In [1]:
```
import pandas as pd
import numpy as np
from calendar import monthrange
```

In [2]:
```
twitch_df = pd.read_csv('Twitch_game_data.csv', encoding='cp1252')
```

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

Out[3]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_chann
0	1	League of Legends	1	2016	94377226	1362044	530270	29
1	2	Counter-Strike: Global Offensive	1	2016	47832863	830105	372654	2
2	3	Dota 2	1	2016	45185893	433397	315083	1
3	4	Hearthstone	1	2016	39936159	235903	131357	5
4	5	Call of Duty: Black Ops III	1	2016	16153057	1151578	71639	36

In [4]:
```
twitch_df['Month'].describe()
```

Out[4]:
```
count    17400.000000
mean         6.344828
std          3.493370
min          1.000000
25%          3.000000
50%          6.000000
75%          9.000000
max         12.000000
Name: Month, dtype: float64
```

In [5]:
```
twitch_df_global = pd.read_csv('Twitch_global_data.csv', encoding='cp1252')
```

In [6]:
```
twitch_df_global.head()
```

Out[6]:

	year	Month	Hours_watched	Avg_viewers	Peak_viewers	Streams	Avg_channels	Games_st
0	2016	1	480241904	646355	1275257	7701675	20076	
1	2016	2	441859897	635769	1308032	7038520	20427	
2	2016	3	490669308	660389	1591551	7390957	20271	
3	2016	4	377975447	525696	1775120	6869719	16791	
4	2016	5	449836631	605432	1438962	7535519	19394	

In [7]:
```
twitch_df.describe()
```

Out[7]:

	Rank	Month	Year	Hours_watched	Hours_streamed	Peak_viewers
count	17400.000000	17400.000000	17400.000000	1.740000e+04	1.740000e+04	1.740000e+04
mean	100.500000	6.344828	2019.137931	5.406489e+06	1.740947e+05	6.129193e+04
std	57.735964	3.493370	2.096430	1.847645e+07	5.546933e+05	1.479578e+05
min	1.000000	1.000000	2016.000000	8.981100e+04	1.900000e+01	4.410000e+02
25%	50.750000	3.000000	2017.000000	4.329688e+05	1.414950e+04	9.270000e+03
50%	100.500000	6.000000	2019.000000	9.594830e+05	3.669700e+04	2.228400e+04
75%	150.250000	9.000000	2021.000000	2.690568e+06	9.990650e+04	5.158575e+04
max	200.000000	12.000000	2023.000000	3.445520e+08	1.024570e+07	3.366021e+06

In [8]:
```
'''RNC and DNC are single events, I need to look out for other single events a
```

Out[8]:
```
'RNC and DNC are single events, I need to look out for other single events an
d remove them from the dataset.'
```

In [9]:
```
twitch_df[twitch_df['Peak_channels'] == 1]
```

Out[9]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
1329	130	RNC 2016	7	2016	264303	28	29021	

In [10]:
```
twitch_df[twitch_df['Game'] == "DNC 2016"]
```

Out[10]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
1287	88	DNC 2016	7	2016	362702	355	47549	24

In [11]:
```
twitch_df.drop([1329, 1287], inplace = True)
twitch_df.reset_index(inplace = True, drop = True)
```

In [12]:
```
#game_list is long, but you can review after downloading code and typing it. C
```

In [13]:
```
game_list = list(set(list(twitch_df['Game'])))
```

In [14]:
```
'''There were a number of things to review/fix from the the game list on twitc
-- all E3 and twitchcon events are single events and would skew data
-- special characters not appearing correctly are to be fixed
-- review near duplicate games, Final Fantasy XIV and Legend of Zelda Games ha

    As an example there are three versions of Legend of Zelda: Ocarina of time
    and tagged as different games so these will not be combined in any fashion

    Final Fantasy XIV: Online has several different iterations of names. Most
    be consistent. Final Fantasy XIV: Online was chosen. There was one instanc
    with Final Fantasy XIV: Online. If it was just a matter of this one overla
    that Heavensward is simply an expansion of Final Fantasy Online, Final Fan
    Final Fantasy XIV: Online and the one month of overlapping data will be co
    erroneously categorized as having dropped out of the top 200

'''
```

Out[14]:
```
'There were a number of things to review/fix from the the game list on twitch
\n-- all E3 and twitchcon events are single events and would skew data\n-- sp
ecial characters not appearing correctly are to be fixed\n-- review near dupl
icate games, Final Fantasy XIV and Legend of Zelda Games have some titles tha
t are almost identical. \n    \n    As an example there are three versions of
Legend of Zelda: Ocarina of time. After some research these are still being s
treamed\n    and tagged as different games so these will not be combined in a
ny fashion. With a similar rationale I did not combine others\n    \n    Fina
l Fantasy XIV: Online has several different iterations of names. Most are jus
t a function of capitalization and need to\n    be consistent. Final Fantasy
XIV: Online was chosen. There was one instance of Final Fantasy XIV: Heavensw
ard overlapping \n    with Final Fantasy XIV: Online. If it was just a matter
of this one overlapping month I would have left it as is. But given\n    that
Heavensward is simply an expansion of Final Fantasy Online, Final Fantasy XI
V: Heavensward will be changed to \n    Final Fantasy XIV: Online and the one
month of overlapping data will be combined. This is because otherwise this wo
uld be \n    erroneously categorized as having dropped out of the top 200\n
\n'
```

In [15]:
```
twitch_df[twitch_df['Game'].str.contains('TwitchCon') == True]
```

Out[15]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_char
1761	164	TwitchCon 2016	9	2016	119450	166	44758	
1848	51	TwitchCon 2016	10	2016	829794	725	125427	
4259	62	TwitchCon 2017	10	2017	878473	1602	86285	

In [16]:
```
twitch_df.drop([1761,1848,4259], inplace = True)
```

In [17]:
```
ff_df = twitch_df[twitch_df['Game'].str.contains('XIV') == True]
ff_df.head()
```

Out[17]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_ch
111	112	Final Fantasy XIV: Heavensward	1	2016	305520	33688	13084	
284	85	Final Fantasy XIV: Heavensward	2	2016	358212	37285	3246	
464	65	Final Fantasy XIV: Heavensward	3	2016	598914	43341	17436	
688	89	Final Fantasy XIV: Heavensward	4	2016	319118	39940	5466	
885	86	Final Fantasy XIV: Heavensward	5	2016	305008	36029	10178	

In [18]:
```
ff_list = list(set(list(ff_df['Game'])))
```

In [19]:
```
ff_list
```

Out[19]:
```
['FINAL FANTASY XIV ONLINE',
 'Final Fantasy XIV: Heavensward',
 'FINAL FANTASY XIV Online',
 'Romance of the Three Kingdoms XIV',
 'The King of Fighters XIV',
 'Final Fantasy XIV Online']
```

In [20]:
```
ff_list.remove('Romance of the Three Kingdoms XIV')
ff_list.remove('The King of Fighters XIV')
```

In [21]:
```
ff_list
```

Out[21]:
```
['FINAL FANTASY XIV ONLINE',
 'Final Fantasy XIV: Heavensward',
 'FINAL FANTASY XIV Online',
 'Final Fantasy XIV Online']
```

In [22]:
```
twitch_df.reset_index(drop = True, inplace = True)
```

In [23]:
```
twitch_df = twitch_df.replace(ff_list, 'Final Fantasy XIV: Online')
```

In [24]:
```
e3_df = twitch_df[twitch_df['Game'].str.contains('E3') == True]
e3_df.index
```

Out[24]:
```
Int64Index([1005, 3402, 3497, 5801, 5848, 5896], dtype='int64')
```

In [25]:
```
e3_df.describe()
```

Out[25]:

	Rank	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_cl
count	6.000000	6.0	6.000000	6.000000e+00	6.000000	6.000000e+00	6
mean	46.333333	6.0	2017.333333	8.987702e+06	18801.333333	8.606043e+05	1698
std	46.855807	0.0	0.816497	9.571019e+06	24962.272186	5.482893e+05	1843
min	6.000000	6.0	2016.000000	3.814990e+05	184.000000	3.882860e+05	177
25%	7.000000	6.0	2017.000000	8.738742e+05	1327.750000	4.375315e+05	408
50%	30.500000	6.0	2017.500000	7.549571e+06	9975.000000	7.290035e+05	1263
75%	90.000000	6.0	2018.000000	1.414050e+07	24437.750000	1.051961e+06	1993
max	102.000000	6.0	2018.000000	2.345944e+07	64970.000000	1.817345e+06	5125

In [26]:
```
twitch_df.drop(e3_df.index, inplace = True)
```

In [27]:
```
twitch_df.drop(twitch_df[twitch_df['Game'] == 'Twitch Presents'].index, inplac
twitch_df.reset_index(inplace = True, drop = True)
```

In [28]:
```
'''One thing that surprises me is the amount of classic games in this dataset.
that games that are popular are not necessarily new so having a recent release
popularity.
'''
```

Out[28]:
```
'One thing that surprises me is the amount of classic games in this dataset.
This will change the nature of the analysis in \nthat games that are popular
are not necessarily new so having a recent release date will likely negativel
y impact future \npopularity. \n'
```

In [29]:
```
twitch_df[twitch_df['Game'] =='Phoenix Wright: Ace Attorney <U+2212> Spirit of
```

