Class 10: Halloween Mini-Project

Youn Soo Na (PID: A17014731)

Import candy data

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

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

> Q1. How many different candy types are in this dataset?

```
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

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

```r
candy["Haribo Gold Bears", ]$winpercent
```

```
[1] 57.11974
```

```r
# candy %>%
#   filter(rownames(candy)=="Haribo Gold Bears") %>%
#   select(winpercent)
```

Q3a. Find fruity candy with a winpercent above 50%

```r
library(dplyr)
```

```


Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
candy %>%
  filter(winpercent >= 50) %>%
  filter(fruity==1)
```

```
                            chocolate fruity caramel peanutyalmondy nougat
Air Heads                           0      1       0              0      0
Haribo Gold Bears                   0      1       0              0      0
Haribo Sour Bears                   0      1       0              0      0
Lifesavers big ring gummies         0      1       0              0      0
Nerds                               0      1       0              0      0
Skittles original                   0      1       0              0      0
Skittles wildberry                  0      1       0              0      0
Sour Patch Kids                     0      1       0              0      0
Sour Patch Tricksters               0      1       0              0      0
```

```
Starburst                           0      1       0              0      0
Swedish Fish                        0      1       0              0      0
                            crispedricewafer hard bar pluribus sugarpercent
Air Heads                                  0    0   0        0        0.906
Haribo Gold Bears                          0    0   0        1        0.465
Haribo Sour Bears                          0    0   0        1        0.465
Lifesavers big ring gummies                0    0   0        0        0.267
Nerds                                      0    1   0        1        0.848
Skittles original                          0    0   0        1        0.941
Skittles wildberry                         0    0   0        1        0.941
Sour Patch Kids                            0    0   0        1        0.069
Sour Patch Tricksters                      0    0   0        1        0.069
Starburst                                  0    0   0        1        0.151
Swedish Fish                               0    0   0        1        0.604
                            pricepercent winpercent
Air Heads                          0.511   52.34146
Haribo Gold Bears                  0.465   57.11974
Haribo Sour Bears                  0.465   51.41243
Lifesavers big ring gummies        0.279   52.91139
Nerds                              0.325   55.35405
Skittles original                  0.220   63.08514
Skittles wildberry                 0.220   55.10370
Sour Patch Kids                    0.116   59.86400
Sour Patch Tricksters              0.116   52.82595
Starburst                          0.220   67.03763
Swedish Fish                       0.755   54.86111
```

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

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

```
[1] 76.7686
```

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

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

```
[1] 49.6535
```

Side-note: the skimr::`skim()` function

```r
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?

Yes, winpercent looks to be on a different scale to the majority of the other columns in the dataset. This requires me to scale my data before doing any analysis like PCA etc.

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

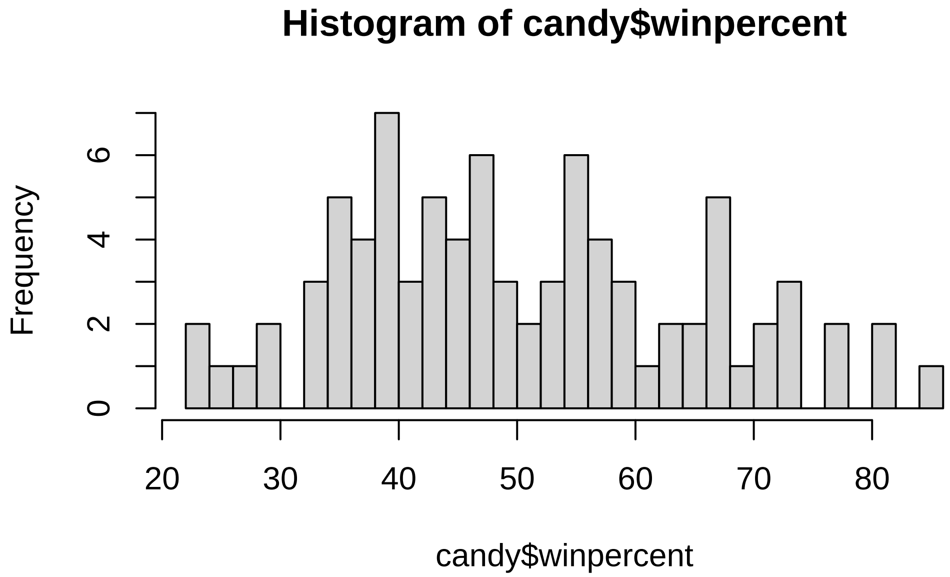
```
candy$chocolate
```

```
 [1] 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 0 0 1 1 0 1 1 1
[39] 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1
[77] 1 1 0 1 0 0 0 0 1
```

