Class 09: Halloween Mini-Project

Melissa Guereca (PID: A16511023)

Here we analyze a candy dataset from the 538 website. this is a CSV fiel from theri GitHub repository.

Data Import

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

> Q1. How many different candy types are in this dataset? Answer: 12

```r
ncol(candy)
```

```
[1] 12
```

> Q2. How many fruity candy types are in the dataset? Answer: 38

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

```
[1] 38
```

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

```
[1] 37
```

Data Exploration

> Q3. What is your favorite candy in the dataset and what is it's winpercent value? Answer: 39.0119

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

```
[1] 39.0119
```

> Q4. What is the winpercent value for “Kit Kat”? Answer: 76.7686

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

```
[1] 76.7686
```

> Q5. What is the winpercent value for “Tootsie Roll Snack Bars”? Answer: 49.6535

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

```
[1] 49.6535
```

> Q. What is the least liked candy? Answer: Nik L Nip

```
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
```

```
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	

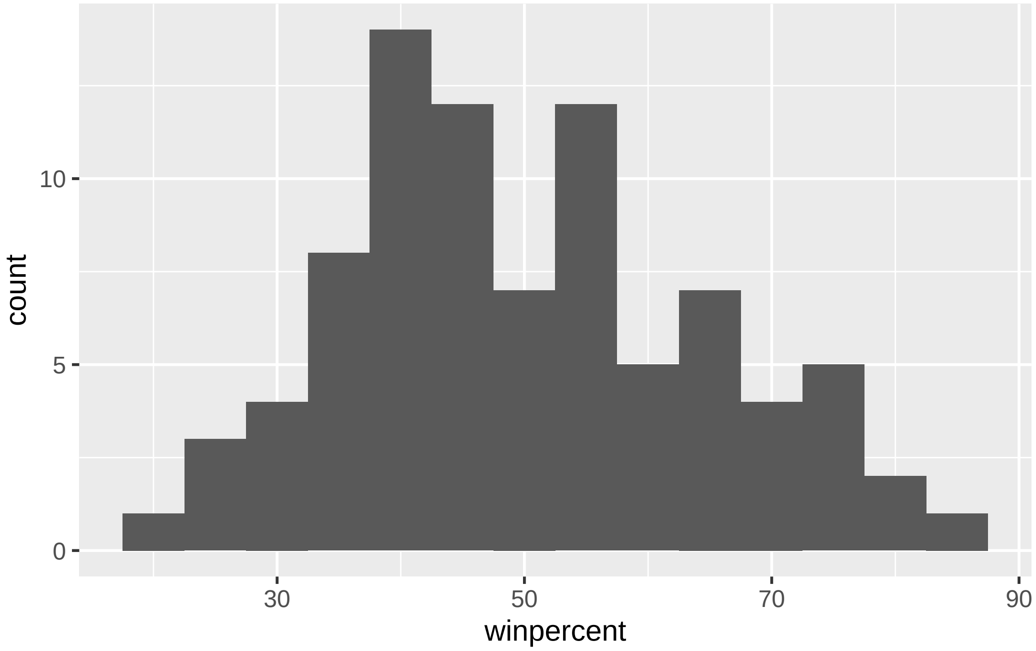
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
sugarpercent	0	1	0.48	0.28	0.01	0.22	0.47	0.73	0.99	
pricepercent	0	1	0.47	0.29	0.01	0.26	0.47	0.65	0.98	
winpercent	0	1	50.32	14.71	22.45	39.14	47.83	59.86	84.18	

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

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

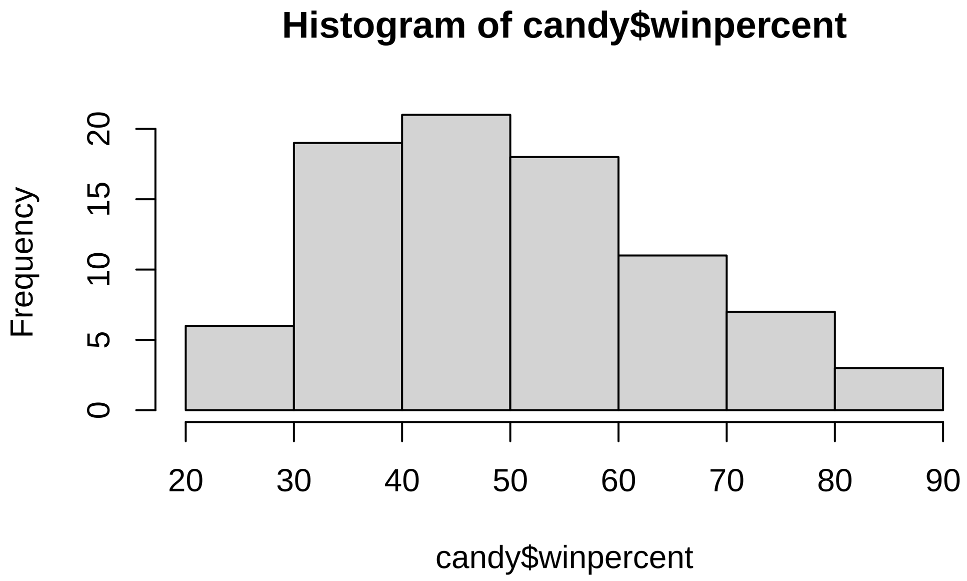
Q8. Plot a histogram of winpercent values.

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



Q9. Is the distribution of winpercent values symmetrical? Answer: No, it is skewed.

```
hist(candy$winpercent, breaks=8)
```



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

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

First find all the chocolate candy and their winpercent values. Next summarize these values into 1 number. Then do the same for fruit candy and compare the numbers.

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

```
[1] 60.92153
```

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

```
[1] 44.11974
```

Q12. Is this difference statistically significant? Answer: Yes

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

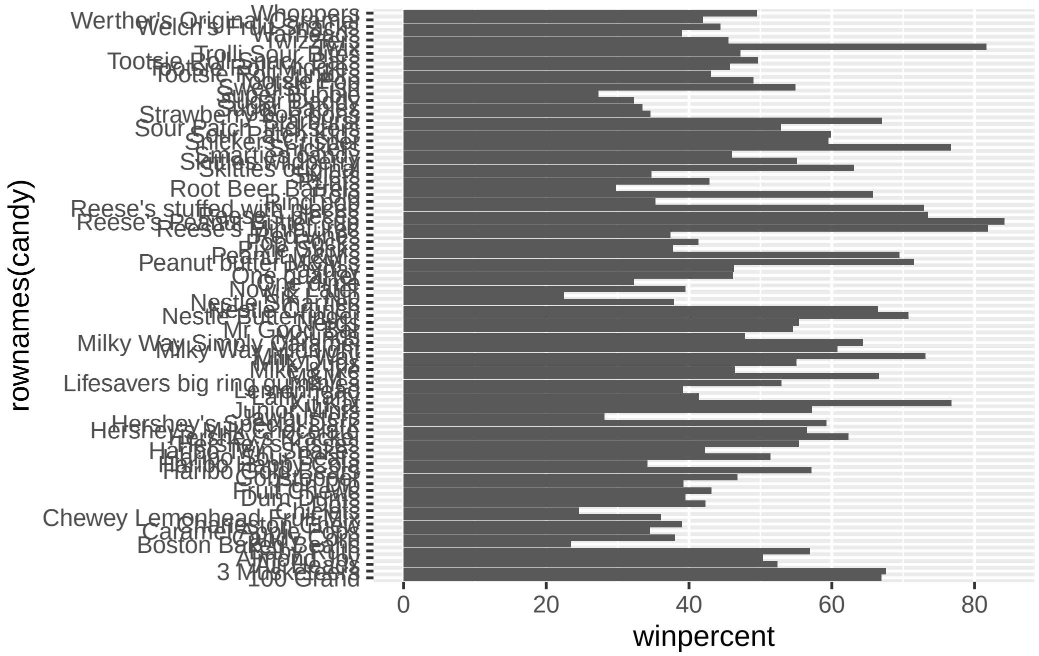
```

	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 
```

> Q13. What are the five least liked candy types in this set? Answer: Jawbusters, Super Bubble, Chiclets, Boston Baked Beans, Nik L Nip

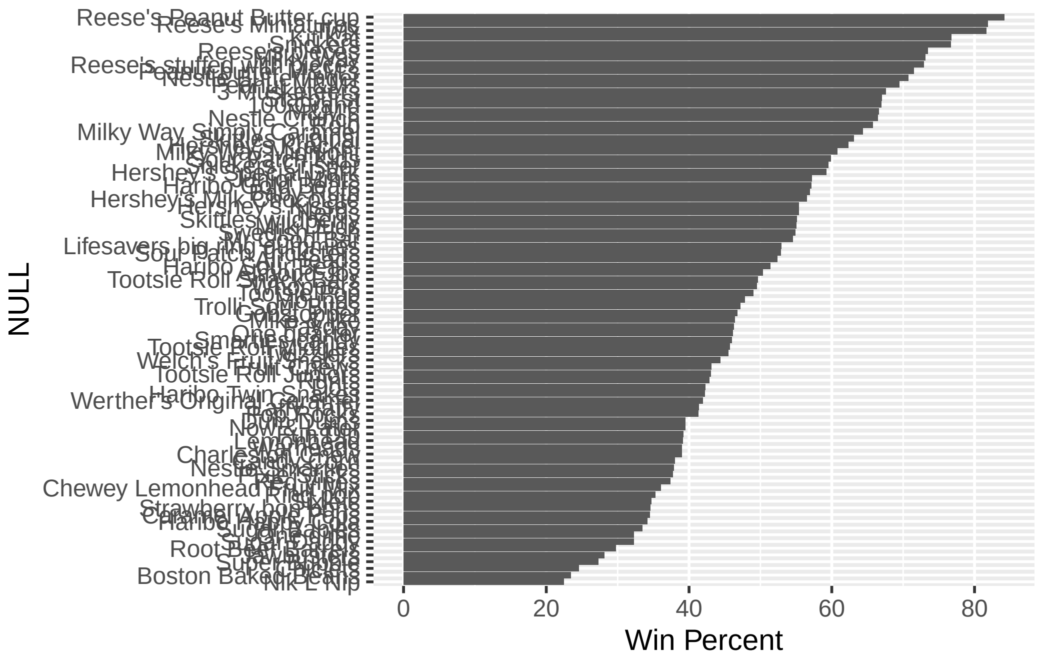
> Q14. What are the top 5 all time favorite candy types out of this set? Answer: Reeses Peanut Butter Cup, Reese's Miniatures, Twix, Kit Kat, Snickers

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

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



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



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

You can insert any image using this markdown syntax.

Add some color to our ggplot, We need to make a custon 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)] = "blue"
my_cols[as.logical(candy$fruity)] = "pink"
my_cols
```

```
 [1] "blue"      "blue"      "black"     "black"     "pink"      "blue"
 [7] "blue"      "black"     "black"     "pink"      "blue"      "pink"
[13] "pink"      "pink"      "pink"      "pink"      "pink"      "pink"
[19] "pink"      "black"     "pink"      "pink"      "chocolate" "blue"
[25] "blue"      "blue"      "pink"      "chocolate" "blue"      "pink"
[31] "pink"      "pink"      "chocolate" "chocolate" "pink"      "chocolate"
[37] "blue"      "blue"      "blue"      "blue"      "blue"      "pink"
[43] "blue"      "blue"      "pink"      "pink"      "blue"      "chocolate"
[49] "black"     "pink"      "pink"      "chocolate" "chocolate" "chocolate"
```

