Class09 halloween mini-project

Jordan Prych (PID: A1780226)

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 multivariable dataset.

Importing Candy Data

```
candy_file <- "candy-data.txt"

candy = read.csv(candy_file, row.names=1)
head(candy)
```

```
             chocolate fruity caramel peanutyalmondy nougat crispedricewafer
100 Grand            1      0       1              0      0                 1
3 Musketeers         1      0       0              0      1                 0
One dime             0      0       0              0      0                 0
One quarter          0      0       0              0      0                 0
Air Heads            0      1       0              0      0                 0
Almond Joy           1      0       0              1      0                 0
             hard bar pluribus sugarpercent pricepercent winpercent
100 Grand       0   1        0        0.732        0.860   66.97173
3 Musketeers    0   1        0        0.604        0.511   67.60294
```

```
One dime         0   0         0         0.011         0.116   32.26109
One quarter      0   0         0         0.011         0.511   46.11650
Air Heads        0   0         0         0.906         0.511   52.34146
Almond Joy       0   1         0         0.465         0.767   50.34755
```

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

```
nrow(candy)
```

```
[1] 85
```

There are 85 different candy types in this dataset.

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

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

```
[1] 38
```

There are 38 fruity candy types.

How many chocolate candy are there in the dataset?

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

```
[1] 37
```

There are 37 chocolate candy types.

What is your Favorite Candy?

For a given candy, `winpercent` is the percentage of people who prefer this candy over another randomly chosen candy in the dataset.

We can find the `winpercent` value by using the candy name to access the corresponding row in the dataset.

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

```
[1] 81.64291
```

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

```
candy["Air Heads", ]$winpercent
```

```
[1] 52.34146
```

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

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

```
[1] 76.7686
```

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

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

```
[1] 49.6535
```

Side Note: the skim()‘ function in the **skimr** package can help give you a quick overview of a given dataset.

Let's install in the package using `install.packages("skimr")`and load the package using `library()`:

```
library("skimr")
skim(candy)
```

Table 1: Data summary

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

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
chocolate	0	1	0.44	0.50	0.00	0.00	0.00	1.00	1.00	
fruity	0	1	0.45	0.50	0.00	0.00	0.00	1.00	1.00	
caramel	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
peanutyalmondy	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
nougat	0	1	0.08	0.28	0.00	0.00	0.00	0.00	1.00	
crispedricewafer	0	1	0.08	0.28	0.00	0.00	0.00	0.00	1.00	
hard	0	1	0.18	0.38	0.00	0.00	0.00	0.00	1.00	
bar	0	1	0.25	0.43	0.00	0.00	0.00	0.00	1.00	
pluribus	0	1	0.52	0.50	0.00	0.00	1.00	1.00	1.00	
sugarpercent	0	1	0.48	0.28	0.01	0.22	0.47	0.73	0.99	
pricepercent	0	1	0.47	0.29	0.01	0.26	0.47	0.65	0.98	
winpercent	0	1	50.32	14.71	22.45	39.14	47.83	59.86	84.18	

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

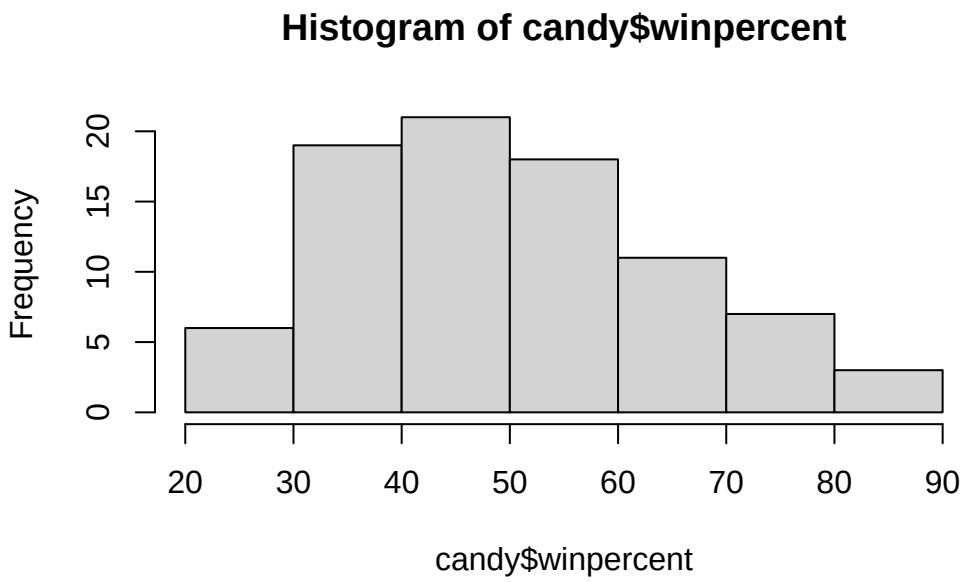
The `winpercent` column looks to be different than the other columns because all the values are higher than the other columns. The scale of the data is different in this column than the others(1-100% rather than 0-1), so therefore you must scale the data before you run a PCA because if not, the PCA will be dominated by this column.

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

a zero indicates that the candy is not a chocolate type and a 1 indicated that the candy is a chocolate type.

