Class 9: Halloween Candy Mini-Project

James Garza (PID: A16300772)

Here we analyze a candy dataset from the 538 website. This is a CSV file from their GitHub repository.

Data Import

```
read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-power-rankin
```

	competitorname	chocolate	fruity	caramel	peanutyalmondy	nougat
1	100 Grand	1	0	1	0	0
2	3 Musketeers	1	0	0	0	1
3	One dime	0	0	0	0	0
4	One quarter	0	0	0	0	0
5	Air Heads	0	1	0	0	0
6	Almond Joy	1	0	0	1	0
7	Baby Ruth	1	0	1	1	1
8	Boston Baked Beans	0	0	0	1	0
9	Candy Corn	0	0	0	0	0
10	Caramel Apple Pops	0	1	1	0	0
11	Charleston Chew	1	0	0	0	1
12	Chewey Lemonhead Fruit Mix	0	1	0	0	0
13	Chiclets	0	1	0	0	0
14	Dots	0	1	0	0	0
15	Dum Dums	0	1	0	0	0
16	Fruit Chews	0	1	0	0	0
17	Fun Dip	0	1	0	0	0
18	Gobstopper	0	1	0	0	0
19	Haribo Gold Bears	0	1	0	0	0
20	Haribo Happy Cola	0	0	0	0	0
21	Haribo Sour Bears	0	1	0	0	0
22	Haribo Twin Snakes	0	1	0	0	0

23	Hershey's Kisses	1	0	0	0	0
24	Hershey's Krackel	1	0	0	0	0
25	Hershey's Milk Chocolate	1	0	0	0	0
26	Hershey's Special Dark	1	0	0	0	0
27	Jawbusters	0	1	0	0	0
28	Junior Mints	1	0	0	0	0
29	Kit Kat	1	0	0	0	0
30	Laffy Taffy	0	1	0	0	0
31	Lemonhead	0	1	0	0	0
32	Lifesavers big ring gummies	0	1	0	0	0
33	Peanut butter M&M's	1	0	0	1	0
34	M&M's	1	0	0	0	0
35	Mike & Ike	0	1	0	0	0
36	Milk Duds	1	0	1	0	0
37	Milky Way	1	0	1	0	1
38	Milky Way Midnight	1	0	1	0	1
39	Milky Way Simply Caramel	1	0	1	0	0
40	Mounds	1	0	0	0	0
41	Mr Good Bar	1	0	0	1	0
42	Nerds	0	1	0	0	0
43	Nestle Butterfinger	1	0	0	1	0
44	Nestle Crunch	1	0	0	0	0
45	Nik L Nip	0	1	0	0	0
46	Now & Later	0	1	0	0	0
47	Payday	0	0	0	1	1
48	Peanut M&Ms	1	0	0	1	0
49	Pixie Sticks	0	0	0	0	0
50	Pop Rocks	0	1	0	0	0
51	Red vines	0	1	0	0	0
52	Reese's Miniatures	1	0	0	1	0
53	Reese's Peanut Butter cup	1	0	0	1	0
54	Reese's pieces	1	0	0	1	0
55	Reese's stuffed with pieces	1	0	0	1	0
56	Ring pop	0	1	0	0	0
57	Rolo	1	0	1	0	0
58	Root Beer Barrels	0	0	0	0	0
59	Runts	0	1	0	0	0
60	Sixlets	1	0	0	0	0
61	Skittles original	0	1	0	0	0
62	Skittles wildberry	0	1	0	0	0
63	Nestle Smarties	1	0	0	0	0
64	Smarties candy	0	1	0	0	0
65	Snickers	1	0	1	1	1

66	Snickers Crisper	1	0	1	1	0		
67	Sour Patch Kids	0	1	0	0	0		
68	Sour Patch Tricksters	0	1	0	0	0		
69	Starburst	0	1	0	0	0		
70	Strawberry bon bons	0	1	0	0	0		
71	Sugar Babies	0	0	1	0	0		
72	Sugar Daddy	0	0	1	0	0		
73	Super Bubble	0	1	0	0	0		
74	Swedish Fish	0	1	0	0	0		
75	Tootsie Pop	1	1	0	0	0		
76	Tootsie Roll Juniors	1	0	0	0	0		
77	Tootsie Roll Midgies	1	0	0	0	0		
78	Tootsie Roll Snack Bars	1	0	0	0	0		
79	Trolli Sour Bites	0	1	0	0	0		
80	Twix	1	0	1	0	0		
81	Twizzlers	0	1	0	0	0		
82	Warheads	0	1	0	0	0		
83	Welch's Fruit Snacks	0	1	0	0	0		
84	Werther's Original Caramel	0	0	1	0	0		
85	Whoppers	1	0	0	0	0		

	crispedricewafer	hard	bar	pluribus	sugarpercent	pricepercent	winpercent
1	1	0	1	0	0.732	0.860	66.97173
2	0	0	1	0	0.604	0.511	67.60294
3	0	0	0	0	0.011	0.116	32.26109
4	0	0	0	0	0.011	0.511	46.11650
5	0	0	0	0	0.906	0.511	52.34146
6	0	0	1	0	0.465	0.767	50.34755
7	0	0	1	0	0.604	0.767	56.91455
8	0	0	0	1	0.313	0.511	23.41782
9	0	0	0	1	0.906	0.325	38.01096
10	0	0	0	0	0.604	0.325	34.51768
11	0	0	1	0	0.604	0.511	38.97504
12	0	0	0	1	0.732	0.511	36.01763
13	0	0	0	1	0.046	0.325	24.52499
14	0	0	0	1	0.732	0.511	42.27208
15	0	1	0	0	0.732	0.034	39.46056
16	0	0	0	1	0.127	0.034	43.08892
17	0	1	0	0	0.732	0.325	39.18550
18	0	1	0	1	0.906	0.453	46.78335
19	0	0	0	1	0.465	0.465	57.11974
20	0	0	0	1	0.465	0.465	34.15896
21	0	0	0	1	0.465	0.465	51.41243
22	0	0	0	1	0.465	0.465	42.17877

