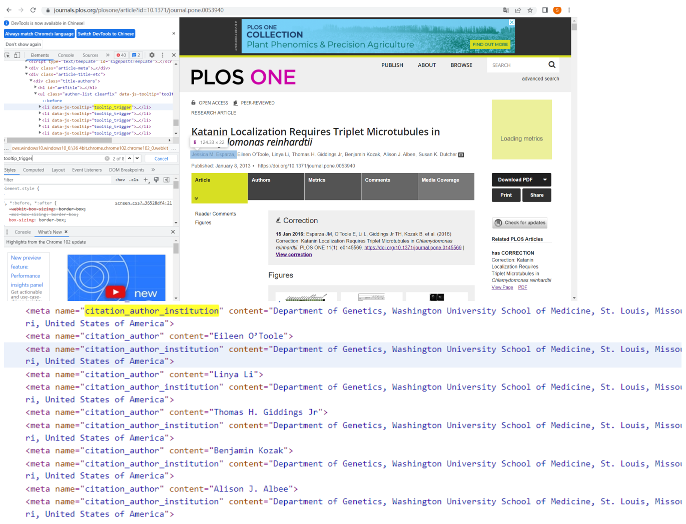
web scraper

1. Affiliation university & lab size

1.1 PLOS papers

In the original data set, the first 48 papers can be viewed on PLOS(https://journals.plos.org/plosone/), so we observed the introduction of the author information on PLoS. Since it contains the school information of the first author and other staff members of this paper, we scrape these information on PLOS.

We found that as long as we input the DOI string of paper in the URL, we can jump to the article page. Therefore, after designing the URL and looking for the label name according to the page content, we used beautiful soup to scrape.



```
<meta name="citation_author_institution" content="Department of Genetics, Washington University School of Medicine, St. Louis, Missouri, United States of America">
<meta name="citation_author" content="Eileen O’Toole">
<meta name="citation_author_institution" content="Department of Genetics, Washington University School of Medicine, St. Louis, Missouri, United States of America">
<meta name="citation_author" content="Linya Li">
<meta name="citation_author_institution" content="Department of Genetics, Washington University School of Medicine, St. Louis, Missouri, United States of America">
<meta name="citation_author" content="Thomas H. Giddings Jr">
<meta name="citation_author_institution" content="Department of Genetics, Washington University School of Medicine, St. Louis, Missouri, United States of America">
<meta name="citation_author" content="Benjamin Kozak">
<meta name="citation_author_institution" content="Department of Genetics, Washington University School of Medicine, St. Louis, Missouri, United States of America">
<meta name="citation_author" content="Alison J. Albee">
<meta name="citation_author_institution" content="Department of Genetics, Washington University School of Medicine, St. Louis, Missouri, United States of America">
```

Therefore, we wrote function get_affiliation_labsize to crawl the affiliation University and lab size in PLOS.

```python
def get_affiliation_labsize(PLOS_url,head):#get PLOS papers' affiliation university and lab size
    affiliation_uni=[]
    labsize=[]
    for i in PLOS_url:
        url="https://journals.plos.org/plosone/article?id="+i
        content=requests.get(url,headers=head,timeout=30)
        text=content.text
        bs=BeautifulSoup(text,'html.parser')
        lab=len(bs.find_all("li",{"data-js-tooltip":"tooltip_trigger"}))
        affiliation=bs.find_all('meta',
{'name':'citation_author_institution'})[0]['content']
        labsize.append(lab)
        affiliation_uni.append(affiliation)
    return affiliation_uni,labsize
```

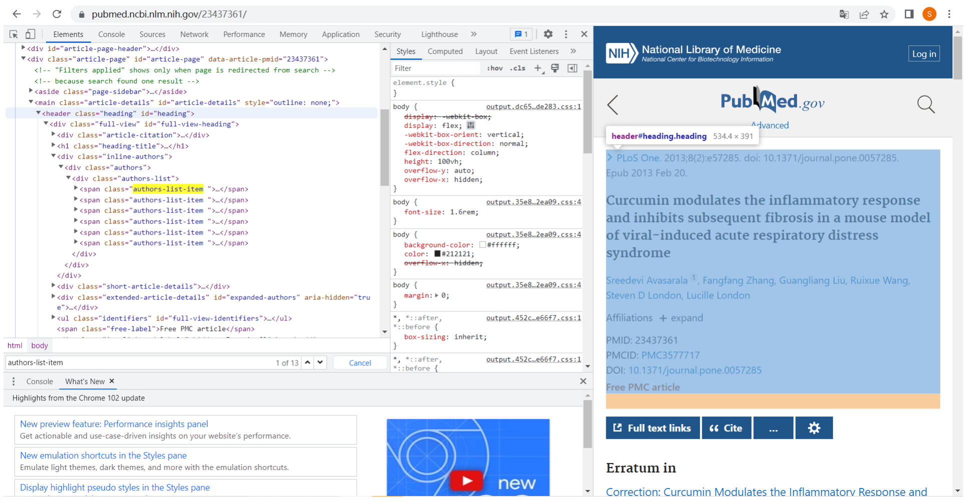
1.2 Pubmed papers

Also, we notice that other 166 papers' affiliation univerisity and lab size can be found on pubmed(https://pubmed.ncbi.nlm.nih.gov/). First, we also observe the URL and find that the search result of the article can be obtained by adding the name of the article to the search URL. But there are many results here. We need to find the search result that contains the author information. So we wrote get_med_link function, used to obtain the article search result link so that we can query the author information from the link.

> ```
> <button class="share-search-result trigger result-action-trigger share-dialog-trigger" aria-haspopup="true" data-twitter-url="http://twitter.com/intent/tweet?text=Curcumin%20modulates%20the%20inflammatory%20response%20and%20inhibits%20subsequent%20fibrosis%20in%20a%20mouse%20model%20of%20viral-induced%20acu%E2%80%A6%20https%3A//pubmed.ncbi.nlm.nih.gov/23437361/" data-facebook-url="http://www.facebook.com/sharer/sharer.php?u=https%3A//pubmed.ncbi.nlm.nih.gov/23437361/" data-permalink-url="https://pubmed.ncbi.nlm.nih.gov/23437361/" aria-expanded="false"> Share </button>
> ```

```
def get_med_link(title,head):#get paper links in pubmed
    links = []
    for i in title:
        url="https://pubmed.ncbi.nlm.nih.gov/?term=" + i.replace(" ","+")
        content=requests.get(url,headers=head,timeout=30)
        text=content.text
        bs=BeautifulSoup(text,'html.parser')
        link = bs.find("button", {"class":"share-search-result trigger
result-action-trigger share-dialog-trigger"})
        if link == None:
            value = url
        else:
            value = link["data-permalink-url"]
        links.append(value)
    return links
```

Then, by accessing links, we can crawl to the author's relevant information through beautiful soup. To do this, we wrote the get_affiliation_university_med function.


pubmed.ncbi.nlm.nih.gov/23437361/

```
▸<div id="article-page-header">…</div>
▾<div class="article-page" id="article-page" data-article-pmid="23437361">
   <!-- "Filters applied" shows only when page is redirected from search -->
   <!-- because search found one result -->
  ▸<aside class="page-sidebar">…</aside>
  ▾<main class="article-details" id="article-details" style="outline: none;">
    ▾<header class="heading" id="heading">
      ▾<div class="full-view" id="full-view-heading">
        ▸<div class="article-citation">…</div>
        ▸<h1 class="heading-title">…</h1>
        ▾<div class="inline-authors">
          ▾<div class="authors">
            ▾<div class="authors-list">
              ▸<span class="authors-list-item ">…</span>
              ▸<span class="authors-list-item ">…</span>
              ▸<span class="authors-list-item ">…</span>
              ▸<span class="authors-list-item ">…</span>
              ▸<span class="authors-list-item ">…</span>
              ▸<span class="authors-list-item ">…</span>
             </div>
           </div>
         </div>
        ▸<div class="short-article-details">…</div>
        ▸<div class="extended-article-details" id="expanded-authors" aria-hidden="true">…</div>
        ▸<ul class="identifiers" id="full-view-identifiers">…</ul>
         <span class="free-label">Free PMC article</span>
```

authors-list-item 1 of 13

Highlights from the Chrome 102 update

New preview feature: Performance insights panel
Get actionable and use-case-driven insights on your website's performance.

New emulation shortcuts in the Styles pane
Emulate light themes, dark themes, and more with the emulation shortcuts.

Display highlight pseudo styles in the Styles pane

National Library of Medicine
National Center for Biotechnology Information

header#heading.heading 534.4 × 391

PLoS One. 2013;8(2):e57285. doi: 10.1371/journal.pone.0057285. Epub 2013 Feb 20.

Curcumin modulates the inflammatory response and inhibits subsequent fibrosis in a mouse model of viral-induced acute respiratory distress syndrome

Sreedevi Avasarala [1], Fangfang Zhang, Guangliang Liu, Ruixue Wang, Steven D London, Lucille London

Affiliations + expand

PMID: 23437361
PMCID: PMC3577717
DOI: 10.1371/journal.pone.0057285
Free PMC article

Erratum in

Correction: Curcumin Modulates the Inflammatory Response and

```
def get_affiliation_labsize_med(links,head):
    affiliation_uni=[]
    labsize=[]
    for i in links:
        content=requests.get(i,headers=head,timeout=30)
        text=content.text
```

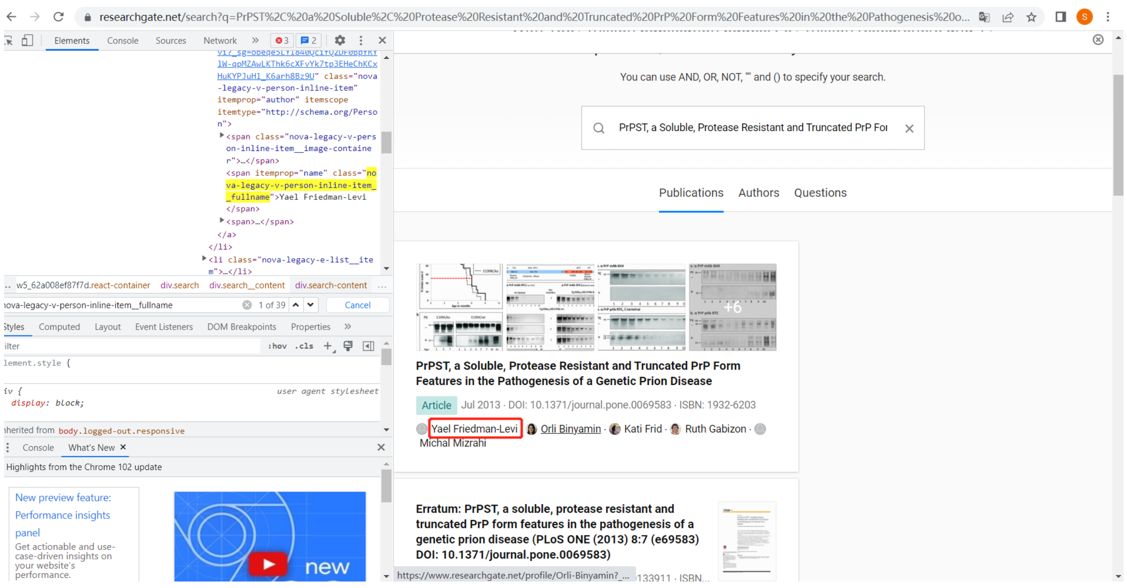
```
        bs=BeautifulSoup(text,'html.parser')
        lab=len(bs.find_all('span',{'class':'authors-list-item'}))
        labsize.append(lab)
        affiliations=bs.find('a',{'class':'affiliation-link'})
        if affiliations != None:
            affiliation=affiliations.get('title')
        affiliation_uni.append(affiliation)
    return affiliation_uni,labsize
```

2. Highest degree and Degree area

In order to crawl the highest degree and degree area, we found relevant information on the research gate(https://www.researchgate.net/). When we try to scrape by directly accessing the URL, we find that the research gate will hinder our scraper. In order to avoid being monitored by the anti crawler mechanism, we chose to use selenium library to simulate the process of accessing websites using chrome.

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
import time
```

We first define a Driver as an object that simulates a Chrome browser. By input the fixed URL + paper's title, we can get the result page of the paper(the image below). On the returned page, Driver will click the first author(in red box) and then it will jump to the author information page.


researchgate.net/search?q=PrPST%2C%20a%20Soluble%2C%20Protease%20Resistant%20and%20Truncated%20PrP%20Form%20Features%20in%20the%20Pathogenesis%20o...

