

# Class 10: Halloween Mini-Project

Seona Patel (A69035519)

## Background

Is it Halloween today and to help get in the mood for tonight we will explore some candy preference data from 538.

### 1. Importing the candy file

```
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

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

chocolate	fruity	caramel	peanut	almond	no nut	crisp	rice wafer	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?

```
nrow(candy)
```

```
[1] 85
```

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

```
[1] 85
```

There are 85 different candy types in this dataset.

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

```
sum(candy$fruity ==1)
```

```
[1] 38
```

There are 38 fruity candy types in the dataset.

## 2. What is your favorite candy?

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

My favorite candy is M&M's and its winpercent is 66.57458%

```
candy["M&M's", ]$winpercent
```

```
[1] 66.57458
```

Q4. What is the winpercent value for "Kit Kat"?

```
candy["Kit Kat", ]$winpercent
```

```
[1] 76.7686
```

The winpercent value for "Kit kat" is 76.7686%.

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

```
candy["Tootsie Roll Snack Bars", ]$winpercent
```

```
[1] 49.6535
```

The winpercent value for Tootsie Roll Snack Bars is 49.6535%

```
library("skimr")
skim(candy)
```

```
Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

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	
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?

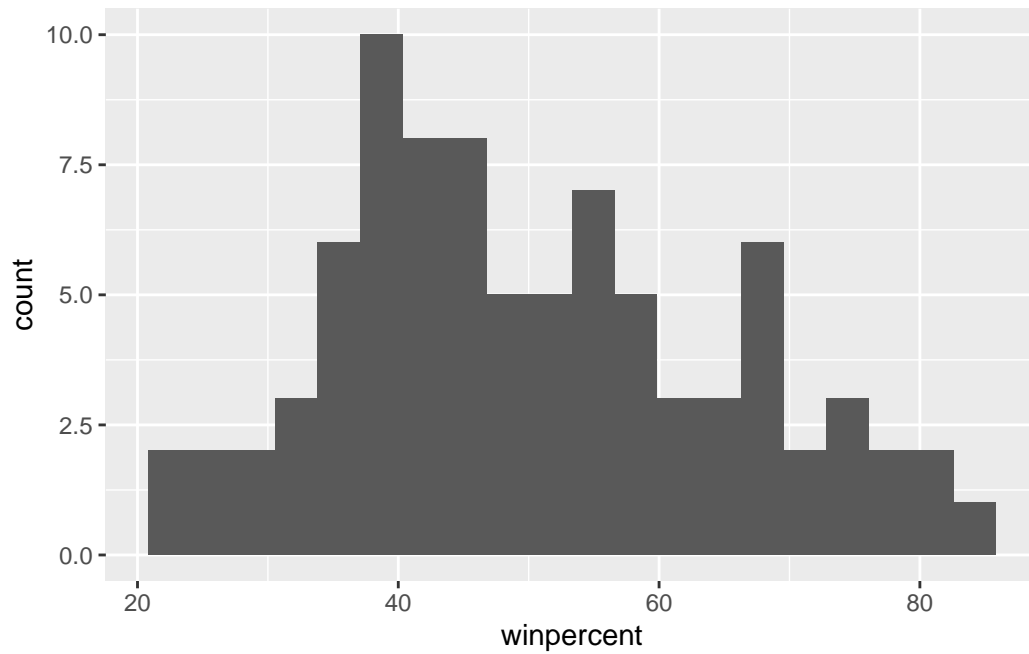
winpercent is on a 0-100 scale rather than 0-1 which is what the rest of the variables are on.

Q7. What do you think a zero and one represent for the candy\$chocolate column? Hint: look at the “Variable type” print out from the skim() function. Most variables (i.e. columns) are on the zero to one scale but not all. Some columns such as chocolate are exclusively either zero or one values.

0 and 1 are logicals, so 0 means it is not a chocolate type and 1 means it is a chocolate type.

Q8. Plot a histogram of winpercent values

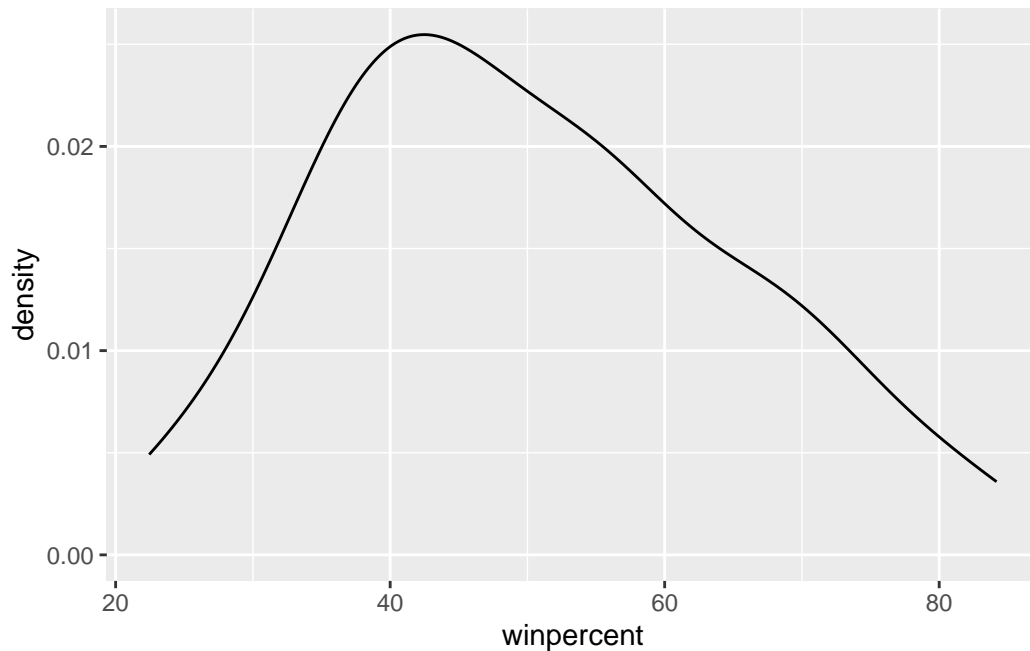
```
library(ggplot2)
ggplot(candy,)+
  aes(winpercent) +
  geom_histogram(bins=20)
```



Q9. Is the distribution of winpercent values symmetrical?

No, the distribution looks right skewed rather than symmetrical.

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



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

It depends on how you measure centrality, if you use the mean it is above 50% but if you use median 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?

```
# 1. Find all chocolate candy in the dataset
choc.inds <- as.logical(candy$chocolate)
choc.candy <- candy[choc.inds,]
# 2. Extract their `winpercent` values
choc.win <- choc.candy$winpercent
# 3. Find the mean of these values
```

```
choc.mean <- mean(choc.win)
# 4-6. DO the same for fruity candy

fruity.inds <- as.logical(candy$fruity)
fruity.candy <- candy[fruity.inds,]
fruity.wins <- fruity.candy$winpercent
fruity.mean <- mean(fruity.wins)
# Which mean value is higher

choc.mean
```

```
[1] 60.92153
```

```
fruity.mean
```

```
[1] 44.11974
```

Chocolate candy is ranked higher than fruity candy on average.

Q12. Is this difference statistically significant?

```
t.test(choc.win, fruity.wins)
```

Welch Two Sample t-test

```
data:  choc.win and fruity.wins
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
```

Yes this is statistically significant.

### 3. Overall Candy Rankings

Let's use the base R `order()` function together with `head()` to sort the whole dataset by `winpercent`. Or if you have been getting into the tidyverse and the `dplyr` package you can use the `arrange()` function together with `head()` to do the same thing and answer the following questions:

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

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

	chocolate	fruity	caramel	peanut	almond	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

	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

	winpercent
Nik L Nip	22.44534
Boston Baked Beans	23.41782
Chiclets	24.52499
Super Bubble	27.30386
Jawbusters	28.12744

The least liked candy types in this set are Nik L Nip, Boston Baked Beans, Chiclets, Super Bubble, and Jawbusters

Q14. What are the top 5 all time favorite candy types out of this set? Hint: Using base R we could use `head(candy[order(candy$winpercent),], n=5)`, whilst using `dplyr` we have: `candy %>% arrange(winpercent) %>% head(5)`. Which approach do you prefer and why?

