Class 09: Halloween Candy Mini-Project

Richard Gao (PID: A16490010)

Importing candy data

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

```
ncol(candy)
```

```
[1] 12
```

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

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

```
[1] 38
```

```
sum(candy$chocolate)
```

```
[1] 37
```

Data Exploration

Q3. What is your favorite candy in the dataset and what is it's win-percent value

```
candy["Sour Patch Kids",]$winpercent
```

```
[1] 59.864
```

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
```

Q What is the least liked candy in the dataset - lowest winpercent

```
x <- c(5, 3, 4, 1)
sort(x)
```

```
[1] 1 3 4 5
```

```r
# orders the indexes to carry out sort function
order(x)
```

```
[1] 4 2 3 1
```

```r
inds <- order(candy$winpercent)
# sorted by winpercent
head(candy[inds,])
```

```
                   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
Root Beer Barrels          0      0       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
Root Beer Barrels                 0    1   0        1        0.732        0.069
                   winpercent
Nik L Nip            22.44534
Boston Baked Beans   23.41782
Chiclets             24.52499
Super Bubble         27.30386
Jawbusters           28.12744
Root Beer Barrels    29.70369
```

```r
# 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?

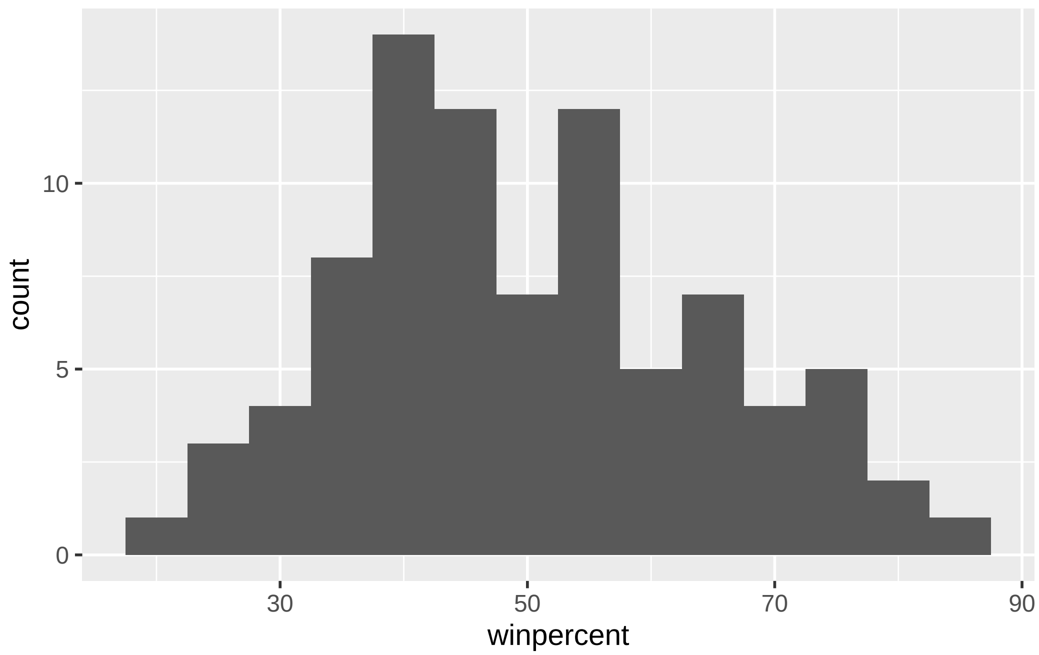
Yes, row 12 (winpercent) is on a scale of 0 to 100 while the other rows are on a scale of 0 to 1 or are either only 0 or 1.

