

Lab_Exercise#5

Janessa Marie Esmalla

2024-02-29

// 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("Arxiv papers on Mathematics.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] "2024-03-15" "2024-03-15" "2024-03-15" "2024-03-15" "2024-03-15"
```

```
## [6] "2024-03-15"
```

```
# 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 column
```

```
cleaned_arxivpapers <- arxiv %>%
```

```
  mutate(date = arxiv_date_type,
         subject = gsub("\\s\\((.*)\\)", "", subject),
```

```

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

# Writing to CSV
write.csv(cleaned_arxivpapers, file = "cleaned_Math_Articles", row.names = FALSE)

// Lab Exercise 5 Cleaning

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

# Load movie Scraped Dataset
movie_reviews <- read_csv("Movie Reviews.csv")

## Rows: 2350 Columns: 14
## -- Column specification -----
## Delimiter: ","
## chr (12): Movie Title, Review Title, Reviewer, Review, Date, Ratings, prod_n...
## lgl (2): ...7, type_of_movies
##
## 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 and changing to date type
reviews_date_type <- as.Date(str_extract(movie_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(movie_reviews$ratings, "\\d+\\.\\d+"))

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

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

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

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

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

# Writing to CSV
write.csv(cleaned_reviews, file = "cleaned_movie_reviews.csv", row.names = FALSE)

```

```

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

# Load movie Scraped Dataset
movie_reviews <- read_csv("Movie Reviews.csv")

## Rows: 2350 Columns: 14
## -- Column specification -----
## Delimiter: ","
## chr (12): Movie Title, Review Title, Reviewer, Review, Date, Ratings, prod_n...
## lgl (2): ...7, type_of_movies
##
## 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 and changing to date type
reviews_date_type <- as.Date(str_extract(movie_reviews$date, "\\d+\\s[A-Za-z]+\\s\\d+"), format = "%d %m %Y")

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

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

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

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

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

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

# Writing to CSV
write_csv(cleaned_reviews, file = "cleaned_movie_reviews.csv", row.names = FALSE)

```