PCA Reducing Dimensionality - SoftDrink

Joseph Shu

2/28/2020

Sixty students rated 10 brands of soft drinks (Coke, Diet Pepsi, Dr. Pepper, Mt. Dew, Pepsi, Royal Crown, 7Up, Sprite, Diet 7Up, Tab) on four attributes (calories, sweetness, thirst–quenching, and popularity with others) at two different times during the semester (September and November). The variables in the data set are defined as follows:

- X1 Calories (September)
- X2 Calories (November)
- X3 Sweetness (September)
- X4 Sweetness (November)
- X5 Thirst–quenching (September)
- X6 Thirst–quenching (November)
- X7 Popularity (September)
- X8 Popularity (November)

Analyze the data using factor analysis. How many factors are there? How would you interpret them? Which of the four attributes has the highest reliability? How do you tell?

Load packages

```
library(psych)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(data.table)
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------ tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts --------------------------------------------------------- tidyverse_conflicts() --
## x ggplot2::%+%()     masks psych::%+%()
## x ggplot2::alpha()   masks psych::alpha()
```

```
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

Read in files

```
softdrink <- read.csv("soft_drinks.csv")
```

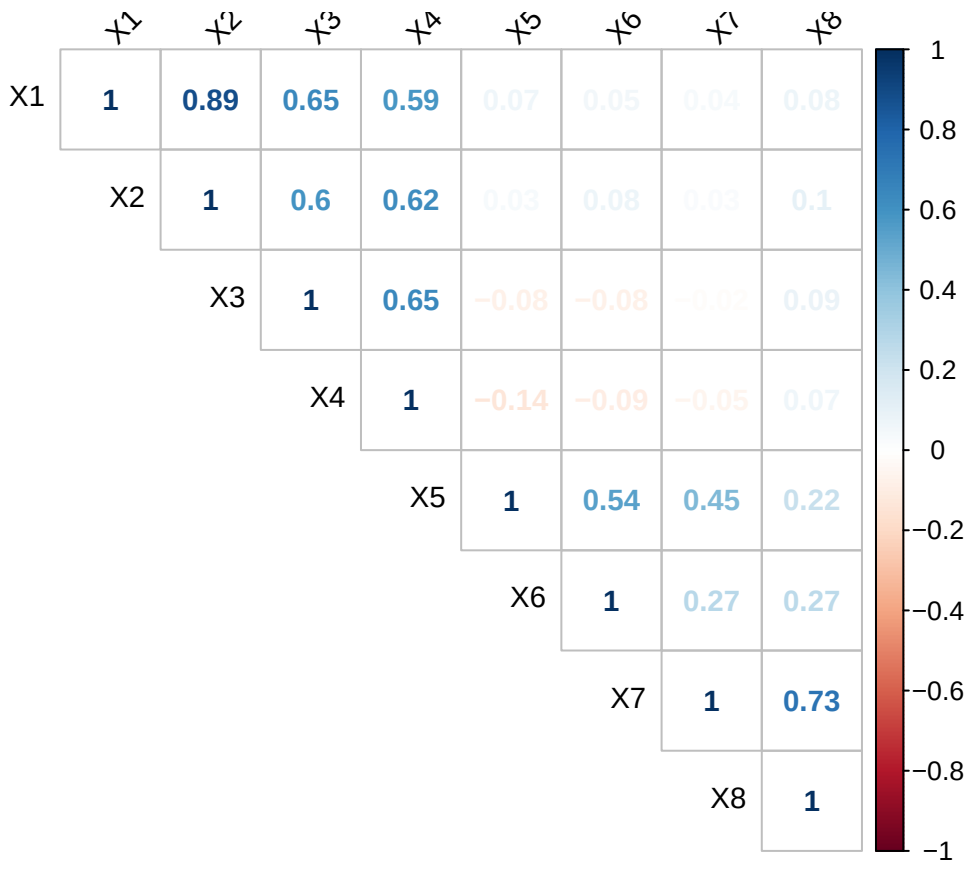
```
rownames(softdrink) <- softdrink[,1]
softdrink <- softdrink[,-1]
softdrink[upper.tri(softdrink)] = t(softdrink)[upper.tri(softdrink)]
softdrink
```

