class09

Laurie Chang A16891192

Here we analyze a candy dataset from he 538 website. This is a CSV file from their GitHub respository.

Importing Candy Data

```
candy <- read.csv("candy-data.csv", 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?

```
dim(candy)
```

```
[1] 85 12
```

There are 85 different types of candy in this dataset.

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

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

```
[1] 38
```

There are 38 fruity candy types in this dataset.

What is your favorite candy?

One of the most interesting variables in the dataset is winpercent. For a given candy this value is the percentage of people who prefer this candy over another randomly chosen candy from the dataset (what 538 term a matchup). Higher values indicate a more popular candy.

We can find the winpercent value for Twix by using its name to access the corresponding row of the dataset. This is because the dataset has each candy name as rownames (recall that we set this when we imported the original CSV file). For example the code for Twix is:

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

```
[1] 81.64291
```

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

Kit Kat is my favorite candy.

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

```
[1] 76.7686
```

Kit Kat has a win percent of 76.7686%.

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

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

```
[1] 76.7686
```

Kit Kat has a win percent of 76.7686%.

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

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

```
[1] 49.6535
```

Tootsie Roll Snack Bars has a win percent of 49.6535%.

SORT AND ORDER

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

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

Puts the numbers from smallest to largest.

```
order(x)
```

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

Tells you the bucket number the smallest number is in, second smallest, etc.

> Which candy has the smallest win percent?

```
inds <- order(candy$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
```

KEY POINT

Looking at one variable, sorting it by whatever you are interested in, and then applying it the whole table.

Using `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	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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?

win percent.

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

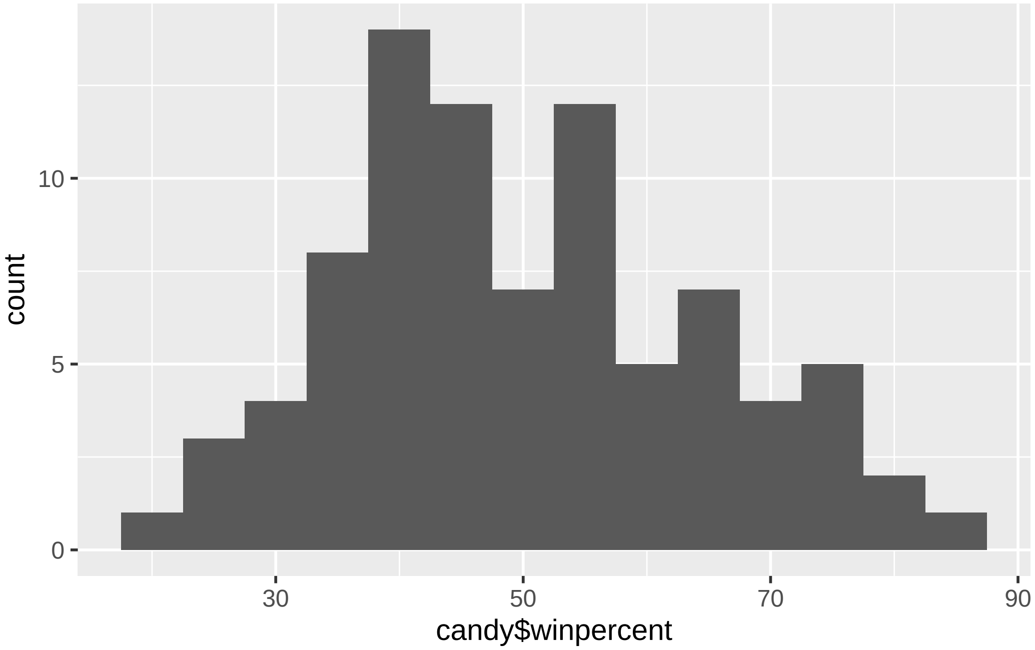
zero = no chocolate one = yes chocolate

PLOTTING

A good place to start any exploratory analysis is with a histogram.

> Q8. Plot a histogram of winpercent values

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



> Q9. Is the distribution of winpercent values symmetrical?

The distribution of winpercent values is not symmetrical.

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

The center of the distribution looks to be below 50%.

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

First find all chocolate candy and their $winpercent values.

Next, summarize these values into one number.

Then do the same for fruit candy and compare the numbers.