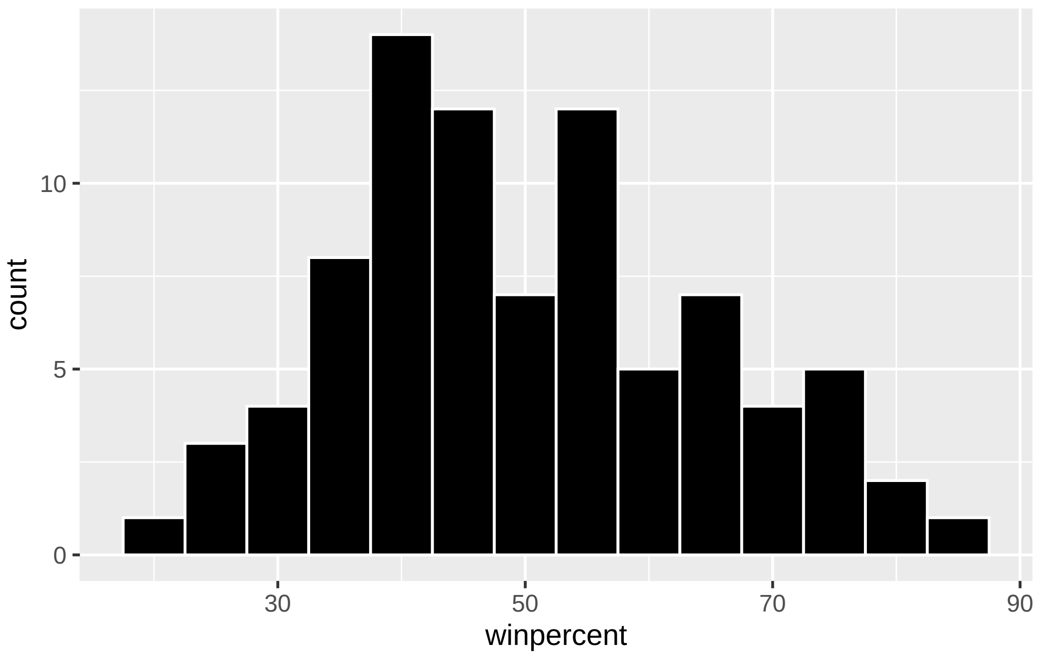
It represents T or F in a logical variable.

> Q8. Plot a histogram of winpercent values

```
hist(candy$winpercent, breaks=30)
```



```
library(ggplot2)

ggplot(candy, aes(winpercent,)) +
         geom_histogram(fill="black", color="white", binwidth = 5)
```



Q9. Is the distribution of winpercent values symmetrical?

NO

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

```
summary(candy$winpercent)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  22.45   39.14   47.83   50.32   59.86   84.18 
```

The center of the distribution, also shown as the median, is below 50%.

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

```
fruit_candy <- candy %>%
  filter(fruity==1)

summary(fruit_candy$winpercent)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  22.45   39.04   42.97   44.12   52.11   67.04 
```

```
chocolate_candy <- candy %>%
  filter(chocolate==1)

summary(chocolate_candy$winpercent)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  34.72   50.35   60.80   60.92   70.74   84.18 
```

```
# base R code for the one above
# summary(candy[as.logical(candy$chocolate),]$winpercent)
```

Chocolate cady is higher ranked than fruit candy, on average.

> Q12. Is this difference statistically significant?

```
t.test(chocolate_candy$winpercent, fruit_candy$winpercent)
```

```

	Welch Two Sample t-test

data:  chocolate_candy$winpercent and fruit_candy$winpercent
t = 6.2582, df = 68.882, p-value = 2.871e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 11.44563 22.15795
sample estimates:
mean of x mean of y 
 60.92153  44.11974 
```

The difference is statistically significant.

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

```
play <- c("d", "a", "c")
sort(play)
```

```
[1] "a" "c" "d"
```

```r
order(play)
```

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

`sort()` organises the output in order, but `order()` is more useful because it specifically shows you the ranking position of the variable (candy)

```r
head( candy[order(candy$winpercent), ], 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
```

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

```r
tail( candy[order(candy$winpercent), ], 5)
```

```
                          chocolate fruity caramel peanutyalmondy nougat
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
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
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
```

