exam.R

aaron

2024-01-25

```
#read data set
us_cereal <- read.csv("/Users/aaron/Downloads/UScereal1.csv")

#Understanding data set
head(us_cereal)
```

```
##                         Name mfr calories protein  fat sodium fibre carbo sugars
## 1                  100% Bran   N   212.12   12.12 3.03 393.94 30.30 15.15  18.18
## 2                   All-Bran   K   212.12   12.12 3.03 787.88 27.27 21.21  15.15
## 3 All-Bran with Extra Fiber   K   100.00    8.00 0.00 280.00 28.00 16.00   0.00
## 4   Apple Cinnamon Cheerios   G   146.67    2.67 2.67 240.00  2.00 14.00  13.33
## 5                Apple Jacks   K   110.00    2.00 0.00 125.00  1.00 11.00  14.00
## 6                    Basic 4   G   173.33    4.00 2.67 280.00  2.67 24.00  10.67
##   shelf potassium vitamins
## 1     3    848.48 enriched
## 2     3    969.70 enriched
## 3     3    660.00 enriched
## 4     1     93.33 enriched
## 5     2     30.00 enriched
## 6     3    133.33 enriched
```

```
# Max protein value of each manufacturer.
max_protein_by_manufacturer <- aggregate(protein ~ mfr, data = us_cereal, FUN = ma
x)
print(max_protein_by_manufacturer)
```

```
##   mfr protein
## 1   G    6.00
## 2   K   12.12
## 3   N   12.12
## 4   P   12.00
## 5   Q    8.00
## 6   R    4.48
```

```
# missing values.
missing_values <- sapply(us_cereal, function(x) sum(is.na(x)))
print(missing_values)
```

```
##      Name       mfr  calories   protein       fat    sodium     fibre     carbo
##         0         0         1         1         1         1         0         1
##    sugars     shelf potassium  vitamins
##         1         0         1         0
```

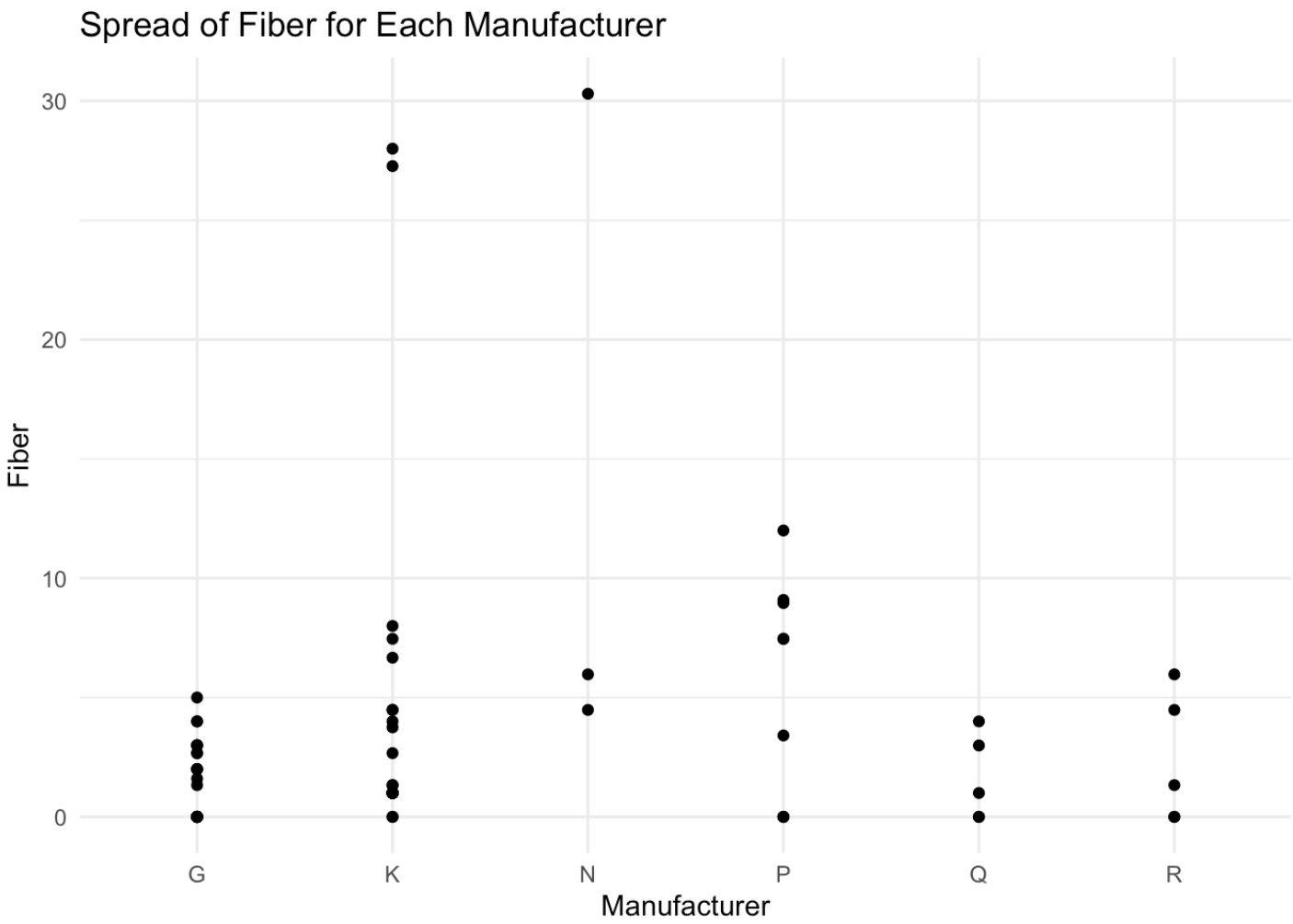
```
# Replacing missing values.
us_cereal$protein <- ifelse(is.na(us_cereal$protein), mean(us_cereal$protein, na.r
m = TRUE), us_cereal$protein)

# summary
summary_statistics <- summary(us_cereal[, c("calories", "protein", "fat", "sodiu
m", "fibre", "carbo", "sugars", "potassium")])
print(summary_statistics)
```

```
##     calories        protein            fat           sodium
##  Min.   : 50.0   Min.   : 0.750   Min.   :0.00   Min.   :  0.0
##  1st Qu.:110.0   1st Qu.: 2.000   1st Qu.:0.00   1st Qu.:180.0
##  Median :137.2   Median : 3.000   Median :1.00   Median :235.4
##  Mean   :149.6   Mean   : 3.726   Mean   :1.42   Mean   :238.6
##  3rd Qu.:179.1   3rd Qu.: 4.480   3rd Qu.:2.00   3rd Qu.:290.0
##  Max.   :440.0   Max.   :12.120   Max.   :9.09   Max.   :787.9
##  NA's   :1                        NA's   :1      NA's   :1
##      fibre            carbo           sugars        potassium
##  Min.   : 0.000   Min.   :10.53   Min.   : 0.00   Min.   : 15.00
##  1st Qu.: 0.000   1st Qu.:14.92   1st Qu.: 3.75   1st Qu.: 45.00
##  Median : 2.000   Median :18.67   Median :12.00   Median : 94.96
##  Mean   : 3.871   Mean   :20.01   Mean   :10.07   Mean   :158.69
##  3rd Qu.: 4.480   3rd Qu.:22.39   3rd Qu.:14.00   3rd Qu.:220.00
##  Max.   :30.300   Max.   :68.00   Max.   :20.90   Max.   :969.70
##                   NA's   :1       NA's   :1       NA's   :1
```

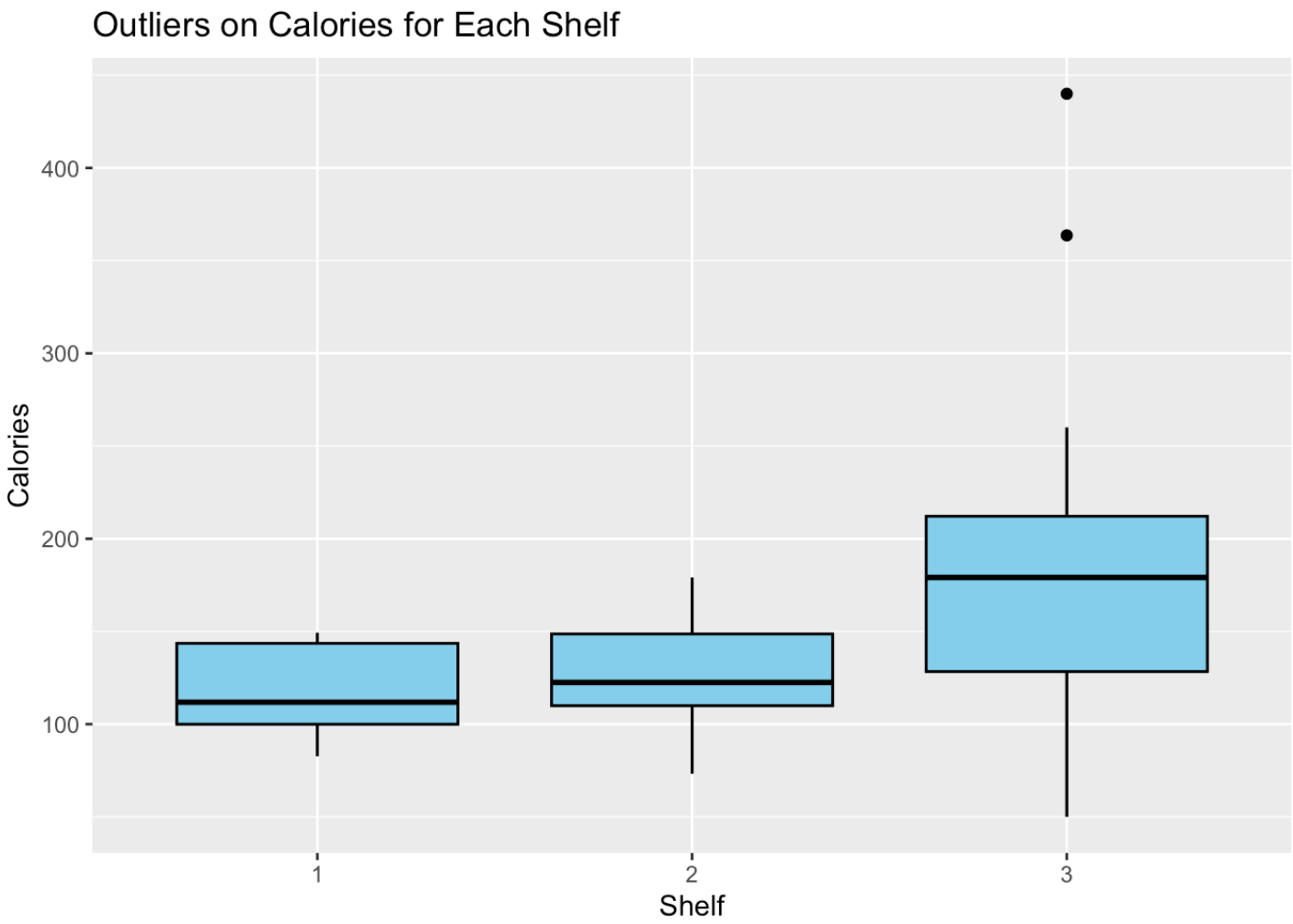
```
library(ggplot2)

# spread of Fiber for each Manufacturer
ggplot(us_cereal, aes(x = mfr, y = fibre)) + geom_point() + labs(title = "Spread o
f Fiber for Each Manufacturer", x = "Manufacturer", y = "Fiber") + theme_minimal()
```



