Class 9: Halloween Candy Project

Yiyu

Table of contents

Background

Today we are delving into an analysis of Halloween Candy data using ggplot, dpylr, basic stats, correlation analysis, and our old friend PCA.

1. Import 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 candy types are in this dataset?

```
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

Q. How many chocolate candy types are in this dataset?

```
sum(candy$chocolate)
```

```
[1] 37
```

2. What is your favorite candy?

Q3. What is your favorite type, what is it's winpercent value?

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

```
[1] 81.64291
```

```
library(dplyr)
```

We can also use the `filter()` and `select()` functions from **dplyr**

```
candy |>
  filter(rownames(candy) == "Haribo Happy Cola") |>
  select(winpercent, sugarpercent)
```

```
                 winpercent sugarpercent
Haribo Happy Cola   34.15896        0.465
```

Q4. What is the winpercent value for "Kit Kat"? Q5. What is the winpercent value for "Tootsie Roll Snack Bars"?

```
candy |>
  filter(rownames(candy) == "Kit Kat") |>
  select(winpercent, sugarpercent)
```

```
        winpercent sugarpercent
Kit Kat    76.7686        0.313
```

```
candy |>
  filter(rownames(candy) == "Tootsie Roll Snack Bars") |>
  select(winpercent, sugarpercent)
```

```
                       winpercent sugarpercent
Tootsie Roll Snack Bars    49.6535        0.465
```

A useful function for a quick look at a new dataset is found in the **skimr** package.

```
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	

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 different "scale" or range than all others.

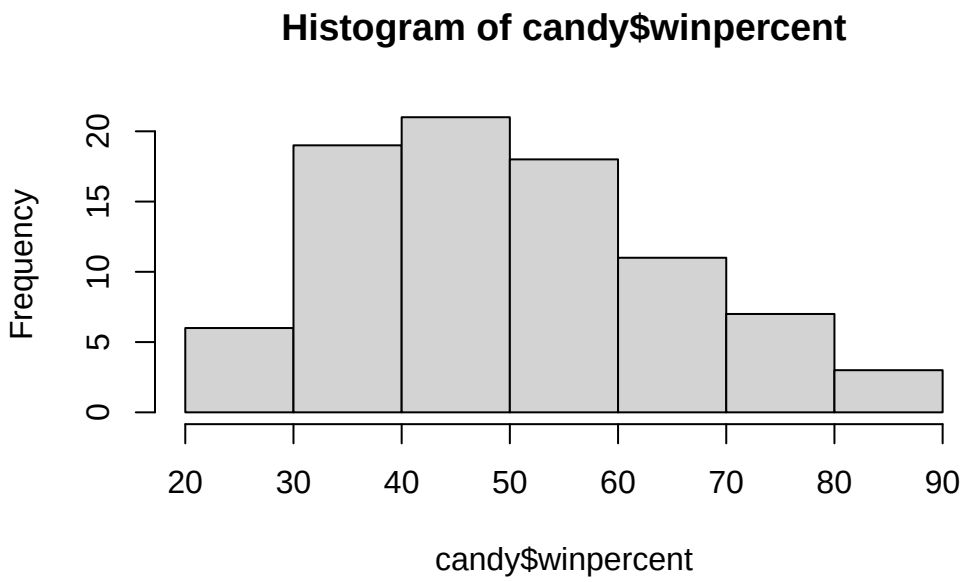
N.B We will need to scale this data before analysis like PCA for example to avoid this one variable from dominating our analysis.

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

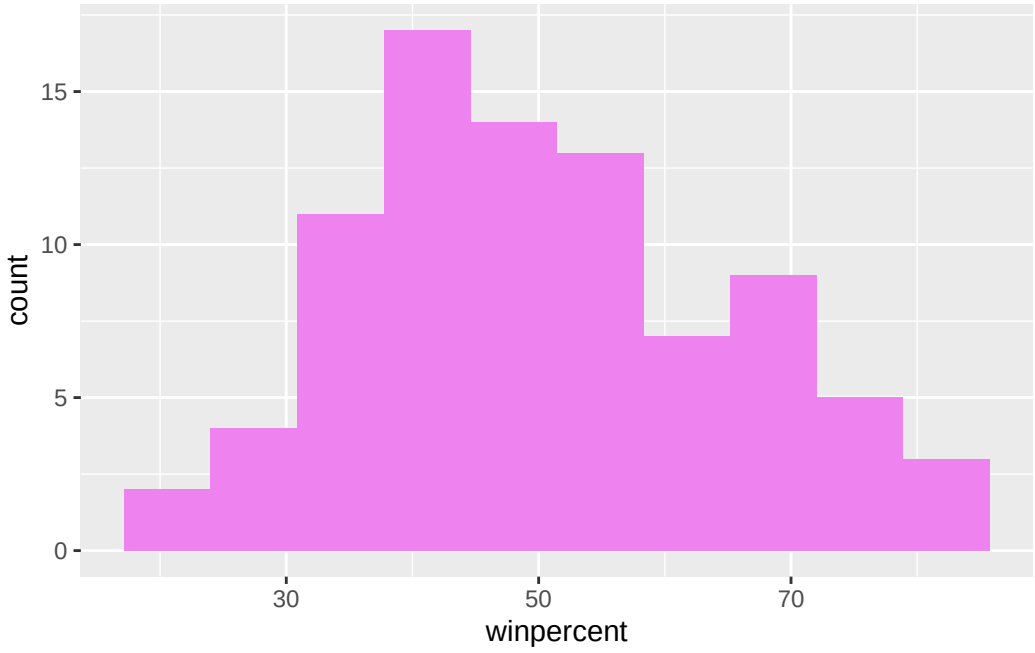
The candy does not have chocolate if candy$chocolate = 0

Q8. Plot a histogram of winpercent values using base R and ggplot.

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




```
library(ggplot2)
ggplot(candy, aes(winpercent)) +
  geom_histogram(bins = 10, fill = "violet")
```



Q9. Is the distribution of winpercent values symmetrical?

No

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

From the histogram, it looks to be below 50%. Can use the `summary()` function to get quantitative value.

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

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

Mean is slightly above, median is below 50%.

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

Step 1. Find/Extract chocolate candy rows in the dataset. Step 2. Get their `winpercent` values Step 3. Calculate their mean `winpercent` values.

Step 4. Find/extract fruity candy rows in the data set. Step 5. Get their `winpercent` values Step 6. Calculate their mean `winpercent` values

Step 7. Compare mean chocolate `winpercent` to mean fruity `winpercent` and see which one is larger.

1. Find chocolate candy

```
choc.inds <- candy$chocolate == 1
choc.candy <- candy[choc.inds,]
choc.candy
```

```
                          chocolate fruity caramel peanutyalmondy nougat
100 Grand                         1      0       1              0      0
3 Musketeers                      1      0       0              0      1
Almond Joy                        1      0       0              1      0
Baby Ruth                         1      0       1              1      1
Charleston Chew                   1      0       0              0      1
Hershey's Kisses                  1      0       0              0      0
Hershey's Krackel                 1      0       0              0      0
Hershey's Milk Chocolate          1      0       0              0      0
Hershey's Special Dark            1      0       0              0      0
Junior Mints                      1      0       0              0      0
```

Kit Kat	1	0	0	0	0
Peanut butter M&M's	1	0	0	1	0
M&M's	1	0	0	0	0
Milk Duds	1	0	1	0	0
Milky Way	1	0	1	0	1
Milky Way Midnight	1	0	1	0	1
Milky Way Simply Caramel	1	0	1	0	0
Mounds	1	0	0	0	0
Mr Good Bar	1	0	0	1	0
Nestle Butterfinger	1	0	0	1	0
Nestle Crunch	1	0	0	0	0
Peanut M&Ms	1	0	0	1	0
Reese's Miniatures	1	0	0	1	0
Reese's Peanut Butter cup	1	0	0	1	0
Reese's pieces	1	0	0	1	0
Reese's stuffed with pieces	1	0	0	1	0
Rolo	1	0	1	0	0
Sixlets	1	0	0	0	0
Nestle Smarties	1	0	0	0	0
Snickers	1	0	1	1	1
Snickers Crisper	1	0	1	1	0
Tootsie Pop	1	1	0	0	0
Tootsie Roll Juniors	1	0	0	0	0
Tootsie Roll Midgies	1	0	0	0	0
Tootsie Roll Snack Bars	1	0	0	0	0
Twix	1	0	1	0	0
Whoppers	1	0	0	0	0

	crispedricewafer	hard	bar	pluribus	sugarpercent
100 Grand	1	0	1	0	0.732
3 Musketeers	0	0	1	0	0.604
Almond Joy	0	0	1	0	0.465
Baby Ruth	0	0	1	0	0.604
Charleston Chew	0	0	1	0	0.604
Hershey's Kisses	0	0	0	1	0.127
Hershey's Krackel	1	0	1	0	0.430
Hershey's Milk Chocolate	0	0	1	0	0.430
Hershey's Special Dark	0	0	1	0	0.430
Junior Mints	0	0	0	1	0.197
Kit Kat	1	0	1	0	0.313
Peanut butter M&M's	0	0	0	1	0.825
M&M's	0	0	0	1	0.825
Milk Duds	0	0	0	1	0.302
Milky Way	0	0	1	0	0.604

Milky Way Midnight	0	0	1	0	0.732
Milky Way Simply Caramel	0	0	1	0	0.965
Mounds	0	0	1	0	0.313
Mr Good Bar	0	0	1	0	0.313
Nestle Butterfinger	0	0	1	0	0.604
Nestle Crunch	1	0	1	0	0.313
Peanut M&Ms	0	0	0	1	0.593
Reese's Miniatures	0	0	0	0	0.034
Reese's Peanut Butter cup	0	0	0	0	0.720
Reese's pieces	0	0	0	1	0.406
Reese's stuffed with pieces	0	0	0	0	0.988
Rolo	0	0	0	1	0.860
Sixlets	0	0	0	1	0.220
Nestle Smarties	0	0	0	1	0.267
Snickers	0	0	1	0	0.546
Snickers Crisper	1	0	1	0	0.604
Tootsie Pop	0	1	0	0	0.604
Tootsie Roll Juniors	0	0	0	0	0.313
Tootsie Roll Midgies	0	0	0	1	0.174
Tootsie Roll Snack Bars	0	0	1	0	0.465
Twix	1	0	1	0	0.546
Whoppers	1	0	0	1	0.872

	pricepercent	winpercent
100 Grand	0.860	66.97173
3 Musketeers	0.511	67.60294
Almond Joy	0.767	50.34755
Baby Ruth	0.767	56.91455
Charleston Chew	0.511	38.97504
Hershey's Kisses	0.093	55.37545
Hershey's Krackel	0.918	62.28448
Hershey's Milk Chocolate	0.918	56.49050
Hershey's Special Dark	0.918	59.23612
Junior Mints	0.511	57.21925
Kit Kat	0.511	76.76860
Peanut butter M&M's	0.651	71.46505
M&M's	0.651	66.57458
Milk Duds	0.511	55.06407
Milky Way	0.651	73.09956
Milky Way Midnight	0.441	60.80070
Milky Way Simply Caramel	0.860	64.35334
Mounds	0.860	47.82975
Mr Good Bar	0.918	54.52645
Nestle Butterfinger	0.767	70.73564

```
Nestle Crunch                       0.767   66.47068
Peanut M&Ms                         0.651   69.48379
Reese's Miniatures                  0.279   81.86626
Reese's Peanut Butter cup           0.651   84.18029
Reese's pieces                      0.651   73.43499
Reese's stuffed with pieces         0.651   72.88790
Rolo                                0.860   65.71629
Sixlets                             0.081   34.72200
Nestle Smarties                     0.976   37.88719
Snickers                            0.651   76.67378
Snickers Crisper                    0.651   59.52925
Tootsie Pop                         0.325   48.98265
Tootsie Roll Juniors                0.511   43.06890
Tootsie Roll Midgies                0.011   45.73675
Tootsie Roll Snack Bars             0.325   49.65350
Twix                                0.906   81.64291
Whoppers                            0.848   49.52411
```

2. Get their winpercent values

```
choc.win <- choc.candy$winpercent
choc.win
```

```
 [1] 66.97173 67.60294 50.34755 56.91455 38.97504 55.37545 62.28448 56.49050
 [9] 59.23612 57.21925 76.76860 71.46505 66.57458 55.06407 73.09956 60.80070
[17] 64.35334 47.82975 54.52645 70.73564 66.47068 69.48379 81.86626 84.18029
[25] 73.43499 72.88790 65.71629 34.72200 37.88719 76.67378 59.52925 48.98265
[33] 43.06890 45.73675 49.65350 81.64291 49.52411
```

3. Calculate their mean winpercent

```
mean(choc.win)
```

```
[1] 60.92153
```

4-6. Repeat for fruity candy.

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

```
[1] 44.11974
```

7. Compare mean chocolate `winpercent` to mean fruity `winpercent` and see which one is larger.

```
mean(choc.win) > mean(fruity.win)
```

```
[1] TRUE
```

Q12. Is this difference statistically significant?

Let's use a t-test

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

```
    Welch Two Sample t-test

data:  choc.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
```

Yes

3. Overall Candy Ratings

Q13. What are the five least liked candy types in this set? Use base R and dplyr.

```
x <- c(10, 1, 100)
sort(x)
```

```
[1]   1  10 100
```

```
order(x)
```

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

So I can use the output of `order(winpercent)` to re-arrange (or order) my whole dataset by `winpercent`

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

```
candy |>
  arrange(winpercent) |>
  head(n=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
```

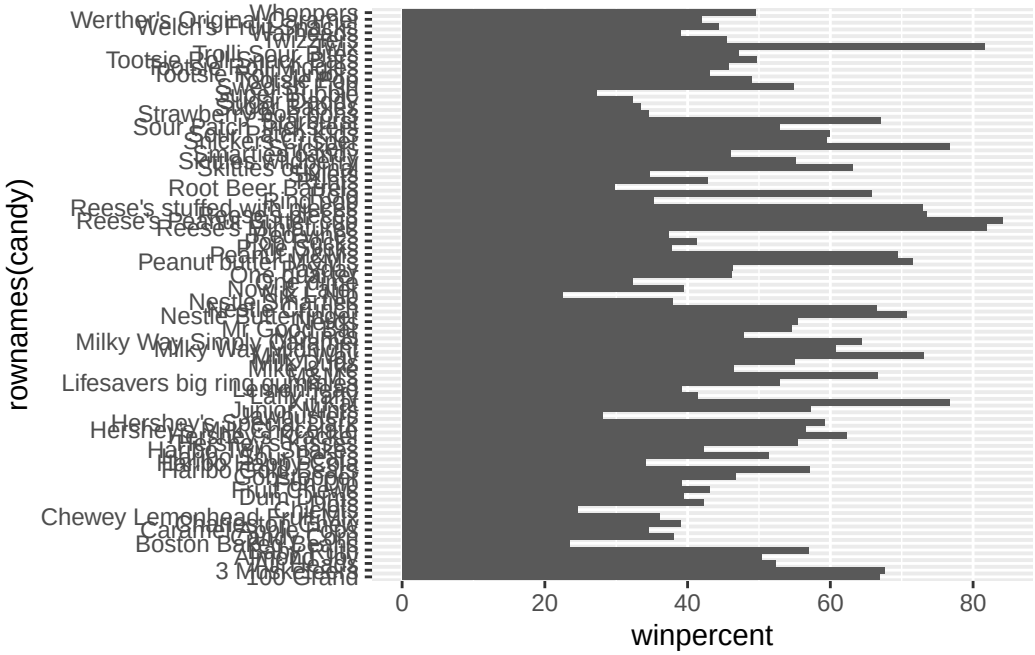
Q14. What are the top 5 all time favorite candy types out of this set?

```
candy |>
  arrange(-(winpercent)) |>
  head(n=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.

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



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

We can make the above plot better by rearrange (with the `reoder()` function) the y-axis by winpercent so the highest scoring candy is at the top and lowest at the bottom.

```
p <- ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent) ) +
  geom_col() +
  ylab("") +
  xlab("Win Percent")
```

```
ggsave("my_plot.png", height=12, width=6)
```

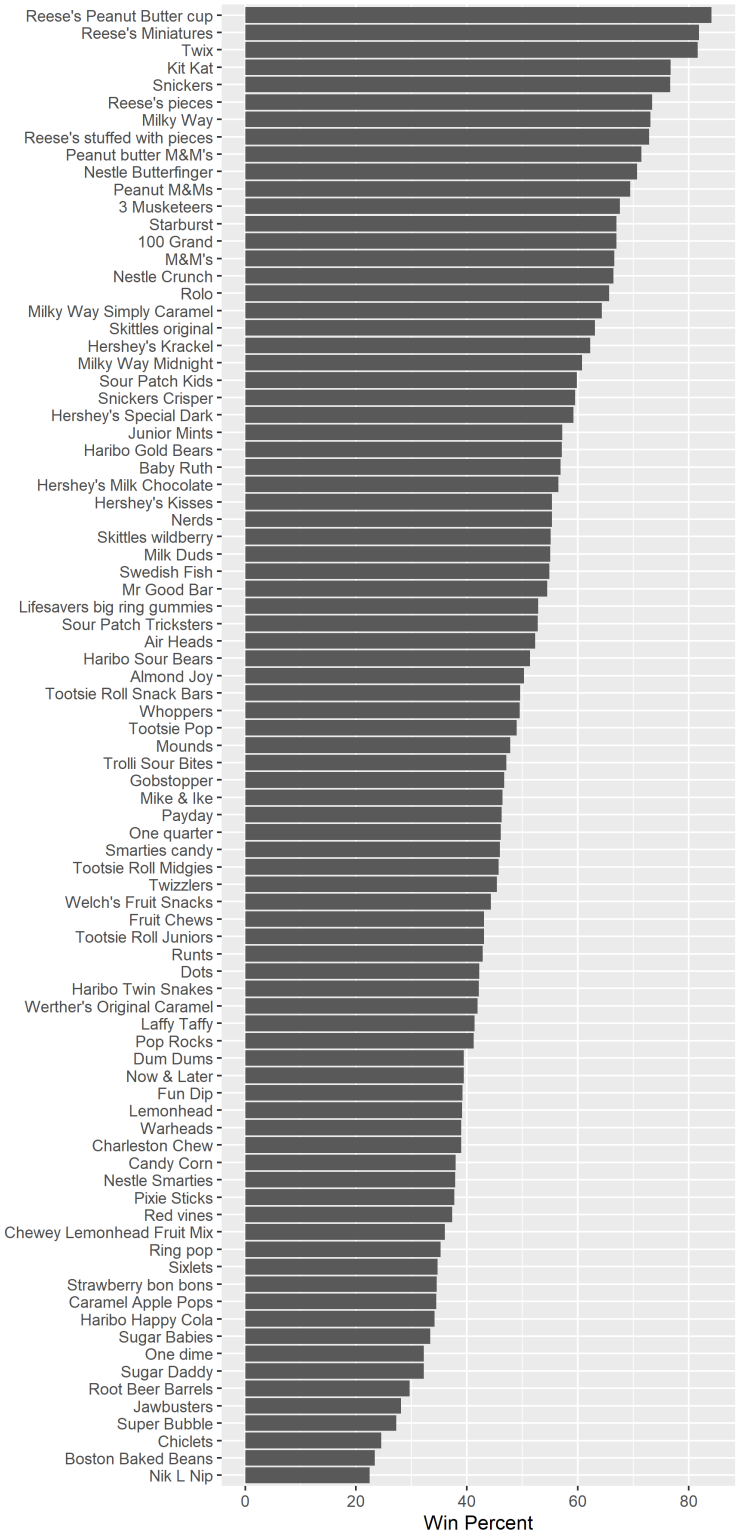
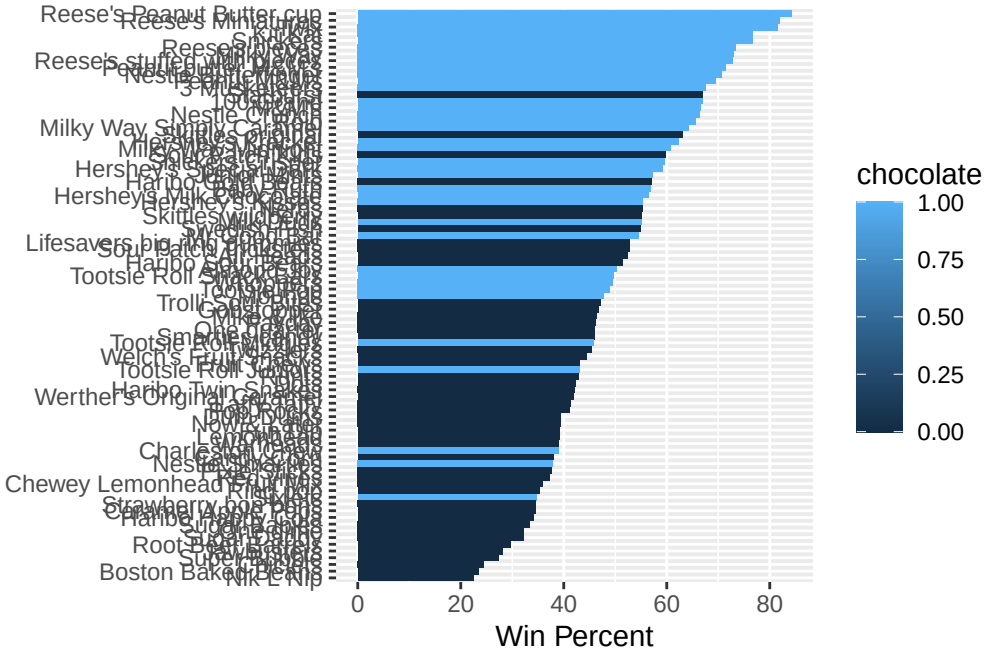


Figure 1: Larger plot for better viewing

Q. Color your bars by "chocolate"

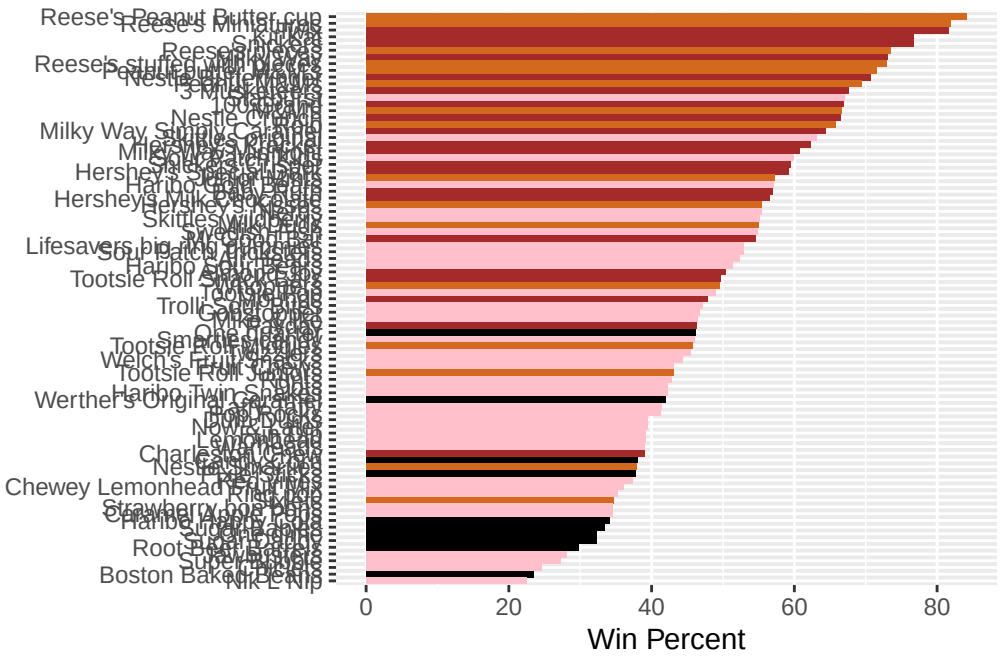
```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent) ) +
  geom_col(aes(fill = chocolate)) +
  ylab("") +
  xlab("Win Percent")
```



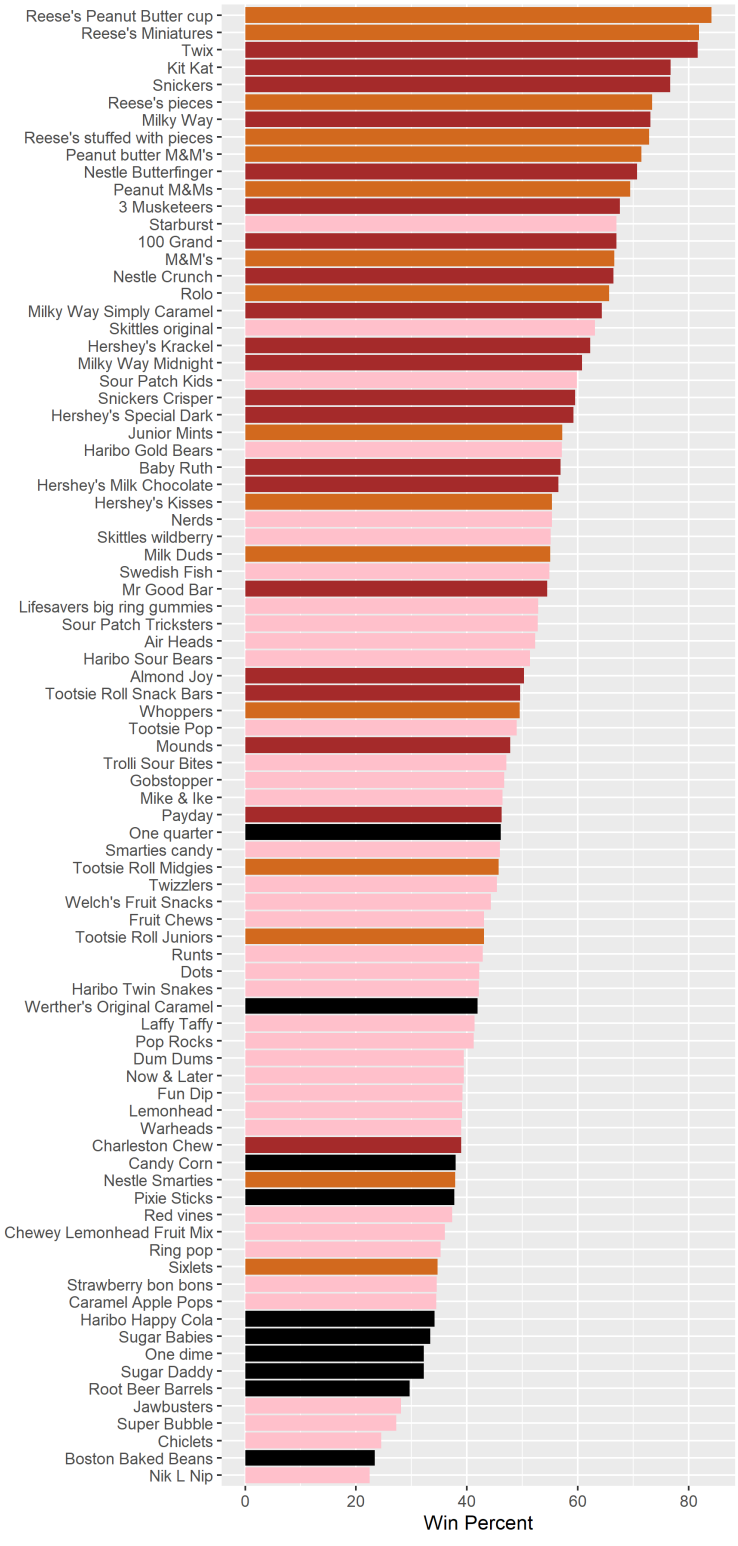
Not exactly what I want (gradient doesn't make sense). What if I want to color chocolate and fruity a specified color? To do this we need to define our own custom color vector that has the exact color mappings we want.

```
mycols <- rep("black", nrow(candy))
mycols[candy$chocolate == 1] <- "chocolate"
mycols[candy$bar == 1] <- "brown"
mycols[candy$fruity == 1] <- "pink"
```

```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent) ) +
  geom_col(fill = mycols) +
  ylab("") +
  xlab("Win Percent")
```




```
ggsave("my_color_plot.png", height=12, width=6)
```



> Q17. What is the worst

ranked chocolate candy?

Sixlets

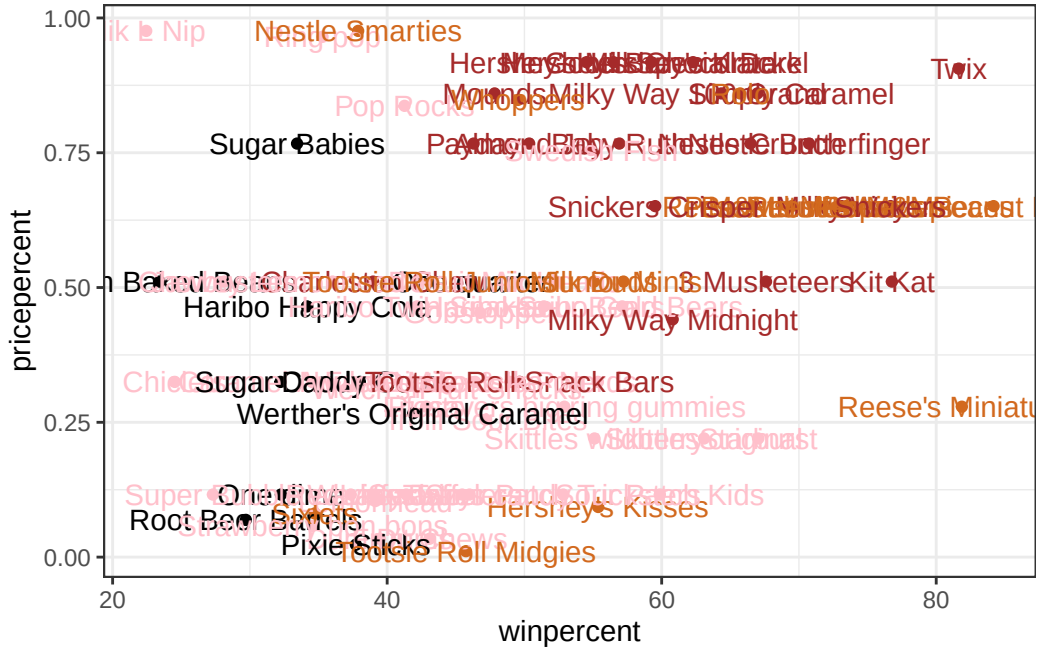
Q18. What is the best ranked fruity candy?

Starburst

4. Taking a look at pricepercent

Plot of winpercent vs pricepercent

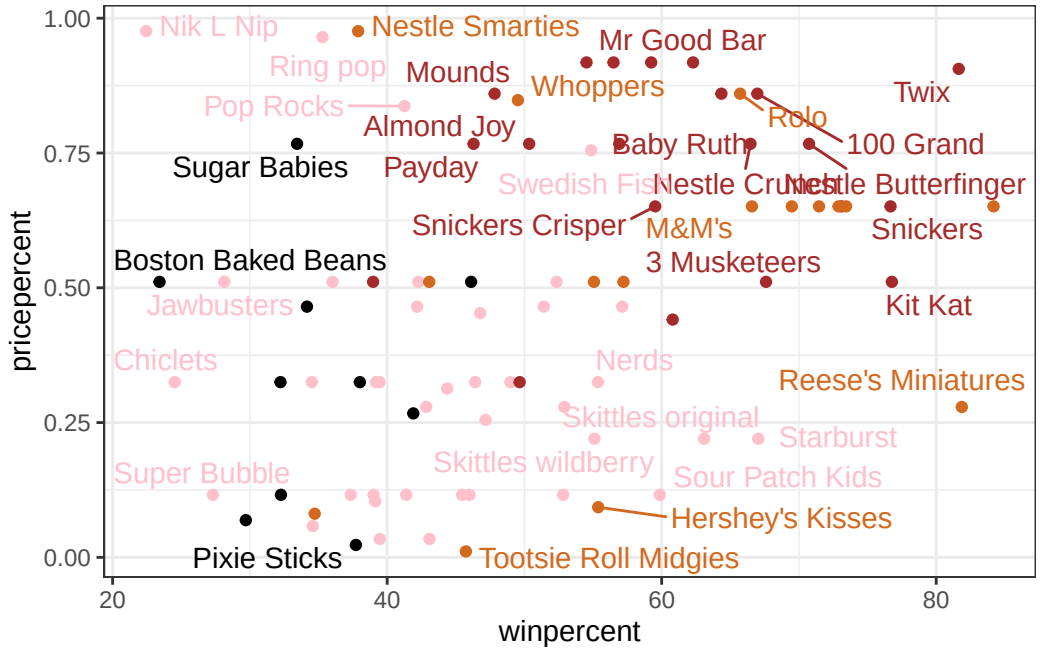
```
ggplot(candy) +
  aes(x=winpercent,
      y=pricepercent,
      label = rownames(candy)) +
  geom_point(color = mycols) +
  geom_text(col = mycols) +
  theme_bw()
```



To avoid the common problem of label or text over-plotting, we can use the **ggrepel** package like so:

```
library(ggrepel)
ggplot(candy) +
  aes(x=winpercent,
      y=pricepercent,
      label = rownames(candy)) +
  geom_point(color = mycols) +
  geom_text_repel(col = mycols) +
  theme_bw()
```

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



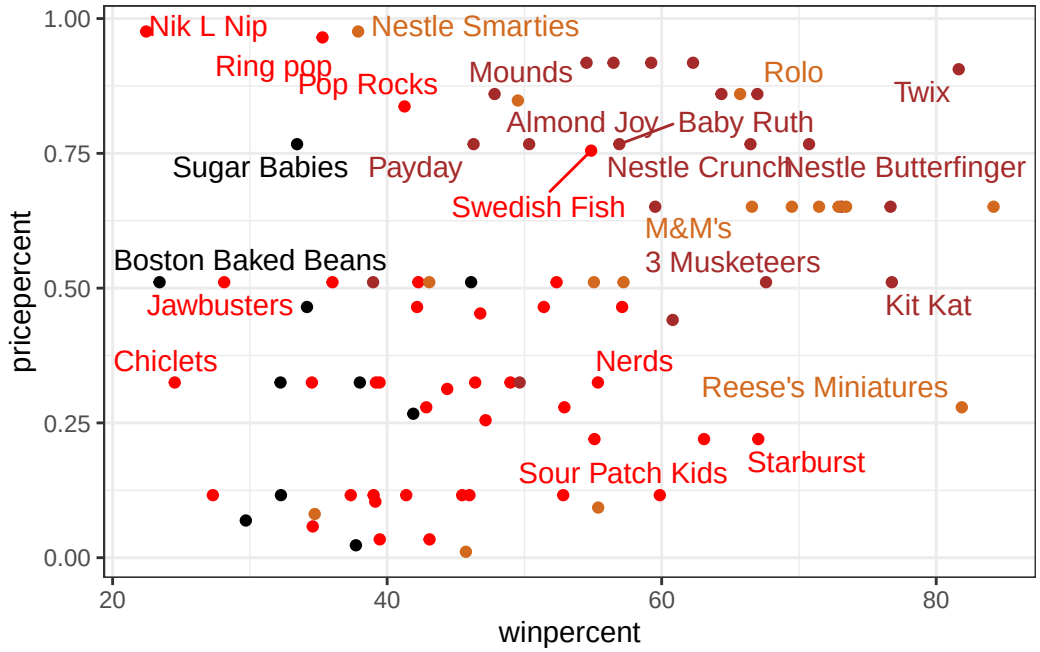
We can control the amount of labels visible by setting different `max.overlaps` values:

```
#Change pink to be red for fruity candy for better visibility
mycols[candy$fruity == 1] <- "red"

ggplot(candy) +
  aes(x=winpercent,
      y=pricepercent,
      label = rownames(candy)) +
  geom_point(color = mycols) +
```

```
  geom_text_repel(col = mycols, max.overlaps = 8) +
  theme_bw()
```

```
Warning: ggrepel: 61 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



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?

Reese's Miniatures

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

Some points too close to tell top 5, so used dyplr code to confirm 5 most expensive.

```
candy |>
  arrange(-(pricepercent)) |>
  head(n=5)
```

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

```
Nestle Smarties                   1      0       0              0      0
Ring pop                          0      1       0              0      0
Hershey's Krackel                 1      0       0              0      0
Hershey's Milk Chocolate          1      0       0              0      0
                        crispedricewafer hard bar pluribus sugarpercent
Nik L Nip                              0    0   0        1        0.197
Nestle Smarties                        0    0   0        1        0.267
Ring pop                               0    1   0        0        0.732
Hershey's Krackel                      1    0   1        0        0.430
Hershey's Milk Chocolate               0    0   1        0        0.430
                        pricepercent winpercent
Nik L Nip                      0.976   22.44534
Nestle Smarties                0.976   37.88719
Ring pop                       0.965   35.29076
Hershey's Krackel              0.918   62.28448
Hershey's Milk Chocolate       0.918   56.49050
```

Nik L Nip, Ring Pop, Nestles Smarties, Hershey's Krackel, Hershey's Milk Chocolate Least popular is Nik L Nip

5. Exploring the correlation structure

The main function for correlation analysis in base R is called `cor()`

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

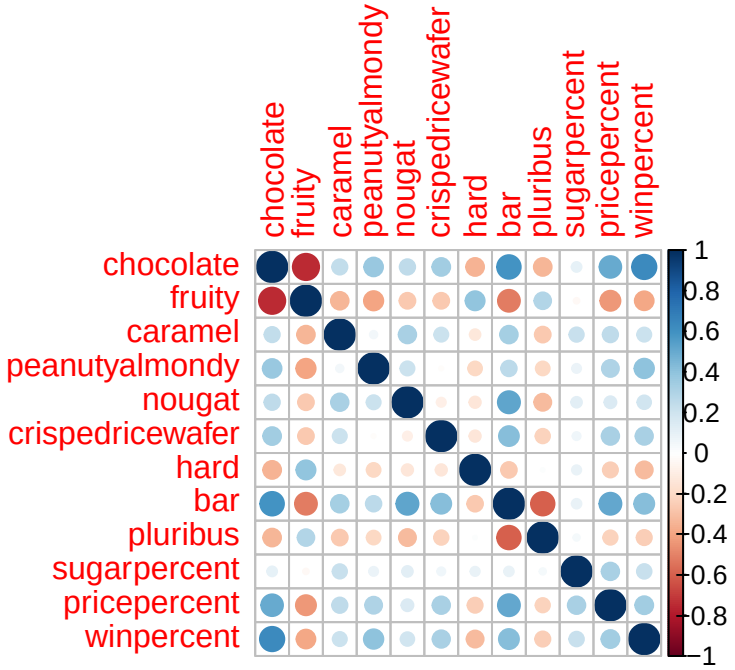
```
                  chocolate     fruity     caramel peanutyalmondy      nougat
chocolate         1.0000000 -0.7417211  0.24987535     0.37782357  0.25489183
fruity           -0.7417211  1.0000000 -0.33548538    -0.39928014 -0.26936712
caramel           0.2498753 -0.3354854  1.00000000     0.05935614  0.32849280
peanutyalmondy    0.3778236 -0.3992801  0.05935614     1.00000000  0.21311310
nougat            0.2548918 -0.2693671  0.32849280     0.21311310  1.00000000
crispedricewafer  0.3412098 -0.2693671  0.21311310    -0.01764631 -0.08974359
                  crispedricewafer       hard        bar   pluribus sugarpercent
chocolate              0.34120978 -0.3441769  0.5974211 -0.3396752   0.10416906
fruity                -0.26936712  0.3906775 -0.5150656  0.2997252  -0.03439296
caramel                0.21311310 -0.1223551  0.3339600 -0.2695850   0.22193335
peanutyalmondy        -0.01764631 -0.2055566  0.2604196 -0.2061093   0.08788927
nougat                -0.08974359 -0.1386750  0.5229764 -0.3103388   0.12308135
crispedricewafer       1.00000000 -0.1386750  0.4237509 -0.2246934   0.06994969
```

```
                  pricepercent winpercent
chocolate            0.5046754  0.6365167
fruity              -0.4309685 -0.3809381
caramel              0.2543271  0.2134163
peanutyalmondy       0.3091532  0.4061922
nougat               0.1531964  0.1993753
crispedricewafer     0.3282654  0.3246797
```

```
library(corrplot)
```

```
corrplot 0.95 loaded
```

```
corrplot(cij)
```



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

Fruity and Chocolate candies

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

Chocolate and winpercent, chocolate and bar

6. Principal Component Analysis (PCA)

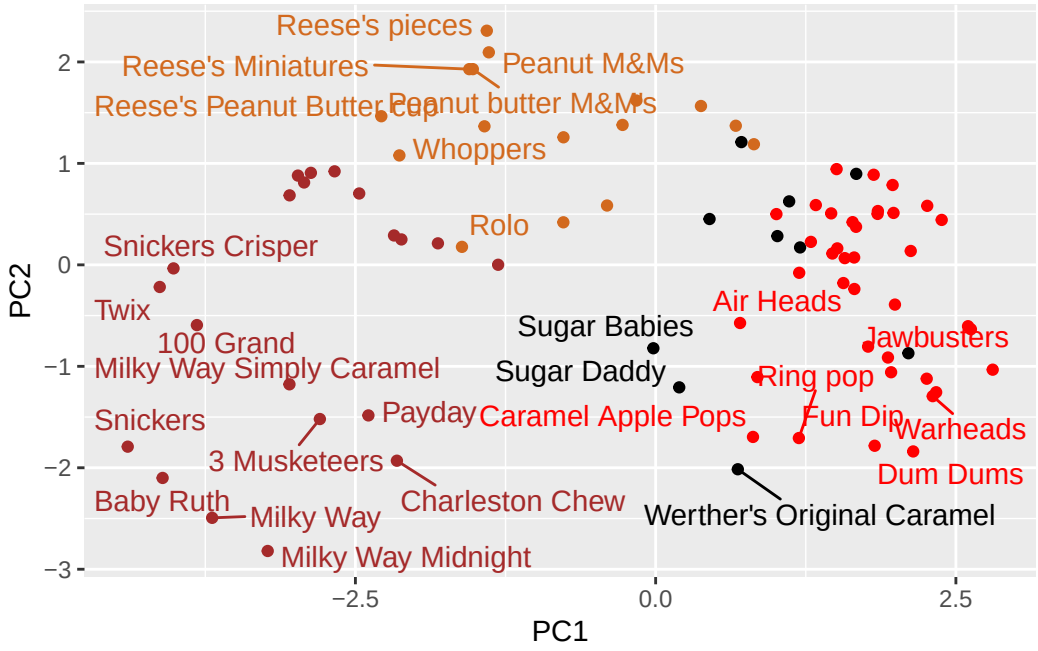
We can use our old friend `prcomp()` function with `scale = TRUE`(recall winpercent was on a different scale. need to set scale to true so it doesn't bias the analysis)

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

Let's make our main results figures, first our score plot (PC Plot)

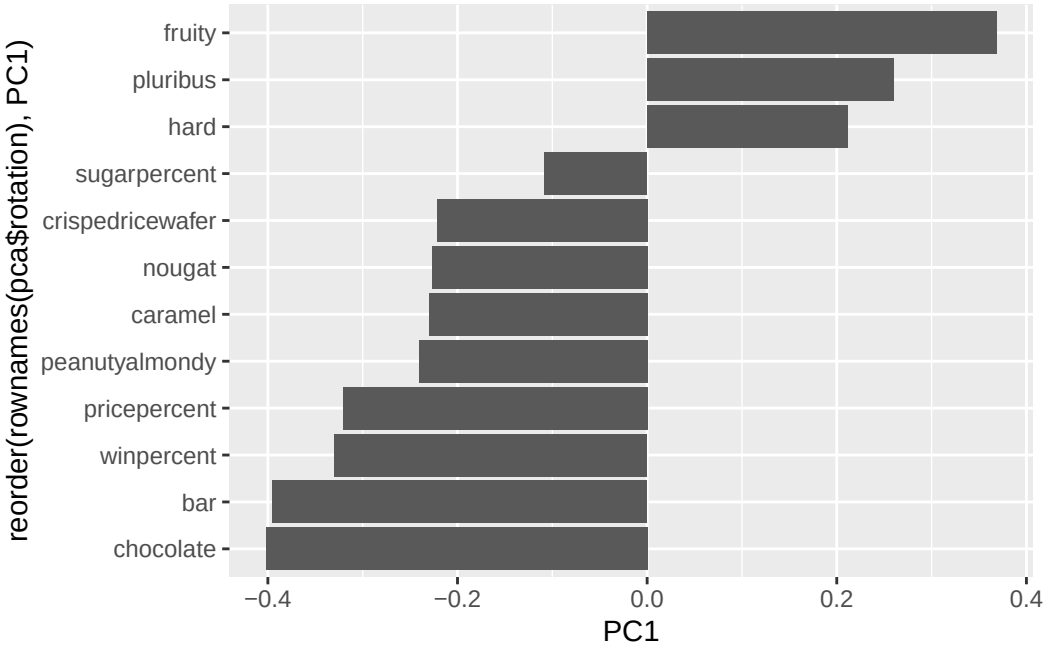
```
ggplot(pca$x) +
  aes(PC1, PC2, label = rownames(candy)) +
  geom_point(col = mycols) +
  geom_text_repel(col = mycols, max.overlaps = 8)
```

```
Warning: ggrepel: 57 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Let's look at how the original variables contribute to our new PCs - this is often called the variable "loadings" or contributions:

```
ggplot(pca$rotation) +
  aes(x=PC1,
      y=reorder(rownames(pca$rotation), PC1)) +
  geom_col()
```



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

Fruity, hard, pluribus. Yes they make sense because fruity candy tend to come as hard candies and in multiples (such as skittles)