23	0	0	0	1	0.127	0.093	55.37545
24	1	0	1	0	0.430	0.918	62.28448
25	0	0	1	0	0.430	0.918	56.49050
26	0	0	1	0	0.430	0.918	59.23612
27	0	1	0	1	0.093	0.511	28.12744
28	0	0	0	1	0.197	0.511	57.21925
29	1	0	1	0	0.313	0.511	76.76860
30	0	0	0	0	0.220	0.116	41.38956
31	0	1	0	0	0.046	0.104	39.14106
32	0	0	0	0	0.267	0.279	52.91139
33	0	0	0	1	0.825	0.651	71.46505
34	0	0	0	1	0.825	0.651	66.57458
35	0	0	0	1	0.872	0.325	46.41172
36	0	0	0	1	0.302	0.511	55.06407
37	0	0	1	0	0.604	0.651	73.09956
38	0	0	1	0	0.732	0.441	60.80070
39	0	0	1	0	0.965	0.860	64.35334
40	0	0	1	0	0.313	0.860	47.82975
41	0	0	1	0	0.313	0.918	54.52645
42	0	1	0	1	0.848	0.325	55.35405
43	0	0	1	0	0.604	0.767	70.73564
44	1	0	1	0	0.313	0.767	66.47068
45	0	0	0	1	0.197	0.976	22.44534
46	0	0	0	1	0.220	0.325	39.44680
47	0	0	1	0	0.465	0.767	46.29660
48	0	0	0	1	0.593	0.651	69.48379
49	0	0	0	1	0.093	0.023	37.72234
50	0	1	0	1	0.604	0.837	41.26551
51	0	0	0	1	0.581	0.116	37.34852
52	0	0	0	0	0.034	0.279	81.86626
53	0	0	0	0	0.720	0.651	84.18029
54	0	0	0	1	0.406	0.651	73.43499
55	0	0	0	0	0.988	0.651	72.88790
56	0	1	0	0	0.732	0.965	35.29076
57	0	0	0	1	0.860	0.860	65.71629
58	0	1	0	1	0.732	0.069	29.70369
59	0	1	0	1	0.872	0.279	42.84914
60	0	0	0	1	0.220	0.081	34.72200
61	0	0	0	1	0.941	0.220	63.08514
62	0	0	0	1	0.941	0.220	55.10370
63	0	0	0	1	0.267	0.976	37.88719
64	0	1	0	1	0.267	0.116	45.99583
65	0	0	1	0	0.546	0.651	76.67378

```
66                1    0   1         0         0.604         0.651   59.52925
67                0    0   0         1         0.069         0.116   59.86400
68                0    0   0         1         0.069         0.116   52.82595
69                0    0   0         1         0.151         0.220   67.03763
70                0    1   0         1         0.569         0.058   34.57899
71                0    0   0         1         0.965         0.767   33.43755
72                0    0   0         0         0.418         0.325   32.23100
73                0    0   0         0         0.162         0.116   27.30386
74                0    0   0         1         0.604         0.755   54.86111
75                0    1   0         0         0.604         0.325   48.98265
76                0    0   0         0         0.313         0.511   43.06890
77                0    0   0         1         0.174         0.011   45.73675
78                0    0   1         0         0.465         0.325   49.65350
79                0    0   0         1         0.313         0.255   47.17323
80                1    0   1         0         0.546         0.906   81.64291
81                0    0   0         0         0.220         0.116   45.46628
82                0    1   0         0         0.093         0.116   39.01190
83                0    0   0         1         0.313         0.313   44.37552
84                0    1   0         0         0.186         0.267   41.90431
85                1    0   0         1         0.872         0.848   49.52411
```

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

candy <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-pow
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?

There are 85 different types of candy in the dataset

```
row(candy)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [1,]    1    1    1    1    1    1    1    1    1     1     1     1
 [2,]    2    2    2    2    2    2    2    2    2     2     2     2
 [3,]    3    3    3    3    3    3    3    3    3     3     3     3
 [4,]    4    4    4    4    4    4    4    4    4     4     4     4
 [5,]    5    5    5    5    5    5    5    5    5     5     5     5
 [6,]    6    6    6    6    6    6    6    6    6     6     6     6
 [7,]    7    7    7    7    7    7    7    7    7     7     7     7
 [8,]    8    8    8    8    8    8    8    8    8     8     8     8
 [9,]    9    9    9    9    9    9    9    9    9     9     9     9
[10,]   10   10   10   10   10   10   10   10   10    10    10    10
[11,]   11   11   11   11   11   11   11   11   11    11    11    11
[12,]   12   12   12   12   12   12   12   12   12    12    12    12
[13,]   13   13   13   13   13   13   13   13   13    13    13    13
[14,]   14   14   14   14   14   14   14   14   14    14    14    14
[15,]   15   15   15   15   15   15   15   15   15    15    15    15
[16,]   16   16   16   16   16   16   16   16   16    16    16    16
[17,]   17   17   17   17   17   17   17   17   17    17    17    17
[18,]   18   18   18   18   18   18   18   18   18    18    18    18
[19,]   19   19   19   19   19   19   19   19   19    19    19    19
[20,]   20   20   20   20   20   20   20   20   20    20    20    20
[21,]   21   21   21   21   21   21   21   21   21    21    21    21
[22,]   22   22   22   22   22   22   22   22   22    22    22    22
[23,]   23   23   23   23   23   23   23   23   23    23    23    23
[24,]   24   24   24   24   24   24   24   24   24    24    24    24
[25,]   25   25   25   25   25   25   25   25   25    25    25    25
[26,]   26   26   26   26   26   26   26   26   26    26    26    26
[27,]   27   27   27   27   27   27   27   27   27    27    27    27
[28,]   28   28   28   28   28   28   28   28   28    28    28    28
[29,]   29   29   29   29   29   29   29   29   29    29    29    29
[30,]   30   30   30   30   30   30   30   30   30    30    30    30
[31,]   31   31   31   31   31   31   31   31   31    31    31    31
[32,]   32   32   32   32   32   32   32   32   32    32    32    32
[33,]   33   33   33   33   33   33   33   33   33    33    33    33
[34,]   34   34   34   34   34   34   34   34   34    34    34    34
[35,]   35   35   35   35   35   35   35   35   35    35    35    35
[36,]   36   36   36   36   36   36   36   36   36    36    36    36
[37,]   37   37   37   37   37   37   37   37   37    37    37    37
```

[38,]	38	38	38	38	38	38	38	38	38	38	38	38
[39,]	39	39	39	39	39	39	39	39	39	39	39	39
[40,]	40	40	40	40	40	40	40	40	40	40	40	40
[41,]	41	41	41	41	41	41	41	41	41	41	41	41
[42,]	42	42	42	42	42	42	42	42	42	42	42	42
[43,]	43	43	43	43	43	43	43	43	43	43	43	43
[44,]	44	44	44	44	44	44	44	44	44	44	44	44
[45,]	45	45	45	45	45	45	45	45	45	45	45	45
[46,]	46	46	46	46	46	46	46	46	46	46	46	46
[47,]	47	47	47	47	47	47	47	47	47	47	47	47
[48,]	48	48	48	48	48	48	48	48	48	48	48	48
[49,]	49	49	49	49	49	49	49	49	49	49	49	49
[50,]	50	50	50	50	50	50	50	50	50	50	50	50
[51,]	51	51	51	51	51	51	51	51	51	51	51	51
[52,]	52	52	52	52	52	52	52	52	52	52	52	52
[53,]	53	53	53	53	53	53	53	53	53	53	53	53
[54,]	54	54	54	54	54	54	54	54	54	54	54	54
[55,]	55	55	55	55	55	55	55	55	55	55	55	55
[56,]	56	56	56	56	56	56	56	56	56	56	56	56
[57,]	57	57	57	57	57	57	57	57	57	57	57	57
[58,]	58	58	58	58	58	58	58	58	58	58	58	58
[59,]	59	59	59	59	59	59	59	59	59	59	59	59
[60,]	60	60	60	60	60	60	60	60	60	60	60	60
[61,]	61	61	61	61	61	61	61	61	61	61	61	61
[62,]	62	62	62	62	62	62	62	62	62	62	62	62
[63,]	63	63	63	63	63	63	63	63	63	63	63	63
[64,]	64	64	64	64	64	64	64	64	64	64	64	64
[65,]	65	65	65	65	65	65	65	65	65	65	65	65
[66,]	66	66	66	66	66	66	66	66	66	66	66	66
[67,]	67	67	67	67	67	67	67	67	67	67	67	67
[68,]	68	68	68	68	68	68	68	68	68	68	68	68
[69,]	69	69	69	69	69	69	69	69	69	69	69	69
[70,]	70	70	70	70	70	70	70	70	70	70	70	70
[71,]	71	71	71	71	71	71	71	71	71	71	71	71
[72,]	72	72	72	72	72	72	72	72	72	72	72	72
[73,]	73	73	73	73	73	73	73	73	73	73	73	73
[74,]	74	74	74	74	74	74	74	74	74	74	74	74
[75,]	75	75	75	75	75	75	75	75	75	75	75	75
[76,]	76	76	76	76	76	76	76	76	76	76	76	76
[77,]	77	77	77	77	77	77	77	77	77	77	77	77
[78,]	78	78	78	78	78	78	78	78	78	78	78	78
[79,]	79	79	79	79	79	79	79	79	79	79	79	79
[80,]	80	80	80	80	80	80	80	80	80	80	80	80

```
[81,]   81   81   81   81   81   81   81   81   81    81    81    81
[82,]   82   82   82   82   82   82   82   82   82    82    82    82
[83,]   83   83   83   83   83   83   83   83   83    83    83    83
[84,]   84   84   84   84   84   84   84   84   84    84    84    84
[85,]   85   85   85   85   85   85   85   85   85    85    85    85
```

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

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

```
[1] 38
```

There are 38 fruit candy

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

```
[1] 37
```

Data Exploration

What is your favorite candy?

