Principal Component Analysis (demo)

Avet Mnatsakanyan

2020-08-24

Principal Component Analysis(PCA) is dimensionality-reduction method used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set

In this project I am going to present PCA analysis using collected data about yogurt brands and the rating of their features according to the surveyed consumers. Note I randomly assigned some of US-popular brand names to the actual data as this project has only a demonstration purpose. The real analysis(including collected data) is not about US market.

Start with importing dataset and installing necessary packages.

Use basic commands to observe imported dataset:

```
head(brand.ratings)
```

```
##   price packaging flavor Freshness Quality trendy Healthiness  brand
## 1    10         6      7         1       2      3           5 Danone
## 2     3         5     10         7       4      6           5 Danone
## 3     1         1      6         7       5      5           2 Danone
## 4     8         7      8         5       2      9           5 Danone
## 5     8        10     10        10       5      6           7 Danone
## 6     7         9      6         7       6      6           7 Danone
```

```
tail(brand.ratings)
```

```
##     price packaging flavor Freshness Quality trendy Healthiness brand
## 395     6         6      1        10       8      1           8  Fage
## 396    10         2      3         5       6      2           5  Fage
## 397     7         1      3         9       9      4           9  Fage
## 398    10         1      2         5       5      1           5  Fage
## 399     3         4      3         9      10      1           7  Fage
## 400     7         5      2         6       8      4           2  Fage
```

```
summary(brand.ratings)
```

```
##      price          packaging        flavor         Freshness
##  Min.   : 1.000   Min.   : 1.00   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 4.000   1st Qu.: 4.00   1st Qu.: 3.000   1st Qu.: 3.000
##  Median : 7.000   Median : 6.00   Median : 6.000   Median : 5.000
##  Mean   : 6.442   Mean   : 5.94   Mean   : 6.032   Mean   : 5.062
##  3rd Qu.: 9.000   3rd Qu.: 8.00   3rd Qu.: 9.000   3rd Qu.: 7.000
##  Max.   :10.000   Max.   :10.00   Max.   :10.000   Max.   :10.000
```

```
##     Quality          trendy        Healthiness        brand
##  Min.   : 1.00   Min.   : 1.000   Min.   : 1.000   Length:400
##  1st Qu.: 4.00   1st Qu.: 2.000   1st Qu.: 4.000   Class :character
##  Median : 5.00   Median : 4.000   Median : 5.000   Mode  :character
##  Mean   : 5.53   Mean   : 4.312   Mean   : 5.522
##  3rd Qu.: 7.00   3rd Qu.: 6.000   3rd Qu.: 7.000
##  Max.   :10.00   Max.   :10.000   Max.   :10.000
```

```r
str(brand.ratings)
```

```
## 'data.frame':    400 obs. of  8 variables:
##  $ price      : int  10 3 1 8 8 7 10 8 9 6 ...
##  $ packaging  : int  6 5 1 7 10 9 7 8 7 4 ...
##  $ flavor     : int  7 10 6 8 10 6 10 7 5 8 ...
##  $ Freshness  : int  1 7 7 5 10 7 6 3 2 3 ...
##  $ Quality    : int  2 4 5 2 5 6 4 4 5 2 ...
##  $ trendy     : int  3 6 5 9 6 6 7 6 10 8 ...
##  $ Healthiness: int  5 5 2 5 7 7 4 5 3 4 ...
##  $ brand      : chr  "Danone" "Danone" "Danone" "Danone" ...
```

So, we have 7 numerical variables presenting the features of brands. The range of these variables is 1 to 10 which corresponds to the scores (rates) given by a consumer per each feature or brands.

