Exploratory Data Analysis DoQA Cooking Training 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-train-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]
```

```
['Tips for grilling duck legs', 'Tips for grilling duck legs']
```

```
TextOnlyBackground[:2]
```

```
['I recently attempted to grill duck legs on my propane Webber I was afraid of flareups due to the high fat content in the duck meat so I grilled with somewhat low and indirect heat It took a long time but I got them looking lovely and brown and not burned The only problem was this they were tough and didnt taste very good at all Clearly I did something very wrong Any advice',
 'I recently attempted to grill duck legs on my propane Webber I was afraid of flareups due to the high fat content in the duck meat so I grilled with somewhat low and indirect heat It took a long time but I got them looking lovely and brown and not burned The only problem was this they were tough and didnt taste very good at all Clearly I did something very wrong Any advice']
```

```
TextOnlyParagraphs[:2]
```

```
['I think grilling is probably a bad plan for duck legs the fat content is a real danger like you said and duck legs are tough enough you probably want to confit them or braise themIf you absolutely have to grill them I would suggest confiting them at 200 degrees for three or four hours first you could use veggie oil in a pinch and then resting them in the fridge for a day or so in oil As for finishing them on the grill rinse them off gently reseason if needed cook flesh side down on a medium heat portion of the grill for a while until mostly heated through then flip them over on a high heat portion of the grill to crisp up the skin watching out for flares CANNOTANSWER',
 'I think grilling is probably a bad plan for duck legs the fat content is a real danger like you said and duck legs are tough enough you probably want to confit them or braise themIf you absolutely have to grill them I would suggest confiting them at 200 degrees for three or four hours first you could use veggie oil in a pinch and then resting them in the fridge for a day or so in oil As for finishing them on the grill rinse them off gently reseason if needed cook flesh side down on a medium heat portion of the grill for a while until mostly heated through then flip them over on a high heat portion of the grill to crisp up the skin watching out for flares 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': 41745,
 'word_count': 7837,
 'sent_count': 1037,
 'sybl_count': 10825,
 'notdalechall_count': 2495,
 'polysyblword_count': 592,
 'flesch_score': 82.309,
 'fleschkincaid_score': 3.656,
 'gunningfog_score': 6.045,
 'smog_score': 7.445,
 'dalechall_score': 9.038}
```

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': 311414,
 'word_count': 70743,
 'sent_count': 959,
 'sybl_count': 89969,
 'notdalechall_count': 14199,
 'polysyblword_count': 3662,
 'flesch_score': 24.369,
 'fleschkincaid_score': 28.186,
 'gunningfog_score': 31.578,
```

```
 'smog_score': 14.292,
 'dalechall_score': 10.465}
```

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': 414674,
 'word_count': 91628,
 'sent_count': 902,
 'sybl_count': 117064,
 'notdalechall_count': 18964,
 'polysyblword_count': 5174,
 'flesch_score': -4.357,
 'fleschkincaid_score': 39.103,
 'gunningfog_score': 42.892,
 'smog_score': 16.811,
 'dalechall_score': 11.943}
```