In [1]:
```
!pip install pyxlsb
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
```

```
Requirement already satisfied: pyxlsb in c:\users\lenovo\anaconda\lib\site-packages (1.0.10)
```

In [2]:
```
dataframe = pd.read_csv('stc.csv')
```

In [3]:
```
dataframe.head()
```

Out[3]:

	Column1	date_	user_id_maped	program_name	duration_seconds	program_class	season	episode	program_desc	program_genre	series_title	hd
0	1	5/27/2017	26138	100 treets	40	MOVIE	0	0	Drama Movie100 Streets	Drama	0	0
1	3	5/21/2017	7946	Moana	17	MOVIE	0	0	Animation MovieMoana (HD)	Animation	0	1
2	4	8/10/2017	7418	The Mermaid Princess	8	MOVIE	0	0	Animation MovieThe Mermaid Princess (HD)	Animation	0	1
3	5	7/26/2017	19307	The Mermaid Princess	76	MOVIE	0	0	Animation MovieThe Mermaid Princess (HD)	Animation	0	1
4	7	7/7/2017	15860	Churchill	87	MOVIE	0	0	Biography MovieChurchill (HD)	Biography	0	1

In [4]:
```
# Data Preprocessing on the input data
dataframe = dataframe.drop(columns=['Column1'])
```

In [5]:
```
dataframe.head()
```

Out[5]:

	date_	user_id_maped	program_name	duration_seconds	program_class	season	episode	program_desc	program_genre	series_title	hd	original_n
0	5/27/2017	26138	100 treets	40	MOVIE	0	0	Drama Movie100 Streets	Drama	0	0	100 t
1	5/21/2017	7946	Moana	17	MOVIE	0	0	Animation MovieMoana (HD)	Animation	0	1	M
2	8/10/2017	7418	The Mermaid Princess	8	MOVIE	0	0	Animation MovieThe Mermaid Princess (HD)	Animation	0	1	The Mer Prin
3	7/26/2017	19307	The Mermaid Princess	76	MOVIE	0	0	Animation MovieThe Mermaid Princess (HD)	Animation	0	1	The Mer Prin
4	7/7/2017	15860	Churchill	87	MOVIE	0	0	Biography MovieChurchill (HD)	Biography	0	1	Chu

In [6]:
```
dataframe['program_name'] = dataframe['program_name'].str.strip()  # trim spaces in movies names to avoid misspellings in inp
```

In [7]:
```
dataframe['date_'] = pd.to_datetime(dataframe['date_'])  # read date column as date data type
dataframe[['duration_seconds', 'season','episode','series_title','hd']] = dataframe[['duration_seconds', 'season','episode','
```

In [8]:
```
dataframe[['user_id_maped', 'program_name','program_class','program_desc','program_genre','original_name']] = dataframe[['use
```

In [9]:
```
dataframe.head()
```

Out[9]:

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

In [10]:
```
dataframe.describe()
```

Out[10]:

	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

In [11]:
```
dataframe.isnull().any()
```

In [11]:
```
dataframe.isnull().any()
```

Out[11]:
```
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
```

In [12]:
```
df=dataframe.copy()
```

In [13]:
```
grouped=df.copy()
```

In [14]:
```
grouped.loc[grouped['program_class'] == 'SERIES/EPISODES', 'program_name'] = grouped['program_name']+'_SE'+grouped['season'].
```

In [15]:
```
grouped = grouped.groupby(['program_name','program_class'])\
.agg({'user_id_maped': [('co1', 'nunique'),('co2', 'count')],\
      'duration_seconds': [('co3', 'sum')] }).reset_index()
```

In [16]:
```
grouped.head()
```

Out[16]:

	program_name	program_class	user_id_maped		duration_seconds
			co1	co2	co3
0	#FollowFriday	MOVIE	316	510	191769
1	10 Days in a Madhouse	MOVIE	301	553	356806
2	100 treets	MOVIE	704	1851	319000
3	101 Dalmatians	MOVIE	85	118	30002
4	102 Dalmatians	MOVIE	130	172	32235

In [17]:
```
grouped.columns = ['program_name','program_class','No of Users who Watched', 'No of watches', 'Total watch time in seconds']
```

In [18]:
```
grouped.head()
```

Out[18]:

	program_name	program_class	No of Users who Watched	No of watches	Total watch time in seconds
0	#FollowFriday	MOVIE	316	510	191769
1	10 Days in a Madhouse	MOVIE	301	553	356806
2	100 treets	MOVIE	704	1851	319000
3	101 Dalmatians	MOVIE	85	118	30002
4	102 Dalmatians	MOVIE	130	172	32235

In [19]:
```
grouped['Total watch time in hours']=grouped['Total watch time in seconds']/3600
```

In [20]:
```
grouped.head()
```

Out[20]:

	program_name	program_class	No of Users who Watched	No of watches	Total watch time in seconds	Total watch time in hours
0	#FollowFriday	MOVIE	316	510	191769	53.269167
1	10 Days in a Madhouse	MOVIE	301	553	356806	99.112778
2	100 treets	MOVIE	704	1851	319000	88.611111
3	101 Dalmatians	MOVIE	85	118	30002	8.333889
4	102 Dalmatians	MOVIE	130	172	32235	8.954167

In [21]:
```
grouped = grouped.drop(columns=['Total watch time in seconds'])
```

In [23]:
```
grouped = grouped.sort_values(by=['Total watch time in hours', 'No of watches','No of Users who Watched'], ascending=False).r
```

In [23]:
```
grouped = grouped.sort_values(by=['Total watch time in hours', 'No of watches','No of Users who Watched'], ascending=False).r
```

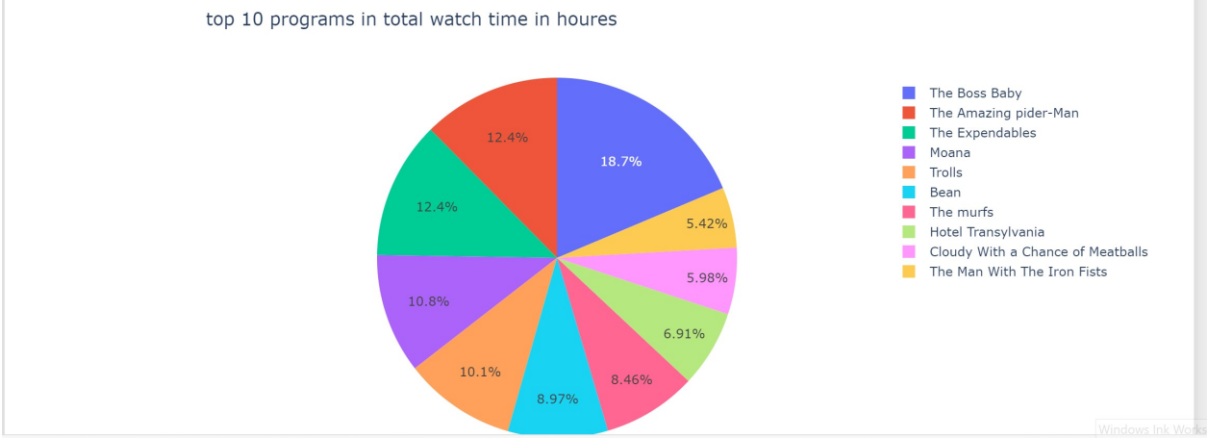
In [24]:
```
grouped.head(50)
```

Out[24]:

	program_name	program_class	No of Users who Watched	No of watches	Total watch time in hours
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

In [30]:
```
import matplotlib.pyplot as plt # a comprehensive library for creating static, animated, and interactive visualizati
import plotly #a graphing library makes interactive, publication-quality graphs. Examples of how to make line plots,
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [33]:
```
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()
```



In [36]:

```
fig2 = px.pie(grouped, values='No of Users who Watched', names='program_class',\
              hover_data=['program_class'],title='Total duration spent by program_class')

fig2.show()
```