

Games Popularity Analysis

Jatupong Oboun
62010096
KMITL





Report: The International 10 sets audience records

By FIELD LEVEL MEDIA

2 MIN READ



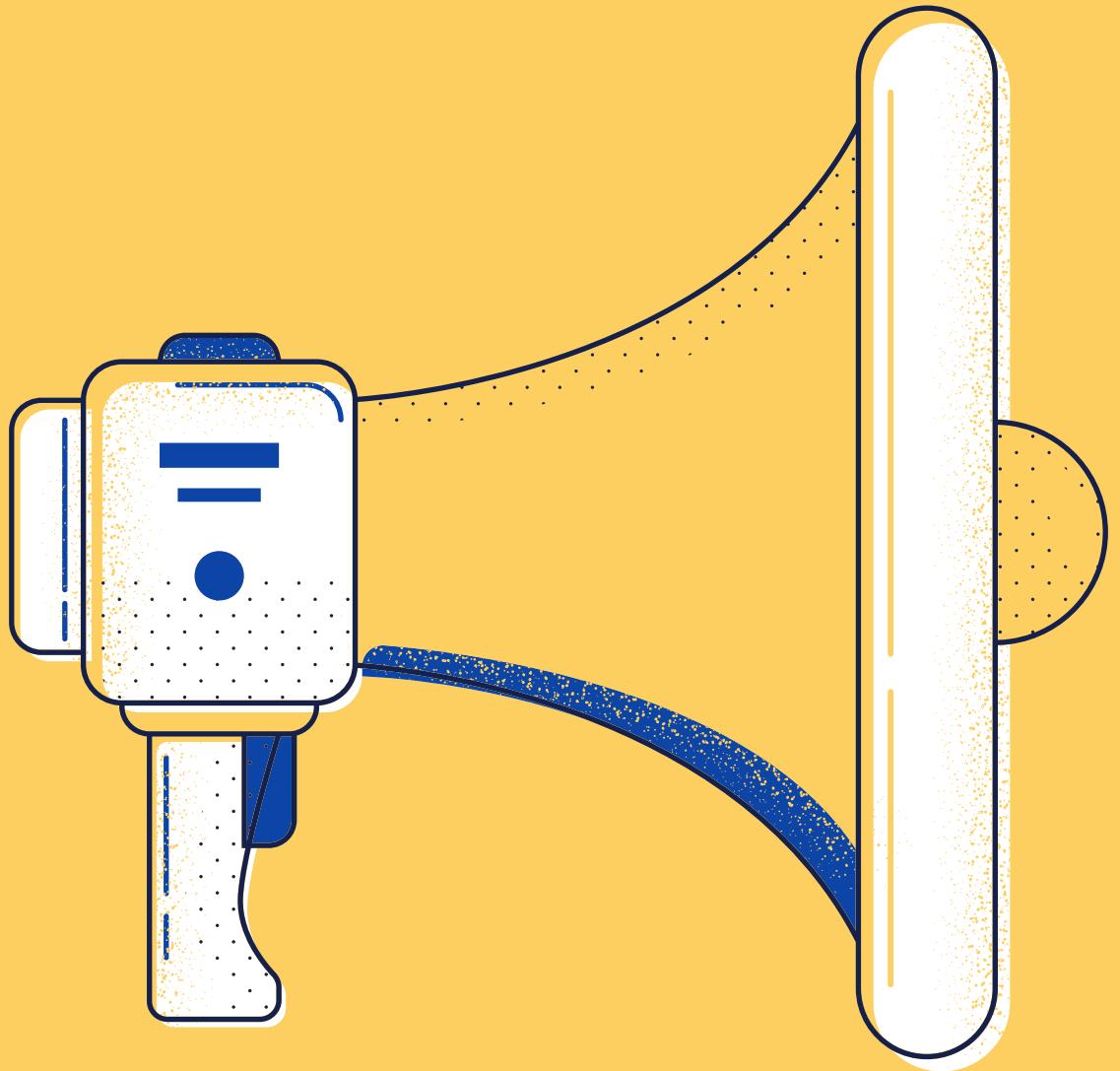
The grand final peaked at 2.7 million viewers (across all platforms excluding China), making it the most watched Dota 2 event ever. The figure represented a 37 percent increase over its peak in 2019. (The event wasn't held in 2020 due to the coronavirus pandemic).

| Place | \$ USD | Percent | Team |
|------------------|--------------|---------|------------------|
| 1st | \$18,208,300 | 45.5% | Team Spirit |
| 2nd | \$5,202,400 | 13% | PSG.LGD |
| 3rd | \$3,601,600 | 9% | Team Secret |
| 4th | \$2,401,100 | 6% | Invictus Gaming |
| place 5 to 18 ^ | | | |
| 5th-6th | \$1,400,600 | 3.5% | Virtus.pro |
| 7th-8th | \$1,000,500 | 2.5% | Vici Gaming |
| 9th-12th | \$800,400 | 2% | OG |
| T1 | | | T1 |
| Fnatic | | | Fnatic |
| Quincy Crew | | | Quincy Crew |
| Alliance | | | Alliance |
| Evil Geniuses | | | Evil Geniuses |
| Team Undying | | | Team Undying |
| Team Aster | | | Team Aster |
| beastcoast | | | beastcoast |
| Elephant | | | Elephant |
| Thunder Predator | | | Thunder Predator |
| SG esports | | | SG esports |



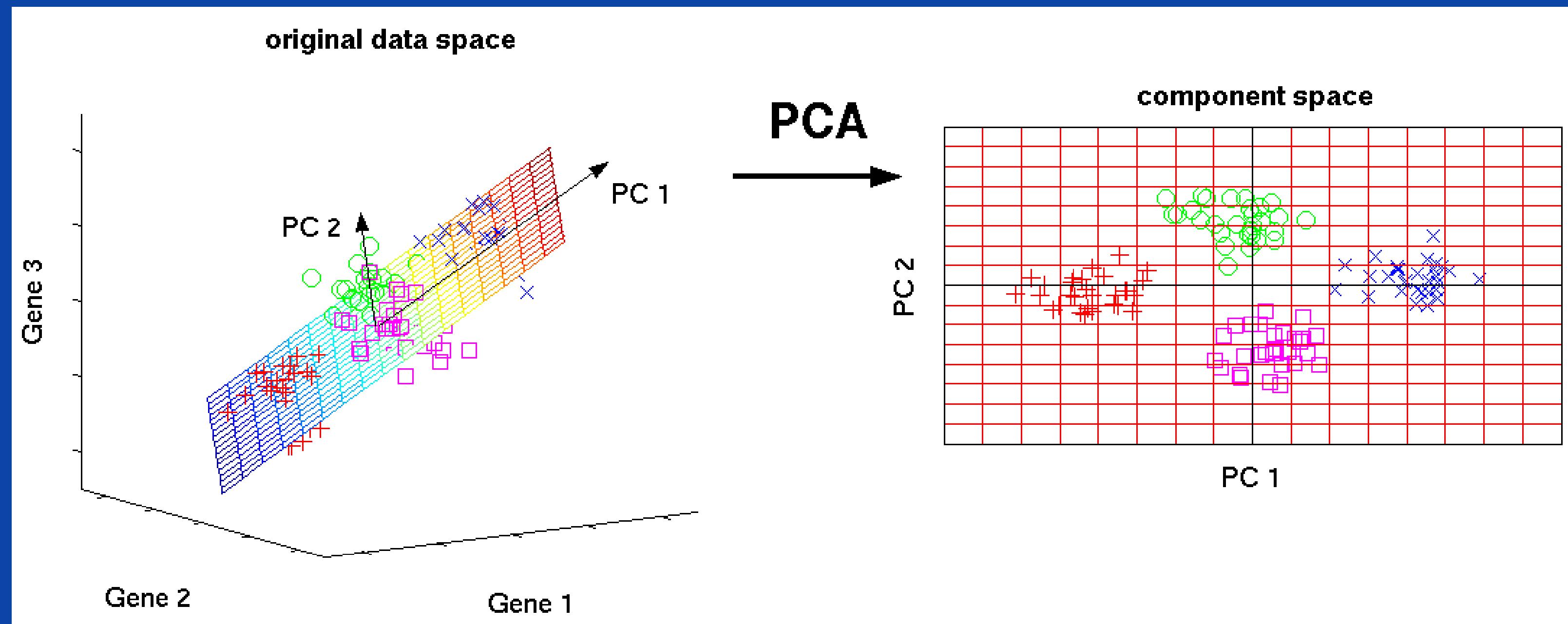
MOTIVATION

นำข้อมูลในวงการเกมส์ มา
วิเคราะห์หา Insight และ^{เพื่อ}
แบ่งกลุ่มของเกมส์ เพื่อนำ^{เพื่อ}
ไปใช้ในด้านผลประโยชน์ใน
ภาคธุรกิจ และ อื่นๆ

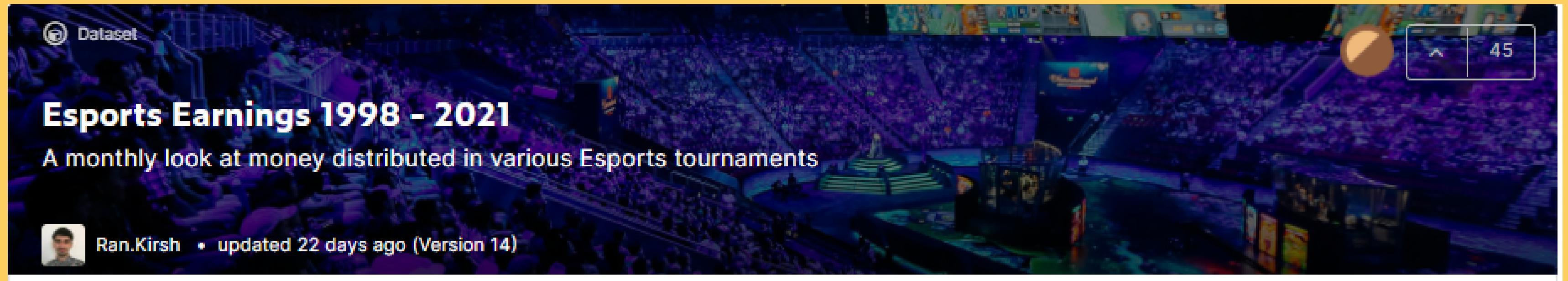


PCA

เพื่อลดมิติของข้อมูลที่มี Relation ต่อกันสูง ให้เหลือตัวแปรที่น้อยลง ทำให้สามารถเข้า Model ได้ง่ายและประสิทธิภาพสูงขึ้น



DATASETS



| Date | Game | Earnings | Players | Tournaments |
|--------|----------------------|----------|----------------------------|-------------|
| Month | Game | Earnings | Players receiving earnings | Tournaments |
| 1Jan98 | StarCraft: Brood War | 3% | 36.7m | 1699 |
| | WarCraft III | 3% | 0 | 1 |
| | Other (6599) | 94% | 0 | 172 |

Ref : kaggle.com/rankirsh/esports-earnings

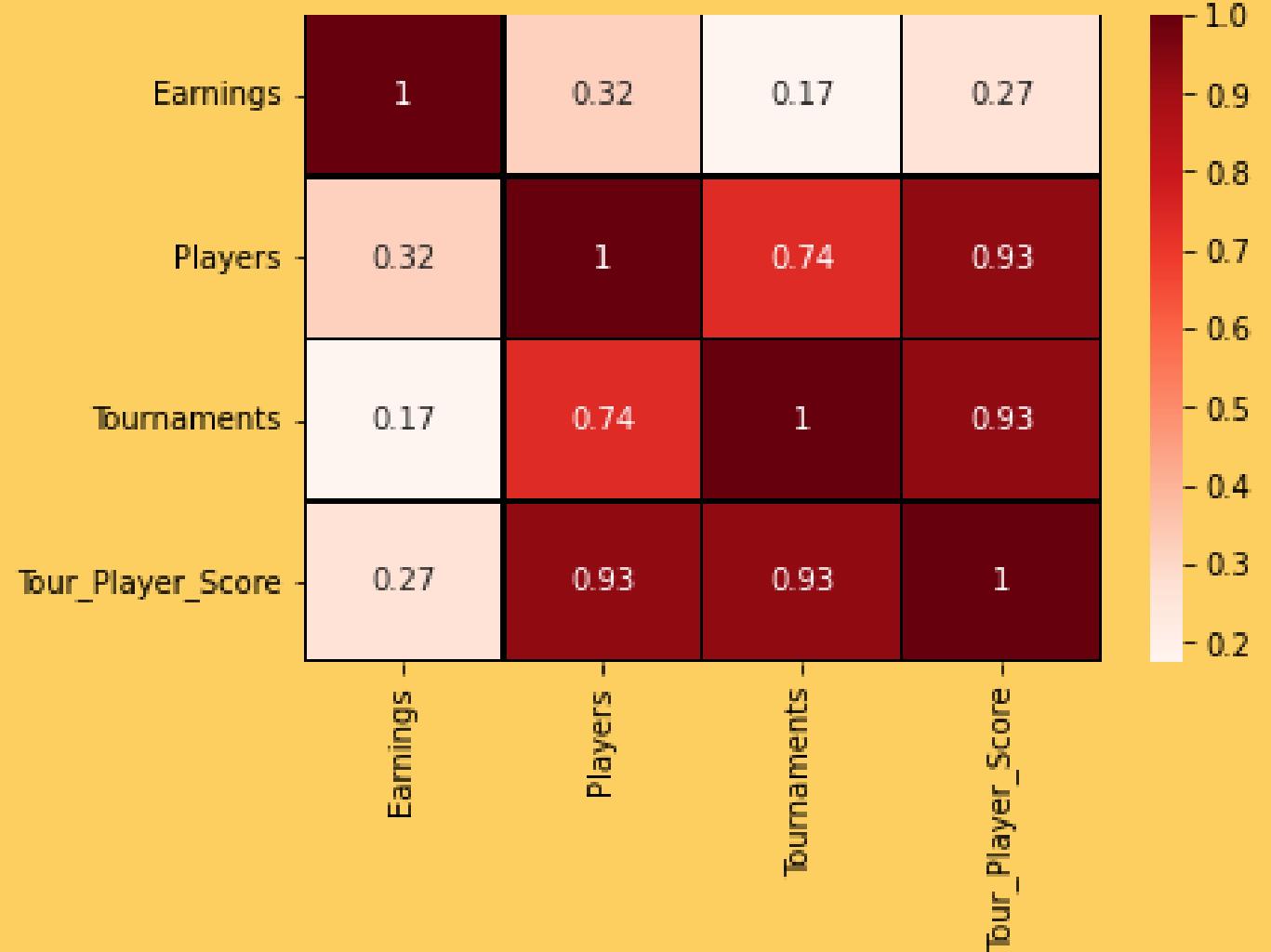
```
✓ [6] from sklearn.decomposition import PCA  
pca = PCA(n_components=1)  
PCA = pca.fit_transform(tournament[['Players','Tournaments']])  
tournament['Tour_Player_Score'] = PCA
```

```
✓ [38] (pca.components_[0][1])
```

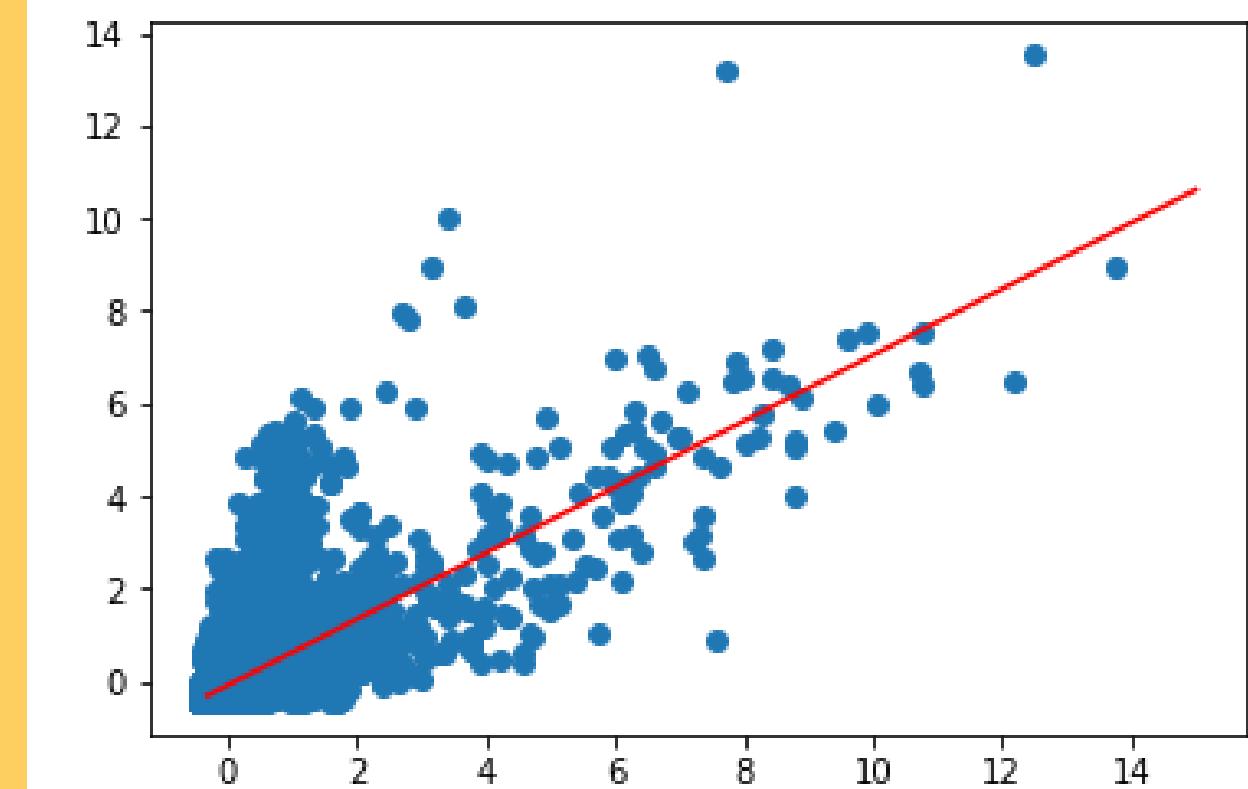
0.7071067811865477

```
✓ [64] x = [(pca.components_[0][0]*min(tournament['Players']),15]  
y = [(pca.components_[0][1]*min(tournament['Tournaments'])),(pca.components_[0][1]*15)]  
#(pca.components_[0][0]*min(tournament['Tournaments']))[0]
```

```
✓ [65] a = list(tournament['Tournaments'])  
b = list(tournament['Players'])
```



```
✓ [66] import matplotlib.pyplot as plt  
  
plt.plot(x, y, 'r') # plotting t, a separately  
plt.scatter(b,a)  
plt.show()
```



```
pca.explained_variance_ratio_
```

```
array([0.86846764, 0.13153236])
```



Top games on Twitch 2016 - 2021

Monthly top 200 games on the platform



Ran.Kirsh • updated 22 days ago (Version 9)

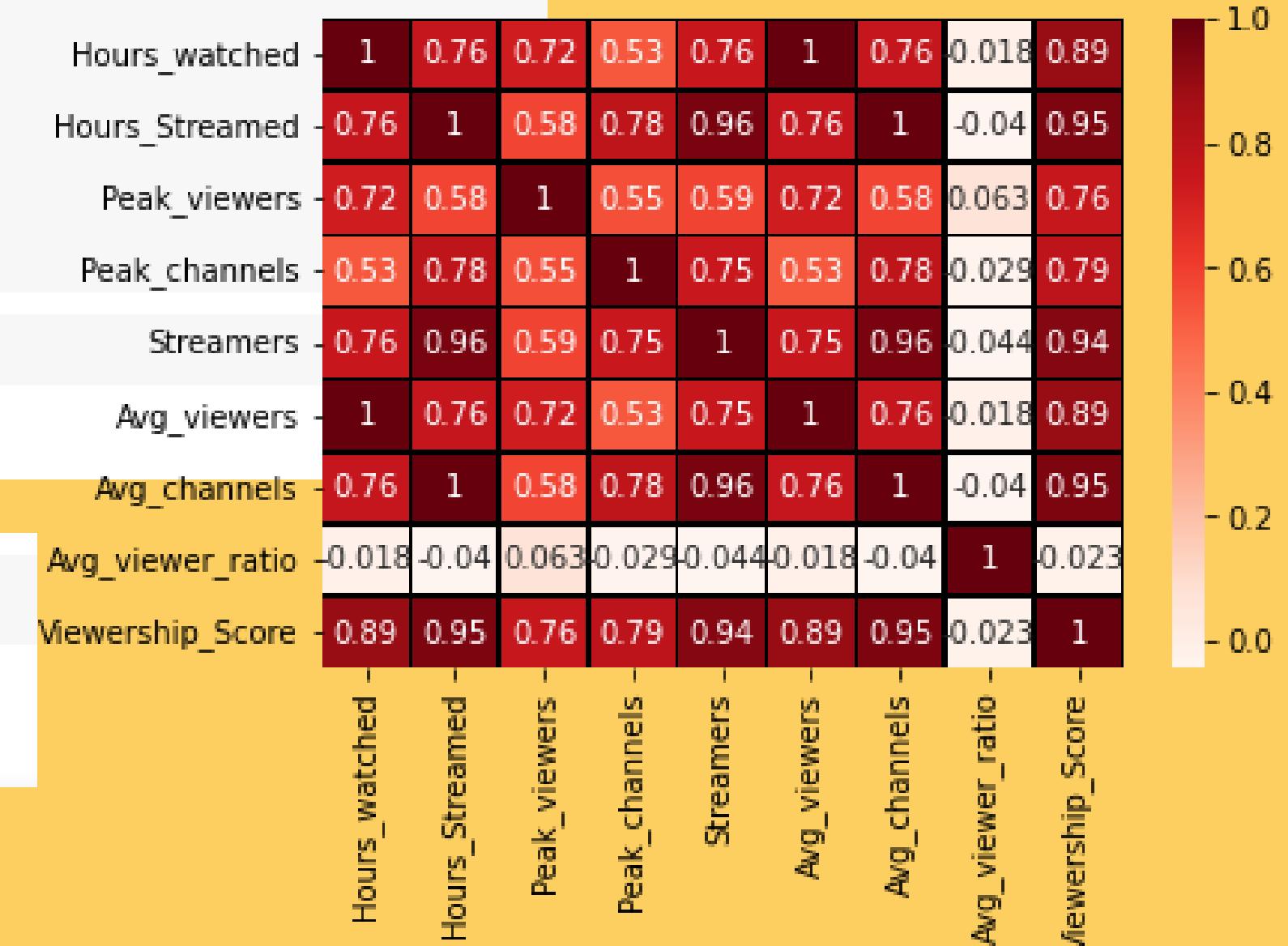
| # Rank | A Game | # Month | # Year | # Hours_watched | A Hours_Streamed |
|--------------------------------|---------------------------------|---|------------------|----------------------------|--------------------------------------|
| Name of game or category | Rank in the month (1 - 200) | Month in question | Year in question | Hours watched on twitch | Hours streamed on twitch |
| | 1738 unique values | | | | 12814 unique values |
| # Peak_viewers | # Peak_channels | # Streamers | # Avg_viewers | # Avg_channels | # Avg_viewer_ratio |
| Maximum viewers at one instant | Maximum channels at one instant | Amount of streamers who streamed the game | Average viewers | Average amount of channels | Average amount of viewer per channel |
| | | | | | |

```
[67] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    twitch = pd.read_csv('https://github.com/markerxz/games-popularity-analysis/raw/master/Twitch_game_data.csv', encoding='cp1252')
    features = ['Hours_watched', 'Hours_Streamed', 'Peak_viewers', 'Peak_channels', 'Streamers', 'Avg_viewers', 'Avg_channels']

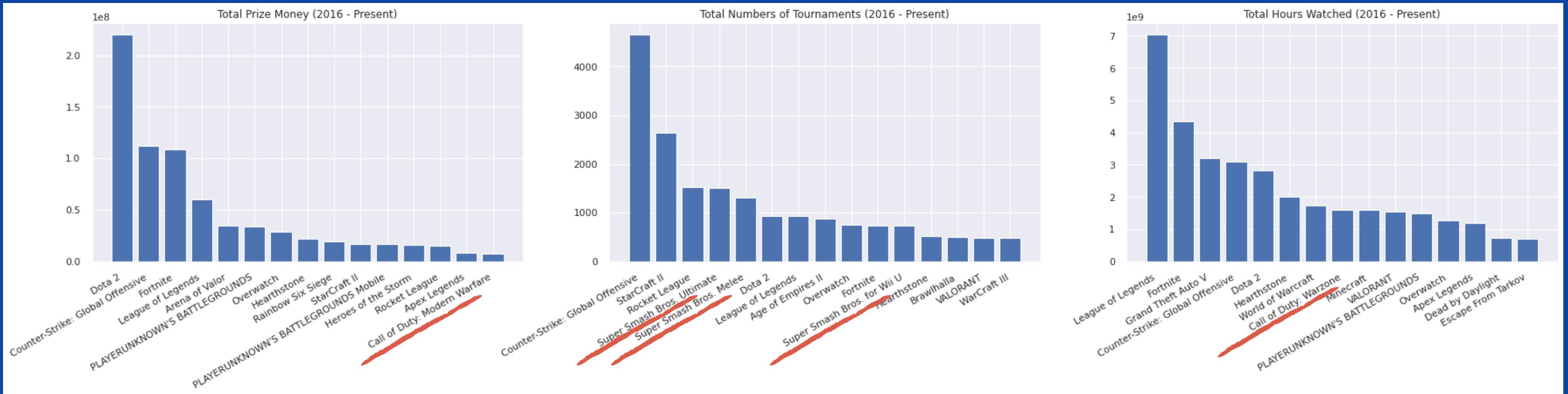
    from sklearn.preprocessing import StandardScaler
    hey = twitch.loc[:, 'Hours_Streamed'].values
    for i in range(len(hey)):
        data = hey[i]
        val = int(data.split()[0])
        hey[i] = val
    scaler = StandardScaler()
    twitch[features] = scaler.fit_transform(twitch[features])
    from sklearn.decomposition import PCA
    pca = PCA(n_components=1)
    PCA = pca.fit_transform(twitch[features])
    twitch['Viewership_Score'] = PCA

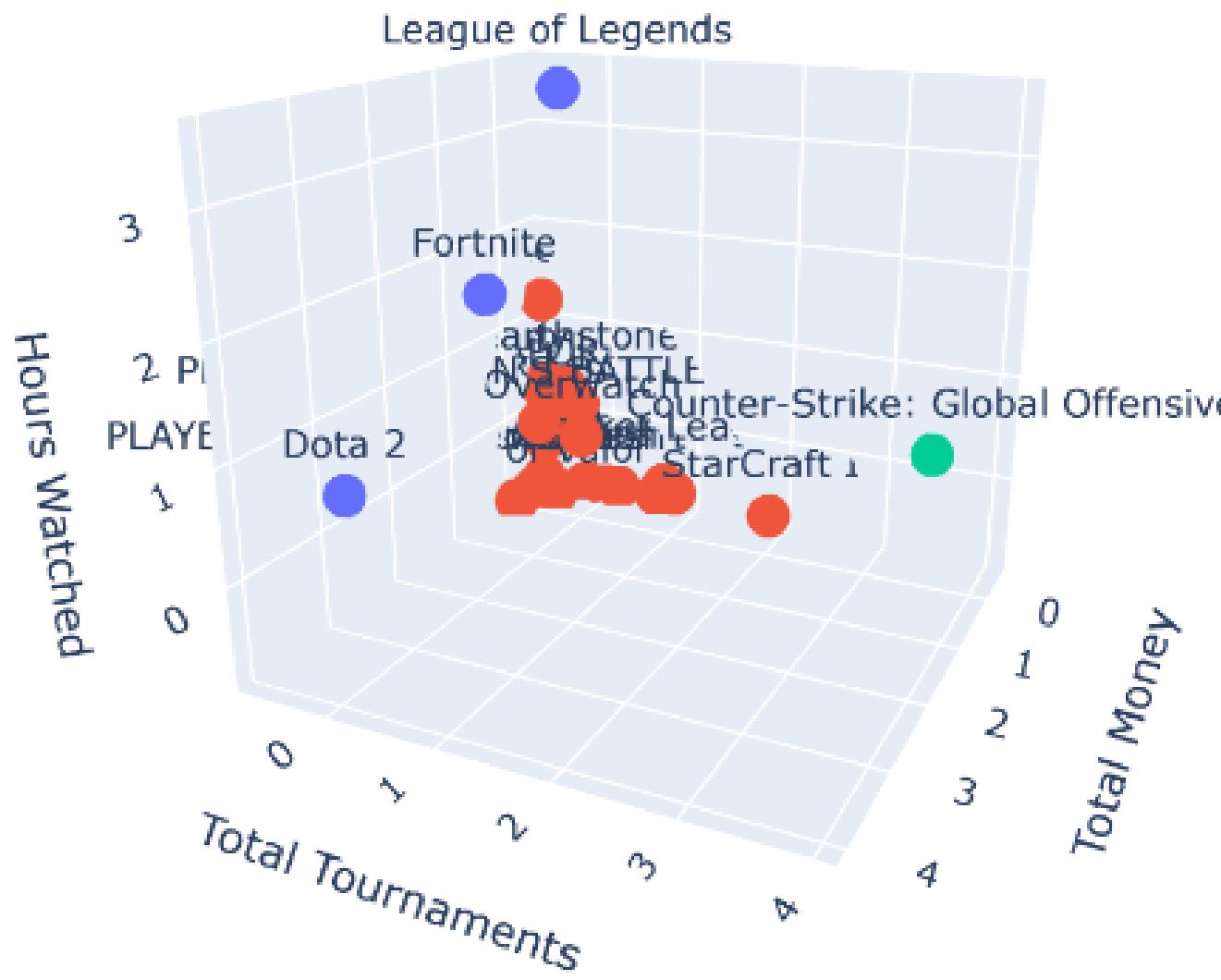
[68] pca.components_
array([[0.38182488, 0.40624682, 0.32232261, 0.33770615, 0.40050088,
       0.38174096, 0.40633589]])

[13] pca.explained_variance_ratio_
array([7.84502464e-01, 1.14329373e-01, 6.61829394e-02, 2.79989884e-02,
       6.86970156e-03, 1.04057899e-04, 1.24756729e-05])
```



Top 15 Games in Each Categories





Cluster : 0
Super Smash Bros. Ultimate
Hearthstone
Call of Duty: Modern Warfare
PLAYERUNKNOWN'S BATTLEGROUNDS
Heroes of the Storm
Grand Theft Auto V
Arena of Valor
Age of Empires II
Brawlhalla
StarCraft II
World of Warcraft
Minecraft
Overwatch
Apex Legends
Super Smash Bros. for Wii U
Call of Duty: Warzone
VALORANT
Escape From Tarkov
Dead by Daylight
Rocket League
WarCraft III
PLAYERUNKNOWN'S BATTLEGROUNDS Mobile
Rainbow Six Siege
Super Smash Bros. Melee

```
!pip install fuzzywuzzy
from fuzzywuzzy import fuzz

def difscore(string1,string2):
    temp1 = ""
    for s in string1.lower():
        if ord('a')<=ord(s)<=ord('z'):
            temp1 += s
    temp2 = ""
    for s in string2.lower():
        if ord('a')<=ord(s)<=ord('z'):
            temp2 += s
    return max(fuzz.token_set_ratio(temp1,temp2),fuzz.partial_ratio(string1,string2))
```

```
print(difscore('Ultra Street Fighter IV','Street Fighter V'))
print(difscore('Animal Crossing: New Horizons','Animal Crossing: New Leaf'))
print(difscore('FIFA 07','FIFA 21'))
print(difscore('Counter-Strike: Global Offensive','Counter-Strike: Condition Zero'))
```

94

84

100

63

```
game_cluster = []
thereshould = 88
cc = 0
for g in sorted(tour.keys()):
    cc += 1
    if cc%500 == 0:
        print('i = ',cc)
    clusterno = 0
    if not game_cluster:
        game_cluster.append([g])
    else:
        is_clustered = False
        for i in range(len(game_cluster)):
            gc = game_cluster[i]

            score = 0
            for each in gc:
                score = max(difscore(g,each),score)
            if score>=thereshould:
                is_clustered = True
                gc.append(g)
                clusterno = i
                break
        if not is_clustered:
            game_cluster.append([g])
            clusterno = len(game_cluster)-1
tour[g]['cluster no.']=clusterno
```

```
['Counter-Strike',
 'Counter-Strike Online',
 'Counter-Strike: Condition Zero',
 'Counter-Strike: Global Offensive',
 'Counter-Strike: Source'],
 ['Counter-Strike']
```

```
['Call Of Duty: Modern Warfare',
 'Call of Duty',
 'Call of Duty 2',
 'Call of Duty 4: Modern Warfare',
 'Call of Duty: Advanced Warfare',
 'Call of Duty: Black Ops',
 'Call of Duty: Black Ops 2',
 'Call of Duty: Black Ops 4',
 'Call of Duty: Black Ops Cold War',
 'Call of Duty: Black Ops II',
 'Call of Duty: Black Ops III',
 'Call of Duty: Blackout',
 'Call of Duty: Ghosts',
 'Call of Duty: Infinite Warfare',
 'Call of Duty: Mobile',
 'Call of Duty: Modern Warfare',
 'Call of Duty: Modern Warfare 2',
 'Call of Duty: Modern Warfare 3',
 'Call of Duty: Modern Warfare Remastered',
 'Call of Duty: Vanguard',
 'Call of Duty: WWII',
 'Call of Duty: Warzone',
 'Call of Duty: World War II',
 'World War 3',
 'World War Z',
 'World War Z: Aftermath'],
 ['Call of Duty']
```

```
['FIFA 03',
 'FIFA 04',
 'FIFA 05',
 'FIFA 06',
 'FIFA 07',
 'FIFA 08',
 'FIFA 09',
 'FIFA 10',
 'FIFA 11',
 'FIFA 12',
 'FIFA 13',
 'FIFA 14',
 'FIFA 15',
 'FIFA 16',
 'FIFA 17',
 'FIFA 18',
 'FIFA 19',
 'FIFA 20',
 'FIFA 2000',
 'FIFA 2001',
 'FIFA 2002 World Cup',
 'FIFA 21',
 'FIFA 22'],
 ['FIFA Online 3', 'FIFA Online 4'],
 ['FIFA Online 5']
```

```
['Hyper Street Fighter II',
 'Street Fighter IV',
 'Street Fighter V',
 'Street Fighter V: Arcade Edition',
 'Street Fighter V: Champion Edition',
 'Street Fighter X Tekken',
 'Super Street Fighter II Turbo',
 'Super Street Fighter II Turbo HD Remix',
 'Super Street Fighter IV',
 'Super Street Fighter IV Arcade Edition',
 'Ultra Street Fighter IV'],
 ['Street Fighter']
```

```

game_cluster_name = []
cc = 0
import re
for cluster in (game_cluster):
    #print(cluster)
    if len(cluster) == 1:
        game_cluster_name.append(cluster[0])
    else:
        arr_of_dict = []

        for g in cluster:
            gsplit = re.split(" |',|:", g)

            #print(gsplit)
            for i in range(len(gsplit)):
                gsplit[i] = gsplit[i].upper()
                if len(arr_of_dict)<=i:
                    arr_of_dict.append({'sumword':0})
                if gsplit[i] not in arr_of_dict[i]:
                    arr_of_dict[i][gsplit[i]] = 1
                    arr_of_dict[i]['sumword'] += 1
                else:
                    arr_of_dict[i][gsplit[i]] += 1
                    arr_of_dict[i]['sumword'] += 1
            seriesname = ""
            for i in range(len(arr_of_dict)):
                d = arr_of_dict[i]
                templ = sorted([[dk,d[dk]]for dk in d.keys() if dk!='sumword'],key = lambda x:(x[1]))[::-1]
                if templ[0][1]/d['sumword']>0.5 and d['sumword'] != 1:

                    seriesname += templ[0][0]+ ' '
                else:
                    if seriesname == '':
                        seriesname = min(cluster, key=len) + ' '
                    break
            game_cluster_name.append(seriesname+"Series")
            if seriesname == '':
                print(arr_of_dict)
        cc += 1

```

```

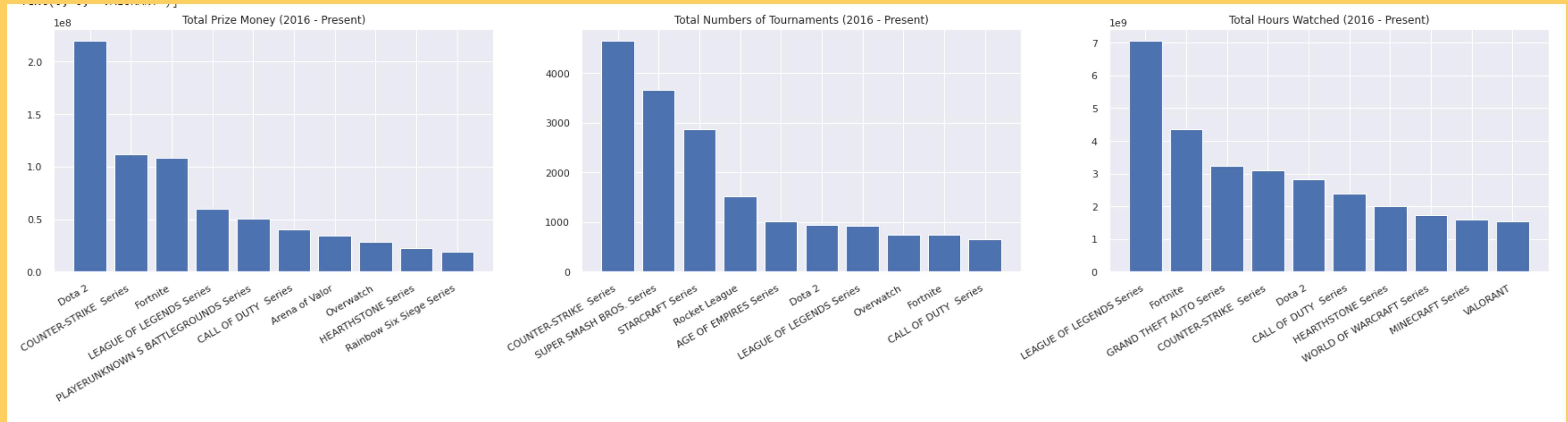
'BORDERLANDS Series',
'Batman: The Telltale Series',
'CALL OF DUTY Series',
'CAPCOM VS. SNK Series',
'CAR MECHANIC SIMULATOR Series',
'CASTLEVANIA Series',
'CATHERINE Series',
'COMMAND & CONQUER Series',
'COUNTER-STRIKE Series',
'CRASH BANDICOOT Series',
'CRUSADER KINGS Series',
'Casino Series',
'Chess Series',
'Crea Series',
'DARK SOULS II Series',
'DARKSIDERS Series',
'DAY OF DEFEAT Series',
'DEAD BY DAYLIGHT Series',
'DEAD OR ALIVE Series',
'DEAD RISING Series',
'DEAD SPACE Series',
'MONSTER HUNTER Series',
'MORDHAU Series',
'MORTAL KOMBAT Series',
'MOUNT YOUR FRIENDS Series',
'Mario Party 10 Series',
'NARAKA BLADEPOINT Series',
'NARUTO SHIPPUDEN ULTIMATE NINJA STORM Series',
'NASCAR HEAT Series',
'NASCAR THUNDER Series',
'NBA Series',
'NEED FOR SPEED Series',
'NHL Series',
'INTERACTIVISION VERSUS SERIES'

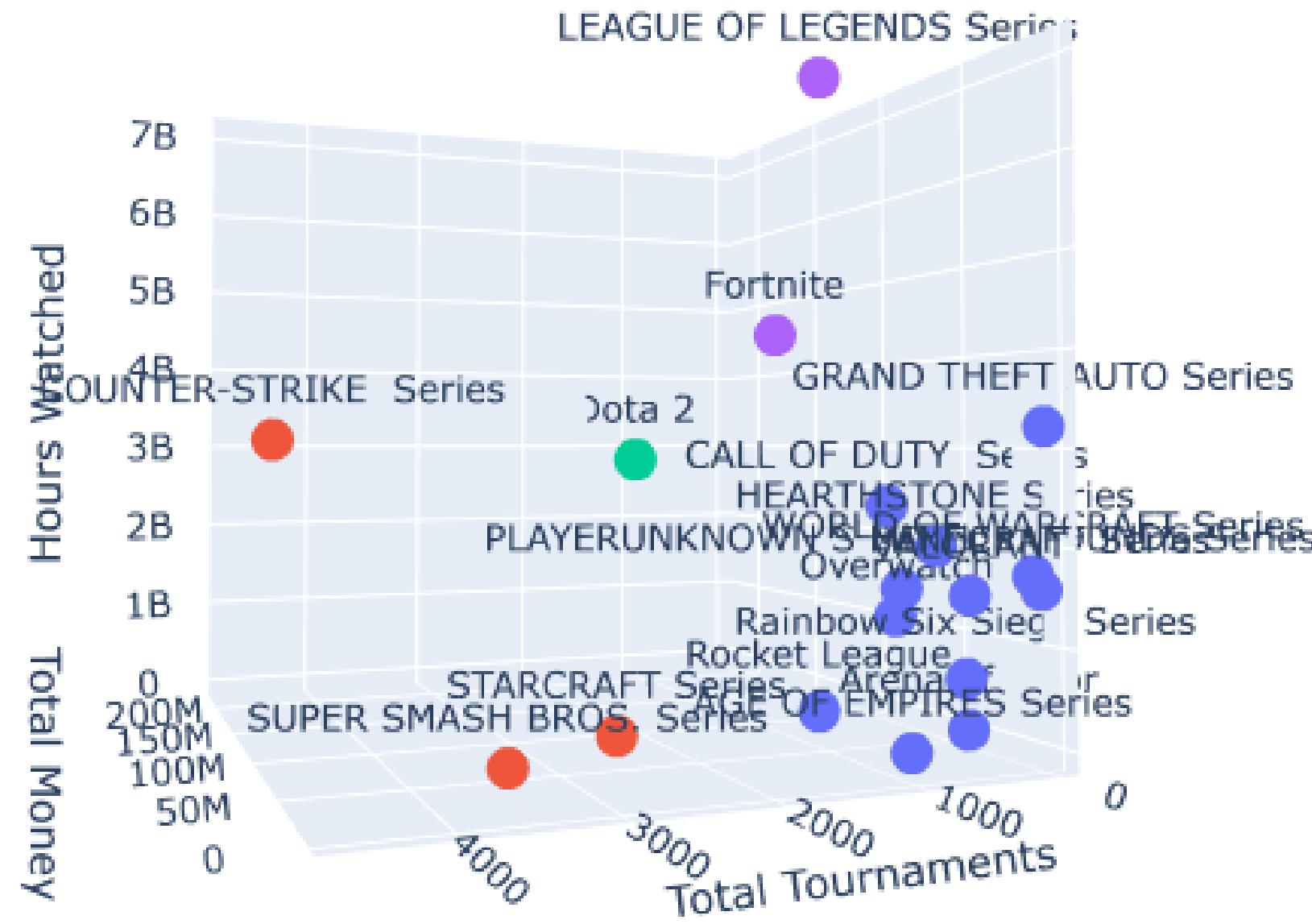
```

```
[ ] tour['Counter-Strike: Global Offensive']

{'cluster no.': 211,
 'cluster_name': 'COUNTER-STRIKE Series',
 'hw': {2016: 526391918,
 2017: 419802259,
 2018: 401775324,
 2019: 457874735,
 2020: 713749581,
 2021: 571944429,
 'total': 3091538246,
 'total2016+': 3091538246},
 'money': {2012: 222538.7700000002,
 2013: 1211869.54,
 2014: 1962515.01,
 2015: 6284970.140000001,
 2016: 17266132.499999996,
 2017: 19279544.51,
 2018: 23320449.03,
 2019: 22214116.83,
 2020: 15943485.719999999,
 2021: 14140226.34,
 'total': 121845848.38999996,
 'total2016+': 112163954.92999995},
 'ntour': {2012: 48,
 2013: 199,
 2014: 279,
 2015: 707,
 2016: 861,
 2017: 906,
 2018: 1020,
 2019: 941,
 2020: 595,
 2021: 338,
 'total': 5894,
 'total2016+': 4661}}]
```

Top 10 Series in Each Categories





Cluster : 0

CALL OF DUTY Series
PLAYERUNKNOWN S BATTLEGROUNDS Series
Rainbow Six Siege Series
Rocket League
WORLD OF WARCRAFT Series
VALORANT
Arena of Valor
GRAND THEFT AUTO Series
AGE OF EMPIRES Series
MINECRAFT Series
Overwatch
HEARTHSTONE Series

Cluster : 1

SUPER SMASH BROS. Series
COUNTER-STRIKE Series
STARCRAFT Series

Cluster : 2

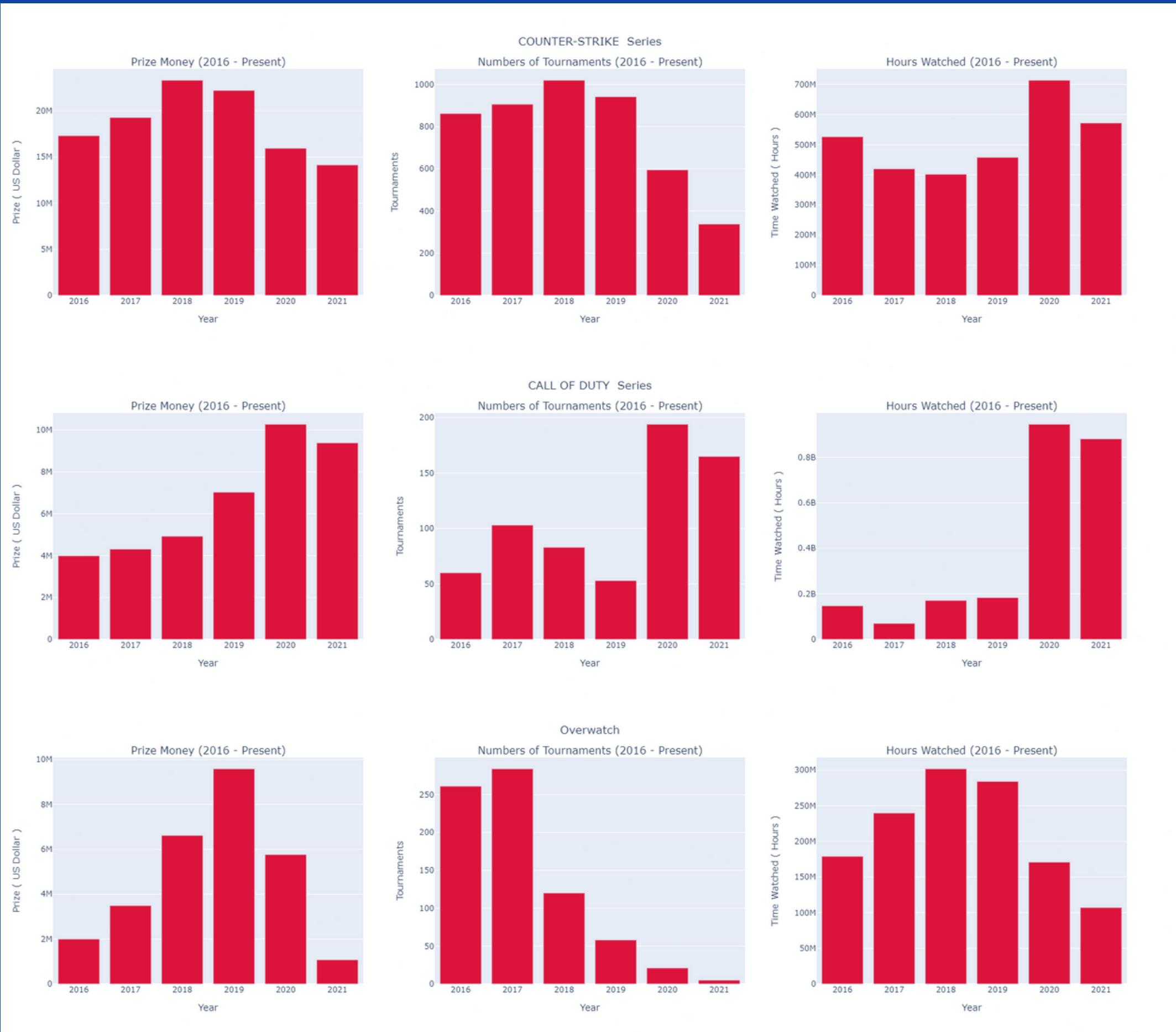
Dota 2

Cluster : 3

Fortnite
LEAGUE OF LEGENDS Series

**Same weight for
every year.**

Is it fair ?



```

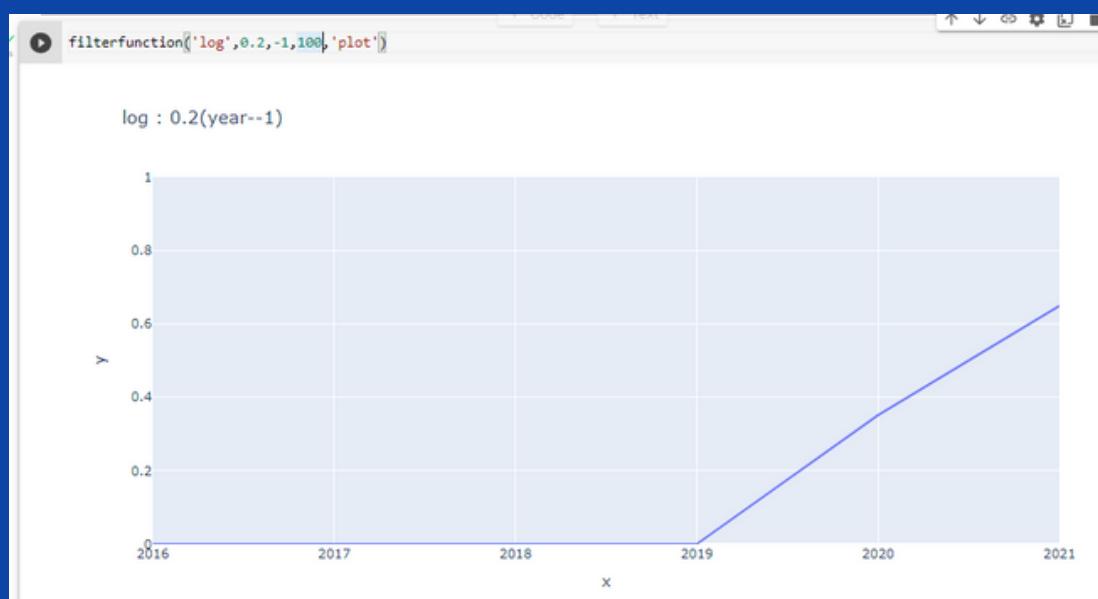
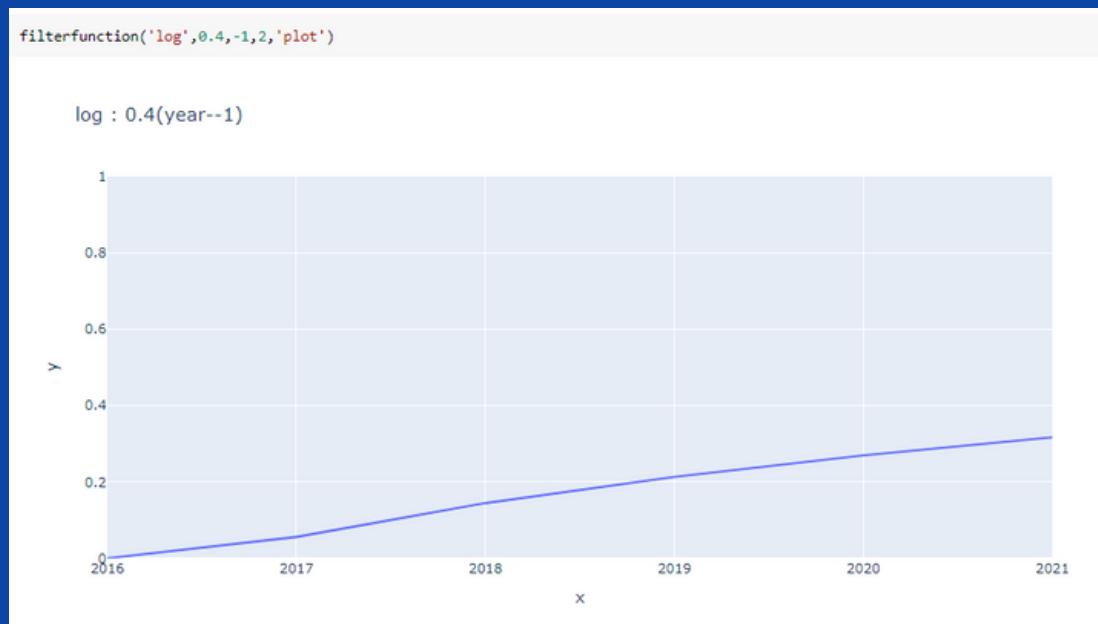
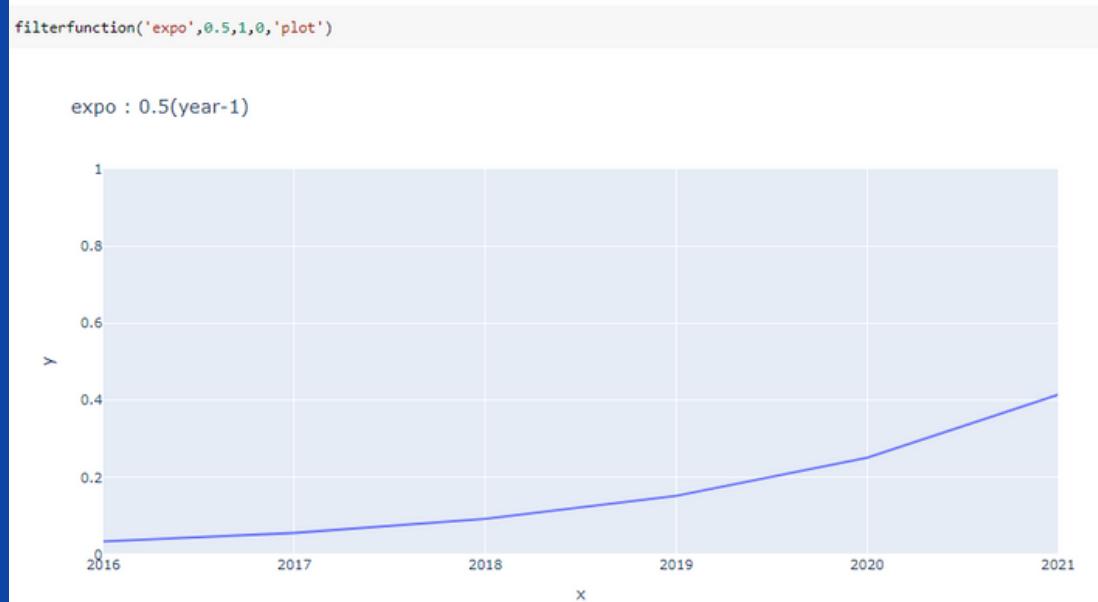
def filterfunction(type,m,k,base,plot):
    import numpy as np
    import math
    import pandas as pd
    import plotly.express as px

    if type == 'expo':
        xd = [i for i in range(1,7)]
        yx = [math.exp(m*(i-k)) for i in xd]
        sumy = sum(yx)
        y = [i/sumy for i in yx]
    elif type == 'log':
        xd = [i-k for i in range(1,7)]
        yx = []
        for i in xd:
            if m*i<1:
                yx.append(math.log(1,base))
            else:
                yx.append(math.log(m*i,base))
        sumy = sum(yx)
        y = [i/sumy for i in yx]
    z = [i+2015 for i in range(1,7)]
    df = pd.DataFrame(dict(
        x = xd,
        y = y,
        z = z
    ))
    if plot.lower() == 'plot':
        fig = px.line(df, x="x", y="y", title=type+" : "+str(m)+"(year-"+str(k)+")")

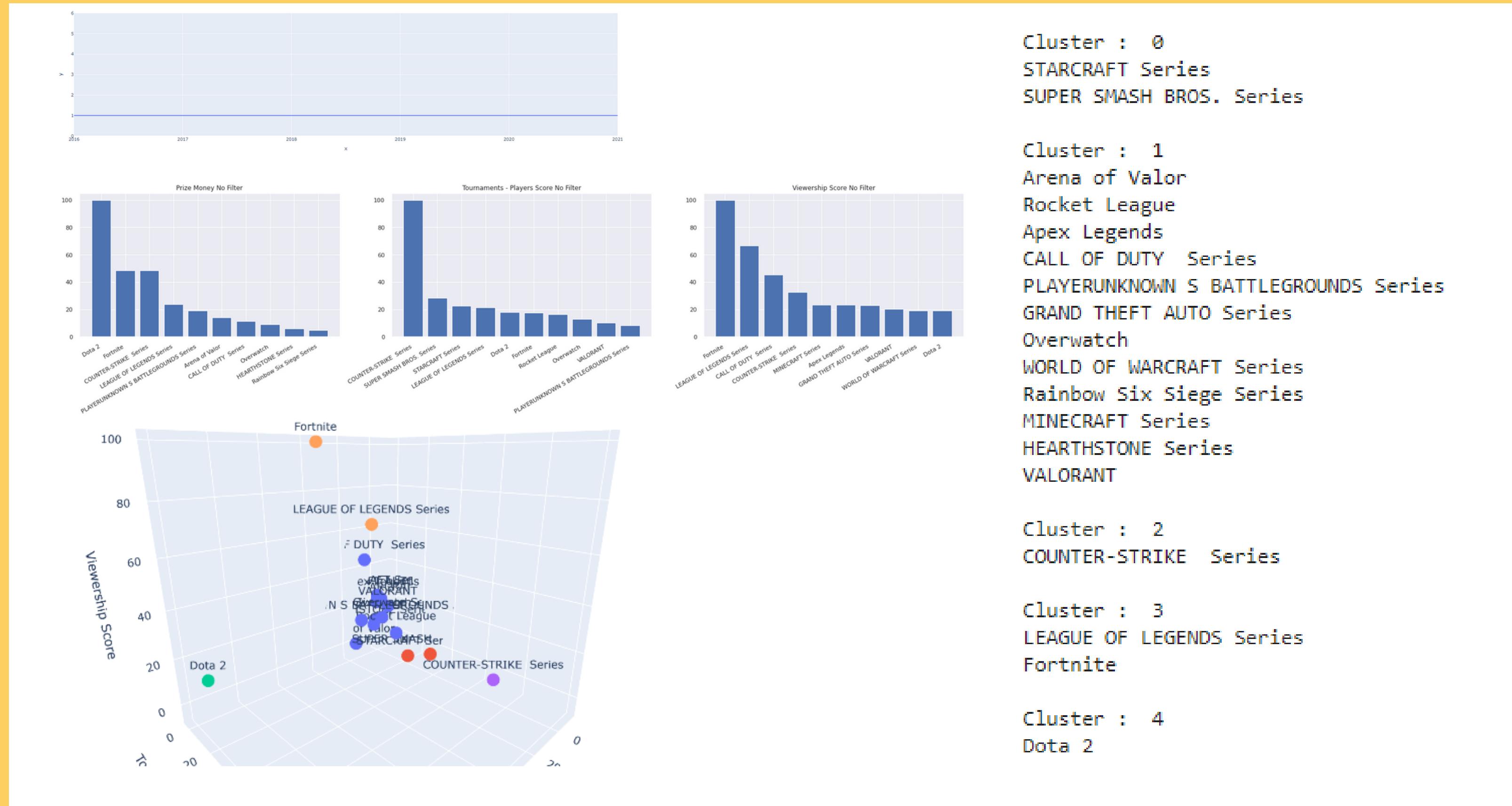
        #fig.update_yaxes(range = [0,5])
        fig.update_layout(
            xaxis = dict(
                tickmode = 'array',
                tickvals = df['x'],
                ticktext = df['z']
            )
        )
        fig.update_yaxes(range=list([0,1]))
        fig.show(render = 'colab')

    return {z[i]:y[i] for i in range(len(y))}


```



NO FILTER



EXPO : e^(2X)

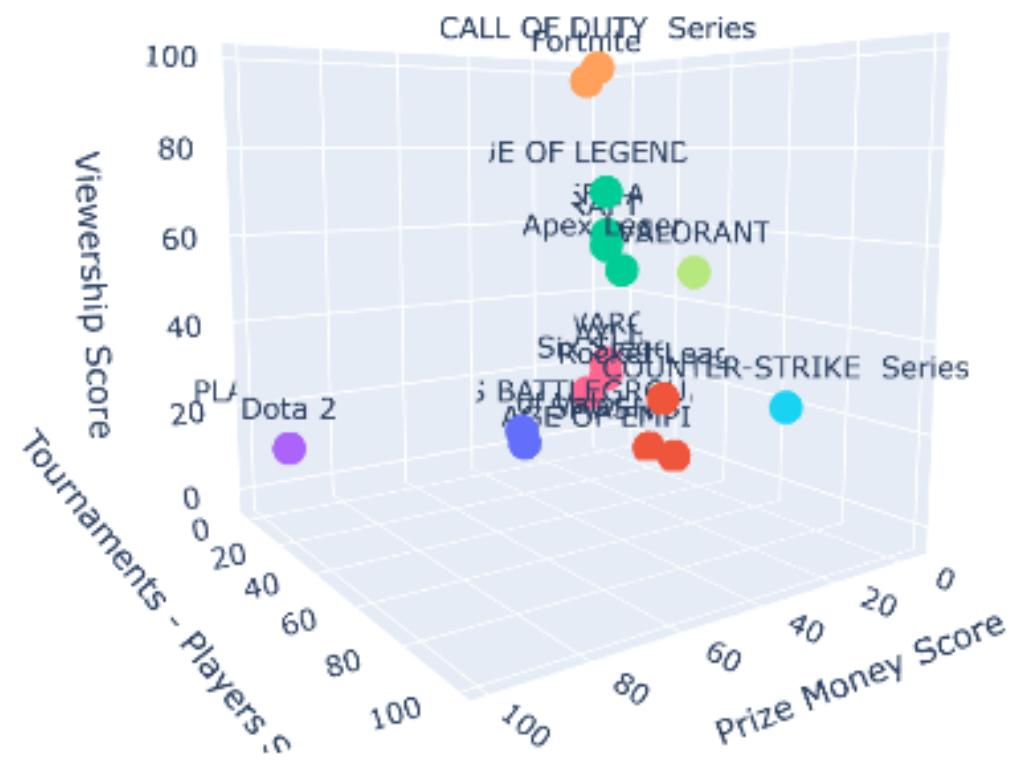
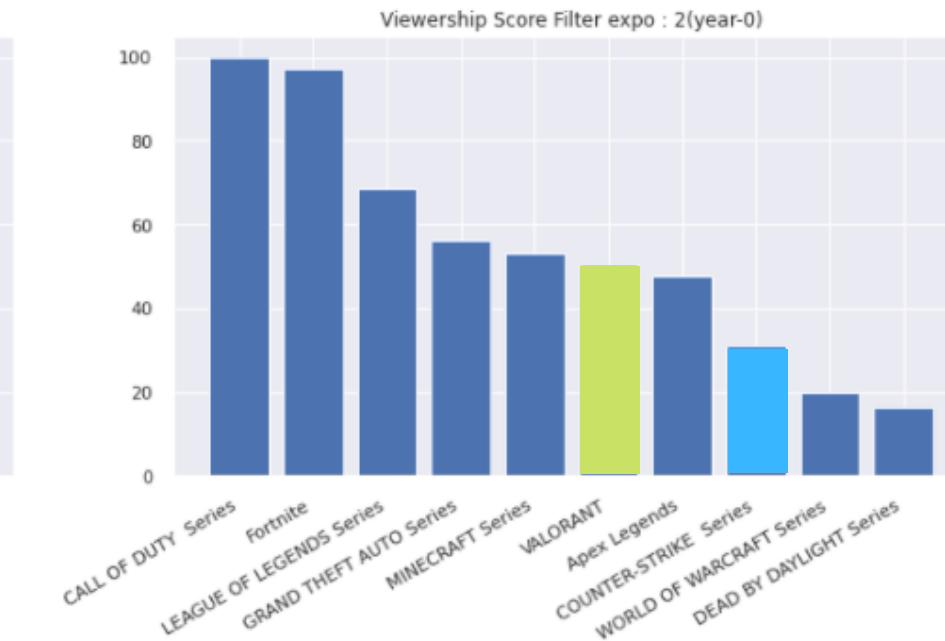
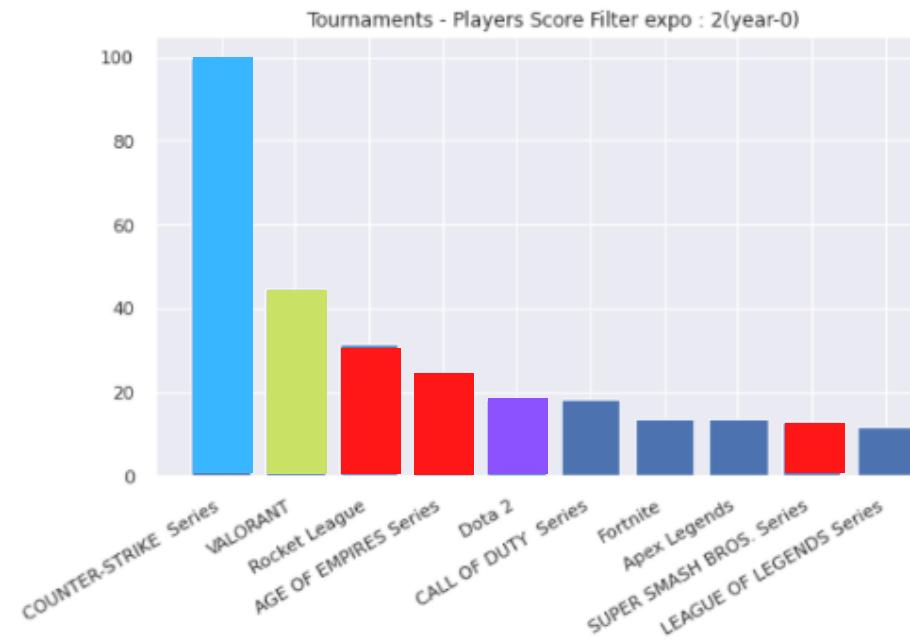
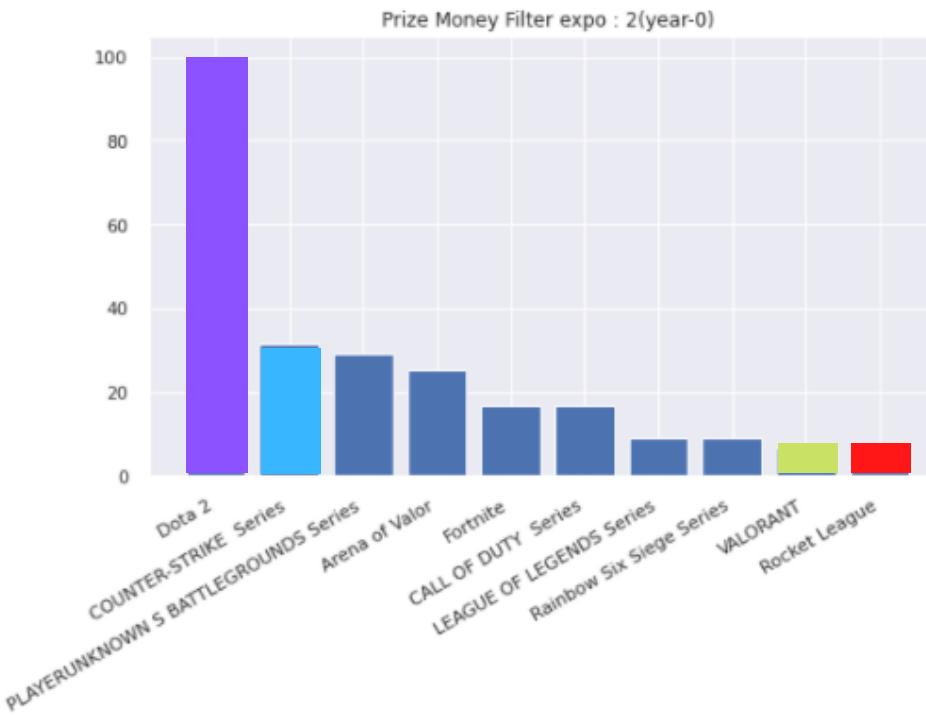


LOG : log100(0.2(X-1))



INSIGHT AND ACTION PLANS

HIGH TOUR-PLAYER, PRIZE CLUSTERS



- Cluster : 0
Rocket League
SUPER SMASH BROS. Series
AGE OF EMPIRES Series
 - Cluster : 3
Dota 2
 - Cluster : 4
COUNTER-STRIKE Series
 - Cluster : 6
VALORANT

EXPO : e^(2X)

MiTH เปิดตัว Line-up ทีม Valorant แบบยกชุดที่นำทัพโดยอดีต IGL ฝีมือดีของวงการ CS:GO อย่าง TOR พร้อมผู้เล่นตัวท็อปอีกเพียบ !!

ไลน์อัพนี้บอกเลยว่าเข้ม !!

2021-04-09 22:32 By 9hos7y

หลังจากเมื่อไม่กี่วันก่อนทาง MiTH ได้ประกาศปล่อยผู้เล่นภายในสังกัดแบบยกชุด ไปเมื่อไม่นาน ซึ่งพวกเค้าได้ออกมาประกาศว่าได้มีทีมที่เตรียมจะตามเท้าเข้าสู่สังกัดเป็นที่เรียบร้อยแล้ว และล่าสุดพวกเค้าก็ได้ประกาศเปิดตัวผู้เล่นยกทีมแบบเป็นทางการที่นำทัพโดย Tor อดีต IGL ตัวท็อปของวงการ CS:GO



รู้จักกับทีม OG จากทีมแห่งมิตรภาพสู่ราชาแห่งโลก DOTA2

ลงวันที่ 04/09/2019



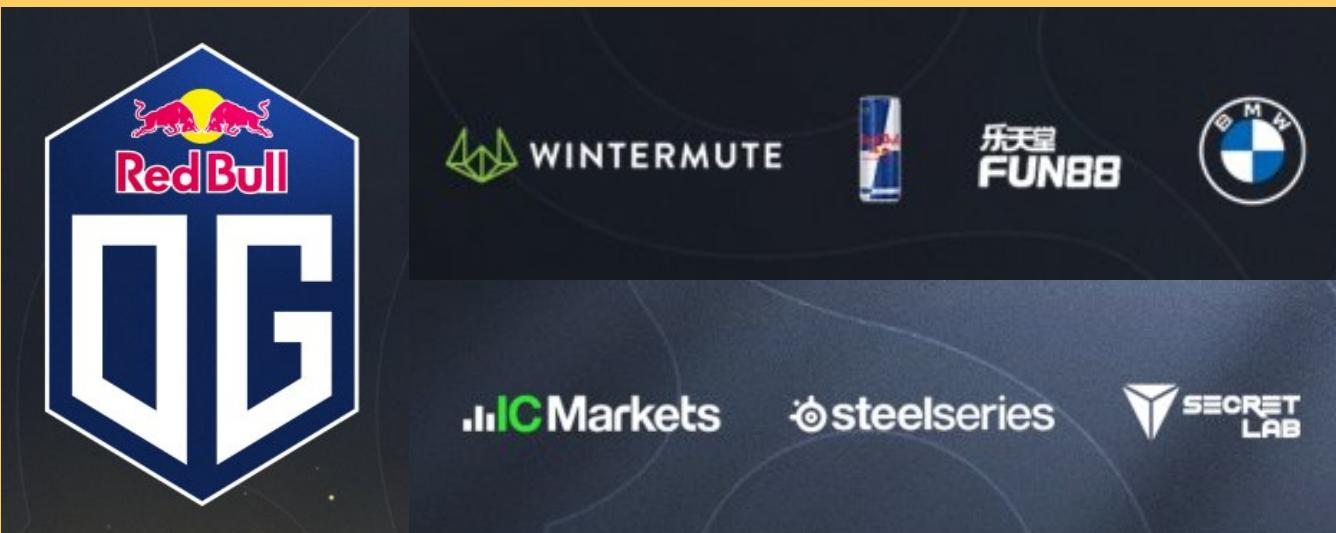
Dota 2 [แก้]

| ID | ชื่อ | สัญชาติ |
|--------|---------------------|----------|
| Sumail | Sumail | ปากีสถาน |
| Topson | Topias Taavitsainen | ฟินแลนด์ |
| Ceb | Sébastien Debs | ฝรั่งเศส |
| Saksa | | ฟินแลนด์ |
| N0tail | Johan Sundstein | เดนมาร์ก |

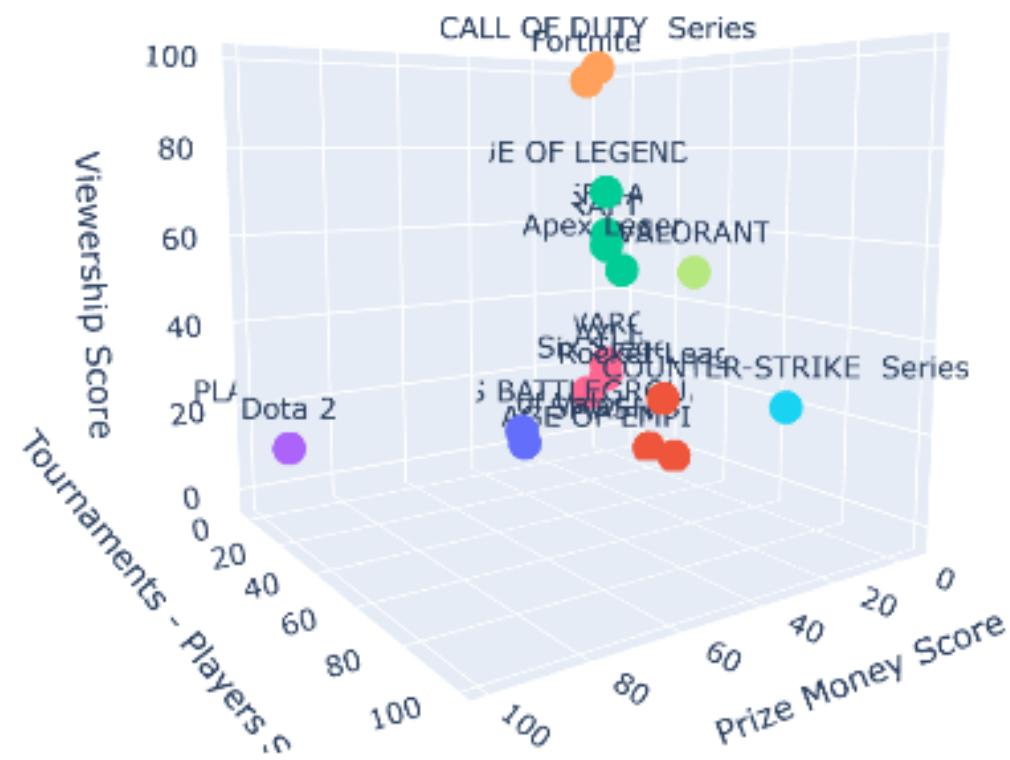
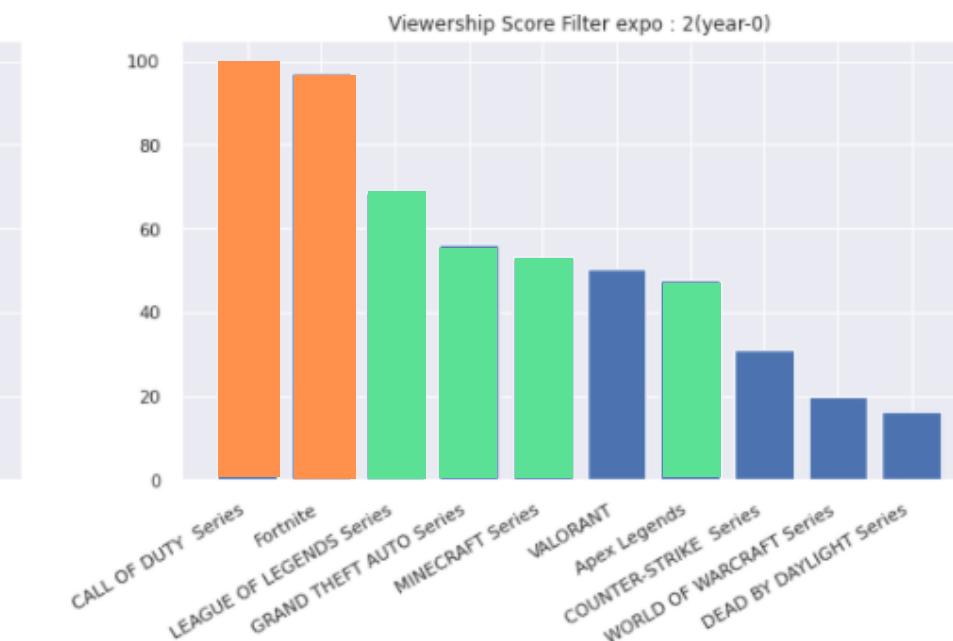
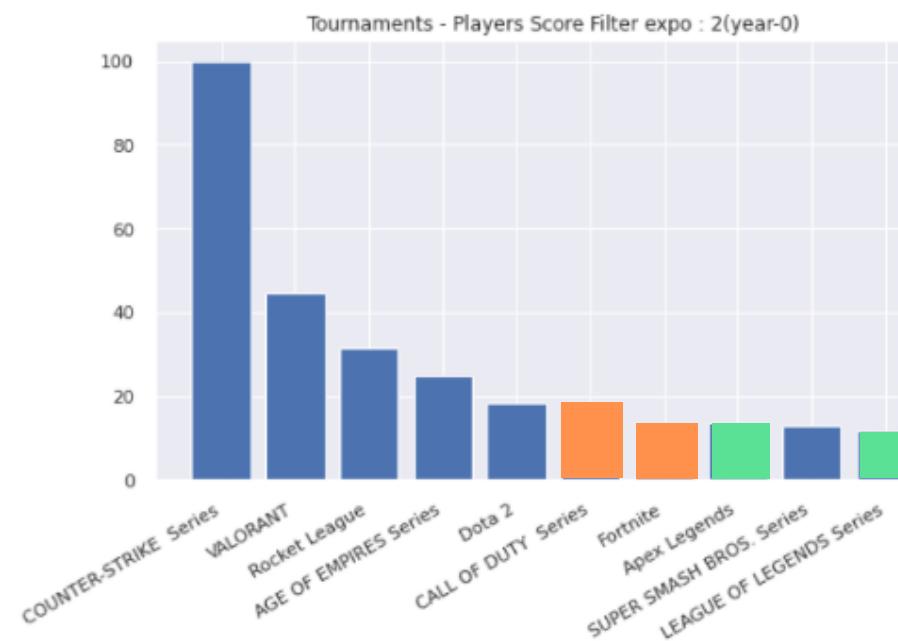
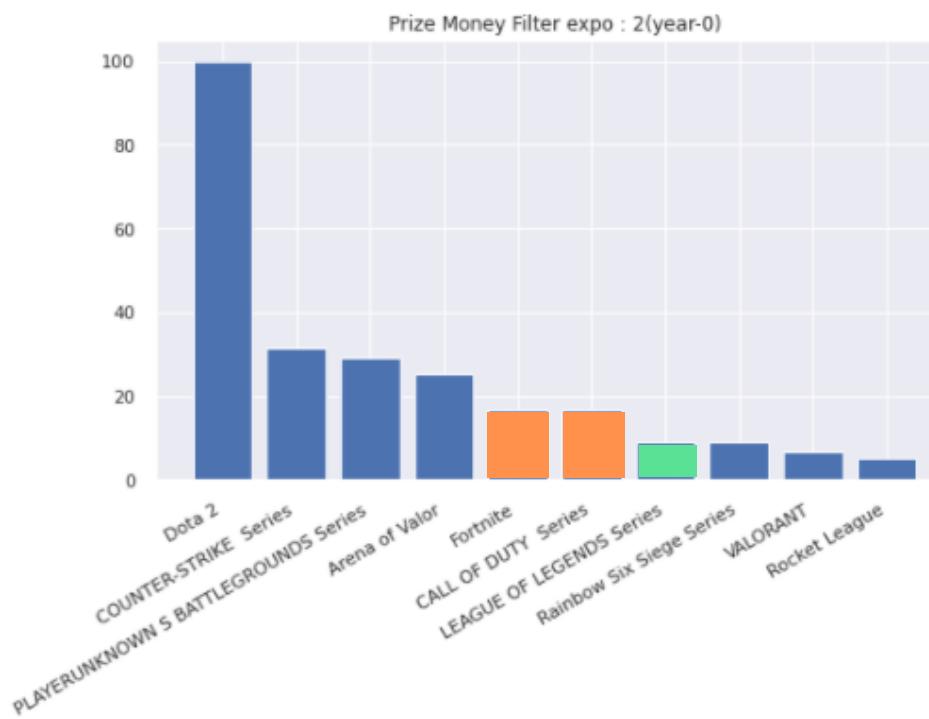
Counter-Strike: Global Offensive [แก้]

| ID | ชื่อ | สัญชาติ |
|---------|--------------------|----------|
| NBK- | Nathan Schmitt | ฝรั่งเศส |
| valde | Valdemar Bjørn | เดนมาร์ก |
| ISSAA | Issa Murad | จอร์แดน |
| mantuu | Mateusz Wilczewski | โปแลนด์ |
| Aleksib | Aleksi Virolainen | ฟินแลนด์ |





HIGH VIEWERSHIP CLUSTERS

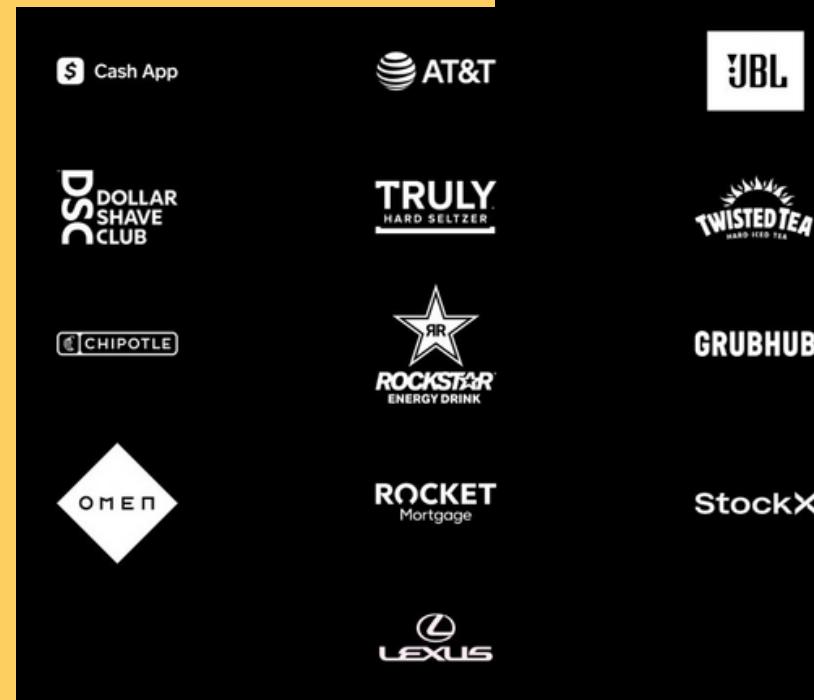


Cluster : 1
CALL OF DUTY Series
Fortnite

Cluster : 5
Apex Legends
GRAND THEFT AUTO Series
LEAGUE OF LEGENDS Series
MINECRAFT Series

EXPO : e^(2X)





ごせいちよう ありがとうございました。

THANK YOU FOR YOUR LISTENING.

終わり
THE END