Chapitre 4: Starting with data

Armelle de le Court

26/04/2021

1: Presentation of the survey data

Dataset stored as a comma separated value (=CSV) file.

If we want to download a file :download.file(url="https://ndownloader.figshare.com/files/2292169",destfile = "data/portal_data_joined.csv")

But I already have it on my data so I just need do to this:

```
surveys <- read.csv("data/portal_data_joined.csv")
head(surveys) #to see the firsy 6 lines
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2        72     8  19 1977       2         NL   M              31     NA
## 3       224     9  13 1977       2         NL                  NA     NA
## 4       266    10  16 1977       2         NL                  NA     NA
## 5       349    11  12 1977       2         NL                  NA     NA
## 6       363    11  12 1977       2         NL                  NA     NA
##     genus  species   taxa plot_type
## 1 Neotoma albigula Rodent   Control
## 2 Neotoma albigula Rodent   Control
## 3 Neotoma albigula Rodent   Control
## 4 Neotoma albigula Rodent   Control
## 5 Neotoma albigula Rodent   Control
## 6 Neotoma albigula Rodent   Control
```

2: What are data frames ?

Its a representation of data in the format of a table where the columns are vectors that all have the same lenght.Columns are vectors, and each column must contain a single type of data.

To inspect the structure of a data frame:

```
str(surveys)
```

```
## 'data.frame':    34786 obs. of  13 variables:
##  $ record_id      : int  1 72 224 266 349 363 435 506 588 661 ...
##  $ month          : int  7 8 9 10 11 11 12 1 2 3 ...
```

```
##  $ day            : int  16 19 13 16 12 12 10 8 18 11 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1978 1978 1978 ...
##  $ plot_id        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ species_id     : chr  "NL" "NL" "NL" "NL" ...
##  $ sex            : chr  "M" "M" "" "" ...
##  $ hindfoot_length: int  32 31 NA NA NA NA NA NA NA NA ...
##  $ weight         : int  NA NA NA NA NA NA NA NA 218 NA ...
##  $ genus          : chr  "Neotoma" "Neotoma" "Neotoma" "Neotoma" ...
##  $ species        : chr  "albigula" "albigula" "albigula" "albigula" ...
##  $ taxa           : chr  "Rodent" "Rodent" "Rodent" "Rodent" ...
##  $ plot_type      : chr  "Control" "Control" "Control" "Control" ...
```

3: Inspectinf data.frame objects

```
dim(surveys)
nrow(surveys)
ncol(surveys)
head(surveys)
tail(surveys)
names(surveys)
rownames(surveys)
summary(surveys)
```

Question

```
#class of object surveys?
class(surveys) #its a data.frame
```

```
## [1] "data.frame"
```

```
#How many rows and how many columns are in this object?
ncol(surveys)
```

```
## [1] 13
```

```
nrow(surveys)
```

```
## [1] 34786
```

```
#How many species (as defined by the species_id variable)
length(unique(surveys$species_id))
```

```
## [1] 48
```

4: Indexing and subsetting data frames

If we want to extract some specific data from it, we need to specify the coordinates we want from it. Row number first, and then col numbers.

```
# first element in the first column of the data frame (as a vector)
surveys[1, 1]
```

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

```r
# first element in the 6th column (as a vector)
surveys[1, 6]
```

```
## [1] "NL"
```

```r
# first column of the data frame (as a vector)
surveys[, 1]
# first column of the data frame (as a data.frame)
surveys[1]
```

```r
# first three elements in the 7th column (as a vector)
surveys[1:3, 7]
```

```
## [1] "M" "M" ""
```

```r
# the 3rd row of the data frame (as a data.frame)
surveys[3, ]
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 3       224     9  13 1977       2         NL                  NA     NA
##     genus  species   taxa plot_type
## 3 Neotoma albigula Rodent   Control
```

```r
# equivalent to head_surveys <- head(surveys)
head_surveys <- surveys[1:6, ]
```

We can also exclude indices using "-" sign :

```r
surveys[, -1] ## The whole data frame, except the first column
surveys[-c(7:34786), ] ## Equivalent to head(surveys)
```

subset by calling indices, but by their names:

```r
surveys["species_id"] # Result is a data.frame
surveys[, "species_id"] # Result is a vector
surveys[["species_id"]] # Result is a vector
surveys$species_id # Result is a vector
```

