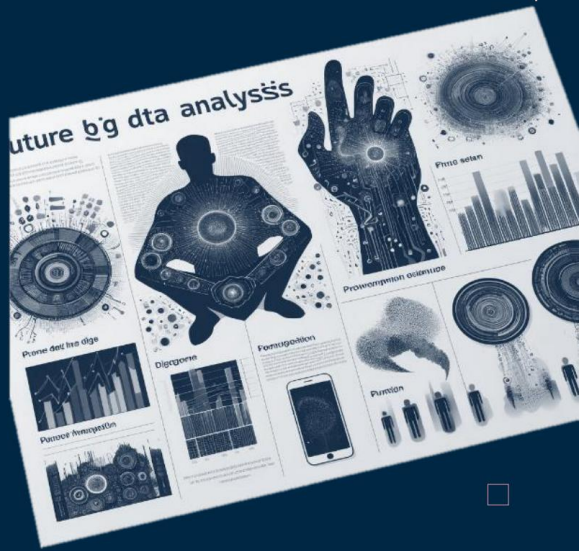
To understand domestic online bookstore consumption trends

“Post-Covid19 Data Analysis”

Shin Jeong-yoon, Yonsei IT Future

Education Institute



Contents

DURING COVID PANDAMIC

#자가 격리

재택근무

#비대면

온라인

Articles DURING COVID

'Corona Depression' also known as **'Corona Blues'**

May 12, 2023

https://www.docdocdoc.co.kr/news/articleView.html?idxno=3005764

October 4, 2022

https://www.whosaeng.com/139197

Personal Experience during COVID-19

Military Life (January 3, 2022 - April 11, 2023) with COVID-19

Personal effects:

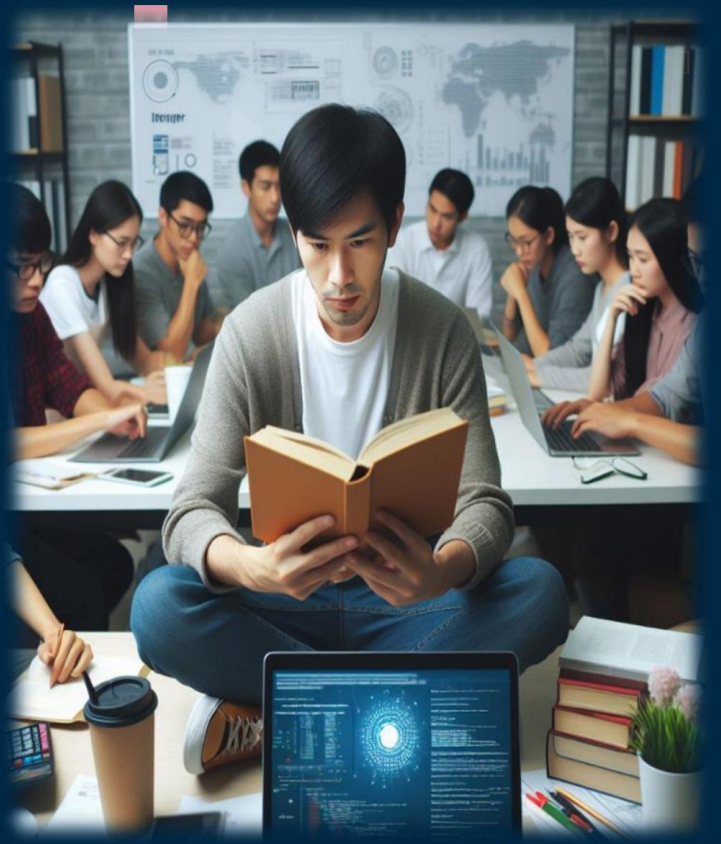
desire for achievement ↑

confidence ↑

Process of Brain-Storming

1. Analysis of the Namseoul Library loan list just before the project.
2. What should I do? 3.
What do I like? 4. Why do I read books? 5.
What kind of learning/
enlightenment do I get?
6. What about other people besides me?

What is the subject of the
data analysis? • A well-known representative online
bookstore? • Kyobo Bookstore's top 100 best sellers.
• Discovering outliers in field-specific figures that change over the years
• 4 years of consumer reviews



Hypothesis

What types of books captured the public's attention during the COVID-19 period (2020-2023) ? What kind of books captured the public's attention?

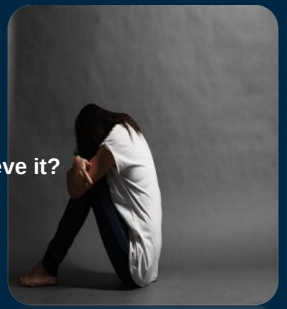
Hypothesis: After Corona, ____ people will **spend the most** on reading **novels/self-help** books .

Approach:
Identify which fields of books have a high proportion of books.

What kind of reviews did the books in each field get?

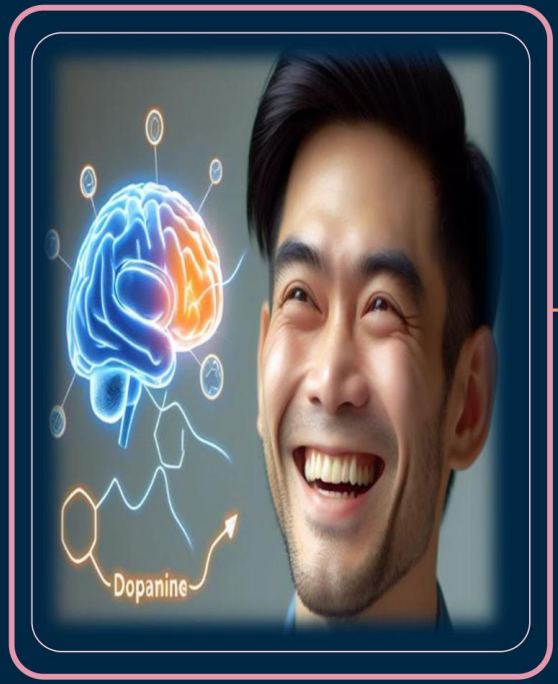
Background of Hypothesis [Background]

Since COVID-19, social connections have become more difficult to form, leaving people with fewer opportunities to release dopamine (the happiness hormone). **So where do people turn for relief? Exercise?** Books? **Travel?** These all stimulate the release of cortisol (the stress hormone) . (Continued release leads to depression.) **No stress! How can we relieve it?**

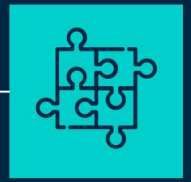


Why did other people read such books?

My hypothesis is that, as humans who must navigate life, novels help us shake off feelings of insecurity (loneliness/depression) and **provide comfort (serotonin** [stability] -> stress-relieving hormone). Furthermore, we read self-improvement books to build confidence and fulfillment. (Dopamine [happiness])



Data Collecting Process [feat. [Python]



01

Genre-Collecting

--Selenium
--BeautifulSoup
--ChromeWebdriver

02

elements

collecting

-Pandas
-HTML parser

03

Understanding

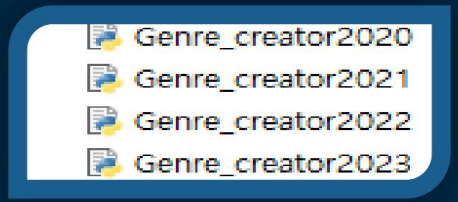
Consumer Psychology

Through TARGET

Reviews (feat. PowerBI)

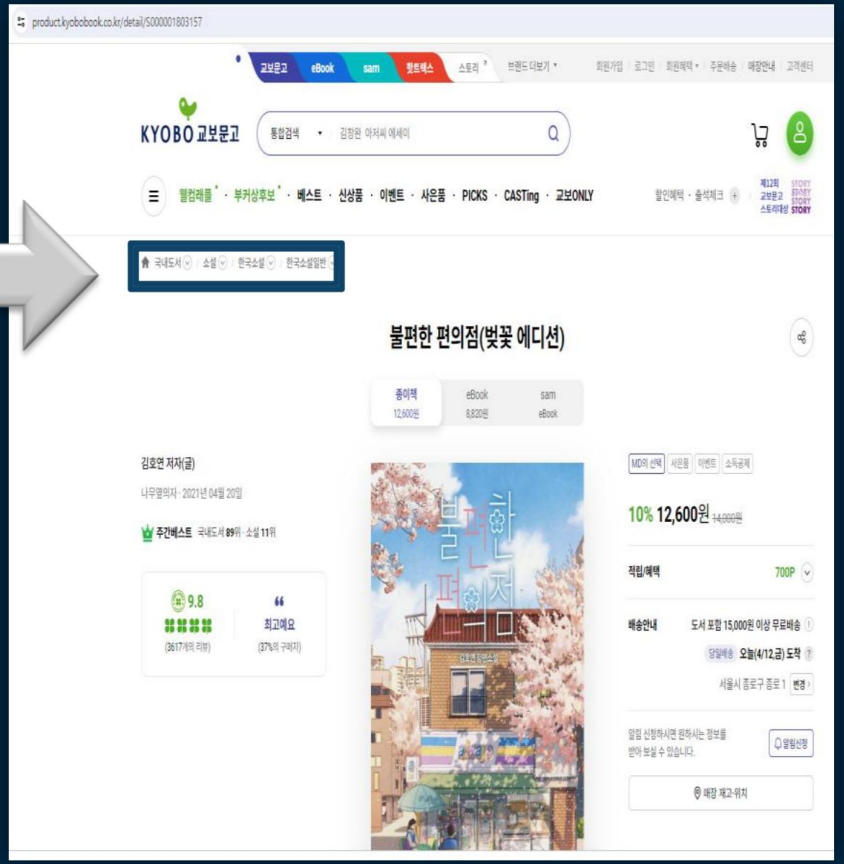
Genre Collecting

01



1. Identifying the location of field information

- There is no field information on the main page
- Access detailed page information and retrieve only genre elements.



