class09

Tin

Importing Data

```
#read.csv("candy-data.csv")
candy_file <- "candy-data.csv"
```

```
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
```

What is your favorite candy?

Q3. What is your favorite candy in the data set and wat is it's winpercent 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
```

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

It looks like the last column 'candy$winpercent' is on a different scale to all others.

```
skimr::skim(candy)
```

Table 1: Data summary

Name	candy
Number of rows	85
Number of columns	12

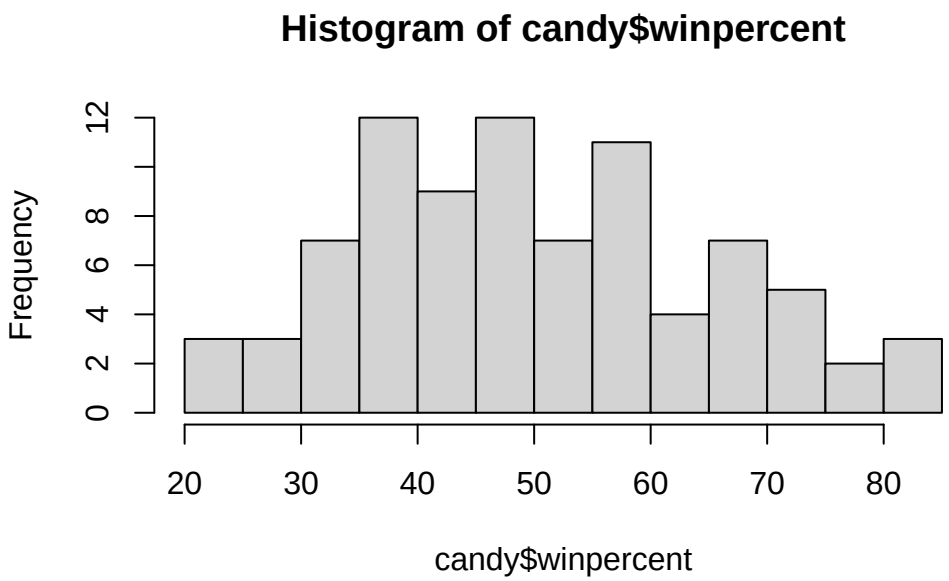
Column type frequency:	
numeric	12
Group variables	None

Variable type: numeric

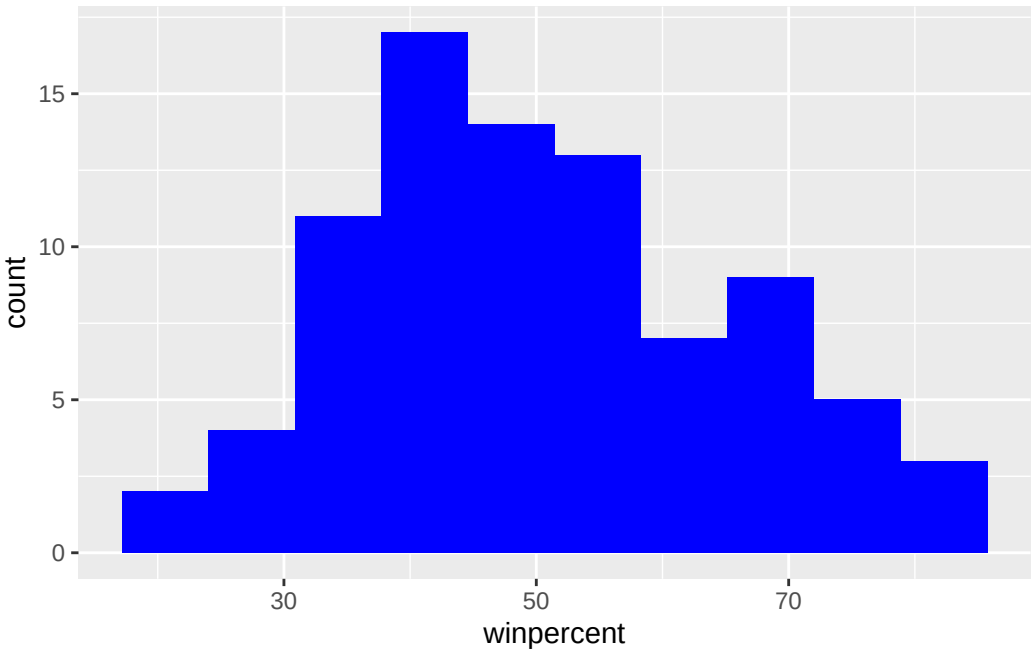
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
chocolate	0	1	0.44	0.50	0.00	0.00	0.00	1.00	1.00	
fruity	0	1	0.45	0.50	0.00	0.00	0.00	1.00	1.00	
caramel	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
peanutyalmondy	0	1	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
nougat	0	1	0.08	0.28	0.00	0.00	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	

Q8. Plot a histogram of winpercent values

