Class 09

Farnam Tavakoli (PID: A17628539)

Here we analyze

```
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
```

Exploration

Q3. What is your favorite candy in the dataset an what is its winerpercent value?

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

```
[1] 81.64291
```

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
```

Q. What is the least liked candy in the dataset - lowest winpercent

```
x <- c(5,3,4,1)
sort(x)
```

```
[1] 1 3 4 5
```

```
order(x)
```

```
[1] 4 2 3 1
```

```
inds <- order(candy$winpercent)
head(candy[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
```

```
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	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
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	

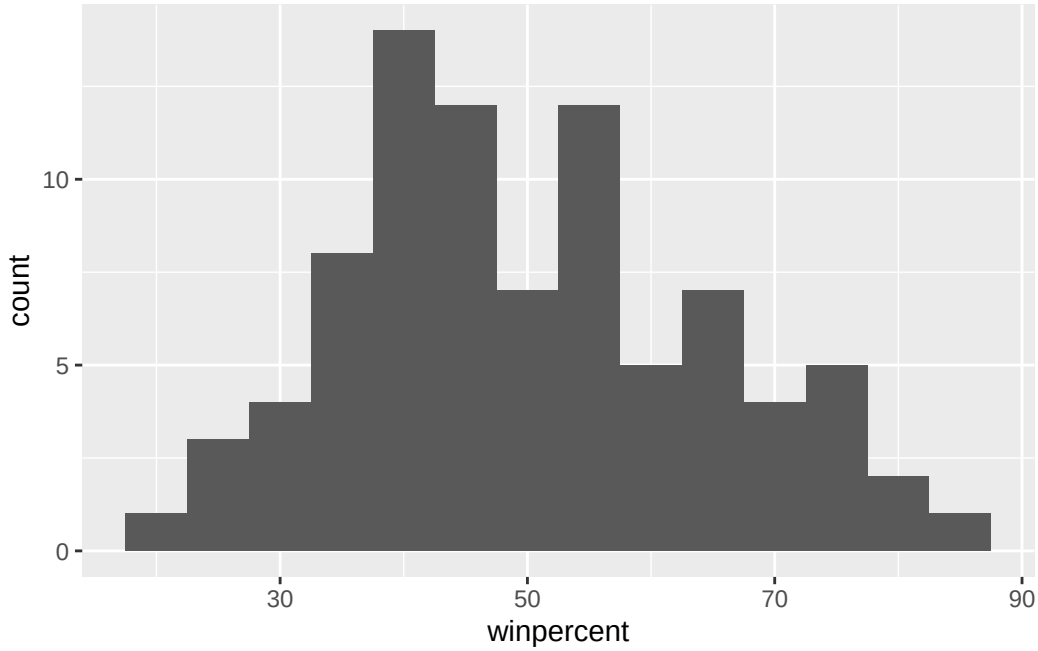
Q6. Is there any variable/column that looks to be on a different scale to the majority of the other columns in the dataset?

winpercent

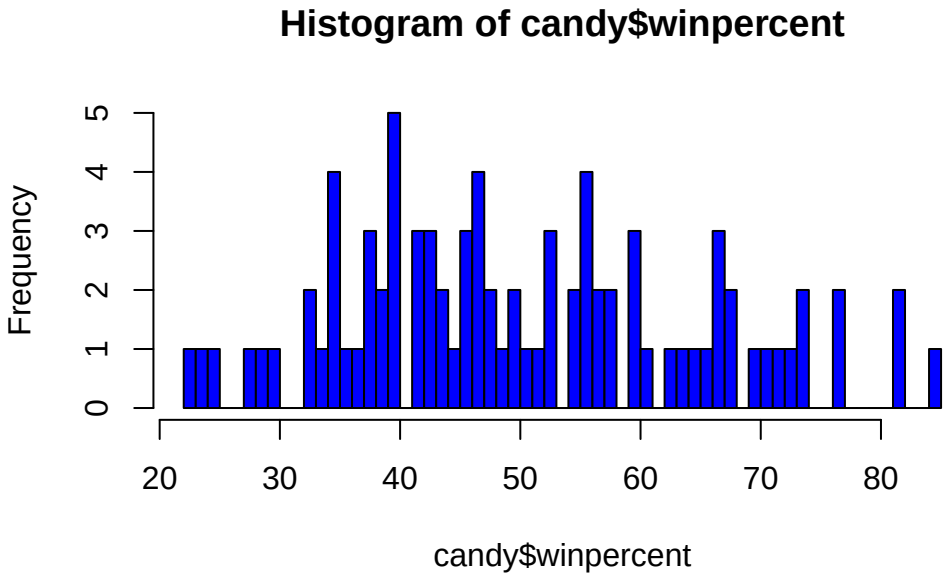
Q7. What do you think a zero and one represent for the candy$chocolate column?

