Lab Exercise 5

2024-03-15

Cleanig Lab Exercise 4

```
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/Datasets/Arxiv papers on Information Extraction.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.
```

```
# The meta column date will be extracted
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] "2024-03-08" "2024-03-07" "2024-03-07" "2024-03-07" "2024-03-07"
## [6] "2024-03-06"
```

```
# Removing the meta and number columns and add the new date column

# Mutating all columns, converting them to lowercase, and remove any text within parentheses in the sub

cleanedArxiv <- arxiv %>%
```

```r
  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")
```

Cleaning Lab Exercise 5

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

# Load Arxiv Scraped Dataset
productsReviews <- read_csv("/cloud/project/Lab Exercise 5/Datasets/allProds.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`
```

```r
# Extract the date information from the meta column and convert it to a date type.
reviews_date_type <- 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.
reviews_ratings_integer <- 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, "")

```

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

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

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

# Combined
cleaned_reviews <- productsReviews %>%
  mutate(date = reviews_date_type, ratings = reviews_ratings_integer)

if (!dir.exists("Cleaned Data Articles/")) {
  dir.create("Cleaned Data Articles/")
}

# Writing to CSV
write.csv(cleaned_reviews, "Cleaned Data Articles/cleaned_reviews.csv")
```