```
NAME: GAURAV ULHAS KULKARNI
CLASS:CS3
ROLL NO: CS3-30
PRN:202401040114
COLAB LINK:https://colab.research.google.com/drive/1-CPq7RKM-
iNYEjTDJNkvj8aTWwf_e0Q2?usp=sharing
```

```
import pandas as pd
import numpy as np
import re
from collections import Counter

# Load the dataset
df = pd.read_csv('/content/drive/MyDrive/new dataset/Book Reviews.csv')
```

Double-click (or enter) to edit

1.Find the top 5 books with the highest average length of reviews (by character count)

```
Start coding or generate with AI.
```

```
df['Review Length'] = df['Review'].str.len()
top_books_by_review_length = df.groupby('Book')['Review Length'].mean().sort_values(ascending=False).head(5)
print(top_books_by_review_length)
```

```
Book
Cloud Atlas              6518.400000
Infinite Jest            6299.633333
Gravity's Rainbow        5301.800000
The Glass Bead Game      5155.833333
The Recognitions         4964.500000
Name: Review Length, dtype: float64
```

Double-click (or enter) to edit

2.Find the top 5 books with the longest average review (by number of characters)

```
df['Review Length'] = df['Review'].str.len()
top5_books_by_avg_review_length = df.groupby('Book')['Review Length'].mean().sort_values(ascending=False).head(5)
print(top5_books_by_avg_review_length)
```

```
Book
Cloud Atlas              6518.400000
Infinite Jest            6299.633333
Gravity's Rainbow        5301.800000
The Glass Bead Game      5155.833333
The Recognitions         4964.500000
Name: Review Length, dtype: float64
```

Double-click (or enter) to edit

3. List books that have exactly 1 review

```
books_with_one_review = df['Book'].value_counts()[df['Book'].value_counts() == 1].index.tolist()
print(books_with_one_review)
```

```
[]
```

4. Find what percentage of reviews mention the word "excellent"

```
excellent_percentage = df['Review'].str.contains('excellent', case=False, na=False).mean() * 100
print(f"{excellent_percentage:.2f}% of reviews mention 'excellent'")
```

```
2.97% of reviews mention 'excellent'
```

5. Find the median length of reviews for each book

```
median_review_length_per_book = df.groupby('Book')['Review Length'].median()
print(median_review_length_per_book)
```

```
Book
10:04                    1702.0
1984                     2264.0
1Q84 (1Q84, #1-3)        2666.0
2001: A Space Odyssey    1677.0
2666                     4206.0
                          ...
Zeno's Conscience        1802.0
Zorba the Greek          1981.5
alias Grace.             1848.0
Под игото                 679.0
태백산맥 세트                   534.5
Name: Review Length, Length: 1096, dtype: float64
```

6.Detect reviews that have duplicate text across different books

```
duplicate_reviews = df[df.duplicated(subset=['Review'], keep=False)].sort_values('Review')
print(duplicate_reviews)
```

```
       Unnamed: 0                             Book             Review \
26210        26210 The Path to the Spiders' Nests  \n|Bettie's Books|\n
25778       25778               Absolute Beginners  \n|Bettie's Books|\n
16841       16841                The Maltese Falcon          *3.5 stars*
12050       12050                 Northanger Abbey          *3.5 stars*
23434       23434                        Adam Bede          *3.5 stars*
...           ...                              ...                  ...
31921       31921             The Midnight Examiner                  NaN
31954       31954           The Return of the Soldier                NaN
31983       31983                   Humboldt's Gift                  NaN
32003       32003            The House of Doctor Dee                 NaN
32009       32009            The House of Doctor Dee                 NaN

              Review Date Review Length
26210        July 5, 2014          18.0
25778       March 6, 2014          18.0
16841      March 17, 2022          11.0
12050        May 29, 2021          11.0
23434    November 6, 2021          11.0
...                   ...           ...
31921     August 15, 2014           NaN
31954   September 6, 2018           NaN
31983    November 8, 2022           NaN
32003   February 19, 2023           NaN
32009    October 11, 2010           NaN

[406 rows x 5 columns]
```

7.Find which book received the oldest review in the dataset

```
oldest_review = df.loc[df['Review Date'].idxmin()]
print(oldest_review['Book'])
```

```
Morvern Callar (Morvern Callar Cycle, #1)
```

8.Find the average review length (characters) overall

```
df['Review Length'] = df['Review'].str.len()
average_review_length = df['Review Length'].mean()
print(average_review_length)
```

```
2040.5771119224473
```

