CASE STUDY

Frequency Analysis, and Proof of Benford's Law for large text files.

Roynaaz Banu - 12191030227
Sunanda Tata - 12191030228
Bhargav Kantheti - 12191030229

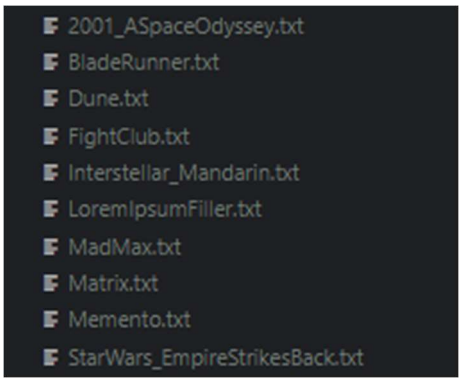
B2 – CSE
Gitam University

Project Repository Link:

https://github.com/bharxhav/WordFrequencyAnalysis

Part 0: Collecting Data

Splitting words, punctuation and whitespaces, can be easy, but to test the efficiency of the program, we need to run it with **huge** data sizes, and the best one is *movie scripts*, since they are thousands of lines.



Here is front page of script of the movie “The Matrix”.

```
                    THE MATRIX



                        by

              Larry and Andy Wachowski



                    June 3, 1997



FADE IN:

ON COMPUTER SCREEN

so close it has no boundaries.

A blinking cursor pulses in the electronic darkness like a
heart coursing with phosphorous light, burning beneath the
derma of black-neon glass.

A PHONE begins to RING, we hear it as though we were
making the call.  The cursor continues to throb,
relentlessly patient, until --

                    MAN (V.O.)
          Yeah?

Data now slashes across the screen, information flashing
faster than we can read:  "Call trans opt:  received
```

The code has been tested with more than one movie, and with languages such as English and Latin.

Here is Latin filler text called “lorem ipsum”.

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incidid

In nisl nisi scelerisque eu ultrices vitae auctor eu. Dolor sit amet consectetur adipi

Urna nunc id cursus metus. Id leo in vitae turpis massa. Blandit turpis cursus in hac

Egestas egestas fringilla phasellus faucibus scelerisque. Sit amet dictum sit amet jus

Laoreet suspendisse interdum consectetur libero id faucibus nisl tincidunt. Enim sit a

Nibh praesent tristique magna sit amet purus gravida quis. Commodo viverra maecenas ac

Eget mi proin sed libero enim sed faucibus turpis in. Sed odio morbi quis commodo odio

Nunc lobortis mattis aliquam faucibus purus in massa tempor. Amet consectetur adipisci

Cras sed felis eget velit aliquet sagittis. Et sollicitudin ac orci phasellus egestas

Elementum facilisis leo vel fringilla est ullamcorper eget nulla. Sit amet nulla facil

Mi eget mauris pharetra et. A erat nam at lectus urna duis convallis. Sit amet porttit

Accumsan lacus vel facilisis volutpat est velit egestas dui id. Diam sit amet nisl sus
```

Here is a part of script from movie “Blade Runner”.

```
                    Then, it dropped into darkness.

                    The great machine knew that this
                    tiny scout was reporting back to
                    its parent; but it was too simple,
                    too primative a device to detect
                    the forces that were gathering
                    round it now.

                    Then the pod came, carrying
                    life. The great machine searched
                    its memories.

                    The logic circuits made their
                    decision when the pod had fallen
                    beyond the last faint glow of the
                    reflected Saturnian light.

                    In a moment of time, too short to
                    be measured, space turned and
                    twisted upon itself.

12/9/65                                    d8
-----------------------------------------------------------------------
                  END OF SCREENPLAY
                    END OF FILE
```

Part 1: Creating Space for Results

To store the frequencies, and also keep them organizes, I created a batch file to automate the results for me. By writing onto the batch file, I can create folders by running it.

Inside the Batch File

results > FolderMaker.bat

```
...
cd H:\bhargav\Projects-DSP-Case_Study\WordFrequencyAnalysis\results

mkdir 2001_ASpaceOdyssey
mkdir BladeRunner
mkdir Dune
mkdir FightClub
mkdir LoremIpsumFiller
mkdir MadMax
mkdir Matrix
mkdir Memento
mkdir StarWars_EmpireStrikesBack
```

To do all this, I created a class called BatchMaker, which will

- Create a batch file.
- Run the batch file.

BatchMaker constructor

```
class BatchMaker:
    def __init__(self, files, directory="H:\\bhargav\\Projects-
DSP-Case_Study\\WordFrequencyAnalysis\\results"):
        self.files = files
        self.directory = directory
```

Batch File Writer

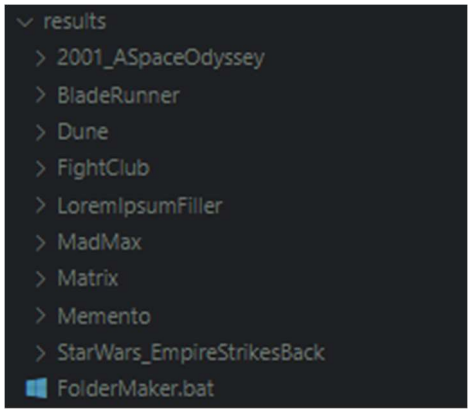
```
    def write_batch(self):
        adj = ["cd " + self.directory, ""]

        for f in self.files:
            adj.append('mkdir ' + f[:-4])

        fp = open('results/FolderMaker.bat', 'w')
        fp.write("\n".join(adj))
        fp.close()

        print("Written The Batch File")
```

Creating can be done by simply writing onto a file, whereas running it can be problematic, but with the **subprocesses** module, it can be done easily.

Batch File Runner

```
    def create_folders(self):
        subprocess.run([self.directory+"\\FolderMaker.bat"],
shell=True)

        print()
        print("Created Folders")
```

Result after Running the Batch



Part 2: Purifying the Data

Now, after creating a place for the results to lie, we need to separate the words from punctuation and whitespaces.

TextSplitter constructor

```
class TextSplitter:
    def __init__(self, filename):
        self.fn = filename

        with open("source/" + filename, 'r') as fp:
            self.contents = fp.read()

        self.name = filename[:-4]
        self.path = "results/" + self.name + "/"
```

This can be done using a for loop, and reading the files in source one by one.

I have created a class called TextSplitter which would do this for me, given that I give the file names.

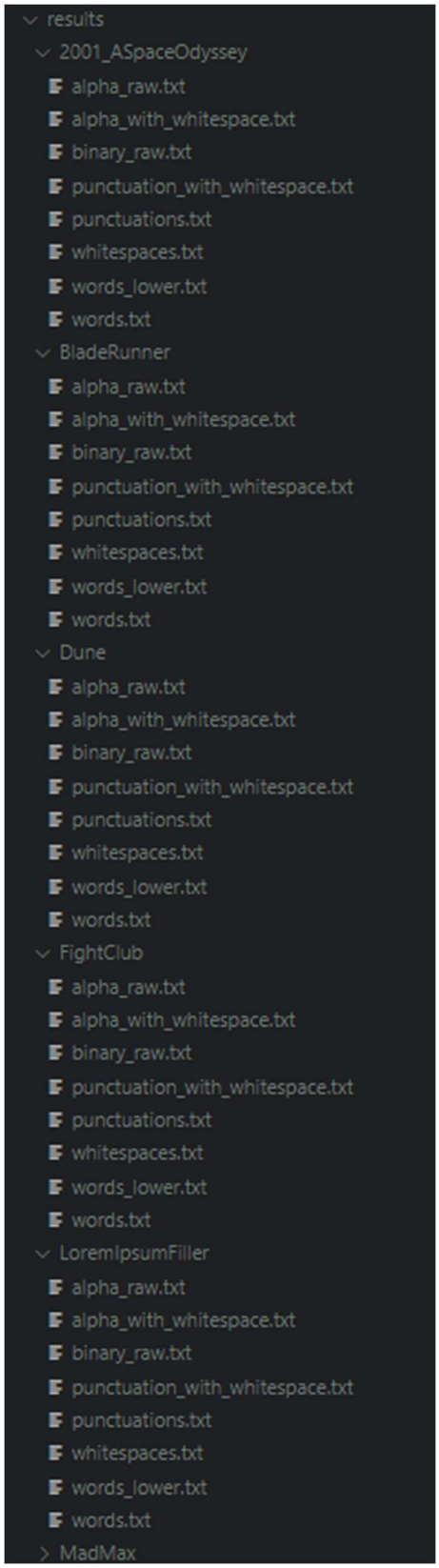
Word Partitioner

```python
def partition_words(self):
    words = []
    words_lowercase = []

    for word in self.contents.split():
        if word.isalpha():
            words.append(word.strip())
            words_lowercase.append(word.strip().lower())

    with open(self.path+'words.txt', 'w') as fp:
        fp.write("\n".join(words))
        print("Written words onto a file.")

    with open(self.path+'words_lower.txt', 'w') as fp:
        fp.write("\n".join(words_lowercase))
        print("Written lowercase words onto a file.")
```

Other Partition maker function

```python
def partition(self):
    alpha_raw = ""
    alpha_with_whitespace = ""
    binary_raw = ""
    punctuations = ""
    punctuation_with_whitespace = ""
    whitespaces = ""

    for c in self.contents:
        if c in whitespace:
            whitespaces += c
            punctuation_with_whitespace += c
            alpha_with_whitespace += c

        if c in punctuation:
            punctuations += c
            punctuation_with_whitespace += c

        if c.isalpha():
            alpha_raw += c
            alpha_with_whitespace += c
            binary_raw += str(ord(c)) + " "

    with open(self.path+'alpha_raw.txt', 'w') as fp:
        print('Written raw alphabets onto a file.')
        fp.write(alpha_raw)

    with open(self.path+'alpha_with_whitespace.txt', 'w') as fp:
        print('Written alphabets with whitespaces onto a file.')
        fp.write(alpha_with_whitespace)

    with open(self.path+'binary_raw.txt', 'w') as fp:
        print('Written binary raw onto a file.')
        fp.write(binary_raw)

    with open(self.path+'punctuations.txt', 'w') as fp:
        print('Written raw punctuations onto a file.')
        fp.write(punctuations)

    with open(self.path+'punctuation_with_whitespace.txt', 'w') as
fp:
        print('Written punctuations with whitespaces onto a file.')
        fp.write(punctuation_with_whitespace)

