CA - Bama Politics Dataset

Sruti Jain

October 3, 2017

## The objectives of this markdown book is running Inferential CA on Bama Politics Dataset

rm(list = ls())  
library(ExPosition)

## Loading required package: prettyGraphs

library(ggplot2)  
library(factoextra)

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library("FactoMineR")  
library(InPosition)  
library(corrplot)  
library("gplots")

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(vcd)

## Loading required package: grid

suppressMessages(library(ExPosition))

## Method: CA

Correspondence analysis (CA) is an extension of principal component analysis suited to explore relationships among qualitative variables (or categorical data). Like principal component analysis, it provides a solution for summarizing and visualizing data set in two-dimension plots.

Here, we describe the simple correspondence analysis, which is used to analyze frequencies formed by two categorical data, a data table known as contengency table. It provides factor scores (coordinates) for both row and column points of contingency table. These coordinates are used to visualize graphically the association between row and column elements in the contingency table.

When analyzing a two-way contingency table, a typical question is whether certain row elements are associated with some elements of column elements. Correspondence analysis is a geometric approach for visualizing the rows and columns of a two-way contingency table as points in a low-dimensional space, such that the positions of the row and column points are consistent with their associations in the table. The aim is to have a global view of the data that is useful for interpretation.

## Bootstrapping & Permutation

Bootstrapping: In statistics, bootstrapping is any test or metric that relies on random sampling with replacement. Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates.This technique allows estimation of the sampling distribution of almost any statistic using random sampling methods. Generally, it falls in the broader class of resampling methods.

Permutation: In simple terms, permutation is all the possible arrangements of a collection of things, where the order is important. In statistics, it is defined as the notion that relates to the act of arranging all the members of a set into some sequence or order, or if the set is already ordered, rearranging (reordering) its elements, a process called permuting.

## Data set: Bama Politics dataset

This statewide survey of 501 adult citizens collected responses to questions on various topics including education, ratings and images of political parties.

*Loading the dataset & running descriptive/exploratory analysis*

load("sor008\_dv.RData")  
  
#isolate the data  
BAMA <- x[,4:46] #questions 1-21 are the data  
  
#isolate the row DESIGN variables  
BAMA\_sup <- x[,c(3,47:56)]

## Perprocessing the dataset & computing a contingency table from them

Let's select a couple interesting variables, and compute a contingency table from them

#q22. On national politics, what is your political party?  
q22\_disjunctive <- makeNominalData(as.matrix(BAMA\_sup$q22))  
colSums(q22\_disjunctive)

## .Independent .Democrat .Republican .dk-na .none   
## 168 172 139 14 5   
## .other   
## 3

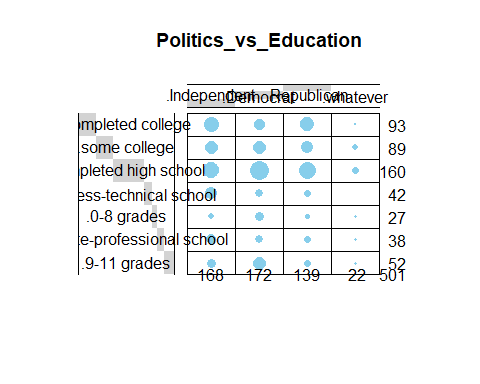
#because several responses contain only a few participants, let's clump them together.  
   
 #let's collapse dk-na, none, and other into 1 group...  
 unsure <- which(q22\_disjunctive[,".dk-na"]==1 | q22\_disjunctive[,".none"]==1 | q22\_disjunctive[,".other"]==1)  
 q22\_clean <- as.matrix(BAMA\_sup$q22)  
 q22\_clean[unsure] <- "whatever"  
 q22\_clean\_disjunctive <- makeNominalData(q22\_clean)  
 colSums(q22\_clean\_disjunctive)

## .Independent .Democrat .Republican .whatever   
## 168 172 139 22

#q27. What's is your level of education?  
q27\_disjunctive <- makeNominalData(as.matrix(BAMA\_sup$q27))  
colSums(q27\_disjunctive)

