**CIS-298 Intro to Python**

**With Professor Robert Mann**

**HW #8**

**Student: Demetrius Johnson**

**23 March 2023**

**Due: 28 March 2023 at 3:30pm**

# #Author: Demetrius Johnson

#Author: Demetrius Johnson

#CIS-298 Intro to Python with Professor Robert Mann

#HW8 - Working with matplotlib

#Remember to copy/paste your code for each question separately.

#Then, follow it with a snippet showing the output.

#Due: Thu March 28, 2023 4pm

# **Description**

#For this project you will be reading a file, counting occurrences of all letters and numbers,

#and displaying the results in a histogram.

# **Requirements**

#Requirements:

#Your input file is the syllabus for this course, found on Canvas under files.

#Use the text version (you may read the docx version if you’d like but it needs a library).

#Read the file, counting the number of times each letter

#(not differentiating between uppercase and lowercase) and number appears (a-z, A-Z, 0-9; 36 total items).

#Display the results as a histogram with appropriate title and labels for x-axis and y-axis.

#Report:

#Submit a report listing your script and a snippet (snipping tool) showing your histogram.

#Make sure your snippet demonstrates all program requirements as having been met.

# Source Code

#Author: Demetrius Johnson

#CIS-298 Intro to Python with Professor Robert Mann

#HW8 - Working with matplotlib

#Remember to copy/paste your code for each question separately.

#Then, follow it with a snippet showing the output.

#Due: Thu March 28, 2023 4pm

#Description

#For this project you will be reading a file, counting occurrences of all letters and numbers,

#and displaying the results in a histogram.

#Requirements:

#Your input file is the syllabus for this course, found on Canvas under files.

#Use the text version (you may read the docx version if you’d like but it needs a library).

#Read the file, counting the number of times each letter

#(not differentiating between uppercase and lowercase) and number appears (a-z, A-Z, 0-9; 36 total items).

#Display the results as a histogram with appropriate title and labels for x-axis and y-axis.

Syllabus\_File = open("Syllabus\_CIS298\_001MannW23.txt")

print(Syllabus\_File.read())

#seek back to beginning of file since I already read all lines and current pos is EOF:

Syllabus\_File.seek(0)

#Now store the contents of the file as a string,

file\_data\_char\_tokens = Syllabus\_File.read()

#First, identify every unique character in the string to create a list of unique characters:

unique\_chars = []

for character in file\_data\_char\_tokens:

if unique\_chars.count(character) == 0:

unique\_chars.append(character)

print(unique\_chars)

#create list of the ASCII decimal equivalent of the unique chars for easier comparisons:

unique\_chars\_ord\_nums = [ord(character) for character in unique\_chars]

print(unique\_chars\_ord\_nums)

#now, we can check criteria by using the unique chars string:

for ord\_character in unique\_chars\_ord\_nums:

if (

not (ord\_character >= 48 and ord\_character <= 57) #characters 0-9

and

not (ord\_character >= 65 and ord\_character <= 90) #characters A-Z

and

not (ord\_character >= 97 and ord\_character <= 122) #characters a-z

):

#if character is not in each of the above ranges,

#then the character is non-alphabetical AND non-numeric, and we need to remove it

file\_data\_char\_tokens = file\_data\_char\_tokens.replace(chr(ord\_character), '')

print(file\_data\_char\_tokens)

#now we need to handle ignore upper and lower case condition (treat lower and upper as the same char);

#use the upper() function to change all lower case char to upper case:

file\_data\_char\_tokens = file\_data\_char\_tokens.upper()

print(file\_data\_char\_tokens)

#Now we create a list, where each element is --> [char, frequency]:

#48-57 decimal = ascii char 0-9

char\_frequencies = [ [chr(ord\_character),0] for ord\_character in range(48,58) ]

#65-90 decimal = ascii char A-Z

char\_frequencies += [ [chr(ord\_character),0] for ord\_character in range(65,91) ]

#now go to each [char, freq] list in char\_frequencies and get num of occurences in our string,

#storing the result in the respective freq in [char, freq] element:

for element\_num, element in enumerate(char\_frequencies):

char\_frequencies[element\_num][1] = file\_data\_char\_tokens.count(element[0])

print(char\_frequencies)

#Now, we have the frequencies of each 0-9, A-Z, and a-z characters where A-Z == a-z characters of the file.

#We can now create a histogram using matplotlib

import matplotlib as mpl

import matplotlib.pyplot as plt

import numpy as np

#change the columns of char\_frequency list into numpy arrays so we can pass it to pyplot histogram function

char\_array=np.array([characters[0] for characters in char\_frequencies])

freq\_array=np.array([freq[1] for freq in char\_frequencies])

print(char\_array)

print(freq\_array)

#now we can plot a bar graph (prof said we can use histogram OR bar gaph;

#in this situtation it makes no sense to create a histogram based on the frequencies of the frequencies, rather

#we only need to plot the category (letter/num) and associated frequency (stored in y):

plt.bar(char\_array, freq\_array)

# Set the x and y axis labels and title

plt.xlabel('Letter/Numbers')

plt.ylabel('Frequency')

plt.title("Syllabus\_CIS298\_001MannW23.txt - letter and number Frequencies")

plt.show()

#print as an array so you can see the letter/number and associated frequency:

print(np.array(char\_frequencies))

#Report:

#Submit a report listing your script and a snippet (snipping tool) showing your histogram.

#Make sure your snippet demonstrates all program requirements as having been met.

# Screenshots:

## Histogram

Chart, bar chart, histogram

Description automatically generated

\*note that z occurs once but the range is so large it appears to not have any occurrences. See the output from running my program which will demonstrate this, here is one screenshot from some of the terminal output as an example:

A picture containing diagram

Description automatically generated

# Special notes:

## How to install matplotlib (and numpy):

* Here is the command to run in order to install matplotlib (and all of its dependencies including numpy):
  + In a windows command line: **py -your\_version\_of\_python -m pip install matplotlib**
  + This will install matplotlib and all of its dependencies (including the correct version) for the environment of your version of Python.
  + Here is the output after running the command, notice it installs any dependencies if they are not present:
  + Text

    Description automatically generated
  + Text

    Description automatically generated