Question

```r
#QUESTION1
#data.frame (surveys_200) containing only the data in row 200 of the surveys dataset

surveys_200 <- surveys[200,]
surveys_200
```

```
##     record_id month day year plot_id species_id sex hindfoot_length weight
## 200     35212    12   7 2002       2         NL   M              33    248
##       genus  species   taxa plot_type
## 200 Neotoma albigula Rodent   Control
```

```
#QUESTION2 Notice how nrow() gave you the number of rows in a data.frame? Use that number to pull out j

nrow(surveys) #34786
```

```
## [1] 34786
```

```
last_row <- surveys[34786,]
last_row
```

```
##       record_id month day year plot_id species_id sex hindfoot_length weight
## 34786     30986     7   1 2000       7         PX                  NA     NA
##             genus species   taxa        plot_type
## 34786 Chaetodipus     sp. Rodent Rodent Exclosure
```

```
#Compare that with what you see as the last row using tail()
tail(surveys)
```

```
##       record_id month day year plot_id species_id sex hindfoot_length weight
## 34781     26787     9  27 1997       7         PL   F              21     16
## 34782     26966    10  25 1997       7         PL   M              20     16
## 34783     27185    11  22 1997       7         PL   F              21     22
## 34784     27792     5   2 1998       7         PL   F              20      8
## 34785     28806    11  21 1998       7         PX                  NA     NA
## 34786     30986     7   1 2000       7         PX                  NA     NA
##             genus  species   taxa        plot_type
## 34781  Peromyscus leucopus Rodent Rodent Exclosure
## 34782  Peromyscus leucopus Rodent Rodent Exclosure
## 34783  Peromyscus leucopus Rodent Rodent Exclosure
## 34784  Peromyscus leucopus Rodent Rodent Exclosure
## 34785 Chaetodipus      sp. Rodent Rodent Exclosure
## 34786 Chaetodipus      sp. Rodent Rodent Exclosure
```

```
#QUESTION3
#Use nrow() to extract the row that is in the middle of the surveys dataframe.Store the content of this

n_rows <- nrow(surveys)
surveys_middle <- surveys[n_rows/2,]
surveys_middle
```

```
##       record_id month day year plot_id species_id sex hindfoot_length weight
## 17393      9828     1  19 1985      14         AB                  NA     NA
##            genus   species taxa plot_type
## 17393 Amphispiza bilineata Bird   Control
```

```
#QUESTION4
#Combine nrow() with the - notation above to reproduce the behavior of head(surveys), keeping just the

surveys_head <- surveys[-(7:n_rows), ]
surveys_head
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2        72     8  19 1977       2         NL   M              31     NA
## 3       224     9  13 1977       2         NL                  NA     NA
## 4       266    10  16 1977       2         NL                  NA     NA
## 5       349    11  12 1977       2         NL                  NA     NA
## 6       363    11  12 1977       2         NL                  NA     NA
```

```
##     genus  species   taxa plot_type
## 1 Neotoma albigula Rodent   Control
## 2 Neotoma albigula Rodent   Control
## 3 Neotoma albigula Rodent   Control
## 4 Neotoma albigula Rodent   Control
## 5 Neotoma albigula Rodent   Control
## 6 Neotoma albigula Rodent   Control
```

5: Factors

Stored as integers associated with labels and they can be ordererd or unordered. By default R always sort levels in alphabetical order.

```
sex <- factor(c("male", "female", "female", "male"))
levels(sex) #level 1 is female bc f comes before m.
```

```
## [1] "female" "male"
```

```
nlevels(sex)
```

```
## [1] 2
```

```
sex
```

```
## [1] male   female female male
## Levels: female male
```

```
# to reorder:
sex <- factor(sex, levels= c("male","female"))
sex # it has been reorder
```

```
## [1] male   female female male
## Levels: male female
```

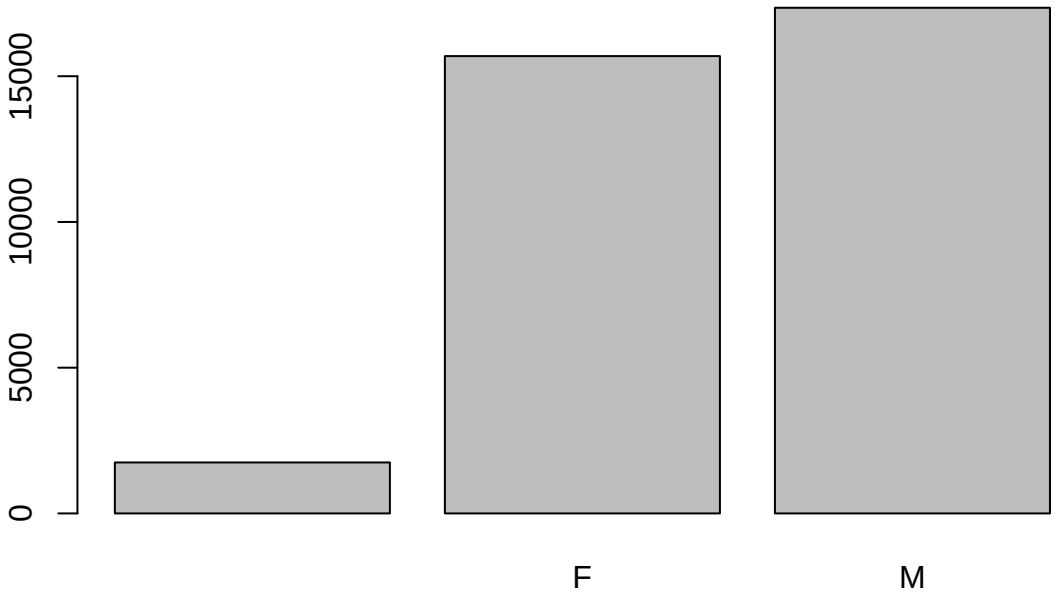
If we need to convert a factor to a character

```
as.character(sex)
```

```
## [1] "male"   "female" "female" "male"
```

If we need do rename factors

```
surv_sex <- factor(surveys$sex)
plot(surv_sex)
```




```
#about 1700 individuals hasnt been recorded for the sex variable. We need to rename that label
head(surv_sex)
```

```
## [1] M M
## Levels:  F M
```

```
levels(surv_sex)
```

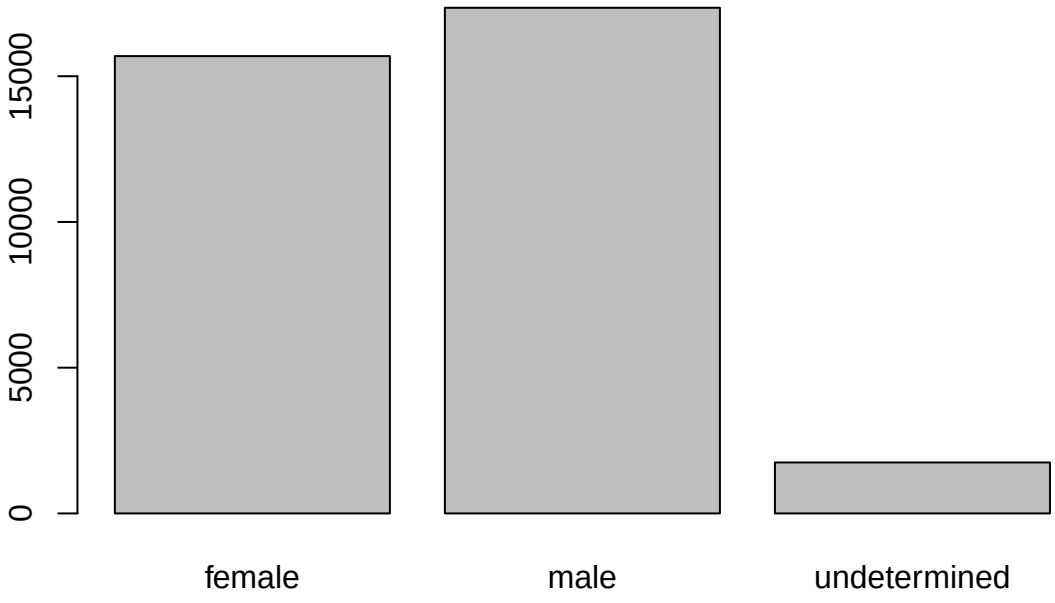
```
## [1] ""  "F" "M"
```

```
levels(surv_sex)[1] <- "undetermined"
levels(surv_sex) # it now has a name
```

```
## [1] "undetermined" "F"            "M"
```

Question

```
#rename F and M to male and female + barplot
levels(surv_sex)[2:3] <- c("female", "male")
surv_sex <- factor(surv_sex, levels = c("female", "male", "undetermined"))
plot(surv_sex)
```