```
winpercent.chocolate <- candy$winpercent[as.logical(candy$chocolate)]
winpercent.fruity <- candy$winpercent[as.logical(candy$fruity)]

averagechocolate <- mean(candy$winpercent[as.logical(candy$chocolate)])
averagefruity <- mean(candy$winpercent[as.logical(candy$fruity)])
```

With an average win percent of 60.92153%, chocolate candy is higher ranked than fruit candy (44.11974%).

> Q12. Is this difference statistically significant?

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

```
    Welch Two Sample t-test

data:  winpercent.chocolate and winpercent.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 
```

Since there is an extremely small p-value, the reuslts are significant.

Overall Candy Rankings

Let's use the base R order() function together with head() to sort the whole dataset by winpercent:

```r
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
```

Or if you have been getting into the tidyverse and the dplyr package you can use the arrange() function together with head() to do the same thing:

```
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
```

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

Nik L Nip, Boston Baked Beans, Chiclets, Super Bubble, and Jawbusters are the 5 least liked candy types in this set.

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

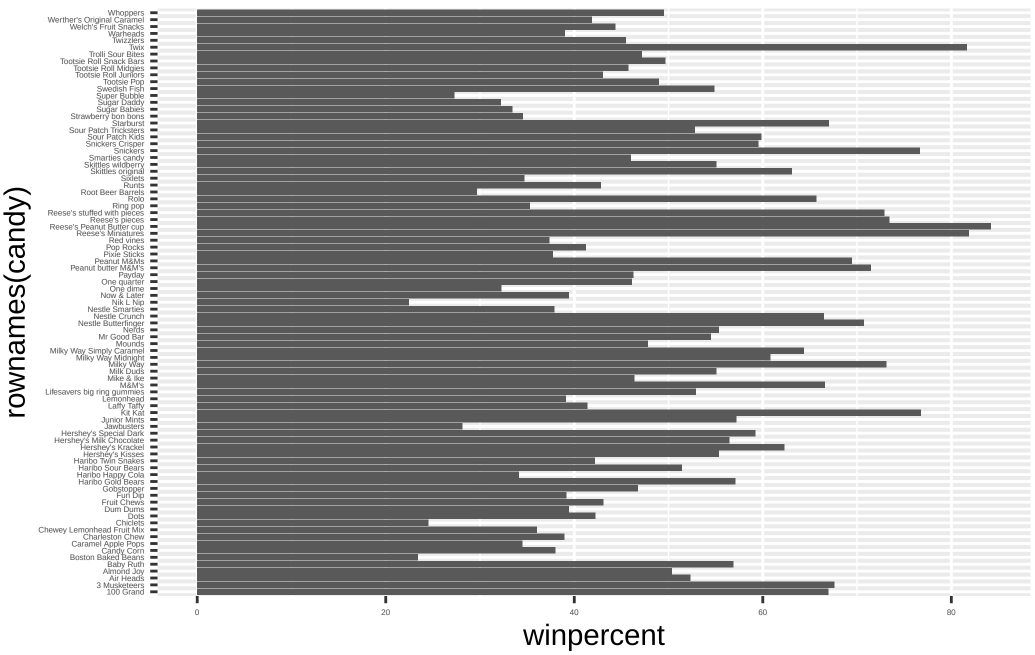
```
candy %>% arrange(desc(winpercent)) %>% head(5)
```

