

What makes a fan fiction popular: Using AO3's Les Misérables fandom as example

Author: Yue Wu (210312838)

Date: June 28, 2023

1 Introduction

1.1 Background

Maybe you have been to a comic con and snapped a photo with a cosplayer dressed as your favorite character. Maybe you have seen an illustration on social media of Batman and Superman kissing. Or maybe you have heard of the best-selling book "Fifty Shades of Grey" and vaguely know that it is related to another top seller "Twilight". And then you wonder: What are these things? Who makes them? What for?

Well, congratulations, you have just entered the world of fanworks.

1.1.1 Basic concepts

Fanwork is "a creative work produced by fans" based on a "source text or event" (Fanlore, 2022). The source text, also known as **canon**, can be a book, a film, a TV series, a video game, or even a historical period. Similarly, fanworks can also take various forms. They can be memes, fictions, illustrations, comics, music, videos, cosplay, etc.. Loosely defined, any work that is based on another (usually original) source work can be considered fanwork.

When a fanwork takes the form of fiction, it is called **fan fiction**. According to OED, fan fiction is a fiction "written by a fan rather than a professional author, esp. that based on already-existing characters from a television series, book, film, etc." (Oxford English Dictionary, 2004).

Furthermore, fans of the same source text also form communities to share their works and connect with each other. A fan community around a specific source work is called a **fandom**. For example, the Harry Potter fan community is called the *Harry Potter fandom*.

1.1.2 Research on fan fiction

Since Henry Jenkins published his foundational work *Textual Poachers: Television Fans and Participatory Culture* back in 1992, extensive research has been conducted on fan fictions from literature, media, social, cultural, and gender studies perspectives, addressing issues such as literary artifact, identity, performance, and more (Hellekson & Busse, 2014).

In contrast, the computer science approach to fan fiction is relatively new, but there is a growing number of recent journal articles in this area, specially focusing on fan fiction metadata. Johnson, for example, compared the advantages and shortcomings of 3 mainstream tagging models (2014). Yin et al. used the dataset from Fanfics.com along with visual analytics methods to illustrate discrepancies between fandoms, the distribution of stories across categories, and the variations in genre preferences across different languages (2017).

Following the trend, this research will also focus on fan fiction metadata. Specifically, this research aims to answer a practical question that has not been addressed by previous studies and that may interest many aspiring fan fiction writers alike: What makes a fan fiction popular? Hopefully, this research will provide readers a glimpse of the fascinating fandom ecology and inspire more research in this area.

1.2 Aims and objectives

This research aims to analyze a few potential factors that might contribute to the popularity of fan fictions. I will begin by proposing several hypotheses and providing my rationale. Then, I will choose a suitable dataset and extract the required information. Next, I will perform data processing and employ visual analytics techniques to analyze the data and validate my hypotheses. Lastly, I will deliberate upon my findings and draw conclusions.

Below are my hypotheses:

- Hypothesis 1: If a fan fiction frequently updates rather than being a one-shot, it is more likely to be popular.

Serialized fan fictions reappear on the front page with each update, thus they are easier for readers to spot and remember. Each update is likely to draw in new readers while bringing back old ones, thus gaining more popularity. Furthermore, serialized stories are usually longer, which means richer plots and more developed characters, all of which will help to attract readers.

- Hypothesis 2: If the length of a fan fiction is within a certain range (neither too short nor too long), it is more likely to be popular.

If a fan fiction is too short, it gives the impression that the content is rough and of low quality. If it is too long, it can be daunting to new readers. Medium-length fictions are probably the most popular.

- Hypothesis 3: If a fan fiction rates higher, it is more likely to be popular.

Most fan fiction websites today use a content rating system similar to the movie rating system for the same purpose: to help readers choose content that suits their preferences and age. "General Audience" means vanilla content for all ages, while "Explicit" contains explicit sexual content or graphic violence and is restricted to readers aged 18 and older.

Given that readers often seek romance and erotica in fan fiction (Döring, 2020), incorporating such elements might increase the popularity of a fan fiction.

- Hypothesis 4: If a fan fiction is published right after a major canon event, it is more likely to be popular.

After the release of a new screen adaption or other major events, new fans will join the fandom and old fans will be more active. Fan fictions published during these times might be more likely to be noticed and read. In comparison, publishing a fan fiction during a fandom's low season might be less likely to attract readers.

- Hypothesis 5: If a fan fiction contains certain elements or depicts the main characters in a certain way, it is more likely to be popular.

Fan fictions are variants of the canon. Theoretically they can variate in any direction, but not all variates will be equally appreciated. "There are a thousand Hamlets in a thousand people's eyes", but not every interpretation of the text will be equally liked. My hypothesis is: each fandom has its own most popular paradigm, and works that fit that paradigm are more likely to be popular.

1.3 Choice of data

1.3.1 Archive of Our Own (AO3)

Archive of Our Own (AO3) is a nonprofit open source fanfiction archive. It is established by the Organization for Transformative Works (OTW) and entered open beta in 2009 (Organization for

Transformative Works, 2009). As of June 2023, it hosts most than 11 million fan works (Archive of Our Own, 2023).

This research choose to analyze the AO3 platform for the following reasons:

1. AO3 is one of the world's largest cross-fandom fan fiction archives and many fan fiction readers' and writers' first choice. The dataset is representative.
2. AO3 has a well-functioning and well-maintained rating and tagging system. Each work contains a rich set of metadata, precisely fulfilling my research requirements.
3. AO3 is nonprofit. One does not need to pay to access fan fictions or their metadata.
4. The AO3 website is clean and well structured, thus a good choice for web scraping.

1.3.2 Les Misérables fandom

Victor Hugo's *Les Misérables* is a French historical novel widely considered one of the greatest novels of the 19th century. It has been translated into multiple languages, read by numerous readers in their school years, and adapted into several films, tv series, and a musicals. In the Les Misérables fandom, the most popular topic is the relationship between Enjolras, a charismatic young revolutionary, and Grantaire, a skeptic drunk.

This research choose to study the Les Misérables fandom for the following reasons:

1. The size of the fandom

As of June 23, 2023, when I collected data from the web, the Les Misérables fandom on AO3 has 20800 English works. Comparing to more popular fandoms with nearly a million works or less popular fandoms with only hundreds of works, the Les Misérables fandom is of medium size. The size of metadata is large enough for analytical purposes, but not too large to be computationally challenging.

2. The timeline of the fandom

Hugo's original novel was published in 1862. The renowned West End English-language musical has been running since 1985. The Les Misérables fandom has existed for a long time before the establishment of AO3.

Since the launch of AO3 in 2009, the Les Misérables fandom has experienced 3 major events:

- The 2010 concert celebrating the 25th anniversary of the musical ("Les Misérables in Concert: The 25th Anniversary (2010) - IMDb," 2010)
- The 2012 film adaption of the musical, which won 3 Oscars ("Les Misérables (2012) - IMDb," 2012)
- The 2018 BBC miniseries adaption of the novel, which scores 7.8 on IMDb ("Les Misérables (TV Mini Series 2018–2019) - IMDb," 2019)

The fact that this fandom has existed for a long time and experienced multiple major events makes it a good choice for testing Hypothesis 4.

3. The author of this research is familiar with the Les Misérables fan community and have prior empirical knowledge of it. It will be easier for me to interpret the data.

1.3.3 Other dataset considered

Two other ready-made datasets were considered for this research:

1. [Harry Potter fanfiction dataset](#)

This dataset contains all Harry Potter fan fiction metadata on fanfiction.net between 2001-2019 in all available languages. It is a clean, well-structured csv file with 648494 entries. It is a good choice for testing Hypothesis 1 and 2. However, it does not contain information about the rating of the fan fictions, thus cannot be used to test Hypothesis 3. It also does not contain descriptive tags, thus cannot be used to test Hypothesis 5.

2. [Ao3: Selective data dump for fan statisticians](#)

This is a 1.55 GB third-party AO3 dataset. It is comprehensive, with detailed tag information, but it is too large for my computational capacity.

With everything considered, I decided to conduct web scraping on AO3 myself.

1.3.4 Scope and limitations

1. This research is a case study of the AO3 Les Misérables fandom. It will not discuss other platforms or fandoms.
2. This research will only analyze the metadata of fan fictions, not the content of fan fictions.
3. Only English works will be included in this research, as multi-language data processing posts additional challenges beyond the scope of this research.

It is worth noticing that "fandom" is not a monolith. It is an umbrella term that covers a wide range of individuals with diverse cultures, identities, interests, and engagements. Although they share "fan-ness" in common, we still expect a sports celebrity fandom to differ significantly from a Japanese manga fandom. Conclusions drawn for one fandom may not apply to all.

The platform's nature also influences our conclusions. Fan behavior is expected to vary across archives, social media platforms, and forums.

Again, this research is a case study of the AO3 Les Misérables fandom. It will help us look into the the world of fan fictions, but its conclusions may not apply universally.

1.3.5 Ethical considerations

1. Copyright

According to AO3's Terms of Service, OTW "does not claim any ownership or copyright in your (users') content" (Archive of Our Own, 2018). The same article further declares, "the OTW believes that transformative fanworks are legal...transformative use is defined by the OTW as adding something new, with a further purpose or different character, altering the source with new expression, meaning, or message" (Archive of Our Own, 2018).

