COMP 7402
ASSIGNMENT 1

Rina Hong
A00964022
01/24/2019

Contents

User Guide

- To run program, please install python.

<u>In terminal:</u>

1) Type “Python”
2) Type “from assign1 import *”
3) Type “counter = letterCounter() ”
4) Type “counter.run()”
5) If prompt for input, enter correspondent value
6) To exit the program, ctrl + d

Design

class letterCounter:

- This class is for calculating frequencies and probabilities of each letter in given text file.

run(self):

- This is a runner function that runs all other functions

ask_user_input():

- ask_user_input() prompts user to input two file names to read text file and to write result.
- User is prompted to enter 1,2 or 3.
    1) Alice in Wonderland
    2) MobyDick
    3) Other file
- If 3 is entered, then user will be prompted again to enter filename. File name should include the file extention. If file name doesn’t exist in the directory, then user will be prompted again to enter right value.

read_file():

- read_file() reads the user specified text file and store each letter and their frequencies in object (a.k.a dictonary – Python terminology).
- At the end, it will print out the result to console.

output_result_to_csv():

- output_result_to_csv() writes the result of read_file() to the user specified csv file.

is_sum_of_probabilities_one():

- is_sum_of_probabilities_one() calculates probability of each letter and sums up all of the probabilities.
- Print out the sum of the probabilities to console.

calculate_conditional_probability():

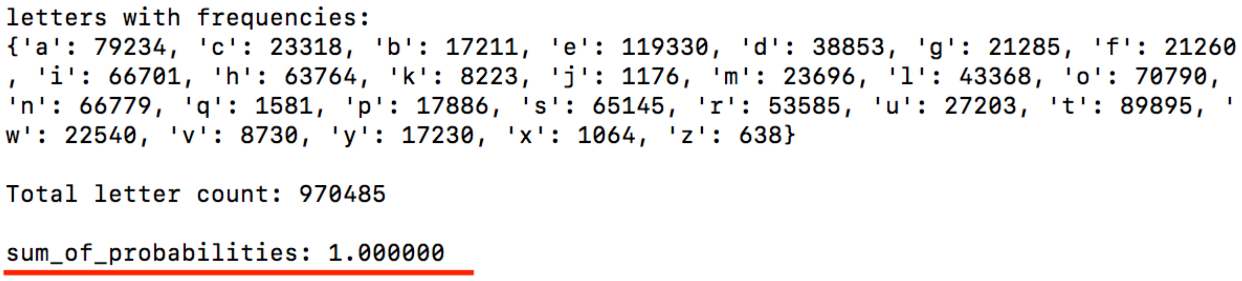
- calculate_conditional_probability() calculates conditional probabilities of most frequent letters. ['e', 't', 'a', 'i', 'o', 'n']
- Print out the result to console.

Report

Task 1:

After running program, calculate_conditional_probability() calculates the sum of the probabilities of each distribution, and it confirms that below equation is correct!

