Markdown_Week11_Cleaning_Tidying_data

Aymeric Collart

1. Prepare the environment

1.1 Load the libraries

```r
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
library(tidyr)
library(openxlsx)
```

1.2 Load the originally scraped data

```r
load(file = "ArticleETToday_CorpusCourse.Rdata")
```

2. Examples of cleaning/preparing processes

2.1 Further annotations

```r
## Example: Adding columns for year, month and day

Article_total$year <- substr(Article_total$time, start = 1, stop = 4)
Article_total$month <- substr(Article_total$time, start = 6, stop = 7)
Article_total$day <- substr(Article_total$time, start = 9, stop = 10)
```

2.2 Data transformation

2.2.1 Split the articles into paragraphs

```
Article_total$original_article <- Article_total$body

Article_total2 <- Article_total %>%
  mutate(body = strsplit(as.character(body), "\r\n")) %>%
  unnest(body)
```

2.2.2 Remove unwanted paragraphs

```


Article_total2 <- Article_total2[-grep(" ", Article_total2$body),]
Article_total2 <- Article_total2[-grep(" ", Article_total2$body),]
```

```


### Most of them start with the " 圖／" (image) and " 文／" (text) source: Check first if
 ↪  that's the case
test <- Article_total2[+grep(" 圖／", Article_total2$body),]
test <- Article_total2[+grep(" 文／", Article_total2$body),]

### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep(" 圖／", Article_total2$body),]
Article_total2 <- Article_total2[-grep(" 文／", Article_total2$body),]
```

Example 1: Corresponds to image legends

```


### Most of them start with the " " symbol: Check first if that's the case
test <- Article_total2[+grep(" ", Article_total2$body),]

### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep(" ", Article_total2$body),]

### Some start with the " " and " " symbol (link to other articles): Check first if that's
 ↪  the case
test <- Article_total2[+grep(" ", Article_total2$body),]

test <- Article_total2[+grep(" ", Article_total2$body),]

### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep(" ", Article_total2$body),]
Article_total2 <- Article_total2[-grep(" ", Article_total2$body),]

### Most of them start with the " " symbol: Check first if that's the case
test <- Article_total2[+grep(" ", Article_total2$body),]

```

```r
### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep(" ", Article_total2$body),]


### Some rows are just made of the message "【其他新聞】" (other news): Check first if that's the case
test <- Article_total2[+grep("【其他新聞】", Article_total2$body),]

### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep("【其他新聞】", Article_total2$body),]

### Some rows are just made of the message "更多新聞" (more news): Check first if that's the case
test <- Article_total2[+grep("更多新聞", Article_total2$body),]

### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep("更多新聞", Article_total2$body),]

### Some rows are just made of the message "延伸閱讀" (read more): Check first if that's the case
test <- Article_total2[+grep("延伸閱讀", Article_total2$body),]

### Indeed the case: Can be removed
Article_total2 <- Article_total2[-grep("延伸閱讀", Article_total2$body),]
```

Example 2: Corresponds to messages from ETToday

```r

### Most of them start with the two characters "記者" (journalist)
test <- Article_total2
test$FirstTwoCharacters <- substr(Article_total2$body, start = 1, stop = 2)
## Only 178 sentences out of 361154 will be wrongly removed, quite acceptable

Article_total2$FirstTwoCharacters <- substr(Article_total2$body, start = 1, stop = 2)
Article_total2 <- Article_total2[-grep("記者", Article_total2$FirstTwoCharacters),]

Article_total2$FirstTwoCharacters <- NULL


### Some start with the four characters "實習記者" (journalist-internship)
test <- Article_total2
test$FirstFourCharacters <- substr(Article_total2$body, start = 1, stop = 4)
test <- test[+grep("實習記者", test$FirstFourCharacters),]

## Indeed all rows need to be removed
Article_total2$FirstFourCharacters <- substr(Article_total2$body, start = 1, stop = 4)
Article_total2 <- Article_total2[-grep("實習記者", Article_total2$FirstFourCharacters),]

Article_total2$FirstFourCharacters <- NULL
```

Example 3: Corresponds to the identity of the journalist

```r
Article_total2 <- Article_total2[!(Article_total2$year=="2023"), ]
```

Example 4: Remove data from 2023 (controversial cleaning part)

2.2.3 Remove empty rows

```r
## Test the code to make sure we are not removing too much
test <- Article_total2[(Article_total2$body==""), ]

## Indeed all the rows correspond to empty paragraphs

Article_total2 <- Article_total2[!(Article_total2$body==""), ]

## Visual inspection: Still empty rows, such as line number 29
Article_total2$body[29] ## corresponds to a space
```

