Lab Exercise 5

Camille Jean Ahumada

2024-04-01

```
library(readr)
library(stringr)
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
```

```
# Load Arxiv Scraped Dataset
arxiv <- read_csv("/cloud/project/Lab Exercise 5/Arxiv papers on Animation.csv")
```

```
## New names:
## * `` -> `...1`

## Rows: 150 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): title, author, subject, abstract
## dbl  (1): ...1
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Extracting the date from the meta column
arxiv_date_only <- str_extract(arxiv$meta, "\\d+\\s[A-Za-z]+\\s\\d+")

# Changing to date type
arxiv_date_type <- as.Date(arxiv_date_only, format = "%d %b %Y")
head(arxiv_date_type)
```

```
## Date of length 0
```

```
# Removing meta and number column and appending the new date column
# Mutating all while converting other columns to lowercase, removing parenthesis text in the subject co
cleaned_arxiv <- arxiv %>%
  mutate(date = arxiv_date_type,
         subject = gsub("\\s\\(.*\\)", "", subject),
         across(where(is.character), tolower)) %>%
  select(-meta, -...1)
```

```
## Error in `mutate()`:
## i In argument: `date = arxiv_date_type`.
## Caused by error:
## ! `date` must be size 150 or 1, not 0.
```

```
# Writing to CSV
write.csv(cleanedArxiv, "/cloud/project/Lab Exercise 5/cleanedArxiv.csv")
```

```
## Error in eval(expr, p): object 'cleanedArxiv' not found
```

```
library(readr)
library(stringr)
library(dplyr)

# Load Arxiv Scraped Dataset
productsReviews <- read_csv("/cloud/project/Lab Exercise 5/50ProductReviews.csv")
```

```
## New names:
## Rows: 2500 Columns: 8
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (7): prod_name, title, reviewer, review, date, ratings, type_of_purchase dbl
## (1): ...1
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
# Extract the date information from the meta column and convert it to a date type.
reviewsDataType <- as.Date(str_extract(productsReviews$date, "\\d+\\s[A-Za-z]+\\s\\d+"), format = "%d %

# Retrieve the rating from the rating column and convert it to an integer.
reviewsRatingsInteger <- as.integer(str_extract(productsReviews$ratings, "\\d+\\.\\d+"))

# Remove all emoticons from the columns.
productsReviews$title <- gsub("\\p{So}", "", productsReviews$title, perl = TRUE)

productsReviews$reviewer <- gsub("\\p{So}", "", productsReviews$reviewer, perl = TRUE)

productsReviews$review <- gsub("\\p{So}", "", productsReviews$review, perl = TRUE)

# Removing non-alphabetical languages from the columns
productsReviews$title <- gsub("[^a-zA-Z ]", "", productsReviews$title)

productsReviews$reviewer <- gsub("[^a-zA-Z ]", "", productsReviews$reviewer)

productsReviews$review <- gsub("[^a-zA-Z ]", "", productsReviews$review)


# All blank will be replace by a NA
productsReviews$title <- na_if(productsReviews$title, "")

productsReviews$reviewer <- na_if(productsReviews$reviewer, "")

productsReviews$review <- na_if(productsReviews$review, "")
```

```r
# Converting all to columns to lowercase
productsReviews <- productsReviews %>%
  mutate(across(where(is.character), tolower)) %>%
  select(-...1)

# Combined
cleanedReviews <- productsReviews %>%
  mutate(date = reviewsDataType, ratings = reviewsRatingsInteger)

# Writing to CSV
write.csv(cleanedReviews, "cleaned50ProductReviews.csv")
```