Class 10 Halloween Mini Project

Carolina Merino (PID: 14484883)

Table of contents

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

1. Importing candy data

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

```
Candy_file <- read.csv("candy-data.csv")
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
```

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

There are a total of **85 candy types** included in this dataset.

```
nrow(candy)
```

```
[1] 85
```

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

There are **38 fruity candy types.**

```
sum(candy[,2])
```

```
[1] 38
```

2. What is your favorite candy?

The winpercent variable shows how often a candy wins when compared to another at random, with higher values meaning greater popularity. We can find Twix's winpercent by using its name as the row label in the dataset.

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

For my favorite candy in the dataset is **Baby Ruth, and its winpercent value is 56.91%.**

```
candy["Baby Ruth", ]$winpercent
```

```
[1] 56.91455
```

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

The winpercent value for Kit Kat is **76.77%.**

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

```
[1] 76.7686
```

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

The winpercent value for Tootsie Roll Snack Bars is **49.65%.**

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

```
[1] 49.6535
```

For Another way to pull winpercent using dplyr package:

```
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)=="Boston Baked Beans") |>
  select(winpercent)
```

```
                  winpercent
Boston Baked Beans   23.41782
```

*let's install skimr package in console to use following code

```
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 appears to be on a different scale compared to most of the other columns.

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

For the chocolate column, 0 signifies the candy lacks chocolate (FALSE) and 1 signifies it contains chocolate (TRUE).

- let's load tidyyverse for the next section

```
library(tidyverse)
```

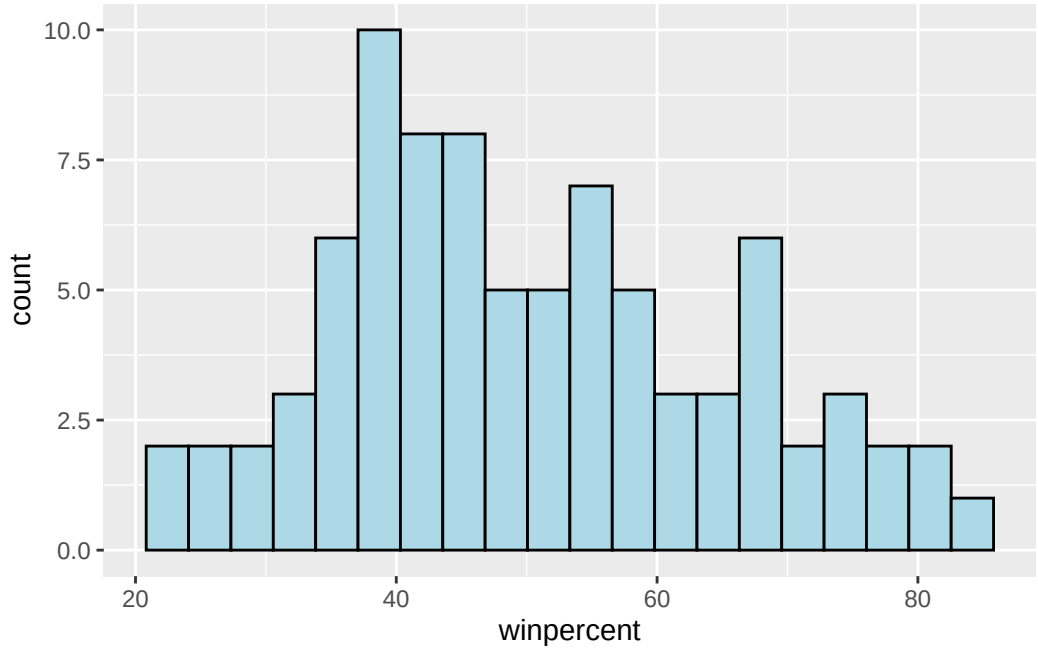
```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.1     v readr     2.1.5
v ggplot2   4.0.0     v stringr   1.5.2
v lubridate 1.9.4     v tibble    3.3.0
v purrr     1.1.0     v tidyr     1.3.1
-- Conflicts ------------------------------------------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

```
library(ggplot2)
```

