

# Homework 1 (Due: Mar. 17, 2021 (11:59 PM))

Name: Yeo hyuksoo

Student ID: 2016312761

Late submission Days used here: 0 (if there is, please modify here)

[Please SUBMIT (1) YOUR IPYNB AND (2) PDF (please use FILE/DOWNLOAD AS/PDF or PRINT PREVIEW/PRINT AS PDF with "printed output") TO iCampus]

For this homework you cannot use the python library scikit-learn (sklearn). You can use the python package BeautifulSoup to parse web pages.

In this assignment you will retrieve and parse webpages. The text file "urls.txt" contains a list of urls for the webpages to be parsed. Each line in the text file corresponds to a url. Use BeautifulSoup to fetch each webpage.

Note: For all questions, the words should be converted to lower case.

## Q1: Part 1 (5)

Parse the first webpage document to retrieve the text enclosed in paragraph tags, find the words that end in "ing", and count how many times each word appears.

Sort these words in decreasing order of frequency, write the words (along with their corresponding frequencies) in (1) this Notebook and (2) an output file named "Q1\_Part1.txt". The most frequent word should appear at the top and the least frequent word at the end, and the format of the output file should be: word TAB frequency

If two words have the same frequency, use the alphabetical order of words for outputs.

Example:

sorting 10 training 8 broadening 6 extracting 3 evergrowing 2 coming 1

In [1]:

```

### YOUR CODE HERE
from bs4 import BeautifulSoup
import urllib
import re

r = urllib.request.urlopen('https://en.wikipedia.org/wiki/Data_science').read()
soup = BeautifulSoup(r)

word_dict={}
words1 = []
words2 = []
words3 = []
words4 = []
words5 = []

paragraphs = soup.find_all('p')
for p in paragraphs:
    pText = p.get_text().split(' ')
    for w in pText:
        w = w.lower()
        if 'ing' in w:
            if len(w)>w.index('ing')+3 and w[w.index('ing')+3].isalpha():
                continue
            pattern = re.compile(r"Ww*ing")
            target = re.findall(pattern, w)
            if target:
                target = str(target)[2:len(target)-3]
                if target not in word_dict:
                    word_dict[target] = 1
                else:
                    word_dict[target] += 1

word_dict = sorted(word_dict.items(), key=lambda x: x[1], reverse=True)
for key, value in word_dict:
    if value == 1:
        words1.append(key)
    elif value == 2:
        words2.append(key)
    elif value == 3:
        words3.append(key)
    elif value == 4:
        words4.append(key)
    else:
        words5.append(key)

words1.sort()
words2.sort()
words3.sort()
words4.sort()
words5.sort()

f = open('Q1_Part1.txt', 'w')

for w in words5:
    res = w+' 5 '
    f.write(res)
    print(res, end="")

for w in words4:

```

```
res = wt+ ' 4 '
f.write(res)
print(res, end=" ")

for w in words3:
    res = wt+ ' 3 '
    f.write(res)
    print(res, end=" ")

for w in words2:
    res = wt+ ' 2 '
    f.write(res)
    print(res, end=" ")

for w in words1:
    res = wt+ ' 1 '
    f.write(res)
    print(res, end=" ")

f.close()
```

learning 5 mining 3 beijing 2 computing 2 accounting 1 analyzing 1 applying 1 becoming 1 being 1 breaking 1 changing 1 combining 1 creating 1 describing 1 developing 1 drawing 1 during 1 emerging 1 enabling 1 everything 1 extracting 1 finding 1 formulating 1 growing 1 including 1 managing 1 preparing 1 presenting 1 reflecting 1 training 1 turing 1

## Q1: Part 2 (5)

Stop words are natural language words which have very little meaning, such as "and", "the", "a", "an", and similar words.

Repeat Part 1, but before counting, remove the stop words given in the file "stop\_words.txt". The output for Part 2 should have the same format as Part 1, and should be written to (1) this Notebook and (2) an output file named "Q1\_Part2.txt".

In [2]:

```

### YOUR CODE HERE
from bs4 import BeautifulSoup
import urllib
import re

r = urllib.request.urlopen('https://en.wikipedia.org/wiki/Data_science').read()
soup = BeautifulSoup(r)

word_dict={}
words1 = []
words2 = []
words3 = []
words4 = []
words5 = []

f = open('stop_words.txt', 'r')
stop_list = f.readlines()
for i in range(0, len(stop_list), 1):
    stop_list[i] = stop_list[i][:len(stop_list[i])-1]
f.close()

paragraphs = soup.find_all('p')
for p in paragraphs:
    pText = p.get_text().split(' ')
    for w in pText:
        w = w.lower()
        if 'ing' in w:
            if len(w)>w.index('ing')+3 and w[w.index('ing')+3].isalpha():
                continue
            pattern = re.compile(r"Ww*ing")
            target = re.findall(pattern, w)
            if target:
                target = str(target)[2:len(target)-3]
                if target not in stop_list:
                    if target not in word_dict:
                        word_dict[target] = 1
                    else:
                        word_dict[target] += 1

word_dict = sorted(word_dict.items(), key=lambda x: x[1], reverse=True)
for key, value in word_dict:
    if value == 1:
        words1.append(key)
    elif value == 2:
        words2.append(key)
    elif value == 3:
        words3.append(key)
    elif value == 4:
        words4.append(key)
    else:
        words5.append(key)

words1.sort()
words2.sort()
words3.sort()
words4.sort()
words5.sort()

f = open('Q1_Part2.txt', 'w')

```

```

for w in words5:
    res = w+' 5 '
    f.write(res)
    print(res, end=" ")

for w in words4:
    res = w+' 4 '
    f.write(res)
    print(res, end=" ")

for w in words3:
    res = w+' 3 '
    f.write(res)
    print(res, end=" ")

for w in words2:
    res = w+' 2 '
    f.write(res)
    print(res, end=" ")

for w in words1:
    res = w+' 1 '
    f.write(res)
    print(res, end=" ")

f.close()

```

learning 5 mining 3 beijing 2 computing 2 accounting 1 analyzing 1 applying 1 breaking 1 changing 1 combining 1 creating 1 describing 1 developing 1 drawing 1 emerging 1 enabling 1 extracting 1 finding 1 formulating 1 growing 1 including 1 managing 1 preparing 1 presenting 1 reflecting 1 training 1 turing 1

## Q2 (10)

Again, parse the first webpage document, but this time find and count all outgoing links to other webpages, and write the output to (1) this Notebook and (2) a file named "Q2.txt", with each url on a new line.

