TORONTO SCHOOL OF MANAGEMENT

Final Project
Working with Data Using Python

- Akwasi Afriyie
- Esteban Omar Rosero Romero
- Mateo Hidalgo Prada
- Yarazeth Garcia Ibarra
- Koichiro Suzuki

Introduction

The aim of this project is to undertake sentiment analysis using Python on a dataset of 142.8 million reviews for 212404 unique books on Amazon between 1996 and 2014.
It involves analyzing two data files, one for book reviews and one for book details, with various features such as book title, price, review rating, and full review text.
Task one requires a brief explanation of text mining, text classification, text clustering, and sentiment analysis.
The second task involves data cleansing, classification, and word clouds to generate analytical visualizations. It includes classifying reviews into positive and negative sentiment categories and generating corresponding word clouds.
The third task involves building both a simple and multinomial logistic regression models to predict the sentiment category and rating of a book based on its text-based review.

BRIEF EXPLANATION OF TERMS

Text Mining:

Text Mining is the extraction of useful and meaningful information from large amounts of unstructured or semi-structured text data. It involves techniques such as natural language processing, machine learning, and data mining to identify patterns, extract key terms, and categorize text into different topics or themes. The goal of text mining is to turn raw text into valuable insights that can inform business decisions, scientific research, and other areas of study.

Text Classification:

Text Classification is the process of categorizing text data into predefined categories or classes based on their content. It is a supervised machine learning task where the system is trained on a set of labeled text examples and uses this training data to classify new text data into the appropriate categories. Some common applications of text classification include sentiment analysis, spam detection, and news categorization.

BRIEF EXPLANATION OF TERMS

Text Clustering:

Text Clustering is an unsupervised machine learning task where the system groups similar text documents into clusters without the use of predefined categories. The goal of text clustering is to identify underlying structures or themes in large collections of text data. Clustering algorithms can be used to identify topics, patterns, and relationships in text data, which can provide valuable insights and help to better understand the content.

Sentiment Analysis:

Sentiment Analysis is the process of determining the sentiment or emotion expressed in a text document, such as positive, negative, or neutral. It is a subfield of text classification and involves the use of natural language processing and machine learning techniques to analyze text data and determine the underlying sentiment. Sentiment Analysis is commonly used in customer feedback analysis, marketing research, and political analysis to understand public opinions and sentiment towards specific products, brands, or events.

Data Cleansing, Classification, and Word Clouds.

Variable analysis

---Books data---

#Get a concise summary of the dataframe.

```
df_data.info()
```

```
Data columns (total 10 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Title          212403 non-null  object
 1   description    143962 non-null  object
 2   authors        180991 non-null  object
 3   image          160329 non-null  object
 4   previewLink    188568 non-null  object
 5   publisher      136518 non-null  object
 6   publishedDate  187099 non-null  object
 7   infoLink       188568 non-null  object
 8   categories     171205 non-null  object
 9   ratingsCount   49752 non-null   float64
dtypes: float64(1), object(9)
memory usage: 16.2+ MB
```

#Check for the number of null values for each variable.

```
df_data.isnull().sum()
```

```
Title             1
description       68442
authors           31413
image             52075
previewLink       23836
publisher         75886
publishedDate     25305
infoLink          23836
categories        41199
ratingsCount      162652
dtype: int64
```

Variable analysis

--Books Ratings---

#Get a concise summary of the dataframe.

```
df_rating.info()
```

```
Data columns (total 10 columns):
 #   Column             Dtype
---  ------             -----
 0   Id                 object
 1   Title              object
 2   Price              float64
 3   User_id            object
 4   profileName        object
 5   review/helpfulness object
 6   review/score       float64
 7   review/time        int64
 8   review/summary     object
 9   review/text        object
dtypes: float64(2), int64(1), object(7)
```

#Check for the number of null values for each variable.

```
df_rating.isnull().sum()
```

```
Id                      0
Title                 208
Price             2518829
User_id            561787
profileName        561886
review/helpfulness      0
review/score            0
review/time             0
review/summary         38
review/text             8
```

Merging Data

#Delete variables that are irrelevant to the creation of the predictive model

```
df_data.drop(columns=['image', 'previewLink','infoLink','ratingsCount'], inplace=True)

df_rating.drop(columns=['Id', 'User_id','review/helpfulness','profileName'], inplace=True)
```

#Do Margin-left Books data to Rating data via 'Title' column.

```
df=pd.merge(df_rating, df_data, on='Title', how='left')
```

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories
0	Its Only Art If Its Well Hung!	NaN	4.0	940636800	Nice collection of Julie Strain images	This is only for Julie Strain fans. It's a col...	NaN	['Julie Strain']	NaN	1996	['Comics & Graphic Novels']
1	Dr. Seuss: American Icon	NaN	5.0	1095724800	Really Enjoyed It	I don't care much for Dr. Seuss but after read...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']
2	Dr. Seuss: American Icon	NaN	5.0	1078790400	Essential for every personal and Public Library	If people become the books they read and if "t...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']
3	Dr. Seuss: American Icon	NaN	4.0	1090713600	Phlip Nel gives silly Seuss a serious treatment	Theodore Seuss Geisel (1904-1991), aka "D...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']
4	Dr. Seuss: American Icon	NaN	4.0	1107993600	Good academic overview	Philip Nel - Dr. Seuss: American IconThis is b...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']