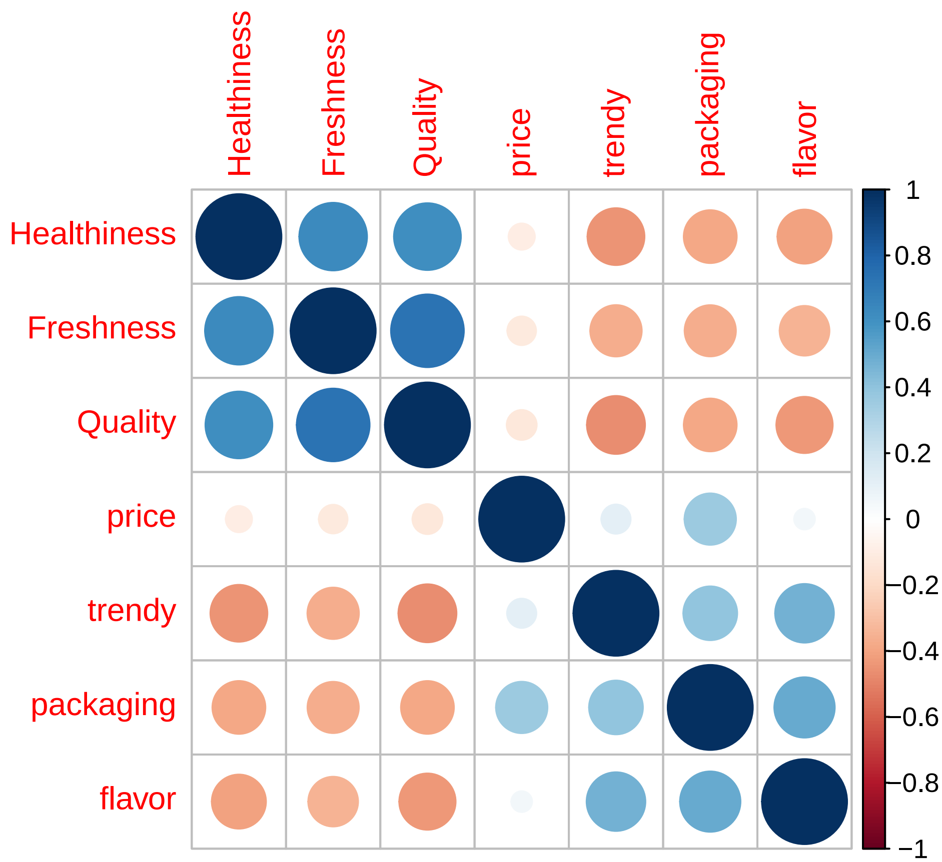
Rescale the data by normalizing or standardizing to get comparable numerical variables:

```r
brand.sc <- brand.ratings

brand.sc[, 1:7] <- scale(brand.ratings[, 1:7])

summary(brand.sc)
```

```
##      price            packaging            flavor            Freshness
##  Min.   :-1.8416   Min.   :-2.05373   Min.   :-1.69757   Min.   :-1.59654
##  1st Qu.:-0.8265   1st Qu.:-0.80652   1st Qu.:-1.02293   1st Qu.:-0.81055
##  Median : 0.1886   Median : 0.02494   Median :-0.01096   Median :-0.02456
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.8654   3rd Qu.: 0.85641   3rd Qu.: 1.00100   3rd Qu.: 0.76143
##  Max.   : 1.2038   Max.   : 1.68788   Max.   : 1.33833   Max.   : 1.94041
##     Quality            trendy          Healthiness          brand
##  Min.   :-1.9636   Min.   :-1.1884   Min.   :-1.9612   Length:400
##  1st Qu.:-0.6632   1st Qu.:-0.8296   1st Qu.:-0.6602   Class :character
##  Median :-0.2297   Median :-0.1121   Median :-0.2266   Mode  :character
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.6372   3rd Qu.: 0.6054   3rd Qu.: 0.6407
##  Max.   : 1.9376   Max.   : 2.0405   Max.   : 1.9416
```

Let's check the correlation between features:

```r
par(mfrow=c(1,1))
corrplot(cor(brand.sc[, 1:7]), order="hclust")
```



From the `corrplot` above we can see that there are two main "clusters" of features: healthiness - freshness - quality , and trendy-packaging - flavor. Note that the "price" variable seems to have correlation only with "packaging". Now, let's try to find out the average (mean) position of the brand on each feature:

```
brand.mean <- aggregate(.~ brand, data=brand.sc, mean)
brand.mean
```

```
##     brand      price  packaging      flavor Freshness    Quality     trendy
## 1 Activia -0.7249799 -0.3409021 -0.02445585 0.6003005  0.5461558 -0.4744666
## 2 Chobani  0.0431435  0.6277588  0.56923095 -0.9520314 -0.9752782  0.3506927
## 3  Danone  0.3476858  0.5279826  0.53212553 -0.2721493 -0.3597693  1.0538720
## 4    Fage  0.3341506 -0.8148392 -1.07690063  0.6238802  0.7888917 -0.9300981
##   Healthiness
## 1   0.7187668
## 2  -0.9247483
## 3  -0.5171218
## 4   0.7231033
```

