Randall_Plyler_CH3-4 EOC

Randall Plyler

1/22/2022

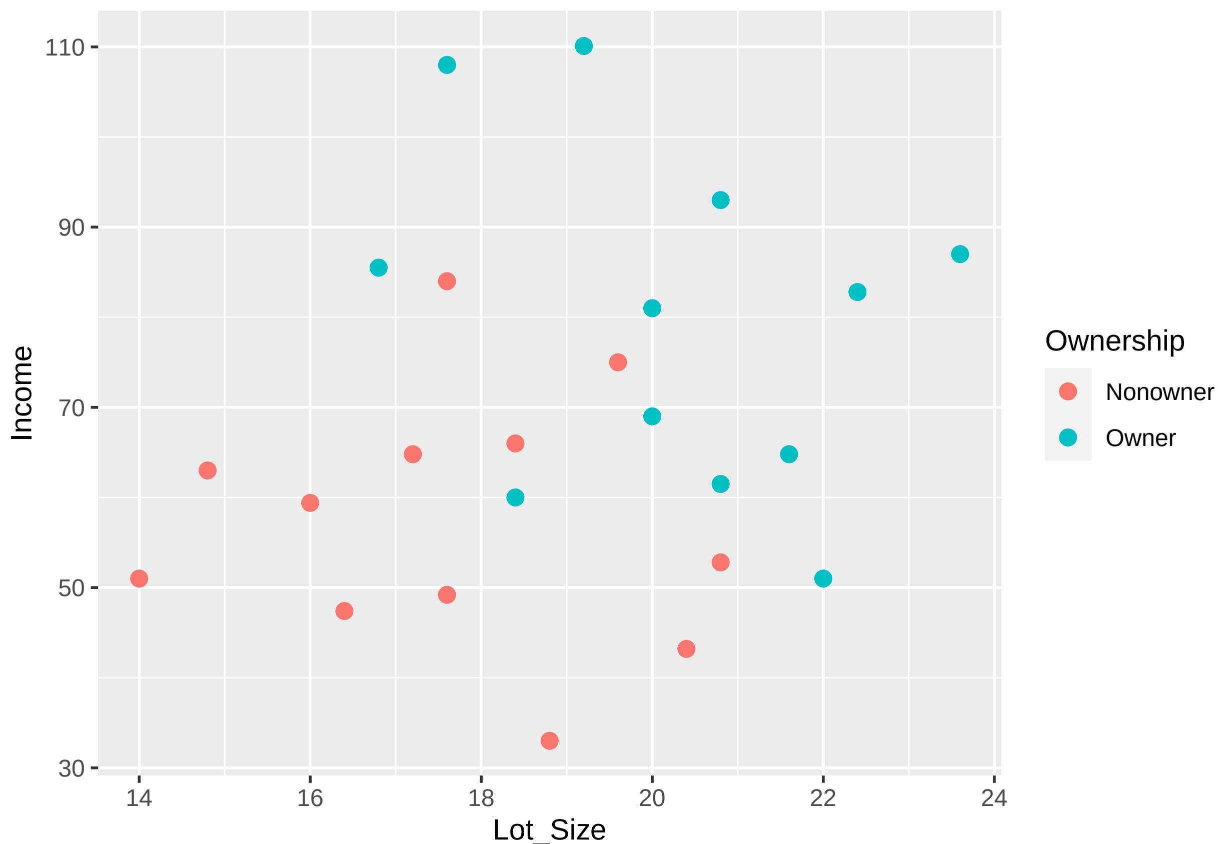
Sales of Riding Mowers: Scatter Plots. A company that manufactures riding mowers wants to identify the best sales prospects for an intensive sales campaign. In particular, the manufacturer is interested in classifying households as prospective owners or nonowners on the basis of Income (in $1000s) and Lot Size (in 1000 ft2).

The marketing expert looked at a random sample of 24 households, given in the file RidingMowers.csv. a. Using R, create a scatter plot of Lot Size vs. Income, color-coded by the outcome variable owner/nonowner. Make sure to obtain a well-formatted plot (create legible labels and a legend, etc.).

```
library(ggplot2)

df <- read.csv("C:/Users/randa/Dropbox/Masters/Winter/TBANLT 560 Data Mining/Files/DMBA-R-datasets/DMBA-
show(df)
```

```
##    Income Lot_Size Ownership
## 1    60.0     18.4     Owner
## 2    85.5     16.8     Owner
## 3    64.8     21.6     Owner
## 4    61.5     20.8     Owner
## 5    87.0     23.6     Owner
## 6   110.1     19.2     Owner
## 7   108.0     17.6     Owner
## 8    82.8     22.4     Owner
## 9    69.0     20.0     Owner
## 10   93.0     20.8     Owner
## 11   51.0     22.0     Owner
## 12   81.0     20.0     Owner
## 13   75.0     19.6  Nonowner
## 14   52.8     20.8  Nonowner
## 15   64.8     17.2  Nonowner
## 16   43.2     20.4  Nonowner
## 17   84.0     17.6  Nonowner
## 18   49.2     17.6  Nonowner
## 19   59.4     16.0  Nonowner
## 20   66.0     18.4  Nonowner
## 21   47.4     16.4  Nonowner
## 22   33.0     18.8  Nonowner
## 23   51.0     14.0  Nonowner
## 24   63.0     14.8  Nonowner
```

```r
ggplot(df, aes(x=Lot_Size, y=Income, colour=Ownership)) + geom_point(shape=19, size=2.5)
```



3 Laptop Sales at a London Computer Chain: Bar Charts and Boxplots. The file LaptopSalesJanuary2008.csv contains data for all sales of laptops at a computer chain in London in January 2008. This is a subset of the full dataset that includes data for the entire year.

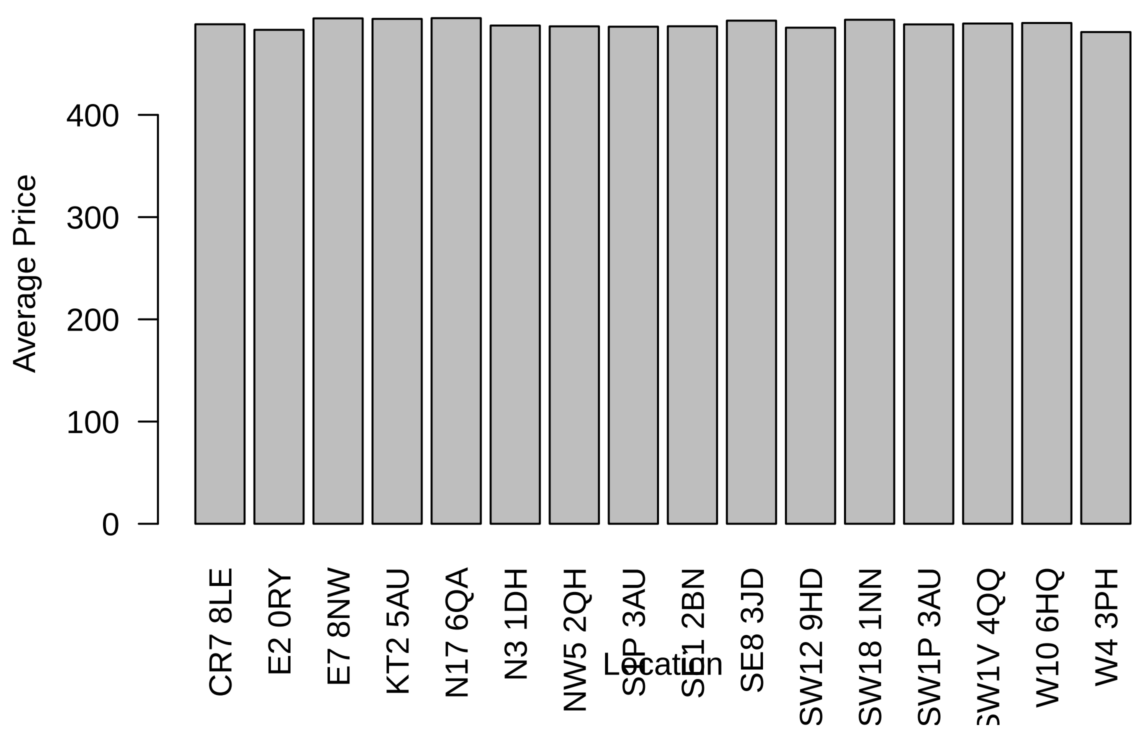
a. Create a bar chart, showing the average retail price by store. Which store has the highest average? Which has the lowest?

b. To better compare retail prices across stores, create side-by-side boxplots of retail price by store. Now compare the prices in the two stores from (a). Does there seem to be a difference between their price distributions?

```r
library(dplyr)
```

