

Class 9: Halloween Mini-Project

Nicolò (PID: A18109144)

Exploratory Analysis of Halloween Candy

1. Importing candy data

```
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?

```
nrow(candy)
```

```
[1] 85
```

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

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

```
[1] 38
```

2. What is your favorite candy?

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

```
[1] 81.64291
```

Q3. What is your favorite candy in the dataset and what is its winpercent value?

My favorite candy is “Haribo Sour Bears”.

```
candy["Haribo Sour Bears", ]$winpercent
```

```
[1] 51.41243
```

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
```

Let’s use the skim() function in the skimr package that can help give us a quick overview of the candy dataset.

```
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?

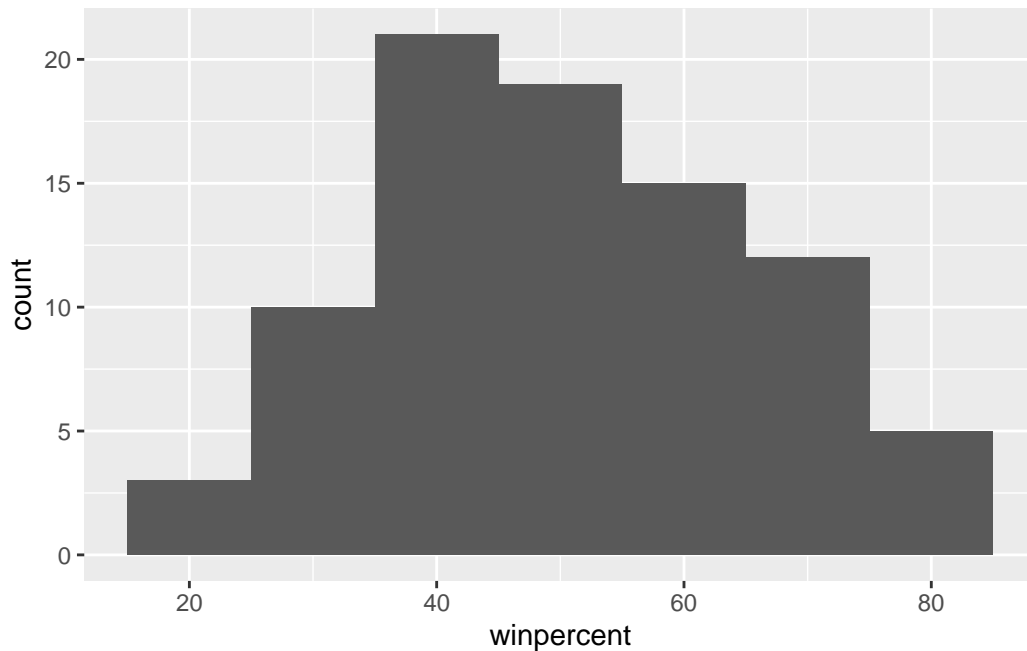
winpercent

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

I think it indicates the presence or not of chocolate in that candy.

Q8. Plot a histogram of winpercent values

```
library("ggplot2")
ggplot(candy, aes(winpercent)) +
  geom_histogram(binwidth=10)
```



Q9. Is the distribution of winpercent values symmetrical?

It is not.

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

It's below 50%.

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

```
choc.inds <- as.logical(candy$chocolate)
fruit.inds <- as.logical(candy$fruity)
choc.win <- candy[choc.inds, "winpercent"]
fruit.win <- candy[fruit.inds, "winpercent"]

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

[1] TRUE

On average, chocolate candy is higher ranked than fruity candy.

Q12. Is this difference statistically significant?

```
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
```

It is, since the p-value is very small

3. Overall Candy Rankings

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

```
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

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

	chocolate	fruity	caramel	peanut	almond	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

	crisped	rice	wafer	hard bar	pluribus	sugar	percent	price	percent
Nik L Nip				0	0	1	0.197	0.976	
Boston Baked Beans				0	0	1	0.313	0.511	
Chiclets				0	0	1	0.046	0.325	
Super Bubble				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?

```
candy %>%
  arrange(winpercent) %>%
  tail(5)
```

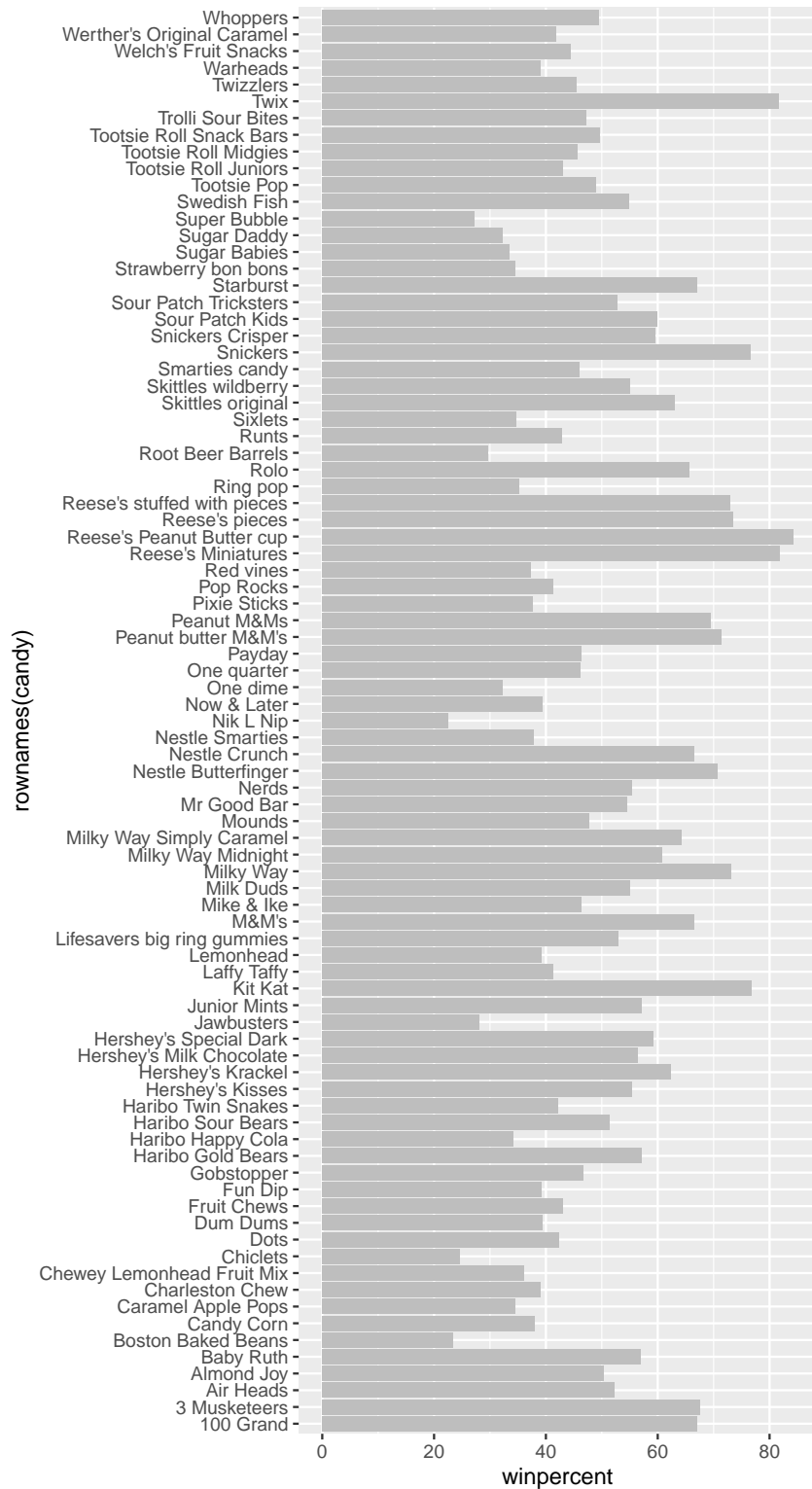
	chocolate	fruity	caramel	peanut	almond	nougat
Snickers	1	0	1		1	1
Kit Kat	1	0	0		0	0
Twix	1	0	1		0	0
Reese's Miniatures	1	0	0		1	0
Reese's Peanut Butter cup	1	0	0		1	0

	crisped	rice	wafer	hard bar	pluribus	sugar	percent