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

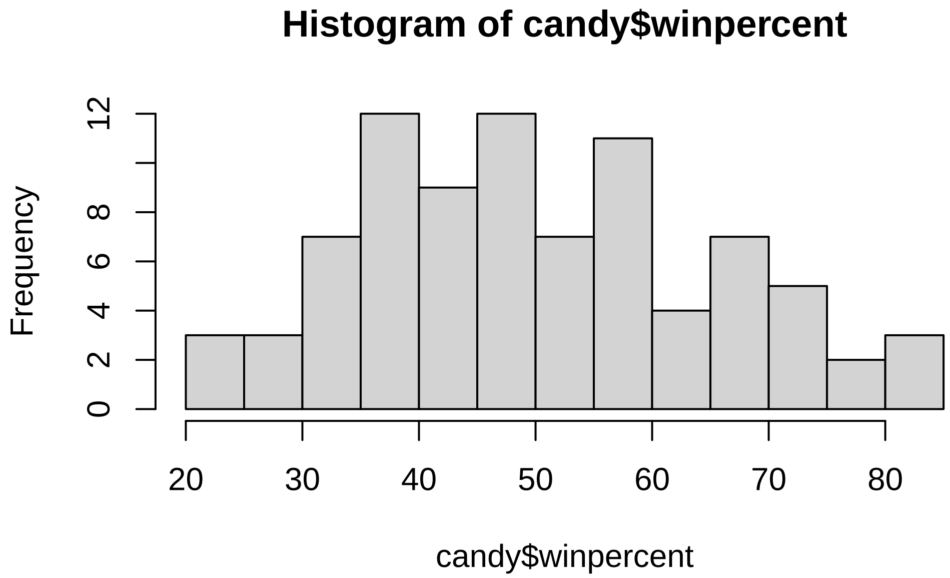
Zero represents false and one represents true.

> Q8. Plot a histogram of winpercent values

```r
library(ggplot2)
ggplot(candy, aes(winpercent)) +
  geom_histogram(binwidth = 5)
```



```
# or
hist(candy$winpercent, breaks = 20)
```



Q9. Is the distribution of winpercent values symmetrical?

No, it is slightly right-skewed.

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

Below:

```r
median(candy$winpercent)
```

```
[1] 47.82975
```

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

First, find all the chocolate candy and their winpercent values

Next, summarize these values into one number

Then do the same for fruit candy and compare their numbers

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

```
[1] 60.92153
```

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

```
[1] 44.11974
```

Or

```r
choc.inds <- as.logical(candy$chocolate)
choc.win <- candy[choc.inds,]$winpercent
mean(choc.win)
```

```
[1] 60.92153
```

```r
fruit.inds <- as.logical(candy$fruity)
fruit.win <- candy[fruit.inds,]$winpercent
mean(fruit.win)
```

```
[1] 44.11974
```

Q12. Is this difference statistically significant?

```
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 
```

Yes this is statistically significant since the p-value is < 0.05

Overall Candy Rankings

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

```
head(candy[inds,])
```

```
                   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
Root Beer Barrels          0      0       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
Root Beer Barrels                 0    1   0        1        0.732        0.069
                   winpercent
Nik L Nip            22.44534
Boston Baked Beans   23.41782
Chiclets             24.52499
```

```
Super Bubble         27.30386
Jawbusters           28.12744
Root Beer Barrels    29.70369
```

Nik L Nip, Boston Baked Beans, Chiclets, Super Bubble, Jawbusters

Q14. What are the top 5 all time favorite candy types out of this set?

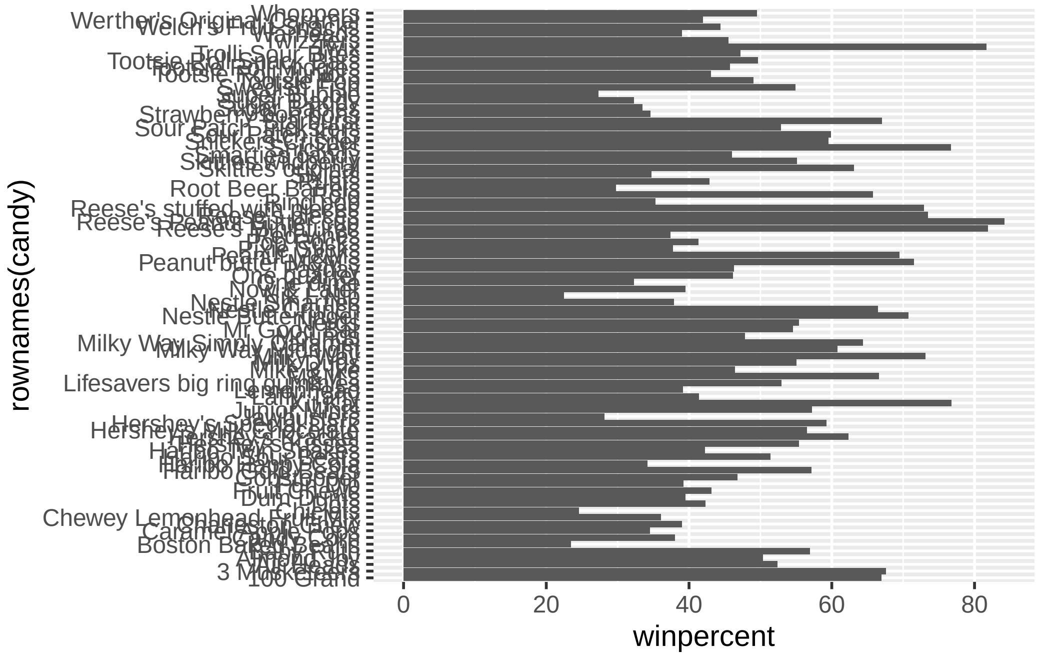
```
tail(candy[inds,])
```

```
                          chocolate fruity caramel peanutyalmondy nougat
Reese's pieces                    1      0       0              1      0
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
Reese's pieces                           0    0   0        1        0.406
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
Reese's pieces                   0.651   73.43499
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
```

Snickers, Kit Kat, Twix, Reese's Miniatures, Reese's Peanut Butter cup

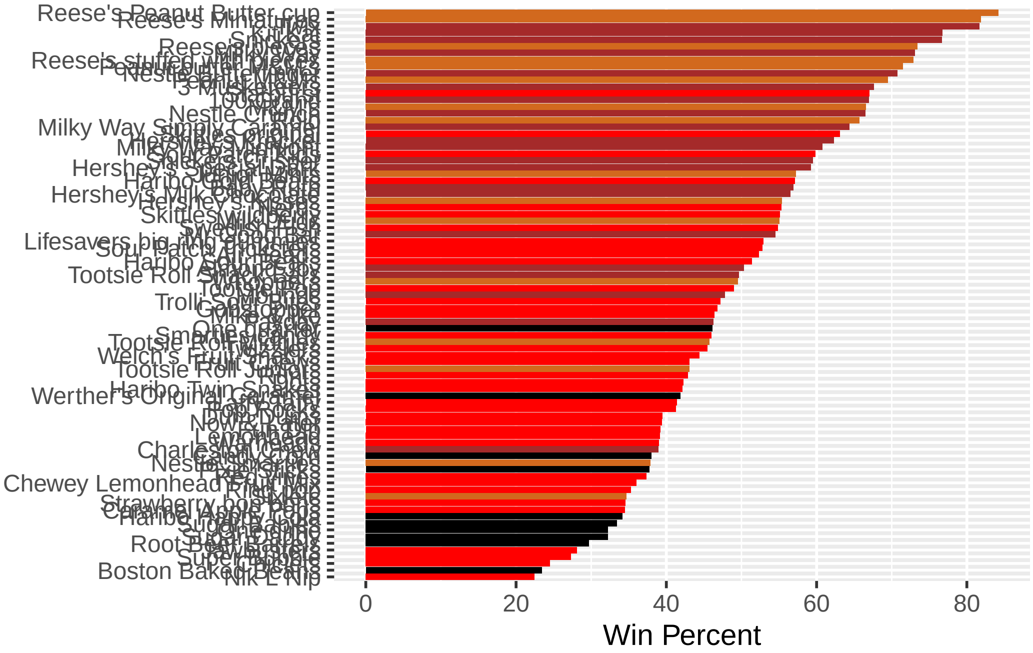
Q15. Make a first barplot of candy ranking based on winpercent values.

```
library(ggplot2)
ggplot(candy) +
  aes(winpercent, rownames(candy)) +
  geom_col()
```

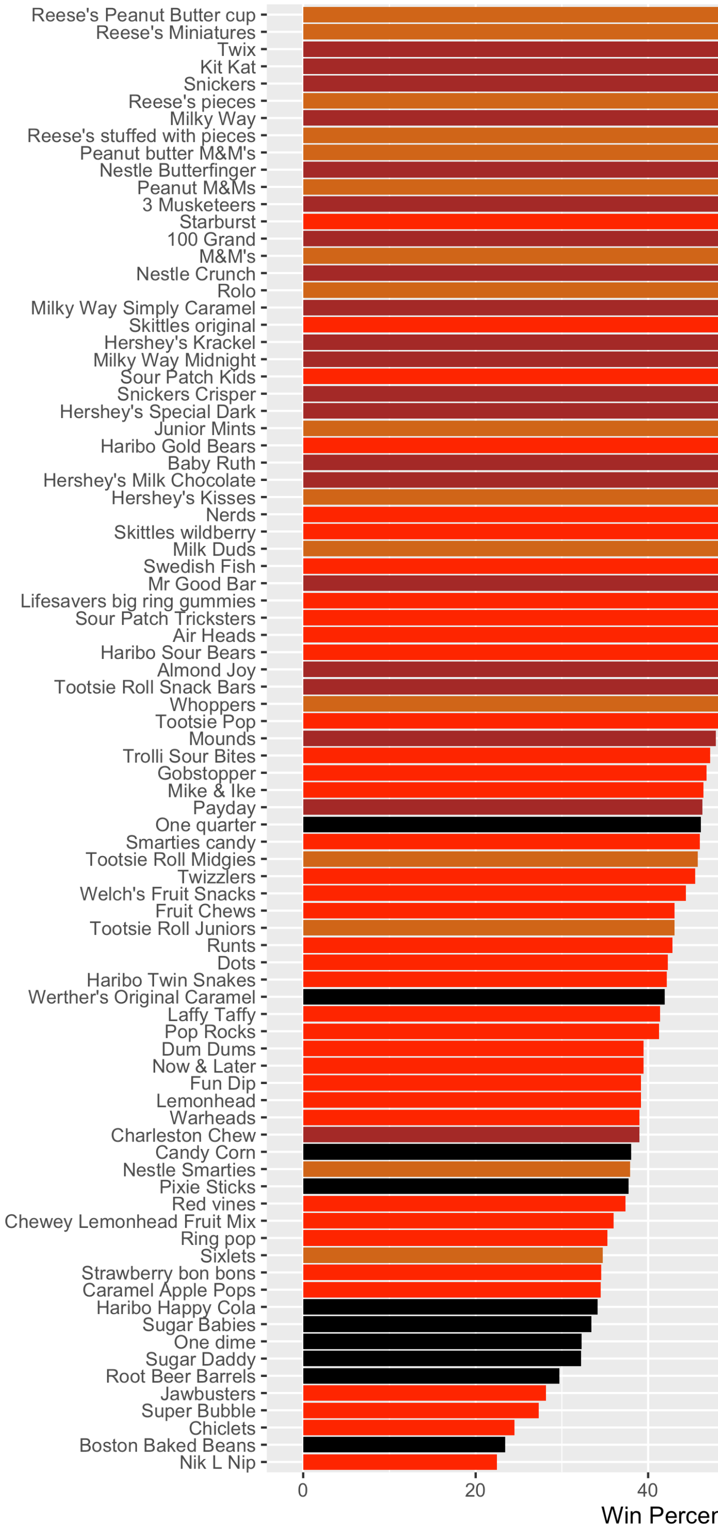


Q16. This is quite ugly, use the reorder() function to get the bars sorted by winpercent?

```
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)] = "red"

ggplot(candy) +
  aes(winpercent, reorder(rownames(candy), winpercent)) +
  geom_col(fill = my_cols) +
  labs(x="Win Percent", y=NULL)
```



```
ggsave('barplot1.png', width=7, height=10)
```



You can insert any image using this markdown syntax:

Q17. What is the worst ranked chocolate candy?

Sixlets

Q18. What is the best ranked fruity candy?