Create a link to the detailed page

```
# genre_creator는 년도 별 액셀파일에 따라 달라지는 장를 추출 module이다
# excel파일을 웹에서 다운받아 '판매상품ID'에 해당하는 열을 교보문고에서
#상세페이지로 이동할 수 있는 https://product.kyobobook.co.kr/detail/ 과 결합하여
# 상세페이지에서 장르 정보만 따와서 excel 파일로 저장해주는 모듈이다.

# 변수 line:
#   20 : 저장된 액셀파일


# 제품코드 추출위해(제품코드가 html상 존재x) 액셀파일 불러오기:
import pandas as pd
df = pd.read_excel("./project/Genrelist_of_bestseller2022.xlsx")
#%%
# 1위 부터 100위까지 상세페이지 링크 리스트자료형으로 변환.
# glist = 판매상품ID를 list자료형으로 변환
glist = df['판매상품ID'].tolist()
# 상세페이지 format
# [https://product.kyobobook.co.kr/detail/] + ['판매상품ID']
# genre_link(glist) : 100개의 상세페이지 링크 완성하는 함수
# genre_data : 상세페이지 링크

def genre_link(glist):
    genre_data = []
    sampleurl = "https://product.kyobobook.co.kr/detail/"
    for pid in glist:
        genre_data.append(sampleurl + str(pid))
    return genre_data

genre_data = genre_link(glist)
print(len(genre_data))
```

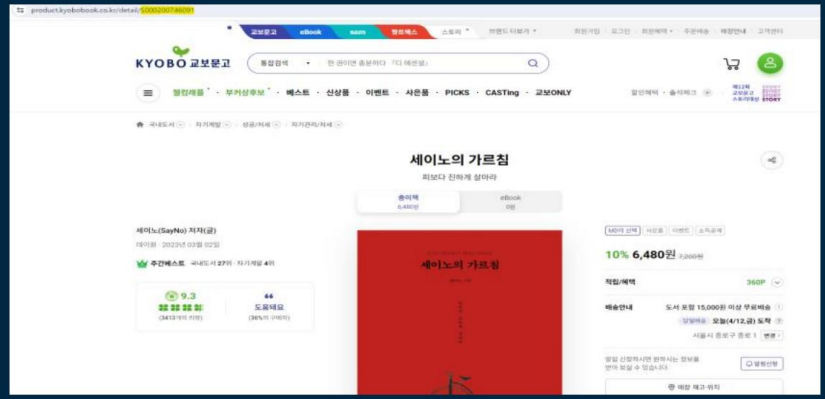
```
url_cnt: 97 https://product.kyobobook.co.kr/detail/S000000610793
soup: True
4
url_cnt: 98 https://product.kyobobook.co.kr/detail/S000001908656
soup: True
5
url_cnt: 99 https://product.kyobobook.co.kr/detail/S000000582008
soup: True
4
url_cnt: 100 https://product.kyobobook.co.kr/detail/S000001020256
soup: True
5
url_cnt: 101 https://product.kyobobook.co.kr/detail/S000001708482
```

Excel다운로드

순위	상품코드	판매상품ID	상품명
1	9791168473690	S000200746091	세이노의 가르침
2	9788997575169	S000001619177	원씽(The One Thing)(60만 부 기념 스페셜 에디션)
3	9788901272580	S000202340164	역행자(확장판)
4	9788954699075	S000208719388	도시와 그 불확실한 벽

Product ID for sale

product.kyobobook.co.kr/detail/S000200746091



Extract data in 20 volumes at a time (preventing output overload in Spyder)

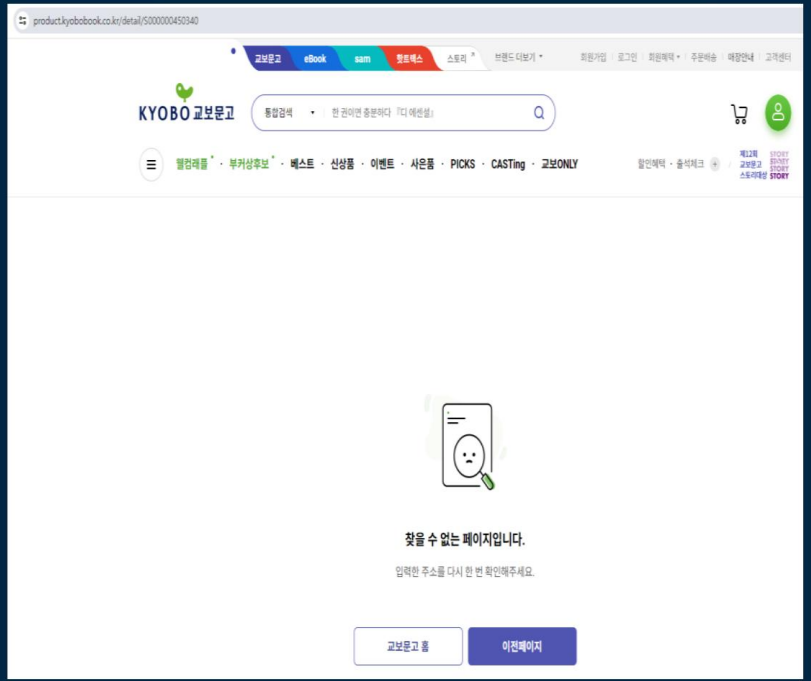
```python
# spyder 과부화 방지를 위해 상세페이지링크 20개씩 끊어줌.
# s_list 20개씩 리스트로 변환(list안에 list)
split_data = [
    genre_data[0:20],
    genre_data[20:40],
    genre_data[40:60],
    genre_data[60:80],
    genre_data[80:100]
]
print(split_data)
#%%
# 0~20개 상세페이지에서 장르 추출(dict자료형으로 추출)
# genre_list1 : 상세페이지에서 장르(0~20개)에 대한 정보를 담을 []
# chunk1 : 상세페이지 링크 20개
from selenium.webdriver import Chrome
from bs4 import BeautifulSoup

driver = Chrome()

genre_list1 = []
chunk1 = split_data[0]

for url in chunk1:
    driver.get(url)
    driver.implicitly_wait(3) # 3초대기(웹로드)

    html = driver.page_source
    soup = BeautifulSoup(html, "lxml")

    genre_elements = soup.find_all('a', attrs={'class': 'btn_sub_depth'})
    # 구성요소값 4개 구성
    if len(genre_elements) ≥ 2: # 구성요소 2개 이상일 때
        second_genre = genre_elements[1].text.strip() # 요소값 중 2번 째 값 공백제거 후 추출
        genre_list1.append({'장르': second_genre})

driver.quit()
```

```python
# 81~100개 상세페이지링크에 대한 장르정보
from selenium.webdriver import Chrome
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.options import Options

chrome_options = Options()
chrome_options.add_argument("--headless")
driver = Chrome(options=chrome_options)


genre_list5 = []
chunk5 = split_data[4]

for url in chunk5:
    driver.get(url)
    driver.implicitly_wait(3)

    html = driver.page_source
    soup = BeautifulSoup(html, "lxml")

    genre_elements = soup.find_all('a', attrs={'class': 'btn_sub_depth'})
    if len(genre_elements) ≥ 2:
        second_genre = genre_elements[1].text.strip()
        genre_list5.append({'장르': second_genre})

driver.quit()
#%%

import pandas as pd
combined_data = genre_list1 + genre_list2 + genre_list3 + genre_list4 + genre_list5
Genre_dataframe = pd.DataFrame(combined_data)

Genre_dataframe.to_excel('./Project/Genrelist_of_bestseller2022.xlsx', index=False)
```

Problems found on Kyobo Bookstore's website:

I accessed the detailed page of the 22-year data, but the genre There is data without information.