Q8. Plot a histogram of winpercent values

```
library(ggplot2)
ggplot(candy, aes(winpercent))+
  geom_histogram(binwidth = 5)
```




```
hist(candy$winpercent, col= "blue", breaks =80)
```



Q9. Is the distribution of winpercent values symmetrical?

No

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

Below

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

First find all chocolatecandy and thier $winpercent values

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

```
[1] 60.92153
```

```
#candy$fruity == 1
fruit.inds <- as.logical(candy$fruity)
fruit.win <- candy[fruit.inds,]$winpercent
mean(fruit.win)
```

```
[1] 44.11974
```

Q12. Is this difference statistically significant?

```
t.test(choc.win, fruit.win)
```

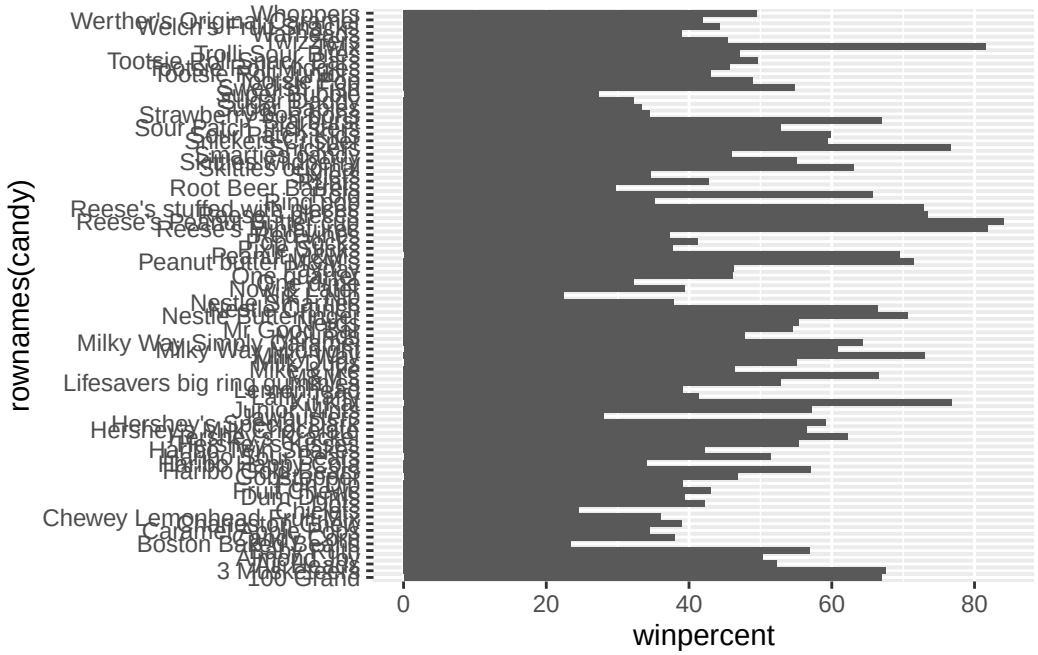
```
    Welch Two Sample t-test

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
```

Q13. What are the five least liked candy types in this set? Q14. What are the top 5 all time favorite candy types out of this set?

