Class 9: Halloween Candy Project

Nancy Leon Rivera pid A12925462

Background

Today we are delving into an analysis of Halloween candy data using ggplot, dplyr, basic stats, correlations analysis, and PCA

##Import the data

```
candy <- read.csv("candy-data.txt", row.names = 1)
head(candy)
```

```
             chocolate fruity caramel peanutyalmondy nougat crispedricewafer
100 Grand            1      0       1              0      0                 1
3 Musketeers         1      0       0              0      1                 0
One dime             0      0       0              0      0                 0
One quarter          0      0       0              0      0                 0
Air Heads            0      1       0              0      0                 0
Almond Joy           1      0       0              1      0                 0
             hard bar pluribus sugarpercent pricepercent winpercent
100 Grand       0   1        0        0.732        0.860   66.97173
3 Musketeers    0   1        0        0.604        0.511   67.60294
One dime        0   0        0        0.011        0.116   32.26109
One quarter     0   0        0        0.011        0.511   46.11650
Air Heads       0   0        0        0.906        0.511   52.34146
Almond Joy      0   1        0        0.465        0.767   50.34755
```

Q1. How many candies types are in this dataset?

```
#nrow is our friend
nrow(candy)
```

```
[1] 85
```

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

```
#access data and call what you are looking for
#candy$fruity
#want total
sum(candy$fruity)
```

```
[1] 38
```

Q3. How many chochlate candy types are in the dataset?

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

```
[1] 37
```

##What is your favorite candy?

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

```
[1] 55.35405
```

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

```
[1] 55.35405
```

```
#Can do candy["Nerds",]$c(winpercent, sugarpercent) to return multiple items
```

```
library(dplyr)
```

We can also use the `filter()` and `select()` function from **deplyr**.

```
candy|>
  filter(rownames(candy)=="Nerds") |>
  #Can select and return mutiple items
  select(winpercent, sugarpercent)
```

```
      winpercent sugarpercent
Nerds   55.35405        0.848
```

```
candy|>
  filter(rownames(candy)=="Twix") |>
  select(winpercent, sugarpercent)
```

```
     winpercent sugarpercent
Twix   81.64291        0.546
```

```
candy|>
  filter(rownames(candy)=="Kit Kat") |>
  select(winpercent, sugarpercent)
```

```
        winpercent sugarpercent
Kit Kat    76.7686        0.313
```

A useful function for a quick look at a new dataset is found in the **skimr** package:

```
##library(skimr) or
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	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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	

```
#based on the output row winpercent dominates the data and therefore, I must skim every datas
```

Q6. Is there any variable/column that looks to be on a different scale to the majority of the other columns in the dataset?

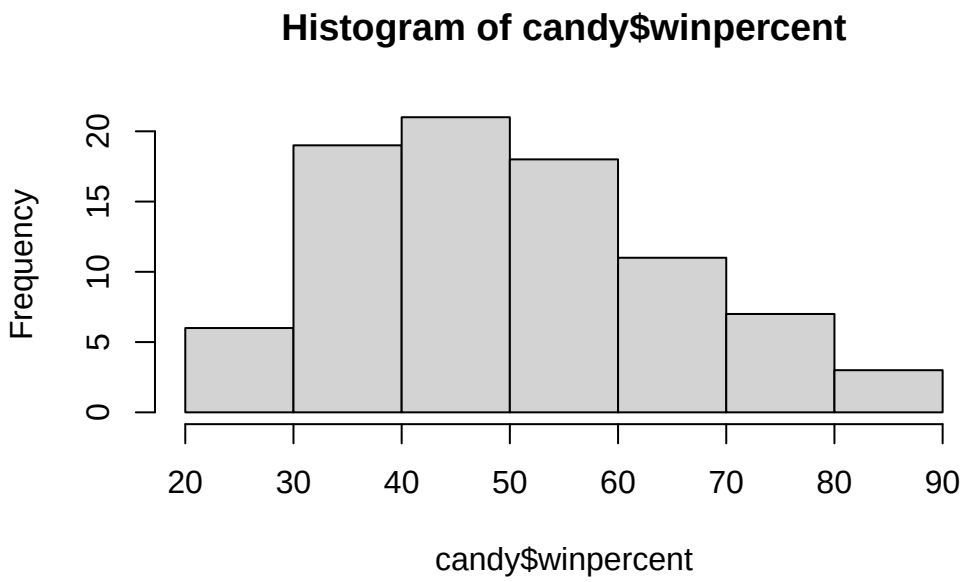
Yes, the 10th column `winpercent` is on a different "scale" or range than all the others. **N.B**We will beed to scale this data before analysis like PCA for example to avoid this one variable dominating our analysis.

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

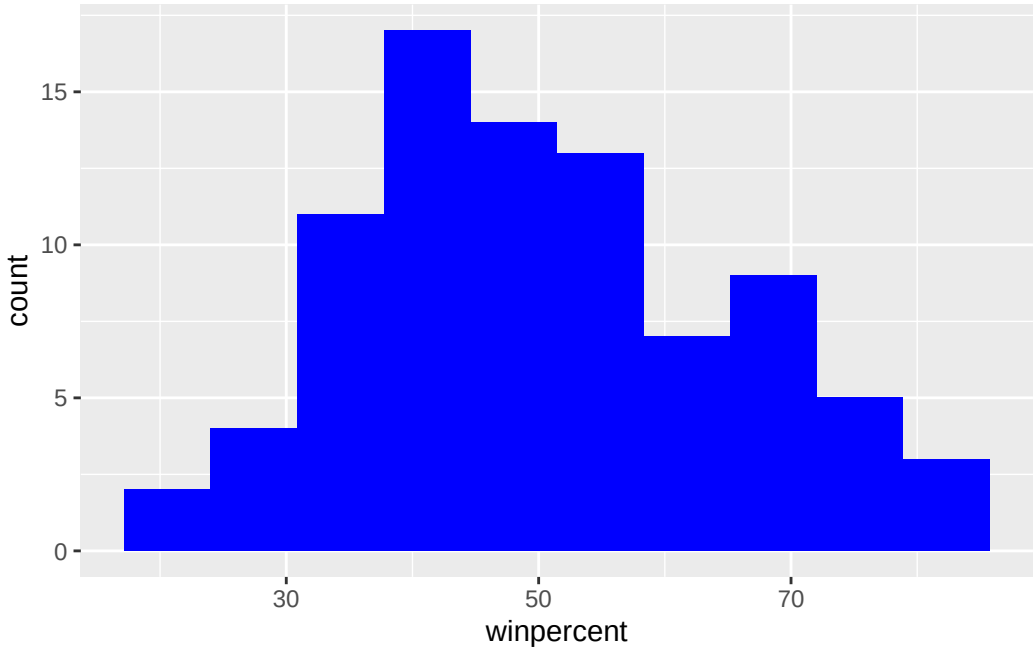
That the candy has no chocolate

Q8. Plot a histogram of winpercent values using R and ggplot.

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




