# Candy

## Table of contents

### Background

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

### Import the data

```r
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
```

Q. How many candy types are in this dataset?

```r
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

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

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

```
[1] 37
```

## What is you favorite candy?

```r
candy["Swedish Fish", "winpercent"]
```

```
[1] 54.86111
```

```r
candy["Swedish Fish",]$winpercent
```

```
[1] 54.86111
```

```r
library(dplyr)
```

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

```
candy |>
  filter(rownames(candy) == "Swedish Fish") |>
select(winpercent, sugarpercent)
```

```
             winpercent sugarpercent
Swedish Fish   54.86111        0.604
```

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

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

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

```
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 | |

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| 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. Is there any variable/column that looks to be on a different scale to the majority of the other columns in the dataset?

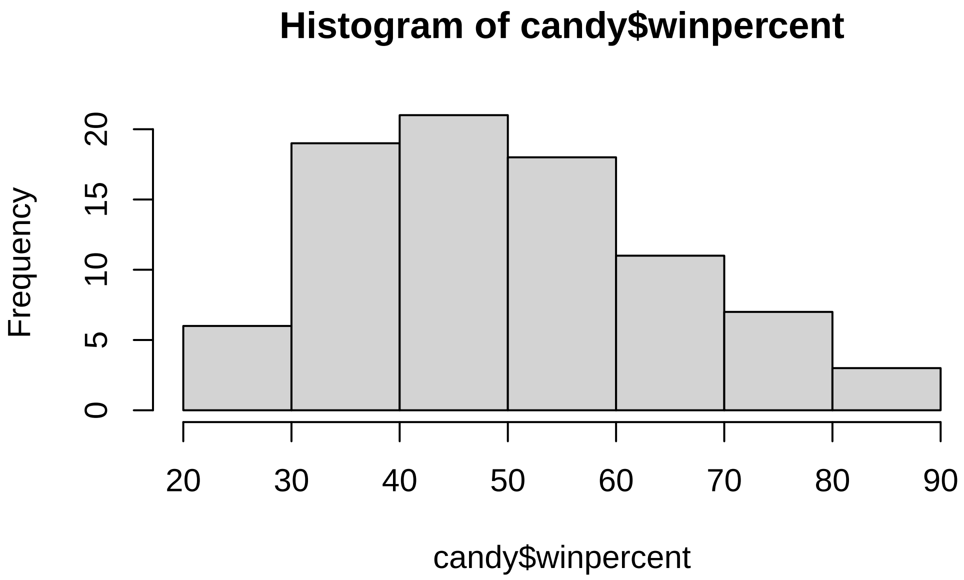
Yes, the `winpercent` column is on a different scale/range than all the others. We will need to scale this data before aalysis like PCA to avoid this one variable dominating our analysis.

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

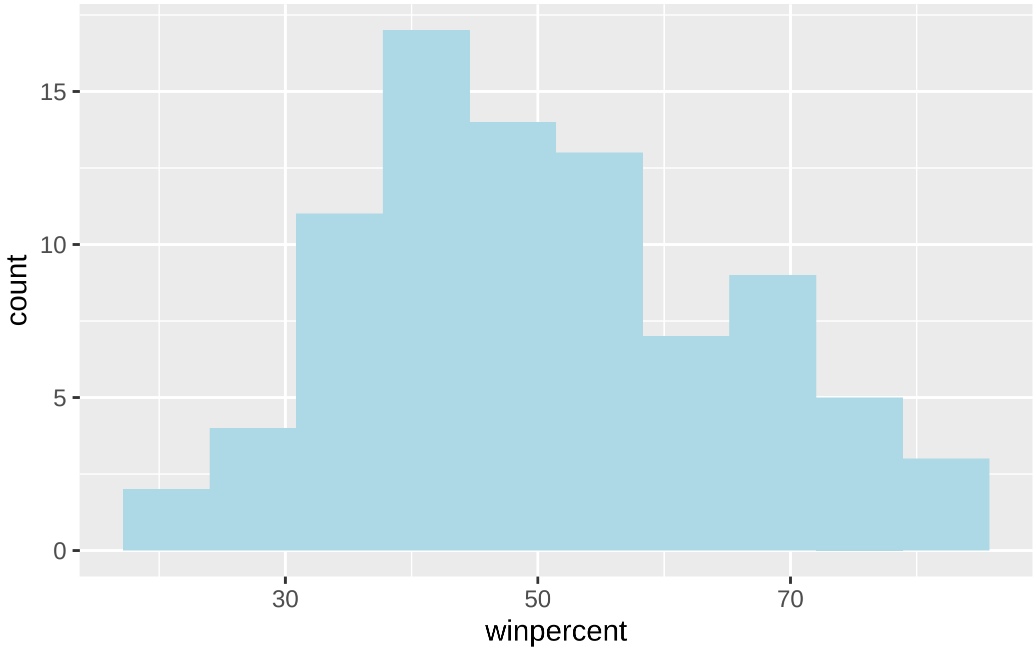
O means the candy does not have chocolate i it, while 1 means that the candy does have chocolate.

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

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



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



> Q. Is the distribution of winpercent values symmetrical?

No

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

From the histogram, the center looks below 50%

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

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

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

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

Step 4. Find fruity candy rows in the dataset Step 5. Get their `winpercent` values Step 6. Calculate their mean value

Step 7. Compare the two means you found

```r
# Step 1. Find chocolate candy rows in the dataset
choc.inds <- candy$chocolate == 1
choc.candy <- candy[choc.inds, ]

#Step 2. Get their `winpercent` values
choc.win <- choc.candy$winpercent

# Step 3. Calculate their mean value
mean(choc.win)
```

```
[1] 60.92153
```

```r
# Step 4. Find fruity candy rows in the dataset
fruit.inds <- candy$fruity == 1
fruit.candy <- candy[fruit.inds, ]

#Step 5. Get their `winpercent` values
fruit.win <- fruit.candy$winpercent

# Step 6. Calculate their mean value
mean(fruit.win)
```

```
[1] 44.11974
```

> Q. Is this difference statistically significant?

Let's use a t-test

```r
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 
```

## Overall Candy Rankings

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

```
sort(candy$winpercent)
```

```
 [1] 22.44534 23.41782 24.52499 27.30386 28.12744 29.70369 32.23100 32.26109
 [9] 33.43755 34.15896 34.51768 34.57899 34.72200 35.29076 36.01763 37.34852
[17] 37.72234 37.88719 38.01096 38.97504 39.01190 39.14106 39.18550 39.44680
[25] 39.46056 41.26551 41.38956 41.90431 42.17877 42.27208 42.84914 43.06890
[33] 43.08892 44.37552 45.46628 45.73675 45.99583 46.11650 46.29660 46.41172
[41] 46.78335 47.17323 47.82975 48.98265 49.52411 49.65350 50.34755 51.41243
[49] 52.34146 52.82595 52.91139 54.52645 54.86111 55.06407 55.10370 55.35405
[57] 55.37545 56.49050 56.91455 57.11974 57.21925 59.23612 59.52925 59.86400
[65] 60.80070 62.28448 63.08514 64.35334 65.71629 66.47068 66.57458 66.97173
[73] 67.03763 67.60294 69.48379 70.73564 71.46505 72.88790 73.09956 73.43499
[81] 76.67378 76.76860 81.64291 81.86626 84.18029
```

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

```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
```

