Class 9: Halloween Candy Mini Project

Amanda Wilpitz | A17463962

Table of contents

Today we will examine data from 538 common Halloween candy. In particular, we will use ggplot, dplyr, and PCA to make sense of this multivariate dataset.

Importing Candy Data

```
candy_file <- "https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-power-ranl

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
```

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
```

Hpw many chocolate candy types are in the dataset?

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

```
[1] 37
```

What is your Favorite Candy?

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

My favorite candy from the dataset was Junior Mints.

```
candy["Junior Mints", ]$winpercent
```

```
[1] 57.21925
```

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
```

```
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?

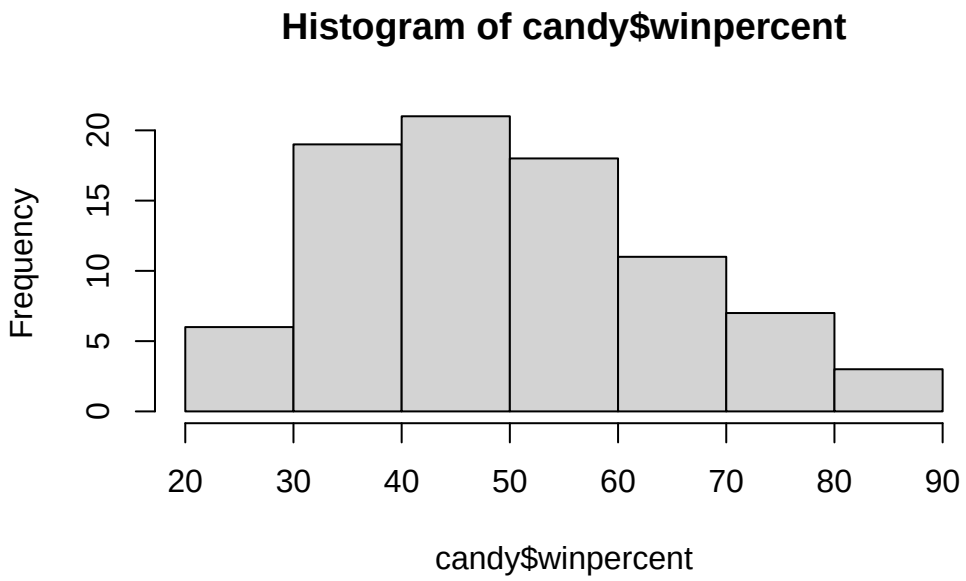
The `winpercent` column seems to be on a different scale (0-100% rather than 0-1) since the mean isn't a decimal compared to the other variables. I will need to scale this dataset before analysis like PCA.

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

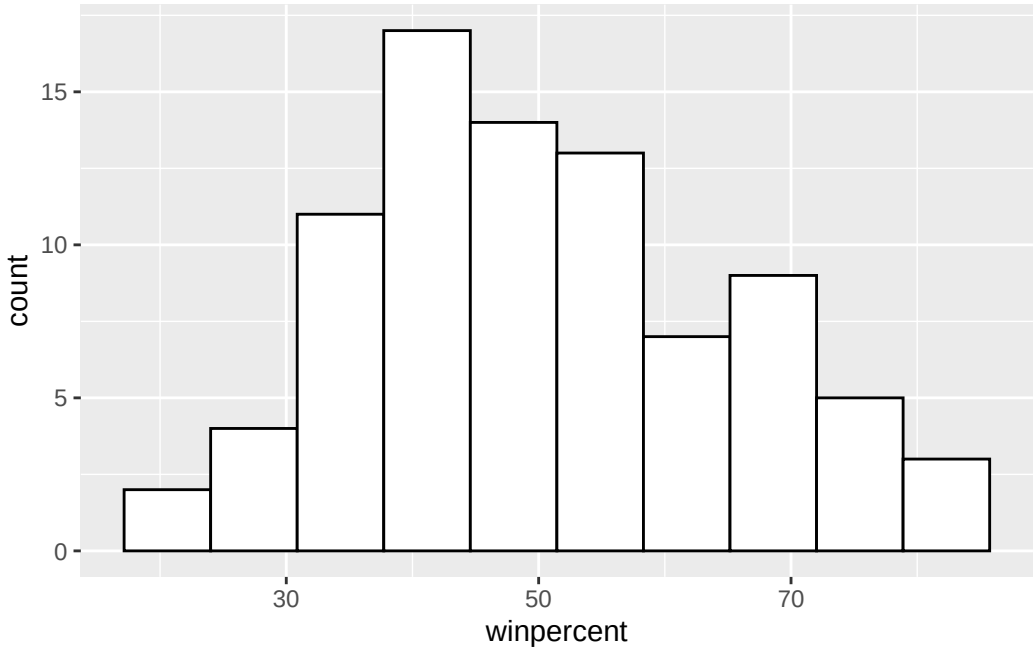
A zero means it isn't a chocolate. A one means it is a chocolate.

Q8. Plot a histogram of winpercent values

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




