FE520 Assignment 2

Dan Wang, Zhiyuan Yao

September 2018

1 Problem 1

1.1 (25pt)

You are required to write a function which can take a string as its argument, and return a dictionary. 1) This function will split this string by blank space into a list of sub-string, 2) then turn each element in this list into lower case without any punctuation (including special characters like '\n', '\t', etc.) and empty string. 3) After this, compute the frequency of each element in this list, and make a dictionary whose key is the element (sub-string) and value is the frequency. 5) Use string formatting to print out a sentence describing the most frequent word and its frequency in this string. 6) Return the dictionary you created in step 4.
Here is an example,

```
>>> myString = '''
    This course is designed for those students have no experience or
    limited experience on Python. This course will cover the basis
    syntax rules, modules, importing packages (Numpy, pandas), data
    visualization, and Intro for machine learning on Python. You will
    need to implement what you learn from this course to do a finance
    related project. This course aims to get you familiar with Python
    language, and can finish a simple project with Python.
    '''

>>> ret = myFun(myString)
    The most frequent key is 'this', its frequency is 4.
>>> print(ret)
{'this': 4, 'course': 4, 'is': 1, 'designed': 1, 'for': 2,
'those': 1, 'students': 1, 'have': 1, 'no': 1, 'experience': 2,
'or': 1, 'limited': 1, 'on': 2, 'python': 4, 'will': 2,
'cover': 1, 'the': 1, 'basis': 1, 'syntax': 1, 'rules': 1,
'modules': 1, 'importing': 1, 'packages': 1, 'numpy': 1,
'pandas': 1, 'data': 1, 'visualization': 1, 'and': 2,
'intro': 1, 'machine': 1, 'learning': 1, 'you': 3, 'need': 1,
'to': 3, 'implement': 1, 'what': 1, 'learn': 1, 'from': 1,
```

```
'do': 1, 'finance': 1, 'related': 1, 'project': 2, 'aims': 1,
'get': 1, 'familiar': 1, 'with': 2, 'language': 1, 'can': 1,
'finish': 1, 'simple': 1}
```

Hint: You can replace all punctuation with blank space before spliting this string.

1.2 (25pt)

Create a class called Rectangular, with two data attributes: length and width. They should be assigned in constructor (__init__()) It should also have two function attributes called area() and perimeter() which return area and perimeter of this rectangular respectively. Here is an example of the class structure:

```
>>> class Rectangular:
...     <Your initializer>
...     <Defination of area()>
...     <Defination of perimeter()>
>>> myRec = Rectangular(10,20)
>>> print(myRec.area())
200
>>> print(myRec.perimeter())
60
```

(5pt Bonus) Write an inherited class called Square from Rectangular, you may only override the constructor, let it only accept one parameter: length. You need to call its superclass's constructor in its constructor, instead of assigning parameters to data attributes.

2 Problem 2: Ordinary Differential Equations. (50pts)

Consider a general one dimensional ODE of the form $\dot{x} = f(x)$, $x(0) = x_0 \in \Re$, where $f : \Re \to \Re$. We say that $x^* \in \Re$ is a stationary point if $f(x^*) = 0$. Furthermore, if $k'(x^*) < 0$, we say that $x^* \in \Re$ is a stable point. if $k'(x^*) > 0$, we say that $x^* \in \Re$ is a unstable point. The purpose of this assignment is to solve numerically the ODE using two methods, Forward Euler and 4th Order Runge-Kutta, and to find numerically the equilibrium points using the bisection method.

2.1 10 pts

Consider the ODE

$$\dot{x} = x + 4 - exp(x) \tag{1}$$

Thus $f(x) = x+4-exp(x)$. Using a numerical ODE solver (forward Euler or Runge–Kutta methods), find the solution of this ODE for initial conditions $x_0 = 0, x_0 = 4$. Check that the algorithm converges (you can increase the number of iterations and check that you get same value, or you can use a do while loop). Does the ODE have an equilibrium point? (plot the function using Google and look for the zeros).

To understand this problem, we can assume we start from $x_0 = 0$, then we get $f(0) = 3$, which push the x_1 to the right. or we assume we get $x_{10} = 3$, then we get $f(3) = -13.08$, which pull the x_{11} to the left. With the increase n in x_n, you may observe a your result converge to a final value.

2.2 15 pts

Find the stationary points of the ODE (using Bisection method)

$$\dot{x} = sin(x) \cdot exp(x) \tag{2}$$

both stable and unstable on the interval [2; 4]. Hint: Plot out the function $\dot{x} = sin(x) \cdot exp(x)$. You can use Google and get a rough estimate of where the fixed points should be, and whether they are stable or unstable.

2.3 25 pts

Using a numerical solver (either Forward Euler or Runge–Kutta methods or bisection method) find the fixed points, both stable and unstable for the following differential equation, with $x \in [-4; 4]$:

$$\dot{x} = x \cdot sin(x) - 1 \tag{3}$$

In this section, you need to define a function with several input arguments, and output a dictionary with index name equal to 'stable' and 'unstable',.

Hint: Solve the ODE for different initial conditions $x_0 \in [-4; 4]$. For some initial points, the solution should converge to an equilibrium point. The stable fixed points are easier to find, while estimating the unstable will be more difficult.

Forward Euler Method: If we have a differential equation defined by

$$\dot{x} = f(x) \tag{4}$$

then given an initial condition of $x(t0) = x_0$ and $\delta t > 0$ the forward Euler algorithm is described as

$$x(t_n + \delta t) = x(t_{n+1}) = x(t_n) + \delta t \cdot f(x(t_n)) \tag{5}$$

We see that the Forward Euler method is a straight forward application of linear approximation.

3 Bonus Question

Finish the problem 9 (Palindrome Number) and problem 13 (Roman to Integer) on Leetcode. Submit a screen-shot of your submission detail where you can see a run time distribution, and submit the file code also.

Submission Requirement:

For all the problems in this assignment you need to design and use Python 3, output and present the results in nicely format. Please submit a written report (pdf), where you detail your results and copy your code into an Appendix. You are required to submit a single python file and a brief report. Your grade will be evaluated by combination of report and code. You are strongly encouraged to write comment for your code, because it is a convention to have your code documented all the time. In your python file, you need contain both function and test part of function. Python script must be a '.py' script, Jupyter notebook '.ipynb is not allowed. Do NOT copy and paste from others, all homework will be firstly checked by plagiarism detection tool.