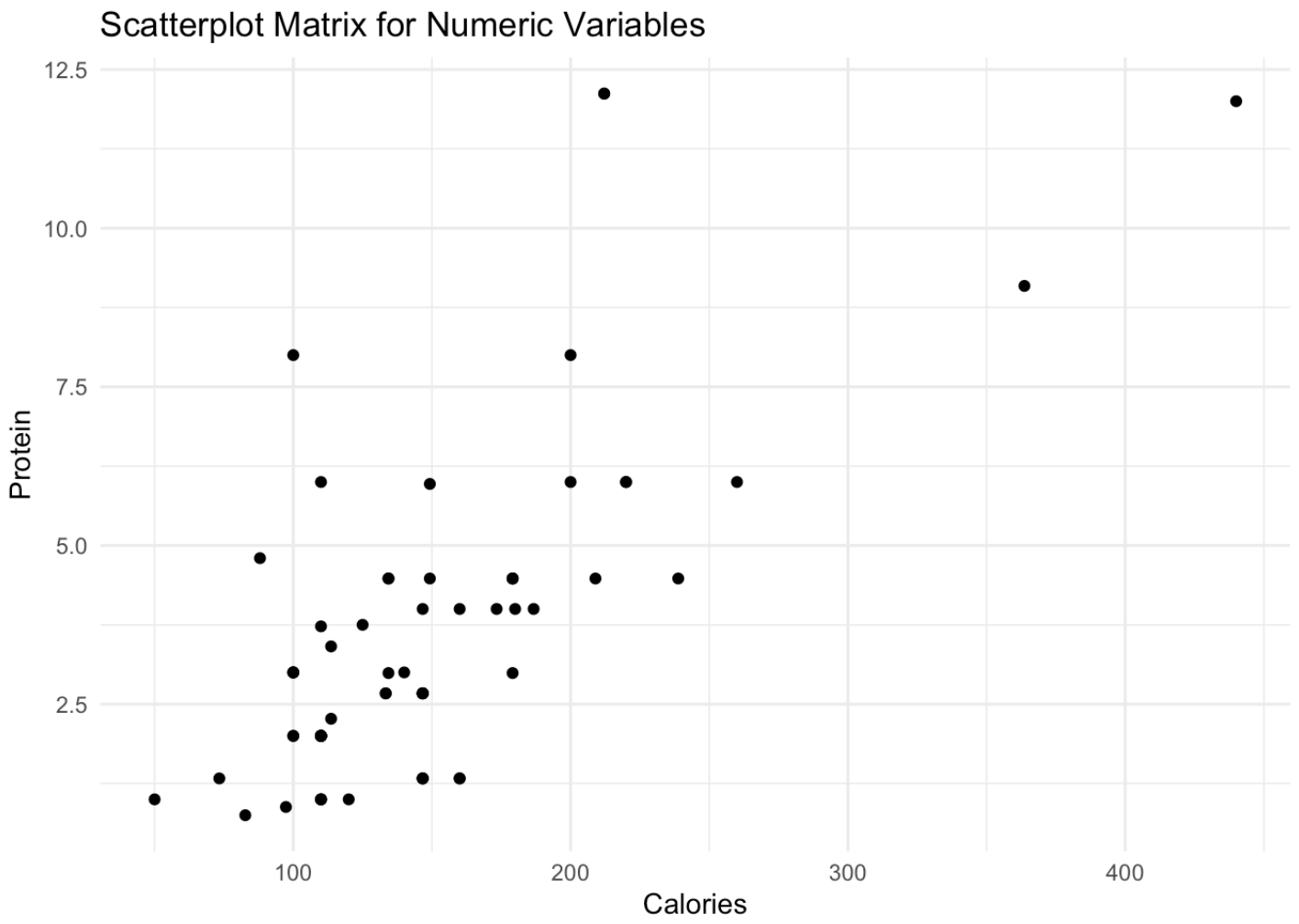

```
#outliers are
# outlines on calories for each shelf
ggplot(us_cereal, aes(x = as.factor(shelf), y = calories)) + geom_boxplot() + lab
s(title = "Outliers on Calories for Each Shelf", x = "Shelf", y = "Calories") + ge
om_boxplot(fill = "skyblue", color = "black")
```