Out[29]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_char
1791	196	Phoenix Wright: Ace Attorney <U+2212> Spirit o...	9	2016	96620	842	2287	

In [30]:
```
twitch_df = twitch_df.replace('Phoenix Wright: Ace Attorney <U+2212> Spirit of
```

In [31]:
```
twitch_df[twitch_df['Game'] =='Phoenix Wright: Ace Attorney - Spirit of Justic
```

Out[31]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_chann
1791	196	Phoenix Wright: Ace Attorney - Spirit of Justice	9	2016	96620	842	2287	

In [32]:
```
poke_df = twitch_df[twitch_df['Game'].str.contains('Pok') == True]
poke_df.head(10)
```

Out[32]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Pea
21	22	Poker	1	2016	3151063	36445	18034	
50	51	Pok<U+00E9>mon Omega Ruby/Alpha Sapphire	1	2016	992216	20237	11278	
78	79	Pok<U+00E9>mon Red/Blue	1	2016	456370	3276	185106	
79	80	Pok<U+00E9>mon Yellow	1	2016	449878	855	210369	
226	27	Poker	2	2016	2465052	36936	14039	
252	53	Pok<U+00E9>mon Omega Ruby/Alpha Sapphire	2	2016	1014081	17894	16439	
321	122	Pok<U+00E9>mon Red/Blue	2	2016	192855	4838	20162	
377	178	Pok<U+00E9>mon FireRed/LeafGreen	2	2016	117710	3260	18360	
422	23	Poker	3	2016	3534525	41966	25720	
444	45	Pok<U+00E9>mon Omega Ruby/Alpha Sapphire	3	2016	1246547	24192	6980	

In [33]:
```
poke_list = list(set(list(poke_df['Game'])))
```

In [34]:
```
poke_list.remove('Poker')
poke_list.remove('Prominence Poker')
```

In [35]:
```
twitch_df.reset_index(drop = True, inplace = True)
```

In [36]:
```
def clean (string):
    lst = []
    for letter in string:
        lst.append(letter)
    if '<' in lst:
        index = lst.index('<')
        del lst[index:index + 8]
        lst.insert(index, 'e')
        lst = ''.join(lst)
        return(lst)
    if 'Ã' in lst:
        index = lst.index('Ã')
        del lst[index:index + 2]
        lst.insert(index, 'e')
        lst = ''.join(lst)
        return(lst)
    else:
        lst = ''.join(lst)
        return(lst)
```

In [37]:
```
x = len(twitch_df)
for i in range(x):
    twitch_df['Game'][i] = str(twitch_df['Game'][i])
    twitch_df['Game'][i] = clean(twitch_df['Game'][i])
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\1143704737.py:3: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df['Game'][i] = str(twitch_df['Game'][i])
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\1143704737.py:4: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df['Game'][i] = clean(twitch_df['Game'][i])
```

In [38]:
```
poke_list[0]
```

Out[38]:
```
'PokÃ©mon GO'
```

In [39]:
```
poke_df = twitch_df[twitch_df['Game'].str.contains('Pok') == True]
poke_df.head(10)
```

Out[39]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Pea
21	22	Poker	1	2016	3151063	36445	18034	
50	51	Pokemon Omega Ruby/Alpha Sapphire	1	2016	992216	20237	11278	
78	79	Pokemon Red/Blue	1	2016	456370	3276	185106	
79	80	Pokemon Yellow	1	2016	449878	855	210369	
226	27	Poker	2	2016	2465052	36936	14039	
252	53	Pokemon Omega Ruby/Alpha Sapphire	2	2016	1014081	17894	16439	
321	122	Pokemon Red/Blue	2	2016	192855	4838	20162	
377	178	Pokemon FireRed/LeafGreen	2	2016	117710	3260	18360	
422	23	Poker	3	2016	3534525	41966	25720	
444	45	Pokemon Omega Ruby/Alpha Sapphire	3	2016	1246547	24192	6980	

In [40]:
```
twitch_df.head()
```

Out[40]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_chann
0	1	League of Legends	1	2016	94377226	1362044	530270	29
1	2	Counter-Strike: Global Offensive	1	2016	47832863	830105	372654	2
2	3	Dota 2	1	2016	45185893	433397	315083	1
3	4	Hearthstone	1	2016	39936159	235903	131357	5
4	5	Call of Duty: Black Ops III	1	2016	16153057	1151578	71639	36

In [41]:
```
#As a result of fixing the pokemon games God of War Ragnarok was changed, but
```

In [42]:
```
gow_df = twitch_df[twitch_df['Game'].str.contains('Ragnarek') == True]
```

In [43]:
```
gow_df
```

Out[43]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_cha
16392	11	God of War Ragnarek	11	2022	40440752	1617975	480655	
16637	56	God of War Ragnarek	12	2022	3798888	477077	34187	
16869	88	God of War Ragnarek	1	2023	1727833	257951	26831	
17175	194	God of War Ragnarek	2	2023	552004	122533	7011	

