

2347126_CIA_C2.R

ASUS

2024-01-25

#R1: Understanding of selected Dataset:

```
setwd("C:/Users/ASUS/Desktop/2nd-trimester/R/AppliedStatUsinR")
cereal=read.csv("UScereal1.csv",header=T)
cereal2=cereal
cal_mean=mean(cereal$calories,na.rm=T)
str(cereal)
```

```
## 'data.frame':    65 obs. of  12 variables:
## $ Name      : chr  "100% Bran" "All-Bran" "All-Bran with Extra Fiber" "Apple Cinnamon Cheerios" ...
## $ mfr       : chr  "N" "K" "K" "G" ...
## $ calories  : num  212 212 100 147 110 ...
## $ protein   : num  12.12 12.12 8 2.67 2 ...
## $ fat       : num  3.03 3.03 0 2.67 0 2.67 1.49 0 2.67 NA ...
## $ sodium    : num  394 788 280 240 125 ...
## $ fibre     : num  30.3 27.3 28 2 1 ...
## $ carbo     : num  15.2 21.2 16 14 11 ...
## $ sugars    : num  18.2 15.2 0 13.3 14 ...
## $ shelf     : int  3 3 3 1 2 3 1 3 2 1 ...
## $ potassium : num  848.5 969.7 660 93.3 30 ...
## $ vitamins  : chr  "enriched" "enriched" "enriched" "enriched" ...
```

```
summary(cereal)
```

```
##      Name           mfr           calories      protein
## Length:65      Length:65      Min.   : 50.0    Min.   : 0.750
## Class :character Class :character 1st Qu.:110.0    1st Qu.: 2.000
## Mode  :character Mode  :character Median :137.2    Median : 3.000
##                                     Mean  :149.6    Mean  : 3.726
##                                     3rd Qu.:179.1    3rd Qu.: 4.480
##                                     Max.   :440.0    Max.   :12.120
##                                     NA's   :1       NA's   :1
##      fat           sodium        fibre        carbo
## Min.   :0.00      Min.   : 0.0    Min.   : 0.000    Min.   :10.53
## 1st Qu.:0.00      1st Qu.:180.0    1st Qu.: 0.000    1st Qu.:14.92
## Median :1.00      Median :235.4    Median : 2.000    Median :18.67
## Mean   :1.42      Mean   :238.6    Mean   : 3.871    Mean   :20.01
## 3rd Qu.:2.00      3rd Qu.:290.0    3rd Qu.: 4.480    3rd Qu.:22.39
## Max.   :9.09      Max.   :787.9    Max.   :30.300    Max.   :68.00
## NA's   :1        NA's   :1
##      sugars        shelf        potassium      vitamins
## Min.   : 0.00      Min.   :1.000    Min.   : 15.00    Length:65
## 1st Qu.: 3.75      1st Qu.:1.000    1st Qu.: 45.00    Class :character
## Median :12.00      Median :2.000    Median : 94.96    Mode  :character
```

```
## Mean :10.07 Mean :2.169 Mean :158.69
## 3rd Qu.:14.00 3rd Qu.:3.000 3rd Qu.:220.00
## Max. :20.90 Max. :3.000 Max. :969.70
## NA's :1 NA's :1
```

```
cereal$mfr=factor(cereal$mfr)
cereal$shelf=factor(cereal$shelf,ordered = T)
cereal$vitamins=factor(cereal$vitamins)

names=c(unique(cereal$mfr))
for(name in names){
  print(name)
  print(mean(cereal$protein[cereal$mfr==name]))
}
```

```
## [1] "N"
## [1] 7.026667
## [1] "K"
## [1] NA
## [1] "G"
## [1] 2.885
## [1] "R"
## [1] 2.604
## [1] "P"
## [1] 4.698889
## [1] "Q"
## [1] 3.46
```

```
max(cereal$protein[cereal$mfr=="G"])
```

```
## [1] 6
```

```
max(cereal$protein[cereal$mfr=="K"])
```

```
## [1] NA
```

```
max(cereal$protein[cereal$mfr=="N"])
```

```
## [1] 12.12
```

```
max(cereal$protein[cereal$mfr=="P"])
```

```
## [1] 12
```

```
max(cereal$protein[cereal$mfr=="Q"])
```

```
## [1] 8
```

```
max(cereal$protein[cereal$mfr=="R"])
```

```
## [1] 4.48
```

```
#R2 - Descriptive Analysis
```

```
#1.
```

```
is.na(cereal)
```

```
##      Name  mfr calories protein  fat sodium fibre carbo sugars shelf
## [1,] FALSE FALSE    FALSE  FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [2,] FALSE FALSE    FALSE  FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [3,] FALSE FALSE    FALSE  FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
```

