Class 11: Halloween Mini-Project

Emily Rodriguez

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

```
nrow(candy)
```

```
[1] 85
```

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

```
candy$fruity
```

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

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

```
[1] 38
```

Q. What are these fruity candy?

We can use the ==

```
rownames( candy[ candy$fruity == 1, ])
```

```
 [1] "Air Heads"                   "Caramel Apple Pops"
 [3] "Chewey Lemonhead Fruit Mix"  "Chiclets"
 [5] "Dots"                        "Dum Dums"
 [7] "Fruit Chews"                 "Fun Dip"
 [9] "Gobstopper"                  "Haribo Gold Bears"
[11] "Haribo Sour Bears"           "Haribo Twin Snakes"
[13] "Jawbusters"                  "Laffy Taffy"
[15] "Lemonhead"                   "Lifesavers big ring gummies"
[17] "Mike & Ike"                  "Nerds"
[19] "Nik L Nip"                   "Now & Later"
[21] "Pop Rocks"                   "Red vines"
[23] "Ring pop"                    "Runts"
[25] "Skittles original"           "Skittles wildberry"
[27] "Smarties candy"              "Sour Patch Kids"
[29] "Sour Patch Tricksters"       "Starburst"
[31] "Strawberry bon bons"         "Super Bubble"
[33] "Swedish Fish"                "Tootsie Pop"
[35] "Trolli Sour Bites"           "Twizzlers"
[37] "Warheads"                    "Welch's Fruit Snacks"
```

How often does my favorite candy win?

```
candy["Twix",]$winpercent
```

```
[1] 81.64291
```

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

```
candy["Starburst",]$winpercent
```

```
[1] 67.03763
```

Q4. What is the winpercent value for “Kit Kat”?

```
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
```

```
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	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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?

Yes, the `winpercent` column is on a 0:100 scale and all other appear to be 0:1 scale.

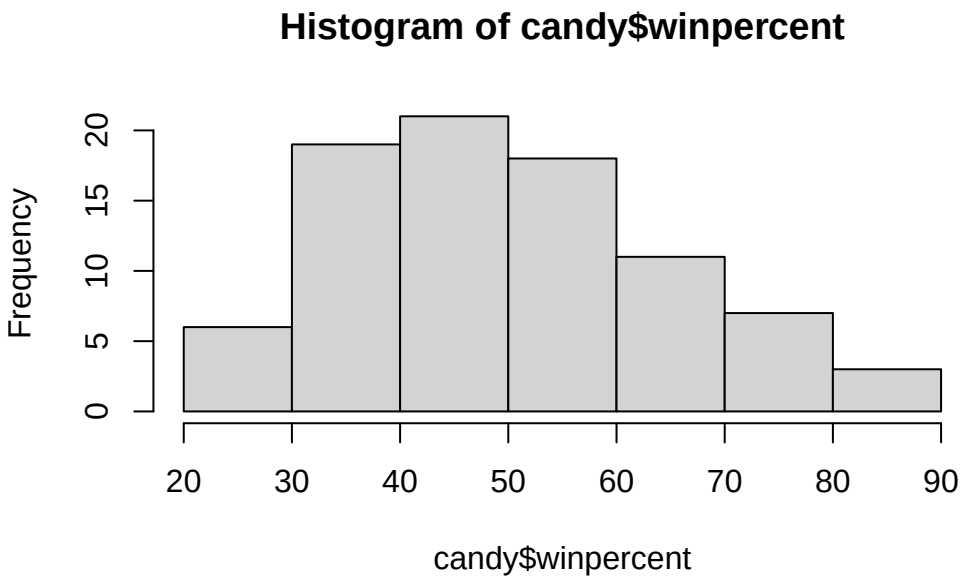
Q7. What do you think a zero and one represent for the candy$chocolate column?

A zero here means that the candy is not classified as containing chocolate.

Q8. Plot a histogram of winpercent values

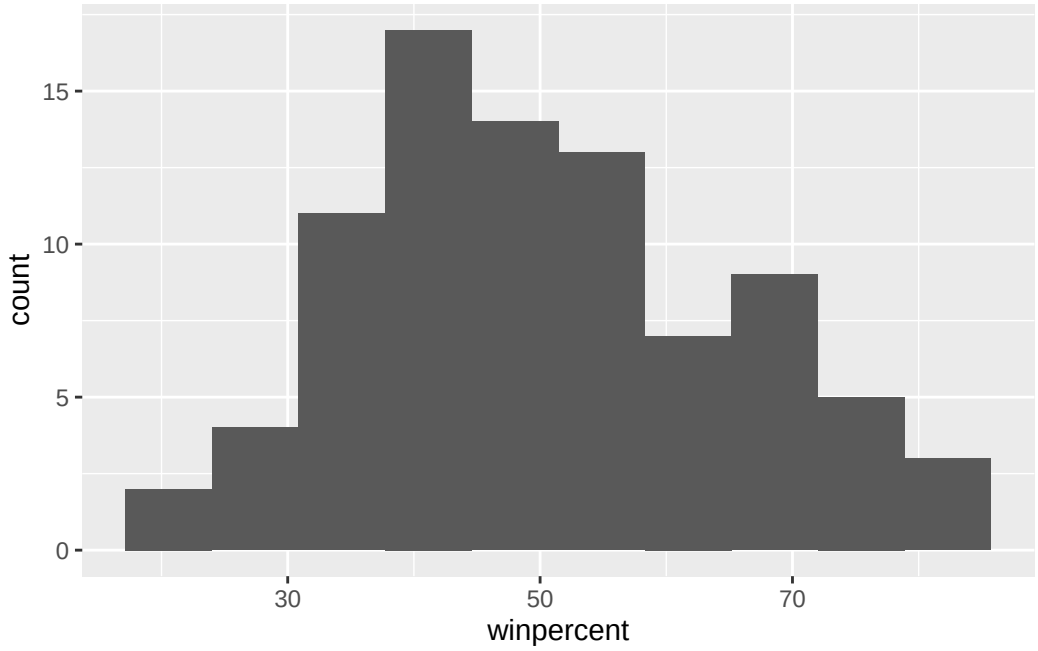
In base R graphics:

```
hist(candy$winpercent)
```



```
library(ggplot2)

ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins = 10)
```



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%?

The center of distribution is below 50% with a mean:

```
mean(candy$winpercent)
```

```
[1] 50.31676
```

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

To answer this question I will need to: - "subset" (a.k.a. "select", "filter") the candy dataset to just chocolate candy - get their winpercent values - calculate the mean of these.

Then do the same for fruity candy and compare.

```
# Filter/select/subset to just chocolate rows
chocolate.candy <- candy[as.logical(candy$chocolate),]

# Get their winpercent values
chocolate.winpercent <- chocolate.candy$winpercent

# Calculate their mean winpercent value
mean(chocolate.winpercent)
```

```
[1] 60.92153
```

Now we will do the same steps for fruity candy.

```
# Filter/select/subset to just fruity rows
fruity.candy <- candy[as.logical(candy$fruity),]

# Get their winpercent values
fruity.winpercent <- fruity.candy$winpercent

# Calculate their mean winpercent value
mean(fruity.winpercent)
```

```
[1] 44.11974
```

On average, chocolate is higer ranked than fruity candy.

Q12. Is this difference statistically significant?

```
t.test(chocolate.winpercent, fruity.winpercent)
```

```
    Welch Two Sample t-test

data:  chocolate.winpercent and fruity.winpercent
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
```

Yes, the difference is statistically significant.

Overall Candy Rankings

There os a base R function called `sort()` for, guess what sorting vectors of input.