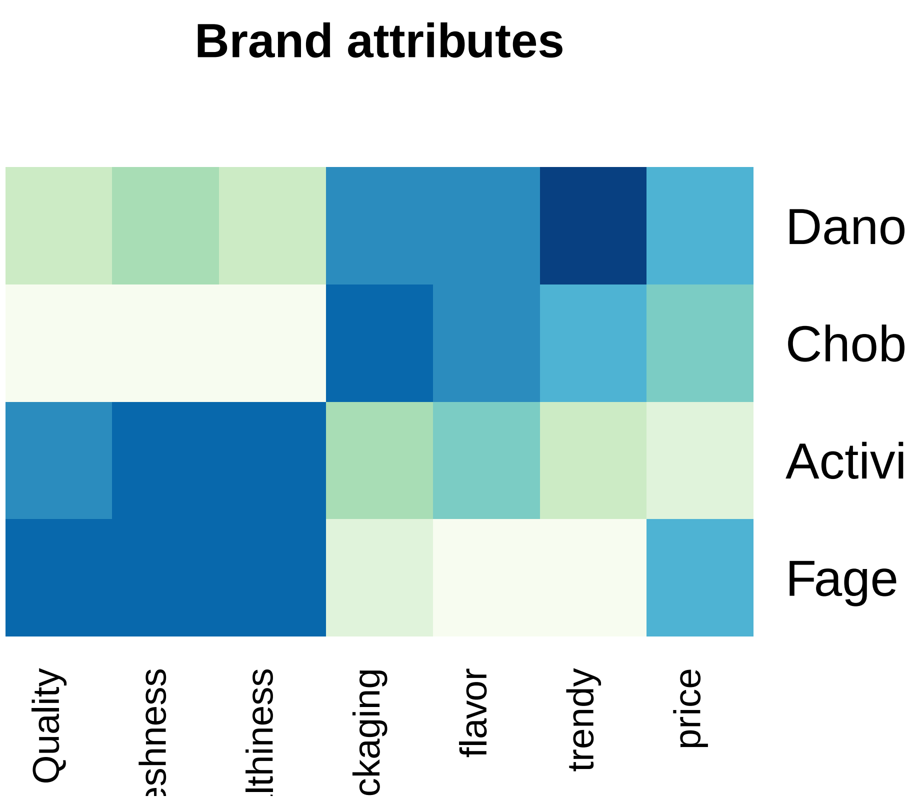
```
rownames(brand.mean) <- brand.mean[, 1] # use brand for the row names
brand.mean <- brand.mean[, -1] # remove brand name column

brand.mean
```

```
##              price  packaging      flavor Freshness    Quality     trendy
## Activia -0.7249799 -0.3409021 -0.02445585 0.6003005  0.5461558 -0.4744666
## Chobani  0.0431435  0.6277588  0.56923095 -0.9520314 -0.9752782  0.3506927
## Danone   0.3476858  0.5279826  0.53212553 -0.2721493 -0.3597693  1.0538720
## Fage     0.3341506 -0.8148392 -1.07690063  0.6238802  0.7888917 -0.9300981
```

```
##          Healthiness
## Activia   0.7187668
## Chobani  -0.9247483
## Danone   -0.5171218
## Fage      0.7231033
```

Tabular form of averages are not so easy to read. Let's visualize the data above using Heatmap for which green color indicates a low value and dark blue indicates a high value() lighter colors are for values in the middle of the range).

```
heatmap.2(as.matrix(brand.mean),
        col=brewer.pal(9, "GnBu"), trace="none", key=FALSE, dend="none",
        main="\nBrand attributes")
```



From the heatmap above we can understand the relationship of brands and their features compared to other brands' features. We can see that the "Danon" brand has high value for being "trendy"(dark blue) while the quality and freshness are rated by consumers quite low. The quality and price combination of "Fage" among presented brands is the highest according to the rates of consumers. For "Activia", consumers like the freshness and healthiness the most even though the flavor for this brand has the lowest rating. Consumers like the packaging and flavor of "Chobani" and at the same time they rated very low the price of this brand. This is interesting finding as you can see that with the"worst" price "Chobani" still is the second "trendy" brand.

