

Class 10: Halloween Candy Mini project

Charlize Molitor (PID: 18515740)

Table of contents

Background	1
Data Import	1
Quick over view of the dataset	4
Overall Candy rating	8
Principle Component Analysis	16

Background

As it is nearly Halloween and the half way point in the quarter let's do a mini project to help us figure out the best candy!

Our data comes from the 538 website and is available as a CSV file:

Data Import

```
candy <- read.csv("candy-data.csv", row.names = 1)
head(candy)
```

	chocolate	fruity	caramel	peanut	almond	nougat	crisped	rice	wafers
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

```
flextable:: flextable(head(candy))
```

chocolate	fruity	caramel	peanut	almond	nougat	crispedrice	wafer	hard	bar	pluribus	s
1	0	1	0	0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0

Q1. How many different candy types are in this dataset?

```
nrow(candy)
```

[1] 85

```
candy |>
  nrow()
```

[1] 85

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.1     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr    1.3.1
v purrr    1.1.0
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()  
x dplyr::lag()    masks stats::lag()  
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become non-conflicting
```

```
candy %>%  
  nrow()
```

```
[1] 85
```

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

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

```
[1] 38
```

Q3. My favorite candy and it's winpercent

```
candy["Reese's Peanut Butter cup",]$winpercent
```

```
[1] 84.18029
```

```
library(dplyr)  
  
candy |>  
  filter(rownames(candy) == "Reese's Peanut Butter cup") |>  
  select(winpercent)
```

```
winpercent  
Reese's Peanut Butter cup 84.18029
```

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

```
library(dplyr)  
  
candy |>  
  filter(rownames(candy) == "Kit Kat") |>  
  select(winpercent)
```

```
winpercent  
Kit Kat 76.7686
```

Q5. What is the winpercent value for “Tootsie Roll Snack Bars”?

```

library(dplyr)

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

```

```

      winpercent
Tootsie Roll Snack Bars    49.6535

```

Quick overview of the dataset

```

library(skimr)

skimr::skim(candy)

```

Table 2: 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	
peanutyalmond	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	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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 is on a 0-100 scale and the rest are 0-1 scale

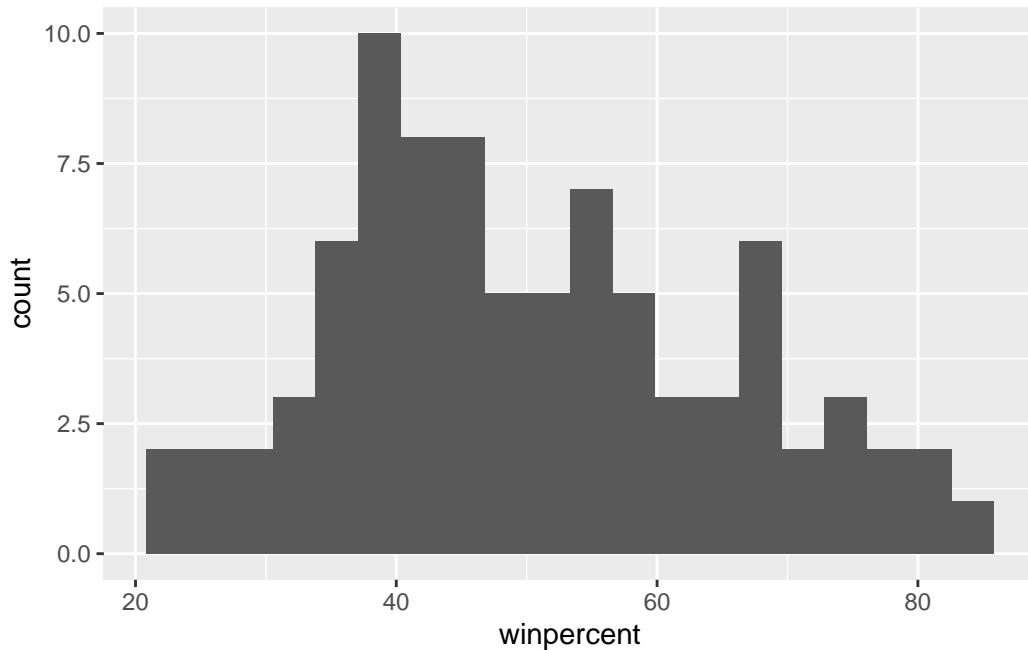
Q7. What do you think a zero and one represent for the candy\$chocolate column?

That the candy does not contain any chocolate

Q8. Plot a histogram of winpercent values