Q8. Plot a histogram of winpercent values

Creating a histogram of the winpercent values

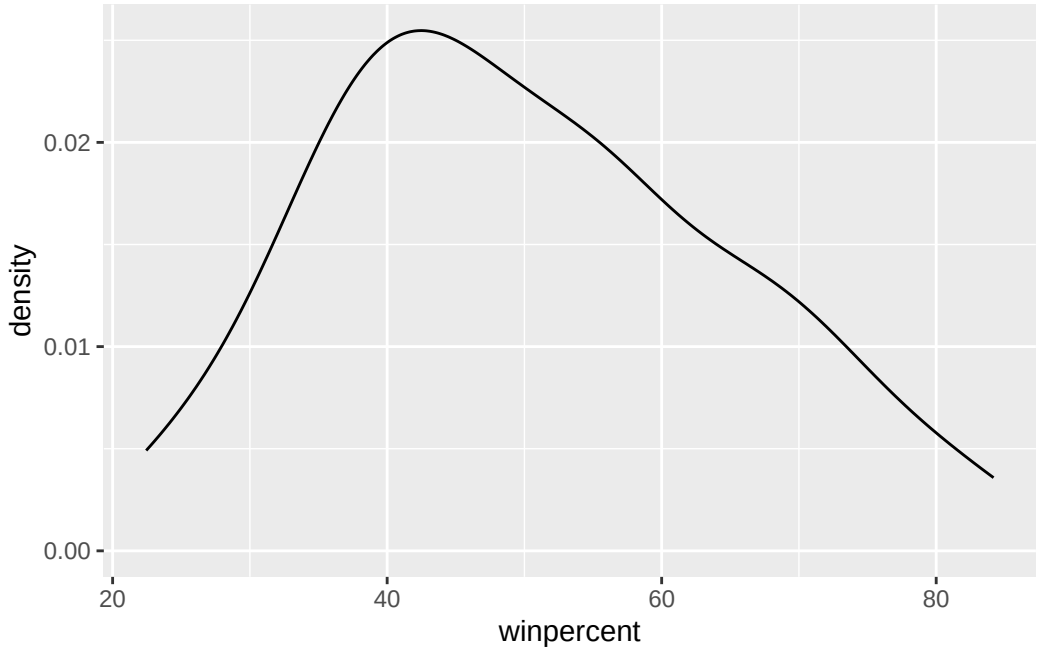
```
ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins=20, color="black", fill="lightblue")
```



Q9. Is the distribution of winpercent values symmetrical?

The graph shows that the distribution of winpercent value is **not*** symmetrical.

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



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

On average, the winpercent values fall below 50%, around 47.83%.

```
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?

The chocolates are ranked higher on average than fruity candies (60.92% vs. 44.12%).

```
mean(candy$winpercent[as.logical(candy$chocolate)])
```

```
[1] 60.92153
```

```
mean(candy$winpercent[as.logical(candy$fruity)])
```

```
[1] 44.11974
```

```
mean(candy$winpercent[as.logical(candy$chocolate)]) > mean(candy$winpercent[as.logical(candy$
```

```
[1] TRUE
```

Q12. Is this difference statistically significant?

With a p-value of 2.871e-08, which falls below 0.05, we can conclude that the difference is statistically significant.

```
t.test(candy$winpercent[as.logical(candy$chocolate)], candy$winpercent[as.logical(candy$fruit
```

```
    Welch Two Sample t-test

data:  candy$winpercent[as.logical(candy$chocolate)] and candy$winpercent[as.logical(candy$fr
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
```

3. Overall Candy Rankings

We can sort the entire dataset by winpercent using base R's `order()` and `head()` functions. Alternatively, the `arrange()` function from the tidyverse's dplyr package achieves the same result.

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

The candies with the lowest winpercent are Nik L Nip, Boston Baked Beans, Chiclets, Super Bubble, and Jawbusters.

```
head(candy[order(candy$winpercent),], n=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?