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

```
candy["Reese's Peanut Butter cup",]$winpercent
```

```
[1] 84.18029
```

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

```r
x <- c(5, 3, 4, 1)
sort(x) # gives the values in ascending order
```

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

```r
order(x) # indicates the index location
```

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

```r
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
```

Nik L Nip has the lowest win percentage and is the least liked candy.

```r
# install.packages("skimr")
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?

Yes, row 12 (winpercent) appears to be on a percentage scale rather than a scale of 0 to 1. Additionally it is the only variable that has STDEV above 1.

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

```
candy$chocolate
```

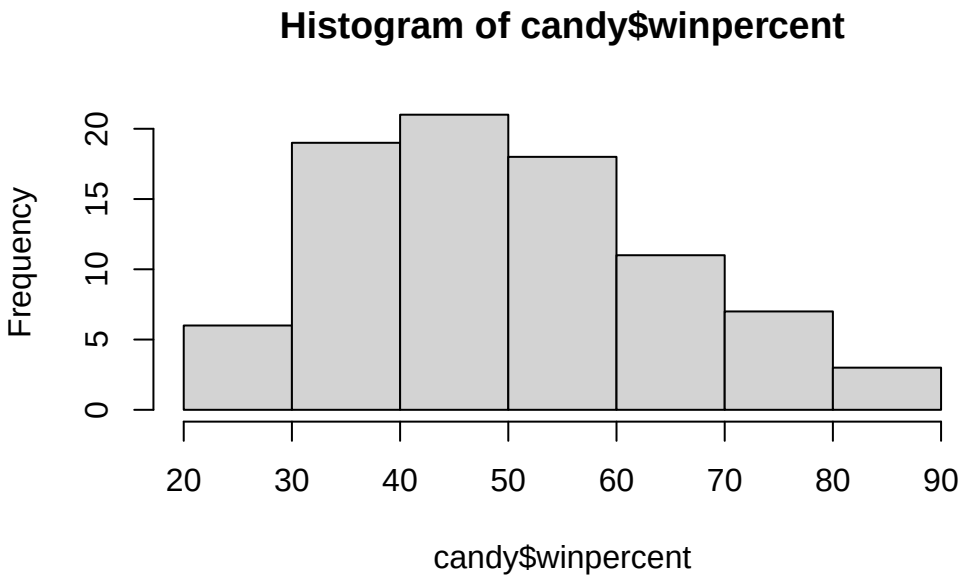
```
 [1] 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 0 0 1 1 0 1 1 1
```

```
 [39] 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1
 [77] 1 1 0 1 0 0 0 0 1
```

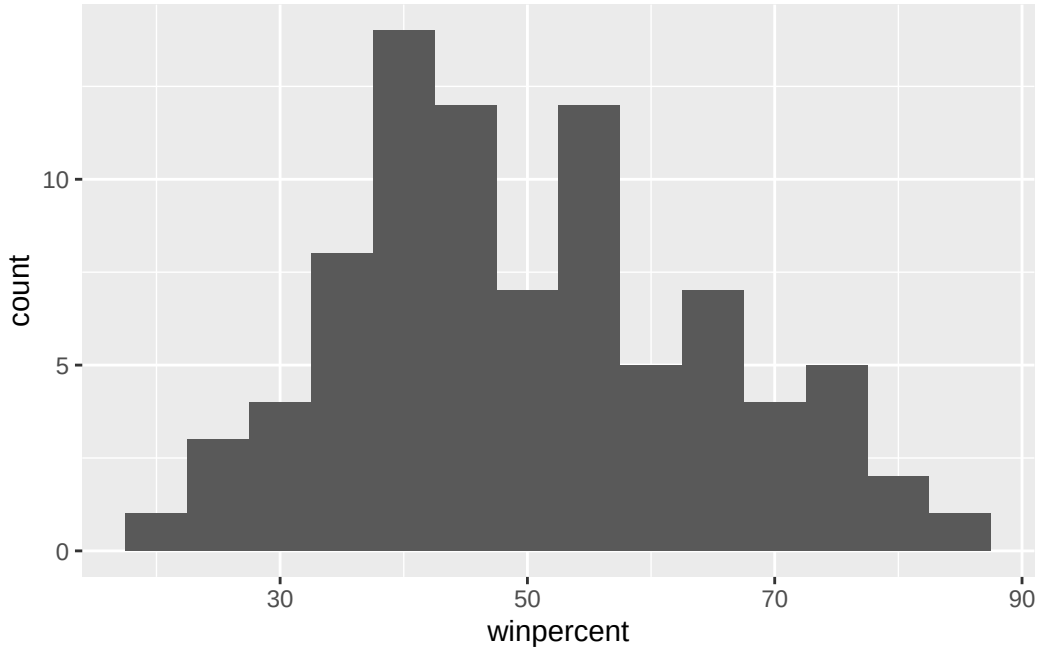
I think a zero and 1 represent "True/False" for the respective categories that they are being counted in where a 1 means true and a 0 means false.

Q8. Plot a histogram of winpercent values

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




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



Q9. Is the distribution of winpercent values symmetrical?

No as it appears to be slightly skewed to the right.

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

The center is below 50%.

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

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

```
[1] 60.92153
```

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

```
[1] 44.11974
```

On average chocolate candy is ranked higher than fruity candy.

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$winpercent[as.logical(candy$fi
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 difference is statistically significant because the p-value is less than 0.05.

Overall Candy Rankings

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

```
head(candy[inds,], 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
```

Nik L Nip, boston baked beans chiclets super bubble and jawbusters are the 5 least liked candies.

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

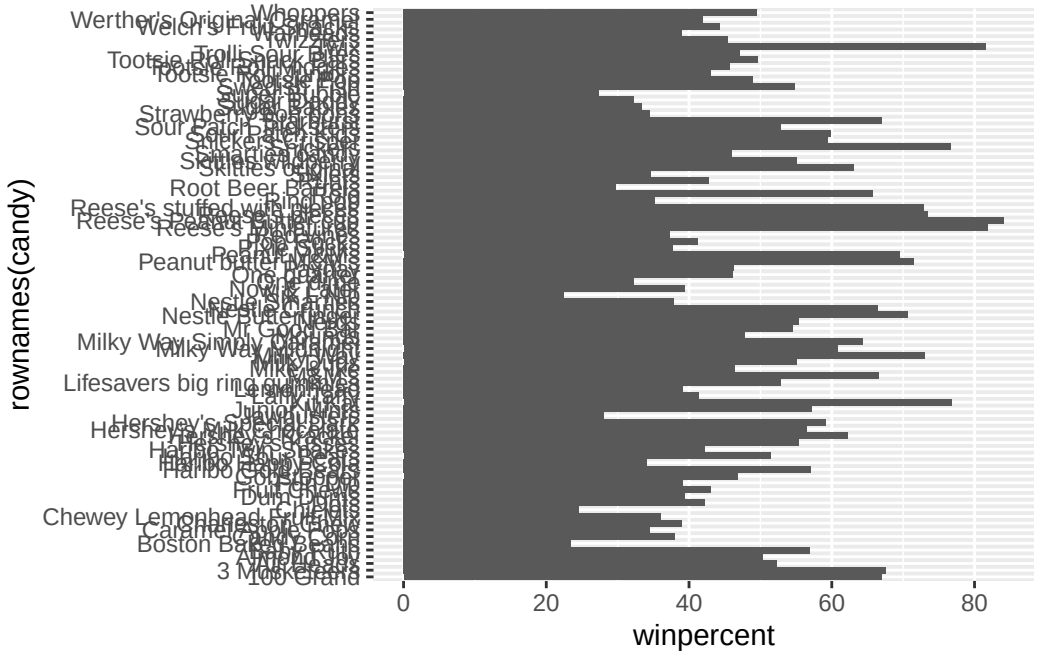
```
tail(candy[inds,], 5)
```