```
hist(candy$winpercent, breaks = 10)
```



```
library(ggplot2)

ggplot(candy) +
  aes(winpercent) +
  geom_histogram(bins = 10, fill ="blue")
```



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

```
summary(candy$winpercent)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  22.45   39.14   47.83   50.32   59.86   84.18
```

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

```
choc.inds <- candy$chocolate == 1

choc.candy <- candy[ choc.inds, ]

choc.win <- choc.candy$winpercent

mean(choc.win)
```

```
[1] 60.92153
```

```
fruity.inds <- candy$fruit == 1

fruity.candy <- candy[ fruity.inds, ]

fruity.win <- fruity.candy$winpercent

mean(fruity.win)
```

```
[1] 44.11974
```

Q12. Is this difference statiscally significant?

```
ans <- t.test(choc.win,fruity.win)

ans
```

```
    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
```

Yes with a P-value of 2.8713778×10^{-8}.

There are two related functions that can help here, one is the classic `sort()` and `order()`

```
x <- c(5,10,1,4)

sort(x, decreasing = T)
```

```
[1] 10  5  4  1
```

```
order(x)
```

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

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

```
inds <- order(candy$winpercent, decreasing = TRUE)

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

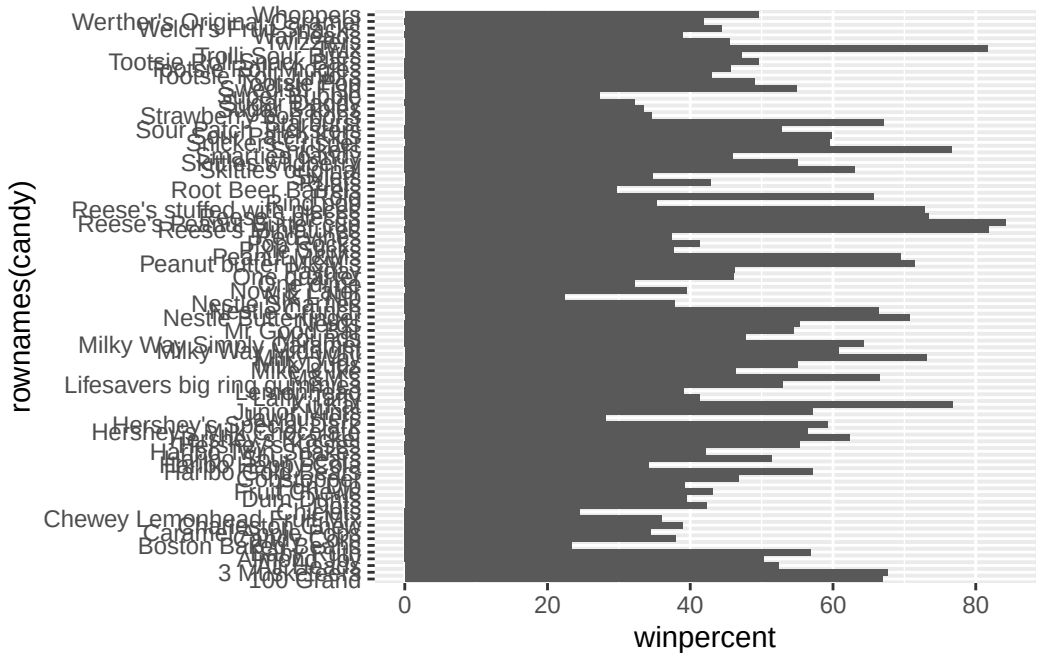
```
                  chocolate fruity caramel peanutyalmondy nougat
Jawbusters                0      1       0              0      0
Super Bubble              0      1       0              0      0
Chiclets                  0      1       0              0      0
Boston Baked Beans        0      0       0              1      0
Nik L Nip                 0      1       0              0      0
                  crispedricewafer hard bar pluribus sugarpercent pricepercent
Jawbusters                       0    1   0        1        0.093        0.511
Super Bubble                     0    0   0        0        0.162        0.116
Chiclets                         0    0   0        1        0.046        0.325
```

```
Boston Baked Beans                0    0    0        1        0.313        0.511
Nik L Nip                         0    0    0        1        0.197        0.976
                  winpercent
Jawbusters          28.12744
Super Bubble        27.30386
Chiclets            24.52499
Boston Baked Beans  23.41782
Nik L Nip           22.44534
```