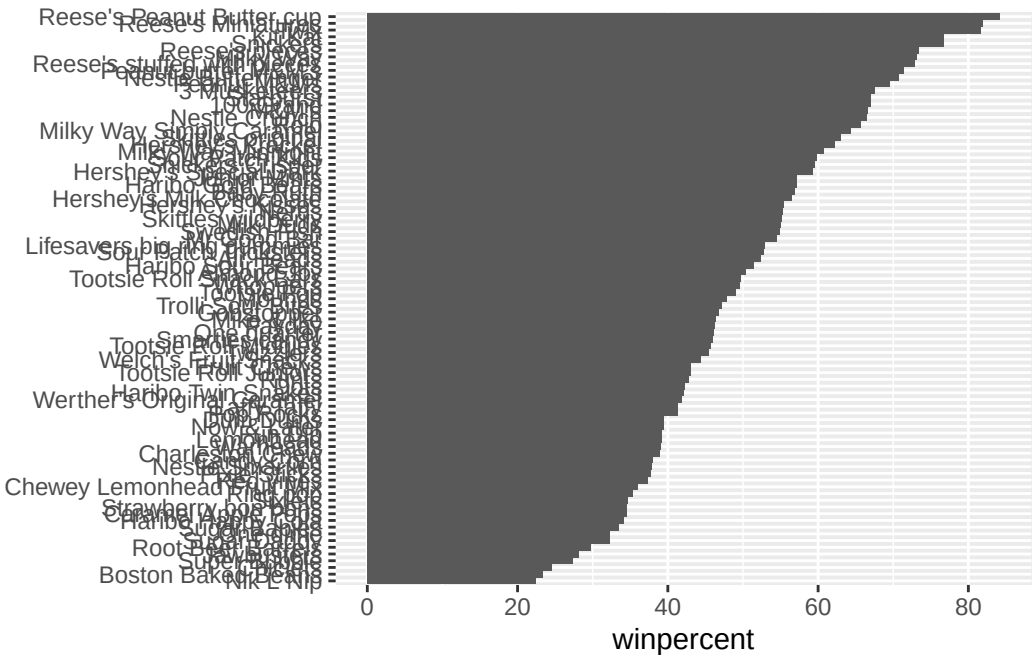
Q15. Make a first barplot of candy ranking based on winpercent values.

```
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() +
  labs(x="winpercent", y= NULL)
```




```
ggsave('barplot1.png', width = 7, height = 10)
```

You can insert any image with this markdown syntax

Add some color to our ggplot. We need to make a custom color vector.

```
# Start with all black vector of colors
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"
my_cols
```

```
 [1] "brown"     "brown"     "black"     "black"     "pink"      "brown"
 [7] "brown"     "black"     "black"     "pink"      "brown"     "pink"
[13] "pink"      "pink"      "pink"      "pink"      "pink"      "pink"
[19] "pink"      "black"     "pink"      "pink"      "chocolate" "brown"
[25] "brown"     "brown"     "pink"      "chocolate" "brown"     "pink"
[31] "pink"      "pink"      "chocolate" "chocolate" "pink"      "chocolate"
[37] "brown"     "brown"     "brown"     "brown"     "brown"     "pink"
```