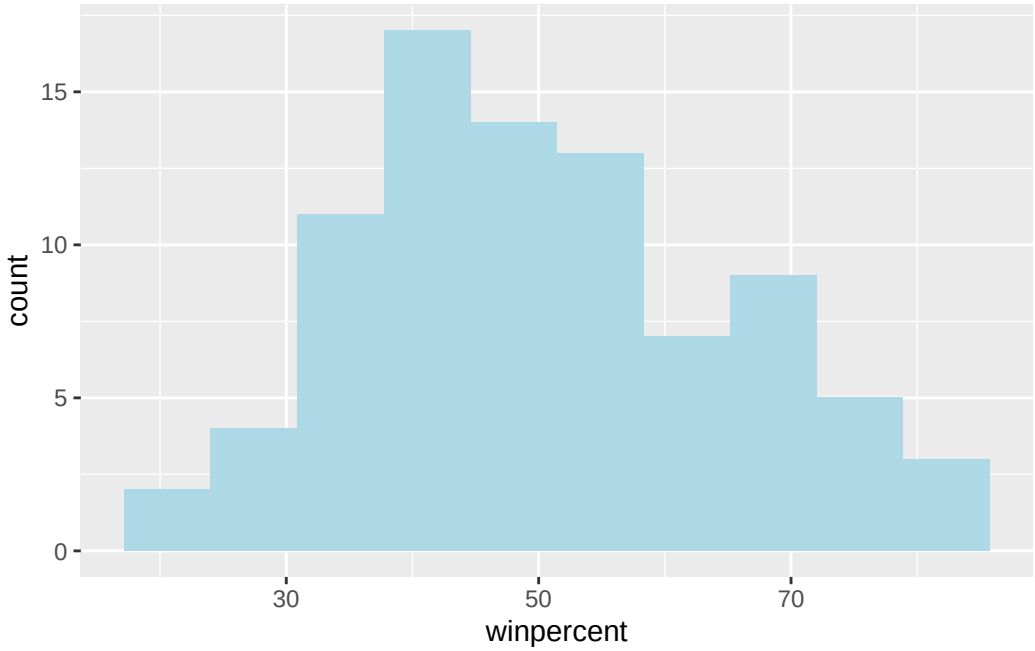
Q8. Plot a histogram of winpercent values.

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



with ggplot:

```
library(ggplot2)

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



Q9. Is the distribution of winpercent values symmetrical?

No

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

From the histogram, we see the center 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 their values

-step 4. find all “fruity” candies

-step 5. find their “winpercent” values

-step 6. summarize these values

Chocolate

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

#Step 2
choc.win <- candy[choc.inds,]$winpercent

#Step 3
choc.mean <- mean(choc.win)
```

Fruity

```
#Step 4
fruity.inds <- candy$fruity==1

#Step 5
fruity.win <- candy[fruity.inds, ]$winpercent

#Step 6
fruity.mean <- mean(fruity.win)
```

Chocolate candy has a higher mean winpercent than fruity candy.

```
choc.mean
```

```
[1] 60.92153
```

```
fruity.mean
```

```
[1] 44.11974
```

Q12. Is this difference statistically significant?

```
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
```

p-value on 2.871e^-08 means that this difference is significantly significant.

Overall Candy Rankings

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

```
#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 elements of the input to make them sorted - i.e. how to order them.

We can determine tha order of winpercent to make them sorted and use that order to arrange whole dataset.

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

```
                  chocolate fruity caramel peanutyalmondy nougat
Nik L Nip                 0      1       0              0      0
Boston Baked Beans        0      0       0              1      0
Chiclets                  0      1       0              0      0
Super Bubble              0      1       0              0      0
Jawbusters                0      1       0              0      0
Root Beer Barrels         0      0       0              0      0
                  crispedricewafer hard bar pluribus sugarpercent pricepercent
Nik L Nip                        0    0   0        1        0.197        0.976
Boston Baked Beans               0    0   0        1        0.313        0.511
Chiclets                         0    0   0        1        0.046        0.325
Super Bubble                     0    0   0        0        0.162        0.116
Jawbusters                       0    1   0        1        0.093        0.511
Root Beer Barrels                0    1   0        1        0.732        0.069
                  winpercent
Nik L Nip           22.44534
Boston Baked Beans  23.41782
Chiclets            24.52499
Super Bubble        27.30386
Jawbusters          28.12744
Root Beer Barrels   29.70369
```

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

```
tail(candy[ord.inds, ])
```

```
                        chocolate fruity caramel peanutyalmondy nougat
Reese's pieces                  1      0       0              1      0
Snickers                        1      0       1              1      1
Kit Kat                         1      0       0              0      0
Twix                            1      0       1              0      0
Reese's Miniatures              1      0       0              1      0
Reese's Peanut Butter cup       1      0       0              1      0
```