```
##       X1    X2     X3     X4     X5     X6     X7    X8
## X1 1.000 0.886  0.649  0.588  0.067  0.054  0.037 0.075
## X2 0.886 1.000  0.597  0.621  0.034  0.076  0.029 0.102
## X3 0.649 0.597  1.000  0.649 -0.080 -0.075 -0.018 0.089
## X4 0.588 0.621  0.649  1.000 -0.136 -0.092 -0.054 0.069
## X5 0.067 0.034 -0.080 -0.136  1.000  0.542  0.446 0.225
## X6 0.054 0.076 -0.075 -0.092  0.542  1.000  0.274 0.267
## X7 0.037 0.029 -0.018 -0.054  0.446  0.274  1.000 0.730
## X8 0.075 0.102  0.089  0.069  0.225  0.267  0.730 1.000
```

Fill in the NAs and make it symmetrical.

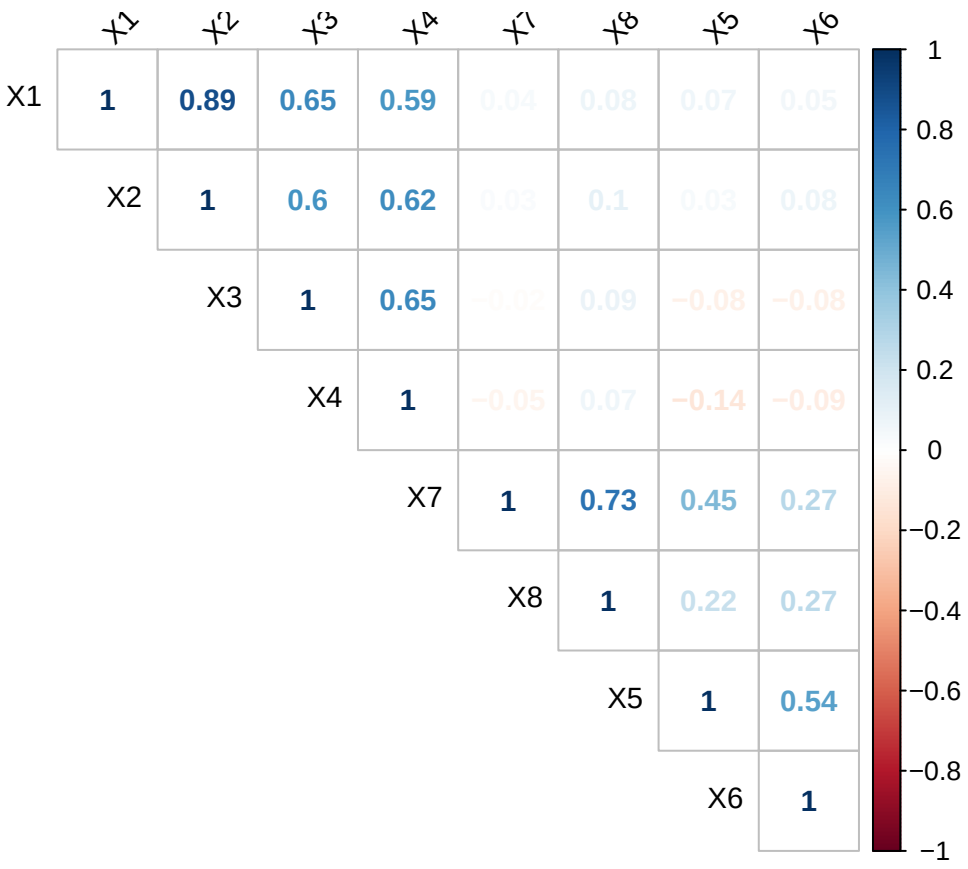
Correlation Matrix

```
library(corrplot)
sd.matrix <- as.matrix(softdrink)
corrplot(sd.matrix,
         method = "number", # correlation coefficients for numeric variables
         type = "upper",    # to display only the lower part of the symmetric correlation matrix
         tl.cex = 0.9,        # font size of the variable labels
         tl.col = "black",    # color of the variable labels
         tl.srt = 45,         # rotation angle for the variable labels
         number.cex = 0.9      # font size of the coefficients
)
```



We can see that X1 is highly correlated with X2, which makes sense, sine they are both calories (one in September, one in November). At the same time, X1 is also highly correlated with X3 and X4. From this observation, we can conclude that Calories are essentially related to Sweetness, which is not surprising. Furthermore, X5 is highly correlated with X6, and X7 is to X8. To elaborate, X5 and X6 are the same variable(Thirst–quenching), so are X7 and X8(Popularity), the only difference is that they were rated in different months.

```
corrplot(sd.matrix,
         method = "number",
         type = "upper",
         order = "hclust", # reorder by the size of the correlation coefficients
         tl.cex = 0.9, # font size of the variable labels
         tl.col = "black", # color of the variable labels
         tl.srt = 45, # rotation angle for the variable labels
         number.cex = 0.9 # font size of the coefficients
)
```



When we change the order method to hclust, the pattern is even more clear. The findings here raises the question of how many factors we should use exactly. Since it's very clear that Calories and Sweetness will belong to the same factor, however, we cannot directly conclude that if we should segment the rest to a factor or take a third factor since there shows no strongly correlation between Thirst-quenching and Popularity.

Eigenvalues and Explained variance

```
EV = eigen(sd.matrix)$values
EV
```

```
## [1] 3.0172030 2.2701428 1.0605542 0.5179262 0.4721255 0.3529867 0.2052413
## [8] 0.1038202
```

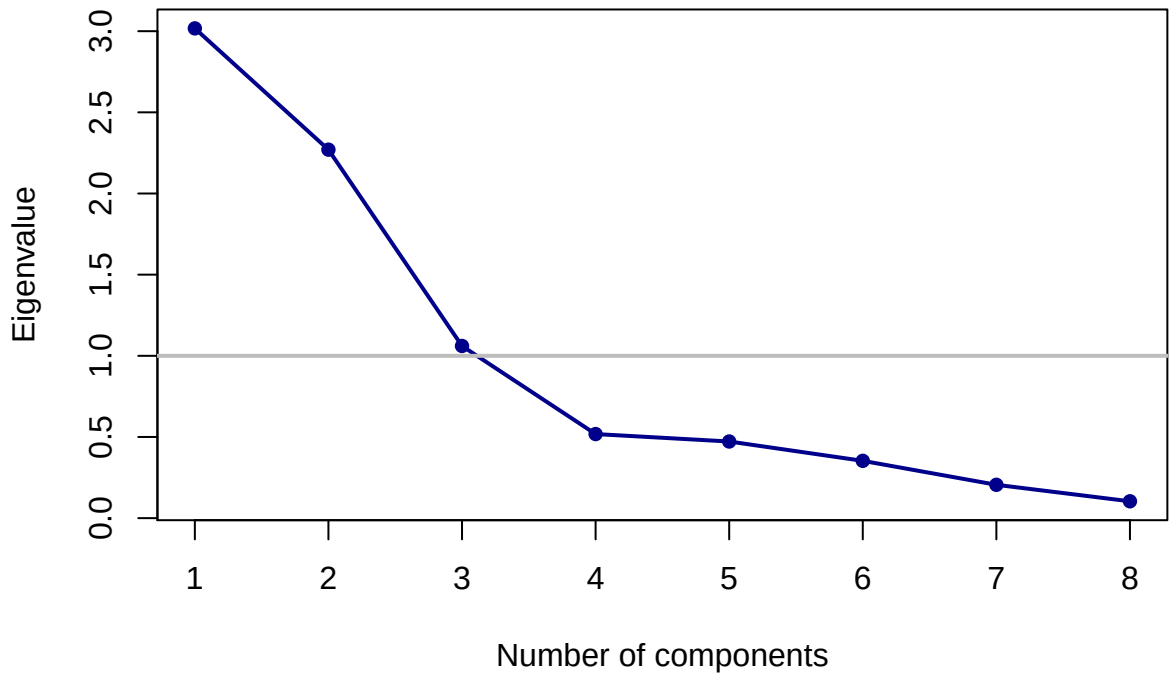
```
EV/length(EV)
```

```
## [1] 0.37715038 0.28376785 0.13256927 0.06474078 0.05901568 0.04412334 0.02565517
## [8] 0.01297753
```

Considering Eigenvalues, we could retain up to three factors since from the fourth on, EV is lower than 1.0. In the second output, we see the variance explained by each factor. This can be seen more clearly in the scree plot.

Scree plot