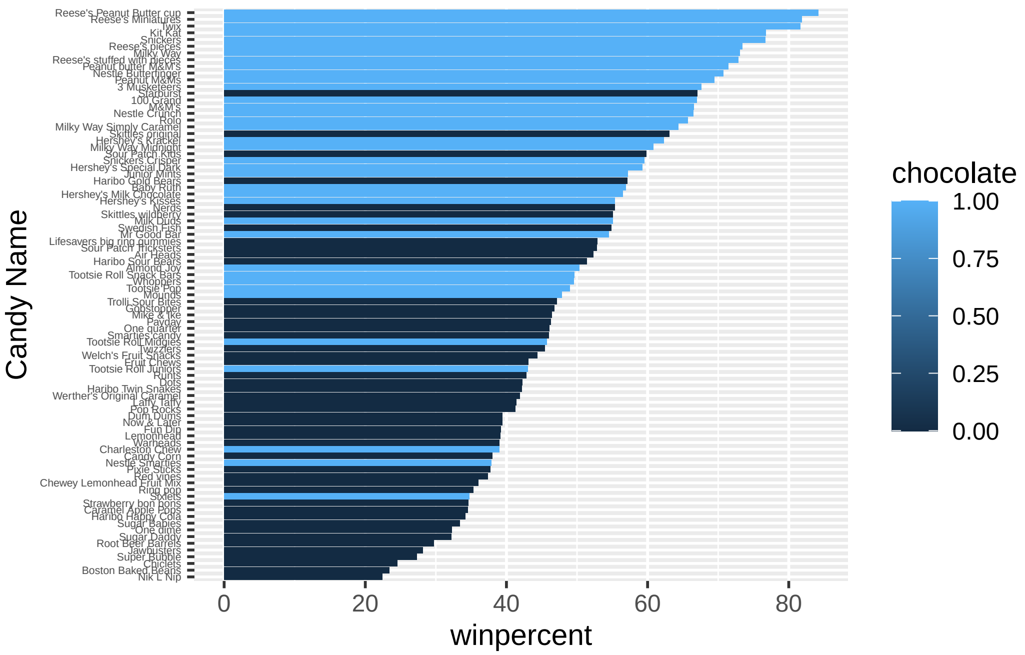
```
head( candy[order(candy$winpercent, decreasing = T), ], 5)
```

```
                          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
```

Let's do a barplot of winpercent values

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

```
ggplot(candy, aes(x = winpercent, y = reorder(row.names(candy), winpercent), fill = chocolate
  geom_col() +
  labs(y="Candy Name") +
  theme(axis.text.y = element_text(size = 4))
```



I want a more custom color scheme where I can see both chocolate and bar and fruity etc. all from the one plot. To do this we can roll our own color vector…

```
library(wesanderson)
# Place holder color vector
# rep()
mycols <- rep("black", nrow(candy))
# I want 2nd entry of my dataset to be blue

mycols[as.logical(candy$bar)] <-  "brown"
mycols[as.logical(candy$chocolate)] <- "chocolate"
mycols[as.logical(candy$fruity)] <-  "pink"



# Use blue for your favorite candy
rownames(candy) == "Haribo Gold Bears"
```

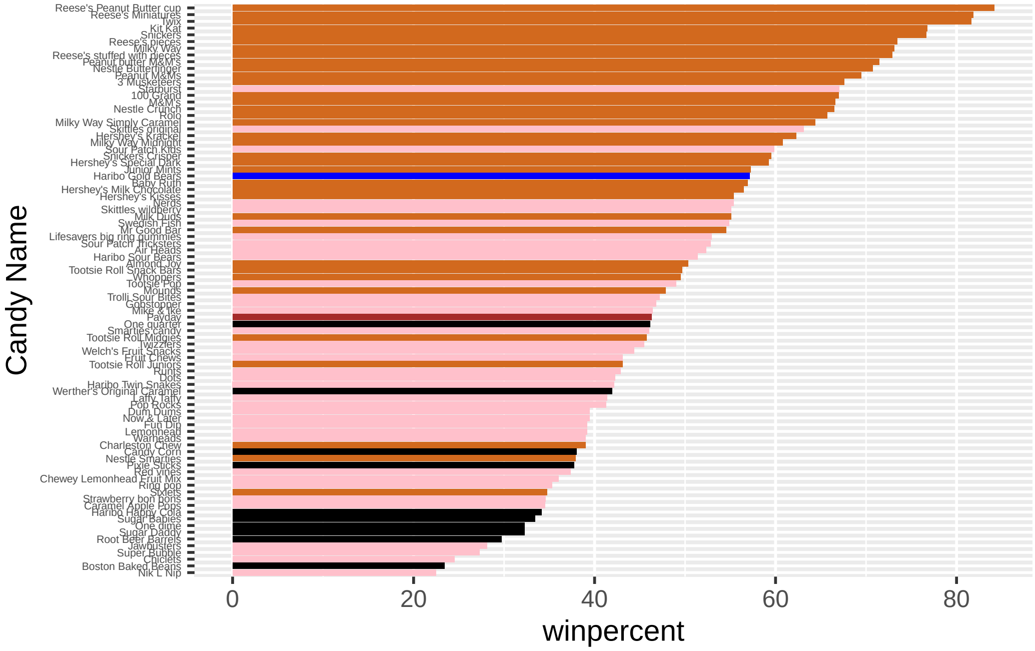
```
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[13] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
[73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[85] FALSE
```

```
mycols[rownames(candy)=="Haribo Gold Bears"] <- "blue"
```

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

```
ggplot(candy,
       aes(x = winpercent,
           y = reorder(rownames(candy),
                       winpercent))) +
  geom_col(fill = mycols) +
  labs(y="Candy Name") +
  theme(axis.text.y = element_text(size = 4))
```



Q17. What is the worst ranked chocolate candy?

