Class10

```
candy = read.csv('https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-powe
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

```
dim(candy)
```

[1] 85 12

Q1: 85 types Q2

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

[1] 81.64291

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

```
[1] 81.64291
```

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

```
[1] 76.7686
```

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

```
Warning: package 'skimr' was built under R version 4.1.2
```

```
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	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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	

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

Histogram of candy$winpercent
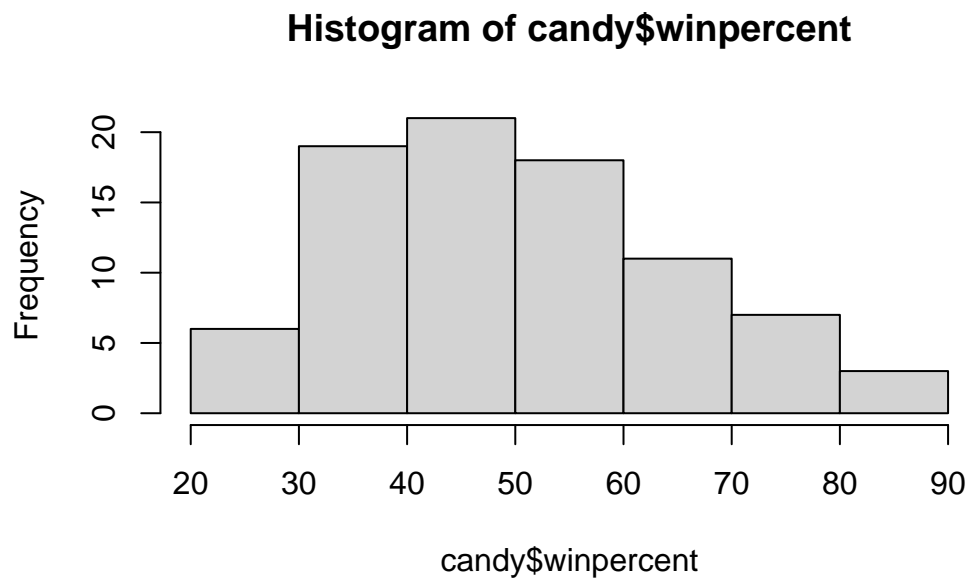

```
mean(candy$winpercent[as.logical(candy$chocolate)])
```

```
[1] 60.92153
```

