Candies

Azat

11 марта 2021 г

## Loading required package: ggplot2

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

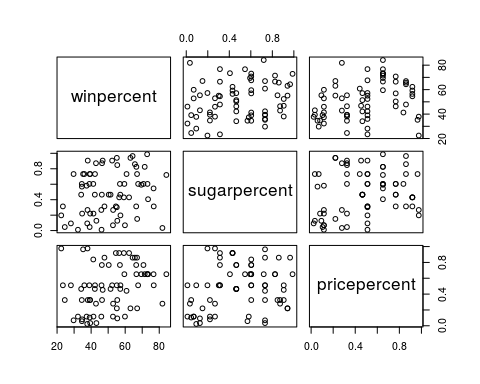
##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

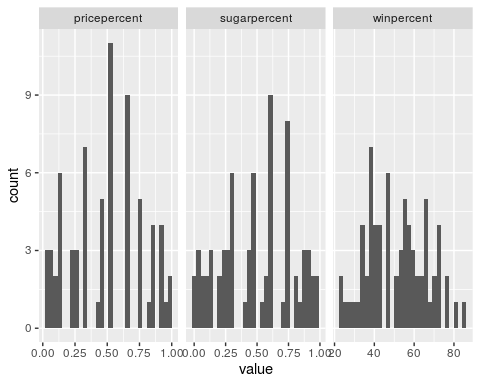
features <- list()  
features$composition <- select(candies, !c(competitorname, Y, winpercent, sugarpercent, pricepercent))  
  
features$percent <- select(candies, c(winpercent, sugarpercent, pricepercent))

pairs(features$percent)

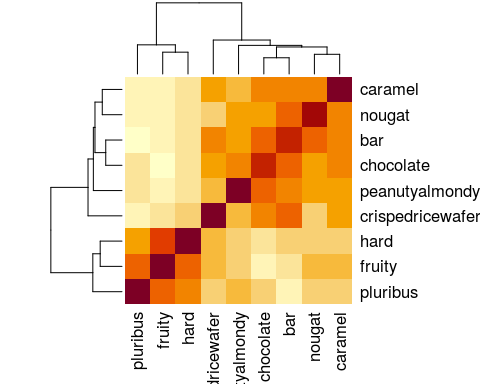


features$percent %>% gather() %>% ggplot(aes(x = value)) + facet\_grid(.~key, scales = "free") + geom\_histogram()

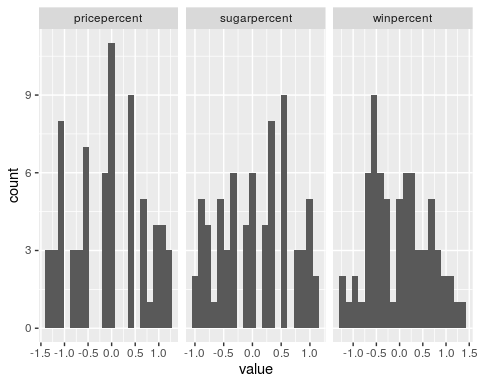
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



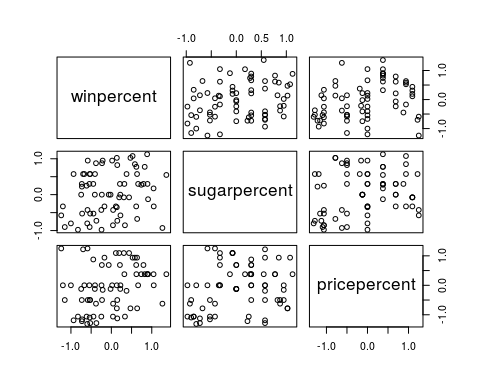
# cor matrix of nominative data  
phi\_matrix <- sapply(colnames(features$composition), function(x) sapply(colnames(features$composition), function(y) phi(table(features$composition[c(x,y)]))))  
  
heatmap(phi\_matrix)



