Class 9: Halloween Candy Mini Project

Ruth Barnes: A16747659

2025-02-04

Table of contents

Halloween Mini-Project

Exploratory Analysis of Halloween Candy

Today we will examine data from 538 common Halloween candies. In particular we will use ggplot, dplyr, and PCA to make sense of this multivariate data-set.

Importing candy data

```
candy = read.csv("candy-data.txt", 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?

```r
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

Q. How many chocolate candy are there in the dataset?

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

```
[1] 37
```

What is your favorate 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.

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

```
[1] 81.64291
```

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

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

```
[1] 55.35405
```

Nerds are my favorite candy within the dataset, and have a winpercent value of 55%

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
```

To get a quick overview of a new dataset, there is a useful `skim()` function in the skimr package that can help give us a quick overview of a given dataset. Let’s install this package and try it on our candy data.

```
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?

```r
candy$winpercent
```

```
 [1] 66.97173 67.60294 32.26109 46.11650 52.34146 50.34755 56.91455 23.41782
 [9] 38.01096 34.51768 38.97504 36.01763 24.52499 42.27208 39.46056 43.08892
[17] 39.18550 46.78335 57.11974 34.15896 51.41243 42.17877 55.37545 62.28448
[25] 56.49050 59.23612 28.12744 57.21925 76.76860 41.38956 39.14106 52.91139
[33] 71.46505 66.57458 46.41172 55.06407 73.09956 60.80070 64.35334 47.82975
[41] 54.52645 55.35405 70.73564 66.47068 22.44534 39.44680 46.29660 69.48379
[49] 37.72234 41.26551 37.34852 81.86626 84.18029 73.43499 72.88790 35.29076
[57] 65.71629 29.70369 42.84914 34.72200 63.08514 55.10370 37.88719 45.99583
[65] 76.67378 59.52925 59.86400 52.82595 67.03763 34.57899 33.43755 32.23100
[73] 27.30386 54.86111 48.98265 43.06890 45.73675 49.65350 47.17323 81.64291
[81] 45.46628 39.01190 44.37552 41.90431 49.52411
```

It looks like the `winpercent` column is on a different scale than the others (0-100% rather than 0-1). I will need to scale this dataset before analysis like PCA.

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

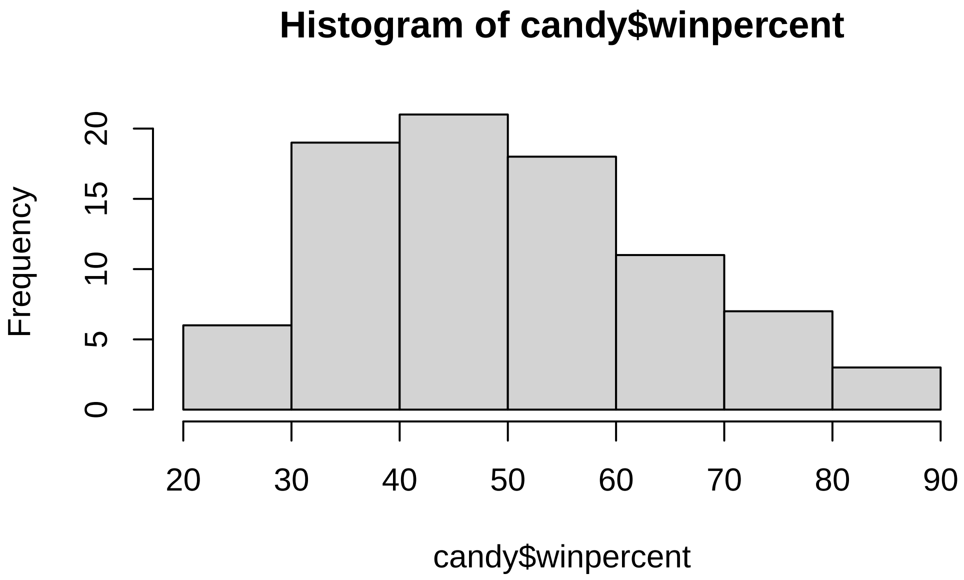
```r
candy$chocolate
```

```
 [1] 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 0 0 1 1 0 1 1 1
[39] 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1
[77] 1 1 0 1 0 0 0 0 1
```

The zero and one represent whether the candy contains chocolate or not for the candy$chocolate column.

Q8. Plot a histogram of winpercent values

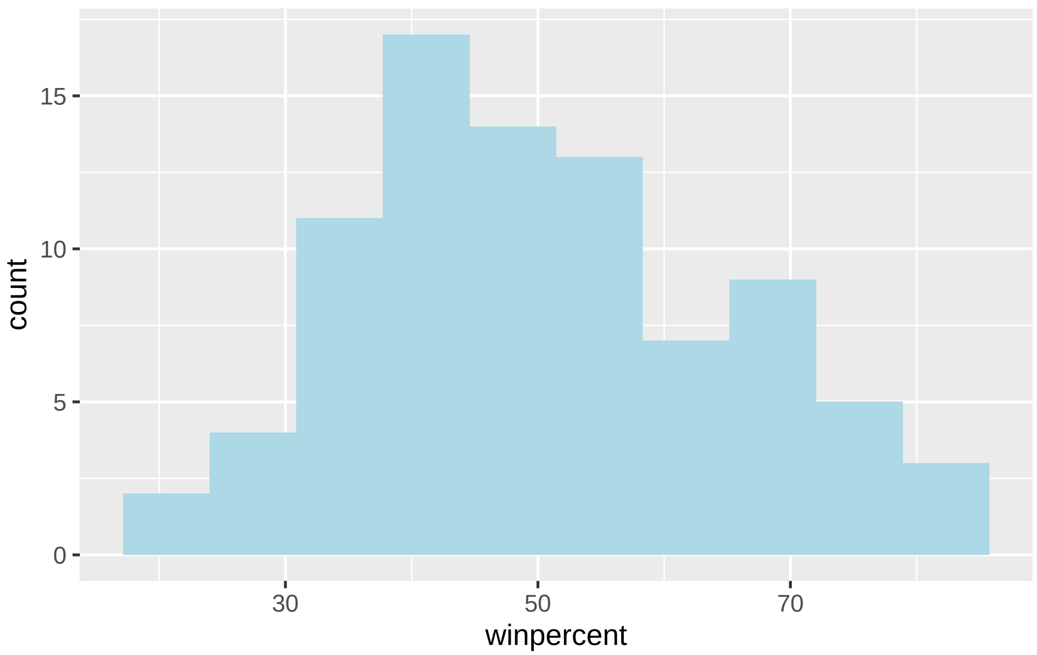
```r
hist(candy$winpercent)
```



or

```r
library(ggplot2)

ggplot(candy) +
  geom_histogram(bins=10, fill = "lightblue") +
  aes(winpercent)
```



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%?

```
summary(candy$winpercent)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  22.45   39.14   47.83   50.32   59.86   84.18 
```

The center of the distribution is above 50%.

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

Step 1: find all "chocolate candy"

```
choc.inds <- candy$chocolate == 1
```

Step 2: find their "winpercent" values

```
choc.win <- candy[choc.inds,]$winpercent
```

Step 3: summarize these values

```
choc.mean <- mean(choc.win)
choc.mean
```

```
[1] 60.92153
```

Step 4: find all "fruity" candy

```
fruit.inds <- candy$fruity == 1
```

Step 5: find their "winpercent" values

```
fruit.win <- candy[fruit.inds,]$winpercent
```

Step 6: summarize these values

```
fruit.mean <- mean(fruit.win)
fruit.mean
```

```
[1] 44.11974
```

Step 7: compare the two summary values

```
choc.mean
```

```
[1] 60.92153
```

```
fruit.mean
```

```
[1] 44.11974
```

Clearly, chocolate has a higher mean winpercent than fruit candy, thus chocolate candy is higher ranked than fruit candy.

Q12. Is this difference statistically significant?

```
t.test(choc.win, fruit.win)
```

```
	Welch Two Sample t-test

data:  choc.win and fruit.win
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 difference is statistically significant.

Overall Candy Rankings

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

Difference between `sort()` and `order()`:

```r
x <- c(10, 1,100)
sort (x)
```

```
[1]   1  10 100
```

```r
# `sort()` is not that useful - it just sorts the values
order(x)
```

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

```r
x[ order(x)]
```

```
[1]   1  10 100
```

The `order()` function tells us how to arrange the elements of the input to make them sorted - i.e. how to order them. We can determine the order of winpercent to make them sorted and use that order to arrange the whole dataset.

