Homework 2R

Likhit Raj Yerramsetty

03-07-2024

PART 1 -things are done for you.Go over everything again. There are parts that you have to do it.

Some define Statistics as the field that focuses on turning information into knowledge. The first step in that process is to summarize and describe the raw information - the data. In this homework, you will gain insight into public health by generating simple graphical and numerical summaries of a data set collected by the Centers for Disease Control and Prevention (CDC). As this is a large data set, along the way you'll also learn the indispensable skills of data processing and subsetting.

Getting started

The Behavioral Risk Factor Surveillance System (BRFSS) is an annual telephone survey of 350,000 people in the United States. As its name implies, the BRFSS is designed to identify risk factors in the adult population and report emerging health trends. For example, respondents are asked about their diet and weekly physical activity, their HIV/AIDS status, possible tobacco use, and even their level of healthcare coverage. The BRFSS Web site (http://www.cdc.gov/brfss) contains a complete description of the survey, including the research questions that motivate the study and many interesting results derived from the data.

We will focus on a random sample of 20,000 people from the BRFSS survey conducted in 2000. While there are over 200 variables in this data set, we will work with a small subset.

We begin by loading the data set of 20,000 observations into the R workspace. After launching RStudio, enter the following commands.

```
cdc <- read.csv("cdc.csv", as.is=T)
```

The data set `cdc` that shows up in your workspace is a *data frame*, with each row representing a *case* and each column representing a *variable*.

To view the names of the variables, type the command

```
names(cdc)
```

```
## [1] "genhlth"  "exerany"  "hlthplan" "smoke100" "height"   "weight"   "wtdesire"
## [8] "age"      "gender"
```

This returns the names `genhlth`, `exerany`, `hlthplan`, `smoke100`, `height`, `weight`, `wtdesire`, `age`, and `gender`. Each one of these variables corresponds to a question that was asked in the survey. For example, for `genhlth`, respondents were asked to evaluate their general health, responding either excellent, very good, good, fair or poor. The `exerany` variable indicates whether the respondent exercised in the past month (1) or did not (0). Likewise, `hlthplan` indicates whether the respondent had some form of health coverage (1)

or did not (0). The `smoke100` variable indicates whether the respondent had smoked at least 100 cigarettes in her lifetime. The other variables record the respondent's `height` in inches, `weight` in pounds as well as their desired weight, `wtdesire`, `age` in years, and `gender`.

A very useful function for taking a quick peek at your dataset is `summary`.

```
summary(cdc)
```

```
##    genhlth             exerany          hlthplan         smoke100
##  Length:20000       Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  Class :character   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000
##  Mode  :character   Median :1.0000   Median :1.0000   Median :0.0000
##                     Mean   :0.7457   Mean   :0.8738   Mean   :0.4721
##                     3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##                     Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##      height          weight         wtdesire          age
##  Min.   :48.00   Min.   : 68.0   Min.   : 68.0   Min.   :18.00
##  1st Qu.:64.00   1st Qu.:140.0   1st Qu.:130.0   1st Qu.:31.00
##  Median :67.00   Median :165.0   Median :150.0   Median :43.00
##  Mean   :67.18   Mean   :169.7   Mean   :155.1   Mean   :45.07
##  3rd Qu.:70.00   3rd Qu.:190.0   3rd Qu.:175.0   3rd Qu.:57.00
##  Max.   :93.00   Max.   :500.0   Max.   :680.0   Max.   :99.00
##     gender
##  Length:20000
##  Class :character
##  Mode  :character
##
##
##
```

Note that categorical variables in R are currently coded as `character` format, which is fine to start with. If we explicitly tell R to convert the variables to `factor` format, we can get some extra insight when we ask for the `summary()`.

```
cdc <- cdc %>% mutate(gender = factor(gender),
                      genhlth = factor(genhlth))
```

1. How many cases are there in this data set? How many variables? For each variable, identify its data type (e.g. categorical: ordinal, if the categories have an ordering, or not, numerical: continuous or discrete). Do not just rely on the R output, also think about the nature of the variables.

- GO: Answer 1: 20,000 number of cases. There are 9 variables in this data genhlth: categorical - ordinal, exerany: categorical- nominal, hlthplan: categorical - nominal, smoke100: categorical - nominal, height: numerical-continuous, weight: numerical-continuous, wtdesire: numerical-continuous, age: numerical-discrete, gender: categorical - nominal.

You could also look at *all* of the data frame at once by typing its name into the console, but that might be unwise here. We know `cdc` has 20,000 rows, so viewing the entire data set would mean flooding your screen. Using `head()` or `tail()` is the way to go!

Summaries and tables

The BRFSS questionnaire is a massive trove of information. A good first step in any analysis is to distill all of that information into a few summary statistics and graphics. As a simple example, the function `favstats` returns a numerical summary: minimum, first quartile, median, third quartile, maximum, mean, standard deviation, number of observations, and number of missing values. For `weight` this is

```
favstats(~weight, data = cdc)
```

```
##  min  Q1 median  Q3 max    mean       sd     n missing
##   68 140    165 190 500 169.683 40.08097 20000       0
```

As we have seen, R can function like a very fancy calculator. If you wanted to compute the interquartile range for the respondents' weight, you could look at the output from the summary command above and then enter

```
190 - 140
```

```
## [1] 50
```

```
iqr(~weight, data = cdc) # or have R do it for you!
```

```
## [1] 50
```

R also has built-in functions to compute summary statistics one by one. For instance, to calculate the mean, median, and variance of `weight`, type

```
mean(~weight, data = cdc)
```

```
## [1] 169.683
```

```
var(~weight, data = cdc)
```

```
## [1] 1606.484
```

```
median(~weight, data = cdc)
```

```
## [1] 165
```

While it makes sense to describe a quantitative variable like `weight` in terms of these statistics, what about categorical data? We would instead consider the sample frequency or relative frequency distribution. The function `tally` does this for you by counting the number of times each kind of response was given. For example, to see the number of people who have smoked 100 cigarettes in their lifetime, type

```
tally(~smoke100, data = cdc)
```

```
## smoke100
##     0     1
## 10559  9441
```

or instead look at the relative frequency distribution by typing

```
tally(~smoke100, data = cdc, format = "proportion")
```

Notice how R automatically divides all entries in the table by 20,000 in the command above. Next, we make a bar chart of the entries in the table by putting the table inside the `barchart` command.

```
barchart(tally(~smoke100, data = cdc, margins=FALSE), horizontal=FALSE)
```

Notice what we've done here! We've computed the table of `smoke100` and then immediately applied the graphical function, `barchart`. This is an important idea: R commands can be nested. You could also break this into two steps by typing the following:

```
smoke <- tally(~smoke100, data = cdc, margins=FALSE)
barchart(smoke, horizontal=FALSE)
```

Here, we've made a new object, a table, called `smoke` (the contents of which we can see by typing `smoke` into the console) and then used it in as the input for `barchart`. The special symbol `<-` performs an *assignment*, taking the output of one line of code and saving it into an object in your workspace.
This is another important idea that we'll return to later.

2. Create a numerical summary for `height` and `age`, and compute the interquartile range for each. Compute the relative frequency distribution (i.e. marginal distribution) for `gender` and `exerany`. How many males are in the sample? What proportion of the sample reports being in excellent health?

- GO: Height:

```
summary(cdc$height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   48.00   64.00   67.00   67.18   70.00   93.00
```

```
IQR(cdc$height)
```

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

Age:

```
summary(cdc$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   31.00   43.00   45.07   57.00   99.00
```

```
IQR(cdc$age)
```

```
## [1] 26
```

```
tally(~gender, data = cdc, format = "proportion")
```

```
## gender
##       f       m
## 0.52155 0.47845
```

```
tally(~exerany, data = cdc, format = "proportion")
```

```
## exerany
##      0      1
## 0.2543 0.7457
```

There are 9569 males in this sample. 23.285% of the sample data are in excellent health.

Modifying/Subsetting the Data

It's often useful to extract all individuals (cases) in a data set that have specific characteristics. We can do this easily using the `filter` function and a series of **logical operators**. The most commonly used logical operators for data analysis are

- `==` means "equal to"
- `!=` means "not equal to"
- `>` or `<` means "greater than" or "less than"
- `>=` or `<=` means "greater than or equal to" or "less than or equal to"

Using these, we can create a subset of the `cdc` dataset for just the men, and save this as a new dataset called `males`:

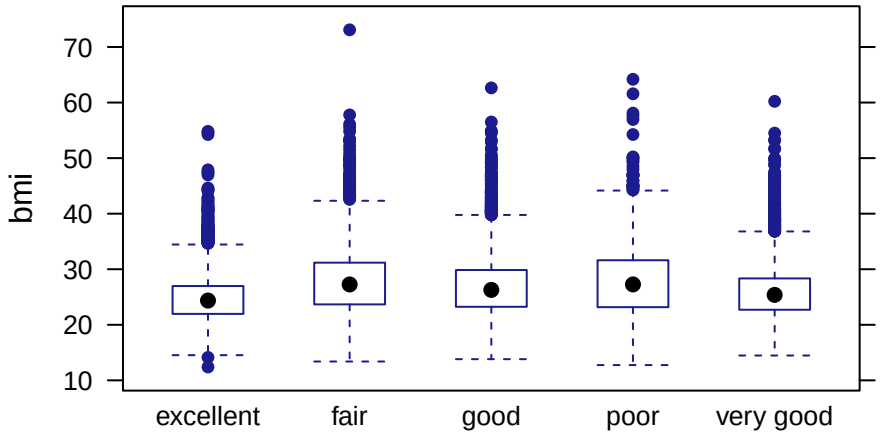
```
males <- cdc %>%
  filter(gender == "m")
```

The following two lines first make a new column called `bmi` and then creates box plots of these values, defining groups by the variable `genhlth`.

```
cdc <- cdc %>% mutate(bmi = (weight/height^2)*703)
names(cdc) # see, we now have a new column called bmi!
```

```
##  [1] "genhlth"  "exerany"  "hlthplan" "smoke100" "height"   "weight"
##  [7] "wtdesire" "age"      "gender"   "bmi"
```

```
bwplot(bmi ~ genhlth, data=cdc)
```



Notice that the first line above is just some arithmetic, but it's applied to all 20,000 numbers in the `cdc` data set. That is, for each of the 20,000 participants, we take their weight, divide by their height-squared and then multiply by 703. The result is 20,000 BMI values, one for each respondent. This is one reason why we like R: it lets us perform computations like this using very simple expressions.

This new data set contains all the same variables but just under half the rows.

As an aside, you can use several of these conditions together with `&` and `|`. The `&` is read "and" so that

```
males_and_over30 <- cdc %>%
  filter(gender == "m" & age > 30)
```

will give you the data for men over the age of 30. The `|` character is read "or" so that

```
males_or_over30 <- cdc %>%
  filter(gender == "m" | age > 30)
```

will take people who are men or over the age of 30 (why that's an interesting group is hard to say, but right now the mechanics of this are the important thing). In principle, you may use as many "and" and "or" clauses as you like when forming a subset.

3. Create a new object called `under23_and_smoke` that contains all observations of respondents under the age of 23 that have smoked 100 cigarettes in their lifetime. Write the command you used to create the new object as the answer to this exercise, and report the number of cases that meet this criteria.

- GO:

```
under23_and_smoke <- cdc %>%
  filter(age < 23 & smoke100 == 1)
nrow(under23_and_smoke)
```

```
## [1] 620
```

Visualization Tools

We've seen several ways to produce visual displays of variables.

- histogram (via `histogram()`)
- bargraph (via `bargraph()`)
- boxplot (via `bwplot()`)
- scatterplot (via `xyplot()`)

4. For each of these graphs, how many variables are displayed simultaneously? Are they categorical or quantitative variables?

- GO: Histogram: typically 1 quantitative variable

Bar graph: can display 1 or 2 variables. If it is displaying 1 variable, it is categorical. If it is displaying 2 variables, 1 variable is categorical and the other is quantitative.

box plot: can display 1 or 2 variables. If it is displaying 1 variable, it is categorical. If it is displaying 2 variables, 1 variable is categorical and the other is quantitative.

scatter plot: 2 quantitative variables

One way to visualize the relationship between two categorical variables, we can use a mosaic plot (no relation to the package name).

```
mosaicplot(tally(gender ~ smoke100, data = cdc))
```

We could have accomplished this in two steps by saving the table in one line and applying `mosaicplot` in the next (see the `tally`/`barchart` example above).

5. What does the mosaic plot reveal about smoking habits and gender?

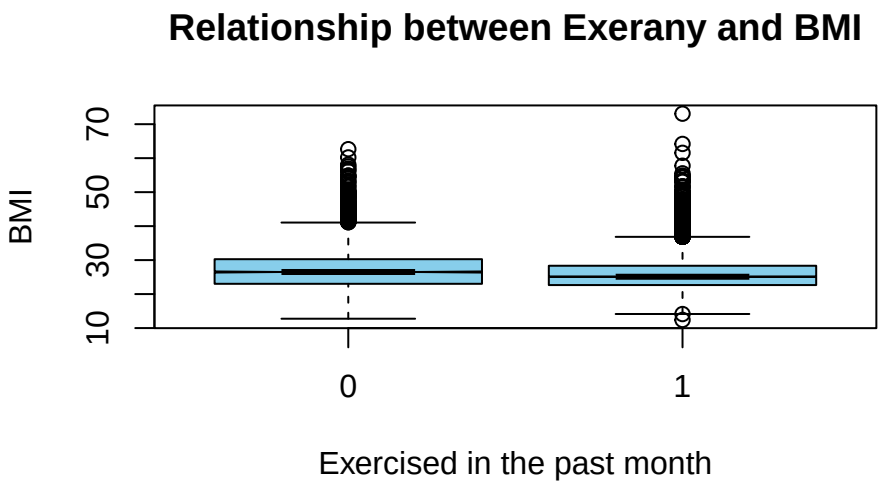
- GO: There are more men who smoke than females in this sample. Furthermore, a majority of females in the sample do not smoke. While a majority of men in the sample are smokers.

Let's get some more practice with the questions below.

6. Pick a categorical variable from the data set and see how it relates to BMI. List the variable you chose, why you might think it would have a relationship to BMI, create an appropriate plot, and provide an interpretation for this plot.

- GO: I chose exerany as the categorical variable as exercise has a strong relation to weight. As the BMI uses weight for its calculation, I thought exerany would be a good categorical variable to choose. As we can see the plot, people who have exercised in the past month have a lower BMI which is closer to the optimal range of BMI, we can see this from the IQR, and the median of the box plot. While, people who haven't exercised have a slightly higher BMI than the BMI of ones who have exercised. The data from the plot makes sense as it aligns with exerany being related to weight.

