Week 1 - Homework

STAT 420, Fall 2024, Banghao Chi

09/08/2024

Exercise 1 (Subsetting and Statistics)

For this exercise, we will use the `msleep` dataset from the `ggplot2` package.

(a) Install and load the `ggplot2` package. **Do not** include the installation command in your `.Rmd` file. (If you do it will install the package every time you knit your file.) **Do** include the command to load the package into your environment.

Solution:

```r
library(ggplot2)
```

(b) Note that this dataset is technically a `tibble`, not a data frame. How many observations are in this dataset? How many variables? What are the observations in this dataset?

Solution:

```r
dataset = ggplot2::msleep
observations = nrow(dataset)
observations
```

```
## [1] 83
```

```r
variables = ncol(dataset)
variables
```

```
## [1] 11
```

```r
head(dataset, 3)
```

```
## # A tibble: 3 x 11
##   name    genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <chr>   <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Cheetah Acin~ carni Carn~ lc                  12.1      NA            NA  11.9
## 2 Owl mo~ Aotus omni  Prim~ <NA>                17         1.8          NA   7
## 3 Mounta~ Aplo~ herbi Rode~ nt                  14.4       2.4          NA   9.6
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

There are **83** observations and **11** variables. From the head function, we can see that the observations in the **`ggplot2::msleep`** dataset are **different species of animals** with their relevant information such as genus, vore, sleeping information, brain weight and body weight.

(c) What is the mean hours of REM sleep of individuals in this dataset?

Solution:

```r
mean_rem_sleep = mean(dataset$sleep_rem, na.rm = TRUE)
mean_rem_sleep
```

```
## [1] 1.87541
```

The mean hours of REM sleep of individuals in this dataset is **1.8754098**

(d) What is the standard deviation of brain weight of individuals in this dataset?

Solution:

```
sd_brainwt = sd(dataset$brainwt, na.rm = TRUE)
sd_brainwt
```

```
## [1] 0.9764137
```

The standard deviation of brain weight of individuals in this dataset is **0.9764137** kg.

(e) Which observation (provide the `name`) in this dataset gets the most REM sleep?

Solution:

```
name_of_max_sleep_rem = dataset[which.max(dataset$sleep_rem), ]$name
name_of_max_sleep_rem
```

```
## [1] "Thick-tailed opposum"
```

Thick-tailed opposum in this dataset gets the most REM sleep.

(f) What is the average bodyweight of carnivores in this dataset?

Solution:

```
carnivores_avg_bodywt = mean(dataset$bodywt[dataset$vore == "carni"], na.rm = TRUE)
carnivores_avg_bodywt
```

```
## [1] 90.75111
```

The average bodyweight of carnivores in this dataset is **90.7511053** kg.

Exercise 2 (Plotting)

For this exercise, we will use the `birthwt` dataset from the `MASS` package.

(a) Note that this dataset is a data frame and all of the variables are numeric. How many observations are in this dataset? How many variables? What are the observations in this dataset?

Solution:

```
library(MASS)
dataset = MASS::birthwt
observations = nrow(dataset)
observations
```