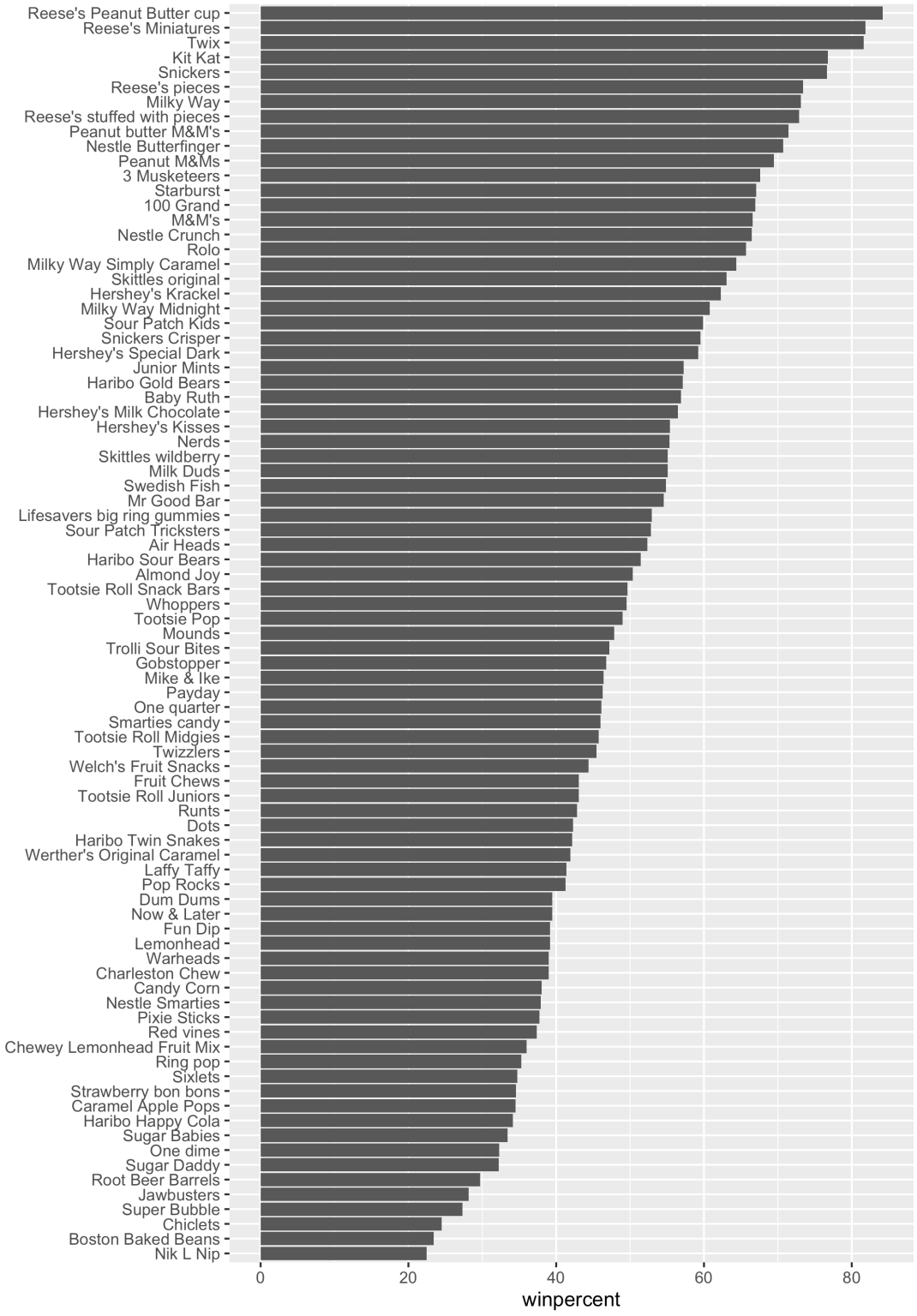
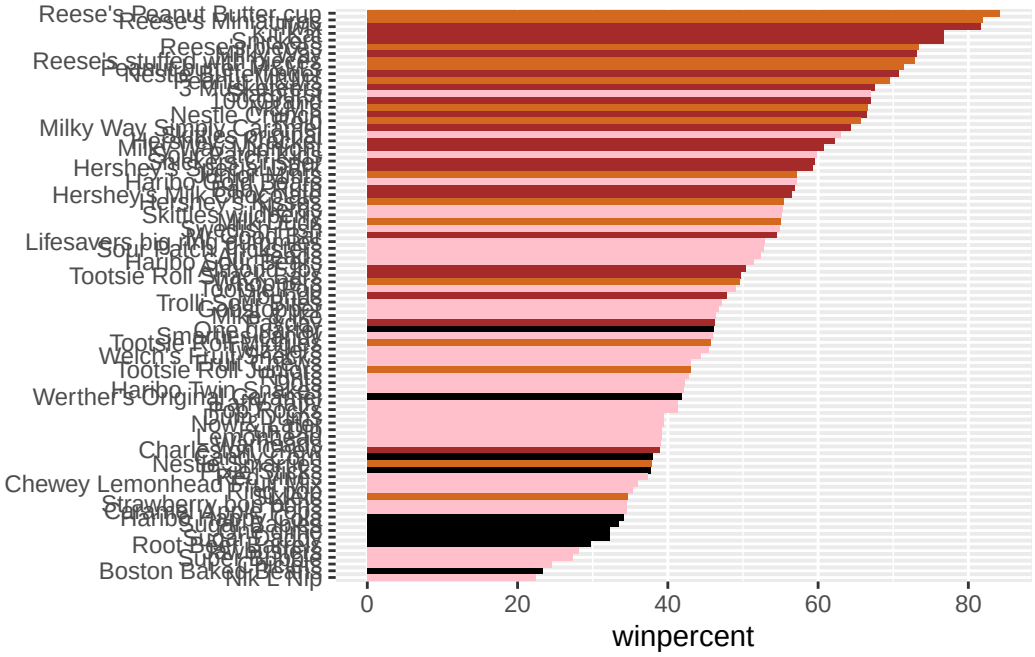


Figure 1: An example of photo insertion

```
[43] "brown"     "brown"     "pink"      "pink"      "brown"     "chocolate"
[49] "black"     "pink"      "pink"      "chocolate" "chocolate" "chocolate"
[55] "chocolate" "pink"      "chocolate" "black"     "pink"      "chocolate"
[61] "pink"      "pink"      "chocolate" "pink"      "brown"     "brown"
[67] "pink"      "pink"      "pink"      "pink"      "black"     "black"
[73] "pink"      "pink"      "pink"      "chocolate" "chocolate" "brown"
[79] "pink"      "brown"     "pink"      "pink"      "pink"      "black"
[85] "chocolate"
```

```
ggplot(candy)+
  aes(winpercent, reorder(rownames(candy), winpercent)) +
  geom_col(fill= my_cols) +
  labs(x="winpercent", y= NULL)
```



Q17. What is the worst ranked chocolate candy?

Q18. What is the best ranked fruity candy?

