The Survey of autobiographical memory - Inference PCA

Sruti Jain

September 22, 2017

## The objectives of this markdown book is running Inferential PCA analysis of SAM 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(InPosition)  
library(corrplot)  
suppressMessages(library(ExPosition))

## Method: PCA

Principle Component Analysis: A statistical technique used to examine the interrelations among a set of variables in order to identify the underlying structure of those variables. Also called factor analysis Where regression determines a line of best fit to a data set, factor analysis determines several orthogonal lines of best fit to the data set. It is a non-parametric analysis and the answer is unique and independent of any hypothesis about data distribution.

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (or sometimes, principal modes of variation). The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

These two properties can be regarded as weaknesses as well as strengths. 1. Since the technique is non-parametric, no prior knowledge can be incorporated. 2. PCA data reduction often incurs a loss of information.

## 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: SAM dataset

The Survey of Autobiographical memory (SAM) was designed to access trait mnemonics of naturalistic episodic autobiographical, semantic, spatial memory & future thinking. It records the data taken from a sample of 153 people (rows) to measure mnemonic abilities using 31 variables. There are two nominal variables (Active, Sex) & one factor variable (MemoryGroup). The rest of the variables are all integer with two(ID & Age) being a ratio-scale and all others interval variables.

*Loading the dataset & running descriptive/exploratory analysis*

SAMdata <- read.csv("SAMdata.csv")  
head(SAMdata)

## ID memoryGroups Active Sex Age E1 E2 E3 E4 E5 E6 E7 E8 S1 S2 S3 S4 S5  
## 1 164 High TRUE 2 36 5 5 5 5 5 5 5 5 5 5 4 5 5  
## 2 220 High TRUE 1 63 4 4 4 3 4 4 4 3 4 4 4 4 4  
## 3 222 High TRUE 1 43 1 5 1 1 1 1 1 1 2 5 4 4 3  
## 4 444 High TRUE 1 34 5 4 5 3 5 5 4 5 4 5 2 2 4  
## 5 586 High TRUE 2 62 4 4 5 3 4 4 3 4 1 2 2 2 2  
## 6 590 High TRUE 2 43 5 5 2 2 4 4 4 4 4 4 2 2 4  
## S6 P1 P2 P3 P4 P5 P6 F1 F2 F3 F4 F5 F6  
## 1 5 4 4 2 5 3 5 4 3 3 4 4 5  
## 2 3 4 5 3 5 4 4 4 4 4 4 4 4  
## 3 4 4 2 5 1 2 3 1 1 1 1 5 5  
## 4 5 1 2 5 2 4 4 5 5 5 5 5 5  
## 5 4 4 4 4 5 5 5 5 5 5 4 5 5  
## 6 4 4 4 2 5 5 4 4 2 2 2 2 5

str(SAMdata)

## 'data.frame': 153 obs. of 31 variables:  
## $ ID : int 164 220 222 444 586 590 601 634 653 717 ...  
## $ memoryGroups: Factor w/ 3 levels "High","Mystery",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Active : logi TRUE TRUE TRUE TRUE TRUE TRUE ...  
## $ Sex : int 2 1 1 1 2 2 1 2 1 2 ...  
## $ Age : int 36 63 43 34 62 43 38 52 50 59 ...  
## $ E1 : int 5 4 1 5 4 5 2 5 4 4 ...  
## $ E2 : int 5 4 5 4 4 5 2 4 5 5 ...  
## $ E3 : int 5 4 1 5 5 2 3 3 5 5 ...  
## $ E4 : int 5 3 1 3 3 2 2 4 4 1 ...  
## $ E5 : int 5 4 1 5 4 4 2 4 4 5 ...  
## $ E6 : int 5 4 1 5 4 4 1 5 3 4 ...  
## $ E7 : int 5 4 1 4 3 4 1 1 3 5 ...  
## $ E8 : int 5 3 1 5 4 4 1 4 5 4 ...  
## $ S1 : int 5 4 2 4 1 4 4 4 4 4 ...  
## $ S2 : int 5 4 5 5 2 4 4 4 4 4 ...  
## $ S3 : int 4 4 4 2 2 2 4 4 4 2 ...  
## $ S4 : int 5 4 4 2 2 2 4 5 5 2 ...  
## $ S5 : int 5 4 3 4 2 4 2 4 3 4 ...  
## $ S6 : int 5 3 4 5 4 4 4 5 4 5 ...  
## $ P1 : int 4 4 4 1 4 4 2 3 4 5 ...  
## $ P2 : int 4 5 2 2 4 4 2 2 5 5 ...  
## $ P3 : int 2 3 5 5 4 2 4 1 4 5 ...  
## $ P4 : int 5 5 1 2 5 5 4 4 4 1 ...  
## $ P5 : int 3 4 2 4 5 5 2 4 4 5 ...  
## $ P6 : int 5 4 3 4 5 4 4 4 4 4 ...  
## $ F1 : int 4 4 1 5 5 4 4 3 4 5 ...  
## $ F2 : int 3 4 1 5 5 2 4 4 3 5 ...  
## $ F3 : int 3 4 1 5 5 2 2 3 3 5 ...  
## $ F4 : int 4 4 1 5 4 2 2 4 4 5 ...  
## $ F5 : int 4 4 5 5 5 2 1 4 3 5 ...  
## $ F6 : int 5 4 5 5 5 5 5 4 3 5 ...

For more info, try: "summary(SAMdata)"

## Perprocessing the dataset & creating design variables for our analysis

In this analysis, we run PCA on the SAM variables that start from the 6th column. Meanwhile, keep the first column which stores the ID as your rownames.

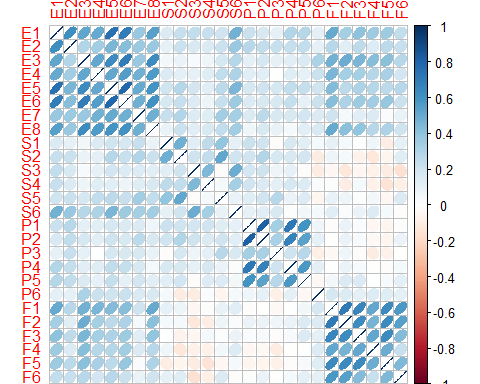
## leaving the mystery group out  
SAMdata <- SAMdata[which(SAMdata$Active == TRUE),]  
ModSamData <- SAMdata[,c(6:ncol(SAMdata))]  
rownames(ModSamData) <- SAMdata[,1]  
## check your data  
head(ModSamData)

## E1 E2 E3 E4 E5 E6 E7 E8 S1 S2 S3 S4 S5 S6 P1 P2 P3 P4 P5 P6 F1 F2 F3  
## 164 5 5 5 5 5 5 5 5 5 5 4 5 5 5 4 4 2 5 3 5 4 3 3  
## 220 4 4 4 3 4 4 4 3 4 4 4 4 4 3 4 5 3 5 4 4 4 4 4  
## 222 1 5 1 1 1 1 1 1 2 5 4 4 3 4 4 2 5 1 2 3 1 1 1  
## 444 5 4 5 3 5 5 4 5 4 5 2 2 4 5 1 2 5 2 4 4 5 5 5  
## 586 4 4 5 3 4 4 3 4 1 2 2 2 2 4 4 4 4 5 5 5 5 5 5  
## 590 5 5 2 2 4 4 4 4 4 4 2 2 4 4 4 4 2 5 5 4 4 2 2  
## F4 F5 F6  
## 164 4 4 5  
## 220 4 4 4  
## 222 1 5 5  
## 444 5 5 5  
## 586 4 5 5  
## 590 2 2 5

### The 2nd columns describes the groups. So, we use it to create a design matrix. When you only have one column, R sees it as a vector. To use \*makeNominalData\*, you need to turn it into a matrix with one column with \*as.matrix\*.   
Des.mat <- makeNominalData(as.matrix(SAMdata[,2]))  
  
## refine the column names  
colnames(Des.mat) <- c("High", "Norm") # give column names   
rownames(Des.mat) <- SAMdata[,1] # set row names as participants' IDs  
## check the design  
head(Des.mat)

## High Norm  
## 164 1 0  
## 220 1 0  
## 222 1 0  
## 444 1 0  
## 586 1 0  
## 590 1 0

## Correlation heatmap  
corrplot(cor(ModSamData), method="ellipse")



Correlation heatmap plot indicates that variables within the questionaire that are testing for a particular memory type are strongly correlated to one another. That is to say that all variable testing episodic memory are highly correlated & this is same for varaibles referring to all memory groups. Also negative correlation was aobserved between variables testing for future & semantic memory types.