```
                         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
```

The 5 most liked candies are Snickers, Kit Kat, Twix, Reese's Minatures and Reese's Peanut Butter cups.

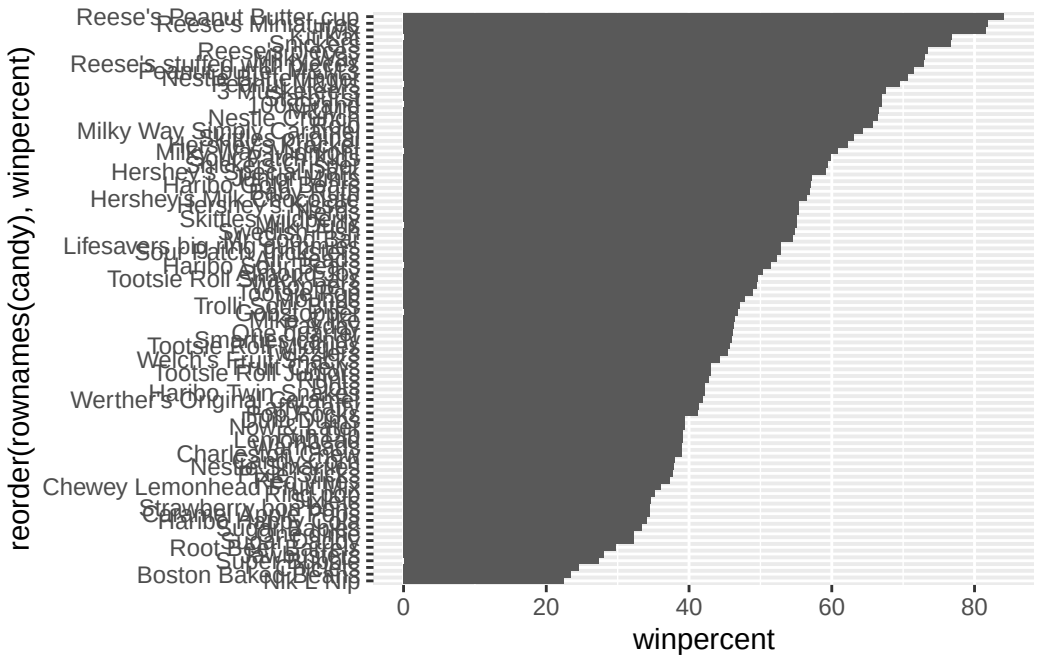
Q15. Make a first barplot of candy ranking based on winpercent values.

```
ggplot(candy) +
  aes(x = winpercent, y = rownames(candy)) +
  geom_col()
```



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

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

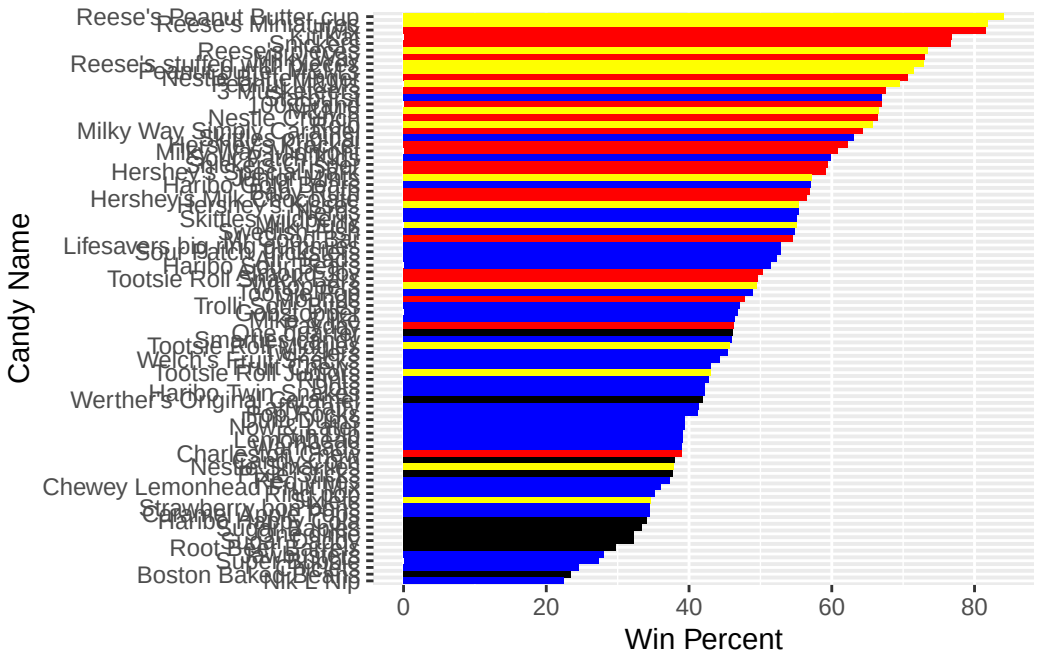


Add some color to our ggplot. We need to make a custom color vector.

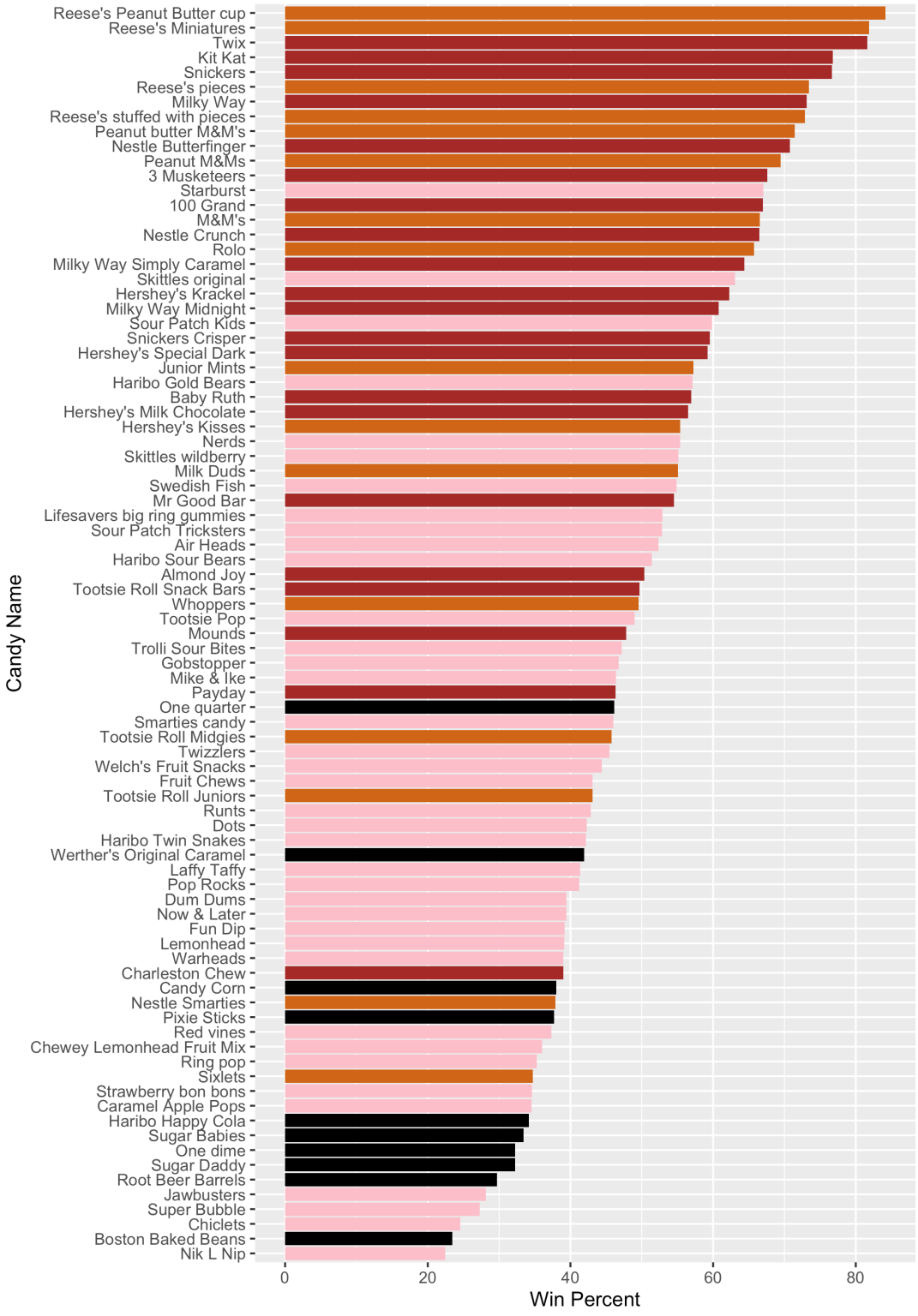
```
# start with all black vector of colors

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

```
ggplot(candy) +
  aes(x = winpercent,
      reorder(rownames(candy),winpercent)) +
  geom_col(fill=my_cols) +
  ylab("Candy Name") + xlab("Win Percent")
```

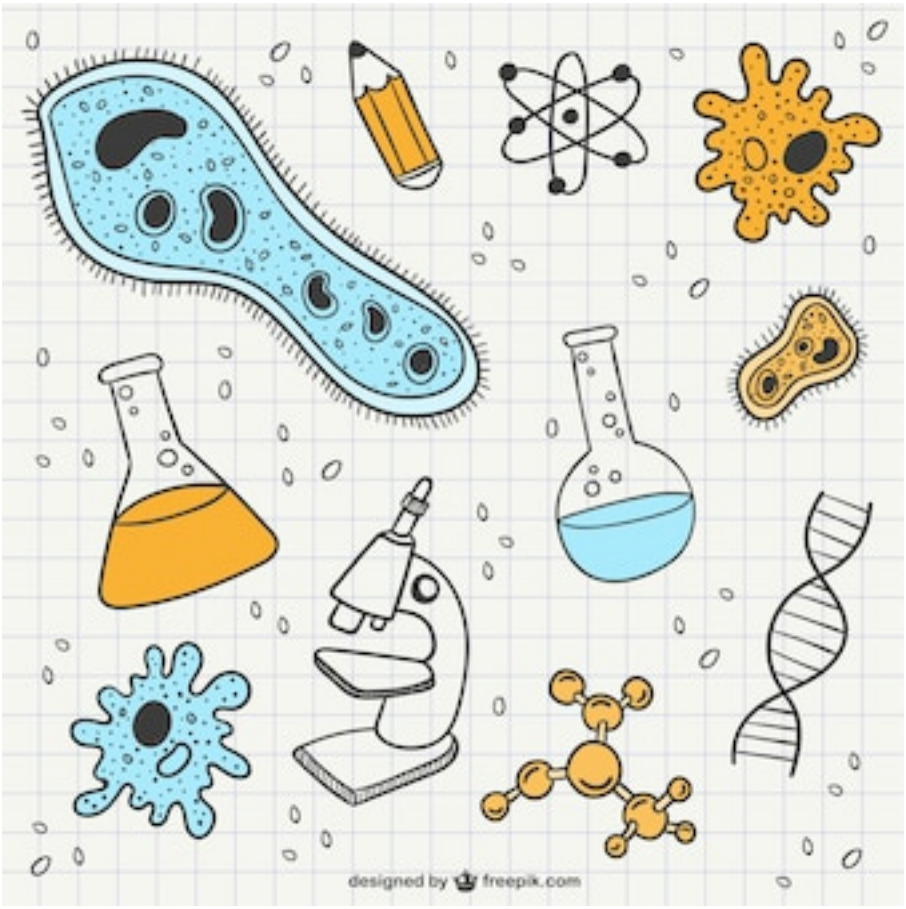



```
# ggsave('barplot1.png', width=7, height=10)
```


Reese's Peanut Butter cup
Reese's Miniatures
Twix
Kit Kat
Snickers
Reese's pieces
Milky Way
Reese's stuffed with pieces
Peanut butter M&M's
Nestle Butterfinger
Peanut M&Ms
3 Musketeers
Starburst
100 Grand
M&M's
Nestle Crunch
Rolo
Milky Way Simply Caramel
Skittles original
Hershey's Krackel
Milky Way Midnight
Sour Patch Kids
Snickers Crisper
Hershey's Special Dark
Junior Mints
Haribo Gold Bears
Baby Ruth
Hershey's Milk Chocolate
Hershey's Kisses
Nerds
Skittles wildberry
Milk Duds
Swedish Fish
Mr Good Bar
Lifesavers big ring gummies
Sour Patch Tricksters
Air Heads
Haribo Sour Bears
Almond Joy
Tootsie Roll Snack Bars
Whoppers
Tootsie Pop
Mounds
Trolli Sour Bites
Gobstopper
Mike & Ike
Payday
One quarter
Smarties candy
Tootsie Roll Midgies
Twizzlers
Welch's Fruit Snacks
Fruit Chews
Tootsie Roll Juniors
Runts
Dots
Haribo Twin Snakes
Werther's Original Caramel
Laffy Taffy
Pop Rocks
Dum Dums
Now & Later
Fun Dip
Lemonhead
Warheads
Charleston Chew
Candy Corn
Nestle Smarties
Pixie Sticks
Red vines
Chewey Lemonhead Fruit Mix
Ring pop
Sixlets
Strawberry bon bons
Caramel Apple Pops
Haribo Happy Cola
Sugar Babies
One dime
Sugar Daddy
Root Beer Barrels
Jawbusters
Super Bubble
Chiclets
Boston Baked Beans
Nik L Nip
Candy Name
0
20
40
60
80
Win Percent

You

can insert any image using this markdown syntax.



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?