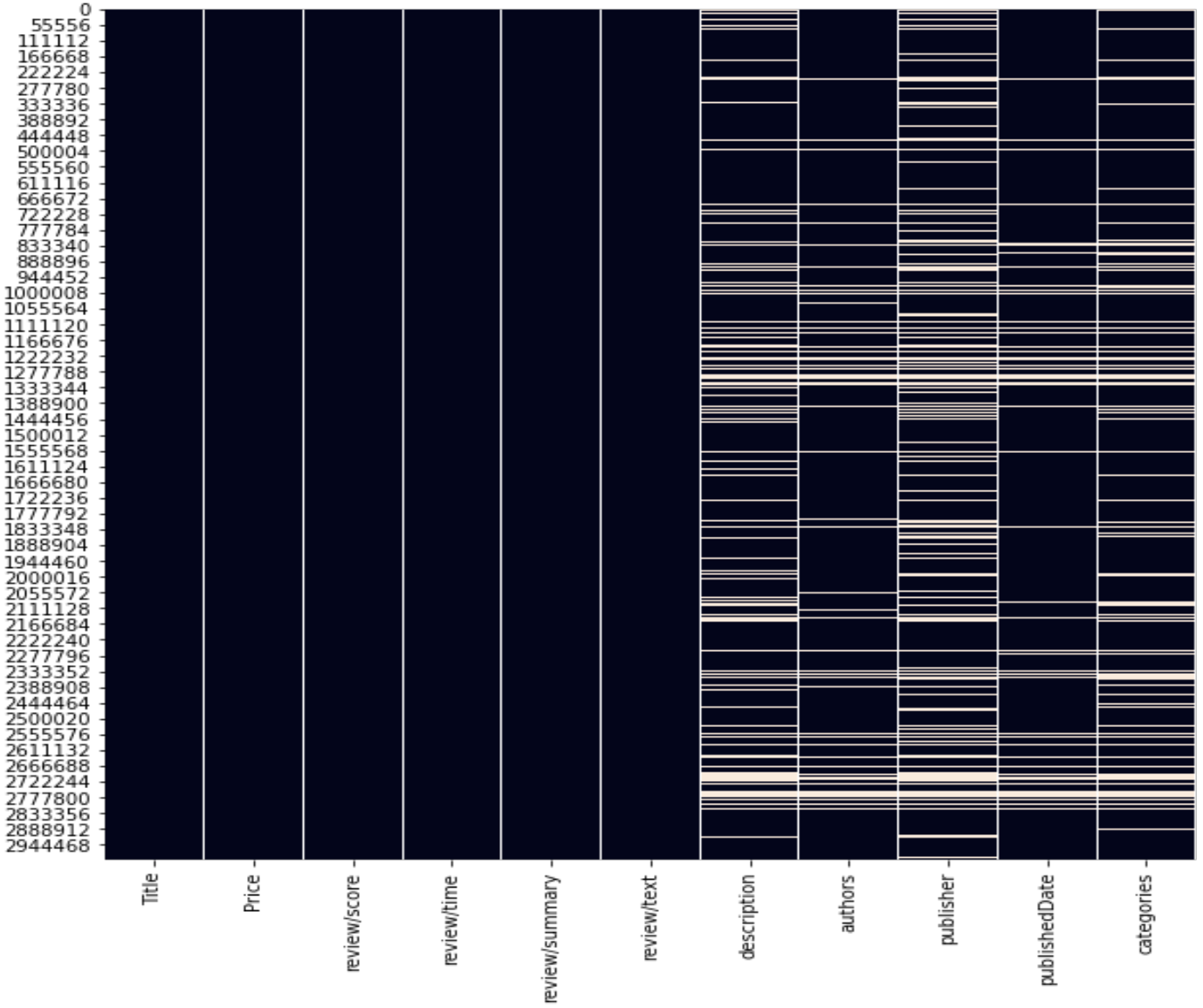
Treating missing values

 # There were 2518829 missing values for 'Price' in 'Book ratings'. Those missing values were replaced with the average value of 'Price'.

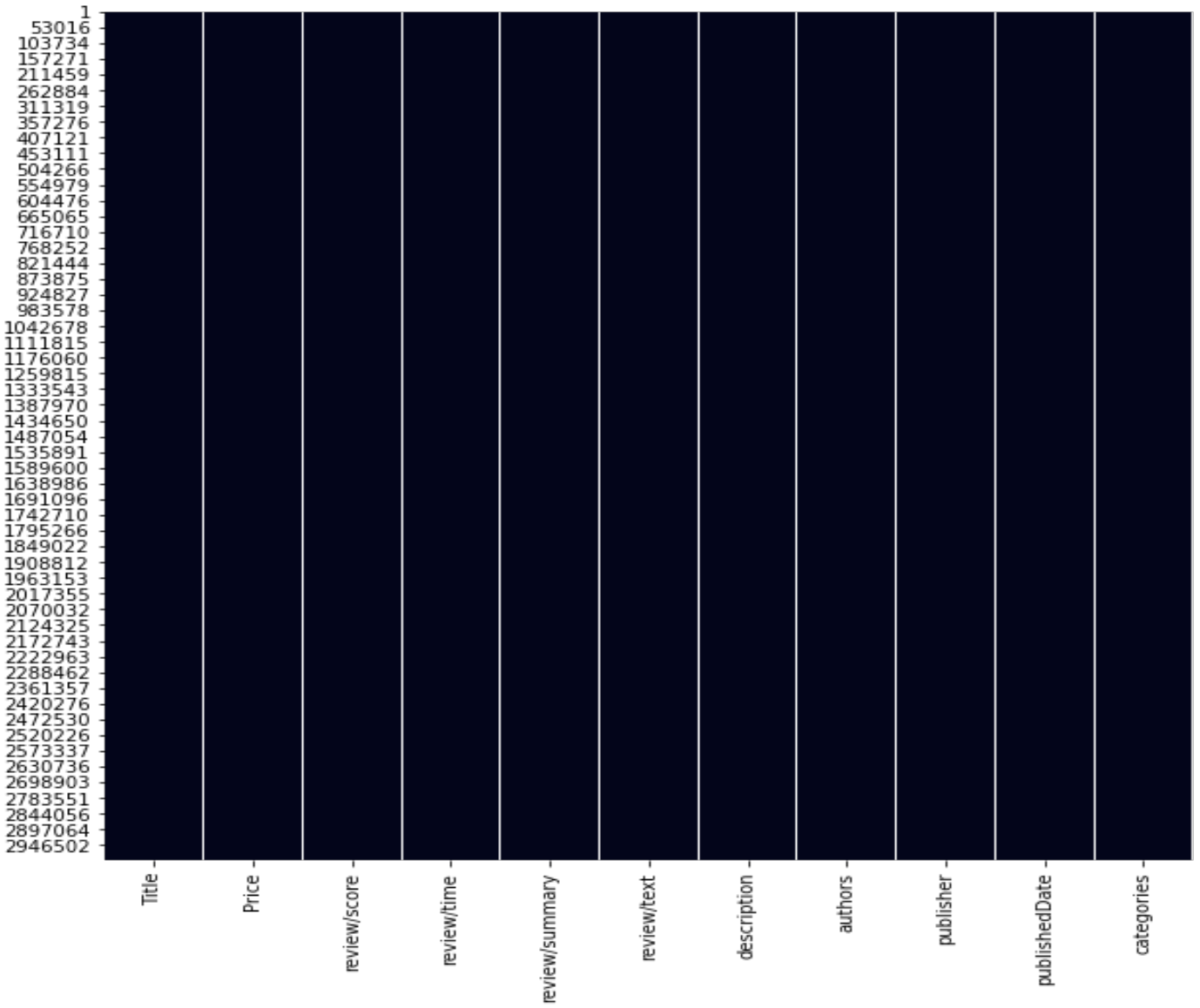
```
df['Price']=df['Price'].fillna(df['Price'].mean())
```

```
Title              208
Price                0
review/score         0
review/time          0
review/summary      38
review/text          8
description     640225
authors         390634
publisher       782617
publishedDate   354581
categories      551498
dtype: int64
```

#No more missing values for 'Price'



The following graph shows the distribution of missing values in the data.



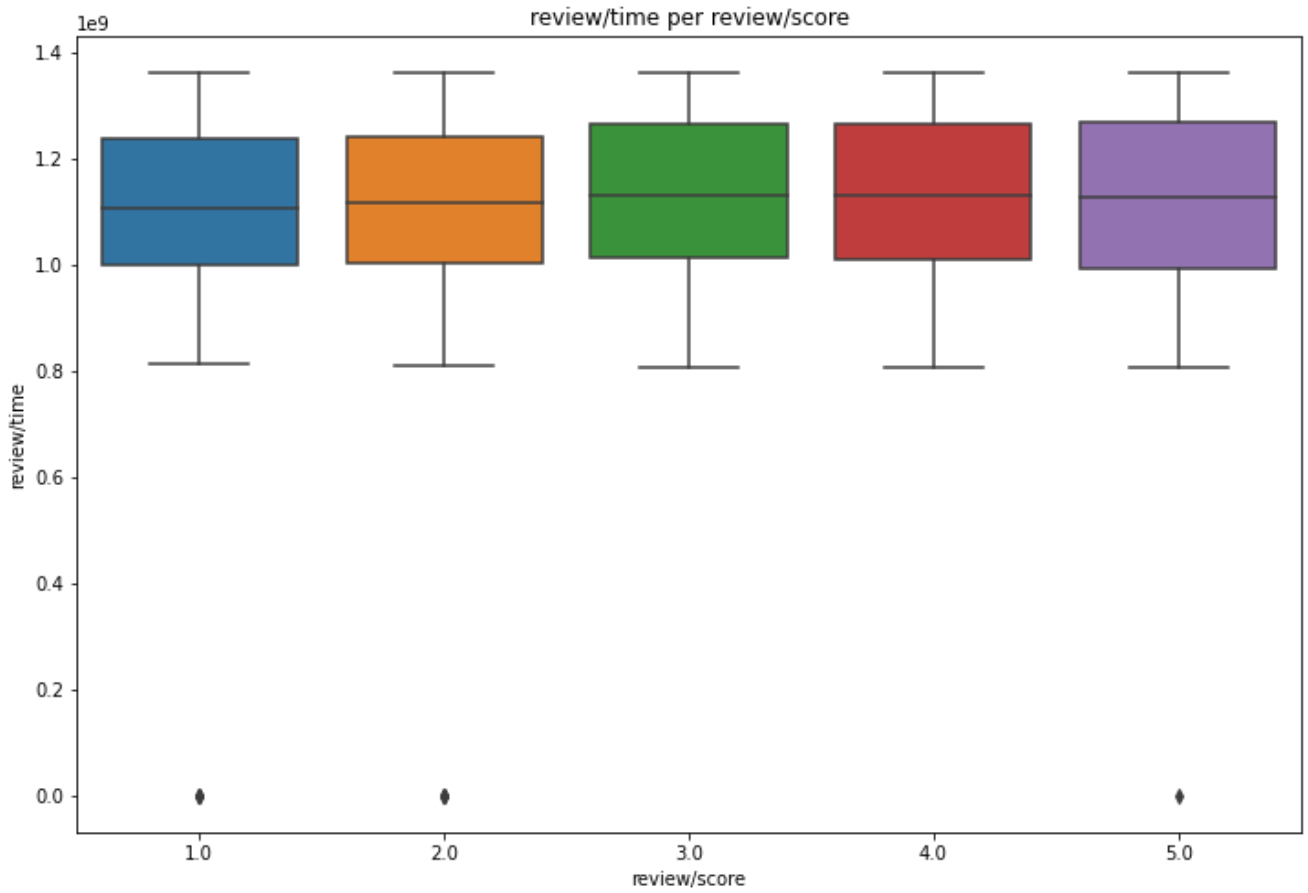
All the missing values were removed

df=df.dropna()

#As a result, 2071176 data remained.

Treating outliers

This graph shows the 'review/time per review/score'. And the existence of outliers in 'review/time'



Treating outliers

#After checking the outliers in 'review/time' there were "-1" values.

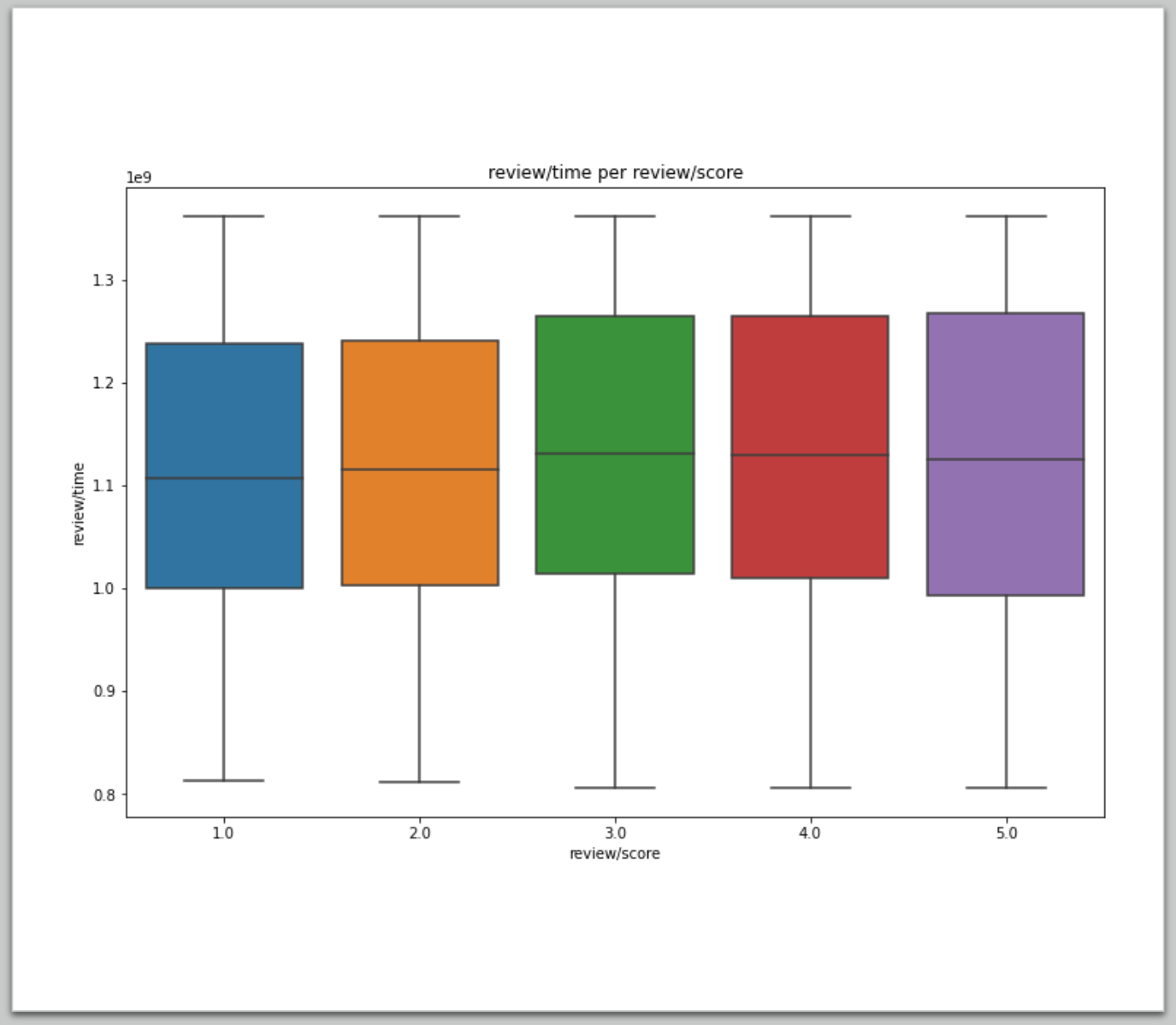
	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories
75745	Julie and Julia: 365 Days, 524 Recipes, 1 Tiny...	21.762656	2.0	-1	For once, the movie was better.	I purchased this book after seeing, and truly ...	Pushing thirty, living in a rundown apartment ...	['Julie Powell']	Viking	2005	['Cookery, French']
75746	Julie and Julia: 365 Days, 524 Recipes, 1 Tiny...	21.762656	1.0	-1	Dazed and Confused	I had such high hopes for this book and I was ...	Pushing thirty, living in a rundown apartment ...	['Julie Powell']	Viking	2005	['Cookery, French']
75747	Julie and Julia: 365 Days, 524 Recipes, 1 Tiny...	21.762656	2.0	-1	Disappointing...read My Life in France by Juli...	I eagerly snatched this book up when I saw it ...	Pushing thirty, living in a rundown apartment ...	['Julie Powell']	Viking	2005	['Cookery, French']
75748	Julie and Julia: 365 Days, 524 Recipes, 1 Tiny...	21.762656	2.0	-1	Narcissistic Hipsters Should Cook, Not Write	After seeing the movie, I knew what I was in f...	Pushing thirty, living in a rundown apartment ...	['Julie Powell']	Viking	2005	['Cookery, French']

#Those values were removed

```
df=df[df['review/time'] != -1]
```

Treating outliers

This graph shows that there´s no more outliers in 'review/time'.



