TURBOLAB TECHNOLOGIES

DATA QUALITY ANALYST

WRITTEN TEST

Candidate Name : SWAROOP N C

Candidate email : ncswaroop1997@gmail.com

Date : 09-05-2022

Instructions: 3 hours to complete

- All three questions should be attempted.
- The solution should be in a file named answer_3.py.
- For question 1, save the created series/dataframe along with executed notebook (or result screenshots)
- For question 2 and 3, the solution should be in a file named answer_.py.
- Output can be submitted by sending the file(s) to aparna.p@scrapehero.com.

```
import numpy as np
import pandas as pd
```

QUESTION 1

Part 01. Create a Pandas series with 100 random dates as it falls between 01-01-20 to 01-01-21

(hint: use Pandas date_range function).

Like:- _ 2020-01-06 2020-06-11 2020-02-18

Part 02. Dedupe it and calculate the number of duplicates and convert it to percentage.

Then by using regex, filter values where either the month is 02,05,09 OR the date is 01,04,07 - the apply function should not be used.

Finally, calculate the percentage of values filtered for the month condition and the date condition.

SOLUTION TO Q 1

Part 01. Create a Pandas series with 100 random dates as it falls between 01-01-20 to 01-01-21 (hint: use Pandas date_range function).

Like:- _ 2020-01-06 2020-06-11 2020-02-18

```
def random_dates2(start, end, n, unit='D', seed=None):
    if not seed:
        np.random.seed(0)
    ndays = (end - start).days + 1
    return start + pd.to_timedelta(np.random.randint(0, ndays, n), unit=unit)
```

```
start = pd.to_datetime('2020/01/01')
end = pd.to_datetime('2021/01/01')
n=100
```

```
dates=random_dates2(start,end,n)
dates
```

```
DatetimeIndex(['2020-06-21', '2020-02-17', '2020-04-27', '2020-07-11',
               '2020-11-19', '2020-09-08', '2020-07-14', '2020-12-25',
               '2020-01-10', '2020-07-30', '2020-10-04', '2020-08-30',
               '2020-10-19', '2020-03-28', '2020-03-11', '2020-03-29',
               '2020-11-10', '2020-07-12', '2020-02-09', '2020-03-28',
               '2020-06-23', '2020-03-29', '2020-12-03', '2020-06-14',
               '2020-01-26', '2020-11-29', '2020-03-13', '2020-09-22',
               '2020-04-25', '2020-08-31', '2020-07-16', '2020-12-01',
               '2020-12-04', '2020-04-09', '2020-06-26', '2020-08-31',
               '2020-10-12', '2020-05-27', '2020-05-27', '2020-10-15',
               '2020-09-22', '2020-07-04', '2020-05-07', '2020-02-02',
               '2020-02-01', '2020-07-21', '2020-09-01', '2020-05-31',
               '2020-06-12', '2020-07-02', '2020-01-29', '2020-10-17',
               '2020-05-08', '2020-05-08', '2020-02-23', '2020-02-08',
               '2020-09-01', '2020-09-30', '2020-12-01', '2020-04-15',
               '2020-02-12', '2020-02-01', '2020-09-14', '2020-11-17',
               '2020-02-27', '2020-10-18', '2020-12-24', '2020-04-29',
               '2020-09-24', '2020-03-23', '2020-04-01', '2020-04-09',
               '2020-02-23', '2020-05-01', '2020-03-25', '2020-07-22',
               '2020-11-20', '2020-09-19', '2020-02-17', '2020-05-07',
               '2020-05-11', '2020-12-22', '2020-06-29', '2020-11-30',
               '2020-05-23', '2020-05-28', '2020-08-15', '2020-10-06',
               '2020-07-26', '2020-12-07', '2020-02-18', '2020-11-01',
               '2020-03-10', '2020-06-18', '2020-06-12', '2020-04-05',
               '2020-07-16', '2020-04-04', '2020-09-13', '2020-06-27'],
              dtype='datetime64[ns]', freq=None)
```

```
dateseries=pd.Series(dates)
dateseries
```

```
0    2020-06-21
1    2020-02-17
2    2020-04-27
3    2020-07-11
4    2020-11-19
        ...
95   2020-04-05
```

```
96   2020-07-16
97   2020-04-04
98   2020-09-13
99   2020-06-27
Length: 100, dtype: datetime64[ns]
```

Part 02. Dedupe it and calculate the number of duplicates and convert it to percentage.

Then by using regex, filter values where either the month is 02,05,09 OR the date is 01,04,07 - the apply function should not be used.

Finally, calculate the percentage of values filtered for the month condition and the date condition.

```
dateseries.describe()[:2]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Treati
  """Entry point for launching an IPython kernel.
count     100
unique     85
dtype: object
```

calculate the number of duplicates and convert it to percentage.

from above it is clear that 85 are unique and rest 15 are duplicates

```
NumofDup = n - dateseries.describe()[1]
NumofDup
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Treati
  """Entry point for launching an IPython kernel.
15
```

```
NumofDupPerc = (NumofDup/n)*100
NumofDupPerc
```

```
15.0
```

Then by using regex, filter values where either the month is 02,05,09 OR the date is 01,04,07 - the apply function should not be used.

Finally, calculate the percentage of values filtered for the month condition and the date condition.

```
#not answered
```

Question 2:

This question has two parts,

a) Sentence Validator and

b) Name Reducer

a) Create a sentence validator function. The validator should return input if it is valid otherwise return False

Validation Criteria:

1. Start letter must be an uppercase letter and it should follow either a lowercase letter or a single whitespace.
2. All letters in the sentence except the start letter must be in lowercase.
3. The last character (aka terminal character) of the sentence must be any of the following:

. (dot) ? (question mark) ! (exclamation mark)

4. Words must be separated with a single whitespace.

If there is a hyphen between any two words then there should be one whitespace before and after that hyphen.

eg: Lab - 1 is valid, but Lab- 7 and Lab -7 are invalid

b) Write a reducer function to clean the sentences. Reducer takes output from the validator function as input and performs the following cleaning steps,

○ Removes terminal characters (see above validation criteria for list of allowed terminal characters)

○ Removes all duplicated word groups (see below examples) but keep its first occurence

○ Removes all leading and trailing whitespaces and hyphens

After completing the functions for part a & b.

Create a function check_and_clean which takes a sentence as input and validates (using validator function) and returns reduced string (output from reducer function) when the sentence is valid, "" otherwise.

Note: You are not allowed to use regex for this question.

Example 1: Input: Melo diagnostics melo Labs

Sentence is invalid, failed validation criteria 2 & 3

Output: invalid

Example 2:

Input: Melo diagnostics - southpark east 29th street - southpark east 29th street - free drug testing not offered. Sentence is valid

Word groups: Melo diagnostics southpark east 29th street free drug testing not offered

Output: Melo diagnostics - southpark east 29th street - free drug testing not offered

Example 3:

Input: Simple labs - covid test available - west side hospital - covid test available!

Sentence is valid

Word groups: Simple labs covid test available west side hospital

Output: Simple labs - covid test available - west side hospital

SOLUTION TO Q 2

sentence validator

Validation Criteria:

1 **Start letter must be an uppercase letter** and **it should follow either a lowercase letter** or **a single whitespace**.

2 **All letters in the sentence except the start letter must be in lowercase.**

3 **The last character (aka terminal character) of the sentence must be any of the following: . (dot) ? (question mark) ! (exclamation mark)**

Words must be separated with a single whitespace. If there is a hyphen between any two words then there should be one whitespace before and after that hyphen.

eg: Lab - 1 is valid, but Lab- 7 and Lab -7 are invalid

```
def SentenceValidator(string):
  length = len(string)
  #Start letter must be an uppercase letter
  if (string[0] < 'A' or string[0] > 'Z'):
    return False
  #last character of sentence must be  . (dot) ? (question mark) ! (exclamation mark)
  elif not(string[length-1] == '.' or string[length-1] == '?' or string[length-1] == '!'):
    return False
  #All letters in the sentence except the start letter must be in lowercase
  else:
    for ele in string[1:]:
      if ele.isupper():
        return False
  #Words must be separated with a single whitespace.
  #check for more than 2 consecutive whitespace
  for i in range(len(string)) :
    if (string[i]==' ' and string[i+1]==' ') :
      return False
  #If there is a hyphen between any two words then
  #there should be one whitespace before and after that hyphen.
  for i in range(len(string)) :
    if (string[i]=='-') :
      if not((string[i-1]==' ')and (string[i+1]==' ')):
        return False
  return string
```

b) Write a reducer function to clean the sentences. Reducer takes output from the validator function as input and performs the following cleaning steps,

○ **Removes terminal characters (see above validation criteria for list of allowed terminal characters)**

○ **Removes all duplicated word groups (see below examples) but keep its first occurence**

○ **Removes all leading and trailing whitespaces** and hyphens

```
def·reducer(ValidOp)·:
··#Removes·terminal·characters
··ValidOp=ValidOp[:-1]
··#Removes·all·leading·and·trailing·whitespaces
··ValidOp=ValidOp.strip()
··#Removes·all·leading·and·trailing·HYPHENS
··ValidOp=ValidOp.strip("-")
··#Removes·all·duplicated·word·groups··but·keep·its·first·occurence
··l·=·ValidOp.split()
··k·=·[]
··for·i·in·l:
····if·(ValidOp.count(i)>=1·and·(i·not·in·k)·or·i·=='-'):
······k.append(i)
··ValidOp='·'.join(k)
··return·ValidOp
```

check_and_clean

Create a function check_and_clean which takes a sentence as input and validates (using validator function) and returns reduced string (output from reducer function) when the sentence is valid, "" otherwise.

```
def check_and_clean() :
  stringcheck=str(input("Input String : "))
  if (SentenceValidator(stringcheck) != False ) :
    print("******** OUTPUT ***********************")
    return (reducer(SentenceValidator(stringcheck)))
  else :
    print("******** OUTPUT **********************")

#driverprogram
#test·case·1·Melo·diagnostics·melo·Labs
check_and_clean()
```

```
Input String : Melo diagnostics melo Labs
******** OUTPUT **********************
'<invalid>'
```

```
#driverprogram
#test case 3  : Simple labs - covid test available - west side hospital - covid test avail
check_and_clean()
```

```
Input String : Simple labs - covid test available - west side hospital - covid test a
******** OUTPUT **********************
'Simple labs - covid test available - west side hospital -'
```

Question 3:

A sample data of posts of random users are given in this link: click here to download Post_id, date of post and post caption details are available in the sample dataset.

Create a function that extracts posts older than 13/11/2021 and finds the 3 most frequently used special characters out of it. The function should return the 3 most frequently used special characters and the number of times they occurred in the filtered data.

Example

post_id	date	caption
post #1	11/11/2021	@bla bl@ bla! 23🔥
post #2	15/11/2021	Foo b@r foOB!a🙌
post #3	12/11/2021	🔥 aerrt!! Qwe r rr$
post #4	13/11/2021	@momo bati$t@🔥

Output

[(@, 4), (!, 3), (🔥, 2)]

Explanation The post #2 is eliminated since it is older than the date 13/11/2021. In the remaining 3 rows, the special character “@” occurred the most i.e, 4 times in the posts #1 and #4. The second most frequent special character is “!” which occurred 3 times and then “�” occurred twice. Since only the top 3 most frequent ones are required, the remaining special characters “�” and “$” are ignored

```
#load dataset
df = pd.read_csv("Captions.csv")
df
```

	post_id	date	caption
0	post #1	15/11/2021	NaN
1	post #2	14/11/2021	Skippers at the ground for a photo call but ca...
2	post #3	18/11/2021	Kia ora everyone, We tried but it wasn't to be...
3	post #4	11/11/2021	Teihorangi Walden 🔒 2022
4	post #5	13/11/2021	NaN
...	...	...	...
95	post #96	17/11/2021	Pink maomao, blue maomao, granddaddy hāpuku (N...
96	post #97	12/11/2021	Hiking in Queenstown? Yes, please! 🥾 Now th...
97	post #98	11/11/2021	Opportunity to engage: Gender and Sex Diverse ...
98	post #99	11/11/2021	7 years signed, sealed and delivered ❤️💙
99	post #100	16/11/2021	🎉GIVEAWAY🎉 True heroes often go unnoticed. Our...

