Halloween Candy Mini Project

Nhi To

2025-02-04

Table of contents

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

Importing candy data

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

ANSWER: There are 85 different candy types in this dataset.

```
nrow(candy)
```

```
[1] 85
```

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

ANSWER: There are 38 fruity candy types in the dataset.

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

```
[1] 38
```

Additional Question: How many chcocolate candy are there in the dataset?

ANSWER: There are 37 chocolate candy in the dataset.

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

```
[1] 37
```

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

ANSWER: My favorite candy is Pop Rocks, and its winpercent is 41.26551.

```
candy["Pop Rocks", "winpercent"]
```

```
[1] 41.26551
```

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

ANSWER: The winpercent for Kit Kat is 76.7686.

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

```
[1] 76.7686
```

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

ANSWER: The winpercent value for Tootsie Roll Snack Bars is 49.6535.

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

```
[1] 49.6535
```

```
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?

ANSWER: The variable 'winpercent' appears to be on a different scale compared to the majority of the other columns in the dataset, because the other variables

are between 0-1. The variable 'winpercent' values are between 0-100%, rather than 0-1, revealing that it is at a different scale value. Therefore, 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?

ANSWER: I think that the zero represents that that particular candy does not have any chocolate inside it. A one would represent that that candy contains chocolate.

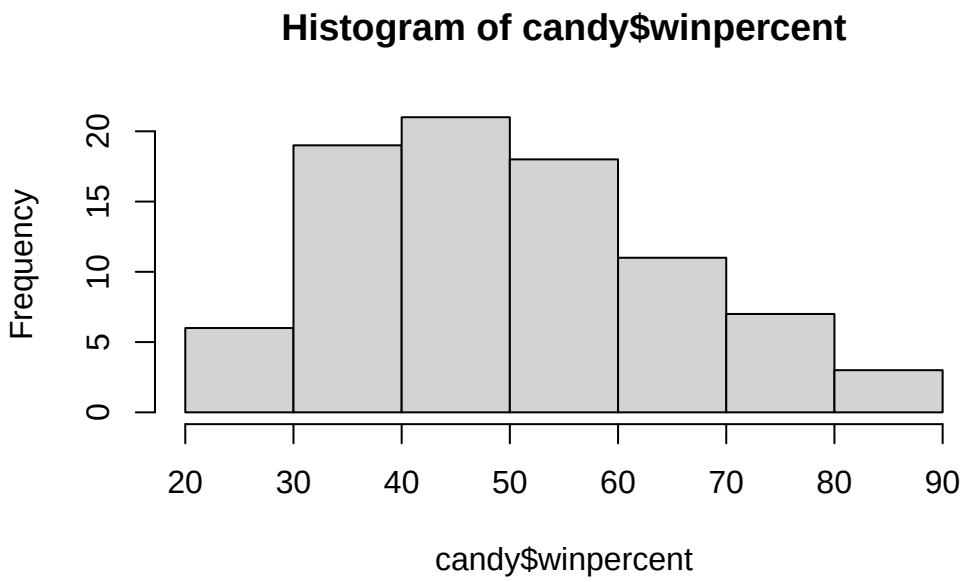
Q8. Plot a histogram of winpercent values

ANSWER:

```
library(ggplot2)

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


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



Q9. Is the distribution of winpercent values symmetrical?

ANSWER:The distribution of winperent values are slightly right skewed, and therefore not symmetrical.

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

ANSWER: The center of the distribution is below 50%. Visually, based on the peak of the data, the center is around 40%. Based on the summary below, the median value is 47.83, 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 -Step 2: find their "winpercent" values -Step 3: Summarize these values -Step 4: find all "fruity" candy -Step 5:find their winpercent values -Step 6: Summarize these values -Step 7: Compare the two summary values.

ANSWER: The mean value for chocolate candy is 60.92153 compared to the mean value of the fruity candy at 44.11974. Therefore we conclude that, on average, chocolate candy higher than fruity candy.

Step 1

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

Step 2

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

Step 3

```
mean(choc.win)
```

```
[1] 60.92153
```

Step 4

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

Step 5:

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

Step 6:

```
mean(fruit.win)
```

```
[1] 44.11974
```

Step 7:

```
mean(choc.win) > mean(fruit.win)
```

```
[1] TRUE
```

Q12. Is this difference statistically significant?

