

```
!wget "https://nm-public-data.s3.us-east-2.amazonaws.com/dataset/all_traffic_time_10.pkl"
```

```
--2020-12-08 01:53:56-- https://nm-public-data.s3.us-east-2.amazonaws.com/dataset/all_1
Resolving nm-public-data.s3.us-east-2.amazonaws.com (nm-public-data.s3.us-east-2.amazona
Connecting to nm-public-data.s3.us-east-2.amazonaws.com (nm-public-data.s3.us-east-2.am
HTTP request sent, awaiting response... 200 OK
Length: 1551386264 (1.4G) [binary/octet-stream]
Saving to: 'all_traffic_time_10.pkl'

all_traffic_time_10 100%[=====>] 1.44G 46.4MB/s in 33s

2020-12-08 01:54:29 (45.4 MB/s) - 'all_traffic_time_10.pkl' saved [1551386264/1551386264]
```



Download the video collection dataset - 2019. Labeled video sessions dataset. https://nm-public-data.s3.us-east-2.amazonaws.com/dataset/all_traffic_time_10.pkl.

```
import pickle as pickle
import pandas as pd
import seaborn as sns
import pprint
```

```
import pickle
import numpy
```

```
with open('/content/all_traffic_time_10.pkl', 'rb') as f:
    u = pickle.Unpickler(f)
    u.encoding = 'latin1'
    p = u.load()
    print(p.head(5))
    pprint.pprint(p.head(5))
```

```

                                10_EWMA_chunksizes ... startup_mc
0  [[12816.0], [211635.2], [60158.47619047619], [... ...      6.0
1  [[198244.0], [486736.8], [457148.38095238095],... ...      6.0
2  [[12786.0], [12811.6], [406599.33333333333], [2... ...      6.0
3  [[772020.0], [642372.0], [536471.8095238095], ... ...      6.0
4  [[503378.0], [572290.0], [287298.0], [474179.5... ...      6.0
```

```
[5 rows x 251 columns]
```

```

                                10_EWMA_chunksizes ... startup_mc
0  [[12816.0], [211635.2], [60158.47619047619], [... ...      6.0
1  [[198244.0], [486736.8], [457148.38095238095],... ...      6.0
2  [[12786.0], [12811.6], [406599.33333333333], [2... ...      6.0
3  [[772020.0], [642372.0], [536471.8095238095], ... ...      6.0
4  [[503378.0], [572290.0], [287298.0], [474179.5... ...      6.0
```

[5 rows x 251 columns]

```
pd.set_option('display.max_rows', None)
```

```
p.columns.tolist()
```

```

    userEndBytesInFlight',
    'userFinFlags',
    'userGoodput',
    'userIdleTime',
    'userKurBytesInFlight',
    'userKurBytesPerPacket',
    'userKurInterArrivalTime',
    'userKurRTT',
    'userKurRetransmit',
    'userKurRwnd',
    'userMaxBytesInFlight',
    'userMaxBytesPerPacket',
    'userMaxInterArrivalTime',
    'userMaxRTT',
    'userMaxRetransmit',
    'userMaxRwnd',
    'userMedBytesInFlight',
    'userMedBytesPerPacket',
    'userMedInterArrivalTime',
    'userMedRTT',
    'userMedRetransmit',
    'userMedRwnd',
    'userMinBytesInFlight',
    'userMinBytesPerPacket',
    'userMinInterArrivalTime',
    'userMinRTT',
    'userMinRetransmit',

    'userMinRwnd',
    'userOneRetransmit',
    'userOutOfOrderBytes',
    'userOutOfOrderPackets',
    'userPacketCount',
    'userPshFlags',
    'userRstFlags',
    'userSkeBytesInFlight',
    'userSkeBytesPerPacket',
    'userSkeInterArrivalTime',
    'userSkeRTT',
    'userSkeRetransmit',
    'userSkeRwnd',
    'userStdBytesInFlight',
    'userStdBytesPerPacket',
    'userStdInterArrivalTime',
    'userStdRTT',
    'userStdRetransmit',
    'userStdRwnd',
    'userStrBytesInFlight',
    'userSynFlags',
    ,

```

```
'userThroughput',
'userTwoRetransmit',
'userUrgFlags',
'userXRetransmit',
'userZeroRetransmit',
'service',
'startup3.3',
'startup6.6',
'startup5',
'startup10',
'startup_mc']
```

```
pd.options.display.max_columns = None
```

```
p.head()
```

	10_EWMA_chunksizes	10_avg_chunksize	10_chunksizes_50	10_chunksizes_50R	10_chu
0	[[12816.0], [211635.2], [60158.47619047619], [...	195061.0	105533.0	105533.0	
1	[[198244.0], [486736.8], [457148.38095238095],...	349524.2	323073.0	323073.0	
2	[[12786.0], [12811.6], [406599.3333333333], [2...	552724.6	596820.0	596820.0	
3	[[772020.0], [642372.0], [536471.8095238095], ...	570248.0	586598.0	586598.0	
4	[[503378.0], [572290.0], [287298.0], [474179.5...	659017.2	586598.0	586598.0	

```
p.columns
```

```
Index(['10_EWMA_chunksizes', '10_avg_chunksize', '10_chunksizes_50',
      '10_chunksizes_50R', '10_chunksizes_75', '10_chunksizes_75R',
      '10_chunksizes_85', '10_chunksizes_85R', '10_chunksizes_90',
      '10_chunksizes_90R',
      ...,
      'userTwoRetransmit', 'userUrgFlags', 'userXRetransmit',
      'userZeroRetransmit', 'service', 'startup3.3', 'startup6.6', 'startup5',
      'startup10', 'startup_mc'],
      dtype='object', length=251)
```

Create subset of the dataframe to include columns of interest

```
columns=['absolute_timestamp',
'avg_flow_age',
'bitrate',
'bitrate_change',
...]
```

```

'c_bitrate_switches',
'c_rebufferings',
'c_resolution_switches',
'n_bitrate_switches',
'n_chunks_down',
'n_chunks_up',
'n_rebufferings',
'parallel_flows',
'quality',
'relative_timestamp',
'resolution',
'service_Video_throughput_down',
'service_Video_throughput_up',
'service_non_video_throughput_down',
'service_non_video_throughput_up',
'total_throughput_down',
'total_throughput_up',
'video_duration',
'userGoodput',
'userMaxRTT',
'userPacketCount',
'service',
'startup3.3',
'startup6.6',
'startup5',
'startup10',
'startup_mc',
'video_id',
'session_id' ]

```

```
dataframe = p[columns]
```

```
dataframe.head()
```

	absolute_timestamp	avg_flow_age	bitrate	bitrate_change	c_bitrate_switches	c_reb
0	1.549056e+09	-1.549056e+12	1496.0	0.0	1	
1	1.549056e+09	-1.549056e+12	1496.0	0.0	1	
2	1.549056e+09	-1.549056e+12	1496.0	0.0	0	
3	1.549057e+09	-1.549056e+12	2246.0	750.0	1	
4	1.549057e+09	-1.549056e+12	2246.0	0.0	0	

Create unique id for each video : video_id + session_id

```
vid=dataframe.video_id+"_" + dataframe.session_id
```

```
dataframe['vid']=vid
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
"""Entry point for launching an IPython kernel.
```



```
dataframe['vid'].head()
```

```
0    80190088_a129535f86da0ead31a26775ceb81431
1    80190088_a129535f86da0ead31a26775ceb81431
2    80190088_a129535f86da0ead31a26775ceb81431
3    80190088_a129535f86da0ead31a26775ceb81431
4    80190088_a129535f86da0ead31a26775ceb81431
Name: vid, dtype: object
```

```
len(dataframe.vid.unique())
```

```
13765
```

13765 videos are present in the dataset comprising services like netflix, youtube, amazon

```
dataframe.video_id.head()
```

```
0    80190088
1    80190088
2    80190088
3    80190088
4    80190088
Name: video_id, dtype: object
```

Separate vieos by video streaming service provider

```
netflix = dataframe[dataframe.service=='netflix']
youtube = dataframe[dataframe.service=='youtube']
amazon = dataframe[dataframe.service=='amazon']
```