Treating outliers

In the variable *"publishedDate", there were multiple abnormal values*

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories	publishedDate_year
200828	The Iliad: The Story of Achilles	21.762656	4.0	1208131200	One of our first war novels	One of our first war novels: the Achains and t...	This new, modern translation of The Iliad is f...	['Homer']	Signet Book	199?	['Epic poetry, Greek']	199?
2919171	Circular breathing for the wind performer	21.762656	4.0	1306108800	/you can find this information online	I was disappointed when I received this book. ...	Author Kynaston discusses the technique of cir...	['Trent Kynaston']	Alfred Music	199?	['Music']	199?

"199?"

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories	publishedDate_year
780252	One wonderful night: A Romance of New York,	21.762656	1.0	1349654400	Not worth reading	Did not care for it. Very old fashioned and to...	Lady Hermione flees to New York to escape marr...	['Louis Tracy']	BEYOND BOOKS HUB	101-01-01	['Fiction']	101-
780253	One wonderful night: A Romance of New	21.762656	3.0	1351468800	One Wonderful Night A Romance of New York	This book was hard to follow at times. It was ...	Lady Hermione flees to New York to escape	['Louis Tracy']	BEYOND BOOKS HUB	101-01-01	['Fiction']	101-

"101-"

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories	publishedDate_year
1384321	A Wealth of Wisdom: Legendary African American...	11.18	5.0	1074384000	Wealth of Knowledge - History's Truths	Thanks to Dr. Cosby and Renee Pouissant we now...	The wisdom of our elders is our most valuable ...	['Camille Cosby', 'Rene Poussaint']	Simon and Schuster	2030-12-31	['Biography & Autobiography']	2030
1384322	A Wealth of Wisdom: Legendary African American...	11.18	5.0	1076544000	Couldn't Put It Down	This is one of those rare books I simply could...	The wisdom of our elders is our most valuable ...	['Camille Cosby', 'Rene Poussaint']	Simon and Schuster	2030-12-31	['Biography & Autobiography']	2030
1384323	A Wealth of Wisdom: Legendary African American...	11.18	4.0	1075939200	Enjoyed It!!	I enjoy reading this book. Really. The reason ...	The wisdom of our elders is our most valuable ...	['Camille Cosby', 'Rene Poussaint']	Simon and Schuster	2030-12-31	['Biography & Autobiography']	2030

"2030"

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories	publishedDate_year
296184	Within a budding grove,	21.762656	3.0	1265414400	Say Hello to the Princess of Luxembourg	Proust is an odd one -- an essayist and memoir...	"A l'ombre des jeunes fi lles en fl eurs" est ...	['Proust M.']	Рипол Классик	19??	['Fiction']	19??
296185	Within a budding grove,	21.762656	5.0	978566400	Stunning!	Without a doubt, this is the best piece of lit...	"A l'ombre des jeunes fi lles en fl eurs" est ...	['Proust M.']	Рипол Классик	19??	['Fiction']	19??
296186	Within a budding grove,	21.762656	5.0	1330128000	Seascape with a frieze of boys...	Reading Proust is like drinking good tea. You ...	"A l'ombre des jeunes fi lles en fl eurs" est ...	['Proust M.']	Рипол Классик	19??	['Fiction']	19??

"19??"

Treating outliers

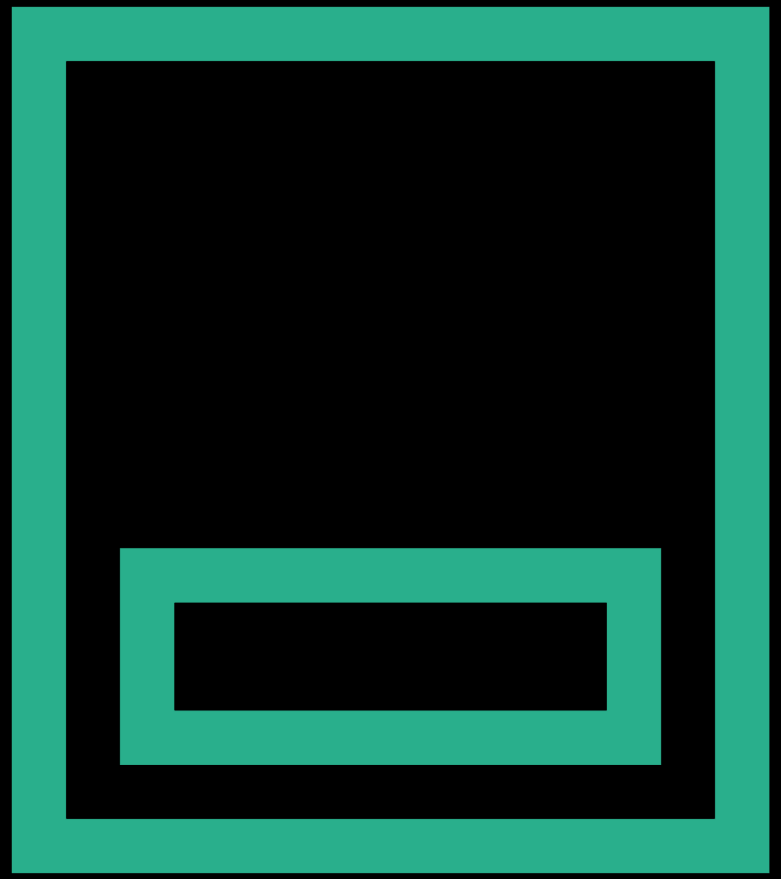
1. *A new column named 'publishedDate_year' was created with only the year taken from 'publishedDate'.*

```
df['publishedDate_year']=df['publishedDate'].str[:4]
```