ANSWER:The p-value (2.871e-08) is less than 0.05, indicating that this difference is statistically signficant.

```
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
```

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

ANSWER: The five least liked candy types in this set is the Nik L Nip, Boston Baked Beans, Chiclets, Super Bubble, and 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
```

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

ANSWER: The top 5 all time favorite candy types out of this set is Reese's Peanut Butter cup, Reese's miniatures, Twix, Kit Kat, and Snickers.

```
# Not that useful- it just sorts the values
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)
sort(x)
```

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

```
order(x)
```

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

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

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

The 'order()' function tells us how to arrange the elments of the input to make them sorted - i.e. how to order them

We can determine the order of winpercent to make sorted and use them sorted and use that order to arrange 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
```

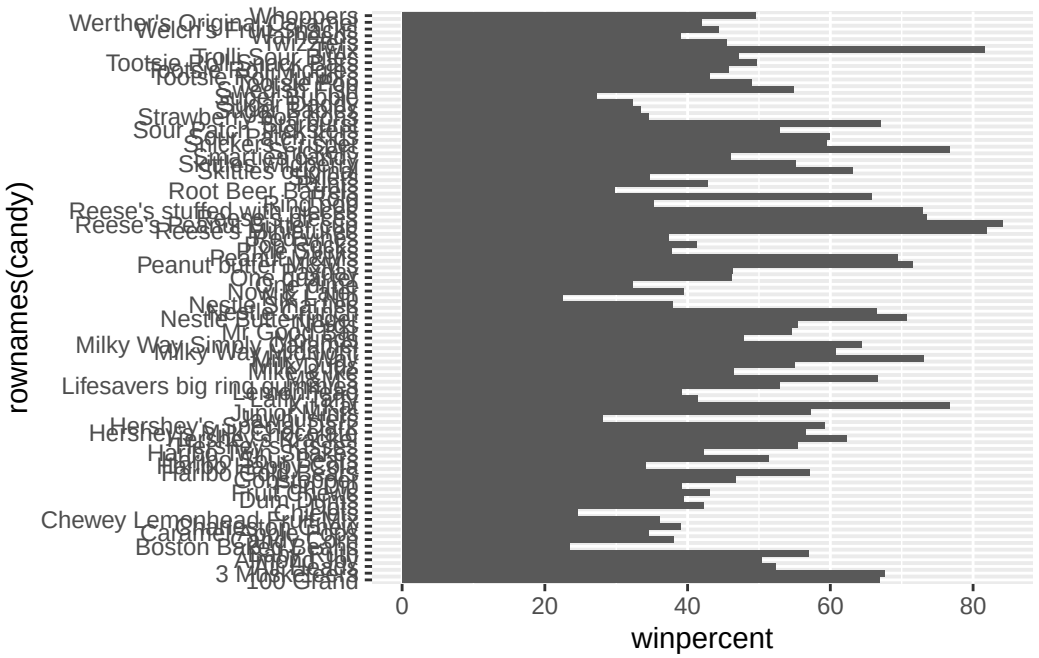
```r
ord.inds <- order(candy$winpercent, decreasing=T)
head( candy[ord.inds,] )
```

```
                        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
```

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

ANSWER:

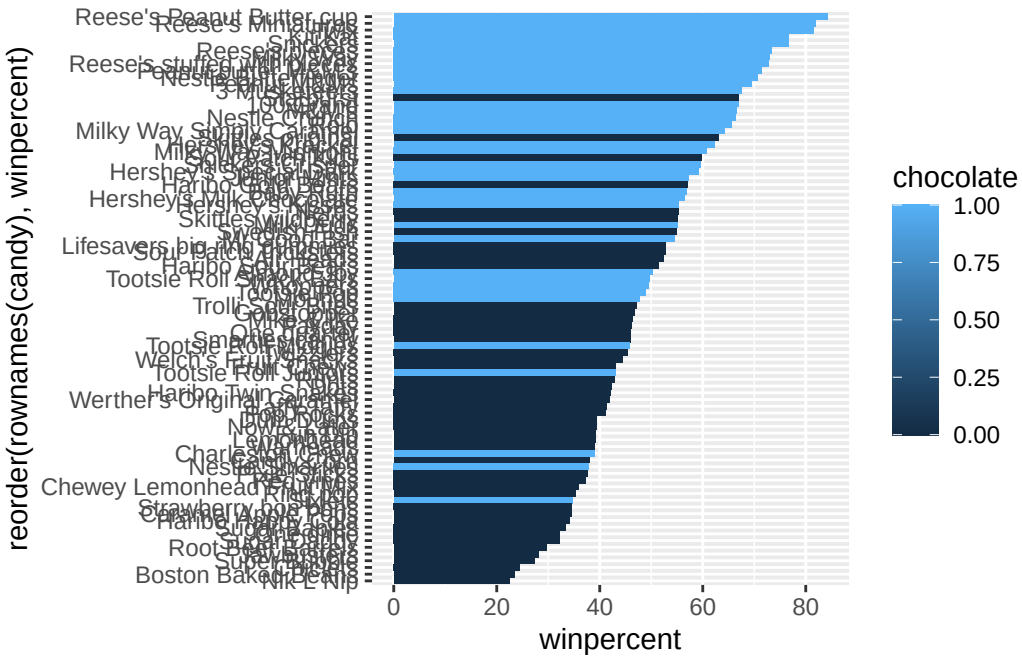
```
ggplot(candy)+
  aes(winpercent, rownames(candy))+
  geom_col()
```



