This notebook loads the entire lexicon of words collected and creates a few visuals to help the viewer understand the content within the corpus.

Visuals Created:

- Word counts of top words
- Histogram of most frequently mentioned nouns
- Word plot of entire corpus

> Takeaway message: There are a number of significant words to be mindful of when discussing space. Now that we have this body of words involved, let's look deeper into understanding the way in which people understand these as it relates to sentiment, or opinion around them.

In [1]:

```
import re
import pandas as pd
import numpy as np
import spacy
import logging
import multiprocessing

from time import time
from collections import defaultdict
from IPython.display import Image

from gensim.models import Word2Vec
from gensim.models.phrases import Phrases, Phraser

%matplotlib inline
import matplotlib.pyplot as plt
from matplotlib import cm
import seaborn as sns

from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import silhouette_samples
```

```
/Users/tlipman/opt/anaconda3/envs/learn-env/lib/python3.6/site-package
s/gensim/similarities/__init__.py:15: UserWarning: The gensim.similarit
ies.levenshtein submodule is disabled, because the optional Levenshtein
package <https://pypi.org/project/python-Levenshtein/> is unavailable.
Install Levenhstein (e.g. `pip install python-Levenshtein`) to suppress
this warning.
  warnings.warn(msg)
```

In [2]:
```
pd.set_option('display.width', None)
pd.set_option('max_columns', None)
pd.set_option('max_colwidth', 200)

logging.basicConfig(format="%(levelname)s - %(asctime)s: %(message)s", d
atefmt= '%H:%M:%S', level=logging.INFO)
```

In [3]:
```
df = pd.read_csv('final.csv')
df.shape
```

```
/Users/tlipman/opt/anaconda3/envs/learn-env/lib/python3.6/site-package
s/IPython/core/interactiveshell.py:2714: DtypeWarning: Columns (4,7) ha
ve mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[3]:
```
(549902, 8)
```

In [4]:
```
df.head()
```

Out[4]:

	Unnamed: 0	**Unnamed: 0.1**	**text**	**favorite_count**	**user_id**	**mentions**
0	0	0	earth order survive must stop global warming mar order survive need global warming	14116.0	UCmERzF_P0BZWGGjr2wGGnMQ	NaN
1	1	1	phase 4 moon declares independence tired earth tax	12898.0	UCRgqsjV2VMb11prjm_bIC8Q	NaN
2	2	3	let get straight guy astronaut great public speaker also play guitar sing many lifetime doe normal person need accomplish	10670.0	UCwrM8uIAgp_QiA2VgdJeJRA	NaN
3	3	5	walk spider web australia thats called assisted suicide	9282.0	UC_m10vuJcLOosqYT5oOAKvg	NaN
4	4	6	love video send existentialist crisis others make want build rocket backyard leave right	6820.0	UCcnv-fzEfAhmRyWC60HFSSg	NaN

In [5]:
```
df.drop(['Unnamed: 0', 'Unnamed: 0.1'], axis=1, inplace=True)
```

In [6]:
```
df.head()
```

Out[6]:

	text	favorite_count	user_id	mentions	repost_count	
0	earth order survive must stop global warming mar order survive need global warming	14116.0	UCmERzF_P0BZWGGjr2wGGnMQ	NaN	0.0	U
1	phase 4 moon declares independence tired earth tax	12898.0	UCRgqsjV2VMb11prjm_bIC8Q	NaN	0.0	Ugxivz
2	let get straight guy astronaut great public speaker also play guitar sing many lifetime doe normal person need accomplish	10670.0	UCwrM8uIAgp_QiA2VgdJeJRA	NaN	0.0	UgzXTA
3	walk spider web australia thats called assisted suicide	9282.0	UC_m10vuJcLOosqYT5oOAKvg	NaN	0.0	U
4	love video send existentialist crisis others make want build rocket backyard leave right	6820.0	UCcnv-fzEfAhmRyWC60HFSSg	NaN	0.0	Ugy3G3

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
text                 0
favorite_count       0
user_id              2
mentions        219494
repost_count         0
post_id              0
dtype: int64
```

In [10]:
```
df_comments = df.drop(['favorite_count', 'user_id', 'mentions', 'repost_
count', 'post_id'], axis=1)
```

```
In [11]: df_comments.head()
```

Out[11]:

	text
0	earth order survive must stop global warming mar order survive need global warming
1	phase 4 moon declares independence tired earth tax
2	let get straight guy astronaut great public speaker also play guitar sing many lifetime doe normal person need accomplish
3	walk spider web australia thats called assisted suicide
4	love video send existentialist crisis others make want build rocket backyard leave right

