Big Data; 02 Practical - Python Basics 2

Ryan Greenup

July 23, 2020

Contents

Calculate Data Statistics

Sum Values

So what we need to do is go through each value and tally it up, it is important however to return the value and the print it:

```
def mysum(x):
    total = float(0)
    for i in x:
        total = total + float(i)
    return total

value = mysum([1,2,3])
print(value)
```

```
6.0
```

Improvements

We could however improve this by using a `try` / `except` test, in the event that a non-numerical list is provided:

```
def mysum(x):
    total = float(0)
    for i in x:
        try:
            total = total + float(i)
        except:
            print("The Values of the list must be numeric")
            print("Discarding Value")
    return total

value = mysum([1,2,3, "apple"])
print(value)
```

```
The Values of the list must be numeric
6.0
```

Minimum Value

Take the first item of the list as a candidate, for every item in the list, compare it to the candidate, if the next value is bigger that will become the new candidate, finally the candidate will be the maximum value.

Just like above we use a `try` / `except` to prevent issues.

```python
def mymax(list_of_vals):
    candidate = list_of_vals[0]  # Unlike R/Mathematica/Julia, python starts from 0.
    for i in list_of_vals:
        try:
            if i > candidate:
                candidate = float(i)
        except:
            print("The list items must be numeric")
            print("Discarding Value")
    return candidate

print(mymax([4, 6, 2, 5, 7, 3, 8, "john", -9]))
```

```
The list items must be numeric
Discarding Value
8.0
```

Maximum Value

Same as above, just remember to:

- wrap in `float()` as appropriate
- print the function call.

```python
def mymin(thelist):
    candidate = thelist[0]
    for i in thelist:
        try:
            if float(i) > float(candidate):
                candidate = i
        except:
            print("list items must be numeric, discarding value")
    return candidate

value = mymin([1, 5, 3, "apple", 8, 2, -9])
print(value)
```

```
list items must be numeric, discarding value
8
```

Vector Norm, Inner Product and Distance

Piping

The *Toolz* Module gives something very similar to piping in bash / julia / R

Using Built ins

```
import math as mt
import copy

## because python counts from 0 indexing is confusing,
## the count will come back as 4, but the indexes will be 0, 1, 2 and 3.

def getNorm(x):
    total = 0
    for i in range(len(x)):
        total=x[i-1]**2+total
    return mt.sqrt(total)

print(len([1,2,3,4]))


xvec = [0, 1, 2, 3, 4]
yvec = [4, 3, 2, 1, 0]

norm = getNorm(xvec);            print(norm)
```

```
4
5.477225575051661
```

Inner Product

```
def getInnerProd(x, y):
    z = copy.deepcopy(x)    ## Careful, you need to copy, not just
    ↪ assign
    if len(x) == len(y):
        for i in range(len(x)):
            z[i] = x[i]*y[i]
        return sum(z)
    else:
        print("The vectors must have the same dimension")
xvec = [0, 1, 2, 3, 4]
yvec = [4, 3, 2, 1, 0]

norm = getNorm(xvec);             print(norm)
norm = getNorm(yvec);             print(norm)
prod = getInnerProd(xvec, yvec); print(prod)
```

```
5.477225575051661
5.477225575051661
10
6.324555320336759
```

Distance

```
def getDist(x, y):
    if len(x) == len(y):
        z = mt.sqrt(getNorm(x)**2 + getNorm(y)**2 - 2 * getInnerProd(x,
        ↪ y))
        return z
    else:
        print("The vectors must have the same dimension")


xvec = [0, 1, 2, 3, 4]
yvec = [4, 3, 2, 1, 0]

norm = getNorm(xvec);             print(norm)
norm = getNorm(yvec);             print(norm)
prod = getInnerProd(xvec, yvec); print(prod)
dist = getDist(xvec, yvec);       print(dist)
```

Vote Counting

```
votes = "N , Y, Y,N,n , N , N  N ,n ,y, n,N,Y, y,Y,N , N , n ,y,N"
def countVotes(ballot):
    ballot = ballot.replace(",","").replace(" ", "").upper()
    neg_ballots = ballot.count("N")
    pos_ballots = ballot.count("Y")

    # Could also have used a loop
    print(str(pos_ballots) + " Yes votes and " + str(neg_ballots) + "
    ↪  No votes")

countVotes(votes)
votes = ",,yyyyn,,y,y,nn,y,,nn,y"
```

```
7 Yes votes and 13 No votes
```

