class10

Xinlong Wan

Mini-project

```
candy_file <- "https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-power-r

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.

```
print(paste(nrow(candy), 'candy types'))
```

```
[1] "85 candy types"
```

Q2.

```
print(paste(nrow(candy[candy$fruity==1,]), 'fruity'))
```

```
[1] "38 fruity"
```

Q3.

```
print(paste('One dime, ', round(candy["One dime", ]$winpercent,2), '%'))
```

```
[1] "One dime,  32.26 %"
```

Q4.

```
print(paste('Kit Kat, ', round(candy["Kit Kat", ]$winpercent,2), '%'))
```

```
[1] "Kit Kat,  76.77 %"
```

Q5.

```
print(paste('Tootsie Roll Snack Bars, ', round(candy["Tootsie Roll Snack Bars", ]$winperce
```

```
[1] "Tootsie Roll Snack Bars,  49.65 %"
```

```
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	

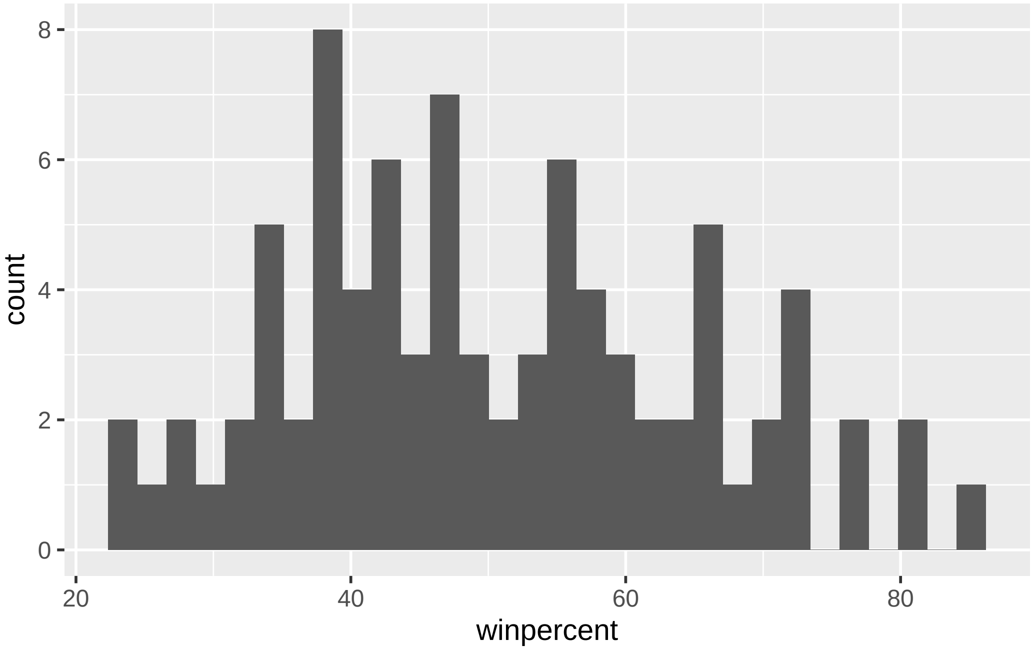
Q6. The sugarpercent, pricepercent, and winpercent columns contains continuous values that are not only 1 and 0. This is because they represent a percantage instead a Yes/No.

Q7. Zero: A specific candy type doesn't contain chocolate. One: A specific candy type contains chocolate.

Q8.

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

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



Q9. It's not entirely symmetrical. The distribution skew to the right.

Q10.

```
mean(candy$winpercent) > median(candy$winpercent)
```

```
[1] TRUE
```

The center of the distribution is above 50% as it's skewed to the right.

Q11.

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

```
[1] TRUE
```

On average chocolate candy is higher ranked than fruit candy in terms of winpercentage/

Q12.

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

p_value < 0.001. The difference is statistically significant at alpha=0.001 level.

Q13.

```r
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
```

```r
candy %>%
  arrange(winpercent) %>%
  head(5) %>%
  rownames()
```

```
[1] "Nik L Nip"          "Boston Baked Beans" "Chiclets"
[4] "Super Bubble"       "Jawbusters"
```

Q14.

