535_HW3

Shijie Gao, USC ID:6037-6293-25

2023-02-01

```
setwd("C:/Users/GAOSHIJIE/Desktop")
df = read.csv("segment.csv")
#head(df)
#1
#a Two way table for subscribers and home owners
two_way_table = table(df$subscribe, df$ownHome)
two_way_table
```

```
##
##          ownNo ownYes
##   subNo    137    123
##   subYes    22     18
```

```
#b Hypothesis H_0: m_1 = m_2, where m_i is the proportion of home owner and
#home renter, let significance level to be alpha = 0.05, as we don't know which
#one is bigger, so H_a: m_1 not equal to m_2, which is a two-side test

sum = apply(two_way_table, 2, sum) #sum of ownNo and ownYes
sum_sub = apply(two_way_table, 1, sum) #sum of subNo and subYes

p_1 = two_way_table[2, 1] / sum[1] #proportion of subYes/ownNO
p_2 = two_way_table[2, 2] / sum[2] #proportion of subYes/ownYes

p_p = sum_sub[2] / (sum[1]+sum[2]) #pool proportion of subYes
p_1
```

```
##     ownNo
## 0.1383648
```

```
p_2
```

```
##    ownYes
## 0.1276596
```

```
p_p
```

```
##    subYes
## 0.1333333
```

```
#observed statistic after normalized
z = (p_1 - p_2) / sqrt(p_p*(1-p_p)*(1/sum[1] + 1/sum[2]))

p_value = 1 - pnorm(z) #p_value
p_value
```

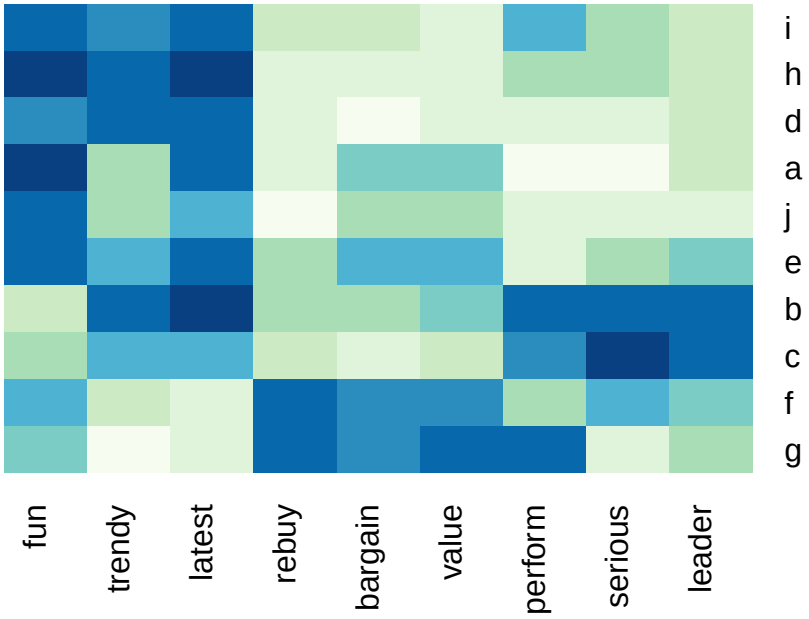
```
##     ownNo
```

```
## 0.3927199
```

```
#As p_value = 0.3927199 > 0.025, so we fail to reject H_0, which means
#We think subscribers is the same between home owners and home renters at
#a significance level of alpha = 0.05(alpha = 0.1 will also accept H_0)

#2
setwd("C:/Users/GAOSHIJIE/Desktop")
df = read.csv("brands.csv")
#head(df)

#a Average rating of each brand on each attribute and store
df1 = aggregate(cbind(perform, leader, latest, fun, serious, bargain,