```
library(ggplot2)
ggplot(candy)+
  aes(winpercent) +
  geom_histogram(bins=10, fill="blue")
```



Q9. Is the distribution of winpercent values symmetrical? No, not symetrical Q10. Is the center of the distribution above or below 50%? From the histogram it looks to be below the 50% mark

```
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?

Step1. Extract/Find chocolate candy rows in the data set. Step2. Get their winpercent values Step3. Calculate their mean winpercent values.

Repeat for fruity Candy Step4. Extract/Find fruity candy rows in teh dataset. Step5. Get their winpercent values Step6. Calculate their mean winpercent values.

Step7. Compare the mean choclate winpercent to mean fruity winpercent and see which one is larger.

1.Find choclate candy

```
choc.inds <- candy$chocolate==1
choc.candy <- candy[choc.inds, ]
```

2.get their winpercent

```
choc.win <- choc.candy$winpercent
```

3. calculate their mean

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

```
[1] 60.92153
```

4.Find fruity candy

```
fruit.inds <- candy$fruity==1
#or can do as.logical(candy$fruity)
fruit.candy <- candy[fruit.inds, ]
```

5.get their winpercent

```
fruit.win <- fruit.candy$winpercent
```

6. calculate their mean

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

```
[1] 44.11974
```

Q12. Is this difference statistically significant?

Let's use student T-test

```
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
```

```
#this is statistically significant
```

##3. Overall Candy Rankings

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

```
#only sorts the input and not helpful
#sort(candy$winpercent)
x <- c(10, 1, 100)
order(x)
```

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

So I can use the output of `order(winpercent)` to re-arrange or order my whole dataset by winpercent.

```
ord.inds <- order(candy$winpercent)
head(candy[order(candy$winpercent),], n=5)
```

```
                  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
                  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
                  winpercent
Nik L Nip           22.44534
Boston Baked Beans  23.41782
Chiclets            24.52499
Super Bubble        27.30386
Jawbusters          28.12744
```

```
#or candy %>% arrange(winpercent) %>% head(5)
```

```
candy|>
  arrange(winpercent) |>
  head()
```

```
                  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?

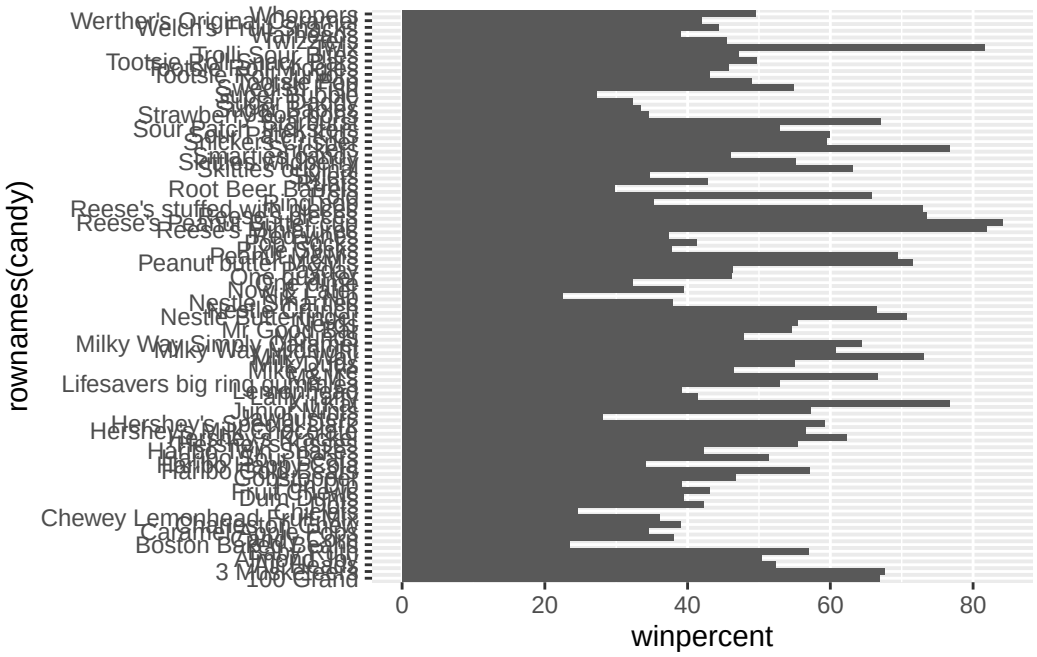
```
candy|>
  arrange(-winpercent) |>
  head(5)
```