In [44]:
```
twitch_df = twitch_df.replace('God of War Ragnarek', 'God of War Ragnarok')
```

In [45]:
```
#As a result of fixing the pokemon games Okami was changed, but incorrectly, t
```

In [46]:
```
okami_df = twitch_df[twitch_df['Game'].str.contains('kami') == True]
okami_df
```

Out[46]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_char
4773	185	ekami	12	2017	181539	19273	29429	
13887	106	Demon Slayer - Kimetsu no Yaiba- The Hinokami C...	10	2021	1641397	93397	57275	

In [47]:
```
twitch_df = twitch_df.replace('ekami', 'Okami')
```

In [48]:
```
twitch_df.reset_index(drop = True, inplace = True)
```

In [49]:
```
twitch_df = twitch_df.sort_values(by = ['Game', 'Year', 'Month'])
twitch_df.head(10)
twitch_df.reset_index(drop = True, inplace = True)
```

In [50]:
```
x = len(twitch_df)
dup_l = []
for i in range(x-1):
    if twitch_df['Game'][i] == twitch_df['Game'][i+1] and twitch_df['Month'][i
        dup_l.append(i)
```

In [51]:
```
dup_l
```

Out[51]:
```
[518, 872, 3894, 4970, 11111, 11120, 11173, 11289, 11291, 13076]
```

In [52]:
```
examine = twitch_df.filter(items = dup_l, axis=0)
examine = examine.sort_values(by = ['Game', 'Year', 'Month'])
```

In [53]:
```
'''As we can see by the dataframe below there are a lot of duplicates, these w
One challenge in this is that Peak_viewers, Peak_channels, and Streamers are n
I will choose the max value of each.

For consistency I should compute the average viewers, channels, and viewer_rat
on the other columns in the dataframe for all columns.'''
```

Out[53]:
```
'As we can see by the dataframe below there are a lot of duplicates, these wi
ll also have to be combined in some fashion.\nOne challenge in this is that P
eak_viewers, Peak_channels, and Streamers are not computable. \nI will choose
the max value of each. \n\nFor consistency I should compute the average viewe
rs, channels, and viewer_ratio based \non the other columns in the dataframe
for all columns.'
```

In [54]:
```
examine
```

Out[54]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_cha
518	68	Among Us	1	2023	2760266	76925	137287	
872	15	Atlas	1	2019	12110031	187032	69614	
3894	75	Dungeons & Dragons	9	2018	790966	21961	38597	
4970	86	Final Fantasy XIV: Online	5	2017	533590	28073	35091	
11111	116	Resident Evil	10	2016	251183	3976	9871	
11120	13	Resident Evil 2	1	2019	16394917	449333	314244	
11173	11	Resident Evil 4	3	2023	27426481	1054133	327946	
11289	72	Retro	9	2021	2976840	131077	20554	
11291	83	Retro	10	2021	2484523	97914	10543	
13076	27	Stray	7	2022	11428809	508590	287683	

In [55]:
```
new_values= []
def add_col(col):
    new_values.clear()
    for i in dup_l:
        new_values.append(twitch_df[col][i] + twitch_df[col][i+1])
```

In [56]:
```
add_col('Hours_watched')
```

In [57]:
```
x = len(new_values)
def replace_cols(cols):
    for i in range(x):
        twitch_df.replace(twitch_df[cols][dup_l[i]], new_values[i], inplace =
```

In [58]:
```
replace_cols('Hours_watched')
```

In [59]:
```
twitch_df['Hours_watched'][518]
```

Out[59]: 3636990

In [60]:
```
add_col('Hours_streamed')
```

In [61]:
```
replace_cols('Hours_streamed')
```

In [62]:
```
def higher(col):
    new_values.clear()
    for i in dup_l:
        if twitch_df[col][i] > twitch_df[col][i+1]:
            new_values.append(twitch_df[col][i])
        else:
            new_values.append(twitch_df[col][i+1])
```

In [63]:
```
higher('Peak_viewers')
```

In [64]:
```
replace_cols('Peak_viewers')
```

In [65]:
```
higher('Peak_channels')
```

In [66]:
```
higher('Streamers')
```

In [67]:
```
'''
(Hours watched/hours in the month) is approximately the same as the current Av
(Hours streamed/hours in the month) is approximately the same as Avg_channels;

Avg_viewer_ratio is (Avg_viewers/Avg_channels); however when computing with th
by 0. To use these values I simply reversed the division to Avg_channels/Avg_v

I will be using these values instead given there are some duplicates not of my
Also, the data card was not forthcoming as to the provenance of these columns.
makes sense.

Rank will not be helpful at this point so I will remove'''
```

Out[67]:
```
'\n(Hours watched/hours in the month) is approximately the same as the curren
t Avg_viewers; \n(Hours streamed/hours in the month) is approximately the sam
e as Avg_channels;\n\nAvg_viewer_ratio is (Avg_viewers/Avg_channels); however
when computing with the values I found that I was sometimes dividing\nby 0. T
o use these values I simply reversed the division to Avg_channels/Avg_viewer
s.\n\nI will be using these values instead given there are some duplicates no
t of my own making that needed combined.\nAlso, the data card was not forthco
ming as to the provenance of these columns. Creating my own similar metrics\n
makes sense. \n\nRank will not be helpful at this point so I will remove'
```

In [68]:
```
no_leap = []
leap = []
for i in range(1,13):
    leap.append(monthrange(2004, i)[1] * 24)
    no_leap.append(monthrange(2003, i)[1] * 24)
```

In [69]:
```
no_leap
```

Out[69]:
```
[744, 672, 744, 720, 744, 720, 744, 744, 720, 744, 720, 744]
```

In [70]:
```
leap
```

Out[70]:
```
[744, 696, 744, 720, 744, 720, 744, 744, 720, 744, 720, 744]
```

In [71]:
```
x = len(twitch_df)
month_hours = []
for i in range(x):
    if twitch_df['Year'][i] %4:
        hours = leap[(twitch_df['Month'][i]-1)]
        month_hours.append(hours)
    else:
        hours = no_leap[(twitch_df['Month'][i]-1)]
        month_hours.append(hours)
```

In [72]:
```
twitch_df['Month_hours'] = month_hours
twitch_df.head()
```

Out[72]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channe
0	199	.hack//G.U. Last Recode	11	2017	145350	35258	1222	2
1	159	20 Minutes Till Dawn	6	2022	911356	12253	29743	
2	109	60 Parsecs!	9	2018	529688	1867	31960	
3	126	60 Seconds!	7	2016	268754	597	32505	
4	54	60 Seconds!	8	2016	772786	2065	56904	

In [73]:
```
twitch_df['Avg_viewers'] = round(twitch_df['Hours_watched']/twitch_df['Month_h
twitch_df.head()
```

Out[73]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channe
0	199	.hack//G.U. Last Recode	11	2017	145350	35258	1222	25
1	159	20 Minutes Till Dawn	6	2022	911356	12253	29743	8
2	109	60 Parsecs!	9	2018	529688	1867	31960	2
3	126	60 Seconds!	7	2016	268754	597	32505	1
4	54	60 Seconds!	8	2016	772786	2065	56904	3

In [74]:
```
twitch_df['Avg_channels'] = round(twitch_df['Hours_streamed']/twitch_df['Month
twitch_df.head()
```

Out[74]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channe
0	199	.hack//G.U. Last Recode	11	2017	145350	35258	1222	25
1	159	20 Minutes Till Dawn	6	2022	911356	12253	29743	8
2	109	60 Parsecs!	9	2018	529688	1867	31960	2
3	126	60 Seconds!	7	2016	268754	597	32505	1
4	54	60 Seconds!	8	2016	772786	2065	56904	3

In [75]:
```
twitch_df['Avg_channel_ratio'] = round(twitch_df['Avg_channels']/twitch_df['Av
twitch_df.head()
```

Out[75]:

	Rank	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channe
0	199	.hack//G.U. Last Recode	11	2017	145350	35258	1222	2
1	159	20 Minutes Till Dawn	6	2022	911356	12253	29743	
2	109	60 Parsecs!	9	2018	529688	1867	31960	
3	126	60 Seconds!	7	2016	268754	597	32505	
4	54	60 Seconds!	8	2016	772786	2065	56904	

In [76]:
```
twitch_df.drop(['Rank', 'Month_hours', 'Avg_viewer_ratio'], axis = 1, inplace
```

In [77]:
```
twitch_df.drop(twitch_df[twitch_df['Game'] == 'Twitch Presents'].index, inplac
twitch_df.reset_index(inplace = True, drop = True)
```

In [78]:
```
twitch_df.sort_values(by = ['Month', 'Year', 'Game'], inplace = True)
twitch_df.head(20)
```

