Untitled.R

amanpanchal

2024-09-26

```
options(repos = c(CRAN = "https://cloud.r-project.org"))

# Install necessary packages
install.packages(c("readr", "ggplot2", "cluster", "factoextra"))
```

```
##
## The downloaded binary packages are in
##  /var/folders/5b/bllk4vpx1w30s3bf3j16c5p00000gn/T//RtmpnsDXqh/downloaded_packages
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
getwd()
```

```
## [1] "/Users/amanpanchal/Downloads"
```

```
# Import the dataset
cereals <- read.csv("Cereals nutritional data.csv", stringsAsFactors = FALSE)
# View the first few rows of the dataset
head(cereals)
```

```
##                        name mfr type calories protein fat sodium fiber carbo
## 1                 100% Bran   N    C       70       4   1    130  10.0   5.0
## 2         100% Natural Bran   Q    C      120       3   5     15   2.0   8.0
## 3                   All-Bran   K    C       70       4   1    260   9.0   7.0
## 4 All-Bran with Extra Fiber   K    C       50       4   0    140  14.0   8.0
## 5             Almond Delight   R    C      110       2   2    200   1.0  14.0
## 6   Apple Cinnamon Cheerios   G    C      110       2   2    180   1.5  10.5
```

```
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 2      8    135        0     3      1 1.00 33.98368
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
## 5      8     -1       25     3      1 0.75 34.38484
## 6     10     70       25     1      1 0.75 29.50954
```

```
# Structure of the dataset
str(cereals)
```

```
## 'data.frame':    77 obs. of  16 variables:
##  $ name    : chr  "100% Bran" "100% Natural Bran" "All-Bran" "All-Bran with Extra Fiber" ...
##  $ mfr     : chr  "N" "Q" "K" "K" ...
##  $ type    : chr  "C" "C" "C" "C" ...
##  $ calories: int  70 120 70 50 110 110 110 130 90 90 ...
##  $ protein : int  4 3 4 4 2 2 2 3 2 3 ...
##  $ fat     : int  1 5 1 0 2 2 0 2 1 0 ...
##  $ sodium  : int  130 15 260 140 200 180 125 210 200 210 ...
##  $ fiber   : num  10 2 9 14 1 1.5 1 2 4 5 ...
##  $ carbo   : num  5 8 7 8 14 10.5 11 18 15 13 ...
##  $ sugars  : int  6 8 5 0 8 10 14 8 6 5 ...
##  $ potass  : int  280 135 320 330 -1 70 30 100 125 190 ...
##  $ vitamins: int  25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : int  3 3 3 3 3 1 2 3 1 3 ...
##  $ weight  : num  1 1 1 1 1 1 1 1.33 1 1 ...
##  $ cups    : num  0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
##  $ rating  : num  68.4 34 59.4 93.7 34.4 ...
```

```
# Summary statistics
summary(cereals)
```

```
##      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.   :-1.0   Min.   :-1.000   Min.   : -1.00   Min.   :  0.00  
##  1st Qu.:12.0   1st Qu.: 3.000   1st Qu.: 40.00   1st Qu.: 25.00  
##  Median :14.0   Median : 7.000   Median : 90.00   Median : 25.00  
##  Mean   :14.6   Mean   : 6.922   Mean   : 96.08   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
```

```
##      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
```

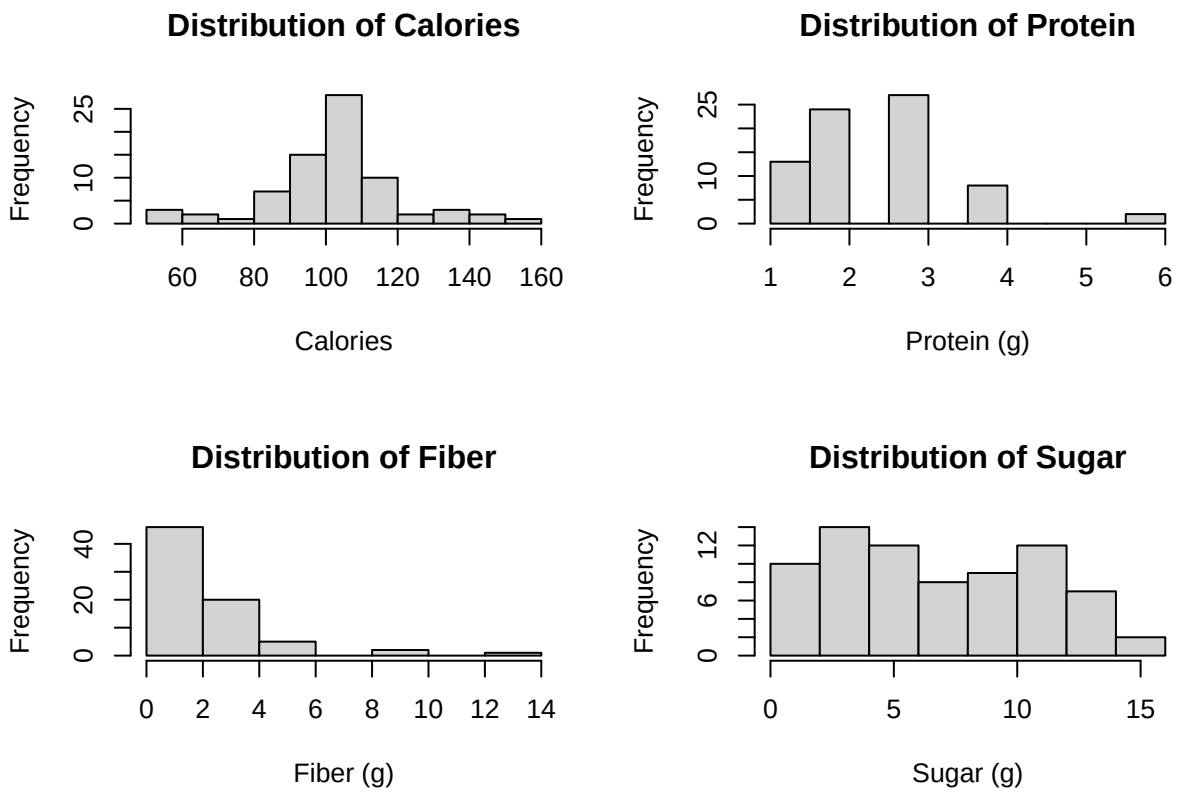
```
# Remove rows with negative values
cereals_clean <- cereals %>%
  filter_all(all_vars(. >= 0))
# Check for any remaining NA values
sum(is.na(cereals_clean))
```

```
## [1] 0
```

```
# Convert manufacturer and type to factors
cereals_clean$mfr <- as.factor(cereals_clean$mfr)
cereals_clean$type <- as.factor(cereals_clean$type)