In [12]:

```
sent = [row.split() for row in df_comments['text']]

phrases = Phrases(sent) # Detect phrases based on collocation counts.

bigram = Phraser(phrases) # The goal of Phraser() is to cut down memory
 consumption of Phrases()

sentences = bigram[sent] # transform the corpus based upon bigrams detec
ted
```

```
INFO - 23:24:29: collecting all words and their counts
INFO - 23:24:29: PROGRESS: at sentence #0, processed 0 words and 0 word
types
INFO - 23:24:29: PROGRESS: at sentence #10000, processed 200978 words a
nd 160468 word types
INFO - 23:24:29: PROGRESS: at sentence #20000, processed 416445 words a
nd 296596 word types
INFO - 23:24:30: PROGRESS: at sentence #30000, processed 634263 words a
nd 417432 word types
INFO - 23:24:30: PROGRESS: at sentence #40000, processed 850290 words a
nd 532143 word types
INFO - 23:24:30: PROGRESS: at sentence #50000, processed 1058834 words
and 628553 word types
INFO - 23:24:31: PROGRESS: at sentence #60000, processed 1267716 words
and 719481 word types
INFO - 23:24:31: PROGRESS: at sentence #70000, processed 1463786 words
and 809526 word types
INFO - 23:24:32: PROGRESS: at sentence #80000, processed 1653370 words
and 887719 word types
INFO - 23:24:32: PROGRESS: at sentence #90000, processed 1882233 words
and 976166 word types
INFO - 23:24:32: PROGRESS: at sentence #100000, processed 2118216 words
and 1080267 word types
INFO - 23:24:33: PROGRESS: at sentence #110000, processed 2342698 words
and 1175968 word types
INFO - 23:24:33: PROGRESS: at sentence #120000, processed 2540055 words
and 1249584 word types
INFO - 23:24:33: PROGRESS: at sentence #130000, processed 2774355 words
and 1345365 word types
INFO - 23:24:34: PROGRESS: at sentence #140000, processed 2885022 words
and 1413230 word types
INFO - 23:24:34: PROGRESS: at sentence #150000, processed 2995096 words
and 1479836 word types
INFO - 23:24:34: PROGRESS: at sentence #160000, processed 3104182 words
and 1544188 word types
INFO - 23:24:34: PROGRESS: at sentence #170000, processed 3211968 words
and 1606475 word types
INFO - 23:24:35: PROGRESS: at sentence #180000, processed 3319186 words
and 1666362 word types
INFO - 23:24:35: PROGRESS: at sentence #190000, processed 3425397 words
and 1725092 word types
INFO - 23:24:35: PROGRESS: at sentence #200000, processed 3531585 words
and 1781691 word types
INFO - 23:24:35: PROGRESS: at sentence #210000, processed 3635676 words
and 1837643 word types
INFO - 23:24:35: PROGRESS: at sentence #220000, processed 3740307 words
and 1890976 word types
INFO - 23:24:36: PROGRESS: at sentence #230000, processed 3843829 words
and 1946686 word types
INFO - 23:24:36: PROGRESS: at sentence #240000, processed 3945837 words
and 1999460 word types
INFO - 23:24:36: PROGRESS: at sentence #250000, processed 4048155 words
and 2050284 word types
INFO - 23:24:36: PROGRESS: at sentence #260000, processed 4148518 words
and 2099775 word types
INFO - 23:24:36: PROGRESS: at sentence #270000, processed 4249765 words
and 2149336 word types
```

```
INFO - 23:24:37: PROGRESS: at sentence #280000, processed 4349848 words
and 2198667 word types
INFO - 23:24:37: PROGRESS: at sentence #290000, processed 4451917 words
and 2249281 word types
INFO - 23:24:37: PROGRESS: at sentence #300000, processed 4557551 words
and 2299920 word types
INFO - 23:24:37: PROGRESS: at sentence #310000, processed 4666239 words
and 2347898 word types
INFO - 23:24:37: PROGRESS: at sentence #320000, processed 4776727 words
and 2399434 word types
INFO - 23:24:38: PROGRESS: at sentence #330000, processed 4884304 words
and 2447606 word types
INFO - 23:24:38: PROGRESS: at sentence #340000, processed 4994571 words
and 2490007 word types
INFO - 23:24:38: PROGRESS: at sentence #350000, processed 5105138 words
and 2532486 word types
INFO - 23:24:38: PROGRESS: at sentence #360000, processed 5216725 words
and 2571411 word types
INFO - 23:24:39: PROGRESS: at sentence #370000, processed 5334626 words
and 2615617 word types
INFO - 23:24:39: PROGRESS: at sentence #380000, processed 5447934 words
and 2660393 word types
INFO - 23:24:39: PROGRESS: at sentence #390000, processed 5557289 words
and 2704649 word types
INFO - 23:24:39: PROGRESS: at sentence #400000, processed 5665982 words
and 2741644 word types
INFO - 23:24:39: PROGRESS: at sentence #410000, processed 5772970 words
and 2784678 word types
INFO - 23:24:40: PROGRESS: at sentence #420000, processed 5885965 words
and 2823156 word types
INFO - 23:24:40: PROGRESS: at sentence #430000, processed 5998007 words
and 2860673 word types
INFO - 23:24:40: PROGRESS: at sentence #440000, processed 6110128 words
and 2900829 word types
INFO - 23:24:40: PROGRESS: at sentence #450000, processed 6222247 words
and 2940414 word types
INFO - 23:24:41: PROGRESS: at sentence #460000, processed 6334615 words
and 2978151 word types
INFO - 23:24:41: PROGRESS: at sentence #470000, processed 6447832 words
and 3022482 word types
INFO - 23:24:41: PROGRESS: at sentence #480000, processed 6557913 words
and 3058589 word types
INFO - 23:24:41: PROGRESS: at sentence #490000, processed 6670900 words
and 3103721 word types
INFO - 23:24:42: PROGRESS: at sentence #500000, processed 6780185 words
and 3148850 word types
INFO - 23:24:42: PROGRESS: at sentence #510000, processed 6895152 words
and 3192137 word types
INFO - 23:24:42: PROGRESS: at sentence #520000, processed 7005375 words
and 3230699 word types
INFO - 23:24:42: PROGRESS: at sentence #530000, processed 7115941 words
and 3273019 word types
INFO - 23:24:42: PROGRESS: at sentence #540000, processed 7227965 words
and 3320404 word types
INFO - 23:24:43: collected 3360764 token types (unigram + bigrams) from
a corpus of 7338277 words and 549902 sentences
INFO - 23:24:43: merged Phrases<3360764 vocab, min_count=5, threshold=1
```

```
0.0, max_vocab_size=40000000>
INFO - 23:24:43: Phrases lifecycle event {'msg': 'built Phrases<3360764
vocab, min_count=5, threshold=10.0, max_vocab_size=40000000> in 14.00
s', 'datetime': '2021-04-12T23:24:43.101731', 'gensim': '4.0.1', 'pytho
n': '3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:42:17) \n[GCC 4.
2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)]', 'platform': 'Dar
win-19.6.0-x86_64-i386-64bit', 'event': 'created'}
INFO - 23:24:43: exporting phrases from Phrases<3360764 vocab, min_coun
t=5, threshold=10.0, max_vocab_size=40000000>
INFO - 23:24:53: FrozenPhrases lifecycle event {'msg': 'exported Frozen
Phrases<33257 phrases, min_count=5, threshold=10.0> from Phrases<336076
4 vocab, min_count=5, threshold=10.0, max_vocab_size=40000000> in 9.95
s', 'datetime': '2021-04-12T23:24:53.150230', 'gensim': '4.0.1', 'pytho
n': '3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:42:17) \n[GCC 4.
2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)]', 'platform': 'Dar
win-19.6.0-x86_64-i386-64bit', 'event': 'created'}
```