100 rows × 3 columns

extracting posts older than 13/11/2021

```
df=df[df['date']>'13/11/2021']
df
```

0	post #1	15/11/2021	NaN
1	post #2	14/11/2021	Skippers at the ground for a photo call but ca...
2	post #3	18/11/2021	Kia ora everyone, We tried but it wasn't to be...
7	post #8	17/11/2021	Last time in QT 📍🔙 Lock in the 11th of Februar...
9	post #10	17/11/2021	It has been a homecoming for Inspector Darren ...
11	post #12	16/11/2021	Vantage Cambridge 3 Day update 📢 To assist wit...
12	post #13	15/11/2021	6 days to go! #StandUpWithYourNix #ALeagueMen...
15	post #16	17/11/2021	Final week focus 👀 #FRAvNZL #LoveEveryMinute
17	post #18	16/11/2021	Free kick benders ✅ Stunning curlers ✅ Smashin...
21	post #22	15/11/2021	Who was your favourite Silver Fern of 2021? Vo...
24	post #25	17/11/2021	Tip 1: Celebrating the little wins 🥳 Over the ...
25	post #26	15/11/2021	“If you don’t sacrifice for what you want, wha...
27	post #28	15/11/2021	Competition time! 🚵 WIN a copy of ‘Bikepacking...
29	post #30	16/11/2021	Master and Apprentice extend their time at the...
30	post #31	16/11/2021	When life tackles you to the ground, remember ...
32	post #33	17/11/2021	The AC40 by the numbers. 18m mast height 📐 11....
33	post #34	15/11/2021	With the first pick of the 2021 draft, Ignite ...
34	post #35	16/11/2021	Awesome to see our skip nominated for @worldru...
36	post #37	15/11/2021	Recently been vibing with spas🙏🏽 . . . . This ...
37	post #38	15/11/2021	🤩 The #SuperRugbyPacific draw is HERE!! Part 1 👉
39	post #40	15/11/2021	Get 4 hours of power (E-bike hire + mid-week d...
40	post #41	17/11/2021	ICYMI | Our 2022 squad has a new addition 💪 *“I...*
41	post #42	14/11/2021	💙World Diabetes Day 💙 #TB to when our boy beca...
42	post #43	15/11/2021	A great few days training trying to get my Sup...
44	post #45	14/11/2021	Smiling because PRE-SEASON starts in 15 Days 🏃...
45	post #46	16/11/2021	What a journey it was. The commitment from thi...
48	post #49	15/11/2021	Long time no see black line! 🏊‍♀️ A perfect s...
49	post #50	15/11/2021	Week 1 of 2 in FL to wrap up the season ✅ than...
50	post #51	15/11/2021	Competition time! 🚵 WIN a copy of ‘Bikepacking...
53	post #54	16/11/2021	Super Impressive🧡 will you try this challenge!...
54	post #55	14/11/2021	Bike and a hike 🔥 loving making the most of NZ...
55	post #56	14/11/2021	BELIEVE | We are right behind you, @blackcapsn...
56	post #57	15/11/2021	Did you know we offer Zip part pay for purchas...

57	post #58	14/11/2021	Back with my bestie 🥰🖤 @_timasavea
61	post #62	15/11/2021	Making the final of the Scottish Cup with @cel...
62	post #63	17/11/2021	Our big boy 🐾
64	post #65	15/11/2021	Less than a month to vote for @ockhamresidenti...
68	post #69	16/11/2021	#TrainingCamp | 🗣 "Definitely out of shape, do...
70	post #71	17/11/2021	So close fellas! We go again next year, up the...
73	post #74	18/11/2021	We're excited to reunite, Tāmaki Makaurau ✈❤ B...
74	post #75	17/11/2021	Will always appreciate the memories made with ...
75	post #76	17/11/2021	No matter how you choose to move your body 💃, ...
77	post #78	14/11/2021	See you today for our first day of Nippers/Jun...
79	post #80	15/11/2021	That first glimpse of Franz Josef glacier is a...
80	post #81	16/11/2021	Mental resilience gets you through a tough wor...
82	post #83	17/11/2021	NEW! Introducing our new premium 104L Ice Box ...

```
def count_special_char(string):
    special_char = 0
    for i in range(len(string)):
        if(string[i].isalpha()):
            continue
        else:
            special_char = special_char + 1

df["new"]=df.apply(count_special_char, axis = 0)
df
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-245-1003b57a28b1> in <module>()
      8             special_char = special_char + 1
      9 
---> 10 df["new"]=df.apply(count_special_char, axis = 1)
     11 df

                              4 frames

<ipython-input-245-1003b57a28b1> in count_special_char(string)
      3 
      4     for i in range(len(string)):
----> 5         if(string[i].isalpha()):
      6             continue
      7         else:

AttributeError: 'float' object has no attribute 'isalpha'
```

SEARCH STACK OVERFLOW