

Hypothetical scenario answer-

- 1) Look for trends.
- 2) Data collection Strategies.
- 3) Check Data Relevance.
- 4) Have an iterative approach.
- 5) Communicate with Domain Experts.

Variables such as 'APM', 'TotalMapExplored', 'WorkersMade', 'NumberOfPACs', 'ComplexAbilitiesUsed', 'SelectByHotkeys', 'AssignToHotkeys'

Regarding the specific variables mentioned ('APM', 'TotalMapExplored', 'WorkersMade', 'NumberOfPACs', 'ComplexAbilitiesUsed', 'SelectByHotkeys', 'AssignToHotkeys'), consider whether these variables are capturing the desired information and if there are any additional variables that could complement them. Assess if there are any data gaps or limitations associated with these variables and determine if collecting more data can address those gaps or enhance the modeling process.

Importing libraries

In [406...]

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import rankdata
```

In [310...]

```
df = pd.read_csv("C:/Users/Yash/Desktop/starcraft_player_data.csv")
data = df.copy()
df.columns
```

Out[310]:

```
Index(['GameID', 'LeagueIndex', 'Age', 'HoursPerWeek', 'TotalHours', 'APM',
       'SelectByHotkeys', 'AssignToHotkeys', 'UniqueHotkeys', 'MinimapAttacks',
       'MinimapRightClicks', 'NumberOfPACs', 'GapBetweenPACs', 'ActionLatency',
       'ActionsInPAC', 'TotalMapExplored', 'WorkersMade', 'UniqueUnitsMade',
       'ComplexUnitsMade', 'ComplexAbilitiesUsed'],
      dtype='object')
```

In [311...]

```
df.head(5)
```

Out[311]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	AssignToH
0	52	5	27	10	3000	143.7180	0.003515	0.0
1	55	5	23	10	5000	129.2322	0.003304	0.0
2	56	4	30	10	200	69.9612	0.001101	0.0
3	57	3	19	20	400	107.6016	0.001034	0.0
4	58	3	32	10	500	122.8908	0.001136	0.0

In [312...]

```
data.head(5)
```

Out[312]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	AssignToH
0	52	5	27	10	3000	143.7180	0.003515	0.0
1	55	5	23	10	5000	129.2322	0.003304	0.0
2	56	4	30	10	200	69.9612	0.001101	0.0
3	57	3	19	20	400	107.6016	0.001034	0.0
4	58	3	32	10	500	122.8908	0.001136	0.0

In [313...]

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3395 entries, 0 to 3394
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   GameID          3395 non-null    int64  
 1   LeagueIndex     3395 non-null    int64  
 2   Age              3395 non-null    object  
 3   HoursPerWeek    3395 non-null    object  
 4   TotalHours       3395 non-null    object  
 5   APM              3395 non-null    float64 
 6   SelectByHotkeys 3395 non-null    float64 
 7   AssignToHotkeys  3395 non-null    float64 
 8   UniqueHotkeys   3395 non-null    int64  
 9   MinimapAttacks  3395 non-null    float64 
 10  MinimapRightClicks 3395 non-null    float64 
 11  NumberOfPACs    3395 non-null    float64 
 12  GapBetweenPACs  3395 non-null    float64 
 13  ActionLatency   3395 non-null    float64 
 14  ActionsInPAC    3395 non-null    float64 
 15  TotalMapExplored 3395 non-null    int64  
 16  WorkersMade      3395 non-null    float64 
 17  UniqueUnitsMade  3395 non-null    int64  
 18  ComplexUnitsMade 3395 non-null    float64 
 19  ComplexAbilitiesUsed 3395 non-null    float64 
dtypes: float64(12), int64(5), object(3)
memory usage: 530.6+ KB
```