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

```
[1] 44.11974
```

```
candy$winpercent[candy$chocolate]
```

```
 [1] 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173
 [9] 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173
[17] 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173
[25] 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173 66.97173
[33] 66.97173 66.97173 66.97173 66.97173 66.97173
```

```
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$winpercent[as.logical(candy$f
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
```

```
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
```

```
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
```
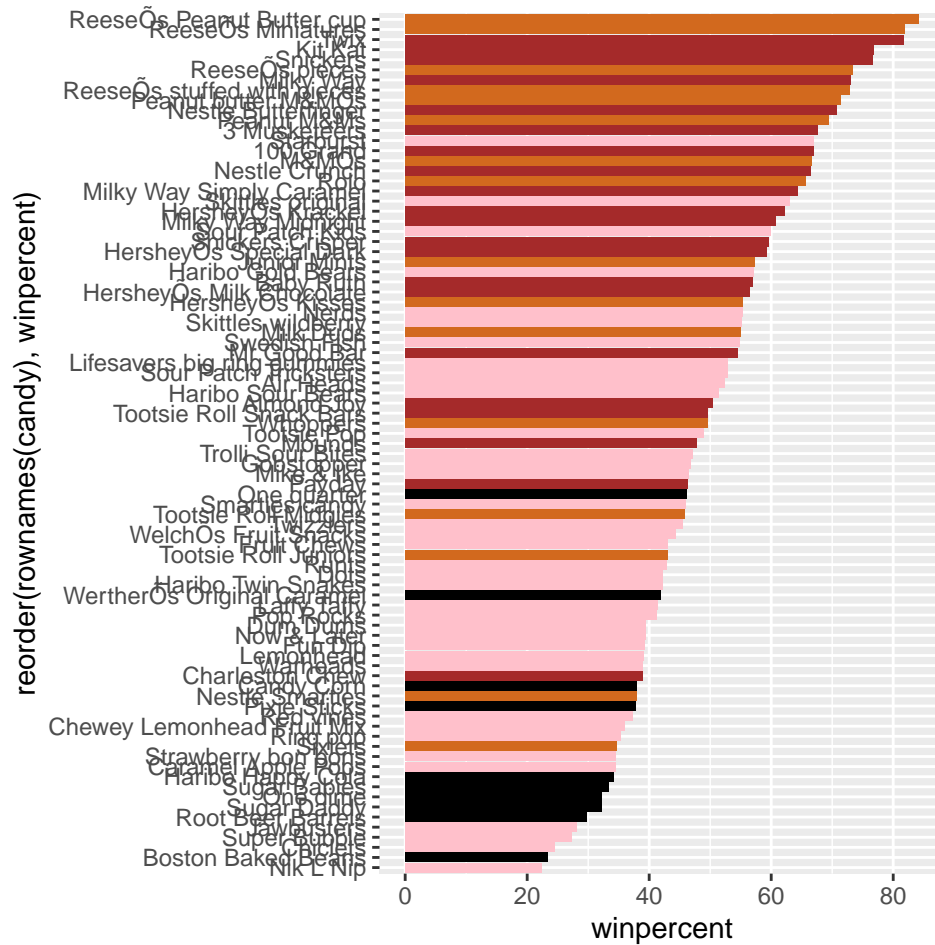
```
library(ggplot2)

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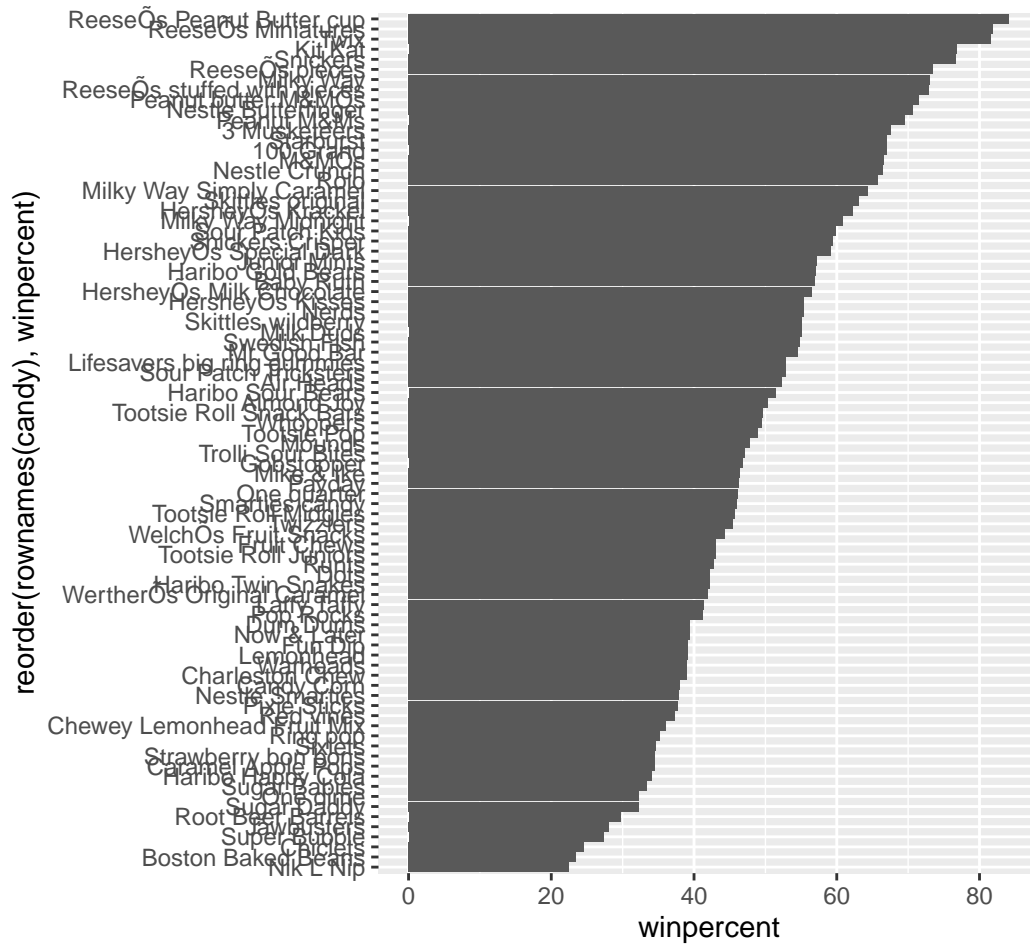