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

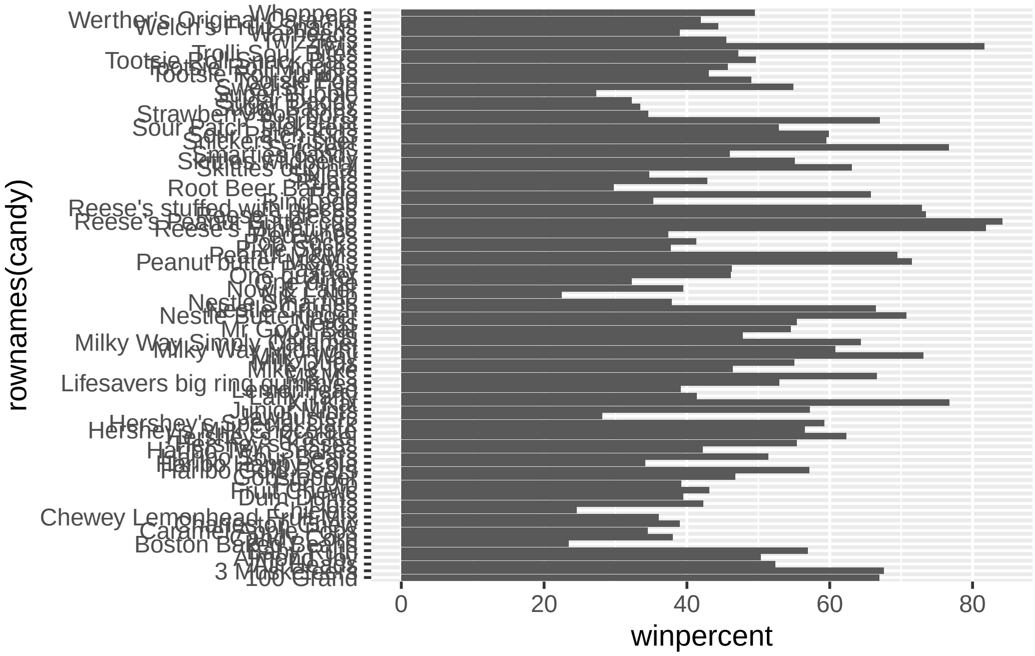
```r
candy |>
  arrange(-winpercent) |>
  head(5)
```

```
                          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
```

Q. Make a first barplot of candy ranking based on winpercent value

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

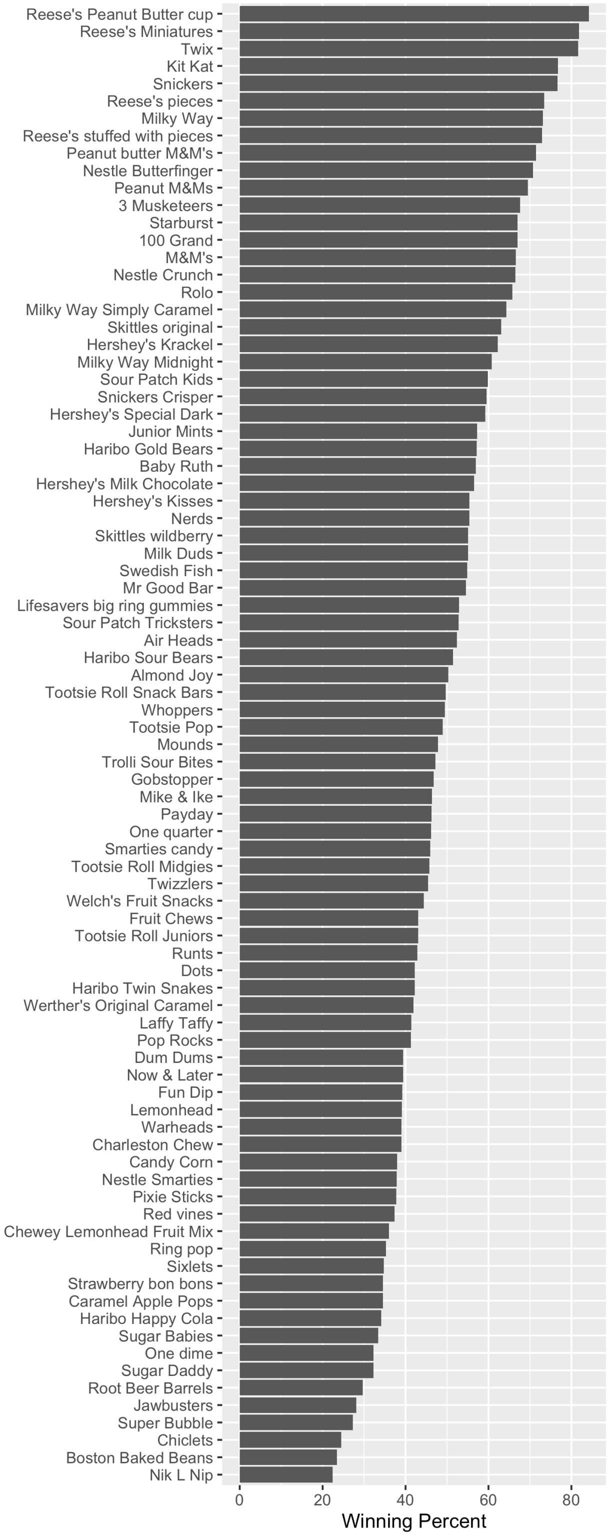


We can make this plot better by rearranging (ordering) the y axis by winpercent so the highest scoring candy is at the topand the lowest is at the bottom

```r
p <- ggplot(candy) +
  aes(x = winpercent, y = reorder(rownames(candy), winpercent)) +
  geom_col() +
  ylab("") +
  xlab("Winning Percent")
```

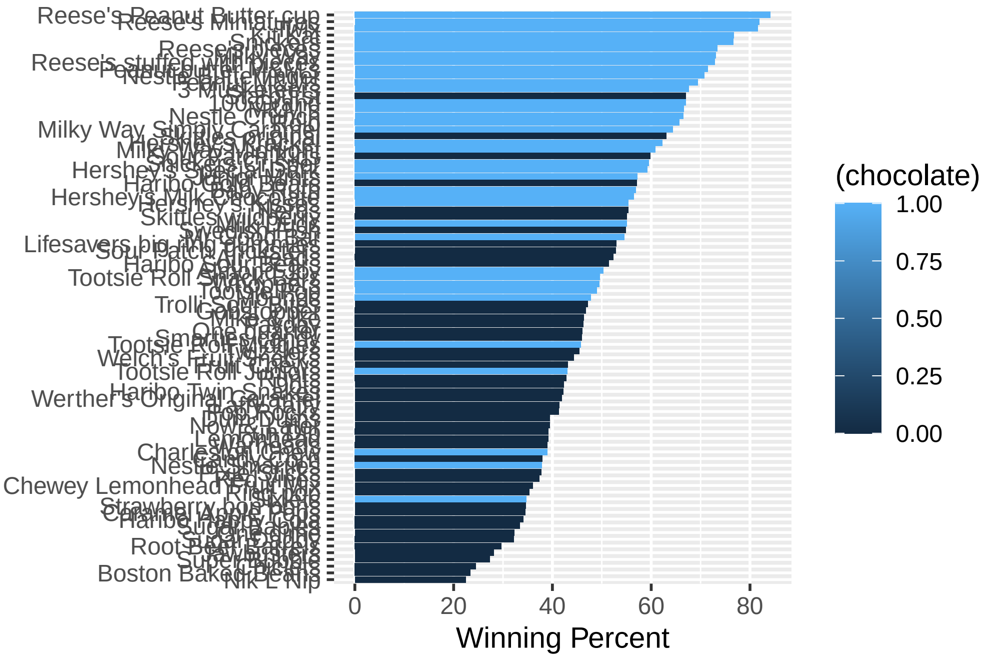
```
ggsave("my_plot.png", height =12, width =5)
```

The following markdown syntax is used to insert an image:


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
0
20
40
60
80
Winning Percent

Q. Color your bars by "chocolate"

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



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

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.

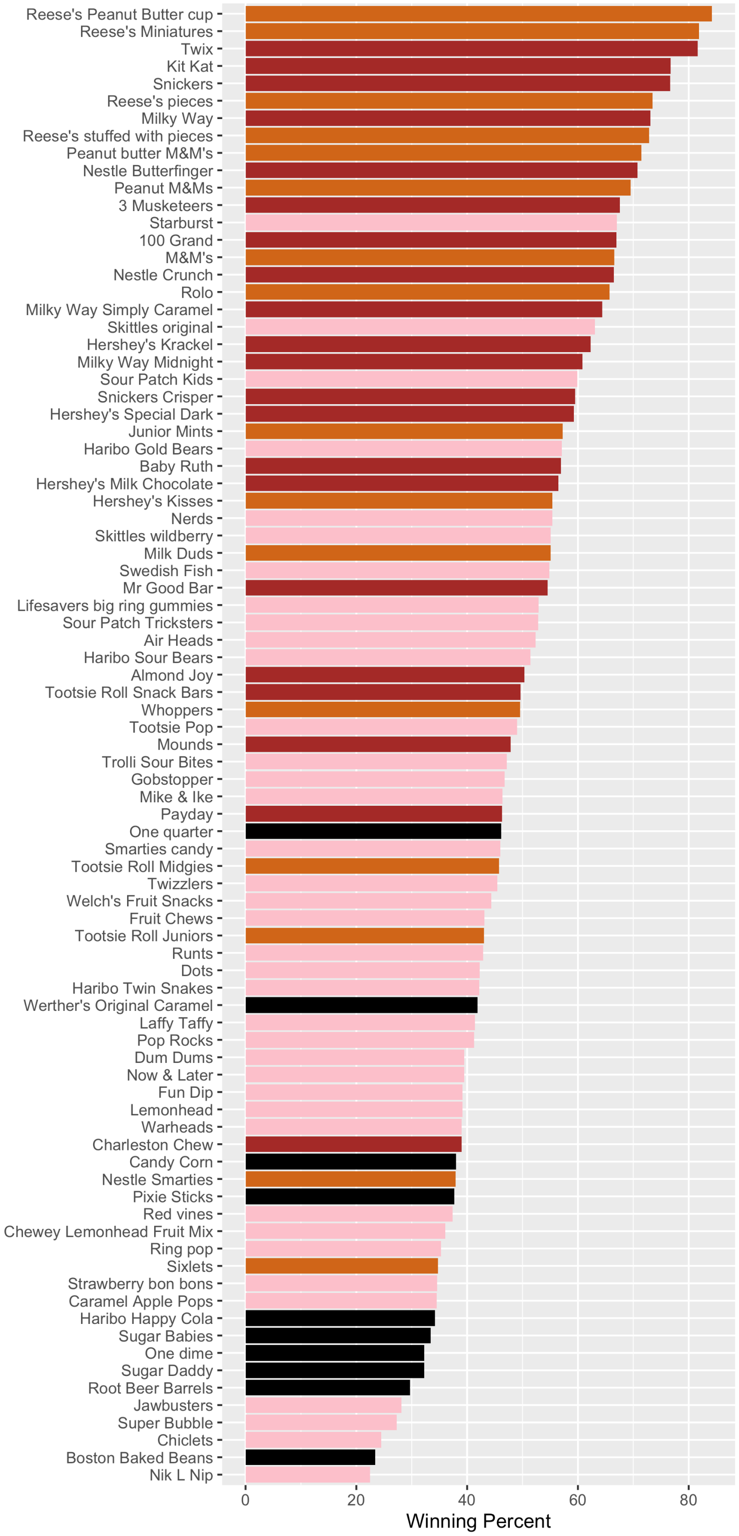
```
mycols <- rep("black", nrow(candy))
mycols[candy$chocolate ==1] <- "chocolate"
mycols[candy$bar ==1] <- "brown"
mycols[candy$fruity ==1] <- "pink"
mycols
```

```
 [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"
```

```
my_color_plot.png <-
  ggplot(candy) +
  aes(x = winpercent, y = reorder(rownames(candy), winpercent)) +
  geom_col(fill = mycols) +
  ylab("") +
  xlab("Winning Percent")
```

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


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
0
20
40
60
80
Winning Percent

Q. What is the worst ranked chocolate candy?

Sixlets

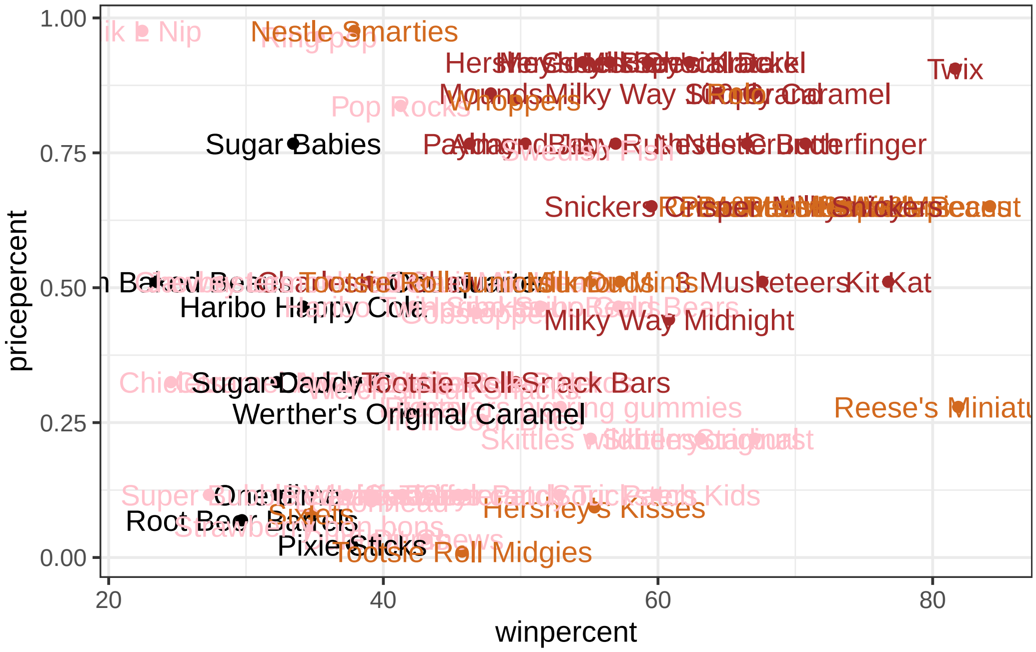
Q. What is the best ranked fruity candy?

Starbusts

### Take a look at pricepercent

plot of winpercent vs pricepercent

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

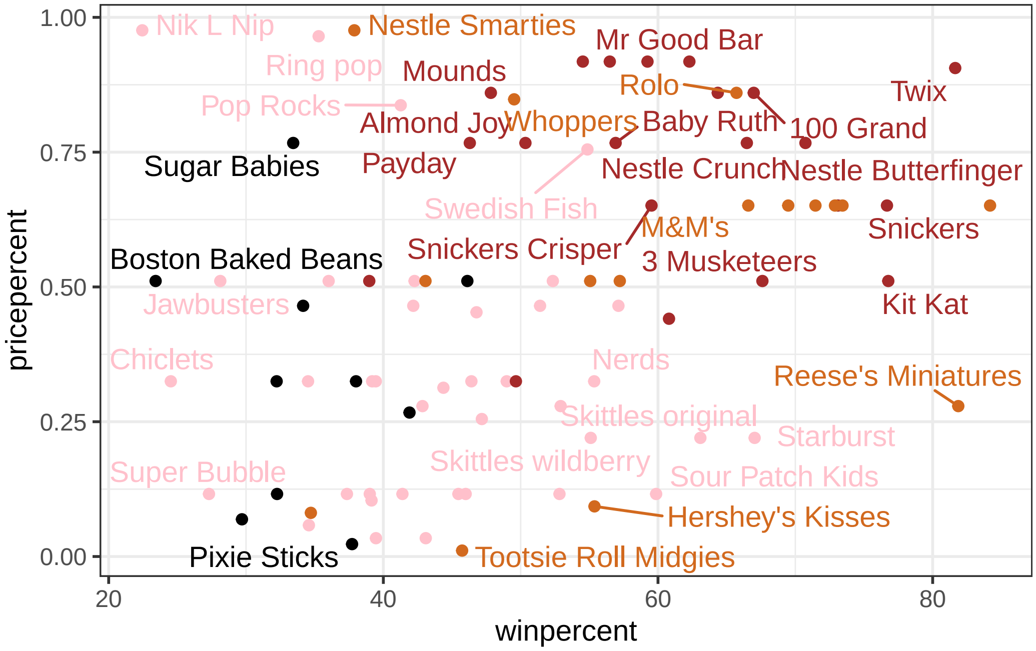


To avoid the common problem of label or text over-plotting, we can use the **ggrepel** package like so

```
library(ggrepel)

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

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



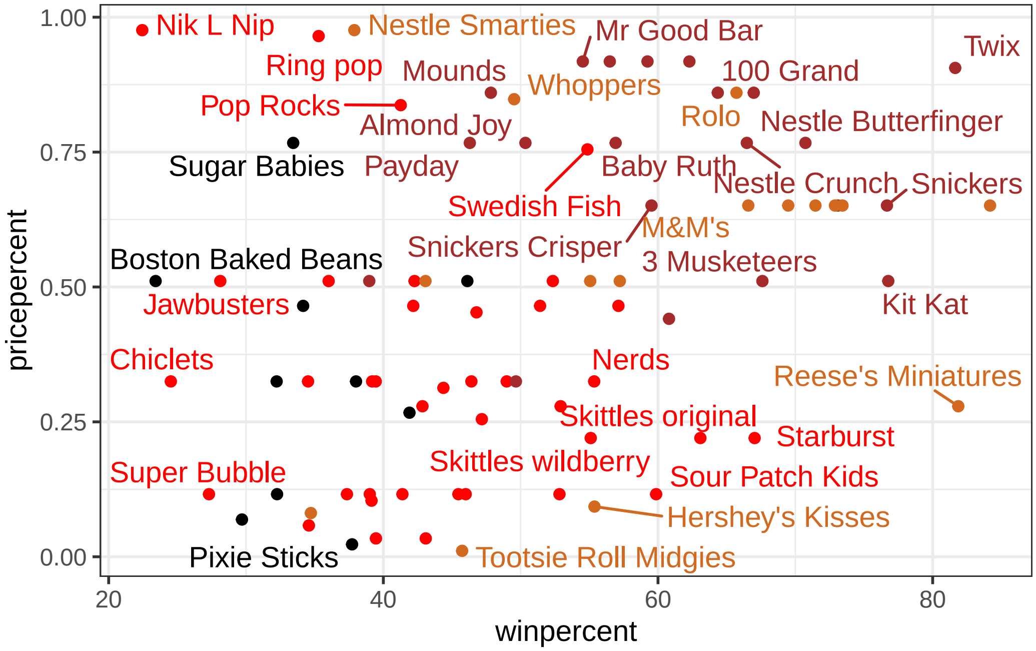
We can control the amount of labels visible by setting different `max.overlaps` values:

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

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

```r
  geom_point(col = mycols) +
  geom_text_repel(col=mycols, max.overlaps =10) +
  theme_bw()
```

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



> Q. 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

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

Nik L Nip

### Exploring the correlation structure

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

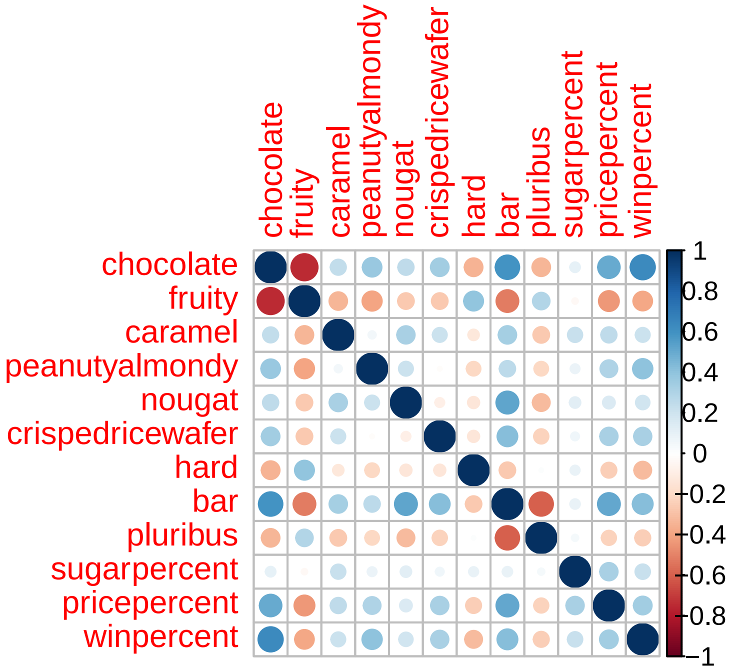
```
cij <- cor(candy)
head(cij)
```