6: Matrices

We want to generate a matrix containing info about all packages installed

```
ip <- installed.packages()
dim(ip)
```

```
## [1] 358  16
```

```
head(colnames(ip))
```

```
## [1] "Package"  "LibPath"  "Version"  "Priority" "Depends"  "Imports"
```

```
head(rownames(ip))
```

```
## [1] "abind"    "affy"     "affyio"   "airway"   "ALL"      "annotate"
```

Question: is lubridate installed?

```
## 1. get the names of all installed packages
ip <- rownames(installed.packages())
## 2. TRUE if installed, FALSE otherwise
"lubridate" %in% ip
```

```
## [1] TRUE
```

```
# it is TRUE so it is installed
```

How to create a matrix

```
m <- matrix(1:9, ncol=3, nrow=3)
```

Question: construct a matrix of dimension 1000 by 3 of normally distributed data (mean 0, sd 1)

```
set.seed(123)
m <- matrix(rnorm(3000), ncol = 3)
dim(m)
```

```
## [1] 1000    3
```

```
head(m)
```

```
##              [,1]        [,2]       [,3]
## [1,] -0.56047565 -0.99579872 -0.5116037
## [2,] -0.23017749 -1.03995504  0.2369379
## [3,]  1.55870831 -0.01798024 -0.5415892
## [4,]  0.07050839 -0.13217513  1.2192276
## [5,]  0.12928774 -2.54934277  0.1741359
## [6,]  1.71506499  1.04057346 -0.6152683
```

7: Formatting dates

```
library("lubridate")
```

ymd() takes a vector representing year,month and day and converts it to a Date vector.

```
my_date <- ymd("2015-01-01")
str(my_date)
```

```
##  Date[1:1], format: "2015-01-01"
```

```
# sep indicates the character to use to separate each component
my_date <- ymd(paste("2015", "1", "1", sep = "-"))
str(my_date)
```

```
##  Date[1:1], format: "2015-01-01"
```

Create a character vector from the year, month and day columns of surveys using paste()

```
head(paste(surveys$year, surveys$month, surveys$day, sep = "-"))
```

```
## [1] "1977-7-16"  "1977-8-19"  "1977-9-13"  "1977-10-16" "1977-11-12"
## [6] "1977-11-12"
```

This character vecor can be used as the argument for ymd():

```
head(ymd(paste(surveys$year, surveys$month, surveys$day, sep = "-")))
```

```
## Warning: 129 failed to parse.
```

```
## [1] "1977-07-16" "1977-08-19" "1977-09-13" "1977-10-16" "1977-11-12"
## [6] "1977-11-12"
```

the resulting date vatcor can be added to surveys as a new col called date:

```
surveys$date <- ymd(paste(surveys$year, surveys$month, surveys$day, sep = "-"))
```

```
## Warning: 129 failed to parse.
```

```
str(surveys) # we can see there is indeed a new col
```

```
## 'data.frame':    34786 obs. of  14 variables:
##  $ record_id      : int  1 72 224 266 349 363 435 506 588 661 ...
##  $ month          : int  7 8 9 10 11 11 12 1 2 3 ...
##  $ day            : int  16 19 13 16 12 12 10 8 18 11 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1978 1978 1978 ...
##  $ plot_id        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ species_id     : chr  "NL" "NL" "NL" "NL" ...
##  $ sex            : chr  "M" "M" "" "" ...
##  $ hindfoot_length: int  32 31 NA NA NA NA NA NA NA NA ...
##  $ weight         : int  NA NA NA NA NA NA NA NA 218 NA ...
##  $ genus          : chr  "Neotoma" "Neotoma" "Neotoma" "Neotoma" ...
##  $ species        : chr  "albigula" "albigula" "albigula" "albigula" ...
##  $ taxa           : chr  "Rodent" "Rodent" "Rodent" "Rodent" ...
##  $ plot_type      : chr  "Control" "Control" "Control" "Control" ...
##  $ date           : Date, format: "1977-07-16" "1977-08-19" ...
```

```
# make sure everything worked correctly
summary(surveys$date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "1977-07-16" "1984-03-12" "1990-07-22" "1990-12-15" "1997-07-29" "2002-12-31"
##         NA's
##        "129"
```

```
# some dates have missing values, we need to look where they come from
is_missing_date <- is.na(surveys$date)
date_columns <- c("year", "month", "day")
missing_dates <- surveys[is_missing_date,  date_columns]

head(missing_dates)
```

```
##      year month day
## 3144 2000     9  31
## 3817 2000     4  31
## 3818 2000     4  31
## 3819 2000     4  31
## 3820 2000     4  31
## 3856 2000     9  31
```

8: Summary of R objects:

vector: one dimension (they have a length), single type of data.