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




```
# numeric variables
ggplot(us_cereal, aes(x = calories, y = protein)) + geom_point() + labs(title = "S
catterplot Matrix for Numeric Variables", x = "Calories", y = "Protein") + theme_m
inimal()
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```




```
# Identify the top-four mean variables
mean_values <- colMeans(us_cereal[, c("calories", "protein", "fat", "sodium", "fib
re", "carbo", "sugars", "potassium")], na.rm = TRUE)
top_four_mean_variables <- names(sort(mean_values, decreasing = TRUE)[1:4]);top_fo
ur_mean_variables
```

```
## [1] "sodium"    "potassium" "calories"  "carbo"
```

```
# Create the data frame GreaterMeanFour
GreaterMeanFour <- us_cereal[, c("Name", top_four_mean_variables)];GreaterMeanFour
```

```
##                                  Name sodium potassium calories carbo
## 1                           100% Bran 393.94    848.48   212.12 15.15
## 2                            All-Bran 787.88    969.70   212.12 21.21
## 3          All-Bran with Extra Fiber 280.00    660.00   100.00 16.00
## 4            Apple Cinnamon Cheerios 240.00     93.33   146.67 14.00
## 5                        Apple Jacks 125.00     30.00   110.00 11.00
## 6                            Basic 4 280.00    133.33   173.33 24.00
## 7                          Bran Chex 298.51        NA   134.33 22.39
## 8                        Bran Flakes 313.43    283.58       NA 19.40
## 9                       Cap'n'Crunch 293.33     46.67   160.00 16.00
## 10                          Cheerios 232.00     84.00    88.00 13.60
## 11           Cinnamon Toast Crunch 280.00     60.00   160.00    NA
## 12                          Clusters 280.00    210.00   220.00 26.00
```