    with open(self.path+'whitespaces.txt', 'w') as fp:
        print('Written raw whitespaces onto a file.')
        fp.write(whitespaces)
```

Result


results
2001_ASpaceOdyssey
alpha_raw.txt
alpha_with_whitespace.txt
binary_raw.txt
punctuation_with_whitespace.txt
punctuations.txt
whitespaces.txt
words_lower.txt
words.txt
BladeRunner
alpha_raw.txt
alpha_with_whitespace.txt
binary_raw.txt
punctuation_with_whitespace.txt
punctuations.txt
whitespaces.txt
words_lower.txt
words.txt
Dune
alpha_raw.txt
alpha_with_whitespace.txt
binary_raw.txt
punctuation_with_whitespace.txt
punctuations.txt
whitespaces.txt
words_lower.txt
words.txt
FightClub
alpha_raw.txt
alpha_with_whitespace.txt
binary_raw.txt
punctuation_with_whitespace.txt
punctuations.txt
whitespaces.txt
words_lower.txt
words.txt
LoremIpsumFiller
alpha_raw.txt
alpha_with_whitespace.txt
binary_raw.txt
punctuation_with_whitespace.txt
punctuations.txt
whitespaces.txt
words_lower.txt
words.txt
MadMax

Alpha Raw

```
ASPACEODYSSEYScreenplaybyStanleyKubrickandArthurCClarkHawkFilmsLtdcoMGMStudios
dlastednowfortenmillionyearsandwouldnotendforanothermillionThereignoftheterrib
rvivalhadreachedanewclimaxofferocityandthevictorwasnotyetinsightInthisdryandba
ERThemanapesofthefieldhadnoneoftheseattributesandtheywereonthelongpatheticroad
byasluggishbrownstreamThetribehadalwaysbeenhungryandnowitwasstarvingAsthefirst
nowtheOldOnewashisfatherforsucharelationshipwasbeyondhisunderstandingbutashest
theroutofthecaveandleaveshimforthehyenasAmonghiskindMoonwatcherisalmostagiantH
isquitemanlikeandhisheadisalreadynearermanthanapeTheforeheadislowandtherearegr
uponthehostileworldthereisalreadyaACONTINUEDsomethinginhisgazebeyondthegraspof
otfulfillitselfforanothertwomillionyearsaAEXTTHESTREAMTHEOTHERSAsthedawnskybri
hersideeverydaythatdidnotmakeitanylessannoyingTherareeighteenofthemanditisimp
oangrilydanceandshriekontheirsideofthestreamandhisownpeoplereplyInkindTheconfr
hemuddywaterHonorhasbeensatisfiedeachgrouphasstakeditsclaimtoitsownterritoryaA
pangsofhungerwhileallaroundthemcompetingwiththemforthesamrfodderisapotentialso
vannaandthroughthebrushisnotonlybeyondtheirreachtheideaofeatingitisbeyondtheir
beslowlywandersacrossthebareflatcountrysideforagingforrootsandoccasionalberrie
oundSuddenlyMoonwatcherbecomesawareofalionstalkingthemaboutyardsawayDefenceles
DSHONEYIthadnotbeenagooddaythoughasMoonwatcherhadnorealremembranceofthepasthec
eeandsoenjoysthefinestdelicacyhispeoplecouldeverknowOfcoursehealsocollectsagoo
sstillhungryheisnotactuallyweakwithhungerThatwasthemostthatanyhominidcouldhope
```

Alphabets with Whitespaces

```
You, 2 days ago | 1 author (You)
      You, 2 days ago • Add outputs

                    A SPACE ODYSSEY

                       Screenplay

                          by

              Stanley Kubrick and Arthur C Clark

                       Hawk Films Ltd
                       co MGM Studios
                       Boreham Wood
                       Herts


TITLE                      PART I
                       AFRICA
                        YEARS AGO

A
VIEWS OF AFRICAN DRYLANDS  DROUGHT

The remorseless drought had lasted now for ten million years
and would not end for another million The reign of the ter
rible lizards had long since passed but here on the continent
which would one day be known as Africa the battle for survival
had reached a new climax of ferocity and the victor was not
yet in sight In this dry and barren land only the small or
the swift or the fierce could flourish or even hope to exist
```

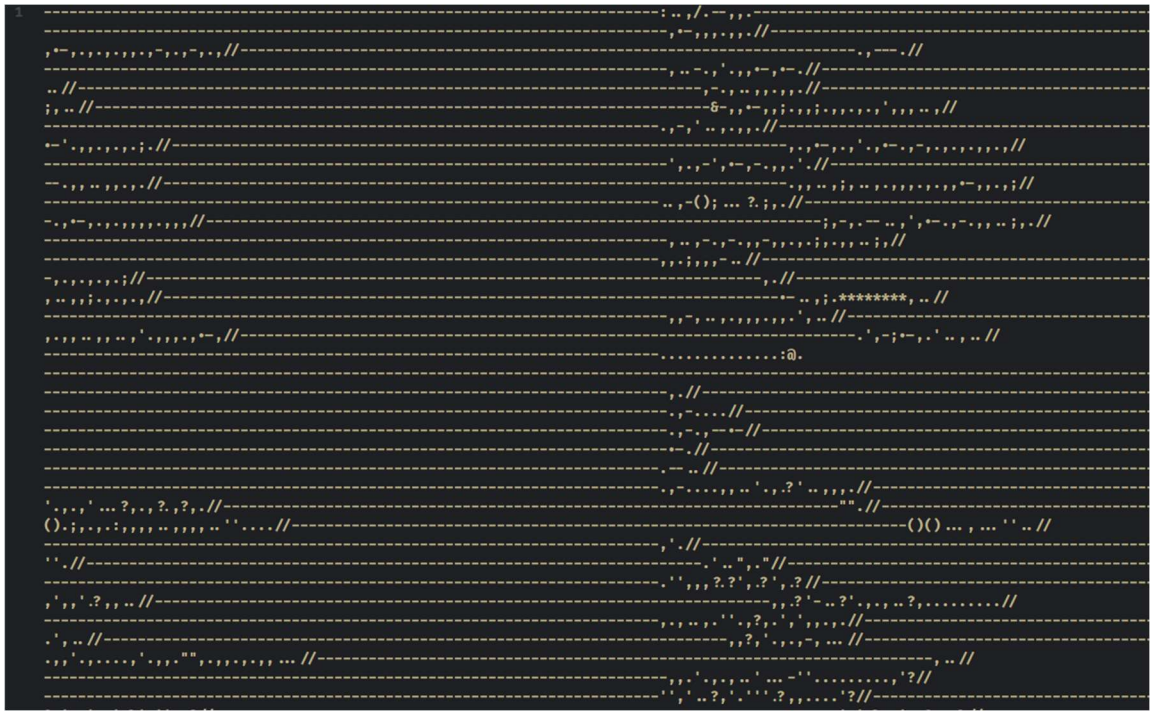
Binary Raw

