Assignment\_5\_2\_Raghuwanshi\_Prashant\_DSC550

October 2, 2021

# Assignment: 2.2 Exercise: Graph Analysis Name: Prashant Raghuwanshi

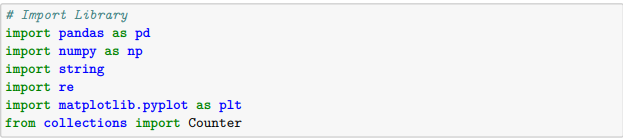
**Date: 10/02/2021**

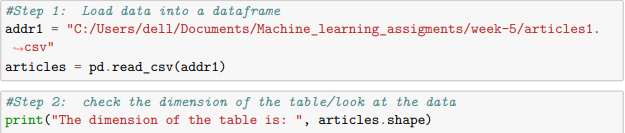
**Course: DSC550-T301 Data Mining (2221-1)**

#Case Study: Testing Hypothesis

#Hypothesis: Articles about Climate Change are more likely to be published by "Liberal" sources

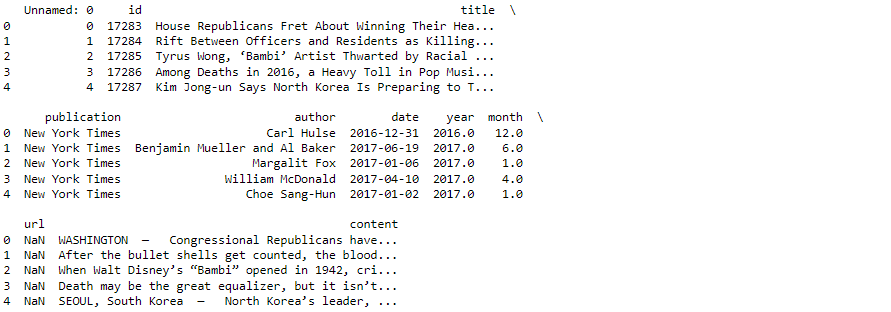
NOTE: This case study is not complete! We are only using the first part of it to practice Graphic Analytics.







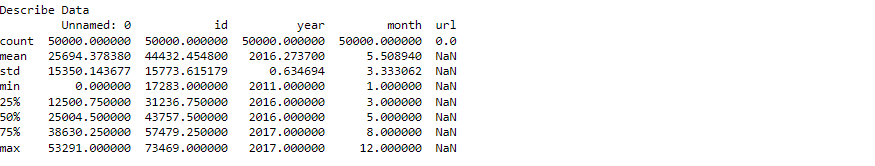




#what type of variables are in the table

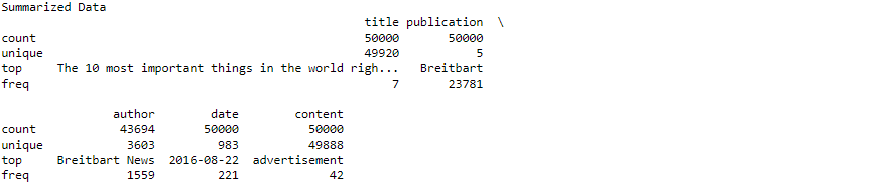
print("Describe Data")

print(articles.describe())



print("Summarized Data")

print(articles.describe(include=['O']))



#display length of data

print(len(articles))



#display publishers (publications)

print(articles.publication.unique())



#display min, max of years published

print(articles['year'].min())

print(articles['year'].max())



#display how many articles from each year

print(articles['year'].value\_counts())