Out[78]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
12	7 Days to Die	1	2016	269681	12131	4405	44
258	Agar.io	1	2016	255617	20705	4183	74
264	Age of Empires	1	2016	248884	232	107455	18
422	Alien: Isolation	1	2016	264294	11799	9590	42
477	American Truck Simulator	1	2016	314055	724	43089	48
694	Ark: Survival Evolved	1	2016	1951875	93060	19486	241
718	Arma 3	1	2016	2542838	86219	32132	275
911	Azure Striker GUNVOLT	1	2016	197178	217	135933	14
928	Banjo-Kazooie	1	2016	241250	2234	108131	28
967	BattleBlock Theater	1	2016	332256	2041	152739	19
1011	Battlefield 4	1	2016	672524	96185	6349	271
1139	Black Desert Online	1	2016	320675	15798	1888	44
1227	Blade & Soul	1	2016	5270251	184279	67256	1099
1268	Blast Corps	1	2016	178810	42	96318	18
1287	Bloodborne	1	2016	1300353	86384	199060	257
1403	Borderlands 2	1	2016	175135	21849	3376	71
1465	Brain Age: Train Your Brain in Minutes a Day!	1	2016	173011	19	207721	14
1661	Call of Duty: Black Ops II	1	2016	309850	29294	4491	100
1679	Call of Duty: Black Ops III	1	2016	16153057	1151578	71639	3620
2031	Cities: Skylines	1	2016	172577	8602	7371	30

In [79]:
```
df = list(twitch_df.Year.astype(str) + '/' + twitch_df.Month.astype(str) + '/0
twitch_df['Date'] = df
twitch_df['Date']= pd.to_datetime(twitch_df['Date'])
```

In [80]:
```
twitch_df_X = twitch_df.drop_duplicates(subset=['Game'])
twitch_df_X.head(10)
```

Out[80]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels	S
12	7 Days to Die	1	2016	269681	12131	4405	44	
258	Agar.io	1	2016	255617	20705	4183	74	
264	Age of Empires	1	2016	248884	232	107455	18	
422	Alien: Isolation	1	2016	264294	11799	9590	42	
477	American Truck Simulator	1	2016	314055	724	43089	48	
694	Ark: Survival Evolved	1	2016	1951875	93060	19486	241	
718	Arma 3	1	2016	2542838	86219	32132	275	
911	Azure Striker GUNVOLT	1	2016	197178	217	135933	14	
928	Banjo-Kazooie	1	2016	241250	2234	108131	28	
967	BattleBlock Theater	1	2016	332256	2041	152739	19	

In [81]:
```
twitch_df_X.describe()
```

Out[81]:

	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
count	2025.000000	2025.000000	2.025000e+03	2.025000e+03	2.025000e+03	2025.00000
mean	4.402469	2018.737778	1.722624e+06	5.265007e+04	5.925928e+04	342.25777
std	3.545371	2.247062	5.395172e+06	2.188540e+05	9.022342e+04	1071.92884
min	1.000000	2016.000000	9.246000e+04	1.900000e+01	7.470000e+02	2.00000
25%	1.000000	2017.000000	2.719870e+05	2.857000e+03	1.644000e+04	25.00000
50%	3.000000	2018.000000	6.132720e+05	1.046700e+04	3.332000e+04	76.00000
75%	7.000000	2021.000000	1.235417e+06	3.050200e+04	6.725600e+04	233.00000
max	12.000000	2023.000000	9.437723e+07	4.948200e+06	1.832845e+06	29306.00000

In [82]:
```
zero_df = twitch_df[twitch_df['Avg_channels'] == 0]
zero_df.head(11)
```

Out[82]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
264	Age of Empires	1	2016	248884	232	107455	18
911	Azure Striker GUNVOLT	1	2016	197178	217	135933	14
1268	Blast Corps	1	2016	178810	42	96318	18
1465	Brain Age: Train Your Brain in Minutes a Day!	1	2016	173011	19	207721	14
6404	Half Life 2: Survivor	1	2016	400106	195	164561	17
7098	Iji	1	2016	179556	30	91449	16
7470	Kirby 64: The Crystal Shards	1	2016	315368	209	158893	16
7478	Kirby: Squeak Squad	1	2016	299628	60	156564	16
8690	Mega Man 10	1	2016	197038	141	158783	29
8862	Mike Tyson's Punch-Out!!	1	2016	197881	307	168929	30
9502	Ninja Gaiden III: The Ancient Ship of Doom	1	2016	183550	122	133273	16

In [83]:
```
one_df = twitch_df[twitch_df['Avg_channels'] == 1]
one_df.head(11)
```

Out[83]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
477	American Truck Simulator	1	2016	314055	724	43089	48
2354	Crash Bandicoot	1	2016	399886	501	159957	29
2865	Darksiders II	1	2016	242099	752	120528	11
3650	Donkey Kong Country	1	2016	243490	826	160369	17
3652	Donkey Kong Country 2: Diddy's Kong Quest	1	2016	316101	993	181740	28
5059	Fire Emblem: Path of Radiance	1	2016	235555	516	122456	10
6411	Halo 4	1	2016	302218	569	135671	13
7983	M.U.G.E.N	1	2016	386581	944	976	5
8211	Mafia LIVE!	1	2016	183378	772	9532	8
8531	Mario Kart 64	1	2016	173981	605	186466	21
8788	Metroid Fusion	1	2016	234149	587	147009	22