Starbursts

Taking a look at pricepercent

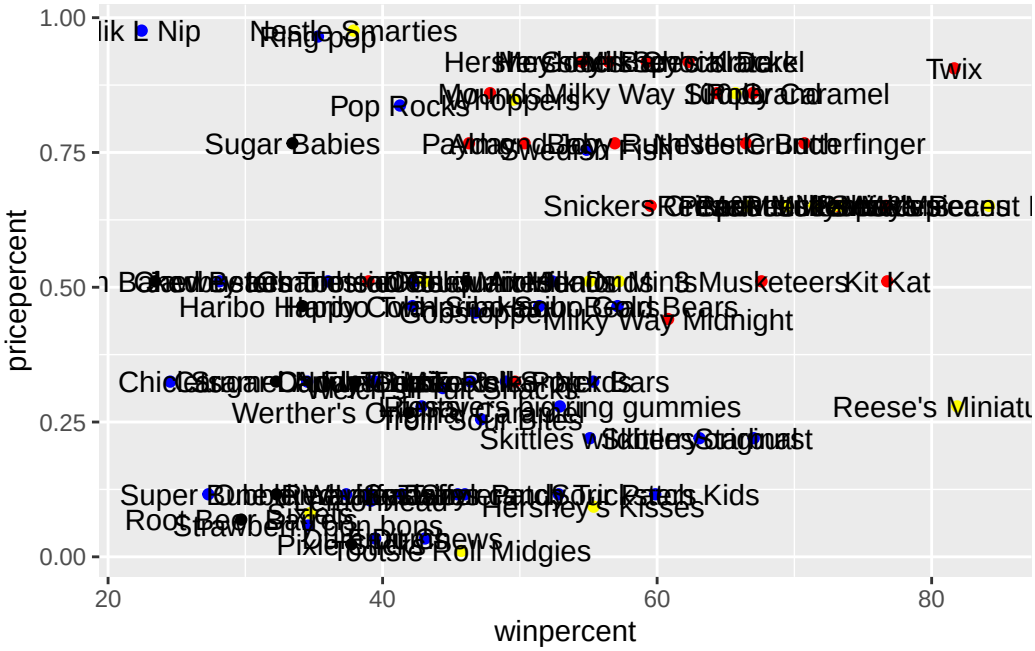
```
# install.packages("ggrepel")
library(ggrepel)
```

```
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 2 variables and then see the best candy for the least amount of money

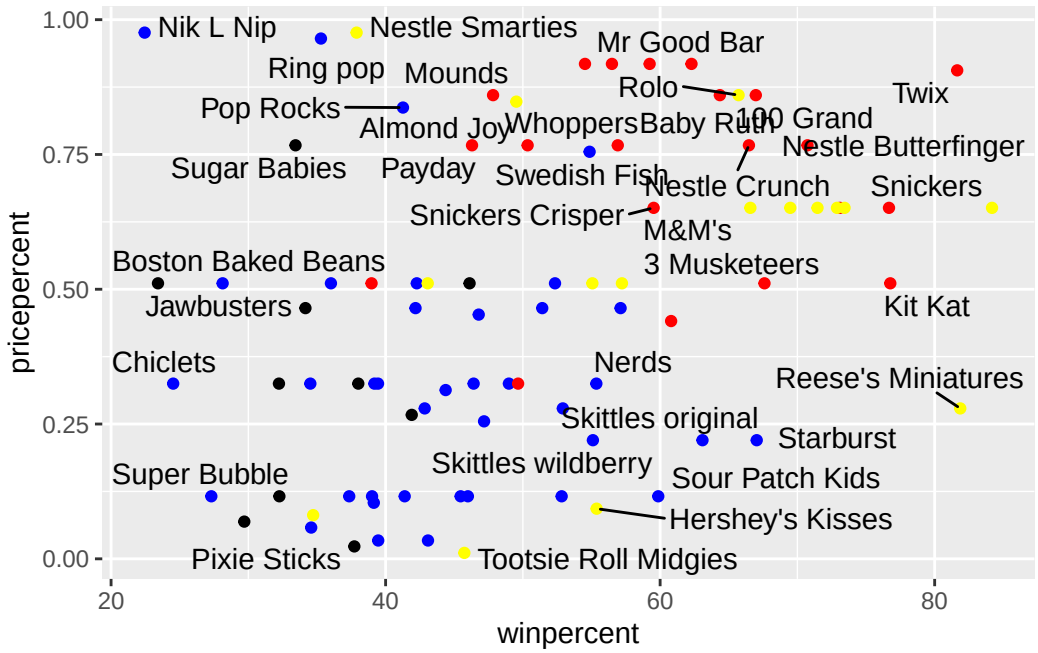
```
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text()
```



to avoid the overplotting of all these labels we can use an add on package called ggrepl

```
# How about a plot of price vs win
ggplot(candy) +
  aes(winpercent, pricepercent, label=rownames(candy)) +
  geom_point(col=my_cols) +
  geom_text_repel()
```

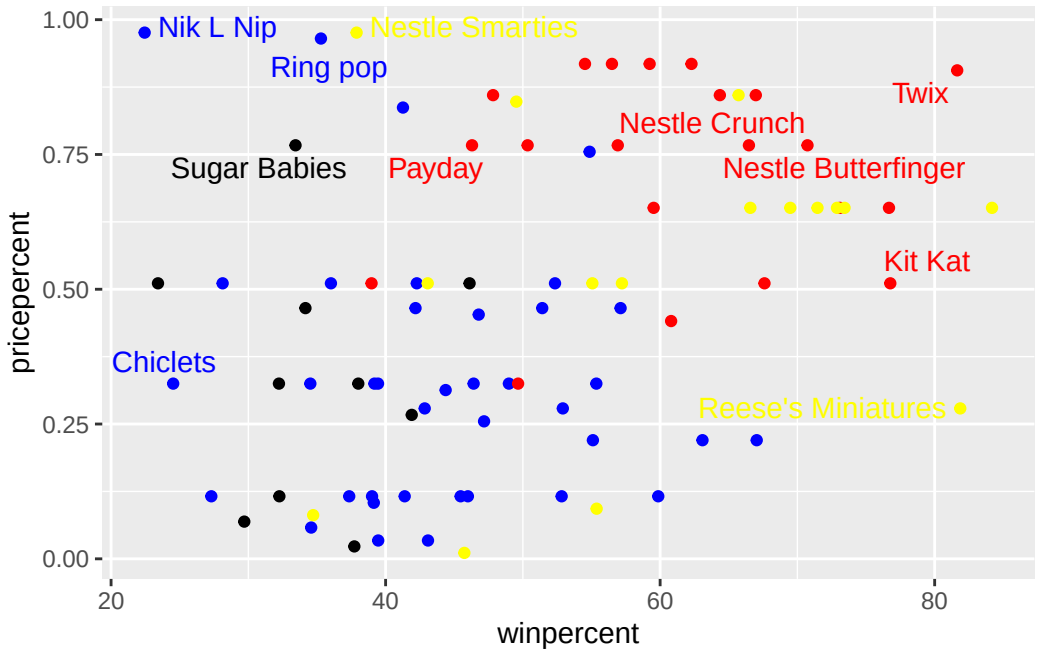
```
Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



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

```
# 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, max.overlaps = 5)
```

```
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?

Reese's minatures offers the highest win percent while offering a low price percent.

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

