```
import requests
from bs4 import BeautifulSoup
from tadm import tadm
import re
from collections import Counter
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt
base_url = "https://github.com/PacktPublishing?page={}&tab=r
headers = {"User-Agent": "Mozilla/5.0"}
all_data = []
for page in tqdm(range(1, 298)): # страницы 1-297
   url = base_url.format(page)
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, "html.parser")
    repos = soup.select("li[itemprop='owns']")
    for repo in repos:
        title_tag = repo.select_one("a[itemprop='name codeRe
        if not title_tag:
           continue
       title = title_tag.text.strip()
       href = "https://github.com" + title_tag['href']
       stars_tag = repo.select_one("a[href$='/stargazers']'
       stars = stars_tag.text.strip() if stars_tag else "0'
       all_data.append((title, href, stars))
    100% | 297/297 [00:32<00:00, 9.07it/s]
all_words = []
for title, _, _ in all_data:
   words = re.findall(r'\b\\w+\b', title.lower())
    all_words.extend(words)
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
import time
options = Options()
options.add_argument("--headless")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")
driver = webdriver.Chrome(options=options)
# Загружаем страницу
url = "https://github.com/PacktPublishing?page=1&tab=reposito"
```

ValueError

ValueError



다음 오류를 설명해 주세요.

ValueError: We need at least 1





### Hypothesis

The error "ValueError: We need at least 1 word to plot a word cloud, got 0" arises because the filtered\_words list is empty, resulting in an empty string being passed to the

WordCloud.generate() method. This happens because either:

- The initial word extraction from the web scraping process failed to retrieve any words.
- 2. The filtering process using stopwords removed all the extracted words, leaving filtered\_words empty.

Since the global variables show that filtered\_words is an empty list, the issue is likely due to the filtering process removing all the words from the list.

## Suggested Changes

# Adding words to stopwords

You might need to add more words to your stopwords set. These could include common words specific to the dataset that don't provide meaningful insights, such as

driver.get(url)

```
time.sleep(3)
soup = BeautifulSoup(driver.page_source, "html.parser")
# Новый селектор: ищем ссылки на ре
repos = soup.select("a[href^='/PacktPublishing/']")
print(" 🔍 Найдено репозиториев:", len(repo
for a in repos[:5]:
    title = a.text.strip()
   href = "https://github.com" + a['href']
    print("-", title, "\rightarrow", href)
driver.quit()
🚁 🔍 Найдено репозиториев: 59

    Advanced-Deep-Learning-with-Keras → https://github.c

     - 1.9k → https://github.com/PacktPublishing/Advanced-D
     - 983 → <a href="https://github.com/PacktPublishing/Advanced-De">https://github.com/PacktPublishing/Advanced-De</a>
     - How-to-Build-an-LLVM-Backend-by-example → https://gi
     - → <a href="https://github.com/PacktPublishing/How-to-Build-a">https://github.com/PacktPublishing/How-to-Build-a</a>
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
import time
import pandas as pd
from tqdm import tqdm
options = Options()
options.add_argument("--headless")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")
driver = webdriver.Chrome(options=options)
data = []
for page in tqdm(range(1, 298)): # Страницы 1~297
    url = f"https://github.com/PacktPublishing?page={page}&t
    driver.get(url)
    time.sleep(2.5) # обязательно, иначе э
    soup = BeautifulSoup(driver.page_source, "html.parser")
    repos = soup.select("a[href^='/PacktPublishing/']")
    for a in repos:
        title = a.text.strip()
        href = "https://github.com" + a['href']
        if title: # пропускаем пустые сс
            data.append((title, href))
driver.guit()
```

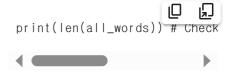
"packt" or "publishing". Here's how you can update the stopwords set in your code:

```
stopwords = set(["and", "with
filtered_words = [word for wo
```

Use code with caution

# Check for errors in web scraping:

Also, make sure that the web scraping is working as expected.
You can inspect the contents of the 'all\_words' list before filtering to see if it contains words from the website.



#### Use code with caution

If the all\_words list is empty or does not contain relevant words, there might be an issue with the web scraping logic itself.

### Example

If the scraping and filtering are working correctly but the word cloud still doesn't work, here's an example of how you can create a word cloud with some sample data if filtered\_words is empty:

```
from IPython import get_ipyth
from IPython.display import d
import requests
from bs4 import BeautifulSoup
from tqdm import tqdm
import re
from collections import Count
import pandas as pd
```

```
# 📊 Сохраним в таблицу
df = pd.DataFrame(data, columns=["Title", "URL"])
df.to_excel("packt_repos.xlsx", index=False)
print("✓ Готово! Данные сохранены в
     100% | 297/297 [15:59<00:00, 3.23s/it]
     ✓ Готово! Данные сохранены в 'р.
import re
from collections import Counter
all words = []
for title in df['Title']:
   words = re.findall(r'₩b₩w+₩b', title.lower()) # per
   all_words.extend(words)
top10 = Counter(all_words).most_common(10)
print(" ♣ Топ-10 слов:", top10)
→ Топ-10 слов: [('0', 5310), ('with', 885), ('
from wordcloud import WordCloud
import matplotlib.pyplot as plt
stopwords = set([
   "and", "with", "the", "for", "from", "using", "in", "of'
   "a", "an", "at", "as", "is", "this", "that", "you", "you
1)
filtered = [w for w in all_words if w not in stopwords]
text = " ".join(filtered)
wc = WordCloud(width=800, height=400, background_color='whit
plt.figure(figsize=(12, 6))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

```
from wordcloud import WordClo
import matplotlib.pyplot as p

# ... (your existing code) ..

text = " ".join(filtered_word

# Check if filtered_words is
if not filtered_words:
    print("filtered_words is
    text = "example sample da

wc = WordCloud(width=800, hei
# ... (rest of your code) ...
```

#### Use code with caution

#### ▶ 출처

답변평가 🖒 尔





```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
import pandas as pd
import time
from tgdm import tgdm
options = Options()
options.add_argument("--headless")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")
driver = webdriver.Chrome(options=options)
data = []
for page in tqdm(range(1, 298)): # 1~297
    url = f"https://github.com/PacktPublishing?page={page}&tal
    driver.get(url)
    time.sleep(2.5)
    soup = BeautifulSoup(driver.page_source, "html.parser")
    repos = soup.select("li[itemprop='owns']")
    for repo in repos:
        title_tag = repo.select_one("a[itemprop='name codeRepo
        if not title_tag:
            continue
        title = title_tag.text.strip()
        href = "https://github.com" + title_tag['href']
        star_tag = repo.select_one("a[href$='/stargazers']")
        stars = star_tag.text.strip() if star_tag else "0"
        data.append((title, href, stars))
```

driver.quit()

# ■ Сохраняем в Excel

df = pd.DataFrame(data, columns=["Title", "URL", "Stars"])

df.to\_excel("packt\_books.xlsx", index=False)

print(" Данные сохранены в packt\_books.xlsx

100% 1297/297 [15:55<00:00, 3.22s/it] Д

여기에 메시지를 입력하세요.



0/2000

답변에 Google의 관점을 반영하지 않는 부정확하 거나 불쾌감을 주는 정보가 표시될 수 있습니다. <u>자세히 알아보기</u>