```
                        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
                        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
                          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
```

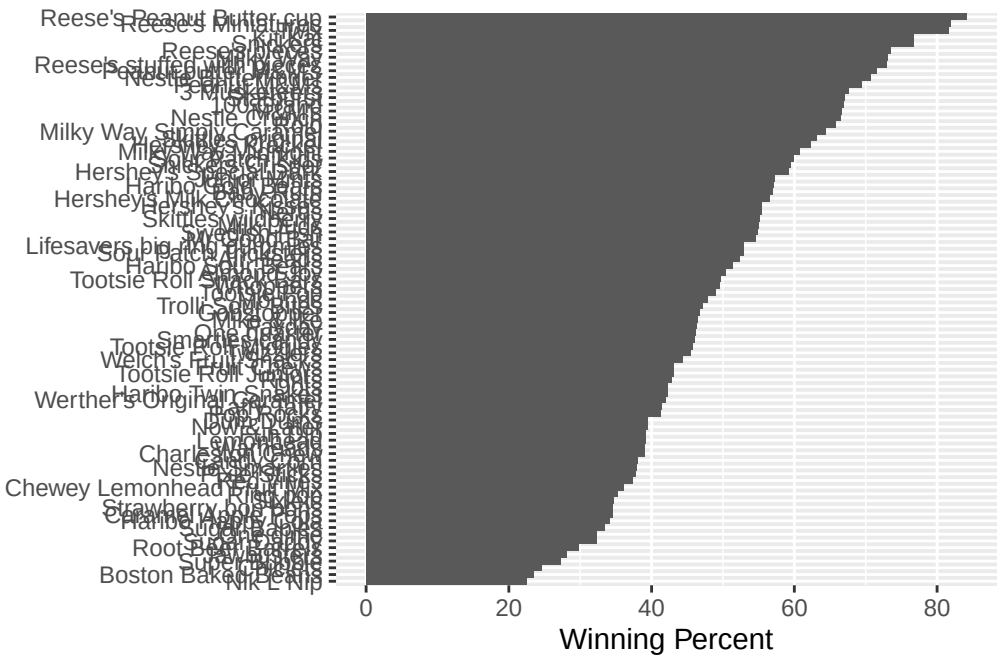
Q15. Make a first barplot of candy ranking based on winpercent values.

```
library(ggplot2)

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



We can make this plot better by rearranging (ordeing) the y-axis by winpercent so the highest scoring candy is at the top and lowest at bottom.

```
library(ggplot2)
ggplot(candy) +
aes(x=winpercent, y=reorder(rownames(candy), winpercent)) +
  geom_col()+
  ylab("") +
  xlab("Winning Percent")
```




```
ggsave("my_plot.png", height =12, width=6)
```

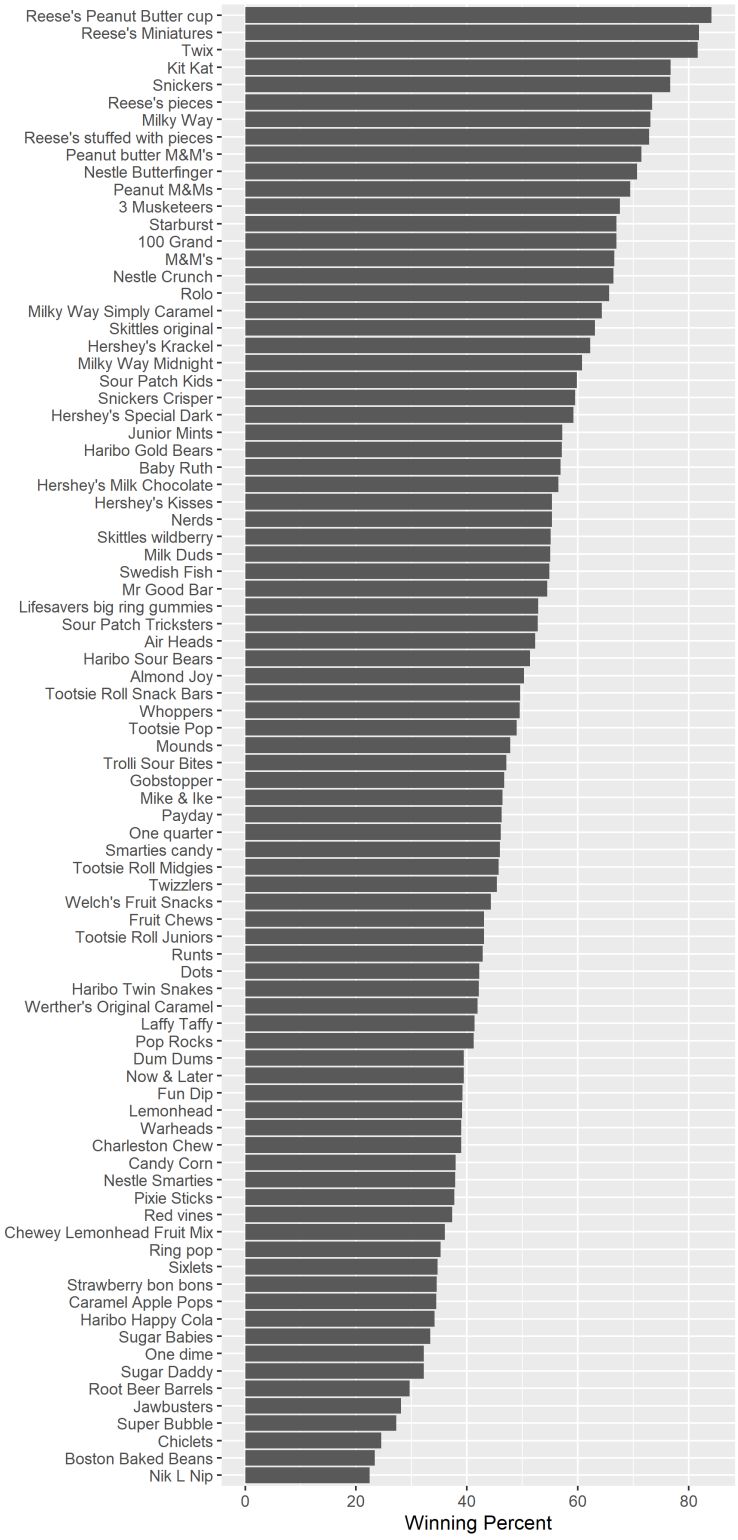
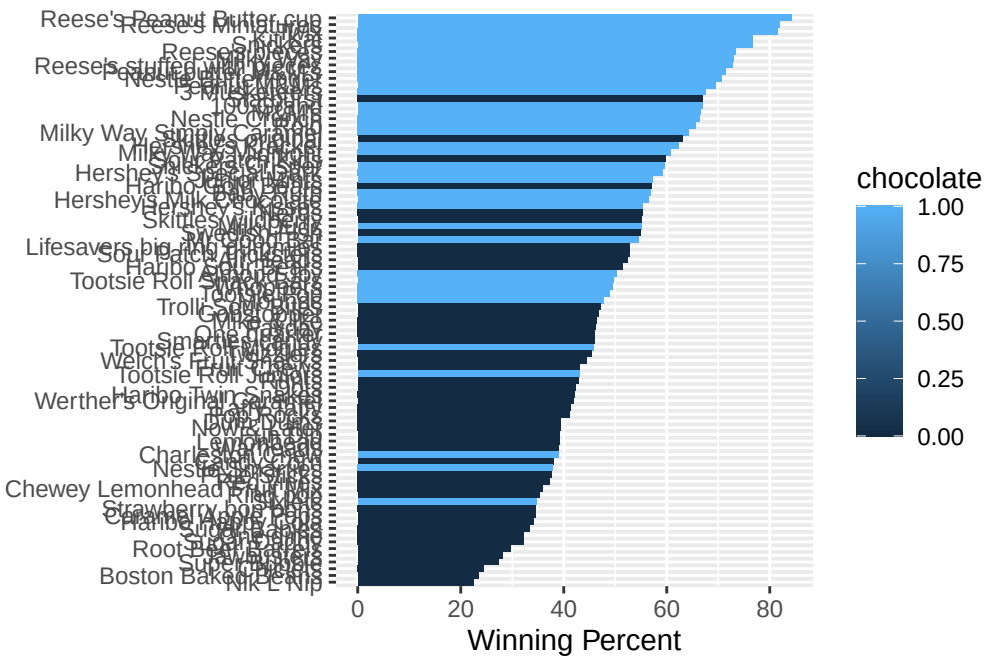