```
                          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
```

Reese’s Peanut Butter Cup, Reese’s Miniatures, Twix, Kit Kat, and Snickers are the top 5 all time favorite candy types out of this set.

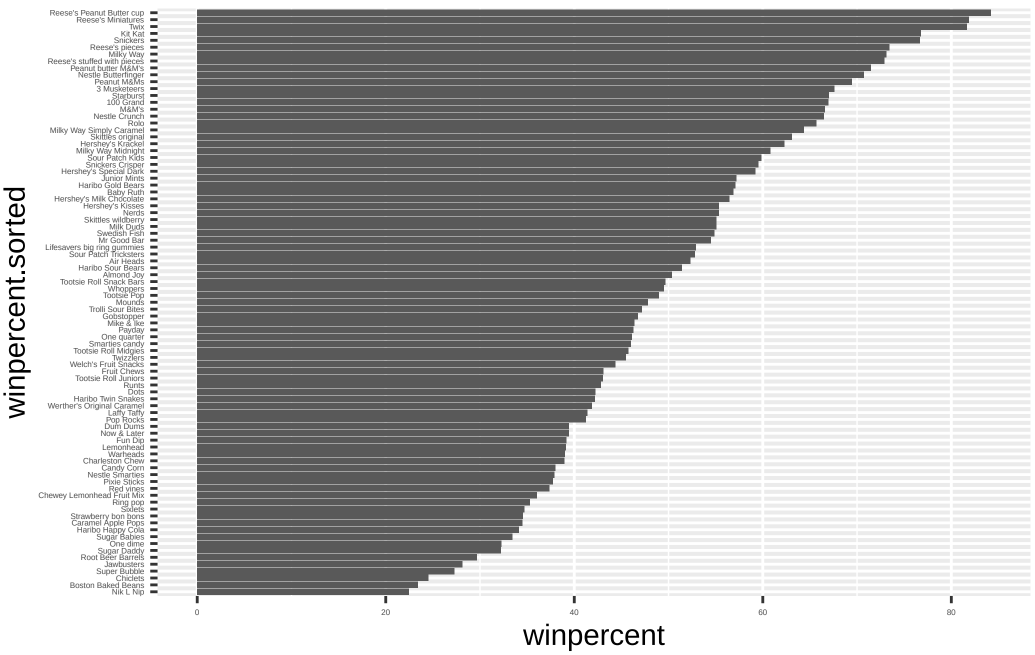
> Q15. Make a first barplot of candy ranking based on winpercent values.

```
ggplot(candy) +
  aes(winpercent, rownames(candy)) +
  geom_col() +
  theme(
    axis.text = element_text(size = 3)
  )
```



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

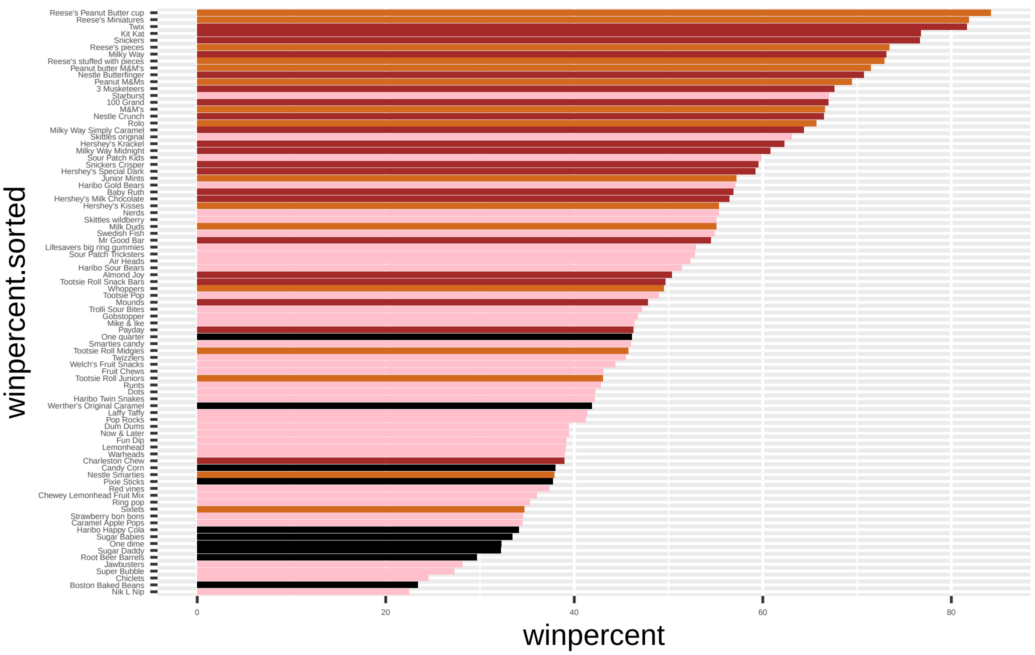
```
winpercent.sorted <- reorder(rownames(candy), candy$winpercent)