Taking alook at pricepercent

```
candy$pricepercent
```

```
 [1] 0.860 0.511 0.116 0.511 0.511 0.767 0.767 0.511 0.325 0.325 0.511 0.511
[13] 0.325 0.511 0.034 0.034 0.325 0.453 0.465 0.465 0.465 0.465 0.093 0.918
[25] 0.918 0.918 0.511 0.511 0.511 0.116 0.104 0.279 0.651 0.651 0.325 0.511
[37] 0.651 0.441 0.860 0.860 0.918 0.325 0.767 0.767 0.976 0.325 0.767 0.651
[49] 0.023 0.837 0.116 0.279 0.651 0.651 0.651 0.965 0.860 0.069 0.279 0.081
[61] 0.220 0.220 0.976 0.116 0.651 0.651 0.116 0.116 0.220 0.058 0.767 0.325
[73] 0.116 0.755 0.325 0.511 0.011 0.325 0.255 0.906 0.116 0.116 0.313 0.267
[85] 0.848
```

If we want to see what is a good candy to buy in terms of winpercent and pricepercent we can plot these two variables and then see the best candy for the least amount of money

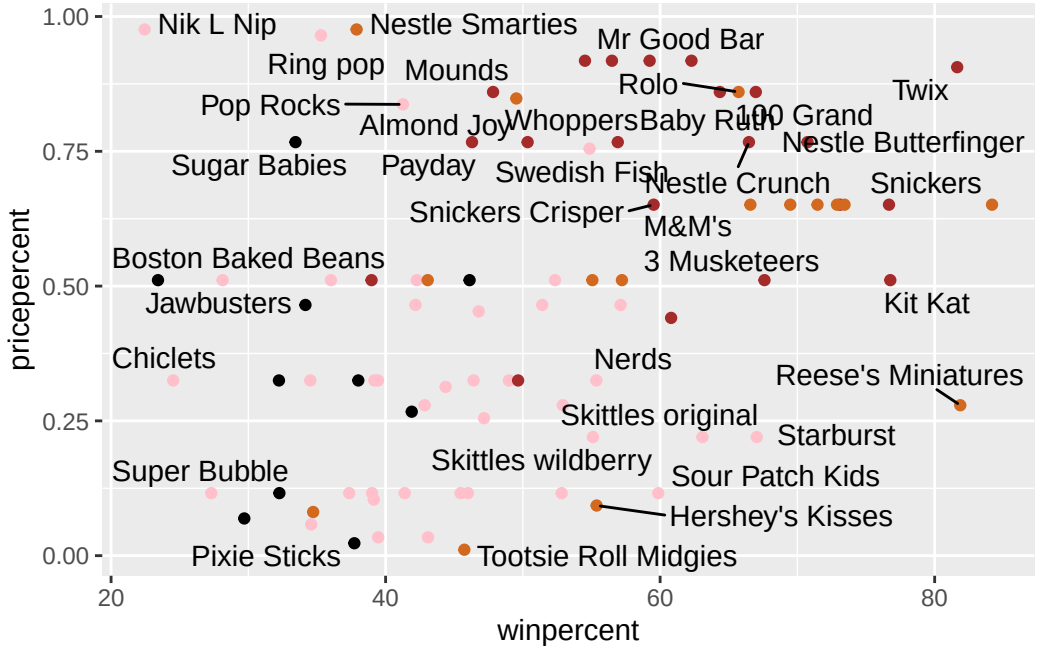
```
ggplot(candy)+
  aes(winpercent, pricepercent, label=rownames(candy))+
  geom_point(col=my_cols) +
  geom_text()
```

