Lab 10: Halloween Mini-Project

Taylor Darby

10/29/2021

Lab 10: Halloween Mini-Project

Import the data 'candy-data.csv'

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

candy = read.csv(candy_file, row.names=1)
# fix font issues
rownames(candy) <- gsub("Ã•","'", rownames(candy))
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?

85

```
nrow(candy)
```

```
## [1] 85
```

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

38

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

```
## [1] 38
```

2. What is your favorite candy

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

Snickers' winpercent is 76.67378%

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

```
## [1] 76.67378
```

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

76.7686

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

```
## [1] 76.7686
```

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

49.6535

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

```
## [1] 49.6535
```

There is a useful skim() function in the skimr package that can help give you a quick overview of a given dataset. Let's install this package and try it on our candy data.

```
# install.packages("skimr")
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?

"winpercent" is on a different scale

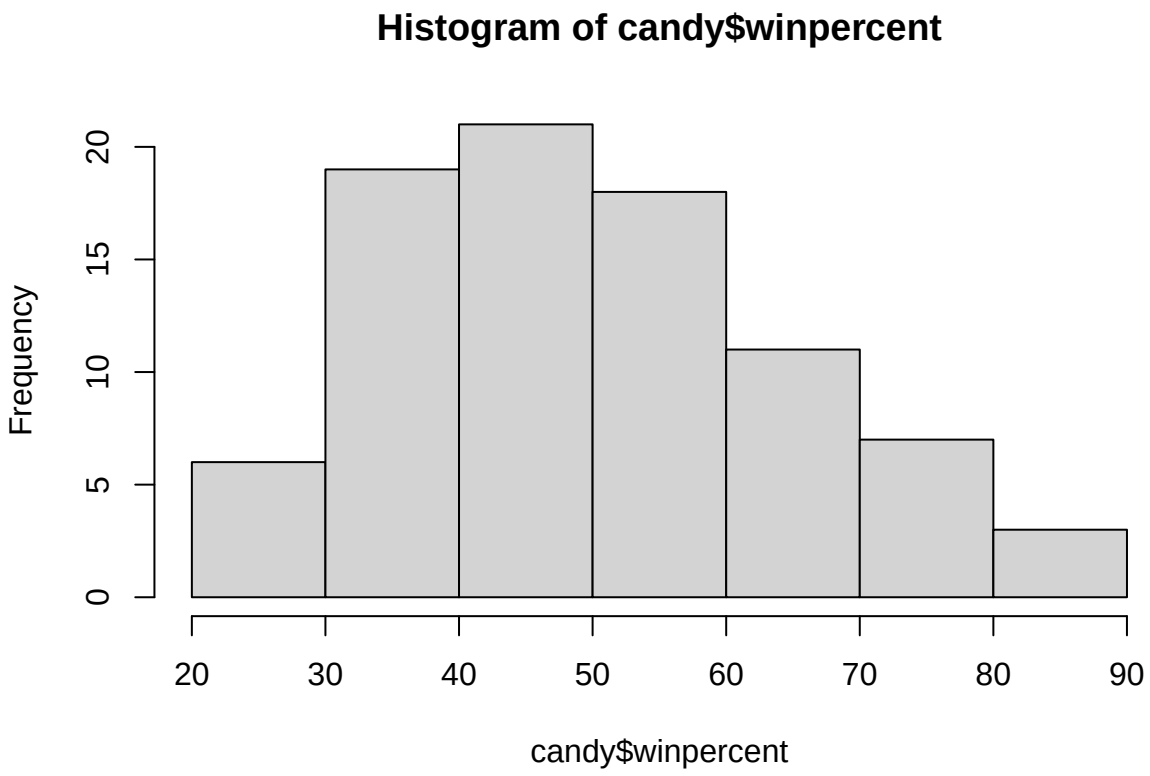
Q7. What do you think a zero and one represent for the candy$chocolate column?

True(=1) or false(=0) for whether or not chocolate was chosen

Q8. Plot a histogram of winpercent values

See table below

```
hist(candy$winpercent)
```



Q9. Is the distribution of winpercent values symmetrical?

No

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

below

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

Chocolate is ranked higher

```
# change the chocolate column to a logical (returns true or false)
#
chocolate <- candy[as.logical(candy$chocolate),]$winpercent
mean(chocolate)
```

```
## [1] 60.92153
```

```
# same but for fruity
fruity <- candy[as.logical(candy$fruity),]$winpercent
mean(fruity)
```

```
## [1] 44.11974
```

Q12. Is this difference statistically significant?

Yes, the p-value = 2.871e-08

```
t.test(chocolate,fruity)
```

```
##
##  Welch Two Sample t-test
##
## data:  chocolate and fruity
## 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

Let's make a barplot of the winpercent values for the various candy types

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

See table below

```
least.liked <- candy[order(candy$winpercent),]
least.liked[1:5,]
```

```
##                    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?

See table below

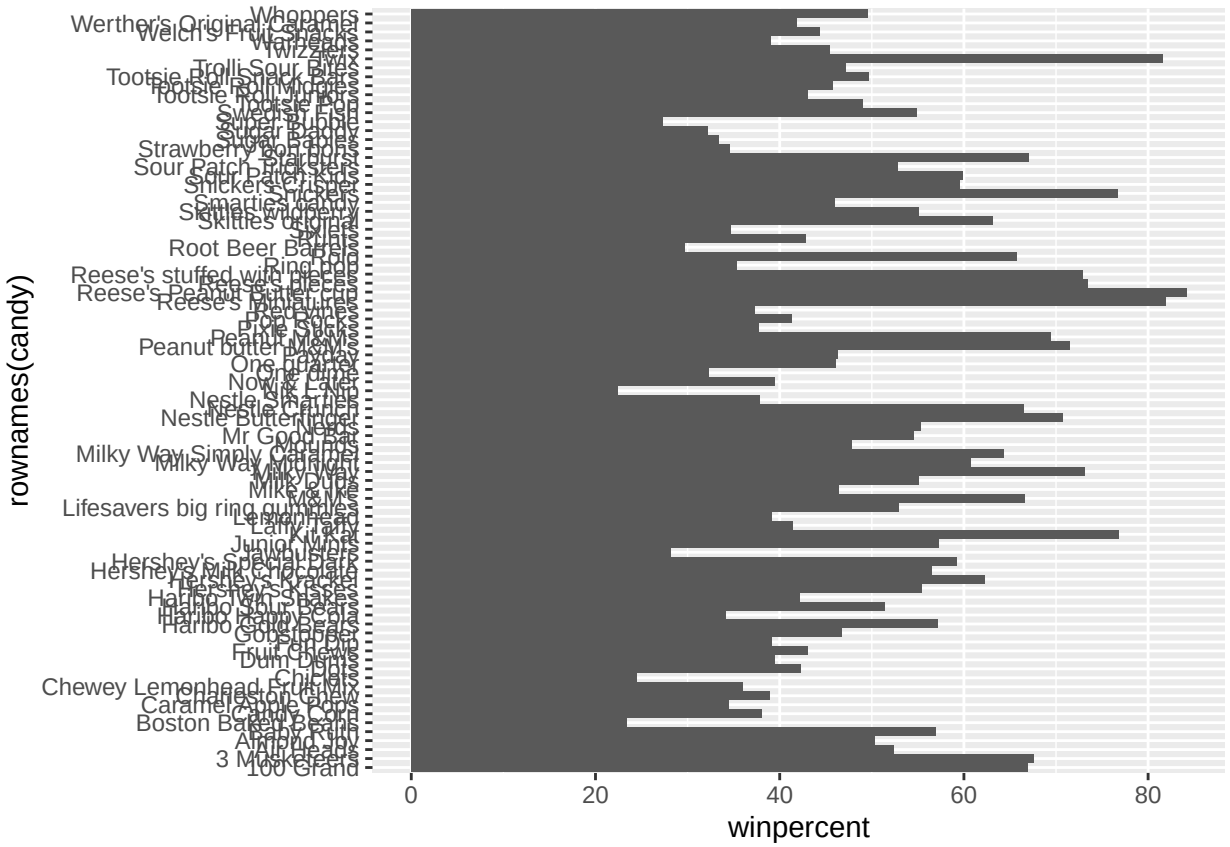
```
most.liked <- candy[order(-candy$winpercent),]
most.liked[1:5,]
```

```
##                          chocolate fruity caramel peanutyalmondy 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
##                          crispedricewafer hard bar pluribus sugarpercent
## 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
##                          pricepercent winpercent
## 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
```

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

See table below

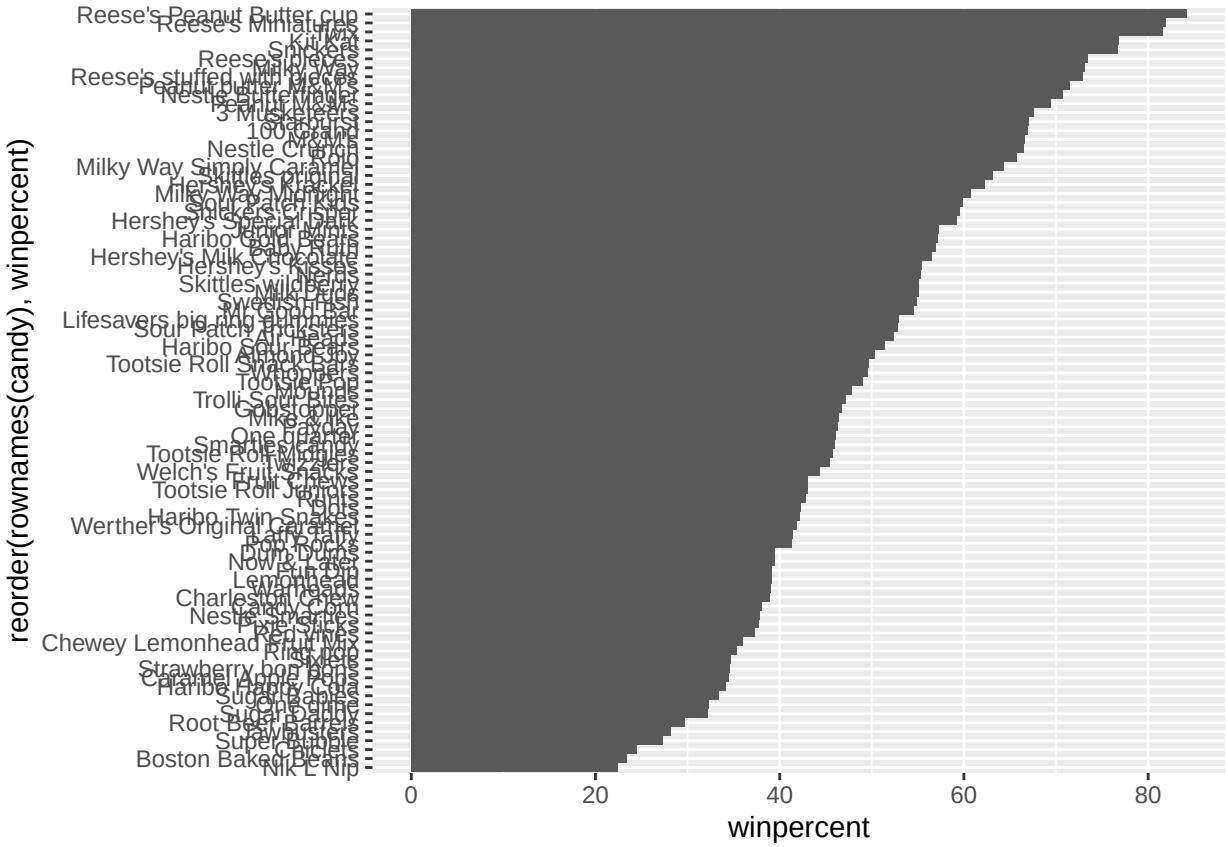
```
library(ggplot2)