```
x <- c(5, 2, 10)

#sort(x, decreasing = TRUE)
sort(x)
```

```
[1]  2  5 10
```

The buddy function to `sort()` that is often more useful is called `order()`. It returns the "indices" of the input that would result in it being sorted

```
order(x)
```

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

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

```
[1]  2  5 10
```

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

I can order by `winpercent`

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

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

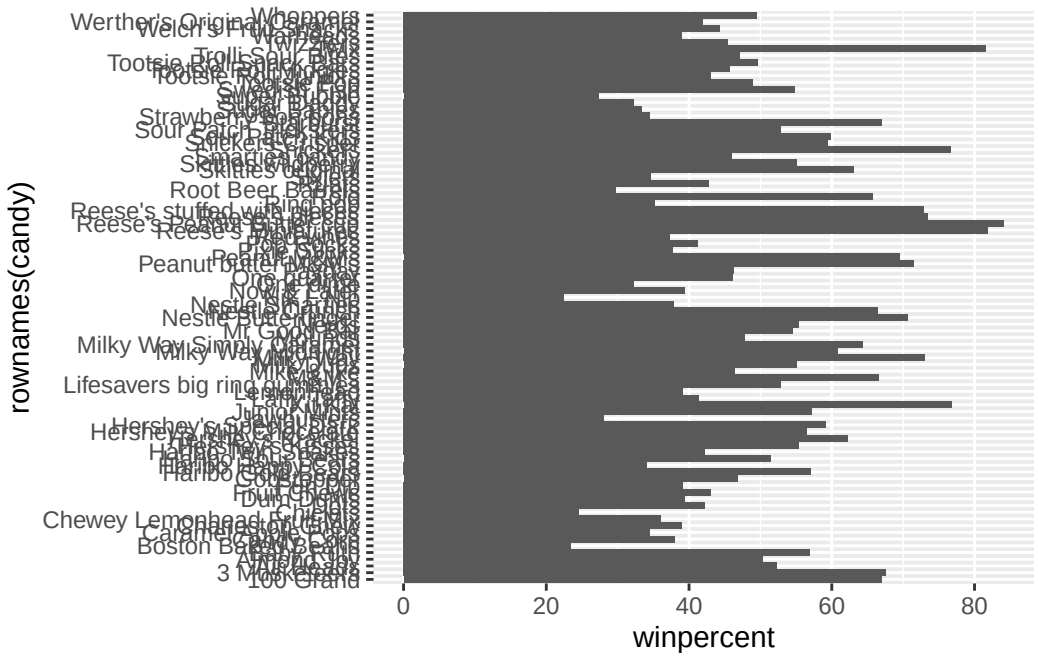
```
ord <- order(candy$winpercent, decreasing = TRUE)
head( candy[ord, ], 5)
```

```
                        chocolate fruity caramel peanutyalmondy nougat
Reese's Peanut Butter cup         1      0       0              1      0
Reese's 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
Reese's Peanut Butter cup               0    0   0        0        0.720
Reese's 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
Reese's Peanut Butter cup        0.651   84.18029
Reese's Miniatures               0.279   81.86626
Twix                             0.906   81.64291
Kit Kat                          0.511   76.76860
Snickers                         0.651   76.67378
```

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

```
library(ggplot2)