Figure 1: Fig X. SOme big plot

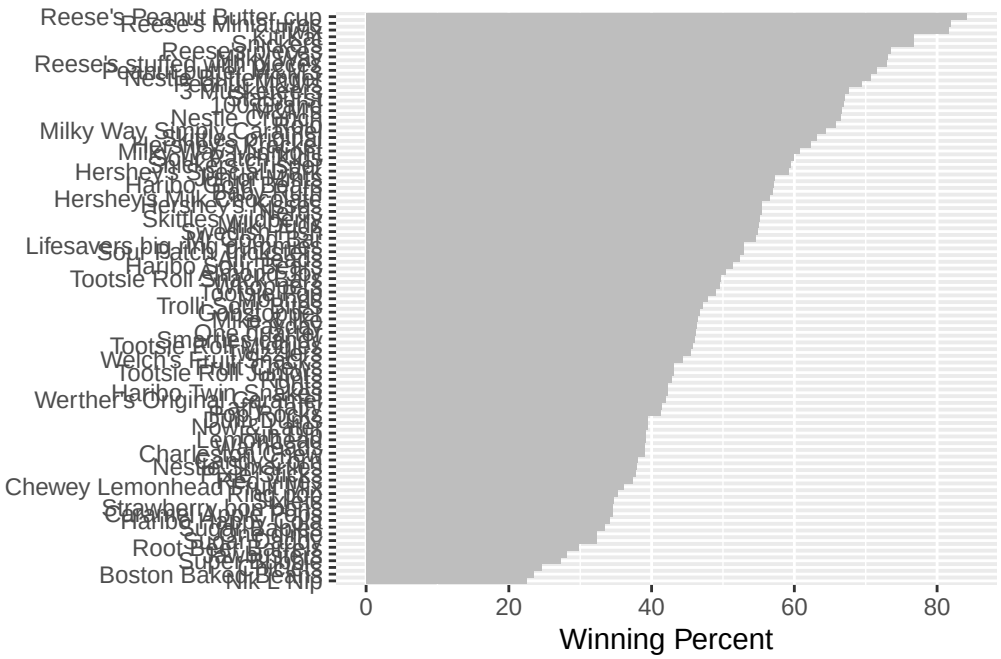
Q. Color your bars by "chocolate"

```
ggplot(candy) +
aes(x=winpercent, y=reorder(rownames(candy), winpercent), fill=chocolate)+
  geom_col()+
  ylab("") +
  xlab("Winning Percent")
```



I want to color choclate and fruity candy a specified color. To do this we need to define our own custom color vector that has the exact color mappings we want.

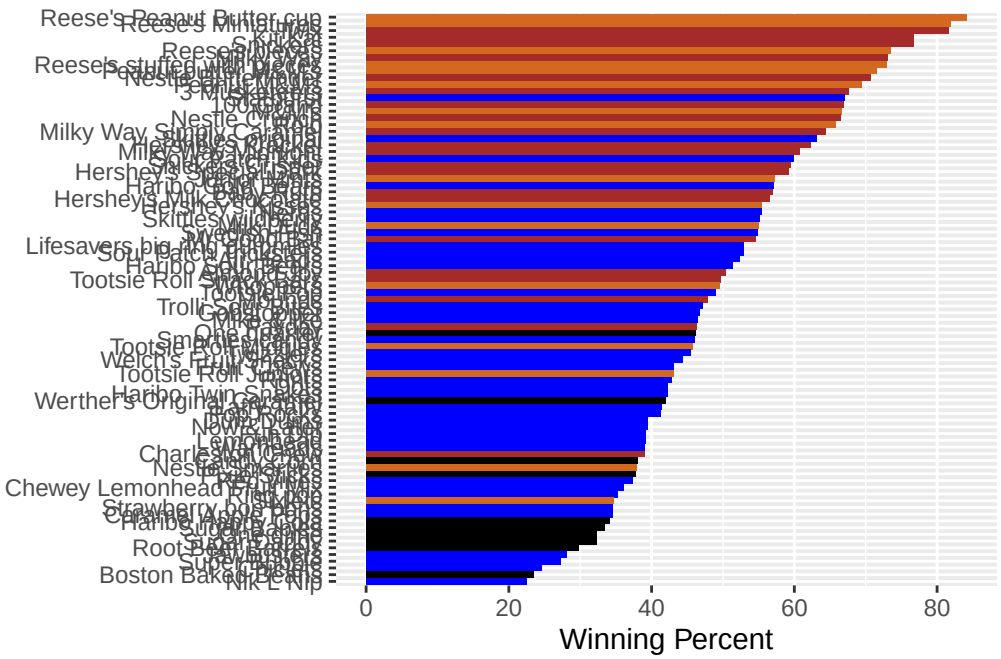
```
mycols <- rep("grey", nrow(candy))
ggplot(candy) +
aes(x=winpercent, y=reorder(rownames(candy), winpercent))+
  geom_col(fill= mycols)+
  ylab("") +
  xlab("Winning Percent")
```