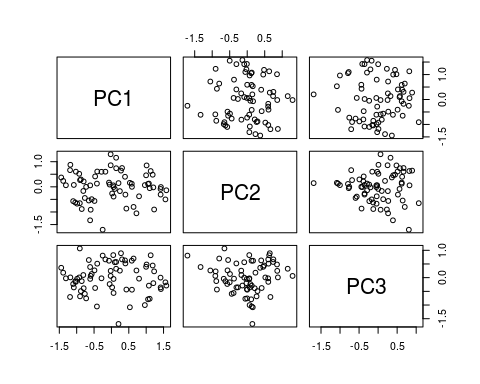
median\_normalize <- function(x) (x - median(x)) / IQR(x)  
features$normalized\_percent <- as.data.frame(apply(features$percent, MARGIN = 2, median\_normalize))  
  
features$normalized\_percent %>% gather() %>% ggplot(aes(x = value)) + facet\_grid(.~key, scales = "free") + geom\_histogram(bins = 20)



# look again   
pairs(features$normalized\_percent)



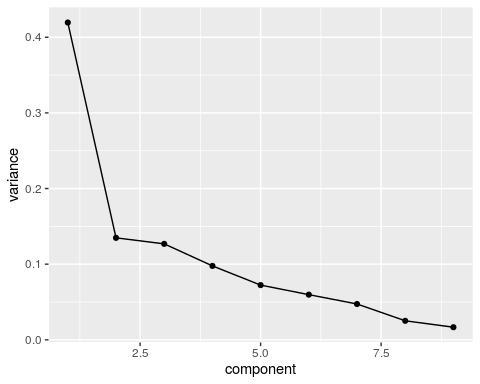
# build PCA  
  
features$percentPCA <- prcomp(features$normalized\_percent)  
pairs(as.data.frame(features$percentPCA$x))



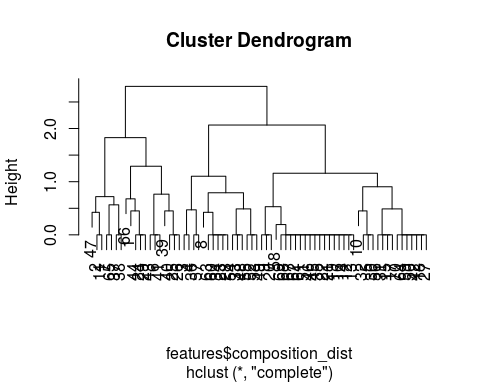
data.frame(components = 1:length(features$percentPCA$sdev),   
 variance = (features$percentPCA$sdev) ^ 2 / sum((features$percentPCA$sdev) ^ 2))

## components variance  
## 1 1 0.5261920  
## 2 2 0.2520919  
## 3 3 0.2217162

phi\_svd <- svd(phi\_matrix)  
  
variance <- data.frame(component = 1:length(phi\_svd$d), variance = phi\_svd$d / sum(phi\_svd$d))  
variance %>% ggplot(aes(x = component, y = variance)) + geom\_point() + geom\_line()



features$scored\_composition <- as.data.frame(as.matrix(features$composition) %\*% phi\_svd$v)  
  
  
features$composition\_dist <- dist(features$scored\_composition[c("V1", "V2", "V3")])  
  
h\_tree <- hclust(features$composition\_dist)  
plot(h\_tree)



features$scored\_composition$hclass <- as.factor(cutree(h\_tree, k = 3))  
  
features$scored\_composition$kmclass <- as.factor(kmeans(features$scored\_composition[c("V1", "V2", "V3")], iter.max = 20, centers = 3)$cluster)  
  
table(features$scored\_composition$hclass)

##   
## 1 2 3   
## 19 16 35

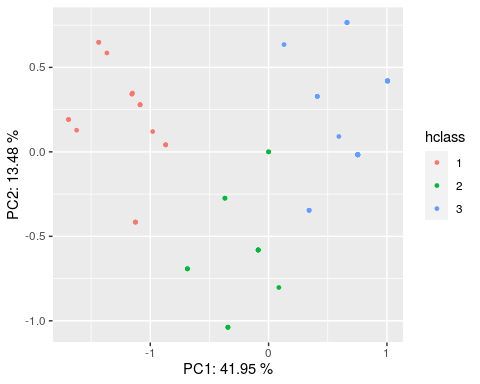
table(features$scored\_composition$kmclass)

##   
## 1 2 3   
## 37 19 14

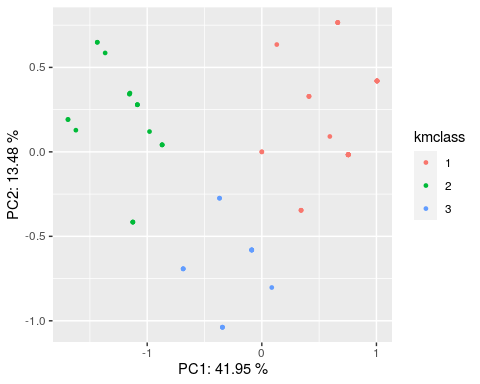
table(candies$Y)

##   
## 0 1   
## 33 37

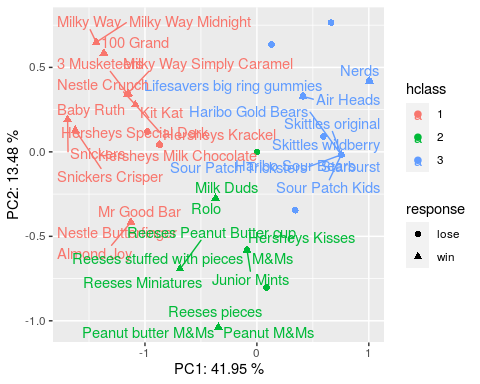
features$scored\_composition %>% ggplot(aes(x = V1, y = V2, color = hclass)) + geom\_point(size = 1) + xlab(paste("PC1:", round(variance[1,2] \* 100, 2), "%")) + ylab(paste("PC2:", round(variance[2,2] \* 100, 2), "%"))



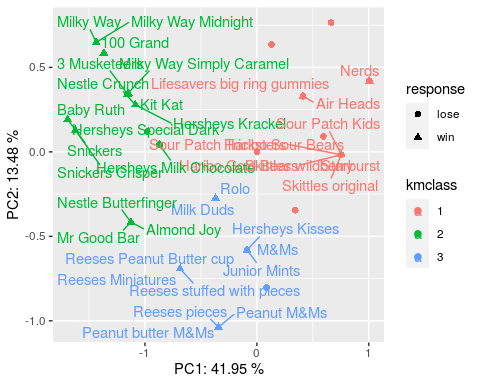
features$scored\_composition %>% ggplot(aes(x = V1, y = V2, color = kmclass)) + geom\_point(size = 1) + xlab(paste("PC1:", round(variance[1,2] \* 100, 2), "%")) + ylab(paste("PC2:", round(variance[2,2] \* 100, 2), "%"))



features$scored\_composition$response <- as.factor(candies$Y)  
levels(features$scored\_composition$response) <- c("lose", "win")  
  
features$scored\_composition$competitors <- candies$competitorname  
  
hclass\_plot <- features$scored\_composition %>% ggplot(aes(x = V1, y = V2, color = hclass, shape = response , label = competitors)) + geom\_point(size = 2) + xlab(paste("PC1:", round(variance[1,2] \* 100, 2), "%")) + ylab(paste("PC2:", round(variance[2,2] \* 100, 2), "%")) + geom\_text\_repel(data = dplyr::filter(features$scored\_composition, response == "win"))  
  
kmean\_plot <- features$scored\_composition %>% ggplot(aes(x = V1, y = V2, color = kmclass, shape = response, label = competitors)) + geom\_point(size = 2) + xlab(paste("PC1:", round(variance[1,2] \* 100, 2), "%")) + ylab(paste("PC2:", round(variance[2,2] \* 100, 2), "%")) + geom\_text\_repel(data = dplyr::filter(features$scored\_composition, response == "win"))  
  
  
hclass\_plot



kmean\_plot



km\_table <- table(features$scored\_composition[c("kmclass", "response")])  
hcl\_table <- table(features$scored\_composition[c("hclass", "response")])  
km\_table

## response  
## kmclass lose win  
## 1 27 10  
## 2 3 16  
## 3 3 11

hcl\_table

## response  
## hclass lose win  
## 1 3 16  
## 2 5 11  
## 3 25 10

chisq.test(km\_table)

##   
## Pearson's Chi-squared test  
##   
## data: km\_table  
## X-squared = 21.117, df = 2, p-value = 2.597e-05

chisq.test(hcl\_table)

##   
## Pearson's Chi-squared test  
##   
## data: hcl\_table  
## X-squared = 17.402, df = 2, p-value = 0.0001665