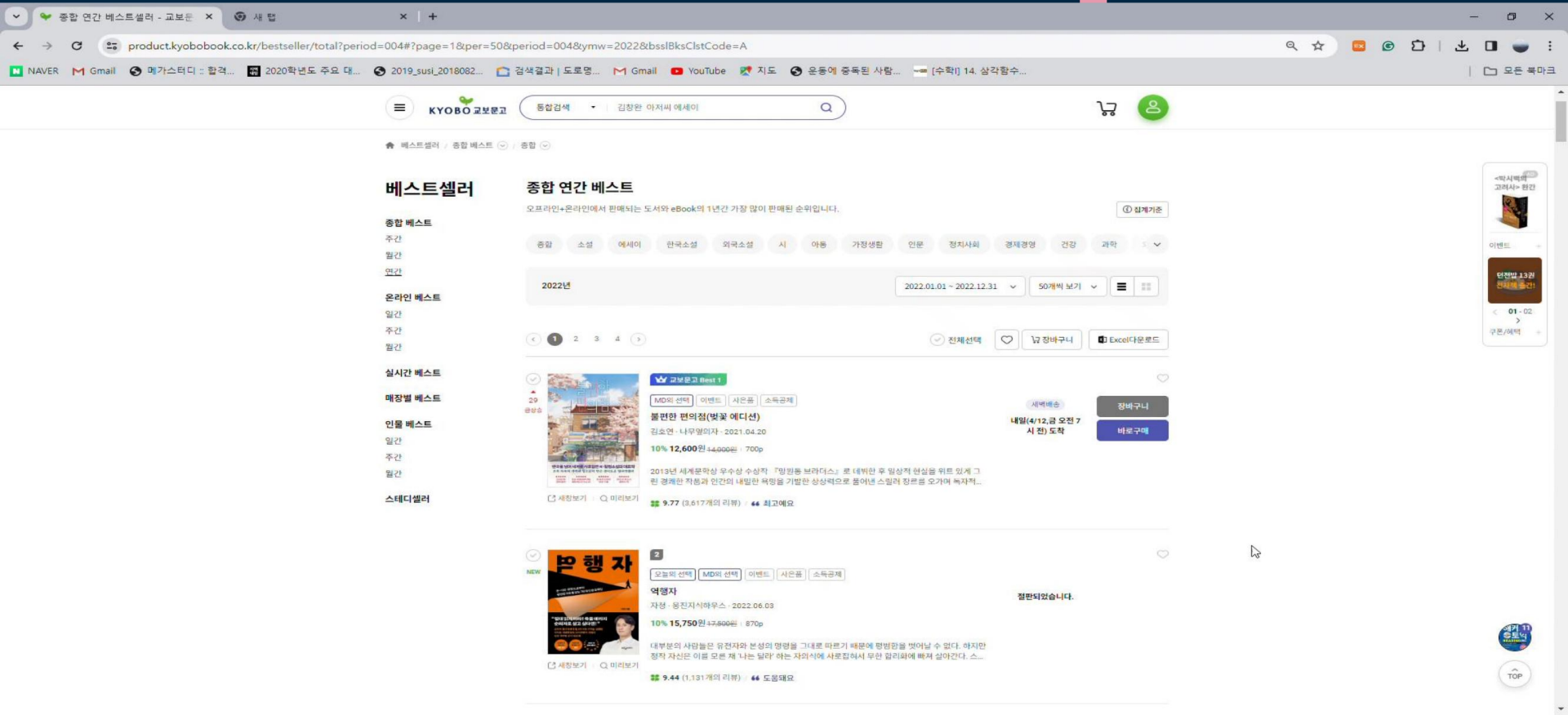
product.kyobobook.co.kr/detail/S000200746091



Process of Problem Solving

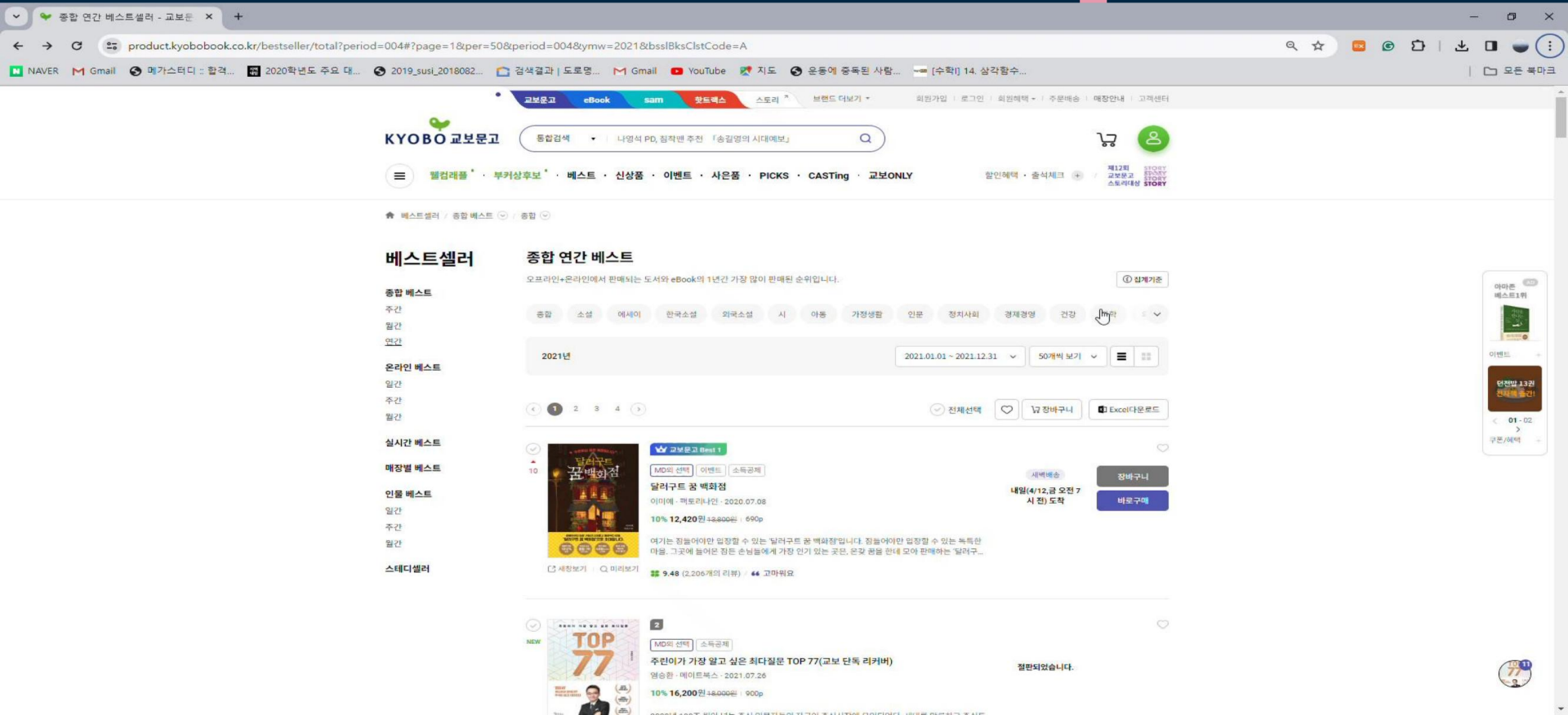
2021 vs 2022

No detailed page exists



2021 vs 2022

Detail page exists O



Fact discovered:

"Out-of-print books do not provide detailed pages.

Data Cleansing 1:

As confirmed in the previous slide, when importing 2022 data ('Out of print' ÿ 'Detail page' does not exist), in order to prevent the above problem in advance, we will import 2021 data for all bestsellers, not just 1st to 100th place.

Genre extraction

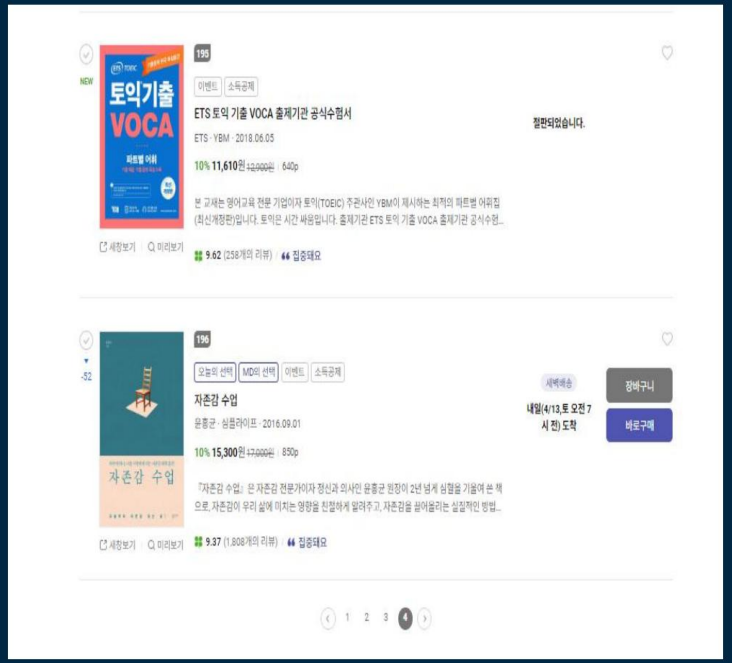
```
from selenium.webdriver import Chrome
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.options import Options

# 크롬옵션객체를 생성하여 파이썬 내부에서 작업처리
chrome_options = Options()
chrome_options.add_argument("--headless")
driver = Chrome(options=chrome_options)


genre_list = []
chunk = genre_data

for url in chunk:
    driver.get(url)
    driver.implicitly_wait(3)

    html = driver.page_source
    soup = BeautifulSoup(html, "lxml")

    genre_elements = soup.find_all('a', attrs={'class': 'btn_sub_depth'})
    if genre_elements ≠ None:
        if len(genre_elements) ≥ 2:
            second_genre = genre_elements[1].text.strip()
            genre_list.append({'장르': second_genre})
    else:
        genre_list.append({'장르': '절판'})


driver.quit()
```