Snickers				0	0	1	0.546
Kit Kat				1	0	1	0.313
Twix				1	0	1	0.546
Reese's Miniatures				0	0	0	0.034
Reese's Peanut Butter cup				0	0	0	0.720

	price	percent	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	

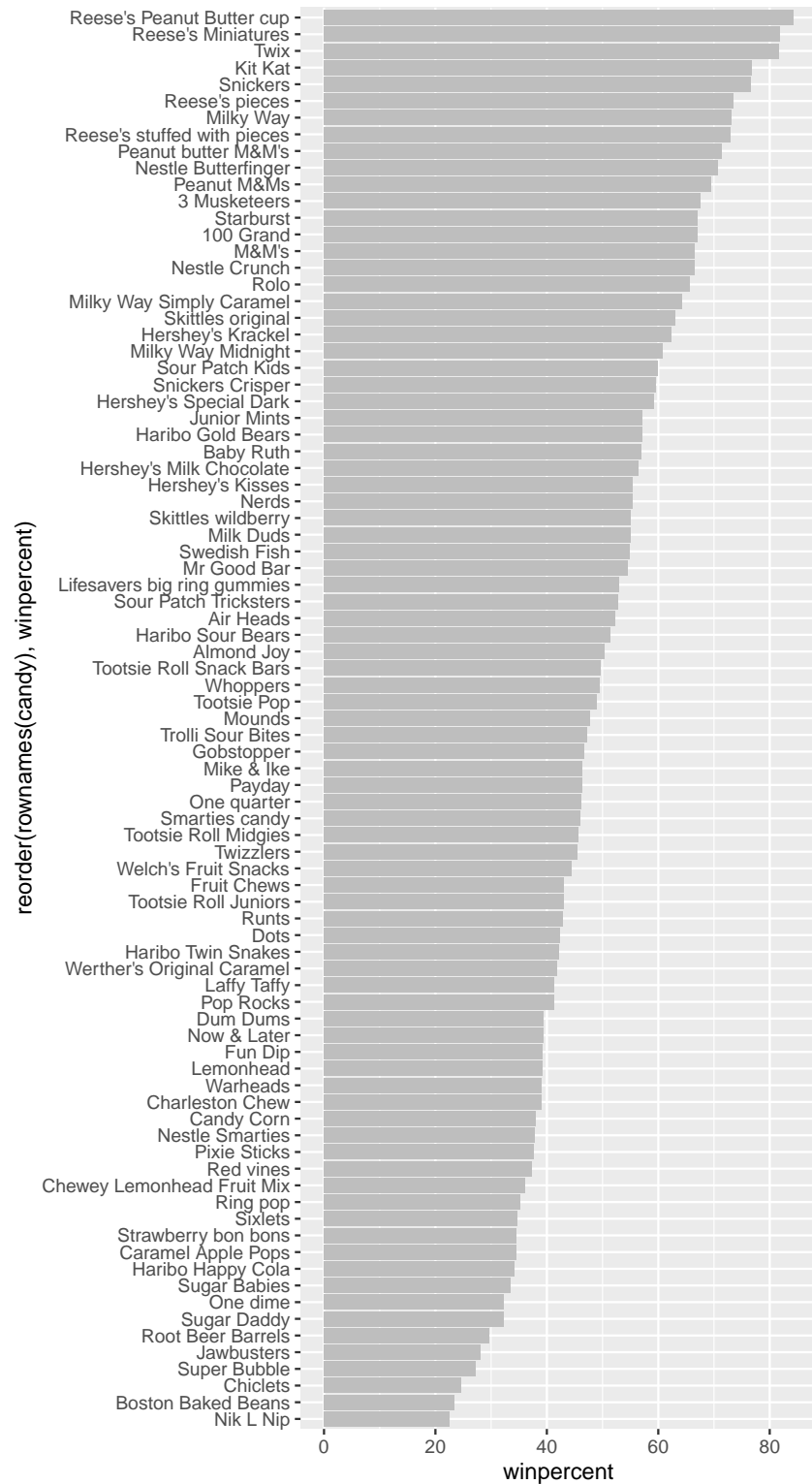
Q15. Make a first barplot of candy ranking based on winpercent values.