To avoid the overlplotting of all these labels we can use an add on package called ggrepl

```
library(ggrepel)

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

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

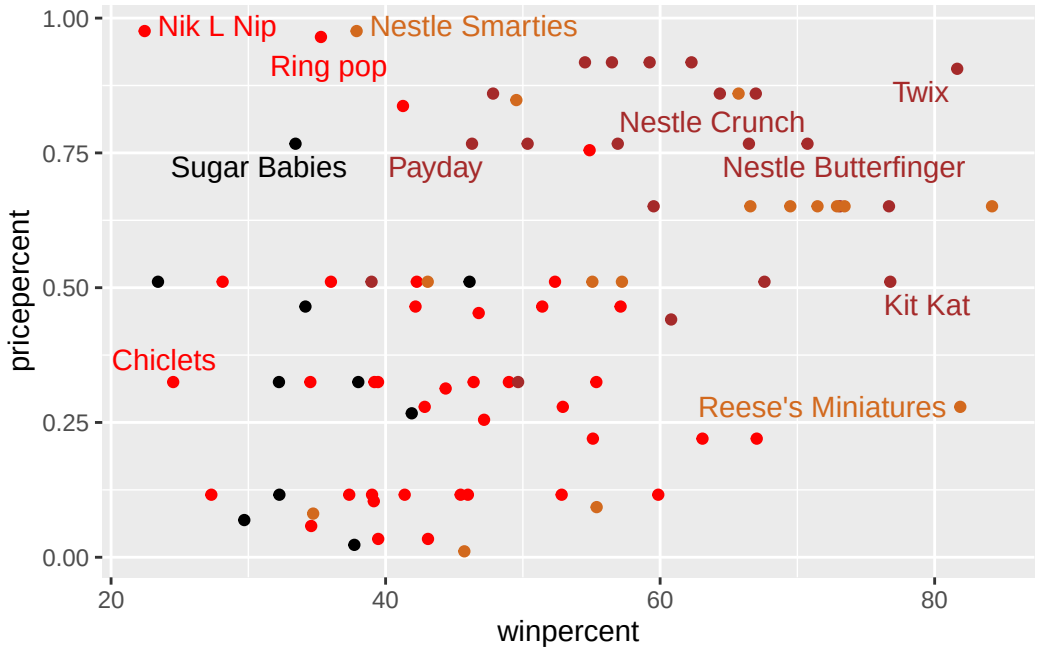


Play with the `max.overlaps` paramater to `geom_text_repel()`

```
# Too hard to see pink (too light) change to "red"
my_cols[as.logical(candy$fruity)] = "red"

ggplot(candy)+
  aes(winpercent, pricepercent, label=rownames(candy))+
  geom_point(col=my_cols) +
  geom_text_repel(max.overlaps = 5, col=my_cols)
```

```
Warning: ggrepel: 74 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

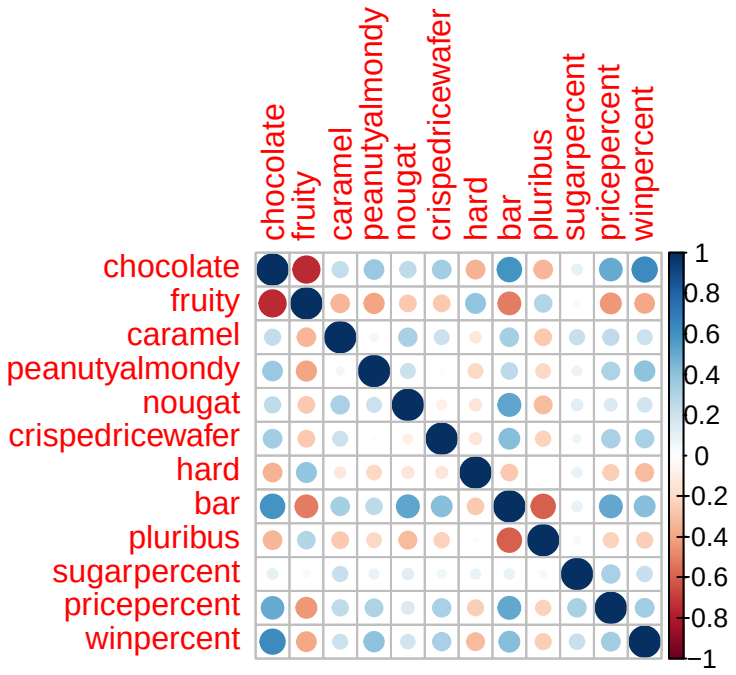


5 Exploring the correlation structure

```
library(corrplot)
```

```
corrplot 0.92 loaded
```

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



on to PCA

The main function for this is called `prcom()` and here we know we need to scale our data eith the `scale = true` arguement.

```
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
```