Word Capitaliser

```
def capitalise(sentence):
    ## Split the words into a list
    wordsList = sentence.split()
    ## These are escape words
    EscWords = ["am", "a", "an", "the", "am", "is", "are", "and", "of",
    ↪  "in", "on", "with", "from", "to"]
    ## The number of words starting from 0
    for i in range(len(wordsList)-1):
        ## if not in the escape words
        if i not in EscWords:
            ## replace the ith word for a capitalized one
            wordsList[i] = wordsList[i].capitalize()
    ## Take a space and use it to join the list together
    sentence_Capitalized = " ".join(wordsList)
    ## Print the output
    print(sentence_Capitalized)
    return sentence_Capitalized

capitalise("The quick brown fox jumped over the lazy dogs")
```

```
The Quick Brown Fox Jumped Over The Lazy dogs
```

Parse File

Set up the Text File

Take the following text:

> Unit ID, unit name, course name 301046, Big Data, MICT 300581, Programming Techniques, BICT 300144, Object Oriented Analysis, BICT 300103, Data Structure, BCS 300147, Object Oriented Programming, BCS 300569, Computer Security, BIS 301044, Data Science, MICT 300582, Technologies for Web Applications, BICT

Let's write it to a file:

```
pwd
ls
```

```
scemunits = """Unit ID, unit name, course name
301046, Big Data, MICT
300581, Programming Techniques, BICT
300144, Object Oriented Analysis, BICT
300103, Data Structure, BCS
300147, Object Oriented Programming, BCS
300569, Computer Security, BIS
301044, Data Science, MICT
300582, Technologies for Web Applications, BICT"""

def writeTextFile(text, filename) :
    f = open(filename, 'w')
    for i in text.split('\n'):
        f.writelines(i+'\n')
    f.close()

writeTextFile(scemunits, 'scemunits.txt')
```

In order to check that worked we can run cat from *Python*:

```
import subprocess
MyCommand = "cat scemunits.txt"
scemunits_txt = subprocess.run(MyCommand.split(), capture_output = True)
print(scemunits_txt)
```

```
CompletedProcess(args=['cat', 'scemunits.txt'], returncode=0, stdout=b'Unit ID, unit name, cou
```

Observe that:

1. The """ are necessary for new line strings
2. The `open(file, w)` will write over any pre-existing file (like > in `bash`)
   (a) usint `open(file,a)` would append to a file (like » in `bash`)
3. Nothing is written to disk until apter `f.close()`, that's when the changes go from memory to disk.

Parse the Text File

Read the Text File

```
## Open the File
scemunits_fid = open('./scemunits.txt')

## Dispense with the first line
header = scemunits_fid.readline()

## Read the remaining Lines into a var
scemunits_txt = scemunits_fid.read()

## Print what we have
print(scemunits_txt)

## Close the file
scemunits_fid.close()
```

```
301046, Big Data, MICT
300581, Programming Techniques, BICT
300144, Object Oriented Analysis, BICT
300103, Data Structure, BCS
300147, Object Oriented Programming, BCS
300569, Computer Security, BIS
301044, Data Science, MICT
300582, Technologies for Web Applications, BICT
```

Return only Matching Data

```
## Split each line into a list element
obs = scemunits_txt.split('\n')

## Throw away the empty line
obs = list(filter(None, obs))

## Get the Course Names
    ## Use replace so whitespace is not required after ,
courses = [ obs[i].replace(', ', ',').split(',')[2] for i in range(len(obs)) ]
units = [ obs[i].replace(', ', ',').split(',')[1] for i in range(len(obs)) ]

## Enumerate the obs so that they
obs = list(obs)

## Make an empty list for the matches
matches = []

##
for i in range(len(obs)):
    ## Don't Require whitespace after comma
    if courses[i] == "MICT":
        matches.append(obs[i])


#print([header] + join(matches).insert(header))
#print([header].append(matches))
print(matches)
matches.insert(0, header.replace('\n',''))
print("\n".join(matches))

out_fid = open('outfile.txt', "w")
# out_fid.write("\n".join(matches))
for i in matches:
    out_fid.write(i+'\n')
    print(i)

out_fid.close()
```

```
['301046, Big Data, MICT', '301044, Data Science, MICT']
Unit ID, unit name, course name
301046, Big Data, MICT
301044, Data Science, MICT
```

```
Unit ID, unit name, course name
301046, Big Data, MICT
301044, Data Science, MICT
```