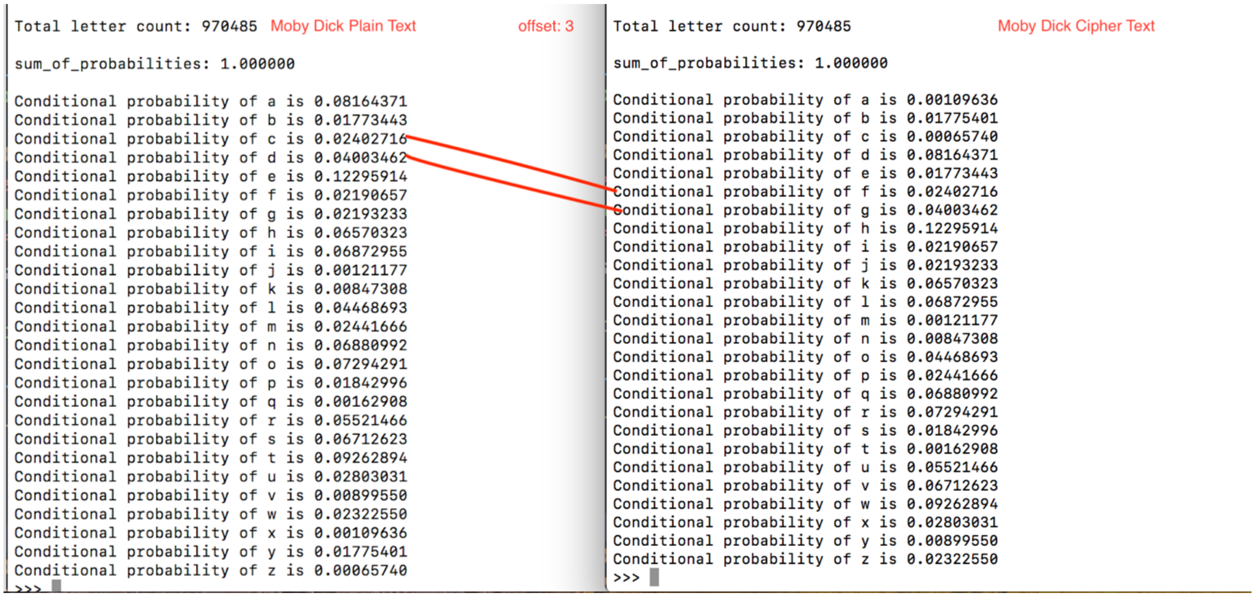
$$P(M) = \sum_{i=a}^{z} P(m_i) = 1$$



```
letters with frequencies:
{'a': 79234, 'c': 23318, 'b': 17211, 'e': 119330, 'd': 38853, 'g': 21285, 'f': 21260
, 'i': 66701, 'h': 63764, 'k': 8223, 'j': 1176, 'm': 23696, 'l': 43368, 'o': 70790,
'n': 66779, 'q': 1581, 'p': 17886, 's': 65145, 'r': 53585, 'u': 27203, 't': 89895, '
w': 22540, 'v': 8730, 'y': 17230, 'x': 1064, 'z': 638}

Total letter count: 970485

sum_of_probabilities: 1.000000
```

Task 2:

After examine the conditional probabilities, I can say results are helpful to determine the offset of Caesar ciphers. For example, if 3 is offset, then conditional probabilities are also shifted by 3 offset.



Moby Dick Plain Text

offset: 3

```
Total letter count: 970485

sum_of_probabilities: 1.000000

Conditional probability of a is 0.08164371
Conditional probability of b is 0.01773443
Conditional probability of c is 0.02402716
Conditional probability of d is 0.04003462
Conditional probability of e is 0.12295914
Conditional probability of f is 0.02190657
Conditional probability of g is 0.02193233
Conditional probability of h is 0.06570323
Conditional probability of i is 0.06872955
Conditional probability of j is 0.00121177
Conditional probability of k is 0.00847308
Conditional probability of l is 0.04468693
Conditional probability of m is 0.02441666
Conditional probability of n is 0.06880992
Conditional probability of o is 0.07294291
Conditional probability of p is 0.01842996
Conditional probability of q is 0.00162908
Conditional probability of r is 0.05521466
Conditional probability of s is 0.06712623
Conditional probability of t is 0.09262894
Conditional probability of u is 0.02803031
Conditional probability of v is 0.00899550
Conditional probability of w is 0.02322550
Conditional probability of x is 0.00109636
Conditional probability of y is 0.01775401
Conditional probability of z is 0.00065740
>>>
```