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

ANSWER:

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



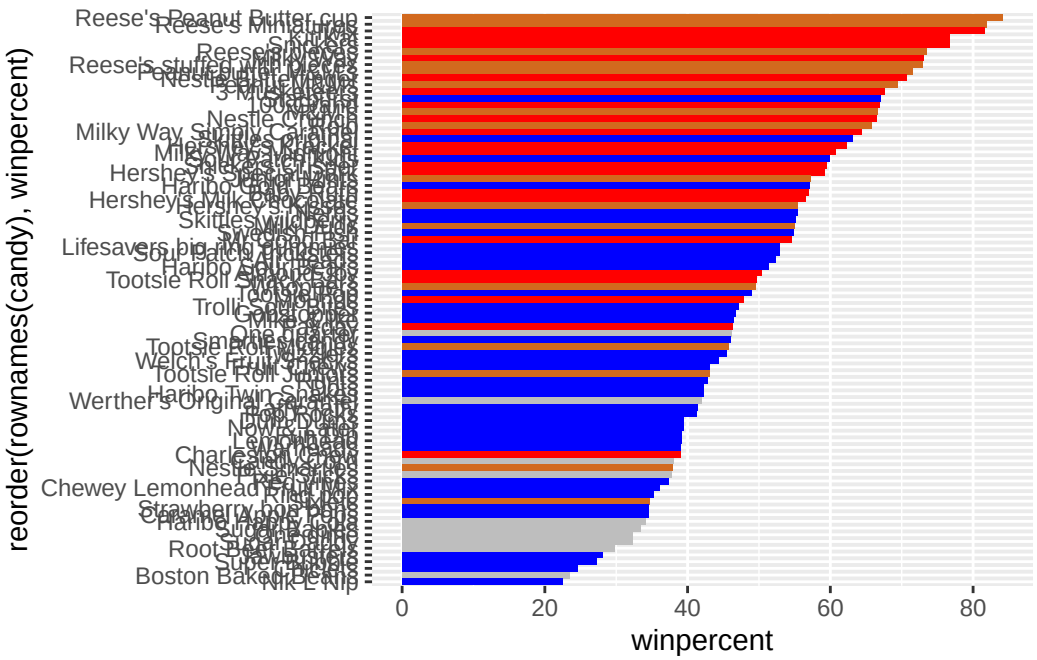
We need to make our own separate color vector where we can spell out what candy is colored a particular color