```
my_cols=rep("black", nrow(candy))
my_cols[as.logical(candy$chocolate)] = "chocolate"
my_cols[as.logical(candy$bar)] = "brown"
my_cols[as.logical(candy$fruity)] = "pink"
```

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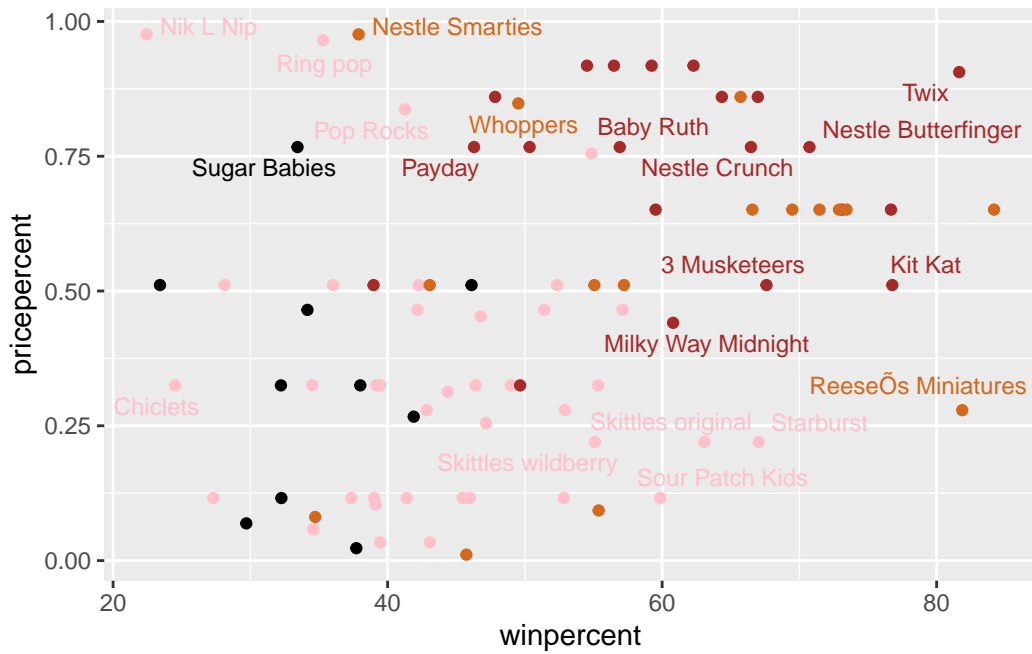


```
library(ggrepel)

# 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`


```
head(candy[order(candy$pricepercent, decreasing = T),])
```