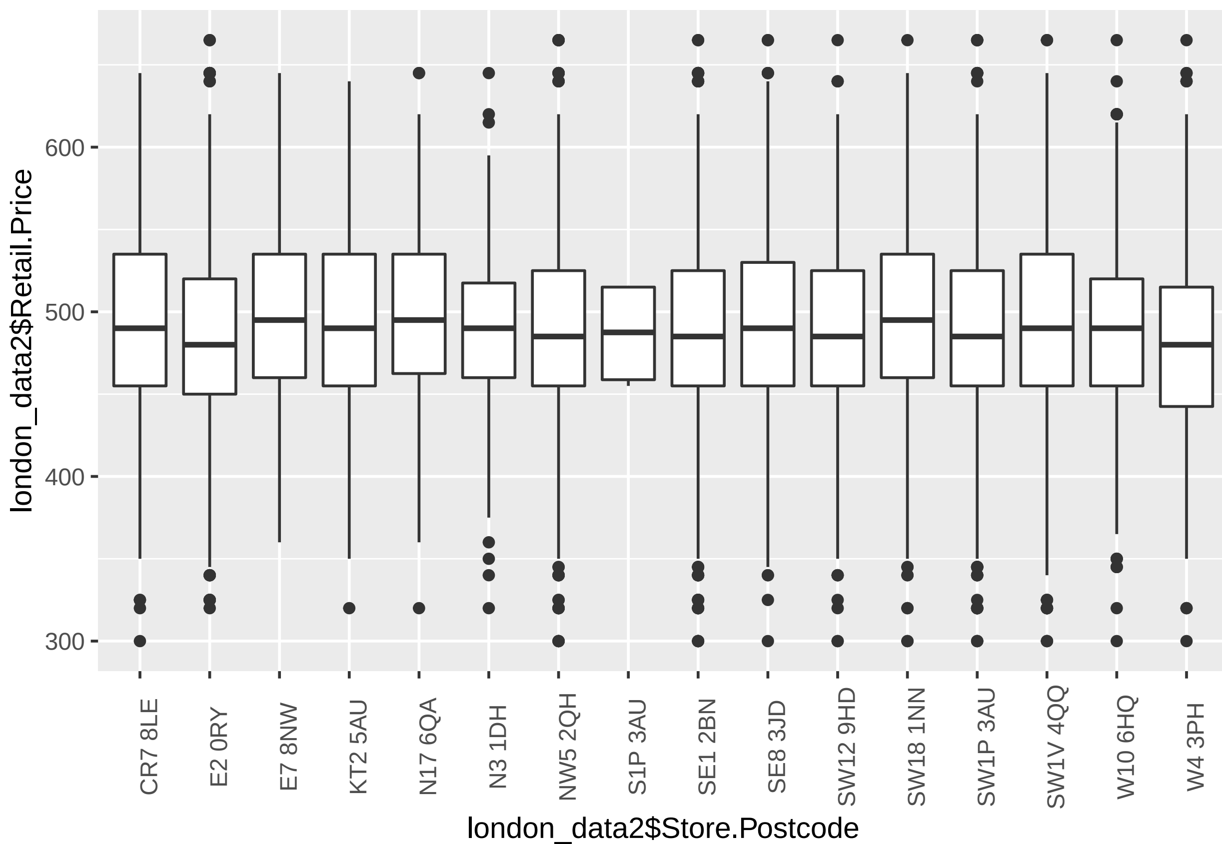
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)

london_data <- read.csv("C:/Users/randa/Dropbox/Masters/Winter/TBANLT 560 Data Mining/Files/DMBA-R-datas
london_data2 <- london_data
#Store.Postcode
#Retail.Price

#london_data <- select(london_data_original, c('Store.Postcode', 'Retail.Price'))
london_data <- aggregate(london_data$Retail.Price, by = list(london_data$Store.Postcode), FUN = mean)
barplot(london_data$x, names.arg =london_data$Group.1, xlab = "Location", ylab = "Average Price", las=2)
```




```r
ggplot(london_data2)+ geom_boxplot(aes(london_data2$Store.Postcode,london_data2$Retail.Price))+theme(ax
```

```
## Warning: Use of `london_data2$Store.Postcode` is discouraged. Use
## `Store.Postcode` instead.

## Warning: Use of `london_data2$Retail.Price` is discouraged. Use `Retail.Price`
## instead.
```



Breakfast Cereals. Use the data for the breakfast cereals example in Section 4.8 to explore and summarize the data as follows: a. Which variables are quantitative/numerical? Which are ordinal? Which are nominal? b. Compute the mean, median, min, max, and standard deviation for each of the quantitative variables. This can be done through R's sapply() function (e.g., sapply(data, mean, na.rm = TRUE)). c. Use R to plot a histogram for each of the quantitative variables. Based on the histograms and summary statistics, answer the following questions: i. Which variables have the largest variability? ii. Which variables seem skewed? iii. Are there any values that seem extreme? d. Use R to plot a side-by-side boxplot comparing the calories in hot vs. cold cereals. What does this plot show us? e. Use R to plot a side-by-side boxplot of consumer rating as a function of the shelf height. If we were to predict consumer rating from shelf height, does it appear that we need to keep all three categories of shelf height? f. Compute the correlation table for the quantitative variable (function cor()). In addition, generate a matrix plot for these variables (function plot(data)). i. Which pair of variables is most strongly correlated? ii. How can we reduce the number of variables based on these correlations? iii. How would the correlations change if we normalized the data first? g. Consider the first PC of the analysis of the 13 numerical variables in Table 4.11. Describe briefly what this PC represents.

```
cerealData <- read.csv("C:/Users/randa/Dropbox/Masters/Winter/TBANLT 560 Data Mining/Files/DMBA-R-datas
summary(cerealData) #calculate the summary statistics of the variables
```

```
##      name               mfr                type              calories    
##  Length:77          Length:77          Length:77          Min.   : 50.0  
##  Class :character   Class :character   Class :character   1st Qu.:100.0  
##  Mode  :character   Mode  :character   Mode  :character   Median :110.0  
```

```
##                                                              Mean   :106.9
##                                                              3rd Qu.:110.0
##                                                              Max.   :160.0
##
##     protein           fat            sodium          fiber
##  Min.   :1.000   Min.   :0.000   Min.   :  0.0   Min.   : 0.000
##  1st Qu.:2.000   1st Qu.:0.000   1st Qu.:130.0   1st Qu.: 1.000
##  Median :3.000   Median :1.000   Median :180.0   Median : 2.000
##  Mean   :2.545   Mean   :1.013   Mean   :159.7   Mean   : 2.152
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:210.0   3rd Qu.: 3.000
##  Max.   :6.000   Max.   :5.000   Max.   :320.0   Max.   :14.000
##
##      carbo          sugars           potass          vitamins
##  Min.   : 5.0   Min.   : 0.000   Min.   : 15.00   Min.   :  0.00
##  1st Qu.:12.0   1st Qu.: 3.000   1st Qu.: 42.50   1st Qu.: 25.00
##  Median :14.5   Median : 7.000   Median : 90.00   Median : 25.00
##  Mean   :14.8   Mean   : 7.026   Mean   : 98.67   Mean   : 28.25
##  3rd Qu.:17.0   3rd Qu.:11.000   3rd Qu.:120.00   3rd Qu.: 25.00
##  Max.   :23.0   Max.   :15.000   Max.   :330.00   Max.   :100.00
##  NA's   :1      NA's   :1        NA's   :2
##      shelf           weight          cups           rating
##  Min.   :1.000   Min.   :0.50   Min.   :0.250   Min.   :18.04
##  1st Qu.:1.000   1st Qu.:1.00   1st Qu.:0.670   1st Qu.:33.17
##  Median :2.000   Median :1.00   Median :0.750   Median :40.40
##  Mean   :2.208   Mean   :1.03   Mean   :0.821   Mean   :42.67
##  3rd Qu.:3.000   3rd Qu.:1.00   3rd Qu.:1.000   3rd Qu.:50.83
##  Max.   :3.000   Max.   :1.50   Max.   :1.500   Max.   :93.70
##
```

```
#name ->nominal
#mfr ->nominal
#type ->nominal
#calories ->numerical
#protein ->numerical
#fat ->numerical
#sodium ->numerical
#fiber ->numerical
#carbo ->numerical
#sugars ->numerical
#potass ->numerical
#vitamins ->numerical
#shelf ->ordinal
#weight ->numerical
#cups ->numerical
#rating ->ordinal

#This text is from the book and what I used to classify the variables
#Categorical variables can be either coded as numerical (1, 2, 3) or text

#(payments current, payments not current, bankrupt). Categorical variables can
#be unordered (called nominal variables) with categories such as North America,
#Europe, and Asia; or they can be ordered (called ordinal variables) with categories
#such as high value, low value, and nil value.
```

```r
standDev <- sapply(cerealData, sd)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
```

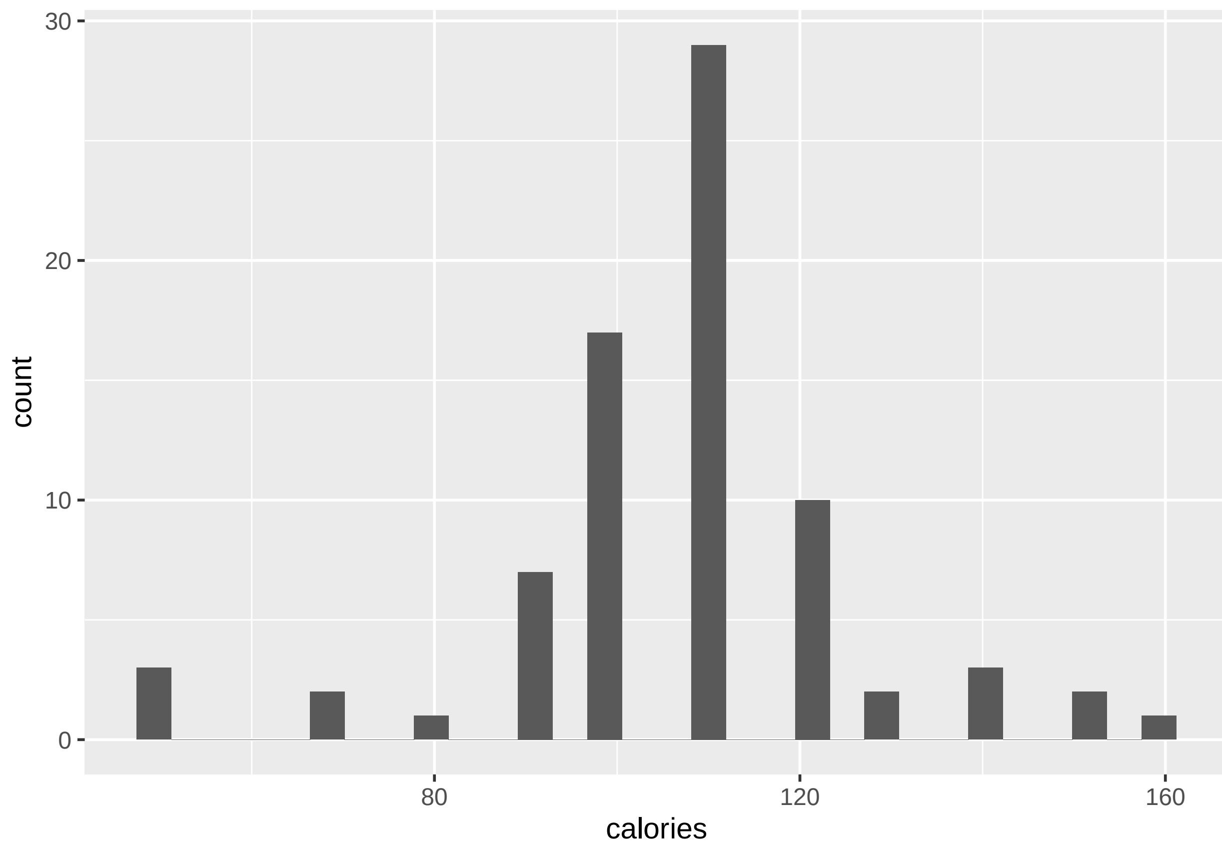
```r
standDev
```

```
##       name        mfr       type   calories    protein        fat     sodium
##         NA         NA         NA 19.4841191  1.0947897  1.0064726 83.8322952
##      fiber      carbo     sugars     potass   vitamins      shelf     weight
##  2.3833640         NA         NA         NA 22.3425225  0.8325241  0.1504768
##       cups     rating
##  0.2327161 14.0472887
```

```r
#This shows all of the standard deviation's of all the attributes that have them.
```

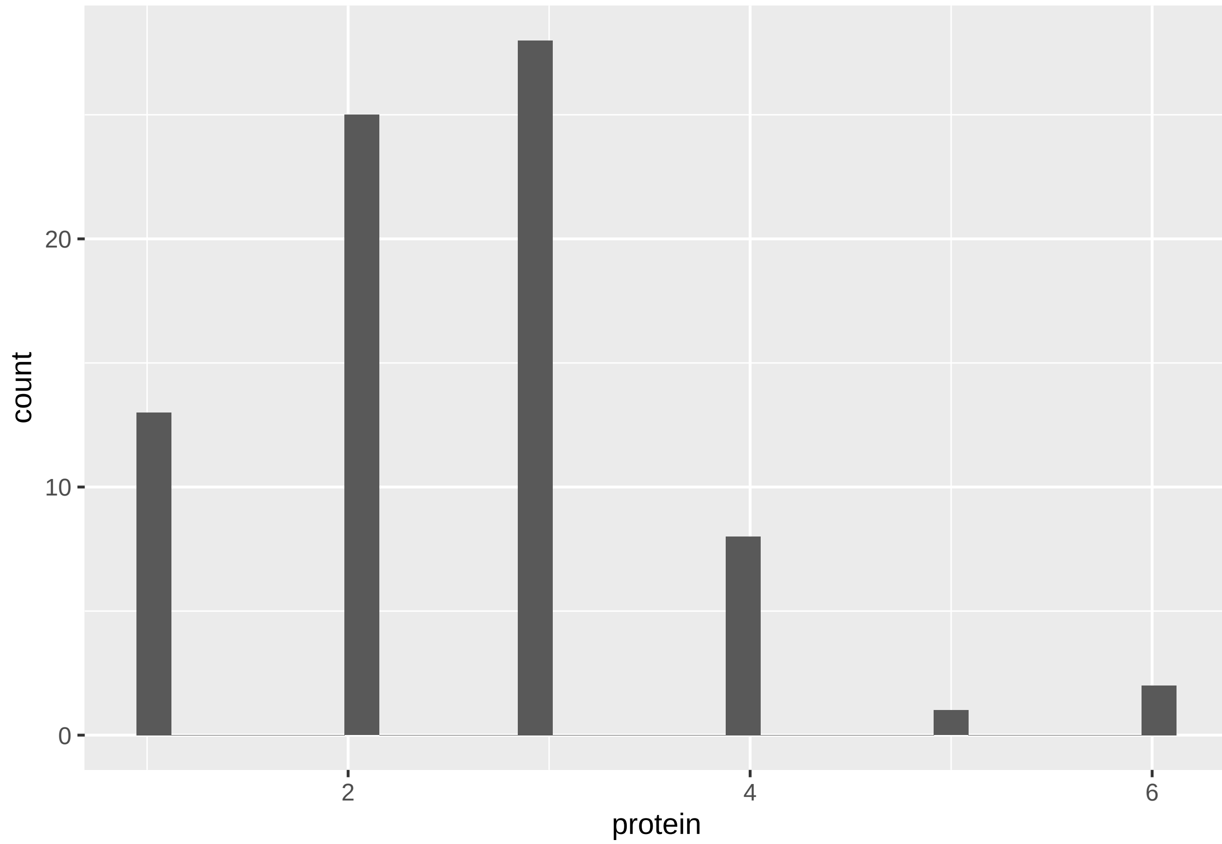
```r
library(ggplot2)
#name ->nominal
#mfr ->nominal
#type ->nominal
#calories ->numerical
#protein ->numerical
#fat ->numerical
#sodium ->numerical
#fiber ->numerical
#carbo ->numerical
#sugars ->numerical
#potass ->numerical
#vitamins ->numerical
#shelf ->ordinal
#weight ->numerical
#cups ->numerical
#rating ->ordinal
ggplot(cerealData, aes(x=calories)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



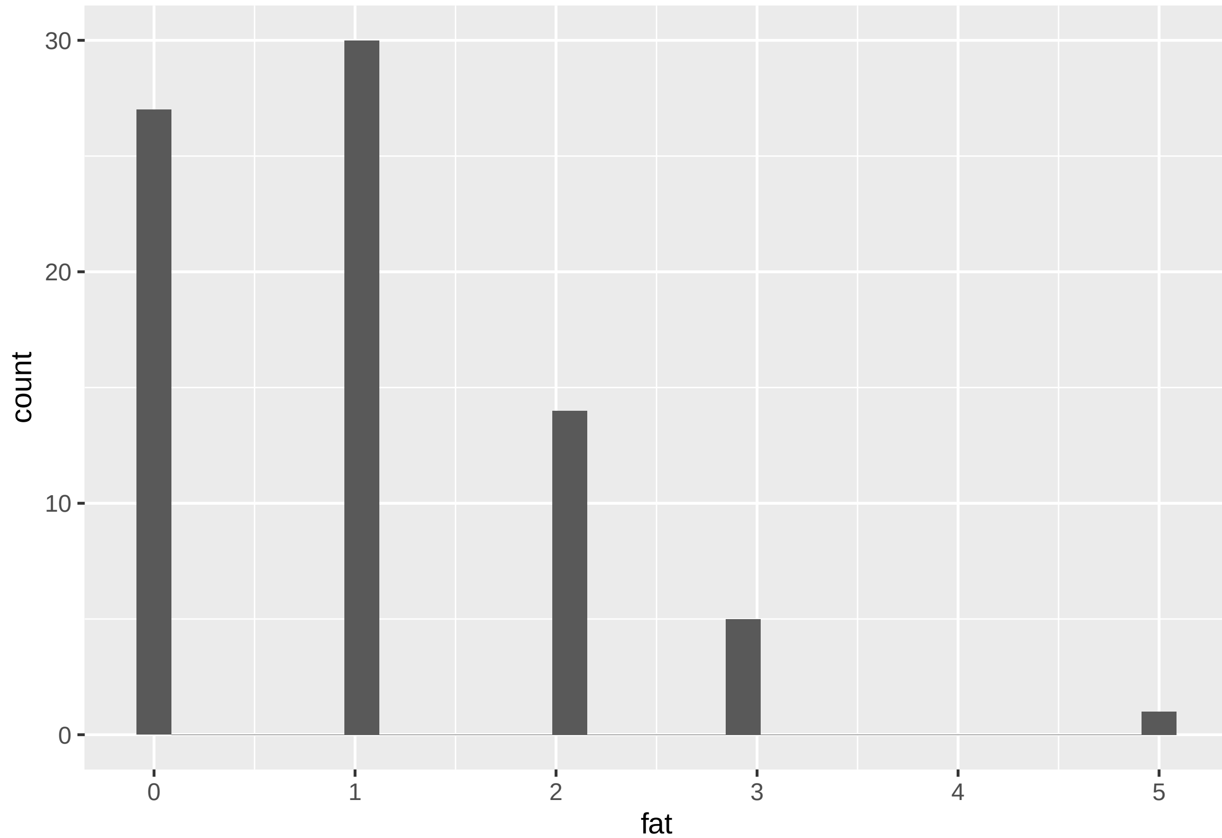

```
ggplot(cerealData, aes(x=protein)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



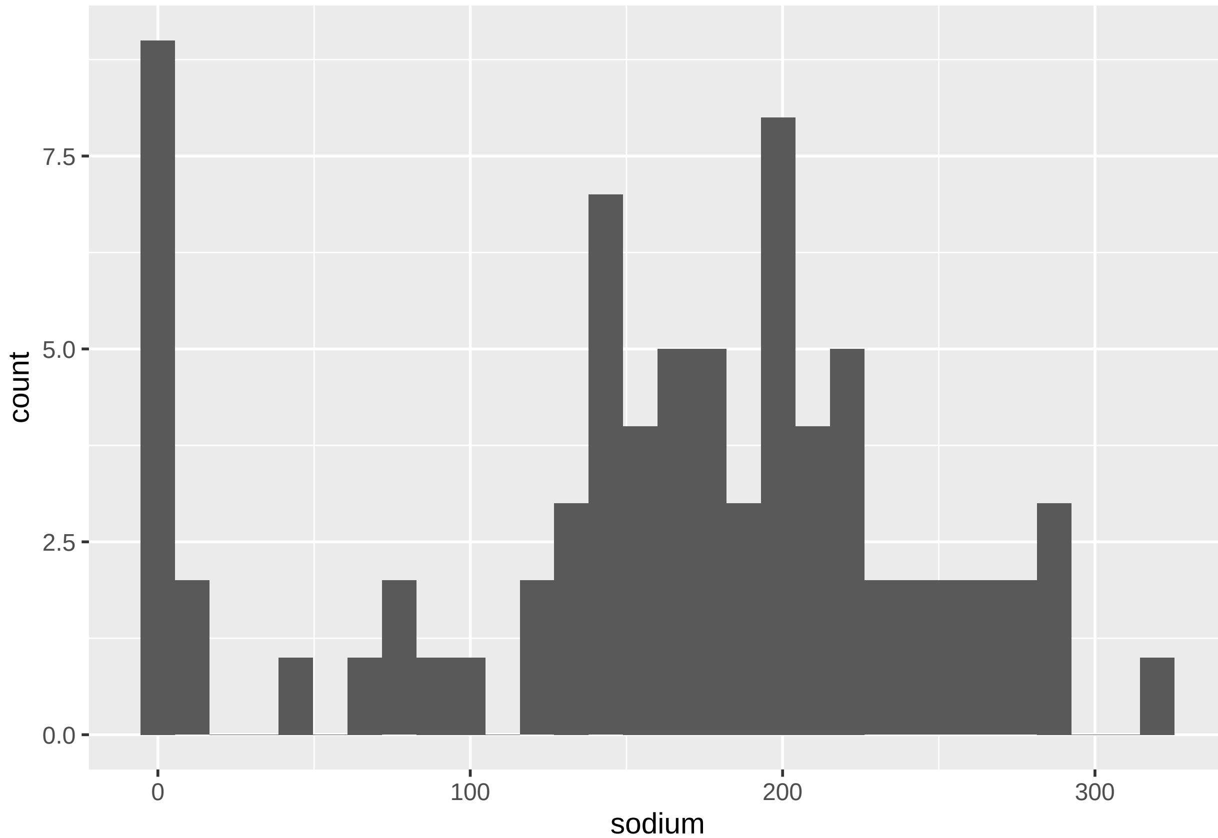

```
ggplot(cerealData, aes(x=fat)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



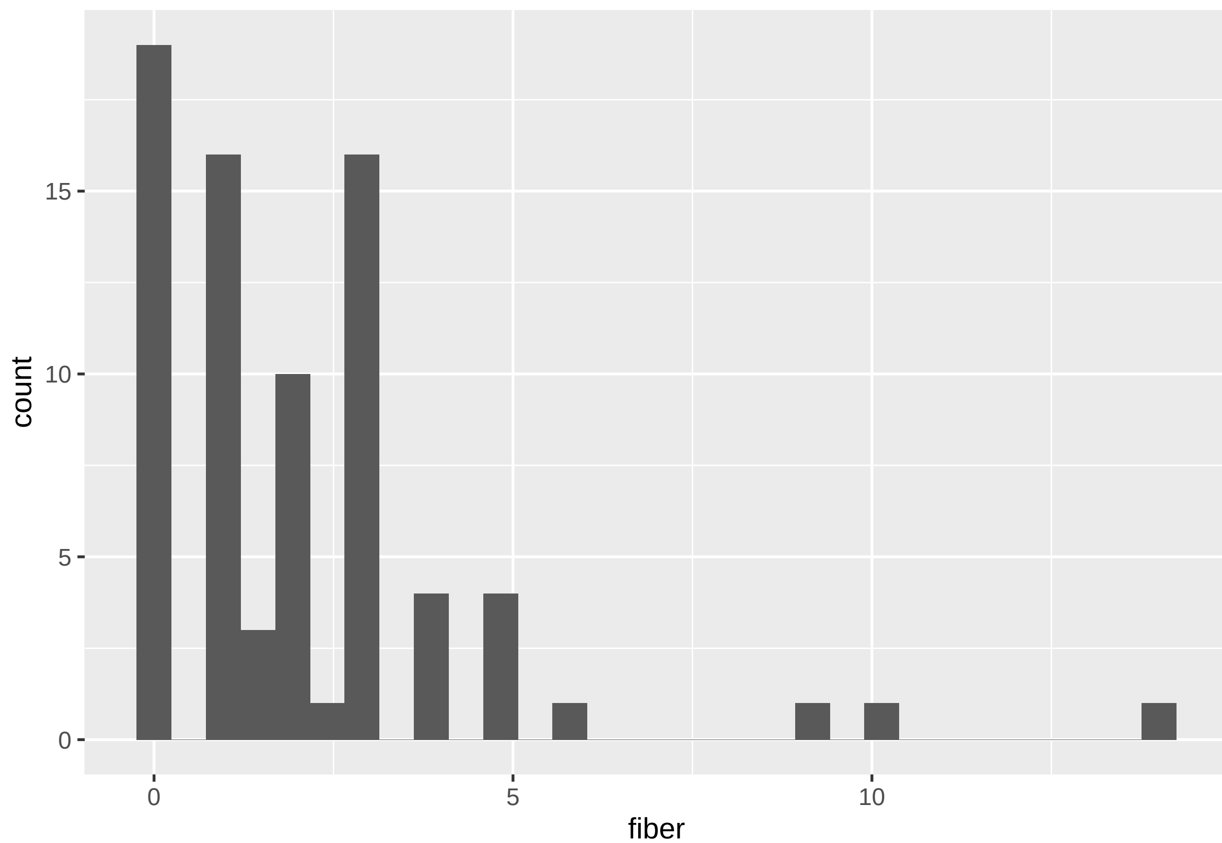

```
ggplot(cerealData, aes(x=sodium)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(cerealData, aes(x=fiber)) + geom_histogram()
```

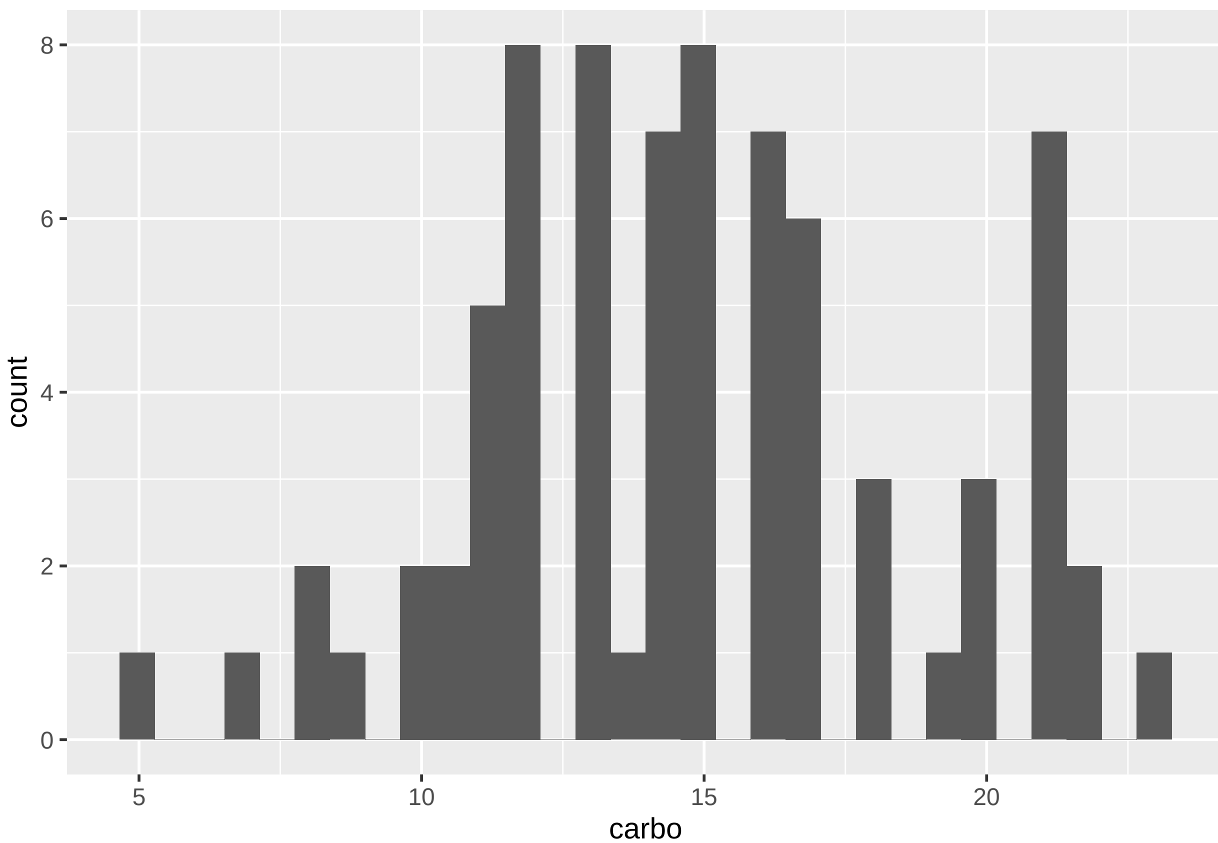
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(cerealData, aes(x=carbo)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

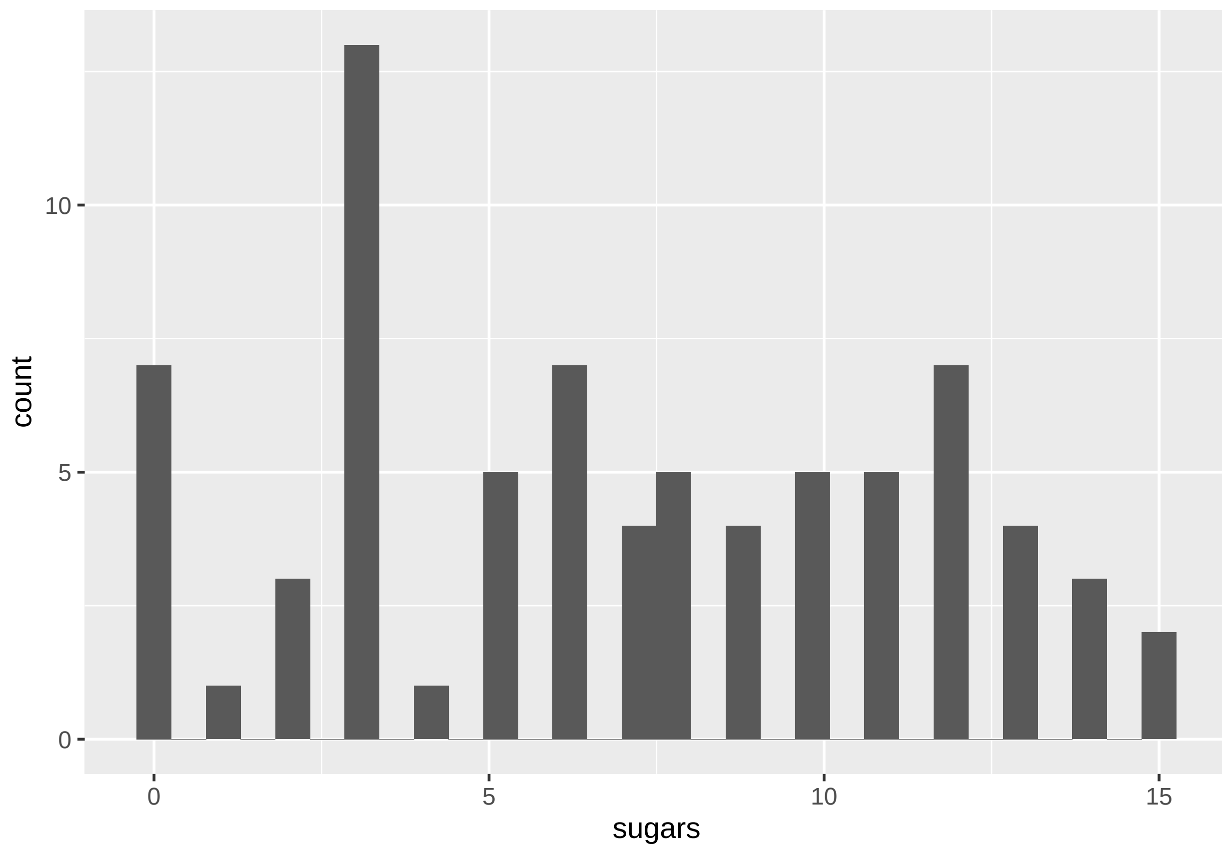
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
ggplot(cerealData, aes(x=sugars)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

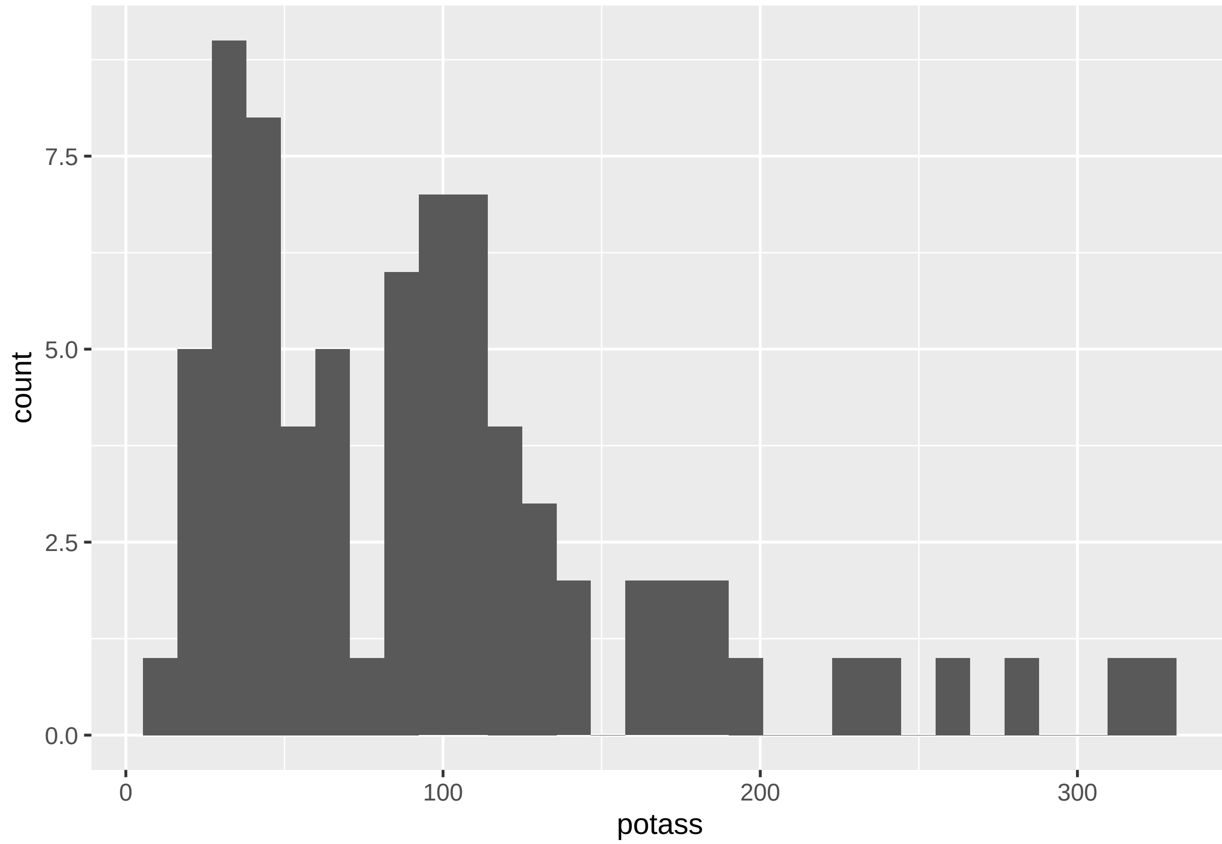
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
ggplot(cerealData, aes(x=potass)) + geom_histogram()
```

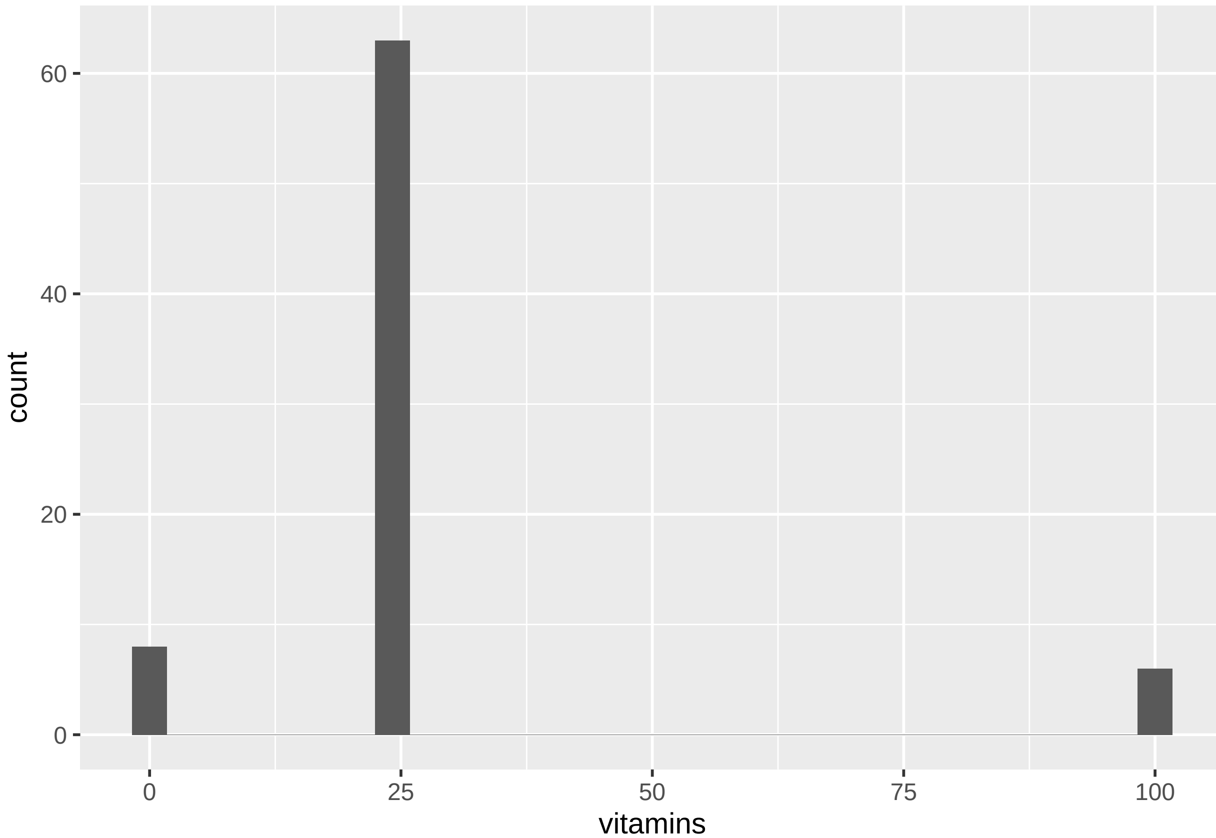
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```



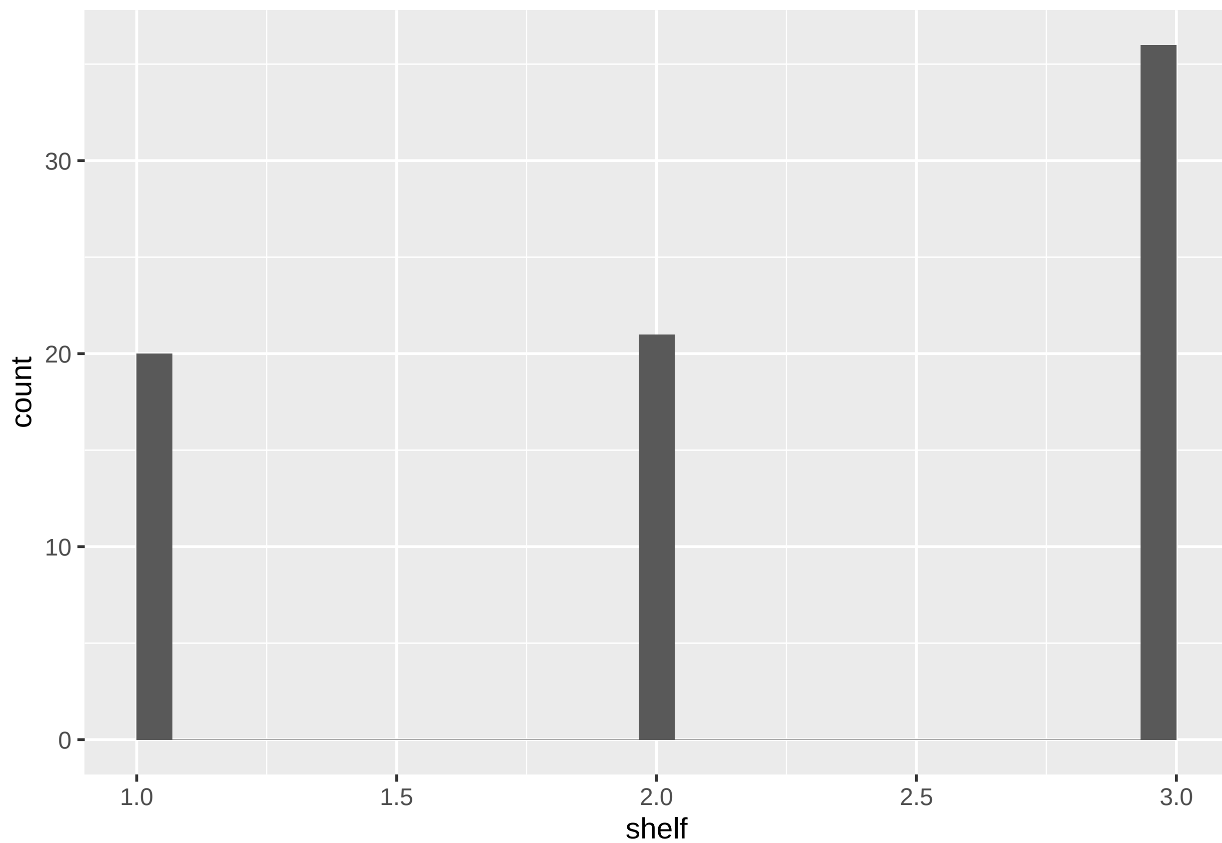

```
ggplot(cerealData, aes(x=vitamins)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



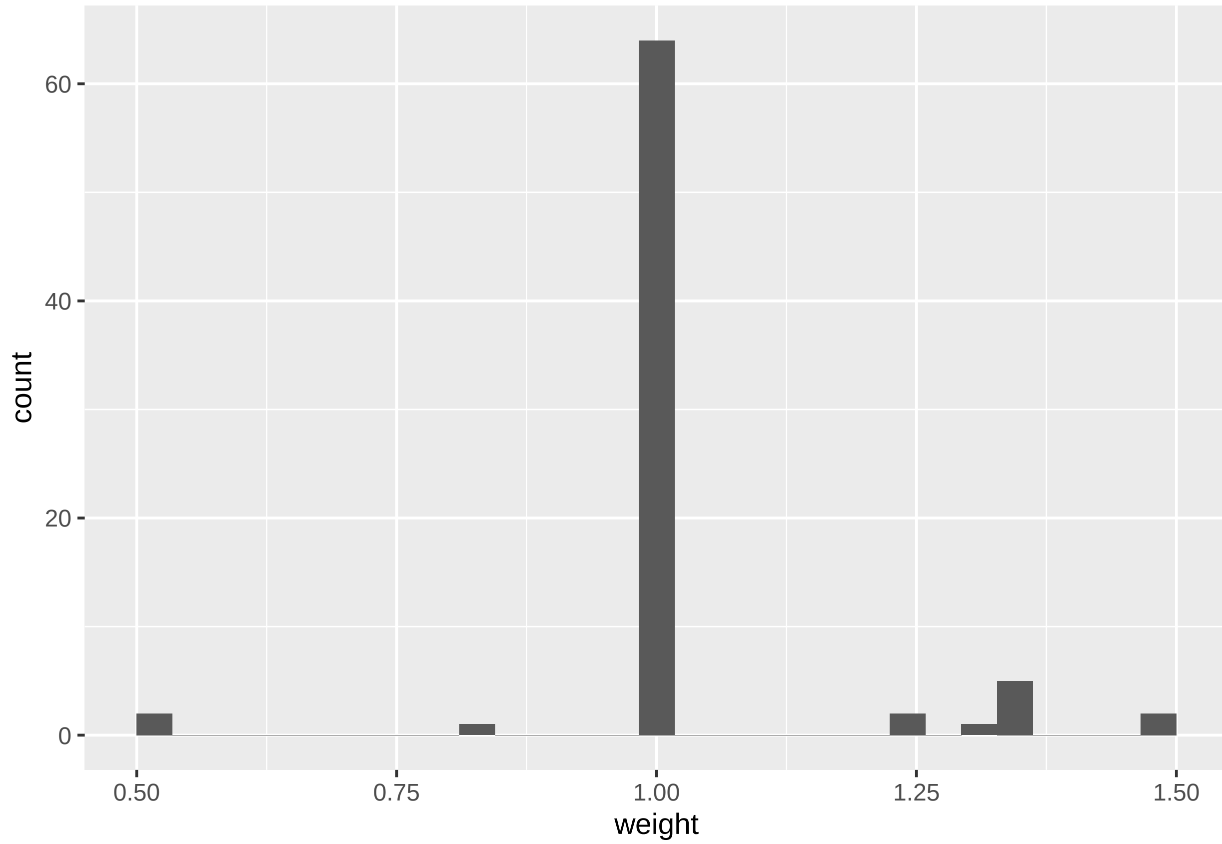

```
ggplot(cerealData, aes(x=shelf)) + geom_histogram()
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



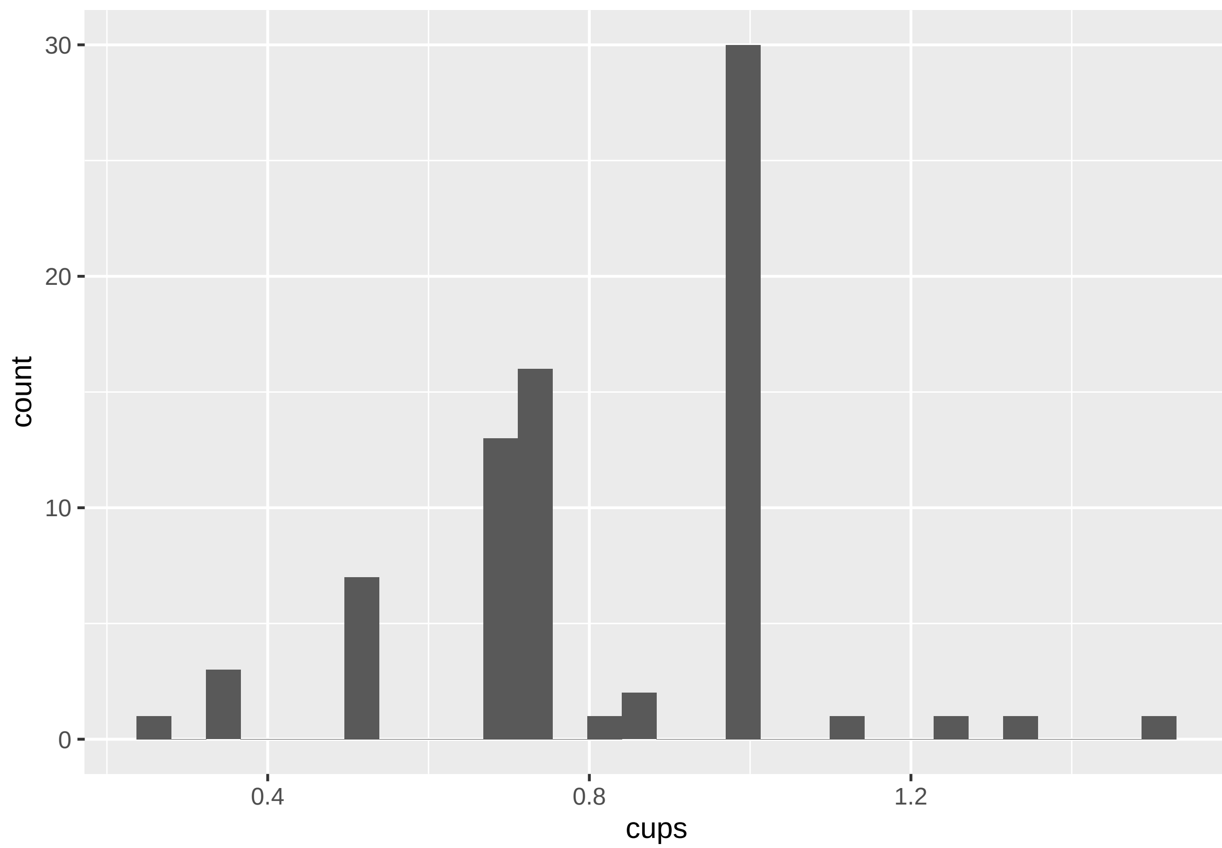

```
ggplot(cerealData, aes(x=weight)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



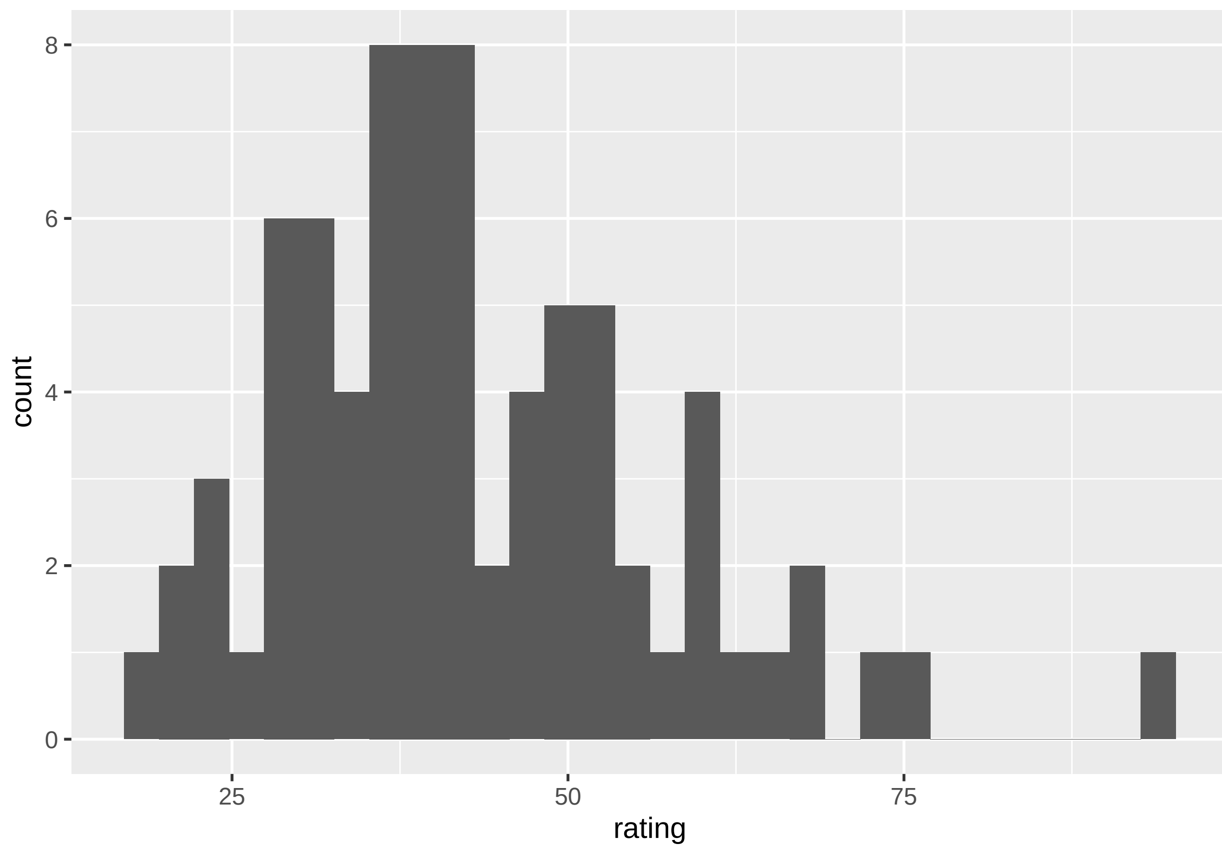

```
ggplot(cerealData, aes(x=cups)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

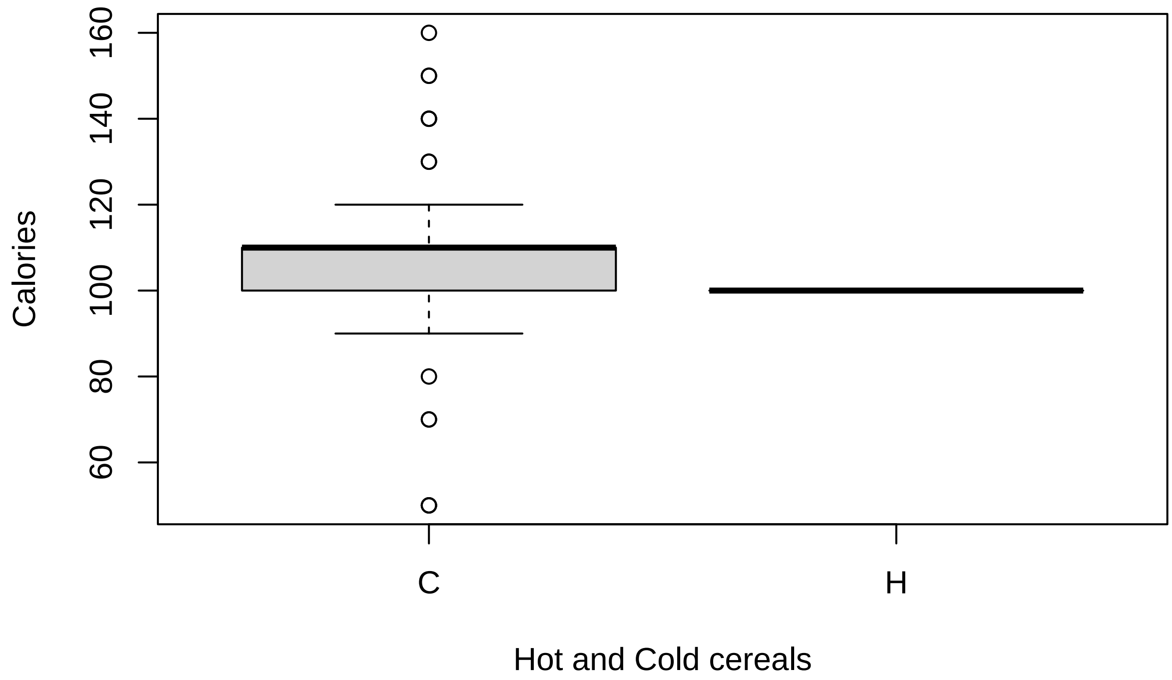



```
ggplot(cerealData, aes(x=rating)) + geom_histogram()
```

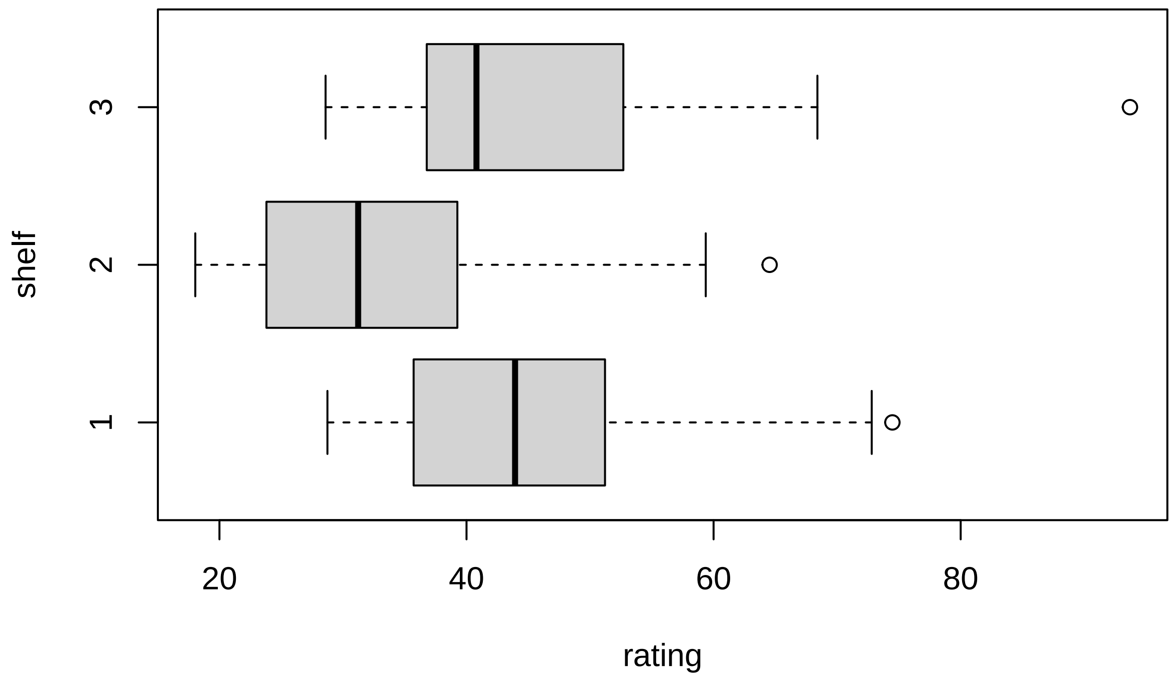
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
boxplot(calories~type,data=cerealData, xlab="Hot and Cold cereals",ylab="Calories")
```




```
boxplot(rating~shelf,data=cerealData,xlab="rating",ylab="shelf",horizontal=TRUE)
```




```
#name ->nominal
#mfr ->nominal
#type ->nominal
#calories ->numerical
#protein ->numerical
#fat ->numerical
#sodium ->numerical
#fiber ->numerical
#carbo ->numerical
#sugars ->numerical
#potass ->numerical
#vitamins ->numerical
#shelf ->ordinal
#weight ->numerical
#cups ->numerical
#rating ->ordinal

#This is a correlation matrix of the numberic and ordinal variables.
dataframeforcorrelationmatrix <- cerealData %>% select(calories,protein,fat,sodium,fiber,carbo,sugars,po
show(dataframeforcorrelationmatrix)
```

```
##   calories protein fat sodium fiber carbo sugars potass vitamins shelf weight
## 1       70       4   1    130  10.0   5.0      6    280       25     3   1.00
## 2      120       3   5     15   2.0   8.0      8    135        0     3   1.00
## 3       70       4   1    260   9.0   7.0      5    320       25     3   1.00
## 4       50       4   0    140  14.0   8.0      0    330       25     3   1.00
```

```
## 5         110       2   2    200   1.0  14.0     8    NA      25    3   1.00
## 6         110       2   2    180   1.5  10.5    10    70      25    1   1.00
## 7         110       2   0    125   1.0  11.0    14    30      25    2   1.00
## 8         130       3   2    210   2.0  18.0     8   100      25    3   1.33
## 9          90       2   1    200   4.0  15.0     6   125      25    1   1.00
## 10         90       3   0    210   5.0  13.0     5   190      25    3   1.00
## 11        120       1   2    220   0.0  12.0    12    35      25    2   1.00
## 12        110       6   2    290   2.0  17.0     1   105      25    1   1.00
## 13        120       1   3    210   0.0  13.0     9    45      25    2   1.00
## 14        110       3   2    140   2.0  13.0     7   105      25    3   1.00
## 15        110       1   1    180   0.0  12.0    13    55      25    2   1.00
## 16        110       2   0    280   0.0  22.0     3    25      25    1   1.00
## 17        100       2   0    290   1.0  21.0     2    35      25    1   1.00
## 18        110       1   0     90   1.0  13.0    12    20      25    2   1.00
## 19        110       1   1    180   0.0  12.0    13    65      25    2   1.00
## 20        110       3   3    140   4.0  10.0     7   160      25    3   1.00
## 21        100       3   0     80   1.0  21.0     0    NA       0    2   1.00
## 22        110       2   0    220   1.0  21.0     3    30      25    3   1.00
## 23        100       2   1    140   2.0  11.0    10   120      25    3   1.00
## 24        100       2   0    190   1.0  18.0     5    80      25    3   1.00
## 25        110       2   1    125   1.0  11.0    13    30      25    2   1.00
## 26        110       1   0    200   1.0  14.0    11    25      25    1   1.00
## 27        100       3   0      0   3.0  14.0     7   100      25    2   1.00
## 28        120       3   2    160   5.0  12.0    10   200      25    3   1.25
## 29        120       3   0    240   5.0  14.0    12   190      25    3   1.33
## 30        110       1   1    135   0.0  13.0    12    25      25    2   1.00
## 31        100       2   0     45   0.0  11.0    15    40      25    1   1.00
## 32        110       1   1    280   0.0  15.0     9    45      25    2   1.00
## 33        100       3   1    140   3.0  15.0     5    85      25    3   1.00
## 34        110       3   0    170   3.0  17.0     3    90      25    3   1.00
## 35        120       3   3     75   3.0  13.0     4   100      25    3   1.00
## 36        120       1   2    220   1.0  12.0    11    45      25    2   1.00
## 37        110       3   1    250   1.5  11.5    10    90      25    1   1.00
## 38        110       1   0    180   0.0  14.0    11    35      25    1   1.00
## 39        110       2   1    170   1.0  17.0     6    60     100    3   1.00
## 40        140       3   1    170   2.0  20.0     9    95     100    3   1.30
## 41        110       2   1    260   0.0  21.0     3    40      25    2   1.00
## 42        100       4   2    150   2.0  12.0     6    95      25    2   1.00
## 43        110       2   1    180   0.0  12.0    12    55      25    2   1.00
## 44        100       4   1      0   0.0  16.0     3    95      25    2   1.00
## 45        150       4   3     95   3.0  16.0    11   170      25    3   1.00
## 46        150       4   3    150   3.0  16.0    11   170      25    3   1.00
## 47        160       3   2    150   3.0  17.0    13   160      25    3   1.50
## 48        100       2   1    220   2.0  15.0     6    90      25    1   1.00
## 49        120       2   1    190   0.0  15.0     9    40      25    2   1.00
## 50        140       3   2    220   3.0  21.0     7   130      25    3   1.33
## 51         90       3   0    170   3.0  18.0     2    90      25    3   1.00
## 52        130       3   2    170   1.5  13.5    10   120      25    3   1.25
## 53        120       3   1    200   6.0  11.0    14   260      25    3   1.33
## 54        100       3   0    320   1.0  20.0     3    45     100    3   1.00
## 55         50       1   0      0   0.0  13.0     0    15       0    3   0.50
## 56         50       2   0      0   1.0  10.0     0    50       0    3   0.50
## 57        100       4   1    135   2.0  14.0     6   110      25    3   1.00
## 58        100       5   2      0   2.7    NA    NA   110       0    1   1.00
```

```
## 59       120        3   1    210   5.0  14.0       12    240         25      2   1.33
## 60       100        3   2    140   2.5  10.5        8    140         25      3   1.00
## 61        90        2   0      0   2.0  15.0        6    110         25      3   1.00
## 62       110        1   0    240   0.0  23.0        2     30         25      1   1.00
## 63       110        2   0    290   0.0  22.0        3     35         25      1   1.00
## 64        80        2   0      0   3.0  16.0        0     95          0      1   0.83
## 65        90        3   0      0   4.0  19.0        0    140          0      1   1.00
## 66        90        3   0      0   3.0  20.0        0    120          0      1   1.00
## 67       110        2   1     70   1.0   9.0       15     40         25      2   1.00
## 68       110        6   0    230   1.0  16.0        3     55         25      1   1.00
## 69        90        2   0     15   3.0  15.0        5     90         25      2   1.00
## 70       110        2   1    200   0.0  21.0        3     35        100      3   1.00
## 71       140        3   1    190   4.0  15.0       14    230        100      3   1.50
## 72       100        3   1    200   3.0  16.0        3    110        100      3   1.00
## 73       110        2   1    250   0.0  21.0        3     60         25      3   1.00
## 74       110        1   1    140   0.0  13.0       12     25         25      2   1.00
## 75       100        3   1    230   3.0  17.0        3    115         25      1   1.00
## 76       100        3   1    200   3.0  17.0        3    110         25      1   1.00
## 77       110        2   1    200   1.0  16.0        8     60         25      1   1.00
##    cups  rating
## 1  0.33 68.40297
## 2  1.00 33.98368
## 3  0.33 59.42551
## 4  0.50 93.70491
## 5  0.75 34.38484
## 6  0.75 29.50954
## 7  1.00 33.17409
## 8  0.75 37.03856
## 9  0.67 49.12025
## 10 0.67 53.31381
## 11 0.75 18.04285
## 12 1.25 50.76500
## 13 0.75 19.82357
## 14 0.50 40.40021
## 15 1.00 22.73645
## 16 1.00 41.44502
## 17 1.00 45.86332
## 18 1.00 35.78279
## 19 1.00 22.39651
## 20 0.50 40.44877
## 21 1.00 64.53382
## 22 1.00 46.89564
## 23 0.75 36.17620
## 24 0.75 44.33086
## 25 1.00 32.20758
## 26 0.75 31.43597
## 27 0.80 58.34514
## 28 0.67 40.91705
## 29 0.67 41.01549
## 30 0.75 28.02576
## 31 0.88 35.25244
## 32 0.75 23.80404
## 33 0.88 52.07690
## 34 0.25 53.37101
```

```
## 35 0.33 45.81172
## 36 1.00 21.87129
## 37 0.75 31.07222
## 38 1.33 28.74241
## 39 1.00 36.52368
## 40 0.75 36.47151
## 41 1.50 39.24111
## 42 0.67 45.32807
## 43 1.00 26.73451
## 44 1.00 54.85092
## 45 1.00 37.13686
## 46 1.00 34.13976
## 47 0.67 30.31335
## 48 1.00 40.10596
## 49 0.67 29.92429
## 50 0.67 40.69232
## 51 1.00 59.64284
## 52 0.50 30.45084
## 53 0.67 37.84059
## 54 1.00 41.50354
## 55 1.00 60.75611
## 56 1.00 63.00565
## 57 0.50 49.51187
## 58 0.67 50.82839
## 59 0.75 39.25920
## 60 0.50 39.70340
## 61 0.50 55.33314
## 62 1.13 41.99893
## 63 1.00 40.56016
## 64 1.00 68.23588
## 65 0.67 74.47295
## 66 0.67 72.80179
## 67 0.75 31.23005
## 68 1.00 53.13132
## 69 1.00 59.36399
## 70 1.00 38.83975
## 71 1.00 28.59278
## 72 1.00 46.65884
## 73 0.75 39.10617
## 74 1.00 27.75330
## 75 0.67 49.78744
## 76 1.00 51.59219
## 77 0.75 36.18756
```

```r
correlation_matrix2 <- cor(dataframeforcorrelationmatrix)
correlation_matrix2
```

```
##             calories      protein          fat       sodium       fiber carbo
## calories  1.00000000  0.019066068  0.498609814  0.300649227 -0.29341275    NA
## protein   0.01906607  1.000000000  0.208430990 -0.054674348  0.50033004    NA
## fat       0.49860981  0.208430990  1.000000000 -0.005407464  0.01671924    NA
## sodium    0.30064923 -0.054674348 -0.005407464  1.000000000 -0.07067501    NA
## fiber    -0.29341275  0.500330043  0.016719237 -0.070675009  1.00000000    NA
## carbo             NA           NA           NA           NA          NA     1
```

```
## sugars            NA           NA           NA           NA          NA    NA
## potass            NA           NA           NA           NA          NA    NA
## vitamins  0.26535630  0.007335371 -0.031156266  0.361476688 -0.03224268    NA
## shelf     0.09723437  0.133864789  0.263691089 -0.069719015  0.29753906    NA
## weight    0.69609108  0.216158486  0.214625033  0.308576451  0.24722563    NA
## cups      0.08719955 -0.244469158 -0.175892142  0.119664615 -0.51306093    NA
## rating   -0.68937603  0.470618465 -0.409283660 -0.401295204  0.58416042    NA
##          sugars potass     vitamins       shelf     weight        cups
## calories     NA     NA  0.265356298  0.09723437  0.6960911  0.08719955
## protein      NA     NA  0.007335371  0.13386479  0.2161585 -0.24446916
## fat          NA     NA -0.031156266  0.26369109  0.2146250 -0.17589214
## sodium       NA     NA  0.361476688 -0.06971902  0.3085765  0.11966461
## fiber        NA     NA -0.032242679  0.29753906  0.2472256 -0.51306093
## carbo        NA     NA           NA          NA         NA          NA
## sugars        1     NA           NA          NA         NA          NA
## potass       NA      1           NA          NA         NA          NA
## vitamins     NA     NA  1.000000000  0.29926167  0.3203241  0.12840454
## shelf        NA     NA  0.299261665  1.00000000  0.1907620 -0.33526876
## weight       NA     NA  0.320324059  0.19076197  1.0000000 -0.19958272
## cups         NA     NA  0.128404543 -0.33526876 -0.1995827  1.00000000
## rating       NA     NA -0.240543611  0.02515882 -0.2981240 -0.20316006
##               rating
## calories -0.68937603
## protein   0.47061846
## fat      -0.40928366
## sodium   -0.40129520
## fiber     0.58416042
## carbo             NA
## sugars            NA
## potass            NA
## vitamins -0.24054361
## shelf     0.02515882
## weight   -0.29812398
## cups     -0.20316006
## rating    1.00000000
```

```
#i. Which pair of variables is most strongly correlated?
#calories and weight with a .6960 correlation
#ii. How can we reduce the number of variables based on these correlations?
#by removing the highly correlated variables the collinearity will improve
#iii. How would the correlations change if we normalized the data first?
#The correlation will not change
```