Moby Dick Cipher Text

```
Total letter count: 970485

sum_of_probabilities: 1.000000

Conditional probability of a is 0.00109636
Conditional probability of b is 0.01775401
Conditional probability of c is 0.00065740
Conditional probability of d is 0.08164371
Conditional probability of e is 0.01773443
Conditional probability of f is 0.02402716
Conditional probability of g is 0.04003462
Conditional probability of h is 0.12295914
Conditional probability of i is 0.02190657
Conditional probability of j is 0.02193233
Conditional probability of k is 0.06570323
Conditional probability of l is 0.06872955
Conditional probability of m is 0.00121177
Conditional probability of n is 0.00847308
Conditional probability of o is 0.04468693
Conditional probability of p is 0.02441666
Conditional probability of q is 0.06880992
Conditional probability of r is 0.07294291
Conditional probability of s is 0.01842996
Conditional probability of t is 0.00162908
Conditional probability of u is 0.05521466
Conditional probability of v is 0.06712623
Conditional probability of w is 0.09262894
Conditional probability of x is 0.02803031
Conditional probability of y is 0.00899550
Conditional probability of z is 0.02322550
>>>
```

How I calculate conditional probabilities:

P(k) = 1/26 , P(c_i) = 1/26

c_i is an element in the set of C. C is all possible English alphabets. Therefore, probability of c_i is 1/all-possible-English-alphabets = 1/26