Now let's move on to Principal component analysis (PCA). At first, let's look at the principal components for the brand rating data:

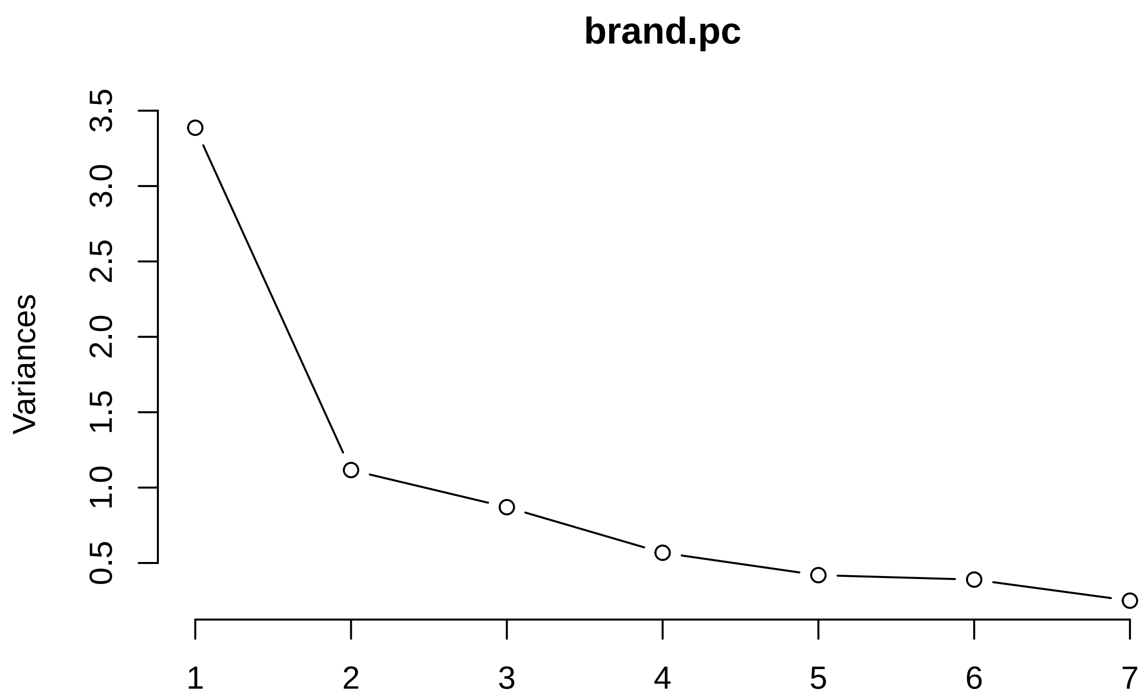
```
brand.pc <- prcomp(brand.sc[, 1:7])
```

```
summary(brand.pc)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8403 1.0566 0.9328 0.75351 0.64750 0.62425 0.50006
## Proportion of Variance 0.4838 0.1595 0.1243 0.08111 0.05989 0.05567 0.03572
## Cumulative Proportion  0.4838 0.6433 0.7676 0.84872 0.90861 0.96428 1.00000
```

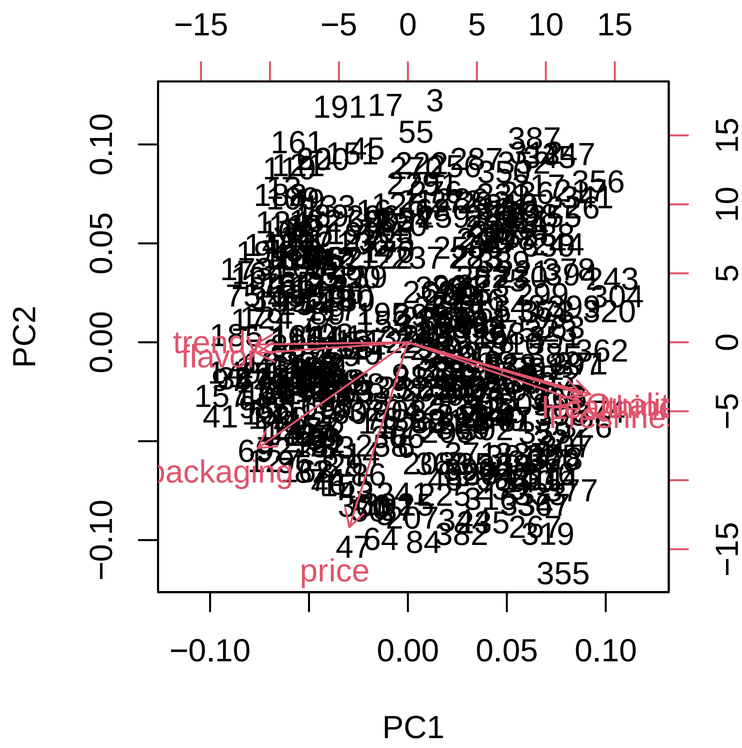
As always, let's visualize the table to get the maximum insight:

```
plot(brand.pc, type="l")
```



As we can see from the plot above most of the variances in our data can be captured using first 2 principal components and this will allow us to present the information with 2D figures.

```
biplot(brand.pc)
```

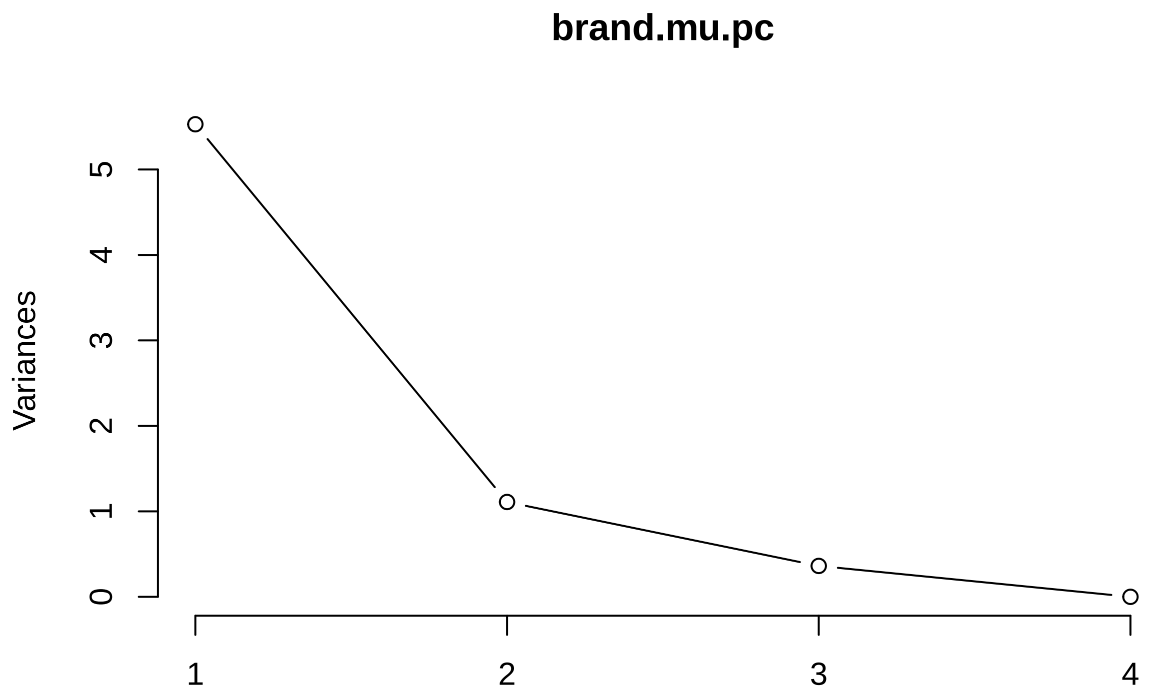


The figure above shows the positions of rating adjectives when usinf first and second principal components. However, using individual rates makes the map above to dense and difficult to read. Instead of using the actual data let's do the same for averages (developed above):

```
brand.mu.pc <- prcomp(brand.mean, scale=TRUE)
summary(brand.mu.pc)
```

```
## Importance of components:
##                           PC1    PC2     PC3       PC4
## Standard deviation     2.3515 1.0533 0.60109 8.052e-17
## Proportion of Variance 0.7899 0.1585 0.05162 0.000e+00
## Cumulative Proportion  0.7899 0.9484 1.00000 1.000e+00
```

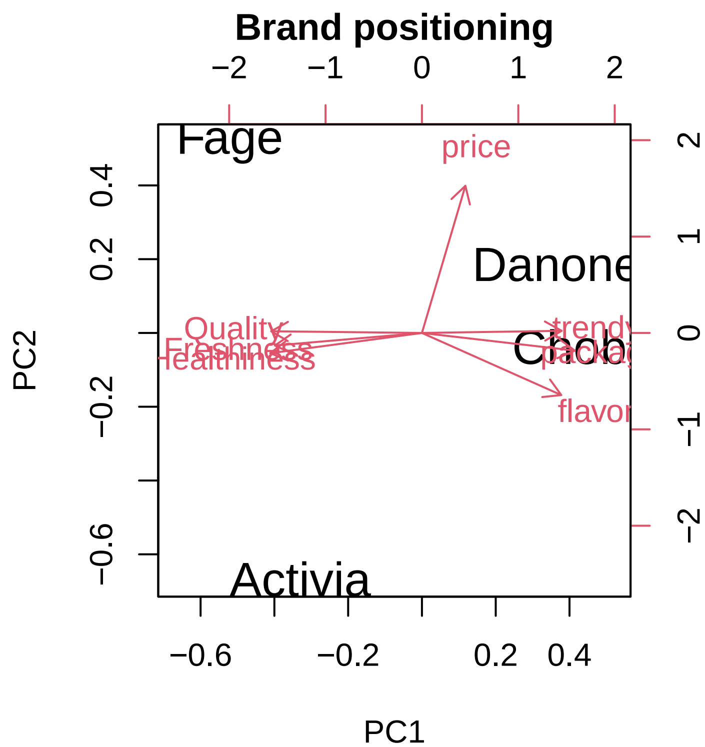
```
plot(brand.mu.pc,type="l")
```



Note that for the aggregated data the first two principal components capture about 95 % variability of our data. In other words, we can analyze our large dataset by using only 2 dimensions(principal components) of aggregated data as they are representative enough. That is, we reduced dimensions of data to 2 while keeping the 95% information and variations of actual data.

Finally, it is time to get the easy-to-interpret perceptual map:

```
biplot(brand.mu.pc, main="Brand positioning", cex=c(1.5, 1))
```



The perceptual map above is the end goal of our analysis that shows the positions of brands in the market with respect to the product features using principal components. For the "Activia" and "Fage" market looks well differentiated: though both of them are rated high for quality, healthiness, and freshness , "Activia" is over-priced based on consumers' ratings. "Chobani" is the brand that is favorite among consumers by it's flavor and packaging. At the same time, brand "Danone" having similar ratings as "Chobani", seems to be priced more fairly according to consumers.

Based on this analysis we might provide a suggestion to the market players. For example, suppose the management of "Danone" thinks that the market segment occupied by "Fage" yogurts has a good potential for their product too. That is, they would like to position "Danone" closer to "Fage" in the map above :

```
brand.mean["Fage", ] - brand.mean["Danone", ]
```

```
##            price packaging    flavor Freshness  Quality   trendy Healthiness
## Fage -0.01353521 -1.342822 -1.609026 0.8960295 1.148661 -1.98397    1.240225
```

To accomplish this market positioning , "Danone" team needs to pay more attention on Quality and Healthiness of their product while giving up some of their efforts aimed to the trendiness, packaging and flavor.

What if, instead of following another brand, "Danone" management team aimed for differentiated space where no brand is positioned between "Chobani" and "Fage". Assuming that the gap reflects approximately the average of those two brands we can calculate the differences of average values of competitors and their values to get the numeric indicators. These indicators will show which features need to be prioritized to achieve the desired result:

```
colMeans(brand.mean[c("Chobani",  "Fage"), ]) - brand.mean["Danon", ]
```

```
##             price  packaging     flavor Freshness  Quality    trendy
## Danone -0.1590388 -0.6215228 -0.7859604 0.1080737 0.266576 -1.343575
```

```
##        Healthiness
## Danone   0.4162993
```

From the positive values in table above we can conclude that “Danone” can achieve better differentiation by improving healthiness, quality and freshness of their yogurts.