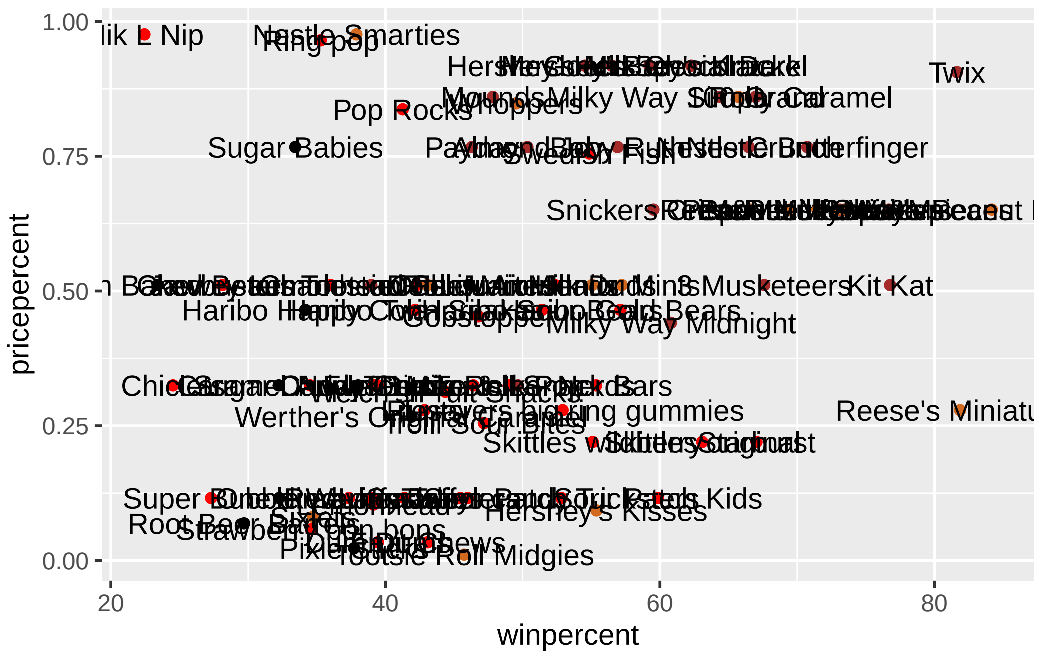
Starbursts

Taking a look at pricepercent

```
candy$pricepercent
```

```
 [1] 0.860 0.511 0.116 0.511 0.511 0.767 0.767 0.511 0.325 0.325 0.511 0.511
[13] 0.325 0.511 0.034 0.034 0.325 0.453 0.465 0.465 0.465 0.465 0.093 0.918
[25] 0.918 0.918 0.511 0.511 0.511 0.116 0.104 0.279 0.651 0.651 0.325 0.511
[37] 0.651 0.441 0.860 0.860 0.918 0.325 0.767 0.767 0.976 0.325 0.767 0.651
[49] 0.023 0.837 0.116 0.279 0.651 0.651 0.651 0.965 0.860 0.069 0.279 0.081
[61] 0.220 0.220 0.976 0.116 0.651 0.651 0.116 0.116 0.220 0.058 0.767 0.325
[73] 0.116 0.755 0.325 0.511 0.011 0.325 0.255 0.906 0.116 0.116 0.313 0.267
[85] 0.848
```

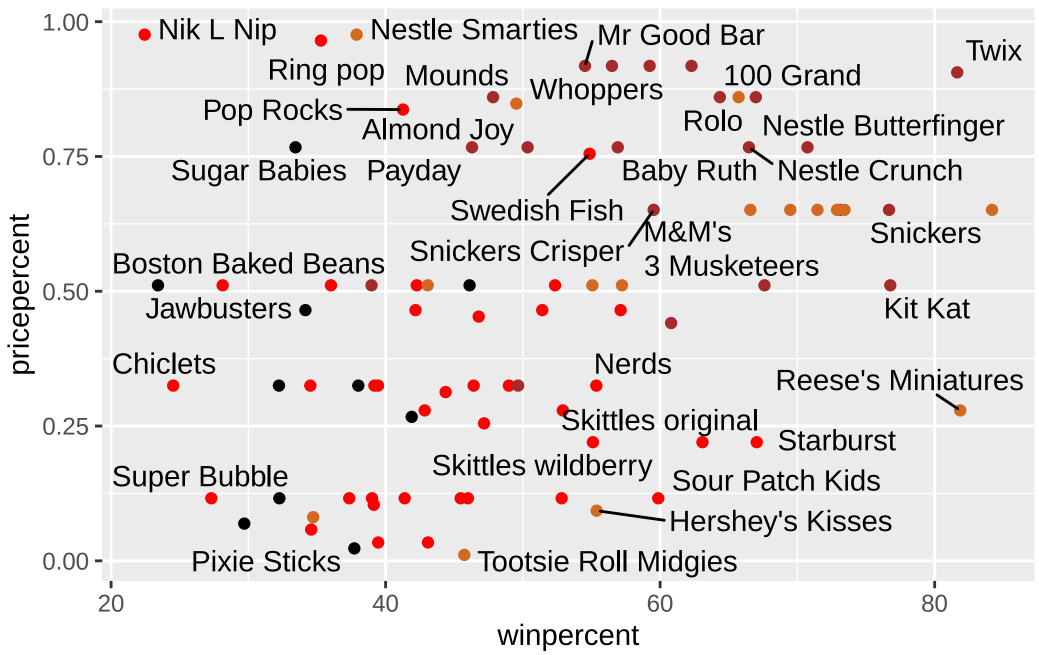
```
ggplot(candy) +
  aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = my_cols) +
  geom_text()
```