We can now inspect the contents of that file:

Wrap it into a function

```
#!/usr/bin/env python3

# * Create the Text File
scemunits = """Unit ID, unit name, course name
301046, Big Data, MICT
300581, Programming Techniques, BICT
300144, Object Oriented Analysis, BICT
300103, Data Structure, BCS
300147, Object Oriented Programming, BCS
300569, Computer Security, BIS
301044, Data Science, MICT
300582, Technologies for Web Applications, BICT"""

def writeTextFile(text, filename) :
    f = open(filename, 'w')
    for i in text.split('\n'):
        f.writelines(i+'\n')
    f.close()

writeTextFile(scemunits, 'scemunits.txt')


# * Main Functions
def readWriteFile(infile, outfile):
    readTheTextFile(infile)
    listOfLines = returnMatchingData(outfile)
    print(listOfLines)
    writeToFile(listOfLines, outfile)


# ** Sub Functions
# *** Input
def readTheTextFile(infile):
    ## Open the File
    scemunits_fid = open(infile)
    ## ////////////////// File Open ///////////////////////

    ## Dispense with the first line
    readTheTextFile.header = scemunits_fid.readline().replace('\n', '')

    ## Read the remaining lines into an attribute
    readTheTextFile.scemunits_txt = scemunits_fid.read()

    ## Close the File
    ## ////////////////// File Closed ///////////////////////
    scemunits_fid.close()


# ** Output
```

```
['Unit ID, unit name, course name', '301046, Big Data, MICT', '301044, Data Science, MICT']
Unit ID, unit name, course name
301046, Big Data, MICT
301044, Data Science, MICT
```

and to confirm that it has written to the file:

```
cat outfile.txt
```

Parse Dictionary

```python
#!/usr/bin/env python3

# * Create the Dictionary

units = {('301046', 'Big Data'): 'MICT',
         ('300581', 'Programming Techniques'): 'BICT',
         ('300144', 'OOA'): 'BICT',
         ('300103', 'Data Structures'): 'BCS',
         ('300147', 'OOP'): 'BCS',
         ('300569', 'Computer Security'): 'BIS',
         ('301044', 'Data Science'): 'MICT',
         ('300582', 'TWA'): 'BICT'}


def displayUnits(unitsDict, keyword):
    # Should Return Gracefully if the input is wrong
    # Could have used Try/Except
    if type(unitsDict) != dict:
        print("ERROR; Require Dictionary of Unit Values")
        return
    # Make an empty List to fill
    matches = []
    # For each dictionary item if it corresponds to the keyword
    # append it to the list
    for i in unitsDict:
        if units[i] == keyword:
            matches.append(i)
    # Use to get back matches[][1]
    matching_units = [matches[i][1] for i in range(len(matches))]
    # Return the Value
    return matching_units


# To Print the Values join the list together with new line characters.
# The function should return data in a list not a string
# (python => data, bash => string)

print("Match MICT \n ----------------------")
print("\n".join(displayUnits(units, 'MICT')))

print("Match BCS  \n ----------------------")
print("\n".join(displayUnits(units, 'BCS')))
```

```
Big Data
Data Science
Data Structures
OOP
```

grep

This is easy, just loop through the lines and print if the word is in the line.

```python
#!/bin/python

def pygrep(filename, expr):
    try:
        inputfile_fid = open(filename)
    except:
        print("ERROR: Could note open file")
    for line in inputfile_fid:
        if expr in line:
            print(line)

pygrep("./bigdata.txt", 'Big')
pygrep("./bigdata.txt", 'technology')
```

```
Big data is a broad term for data sets so large or complex that they are difficult to process u
their tools, and expanding capabilities make Big Data a moving target. Thus, what is considered
Big data usually includes data sets with sizes beyond the ability of commonly used software too
curate, manage, and process data within a tolerable elapsed time. Big data "size" is a constant
target, as of 2012 ranging from a few dozen terabytes to many petabytes of data. Big data is a
definition as follows: "Big data is high volume, high velocity, and/or high variety informatio
Big data uses inductive statistics and concepts from nonlinear system identification to infer l
Big data can also be defined as "Big data is a large volume unstructured data which cannot be h
Big data can be described by the following characteristics: Volume  The quantity of data that i
data under consideration and whether it can actually be considered as Big Data or not. The name
Variety - The next aspect of Big Data is its variety. This means that the category to which Big
upholding the importance of the Big Data.
```

complexity of Big Data.