9.Find books where at least one review contains the word 'boring'

```
books_with_boring = df[df['Review'].str.contains('boring', case=False, na=False)]['Book'].unique()
print(books_with_boring)
```

```
['1984' 'Jane Eyre' 'Animal Farm' 'The Catcher in the Rye'  'The Picture of Dorian
Gray' 'Little Women' 'The Count of Monte Cristo'  'One Hundred Years of Solitude'
"The Handmaid’s Tale (The Handmaid's Tale, #1)" 'Les Misérables'
 'Dracula' 'The Grapes of Wrath' 'The Adventures of Huckleberry Finn'
 'Great Expectations' 'Slaughterhouse-Five'  'The Curious Incident of the Dog in
the Night-Time' 'Rebecca'  'The Bell Jar' 'The Old Man and the Sea' 'The Scarlet
Letter'  'War and Peace' 'Perfume: The Story of a Murderer'  'Interview with the
Vampire (The Vampire Chronicles, #1)'
 'A Clockwork Orange' 'Persuasion' 'In Cold Blood'
 'The Brothers Karamazov' 'The Time Machine' 'A Prayer for Owen Meany'
 'The Name of the Rose' 'Atonement' 'Oliver Twist' 'Robinson Crusoe'
 "Gulliver's Travels: Travels into Several Remote Nations of the World."
 'Watchmen' 'On the Road' 'Don Quixote' 'The House of the Spirits'
 "Uncle Tom's Cabin" 'The Sun Also Rises' 'The Reader'
 'The World According to Garp' 'Candide' 'The Arabian Nights'
 'Mansfield Park' 'As a Man Grows Older (New York Review Books Classics)'
 'Euphues: The Anatomy of Wit' 'The Making of Americans'
 'Rob Roy (Waverley Novels, #4)' '10:04' 'To Have and Have Not'
 'Lieutenant Gustl' 'Super-Cannes' 'Crash' 'Her Privates We'
 'Pointed Roofs, Backwater, Honeycomb (Pilgrimage, Volume 1)'
 'Impressions of Africa'
 'A Dance to the Music of Time: 1st Movement (A Dance to the Music of Time, #1-3)'
 "Wittgenstein's Mistress" 'Clarissa, or, the History of a Young Lady'
 'Austerlitz' 'Röda Rummet'
 'Fanny Hill, or Memoirs of a Woman of Pleasure' 'The Riddle of the Sands'
 'Goodbye to Berlin' 'Born in Exile' 'The Plumed Serpent'
 'Fantômas (Fantômas, #1)' 'The Glass Key' 'The Golden Bowl'
 "Parade's End" 'Henderson the Rain King' 'The Iron Heel'
 'The Master of Ballantrae' 'The Child in Time' 'Disappearance'
 "The Old Wives' Tale" 'Whatever' 'The Confusions of Young Törless'
 'The Sea' 'Living' 'The Nice and the Good' 'The Radiant Way'
 'The Heather Blazing' 'Elective Affinities' 'The Blindness of the Heart'
 'Belle du Seigneur'
 'The Book of Evidence (The Freddie Montgomery Trilogy #1)'
 'The Enigma of Arrival: A Novel in Five Sections'
 'Memoirs of Martinus Scriblerus' 'Loving' 'Fury' 'Party Going'
 'The Thinking Reed' 'Simplicissimus' 'Fall on Your Knees'
 'Petals of Blood' 'The Guiltless' 'Black Box' 'The First Garden'
 'How the Dead Live' 'London Orbital' 'Vernon God Little' 'The Birds'
 'The Castle of Crossed Destinies' 'The Holy Terrors'
 'Almost Transparent Blue' "Fool's Gold" 'The Marble Faun'
 'The Garden Where the Brass Band Played' 'Mr. Norris Changes Trains'
 'The Bell' 'Eyeless in Gaza' 'Julie, or the New Heloise'
 'Strait is the Gate' 'The Quest for Christa T.' 'The Victim'
 "Ratner's Star" 'The Circle (The Circle, #1)' 'London Fields'
 'Joseph Andrews' 'The Shadow Lines' "Pavel's Letters" 'Antic Hay'
 'A Girl Is a Half-formed Thing' 'Borstal Boy' 'The Dark Child'
 'The Comfort of Strangers' 'The Last World' 'Against the Day' 'Eva Trout'
 'The Recognitions' 'Untouchable' 'The Mandarins' 'Amongst Women' 'Indigo'
 'The Unfortunate Traveller and Other Works' 'Tono-Bungay'
 'The Lion of Flanders' 'Het verboden rijk'
 'Queen Margot (The Last Valois, #1)' 'De wetten' 'There but for the'
 'Night Boat to Tangier' 'Wild Harbour' 'Jacques the Fatalist'
 'Ferdydurke' 'Blindness' 'News from Nowhere' 'Platero y yo' 'Dead Babies'
```

