Class 09 Halloween Mini project

Ani A16647613

Here we are going to analyze a candy dataset from the 538 website, which is a CSV file

Importing candy data

```
candy_file <- "candy-data.txt"
candy = read.csv(candy_file, 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?

```
dim(candy)
```

```
[1] 85 12
```

85 different candy types

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

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

```
[1] 38
```

Exploring the Dataset

Q3. What is your favorite candy in the dataset and what is its `winpercent` value?

```
candy["Reese's Peanut Butter cup",]$winpercent
```

```
[1] 84.18029
```

Q4. What is the `winpercent` value for KitKat?

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

```
[1] 76.7686
```

Q5. What is the `winpercent` value for "Tootsie Roll Snack bars?

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

```
[1] 49.6535
```

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

Table 1: 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	
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	

Q. What is the least liked candy in the dataset- lowest winpercent

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

```
                  chocolate fruity caramel peanutyalmondy nougat
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
Root Beer Barrels         0      0       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
Root Beer Barrels                0    1   0        1        0.732        0.069
```

```
                   winpercent
Nik L Nip            22.44534
Boston Baked Beans   23.41782
Chiclets             24.52499
Super Bubble         27.30386
Jawbusters           28.12744
Root Beer Barrels    29.70369
```

```
skimr::skim(candy)
```

Table 3: 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	
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?

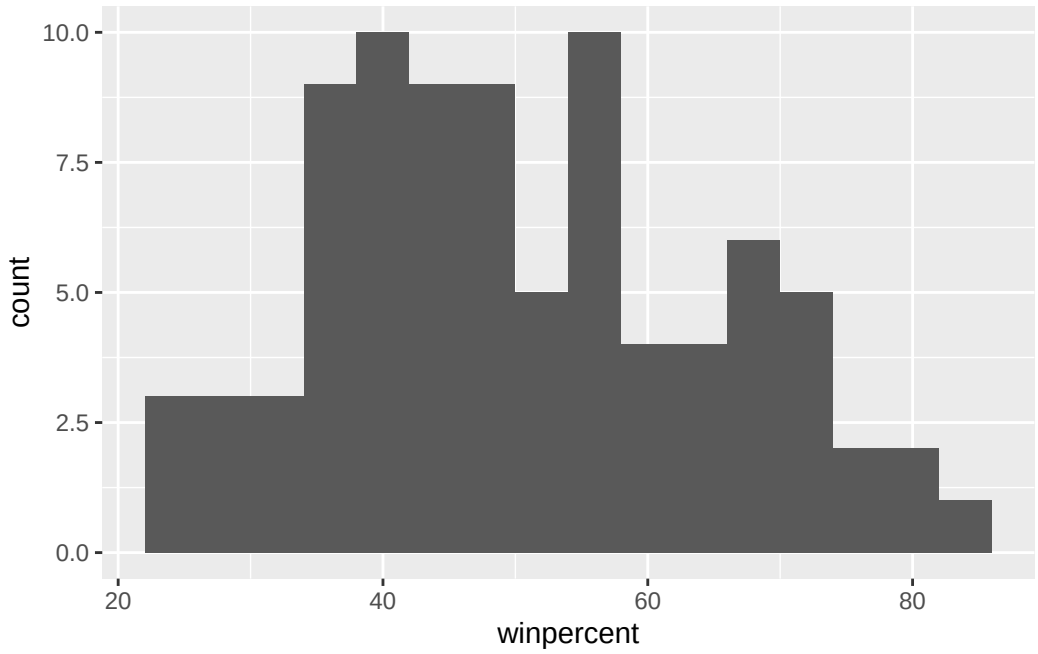
The variable `winpercent` has a different scale compared to the majority of the others.

Q7. What do you think a zero and one represent for the `candy$chocolate` column>

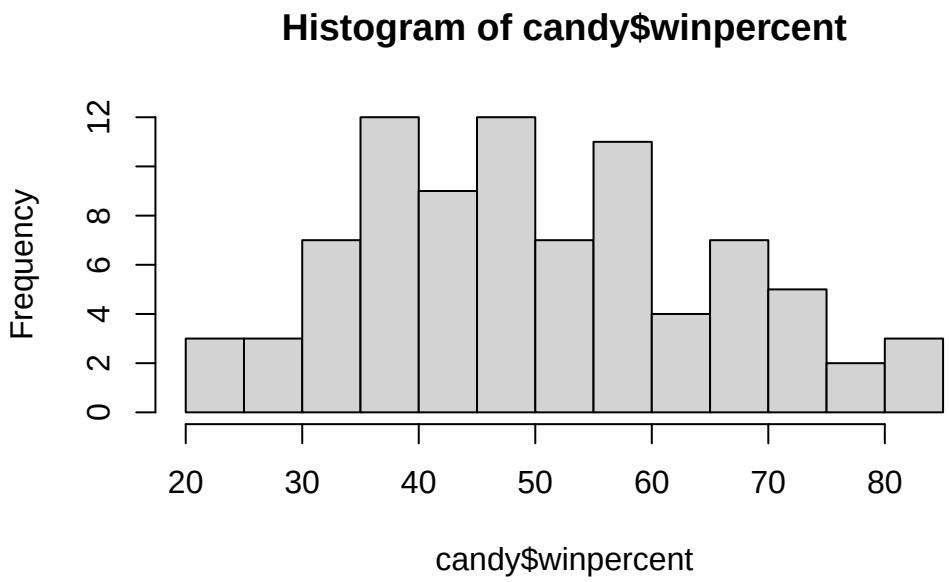
The zero represents candy that is not considered a part of the chocolate subset, one represents candy that is a part of the chocolate subset.

Q8. Plot a histogram of winpercent values.