In [84]:
```
two_df = twitch_df[twitch_df['Avg_channels'] == 2]
two_df.head(11)
```

Out[84]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels	S
4878	Final Fantasy IV	1	2016	535452	1158	147048	21	
5974	Grand Theft Auto III	1	2016	216252	1118	142392	22	
6141	Grid 2	1	2016	176713	1173	11348	9	
8694	Mega Man 2	1	2016	208756	1841	150406	121	
8697	Mega Man X	1	2016	180986	1415	181667	22	
8781	Metro: Last Light	1	2016	186279	1309	30280	13	
8790	Metroid Prime	1	2016	248704	1136	150677	25	
8958	Mirror's Edge	1	2016	209507	1528	186567	35	
9158	Move or Die	1	2016	202884	1275	32211	12	
10053	Paper Mario	1	2016	412453	1272	115905	16	
13447	Super Mario Bros. 3	1	2016	426084	1161	206252	17	

In [85]:
```
plus_one = 1
X = twitch_df_X.Date + pd.DateOffset(months=plus_one)
```

In [86]:
```
twitch_df_X['one_month_future'] =  twitch_df_X.Date + pd.DateOffset(months=plu
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\2229078347.py:1: SettingWith
CopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df_X['one_month_future'] =  twitch_df_X.Date + pd.DateOffset(months=
plus_one)
```

In [87]:
```
twitch_df_X.head()
```

Out[87]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels	St
12	7 Days to Die	1	2016	269681	12131	4405	44	
258	Agar.io	1	2016	255617	20705	4183	74	
264	Age of Empires	1	2016	248884	232	107455	18	
422	Alien: Isolation	1	2016	264294	11799	9590	42	
477	American Truck Simulator	1	2016	314055	724	43089	48	

In [88]:
```
plus_three = 3
X3 = twitch_df_X.Date + pd.DateOffset(months=plus_three)
```

In [89]:
```
twitch_df_X['three_month_future'] =  twitch_df_X.Date + pd.DateOffset(months=p
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\4009328254.py:1: SettingWith
CopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df_X['three_month_future'] =  twitch_df_X.Date + pd.DateOffset(month
s=plus_three)
```

In [90]:
```
twitch_df_X.head()
```

Out[90]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels	St
12	7 Days to Die	1	2016	269681	12131	4405	44	
258	Agar.io	1	2016	255617	20705	4183	74	
264	Age of Empires	1	2016	248884	232	107455	18	
422	Alien: Isolation	1	2016	264294	11799	9590	42	
477	American Truck Simulator	1	2016	314055	724	43089	48	

In [91]:
```
plus_six = 6
X6 = twitch_df_X.Date + pd.DateOffset(months=plus_six)
```

In [92]:
```
twitch_df_X['six_month_future'] =  twitch_df_X.Date + pd.DateOffset(months=plu
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\3018229530.py:1: SettingWith
CopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df_X['six_month_future'] =  twitch_df_X.Date + pd.DateOffset(months=
plus_six)
```

In [93]:
```
twitch_df_X.head()
```

Out[93]:

	Game	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels	St
12	7 Days to Die	1	2016	269681	12131	4405	44	
258	Agar.io	1	2016	255617	20705	4183	74	
264	Age of Empires	1	2016	248884	232	107455	18	
422	Alien: Isolation	1	2016	264294	11799	9590	42	
477	American Truck Simulator	1	2016	314055	724	43089	48	

In [94]:
```
twitch_df_X.reset_index(drop = True, inplace = True)
```

In [95]:
```
#twitch_df_X one_mnth_hrs = twitch_df['hours_watched'] where twitch_df['Date']
# if it doesn't exist make it 0
# current strategy append the index and value as a dict, when there is a missi
#and make the list the column hours_watched_omf
def f_hrs_watch (og_col_name, end_col_name):
    month_i = []
    month = []
    x = len(twitch_df_X)
    y = len(twitch_df)
    for i in range(x):
        for j in range(y):
            if twitch_df_X[og_col_name][i] == twitch_df['Date'][j] and twitch_
                month.append(twitch_df['Hours_watched'][j])
                month_i.append(i)
    zeros = []
    x = len(twitch_df_X)
    for i in range(x):
        if i not in month_i:
            zeros.append((i,0))
    merged_mth = list(zip(month_i, month))
    merged_z_mth = (merged_mth + zeros)
    merged_z_mth.sort()
    fin_mth = []
    for i in range(x):
        fin_mth.append(merged_z_mth[i][1])
    twitch_df_X[end_col_name] = fin_mth
```