```
library(ggplot2)

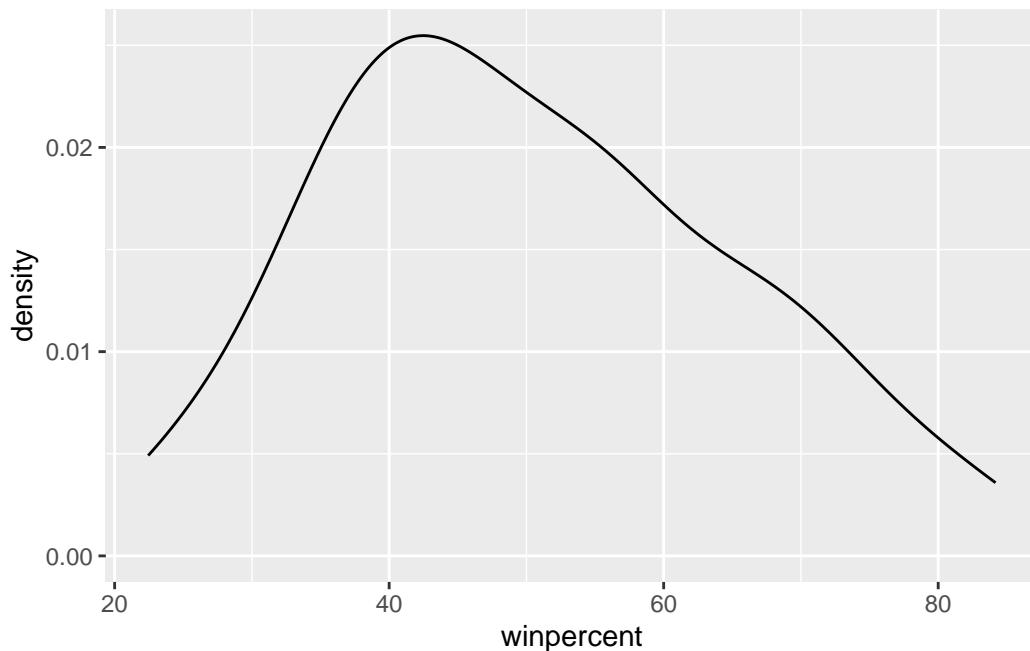
ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins = 20)
```



Q9. Is the distribution of winpercent values symmetrical?

No, not symmetrical based on the graph.

```
ggplot(candy) +  
  aes(winpercent) +  
  geom_density()
```



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

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

[1] 50.31676

```
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?

Chocolate is higher ranked than fruit candy

```
# 1. Find all chocolate candy in the dataset
# 2. Find their winpercent values
# 3. Calculate the mean of these values

# 4-6. Do the same for the fruity candy
# 7. Compare mean winpercent of chocolate vs fruity
# 8. Pick the highest as the winner

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

```
[1] 60.92153
```

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

```
[1] 44.11974
```

Q12. Is this difference statistically significant?

There is a staticical significance when it comes to chocolate and fruity candy

```
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 rating

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

```
library(dplyr)
```

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

	chocolate	fruity	caramel	peanut	yalmond	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
	crispedrice	wafer	hard	bar	pluribus	sugarpercent
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					

```
x <- c(5,10,15,4)  
#sort(x)  
order(x)
```

[1] 4 1 2 3

```
#arrange(candy$winpercent)
```

```
ord.ind <- order(candy$winpercent)  
head(candy[ord.ind,], 5)
```

	chocolate	fruity	caramel	peanuty	almondy	nougat				
Nik L Nip	0	1	0	0	0	0				
Boston Baked Beans	0	0	0	1	0	0				
Chiclets	0	1	0	0	0	0				
Super Bubble	0	1	0	0	0	0				
Jawbusters	0	1	0	0	0	0				
	crisped	rice	wafer	hard	bar	pluribus	sugar	percent	price	percent
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	
	win	percent								
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?

