Introduction to Big Data - Practice 3 (Data Manipulation I)

For each question, make sure you not only just “write down” the Python codes but also “explain the answer with your own language”. All answers without explanation will not be accepted.

Problem

< Question 1 – All Movies data >

Write down Python codes that corresponds to the following questions:

(1-1, 5 pts) Import “movies_dataset_from_allmovie.json” to a pandas dataframe typed variable called *movie_df* and show the first 5 rows.

(1-2, 5 pts) Describe the data dimension of *movie_df*. (Hint: .shape)

(1-3, 5 pts) What are the names of columns used in *movie_df*?

(1-4, 5 pts) How many unique movie genres are included in *movie_df*?

(1-5, 10 pts) What are the list of movies released in year 2004? To answer this question, we have to use datetime64 variable called “released_at”. Write down Python codes that filters samples released in year 2004. (Hint: Use .dt.strftime()).

Expected Result

```
>>> ???
2810                              Beyond the Next Mountain
2912                                       The Big Red One
3002    Billy Graham Classic Message: The Road to Arma...
7071                                 D-Day: Plus 40 Years
9099                           Salute to Jelly Roll Morton
Name: name, dtype: object
```

(1-6, 10 pts) Which genres take the greatest proportion in *movie_df* and its proportion? To figure this out, write down Python codes that generate results below. Do you think this result is sufficient? If you think not, suggest an alternative solution.

Expected Result

```
>>> ???
Drama                                    0.094166
Music                                    0.082003
Comedy                                   0.062429
Sports &amp;amp; Recreation              0.047605
History                                  0.043995
                                             ...
Action, Fantasy, Horror                  0.000095
Action, Horror, Thriller                 0.000095
Action, Drama, Thriller, War             0.000095
Romance, Spy Film, War                   0.000095
Action, Adult, Adventure, Drama, Western 0.000095
Name: genre, Length: 590, dtype: float64
```

(1-7, 10 pts) What is the name of director who has made the most films? To answer this question, write down Python codes that generates results below.

Expected Result

```
>>> ???
                    4551
Sam Newfield          20
Michael Curtiz        16
William Berke         12
Richard Fleischer     12
                     ...
James Bond III         1
Dale Hartleban         1
Paul Donovan           1
Tony Garnett           1
Bill D'Elia            1
Name: director, Length: 3404, dtype: int64
```

(1-8, 10 pts) It seems Sam Newfield is the director who made 20 films. Write down Python codes that generates data.frame that presents the frequency of Sam Newfield's movie genre as shown below.

Expected Result

```
>>> ???
```

	Genre	Count
0	Action, Western	8
1	Western	5
2	Adventure, Drama	1
3	Action, Adventure, Western	1
4	Action, Adventure, Mystery	1
5	Action, Mystery, Western	1
6	Drama, Horror	1
7	Action, Spy Film, Western	1
8	Action, Crime, Drama, Spy Film, Western	1

(1-9, 10 pts) When running a data analysis, not all columns are used, but only the important or highly relevant ones are used. Use *movie_df* to create a new variable called *movie_df_new*, which does not include column 'trailer','url','poster', and 'crawled_at'. Use .drop() method.

Expected Result

```
>>> movie_df_new.head(1)
```

	name	genre	released_at	language	director	domain	duration	synopsis	average_rating	cast	_id	Year
0	10 Days, 10 Years: Nicaraguan Elections of 1990	Culture & Society	1990-01-01			https://www.allmovie.com/	0H54M				72706835-65b7-5559-9d16-73c43b2667bd	1990

(1-10, 10 pts) Using *movie_df_new*, create a new variable called *movie_df_new_rename* with the changed names as shown below.

Expected Result

```
>>> movie_df_new_rename.columns
Index(['MovieTitle', 'Genre', 'ReleaseDate', 'Language', 'director', 'domain',
       'duration', 'synopsis', 'average_rating', 'cast', '_id', 'Year'],
      dtype='object')
```

< Question 2 – Numpy Array >

(2-1, 2.5 pts) Generate numpy matrix as shown below.

Expected Result

```
>>> ???
array([1, 3, 5, 7, 9])
```

(2-2, 2.5 pts) Generate numpy matrix as shown below.

Expected Result

```
>>> ???
array([[ 1,  3,  5],
       [ 7,  9, 11]])
```

(2-3, 2.5 pts) Generate numpy matrix as shown below.

Expected Result

```
>>> ???
array([[[ 1,  3],
        [ 5,  7]],

       [[ 9, 11],
        [13, 15]]])
```

(2-4, 2.5 pts) Generate numpy matrix as shown below.

Expected Result

```
>>> ???
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

(2-5, 10 pts, 2.5 pts each) Given numpy matrix, *mynumpy*, answer the following questions: 1) summation of 0D, 2) average of 1D, 3) maximum value, and 4) standard deviation. (Hint: Adjust the axis parameter for questions 2-5-1 and 2-5-2)

Expected Result

```
>>> mynumpy
array([[[ 5, 10, 15],
        [20, 25, 30],
        [35, 40, 45]],

       [[50, 55, 60],
        [65, 70, 75],
        [80, 85, 90]]])
```