Class 10: Halloween Mini-Project

Patrick Tran

Importing candy data

First, let's get the data from the FiveThirtyEight GitHub repo.

```
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
```

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

```
nrow(candy)
```

```
[1] 85
```

85 candy types.

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

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

```
[1] 38
```

38 fruity candy types.

What is your favorite candy?

For a given candy, `winpercent` is the percentage of people who prefer this candy over another randomly chosen candy from the dataset. Higher values indicate a more popular candy.

Example code for Twix:

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

```
[1] 81.64291
```

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

I like Swedish Fish.

```
candy["Swedish Fish",]$winpercent
```

```
[1] 54.86111
```

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
```

The `skim()` function helps give a quick overview of a given dataset.

```
library("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?

12 of the 12 total columns are numeric type. There are no group variables.

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

```
skim(candy$chocolate)
```

Table 3: Data summary

Name	candy$chocolate
Number of rows	85
Number of columns	1
_____	
Column type frequency:	
numeric	1
_____	
Group variables	None

Variable type: numeric

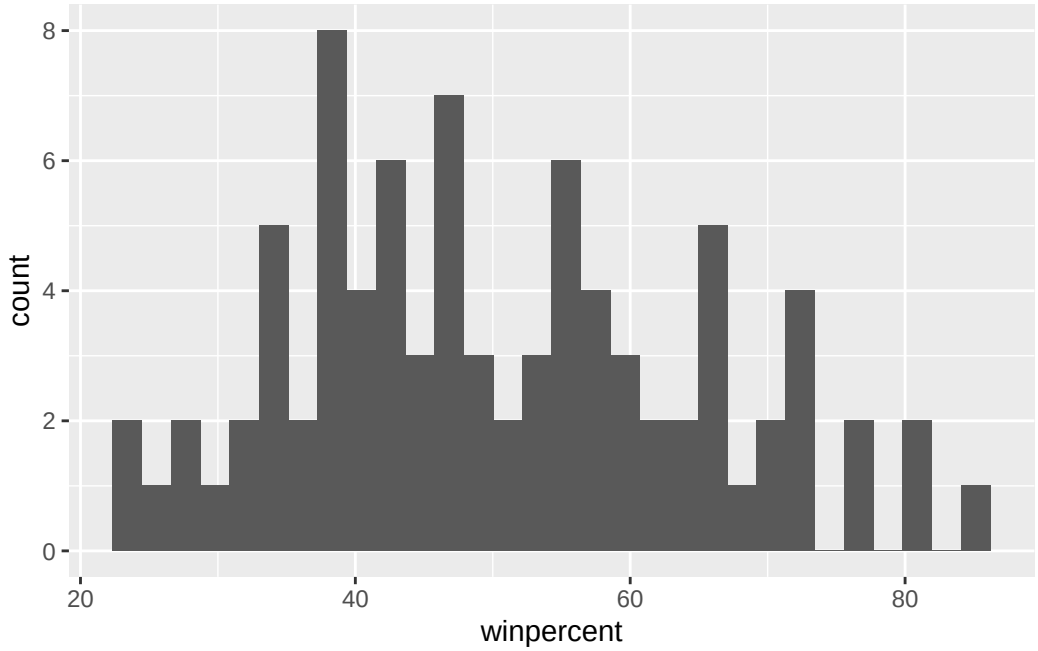
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
data	0	1	0.44	0.5	0	0	0	1	1	

The 0 means the candy is not chocolate. The 1 means the candy is chocolate.

Q8. Plot a histogram of `winpercent` values

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

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Q9. Is the distribution of `winpercent` values symmetrical?

No, the distribution is slightly right skewed.

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

The center of distribution is below 50%.

Q11. On average is chocolate candy higher or lower ranked than fruit candy?

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

```
[1] 60.92153
```

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

```
[1] 44.11974
```

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

```
    Welch Two Sample t-test

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

On average, chocolate candy is higher ranked than fruit candy.

Q12. Is this difference statistically significant?

Since the p-value of 2.871e-08 is less than 0.05, the difference is statistically significant.

#Overall candy rankings

Use base R `order()` function together with `head()` to sort the whole dataset by `winpercent`. Or use *dplyr* package using `arrange()` function with `head()` to do the same.

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

```
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 %>%
  arrange(winpercent) %>%
  head(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?

```
library(dplyr)
candy %>%
  arrange(winpercent) %>%
  tail(5)
```

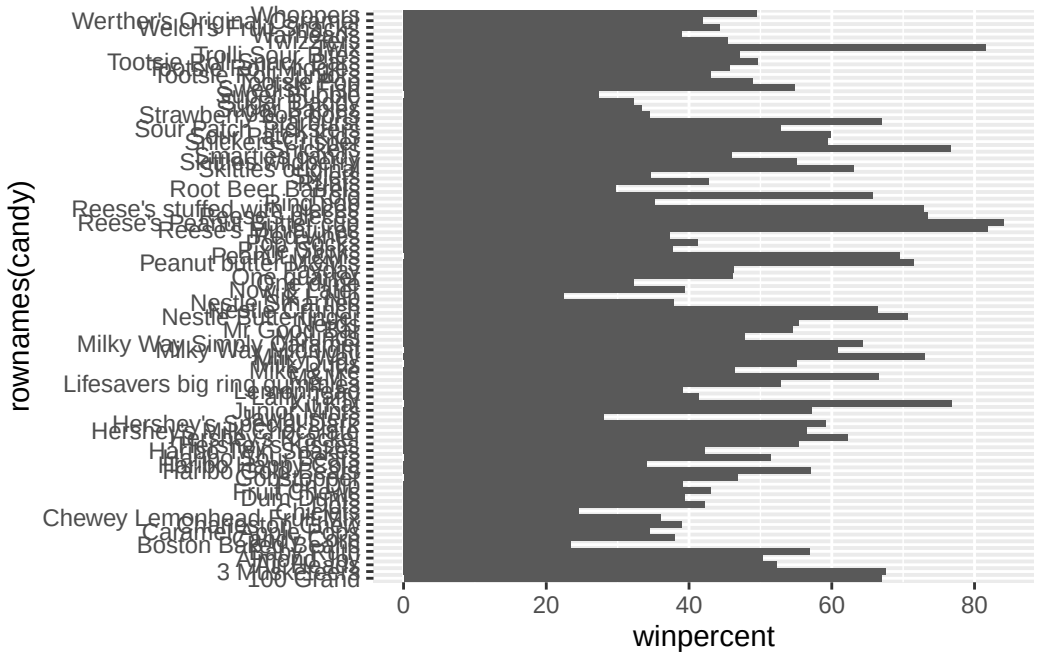
```
                          chocolate fruity caramel peanutyalmondy nougat
Snickers                          1      0       1              1      1
Kit Kat                           1      0       0              0      0
Twix                              1      0       1              0      0
Reese's 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
```

We can make a barplot to visualize the overall rankings.

