CookbookTemplate\_v3

Michael A. Kriegsman

September 18, 2017

# Method: PCA

Principal component analysis (PCA) is...

## Data set: state.x77

State.x77 is...

head(state.x77)

## Population Income Illiteracy Life Exp Murder HS Grad Frost  
## Alabama 3615 3624 2.1 69.05 15.1 41.3 20  
## Alaska 365 6315 1.5 69.31 11.3 66.7 152  
## Arizona 2212 4530 1.8 70.55 7.8 58.1 15  
## Arkansas 2110 3378 1.9 70.66 10.1 39.9 65  
## California 21198 5114 1.1 71.71 10.3 62.6 20  
## Colorado 2541 4884 0.7 72.06 6.8 63.9 166  
## Area  
## Alabama 50708  
## Alaska 566432  
## Arizona 113417  
## Arkansas 51945  
## California 156361  
## Colorado 103766

Create DESIGN variables.

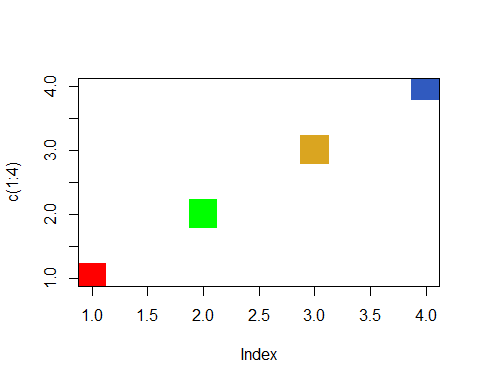
Below, I show you how to work with DESIGN variables and colors. Please Delete our comments, and write your own comments in here.

#Create an empty list, called DESIGN  
DESIGN <- list()  
  
#Let's set the group names (labels), and establish which row belongs to which group (as a DESIGN vector, and then as a DESIGN matrix)  
#For now, let's work on a DESIGN on the rows. The DESIGN vairable is called region.   
#The group names are labeled:  
DESIGN$rows$Region$labels <- unique(state.region) #to see what's here, highlight and run "state.region". Then also highlight and run "unique(state.region)".   
  
#This variable is pretty easy... the DESIGN vector (a vector that describes which row belongs to which group) is given by state.region  
DESIGN$rows$Region$vec <- state.region  
  
#Convert the vector to a matrix  
DESIGN$rows$Region$mat <- makeNominalData(as.matrix(DESIGN$rows$Region$vec))  
#notice that makeNominalData wants to have a matrix, so we use as.matrix(DESIGN\_vector)

We also want to work with the colors of the groups.

#One way to pick colors is automatically, like how epPCA does it...  
automatic\_colors <- createColorVectorsByDesign(DESIGN$rows$Region$mat)  
#notice that automatic\_colors$oc gives the colors for each observation, and automatic\_colors$gc gives the colors for each group.

###Make your own colors, and write your own comments ###  
  
#But, instead of being automatic, we often want to select our colors by hand.   
#Let's just pick 4 colors (you should choose your own)  
DESIGN$rows$Region$color\_groups <- c("red", "green", "goldenrod", "#305ABF")  
  
#and let's just look at the colors we've picked, to be sure they're pretty enough  
plot(c(1:4), pch=15, cex=4, col=DESIGN$rows$Region$color\_groups)



#Actually, these are quite ugly, but I'll let you pick your own pretty colors.   
#But don't show this plot in your cookbook, this is just for you to select colors.   
  
#Now, we need to create a vector of the group colors, to show which row belongs to which group.  
#One group at a time, let's trade the group names for the color names...   
### Follow me slowly, and run each line to see what's inside  
  
#First, copy the group names  
DESIGN$rows$Region$color\_observ <- as.matrix(DESIGN$rows$Region$vec) #we want to change the factor to a matrix, because they work differently  
  
#Then, for each group, replace the group name with the group's color  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="South")] <- DESIGN$rows$Region$color\_groups[1]  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="West")] <- DESIGN$rows$Region$color\_groups[2]  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="Northeast")] <- DESIGN$rows$Region$color\_groups[3]  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="North Central")] <- DESIGN$rows$Region$color\_groups[4]  
#Be sure you get the group names right, and keep track of which group is which, so you can create your legend!