```
inds_2 <- order(candy$pricepercent)
tail(candy[inds_2,], 5)
```

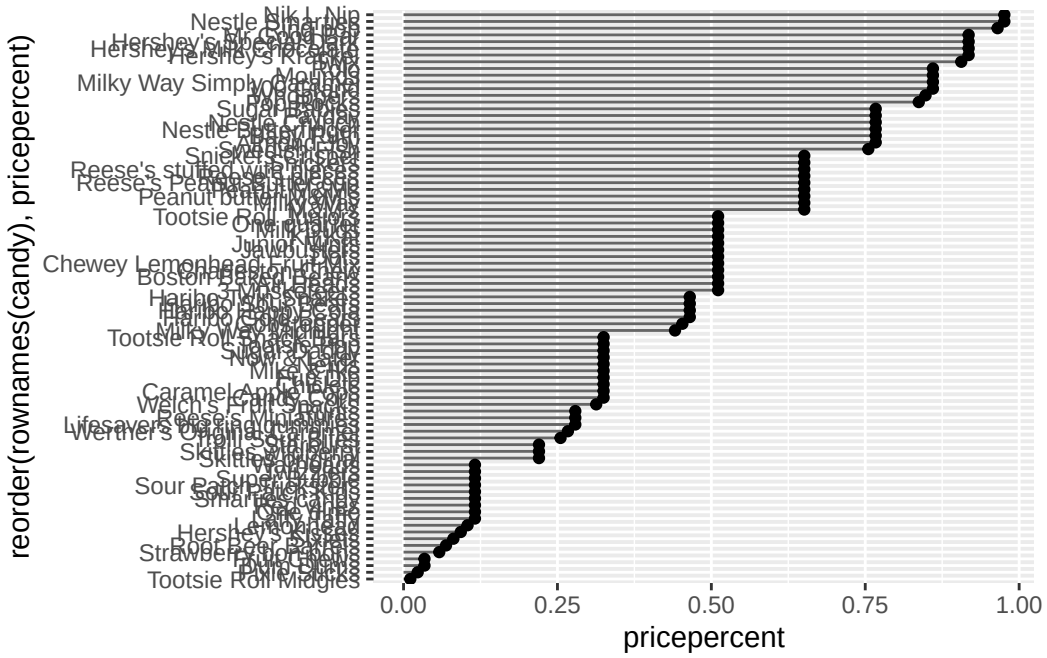
```
                      chocolate fruity caramel peanutyalmondy nougat
Hershey's Special Dark         1      0       0              0      0
Mr Good Bar                    1      0       0              1      0
Ring pop                       0      1       0              0      0
Nik L Nip                      0      1       0              0      0
Nestle Smarties                1      0       0              0      0
                      crispedricewafer hard bar pluribus sugarpercent
Hershey's Special Dark                0    0   1        0        0.430
Mr Good Bar                           0    0   1        0        0.313
Ring pop                              0    1   0        0        0.732
Nik L Nip                             0    0   0        1        0.197
Nestle Smarties                       0    0   0        1        0.267
```

```
                      pricepercent winpercent
Hershey's Special Dark        0.918   59.23612
Mr Good Bar                   0.918   54.52645
Ring pop                      0.965   35.29076
Nik L Nip                     0.976   22.44534
Nestle Smarties               0.976   37.88719
```

The 5 most expensive candies are nestle smarties, Nik L Nip, Ring pop, Mr Good Bar and hershey's special dark. The least popular is Nik L Nip.

> 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()
```

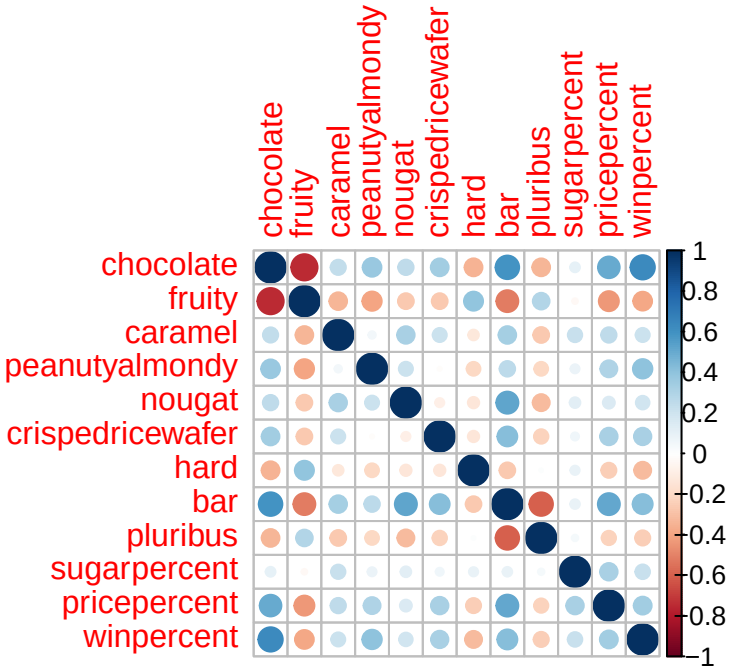


Exploring the correlation structure

```
# install.packages("corrplot")
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)?

two variables that are anti-correlated are fruity and chocolate

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

chocolate and win percent are the most positively correlated variables

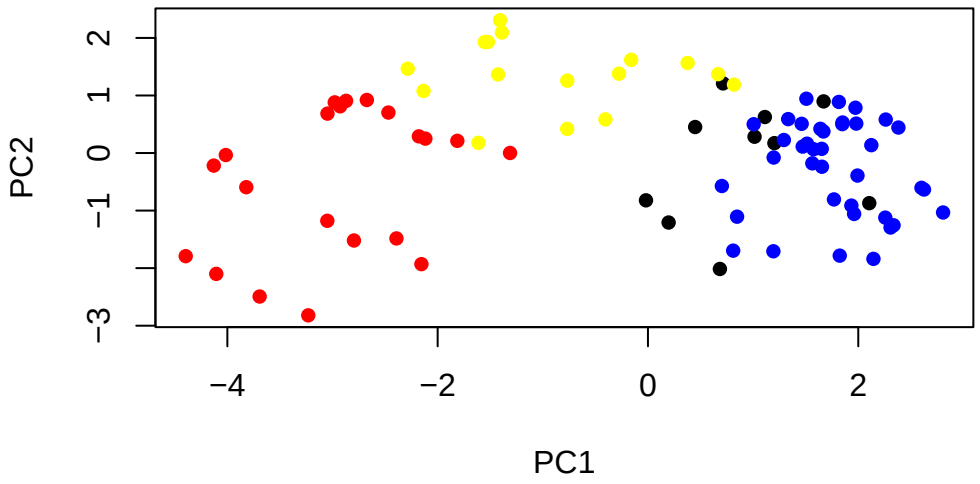
Principal Componenet Analysis

The main function for this is called `prcomp()` 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
```

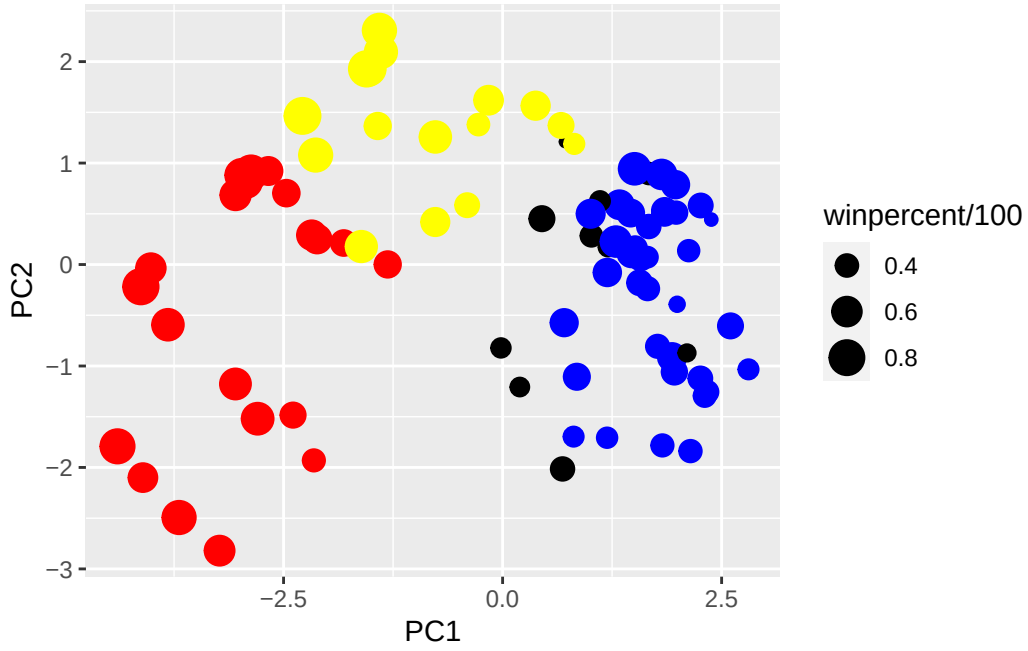
```
plot(pca$x[,1:2], col=my_cols, pch=16)
```