The five most popular candies in the dataset are **Snickers, Kit Kat, Twix, Reese's Miniatures, and Reese's Peanut Butter Cup.**

```
tail(candy[order(candy$winpercent),], n=5)
```

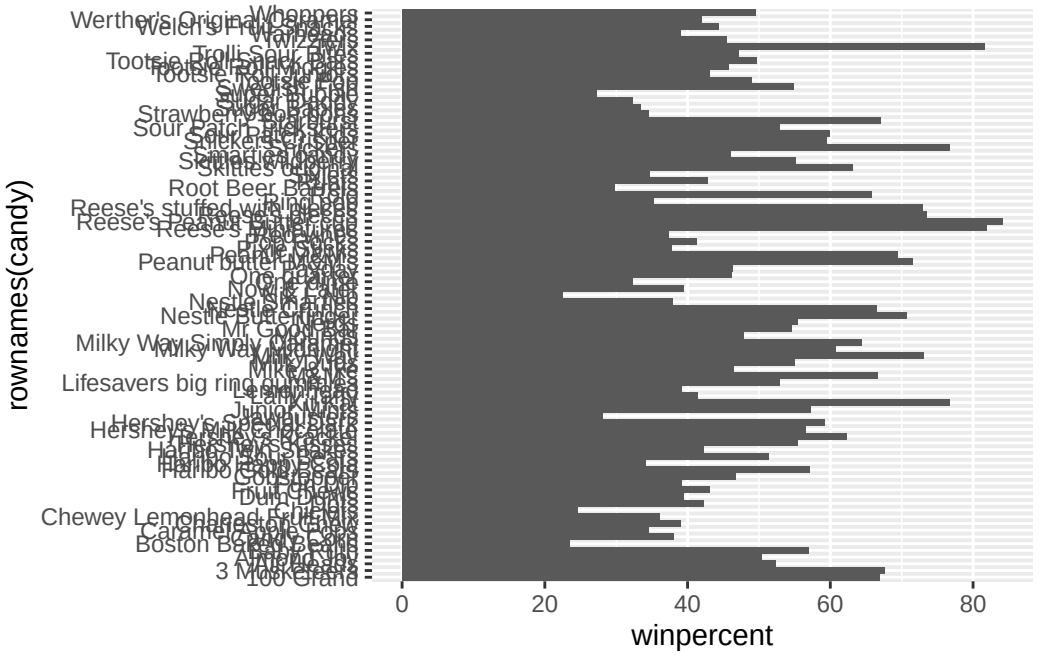
```
                        chocolate fruity caramel peanutyalmondy nougat
Snickers                        1      0       1              1      1
Kit Kat                         1      0       0              0      0
Twix                            1      0       1              0      0
Reese's Miniatures              1      0       0              1      0
Reese's Peanut Butter cup       1      0       0              1      0
                        crispedricewafer hard bar pluribus sugarpercent
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
                         pricepercent 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.

Creating the Bar Plot..

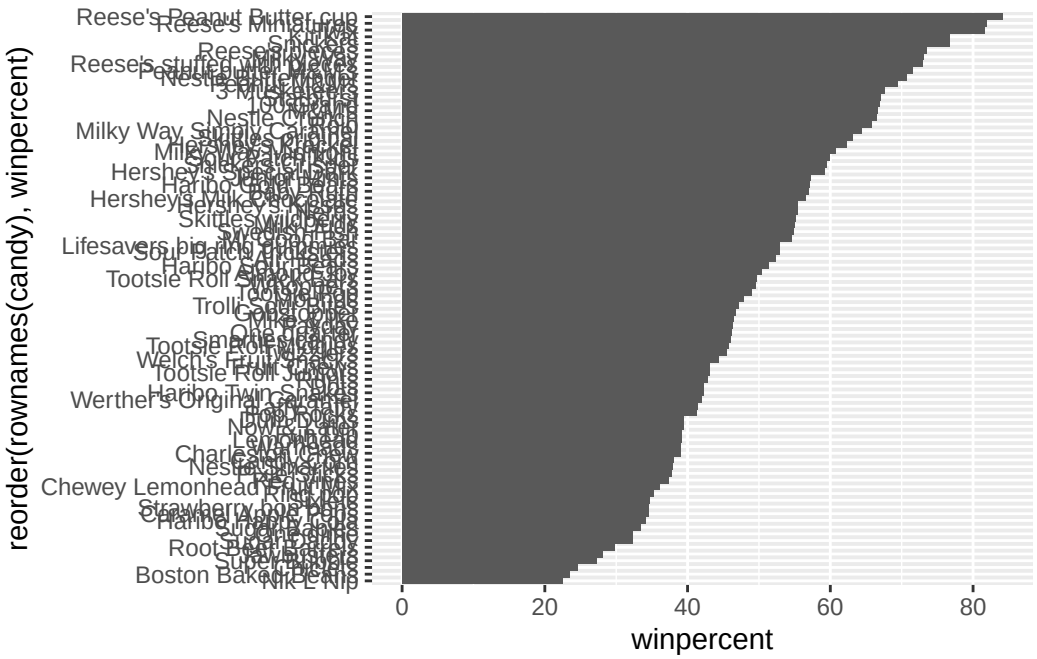
```
ggplot(candy) +
  aes(winpercent, rownames(candy)) +
  geom_col()
```



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

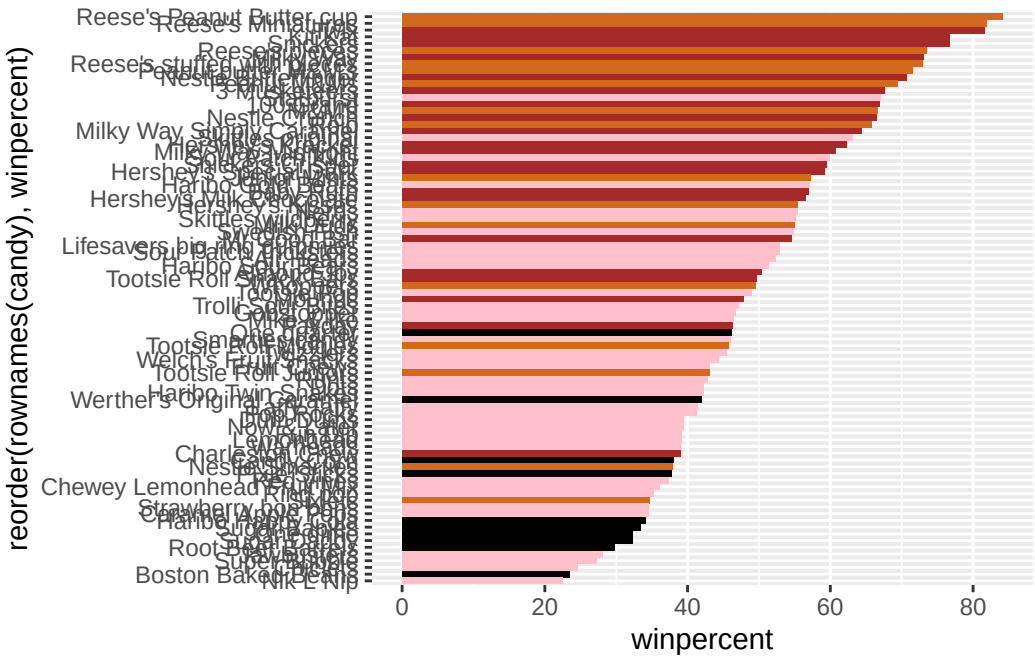
Making a better bar plot:

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



**Now, let's go ahead and add some color to out plot:

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



Q17. What is the worst ranked chocolate candy?

Among the chocolate candies, *Sixlets ranks the lowest.*

Q18. What is the best ranked fruity candy?

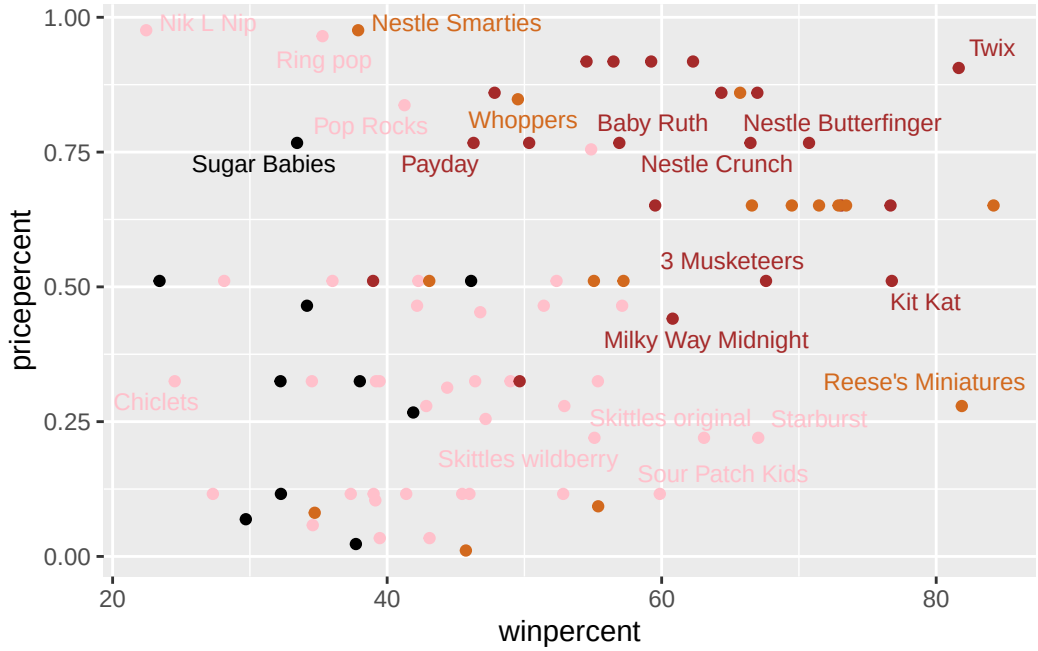
Starburst holds the highest ranking among the fruit-flavored candies.

4. Taking a look at pricepercent

To evaluate value for money, we can plot winpercent against pricepercent, which ranks candy prices relative to the dataset. Since many labels may overlap, we can use `geom_text_repel()` from the ggrepel package to make the candy names readable.