```
ggplot(candy) +  
  aes(winpercent, rownames(candy)) +  
  geom_col(fill = "gray")
```



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(fill = "gray")
```



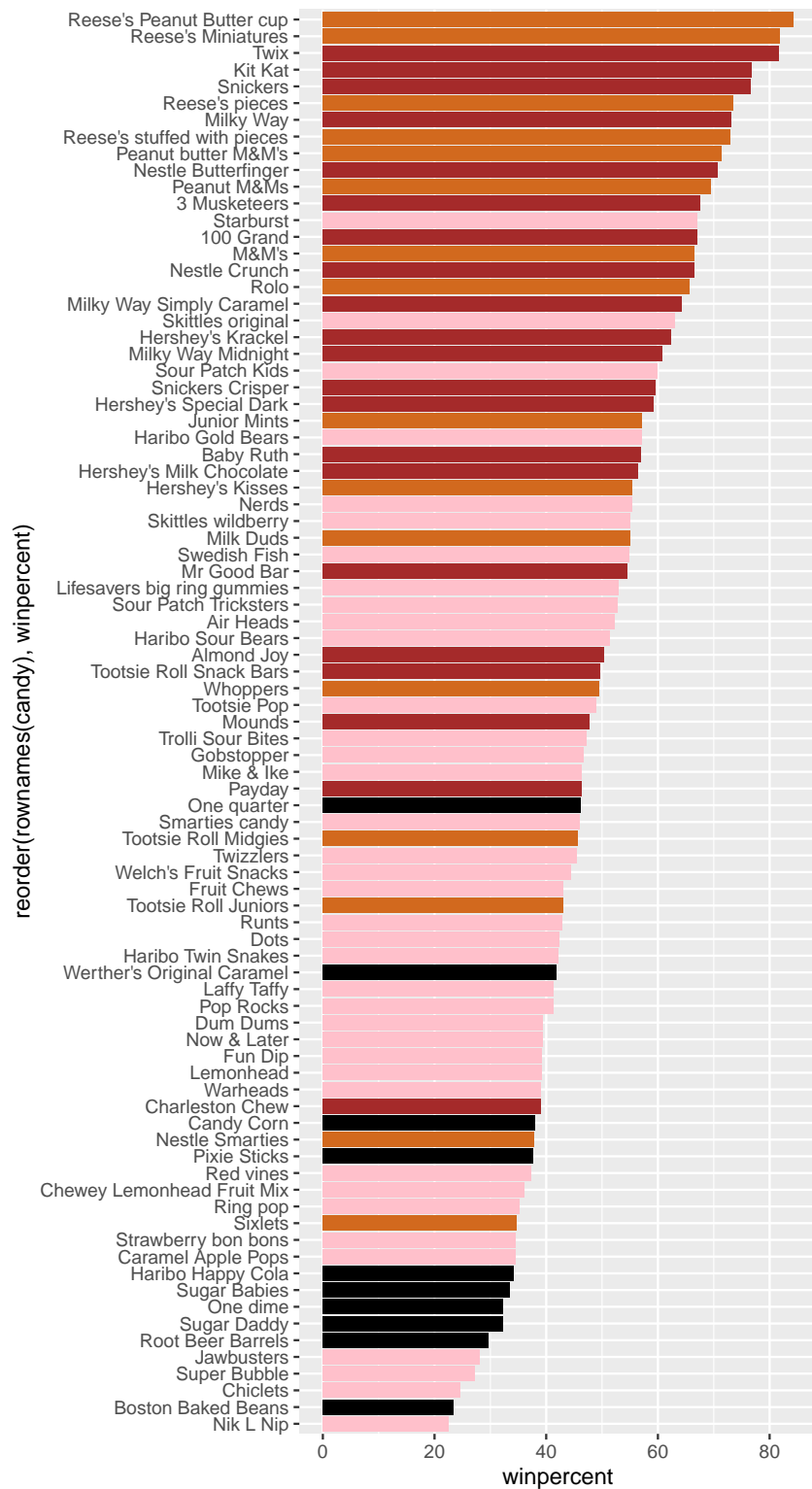
Time to add some useful color

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"
```

And make a colored bar plot

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



Now, for the first time, using this plot we can answer questions like:

Q17. What is the worst ranked chocolate candy?

Sixlets

Q18. What is the best ranked fruity candy?

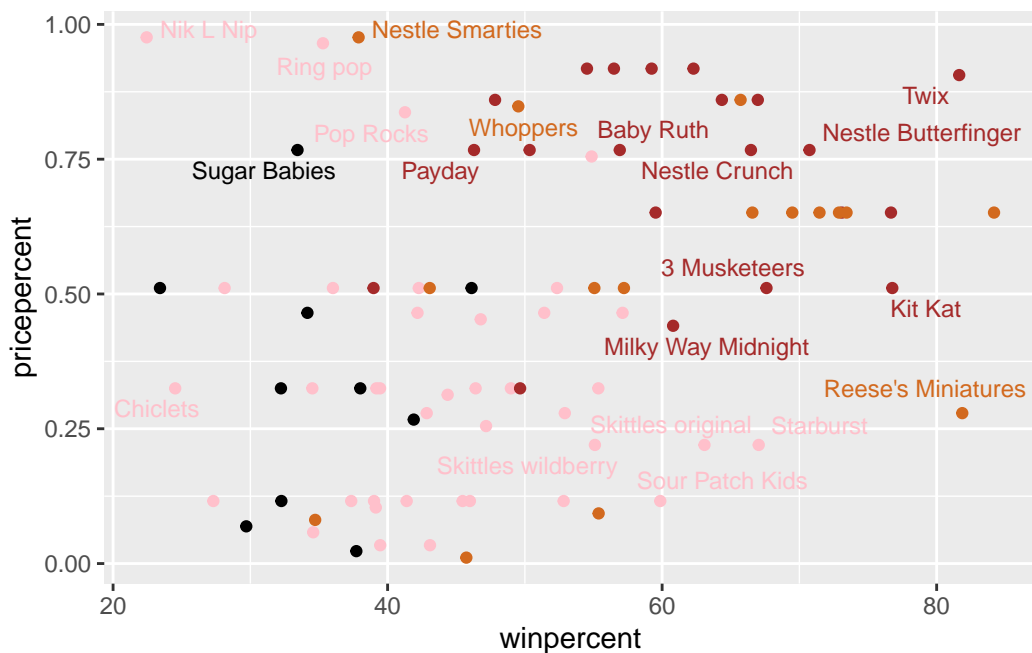
Starburst

4. Taking a look at pricepercent

```
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?

Reese's Miniatures

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

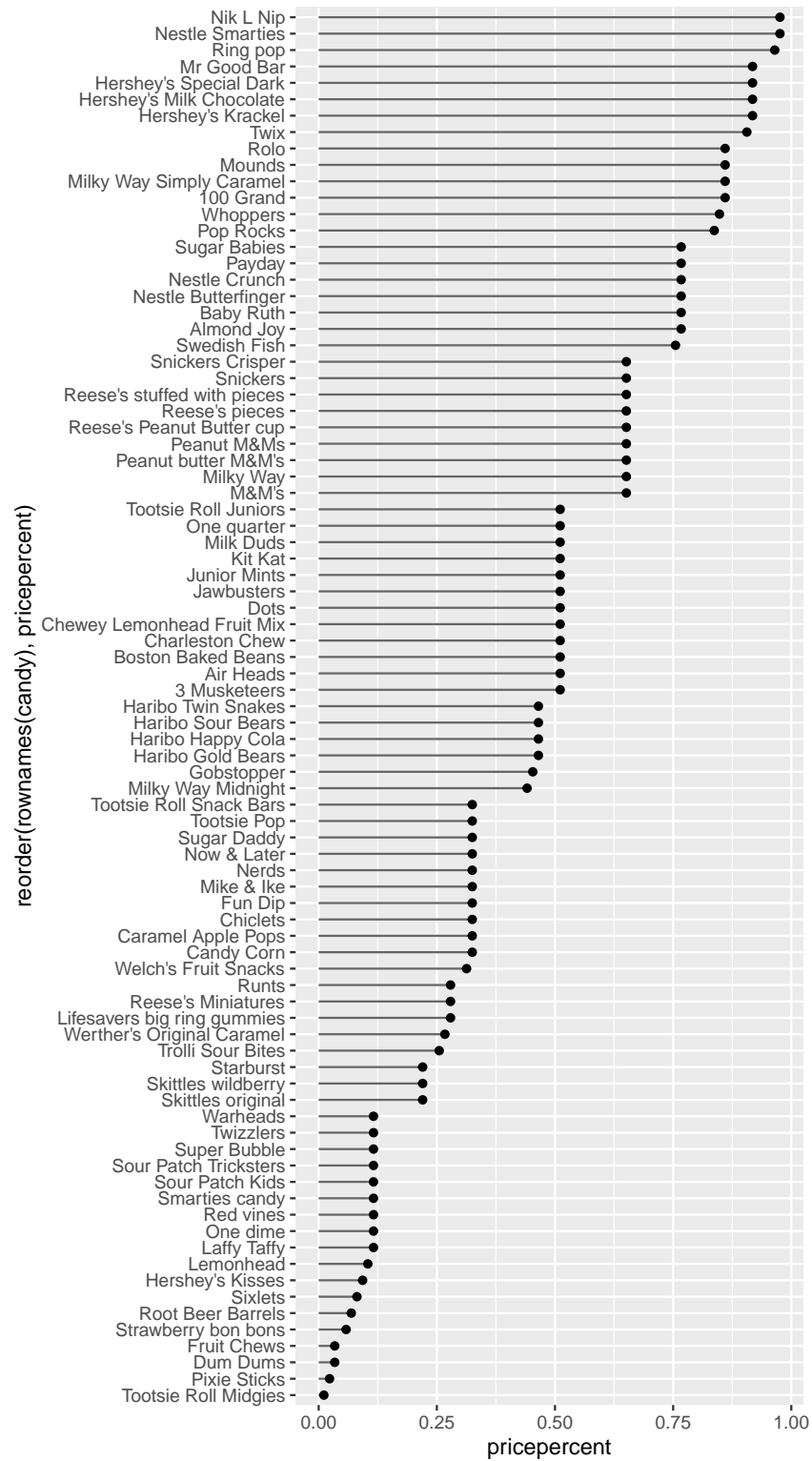
```
ord <- order(candy$pricepercent, decreasing = TRUE)
head( candy[ord,c(11,12)], n=5 )
```

