

Class 10: Halloween Mini-Project

Wade Ingersoll (PID: A69038080)

Background

In this mini-project, you will explore FiveThirtyEight's Halloween Candy dataset. FiveThirtyEight, sometimes rendered as just 538, is an American website that focuses mostly on opinion poll analysis, politics, economics, and sports blogging. They recently ran a rather large poll to determine which candy their readers like best. From their website: "While we don't know who exactly voted, we do know this: 8,371 different IP addresses voted on about 269,000 randomly generated candy matchups".

Your task is to explore their candy dataset to find out answers to these types of questions - but most of all your job is to have fun, learn by doing hands on data analysis, and hopefully make this type of analysis less frightening for the future! **Let's get started.**

1. Importing candy data

First things first, let's get the data from the FiveThirtyEight GitHub repo. You can either read from the URL directly or download this `candy-data.csv` file and place it in your project directory. Either way we need to load it up with `read.csv()` and inspect the data to see exactly what we're dealing with.

```
candy_file <- "candy-data.csv"

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

What is in this dataset?

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

chocolate	fruity	caramel	peanut	almond	ynougat	crisped	ricewafer	hard	bar	pluribus s
1	0	1	0	0	1	0	0	1	0	
1	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	1	0	0	0	0	0	0	0	0	
1	0	0	1	0	0	0	0	1	0	

We will take a whirlwind tour of this dataset and in the process answer the questions highlighted in red through this page that aim to guide your exploration process. We will then wrap up by trying Principal Component Analysis (PCA) on this dataset to get yet more experience with this important multivariate method. It will yield a kind of “Map of Halloween Candy Space”. How cool is that! Let’s explore...

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

Answer1: There are 85 different candy types; see code below for calculations.

```
library(dplyr)
candy |> nrow()
```

[1] 85

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

Answer: There are 38 fruity candy types; see code below for calculations.

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

```
[1] 38
```

Q3. What is your favorite candy in the dataset and what is its winpercent value?

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

```
[1] 81.64291
```

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

```
library(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

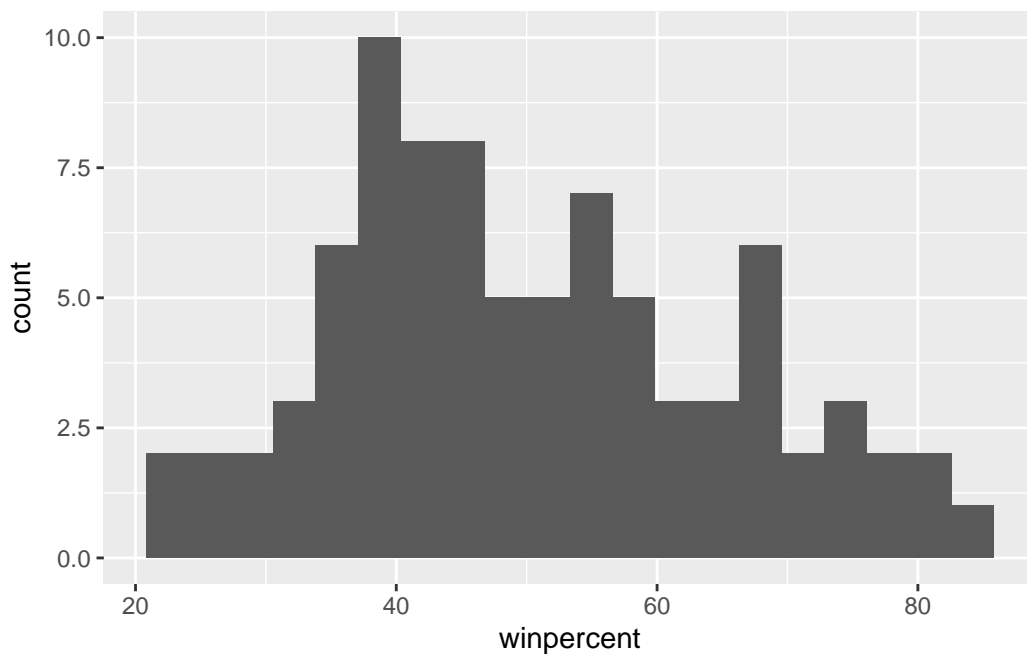
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
chocolate	0	1	0.44	0.50	0.00	0.00	0.00	1.00	1.00	
fruity	0	1	0.45	0.50	0.00	0.00	0.00	1.00	1.00	
caramel	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
peanutyalmondy	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
nougat	0	1	0.08	0.28	0.00	0.00	0.00	0.00	1.00	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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	

Q8. Plot a histogram of winpercent values

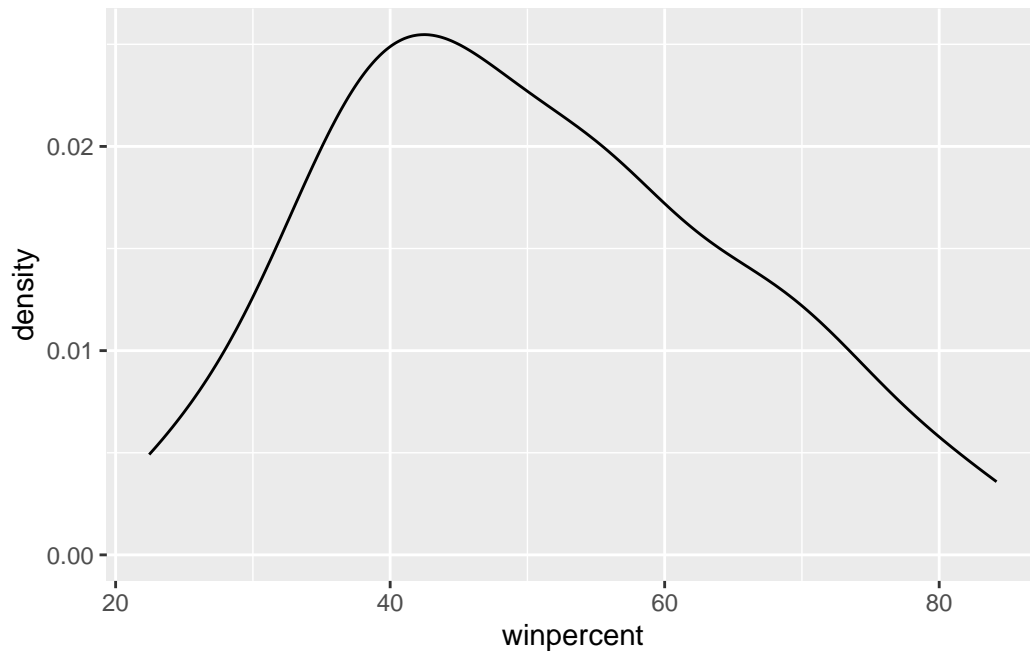
```
library(ggplot2)

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



Q9. Is the distribution of winpercent values symmetrical?

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



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

Answer: It is 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?

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

	chocolate	fruity	caramel	peanut	almond	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
	crisped	rice	wafer	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

```
# Step 2. Extract 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
```

```
# Step 3. Find the mean of these values
choc.mean <- mean(choc.win)
```



```
# Step 4-6. Do the same for fruity candy
fruity.inds <- candy$fruity==1
fruity.candy <- candy[fruity.inds,]
fruity.win <- fruity.candy$winpercent
fruity.mean <- mean(fruity.win)

# Step 7. Which mean value is higher

choc.mean
```

```
[1] 60.92153
```

```
fruity.mean
```

```
[1] 44.11974
```

```
# chocolate is higher
```

Q12. Is this difference statistically significant?