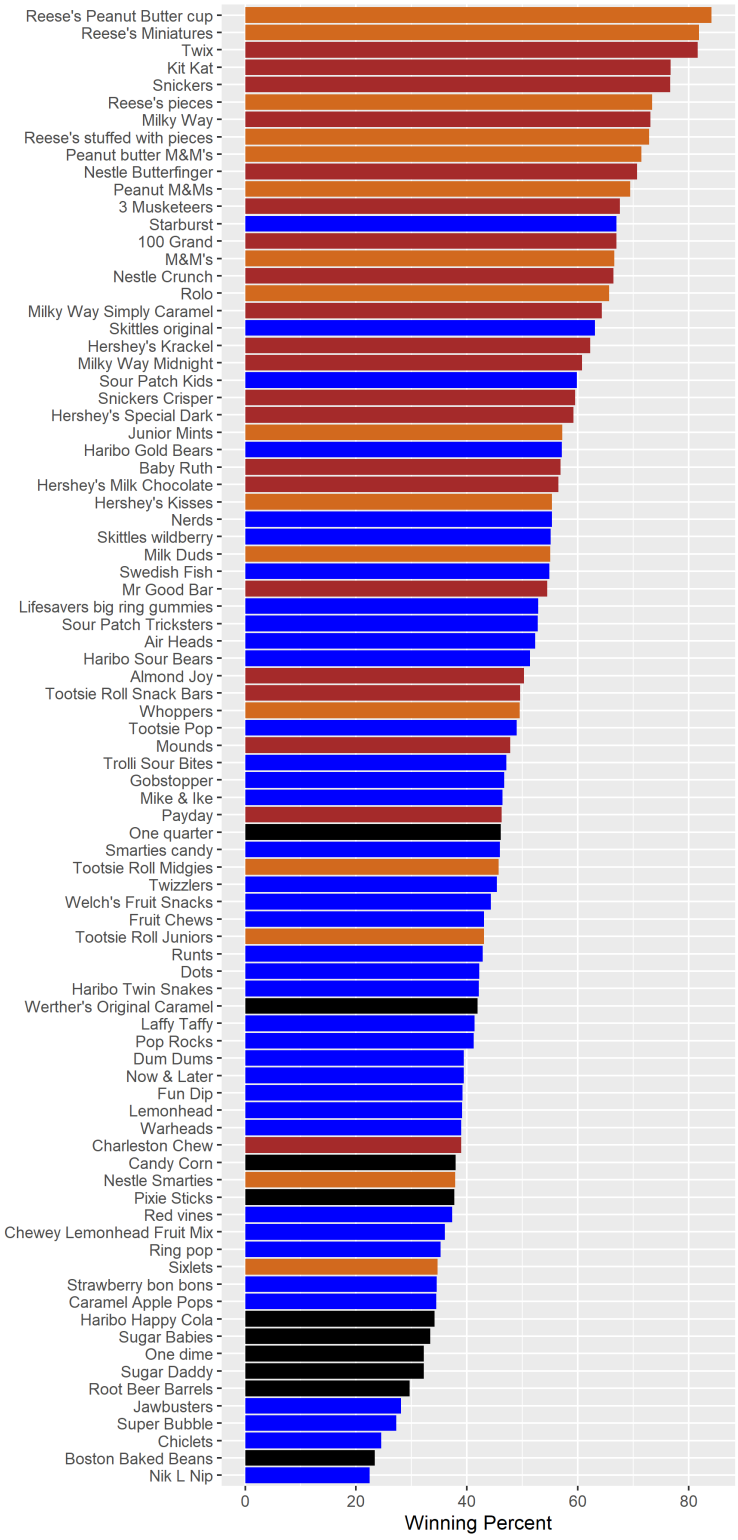

```
mycols <- rep("black", nrow(candy))
mycols[candy$chocolate==1] <- "chocolate"
mycols[candy$bar==1] <- "brown"
mycols[candy$fruity==1] <- "blue"
mycols
```

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

```
ggplot(candy) +
aes(x=winpercent,
    y=reorder(rownames(candy), winpercent))+
  #we put it in geom code becuase its not comming from the dataset
  geom_col(fill=mycols) +
  ylab("") +
  xlab("Winning Percent")
```




```
ggsave("my_color_plot.png", height=12, width=6)
```


