PCA

Chase Enzweiler

9/18/2017

Principal Component Analysis

```
wholesale <- read.csv("~/Desktop/stat 154/wholesale.csv")
```

convert channel and region variables to factors

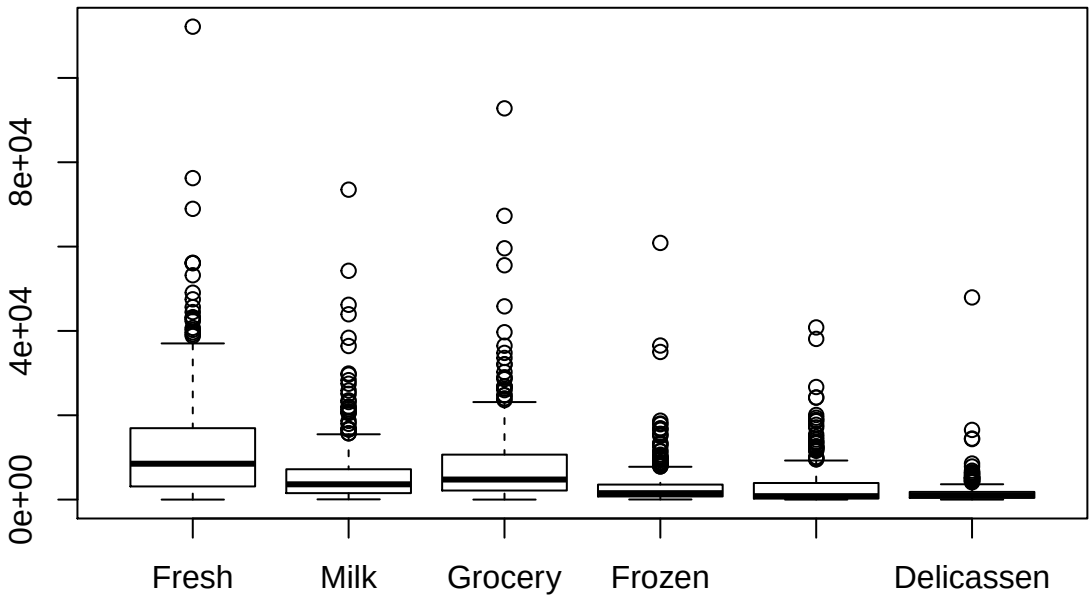
```
# factor and horeca = 1 and retail = 2
wholesale$Channel[wholesale$Channel == 1] <- "Horeca"
wholesale$Channel[wholesale$Channel == 2] <- "Retail"
wholesale$Channel <- as.factor(wholesale$Channel)

# 1= lisbon, 2 = oporto, 3 = other
wholesale$Region[wholesale$Region == 1] <- "Lisbon"
wholesale$Region[wholesale$Region == 2] <- "Oporto"
wholesale$Region[wholesale$Region == 3] <- "Other"
wholesale$Region <- as.factor(wholesale$Region)
```

