70-374 HW1 Solution

Sunan Qian

2/22/2021

Problem 1: Find Dateset (40 PTS)

Full credit if you have delivered. 10% bonus points to those of you who are using non-Kaggle datasets.

Problem 2: Read Chapter 1 and 2 of MLR (Machine Learning with R 2nd ed by Lantz) (30 Points)

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

2.1 Go through the code for Chapter 2

2.2 Load "usedcars.csv". Then answer the following questions

```
usedcars <- read.csv("usedcars.csv", stringsAsFactors = FALSE)
```

2.2.1 Name the columns of the data and the data type. (5 pts)

```
str(usedcars)
```

```
## 'data.frame':    150 obs. of  6 variables:
##  $ year        : int  2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...
##  $ model       : chr  "SEL" "SEL" "SEL" "SEL" ...
##  $ price       : int  21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...
##  $ mileage     : int  7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...
##  $ color       : chr  "Yellow" "Gray" "Silver" "Gray" ...
##  $ transmission: chr  "AUTO" "AUTO" "AUTO" "AUTO" ...
```

```
colnames(usedcars)
```

```
## [1] "year"         "model"        "price"        "mileage"
## [5] "color"        "transmission"
```

2.2.2 What's the average price of the used car? Average mileage? (5 pts)

```
mean(usedcars$price)
```

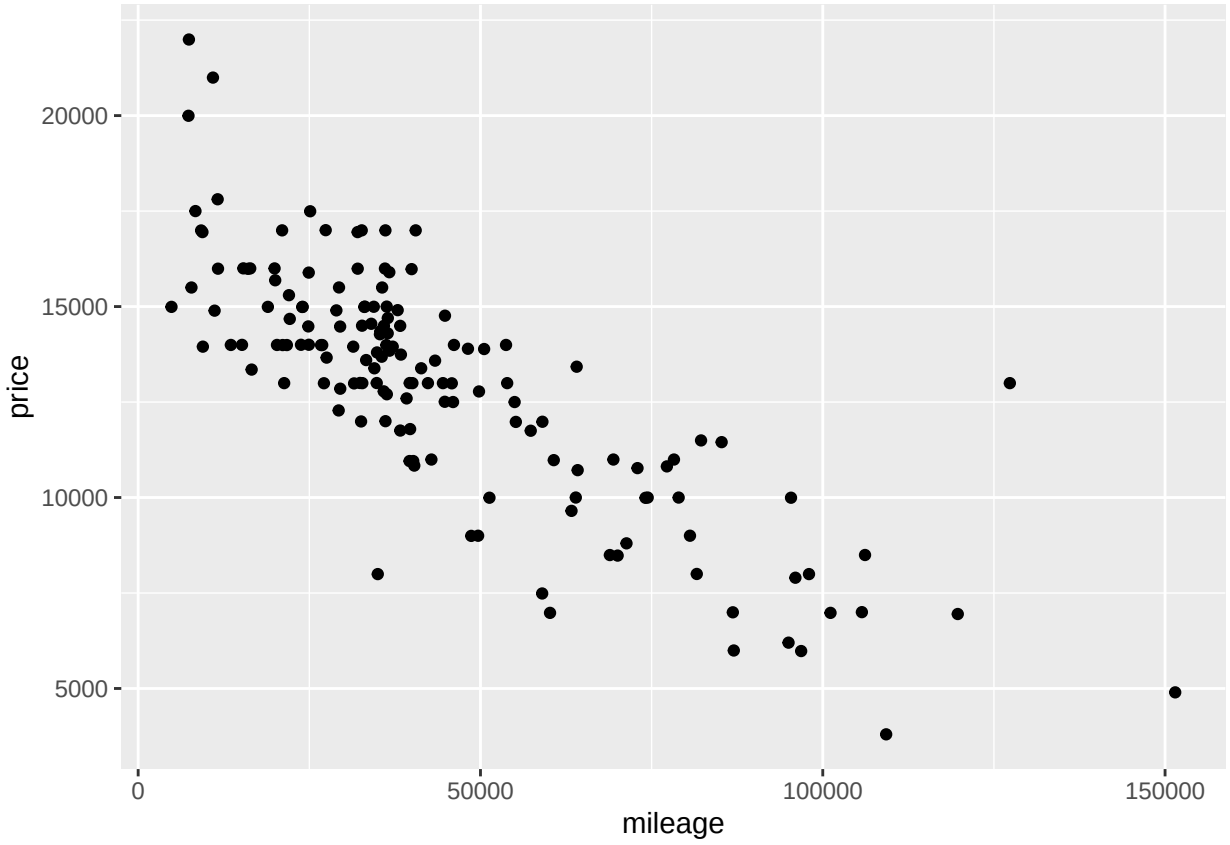
```
## [1] 12961.93
```

```
mean(usedcars$mileage)
```

```
## [1] 44260.65
```

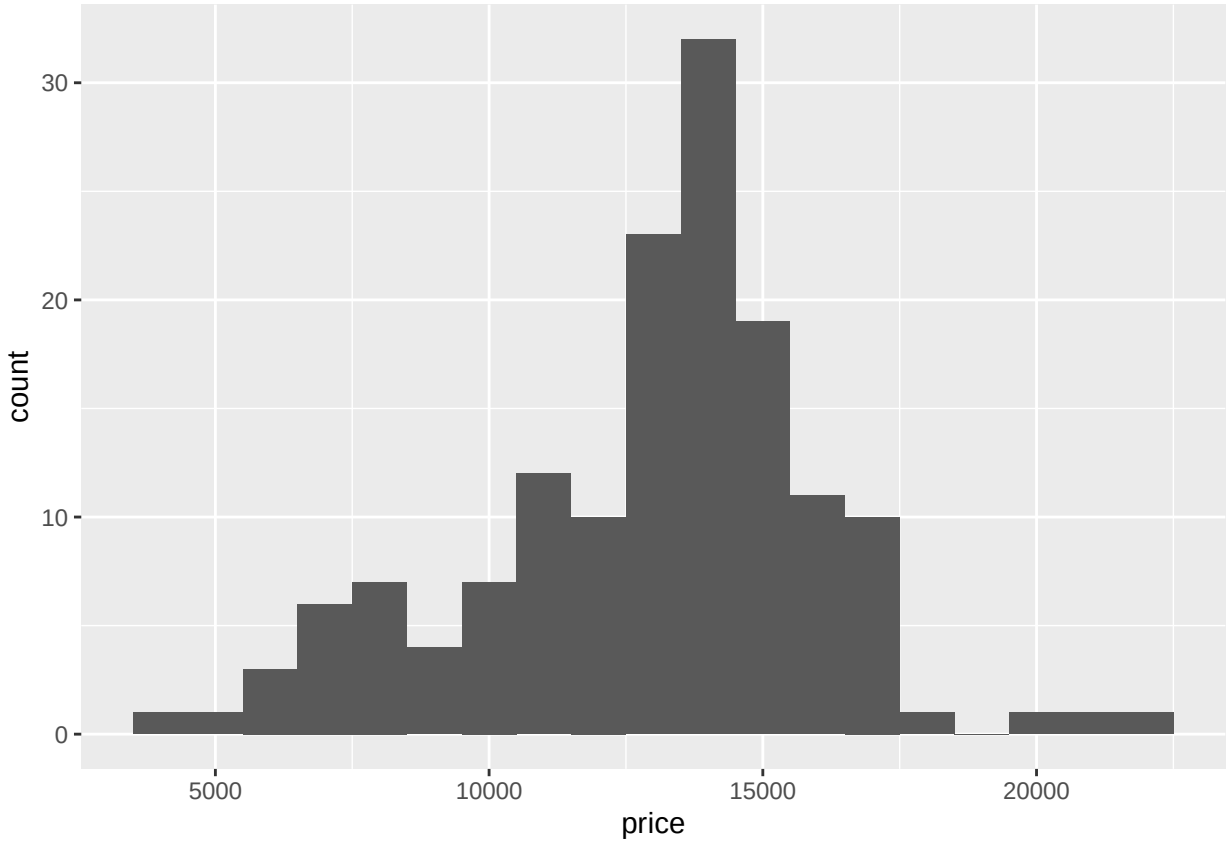
2.2.3 Provide the scatterplot of Price vs. Mileage using ggplot. Label the axis appropriately. (10 pts)

```
ggplot(usedcars, aes(x = mileage,y= price))+
  geom_point()
```



2.2.4 Provide the histogram of Price using ggplot. Label the axis appropriately. (10 pts)

```
ggplot(usedcars, aes(x = price))+
  geom_histogram(binwidth = 1000)
```



Part 3: Exploring the Cereal Data (30 PTS)

