PythonCodeDairyTextCleaning.md

Python codes cleaning text data and generate wordcloud visualization

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

Word Count using CountVectorizer

Read the csv file into a DF

```
filelocation = "content_dairy.csv"

CSV_DF = pd.read_csv(filelocation)

print(CSV_DF)

content = CSV_DF["abstract"]
type(content)
```

Change series to list

```
content_list = content.tolist()
```

Instantiate CV

```
MyCV_content = CountVectorizer()

My_DTM = MyCV_content.fit_transform(content_list)
```

Convert DTM to a DF

```
print(MyCV_content.vocabulary_)
vocab_dict = MyCV_content.vocabulary_
dict_key = vocab_dict.keys()

print("The vocab is: ", dict_key, "\n\n")
```

NEXT - Use pandas to create data frames

```
My_DF_content = pd.DataFrame(My_DTM.toarray(), columns=dict_key)
```

Write to csv file

```
My_DF_content.to_csv('PubAg_Dairy_Content_wordcount.csv', index=False)
```

WordCloud Generation

Checking for NaN values

```
CSV_DF.isna().sum()
```

As shown in the result, no NaN values in the "abstract" column

Removing NaN Values

```
content_WC = CSV_DF["abstract"]
content_WC = content_WC.tolist()
```

Creating the text variable

```
text = " ".join(content_WC)
```

Creating word_cloud with text as argument in .generate() method

```
word_cloud = WordCloud(collocations=False, background_color='#fbeed1', width=800, height=400).generate(text)
```

Display the generated Word Cloud

```
plt.imshow(word_cloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```