Class10

Brianna Smith

Importing candy data

First things first, let's get the data from the FiveThirtyEight GitHub repo. You can either read from the URL directely or download this candy-data.csv file and place it in your project directory. Either way we need to load it up with `read.csv()` and inspect the data to see exactly what we're dealing with.

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

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?

```
ncol(candy)*5
```

```
[1] 60
```

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

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

```
[1] 38
```

What is your favorate candy?

One of the most interesting variables in the dataset is winpercent. For a given candy this value is the percentage of people who prefer this candy over another randomly chosen candy from the dataset (what 538 term a matchup). Higher values indicate a more popular candy.

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

```
[1] 81.64291
```

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

```
candy["Sour Patch Kids", ]$winpercent
```

```
[1] 59.864
```

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
```

library("skimr") skim(candy)

```
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?

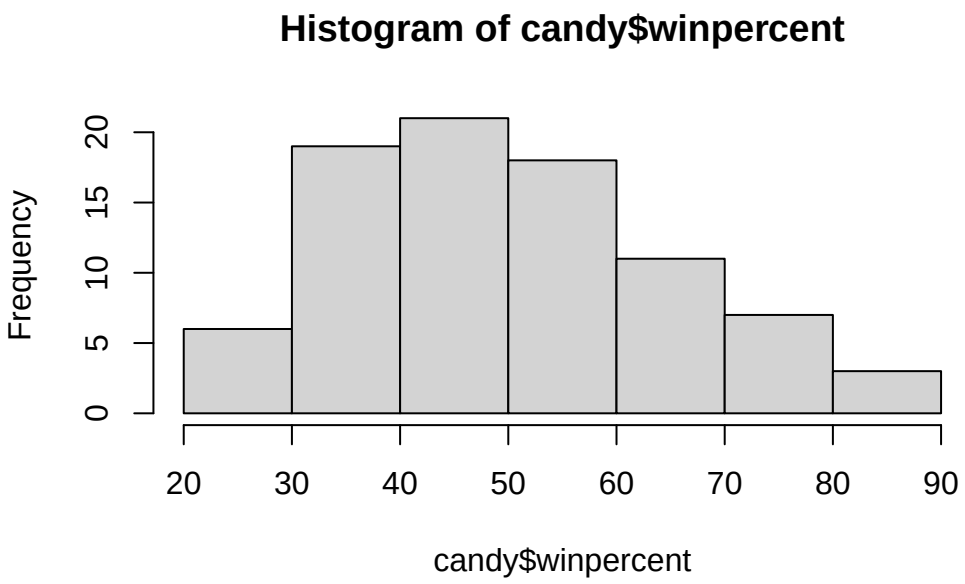
Winpercent is very different because it has whole numbers in the mean, sd, p0,p25, p50, p75, and p100.

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

A good place to start any exploratory analysis is with a histogram. You can do this most easily with the base R function `hist()`. Alternatively, you can use `ggplot()` with `geom_hist()`. Either works well in this case and (as always) its your choice.

Q8. Plot a histogram of winpercent values

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



Q9. Is the distribution of winpercent values symmetrical?

The distribution is skewed to the right and it is not symentrical.

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

The center is below 50 at about 45%.

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

```
chocolate <- mean(candy$winpercent[as.logical(candy$chocolate)])
chocolate
```

```
[1] 60.92153
```

```
fruit <- mean(candy$winpercent[as.logical(candy$fruity)])
fruit
```

```
[1] 44.11974
```

The chocolate candies are prefered over the fruity candies.

Q12. Is this difference statistically significant?

```
t.test(candy$winpercent[as.logical(candy$chocolate)], candy$winpercent[as.logical(candy$fr
```

```
    Welch Two Sample t-test

data:  candy$winpercent[as.logical(candy$chocolate)] and candy$winpercent[as.logical(candy$fr
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 p value is very low, therefore this is a significant statistical difference.

Overall Candy Rankings

Let's use the base R order() function together with head() to sort the whole dataset by winpercent

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

```
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
```

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

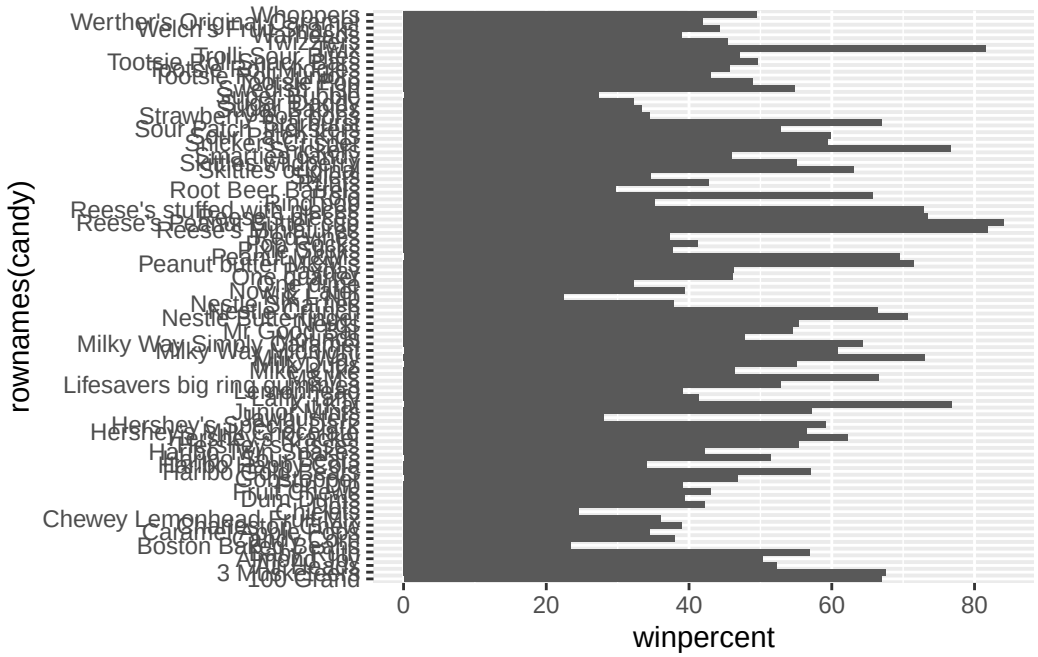
```
tail(candy[order(candy$winpercent),], n=5)
```

```
                        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
```

To examine more of the dataset in this vain we can make a barplot to visualize the overall rankings. We will use an iterative approach to building a useful visulization by getting a rough starting plot and then refining and adding useful details in a stepwise process.

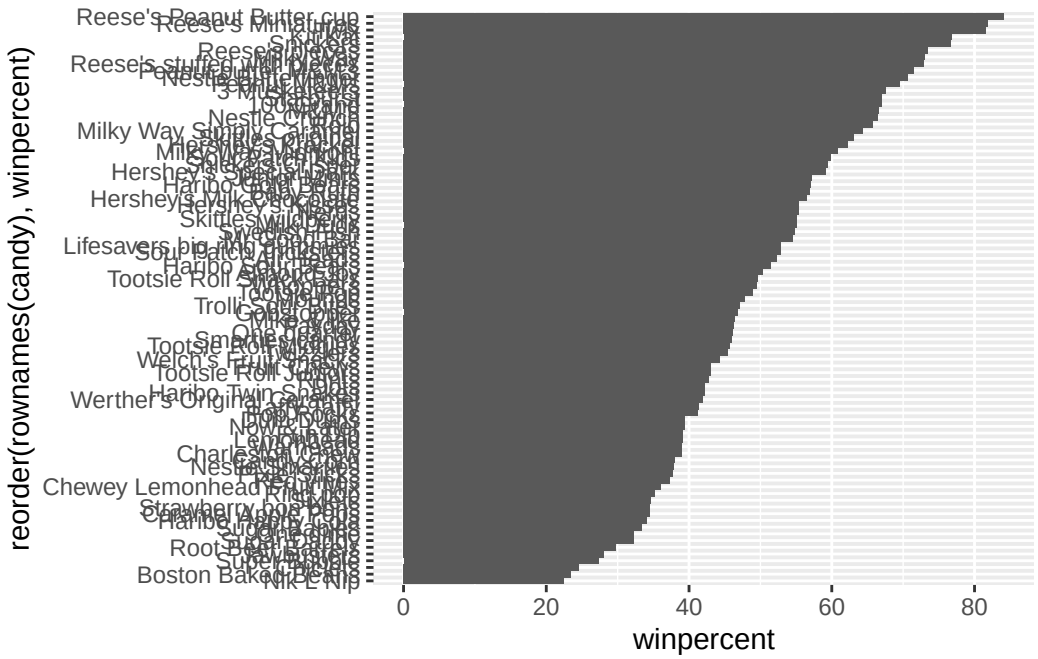
Q15. Make a first barplot of candy ranking based on winpercent values.

