Exploratory Data Analysis DoQA Cooking Development Set Output

```
pip install textatistic
```

```
Requirement already satisfied: textatistic in /usr/local/lib/python3.10/dist-packages (0.0.1)
Requirement already satisfied: pyhyphen>=2.0.5 in /usr/local/lib/python3.10/dist-packages (from textatistic) (4.
Requirement already satisfied: wheel>=0.36.0 in /usr/local/lib/python3.10/dist-packages (from pyhyphen>=2.0.5->t
Requirement already satisfied: setuptools>=52.0 in /usr/local/lib/python3.10/dist-packages (from pyhyphen>=2.0.5
Requirement already satisfied: appdirs>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from pyhyphen>=2.0.5->
Requirement already satisfied: requests>=2.25 in /usr/local/lib/python3.10/dist-packages (from pyhyphen>=2.0.5->
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from request
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25->pyh
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.2
```

```
import json
import pandas as pd
from textatistic import Textatistic
from pathlib import Path
import re


with open("/content/doqa-cooking-dev-v2.1.json", 'r') as file:
    # Load the JSON data
    json_data = json.load(file)


# Access the 'data' key
data = json_data['data']


# Create lists to store the extracted data
titles = []
backgrounds = []
all_paragraphs = []

for entry in data:
   title = entry["title"]
   background = entry["background"]
   paragraphs = [c["context"] for c in entry["paragraphs"]]
   # Store each variable in separate lists
   titles.append(title)
   backgrounds.append(background)
   all_paragraphs.append(paragraphs)
#three separate lists: titles, backgrounds, and all_paragraphs
title_df = pd.DataFrame({'Title': titles})
background_df = pd.DataFrame({'Background': backgrounds})
paragraphs_df = pd.DataFrame({'Paragraphs': all_paragraphs})


title_df = title_df['Title'].tolist()
background_df = background_df['Background'].tolist()
paragraphs_df = paragraphs_df['Paragraphs'].tolist()
```

```
def clean_string(text): # Cleaning
    """re.sub(pattern, repl, string).
    Returns the string obtained by replacing the leftmost
    non-overlapping occurrences of pattern in string by the
    replacement thus removing any urls
    """
    return " ".join(re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", str(text)).split())


TextOnlyTitle = [clean_string(Title) for Title in title_df]
TextOnlyBackground = [clean_string(Background) for Background in background_df]
TextOnlyParagraphs = [clean_string(Paragraphs) for Paragraphs in paragraphs_df]


TextOnlyTitle[:2]
```

```
['What to add to the batter of the cake to avoid hardening when the gluten formation cant be avoided',
 'Is bacon fat supposed to congeal at room temperature']
```

```
TextOnlyBackground[:2]
```

```
['So over mixing batter forms gluten which in turn hardens the cake FineThe problem is that I dont want lumps in the cakes and the above statement prevents me from fine mixing the batter So is there something which I can add to the batter more milk to make the outcome soft despite Gluten',
 'My grandma told me its a good idea to save the bacon drippings in a sealable container to cook with later I remember when I used to watch her cook with it it was always solid I have started saving the fat from my bacon only the bottom of the can is the only part that ever congeals The top always seems sort of semiliquid Is that ok When cooking with it what part should I use and what is the difference between solid and merely viscous bacon fat']
```

```
TextOnlyParagraphs[:2]
```

```
['Milk wont help you its mostly water and gluten develops from flour more accurately specific proteins in flour and waterThe way to reduce gluten development is to incorporate more fat into the batter Lipids are hydrophobic and will prevent further hydration of the gluteninUsing a lowerprotein flour will also help If youre not already using cake flour the reason its called cake flour is because of the lower protein contentThat being said have you actually tried leaving the batter coarse Just because the batter is lumpy does not mean that the cake will have big lumps The entire mixture is wet so unless you leave huge lumps of dry flour in the batter you wont end up with a lumpy cake Theres a difference between dont overmix and dont mix youre supposed to mix enough to incorporate just dont try homogenize it CANNOTANSWER',
 'To answer what I think is the question you put all of the grease into a container and theres a residue at the top bacon drippings are not 100 fat There are also solid pieces of bacon in there and other impurities from the curing processWhen rendering bacon fat you should line the container with a paper towel first or cheesecloth if you have it Pour the bacon drippings onto the paper towel and the fat will drain out the bottom the solids will be left behind and you can dispose of them Youll be left with mostly pure fatThe rendered fat will most definitely congeal the vessel once cooled should contain only a solid offwhite substance CANNOTANSWER']
```

Title

```
a_list = TextOnlyTitle


new_list_a = [x for x in a_list if len(x) < 1000]


textfile_a = open("fileA_file.txt", "w")


for element in new_list_a:
    textfile_a.write(element + "\n")


textfile_a.close()
```

```
textA = Path('fileA_file.txt').read_text()
```

```
a_string = textA
```

```
Str_EndFixA = a_string.replace("\n", ".")
```

```
readability = Textatistic(Str_EndFixA)
```

```
%precision 3
```

```
'%.3f'
```

```
readability.dict()
```

```
{'char_count': 7914,
 'word_count': 1480,
 'sent_count': 198,
 'sybl_count': 2070,
 'notdalechall_count': 447,
 'polysyblword_count': 123,
 'flesch_score': 80.922,
 'fleschkincaid_score': 3.829,
 'gunningfog_score': 6.314,
 'smog_score': 7.632,
 'dalechall_score': 8.776}
```

For Background

```
b_list = TextOnlyBackground
```

```
new_list_b = [x for x in b_list if len(x) < 1000]
```

```
textfile_b = open("fileB_file.txt", "w")
```

```
for element in new_list_b:
    textfile_b.write(element + "\n")
```

```
textfile_b.close()
```

```
textB = Path('fileB_file.txt').read_text()
```

```
b_string = textB
```

```
Str_EndFixB = b_string.replace("\n", ".")
```

```
readability = Textatistic(Str_EndFixB)
```

```
%precision 3
```

```
'%.3f'
```

```
readability.dict()
```

```
{'char_count': 61253,
 'word_count': 13930,
 'sent_count': 179,
 'sybl_count': 17688,
 'notdalechall_count': 2685,
 'polysyblword_count': 735,
 'flesch_score': 20.423,
 'fleschkincaid_score': 29.744,
 'gunningfog_score': 33.239,
 'smog_score': 14.705,
 'dalechall_score': 10.540}
```

For Paragraph

```
p_list = TextOnlyParagraphs
```

```
new_list_p = [x for x in p_list if len(x) < 1000]
```

```
textfile_p = open("fileP_file.txt", "w")
```

```
for element in new_list_p:
    textfile_p.write(element + "\n")
```

```
textfile_p.close()
```

```
textP = Path('fileP_file.txt').read_text()
```

```
p_string = textP
```

```
Str_EndFixP = p_string.replace("\n", ".")
```

```
readability = Textatistic(Str_EndFixP)
```

```
%precision 3
```

```
'%.3f'
```

```
readability.dict()
```

```
{'char_count': 75775,
 'word_count': 16675,
 'sent_count': 162,
 'sybl_count': 21245,
 'notdalechall_count': 3465,
 'polysyblword_count': 922,
 'flesch_score': -5.427,
 'fleschkincaid_score': 39.587,
 'gunningfog_score': 43.385,
 'smog_score': 16.758,
 'dalechall_score': 12.023}
```