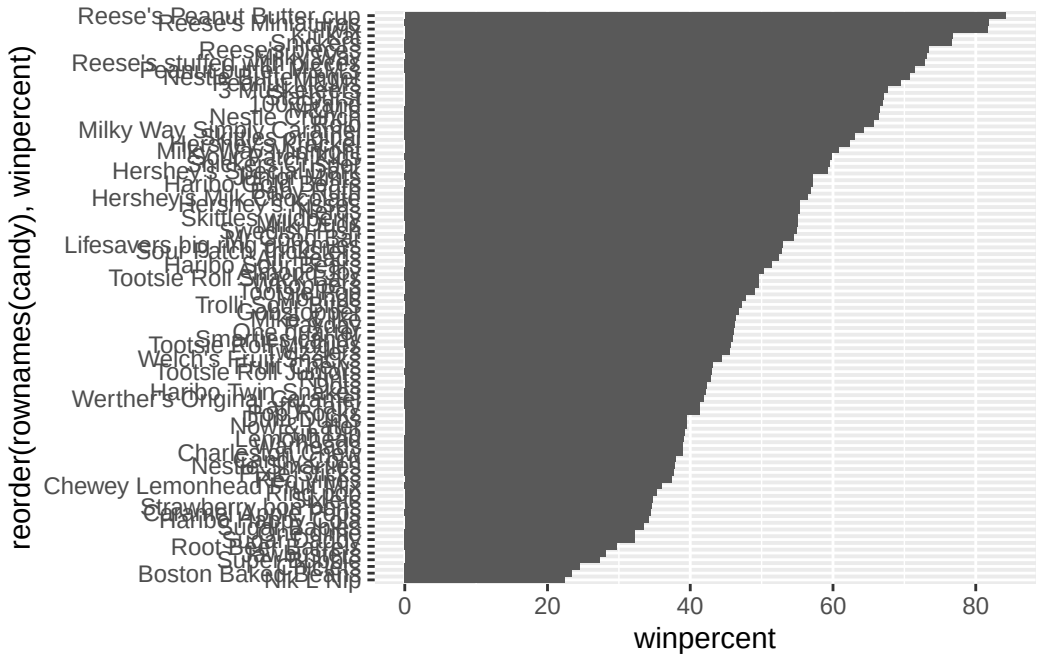
Q15.

Make a bar plot and order it by winpercent values

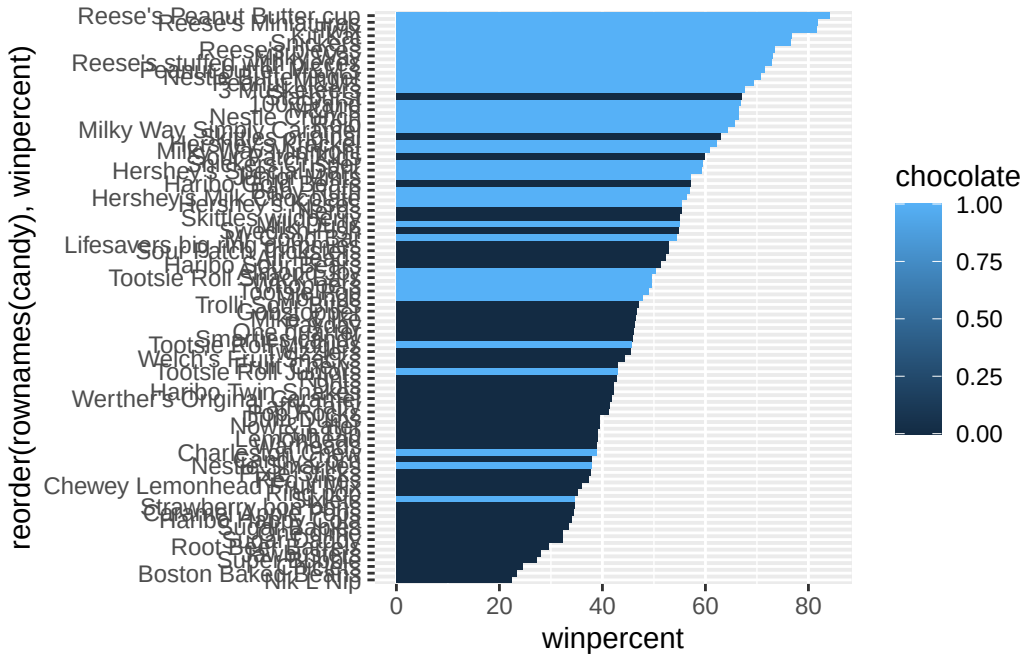
```
ggplot(candy) +
  aes(winpercent, rownames(candy)) +
  geom_col()
```




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

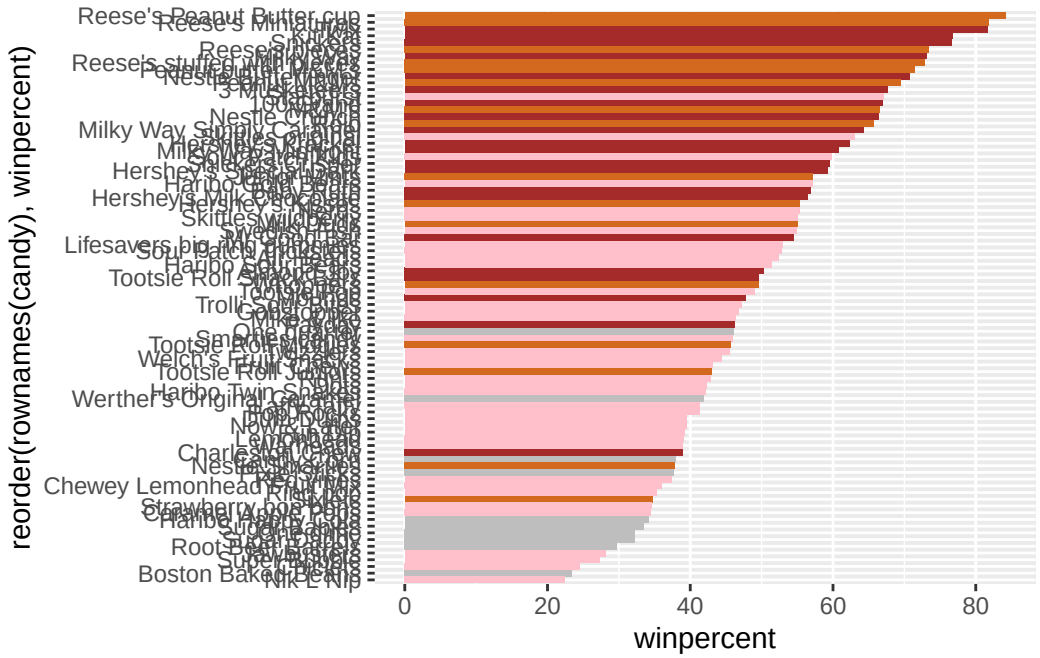



```
ggplot(candy) +
  aes( x=winpercent,
       y = reorder(rownames(candy),winpercent),
                   fill= chocolate)+
  geom_col()
```