The format of the output file should: number of outgoing urls in the first line, followed by each url on a new line. For example, 4 <https://eng.skku.edu/eng/edu/education.do> (<https://eng.skku.edu/eng/edu/education.do>) <https://eng.skku.edu/eng/Research/industry/researchStory.do> (<https://eng.skku.edu/eng/Research/industry/researchStory.do>) <https://eng.skku.edu/eng/Univ-Industry/Research-Business-Found/FactsandFigures.do> (<https://eng.skku.edu/eng/Univ-Industry/Research-Business-Found/FactsandFigures.do>) <https://eng.skku.edu/eng/CampusLife/support/employment.do> (<https://eng.skku.edu/eng/CampusLife/support/employment.do>).

In [16]:

```
### YOUR CODE HERE
from bs4 import BeautifulSoup
import urllib
import re

r = urllib.request.urlopen('https://en.wikipedia.org/wiki/Data_science').read()
soup = BeautifulSoup(r)

links = soup.find_all('a')
url_list = []
num = 0

for link in links:
    linkUrl = str(link.get('href'))
    if linkUrl != 'None' and linkUrl[0] != '#':
        url_list.append(linkUrl)
        num+=1

f = open('Q2.txt', 'w')

f.write(str(num)+'\n')
print(num)
for linkUrl in url_list:
    f.write(linkUrl+'\n')
    print(linkUrl)

f.close()
```

417

/wiki/Information\_science  
/wiki/File:PIA23792-1600x1200(1).jpg  
/wiki/File:PIA23792-1600x1200(1).jpg  
/wiki/Comet\_NEOWISE  
/wiki/Astronomical\_survey  
/wiki/Space\_telescope  
/wiki/Wide-field\_Infrared\_Survey\_Explorer  
/wiki/Machine\_learning  
/wiki/Data\_mining  
/wiki/File:Multi-Layer\_Neural\_Network-Vector-Blank.svg  
/wiki/Statistical\_classification  
/wiki/Cluster\_analysis  
/wiki/Regression\_analysis  
/wiki/Anomaly\_detection  
/wiki/Automated\_machine\_learning  
/wiki/Association\_rule\_learning  
/wiki/Reinforcement\_learning  
/wiki/Structured\_prediction  
/wiki/Feature\_engineering  
/wiki/Feature\_learning  
/wiki/Online\_machine\_learning  
/wiki/Semi-supervised\_learning  
/wiki/Unsupervised\_learning  
/wiki/Learning\_to\_rank  
/wiki/Grammar\_induction  
/wiki/Supervised\_learning  
/wiki/Statistical\_classification  
/wiki/Regression\_analysis  
/wiki/Decision\_tree\_learning  
/wiki/Ensemble\_learning  
/wiki/Bootstrap\_aggregating  
/wiki/Boosting\_(machine\_learning)  
/wiki/Random\_forest  
/wiki/K-nearest\_neighbors\_algorithm  
/wiki/Linear\_regression  
/wiki/Naive\_Bayes\_classifier  
/wiki/Artificial\_neural\_network  
/wiki/Logistic\_regression  
/wiki/Perceptron  
/wiki/Relevance\_vector\_machine  
/wiki/Support-vector\_machine  
/wiki/Cluster\_analysis  
/wiki/BIRCH  
/wiki/CURE\_data\_clustering\_algorithm  
/wiki/Hierarchical\_clustering  
/wiki/K-means\_clustering  
/wiki/Expectation%E2%80%93maximization\_algorithm  
/wiki/DBSCAN  
/wiki/OPTICS\_algorithm  
/wiki/Mean-shift  
/wiki/Dimensionality\_reduction  
/wiki/Factor\_analysis  
/wiki/Canonical\_correlation  
/wiki/Independent\_component\_analysis  
/wiki/Linear\_discriminant\_analysis  
/wiki/Non-negative\_matrix\_factorization  
/wiki/Principal\_component\_analysis  
/wiki/Proper\_generalized\_decomposition  
/wiki/T-distributed\_stochastic\_neighbor\_embedding  
/wiki/Structured\_prediction