In [13]:

```
model = Word2Vec()

t = time()

model.build_vocab(sentences)

print('Time to build vocab: {} mins'.format(round((time() - t) / 60, 2
)))
```

```
INFO - 23:26:39: Word2Vec lifecycle event {'params': 'Word2Vec(vocab=0,
vector_size=100, alpha=0.025)', 'datetime': '2021-04-12T23:26:39.80382
4', 'gensim': '4.0.1', 'python': '3.6.9 |Anaconda, Inc.| (default, Jul
30 2019, 13:42:17) \n[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_40
1/final)]', 'platform': 'Darwin-19.6.0-x86_64-i386-64bit', 'event': 'cr
eated'}
INFO - 23:26:39: collecting all words and their counts
INFO - 23:26:39: PROGRESS: at sentence #0, processed 0 words, keeping 0
word types
INFO - 23:26:40: PROGRESS: at sentence #10000, processed 181605 words,
keeping 22936 word types
INFO - 23:26:40: PROGRESS: at sentence #20000, processed 376652 words,
keeping 34046 word types
INFO - 23:26:40: PROGRESS: at sentence #30000, processed 574037 words,
keeping 42409 word types
INFO - 23:26:41: PROGRESS: at sentence #40000, processed 769980 words,
keeping 49490 word types
INFO - 23:26:41: PROGRESS: at sentence #50000, processed 957696 words,
keeping 54846 word types
INFO - 23:26:41: PROGRESS: at sentence #60000, processed 1146640 words,
keeping 59573 word types
INFO - 23:26:42: PROGRESS: at sentence #70000, processed 1324129 words,
keeping 64631 word types
INFO - 23:26:42: PROGRESS: at sentence #80000, processed 1494027 words,
keeping 69111 word types
INFO - 23:26:42: PROGRESS: at sentence #90000, processed 1700627 words,
keeping 73112 word types
INFO - 23:26:43: PROGRESS: at sentence #100000, processed 1915306 word
s, keeping 78153 word types
INFO - 23:26:43: PROGRESS: at sentence #110000, processed 2119482 word
s, keeping 82519 word types
INFO - 23:26:43: PROGRESS: at sentence #120000, processed 2298568 word
s, keeping 86051 word types
INFO - 23:26:44: PROGRESS: at sentence #130000, processed 2510600 word
s, keeping 91023 word types
INFO - 23:26:44: PROGRESS: at sentence #140000, processed 2608261 word
s, keeping 101129 word types
INFO - 23:26:44: PROGRESS: at sentence #150000, processed 2706556 word
s, keeping 110776 word types
INFO - 23:26:44: PROGRESS: at sentence #160000, processed 2803828 word
s, keeping 120309 word types
INFO - 23:26:44: PROGRESS: at sentence #170000, processed 2899512 word
s, keeping 129843 word types
INFO - 23:26:45: PROGRESS: at sentence #180000, processed 2994701 word
s, keeping 138893 word types
INFO - 23:26:45: PROGRESS: at sentence #190000, processed 3089187 word
s, keeping 147967 word types
INFO - 23:26:45: PROGRESS: at sentence #200000, processed 3182770 word
s, keeping 156415 word types
INFO - 23:26:45: PROGRESS: at sentence #210000, processed 3275792 word
s, keeping 165288 word types
INFO - 23:26:45: PROGRESS: at sentence #220000, processed 3368117 word
s, keeping 173204 word types
INFO - 23:26:46: PROGRESS: at sentence #230000, processed 3460520 word
s, keeping 181755 word types
INFO - 23:26:46: PROGRESS: at sentence #240000, processed 3551375 word
s, keeping 189917 word types
```

```
INFO - 23:26:46: PROGRESS: at sentence #250000, processed 3642193 word
s, keeping 197299 word types
INFO - 23:26:46: PROGRESS: at sentence #260000, processed 3732066 word
s, keeping 204484 word types
INFO - 23:26:46: PROGRESS: at sentence #270000, processed 3822471 word
s, keeping 211565 word types
INFO - 23:26:47: PROGRESS: at sentence #280000, processed 3912851 word
s, keeping 219017 word types
INFO - 23:26:47: PROGRESS: at sentence #290000, processed 4004143 word
s, keeping 226223 word types
INFO - 23:26:47: PROGRESS: at sentence #300000, processed 4098876 word
s, keeping 233657 word types
INFO - 23:26:47: PROGRESS: at sentence #310000, processed 4195449 word
s, keeping 240955 word types
INFO - 23:26:47: PROGRESS: at sentence #320000, processed 4295193 word
s, keeping 249097 word types
INFO - 23:26:47: PROGRESS: at sentence #330000, processed 4391707 word
s, keeping 256550 word types
INFO - 23:26:48: PROGRESS: at sentence #340000, processed 4488286 word
s, keeping 262203 word types