#Step 3: Create some bar charts to show articles

#display bar chart of articles sorted by Publication Name

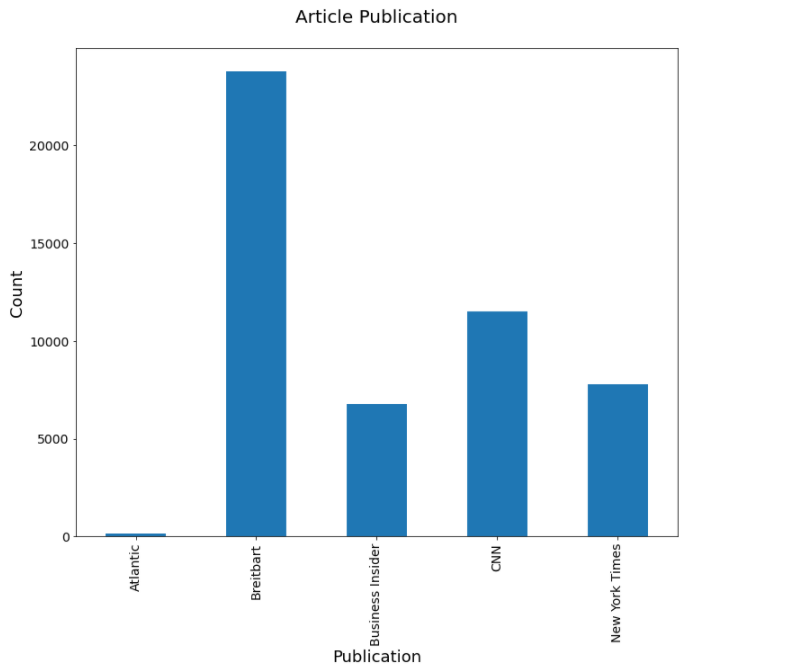
ax = articles['publication'].value\_counts().sort\_index().plot(kind='bar', fontsize=14, figsize=(12,10))

ax.set\_title('Article Publication\n', fontsize=20)

ax.set\_xlabel('Publication', fontsize=18)

ax.set\_ylabel('Count', fontsize=18);

plt.show()



#display bar chart of articles sorted by counts

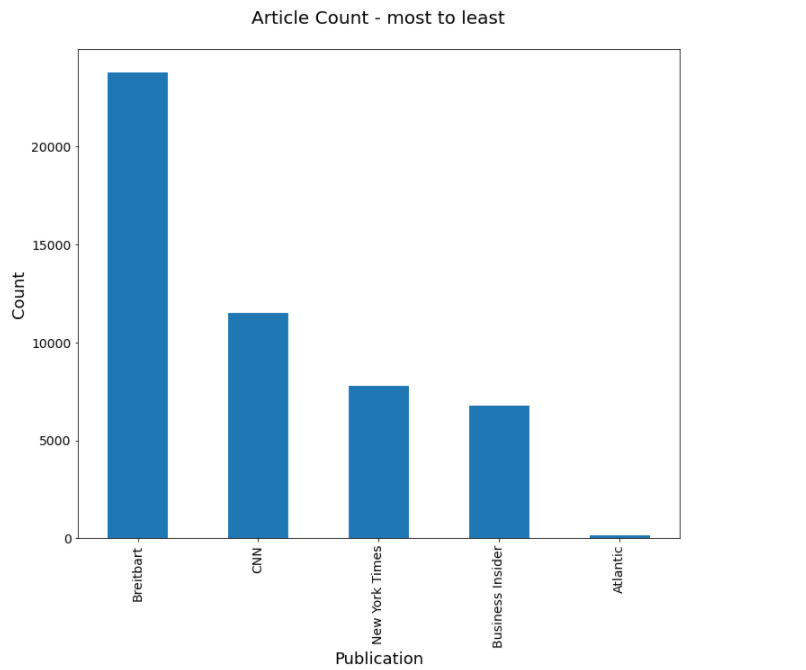
ax = articles['publication'].value\_counts().plot(kind='bar', fontsize=14, figsize=(12,10))

ax.set\_title('Article Count - most to least\n', fontsize=20)

ax.set\_xlabel('Publication', fontsize=18)

ax.set\_ylabel('Count', fontsize=18);

plt.show()



#Step 4: clean text: no punctuation/all lowercase

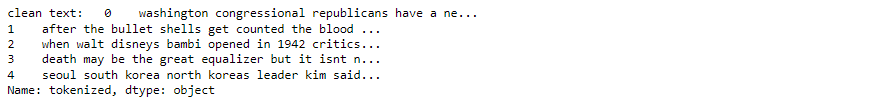
def clean\_text(article):

clean1 = re.sub(r'['+string.punctuation + '’—”'+']', "", article.lower())

return re.sub(r'\W+', ' ', clean1)

articles['tokenized'] = articles['content'].map(lambda x: clean\_text(x))

print("clean text: ",articles['tokenized'].head())



#look at mean, min, max article lengths

articles['num\_wds'] = articles['tokenized'].apply(lambda x: len(x.split()))

print("Mean: ",articles['num\_wds'].mean())

print("Min: ",articles['num\_wds'].min())

print("Max: ",articles['num\_wds'].max())



#Step 5: remove articles with no words

len(articles[articles['num\_wds']==0])

articles = articles[articles['num\_wds']>0]

print("new mean: ",articles['num\_wds'].mean())

print("new min: ",articles['num\_wds'].min())



#Step 6: Check for Outliers: show bar graph of outliers

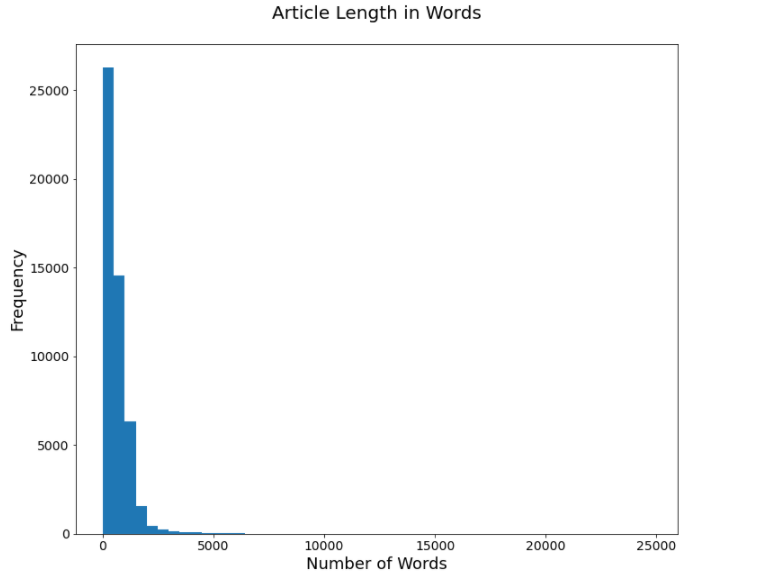
ax=articles['num\_wds'].plot(kind='hist', bins=50, fontsize=14, figsize=(12,10))

ax.set\_title('Article Length in Words\n', fontsize=20)

ax.set\_ylabel('Frequency', fontsize=18)

ax.set\_xlabel('Number of Words', fontsize=18);

plt.show()



What you learned last week with the MatLabLib tutorials, and what you learned from this tutorial, be thinking about an original Analysis Case Study that you will be developing throughout the rest of this course.

The first part of your original Analysis Case Study (Graph Analysis) will be part of your assignment for next week but you can start looking for ideas this week.,

Visualization feature space is not as easy when our data is text.

This is in apart because visualizing high-dimensional data is inherently more difficult but also because visualizing text data in python requires additional logic as compared to plotting purely numeric data.

By using Mataplotlib we can visualize feature analysis and feature engineering.

Feature analysis techniques:

n-gram time series, network analysis, and projection plots

Feature engineering techniques:

visual parts of speech tagging and frequency distributions

Model Diagnostics:

We will select and compare multiple models and will determine which model will perform well or poorly.

Model diagnostic techniques : Visualizing clusters, visualizing classes, confusion matrices, visual steering

Graphs are used to represent and model complex system in the real word such as communication network and biological ecosystems

## Week 5 Sample code #9.1 to 9.4

import pandas as pd

import numpy as np

import json

import sys

import warnings

from sklearn.datasets import make\_regression

from sklearn.feature\_selection import RFECV

from sklearn import datasets, linear\_model

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

from sklearn.decomposition import NMF

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

**#9.1 reducing features using Principal Components**

digits = datasets.load\_digits()

features= StandardScaler().fit\_transform(digits.data)

pca=PCA(n\_components=0.99, whiten=True)

features\_pca = pca.fit\_transform(features)

print("original number of features:", features.shape[1])

print("reduced number of features:", features\_pca.shape[1])

print("output from 9.1 done!")



#**9.4 Reducing Features Using Matrix Factorization**

features = digits.data

nmf=NMF(n\_components=10, random\_state=1)

features\_nmf=nmf.fit\_transform(features)

print("Original number of features:", features.shape[1])

print("reduced number of features:", features\_nmf.shape[1])

print("output from 9.4 done!")