perform exploratory data analysis

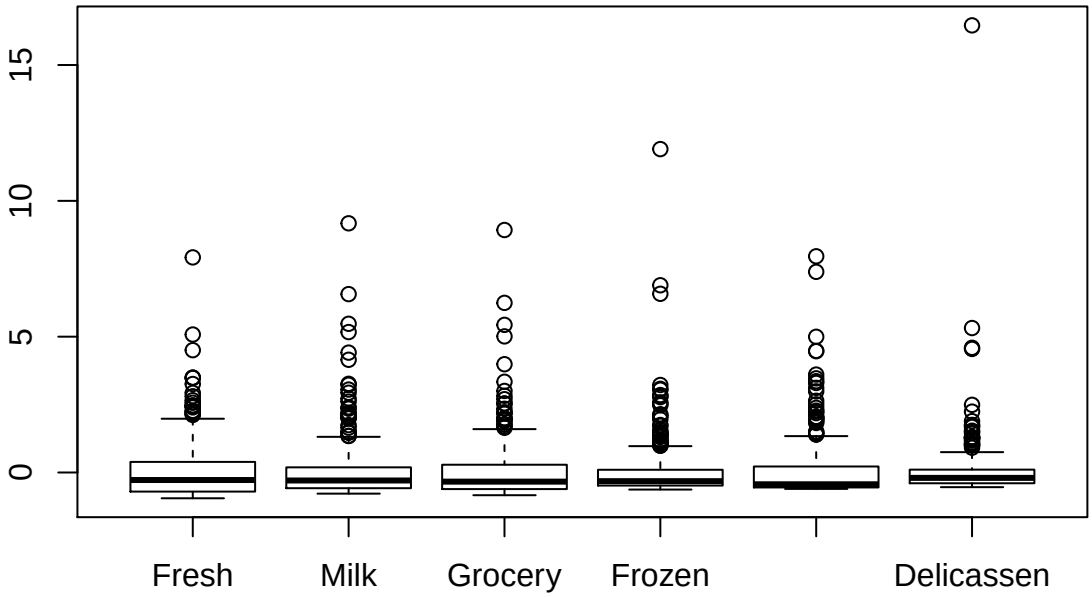
```
#summary statistics
summary(wholesale)
```

```
##    Channel       Region        Fresh             Milk
##  Horeca:298   Lisbon: 77   Min.   :     3   Min.   :   55
##  Retail:142   Oporto: 47   1st Qu.:  3128   1st Qu.: 1533
##               Other :316   Median :  8504   Median : 3627
##                            Mean   : 12000   Mean   : 5796
##                            3rd Qu.: 16934   3rd Qu.: 7190
##                            Max.   :112151   Max.   :73498
##     Grocery          Frozen        Detergents_Paper    Delicassen
##  Min.   :    3   Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
##  Median : 4756   Median : 1526.0   Median :  816.5   Median :  965.5
##  Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
##  3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
##  Max.   :92780   Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```
#boxplots for quant. varibles and standardized quant. variables
boxplot(wholesale[,3:8])
```




```
boxplot(scale(wholesale[,3:8]))
```

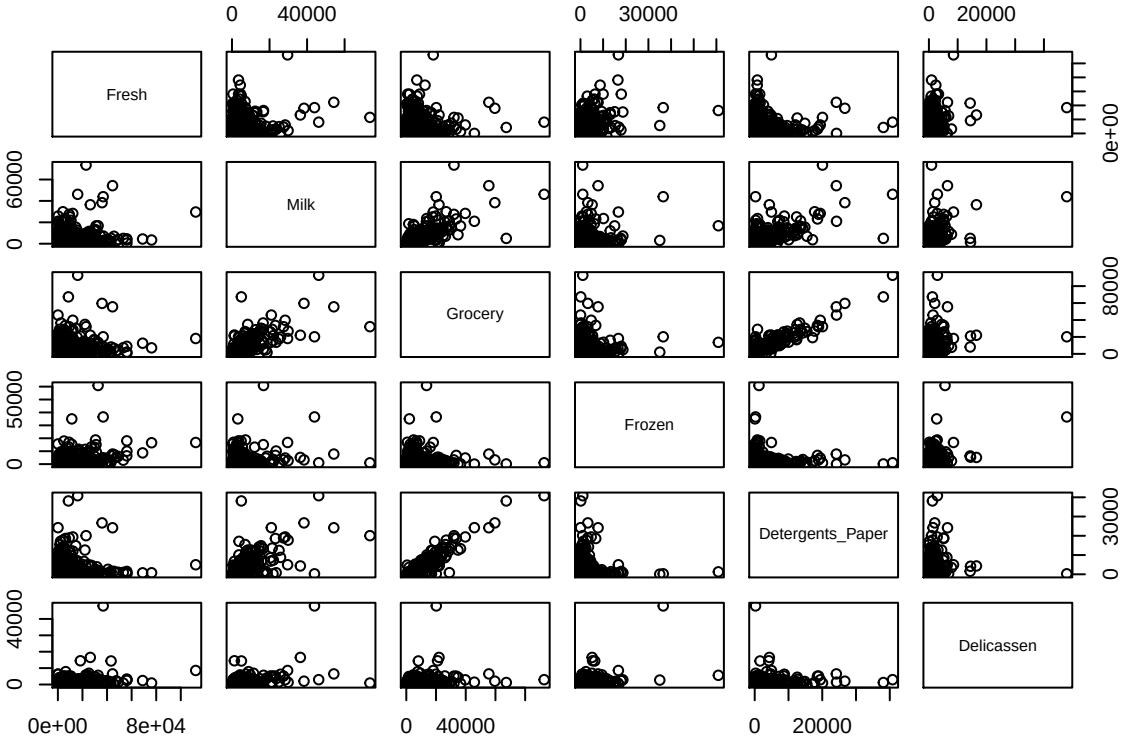



```
# correlation matrix
cor(wholesale[,3:8])
```

```
##                       Fresh      Milk     Grocery      Frozen
## Fresh             1.00000000 0.1005098 -0.01185387  0.34588146
## Milk              0.10050977 1.0000000  0.72833512  0.12399376
## Grocery          -0.01185387 0.7283351  1.00000000 -0.04019274
## Frozen            0.34588146 0.1239938 -0.04019274  1.00000000
## Detergents_Paper -0.10195294 0.6618157  0.92464069 -0.13152491
## Delicassen        0.24468997 0.4063683  0.20549651  0.39094747
##                  Detergents_Paper Delicassen
## Fresh                  -0.1019529  0.2446900
## Milk                    0.6618157  0.4063683
## Grocery                 0.9246407  0.2054965
```

```
## Frozen                -0.1315249  0.3909475
## Detergents_Paper       1.0000000  0.0692913
## Delicassen             0.0692913  1.0000000
```

```
# pairs plot
pairs(wholesale[,3:8])
```