```
library(ggplot2)

ggplot(candy, aes(x=winpercent)) +
geom_histogram(bins=10, col = "black", fill = "white")
```



Q9. Is the distribution of winpercent values symmetrical?

The distribution of winpercent 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 of the distribution is below 50%.

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

- Step 1: Find all "chocolate" candy
- Step 2: Find their "winpercent" values
- Step 3: Summarize these values
- Step 4: Find all "fruity" candy
- Step 5: Find their "winpercent" values
- Step 6: Summarize these values
- Step 7: Compare the two summary values

1. Find all chocolate candy

```
choc.inds <- candy$chocolate == 1
```

2. Find their winpercent values

```
choc.win <- candy[choc.inds,]$winpercent
```

3. Summarize these values

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

```
[1] 60.92153
```

4-6. Find fruity candy, winpercent values, summarize

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

```
[1] 44.11974
```

7. Compare two summary values.

Clearly, chocolate has a higher mean `winpercent` than fruity candy.

Q12. Is this difference statistically significant?

```
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
```

This difference is statistically significant.

Overall Candy Rankings

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

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

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

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

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

The `order()` function tells us how to arrange the elements of the input to make them sorted - i.e. how to order them.

We can determine the order of winpercent to make them sorted and use that order to arrange the whole dataset.

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

```
                  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
```

```
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
```

```
candy %>%
  #default of arrange is ascending
  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
```

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

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

```
                         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
```

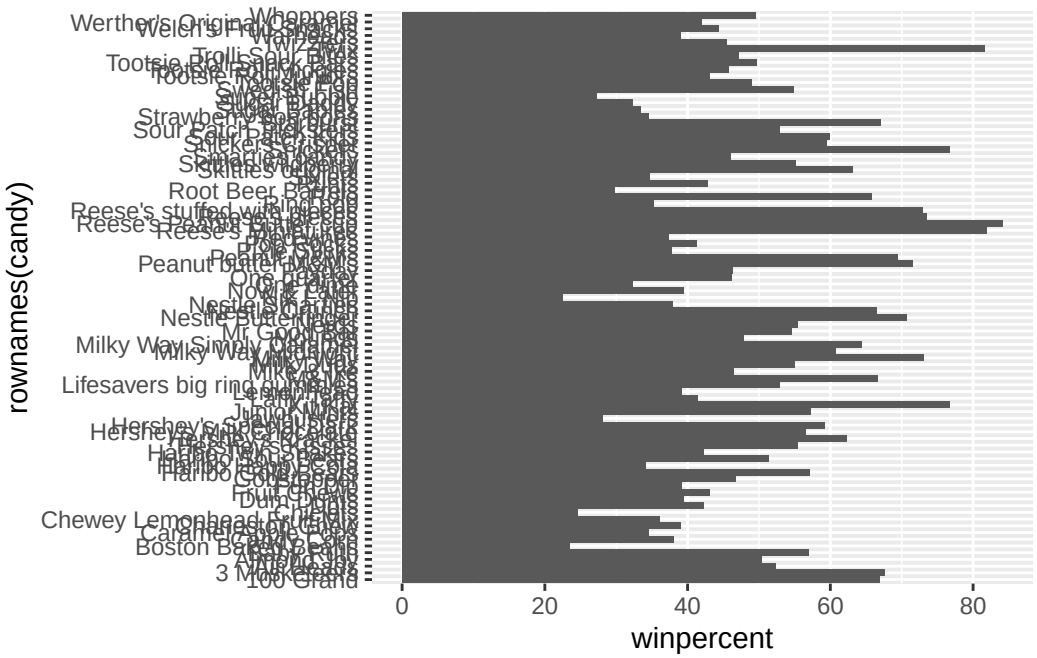
```
candy %>%
  arrange(desc(winpercent)) %>%
  head(5)
```