# Display summary of cleaned data
summary(cereals_clean)
```

```
##      name               mfr    type      calories      protein           fat
##  Length:74          A: 1   C:73   Min.   : 50   Min.   :1.000   Min.   :0
##  Class :character   G:22   H: 1   1st Qu.:100   1st Qu.:2.000   1st Qu.:0
##  Mode  :character   K:23          Median :110   Median :2.500   Median :1
##                     N: 5          Mean   :107   Mean   :2.514   Mean   :1
##                     P: 9          3rd Qu.:110   3rd Qu.:3.000   3rd Qu.:1
##                     Q: 7          Max.   :160   Max.   :6.000   Max.   :5
##                     R: 7
##      sodium          fiber            carbo           sugars
##  Min.   :  0.0   Min.   : 0.000   Min.   : 5.00   Min.   : 0.000
##  1st Qu.:135.0   1st Qu.: 0.250   1st Qu.:12.00   1st Qu.: 3.000
##  Median :180.0   Median : 2.000   Median :14.50   Median : 7.000
##  Mean   :162.4   Mean   : 2.176   Mean   :14.73   Mean   : 7.108
##  3rd Qu.:217.5   3rd Qu.: 3.000   3rd Qu.:17.00   3rd Qu.:11.000
##  Max.   :320.0   Max.   :14.000   Max.   :23.00   Max.   :15.000
##
##      potass          vitamins          shelf           weight
##  Min.   : 15.00   Min.   :  0.00   Min.   :1.000   Min.   :0.500
##  1st Qu.: 41.25   1st Qu.: 25.00   1st Qu.:1.250   1st Qu.:1.000
##  Median : 90.00   Median : 25.00   Median :2.000   Median :1.000
##  Mean   : 98.51   Mean   : 29.05   Mean   :2.216   Mean   :1.031
##  3rd Qu.:120.00   3rd Qu.: 25.00   3rd Qu.:3.000   3rd Qu.:1.000
##  Max.   :330.00   Max.   :100.00   Max.   :3.000   Max.   :1.500
##
##       cups            rating
##  Min.   :0.2500   Min.   :18.04
##  1st Qu.:0.6700   1st Qu.:32.45
##  Median :0.7500   Median :40.25
##  Mean   :0.8216   Mean   :42.37
##  3rd Qu.:1.0000   3rd Qu.:50.52
##  Max.   :1.5000   Max.   :93.70
##
```

```r
par(mfrow=c(2,2))
hist(cereals_clean$calories, main="Distribution of Calories", xlab="Calories")
hist(cereals_clean$protein, main="Distribution of Protein", xlab="Protein (g)")
hist(cereals_clean$fiber, main="Distribution of Fiber", xlab="Fiber (g)")
hist(cereals_clean$sugars, main="Distribution of Sugar", xlab="Sugar (g)")
```




```r
plot(cereals_clean$sugars, cereals_clean$rating,
     main="Sugar Content vs Rating",
     xlab="Sugar Content (g)", ylab="Rating")
abline(lm(rating ~ sugars, data=cereals_clean), col="red")

nutritional_elements <- cereals_clean %>%
  select(calories, protein, fiber, sugars)

nutritional_long <- nutritional_elements %>%
  gather(key = "nutrient", value = "amount")

ggplot(nutritional_long, aes(x = amount)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ nutrient, scales = "free") +
  theme_minimal() +
  labs(title = "Distribution of Key Nutritional Elements in Cereals",
       x = "Amount", y = "Count")

# Correlation test
```

```r
cor.test(cereals_clean$sugars, cereals_clean$rating)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cereals_clean$sugars and cereals_clean$rating
## t = -9.7987, df = 72, p-value = 6.924e-15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8394514 -0.6375904
## sample estimates:
##       cor
## -0.7559551
```

```r
cor_test <- cor.test(cereals_clean$sugars, cereals_clean$rating)
print(cor_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cereals_clean$sugars and cereals_clean$rating
## t = -9.7987, df = 72, p-value = 6.924e-15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8394514 -0.6375904
## sample estimates:
##       cor
## -0.7559551
```

```r
boxplot(calories ~ mfr, data=cereals_clean,
        main="Calorie Content by Manufacturer",
        xlab="Manufacturer", ylab="Calories")
aggregate(calories ~ mfr, data=cereals_clean, mean)
```

```
##   mfr  calories
## 1   A 100.00000
## 2   G 111.36364
## 3   K 108.69565
## 4   N  84.00000
## 5   P 108.88889
## 6   Q  94.28571
## 7   R 115.71429
```

```r
anova_result <- aov(calories ~ mfr, data=cereals_clean)
summary(anova_result)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## mfr          6   4874   812.4    2.28 0.0461 *
## Residuals   67  23872   356.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Post-hoc test (if ANOVA is significant)
if(summary(anova_result)[[1]]$`Pr(>F)`[1] < 0.05) {
  TukeyHSD(anova_result)
}
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = calories ~ mfr, data = cereals_clean)
##
## $mfr
##            diff        lwr       upr     p adj
## G-A  11.3636364 -47.302531 70.029804 0.9969446
## K-A   8.6956522 -49.915039 67.306344 0.9993197
## N-A -16.0000000 -78.852964 46.852964 0.9867043
## P-A   8.8888889 -51.591404 69.369182 0.9993554
## Q-A  -5.7142857 -67.052498 55.623927 0.9999546
## R-A  15.7142857 -45.623927 77.052498 0.9862558
## K-G  -2.6679842 -19.778620 14.442651 0.9990965
## N-G -27.3636364 -55.789959  1.062686 0.0667104
## P-G  -2.4747475 -25.177755 20.228260 0.9998858
## Q-G -17.0779221 -41.976456  7.820612 0.3733926
## R-G   4.3506494 -20.547885 29.249184 0.9982796
## N-K -24.6956522 -53.007306  3.616002 0.1273075
## P-K   0.1932367 -22.366029 22.752502 1.0000000
## Q-K -14.4099379 -39.177475 10.357600 0.5734714
## R-K   7.0186335 -17.748904 31.786171 0.9769721
## P-N  24.8888889  -7.114274 56.892052 0.2302293
## Q-N  10.2857143 -23.310608 43.882037 0.9662169
## R-N  31.7142857  -1.882037 65.310608 0.0766817
## Q-P -14.6031746 -43.518285 14.311936 0.7228284
## R-P   6.8253968 -22.089714 35.740507 0.9910630
## R-Q  21.4285714  -9.240535 52.097678 0.3510519
```

```r
install.packages(c("cluster", "factoextra"))
```

```
##
## The downloaded binary packages are in
##  /var/folders/5b/bllk4vpx1w30s3bf3j16c5p00000gn/T//RtmpnsDXqh/downloaded_packages
```

```r
# Load required libraries
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
# Assuming you've already performed the k-means clustering and have km_result and scaled_data

if (!requireNamespace("cluster", quietly = TRUE)) install.packages("cluster")
if (!requireNamespace("factoextra", quietly = TRUE)) install.packages("factoextra")
library(cluster)
```

```r
library(factoextra)

# Prepare data for clustering
cluster_data <- cereals_clean[, c("calories", "protein", "fat", "sodium", "fiber", "carbo", "sugars")]
# Scale the data
scaled_data <- scale(cluster_data)

# Determine optimal number of clusters
fviz_nbclust(scaled_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal k")
# Perform k-means clustering
set.seed(123)
km_result <- kmeans(scaled_data, centers = 3, nstart = 25)
# Visualize clusters
fviz_cluster(km_result, data = scaled_data,
             geom = "point",
             ellipse.type = "convex",
             palette = "jco",
             ggtheme = theme_minimal())
#
```