Exploratory Data Analysis (EDA)

Load the data

In [1]:

```
from pickle import load, dumps
import json
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import os
import seaborn as sns
```

In [2]:

```
# Load the filepaths
with open('../raw_data/corpus_dict.pkl', 'rb') as handle:
    corpus_dict = load(handle)

json.loads(json.dumps(corpus_dict))
```

Out[2]:

```
{'1': {'1': 'S1-E1.pkl',
  '2': 'S1-E2.pkl',
  '3': 'S1-E3.pkl',
  '4': 'S1-E4.pkl',
  '5': 'S1-E5.pkl',
  '6': 'S1-E6.pkl',
  '7': 'S1-E7.pkl',
  '8': 'S1-E8.pkl',
  '9': 'S1-E9.pkl',
  '10': 'S1-E10.pkl',
  '11': 'S1-E11.pkl',
  '12': 'S1-E12.pkl',
  '13': 'S1-E13.pkl',
  '14': 'S1-E14.pkl',
  '15': 'S1-E15.pkl',
  '16': 'S1-E16.pkl',
  '17': 'S1-E17.pkl',
  '18': 'S1-E18.pkl',
```

In [3]:

```python
# Load in the dictionaries
raw_data_folder = "../raw_data/"

all_scripts_df = pd.DataFrame(columns=['character', 'text'])

for season, episodes in corpus_dict.items():
    for episode_nr, episode in episodes.items():
        with open(raw_data_folder+episode, 'rb') as handle:
            episode_script = load(handle)
            all_scripts_df = all_scripts_df.append(episode_script)

#all_scripts_df
```

...

In [4]:

```python
all_scripts_df
```

Out[4]:

```
              character                            text
  0              AIRMAN        Oh, man, this hand's ...
  1           interlude  One of the personnel deals out ...
  2              AIRMAN          Seven to the deuce, n...
  3              FEMALE           Aren't you guys afrai...
  4             OFFICER          Trust me. Nobody ever...
...                 ...                              ...
359             YAT'YIR          you merely seek to de...
360               WOMAN            if and when all Jaffa...
361             YAT'YIR            if our brothers refus...
362               GERAK            and they will see the...
363    TO BE CONTINUED

75210 rows × 2 columns
```

General Statistics

In [5]:

```python
# General statistics
all_scripts_df.describe()
```

Out[5]:

```
         character    text
count        75210   75210
unique        1943   66827
   top   interlude
  freq       13330    1308
```

In [6]:

```
# Total word count

#-- Add column with word count
words = all_scripts_df.text
all_scripts_df['word_count'] = words.apply(lambda x: len(x.split()))
#all_scripts_df.head()

total_word_count = all_scripts_df.word_count.sum()
print(f"The total word count is {total_word_count}")

# Total dialouge word count

dialouge_word_count = all_scripts_df.loc[all_scripts_df.character != 'interlude'].word_
print(f"The dialouge word count is {dialouge_word_count}")

# Percentage dialouge to total words
dalouge_word_percentage = round(dialouge_word_count/total_word_count,4)*100
print(f"{dalouge_word_percentage}% of words in all scripts are dialouge")
```

```
The total word count is 884424
The dialouge word count is 704473
79.65% of words in all scripts are dialouge
```

Word Clouds

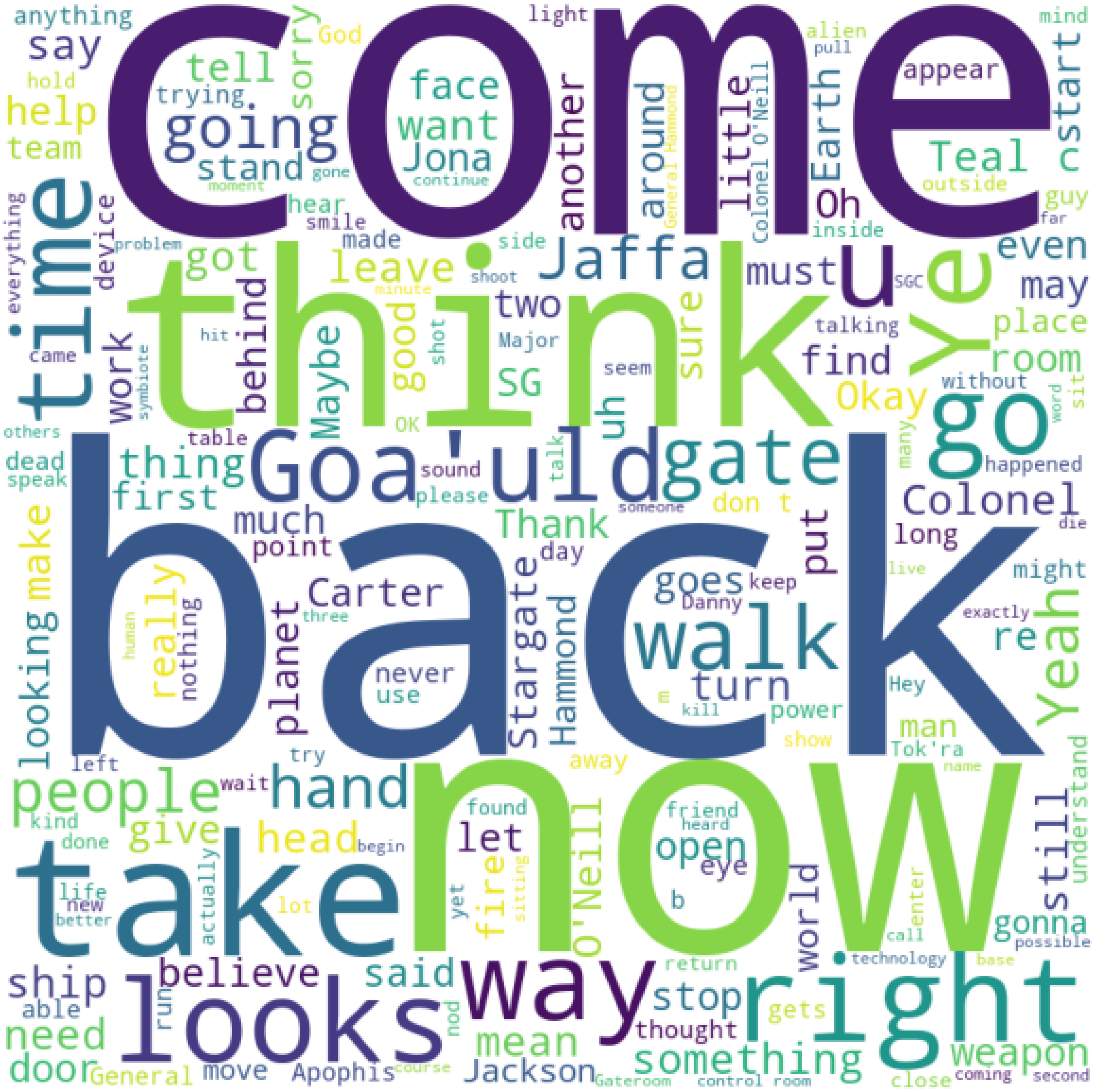
In [5]:

```
# Most frequent words word cloud
#-- Create a long string with all text
all_words = ' '.join(all_scripts_df.text)
all_words
```

...

In [10]:

```
#-- Create word cloud
#-- Source: https://www.geeksforgeeks.org/generating-word-cloud-python/ ; 17.10.2021

stopwords = STOPWORDS
stopwords.update(['know', 'look', 'see', 'well', 'S', 'one', "Teal'C", "Teal c", 'Danie
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(all_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Characters

In [9]:

```
# charachters sorted by most turns speaking, not most words
character_occourences = all_scripts_df.character.value_counts()
character_occourences.head(20)
```

Out[9]:

```
interlude                                           13330
                                     JACK            6188
                                     DANIEL          5282
                                     SAM             4613
                                     CARTER          4350
                                     O'NEILL         3598
                                     TEAL'C          3173
                                     HAMMOND         2996
                                     O NEILL         1240
                                     JACKSON         1217
                                     DANNY           1061
                                     JONAS           1010
                                     TEAL C           754
                                     JANET            681
                                     JACOB            676
                                     FRAISER          586
                                     MITCHELL         580
                                     VALA             485
                                     MAYBOURNE        422
                                     DAVIS            390
Name: character, dtype: int64
```

In [10]:

```
character_occourences[10:].tail(1000)
```

Out[10]:

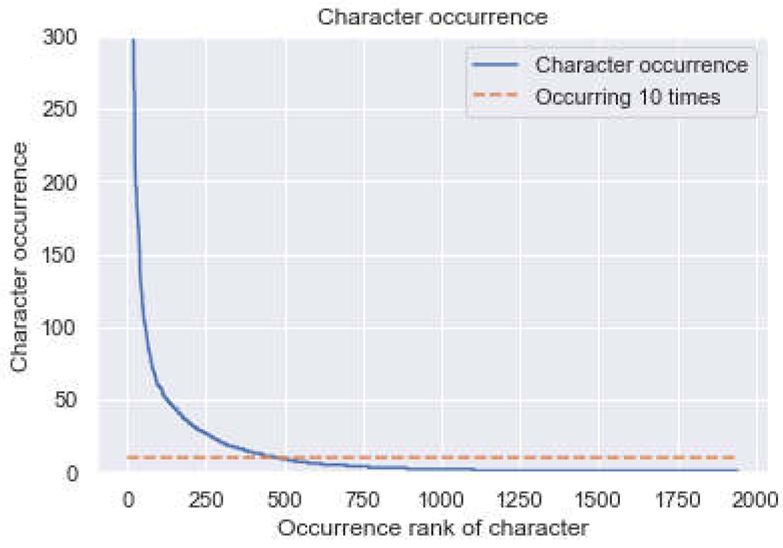
```
               OUTSIDE THE WAREHOUSE.                   2
               TO BE CONTINUED.                         2
                                     ALT-CARTER         2
                                     ELDER1             2
                                     MISSION CONTROL    2
                                                       ..
                                     CYLER              1
                                     GEEK #2            1
                                     FRANCE             1
                                     HAIMDALL           1
               INT  BRIDGE, PYRAMID SHIP                1
Name: character, Length: 1000, dtype: int64
```

In [11]:

```
# imports
# import seaborn as sns
# import numpy as np
# import pandas as pd

# inputs
cutoff = 0
character_occourence = character_occourences[cutoff:].values
character_number = list(range(cutoff,len(character_occourences),1))
occourence_cut = [10 for x in range(len(character_occourence))]

# convert to pandas dataframe
d = {'Character occurrence': character_occourence,
     'Occurring 10 times': occourence_cut}
df = pd.DataFrame(d, index = character_number)
#df.reset_index('Character number')

# plot using lineplot
sns.set(style='darkgrid')
# h = sns.lineplot(x='Character number', y='Character occourence', data=pdnumsqr)
h = sns.lineplot(data=df)
h. set(ylim=(0, 300))
h.set(xlabel='Occurrence rank of character', ylabel='Character occurrence')

h.set_title('Character occurrence')

```

Out[11]:

```
Text(0.5, 1.0, 'Character occurrence')
```



In [14]:

```
len(all_scripts_df.character.value_counts())
```

Out[14]:

```
1943
```

In [16]:

```
# characters sorted by total word count
test=all_scripts_df[['character', 'word_count']].groupby('character').sum().sort_values
test
```

Out[16]:

	word_count
character	
interlude	179951
DANIEL	65028
JACK	55680
CARTER	53193
SAM	52928
...	...
INSIDE SENTINEL CAVE.	0
INSIDE THE AIRPORT	0
INSIDE THE SGC.	0
INSIDE THE SHIP	0
BACK TO SGC	0

1943 rows × 1 columns

Next: Cleaning the data

woman = female

leerzeichen namen