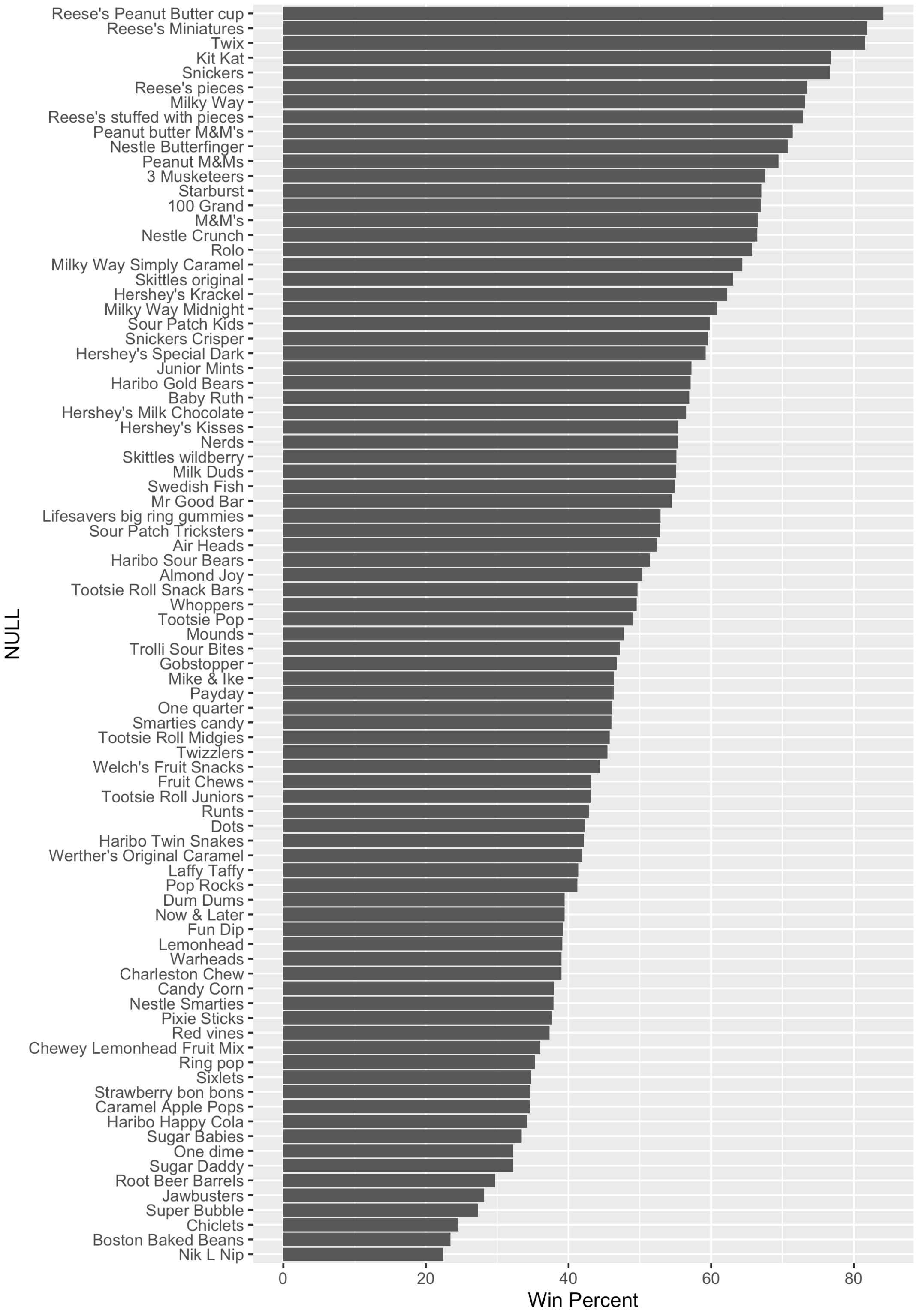
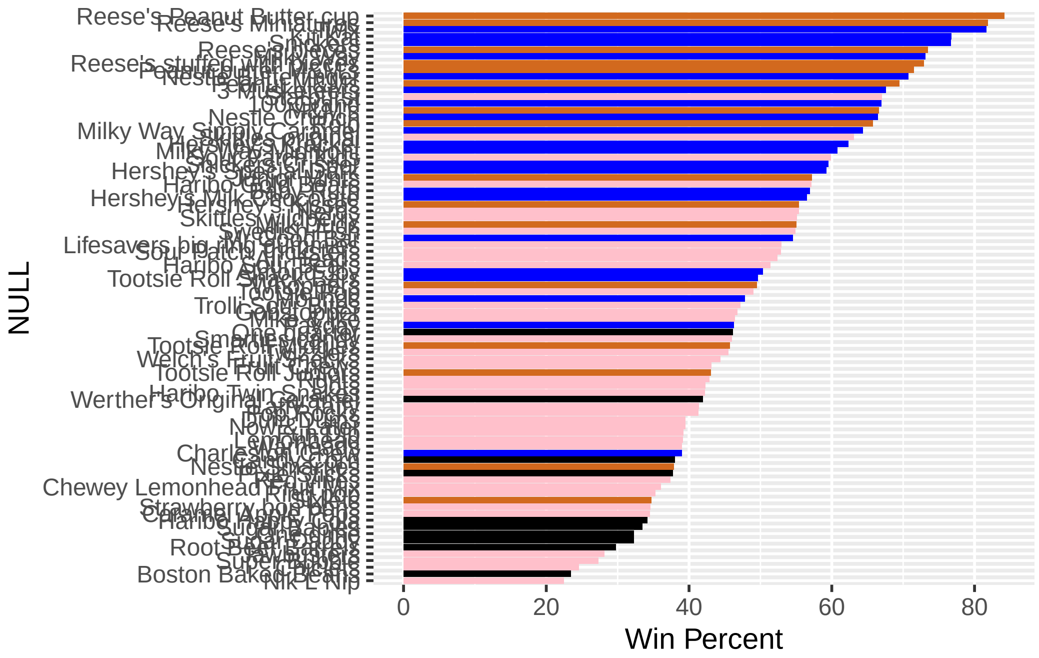


