Class10_HalloweenMini-Project_Code

May 7, 2024

[1]:
```
# Counting the number of candy types and the number of fruity types
library(dplyr)
candy_file <- "candy-data.csv"

candy <- read.csv(candy_file, row.names = 1)
print(head(candy))
```

```
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


             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
```

[2]:

```
# Q1: How many different candy types are in this dataset?
num_candies <- nrow(candy)
print(num_candies)
```

```
[1] 85
```

[3]:

```
# Q2: How many fruity candy types are in the dataset?
num_fruity <- sum(candy$fruity)
print(num_fruity)
```

```
[1] 38
```

[4]:

```
# Q3, Q4, Q5: Finding winpercent values for specific candies
winpercent_100_grand <- candy["100 Grand", "winpercent"]
winpercent_kit_kat <- candy["Kit Kat", "winpercent"]
winpercent_tootsie_roll <- candy["Tootsie Roll Snack Bars", "winpercent"]

print(winpercent_100_grand)
print(winpercent_kit_kat)
print(winpercent_tootsie_roll)
```

```
[1] 66.97173
[1] 76.7686
[1] 49.6535
```

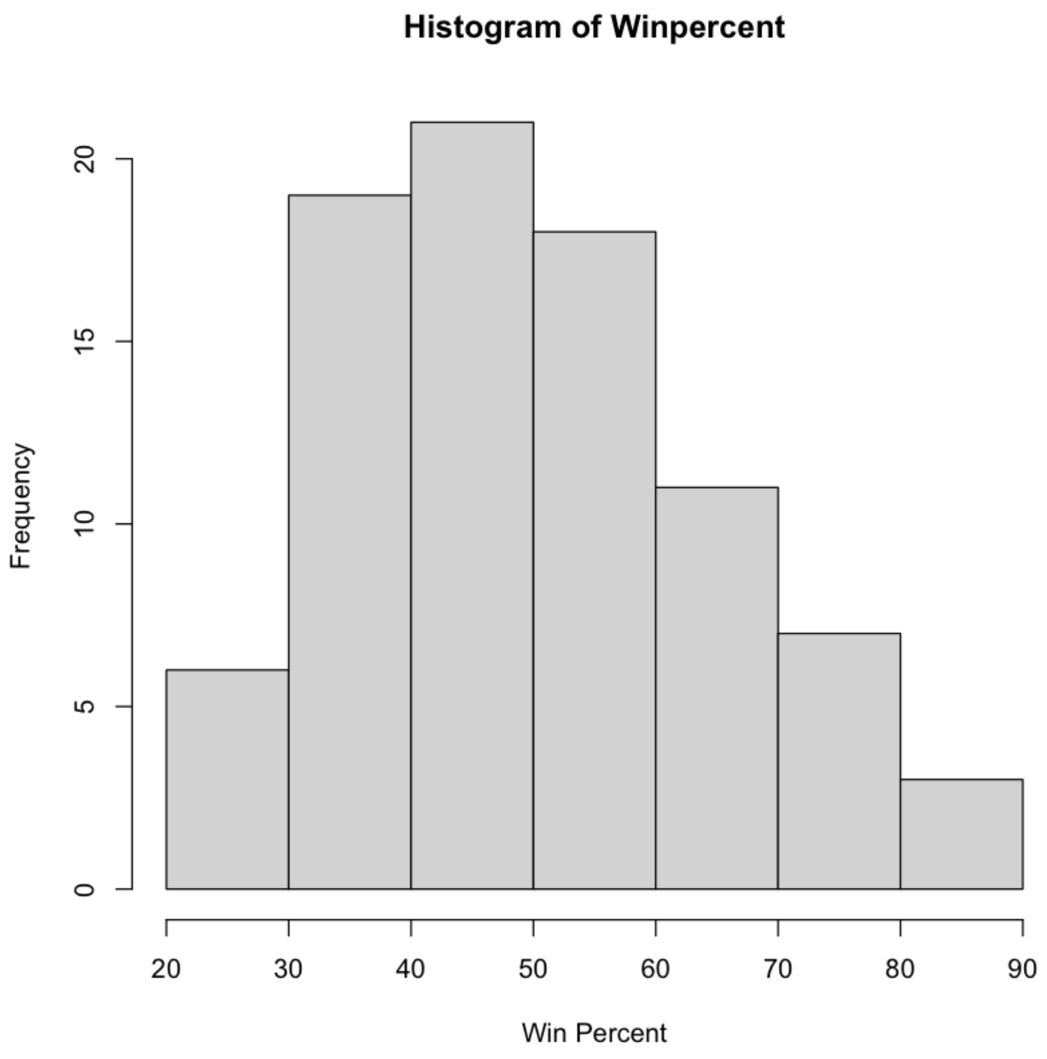
[5]:

```
# Q6 and Q7: Using skimr to inspect the scale of variables
library(skimr)
skim(candy)
```

```
Error in library(skimr): there is no package called ‘skimr’
Traceback:

1. library(skimr)
```

[6]:

```
# Q8: Plotting a histogram of winpercent values
hist(candy$winpercent, main="Histogram of Winpercent", xlab="Win Percent")
```



[7]:

```
# Q9 and Q10: Analyzing the histogram
# Answers would be based on the shape and distribution of the histogram
```

[8]:

```
# Q11 and Q12: Comparing chocolate vs fruity candies
avg_chocolate <- mean(candy$winpercent[candy$chocolate == 1])
avg_fruity <- mean(candy$winpercent[candy$fruity == 1])
print(avg_chocolate)
print(avg_fruity)

t.test(candy$winpercent[candy$chocolate == 1], candy$winpercent[candy$fruity ==␣
 ↪1])
```

```
[1] 60.92153
```

```
[1] 44.11974


        Welch Two Sample t-test

data:  candy$winpercent[candy$chocolate == 1] and candy$winpercent[candy$fruity␣
 ↪== 1]
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 
```

[9]:

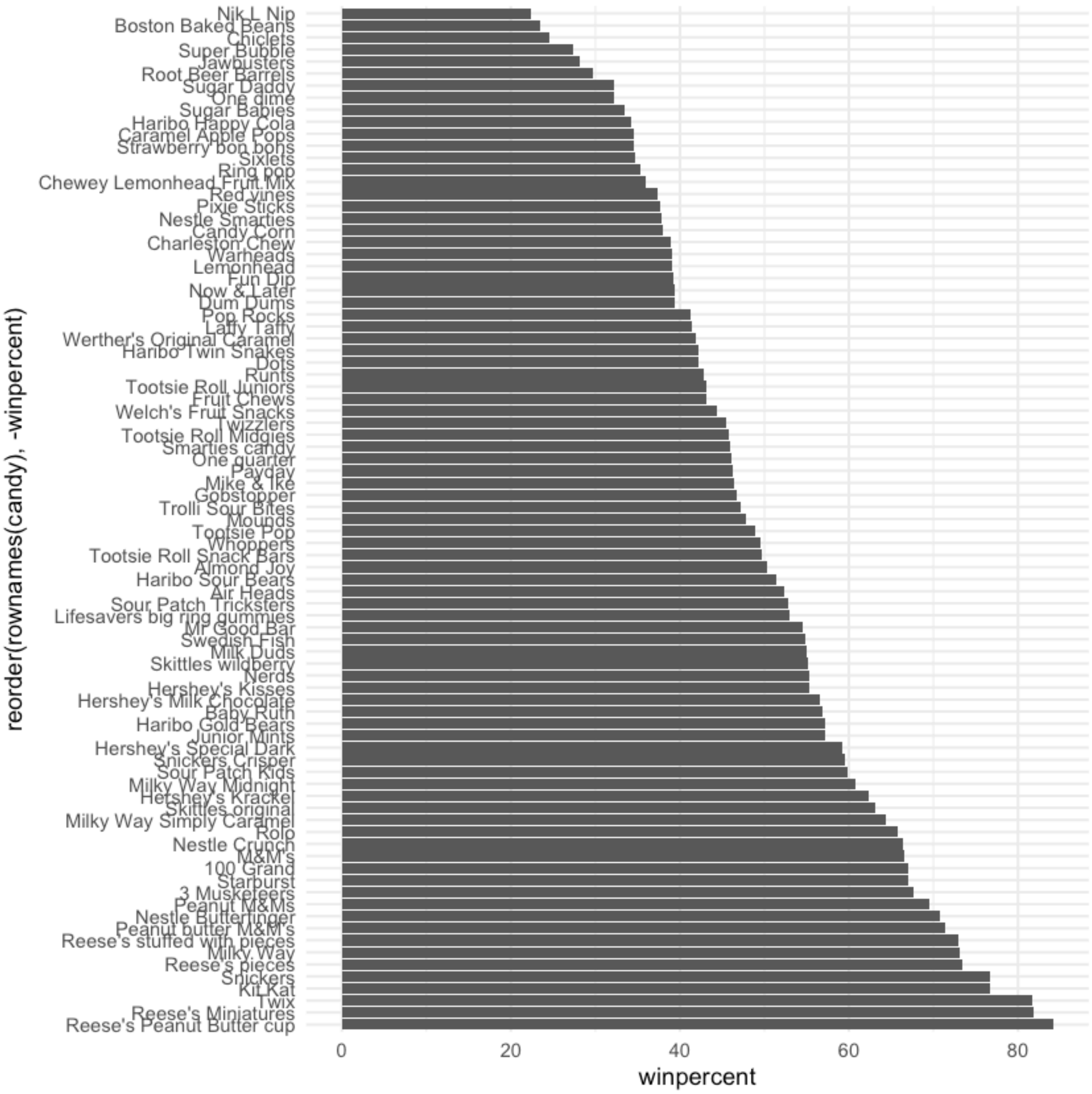
```
# Q13 and Q14: Finding the least and most liked candy types
least_liked <- head(candy[order(candy$winpercent), ], n = 5)
most_liked <- head(candy[order(candy$winpercent, decreasing = TRUE), ], n = 5)

print(least_liked)
print(most_liked)
```

```
                   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
                          chocolate fruity caramel peanutyalmondy nougat
Reese's Peanut Butter cup         1      0       0              1      0
Reese's Miniatures                1      0       0              1      0
Twix                              1      0       1              0      0
Kit Kat                           1      0       0              0      0
Snickers                          1      0       1              1      1
                          crispedricewafer hard bar pluribus sugarpercent
```

```
Reese's Peanut Butter cup                0    0   0        0        0.720
Reese's Miniatures                       0    0   0        0        0.034
Twix                                     1    0   1        0        0.546
Kit Kat                                  1    0   1        0        0.313
Snickers                                 0    0   1        0        0.546
                          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
```

[10]:

```r
# Q15 and Q16: Creating a barplot of candy rankings
library(ggplot2)
ggplot(candy, aes(x = reorder(rownames(candy), -winpercent), y = winpercent)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal()
```



[11]:

```
# Q17 and Q18: Identifying specific candy rankings
worst_chocolate <- candy[candy$chocolate == 1, ][which.
 ↪min(candy[candy$chocolate == 1, "winpercent"]), ]
best_fruity <- candy[candy$fruity == 1, ][which.max(candy[candy$fruity == 1,␣
 ↪"winpercent"]), ]

print(worst_chocolate)
print(best_fruity)
```

```
        chocolate fruity caramel peanutyalmondy nougat crispedricewafer hard
Sixlets         1      0       0              0      0                0    0
        bar pluribus sugarpercent pricepercent winpercent
Sixlets   0        1         0.22        0.081     34.722
```

```
          chocolate fruity caramel peanutyalmondy nougat crispedricewafer hard
Starburst         0      1       0              0      0                0    0
          bar pluribus sugarpercent pricepercent winpercent
Starburst   0        1        0.151         0.22   67.03763
```

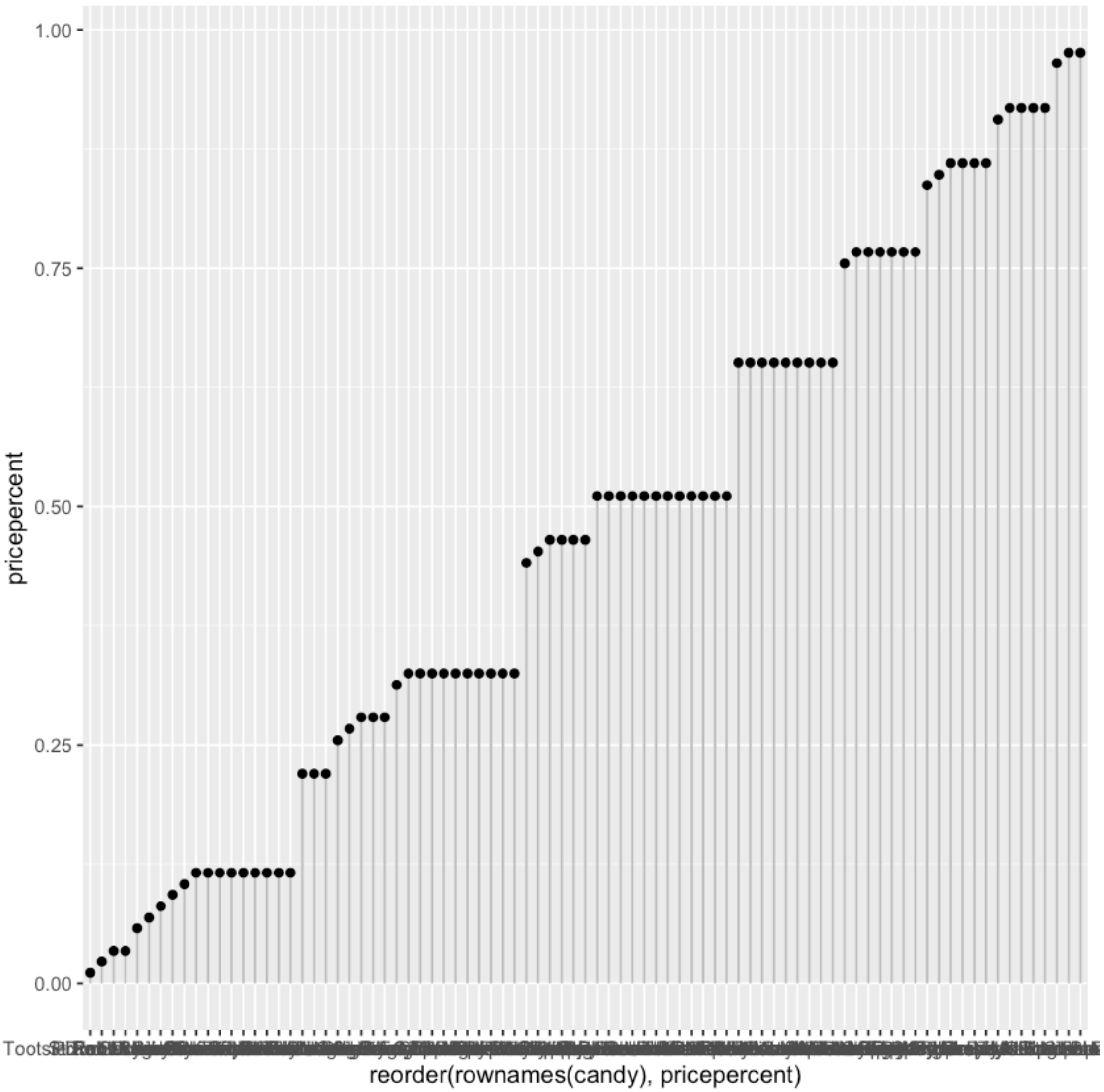
[12]:

```
# Q19 and Q20: Analyzing candy by price and winpercent
ggplot(candy, aes(x = pricepercent, y = winpercent, label = rownames(candy))) +
  geom_point() +
  geom_text_repel()

# Using order to find the most expensive and least popular
ord <- order(candy$pricepercent, decreasing = TRUE)
expensive_candies <- head(candy[ord, ], n = 5)
print(expensive_candies)
```

```
Error in geom_text_repel(): could not find function "geom_text_repel"
Traceback:
```

[13]:

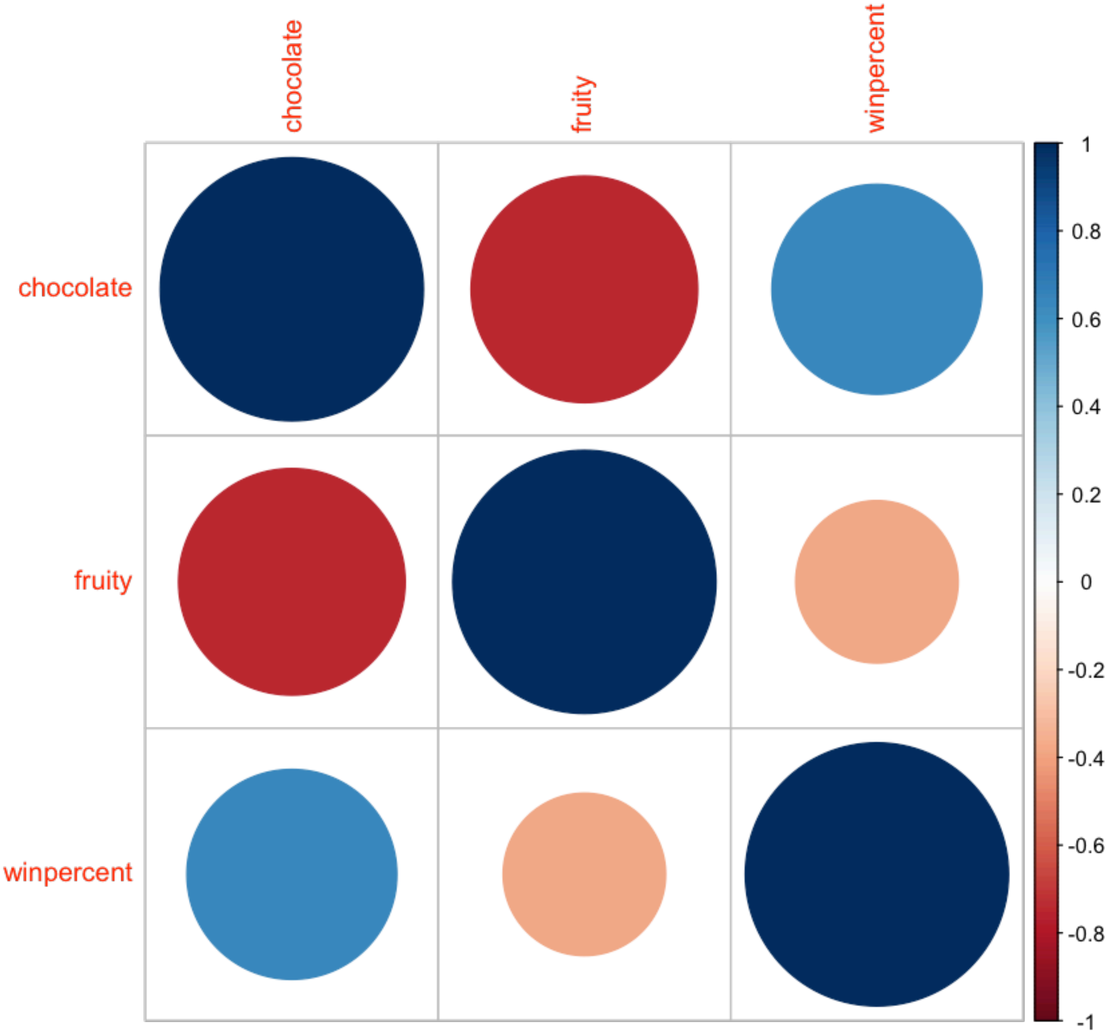
```
# Q21: Creating a lollipop chart
ggplot(candy, aes(x = reorder(rownames(candy), pricepercent), y =␣
 ↪pricepercent)) +
  geom_segment(aes(xend = rownames(candy), yend = 0), colour = "gray") +
  geom_point()
```



[14]:

```
# Q22 and Q23: Correlation analysis
library(corrplot)
cij <- cor(candy[, c("chocolate", "fruity", "winpercent")])
corrplot(cij)
```

```
corrplot 0.92 loaded
```



```
[15]: # Q24: Principal Component Analysis
pca <- prcomp(candy[, -1], scale. = TRUE)
summary(pca)
plot(pca$x[, 1:2], col = my_cols)

# Adding color and labels to PCA plot
ggplot(as.data.frame(pca$x[, 1:2]), aes(x = PC1, y = PC2, color = my_cols,␣
 ↪label = rownames(candy))) +
  geom_point() +
  geom_text_repel()
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
```

```
Standard deviation     1.9200 1.1143 1.1085 1.0751 0.95010 0.81815 0.81352
Proportion of Variance 0.3351 0.1129 0.1117 0.1051 0.08206 0.06085 0.06016
Cumulative Proportion  0.3351 0.4480 0.5597 0.6648 0.74685 0.80770 0.86787
                           PC8     PC9    PC10    PC11
Standard deviation     0.68950 0.64410 0.60875 0.43887
Proportion of Variance 0.04322 0.03772 0.03369 0.01751
Cumulative Proportion  0.91109 0.94880 0.98249 1.00000
```

```
Error in eval(expr, envir, enclos): object 'my_cols' not found
Traceback:

1. plot(pca$x[, 1:2], col = my_cols)
2. plot.default(pca$x[, 1:2], col = my_cols)
3. plot.xy(xy, type, …)
```