Class 9: Candy

Emily Ignatoff (A16732102)

Table of contents

Loading Data and preliminary observations

Today we will examine data from 538 on common Halloween candy, particularly using ggplot, dplyr, and PCA to understand this dataset.

```
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 in this dataset?

```
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

How many chocolate candies are there?

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

```
[1] 37
```

We can look at `winpercent` as a metric for how popular a candy is compared to others in the dataset.

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

My favorite candy in the dataset is Milky Way

```
candy["Milky Way", ]$winpercent
```

```
[1] 73.09956
```

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"?

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

```
[1] 49.6535
```

We can get a good overview of our data using the `skimr` package:

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

```
Warning: package 'skimr' was built under R version 4.3.3
```

```r
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	

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

Winpercent has much higher values from 22.45-84.18 whereas other values are only fractions in a range of 0-1. Therefore, it will need to be scaled back for use in a PCA.

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

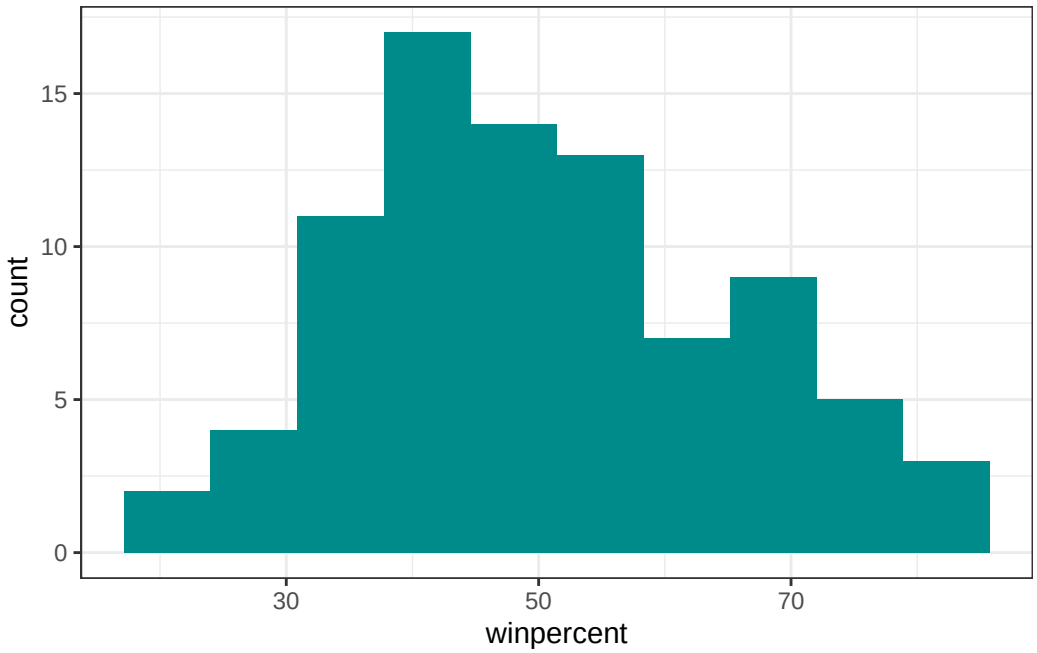
Zero means that a candy is not chocolate, one means that a candy is chocolate.

Q8. Plot a histogram of winpercent values

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.3.3
```

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



Q9. Is the distribution of winpercent values symmetrical?

No

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 
```

Median (center) is below 50%, it is 47.83%

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

- Step 1: find all "chocolate" candy
- Step 2: find their "winpercent" values
- Step 3: summarize these values
- Step 4: repeat for "fruity" candy
- Step 5: compare the two summary values

```
#step 1
choc.inds <- candy$chocolate == 1
#step 2
choc.win <- candy[choc.inds,]$winpercent
#step 3
choc.mean <- mean(choc.win)
choc.mean
```

```
[1] 60.92153
```

Same steps for fruity candy:

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

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

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

```
[1] 44.11974
```

Chocolate has a higher mean winpercent than fruity candy.

Q12. Is this difference statistically significant?

We can use a T-test to determine this:

```
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
```

The results of this t-test indicate that the difference in means IS statistically significant.

Overall Candy Rankings

```
#Not very useful, just sorts the candy:
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
```

```
x <- c(10, 1, 100)
x[order(x)]
```

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

The `order()` function can help us to arrange 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

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

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

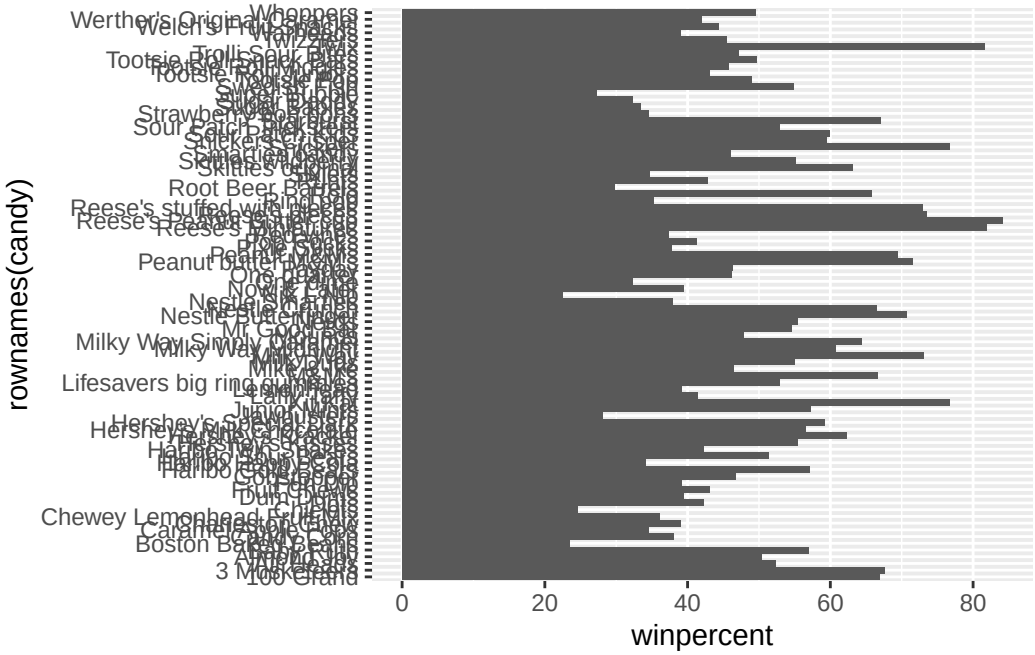
Q14. What are the top 5 all time favorite candy types out of this set?

```
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
```

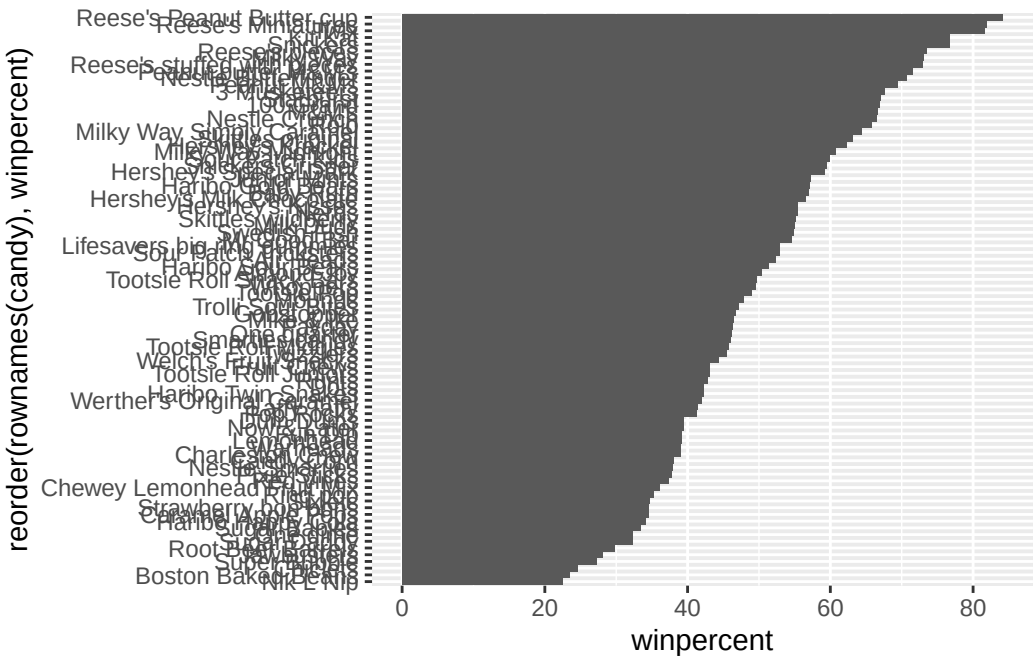
Q15. Make a first barplot of candy ranking based on winpercent values.

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



Let us improve this plot:

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

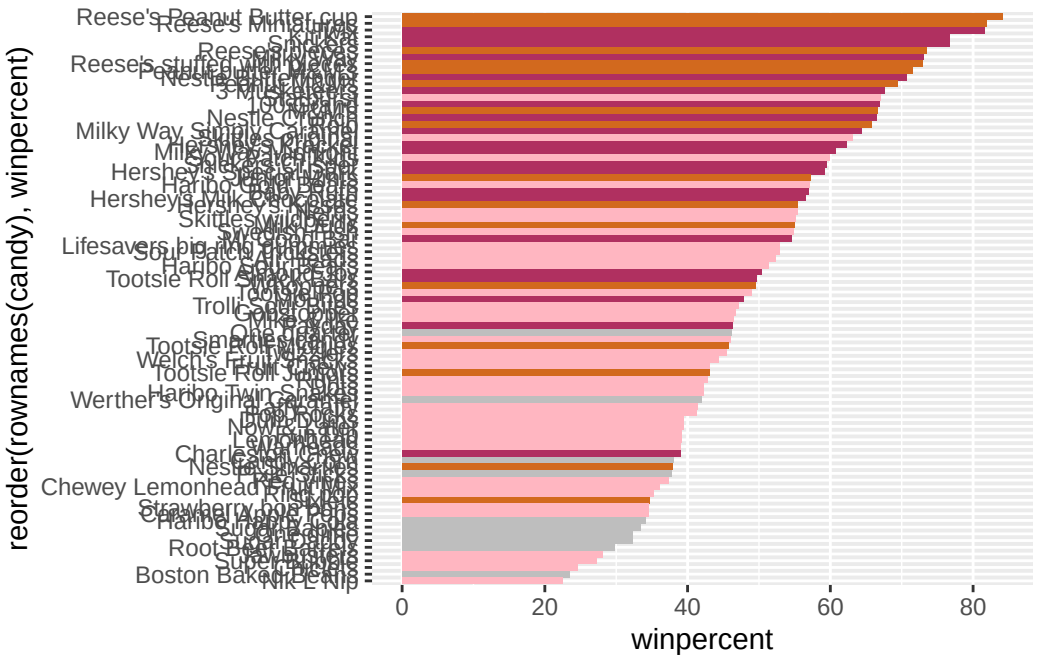


Let's add some useful color! We need to make our own color vector to do this:

```
mycols <- rep("gray", nrow(candy))
mycols[candy$chocolate == 1] <- "chocolate"
mycols[candy$fruity == 1] <- "lightpink"
mycols[candy$bar ==1] <- "maroon"
mycols
```

```
 [1] "maroon"    "maroon"    "gray"      "gray"      "lightpink" "maroon"
 [7] "maroon"    "gray"      "gray"      "lightpink" "maroon"    "lightpink"
[13] "lightpink" "lightpink" "lightpink" "lightpink" "lightpink" "lightpink"
[19] "lightpink" "gray"      "lightpink" "lightpink" "chocolate" "maroon"
[25] "maroon"    "maroon"    "lightpink" "chocolate" "maroon"    "lightpink"
[31] "lightpink" "lightpink" "chocolate" "chocolate" "lightpink" "chocolate"
[37] "maroon"    "maroon"    "maroon"    "maroon"    "maroon"    "lightpink"
[43] "maroon"    "maroon"    "lightpink" "lightpink" "maroon"    "chocolate"
[49] "gray"      "lightpink" "lightpink" "chocolate" "chocolate" "chocolate"
[55] "chocolate" "lightpink" "chocolate" "gray"      "lightpink" "chocolate"
[61] "lightpink" "lightpink" "chocolate" "lightpink" "maroon"    "maroon"
[67] "lightpink" "lightpink" "lightpink" "lightpink" "gray"      "gray"
[73] "lightpink" "lightpink" "lightpink" "chocolate" "chocolate" "maroon"
[79] "lightpink" "maroon"    "lightpink" "lightpink" "lightpink" "gray"
[85] "chocolate"
```

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



Q17. What is the worst ranked chocolate candy?

sixlets

Q18. What is the best ranked fruity candy?

starbusrt

#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() +
  theme_minimal()
```

