Apuntes Data Cleaning II

Pilar Amat Rodrigo

7/12/2017

Contents

Exercise 1__Sales

Data_Sales: https://assets.datacamp.com/production/course_1294/datasets/sales.csv

Importing

```
#Import sales.csv to the variable sales using the read.csv() function. Set the stringsAsFactors argumen

sales<- read.csv("sales.csv",stringsAsFactors=FALSE)
```

Examining the data

```
# View dimensions of sales
dim(sales)
```

```
## [1] 5000   46
```

```
# Inspect first 6 rows of sales
head(sales, n=6)
```

```
##   X             event_id       primary_act_id     secondary_act_id
## 1 1 abcaf1adb99a935fc661 43f0436b905bfa7c2eec b85143bf51323b72e53c
## 2 2 6c56d7f08c95f2aa453c 1a3e9aecd0617706a794 f53529c5679ea6ca5a48
## 3 3 c7ab4524a121f9d687d2 4b677c3f5bec71eec8d1 b85143bf51323b72e53c
## 4 4 394cb493f893be9b9ed1 b1ccea01ad6ef8522796 b85143bf51323b72e53c
## 5 5 55b5f67e618557929f48 91c03a34b562436efa3c b85143bf51323b72e53c
## 6 6 4f10fd8b9f550352bd56 ac4b847b3fde66f2117e 63814f3d63317f1b56c4
##    purch_party_lkup_id
## 1 7dfa56dd7d5956b17587
## 2 4f9e6fc637eaf7b736c2
## 3 6c2545703bd527a7144d
## 4 527d6b1eaffc69ddd882
## 5 8bd62c394a35213bdf52
## 6 3b3a628f83135acd0676
##                                                        event_name
## 1 Xfinity Center Mansfield Premier Parking: Florida Georgia Line
## 2                  Gorge Camping - dave matthews band - sept 3-7
## 3                    Dodge Theatre Adams Street Parking - benise
## 4   Gexa Energy Pavilion Vip Parking : kid rock with sheryl crow
## 5                                 Premier Parking - motley crue
## 6                                     Fast Lane Access: Journey
##                         primary_act_name secondary_act_name
## 1 XFINITY Center Mansfield Premier Parking               NULL
## 2                            Gorge Camping Dave Matthews Band
## 3                            Parking Event               NULL
## 4         Gexa Energy Pavilion VIP Parking               NULL
## 5 White River Amphitheatre Premier Parking               NULL
## 6                         Fast Lane Access            Journey
##   major_cat_name         minor_cat_name la_event_type_cat
## 1           MISC                PARKING           PARKING
## 2           MISC                CAMPING           INVALID
## 3           MISC                PARKING           PARKING
## 4           MISC                PARKING           PARKING
## 5           MISC                PARKING           PARKING
## 6           MISC SPECIAL ENTRY (UPSELL)            UPSELL
##                                                 event_disp_name
## 1 Xfinity Center Mansfield Premier Parking: Florida Georgia Line
## 2                 Gorge Camping - dave matthews band - sept 3-7
## 3                   Dodge Theatre Adams Street Parking - benise
## 4   Gexa Energy Pavilion Vip Parking : kid rock with sheryl crow
## 5                                 Premier Parking - motley crue
## 6                                     Fast Lane Access: Journey
##
## 1    THIS TICKET IS VALID        FOR PARKING ONLY         GOOD THIS DAY ONLY       PREMIER PARKING PA
## 2                                                               %OVERNIGHT C A M P I N G%* * * * *
## 3                               ADAMS STREET GARAGE%PARKING FOR 4/21/06 ONLY%DODGE THEATRE PARKING PA
## 4    THIS TICKET IS VALID        FOR PARKING ONLY      GOOD FOR THIS DATE ONLY       VIP PARKING PASS
## 5                              THIS TICKET IS VALID%FOR PARKING ONLY%GOOD THIS DATE ONLY%PREMIER PARK
## 6         FAST LANE                  JOURNEY               FAST LANE EVENT         THIS IS NOT A TICK
##   tickets_purchased_qty trans_face_val_amt delivery_type_cd
## 1                     1                 45          eTicket
## 2                     1                 75       TicketFast
## 3                     1                  5       TicketFast
## 4                     1                 20             Mail
## 5                     1                 20             Mail
## 6                     2                 10       TicketFast
##       event_date_time   event_dt presale_dt  onsale_dt
## 1 2015-09-12 23:30:00 2015-09-12       NULL 2015-05-15
## 2 2009-09-05 01:00:00 2009-09-04       NULL 2009-03-13
## 3 2006-04-22 01:30:00 2006-04-21       NULL 2006-02-25
## 4 2011-09-03 00:00:00 2011-09-02       NULL 2011-04-22
## 5 2005-07-31 01:00:00 2005-07-30 2005-03-02 2005-03-04
## 6 2012-07-22 02:00:00 2012-07-21       NULL 2012-04-11
##   sales_ord_create_dttm sales_ord_tran_dt   print_dt timezn_nm
## 1   2015-09-11 18:17:45        2015-09-11 2015-09-12       EST
## 2   2009-07-06 00:00:00        2009-07-05 2009-09-01       PST
## 3   2006-04-05 00:00:00        2006-04-05 2006-04-05       MST
## 4   2011-07-01 17:38:50        2011-07-01 2011-07-06       CST
## 5   2005-06-18 00:00:00        2005-06-18 2005-06-28       PST
## 6   2012-07-21 17:20:18        2012-07-21 2012-07-21       PST
##       venue_city   venue_state venue_postal_cd_sgmt_1
## 1      MANSFIELD MASSACHUSETTS                  02048
## 2         QUINCY    WASHINGTON                  98848
## 3        PHOENIX       ARIZONA                  85003
## 4         DALLAS         TEXAS                  75210
## 5         AUBURN    WASHINGTON                  98092
## 6 SAN BERNARDINO    CALIFORNIA                  92407
##             sales_platform_cd print_flg la_valid_tkt_event_flg  fin_mkt_nm
## 1 www.concerts.livenation.com         T                      N      Boston
## 2                        NULL         T                      N     Seattle
## 3                        NULL         T                      N     Arizona
## 4                        NULL         T                      N      Dallas
## 5                        NULL         T                      N     Seattle
## 6          www.livenation.com         T                      N Los Angeles
##   web_session_cookie_val gndr_cd age_yr income_amt edu_val
## 1   7dfa56dd7d5956b17587    <NA>   <NA>       <NA>    <NA>
## 2   4f9e6fc637eaf7b736c2    <NA>   <NA>       <NA>    <NA>
## 3   6c2545703bd527a7144d    <NA>   <NA>       <NA>    <NA>
## 4   527d6b1eaffc69ddd882    <NA>   <NA>       <NA>    <NA>
## 5   8bd62c394a35213bdf52    <NA>   <NA>       <NA>    <NA>
## 6   3b3a628f83135acd0676    <NA>   <NA>       <NA>    <NA>
##   edu_1st_indv_val edu_2nd_indv_val adults_in_hh_num married_ind
## 1             <NA>             <NA>             <NA>        <NA>
## 2             <NA>             <NA>             <NA>        <NA>
## 3             <NA>             <NA>             <NA>        <NA>
## 4             <NA>             <NA>             <NA>        <NA>
## 5             <NA>             <NA>             <NA>        <NA>
## 6             <NA>             <NA>             <NA>        <NA>
##   child_present_ind home_owner_ind occpn_val occpn_1st_val occpn_2nd_val
## 1              <NA>           <NA>      <NA>          <NA>          <NA>
## 2              <NA>           <NA>      <NA>          <NA>          <NA>
## 3              <NA>           <NA>      <NA>          <NA>          <NA>
## 4              <NA>           <NA>      <NA>          <NA>          <NA>
## 5              <NA>           <NA>      <NA>          <NA>          <NA>
## 6              <NA>           <NA>      <NA>          <NA>          <NA>
##   dist_to_ven
## 1          NA
## 2          59
## 3          NA
## 4          NA
## 5          NA
## 6          NA
```

```r
# View column names of sales
names(sales)
```

```
##  [1] "X"                      "event_id"
##  [3] "primary_act_id"         "secondary_act_id"
##  [5] "purch_party_lkup_id"    "event_name"
##  [7] "primary_act_name"       "secondary_act_name"
##  [9] "major_cat_name"         "minor_cat_name"
## [11] "la_event_type_cat"      "event_disp_name"
## [13] "ticket_text"            "tickets_purchased_qty"
## [15] "trans_face_val_amt"     "delivery_type_cd"
## [17] "event_date_time"        "event_dt"
## [19] "presale_dt"             "onsale_dt"
## [21] "sales_ord_create_dttm"  "sales_ord_tran_dt"
## [23] "print_dt"               "timezn_nm"
## [25] "venue_city"             "venue_state"
## [27] "venue_postal_cd_sgmt_1" "sales_platform_cd"
## [29] "print_flg"              "la_valid_tkt_event_flg"
## [31] "fin_mkt_nm"             "web_session_cookie_val"
## [33] "gndr_cd"                "age_yr"
## [35] "income_amt"             "edu_val"
## [37] "edu_1st_indv_val"       "edu_2nd_indv_val"
## [39] "adults_in_hh_num"       "married_ind"
## [41] "child_present_ind"      "home_owner_ind"
## [43] "occpn_val"              "occpn_1st_val"
## [45] "occpn_2nd_val"          "dist_to_ven"
```

```r
#take a look with glimpse it's part of the dplyr package
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
glimpse(sales)
```

```
## Observations: 5,000
## Variables: 46
## $ X                     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ...
## $ event_id              <chr> "abcaf1adb99a935fc661", "6c56d7f08c95f2...
## $ primary_act_id        <chr> "43f0436b905bfa7c2eec", "1a3e9aecd06177...
## $ secondary_act_id      <chr> "b85143bf51323b72e53c", "f53529c5679ea6...
## $ purch_party_lkup_id   <chr> "7dfa56dd7d5956b17587", "4f9e6fc637eaf7...
## $ event_name            <chr> "Xfinity Center Mansfield Premier Parki...
## $ primary_act_name      <chr> "XFINITY Center Mansfield Premier Parki...
## $ secondary_act_name    <chr> "NULL", "Dave Matthews Band", "NULL", "...
## $ major_cat_name        <chr> "MISC", "MISC", "MISC", "MISC", "MISC",...
## $ minor_cat_name        <chr> "PARKING", "CAMPING", "PARKING", "PARKI...
## $ la_event_type_cat     <chr> "PARKING", "INVALID", "PARKING", "PARKI...
## $ event_disp_name       <chr> "Xfinity Center Mansfield Premier Parki...
## $ ticket_text           <chr> "   THIS TICKET IS VALID        FOR PAR...
## $ tickets_purchased_qty <int> 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4, ...
## $ trans_face_val_amt    <dbl> 45, 75, 5, 20, 20, 10, 30, 28, 20, 25, ...
## $ delivery_type_cd      <chr> "eTicket", "TicketFast", "TicketFast", ...
## $ event_date_time       <chr> "2015-09-12 23:30:00", "2009-09-05 01:0...
## $ event_dt              <chr> "2015-09-12", "2009-09-04", "2006-04-21...
## $ presale_dt            <chr> "NULL", "NULL", "NULL", "NULL", "2005-0...
## $ onsale_dt             <chr> "2015-05-15", "2009-03-13", "2006-02-25...
## $ sales_ord_create_dttm <chr> "2015-09-11 18:17:45", "2009-07-06 00:0...
## $ sales_ord_tran_dt     <chr> "2015-09-11", "2009-07-05", "2006-04-05...
## $ print_dt              <chr> "2015-09-12", "2009-09-01", "2006-04-05...
## $ timezn_nm             <chr> "EST", "PST", "MST", "CST", "PST", "PST...
## $ venue_city             <chr> "MANSFIELD", "QUINCY", "PHOENIX", "DALL...
## $ venue_state            <chr> "MASSACHUSETTS", "WASHINGTON", "ARIZONA...
## $ venue_postal_cd_sgmt_1 <chr> "02048", "98848", "85003", "75210", "98...
## $ sales_platform_cd      <chr> "www.concerts.livenation.com", "NULL", ...
## $ print_flg              <chr> "T ", "T ", "T ", "T ", "T ", "T ", "T ...
## $ la_valid_tkt_event_flg <chr> "N ", "N ", "N ", "N ", "N ", "N ", "N ...
## $ fin_mkt_nm             <chr> "Boston", "Seattle", "Arizona", "Dallas...
## $ web_session_cookie_val <chr> "7dfa56dd7d5956b17587", "4f9e6fc637eaf7...
## $ gndr_cd                <chr> NA, NA, NA, NA, NA, NA, "M", NA, NA, NA...
## $ age_yr                 <chr> NA, NA, NA, NA, NA, NA, "28", NA, NA, N...
## $ income_amt             <chr> NA, NA, NA, NA, NA, NA, "112500", NA, N...
## $ edu_val                <chr> NA, NA, NA, NA, NA, NA, "High School", ...
## $ edu_1st_indv_val       <chr> NA, NA, NA, NA, NA, NA, "High School", ...
## $ edu_2nd_indv_val       <chr> NA, NA, NA, NA, NA, NA, "NULL", NA, NA,...
## $ adults_in_hh_num       <chr> NA, NA, NA, NA, NA, NA, "4", NA, NA, NA...
## $ married_ind            <chr> NA, NA, NA, NA, NA, NA, "0", NA, NA, NA...
## $ child_present_ind      <chr> NA, NA, NA, NA, NA, NA, "1", NA, NA, NA...
## $ home_owner_ind         <chr> NA, NA, NA, NA, NA, NA, "0", NA, NA, NA...
## $ occpn_val              <chr> NA, NA, NA, NA, NA, NA, "NULL", NA, NA,...
## $ occpn_1st_val          <chr> NA, NA, NA, NA, NA, NA, "Craftsman Blue...
## $ occpn_2nd_val          <chr> NA, NA, NA, NA, NA, NA, "NULL", NA, NA,...
## $ dist_to_ven            <int> NA, 59, NA, NA, NA, NA, NA, NA, NA, NA,...
```

Note: Notice the first column, X, which appears to just be counting.