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

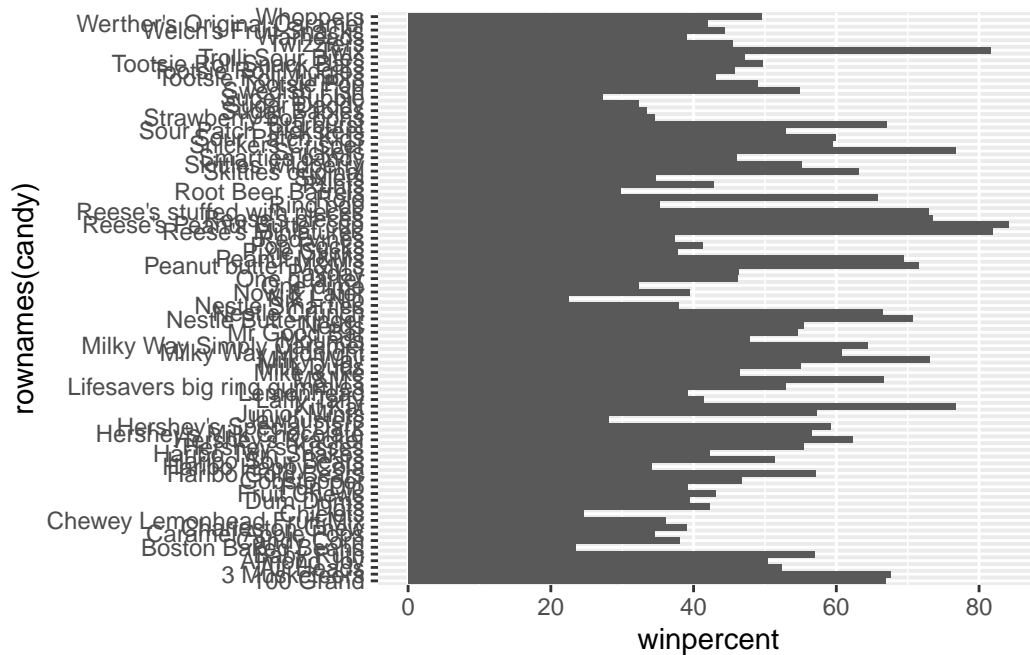
	chocolate	fruity	caramel	peanut	almond	nougat		
Reese's Peanut Butter cup	1	0	0		1	0		
Reese's Miniatures	1	0	0		1	0		
Twix	1	0	1		0	0		
Kit Kat	1	0	0		0	0		
Snickers	1	0	1		1	1		
	crisped	rice	wafer	hard	bar	pluribus	sugar	percent
Reese's Peanut Butter cup		0	0	0		0		0.720
Reese's Miniatures		0	0	0		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					

The top 5 favorite candy types are Reese's Peanut Butter cup, Reese's Miniatures, Twix, Kit Kat, And Snickers

To examine more of the dataset in this vain we can make a barplot to visualize the overall rankings. We will use an iterative approach to building a useful visulization by getting a rough starting plot and then refining and adding useful details in a stepwise process.

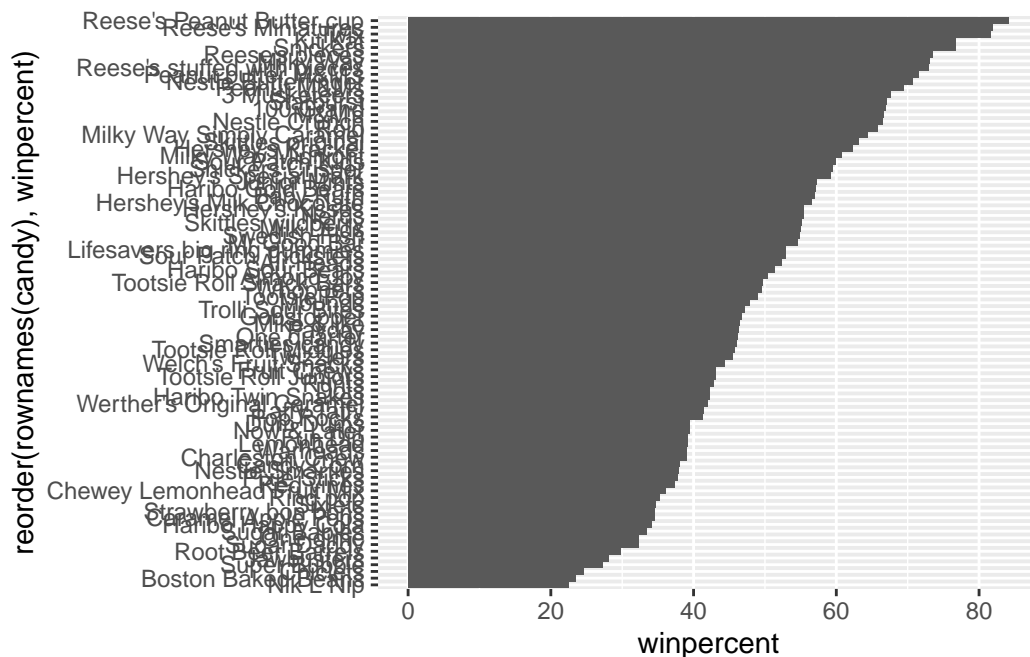
Q15. Make a first barplot of candy ranking based on winpercent values. HINT:  
Use the `aes(winpercent, rownames(candy))` for your first ggplot like so