	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
HersheyÕs Krackel	1	0	0	0	0
HersheyÕs Milk Chocolate	1	0	0	0	0
HersheyÕs Special Dark	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
HersheyÕs Krackel	1	0	1	0	0.430
HersheyÕs Milk Chocolate	0	0	1	0	0.430
HersheyÕs Special Dark	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
HersheyÕs Krackel               0.918    62.28448
HersheyÕs Milk Chocolate        0.918    56.49050
HersheyÕs Special Dark          0.918    59.23612
```
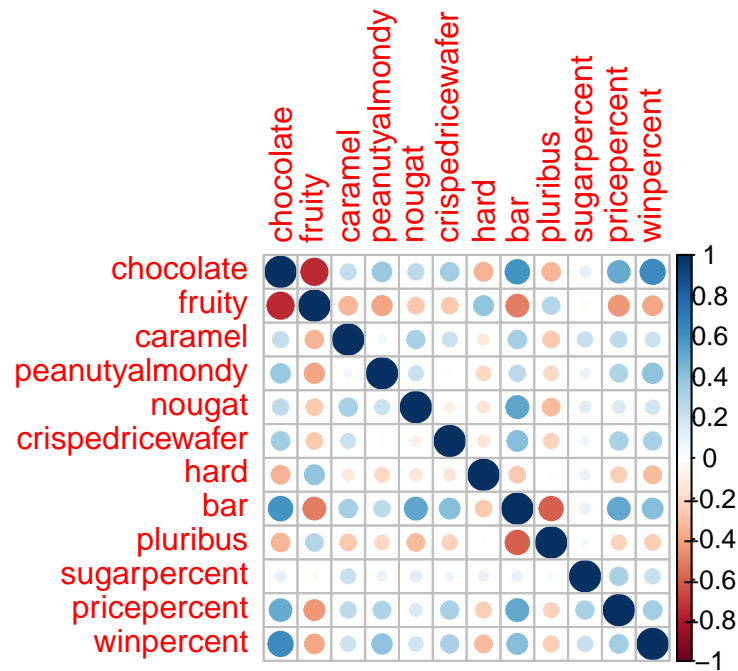
```r
# Make a lollipop chart of pricepercent
ggplot(candy) +
  aes(pricepercent, reorder(rownames(candy), pricepercent)) +
  geom_segment(aes(yend = reorder(rownames(candy), pricepercent),
                   xend = 0), col="gray40") +
  geom_point()
```
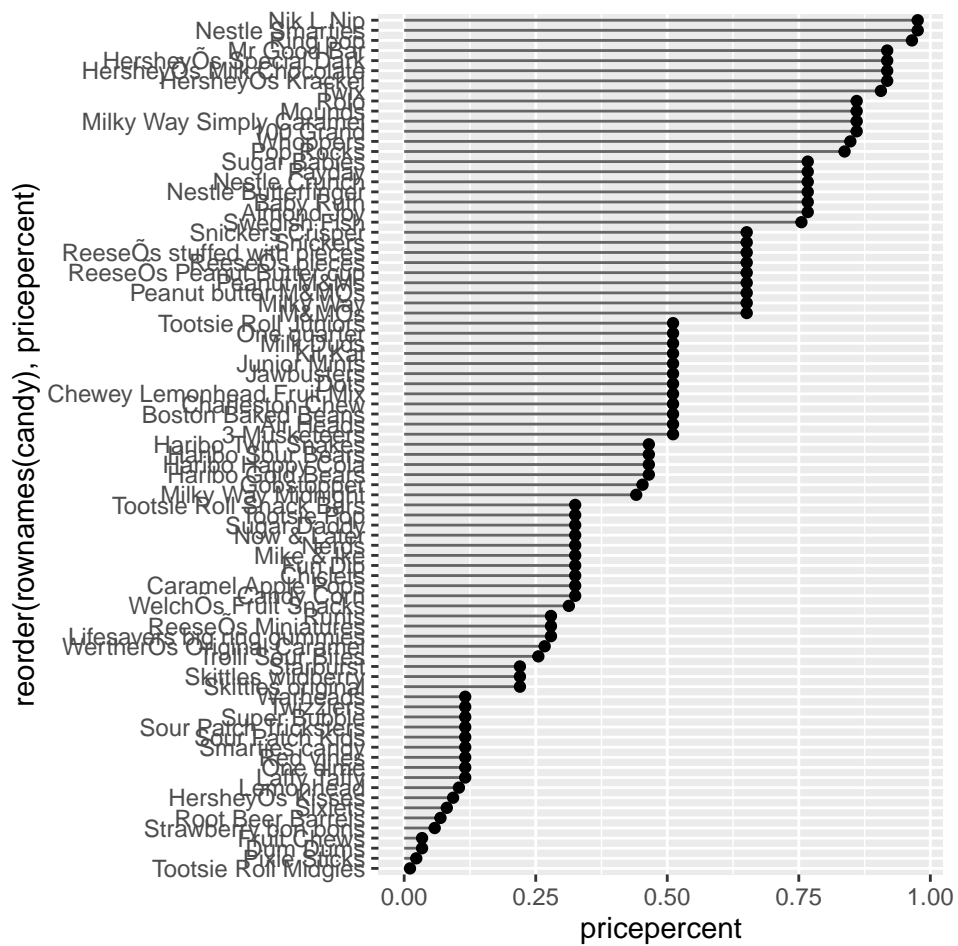