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



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

See table below

```
# We need to improve this to reorder the candy by winpercent values
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy),winpercent)) +
  geom_col()
```



Time to add some color:

Create a vector of colors to use to color the plot

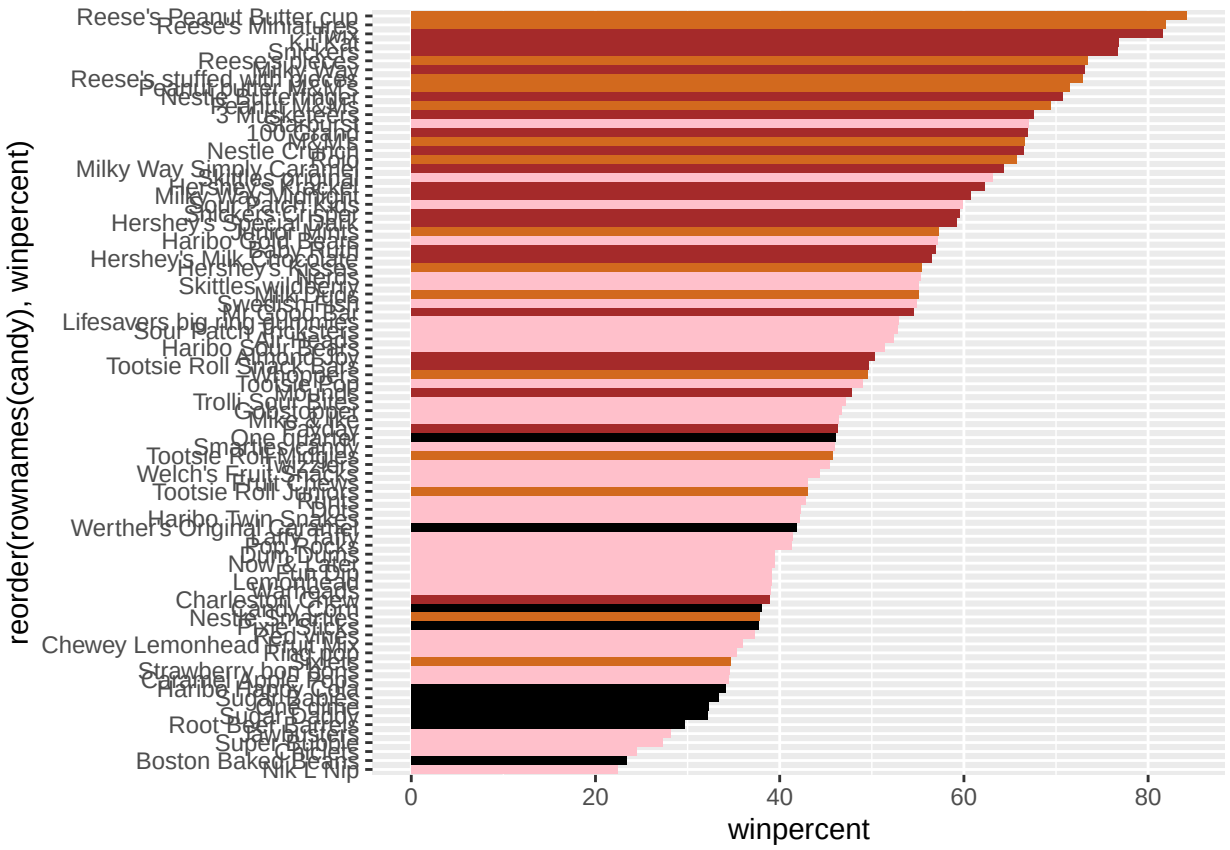
```
# Create a color vector. All black to start
my_cols=rep("black", nrow(candy))

# Now overwrite the black with colors by candy type
my_cols[as.logical(candy$chocolate)] = "chocolate"
my_cols[as.logical(candy$bar)] = "brown"
my_cols[as.logical(candy$fruity)] = "pink"
my_cols
```

```
##  [1] "brown"     "brown"     "black"     "black"     "pink"      "brown"
##  [7] "brown"     "black"     "black"     "pink"      "brown"     "pink"
## [13] "pink"      "pink"      "pink"      "pink"      "pink"      "pink"
## [19] "pink"      "black"     "pink"      "pink"      "chocolate" "brown"
## [25] "brown"     "brown"     "pink"      "chocolate" "brown"     "pink"
## [31] "pink"      "pink"      "chocolate" "chocolate" "pink"      "chocolate"
## [37] "brown"     "brown"     "brown"     "brown"     "brown"     "pink"
## [43] "brown"     "brown"     "pink"      "pink"      "brown"     "chocolate"
## [49] "black"     "pink"      "pink"      "chocolate" "chocolate" "chocolate"
## [55] "chocolate" "pink"      "chocolate" "black"     "pink"      "chocolate"
## [61] "pink"      "pink"      "chocolate" "pink"      "brown"     "brown"
## [67] "pink"      "pink"      "pink"      "pink"      "black"     "black"
## [73] "pink"      "pink"      "pink"      "chocolate" "chocolate" "brown"
```

```
## [79] "pink"      "brown"     "pink"      "pink"      "pink"      "black"
## [85] "chocolate"
```

Plot using color

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



Q17. What is the worst ranked chocolate candy?