```
1W-qpMZAwLKThk6cXFvYk7tp3EHeChKCx
HuKYPJuHl_K6arh8Bz9U" class="nova
-legacy-v-person-inline-item"
itemprop="author" itemscope
itemtype="http://schema.org/Perso
n">
  <span class="nova-legacy-v-pers
  on-inline-item__image-containe
  r">…</span>
  <span itemprop="name" class="no
  va-legacy-v-person-inline-item_
  _fullname">Yael Friedman-Levi
  </span>
  <span>…</span>
</a>
</li>
<li class="nova-legacy-e-list__ite
m">…</li>
```

… w5_62a008ef87f7d.react-container div.search div.search__content div.search-content

nova-legacy-v-person-inline-item__fullname 1 of 39 Cancel

```
element.style {
}
div {                                   user agent stylesheet
  display: block;
}
Inherited from body.logged-out.responsive
```

Highlights from the Chrome 102 update

New preview feature: Performance insights panel

Get actionable and use-case-driven insights on your website's performance.

You can use AND, OR, NOT, "" and () to specify your search.

PrPST, a Soluble, Protease Resistant and Truncated PrP For

Publications Authors Questions

PrPST, a Soluble, Protease Resistant and Truncated PrP Form Features in the Pathogenesis of a Genetic Prion Disease

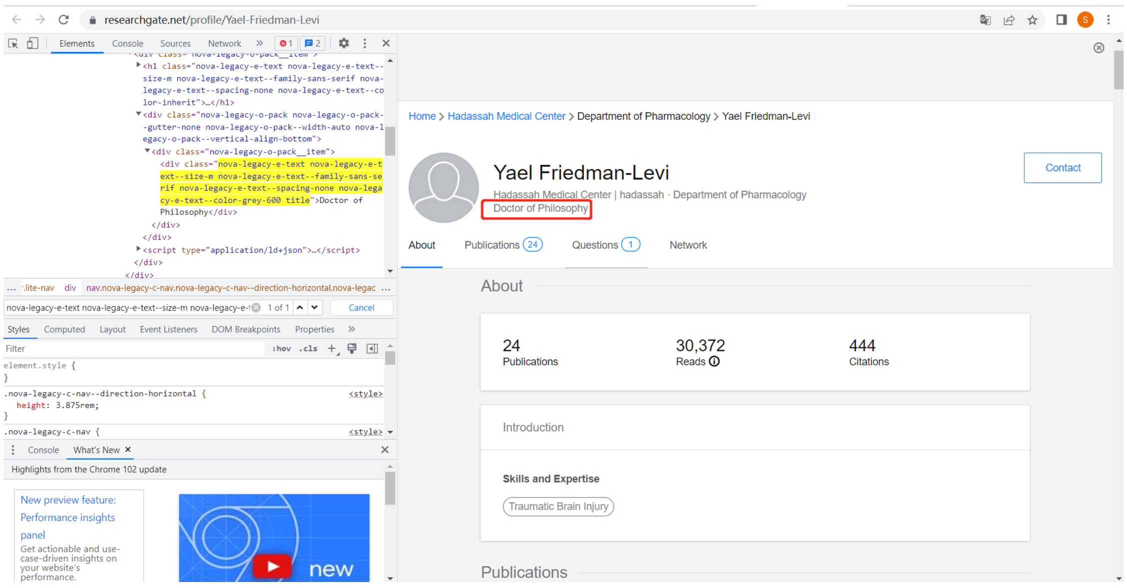
Article Jul 2013 · DOI: 10.1371/journal.pone.0069583 · ISBN: 1932-6203

Yael Friedman-Levi · Orli Binyamin · Kati Frid · Ruth Gabizon · Michal Mizrahi

Erratum: PrPST, a soluble, protease resistant and truncated PrP form features in the pathogenesis of a genetic prion disease (PLoS ONE (2013) 8:7 (e69583) DOI: 10.1371/journal.pone.0069583)

https://www.researchgate.net/profile/Orli-Binyamin?_...

From the author information page, we can scrape the highest degree and the degree area by the content in the red box. After get content from this label, we still need to do some preparation such as divide the degree and area. After that, we finish scraping the information of author's highest degree and degree area.



researchgate.net/profile/Yael-Friedman-Levi

