STC Jawwy

```
"""
Here we install libraries that are not installed by default
Example:  pyslsb
Feel free to add any library you are planning to use.
"""
!pip install pyxlsb
```

```
     Collecting pyxlsb
       Downloading pyxlsb-1.0.10-py2.py3-none-any.whl (23 kB)
     Installing collected packages: pyxlsb
     Successfully installed pyxlsb-1.0.10
```

```
# Import the required libraries
"""
Please feel free to import any required libraries as per your needs
"""
import pandas as pd     # provides high-performance, easy to use structures and data analysis tools
import pyxlsb           # Excel extention to read xlsb files (the input file)
import numpy as np      # provides fast mathematical computation on arrays and matrices
```

Jawwy dataset

The dataset consists of meta details about the movies and tv shows as genre. Also details about Users activities, spent duration and if watching in High definition or standard definition. You have to analyse this dataset to find top insights, findings and to solve the four tasks assigned to you.

```
dataframe = pd.read_excel("/content/stc TV Data Set_T1.xlsb",sheet_name="Final_Dataset")
df =dataframe.copy
# Please make a copy of dataset if you are going to work directly and make changes on the dataset
# you can use   df=dataframe.copy()
```

```
# check the data shape
dataframe.shape
```

```
     (1048575, 13)
```

```
# display the first 5 rows
dataframe.head()
```

	Column1	date_	user_id_maped	program_name	duration_seconds	program_class	season	episode	progra
0	1	42882	26138	100 treets	40	MOVIE	0	0	M
1	3	42876	7946	Moana	17	MOVIE	0	0	Ar Movi
2	4	42957	7418	The Mermaid Princess	8	MOVIE	0	0	Ar M I Prince
3	5	42942	19307	The Mermaid Princess	76	MOVIE	0	0	Ar M I Prince
4	7	42923	15860	Churchill	87	MOVIE	0	0	Bi Movie(

```
# Data Preprocessing on the input data
dataframe = dataframe.drop(columns=['Column1'])         # dropping the index column
dataframe['program_name'] = dataframe['program_name'].str.strip()  # trim spaces in movies names to avoid m
dataframe['date_'] = pd.to_datetime(dataframe['date_'], unit='d', origin='30/12/1899')  # read date column
dataframe[['duration_seconds', 'season','episode','series_title','hd']] = dataframe[['duration_seconds', 's
dataframe[['user_id_maped', 'program_name','program_class','program_desc','program_genre','original_name']]
```

```
/Users/mahmoud/opt/anaconda3/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3361: UserWar
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
# display the dataset after applying data types
dataframe.head()
```

	date_	user_id_maped	program_name	duration_seconds	program_class	season	episode	program_desc
0	2017-05-27	26138	100 treets	40	MOVIE	0	0	Drama Movie100 Streets
1	2017-05-21	7946	Moana	17	MOVIE	0	0	Animation MovieMoana (HD)
2	2017-08-10	7418	The Mermaid Princess	8	MOVIE	0	0	Animation MovieThe Mermaid Princess (HD)
3	2017-07-26	19307	The Mermaid Princess	76	MOVIE	0	0	Animation MovieThe Mermaid Princess (HD)
4	2017-07-07	15860	Churchill	87	MOVIE	0	0	Biography MovieChurchill (HD)

```
# describe the numeric values in the dataset
dataframe.describe()
```

	duration_seconds	season	episode	series_title	hd
count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06
mean	1.230957e+03	1.342139e+00	6.157952e+00	1.205922e-02	3.862728e-01
std	6.821058e+03	2.104095e+00	1.222015e+01	1.091504e-01	4.868946e-01
min	2.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	5.200000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	1.190000e+02	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
75%	1.328000e+03	1.000000e+00	9.000000e+00	0.000000e+00	1.000000e+00
max	1.461329e+06	2.300000e+01	2.820000e+02	1.000000e+00	1.000000e+00

```
# check if any column has null value in the dataset
dataframe.isnull().any()
```

```
date_               False
user_id_maped       False
program_name        False
duration_seconds    False
program_class       False
season              False
episode             False
program_desc        False
program_genre       False
series_title        False
hd                  False
original_name       False
dtype: bool
```

Task 1

You are required to work on task one to study and HD flag for available dataset

```
# make a copy of the dataframe for working on task 1
df=dataframe.copy()
```

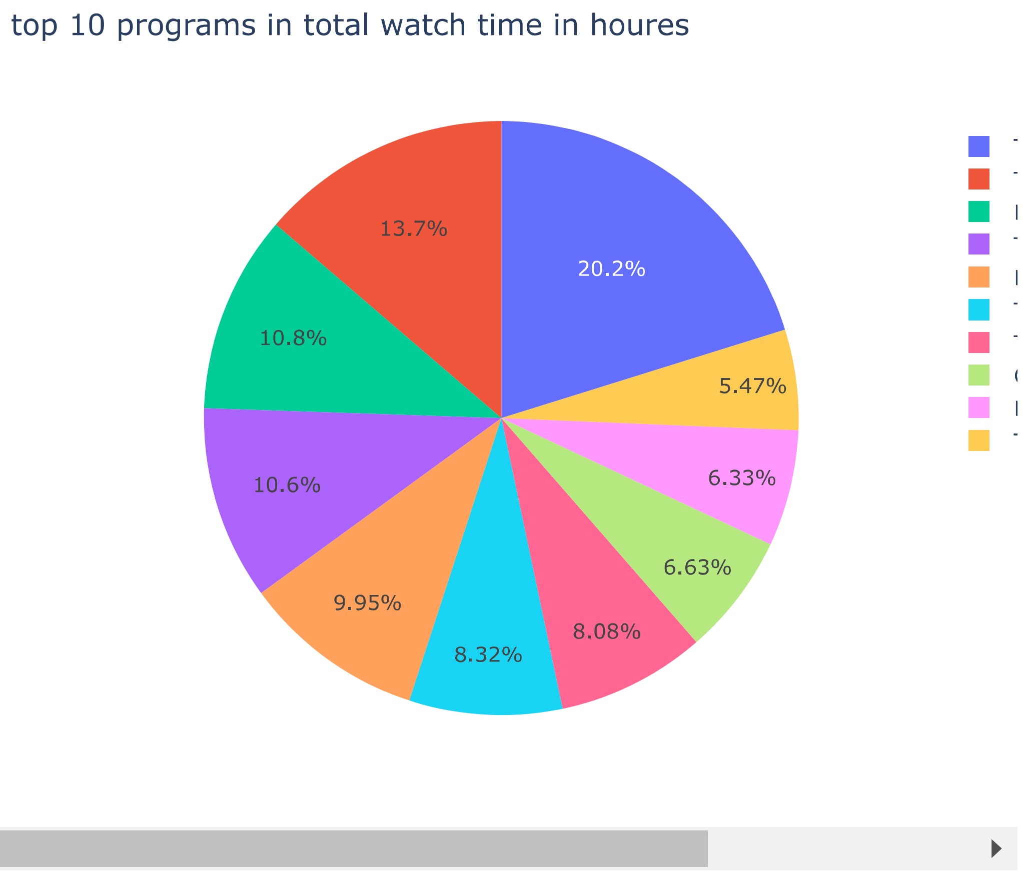
```
# Here we try to get the most watched movies (Total Views / Total Users Views / Total watch time)
# For series we concatenated the Session episode to differentiate between episodes
grouped=df.copy()
grouped.loc[grouped['program_class'] == 'SERIES/EPISODES', 'program_name'] = grouped['program_name']+'_SE'+
grouped = grouped.groupby(['program_name','program_class'])\
.agg({'user_id_maped': [('co1', 'nunique'),('co2', 'count')],\
      'duration_seconds': [('co3', 'sum')] }).reset_index()
grouped.columns = ['program_name','program_class','No of Users who Watched', 'No of watches', 'Total watch
grouped['Total watch time in houres']=grouped['Total watch time in seconds']/3600
grouped = grouped.drop(columns=['Total watch time in seconds'])
grouped = grouped.sort_values(by=['Total watch time in houres', 'No of watches','No of Users who Watched'],
```

```
# show the result
grouped.head(35)
```

	program_name	program_class	No of Users who Watched	No of watches	Total watch time in houres
0	The Boss Baby	MOVIE	3389	24047	2961.350833
1	The Amazing pider-Man	MOVIE	1011	2877	1966.119167
2	The Expendables	MOVIE	853	2119	1961.159444
3	Moana	MOVIE	2173	8081	1706.176944
4	Trolls	MOVIE	2613	13793	1601.023056
5	Bean	MOVIE	949	3617	1423.955000
6	The murfs	MOVIE	867	3132	1342.141111
7	Hotel Transylvania	MOVIE	491	1947	1096.533611
8	Cloudy With a Chance of Meatballs	MOVIE	683	2076	948.674722
9	The Man With The Iron Fists	MOVIE	707	2505	859.626389
10	Salt	MOVIE	563	1082	767.392778
11	Unbroken	MOVIE	625	1429	763.078333
12	ParaNorman	MOVIE	614	1746	747.065556
13	Youm Maloosh Lazma	MOVIE	1131	2278	718.109722
14	Ferdinand	MOVIE	1278	6817	714.223056
15	White Chicks	MOVIE	307	916	711.840833
16	Jurassic Park	MOVIE	504	1192	693.394444
17	The November Man	MOVIE	494	1219	679.492222
18	Total Recall	MOVIE	587	1108	661.820000
19	Robin Hood	MOVIE	588	1209	643.935000
20	Public Enemies	MOVIE	368	716	634.035000
21	Daddy Day Camp	MOVIE	263	647	625.338333
22	Oblivion	MOVIE	790	1678	609.391111
23	Blitz	MOVIE	562	1200	570.521944
24	War for the Planet of the Apes	MOVIE	879	2028	567.597778
25	Inside Man	MOVIE	532	1567	560.386111
26	Bad Boys	MOVIE	438	871	559.277500

```
# we import Visualization libraries
# you can ignore and use any other graphing libraries
import matplotlib.pyplot as plt # a comprehensive library for creating static, animated, and interactive vi
import plotly #a graphing library makes interactive, publication-quality graphs. Examples of how to make li
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

```
# plot top 10 Programs
fig = px.pie(grouped.head(10), values='Total watch time in houres', names='program_name',\
             hover_data=['program_class'],title='top 10 programs in total watch time in houres')
fig.show()
```

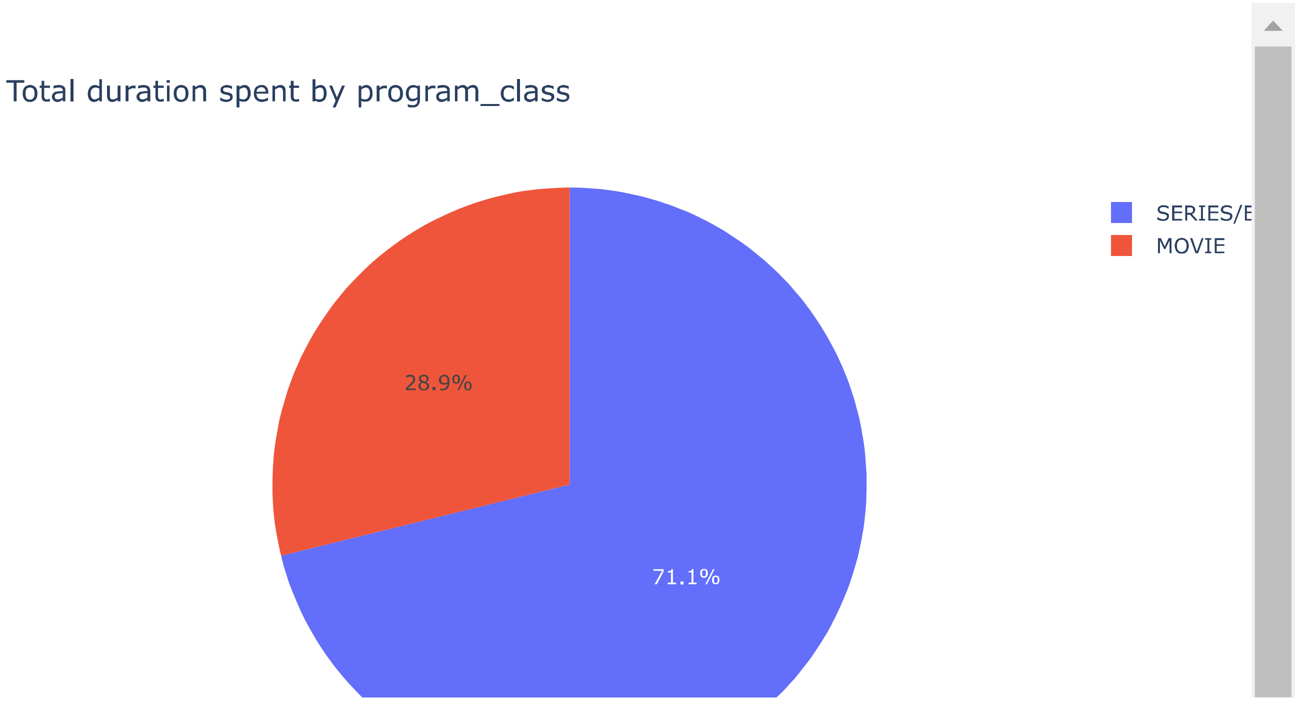


```
# Here we try to study the customer experience against Program class
grouped=df.copy()
grouped = grouped.groupby('program_class')\
.agg({'user_id_maped': [('co1', 'nunique'),('co2', 'count')],\
      'duration_seconds': [('co3', 'sum')] }).reset_index()
grouped.columns = ['program_class','No of Users who Watched', 'No of watches', 'Total watch time in seconds
grouped['Total watch time in houres']=grouped['Total watch time in seconds']/3600
grouped = grouped.drop(columns=['Total watch time in seconds'])
grouped = grouped.sort_values(by=['Total watch time in houres', 'No of watches','No of Users who Watched'],
# show the result
grouped.head()
```

	program_class	No of Users who Watched	No of watches	Total watch time in houres
0	SERIES/EPISODES	3901	560174	255097.787500
1	MOVIE	[illegible]	[illegible]	[illegible]

```
# plot the total watch time against total number of users and report your findings
fig = px.pie(grouped, values='Total watch time in houres', names='program_class',\
             hover_data=['program_class'],title='Total duration spent by program_class')
fig2 = px.pie(grouped, values='No of Users who Watched', names='program_class',\
             hover_data=['program_class'],title='Total Users watching by program_class')

fig.update_traces(sort=False)
fig2.update_traces(sort=False)
fig.show()
fig2.show()
```


Total duration spent by program_class
SERIES/E
MOVIE