```
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?

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

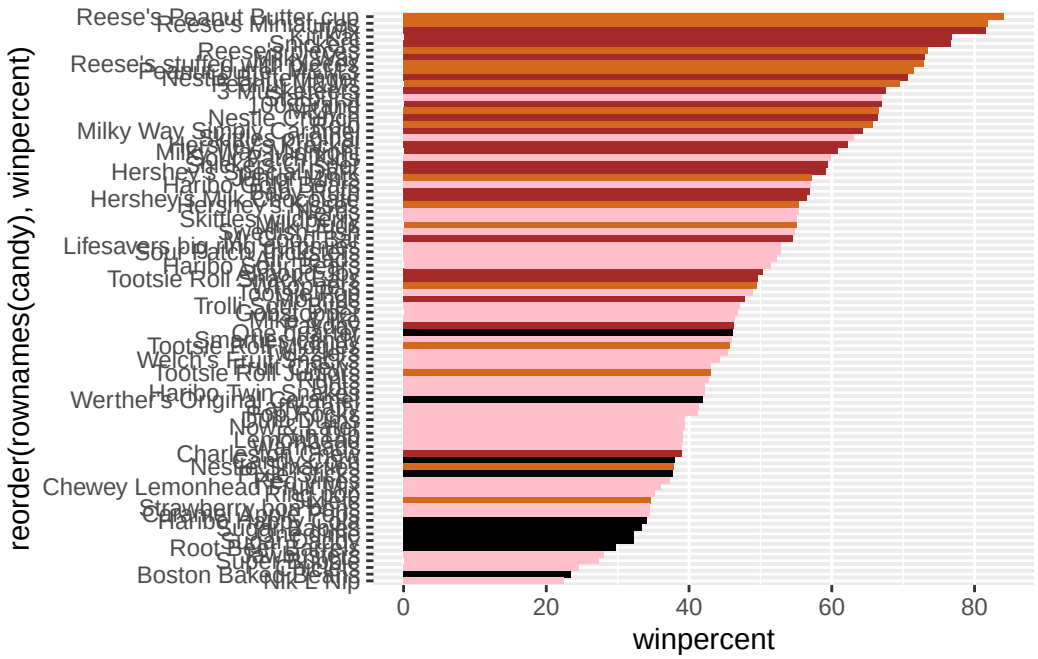


Time to add some useful color

Let's setup a color vector (that signifies candy type) that we can then use for some future plots. We start by making a vector of all black values (one for each candy). Then we overwrite chocolate (for chocolate candy), brown (for candy bars) and red (for fruity candy) values.

```
my_cols=rep("black", nrow(candy))
my_cols[as.logical(candy$chocolate)] = "chocolate"
my_cols[as.logical(candy$bar)] = "brown"
my_cols[as.logical(candy$fruity)] = "pink"
```

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



Q17. What is the worst ranked chocolate candy?

Sixlets

Q18. What is the best ranked fruity candy?