ggplot(candy) +
  aes(winpercent, winpercent.sorted) +
  geom_col() +
  theme(
    axis.text = element_text(size = 3)
  )
```



Let's setup a color vector (that signifies candy type) that we can then use for some future plots. We start by making a vector of all black values (one for each candy). Then we overwrite chocolate (for chocolate candy), brown (for candy bars) and red (for fruity candy) values.

```
#Start with an all black number of vectors that is equal to the number of rows
my_cols <- rep("black", nrow(candy))
# means repeat black for the number of rows that exist in candy
my_cols[as.logical(candy$chocolate)] <- "chocolate"
# colors chocolate for every true that is called
my_cols[as.logical(candy$bar)] <- "brown"
my_cols[as.logical(candy$fruity)] <- "pink"
```

```
ggplot(candy) +
  aes(winpercent, winpercent.sorted) +
  geom_col(fill = my_cols) +
  theme(
    axis.text = element_text(size = 3)
  )
```



Q17. What is the worst ranked chocolate candy?

Sixlets

Q18. What is the best ranked fruity candy?

Starburst

INSERTING PICTURES `![](filename)` should appear when rendered can add caption text inside the square parenthesis can also add an image link inside the parenthesis

Taking a look at pricepercent

What about value for money? What is the the best candy for the least money? One way to get at this would be to make a plot of winpercent vs the pricepercent variable. The pricepercent variable records the percentile rank of the candy's price against all the other candies in the dataset. Lower vales are less expensive and high values more expensive.

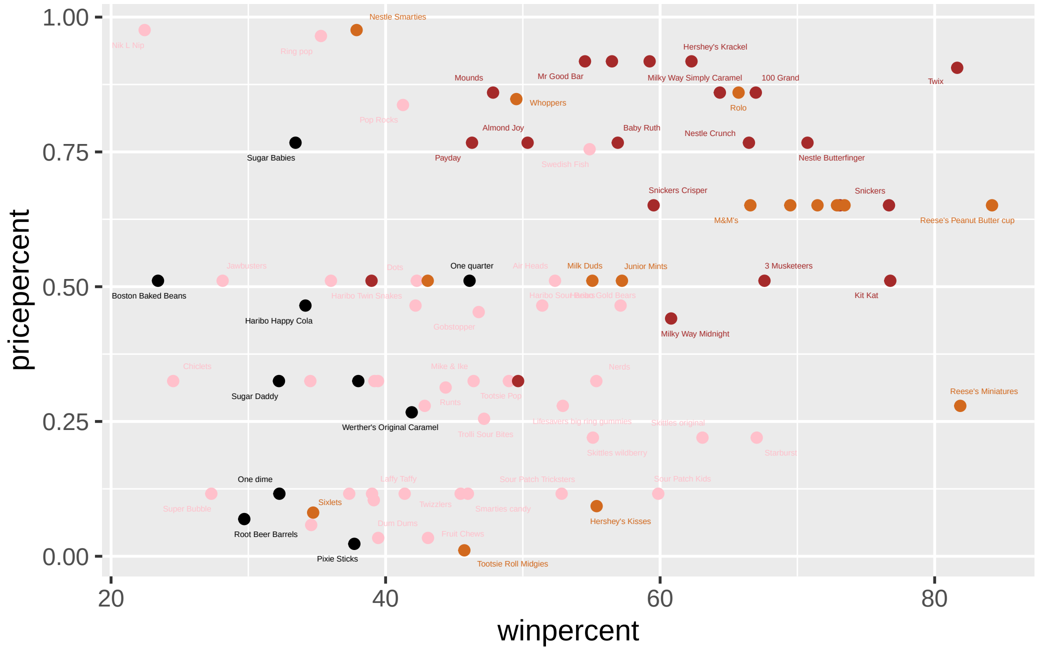
If we want to see what is a good candy to buy in terms of winpercent and pricepercent we can plot these two variables and then see the best candy for the least amount of money.

To this plot we will add text labels so we can more easily identify a given candy. There is a regular geom_label() that comes with ggplot2. However, as there are quite a few candys in our dataset lots of these labels will be overlapping and hard to read. To help with this we can use the geom_text_repel() function from the ggrepel package.

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