```
65 83 80 65 67 69 79 68 89 83 83 69 89 83 99 114 101 101 110 112 108 97 121 98 121
67 108 97 114 107 72 97 119 107 70 105 108 109 115 76 116 100 99 111 77 71 77 83 11
84 76 69 80 65 82 84 73 65 70 82 73 67 65 89 69 65 82 83 65 71 79 65 86 73 69 87 83
101 109 111 114 115 101 108 101 115 115 100 114 111 117 103 104 116 104 97 100 108
101 97 114 115 97 110 100 119 111 117 108 100 110 111 116 101 110 100 102 111 114 9
111 102 116 104 101 116 101 114 114 105 98 108 101 108 105 122 97 114 100 115 104 9
101 111 110 116 104 101 99 111 110 116 105 110 101 110 116 119 104 105 99 104 119 1
99 97 116 104 101 98 97 116 116 108 101 102 111 114 115 117 114 118 105 118 97 108
101 114 111 99 105 116 121 97 110 100 116 104 101 118 105 99 116 111 114 119 97 115
97 110 100 98 97 114 114 101 110 108 97 110 100 111 110 108 121 116 104 101 115 109
99 101 99 111 117 108 100 102 108 111 117 114 105 115 104 111 114 101 118 101 110 1
79 78 87 65 84 67 72 69 82 84 104 101 109 97 110 97 112 101 115 111 102 116 104 101
114 105 98 117 116 101 115 97 110 100 116 104 101 121 119 101 114 101 111 110 116 1
105 97 108 101 120 116 105 110 99 116 105 111 110 65 98 111 117 116 116 119 101 110
111 102 99 97 118 101 115 111 118 101 114 108 111 111 107 105 110 103 97 115 109 97
98 121 97 115 108 117 103 103 105 115 104 98 114 111 119 110 115 116 114 101 97 109
110 103 114 121 97 110 100 110 111 119 105 116 119 97 115 115 116 97 114 118 105 11
119 110 99 114 101 101 112 115 105 110 116 111 116 104 101 99 97 118 101 77 111 111
105 115 102 97 116 104 101 114 104 97 115 100 105 101 100 100 117 114 105 110 103 1
101 79 108 100 79 110 101 119 97 115 104 105 115 102 97 116 104 101 114 102 111 114
111 110 100 104 105 115 117 110 100 101 114 115 116 97 110 100 105 110 103 98 117 1
116 116 104 101 101 109 97 99 105 97 116 101 100 98 111 100 121 104 101 102 101 101
105 110 116 111 115 97 100 110 101 115 115 84 104 101 110 104 101 99 97 114 114 105
101 99 97 118 101 97 110 100 108 101 97 118 101 115 104 105 109 102 111 114 116 104
110 119 97 116 99 104 101 114 105 115 97 108 109 111 115 116 97 103 105 97 110 116
```

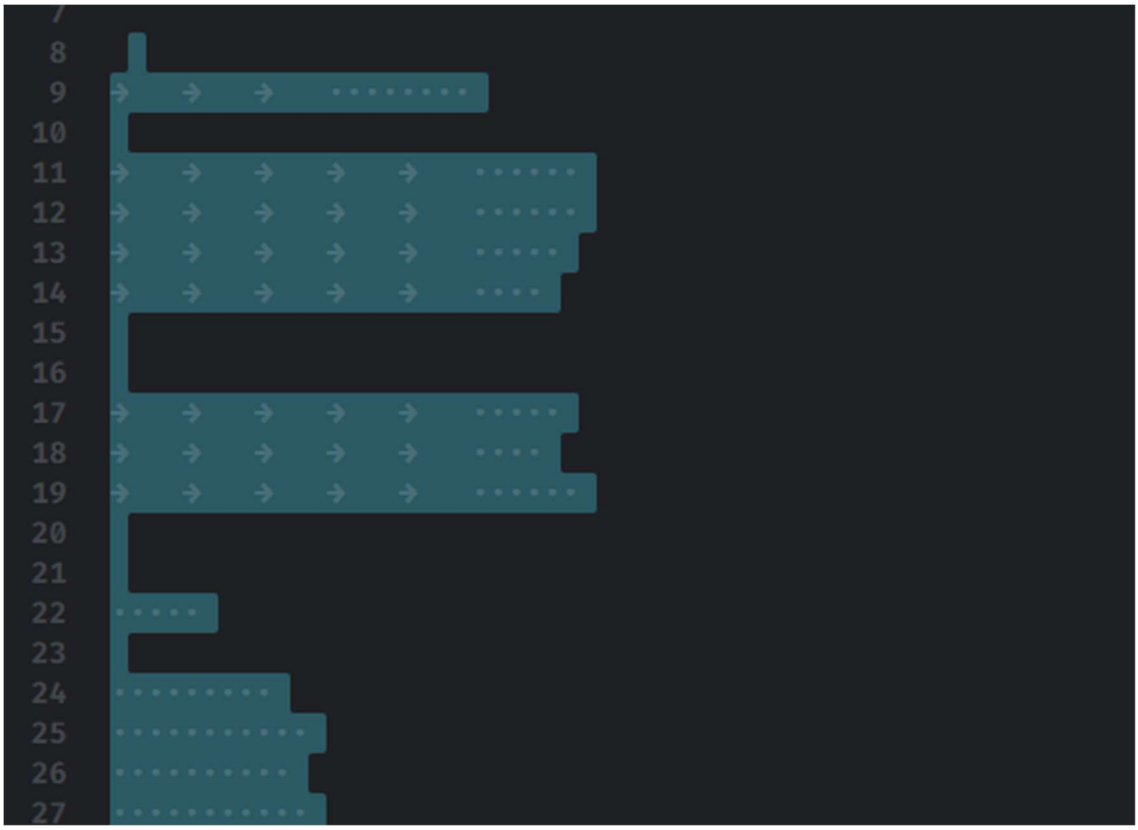
Punctuation with Whitespaces