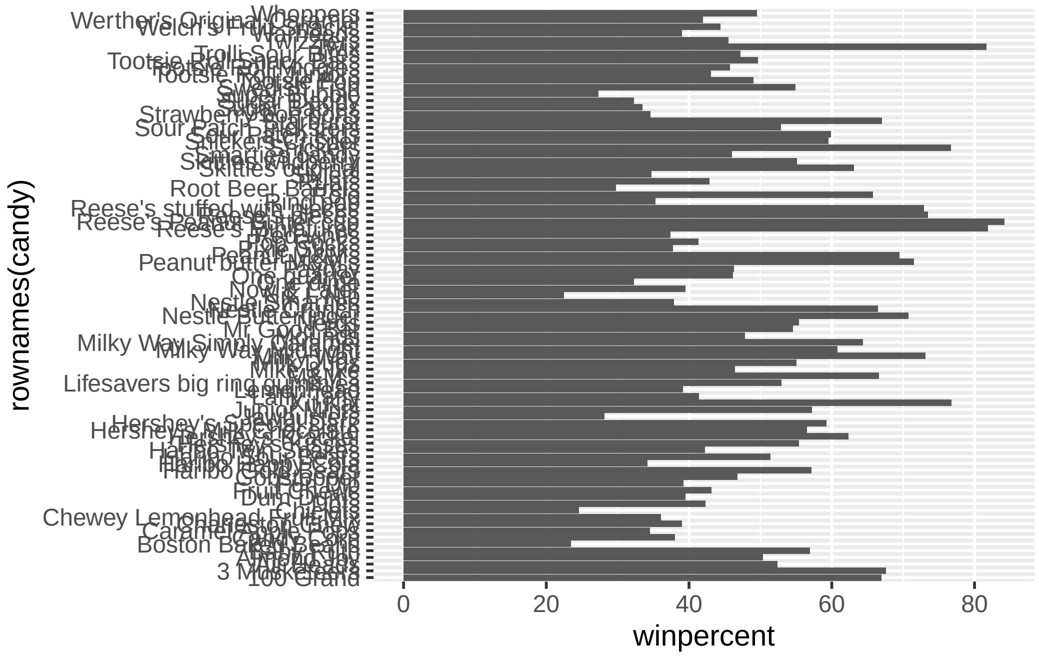
```r
candy %>%
  arrange(desc(winpercent)) %>%
  head(5) %>%
  rownames()
```

```
[1] "Reese's Peanut Butter cup" "Reese's Miniatures"
[3] "Twix"                      "Kit Kat"
[5] "Snickers"
```

I prefer using arrange, as it allows us to build step by step pipeline that are easier to remember and read.

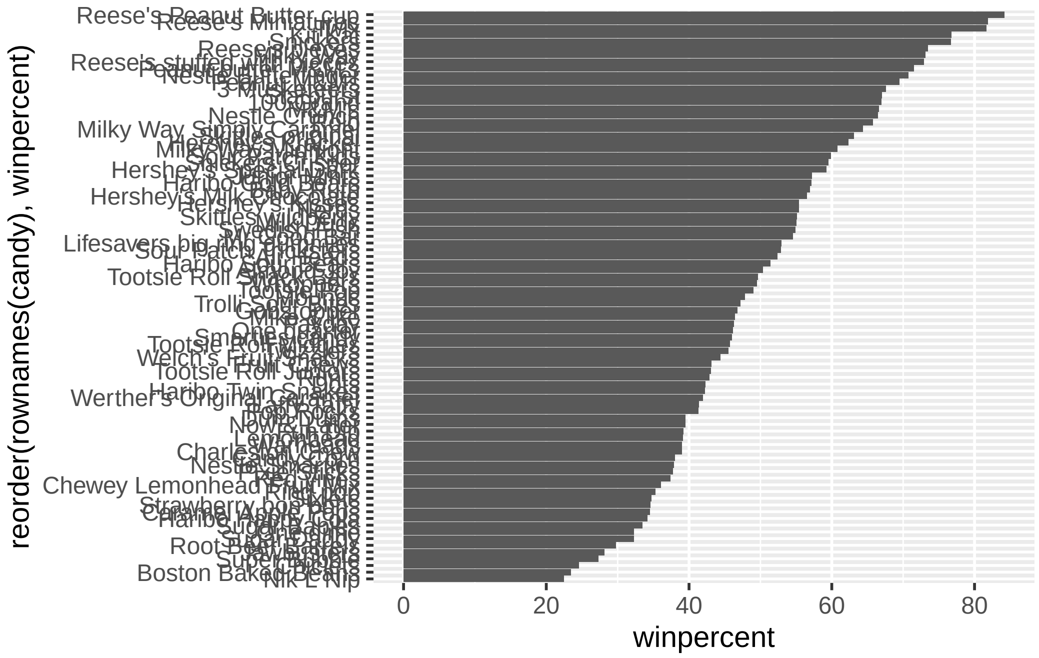
Q15.

```r
library(ggplot2)

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

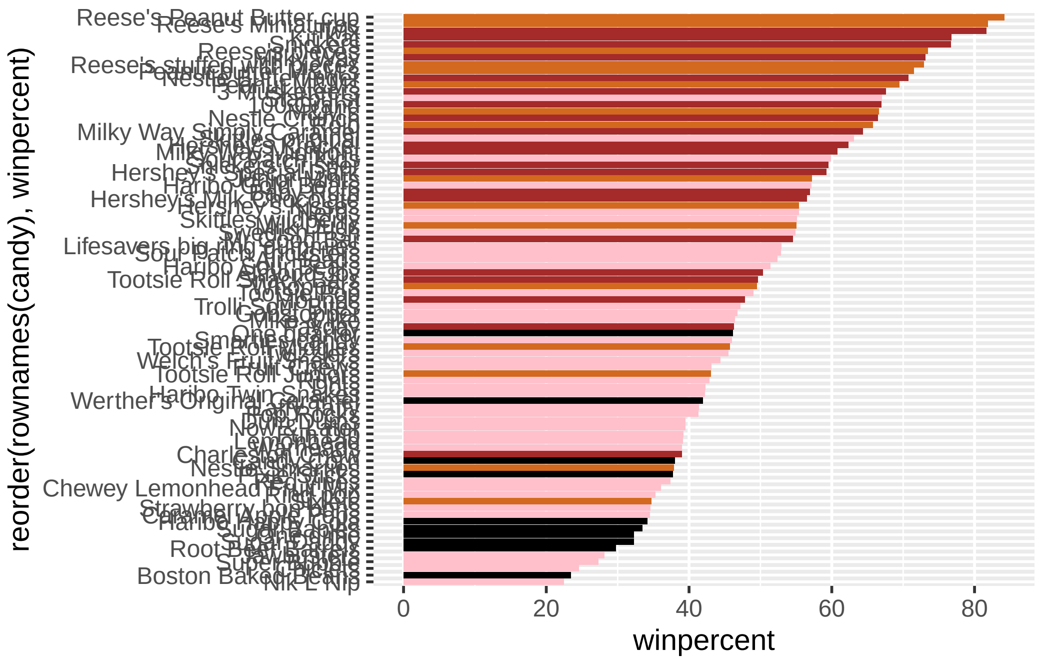


Q16

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



```
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)
```

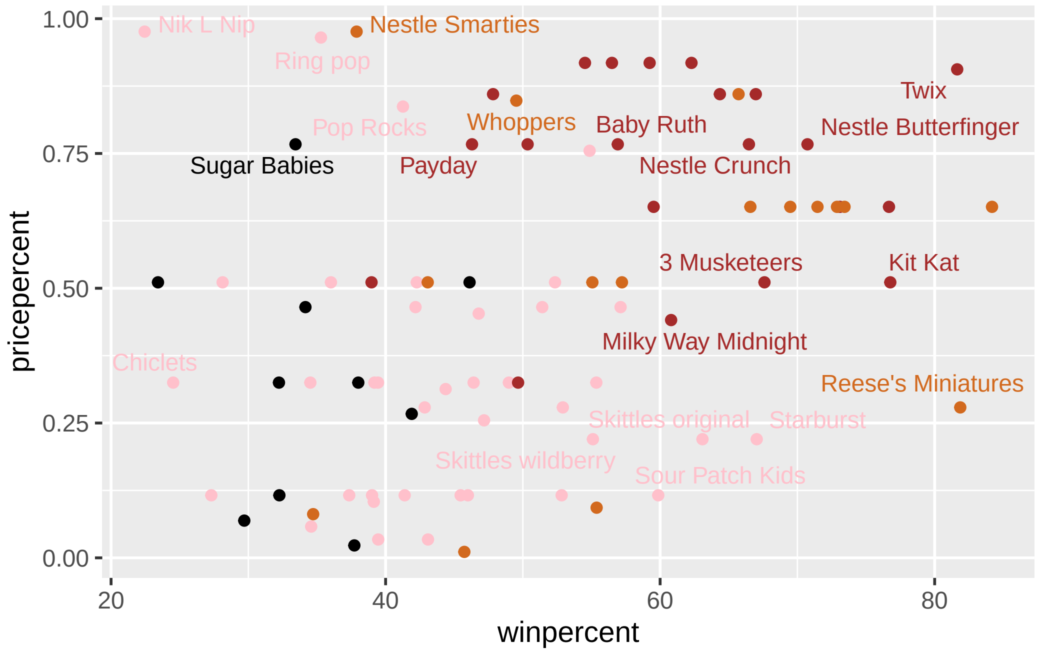


Q17. Sixlets

Q18. Starburst

```
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.

```
candy %>%
  arrange(desc(pricepercent)) %>%
  head(5)
```

```
                         chocolate fruity caramel peanutyalmondy nougat
Nik L Nip                        0      1       0              0      0
Nestle Smarties                  1      0       0              0      0
Ring pop                         0      1       0              0      0
Hershey's Krackel                1      0       0              0      0
Hershey's Milk Chocolate         1      0       0              0      0
                         crispedricewafer hard bar pluribus sugarpercent
Nik L Nip                               0    0   0        1        0.197
Nestle Smarties                         0    0   0        1        0.267
Ring pop                                0    1   0        0        0.732
Hershey's Krackel                       1    0   1        0        0.430
Hershey's Milk Chocolate                0    0   1        0        0.430
                         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
```

Reese's Miniatures cost the least among the top five popular candies.

Q20.

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

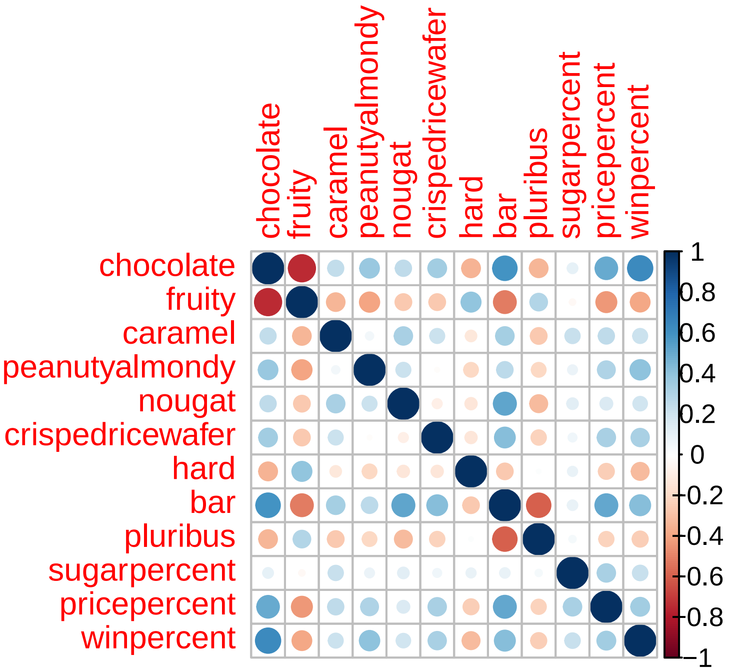
```
                         pricepercent winpercent
Nik L Nip                       0.976   22.44534
Nestle Smarties                 0.976   37.88719
Ring pop                        0.965   35.29076
Hershey's Krackel               0.918   62.28448
Hershey's Milk Chocolate        0.918   56.49050
```

Nik L Nip is the least popular.

```
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
cij <- cor(candy)
corrplot(cij)
```



Q22. fruity and chocolate are anti-correlated. Bar and pluribus are very highly nagatively negatively correlated. Many others are slighly negatively correlated.

Q23. winpercentage and chocolate. Bar and chocolate.

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

```
Importance of components:
                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     14.7231 0.70241 0.47762 0.37292 0.34641 0.33614 0.30748
Proportion of Variance  0.9935 0.00226 0.00105 0.00064 0.00055 0.00052 0.00043
Cumulative Proportion   0.9935 0.99574 0.99678 0.99742 0.99797 0.99849 0.99892
                           PC8     PC9    PC10    PC11    PC12
Standard deviation     0.27417 0.23826 0.21435 0.18434 0.15331
Proportion of Variance 0.00034 0.00026 0.00021 0.00016 0.00011
Cumulative Proportion  0.99927 0.99953 0.99974 0.99989 1.00000
```

```
pca$rotation[,1]
```

```
       chocolate            fruity           caramel    peanutyalmondy
```

```
    -0.021594409      0.012979597     -0.005422631     -0.010306565 
          nougat crispedricewafer             hard              bar 
    -0.003755535     -0.006106095      0.008097109     -0.012699628 
        pluribus     sugarpercent     pricepercent       winpercent 
     0.008474153     -0.004401607     -0.006717124     -0.999407346 
```

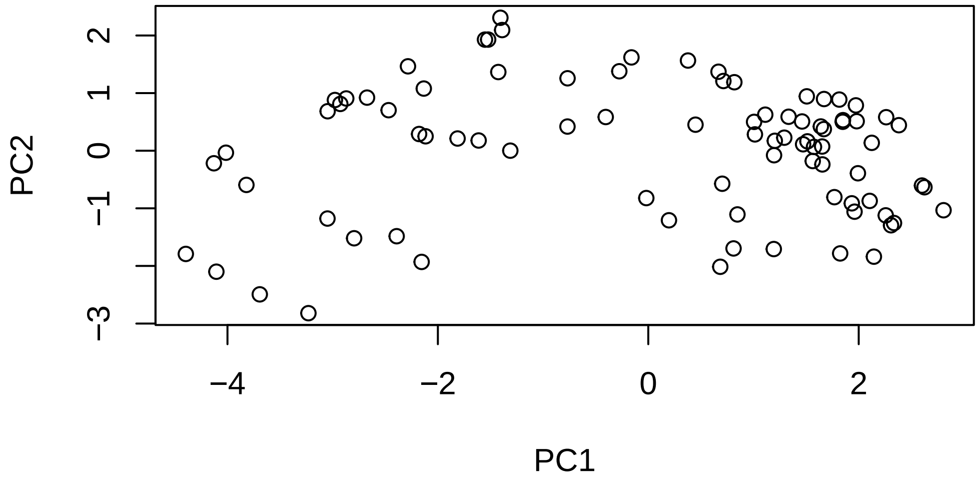
PC1 is dominated by winpercentage, since it's on a very different scale.

We should scale it.

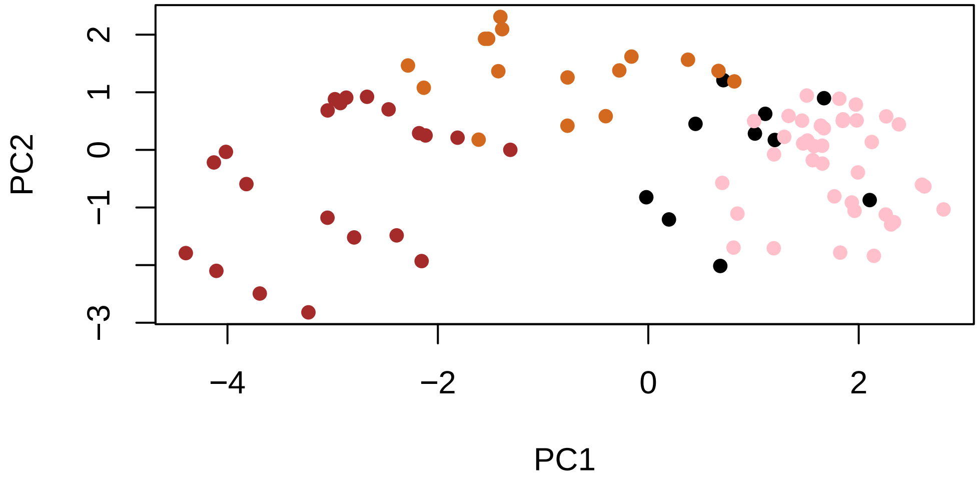
```
pca <- prcomp(candy, scale=TRUE)
summary(pca)
```

```
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
```

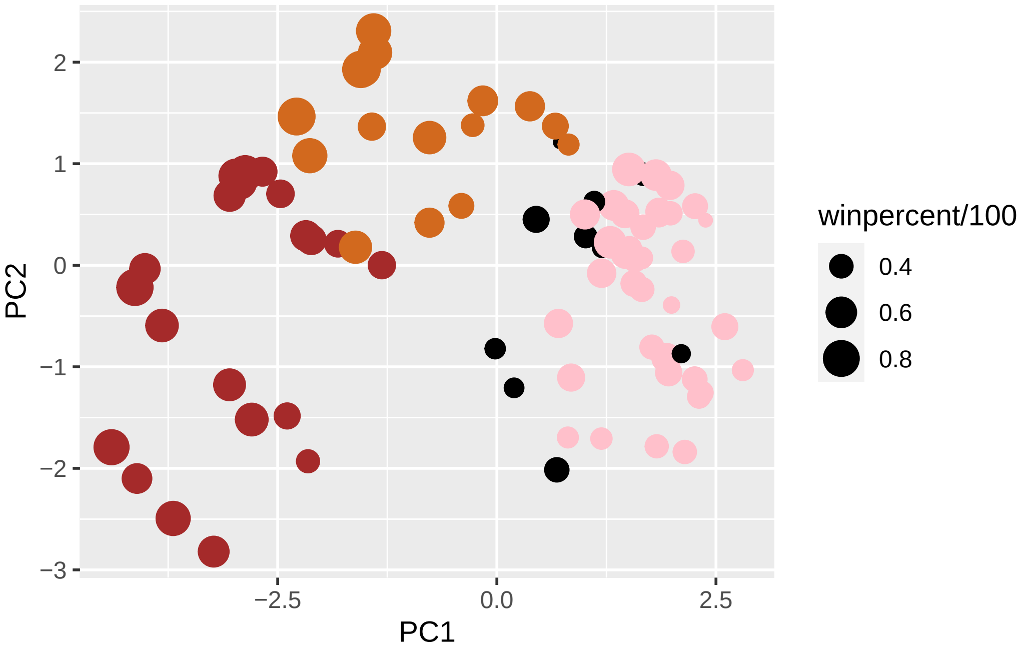
```
plot(pca$x[,1:2])
```



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

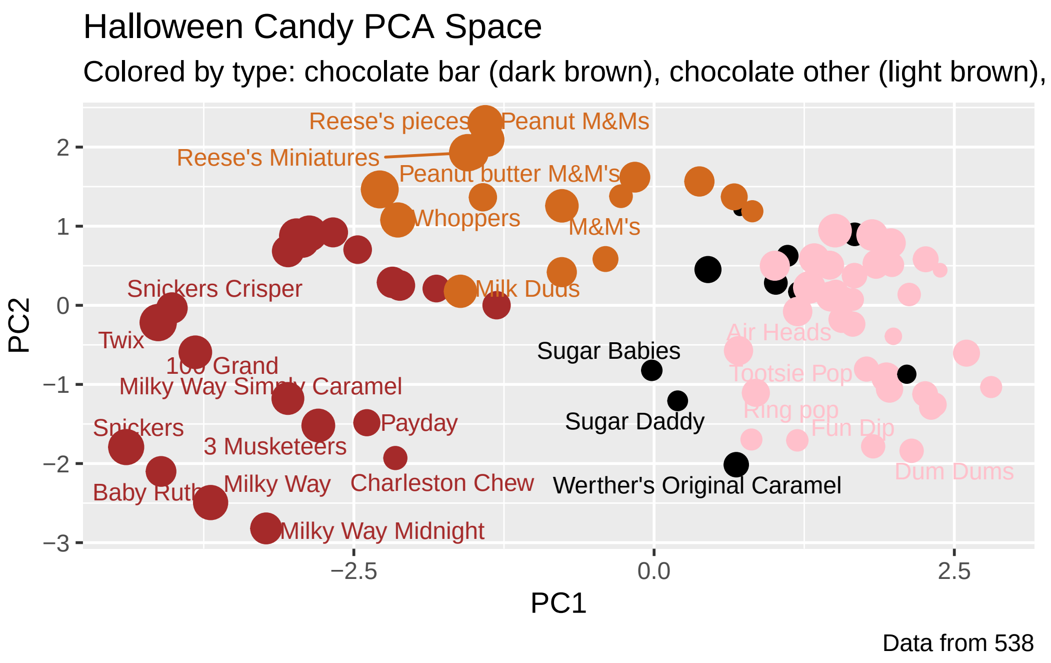


```
# Make a new data-frame with our PCA results and candy data
my_data <- cbind(candy, pca$x[,1:3])

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
```



```
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

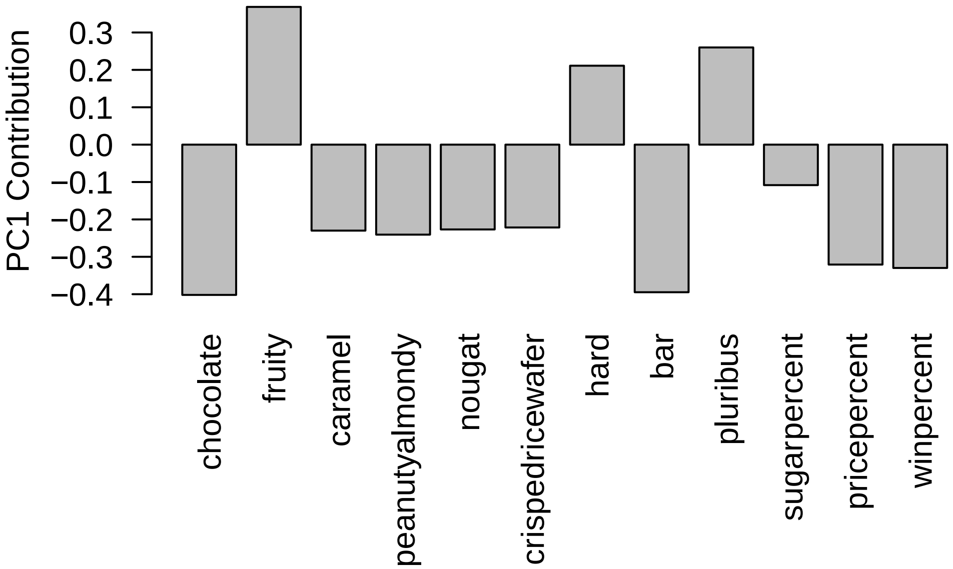
If you want to see more candy labels you can change the max.overlaps value to allow more overlapping labels or pass the ggplot object p to plotly like so to generate an interactive plot that you can mouse over to see labels:

```
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
```

```
# ggplotly(p)
```

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



Q24. fruity, hard, pluribus. PC1 has a large positive association with fruity, hard, and pluribus. This makes sense since these three factors are highly correlated. Fruity candy is usually hard and packed together in a box of multiple candies.