To avoid overplotting of all these labels we can use an add on package called ggrepel

```
# install.packages("ggrepel")
library(ggrepel)
ggplot(candy) +
  aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = my_cols) +
  geom_text_repel()
```

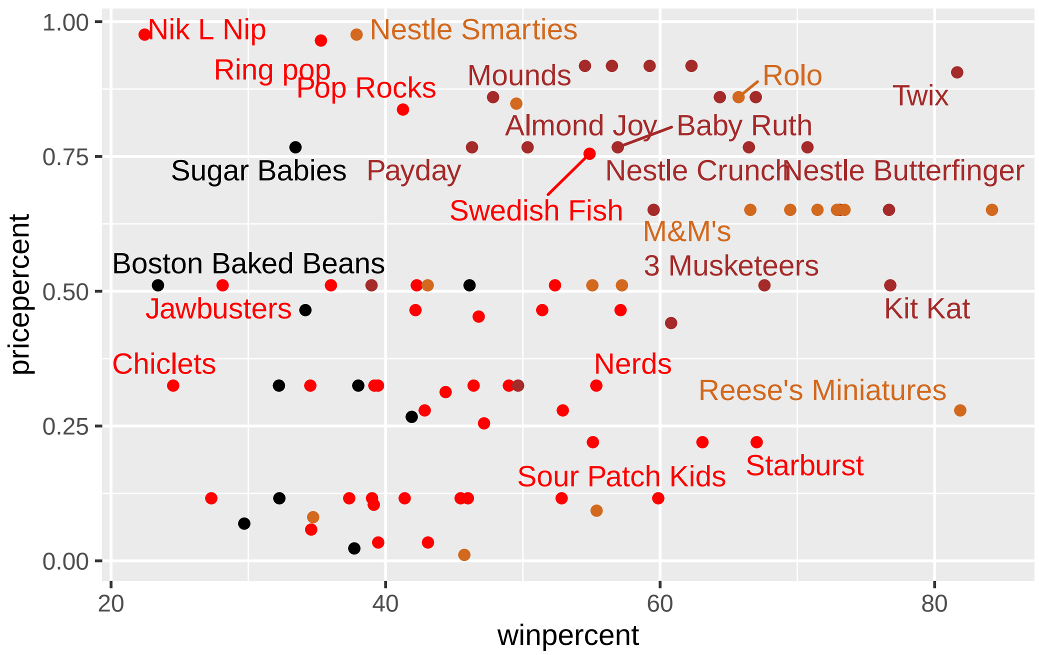
```
Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Play with the max.overlaps parameter to geom_text_repel()

```
ggplot(candy) +
  aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = my_cols) +
  geom_text_repel(max.overlaps = 8, col = my_cols)
```

```
Warning: ggrepel: 61 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?

Based off the plot, Reese's miniatures

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

```r
tail(candy[order(candy$pricepercent),])
```

```
                         chocolate fruity caramel peanutyalmondy nougat
Hershey's Milk Chocolate         1      0       0              0      0
Hershey's Special Dark           1      0       0              0      0
Mr Good Bar                      1      0       0              1      0
Ring pop                         0      1       0              0      0
Nik L Nip                        0      1       0              0      0
Nestle Smarties                  1      0       0              0      0
                         crispedricewafer hard bar pluribus sugarpercent
Hershey's Milk Chocolate                0    0   1        0        0.430
Hershey's Special Dark                  0    0   1        0        0.430
Mr Good Bar                             0    0   1        0        0.313
Ring pop                                0    1   0        0        0.732
Nik L Nip                               0    0   0        1        0.197
```