###matrix: two dimensions, single type of data. ### data.frame: two dimensions, one type per column.

9: Lists

List is 1D, every item can be different data type.

For example, we can create a list with numbers, character, dataframe,...

```
l <- list(1:10, ## numeric
          letters, ## character
          installed.packages(), ## a matrix
          cars, ## a data.frame
          list(1, 2, 3)) ## a list
length(l)
```

```
## [1] 5
```

```
str(l)
```

```
## List of 5
##  $ : int [1:10] 1 2 3 4 5 6 7 8 9 10
##  $ : chr [1:26] "a" "b" "c" "d" ...
##  $ : chr [1:358, 1:16] "abind" "affy" "affyio" "airway" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:358] "abind" "affy" "affyio" "airway" ...
##   .. ..$ : chr [1:16] "Package" "LibPath" "Version" "Priority" ...
##  $ :'data.frame':    50 obs. of  2 variables:
##   ..$ speed: num [1:50] 4 4 7 7 8 9 10 10 10 11 ...
##   ..$ dist : num [1:50] 2 10 4 22 16 10 18 26 34 17 ...
##  $ :List of 3
##   ..$ : num 1
##   ..$ : num 2
##   ..$ : num 3
```

List subsetting by using [] for a new sub-list, or [[]] to extract a single element of that list.

```
l[[1]] ## first element
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
l[1:2] ## list of length 2
```

```
## [[1]]
##  [1]  1  2  3  4  5  6  7  8  9 10
##
## [[2]]
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
l[1] ## list of length 1
```

```
## [[1]]
##  [1]  1  2  3  4  5  6  7  8  9 10
```

10: exporting and saving data

```
# to export the surveys data to the my_surveys.csv file in the data_output directory
write.csv(surveys, file = "data_output/my_surveys.csv")

#to save data as rda
save(surveys, file = "data_output/surveys.rda")
rm(surveys)
load("data_output/surveys.rda")
head(surveys)
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2        72     8  19 1977       2         NL   M              31     NA
## 3       224     9  13 1977       2         NL                  NA     NA
## 4       266    10  16 1977       2         NL                  NA     NA
## 5       349    11  12 1977       2         NL                  NA     NA
## 6       363    11  12 1977       2         NL                  NA     NA
##     genus  species   taxa plot_type       date
## 1 Neotoma albigula Rodent   Control 1977-07-16
## 2 Neotoma albigula Rodent   Control 1977-08-19
## 3 Neotoma albigula Rodent   Control 1977-09-13
## 4 Neotoma albigula Rodent   Control 1977-10-16
## 5 Neotoma albigula Rodent   Control 1977-11-12
## 6 Neotoma albigula Rodent   Control 1977-11-12
```

```
#to save as RDS
saveRDS(surveys, file = "data_output/surveys.rds")
rm(surveys)
surveys <- readRDS("data_output/surveys.rds")
head(surveys)
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2        72     8  19 1977       2         NL   M              31     NA
## 3       224     9  13 1977       2         NL                  NA     NA
## 4       266    10  16 1977       2         NL                  NA     NA
## 5       349    11  12 1977       2         NL                  NA     NA
## 6       363    11  12 1977       2         NL                  NA     NA
##     genus  species   taxa plot_type       date
## 1 Neotoma albigula Rodent   Control 1977-07-16
## 2 Neotoma albigula Rodent   Control 1977-08-19
## 3 Neotoma albigula Rodent   Control 1977-09-13
## 4 Neotoma albigula Rodent   Control 1977-10-16
## 5 Neotoma albigula Rodent   Control 1977-11-12
## 6 Neotoma albigula Rodent   Control 1977-11-12
```

11: Additional exercises

Question 1: You're doing an colony counting experiment, counting every day how many molds you see in your cell cultures.

