ASSIGNMENT 2 (Spring 2020).   PCA in R

This assignment uses data from two sections of this course, using the forms we filled out. The form asks students to indicate previous experience with various statistics software packages, and also math background. Even though some of the variable are binary (1=yes, 0=no) and some are numeric (asking students to indicate the number of each type of math course taken), we will simply correlate them with minimal processing.

STEP 1. Read in the data

The data is contained in file “class skills data_2016-2020.csv”. You can read these data into R using the same “read.csv” function I used in the class example. In fact, that code used to run the basic analyses I showed in class can just be edited for this assignment.

However, there is one complication with these data: there are many data fields that are blank in the .CSV version of the data file, and these cells become missing values, coded “NA” in the R data frame you read the data into.

We don’t want to ignore or delete these data observations – in this survey a missing value means the same thing as a “0” – that one doesn’t have that stats skill. So we need to write R code to plug “0” for each missing value. The R code below will read in the data and do this.

```
-------------------------------------------------------------------------------------------
# R code to read in skill set multivariate data, compute a correlation matrix Rx
skills2019 <- read.csv("C:/Users/corter/Desktop/hudm5124/class skills data_2016 18 19.csv")
skills2019
str(skills2019)

# now set "sk" = columns 7 to M of the data matrix, & plug 0 for each missing value
N<-nrow(skills2019)   # N is = the number of subjects (observations)
N
M<-ncol(skills2019)  # M is the number of variables
M
sk<-skills2019[,7:M]

# plus 0 for missing values
M<-ncol(sk)  # M is the number of variables
for (j in 1:M)  # for each colum,
{
  cx<-sk[,j]
  z<-ifelse(is.na(cx),c(rep(0,N)),cx)
  sk[,j]<-z
}
sk
# now all values "NA" have been converted to 0
Rx<-cor(sk)

-------------------------------------------------------------------------------------------------------
```

At this point you will detect another problem: one of the “variables” in the data set is actually a constant (because no one had that obscure skill, SYSTAT). So you will have to delete that column before you compute the correlation matrix. One way to do that is to select the first four columns of the data matrix, and select columns 6-M (which omits column 5), then paste those two chunks back together with function cbind:

```
skc<-cbind(sk[,1:4],sk[,6:24])
Rxc<-cor(skc)        # correlation matrix Rxc has only complete data with non-trivial variables
```

STEP 2. Run a Principal Components Analysis (PCA).

After the above steps, you should have a 22 by 22 correlation matrix (or a 40 x 22 raw data matrix). Using and modifying the code I presented in class, do the following analyses.

A. Use princomp to run a PCA of the data. Generate the scree plot, and use it to decide on the number of dimensions, k. Discuss your decision in a sentence or two.

B. print out the columns of P for the first k principal components, and plot pairs of columns (component loadings) against each other. If you have a 2-component solution, plot P1 vs P2. If you have a 3-component solution, to save some space and effort you could just plot P1 vs P2, then P2 vs P3. For a 4-component solution, you could plot just P1 vs P2, then P3 vs P4, etc.

Interpret each component by picking out the large positive loadings (the elements of the P vector) and any large negative ones (if they exist) and discussing those two sets of variables.

C. use the eigenvalue/eigenvector routine in R to solve for the principal components, as demonstrated in class. Do they match the ones obtained by princomp?

[We will talk more about interpreting PCA solutions, and about rotations of the solution, next week.]