```
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
```

```
round ( mean(candy$winpercent), 4)
```

```
[1] 50.3168
```

```
candy |> select(winpercent)
```

	winpercent
100 Grand	66.97173
3 Musketeers	67.60294
One dime	32.26109
One quarter	46.11650
Air Heads	52.34146
Almond Joy	50.34755
Baby Ruth	56.91455
Boston Baked Beans	23.41782
Candy Corn	38.01096
Caramel Apple Pops	34.51768
Charleston Chew	38.97504
Chewy Lemonhead Fruit Mix	36.01763
Chiclets	24.52499
Dots	42.27208
Dum Dums	39.46056
Fruit Chews	43.08892
Fun Dip	39.18550
Gobstopper	46.78335
Haribo Gold Bears	57.11974
Haribo Happy Cola	34.15896
Haribo Sour Bears	51.41243
Haribo Twin Snakes	42.17877
Hershey's Kisses	55.37545
Hershey's Krackel	62.28448
Hershey's Milk Chocolate	56.49050
Hershey's Special Dark	59.23612
Jawbusters	28.12744
Junior Mints	57.21925
Kit Kat	76.76860
Laffy Taffy	41.38956
Lemonhead	39.14106
Lifesavers big ring gummies	52.91139
Peanut butter M&M's	71.46505
M&M's	66.57458
Mike & Ike	46.41172
Milk Duds	55.06407
Milky Way	73.09956
Milky Way Midnight	60.80070
Milky Way Simply Caramel	64.35334

Mounds	47.82975
Mr Good Bar	54.52645
Nerds	55.35405
Nestle Butterfinger	70.73564
Nestle Crunch	66.47068
Nik L Nip	22.44534
Now & Later	39.44680
Payday	46.29660
Peanut M&Ms	69.48379
Pixie Sticks	37.72234
Pop Rocks	41.26551
Red vines	37.34852
Reese's Miniatures	81.86626
Reese's Peanut Butter cup	84.18029
Reese's pieces	73.43499
Reese's stuffed with pieces	72.88790
Ring pop	35.29076
Rolo	65.71629
Root Beer Barrels	29.70369
Runts	42.84914
Sixlets	34.72200
Skittles original	63.08514
Skittles wildberry	55.10370
Nestle Smarties	37.88719
Smarties candy	45.99583
Snickers	76.67378
Snickers Crisper	59.52925
Sour Patch Kids	59.86400
Sour Patch Tricksters	52.82595
Starburst	67.03763
Strawberry bon bons	34.57899
Sugar Babies	33.43755
Sugar Daddy	32.23100
Super Bubble	27.30386
Swedish Fish	54.86111
Tootsie Pop	48.98265
Tootsie Roll Juniors	43.06890
Tootsie Roll Midgies	45.73675
Tootsie Roll Snack Bars	49.65350
Trolli Sour Bites	47.17323
Twix	81.64291
Twizzlers	45.46628
Warheads	39.01190

Welch's Fruit Snacks	44.37552
Werther's Original Caramel	41.90431
Whoppers	49.52411

```
win <- candy$winpercent
win.mean <- mean(win)
round(win.mean)
```

[1] 50

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