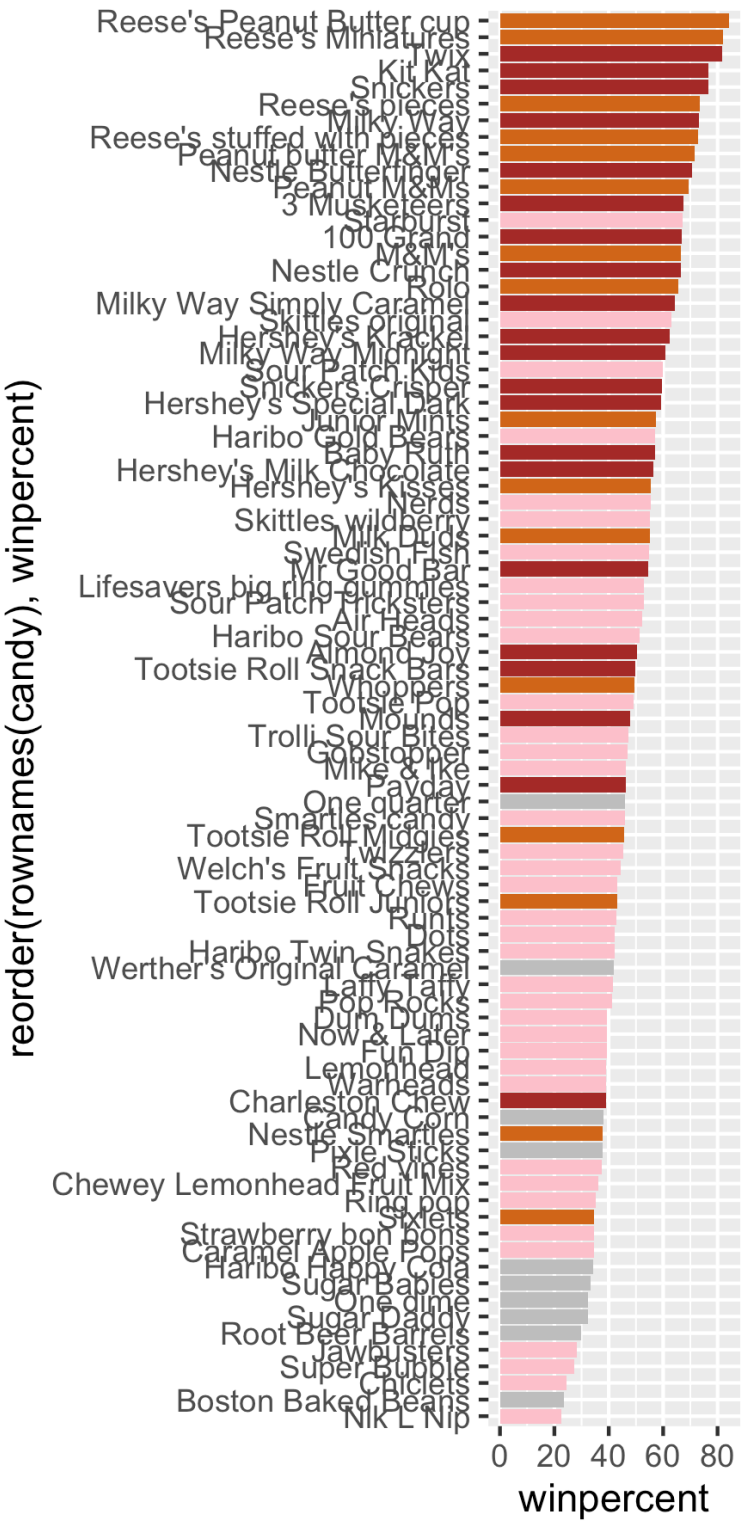


Here we want a custom color vector to color each bar the way we want - with chocoalte and fruity candy together with 'chocolate' and 'fruity' candy together with whter it is a 'bar' or not

```
mycols<- rep("gray",nrow(candy))
mycols[as.logical(candy$chocolate)] <- "chocolate"

mycols[as.logical(candy$fruity)] <- "pink"

mycols[as.logical(candy$bar)] <- "brown"



ggplot(candy) +
  aes( x=winpercent,
       y = reorder(rownames(candy),winpercent),
                   fill= chocolate)+
  geom_col(fill= mycols)
```




```
ggsave("mybarplot.png", width = 3, height = 6)
```