## Results

Center? Scale?

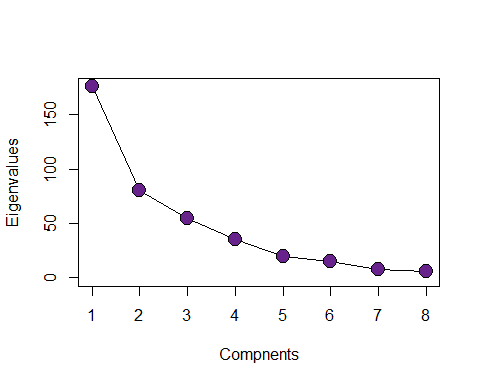
res\_pca <- epPCA(state.x77, center = TRUE, scale = TRUE, graphs = FALSE)

## Plots

PCA gives... THESE plots...

### Scree Plot

name\_the\_scree <- plot(res\_pca$ExPosition.Data$eigs,  
 ylab = "Eigenvalues",  
 xlab = "Compnents",  
 type = "l",  
 main = "",  
 ylim = c(-1, max(res\_pca$ExPosition.Data$eigs)))  
points(res\_pca$ExPosition.Data$eigs, cex = 2, pch = 19, col = "darkorchid4")  
points(res\_pca$ExPosition.Data$eigs, cex = 2, pch = 21, col = "black")

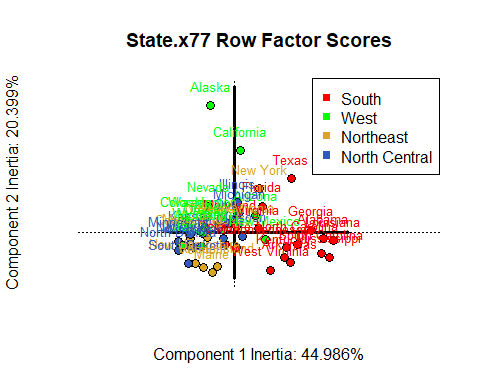


### Factor scores

Factor scores are the coordinates of the 50 states on the components. The distances between them show which states are most similar. Factor scores (states) can be color-coded to help interpret the components.

You may or may not want to leave display\_names=TRUE...

name\_the\_plot <- prettyPlot(data\_matrix = res\_pca$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "State.x77 Row Factor Scores",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = res\_pca$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN$rows$Region$color\_observ,   
 display\_names = TRUE,   
 xlab = paste0("Component 1 Inertia: ", round(res\_pca$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(res\_pca$ExPosition.Data$t[2],3), "%")  
 )  
  
legend(x="topright", pch = 15, legend = DESIGN$rows$Region$labels, col=DESIGN$rows$Region$color\_groups)

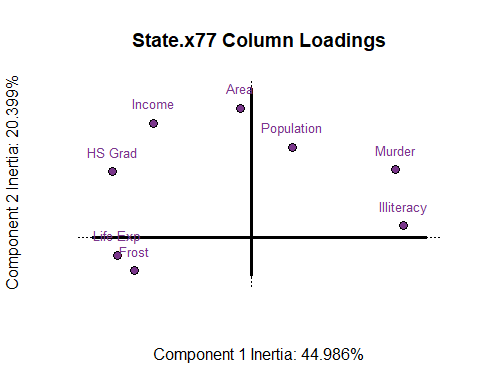


* Component 1:
* Component 2:
* Component 3? Include whatever is relevant.

### Loadings

Loadings are...

name\_another\_plot <- prettyPlot(data\_matrix = res\_pca$ExPosition.Data$fj,   
 dev.new=FALSE,  
 main = "State.x77 Column Loadings",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = res\_pca$ExPosition.Data$cj,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = res\_pca$Plotting.Data$fj.col,   
 display\_names = TRUE,   
 xlab = paste0("Component 1 Inertia: ", round(res\_pca$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(res\_pca$ExPosition.Data$t[2],3), "%")  
 )



* Component 1:
* Component 2:
* Component 3? Include whatever is relevant.

### Include other relevant plots...

## Conclusions