```r
library(corrplot)
```

```
corrplot 0.91 loaded
```

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


PCA

```
pca <- prcomp(candy, scale = T)
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
```
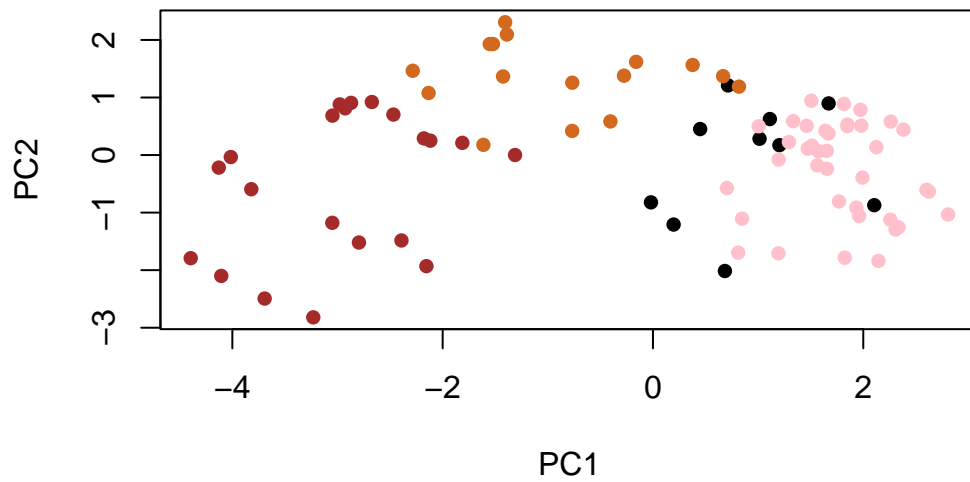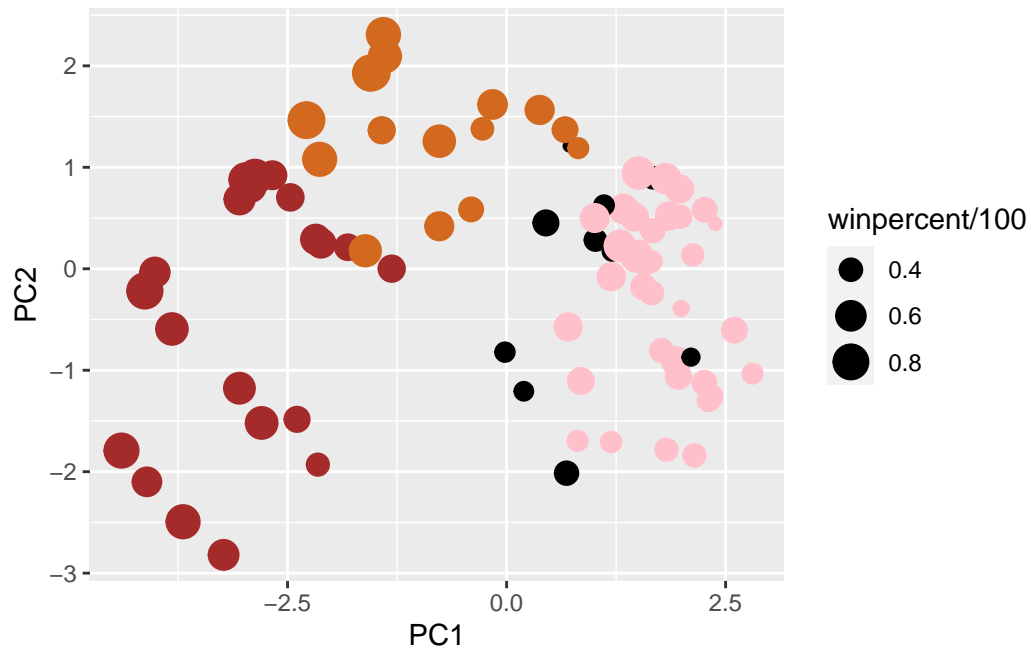
```
plot(pca$x[,1:2])
```
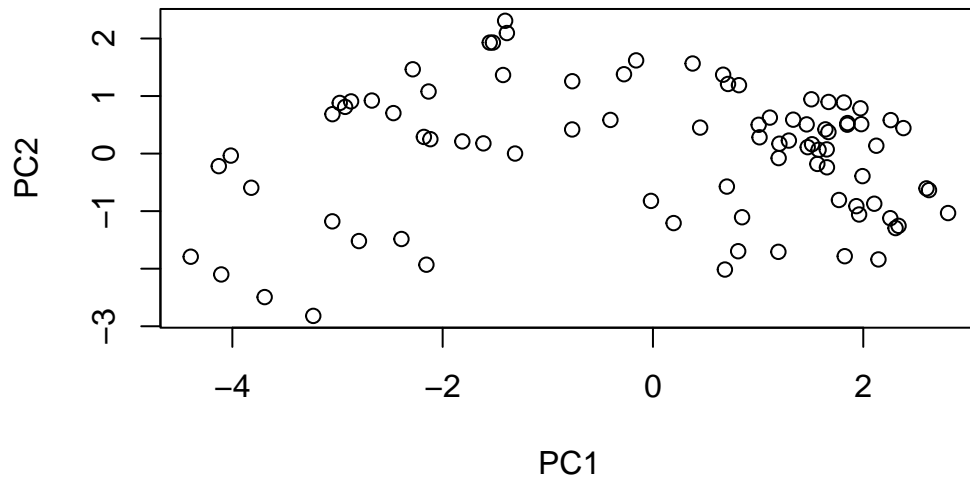