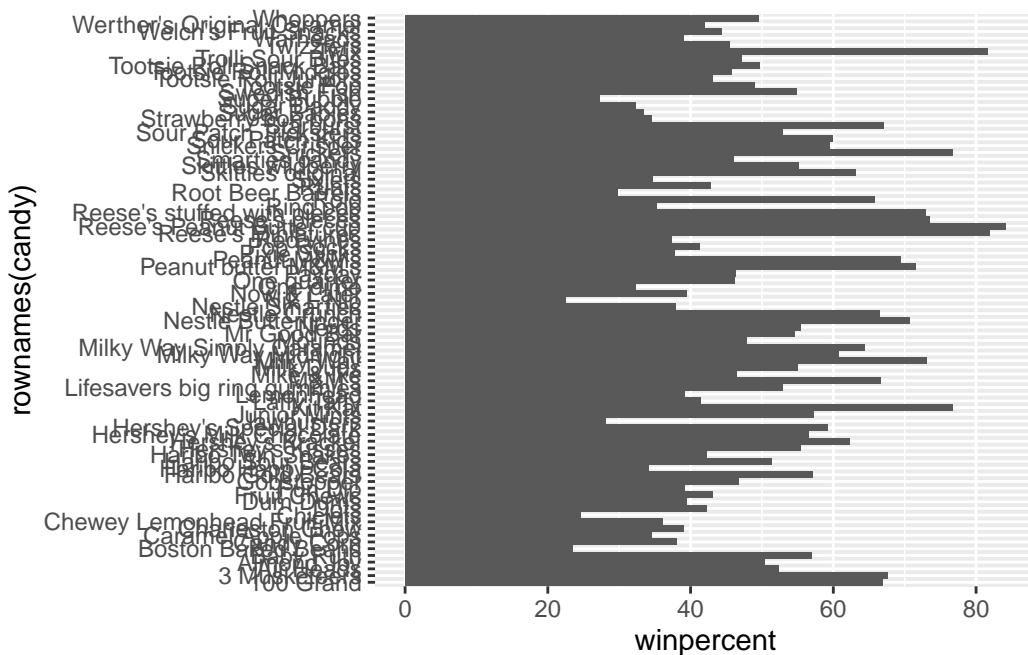
```
candy |>
  arrange(-winpercent) |>
  head(5)
```

	chocolate	fruity	caramel	peanuty	almondy	nougat			
Reese's Peanut Butter cup	1	0	0	1	0	0			
Reese's Miniatures	1	0	0	1	0	0			
Twix	1	0	1	0	0	0			
Kit Kat	1	0	0	0	0	0			
Snickers	1	0	1	1	1	1			
	crisped	rice	wafer	hard	bar	pluribus	sugar	percent	
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		
	price	percent	win	percent					
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.