/wiki/Graphical\_model  
/wiki/Bayesian\_network  
/wiki/Conditional\_random\_field  
/wiki/Hidden\_Markov\_model  
/wiki/Anomaly\_detection  
/wiki/K-nearest\_neighbors\_classification  
/wiki/Local\_outlier\_factor  
/wiki/Artificial\_neural\_network  
/wiki/Autoencoder  
/wiki/Cognitive\_computing  
/wiki/Deep\_learning  
/wiki/DeepDream  
/wiki/Multilayer\_perceptron  
/wiki/Recurrent\_neural\_network  
/wiki/Long\_short-term\_memory  
/wiki/Gated\_recurrent\_unit  
/wiki/Echo\_state\_network  
/wiki/Restricted\_Boltzmann\_machine  
/wiki/Generative\_adversarial\_network  
/wiki/Self-organizing\_map  
/wiki/Convolutional\_neural\_network  
/wiki/U-Net  
/wiki/Transformer\_(machine\_learning\_model)  
/wiki/Spiking\_neural\_network  
/wiki/Memtransistor  
/wiki/Electrochemical\_RAM  
/wiki/Reinforcement\_learning  
/wiki/Q-learning  
/wiki/State%E2%80%93action%E2%80%93reward%E2%80%93state%E2%80%93action  
/wiki/Temporal\_difference\_learning  
/wiki/Bias%E2%80%93variance\_tradeoff  
/wiki/Computational\_learning\_theory  
/wiki/Empirical\_risk\_minimization  
/wiki/Occam\_learning  
/wiki/Probably\_approximately\_correct\_learning  
/wiki/Statistical\_learning\_theory  
/wiki/Vapnik%E2%80%93Chervonenkis\_theory  
/wiki/Conference\_on\_Neural\_Information\_Processing\_Systems  
/wiki/International\_Conference\_on\_Machine\_Learning  
/wiki/Machine\_Learning\_(journal)  
/wiki/Journal\_of\_Machine\_Learning\_Research  
<https://arxiv.org/list/cs.LG/recent>  
/wiki/Glossary\_of\_artificial\_intelligence  
/wiki/Glossary\_of\_artificial\_intelligence  
/wiki/List\_of\_datasets\_for\_machine-learning\_research  
/wiki/Outline\_of\_machine\_learning  
/wiki/Template:Machine\_learning\_bar  
/wiki/Template\_talk:Machine\_learning\_bar  
[https://en.wikipedia.org/w/index.php?title=Template:Machine\\_learning\\_bar&action=edit](https://en.wikipedia.org/w/index.php?title=Template:Machine_learning_bar&action=edit)  
/wiki/Inter-disciplinary  
/wiki/Knowledge  
/wiki/Unstructured\_data  
/wiki/Data\_mining  
/wiki/Machine\_learning  
/wiki/Big\_data  
/wiki/Statistics  
/wiki/Data\_analysis  
/wiki/Informatics  
/wiki/Mathematics  
/wiki/Statistics



/wiki/Computer\_science  
/wiki/Information\_science  
/wiki/Domain\_knowledge  
/wiki/Turing\_award  
/wiki/Jim\_Gray\_(computer\_scientist)  
/wiki/Empirical\_research  
/wiki/Basic\_research  
/wiki/Computational\_science  
/wiki/Information\_technology  
/wiki/Information\_explosion  
/w/index.php?title=Data\_science&action=edit&section=1  
/wiki/Big\_data  
/wiki/Information\_visualization  
/wiki/Complex\_systems  
/wiki/Communication  
/wiki/Nathan\_Yau  
/wiki/Ben\_Fry  
/wiki/Human%E2%80%93computer\_interaction  
/wiki/American\_Statistical\_Association  
/wiki/Database  
/wiki/Machine\_learning  
/wiki/Distributed\_computing  
/w/index.php?title=Data\_science&action=edit&section=2  
/wiki/Nate\_Silver  
/wiki/Vasant\_Dhar  
/wiki/Andrew\_Gelman  
/wiki/Columbia\_University  
/wiki/David\_Donoho  
/w/index.php?title=Data\_science&action=edit&section=3  
/w/index.php?title=Data\_science&action=edit&section=4  
/wiki/John\_Tukey  
/wiki/C.F.\_Jeff\_Wu  
/wiki/Data\_Science  
/wiki/Montpellier\_2\_University  
/wiki/Peter\_Naur  
/wiki/C.F.\_Jeff\_Wu  
/w/index.php?title=Data\_science&action=edit&section=5  
/wiki/William\_S.\_Cleveland  
/wiki/Committee\_on\_Data\_for\_Science\_and\_Technology  
/wiki/American\_Statistical\_Association  
/wiki/DJ\_Patil  
/wiki/Jeff\_Hammerbacher  
/wiki/National\_Science\_Board  
/w/index.php?title=Data\_science&action=edit&section=6  
/w/index.php?title=Data\_science&action=edit&section=7  
/wiki/Wikipedia:LSC  
/wiki/Wikipedia:Manual\_of\_Style/Stand-alone\_lists  
https://en.wikipedia.org/w/index.php?title=Data\_science&action=edit  
/wiki/Talk:Data\_science  
/w/index.php?title=Data\_science&action=edit&section=8  
/wiki/Statistics#Methods  
/wiki/Linear\_regression  
/wiki/Logistic\_regression  
/wiki/Decision\_tree  
/wiki/Support\_vector\_machine  
/wiki/Cluster\_analysis  
/wiki/Dimensionality\_reduction  
/wiki/Machine\_learning  
/w/index.php?title=Data\_science&action=edit&section=9  
/wiki/Python\_(programming\_language)  
/wiki/R\_(programming\_language)