```
                         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
```

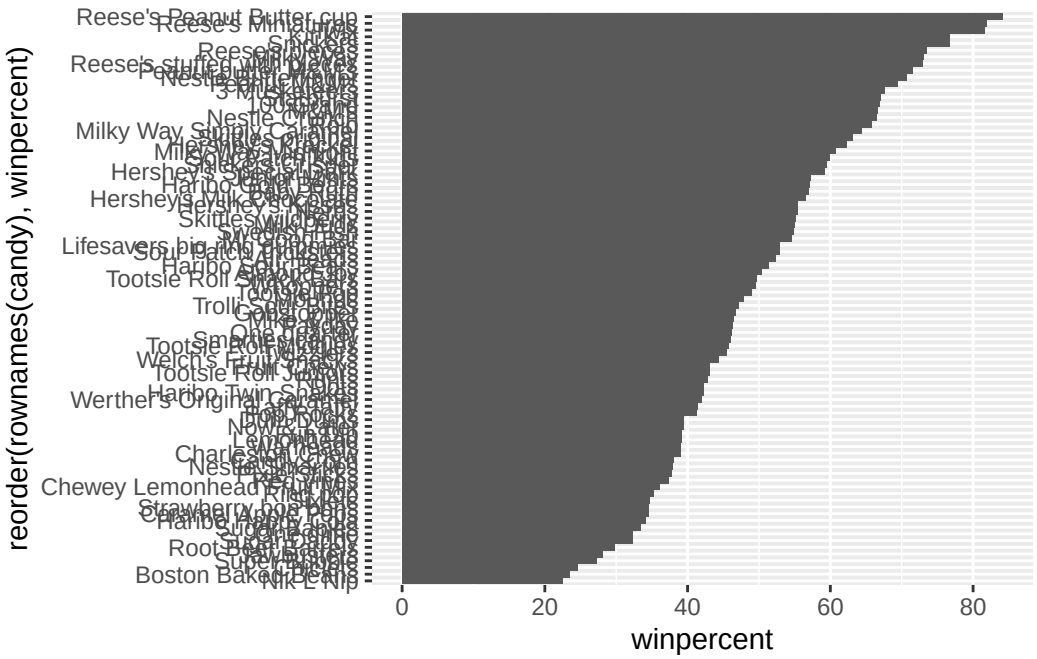
Q15. Make a first barplot of candy ranking based on winpercent values.

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



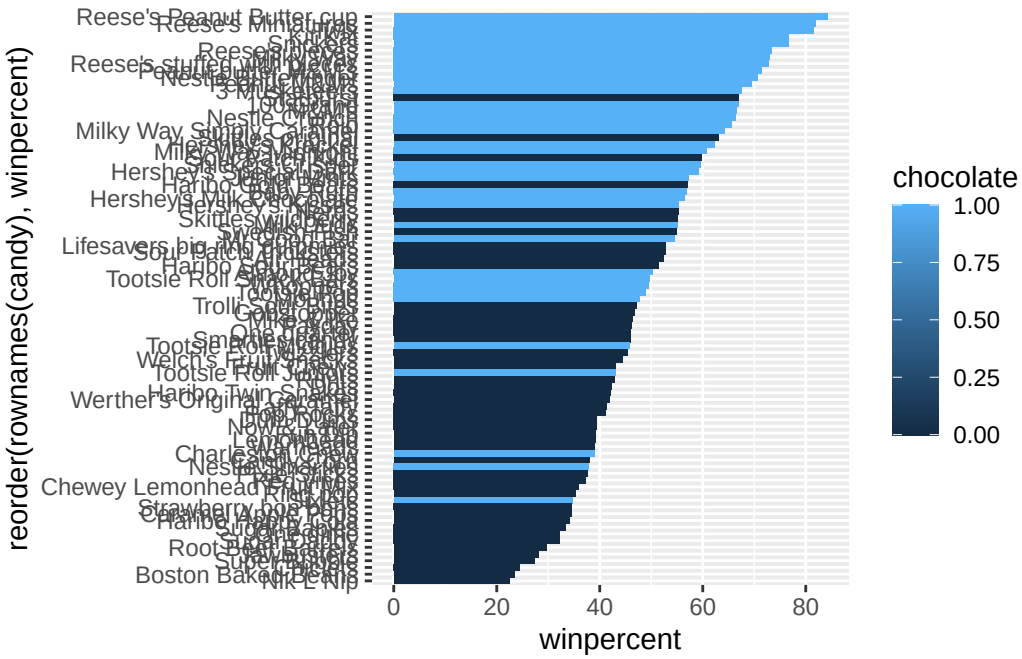
Q16. This is quite ugly, use the reorder() function to get the bars sorted by winpercent?

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



