Class 10: Haloween Candy Mini Project

AUTHOR
Toheeb Balogun

#Background

In this mini-project we will examine 538 Halloween Candy data. What is your favorite candy? What is nougat anyway? And how do you say it in America?

First step is to read the data

```
candy <- read.csv("candy-data.txt", row.names=1)
head(candy)
```

```
             chocolate fruity caramel peanutyalmondy nougat crispedricewafer
100 Grand            1      0       1              0      0                1
3 Musketeers         1      0       0              0      1                0
One dime             0      0       0              0      0                0
One quarter          0      0       0              0      0                0
Air Heads            0      1       0              0      0                0
Almond Joy           1      0       0              1      0                0
             hard bar pluribus sugarpercent pricepercent winpercent
100 Grand       0   1        0        0.732        0.860   66.97173
3 Musketeers    0   1        0        0.604        0.511   67.60294
One dime        0   0        0        0.011        0.116   32.26109
One quarter     0   0        0        0.011        0.511   46.11650
Air Heads       0   0        0        0.906        0.511   52.34146
Almond Joy      0   1        0        0.465        0.767   50.34755
```

Q1. How many different candy types are in this dataset? Answer:85

```
nrow(candy)
```

```
[1] 85
```

Q2. How many fruity candy types are in the dataset? Answer: 38

```
sum(candy$fruity)
```

```
[1] 38
```

Q3. What is your favorite candy in the dataset and what is it's winpercent value? Answer: 39.46056

```
#rownames(candy)
candy["Dum Dums", ]$winpercent
```

```
[1] 39.46056
```

Q4: What is the winpercent value for “Kit Kat”? Answer: 76.7686

```
candy["Kit Kat", ]$winpercent
```

```
[1] 76.7686
```

Q5: What is the winpercent value for “Tootsie Roll Snack Bars”? Answer:49.6535

```
candy["Tootsie Roll Snack Bars", ]$winpercent
```

```
[1] 49.6535
```

```
library("skimr")
skim(candy)
```

Data summary

Name	candy
Number of rows	85
Number of columns	12
_____	
Column type frequency:	
numeric	12
_____	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
chocolate	0	1	0.44	0.50	0.00	0.00	0.00	1.00	1.00	
fruity	0	1	0.45	0.50	0.00	0.00	0.00	1.00	1.00	
caramel	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
peanutyalmondy	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
nougat	0	1	0.08	0.28	0.00	0.00	0.00	0.00	1.00	
crispedricewafer	0	1	0.08	0.28	0.00	0.00	0.00	0.00	1.00	
hard	0	1	0.18	0.38	0.00	0.00	0.00	0.00	1.00	
bar	0	1	0.25	0.43	0.00	0.00	0.00	0.00	1.00	
pluribus	0	1	0.52	0.50	0.00	0.00	1.00	1.00	1.00	
sugarpercent	0	1	0.48	0.28	0.01	0.22	0.47	0.73	0.99	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
pricepercent	0	1	0.47	0.29	0.01	0.26	0.47	0.65	0.98	
winpercent	0	1	50.32	14.71	22.45	39.14	47.83	59.86	84.18	

Q6. Is there any variable/column that looks to be on a different scale to the majority of the other columns in the dataset? Answer: Winpercent

Q7: Q7. What do you think a zero and one represent for the candy$chocolate column? Answer: Zero means FALSE while one means TRUE

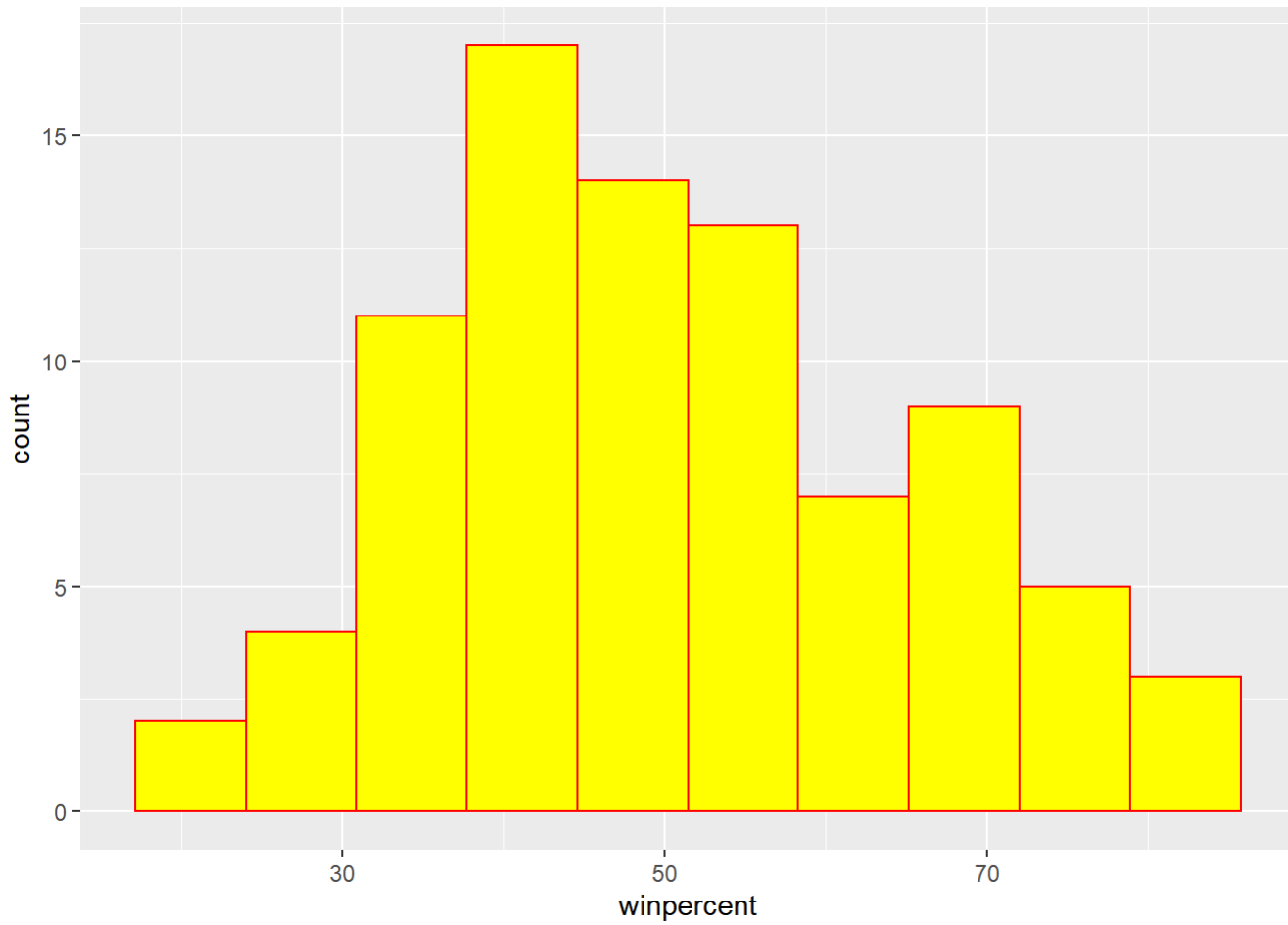
```
candy$chocolate
```

```
 [1] 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 0 0 1 1 0 1 1 1
[39] 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1
[77] 1 1 0 1 0 0 0 0 1
```

Q8. Plot a histogram of winpercent values

```
library(ggplot2)

ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins=10, col="red", fill="yellow")
```



Q9. Is the distribution of winpercent values symmetrical? Answer: The distribution of winpercent is not symmetric #The centre is the highest point Q10. Is the center of the distribution above or below 50%? Answer:The centre of distribution is below 50% Q11. On average is chocolate candy higher or lower ranked than fruit candy? Answer:The chocolate candy is higher ranked than fruit candy

```
chocolate.inds <- as.logical(candy$chocolate)
chocolate.win <- candy[chocolate.inds,]$winpercent
mean(chocolate.win)
```

```
[1] 60.92153
```

And for fruit candy

```
fruiti.inds <-  as.logical(candy$fruity)
fruiti.win <- candy[fruiti.inds,]$winpercent
mean(fruiti.win)
```

```
[1] 44.11974
```

Q12. Is this difference statistically significant? Answer: There is significant different which means chocolate is better than fruiti

```
t.test(chocolate.win, fruiti.win)
```

```
t.test(chocolate.win, fruiti.win)
```

```
	Welch Two Sample t-test

data:  chocolate.win and fruiti.win
t = 6.2582, df = 68.882, p-value = 2.871e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 11.44563 22.15795
sample estimates:
mean of x mean of y 
 60.92153  44.11974 
```

Overall candy rating

The base R `sort()` and `order` functions are very useful

```
x <- c(5,1,2,6)
sort(x, decreasing = T)
```

```
[1] 6 5 2 1
```

```
x[order(x)]
```

```
[1] 1 2 5 6
```

```
y <- c("berry", "alice", "chandra")
y
```

```
[1] "berry"   "alice"   "chandra"
```

```
sort(y)
```

```
[1] "alice"   "berry"   "chandra"
```

```
order(y)
```

```
[1] 2 1 3
```

Q13. What are the five least liked candy types in this set? First, I want to order/manage the whole dataset by winpercent values

```
inds <- order(candy$winpercent)
head (candy[inds,], n=5)
```

```
                    chocolate fruity caramel peanutyalmondy nougat
Nik L Nip                   0      1       0               0      0
```

```
Nik L Nip                  0       1       0               0       0
Boston Baked Beans         0       0       0               1       0
Chiclets                   0       1       0               0       0
Super Bubble               0       1       0               0       0
Jawbusters                 0       1       0               0       0
                   crispedricewafer hard bar pluribus sugarpercent pricepercent
Nik L Nip                         0    0   0        1        0.197        0.976
Boston Baked Beans                0    0   0        1        0.313        0.511
Chiclets                          0    0   0        1        0.046        0.325
Super Bubble                      0    0   0        0        0.162        0.116
Jawbusters                        0    1   0        1        0.093        0.511
                   winpercent
Nik L Nip            22.44534
Boston Baked Beans   23.41782
Chiclets             24.52499
Super Bubble         27.30386
Jawbusters           28.12744
```

```
#candy[inds,]
```

```
#head(candy[order(candy$winpercent),], n=5)
```

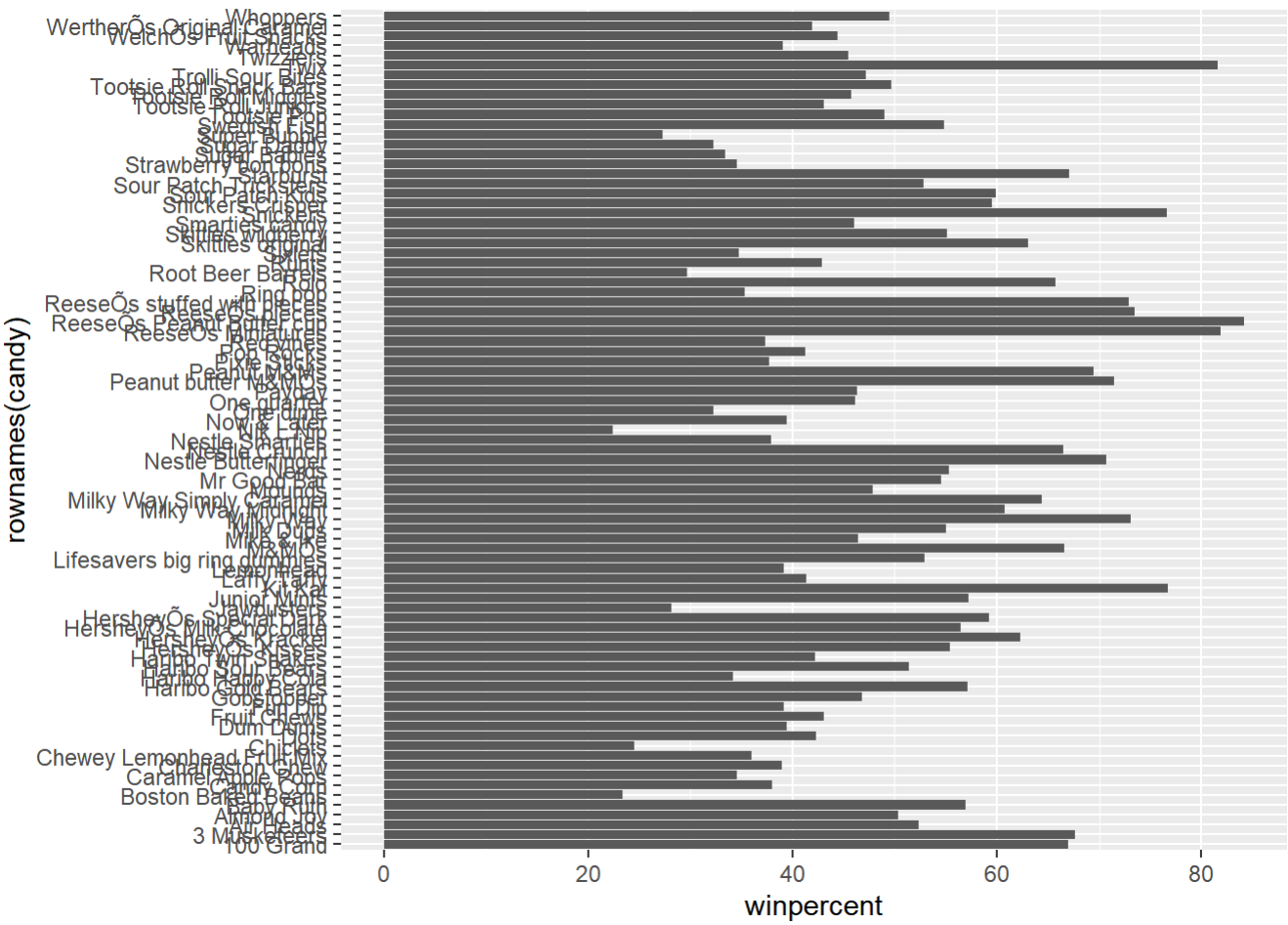
Q14. What are the top 5 all time favorite candy types out of this set?

```
inds <- order(candy$winpercent, decreasing = T)
head (candy[inds,], n=5)
```

```
                         chocolate fruity caramel peanutyalmondy nougat
ReeseÕs Peanut Butter cup         1      0       0              1      0
ReeseÕs Miniatures                1      0       0              1      0
Twix                              1      0       1              0      0
Kit Kat                           1      0       0              0      0
Snickers                          1      0       1              1      1
                         crispedricewafer hard bar pluribus sugarpercent
ReeseÕs Peanut Butter cup                0    0   0        0        0.720
ReeseÕs Miniatures                       0    0   0        0        0.034
Twix                                     1    0   1        0        0.546
Kit Kat                                  1    0   1        0        0.313
Snickers                                 0    0   1        0        0.546
                         pricepercent winpercent
ReeseÕs Peanut Butter cup        0.651   84.18029
ReeseÕs Miniatures               0.279   81.86626
Twix                             0.906   81.64291
Kit Kat                          0.511   76.76860
Snickers                         0.651   76.67378
```

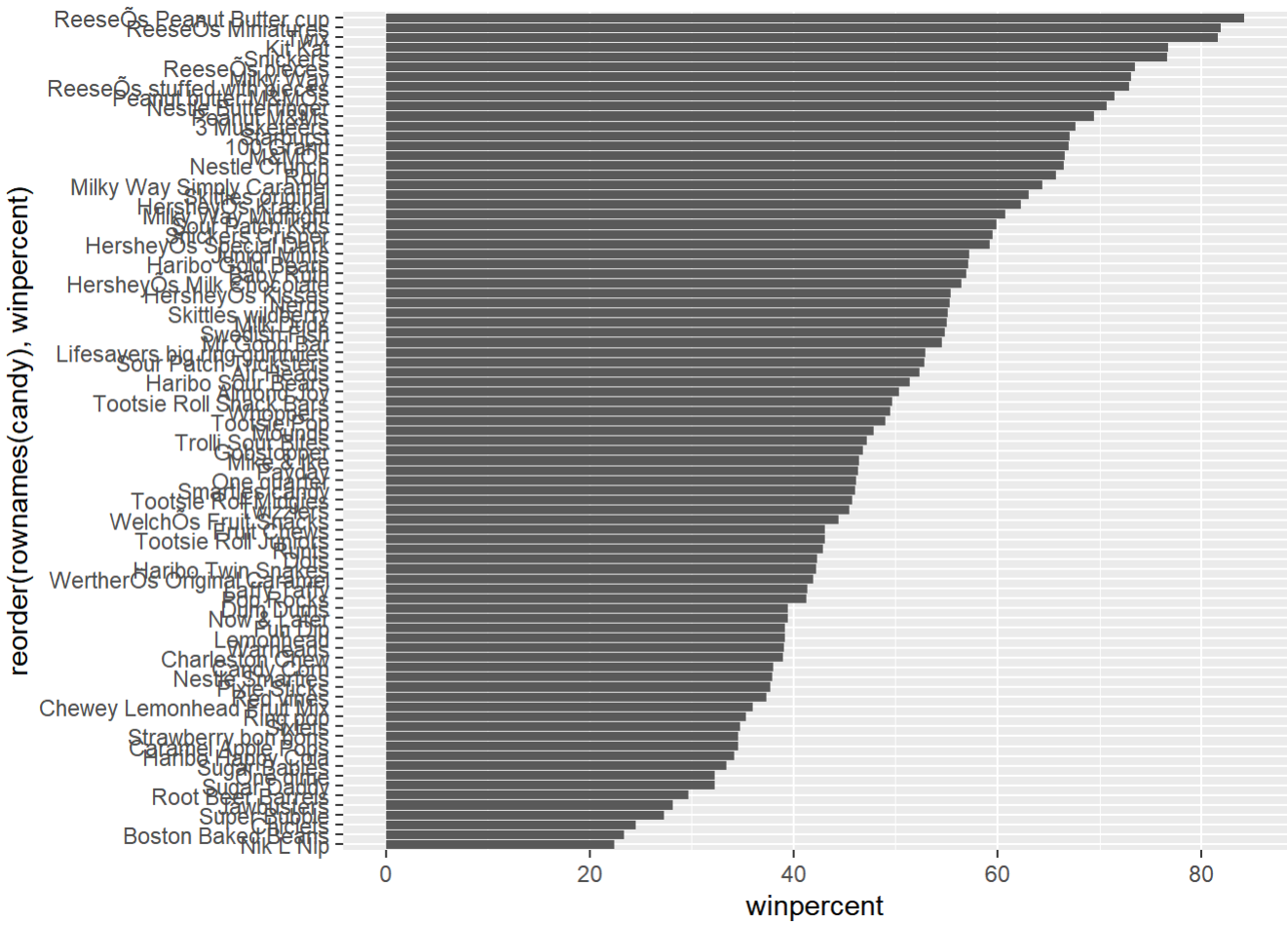
#Barplot The default barplot, made with `geom_col` has the bars in the order they are in the dataset Q15. Make a first barplot of candy ranking based on winpercent values.

```
#library(ggplot2)
ggplot(candy) +
  aes(winpercent, rownames(candy)) +
  geom_col()
```



Q16. This is quite ugly, use the reorder() function to get the bars sorted by winpercent?

```
ggplot(candy) +
    aes(winpercent, reorder( rownames(candy), winpercent)) +
    geom_col()
```



How to make high quality plots for publication

```
ggsave("mybarplot.png")
```

```
Saving 7 x 5 in image
```

Time to add some useful color Let's setup a color vector (that signifies candy type) that we can then use for some future plots. We start by making a vector of all black values (one for each candy). Then we overwrite chocolate (for chocolate candy), brown (for candy bars) and red (for fruity candy) values.

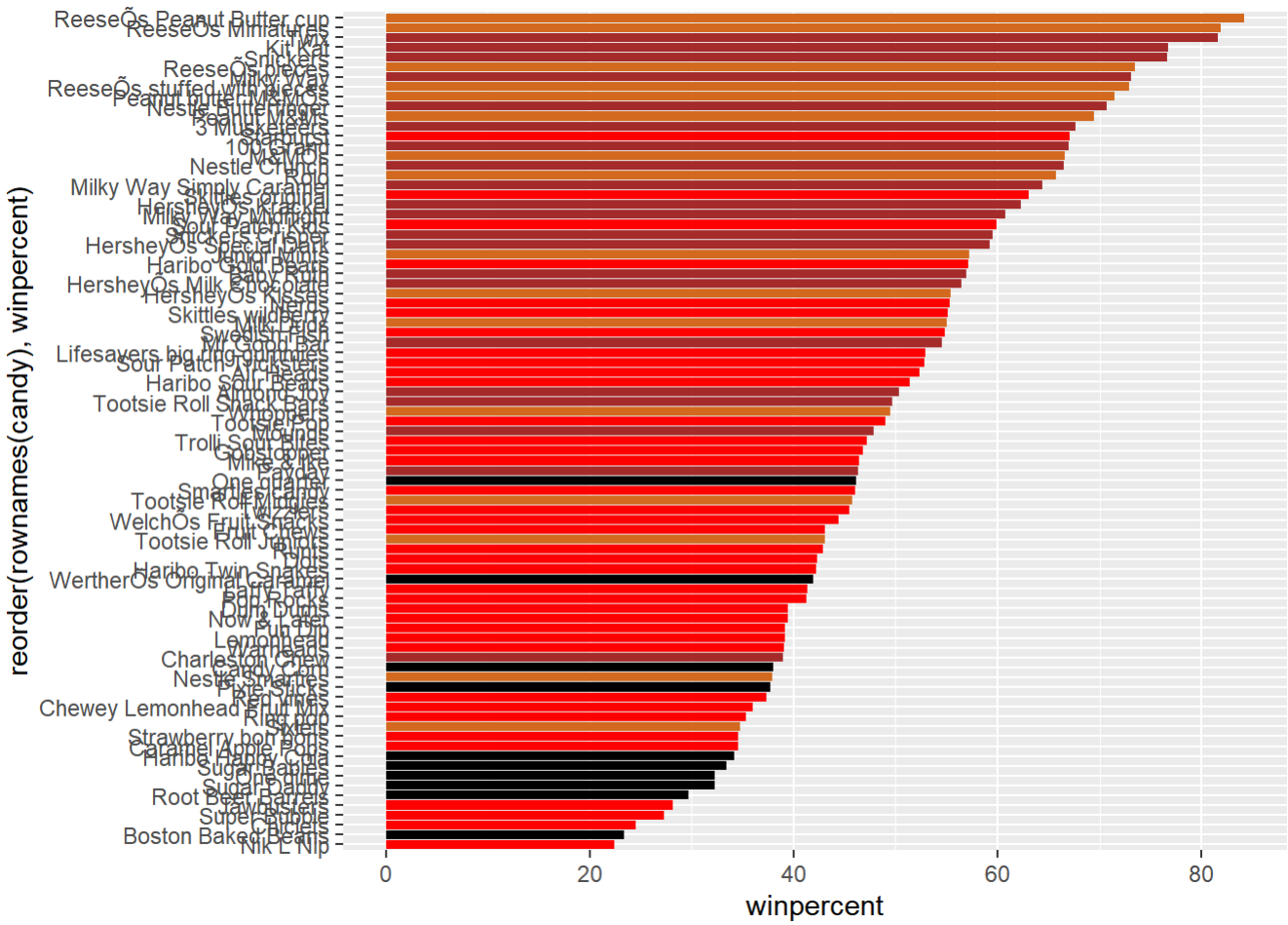
```
my_cols <- rep("black", nrow(candy))
#my_cols
my_cols [ as.logical(candy$chocolate)] <- "chocolate"
my_cols [ as.logical(candy$bar)] <- "brown"
my_cols [ as.logical(candy$fruity)] <- "red"
my_cols
```

```
 [1] "brown"     "brown"     "black"     "black"     "red"       "brown"
 [7] "brown"     "black"     "black"     "red"       "brown"     "red"
[13] "red"       "red"       "red"       "red"       "red"       "red"
[19] "red"       "black"     "red"       "red"       "chocolate" "brown"
[25] "brown"     "brown"     "red"       "chocolate" "brown"     "red"
[31] "red"       "red"       "chocolate" "chocolate" "red"       "chocolate"
```

```
[37] "brown"     "brown"     "brown"     "brown"     "brown"     "red"
[43] "brown"     "brown"     "red"       "red"       "brown"     "chocolate"
[49] "black"     "red"       "red"       "chocolate" "chocolate" "chocolate"
[55] "chocolate" "red"       "chocolate" "black"     "red"       "chocolate"
[61] "red"       "red"       "chocolate" "red"       "brown"     "brown"
[67] "red"       "red"       "red"       "red"       "black"     "black"
[73] "red"       "red"       "red"       "chocolate" "chocolate" "brown"
[79] "red"       "brown"     "red"       "red"       "red"       "black"
[85] "chocolate"
```

Now I can use this vector to color up my plot

```
ggplot(candy) +
    aes(winpercent, reorder( rownames(candy), winpercent)) +
    geom_col(fill=my_cols)
```



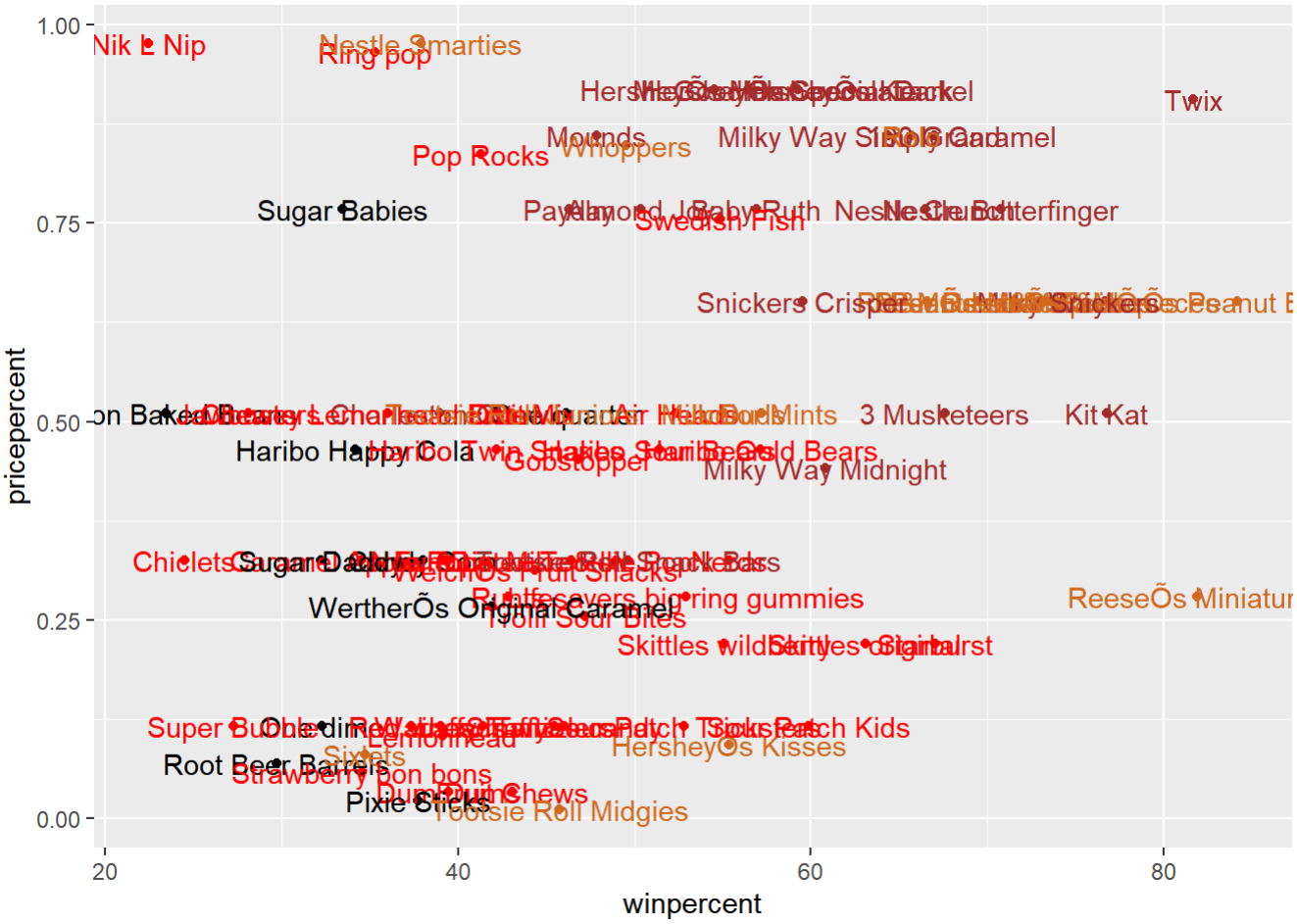
#4. Taking a look at pricepercent

What about value for money What is the candy for the least money?

One way to get this would be to make a plot of `winpercent` vs the `pricepercent` values

```
ggplot(candy) +
        aes(winpercent, pricepercent, label=rownames(candy)) +
```

```
  geom_point(col=my_cols) +
  geom_text(col=my_cols)
```

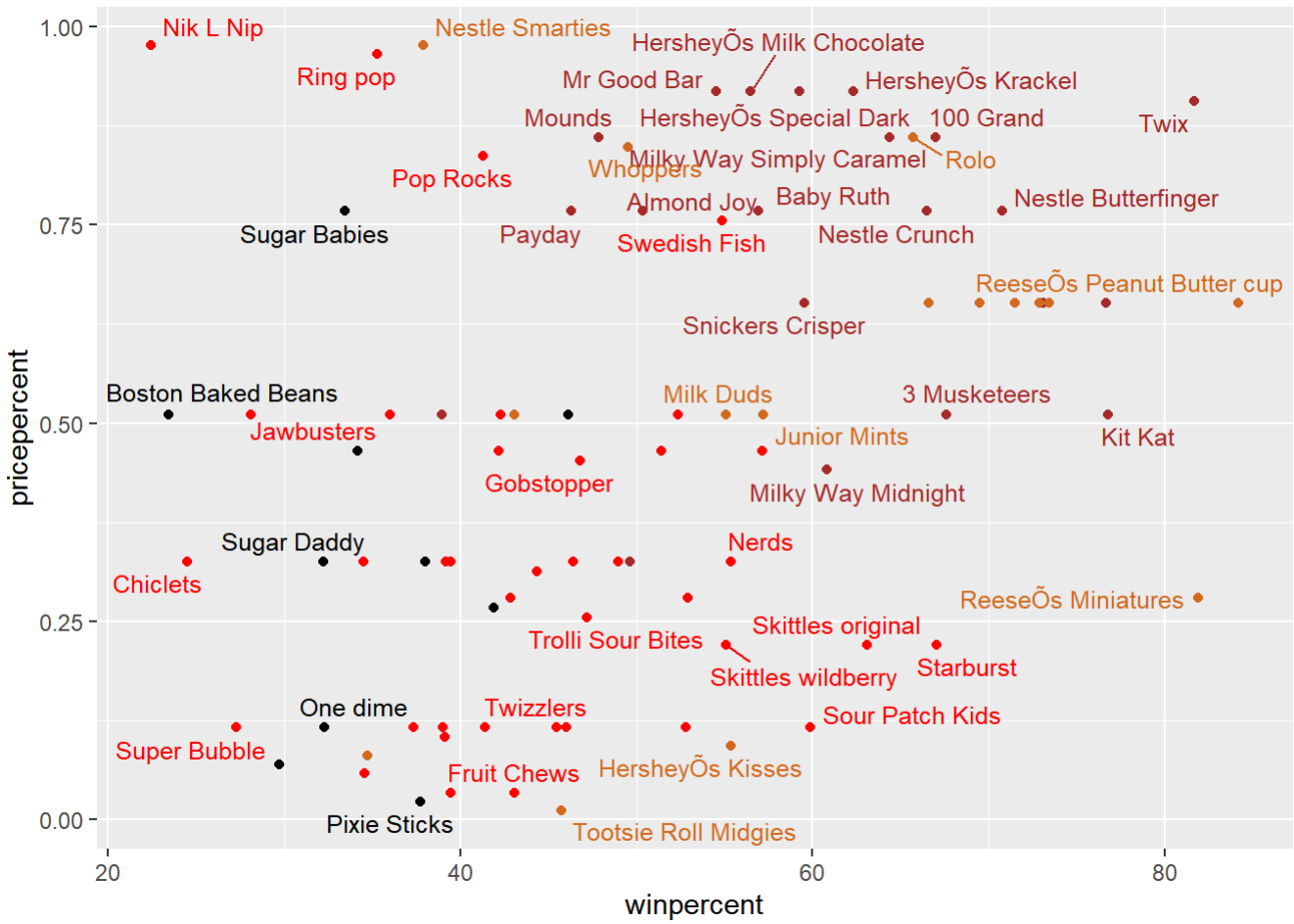


This plot sucks! I cannot read the labels.... We can use ggrepl package to help with this

```
library(ggrepel)

# How about a plot of price vs win
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols, size=3.3, max.overlaps = 7)
```

```
Warning: ggrepel: 38 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

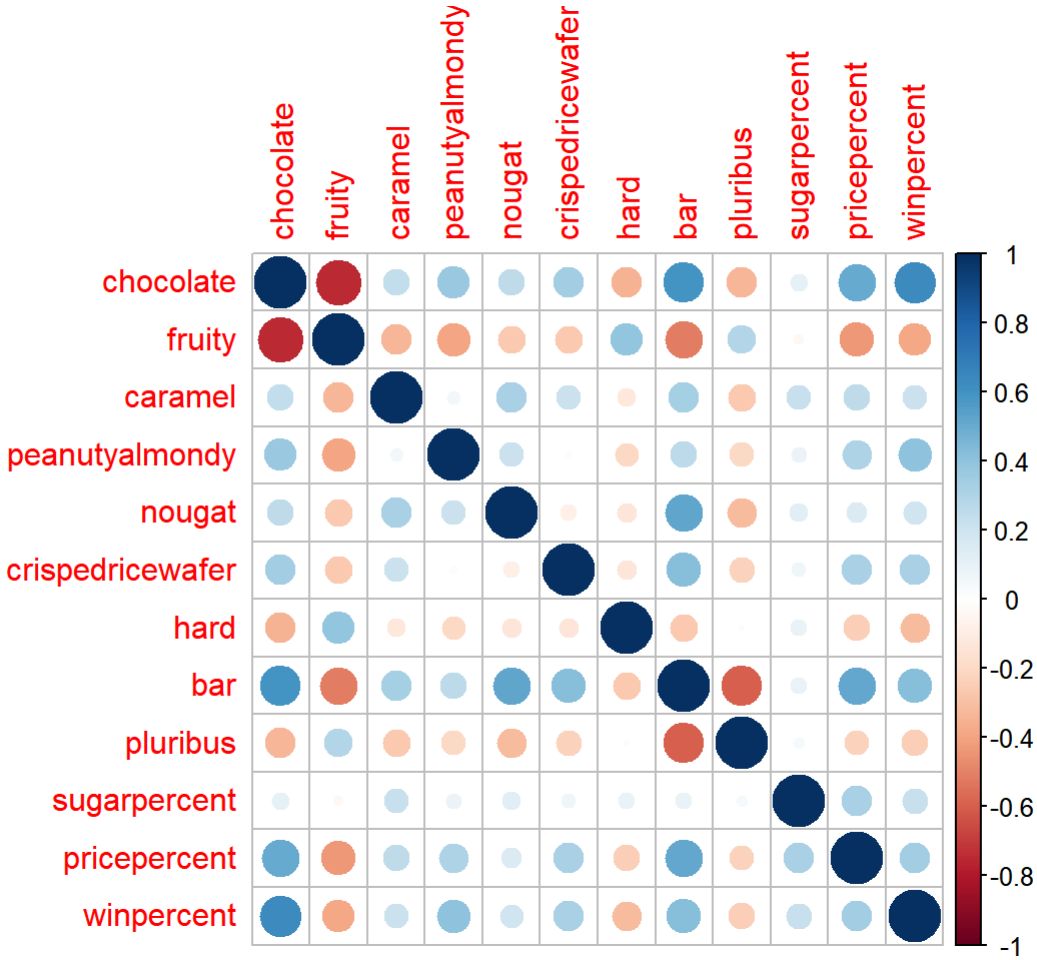


#5 Exploring the correlation structure

```
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
## corrplot 0.90 loaded
cij <- cor(candy)
corrplot(cij)
```



PCA Principal Component Ananlysis

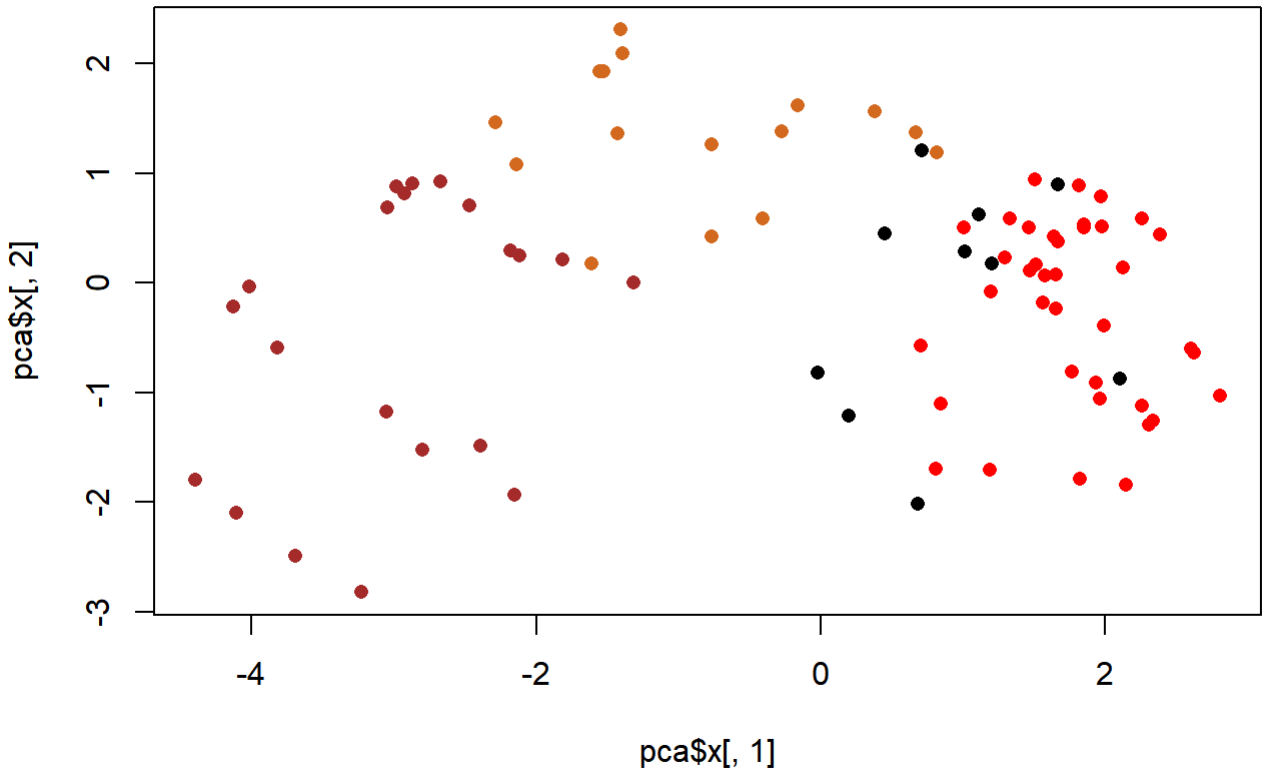
The main function that always there for us is `prcomp`. It has an important argument that is set to `scale=FALSE`

```
pca <- prcomp(candy, scale=TRUE)
summary(pca)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7
Standard deviation     2.0788 1.1378 1.1092 1.07533 0.9518 0.81923 0.81530
Proportion of Variance 0.3601 0.1079 0.1025 0.09636 0.0755 0.05593 0.05539
Cumulative Proportion  0.3601 0.4680 0.5705 0.66688 0.7424 0.79830 0.85369
                           PC8     PC9    PC10    PC11    PC12
Standard deviation     0.74530 0.67824 0.62349 0.43974 0.39760
Proportion of Variance 0.04629 0.03833 0.03239 0.01611 0.01317
Cumulative Proportion  0.89998 0.93832 0.97071 0.98683 1.00000
```

My PCA plot (a.k.a) PC1 vs PC2 score plot

```
plot(pca$x[,1], pca$x[,2], col=my_cols, pch=16)
```

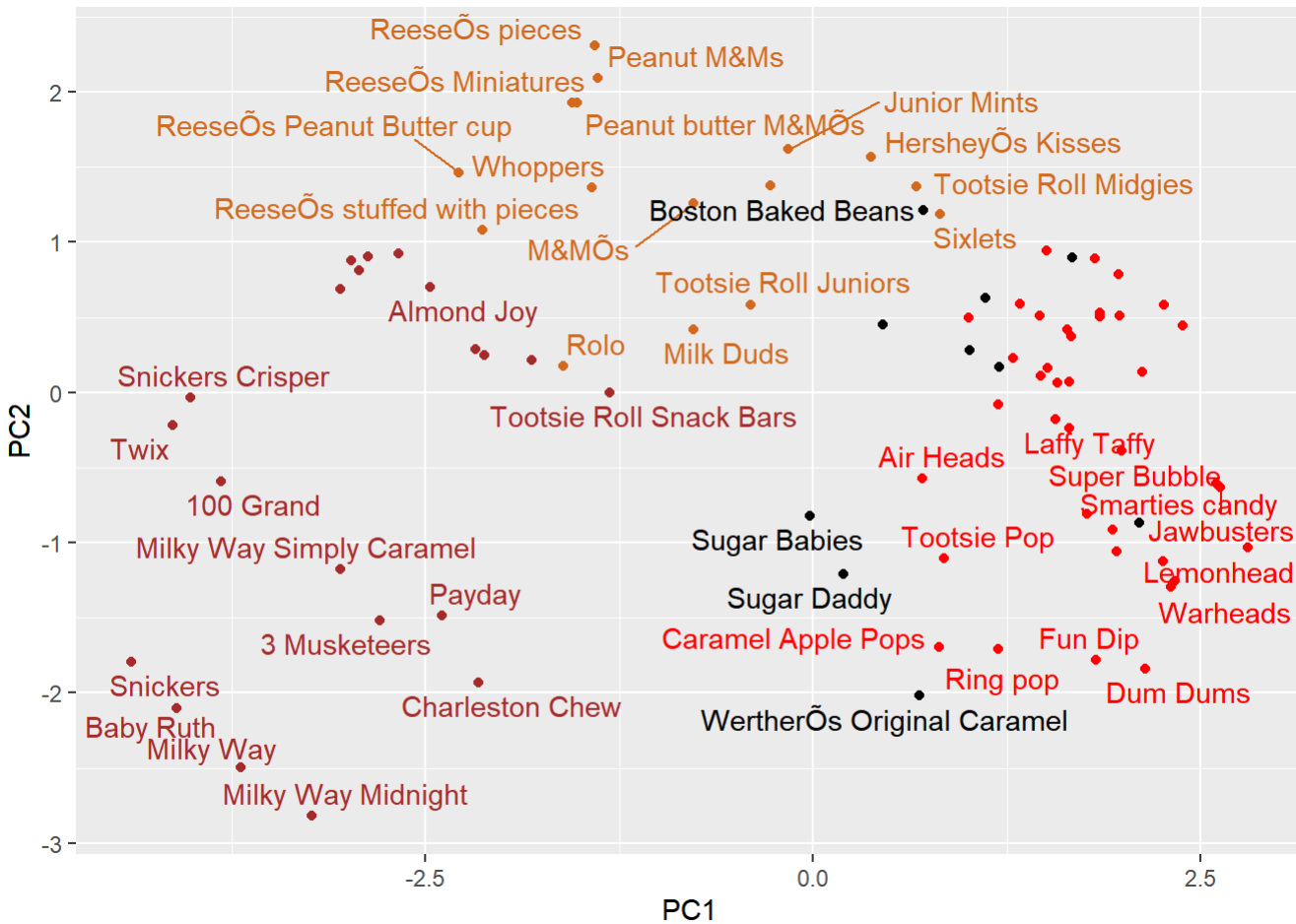


I will make a “nicer” plot with ggplot. ggplot only works with data.frames as input so I need to make one for it first

```
# Make a new data-frame with our PCA results and candy data
my_data <- cbind(candy, pca$x[,1:3])
```

```
p <- ggplot(my_data) +
  aes(PC1, PC2, label=rownames(my_data)) +
  geom_point(col=my_cols) +
geom_text_repel(col=my_cols, max.overlaps = 7)
p
```

```
Warning: ggrepel: 41 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




```
library(plotly)
```

```
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
```

```
ggplotly(p)
```

```
Warning in geom2trace.default(dots[[1L]][[1L]], dots[[2L]][[1L]], dots[[3L]][[1L]]):
geom_GeomTextRepel() has yet to be implemented in plotly.
  If you'd like to see this geom implemented,
```

```
Please open an issue with your example code at
https://github.com/ropensci/plotly/issues
```