The copyright of each fan fiction belongs to its writer. The ownership or copyright of the metadata of fictions is not explicitly stated, thus can either belong to the authors or to AO3. Hence, the dataset is credited to the author of each fan fiction and the AO3 platform. The data processing, analysis, and conclusion of this research are my own.

2. Web scraping policy

The Organization for Transformative Works website states, "we (OTW) don't have a policy against responsible data collection — such as those done by academic researchers..." (Eskici, 2023).

Since this research only collects metadata of fictions for academic purposes, it should not violate website policies or infringe copyright.

3. Data privacy

The data is readily accessible in my notebook. The data in this research is not anonymized. The work ID and author ID are listed in the dataset for several reasons: 4. Potential harm

- It is part of the metadata publicly available on the AO3 website.
- This research can be considered as an experimental literature analysis. The credits need to be given to the authors of each text.
- Work ID and author ID are potentially important for further research/ onward usage. For example, the author ID can be used to calculate the number of works each author published. The work ID can be used to find the full text of the work on the AO3 website.
- The author ID is not linked to the author's real name or email address. The potential harm or misuse of personal information is minimal.

The potential harm of this research is minimal. The dataset does not contain any personal information. The conclusions of this research are not likely to be used to discriminate against any group of people or to produce dangerous or harmful assumptions.

2 Web scraping

The first step in hypothesis testing is to collect necessary data. For this research, I will collect the metadata of all English fan fictions in the Les Misérables fandom on the AO3 platform by web scraping. As of June 23, 2023, the fandom has 20800 English works spread across 1040 pages.



2.1 Define key variables

First, let's define some key variables for web scraping.

1. The HTTP request header User-Agent string I use here is a common one for Mac system Chrome browser users. This is for responsible web scraping.
2. The AO3 URL structure is as follows:
 - <https://archiveofourown.org/> is the website base URL.
 - We are in the "work_search" page.
 - I manually choose to sort worked by "Date Posted (Newest First)", thus "sort_column=created_at".
 - I also manually choose to filter English works only, thus "language_id=en".
 - To sort and filter, "commit=Sort+and+Filter".
 - I want works in "Les Misérables - All Media Types" fandom only, thus "tag_id=Les+Misérables+-+All+Media+Types"
 - For web scraping, the only variable is the page number. "page=" or "page=1" means the first page, "page=2" means the second page, and so on.
3. Visually examine the search page, it is easy to spot that the highest page number is 1040. Combining the base URL and the page number variable, we will be able to get the URL for each page.

```
In [ ]: # define key variables
headers = {
    "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
```

```
(KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36"
}
url = "https://archiveofourown.org/works?
work_search%5Bsort_column%5D=created_at&work_search%5Blanguage_id%5D=en&commit=Sort+and+Fi
lter&tag_id=Les+Mis%C3%A9rables+--+All+Media+Types&page="
max_page_number = 1040
```

2.2 Import necessary libraries and modules

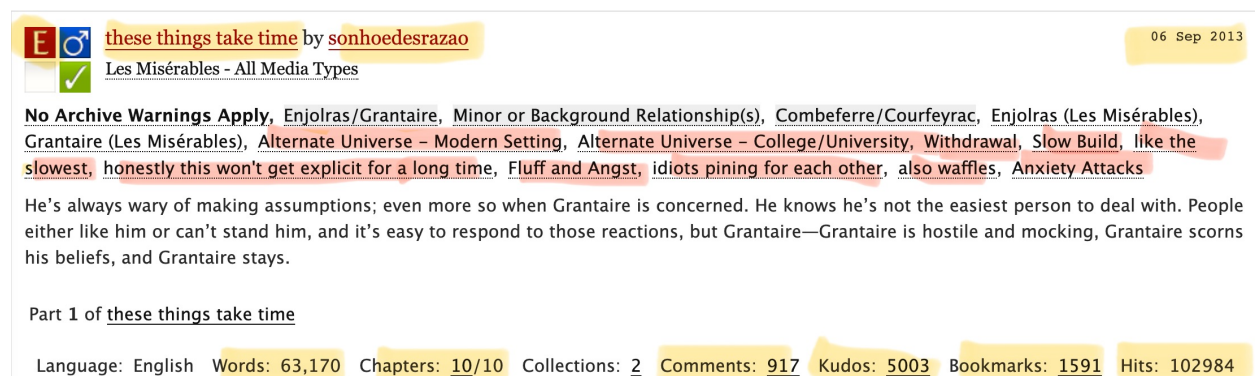
```
In [ ]: # Import necessary libraries and modules
# for web scraping
import csv
import requests
from bs4 import BeautifulSoup
from time import sleep
import random

# for data manipulation
import re
from datetime import datetime
import numpy as np
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
from unicodedata import unidecode
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from collections import Counter

# for data visualization
import seaborn as sns
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Tell Jupyter how to display plots
%matplotlib inline
```

2.3 Define data structure



these things take time by **sonhoedesrazao** 06 Sep 2013

Les Misérables - All Media Types

No Archive Warnings Apply. Enjolras/Grantaire, Minor or Background Relationship(s), Combeferre/Courfeyrac, Enjolras (Les Misérables), Grantaire (Les Misérables), Alternate Universe – Modern Setting, Alternate Universe – College/University, Withdrawal, Slow Build, like the slowest, honestly this won't get explicit for a long time, Fluff and Angst, idiots pining for each other, also waffles, Anxiety Attacks

He's always wary of making assumptions; even more so when Grantaire is concerned. He knows he's not the easiest person to deal with. People either like him or can't stand him, and it's easy to respond to those reactions, but Grantaire—Grantaire is hostile and mocking, Grantaire scorns his beliefs, and Grantaire stays.

Part 1 of **these things take time**

Language: English Words: 63,170 Chapters: 10/10 Collections: 2 Comments: 917 Kudos: 5003 Bookmarks: 1591 Hits: 102984

On the AO3 search page, each work is displayed in a structure shown above. Given the objectives of this research, I will collect the following metadata (also marked in the screenshot above):

- Work ID. Work ID is the unique ID of each work. It is a string of numbers and also the last part of the URL that links to the work. For example, if the URL of the work is <https://archiveofourown.org/works/123456>, the work ID is 123456. It also means that this is the 123456th work posted on AO3.
- Author name
- Date updated

- Words
- Chapters
- Kudos. Kudos means "likes".
- Bookmarks
- Hits. Hits means "views".
- Rating. Rating in on the top left corner of the work. The work in the screenshot, for example, is rated "E", which means "Explicit".
- Freeform tags. AO3 has a categorized tagging system. "Explicit", for example, is a required rating tag. "No Archive Warnings Apply" is a required warning tag. "Les Misérables - All Media Types" is a fandom tag. "Enjolras/Grantaire" is a relationship tag. "Enjolras (Les Misérables)" is a character tag. Freeform tags, marked in red, are tags that don't fit into the above categories. In the freeform tags section, writers can add anything they want. A freeform tag can be a descriptive word, a phrase, a sentence, or anything. These tags give us additional expressive information that controlled-vocabulary tags can't provide.

Considering the scope of this research, I will store the data in a csv file. Each row will represent a work. Each column will represent a piece of metadata.

```
In [ ]: # help function to prepare a csv file for storing the data
def csv_prep():
    # define the headers we need for this project
    header = [
        "work_id",
        "Author",
        "Date_updated",
        "Words",
        "Chapters",
        "Kudos",
        "Bookmarks",
        "Hits",
        "Rating",
        "Freeform_tags",
    ]
    # create a csv file with those headers
    with open("fanfic_data.csv", "w", newline="") as fanfic_data:
        csv.writer(fanfic_data).writerow(header)
```

2.4 Scrape one fiction

I use the BeautifulSoup4 library to select HTML elements for each piece of metadata and then store them in the corresponding column.