```
library(ggrepel)

ggplot(candy) +
  aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = my_cols) +
  geom_text_repel(col= my_cols, size=1.0, max.overlaps = 5)
```

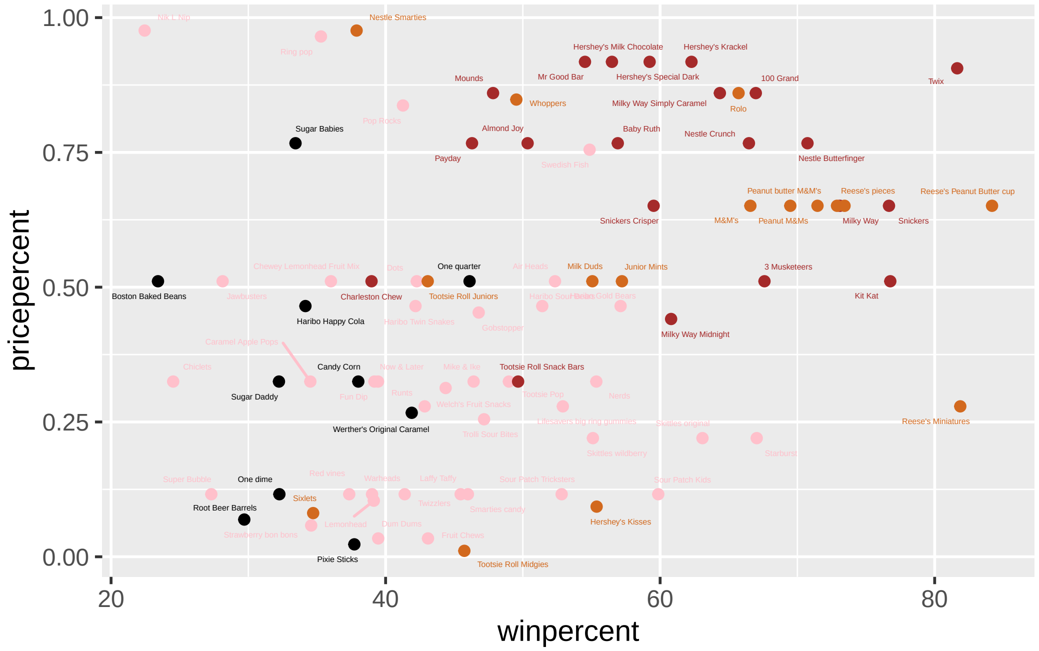
```
Warning: ggrepel: 20 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Play with the max overlaps parameter to see how that changes the plot.

```
library(ggrepel)

ggplot(candy) +
  aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = my_cols) +
  geom_text_repel(col= my_cols, size=1.0, max.overlaps = 10)
```

```
Warning: ggrepel: 1 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?

Tootsie Roll Midgies

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