                      value, trendy, rebuy)~brand, df, mean)
rownames(df1) = df1$brand
df1$brand = NULL
df1
```

```
##   perform leader latest  fun serious bargain value trendy rebuy
## a    1.65   3.04   7.46 7.87    1.77    4.83  4.78   3.78  2.21
## b    7.47   7.21   8.43 3.40    7.61    4.37  4.70   7.25  4.33
## c    6.57   7.45   5.88 3.75    7.72    2.64  3.28   5.29  3.39
## d    2.31   2.87   7.28 6.58    2.40    1.91  2.10   7.24  2.47
## e    2.68   4.92   7.60 6.88    4.44    5.73  5.34   5.60  3.82
## f    4.30   5.12   2.31 5.47    5.96    6.59  6.79   2.99  7.18
## g    7.43   3.98   2.24 4.65    2.84    6.65  7.35   1.72  7.19
## h    4.44   3.64   7.74 8.03    3.93    2.29  2.46   7.59  2.19
## i    5.56   3.58   7.29 7.20    3.91    3.58  2.41   6.84  3.21
## j    2.47   2.36   5.72 6.85    2.65    4.00  4.16   3.90  1.28
```

```
#b Display a heatmap using the average ratings from df1
library(gplots)
library(RColorBrewer)
heatmap.2(as.matrix(df1),col=brewer.pal(9, "GnBu"),
          trace="none", key=FALSE, dend="none",main="")
```




```
#Through heatmap, brand b and c are highly rated on attributes leader,
#and serious

#c
df$brand = NULL
head(df)
```

```
##   perform leader latest fun serious bargain value trendy rebuy
## 1       2      4      8   8       2       9     7      4     6
## 2       1      1      4   7       1       1     1      2     2
## 3       2      3      5   9       2       9     5      1     6
## 4       1      6     10   8       3       4     5      2     1
## 5       1      1      5   8       1       9     9      1     1
## 6       2      8      9   5       3       8     7      1     2
```

```
prcomp1 = prcomp(df, center = TRUE, scale = TRUE) #do pca on df
#and prcomp finishes centering the data in its function
prcomp1
```

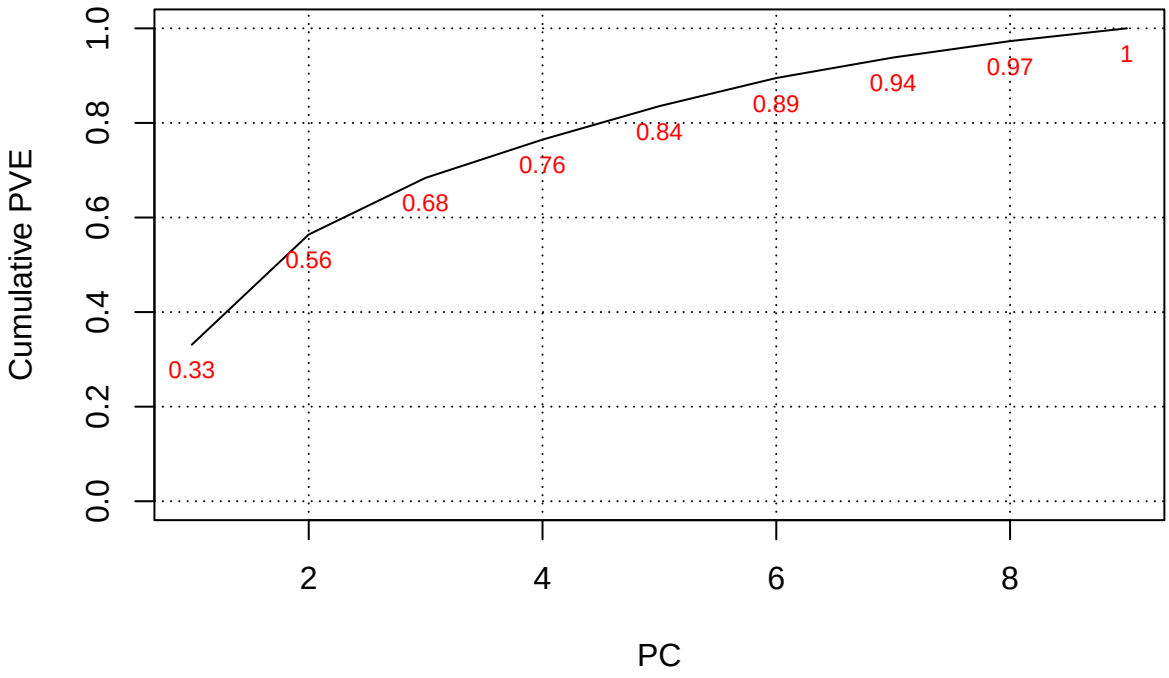
```
## Standard deviations (1, .., p=9):
## [1] 1.7260636 1.4479474 1.0388719 0.8527667 0.7984647 0.7313298 0.6245834
## [8] 0.5586112 0.4930993
##
## Rotation (n x k) = (9 x 9):
##                PC1         PC2         PC3         PC4         PC5         PC6
## perform  0.2374679  0.41991179  0.03854006 -0.52630873  0.46793435 -0.3370676
## leader   0.2058257  0.52381901 -0.09512739 -0.08923461 -0.29452974 -0.2968860
```

```
## latest  -0.3703806  0.20145317 -0.53273054  0.21410754  0.10586676 -0.1742059
## fun     -0.2510601 -0.25037973 -0.41781346 -0.75063952 -0.33149429  0.1405367
## serious  0.1597402  0.51047254 -0.04067075  0.09893394 -0.55515540  0.3924874
## bargain  0.3991731 -0.21849698 -0.48989756  0.16734345 -0.01257429 -0.1393966
## value    0.4474562 -0.18980822 -0.36924507  0.15118500 -0.06327757 -0.2195327
## trendy  -0.3510292  0.31849032 -0.37090530  0.16764432  0.36649697  0.2658186
## rebuy    0.4390184  0.01509832 -0.12461593 -0.13031231  0.35568769  0.6751400
##                 PC7         PC8         PC9
## perform  0.364179109 -0.14444718  0.05223384
## leader  -0.613674301  0.28766118 -0.17889453
## latest  -0.185480310 -0.64290436  0.05750244
## fun     -0.007114761  0.07461259  0.03153306
## serious  0.445302862 -0.18354764  0.09072231
## bargain  0.288264900  0.05789194 -0.64720849
## value    0.017163011  0.14829295  0.72806108
## trendy   0.153572108  0.61450289  0.05907022
## rebuy   -0.388656160 -0.20210688 -0.01720236
```

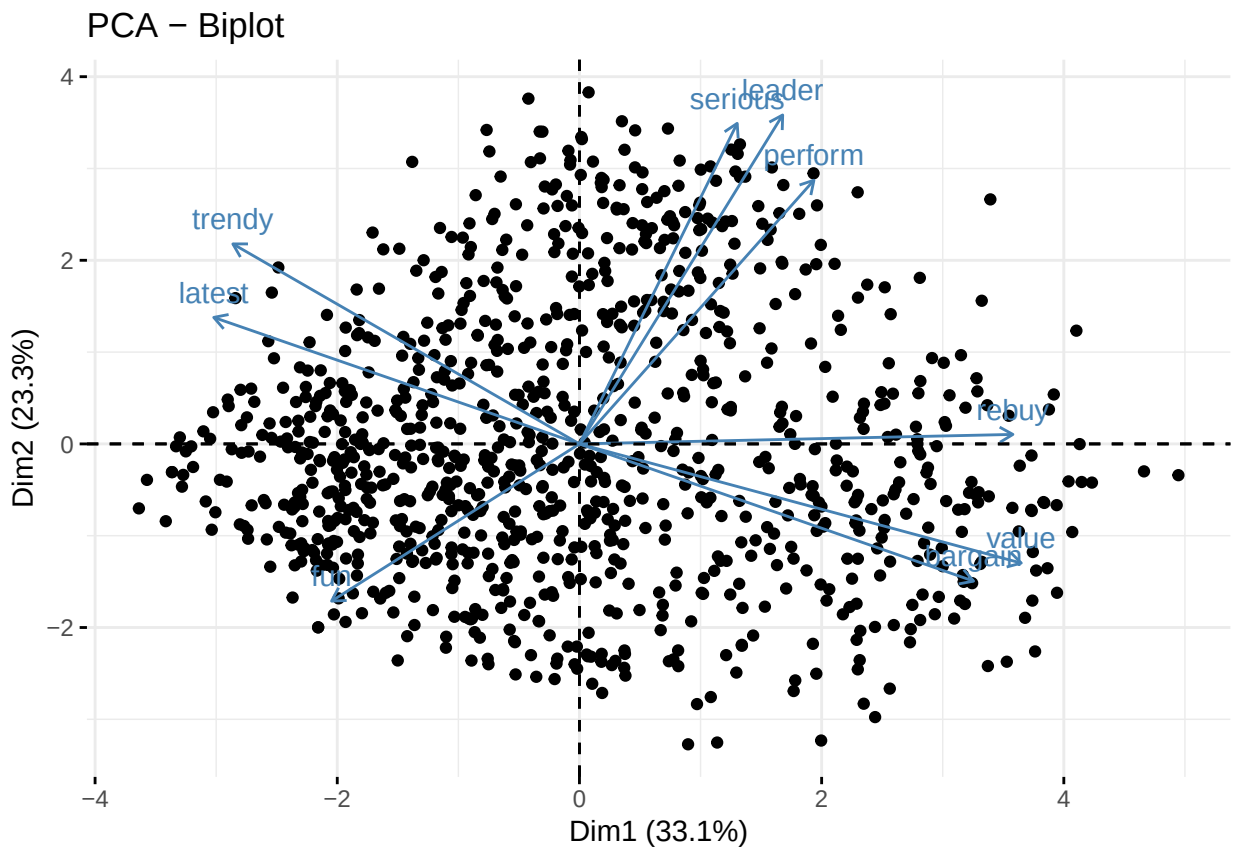
```
summary(prcomp1)#get info of Standard deviation, Proportion of Variance and
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.726 1.4479 1.0389 0.8528 0.79846 0.73133 0.62458
## Proportion of Variance 0.331 0.2329 0.1199 0.0808 0.07084 0.05943 0.04334
## Cumulative Proportion  0.331 0.5640 0.6839 0.7647 0.83554 0.89497 0.93831
##                           PC8     PC9
## Standard deviation     0.55861 0.49310
## Proportion of Variance 0.03467 0.02702
## Cumulative Proportion  0.97298 1.00000
```

```
#Cumulative Proportion. This item can not transfer into df

var_eig = prcomp1$sdev^2 #get variances = eigenvalues

PVE = var_eig/sum(var_eig) #proportion of var_eig
CPVE = cumsum(PVE) #cumulative of PVE

#draw the line
plot(CPVE, xlab = "PC", ylab = "Cumulative PVE", type = "l", ylim = c(0, 1))
text(CPVE, labels = round(CPVE, 2), cex = 0.75, pos = 1, offset = 0.5,
     col = "red") #label for each var_eig
grid(col = "black") #draw the grid
#five principle components explain at least 80% of the variation

#d  Construct a biplot from prcomp1
library(factoextra)
```




```
fviz_pca_biplot(prcomp1, label = "var")
```

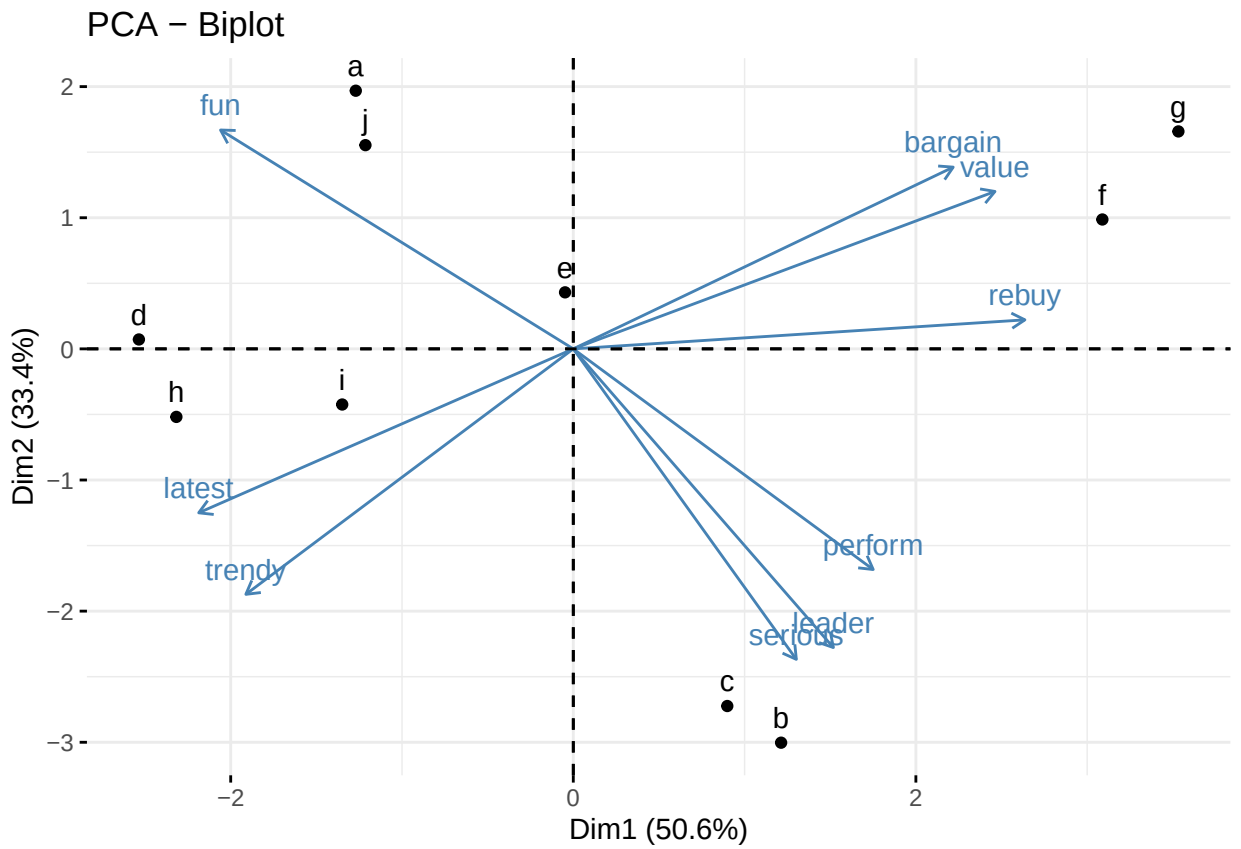



```
#e Find principal components from df1. Use prcomp2 = prcomp(df1,scale=T)
prcomp2 = prcomp(df1,scale=T)
#prcomp2$x
prcomp2
```

```
## Standard deviations (1, .., p=9):
## [1] 2.13452052 1.73494730 0.76898915 0.61498280 0.50982614 0.36661576 0.21506243
## [8] 0.14588236 0.04866747
##
## Rotation (n x k) = (9 x 9):
##                PC1         PC2         PC3         PC4         PC5         PC6
## perform  0.2852486 -0.33729698 -0.48121446  0.46995620  0.39623804 -0.43471514
## leader   0.2473668 -0.45654557  0.31711577 -0.19084693 -0.06130157 -0.11868317
## latest  -0.3562989 -0.25056983  0.49589600  0.27477470  0.46061874  0.08173299
## fun     -0.3355152  0.33455495  0.15167546  0.32394053 -0.38757837 -0.63609709
## serious  0.2121240 -0.47463096  0.24371327 -0.21229430 -0.39428137 -0.33437227
## bargain  0.3613409  0.27776101  0.45940272  0.29120398  0.11248446 -0.12716342
## value    0.4010778  0.24062869  0.33576144  0.05052374  0.20581208  0.08329187
## trendy  -0.3114405 -0.37521575  0.08724910  0.48392969 -0.27261916  0.33925412
## rebuy    0.4295359  0.04438337 -0.09031492  0.44234693 -0.43824713  0.36828116
##                PC7          PC8          PC9
## perform  0.02784431 -0.074243080  0.012984626
## leader  -0.60997229 -0.021119910  0.450594077
## latest  -0.19587019  0.119316063 -0.466262266
## fun     -0.24602385 -0.179248006 -0.008094488
## serious  0.43881277 -0.005157446 -0.406716076
```

```
## bargain  0.31905166  0.512721569  0.320827507
## value    0.08325891 -0.778125659 -0.065102236
## trendy   0.32150758 -0.243224760  0.410460300
## rebuy   -0.35159046  0.141872872 -0.371841553
```

```
#f
fviz_pca_biplot(prcomp2)
```




```
#Average position of the brand on each attribute can be seen by biplot,
#the brands with the same direction of one particular attribute is highly rated,
#on the opposite, the brands with the opposite direction of one particular
#attribute is lowly rated, for more specific, the comparison is about the length
#for the projection of each points on the vector of each attribute

#brands b,c are highly rated on attributes leader, and serious
#brands f,g are highly rated on bargain, and value
```