Big data analytics consists of 6 Cs in the integrated industry 4.0 and Cyber Physical Systems e

implications in an article titled "Big Data Solution Offering". The methodology addresses handl

Big Data Analytics for Manufacturing Applications can be based on a 5C architecture (connectior

Big Data Lake - With the changing face of business and IT sector, capturing and storage of data

Big data requires exceptional technologies to efficiently process large quantities of data witl

Bus wrapped with SAP Big data parked outside IDF13.

Big data has increased the demand of information management specialists in that Software AG, Or

While many vendors offer off-the-shelf solutions for Big Data, experts recommend the developmer

The use and adoption of Big Data, within governmental processes, is beneficial and allows effic

Governmental Big Data space.

In 2012, the Obama administration announced the Big Data Research and Development Initiative, t

different big data programs spread across six departments.Big data analysis played a large role

Big data analysis was, in parts, responsible for the BJP and its allies to win a highly success

benefit of big data for manufacturing. Big data provides an infrastructure for transparency in

In order to hone into the manner in which the media utilises Big Data, it is first necessary to

Practitioners in Advertising and Media approach Big Data as many actionable points of informati

The media industries process Big Data in a dual, interconnected manner:

Big Data and the IoT work in conjunction. From a media perspective, Data is the key derivative

far-reaching impacts on media efficiency. The wealth of data generated by this industry (i.e. I

Engineering Education. Gautam Siwach engaged at Tackling the challenges of Big Data by MIT Comp

In March 2012, The White House announced a national "Big Data Initiative" that consisted of six

The White House Big Data Initiative also included a commitment by the Department of Energy to p

The U.S. state of Massachusetts announced the Massachusetts Big Data Initiative in May 2012, wh

Massachusetts Institute of Technology hosts the Intel Science and Technology Center for Big Dat

The European Commission is funding the 2-year-long Big Data Public Private Forum through their

presenters from various industrial companies discussed their concerns, issues and future goals

Computational social sciences Anyone can use Application Programming Interfaces (APIs) provide

emergence of the typical network characteristics of Big Data". In their critique, Snijders, Mat

Big data has been called a "fad" in scientific research and its use was even made fun of as an

Questions for Big Data", the authors title big data a part of mythology: "large data sets offer

often "lost in the sheer volume of numbers", and "working with Big Data is still subjective, an

such as pro-active reporting especially target improvements in usability of Big Data, through a

Big data analysis is often shallow compared to analysis of smaller data sets. In many big data

Big data is a buzzword and a "vague term", but at the same time an "obsession" with entrepreneu

consultants, scientists and the media. Big data showcases such as Google Flu Trends failed to c

election predictions solely based on Twitter were more often off than on target. Big data often

technology went public with the launch of a company called Ayasdi.

ICT4D) suggests that big data technology can make important contributions but also present unic

the later utilization stage. Finally, with ubiquitous connectivity offered by cloud computing t

well as queries from more than half a million third-party sellers. The core technology that kee

form at cloud interface by providing the raw definitions and real time examples within the tech

experiments (i.e. process a big amount of scientific data; although not with big data technolog

Top 10 Words

ope

Create the Dictionary

So the idea here is to first try and open the file, use try/catch so that errors relating to missing files are descriptive and exit gracefully.

Then make an empty dictionary and go through each word in the list:

- If the words is in the list but not the dictionary
  - Put it in the dictionary with a value of 1
- If the word is in the dictionary then increment its value

```
#!/bin/python
from operator import itemgetter

def createDict(filename):
    try:
        filename_fid = open(filename)
        filename_str = filename_fid.read()
        word_list = filename_str.lower().split(' ')
    except:
        print("ERROR: cannot open file")
        return

    matches = dict()

    for word in word_list:
        if word not in matches:
            matches[word] = 1
        else:
            matches[word] = matches[word] + 1

    return matches

createDict('scemunits.txt')
print(createDict('scemunits.txt'))
```

```
{'unit': 2, 'id,': 1, 'name,': 1, 'course': 1, 'name\n301046,': 1, 'big': 1, 'data,': 1, '
```

The rest of the code was given to us and so the sorted values can be returned thusly:

```
from operator import itemgetter
myDict = createDict('bigdata.txt')

sortedList = sorted(myDict.items(), key = itemgetter(1), reverse = True)

for key, value in sortedList[:10]:
    print(key, value)
```

```
the 302
of 213
and 201
data 178
to 170
in 122
a 99
big 80
is 75
as 59
```