```
plot(pca$x[,1:2], col=my_cols, pch=16)
```

```
# 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
```

```
library(ggrepel)

p + geom_text_repel(size=3.3, col=my_cols, max.overlaps = 10)  +
  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: 54 unlabeled data points (too many overlaps). Consider
increasing max.overlaps

13

Halloween Candy PCA Space

Colored by type: chocolate bar (dark brown), chocolate other (light brown),
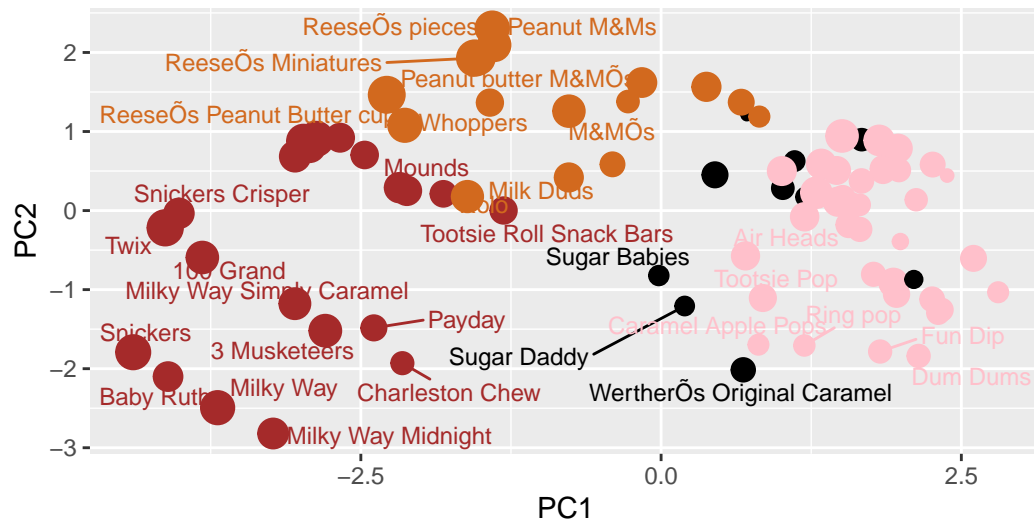
Data from 538

```r
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
```

```r
#ggplotly(p)
```

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