```
x <- c(10, 2, 5, 1)
order(x)
```

[1] 4 2 3 1

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

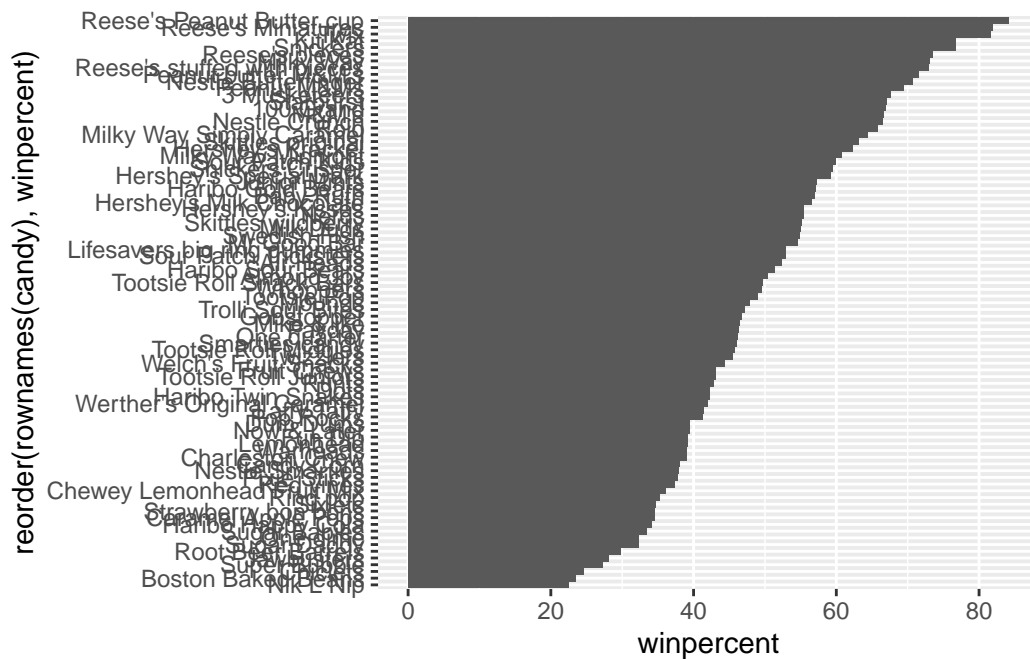
	chocolate	fruity	caramel	peanut	almond	nougat
Snickers	1	0	1		1	1
Kit Kat	1	0	0		0	0
Twix	1	0	1		0	0
Reese's Miniatures	1	0	0		1	0
Reese's Peanut Butter cup	1	0	0		1	0

	crisped	rice	wafer	hard	bar	pluribus	sugar
Snickers		0	0	1		0	0.546
Kit Kat		1	0	1		0	0.313
Twix		1	0	1		0	0.546
Reese's Miniatures		0	0	0		0	0.034
Reese's Peanut Butter cup		0	0	0		0	0.720

	price	percent	winpercent
Snickers	0.651	76.67378	
Kit Kat	0.511	76.76860	
Twix	0.906	81.64291	
Reese's Miniatures	0.279	81.86626	
Reese's Peanut Butter cup	0.651	84.18029	

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

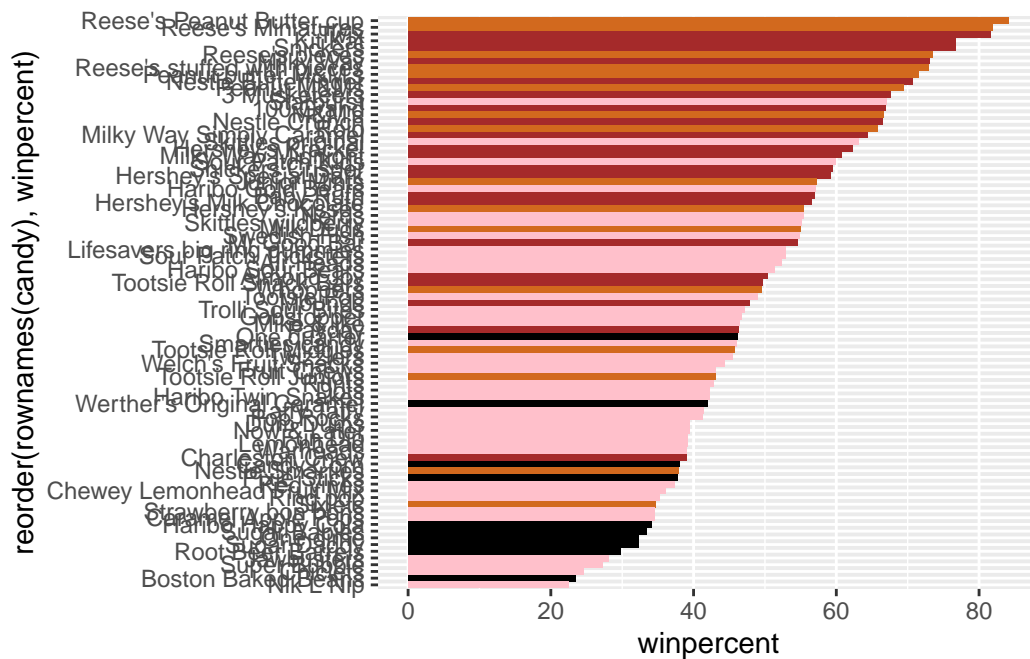
```
ggplot(candy) +
  aes(winpercent, reorder( rownames(candy), winpercent)) +
  geom_col()
```



Add some color

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

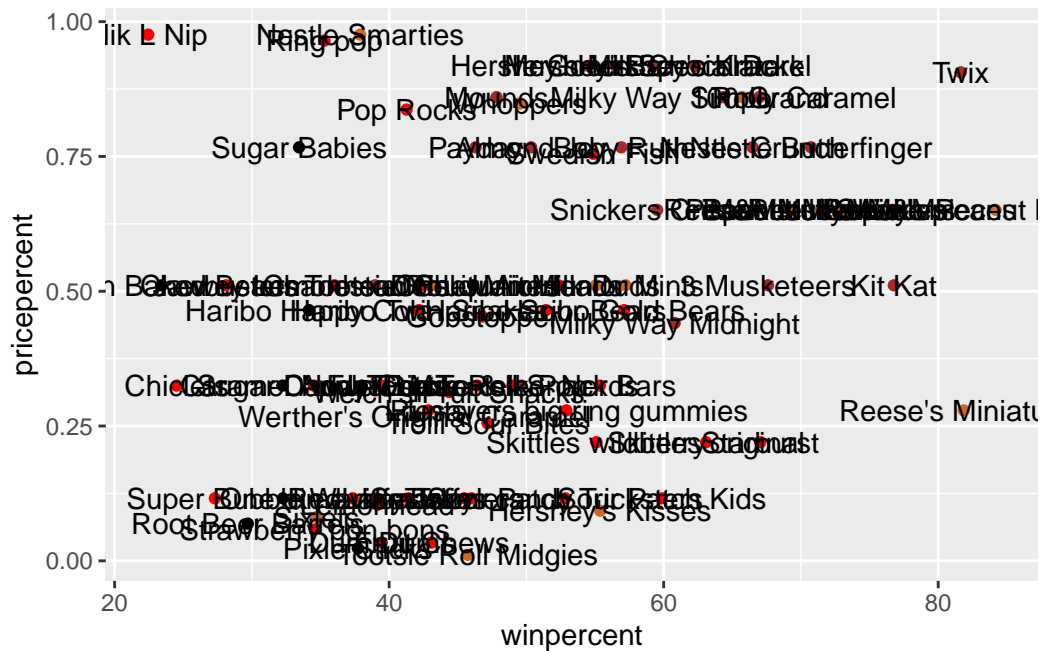
ggplot(candy) +
  aes(x=winpercent,
      y=reorder( rownames(candy), winpercent)) +
  geom_col(fill=my_cols)
```