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

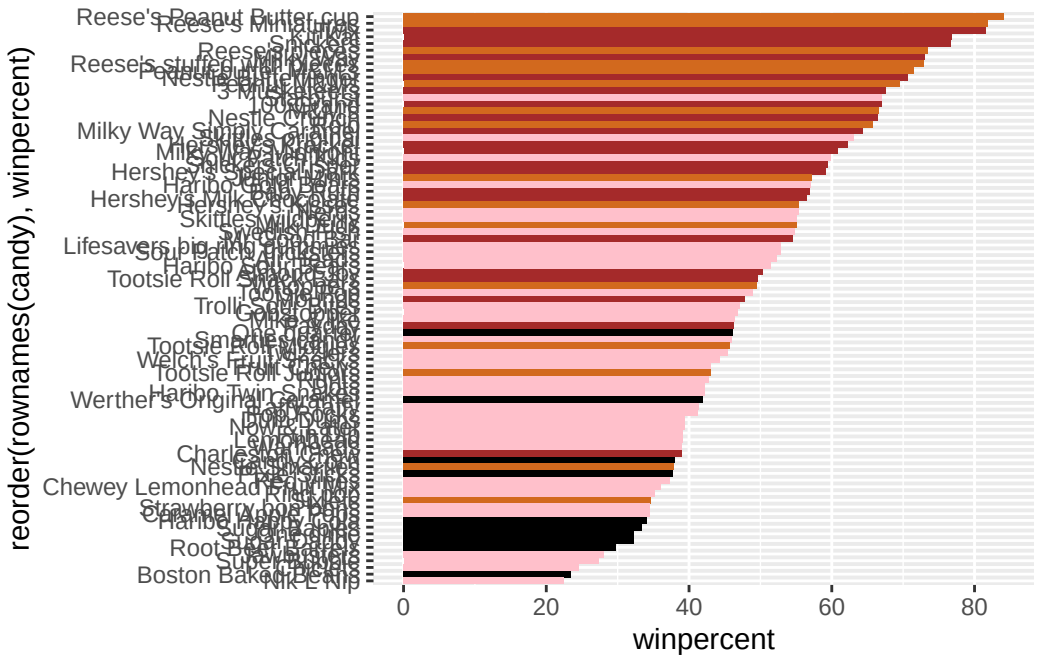


Q16. This is quite ugly, use the reorder() function to get the bars sorted by winpercent?

```
library(ggplot2)

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)] = "pink"

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



Q17. What is the worst ranked chocolate candy?

Sixlets

Q18. What is the best ranked fruity candy?