	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 within the 5 most expensive candies.

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()`

```
# Make a lollipop chart of pricepercent
ggplot(candy) +
  aes(pricepercent, reorder(rownames(candy), pricepercent)) +
  geom_segment(aes(yend = reorder(rownames(candy), pricepercent),
                  xend = 0), col="gray40") +
  geom_point()
```

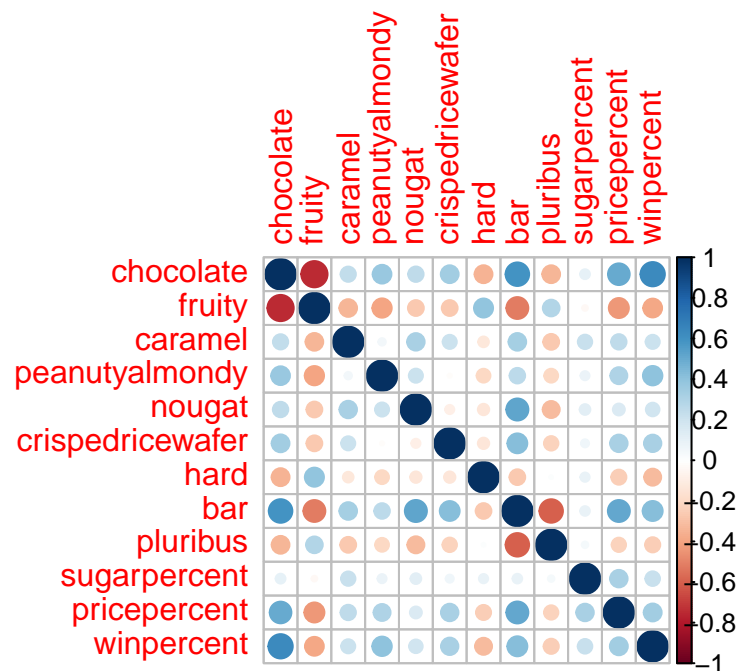


5. Exploring the correlation structure

```
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 vs fruity

bar vs pluribus

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

chocolate with bar, pricepercent and winpercent

6. Principal Component Analysis

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