[illegible]

```

## [58,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [59,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [60,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [61,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [62,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [63,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [64,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
## [65,] FALSE FALSE      FALSE  FALSE FALSE  FALSE FALSE  FALSE FALSE
##      potassium vitamins
## [1,]      FALSE      FALSE
## [2,]      FALSE      FALSE
## [3,]      FALSE      FALSE
## [4,]      FALSE      FALSE
## [5,]      FALSE      FALSE
## [6,]      FALSE      FALSE
## [7,]      TRUE       FALSE
## [8,]      FALSE      FALSE
## [9,]      FALSE      FALSE
## [10,]     FALSE      FALSE
## [11,]     FALSE      FALSE
## [12,]     FALSE      FALSE
## [13,]     FALSE      FALSE
## [14,]     FALSE      FALSE
## [15,]     FALSE      FALSE
## [16,]     FALSE      FALSE
## [17,]     FALSE      FALSE
## [18,]     FALSE      FALSE
## [19,]     FALSE      FALSE
## [20,]     FALSE      FALSE
## [21,]     FALSE      FALSE
## [22,]     FALSE      FALSE
## [23,]     FALSE      FALSE
## [24,]     FALSE      FALSE
## [25,]     FALSE      FALSE
## [26,]     FALSE      FALSE
## [27,]     FALSE      FALSE
## [28,]     FALSE      FALSE
## [29,]     FALSE      FALSE
## [30,]     FALSE      FALSE
## [31,]     FALSE      FALSE
## [32,]     FALSE      FALSE
## [33,]     FALSE      FALSE
## [34,]     FALSE      FALSE
## [35,]     FALSE      FALSE
## [36,]     FALSE      FALSE
## [37,]     FALSE      FALSE
## [38,]     FALSE      FALSE
## [39,]     FALSE      FALSE
## [40,]     FALSE      FALSE
## [41,]     FALSE      FALSE
## [42,]     FALSE      FALSE
## [43,]     FALSE      FALSE
## [44,]     FALSE      FALSE
## [45,]     FALSE      FALSE

```

```
## [46,] FALSE FALSE
## [47,] FALSE FALSE
## [48,] FALSE FALSE
## [49,] FALSE FALSE
## [50,] FALSE FALSE
## [51,] FALSE FALSE
## [52,] FALSE FALSE
## [53,] FALSE FALSE
## [54,] FALSE FALSE
## [55,] FALSE FALSE
## [56,] FALSE FALSE
## [57,] FALSE FALSE
## [58,] FALSE FALSE
## [59,] FALSE FALSE
## [60,] FALSE FALSE
## [61,] FALSE FALSE
## [62,] FALSE FALSE
## [63,] FALSE FALSE
## [64,] FALSE FALSE
## [65,] FALSE FALSE
```

```
# install.packages("moments")
library(moments)
col_names = c("calories", "protein", "fat", "sodium", "fibre", "carbo", "sugars", "shelf", "potassium", "vitamin")
for (col_name in col_names) {
  if (is.numeric(cereal[[col_name]])) {
    skew_value = skewness(cereal[[col_name]], na.rm = TRUE)
    if (skew_value < 0) {
      cereal[[col_name]][is.na(cereal[[col_name]])] = min(cereal[[col_name]], na.rm = TRUE)
    } else if (skew_value == 0) {
      cereal[[col_name]][is.na(cereal[[col_name]])] = mean(cereal[[col_name]], na.rm = TRUE)
    } else {
      cereal[[col_name]][is.na(cereal[[col_name]])] = max(cereal[[col_name]], na.rm = TRUE)
    }
  }
}
is.na(cereal)
```

```
##      Name  mfr calories protein   fat sodium fibre carbo sugars shelf
## [1,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [2,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [3,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [4,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [5,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [6,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [7,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [8,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [9,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [10,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [11,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [12,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [13,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [14,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [15,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
## [16,] FALSE FALSE    FALSE    FALSE FALSE  FALSE FALSE FALSE  FALSE FALSE
```