5. Winpercent vs priceper-

cent

```
library(corrplot)
```

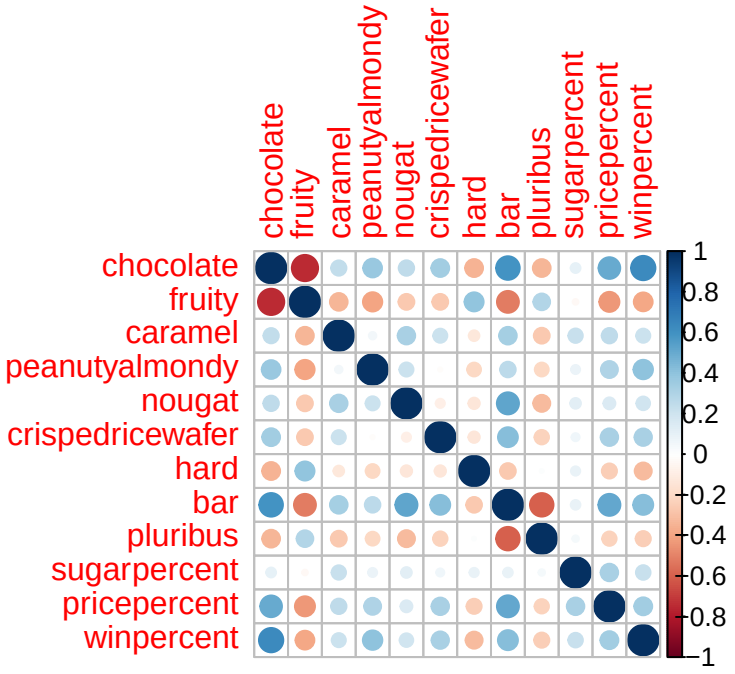
```
corrplot 0.95 loaded
```

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

```
                  chocolate      fruity     caramel peanutyalmondy      nougat
chocolate         1.0000000 -0.74172106  0.24987535     0.37782357  0.25489183
fruity           -0.7417211  1.00000000 -0.33548538    -0.39928014 -0.26936712
caramel           0.2498753 -0.33548538  1.00000000     0.05935614  0.32849280
peanutyalmondy    0.3778236 -0.39928014  0.05935614     1.00000000  0.21311310
nougat            0.2548918 -0.26936712  0.32849280     0.21311310  1.00000000
crispedricewafer  0.3412098 -0.26936712  0.21311310    -0.01764631 -0.08974359
hard             -0.3441769  0.39067750 -0.12235513    -0.20555661 -0.13867505
bar               0.5974211 -0.51506558  0.33396002     0.26041960  0.52297636
pluribus         -0.3396752  0.29972522 -0.26958501    -0.20610932 -0.31033884
sugarpercent      0.1041691 -0.03439296  0.22193335     0.08788927  0.12308135
pricepercent      0.5046754 -0.43096853  0.25432709     0.30915323  0.15319643
winpercent        0.6365167 -0.38093814  0.21341630     0.40619220  0.19937530
                  crispedricewafer        hard         bar    pluribus
chocolate              0.34120978 -0.34417691  0.59742114 -0.33967519
fruity                -0.26936712  0.39067750 -0.51506558  0.29972522
caramel                0.21311310 -0.12235513  0.33396002 -0.26958501
peanutyalmondy        -0.01764631 -0.20555661  0.26041960 -0.20610932
nougat                -0.08974359 -0.13867505  0.52297636 -0.31033884
crispedricewafer       1.00000000 -0.13867505  0.42375093 -0.22469338
hard                  -0.13867505  1.00000000 -0.26516504  0.01453172
bar                    0.42375093 -0.26516504  1.00000000 -0.59340892
pluribus              -0.22469338  0.01453172 -0.59340892  1.00000000
sugarpercent           0.06994969  0.09180975  0.09998516  0.04552282
pricepercent           0.32826539 -0.24436534  0.51840654 -0.22079363
winpercent             0.32467965 -0.31038158  0.42992933 -0.24744787
                  sugarpercent pricepercent winpercent
chocolate           0.10416906    0.5046754  0.6365167
fruity             -0.03439296   -0.4309685 -0.3809381
caramel             0.22193335    0.2543271  0.2134163
peanutyalmondy      0.08788927    0.3091532  0.4061922
nougat              0.12308135    0.1531964  0.1993753
```

```
crispedricewafer   0.06994969    0.3282654  0.3246797
hard               0.09180975   -0.2443653 -0.3103816
bar                0.09998516    0.5184065  0.4299293
pluribus           0.04552282   -0.2207936 -0.2474479
sugarpercent       1.00000000    0.3297064  0.2291507
pricepercent       0.32970639    1.0000000  0.3453254
winpercent         0.22915066    0.3453254  1.0000000
```

```
corrplot(cij)
```



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

Chocolate and fruity and negartiely correlated

```
round (cij["chocolate","fruity"],2)
```

```
[1] -0.74
```

Q23. Similarly, what two varaibles are most possitively correlated?