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

```r
## Test the code to make sure we are not removing too much
test <- Article_total2[(Article_total2$body==" "), ]

## Indeed all the rows correspond to paragraphs with a space
Article_total2 <- Article_total2[!(Article_total2$body==" "), ]

## Potentially more than one space: Test the code to make sure we are not removing too
↪  much
test <- Article_total2[(Article_total2$body=="  "), ] #test two spaces
test <- Article_total2[(Article_total2$body=="   "), ] #test three spaces
test <- Article_total2[(Article_total2$body=="    "), ] #test four spaces
test <- Article_total2[(Article_total2$body=="     "), ] #test five spaces
test <- Article_total2[(Article_total2$body=="      "), ] #test six spaces
test <- Article_total2[(Article_total2$body=="       "), ] #test seven spaces

## Indeed all the rows correspond to paragraphs with 2 to 5 spaces
Article_total2 <- Article_total2[!(Article_total2$body=="  "), ]
Article_total2 <- Article_total2[!(Article_total2$body=="   "), ]
Article_total2 <- Article_total2[!(Article_total2$body=="    "), ]
Article_total2 <- Article_total2[!(Article_total2$body=="     "), ]

## Visual inspection: Still empty rows, such as line number 29
Article_total2$body[29]
```

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

```r
## I don't really know what this is, check using the row number instead of the symbol
↪  itself
test <- Article_total2[(Article_total2$body==Article_total2$body[29]), ]
```

```
## Extract the symbol
ToRemove <- Article_total2$body[29]

## Indeed all the rows correspond to paragraphs with a space --> Removing using the
↪  symbol itself
Article_total2 <- Article_total2[!(Article_total2$body==ToRemove), ]

## Visual inspection: Still empty rows, such as line number 13974
Article_total2$body[13974] ## corresponds to nothing, but wasn't caught earlier
```

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

```
## Extract the symbol, test and remove
ToRemove <- Article_total2$body[13974]
test <- Article_total2[(Article_total2$body==ToRemove), ]
Article_total2 <- Article_total2[!(Article_total2$body==ToRemove), ]

## Visual inspection: Still empty rows, such as line number 8368
Article_total2$body[8368] ## I don't know what kind of space it is
```

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

```
## Extract the symbol, test and remove
ToRemove <- Article_total2$body[8368]
test <- Article_total2[(Article_total2$body==ToRemove), ]
Article_total2 <- Article_total2[!(Article_total2$body==ToRemove), ]

## Visual inspection: Still empty rows, such as line number 67
Article_total2$body[67] ## I don't know what kind of space it is
```

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

```
## Extract the symbol, test and remove
ToRemove <- Article_total2$body[67]
test <- Article_total2[(Article_total2$body==ToRemove), ]
Article_total2 <- Article_total2[!(Article_total2$body==ToRemove), ]

## Visual inspection: Still empty rows, such as line number 15279
Article_total2$body[15279] ## I don't know what kind of space it is
```

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

```
## Extract the symbol, test and remove
ToRemove <- Article_total2$body[15279]
test <- Article_total2[(Article_total2$body==ToRemove), ]
Article_total2 <- Article_total2[!(Article_total2$body==ToRemove), ]
```

This could go on forever, but blank spaces will not affect the analysis.

2.2.4 Remove weird scraping instances

```
## This correspond to cases where the signs "<" + other signs appear, these are rows with
↪  HTML language
test <- Article_total2
test_HTML <- grepl("<[^>]+>", Article_total2$body)
test$HTML <- test_HTML
test <- test[(test$HTML=="TRUE"), ]

## Indeed corresponds to rows to remove
Article_total2$HTML <- test_HTML
Article_total2 <- Article_total2[!(Article_total2$HTML=="TRUE"), ]

Article_total2$HTML <- NULL
```

There still remains “unclean” rows (around 140), but we did the best we can so far. Because such rows are quite rare, this will not affect further analyses

3. Save the data

3.1 Save as an Excel file

```
write.xlsx(Article_total2, "ArticleETToday_CorpusCourse_CLEAN.xlsx")
```

```
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
```

```
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
## Warning in wb$writeData(df = x, colNames = colNames, sheet = sheet, startCol = startCol, : icy&quot;
```

```
## &lt;/html&gt; is truncated.
## Number of characters exeed the limit of 32767.
```

3.2 Save as an RData file

```r
save(Article_total2, file = "ArticleETToday_CorpusCourse_CLEAN.Rdata")
```