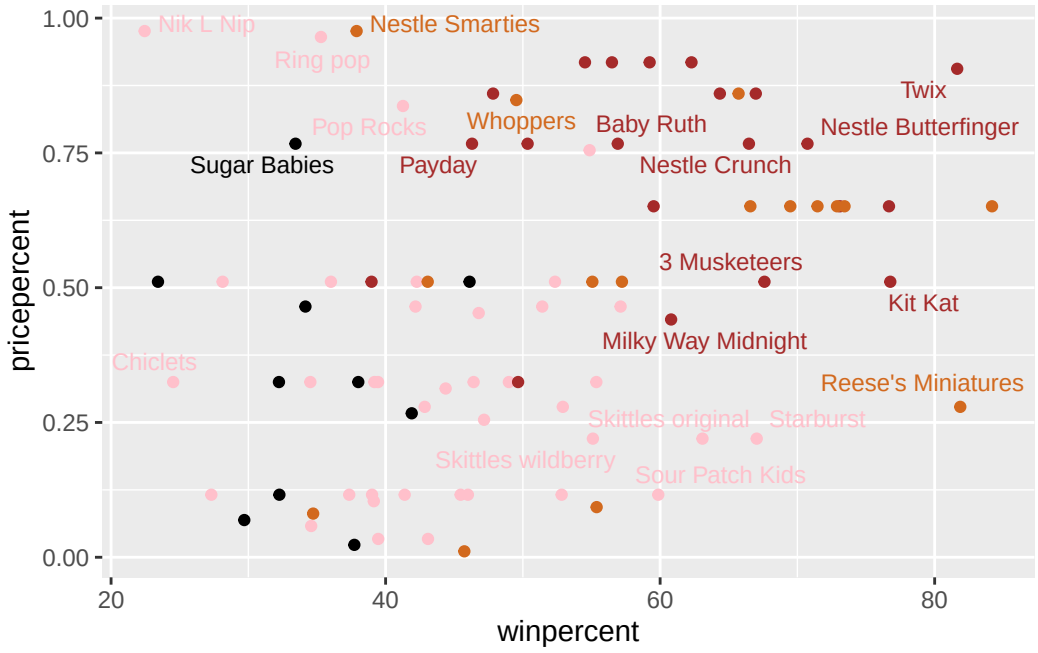
Starburst

Taking a look at pricepercent

The pricepercent variable records the percentile rank of the candy's price against all the other candies in the dataset. Lower vales are less expensive and high values more expensive. To this plot we will add text labels so we can more easily identify a given candy. There is a regular geom_label() that comes with ggplot2. To help with overlapping, we can use the geom_text_repel() function from the ggrepel package.

```
library(ggrepel)
# How about a plot of price vs win
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(col=my_cols, size=3.3, max.overlaps = 5)
```

```
Warning: ggrepel: 65 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?

```
ord <- order(candy$pricepercent, decreasing = TRUE)
tail( candy[ord,c(11,12)], n=5 )
```

```
                    pricepercent winpercent
Strawberry bon bons        0.058   34.57899
Dum Dums                   0.034   39.46056
Fruit Chews                0.034   43.08892
Pixie Sticks               0.023   37.72234
Tootsie Roll Midgies       0.011   45.73675
```

Tootsie roll midgeies are the least expensive with the highest winpercent of the five.

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 = 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
```

Nik L Nip is the least popular of the most expensive candies.