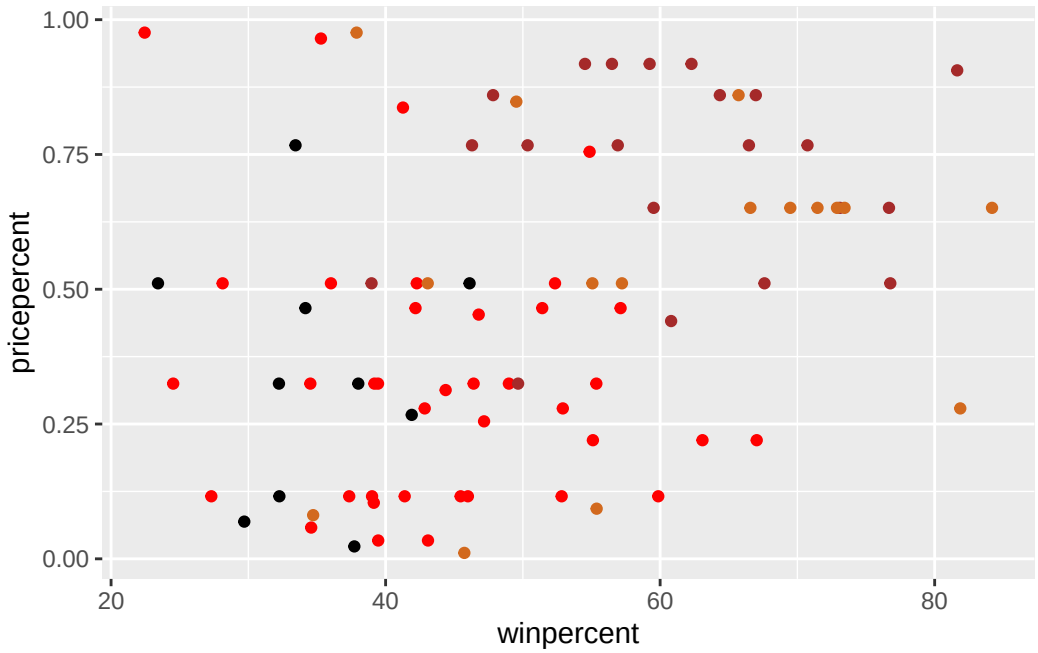
Starburst

Taking a look at pricepercent

Q. What is the best candy for the least money?

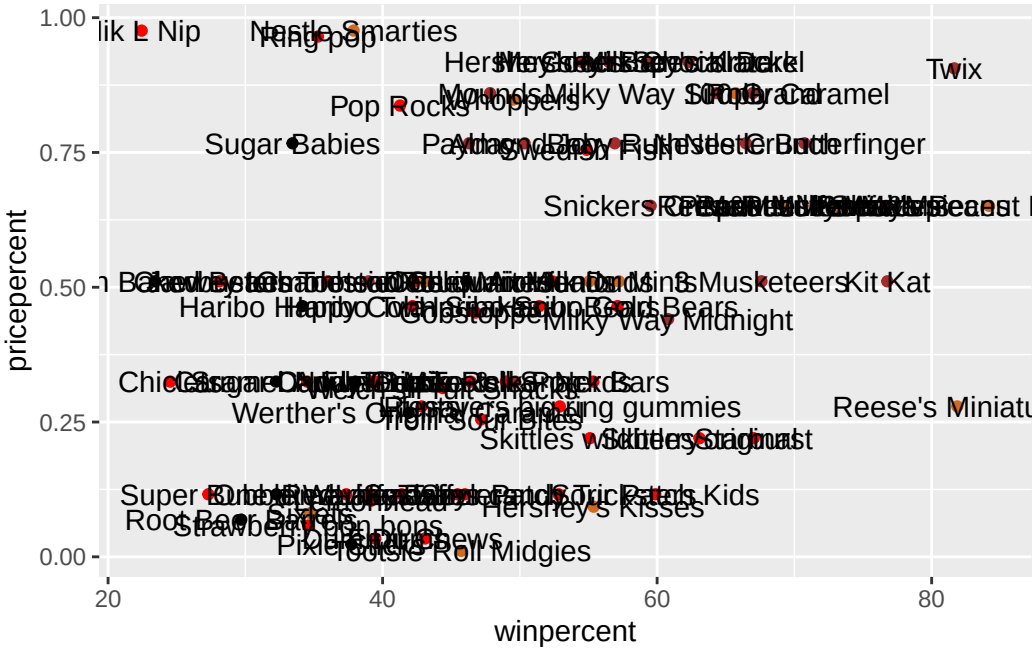
```
my_cols[as.logical(candy$fruity)] = "red"
```

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



Add some labels

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

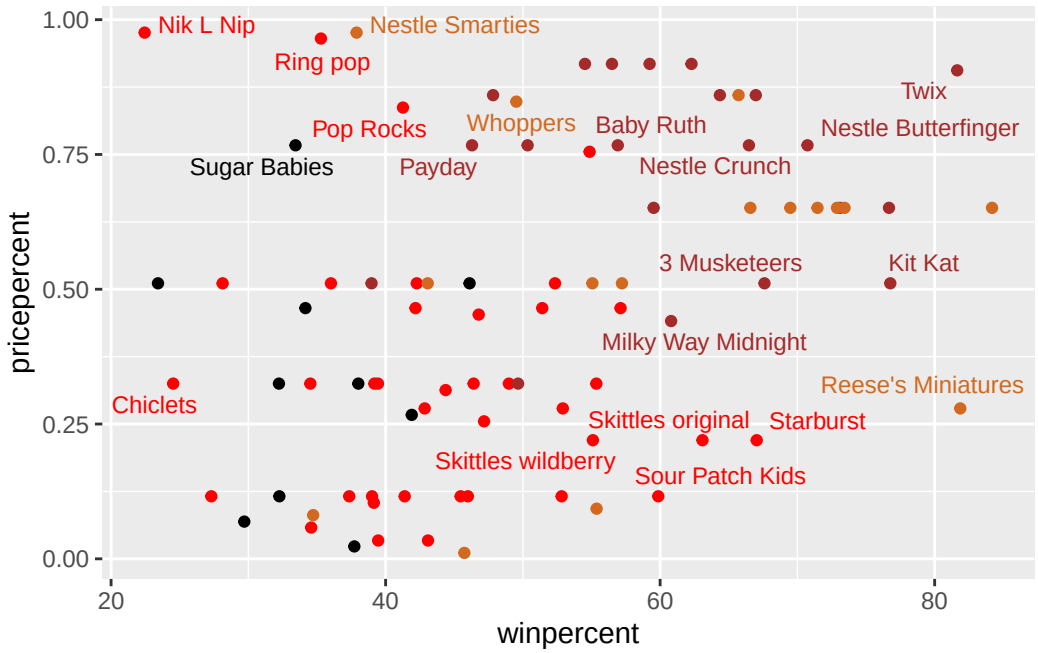


To deal with overlapping labels I can use the **ggrepel** package.

