FiveThirtyEight's May 22, 2020 Riddler

Emma Knight

May 24, 2020

This week's riddler, due to Mark Bradwin, is about interesting word properties:

Question 1. *Ohio is the only state whose name doesn't share any letters with the word "mackerel." It's strange, but it's true.*

But that isn't the only pairing of a state and a word you can say that about it's not even the only fish! Kentucky has "goldfish" to itself, Montana has "jellyfish" and Delaware has "monkfish," just to name a few.

What is the longest "mackerel?" That is, what is the longest word that doesn't share any letters with exactly one state? (If multiple "mackerels" are tied for being the longest, can you find them all?)

Extra credit: Which state has the most "mackerels?" That is, which state has the most words for which it is the only state without any letters in common with those words?

There isn't that much mathematical insight to be provided, so I will just give my code and answers.

```
import time

##Here are a bunch of useful functions.  The first one splits
##a string into its characters.  The second one tests wheter
##two words (state and word) have no letters in common.  The
##third one takes in a word and a list of states and spits out
##all of the mackerel matches between the word and the list of
##states.  Finally, the fourth one turns a file of words into
##a list of the words in it.

def ToChars(str):
    return list(str)

def MackerelMatch(state, word):
    ret = True
    lets = ToChars(state.lower())
    letw = ToChars(word.lower())
    for i in range(len(lets)):
        for j in range(len(letw)):
            if lets[i] == letw[j]:
                ret = False
    return(ret)

def MackerelList(word, states):
    stc = len(states)
    results = []
    for i in range(stc):
        if MackerelMatch(states[i], word):
            results.append([states[i], i])
    return(results)

def WordList(file):
    f = open(file, ’r’)
    output = []
    for line in f:
        length = len(line)
        output.append(line[:length-1])
    return output

##Here are some basic variables for the main loop.  states
##and words are lists of the states/words respectively, and
##are just read from files.  counts is a list of how many
##mackerels there are for a given state.  winninglength is
##the length of the winning mackerel word, winner is the
##word that won, statewinner is the state corresponding to
##the state that won, ties is a list of all the other word
##-state combos that had the same length, winningstatecounts
##is the amount of mackerels for the state with the most
##mackerels, and stateties is a list of all the states with
##the most mackerels.

states = WordList(’states.txt’)
words = WordList(’words.txt’)

counts = []

for i in range(len(states)):
    counts.append(0)

winninglength = 0
winner = ’’
statewinner = ’’
ties = []
```

```
winningstatecounts = 0
stateties = []

##This is the main loop.  It finds all mackerels and updates
##the relevant variables.

for w in words:
    l = MackerelList(w, states)
    if (len(l) == 1):
        counts[l[0][1]] += 1
        if (len(w) == winninglength):
            ties.append([w, l[0][0]])
        if (len(w) > winninglength):
            ties = []
            winninglength = len(w)
            winner = w
            statewinner = l[0][0]

##This is the display loop.  It displays all of the relevant
##information.  The drumrolls are CRUCIAL to the functioning
##of this code, and adding more is recommended if you want to
##run this code yourself.

print(’And the winning mackerel is’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
```

```
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(’(drumroll)’)
time.sleep(1)
print(winner, ’, for the state of ’, statewinner, ’.’, sep=’’)
if (len(ties) > 0):
    print(’Congratulations to all the ties which lost the tiebreaker of alphabeticalness:’)
    for i in range(len(ties)):
        print(ties[i][1], ’with a mackerel of’, ties[i][0])

for i in range(len(states)):
    if (counts[i] == winningstatecounts):
        stateties.append(i)
    if (counts[i] > winningstatecounts):
        stateties = [i]
        winningstatecounts = counts[i]

if (len(stateties) == 1):
    print(’The state of ’, states[stateties[0]], ’ had the most mackerels, with a total of ’,
          winningstatecounts, ’ mackerels.’, sep=’’)
if (len(stateties) > 1):
    print(’The following states all had the most mackerels, totalling’, winningstatecounts)
    for i in stateties:
        print(states[i])
```

This code gave two longest mackerels: counterproductivenesses for Alabama and hydrochlorofluorocarbon for Mississippi. It also said that the state of Ohio had the most mackerels, at 11342.

Convieniently, nothing is hardcoded in this code, so it’s really easy to switch to a different set of states or a different set of words. Canada had otorhinolaryngoligists as a mackerel for Quebec as the longest mackerel, and Yukon had the most mackerels at 28714. Spain had four longest mackerels: caricaturizaríamos, circunscribiésemos, and superentendisemos for Álava, and desinteresadamente for Lugo. Lugo had the most mackerels at 19757. Mexico had independientemente as a mackerel for Oaxaca as the longest mackerel, and the state of Mexico had the most mackerels at 13777. Germany had Programkonfiguration as a mackerel for Hesse. Hamburg had the most German mackerels at 4031. Finally, France had five(!) longest mackerels: dénationalisassions, désillusionnassions, individualisassions, and échantillonnassions for Berry, and transcendantalement for Foix. Foix had the most mackerels at 18176.

Disclaimer: I am not 100% certain that the mackerels for languages that aren’t English are correct. I had issues with invisible formatting in my state lists, and while I think I solved them, it’s hard to know for certain and also hard to know if something got into the word lists.

I got my English word list from here, and got my other word lists from here. The word lists in the second link required a bit of post-processing.