Sixlets

Q18. What is the best ranked fruity candy?

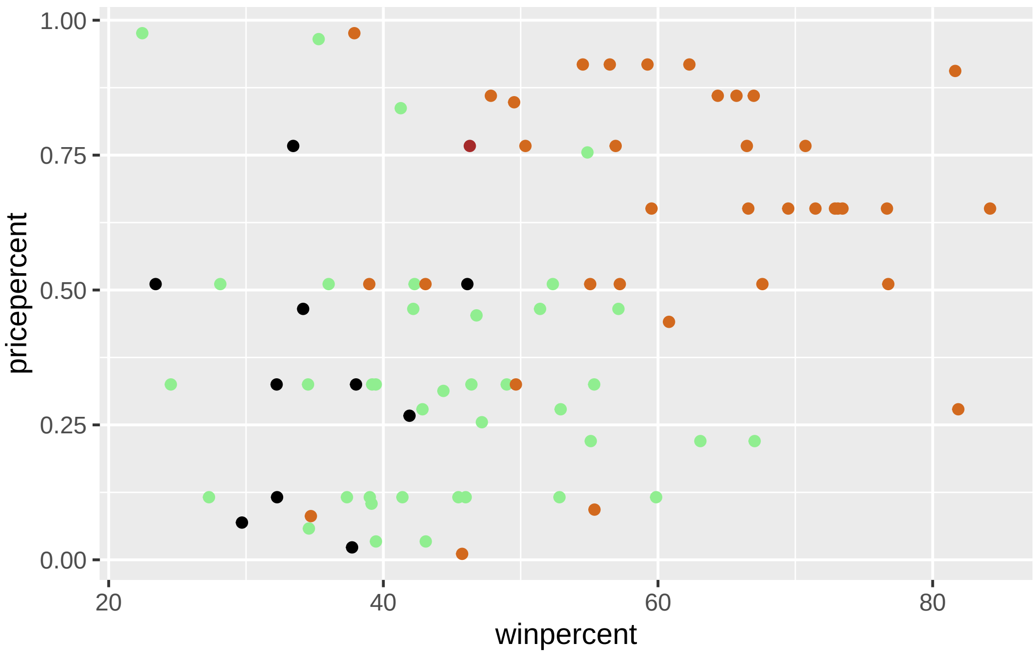
Starbursts

Price Percent

Plot a winpercent vs pricepercent to see what would be the best candy to buy…

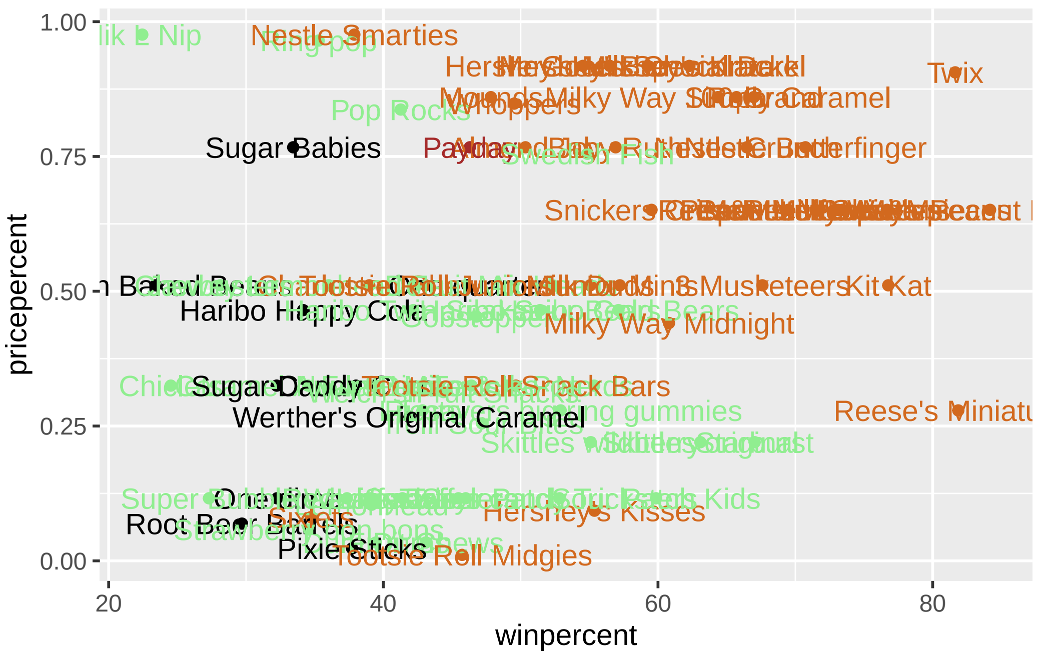
```
mycols[as.logical(candy$fruity)] <- "lightgreen"
```

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



add labels

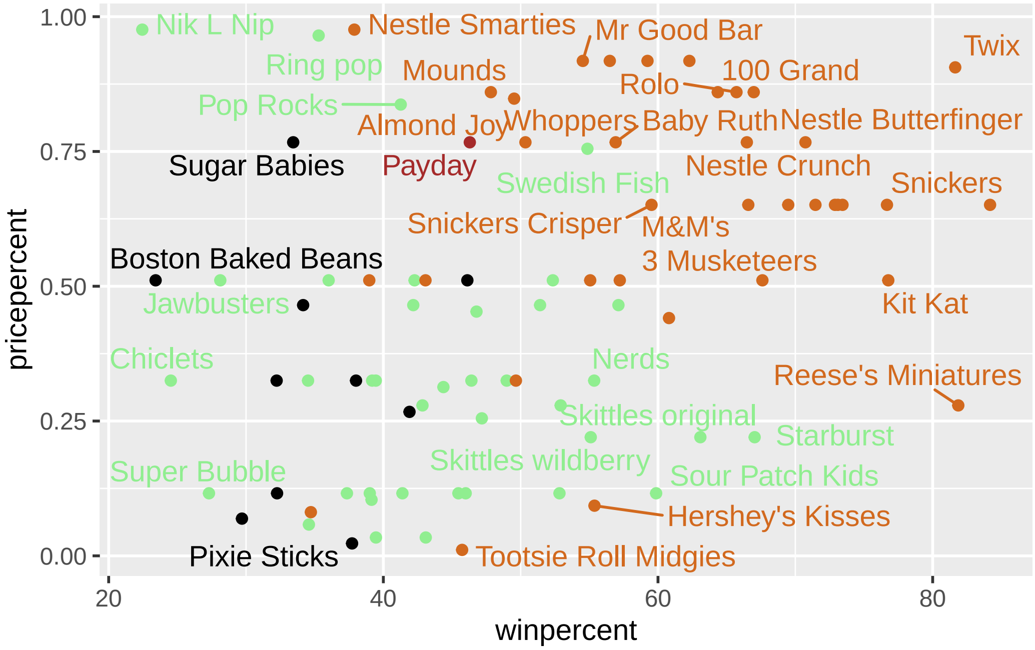
```
ggplot(candy,
       aes(x = winpercent,
           y = pricepercent,
           label = rownames(candy)
           )
       ) +
         geom_point(col = mycols) +
  geom_text(col = mycols)
```



Make the labels non-overlapping

```
library(ggrepel)

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

```
Warning: ggrepel: 50 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 minatures

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

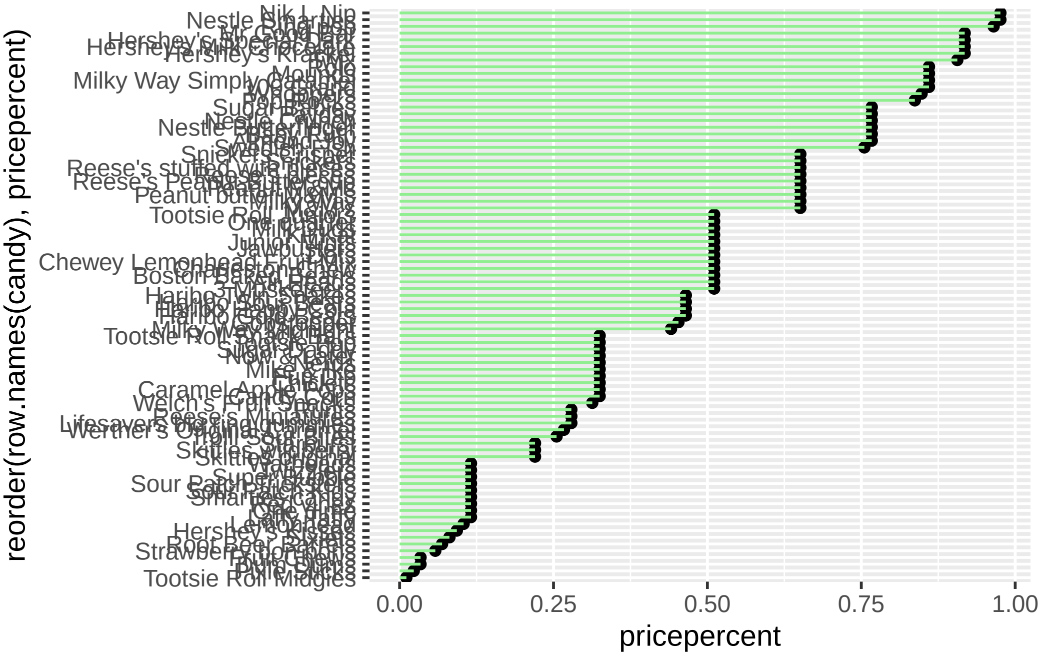
Nik L Nip, Nestle Smarties, Ring Pop, Hershey’s Krackel, Hershey’s Milk Chocolate

```r
ord <- order(candy$pricepercent, decreasing = T)
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
```

