MEMO SHEET FOR AUDIENCE

"貫　　通　　全　　球"?

DE MACAU PARA O RESTO DO MUNDO
FROM MACAO TO THE REST OF THE WORLD

An analysis of Macau's international postal service based on postcard data from Postcrossing.com(R) retrieved with a BeautifulSoup-based crawler

24 November, 2022

An analysis of Macau's international postal service

based on postcard data from Postcrossing.com(R)

retrieved with a BeautifulSoup-based crawler

1- <u>Introduction. Speaker: Garfield (Ke Siyun DB927871)</u>

Background

-Does the postal service of our Post Office really reach every corner of the world efficiently?

How to find patterns for mails without tracking number?

Focus> **EXPORT mails from Macau.**

Data source> **Postcrossing.com** (A website to send postcards to temporary penpals)

Questions to explore

a. Questions related to Postcrossing itself:

Whether there are **local** postcard exchanges existed on Postcrossing? (Solved: No)

b. Further questions on Macau Postal System:

How was the effect of pandemic on **time** of shipping (e.g. longer shipping time)?

Does the postal service's **stability** vary during the COVID years?

2- <u>Implement a Beautiful Soup-based crawler Speaker: Rupert (Zhou Jicheng DB927793)</u>

The package used: Beautiful Soup (not using scrapy, for accessibility in Jupyter Notebook)

Crawler access the formatted URLs

Used css selectors and regular express to split and save data.

Designed to save interim json result and use other functions to combine (to avoid lost connections)

Concurrent processing to save time.

Data (in Json/ calculate with Python-Dictionarys):

Format: "XXXXX":["travelled_date", "distance", "date", "destination"]

3- **<u>Data processing Speaker: Keith (Jiang Chaoyu DB927938)</u>**

CLEAN DATA

-gather interim json files to calculate

-clear "N/A" data (lost postcards)

EMPERICAL ANALYSIS

-determine the date to explore (11240 postcard before/after Feb, 2022)

-Show the destination change:

-calculate average shipping time

-check the stability (std error)

VISULATION

- Destination, shipping time, std error
-

4- <u>Conclusion Speaker: Garfield (Ke Siyun DB927871)</u>

Summary: Facts and reasonings-

Limitation

In-accurate data/ doubtful assumption

Furfure study

Improve our study

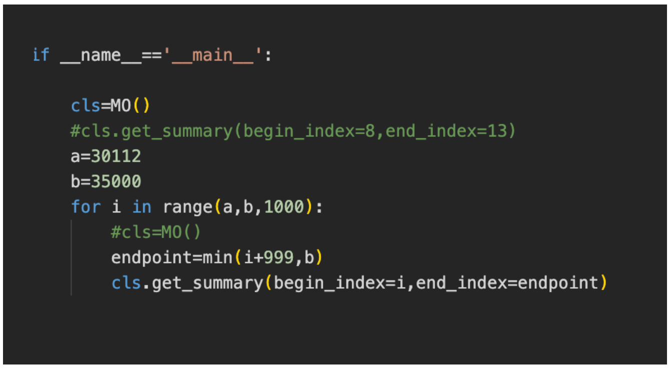
More questions on Postcrossing website

NLP processing on user profiles.

**

Ref codes

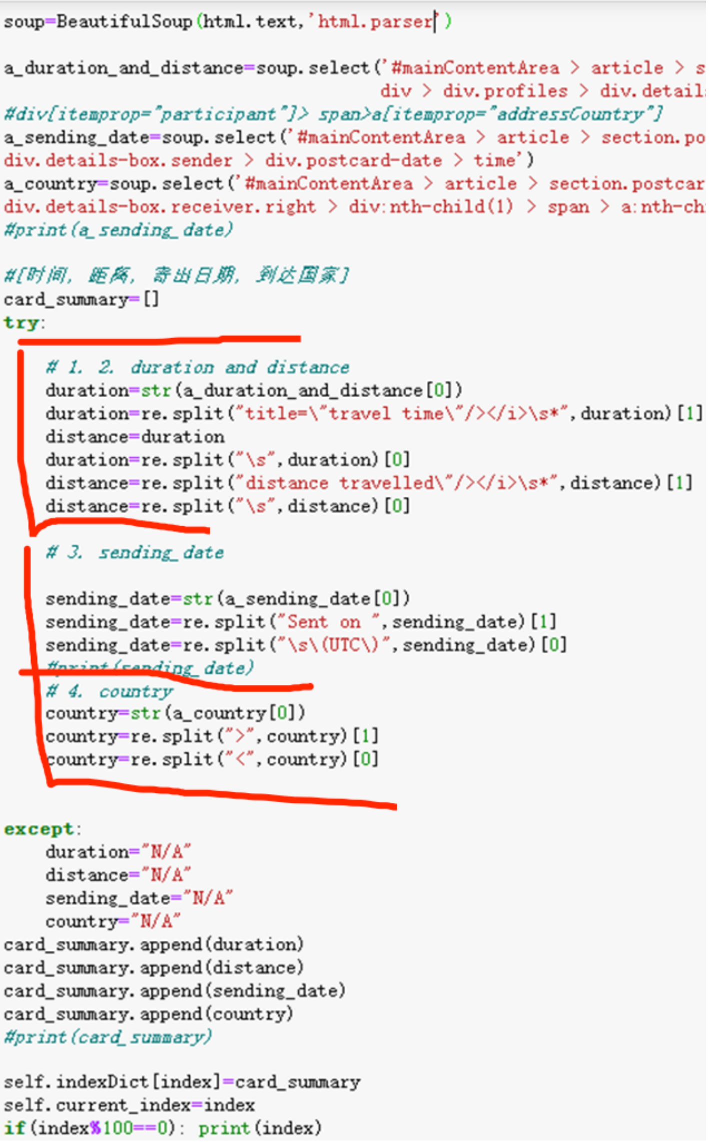
//use the spider



```
if __name__=='__main__':

    cls=MO()
    #cls.get_summary(begin_index=8,end_index=13)
    a=30112
    b=35000
    for i in range(a,b,1000):
        #cls=MO()
        endpoint=min(i+999,b)
        cls.get_summary(begin_index=i,end_index=endpoint)
```

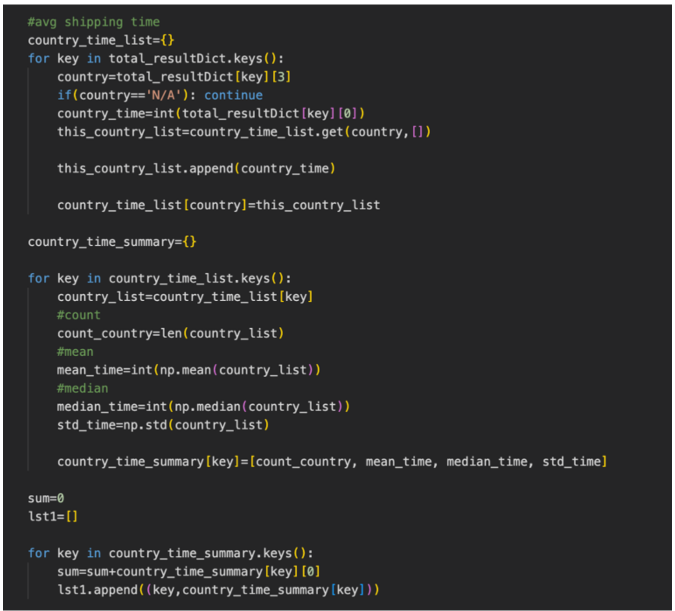
```
//save result when connection lost

json_indexDict = json.dumps(cls.indexDict)

with

open('data/interimResult_'+str(cls.begin_index)+"_"+str(cls.current_index)+'.json', 'w') as json_file1:

    json_file1.write(json_indexDict)

    json_file1.close()
```