```
sort(candy$winpercent)
```

```
 [1] 22.44534 23.41782 24.52499 27.30386 28.12744 29.70369 32.23100 32.26109
 [9] 33.43755 34.15896 34.51768 34.57899 34.72200 35.29076 36.01763 37.34852
[17] 37.72234 37.88719 38.01096 38.97504 39.01190 39.14106 39.18550 39.44680
[25] 39.46056 41.26551 41.38956 41.90431 42.17877 42.27208 42.84914 43.06890
[33] 43.08892 44.37552 45.46628 45.73675 45.99583 46.11650 46.29660 46.41172
[41] 46.78335 47.17323 47.82975 48.98265 49.52411 49.65350 50.34755 51.41243
[49] 52.34146 52.82595 52.91139 54.52645 54.86111 55.06407 55.10370 55.35405
[57] 55.37545 56.49050 56.91455 57.11974 57.21925 59.23612 59.52925 59.86400
[65] 60.80070 62.28448 63.08514 64.35334 65.71629 66.47068 66.57458 66.97173
[73] 67.03763 67.60294 69.48379 70.73564 71.46505 72.88790 73.09956 73.43499
[81] 76.67378 76.76860 81.64291 81.86626 84.18029
```

```
ord.inds <- order(candy$winpercent)
ord.inds
```

```
 [1] 45  8 13 73 27 58 72  3 71 20 10 70 60 56 12 51 49 63  9 11 82 31 17 46 15
[26] 50 30 84 22 14 59 76 16 83 81 77 64  4 47 35 18 79 40 75 85 78  6 21  5 68
[51] 32 41 74 36 62 42 23 25  7 19 28 26 66 67 38 24 61 39 57 44 34  1 69  2 48
[76] 43 33 55 37 54 65 29 80 52 53
```

45th candy has the lowest winpercent, then the 8th, 13th, etc.

```
head(candy[ord.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, and Jawbusters are the five least liked candy types in this set.

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

```
tail(candy[ord.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
```

or

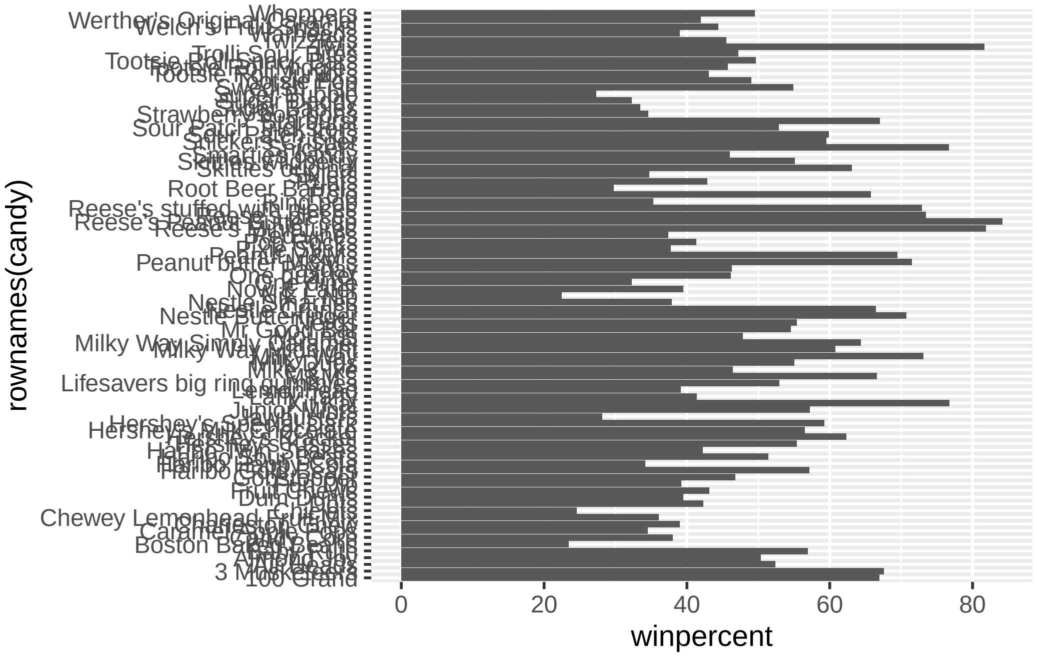
```
ord.inds <- order(candy$winpercent, decreasing = T)
head(candy[ord.inds,])
```

```
                          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
Reese's pieces                    1      0       0              1      0
                          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
Reese's pieces                           0    0   0        1        0.406
                          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 pieces                   0.651   73.43499
```