Time to Add Some Useful Color

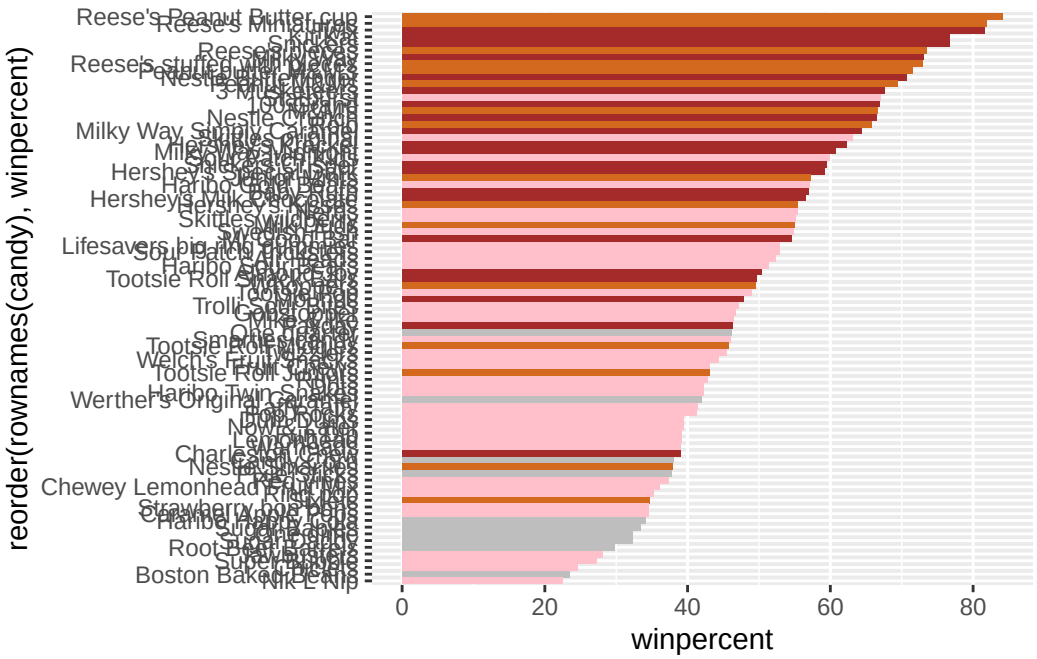
```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy),winpercent), fill = chocolate) +
  geom_col()
```



We need to make our own separate color vector where we can spell out exactly what candy is colored a particular color.

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

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