Next will be removing first column

```
#Take a subset of sales to omit the first column. Assign the result to sales2

sales2<- sales[,-1]
head(sales2,n=3)
```

```
##                event_id       primary_act_id     secondary_act_id
## 1 abcaf1adb99a935fc661 43f0436b905bfa7c2eec b85143bf51323b72e53c
## 2 6c56d7f08c95f2aa453c 1a3e9aecd0617706a794 f53529c5679ea6ca5a48
## 3 c7ab4524a121f9d687d2 4b677c3f5bec71eec8d1 b85143bf51323b72e53c
##    purch_party_lkup_id
## 1 7dfa56dd7d5956b17587
## 2 4f9e6fc637eaf7b736c2
## 3 6c2545703bd527a7144d
##                                                         event_name
## 1 Xfinity Center Mansfield Premier Parking: Florida Georgia Line
## 2                  Gorge Camping - dave matthews band - sept 3-7
## 3                    Dodge Theatre Adams Street Parking - benise
##                        primary_act_name secondary_act_name
## 1 XFINITY Center Mansfield Premier Parking               NULL
## 2                         Gorge Camping Dave Matthews Band
## 3                         Parking Event               NULL
##   major_cat_name minor_cat_name la_event_type_cat
## 1           MISC        PARKING           PARKING
## 2           MISC        CAMPING           INVALID
## 3           MISC        PARKING           PARKING
##                                                    event_disp_name
## 1 Xfinity Center Mansfield Premier Parking: Florida Georgia Line
## 2                  Gorge Camping - dave matthews band - sept 3-7
## 3                    Dodge Theatre Adams Street Parking - benise
```

```
##
## 1    THIS TICKET IS VALID        FOR PARKING ONLY         GOOD THIS DAY ONLY       PREMIER PARKING PA
## 2                                                                %OVERNIGHT C A M P I N G%* * * * *
## 3                               ADAMS STREET GARAGE%PARKING FOR 4/21/06 ONLY%DODGE THEATRE PARKING PA
##   tickets_purchased_qty trans_face_val_amt delivery_type_cd
## 1                     1                 45          eTicket
## 2                     1                 75       TicketFast
## 3                     1                  5       TicketFast
##       event_date_time   event_dt presale_dt  onsale_dt
## 1 2015-09-12 23:30:00 2015-09-12       NULL 2015-05-15
## 2 2009-09-05 01:00:00 2009-09-04       NULL 2009-03-13
## 3 2006-04-22 01:30:00 2006-04-21       NULL 2006-02-25
##   sales_ord_create_dttm sales_ord_tran_dt   print_dt timezn_nm venue_city
## 1   2015-09-11 18:17:45        2015-09-11 2015-09-12       EST  MANSFIELD
## 2   2009-07-06 00:00:00        2009-07-05 2009-09-01       PST     QUINCY
## 3   2006-04-05 00:00:00        2006-04-05 2006-04-05       MST    PHOENIX
##     venue_state venue_postal_cd_sgmt_1           sales_platform_cd
## 1 MASSACHUSETTS                  02048 www.concerts.livenation.com
## 2    WASHINGTON                  98848                        NULL
## 3       ARIZONA                  85003                        NULL
##   print_flg la_valid_tkt_event_flg fin_mkt_nm web_session_cookie_val
## 1        T                      N      Boston   7dfa56dd7d5956b17587
## 2        T                      N     Seattle   4f9e6fc637eaf7b736c2
## 3        T                      N     Arizona   6c2545703bd527a7144d
##   gndr_cd age_yr income_amt edu_val edu_1st_indv_val edu_2nd_indv_val
## 1    <NA>   <NA>       <NA>    <NA>             <NA>             <NA>
## 2    <NA>   <NA>       <NA>    <NA>             <NA>             <NA>
## 3    <NA>   <NA>       <NA>    <NA>             <NA>             <NA>
##   adults_in_hh_num married_ind child_present_ind home_owner_ind occpn_val
## 1             <NA>        <NA>              <NA>           <NA>      <NA>
## 2             <NA>        <NA>              <NA>           <NA>      <NA>
## 3             <NA>        <NA>              <NA>           <NA>      <NA>
##   occpn_1st_val occpn_2nd_val dist_to_ven
## 1          <NA>          <NA>          NA
## 2          <NA>          <NA>          59
## 3          <NA>          <NA>          NA
```

Many of the columns have information that's of no use to us. For example, the first four columns contain internal codes representing particular events. The last fifteen columns also aren't worth keeping; there are too many missing values to make them worthwhile.

An easy way to get rid of unnecessary columns is to create a vector containing the column indices you want to keep, then subset the data based on that vector using single bracket subsetting.

```
# Define a vector of column indices: keep

keep<-(5:(ncol(sales2)-15))

# Subset sales2 using keep: sales3
sales3<- sales2[keep]
```

We have a sales3 with 26 variables.

Separating columns

Some of the columns in your data frame include multiple pieces of information that should be in separate columns. In this exercise, you will separate such a column into two: one for date and one for time. You will use the separate() function from the tidyr package (already installed for you).

Take a look at the event_date_time column by typing head(sales3$event_date_time) in the console. You'll notice that the date and time are separated by a space. Therefore, you'll use sep = " " as an argument to separate().

```
# Load tidyr
library(tidyr)

# Split event_date_time in "event_dt", "event_time": sales4
sales4 <- separate(sales3, event_date_time,
                   c("event_dt","event_time"), sep = " ")

# Split sales_ord_create_dttm in "ord_create_dt", "ord_create_time": sales5
sales5 <- separate(sales4, sales_ord_create_dttm,c("ord_create_dt", "ord_create_time"),sep=" ")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 4 rows
## [2516, 3863, 4082, 4183].
```

¡¡Warning message!!

Dealing with warnings

Looks like that second call to separate() threw a warning. Not to worry; warnings aren't as bad as error messages. It's not saying that the command didn't execute; it's just a heads-up that something unusual happened.

The warning says Too few values at 4 locations. You may be able to guess already what the issue is, but it's still good to take a look.

The locations (i.e. rows) given in the warning are 2516, 3863, 4082, and 4183. Have a look at the contents of the sales_ord_create_dttm column in those rows.

```
# lets look at the warnings, we see 4 NA in  $ord_create_time
# Define an issues vector
issues<-c(2516,3863,4082,4183)

# Print values of sales_ord_create_dttm at these indices
sales3$sales_ord_create_dttm[issues]
```

```
## [1] "NULL" "NULL" "NULL" "NULL"
```

```
# Print a well-behaved value of sales_ord_create_dttm
sales3$sales_ord_create_dttm[2517]
```

```
## [1] "2013-08-04 23:07:19"
```

The warning was just because of four missing values. You'll ignore them for now, but if your analysis depended on complete date/time information, you would probably need to delete those rows.

Identifying dates

Some of the columns in your dataset contain dates of different events. Right now, they are stored as character strings. That's fine if all you want to do is look up the date associated with an event, but if you want to do any comparisons or math with the dates, it's MUCH easier to store them as Date objects.

Luckily, all of the date columns in this dataset have the substring "dt" in their name, so you can use the **str_detect()** function of the **stringr package** to find the date columns. Then you can coerce them to Date objects using a function from the **lubridate package**.

You'll use **lapply()** to apply the appropriate lubridate function to all of the columns that contain dates. Recall the following syntax for lapply() applied to some data frame columns of interest:

`lapply(my_data_frame[, cols], function_name)`

Also recall that function names in lubridate combine the letters y, m, d, h, m, and s depending on the format of the date/time string being read in.

```r
# Load stringr
library(stringr)

# Find columns of sales5 containing "dt": date_cols
date_cols<-str_detect(names(sales5), pattern="dt")

# Load lubridate
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```r
# Coerce date columns into Date objects
sales5[, date_cols] <- lapply(sales5[, date_cols], ymd)
```

```
## Warning: 2892 failed to parse.

## Warning: 101 failed to parse.

## Warning: 4 failed to parse.

## Warning: 424 failed to parse.
```

More warnings! As you saw, some of the calls to ymd() caused a failure to parse warning. That's probably because of more missing data, but again, it's good to check to be sure.

The first two lines of code (provided for you here) create a list of logical vectors called missing. Each vector in the list indicates the presence (or absence) of missing values in the corresponding column of sales5. See if the number of missing values in each column is the same as the number of rows that failed to parse in the previous exercise.

```r
# Find date columns (don't change)
date_cols <- str_detect(names(sales5), "dt")

# Create logical vectors indicating missing values (don't change)
missing<- lapply(sales5[,date_cols],is.na)

# Create a numerical vector that counts missing values: num_missing
num_missing<- sapply(missing,sum)

# Print num_missing
print(num_missing)
```

```
##          event_dt        presale_dt         onsale_dt     ord_create_dt
##                 0              2892               101                 4
## sales_ord_tran_dt          print_dt
##                 0               424
```

Combining columns

Sure enough, the number of NAs in each column match the numbers from the warning messages, so missing data is the culprit. How to proceed depends on your desired analysis. If you really need complete sets of date/time information, you might delete the rows or columns containing NAs.

As your last step, you'll use the tidyr function **unite()** to combine the venue_city and venue_state columns into one column with the two values separated by a comma and a space. For example, "PORTLAND" "MAINE" should become "PORTLAND, MAINE".

```r
# Combine the venue_city and venue_state columns
sales6<- unite(sales5, "venue_city_state", "venue_city", "venue_state", sep=", ")


# View the head of sales6
glimpse(sales6$venue_city_state)
```

```
##  chr [1:5000] "MANSFIELD, MASSACHUSETTS" "QUINCY, WASHINGTON" ...
```

Exercise 2_MBTransportation

Data_Sales: https://www.datacamp.com/courses/importing-cleaning-data-in-r-case-studies

Importing

The dataset is stored as an Excel spreadsheet called mbta.xlsx in your working directory. You'll use the read_excel() function from Hadley Wickham's readxl package to import it.

The first time you import a dataset, you might not know how many rows need to be skipped. In this case, the first row is a title, so you'll need to skip the first row.

```r
library(readxl)

# Import mbta.xlsx and skip first row: mbta
mbta<-read_excel("mbta.xlsx", skip = 1)
```

Examining the Data

```r
# View the structure of mbta
str(mbta)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    11 obs. of  60 variables:
##  $ X__1   : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ mode   : chr  "All Modes by Qtr" "Boat" "Bus" "Commuter Rail" ...
##  $ 2007-01: chr  "NA" "4" "335.819" "142.2" ...
##  $ 2007-02: chr  "NA" "3.6" "338.675" "138.5" ...
##  $ 2007-03: num  1188 40 340 138 459 ...
##  $ 2007-04: chr  "NA" "4.3" "352.162" "139.5" ...
##  $ 2007-05: chr  "NA" "4.9" "354.367" "139" ...
##  $ 2007-06: num  1246 5.8 350.5 143 477 ...
##  $ 2007-07: chr  "NA" "6.521" "357.519" "142.391" ...
##  $ 2007-08: chr  "NA" "6.572" "355.479" "142.364" ...
##  $ 2007-09: num  1256.57 5.47 372.6 143.05 499.57 ...
##  $ 2007-10: chr  "NA" "5.145" "368.847" "146.542" ...
##  $ 2007-11: chr  "NA" "3.763" "330.826" "145.089" ...
##  $ 2007-12: num  1216.89 2.98 312.92 141.59 448.27 ...
##  $ 2008-01: chr  "NA" "3.175" "340.324" "142.145" ...
##  $ 2008-02: chr  "NA" "3.111" "352.905" "142.607" ...
##  $ 2008-03: num  1253.52 3.51 361.15 137.45 494.05 ...
##  $ 2008-04: chr  "NA" "4.164" "368.189" "140.389" ...
##  $ 2008-05: chr  "NA" "4.015" "363.903" "142.585" ...
##  $ 2008-06: num  1314.82 5.19 362.96 142.06 518.35 ...
##  $ 2008-07: chr  "NA" "6.016" "370.921" "145.731" ...
##  $ 2008-08: chr  "NA" "5.8" "361.057" "144.565" ...
##  $ 2008-09: num  1307.04 4.59 389.54 141.91 517.32 ...
##  $ 2008-10: chr  "NA" "4.285" "357.974" "151.957" ...
##  $ 2008-11: chr  "NA" "3.488" "345.423" "152.952" ...
##  $ 2008-12: num  1232.65 3.01 325.77 140.81 446.74 ...
##  $ 2009-01: chr  "NA" "3.014" "338.532" "141.448" ...
##  $ 2009-02: chr  "NA" "3.196" "360.412" "143.529" ...
##  $ 2009-03: num  1209.79 3.33 353.69 142.89 467.22 ...
##  $ 2009-04: chr  "NA" "4.049" "359.38" "142.34" ...
##  $ 2009-05: chr  "NA" "4.119" "354.75" "144.225" ...
##  $ 2009-06: num  1233.1 4.9 347.9 142 473.1 ...
##  $ 2009-07: chr  "NA" "6.444" "339.477" "137.691" ...
##  $ 2009-08: chr  "NA" "5.903" "332.661" "139.158" ...
##  $ 2009-09: num  1230.5 4.7 374.3 139.1 500.4 ...
##  $ 2009-10: chr  "NA" "4.212" "385.868" "137.104" ...
##  $ 2009-11: chr  "NA" "3.576" "366.98" "129.343" ...
##  $ 2009-12: num  1207.85 3.11 332.39 126.07 440.93 ...
##  $ 2010-01: chr  "NA" "3.207" "362.226" "130.91" ...
##  $ 2010-02: chr  "NA" "3.195" "361.138" "131.918" ...
##  $ 2010-03: num  1208.86 3.48 373.44 131.25 483.4 ...
##  $ 2010-04: chr  "NA" "4.452" "378.611" "131.722" ...
##  $ 2010-05: chr  "NA" "4.415" "380.171" "128.8" ...
##  $ 2010-06: num  1244.41 5.41 363.27 129.14 490.26 ...
##  $ 2010-07: chr  "NA" "6.513" "353.04" "122.935" ...
##  $ 2010-08: chr  "NA" "6.269" "343.688" "129.732" ...
##  $ 2010-09: num  1225.5 4.7 381.6 132.9 521.1 ...
##  $ 2010-10: chr  "NA" "4.402" "384.987" "131.033" ...
##  $ 2010-11: chr  "NA" "3.731" "367.955" "130.889" ...
##  $ 2010-12: num  1216.26 3.16 326.34 121.42 450.43 ...
##  $ 2011-01: chr  "NA" "3.14" "334.958" "128.396" ...
##  $ 2011-02: chr  "NA" "3.284" "346.234" "125.463" ...
##  $ 2011-03: num  1223.45 3.67 380.4 134.37 516.73 ...
##  $ 2011-04: chr  "NA" "4.251" "380.446" "134.169" ...
##  $ 2011-05: chr  "NA" "4.431" "385.289" "136.14" ...
##  $ 2011-06: num  1302.41 5.47 376.32 135.58 529.53 ...
##  $ 2011-07: chr  "NA" "6.581" "361.585" "132.41" ...
##  $ 2011-08: chr  "NA" "6.733" "353.793" "130.616" ...
##  $ 2011-09: num  1291 5 388 137 550 ...
##  $ 2011-10: chr  "NA" "4.484" "398.456" "128.72" ...
```

```r
# View the first 6 rows of mbta
head(mbta, n=6)
```

```
## # A tibble: 6 x 60
##    X__1 mode   `2007-01` `2007-02` `2007-03` `2007-04` `2007-05` `2007-06`
##   <dbl> <chr>  <chr>     <chr>         <dbl> <chr>     <chr>         <dbl>
## 1     1 All M~ NA        NA            1188. NA        NA            1246.
## 2     2 Boat   4         3.6             40  4.3       4.9             5.8
## 3     3 Bus    335.819   338.675         340. 352.162   354.367       351.
## 4     4 Commu~ 142.2     138.5           138. 139.5     139           143
## 5     5 Heavy~ 435.294   448.271         459. 472.201   474.579       477.
## 6     6 Light~ 227.231   240.262         241. 255.557   248.262       246.
## # ... with 52 more variables: `2007-07` <chr>, `2007-08` <chr>,
## #   `2007-09` <dbl>, `2007-10` <chr>, `2007-11` <chr>, `2007-12` <dbl>,
## #   `2008-01` <chr>, `2008-02` <chr>, `2008-03` <dbl>, `2008-04` <chr>,
## #   `2008-05` <chr>, `2008-06` <dbl>, `2008-07` <chr>, `2008-08` <chr>,
## #   `2008-09` <dbl>, `2008-10` <chr>, `2008-11` <chr>, `2008-12` <dbl>,
## #   `2009-01` <chr>, `2009-02` <chr>, `2009-03` <dbl>, `2009-04` <chr>,
## #   `2009-05` <chr>, `2009-06` <dbl>, `2009-07` <chr>, `2009-08` <chr>,
## #   `2009-09` <dbl>, `2009-10` <chr>, `2009-11` <chr>, `2009-12` <dbl>,
## #   `2010-01` <chr>, `2010-02` <chr>, `2010-03` <dbl>, `2010-04` <chr>,
## #   `2010-05` <chr>, `2010-06` <dbl>, `2010-07` <chr>, `2010-08` <chr>,
## #   `2010-09` <dbl>, `2010-10` <chr>, `2010-11` <chr>, `2010-12` <dbl>,
## #   `2011-01` <chr>, `2011-02` <chr>, `2011-03` <dbl>, `2011-04` <chr>,
## #   `2011-05` <chr>, `2011-06` <dbl>, `2011-07` <chr>, `2011-08` <chr>,
## #   `2011-09` <dbl>, `2011-10` <chr>
```

```r
# View a summary of mbta
summary(mbta)
```

```
##       X__1          mode              2007-01            2007-02
##  Min.   : 1.0   Length:11          Length:11          Length:11
##  1st Qu.: 3.5   Class :character   Class :character   Class :character
##  Median : 6.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 6.0
##  3rd Qu.: 8.5
##  Max.   :11.0
##     2007-03            2007-04            2007-05
##  Min.   :   0.114   Length:11          Length:11
##  1st Qu.:   9.278   Class :character   Class :character
##  Median : 137.700   Mode  :character   Mode  :character
##  Mean   : 330.293
##  3rd Qu.: 399.225
##  Max.   :1204.725
##     2007-06            2007-07            2007-08
##  Min.   :   0.096   Length:11          Length:11
##  1st Qu.:   5.700   Class :character   Class :character
##  Median : 143.000   Mode  :character   Mode  :character
##  Mean   : 339.846
##  3rd Qu.: 413.788
##  Max.   :1246.129
##     2007-09            2007-10            2007-11
##  Min.   :  -0.007   Length:11          Length:11
##  1st Qu.:   5.539   Class :character   Class :character
##  Median : 143.051   Mode  :character   Mode  :character
##  Mean   : 352.554                                    
##  3rd Qu.: 436.082                                    
##  Max.   :1310.764                                    
##     2007-12            2008-01            2008-02
##  Min.   :  -0.060   Length:11          Length:11
##  1st Qu.:   4.385   Class :character   Class :character
##  Median : 141.585   Mode  :character   Mode  :character
##  Mean   : 321.588
##  3rd Qu.: 380.594
##  Max.   :1216.890
##     2008-03            2008-04            2008-05
##  Min.   :   0.058   Length:11          Length:11
##  1st Qu.:   5.170   Class :character   Class :character
##  Median : 137.453   Mode  :character   Mode  :character
##  Mean   : 345.604
##  3rd Qu.: 427.601
##  Max.   :1274.031
##     2008-06            2008-07            2008-08
##  Min.   :   0.060   Length:11          Length:11
##  1st Qu.:   5.742   Class :character   Class :character
##  Median : 142.057   Mode  :character   Mode  :character
##  Mean   : 359.667
##  3rd Qu.: 440.656
##  Max.   :1320.728
##     2008-09            2008-10            2008-11
##  Min.   :   0.021   Length:11          Length:11
##  1st Qu.:   5.691   Class :character   Class :character
##  Median : 141.907   Mode  :character   Mode  :character
##  Mean   : 362.099
##  3rd Qu.: 453.430
##  Max.   :1338.015
##     2008-12            2009-01            2009-02
##  Min.   :  -0.015   Length:11          Length:11
##  1st Qu.:   4.689   Class :character   Class :character
##  Median : 140.810   Mode  :character   Mode  :character
##  Mean   : 319.882
##  3rd Qu.: 386.255
##  Max.   :1232.655
##     2009-03            2009-04            2009-05
##  Min.   :  -0.050   Length:11          Length:11
##  1st Qu.:   5.003   Class :character   Class :character
##  Median : 142.893   Mode  :character   Mode  :character
##  Mean   : 330.142
##  3rd Qu.: 410.455
##  Max.   :1210.912
##     2009-06            2009-07            2009-08
##  Min.   :  -0.079   Length:11          Length:11
##  1st Qu.:   5.845   Class :character   Class :character
##  Median : 142.006   Mode  :character   Mode  :character
##  Mean   : 333.194
##  3rd Qu.: 410.482
##  Max.   :1233.085
##     2009-09            2009-10            2009-11
##  Min.   :  -0.035   Length:11          Length:11
##  1st Qu.:   5.693   Class :character   Class :character
##  Median : 139.087   Mode  :character   Mode  :character
##  Mean   : 346.687
##  3rd Qu.: 437.332
##  Max.   :1291.564
##     2009-12           2010-01            2010-02
##  Min.   :  -0.022   Length:11          Length:11
##  1st Qu.:   4.784   Class :character   Class :character
##  Median : 126.066   Mode  :character   Mode  :character
##  Mean   : 312.962
##  3rd Qu.: 386.659
##  Max.   :1207.845
##     2010-03           2010-04            2010-05
##  Min.   :   0.012   Length:11          Length:11
##  1st Qu.:   5.274   Class :character   Class :character
##  Median : 131.252   Mode  :character   Mode  :character
##  Mean   : 332.726
##  3rd Qu.: 428.420
##  Max.   :1225.556
##     2010-06           2010-07            2010-08
##  Min.   :   0.008   Length:11          Length:11
##  1st Qu.:   6.436   Class :character   Class :character
##  Median : 129.144   Mode  :character   Mode  :character
##  Mean   : 335.964
##  3rd Qu.: 426.769
##  Max.   :1244.409
##     2010-09           2010-10            2010-11
##  Min.   :   0.001   Length:11          Length:11
##  1st Qu.:   5.567   Class :character   Class :character
##  Median : 132.892   Mode  :character   Mode  :character
##  Mean   : 346.524
##  3rd Qu.: 451.361
##  Max.   :1293.117
##     2010-12           2011-01            2011-02
##  Min.   :  -0.004   Length:11          Length:11
##  1st Qu.:   4.466   Class :character   Class :character
##  Median : 121.422   Mode  :character   Mode  :character
##  Mean   : 312.917
##  3rd Qu.: 388.385
##  Max.   :1216.262
##     2011-03          2011-04            2011-05
##  Min.   :   0.05   Length:11          Length:11
##  1st Qu.:   6.03   Class :character   Class :character
##  Median : 134.37   Mode  :character   Mode  :character
##  Mean   : 345.17
##  3rd Qu.: 448.56
##  Max.   :1286.66
##     2011-06           2011-07            2011-08
##  Min.   :   0.054   Length:11          Length:11
##  1st Qu.:   6.926   Class :character   Class :character
##  Median : 135.581   Mode  :character   Mode  :character
##  Mean   : 353.331
##  3rd Qu.: 452.923
##  Max.   :1302.414
##     2011-09           2011-10
##  Min.   :   0.043   Length:11
##  1st Qu.:   6.660   Class :character
##  Median : 136.901   Mode  :character
##  Mean   : 362.555
##  3rd Qu.: 469.204
##  Max.   :1348.754
```

Removing unnecessary rows and columns

```
# Remove the first, seventh, and eleventh rows of mbta (All Modes By Qtr, Pct Chg / Yr, and TOTAL). Nam
mbta2<- mbta[-(c(1,7,11)),]

# Remove the first column of mbta2. Name the resulting data frame mbta3

mbta3<-mbta2[,-1]
head(mbta3)
```

```
## # A tibble: 6 x 59
##   mode          `2007-01` `2007-02` `2007-03` `2007-04` `2007-05` `2007-06`
##   <chr>         <chr>     <chr>         <dbl> <chr>     <chr>         <dbl>
## 1 Boat          4         3.6           40    4.3       4.9             5.8
## 2 Bus           335.819   338.675      340.   352.162   354.367       351.
## 3 Commuter Ra~ 142.2     138.5        138.   139.5     139           143
## 4 Heavy Rail    435.294   448.271      459.   472.201   474.579       477.
## 5 Light Rail    227.231   240.262      241.   255.557   248.262       246.
## 6 Private Bus   4.772     4.417          4.57 4.542     4.768           4.72
## # ... with 52 more variables: `2007-07` <chr>, `2007-08` <chr>,
## #   `2007-09` <dbl>, `2007-10` <chr>, `2007-11` <chr>, `2007-12` <dbl>,
## #   `2008-01` <chr>, `2008-02` <chr>, `2008-03` <dbl>, `2008-04` <chr>,
## #   `2008-05` <chr>, `2008-06` <dbl>, `2008-07` <chr>, `2008-08` <chr>,
## #   `2008-09` <dbl>, `2008-10` <chr>, `2008-11` <chr>, `2008-12` <dbl>,
## #   `2009-01` <chr>, `2009-02` <chr>, `2009-03` <dbl>, `2009-04` <chr>,
## #   `2009-05` <chr>, `2009-06` <dbl>, `2009-07` <chr>, `2009-08` <chr>,
## #   `2009-09` <dbl>, `2009-10` <chr>, `2009-11` <chr>, `2009-12` <dbl>,
## #   `2010-01` <chr>, `2010-02` <chr>, `2010-03` <dbl>, `2010-04` <chr>,
## #   `2010-05` <chr>, `2010-06` <dbl>, `2010-07` <chr>, `2010-08` <chr>,
## #   `2010-09` <dbl>, `2010-10` <chr>, `2010-11` <chr>, `2010-12` <dbl>,
## #   `2011-01` <chr>, `2011-02` <chr>, `2011-03` <dbl>, `2011-04` <chr>,
## #   `2011-05` <chr>, `2011-06` <dbl>, `2011-07` <chr>, `2011-08` <chr>,
## #   `2011-09` <dbl>, `2011-10` <chr>
```

Observations are stored in columns

As is customary, you want to represent variables in columns rather than rows. The first step is to use the **gather()** function from the **tidyr package**, which will gather columns into key-value pairs.

```
library(tidyr)

# Gather columns of mbta3: mbta4
mbta4<- gather(data = mbta3, key = "month",value = "thou_riders", -mode)
```

```
# View the head of mbta4
head(mbta4)
```

```
## # A tibble: 6 x 3
##   mode           month   thou_riders
##   <chr>          <chr>   <chr>
## 1 Boat           2007-01 4
## 2 Bus            2007-01 335.819
## 3 Commuter Rail  2007-01 142.2
## 4 Heavy Rail     2007-01 435.294
## 5 Light Rail     2007-01 227.231
## 6 Private Bus    2007-01 4.772
```

Type conversions

But first, take this opportunity to change the average weekday ridership column, thou_riders, into numeric values rather than character strings. That way, you'll be able to do things like compare values and do math.

```
# Coerce thou_riders to numeric

mbta4$thou_riders<-as.numeric(mbta4$thou_riders)

head(mbta4,n=5)
```

```
## # A tibble: 5 x 3
##   mode           month   thou_riders
##   <chr>          <chr>         <dbl>
## 1 Boat           2007-01          4
## 2 Bus            2007-01        336.
## 3 Commuter Rail  2007-01        142.
## 4 Heavy Rail     2007-01        435.
## 5 Light Rail     2007-01        227.
```

Variables are stored in both rows and columns

Now, you can finish the job you started earlier: getting variables into columns. Right now, variables are stored as "keys" in the mode column. You'll use the tidyr function spread() to make them into columns containing average weekday ridership for the given month and mode of transport.

```
# Spread the contents of mbta4: mbta5
mbta5<- spread(mbta4,mode,thou_riders)

# View the head of mbta5
head(mbta5)
```

```
## # A tibble: 6 x 9
##   month    Boat   Bus `Commuter Rail` `Heavy Rail` `Light Rail`
##   <chr>   <dbl> <dbl>           <dbl>        <dbl>        <dbl>
## 1 2007-01   4    336.            142.         435.         227.
## 2 2007-02   3.6  339.            138.         448.         240.
## 3 2007-03  40    340.            138.         459.         241.
## 4 2007-04   4.3  352.            140.         472.         256.
## 5 2007-05   4.9  354.            139          475.         248.
## 6 2007-06   5.8  351.            143          477.         246.
## # ... with 3 more variables: `Private Bus` <dbl>, RIDE <dbl>, `Trackless
## #   Trolley` <dbl>
```

Separating columns

In this exercise, you'll separate the month column into distinct month and year columns to make life easier.

```
# Split month column into month and year: mbta6
mbta6<- separate(mbta5, month, c("year","month"), sep="-")

# View the head of mbta6
head(mbta6,n=3)
```

```
## # A tibble: 3 x 10
##   year  month  Boat   Bus `Commuter Rail` `Heavy Rail` `Light Rail`
##   <chr> <chr> <dbl> <dbl>           <dbl>        <dbl>        <dbl>
## 1 2007  01      4    336.            142.         435.         227.
## 2 2007  02      3.6  339.            138.         448.         240.
## 3 2007  03     40    340.            138.         459.         241.
## # ... with 3 more variables: `Private Bus` <dbl>, RIDE <dbl>, `Trackless
## #   Trolley` <dbl>
```

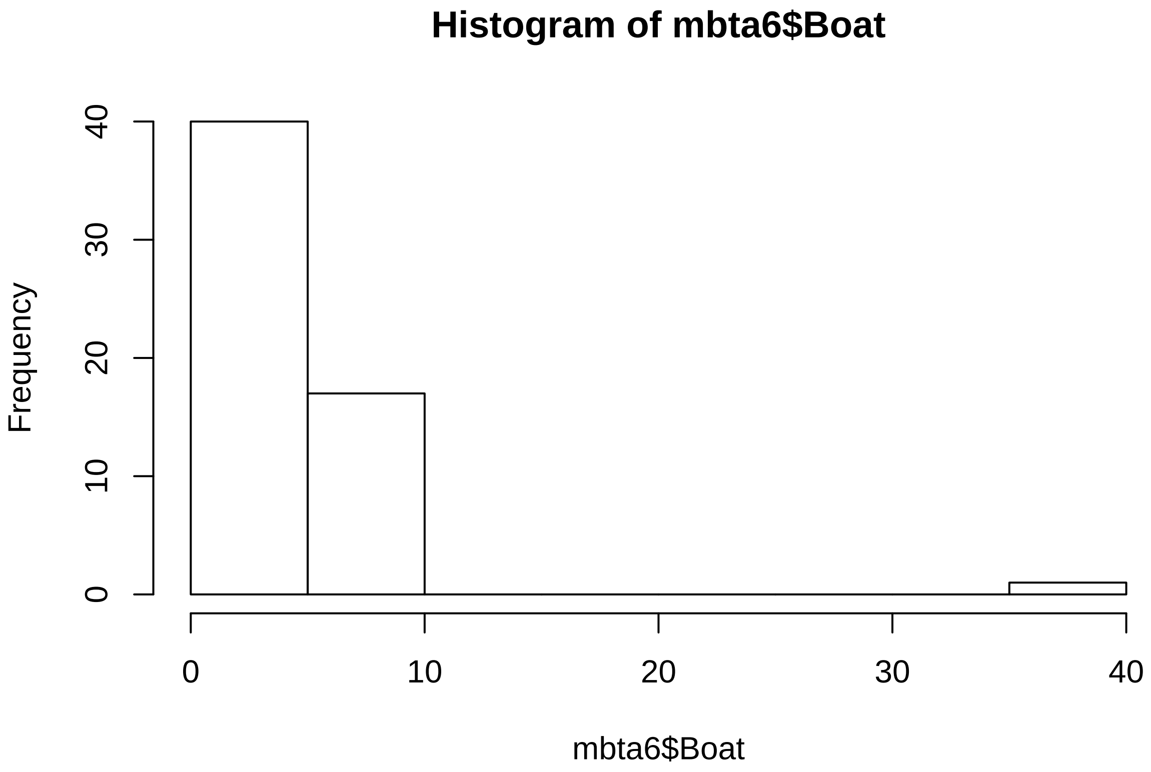
Outliers

Let's take a look of summary to see if there is any possible outlier. It may be sthng weird in Boat.

```
# View a summary of mbta6
summary(mbta6)
```

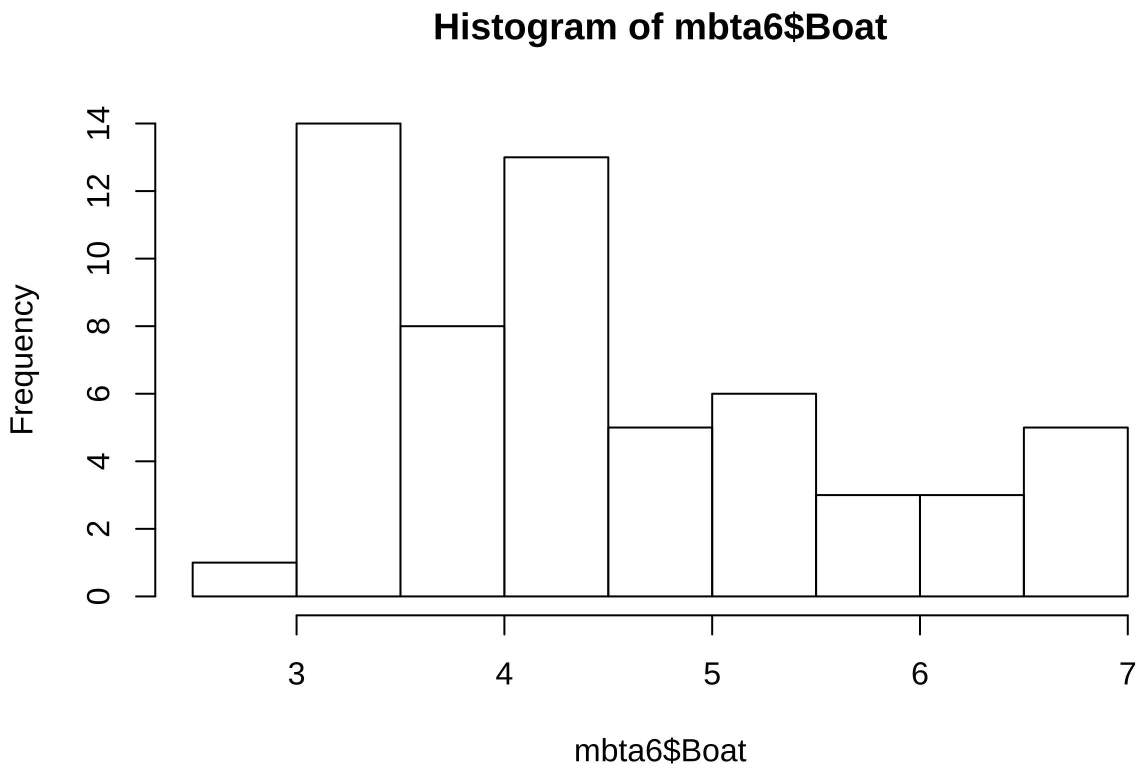
```
##      year              month                Boat             Bus
##  Length:58          Length:58          Min.   : 2.985   Min.   :312.9
##  Class :character   Class :character   1st Qu.: 3.494   1st Qu.:345.6
##  Mode  :character   Mode  :character   Median : 4.293   Median :359.9
##                                        Mean   : 5.068   Mean   :358.6
##                                        3rd Qu.: 5.356   3rd Qu.:372.2
##                                        Max.   :40.000   Max.   :398.5
##  Commuter Rail     Heavy Rail      Light Rail     Private Bus
##  Min.   :121.4   Min.   :435.3   Min.   :194.4   Min.   :2.213
##  1st Qu.:131.4   1st Qu.:471.1   1st Qu.:220.6   1st Qu.:2.641
##  Median :138.8   Median :487.3   Median :231.9   Median :2.820
##  Mean   :137.4   Mean   :489.3   Mean   :233.0   Mean   :3.352
##  3rd Qu.:142.4   3rd Qu.:511.3   3rd Qu.:244.5   3rd Qu.:4.167
##  Max.   :153.0   Max.   :554.9   Max.   :271.1   Max.   :4.878
##       RIDE        Trackless Trolley
##  Min.   :4.900   Min.   : 5.777
##  1st Qu.:5.965   1st Qu.:11.679
##  Median :6.615   Median :12.598
##  Mean   :6.604   Mean   :12.125
##  3rd Qu.:7.149   3rd Qu.:13.320
##  Max.   :8.598   Max.   :15.109
```

```
# Generate a histogram of Boat column
hist(mbta6$Boat)
```



Replace this 40 by a 4 to correct the outlier.

```r
# Find the row number of the incorrect value: i
i<-which(mbta6$Boat==40)

# Replace the incorrect value with 4
mbta6$Boat[i]<-4

# Generate a histogram of Boat column
hist(mbta6$Boat)
```



Exercise 3_Nutrition_Food

Data_Sales: https://www.datacamp.com/courses/importing-cleaning-data-in-r-case-studies Data_file: food.csv

Importing

A large dataset called food.csv is ready for your use in the working directory. Instead of the usual read.csv(), however, you're going to use the faster **fread()** from the **data.table package**. The data will come in as a data table, but since you're used to working with data frames, you can just convert it.

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
# Import food.csv: dt_food
dt_food<- fread("food.csv")

# Convert dt_food to a data frame
df_food<- data.frame(dt_food)
```

Examining

```r
summary(df_food)
```

```
##        V1              code             url              creator         
##  Min.   :   1.0   Min.   :100030   Length:1500        Length:1500       
##  1st Qu.: 375.8   1st Qu.:124974   Class :character   Class :character  
##  Median : 750.5   Median :149514   Mode  :character   Mode  :character  
##  Mean   : 750.5   Mean   :149613                                        
##  3rd Qu.:1125.2   3rd Qu.:174506                                        
##  Max.   :1500.0   Max.   :199880                                        
##                                                                         
##    created_t         created_datetime   last_modified_t    
##  Min.   :1.332e+09   Length:1500        Min.   :1.340e+09  
##  1st Qu.:1.394e+09   Class :character   1st Qu.:1.424e+09  
##  Median :1.425e+09   Mode  :character   Median :1.437e+09  
##  Mean   :1.414e+09                      Mean   :1.430e+09  
##  3rd Qu.:1.436e+09                      3rd Qu.:1.446e+09  
##  Max.   :1.453e+09                      Max.   :1.453e+09  
##                                                            
##  last_modified_datetime product_name       generic_name      
##  Length:1500            Length:1500        Length:1500       
##  Class :character       Class :character   Class :character  
##  Mode  :character       Mode  :character   Mode  :character  
##                                                              
##                                                              
##                                                              
##                                                              
##    quantity          packaging         packaging_tags    
##  Length:1500        Length:1500        Length:1500       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##     brands          brands_tags         categories       
##  Length:1500        Length:1500        Length:1500       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##  categories_tags    categories_en        origins         
##  Length:1500        Length:1500        Length:1500       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##  origins_tags       manufacturing_places manufacturing_places_tags
##  Length:1500        Length:1500          Length:1500              
##  Class :character   Class :character     Class :character
##  Mode  :character   Mode  :character     Mode  :character
##
##
##
##
##     labels          labels_tags         labels_en
##  Length:1500        Length:1500        Length:1500
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   emb_codes         emb_codes_tags     first_packaging_code_geo
##  Length:1500        Length:1500        Length:1500
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   cities        cities_tags        purchase_places       stores
##  Mode:logical   Length:1500        Length:1500        Length:1500
##  NA's:1500      Class :character   Class :character   Class :character
##                 Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   countries         countries_tags     countries_en
##  Length:1500        Length:1500        Length:1500
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  ingredients_text    allergens         allergens_en      traces
##  Length:1500        Length:1500        Mode:logical   Length:1500
##  Class :character   Class :character   NA's:1500      Class :character
##  Mode  :character   Mode  :character                  Mode  :character
##
##
##
##
##  traces_tags         traces_en         serving_size       no_nutriments
##  Length:1500        Length:1500        Length:1500        Mode:logical
##  Class :character   Class :character   Class :character   NA's:1500
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##   additives_n      additives         additives_tags     additives_en
##  Min.   : 0.000   Length:1500        Length:1500        Length:1500
##  1st Qu.: 0.000   Class :character   Class :character   Class :character
##  Median : 1.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 1.846
##  3rd Qu.: 3.000
##  Max.   :17.000
##  NA's   :514
##  ingredients_from_palm_oil_n ingredients_from_palm_oil
##  Min.   :0.0000              Mode:logical
##  1st Qu.:0.0000              NA's:1500
##  Median :0.0000
##  Mean   :0.0487
##  3rd Qu.:0.0000
##  Max.   :1.0000
##  NA's   :514
##  ingredients_from_palm_oil_tags ingredients_that_may_be_from_palm_oil_n
##  Length:1500                    Min.   :0.0000
##  Class :character               1st Qu.:0.0000
##  Mode  :character               Median :0.0000
##                                 Mean   :0.1379
##                                 3rd Qu.:0.0000
##                                 Max.   :4.0000
##                                 NA's   :514
##  ingredients_that_may_be_from_palm_oil
##  Mode:logical
##  NA's:1500
##
##
##
##
##
##  ingredients_that_may_be_from_palm_oil_tags nutrition_grade_uk
##  Length:1500                                Mode:logical
##  Class :character                           NA's:1500
##  Mode  :character
##
##
##
##
##  nutrition_grade_fr pnns_groups_1      pnns_groups_2
##  Length:1500        Length:1500        Length:1500
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     states          states_tags         states_en
##  Length:1500        Length:1500        Length:1500
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##
##  main_category      main_category_en    image_url
##  Length:1500        Length:1500        Length:1500
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  image_small_url     energy_100g     energy_from_fat_100g    fat_100g
##  Length:1500        Min.   :   0.0   Min.   :   0.00      Min.   :  0.00
##  Class :character   1st Qu.: 369.8   1st Qu.:  35.98      1st Qu.:  0.90
##  Mode  :character   Median : 966.5   Median : 237.00      Median :  6.00
##                     Mean   :1083.2   Mean   : 668.41      Mean   : 13.39
##                     3rd Qu.:1641.5   3rd Qu.: 974.00      3rd Qu.: 20.00
##                     Max.   :3700.0   Max.   :2900.00      Max.   :100.00
##                     NA's   :700      NA's   :1486         NA's   :708
##  saturated_fat_100g butyric_acid_100g caproic_acid_100g caprylic_acid_100g
##  Min.   : 0.000     Mode:logical      Mode:logical      Mode:logical
##  1st Qu.: 0.200     NA's:1500         NA's:1500         NA's:1500
##  Median : 1.700
##  Mean   : 4.874
##  3rd Qu.: 6.500
##  Max.   :57.000
##  NA's   :797
##  capric_acid_100g lauric_acid_100g myristic_acid_100g palmitic_acid_100g
##  Mode:logical     Mode:logical     Mode:logical       Mode:logical
##  NA's:1500        NA's:1500        NA's:1500          NA's:1500
##
##
##
##
##
##  stearic_acid_100g arachidic_acid_100g behenic_acid_100g
##  Mode:logical      Mode:logical        Mode:logical
##  NA's:1500         NA's:1500           NA's:1500
##
##
##
##
##
##  lignoceric_acid_100g cerotic_acid_100g montanic_acid_100g
##  Mode:logical         Mode:logical      Mode:logical
##  NA's:1500            NA's:1500         NA's:1500
##
##
##
##
##
##  melissic_acid_100g monounsaturated_fat_100g polyunsaturated_fat_100g
##  Mode:logical       Min.   : 0.00            Min.   : 0.400
##  NA's:1500          1st Qu.: 3.87            1st Qu.: 1.653
##                     Median : 9.50            Median : 3.900
##                      Mean   :19.77            Mean   : 9.986  
##                      3rd Qu.:29.00            3rd Qu.:12.700  
##                      Max.   :75.00            Max.   :46.200  
##                      NA's   :1465             NA's   :1464    
##  omega_3_fat_100g alpha_linolenic_acid_100g eicosapentaenoic_acid_100g
##  Min.   : 0.033   Min.   :0.0800            Min.   :0.721             
##  1st Qu.: 1.300   1st Qu.:0.0905            1st Qu.:0.721             
##  Median : 3.000   Median :0.1010            Median :0.721             
##  Mean   : 3.726   Mean   :0.1737            Mean   :0.721             
##  3rd Qu.: 3.200   3rd Qu.:0.2205            3rd Qu.:0.721             
##  Max.   :12.400   Max.   :0.3400            Max.   :0.721             
##  NA's   :1491     NA's   :1497              NA's   :1499              
##  docosahexaenoic_acid_100g omega_6_fat_100g linoleic_acid_100g
##  Min.   :1.09              Min.   :0.25     Min.   :0.5000    
##  1st Qu.:1.09              1st Qu.:0.25     1st Qu.:0.5165    
##  Median :1.09              Median :0.25     Median :0.5330    
##  Mean   :1.09              Mean   :0.25     Mean   :0.5330    
##  3rd Qu.:1.09              3rd Qu.:0.25     3rd Qu.:0.5495    
##  Max.   :1.09              Max.   :0.25     Max.   :0.5660    
##  NA's   :1499              NA's   :1499     NA's   :1498      
##  arachidonic_acid_100g gamma_linolenic_acid_100g
##  Mode:logical          Mode:logical             
##  NA's:1500             NA's:1500                
##                                                 
##                                                 
##                                                 
##                                                 
##                                                 
##  dihomo_gamma_linolenic_acid_100g omega_9_fat_100g oleic_acid_100g
##  Mode:logical                     Mode:logical     Mode:logical   
##  NA's:1500                        NA's:1500        NA's:1500      
##                                                                   
##                                                                   
##                                                                   
##                                                                   
##                                                                   
##  elaidic_acid_100g gondoic_acid_100g mead_acid_100g erucic_acid_100g
##  Mode:logical      Mode:logical      Mode:logical   Mode:logical    
##  NA's:1500         NA's:1500         NA's:1500      NA's:1500       
##                                                                     
##                                                                     
##                                                                     
##                                                                     
##                                                                     
##  nervonic_acid_100g trans_fat_100g   cholesterol_100g carbohydrates_100g
##  Mode:logical       Min.   :0.0000   Min.   :0.0000   Min.   :  0.000   
##  NA's:1500          1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  3.792   
##                     Median :0.0000   Median :0.0000   Median : 13.500   
##                     Mean   :0.0105   Mean   :0.0265   Mean   : 27.958   
##                     3rd Qu.:0.0000   3rd Qu.:0.0026   3rd Qu.: 55.000   
##                     Max.   :0.1000   Max.   :0.4300   Max.   :100.000   
##                     NA's   :1481     NA's   :1477     NA's   :708       
##   sugars_100g     sucrose_100g   glucose_100g   fructose_100g
##  Min.   :  0.00   Mode:logical   Mode:logical   Min.   :100  
##  1st Qu.:  1.00   NA's:1500      NA's:1500      1st Qu.:100
##  Median :  4.05                                 Median :100
##  Mean   : 12.66                                 Mean   :100
##  3rd Qu.: 14.70                                 3rd Qu.:100
##  Max.   :100.00                                 Max.   :100
##  NA's   :788                                    NA's   :1499
##   lactose_100g   maltose_100g   maltodextrins_100g  starch_100g
##  Min.   :0.000   Mode:logical   Mode:logical       Min.   : 0.00
##  1st Qu.:0.250   NA's:1500      NA's:1500          1st Qu.: 9.45
##  Median :0.500                                     Median :39.50
##  Mean   :2.933                                     Mean   :30.73
##  3rd Qu.:4.400                                     3rd Qu.:42.85
##  Max.   :8.300                                     Max.   :71.00
##  NA's   :1497                                      NA's   :1493
##   polyols_100g     fiber_100g     proteins_100g     casein_100g
##  Min.   : 8.60   Min.   : 0.000   Min.   : 0.000   Min.   :1.1
##  1st Qu.:59.10   1st Qu.: 0.500   1st Qu.: 1.500   1st Qu.:1.1
##  Median :67.00   Median : 1.750   Median : 6.000   Median :1.1
##  Mean   :56.06   Mean   : 2.823   Mean   : 7.563   Mean   :1.1
##  3rd Qu.:69.80   3rd Qu.: 3.500   3rd Qu.:10.675   3rd Qu.:1.1
##  Max.   :70.00   Max.   :46.700   Max.   :61.000   Max.   :1.1
##  NA's   :1491    NA's   :994      NA's   :710      NA's   :1499
##  serum_proteins_100g nucleotides_100g   salt_100g         sodium_100g
##  Mode:logical        Mode:logical     Min.   :  0.0000   Min.   : 0.0000
##  NA's:1500           NA's:1500        1st Qu.:  0.0438   1st Qu.: 0.0172
##                                       Median :  0.4498   Median : 0.1771
##                                       Mean   :  1.1205   Mean   : 0.4409
##                                       3rd Qu.:  1.1938   3rd Qu.: 0.4700
##                                       Max.   :102.0000   Max.   :40.0000
##                                       NA's   :780        NA's   :780
##   alcohol_100g   vitamin_a_100g   beta_carotene_100g vitamin_d_100g
##  Min.   : 0.00   Min.   :0.0000   Mode:logical       Min.   :0e+00
##  1st Qu.: 0.00   1st Qu.:0.0000   NA's:1500          1st Qu.:0e+00
##  Median : 5.50   Median :0.0001                      Median :0e+00
##  Mean   :10.07   Mean   :0.0003                      Mean   :0e+00
##  3rd Qu.:13.00   3rd Qu.:0.0006                      3rd Qu.:0e+00
##  Max.   :50.00   Max.   :0.0013                      Max.   :1e-04
##  NA's   :1433    NA's   :1477                        NA's   :1485
##  vitamin_e_100g   vitamin_k_100g vitamin_c_100g  vitamin_b1_100g
##  Min.   :0.0005   Min.   :0      Min.   :0.000   Min.   :0.0001
##  1st Qu.:0.0021   1st Qu.:0      1st Qu.:0.002   1st Qu.:0.0003
##  Median :0.0044   Median :0      Median :0.019   Median :0.0004
##  Mean   :0.0069   Mean   :0      Mean   :0.025   Mean   :0.0006
##  3rd Qu.:0.0097   3rd Qu.:0      3rd Qu.:0.030   3rd Qu.:0.0010
##  Max.   :0.0320   Max.   :0      Max.   :0.217   Max.   :0.0013
##  NA's   :1478     NA's   :1498   NA's   :1459    NA's   :1478
##  vitamin_b2_100g  vitamin_pp_100g  vitamin_b6_100g  vitamin_b9_100g
##  Min.   :0.0002   Min.   :0.0006   Min.   :0.0001   Min.   :0e+00
##  1st Qu.:0.0003   1st Qu.:0.0033   1st Qu.:0.0002   1st Qu.:0e+00
##  Median :0.0009   Median :0.0069   Median :0.0008   Median :1e-04
##  Mean   :0.0011   Mean   :0.0086   Mean   :0.0112   Mean   :1e-04
##  3rd Qu.:0.0013   3rd Qu.:0.0140   3rd Qu.:0.0012   3rd Qu.:2e-04
##  Max.   :0.0066   Max.   :0.0160   Max.   :0.2000   Max.   :2e-04
##  NA's   :1483     NA's   :1484     NA's   :1481     NA's   :1483
##  vitamin_b12_100g biotin_100g    pantothenic_acid_100g  silica_100g
##  Min.   :0        Min.   :0      Min.   :0.0000        Min.   :8e-04
##  1st Qu.:0        1st Qu.:0      1st Qu.:0.0007        1st Qu.:8e-04
##  Median :0        Median :0      Median :0.0020        Median :8e-04
##  Mean   :0        Mean   :0      Mean   :0.0027        Mean   :8e-04
##  3rd Qu.:0        3rd Qu.:0      3rd Qu.:0.0051        3rd Qu.:8e-04
##  Max.   :0        Max.   :0      Max.   :0.0060        Max.   :8e-04
##  NA's   :1489     NA's   :1498   NA's   :1486          NA's   :1499
##  bicarbonate_100g potassium_100g   chloride_100g     calcium_100g
##  Min.   :0.0006   Min.   :0.0000   Min.   :0.0003   Min.   :0.0000
##  1st Qu.:0.0678   1st Qu.:0.0650   1st Qu.:0.0006   1st Qu.:0.0450
##  Median :0.1350   Median :0.1940   Median :0.0009   Median :0.1200
##  Mean   :0.1692   Mean   :0.3288   Mean   :0.0144   Mean   :0.2040
##  3rd Qu.:0.2535   3rd Qu.:0.3670   3rd Qu.:0.0214   3rd Qu.:0.1985
##  Max.   :0.3720   Max.   :1.4300   Max.   :0.0420   Max.   :1.0000
##  NA's   :1497     NA's   :1487     NA's   :1497     NA's   :1449
##  phosphorus_100g    iron_100g      magnesium_100g     zinc_100g
##  Min.   :0.0430   Min.   :0.0000   Min.   :0.0000   Min.   :0.0005
##  1st Qu.:0.1938   1st Qu.:0.0012   1st Qu.:0.0670   1st Qu.:0.0009
##  Median :0.3185   Median :0.0042   Median :0.1040   Median :0.0017
##  Mean   :0.3777   Mean   :0.0045   Mean   :0.1066   Mean   :0.0016
##  3rd Qu.:0.4340   3rd Qu.:0.0077   3rd Qu.:0.1300   3rd Qu.:0.0022
##  Max.   :1.1550   Max.   :0.0137   Max.   :0.3330   Max.   :0.0026
##  NA's   :1488     NA's   :1463     NA's   :1479     NA's   :1493
##   copper_100g    manganese_100g fluoride_100g  selenium_100g
##  Min.   :0e+00   Min.   :0      Min.   :0      Min.   :0
##  1st Qu.:1e-04   1st Qu.:0      1st Qu.:0      1st Qu.:0
##  Median :1e-04   Median :0      Median :0      Median :0
##  Mean   :1e-04   Mean   :0      Mean   :0      Mean   :0
##  3rd Qu.:1e-04   3rd Qu.:0      3rd Qu.:0      3rd Qu.:0
##  Max.   :1e-04   Max.   :0      Max.   :0      Max.   :0
##  NA's   :1498    NA's   :1499   NA's   :1498   NA's   :1499
##  chromium_100g  molybdenum_100g  iodine_100g   caffeine_100g
##  Mode:logical   Mode:logical    Min.   :0      Mode:logical
##  NA's:1500      NA's:1500       1st Qu.:0      NA's:1500
##                                 Median :0
##                                 Mean   :0
##                                 3rd Qu.:0
##                                 Max.   :0
##                                 NA's   :1499
##  taurine_100g   ph_100g        fruits_vegetables_nuts_100g
##  Mode:logical   Mode:logical   Min.   : 2.00
##  NA's:1500      NA's:1500      1st Qu.:11.25
##                                Median :42.00
##                                Mean   :36.88
##                                3rd Qu.:52.25
##                                Max.   :80.00
##                                NA's   :1470
##  collagen_meat_protein_ratio_100g   cocoa_100g   chlorophyl_100g
##  Min.   :12.00                    Min.   :30     Mode:logical
##  1st Qu.:13.50                    1st Qu.:47     NA's:1500
##  Median :15.00                    Median :60
##  Mean   :15.67                    Mean   :57
##  3rd Qu.:17.50                    3rd Qu.:70
##  Max.   :20.00                    Max.   :81
##  NA's   :1497                     NA's   :1491
##  carbon_footprint_100g nutrition_score_fr_100g nutrition_score_uk_100g
##  Min.   : 12.00        Min.   :-12.000         Min.   :-12.000
##  1st Qu.: 97.42        1st Qu.:  1.000         1st Qu.:  0.000
##  Median :182.85        Median :  7.000         Median :  6.000
##  Mean   :131.18        Mean   :  7.941         Mean   :  7.631
##  3rd Qu.:190.78        3rd Qu.: 15.000         3rd Qu.: 16.000
##  Max.   :198.70        Max.   : 28.000         Max.   : 28.000
##  NA's   :1497          NA's   :825             NA's   :825
```

```
library(dplyr)
glimpse(df_food)
```

```
## Observations: 1,500
## Variables: 160
## $ V1                                         <int> 1, 2, 3, 4, 5, 6, 7...
## $ code                                       <int> 100030, 100050, 100...
## $ url                                        <chr> "http://world-en.op...
## $ creator                                    <chr> "sebleouf", "foodor...
## $ created_t                                  <int> 1424747544, 1450316...
## $ created_datetime                           <chr> "2015-02-24T03:12:2...
## $ last_modified_t                            <int> 1438445887, 1450817...
## $ last_modified_datetime                     <chr> "2015-08-01T16:18:0...
## $ product_name                               <chr> "Confiture de frais...
## $ generic_name                               <chr> "", "", "Pâtes de f...
## $ quantity                                   <chr> "265 g", "375g", "1...
## $ packaging                                  <chr> "Bocal,Verre", "Pla...
## $ packaging_tags                             <chr> "bocal,verre", "pla...
## $ brands                                     <chr> "Casino Délices", "...
## $ brands_tags                                <chr> "casino-delices", "...
## $ categories                                 <chr> "Aliments et boisso...
## $ categories_tags                            <chr> "en:plant-based-foo...
## $ categories_en                              <chr> "Plant-based foods ...
## $ origins                                    <chr> "", "", "", "", "Ar...
## $ origins_tags                               <chr> "", "", "", "", "ar...
## $ manufacturing_places                       <chr> "France", "Belgium"...
## $ manufacturing_places_tags                  <chr> "france", "belgium"...
## $ labels                                     <chr> "", "", "", "Vegeta...
## $ labels_tags                                <chr> "", "", "", "en:veg...
## $ labels_en                                  <chr> "", "", "", "Vegeta...
## $ emb_codes                                  <chr> "EMB 78015", "", ""...
## $ emb_codes_tags                             <chr> "emb-78015", "", ""...
## $ first_packaging_code_geo                   <chr> "48.983333,2.066667...
## $ cities                                     <lgl> NA, NA, NA, NA, NA,...
## $ cities_tags                                <chr> "andresy-yvelines-f...
## $ purchase_places                            <chr> "Lyon,France", "NSW...
## $ stores                                     <chr> "Casino", "", "", "...
## $ countries                                  <chr> "France", "Australi...
## $ countries_tags                             <chr> "en:france", "en:au...
## $ countries_en                               <chr> "France", "Australi...
## $ ingredients_text                           <chr> "Sucre de canne, fr...
## $ allergens                                  <chr> "", "", "", "", "",...
## $ allergens_en                               <lgl> NA, NA, NA, NA, NA,...
## $ traces                                     <chr> "Lait,Fruits à coqu...
## $ traces_tags                                <chr> "en:milk,en:nuts", ...
## $ traces_en                                  <chr> "Milk,Nuts", "", ""...
## $ serving_size                               <chr> "15 g", "", "", "",...
## $ no_nutriments                              <lgl> NA, NA, NA, NA, NA,...
## $ additives_n                                <int> 1, NA, 2, 5, 0, NA,...
## $ additives                                  <chr> "[ sucre-de-canne -...
## $ additives_tags                             <chr> "en:e440", "", "en:...
## $ additives_en                               <chr> "E440 - Pectins", "...
## $ ingredients_from_palm_oil_n                <int> 0, NA, 0, 0, 0, NA,...
## $ ingredients_from_palm_oil                  <lgl> NA, NA, NA, NA, NA,...
## $ ingredients_from_palm_oil_tags             <chr> "", "", "", "", "",...
## $ ingredients_that_may_be_from_palm_oil_n    <int> 0, NA, 0, 1, 0, NA,...
## $ ingredients_that_may_be_from_palm_oil      <lgl> NA, NA, NA, NA, NA,...
## $ ingredients_that_may_be_from_palm_oil_tags <chr> "", "", "", "e471-m...
## $ nutrition_grade_uk                         <lgl> NA, NA, NA, NA, NA,...
## $ nutrition_grade_fr                         <chr> "d", "", "", "d", "...
## $ pnns_groups_1                              <chr> "Sugary snacks", "S...
## $ pnns_groups_2                              <chr> "Sweets", "Chocolat...
## $ states                                     <chr> "en:to-be-checked, ...
## $ states_tags                                <chr> "en:to-be-checked,e...
## $ states_en                                  <chr> "To be checked,Comp...
## $ main_category                              <chr> "en:plant-based-foo...
## $ main_category_en                           <chr> "Plant-based foods ...
## $ image_url                                  <chr> "http://en.openfood...
## $ image_small_url                            <chr> "http://en.openfood...
## $ energy_100g                                <dbl> 918, NA, NA, 766, 2...
## $ energy_from_fat_100g                       <dbl> NA, NA, NA, NA, NA,...
## $ fat_100g                                   <dbl> 0.00, NA, NA, 16.70...
## $ saturated_fat_100g                         <dbl> 0.000, NA, NA, 9.90...
## $ butyric_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ caproic_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ caprylic_acid_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ capric_acid_100g                           <lgl> NA, NA, NA, NA, NA,...
## $ lauric_acid_100g                           <lgl> NA, NA, NA, NA, NA,...
## $ myristic_acid_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ palmitic_acid_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ stearic_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ arachidic_acid_100g                        <lgl> NA, NA, NA, NA, NA,...
## $ behenic_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ lignoceric_acid_100g                       <lgl> NA, NA, NA, NA, NA,...
## $ cerotic_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ montanic_acid_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ melissic_acid_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ monounsaturated_fat_100g                   <dbl> NA, NA, NA, 2.9, 9....
## $ polyunsaturated_fat_100g                   <dbl> NA, NA, NA, 3.9, 32...
## $ omega_3_fat_100g                           <dbl> NA, NA, NA, NA, NA,...
## $ alpha_linolenic_acid_100g                  <dbl> NA, NA, NA, NA, NA,...
## $ eicosapentaenoic_acid_100g                 <dbl> NA, NA, NA, NA, NA,...
## $ docosahexaenoic_acid_100g                  <dbl> NA, NA, NA, NA, NA,...
## $ omega_6_fat_100g                           <dbl> NA, NA, NA, NA, NA,...
## $ linoleic_acid_100g                         <dbl> NA, NA, NA, NA, NA,...
## $ arachidonic_acid_100g                      <lgl> NA, NA, NA, NA, NA,...
## $ gamma_linolenic_acid_100g                  <lgl> NA, NA, NA, NA, NA,...
## $ dihomo_gamma_linolenic_acid_100g           <lgl> NA, NA, NA, NA, NA,...
## $ omega_9_fat_100g                           <lgl> NA, NA, NA, NA, NA,...
## $ oleic_acid_100g                            <lgl> NA, NA, NA, NA, NA,...
## $ elaidic_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ gondoic_acid_100g                          <lgl> NA, NA, NA, NA, NA,...
## $ mead_acid_100g                             <lgl> NA, NA, NA, NA, NA,...
## $ erucic_acid_100g                           <lgl> NA, NA, NA, NA, NA,...
## $ nervonic_acid_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ trans_fat_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ cholesterol_100g                           <dbl> NA, NA, NA, 0.00020...
## $ carbohydrates_100g                         <dbl> 54.00, NA, NA, 5.70...
## $ sugars_100g                                <dbl> 54.00, NA, NA, 4.20...
## $ sucrose_100g                               <lgl> NA, NA, NA, NA, NA,...
## $ glucose_100g                               <lgl> NA, NA, NA, NA, NA,...
## $ fructose_100g                              <int> NA, NA, NA, NA, NA,...
## $ lactose_100g                               <dbl> NA, NA, NA, NA, NA,...
## $ maltose_100g                               <lgl> NA, NA, NA, NA, NA,...
## $ maltodextrins_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ starch_100g                                <dbl> NA, NA, NA, NA, NA,...
## $ polyols_100g                               <dbl> NA, NA, NA, NA, NA,...
## $ fiber_100g                                 <dbl> NA, NA, NA, 0.2, 9....
## $ proteins_100g                              <dbl> 0.00, NA, NA, 2.90,...
## $ casein_100g                                <dbl> NA, NA, NA, NA, NA,...
## $ serum_proteins_100g                        <lgl> NA, NA, NA, NA, NA,...
## $ nucleotides_100g                           <lgl> NA, NA, NA, NA, NA,...
## $ salt_100g                                  <dbl> 0.0000000, NA, NA, ...
## $ sodium_100g                                <dbl> 0.0000000, NA, NA, ...
## $ alcohol_100g                               <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_a_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ beta_carotene_100g                         <lgl> NA, NA, NA, NA, NA,...
## $ vitamin_d_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_e_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_k_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_c_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_b1_100g                            <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_b2_100g                            <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_pp_100g                            <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_b6_100g                            <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_b9_100g                            <dbl> NA, NA, NA, NA, NA,...
## $ vitamin_b12_100g                           <dbl> NA, NA, NA, NA, NA,...
## $ biotin_100g                                <dbl> NA, NA, NA, NA, NA,...
## $ pantothenic_acid_100g                      <dbl> NA, NA, NA, NA, NA,...
## $ silica_100g                                <dbl> NA, NA, NA, NA, NA,...
## $ bicarbonate_100g                           <dbl> NA, NA, NA, NA, NA,...
## $ potassium_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ chloride_100g                              <dbl> NA, NA, NA, NA, NA,...
## $ calcium_100g                               <dbl> NA, NA, NA, NA, NA,...
## $ phosphorus_100g                            <dbl> NA, NA, NA, NA, 1.1...
## $ iron_100g                                  <dbl> NA, NA, NA, NA, 0.0...
## $ magnesium_100g                             <dbl> NA, NA, NA, NA, 0.1...
## $ zinc_100g                                  <dbl> NA, NA, NA, NA, NA,...
## $ copper_100g                                <dbl> NA, NA, NA, NA, NA,...
## $ manganese_100g                             <dbl> NA, NA, NA, NA, NA,...
## $ fluoride_100g                              <dbl> NA, NA, NA, NA, NA,...
## $ selenium_100g                              <dbl> NA, NA, NA, NA, NA,...
## $ chromium_100g                              <lgl> NA, NA, NA, NA, NA,...
## $ molybdenum_100g                            <lgl> NA, NA, NA, NA, NA,...
## $ iodine_100g                                <dbl> NA, NA, NA, NA, NA,...
## $ caffeine_100g                              <lgl> NA, NA, NA, NA, NA,...
## $ taurine_100g                               <lgl> NA, NA, NA, NA, NA,...
## $ ph_100g                                    <lgl> NA, NA, NA, NA, NA,...
## $ fruits_vegetables_nuts_100g                <dbl> 54, NA, NA, NA, NA,...
## $ collagen_meat_protein_ratio_100g           <int> NA, NA, NA, NA, NA,...
## $ cocoa_100g                                 <int> NA, NA, NA, NA, NA,...
## $ chlorophyl_100g                            <lgl> NA, NA, NA, NA, NA,...
## $ carbon_footprint_100g                      <dbl> NA, NA, NA, NA, NA,...
## $ nutrition_score_fr_100g                    <int> 11, NA, NA, 11, 17,...
## $ nutrition_score_uk_100g                    <int> 11, NA, NA, 11, 17,...
```

```
# View column names of food
names(df_food)
```

```
##   [1] "V1"
##   [2] "code"
##   [3] "url"
##   [4] "creator"
##   [5] "created_t"
##   [6] "created_datetime"
##   [7] "last_modified_t"
##   [8] "last_modified_datetime"
##   [9] "product_name"
##  [10] "generic_name"
##  [11] "quantity"
##  [12] "packaging"
##  [13] "packaging_tags"
##  [14] "brands"
##  [15] "brands_tags"
##  [16] "categories"
##  [17] "categories_tags"
##  [18] "categories_en"
##  [19] "origins"
##  [20] "origins_tags"
##  [21] "manufacturing_places"
##  [22] "manufacturing_places_tags"
##  [23] "labels"
##  [24] "labels_tags"
##  [25] "labels_en"
##  [26] "emb_codes"
##  [27] "emb_codes_tags"
##  [28] "first_packaging_code_geo"
##  [29] "cities"
##  [30] "cities_tags"
##  [31] "purchase_places"
##  [32] "stores"
##  [33] "countries"
##  [34] "countries_tags"
##  [35] "countries_en"
##  [36] "ingredients_text"
##  [37] "allergens"
##  [38] "allergens_en"
##  [39] "traces"
##  [40] "traces_tags"
##  [41] "traces_en"
##  [42] "serving_size"
##  [43] "no_nutriments"
##  [44] "additives_n"
##  [45] "additives"
##  [46] "additives_tags"
##  [47] "additives_en"
##  [48] "ingredients_from_palm_oil_n"
##  [49] "ingredients_from_palm_oil"
##  [50] "ingredients_from_palm_oil_tags"
##  [51] "ingredients_that_may_be_from_palm_oil_n"
##  [52] "ingredients_that_may_be_from_palm_oil"
##  [53] "ingredients_that_may_be_from_palm_oil_tags"
##  [54] "nutrition_grade_uk"
##  [55] "nutrition_grade_fr"
##  [56] "pnns_groups_1"
##  [57] "pnns_groups_2"
##  [58] "states"
##  [59] "states_tags"
##  [60] "states_en"
##  [61] "main_category"
##  [62] "main_category_en"
##  [63] "image_url"
##  [64] "image_small_url"
##  [65] "energy_100g"
##  [66] "energy_from_fat_100g"
##  [67] "fat_100g"
##  [68] "saturated_fat_100g"
##  [69] "butyric_acid_100g"
##  [70] "caproic_acid_100g"
##  [71] "caprylic_acid_100g"
##  [72] "capric_acid_100g"
##  [73] "lauric_acid_100g"
##  [74] "myristic_acid_100g"
##  [75] "palmitic_acid_100g"
##  [76] "stearic_acid_100g"
##  [77] "arachidic_acid_100g"
##  [78] "behenic_acid_100g"
##  [79] "lignoceric_acid_100g"
##  [80] "cerotic_acid_100g"
##  [81] "montanic_acid_100g"
##  [82] "melissic_acid_100g"
##  [83] "monounsaturated_fat_100g"
##  [84] "polyunsaturated_fat_100g"
##  [85] "omega_3_fat_100g"
##  [86] "alpha_linolenic_acid_100g"
##  [87] "eicosapentaenoic_acid_100g"
##  [88] "docosahexaenoic_acid_100g"
##  [89] "omega_6_fat_100g"
##  [90] "linoleic_acid_100g"
##  [91] "arachidonic_acid_100g"
##  [92] "gamma_linolenic_acid_100g"
##  [93] "dihomo_gamma_linolenic_acid_100g"
##  [94] "omega_9_fat_100g"
##  [95] "oleic_acid_100g"
##  [96] "elaidic_acid_100g"
##  [97] "gondoic_acid_100g"
##  [98] "mead_acid_100g"
##  [99] "erucic_acid_100g"
## [100] "nervonic_acid_100g"
## [101] "trans_fat_100g"
## [102] "cholesterol_100g"
## [103] "carbohydrates_100g"
## [104] "sugars_100g"
## [105] "sucrose_100g"
## [106] "glucose_100g"
## [107] "fructose_100g"
## [108] "lactose_100g"
## [109] "maltose_100g"
## [110] "maltodextrins_100g"
## [111] "starch_100g"
## [112] "polyols_100g"
## [113] "fiber_100g"
## [114] "proteins_100g"
## [115] "casein_100g"
## [116] "serum_proteins_100g"
## [117] "nucleotides_100g"
## [118] "salt_100g"
## [119] "sodium_100g"
## [120] "alcohol_100g"
## [121] "vitamin_a_100g"
## [122] "beta_carotene_100g"
## [123] "vitamin_d_100g"
## [124] "vitamin_e_100g"
## [125] "vitamin_k_100g"
## [126] "vitamin_c_100g"
## [127] "vitamin_b1_100g"
## [128] "vitamin_b2_100g"
## [129] "vitamin_pp_100g"
## [130] "vitamin_b6_100g"
## [131] "vitamin_b9_100g"
## [132] "vitamin_b12_100g"
## [133] "biotin_100g"
## [134] "pantothenic_acid_100g"
## [135] "silica_100g"
## [136] "bicarbonate_100g"
## [137] "potassium_100g"
## [138] "chloride_100g"
## [139] "calcium_100g"
## [140] "phosphorus_100g"
## [141] "iron_100g"
## [142] "magnesium_100g"
## [143] "zinc_100g"
## [144] "copper_100g"
## [145] "manganese_100g"
## [146] "fluoride_100g"
## [147] "selenium_100g"
## [148] "chromium_100g"
## [149] "molybdenum_100g"
## [150] "iodine_100g"
## [151] "caffeine_100g"
## [152] "taurine_100g"
## [153] "ph_100g"
## [154] "fruits_vegetables_nuts_100g"
## [155] "collagen_meat_protein_ratio_100g"
## [156] "cocoa_100g"
## [157] "chlorophyl_100g"
## [158] "carbon_footprint_100g"
## [159] "nutrition_score_fr_100g"
## [160] "nutrition_score_uk_100g"
```

Removing Duplicates

A vector has been created for you that lists out all of the duplicates; all you need to do is remove those columns from the dataset.

```r
# Define vector of duplicate cols
duplicates <- c(4, 6, 11, 13, 15, 17, 18, 20, 22,
                24, 25, 28, 32, 34, 36, 38, 40,
                44, 46, 48, 51, 54, 65, 158)

# Remove duplicates from food: food2
food2<-df_food[,-duplicates]
```

Removing Useless Info

```r
# Define useless vector
useless <- c(1, 2, 3, 32:41)

# Remove useless columns from food2: food3
food3<-food2[,-useless]
```

Finding columns

Looking much nicer! Recall from the first exercise that you are assuming you will be analyzing the sugar content of these foods. Therefore, your next step is to look at a summary of the nutrition information.

All of the columns with nutrition info contain the character string “100g” as part of their name, which makes it easy to identify them.

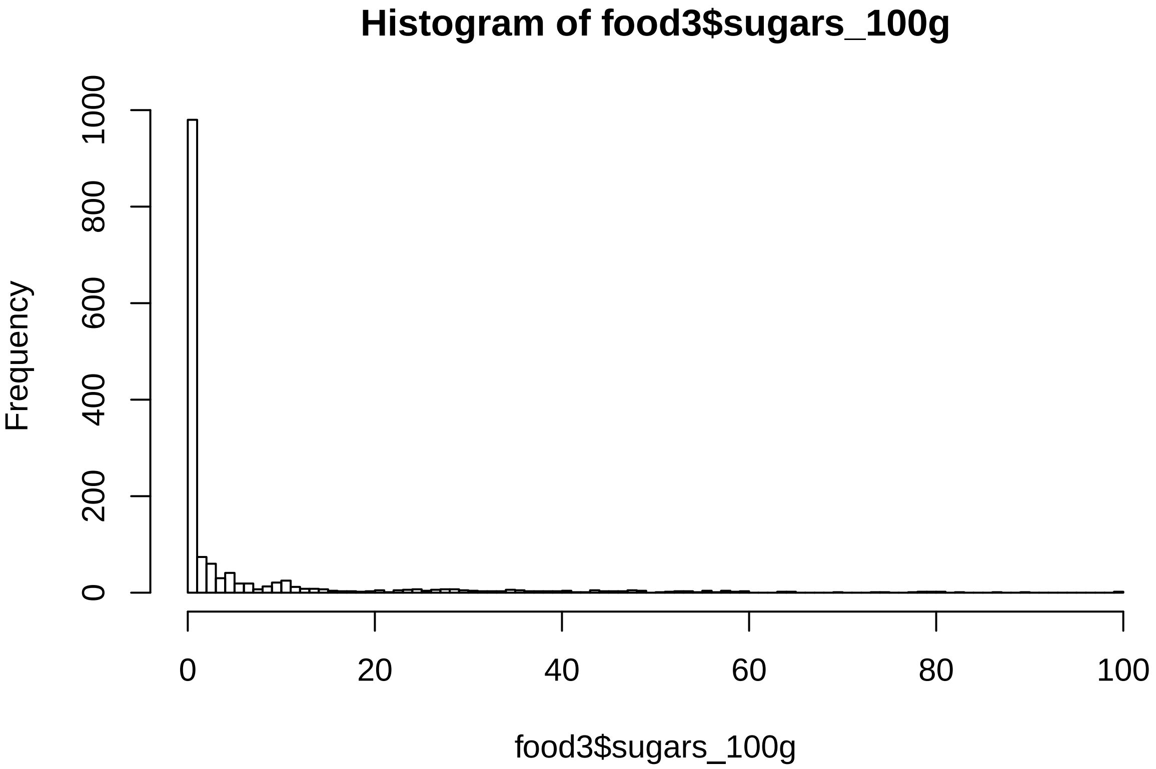
```r
#Create a vector called nutrition containing the column indices of the nutrition data. To do this, use

library(stringr)
nutrition <- str_detect(names(food3), pattern = "100g")

#View a summary of the nutrition columns.
summary(food3[,nutrition])
```

```
##  energy_from_fat_100g    fat_100g      saturated_fat_100g
##  Min.   :   0.00      Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  35.98      1st Qu.:  0.90   1st Qu.: 0.200
##  Median : 237.00      Median :  6.00   Median : 1.700
##  Mean   : 668.41      Mean   : 13.39   Mean   : 4.874
##  3rd Qu.: 974.00      3rd Qu.: 20.00   3rd Qu.: 6.500
##  Max.   :2900.00      Max.   :100.00   Max.   :57.000
##  NA's   :1486         NA's   :708      NA's   :797
##  butyric_acid_100g caproic_acid_100g caprylic_acid_100g capric_acid_100g
##  Mode:logical      Mode:logical      Mode:logical       Mode:logical
##  NA's:1500         NA's:1500         NA's:1500          NA's:1500
##
##
##
##
##
##  lauric_acid_100g myristic_acid_100g palmitic_acid_100g stearic_acid_100g
##  Mode:logical     Mode:logical       Mode:logical       Mode:logical
##  NA's:1500        NA's:1500          NA's:1500          NA's:1500
##
##
##
##
##
##  arachidic_acid_100g behenic_acid_100g lignoceric_acid_100g
##  Mode:logical        Mode:logical      Mode:logical
##  NA's:1500           NA's:1500         NA's:1500
##
##
##
##
##
##  cerotic_acid_100g montanic_acid_100g melissic_acid_100g
##  Mode:logical      Mode:logical       Mode:logical
##  NA's:1500         NA's:1500          NA's:1500
##
##
##
##
##
##  monounsaturated_fat_100g polyunsaturated_fat_100g omega_3_fat_100g
##  Min.   : 0.00            Min.   : 0.400           Min.   : 0.033
##  1st Qu.: 3.87            1st Qu.: 1.653           1st Qu.: 1.300
##  Median : 9.50            Median : 3.900           Median : 3.000
##  Mean   :19.77            Mean   : 9.986           Mean   : 3.726
##  3rd Qu.:29.00            3rd Qu.:12.700           3rd Qu.: 3.200
##  Max.   :75.00            Max.   :46.200           Max.   :12.400
##  NA's   :1465             NA's   :1464             NA's   :1491
##  alpha_linolenic_acid_100g eicosapentaenoic_acid_100g
##  Min.   :0.0800            Min.   :0.721
##  1st Qu.:0.0905            1st Qu.:0.721
##  Median :0.1010            Median :0.721
##  Mean   :0.1737            Mean   :0.721
##  3rd Qu.:0.2205            3rd Qu.:0.721
##  Max.   :0.3400            Max.   :0.721
##  NA's   :1497              NA's   :1499
##  docosahexaenoic_acid_100g omega_6_fat_100g linoleic_acid_100g
##  Min.   :1.09              Min.   :0.25     Min.   :0.5000
##  1st Qu.:1.09              1st Qu.:0.25     1st Qu.:0.5165
##  Median :1.09              Median :0.25     Median :0.5330
##  Mean   :1.09              Mean   :0.25     Mean   :0.5330
##  3rd Qu.:1.09              3rd Qu.:0.25     3rd Qu.:0.5495
##  Max.   :1.09              Max.   :0.25     Max.   :0.5660
##  NA's   :1499              NA's   :1499     NA's   :1498
##  arachidonic_acid_100g gamma_linolenic_acid_100g
##  Mode:logical          Mode:logical
##  NA's:1500             NA's:1500
##
##
##
##
##
##  dihomo_gamma_linolenic_acid_100g omega_9_fat_100g oleic_acid_100g
##  Mode:logical                     Mode:logical     Mode:logical
##  NA's:1500                        NA's:1500        NA's:1500
##
##
##
##
##
##  elaidic_acid_100g gondoic_acid_100g mead_acid_100g erucic_acid_100g
##  Mode:logical      Mode:logical      Mode:logical   Mode:logical
##  NA's:1500         NA's:1500         NA's:1500      NA's:1500
##
##
##
##
##
##  nervonic_acid_100g trans_fat_100g   cholesterol_100g carbohydrates_100g
##  Mode:logical       Min.   :0.0000   Min.   :0.0000   Min.   :  0.000
##  NA's:1500          1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  3.792
##                     Median :0.0000   Median :0.0000   Median : 13.500
##                     Mean   :0.0105   Mean   :0.0265   Mean   : 27.958
##                     3rd Qu.:0.0000   3rd Qu.:0.0026   3rd Qu.: 55.000
##                     Max.   :0.1000   Max.   :0.4300   Max.   :100.000
##                     NA's   :1481     NA's   :1477     NA's   :708
##   sugars_100g     sucrose_100g   glucose_100g   fructose_100g
##  Min.   :  0.00   Mode:logical   Mode:logical   Min.   :100
##  1st Qu.:  1.00   NA's:1500      NA's:1500      1st Qu.:100
##  Median :  4.05                                 Median :100
##  Mean   : 12.66                                 Mean   :100
##  3rd Qu.: 14.70                                 3rd Qu.:100
##  Max.   :100.00                                 Max.   :100
##  NA's   :788                                    NA's   :1499
##   lactose_100g   maltose_100g   maltodextrins_100g  starch_100g
##  Min.   :0.000   Mode:logical   Mode:logical       Min.   : 0.00
##  1st Qu.:0.250   NA's:1500      NA's:1500          1st Qu.: 9.45
##  Median :0.500                                     Median :39.50
##  Mean   :2.933                                     Mean   :30.73  
##  3rd Qu.:4.400                                     3rd Qu.:42.85  
##  Max.   :8.300                                     Max.   :71.00  
##  NA's   :1497                                      NA's   :1493  
##   polyols_100g     fiber_100g     proteins_100g     casein_100g 
##  Min.   : 8.60   Min.   : 0.000   Min.   : 0.000   Min.   :1.1  
##  1st Qu.:59.10   1st Qu.: 0.500   1st Qu.: 1.500   1st Qu.:1.1  
##  Median :67.00   Median : 1.750   Median : 6.000   Median :1.1  
##  Mean   :56.06   Mean   : 2.823   Mean   : 7.563   Mean   :1.1  
##  3rd Qu.:69.80   3rd Qu.: 3.500   3rd Qu.:10.675   3rd Qu.:1.1  
##  Max.   :70.00   Max.   :46.700   Max.   :61.000   Max.   :1.1  
##  NA's   :1491    NA's   :994      NA's   :710      NA's   :1499  
##  serum_proteins_100g nucleotides_100g   salt_100g         sodium_100g     
##  Mode:logical        Mode:logical     Min.   :  0.0000   Min.   : 0.0000  
##  NA's:1500           NA's:1500        1st Qu.:  0.0438   1st Qu.: 0.0172  
##                                       Median :  0.4498   Median : 0.1771  
##                                       Mean   :  1.1205   Mean   : 0.4409  
##                                       3rd Qu.:  1.1938   3rd Qu.: 0.4700  
##                                       Max.   :102.0000   Max.   :40.0000  
##                                       NA's   :780        NA's   :780  
##   alcohol_100g   vitamin_a_100g   beta_carotene_100g vitamin_d_100g 
##  Min.   : 0.00   Min.   :0.0000   Mode:logical       Min.   :0e+00  
##  1st Qu.: 0.00   1st Qu.:0.0000   NA's:1500          1st Qu.:0e+00  
##  Median : 5.50   Median :0.0001                      Median :0e+00  
##  Mean   :10.07   Mean   :0.0003                      Mean   :0e+00  
##  3rd Qu.:13.00   3rd Qu.:0.0006                      3rd Qu.:0e+00  
##  Max.   :50.00   Max.   :0.0013                      Max.   :1e-04  
##  NA's   :1433    NA's   :1477                        NA's   :1485  
##  vitamin_e_100g   vitamin_k_100g vitamin_c_100g  vitamin_b1_100g 
##  Min.   :0.0005   Min.   :0      Min.   :0.000   Min.   :0.0001  
##  1st Qu.:0.0021   1st Qu.:0      1st Qu.:0.002   1st Qu.:0.0003  
##  Median :0.0044   Median :0      Median :0.019   Median :0.0004  
##  Mean   :0.0069   Mean   :0      Mean   :0.025   Mean   :0.0006  
##  3rd Qu.:0.0097   3rd Qu.:0      3rd Qu.:0.030   3rd Qu.:0.0010  
##  Max.   :0.0320   Max.   :0      Max.   :0.217   Max.   :0.0013  
##  NA's   :1478     NA's   :1498   NA's   :1459    NA's   :1478  
##  vitamin_b2_100g  vitamin_pp_100g  vitamin_b6_100g  vitamin_b9_100g
##  Min.   :0.0002   Min.   :0.0006   Min.   :0.0001   Min.   :0e+00  
##  1st Qu.:0.0003   1st Qu.:0.0033   1st Qu.:0.0002   1st Qu.:0e+00  
##  Median :0.0009   Median :0.0069   Median :0.0008   Median :1e-04  
##  Mean   :0.0011   Mean   :0.0086   Mean   :0.0112   Mean   :1e-04  
##  3rd Qu.:0.0013   3rd Qu.:0.0140   3rd Qu.:0.0012   3rd Qu.:2e-04  
##  Max.   :0.0066   Max.   :0.0160   Max.   :0.2000   Max.   :2e-04  
##  NA's   :1483     NA's   :1484     NA's   :1481     NA's   :1483  
##  vitamin_b12_100g  biotin_100g   pantothenic_acid_100g  silica_100g   
##  Min.   :0        Min.   :0      Min.   :0.0000        Min.   :8e-04  
##  1st Qu.:0        1st Qu.:0      1st Qu.:0.0007        1st Qu.:8e-04  
##  Median :0        Median :0      Median :0.0020        Median :8e-04  
##  Mean   :0        Mean   :0      Mean   :0.0027        Mean   :8e-04  
##  3rd Qu.:0        3rd Qu.:0      3rd Qu.:0.0051        3rd Qu.:8e-04  
##  Max.   :0        Max.   :0      Max.   :0.0060        Max.   :8e-04  
##  NA's   :1489     NA's   :1498   NA's   :1486          NA's   :1499  
##  bicarbonate_100g potassium_100g   chloride_100g     calcium_100g   
##  Min.   :0.0006   Min.   :0.0000   Min.   :0.0003   Min.   :0.0000  
##  1st Qu.:0.0678   1st Qu.:0.0650   1st Qu.:0.0006   1st Qu.:0.0450  
##  Median :0.1350   Median :0.1940   Median :0.0009   Median :0.1200  
##  Mean   :0.1692   Mean   :0.3288   Mean   :0.0144   Mean   :0.2040  
##  3rd Qu.:0.2535   3rd Qu.:0.3670   3rd Qu.:0.0214   3rd Qu.:0.1985  
##  Max.   :0.3720   Max.   :1.4300   Max.   :0.0420   Max.   :1.0000  
##  NA's   :1497     NA's   :1487     NA's   :1497     NA's   :1449    
##  phosphorus_100g    iron_100g      magnesium_100g     zinc_100g     
##  Min.   :0.0430   Min.   :0.0000   Min.   :0.0000   Min.   :0.0005  
##  1st Qu.:0.1938   1st Qu.:0.0012   1st Qu.:0.0670   1st Qu.:0.0009  
##  Median :0.3185   Median :0.0042   Median :0.1040   Median :0.0017  
##  Mean   :0.3777   Mean   :0.0045   Mean   :0.1066   Mean   :0.0016  
##  3rd Qu.:0.4340   3rd Qu.:0.0077   3rd Qu.:0.1300   3rd Qu.:0.0022  
##  Max.   :1.1550   Max.   :0.0137   Max.   :0.3330   Max.   :0.0026  
##  NA's   :1488     NA's   :1463     NA's   :1479     NA's   :1493    
##   copper_100g    manganese_100g fluoride_100g  selenium_100g 
##  Min.   :0e+00   Min.   :0      Min.   :0      Min.   :0     
##  1st Qu.:1e-04   1st Qu.:0      1st Qu.:0      1st Qu.:0     
##  Median :1e-04   Median :0      Median :0      Median :0     
##  Mean   :1e-04   Mean   :0      Mean   :0      Mean   :0     
##  3rd Qu.:1e-04   3rd Qu.:0      3rd Qu.:0      3rd Qu.:0     
##  Max.   :1e-04   Max.   :0      Max.   :0      Max.   :0     
##  NA's   :1498    NA's   :1499   NA's   :1498   NA's   :1499  
##  chromium_100g  molybdenum_100g  iodine_100g   caffeine_100g 
##  Mode:logical   Mode:logical    Min.   :0      Mode:logical  
##  NA's:1500      NA's:1500       1st Qu.:0      NA's:1500     
##                                 Median :0                    
##                                 Mean   :0                    
##                                 3rd Qu.:0                    
##                                 Max.   :0                    
##                                 NA's   :1499                 
##  taurine_100g   ph_100g        fruits_vegetables_nuts_100g
##  Mode:logical   Mode:logical   Min.   : 2.00              
##  NA's:1500      NA's:1500      1st Qu.:11.25              
##                                Median :42.00              
##                                Mean   :36.88              
##                                3rd Qu.:52.25              
##                                Max.   :80.00              
##                                NA's   :1470               
##  collagen_meat_protein_ratio_100g   cocoa_100g   chlorophyl_100g
##  Min.   :12.00                    Min.   :30     Mode:logical   
##  1st Qu.:13.50                    1st Qu.:47     NA's:1500      
##  Median :15.00                    Median :60                    
##  Mean   :15.67                    Mean   :57                    
##  3rd Qu.:17.50                    3rd Qu.:70                    
##  Max.   :20.00                    Max.   :81                    
##  NA's   :1497                     NA's   :1491                  
##  nutrition_score_fr_100g nutrition_score_uk_100g
##  Min.   :-12.000         Min.   :-12.000        
##  1st Qu.:  1.000         1st Qu.:  0.000        
##  Median :  7.000         Median :  6.000        
##  Mean   :  7.941         Mean   :  7.631        
##  3rd Qu.: 15.000         3rd Qu.: 16.000        
##  Max.   : 28.000         Max.   : 28.000        
##  NA's   :825             NA's   :825            
```

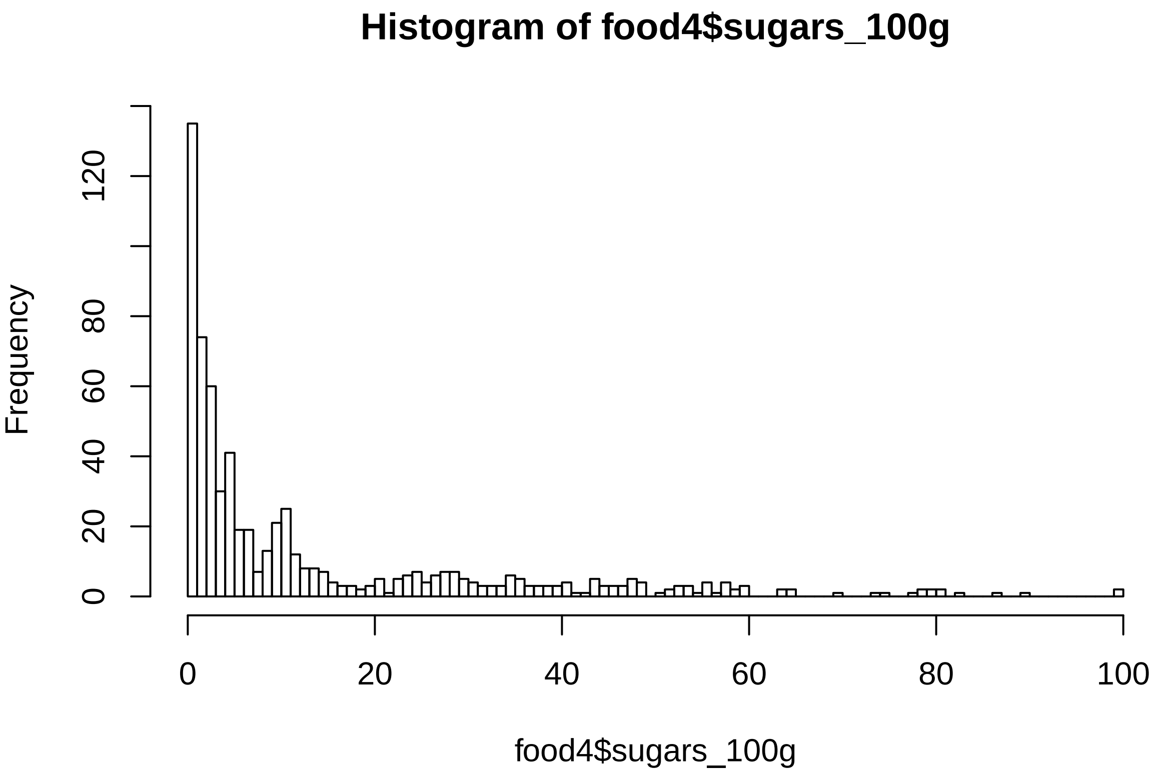
Replacing missing values

In this exercise, you'll replace all NA values with zeroes in the sugars_100g column and make histograms to visualize the result. Then, you will exclude the observations which have no sugar to see how the distribution changes.

```
# Find indices of sugar NA values: missing
missing <- is.na(food3$sugars_100g)

# Replace NA values with 0
food3$sugars_100g[missing] <- 0

# Create first histogram
hist(food3$sugars_100g, breaks = 100)
```



```
# Create food4
food4 <- food3[food3$sugars_100g > 0, ]

# Create second histogram
hist(food4$sugars_100g, breaks = 100)
```



```r
#To get a general idea of how many of these foods are packaged in plastic, you can look through the pac

# Find entries containing "plasti": plastic
plastic <- str_detect(food3$packaging, "plasti")

# Print the sum of plastic
sum(plastic)
```

```
## [1] 232
```

Exercise 4_PublicSchools_Attendance

Data_Sales: https://www.datacamp.com/courses/importing-cleaning-data-in-r-case-studies Data_file: attendance.xls

In this chapter, you'll work with attendance data from public schools in the US, organized by school level and state, during the 2007-2008 academic year. The data contain information on average daily attendance (ADA) as a percentage of total enrollment, school day length, and school year length.

Importing

```r
# Load the gdata package
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

##

## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'

## The following objects are masked from 'package:data.table':
##
##     first, last

## The following objects are masked from 'package:dplyr':
##
##     combine, first, last

## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith
```

```
# Import the spreadsheet: att
att<- read.xls("attendance.xls")
```

Examining the data

```
str(att)
```

```
## 'data.frame':    59 obs. of  17 variables:
##  $ Table.43..Average.daily.attendance..ADA..as.a.percentage.of.total.enrollment..school.day.length..a
##  $ X
##  $ X.1
##  $ X.2
##  $ X.3
##  $ X.4
##  $ X.5
##  $ X.6
##  $ X.7
##  $ X.8
##  $ X.9
##  $ X.10
##  $ X.11
##  $ X.12
##  $ X.13
##  $ X.14
##  $ X.15
```

These are some messy data! The column names are mostly missing, there are irrelevant notes at the end of the data frame, and it looks like the numeric data were imported as factors. Let's start the cleaning process!

Removing unnecessary rows

When you're importing a messy spreadsheet into R, it's good practice to compare the original spreadsheet with what you've imported. It turns out that, by default, the read.xls() function skips empty rows such as the 11th and 17th.

After viewing your data frame, you realize you still need to get rid of the third row of att, as well as rows 56 through 59.

```
# Create remove
remove<-c(3,56:59)

# Create att2
att2<- att[-(remove),]
```

Removing useless columns

Once more, for reference, here is an image of the first 22 rows of the original spreadsheet. You can see here that the columns 3, 5, 7, 9, 11, 13, 15, and 17 (or columns C, E, G, I, K, M, O, Q in Excel) don't contain the values of average daily attendance (ADA). You'll get rid of them in this exercise.

```
# Create remove
remove<-c(3, 5, 7, 9, 11, 13, 15, 17)

# Create att3
att3<- att2[,-remove]
```

Splitting the data

In many cases, a single data frame stores multiple "tables" of information. You can often diagnose this problem by looking at the column names and noticing duplicate rows.

In this data frame, columns 1, 6, and 7 represent attendance data for US elementary schools, columns 1, 8, and 9 represent data for secondary schools, and columns 1 through 5 represent data for all schools in the US.

Each of these should be stored as its own separate data frame, so you'll split them up here.

```
#Subset att3 to include only data for elementary schools (columns 1, 6, and 7). Name the resulting data

att_elem<- att3[,c(1,6,7)]

#Subset att3 to include only data for secondary schools (columns 1, 8, and 9). Name the resulting data 

att_sec<- att3[,c(1,8,9)]

#Subset att3 to include data for all schools (columns 1 through 5). Name the resulting data frame att4.
att4<- att3[, c(1:5)]
```

Replacing the names

Since you went through so much trouble finding out which row stored the variable names, you should store that row as the actual column names of the data frame. We've modified the names a bit in order to be more stylistically sound; they're stored as cnames in the editor.

This will also allow you to remove the first two rows (currently storing variable names).

```
# Define cnames vector (don't change)
cnames <- c("state", "avg_attend_pct", "avg_hr_per_day",
            "avg_day_per_yr", "avg_hr_per_yr")
```

```r
# Assign column names of att4
colnames(att4) <- cnames

# Remove first two rows of att4: att5
att5<- att4[-c(1,2),]

# View the names of att5
names(att5)
```

```
## [1] "state"          "avg_attend_pct" "avg_hr_per_day" "avg_day_per_yr"
## [5] "avg_hr_per_yr"
```

Cleaning up extra characters

One of the most irritating things about this dataset is that the state names are all stored as the same number of characters, with periods padding the ends of the shorter states. That may be helpful for reading the spreadsheet, but it makes your life harder, so you'll deal with it in this exercise.

One pitfall to avoid: . is a special character in the language of regular expressions (a.k.a. regex). In order to specify that you actually want to remove periods and not their regex equivalent (which is "all characters"), use \.. This is called an "escape" sequence.

```r
library(stringr)

#Use the function str_replace_all() to replace all periods in the state column of att5 with "". Remembe

att5$state<- str_replace_all(att5$state,pattern="\\.", "")

#Remove white space around the state names, assigning the result back to att5$state once more. There's
att5$state <- str_trim(att5$state)
head(att5, n=20)
```

```
##                    state avg_attend_pct avg_hr_per_day avg_day_per_yr
## 4          United States           93.1            6.6            180
## 5                Alabama           93.8            7.0            180
## 6                 Alaska           89.9            6.5            180
## 7                Arizona           89.0            6.4            181
## 8               Arkansas           91.8            6.9            179
## 9             California           93.2            6.2            181
## 10              Colorado           93.9            7.0            171
## 11           Connecticut           87.9            6.5            181
## 12              Delaware           89.8            6.7            181
## 13  District of Columbia           91.2            6.9            181
## 14               Florida           92.7            6.4            184
## 15               Georgia           93.3            6.8            181
## 16                Hawaii           90.7            6.3            179
## 17                 Idaho           92.4            6.6            173
## 18              Illinois           94.0            6.5            177
## 19               Indiana           95.7            6.8            180
## 20                  Iowa           94.8            6.9            180
## 21                Kansas           95.4            7.0            178
## 22              Kentucky           93.1            6.7            180
## 23             Louisiana           90.3            7.1            178
##    avg_hr_per_yr
## 4           1,193
## 5           1,267
## 6           1,163
## 7           1,159
## 8           1,229
## 9           1,129
## 10          1,199
## 11          1,173
## 12          1,208
## 13          1,256
## 14          1,184
## 15          1,229
## 16          1,118
## 17          1,143
## 18          1,147
## 19          1,222
## 20          1,232
## 21          1,240
## 22          1,202
## 23          1,263
```

———————FIN————————