Get aggregate statictics for each service

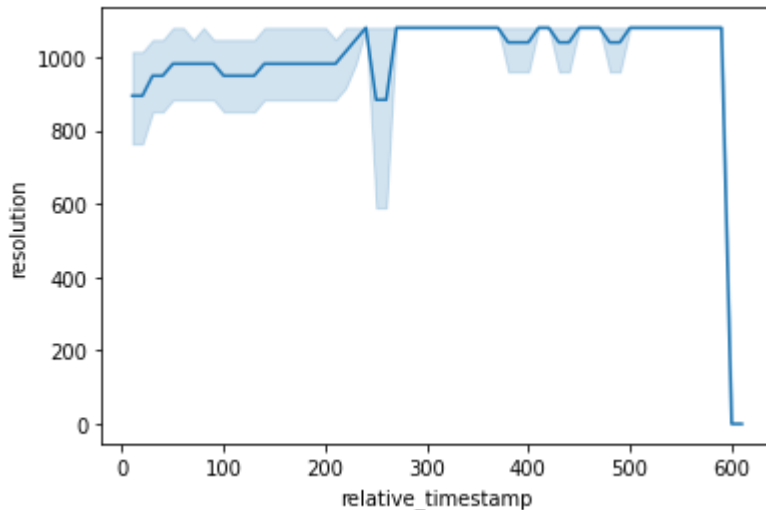
```
netflix_videos = netflix.groupby('vid').mean()
youtube_videos = youtube.groupby('vid').mean()
amazon_videos = amazon.groupby('vid').mean()
```

Observe trends in video resolution, throughput, bit rate changes for a single video

```
video = dataframe[dataframe.video_id=="80190088"]
```

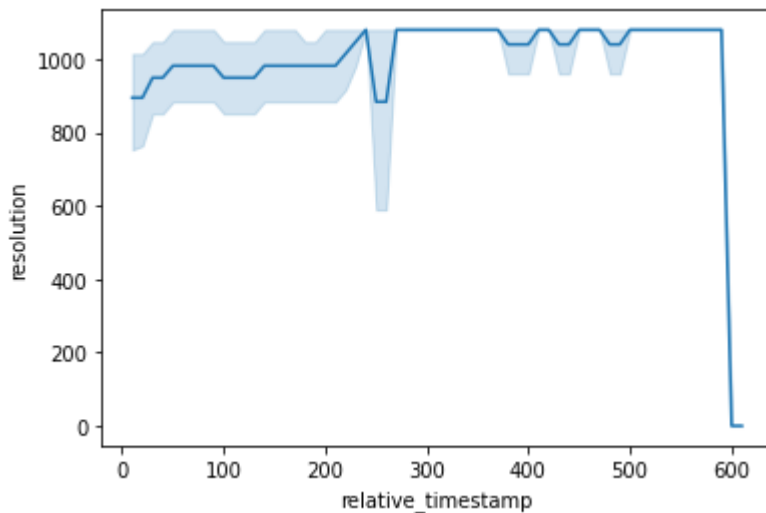
```
import seaborn as sns
sns.lineplot(video.relative_timestamp, video.resolution)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword arguments: {'x': 'relative_timestamp', 'y': 'resolution'} (or name all variables individually) (Deprecated since v0.11)
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f79563b0208>
```



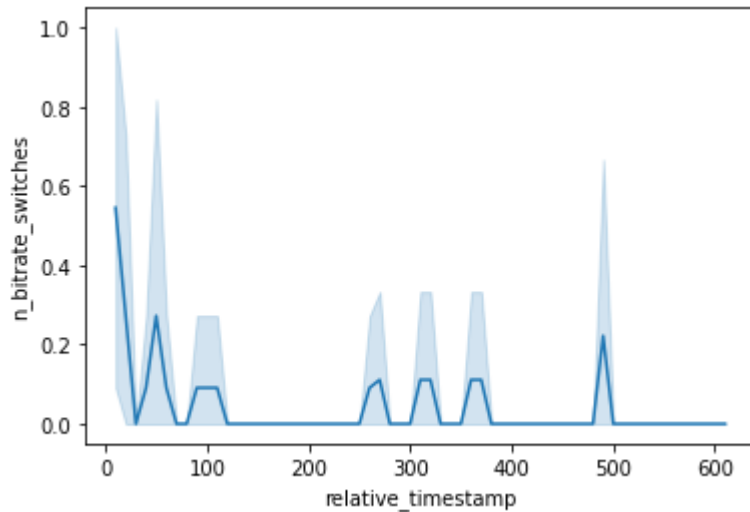
```
sns.lineplot(video.relative_timestamp, video.resolution)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword arguments: {'x': 'relative_timestamp', 'y': 'resolution'} (or name all variables individually) (Deprecated since v0.11)
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f795608a780>
```



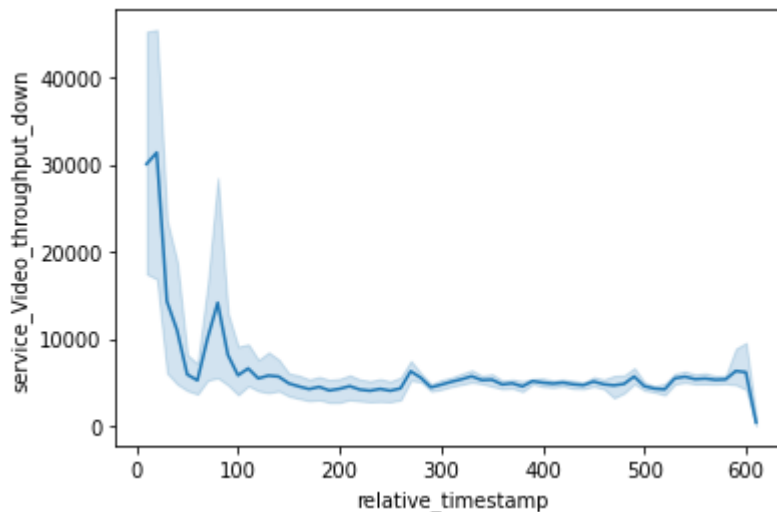
```
sns.lineplot(video.relative_timestamp, video.n_bitrate_switches)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f795321beb8>
```



```
sns.lineplot(video.relative_timestamp, video.service_Video_throughput_down)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f7952650c50>
```

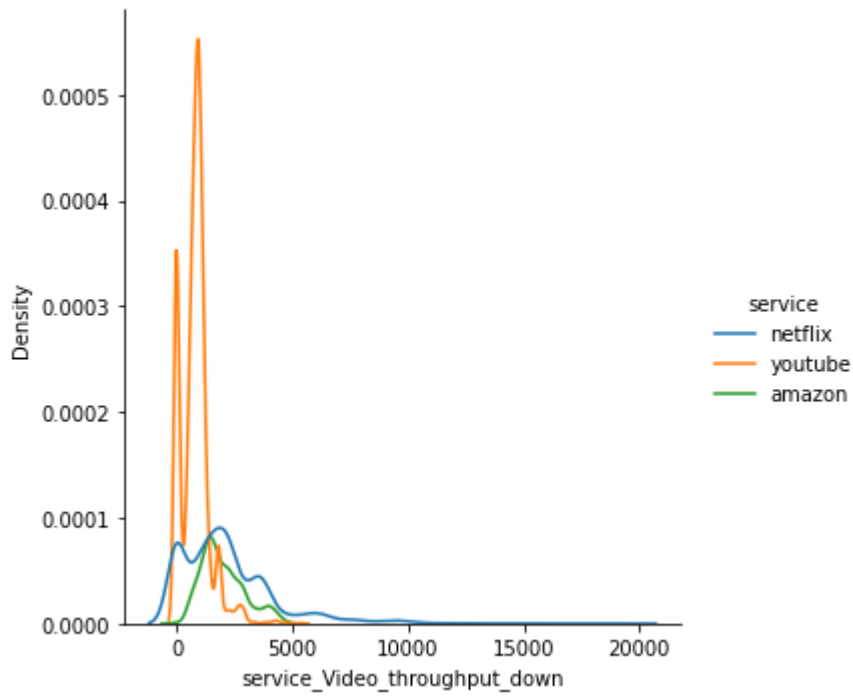


```
netflix_videos['service']='netflix'
amazon_videos['service']='amazon'
youtube_videos['service']='youtube'
```

```
dataset_combined = pd.concat([netflix_videos,youtube_videos,amazon_videos])
```

Visualize density plots of average throughput and average resolution for each service

```
sns.displot(dataset_combined,x="service Video throughput down",kind="kde", hue='service')
```



```
sns.displot(dataset_combined,x="resolution",kind="kde", hue='service')
```

<seaborn.axisgrid.FacetGrid at 0x7f7953f3f6d8>