we see not a lot of spread in the different scatterplots ecept for the the milk and grocery scatter plot which has the closest resemblance to a linear relationship.

PCA with prcomp()

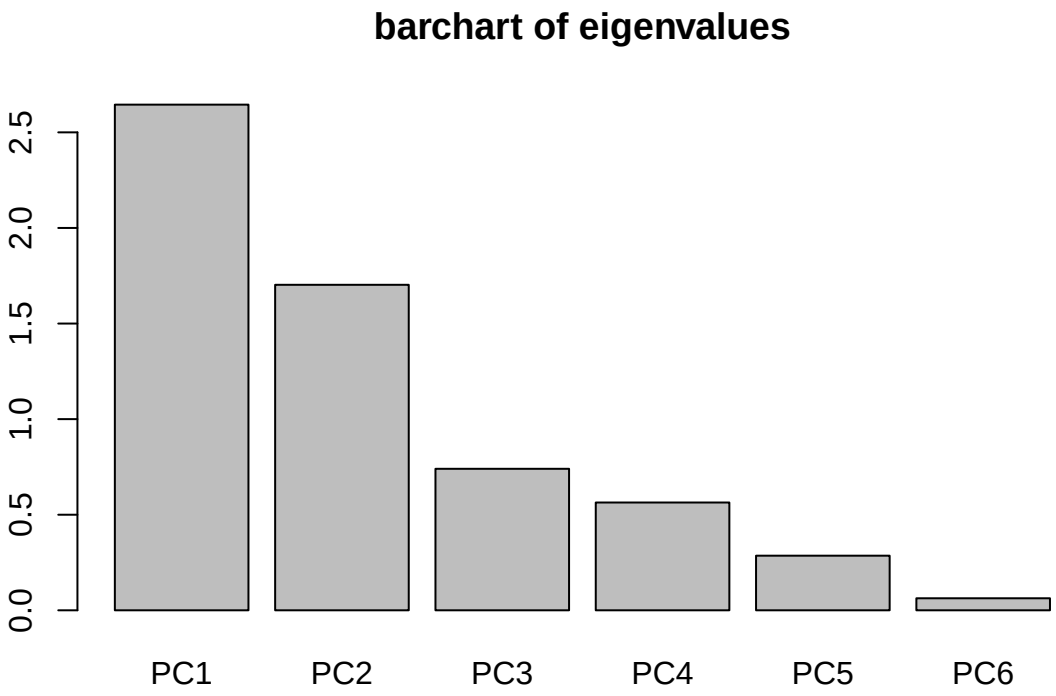
What are the differences between prcomp() and princomp()?
Princomp() is calculated using eigen decomposition of a correlation matrix (1/n)t(x)x or covariance matrix. it is done for compatibility with SPLUS. it uses divisor n for the covariance matrix. It prints results in a nice format to and has plot methods to produce scree plots and a biplot method. May not necesarilly center data beforehand? prints the number of observations
prcomp() is calculated using singular value decomposition and not eigen decompostition on a covariance matrix. SVD is used for numerical accuracy. print method prints in a nice format and has a plot method for scree plots. Variances are computed using the divisor N - 1
it would be better to use prcomp() most of the time because it uses SVD which produces better numerical accuracy. princomp may be better sometimes if we wanted to create a biplot because it has a biplot method.
Stages of PCA

```
pca <- prcomp(wholesale[,3:8], scale. = TRUE)

# eigenvalues are the variances of the pc's
eigenvalues <- (pca$sdev)^2
proportion <- eigenvalues/6

cum_prop <- c(proportion[1])
```

```
for(i in 2:6){
  cum_prop <- append(cum_prop, sum(proportion[1:i]))
}

eigenvalue_table <- cbind(eigenvalues, proportion, cum_prop)
rownames(eigenvalue_table) <- paste("PC", 1:6, sep = "")

eigenvalue_table <- as.table(eigenvalue_table)

eigenvalue_table
```

```
##     eigenvalues proportion   cum_prop
## PC1  2.64497357 0.44082893 0.44082893
## PC2  1.70258397 0.28376400 0.72459292
## PC3  0.74006477 0.12334413 0.84793705
## PC4  0.56373023 0.09395504 0.94189209
## PC5  0.28567634 0.04761272 0.98950481
## PC6  0.06297111 0.01049519 1.00000000
```

```
barplot(eigenvalue_table[,1], main = "barchart of eigenvalues")
```



about 43 percent of the total variance is captured din the first PC. About 33 percent of the total variance is captured in the second PC. About 77 percent of the total variation is captured by the first two PC's

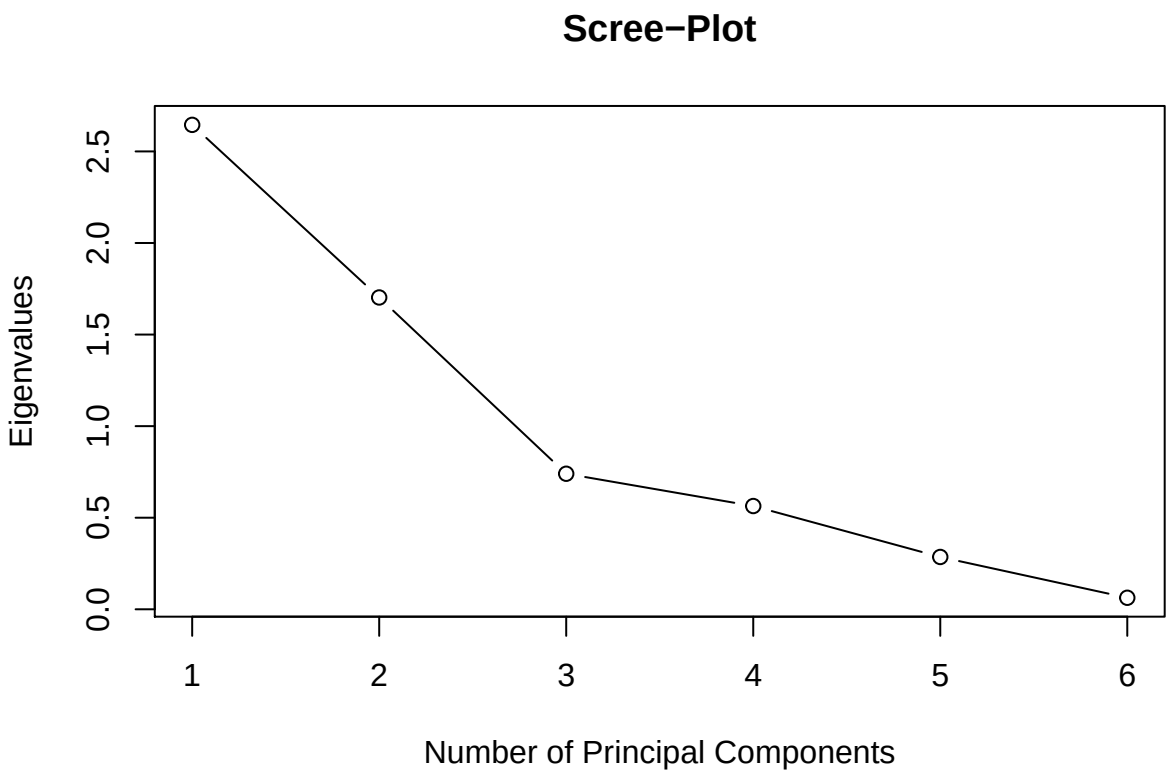
Choosing the number of components

- to capture 70 percent of the total variatiion we would choose two PC's

```
#eigenvalue average
sum(eigenvalues)/6
```

```
## [1] 1
```

```
plot(1:6, eigenvalues, ylab = "Eigenvalues", type = "b", xlab = "Number of Principal Components", main =
```



/ * using the average eigenvalue criterion where our average is 1, we would keep the first two PC's * kaiser rule is the same as above keep the first 2 PC's * With jollifes rule we still keep the first two PC's * looking at the scree plot we could keep the first two or three PC's

Variable loadings and correlations with PC's

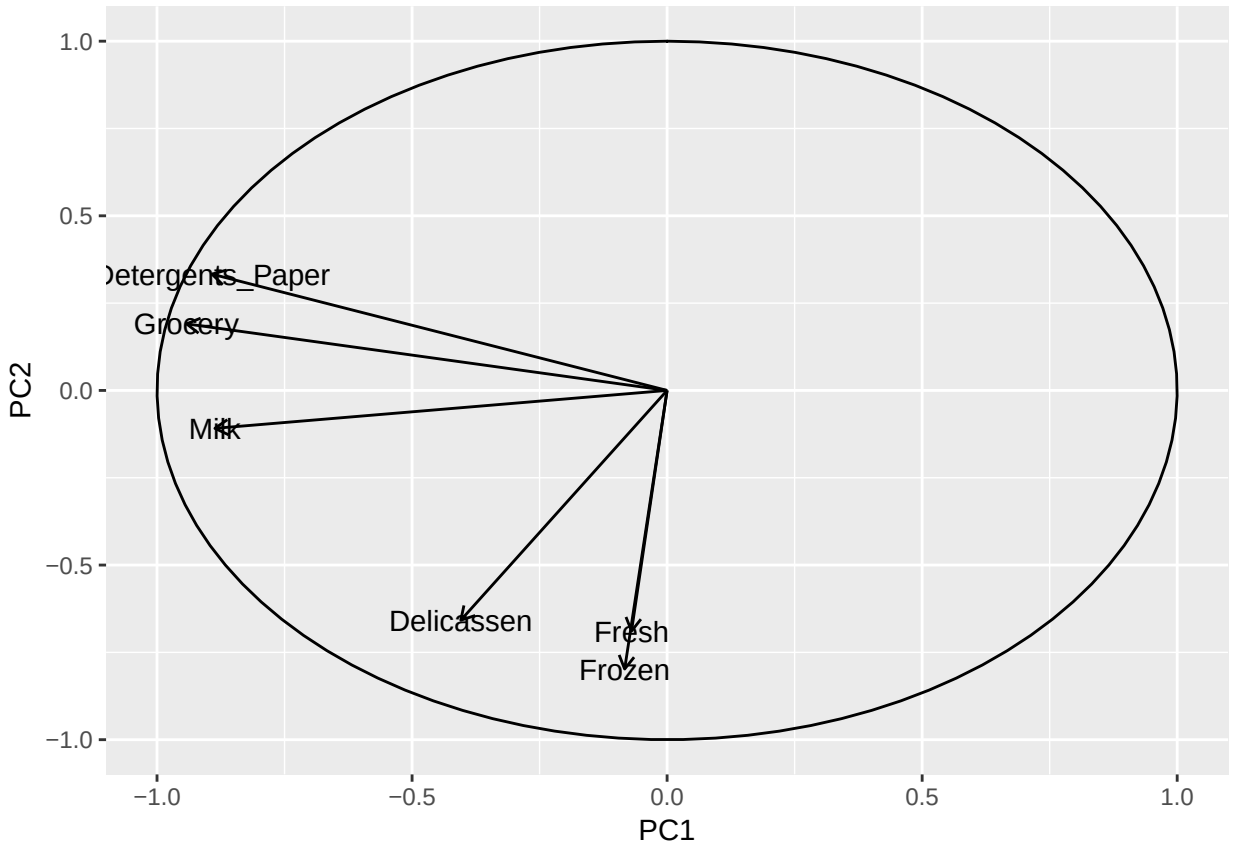
```
# calculate the correlations between the active variables and the Pc's

correlations <- cor(wholesale[,3:8], pca$x)
```

graph the circle of correlations

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
correlations <- as.data.frame(correlations)

radians <- seq(0, 2*pi, length = 100)
circle_frame <- data.frame(x = sin(radians), y = cos(radians))

ggplot(data = correlations, aes(PC1, PC2)) + geom_segment(data = correlations, aes(x = 0, y = 0, xend =
  geom_text(data = correlations, label = rownames(correlations))
```