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

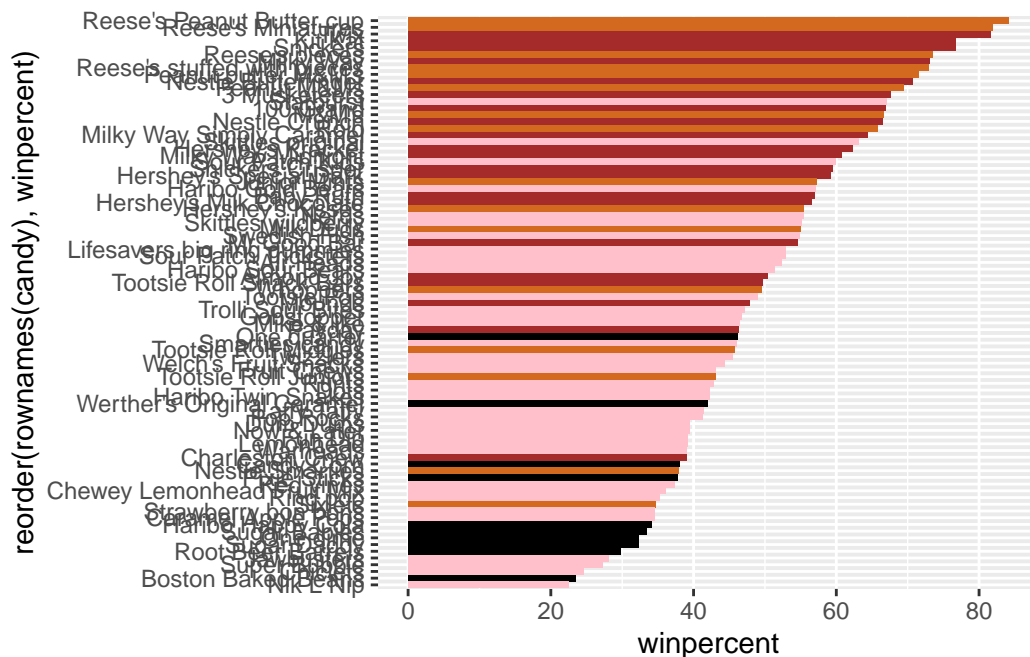


Time to add some useful color Let's setup a color vector (that signifies candy type) that we can then use for some future plots. We start by making a vector of all black values (one for each candy). Then we overwrite chocolate (for chocolate candy), brown (for candy bars) and red (for fruity candy) values.

```
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"
```

Now let's try our barplot with these colors. Note that we use fill=my\_cols for geom\_col(). Experiment to see what happens if you use col=mycols.

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



Now, for the first time, using this plot we can answer questions like: > Q17. What is the worst ranked chocolate candy?

The worst ranked chocolate is Sixlets.

Q18. What is the best ranked fruity candy?

The best ranked fruity candy is Starburst.

#### 4. Taking a look at pricepercent

What about value for money? What is the the best candy for the least money? One way to get at this would be to make a plot of winpercent vs the pricepercent variable. The pricepercent variable records the percentile rank of the candy's price against all the other candies in the dataset. Lower vales are less expensive and high values more expensive.