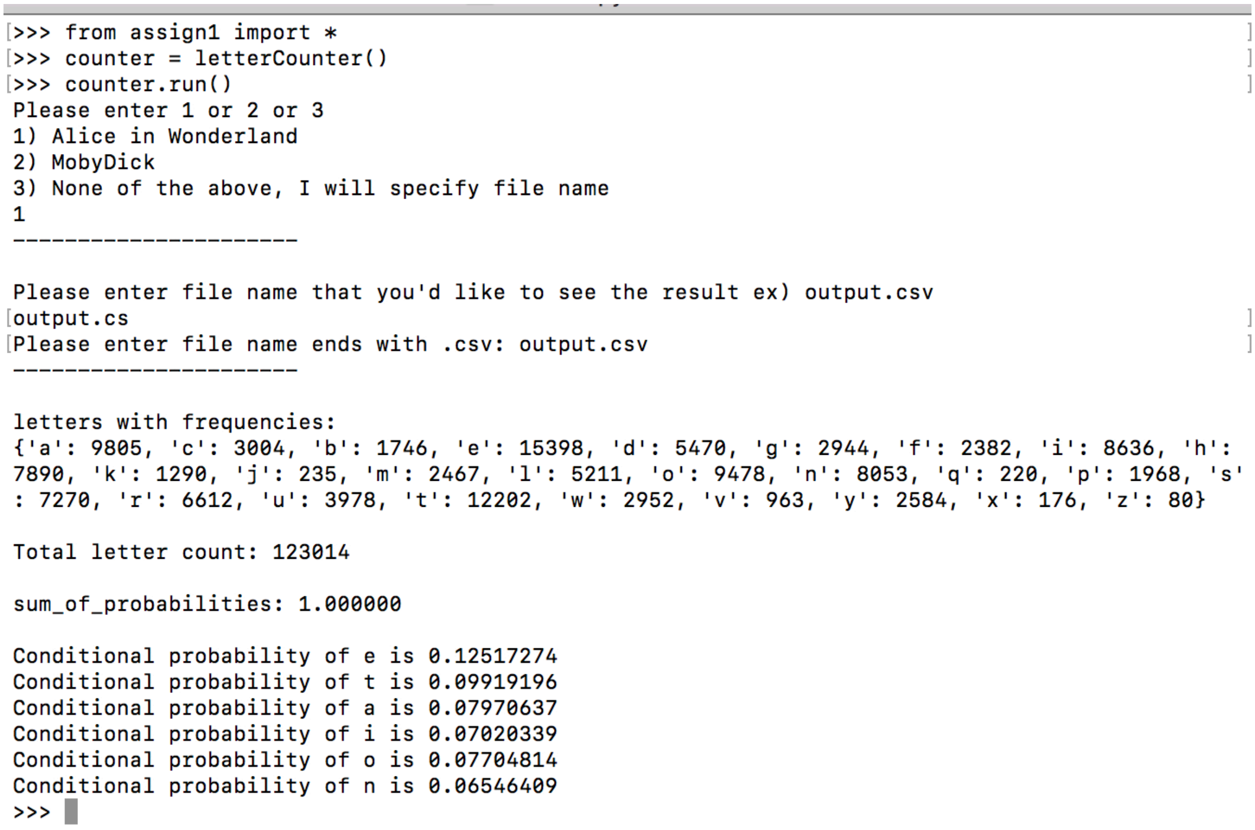
This can be proved by

$$P(C = c) = \sum_{\{k:c \in C(k)\}} P(K = k) \cdot P(M = d_k(c))$$

Testing and Supporting Data

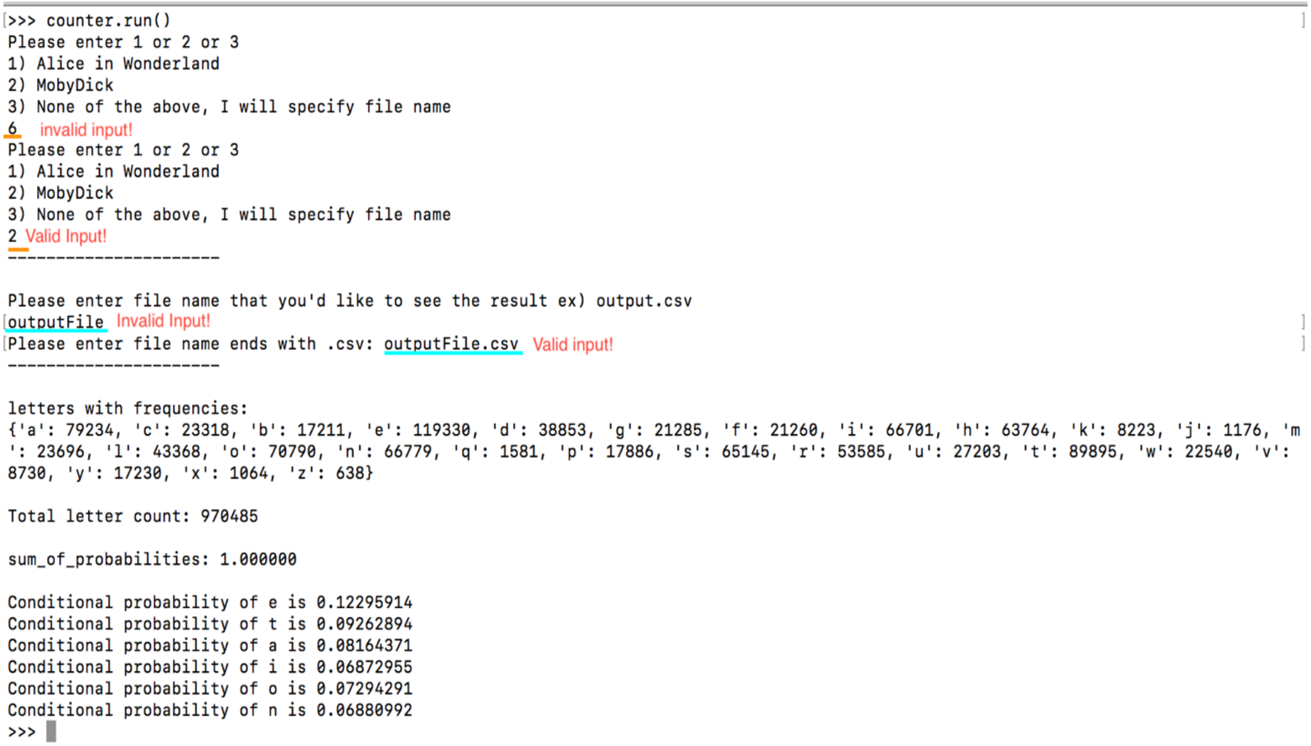
Test Cases:

1) When all the input entered correctly
   Expected Result == Actual Result ➔ CSV is generated and print the result to terminal



```
[>>> from assign1 import *
[>>> counter = letterCounter()
[>>> counter.run()
 Please enter 1 or 2 or 3
 1) Alice in Wonderland
 2) MobyDick
 3) None of the above, I will specify file name
 1
 ----------------------

 Please enter file name that you'd like to see the result ex) output.csv
[output.cs
[Please enter file name ends with .csv: output.csv
 ----------------------

 letters with frequencies:
 {'a': 9805, 'c': 3004, 'b': 1746, 'e': 15398, 'd': 5470, 'g': 2944, 'f': 2382, 'i': 8636, 'h':
 7890, 'k': 1290, 'j': 235, 'm': 2467, 'l': 5211, 'o': 9478, 'n': 8053, 'q': 220, 'p': 1968, 's'
 : 7270, 'r': 6612, 'u': 3978, 't': 12202, 'w': 2952, 'v': 963, 'y': 2584, 'x': 176, 'z': 80}

 Total letter count: 123014

 sum_of_probabilities: 1.000000

 Conditional probability of e is 0.12517274
 Conditional probability of t is 0.09919196
 Conditional probability of a is 0.07970637
 Conditional probability of i is 0.07020339
 Conditional probability of o is 0.07704814
 Conditional probability of n is 0.06546409
 >>>
```

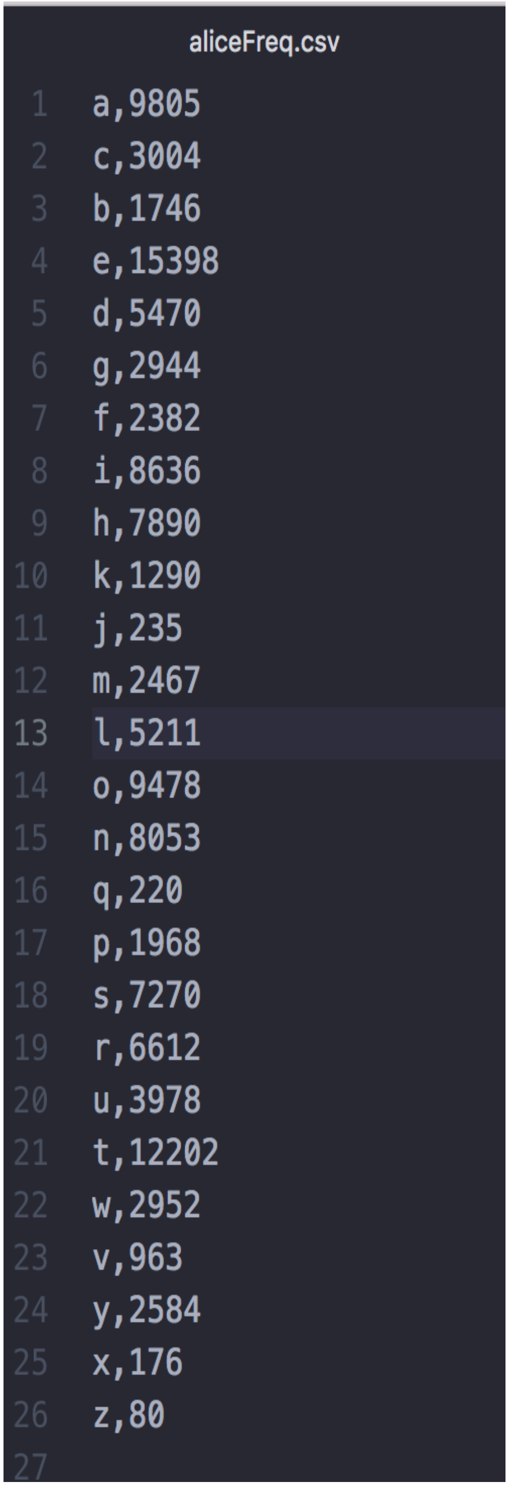
2) When input is not valid
   Expetected Result == Actual Result
   - ➔ Prompt user again for valid input.
   - ➔ Once valid input entered, then csv generated and print the result to the terminal.



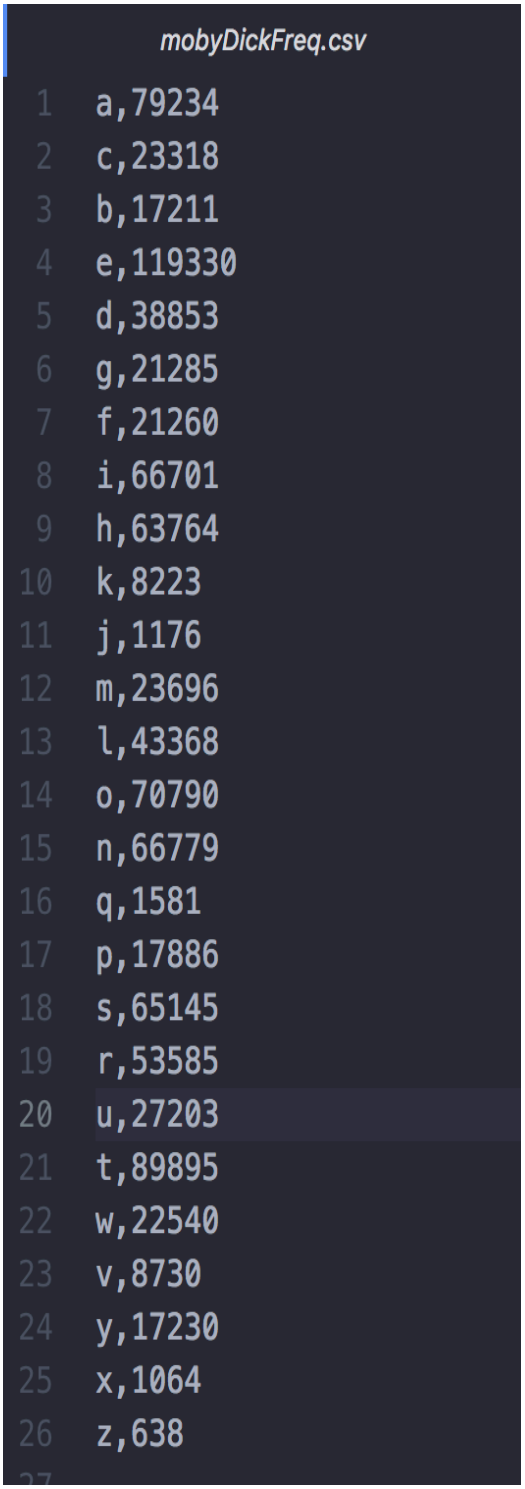
```
>>> counter.run()
Please enter 1 or 2 or 3
1) Alice in Wonderland
2) MobyDick
3) None of the above, I will specify file name
6   invalid input!
Please enter 1 or 2 or 3
1) Alice in Wonderland
2) MobyDick
3) None of the above, I will specify file name
2 Valid Input!
----------------------

Please enter file name that you'd like to see the result ex) output.csv
outputFile  Invalid Input!
Please enter file name ends with .csv: outputFile.csv  Valid input!
----------------------

letters with frequencies:
{'a': 79234, 'c': 23318, 'b': 17211, 'e': 119330, 'd': 38853, 'g': 21285, 'f': 21260, 'i': 66701, 'h': 63764, 'k': 8223, 'j': 1176, 'm
': 23696, 'l': 43368, 'o': 70790, 'n': 66779, 'q': 1581, 'p': 17886, 's': 65145, 'r': 53585, 'u': 27203, 't': 89895, 'w': 22540, 'v':
8730, 'y': 17230, 'x': 1064, 'z': 638}

Total letter count: 970485

sum_of_probabilities: 1.000000

Conditional probability of e is 0.12295914
Conditional probability of t is 0.09262894
Conditional probability of a is 0.08164371
Conditional probability of i is 0.06872955
Conditional probability of o is 0.07294291
Conditional probability of n is 0.06880992
>>> 
```

