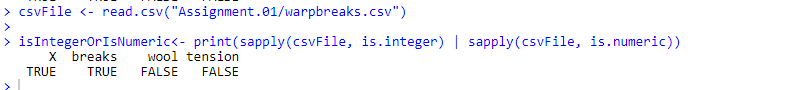
**Exercise 2.1.** Type conversions.

1. Load the built in warpbreaks data set. Find out, in a single command, which columns of warpbreaks are either numeric or integer.

Ans) For the warpbreaks data set I used read.csv as it is a csv file to read the data set. Then I used Or in R to write both codes in one line to give the result. The code snippet is shown below:



The result from console is shown below:



1. Is numeric a natural data type for the columns which are stored as such? Convert to integer

when necessary. (See also ? warpbreaks for an explanation of the data).

Ans) It is a natural data type for columns “X” ad “breaks” and it is not for columns “wool” and “tension”.

1. Error messages in R sometimes report the underlying type of an object rather than the

user-level class. Derive from the following code and error message what the underlying type

of an R function is.

mean[1]

## Error: object of type 'closure' is not subsettable

Confirm your answer using typeof.

Ans) “typeof” determines the (R internal) type or storage mode of any object. Mean is a closure type as shown from this code execution:



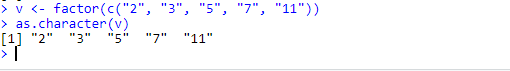
In R The square brackets “[]” are used for indexing into a vector, matrix, array, list or dataframe. That is why mean[1] gives the following error as you’re mistakenly using square brackets to try and subset it, thinking that it represents a data.frame or vector or something.

**Exercise 2.2. Type the following code in your R terminal.**

v <- factor(c("2", "3", "5", "7", "11"))

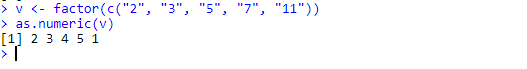
1. Convert v to character with as.character. Explain what just happened.

Ans) The result is showing the numbers as characters as it was initially saved as string data type:



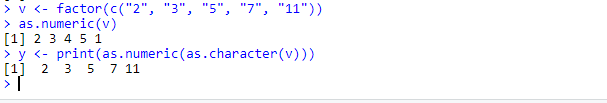
1. Convert v to numeric with as.numeric. Explain what just happened.

Ans) The reason this happens is that factors are actually stored internally as integers, and the labels are what is actually displayed when you print them out (things like "11" in this case). The function as.numeric thinks that what you want is the underlying integer representation.



1. How would you convert the values of v to integers?

Ans) This is one approach of extracting the values as integers in this vector. Another approach is setting stringsAsFactors = FALSE



**Exercise 2.3. In this exercise we'll use readLines to read in an irregular textfile. The file looks like**

**this (without numbering).**

1 // Survey data. Created : 21 May 2013

2 // Field 1: Gender

3 // Field 2: Age (in years)

4 // Field 3: Weight (in kg)

5 M;28;81.3

6 male;45;

7 Female;17;57,2

8 fem.;64;62.8

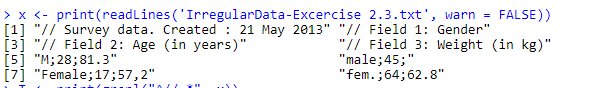
You may copy the text from this pdf file in a textfile called example.txt or download the file from

our Github page.

1. Read the complete file using readLines.

Ans) The code and result in console for this is:

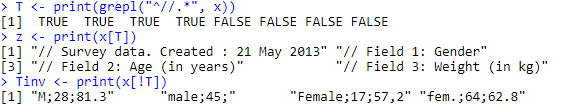
We used warn= FALSE to disable the warnings.



1. Separate the vector of lines into a vector containing comments and a vector containing

the data. Hint: use grepl.

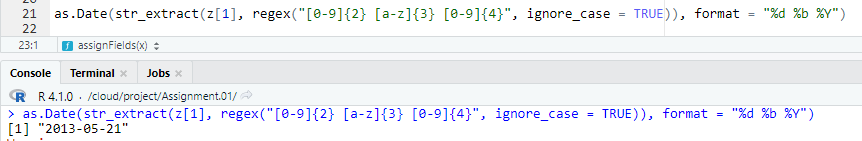
Ans) As hinted, we used grepl to separate the lines. The code for this is:



Z is the vector that contains the comments, Tinv is the vector that contains the data.

1. Extract the date from the first comment line.

Ans) The date was extracted in 2 steps, as shown in the code:

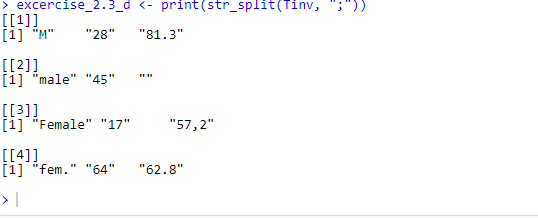


We used Regex to get the date from string and used as.Date() to convert it into the right format.