Reese's Peanut Butter cup
Reese's Miniatures
Twix
Kit Kat
Snickers
Reese's pieces
Milky Way
Reese's stuffed with pieces
Peanut butter M&M's
Nestle Butterfinger
Peanut M&Ms
3 Musketeers
Starburst
100 Grand
M&M's
Nestle Crunch
Rolo
Milky Way Simply Caramel
Skittles original
Hershey's Krackel
Milky Way Midnight
Sour Patch Kids
Snickers Crisper
Hershey's Special Dark
Junior Mints
Haribo Gold Bears
Baby Ruth
Hershey's Milk Chocolate
Hershey's Kisses
Nerds
Skittles wildberry
Milk Duds
Swedish Fish
Mr Good Bar
Lifesavers big ring gummies
Sour Patch Tricksters
Air Heads
Haribo Sour Bears
Almond Joy
Tootsie Roll Snack Bars
Whoppers
Tootsie Pop
Mounds
Trolli Sour Bites
Gobstopper
Mike & Ike
Payday
One quarter
Smarties candy
Tootsie Roll Midgies
Twizzlers
Welch's Fruit Snacks
Fruit Chews
Tootsie Roll Juniors
Runts
Dots
Haribo Twin Snakes
Werther's Original Caramel
Laffy Taffy
Pop Rocks
Dum Dums
Now & Later
Fun Dip
Lemonhead
Warheads
Charleston Chew
Candy Corn
Nestle Smarties
Pixie Sticks
Red vines
Chewey Lemonhead Fruit Mix
Ring pop
Sixlets
Strawberry bon bons
Caramel Apple Pops
Haribo Happy Cola
Sugar Babies
One dime
Sugar Daddy
Root Beer Barrels
Jawbusters
Super Bubble
Chiclets
Boston Baked Beans
Nik L Nip
0
20
40
60
80
Winning Percent

Q17. What is the worst ranked chocolate candy?

sixlets

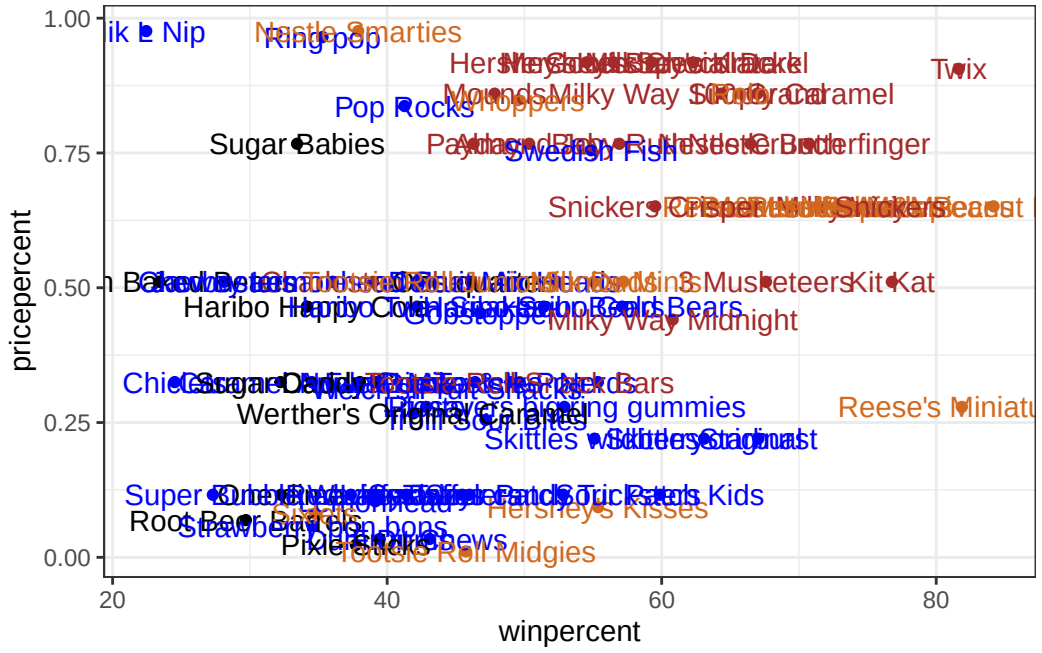
Q18. What is the best ranked fruity candy?

starbursts

##4 Taking a look at pricepercent

Plot of winpercent vs pricepercent

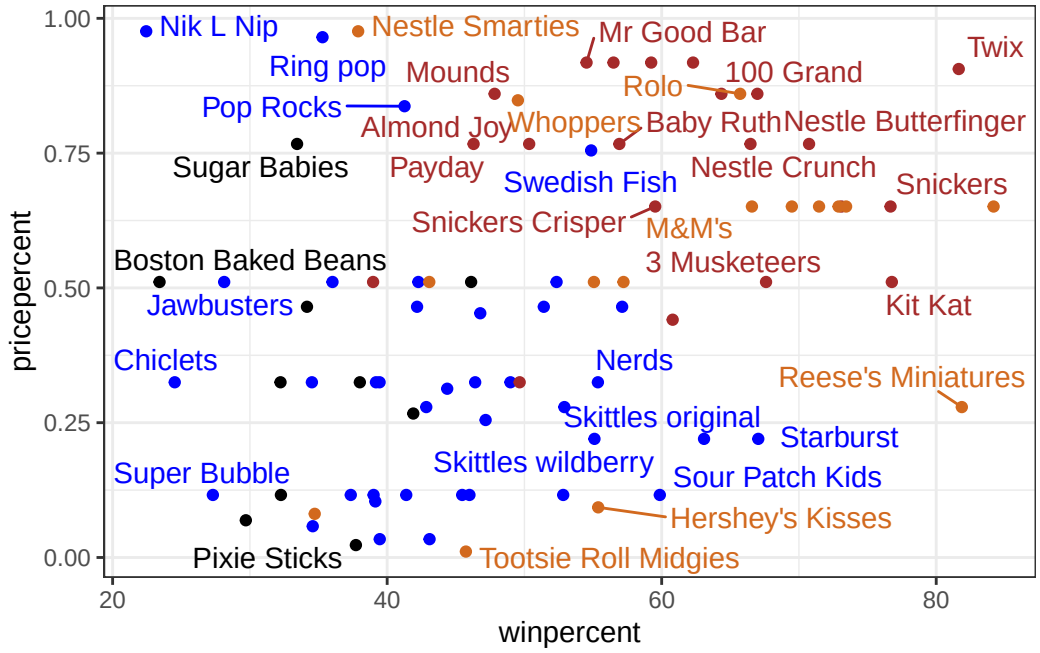
```
ggplot(candy) +
aes(x=winpercent,
    y=pricepercent, label=row.names(candy))+
  geom_point(col=mycols) +
  geom_text(col=mycols) +
  theme_bw()
```