Principal Component Analysis (PCA)

We need to be sure to scale our input 'candy' data before PCA as we have the 'winpercent' column on a different scale to all others in the dataset.

```
pca <- prcomp(candy, scale = T)

summary(pca)
```

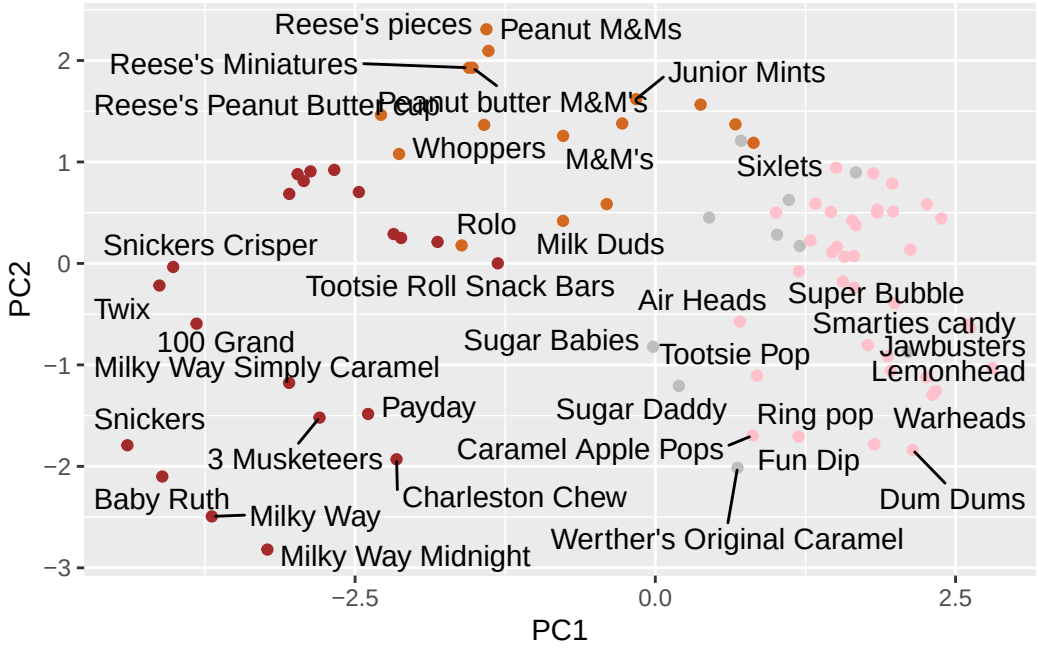
```
Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7
Standard deviation     2.0788 1.1378 1.1092 1.07533 0.9518 0.81923 0.81530
Proportion of Variance 0.3601 0.1079 0.1025 0.09636 0.0755 0.05593 0.05539
Cumulative Proportion  0.3601 0.4680 0.5705 0.66688 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
```

First Main result figure is my "PCA plot"

```
library(ggrepel)

ggplot(pca$x)+
  aes(PC1, PC2, label = rownames(pca$x)) +
  geom_point(col=mycols) +
  geom_text_repel(max.overlap = 1)
```