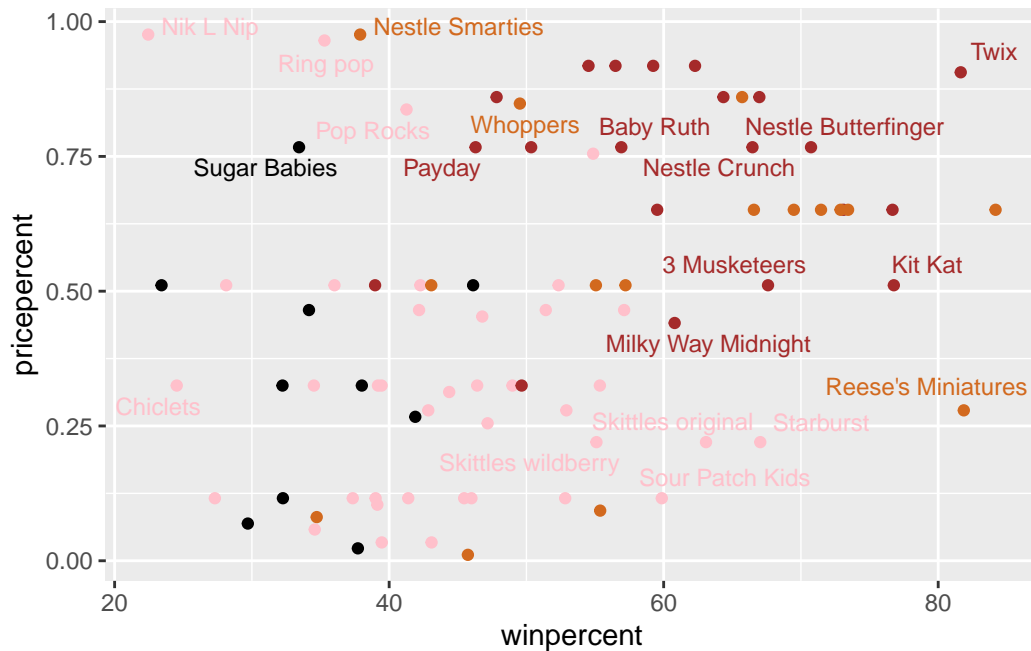
To this plot we will add text labels so we can more easily identify a given candy. There is a regular `geom_label()` that comes with `ggplot2`. However, as there are quite a few candys in our dataset lots of these labels will be overlapping and hard to read. To help with this we can use the `geom_text_repel()` function from the `ggrepel` package.

```
library(ggrepel)

# How about a plot of price vs win
```

```
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols, size=3.3, max.overlaps = 5)
```

Warning: ggrepel: 65 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

Of the least expensive candies, Tootsie Roll Midgies has the highest winpercent, making it the greatest bang for your buck.

Q20. What are the top 5 most expensive candy types in the dataset and of these which is the least popular? Hint: To see which candy is the most expensive (and which is the least expensive) we can order() the dataset by pricepercent.

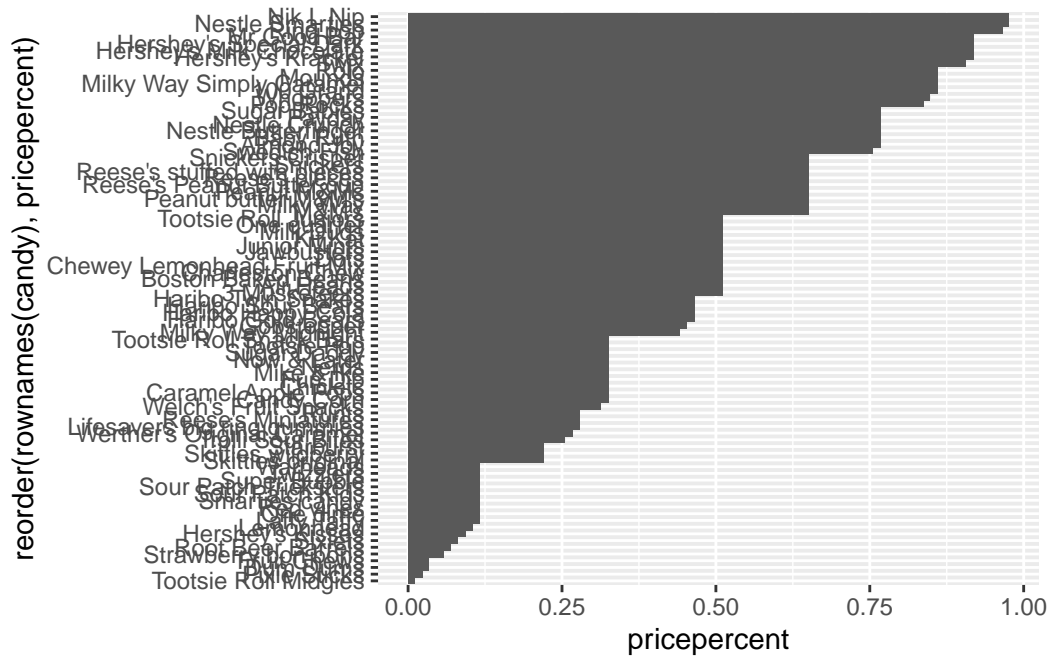
```
ord <- order(candy$pricepercent, decreasing = TRUE)
head( candy[ord,c(11,12)], n=5 )
```

	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

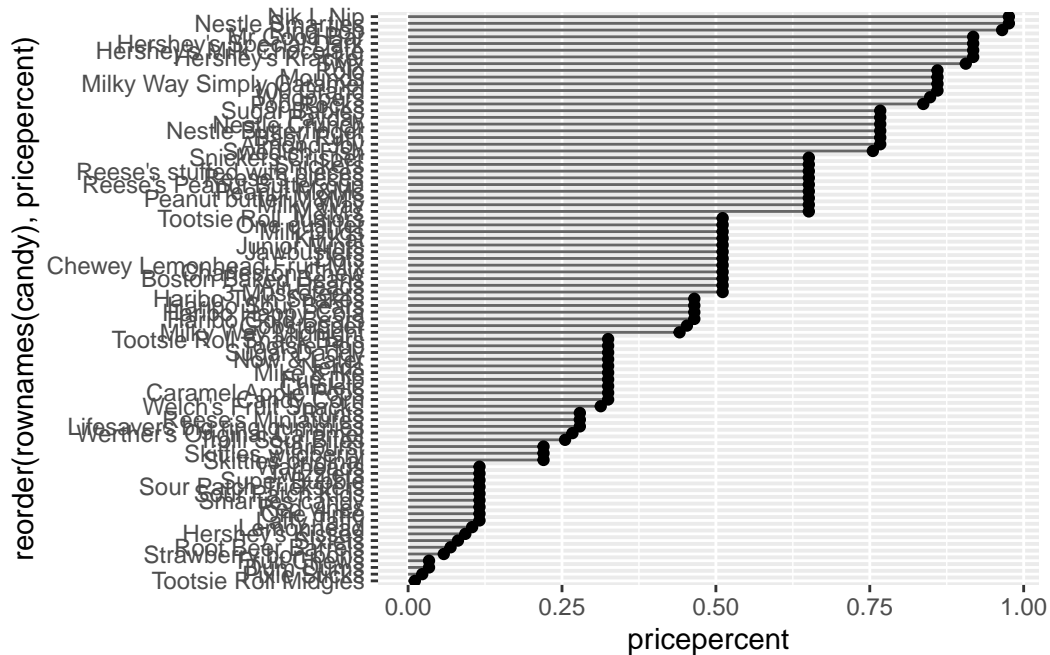
The top 5 most expensive candies are Nik L Nip, Nestle Smarties, Ring pop, Hershey's Krackel, and Hershey's Milk Chocolate. Of these, the least popular is Nik L Nip.

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_col()
```



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



## Correlation