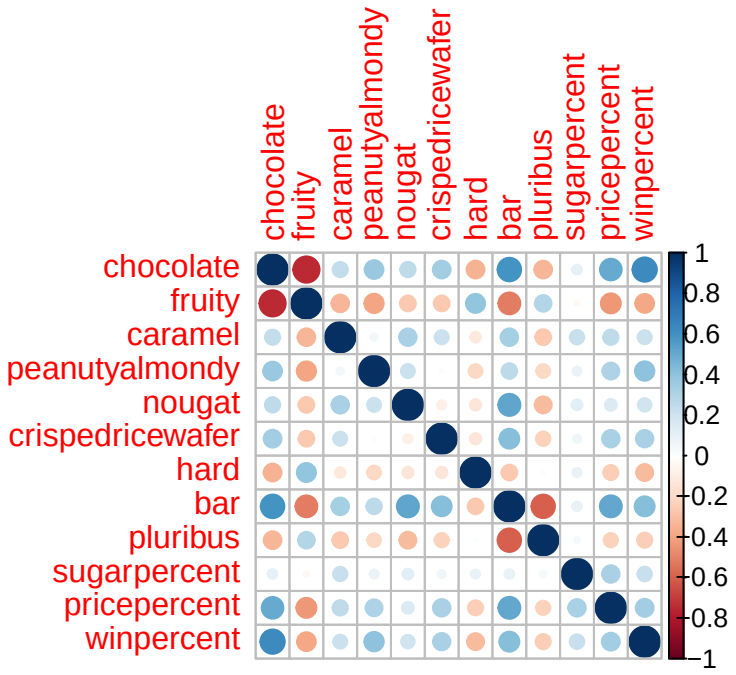
Exploring the correlation structure

Now that we've explored the dataset a little, we'll see how the variables interact with one another. We'll use correlation and view the results with the corrplot package to plot a correlation matrix

```
library(corrplot)
```

```
corrplot 0.92 loaded
```

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



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

Chocolate and fruity are the most anti-correlated

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

chocolate and winpercent are most positively correlated.

Principal Component Analysis

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

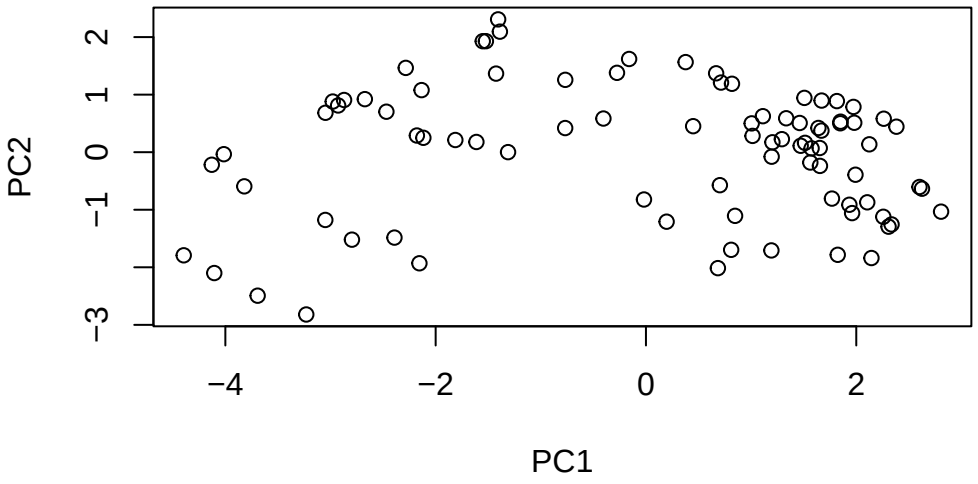
```
pca <- prcomp(candy, scale=TRUE)
summary(pca)
```

```
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
```

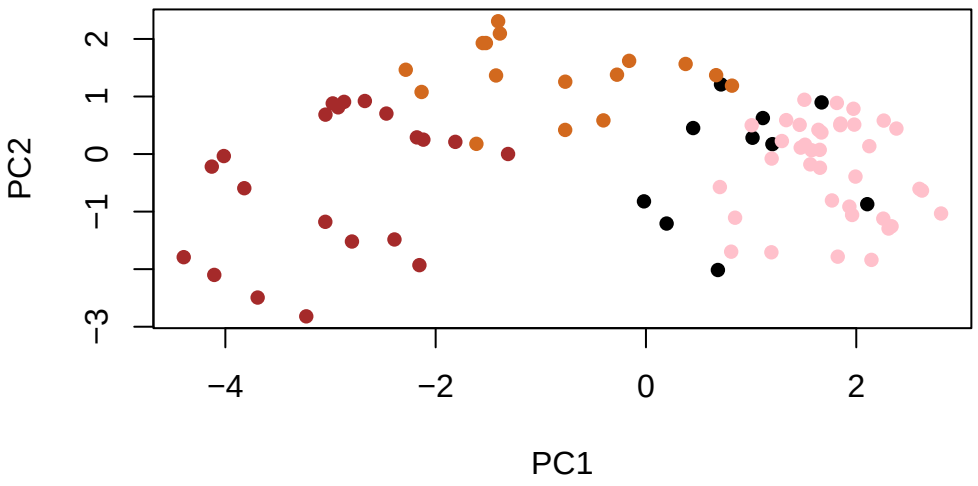
Now we can plot our main PCA score plot of PC1 vs PC2.