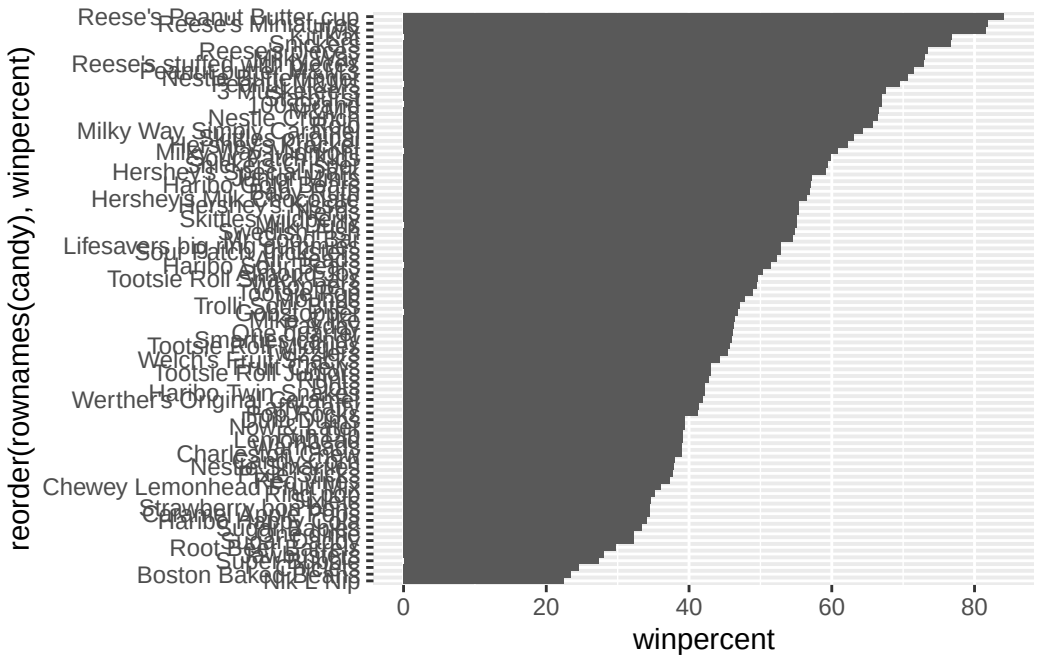
Q15. Make a first barplot of candy ranking based on `winpercent` values.

```
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`?

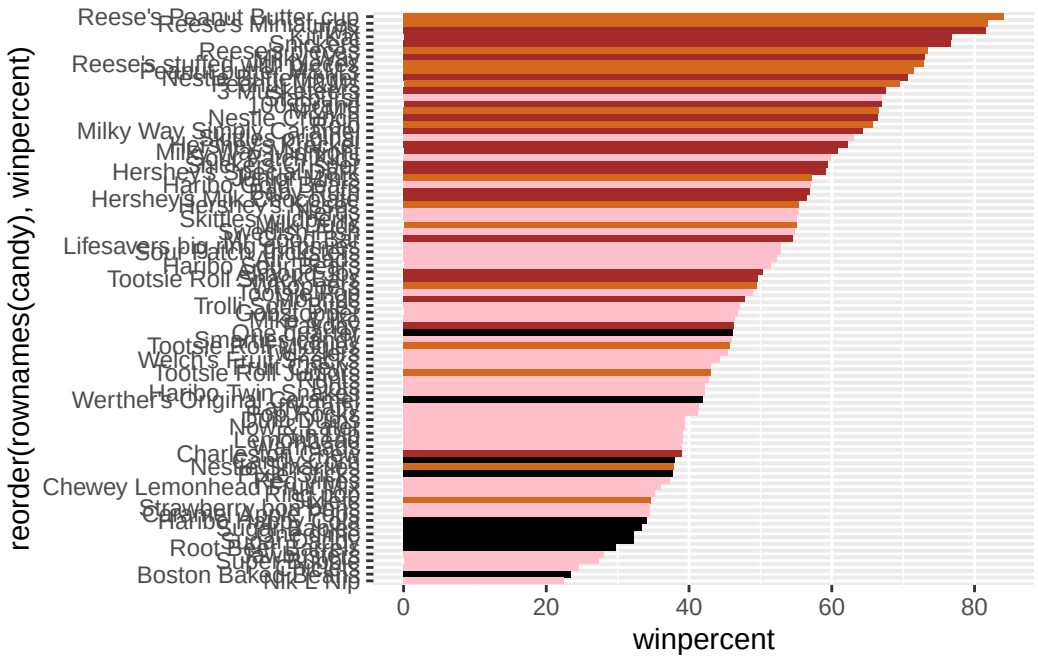
```
library(ggplot2)