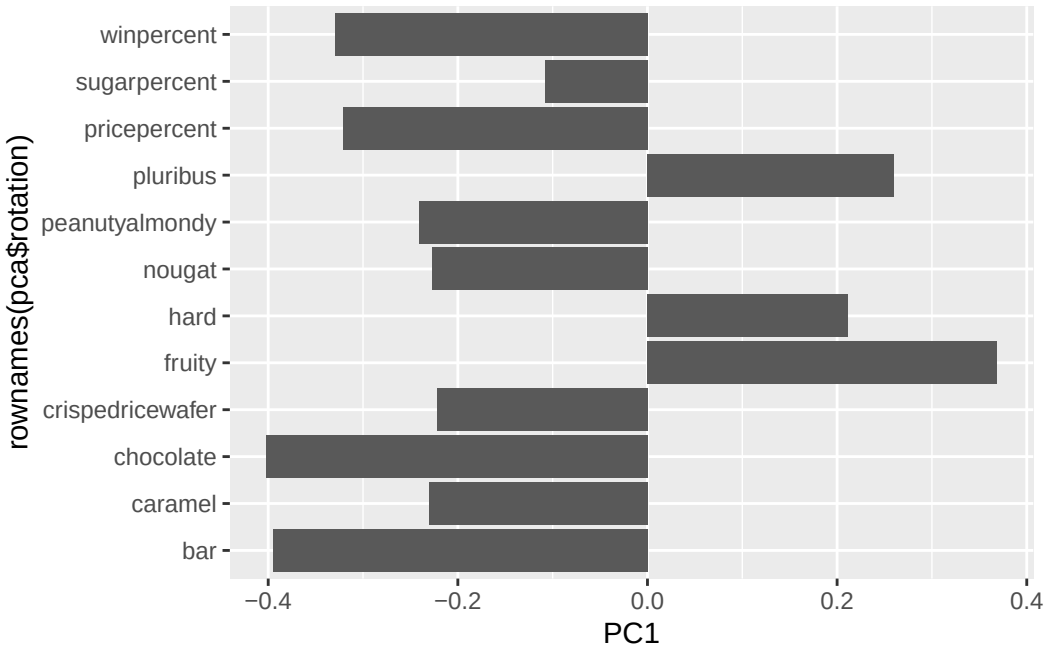
```
Warning in geom_text_repel(max.overlap = 1): Ignoring unknown parameters:
`max.overlap`
```

```
Warning: ggrepel: 48 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

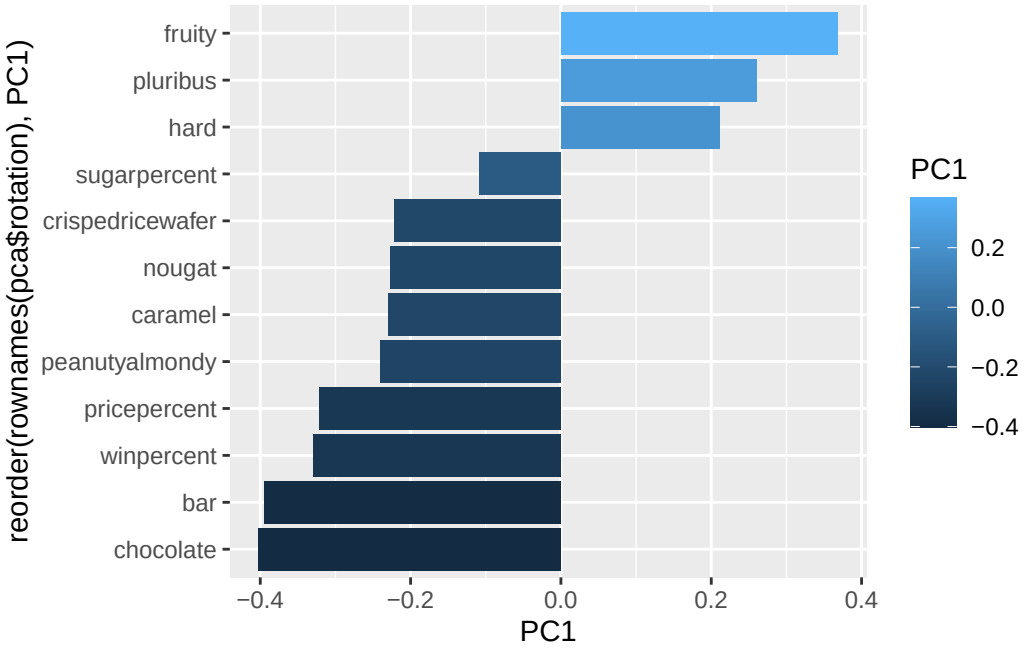


The second main PCA result is in the 'pca$rotation' we can plot this to generate a so-called "loadings" plot.

```
ggplot(pca$rotation)+
  aes(PC1, rownames(pca$rotation))+
  geom_col()
```




```
ggplot(pca$rotation)+
  aes(PC1, reorder(rownames(pca$rotation), PC1), fill=PC1)+
  geom_col()
```



Q24.