Class 9: Halloween Candy Project

Shivani Lakkaraju

Table of contents

Background

Today we are delving into an analysis of Halloween candy data using ggplot, dplyr, basic stats, correlation analysis, and PCA.

Import 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?

```
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

Q3. how many chocolate candy types in this dataset?

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

```
[1] 37
```

What is your favorite candy?

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

```
[1] 81.64291
```

```
library(dplyr)
```

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

```
candy |>
  filter(rownames(candy)=="Almond Joy") |>
  select(winpercent, sugarpercent)
```

```
           winpercent sugarpercent
Almond Joy   50.34755        0.465
```

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

```
#library("skimr")
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 - `winpercent` column is on a different scale or range than all the others.

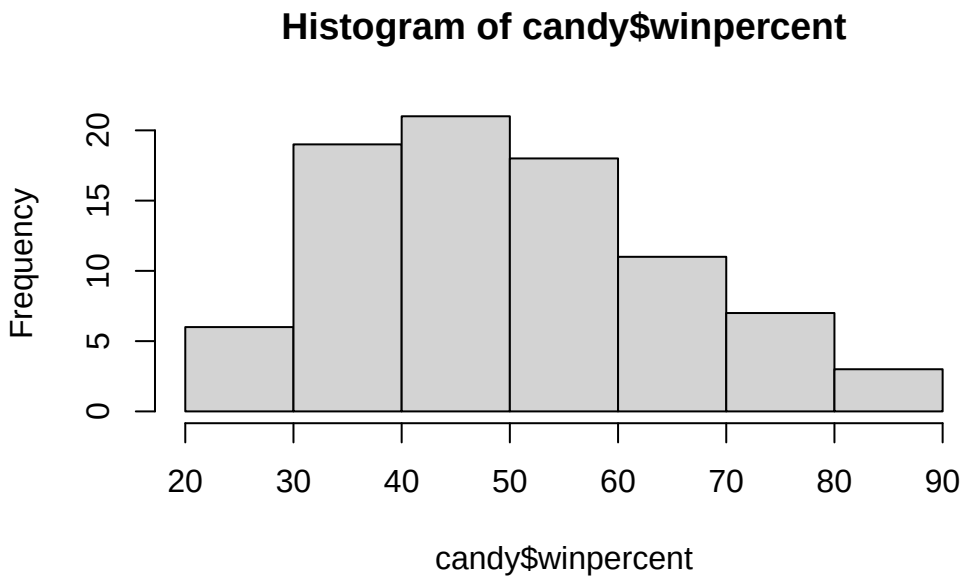
N.B I will scale this data before analysis like PCA for example to avoid this one variable dominating our analysis.

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

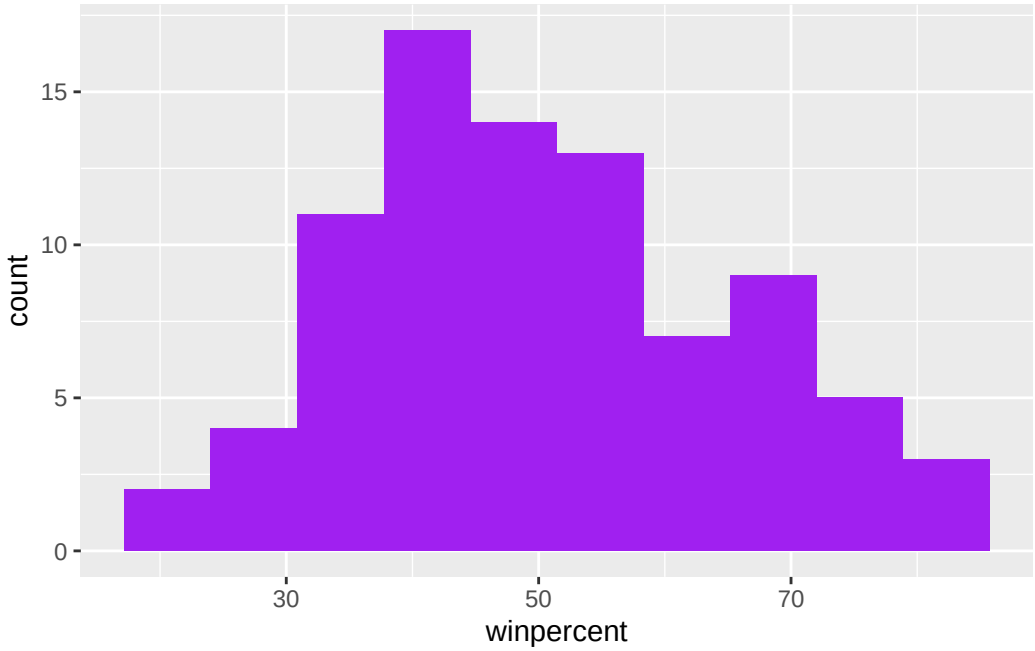
zero means no chocolate and one means it contains chocolate.

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

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




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



Q9. Is the distribution of winpercent values symmetrical?

no!

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

below

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

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

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

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

```
[1] 60.92153
```

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

```
[1] 44.11974
```

Q12. Is this difference statistically significant?

yes!

```
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
```

3. Overall Candy Rankings

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

Can use the output of `order(winpercent)` to rearrange the dataset by `winpercent`.

```
ord.inds <- order(candy$winpercent)
head(candy[ord.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
```

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

```
                  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
```

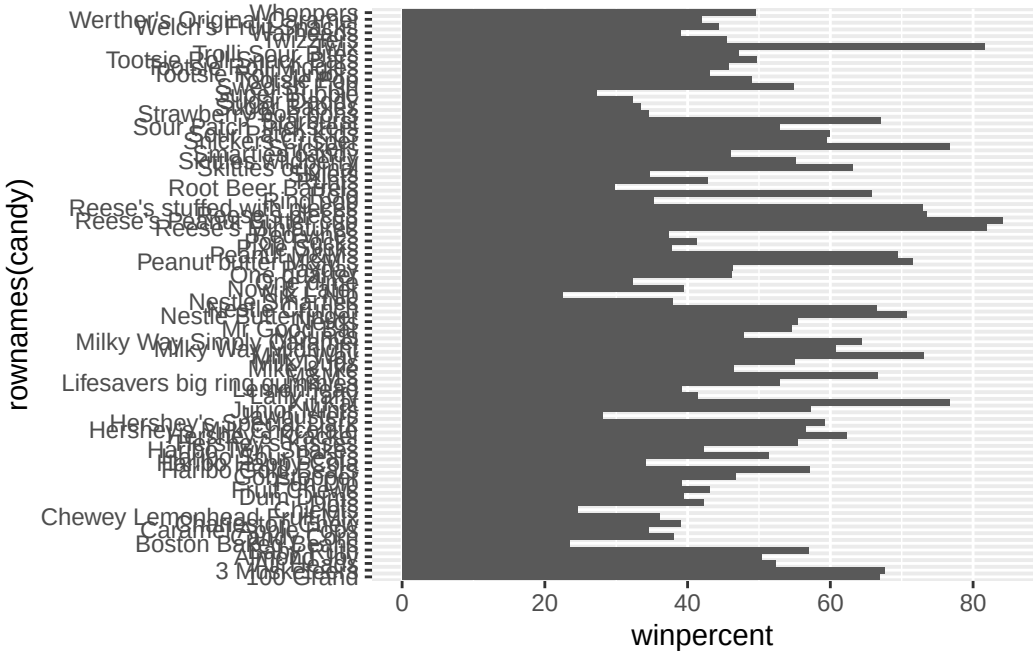
Q14. What are the top 5 all time favorite candy types out of this set?

```
candy |>
  arrange(-winpercent) |>
  head()
```

```
                         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
Reese's pieces                    1      0       0              1      0
                         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
Reese's pieces                           0    0   0        1        0.406
                         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
Reese's pieces                   0.651   73.43499
```

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

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



Improved plot with `reorder()`

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

```
ggsave("my_plot.png", height=12, width=5)
```

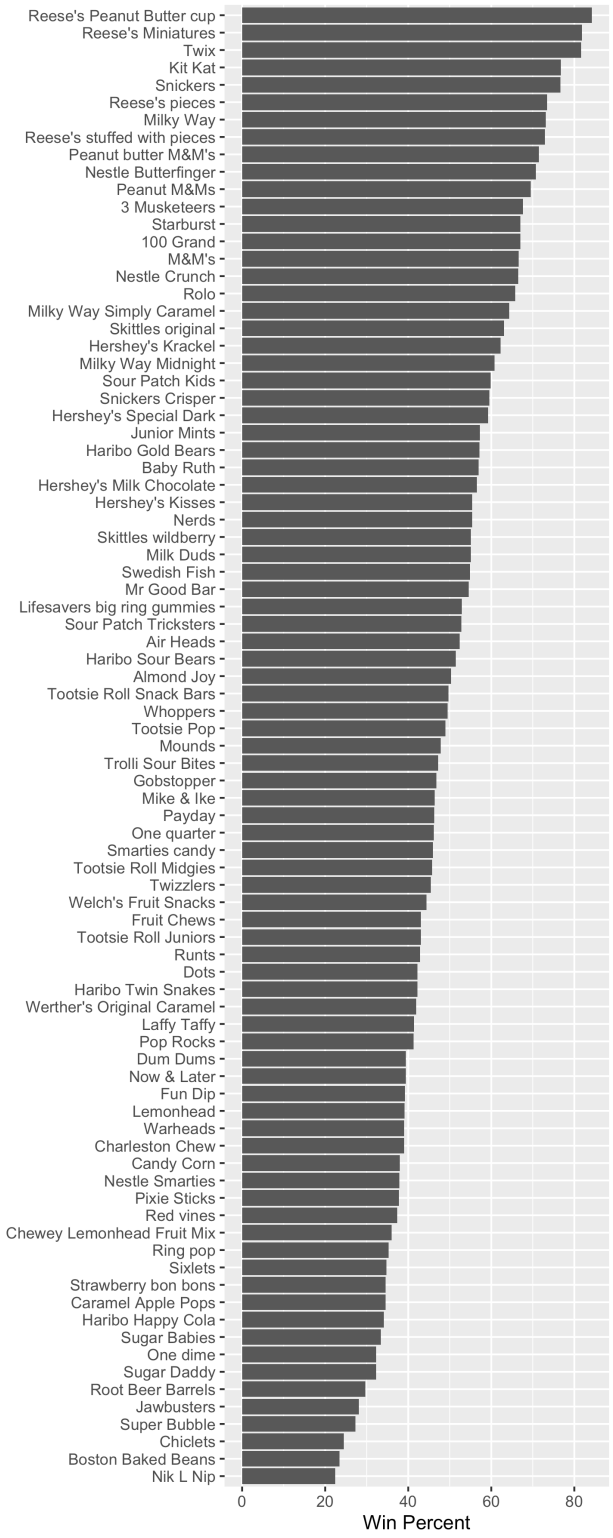
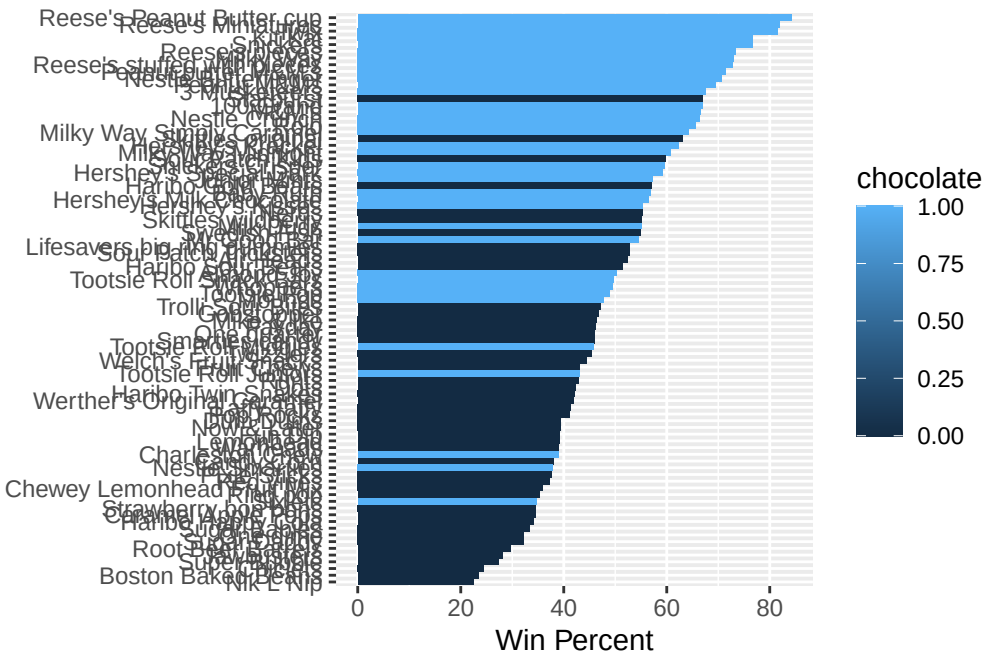


Figure 1: Fig. X

Q. color bars by "chocolate"

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

p
```



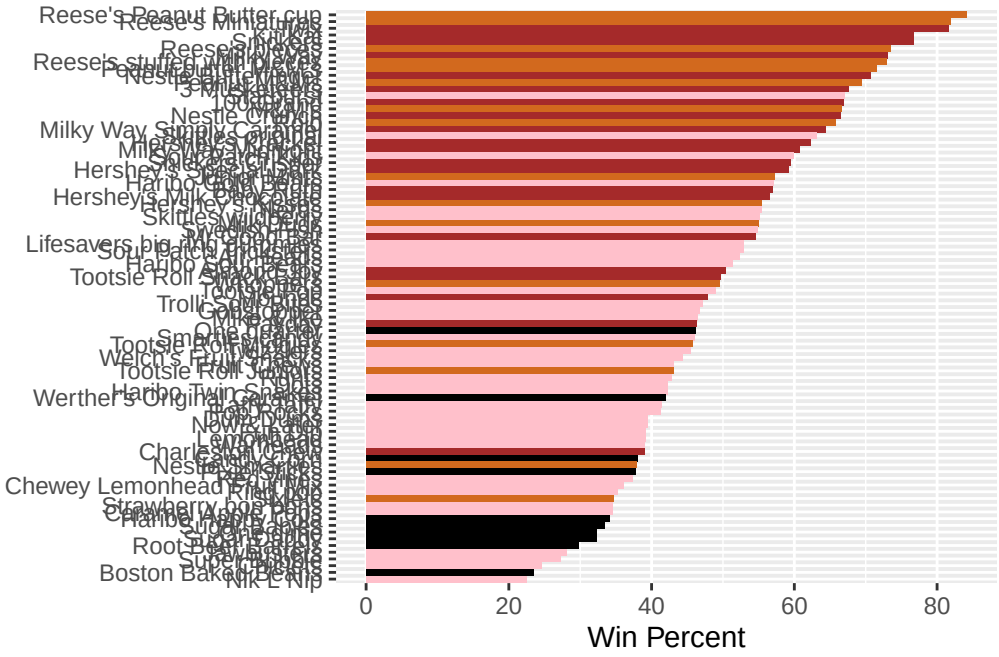
Define our own custom color vector that has the exact color mappings we want:

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

mycols
```

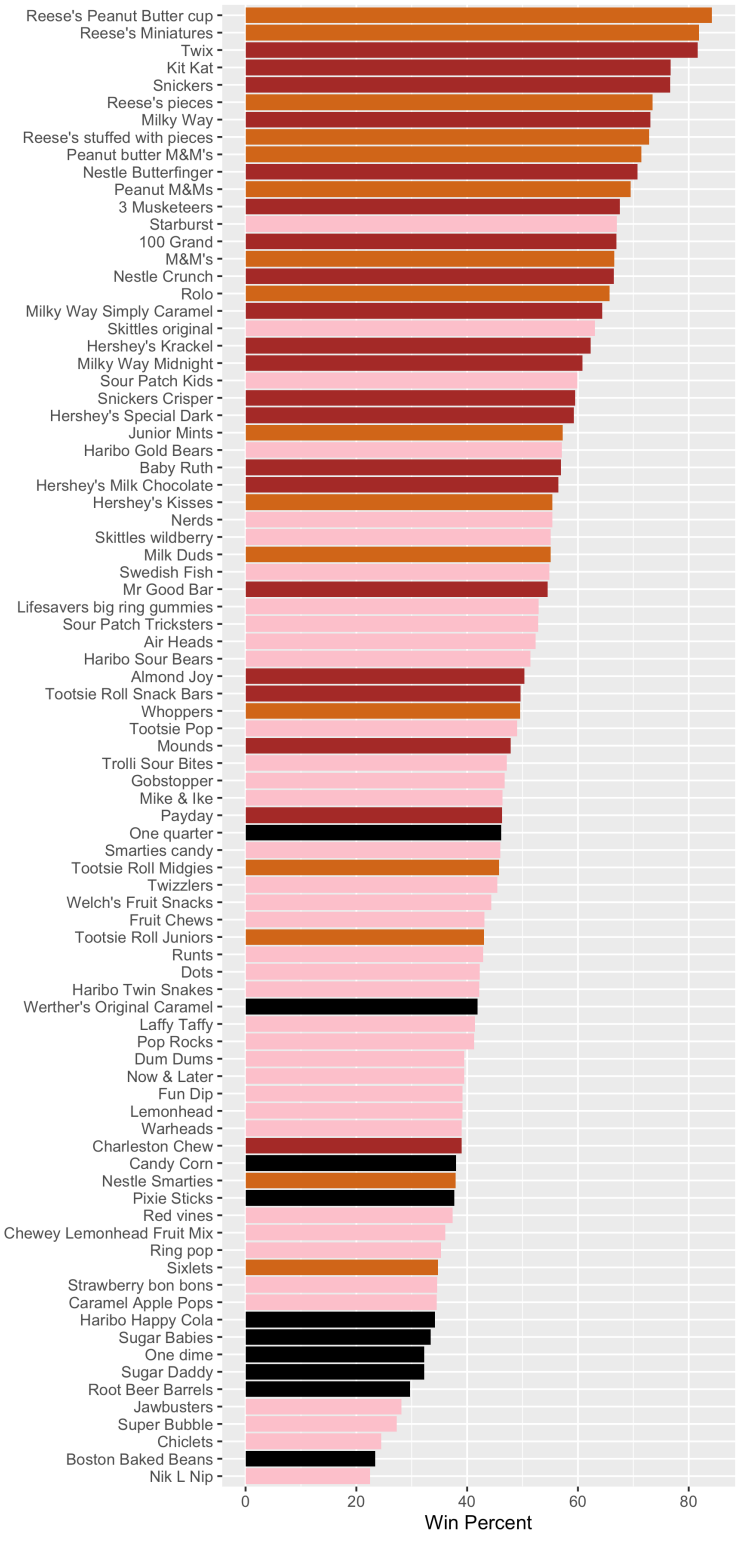
```
 [1] "brown"     "brown"     "black"     "black"     "pink"      "brown"
 [7] "brown"     "black"     "black"     "pink"      "brown"     "pink"
[13] "pink"      "pink"      "pink"      "pink"      "pink"      "pink"
[19] "pink"      "black"     "pink"      "pink"      "chocolate" "brown"
[25] "brown"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

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

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

p
```




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



Now, for the first time, using

this plot we can answer questions like:

Q17. What is the worst ranked chocolate candy?

Sixlets

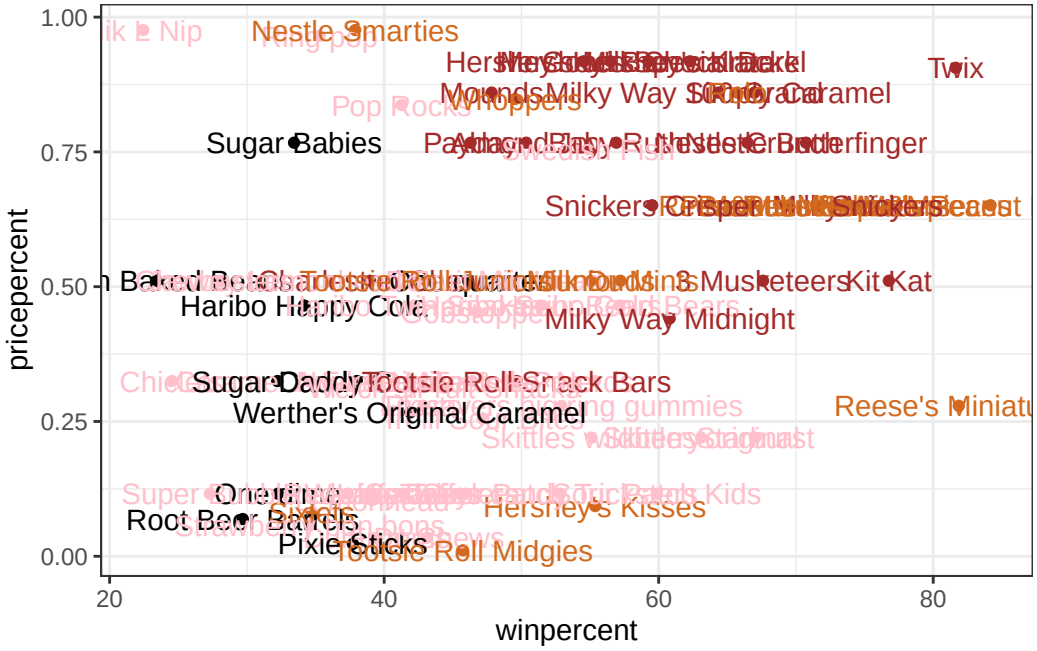
Q18. What is the best ranked fruity candy?

Starburst

Taking a look at pricepercent

plot of winpercent vs the pricepercent

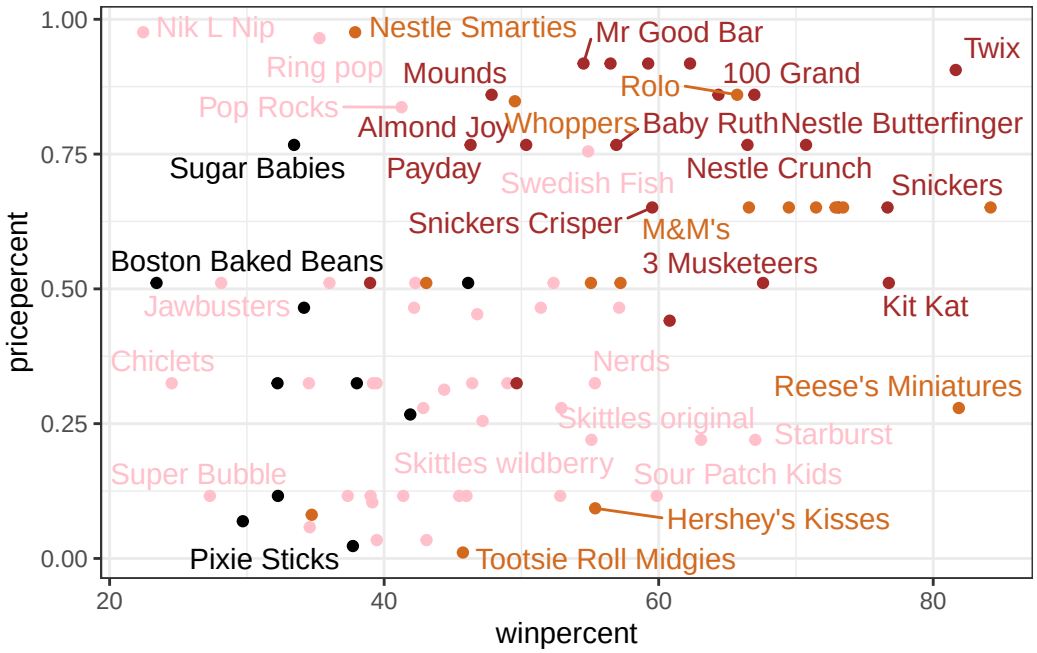
```
ggplot(candy, aes(winpercent, pricepercent, label=rownames(candy))) + geom_point(col=mycols)
```



To avoid overplotting, we use **ggrepel** package :

```
library(ggrepel)

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

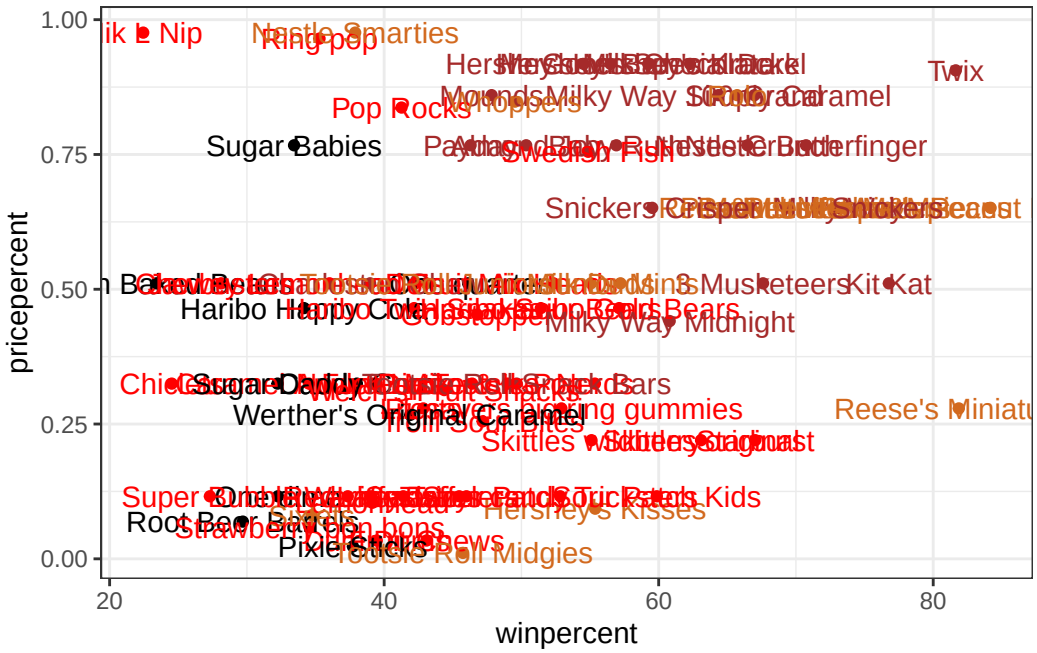
```
Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Control amount of visible labels by setting different `max.overlaps` value:

```
#change pink to red for fruity:
mycols[candy$fruity==1] <- "red"

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

```
Warning in geom_text(col = mycols, max.overlaps = 8): Ignoring unknown
parameters: `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?

Reeses miniatures

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

Nik L nip is least popular, Nestle Smarties, Ring pop, Hershey's Krackel, Hershey's Milk Chocolate

```
candy |>
  arrange(-pricepercent) |>
  head()
```

```
                        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
Hershey's Special Dark          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
Hershey's Special Dark               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
Hershey's Special Dark          0.918   59.23612
```

5 Exploring the correlation structure

The main function for correlation analysis in base R: `corr()`:

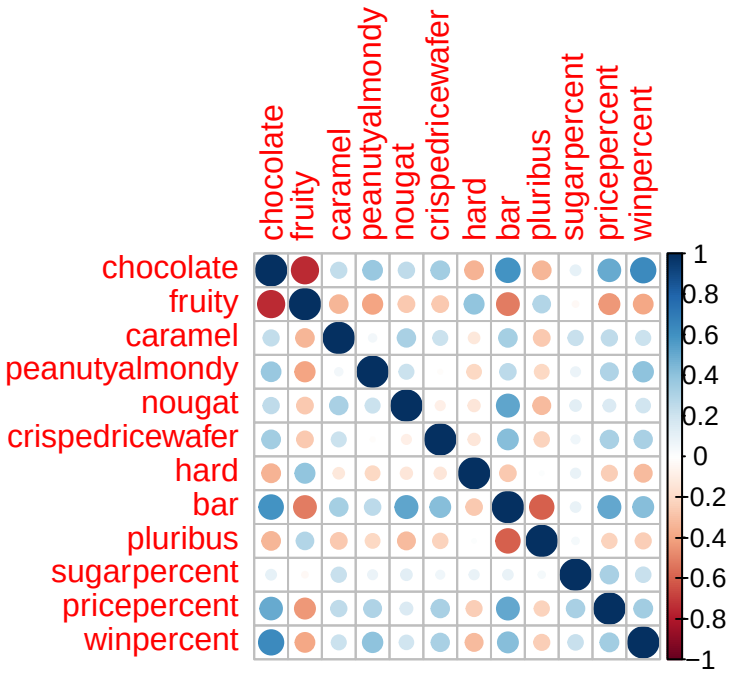
```
cij <- cor(candy)
head(cij)
```

```
                  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
```

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



6. Principal Component Analysis

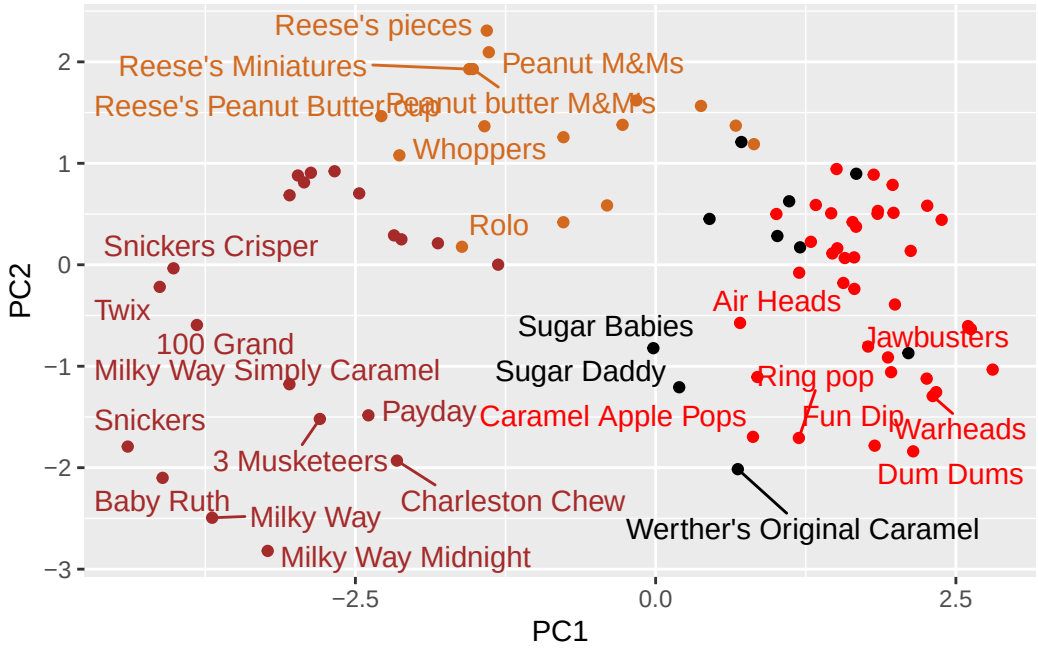
use `prcom()` function remembering to set the scale=TRUE argument.

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

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

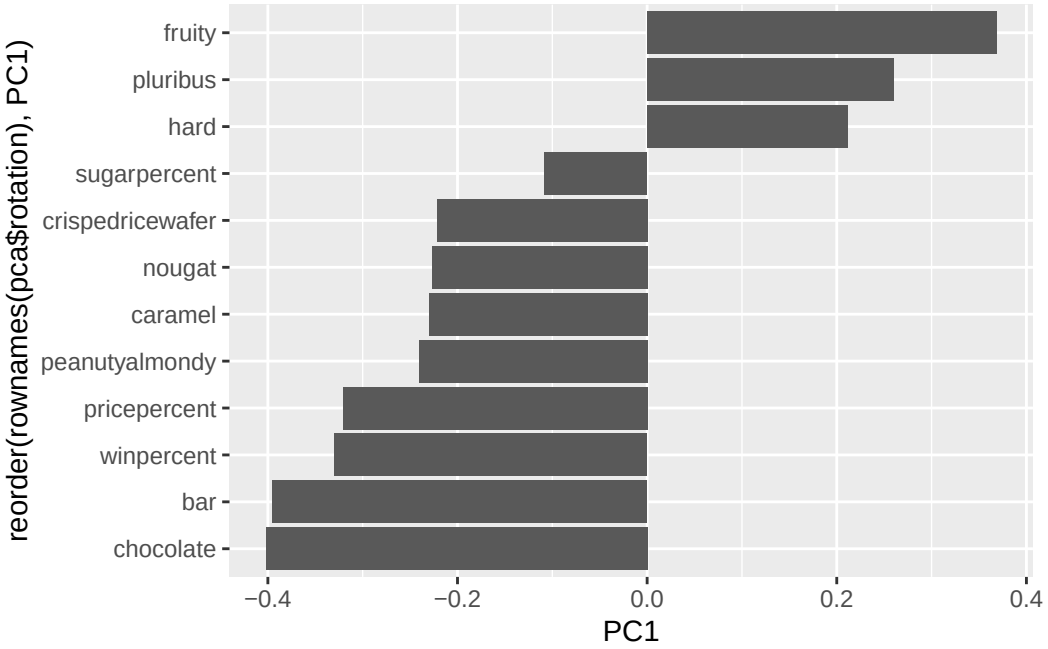
```
ggplot(pca$x, aes(PC1, PC2, label=rownames(candy))) + geom_point(col=mycols) + geom_text_repe
```

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



look at variable loadings

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


reorder(rownames(pca$rotation), PC1)
fruity
pluribus
hard
sugarpercent
crispedricewafer
nougat
caramel
peanutyalmondy
pricepercent
winpercent
bar
chocolate
−0.4
−0.2
0.0
0.2
0.4
PC1