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



Setup a color vector that we can use for future plots.

```
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?

Sixlets

Q18. What is the best ranked fruity candy?

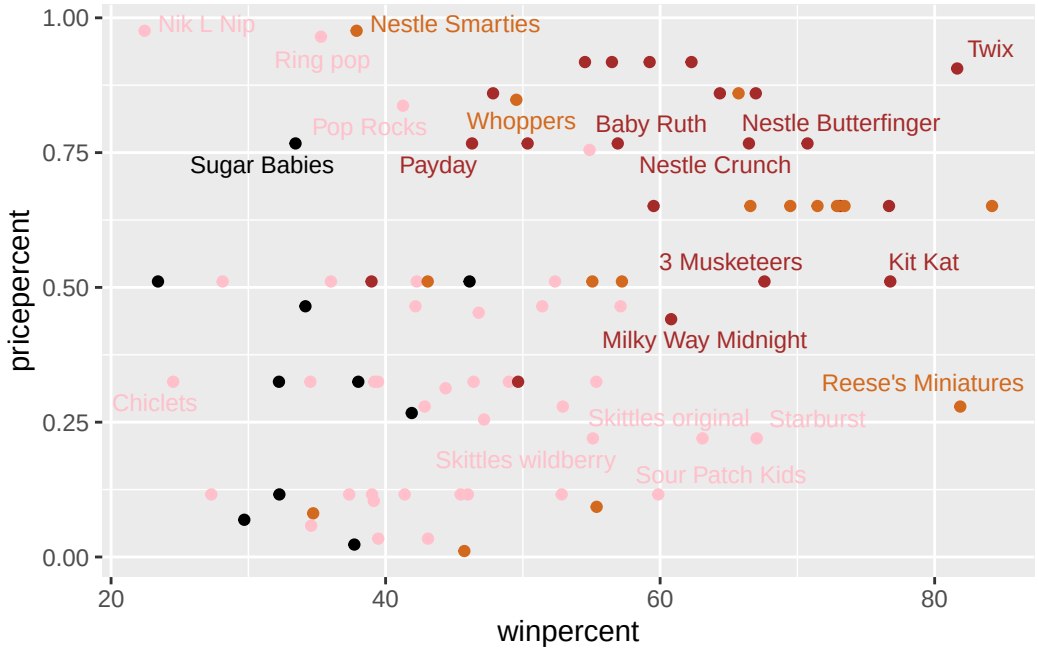
Starbursts

Taking a look at pricepercent

What about value for money or the best candy for the least money? We can make a plot of `winpercent` vs the `pricepercent` variable. `pricepercent` variable records the percentile rank of the candy's price against all of the other candies in the dataset.

```
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

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
```

Nik L Nip is the least popular out of the top 5 most expensive candy types.

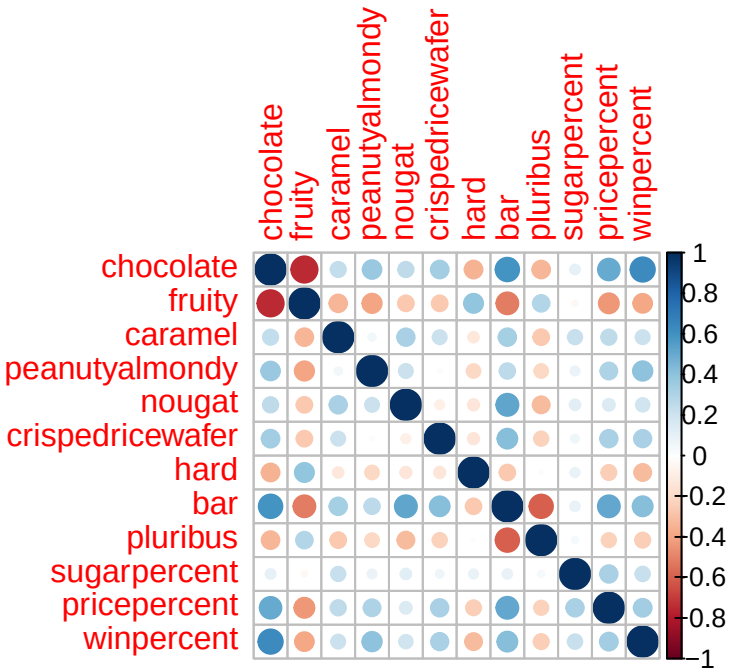
Exploring the correlation structure

We'll see how the variables interact with one another by using correlation and viewing the results with the *corrplot* package to plot a correlation matrix