##	[5,]	FALSE	FALSE
##	[6,]	FALSE	FALSE
##	[7,]	FALSE	FALSE
##	[8,]	FALSE	FALSE
##	[9,]	FALSE	FALSE
##	[10,]	FALSE	FALSE
##	[11,]	FALSE	FALSE
##	[12,]	FALSE	FALSE
##	[13,]	FALSE	FALSE
##	[14,]	FALSE	FALSE
##	[15,]	FALSE	FALSE
##	[16,]	FALSE	FALSE
##	[17,]	FALSE	FALSE
##	[18,]	FALSE	FALSE
##	[19,]	FALSE	FALSE
##	[20,]	FALSE	FALSE
##	[21,]	FALSE	FALSE
##	[22,]	FALSE	FALSE
##	[23,]	FALSE	FALSE
##	[24,]	FALSE	FALSE
##	[25,]	FALSE	FALSE
##	[26,]	FALSE	FALSE
##	[27,]	FALSE	FALSE
##	[28,]	FALSE	FALSE
##	[29,]	FALSE	FALSE
##	[30,]	FALSE	FALSE
##	[31,]	FALSE	FALSE
##	[32,]	FALSE	FALSE
##	[33,]	FALSE	FALSE
##	[34,]	FALSE	FALSE
##	[35,]	FALSE	FALSE
##	[36,]	FALSE	FALSE
##	[37,]	FALSE	FALSE
##	[38,]	FALSE	FALSE
##	[39,]	FALSE	FALSE
##	[40,]	FALSE	FALSE
##	[41,]	FALSE	FALSE
##	[42,]	FALSE	FALSE
##	[43,]	FALSE	FALSE
##	[44,]	FALSE	FALSE
##	[45,]	FALSE	FALSE
##	[46,]	FALSE	FALSE
##	[47,]	FALSE	FALSE
##	[48,]	FALSE	FALSE
##	[49,]	FALSE	FALSE
##	[50,]	FALSE	FALSE
##	[51,]	FALSE	FALSE
##	[52,]	FALSE	FALSE
##	[53,]	FALSE	FALSE
##	[54,]	FALSE	FALSE
##	[55,]	FALSE	FALSE
##	[56,]	FALSE	FALSE
##	[57,]	FALSE	FALSE
##	[58,]	FALSE	FALSE

```
## [59,] FALSE FALSE
## [60,] FALSE FALSE
## [61,] FALSE FALSE
## [62,] FALSE FALSE
## [63,] FALSE FALSE
## [64,] FALSE FALSE
## [65,] FALSE FALSE
```

#2.

```
summary(cereal)
```

```
##      Name      mfr      calories      protein      fat
## Length:65      G:22   Min.   : 50.0   Min.   : 0.750   Min.   :0.000
## Class :character K:21   1st Qu.:110.0   1st Qu.: 2.000   1st Qu.:0.000
## Mode  :character N: 3   Median :140.0   Median : 3.000   Median :1.000
##                P: 9   Mean   :154.1   Mean   : 3.855   Mean   :1.538
##                Q: 5   3rd Qu.:179.1   3rd Qu.: 4.480   3rd Qu.:2.000
##                R: 5   Max.   :440.0   Max.   :12.120   Max.   :9.090
##      sodium      fibre      carbo      sugars      shelf
## Min.   : 0.0   Min.   : 0.000   Min.   :10.53   Min.   : 0.000   1:18
## 1st Qu.:180.0   1st Qu.: 0.000   1st Qu.:15.00   1st Qu.: 3.000   2:18
## Median :238.8   Median : 2.000   Median :18.67   Median :12.000   3:29
## Mean   :247.1   Mean   : 3.871   Mean   :20.75   Mean   : 9.916
## 3rd Qu.:290.0   3rd Qu.: 4.480   3rd Qu.:22.39   3rd Qu.:14.000
## Max.   :787.9   Max.   :30.300   Max.   :68.00   Max.   :20.900
##      potassium      vitamins
## Min.   : 15.00   100%   : 5
## 1st Qu.: 45.00   enriched:57
## Median : 96.59   none    : 3
## Mean   :171.17
## 3rd Qu.:220.00
## Max.   :969.70
```