```
plot(eigen(sd.matrix)$values,
     type = "o", # type of points
     col = "darkblue",
     pch = 16, # symbol type (here, filled circle)
     cex = 1, # size of plot symbols
     xlab = "Number of components",
     ylab = "Eigenvalue",
     lwd = 2) # line width
abline(h = 1, lwd = 2, col = "grey") # horizontal line at 1
```



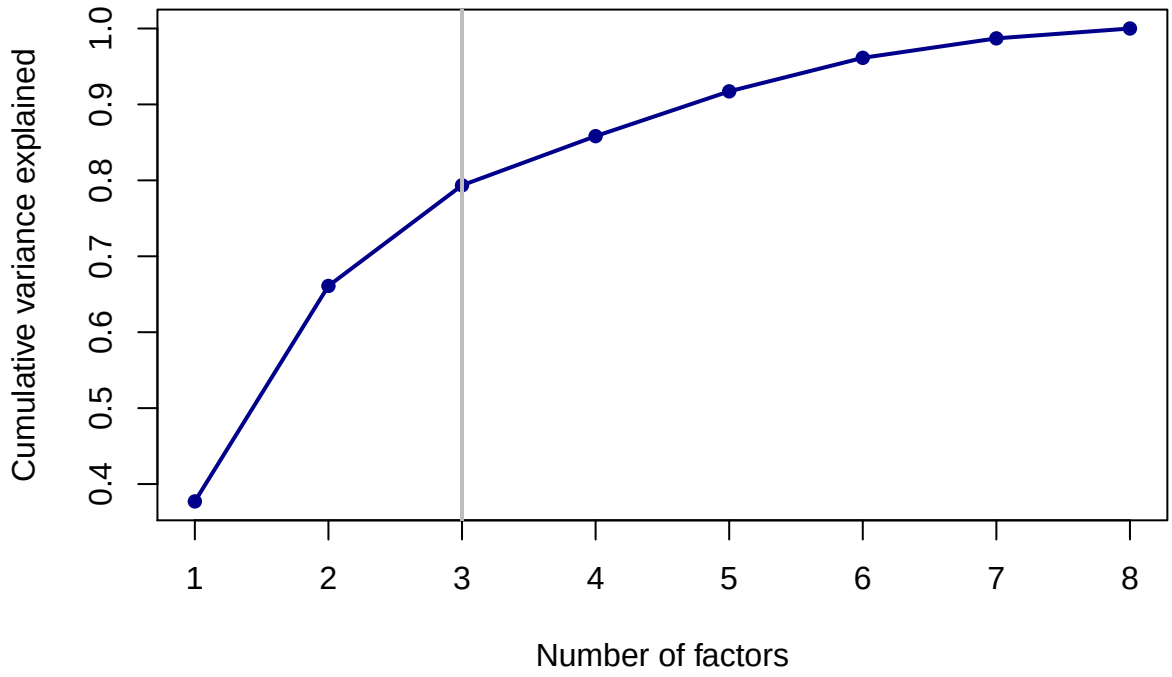
It could may be justified to extract three factors, since the elbow seems to appear after the third eigenvalue. I thus decided to proceed with three factors.

```
cumsum(EV/length(EV))
```

```
## [1] 0.3771504 0.6609182 0.7934875 0.8582283 0.9172440 0.9613673 0.9870225
## [8] 1.0000000
```

The cumulative percentage of variance explained can be depicted here, that if we take two factors , they account for 66% of variance; if we take three, they account for almost 80% of variance. We can visualize it with a better interpretation.

```r
# Shares for the cumulative variance explained
plot(cumsum(EV/length(EV)),
     type = "o", # type of plot: "o" for points and lines 'overplotted'
     col = "darkblue",
     pch = 16, # plot symbol: 16 = filled circle
     cex = 1, # size of plot symbols
     xlab = "Number of factors", # a title for the x axis
     ylab = "Cumulative variance explained", # a title for the y axis
     lwd = 2) # line width
abline(v = 3, lwd = 2, col = "grey") # draw a vertical line at v = 3
```



Execute in EFA

Since I have decided earlier to proceed with three factors, I will first analyze the unrotated solution to see if it could already be interpreted easily.

```r
EFA1 <- fa(r = softdrink,
           nfactors = 3,
           fm = "pa",
           rotate = "none")
```

```
## maximum iteration exceeded
```

```r
print(EFA1,
      digits = 3, # to round numbers to the third digit
      cut = 0.35, # to show only values > 0.35
      sort = TRUE # to sort rows by loading size
)
```