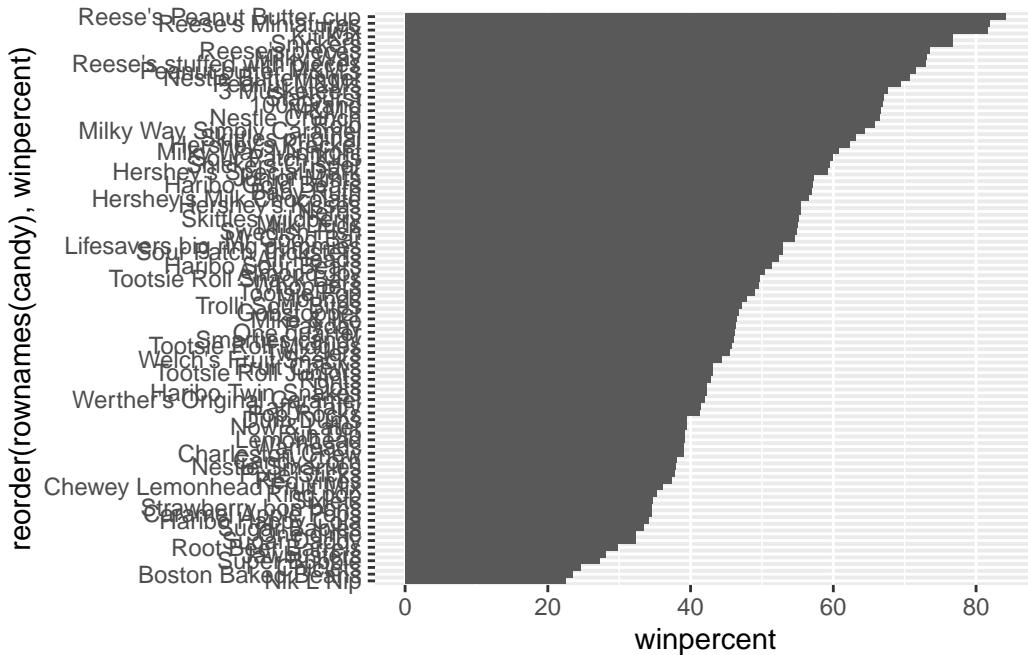
```
library(ggplot2)

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



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

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



Q17. What is the worst ranked chocolate candy?

One dime

Q18. What is the best ranked fruity candy?

Starburst

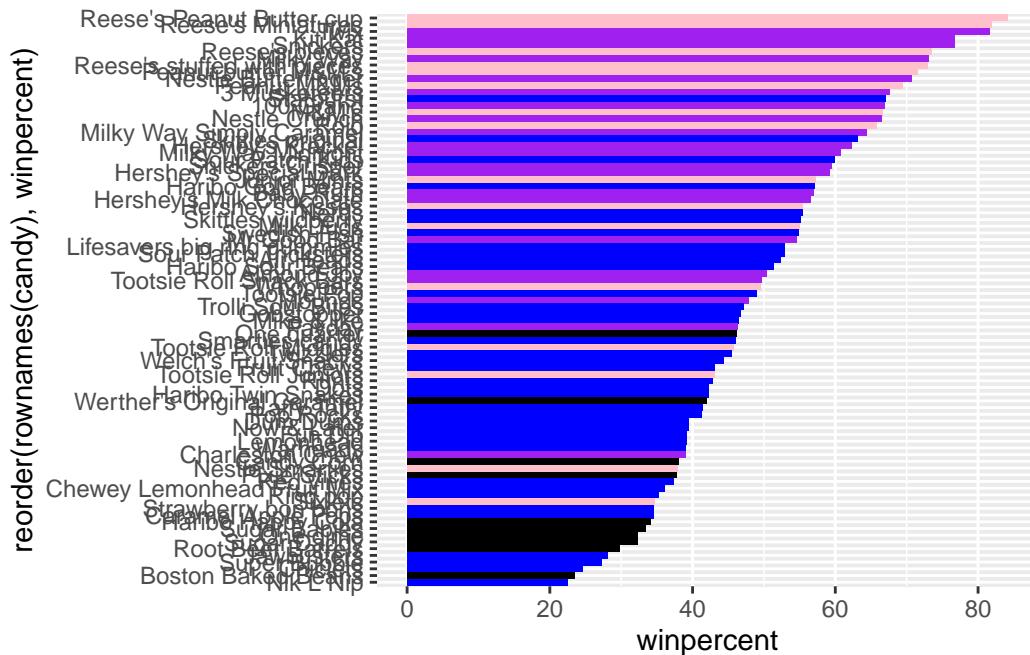
Add some color based on the type of candy

```
my_cols <- rep("black", nrow(candy))
my_cols[as.logical(candy$chocolate)] <- "pink"
my_cols[as.logical(candy$fruity)] <- "blue"
my_cols[as.logical(candy$bar)] <- "purple"
my_cols
```

```
[1] "purple" "purple" "black" "black" "blue" "purple" "purple" "black"
[9] "black" "blue" "purple" "blue" "blue" "blue" "blue" "blue"
[17] "blue" "blue" "blue" "black" "blue" "blue" "pink" "purple"
[25] "purple" "purple" "blue" "pink" "purple" "blue" "blue" "blue"
[33] "pink" "pink" "blue" "pink" "purple" "purple" "purple" "purple"
[41] "purple" "blue" "purple" "purple" "blue" "blue" "purple" "pink"
[49] "black" "blue" "blue" "pink" "pink" "pink" "pink" "blue"
[57] "pink" "black" "blue" "pink" "blue" "blue" "pink" "blue"
```