3.1 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)). (5 pts)

```
cereals.df <- read.csv("Cereals.csv", stringsAsFactors = FALSE)

sapply(cereals.df[,-c(1:3)], mean, na.rm=TRUE)
```

```
##   calories    protein        fat     sodium      fiber      carbo
## 106.883117   2.545455   1.012987 159.675325   2.151948  14.802632
##     sugars     potass   vitamins      shelf     weight       cups
##   7.026316  98.666667  28.246753   2.207792   1.029610   0.821039
##     rating
##  42.665705
```

```
sapply(cereals.df[,-c(1:3)], median, na.rm=TRUE)
```

```
##  calories   protein       fat    sodium     fiber     carbo    sugars
## 110.00000   3.00000   1.00000 180.00000   2.00000  14.50000   7.00000
##    potass  vitamins     shelf    weight      cups    rating
##  90.00000  25.00000   2.00000   1.00000   0.75000  40.40021
```

```
sapply(cereals.df[,-c(1:3)], min, na.rm=TRUE)
```

```
## calories  protein      fat   sodium    fiber    carbo   sugars   potass
## 50.00000  1.00000  0.00000  0.00000  0.00000  5.00000  0.00000 15.00000
## vitamins    shelf   weight     cups   rating
##  0.00000  1.00000  0.50000  0.25000 18.04285
```

```
sapply(cereals.df[,-c(1:3)], max, na.rm=TRUE)
```

```
##  calories   protein       fat    sodium     fiber     carbo    sugars
## 160.00000   6.00000   5.00000 320.00000  14.00000  23.00000  15.00000
##    potass  vitamins     shelf    weight      cups    rating
## 330.00000 100.00000   3.00000   1.50000   1.50000  93.70491
```

```
sapply(cereals.df[,-c(1:3)], sd, na.rm=TRUE)
```

```
##   calories    protein        fat     sodium      fiber      carbo
## 19.4841191  1.0947897  1.0064726 83.8322952  2.3833640  3.9073256
##     sugars     potass   vitamins      shelf     weight       cups
##  4.3786564 70.4106360 22.3425225  0.8325241  0.1504768  0.2327161
##     rating
## 14.0472887
```

3.2 Use R to plot a histogram for each of the quantitative variables. Based on the histograms and summary statistics, answer the following questions:

You will get full points if you can come up with the plots and the answers. You don't need to use the approach I am using.

There are two ways to plot these histograms efficiently:

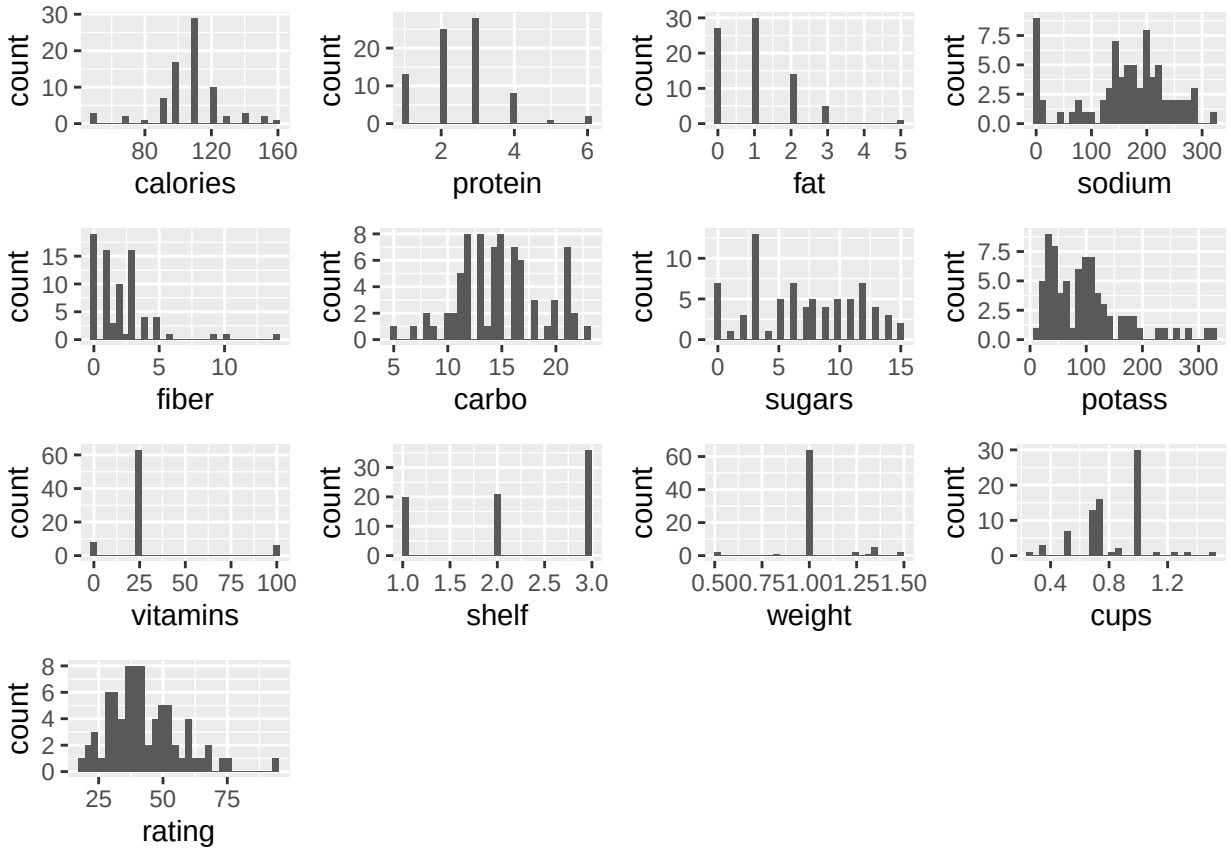
Option 1:

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.6.3
```

```
p <- list()
j = 0
for (i in c(4:16)){
  j = j+1
  p[[j]] <- ggplot(cereals.df, aes_string(x=names(cereals.df)[i]))+geom_histogram(bins = 30)
}

do.call(grid.arrange,p)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
## Warning: Removed 1 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```



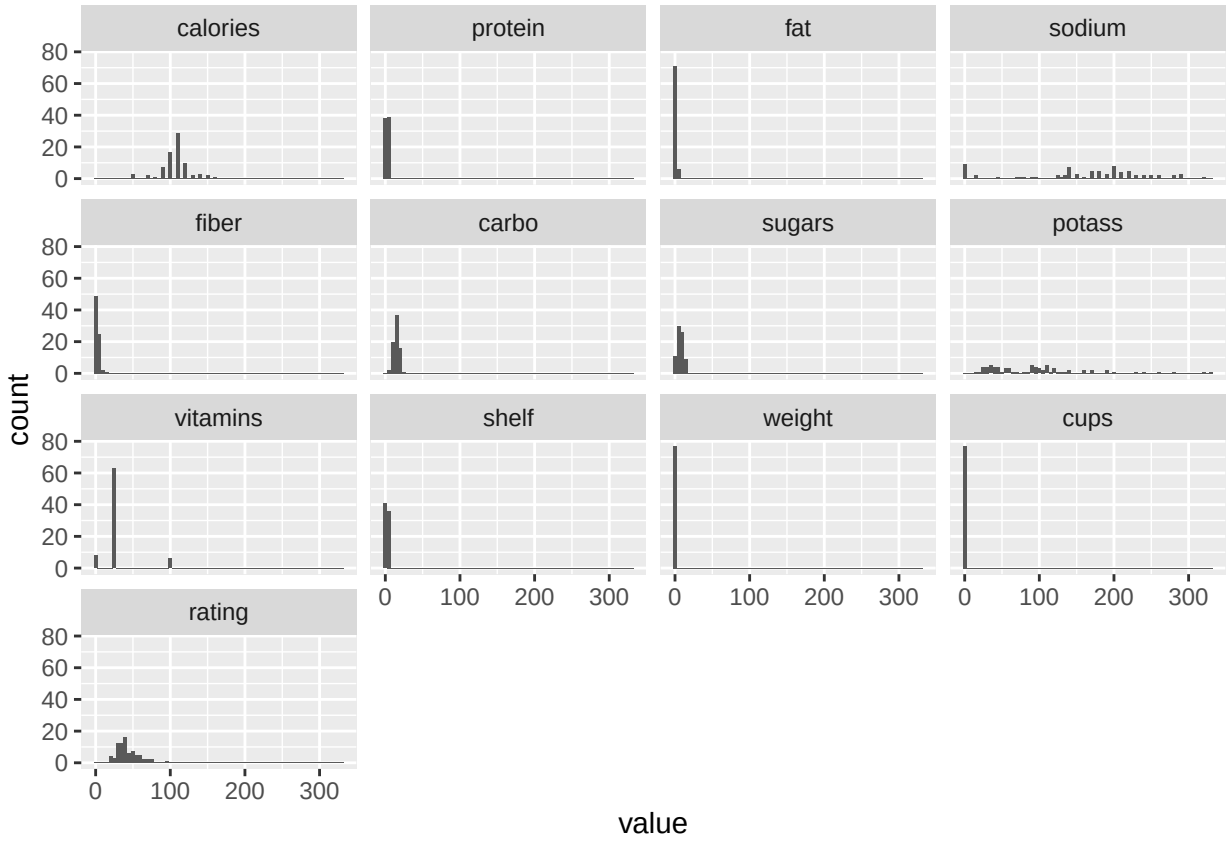
Option 2: (more advanced)

```
library(reshape2)

d <- melt(cereals.df[,c(4:16)])
```

```
## No id variables; using all as measure variables
```

```
ggplot(d,aes(x = value)) +
  facet_wrap(~variable) +
  geom_histogram(binwidth = 5)
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```



3.2.1 Which variables have the largest variability? (5 pts)

Sodium and Potass have the largest variability.

3.2.2 Which variables seem skewed? (5 pts)

Variable Fiber is positively skewed.

Variable Potass is positively skewed.

Variable Rating is positively skewed.

3.2.3 Are there any values that seem extreme? (5 pts)

For the following variables, extreme values (outliers) are present:

Protein (extreme values are 5 and 6)

Fat (extreme value is 5)

Fiber (extreme values are 14, 10, 9 and 6)

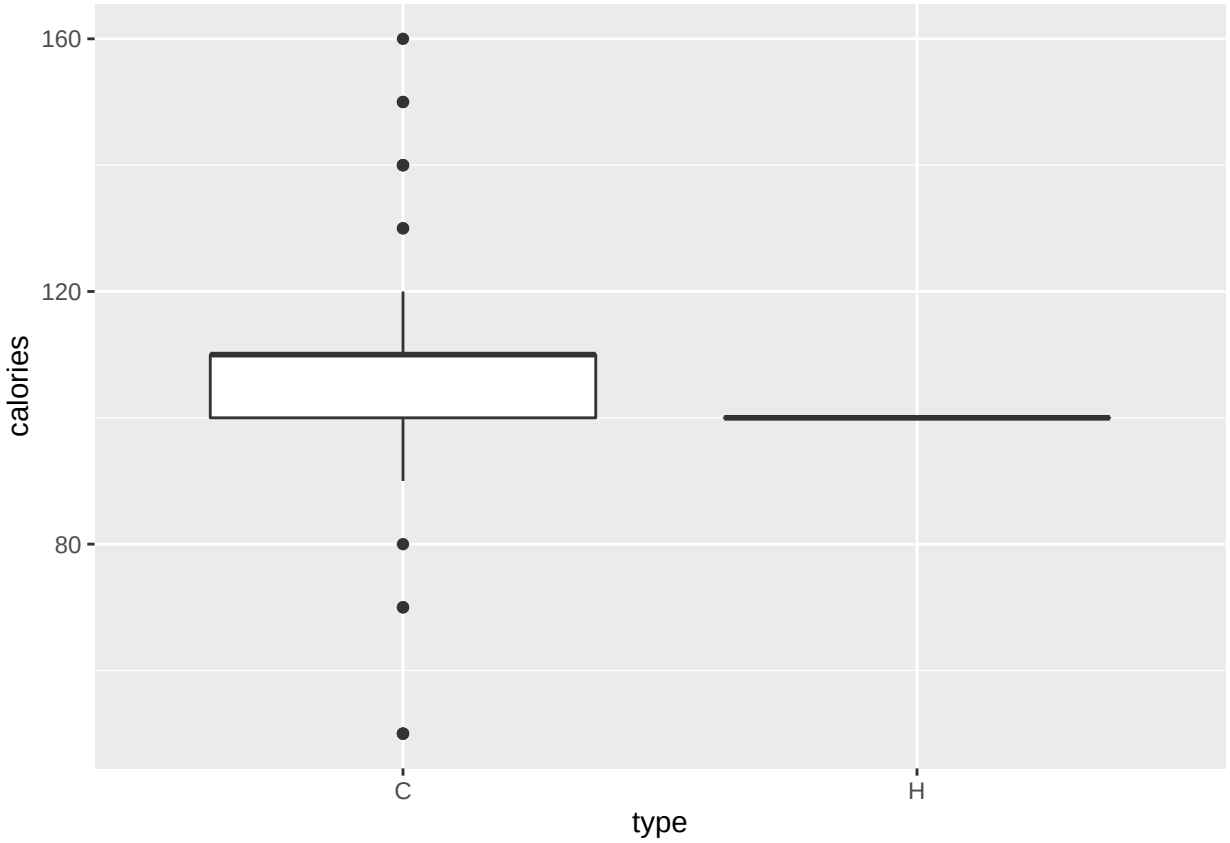
Vitamins (extreme value is 100)

Weight (extreme values are 0.5, 1.33, 1.5)

Rating (extreme value is 93.7)

3.3 Use R to plot a side-by-side boxplot comparing the calories in hot vs. cold cereals.

```
ggplot(cereals.df, aes(y = calories, x = type))+
  geom_boxplot()
```

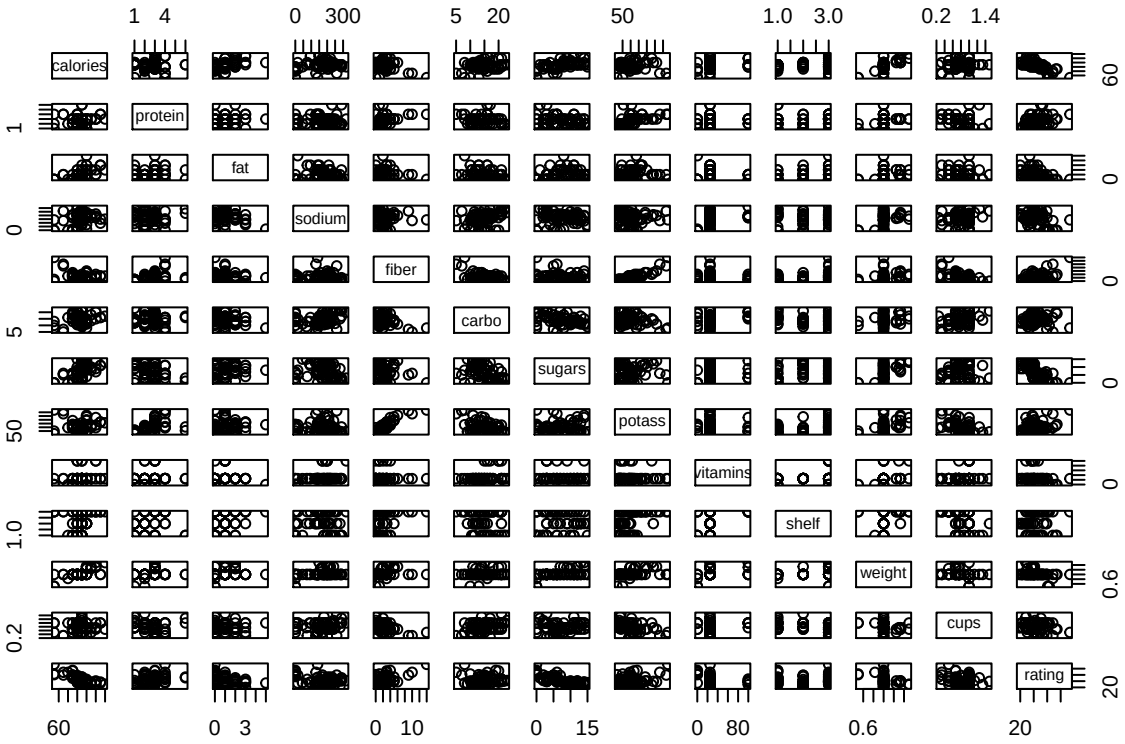


3.3.1 What does this plot show us? (5 pts)

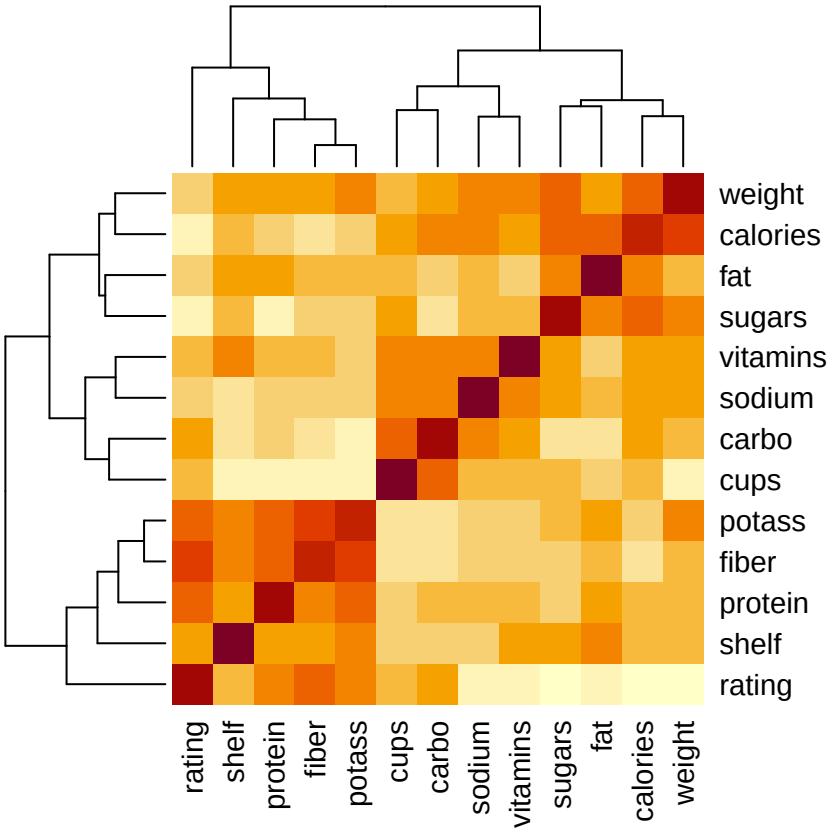
We see that in cold cereals, the different cereals vary in the amount of calories mainly between approximately 90-120, whereas all hot cereals have 100 calories.

3.4 Compute the correlation table for the quantitative variable (function cor()). In addition, generate a correlation heatmap of these variables (also show correlation value on this heatmap).

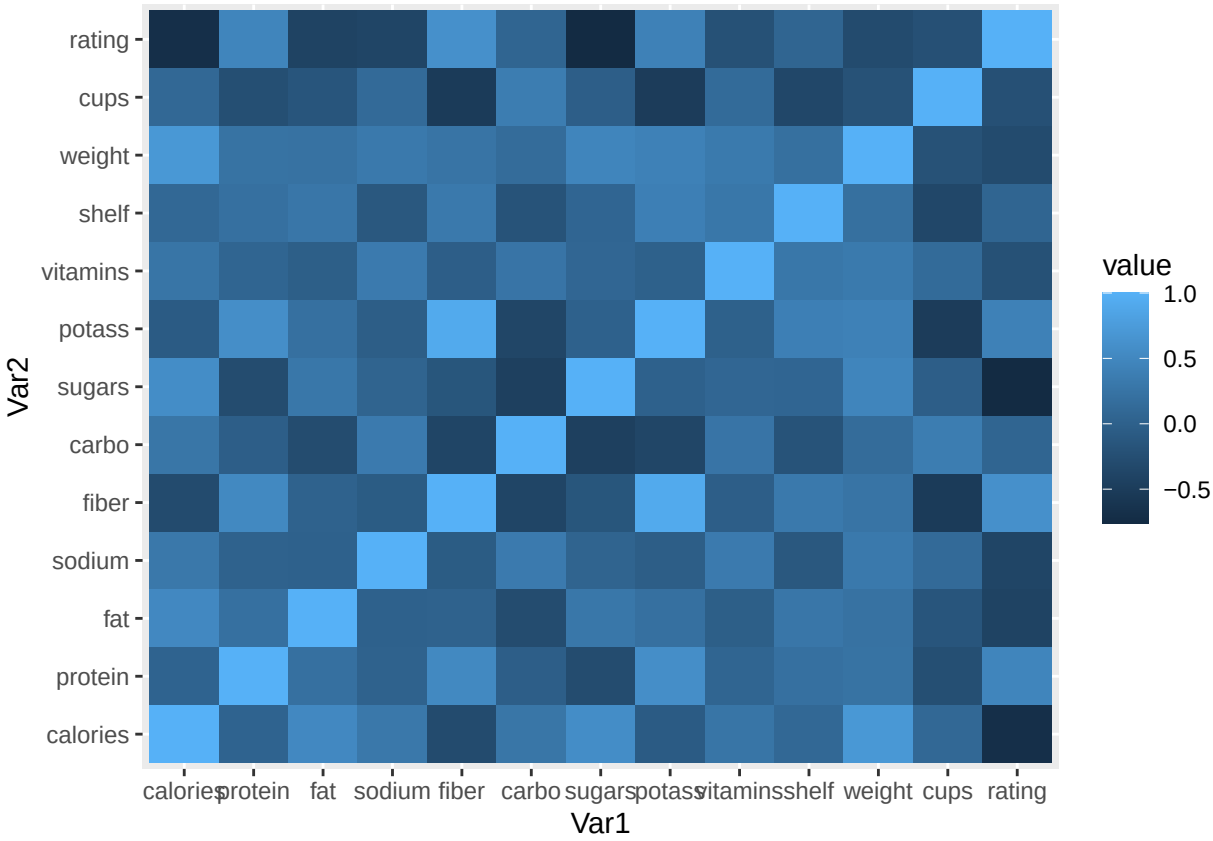
```
options(digits = 1) # to print less decimals in correlation matrix
correlation <- cor(na.omit(cereals.df[,-c(1:3)]))

# Plot 1
plot(na.omit(cereals.df[,-c(1:3)]))
```




```
# Plot 2
heatmap(correlation, scale="column")
```



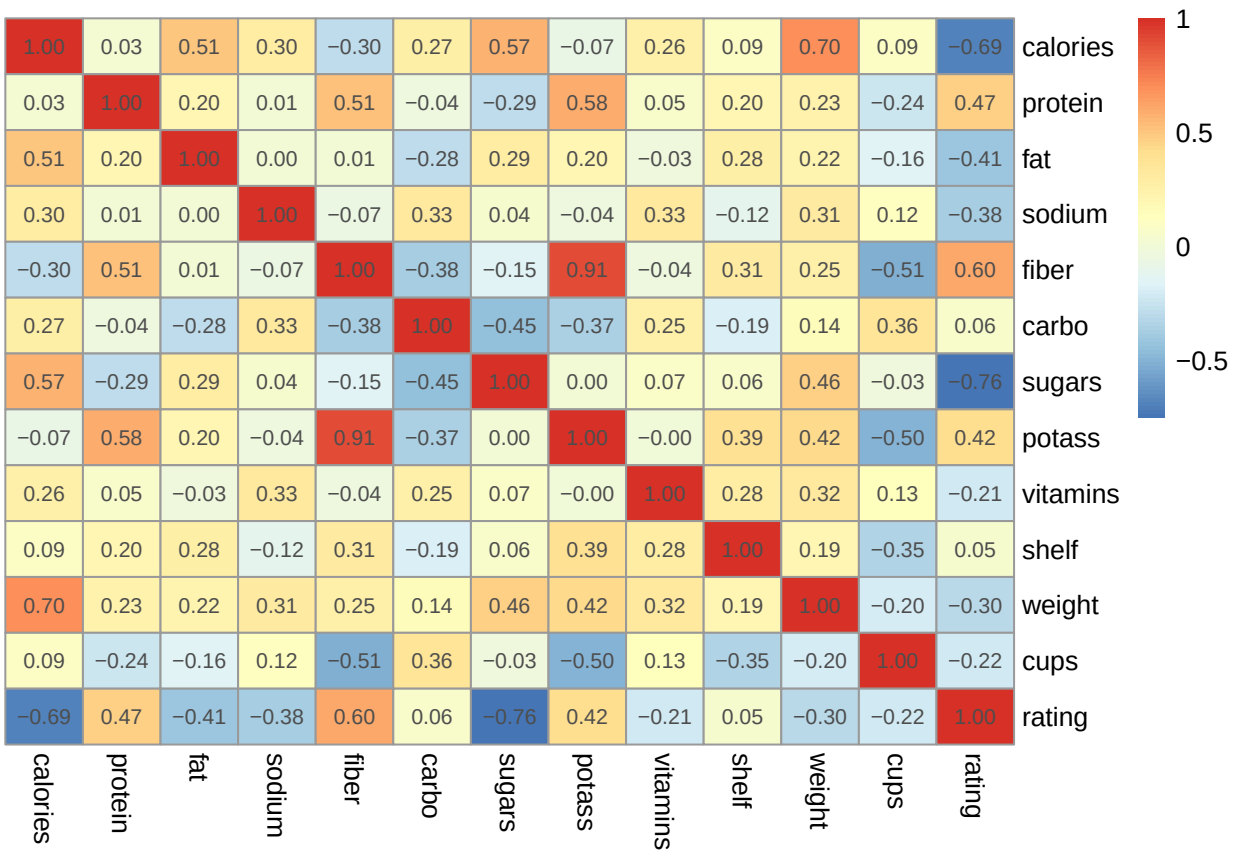

```
#heatmap(correlation, Colv = NA, Rowv = NA, scale="column")

# Plot 3
melted_cor <- melt(correlation)
ggplot(data = melted_cor, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
```




```
# Plot 4
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 3.6.3
```

```
names(correlation) <- paste("X", 1:10)
pheatmap(correlation, display_numbers = T, cluster_rows = F, cluster_cols = F)
```



3.4.1 Which pair of variables is most strongly correlated? (5 pts)

From the correlation matrix, fiber and potass are the most strongly (positively) correlated. Fat and calories are also positively correlated.

(Check the table "correlation" in R for the full correlation matrix.)