

1> สร้าง function ชื่อ createDict ซึ่งทำหน้าที่:

- function รับ input เป็น list 2 ตัวที่มีขนาดเท่ากัน และ ค่า True/False
- ถ้า True จะสร้าง dict ที่มี key เป็นค่าจาก list ตัวที่ 1 และ value จาก list ตัวที่ 2
- ถ้า False จะสร้าง dict ที่มี key เป็นค่าจาก list ตัวที่ 2 และ value จาก list ตัวที่ 1
- output ของ function คือ dict ที่สร้างขึ้น

เช่น

output1 = createDict(['apple', 'banana'], [1, 2], True)

จะได้ output1 เป็น {'apple': 1, 'banana': 2}

output2 = createDict(['apple', 'banana'], [1, 2], False)

จะได้ output2 เป็น {1: 'apple', 2: 'banana'}

```
In [6]: def createDict(listtext, listnumber, logicV):
        if logicV: #logicV == True
            retDict = {}
            for index in range(len(listtext)):
                retDict[listtext[index]] = listnumber[index]
        else:
            retDict = {}
            for index in range(len(listtext)):
                retDict[listnumber[index]] = listtext[index]
        return retDict
```

สร้าง dict ใหม่
key value.
key : value.
'value' : 'key'
ถ้า T, F
กรณี False.

```
In [7]: createDict(['apple', 'banana'], [1, 2], True)
```

เสร็จ.

1> สร้าง function ชื่อ createDict ซึ่งทำหน้าที่:

- function รับ input เป็น list 2 ตัวที่มีขนาดเท่ากัน และ ค่า True/False
- ถ้า True จะสร้าง dict ที่มี key เป็นค่าจาก list ตัวที่ 1 และ value จาก list ตัวที่ 2
- ถ้า False จะสร้าง dict ที่มี key เป็นค่าจาก list ตัวที่ 2 และ value จาก list ตัวที่ 1
- output ของ function คือ dict ที่สร้างขึ้น

เช่น

output1 = createDict(['apple', 'banana'], [1, 2], True)

จะได้ output1 เป็น {'apple': 1, 'banana': 2}

output2 = createDict(['apple', 'banana'], [1, 2], False)

จะได้ output2 เป็น {1: 'apple', 2: 'banana'}

```
In [6]: def createDict(listtext, listnumber, logicV):
        if logicV: #logicV == True
            retDict = {}
            for index in range(len(listtext)):
                retDict[listtext[index]] = listnumber[index]
        else:
            retDict = {}
            for index in range(len(listtext)):
                retDict[listnumber[index]] = listtext[index]
        return retDict
```

```
In [7]: createDict(['apple', 'banana'], [1, 2], True)
```

```

def createDict1(l_text):
    for i in range(len(l_text)):
        retDict[l_number[i]] = l_text[i]
    return retDict

```

In [31]: createDict2(['apple','banana'],[1,2],True)

Out[31]: {'apple': 1, 'banana': 2}

2> Use "Wuhan_nCoV_GNome.txt" to answer these questions

ใช้ไฟล์นี้ตอบคำถามต่อไปนี้

- how long is the sequence? (ลำดับกรดอะมิโนนี้มีความยาวเท่าไร)
- how many time A, T, G, C in the sequence? (แต่ละกรดอะมิโน A, T, G, C นี้มีเท่าไร)
- what is the most frequent 5-char-long subsequence? (ลำดับกรดอะมิโนที่พบบ่อยที่สุดคืออะไร)

In [9]: import pandas as pd

In [11]: DNAseq = pd.read_csv('C:\\Users\\User\\Desktop\\Basic Programming by Aj.Ton\\Python_dataviz\\Wuhan_nCoV_GNome.txt',header=None)
DNAseq.head

Out[11]: <bound method NDFrame.head of
0 ATTAAGGTTTATACCTTCCAGGTAACAAACCAACCACTTTTGA...
1 CGAACTTTAAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTG...
2 TAATTACTGTGTTGACAGGACACGAGTAACCTGCTATCTTCTGC...
3 TTGCAGCCGATCATCAGCACATCTAGGTTTCTGTCGGGTGTGACCG...
4 CCTGGTTTCAACGAGAAAACACACGTCCAACCTCAGTTTGCCTGTTT...
... ..
423 GTGCAGAATGAATTCTGTAACATACATAGCACAGTAGATGATGTT...

[428 rows x 1 columns]>

2.1 how long is the sequence? (ลำดับกรดอะมิโนนี้มีความยาวเท่าไร)

In [22]: len('ATTAAGGTTTATACCTTCCAGGTAACAAACCAACCACTTTTGC')

Out[22]: 45

In [23]: DNAseq.shape

Out[23]: (428, 1)

In [24]: DNAseq.iloc[1][0]

Out[24]: 'CGAACTTTAAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTGCACTCAGCAGTATAATTAATAAC'

ความยาว
DNA

In [25]: DNAseq = 0
for row in range(DNAseq.shape[0]):
 DNAseq = DNAseq + len(DNAseq.iloc[row][0]) # DNAseq += len(DNAseq)
print(f'sequence is {DNAseq} characters long')

sequence is 29903 characters long

2.2 how many time A, T, G, C in the sequence? (แต่ละกรดอะมิโน A, T, G, C นี้มีเท่าไร)

In [29]: def countChar(seq,character):

charcount = 0

for char in seq:

if char == character:

charcount = charcount + 1

return charcount

จดดูค่าลงท.

เรียงแถว. มี 428 คอลัมน์ และ 1 row.

แถวที่ 1

ลำดับ.

→ ตัวแปร.

: เริ่มนับ + 1 ไปเรื่อยๆ

```
        charcount = charcount + 1
    return charcount
```

```
In [30]: countChar('ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCACTTTCG','A')
```

```
Out[30]: 16
```

```
In [32]: ANum = 0
         TNum = 0
         GNum = 0
         CNum = 0
         for row in range(DNAseq.shape[0]):
             ANum += countChar(DNAseq.iloc[row][0],'A')
             TNum += countChar(DNAseq.iloc[row][0],'T')
             GNum += countChar(DNAseq.iloc[row][0],'G')
             CNum += countChar(DNAseq.iloc[row][0],'C')

         print('There are {ANum} As, {TNum} Ts, {GNum} Gs, { CNum} Cs')

There are 8954 As, 9594 Ts, 5863 Gs, 5492 Cs
```

2.3 what is the most frequent 5-char-long subsequence? (ลำดับกรดอะมิโนที่มันติดกัน และซ้ำกันมากที่สุดคืออะไร)

```
In [ ]: 'ATTAAAGGTTT'
```

```
In [35]: def populate5charstring(seq):
         str_list = list()
         for i in range(len(seq)-4):
             str_list.append(seq[i:i+5])
         return str_list
```

ถ้าให้หา 5 ตัวติดกัน - 4
20 " ~ 19

```
In [36]: populate5charstring('ATTAAAGGTTT')
```

```
Out[36]: ['ATTAA', 'TTAAA', 'TAAAG', 'AAAGG', 'AAGGT', 'AGGTT', 'GGTTT']
```

```
In [36]: populate5charstring('ATTAAAGGTTT')
```

```
Out[36]: ['ATTAA', 'TTAAA', 'TAAAG', 'AAAGG', 'AAGGT', 'AGGTT', 'GGTTT']
```

```
In [37]: list_all_5char = list()
         for row in range(DNAseq.shape[0]):
             list_all_5char = list_all_5char + populate5charstring(DNAseq.iloc[row][0])
```

```
In [38]: len(list_all_5char)
```

```
Out[38]: 28191
```

```
In [39]: unique_5char = list(set(list_all_5char))
```

```
In [40]: len(unique_5char)
```

```
Out[40]: 1022
```

```
In [41]: dict_5char = {}
         for sub_str in unique_5char:
             num = 0
             for s in list_all_5char:
                 if sub_str == s:
                     num += 1
             dict_5char[sub_str] = num
```

```
In [42]: dict_5char
```

```
Out[42]: {'AGACC': 12,
          'AGCAT': 19,
          'ACAAG': 45,
          'CGTTC': 10,
          'CTGAA': 32,
          'GATAG': 13,
          'CCAGC': 10,
```

สุภา พร่งหงษ์ ทองจะ น่อง ไอ เนาะเงะ พน่อง อัน เทน ซื่อ อัน เทน สง พศุต มา เท เนาะเงะ เนาะเงะ',
'ถาม ว่า เอา ไป ได้ บั เอ้า ไป บั ได้ เทีย จะ เอ้ย น่อง ไอ สง มา ให้ บัก หล้าย เลย จะ นัน สะ',
'โห้ ได้ เบ่ง คลิป นี เนาะเงะ',
'เดี๋ยว ให้ เทน มา เป็ด อีก เท้า หลัง จะ เดี่ยว ให้',
'อัน นี คือ คือ บ้า น่อง ไอ น่อง ไอ เทน ซื่อ ไอ โว ดัน แล้ว นะ จะ เทน ซื่อ ไอ ดัน แล้ว บัดทีนี้ แม่ บ้าน นั้นจ อด รับ อ่า เอ็ม ปไรเซย์ จด รับ ส
ง มา โดน แล้ว บัดทีนี้ เทน กะ',
'เท้น กะ กรง โจ เฮา เฮา กะ กรง โจ เทน เด้จะ นี เพราะ ว่า อ่า เทน ซื่อ ไอ แล้ว เทน กะ กะ เลย สง มา ให้ จัง สี่ น้า',
'อ้อย ซื่อ ของ อีก โจ มา ให้ เทน เด้ น้า ไอ เอ้ย',
'อัน เทน บัก ลี เวอร์ พูล อยู่ แล้ว หัน น้า',
'นี่ เป็ด จัง สี่ คือ เอา อัน นี ออก จะ',
'อัน นี กะ เป็น ของ อะ เล็ก แล้ว นี อะ เล็ก เพราะ ว่า อะ เล็ก มั้ก เลย โ้ก หัน น้า ที่น้อง',
'อ้อย กะ ขอบ คุณ สุปัสสัน หลายเด้อ จะ',

975 rows x 2 columns

```
In [5]: Data.iloc[0]['ภาษาไทย'].replace("/", " ")
```

```
Out[5]: 'สวัสดี สู้สุดคน เต๋อ จ๋า'
```

```
In [6]: def replacestr(col1):  
    col = col1.replace("/", " ")  
    return col
```

```
In [7]: replacestr(Data.iloc[0,1])
```

```
Out[7]: 'สวัสดี ทุก ๆ คน นะ ค่ะ'
```

```
In [8]: list_1 = []  
list_2 = []  
for i in range(len(Data)):  
    list_1.append(replacestr(Data.iloc[i]['ภาษาไทย']))  
    list_2.append(replacestr(Data.iloc[i]['ภาษาอังกฤษ']))
```

```
In [9]: list_1
```

```
Out[9]: ['สวัสดี สู้สุดคน เต๋อ จ๋า',  
'กิน ข้าว ก่อน เต๋อ จ๋า ไหม สิ ถ้า เต๋อ คาวาน',  
'เต๋อ คาว กะ ขอ ให้ เต๋อ คาว อย่าง มี ความ สุข นะ จ๊ะ',  
'นี่ จ๋า อา ทาร เอ็น เป็น ลา ชาน ย่า กับ ผัก สด เต๋อ จ๋า',  
'ขอ ให้ เขียว แซ่ม เต๋อ ที่ นื่อง',  
'แมท ธิว บอก ว่า อี แม่ อย่า ทิ้ง หลับ เต๋อ',  
'เจ๊บบ แสงจิด',  
'ป มี ความ รื่น เริง อี หย้ง เลย ข้าง นอก',  
'มื่อ นี สิ เบ่ง พุ ตอน หก ทุ่ม จะ',  
'ถ้า ผ่น ตก ที่ มะ ตก คี สิ บ ได้ ออก ไป',  
'ปี ที่ แล้ว ออก ไป เบ่ง เขา จุด พุ มั่น เป็น ผ่น ที่ มะ นะ จะ',  
'เห็น กับ แบบ คน เม่า นะ จ๊ะ',  
'ถ้า ผ่น ตก ที่ มะ ตก คี สิ บ ได้ ออก ไป',  
'ปี ที่ แล้ว ออก ไป เบ่ง เขา จุด พุ มั่น เป็น ผ่น ที่ มะ นะ จะ',  
'เห็น กับ แบบ คน เม่า นะ จ๊ะ',  
'ก้ม ก้ม สอง พุ เห็น จุด ละ เห็น กะ ก้ม สอง เห็น บ่ เหย หน่า ขึ้น หัน นะ',  
'เห็น สิ โข กล้าม เห็น หัน นะ ที่ นื่อง',  
'โอ เค จ๋า สู้สุดคน มื่อ นี จัง ก็ ขอ คุณ สู้สุดคน ที่ แวะ เวียน เข้า มา ทัก ทาย กับ ครั้ว คุณ สม บุรณ',  
'จ๋า หือ กัน ใหม่ ใน คลิป วิ ดี โอ หน่า สวัสดี ปี ใหม่ บ้าย บาย จ๋า',  
'สະ หวัด ดี จ๋า สู้สุดคน เต๋อ จ๋า',  
'อ้อ ได้ รับ อัน 'ไปรษณีย์' เนาะ จ๋า 'ไปรษณีย์' เดี่ยว สิ มา บอก ว่า 'ไปรษณีย์' ของ ใคร',  
'เออ เห็น เขียน อัน ลุง อัน เขียน อัน อ...',  
'ลอง ลง ส่า หลาด เป็น เป็น ส่า มี ของ ที่ สาว เนาะ จะ ชึ่ง เอ้า เอ็น ป่า เออ นั้น หละ'.
```

```
In [10]: Data['ภาษาไทยใหม่'] = list_1  
Data['ภาษาอังกฤษใหม่'] = list_2
```

```
In [11]: Data
```

18	เออ/เห็น/เขียน/อัน/ลุง/อัน/ลุง/เขียน/อัน/...	เออ/เขา/เขียน/เอ้ม/ลุง/เอ้ม/ลุง/เขียน/เอ้ม/...	เออ เห็น เขียน อัน ลุง อัน ลุง เขียน อัน ...	เออ เขา เขียน เอ้ม ลุง เอ้ม ลุง เขียน เอ้ม...
19	ลุง/ลุง/ส่า/หลาด/เป็น/เป็น/ส่า/มี/ของ/ที่/ส่า/...	ลุง/ลุง/ส่า/หลาด/เป็น/เป็น/ส่า/มี/ของ/ที่/ส่า/...	ลุง ลุง ส่า หลาด เป็น เป็น ส่า มี ของ ที่ ส่า ...	ลุง ลุง ส่า หลาด เป็น เป็น ส่า มี ของ ที่ ส่า ...
20	โอ/เค/ละ/กะ/มา/ต่อ/อัน/เปิด/อัน/นี่/นะ/จะ/...	โอ/เค/แล้ว/ก็/มา/ต่อ/อัน/เปิด/อัน/นี่/นะ/จะ/...	โอ เค ละ กะ มา ต่อ อัน เปิด อัน นี่ นะ จะ ...	โอ เค แล้ว ก็ มา ต่อ เอ้ม เปิด อัน นี่ นะ จะ ...
21	สุ/ภา/พร/หงส์/ทอง/จะ/น้อง/โอ้/เนาะ/จะ/ที่/...	สุ/ภา/พร/หงส์/ทอง/จะ/น้อง/โอ้/เนาะ/จะ/ที่/...	สุ ภา พร หงส์ ทอง จะ น้อง โอ้ เนาะ จะ ที่ ...	สุ ภา พร หงส์ ทอง จ่า น้อง โอ้ เนาะ จ่า ทุก ...
22	ถาม/ว่า/เอา/ไป/ได้/ไป/เอา/ไป/ได้/ที่/จะ/...	ถาม/ว่า/เอา/ไป/ได้/ไป/เอา/ไป/ได้/ที่/จะ/...	ถาม ว่า เอา ไป ได้ ไป เอา ไป ไป ได้ ที่ จะ ...	ถาม ว่า เอา ไป ได้ ใหม่ เอา ไป ไป ได้ ที่ ...
23	ไหม/ที่/ได้/เบ่ง/คลิป/นี่/นะ/จะ/...	ใคร/ที่/ได้/ดู/คลิป/นี่/นะ/จะ/...	ไหม ที่ ได้ เบ่ง คลิป นี่ นะ จะ ...	ใคร ที่ ได้ ดู คลิป นี่ นะ จะ ...
24	เดี๋ยว/ให้/เห็น/มา/เปิด/อีก/ที่/หลัง/จ่า/เดี่ยว/ให้/...	เดี๋ยว/ให้/เขา/มา/เปิด/อีก/ที่/หลัง/จ่า/เดี่ยว/ให้/...	เดี๋ยว ให้ เห็น มา เปิด อีก ที่ หลัง จะ เด ...	เดี๋ยว ให้ เขา มา เปิด อีก ที่ หลัง จ่า เดี่ยว ให้ ...
25	อัน/นี่/คือ/คือ/คือ/นี่/น้อง/โอ้/น้อง/โอ้/ที่/...	อัน/นี่/คือ/คือ/คือ/นี่/น้อง/โอ้/น้อง/โอ้/ที่/...	อัน นี่ คือ คือ คือ นี่ น้อง โอ้ น้อง โอ้ ที่ ...	อัน นี่ คือ คือ คือ นี่ น้อง โอ้ น้อง โอ้ ...
26	เห็น/กะ/เกรง/ใจ/เขา/ส่า/กะ/เกรง/ใจ/เขา/นั่น/...	เขา/ก็/เกรง/ใจ/เขา/เรา/ก็/เกรง/ใจ/เขา/นั่น/...	เห็น กะ เกรง ใจ เขา ส่า กะ เกรง ใจ เห็น ...	เขา ก็ เกรง ใจ เขา เรา ก็ เกรง ใจ เขา นั่น ...
27	อ้อย/ชื่อ/ของ/ถูก/ใจ/มา/ให้/เขา/เนาะ/มา/โอ้/...	อ้อย/ชื่อ/ของ/ถูก/ใจ/มา/ให้/เขา/เนาะ/มา/โอ้/...	อ้อย ชื่อ ของ ถูก ใจ มา ให้ เขา เนาะ มา โอ้ ...	อ้อย ชื่อ ของ ถูก ใจ มา ให้ เขา เนาะ มา โอ้ ...

```
974 ข้อยก/กระ/เคย/ดต/สน/เจ/กน/ญ/หว่า/เออ/... ข้อยก/กระ/เคย/ดต/สน/เจ/กน/ญ/หว่า/เออ/... ข้อยก/กระ/เคย/ดต/สน/เจ/กน/ญ/หว่า/เออ/...
975 rows x 4 columns

In [12]: Data.iloc[0][1]
Out[12]: 'สวัสดี/ทุก ๆ คน/นะ/ค่ะ'

In [13]: len(Data.iloc[0][1])
Out[13]: 21

In [14]: def splitstr(col1):
          col = col1.split('/')
          return col
          # ส่วน col ใหม่

In [15]: splitstr(df.iloc[0]['ภาษาไทยอีสาน'])
Out[15]: ['พัก',
          'หาย',
          'กัน',
          'ขี้มบายดี',
          'สวัสดี',
          'วัน',
          'ล้ง',
          'คาร',
          'เนาะ',
          'จ้า',
          'อา',
          'ภาค',
          'มีอ',
          'ณ',
          'หนาว',
          'พี',
          '...
```