Q17. What is the worst ranked chocolate candy?

Sixlets

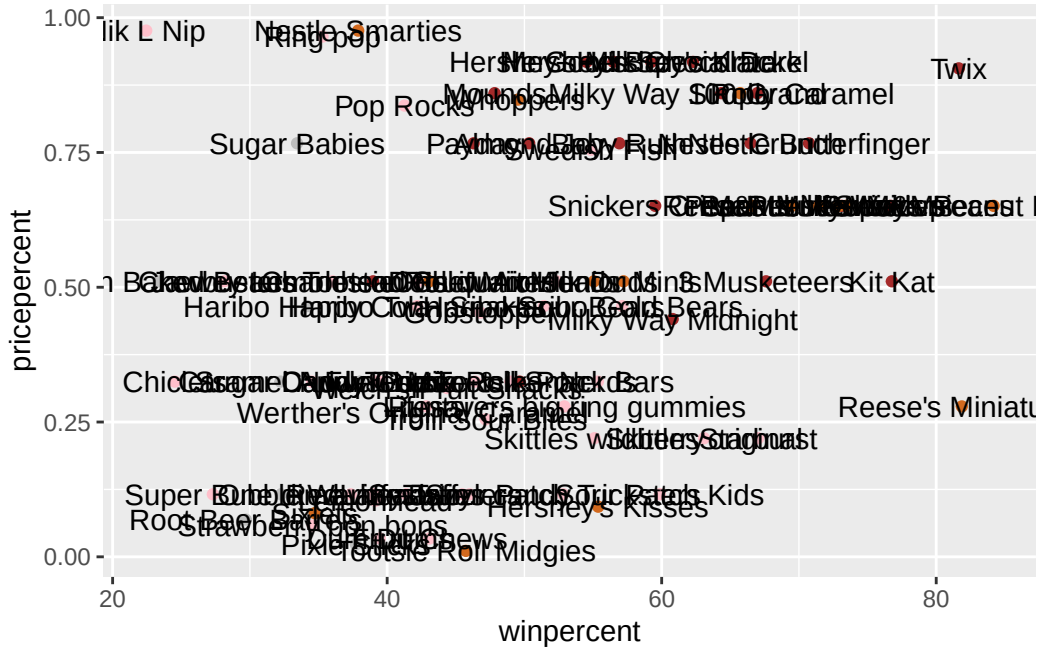
Q18. What is the best ranked fruity candy

Starburst

Taking a look at pricepercent

Make a plot of winpercent (x-axis) vs pricepercent (y-axis)

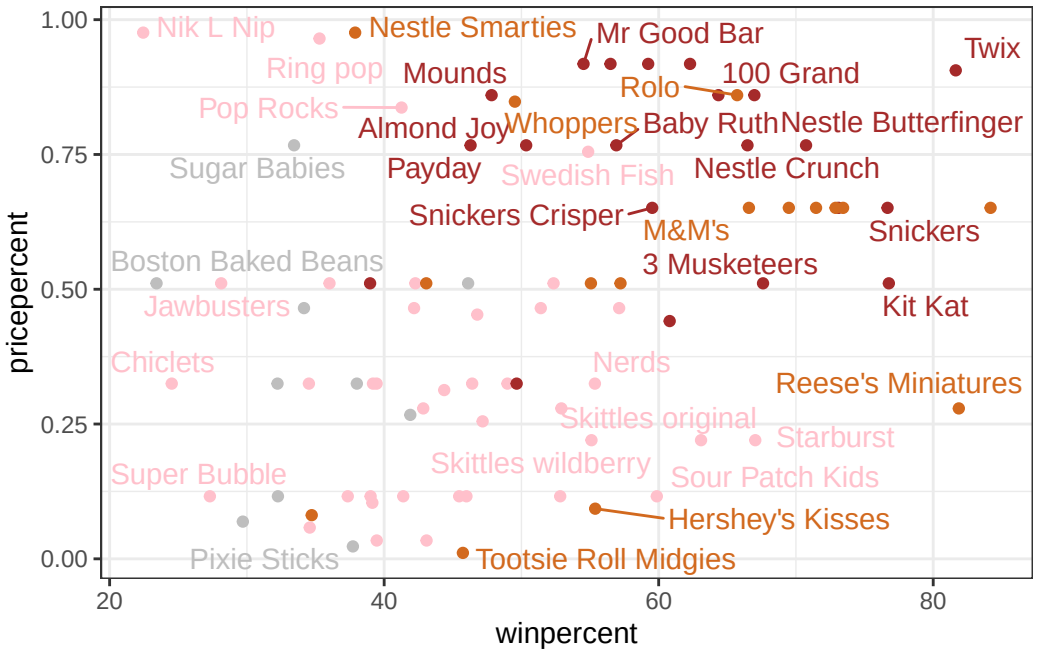
```
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text()
```



To avoid the overplotting of the text on the text labels, we can use the add on package **ggrepel**.

