Lab_Exercise#5_Gallenero

2024-03-15

Cleaning of 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 the Arxiv Scraped Dataset
scraped_arxiv <- read_csv("/cloud/project/Lab Exercise 5/csv files/arxiv_agriculture.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 the date from the meta column
date_arxiv <- str_extract(scraped_arxiv$meta, "\\d+\\s[A-Za-z]+\\s\\d+")

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

```
## [1] "14 Mar 2024" "14 Mar 2024" "14 Mar 2024" "13 Mar 2024" "13 Mar 2024"
## [6] "13 Mar 2024"
```

```
# Removing the meta and number columns and adding the new date column.
# Mutating all of it, converting other columns to lowercase, eliminate parenthesis text in the subject

cleaned_arxiv <- scraped_arxiv %>%
  mutate(date = date_type_arxiv,
         subject = gsub("\\s\\(.*\\)", "", subject),
```

```
        across(where(is.character), tolower)) %>%
  select(-meta, -...1)

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

Cleaning of Lab Exercise 5

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

# Load Arxiv Scraped Dataset
prod50_reviews <- read_csv("/cloud/project/Lab Exercise 5/csv files/All_50Product_Reviews.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`
```

```
# Extracting the date from the meta column and changing to date type
date_type_reviews <- as.Date(str_extract(prod50_reviews$date, "\\d+\\s[A-Za-z]+\\s\\d+"), format = "%d %

# Extracting the rating from the rating column and changing to integer
reviews_ratings_integer <- as.integer(str_extract(prod50_reviews$ratings, "\\d+\\.\\d+"))

# Removing all emoticons from the columns
prod50_reviews$title <- gsub("\\p{So}", "", prod50_reviews$title, perl = TRUE)
prod50_reviews$reviewer <- gsub("\\p{So}", "", prod50_reviews$reviewer, perl = TRUE)
prod50_reviews$review <- gsub("\\p{So}", "", prod50_reviews$review, perl = TRUE)

# Removing non-alphabetical languages from the columns
prod50_reviews$title <- gsub("[^a-zA-Z ]", "", prod50_reviews$title)
prod50_reviews$reviewer <- gsub("[^a-zA-Z ]", "", prod50_reviews$reviewer)
prod50_reviews$review <- gsub("[^a-zA-Z ]", "", prod50_reviews$review)

# Replace all blank string with NA
prod50_reviews$title <- na_if(prod50_reviews$title, "")
prod50_reviews$reviewer <- na_if(prod50_reviews$reviewer, "")
prod50_reviews$review <- na_if(prod50_reviews$review, "")

# Converting all columns to lowercase
prod50_reviews <- prod50_reviews %>%
  mutate(across(where(is.character), tolower)) %>%
  select(-...1)

# Combine all together
cleaned_reviews <- prod50_reviews %>%
```

```
  mutate(date = date_type_reviews, ratings = reviews_ratings_integer)

# Writing to CSV
write.csv(cleaned_reviews, "/cloud/project/Lab Exercise 5/cleaned/cleaned_50prods_reviews.csv")
```