```
                     pricepercent winpercent
Tootsie Roll Midgies        0.011   45.73675
Pixie Sticks                0.023   37.72234
Dum Dums                    0.034   39.46056
Fruit Chews                 0.034   43.08892
Strawberry bon bons         0.058   34.57899
```

> 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, Nestle Smarties, Ring pop, Hershey's Krackel, and Hershey's Milk Chocolate are the top 5 most expensive candy types in the dataset. Nik L Nip are the least popular candy out of the 5 most expensive.

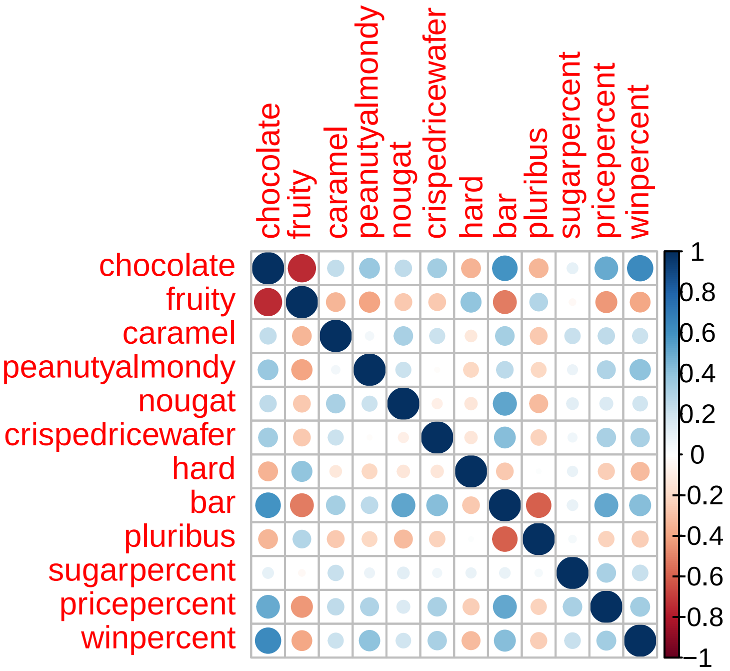
*We can also do this for which is the least expensive, see Q19.

Exploring Correlation Structure

```
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?

Winpercent and chocolate Chocolate and bar

Principal Component Analysis

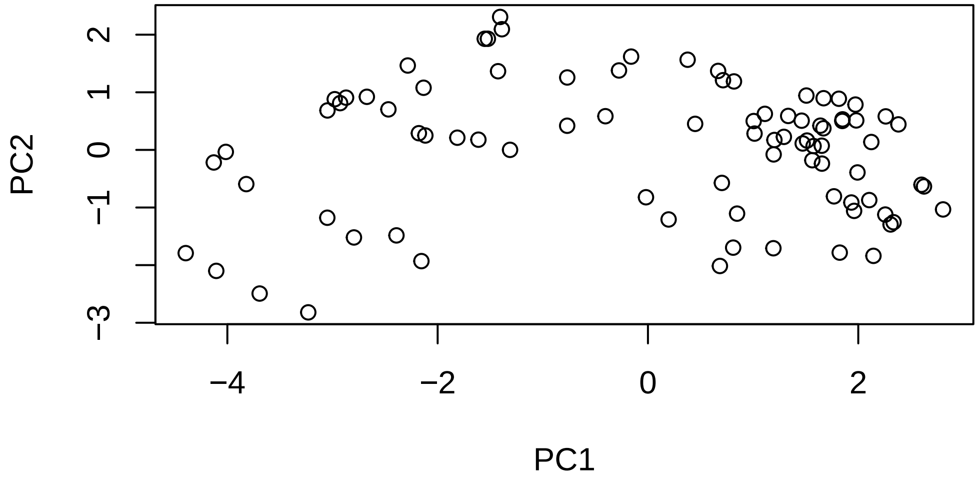
The main function for this is `prcomp()` and here we know we need to scale our data with the `scale = TRUE` argument (or else winpercent will overpower all other variables due to those values being much larger than all the other values).

```
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])
```