Screenshot

- Generated CSV for Alice in Wonderland and Moby dick

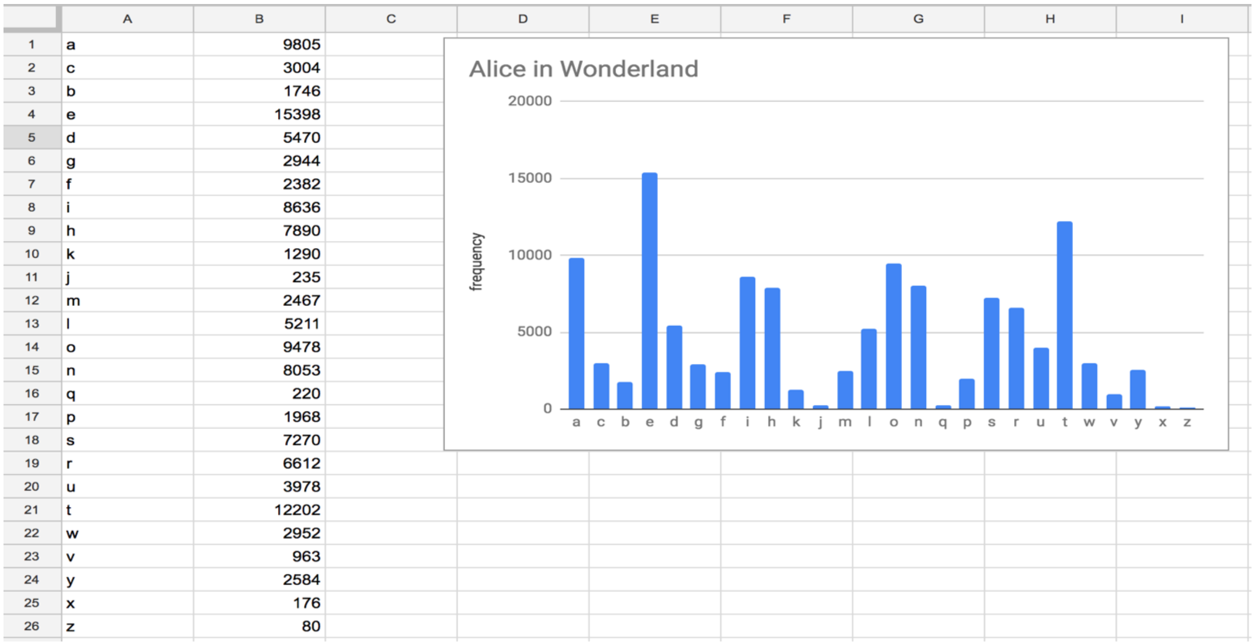


```
a,9805
c,3004
b,1746
e,15398
d,5470
g,2944
f,2382
i,8636
h,7890
k,1290
j,235
m,2467
l,5211
o,9478
n,8053
q,220
p,1968
s,7270
r,6612
u,3978
t,12202
w,2952
v,963
y,2584
x,176
z,80
```



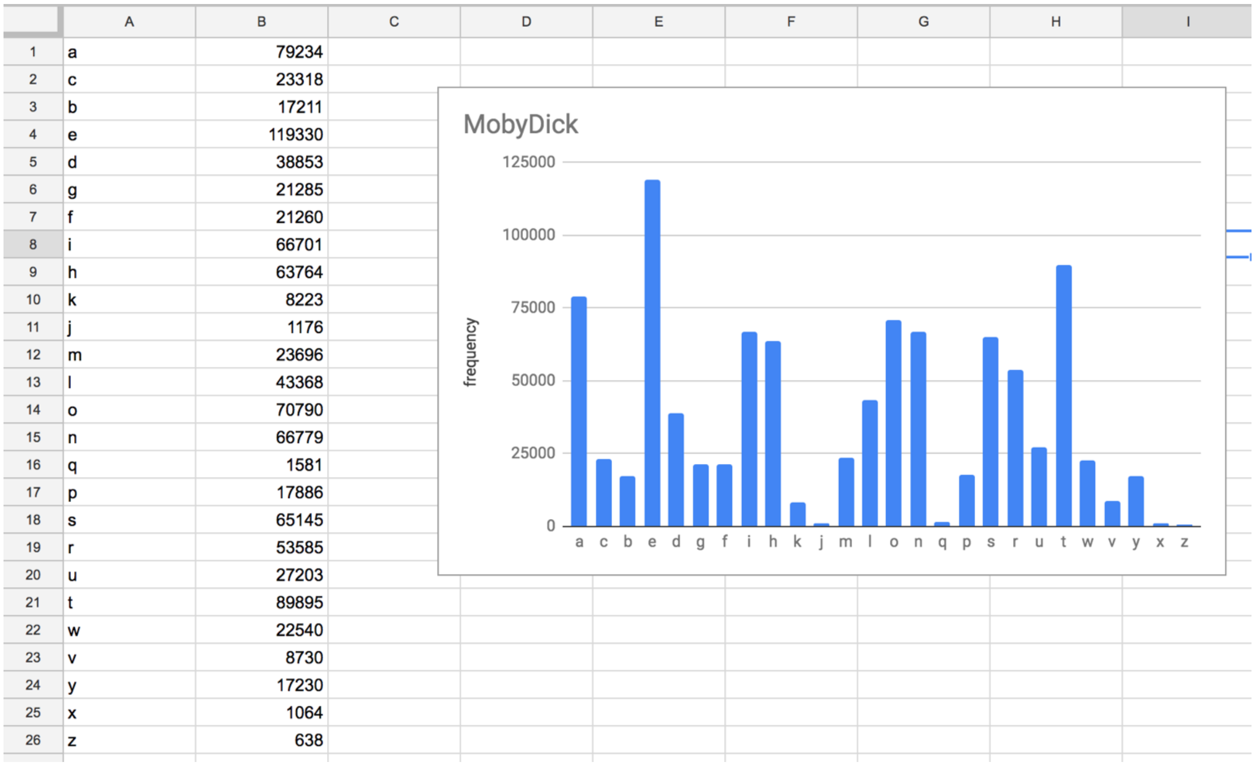
```
a,79234
c,23318
b,17211
e,119330
d,38853
g,21285
f,21260
i,66701
h,63764
k,8223
j,1176
m,23696
l,43368
o,70790
n,66779
q,1581
p,17886
s,65145
r,53585
u,27203
t,89895
w,22540
v,8730
y,17230
x,1064
z,638
```

- Alice in Wonderland spreadsheet with graph



	A	B
1	a	9805
2	c	3004
3	b	1746
4	e	15398
5	d	5470
6	g	2944
7	f	2382
8	i	8636
9	h	7890
10	k	1290
11	j	235
12	m	2467
13	l	5211
14	o	9478
15	n	8053
16	q	220
17	p	1968
18	s	7270
19	r	6612
20	u	3978
21	t	12202
22	w	2952
23	v	963
24	y	2584
25	x	176
26	z	80

- Moby Dick spreadsheet with graph



	A	B
1	a	79234
2	c	23318
3	b	17211
4	e	119330
5	d	38853
6	g	21285
7	f	21260
8	i	66701
9	h	63764
10	k	8223
11	j	1176
12	m	23696
13	l	43368
14	o	70790
15	n	66779
16	q	1581
17	p	17886
18	s	65145
19	r	53585
20	u	27203
21	t	89895
22	w	22540
23	v	8730
24	y	17230
25	x	1064
26	z	638