> 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().

```r
ggplot(candy, aes(x = pricepercent, y = reorder(row.names(candy), pricepercent))) +
  geom_point()+
  geom_segment(aes(xend = 0, yend = reorder(row.names(candy), pricepercent)),
               color = "lightgreen")
```



Correlation

```r
library(corrplot)
```

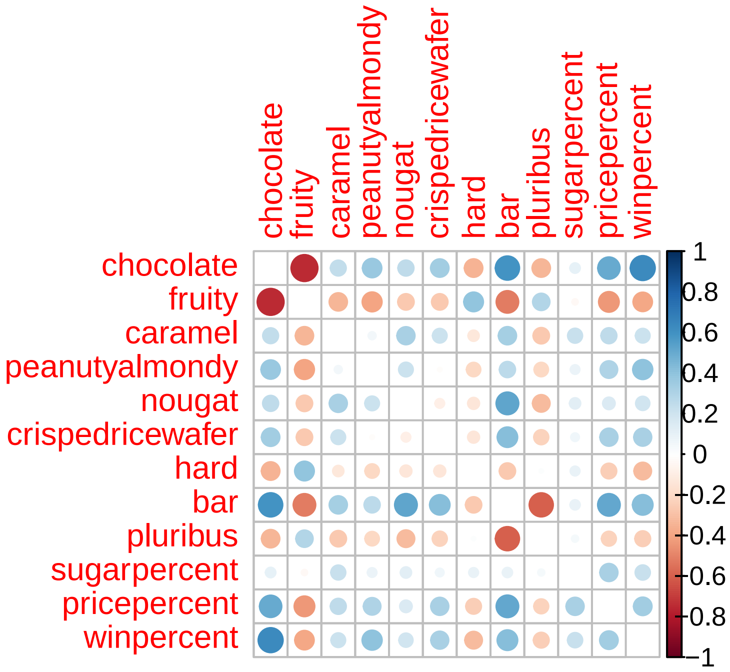
```
corrplot 0.95 loaded
```

```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
```

```r
corrplot(cij, diag = F)
```



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

Chocolate and Fruity

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

Variables that are identical, variable compared to themselves (i.e. Chocolate=Chocolate, fruity=fruity) Other than that, Chocolate and bar or Chocolate and winpercent or bar and pricepercent or bar and nougat

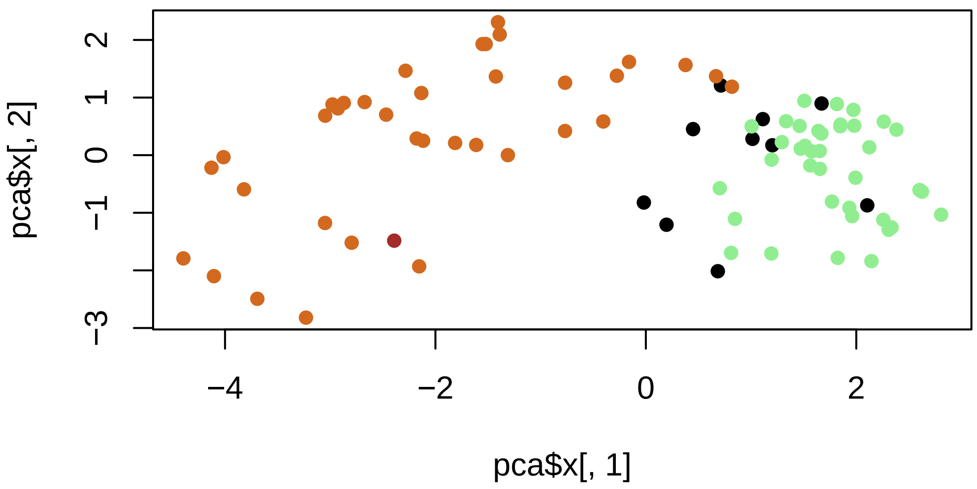
PCA

```
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
```