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

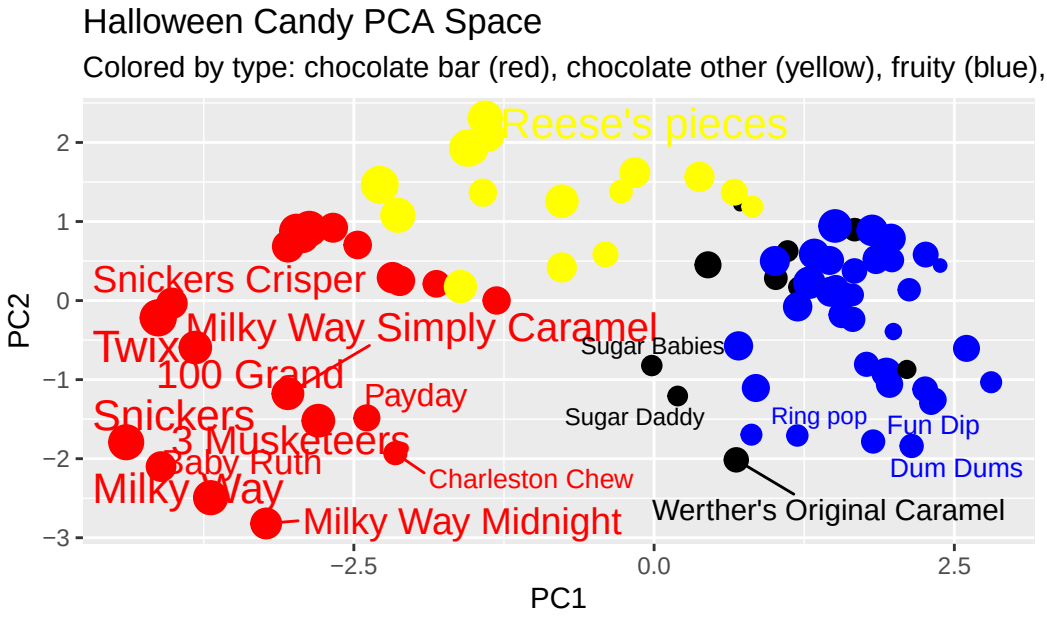
```
p <- ggplot(my_data) +
        aes(x=PC1, y=PC2,
            size=winpercent/100,
            text=rownames(my_data),
            label=rownames(my_data)) +
        geom_point(col=my_cols)

p
```

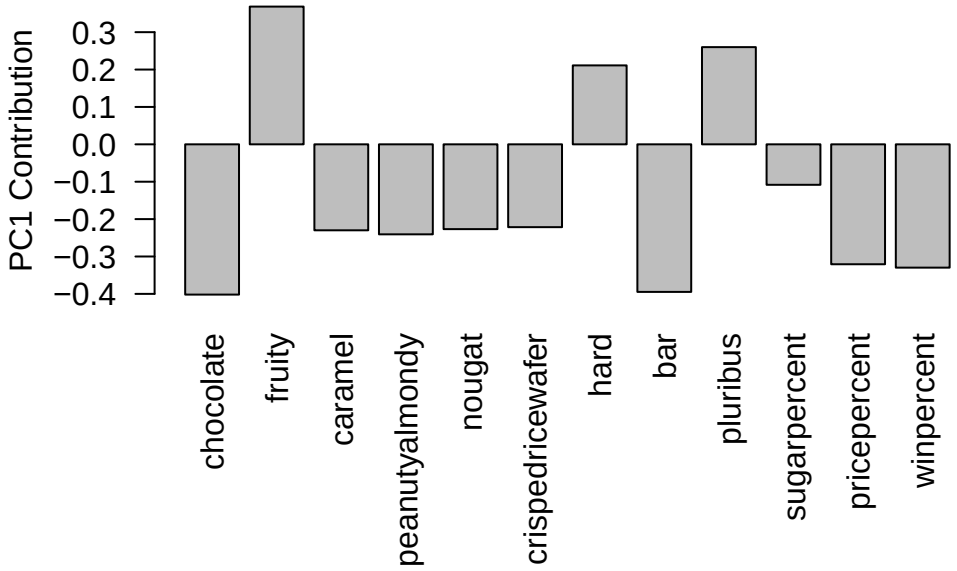



```
p + geom_text_repel(col=my_cols, max.overlaps = 8)  +
  theme(legend.position = "none") +
  labs(title="Halloween Candy PCA Space",
       subtitle="Colored by type: chocolate bar (red), chocolate other (yellow), fruity (b
       caption="Data from 538")
```

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

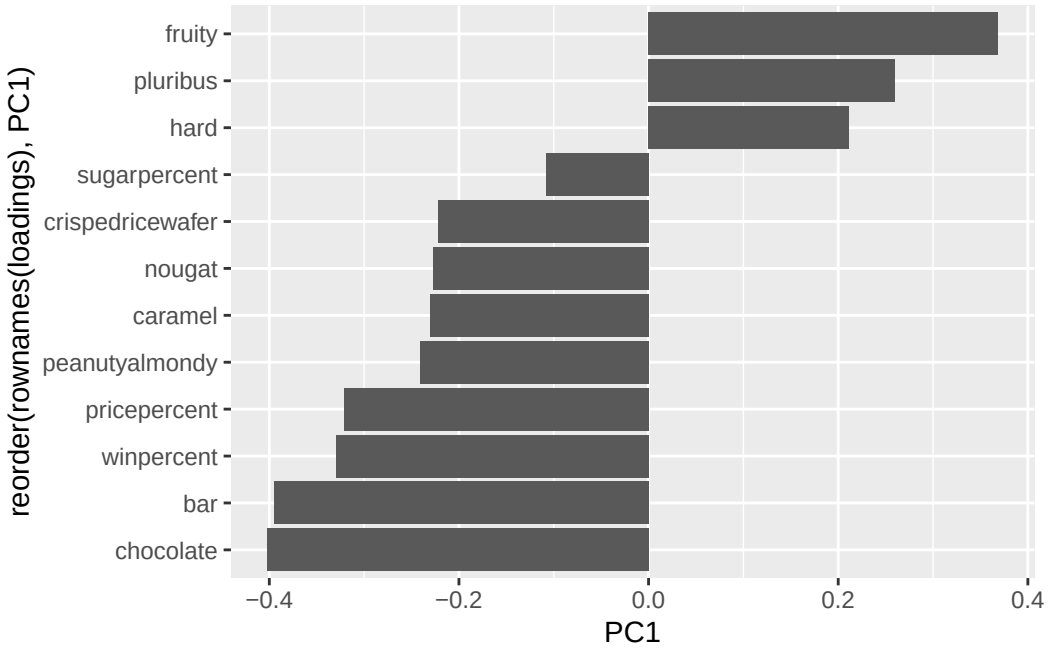



```
par(mar=c(8,4,2,2))
barplot(pca$rotation[,1], las=2, ylab="PC1 Contribution")
```



loadings plot

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



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

The three variables that are picked up by PC1 in the positive direction are fruity hard and pluribus. This makes sense as it lines up with the data (from the correlation matrix) we saw in addition to my own experiences.