The helper function to scrape one fiction is as follows:

```
In [ ]: # helper function for one fan fiction
def fic_to_csv_row(fic, csv_writer):
    # work_id
    # work id is the unique id of each fan work and the last part of the url which links
    to the fan work
    # e.g., https://archiveofourown.org/works/12345678, the work id is 12345678
    # work id is a link with "h4" tag and "heading" class
    work_id = fic.select_one("h4.heading").select_one("a").get("href").split("/")[-1]

    # author
    # author is a link with "rel=author" attribute
    # if there is no author (e.g., when the author orphaned the work), set author to
```

```

"Anonymous"
author = (
    fic.select_one("a[rel=author]").get_text()
    if fic.select("a[rel=author]")
    else "Anonymous"
)

# date_updated
# date_updated has "p" tag and "datetime" class
date_updated = fic.select_one("p.datetime").get_text()

# words
# words has "dd" tag and "words" class
# if the fan work has no word count (e.g., a video), set words to 0
words = (
    fic.select_one("dd.words").get_text().replace(",", "")
    if fic.select_one("dd.words").get_text()
    else 0
)

# chapters
# if the work has multiple chapters, chapters is a link with "dd" tag and "chapters"
class
# if the work has only one chapter, the webpage shows plaintext "1/1"
chapters = (
    fic.select_one("dd.chapters").select_one("a").get_text()
    if fic.select_one("dd.chapters").select_one("a")
    else 1
)

# kudos
# kudos has "dd" tag and "kudos" class
# if the work has no kudos, set kudos to 0
kudos = (
    fic.select_one("dd.kudos").select_one("a").get_text()
    if fic.select("dd.kudos")
    else 0
)

# bookmarks
# bookmarks has "dd" tag and "bookmarks" class
# if the work has no bookmarks, set bookmarks to 0
bookmarks = (
    fic.select_one("dd.bookmarks").select_one("a").get_text()
    if fic.select("dd.bookmarks")
    else 0
)

# hits
# hits has "dd" tag and "hits" class
# if the work has no hits, set hits to 0
hits = fic.select_one("dd.hits").get_text() if fic.select("dd.hits") else 0

# rating
# rating has "span" tag and "rating" class
# for A03, rating is mandatory, so no need to check if it exists
rating = fic.select_one("span.rating").get_text()

# freeform_tags
# freeform tags have "li" tag and "freeforms" class
tag_list = fic.select("li.freeforms")
# freeform tags should be a list of tags
# we can do some primitive text cleaning here
# 1. remove non-alphanumeric characters
# 2. convert all text to lowercase
# 3. remove extra whitespaces
tags = [
    re.sub(

```



```

        "[^A-Za-z0-9À-ÖØ-öø-ÿ]+", " ", tag.select_one("a").get_text().lower()
    ).strip()
    for tag in tag_list
]

# write to csv
csv_writer.writerow(
    [
        work_id,
        author,
        date_updated,
        words,
        chapters,
        kudos,
        bookmarks,
        hits,
        rating,
        tags,
    ]
)

```

2.5 Scrape one page

For each page, I use requests library to get the HTML and beautifulsoup4 library to parse it.

The major challenge here is to not get blocked by the website. To avoid this, I use the sleep function to show down my scraping speed.

The helper function to scrape one page is as follows:

```

In [ ]: # helper function for one webpage
def page_to_csv(url, headers, page_number, csv_writer):
    # As discussed above, the url of each page is the base URL plus the page number variable.
    url = url + str(page_number)

    # access the webpage
    r = requests.get(url, headers=headers)

    # if the webpage is not accessible (e.g., network issues, too many requests, etc.)
    # wait for 5 minutes and try again
    while r.status_code != 200:
        sleep(300)
        r = requests.get(url, headers=headers)
        print("Failed to access page " + str(page_number) + ", retrying...")
    # otherwise, by default, conduct one scraping every 5-10 seconds as a responsible approach
    sleep(random.choice(range(5, 10)))

    # import the html into BeautifulSoup
    soup = BeautifulSoup(r.text, "lxml")

    # find all the fictions on the page
    # each page has 20 fictions
    # each fiction is a <li> element with class "work blurb group"
    fics = soup.select("li.work.blurb.group")

    # write each fiction to csv
    for fic in fics:
        fic_to_csv_row(fic, csv_writer)

```

2.6 Scrape all pages

I use exception handling for unforeseeable errors. If there's a problem with the scraping process, I can check it manually, debug my code, and make it more robust.

Fortunate though, my scraping process went smoothly without any issues.

```
In [ ]: # this cell takes 2-3 hours to run, so please skip it and use my scraped data directly
# if you really want to run this cell, please comment the next line
%%script echo skipping

# scraping time!
csv_prep()

with open("fanfic_data.csv", "a", newline="") as fanfic_data:
    csv_writer = csv.writer(fanfic_data, delimiter=",")
    # scrape each page
    try:
        for page_number in range(1, max_page_number + 1):
            page_to_csv(url, headers, page_number, csv_writer)
        # if something goes wrong, print the error message and the page number for manual
inspection
    except Exception as e:
        print(e)
        print(
            "Trouble with page " + str(page_number) + ". Please check what went wrong."
        )
```

2.7 Examine scraped data

After the web scraping is done, the first thing to do is to examine the scraped data to check its integrity:

- How does our csv file look so far?
- Did we scrape all 20800 works?
- Are there any missing values?

The results look good:

- Each column has the correct data stored in it.
- There are 20800 rows as expected.
- There is no missing value.

```
In [ ]: # check the results of the scraping
with open("fanfic_data.csv", "r") as fanfic_data:
    fanfic_data = pd.read_csv(fanfic_data)

# an intuitive preview of the scraped data:
print("A preview of the scraped data: \n", fanfic_data.head())
# did we scrape all the fictions?
print("\n The total number of scraped fan fictions: ", len(fanfic_data))
# are there any missing values?
print("\n The number of missing values in each column: \n", fanfic_data.isna().sum())
```

A preview of the scraped data:

	work_id	Author	Date_updated	Words	Chapters	Kudos	Bookmarks	\
0	48038584	StoryReader01	21 Jun 2023	2343	1	0	0	
1	48025039	Pallas_TM	21 Jun 2023	1714	1	6	1	
2	48006547	AmrevHistoryNerd	20 Jun 2023	1160	1	2	0	
3	47974648	combeauferre	19 Jun 2023	10146	1	0	0	
4	47928559	GEGabriels	17 Jun 2023	2229	1	6	0	

	Hits	Rating	\
0	8	General Audiences	
1	25	Teen And Up Audiences	
2	17	General Audiences	
3	27	Explicit	
4	65	General Audiences	

	Freeform_tags
0	['fantine keeps cosette', 'alternate universe ...
1	['alternate universe modern setting', 'alterna...
2	['fluff', 'babies']
3	['puppy play', 'puppy courf', 'dom ferre', 'su...
4	['friendship', 'brotherhood', 'fluff', 'sickfi...

The total number of scraped fan fictions: 20800

The number of missing values in each column:

work_id	0
Author	0
Date_updated	0
Words	0
Chapters	0
Kudos	0
Bookmarks	0
Hits	0
Rating	0
Freeform_tags	0

dtype: int64

2.8 Afterthoughts

After I finished scraping the web, I reflect on my web scraping process and think about how to improve it in the future.

1. It would have been better to scrape all available metadata on the search page, not just the selected ones I needed for this research. Scraping all the metadata would not take much longer but would provide a more complete dataset that can be used for other projects.
2. I should have optimized the data type for each column during the scraping. For example, the date updated column is stored as text, but it could have been converted to a proper date format. This would make the dataset more accurate and reusable for other projects.
3. I scraped the web without authentication settings. A few days later, when I logged into my AO3 account, I accidentally found that the number of works on the same search page had increased by around 10% (around 2000 works). After some investigation, I realized that it is because some authors choose not to show their works to guest users. This means that my dataset is not complete. Although it should not affect the validity of my dataset and authors' preferences should be respected, it serves as a reminder that I should have researched the website more before conducting the web scraping.
4. Fortunately, my web scraping went well without problems thanks to the reliable and responsive AO3 website. But theoretically several things could have gone wrong. For example, what if my code throws an error after it scraped 1000 pages already? What if someone posts a new work while I am scraping? What if the internet is slow? I should have included additional functions to manage these scenarios and maintain the integrity of my dataset.

3 Data processing

3.1 Convert csv to pandas DataFrame

Before the actual data analysis, we need to perform some basic data processing steps. The initial step is to change the csv file into a pandas DataFrame.

```
In [ ]: # convert the scraped data into a pandas DataFrame
with open("fanfic_data.csv", "r") as fanfic_data:
    fanfic_data = pd.read_csv(fanfic_data)
fanfic_df = pd.DataFrame(fanfic_data)
fanfic_df.index = range(0, len(fanfic_df))
```

3.2 Convert the data type of "Date_updated" to datetime

Currently, the column "date_updated" is in the string data type. It needs to be converted to the correct datetime data type.

```
In [ ]: # convert the date_updated column to datetime
fanfic_df["Date_updated"] = [
    datetime.strptime(x, "%d %b %Y") for x in fanfic_df["Date_updated"]
]
```

3.3 Calculate potential post date using update date (failed)

To test my fourth hypothesis about the relationship between the timing of publication and popularity, we need to obtain the release date of the work. Unfortunately, the AO3 website does not provide this information on the search page but only on the work page. Considering we are talking about 20800 works here, scraping 20800 more webpages just to retrieve a small piece of information is not economical.

My initial plan is to estimate the post date using the update date. Here's the logic:

1. The update date is available on the search page.
2. For each work, the update date can either be the same as the post date (for one-shots) or later (for multi-chapter works).
3. Before scraping, I have already sorted all works by "Date Posted (Newest First)". Hence, we know that the post date is in descending order.
4. So, let's assume the update date is the post date initially. Then, for each work, if the post date of the current fan fiction is later than the post date of the previous fan fiction, we replace it with the previous fan fiction's post date.