```
[65] "purple" "purple" "blue"     "blue"    "blue"    "blue"    "black"   "black"
[73] "blue"    "blue"    "blue"    "pink"   "pink"   "purple"  "blue"    "purple"
[81] "blue"    "blue"    "blue"    "black"   "pink"
```

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



Winpercent and Pricepercent

A plot with both variables/columns winpercent and pricepercent

```
my_cols[as.logical(candy$fruity)] <- "blue"

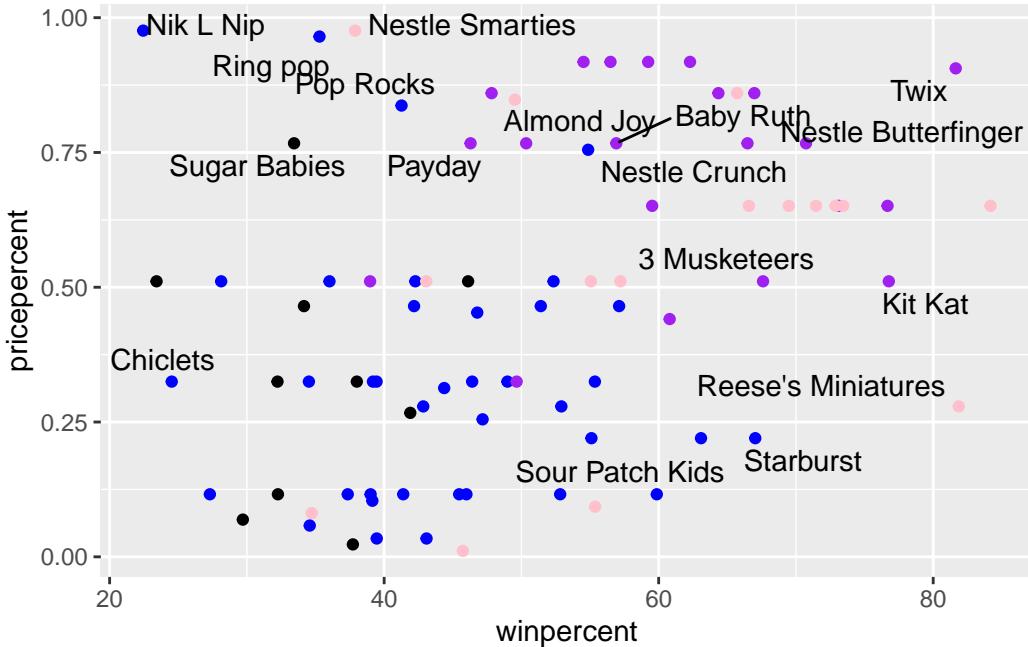
ggplot(candy) +
  aes(x= winpercent,
      y= pricepercent,
      label=reorder(rownames(candy))) +
  geom_point(col= my_cols) +
  geom_text()
```



```
library(ggrepel)