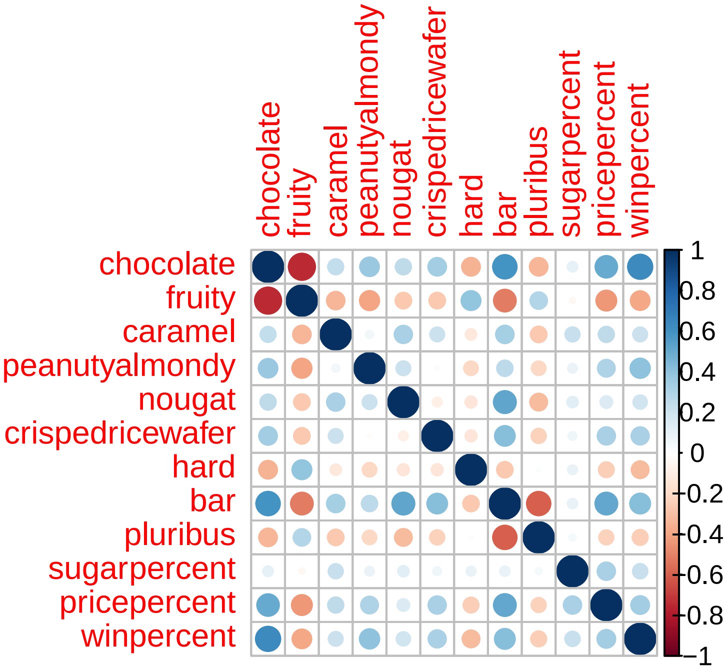
```
Nestle Smarties                                0    0   0        1        0.267
                         pricepercent winpercent
Hershey's Milk Chocolate        0.918   56.49050
Hershey's Special Dark          0.918   59.23612
Mr Good Bar                     0.918   54.52645
Ring pop                        0.965   35.29076
Nik L Nip                       0.976   22.44534
Nestle Smarties                 0.976   37.88719
```

Nestle Smarties, Nik L Nip, Ring pop, Mr Good Bar, Hershey's Special Dark - the least popular is Nik L Nip.

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

Chocolate and Fruity.

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

Chocolate and winpercent.

On to PCA

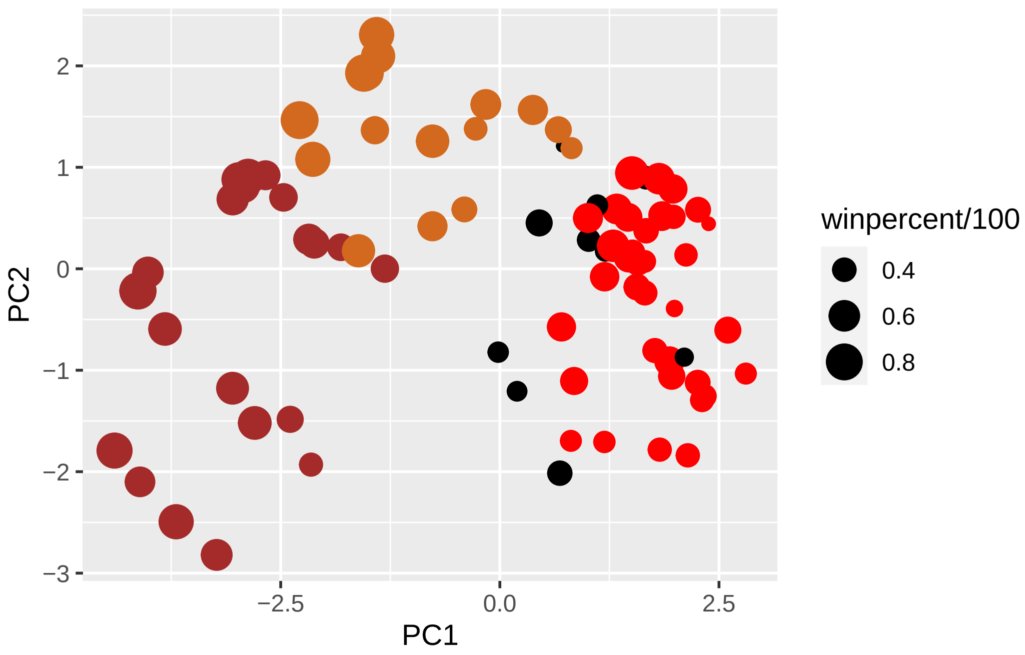
The main function for this is called `prcomp()` and here we know we need to scale our data with the `scale=TRUE` argument.

```
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
```