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

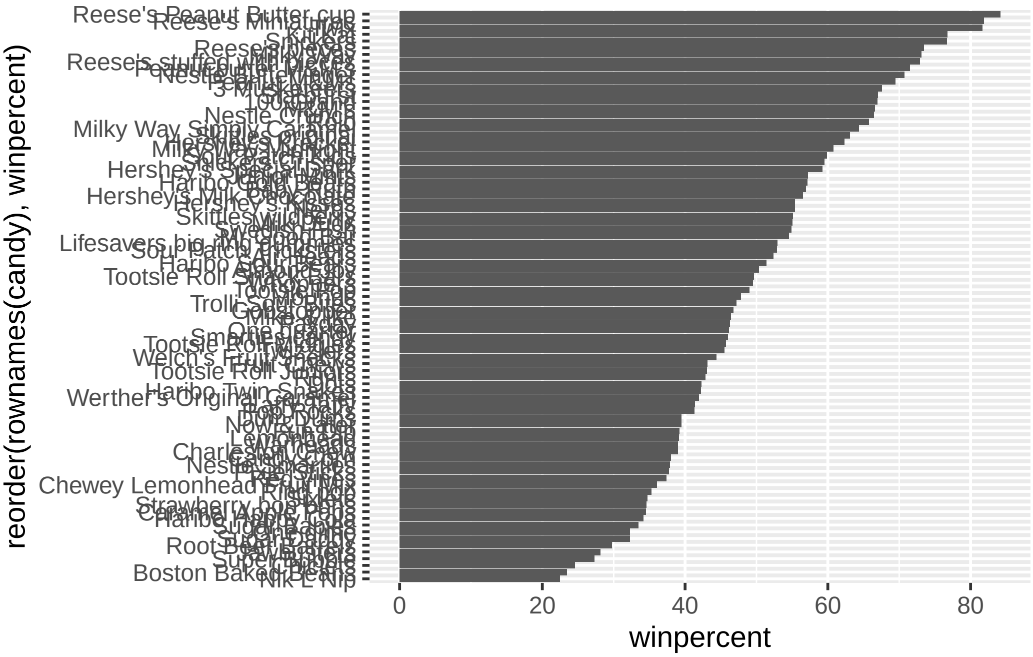
Q15. Make a first barplot of candy ranking based on winpercent values.

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



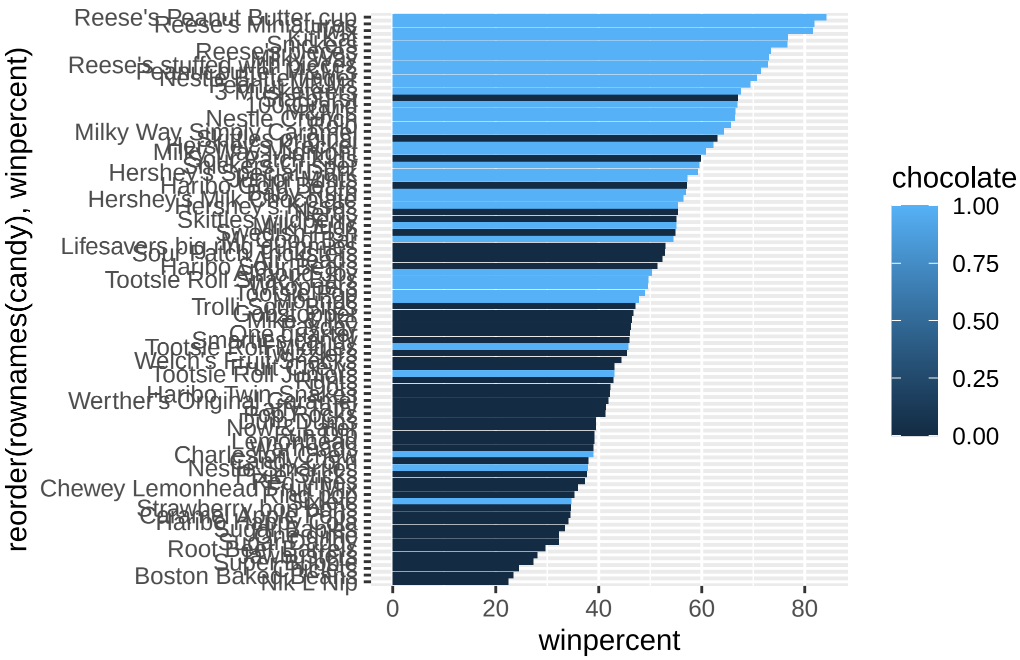
Q16. This is quite ugly, use the reorder() function to get the bars sorted by winpercent?