Create a vector named molds containing the results of your counts: 1, 2, 5, 8 and 10. Create a vector days containing the week day, from Monday to Friday. Use these two vector to create a data.frame named molds_study containing two variables, Day and Molds_count.

```
molds <- c(1,2,5,8,10)
days <- c("Monday","Tuesday","Wednesday","Thursday","Friday")
molds_study <- data.frame("Day"= days,"Molds_count"= molds)
molds_study
```

```
##         Day Molds_count
## 1    Monday           1
## 2   Tuesday           2
## 3 Wednesday           5
## 4  Thursday           8
## 5    Friday          10
```

```
class(molds_study)
```

```
## [1] "data.frame"
```

Create a new data.frame that contains the observations where more than 2 colonies were counted. How many observations are there? How many counts are there in total for these observations.

```
m2 <- molds_study$Molds_count > 2 #trouver obs avec >2 colonies
molds2 <- molds_study[m2,] # créer le nouveau data.frame
molds2
```

```
##         Day Molds_count
## 3 Wednesday           5
## 4  Thursday           8
## 5    Friday          10
```

```
nrow(molds2) # compter le nombre d'obs
```

```
## [1] 3
```

```
sum(molds2$Molds_count) # combien de colonies y comptons nous?
```

```
## [1] 23
```

You repeat the molds study experiment the following week and count the following numbers of molds: 1, 6, 6, 5 and 4.

Add these data as a third column to the molds_study data.frame and rename the variables as Day, Molds_1 and Molds_2.

```
molds_2 <- c(1,6,6,5,4)
molds_study2 <- data.frame("Day"= days,"Molds_1"= molds, "Molds_2"=molds_2)
molds_study2
```

```
##         Day Molds_1 Molds_2
## 1    Monday       1       1
## 2   Tuesday       2       6
## 3 Wednesday       5       6
## 4  Thursday       8       5
## 5    Friday      10       4
```

Calculate for each experiment the total number of molds counted. Check if the first experiment counted more molds than the second one.

```
sum1 <- sum(molds_study2$Molds_1)
sum2 <- sum(molds_study2$Molds_2)
sum1>sum2 #it returns TRUE so 1st experiment counted more molds
```

```
## [1] TRUE
```

Save the molds_study variable in a file named molds_study.rda.

```
ave(molds_study, file = "data_output/molds_study.rda")
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
##    Day Molds_count
## 1 <NA>          NA
## 2 <NA>          NA
## 3 <NA>          NA
## 4 <NA>          NA
## 5 <NA>          NA
```

Question 2: We are going to analyse beer consumption in 48 individuals. The data are available in the rWSBIM1207 package. The data illustrated the fictive beer consumption in liters per year at different age according to gender and employment.

Using the beers.csv() function from rWSBIM1207, find the path the beers.csv file and read it to produce a data.frame named beers. The spreadsheet uses semi-colons ; to separate cells. Use read.csv2() and read.delim() and set the separator appropriately, and verify that the two variables are identical.

```
library(rWSBIM1207)
beers <- read.csv(beers.csv())
beers2 <- read.csv2(beers.csv())
beers2
```

Check the number of observations and identify the variables that are available. Calculate a summary of each variable using the summary function directly on the data.frame.

```
sum(beers2$Record_ID)
```

```
## [1] 1176
```

```
summary(beers2)
```

```
##    Record_ID         Work            Consumption       Gender
##  Min.   : 1.00   Length:48          Min.   :100.0   Length:48
##  1st Qu.:12.75   Class :character   1st Qu.:142.0   Class :character
##  Median :24.50   Mode  :character   Median :180.0   Mode  :character
##  Mean   :24.50                      Mean   :180.2
##  3rd Qu.:36.25                      3rd Qu.:200.0
##  Max.   :48.00                      Max.   :350.0
##                                     NA's   :1
##       Age             Day             Month             Year
##  Min.   :25.0   Min.   : 1.00   Min.   : 1.000   Min.   :2017
##  1st Qu.:32.5   1st Qu.: 8.75   1st Qu.: 6.000   1st Qu.:2017
##  Median :40.0   Median :16.50   Median : 8.000   Median :2017
##  Mean   :40.0   Mean   :15.85   Mean   : 7.417   Mean   :2017
##  3rd Qu.:47.5   3rd Qu.:22.50   3rd Qu.: 9.250   3rd Qu.:2017
##  Max.   :55.0   Max.   :30.00   Max.   :12.000   Max.   :2017
##
```

Calculate the mean and the median age and consumption.

```
mean(beers2$Age)
```

```
## [1] 40
```

```
median(beers2$Age)
```

```
## [1] 40
```

```
#OU
mean(beers2[,5])
```

```
## [1] 40
```

```
median(beers2[,5])
```

```
## [1] 40
```