In [96]:
```
f_hrs_watch('one_month_future', 'Hours_watched_1mth')
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\2513141890.py:26: SettingWit
hCopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df_X[end_col_name] = fin_mth
```

In [97]:
```
twitch_df_X.describe()
```

Out[97]:

	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
count	2025.000000	2025.000000	2.025000e+03	2.025000e+03	2.025000e+03	2025.000000
mean	4.402469	2018.737778	1.722624e+06	5.265007e+04	5.925928e+04	342.257778
std	3.545371	2.247062	5.395172e+06	2.188540e+05	9.022342e+04	1071.928845
min	1.000000	2016.000000	9.246000e+04	1.900000e+01	7.470000e+02	2.000000
25%	1.000000	2017.000000	2.719870e+05	2.857000e+03	1.644000e+04	25.000000
50%	3.000000	2018.000000	6.132720e+05	1.046700e+04	3.332000e+04	76.000000
75%	7.000000	2021.000000	1.235417e+06	3.050200e+04	6.725600e+04	233.000000
max	12.000000	2023.000000	9.437723e+07	4.948200e+06	1.832845e+06	29306.000000

In [98]:
```
twitch_df_X[twitch_df_X['Hours_watched_1mth'] != 0].describe()
```

Out[98]:

	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
count	809.000000	809.000000	8.090000e+02	8.090000e+02	8.090000e+02	809.000000
mean	3.138443	2018.784920	3.173144e+06	1.082643e+05	6.265248e+04	613.195303
std	2.998039	2.272461	8.149991e+06	3.359434e+05	9.823830e+04	1566.070656
min	1.000000	2016.000000	9.975600e+04	1.420000e+02	7.470000e+02	5.000000
25%	1.000000	2017.000000	4.677800e+05	9.610000e+03	1.388000e+04	58.000000
50%	1.000000	2019.000000	9.968500e+05	2.447200e+04	3.505000e+04	157.000000
75%	5.000000	2021.000000	2.443756e+06	7.501500e+04	6.746900e+04	545.000000
max	11.000000	2023.000000	9.437723e+07	4.948200e+06	1.249796e+06	29306.000000

In [99]:
```
f_hrs_watch('three_month_future', 'Hours_watched_3mth')
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\2513141890.py:26: SettingWit
hCopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df_X[end_col_name] = fin_mth
```

In [100]:
```
f_hrs_watch('six_month_future', 'Hours_watched_6mth')
```

```
C:\Users\danie\AppData\Local\Temp\ipykernel_3888\2513141890.py:26: SettingWit
hCopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  twitch_df_X[end_col_name] = fin_mth
```

In [101]:
```
twitch_df_X[twitch_df_X['Hours_watched_3mth'] != 0].describe()
```

Out[101]:

	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
count	408.000000	408.000000	4.080000e+02	4.080000e+02	408.000000	408.000000
mean	1.610294	2018.200980	4.026296e+06	1.419502e+05	58421.044118	572.259804
std	1.603924	2.132883	1.022881e+07	3.902082e+05	98120.193932	1298.057629
min	1.000000	2016.000000	9.975600e+04	7.230000e+02	747.000000	5.000000
25%	1.000000	2016.000000	3.932162e+05	1.096575e+04	8568.250000	53.750000
50%	1.000000	2018.000000	1.018976e+06	3.184700e+04	25098.000000	136.000000
75%	1.000000	2020.000000	3.134386e+06	1.077270e+05	60421.750000	449.500000
max	9.000000	2022.000000	9.437723e+07	4.948200e+06	837094.000000	13547.000000

In [102]:
```
twitch_df_X[twitch_df_X['Hours_watched_6mth'] != 0].describe()
```

Out[102]:

	Month	Year	Hours_watched	Hours_streamed	Peak_viewers	Peak_channels
count	331.000000	331.000000	3.310000e+02	3.310000e+02	331.000000	331.000000
mean	1.247734	2018.193353	4.408722e+06	1.589651e+05	60451.353474	578.939577
std	0.758159	2.134582	1.117714e+07	4.267634e+05	104835.481296	1334.597625
min	1.000000	2016.000000	1.041330e+05	5.390000e+02	976.000000	5.000000
25%	1.000000	2016.000000	4.295595e+05	1.326300e+04	8420.500000	52.000000
50%	1.000000	2018.000000	1.025337e+06	3.705600e+04	23330.000000	137.000000
75%	1.000000	2020.000000	3.166135e+06	1.188490e+05	60488.500000	505.500000
max	6.000000	2022.000000	9.437723e+07	4.948200e+06	837094.000000	13547.000000

In [105]:
```
twitch_df.to_pickle('twitch_df_og.pkl')
```

In [106]:
```
twitch_df_X.to_pickle('twitch_df_wrng.pkl')
```

In [ ]:
```

```