```
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?

Reese's Miniatures offers the most 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?

The priciest five candies are *Nik L Nip, Nestle Smarties, Ring Pop, Hershey's Krackel, and Hershey's Milk Chocolate, with Nik L Nip* being the least liked among them.

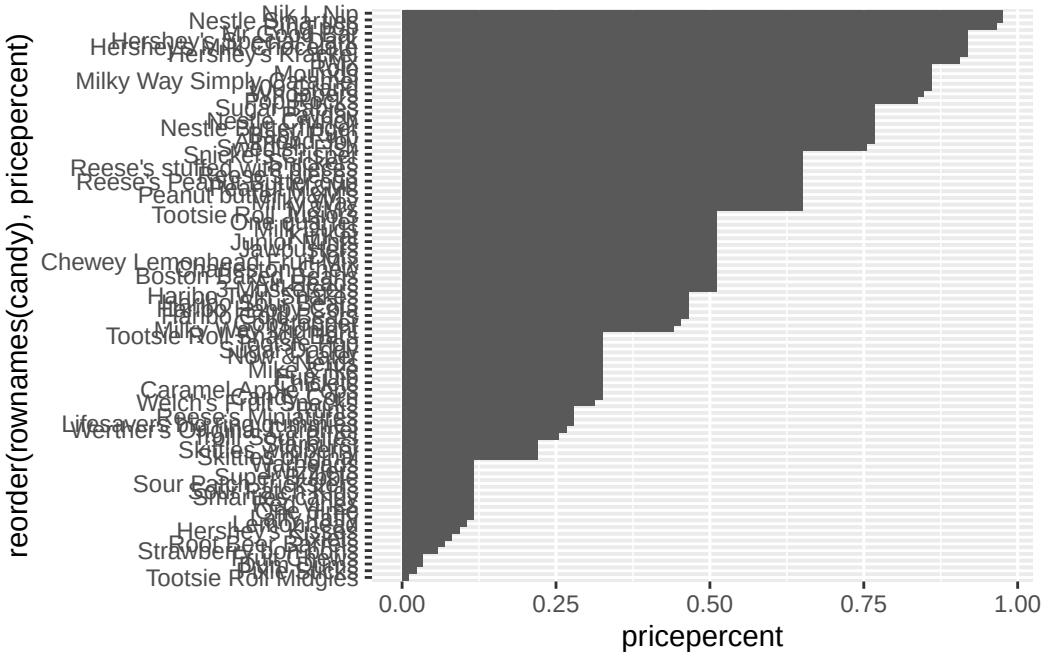
```
ord <- order(candy$pricepercent, decreasing = TRUE)
head( candy[ord,c(11,12)], n=5 )
```

```
                      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
```

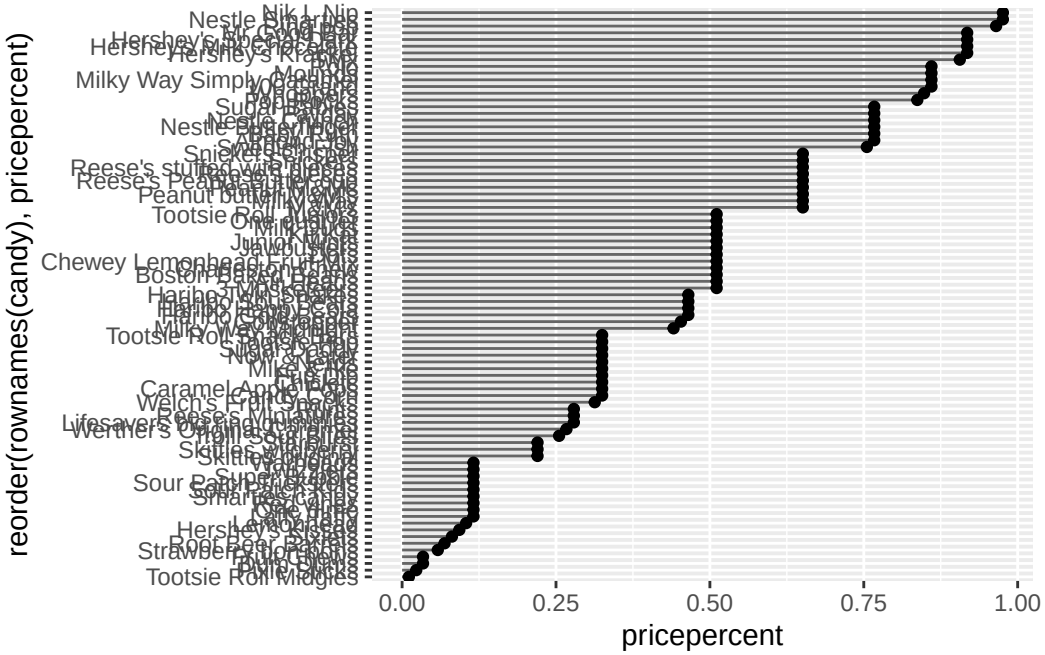
OPTIONAL:

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_col()
```




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



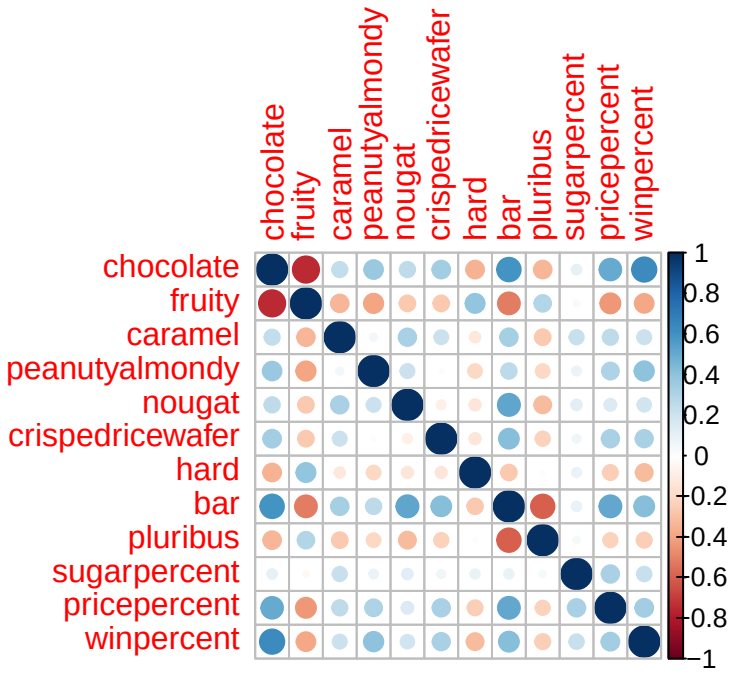
5. Exploring the correlation structure

Next, we'll examine how the variables relate using a correlation matrix visualized with the corrplot package.