```
                         crispedricewafer hard bar pluribus sugarpercent
Reese's pieces                          0    0   0        1        0.406
Snickers                                0    0   1        0        0.546
Kit Kat                                 1    0   1        0        0.313
Twix                                    1    0   1        0        0.546
Reese's Miniatures                      0    0   0        0        0.034
Reese's Peanut Butter cup               0    0   0        0        0.720
                         pricepercent winpercent
Reese's pieces                  0.651   73.43499
Snickers                        0.651   76.67378
Kit Kat                         0.511   76.76860
Twix                            0.906   81.64291
Reese's Miniatures              0.279   81.86626
Reese's Peanut Butter cup       0.651   84.18029
```

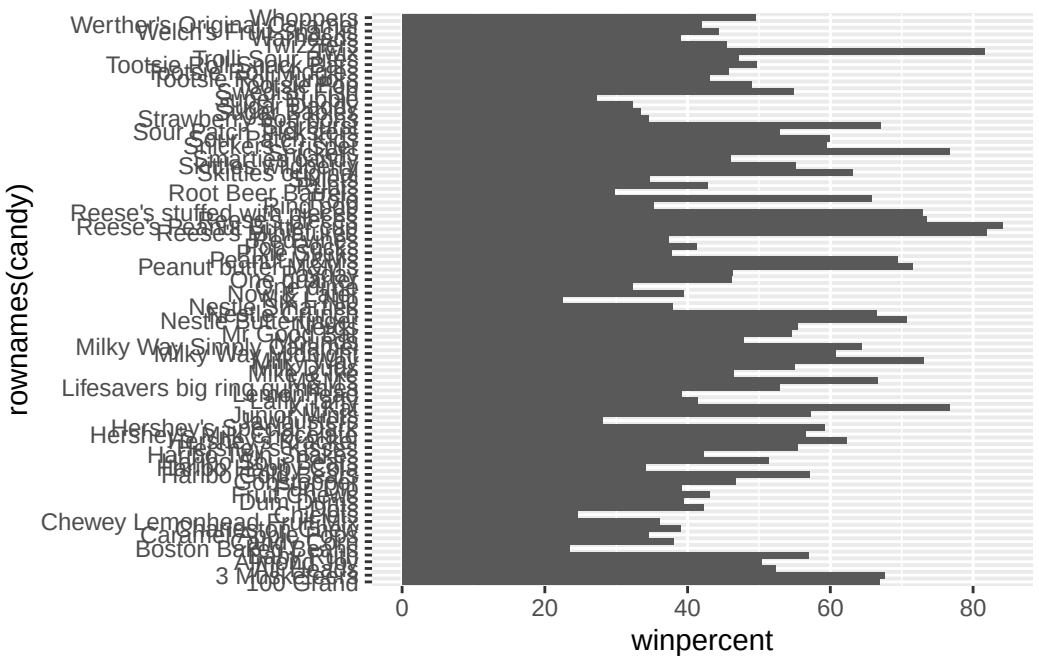
OR...

```
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
```

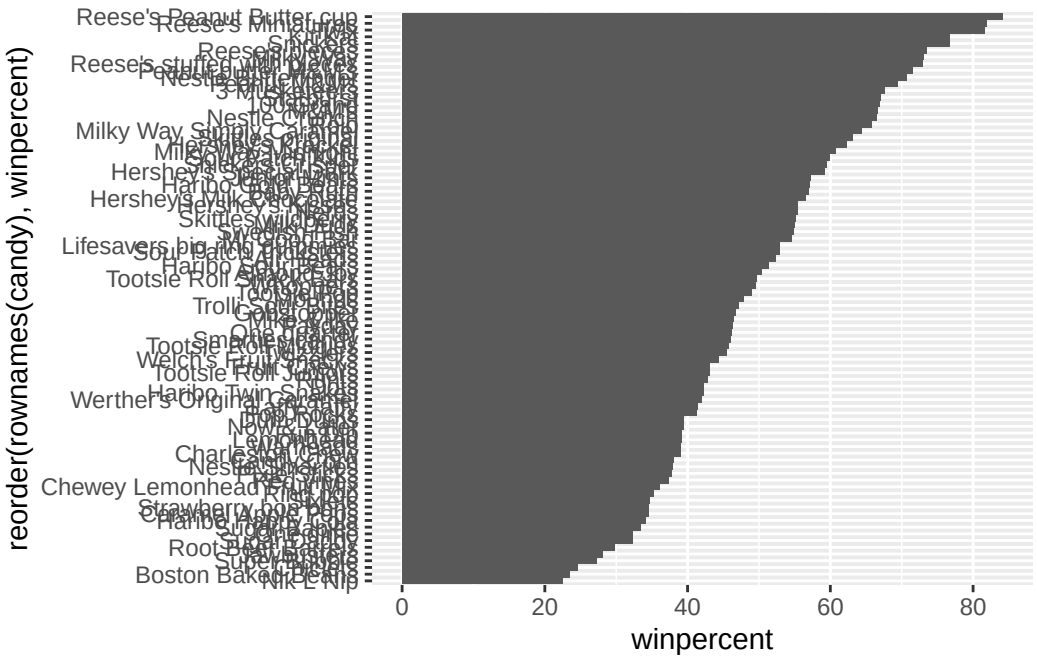
Q15. Make a first barplot of candy ranking based on winpercent values.