```
library(ggrepel)

ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols, size=3.3, max.overlaps = 5)
```

```
Warning: ggrepel: 65 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



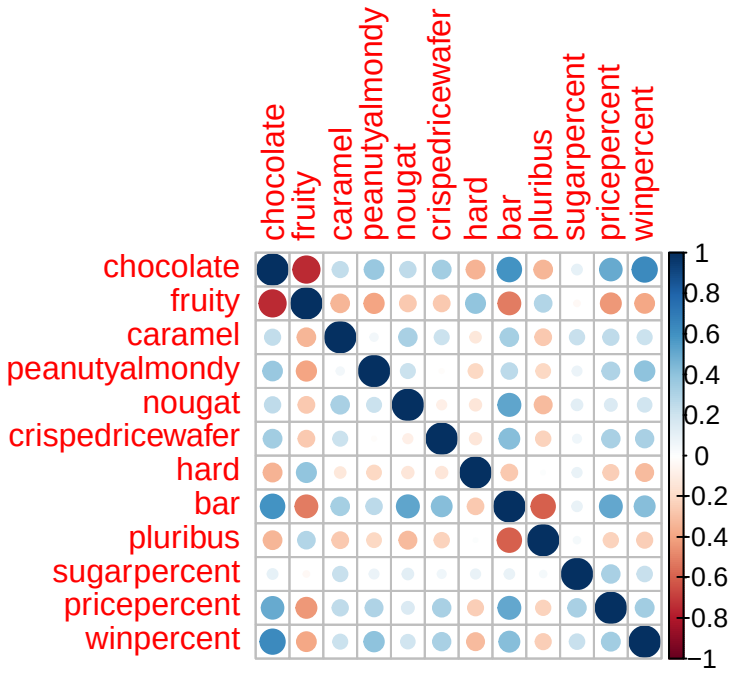
5 Exploring the correlation structure

Pearson correlation goea between -1 and +1 with zero indicating no correlation and the values close to one being very higly (anti) correlated.

```
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 are anti-correlated.

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

Chocolate and winpercent ot chocolate and bar.

6 Principal Component Analysis

The base R function for PCA is called `prcomp()` and we can set "scale=TRUE/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
```