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

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

```
[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?

> **ANSWER**: The worst ranked chcolate candy is Sixlets.

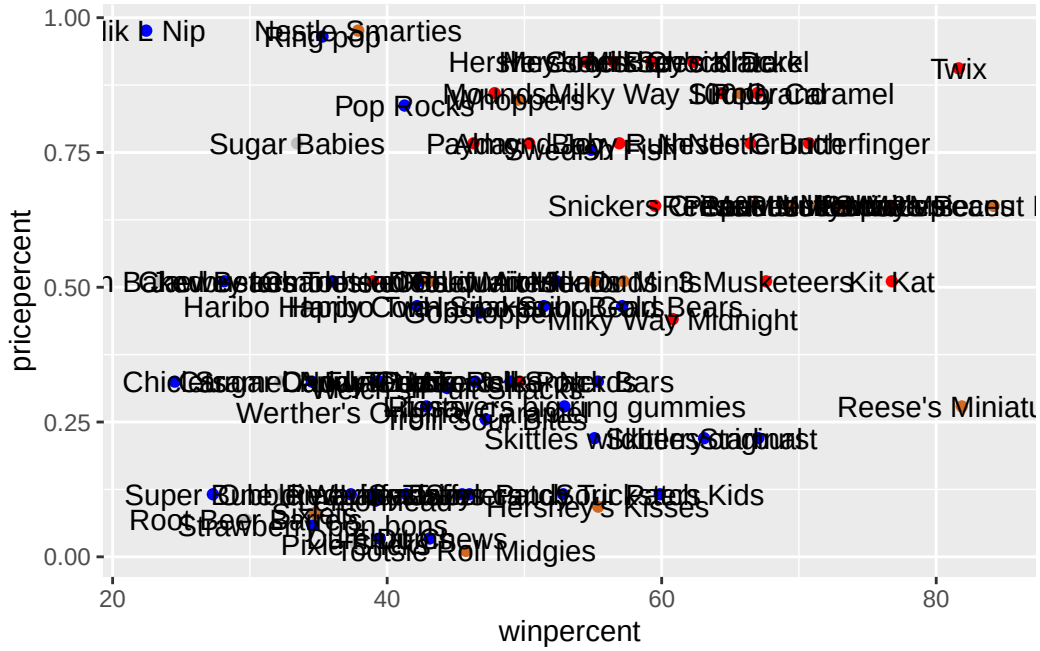
> Q18. What is the best ranked fruity candy?

> **ANSWER**: The best ranked fruity candy is Starburst.

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()
```

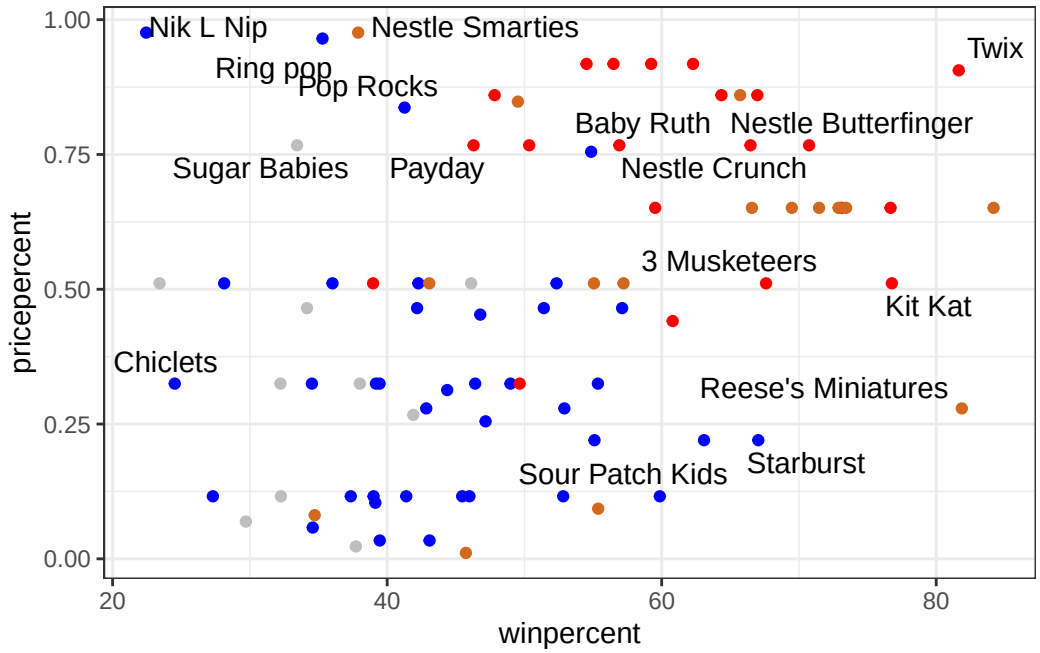


To avoid the overplotting of the text label, we can use the add on package **ggrepel**

```
library (ggrepel)

ggplot(candy)+
  aes(winpercent, pricepercent, label=rownames(candy))+
  geom_point(col=mycols)+
  geom_text_repel(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?

ANSWER: The candy type that is the highest ranked in terms of winpercent for the least money is Tootsie Roll Midgies. The highest winpercent/priceperent ratio is 4157.8862, which is the for the Tootsie Roll Midgies.