```
library(ggplot2)

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



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

```
library(ggplot2)

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

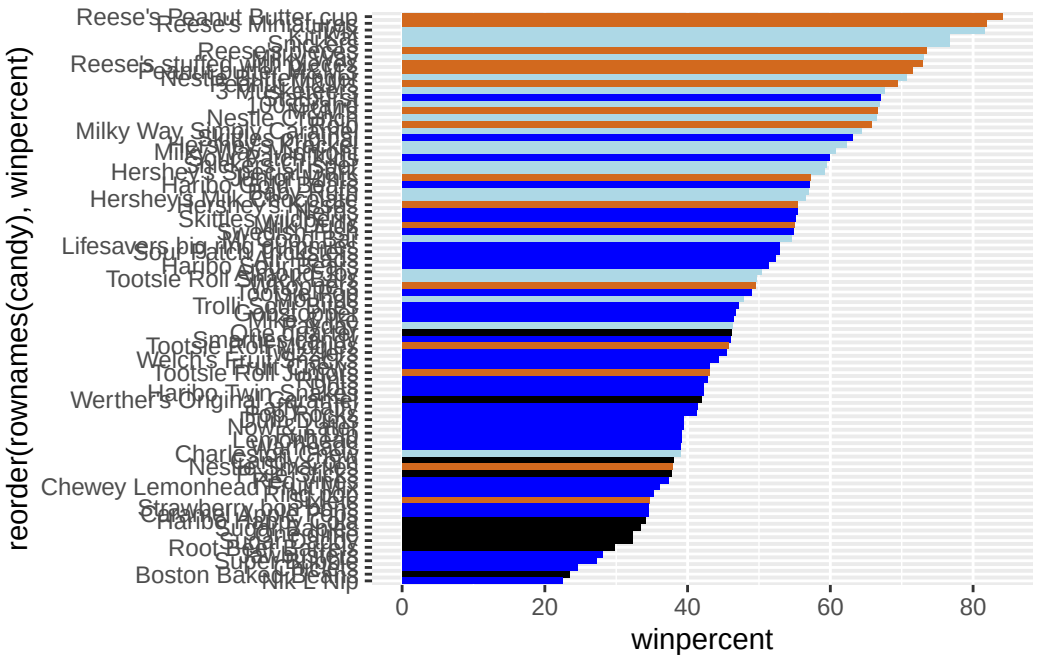


Let's set up a color vector that signifies candy type to use for future plots:

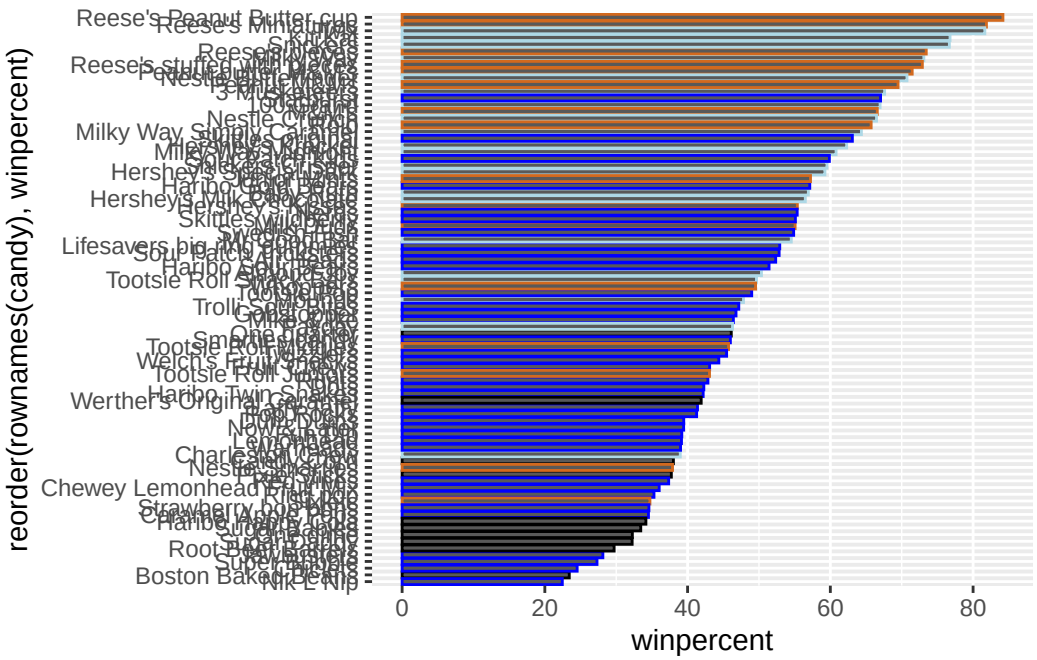
```
my_cols=rep("black", nrow(candy))
my_cols[candy$chocolate==1] = "chocolate"
my_cols[candy$fruity==1] = "blue"
my_cols[candy$bar==1] = "lightblue"
```

Use `fill=my_cols` for `geom_col()`.

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