In [314...]: `data.isnull().sum()`

Out[314]:

```
GameID          0
LeagueIndex     0
Age             0
HoursPerWeek    0
TotalHours      0
APM             0
SelectByHotkeys 0
AssignToHotkeys  0
UniqueHotkeys   0
MinimapAttacks  0
MinimapRightClicks 0
NumberOfPACs    0
GapBetweenPACs  0
ActionLatency   0
ActionsInPAC    0
TotalMapExplored 0
WorkersMade      0
UniqueUnitsMade  0
ComplexUnitsMade 0
ComplexAbilitiesUsed 0
dtype: int64
```

In [315...]: `data.isna().sum()`

```
Out[315]: GameID          0
LeagueIndex      0
Age              0
HoursPerWeek    0
TotalHours       0
APM              0
SelectByHotkeys 0
AssignToHotkeys  0
UniqueHotkeys    0
MinimapAttacks  0
MinimapRightClicks 0
NumberOfPACs     0
GapBetweenPACs   0
ActionLatency    0
ActionsInPAC     0
TotalMapExplored 0
WorkersMade       0
UniqueUnitsMade  0
ComplexUnitsMade 0
ComplexAbilitiesUsed 0
dtype: int64
```

Checking how many entries has '?' value

```
In [318... question_mark_rows = data[(data['Age'] == '?') | (data['HoursPerWeek'] == '?') | (data['TotalHours'] == '?')]
```

```
In [330... print(len(question_mark_rows))
```

```
57
```

```
In [321... inval_dtypecol = ["Age", "HoursPerWeek", "TotalHours"]
data[inval_dtypecol] = data[inval_dtypecol].apply(lambda x: x.replace("?", np.nan))
for col in inval_dtypecol:
    data[col] = pd.to_numeric(data[col])
```

```
In [322... data.tail(5)
```

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	Assign
3390	10089	8	NaN	NaN	NaN	259.6296	0.020425	
3391	10090	8	NaN	NaN	NaN	314.6700	0.028043	
3392	10092	8	NaN	NaN	NaN	299.4282	0.028341	
3393	10094	8	NaN	NaN	NaN	375.8664	0.036436	
3394	10095	8	NaN	NaN	NaN	348.3576	0.029855	

Selecting Columns

```
In [324... selectedColumns = ['GameID', 'LeagueIndex', 'Age', 'HoursPerWeek', 'APM', 'MinimapAttacks', 'UniqueHotkeys']
data = data[selectedColumns].dropna()
data.describe()
```

Out[324]:

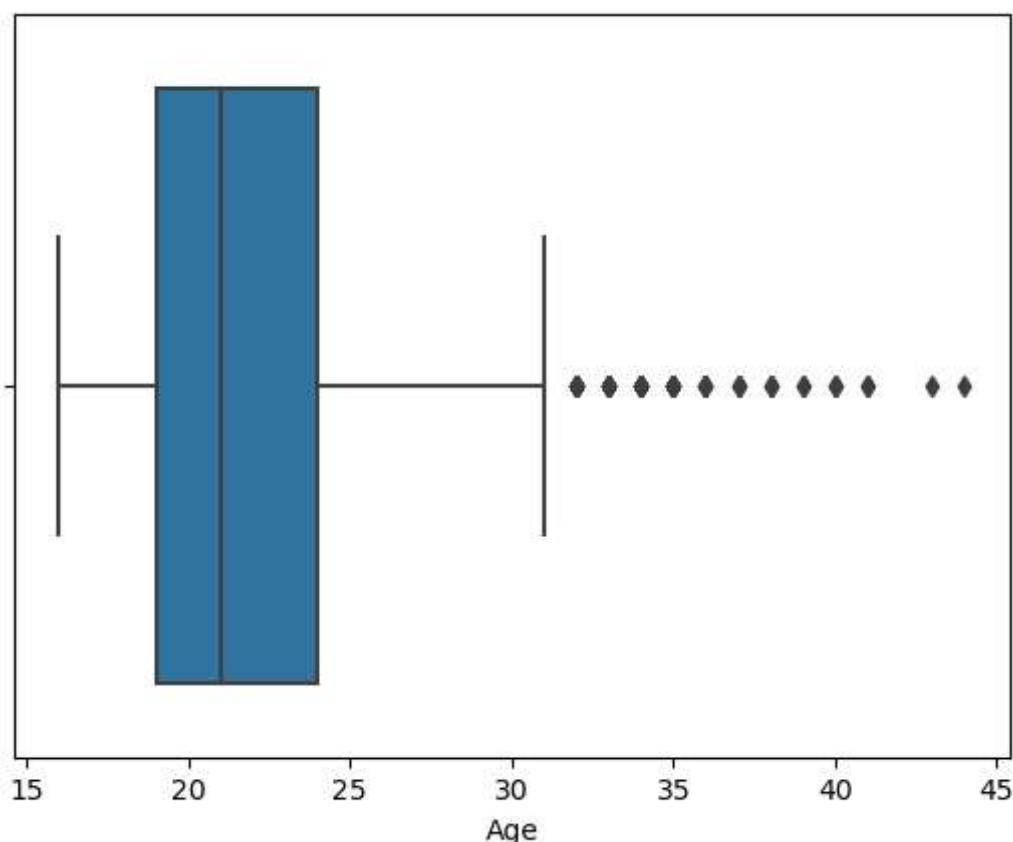
	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	Nun
count	3339.000000	3339.000000	3339.000000	3339.000000	3339.000000	3339.000000	3339.000000
mean	4718.457323	4.120994	21.648997	15.910752	114.569743	0.000094	0.000094
std	2657.274775	1.448033	4.206497	11.962912	48.105963	0.000159	0.000159
min	52.000000	1.000000	16.000000	0.000000	22.059600	0.000000	0.000000
25%	2422.000000	3.000000	19.000000	8.000000	79.233600	0.000000	0.000000
50%	4787.000000	4.000000	21.000000	12.000000	107.067600	0.000039	0.000039
75%	6994.500000	5.000000	24.000000	20.000000	140.134200	0.000113	0.000113
max	9271.000000	7.000000	44.000000	168.000000	389.831400	0.003019	0.003019



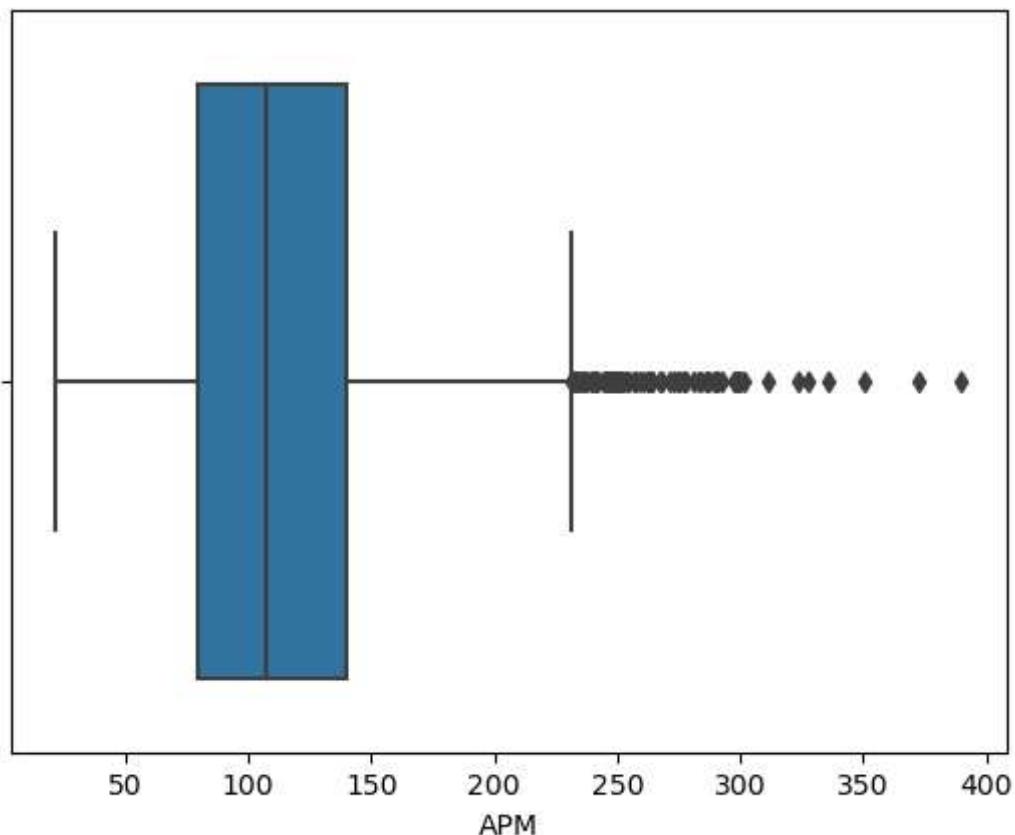
Checking Outliers

In [325...]: `sns.boxplot(x=data['Age'])`

Out[325]: <AxesSubplot:xlabel='Age'>

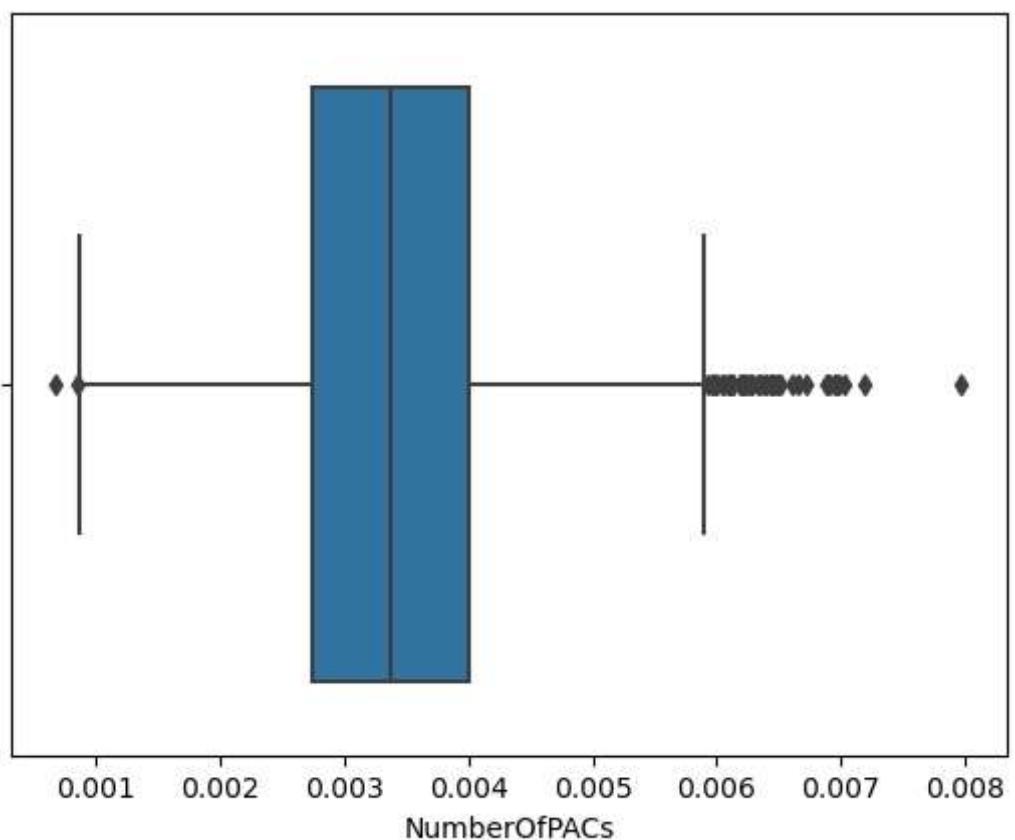
In [326...]: `sns.boxplot(x=data['APM'])`

Out[326]: <AxesSubplot:xlabel='APM'>



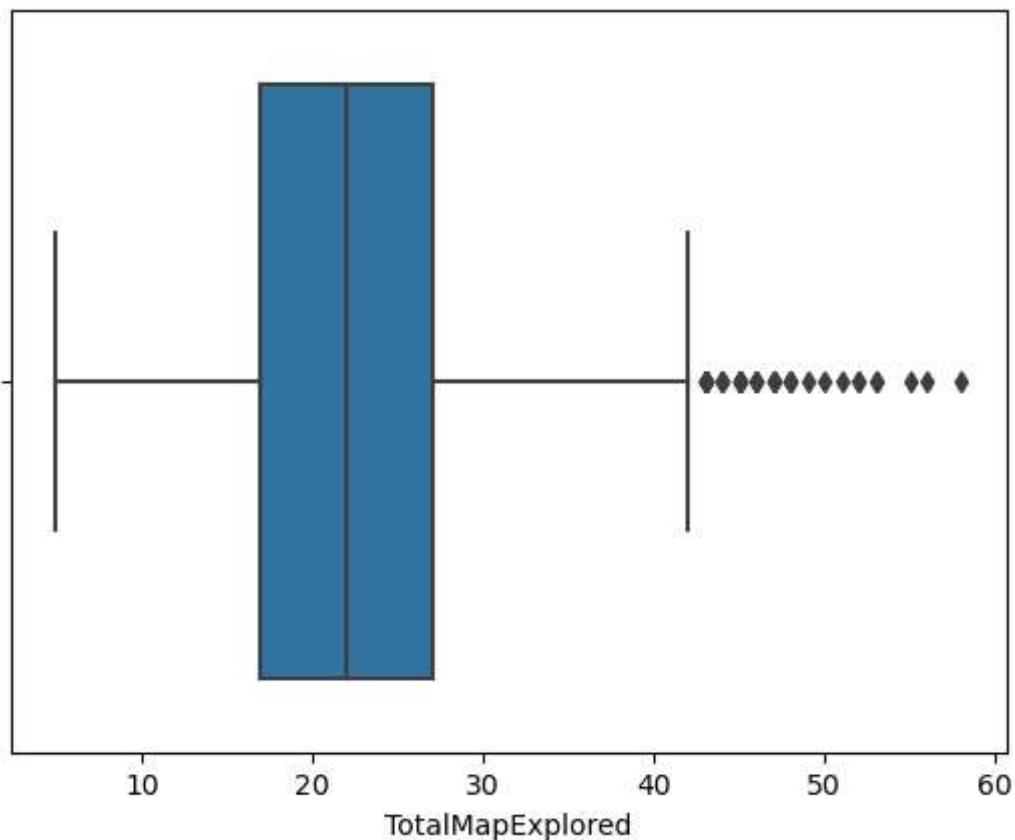
```
In [327]: sns.boxplot(x=data['NumberOfPACs'])
```

```
Out[327]: <AxesSubplot:xlabel='NumberOfPACs'>
```



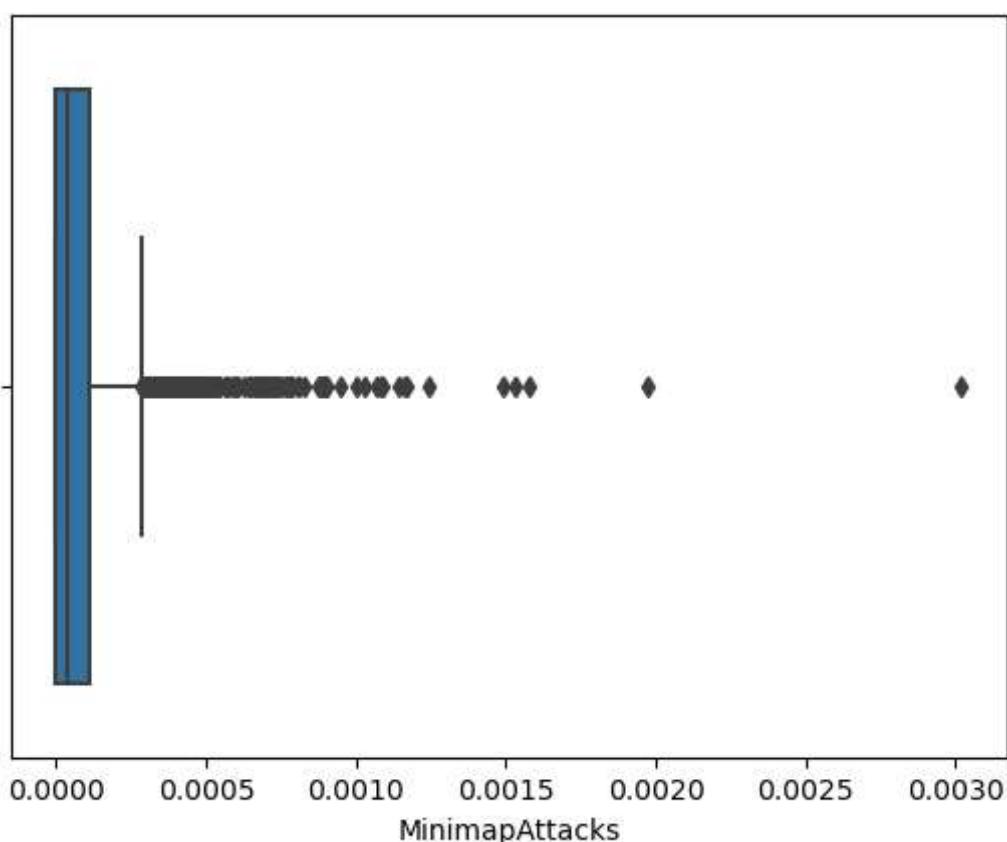
```
In [328]: sns.boxplot(x=data['TotalMapExplored'])
```

```
Out[328]: <AxesSubplot:xlabel='TotalMapExplored'>
```



```
In [329]: sns.boxplot(x=data['MinimapAttacks'])
```

```
Out[329]: <AxesSubplot:xlabel='MinimapAttacks'>
```



```
In [331]: data.describe()
```

Out[331]:

	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	Nun
count	3339.000000	3339.000000	3339.000000	3339.000000	3339.000000	3339.000000	3339.000000
mean	4718.457323	4.120994	21.648997	15.910752	114.569743	0.000094	0.000094
std	2657.274775	1.448033	4.206497	11.962912	48.105963	0.000159	0.000159
min	52.000000	1.000000	16.000000	0.000000	22.059600	0.000000	0.000000
25%	2422.000000	3.000000	19.000000	8.000000	79.233600	0.000000	0.000000
50%	4787.000000	4.000000	21.000000	12.000000	107.067600	0.000039	0.000039
75%	6994.500000	5.000000	24.000000	20.000000	140.134200	0.000113	0.000113
max	9271.000000	7.000000	44.000000	168.000000	389.831400	0.003019	0.003019

In [332...]

data['Age'].value_counts()

Out[332]:

20.0	357
21.0	344
18.0	324
22.0	314
19.0	313
23.0	259
16.0	256
17.0	248
24.0	225
25.0	168
26.0	136
27.0	111
28.0	73
29.0	52
30.0	32
31.0	29
32.0	21
35.0	17
33.0	15
34.0	15
36.0	8
38.0	5
37.0	5
40.0	4
41.0	3
39.0	3
43.0	1
44.0	1