```
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
```

```
candy$Q19 <- candy$winpercent/candy$pricepercent

ord2<- order(candy$Q19, decreasing=TRUE)
head( candy[ord2, ], n=5 )
```

```
                   chocolate fruity caramel peanutyalmondy nougat
Tootsie Roll Midgies         1      0       0              0      0
Pixie Sticks                 0      0       0              0      0
Fruit Chews                  0      1       0              0      0
Dum Dums                     0      1       0              0      0
Strawberry bon bons          0      1       0              0      0
                   crispedricewafer hard bar pluribus sugarpercent
Tootsie Roll Midgies                0    0   0        1        0.174
Pixie Sticks                        0    0   0        1        0.093
Fruit Chews                         0    0   0        1        0.127
Dum Dums                            0    1   0        0        0.732
Strawberry bon bons                 0    1   0        1        0.569
                   pricepercent winpercent       Q19
Tootsie Roll Midgies        0.011   45.73675 4157.8862
Pixie Sticks                0.023   37.72234 1640.1016
Fruit Chews                 0.034   43.08892 1267.3212
Dum Dums                    0.034   39.46056 1160.6045
Strawberry bon bons         0.058   34.57899  596.1895
```

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

ANSWER: The top 5 most expensive candy types in the dataseet are Nik L Nip, Nestle Smarties, Ring pop, Hershe's Krackel, and Hershey's Milk Chocolate. Out of these 5, Nik L Nip has the lowest winpercent, and thus is the least popular out of the 5 most expensive candy types.

5 Exploring the correlation structure

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

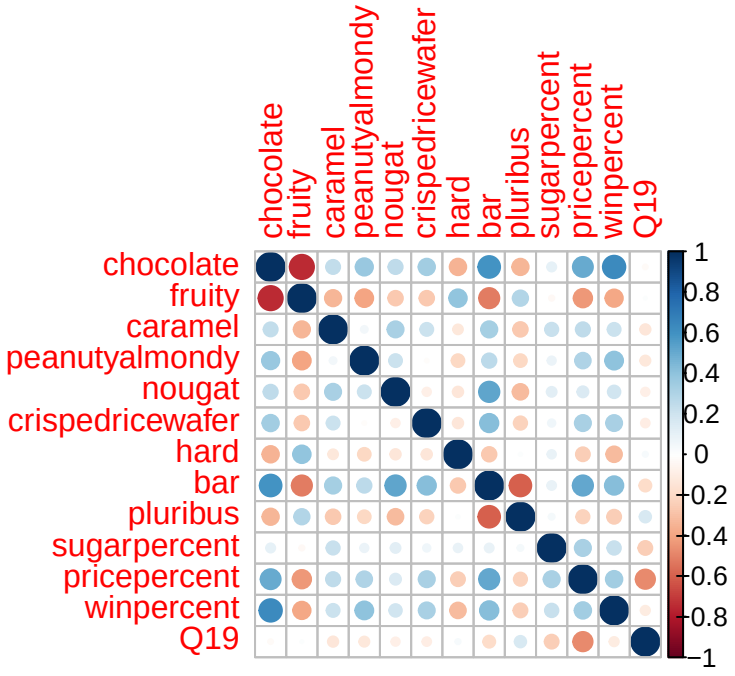
First, we wil use correlation and view the results with the **corplot** package to plot a correlation matrix

```
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)?

ANSWER: Based on this plot, the two variables "fruity" and "chocolate" are anti-correlated, because they both have negative values (depicted by the red circle). This means that fruity candy tends to not have chocolate in it.

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

ANSWER: Based on this plot, the two variables "chocolate" and "caramel" are positively correlated, because they both have positive vlaues (depicted by the blue circle). This means that chocolate candy tends to have caramel in it.

6. Principal Component Analysis

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

ANSWER: The original variables that are picked up by PC1 in the positive direction are the fruity, pluribis, and hard. This means that that fruity candy are often pluribis and hard as well.