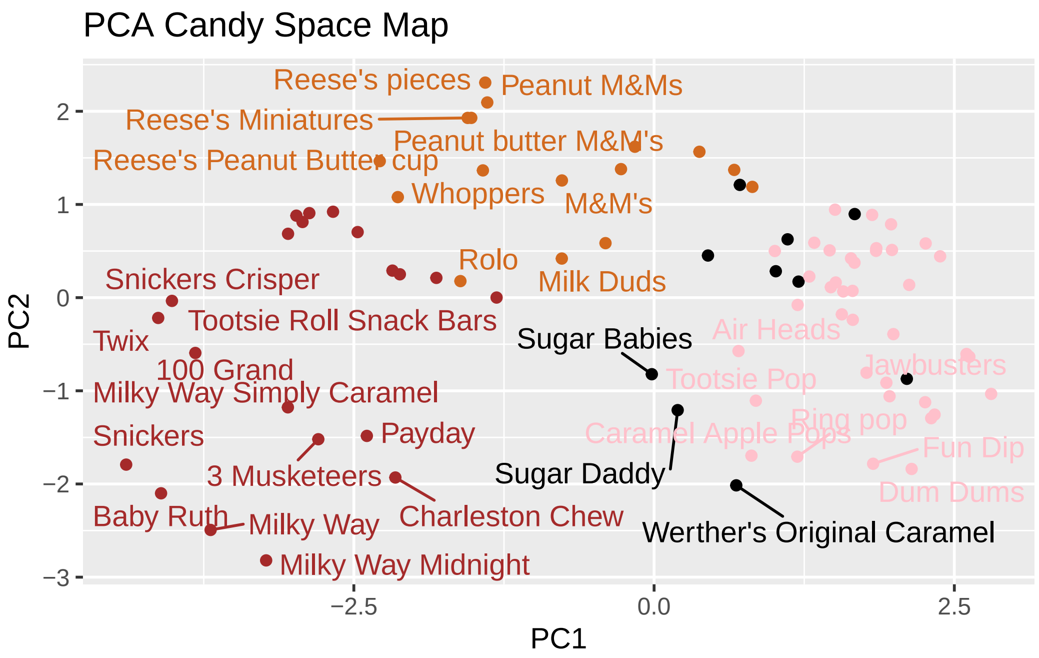


Plot my main PCA score plot with ggplot

```
# 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,label=rownames(candy)) +
        geom_point(col=my_cols) +
        geom_text_repel(col = my_cols) +
        labs(title = "PCA Candy Space Map")

p
```

```
Warning: ggrepel: 54 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



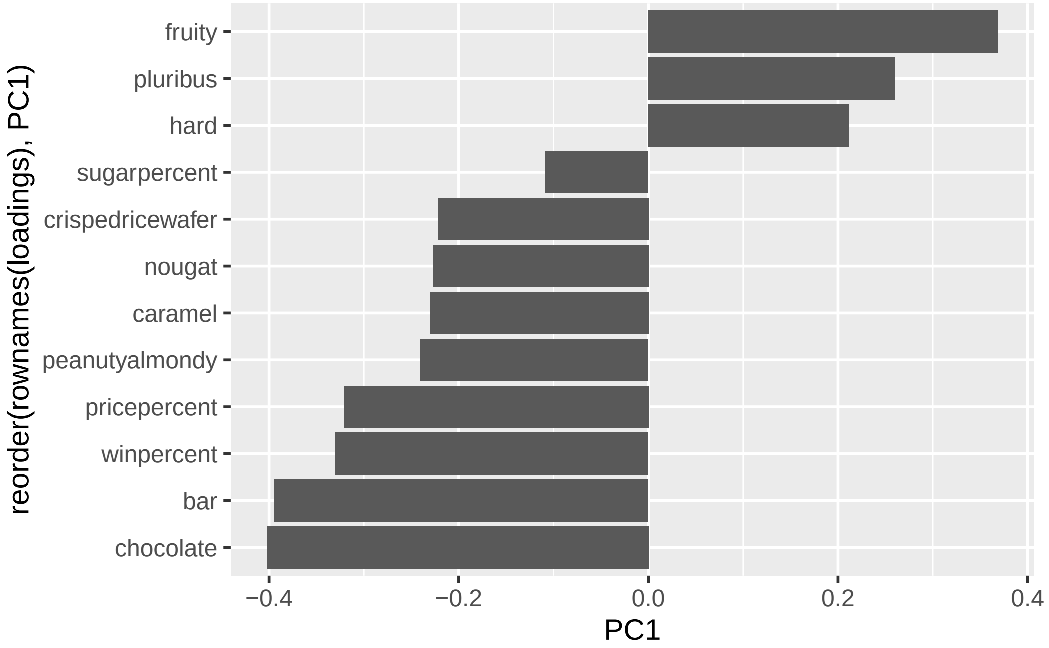
Loadings plot