```
## Factor Analysis using method =  pa
## Call: fa(r = softdrink, nfactors = 3, rotate = "none", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##    item   PA1    PA2    PA3    h2    u2  com
## X1    1 0.909               0.849 0.151 1.05
## X2    2 0.894               0.813 0.187 1.03
## X3    3 0.740               0.567 0.433 1.07
## X4    4 0.723               0.560 0.440 1.15
## X7    7        0.778        0.656 0.344 1.17
## X8    8        0.775 -0.515 0.899 0.101 1.87
## X5    5        0.687  0.548 0.772 0.228 1.91
## X6    6        0.507        0.365 0.635 1.72
##
##                        PA1   PA2   PA3
## SS loadings           2.737 1.971 0.775
## Proportion Var        0.342 0.246 0.097
## Cumulative Var        0.342 0.588 0.685
## Proportion Explained  0.499 0.359 0.141
## Cumulative Proportion 0.499 0.859 1.000
##
## Mean item complexity =  1.4
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  28  and the objective function was  4.315
## The degrees of freedom for the model are 7  and the objective function was  0.309
##
## The root mean square of the residuals (RMSR) is  0.031
## The df corrected root mean square of the residuals is  0.062
##
## Fit based upon off diagonal values = 0.993
## Measures of factor score adequacy
##                                                  PA1   PA2   PA3
## Correlation of (regression) scores with factors   0.960 0.952 0.888
## Multiple R square of scores with factors          0.921 0.907 0.789
## Minimum correlation of possible factor scores     0.843 0.813 0.578
```

the cumulative proportion of variation in the first output shows that the first three factors together explain 68.5% of the variation. In the second output, however, we cannot distinctly the loading values between PA2 and PA3. We can see if we could improve it by rotating them.

```r
sort(EFA1$communality)
```

```
##        X6        X4        X3        X7        X5        X2        X1        X8
## 0.3654182 0.5604211 0.5674708 0.6564170 0.7717623 0.8129065 0.8492137 0.8989743
```

The communalities (h2) are between 0.24 and 0.81. We can see that X5, X6 have the lowest communality value, which implies that there is not so much of the variation that is explained by three factors together.

```r
L <- unclass(EFA1$loadings)
round(L, 3)
```

```
##      PA1    PA2    PA3
## X1 0.909 -0.017  0.149
## X2 0.894 -0.015  0.116
## X3 0.740 -0.114 -0.085
## X4 0.723 -0.157 -0.113
## X5 0.013  0.687  0.548
## X6 0.030  0.507  0.328
## X7 0.075  0.778 -0.215
## X8 0.185  0.775 -0.515
```

We can store the factor loadings to L for future use.

Rotated factor solution

```r
EFA2 <- fa(r = softdrink,
           nfactors = 3,
           fm = "pa",
           rotate = "varimax")
```

```
## maximum iteration exceeded
```

```r
print(EFA2,
      digits = 3, # to round numbers to the third digit
      cut = 0.35, # to show only values > 0.35
      sort = TRUE # to sort rows by loading size
)
```

```
## Factor Analysis using method =  pa
## Call: fa(r = softdrink, nfactors = 3, rotate = "varimax", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##    item    PA1    PA2    PA3    h2    u2  com
## X1    1  0.912              0.849 0.151 1.04
## X2    2  0.895              0.813 0.187 1.03
## X3    3  0.743              0.567 0.433 1.06
## X4    4  0.730              0.560 0.440 1.11
## X8    8         0.937        0.899 0.101 1.05
## X7    7         0.733        0.656 0.344 1.42
## X5    5                0.863 0.772 0.228 1.07
## X6    6                0.580 0.365 0.635 1.17
##
##                        PA1   PA2   PA3
## SS loadings          2.725 1.475 1.282
## Proportion Var       0.341 0.184 0.160
## Cumulative Var       0.341 0.525 0.685
## Proportion Explained 0.497 0.269 0.234
## Cumulative Proportion 0.497 0.766 1.000
##
```

```
## Mean item complexity =  1.1
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  28  and the objective function was  4.315
## The degrees of freedom for the model are 7  and the objective function was  0.309
##
## The root mean square of the residuals (RMSR) is  0.031
## The df corrected root mean square of the residuals is  0.062
##
## Fit based upon off diagonal values = 0.993
## Measures of factor score adequacy
##                                                  PA1   PA2   PA3
## Correlation of (regression) scores with factors   0.959 0.949 0.893
## Multiple R square of scores with factors          0.920 0.901 0.797
## Minimum correlation of possible factor scores     0.840 0.801 0.593
```

After rotation, it is clear that X1 to X4 can be explained mostly by PA1, with each attribute loading from 0.73 to 0.91 ; X7 and X8 by PA2, each with high loading 0.73 and 0.94 ; X5 and X6 by PA3, with loading 0.58 and 0.86. Moreover, we can tell from u2("uniqness") of the attributes, which stands for the variance of error term, that on average, X1 and X2 (Calories) have the lowest values. Thus, we can conclude that "Calories" is the most reliable attribute since the errors generated from factor analysis is the lowest. Similarly, we can also look at the communality(h2), which tells us the proportion of the variance that is explained. (h2 plus u2 will equal to 1). Notably, X8 has actually the lowest u2 and highest h2. Nonetheless, the corresponding attribue X7 has a much higher u2 and lower h2 than X1 and X2.