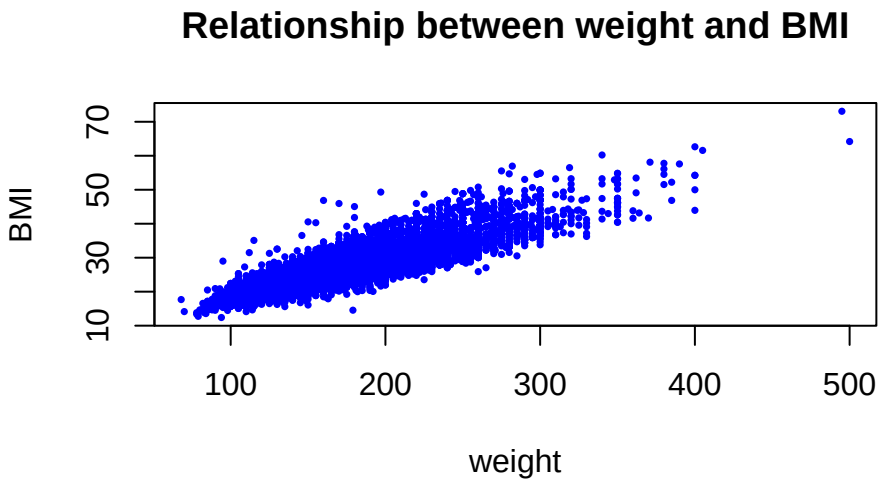
```
boxplot(bmi ~ exerany, data = cdc,
        xlab = "Exercised in the past month",
        ylab = "BMI",
        main = "Relationship between Exerany and BMI",
        col = "skyblue",
        notch = TRUE)
```



7. Pick a quantitative variable from the data set and see how it relates to BMI. List the variable you chose, why you might think it would have a relationship to BMI, create an appropriate plot, and provide an interpretation for this plot.

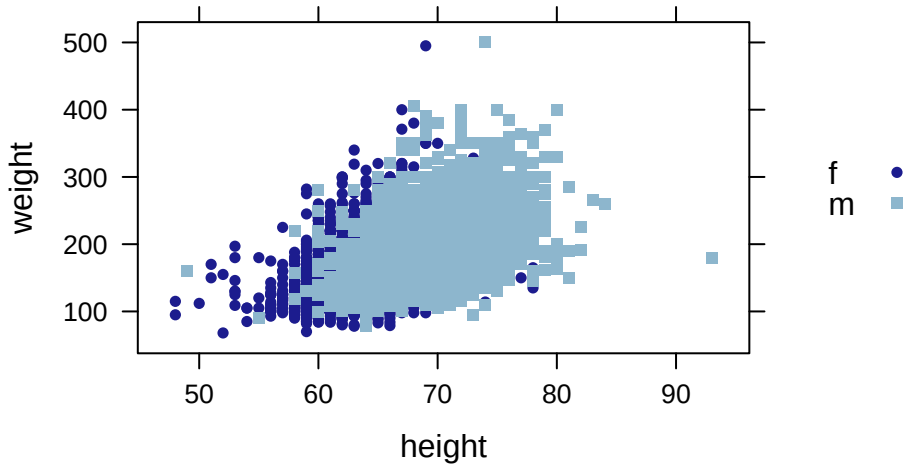
- GO: I chose weight to check the relation between a quantitative variable and BMI. We know that weight is directly proportional to BMI and this can see in the graph. As weight increases along the x axis, BMI also increases on the Y axis.

```
plot(cdc$weight, cdc$bmi,
     xlab = "weight", ylab = "BMI",
     main = "Relationship between weight and BMI",
     pch = 16, col = "blue", cex = 0.5)
```

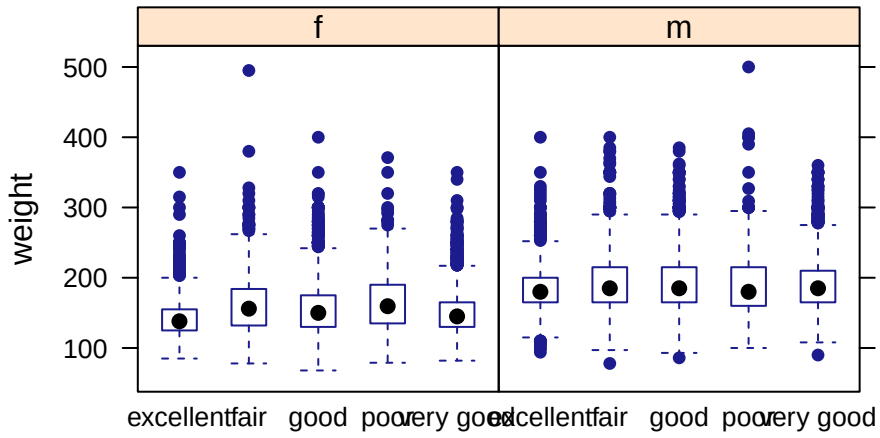


Remember that we can examine more than two variables in a plot by using `groups=` and adding panels using the `|` operator in our formula. Study the following examples:

```
xyplot(weight ~ height, groups=gender, data=cdc, auto.key=T)
```



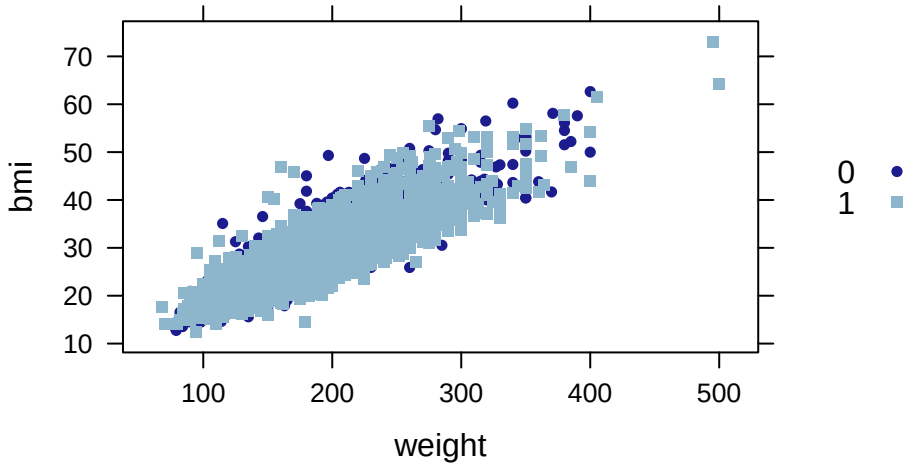

```
bwplot(weight ~ genhlth | gender, data=cdc )
```



8. Now combine all three variables used in the last two displays together into one plot. Describe what you learn from your plot.

- GO: From the graph, we can see that there are a lot more people who have exercised in the past month than people who haven't exercised in the past month. We can see that majority of the people who exercise are closer to the optimal BMI while people haven't exercised are scattered evenly across the graph.

```
xyplot(bmi ~ weight, groups=exerany, data=cdc, auto.key=T)
```



Part 2 - SLR

In this problem, we aim to produce a model to predict the number of calories using the amount of carbohydrates, fat, fiber, or protein that Starbucks food menu items contain.

```
#Be sure to download the file from Moodle and edit
#the path or working directory if needed
starbucks = read.csv("starbucks.csv")
```

(a) What is the most frequently appearing type of food product that is included in this dataset? Use some code to show how you arrived at your answer.

```
starbucks <- starbucks %>% mutate(type = factor(type))
summary(starbucks)
```

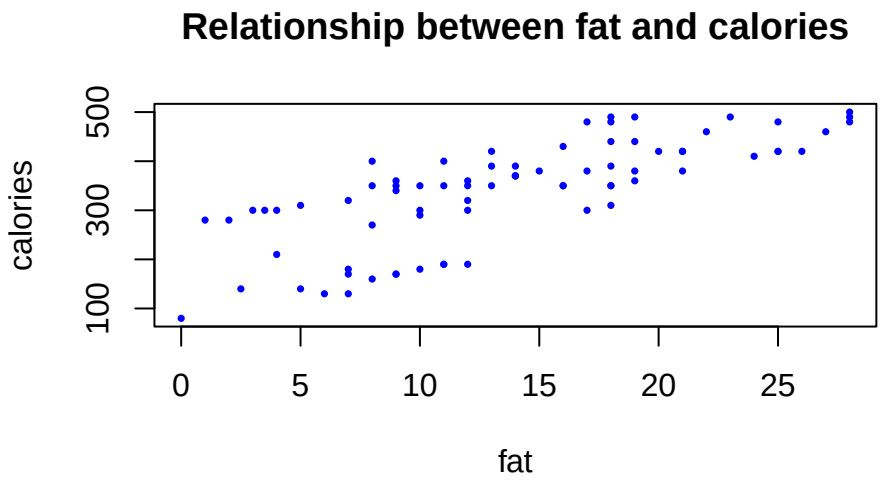
```
##      item              calories          fat              carb
##  Length:77          Min.   : 80.0   Min.   : 0.00   Min.   :16.00
##  Class :character   1st Qu.:300.0   1st Qu.: 9.00   1st Qu.:31.00
##  Mode  :character   Median :350.0   Median :13.00   Median :45.00
##                     Mean   :338.8   Mean   :13.77   Mean   :44.87
##                     3rd Qu.:420.0   3rd Qu.:18.00   3rd Qu.:59.00
##                     Max.   :500.0   Max.   :28.00   Max.   :80.00
##
##      fiber          protein                  type
##  Min.   :0.000   Min.   : 0.000   bakery        :41
##  1st Qu.:0.000   1st Qu.: 5.000   bistro box    : 8
##  Median :2.000   Median : 7.000   hot breakfast: 8
##  Mean   :2.221   Mean   : 9.481   parfait       : 3
##  3rd Qu.:4.000   3rd Qu.:15.000   petite        : 9
##  Max.   :7.000   Max.   :34.000   salad         : 1
##                                   sandwich      : 7
```

SOLUTION: Bakery type is the one with the highest frequency from this dataset.

(b) Use a scatterplot matrix to determine which ONE of the variables correlates best with `calories`.

SOLUTION:

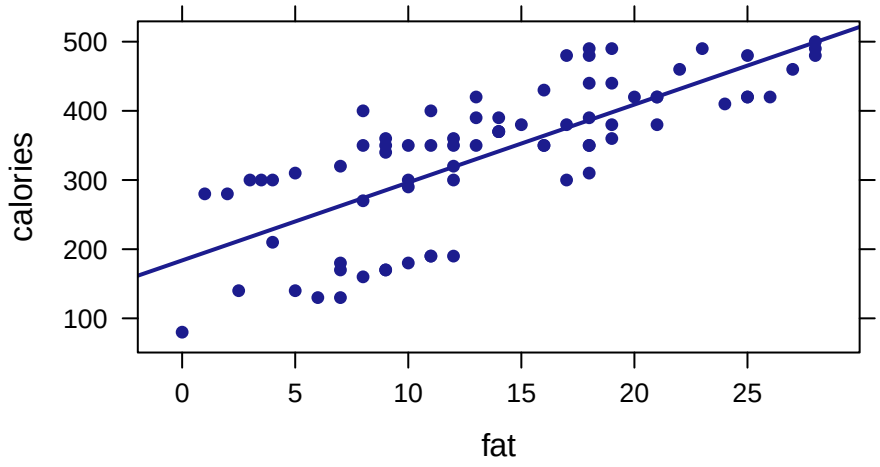
```
plot(starbucks$fat, starbucks$calories,
     xlab = "fat", ylab = "calories",
     main = "Relationship between fat and calories",
     pch = 16, col = "blue", cex = 0.5)
```



(c) Fit a simple linear regression model using the variable from (b) to predict `calories`. Show the model summary.

SOLUTION:

```
xyplot(calories ~ fat, starbucks, type=c("p", "r"))
```




```r
lm0 <- lm(calories ~ fat, starbucks)
summary(lm0)
```

```
##
## Call:
## lm(formula = calories ~ fat, data = starbucks)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -132.599  -44.130    3.469   54.868  126.134
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  183.734     17.277   10.63  < 2e-16 ***
## fat           11.267      1.117   10.09 1.32e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.1 on 75 degrees of freedom
## Multiple R-squared:  0.5756, Adjusted R-squared:  0.5699
## F-statistic: 101.7 on 1 and 75 DF,  p-value: 1.32e-15
```

(d) Interpret the intercept and slope based on the model summary.

SOLUTION: The intercept is 183.734 which is the estimated value of calories when fat is equal to zero. The slope is 11.267 which indicates the change in calories for every 5 units of increase in fat.

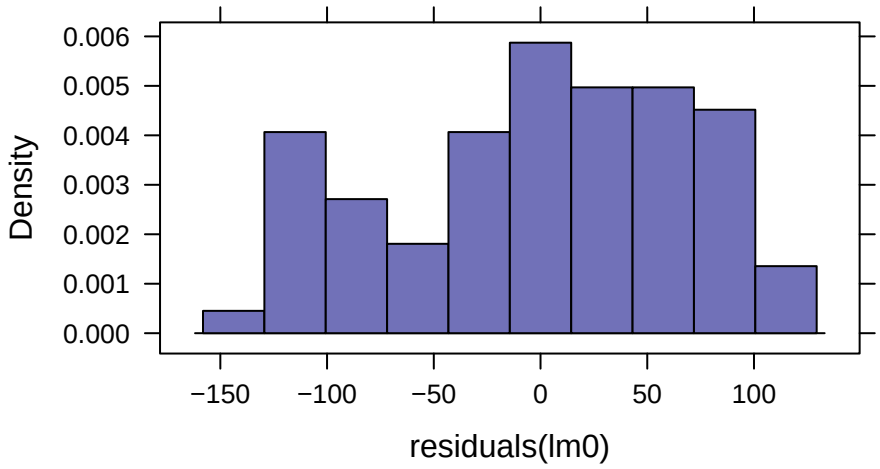
(e) Interpret the value of R^2.

SOLUTION: R^2 is 0.5756. So, approximately 57.56% of the variance in calories can be explained by the fat content.

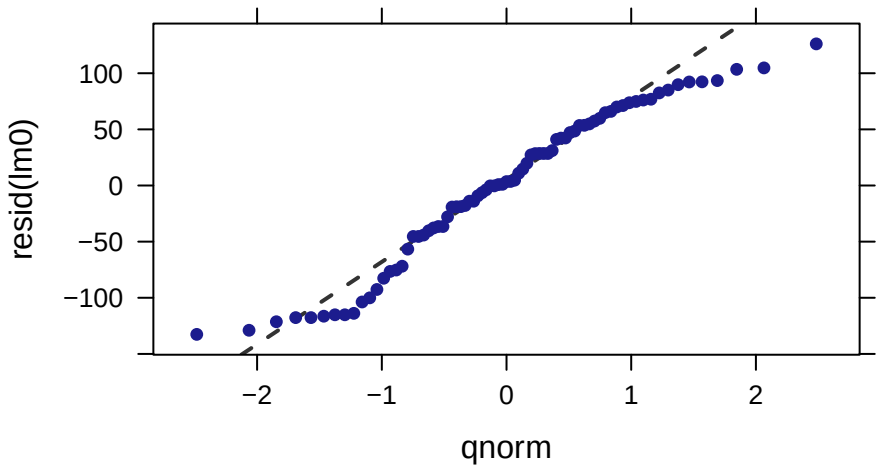
(f) Check the assumptions of the linear model using some plots. Do you conclude that a linear model is appropriate?

SOLUTION: a) Linearity: From the scatter plot of calories to fat, we can see that the calories increase as fat content increases, so, we can say that they are linear. b) Independence: We are selecting menu items from the starbucks csv file, assuming that the csv file has all the menu items or they were selected randomly, we can say that they are independent. c)Normality: Since, most of the points fall within the diagonal line from the resid-qnorm graph, we can say that it is normally distributed. (The points are evenly scattered in the residuals - fitted graph which also shows normality)

```
histogram(~ residuals(lm0), nint=10)
```




```
xqqmath(~resid(lm0))
```



```
xyplot(residuals(lm0) ~ fitted(lm0), type=c("p", "r"))
```