```
ggplot(candy) +
  aes(winpercent, reorder(rownames(candy),winpercent)) +
  geom_col(col=my_cols)
```



using `col=my_cols` outlines the columns in the color and does not fill the inside.

Q17. What is the worst ranked chocolate candy?

the worst ranked chocolate candy is Sixlets

Q18. What is the best ranked fruity candy?

The best ranked fruity candy is Starburst

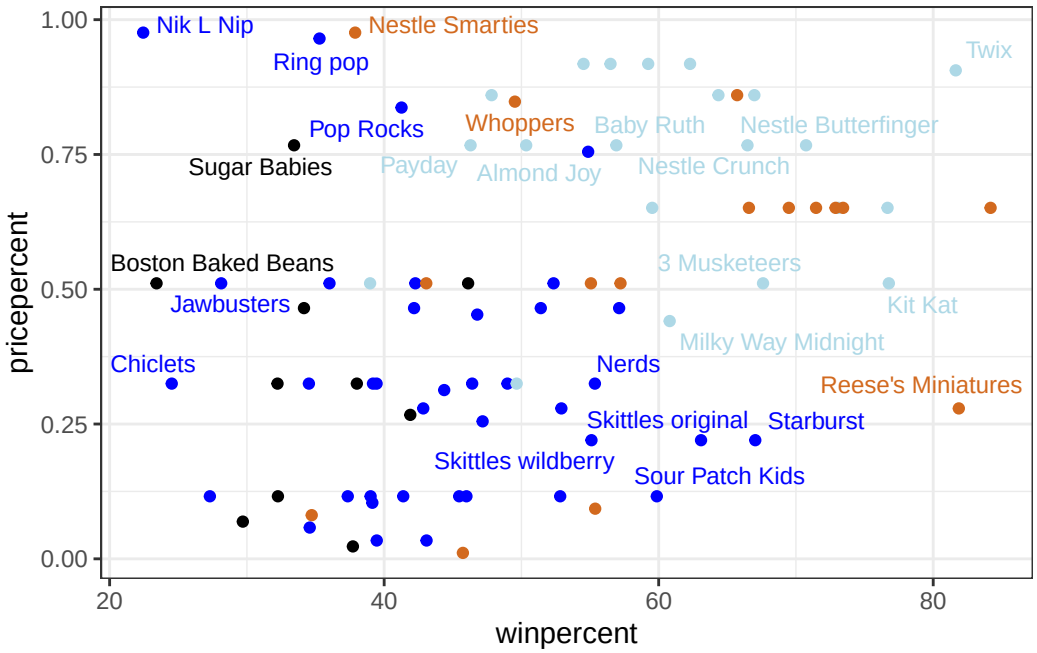
Taking a look at Pricepercent

One way we can get the value for money is to make a plave of `winpercent` vs the `pricepercent` variables. The `pricepercent` variable records the percentile rank of the candy's price against all the other candies in the dataset. Lower values are less expensive and higher values are more expensive.

To avoid the overplotting of the text labels we can use the add on package `ggrepel`. Install the `ggrepel` package using `Install.packages()`.

```
library(ggrepel)

#Plot of price vs. win
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) + geom_point(col=my_cols) + geom_text
```

```
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?

```
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
```

Top 5 most expensive candy types Nik L Nip, Nestle Smarties, Ring pop, Hershey's Krackle, and Hershey's Milk Chocolate. The least popular among these is the Nik L Nips.

Exploring the Correlation Structure

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

First we will use correlation and view the results with the **corrplot** package to plot a correlation matrix.

```
library(corrplot)
```

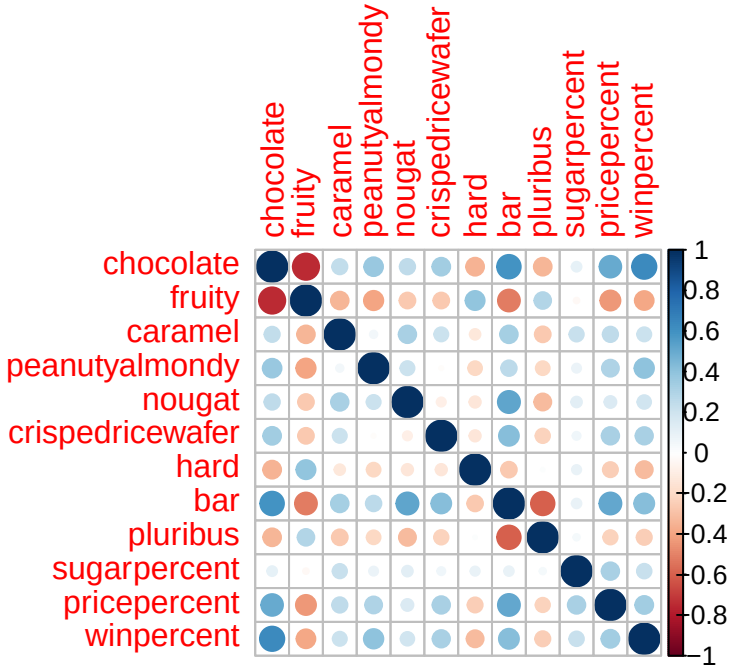
```
corrplot 0.95 loaded
```

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

```
                  chocolate      fruity     caramel peanutyalmondy      nougat
chocolate         1.0000000 -0.74172106  0.24987535     0.37782357  0.25489183
fruity           -0.7417211  1.00000000 -0.33548538    -0.39928014 -0.26936712
caramel           0.2498753 -0.33548538  1.00000000     0.05935614  0.32849280
peanutyalmondy    0.3778236 -0.39928014  0.05935614     1.00000000  0.21311310
nougat            0.2548918 -0.26936712  0.32849280     0.21311310  1.00000000
crispedricewafer  0.3412098 -0.26936712  0.21311310    -0.01764631 -0.08974359
hard             -0.3441769  0.39067750 -0.12235513    -0.20555661 -0.13867505
bar               0.5974211 -0.51506558  0.33396002     0.26041960  0.52297636
pluribus         -0.3396752  0.29972522 -0.26958501    -0.20610932 -0.31033884
sugarpercent      0.1041691 -0.03439296  0.22193335     0.08788927  0.12308135
pricepercent      0.5046754 -0.43096853  0.25432709     0.30915323  0.15319643
winpercent        0.6365167 -0.38093814  0.21341630     0.40619220  0.19937530
                  crispedricewafer        hard         bar    pluribus
chocolate              0.34120978 -0.34417691  0.59742114 -0.33967519
fruity                -0.26936712  0.39067750 -0.51506558  0.29972522
caramel                0.21311310 -0.12235513  0.33396002 -0.26958501
peanutyalmondy        -0.01764631 -0.20555661  0.26041960 -0.20610932
nougat                -0.08974359 -0.13867505  0.52297636 -0.31033884
crispedricewafer       1.00000000 -0.13867505  0.42375093 -0.22469338
hard                  -0.13867505  1.00000000 -0.26516504  0.01453172
bar                    0.42375093 -0.26516504  1.00000000 -0.59340892
pluribus              -0.22469338  0.01453172 -0.59340892  1.00000000
sugarpercent           0.06994969  0.09180975  0.09998516  0.04552282
pricepercent           0.32826539 -0.24436534  0.51840654 -0.22079363
winpercent             0.32467965 -0.31038158  0.42992933 -0.24744787
                  sugarpercent pricepercent winpercent
```

```
chocolate          0.10416906    0.5046754  0.6365167
fruity            -0.03439296   -0.4309685 -0.3809381
caramel            0.22193335    0.2543271  0.2134163
peanutyalmondy     0.08788927    0.3091532  0.4061922
nougat             0.12308135    0.1531964  0.1993753
crispedricewafer   0.06994969    0.3282654  0.3246797
hard               0.09180975   -0.2443653 -0.3103816
bar                0.09998516    0.5184065  0.4299293
pluribus           0.04552282   -0.2207936 -0.2474479
sugarpercent       1.00000000    0.3297064  0.2291507
pricepercent       0.32970639    1.0000000  0.3453254
winpercent         0.22915066    0.3453254  1.0000000
```

```
corrplot(cij)
```



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

fruity and chocolate are anti-correlated. pluribus and bar also have anti-correlation.

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

The chocolate and winpercent are most positively correlated.

Principal Component Analysis

Let's apply PCA using `prcomp()` function to our candy dataset, remembering to set `scale=TRUE` argument.

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

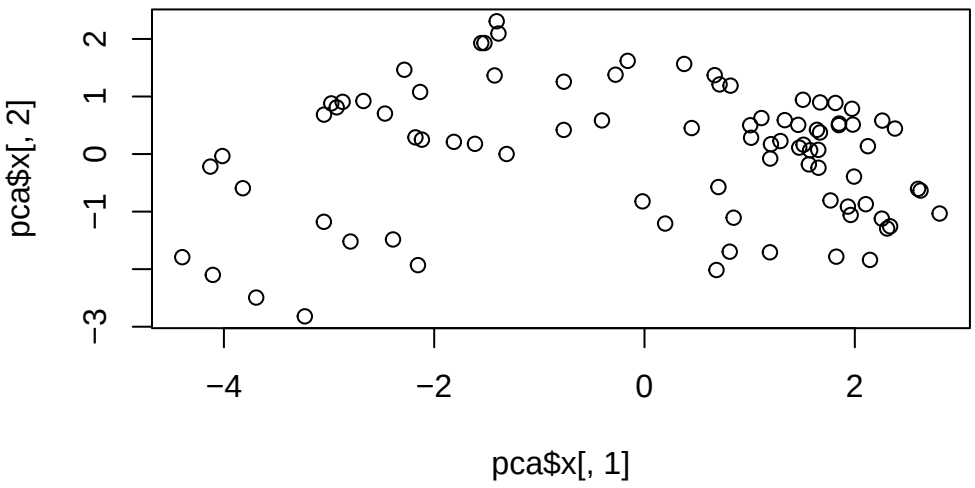
```
Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7
Standard deviation     2.0788 1.1378 1.1092 1.07533 0.9518 0.81923 0.81530
Proportion of Variance 0.3601 0.1079 0.1025 0.09636 0.0755 0.05593 0.05539
Cumulative Proportion  0.3601 0.4680 0.5705 0.66688 0.7424 0.79830 0.85369
                           PC8     PC9    PC10    PC11    PC12
Standard deviation     0.74530 0.67824 0.62349 0.43974 0.39760
Proportion of Variance 0.04629 0.03833 0.03239 0.01611 0.01317
Cumulative Proportion  0.89998 0.93832 0.97071 0.98683 1.00000
```

```
attributes(pca)
```

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

$class
[1] "prcomp"
```

Now we can plot our main PCA score plot of PC1 vs. PC2

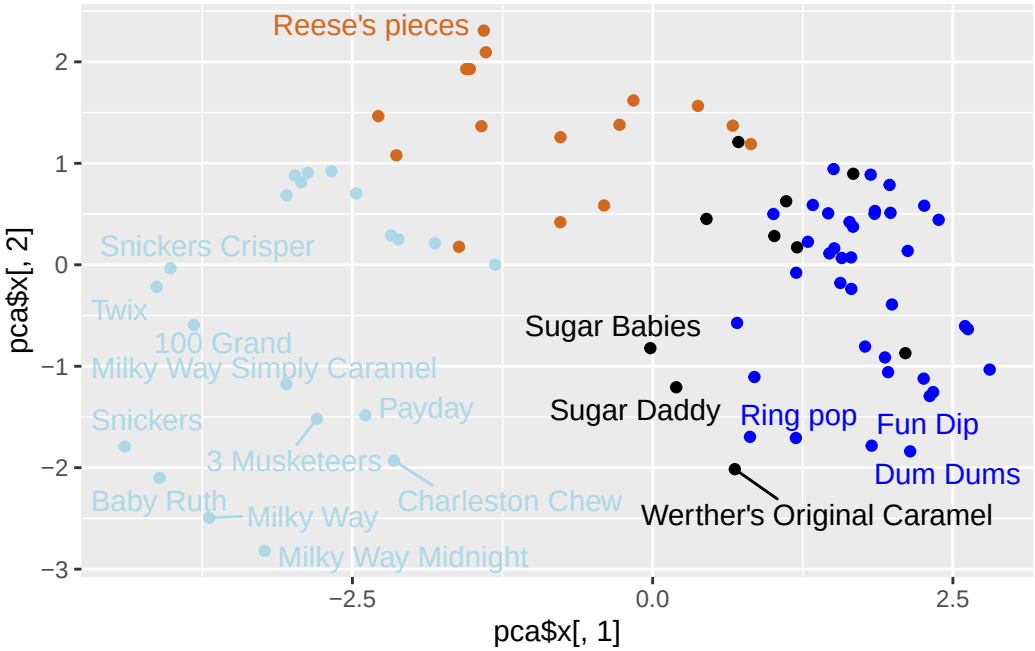
```
plot(pca$x[,1], pca$x[,2])
```



With ggplot:

```
ggplot(candy) + aes(pca$x[,1], pca$x[,2], label=rownames(pca$x)) + geom_point(col=my_cols) +
```

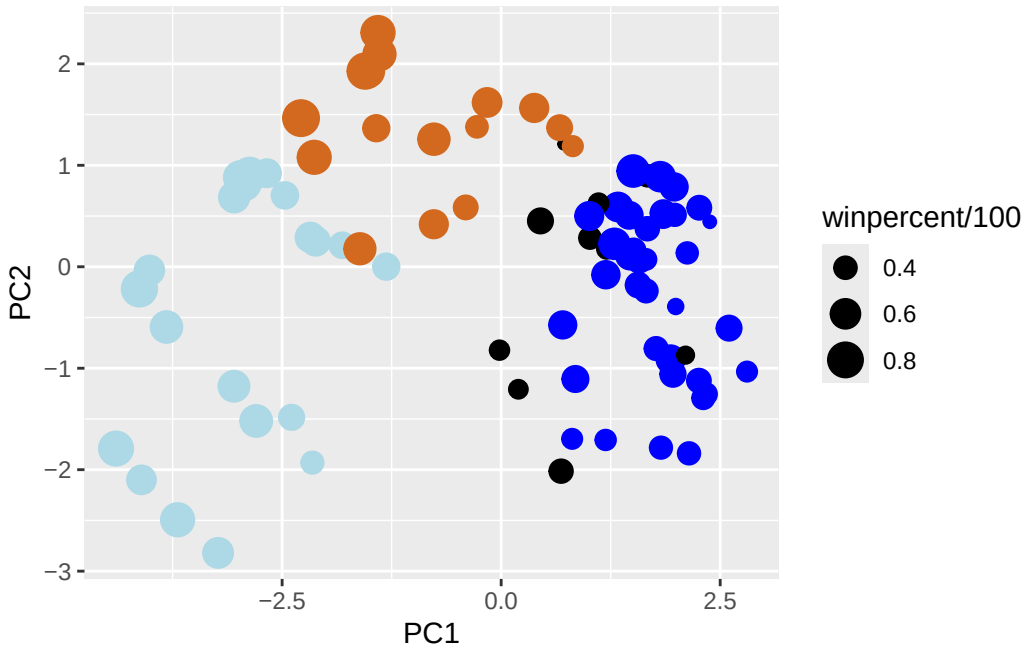
```
Warning: ggrepel: 67 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



ggplot works best when you supply an input data.frame that includes a separate column for each aestetics you would like displayed in your final plot. To acomplish this, we make a new data.frame that contain out PCA results with all the rest of our candy data.