```
cor(candy)
```

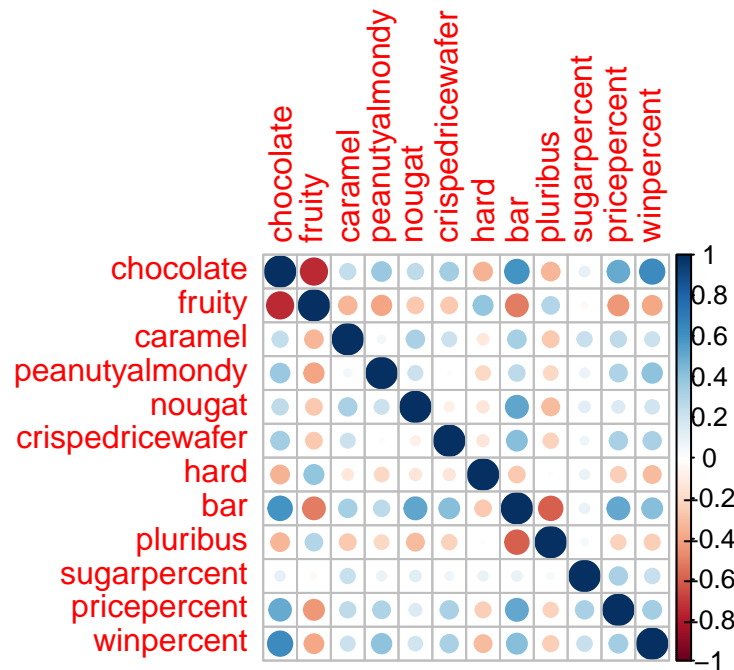
	chocolate	fruity	caramel	peanutyalmondy	nougat
chocolate	1.0000000	-0.74172106	0.24987535	0.37782357	0.25489183
fruity	-0.7417211	1.00000000	-0.33548538	-0.39928014	-0.26936712
caramel	0.2498753	-0.33548538	1.00000000	0.05935614	0.32849280
peanutyalmondy	0.3778236	-0.39928014	0.05935614	1.00000000	0.21311310
nougat	0.2548918	-0.26936712	0.32849280	0.21311310	1.00000000
crispedricewafer	0.3412098	-0.26936712	0.21311310	-0.01764631	-0.08974359
hard	-0.3441769	0.39067750	-0.12235513	-0.20555661	-0.13867505
bar	0.5974211	-0.51506558	0.33396002	0.26041960	0.52297636
pluribus	-0.3396752	0.29972522	-0.26958501	-0.20610932	-0.31033884
sugarpercent	0.1041691	-0.03439296	0.22193335	0.08788927	0.12308135
pricepercent	0.5046754	-0.43096853	0.25432709	0.30915323	0.15319643
winpercent	0.6365167	-0.38093814	0.21341630	0.40619220	0.19937530
	crispedricewafer	hard	bar	pluribus	
chocolate	0.34120978	-0.34417691	0.59742114	-0.33967519	
fruity	-0.26936712	0.39067750	-0.51506558	0.29972522	

caramel	0.21311310	-0.12235513	0.33396002	-0.26958501
peanutyalmondy	-0.01764631	-0.20555661	0.26041960	-0.20610932
nougat	-0.08974359	-0.13867505	0.52297636	-0.31033884
crispedricewafer	1.00000000	-0.13867505	0.42375093	-0.22469338
hard	-0.13867505	1.00000000	-0.26516504	0.01453172
bar	0.42375093	-0.26516504	1.00000000	-0.59340892
pluribus	-0.22469338	0.01453172	-0.59340892	1.00000000
sugarpercent	0.06994969	0.09180975	0.09998516	0.04552282
pricepercent	0.32826539	-0.24436534	0.51840654	-0.22079363
winpercent	0.32467965	-0.31038158	0.42992933	-0.24744787
	sugarpercent	pricepercent	winpercent	
chocolate	0.10416906	0.5046754	0.6365167	
fruity	-0.03439296	-0.4309685	-0.3809381	
caramel	0.22193335	0.2543271	0.2134163	
peanutyalmondy	0.08788927	0.3091532	0.4061922	
nougat	0.12308135	0.1531964	0.1993753	
crispedricewafer	0.06994969	0.3282654	0.3246797	
hard	0.09180975	-0.2443653	-0.3103816	
bar	0.09998516	0.5184065	0.4299293	
pluribus	0.04552282	-0.2207936	-0.2474479	
sugarpercent	1.00000000	0.3297064	0.2291507	
pricepercent	0.32970639	1.0000000	0.3453254	
winpercent	0.22915066	0.3453254	1.0000000	

```
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)

Fruity and chocolate are anti-correlated.

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

winpercent and chocolate are most strongly positively correlated.

## 6. 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=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

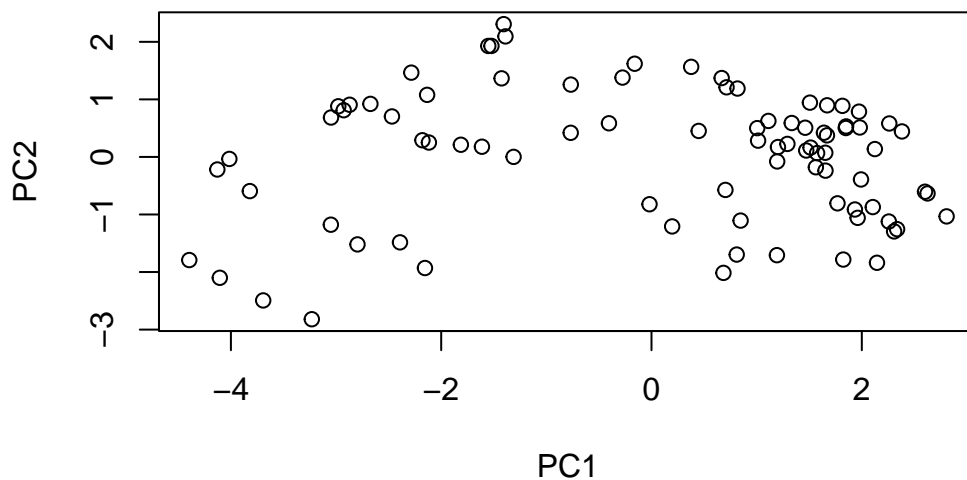
  

	PC8	PC9	PC10	PC11	PC12
--	-----	-----	------	------	------

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

Now we can plot our main PCA score plot of PC1 vs PC2.