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



Time to add some useful color

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



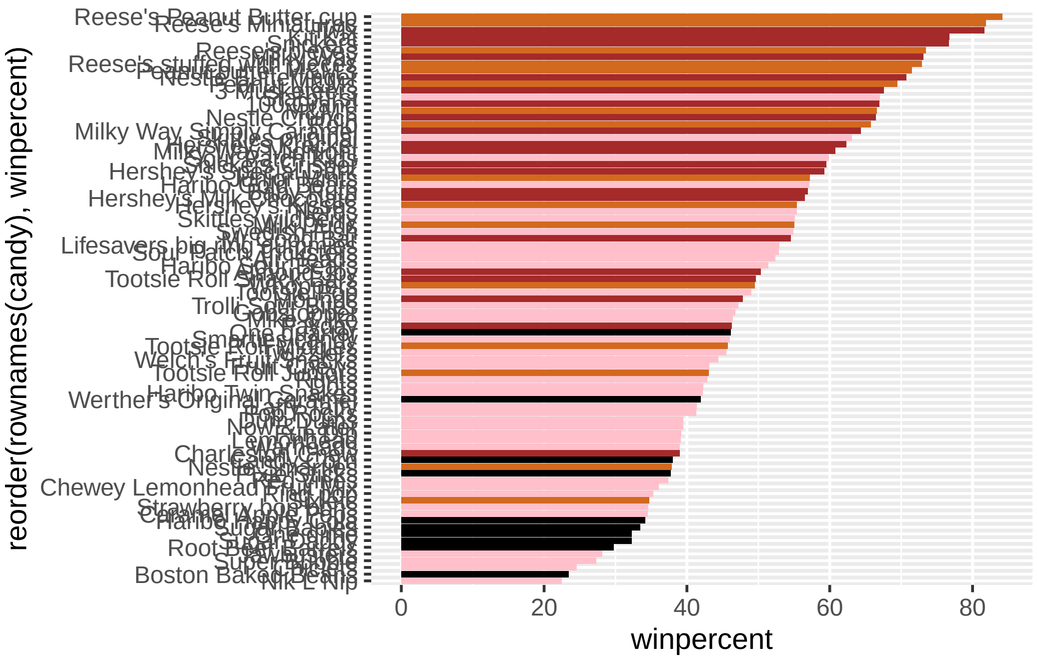
We need to make our own separate color vector where we can spell out exactly what candy is colored a particular color.

```r
mycols <- rep("black", nrow(candy))
mycols[candy$chocolate == 1] <- "chocolate"
mycols[candy$fruity == 1] <- "pink"
mycols[candy$bar == 1] <- "brown"
mycols
```

```
 [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"
```

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



Q17. What is the worst ranked chocolate candy?

Sixlets is the worst ranked chocolate candy

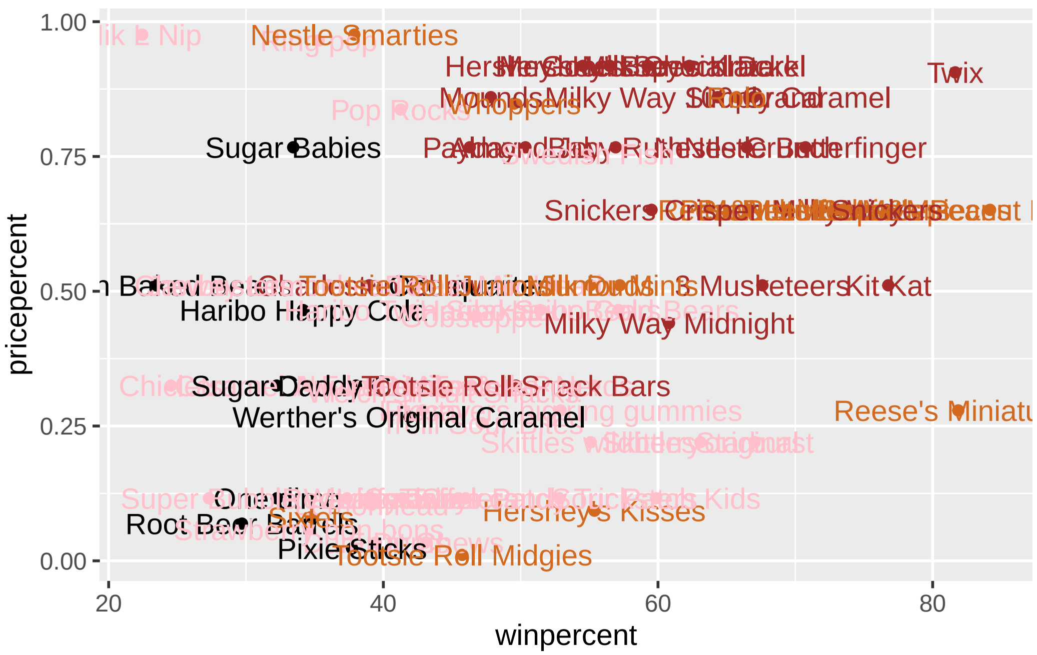
Q18. What is the best ranked fruity candy?

