Clean Data

Kathryn Sam

2025-12-11

Read in Files

File 1: Movie Lines

```
guess_encoding("movie_lines.txt") # Check file encoding
```

```
## # A tibble: 1 x 2
##   encoding   confidence
##   <chr>           <dbl>
## 1 ISO-8859-1       0.48
```

```
# convert to UTF-8
writeLines(iconv(readLines("movie_lines.txt"), from = "ISO-8859-1", to = "UTF8"), "movie_lines_utf.txt")

# Read in file
movie_lines<-readLines("movie_lines_utf.txt", encoding="UTF-8")

# read.table() does not handle multi-byte separators so replace them with single-byte:
movie_lines<-as.data.frame(gsub(" \\+\\+\\+\\$\\+\\+\\+ ", "@", movie_lines))

# Separate data into columns based on new separator "@"
movie_lines<-separate(movie_lines, everything(), sep="\\@", into=c("LineID", "CharID", "MovieID", "Char
```

File 2: Character metadata

```
# Newly encoded file
writeLines(iconv(readLines("movie_characters_metadata.txt"), from = "ISO-8859-1", to = "UTF8"), "movie_
char_meta<-readLines("movie_characters_metadata_utf.txt", encoding="UTF-8")
char_meta<-as.data.frame(gsub(" \\+\\+\\+\\$\\+\\+\\+ ", "@", char_meta)) #Replace separator
char_meta<-separate(char_meta, everything(), sep="\\@", into=c("CharID", "CharName", "MovieID", "MovieT
```

File 3: Movie metadata

```
writeLines(iconv(readLines("movie_titles_metadata.txt"), from = "ISO-8859-1", to = "UTF8"), "movie_titl
titles_meta<-readLines("movie_titles_metadata_utf.txt", encoding="UTF-8")
titles_meta<-as.data.frame(gsub(" \\+\\+\\+\\$\\+\\+\\+ ", "@", titles_meta)) #Replace separator
titles_meta<-separate(titles_meta, everything(), sep="\\@", into=c("MovieID", "MovieTitle", "MovieYear"
```

File 4: Writer data

```
writers<-read_excel("Writers2.xlsx")
```

Combine the datasets

```
# Merge datasets based on columns that match between the dataframes
# (i.e., CharID, CharName, MovieID, MovieTitle)

df1<-full_join(movie_lines, char_meta, relationship="many-to-one")
```

```
## Joining with `by = join_by(CharID, MovieID, CharName)`
```

```
df2<-full_join(df1, titles_meta, relationship = "many-to-one")
```

```
## Joining with `by = join_by(MovieID, MovieTitle)`
```

```
df3<-full_join(df2, writers, relationship = "many-to-one")
```

```
## Joining with `by = join_by(MovieID, MovieTitle, MovieYear)`
```

Clean the dataset

```
# Delete unwanted variables

df4<-df3[ ,-c(1,2,4,8, 10, 11, 12)] # LineID, CharID, CharName, PosInCredits,
# IMDB rating, IMDB votes, IMDB genres

#Make gender data consistently lowercase
df4$Gender <- gsub('M', 'm', df3$Gender)
df4$Gender<-gsub('F', 'f', df3$Gender)
```

Remove HTML Tags. Before:

```
print(df4$Text[168920])
```

```
## [1] "<i>I once vowed never to invest too much emotion into anyone, anything.</i>"
```

After:

```
df4 <- df4 %>%
  mutate(Text = gsub("<.+?>", "", Text))

print(df4$Text[168920])
```

```
## [1] "I once vowed never to invest too much emotion into anyone, anything."
```

Clean year column. Before:

```r
print(df4$MovieYear[55003])  # R is 1-indexed, so use 30 instead of 29
```

```
## [1] "1990/I"
```

After:

```r
df4 <- df4 %>%
  mutate(Text = gsub("<.+?>", "", Text))

print(df4$Text[168920])
```

```
## [1] "I once vowed never to invest too much emotion into anyone, anything."
```

Concatenate text by gender. Drop cases where gender is unknown.

```r
# helper to handle empty groups (returns NA if no texts)
concat <- function(x) if (length(x) > 0) paste(x, collapse = " ") else NA_character_

# For each movie, aggregate text by gender.
df5 <- df4 %>%
  group_by(MovieID) %>%
  summarize(
    MovieTitle = first(MovieTitle),
    MovieYear  = first(MovieYear),
    Text_F     = concat(Text[Gender == "f"]),
    Text_M     = concat(Text[Gender == "m"]),
    Text_unknown        = concat(Text[Gender == "?"]),
    NWriters = first(N_writers),
    WriterGender = first(Writer_Gender),
    .groups    = "drop"
  )

# Drop text_unknown
df6<-df5[ ,-c(6)]
```

Convert data to long format for further processing

```r
## Convert to long format for later analysis
df7<-pivot_longer(df6,
                  cols = c(4:5),
                  names_to = "Gender",
                  names_prefix="Text_",
                  values_to="Text"
                  )

write.csv(df7, "MovieDataClean1.csv", row.names=T)
# Export for further manual cleaning and NLP analysis
```