```
                  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)
```



## Principal Component Analysis (PCA)

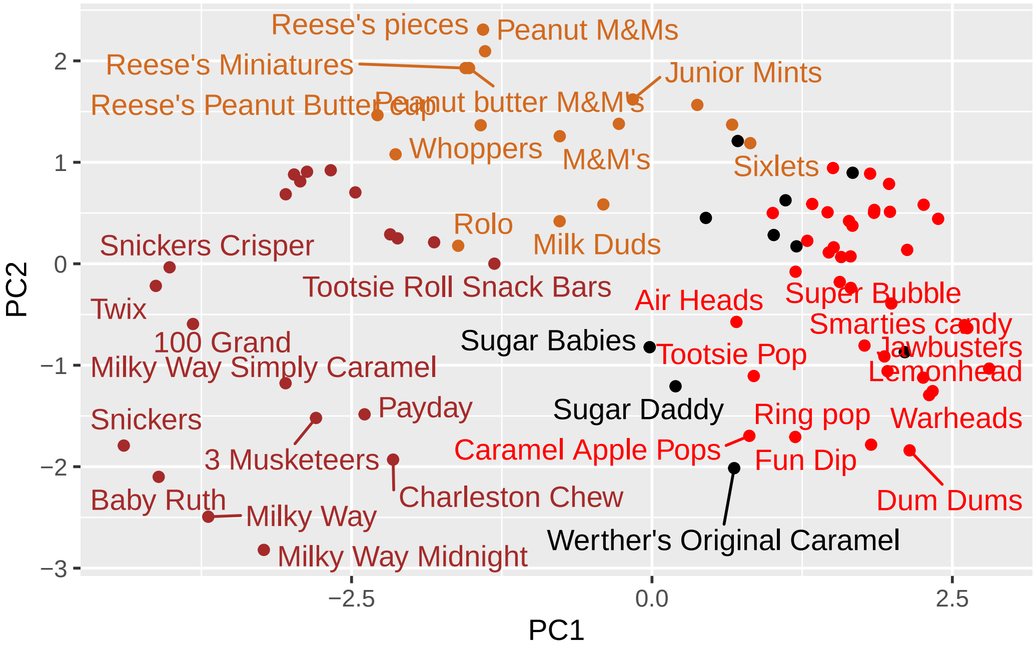
We can use our old friend `prcomp()` function with `scale = TRUE`

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

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

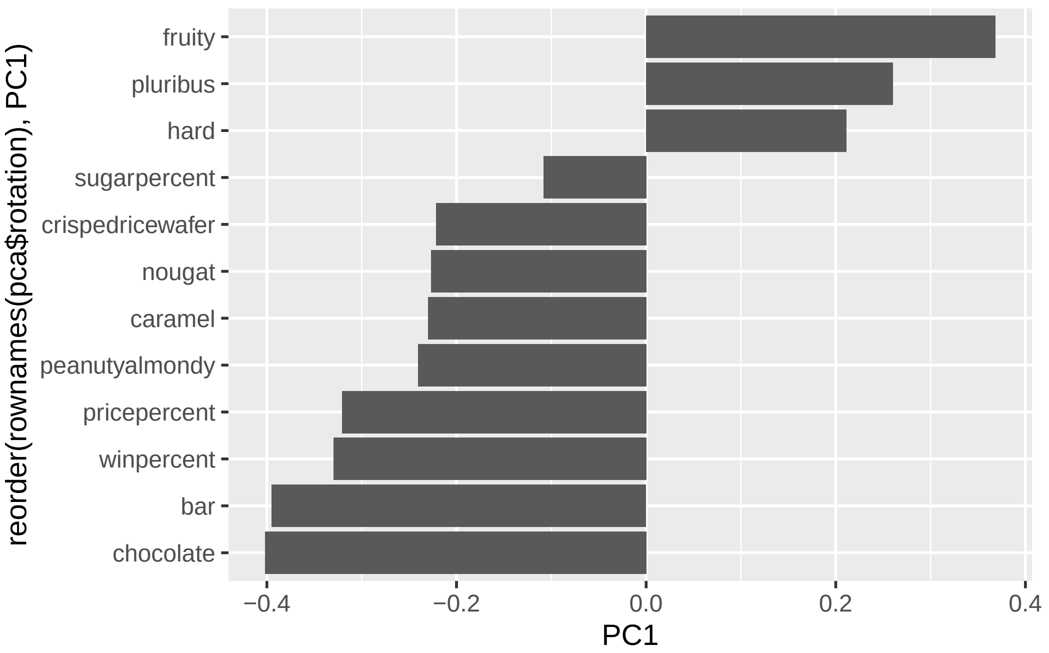
```
ggplot(pca$x) +
  aes(x = PC1, y = PC2, label = rownames(candy)) +
  geom_point(col= mycols) +
  geom_text_repel(col = mycols, max.overlaps = 10)
```

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



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

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