```
library(ggplot2)
ggplot(candy, aes(winpercent)) + geom_histogram(binwidth=4)
```




```
hist(candy$winpercent, breaks=12)
```



Q9. Is the distribution of `winpercent` values symmetrical?

The distribution is not symmetrical

Q10. Is the center of the distribution above or below 50%?

The distribution is below 50%

Q11. On average is chocolate candy higher or lower ranked than fruit candy?

```
mean(candy$winpercent[as.logical(candy$chocolate)])
```

```
[1] 60.92153
```

```
mean(candy$winpercent[as.logical(candy$fruity)])
```

```
[1] 44.11974
```

or

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

```
[1] 60.92153
```

```
#candy$fruity == 1
fruit.inds <- as.logical(candy$fruity)
fruit.win <- candy[fruit.inds,]$winpercent
mean(fruit.win)
```

```
[1] 44.11974
```

Q12. Is the difference statistically significant?

```
t.test(choc.win, fruit.win)
```

```
    Welch Two Sample t-test

data:  choc.win and fruit.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
```

The difference is statistically significant (p-valuue: 2.87e-08).

Overall Candy Rankings

Q13. What are the five least liked candy types in this set?

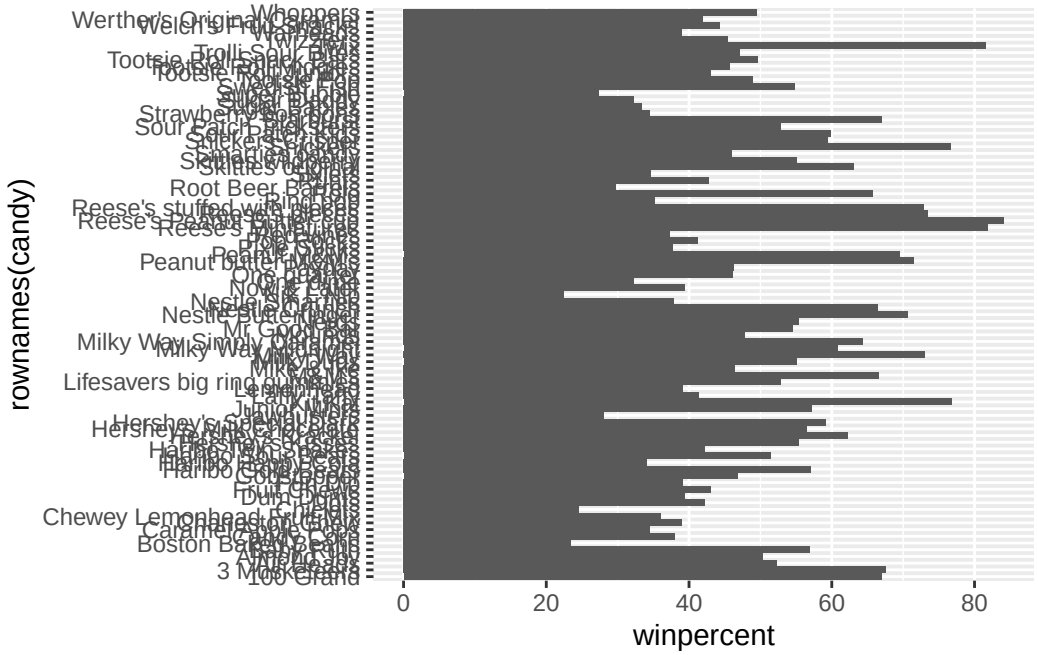
Nik L Nip, Boston Baked Beans, Chiclets, Super Bubble, Jawbusters

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

Reese's peanutbutter cups, reese's miniatures, twix, Kit Kat, Snickers

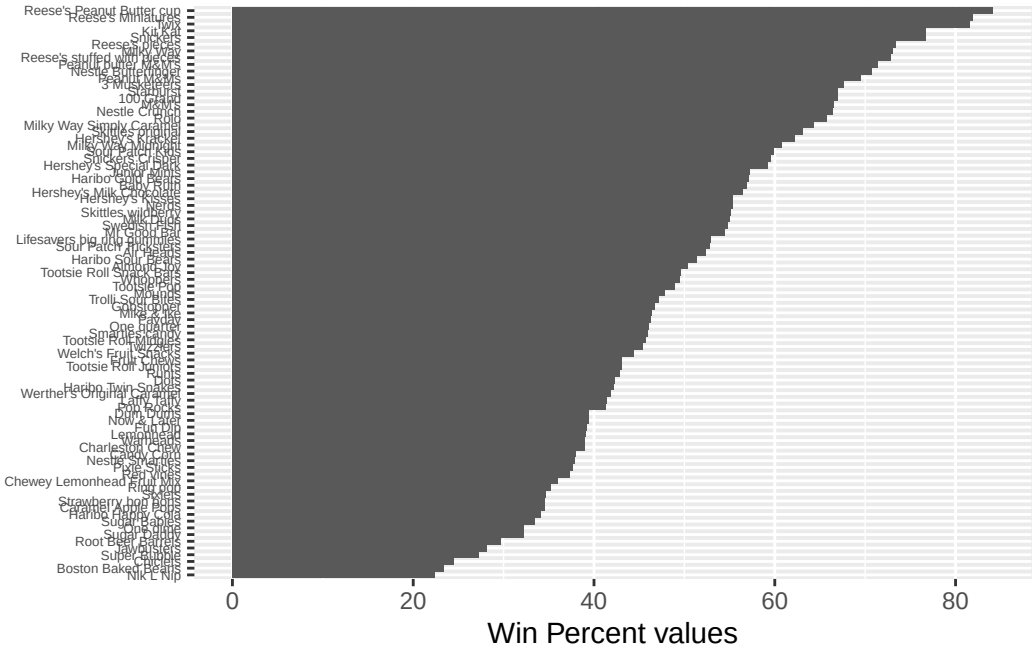
Q15. Make a first barplot of candy ranking based on `winpercent` values.

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



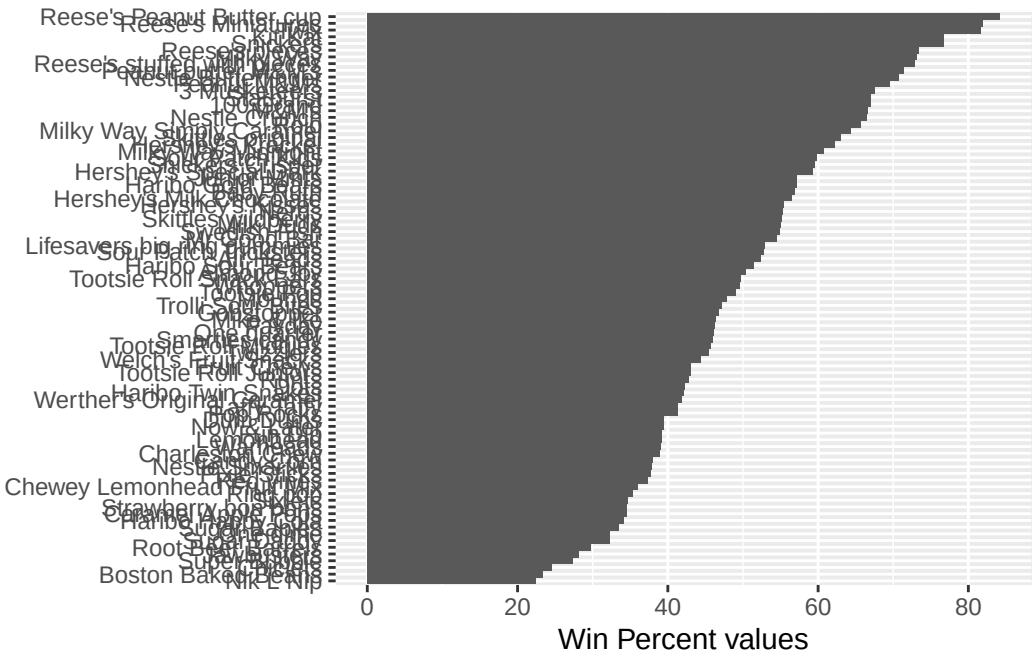
Q16. Use the reorder() function to get the bars sorted by `winpercent`.

```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent))  +
   labs(x= "Win Percent values", y=NULL) +
  geom_col() + theme(axis.text.y = element_text(size = 4.8))
```



or

```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent))  +
   labs(x= "Win Percent values", y=NULL) +
  geom_col()
```




```
ggsave("barplot1.png", width=7, height=10)
```

You can insert any image using this markdown syntax:

Add some color to our ggplot. We need to make a custom color vector:

```
#start with all black
my_cols <- rep("black", nrow(candy))
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"
```

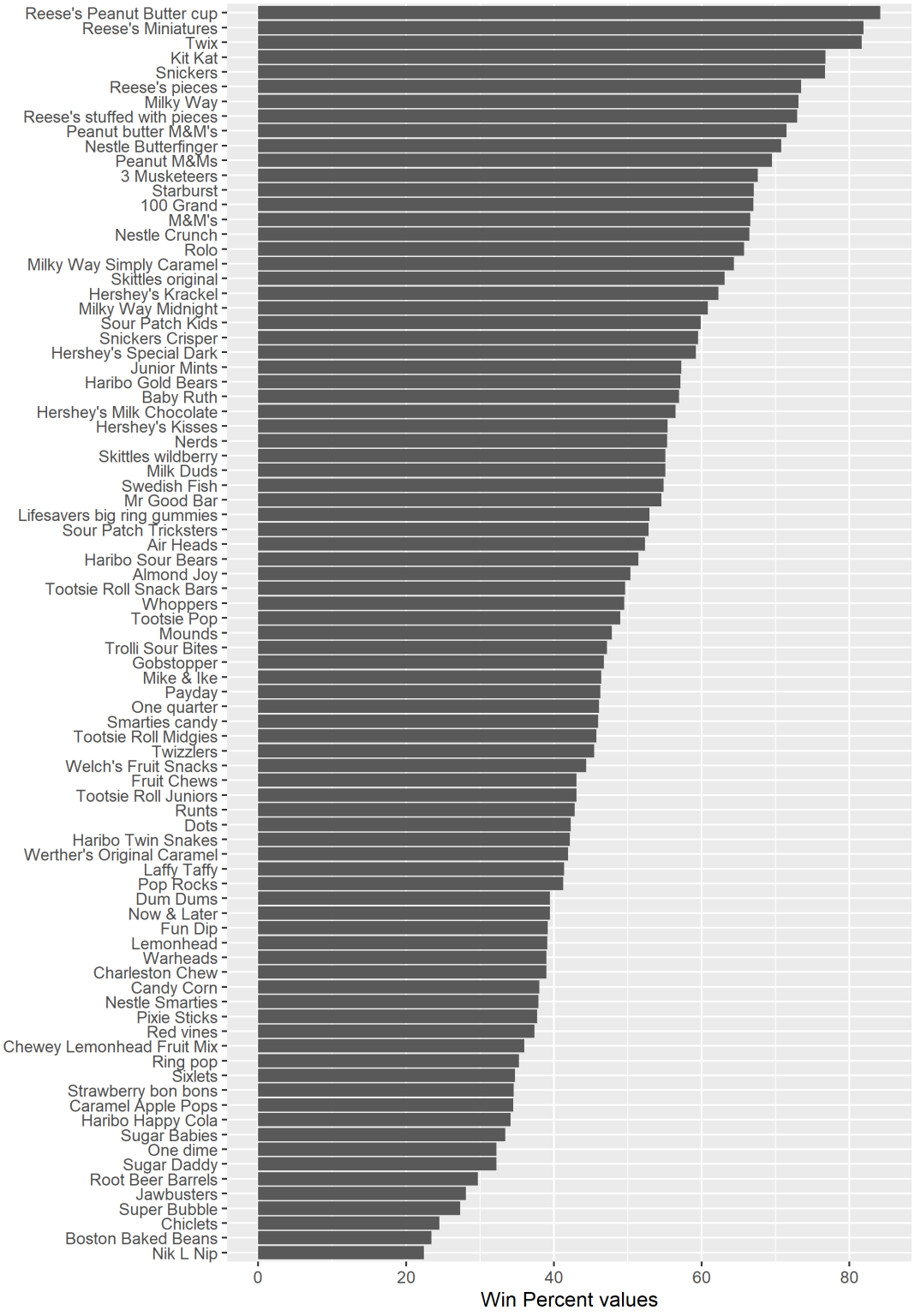
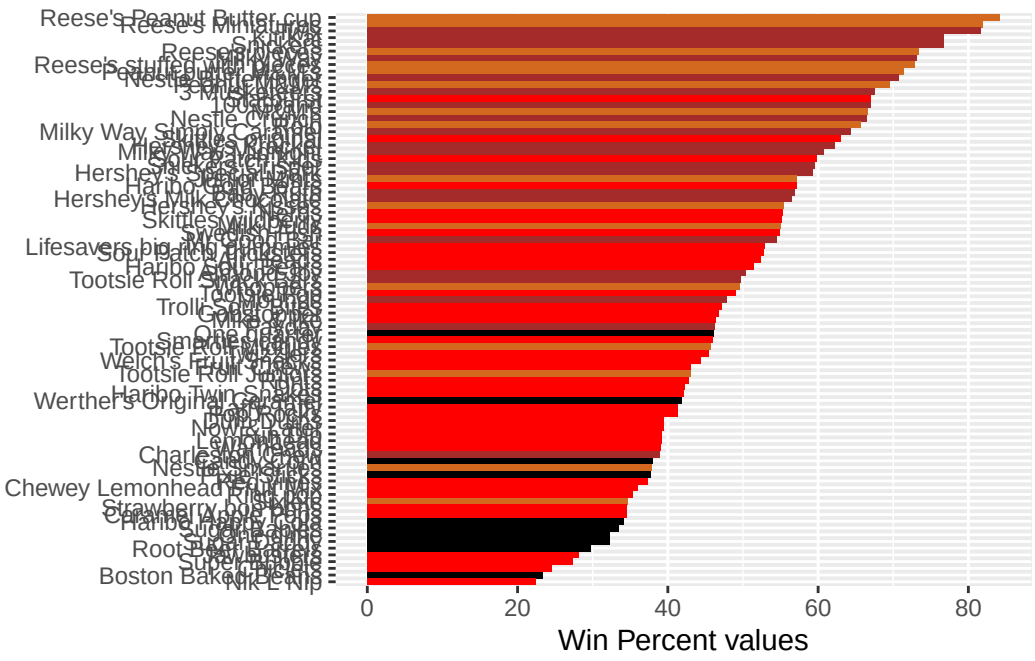


Figure 1: Candy Dataset

Figure 2: an example

```
[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"
```

```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent))  +
   labs(x= "Win Percent values", y=NULL) +
  geom_col(fill=my_cols)
```



Based on the plot:

Q17. What is the worst ranked chocolate candy?

Charleston Chew

Q18. What is the best ranked fruit candy?

Skittles

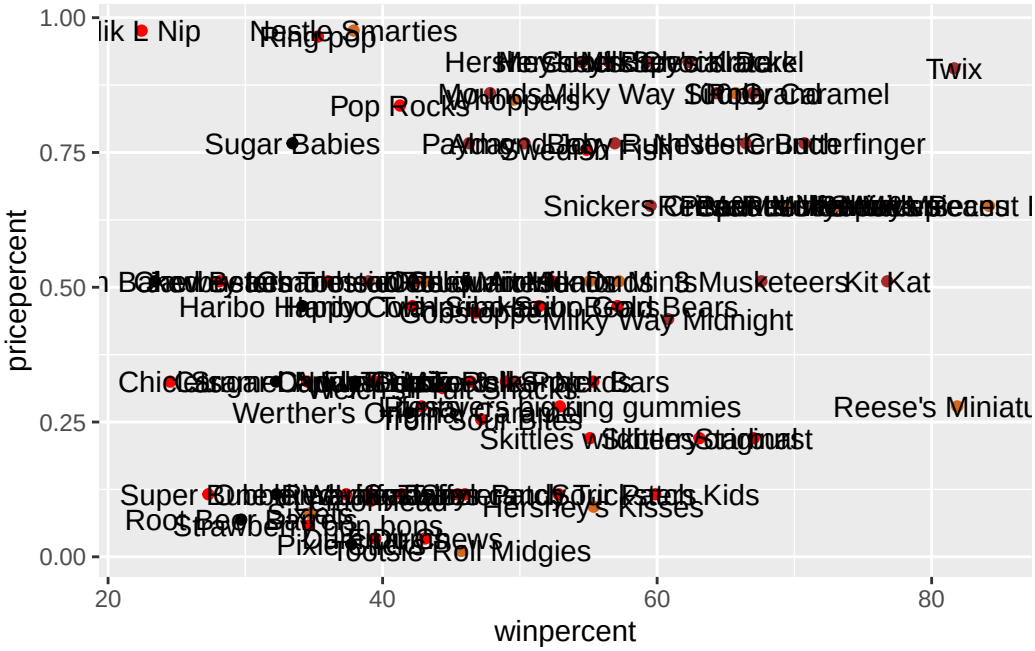
Takinf a look at pricepercent

```
candy$pricepercent
```

```
 [1] 0.860 0.511 0.116 0.511 0.511 0.767 0.767 0.511 0.325 0.325 0.511 0.511
[13] 0.325 0.511 0.034 0.034 0.325 0.453 0.465 0.465 0.465 0.465 0.093 0.918
[25] 0.918 0.918 0.511 0.511 0.511 0.116 0.104 0.279 0.651 0.651 0.325 0.511
[37] 0.651 0.441 0.860 0.860 0.918 0.325 0.767 0.767 0.976 0.325 0.767 0.651
[49] 0.023 0.837 0.116 0.279 0.651 0.651 0.651 0.965 0.860 0.069 0.279 0.081
[61] 0.220 0.220 0.976 0.116 0.651 0.651 0.116 0.116 0.220 0.058 0.767 0.325
[73] 0.116 0.755 0.325 0.511 0.011 0.325 0.255 0.906 0.116 0.116 0.313 0.267
[85] 0.848
```

If we want to see what is a good candy to buy in terms of winpercent and pricepercent we can plot these two variables and then see the best candy for the least amount of money

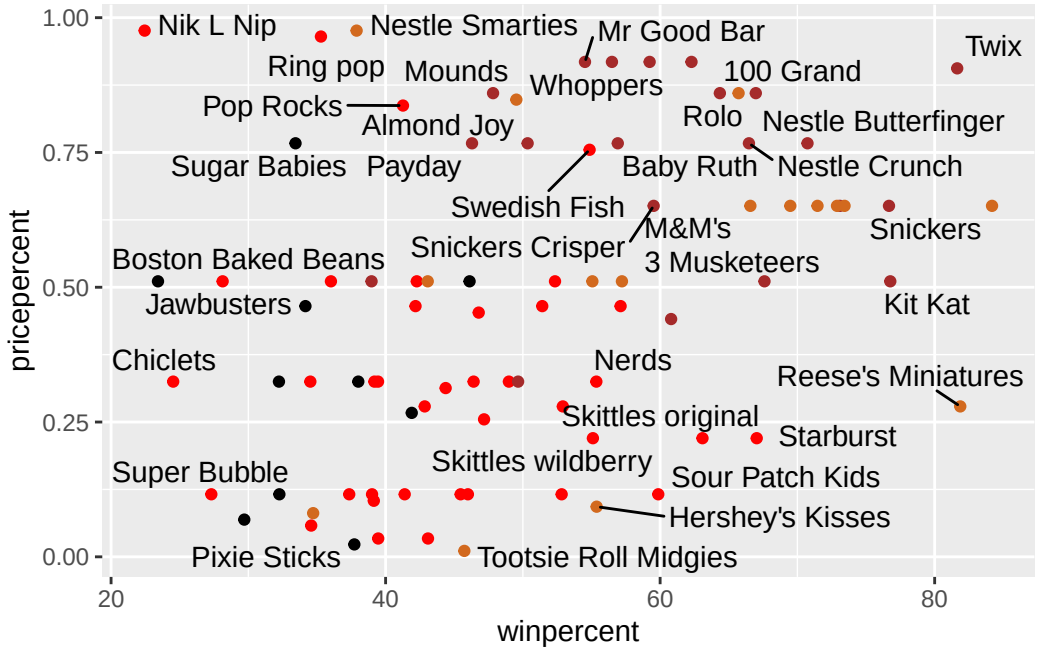
```
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text()
```



To avoid the overplotting of all the labels we can use an add on package called ggrepel

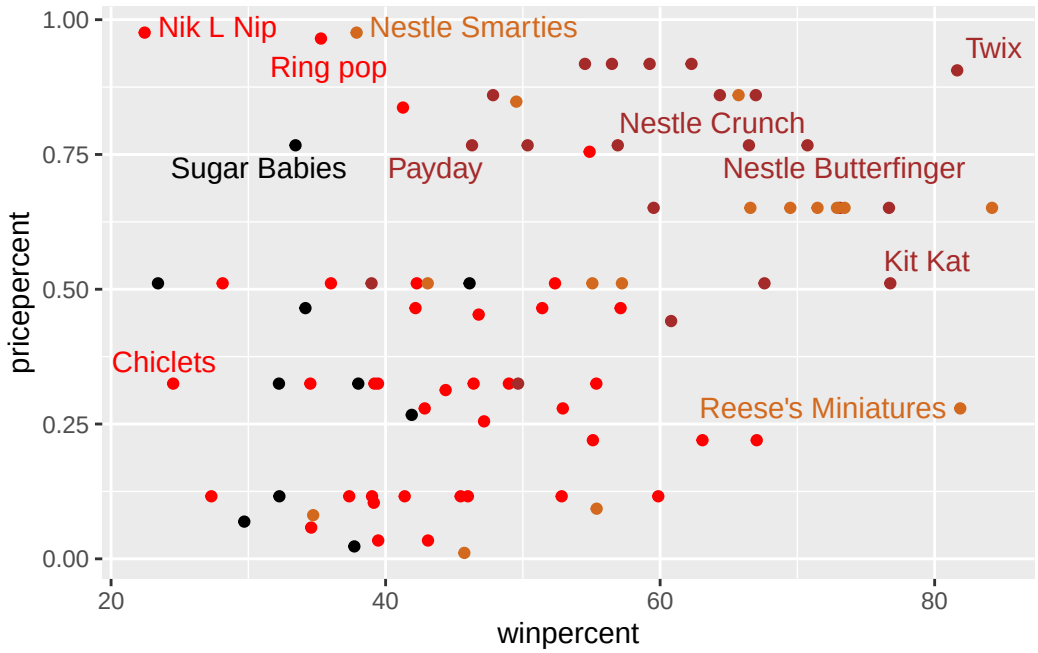
```
library(ggrepel)

ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel()
```

```
Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




```
# pink was too hard to see so changed fruity color to red
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(max.overlaps = 5, col=my_cols)
```

```
Warning: ggrepel: 74 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



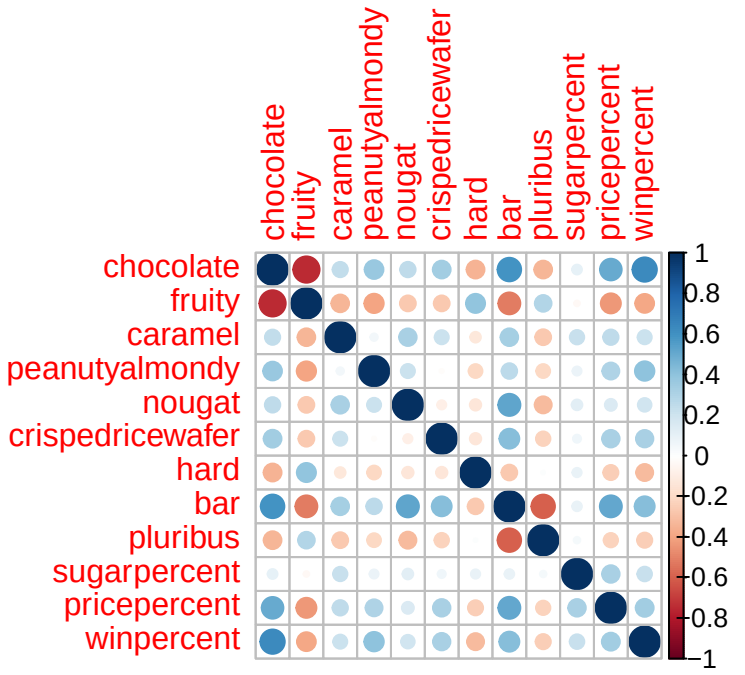
Q21. Make a barplot again with geom_col() this time using pricepercent and then improve this step by step, first ordering the x-axis by value and finally making a so called "dot chat" or "lollipop" chart by swapping geom_col() for geom_point() + geom_segment().

Section 5 Exploring the correlation structure

```
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
cij <- cor(candy)
corrplot(cij)
```



Q22. Examining this plot what two variables are anti-correlated (i.e. have minus values)?

Chocolate and fruity

Q23. Similarly, what two variables are most positively correlated?

chocolate and chocolate, caramel and caramel, etc as going down the diagonal middle line also, chocolate and bar

PCA

The main function of this is `prcom()`, an we know we need to scale our data with the `scale= TRUE` argument

```
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
```

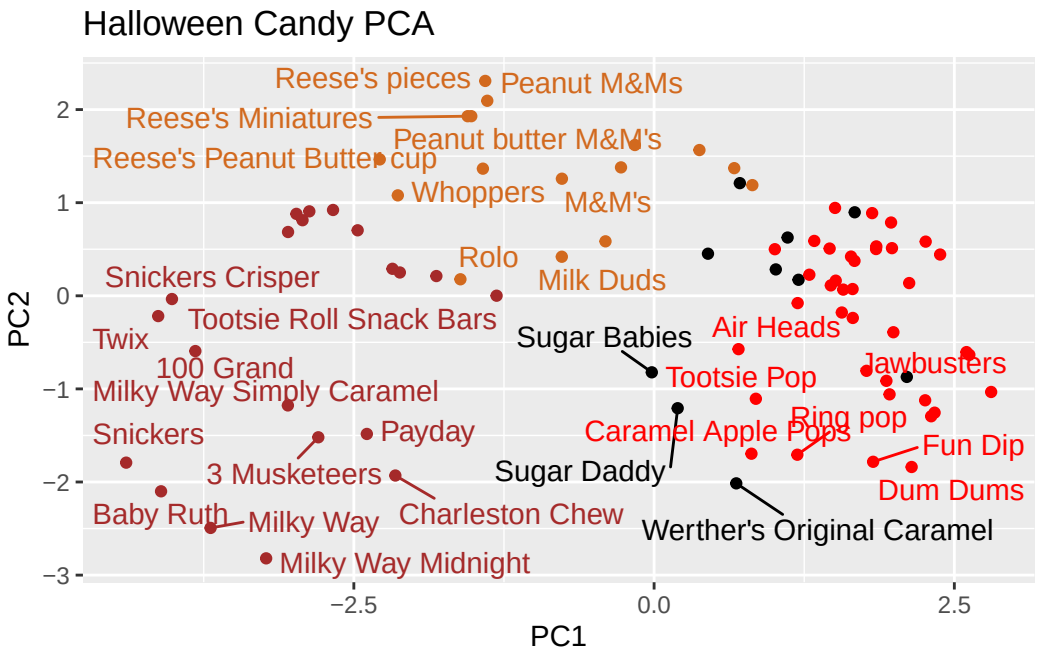
```
Cumulative Proportion  0.3601 0.4680 0.5705 0.66688 0.7424 0.79830 0.85369
                           PC8     PC9    PC10    PC11    PC12
Standard deviation     0.74530 0.67824 0.62349 0.43974 0.39760
Proportion of Variance 0.04629 0.03833 0.03239 0.01611 0.01317
Cumulative Proportion  0.89998 0.93832 0.97071 0.98683 1.00000
```

Plot my main PCA score plot with ggplot

```
# new dataframe with PCA results
my_data <- cbind(candy, pca$x[,1:3])
```

```
ggplot(my_data) +
        aes(x=PC1, y=PC2,
            label=rownames(candy)) +
        geom_point(col=my_cols) +
    geom_text_repel(col=my_cols) +
  labs(title= "Halloween Candy PCA ")
```

```
Warning: ggrepel: 54 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



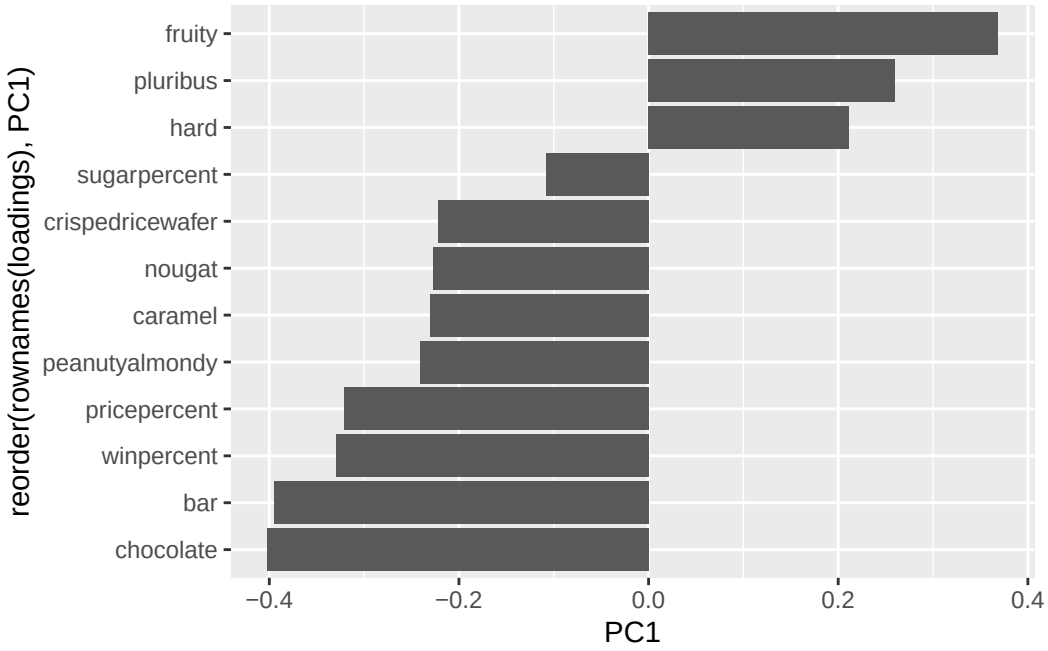
Loadings plot

```
pca$rotation
```

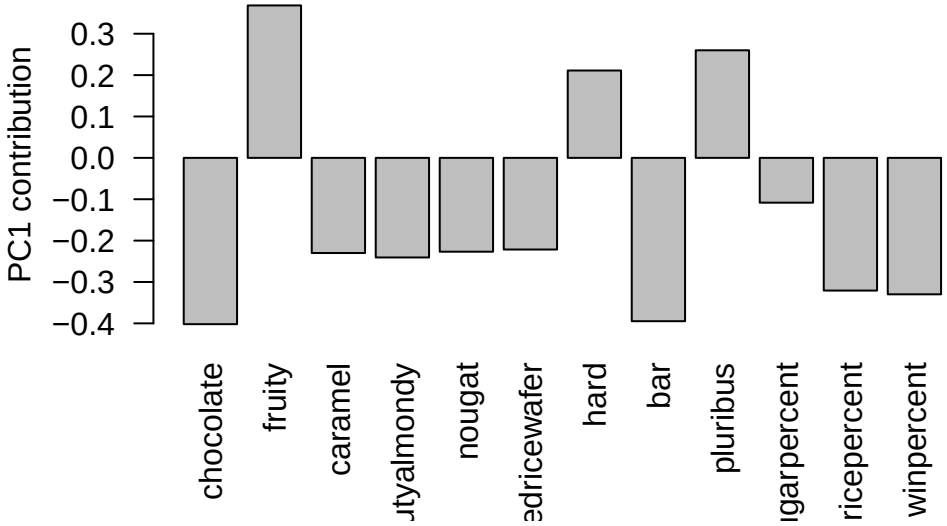
```
                      PC1         PC2         PC3          PC4          PC5
chocolate       -0.4019466  0.21404160  0.01601358 -0.016673032  0.066035846
fruity           0.3683883 -0.18304666 -0.13765612 -0.004479829  0.143535325
caramel         -0.2299709 -0.40349894 -0.13294166 -0.024889542 -0.507301501
peanutyalmondy  -0.2407155  0.22446919  0.18272802  0.466784287  0.399930245
nougat          -0.2268102 -0.47016599  0.33970244  0.299581403 -0.188852418
crispedricewafer -0.2215182  0.09719527 -0.36485542 -0.605594730  0.034652316
hard             0.2111587 -0.43262603 -0.20295368 -0.032249660  0.574557816
bar             -0.3947433 -0.22255618  0.10696092 -0.186914549  0.077794806
pluribus         0.2600041  0.36920922 -0.26813772  0.287246604 -0.392796479
sugarpercent    -0.1083088 -0.23647379 -0.65509692  0.433896248  0.007469103
pricepercent    -0.3207361  0.05883628 -0.33048843  0.063557149  0.043358887
winpercent      -0.3298035  0.21115347 -0.13531766  0.117930997  0.168755073
                      PC6         PC7         PC8          PC9         PC10
chocolate       -0.09018950 -0.08360642 -0.49084856 -0.151651568  0.107661356
fruity          -0.04266105  0.46147889  0.39805802 -0.001248306  0.362062502
caramel         -0.40346502 -0.44274741  0.26963447  0.019186442  0.229799010
peanutyalmondy  -0.09416259 -0.25710489  0.45771445  0.381068550 -0.145912362
nougat           0.09012643  0.36663902 -0.18793955  0.385278987  0.011323453
crispedricewafer -0.09007640  0.13077042  0.13567736  0.511634999 -0.264810144
hard            -0.12767365 -0.31933477 -0.38881683  0.258154433  0.220779142
bar              0.25307332  0.24192992 -0.02982691  0.091872886 -0.003232321
pluribus         0.03184932  0.04066352 -0.28652547  0.529954405  0.199303452
sugarpercent     0.02737834  0.14721840 -0.04114076 -0.217685759 -0.488103337
pricepercent     0.62908570 -0.14308215  0.16722078 -0.048991557  0.507716043
winpercent      -0.56947283  0.40260385 -0.02936405 -0.124440117  0.358431235
                     PC11        PC12
chocolate        0.10045278  0.69784924
fruity           0.17494902  0.50624242
caramel          0.13515820  0.07548984
peanutyalmondy   0.11244275  0.12972756
nougat          -0.38954473  0.09223698
crispedricewafer -0.22615618  0.11727369
hard             0.01342330 -0.10430092
bar              0.74956878 -0.22010569
pluribus         0.27971527 -0.06169246
sugarpercent     0.05373286  0.04733985
```

```
pricepercent      -0.26396582 -0.06698291
winpercent        -0.11251626 -0.37693153
```

```
loadings <- as.data.frame(pca$rotation)
ggplot(loadings) +
  aes(PC1, reorder(rownames(loadings), PC1)) +
  geom_col()
```




```
barplot(pca$rotation[,1], las=2, ylab= "PC1 contribution")
```



Q24. What original variables are picked up strongly by PC1 in the positive direction? Do these make sense to you?

Fruity, hard, and pluribus captured by PC1 and by correlation, therefore they do make sense.