10.Find the earliest and latest review dates for each book

```
earliest_latest_reviews = df.groupby('Book')['Review Date'].agg(['min', 'max'])
print(earliest_latest_reviews)
```

```
                                  min                              max
Book
10:04                      April 3, 2023            September 5, 2014
1984                      April 15, 2012            September 6, 2022
1Q84 (1Q84, #1-3)        August 1, 2022    Want to read|October 6, 2011
2001: A Space Odyssey April 25, 2022               September 14, 2017
2666                      April 11, 2021           September 17, 2012
...                                  ...                          ...
Zeno's Conscience         April 16, 2023            September 8, 2019
Zorba the Greek           April 10, 2008           September 30, 2015
alias Grace.              April 13, 2022            September 2, 2015
Под игото                April 10, 2011            September 4, 2017
태백산맥 세트
                                May 26, 2023 Want to read|September 6, 2021

[1096 rows x 2 columns]
```

11.Find the percentage of reviews that are less than 50 characters

```
short_reviews_pct = (df['Review Length'] < 50).mean() * 100
print(f"{short_reviews_pct:.2f}% of reviews are shorter than 50 characters")
```

```
2.55% of reviews are shorter than 50 characters
```

12.List books whose average review length is greater than 300 characters

```
df['Review Length'] = df['Review'].str.len()
books_with_long_reviews = df.groupby('Book')['Review Length'].mean()
long_books = books_with_long_reviews[books_with_long_reviews > 300].index.tolist()
print(long_books)
```

```
['10:04', '1984', '1Q84 (1Q84, #1-3)', '2001: A Space Odyssey', '2666', 'A Ballad for Georg Henig', 'A Bend i
```

13.Peak review day (exact date with most reviews)

```
peak_day = df['Review Date'].value_counts().idxmax()
print(f"The peak review date was {peak_day}.")
```

```
The peak review date was December 4, 2013.
```

14.Book with highest proportion of long reviews (>1000 chars)

```
df['is_long'] = df['Review Length'] > 1000
long_review_ratio = df.groupby('Book')['is_long'].mean()
top_book = long_review_ratio.idxmax()
print(f"'{top_book}' has the highest proportion of long reviews.")
```

```
'Austerlitz' has the highest proportion of long reviews.
```

14.Reviews that mention the word “disappointed”

```
disappointed_reviews = df[df['Review'].str.contains('disappointed', case=False, na=False)]
print(f"{len(disappointed_reviews)} reviews mention the word 'disappointed'.")
```

```
487 reviews mention the word 'disappointed'.
```

15.Find outliers in review lengths

```
q1 = df['Review Length'].quantile(0.25) q3 = df['Review Length'].quantile(0.75) iqr = q3 - q1
outliers = df[(df['Review Length'] < q1 - 1.5 * iqr) | (df['Review Length'] > q3 + 1.5 * iqr)]
print(f"There are {len(outliers)} outlier reviews based on review length.")
```

```
There are 1900 outlier reviews based on review length.
```

16.Number of books with average review length over 1000 characters

```
long_books = df.groupby('Book')['Review Length'].mean()
over_1000 = (long_books > 1000).sum()
print(f"{over_1000} books have average review length over 1000 characters.")
```

```
992 books have average review length over 1000 characters.
```

17.Longest review in the dataset

```
longest_review_idx = df['Review Length'].idxmax()
print("Longest review:", df.loc[longest_review_idx, 'Review'])
```

```
Longest review: A mumbo-jumbo of words trying desperately to congeal into a plot. And failing at it, miserabl
```

18.Books reviewed only once

```
rare_books = df['Book'].value_counts()
unique_reviews = rare_books[rare_books == 1].index.tolist()
print(f"{len(unique_reviews)} books were reviewed only once.")
```

```
0 books were reviewed only once.
```

19.Count how many reviews are marked as long

```
long_reviews_count = df['is_long'].sum()
print(f"Number of long reviews: {long_reviews_count}")
```

```
Number of long reviews: 19035
```

20.Get the number of reviews per book:

```
review_counts = df['Book'].value_counts()
print(review_counts.head(10))
```

```
Book
In the Heart of the Country            30
```

```
To Kill a Mockingbird 30 1984 30 Jane Eyre
30
Animal Farm 30
The Catcher in the Rye 30 The Picture of
Dorian Gray 30 Little Women 30 La
Désobéissance 30
The Autobiography of Alice B. Toklas 30
Name: count, dtype: int64
```