## .completed college .some college   
## 93 89   
## .completed high school .business-technical school   
## 160 42   
## .0-8 grades .graduate-professional school   
## 27 38   
## .9-11 grades   
## 52

#Finally, compute a contingency table from the above 2 variables using matrix multiplication.   
Politics\_vs\_Education <- t(q27\_disjunctive) %\*% q22\_clean\_disjunctive  
  
# 1. convert the data as a table  
dt <- as.table(as.matrix(Politics\_vs\_Education))  
# 2. Balloon Plot to visualize the contingency table  
balloonplot(t(dt), main ="Politics\_vs\_Education", xlab ="", ylab="", label = FALSE, show.margins = TRUE)



# Descriptive Analysis- Chi-square analysis & Correlation plot  
chisq <- chisq.test(Politics\_vs\_Education)

## Warning in chisq.test(Politics\_vs\_Education): Chi-squared approximation may  
## be incorrect

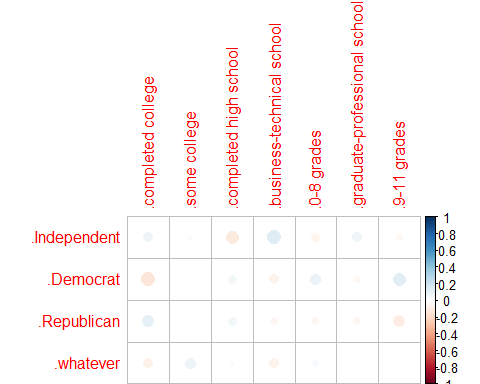
chisq

##   
## Pearson's Chi-squared test  
##   
## data: Politics\_vs\_Education  
## X-squared = 44.047, df = 18, p-value = 0.0005681

CramersV <- sqrt(chisq$statistic / sum(Politics\_vs\_Education))  
CramersV

## X-squared   
## 0.2965095

corrplot(cor(q22\_clean\_disjunctive,q27\_disjunctive))



# Then, we can move on to CA & Inferential CA  
dev.new(TRUE)  
res\_ca <- epCA(Politics\_vs\_Education, graphs = TRUE)  
resinf\_ca <- epCA.inference.battery(Politics\_vs\_Education, graphs = TRUE, test.iters = 1000)

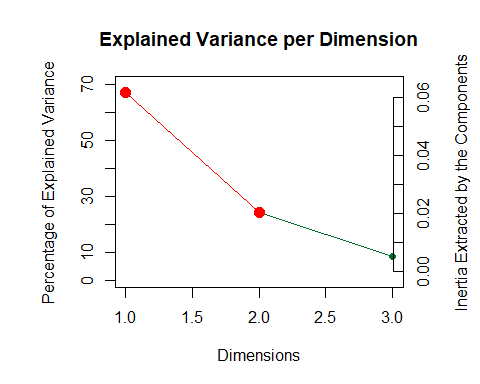
## [1] "It is estimated that your iterations will take 1 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the progress bar."  
## ===========================================================================

### Scree Plot

A scree plot shows the eigenvalues on the y-axis and the number of factors on the x-axis. The number of components is min(nrow(DATA), ncol(DATA)) minus one. Here, we get max of 3 components. The scree plot is used to determine how many of the components should be interpreted.

