CS 102 Practical#1

Anjelo Marco Cabaña

2024-03-07

```
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
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.4.4     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
#A)
#1
data(warpbreaks)
str(warpbreaks)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ breaks : num  26 30 54 25 70 52 51 26 67 18 ...
##  $ wool   : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1 1 2 ...
```

```
sapply(warpbreaks, class)
```

```
##    breaks      wool   tension
## "numeric"  "factor"  "factor"
```

```
# The data types of each columns is breaks is numeric, while both wool and tension are factor.
```

```
#2 We can see that the breaks column is numeric while the wool and tension are factors. The number of o

data(warpbreaks)
nrow(warpbreaks)
```

```
## [1] 54
```

```r
#3 Yes, numeric is a natural data type for the columns that are stored. However if necessary, you can c

str(warpbreaks)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ breaks : num  26 30 54 25 70 52 51 26 67 18 ...
##  $ wool   : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1 1 2 ...
```

```r
sapply(warpbreaks, class)
```

```
##    breaks      wool   tension
## "numeric"  "factor"  "factor"
```

```r
data(warpbreaks)
warpbreaks$breaks <- as.integer(warpbreaks$breaks)
class(warpbreaks$breaks)
```

```
## [1] "integer"
```

```r
#4 To avoid errors it is important to ensure that the variables you are working with has the correct da
```

```r
#B)
#1
data <- readLines("exampleFile.txt")
```

```
## Warning in readLines("exampleFile.txt"): incomplete final line found on
## 'exampleFile.txt'
```

```r
print(data)
```

```
## [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"
```

```r
#2
lines <- readLines("exampleFile.txt")
```

```
## Warning in readLines("exampleFile.txt"): incomplete final line found on
## 'exampleFile.txt'
```

```r
comment_lines <- grepl("^\\/{3}", lines)
comments <- lines[comment_lines]
data <- lines[!comment_lines]
print(comments)
```

```
## character(0)
```

```r
print(data)
```

```
## [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"
```

```
#3
lines <- readLines("exampleFile.txt")
```

```
## Warning in readLines("exampleFile.txt"): incomplete final line found on
## 'exampleFile.txt'
```

```
comment_lines <- grepl("^\\/{3}", lines)
comments <- lines[comment_lines]
data <- lines[!comment_lines]
date_pattern <- "Created: ([0-9]+ [A-Za-z]+ [0-9]+)"
date_match <- regexpr(date_pattern, comments[1])
date_text <- regmatches(comments[1], date_match)
date_text <- gsub("Created: ", "", date_text)
cat("It was created on", date_text, "\n")
```

```
## It was created on
```