```
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)?

Fruit and chocolate are the two variables showing the strongest anti-correlation.

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

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

6. Principal Component Analysis

We perform PCA on the candy dataset using the `prcomp()` function, making sure to set `scale=TRUE`.

For the main PCA result firgure:

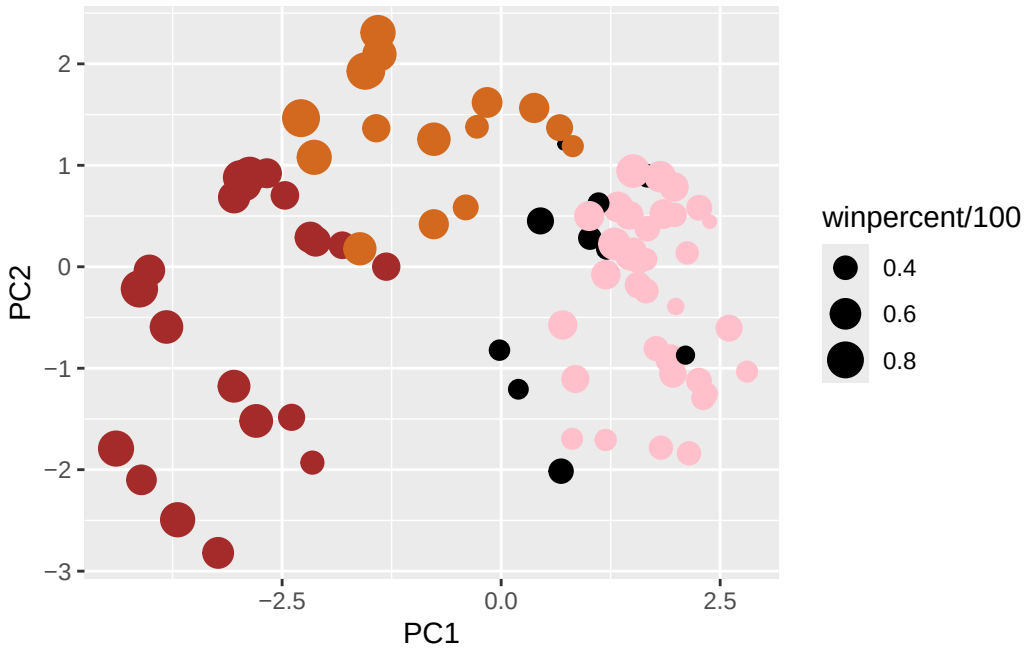
```
pca <- prcomp(candy, scale=TRUE)
```

```
str(candy)
```

```
'data.frame':   85 obs. of  12 variables:
 $ chocolate       : int  1 1 0 0 0 1 1 0 0 0 ...
 $ fruity          : int  0 0 0 0 1 0 0 0 0 1 ...
 $ caramel         : int  1 0 0 0 0 0 1 0 0 1 ...
 $ peanutyalmondy  : int  0 0 0 0 0 1 1 1 0 0 ...
 $ nougat          : int  0 1 0 0 0 0 1 0 0 0 ...
 $ crispedricewafer: int  1 0 0 0 0 0 0 0 0 0 ...
 $ hard            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ bar             : int  1 1 0 0 0 1 1 0 0 0 ...
 $ pluribus        : int  0 0 0 0 0 0 0 1 1 0 ...
 $ sugarpercent    : num  0.732 0.604 0.011 0.011 0.906 ...
 $ pricepercent    : num  0.86 0.511 0.116 0.511 0.511 ...
 $ winpercent      : num  67 67.6 32.3 46.1 52.3 ...
```