Let's apply PCA using the 'prcomp()' function

```
pca <- prcomp(candy, scale=TRUE)
summary(pca)
```

```
Importance of components:
                          PC1    PC2     PC3    PC4     PC5     PC6     PC7
Standard deviation     2.0938 1.2127 1.13054 1.0787 0.98027 0.93656 0.81530
Proportion of Variance 0.3372 0.1131 0.09832 0.0895 0.07392 0.06747 0.05113
Cumulative Proportion  0.3372 0.4503 0.54866 0.6382 0.71208 0.77956 0.83069
                           PC8     PC9    PC10    PC11    PC12    PC13
Standard deviation     0.78462 0.68466 0.66328 0.57829 0.43128 0.39534
Proportion of Variance 0.04736 0.03606 0.03384 0.02572 0.01431 0.01202
Cumulative Proportion  0.87804 0.91410 0.94794 0.97367 0.98798 1.00000
```

```
attributes(pca)
```

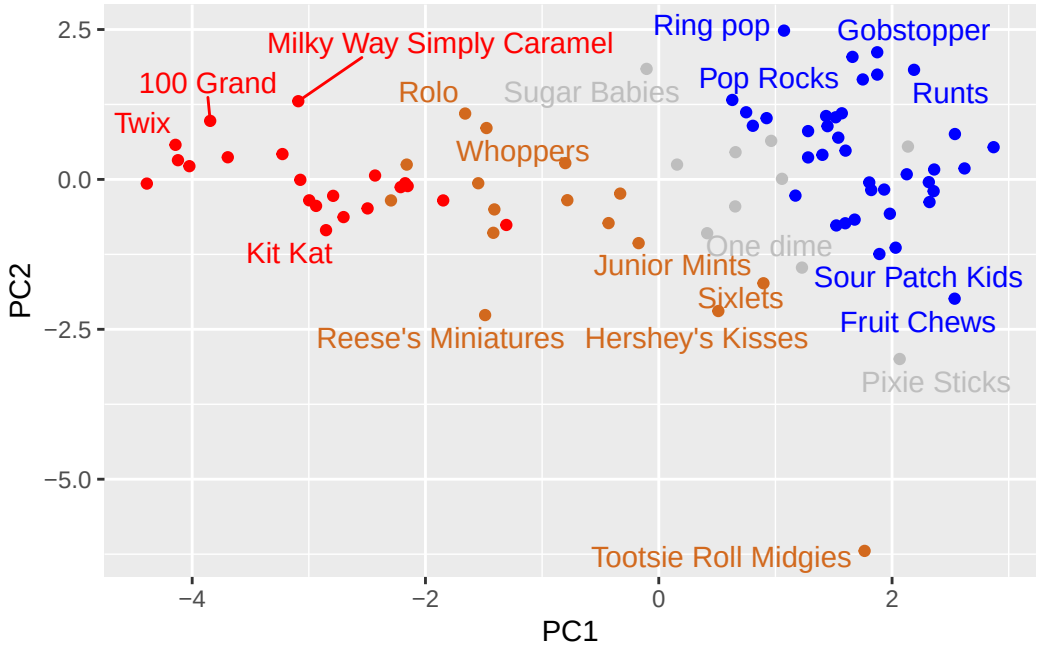
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$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: 65 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



