Post answers come from

December 12, 2022

0.1 Find the posts which the answers origin from

0.1.1 Description of code

As Stack Exchange (SE) saves all information on tags in the posed question (and not in the answer) I need to get all the questions which the answers are answering to calculate measures as the share of feminine users in a tag.

[1]:
```
## Import packages and make path to file

import random
import pandas as pd
import matplotlib.pyplot as plt
import math
import time # to see how long it takes to run different parts of code
import linecache
import re


file = 'D:/Data/Posts.xml'
```

[6]:
```
#Import a list of users which is missing the answers (from R file)
missing_posts = pd.read_csv("D:/Data/PostId_Answers.csv")

# Make this into a list of post ids
list_of_users = missing_posts["x"].values.tolist()
```

0.2 Set up to make loop to extract the posts and comments

First - we need to see how many lines there are in total

[6]:
```
## Only run first time (very time-consuming):

## Get number of lines to make a random sample:

with open(file, 'r', encoding='UTF-8') as f:
    num_lines = sum(1 for line in f)
    print('Total lines:', num_lines)
```

```
Total lines: 55513871
```

[7]:
```
# Else - make an abject which holds the total number of lines (for re-use of the␣
 ↪script)

num_lines = 55513871
```

Total number of lines is : 55513871

[8]:
```
# Make a list with all usernames in, in the same format as in the big lines␣
 ↪(Parsing)


list_of_search_strings = []

for i in range(len(list_of_users)):
        list_of_search_strings.append(str('<row Id="' + str(list_of_users[i]) +␣
 ↪'"'))
```

0.3 Make a loop that extracts the posts and comments from the random subset

[9]:
```
# Make a list of numbers equalling fractions of the data. This is to enable one␣
 ↪to see how "far" we are getting

perc = num_lines/100
list_of_numbers=[]

for i in range(1,101):

    list_of_numbers.append(round(i*perc))
```

[10]:
```
start = time.time()

# Make a regex-string, which contains all of the user-numbers randomly drawn
temp = '(?:% s)' % '|'.join(list_of_search_strings)

# Make a list which can contain all of the posts
list_of_posts = []



# Make i and b == 0, which will be used to count during the loop
i = 0
b = 0

# Open the file and read each line
with open(file, 'r', encoding = 'UTF-8') as f:
    for line in range(num_lines):
        i = i+1
```

```
        z = f.readline()
        if  re.search(temp, z):
                list_of_posts.append(z)
        if i in list_of_numbers:
            b = b+1
            print(b)

end = time.time()
print((end - start)/60)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

```
87
88
89
90
91
92
93
94
95
96
97
98
99
100
1671.3798207998275
```

Startede 20:06

[16]:
```
list_of_posts[5]
```

[16]:
```
'  <row Id="1505" PostTypeId="1" CreationDate="2008-08-04T19:02:36.777"
Score="31" ViewCount="8405" Body="&lt;p&gt;How do I set the icon that appears
proper on the &lt;code&gt;iPhone&lt;/code&gt; for the websites I have
created?&lt;/p&gt;&#xA;" OwnerUserId="83" LastEditorUserId="14152908"
LastEditorDisplayName="Zack" LastEditDate="2020-09-09T09:53:42.277"
LastActivityDate="2021-11-18T20:34:49.467" Title="How do I give my websites an
icon for iPhone?"
Tags="&lt;html&gt;&lt;ios&gt;&lt;iphone&gt;&lt;favicon&gt;&lt;apple-touch-
icon&gt;" AnswerCount="1" CommentCount="1" FavoriteCount="3"
ClosedDate="2021-11-18T21:34:13.877" ContentLicense="CC BY-SA 4.0" />\n'
```

[11]:
```
## Parse the dataframe

start = time.time()

Id = []
PostTypeId = []
ParentID = []
AcceptedAnswerId = []
CreationDate = []
Score = []
ViewCount = []
Body = []
OwnerUserId = []
LastEditorUserId = []
LastEditorDisplayName = []
LastEditDate = []
CommunityOwnedDate = []
```

```
ClosedDate = []
Title = []
Tags = []
AnswerCount = []
CommentCount = []
FavoriteCount = []

for i in range(len(list_of_posts)):
    if "w Id=" in list_of_posts[i]:
        z = list_of_posts[i].split('Id="')[1].split('"')[0]
        Id.append(z)
    else:
        Id.append("missing")
    if "PostTypeId=" in list_of_posts[i]:
        z = list_of_posts[i].split('PostTypeId="')[1].split('"')[0]
        PostTypeId.append(z)
    else:
        PostTypeId.append("missing")
    if '" ParentId="' in list_of_posts[i]:
        z = list_of_posts[i].split('" ParentId="')[1].split('"')[0]
        ParentID.append(z)
    else:
        ParentID.append("missing")
    if '" AcceptedAnswerId=' in list_of_posts[i]:
        z = list_of_posts[i].split('AcceptedAnswerId="')[1].split('"')[0]
        AcceptedAnswerId.append(z)
    else:
        AcceptedAnswerId.append("missing")
    if "CreationDate=" in list_of_posts[i]:
        z = list_of_posts[i].split('CreationDate="')[1].split('"')[0]
        CreationDate.append(z)
    else:
        CreationDate.append("missing=")
    if "Score=" in list_of_posts[i]:
        z = list_of_posts[i].split('Score="')[1].split('"')[0]
        Score.append(z)
    else:
        Score.append("missing")
    if "ViewCount=" in list_of_posts[i]:
        z = list_of_posts[i].split('ViewCount="')[1].split('"')[0]
        ViewCount.append(z)
    else:
        ViewCount.append("missing")
    if "Body=" in list_of_posts[i]:
        z = list_of_posts[i].split('Body="')[1].split('"')[0]
        Body.append(z)
    else:
```

```
        Body.append("missing")
    if "OwnerUserId=" in list_of_posts[i]:
        z = list_of_posts[i].split('OwnerUserId="')[1].split('"')[0]
        OwnerUserId.append(z)
    else:
        OwnerUserId.append("missing")
    if "LastEditorUserId=" in list_of_posts[i]:
        z = list_of_posts[i].split('LastEditorUserId="')[1].split('"')[0]
        LastEditorUserId.append(z)
    else:
        LastEditorUserId.append("missing")
    if "LastEditorDisplayName=" in list_of_posts[i]:
        z = list_of_posts[i].split('LastEditorDisplayName="')[1].split('"')[0]
        LastEditorDisplayName.append(z)
    else:
        LastEditorDisplayName.append("missing")
    if "LastEditDate=" in list_of_posts[i]:
        z = list_of_posts[i].split('LastEditDate="')[1].split('"')[0]
        LastEditDate.append(z)
    else:
        LastEditDate.append("missing")
    if "CommunityOwnedDate=" in list_of_posts[i]:
        z = list_of_posts[i].split('CommunityOwnedDate="')[1].split('"')[0]
        CommunityOwnedDate.append(z)
    else:
        CommunityOwnedDate.append("missing")
    if "ClosedDate=" in list_of_posts[i]:
        z = list_of_posts[i].split('ClosedDate="')[1].split('"')[0]
        ClosedDate.append(z)
    else:
        ClosedDate.append("missing")
    if '" Tags="' in list_of_posts[i]:
        z = list_of_posts[i].split('" Tags="')[1].split('"')[0]
        Tags.append(z)
    else:
        Tags.append("missing")
    if "AnswerCount=" in list_of_posts[i]:
        z = list_of_posts[i].split('AnswerCount="')[1].split('"')[0]
        AnswerCount.append(z)
    else:
        AnswerCount.append("missing")
    if "CommentCount=" in list_of_posts[i]:
        z = list_of_posts[i].split('CommentCount="')[1].split('"')[0]
        CommentCount.append(z)
    else:
        CommentCount.append("missing")
    if '" FavoriteCount=' in list_of_posts[i]:
```

```
        z = list_of_posts[i].split('FavoriteCount="')[1].split('"')[0]
        FavoriteCount.append(z)
    else:
        FavoriteCount.append("missing")
    if '" Title="' in list_of_posts[i]:
        z = list_of_posts[i].split('Title="')[1].split('"')[0]
        Title.append(z)
    else:
        Title.append("missing")



df = pd.DataFrame(Id)

df['PostTypeId'] = PostTypeId
df['ParentID'] = ParentID
df['AcceptedAnswerId'] = AcceptedAnswerId
df['CreationDate'] = CreationDate
df['Score'] = Score
df['ViewCount'] = ViewCount
df['Body'] = Body
df['OwnerUserId'] = OwnerUserId
df['LastEditorUserId'] = LastEditorUserId
df['LastEditorDisplayName'] = LastEditorDisplayName
df['LastEditDate'] = LastEditDate
df['CommunityOwnedDate'] = CommunityOwnedDate
df['ClosedDate'] = ClosedDate
df['Title'] = Title
df['Tags'] = Tags
df['AnswerCount'] = AnswerCount
df['CommentCount'] = CommentCount
df['FavoriteCount'] = FavoriteCount


end = time.time()
print((end - start)/60)
```

```
0.2224652091662089
```

[12]:
```
df.to_csv("D:/Data/posts_answers.csv")
```