To avoid the common problem of label ir text overlapping/over-plotting we can use the **ggrepel** package like so:

```
library(ggrepel)

ggplot(candy) +
aes(x=winpercent,
    y=pricepercent, label=row.names(candy))+
  geom_point(col=mycols) +
  geom_text_repel(col=mycols) +
  theme_bw()
```

```
Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

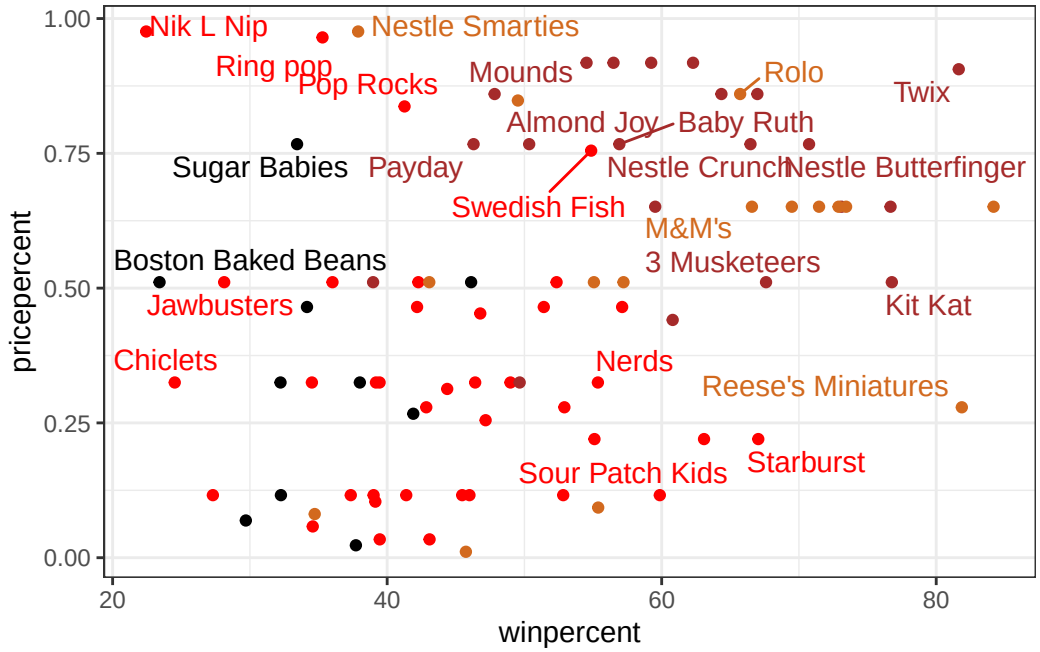


We can control the amount of labels visible by setting different `max.overlaps` values:

```
#Change pink to be red for fruity candy
mycols[candy$fruity==1] <- "red"

ggplot(candy) +
aes(x=winpercent,
    y=pricepercent, label=row.names(candy))+
  geom_point(col=mycols) +
```

```
  geom_text_repel(col=mycols, max.overlaps = 8) +
  theme_bw()
```

```
Warning: ggrepel: 61 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

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

Top 5 most expensive: Twix, Nestle smarties, Nik n Lip, Ring pop, Mr. Good bar

##5 Exploring the correlation strucuture

The main function for correlation analysis in base R is called `cor()`

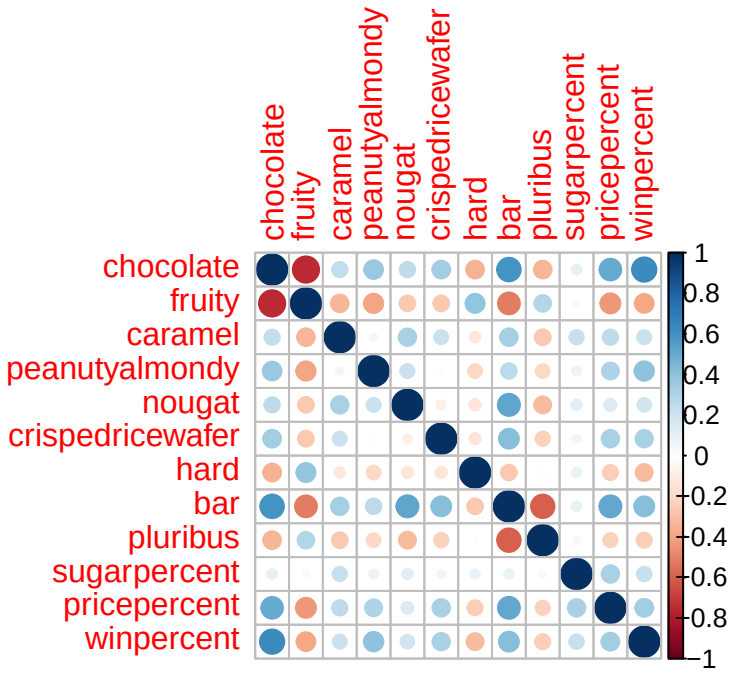
```
cij <- cor(candy)
head(cij)
```

```
                 chocolate     fruity     caramel peanutyalmondy      nougat
chocolate        1.0000000 -0.7417211  0.24987535     0.37782357  0.25489183
fruity          -0.7417211  1.0000000 -0.33548538    -0.39928014 -0.26936712
caramel          0.2498753 -0.3354854  1.00000000     0.05935614  0.32849280
peanutyalmondy   0.3778236 -0.3992801  0.05935614     1.00000000  0.21311310
nougat           0.2548918 -0.2693671  0.32849280     0.21311310  1.00000000
crispedricewafer 0.3412098 -0.2693671  0.21311310    -0.01764631 -0.08974359
                crispedricewafer       hard        bar   pluribus sugarpercent
chocolate             0.34120978 -0.3441769  0.5974211 -0.3396752   0.10416906
fruity               -0.26936712  0.3906775 -0.5150656  0.2997252  -0.03439296
caramel               0.21311310 -0.1223551  0.3339600 -0.2695850   0.22193335
peanutyalmondy       -0.01764631 -0.2055566  0.2604196 -0.2061093   0.08788927
nougat               -0.08974359 -0.1386750  0.5229764 -0.3103388   0.12308135
crispedricewafer      1.00000000 -0.1386750  0.4237509 -0.2246934   0.06994969
                pricepercent winpercent
chocolate          0.5046754  0.6365167
fruity            -0.4309685 -0.3809381
caramel            0.2543271  0.2134163
peanutyalmondy     0.3091532  0.4061922
nougat             0.1531964  0.1993753
crispedricewafer   0.3282654  0.3246797
```

```
library(corrplot)
```

```
corrplot 0.95 loaded
```

```
corrplot(cij)
```