2. Irregular values *"19??", "199?","101-","2030"* were removed

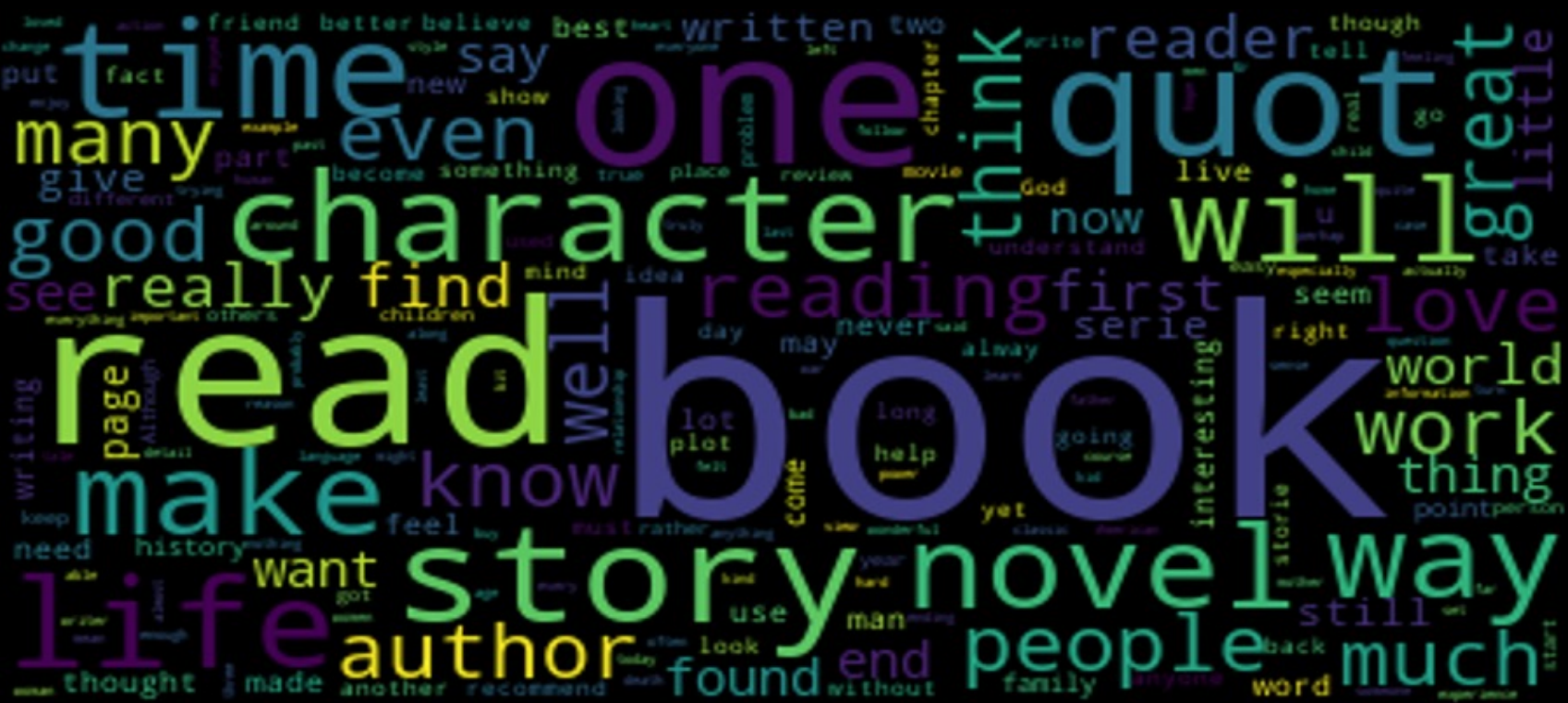
```
df=df[df['publishedDate_year'] != '101-']

df=df[df['publishedDate_year'] != '2030']

df=df[df['publishedDate_year'] != '199?']

df=df[df['publishedDate_year'] != '19??']
```

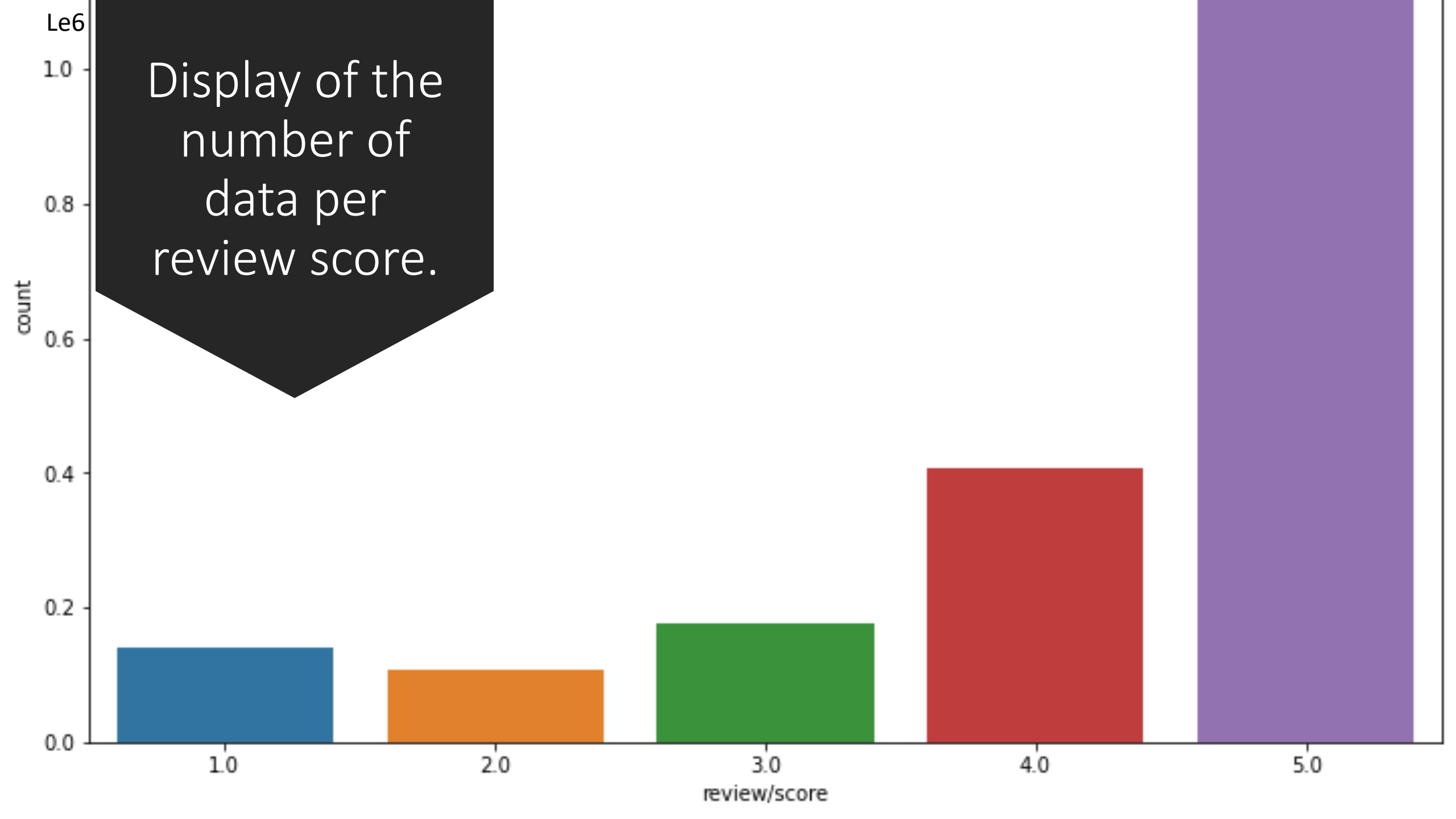
Generate initial analytical visualizations to understand the reviews