Nik L Nip

Q18. What is the best ranked fruity candy?

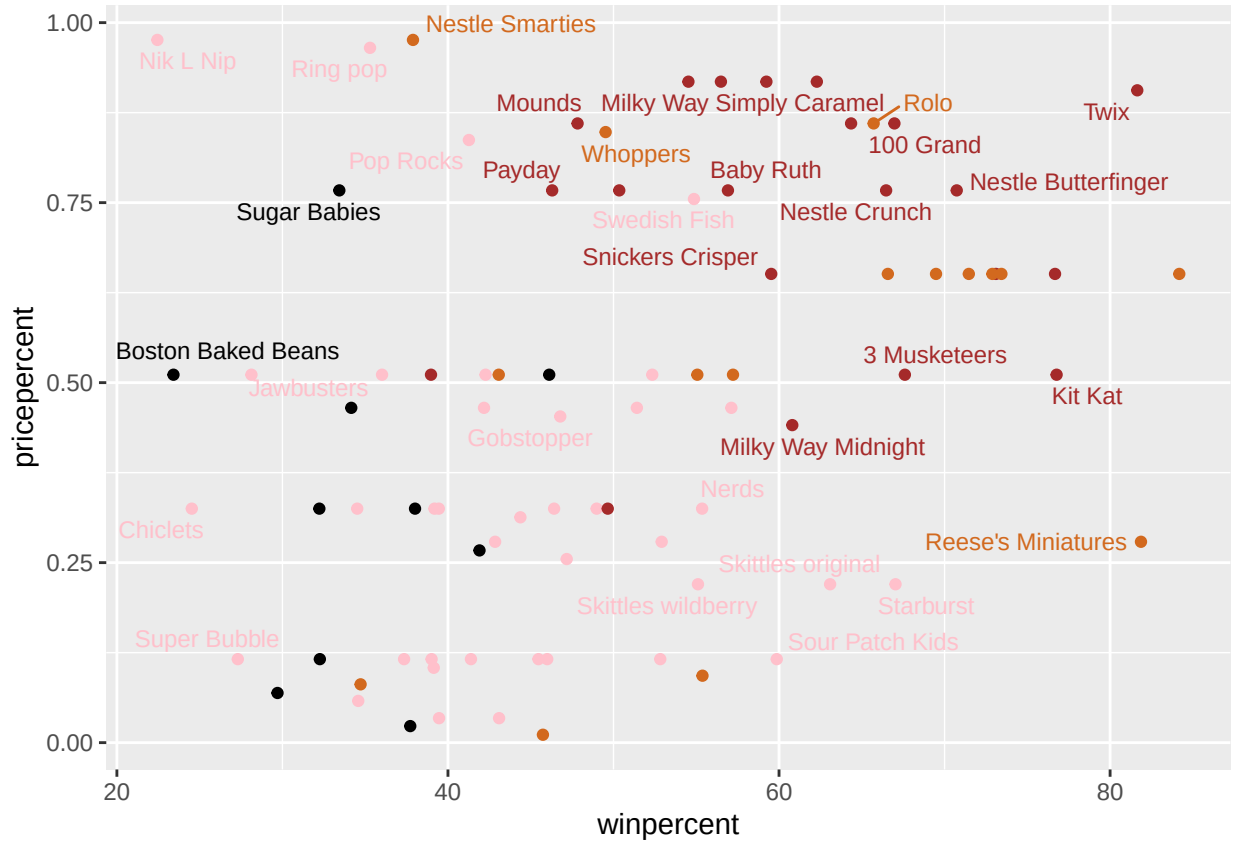
Reese's Peanut Butter Cups

4. Taking a look at pricepercent

```
#install.packages(ggrepel)
## 'ggrepel' helps read labels more easily by repelling them away from each other
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: 54 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?

Out of the top 5 candies Reese's Miniatures are the least expensive.

Q20. What are the top 5 most expensive candy types in the dataset and of these which is the least popular?

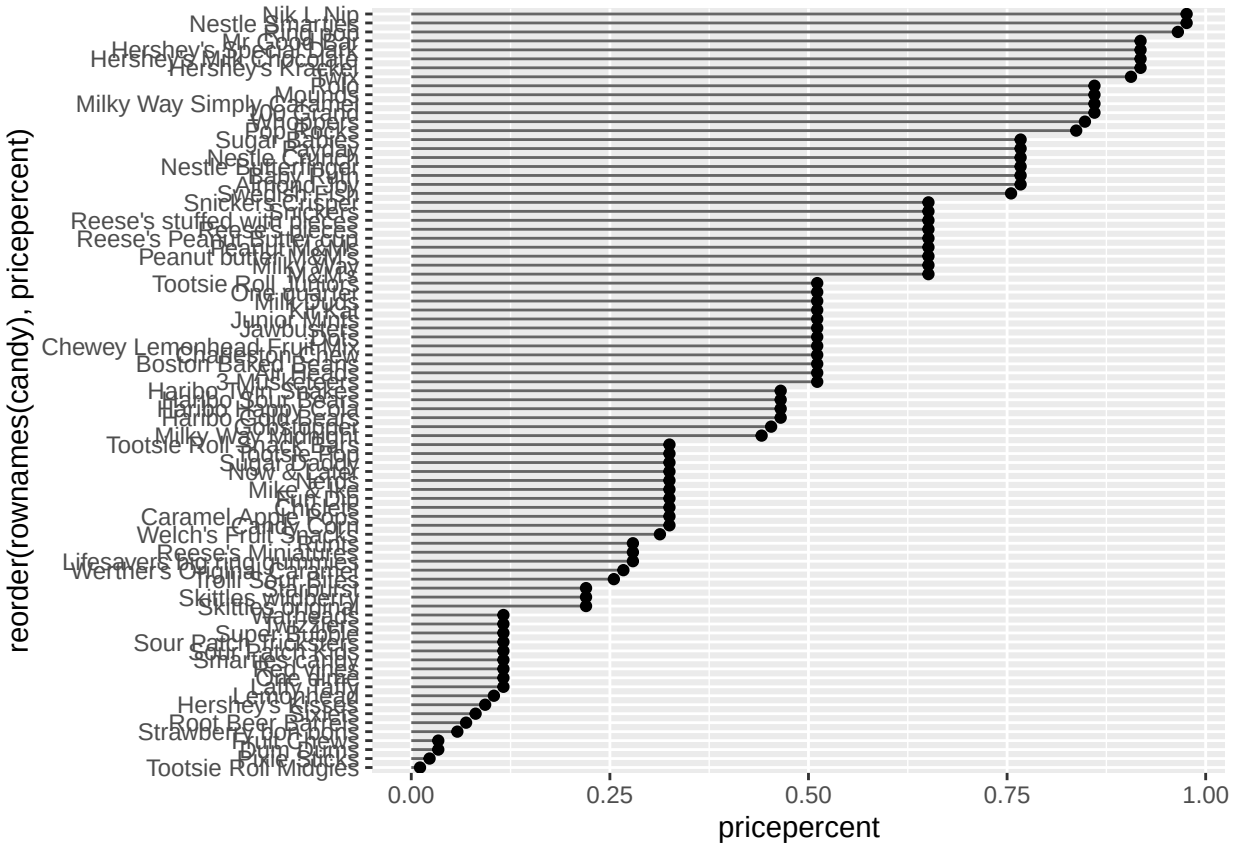
See table below for the top 5 most expensive candy types in ascending order of popularity. Nik L Nip is the least popular of the top 5 most expensive candy types.

```
most.expensive <- candy[order(-candy$pricepercent),]
top5.most.expensive <- most.expensive[1:5,]
top5.most.expensive[order(top5.most.expensive$winpercent),]
```

```
##                          chocolate fruity caramel peanutyalmondy nougat
## Nik L Nip                        0      1       0              0      0
## Ring pop                         0      1       0              0      0
## Nestle Smarties                  1      0       0              0      0
## Hershey's Milk Chocolate         1      0       0              0      0
## Hershey's Krackel                1      0       0              0      0
##                          crispedricewafer hard bar pluribus sugarpercent
## Nik L Nip                               0    0   0        1        0.197
## Ring pop                                0    1   0        0        0.732
## Nestle Smarties                         0    0   0        1        0.267
## Hershey's Milk Chocolate                0    0   1        0        0.430
## Hershey's Krackel                       1    0   1        0        0.430
##                          pricepercent winpercent
## Nik L Nip                       0.976   22.44534
## Ring pop                        0.965   35.29076
## Nestle Smarties                 0.976   37.88719
## Hershey's Milk Chocolate        0.918   56.49050
## Hershey's Krackel               0.918   62.28448
```

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().

```
# Make a lollipop chart of pricepercent
ggplot(candy) +
  aes(pricepercent, reorder(rownames(candy), pricepercent)) +
  geom_segment(aes(yend = reorder(rownames(candy), pricepercent),
                   xend = 0), col="gray40") +
    geom_point()
```

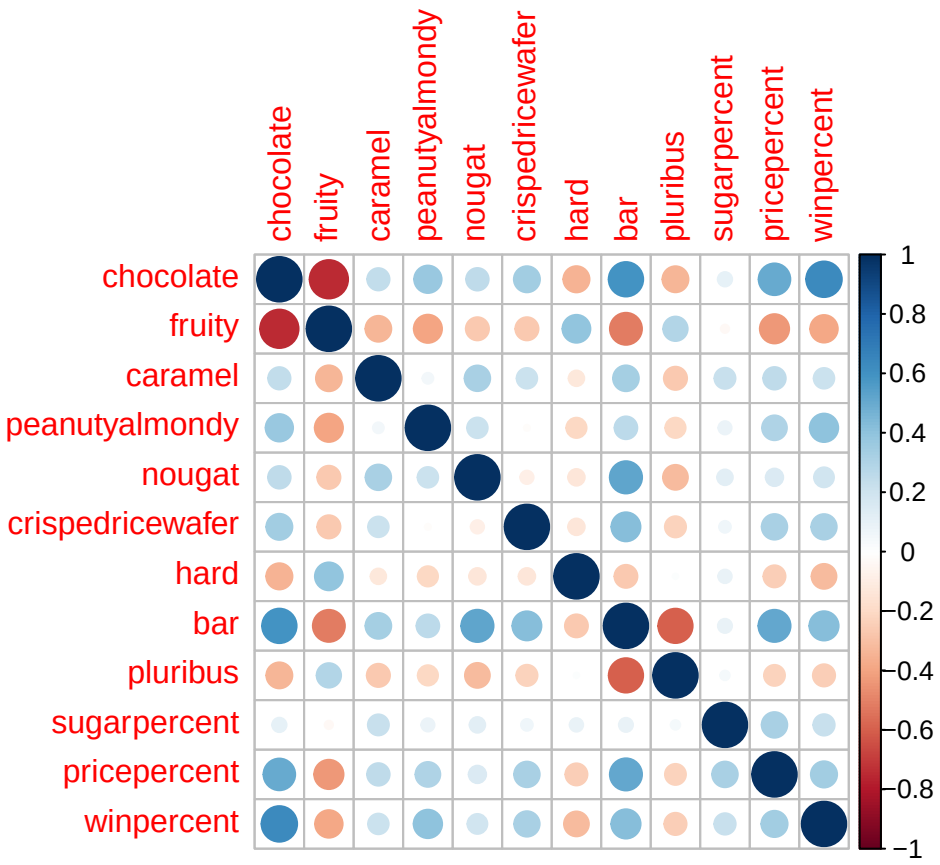


5. Exploring the correlation structure

```
# Correlation analysis
library(corrplot)
```

```
## corrplot 0.90 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 least correlated (anti-correlated)

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

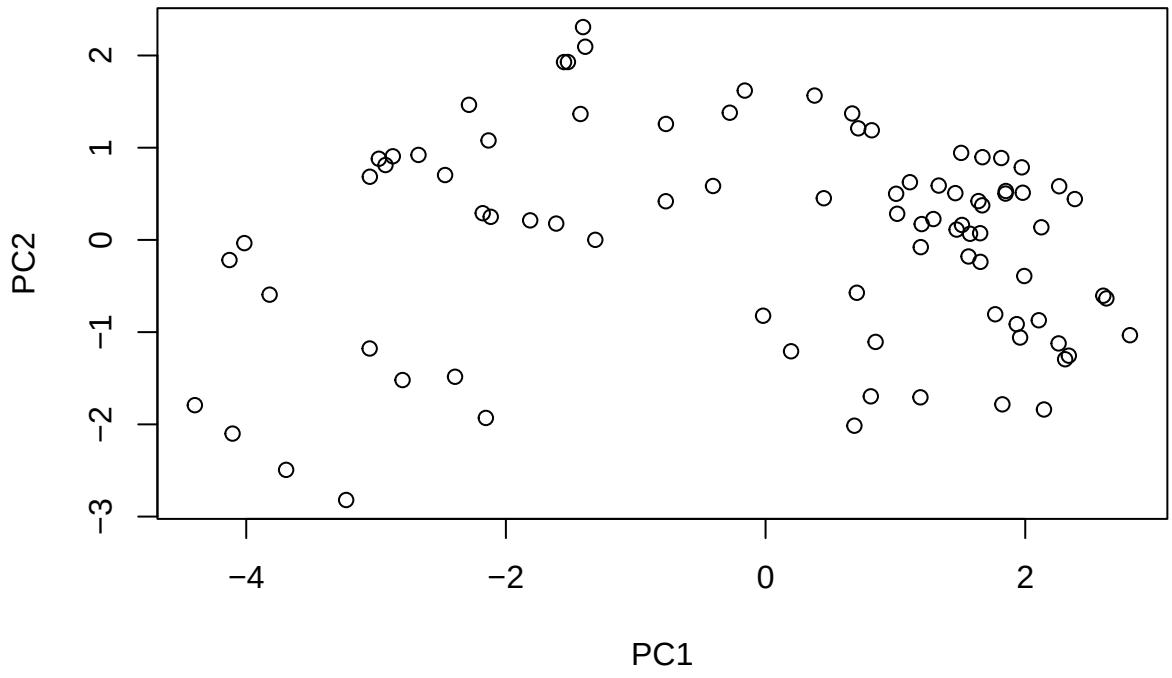
It's difficult to tell if "chololate" is more positively correlated with "bar" or "winpercent"

6. Principle Componenent Analysis

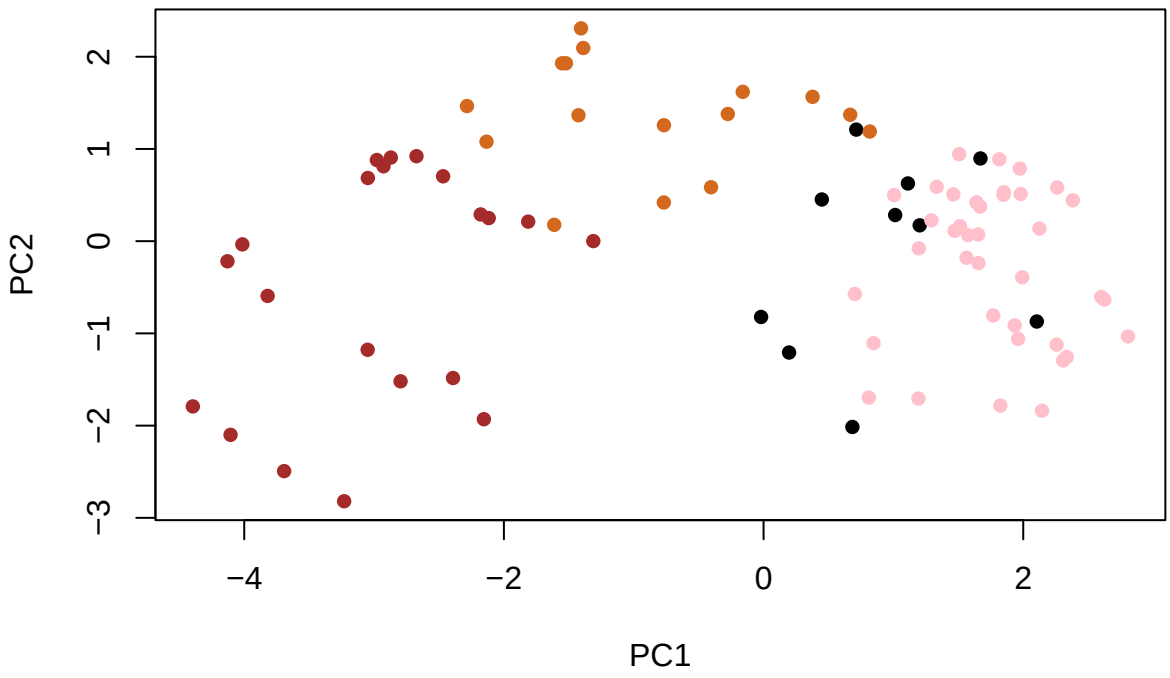
```
# PCA Analysis
pca <- prcomp(candy, scale = TRUE)
summary(pca)
```

```
## 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
```

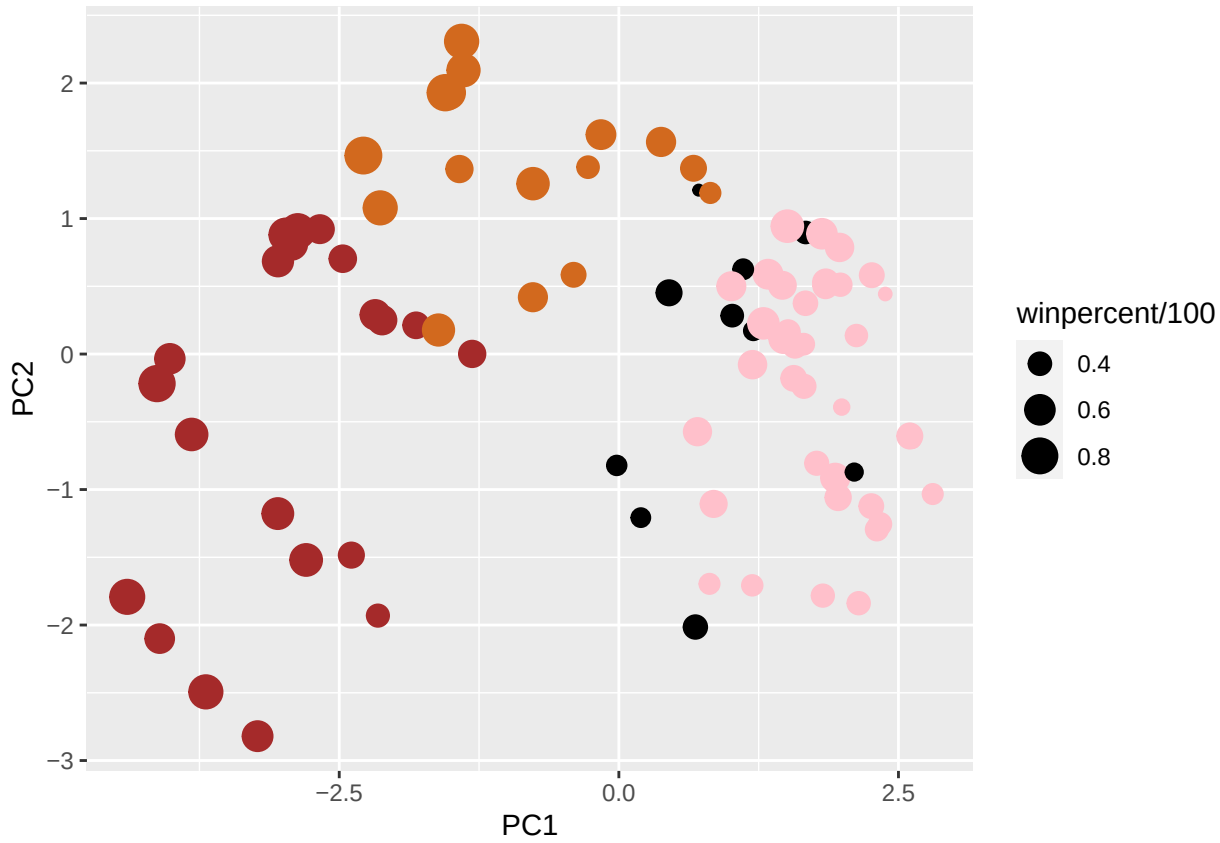
```
# Now we can plot our main PCA score plot of PC1 vs PC2.
plot(pca$x[,1:2])
```




```
# We can change the plotting character and add some color:

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

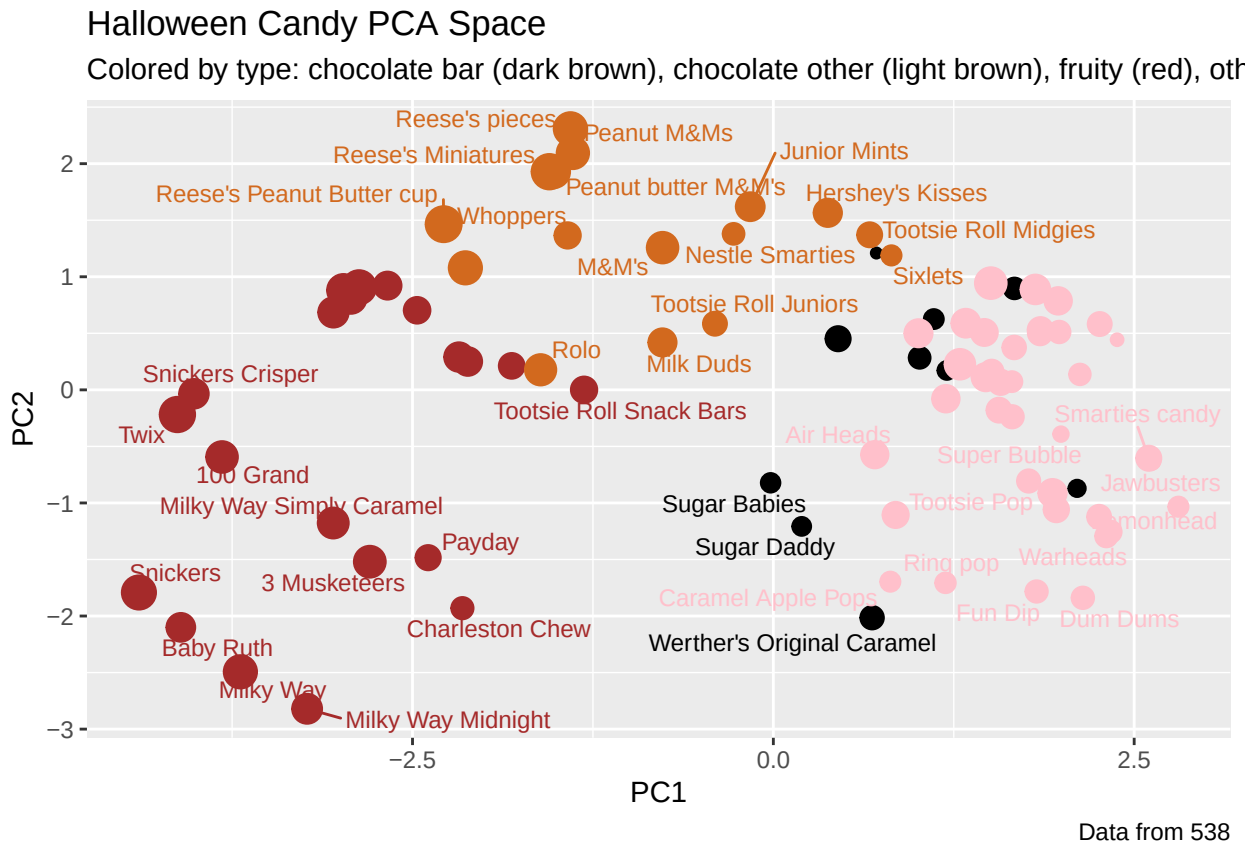



```
# Make a new data-frame with our PCA results and candy data
my_data <- cbind(candy, pca$x[,1:3])

# Same plot as before but colored by candy type and sized by popularity
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
```




```
# Now we'll label the plot and the data points
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), fruity (re
       caption="Data from 538")
```

```
## Warning: ggrepel: 44 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



If you want to see more candy labels you can change the max.overlaps value to allow more overlapping labels or pass the ggplot object p to plotly like so to generate an interactive plot that you can mouse over to see labels:

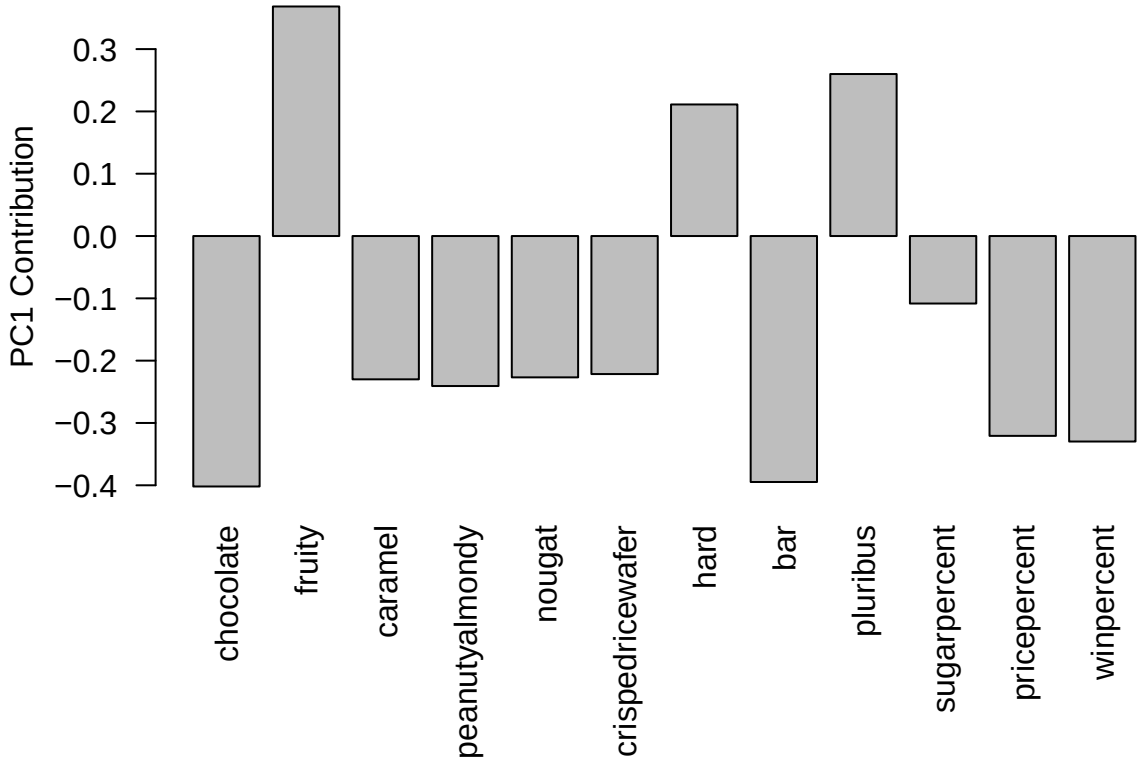
```
library(plotly)
```

```
##
## 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
```

```
# Note: hid this plot to more easily knit my pdf for submission
# ggplotly(p)
```

Let's finish by taking a quick look at PCA our loadings. Do these make sense to you? Notice the opposite effects of chocolate and fruity and the similar effects of chocolate and bar (i.e. we already know they are correlated).

```
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?

all of the fruity candies were plotted in the positive direction on the PC1 axis of the PCA plot and the chocolate candies were plotted in the negative direction. They were also the farthest from each other on the PC1 axis of the plot. This is consitent with how distant the bars are in this barplot of the loadings.