

1. Read and write simple text files

A text file `addition.txt` contains a number in each line. Write a python code that

1.1. Reads all numbers

```
# YOUR CODE HERE
# Hint: Use the .strip() function to remove whitespaces and newlines characters
from a string
file_path = 'addition.txt'
with open(file_path, 'r') as fp:
    lines = fp.readlines()
    for line in lines:
        line = line.strip()
        print(line)
```

```
2
3
5
```

1.2. Calculats their sum

```
# YOUR CODE HERE
file_path = 'addition.txt'
summ = int()
list1 = []
with open(file_path, 'r') as fp:
    lines = fp.readlines()
    for line in lines:
        line = line.strip()
        list1.append(line)
    while '' in list1:
        list1.remove('')
    list1 = list(map(int, list1))
    for line in list1:
        summ = summ + line
print(summ)
```

```
10
```

1.3 Writes the sum at the bottom of the text file

```
# YOUR CODE HERE
# Hint:
# - You might want to open the file in the "Append" mode
# - Add \n characters to start a new line before writing the sum value
with open('addition.txt', 'a') as f:
    f.write('\n')
    f.write(summ)
```

2. Pandas

In this part, you will practice loading and exploring a data set from Netflix.

2.1. Dataset

Read the [Netflix dataset](#)

```
# import
import pandas as pd
```

Solution to the encoding problem [here](#)

```
# YOUR CODE HERE
file_path = 'NetflixOriginals.csv'
df = pd.read_csv(file_path, encoding = 'Windows-1252')
df
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Title	Genre	Premiere	Runtime	IMDB Score	Language
0	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
1	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
2	The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian
3	The Open House	Horror thriller	January 19, 2018	94	3.2	English
4	Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi
...
579	Taylor Swift: Reputation Stadium Tour	Concert Film	December 31, 2018	125	8.4	English
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	October 9, 2015	91	8.4	English/Ukrainian/Russian
581	Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English
582	Emicida: AmarElo - It's All For Yesterday	Documentary	December 8, 2020	89	8.6	Portuguese
583	David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English

584 rows × 6 columns

2.2 Show the first 3 rows

```
# YOUR CODE HERE
df.head(3)
```

```
.dataframe tbody tr th {
  vertical-align: top;
}
```

```
.dataframe thead th {
  text-align: right;
}
```

	Title	Genre	Premiere	Runtime	IMDB Score	Language
0	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
1	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
2	The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian

2.3. Show the last 5 rows

```
# YOUR CODE HERE
df.tail(5)
```

```
.dataframe tbody tr th {
  vertical-align: top;
}

.dataframe thead th {
  text-align: right;
}
```

	Title	Genre	Premiere	Runtime	IMDB Score	Language
579	Taylor Swift: Reputation Stadium Tour	Concert Film	December 31, 2018	125	8.4	English
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	October 9, 2015	91	8.4	English/Ukrainian/Russian
581	Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English
582	Emicida: AmarElo - It's All For Yesterday	Documentary	December 8, 2020	89	8.6	Portuguese

	Title	Genre	Premiere	Runtime	IMDB Score	Language
583	David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English

2.4. Show the title and the genre of the 100th movie in the database?

```
# YOUR CODE HERE
df.loc[99,['Title','Genre']]
```

```
Title    Game Over, Man!
Genre    Action/Comedy
Name: 99, dtype: object
```

2.5. Set the column Title as index for the dataset.

```
# YOUR CODE HERE
df.index
df.set_index('Title',inplace=True)
```

Show the first 5 rows

```
df.head(5)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Genre	Premiere	Runtime	IMDB Score	Language
Title					

	Genre	Premiere	Runtime	IMDB Score	Language
Title					
Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian
The Open House	Horror thriller	January 19, 2018	94	3.2	English
Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi

2.6. How many movies are there in the dataset?

```
# YOUR CODE HERE
information=df.shape
print(information[0])
```

584

2.7. For each movie, what information are available?

```
# YOUR CODE HERE
df.columns
```

```
Index(['Genre', 'Premiere', 'Runtime', 'IMDB Score', 'Language'], dtype='object')
```

2.8. What is the highest IMDB score in the dataset?

```
# YOUR CODE HERE
df['IMDB Score'].max()
```

9.0

2.9. Which movies are scored the highest?

```
# YOUR CODE HERE
df = df.rename(columns=lambda x:x.replace(' ',''))
df[
    df.IMDBScore == 9.0
]
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Genre	Premiere	Runtime	IMDBScore	Language
Title					
David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English

2.10. What is the lowest IMDB score?

```
df['IMDBScore'].min()
```

2.5

2.11. What movies are scored the lowest?

```
df[
    df.IMDBScore == 2.5
]
```

```
.dataframe tbody tr th {
    vertical-align: top;
}
```

```
.dataframe thead th {
    text-align: right;
}
```

	Genre	Premiere	Runtime	IMDBScore	Language
Title					
Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese

2.12. Consider only-English-language movies, find those that are scored above their average score

```
# YOUR CODE HERE
dfinEng=df[
    df.Language == 'English'
]
# print the average IMDB score
avscore = dfinEng['IMDBScore'].mean()
print('the average IMDB score of English movies is',avscore)
# show all those score higher than the average
dfinEng[
    dfinEng.IMDBScore > avscore
]
```

the average IMDB score of English movies is 6.380049875311723

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Genre	Premiere	Runtime	IMDBScore	Language
Title					
Ariana Grande: Excuse Me, I Love You	Concert Film	December 21, 2020	97	6.4	English
ARQ	Science fiction/Thriller	September 16, 2016	88	6.4	English

	Genre	Premiere	Runtime	IMDBScore	Language
Title					
First Match	Sports-drama	March 30, 2018	102	6.4	English
Fractured	Thriller	October 11, 2019	100	6.4	English
Irreplaceable You	Drama	February 16, 2018	96	6.4	English
...
Dancing with the Birds	Documentary	October 23, 2019	51	8.3	English
Ben Platt: Live from Radio City Music Hall	Concert Film	May 20, 2020	85	8.4	English
Taylor Swift: Reputation Stadium Tour	Concert Film	December 31, 2018	125	8.4	English
Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English
David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English

210 rows × 5 columns

2.13. (Optional) Is there a correlation between runtime and IMDB score?

```

cornum = df.corr()
cornum = cornum.iloc[0,1]
if cornum>-0.05:
    if cornum<0.05:
        print('There is no correlation between runtime and IMDB score')
    else:
        print('There is a correlation between runtime and IMDB score')
else:
    print('There is a correlation between runtime and IMDB score')

```

There is no correlation between runtime and IMDB score

2.14 How many languages available?

```
# YOUR CODE HERE
df['Language'].unique()
```

```
array(['English/Japanese', 'Spanish', 'Italian', 'English', 'Hindi',
      'Turkish', 'Korean', 'Indonesian', 'Malay', 'Dutch', 'French',
      'English/Spanish', 'Portuguese', 'Filipino', 'German', 'Polish',
      'Norwegian', 'Marathi', 'Thai', 'Swedish', 'Japanese',
      'Spanish/Basque', 'Spanish/Catalan', 'English/Swedish',
      'English/Taiwanese/Mandarin', 'Thia/English', 'English/Mandarin',
      'Georgian', 'Bengali', 'Khmer/English/French', 'English/Hindi',
      'Tamil', 'Spanish/English', 'English/Korean', 'English/Arabic',
      'English/Russian', 'English/Akan', 'English/Ukranian/Russian'],
      dtype=object)
```

2.15. Extract all action movies. Save them into a csv file 'action_movies.csv'

```
# YOUR CODE HERE
df['isAction'] = (df['Genre'].str.contains('Action'))
action_movies=df[
    df.isAction == True
]
action_movies.to_csv('action_movies.csv')
```

2.16. For each genre of movies, answer the following question

- How many movies are available in the dataset?

```
# YOUR CODE HERE
df_genre=df.groupby('Genre')
df_genre_number=df_genre.count()
df_genre_number['Premiere']
```

Genre	
Action	7
Action comedy	5
Action thriller	1
Action-adventure	1
Action-thriller	3
	..
War	2

```

War drama          2
War-Comedy          1
Western             3
Zombie/Heist        1
Name: Premiere, Length: 115, dtype: int64

```

- What are their average score?

```

# YOUR CODE HERE
df_genre_avscore=df_genre['IMDBScore'].mean()
df_genre_avscore

```

```

Genre
Action          5.414286
Action comedy    5.420000
Action thriller  6.400000
Action-adventure 7.300000
Action-thriller  6.133333
...
War              6.750000
War drama        7.100000
War-Comedy        6.000000
Western          6.066667
Zombie/Heist      5.900000
Name: IMDBScore, Length: 115, dtype: float64

```

2.17. On average, what type of movies scores the highest?

```

df_genre_avscore = pd.DataFrame(df_genre_avscore)
df_genre_avscore[
    df_genre_avscore.IMDBScore == df_genre_avscore.IMDBScore.max()]

```

```

.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}

```

	IMDBScore
--	-----------

Genre	IMDBScore
Genre	
Animation/Christmas/Comedy/Adventure	8.2