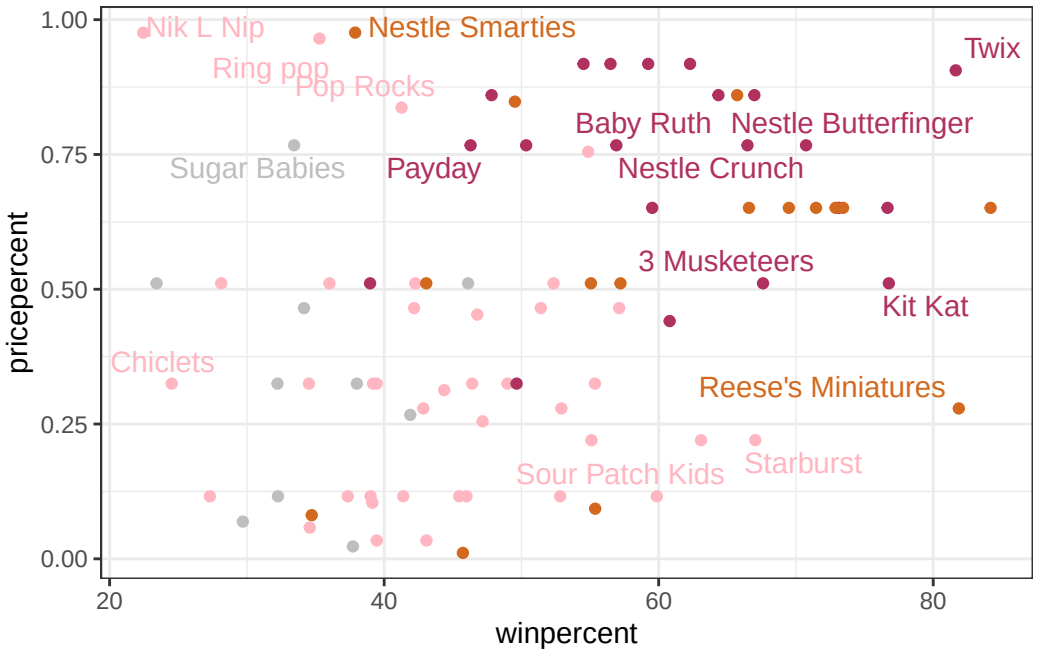
Let's use a different package to avoid this overplotting of text. This package is called `ggrepel`

```
library(ggrepel)
```

```
Warning: package 'ggrepel' was built under R version 4.3.3
```

```
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=mycols) +
  geom_text_repel(col=mycols, max.overlaps = 6) +
  theme_bw()
```

```
Warning: ggrepel: 69 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?

Reese's miniatures are most popular for the lowest price

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

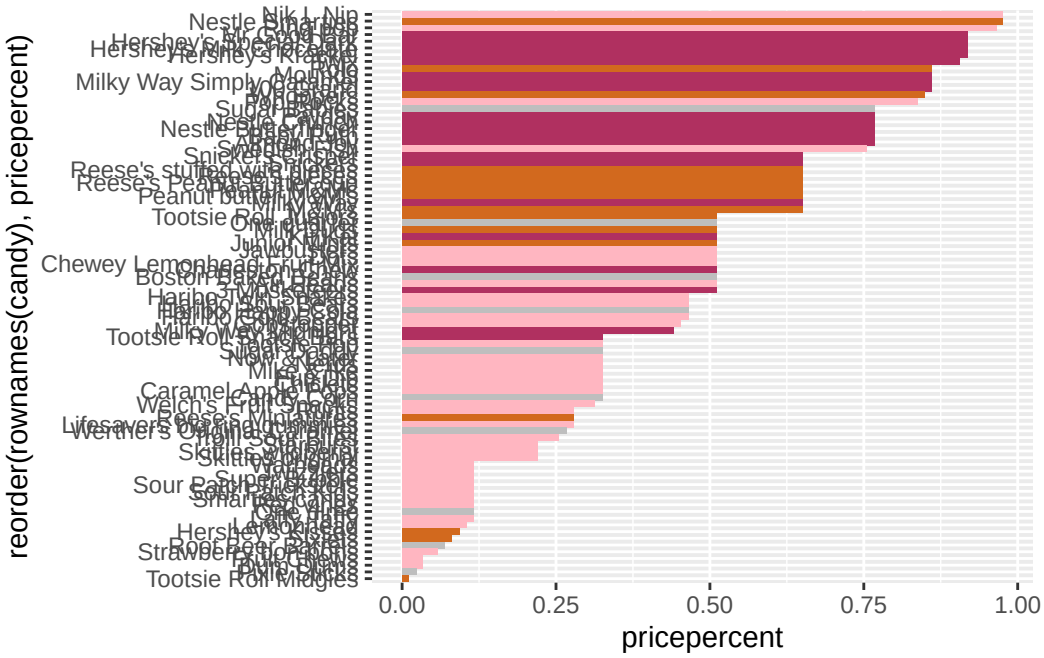
The most expensive candies are: Nik L Nip, Ring Pops, Nestle Smarties, Hershey's Krackel, and Hershey's Milk Chocolate. Nik L Nips is the least popular

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

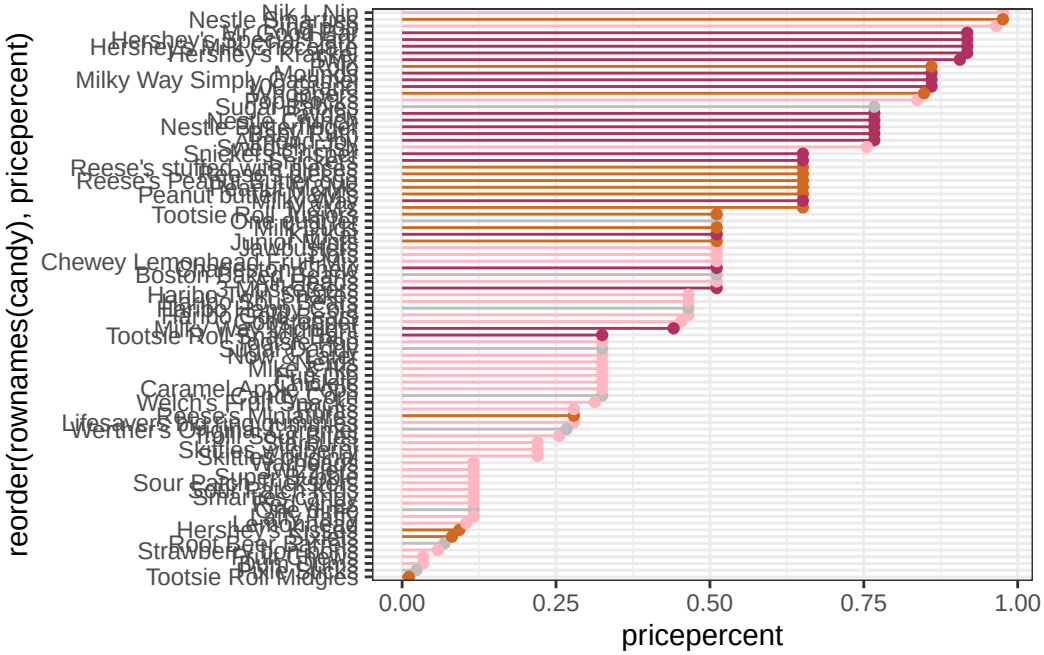
```
                      pricepercent winpercent
Nik L Nip                    0.976   22.44534
Nestle Smarties              0.976   37.88719
Ring pop                     0.965   35.29076
Hershey's Krackel            0.918   62.28448
Hershey's Milk Chocolate     0.918   56.49050
```

Optional Bar charts

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




```
ggplot(candy) +
  aes(pricepercent, reorder(rownames(candy), pricepercent)) +
  geom_segment(aes(yend = reorder(rownames(candy), pricepercent),
                   xend = 0), col=mycols) +
    geom_point(col=mycols) +
  theme_bw()
```



Exploring Correlations

We will first use correlation and view the results in a correlation matrix using the `corrplot` package

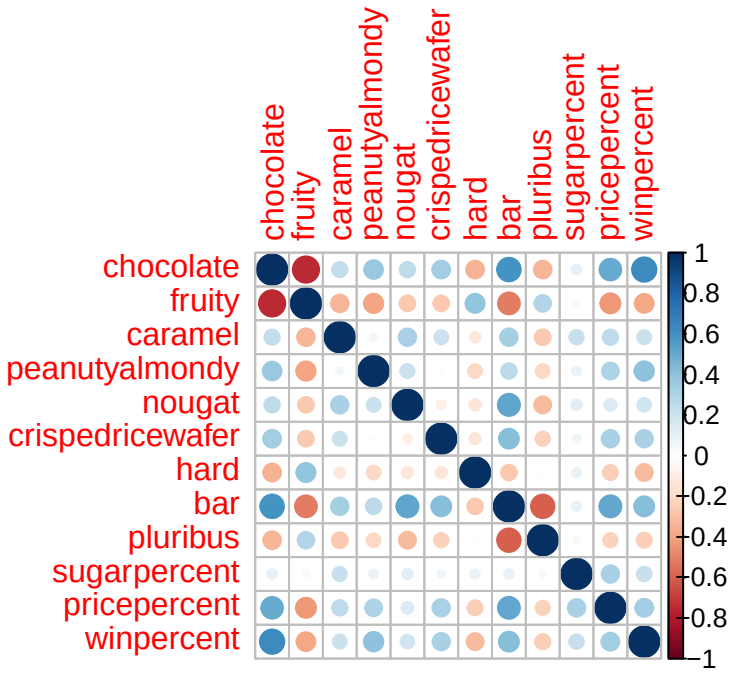
```
cij <- cor(candy)
```

```
library(corrplot)
```

```
Warning: package 'corrplot' was built under R version 4.3.3
```

```
corrplot 0.95 loaded
```

```
corrplot(cij)
```



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

Fruit and chocolate are very anti-correlated (chocolate candies do not tend to have fruit in them)

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

Chocolate and winpercent are most positively correlated, showing that chocolate is very popular.

Prinicpal Component Analysis

Let's run a PCA on this data where scaling is set to "TRUE"

```
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
```

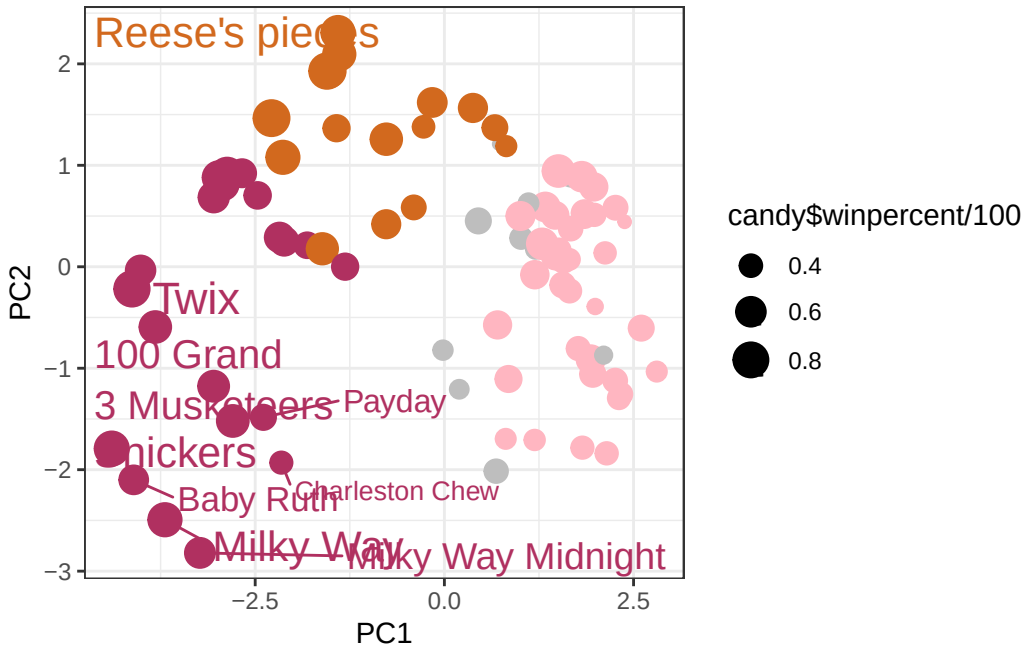
```
attributes(pca)
```

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

$class
[1] "prcomp"
```

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

```
p <- ggplot(pca$x) +
  aes(PC1, PC2, label=rownames(pca$x), size=candy$winpercent/100) +
  geom_point(col=mycols) + theme_bw()


p + geom_text_repel(col=mycols, max.overlaps = 6)
```

```
Warning: ggrepel: 75 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Let's make this using the `plotly` library:

```
#library(plotly)
#ggplotly(p)
```

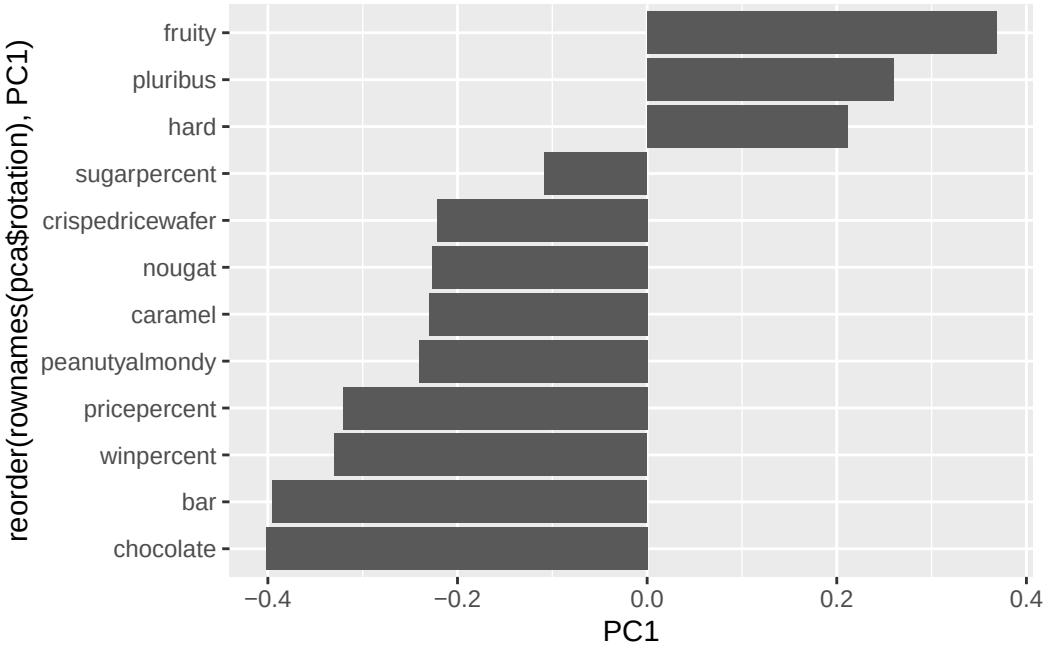
Finally, let's look at how the original variables contribute to the PCs starting with PC1

```
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
```

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



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

In the positive direction, fruity, hard, and pluribus candies push PC1. This makes sense since those variables are all correlated with each other in our correlation plot. Fruity candies are likely to be hard candies in a multi-pack!