```
# contributions of observations
contr <- t(t(pca$x ^ 2) / eigenvalues)

contr[1:6,]
```

```
##              PC1        PC2         PC3        PC4       PC5          PC6
## [1,] 0.014093269 0.05454910 0.026756620 0.41877812 0.8567256 0.0008709474
## [2,] 0.071188513 0.06320377 0.137196013 0.05660086 0.4667677 0.0471854145
## [3,] 0.248190759 0.38933237 3.128808032 2.78350544 0.5018092 1.2204421228
## [4,] 0.228703423 0.24969051 0.035824712 0.25564956 0.2663107 0.0584115578
## [5,] 0.010430580 0.94730701 0.005922419 1.20820043 0.5427431 0.0114008784
## [6,] 0.009199947 0.05104608 0.029375488 0.30966487 0.8028474 0.0460996372
```

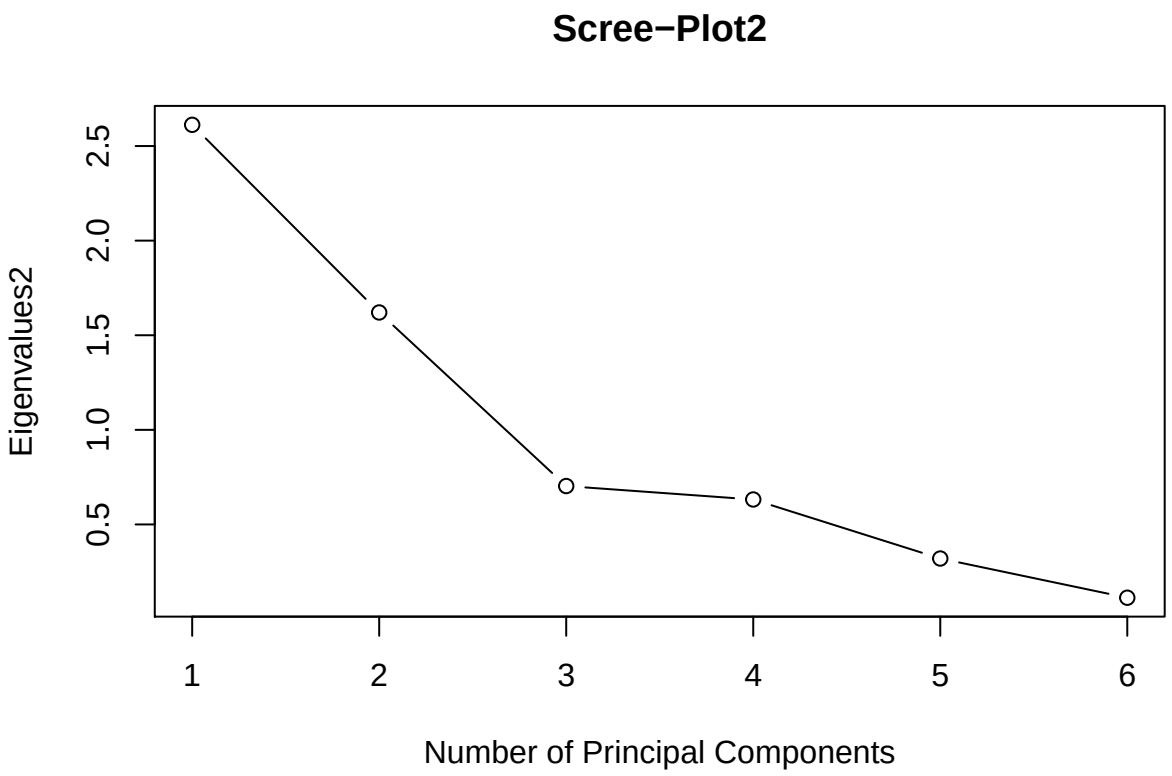
delete observations with contributions of 10 or more on PC1