```
                    ''
-----------------------------------------------------------------------

   -

        ,
     .     -
    ,
      ,
     ,
  .     ,
     ,     .

//
-----------------------------------------------------------------------

 &   -

 -        ,
    ,     .

 , ,    ,  .

    ,     .
        ,
      .
```

Punctuation



Whitespaces



Words

```
ODYSSEY
Screenplay
by
Stanley
Kubrick
and
Arthur
Clark
Hawk
Films
Boreham
TITLE
PART
I
AFRICA
YEARS
```

Lowercase Words

```
space
odyssey
screenplay
by
stanley
kubrick
and
arthur
clark
hawk
films
boreham
title
part
i
africa
years
ago
views
of
african
drylands
drought
the
```

Part 3: Finding Frequencies

After having the required data, I found the frequencies using the Counter class, which will find frequencies of each word and character.

I did all this in a class called FrequencyAnalyser which does this and one more other task which I will discuss in the next part.

FrequencyAnalyser constructor

```
class FrequencyAnalyser:
    def __init__(self, filename):
        self.fn = filename
        self.name = filename[:-4]

        self.path = "results/" + self.name + "/"
```

Analyse Function

```
    def analyse(self):
        # Reading Words and Characters and Marking Frequencies
        with open(self.path+'words_lower.txt', 'r') as wl,
open(self.path+'alpha_raw.txt', 'r') as cl:
            self.words = dict(Counter(wl.read().split()))
            self.chars = dict(Counter(cl.read().lower()))

        print("Analysed the word frequencies.")
        print("Analysed the character frequencies.")
```

Character Frequencies

```
{'t': 10583, 'h': 6882, 'e': 14843, 'm': 2660, 'p': 2540, 'i': 8344, 'r': 7470, 's': 7
, 'f': 2311, 'o': 9621, 'u': 3547, 'x': 306, 'v': 1081, 'z': 143, 'j': 137, 'q': 97}
```

Word Frequencies

```
Script :: StarWars_EmpireStrikesBack.txt
{'the': 1782, 'empire': 8, 'strikes': 2, 'back': 53, 'written': 1, 'by': 87, 'lawrence':
: 1, 'galaxy': 5, 'planet': 7, 'hoth': 69, 'star': 69, 'destroyer': 39, 'moves': 60, 'thr
 5, 'probes': 1, 'zooms': 4, 'toward': 74, 'lands': 3, 'on': 210, 'covered': 4, 'an': 77,
rd': 1, 'mechanical': 5, 'sound': 11, 'rises': 10, 'above': 13, 'whining': 1, 'strange':
rs': 7, 'into': 124, 'small': 27, 'figure': 5, 'gallops': 2, 'windswept': 2, 'ice': 48,
, 'beneath': 4, 'speeding': 1, 'paws': 1, 'up': 83, 'slope': 3, 'reins': 2, 'his': 315,
something': 18, 'in': 276, 'he': 217, 'takes': 19, 'pair': 3, 'electrobinoculars': 2, 'ut
ind': 11, 'whips': 1, 'at': 144, 'cap': 1, 'activates': 3, 'comlink': 3, 'tauntaun': 8,
```

Part 4: The Plot

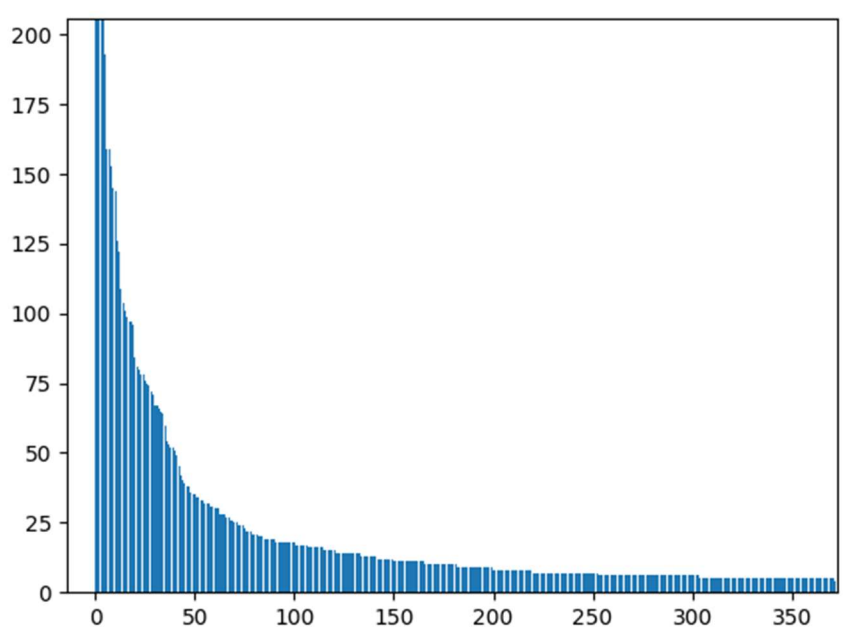
Here, in the same class, FrequencyAnalyser I added another subroutine, which will plot all this using pyplot. Here we can find a trend / pattern, which I will explain later.

Plot Subroutine

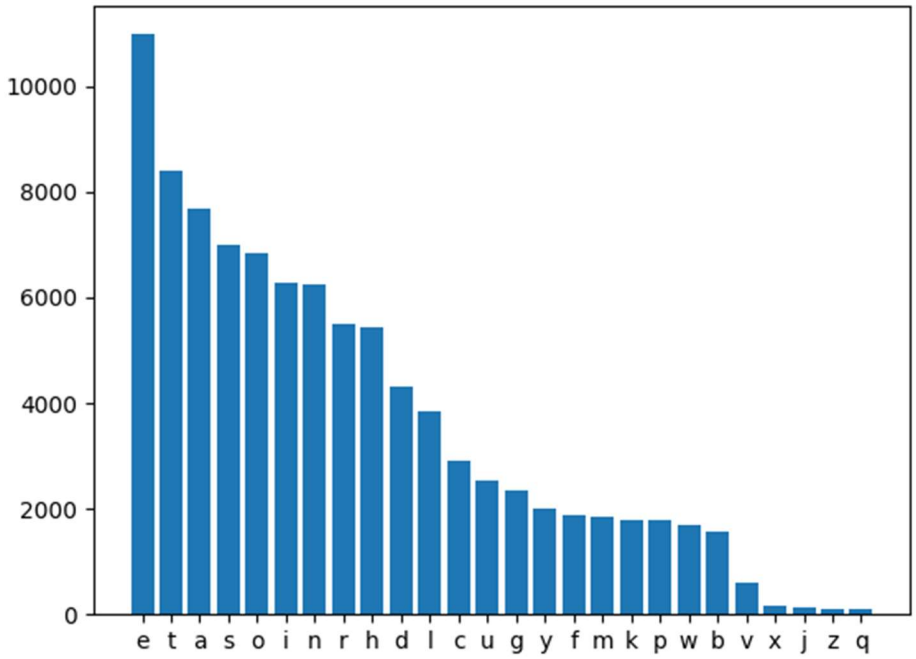
```
def plot(self):
    def second(x): return x[1]

    self.words = list(self.words.items())
    self.chars = list(self.chars.items())

    self.words.sort(key=second, reverse=True)
    self.chars.sort(key=second, reverse=True)

    print("Plotting bar graph for word frequencies.")
    plt.bar(range(len(self.words)), list(
        map(second, self.words)), align='center')
    # plt.xticks(range(len(self.words)), list(
    #     map(lambda x: x[0], self.words)))
    plt.show()