[/wiki/Computation](#)  
[/wiki/Julia\\_\(programming\\_language\)](#)  
[/w/index.php?title=Data\\_science&action=edit&section=10](#)  
[/wiki/TensorFlow](#)  
[/wiki/Pytorch](#)  
[/wiki/Jupyter\\_Notebook](#)  
[/wiki/Apache\\_Hadoop](#)  
[/w/index.php?title=Data\\_science&action=edit&section=11](#)  
[/wiki/Data\\_visualization](#)  
[/wiki/Plotly](#)  
[/wiki/Tableau\\_Software](#)  
[/wiki/Microsoft\\_Power\\_BI](#)  
[/wiki/Qlik](#)  
[/wiki/AnyChart](#)  
[/wiki/Google\\_Charts](#)  
[/wiki/Sisense](#)  
[/wiki/Webix](#)  
[/w/index.php?title=Data\\_science&action=edit&section=12](#)  
[/wiki/RapidMiner](#)  
[/wiki/Dataiku](#)  
[/wiki/Anaconda\\_\(Python\\_distribution\)](#)  
[/wiki/MATLAB](#)  
[/wiki/Databricks](#)  
[/wiki/IBM\\_Watson\\_Studio](#)  
[/w/index.php?title=Data\\_science&action=edit&section=13](#)  
<http://cacm.acm.org/magazines/2013/12/169933-data-science-and-prediction/fulltext>  
[/wiki/Doi\\_\(identifier\)](#)  
<https://doi.org/10.1145%2F2500499>  
[/wiki/S2CID\\_\(identifier\)](#)  
<https://api.semanticscholar.org/CorpusID:6107147>  
<https://web.archive.org/web/20141109113411/http://cacm.acm.org/magazines/2013/12/169933-data-science-and-prediction/fulltext>  
[/wiki/Jeffrey\\_T.\\_Leek](#)  
<http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/>  
<https://web.archive.org/web/20140102194117/http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/>  
<https://www.springer.com/book/9784431702085>  
[/wiki/Doi\\_\(identifier\)](#)  
[https://doi.org/10.1007%2F978-4-431-65950-1\\_3](https://doi.org/10.1007%2F978-4-431-65950-1_3)  
[/wiki/ISBN\\_\(identifier\)](#)  
[/wiki/Special:BookSources/9784431702085](#)  
[https://books.google.com/books?id=oGs\\_AQAAIAAJ](https://books.google.com/books?id=oGs_AQAAIAAJ)  
[/wiki/ISBN\\_\(identifier\)](#)  
[/wiki/Special:BookSources/978-0-9825442-0-4](#)  
[https://web.archive.org/web/20170320193019/https://books.google.com/books?id=oGs\\_AQAAIAAJ](https://web.archive.org/web/20170320193019/https://books.google.com/books?id=oGs_AQAAIAAJ)  
[/wiki/Doi\\_\(identifier\)](#)  
<https://doi.org/10.1126%2Fscience.1170411>  
[/wiki/ISSN\\_\(identifier\)](#)  
<http://www.worldcat.org/issn/0036-8075>  
[/wiki/PMID\\_\(identifier\)](#)  
<http://pubmed.ncbi.nlm.nih.gov/19265007>  
[/wiki/S2CID\\_\(identifier\)](#)  
<https://api.semanticscholar.org/CorpusID:9743327>  
<http://www.datascienceassn.org/about-data-science>  
<https://www.oreilly.com/library/view/doing-data-science/9781449363871/ch01.html>  
<https://medriscoll.com/post/4740157098/the-three-sexy-skills-of-data-geeks>  
<https://flowingdata.com/2009/06/04/rise-of-the-data-scientist/>  
<https://benfry.com/phd/dissertation/2.html>  
<https://magazine.amstat.org/blog/2015/10/01/asa-statement-on-the-role-of-statistic>

s-in-data-science/  
/wiki/American\_Statistical\_Association  
<https://web.archive.org/web/20190620184935/https://magazine.amstat.org/blog/2015/10/01/asa-statement-on-the-role-of-statistics-in-data-science/>  
<https://www.statisticviews.com/article/nate-silver-what-i-need-from-statisticians/>  
<http://priceonomics.com/whats-the-difference-between-data-science-and/>  
/wiki/Doi\_(identifier)  
<https://doi.org/10.1145%2F2500499>  
/wiki/S2CID\_(identifier)  
<https://api.semanticscholar.org/CorpusID:6107147>  
<https://statmodeling.stat.columbia.edu/2013/11/14/statistics-least-important-part-data-science/>  
<https://www.datasciencecentral.com/profiles/blogs/data-science-without-statistics-is-possible-even-desirable>  
<http://courses.csail.mit.edu/18.337/2015/docs/50YearsDataScience.pdf>  
<https://www2.isye.gatech.edu/~jeffwu/publications/fazhan.pdf>  
/wiki/ISBN\_(identifier)  
/wiki/Special:BookSources/0-12-241770-4  
/wiki/OCLC\_(identifier)  
<http://www.worldcat.org/oclc/489990740>  
/wiki/Category:CS1\_maint:\_others  
<https://www.mdpi.com/2504-2289/2/2/14>  
/wiki/Doi\_(identifier)  
<https://doi.org/10.3390%2Fbdc2020014>  
<https://doi.org/10.1145%2F3076253>  
/wiki/Doi\_(identifier)  
<https://doi.org/10.1145%2F3076253>  
<http://www2.isye.gatech.edu/~jeffwu/presentations/datascience.pdf>  
<https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/>  
<https://www.stat.purdue.edu/~wsc/>  
<https://magazine.amstat.org/blog/2016/06/01/datascience-2/>  
<https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>  
/wiki/ISSN\_(identifier)  
<http://www.worldcat.org/issn/0017-8012>  
<https://www.nsf.gov/pubs/2005/nsb0540/>  
<https://www.forbes.com/sites/gilpress/2013/08/19/data-science-whats-the-half-life-of-a-buzzword/>  
<https://www.forbes.com/sites/peterpham/2015/08/28/the-impacts-of-big-data-that-you-may-not-have-heard-of/>  
<https://towardsdatascience.com/how-data-science-will-impact-future-of-businesses-7f11f5699c4d>  
<https://sites.engineering.ucsb.edu/~shell/che210d/python.pdf>  
[https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R\\_003f](https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f)  
<https://www.wired.com/2014/07/a-drag-and-drop-toolkit-that-lets-anyone-create-interactive-maps/>  
/wiki/Template:Data  
/wiki/Template\_talk:Data  
<https://en.wikipedia.org/w/index.php?title=Template:Data&action=edit>  
/wiki/Data\_(computing)  
/wiki/Data\_augmentation  
/wiki/Data\_analysis  
/wiki/Data\_archaeology  
/wiki/Data\_cleansing  
/wiki/Data\_collection  
/wiki/Data\_compression  
/wiki/Data\_corruption  
/wiki/Data\_curation  
/wiki/Data\_degradation  
/wiki/Data\_editing