// design of spider to find info

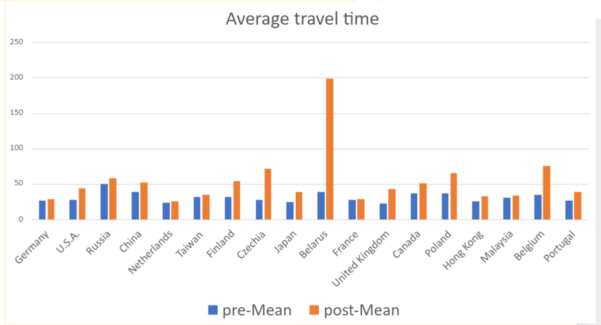


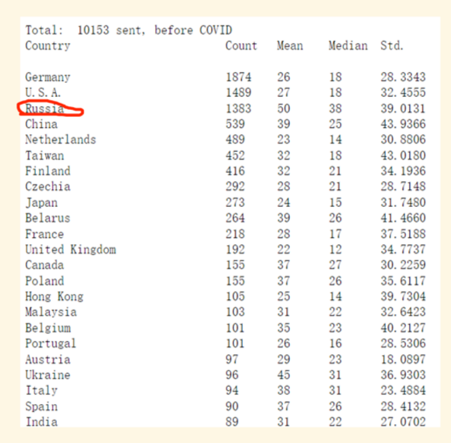
```
soup=BeautifulSoup(html.text,'html.parser')

a_duration_and_distance=soup.select('#mainContentArea > article > s
                                     div > div.profiles > div.detail
#div[itemprop="participant"]> span>a[itemprop="addressCountry"]
a_sending_date=soup.select('#mainContentArea > article > section.po
div.details-box.sender > div.postcard-date > time')
a_country=soup.select('#mainContentArea > article > section.postcar
div.details-box.receiver.right > div:nth-child(1) > span > a:nth-ch
#print(a_sending_date)

#[时间, 距离, 寄出日期, 到达国家]
card_summary=[]
try:

    # 1. 2. duration and distance
    duration=str(a_duration_and_distance[0])
    duration=re.split("title=\"travel time\"/></i>\s*",duration)[1]
    distance=duration
    duration=re.split("\s",duration)[0]
    distance=re.split("distance travelled\"/></i>\s*",distance)[1]
    distance=re.split("\s",distance)[0]

    # 3. sending_date

    sending_date=str(a_sending_date[0])
    sending_date=re.split("Sent on ",sending_date)[1]
    sending_date=re.split("\s\(UTC\)",sending_date)[0]
    #print(sending_date)
    # 4. country
    country=str(a_country[0])
    country=re.split(">",country)[1]
    country=re.split("<",country)[0]


except:
    duration="N/A"
    distance="N/A"
    sending_date="N/A"
    country="N/A"
card_summary.append(duration)
card_summary.append(distance)
card_summary.append(sending_date)
card_summary.append(country)
#print(card_summary)

self.indexDict[index]=card_summary
self.current_index=index
if(index%100==0): print(index)
```

//Calculate mean_time from JSON saved

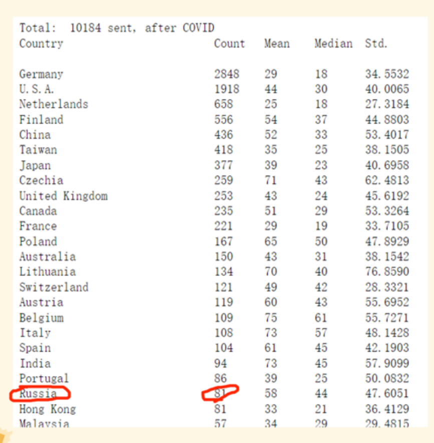


```
#avg shipping time
country_time_list={}
for key in total_resultDict.keys():
    country=total_resultDict[key][3]
    if(country=='N/A'): continue
    country_time=int(total_resultDict[key][0])
    this_country_list=country_time_list.get(country,[])

    this_country_list.append(country_time)

    country_time_list[country]=this_country_list

country_time_summary={}

for key in country_time_list.keys():
    country_list=country_time_list[key]
    #count
    count_country=len(country_list)
    #mean
    mean_time=int(np.mean(country_list))
    #median
    median_time=int(np.median(country_list))
    std_time=np.std(country_list)

    country_time_summary[key]=[count_country, mean_time, median_time, std_time]

sum=0
lst1=[]

for key in country_time_summary.keys():
    sum=sum+country_time_summary[key][0]
    lst1.append((key,country_time_summary[key]))
```





```
Total:  10153 sent, before COVID
Country                    Count   Mean    Median  Std.

Germany                    1874    26      18      28.3343
U.S.A.                     1489    27      18      32.4555
Russia                     1383    50      38      39.0131
China                      539     39      25      43.9366
Netherlands                489     23      14      30.8806
Taiwan                     452     32      18      43.0180
Finland                    416     32      21      34.1936
Czechia                    292     28      21      28.7148
Japan                      273     24      15      31.7480
Belarus                    264     39      26      41.4660
France                     218     28      17      37.5188
United Kingdom             192     22      12      34.7737
Canada                     155     37      27      30.2259
Poland                     155     37      26      35.6117
Hong Kong                  105     25      14      39.7304
Malaysia                   103     31      22      32.6423
Belgium                    101     35      23      40.2127
Portugal                   101     26      16      28.5306
Austria                    97      29      23      18.0897
Ukraine                    96      45      31      36.9303
Italy                      94      38      31      23.4884
Spain                      90      37      26      28.4132
India                      89      31      22      27.0702
```



```
Total:  10184 sent, after COVID
Country                    Count   Mean    Median  Std.

Germany                    2848    29      18      34.5532
U.S.A.                     1918    44      30      40.0065
Netherlands                658     25      18      27.3184
Finland                    556     54      37      44.8803
China                      436     52      33      53.4017
Taiwan                     418     35      25      38.1505
Japan                      377     39      23      40.6958
Czechia                    259     71      43      62.4813
United Kingdom             253     43      24      45.6192
Canada                     235     51      29      53.3264
France                     221     29      19      33.7105
Poland                     167     65      50      47.8929
Australia                  150     43      31      38.1542
Lithuania                  134     70      40      76.8590
Switzerland                121     49      42      28.3321
Austria                    119     60      43      55.6952
Belgium                    109     75      61      55.7271
Italy                      108     73      57      48.1428
Spain                      104     61      45      42.1903
India                      94      73      45      57.9099
Portugal                   86      39      25      50.0832
Russia                     81      58      44      47.6051
Hong Kong                  81      33      21      36.4129
Malaysia                   57      34      29      29.4815
```