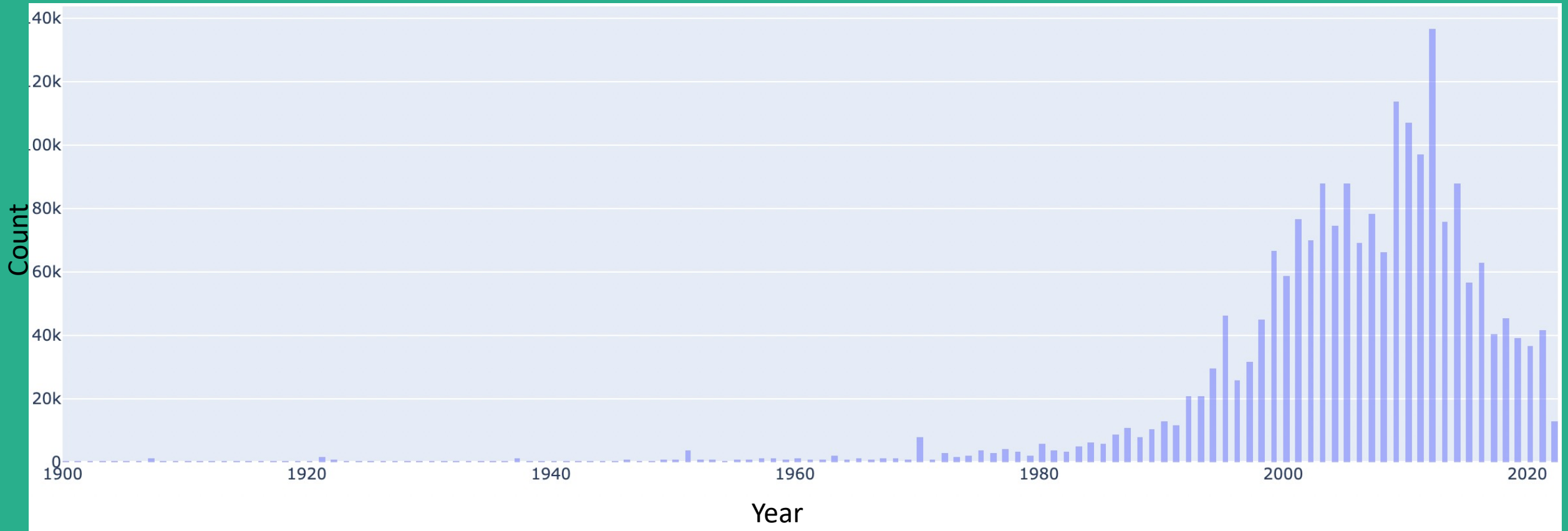
Visualize the words that are often included in 'review/text' using a word cloud.

- *#(10000 samples)* df_rating_sample10000=df.sample(n=10000)
- *#Create stopword list* stops= set(STOPWORDS)





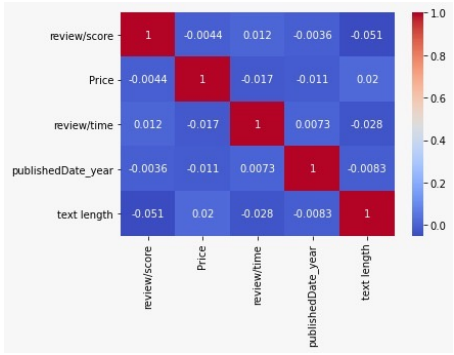
Display of the number of data per review score.



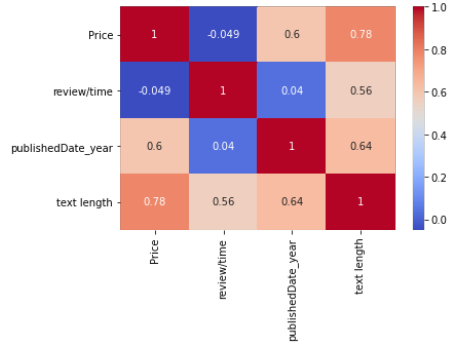
#Number of data per publication year.

Most of the publication were between 2000 and 2020

Correlation analysis between 'review/score' and 'price', 'review/time' and 'text length'.



review/score	Price	review/time	publishedDate_year	text length
1.0	21.653179	1.118128e+09	2004.820585	793.900548
2.0	21.711868	1.121717e+09	2005.093863	928.639748
3.0	21.673666	1.136735e+09	2005.073905	986.441841
4.0	21.678704	1.135118e+09	2004.789454	946.258080
5.0	21.556469	1.128884e+09	2004.798769	756.711431



Classifying reviews

In this study, a 'review/score' score higher than 3 was defined as a positive review, and a score of less or equal than 3 was defined as a negative review.

df['sentiment'] = ['positive' if x > 3 else 'negative' for x in df['review/score']]

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories	publishedDate_year	text length	sentiment
1	Dr. Seuss: American Icon	21.762656	5.0	1095724800	Really Enjoyed It	I don't care much for Dr. Seuss but after read...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']	2005	1423	positive
2	Dr. Seuss: American Icon	21.762656	5.0	1078790400	Essential for every personal and Public Library	If people become the books they read and if "t...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']	2005	1752	positive
3	Dr. Seuss: American Icon	21.762656	4.0	1090713600	Phlip Nel gives silly Seuss a serious treatment	Theodore Seuss Geisel (1904-1991), aka "D...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']	2005	3662	positive

Positive and negative word clouds

1. Since there are about 2 million data, we randomly selected 10,000 each of 'Positive' and 'Negative'.

```
df_Psample10000=df[df['sentiment']=='positive'].sample(n=10000)
df_Nsample10000=df[df['sentiment']=='negative'].sample(n=10000)
```

2. We created a stopword list to remove the words that don't add meaning to a review

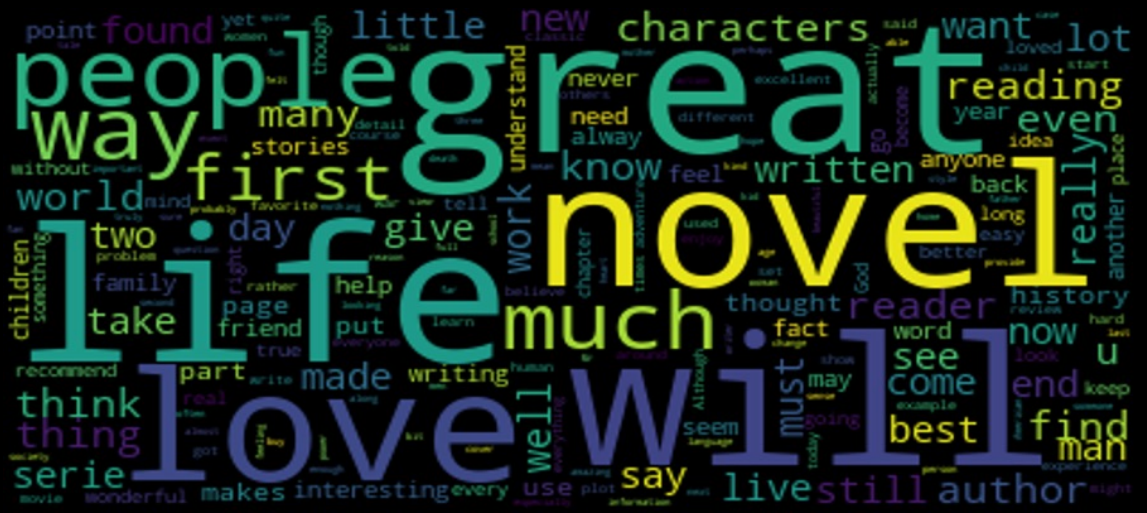
```
stops= set(STOPWORDS)
stops.update(["book", "read","one","story","time","quot","character","books","make"])
```