/wiki/Extract,\_transform,\_load  
/wiki/Data\_extraction  
/wiki/Data\_transformation  
/wiki/Data\_loading  
/wiki/Data\_farming  
/wiki/Data\_format\_management  
/wiki/Data\_fusion  
/wiki/Data\_integration  
/wiki/Data\_integrity  
/wiki/Data\_library  
/wiki/Data\_loss  
/wiki/Data\_management  
/wiki/Data\_migration  
/wiki/Data\_mining  
/wiki/Data\_pre-processing  
/wiki/Data\_preservation  
/wiki/Information\_privacy  
/wiki/Data\_recovery  
/wiki/Data\_reduction  
/wiki/Data\_retention  
/wiki/Data\_quality  
/wiki/Data\_scraping  
/wiki/Data\_scrubbing  
/wiki/Data\_security  
/wiki/Data\_steward  
/wiki/Data\_storage  
/wiki/Data\_validation  
/wiki/Data\_warehouse  
/wiki/Data\_wrangling  
[https://en.wikipedia.org/w/index.php?title=Data\\_science&oldid=1011117061](https://en.wikipedia.org/w/index.php?title=Data_science&oldid=1011117061)  
/wiki/Help:Category  
/wiki/Category:Information\_science  
/wiki/Category:Computer\_occupations  
/wiki/Category:Computational\_fields\_of\_study  
/wiki/Category:Data\_analysis  
/wiki/Category:CS1\_maint:\_others  
/wiki/Category:Articles\_with\_short\_description  
/wiki/Category:Short\_description\_matches\_Wikidata  
/wiki/Category:Use\_dmy\_dates\_from\_December\_2012  
/wiki/Category:Lists\_having\_no\_precise\_inclusion\_criteria\_from\_June\_2020  
/wiki/Category:All\_lists\_having\_no\_precise\_inclusion\_criteria  
/wiki/Special:MyTalk  
/wiki/Special:MyContributions  
/w/index.php?title=Special:CreateAccount&returnto=Data+science  
/w/index.php?title=Special:UserLogin&returnto=Data+science  
/wiki/Data\_science  
/wiki/Talk:Data\_science  
/wiki/Data\_science  
/w/index.php?title=Data\_science&action=edit  
/w/index.php?title=Data\_science&action=history  
/wiki/Main\_Page  
/wiki/Main\_Page  
/wiki/Wikipedia:Contents  
/wiki/Portal:Current\_events  
/wiki/Special:Random  
/wiki/Wikipedia:About  
//en.wikipedia.org/wiki/Wikipedia:Contact\_us  
[https://donate.wikimedia.org/wiki/Special:FundraiserRedirector?utm\\_source=donate&utm\\_medium=sidebar&utm\\_campaign=C13\\_en.wikipedia.org&uselang=en](https://donate.wikimedia.org/wiki/Special:FundraiserRedirector?utm_source=donate&utm_medium=sidebar&utm_campaign=C13_en.wikipedia.org&uselang=en)  
/wiki/Help:Contents  
/wiki/Help:Introduction

/wiki/Wikipedia:Community\_portal  
/wiki/Special:RecentChanges  
/wiki/Wikipedia:File\_Upload\_Wizard  
/wiki/Special:WhatLinksHere/Data\_science  
/wiki/Special:RecentChangesLinked/Data\_science  
/wiki/Wikipedia:File\_Upload\_Wizard  
/wiki/Special:SpecialPages  
/w/index.php?title=Data\_science&oldid=1011117061  
/w/index.php?title=Data\_science&action=info  
/w/index.php?title=Special:CiteThisPage&page=Data\_science&id=1011117061&wpFormIdentifier=titleform  
https://www.wikidata.org/wiki/Special:EntityPage/Q2374463  
/w/index.php?title=Special:DownloadAsPdf&page=Data\_science&action=show-download-screen  
/w/index.php?title=Data\_science&printable=yes  
https://commons.wikimedia.org/wiki/Category:Data\_science  
https://ar.wikipedia.org/wiki/%D8%B9%D9%84%D9%85\_%D8%A7%D9%84%D8%A8%D9%8A%D8%A7%D9%86%D8%A7%D8%AA  
https://az.wikipedia.org/wiki/Veril%C9%99nl%C9%99r\_elmi\_(Data\_Science)  
https://bn.wikipedia.org/wiki/%E0%A6%89%E0%A6%AA%E0%A6%BE%E0%A6%A4%E0%A7%8D%E0%A6%A4\_%E0%A6%AC%E0%A6%BF%E0%A6%9C%E0%A7%8D%E0%A6%9E%E0%A6%BE%E0%A6%A8  
https://ca.wikipedia.org/wiki/Ci%C3%A8ncia\_de\_les\_dades  
https://cs.wikipedia.org/wiki/Data\_science  
https://de.wikipedia.org/wiki/Data\_Science  
https://et.wikipedia.org/wiki/Andmeteadus  
https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B9%CF%83%CF%84%CE%AE%CE%BC%CE%B7\_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD  
https://es.wikipedia.org/wiki/Ciencia\_de\_datos  
https://eu.wikipedia.org/wiki/Datu\_zientzia  
https://fa.wikipedia.org/wiki/%D8%B9%D9%84%D9%85\_%D8%AF%D8%A7%D8%AF%D9%87%E2%80%8C%D9%87%D8%A7  
https://fr.wikipedia.org/wiki/Science\_des\_donn%C3%A9es  
https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0\_%EC%82%AC%EC%9D%B4%EC%96%B8%EC%8A%A4  
https://hy.wikipedia.org/wiki/%D5%8F%D5%BE%D5%B5%D5%A1%D5%AC%D5%B6%D5%A5%D6%80%D5%AB\_%D5%A3%D5%AB%D5%BF%D5%B8%D6%82%D5%A9%D5%B5%D5%B8%D6%82%D5%B6  
https://hi.wikipedia.org/wiki/%E0%A4%86%E0%A4%81%E0%A4%95%E0%A4%A1%E0%A4%BC%E0%A4%BE\_%E0%A4%B5%E0%A4%BF%E0%A4%9C%E0%A5%8D%E0%A4%9E%E0%A4%BE%E0%A4%A8  
https://id.wikipedia.org/wiki/Ilmu\_data  
https://it.wikipedia.org/wiki/Scienza\_dei\_dati  
https://he.wikipedia.org/wiki/%D7%9E%D7%93%D7%A2\_%D7%94%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D  
https://kk.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%BA%D1%82%D0%B5%D1%80\_%D1%82%D1%83%D1%80%D0%B0%D0%BB%D1%8B\_%D2%93%D1%8B%D0%BB%D1%8B%D0%BC  
https://lv.wikipedia.org/wiki/Datu\_m%C4%81c%C4%ABba  
https://mk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0\_%D0%B7%D0%B0\_%D0%BF%D0%BE%D0%B4%D0%B0%D1%82%D0%BE%D1%86%D0%B8  
https://ms.wikipedia.org/wiki/Sains\_data  
https://my.wikipedia.org/wiki/%E1%80%A1%E1%80%81%E1%80%BB%E1%80%80%E1%80%BA%E1%80%A1%E1%80%9C%E1%80%80%E1%80%BA%E1%80%9E%E1%80%AD%E1%80%95%E1%80%B9%E1%80%95%E1%80%B6%E1%80%95%E1%80%8A%E1%80%AC  
https://nl.wikipedia.org/wiki/Datawetenschap  
https://ja.wikipedia.org/wiki/%E3%83%87%E3%83%BC%E3%82%BF%E3%82%B5%E3%82%A4%E3%82%A8%E3%83%B3%E3%82%B9  
https://pl.wikipedia.org/wiki/Danologia  
https://pt.wikipedia.org/wiki/Ci%C3%A2ncia\_de\_dados  
https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0\_%D0%BE\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85  
https://simple.wikipedia.org/wiki/Data\_science  
https://fi.wikipedia.org/wiki/Datatiede  
https://ta.wikipedia.org/wiki/%E0%AE%A4%E0%AE%B0%E0%AE%B5%E0%AF%81\_%E0%AE%85%E0%A