##6 Principal COmponent Analysis (PCA)

We can use our old friend `prcomp()` function with `scale=T`:

```
pca <- prcomp(candy, scale=T)
pca
```

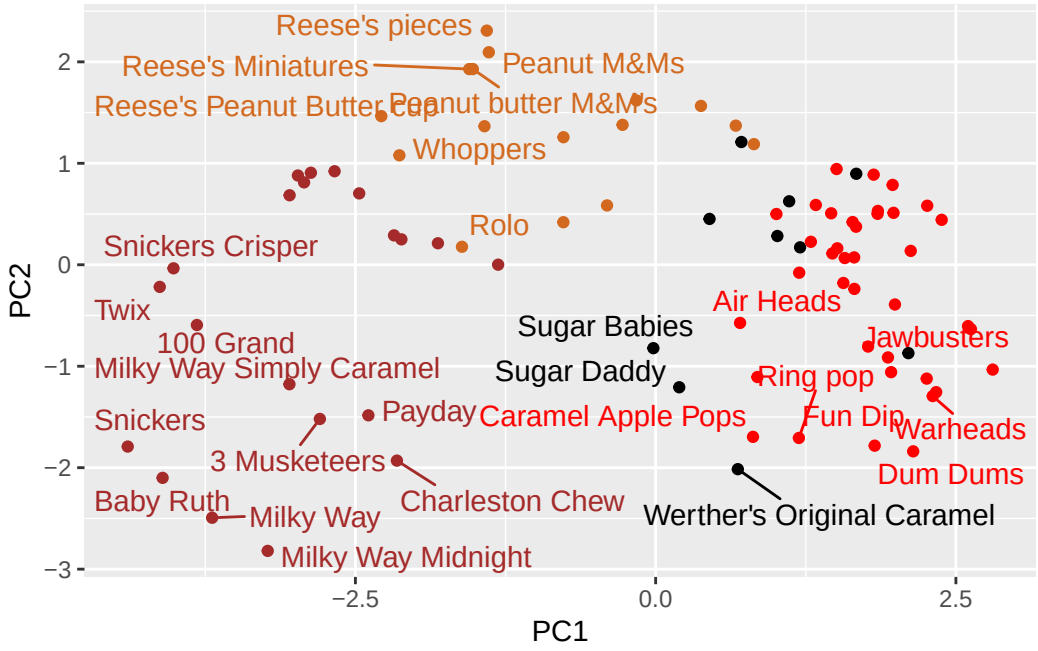
```
Standard deviations (1, .., p=12):
 [1] 2.0787503 1.1378302 1.1091894 1.0753337 0.9518204 0.8192321 0.8153014
 [8] 0.7452991 0.6782391 0.6234867 0.4397418 0.3976039

Rotation (n x k) = (12 x 12):
                          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
```

Let's make our main results figures, first our score plot (PC plot)

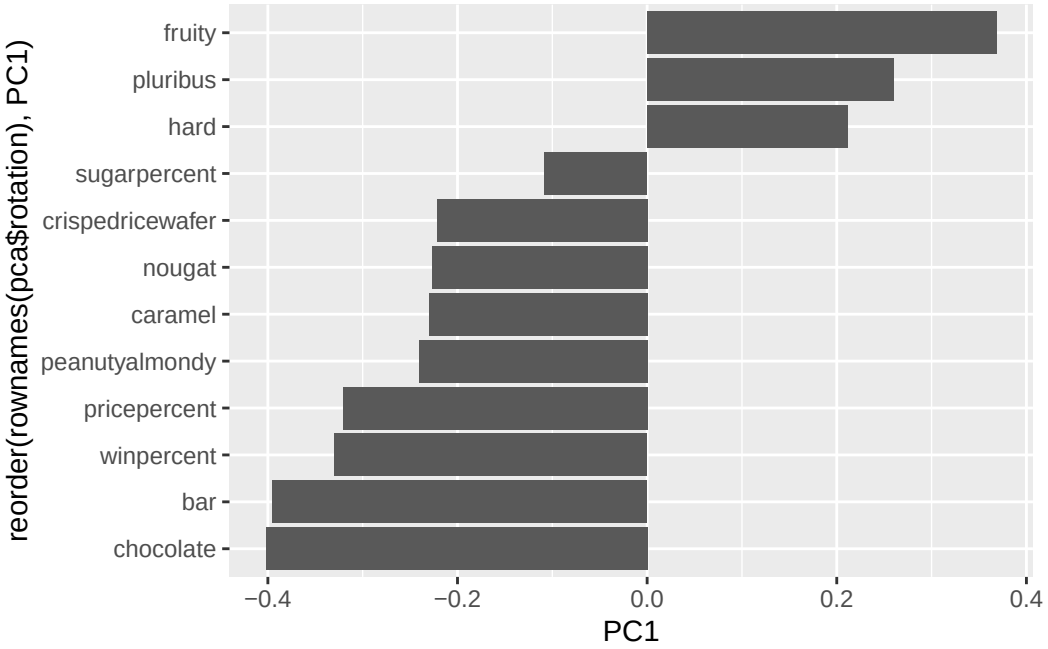
```
#attributes(pca)
#pca$x

ggplot(pca$x) +
  aes(PC1, PC2, label=row.names(candy))+
  geom_point(col=mycols) +
  geom_text_repel(col=mycols, max.overlaps = 8)
```

```
Warning: ggrepel: 57 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Let's look at how the original variables contribute to our new PC's - this is often called the valiable "loadings"

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


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