3. Creating the Word Clouds

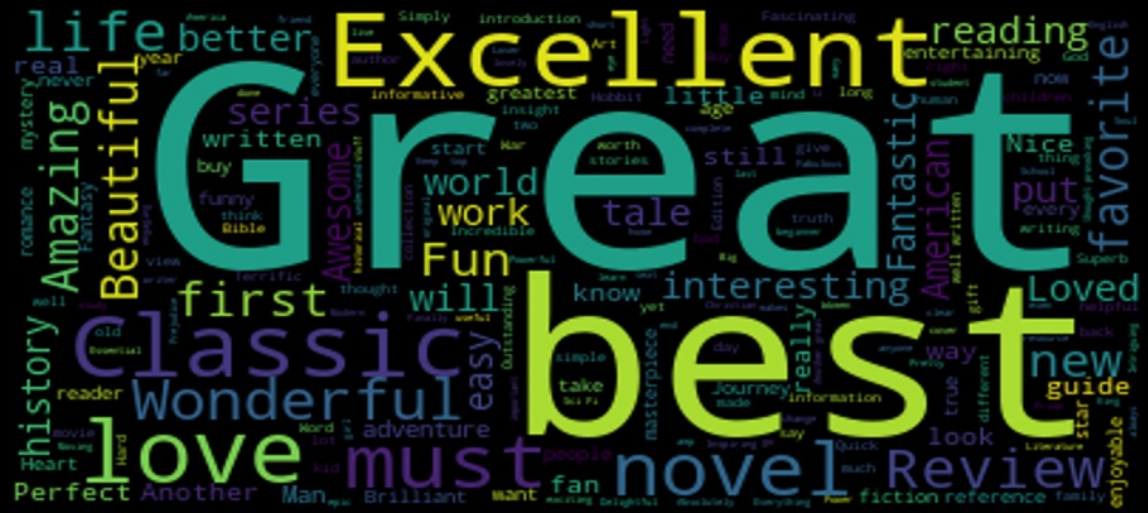
```
textt_p = " ".join(x for x in df_Psample10000["review/text"])
wordcloud = WordCloud(stopwords=stops).generate(textt_p)
wordcloud.to_file("review_p.png")
```

Positive Word Cloud

Based on the text of the review

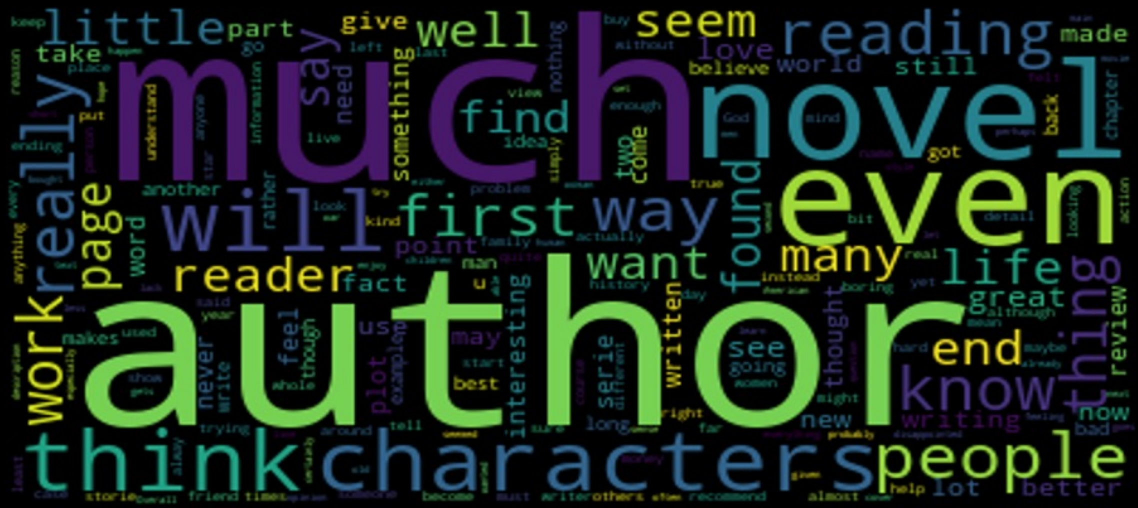


Based on the summary of the review



Negative Word Cloud

Based on the text of the review



Based on the summary of the review

Prediction

Prediction

- 3.1 Build a simple logistic regression model to predict the sentiment category based on a text-based review.

```
#Execution of required modules
import string
import re as re
import random as ran
import plotly
import matplotlib
import pandas as pd
import nltk as nltk
from nltk.corpus import stopwords
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import CountVectorizer
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

Prediction

```
df.head()
```

	Title	Price	review/score	review/time	review/summary	review/text	description	authors	publisher	publishedDate	categories	publishedDate_year	text length	sentiment
1	Dr. Seuss: American Icon	21.762656	5.0	1095724800	Really Enjoyed It	I don't care much for Dr. Seuss but after read...	Philip Nel takes a fascinating look into the k...	['Philip Nel']	A&C Black	2005-01-01	['Biography & Autobiography']	2005	1423	positive

```
df_sub = df[['review/score','review/text',  'sentiment']]
df_sub.head()
```

	review/score	review/text	sentiment
1	5.0	I don't care much for Dr. Seuss but after read...	positive

```
def remove_punc_stopwords(text):
    text = re.sub(f"[{string.punctuation}]"," ",text)
    text_tokens = set(text.split())
     text_tokens = text_tokens.difference(stops)
    return " ".join(text_tokens)
```

#We randomly extracted 100000 pieces of data

df_sub_sample100000=df_sub.sample(n=100000)

#Remove punctuations and stopwords from the text data

df_sub_sample100000['review/text'] = df_sub_sample100000['review/text'].apply(remove_punc_stopwords)

Prediction

```
df_sub_sample100000.head()
```

	review/score	review/text	sentiment
1578389	5.0	wellbeing figure even noggin wisdoms heartedly...	positive

 # split the dataset into two: train (85% of the obs.) and test (15% of the obs.)

```
df_sub_sample100000['random_index'] =[ ran.uniform(0,1) for x in range(len(df_sub_sample100000))]
df_sub_train=df_sub_sample100000[df_sub_sample100000.random_index < 0.85]
df_sub_test=df_sub_sample100000[df_sub_sample100000.random_index >= 0.85]
```

```
         review/score                                        review/text  \
1578389           5.0  wellbeing figure even noggin wisdoms heartedly...
1840683           3.0  suggest gone past seemed blood I might descrip...
1441591           2.0  stands 1 18 humility doesn Taoist truths perso...
2564758           3.0  practically pretty paragraph Robards total AND...
956313            1.0  days Einstein evolutionists confess makes appl...

         sentiment  random_index
1578389  positive      0.379231
1840683  negative      0.668277
1441591  negative      0.056891
2564758  negative      0.279974
956313   negative      0.659383
```

```
         review/score                                        review/text  \
758886            4.0  give So far draw don explain high written pret...
37218             5.0  days disclosure far particular written third o...
259778            5.0  complicated involvement written Though might f...
1267691           5.0  notes s price 3 comments whatnot Nietzche edit...
1684936           4.0  promises order favortie rewards power somethin...

         sentiment  random_index
758886   positive      0.974274
37218    positive      0.896959
259778   positive      0.941015
1267691  positive      0.983960
1684936  positive      0.934039
```

Prediction

#We used count vectorizer to transform the text into a 'bag of words' model, which will contain a sparse matrix of integers since the logistic regression algorithm cannot understand text

```
vectorizer = CountVectorizer(token_pattern=r'\b\w+\b')
train_matrix = vectorizer.fit_transform(df_sub_train['review/text'])
test_matrix = vectorizer.transform(df_sub_test['review/text'])
```

#Perform Logistic Regression

```
lr = LogisticRegression(solver='liblinear')
X_train = train_matrix
X_test = test_matrix
y_train = df_sub_train['sentiment']
y_test = df_sub_test['sentiment']
lr.fit(X_train,y_train)
```

#Generate the predictions for the test dataset

```
predictions = lr.predict(X_test)
df_sub_test['predictions'] = predictions
print(df_sub_test.head(30))
```

	review/score	review/text	sentiment	random_index	predictions
758886	4.0	give So far draw don explain high written pret...	positive	0.974274	positive
37218	5.0	days disclosure far particular written third o...	positive	0.896959	positive
259778	5.0	complicated involvement written Though might f...	positive	0.941015	positive

Prediction

#Calculate the prediction accuracy
df_sub_test['match'] = df_sub_test['sentiment'] == df_sub_test['predictions']
print(sum(df_sub_test['match'])/len(df_sub_test))

#We created a good model with about 87% accuracy
0.8685206079511515

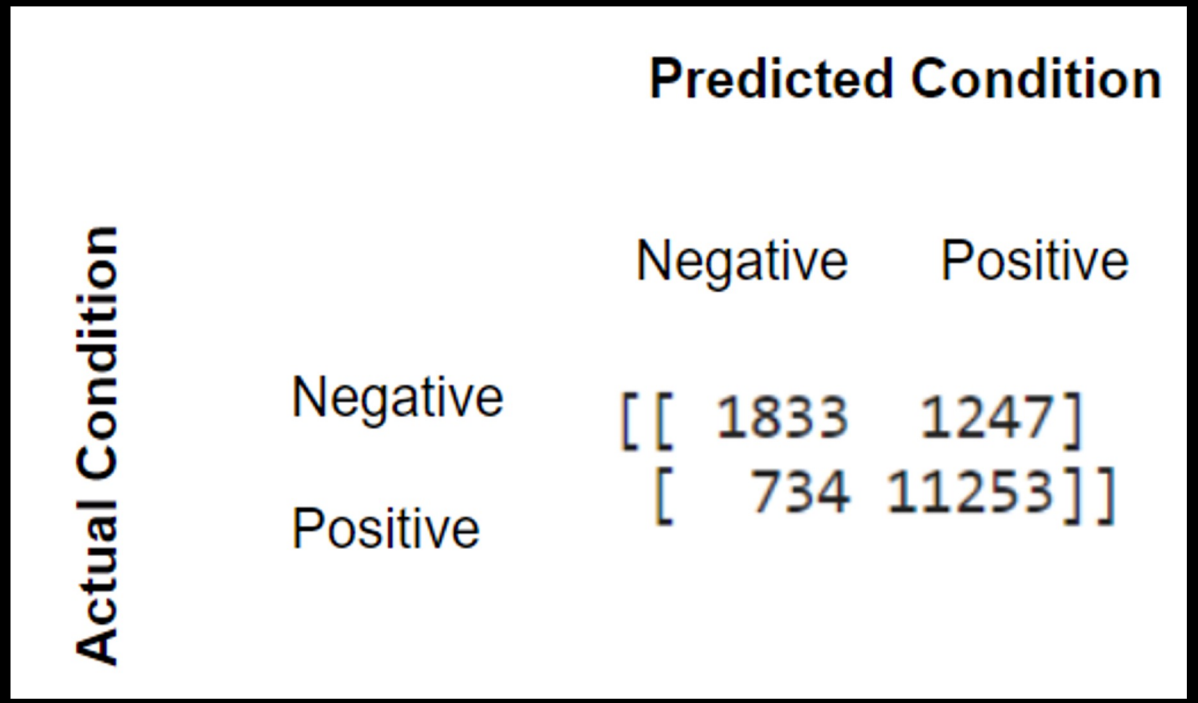
#We can use SciKit Learn's built-in confusion matrix to give us a better idea of what our classification model is getting right and what types of errors it is making.

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,predictions))
```

1833 Negative numbers were correctly predicted as Negative.
1247 Negative numbers were incorrectly predicted as Positive.

734 Positive numbers were incorrectly predicted as Negative.
11253 Positive numbers were correctly predicted as Positive.

1833 + 11253 / (1833 + 1247 + 734 + 11253) =
0.8685206079511515(accuracy)



Prediction

Building a multinomial logistic regression model to predict the rating of a book.

- **Creating a logistic model**

~Predicted Target ~
‘review/score’ [1,2,3,4,5]

```
#Perform Logistic Regression
lr2 = LogisticRegression(solver='liblinear')

X_train = train_matrix
X_test = test_matrix
y_train2 = df_sub_train['review/score']
y_test2 = df_sub_test['review/score']


lr2.fit(X_train,y_train2)
```

```
#Generate the predictions for the test dataset
predictions2 = lr2.predict(X_test)
```

Prediction

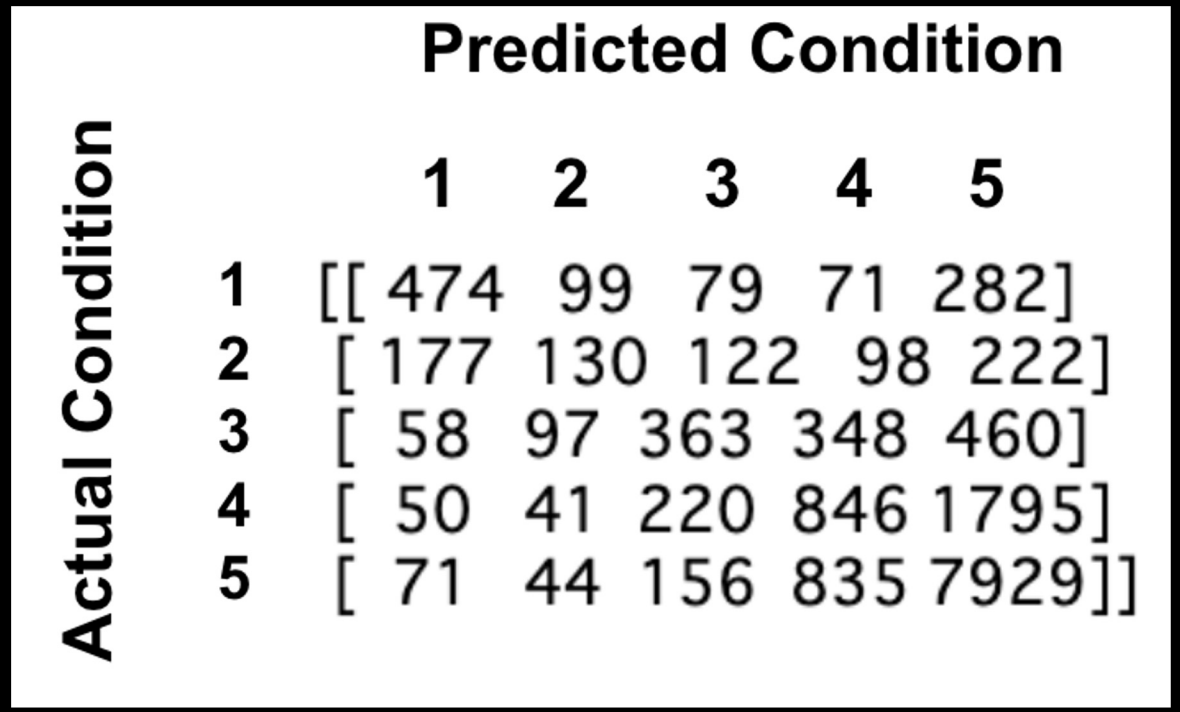
#The accuracy of logistic model

As expected, the prediction accuracy was lower than the Simple logistic regression model because the prediction target 'review/score' also has five categories.

#classification context

```
print(confusion_matrix(y_test2,predictions2))
```

>>> Output



#We created a model with about 65% accuracy

0.6465786155173558