```
pca$rotation
```

```
                        PC1         PC2         PC3          PC4          PC5
chocolate        -0.4019466  0.21404160  0.01601358 -0.016673032  0.066035846
fruity            0.3683883 -0.18304666 -0.13765612 -0.004479829  0.143535325
caramel          -0.2299709 -0.40349894 -0.13294166 -0.024889542 -0.507301501
peanutyalmondy   -0.2407155  0.22446919  0.18272802  0.466784287  0.399930245
nougat           -0.2268102 -0.47016599  0.33970244  0.299581403 -0.188852418
crispedricewafer -0.2215182  0.09719527 -0.36485542 -0.605594730  0.034652316
hard              0.2111587 -0.43262603 -0.20295368 -0.032249660  0.574557816
bar              -0.3947433 -0.22255618  0.10696092 -0.186914549  0.077794806
pluribus          0.2600041  0.36920922 -0.26813772  0.287246604 -0.392796479
sugarpercent     -0.1083088 -0.23647379 -0.65509692  0.433896248  0.007469103
pricepercent     -0.3207361  0.05883628 -0.33048843  0.063557149  0.043358887
winpercent       -0.3298035  0.21115347 -0.13531766  0.117930997  0.168755073
```

```
                         PC6         PC7         PC8          PC9         PC10
chocolate        -0.09018950 -0.08360642 -0.49084856 -0.151651568  0.107661356
fruity           -0.04266105  0.46147889  0.39805802 -0.001248306  0.362062502
caramel          -0.40346502 -0.44274741  0.26963447  0.019186442  0.229799010
peanutyalmondy   -0.09416259 -0.25710489  0.45771445  0.381068550 -0.145912362
nougat            0.09012643  0.36663902 -0.18793955  0.385278987  0.011323453
crispedricewafer -0.09007640  0.13077042  0.13567736  0.511634999 -0.264810144
hard             -0.12767365 -0.31933477 -0.38881683  0.258154433  0.220779142
bar               0.25307332  0.24192992 -0.02982691  0.091872886 -0.003232321
pluribus          0.03184932  0.04066352 -0.28652547  0.529954405  0.199303452
sugarpercent      0.02737834  0.14721840 -0.04114076 -0.217685759 -0.488103337
pricepercent      0.62908570 -0.14308215  0.16722078 -0.048991557  0.507716043
winpercent       -0.56947283  0.40260385 -0.02936405 -0.124440117  0.358431235
                        PC11        PC12
chocolate         0.10045278  0.69784924
fruity            0.17494902  0.50624242
caramel           0.13515820  0.07548984
peanutyalmondy    0.11244275  0.12972756
nougat           -0.38954473  0.09223698
crispedricewafer -0.22615618  0.11727369
hard              0.01342330 -0.10430092
bar               0.74956878 -0.22010569
pluribus          0.27971527 -0.06169246
sugarpercent      0.05373286  0.04733985
pricepercent     -0.26396582 -0.06698291
winpercent       -0.11251626 -0.37693153
```

```r
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?

The fruity variable is picked up strongly by PC1 in the positive direction. This makes sense because most fruity candies are hard and come plentiful in a bag.