```
## 13                            Cocoa Puffs 180.00     55.00   110.00 12.00
## 14                              Corn Chex 280.00     25.00   110.00 22.00
## 15                            Corn Flakes 290.00     35.00   100.00 21.00
## 16                              Corn Pops  90.00     20.00   110.00 13.00
## 17                           Count Chocula 180.00     65.00   110.00 12.00
## 18                     Cracklin' Oat Bran 280.00    320.00   220.00 20.00
## 19                                Crispix 220.00     30.00   110.00 21.00
## 20                  Crispy Wheat & Raisins     NA    160.00   133.33 14.67
## 21                            Double Chex 253.33    106.67   133.33 24.00
## 22                            Froot Loops 125.00     30.00   110.00 11.00
## 23                          Frosted Flakes 266.67     33.33   146.67 18.67
## 24                     Frosted Mini-Wheats   0.00    125.00   125.00 17.50
## 25 Fruit & Fibre: Dates Walnuts and Oats 238.81    298.51   179.10 17.91
## 26                           Fruitful Bran 358.21    283.58   179.10 20.90
## 27                          Fruity Pebbles 180.00     33.33   146.67 17.33
## 28                            Golden Crisp  51.14     45.45   113.64 12.50
## 29                          Golden Grahams 373.33     60.00   146.67 20.00
## 30                       Grape Nuts Flakes 159.09     96.59   113.64 17.05
## 31                               Grape-Nuts 680.00    360.00   440.00 68.00
## 32                     Great Grains Pecan 227.27    303.03   363.64 39.39
## 33                       Honey Graham Ohs 220.00     45.00   120.00 12.00
## 34                     Honey Nut Cheerios 333.33    120.00   146.67 15.33
## 35                              Honey-comb 135.34     26.32    82.71 10.53
## 36                  Just Right Fruit & Nut 226.67    126.67   186.67 26.67
## 37                                     Kix 173.33     26.67    73.33 14.00
## 38                                    Life 223.88    141.79   149.25 17.91
## 39                           Lucky Charms 180.00     55.00   110.00 12.00
## 40                     Mueslix Crispy Blend 223.88    238.81   238.81 25.37
## 41                     Multi-Grain Cheerios 220.00     90.00   100.00 15.00
## 42                         Nut&Honey Crunch 283.58     59.70   179.10 22.39
## 43               Nutri-Grain Almond-Raisin 328.36    194.03   208.96 31.34
## 44                    Oatmeal Raisin Crisp 340.00    240.00   260.00 27.00
## 45                  Post Nat. Raisin Bran 298.51    388.06   179.10 16.42
## 46                               Product 19 320.00     45.00   100.00 20.00
## 47                            Puffed Rice   0.00     15.00    50.00 13.00
## 48                     Quaker Oat Squares 270.00    220.00   200.00 28.00
## 49                            Raisin Bran 280.00    320.00   160.00 18.67
## 50                        Raisin Nut Bran 280.00    280.00   200.00 21.00
## 51                         Raisin Squares   0.00    220.00   180.00 30.00
## 52                              Rice Chex 212.39     26.55    97.35 20.35
## 53                           Rice Krispies 290.00     35.00   110.00 22.00
## 54                  Shredded Wheat 'n'Bran   0.00    208.96   134.33 28.36
## 55               Shredded Wheat spoon size   0.00    179.10   134.33 29.85
## 56                                  Smacks  93.33     53.33   146.67 12.00
## 57                              Special K 230.00     55.00   110.00 16.00
## 58                       Total Corn Flakes 200.00     35.00   110.00 21.00
## 59                       Total Raisin Bran 190.00    230.00   140.00 15.00
## 60                       Total Whole Grain 200.00    110.00   100.00 16.00
## 61                                Triples 333.33     80.00   146.67 28.00
## 62                                   Trix 140.00     25.00   110.00 13.00
## 63                              Wheat Chex 343.28    171.64   149.25 25.37
## 64                                Wheaties 200.00    110.00   100.00 17.00
## 65                     Wheaties Honey Gold 266.67     80.00   146.67 21.33
```

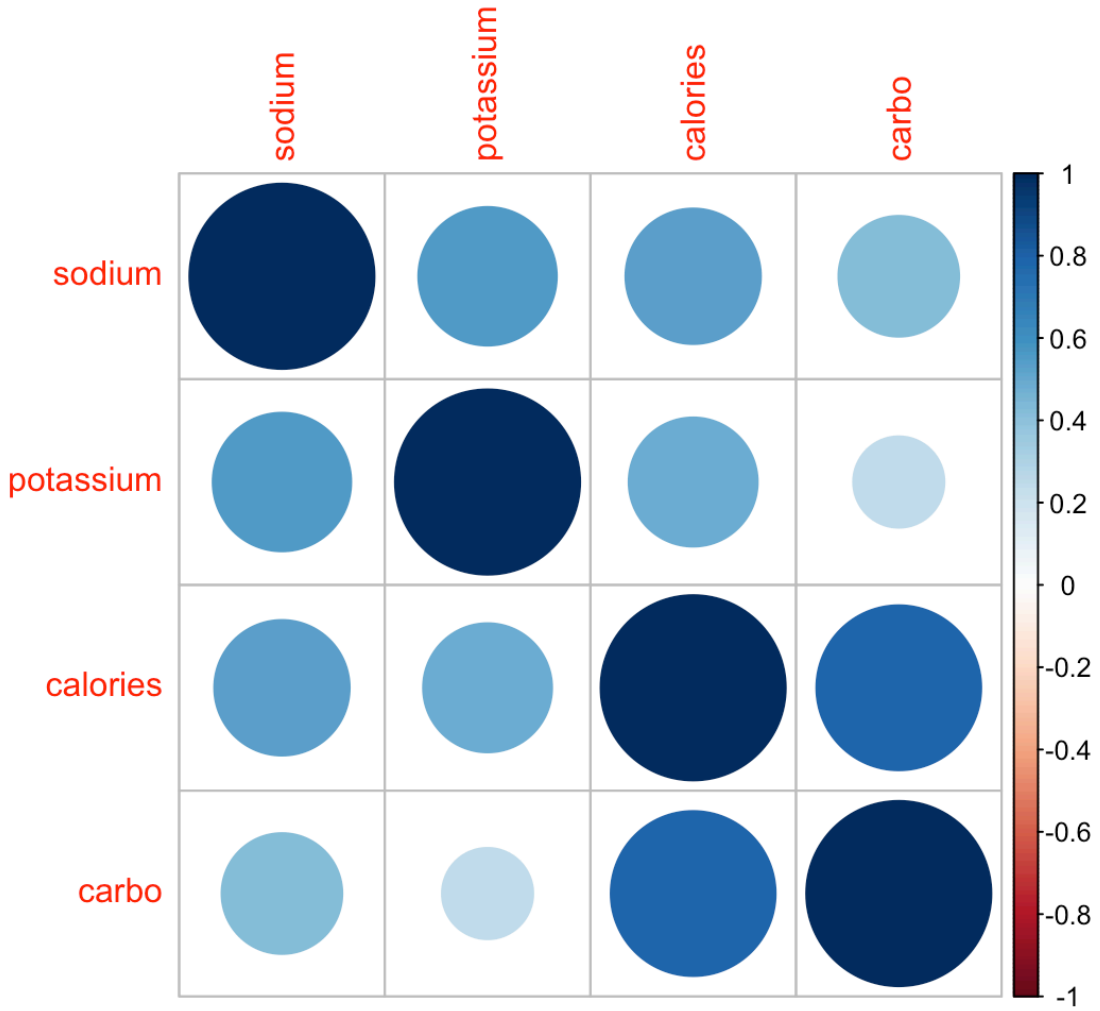
```
# Correlation matrix
correlation_matrix <- cor(us_cereal[, top_four_mean_variables], use = "complete.obs");correlation_matrix
```

```
##                sodium potassium  calories     carbo
## sodium     1.0000000 0.5591352 0.5346509 0.4246589
## potassium 0.5591352 1.0000000 0.4851627 0.2421264
## calories   0.5346509 0.4851627 1.0000000 0.7926986
## carbo      0.4246589 0.2421264 0.7926986 1.0000000
```

```
# Plot the correlation matrix
library(corrplot)
```

```
## corrplot 0.92 loaded
```

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



```
 # calories to predict protein
model <- lm(protein ~ calories, data = us_cereal)

# Plot the linear regression line
ggplot(us_cereal, aes(x = calories, y = protein)) + geom_point() + geom_smooth(met
hod = "lm", se = FALSE, color = "red") + labs(title = "Simple Linear Regression: P
rotein vs. Calories", x = "Calories", y = "Protein") + theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

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

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```


```
# Predictions before removing outliers
predictions_before <- predict(model, newdata = data.frame(calories = 150));predict
ions_before
```

```
##        1
## 3.724665
```

```
# remove outliers
outliers <- which(model$residuals > quantile(model$residuals, 0.975) | model$resid
uals < quantile(model$residuals, 0.025))
us_cereal_no_outliers <- us_cereal[-outliers, ]

# Create a new model without outliers
model_no_outliers <- lm(protein ~ calories, data = us_cereal_no_outliers);model_no
_outliers
```

```
##
## Call:
## lm(formula = protein ~ calories, data = us_cereal_no_outliers)
##
## Coefficients:
## (Intercept)     calories
##    -0.55752      0.02677
```

```
# Predictions after removing outliers
predictions_after <- predict(model_no_outliers, newdata = data.frame(calories = 15
0))

# Show predictions
print(paste("Prediction Before Removing Outliers:", predictions_before))
```

```
## [1] "Prediction Before Removing Outliers: 3.72466465502304"
```

```
print(paste("Prediction After Removing Outliers:", predictions_after))
```

```
## [1] "Prediction After Removing Outliers: 3.45806071040708"
```