```
## [1] 189
```

```
variables = ncol(dataset)
variables
```

```
## [1] 10
```

```
head(dataset, 3)
```

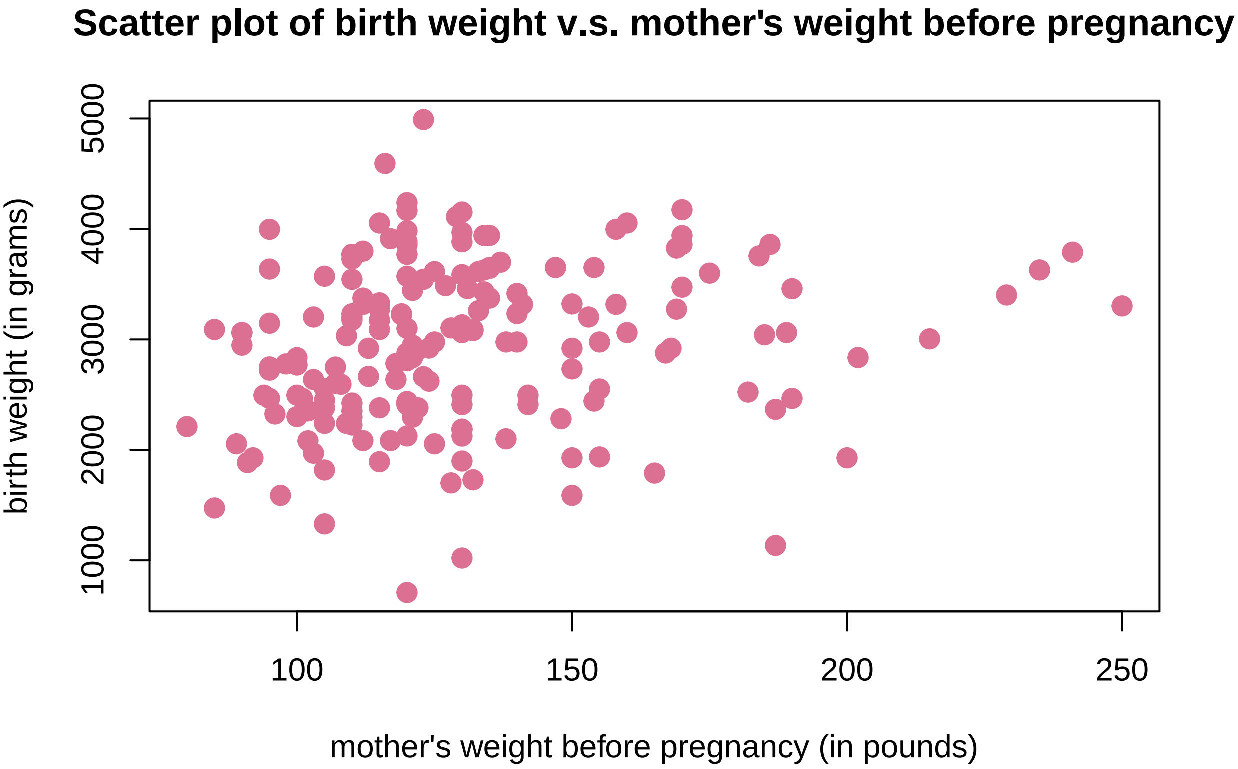
```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
```

There are **189** observations and **10** variables. Each row is a single **birth case** with their mothers' information such as age, weight, race, smoking habit etc, and finally the baby's birth weight.

(b) Create a scatter plot of birth weight (y-axis) vs mother's weight before pregnancy (x-axis). Use a non-default color for the points. (Also, be sure to give the plot a title and label the axes appropriately.) Based on the scatter plot, does there seem to be a relationship between the two variables? Briefly explain.

Solution:

```r
plot( bwt ~ lwt, data = dataset,
  main = "Scatter plot of birth weight v.s. mother's weight before pregnancy",
  xlab = "mother's weight before pregnancy (in pounds)",
  ylab = "birth weight (in grams)",
  pch = 20,
  cex = 2,
  col = "palevioletred"
)
```

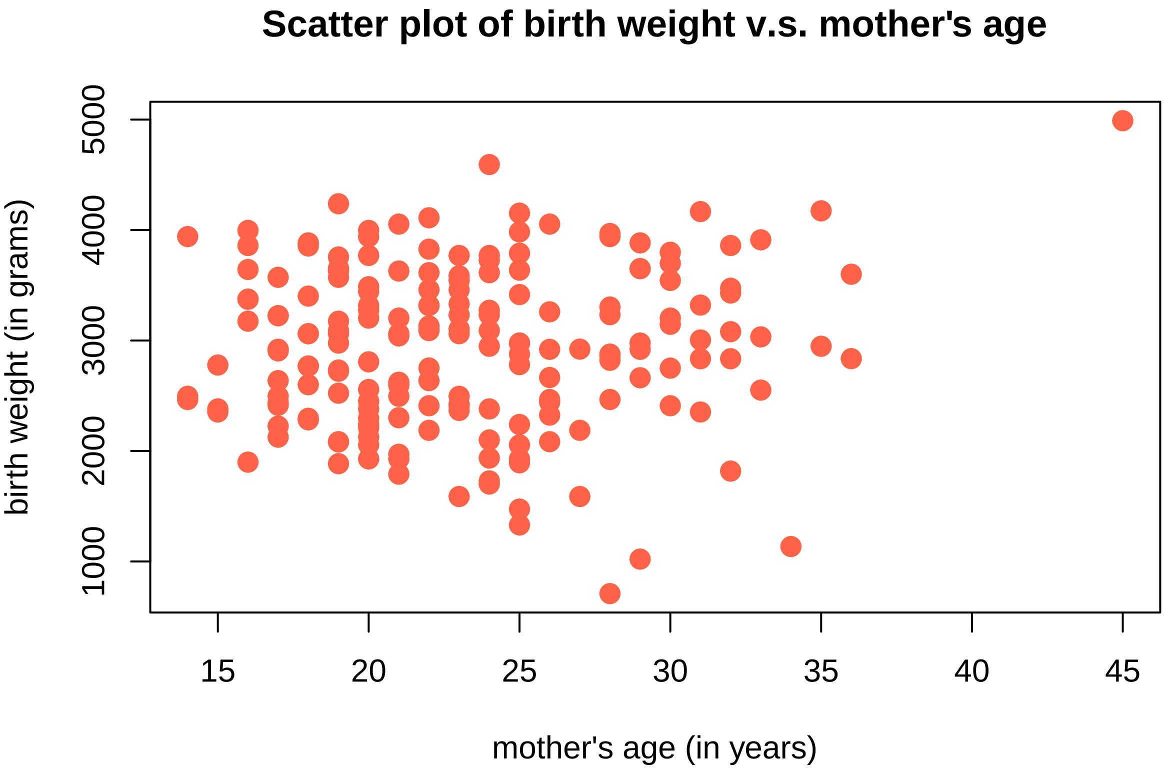


Yes, it seems that there's some relationships between the two variables. The points gather together mostly along the line that crosses (2000, 100) and (3000, 125), with some other points surronding it, indicating that generally if the baby's mother's weight is greater, the heavier the baby itself will be.

(c) Create a scatter plot of birth weight (y-axis) vs mother's age (x-axis). Use a non-default color for the points. (Also, be sure to give the plot a title and label the axes appropriately.) Based on the scatter plot, does there seem to be a relationship between the two variables? Briefly explain.

Solution:

```r
plot(bwt ~ age, data = dataset,
  main = "Scatter plot of birth weight v.s. mother's age",
  xlab = "mother's age (in years)",
  ylab = "birth weight (in grams)",
  pch = 20,
  cex = 2,
  col = "tomato"
)
```

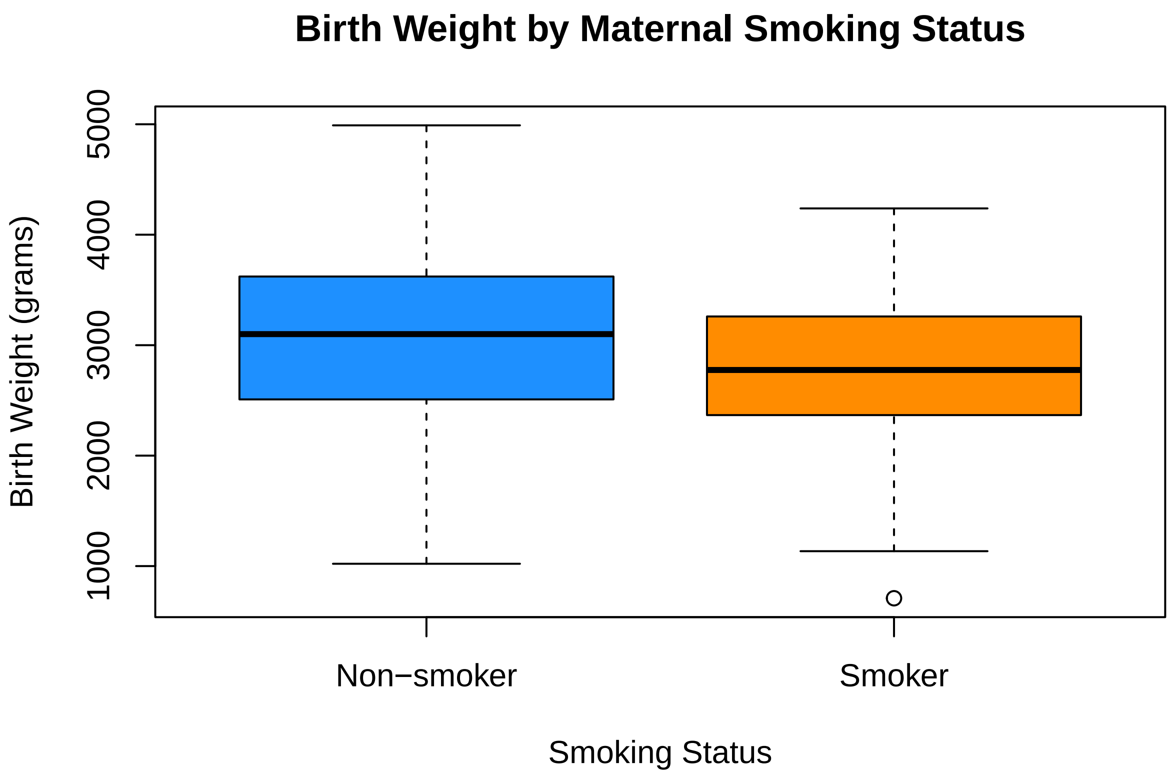


No, from the plot, these two variables don't seem to have a relationship, since the points are scattered around evenly around the plot.

(d) Create side-by-side boxplots for birth weight grouped by smoking status. Use non-default colors for the plot. (Also, be sure to give the plot a title and label the axes appropriately.) Based on the boxplot, does there seem to be a difference in birth weight for mothers who smoked? Briefly explain.

Solution:

```
boxplot(bwt ~ smoke, data = dataset,
        main = "Birth Weight by Maternal Smoking Status",
        xlab = "Smoking Status",
        ylab = "Birth Weight (grams)",
        col = c("dodgerblue", "darkorange"),
        names = c("Non-smoker", "Smoker"))
```



Based on the boxplot, there seems to exist a difference in birth weight for mothers who smoked and who don't smoke. From the graph, we can see that the baby's birth weight, whose mother doesn't smoke, are less than the baby's birth weight whose mother smoked by looking at both the box and also the median line.

Exercise 3 (Importing Data, More Plotting)

For this exercise we will use the data stored in `nutrition-2018.csv`. It contains the nutritional values per serving size for a large variety of foods as calculated by the USDA in 2018. It is a cleaned version totaling 5956 observations and is current as of April 2018.

The variables in the dataset are:

- `ID`
- `Desc` - short description of food
- `Water` - in grams
- `Calories` - in kcal
- `Protein` - in grams
- `Fat` - in grams
- `Carbs` - carbohydrates, in grams
- `Fiber` - in grams
- `Sugar` - in grams
- `Calcium` - in milligrams
- `Potassium` - in milligrams
- `Sodium` - in milligrams
- `VitaminC` - vitamin C, in milligrams
- `Chol` - cholesterol, in milligrams
- `Portion` - description of standard serving size used in analysis

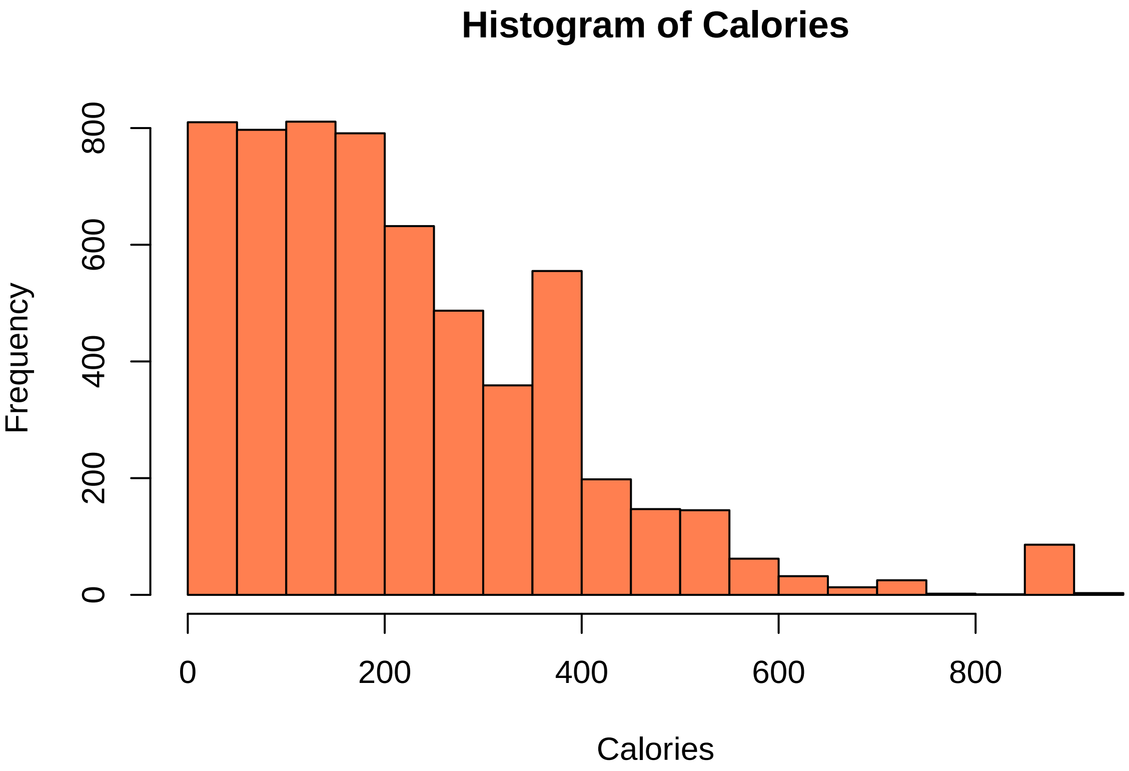
```
dataset = read.csv("nutrition-2018.csv")
str(dataset)
```

```
## 'data.frame':    5956 obs. of  15 variables:
##  $ ID       : int  1001 1002 1003 1004 1005 1006 1007 1009 1011 1012 ...
##  $ Desc     : chr  "BUTTER,WITH SALT" "BUTTER,WHIPPED,W/ SALT" "BUTTER OIL,ANHYDROUS" "CHEESE,BLUE"
##  $ Water    : num  15.87 16.72 0.24 42.41 41.11 ...
##  $ Calories : int  717 718 876 353 371 334 300 404 394 98 ...
##  $ Protein  : num  0.85 0.49 0.28 21.4 23.24 ...
##  $ Fat      : num  81.1 78.3 99.5 28.7 29.7 ...
##  $ Carbs    : num  0.06 2.87 0 2.34 2.79 0.45 0.46 3.09 2.57 3.38 ...
##  $ Fiber    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sugar    : num  0.06 0.06 0 0.5 0.51 0.45 0.46 0.48 0.52 2.67 ...
##  $ Calcium  : int  24 23 4 528 674 184 388 710 685 83 ...
##  $ Potassium: int  24 41 5 256 136 152 187 76 127 104 ...
##  $ Sodium   : int  643 583 2 1146 560 629 842 653 604 364 ...
##  $ VitaminC : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chol     : int  215 225 256 75 94 100 72 99 95 17 ...
##  $ Portion  : chr  "1 pat,  (1\" sq, 1/3\" high)" "1 pat,  (1\" sq, 1/3\" high)" "1 tbsp" "1 oz" ...
```

(a) Create a histogram of `Calories`. Do not modify `R`'s default bin selection. Make the plot presentable. Describe the shape of the histogram. Do you notice anything unusual?

Solution:

```
hist(
  dataset$Calories,
  main = "Histogram of Calories",
  col = "coral",
  xlab = "Calories"
  )
```

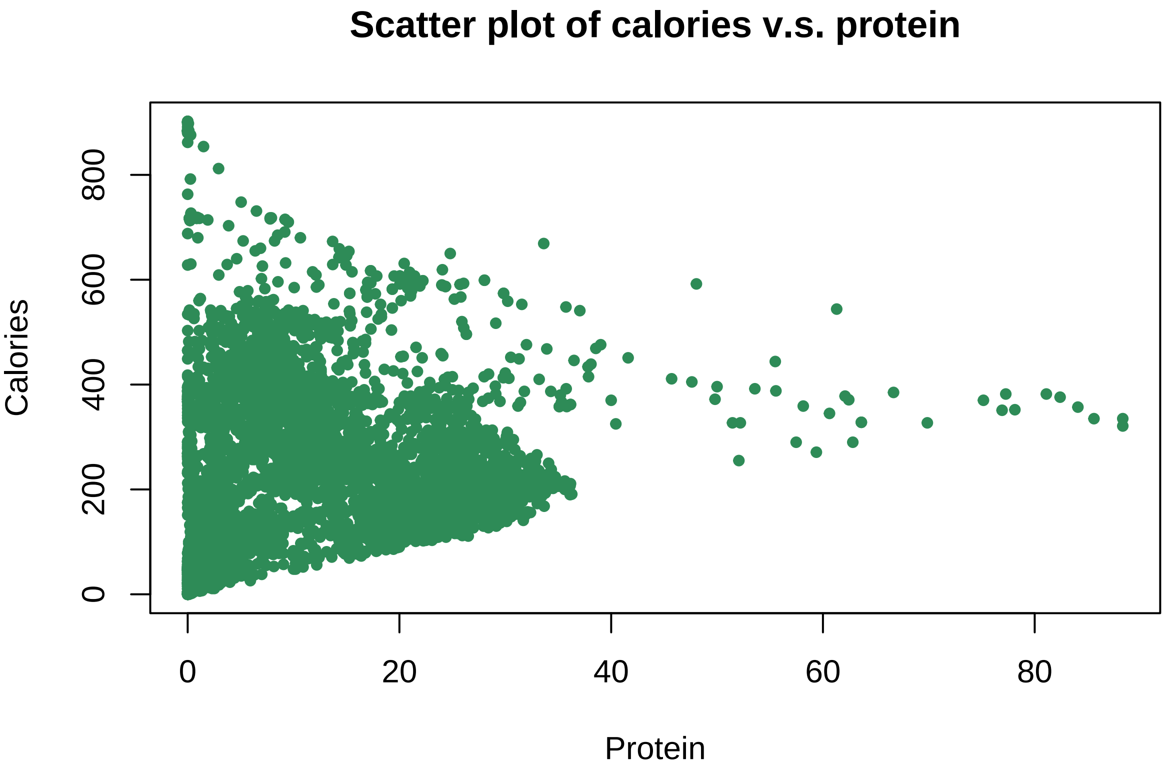


One unusual point to me is that the distribution is not of normal distribution, or say it's right-skewed since most of the food has low calories. Another could be that there's an outlier which carries more than 800 calories.

(b) Create a scatter plot of calories (y-axis) vs protein (x-axis). Make the plot presentable. Do you notice any trends? Do you think that knowing only the protein content of a food, you could make a good prediction of the calories in the food?

Solution:

```
plot(Calories ~ Protein, data = dataset,
  main = "Scatter plot of calories v.s. protein",
  ylab = "Calories",
  xlab = "Protein",
  col = "seagreen",
  pch = 20,
  cex = 1
)
```

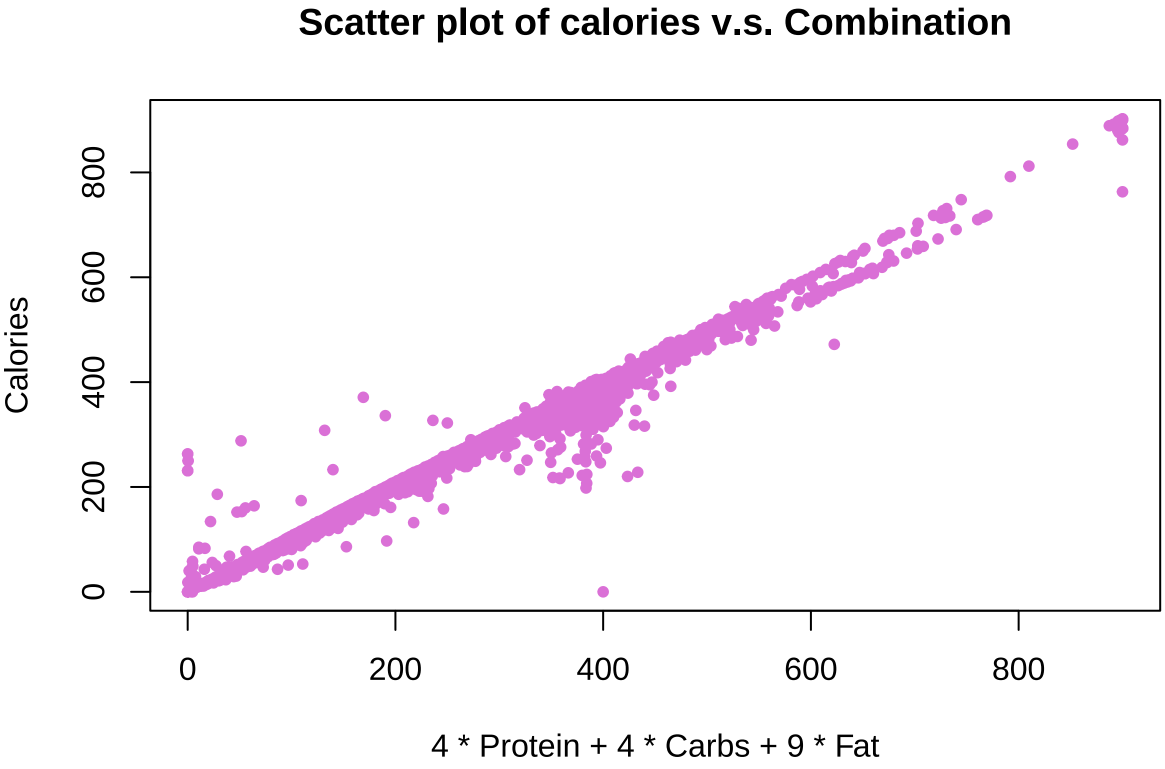


Yes, based on the plot, we can clearly see that there seems to have three edges that wraps up the overall distribution. Therefore, when I only know the protein content of a food, I can give a range of the calories that the food may carry with high confidence, but not an accurate prediction of the exact value.

(c) Create a scatter plot of `Calories` (y-axis) vs `4 * Protein + 4 * Carbs + 9 * Fat` (x-axis). Make the plot presentable. You will either need to add a new variable to the data frame, or use the `I()` function in your formula in the call to `plot()`. If you are at all familiar with nutrition, you may realize that this formula calculates the calorie count based on the protein, carbohydrate, and fat values. You'd expect then that the result here is a straight line. Is it? If not, can you think of any reasons why it is not?

Solution:

```
dataset$x_axis = 4 * dataset$Protein + 4 * dataset$Carbs + 9 * dataset$Fat
plot(Calories ~ x_axis, data = dataset,
  main = "Scatter plot of calories v.s. Combination",
  ylab = "Calories",
  xlab = "4 * Protein + 4 * Carbs + 9 * Fat",
  col = "orchid",
  pch = 20,
  cex = 1
)
```



Yes, it is a straight line with some of the points surronding it.

Exercise 4 (Writing and Using Functions)

For each of the following parts, use the following vectors:

```
a = 1:10
b = 10:1
c = rep(1, times = 10)
d = 2 ^ (1:10)
```

(a) Write a function called `sum_of_squares`.

- Arguments:
  - A vector of numeric data `x`
- Output:
  - The sum of the squares of the elements of the vector $\sum_{i=1}^{n} x_i^2$

Provide your function, as well as the result of running the following code:

Solution:

```r
sum_of_squares = function(x) {
  sum(x^2)
}
sum_of_squares(x = a)
```

```
## [1] 385
```

```r
sum_of_squares(x = c(c, d))
```

```
## [1] 1398110
```

(b) Using only your function `sum_of_squares()`, `mean()`, `sqrt()`, `length()`, and basic math operations such as `+` and `-`, calculate

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - 0)^2}$$

where the x vector is `d`.

Solution:

```r
sqrt(sum_of_squares(d - 0) / length(d))
```

```
## [1] 373.9118
```

(c) Using only your function `sum_of_squares()`, `mean()`, `sqrt()`, and basic math operations such as `+` and `-`, calculate

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

where the x vector is `a` and the y vector is `b`.

Solution:

```r
sqrt(sum_of_squares(a - b) / length(a))
```

```
## [1] 5.744563
```

Exercise 5 (More Writing and Using Functions)

For each of the following parts, use the following vectors:

```r
set.seed(2024)
x = 1:100
y = rnorm(1000)
z = runif(150, min = 0, max = 1)
```

(a) Write a function called `list_extreme_values`.

- Arguments:
  - A vector of numeric data `x`
  - A positive constant, `k`, with a default value of `2`
- Output:
  - A list with two elements:

- small, a vector of elements of x that are k sample standard deviations less than the sample mean. That is, the observations that are smaller than $\bar{x} - k \cdot s$.
- large, a vector of elements of x that are k sample standard deviations greater than the sample mean. That is, the observations that are larger than $\bar{x} + k \cdot s$.

Provide your function, as well as the result of running the following code:

Solution:

```r
list_extreme_values = function(x, k = 2) {
  standard_deviation = sd(x)
  list(
    small = x[x < mean(x) - k*standard_deviation],
    large = x[x > mean(x) + k*standard_deviation]
  )
}

list_extreme_values(x = x, k = 1)
```

```
## $small
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
## 
## $large
##  [1]  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
## [20]  99 100
```

```r
list_extreme_values(x = y, k = 3)
```

```
## $small
## [1] -3.274286 -2.951515
## 
## $large
## [1] 3.080996
```

```r
list_extreme_values(x = y, k = 2)
```

```
## $small
##  [1] -3.274286 -2.011471 -2.055722 -1.995504 -2.118813 -2.065853 -2.951515
##  [8] -2.742318 -2.540768 -2.075155 -2.039910 -2.060187 -2.265425 -2.090042
## [15] -2.121709 -1.962237 -2.582683 -2.096616 -2.042061 -2.916337 -1.995594
## [22] -1.968898 -1.990721
## 
## $large
##  [1] 1.972819 2.208662 2.393530 2.196040 2.296550 2.533193 2.930535 2.657171
##  [9] 2.325431 2.000434 2.098684 2.233508 2.027840 2.270658 2.007886 2.925858
## [17] 2.363357 3.080996 2.153014 2.189252 2.377901 2.082377 2.290894 2.180301
```

```r
list_extreme_values(x = z, k = 1.5)
```

```
## $small
## [1] 0.009706145 0.040536088 0.038652520 0.028539294 0.017371400 0.037101293
## [7] 0.015243588 0.032695471
## 
## $large
##  [1] 0.9707399 0.9259785 0.9604522 0.9558223 0.9339387 0.9291420 0.9784770
##  [8] 0.9135171 0.9880211 0.9927841 0.9609566 0.9667327 0.9203587 0.9131100
## [15] 0.9130935
```

(b) Using only your function `list_extreme_values()`, `mean()`, and basic list operations, calculate the mean of observations that are greater than 1.5 standard deviation above the mean in the vector `y`.

Solution:

```
mean(list_extreme_values(y, 1.5)$large)
```

```
## [1] 1.904525
```