```
<div class="nova-legacy-o-pack__item">
  <h1 class="nova-legacy-e-text nova-legacy-e-text--size-m nova-legacy-e-text--family-sans-serif nova-legacy-e-text--spacing-none nova-legacy-e-text--color-inherit">…</h1>
  <div class="nova-legacy-o-pack nova-legacy-o-pack--gutter-none nova-legacy-o-pack--width-auto nova-legacy-o-pack--vertical-align-bottom">
    <div class="nova-legacy-o-pack__item">
      <div class="nova-legacy-e-text nova-legacy-e-text--size-m nova-legacy-e-text--family-sans-serif nova-legacy-e-text--spacing-none nova-legacy-e-text--color-grey-600 title">Doctor of Philosophy</div>
    </div>
  </div>
  <script type="application/ld+json">…</script>
</div>
</div>
```

```
… :.lite-nav  div  nav.nova-legacy-c-nav.nova-legacy-c-nav--direction-horizontal.nova-legac …
nova-legacy-e-text nova-legacy-e-text--size-m nova-legacy-e-t   1 of 1   Cancel

element.style {
}
.nova-legacy-c-nav--direction-horizontal {                        <style>
  height: 3.875rem;
}
.nova-legacy-c-nav {                                              <style>
```

Highlights from the Chrome 102 update

New preview feature: Performance insights panel
Get actionable and use-case-driven insights on your website's performance.

Home > Hadassah Medical Center > Department of Pharmacology > Yael Friedman-Levi

Yael Friedman-Levi
Hadassah Medical Center | hadassah · Department of Pharmacology
Doctor of Philosophy

About | Publications 24 | Questions 1 | Network

About

24 Publications | 30,372 Reads | 444 Citations

Introduction

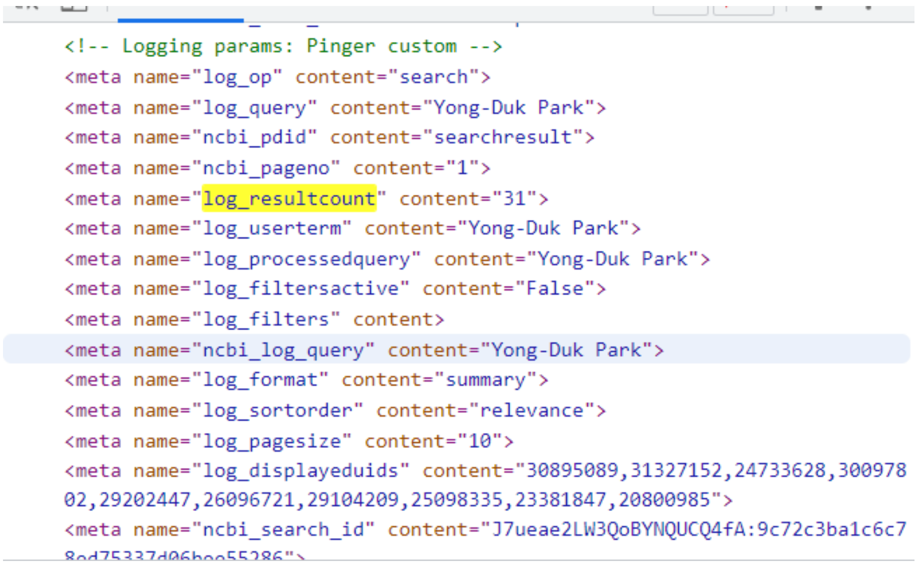
Skills and Expertise
Traumatic Brain Injury

Publications