Starburst is the best ranked fruity candy.

Taking a look at pricepercent

Make a plot of winpercent (x-axis) vs pricepercent (y-axis)

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

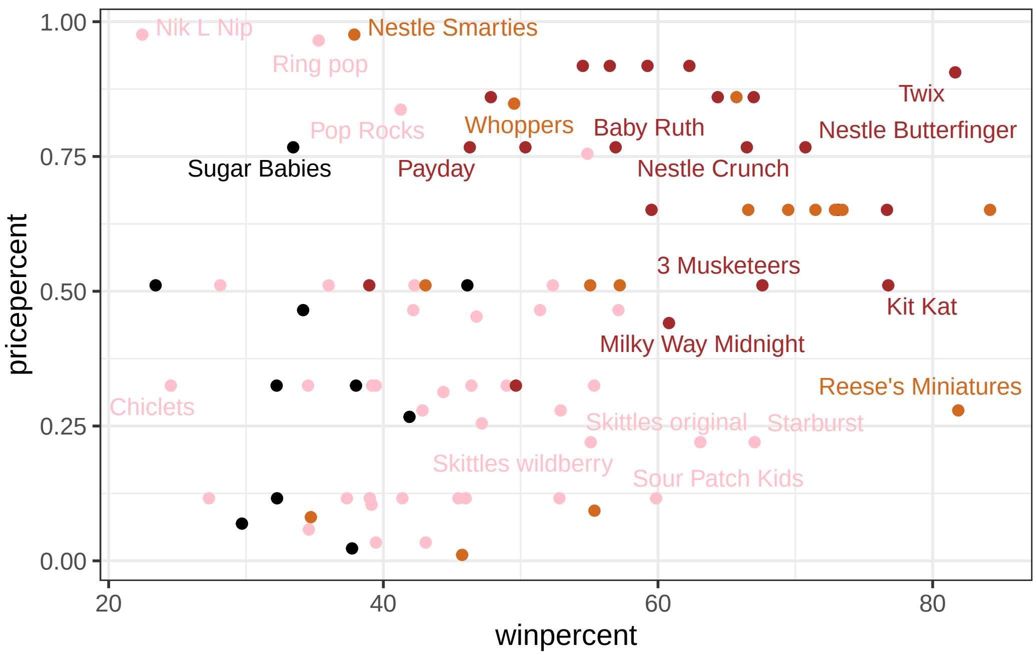


To avoid the overplotting of the text labels, we can use the add on package **ggrepelt**

```
library(ggrepel)

# How about a plot of price vs win
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=mycols) +
  geom_text_repel(col=mycols, max.overlaps = 5, size = 3.3) +
  theme_bw()
```

```
Warning: ggrepel: 65 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?

```
ord <- order(candy$winpercent, decreasing = T)
ord2<- order(candy$pricepercent, decreasing = F)
head(candy[ord, c(11,12)], n=5 )
```

```
                            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
```

```
head(candy[ord2, 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
```

Reese's Miniatures is the highest ranked in terms of winpercent for the least money.

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 = T)
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 Nop, Nestle Smarties, Ring pop, Hershey's Krackel, and Hershey's Milk Chocolate are the top 5 most expensive candy types in the dataset, with Nik L Nip being the least popular.

Exploring the correlation structure

Now that we have explored the dataset a little, we will see how the variables interact with one another.

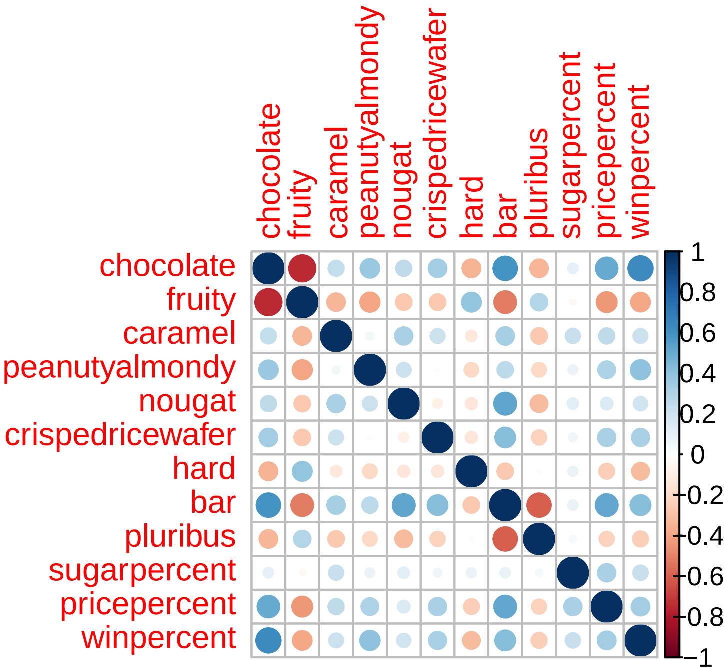
First we will use correlation and view the results with the **corrplot** packag to plot a correlation matrix.

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

```
library(corrplot)
```

```
corrplot 0.95 loaded
```

```
corrplot(cij)
```



Q22. Examining this plot what two variables are anti-correlated (i.e. have minus values)?

Fruity and chocolate are anti-correlated.

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

Chocolate and winpercent are the most positively correlated.

Principal Component Analysis

Let's apply PCA using the `prcomp()` function to our candy dataset remembering to set the **scale=TRUE** argument.

```
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
```

```
attributes(pca)
```

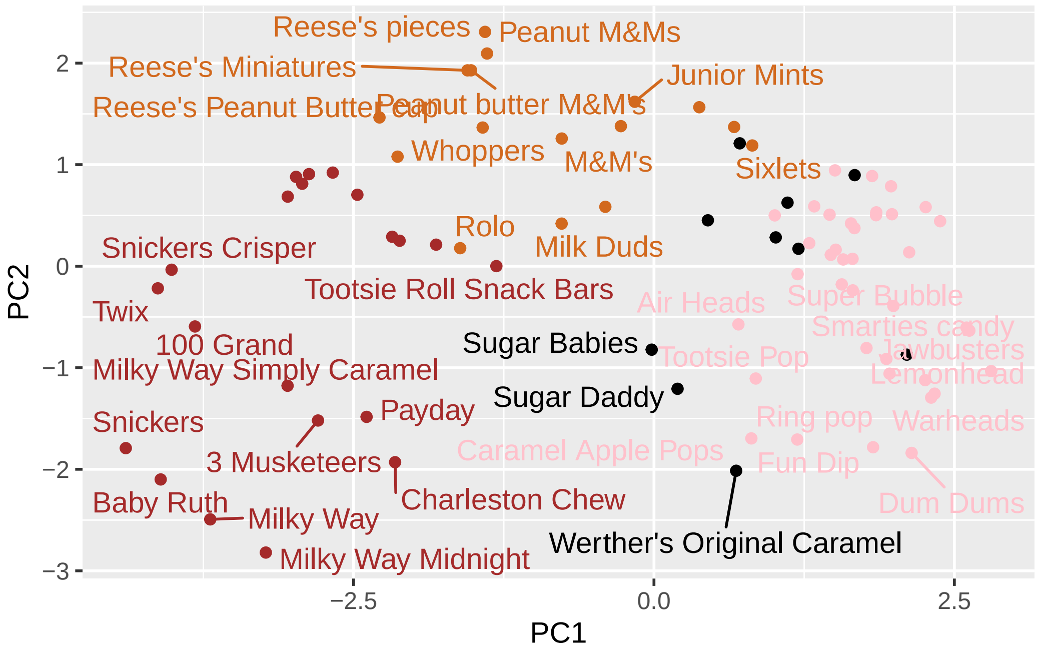
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

Let's plot our main results as our PCA "score plot":

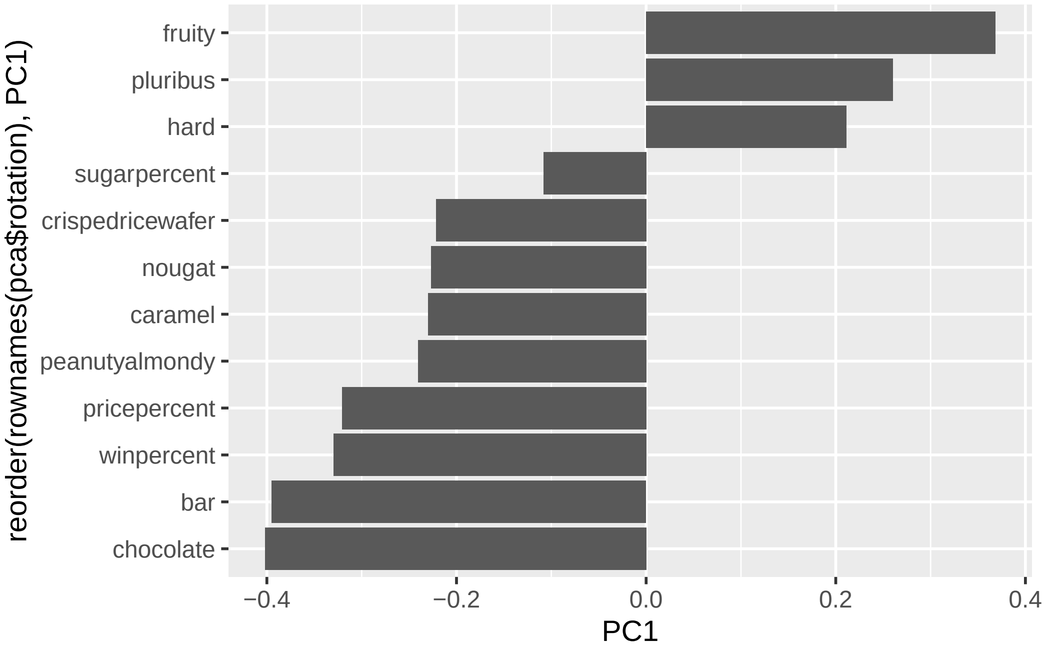
```
ggplot(pca$x) +
  aes(PC1, PC2, label=rownames(pca$x)) +
  geom_point(col=mycols) +
  geom_text_repel(col=mycols)
```

```
Warning: ggrepel: 48 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



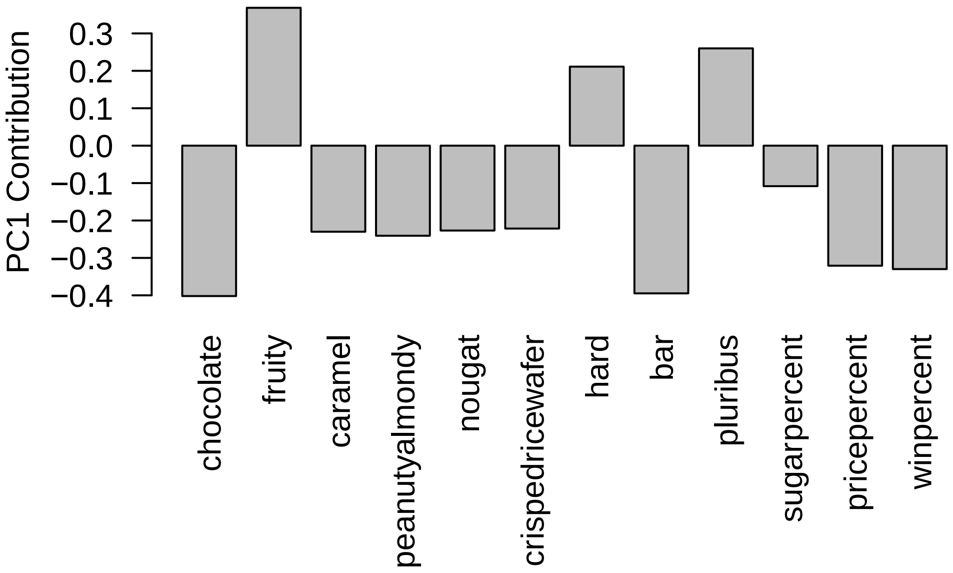
Finally let's look at how the original variables contribute to the PCs, start with PC1:

```
ggplot(pca$rotation) +
  aes(PC1, reorder(rownames(pca$rotation), PC1)) +
  geom_col()
```



or

```
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?

Fruity, hard, and pluribus candies are pick up strongly by PC1 in the positive direction. Yes, these do make sense to me.