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



We can change the plotting character and add some color:

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

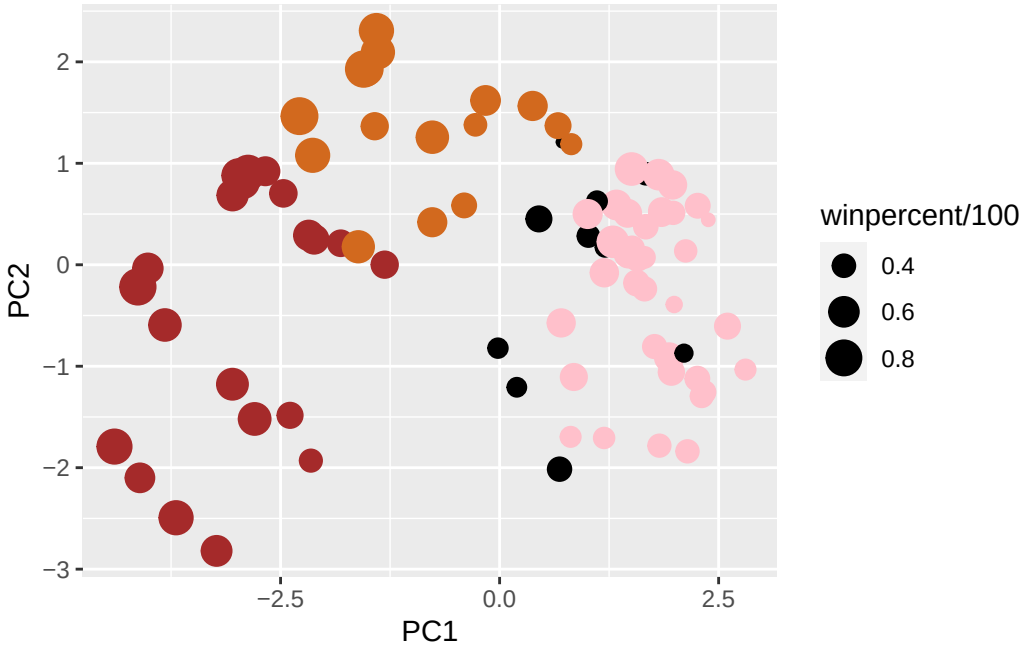


We can make a much nicer plot with the ggplot2 package but it is important to note that ggplot works best when you supply an input data.frame that includes a separate column for each of the aesthetics you would like displayed in your final plot.

```
# Make a new data-frame with our PCA results and candy data
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
```