Figure 1: A plot with better aspect ratio

```
[55] "chocolate" "pink"      "chocolate" "black"     "pink"      "chocolate"
[61] "pink"      "pink"      "chocolate" "pink"      "blue"      "blue"
[67] "pink"      "pink"      "pink"      "pink"      "black"     "black"
[73] "pink"      "pink"      "pink"      "chocolate" "chocolate" "blue"
[79] "pink"      "blue"      "pink"      "pink"      "pink"      "black"
[85] "chocolate"
```

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



Q17. What is the worst ranked chocolate candy? Answer: Sixlets

Q18. What is the best ranked fruity candy? Answer:Starbusrt

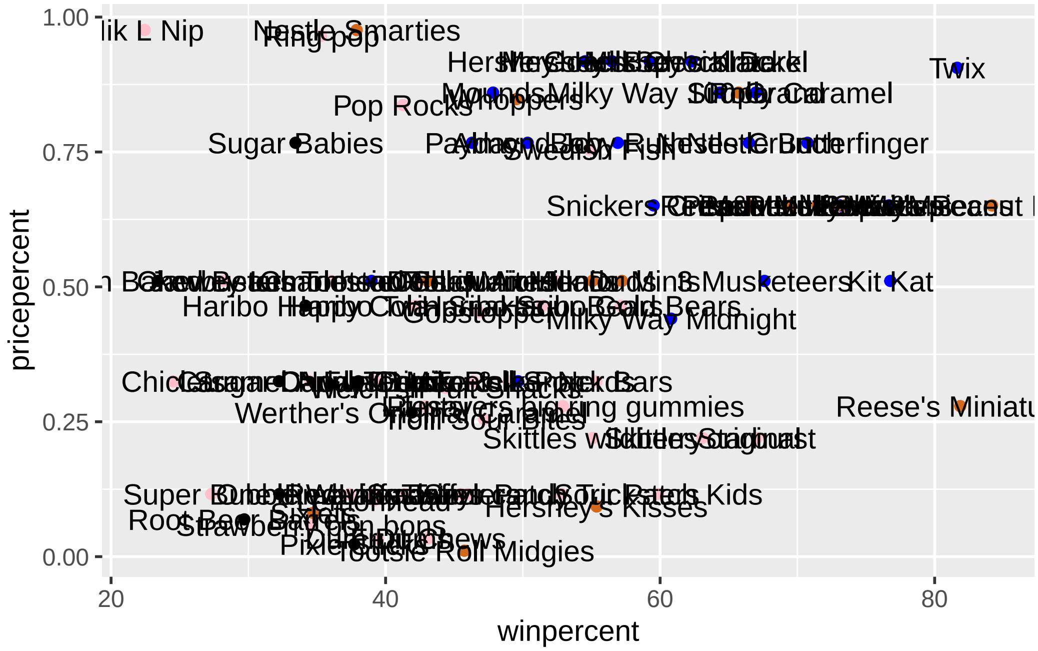
Taking a look at pricepercent

```
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.

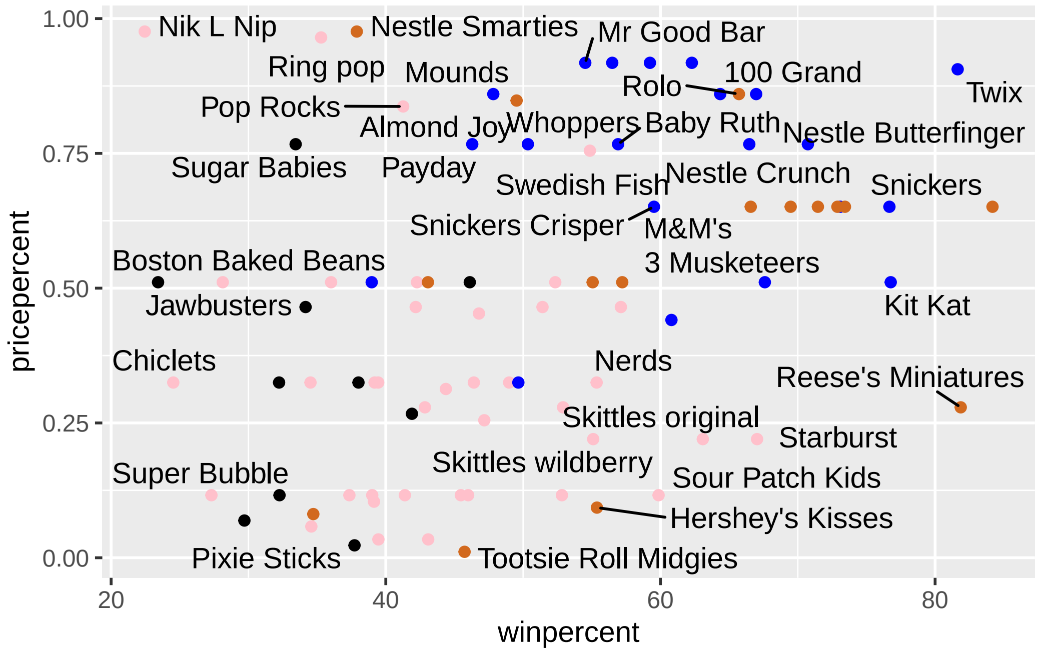
```
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text()
```



To avoid the over plotting of all these labels we ca 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()
```

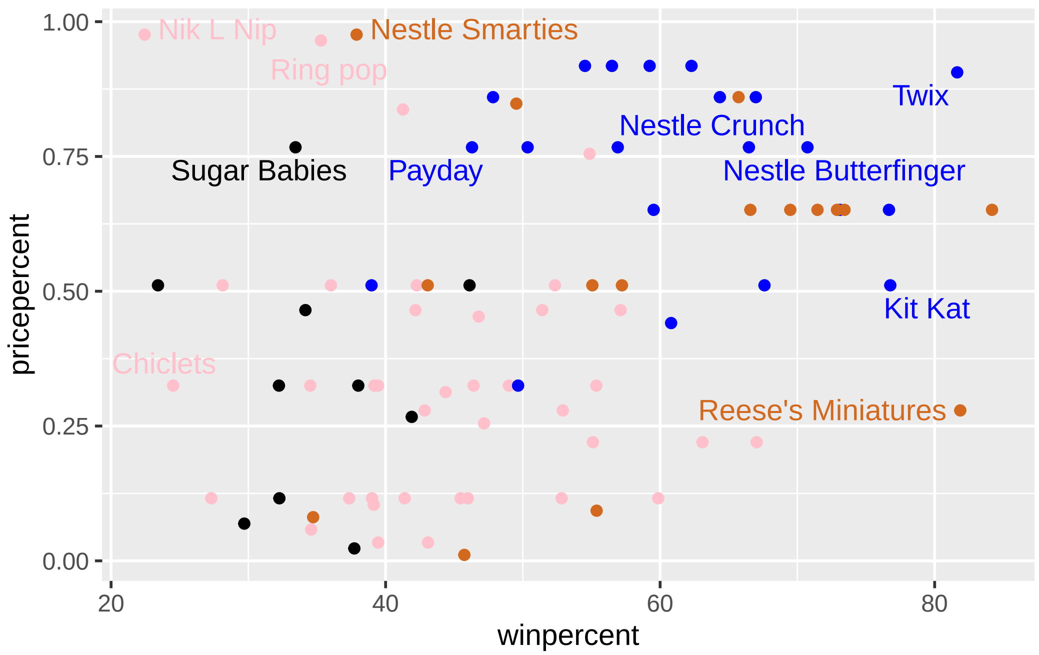
```
Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



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

```
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
```



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? Answer: Reese's Miniatures

```r
ord <- order(candy$winpercent, decreasing = TRUE)
head( candy[ord,c(11,12)], n=20 )
```

```
                            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
Milky Way                          0.651   73.09956
Reese's stuffed with pieces        0.651   72.88790
Peanut butter M&M's                0.651   71.46505
Nestle Butterfinger                0.767   70.73564
Peanut M&Ms                        0.651   69.48379
3 Musketeers                       0.511   67.60294
Starburst                          0.220   67.03763
100 Grand                          0.860   66.97173
M&M's                              0.651   66.57458
```

```
Nestle Crunch                        0.767   66.47068
Rolo                                 0.860   65.71629
Milky Way Simply Caramel             0.860   64.35334
Skittles original                    0.220   63.08514
Hershey's Krackel                    0.918   62.28448
```

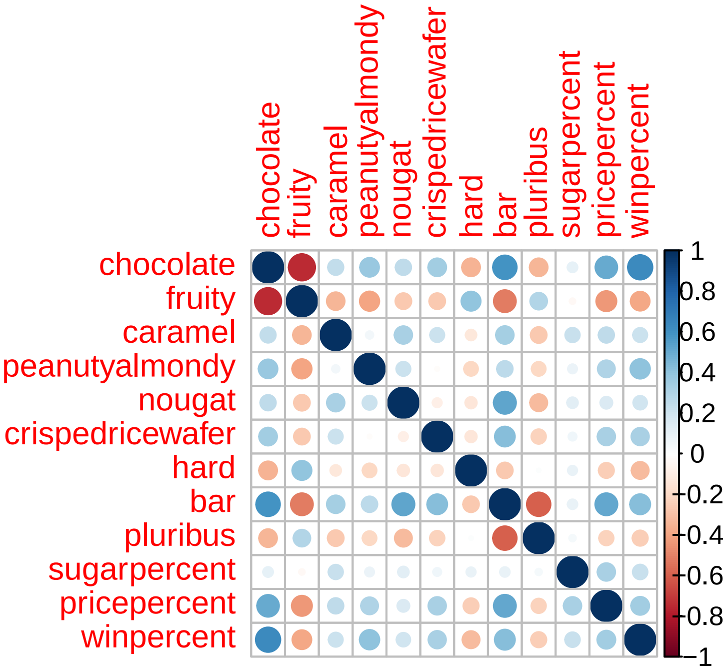
> Q20. What are the top 5 most expensive candy types in the dataset and of these which is the least popular? Answer: Nik L Nip, Nestle Smarties, Ring pop, Mr Good Bar, Hersheys Special Dark. Nik L Nip is the least popular.

5. Exploring

```r
library(corrplot)
```

```
corrplot 0.92 loaded
```

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



> Q22. Examining this plot what two variables are anti-correlated (i.e. have minus values)? Answer: fruity and chocolate
>
> Q23. Similarly, what two variables are most positively correlated? Answer: winpercent and chocolate

On to PCA

The main function for this is called `prcom()` and here we know we need to scale our data with the `scale=TRUE` argument.

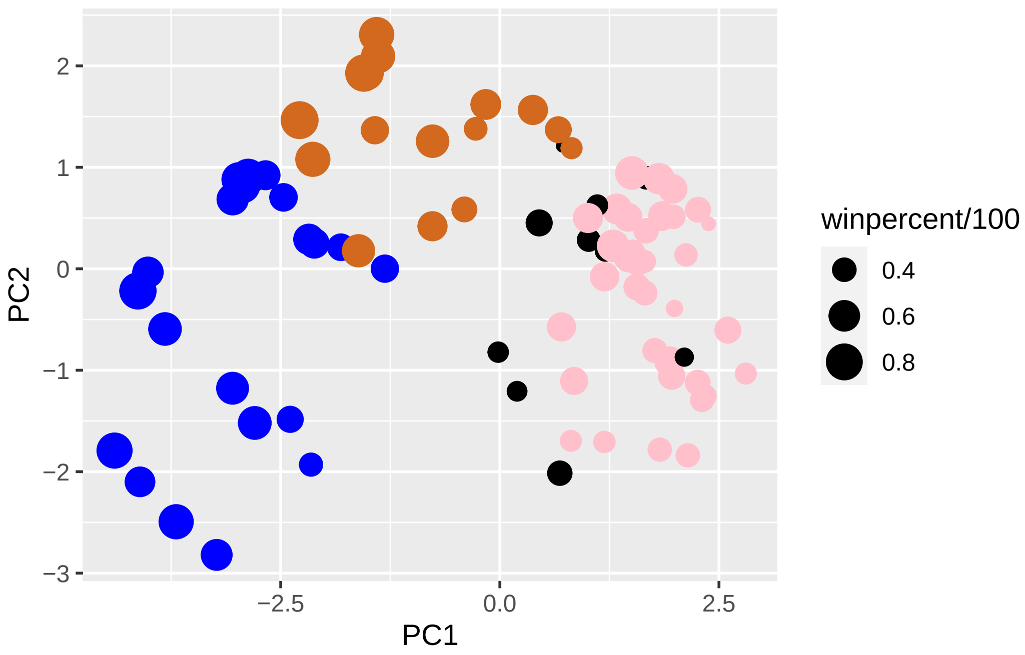
```
pca <- prcomp(candy, scale=TRUE)
summary(pca)
```

```
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
```

Plot my main PCA score plot with ggplot.

```
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
```

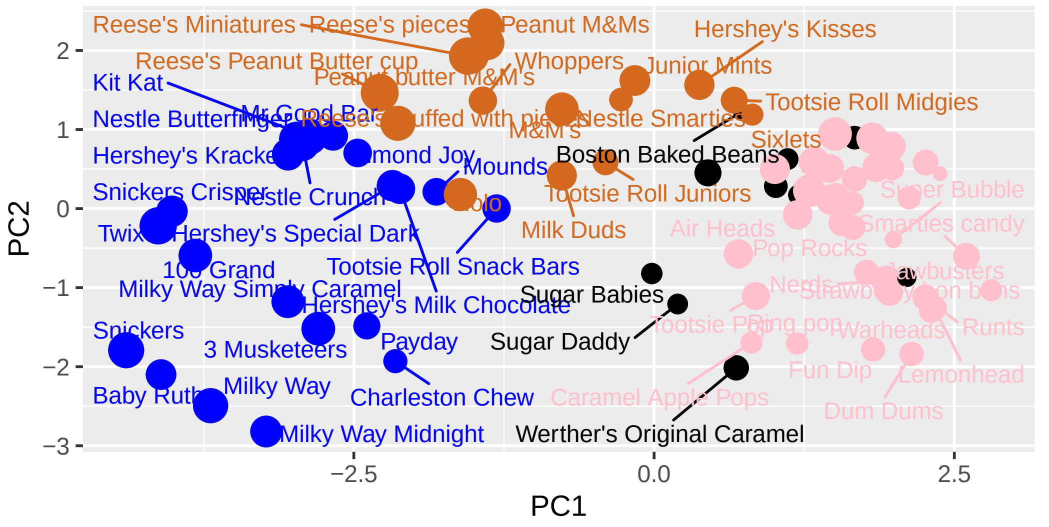


```
p + geom_text_repel(size=3.3, col=my_cols, max.overlaps = 15)  +
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
       subtitle="Colored by type: chocolate bar (dark brown), chocolate other (light brown
       caption="Data from 538")
```

```
Warning: ggrepel: 29 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

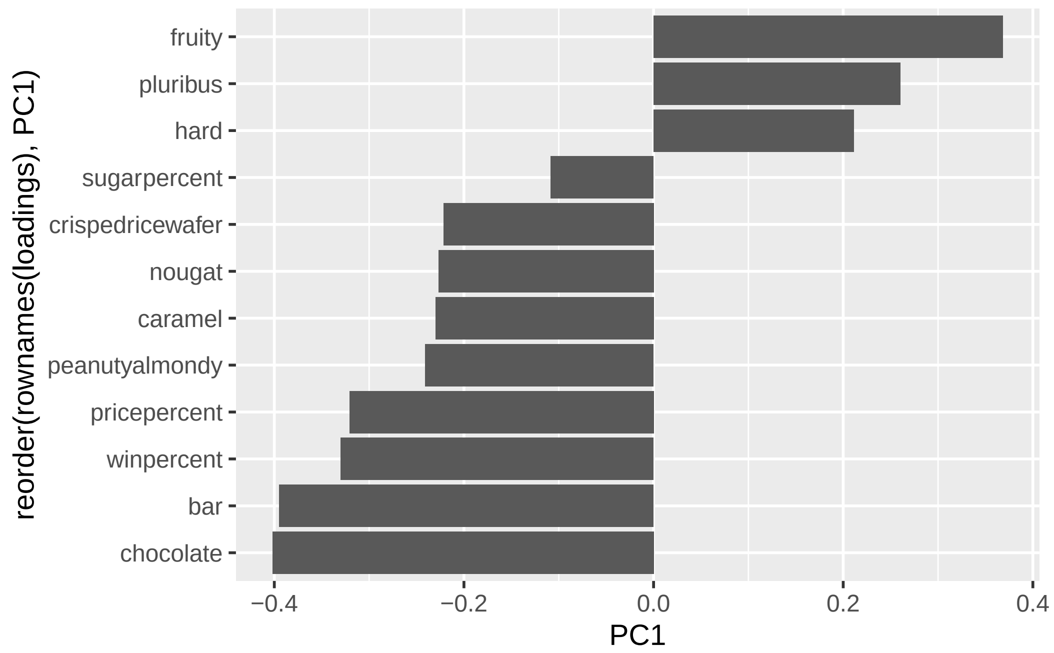
Loadings Plot

```
pca$rotation
```

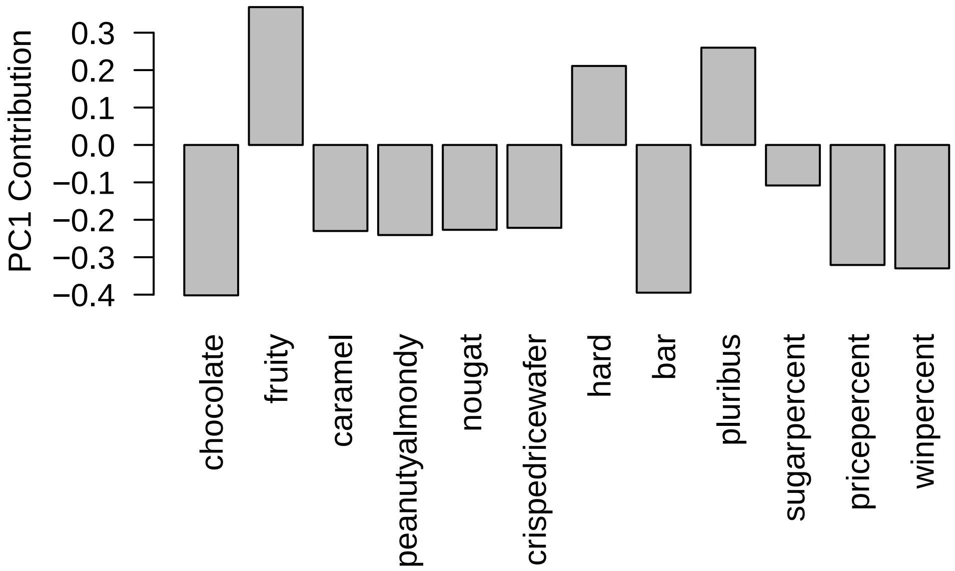
```
                       PC1         PC2         PC3          PC4          PC5
chocolate       -0.4019466  0.21404160  0.01601358 -0.016673032  0.066035846
fruity           0.3683883 -0.18304666 -0.13765612 -0.004479829  0.143535325
caramel         -0.2299709 -0.40349894 -0.13294166 -0.024889542 -0.507301501
peanutyalmondy  -0.2407155  0.22446919  0.18272802  0.466784287  0.399930245
nougat          -0.2268102 -0.47016599  0.33970244  0.299581403 -0.188852418
crispedricewafer -0.2215182  0.09719527 -0.36485542 -0.605594730  0.034652316
hard             0.2111587 -0.43262603 -0.20295368 -0.032249660  0.574557816
bar             -0.3947433 -0.22255618  0.10696092 -0.186914549  0.077794806
pluribus         0.2600041  0.36920922 -0.26813772  0.287246604 -0.392796479
sugarpercent    -0.1083088 -0.23647379 -0.65509692  0.433896248  0.007469103
pricepercent    -0.3207361  0.05883628 -0.33048843  0.063557149  0.043358887
winpercent      -0.3298035  0.21115347 -0.13531766  0.117930997  0.168755073
                        PC6         PC7         PC8          PC9        PC10
chocolate       -0.09018950 -0.08360642 -0.49084856 -0.151651568  0.107661356
fruity          -0.04266105  0.46147889  0.39805802 -0.001248306  0.362062502
caramel         -0.40346502 -0.44274741  0.26963447  0.019186442  0.229799010
```

```
peanutyalmondy   -0.09416259 -0.25710489  0.45771445  0.381068550 -0.145912362
nougat            0.09012643  0.36663902 -0.18793955  0.385278987  0.011323453
crispedricewafer -0.09007640  0.13077042  0.13567736  0.511634999 -0.264810144
hard             -0.12767365 -0.31933477 -0.38881683  0.258154433  0.220779142
bar               0.25307332  0.24192992 -0.02982691  0.091872886 -0.003232321
pluribus          0.03184932  0.04066352 -0.28652547  0.529954405  0.199303452
sugarpercent      0.02737834  0.14721840 -0.04114076 -0.217685759 -0.488103337
pricepercent      0.62908570 -0.14308215  0.16722078 -0.048991557  0.507716043
winpercent       -0.56947283  0.40260385 -0.02936405 -0.124440117  0.358431235
                        PC11        PC12
chocolate         0.10045278  0.69784924
fruity            0.17494902  0.50624242
caramel           0.13515820  0.07548984
peanutyalmondy    0.11244275  0.12972756
nougat           -0.38954473  0.09223698
crispedricewafer -0.22615618  0.11727369
hard              0.01342330 -0.10430092
bar               0.74956878 -0.22010569
pluribus          0.27971527 -0.06169246
sugarpercent      0.05373286  0.04733985
pricepercent     -0.26396582 -0.06698291
winpercent       -0.11251626 -0.37693153
```

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

ggplot(loadings) +
  aes(PC1, reorder(rownames(loadings), PC1)) +
  geom_col()
```



```
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? Answer: fruity, hard, pluribus