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

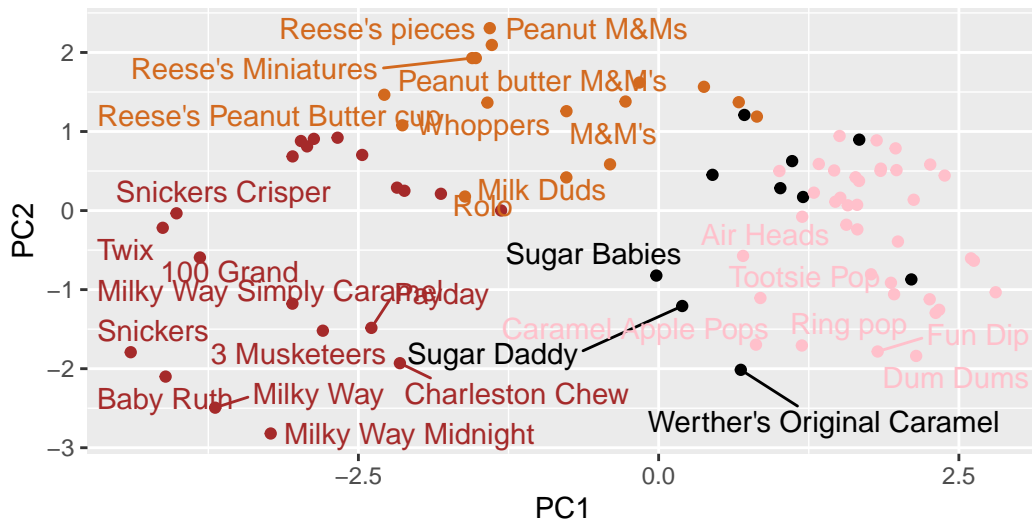


```
ggplot(pca$x) +
  aes(PC1, PC2, label=rownames(pca$x)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols)+
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
        subtitle="Colored by type: chocolate bar (dark brown), chocolate other (light brown),",
        caption="Data from 538")
```

Warning: ggrepel: 56 unlabeled data points (too many overlaps). Consider increasing max.overlaps

## Halloween Candy PCA Space

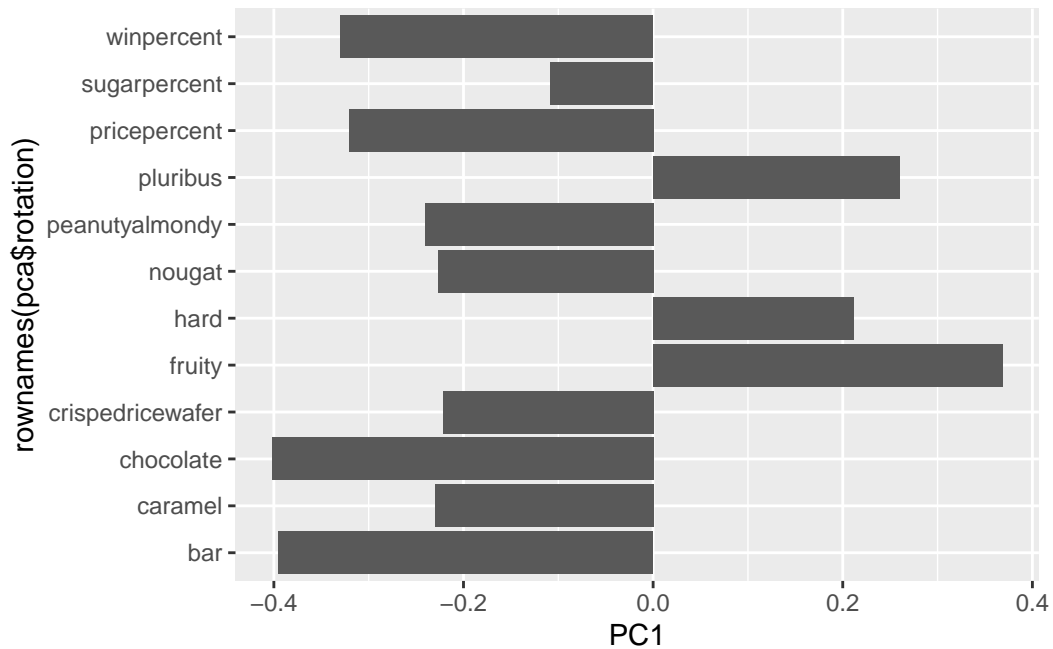
Colored by type: chocolate bar (dark brown), chocolate other (light brown),



Data from 538

Dont forget about your variable “loadings” - how the original variables contribute to your new PC's...

```
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? HINT. pluribus means the candy comes in a bag or box of multiple candies.

The variables fruity, hard, and pluribus are picked up strongly by PC1 in the positive direction. This makes sense as the fruity candy tends to be hard and come in a bag or box of multiple candies.