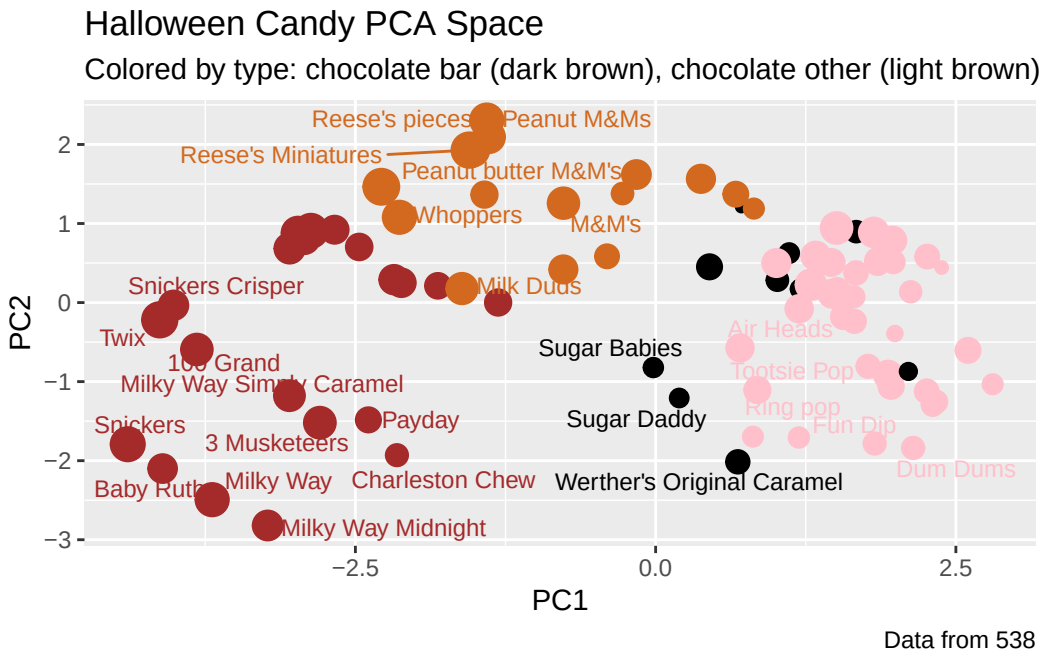
```
numeric_candy <- candy[, sapply(candy, is.numeric)]
pca <- prcomp(numeric_candy, scale=TRUE)
```

```
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
```

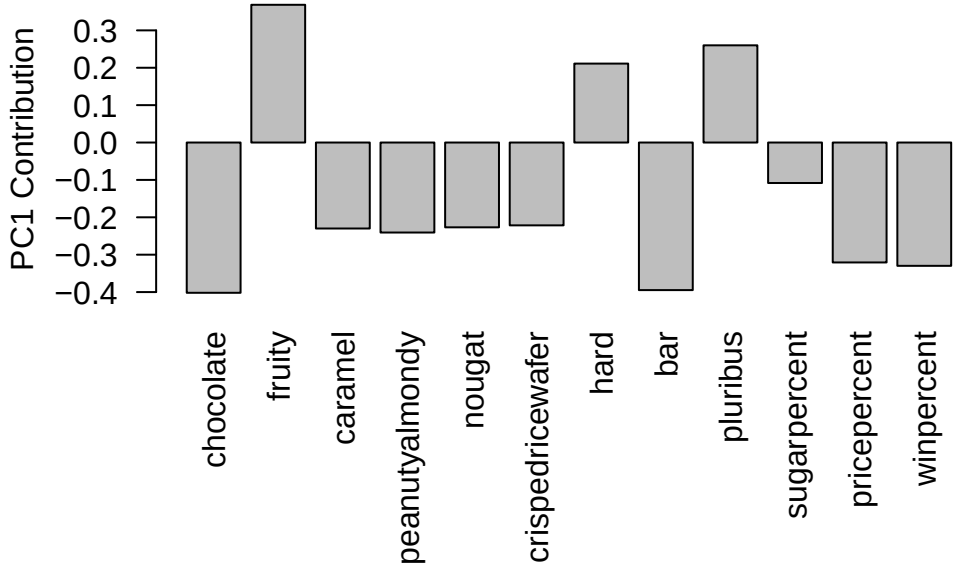


- Let's install 'plotly' package

```
library(plotly)
ggplotly(p)
```

It's also important to check the "loadings" to see how the original variables influence the principal components.

```
par(mar=c(8,4,2,2))
barplot(pca$rotation[,1], las=2, ylab="PC1 Contribution")
```



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

PC1 highlights the original variables fruity, hard, and pluribus in the positive direction. This aligns with expectations, as these variables tend to be correlated for example, Skittles demonstrate all three traits.