```
library(ggrepel)

ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols, max.overlaps = 10) +
  theme_bw()
```

```
Warning: ggrepel: 50 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?

The highest ranking in winpercent but lowest ranked in pricepercent is Reese's Miniatures.

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

```
ord <- order(candy$pricepercent, decreasing = TRUE)
head( candy[ord,c(11,12)], n=5 )
```

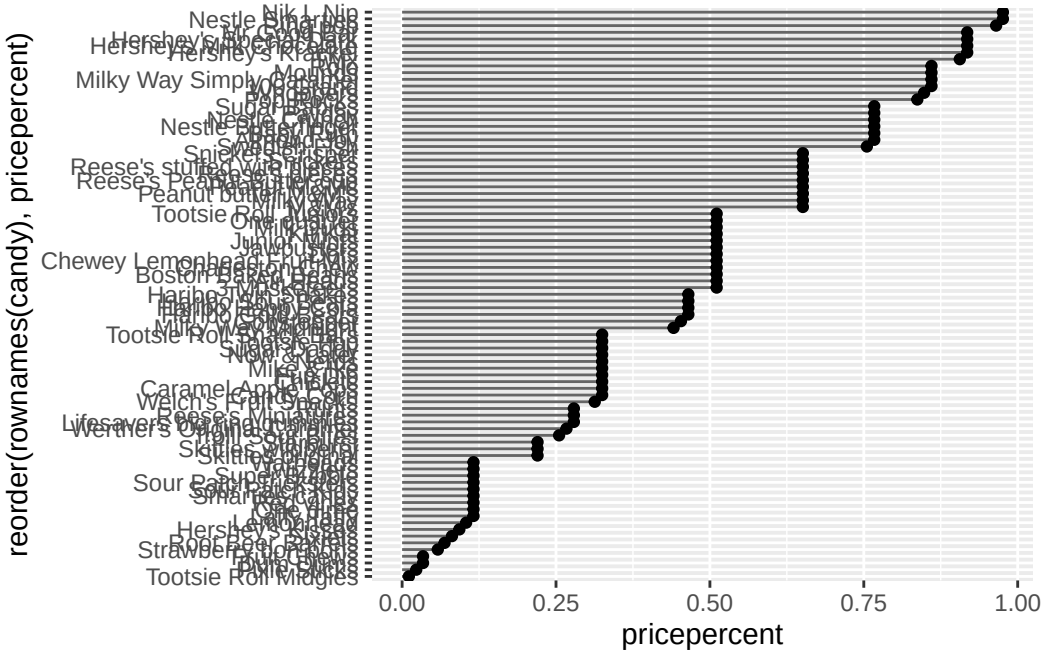
```
                           pricepercent winpercent
Nik L Nip                         0.976   22.44534
Nestle Smarties                   0.976   37.88719
Ring pop                          0.965   35.29076
Hershey's Krackel                 0.918   62.28448
Hershey's Milk Chocolate          0.918   56.49050
```

The top 5 most expensive candy types are Nestle Smarties, Ring Pop, Nik L Nip, Mr. Good Bar, Hershey's Krackel.

Nik L Nip is the least popular.

Q21. Make a barplot again with geom_col() this time using pricepercent and then improve this step by step, first ordering the x-axis by value and finally making a so called "dot chat" or "lollipop" chart by swapping geom_col() for geom_point() + geom_segment().

```
ggplot(candy) +
  aes(pricepercent, reorder(rownames(candy), pricepercent)) +
  geom_segment(aes(yend = reorder(rownames(candy), pricepercent),
                   xend = 0), col="gray40") +
    geom_point()
```



Exploring the correlation structure

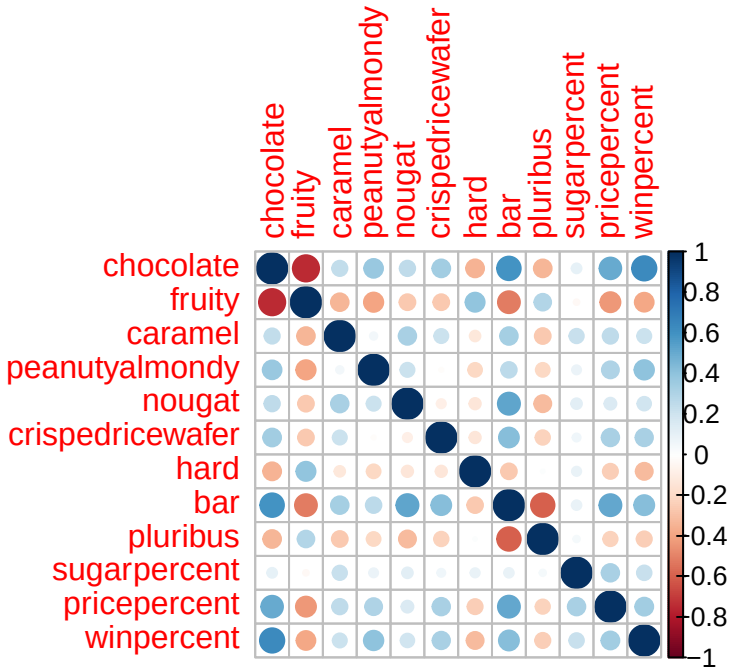
Now that we have explored the dataset a little, we will see how the variables interact with one another.

First we will use correlation and view the results with the **corrplot** package to plot a correlation matrix.

```
library(corrplot)
```

```
corrplot 0.95 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 bar are most positively correlated.

Principal Component Analysis

Let's apply PCA using the `prcomp()` function to our candy dataset remembering to set the scale=TRUE argument.

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

```
attributes(pca)
```

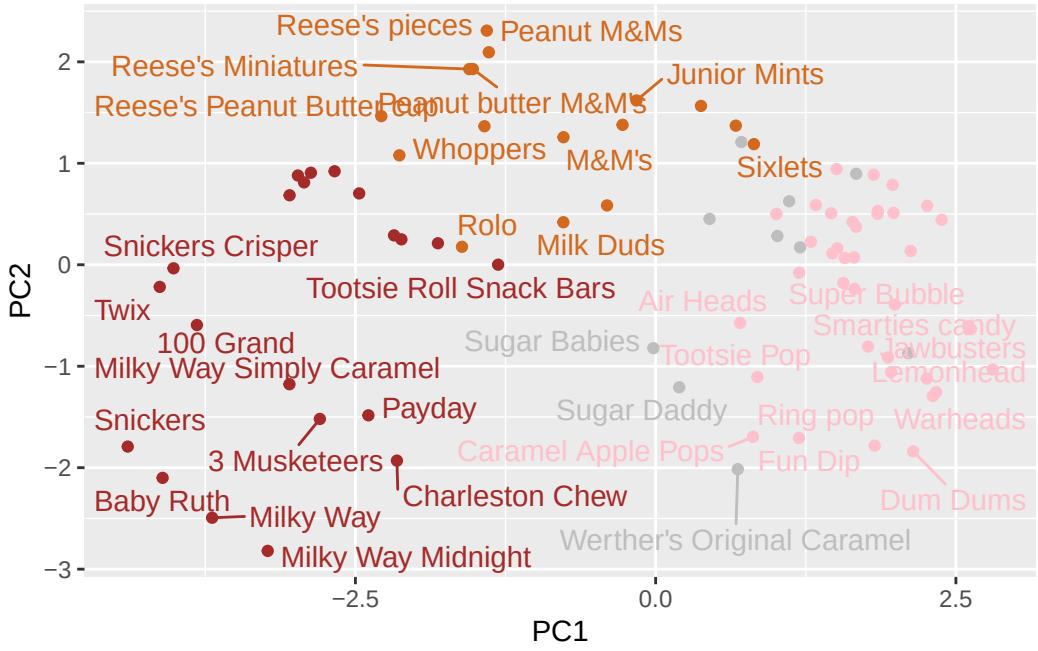
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

x is where the main results are

```
ggplot(pca$x) +
  aes(x = PC1, y = PC2, label=rownames(pca$x)) +
  geom_point(col=my_cols) +
  geom_text_repel(overlap = 5, col = my_cols)
```

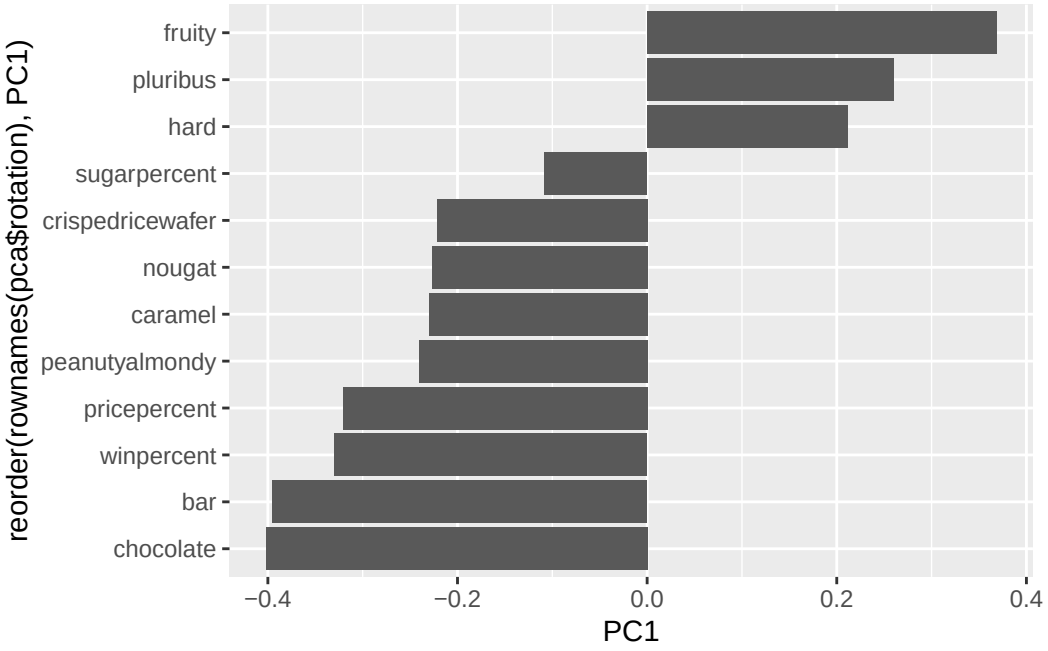
```
Warning in geom_text_repel(overlap = 5, col = my_cols): Ignoring unknown
parameters: `overlap`
```

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



Finally, let's look at how the original variables contribute to the PCs, starting with PC1.

```
ggplot(pca$rotation) +
  aes(PC1, reorder(rownames(pca$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, pluribus, and hard are the variables that are picked up strongly by PC1 in the positive direction. This makes sense because these are the most correlated with each other and are clustered together on the graph as a result.