    print("Plotting bar graph for character frequencies.")
    plt.bar(range(len(self.chars)), list(
        map(second, self.chars)), align='center')
    plt.xticks(range(len(self.chars)), list(
        map(lambda x: x[0], self.chars)))
    plt.show()
```

Result

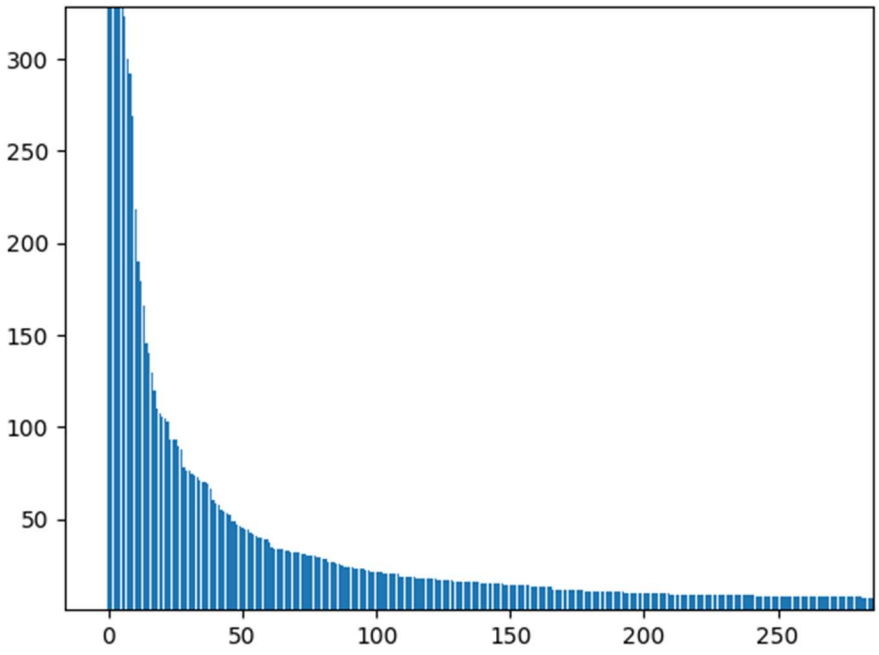
Word Frequencies of Space Odyssey



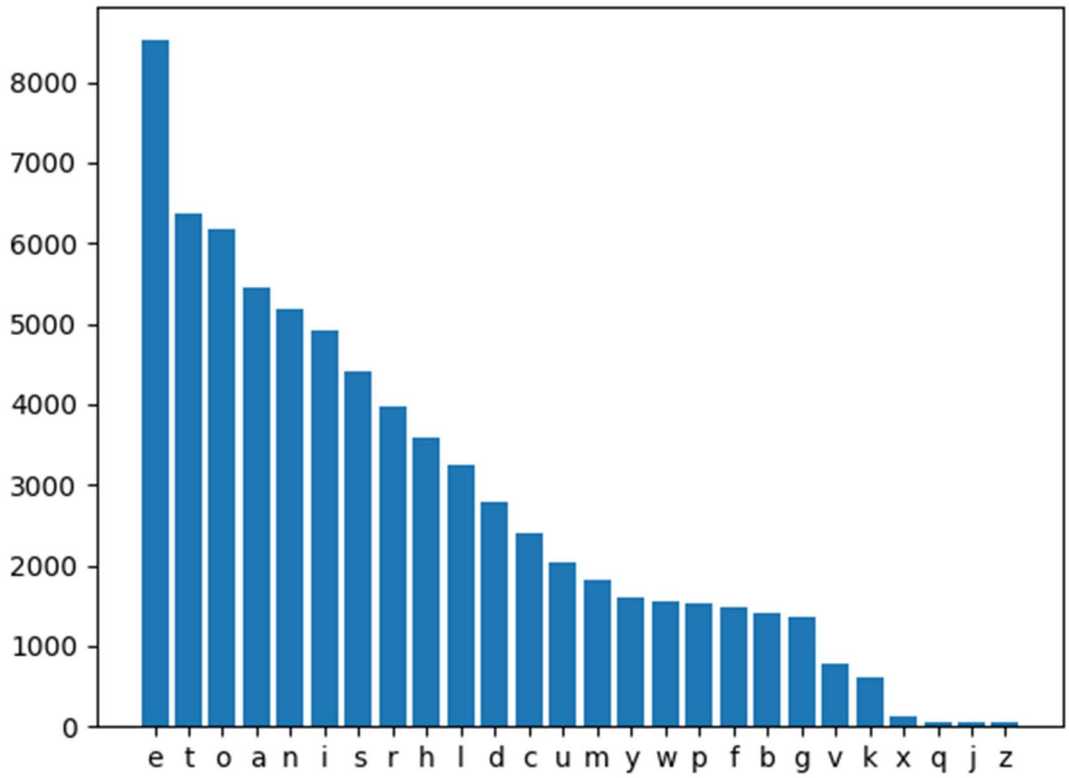
Character Frequencies of Space Odyssey



Word Frequencies of Blade Runner



Character Frequencies of Blade Runner



Part 5: The Proof

In the previous part, the curve that was common in all movies, is called a **Benford's Curve** which is a very common trend observed in every trend aspect.

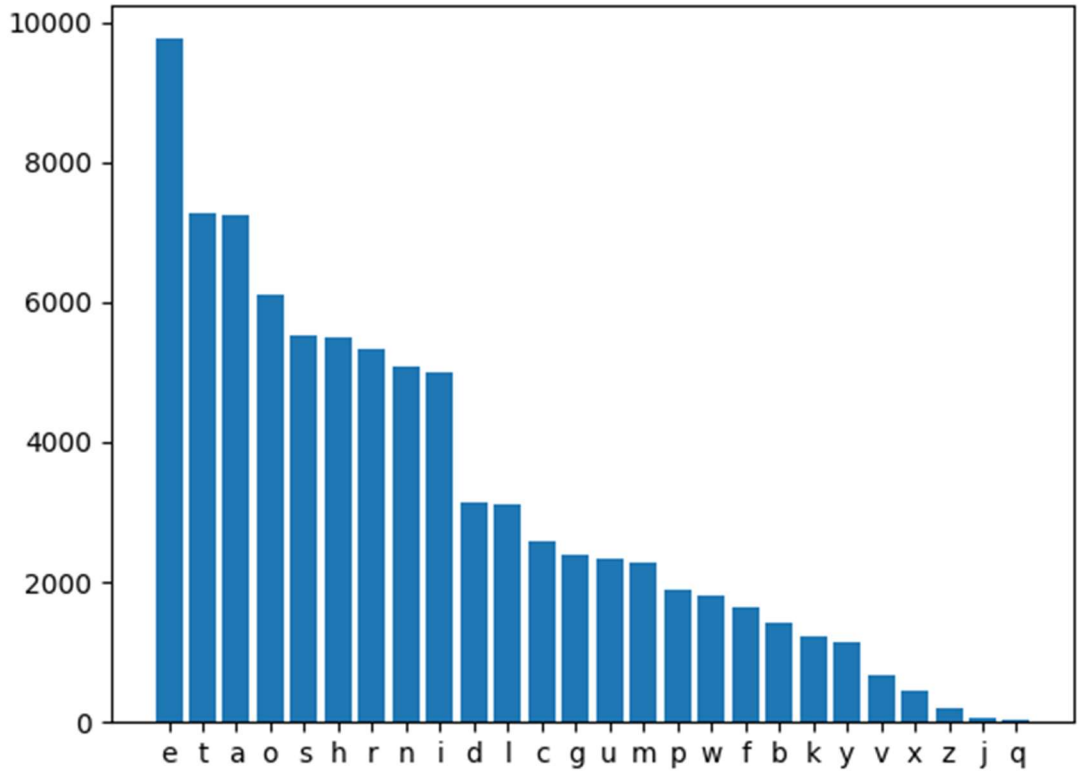
The Benford's law can be used to determine **voting fraud, flood detections, etc.**

I have proved Benford's law by using these movies, now with this proof, I will be showing how to detect languages of origin.

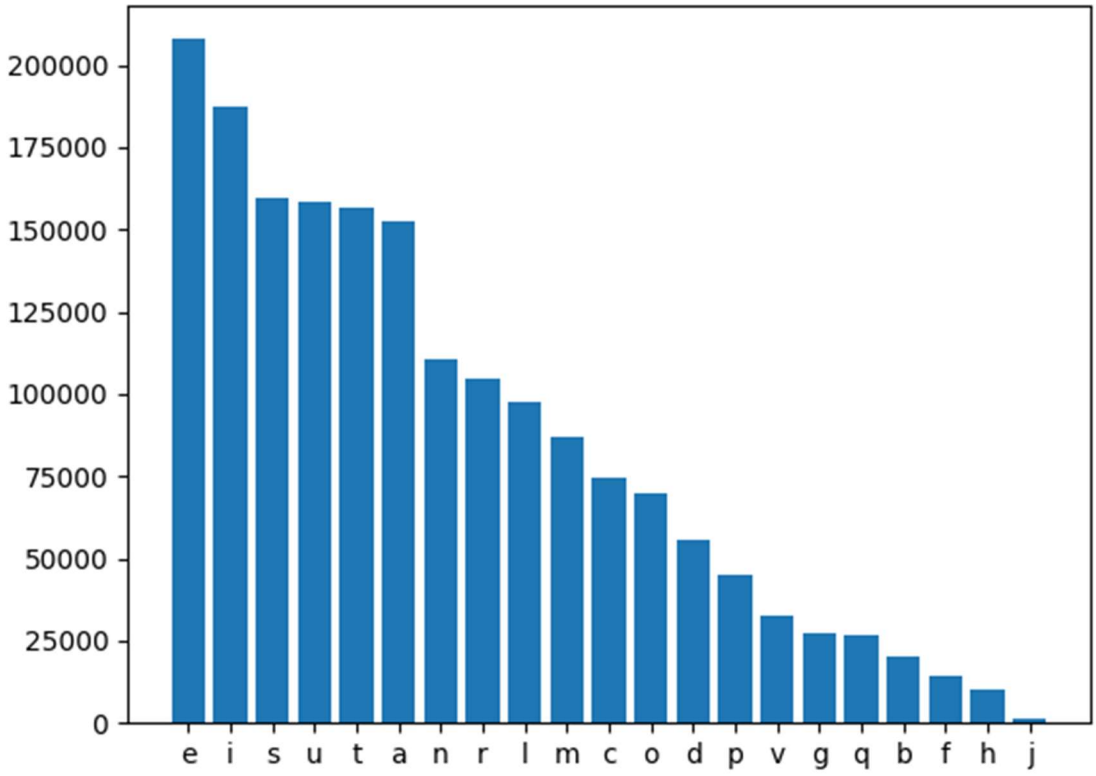
Previously, I have taken the *Lorem Ipsum* file, containing 2MB of filler information. This has a **Latin** origin, and English has **Greek and Latin** origin. Showing us some similarities, as well as **distinct** features that separate them.

Observe the frequency of **q**, **h** and **z** on these.

Here is an English Movie, MadMax.



Here is the Latin, Lorem Ipsum filler text.



We can see very distinct differences in the usage of syllables. The letter **h** has rarely been used, where as in English, it is part of most common word, which is “The”. Thus, we can find differences like this.

Here is the main, which has been running all these parts

```python
# Word Frequency Analysis
from TextSplitter import TextSplitter
from BatchMaker import BatchMaker
from FrequencyAnalyser import FrequencyAnalyser

if __name__ == '__main__':
    files = [
        "2001_ASpaceOdyssey.txt",
        "BladeRunner.txt",
        "Dune.txt",
        "FightClub.txt",
        "LoremIpsumFiller.txt",
        "MadMax.txt",
        "Matrix.txt",
        "Memento.txt",
        "StarWars_EmpireStrikesBack.txt"
    ]

    bm = BatchMaker(files)
    bm.write_batch()
    bm.create_folders()

    for f in files:
        print()
        print(f"Script :: {f}")

        movie = TextSplitter(f)
        movie.partition_words()
        movie.partition()

    for f in files:
        print()
        print(f"Script :: {f}")
        fs = FrequencyAnalyser(f)
        fs.analyse()
        fs.plot()
```

Thank You for reading patiently.

Bhargav Kantheti