Scenario_3–Roman_Alphabet

December 1, 2022

1 Scenario 3: Generating English Words

1. Create a die of letters from 'A' to 'Z' with weights based on their frequency of usage. See 'Letters_and_Weights.txt' for a tab-separated data frame of letters and weights.

[1]:
```
import numpy as np
import pandas as pd

data_frame_of_letters_and_weights = pd.read_csv('Letter_Weights.txt', delimiter␣
 ↪= '\t', header = None).rename(columns = {0: 'letter', 1: 'weight'}).
 ↪astype({'letter': str, 'weight': np.float64})
data_frame_of_letters_and_weights
```

[1]:

	letter	weight
0	A	8.4966
1	B	2.0720
2	C	4.5388
3	D	3.3844
4	E	11.1607
5	F	1.8121
6	G	2.4705
7	H	3.0034
8	I	7.5448
9	J	0.1965
10	K	1.1016
11	L	5.4893
12	M	3.0129
13	N	6.6544
14	O	7.1635
15	P	3.1671
16	Q	0.1962
17	R	7.5809
18	S	5.7351
19	T	6.9509
20	U	3.6308
21	V	1.0074
22	W	1.2899
23	X	0.2902

```
24      Y   1.7779
25      Z   0.2722
```

[2]:
```
from montecarlosimulator import Die

def generate_die():
    array_of_faces = np.array(data_frame_of_letters_and_weights['letter'],␣
 ↪dtype = str)
    die = Die(array_of_faces)
    list_of_weights = data_frame_of_letters_and_weights['weight'].to_list()
    for i in range(0, len(array_of_faces)):
        face = array_of_faces[i]
        weight = list_of_weights[i]
        die.change_weight(face, weight)
        return die

die = generate_die()
die.show()
```

[2]:
```
   face  weight
0     A  8.4966
1     B  1.0000
2     C  1.0000
3     D  1.0000
4     E  1.0000
5     F  1.0000
6     G  1.0000
7     H  1.0000
8     I  1.0000
9     J  1.0000
10    K  1.0000
11    L  1.0000
12    M  1.0000
13    N  1.0000
14    O  1.0000
15    P  1.0000
16    Q  1.0000
17    R  1.0000
18    S  1.0000
19    T  1.0000
20    U  1.0000
21    V  1.0000
22    W  1.0000
23    X  1.0000
24    Y  1.0000
25    Z  1.0000
```

2. Play a game involving rolling 5 dice of letters from ‘A’ to ‘Z’ with weights based on their

frequency of usage and 1000 rolls.

[3]:
```
from montecarlosimulator import Game

list_of_dice = []
for i in range(0, 5):
    die = generate_die()
    list_of_dice.append(die)
game = Game(list_of_dice)
game.play(1000)
data_frame_of_rolls_and_dice = game.show('wide')
data_frame_of_rolls_and_dice
```

[3]:

roll_index	0	1	2	3	4
0	A	O	A	A	V
1	Y	A	A	Z	S
2	A	A	M	X	A
3	A	B	L	G	J
4	I	A	P	O	R
…	..	..	..	..	..
995	G	D	U	K	C
996	S	U	D	I	A
997	C	T	A	R	X
998	A	K	Q	A	Y
999	J	E	M	C	F

[1000 rows x 5 columns]

3. Generate 10 random samples of 10 rows each from the data frame of rolls and dice returned by the game showing a data frame of rolls and dice. Keep a running count; this will result in an estimate of the percent of English words in the data.

[4]:
```
for i in range(0, 10):
    sample = data_frame_of_rolls_and_dice.sample(n = 10, replace = True,␣
 ↪weights = None, random_state = None, axis = None, ignore_index = False)
    print(sample)
    print()
```

roll_index	0	1	2	3	4
228	S	A	L	H	B
305	Q	A	L	X	W
253	A	U	A	G	D
75	Z	A	Y	A	T
490	D	K	Z	A	A
517	G	B	C	R	A
820	U	Q	P	A	H
121	A	D	L	I	J

598	S	Q	J	Y	B
659	M	N	Y	J	A

roll_index	0	1	2	3	4
915	M	A	N	Q	S
42	J	G	R	A	Z
467	Z	A	A	A	L
6	K	V	A	A	A
86	E	V	X	K	A
376	C	D	H	E	A
947	O	V	H	A	Y
531	X	G	O	A	J
843	V	E	R	F	M
646	A	A	M	T	J

roll_index	0	1	2	3	4
847	Y	E	K	J	D
36	D	E	A	V	N
676	H	Q	J	R	G
999	J	E	M	C	F
800	I	A	H	P	H
964	A	A	S	T	Y
768	Y	U	I	C	V
217	M	G	A	Z	H
868	A	J	V	B	N
161	Q	I	U	L	A

roll_index	0	1	2	3	4
451	M	D	A	S	C
663	A	A	B	W	A
136	A	Y	E	A	A
107	B	J	T	L	R
832	Z	A	B	A	A
549	A	O	A	F	A
341	U	V	K	D	A
748	S	A	W	K	E
273	R	Y	C	F	A
548	O	S	A	X	G

roll_index	0	1	2	3	4
972	G	A	M	Q	T
427	I	A	Z	E	M
638	W	A	X	J	P
481	A	H	D	Z	T

927	S	E	A	M	B
109	P	D	A	T	R
107	B	J	T	L	R
14	A	U	R	U	A
876	I	A	D	Q	A
196	E	L	S	A	K

	0	1	2	3	4
roll_index					
678	A	H	U	Y	K
285	Q	C	K	A	G
916	J	Y	A	A	A
849	X	A	I	E	U
538	D	T	X	E	Q
605	A	Q	T	J	A
581	Z	R	A	T	Y
668	Z	Q	V	X	D
58	C	O	P	A	A
984	M	Z	U	Z	J

	0	1	2	3	4
roll_index					
859	D	E	A	U	F
686	A	C	O	R	B
477	W	L	V	S	Q
423	P	W	A	W	X
112	U	W	M	C	S
909	Q	W	A	V	N
631	T	R	K	J	A
685	S	A	U	K	A
284	B	C	F	A	A
87	S	L	A	X	A

	0	1	2	3	4
roll_index					
319	D	N	A	O	Y
738	C	C	Q	A	I
851	J	H	U	I	I
614	A	Q	H	A	Q
259	A	G	W	X	A
995	G	D	U	K	C
691	R	Q	Y	F	I
240	T	E	W	H	B
723	A	A	B	M	R
83	L	A	A	F	A

	0	1	2	3	4
roll_index					

```
958          R  U  A  X  A
177          J  W  N  H  I
699          K  I  T  M  M
704          K  A  A  W  G
684          A  I  A  C  A
921          B  D  D  I  C
564          Y  B  I  A  A
409          A  U  C  A  Z
631          T  R  K  J  A
332          U  Z  U  U  B

             0  1  2  3  4
roll_index
435          D  K  U  P  A
405          X  O  C  N  L
754          M  V  A  G  C
272          A  A  J  A  A
961          E  O  S  A  W
53           E  P  A  X  A
146          A  T  V  Y  T
234          A  R  Z  A  A
229          A  H  M  J  C
187          Q  H  U  A  F
```

By inspection and comparison with Tom Lever's vocabulary, the number of English words in the samples is 0. The estimated probability of a word in the samples being English

$$P = \frac{0}{100} = 0$$

[ ]: