# Class09: Candy Analysis Mini Project

Kalisa Kang (PID: A16741690)

In today's class, we will examine some data about candy from the 538 website.

### Import 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
```

### Data exploration

> Q1. How many different candy types are in this dataset?

```
nrow(candy)
```

```
[1] 85
```

There are 85 candy types in this dataset.

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

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

```
[1] 38
```

There are 38 fruity candy types in this dataset.

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

```r
candy["Nerds", ]$winpercent
```

```
[1] 55.35405
```

> Q4. What is the winpercent value for "Kit Kat"?

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

```
[1] 76.7686
```

> Q5. What is the winpercent value for "Tootsie Roll Snack Bars"?

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

```
[1] 49.6535
```

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

```r
# Short cut syntax that tells R to go into package and execute the function
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 | |

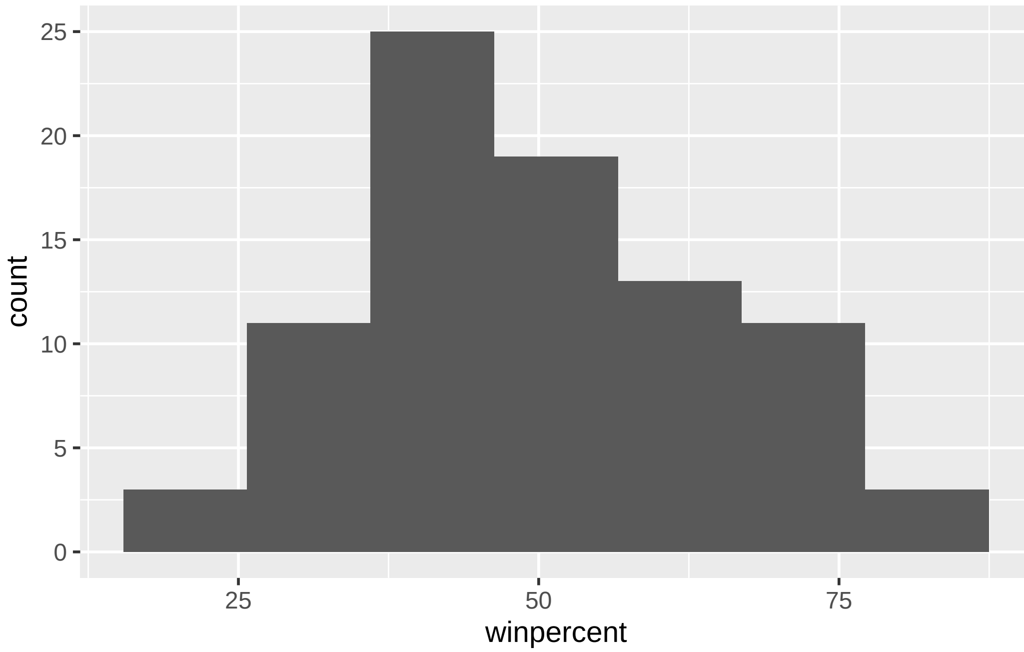
The winpercent variable is on a different scale than the other variables.

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

Zero represents that chocolate is not present, while 1 represents that chocolate is present in the candy.

Q8. Plot a histogram of winpercent values

```
library(ggplot2)
ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins=7)
```




Q9. Is the distribution of winpercent values symmetrical?

No, the distribution is not symmetrical.

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

```
summary(candy$winpercent)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  22.45   39.14   47.83   50.32   59.86   84.18 
```

The center is below 50%.

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

- First find all chocolate candy
- Then find their winpercent values
- Find the mean winpercent
- Then do the same for fruity candy and compare w/ the mean for chocolate candy

```
choc_winpercent <- candy$winpercent[as.logical(candy$chocolate)]
fruity_winpercent <- candy$winpercent[as.logical(candy$fruity)]
mean(choc_winpercent) > mean(fruity_winpercent)
```

```
[1] TRUE
```

```r
chocolate.inds <- candy$chocolate == 1
chocolate.win <- candy[chocolate.inds,]$winpercent
mean(chocolate.win)
```

```
[1] 60.92153
```

```r
fruity.inds <- candy$fruity == 1
fruity.win <- candy[fruity.inds,]$winpercent
mean(fruity.win)
```

```
[1] 44.11974
```

```r
mean(chocolate.win) > mean(fruity.win)
```

```
[1] TRUE
```

On average, chocolate candy is higher ranked than fruit candy (ie higher winpercent).

> Q12. Is this difference statistically significant?

```r
t.test(chocolate.win, fruity.win)
```

```

	Welch Two Sample t-test

data:  chocolate.win and fruity.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 because the p-value of 2.871e-8 is less than .05.

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

```r
x <- c(5,6,4)
sort(x)
```

```
[1] 4 5 6
```

```r
order(x)
```

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

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

```
[1] 4 5 6
```

The order function returns the indices that make the input sorted.

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

```
                   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
                   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
```

```r
library(dplyr)
```

```

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
```

```r
candy %>%
  arrange(winpercent) %>%
  head(5)
```

```
                   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
                   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
```

The 5 least likely candy types in this set are Nik L Nip, Boston Baked Beans, Chiclets, Super Bubbles, and Jawbusters.

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

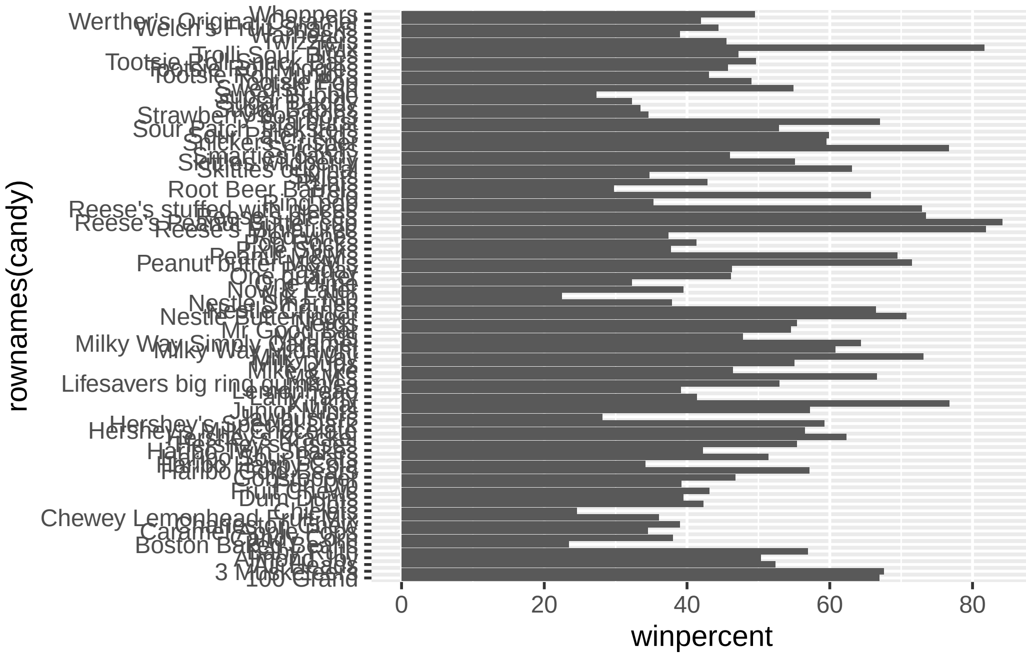
```r
candy %>%
  arrange(winpercent) %>%
  tail(5)
```

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

The top 5 favorite candies are Snickers, Kit Kat, Twix, Reese's Miniatures, and Reese's peanut butter cup.

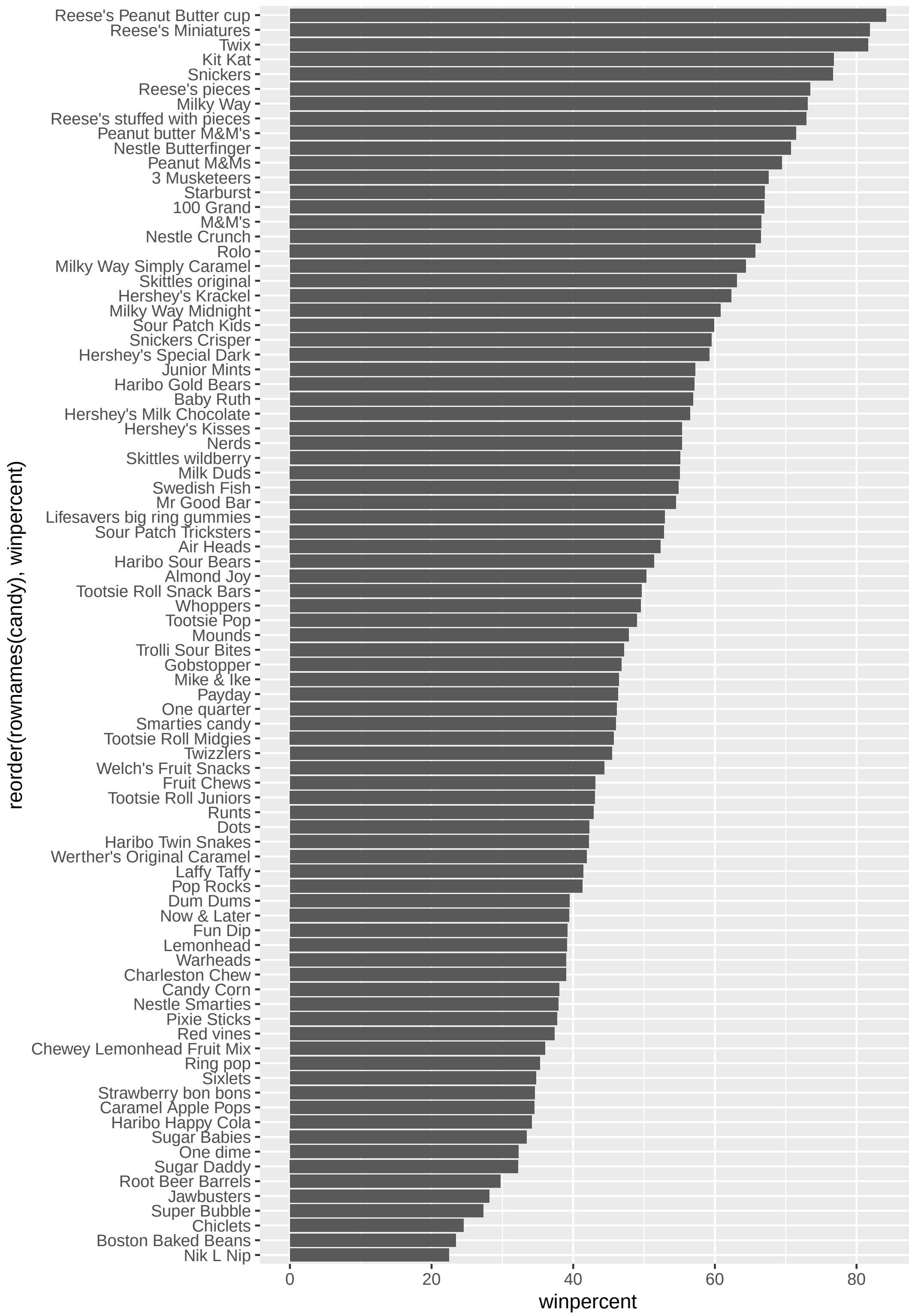
> Q15. Make a first barplot of candy ranking based on winpercent values.