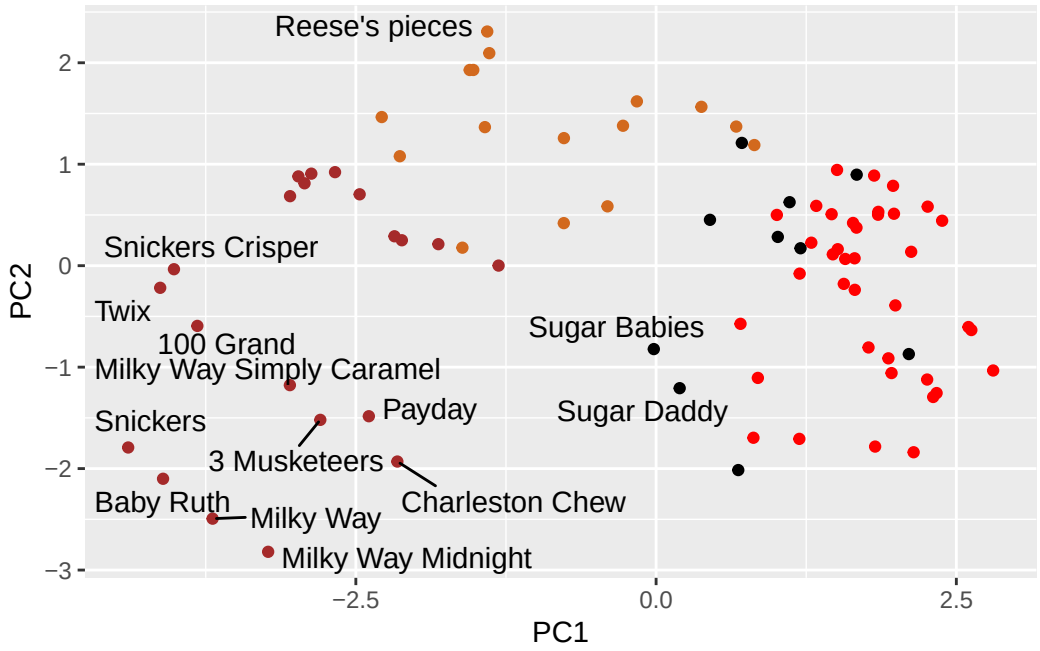
```
Standard deviation     0.74530 0.67824 0.62349 0.43974 0.39760
Proportion of Variance 0.04629 0.03833 0.03239 0.01611 0.01317
Cumulative Proportion  0.89998 0.93832 0.97071 0.98683 1.00000
```

The main result of PCA - i.e. the new PC plot (projection of candy on our new PC axis) is contained in `pca$x`

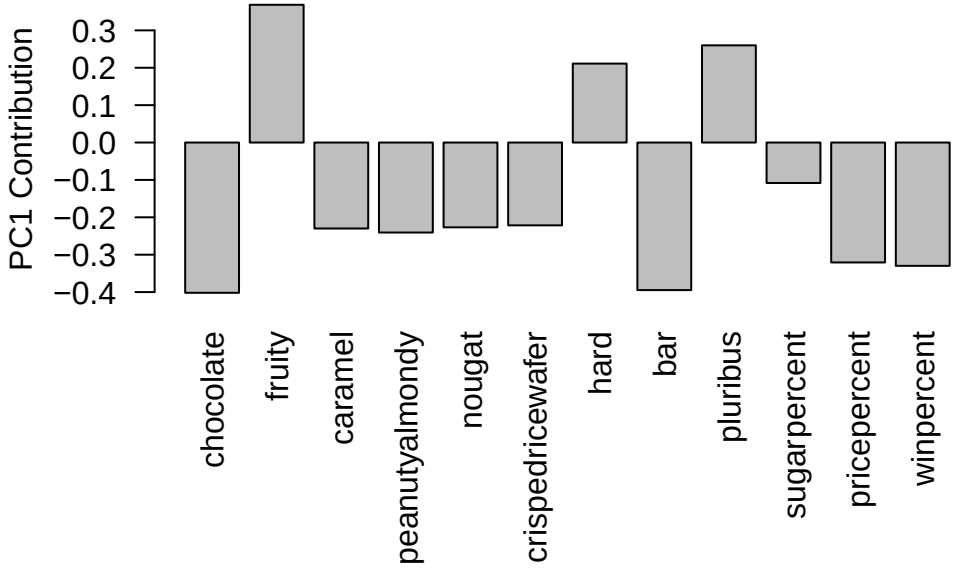
```
pc <- as.data.frame(pca$x)

ggplot(pc) +
  aes(PC1, PC2, label=rownames(pc)) +
  geom_point(col=my_cols) +
  geom_text_repel(max.overlaps = 5)
```

```
Warning: ggrepel: 71 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




```
par(mar=c(8,4,2,2))
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 are picked up strongly by PC1 in the positive direction.