สร้าง list ใหม่

```
In [16]: list_11 = []
          list_22 = []
          for i in range(len(Data)):
              if len(splitstr(Data.iloc[i]['ภาษาไทยอีสาน'])) == len(splitstr(Data.iloc[i]['ภาษาไทย'])): # คัดก่อนแล้วค่อยดู
                  list_11 += splitstr(Data.iloc[i]['ภาษาไทยอีสาน']) # list_11 = list_11 + splitstr(Data.iloc[i]['ภาษาไทยอีสาน'])
                  list_22 += splitstr(Data.iloc[i]['ภาษาไทย'])

In [17]: len(list_11)
Out[17]: 30852

In [18]: len(list_22)
Out[18]: 30852

In [19]: def google(col1,col2):
          str = f'{col1} แปลว่า {col2}'
          return str

In [25]: AA = []
          for i in range(len(list_11)):
              if len(list_11[i]) == len(list_22[i]): # ไม่ต้องยาว
              AA += [google(list_11[i],list_22[i])]

In [26]: Uminmin = set(AA)

In [27]: Uminmin
Out[27]: {'ขะ แปลว่า ขะ',
          'ขอนแก่น แปลว่า ขอน',
          'ขวด แปลว่า ขวด',
          'บริมาณ แปลว่า ปริมาณ',
          ...}
```