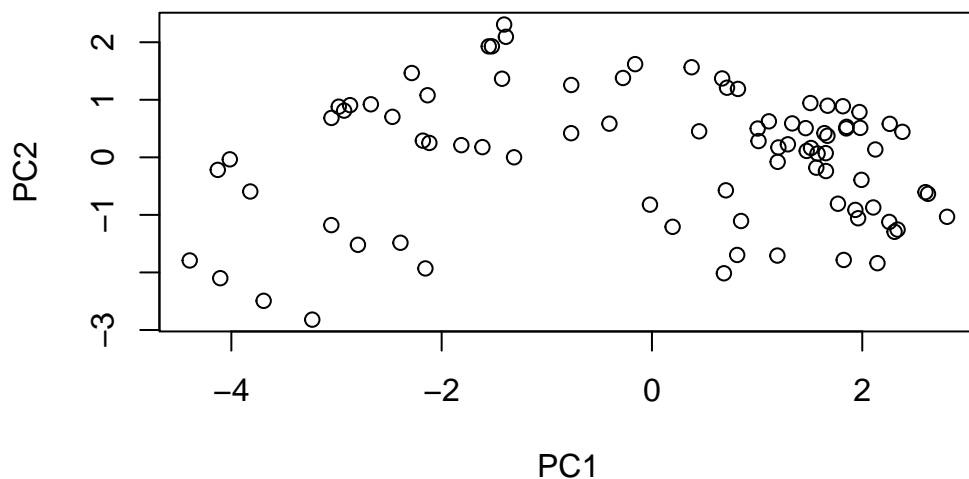
Importance of components:

	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

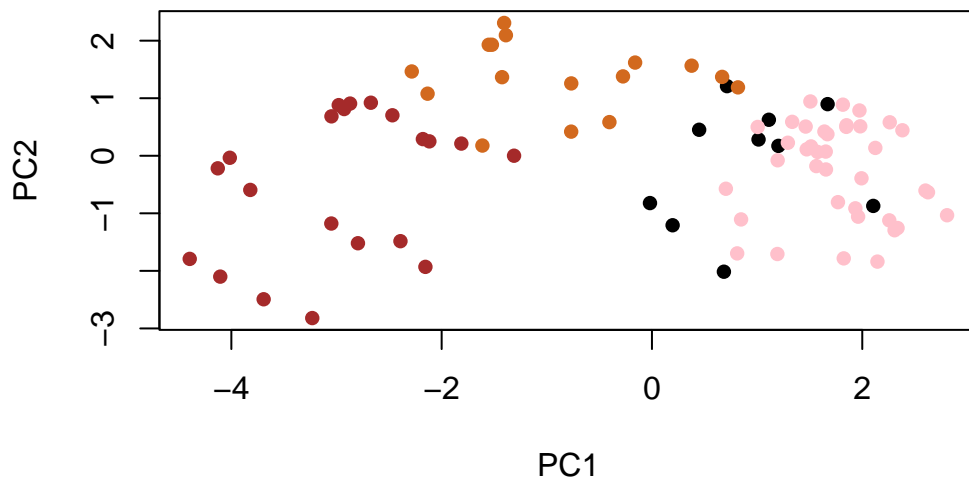
Now we can plot our main PCA score plot of PC1 vs PC2

```
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)
```

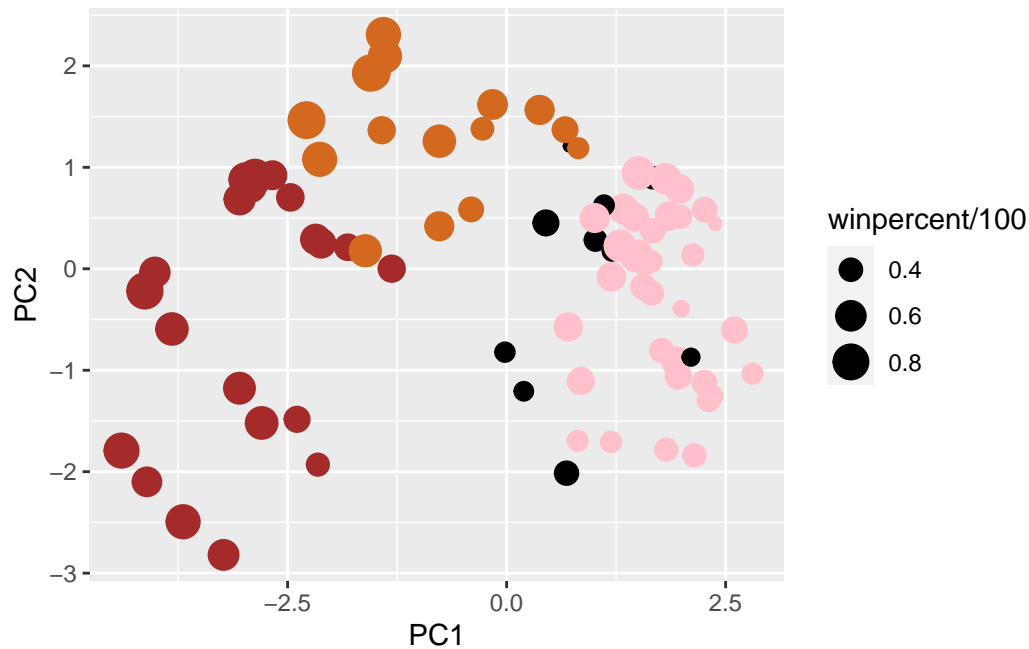


We can make a much nicer plot with the ggplot2 package but it is important to note that ggplot works best when you supply an input data.frame that includes a separate column for each of the aesthetics you would like displayed in your final plot. To accomplish this we make a new data.frame here that contains our PCA results with all the rest of our candy data. We will then use this for making plots below

```
# 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



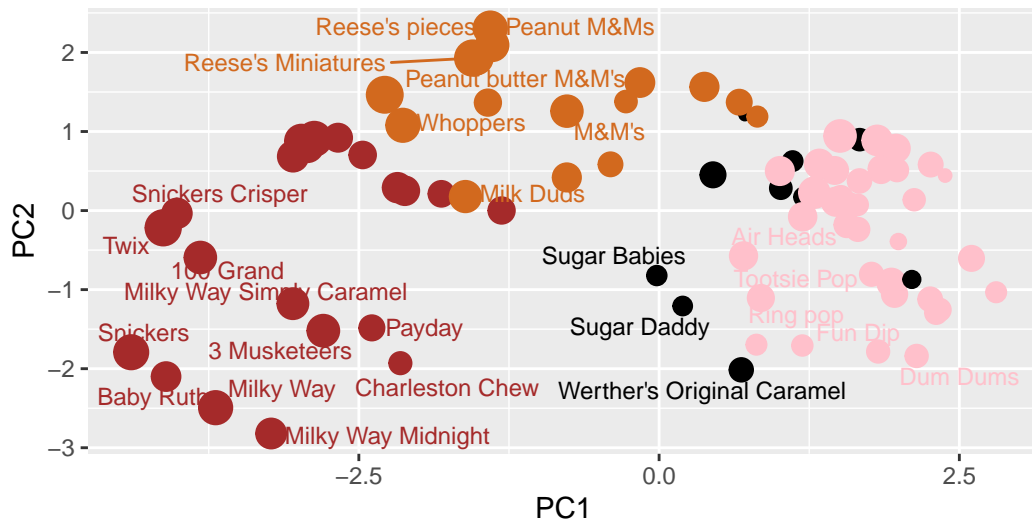
Again we can use the `ggrepel` package and the function `ggrepel::geom_text_repel()` to label up the plot with non overlapping candy names like. We will also add a title and subtitle like so:

```
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`

Halloween Candy PCA Space

Colored by type: chocolate bar (dark brown), chocolate other (light brown),



Data from 538

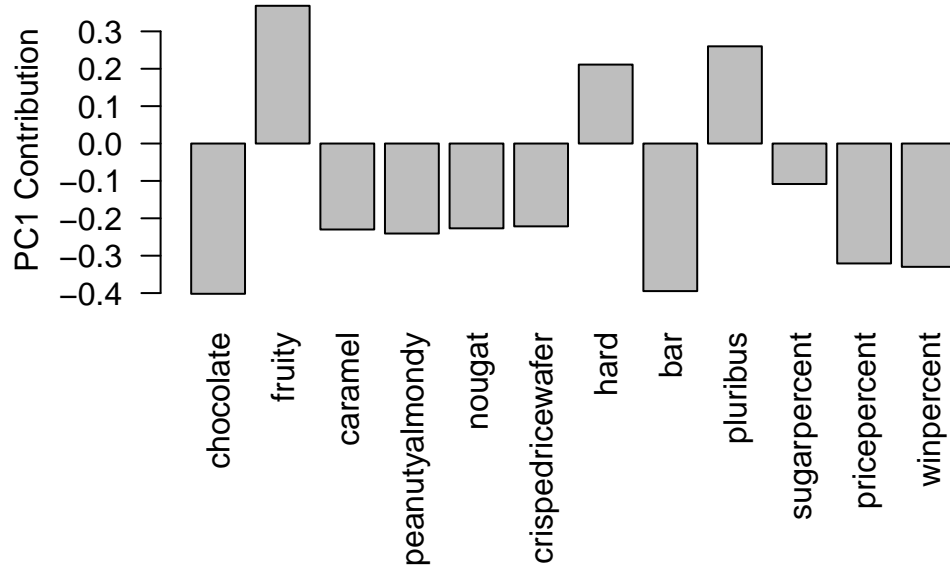
To have more candy labels we can change the max.overlaps value to allow more overlapping labels or pass the ggplot object p to plotly like so to generate an interactive plot that you can mouse over to see labels:

```
# library(plotly)

# ggplotly(p)
```

Let's finish by taking a quick look at PCA our loadings. Do these make sense to you? Notice the opposite effects of chocolate and fruity and the similar effects of chocolate and bar (i.e. we already know they are correlated).

```
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, hard and pluribus. It makes sense since they seem to be positively correlated with each other.