Do men consume more beer than women on average? To answer this question, calculate the mean consumption for men only, selecting the observations with Gender equal to Male. Then do the same for observations with Gender equal to Female.

```
m <- beers2$Gender == "Male" #extraire lignes correspondant aux hommes
beers2[m, "Consumption"] # extraire consommation des hommes
```

```
##  [1] 100 110 120 150 155 153 175 190 200 180 200 250 150 165 170 190 180 195 223
## [20] 225 250 290 300 350
```

```
mean(beers2[m, "Consumption"]) # moyenne
```

```
## [1] 194.625
```

```
f <- beers2$Gender == "Female"                  #extraire lignes correspondant aux femmes
beers2[f, "Consumption"]                        # extraire consommation des femmes
```

```
##  [1] 210 220 230 200 190 180 170 175 180 120 125 140 200 190 180 150 135 144 125
## [20] 130 140  NA 125 140
```

```
mean(beers2[f, "Consumption"], na.rm=TRUE)      # moyenne
```

```
## [1] 165.1739
```

Calculate a two-way table of gender and employment status.

```
genderXemployment <- table(beers2$Gender,beers2$Work)
genderXemployment
```

```
##
##          Employed Unemployed
##   Female       12         12
##   Male         12         12
```

Remove observations with missing values and export the data into a new csv file called beers__no__na.csv.

```
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
```

```
beers_no_na <- beers2 %>%
  na.omit

save(beers_no_na, file = "data_output/beers_no_na.csv")
```

Question 3: We are going to analyse clinical data from The Cancer Genome Atlas (TCGA). The data are available in the rWSBIM1207 package.

Obtain the path to th csv file containing the clinical data need for this exercise using the clinical1.csv function and read it into R as a data.frame called clinical.

```
clinical <- read.csv(clinical1.csv())
class(clinical)
```

```
## [1] "data.frame"
```

Inspect the data using str and View. How many patients are recorded in the table?

```
# str(clinical)
# View(clinical)
nrow(clinical)
```

Print the column names using two different functions.

```
#?
```

Create a smaller data frame called clinical_mini containing only the columns corresponding to the patientID, gender, age_at_diagnosis and smoking_history. Try to do this using column indices and column names.

```
clinical_mini <- data_frame(clinical$patientID,clinical$gender,clinical$age_at_diagnosis,clinical$smoki
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   1.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
names(clinical_mini)[1] <- "patientID"
names(clinical_mini)[2] <- "gender"
names(clinical_mini)[3] <- "age_at_diagnosis"
names(clinical_mini)[4] <- "smoking_history"
```

Inspect the smoking_history column. How many categories are recorded? How many observations are there for each category?

```
tablesm<- table(clinical_mini$smoking_history)
length(unique(tablesm))
```

```
## [1] 5
```

The column age at diagnosis is recorded in days. Create a new column years_at_diagnosis corresponding to the age at diagnosis converted in years.

```
new_clinical_mini <- clinical_mini %>%
  mutate(years_at_diagnosis = age_at_diagnosis/365)
```

Calculate the mean and median age at diagnosis. Hint: pay attention to missing values!

```
mean(new_clinical_mini$years_at_diagnosis, na.rm=TRUE)
```

```
## [1] 65.81568
```

```
median(new_clinical_mini$years_at_diagnosis, na.rm=TRUE)
```

```
## [1] 66.85479
```

Is there a difference between the years_at_diagnosis for male and female patients?

```
library(tidyverse)
```

```
male<- new_clinical_mini %>%
  group_by(gender="male", na.rm=TRUE)
mean(male$years_at_diagnosis, na.rm=TRUE)
```

```
## [1] 65.81568
```

```
female<- new_clinical_mini %>%
  group_by(gender="female", na.rm=TRUE)
mean(female$years_at_diagnosis, na.rm=TRUE)
```

```
## [1] 65.81568
```

Use the quantile function to calculate the first and last quartile of age at diagnosis. Use the help function (?quantile) to see how to use the quantile function.

```
?quantile
quantile(new_clinical_mini$years_at_diagnosis, na.rm=TRUE)
```

```
##       0%      25%      50%      75%     100%
## 38.53151 59.54521 66.85479 72.94521 88.85479
```

Use the summary function to confirm your previous results.

```
summary(new_clinical_mini$years_at_diagnosis)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   38.53   59.55   66.85   65.82   72.95   88.85      31
```