INFO - 23:26:48: PROGRESS: at sentence #350000, processed 4585971 word
s, keeping 267858 word types
INFO - 23:26:48: PROGRESS: at sentence #360000, processed 4683238 word
s, keeping 273072 word types
INFO - 23:26:48: PROGRESS: at sentence #370000, processed 4784495 word
s, keeping 279430 word types
INFO - 23:26:49: PROGRESS: at sentence #380000, processed 4884361 word
s, keeping 285637 word types
INFO - 23:26:49: PROGRESS: at sentence #390000, processed 4981192 word
s, keeping 291738 word types
INFO - 23:26:49: PROGRESS: at sentence #400000, processed 5074761 word
s, keeping 296551 word types
INFO - 23:26:49: PROGRESS: at sentence #410000, processed 5170078 word
s, keeping 301670 word types
INFO - 23:26:49: PROGRESS: at sentence #420000, processed 5267302 word
s, keeping 306636 word types
INFO - 23:26:50: PROGRESS: at sentence #430000, processed 5364308 word
s, keeping 311369 word types
INFO - 23:26:50: PROGRESS: at sentence #440000, processed 5462759 word
s, keeping 316415 word types
INFO - 23:26:50: PROGRESS: at sentence #450000, processed 5561254 word
s, keeping 321416 word types
INFO - 23:26:50: PROGRESS: at sentence #460000, processed 5659337 word
s, keeping 326008 word types
INFO - 23:26:50: PROGRESS: at sentence #470000, processed 5759892 word
s, keeping 332245 word types
INFO - 23:26:51: PROGRESS: at sentence #480000, processed 5856778 word
s, keeping 337080 word types
INFO - 23:26:51: PROGRESS: at sentence #490000, processed 5956006 word
s, keeping 343607 word types
INFO - 23:26:51: PROGRESS: at sentence #500000, processed 6054326 word
s, keeping 350213 word types
INFO - 23:26:51: PROGRESS: at sentence #510000, processed 6153503 word
s, keeping 355889 word types
INFO - 23:26:51: PROGRESS: at sentence #520000, processed 6251075 word
s, keeping 361135 word types
INFO - 23:26:52: PROGRESS: at sentence #530000, processed 6350587 word
```

```
s, keeping 367368 word types
INFO - 23:26:52: PROGRESS: at sentence #540000, processed 6453123 word
s, keeping 374338 word types
INFO - 23:26:52: collected 380095 word types from a corpus of 6551262 r
aw words and 549902 sentences
INFO - 23:26:52: Creating a fresh vocabulary
INFO - 23:26:53: Word2Vec lifecycle event {'msg': 'effective_min_count=
5 retains 78526 unique words (20.659571949117982%% of original 380095,
drops 301569)', 'datetime': '2021-04-12T23:26:53.191413', 'gensim': '4.
0.1', 'python': '3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:42:1
7) \n[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)]', 'pla
tform': 'Darwin-19.6.0-x86_64-i386-64bit', 'event': 'prepare_vocab'}
INFO - 23:26:53: Word2Vec lifecycle event {'msg': 'effective_min_count=
5 leaves 6106224 word corpus (93.2068355684752%% of original 6551262, d
rops 445038)', 'datetime': '2021-04-12T23:26:53.192142', 'gensim': '4.
0.1', 'python': '3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:42:1
7) \n[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)]', 'pla
tform': 'Darwin-19.6.0-x86_64-i386-64bit', 'event': 'prepare_vocab'}
INFO - 23:26:53: deleting the raw counts dictionary of 380095 items
INFO - 23:26:54: sample=0.001 downsamples 25 most-common words
INFO - 23:26:54: Word2Vec lifecycle event {'msg': 'downsampling leaves
estimated 5844686.7114740275 word corpus (95.7%% of prior 6106224)', 'd
atetime': '2021-04-12T23:26:54.004688', 'gensim': '4.0.1', 'python':
'3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:42:17) \n[GCC 4.2.1 C
ompatible Clang 4.0.1 (tags/RELEASE_401/final)]', 'platform': 'Darwin-1
9.6.0-x86_64-i386-64bit', 'event': 'prepare_vocab'}
INFO - 23:26:55: estimated required memory for 78526 words and 100 dime
nsions: 102083800 bytes
INFO - 23:26:55: resetting layer weights
INFO - 23:26:55: Word2Vec lifecycle event {'update': False, 'trim_rul
e': 'None', 'datetime': '2021-04-12T23:26:55.126631', 'gensim': '4.0.
1', 'python': '3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:42:17)
\n[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)]', 'platfo
rm': 'Darwin-19.6.0-x86_64-i386-64bit', 'event': 'build_vocab'}

Time to build vocab: 0.26 mins
```

EXPLORATORY DATA ANALYSIS

In [28]:

```python
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /Users/tlipman/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/tlipman/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

Out[28]:

```
True
```

In [18]:

```
#most frequent and least frequent words
freq = pd.Series(' '.join(df['text']).split()).value_counts()[:20]
freq
```

Out[18]:

```
space     107062
mar        79619
nasa       67312
would      43012
like       42940
wa         38483
spacex     37707
one        33798
earth      33484
moon       29518
time       29140
ha         28651
people     28428
get        27454
year       24171
know       23947
go         23305
think      23224
make       22174
dont       21944
dtype: int64
```

In [21]:

```
desc_str = ' '.join(df['text'].tolist())
```

In [22]:

```
tokens = nltk.word_tokenize(desc_str) #tokenizing
print(len(tokens))
```

```
7347713
```

```
In [29]: tokens_pos = nltk.pos_tag(tokens)
         pos_df = pd.DataFrame(tokens_pos, columns = ('word','POS'))
         pos_sum = pos_df.groupby('POS', as_index=False).count() # group by POS t
         ags
         pos_sum.sort_values(['word'], ascending=[False]) # in descending order o
         f number of words per tag
```

Out[29]:

	POS	word
12	NN	3377507
8	JJ	1426714
20	RB	433568
31	VBP	306608
3	CD	300477
29	VBG	289131
28	VBD	249758
27	VB	215184
15	NNS	191311
7	IN	153813
30	VBN	114729
11	MD	88394
32	VBZ	55768
9	JJR	24502
10	JJS	23262
6	FW	20201
4	DT	17656
21	RBR	15145
13	NNP	10901
2	CC	6404
25	TO	6243
23	RP	4077
18	PRP	2771
26	UH	2771
0	$	2582
33	WDT	2063
34	WP	2034
36	WRB	1851
22	RBS	1335
35	WP$	479
19	PRP$	172
5	EX	141
17	POS	106
16	PDT	27

	POS	word
24	SYM	12
14	NNPS	12
1	''	3
37	``	1

In [60]:

```
#the 100 most common nouns
filtered_pos = [ ]
for one in tokens_pos:
    if one[1] == 'NN' or one[1] == 'NNS' or one[1] == 'NNP' or one[1] == 'NNPS':
        filtered_pos.append(one)
print ("There are a total of", round(len(filtered_pos)/1000000, 4), "million nouns within the corpus.")
fdist_pos = nltk.FreqDist(filtered_pos)
top_100_words = fdist_pos.most_common(100)
```

```
There are a total of 3.5797 million nouns within the corpus.
```

In [40]:

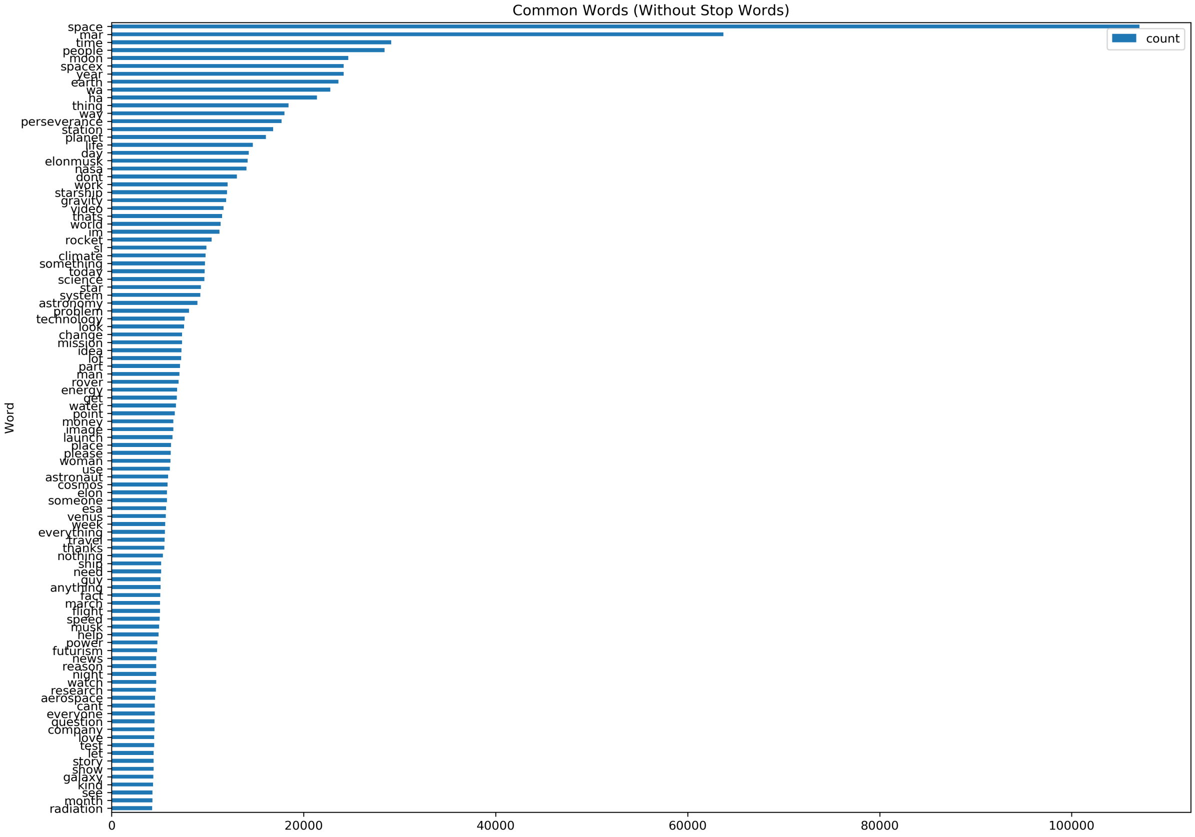
```
top_words_df = pd.DataFrame(top_100_words, columns = ('pos','count'))
top_words_df['Word'] = top_words_df['pos'].apply(lambda x: x[0]) # split the tuple of POS
top_words_df = top_words_df.drop('pos', 1) # drop the previous column
top_words_df.head(20)
```

Out[40]:

	count	Word
0	107062	space
1	63736	mar
2	29140	time
3	28428	people
4	24650	moon
5	24173	spacex
6	24171	year
7	23634	earth
8	22788	wa
9	21398	ha
10	18419	thing
11	18006	way
12	17707	perseverance
13	16817	station
14	16069	planet
15	14717	life
16	14285	day
17	14177	elonmusk
18	14049	nasa
19	13041	dont

In [38]:

```
fig, ax = plt.subplots(figsize=(16,12), dpi=300)
top_words_df.sort_values(by='count').plot.barh(x='Word',
                         y='count',
                         ax=ax)

ax.set_title("Common Words (Without Stop Words)")

plt.show()
```

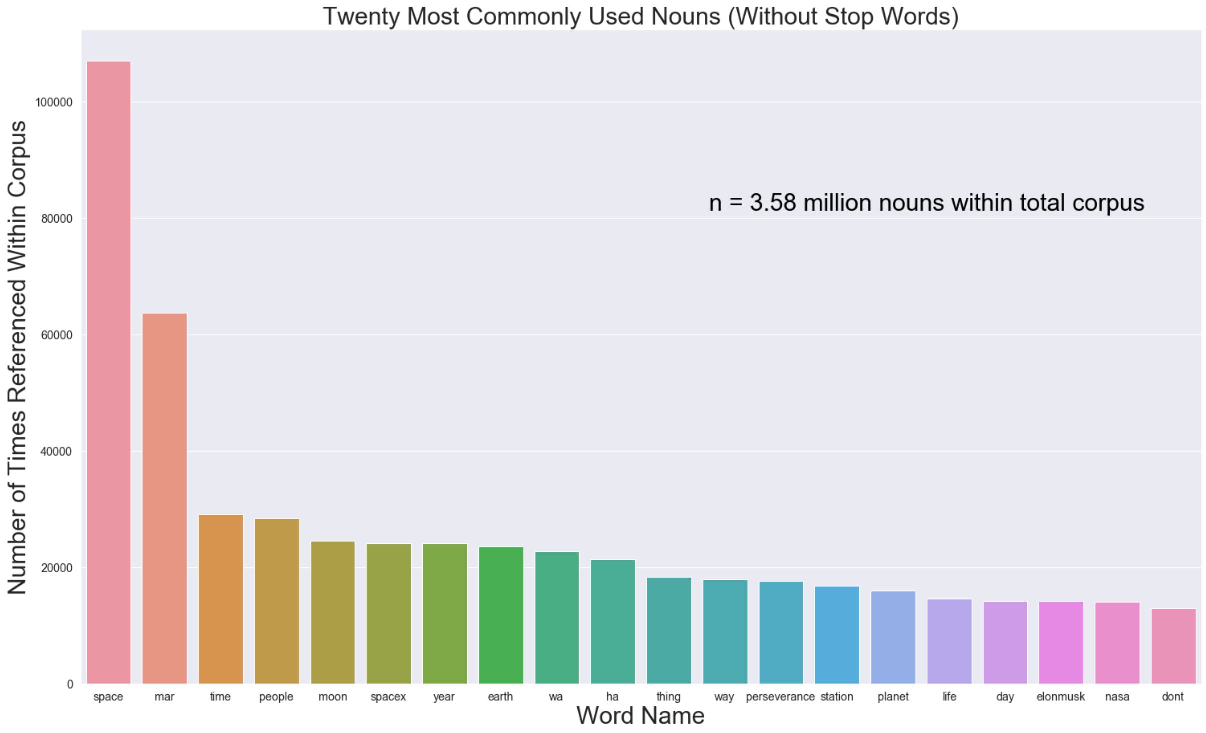


In [52]:

```
top_20 = top_words_df.head(20)
```

In [59]:

```
plt.figure(figsize=(25,15))
sns.set(font_scale=1.3)
pal = sns.color_palette("husl", 8)
ax = sns.barplot(x="Word", y="count",
                 data=top_20)
ax.set_title('Twenty Most Commonly Used Nouns (Without Stop Words)', fon
tsize=30)
ax.set_ylabel('Number of Times Referenced Within Corpus', fontsize=30)
ax.set_xlabel('Word Name', fontsize=30)
ax.text(.95, .75, 'n = 3.58 million nouns within total corpus',
        color='black', fontsize=30,
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes);
```



In [66]:

```
from wordcloud import WordCloud
```

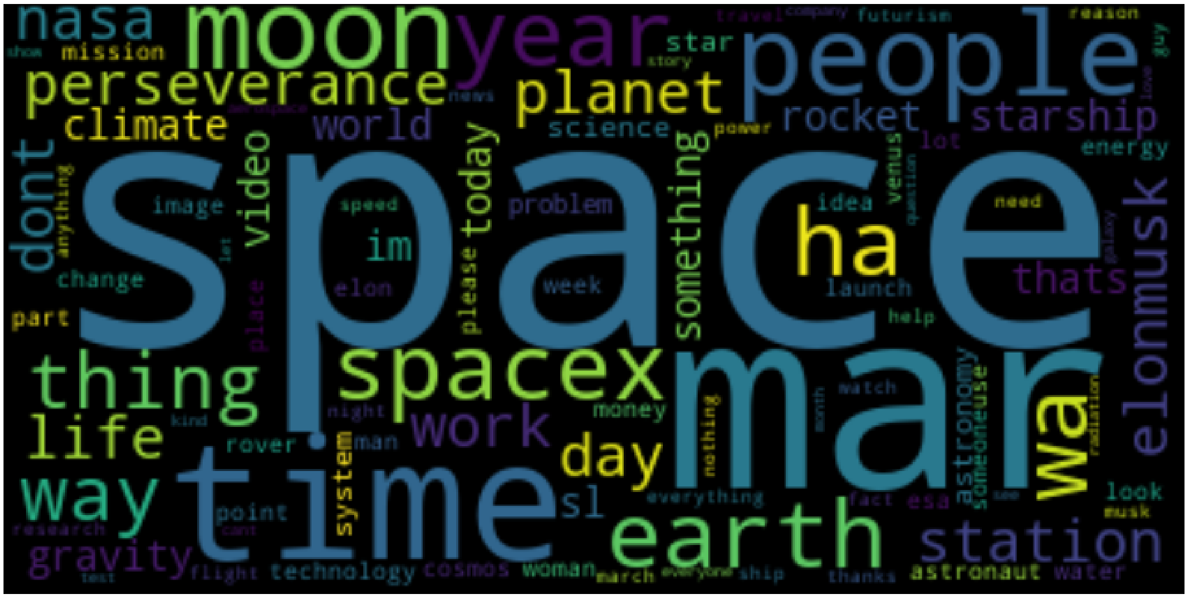
In [67]:

```
word_counts = ' '.join(top_words_df['Word'].tolist())
print(type(word_counts))
```

```
<class 'str'>
```

In [84]:

```
wordcloud = WordCloud().generate(word_counts)
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



In [14]:

```
# creating a word frequency count for each individual word
# ensuring that lemmatization, removal of stop words, and bigrams reduce
d
# the total diversity of sentiment to be able to be more accurately meas
ured and understood

word_freq = defaultdict(int)
for sent in sentences:
    for i in sent:
        word_freq[i] += 1
len(word_freq)
```

Out[14]:

```
380095
```

In [15]:

```
sorted(word_freq, key=word_freq.get, reverse=True)[:10]
```

Out[15]:

```
['space', 'mar', 'nasa', 'would', 'wa', 'like', 'one', 'earth', 'time',
'get']
```

Clustering

In [92]:

```
text = df_comments['text'] # establish the of text documents

vectorizer = TfidfVectorizer() # create the transform

X_tfidf = vectorizer.fit_transform(text)
```

In [86]:

```
x = df_comments.drop('text', 1)
y = vectorizer.fit_transform(df_comments['text'])
```

In [87]:

```
# displaying the multidimentionality of the dataset
y.shape
```

Out[87]:

```
(549902, 351282)
```

In [93]:

```
kmeans = KMeans(n_clusters=10,
            init='k-means++',
            n_init=10,
            max_iter=300,
            tol=1e-04,
            random_state=0)

y_kmeans = kmeans.fit_predict(X_tfidf, df_comments['retweets'])

kmeans.inertia_
```

Out[93]:

```
539320.4262618574
```

In [94]:

```
df_comments['cluster'] = y_kmeans
```

In [96]:

```
df_comments.head(40)
```

Out[96]:

	text	cluster
0	earth order survive must stop global warming mar order survive need global warming	2
1	phase 4 moon declares independence tired earth tax	4
2	let get straight guy astronaut great public speaker also play guitar sing many lifetime doe normal person need accomplish	0
3	walk spider web australia thats called assisted suicide	0
4	love video send existentialist crisis others make want build rocket backyard leave right	0
5	man guy really know paint picture word	9
6	go mar people start flat mar society	2
7	danger entirely different fear coolest quote ever	0
8	build city moon imagine looking seeing crescent moon dark part lit light citites	4
9	else thought title meant went physically blind whilst space	3
10	chris next time walk spider web dont go crazy go caveman instinct well australia	9
11	iraq war cost 1 7 trillion imagine elon could done fraction money	0
12	radiates steady coolness like space exploring james bond could listen talk hour	3
13	room childhood hero got dressed childhood hero see chris hadfield	0
14	kerbal space program player confirm approx 1 9 launch end fiery death	3
15	want guy walk around behind narrating life acclaimed optimistic nihilism	0
16	hope elon musk life 100 year old	9
17	wish alien met guy would great impression human	9
18	chris hadfield either ull magically floating space get excited chris hadfield ull dead go depressed	3
19	1950 moon base one day soon 2019 moon base one day soon	4
20	support kurzgesagt learn brilliant go rg nutshell sign free first 688 people go link get 20 annual premium subscription	0
21	arguement earthling moon people would end like moon person stupid earthling earthling lunatic	4
22	planet trying best keep u safe space demon yet fing	3
23	swear scientist smash head keyboard make name	0
24	milky way come andromeda cant milky way parent arent home andromeda travel 300 km	0
25	billion year like milkdromedians take comment section	9
26	wa sweating watching heart wa going crazy even though happened 50 year ago imagine crew wa feeling time	8
27	look manly yet nerdy gotta keep balance right	0
28	2019 kid might live mar 2576 kid might live earth	2
29	school got skype call told u story got ask question experience wa like best day life	8
30	want learn space check space product kurzgesagt shop designed love produced care getting something kurzgesagt shop best way support u keep video free everyone worldwide shipping available	3

	text	cluster
31	set shipping address next local group amazon prime get order shipped free within two day	0
32	since science know century old like think much time ahead u could eventually find solution right unimaginable	7
33	weird think specie born galaxy future way know big bang wonder suffering fate different subject probably	9
34	walk every spiderweb see spider cry hour hard work	0
35	hand best presentation ive ever seen subject amazing	0
36	took away even spider ridiculously polite canada	0
37	born late explore earth born early discover universe	9
38	imagine math broken spaceship get back home insane	0
39	wait second astronaut sing play guitar great public speaker epitome perfection	0

Visualizing the clusters

Elbow Method | Quantifying Distortion

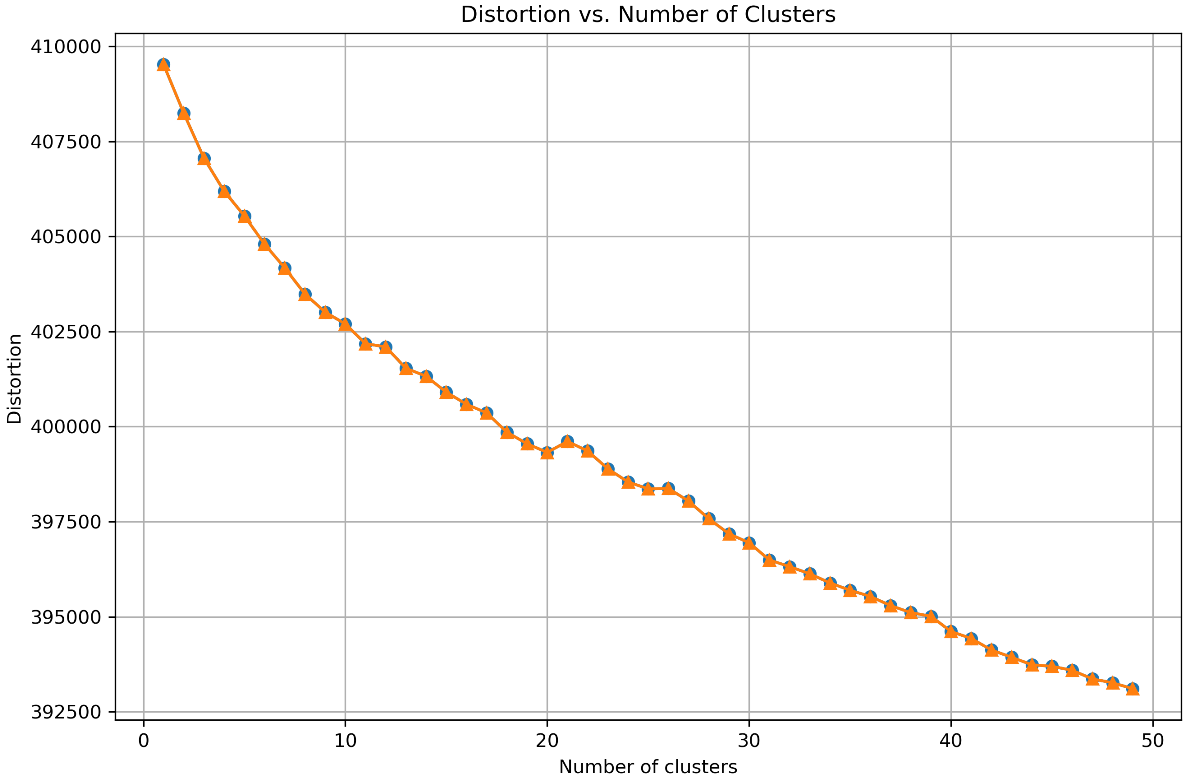
In [59]:

```
distortions = []
ScoreList    = []
maxNumberOfClusters = 50

for i in range(1, maxNumberOfClusters):
    km = KMeans(n_clusters=i,
                init='k-means++',
                n_init=10,
                max_iter=300,
                random_state=0)
    km.fit(X_tfidf)
    distortions.append(km.inertia_)
    ScoreList.append(-km.score(x_train))
```

```
------------------------------------------------------------------------
----
AttributeError                          Traceback (most recent call l
ast)
<ipython-input-59-ea1cf397e938> in <module>()
     14
     15
---> 16 plt.plot(range(1, maxNumberOfClusters), distortions, marker='o'
)
     17 plt.plot(range(1, maxNumberOfClusters), ScoreList, marker='^')
     18 plt.xlabel('Number of clusters')

AttributeError: module 'matplotlib' has no attribute 'plot'
```

In [66]:

```
plt.figure(figsize=(9, 6), dpi=300)
plt.plot(range(1, maxNumberOfClusters), distortions, marker='o')
plt.plot(range(1, maxNumberOfClusters), ScoreList, marker='^')
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.title('Distortion vs. Number of Clusters')
plt.tight_layout()
plt.grid(True)
#plt.savefig('images/11_03.png', dpi=300)
plt.show()
```


Distortion vs. Number of Clusters
Distortion
Number of clusters