#R3 - Exploratory Analysis

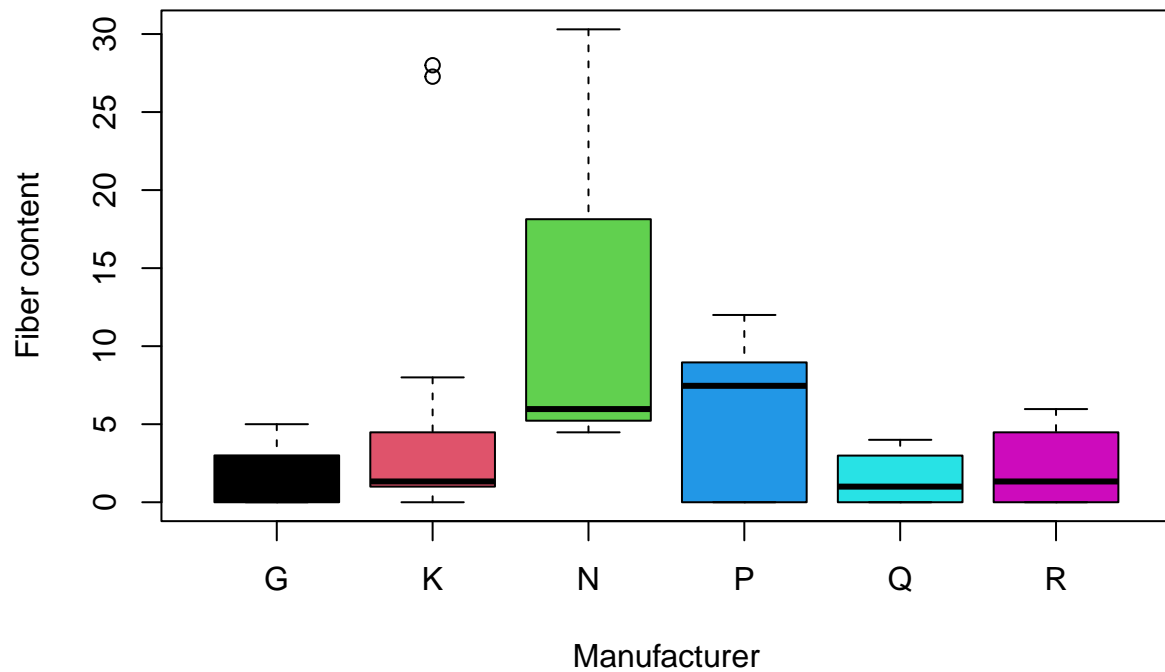
#1.

```
boxplot(fibre~mfr,data=cereal,
        main="Box plot for relation between Fiber and manufactuere",
        ylab="Fiber content",
        xlab="Manufacturer",
        col=c(1:8))
```

2.

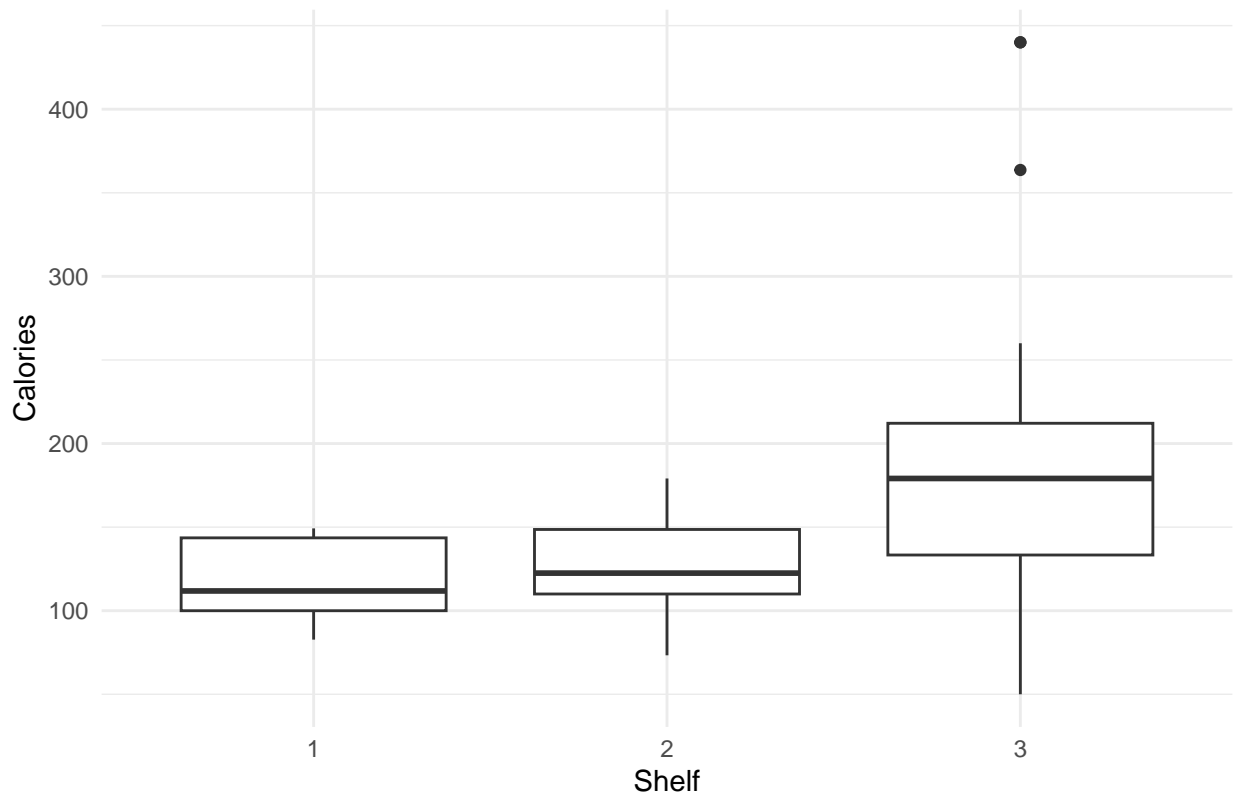
```
library(ggplot2)
```


Box plot for relation between Fiber and manufactuere



```
ggplot(cereal, aes(x = shelf, y = calories)) +  
  geom_boxplot() +  
  labs(  
    title = "Boxplot of Calories for Each Shelf",  
    x = "Shelf",  
    y = "Calories"  
  ) +  
  theme_minimal()
```

Boxplot of Calories for Each Shelf



```
#3.
library(corrplot)

## corrplot 0.92 loaded

str(cereal)

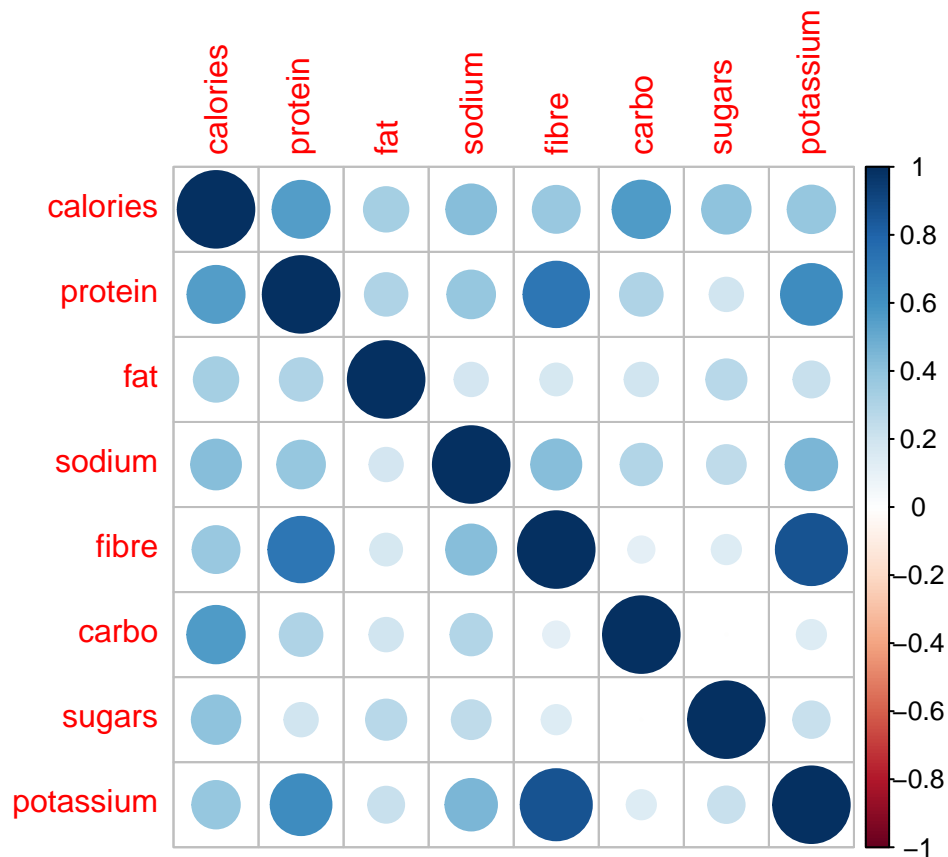
## 'data.frame': 65 obs. of 12 variables:
## $ Name : chr "100% Bran" "All-Bran" "All-Bran with Extra Fiber" "Apple Cinnamon Cheerios" ...
## $ mfr : Factor w/ 6 levels "G","K","N","P",...: 3 2 2 1 2 1 6 4 5 1 ...
## $ calories : num 212 212 100 147 110 ...
## $ protein : num 12.12 12.12 8 2.67 2 ...
## $ fat : num 3.03 3.03 0 2.67 0 2.67 1.49 0 2.67 9.09 ...
## $ sodium : num 394 788 280 240 125 ...
## $ fibre : num 30.3 27.3 28 2 1 ...
## $ carbo : num 15.2 21.2 16 14 11 ...
## $ sugars : num 18.2 15.2 0 13.3 14 ...
## $ shelf : Ord.factor w/ 3 levels "1"<"2"<"3": 3 3 3 1 2 3 1 3 2 1 ...
## $ potassium: num 848.5 969.7 660 93.3 30 ...
## $ vitamins : Factor w/ 3 levels "100%","enriched",...: 2 2 2 2 2 2 2 2 2 2 ...

num_variables=c("calories","protein","fat","sodium","fibre","carbo","sugars","potassium")
num_var = cor(cereal[, num_variables])
print(num_var)

## calories protein fat sodium fibre carbo
## calories 1.0000000 0.5542111 0.3367551 0.4221615 0.3751820 0.5619747434
## protein 0.5542111 1.0000000 0.3082560 0.3875657 0.7298418 0.3096598790
```

```
## fat      0.3367551 0.3082560 1.0000000 0.1899921 0.1730156 0.1907329559
## sodium   0.4221615 0.3875657 0.1899921 1.0000000 0.4272392 0.2921130507
## fibre    0.3751820 0.7298418 0.1730156 0.4272392 1.0000000 0.1179358410
## carbo    0.5619747 0.3096599 0.1907330 0.2921131 0.1179358 1.0000000000
## sugars   0.4002380 0.1903912 0.2751081 0.2569413 0.1460837 -0.0004772179
## potassium 0.3843515 0.6292037 0.2206087 0.4524640 0.8621269 0.1456948311
##          sugars potassium
## calories  0.4002380271 0.3843515
## protein   0.1903912022 0.6292037
## fat        0.2751081158 0.2206087
## sodium     0.2569412601 0.4524640
## fibre      0.1460837329 0.8621269
## carbo     -0.0004772179 0.1456948
## sugars     1.0000000000 0.2264985
## potassium  0.2264985146 1.0000000
```

```
corrplot(num_var, method = "circle")
```



```
#R-4 - Model Building
#1.
num_vars = sapply(cereal, is.numeric)
mean_values = numeric(length = sum(num_vars))

i = 1
for (col in names(cereal)[num_vars]) {
  mean_values[i] = mean(cereal[[col]], na.rm = TRUE)
  i = i + 1
}
```

```

}
mean_values

## [1] 154.111231   3.855077   1.537692 247.087692   3.870923  20.747077   9.916462
## [8] 171.167846

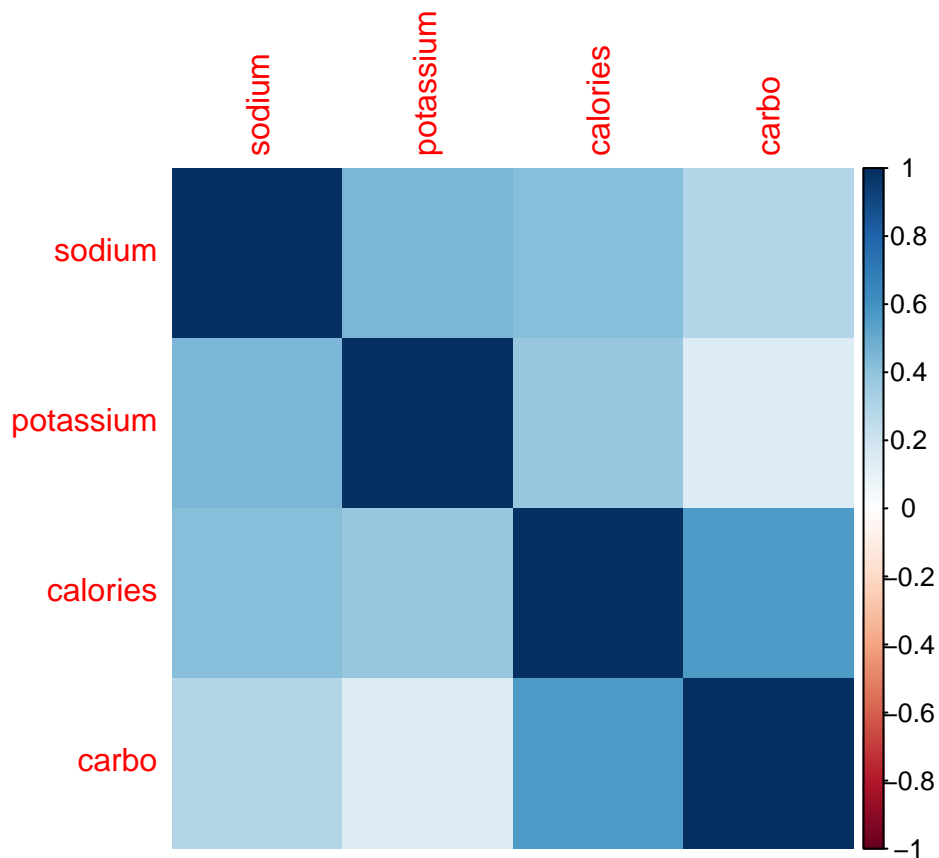
top_four_indices = order(mean_values, decreasing = TRUE)[1:4]
top_four_variable_names = names(cereal)[num_vars][top_four_indices]
GreaterMeanFour = cereal[, top_four_variable_names]
GreaterMeanFour

##      sodium potassium calories carbo
## 1  393.94      848.48   212.12 15.15
## 2  787.88      969.70   212.12 21.21
## 3  280.00      660.00   100.00 16.00
## 4  240.00       93.33   146.67 14.00
## 5  125.00       30.00   110.00 11.00
## 6  280.00     133.33   173.33 24.00
## 7  298.51      969.70   134.33 22.39
## 8  313.43      283.58   440.00 19.40
## 9  293.33       46.67   160.00 16.00
## 10 232.00       84.00    88.00 13.60
## 11 280.00       60.00   160.00 68.00
## 12 280.00     210.00   220.00 26.00
## 13 180.00       55.00   110.00 12.00
## 14 280.00       25.00   110.00 22.00
## 15 290.00       35.00   100.00 21.00
## 16  90.00       20.00   110.00 13.00
## 17 180.00       65.00   110.00 12.00
## 18 280.00     320.00   220.00 20.00
## 19 220.00       30.00   110.00 21.00
## 20 787.88     160.00   133.33 14.67
## 21 253.33     106.67   133.33 24.00
## 22 125.00       30.00   110.00 11.00
## 23 266.67       33.33   146.67 18.67
## 24   0.00     125.00   125.00 17.50
## 25 238.81     298.51   179.10 17.91
## 26 358.21     283.58   179.10 20.90
## 27 180.00       33.33   146.67 17.33
## 28  51.14       45.45   113.64 12.50
## 29 373.33       60.00   146.67 20.00
## 30 159.09       96.59   113.64 17.05
## 31 680.00     360.00   440.00 68.00
## 32 227.27     303.03   363.64 39.39
## 33 220.00       45.00   120.00 12.00
## 34 333.33     120.00   146.67 15.33
## 35 135.34       26.32    82.71 10.53
## 36 226.67     126.67   186.67 26.67
## 37 173.33       26.67    73.33 14.00
## 38 223.88     141.79   149.25 17.91
## 39 180.00       55.00   110.00 12.00
## 40 223.88     238.81   238.81 25.37
## 41 220.00       90.00   100.00 15.00
## 42 283.58       59.70   179.10 22.39
## 43 328.36     194.03   208.96 31.34

```

```
## 44 340.00    240.00    260.00 27.00
## 45 298.51    388.06    179.10 16.42
## 46 320.00     45.00    100.00 20.00
## 47  0.00     15.00     50.00 13.00
## 48 270.00    220.00    200.00 28.00
## 49 280.00    320.00    160.00 18.67
## 50 280.00    280.00    200.00 21.00
## 51  0.00    220.00    180.00 30.00
## 52 212.39     26.55     97.35 20.35
## 53 290.00     35.00    110.00 22.00
## 54  0.00    208.96    134.33 28.36
## 55  0.00    179.10    134.33 29.85
## 56  93.33     53.33    146.67 12.00
## 57 230.00     55.00    110.00 16.00
## 58 200.00     35.00    110.00 21.00
## 59 190.00    230.00    140.00 15.00
## 60 200.00    110.00    100.00 16.00
## 61 333.33     80.00    146.67 28.00
## 62 140.00     25.00    110.00 13.00
## 63 343.28    171.64    149.25 25.37
## 64 200.00    110.00    100.00 17.00
## 65 266.67     80.00    146.67 21.33
```

```
#2.
corr_matrix <- cor(GreaterMeanFour)
corrplot(corr_matrix,method="shade")
```



```

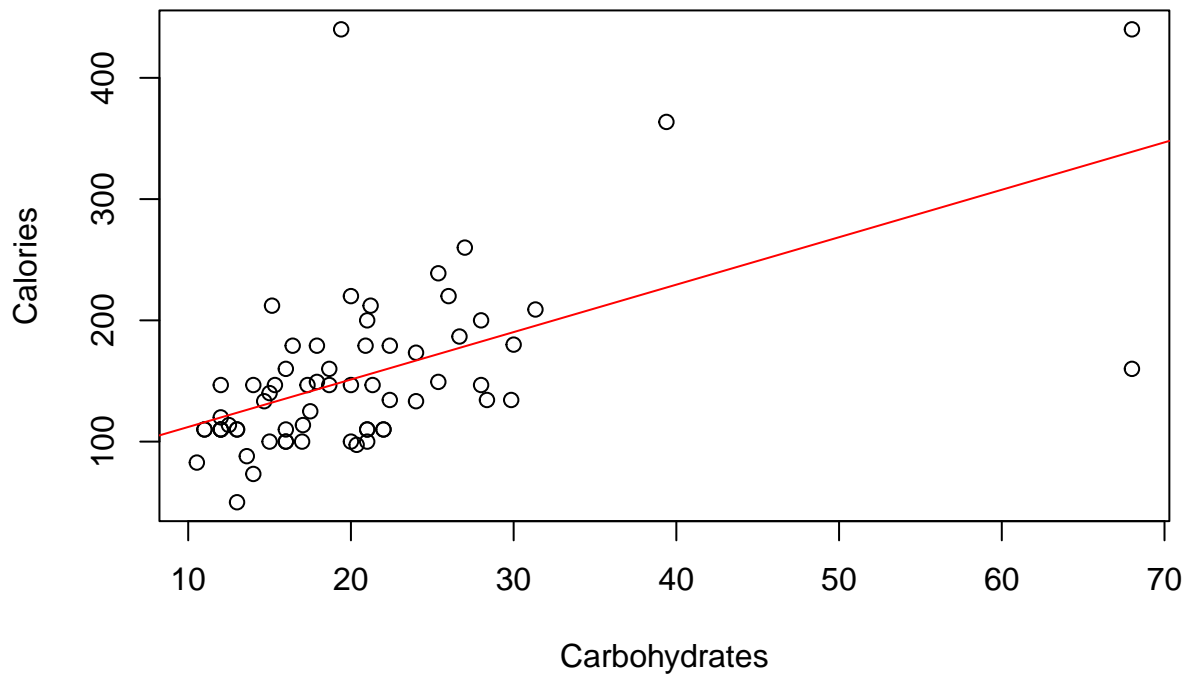
#3.
lm_model <- lm(calories ~ carbo, data = GreaterMeanFour)
summary(lm_model)

##
## Call:
## lm(formula = calories ~ carbo, data = GreaterMeanFour)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -179.002  -35.537   -8.202   18.560  291.160
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  72.9325    16.7957   4.342 5.22e-05 ***
## carbo         3.9128     0.7256   5.393 1.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.05 on 63 degrees of freedom
## Multiple R-squared:  0.3158, Adjusted R-squared:  0.305
## F-statistic: 29.08 on 1 and 63 DF,  p-value: 1.11e-06

plot(GreaterMeanFour$carbo, GreaterMeanFour$calories,
     main = "Simple Linear Regression",
     xlab = "Carbohydrates", ylab = "Calories",)
abline(lm_model, col = "red")

```

Simple Linear Regression



```
#4.
lm_model <- lm(calories ~ carbo, data = cereal)
lm_model

##
## Call:
## lm(formula = calories ~ carbo, data = cereal)
##
## Coefficients:
## (Intercept)      carbo
##      72.932      3.913

outliers <- which(abs(residuals(lm_model)) > 2 * sd(residuals(lm_model)))
calories_original <- cereal$calories
calories_no_outliers <- cereal$calories[-outliers]
t.test(calories_original, mu=cal_mean, paired=F, conf.level=0.95)

##
## One Sample t-test
##
## data:  calories_original
## t = 0.49997, df = 64, p-value = 0.6188
## alternative hypothesis: true mean is not equal to 149.6442
## 95 percent confidence interval:
##  136.2625 171.9600
## sample estimates:
## mean of x
```

```
## 154.1112
```

```
t.test(calories_no_outliers,mu=cal_mean,paired=F,conf.level=0.95)
```

```
##
```

```
## One Sample t-test
```

```
##
```

```
## data: calories_no_outliers
```

```
## t = -0.4947, df = 61, p-value = 0.6226
```

```
## alternative hypothesis: true mean is not equal to 149.6442
```

```
## 95 percent confidence interval:
```

```
## 131.3991 160.6522
```

```
## sample estimates:
```

```
## mean of x
```

```
## 146.0256
```

```
#R5 - Conclusion
```

```
#1. The Manufacturer N (Nabisco) has more preference to Fiber than any other manufacturer.(Boxplot-1)
```

```
#2. There might be defected products if the shelf life is very longer for a large content of calories.(
```

```
#3. There is a strong relationship between fiber and potassium, because if the content of fibre is incr
```

```
#4. The Content of Sodium is found to be very high in overall Cereals.(greaterMean)
```

```
#5. The manufacturer N (Nabisco) has more quantity of all the contents compared to other manufacturers.
```