```
rows_to_del <- which(contr[,1] >= 10)

wholesale2 <- wholesale[-rows_to_del,]
```

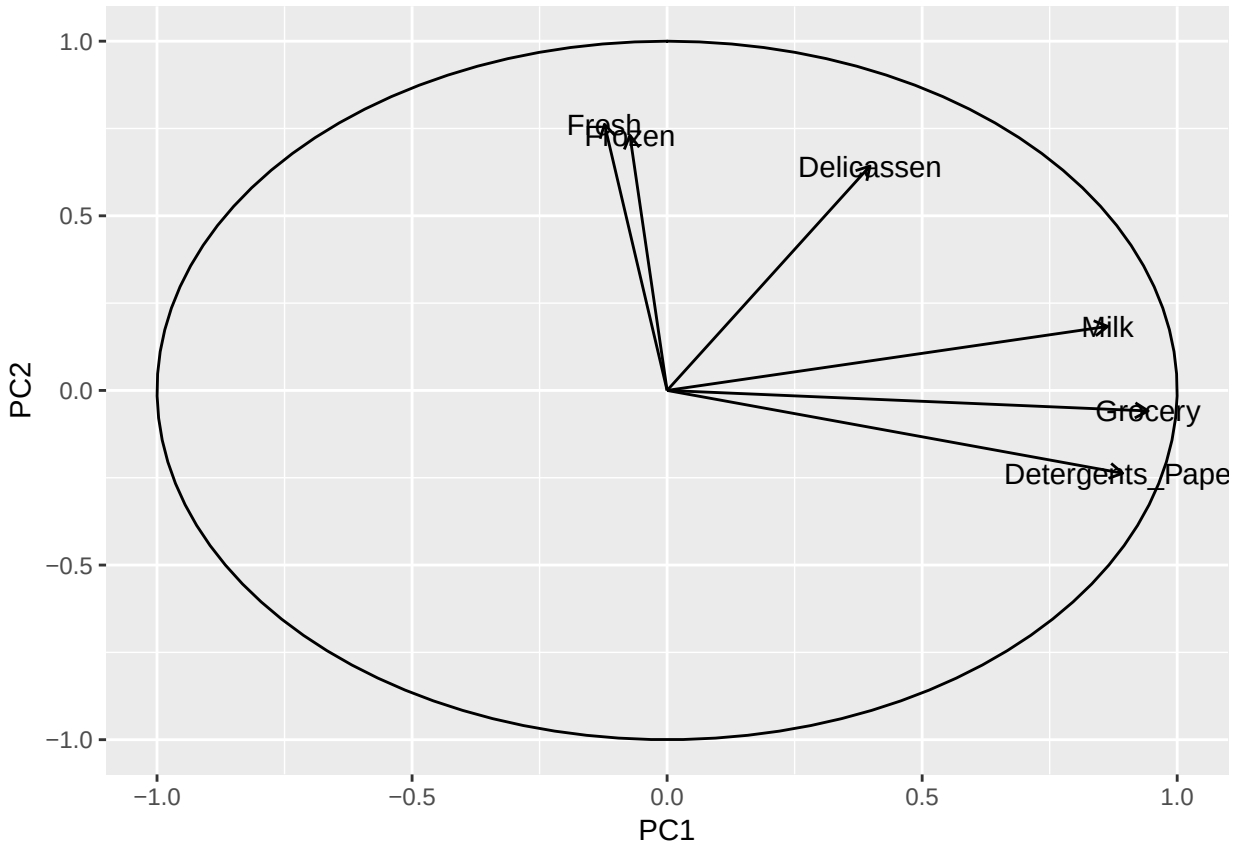
now perform all the before steps on the data wholesale2

```
pca2 <- prcomp(wholesale2[,3:8], scale. = TRUE)

eigenvalues2 <- pca2$sdev ^ 2

proportion2 <- eigenvalues2/6

cum_prop2 <- c(proportion2[1])
```

```
for(i in 2:6){
  cum_prop <- append(cum_prop2, sum(proportion2[1:i]))
}

plot(1:6, eigenvalues2, ylab = "Eigenvalues2", type = "b", xlab = "Number of Principal Components", mai
```




```
correlations2 <- cor(wholesale2[,3:8], pca2$x)


correlations2 <- as.data.frame(correlations2)

radians <- seq(0, 2*pi, length = 100)
circle_frame <- data.frame(x = sin(radians), y = cos(radians))

ggplot(data = correlations2, aes(PC1, PC2)) + geom_segment(data = correlations2, aes(x = 0, y = 0, xend
  geom_text(data = correlations2, label = rownames(correlations2))
```



biplot

```
biplot(pca, scale = 0)
```