```
library(corrplot)
```

```
corrplot 0.92 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 the most anti-correlated. Bar and pluribus are also anti-correlated.

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

Chocolate and bar are the most positively correlated.

Principal Component Analysis

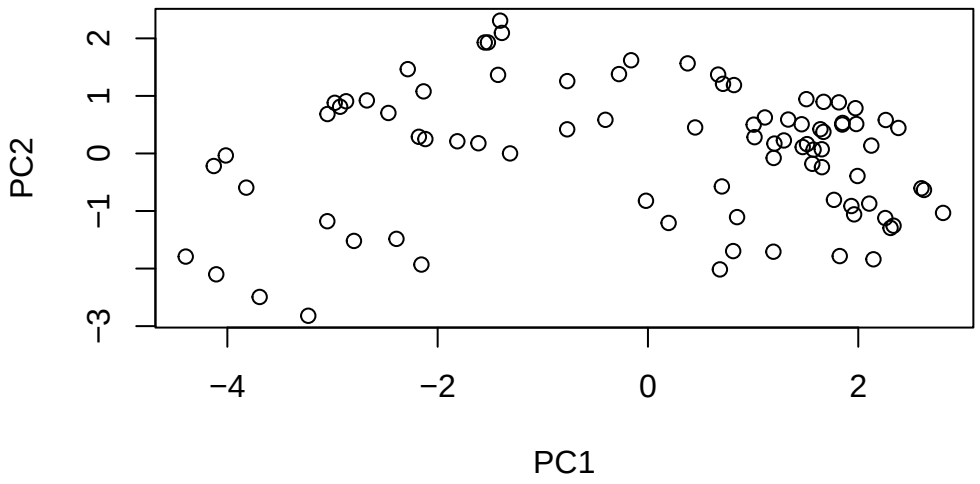
Let's apply PCA using the `prcom()` function to our candy dataset remembering to set the the `scale=TRUE` argument.

```
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
```

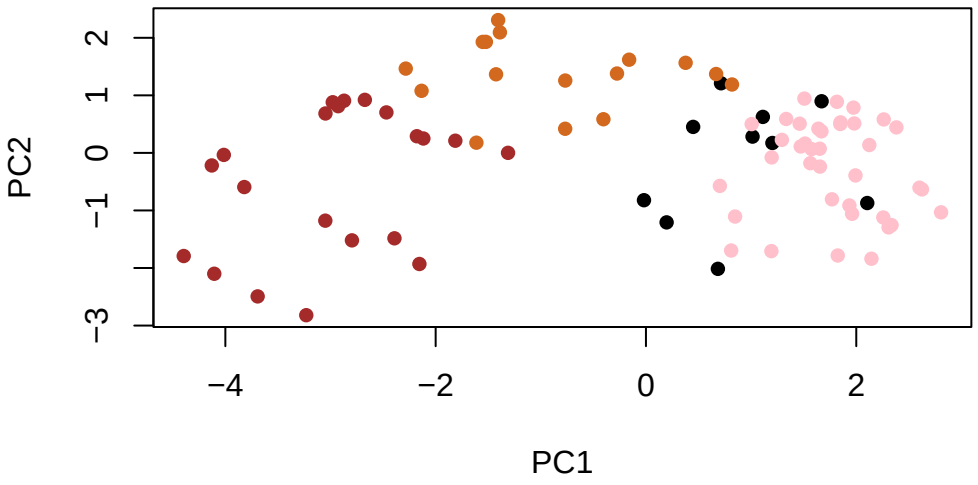
Now we can plot our main PCA score plot of PC1 vs PC2.

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



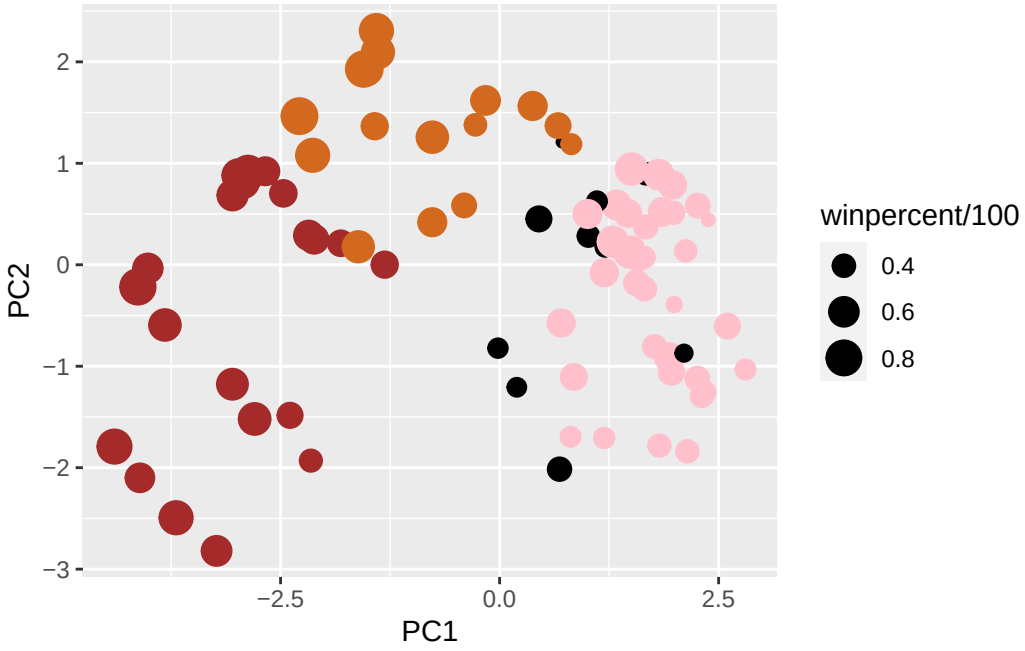
We can change the plotting character and add some color:

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



We can make a nicer plot with the `ggplot2` package.

