# LabExer#5

Rey Angelo Calopez

2024-03-17

// Lab Exercise 4 Cleaning

```
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("csvFiles/Arxiv_Data_Science.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
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)
```

```
## [1] "2024-03-14" "2024-03-14" "2024-03-14" "2024-03-14" "2024-03-14"
## [6] "2024-03-13"
```

```
# Eliminating the 'meta' and 'number' columns and appending the new 'date' column.
# Modifying all columns by converting them to lowercase, removing text within parentheses in the 'subje

cleaned_arxiv <- arxiv %>%
  mutate(date = arxiv_date_type) %>%
```

```
  mutate(subject = gsub("\\s\\(.*\\)", "", subject),
         across(where(is.character), tolower)) %>%
  select(-meta, -...1)

# writing to csv
write.csv(cleaned_arxiv, "cleaned/cleaned_arxivDataScience_Papers.csv")
```

// Lab Exercise 5 Cleaning

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

# Load Amazon Scraped Dataset
products_reviews <- read_csv("csvFiles/ScrapedAmazonProducts.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
reviews_date_type <- as.Date(str_extract(products_reviews$date, "\\d+\\s[A-Za-z]+\\s\\d+"), format = "%

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

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

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

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

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

# Combine all together
cleaned_reviews <- products_reviews %>%
  mutate(date = reviews_date_type, ratings = reviews_ratings_integer)
```

```r
# Writing to CSV
write.csv(cleaned_reviews, "cleaned/cleaned_AmazonReviews_Lab2.csv")
```