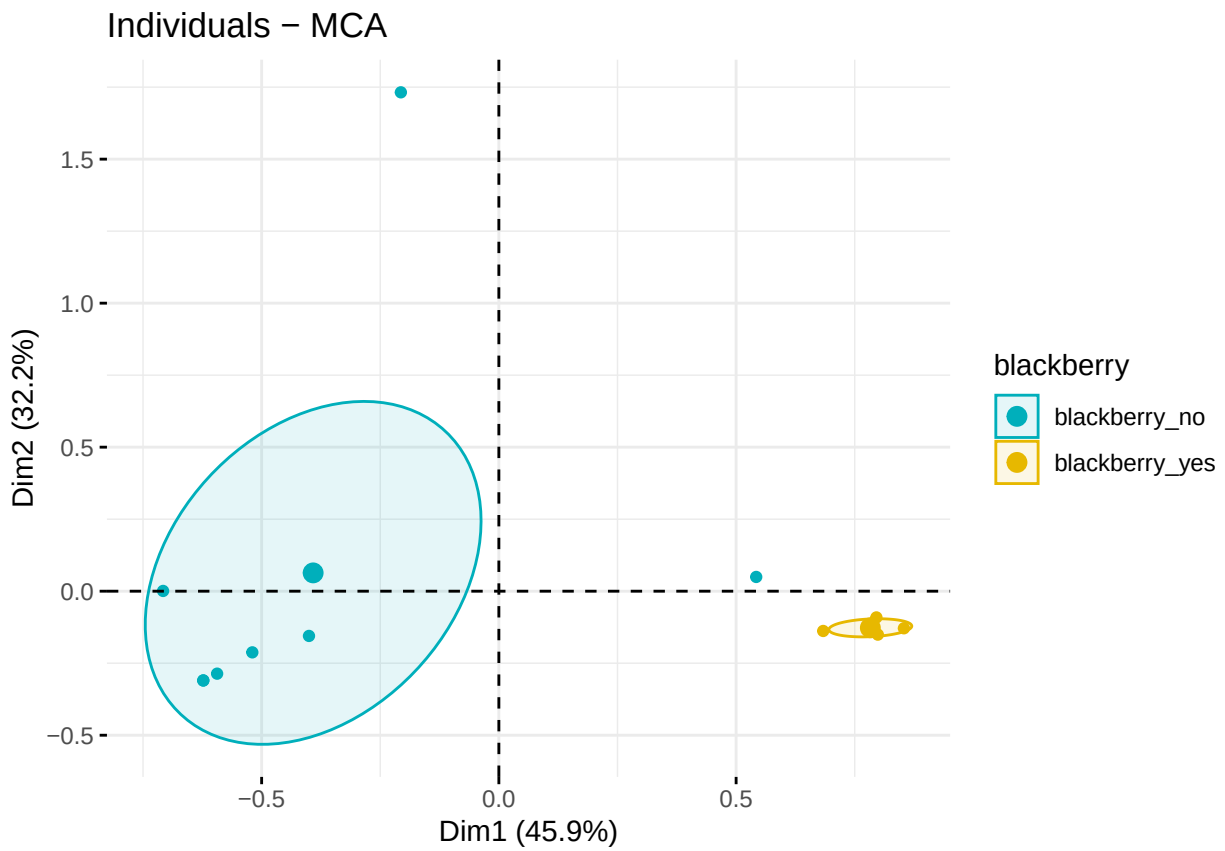
```
#MCA(Class_data, ncp = 2, graph = TRUE)

res.mca <- MCA(Class_data, ncp = 2, graph = FALSE)

fviz_mca_ind(res.mca, col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, # Avoid text overlapping (slow if many points)
             ggtheme = theme_minimal())
```



The quality of the representation is called the squared cosine (cos2), which measures the degree of association between individuals (in this case) and a particular axis.

You can also color based on a particular food. For example, from the variables_representation plot, you can pick out food items that seem to have the largest influence on separating people's diets. For example, maybe people who eat blackberry tend to each other similar foods.

```
fviz_mca_ind(res.mca,
             label = "none", # hide individual labels
             habillage = "blackberry", # color by groups
             palette = c("#00AFBB", "#E7B800"),
             addEllipses = TRUE, ellipse.type = "confidence",
             ggtheme = theme_minimal())
```

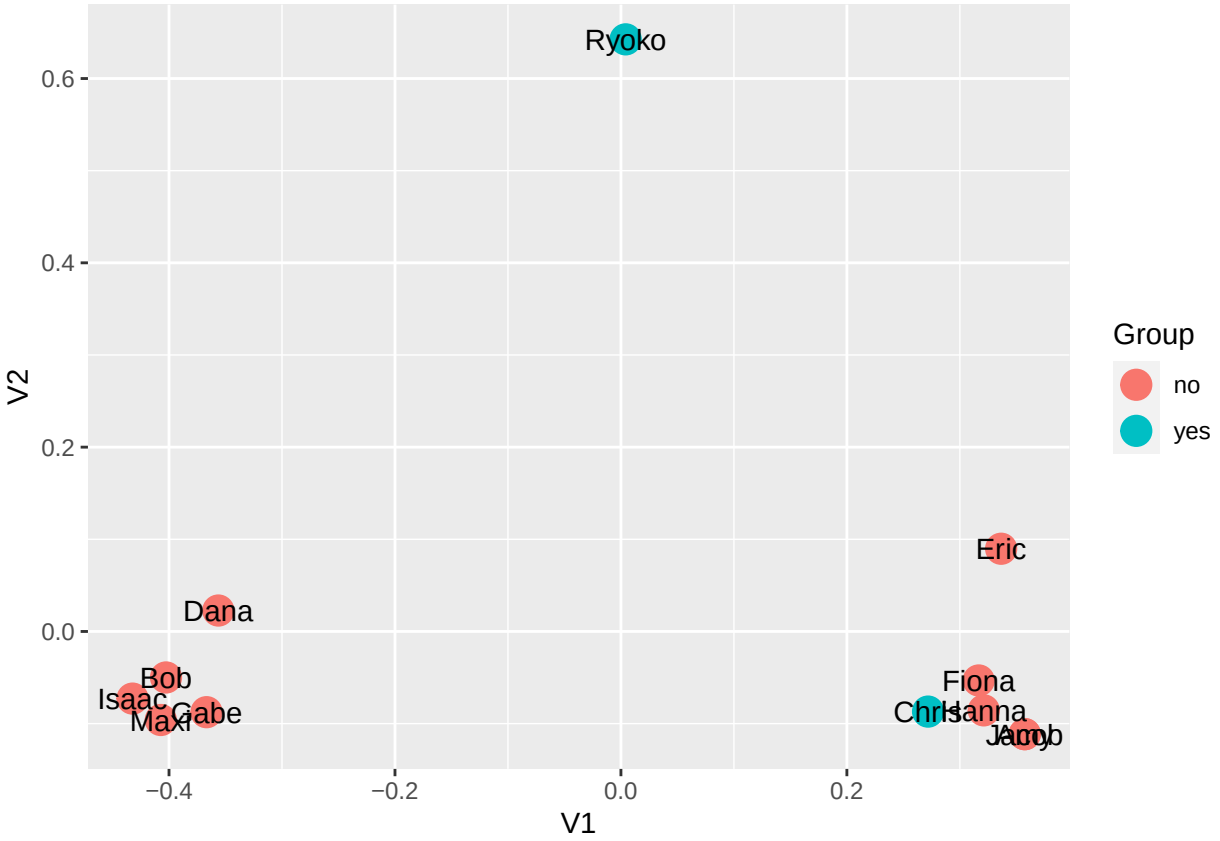


If that was too easy for you can also try modifying the original dataset to include other metadata, such as 'Vegetarian vs. Omnivore' to test if vegetarians all eat more similar plants than omnivores.

Now we're going to visualize how our diets differ by first calculating dissimilarity index. This will help us plot individuals using principal coordinates analysis (PCoA). PCoA is similar to multiple correspondence analysis in that they simply calculate new coordinate system for plotting the points. The point plots are basied on dissimilarity matrix now, not the raw data. PCoA, again, uses eigenvectors that preserve the original distances among the objects in the matrix. We still have to designate a lower dimension than the number of samples.

PCoA works on dissimilarity matrix. PCA can work with raw traits. One might argue that PCoA is necessariyl inferior to PCA because PCoA work with data that are already reduced from its original state (i.e., a dissimilarity matrix).

```
Plant_JC <- vegdist(t(Class_plant_diet), method="jaccard", binary=TRUE, diag=TRUE, upper=TRUE)
PcoA <- as.data.frame(cmdscale(Plant_JC, k =2))
PcoA$Group <- Class_data$coconut #whatever group you want to use

ggplot(PcoA, aes(x = V1, y = V2, colour = Group,
                                 label = row.names(PcoA)))+
 geom_point(size =5) +
        geom_text(col = 'black')
```



As you can see, the plot looks very similar to the biplot we made with multiple correspondence analysis.

All in all, I hope this exercise helped you get to know your fellow classmates and TAs, learned a little bit of R, and brought awareness to the incredible diversity of plants in your daily diets.