Lab Exercise 5

Charles Huervana

2024-03-15

```
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
arxiv <- read_csv("/cloud/project/Lab Exercise 5/Arxiv papers on Online Learning.csv")
```

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

## Rows: 150 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): title, author, subject, abstract, meta
## 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.
```

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


# Data type change
arxivDateType <- as.Date(arxiv_date_only, format = "%d %b %Y")
head(arxivDateType)
```

```
## [1] "2024-03-12" "2024-03-12" "2024-03-11" "2024-03-11" "2024-03-11"
## [6] "2024-03-11"
```

```
# using mutate()to transform all columns to lowercase and eliminate text within parentheses in the subj

cleanedArxiv <- arxiv %>%
  mutate(date = arxivDateType,
         subject = gsub("\\s\\(.*\\)", "", subject),
         across(where(is.character), tolower)) %>%
  select(-meta, -...1)



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

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

# Load the dataset containing Arxiv scraped reviews.
productsReviews <- read_csv("/cloud/project/Lab Exercise 5/2500Reviews.csv")
```

```
## New names:
## Rows: 2550 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 date information from the meta column and convert it into a date format.
reviewsDataType <- as.Date(str_extract(productsReviews$date, "\\d+\\s[A-Za-z]+\\s\\d+"), format = "%d %

# Convert the rating column values to integers.
reviewsRatingsInteger <- as.integer(str_extract(productsReviews$ratings, "\\d+\\.\\d+"))

# Remove all emoticons from the title, reviewer, and review 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)

# Eliminate non-alphabetical characters from the title, reviewer, and review 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)


# Replace all blank values with NA.
productsReviews$title <- na_if(productsReviews$title, "")

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

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

# Convert all columns to lowercase.
```

```r
productsReviews <- productsReviews %>%
  mutate(across(where(is.character), tolower)) %>%
  select(-...1)

# Merge the date and ratings columns into the dataset.
cleanedReviews <- productsReviews %>%
  mutate(date = reviewsDataType, ratings = reviewsRatingsInteger)

# Write the cleaned dataset to a CSV file.
write.csv(cleanedReviews, "cleaned2500Reviews.csv")
```