# --------------------------------------------------------------------  
# Creating a function to plot the scree  
# ev: the eigen values to plot. no default  
# max.ev the max eigen value  
# needed because ExPosition does not return all ev  
# but only the requested one. but return all tau  
# so if max.ev is specified, it is used to recompute  
# all eigenvalues  
# p.ep: the probabilities associated to the ev  
# alpha: threshold for significance. Default = .05  
# col.ns = color for ns ev. Default is Green  
# col.sig = color for significant ev. Default is Violet  
PlotScree <- function(ev,p.ev=NULL,max.vp=NULL, alpha=.05,  
 col.ns = '#006D2C',col.sig='red',  
 title = "Explained Variance per Dimension"  
){  
 # percentage of inertia  
 val.tau = (100\*ev/sum(ev))  
 Top.y = ceiling(max(val.tau)\*.1)\*10  
 # if ev is already a percentage convert it back  
 if (!is.null(max.vp)){ev = ev\*(max.vp/ev[1])}  
 #  
 par(mar=c(5,6,4,4))  
 # plot.window(xlim = c(0, length(val.tau)+5),  
 # ylim = c(0,Top.y),asp = .6)  
 plot(x = seq(1,length(val.tau)),y=val.tau,xlab='Dimensions',  
 ylab = 'Percentage of Explained Variance',  
 main = title,  
 type = 'l', col = col.ns, lwd= 1,  
 xlim = c(1, length(val.tau)),  
 ylim = c(0,Top.y)  
 )  
 points(x = seq(1,length(val.tau)),y=val.tau,  
 pch=16, cex=1, col = col.ns, lwd= 2.5  
 )  
 if (!is.null(resinf\_ca$Inference.Data$components$p.vals)){# plot the significant vp if exist  
 # Plot the significant factors  
 signi.vp = which(resinf\_ca$Inference.Data$components$p.vals < alpha)  
 lines(x = seq(1,length(signi.vp)),y=val.tau[signi.vp],  
 type = 'l', col = col.sig, lwd= 1.5  
 )  
 points(x = seq(1,length(signi.vp)),y=val.tau[signi.vp],  
 pch=16, cex=1.5, col = col.sig, lwd= 3.5)  
 } # end of plot significant vp  
 par(new = TRUE)  
 par(mar=c(5,6,4,4)+.5)  
 le.max.vp = Top.y\*(ev[1]/val.tau[1])  
 plot(ev, ann=FALSE,axes=FALSE,type="n",#line=3,  
 ylim = c(0,le.max.vp))  
 mtext("Inertia Extracted by the Components",side=4,line=3)  
 axis(4)  
} # end of function PlotScree  
  
# --------------------------------------------------------------------

# Calling the above function to plot the scree  
EigenValues <- resinf\_ca$Fixed.Data$ExPosition.Data$eigs  
PlotScree(EigenValues)

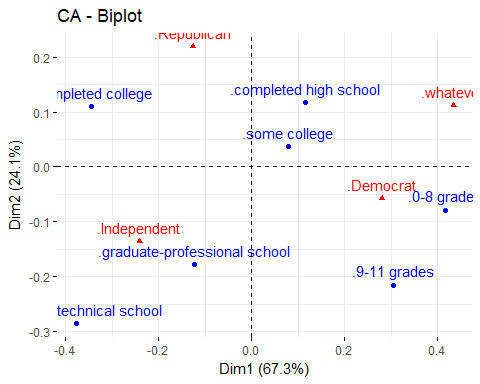


# We will be analyzing first two component in our analysis

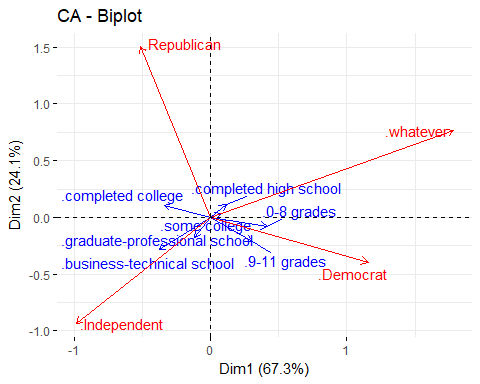
## Including Plots

You can embed plots for plotting the various level in the rows and column in a single plot using biplot function

fviz\_ca\_biplot(res\_ca, repel = FALSE)



fviz\_ca\_biplot(res\_ca,   
 map ="rowprincipal", arrow = c(TRUE, TRUE),  
 repel = TRUE)



# Plotting the correlation cicle plot for the columns  
corrplot <- correlationPlotter(Politics\_vs\_Education, resinf\_ca$Fixed.Data$ExPosition.Data$fi)

## Summary

When we interpret the Biplot and correlation circle plot together, the CA revealed:

1. Component 1: The latent structure of the Bama Politics data as revealed by CA indicated that the first component characterized Republican & Independent versus Democrat. Also it indicates that educated people prefer Republican & Independent over Democrat.
2. Component 2: Mainly distinguishes people supporting the Republican versus Independent & Democrats.