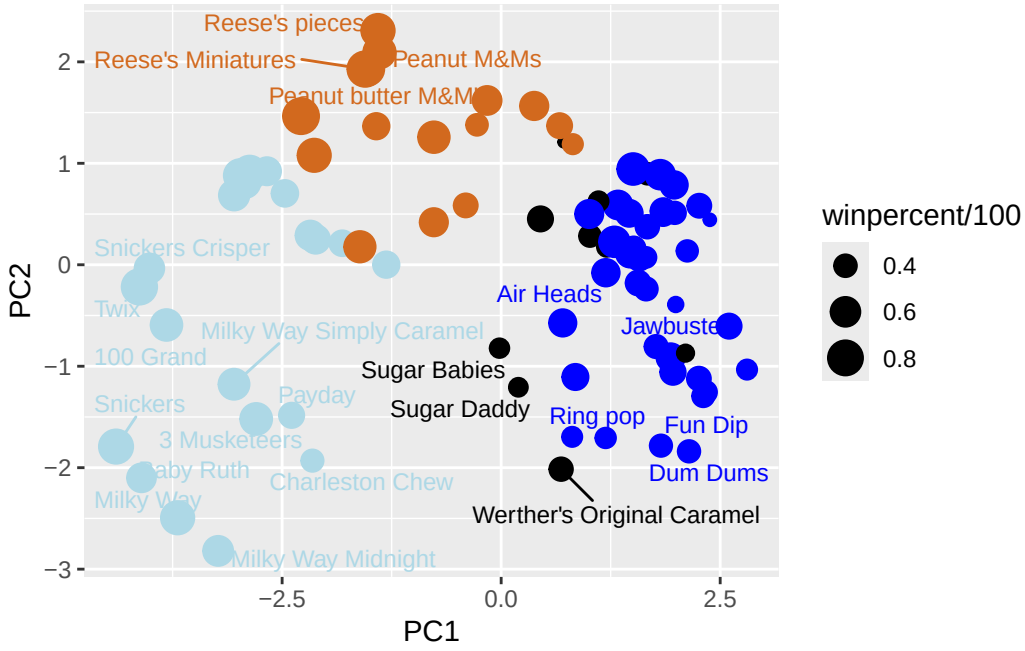
```
# Make a new datframe
my_data <- cbind(candy, pca$x[,1:3])
```

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

p
```




```
library(ggrepel)

p + geom_text_repel(size=3.3, col=my_cols, max.overlaps=7)
```

```
Warning: ggrepel: 62 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



We can pass the ggplot object `p` to **plotly** to generate an interactive plot that you can moouse over to see labels.

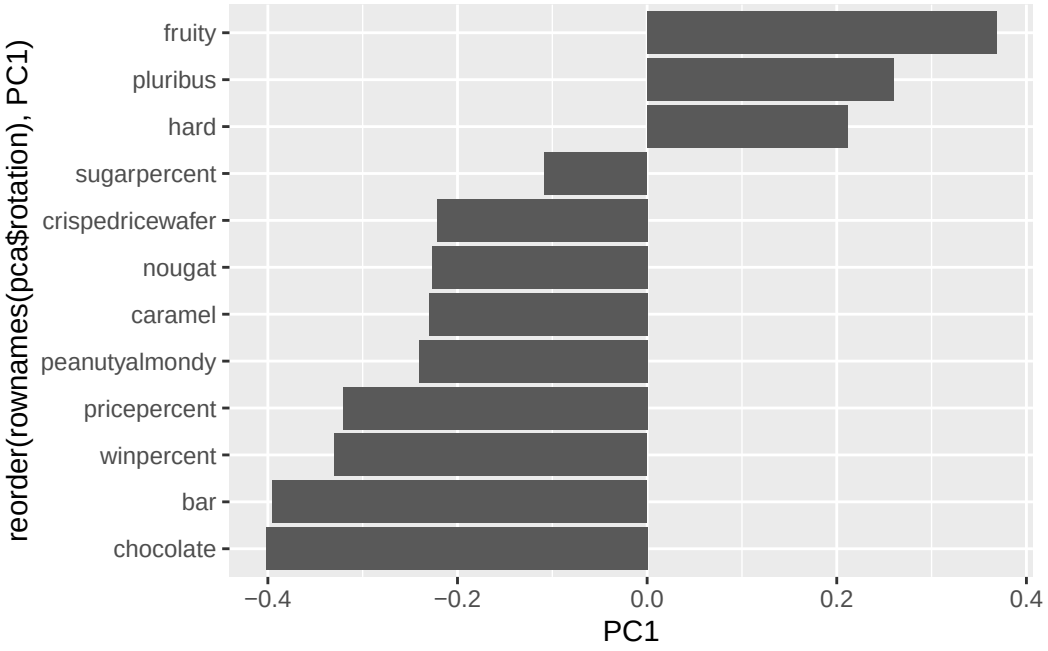
use `install.packages()` to install plotly.

Note: plot was made in R Studio, but since it cannot be rendered into PDF format, plotly steps are left out of report.

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

```
PC1 <- pca$rotation[,1]
```

```
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?

fruity, pluribus(comes in a bag), and hard are picked up strongly by PC1. Yes, this does make sense because we can see in the PCA plot that the blue dots(fruity candies) are grouped in the positive direction on the x-axis(PC1).