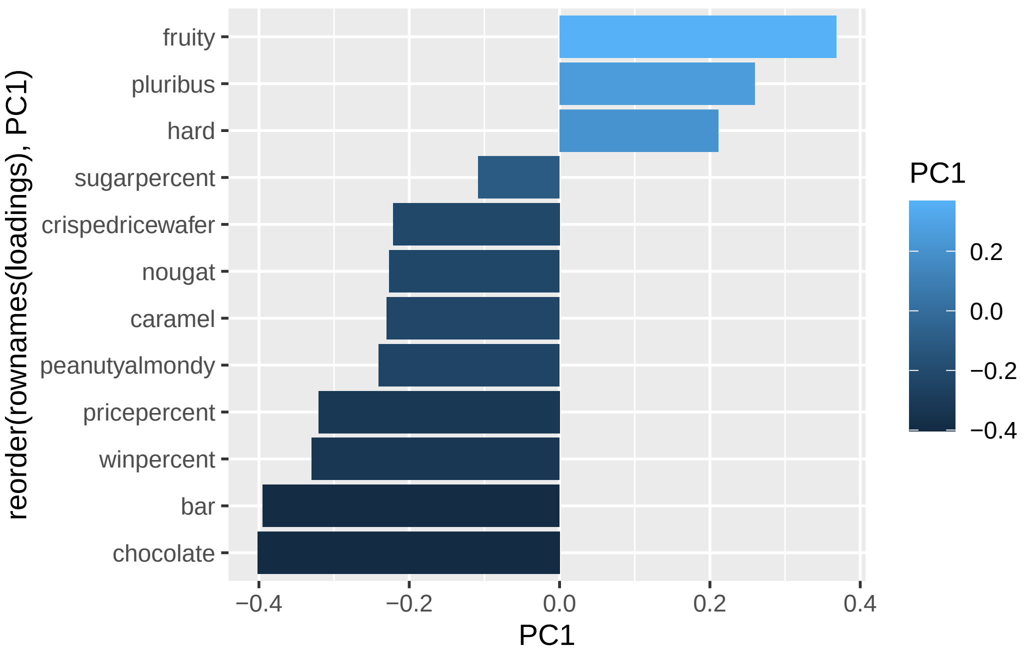
```
Standard deviation     0.74530 0.67824 0.62349 0.43974 0.39760
Proportion of Variance 0.04629 0.03833 0.03239 0.01611 0.01317
Cumulative Proportion  0.89998 0.93832 0.97071 0.98683 1.00000
```

```
plot(pca$x[,1], pca$x[,2], col=mycols, pch=16)
```



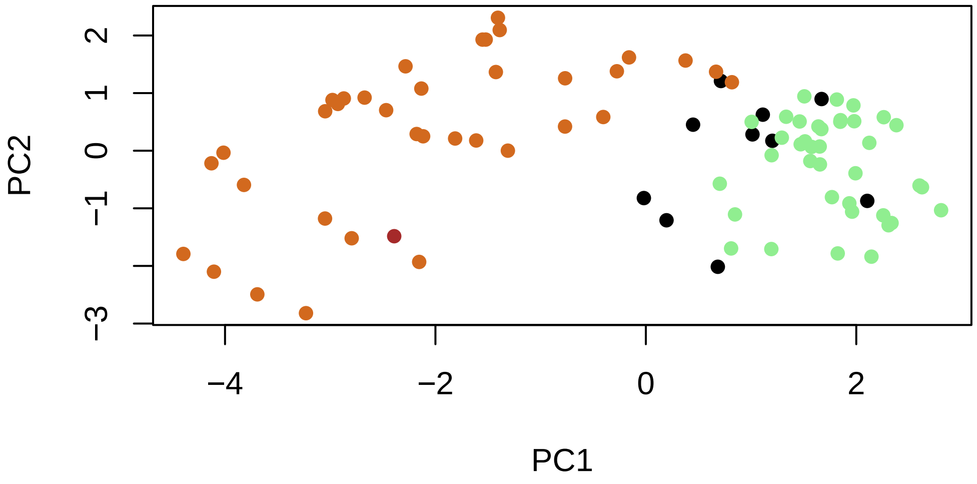
How do the original variable (columns) contribute to the new PCs? I will look at PC1 here:

```
loadings <- as.data.frame(pca$rotation)

ggplot(loadings, aes(x = PC1,
                     y = reorder(rownames(loadings),
                                 PC1),
                     fill = PC1
                     )
       ) +
  geom_col()
```



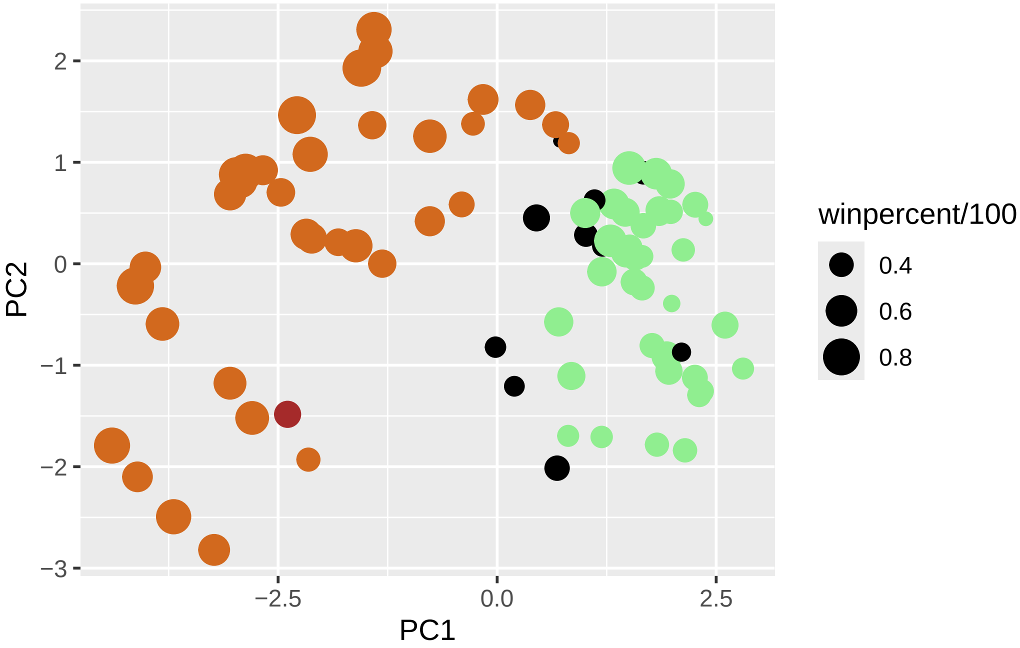
Q24. What original variables are picked up strongly by PC1 in the positive direction? Do these make sense to you? Fruity, pluribus, and hard variables are picked up strongly by PC1 in the positive direction. This makes sense becuase it correlates to the previous correlation graph that told us the three variable, shown in the positive direction, positively correlate with each other. Similarly, variables "chocolate" and "bar" being heavily in the negative direction makes sense since the correlation graph told us that those two variables correlate with each other well.

```
plot(pca$x[,1:2], col=mycols, pch=16)
```



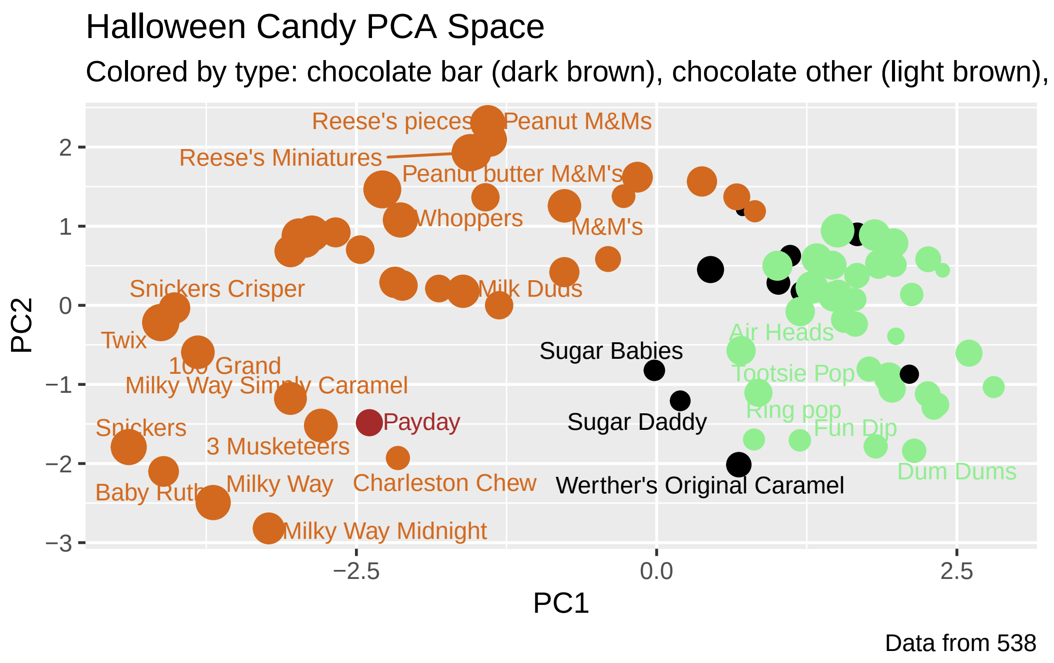
```
my_data <- cbind(candy, pca$x[,1:3])
```

```
p <- ggplot(my_data) +
        aes(x=PC1, y=PC2,
            size=winpercent/100,
            text=rownames(my_data),
            label=rownames(my_data)) +
        geom_point(col=mycols)

p
```



```
library(ggrepel)

p + geom_text_repel(size=3.3, col=mycols, max.overlaps = 7)  +
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
       subtitle="Colored by type: chocolate bar (dark brown), chocolate other (light brown),
       caption="Data from 538")
```

```
Warning: ggrepel: 59 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



```r
library(plotly)
```

```

Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

    last_plot

The following object is masked from 'package:stats':

    filter

The following object is masked from 'package:graphics':

    layout
```

```r
# ggplotly(p)
```