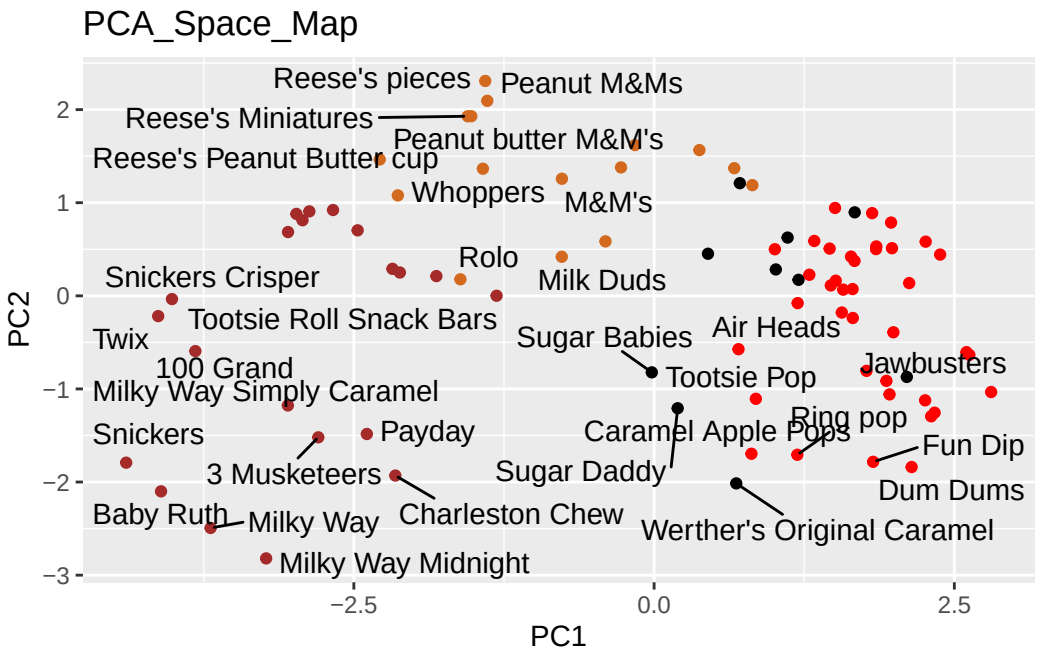
Plot my main PCA score plot with ggplot

```
# Make a new data-frame with our PCA results and candy data
my_data <- cbind(candy, pca$x[,1:3])
```

```
ggplot(my_data) +
  aes(PC1,PC2, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel(fill=my_cols) +
  labs(title= "PCA_Space_Map")
```

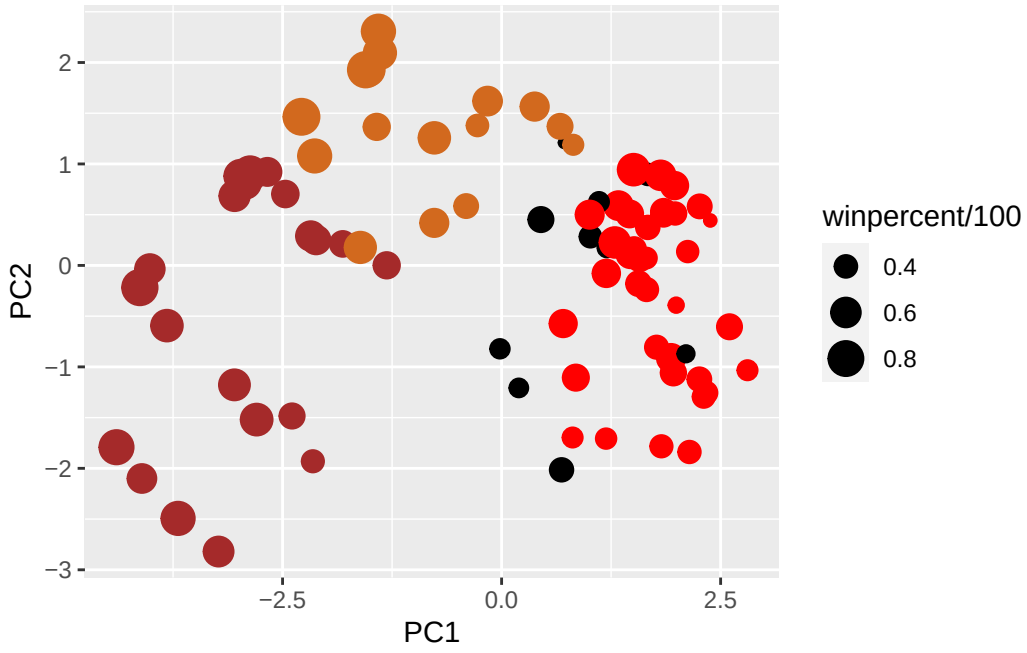
```
Warning in geom_text_repel(fill = my_cols): Ignoring unknown parameters: `fill`
```

```
Warning: ggrepel: 54 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



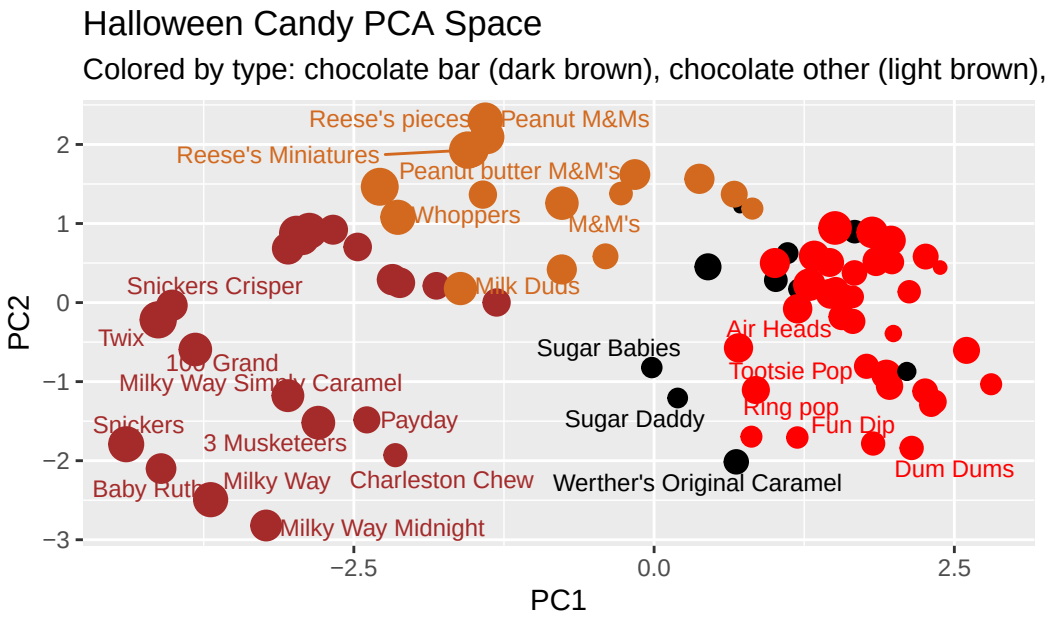
Loading plot

```
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
```

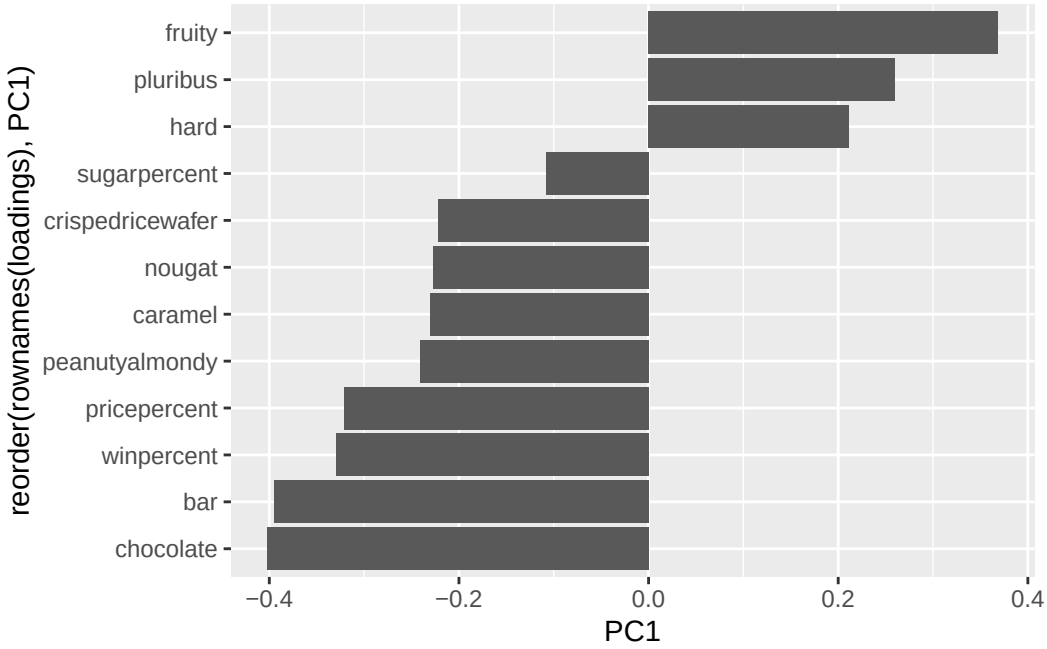



```
p + geom_text_repel(size=3.3, col=my_cols, max.overlaps = 7)  +
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
       subtitle="Colored by type: chocolate bar (dark brown), chocolate other (light brown
       caption="Data from 538")
```

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




```
loadings <- as.data.frame(pca$rotation)
ggplot(loadings)+
  aes(PC1, reorder(rownames(loadings), PC1))+
  geom_col()
```


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