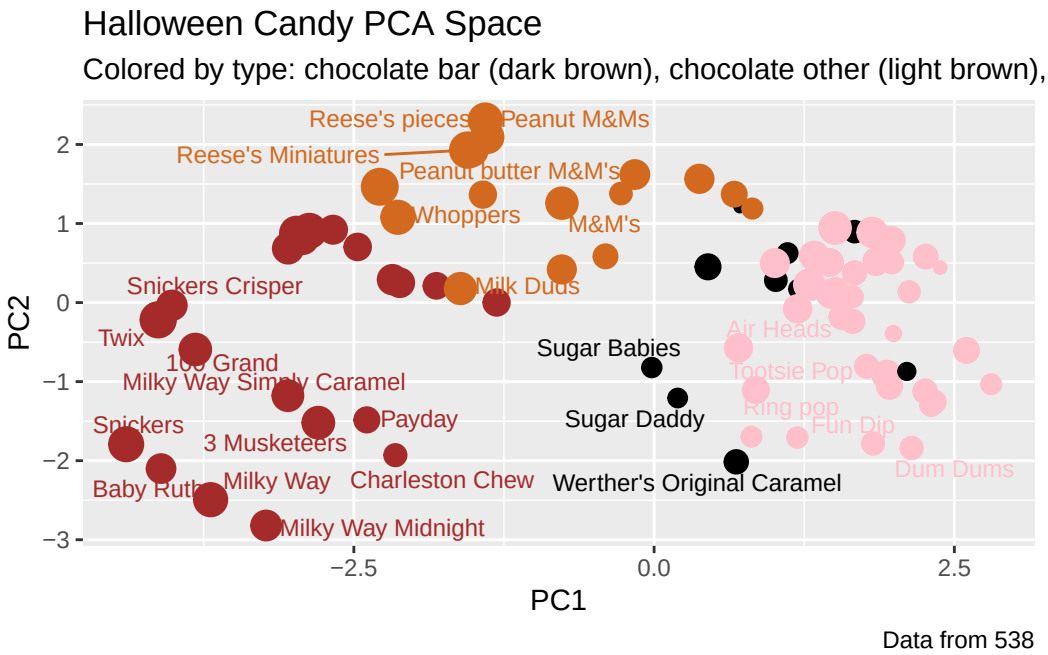


Again we can use the ggrepel package and the function ggrepel::geom_text_repel() to label up the plot with non overlapping candy names like. We will also add a title and subtitle like so:

```
library(ggrepel)

p + geom_text_repel(size=3.3, col=my_cols, 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
```



If you want to see more candy labels you can change the max.overlaps value to allow more overlapping labels or pass the ggplot object p to plotly like so to generate an interactive plot that you can mouse over to see labels:

```
library(plotly)
```

```
Attaching package: 'plotly'
```

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

    last_plot
```

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

    filter
```

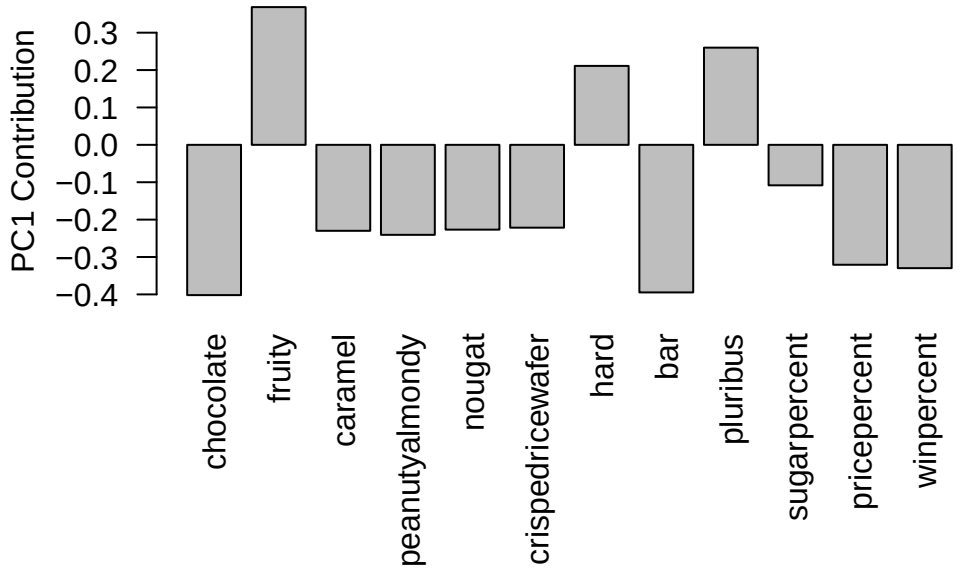
```
The following object is masked from 'package:graphics':

    layout
```

```
#ggplotly(p)
```

Let's finish by taking a quick look at PCA our loadings. Do these make sense to you?

```
par(mar=c(8,4,2,2))
barplot(pca$rotation[,1], las=2, ylab="PC1 Contribution")
```



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

Fruity, pluribus and hard are picked up in the positive direction. This would make sense to me because the fruity candy is correlated with being hard and coming in packs of multiples.