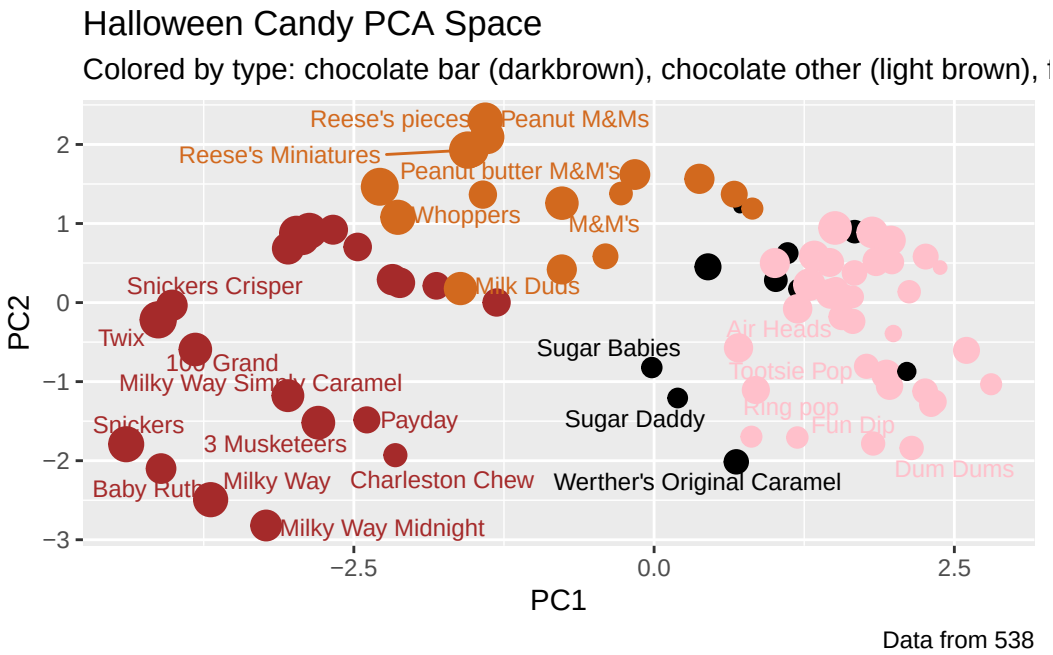
```
my_data <- cbind(candy,pca$x[,1:3])
p <- ggplot(my_data)+
  aes(x=PC1, y=PC2,
      size=winpercent/100,
      text=rownames(my_data),
      label=rownames(my_data))+
  geom_point(col=my_cols)
p
```



We can use `ggrepel` package to label up the plot with non-overlapping candy names like.

```
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 (darkbrown), chocolate other (light brown)
       caption="Data from 538")
```

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



Pass the ggplot object to `plotly` like so to generate and interactive plot.

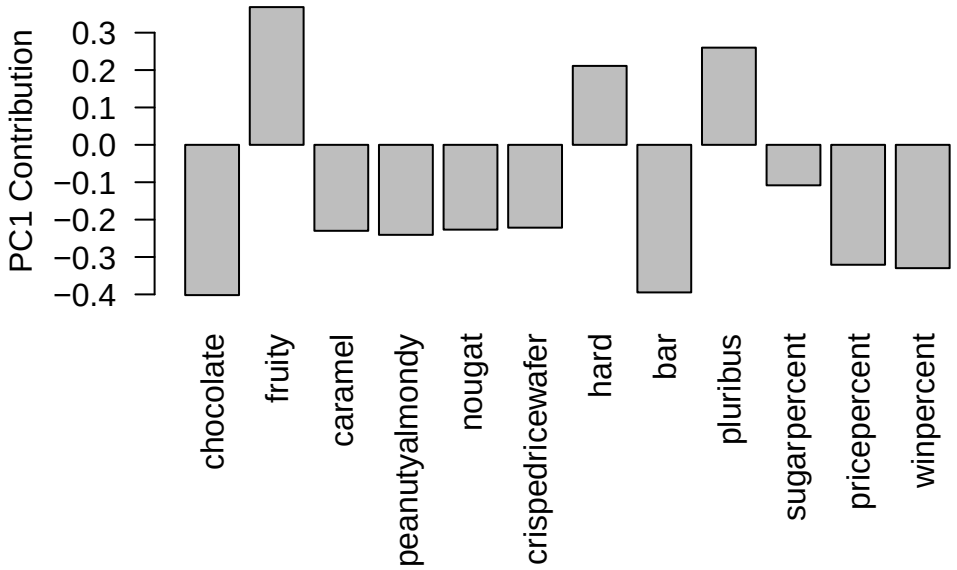
```
library(plotly)
```

```
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
```

```
ggplotly(p)
```

Let's finish by taking a quick look at PCA our loadings.

```
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?

Fruity, hard, and pluribus are picked up strongly by PC1 in the positive direction. It makes sense. Fruity candies come in a bag or box of multiple candies.