This approach should give us a rough estimate of the post date. The code is as follows:

```
In [ ]: # Do not run this cell
%%script echo skipping

# insert a new column "Date_likely_posted" with its initial value being the same as
"Date_updated"
# since most fan fictions have only one chapter, for them, the post date is the same as
the update date
fanfic_df.insert(2, "Date_likely_posted", fanfic_df["Date_updated"], True)

# because we sorted all fan fictions by Date Posted on the website before scraping
# we know that the post date is in descending order
```

```
# if the post date of the current fan fiction is later than the post date of the previous
fan fiction,
# we replace it with the post date of the previous fan fiction
for i in range(1, len(fanfic_df)):
    if fanfic_df.iloc[i, 2] > fanfic_df.iloc[i - 1, 2]:
        fanfic_df.iloc[i, 2] = fanfic_df.iloc[i - 1, 2]
```

After running the code above, however, the results are not as expected. After more investigation, I found the reason:

There are actually three timestamps for each work: the publish date, the post date, and the update date. The post date is the date when the work is posted on AO3. The publish date is the date when the work is first published. The update date is the date when the work is last updated.

The update date can be found on the search page, the publish date on the work page. The post date is not available anywhere on the AO3 website but only indicated by each work's ID.

AO3 supports importing works from other websites. In such cases, the publish and update dates may precede the post date. For example, a work could have been published in 2002, updated in 2003, but only posted on AO3 in 2012.

Consequently, the update date can be earlier or later than the post date. The previous method of estimating the post date is no longer valid. I have not discovered a suitable approach to estimate the post date using the search page information. For the rest of this research, the update date will be used instead of the post date.

3.4 Convert ratings to numerical values

Since ratings are ordinal data in text format, converting them to numerical values will help us perform statistical analysis later. I use `OrdinalEncoder()` from the `scikit-learn` library to perform this task.

The code is as follows:

```
In [ ]: # use ordinal encoding to convert the "Rating" column to numeric values
# the higher the rating, the higher the value
# "Not Rated" = 0, "General Audiences" = 1, "Teen And Up Audiences" = 2, "Mature" = 3,
"Explicit" = 4
enc = OrdinalEncoder(
    categories=[
        [
            "Not Rated",
            "General Audiences",
            "Teen And Up Audiences",
            "Mature",
            "Explicit",
        ]
    ]
)

# create a new column "Rating_num" for the numeric rating values
fanfic_df["Rating_num"] = enc.fit_transform(fanfic_df[["Rating"]])
```

3.5 Group works by their kudo counts

This research aims to determine the factors contributing to the popularity of fan fiction. I measure popularity by the number of likes (*kudos*) a work receives. For analysis purposes, we can group works into different categories based on their kudo counts, such as the top 5% most liked works, the top 10% most liked works, and so on.

The code is as follows:

```
In [ ]: # use quantile to divide the Kudos column into 5 categories
kudos_top_5pct = fanfic_df["Kudos"].quantile(0.95)
kudos_top_10pct = fanfic_df["Kudos"].quantile(0.9)
kudos_top_25pct = fanfic_df["Kudos"].quantile(0.75)
kudos_top_50pct = fanfic_df["Kudos"].quantile(0.5)

# helper function to categorize Kudos
def categorize_kudos(x):
    if x > kudos_top_5pct:
        return "Top 5 pct"
    elif x > kudos_top_10pct:
        return "5 - 10 pct"
    elif x > kudos_top_25pct:
        return "10 - 25 pct"
    elif x > kudos_top_50pct:
        return "25 - 50 pct"
    else:
        return "Lower 50 pct"

# create a new column "Kudos_category" for Kudos categories
fanfic_df["Kudos_category"] = fanfic_df["Kudos"].apply(categorize_kudos)
```

3.6 Optimize data types

```
In [ ]: fanfic_df = fanfic_df.convert_dtypes()
```

4 Data analysis

4.1 Basic statistics

```
In [ ]: # basic statistics of all fan fictions
fanfic_df.describe()
```

```
Out[ ]:
```

	work_id	Words	Chapters	Kudos	Bookmarks	Hits	Rating_num
count	20800.0	20800.0	20800.0	20800.0	20800.0	20800.0	20800.0
mean	10626873.429087	5457.900529	2.205529	122.24024	14.763077	1975.290433	1.736875
std	12515284.70101	14990.56399	4.839848	285.037392	61.698613	5251.698352	1.111568
min	1579.0	0.0	1.0	0.0	0.0	0.0	0.0
25%	985271.5	965.0	1.0	19.0	1.0	325.0	1.0
50%	4265283.0	1921.0	1.0	46.0	3.0	760.0	2.0
75%	17411911.25	4206.0	1.0	119.0	10.0	1829.0	2.0
max	48038584.0	688405.0	252.0	12633.0	3970.0	340886.0	4.0

For the basic statistics above, one can see that:

- The differences between different works are huge.
- Most fan fictions are short stories. The average length is 5458 words. The middle length is 1921 words. However, there are also long fan fictions with 0.6 million words.

- Most fan fictions are one shots. But there are also extreme works with 252 chapters.
- 75% fan fictions receives fewer than 2000 views, fewer than 120 kudos, and fewer than 10 bookmarks.
- But the most popular fan fictions are extremely popular, receiving 340K views, 12K kudos, and 4K bookmarks.

```
In [ ]: # basic statistics of the top 10% most liked fan fictions
top_10pc_fanfic = fanfic_df[
    (fanfic_df["Kudos_category"] == "Top 5 pct")
    | (fanfic_df["Kudos_category"] == "5 - 10 pct")
]
top_10pc_fanfic.describe()
```

	work_id	Words	Chapters	Kudos	Bookmarks	Hits	Rating_num
count	2075.0	2075.0	2075.0	2075.0	2075.0	2075.0	2075.0
mean	4922073.884819	15497.54747	4.021687	669.111807	98.849639	9998.595181	2.20241
std	6589176.651508	35178.966136	10.350838	670.756263	172.705737	13664.375428	1.223955
min	198246.0	0.0	1.0	267.0	2.0	1322.0	0.0
25%	942657.5	2227.5	1.0	334.0	31.0	4187.5	1.0
50%	1906524.0	4676.0	1.0	444.0	51.0	6373.0	2.0
75%	5647823.5	13161.0	2.0	726.0	98.0	10963.5	3.0
max	44145723.0	688405.0	252.0	12633.0	3970.0	340886.0	4.0

I define popularity based on the number of likes (*kudos*) a work receives. Comparing to all fan fictions in general, the top 10% most popular fan fictions:

- are 3 times longer
- are still mostly one shots, but have more chapters on average
- receive 10 times more views, more kudos, and more bookmarks
- rate higher

4.2 Preparing for data visualization

1. Set styles

```
In [ ]: # set seaborn styles
sns.set_style("ticks")
sns.set_context("notebook")
colors = ["#6A4A3C", "#00A0B0", "#EDC951", "#EB6841", "#CC333F"]
p = sns.color_palette(colors)
sns.palplot(p)
```



2. Remove outliers

The differences between fan fictions can be very huge. Removing outliers helps provide a clearer picture of the overall distribution and general pattern. It also enhances the readability of graphs.

```
In [ ]: # remove upper 0.05% outliers (and lower 0.05% outliers too if applicable)
filtered_fanfic = fanfic_df.copy()
for col in ["work_id", "Date_updated", "Words"]:
    filtered_fanfic = filtered_fanfic[
        (filtered_fanfic[col] < filtered_fanfic[col].quantile(0.9995))
        & (filtered_fanfic[col] > filtered_fanfic[col].quantile(0.0005))
    ]

for col in ["Chapters", "Kudos", "Bookmarks", "Hits"]:
    filtered_fanfic = filtered_fanfic[
        filtered_fanfic[col] < filtered_fanfic[col].quantile(0.9995)
    ]

filtered_fanfic.index = range(0, len(filtered_fanfic))
```

4.3 The relationship between popularity and story updates

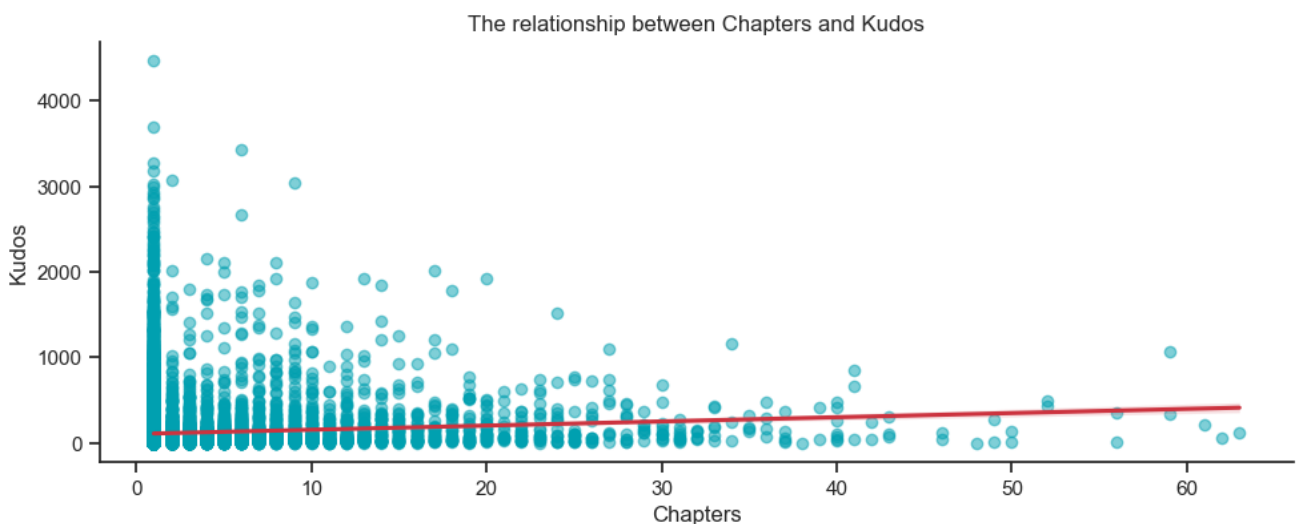
Hypothesis 1: If a fan fiction frequently updates rather than being a one-shot, it is more likely to be popular.

For the first hypothesis, I want to find out whether more updates will increase the popularity of a fan fiction. I define popularity by the number of kudos a work receives. I measure updates by the number of chapters a work has. By using a linear regression model, we can see the relationship between the two variables.

```
In [ ]: # convert data types to int
filtered_fanfic["Chapters"] = filtered_fanfic["Chapters"].astype(int)
filtered_fanfic["Kudos"] = filtered_fanfic["Kudos"].astype(int)

# plot the relationship between Chapters and Kudos using linear regression model
sns.lmplot(
    data=filtered_fanfic,
    x="Chapters",
    y="Kudos",
    height=4,
    aspect=2.5,
    line_kws={"color": "#CC333F"},
    scatter_kws={"color": "#00A0B0", "alpha": 0.5},
)

plt.title("The relationship between Chapters and Kudos")
plt.show()
```



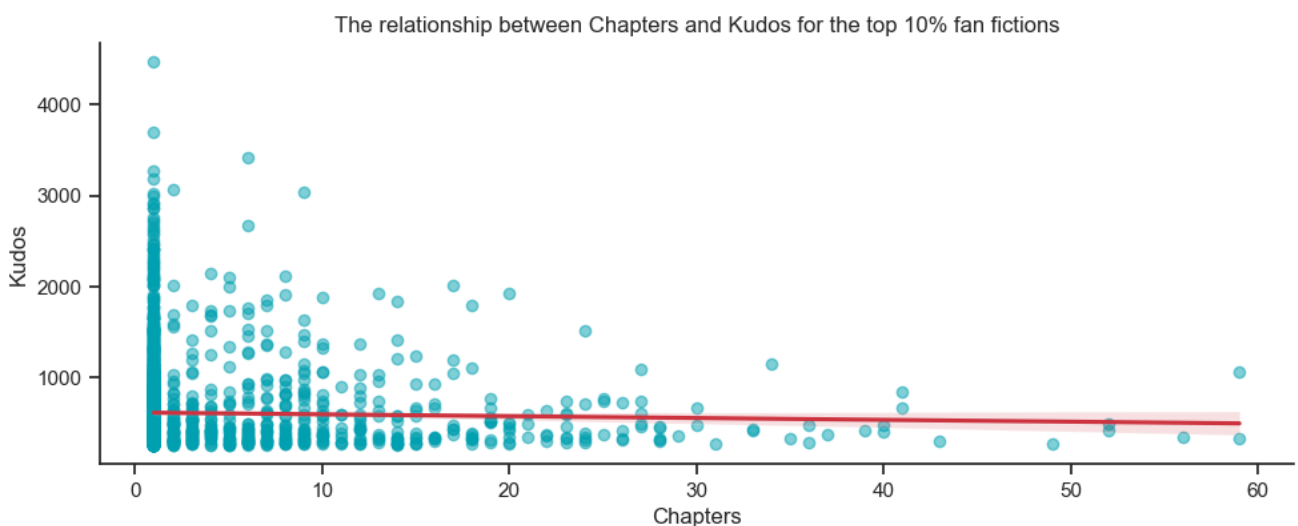
From the chart above, we can see that the number of updates is positively correlated with the number of kudos. The more updates a work has, the more kudos it receives. The correlation coefficient, however, is not very strong.

It is possible that those one-shots with very few kudos (which means, most of the works) heavily affects the result. Let us focus on the top 10% most popular works and reexamine the correlation.

```
In [ ]: # select the top 10% fan fictions
featured = filtered_fanfic[
    filtered_fanfic["Kudos"] > filtered_fanfic["Kudos"].quantile(0.9)
]

# plot the relationship between Chapters and Kudos using nonparametric lowess model
sns.lmplot(
    data=featured,
    x="Chapters",
    y="Kudos",
    height=4,
    aspect=2.5,
    line_kws={"color": "#CC333F"},
    scatter_kws={"color": "#00A0B0", "alpha": 0.5},
)

plt.title(r"The relationship between Chapters and Kudos for the top 10% fan fictions")
plt.show()
```



If we only consider the top 10% most popular fan fictions, the correlation between the number of chapters and the number of kudos becomes even weakly negative. This means that having more updates does not necessarily increase the popularity of a fan fiction.

Several reasons could explain this result:

1. AO3 is an archive, not a forum or social media platform. Active updating does not necessarily attract more attention. In fact, it could be the opposite: readers who visit archives may prefer to read completed works.
2. Most fan fictions are one shots. Even among the top 10% most liked fan fictions, more than half of them still consist of only one chapter. The majority of standalone works prevent us from observing the impact of frequent updates on multi-chaptered works.

4.4 The relationship between popularity and fiction length

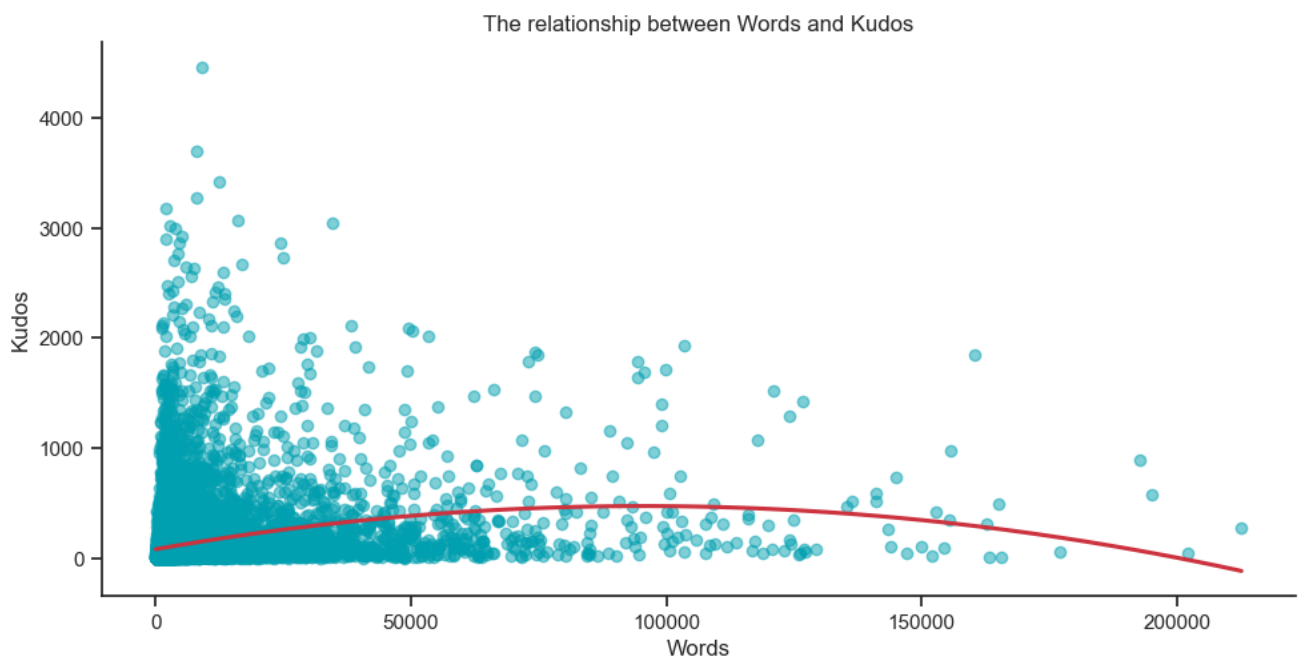
Hypothesis 2: If the length of a fan fiction is within a certain range (neither too short nor too long), it is more likely to be popular.

To test the second hypothesis, we need to use a binomial regression model to examine the relationship between the length of a fan fiction and its popularity, with the number of kudos as the dependent variable and the number of words as the independent variable.

```
In [ ]: # convert data types to int
filtered_fanfic["Words"] = filtered_fanfic["Words"].astype(int)
filtered_fanfic["Kudos"] = filtered_fanfic["Kudos"].astype(int)

# plot the relationship between Words and Kudos using binomial regression model
sns.lmplot(
    data=filtered_fanfic,
    x="Words",
    y="Kudos",
    order=2,
    ci=None,
    height=5,
    aspect=2,
    line_kws={"color": "#CC333F"},
    scatter_kws={"color": "#00A0B0", "alpha": 0.5},
)

plt.title("The relationship between Words and Kudos")
plt.show()
```



The chart above shows, it is true that fan fictions within a specific range of length tend to be more popular. Particularly, fan fictions comprising approximately 100,000 words are the most popular. Popularity decreases when the length either exceeds or falls short of this range.

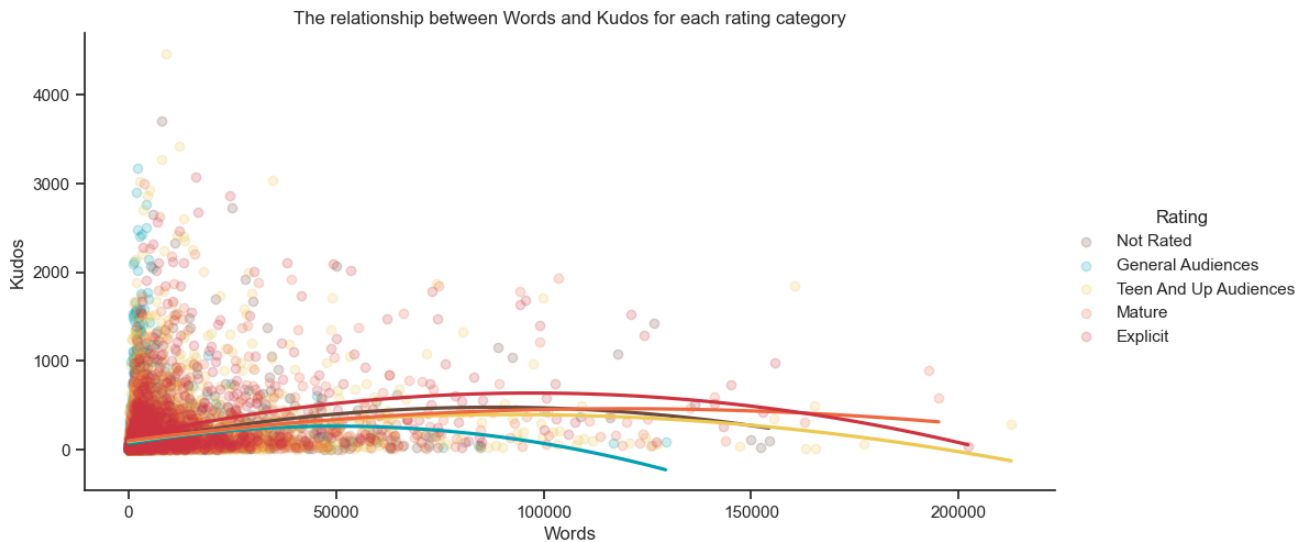
```
In [ ]: # for each rating category
# plot the relationship between Words and Kudos using polynomial regression model
sns.lmplot(
    data=filtered_fanfic,
    x="Words",
    y="Kudos",
    hue="Rating",
    hue_order=[
        "Not Rated",
        "General Audiences",
        "Teen And Up Audiences",
    ],
)
```

```

        "Mature",
        "Explicit",
    ],
    order=2,
    ci=None,
    height=5,
    aspect=2,
    palette=p,
    scatter_kws={"alpha": 0.2},
    facet_kws={"legend_out": True},
)

plt.title("The relationship between Words and Kudos for each rating category")
plt.show()

```



If we further examine the correlation between fiction length and fiction popularity in each rating category, we can see that audiences are less patient with vanilla works. When fiction is rated for general audiences, its popularity decreases when it exceeds about 50,000 words. If works are rated Mature or Explicit, however, longer length is preferred. For works with an explicit rating, the popularity peaks when the length is around 100,000 words.

4.5 The relationship between popularity and content rating

Hypothesis 3: If a fan fiction rates higher, it is more likely to be popular.

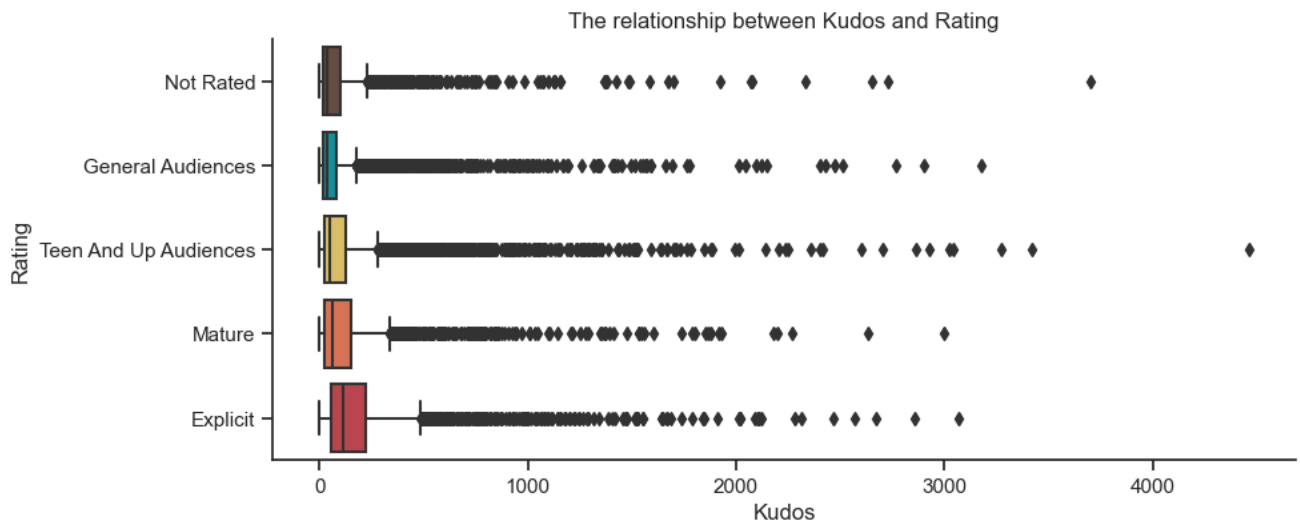
To verify the third hypothesis, we need to use a box chart.

```

In [ ]: # plot the relationship between Kudos and Rating using box plot
sns.catplot(
    kind="box",
    data=filtered_fanfic,
    x="Kudos",
    y="Rating",
    order=[
        "Not Rated",
        "General Audiences",
        "Teen And Up Audiences",
        "Mature",
        "Explicit",
    ],
    orient="h",
    height=4,
    aspect=2.5,
    palette=p,
)

```

```
plt.title("The relationship between Kudos and Rating")
plt.show()
```



Hypothesis 3 is correct. Fan fictions with a higher content rating receive more kudos for all quantiles. Audiences love erotica.

4.6 The relationship between popularity and the timing of publication

Hypothesis 4: If a fan fiction is published right after a major canon event, it is more likely to be popular.

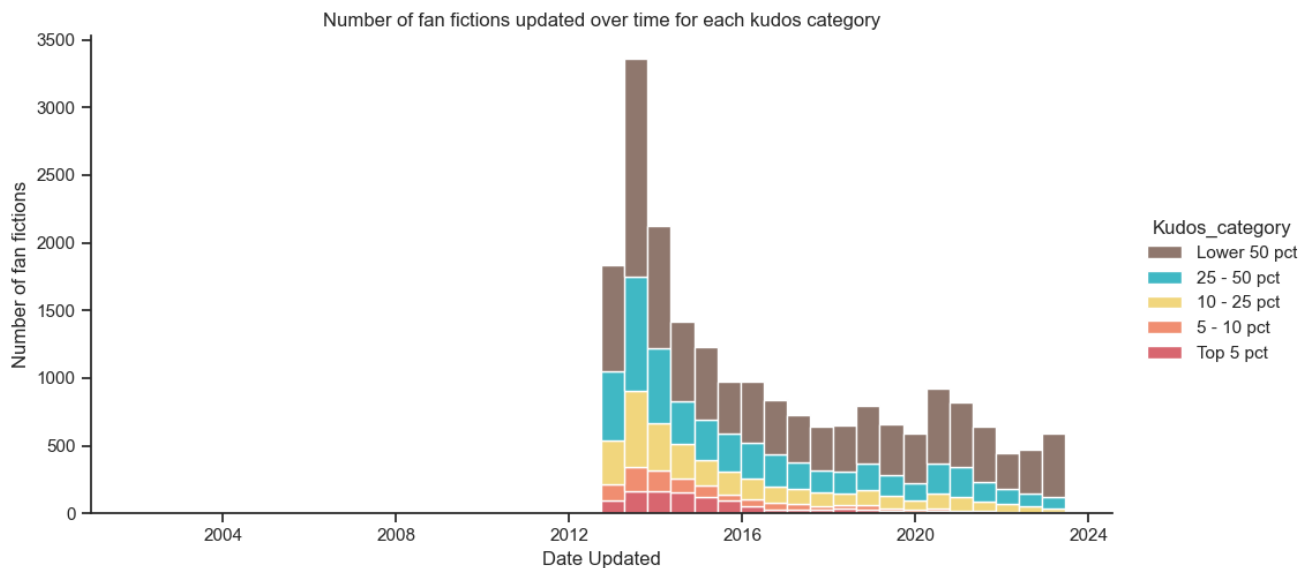
Since the launch of AO3 in 2009, the Les Misérables fandom has lived 3 major events:

- October 3rd, 2010: Les Misérables in Concert: The 25th Anniversary
- December 5th, 2012: Les Misérables (2012 film)
- December 30th, 2018: Les Misérables (British TV series)

To evaluate the fourth hypothesis, we must first examine the monthly updates of fan fictions to see if there are increased activity periods following major events. We can use a histogram to visually represent the distribution.

```
In [ ]: # plot the amount of fan fictions posted each year for each kudos category
sns.displot(
    data=fanfic_df,
    x="Date_updated",
    hue="Kudos_category",
    multiple="stack",
    bins=40,
    height=5,
    aspect=2,
    palette=p,
)

plt.xlabel("Date Updated")
plt.ylabel("Number of fan fictions")
plt.title("Number of fan fictions updated over time for each kudos category")
plt.show()
```

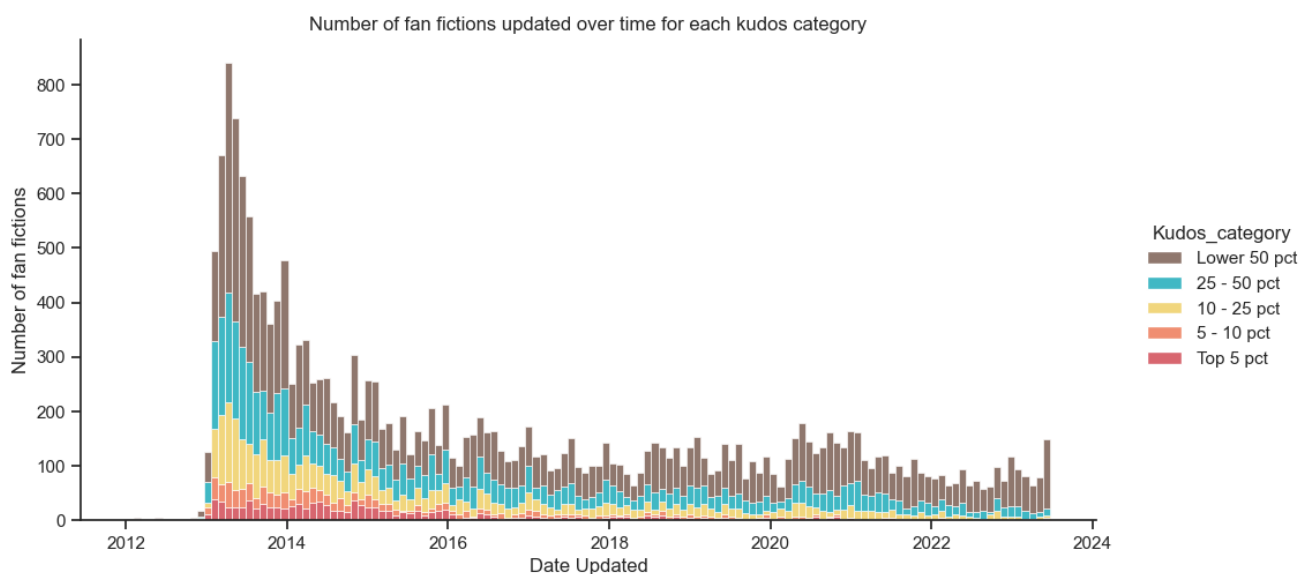


The graph shows that the AO3 Les Misérables fandom wasn't active till 2012. Now, let's zoom in to the period after 2012.

```
In [ ]: # remove data before 2012 because there are too few fan fictions before that time
        featured = fanfic_df[fanfic_df["Date_updated"] > "2012-01-01"]

        # plot the amount of fan fictions posted each year for each kudos category
        sns.displot(
            data=featured,
            x="Date_updated",
            hue="Kudos_category",
            multiple="stack",
            bins=132, # there are 132 months in our selection
            height=5,
            aspect=2,
            palette=p,
        )

        plt.xlabel("Date Updated")
        plt.ylabel("Number of fan fictions")
        plt.title("Number of fan fictions updated over time for each kudos category")
        plt.show()
```



The two graphs above show:

- The 2010 concert hardly triggered any fandom activities on the AO3 platform.

- Before the 2012 film, the AO3 Les Misérables fandom barely exists.
- The release of the 2012 Oscar-winning film was a groundbreaking moment for the AO3 Les Misérables fandom. Updates increased significantly after the film's release, peaked shortly after, and gradually declined over the next 11 years.
- The 2018 TV series had little effect on the fandom. Activity did not significantly increase post-series release.
- It was unexpected that the COVID pandemic also had a great impact on the fandom, even greater comparing to the 2018 TV series. It sparked a revival, leading to a small peak in mid-2020. As the influence of the pandemic diminishes, the fandom is returning to its normal state.
- Most of the top 10% most-liked fan fictions were published before 2016.

Possible reasons for this are:

- Before 2012, the Les Misérables fandom on AO3 was barely active. It quickly gained popularity in early 2013. This happened possibly because prior to that time, Les Misérables fans primarily engaged with other established platforms, such as LiveJournal (founded in 1999) or FanFiction.net (founded in 1998). They only started to migrate to AO3 since a certain moment. As fans migrated to AO3, they also imported their works to AO3 in batches, resulting in a surge of monthly updates.
- This could also be because most Les Misérables fans are fans of the 2012 film, or at least introduced to the fandom by the 2012 film. Many fans only started to write fan fictions after watching the film. As the popularity of the film gradually fades, the fandom is also declining.
- The fandom's popularity during the pandemic could be due to people being quarantined at home, having more time for online fan activities during the global lockdown.

4.7 The relationship between popularity and the timing of publication (cont.)

To better understand how the timing of publication influences a work, some more data processing is needed.

In the code below, I will group the data based on their year and month of update. Then, I'll calculate the average words, chapters, hits, kudos, bookmarks, and ratings for fan fictions updated in each month. This will provide us a clearer picture of the overall fandom trend.

```
In [ ]: # extract the month and year information from the "Date_updated" column
# and create "Year" and "Month" columns
filtered_fanfic["Year"] = filtered_fanfic["Date_updated"].dt.year
filtered_fanfic["Month"] = filtered_fanfic["Date_updated"].dt.month

# group the dataframe by "Year" and "Month" columns
# and calculate the average words, bookmarks, hits, etc. for each group
grouped_fanfic = filtered_fanfic.copy()
grouped_fanfic = (
    grouped_fanfic.groupby(["Year", "Month"]).mean(numeric_only=True).reset_index()
)

# convert the "Year" and "Month" columns back to "Date_updated" column
# and delete the "Year" and "Month" columns
grouped_fanfic["Date_updated"] = pd.to_datetime(
    grouped_fanfic["Year"].astype(str) + "-" + grouped_fanfic["Month"].astype(str)
)
grouped_fanfic = grouped_fanfic.drop(["Year", "Month"], axis=1)
```

For better visualization, let's remove the few outliers before June 2012.

```
In [ ]: # remove the data before June 2012
grouped_fanfic = grouped_fanfic[grouped_fanfic["Date_updated"] > "2012-06-01"]
```

Technically, the data is prepared at this point. However, for visualization purposes, two more steps are needed:

1. As the variables vary greatly in scales (for example, the average hits could be 10 times the average kudos and 100 times the average bookmarks) and our objective is to observe the trend only, we will normalize all the data to values ranging between 0 and 1.

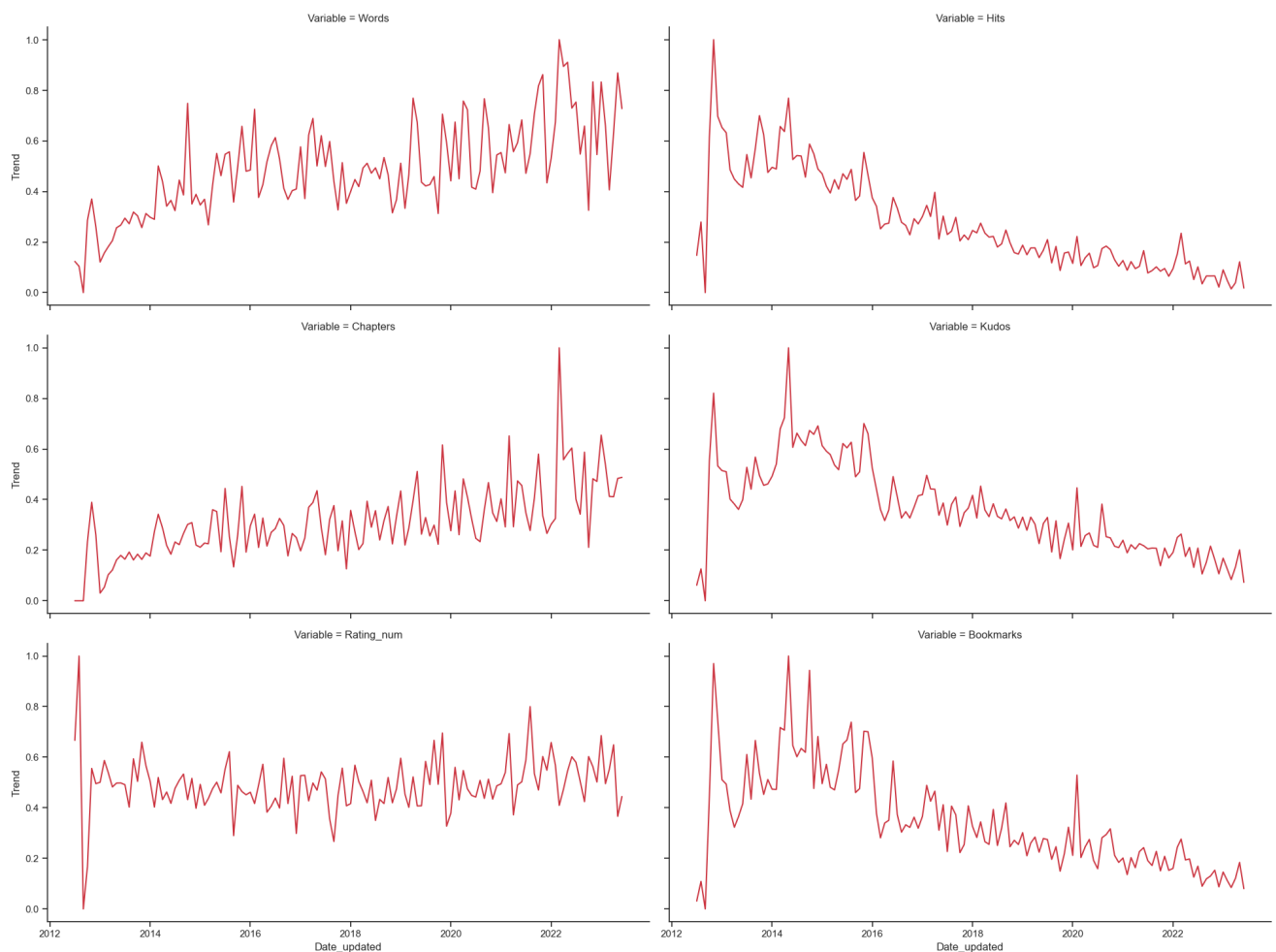
```
In [ ]: # normalize the data
for col in ["Words", "Chapters", "Hits", "Kudos", "Bookmarks", "Rating_num"]:
    grouped_fanfic[col] = (grouped_fanfic[col] - grouped_fanfic[col].min()) / (
        grouped_fanfic[col].max() - grouped_fanfic[col].min()
    )
```