E%B1%E0%AE%BF%E0%AE%B5%E0%AE%BF%E0%AE%AF%E0%AE%B2%E0%AF%8D  
https://th.wikipedia.org/wiki/%E0%B8%A7%E0%B8%B4%E0%B8%97%E0%B8%A2%E0%B8%B2%E0%B8%  
81%E0%B8%B2%E0%B8%A3%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5  
https://tr.wikipedia.org/wiki/Veri\_bilimi  
https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0\_%D0%BF%D1%80%D0%BE\_%D  
0%B4%D0%B0%D0%BD%D1%96  
https://ur.wikipedia.org/wiki/%DA%88%DB%8C%D9%B9%D8%A7\_%D8%B3%D8%A7%D8%A6%D9%86%D  
8%B3  
https://vi.wikipedia.org/wiki/Khoa\_h%E1%BB%8Dc\_d%E1%BB%AF\_li%E1%BB%87u  
https://zh-yue.wikipedia.org/wiki/%E6%95%B8%E6%93%9A%E7%A7%91%E5%AD%B8  
https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E7%A7%91%E5%AD%A6  
https://www.wikidata.org/wiki/Special:EntityPage/Q2374463#sitelinks-wikipedia  
//en.wikipedia.org/wiki/Wikipedia:Text\_of\_Creative\_Commons\_Attribution-ShareAlike\_  
3.0\_Unported\_License  
//creativecommons.org/licenses/by-sa/3.0/  
//foundation.wikimedia.org/wiki/Terms\_of\_Use  
//foundation.wikimedia.org/wiki/Privacy\_policy  
//www.wikimediafoundation.org/  
https://foundation.wikimedia.org/wiki/Privacy\_policy  
/wiki/Wikipedia:About  
/wiki/Wikipedia:General\_disclaimer  
//en.wikipedia.org/wiki/Wikipedia:Contact\_us  
//en.m.wikipedia.org/w/index.php?title=Data\_science&mobileaction=toggle\_view\_mobi  
l  
https://www.mediawiki.org/wiki/Special:MyLanguage/How\_to\_contribute  
https://stats.wikimedia.org/#/en.wikipedia.org  
https://foundation.wikimedia.org/wiki/Cookie\_statement  
https://wikimediafoundation.org/  
https://www.mediawiki.org/

## Q3: Part 1 (10)

1. Retrieve and parse multiple web pages. The text file "urls.txt" contains a list of webpages to be parsed. Each line in the text file corresponds to a url. Use BeautifulSoup to fetch each webpage and parse it as specified below.
2. For each webpage document do the following:
  - A. Retrieve all text enclosed in paragraph tags.
  - B. Convert the text to lowercase.
  - C. Strip out punctuation. Note: if you use `translate()` with `string.punctuation`, then it may not strip out all characters. Use a regular expression involving `\W` to strip out all non alpha-numeric characters.
  - D. Tokenize into words based on whitespace separation.
3. Find the number of unique words in each webpage document.
4. Find the Length of each webpage document. The length of a document is defined as the total number of words in the document (not just unique words).
5. For each of the following words: "statistics", "analytics", "data", and "science", a. Find Term Frequency (tf). The term frequency (tf) of a term (word) is defined as the number of times that term  $t$  occurs in document  $d$ , divided by the total number of words in the document. The tf of a word depends on the document under consideration.
  - b. Find Inverse Document Frequency (idf). The inverse document frequency of a word is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that ratio. The idf of a word doesn't depend on any document in which the word is present. To calculate the idf, you will have to use the log function. The base for the log function must be  $e$ .
  - c. Find tf-idf. The tf-idf of a word is the product of the term frequency of the word in document  $d$ , and its inverse document frequency. The tf-idf of a word depends on the document under consideration.