Winpercent vs Pricepercent

```
my_cols[candy$fruity==1] <- "red"

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



Correlation

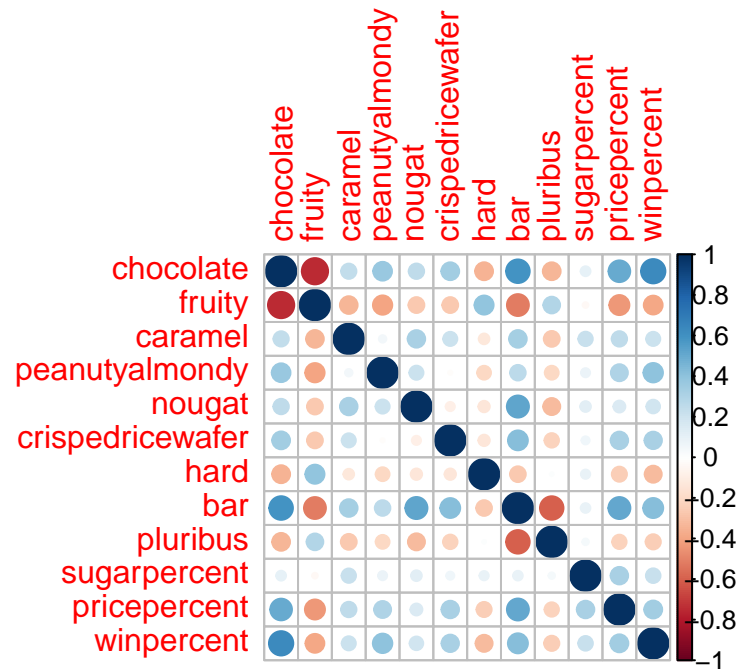
```
cij <- cor(candy)
```

```
library(corrplot)
```

```
corrplot 0.95 loaded
```



```
corrplot(cij)
```



Principal Component Analysis

The main function in base R for this `prcomp()` and we want to set `scale=TRUE` here:

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

Importance of components:

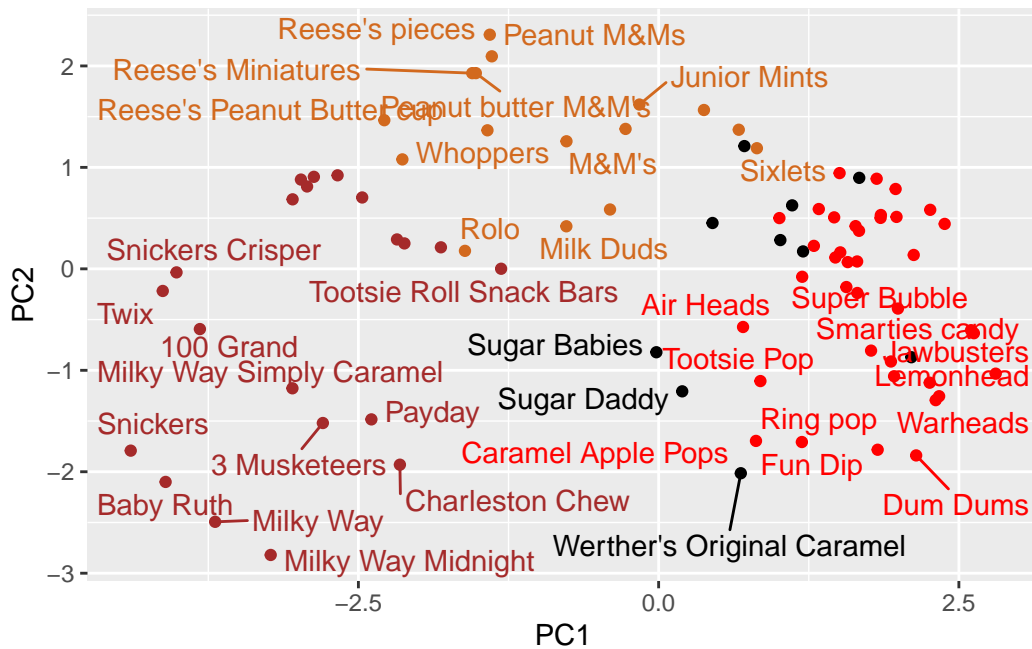
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	2.0788	1.1378	1.1092	1.07533	0.9518	0.81923	0.81530
Proportion of Variance	0.3601	0.1079	0.1025	0.09636	0.0755	0.05593	0.05539
Cumulative Proportion	0.3601	0.4680	0.5705	0.66688	0.7424	0.79830	0.85369

	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

Let's look at our first main result figure - the "PC plot" or PC1 vs PC2

```
library(ggrepel)
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



Don't forget about your variable "loadings" - how the original variables contribute to your new PCs...

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