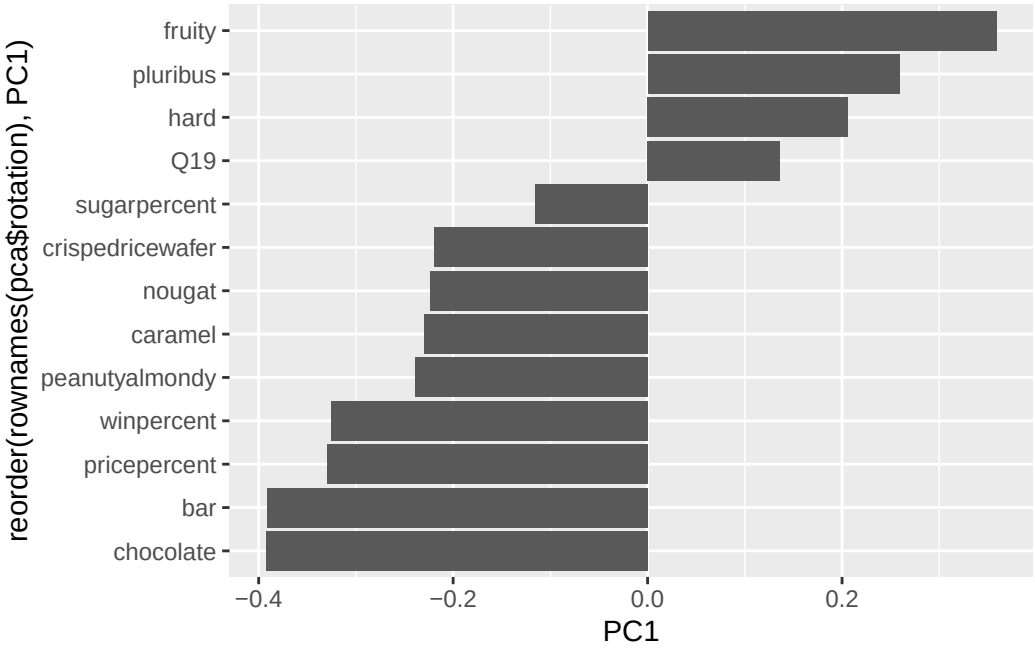
Finally let's look at how the original variables contribute to the PCs, start with PC1

```
pca$rotation
```

```
                    PC1          PC2         PC3          PC4         PC5
chocolate     -0.3924439 -0.219448144  0.15058469 -0.0114250051 -0.07105041
fruity         0.3588085  0.265751128 -0.06419261 -0.0414216903  0.10444609
caramel       -0.2293954  0.185201638 -0.29188009 -0.1131639560 -0.51223340
peanutyalmondy -0.2389173 -0.112739803  0.13689124  0.5237380179  0.27665942
nougat        -0.2241826  0.022531785 -0.55962535  0.3358934505 -0.14260284
crispedricewafer -0.2195121  0.039093933  0.22057443 -0.6773790754 -0.03722188
hard           0.2059573  0.320563450 -0.26209629 -0.1224691573  0.06236033
bar           -0.3912663 -0.004827650 -0.25701611 -0.1634667675  0.12009145
pluribus       0.2590791 -0.002370332  0.46082634  0.2144772981 -0.38079984
sugarpercent  -0.1161206  0.548733387  0.16560684  0.2005637175 -0.46041761
pricepercent  -0.3299041  0.311975067  0.25547945 -0.0007214477  0.16401024
winpercent    -0.3250778 -0.075548141  0.24143214  0.0768869968 -0.11815717
Q19            0.1359085 -0.570297097 -0.09663968 -0.0743019060 -0.45522982
                    PC6           PC7         PC8          PC9
chocolate      0.168012098 -0.0820264438 -0.22902045 -0.356733735
fruity         0.070137283  0.4628654891  0.27452454  0.094856568
caramel       -0.244254122 -0.4325648105  0.48368965  0.006498170
peanutyalmondy 0.283589046 -0.2543683871  0.30643273  0.537530170
```

```
nougat           -0.132132306  0.3641241953 -0.19720243  0.245456190
crispedricewafer  0.080758716  0.1328929581  0.09283457  0.530877726
hard              0.660819574 -0.3168714332 -0.18487010  0.009347836
bar              -0.003492824  0.2356475753 -0.24112853  0.147918692
pluribus         -0.219240043  0.0395262256 -0.20700502  0.282258901
sugarpercent      0.267637811  0.1460393254 -0.16224957 -0.007795535
pricepercent     -0.166682572 -0.1579120355 -0.34578465  0.047548219
winpercent        0.333724015  0.4161740441  0.42330437 -0.307143205
Q19               0.327480160 -0.0007099284 -0.19123968  0.185435980
                         PC10         PC11         PC12         PC13
chocolate         0.204343511  0.03055033  0.04150283  0.71477134
fruity           -0.010883670 -0.48239113  0.05817828  0.49412325
caramel           0.096671412 -0.21146289  0.10638886  0.08089608
peanutyalmondy   -0.089075185 -0.01744526  0.08152461  0.13332819
nougat            0.297846590  0.13181443 -0.38006014  0.05637103
crispedricewafer  0.081590021  0.24987963 -0.21829437  0.10012004
hard              0.429260577 -0.02949363  0.03199238 -0.09862215
bar              -0.057997495 -0.16266859  0.74322586 -0.14802838
pluribus          0.517349972 -0.02681627  0.29738048 -0.02339919
sugarpercent     -0.478913585  0.22932098  0.04480374  0.04974921
pricepercent     -0.002069487 -0.62192962 -0.34680471 -0.14157619
winpercent        0.289101852 -0.12797561 -0.06613302 -0.38474323
Q19              -0.277946525 -0.39713642 -0.12779672 -0.06847945
```

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


reorder(rownames(pca$rotation), PC1)
fruity
pluribus
hard
Q19
sugarpercent
crispedricewafer
nougat
caramel
peanutyalmondy
winpercent
pricepercent
bar
chocolate
−0.4
−0.2
0.0
0.2
PC1