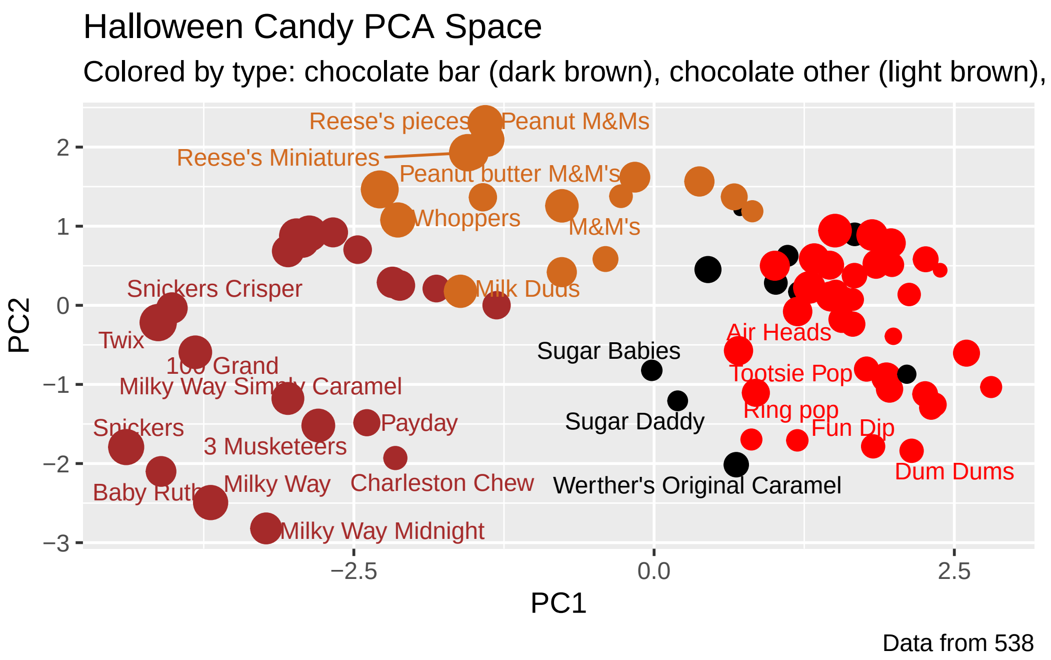
```
# Make a new data frame with our PCA results and candy data
my_data <- cbind(candy, pca$x[, 1:3])
```

```
p <- ggplot(my_data) +
  aes(PC1, 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
```



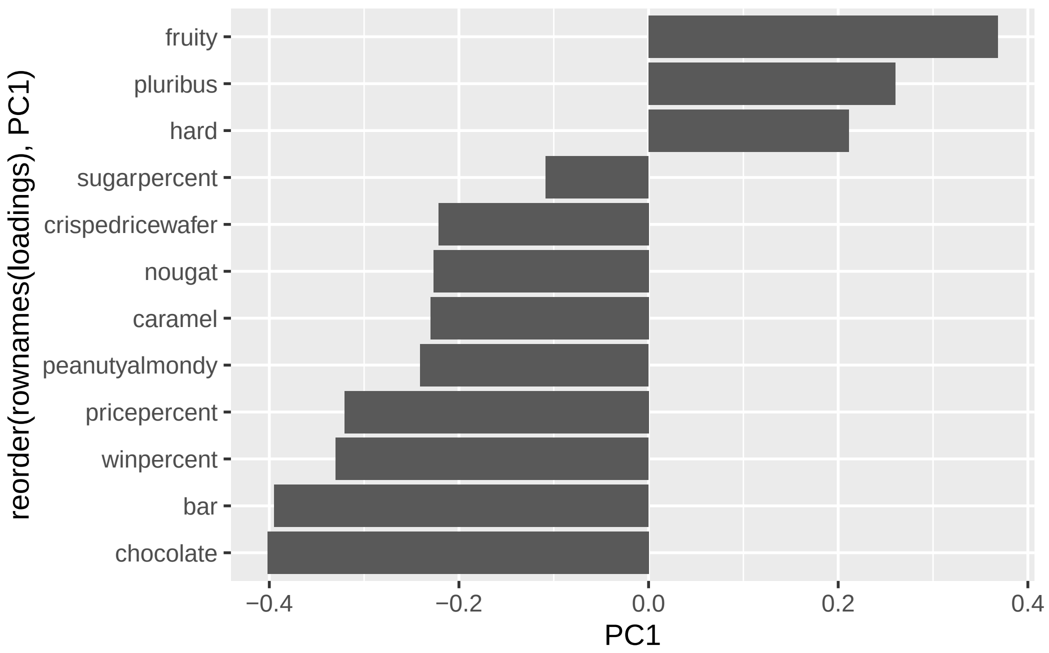
Halloween Candy PCA Space

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

Data from 538

loadings plot

```
loadings <- as.data.frame(pca$rotation)

ggplot(loadings) +
  aes(PC1, reorder(rownames(loadings), PC1)) +
  geom_col()
```



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

Fruity, pluribus, hard - this does make sense because it lines up with the correlation matrix earlier and fruity candy tends to come in packages with many hard pieces.