DAV 5400 Module 7 Assignment (30 Points)

Regular Expressions

Text data is often in need of "cleaning" and preparation before it can be effectively used for analysis purposes. Consider the following poorly formatted text string containing information for five concerts held during the month of June:

```
''' JUNE:*****Black Stone Cherry---CAPACITY---:1500 -- $ATTENDANCE: 1,315--GATE:--$28,492
;*****Lady Gaga ----CAPACITY---:25,000--- $ATTENDANCE: 24,368---GATE:--$461,956#;*****Par
amore ----CAPACITY---:3000 ---$ATTENDANCE: 3,000 ---GATE:-$150,000;*****Rage Against the
Machine---CAPACITY---:12000 ---$ATTENDAMCE: 10.782 ---GATE: --$724,087;*****BEYONCE---CAP
ACITY--:20000---$ATTENDANCE: 20,000—-GATE:$2,400,000***** '''
```

Within the text string we are provided with the following information for each of the five concerts:

- **Artist Name**: Prefaced with five asterisks, e.g., *****
- **Capacity of Concert Venue**: Prefaced by the word '**CAPACITY**'
- **Number of concert attendees**: Prefaced by the word "**ATTENDANCE**"
- **Gross Ticket Revenue**: Prefaced by the word "**GATE**"

Use **Python regular expressions** ("regex") along with your knowledge of Python list and dictionary object to complete the following tasks:

1. (**4 Points**) Using regular expressions, extract the **Capacity** and **Attendance** counts for each concert from the unformatted text string shown above and store them in two separate Python list objects, i.e., one list containing the Capacity values and one list containing the Attendance values.

2. (**4 Points**) Using regular expressions, extract the names of each musical artist from the unformatted text string and store them in a Python list object. When complete, your list should contain the following entries:

```
"Black Stone Cherry"   "Lady Gaga"  "Paramore"
"Rage Against the Machine"   "Beyonce"
```

3. (**4 Points**) Using regular expressions, extract the **Gross Ticket Revenue** for each concert from the unformatted text string shown above and store the dollar amounts in a Python list object.

4. Using your newly created list objects, complete the following tasks:

> **a.** (**4 Points**) Using the lists you created for **Questions 1 and 3** above, use your Python skills to create a new dictionary object containing **the average ticket price for each concert** based on the number of concert attendees and the gross ticket revenue. The resulting dictionary object should use the name of each musical artist as **key** values while the average ticket price for their concert is used to populate the associated data values for each **key:value** pair within the dictionary.
>
> **b.** (**4 Points**) Using your regex and/or string processing skills and the list you created for **Question 2** above, construct a new dictionary object indicating whether each musical artist's name is comprised of more than one word. The resulting dictionary should be comprised of one entry for each musical artist, wherein the key value is the musical artist's name and the associated data value contains either the Python keyword '**TRUE**' or the Python keyword '**FALSE**' (relative to whether or not the artist's name is comprised of more than just a single word).

5. (**5 Points**) Consider the character string '**FIdD1E7h=**'. We would like to match this string using the regular expression "**\D[a-zA-Z]*[^,]=**", but the regular expression fails to match the text string. Explain why the regular expression fails and correct it.

6. (**5 Points**) Consider the character string "**The spy was carefully disguised**". We would like to extract only the adverb '**carefully**' from the string. To do so we write the regular expression "**$*\s+ly\w+"**. Explain why this fails and correct the expression.

Be sure to include some commentary **in formatted Markdown cells** explaining your approach to solving each of the individual problems. Save all of your work for this assignment within **a single Jupyter Notebook** submit it via the M7 Assignment page within Canvas. Be sure to save your Notebook using the nomenclature we've been using, i.e., **first initial_last name_M7_assn**" (e.g., J_Smith_M7_assn_).