ggplot(candy) +
  aes(x= winpercent,
      y= pricepercent,
      label=rownames(candy)) +
  geom_point(col= my_cols) +
  geom_text_repel(max.overlaps = 7)
```

Warning: ggrepel: 68 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?

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

	pricepercent	winpercent
Tootsie Roll Midgies	0.011	45.73675
Pixie Sticks	0.023	37.72234
Dum Dums	0.034	39.46056
Fruit Chews	0.034	43.08892
Strawberry bon bons	0.058	34.57899

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

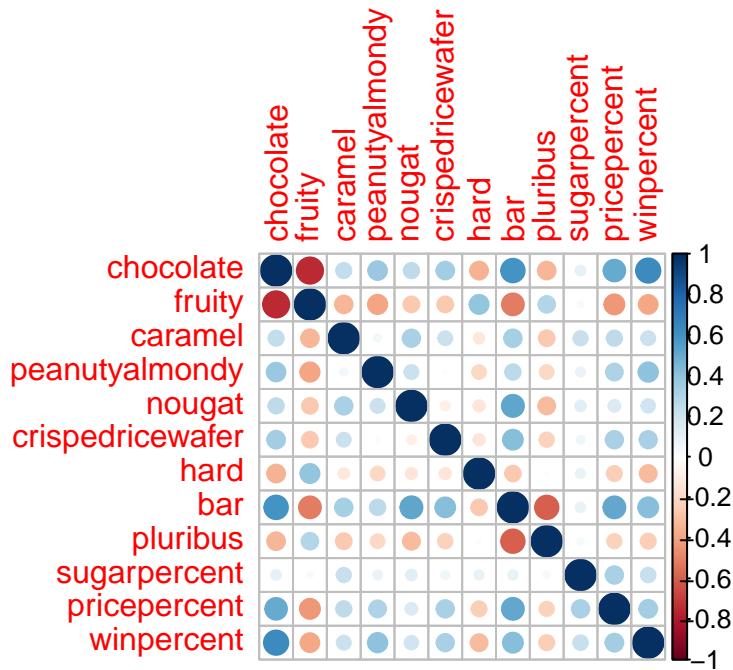
Exploring the correlation structure

Now that we've explored the dataset a little, we'll see how the variables interact with one another. We'll 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 fruit candies, mixed together.

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

Chocolate with chocolate candies.

Principle Component Analysis

The function to use it called `prcomp()` with the optional `scale=T/F` argument.

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

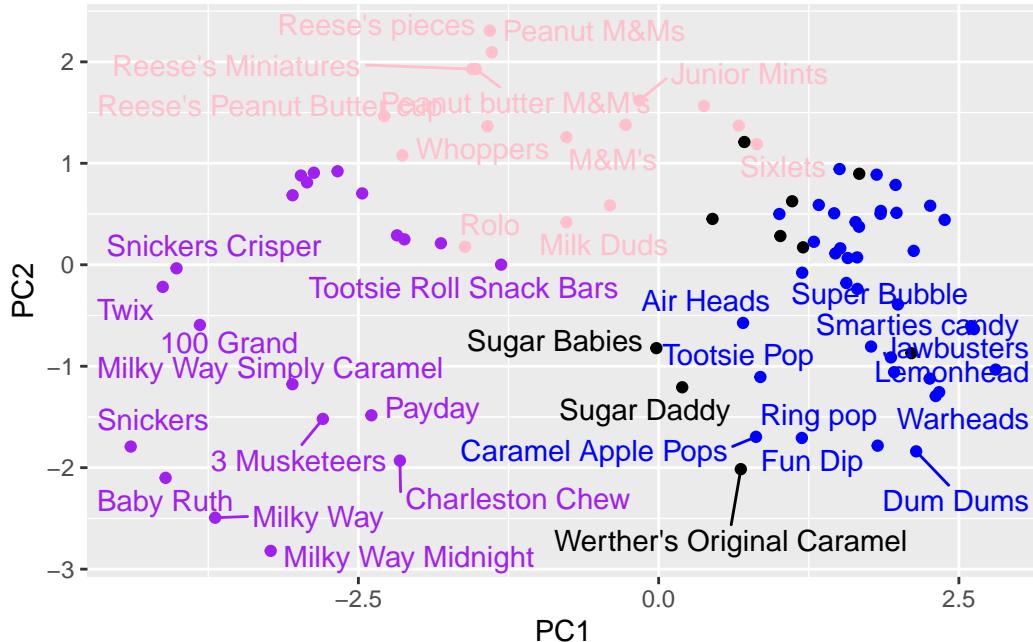
Importance of components:

	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		

Our main PCA results figure

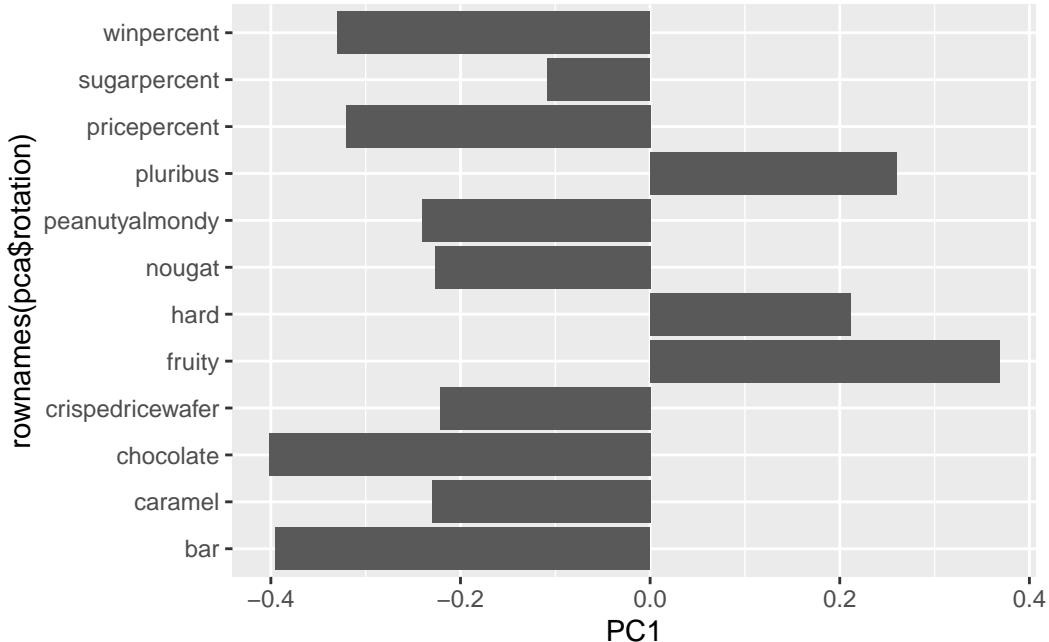
```
ggplot(pca$x) +
  aes(PC1, PC2, label = rownames(pca$x)) +
  geom_point(col = my_cols) +
  geom_text_repel(col = my_cols)
```

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



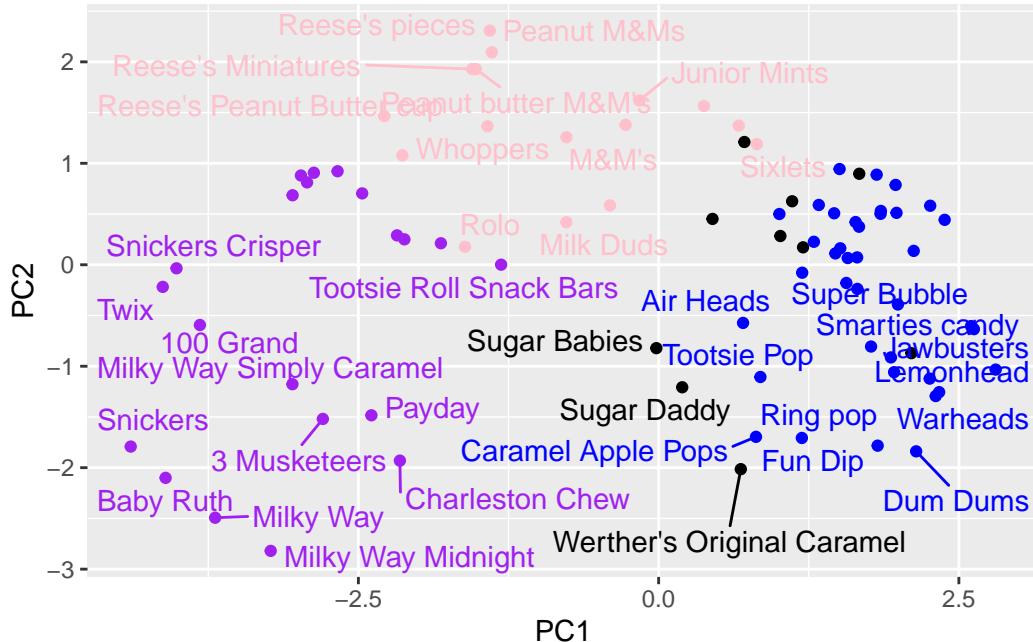
We should also examine the variable “loading’s” or contributions of the original variables to the new PCs

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



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

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



Interactive plots that can be zoomed on and “brushed” over can be made with the **plotly** package. Its output is interactive and will not render to PDF :(

```
library(plotly)
```

```
Attaching package: 'plotly'
```

```
The following object is masked from 'package:ggplot2':
```

```
last_plot
```

```
The following object is masked from 'package:stats':
```

```
filter
```

```
The following object is masked from 'package:graphics':
```

```
layout
```

```
##plotly(p)
```