When limiting the range from 1 to 100, specify the chunk range as follows:

```
genre_list = []
chunk = genre_data[0:100]
```

Data Cleansing 2:

Process of narrowing the range to the top 100



genre_data	list	197	['https://product.…
genre_elements	element.ResultSet	5	ResultSet object o…
genre_list	list	100	[{'장르':'소설'}, {'…

```
Genre_dataframe = pd.DataFrame(genre_list)
Genre_dataframe.to_excel("./project/Genrelist_of_bestseller2022.xlsx", index=False)
df = pd.read_excel('./Project/Genrelist_of_bestseller2021.xlsx')
df2.drop(df2.index[100:], inplace=True) # 데이터 축소(196권 -> 100권)
df2.to_excel('./Project/Genrelist_of_bestseller2021.xlsx', index=False)
```

Data Cleansing 3:

When processing out-of-print data for 2022 data

Modify the code to output 'Out of Print'

```
from selenium.webdriver import Chrome
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.options import Options

# 크롬옵션객체를 생성하여 파이썬 내부에서 작업처리
chrome_options = Options()
chrome_options.add_argument("--headless")
driver = Chrome(options=chrome_options)

genre_list = []
chunk = genre_data[0:100]
#url_cnt = 0

for url in chunk:
    #url_cnt += 1
    #print("url_cnt:", url_cnt, url)
    driver.get(url)
    driver.implicitly_wait(3)

    html = driver.page_source
    soup = BeautifulSoup(html, "lxml")
    #print('soup:', soup != None)

    genre_elements = soup.find_all('a', attrs={'class': 'btn_sub_depth'})
    #print(len(genre_elements))
    if genre_elements != None:
        if len(genre_elements) >= 2:
            second_genre = genre_elements[1].text.strip()
            genre_list.append({'장르': second_genre})
        else:
            genre_list.append({'장르': '절판'})
    else:
        genre_list.append({'장르': '절판'})


driver.quit()
```

장르	count
소설	27
경제/경영	15
인문	12
자기계발	11
시/에세이	8
어린이(초등)	8
정치/사회	5
과학	3
외국어	3
절판	2
역사/문화	2
예술/대중문화	1
가정/육아	1
컴퓨터/IT	1
건강	1

Changes after code change

Index	소설	경제/경영	인문	자기계발	시/에세이	어린이(초등)	정치/사회	과학	외국어	절판	역사/문화
권수	27	15	12	11	8	8	5	3	3	2	2

Data Cleansing 4:

Using Numpy for 2022 Data

```
df3 = pd.read_excel('./Excel_file/Genrelist_of_bestseller2022.xlsx')
import numpy as np
npt = np.array(df3)
noprint = np.where(npt == '절판')
# 1행/ 92행 값 웹 상에서 검색 후 변경해줌.
# 1번째 행의 '장르' 값인 '절판'을 '자기계발'로 변경
df3.loc[0, '장르'] = '자기계발'
# 92번째 행의 '장르' 값인 '절판'을 '외국어'로 변경
df3.loc[92, '장르'] = '외국어'
df3.to_excel('./Excel_file/Genrelist_of_bestseller2022.xlsx')
```

```
df3 = pd.read_excel('./Project/Genrelist_of_bestseller2022.xlsx')

import numpy as np
npt = np.array(df3)
noprint = np.where(npt == '절판')
```

noprint - Tuple (2 elements)

Inde	Type	Size	
0	Array of int64	(2,)	[1 92]
1	Array of int64	(2,)	[0 0]

'Out of print' ÿ Name changed to fit the field

```
# 2022년 데이터를 받아오는 과정에서 이상값 발견:

df3 = pd.read_excel('./Project/Genrelist_of_bestseller2022.xlsx')

import numpy as np
npt = np.array(df3)
noprint = np.where(npt == '절판')
# 1행/ 92행 값 웹 상에서 검색 후 변경해줌.
# 1번째 행의 '장르' 값을 '자기계발'로 변경
df3.loc[0, '장르'] = '자기계발'
# 93번째 행의 '장르' 값을 '외국어'로 변경
df3.loc[92, '장르'] = '외국어'
df3.to_excel('./Project/Genrelist_of_bestseller2022.xlsx')
```

noprint - Tuple (2 elements)

Inde	Type	Size	Value
0	Array of int64	(2,)	[1 92]
1	Array of int64	(2,)	[0 0]

npt - NumPy object array

	0
0	소설
1	절판
2	소설
3	정치/사회
4	소설
5	경제/경영
6	소설
7	인문
8	과학

Format Resize

df3 - DataFrame

Index	장르
0	자기계발
1	자기계발
2	소설
3	정치/사회
4	소설
5	경제/경영
6	소설
7	인문
8	과학

Format Resize Background

Data Integration

Functionalization of data integration process for 2020-2023 based on refined data ÿ 4-year dictionary data

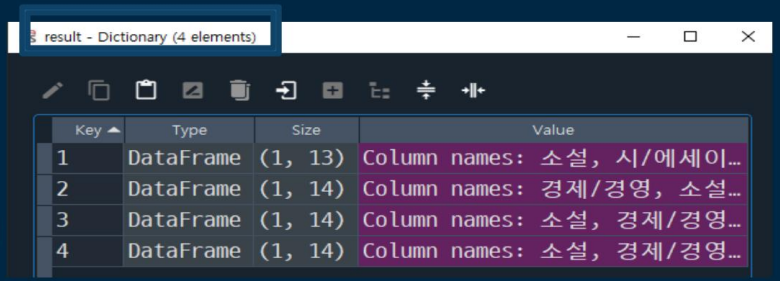
```
# 연도 별로 데이터전처리

# 열거된 dataframe바탕으로 장르 별 빈도수 계산
# 2020년
df_gen = df_dict['df1']
# 각 데이터프레임에 대해 '장르'열의 빈도수를 딕셔너리로 변환
genre_dict = df_gen['장르'].value_counts().to_dict()
# 딕셔너리를 데이터프레임으로 변환하여 출력
genre_df_2020 = pd.DataFrame([genre_dict])
genre_df_2020.rename(index = { 0 :'권수'}, inplace=True)

# 2021년
df_gen = df_dict['df2']
# 각 데이터프레임에 대해 '장르'열의 빈도수를 딕셔너리로 변환
genre_dict = df_gen['장르'].value_counts().to_dict()
# 딕셔너리를 데이터프레임으로 변환하여 출력
genre_df_2021 = pd.DataFrame([genre_dict])
genre_df_2021.rename(index = { 0 :'권수'}, inplace=True)

# 2022년
df_gen = df_dict['df3']
# 각 데이터프레임에 대해 '장르'열의 빈도수를 딕셔너리로 변환
genre_dict = df_gen['장르'].value_counts().to_dict()
# 딕셔너리를 데이터프레임으로 변환하여 출력
genre_df_2022 = pd.DataFrame([genre_dict])
genre_df_2022.rename(index = { 0 :'권수'}, inplace=True)

# 2023년
df_gen = df_dict['df4']
# 각 데이터프레임에 대해 '장르'열의 빈도수를 딕셔너리로 변환
genre_dict = df_gen['장르'].value_counts().to_dict()
# 딕셔너리를 데이터프레임으로 변환하여 출력
genre_df_2023 = pd.DataFrame([genre_dict])
genre_df_2023.rename(index = { 0 :'권수'}, inplace=True)


genre_df_2020.to_excel('./Project/Num_of_Genrelist2020.xlsx')
genre_df_2021.to_excel('./Project/Num_of_Genrelist2021.xlsx')
genre_df_2022.to_excel('./Project/Num_of_Genrelist2022.xlsx')
genre_df_2023.to_excel('./Project/Num_of_Genrelist2023.xlsx')
```

```
# 4개의 장르 파일 -> 빈 리스트에 저장
excel_f = []
for year in range(2020, 2024):
    excel_f.append(pd.read_excel(f"./Project/Genrelist_of_bestseller{year}.xlsx"))
    print(excel_f)


# excel_f 의 value값이 dataframe이기 때문에,
#value값을 위와 같이 리스트자료형으로 묶어주었다.
#excel_f에서 4개파일의 value들을 차례대로 함수에 할당할 수 있는 변수로 만들어줌.

# 4개년도 한번에 처리
import pandas as pd

def cleans_integrate(excel_f):
    genre_freq_years = {} # 연도 별 '장르'빈도수 저장할 딕셔너리
    for idx, df in enumerate(excel_f, start = 1):
        genre_freq = df['장르'].value_counts().to_dict() # '장르'열 빈도수 계산 후 딕셔너리로 변환
        genre_df = pd.DataFrame([genre_freq]) # 딕셔너리 -> dataframe 변환
        # f'{idx}'에서 idx값을 위에 열거 순에 따라 인덱스 값을 1~4까지 반환
        genre_df.rename(index={0: f'{idx}'}, inplace=True)
        genre_freq_years[f'{idx}'] = genre_df


    return genre_freq_years
# 'cleans_integrate' 함수 호출 [데이터 정제 및 통합]
result = cleans_integrate(excel_f)
```

Data Integration

Result (=4-year dictionary data) ÿ Convert only values to list ÿ Merge 'result_list' with concat()

```python
import pandas as pd

def cleans_integrate(excel_f):
    genre_freq_years = {} # 연도 별 '장르'빈도수 저장할 딕셔너리
    for idx, df in enumerate(excel_f, start = 1):
        genre_freq = df['장르'].value_counts().to_dict() # '장르'열 빈도수 계산 후 딕셔너리로 변환
        genre_df = pd.DataFrame([genre_freq]) # 딕셔너리 -> dataframe 변환
        # f'{idx}'에서 idx값을 위에 열거 순에 따라 인덱스 값을 1~4까지 반환
        genre_df.rename(index={0: f'{idx}'}, inplace=True)
        genre_freq_years[f'{idx}'] = genre_df

    return genre_freq_years
# 'cleans_integrate' 함수 호출 [데이터 정제 및 통합]
result = cleans_integrate(excel_f)
```



result	dict	4	{'1':Dataframe, '2':Dataframe, '3':Datafr…
result_list	list	4	[Dataframe, Dataframe, Dataframe, Datafra…

```python
#result 결과 = {key : 1~4 , value : 4년치 dataframe}
result_list = list(result.values()) # DataFrame인 value값 리스트에 넣기
result_integrate = pd.concat(result_list) # value값들을 합쳐 DataFrame만들기
tot_book = [100,100,100,100]
result_integrate['총 권수'] = tot_book
result_integrate.index = range(2020, 2024)
result_integrate = result_integrate.rename_axis('연도')
result_integrate.to_excel('./Excel_file/Genre_stat.xlsx', index=True)
```

Result

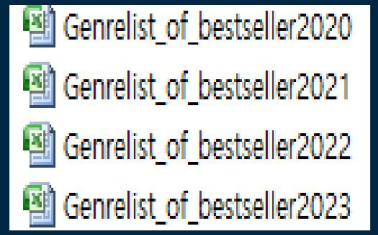
Missing values are handled by Power-BI

연도	소설	시/에세이	경제/경영	자기계발	인문	어린이(초등)	외국어	정치/사회	과학	역사/문화	청소년	예술/대중문화	가정/육아	만화	절판	컴퓨터/IT	건강	총 권수
2020	18	17	16	14	13	6	5	3	3	2	1	1	1	nan	nan	nan	nan	100
2021	22	12	23	13	10	6	5	1	2	1	2	1	1	1	nan	nan	nan	100
2022	26	8	15	12	12	8	4	5	3	2	nan	1	1	nan	1	1	1	100
2023	22	5	17	15	14	5	6	2	5	2	1	nan	nan	4	nan	1	1	100

Result

Extracting 20 volumes per year for 2023 vs. 196 volumes per year for 2021: which will be faster? (Each Excel file takes about 7 minutes)

Create Excel files by year



Integrated data

Sample:

	A
1	장르
2	자기계발
3	경제/경영
4	소설
5	자기계발
97	어린이(초등)
98	소설
99	경제/경영
100	시/에세이
101	외국어

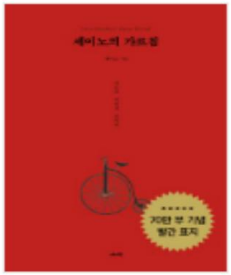
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	연도	소설	시/에세이	경제/경영	자기계발	인문	어린이(초등	외국어	정치/사회	과학	역사/문화	청소년	예술/대중문화	가정/육아	만화	절판	컴퓨터/IT	건강	총 권수
2	2020	18	17	16	14	13	6	5	3	3	2	1	1	1					100
3	2021	22	12	23	13	10	6	5	1	2	1	2	1	1	1				100
4	2022	26	8	15	12	12	8	4	5	3	2		1	1		1	1	1	100
5	2023	22	5	17	15	14	5	6	2	5	2	1			4		1	1	100

Elements collecting

02

Where is the title/author/one-line review?

NEW



새창보기 | 미리보기

교보문고 Best 1

MD의 선택 | 이벤트 | 사은품 | 소득공제

세이노의 가르침

세이노(SayNo) · 데이원 · 2023.03.02

10% 6,480원 ~~7,200원~~ | 360p

2000년부터 발표된 그의 주옥같은 글들. 독자들이 자발적으로 만든 제본서는 물론, 전자책과 앱까지 나왔던 《세이노의 가르침》이 드디어 전국 서점에서 독자들을 마주한다. 여러 판본을 모으고 ...

9.26 (3,413개의 리뷰) 도움돼요

당일배송

오늘(4/12,금) 도착

장바구니

바로구매

27

금상승

새창보기 | 미리보기

2

오늘의 선택 | MD의 선택 | 이벤트 | 사은품 | 소득공제

원씽(The One Thing)(60만 부 기념 스페셜 에디션)

게리 켈러 외 · 비즈니스북스 · 2013.08.30

10% 15,120원 ~~16,800원~~ | 840p

2022, 2023년 연이어 종합 베스트셀러에 선정된 《원씽》이 국내 60만 부 판매를 기념하여 스페셜 에디션으로 탄생했다! 수많은 언론과 매체, 인플루언서들이 연이어 《원씽》을 다시 찾아 읽고 강...

9.66 (1,456개의 리뷰) 도움돼요

당일배송

오늘(4/12,금) 도착

장바구니

바로구매

Bringing HTML (Feat. Web-Scrolling) -1

```
# 아래 3개 정보를 교보문고 연간베스트셀러 페이지에서 뽑아 출력하기
# 책 제목/저자/한줄평
#  Web-Scrapping Tools
#   -BeautifulSoup
#   -selenium
#   -chrome webdriver
#   -pandas

# 아래 코드는 top100 베스트셀러의 제목/저자/한줄평을 액셀파일로 변환추출해주는 모듈
# 변수 line: 65(1페이지: 1~50위리스트)
#           66(2페이지: 51~100위 리스트)
#           74(파일이름 변환)

import pandas as pd
import time
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
def web_scroll(url):

    options = Options()
    options.headless = False  # GUI 웹 구현
    options.add_argument('--window-size=968,1056') # 절반크기 화면
    driver = webdriver.Chrome(options=options)
    driver.get(url)
    time.sleep(3) # 웹 로드
    step = 0.9 #웹 페이지의 90%만큼 이동
    scroll = 8 # 총 8번이 스크롤 될 동안 실행
    screen_size = driver.execute_script("return window.screen.height;") # 1056pixel
    while scroll> 0:
        driver.execute_script("window.scrollTo(0,{screen_height}*{step})".format(screen_height=screen_size, step=step))
        step += 0.9
        step+= 0.9
        time.sleep(3)
        scroll -= 1
    html_text = driver.page_source #웹페이지의 소스코드(html) python에 가져오기
    driver.close()
    soup = BeautifulSoup(html_text,'lxml') # lxml 파서는 큰 html문서처리에 용이(반면에 html_parser는 간단한 문서처리에 활용)
    return soup
```

See Web Scraping p298

Def web_scroll(url): What is the url variable?

Def web_scroll(url): When executed, **the web page is scaled to the scroll size.**

```
from selenium import webdriver

# Chrome 웹 드라이버 생성
driver = webdriver.Chrome()

# 현재 창의 크기와 위치 확인
print("현재 창의 크기 및 위치:", driver.get_window_position(), driver.get_window_size())

# 브라우저 창을 최대화
driver.maximize_window()

# 최대화된 창의 크기와 위치 확인
print("최대화된 창의 크기 및 위치:", driver.get_window_position(), driver.get_window_size())

# 브라우저 창 닫기
driver.quit()


window_position = {'x': -8, 'y': -8} # x: 왼쪽 상단 모퉁이의 x좌표/ y: 오른쪽 상단 모퉁이의 y좌표
window_size = {'width': 1936, 'height': 1056} # 창의 너비와 높이 정보

# 현재 창의 가로 및 세로 크기를 반으로 나누기
half_width = window_size['width'] // 2
half_height = window_size['height'] // 2

# 결과 출력
print("절반 크기:", half_width, "x", half_height)
```

Bringing HTML (Feat. Web-Scrolling) - 2

A function that retrieves 3-element (title/author/one-line comment) data from the web.

```
# 책 품목에서 제목/작가/한줄평 추출
def extract_product_data(soup):

    product_data = []

    for product in soup.find_all(attrs = {'class':"prod_item"}):
        name_elem = product.find('a', attrs={'class':'prod_info'})
        author_elem = product.find("span", attrs={"class": "prod_author"})
        shortreview_elem = product.find('span', attrs={"class":"review_quotes_text font_size_xxs"})

        if name_elem and author_elem:
            product_data.append({
                'Product': name_elem.text.strip(), # 책의 양쪽 공백제거(데이터의 일관성유지 및 처리과정에서 발생할 수 있는 오류 미연에 방지)
                'Author': author_elem.text.strip(),
                'shortreview': shortreview_elem.text.strip()
            })

    return pd.DataFrame(product_data)

link1 = 'https://product.kyobobook.co.kr/bestseller/total?period=004#?page=1&per=50&period=004&vmw=&bsslBksClstCode=A' # Page 1
link2 = 'https://product.kyobobook.co.kr/bestseller/total?period=004#?page=2&per=50&period=004&vmw=&bsslBksClstCode=A' # Page 2

main_soup1 = web_scroll(link1)
df_main1 = extract_product_data(main_soup1)
main_soup2 = web_scroll(link2)
df_main2 = extract_product_data(main_soup2)
df_features = pd.concat([df_main1, df_main2], ignore_index=True)
#df_features  = df_main1.append(df_main2)
import pandas as pd
directory_loc = './project/book_info(2023).xlsx'
df_features.to_excel(directory_loc, index=False)
```

Name	Type	Size	Value
df_features	DataFrame	(100, 3)	Column names: Product, Author, shortreview
df_main1	DataFrame	(50, 3)	Column names: Product, Author, shortreview
df_main2	DataFrame	(50, 3)	Column names: Product, Author, shortreview
directory_loc	str	30	./project/book_info(2023).xlsx
link1	str	108	https://product.kyobobook.co.kr/bestseller/…
link2	str	108	https://product.kyobobook.co.kr/bestseller/…

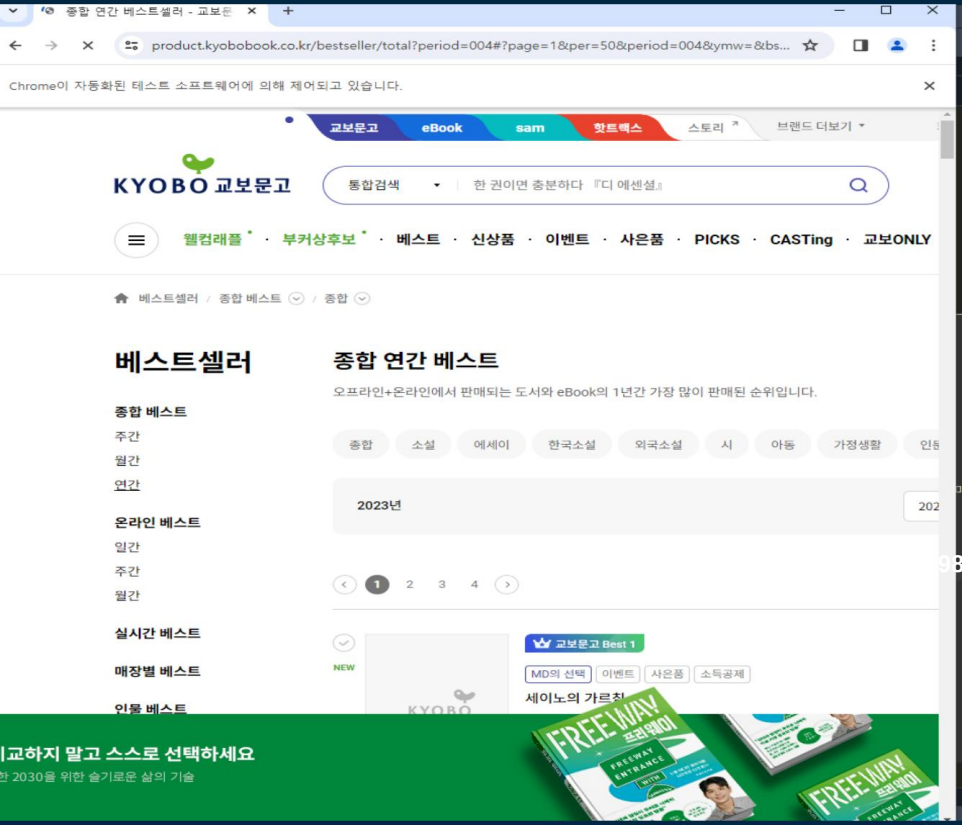
Variable

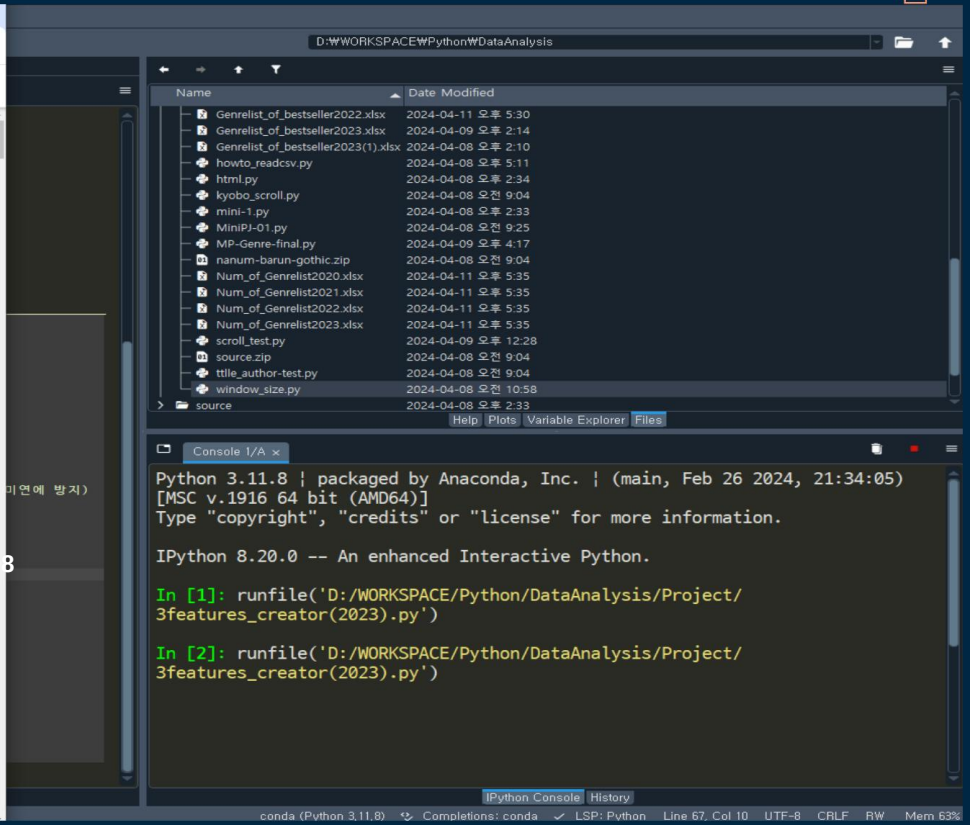
specification: 2 page links by year (select to view from 1 to 50 on the w

Your own file name

Bringing HTML (Feat. Web-Scrolling) - 3

Start scrolling at the web page cutoff size when the function is executed.





Integrating Data

Book_info Excel file by year ÿ Consolidate into one Excel file

```python
import pandas as pd
# 액셀파일 불러오기
def book_info():
    excel_bookinfo = []

    for year in range(2020, 2024):
        excel_bookinfo.append(pd.read_excel(f"./project/book_info({year}).xlsx"))
    return excel_bookinfo
# 'book_rv' [데이터 정제 및 통합]
def book_rv():
    shortreview_by_years = {}
    excel_files = book_info()
    for idx, df in enumerate(excel_files, start=1):
        rv_freq = df['shortreview'].value_counts().to_dict()
        genre_df = pd.DataFrame([rv_freq]) # 딕셔너리 -> dataframe 변환
        # f'{idx}'에서 idx값을 위에 열거 순에 따라 인덱스 값을 1~4까지 반환
        genre_df.rename(index={0: f'{idx}'}, inplace=True)
        shortreview_by_years[f'{idx}'] = genre_df

    return shortreview_by_years


result = book_rv()
```

```python
# 액셀 파일 저장

#result 결과 = {key : 1~4 , value : 4년치 dataframe}
result_list = list(result.values()) # DataFrame인 value값 리스트에 넣기
result_integrate = pd.concat(result_list) # value값들을 합쳐 DataFrame만들기
years = result_integrate.set_index(pd.Index(['2020','2021','2022','2023']), inplace = True) # index 지정

tot_book = [100,100,100,100]
result_integrate['총 권수'] = tot_book

shortreview = result_integrate.rename_axis('연도')
shortreview.rename(columns={'Unnamed: 0':'리뷰'}, inplace=True)

shortreview.to_excel('./project/Review_stat.xlsx', index=True)
```

Result

Apply the same function to all years with variables (web-page x 2/file name)

Create Excel files by year

Sample:

	Product	Author	shortreview
2	더 해빙(The Having)(50만부 기념 리커버 에디션)	이서윤 외 · 수오서재 · 2020.03.01	집중돼요
3	돈의 속성(300쇄 리커버에디션)	김승호 · 스노우폭스북스 · 2020.06.15	집중돼요
4	아몬드	손원평 · 창비 · 2017.03.31	최고예요
5	하버드 상위 1퍼센트의 비밀(리커버 에디션)	정주영 · 한국경제신문 · 2018.10.17	집중돼요

97	이상한 과자 가게 전천당 1	히로시마 레이코 · 길벗스쿨 · 2019.07.05	좋아해요
98	일의 기쁨과 슬픔	장류진 · 창비 · 2019.10.25	최고예요
99	내일의 부 2: 오메가편	김장섭(조던) · 트러스트북스 · 2020.02.20	집중돼요
100	배움의 발견	타라 웨스트오버 · 열린책들 · 2020.01.05	고마워요
101	해커스 토익 스타트 LC Listening (리스닝) 입문서	David Cho · 해커스어학연구소 · 2023.06.12	집중돼요

	A	B	C	D	E	F	G	H	I	J
1	연도	집중돼요	고마워요	좋아해요	최고예요	도움돼요	추천해요	재밌어요	힐링돼요	총 권수
2	2020	55	29	7	6	3				100
3	2021	51	28	8	7	5	1			100
4	2022	44	24	8	9	11	2	1	1	100
5	2023	11	9	6	13	48	2	6	5	100

Result

What is the total for each of the top 5 fields?

연도	소설	시/에세이	경제/경영	자기계발	인문	커린이(초등	외국어	정치/사회	과학	역사/문화	청소년	예술/대중문화	가정/육아	만화	절판	컴퓨터/IT	건강	총 권수
2020	18	17	16	14	13	6	5	3	3	2	1	1	1	nan	nan	nan	nan	100
2021	22	12	23	13	10	6	5	1	2	1	2	1	1	1	nan	nan	nan	100
2022	26	8	15	12	12	8	4	5	3	2	nan	1	1	nan	1	1	1	100
2023	22	5	17	15	14	5	6	2	5	2	1	nan	nan	4	nan	1	1	100

```
# top5로 분야로 dataframe만들기 :
top5 = result_integrate.iloc[:4, :5]
top5.index = range(2020, 2024)

sum_row = top5.sum(axis=0)
sum_row.name = '합계'

# transpose() 활용을 통해 새로운 열이아닌 '합계'행 추가
top5 = pd.concat([top5, pd.DataFrame(sum_row).transpose()], axis=0)
top5 = top5.rename_axis('연도')
top5.to_excel('./Miniproject_2_excel data/top5_Genre.xlsx', index=True)
```

연도	소설	시/에세이	경제/경영	자기계발	인문
2020	18	17	16	14	13
2021	22	12	23	13	10
2022	26	8	15	13	12
2023	22	5	17	15	14
합계	88	42	71	55	49

Target

03

Understanding consumer psychology through reviews

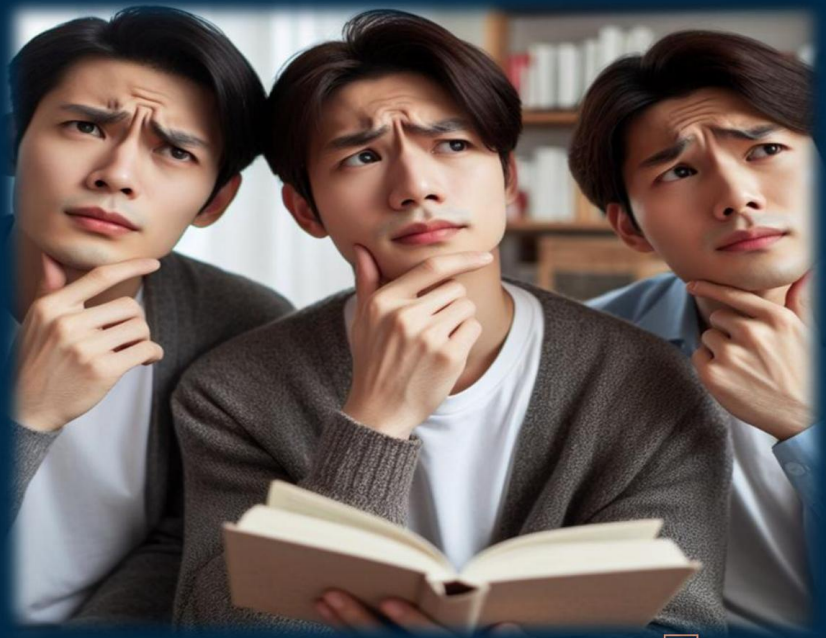
Review ⟺ Genre correlation

Q1. What kind of review is there for what book?

Did you run?

Q2. Relationship between reviews and fields

what?



Integrating Data

Aim: Integrate four years of **data to identify the top three consumer reviews by category.**

```python
def book_review():
    reviews = []  # 4개년 3요소+장르 정보 DataFrame
    for year in range(2020, 2024):
        df_features = pd.read_excel(f"./project/book_info({year}).xlsx")
        df_genre = pd.read_excel(f"./Project/Genrelist_of_bestseller{year}.xlsx")
        review = pd.concat([df_features, df_genre], axis=1)
        reviews.append(review)  # 각각의 합쳐진 데이터프레임 추가해주기

    combined_reviews = pd.concat(reviews, axis=0)   # 행방향 결합
    combined_reviews.reset_index(drop=True, inplace=True)
    return combined_reviews

for_rev = book_review()

# 리뷰로 알아보는 소비자들의 심리
shortreview_freq = for_rev['shortreview'].value_counts().to_dict()
# 소비자가 남긴 top3 comment review
# 1. "집중돼요"
# 2. "고마워요"
# 3. "도움돼요"
```

Dictionary:{ key :comment, value :count}

Key	Type	Size	Value
집중돼요	int	1	161
고마워요	int	1	90
도움돼요	int	1	67
최고예요	int	1	35
좋아해요	int	1	29
재밌어요	int	1	7
힐링돼요	int	1	6
추천해요	int	1	5

Analyzing Data

Data analysis reveals the top three consumer reviews: 1. "Focused." 2. "Thank you." 3. "Helpful."

Top1: What field does “Focus” fall into?

```
shortreview_top1= for_rev[for_rev['shortreview'].str.contains('집중돼요')]
which_genre = shortreview_top1['장르'].value_counts()
df1 = pd.DataFrame(which_genre)
# Pandas 의 df.nlargest(n, columns, keep='first') 활용
# n :출력할 행의 수
# columns : 정렬의 기준이 될 열
# keep : first면 위부터, last면 아래부터, all이면 모두 출력
top3_genre_1 = df1.nlargest(3, 'count')
# 출력할 행의 수 : top3 가장 많이 출현한 장르 , 열 : 'count'
```

Index	Product	Author	shortreview	장르
0	더 해빙(The Having)(50만부 기념 …	이서윤 외 · 수오서재 · 2020.03…	집중돼요	자기계발
1	돈의 속성(300쇄 리커버에디션)	김승호 · 스노우폭스북스 · 2020…	집중돼요	경제/경영
3	하버드 상위 1퍼센트의 비밀(리커버 …	정주영 · 한국경제신문 · 2018.1…	집중돼요	자기계발
4	지적 대화를 위한 넓고 얕은 지식: 제…	채사장 · 웨일북(whalebooks) · …	집중돼요	인문
5	존리의 부자되기 습관(20만부 기념 리…	존 리 · 지식노마드 · 2020.01…	집중돼요	경제/경영

장르	count
경제/경영	49
자기계발	36
인문	33
외국어	13
과학	11
정치/사회	8
역사/문화	4
예술/대중문화	3
가정/육아	3
절판	1

which_genre

Top 3 fields

장르	count
경제/경영	49
자기계발	36
인문	33

Top3_genre_1

Analyzing Data

Novels and poetry/essays overwhelmingly receive **"Thank You"** reviews.

Does this increase the reliability of the hypothesis?

```
# Top2 → '고마워요'
shortreview_top2= for_rev[for_rev['shortreview'].str.contains('고마워요')]
which_genre2 = shortreview_top2['장르'].value_counts()
df2 = pd.DataFrame(which_genre2)
top3_genre_2 = df2.nlargest(3, 'count')
# Top3 → '도움돼요'
shortreview_top3= for_rev[for_rev['shortreview'].str.contains('도움돼요')]
which_genre3 = shortreview_top3['장르'].value_counts()
df3 = pd.DataFrame(which_genre3)
top3_genre_3 = df3.nlargest(3, 'count')
```

What are the top 3 fields that received 'Thank You'?

which_genre2 - Series

장르	count
소설	50
시/에세이	38
만화	2

top3_genre_2 - DataFrame

장르	count
소설	50
시/에세이	38
만화	2

which_genre3 - Series

장르	count
경제/경영	20
자기계발	17
인문	16
외국어	6
역사/문화	3
컴퓨터/IT	2
건강	2
절판	1

top3_genre_3 - DataFrame

장르	count
경제/경영	20
자기계발	17
인문	16

Hypothesis Validation

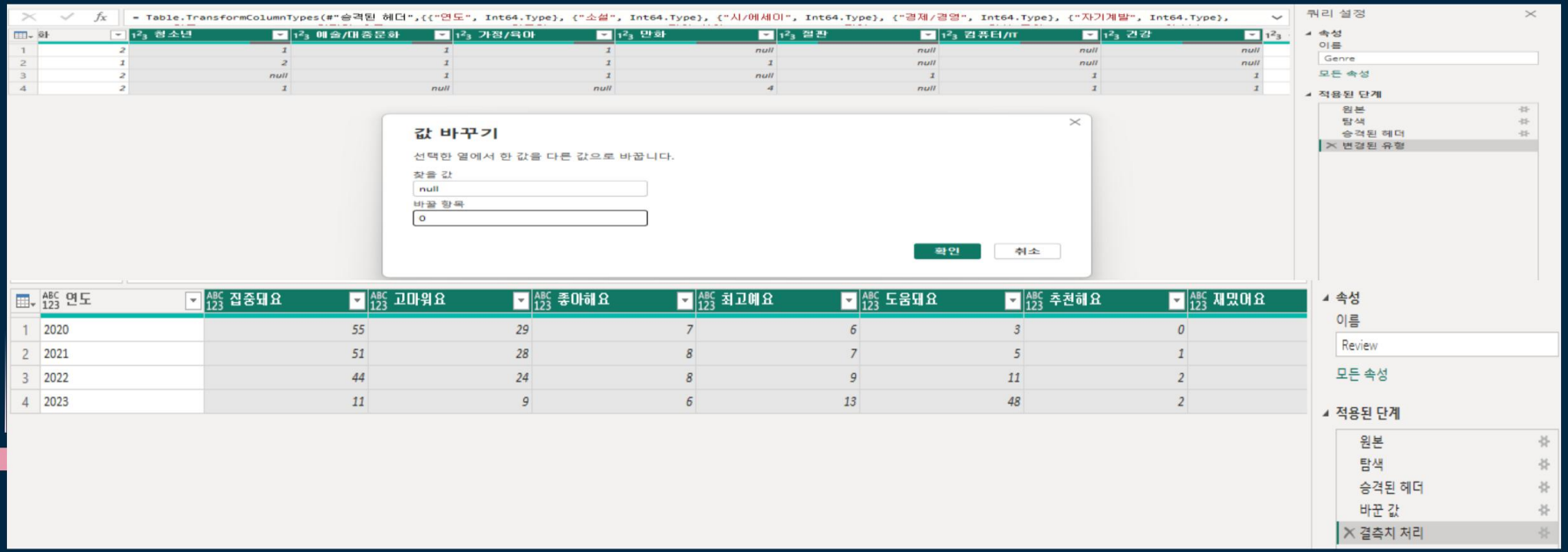
The top genre loved by people during the COVID-19 era: novels. Did people really find comfort in reading novels?

```
# 연도 별 중복 된 도서를 제외하고 어떤 분야가 '고마워요'리뷰를 받았을까?
shortreview_top2 # '고마워요'리뷰를 받은 책 중에 중복된 도서 포함.
dup_deleted = shortreview_top2.drop_duplicates() # 중복제거
fic_or_poet_cnt = dup_deleted[dup_deleted['shortreview'].str.contains('고마워요')]
fic_or_poet_cnt = fic_or_poet_cnt['장르'].value_counts() # 분야 별 차지한 권수
```

장르	count
시/에세이	34
소설	33
만화	2

Data Cleansing via Power-BI

Missing values ÿ treated as 0 / Data type ÿ (random ÿ integer)



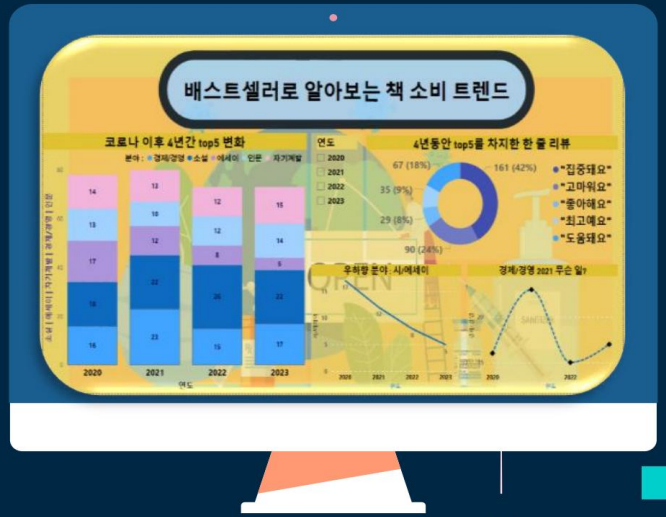
Implication

Before we look at the visualized data,

what are the implications of this mini-project?

- Are bestsellers , an indicator used to estimate consumer sentiment, suitable as analysis data?

- Do reviews of bestsellers truly reflect people's psychology and serve as meaningful indicators for consumers?



Conclusion

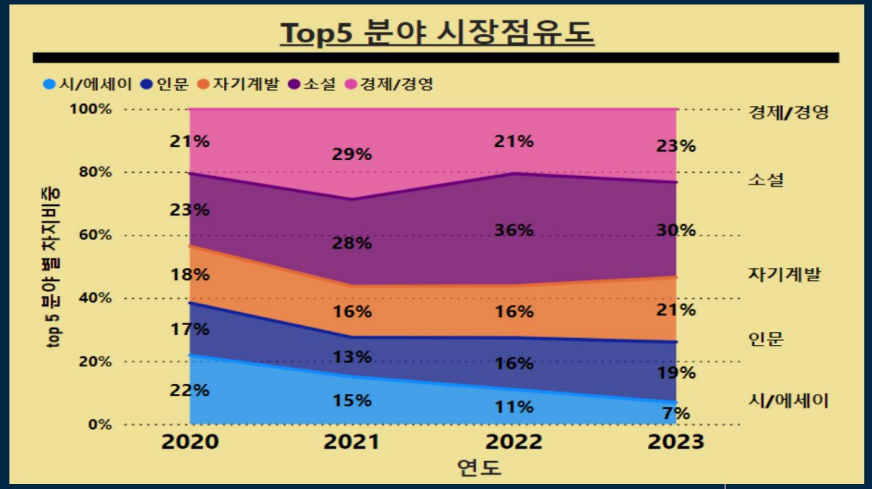
What we learned from data analysis:

1. Some of the hypotheses are correct

(top 1: novel / top 2: economics and management, not self-development)

If only it weren't for 2021

It's a shame my hypothesis was almost right.



Conclusion

2. It ranked 2nd in the previous top 3 review.
"thank you"

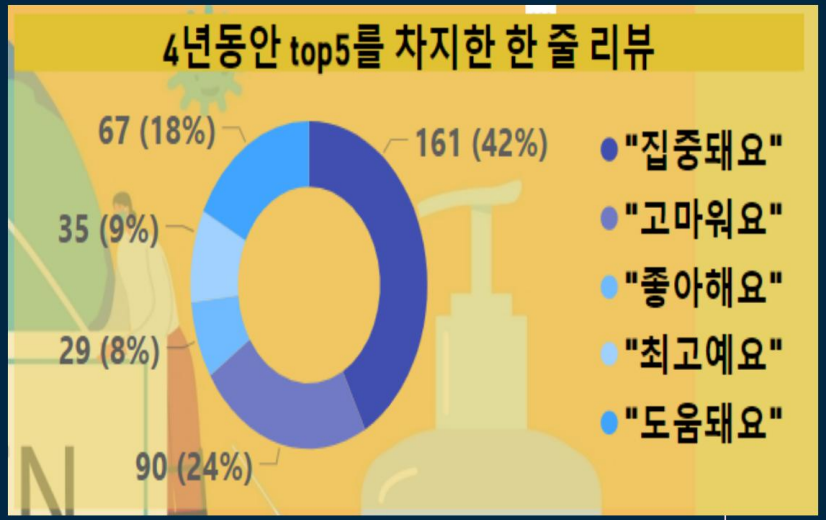
People are talking about novel-related books

I see you left a review saying 'Thank you'

People read novels and find comfort and strength.

I think I got it.

장르	count
시/에세이	34
소설	33
만화	2



Self Reflection [Looking back…]

1. **Python/Power-BI data analysis and visualization takes a lot of time in the collection and refinement stages.**

2. **In the future, the main project will expand the data sample to a country level. I would like to analyze the values and identities of the members of the group.**

Reference

PICTURE CITATION(Google Image/ Copilot-Dalle3) 1.
(Happiness - Google Search, n.d.)

2. Reporter's Note, & Reporter's Note. (2017, November 28). Is all depression the same? Comedy.com

3. BBC News Korea. (April 16, 2020). COVID-19: Self-Quarantine Tips for Those Living Alone. BBC News Korea. https://www.bbc.com/korean/international-5228624

4. How To Make YouTube Shorts For Your Business (thevideoanimationcompany.com)

ARTICLE CITATION

5. A Study on COVID-19 Depression Using Big Data (nd). https://www.kci.go.kr/kciportal/landing/article.kci?arti_id=ART002979085

6. Depression left by COVID-19... Depression and anxiety disorders surge among children under 18 (May 12, 2023). Young Doctor. https://www.docdocdoc.co.kr/news/articleView.html?idxno=3005764

7. ÿWhosaeng Shinboÿ 8.99 million patients receiving treatment for depression and anxiety disorders, 42.3% increase among people in their 20s since COVID-19 (October 4, 2022). Whosaeng Shinbo. https://www.whosaeng.com/139197

8. Bitcoin surpasses $60,000... for the first time since November 2021 (February 28, 2024). Hankyung Global Market. https://www.hankyung.com/article/202402287620i

9. Impact of rising real estate prices... National net worth increased by 11.4% in 2021 (July 21, 2022). Nongman Newspaper https://www.nongmin.com/article/20220721359649

Do you have any questions?

THANKS