Reference: <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>  
<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>

The output should be written to (1) this Notebook and (2) an output file named "Q3\_Part1.txt".

The format of the output file is as shown below:

1. Number of unique words in documents: [702, 723, 280]
2. Length of documents: [1711, 1928, 563]
3. tf statistics: [0.0070134424313267095, 0.0025933609958506223, 0.0] analytics:  
 [0.0029222676797194622, 0.0031120331950207467, 0.0] data: [0.056107539450613676,  
 0.05446058091286307, 0.0] science: [0.03798947983635301, 0.011410788381742738,  
 0.028419182948490232]
4. idf statistics: 0.510825623766 analytics: 0.510825623766 data: 0.223143551314 science: 0.0
5. tf-idf statistics: [0.0028437061936282715, 0.0010515173965460695, 0.0] analytics:  
 [0.0011848775806784465, 0.0012618208758552834, 0.0] data: [0.022749649549026172,  
 0.022081865327467459, 0.0] science: [0.0, 0.0, 0.0]

The above values are for the first three webpage urls given in the file "urls.txt". The number of unique words in documents, average length of documents, tf and tf-idf values for the four words, must be in the order of the urls given in "urls.txt".

In [9]:

```

### YOUR CODE HERE
from bs4 import BeautifulSoup
import urllib
import re
import math

f = open('urls.txt', 'r')
url = f.readline()
word_list = []
unique_words = 0
tf_words=["statistics", "analytics", "data", "science"]
tf_freq=[0, 0, 0, 0]
idf_docfreq=[0, 0, 0, 0]
idf_flag=[0, 0, 0, 0]

unique_words_res = []
length_res = []
tf_res = [[0 for col in range(5)] for row in range(4)]
idf_res = []
cnt = 0

while url:
    r = urllib.request.urlopen(url).read()
    soup = BeautifulSoup(r)
    paragraphs = soup.find_all('p')
    unique_words = 0
    words = 0
    word_list = []
    for i in range(0, 4, 1):
        idf_flag[i] = 0
    for p in paragraphs:
        pText = p.get_text()
        pText = pText.lower()
        pattern = re.compile(r"ww")
        for match in re.finditer(pattern, pText):
            if match.group() != ' ':
                pText = pText.replace(match.group(), ' ')
        pText = pText.split(' ')
        for w in pText:
            if w != '':
                words+=1
                if w not in word_list:
                    word_list.append(w)
                    unique_words+=1
                if w in tf_words:
                    for i in range(0, 4, 1):
                        if tf_words[i] == w:
                            tf_freq[i] += 1
                            if idf_flag[i] == 0:
                                idf_flag[i] = 1
                                idf_docfreq[i]+=1

    unique_words_res.append(unique_words)
    length_res.append(words)
    for i in range(0, 4, 1):
        tf_res[i][cnt] = tf_freq[i]/words
    cnt+=1

    url = f.readline()

```



```

for i in range(0, 4, 1):
    idf_res.append(math.log(5/idf_docfreq[i]))

f.close()
f = open('Q3_Part1.txt', 'w')
f.write("1. Number of unique words in documents: ")
f.write(str(unique_words_res)+'\n')
print("1. Number of unique words in documents:", unique_words_res)
f.write("2. Length of documents: ")
f.write(str(length_res)+'\n')
print("2. Length of documents:", length_res)
f.write("3. tf ")
print("3. tf ", end="")
for i in range(0, 4, 1):
    f.write(tf_words[i]+": ")
    f.write(str(tf_res[i])+' ')
    print(tf_words[i]+':', str(tf_res[i]), end=" ")

f.write("\n4. idf ")
print("\n4. idf ", end="")
for i in range(0, 4, 1):
    f.write(tf_words[i]+": ")
    f.write(str(idf_res[i])+' ')
    print(tf_words[i]+':', str(idf_res[i]), end=" ")

tf_idf_res = [[0 for col in range(5)] for row in range(4)]
for i in range(0, 4, 1):
    for j in range(0, 5, 1):
        tf_idf_res[i][j] = tf_res[i][j]*idf_res[i]
f.write("\n5. tf-idf ")
print("\n5. tf-idf", end=" ")
for i in range(0, 4, 1):
    f.write(tf_words[i]+": ")
    f.write(str(tf_idf_res[i])+' ')
    print(tf_words[i]+':', str(tf_idf_res[i]), end=" ")

f.close()

```

1. Number of unique words in documents: [473, 1975, 727, 930, 1170]  
 2. Length of documents: [1049, 6306, 1950, 2380, 3681]  
 3. tf statistics: [0.017159199237368923, 0.003171582619727244, 0.01282051282051282, 0.01050420168067227, 0.010051616408584624] analytics: [0.0009532888465204957, 0.0030130034887408817, 0.01282051282051282, 0.010924369747899159, 0.008149959250203748] data: [0.06482364156339371, 0.05629559150015858, 0.23743589743589744, 0.19453781512605042, 0.17440912795436023] science: [0.03813155386081983, 0.008087535680304472, 0.03743589743589744, 0.03235294117647059, 0.037489812550937245]  
 4. idf statistics: 0.22314355131420976 analytics: 0.0 data: 0.22314355131420976 science: 0.0  
 5. tf-idf statistics: [0.0038289646555345813, 0.0007077182090523621, 0.002860814760438587, 0.002343944866745901, 0.002242953381859756] analytics: [0.0, 0.0, 0.0, 0.0, 0.0] data: [0.014464977587575084, 0.012561998210679427, 0.052982289363322624, 0.04340985893213409, 0.03891827219335036] science: [0.0, 0.0, 0.0, 0.0, 0.0]

## Q3: Part 2 (10)

Repeat Part 1, but first remove the stop words given in the file "stop\_words.txt".

The output for Part 2 should have the same format as Part 1, and should be written to (1) this Notebook and (2) an output file named "Q3\_Part2.txt". Note: The length of document, in this case, will not include stop words. Similarly, the number of unique words in documents, and the calculation of tf, idf, tf-idf should be done after removing the stop words.