```r
library(ggplot2)

ggplot(candy) +
  aes(x=winpercent, y=rownames(candy)) +
  geom_col()
```



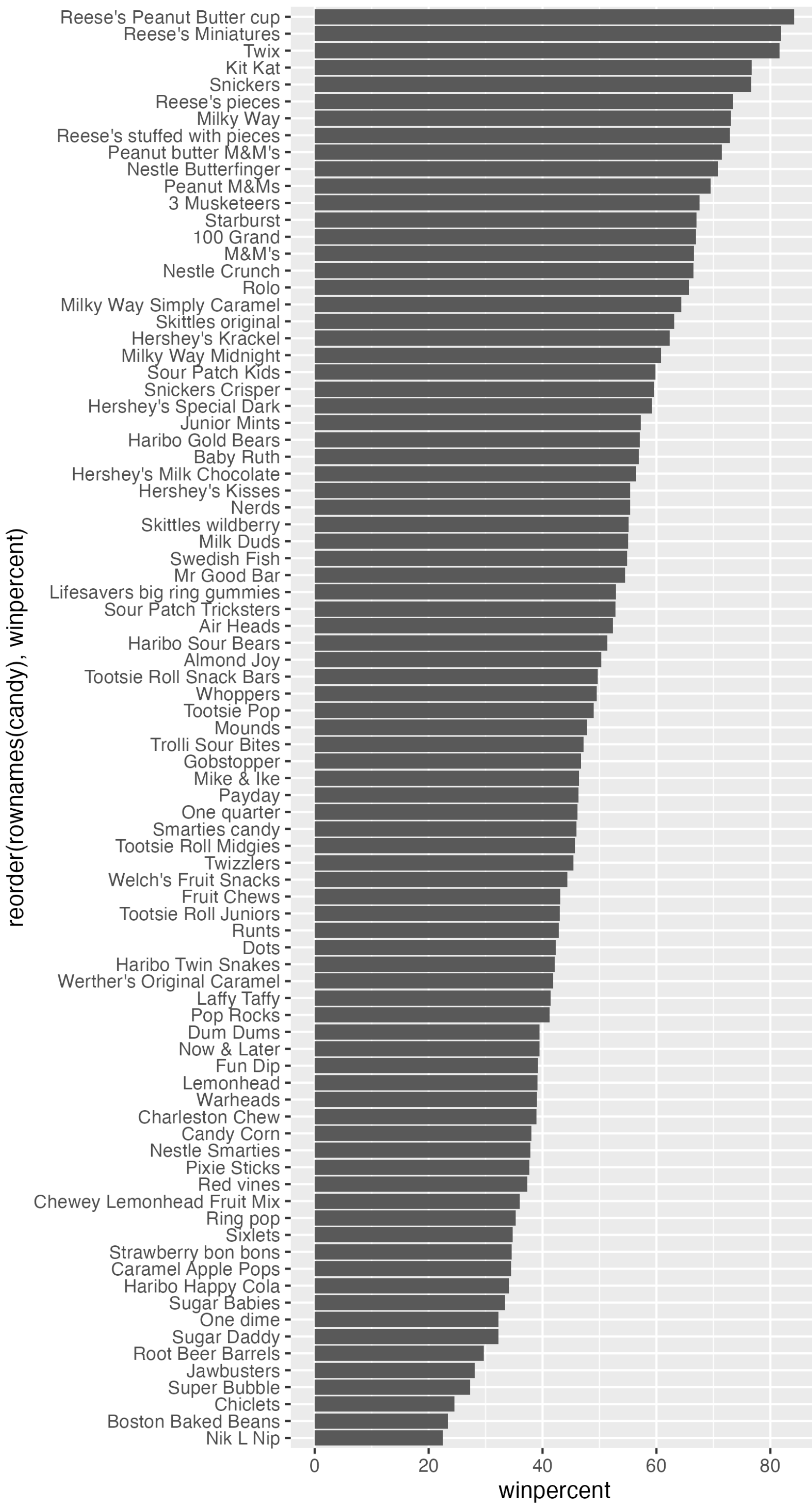

Q16. This is quite ugly, use the reorder() function to get the bars sorted by winpercent?

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


Reese's Peanut Butter cup
Reese's Miniatures
Twix
Kit Kat
Snickers
Reese's pieces
Milky Way
Reese's stuffed with pieces
Peanut butter M&M's
Nestle Butterfinger
Peanut M&Ms
3 Musketeers
Starburst
100 Grand
M&M's
Nestle Crunch
Rolo
Milky Way Simply Caramel
Skittles original
Hershey's Krackel
Milky Way Midnight
Sour Patch Kids
Snickers Crisper
Hershey's Special Dark
Junior Mints
Haribo Gold Bears
Baby Ruth
Hershey's Milk Chocolate
Hershey's Kisses
Nerds
Skittles wildberry
Milk Duds
Swedish Fish
Mr Good Bar
Lifesavers big ring gummies
Sour Patch Tricksters
Air Heads
Haribo Sour Bears
Almond Joy
Tootsie Roll Snack Bars
Whoppers
Tootsie Pop
Mounds
Trolli Sour Bites
Gobstopper
Mike & Ike
Payday
One quarter
Smarties candy
Tootsie Roll Midgies
Twizzlers
Welch's Fruit Snacks
Fruit Chews
Tootsie Roll Juniors
Runts
Dots
Haribo Twin Snakes
Werther's Original Caramel
Laffy Taffy
Pop Rocks
Dum Dums
Now & Later
Fun Dip
Lemonhead
Warheads
Charleston Chew
Candy Corn
Nestle Smarties
Pixie Sticks
Red vines
Chewey Lemonhead Fruit Mix
Ring pop
Sixlets
Strawberry bon bons
Caramel Apple Pops
Haribo Happy Cola
Sugar Babies
One dime
Sugar Daddy
Root Beer Barrels
Jawbusters
Super Bubble
Chiclets
Boston Baked Beans
Nik L Nip
reorder(rownames(candy), winpercent)
0
20
40
60
80
winpercent

```r
ggsave("mybarplot.png", height=10)
```

```
Saving 5.5 x 10 in image
```

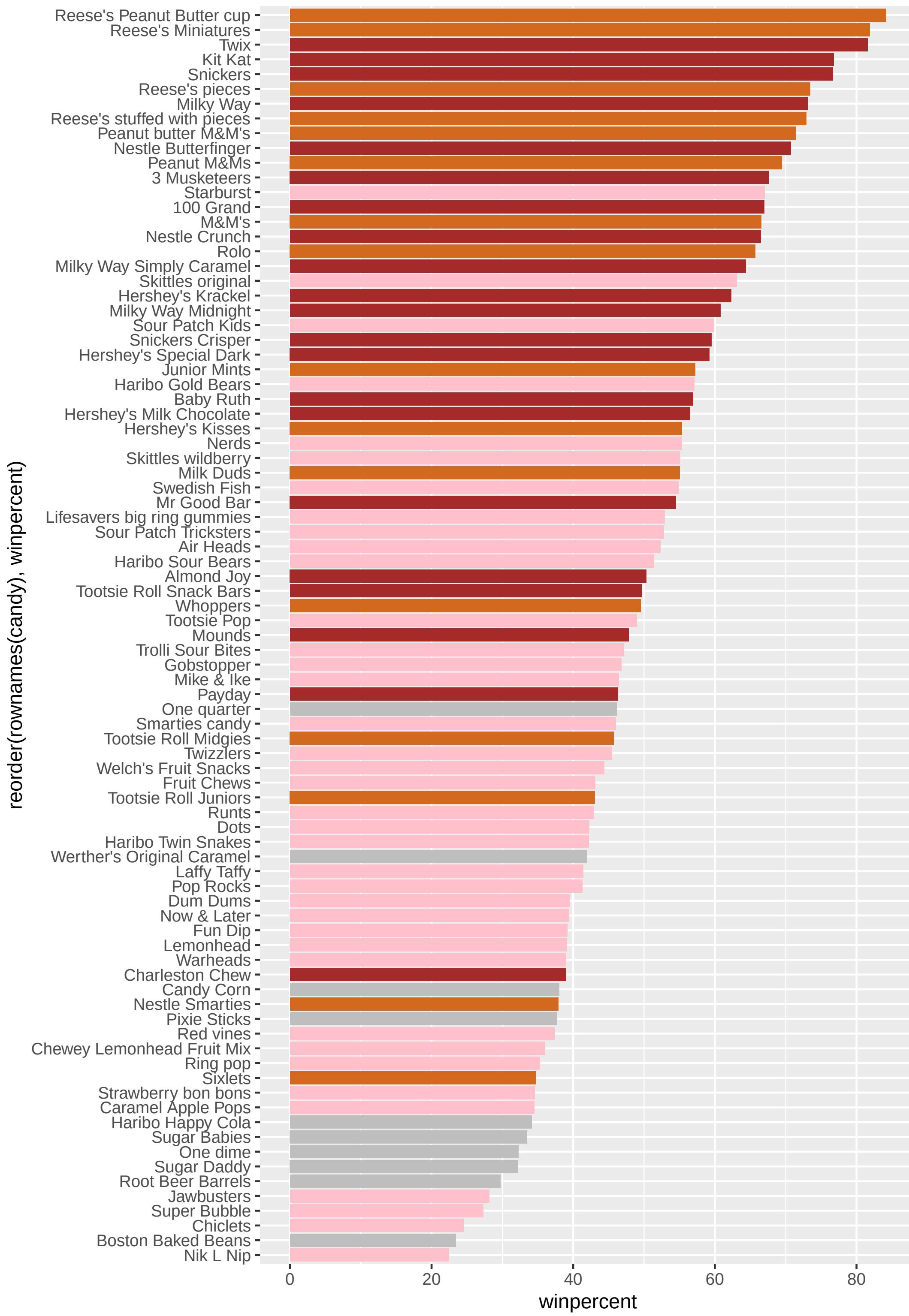



Adding my custom colors

to my bar plot

```
my_cols=rep("grey", nrow(candy))
my_cols[as.logical(candy$chocolate)] = "chocolate"
my_cols[as.logical(candy$bar)] = "brown"
my_cols[as.logical(candy$fruity)] = "pink"
```

```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy),winpercent)) +
# `reorder(rownames(candy)), winpercent` reorders the candy row names by winpercent
  geom_col(fill=my_cols)
```


Reese's Peanut Butter cup
Reese's Miniatures
Twix
Kit Kat
Snickers
Reese's pieces
Milky Way
Reese's stuffed with pieces
Peanut butter M&M's
Nestle Butterfinger
Peanut M&Ms
3 Musketeers
Starburst
100 Grand
M&M's
Nestle Crunch
Rolo
Milky Way Simply Caramel
Skittles original
Hershey's Krackel
Milky Way Midnight
Sour Patch Kids
Snickers Crisper
Hershey's Special Dark
Junior Mints
Haribo Gold Bears
Baby Ruth
Hershey's Milk Chocolate
Hershey's Kisses
Nerds
Skittles wildberry
Milk Duds
Swedish Fish
Mr Good Bar
Lifesavers big ring gummies
Sour Patch Tricksters
Air Heads
Haribo Sour Bears
Almond Joy
Tootsie Roll Snack Bars
Whoppers
Tootsie Pop
Mounds
Trolli Sour Bites
Gobstopper
Mike & Ike
Payday
One quarter
Smarties candy
Tootsie Roll Midgies
Twizzlers
Welch's Fruit Snacks
Fruit Chews
Tootsie Roll Juniors
Runts
Dots
Haribo Twin Snakes
Werther's Original Caramel
Laffy Taffy
Pop Rocks
Dum Dums
Now & Later
Fun Dip
Lemonhead
Warheads
Charleston Chew
Candy Corn
Nestle Smarties
Pixie Sticks
Red vines
Chewey Lemonhead Fruit Mix
Ring pop
Sixlets
Strawberry bon bons
Caramel Apple Pops
Haribo Happy Cola
Sugar Babies
One dime
Sugar Daddy
Root Beer Barrels
Jawbusters
Super Bubble
Chiclets
Boston Baked Beans
Nik L Nip
reorder(rownames(candy), winpercent)
0
20
40
60
80
winpercent

Q17. What is the worst ranked chocolate candy?

The worst ranked chocolate Sixlets.

Q18. What is the best ranked fruity candy?

The best ranked fruity candy is Starburst.

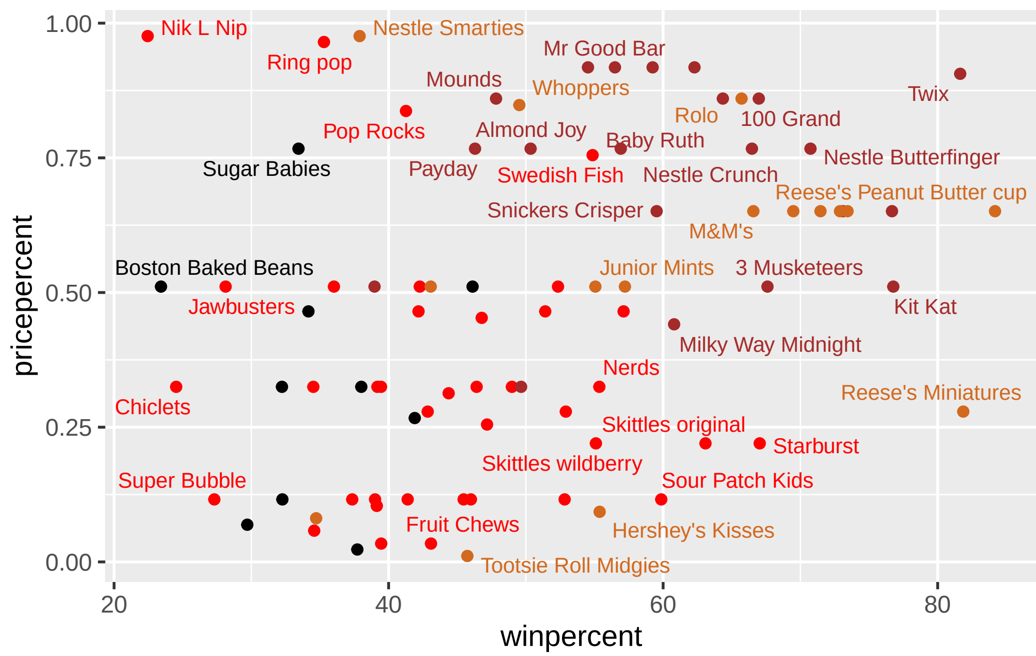
Q19. Which candy type is the highest ranked in terms of winpercent for the least money - i.e. offers the most bang for your buck?

```
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"
```

We can use `ggrepel` package to do a better job of placing labels next to the data points.

```
library(ggrepel)

# How about a plot of price vs win
ggplot(candy) +
  aes(x=winpercent, y=pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols, size=2.7, max.overlaps = 8)
```

```
Warning: ggrepel: 48 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




Reese's minatures give the most bang for your buck!

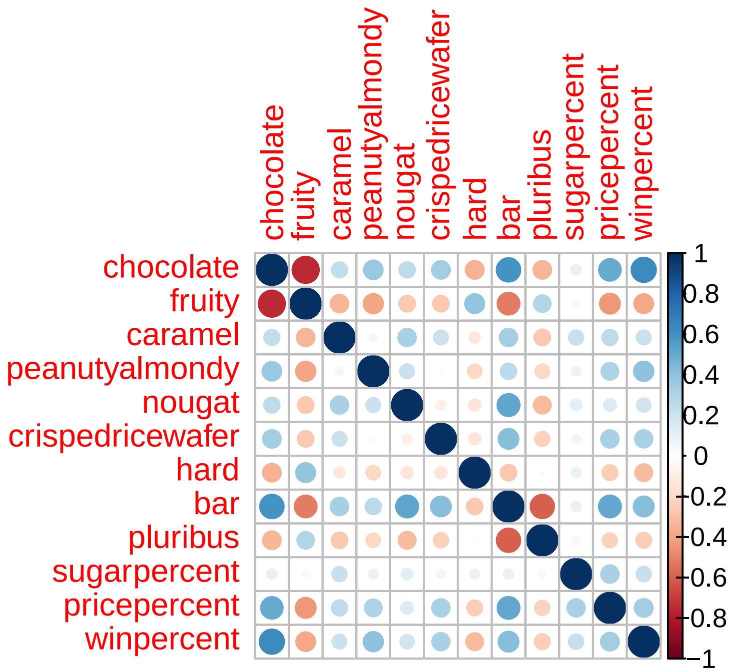
> Q20. What are the top 5 most expensive candy types in the dataset and of these which is the least popular?

## 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)?

Fruity and chocolate, and bar and pluribus are anti-correlated.

There's some redundancy in this correlation plot and PCA is better.

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

Bar and chocolate, chocolate and winpercent are the most positively correlated.

## 6 Principal Component Analysis

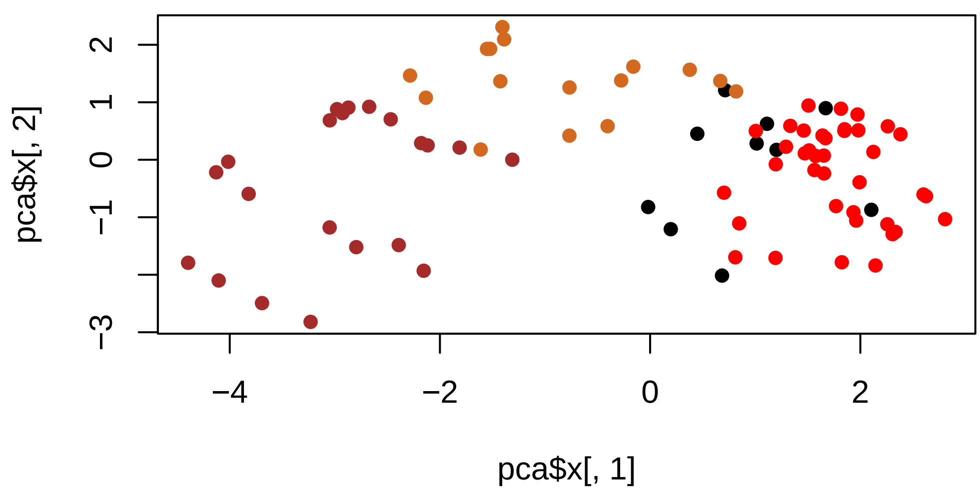
We will perform a PCA of the candy. Key question: do we need to scale the data before PCA?

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

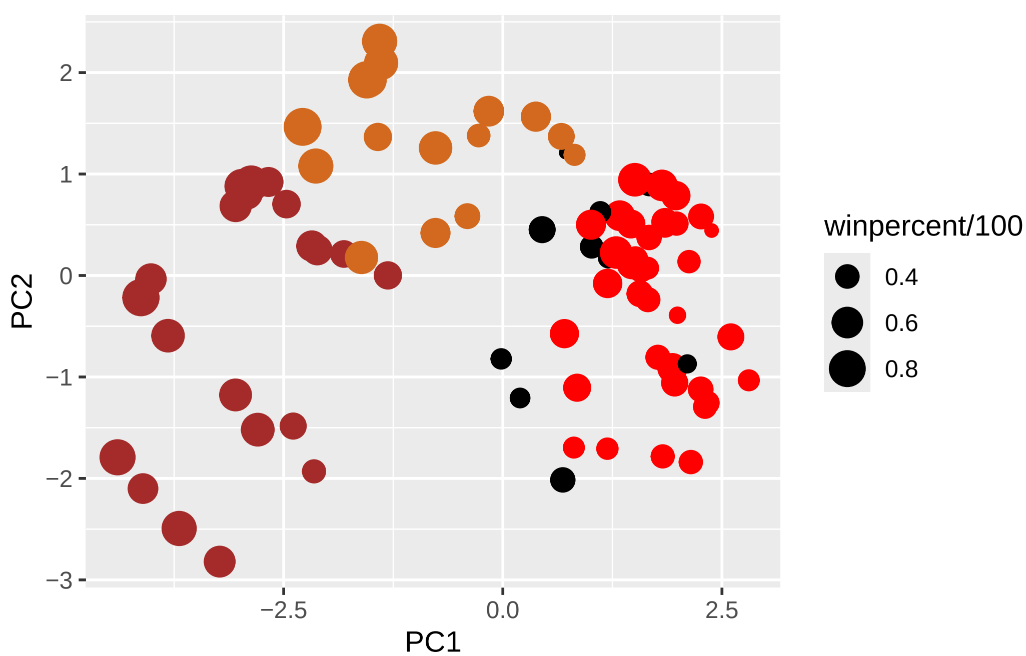



```
# Make a new data-frame with our PCA results and candy data
my_data <- cbind(candy, pca$x[,1:3])
head(my_data)
```

```
             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       PC1
```

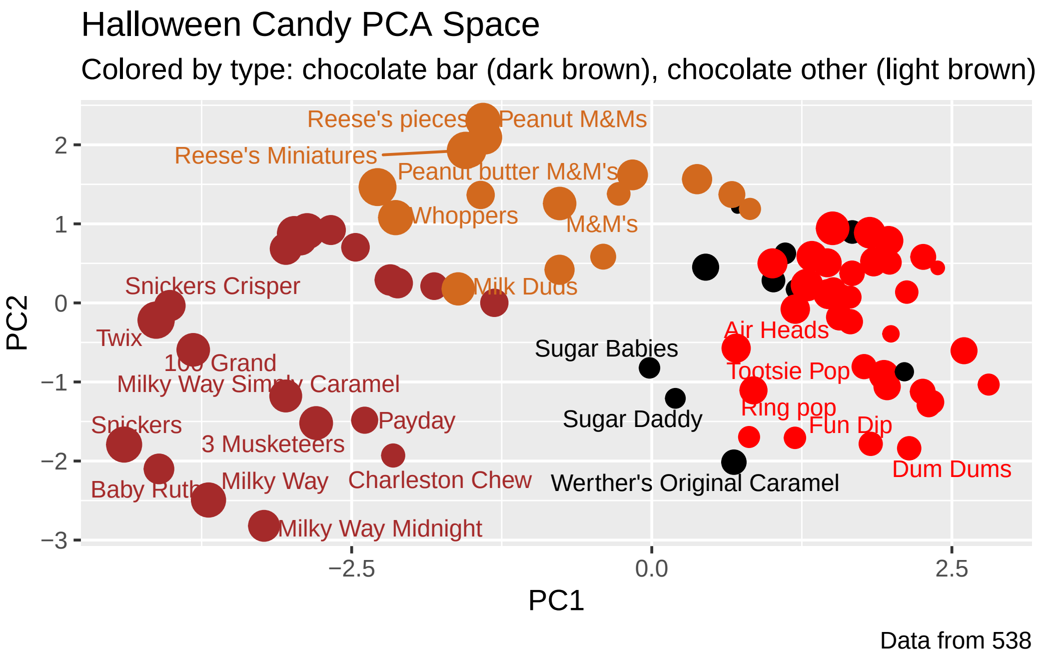
```
100 Grand        0    1          0          0.732          0.860    66.97173 -3.8198617
3 Musketeers     0    1          0          0.604          0.511    67.60294 -2.7960236
One dime         0    0          0          0.011          0.116    32.26109  1.2025836
One quarter      0    0          0          0.011          0.511    46.11650  0.4486538
Air Heads        0    0          0          0.906          0.511    52.34146  0.7028992
Almond Joy       0    1          0          0.465          0.767    50.34755 -2.4683383
                        PC2        PC3
100 Grand    -0.5935788 -2.1863087
3 Musketeers -1.5196062  1.4121986
One dime      0.1718121  2.0607712
One quarter   0.4519736  1.4764928
Air Heads    -0.5731343 -0.9293893
Almond Joy    0.7035501  0.8581089
```

```
p <- ggplot(my_data) +
        aes(x=PC1, y=PC2,
            size=winpercent/100,
            text=rownames(my_data),
            label=rownames(my_data)) +
        geom_point(col=my_cols)

p
```



```r
library(ggrepel)

p + geom_text_repel(size=3.3, col=my_cols, max.overlaps = 7)  +
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
       subtitle="Colored by type: chocolate bar (dark brown), chocolate other (light brown
       caption="Data from 538")
```

```
Warning: ggrepel: 59 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




```r
#library(plotly)
#ggplotly(p)
```

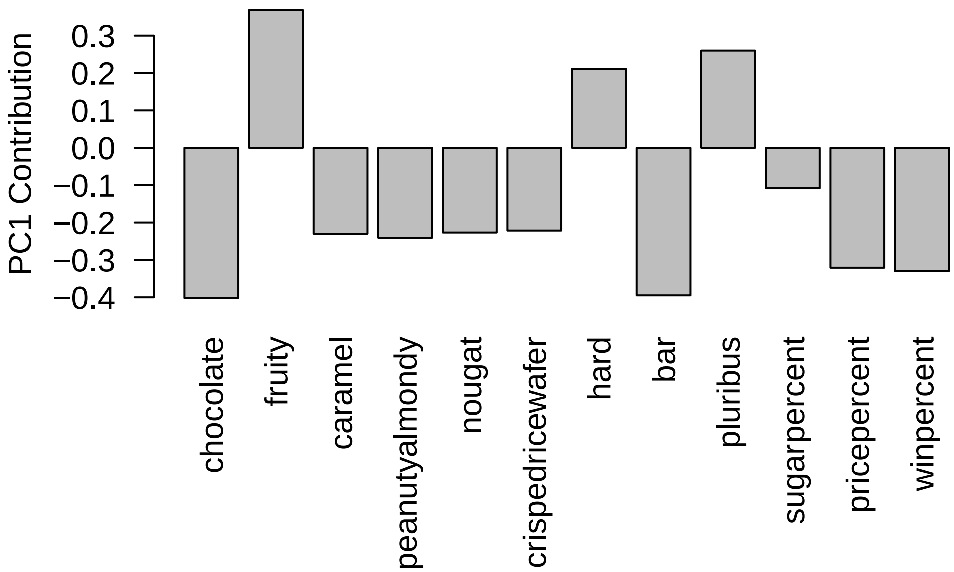
How do the original variables contribute to our PCs? For this, we look at the loadings component of our results object i.e. the `pca$rotation` object.l

```r
head(pca$rotation)
```

```
                PC1         PC2         PC3          PC4          PC5
```

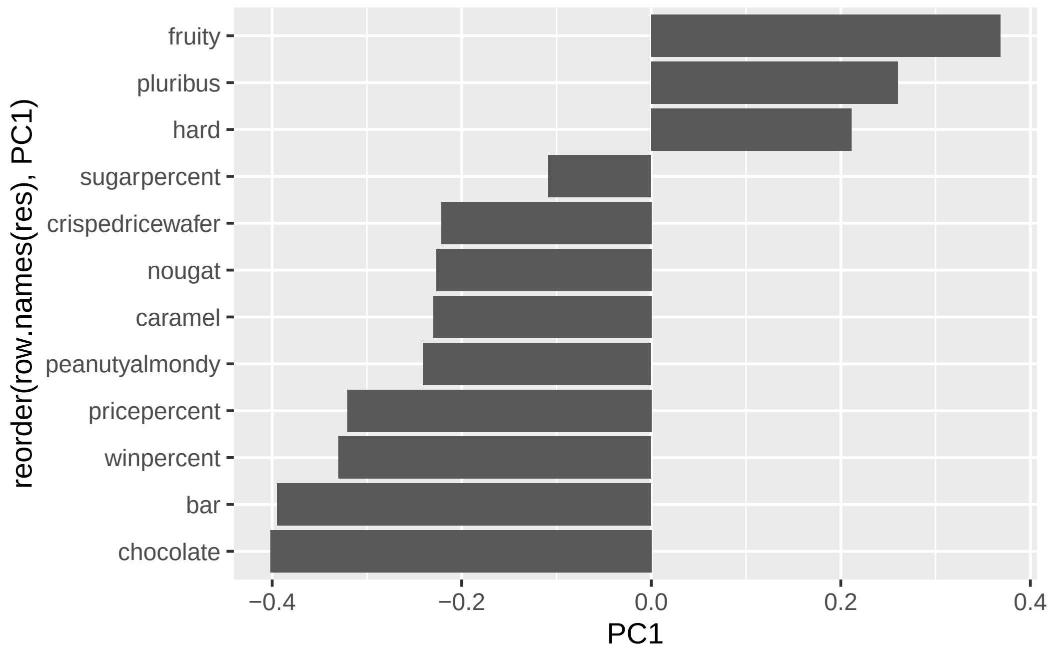
```
chocolate        -0.4019466  0.21404160  0.01601358 -0.016673032  0.06603585
fruity            0.3683883 -0.18304666 -0.13765612 -0.004479829  0.14353533
caramel          -0.2299709 -0.40349894 -0.13294166 -0.024889542 -0.50730150
peanutyalmondy   -0.2407155  0.22446919  0.18272802  0.466784287  0.39993025
nougat           -0.2268102 -0.47016599  0.33970244  0.299581403 -0.18885242
crispedricewafer -0.2215182  0.09719527 -0.36485542 -0.605594730  0.03465232
                         PC6         PC7        PC8          PC9        PC10
chocolate        -0.09018950 -0.08360642 -0.4908486 -0.151651568  0.10766136
fruity           -0.04266105  0.46147889  0.3980580 -0.001248306  0.36206250
caramel          -0.40346502 -0.44274741  0.2696345  0.019186442  0.22979901
peanutyalmondy   -0.09416259 -0.25710489  0.4577145  0.381068550 -0.14591236
nougat            0.09012643  0.36663902 -0.1879396  0.385278987  0.01132345
crispedricewafer -0.09007640  0.13077042  0.1356774  0.511634999 -0.26481014
                       PC11       PC12
chocolate         0.1004528 0.69784924
fruity            0.1749490 0.50624242
caramel           0.1351582 0.07548984
peanutyalmondy    0.1124428 0.12972756
nougat           -0.3895447 0.09223698
crispedricewafer -0.2261562 0.11727369
```

```r
par(mar=c(8,4,2,2))
barplot(pca$rotation[,1], las=2, ylab="PC1 Contribution")
```




Make a bar plot with ggplot and order the bars by their value.

```
res <- pca$rotation

ggplot(res) +
  aes(PC1, reorder(row.names(res), PC1)) +
  geom_col()
```




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

The fruity, hard, and pluribus variables are picked up strongly by PC1 in the positive direction. This makes sense to me because based on the correlation structure in the dataset, if it is a fruity candy, then it will tend to be hard and come in a packet with multiple candies.