2. Also, seaborn does not support plotting multiple columns in a single facet grid. We will need to unpivot the dataframe from a wide format to a long format by converting the column names into variables and the associated values into the corresponding values.

```
In [ ]: # unpivot the dataframe from wide to long format
grouped_fanfic = grouped_fanfic.melt(
    id_vars=["Date_updated"],
    value_vars=[
        "Words",
        "Hits",
        "Chapters",
        "Kudos",
        "Rating_num",
        "Bookmarks",
    ],
    var_name="Variable",
    value_name="Trend",
)
```

```
In [ ]: # plot the average words, hits, etc. each month using line plot facet grid
sns.relplot(
    kind="line",
    data=grouped_fanfic,
    x="Date_updated",
    y="Trend",
    col="Variable",
    col_wrap=2,
    height=5,
    aspect=2,
    color="#CC333F",
)

plt.show()
```



The grid above shows:

- The average words and average chapters per fan fiction have been increasing over the years.
- The average hits, average kudos, and average bookmarks per fan fiction, however, have been decreasing.
- The rating preference remains unchanged.
- Compared to the average hits, average kudos and average bookmarks have a certain lag. While hits peak in early 2013, kudos and bookmarks only peak in 2014.

One may draw from the observations above:

- It is a sad story that fans are writing longer stories but getting fewer likes over time.
- A good story can wait. The most-liked stories are updated more than a year after the peak of fandom popularity. But one shouldn't make the audience wait too long and risk losing their interest completely.

4.8 The relationship between popularity and story settings

Hypothesis 5: If a fan fiction contains certain elements or depicts the main characters in a certain way, it is more likely to be popular.

To figure out the most popular fandom settings, we can build a word cloud based on the freeform tags corpus.

It involves three steps. First, create a featured corpus. Second, process the words. Finally, make a word cloud using the most common words in the corpus.

Because *Les Misérables* is a French novel, it includes many proper nouns with accented letters. Some fan fiction writers choose to omit these accents but some do not. Let us remove the letter accents first for consistency.

```
In [ ]: # remove letter accents from freeform tags
fanfic_df["Freeform_tags"] = [unicode(x) for x in fanfic_df["Freeform_tags"]]
```

Word processing involves the following steps:

- tokenize the text into words
- flatten the list of lists into a single list of words
- remove non-alphanumeric characters
- remove stop words. Except for the standard English stop words, I also removed words with length less than 2--they are less expressive--and some common unmeaningful words in the fandom. "Centric", for example, is always used in "xxx centric" tags. It just means the fiction is centered around xxx. It does not contain useful information for our analysis.
- lemmatize the words

I use regular expression and natural language toolkit (nltk) library to perform this task.

```
In [ ]: # the helper function for word processing
def corpus_processing(corpus):
    # tokenize
    corpus = corpus.apply(lambda x: x.split())

    # convert to a list of strings
    corpus = [word for tag in corpus for word in tag]

    # remove non-alphanumeric characters
    corpus = [re.sub(r"[^A-Za-z0-9]+", "", word) for word in corpus]

    # remove stop words and words with length less than 2
    stop_words = set(stopwords.words("english"))
    costume_stop_words = [
        "les",
        "miserables",
        "enjolas",
        "grantaire",
        "alternative",
        "universe",
        "alternate",
        "setting",
        "centric",
        "character",
    ]
    stop_words.update(costume_stop_words)
    corpus = [word for word in corpus if not word in stop_words and len(word) > 2]

    # lemmatize
    lemmatizer = WordNetLemmatizer()
    corpus = [lemmatizer.lemmatize(word) for word in corpus]

    return corpus
```

To create a word cloud, we need to first count the frequency of each word in the corpus. Then, we can choose the top X most common words and use the word frequency as the weight of the word in the word cloud.

- ## 2. Freeform tags with "Enjolras" corpus

```
In [ ]: # corpus 2: freeform tags with "enjolras"
corpus_enjolras = corpus_all[corpus_all.str.contains("enjolras")]
print("There are", corpus_enjolras.size, "tags with 'Enjolras'.")
create_word_cloud(corpus_processing(corpus_enjolras), 100)
```

This allows us to gain insight into how fans perceive not only a fictional character from the 19th century, but also our present-day society. Through writing fan fiction, fans draw parallels between the modern LGBT movement and great revolutions in human history.

```
In [ ]: # corpus 3: freeform tags with "grantaire"
corpus_grantaire = corpus_all[corpus_all.str.contains("grantaire")]
print("There are", corpus_grantaire.size, "tags with 'Grantaire'.")
create_word_cloud(corpus_processing(corpus_grantaire), 100)
```

There are 4136 tags with 'Grantaire'.

Archive of Our Own. (2023, June 27). Search Works |. Retrieved June 27, 2023, from https://archiveofourown.org/works/search?work_search%5Bquery%5D=

Brownlee. (2018, April 25). How to Remove Outliers for Machine Learning. Retrieved June 24, 2023, from <https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/>

Döring, N. (2020). Erotic Fan Fiction. Encyclopedia of Sexuality and Gender, 1–8. https://doi.org/10.1007/978-3-319-59531-3_65-1

Eskici. (2023, May 13). AI and Data Scraping on the Archive. Retrieved June 22, 2023, from <https://www.transformativeworks.org/ai-and-data-scraping-on-the-archive/>

Fanlore. (2022, August 5). Fanwork. Retrieved June 25, 2023, from <https://fanlore.org/wiki/Fanwork>

Hellekson, K., & Busse, K. (2014). The Fan Fiction Studies Reader. University of Iowa Press.

Johnson, S. F. (2014). Fan fiction metadata creation and utilization within fan fiction archives: Three primary models. Transformative Works and Cultures, 17. <https://doi.org/10.3983/twc.2014.0578>

Les Misérables (2012) - IMDb. (2012, December 25). Retrieved from <https://www.imdb.com/title/tt1707386/>

Les Misérables (TV Mini Series 2018–2019) - IMDb. (2019, April 14). Retrieved from <https://www.imdb.com/title/tt5900600>

Les Misérables in Concert: The 25th Anniversary (2010) - IMDb. (2010, October 3). Retrieved from <https://www.imdb.com/title/tt1754109/>

M. (2021, October 29). Remove special characters but not accented letters. Retrieved June 22, 2023, from <https://stackoverflow.com/a/56280214>

Mishra, A. (2021, October 22). How to categorize a column by applying a function in pandas? Retrieved June 24, 2023, from <https://aparnamishra144.medium.com/how-to-categorize-a-column-by-applying-a-function-in-pandas-135f47f7ab34>

Organization for Transformative Works. (2009, November 13). Announcing Open Beta! Retrieved June 27, 2023, from <https://www.transformativeworks.org/announcing-open-beta/>

Oxford English Dictionary. (2004, December). fan, n.2 . Retrieved June 25, 2023, from <https://www.oed.com/viewdictionaryentry/Entry/68000>

sklearn.preprocessing.OrdinalEncoder. (n.d.). Retrieved June 23, 2023, from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html>

Yin, K., Aragon, C.R., Evans, S., & Davis, K. (2017). Where No One Has Gone Before: A Meta-Dataset of the World's Largest Fanfiction Repository. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.