## Then, we can move on to Inferential PCA  
res\_pca <- epPCA.inference.battery(ModSamData, center = TRUE, scale = FALSE, graphs = FALSE, DESIGN = Des.mat, make\_design\_nominal = FALSE)

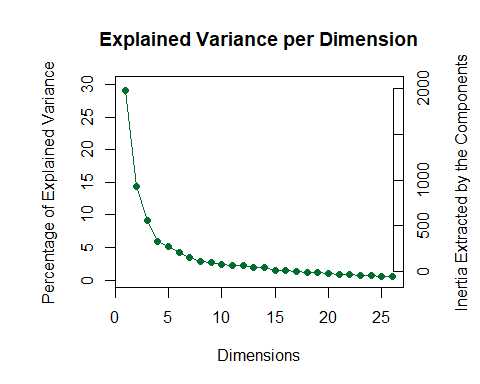
## [1] "It is estimated that your iterations will take 1.48 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the progress bar."  
## ===========================================================================

### 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)). Here, 26 columns gives max of 26 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='#54278F',  
 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(p.ev)){# plot the significant vp if exist  
 # Plot the significant factors  
 signi.vp = which(p.ev < 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 <- res\_pca$Fixed.Data$ExPosition.Data$eigs  
  
PlotScree(EigenValues)



#PlotScree(EigenValues[-length(EigenValues)])

### Factor scores

Factor scores are the coordinates of the observations on the components. The distances between them show which individuals are most similar. Factor scores can be color-coded to help interpret the components.

* prettyPlot helps plot the factor scores. In order to print the result in an Rmd, dev.new needs to be FALSE.

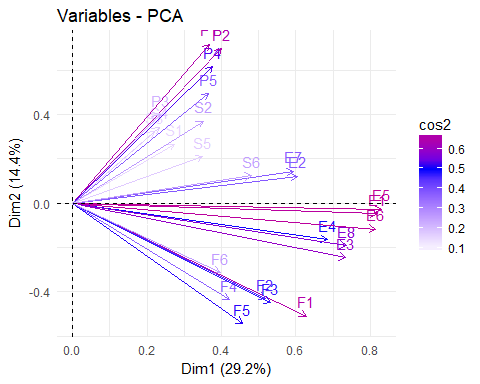
## Average memory score for every observation with respect to the components  
factor\_score <- prettyPlot(data\_matrix = res\_pca$Fixed.Data$ExPosition.Data$fi,   
 dev.new=TRUE,  
 main = "SAMData Row Factor Scores",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = res\_pca$Fixed.Data$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = res\_pca$Fixed.Data$Plotting.Data$fi.col,   
 display\_names = FALSE,   
 xlab = paste0("Component 1 Inertia: ", round(res\_pca$Fixed.Data$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(res\_pca$Fixed.Data$ExPosition.Data$t[2],3), "%")  
 )  
color <- unique(res\_pca$Fixed.Data$Plotting.Data$fi.col)  
legend(x="topright", pch = 15, legend = colnames(Des.mat), col=color)

Component 1 mainly distinguishes people with high versus normal memory groups. Component 2 doesn't essentially show any difference.

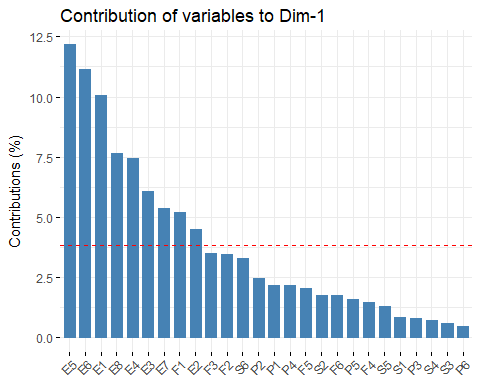
### Loadings

Loadings describe the similarity (angular distance) between the variables. Loadings show how the input variables relate to each other. Loadings also show which variables are important for (which components load on) a certain component.

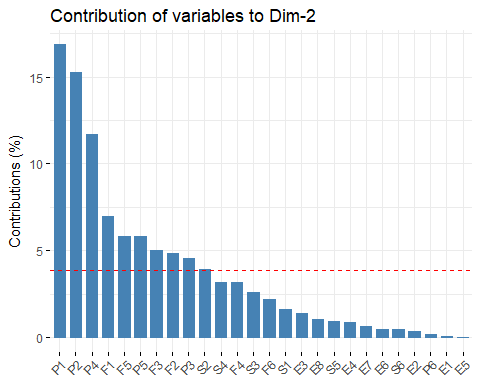
## Plotting the similarity between the variables by analysing the cosine between them.  
fviz\_pca\_var(res\_pca$Fixed.Data, col.var="cos2") +  
scale\_color\_gradient2(low="white", mid="blue",   
 high="red", midpoint=0.5) + theme\_minimal()



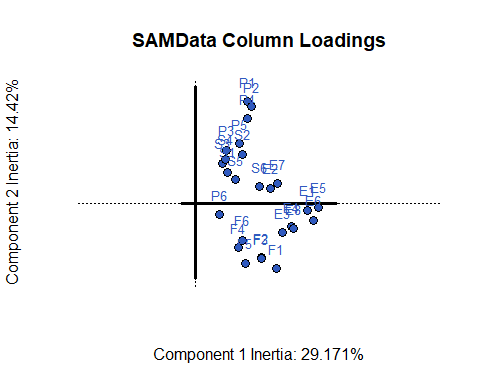
# Percentage Contribution of every variable towards each component  
fviz\_contrib(res\_pca$Fixed.Data, choice = "var", axes = 1)



fviz\_contrib(res\_pca$Fixed.Data, choice = "var", axes = 2)



col\_name <- as.matrix(colnames(ModSamData))  
col\_name <- strsplit(gsub("[^[:alpha:] ]", "", col\_name), " +")  
color\_groups <- c("red", "green", "goldenrod", "yellow")  
col\_name$color\_observ[which(col\_name=="F")] <- col\_name$color\_groups[1]  
col\_name$color\_observ[which(col\_name=="S")] <- col\_name$color\_groups[2]  
col\_name$color\_observ[which(col\_name=="P")] <- col\_name$color\_groups[3]  
col\_name$color\_observ[which(col\_name=="F")] <- col\_name$color\_groups[4]  
  
## Plotting the various variables with respect to the two primary components  
loading\_plot <- prettyPlot(data\_matrix = res\_pca$Fixed.Data$ExPosition.Data$fj,   
 dev.new=FALSE,  
 main = "SAMData Column Loadings",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = res\_pca$Fixed.Data$ExPosition.Data$cj,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = col\_name$color\_observ,   
 display\_names = TRUE,  
 xlab = paste0("Component 1 Inertia: ", round(res\_pca$Fixed.Data$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(res\_pca$Fixed.Data$ExPosition.Data$t[2],3), "%")  
 )



1. Component 1: Seperates normal memory score versus high memory scores and all the varaibles test for the memory scores in positive terms.
2. Component 2 is mainly composed of Spatial memory variable. People who are generally good at Spatial memory task have a poor future thinking ability as they both are orthogonal and negatively correlated.

### Bootstrap ratios

Here we are going to plot the bootstrap ratios for component1 & component2 and also plotting the correlation circle

# This prints the Bootstrap for Component 1  
bootstrap\_C1 <- prettyBars(res\_pca$Inference.Data$fj.boots$tests$boot.ratios[,c(1,2)], axis = 1, threshold.line = TRUE,bg.lims = c(-res\_pca$Inference.Data$fj.boots$tests$critical.value,res\_pca$Inference.Data$fj.boots$tests$critical.value) )

# This prints the Bootstrap for Component 2  
bootstrap\_C2 <- prettyBars(res\_pca$Inference.Data$fj.boots$tests$boot.ratios[,c(1,2)], axis = 2, threshold.line = TRUE,bg.lims = c(-res\_pca$Inference.Data$fj.boots$tests$critical.value,res\_pca$Inference.Data$fj.boots$tests$critical.value) )

# For plotting Correlation Circle   
corrplot <- correlationPlotter(ModSamData, res\_pca$Fixed.Data$ExPosition.Data$fi)

## Summary

When we interpret the factor scores and loadings together, the PCA revealed:

1. Component 1: The latent structure of the SAM data as revealed by PCA indicated that the first component characterized good versus poor memory. In other words,when people reported having high or low abilities for one category of memory,they tended to do the same for other categories.
2. People who are good at Spatial memory & Future thinking are indicated to have an overall good memory in other categories as well.
3. Component 2 is mainly composed of Spatial memory variable. People who are generally good at Spatial memory task have a poor future thinking ability as they both are orthogonal and negatively correlated.