1. Read the data into a matrix as follows.
2. Split the character vectors in the vector containing data lines by semicolon (;) using

strsplit.

Ans) Here as instructed strsplit was used to split the vector:

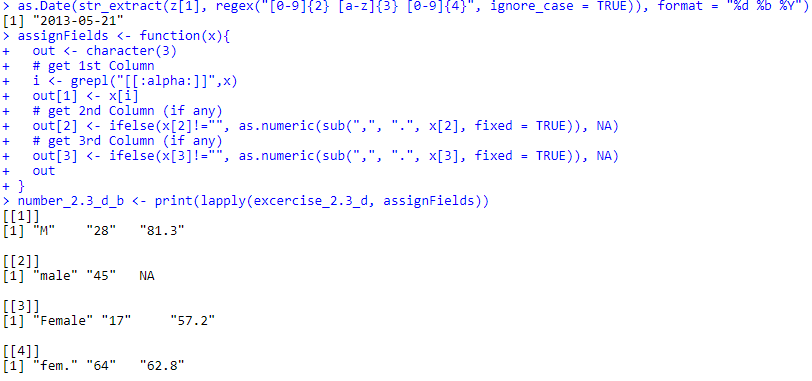


I used the print() function so that it displays in the console in a single line of code. As you have noticed I did the same with previous examples. This is to minimize the use of extra lines of code.

1. Find the maximum number of fields retrieved by split. Append rows that are shorter

with NA's.

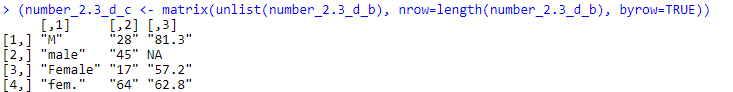
Ans) Here I used the function (assignFields) used from this book to create the maximum number of fields retrieved by split and replaced the empty strings with NAs. Then used lapply to combine the function with my dataset:



I’m only giving the screenshot of the console because it also shows the code and what was written.

1. Use unlist and matrix to transform the data to row-column format.

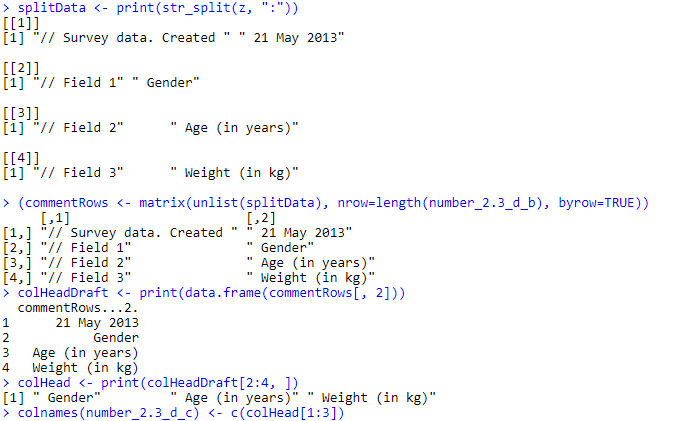
Ans) The console code and result are shown below:



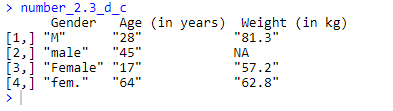
1. From comment lines 2-4, extract the names of the fields. Set these as colnames for the

matrix you just created.

Ans) I completed this using multiple steps, as shown below, I first split the variable that contained only the comments, then I used matrix and unlist to organize the data in columns and rows, then I took the second column and took the last 3 rows of the second column and placed it as column header for our prepared dataset:



Finally, that gives us:

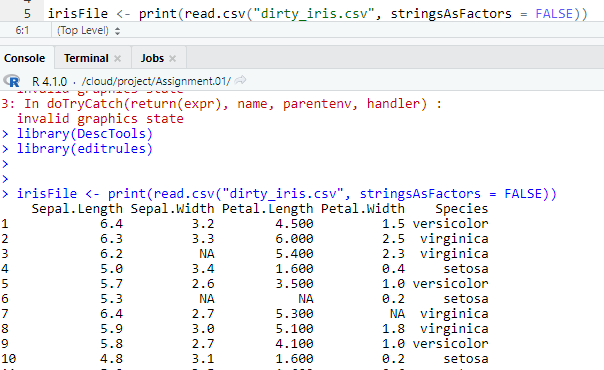


**Exercise 3.1. Reading and manually checking.**

1. View the file in a text-editor to determine its format and read the file into R. Make sure that

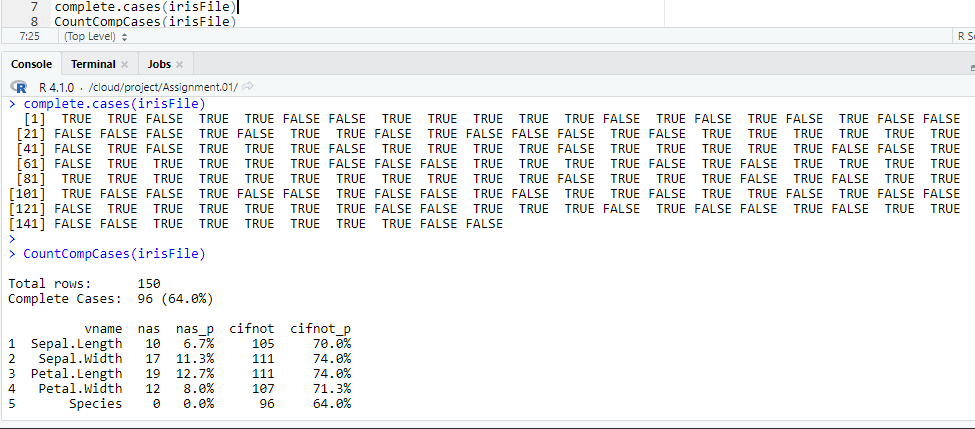
strings are not converted to factor.

Ans) Since this is a csv file, we used read.csv to obtain the values of the file and stringAsFactors is set to FALSE to avoid strings being converted to factors:



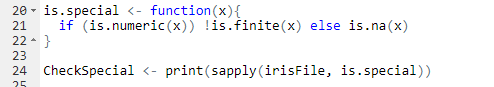
1. Calculate the number and percentage of observations that are complete.

Ans) The total number of complete cases is 96 and the percentage is 64%.

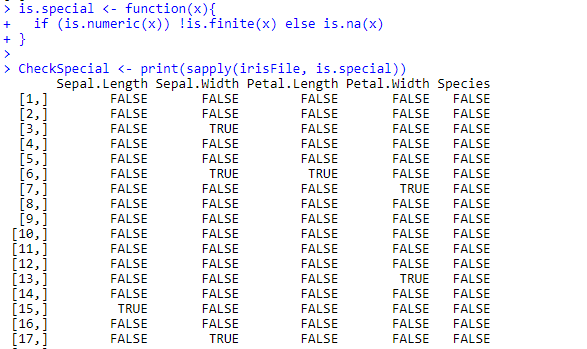


1. Does the data contain other special values? If it does, replace them with NA

Ans) We used a function to define if it is special and used it to calculate the values:



This is the result from console:



The dataset does not contain any other special values.

**Exercise 3.2. Checking with rules**

1. Besides missing values, the data set contains errors. We have the following background

knowledge:

– Species should be one of the following values: setosa, versicolor or virginica.

– All measured numerical properties of an iris should be positive.

– The petal length of an iris is at least 2 times its petal width.

– The sepal length of an iris cannot exceed 30 cm.

– The sepals of an iris are longer than its petals.

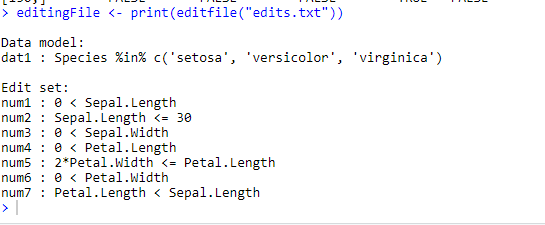
Define these rules in a separate text file and read them into R using editfile (package

editrules). Print the resulting constraint object.

Ans) We stored the rules in the file “edits.txt” as shown below:



This is the result from the console in rstudio cloud:

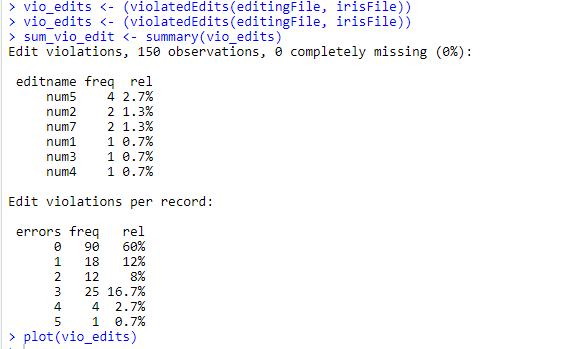


1. Determine how often each rule is broken (violatedEdits). Also summarize and plot the

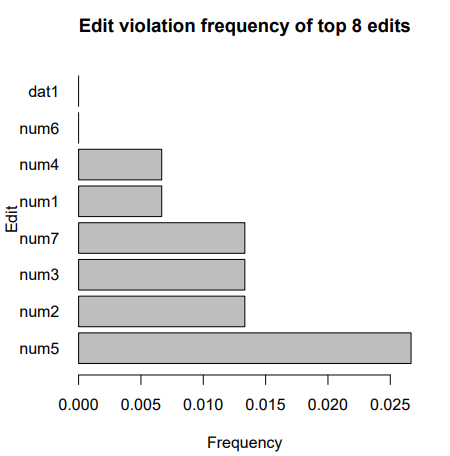
result.

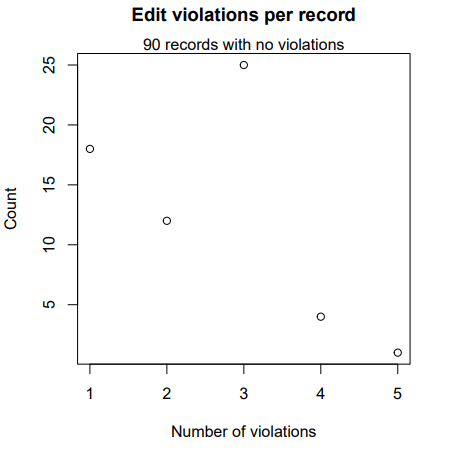
Ans) The first image shows the code written at script and then we have the code run in console.





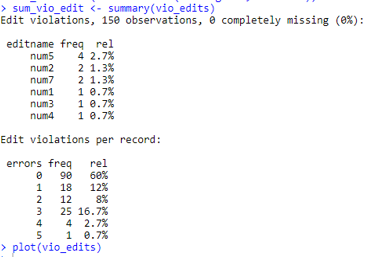
Below is an image of the plot-vio\_edits:





1. What percentage of the data has no errors?

Ans) As shown in the “Edit Violations per record” in the previous screenshot, 90 observations has no errors which is 60% of the data.



1. Find out which observations have too long petals using the result of violatedEdits.

Ans) Here I preferred using which() as I could not find a way to use “violatedEdits” to find the error.

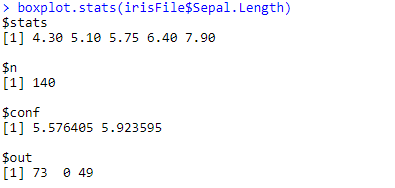


1. Find outliers in sepal length using boxplot and boxplot.stats. Retrieve the

corresponding observations and look at the other values. Any ideas what might have

happened? Set the outliers to NA (or a value that you find more appropriate)

Ans) Here I used $ to find the column “Sepal.Length” and run boxplot.stats() on that specific column as that was the only column that was required to show it.

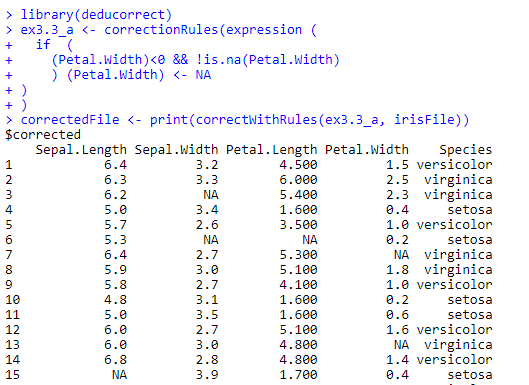


**Exercise 3.3. Correcting**

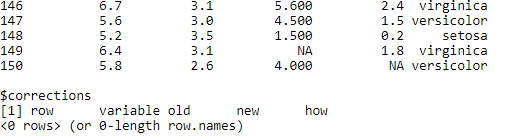
1. Replace non positive values from Petal.Width with NA using correctWithRules from

the library deducorrect.

Ans) The console snippet below shows the code:

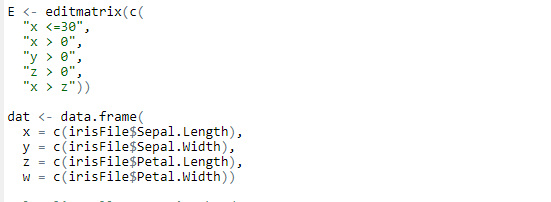


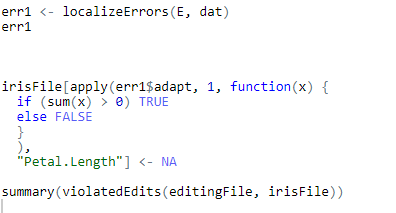
This is the snippet for the last lines:



1. Replace all erronous values with NA using (the result of) localizeErrors.

Ans) The code snippet is shown below for the problem:





The console output is shown below:

