Class 9: Halloween Candy Mini Project

Snehita Vallumchetla (A16853399)

Table of contents

We will examine data from 538 on common Halloween candy. In particular we will use ggplot, dplyr, and PCA to make sense of this multivariant dataset.

1. Importing Candy Data:

```
candy_file <- 'candy-data.csv'

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 data set?

There are 85 different candy types in this data set.

```
# The number of candy types is given by the number of rows which can be determined using the
nrow(candy)
```

```
[1] 85
```

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

There are 38 fruity candy types in this dataset.

```
sum(candy$fruit)
```

```
[1] 38
```

How many chocolate candy are in the dataset

There are 37 chocolate candy in the data ste

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

```
[1] 37
```

2. What is your favorite candy:

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

My favorite candy in the data set is sour patch kids and its winpercent value is 59.864.

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

```
[1] 59.864
```

Q4. What is the winpercent value for “Kit Kat”?

The winpercent value for kit kat is 76.7686.

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

```
[1] 76.7686
```

Q5. What is the winpercent value for “Tootsie Roll Snack Bars”?

The winpercent value for the tootsie roll snack bars is 49.6535.

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

```
[1] 49.6535
```

The skim package is useful for “skimming” through a dataset!

```
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?

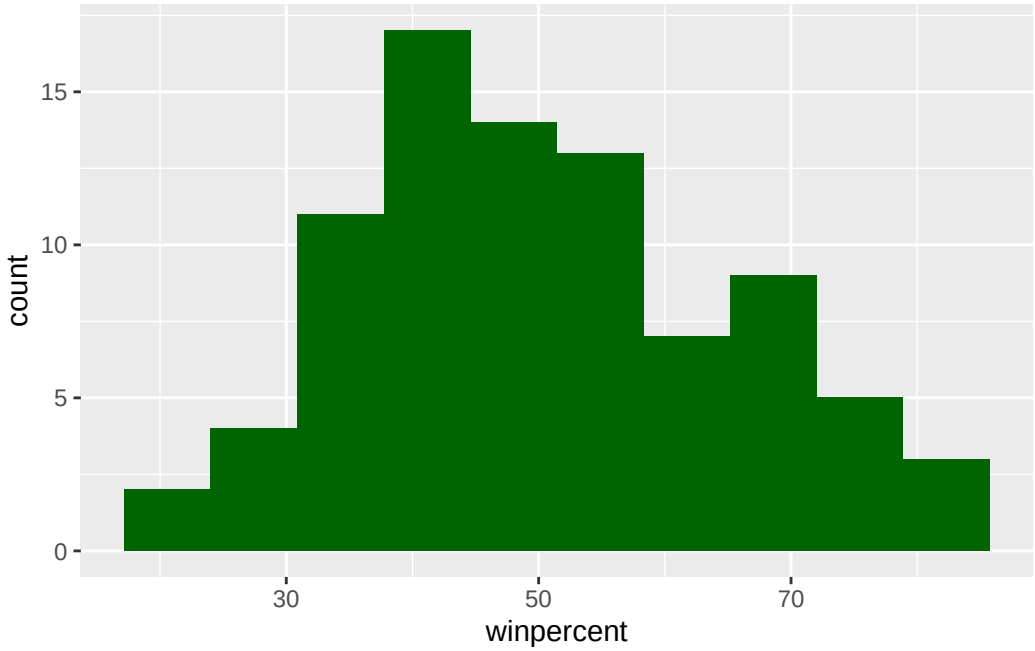
The `winpercent` column is on a different scale that the others (0-100% rather than 0-1). I will need to scale this dataset before analysis like PCA.

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

The zero represents that the candy of choice does not contain chocolate while the 1 represents that the candy of choice contains chocolate

Q8. Plot a histogram of winpercent values

```
library(ggplot2)

ggplot(candy) +
  aes(x = winpercent) +
  geom_histogram(bins = 10, fill = 'darkgreen')
```



Q9. Is the distribution of winpercent values symmetrical?

The distribution of the winpercent values is slightly skewed to the left.

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

The center of the distribution (the median) is around 47.83 percent, which is 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 
```

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 the 'winpercent' values

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

- step 3: summarize these values

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

- step 4: find all 'fruity' candy

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

- step 5: find the 'winpercent' values

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

- step 6: summarize these values

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

- step 7: compare

Clearly chocolate has a higher mean winpercent. The average winpercent for chocolate candy is 61 which is greater than the average winpercent for fruity candy being 44.

Q12. Is this difference statistically significant?

since the p-value is less than 0.05, the difference between the winpercent of chocolate candy and fruity candy is 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
```

3. Overall Candy Rankings:

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

The 5 least liked candy types are: 'Nik L Nip', 'Boston Baked Beans', 'Chiclets', 'Super Bubble', 'Jawbusters.'

```
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
```

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 arrage the whole dataset.

```
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?

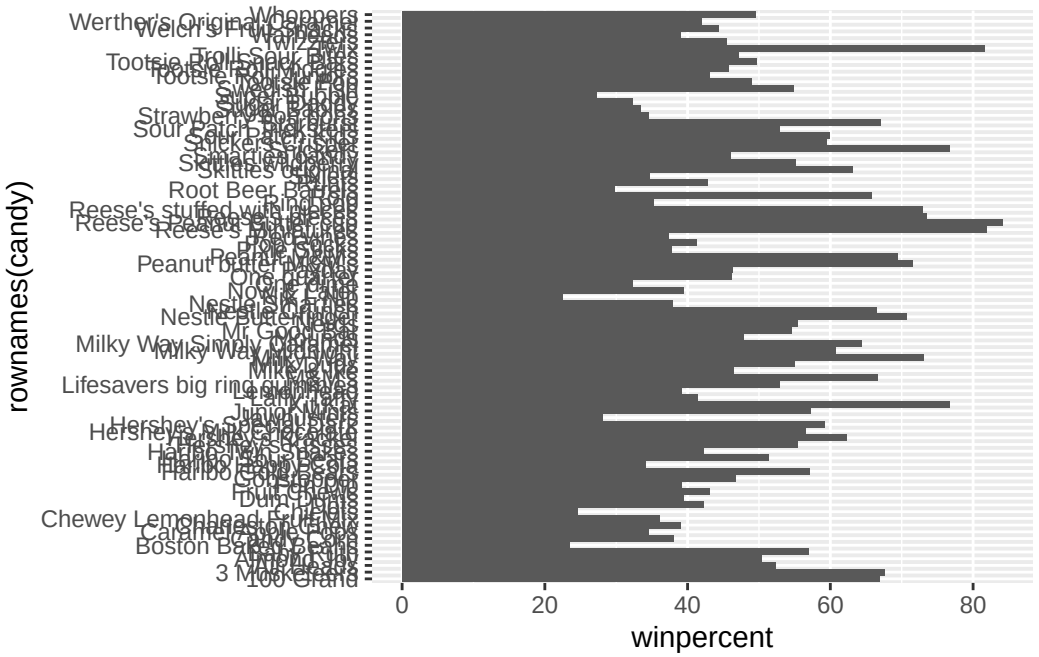
The tope 5 all time favorite candy types of this set are: Reese's pieces, Snickers, Kit Kat, Twix, Reese's Miniatures.

```
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
```

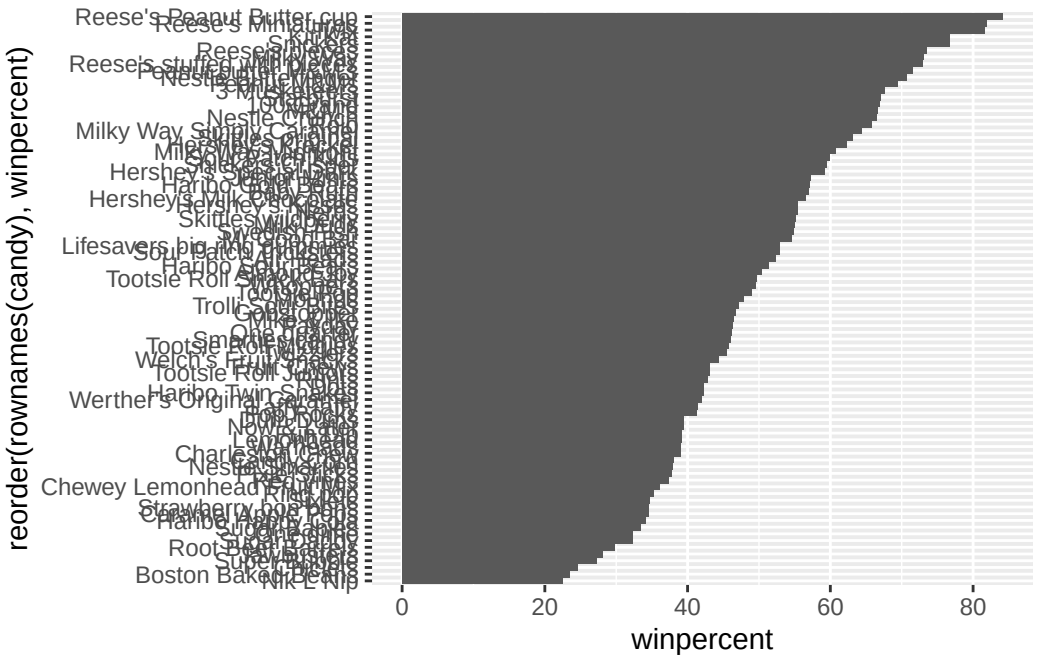
Q15. Make a first barplot of candy ranking based on winpercent values.

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



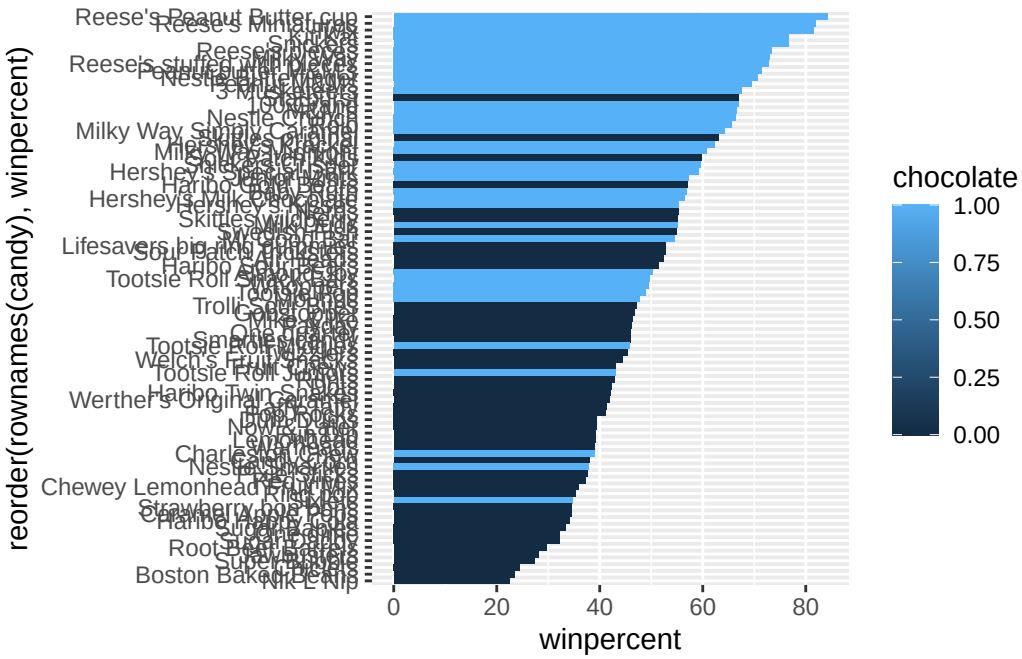
Let's reorder this, shows highest at the top of the plot and lowest at the bottom of the plot.

```
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()
```



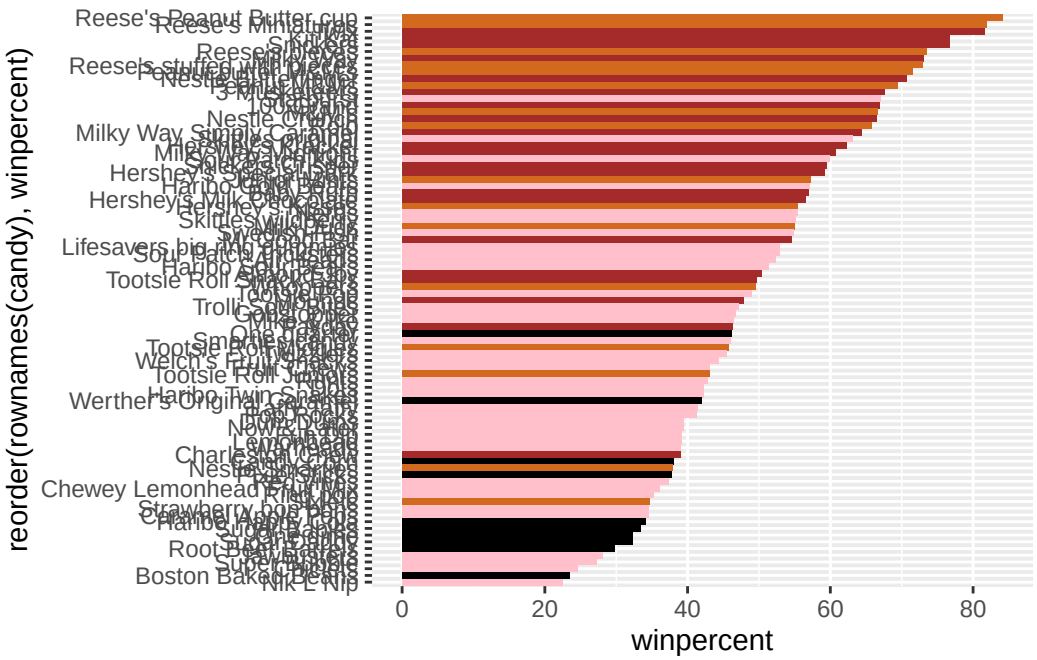
We need to make our own color vector where we can spell out exactly what candy is colored a particular color.

```
mycols <- rep("black", nrow(candy))

mycols[candy$chocolate == 1] <- 'chocolate'
mycols[candy$bar == 1] <- 'brown'
mycols[candy$fruity == 1] <- 'pink'

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)) +
  geom_col(fill = mycols)
```



Now, for the first time, using this plot we can answer questions like:

Q17. What is the worst ranked chocolate candy?

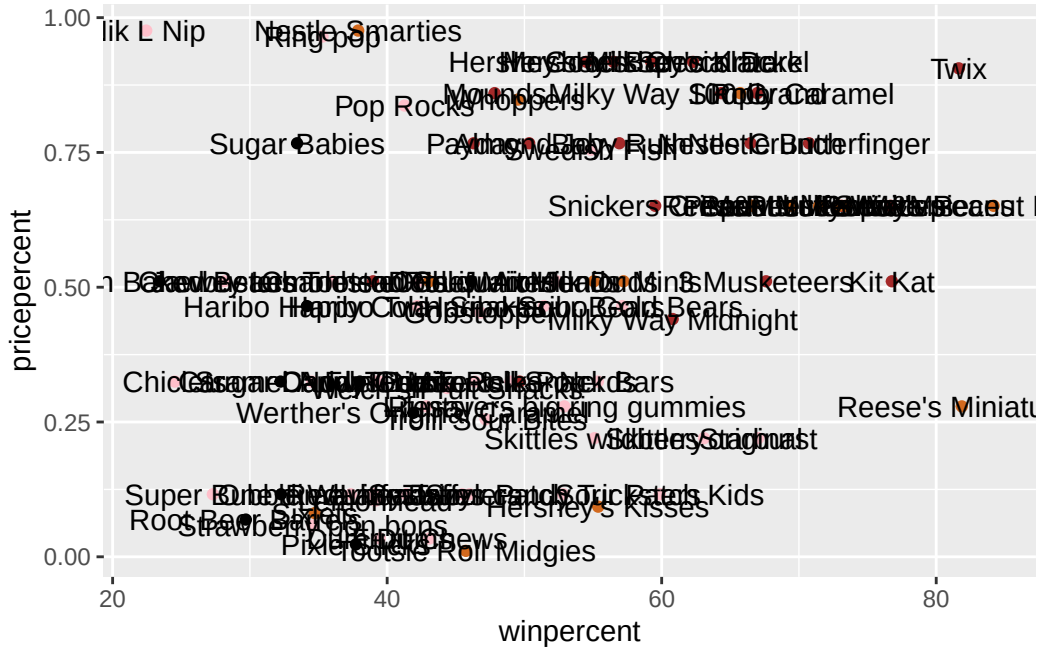
The worst ranked chocolate candy are Sixlets.

Q18. What is the best ranked fruity candy?

The best ranked fruity candy are Starburts.

4. Taking a look at pricepercent:

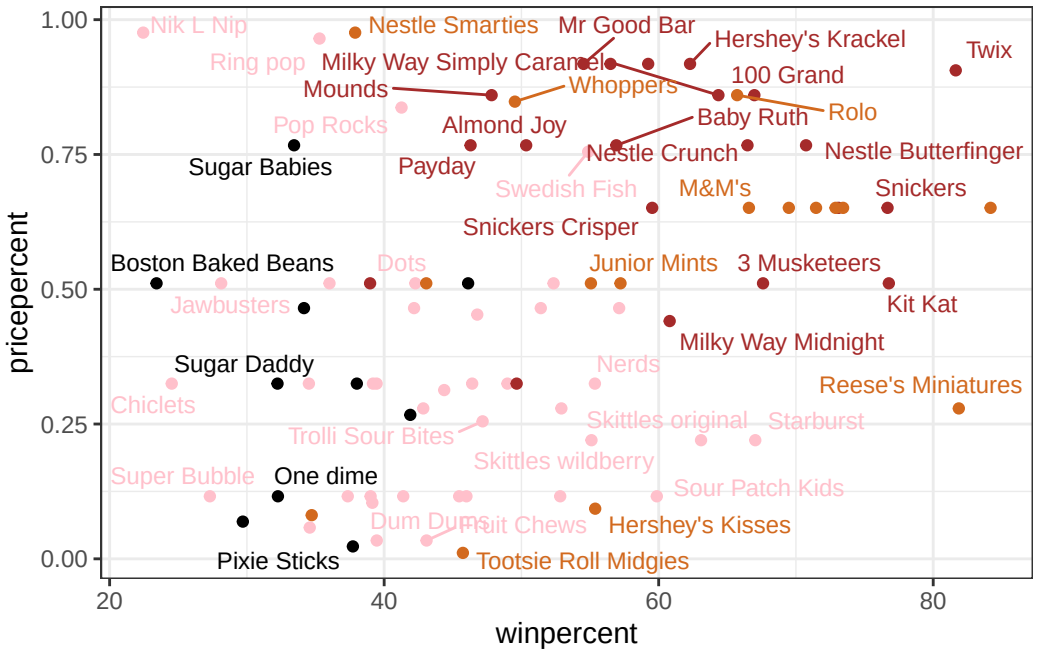
```
ggplot(candy) + aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = mycols) +
  geom_text()
```



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

```
library(ggrepel)

ggplot(candy) + aes(winpercent, pricepercent, label = rownames(candy)) +
  geom_point(col = mycols) +
  geom_text_repel(col = mycols, size = 3.3, max.overlaps = 10) + theme_bw()
```

```
Warning: ggrepel: 40 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?

The Reese's Miniatures are ranked the highest in terms of winpercent while they have the lowest pricepoint!

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

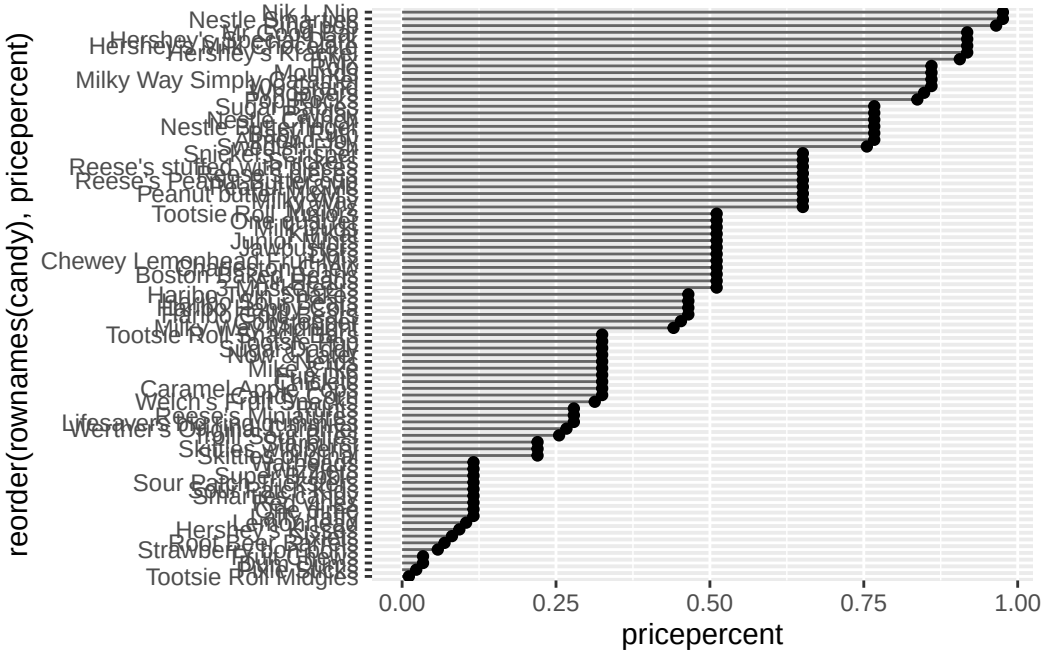
```
ord <- order(candy$pricepercent, decreasing = TRUE)
head( candy[ord,c(11,12)], n=5 )
```

```
                      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
```

The top 5 most expensive candy types in the dataset are Nik L Nips, Nestle Smarties, Ring Pops, Hershey Krackel, Hershey's Milk Chocolate. The least popular of these is the Nik L Nip.

Q21.Make a barplot again with geom_col() this time using pricepercent and then improve this step by step, first ordering the x-axis by value and finally making a so called "dot chat" or "lollipop" chart by swapping geom_col() for geom_point() + geom_segment().

```
ggplot(candy) +
  aes(pricepercent, reorder(rownames(candy), pricepercent)) +
  geom_segment(aes(yend = reorder(rownames(candy), pricepercent),
                   xend = 0), col="gray40") + geom_point()
```



5. 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** package to plot a correlation matrix.

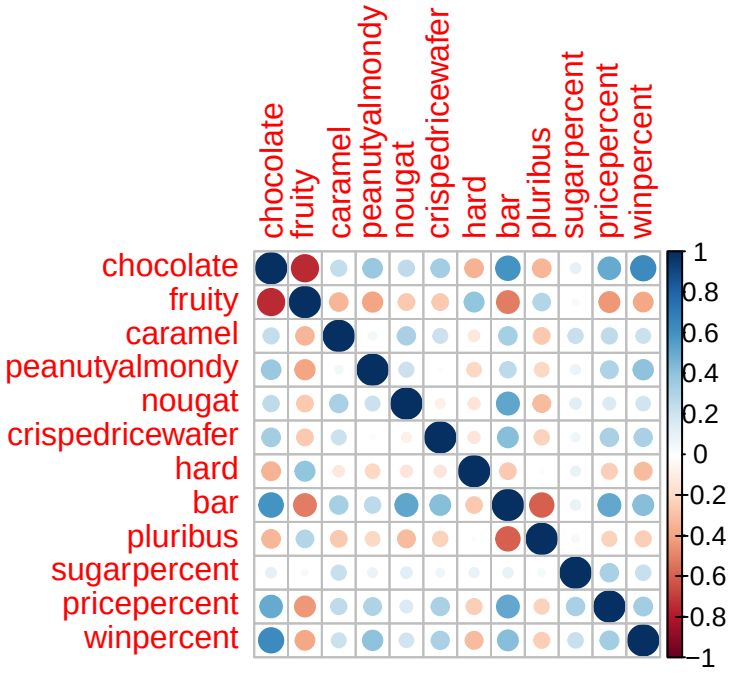
```
library(corrplot)
```

```
corrplot 0.95 loaded
```

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

```
                  chocolate      fruity     caramel peanutyalmondy      nougat
chocolate         1.0000000 -0.74172106  0.24987535     0.37782357  0.25489183
fruity           -0.7417211  1.00000000 -0.33548538    -0.39928014 -0.26936712
caramel           0.2498753 -0.33548538  1.00000000     0.05935614  0.32849280
peanutyalmondy    0.3778236 -0.39928014  0.05935614     1.00000000  0.21311310
nougat            0.2548918 -0.26936712  0.32849280     0.21311310  1.00000000
crispedricewafer  0.3412098 -0.26936712  0.21311310    -0.01764631 -0.08974359
hard             -0.3441769  0.39067750 -0.12235513    -0.20555661 -0.13867505
bar               0.5974211 -0.51506558  0.33396002     0.26041960  0.52297636
pluribus         -0.3396752  0.29972522 -0.26958501    -0.20610932 -0.31033884
sugarpercent      0.1041691 -0.03439296  0.22193335     0.08788927  0.12308135
pricepercent      0.5046754 -0.43096853  0.25432709     0.30915323  0.15319643
winpercent        0.6365167 -0.38093814  0.21341630     0.40619220  0.19937530
                  crispedricewafer        hard         bar    pluribus
chocolate              0.34120978 -0.34417691  0.59742114 -0.33967519
fruity                -0.26936712  0.39067750 -0.51506558  0.29972522
caramel                0.21311310 -0.12235513  0.33396002 -0.26958501
peanutyalmondy        -0.01764631 -0.20555661  0.26041960 -0.20610932
nougat                -0.08974359 -0.13867505  0.52297636 -0.31033884
crispedricewafer       1.00000000 -0.13867505  0.42375093 -0.22469338
hard                  -0.13867505  1.00000000 -0.26516504  0.01453172
bar                    0.42375093 -0.26516504  1.00000000 -0.59340892
pluribus              -0.22469338  0.01453172 -0.59340892  1.00000000
sugarpercent           0.06994969  0.09180975  0.09998516  0.04552282
pricepercent           0.32826539 -0.24436534  0.51840654 -0.22079363
winpercent             0.32467965 -0.31038158  0.42992933 -0.24744787
                  sugarpercent pricepercent winpercent
chocolate           0.10416906    0.5046754  0.6365167
fruity             -0.03439296   -0.4309685 -0.3809381
caramel             0.22193335    0.2543271  0.2134163
peanutyalmondy      0.08788927    0.3091532  0.4061922
nougat              0.12308135    0.1531964  0.1993753
crispedricewafer    0.06994969    0.3282654  0.3246797
hard                0.09180975   -0.2443653 -0.3103816
bar                 0.09998516    0.5184065  0.4299293
pluribus            0.04552282   -0.2207936 -0.2474479
sugarpercent        1.00000000    0.3297064  0.2291507
pricepercent        0.32970639    1.0000000  0.3453254
winpercent          0.22915066    0.3453254  1.0000000
```

```
corrplot(cij)
```



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

Looking at this plot we can see that chocolate and fruity, since colored red, have an anti-correlation. There is also a slight negative correlation between pluribus and bar type candy.

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

The two variables that are the most positively correlated are chocolate type candies along with bar types.

6. Principle component analysis:

Let's apply PCA using the prcom() function to our candy dataset remembering to set 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
```

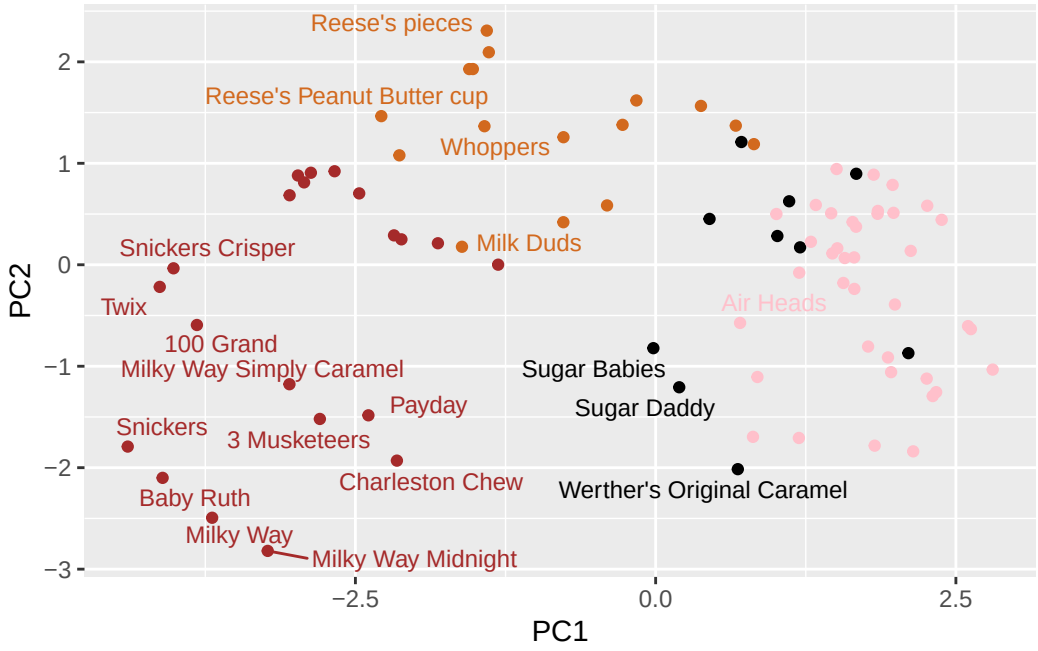
```
attributes(pca)
```

```
$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, size = 3.3, max.overlaps = 5)
```

```
Warning: ggrepel: 66 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



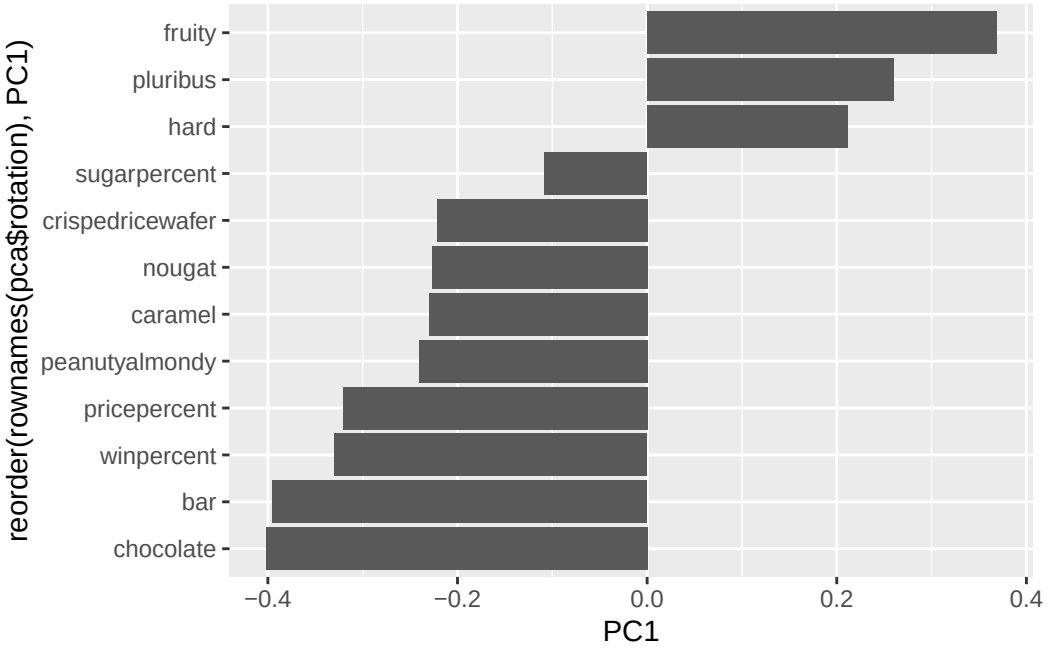
Let's look at how the original variables contribute to the original PCs, start with PC1.

```
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
```

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



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

The original variables that are picked up strongly by PC1 in the positive direction are fruity, pluribus, and hard. This makes sense because it shows that most fruity candies are hard and come in packs of multiple pieces, which is what is expected!