

# Class 10: Halloween mini project

Serena Quezada (PID: A18556865)

## Table of contents

Data Import . . . . .	1
2. What is your favorite candy? . . . . .	2
Quick overview of the dataset . . . . .	4
3. Overall Candy Rankings . . . . .	8
4. Winpercent and Pricepercent . . . . .	12
5. Exploring the correlation structure . . . . .	14
6. Principal Component Analysis . . . . .	15

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 come 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	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

```
flextable::flextable(head(candy, 10))
```

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

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
```

**2. What is your favorite candy?**

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

```
[1] 81.64291
```

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

My favorite candy is Twix and its winpercent value is 81.64%

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

```
[1] 81.64291
```

With tidyverse!

```
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 |>  
  filter(rownames(candy)=="Twix") |>  
  select(winpercent)
```

```
  winpercent  
Twix    81.64291
```

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
```

## Quick overview of the dataset

```
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	
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	

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

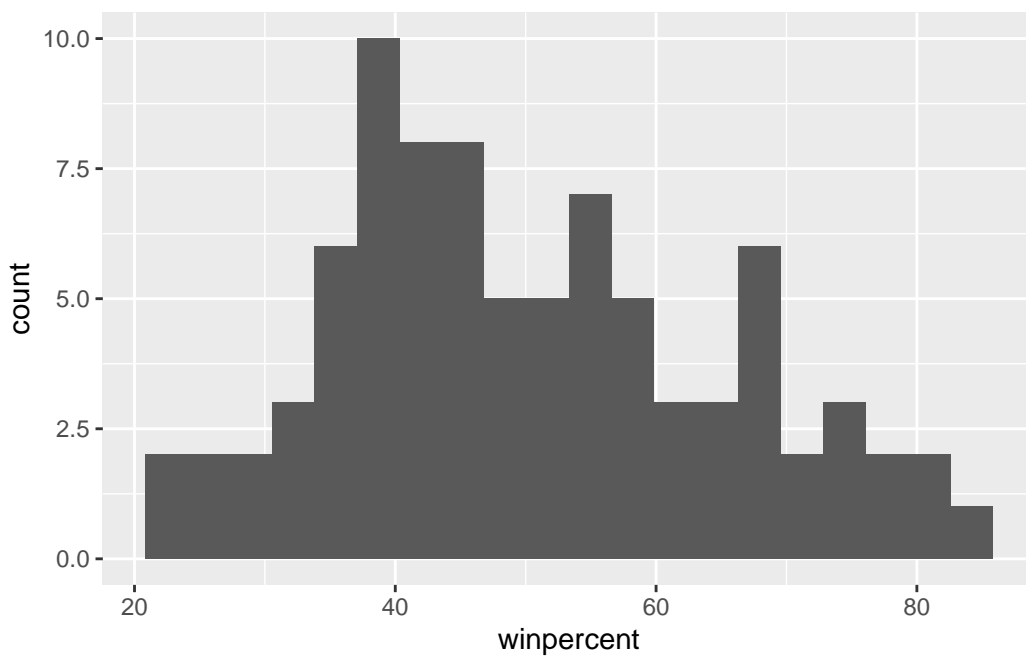
Q7. What do you think a zero and one represent for the candy\$chocolate column?

That the candy does not contain chocolate

Q8. Plot a histogram of winpercent values

```
library(ggplot2)

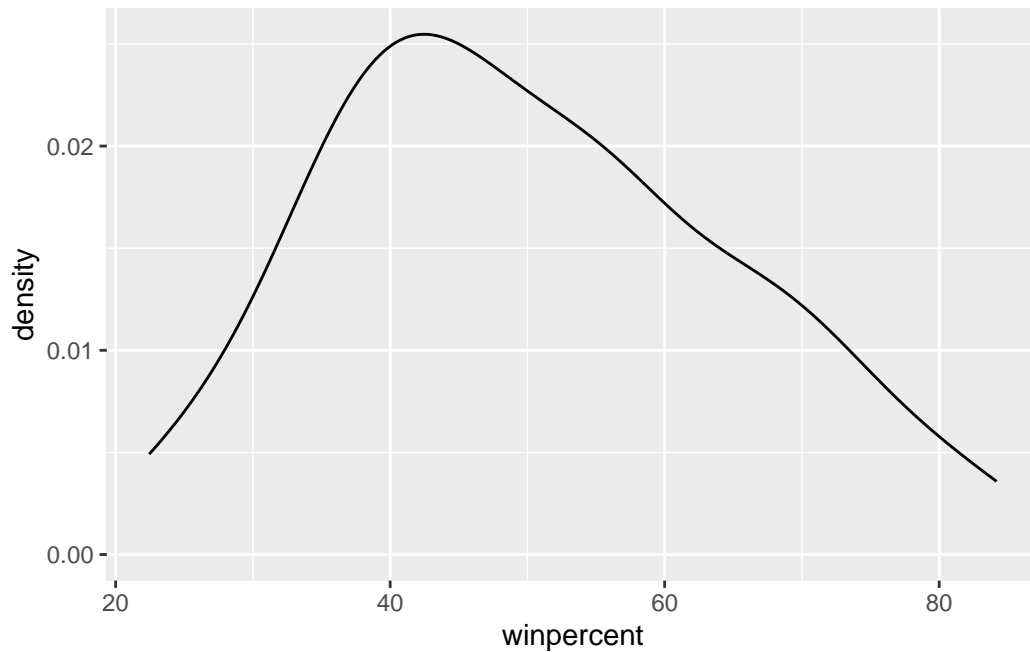
ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins = 20)
```



Q9. Is the distribution of winpercent values symmetrical?

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

Warning in geom\_density(bins = 20): Ignoring unknown parameters: `bins`



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

The mean is above 50% and the median is below 50%, it's in between 47% - 50% so it's below 50%.

Q11. On average is chocolate candy higher or lower 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 fruity candy
# 7. Compare mean winpercents of chocolate vs fruity
# 8. Pick the highest as the winner
```

Steps # 1-3:

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

```
[1] 60.92153
```

Steps # 4-6:

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

```
[1] 44.11974
```

Chocolate candy has a higher winpercent compared to fruity candy.

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

This test is statistically significant

### 3. Overall Candy Rankings

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

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

	chocolate	fruity	caramel	peanut	yalmondy	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

	crisp	edrice	wafer	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

```
x <- c(5,1,10,4)
#sort(x)
order(x)
```

[1] 2 4 1 3

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

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

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

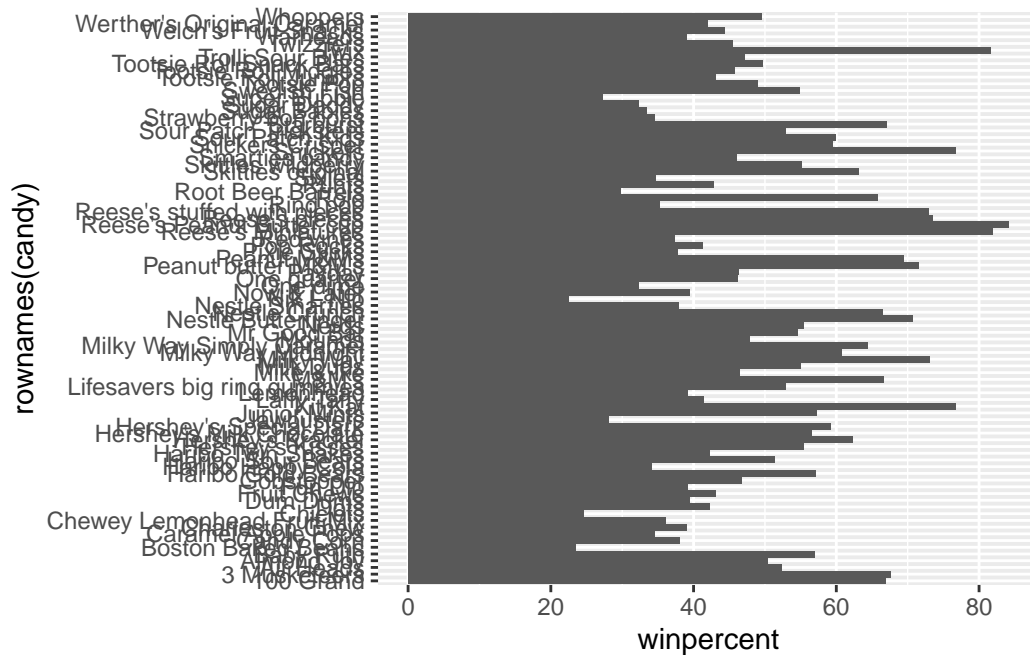
	crisped	rice	wafer	hard bar	pluribus	sugar	percent
Snickers		0	0	1	0		0.546
Kit Kat		1	0	1	0		0.313
Twix		1	0	1	0		0.546
Reese's Miniatures		0	0	0	0		0.034
Reese's Peanut Butter cup		0	0	0	0		0.720

	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	

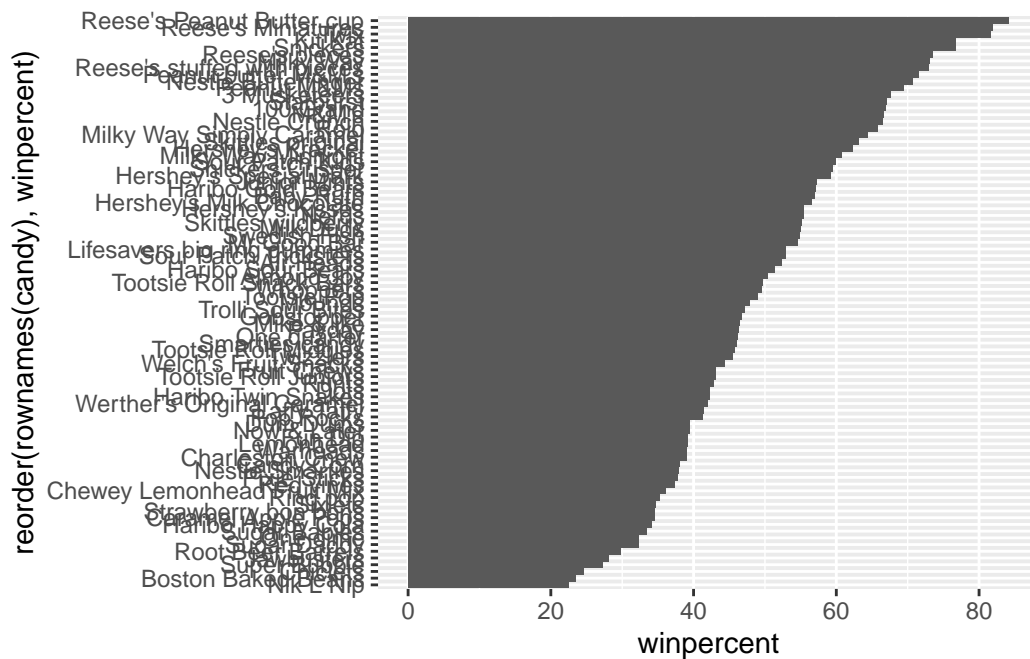
Q15. Make a first barplot of candy ranking based on winpercent values

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



Q16. This is quite ugly, use the `reorder()` function to get the bars sorted by winpercent? HINT: you can use `aes(winpercent, reorder(rownames(candy), winpercent))` to improve your plot

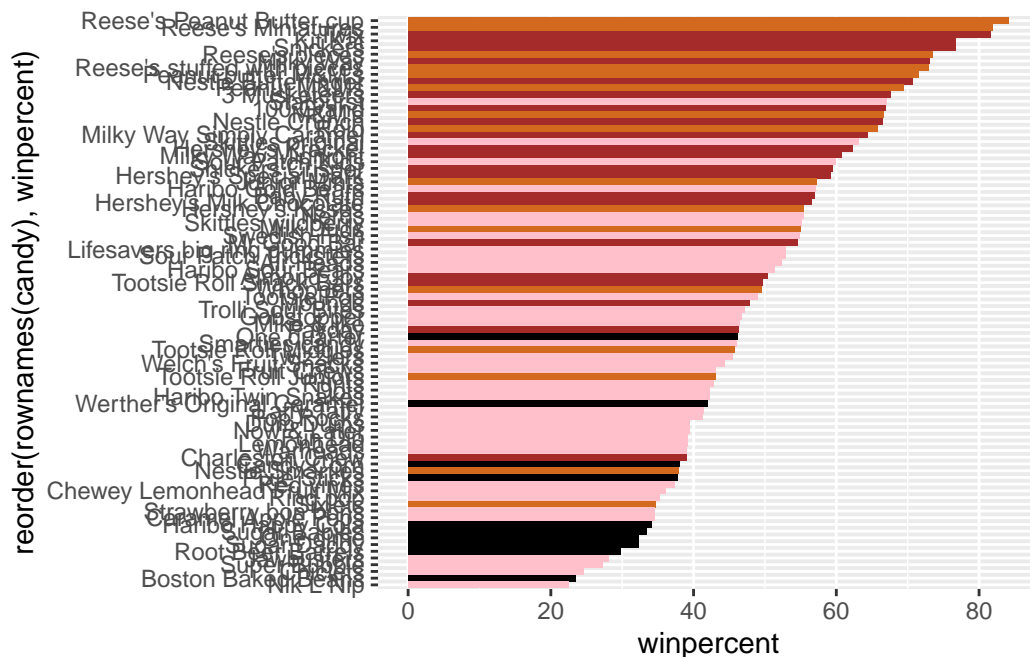
```
ggplot(candy) +
  aes(x = winpercent,
      y = reorder(rownames(candy), winpercent)) +
  geom_col()
```



Adding colors based on the “type of candy”

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

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



Q17. What is the worst ranked chocolate candy?

The worst ranked chocolate candy is sixlets

Q18. What is the best ranked fruity candy?

The best ranked fruity candy is starbust

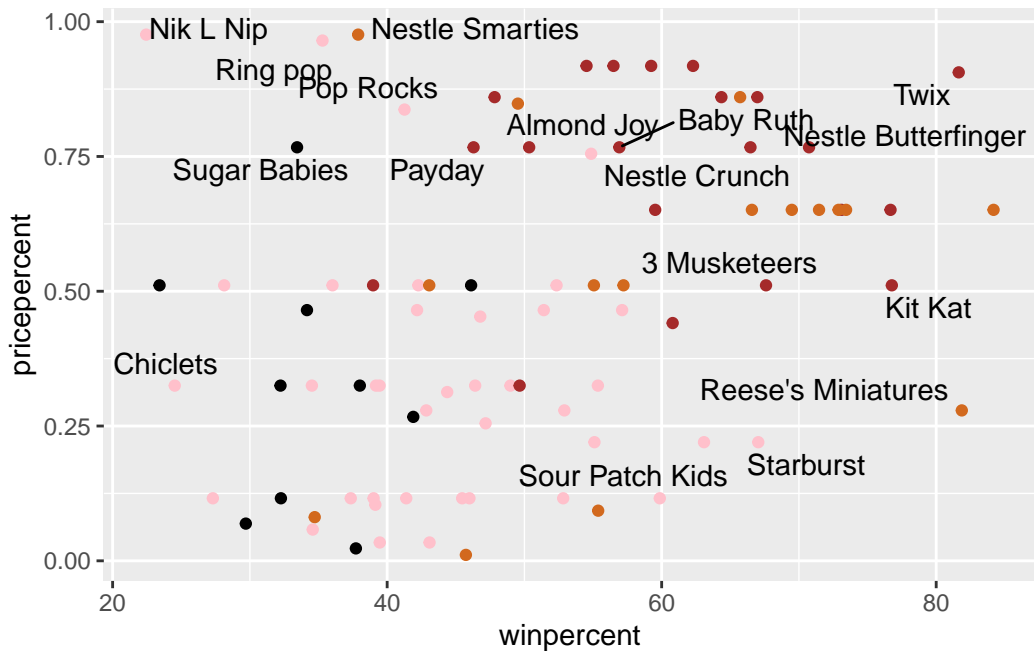
#### 4. Winpercent and Pricepercent

A plot with both variables/columns winpercent and pricepercent

```
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?

The highest ranked candy in terms of winpercent for the least money is chocolate

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

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

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

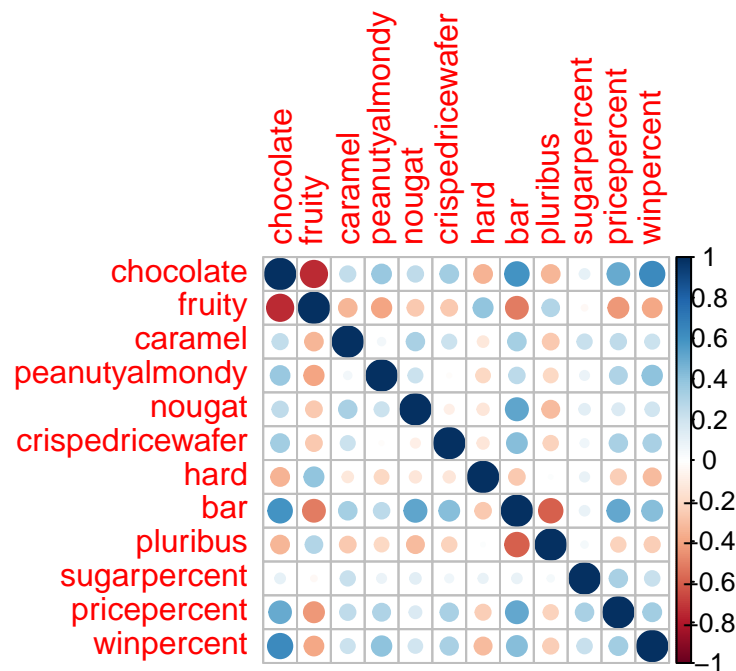
## 5. Exploring the correlation structure

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)?

The two variables that are anti-correlated are chocolate and fruity candy

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

The two variables that are positively correlated are chocolate and bar.

## 6. Principal Component Analysis

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

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

Importance of components:

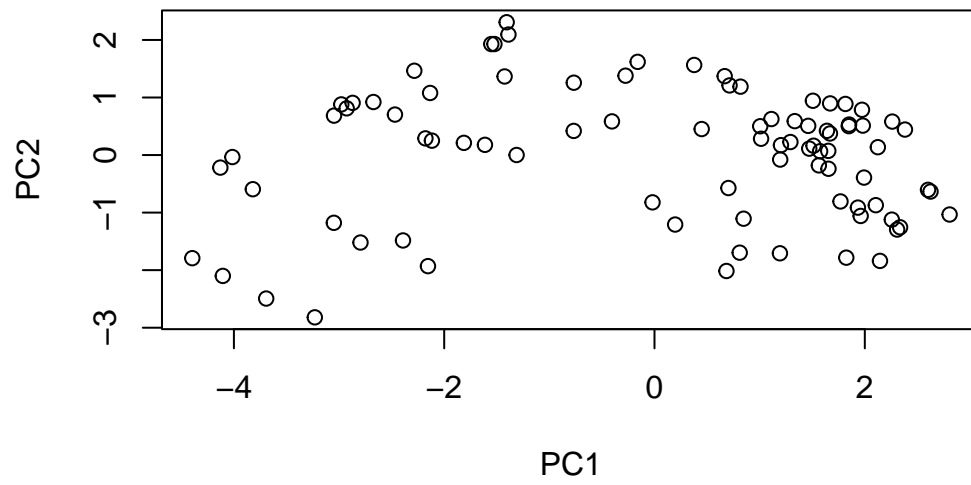
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	2.0788	1.1378	1.1092	1.07533	0.9518	0.81923	0.81530
Proportion of Variance	0.3601	0.1079	0.1025	0.09636	0.0755	0.05593	0.05539
Cumulative Proportion	0.3601	0.4680	0.5705	0.66688	0.7424	0.79830	0.85369

	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

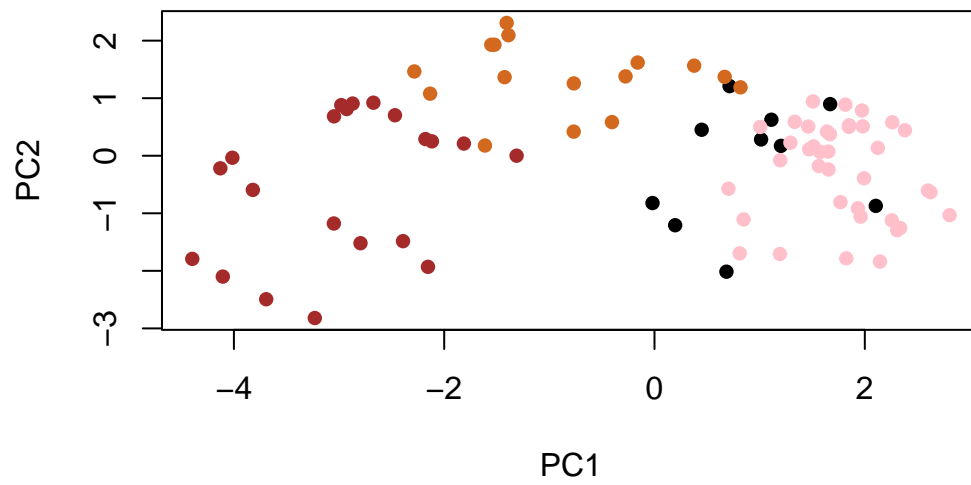
Our main PCA result figure

```
plot(pca$x [, 1:2])
```



Add some color!

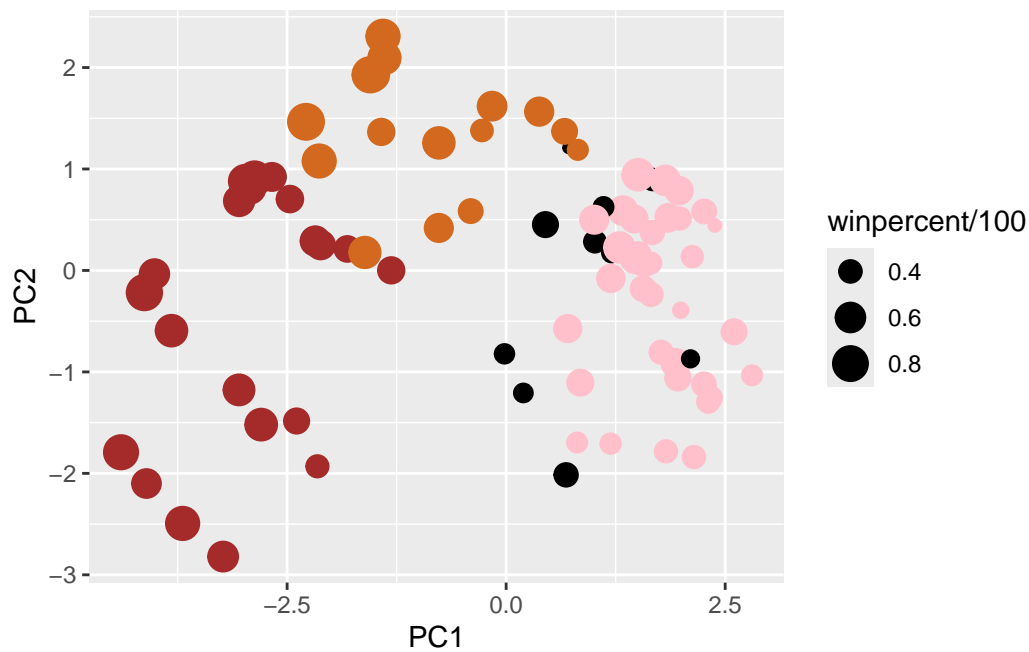
```
plot(pca$x[,1:2], col=my_cols, pch=16)
```



```
my_data <- cbind(candy, pca$x[,1:3])

p <- ggplot(my_data) +
  aes(x=PC1, y=PC2,
      size=winpercent/100,
      text=rownames(my_data),
      label=rownames(my_data)) +
  geom_point(col=my_cols)
```

p



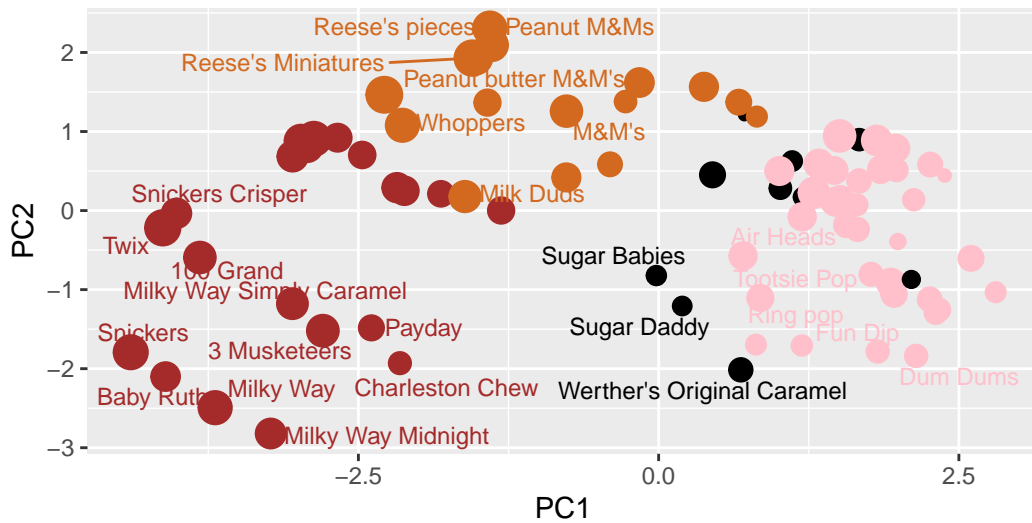
```
library(ggrepel)

p + geom_text_repel(size=3.3, col=my_cols, max.overlaps = 7) +
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
       subtitle="Colored by type: chocolate bar (dark brown), chocolate other (light brown),",
       caption="Data from 538")
```

Warning: ggrepel: 59 unlabeled data points (too many overlaps). Consider increasing max.overlaps

## Halloween Candy PCA Space

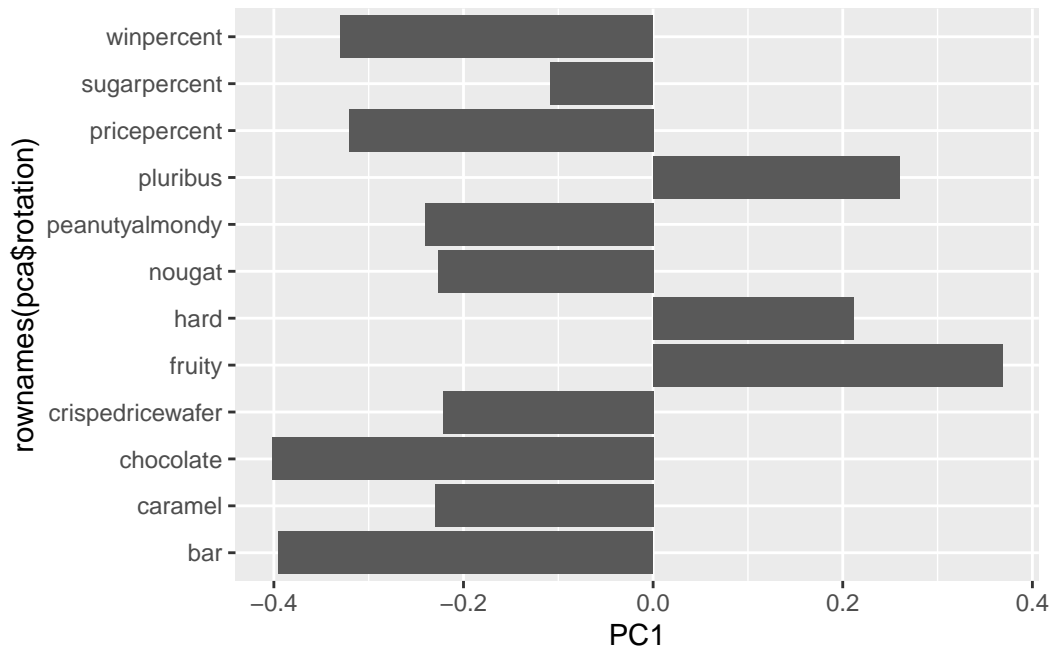
Colored by type: chocolate bar (dark brown), chocolate other (light brown),



Data from 538

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

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



Q24. What original variables are picked up strongly by PC1 in the positive direction? Do these make sense to you?

The original variables that are picked up strongly by PC1 in the positive direction are hard, fruity and pluribus candy. This makes sense to me because fruity is a highly favored type of candy.

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

Interactive plots that can be zoomed on and “brushed” over can be made with the **plotly** package. It’s 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)
```