Example output for the file "urls.txt" (Note that this is just an example (NOT correct answer).)

1. Number of unique words in documents: [596, 600, 231]
2. Length of documents: [1020, 1079, 357]
3. tf statistics: [0.011764705882352941, 0.004633920296570899, 0.0] analytics: [0.004901960784313725, 0.005560704355885079, 0.0] data: [0.09411764705882353, 0.09731232622798888, 0.0] science: [0.06372549019607843, 0.020389249304911955, 0.04481792717086835]
4. idf statistics: 0.405465108108 analytics: 0.405465108108 data: 0.405465108108 science: 0.0
5. tf-idf statistics: [0.0047701777424489925, 0.0018788929940137366, 0.0] analytics: [0.0019875740593537469, 0.002254671592816484, 0.0] data: [0.03816142193959194, 0.039456752874288473, 0.0] science: [0.0, 0.0, 0.0]

The above values are for the first three webpage urls given in the file "urls.txt". The number of unique words in documents, average length of documents, tf and tf-idf values for the four words, must be in the order of the urls given in "urls.txt".

In [10]:

```

## YOUR CODE HERE
from bs4 import BeautifulSoup
import urllib
import re
import math

f = open('urls.txt', 'r')
url = f.readline()
word_list = []
unique_words = 0
tf_words=["statistics", "analytics", "data", "science"]
tf_freq=[0, 0, 0, 0]
idf_docfreq=[0, 0, 0, 0]
idf_flag=[0, 0, 0, 0]

unique_words_res = []
length_res = []
tf_res = [[0 for col in range(5)] for row in range(4)]
idf_res = []
cnt = 0

f2 = open('stop_words.txt', 'r')
stop_list = f2.readlines()
for i in range(0, len(stop_list), 1):
    stop_list[i] = stop_list[i][:len(stop_list[i])-1]
f2.close()

while url:
    r = urllib.request.urlopen(url).read()
    soup = BeautifulSoup(r)
    paragraphs = soup.find_all('p')
    unique_words = 0
    words = 0
    word_list = []
    for i in range(0, 4, 1):
        idf_flag[i] = 0
    for p in paragraphs:
        pText = p.get_text()
        pText = pText.lower()
        pattern = re.compile(r"WW")
        for match in re.finditer(pattern, pText):
            if match.group() != ' ':
                pText = pText.replace(match.group(), ' ')
        pText = pText.split(' ')
        for w in pText:
            if w not in stop_list:
                if w != '':
                    words+=1
                    if w not in word_list:
                        word_list.append(w)
                        unique_words+=1
                    if w in tf_words:
                        for i in range(0, 4, 1):
                            if tf_words[i] == w:
                                tf_freq[i] += 1
                                if idf_flag[i] == 0:
                                    idf_flag[i] = 1
                                idf_docfreq[i]+=1

```

```

unique_words_res.append(unique_words)
length_res.append(words)
for i in range(0, 4, 1):
    tf_res[i][cnt] = tf_freq[i]/words
    cnt+=1

url = f.readline()

for i in range(0, 4, 1):
    idf_res.append(math.log(5/idf_docfreq[i]))

f.close()
f = open('Q3_Part2.txt', 'w')
f.write("1. Number of unique words in documents: ")
f.write(str(unique_words_res)+'\n')
print("1. Number of unique words in documents:", unique_words_res)
f.write("2. Length of documents: ")
f.write(str(length_res)+'\n')
print("2. Length of documents:", length_res)
f.write("3. tf ")
print("3. tf ", end="")
for i in range(0, 4, 1):
    f.write(tf_words[i]+": ")
    f.write(str(tf_res[i])+' ')
    print(tf_words[i]+':', str(tf_res[i]), end=" ")

f.write("\n4. idf ")
print("\n4. idf ", end="")
for i in range(0, 4, 1):
    f.write(tf_words[i]+": ")
    f.write(str(idf_res[i])+' ')
    print(tf_words[i]+':', str(idf_res[i]), end=" ")

tf_idf_res = [[0 for col in range(5)] for row in range(4)]
for i in range(0, 4, 1):
    for j in range(0, 5, 1):
        tf_idf_res[i][j] = tf_res[i][j]*idf_res[i]
f.write("\n5. tf-idf ")
print("\n5. tf-idf", end=" ")
for i in range(0, 4, 1):
    f.write(tf_words[i]+": ")
    f.write(str(tf_idf_res[i])+' ')
    print(tf_words[i]+':', str(tf_idf_res[i]), end=" ")

f.close()

```

1. Number of unique words in documents: [389, 1794, 606, 825, 1017]  
2. Length of documents: [644, 3837, 1102, 1521, 2051]  
3. tf statistics: [0.027950310559006212, 0.005212405525149857, 0.02268602540834845, 0.01643655489809336, 0.01803998049731838] analytics: [0.0015527950310559005, 0.0049517852488923635, 0.022686025408348458, 0.017094017094017096, 0.01462701121404193] data: [0.10559006211180125, 0.09252019807140996, 0.42014519056261346, 0.304404996712689, 0.3130180399804973] science: [0.062111801242236024, 0.013291634089132134, 0.0662431941923775, 0.05062458908612755, 0.06728425158459288]  
4. idf statistics: 0.22314355131420976 analytics: 0.0 data: 0.22314355131420976 science: 0.0  
5. tf-idf statistics: [0.006236931558471701, 0.0011631146797717476, 0.005062240274823271, 0.0036677112313315214, 0.004025505313810708] analytics: [0.0, 0.0, 0.0, 0.0, 0.0] data: [0.02356174144311532, 0.02064528556594852, 0.09375268988972697, 0.06792601200425977, 0.06984795706666147] science: [0.0, 0.0, 0.0, 0.0, 0.0]

In [ ]: