class09

Bayah Essayem (A17303992)

2025-02-10

Table of contents

```
candy_file <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-
head(candy_file)
```

```
             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?

```
nrow(candy_file)
```

```
[1] 85
```

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

```
sum(candy_file$fruity)
```

```
[1] 38
```

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

```
candy_file["Twix", "winpercent"]
```

```
[1] 81.64291
```

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

```
[1] 81.64291
```

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

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

```
[1] 76.7686
```

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

```
candy_file["Tootsie Roll",]$winpercent
```

```
[1] NA
```

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

```
sum(candy_file$chocolate)
```

```
[1] 37
```

To get a quick overview of the dataset the 'skimr' package can be useful:

```r
library(skimr)
skim(candy_file)
```

Table 1: Data summary

Name	candy_file
Number of rows	85
Number of columns	12
Column type frequency:	
numeric	12
Group variables	None

Variable type: numeric

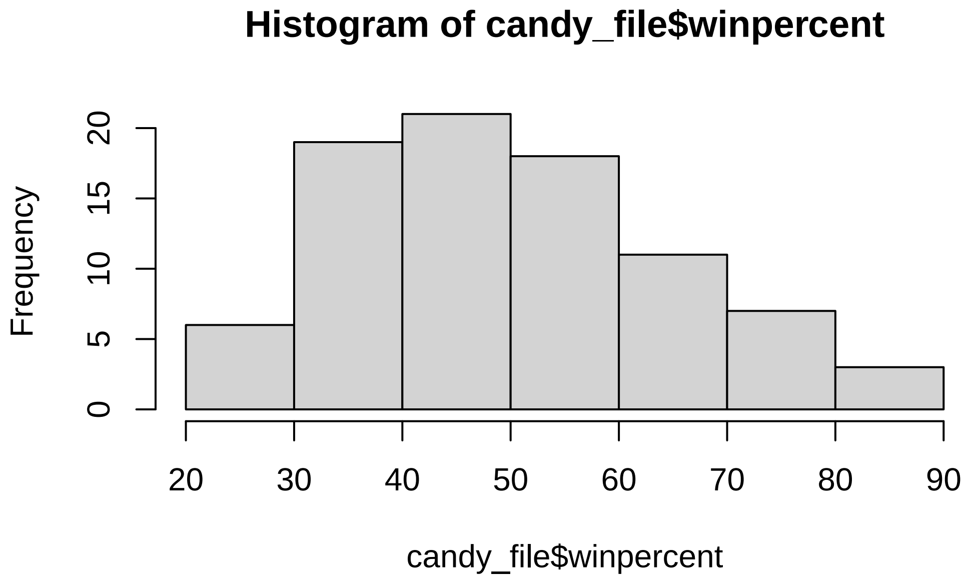
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
chocolate	0	1	0.44	0.50	0.00	0.00	0.00	1.00	1.00	
fruity	0	1	0.45	0.50	0.00	0.00	0.00	1.00	1.00	
caramel	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
peanutyalmondy	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
nougat	0	1	0.08	0.28	0.00	0.00	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? **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? A7. It means that the candy listed is either true for chocolate (represented by 1) or false for chocolate (represented by 0).

Q8. Plot a histogram of winpercent values

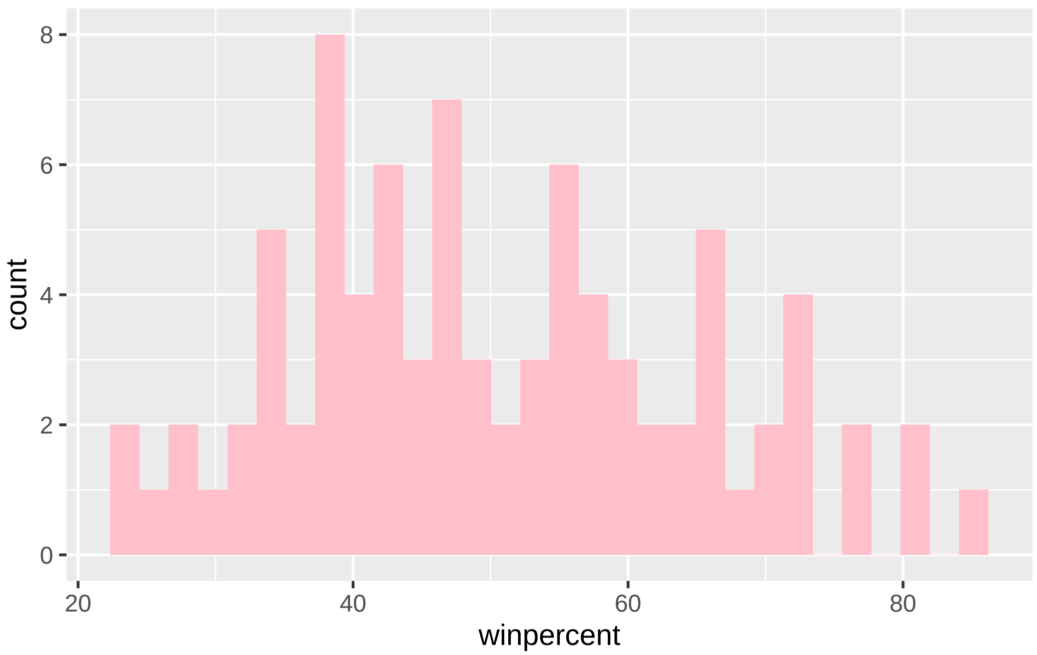
```r
hist(candy_file$winpercent)
```



If we were to do this in ggplot:

```r
library(ggplot2)

ggplot(candy_file) +
  aes(winpercent) +
  geom_histogram(bins = 30, fill = "pink")
```



Q9. Is the distribution of winpercent values symmetrical? A9. The disturbition of winepercent values are slightly right skewed, and not exactly symmetrical.

Q10. Is the center of the distribution above or below 50%? A10. The center of distribution is below 50%. Visualize the data, the center is around 40%. Based on the summary below, the median is below 50%.

```
summary(candy_file$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
- step 2: find their "winpercent" values
- step 3: summarize these values
- step 4: find all the "fruity" candy
- step 5: find their "winpercent" values
- step 6: summarize these values
- step 7: compare the two summary values

Step 1:

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

Step 2:

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

Step 3:

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

Step 4:

```
fruity.inds <- candy_file$fruity == 1
```

Step 5:

```
fruity.win <- candy_file[fruity.inds,]$winpercent
```

Step 6:

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

Compare to see which is greater:

```
fruity.mean
```

```
[1] 44.11974
```

```
choc.mean
```

```
[1] 60.92153
```

```
fruity.mean > choc.mean
```

```
[1] FALSE
```

> Q12. Is this difference statistically significant? A12. Yes, because by completing the t.test we can see that the p-value outputted is very low showing that there is a large enough of a difference for the chocolate and fruity candy be statistically different.

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

```

	Welch Two Sample t-test

data:  choc.win and fruity.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 
```

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

```
sort(candy_file$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
```

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

```
# Not that useful - it just sorts the values

sort(candy_file$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
```

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

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

```r
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 win.percent to make them sorted and use that order to arrange the whole dataset.

```r
can.win <- candy_file$winpercent
order(can.win)
```

```
 [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
```

```r
ord.inds <- order (can.win)
head(candy_file[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
```

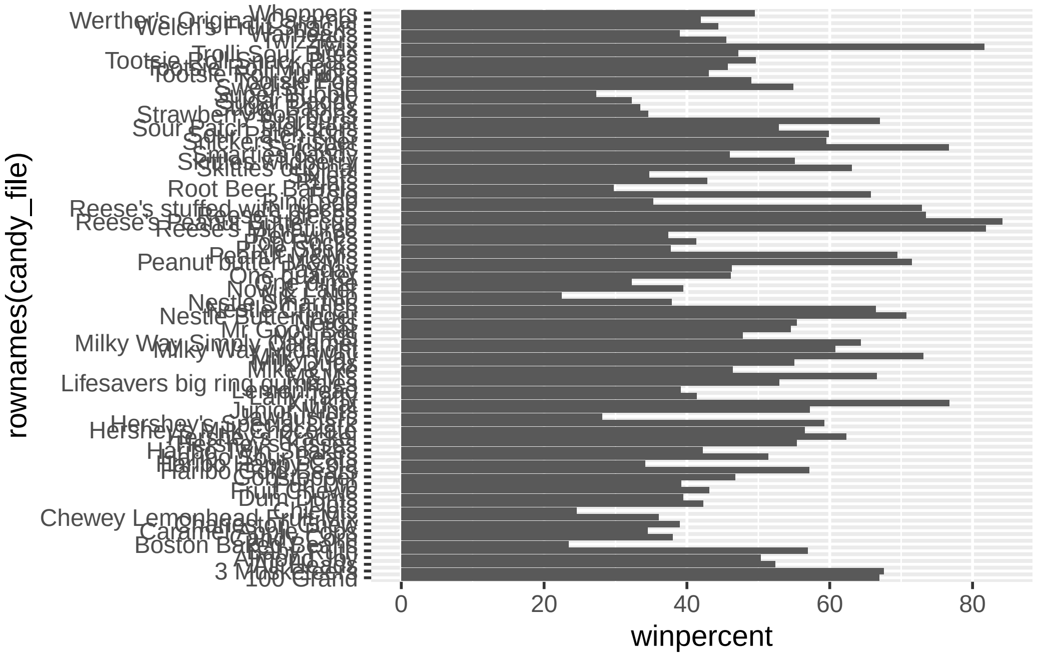
```
ord.inds <- order(can.win, decreasing = T)
head(candy_file[ord.inds,])
```

```
                          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
```

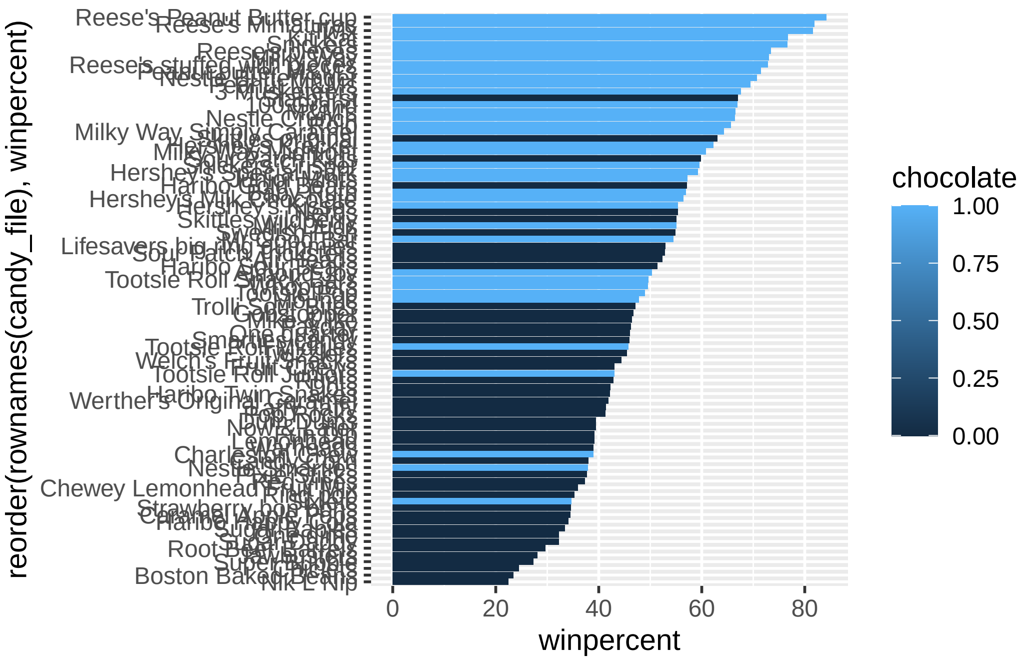
```
Twix                                0.906   81.64291
Kit Kat                             0.511   76.76860
Snickers                            0.651   76.67378
Reese's pieces                      0.651   73.43499
```

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

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



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



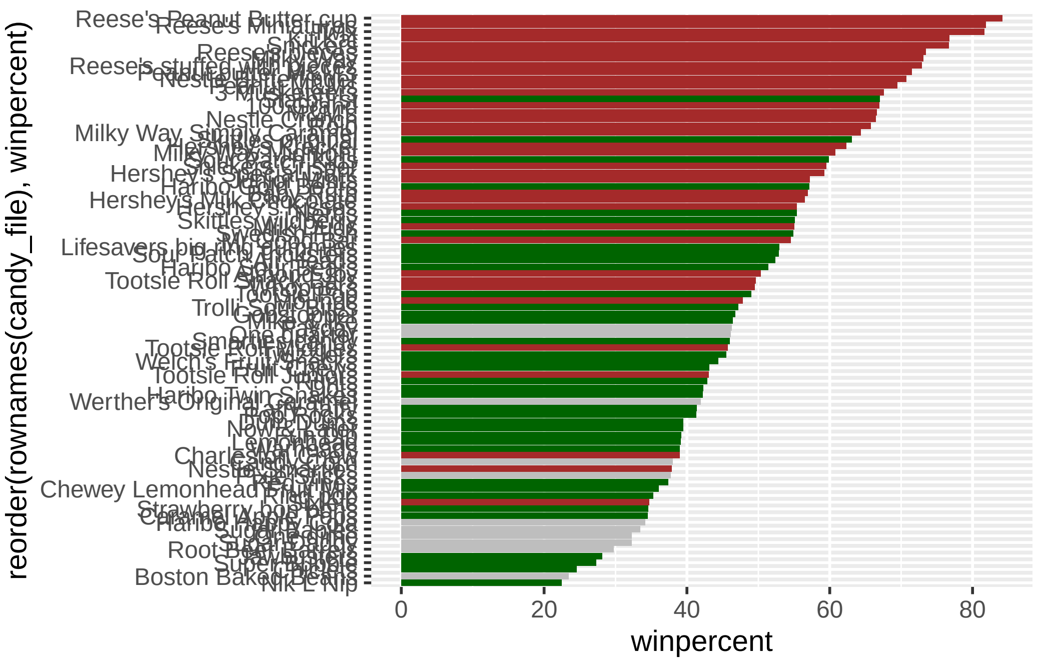
We need to make our own color vector where we can spell out exactly what candy is colored a particular color.

```
mycols <- rep("gray", nrow(candy_file))
mycols[candy_file$chocolate == 1] <- "brown"
mycols[candy_file$fruity == 1] <- "darkgreen"
mycols
```

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

Now, this color vector can be applied to the previous graph.

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



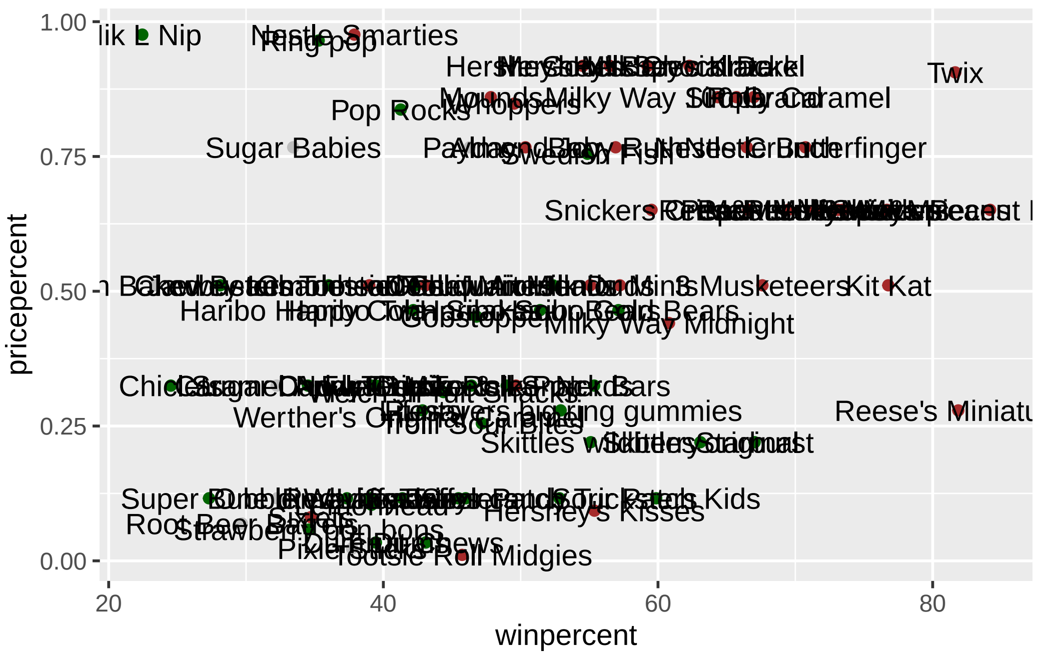
Q17. What is the worst ranked chocolate candy? A17. The worst ranked chocolate candy is Sixlets.

Q18. What is the best ranked fruity candy? A18. The best ranked fruity candy is starburst

Taking a look at priceprecent

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

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

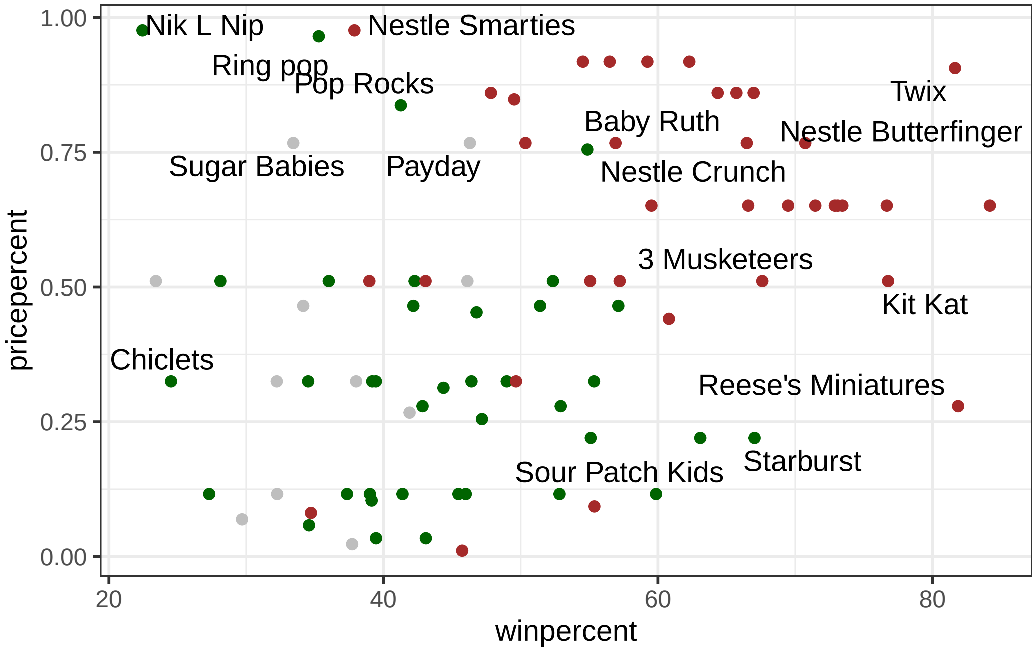


To avoid the overplotting of the text labels we can use the add on package **ggrepel**

```r
library(ggrepel)
```

```r
ggplot(candy_file) +
  aes(winpercent, pricepercent, label = rownames(candy_file)) +
  geom_point(col = mycols) +
  geom_text_repel(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? A19. The candy type that is highest ranked in terms of winpercent fpr the least money is

```
ord <- order(candy_file$pricepercent, decreasing = TRUE)
head( candy_file[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
```

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

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 **corrplot** package to plot a correlation matrix.

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

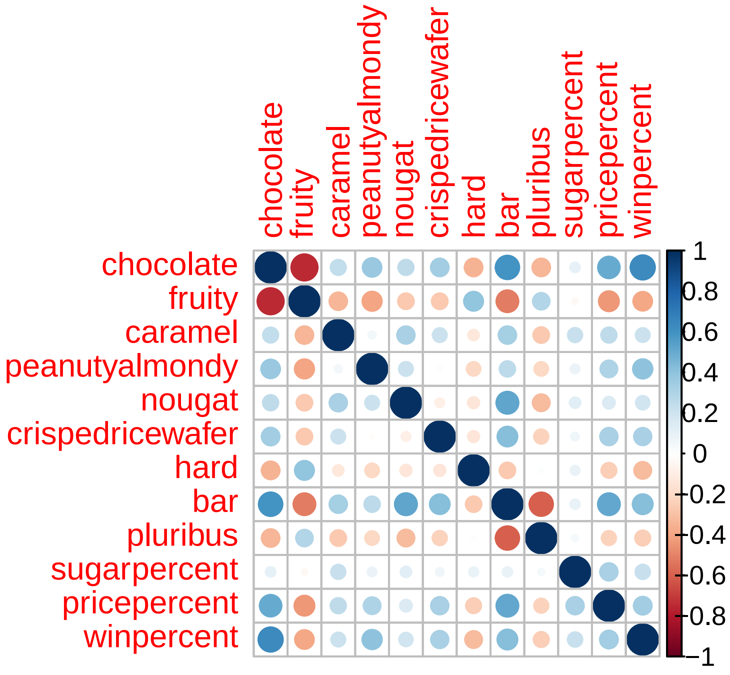
```
                  chocolate      fruity     caramel peanutyalmondy      nougat
chocolate         1.0000000 -0.74172106  0.24987535     0.37782357  0.25489183
fruity           -0.7417211  1.00000000 -0.33548538    -0.39928014 -0.26936712
caramel           0.2498753 -0.33548538  1.00000000     0.05935614  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
```

```
library(corrplot)
```

```
corrplot 0.95 loaded
```

```
corrplot(cij)
```



```
#size of circle correlates with magnitude of the correlation
```

6. 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_file, scale = T)
```

```r
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
```

```r
attributes(pca)
```

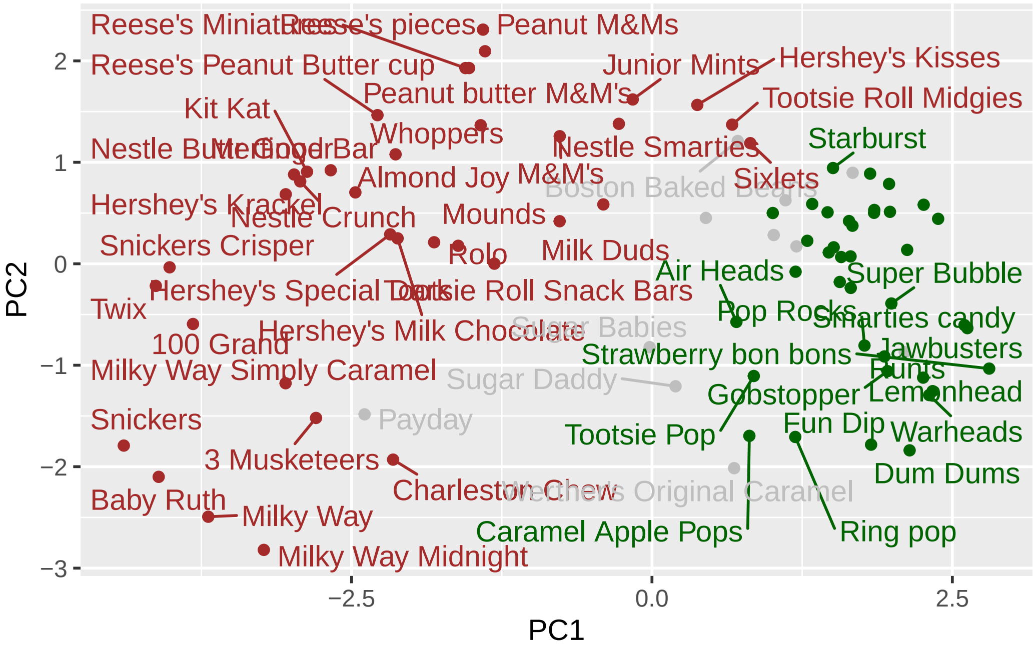
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"       

$class
[1] "prcomp"
```

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

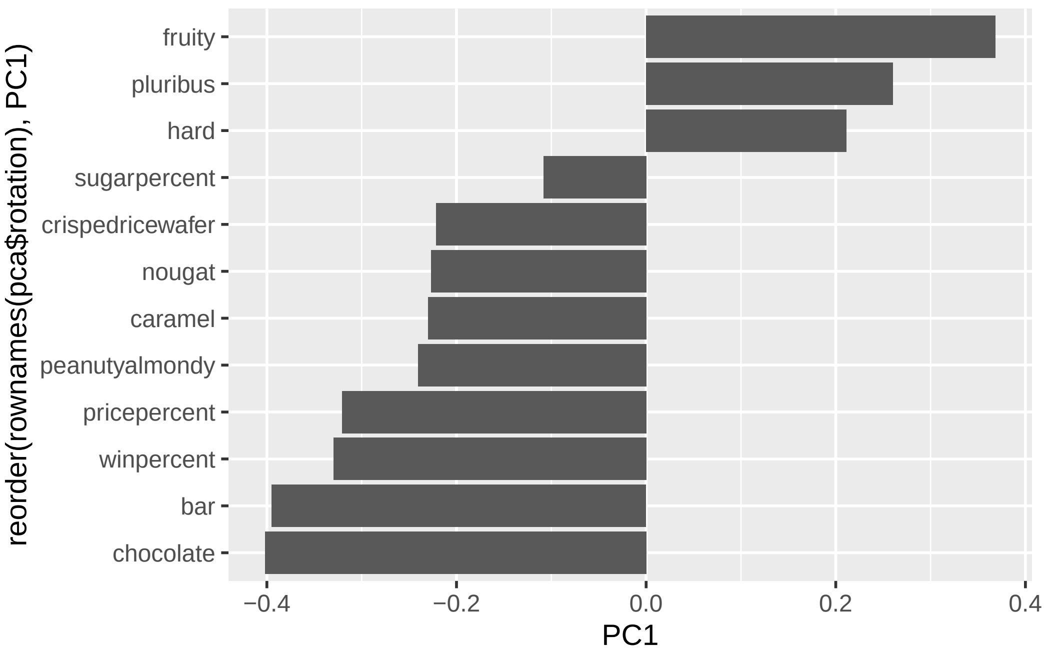
```r
ggplot(pca$x) + aes(PC1, PC2, label = rownames(pca$x)) +
  geom_point(col=mycols) +
  geom_text_repel(col = mycols, max.overlaps = 13)
```

```
Warning: ggrepel: 30 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



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

```
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? A24. They picked up hard, fruity, pluribus. These make sense since they can be grouped up together but are dissmilar to the other rownames that are common features of chocolate unlike the hard, fruity, pluribus rows