```
def get_degree_area(titles):
    degree_list=[]
    area_list=[]
    driver = webdriver.Chrome()
    driver.implicitly_wait(1)
    for i in titles:
        curr_page = 'https://www.researchgate.net/search/publication?q='+i
        print(i)
        driver.get(curr_page)
        driver.implicitly_wait(1)
        driver.find_element(By.CLASS_NAME, "nova-legacy-v-person-inline-
item__fullname").find_element(By.XPATH, "./..").click()#Simulate browser
action and click the first author
        driver.implicitly_wait(1)
        resp=driver.page_source
        bs_test=BeautifulSoup(resp,'html.parser')
        if bs_test.find_all('h5',{'class':'nova-legacy-e-text nova-legacy-e-
text--size-m nova-legacy-e-text--family-sans-serif nova-legacy-e-text--
spacing-s nova-legacy-e-text--color-grey-400'})!=[]:
            degree_list.append('N/A')
            area_list.append('N/A')
        else:
            try:
                degree=bs_test.find_all('div',{'class':'nova-legacy-e-text
nova-legacy-e-text--size-m nova-legacy-e-text--family-sans-serif nova-
legacy-e-text--spacing-none nova-legacy-e-text--color-grey-600 title'})
[0].contents
                if type(degree)==list:
                    degree=degree[0].split(',')[0]
                if 'of' in degree:
                    area=degree[degree.index('of')+3:]
                else:
                    if 'in' in degree:
                        area=degree[degree.index('in')+3:]
                    else:
                        area='N/A'
            except Exception as e:
                degree='N/A'
                area='N/A'
            degree_list.append(degree)
            area_list.append(area)
    return degree_list,area_list
```

3. **Other journals & Duration of Career & Publication Rate**

In order to crawl the papers published by authors and calculate the publishing rate, we crawl the author information on Pubmed. The reason for using Pubmed instead of PLOS is that PubMed has the author information of 214 papers. We can still get the author's search results page by directly accessing the fixed url+ author's name. By querying the labels, we can get the number of papers published by the author.



```
<!-- Logging params: Pinger custom -->
<meta name="log_op" content="search">
<meta name="log_query" content="Yong-Duk Park">
<meta name="ncbi_pdid" content="searchresult">
<meta name="ncbi_pageno" content="1">
<meta name="log_resultcount" content="31">
<meta name="log_userterm" content="Yong-Duk Park">
<meta name="log_processedquery" content="Yong-Duk Park">
<meta name="log_filtersactive" content="False">
<meta name="log_filters" content>
<meta name="ncbi_log_query" content="Yong-Duk Park">
<meta name="log_format" content="summary">
<meta name="log_sortorder" content="relevance">
<meta name="log_pagesize" content="10">
<meta name="log_displayeduids" content="30895089,31327152,24733628,300978
02,29202447,26096721,29104209,25098335,23381847,20800985">
<meta name="ncbi_search_id" content="J7ueae2LW3QoBYNQUCQ4fA:9c72c3ba1c6c7
8ed75337d06bee55286">
```

Then calculate the author's duration career and publication rate through the following formulas.

$$duration\ career = int(year\ of\ latest\ publication - year\ of\ first\ publication)$$

$$publication\ rate = \frac{total\ journal\ number}{dutation\ career}$$

Because the website has the problem of inconsistent format, we have also added the classification of different page formats and the handling of abnormal error reports to the function.

```
def get_journalNumber_duration_publicationRate(author_list):
    journal_number_list=[]
    duration_list=[]
    rate_list=[]
    for i in author_list:
        print(i)
        url="https://pubmed.ncbi.nlm.nih.gov/?term="+i
        content=requests.get(url,headers=head,timeout=30)
        text=content.text
        bs=BeautifulSoup(text,'html.parser')
        if bs.find_all('div',{'class': 'results-amount'})==[]:
            duration_list.append('N/A')
            rate_list.append('N/A')
            journal_number_list.append(1)
        else:
            if bs.find_all('div',{'class': 'results-amount'})[0].contents==
['\n  \n    No results were found.\n  \n']:
                journal_number_list.append(1)
                duration_list.append('N/A')
                rate_list.append('N/A')
            else:
                journal_number=bs.find_all('meta',{'name':
'log_resultcount'})[0]['content']
                journal_number=int(journal_number)
                pub_time=[]
                if journal_number>300:
                    journal_number=300#
                for j in range(1,int(journal_number/10)+2):
                    url_temp=url+'&page='+str(j)

 content_temp=requests.get(url_temp,headers=head,timeout=30)
                    text_temp=content_temp.text
                    bs_temp=BeautifulSoup(text_temp,'html.parser')
                    for x in bs.find_all('span',{'class':'docsum-journal-
citation short-journal-citation'}):
                        try:
                            pub_time.append(int(x.contents[0][-5:-1]))
                        except Exception as e:
                            pass
                duration=max(pub_time)-min(pub_time)
                duration_list.append(duration)
                if duration==0:
                    publication_rate=1
                else:
                    publication_rate=journal_number/duration
                rate_list.append(publication_rate)
                journal_number_list.append(journal_number)
    return journal_number_list,duration_list,rate_list
```