Name: Age, dtype: int64

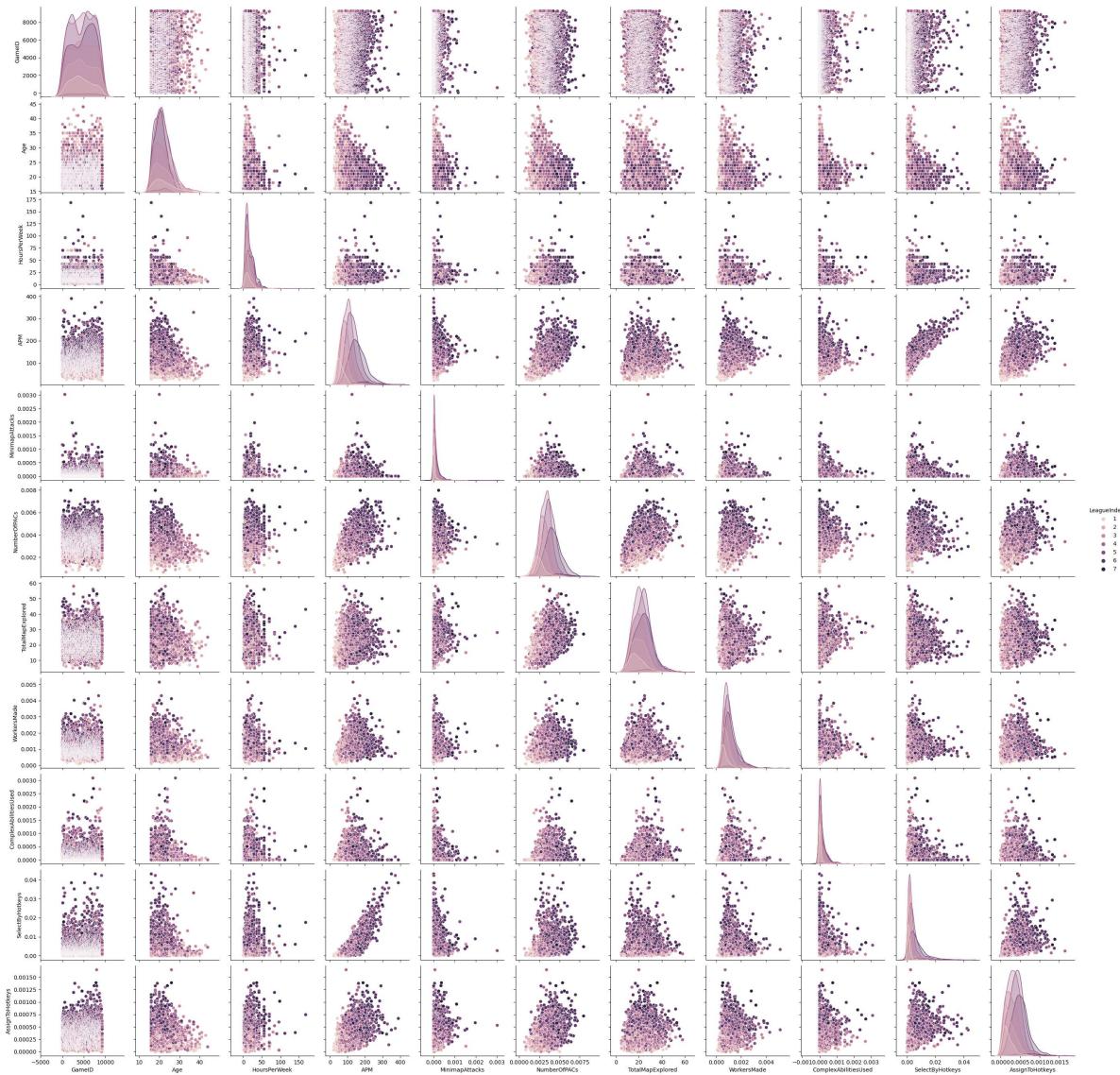
Checking relation between variables

In [333...]

sns.pairplot(data, hue='LeagueIndex')

Out[333]:

<seaborn.axisgrid.PairGrid at 0x1b2b84de850>



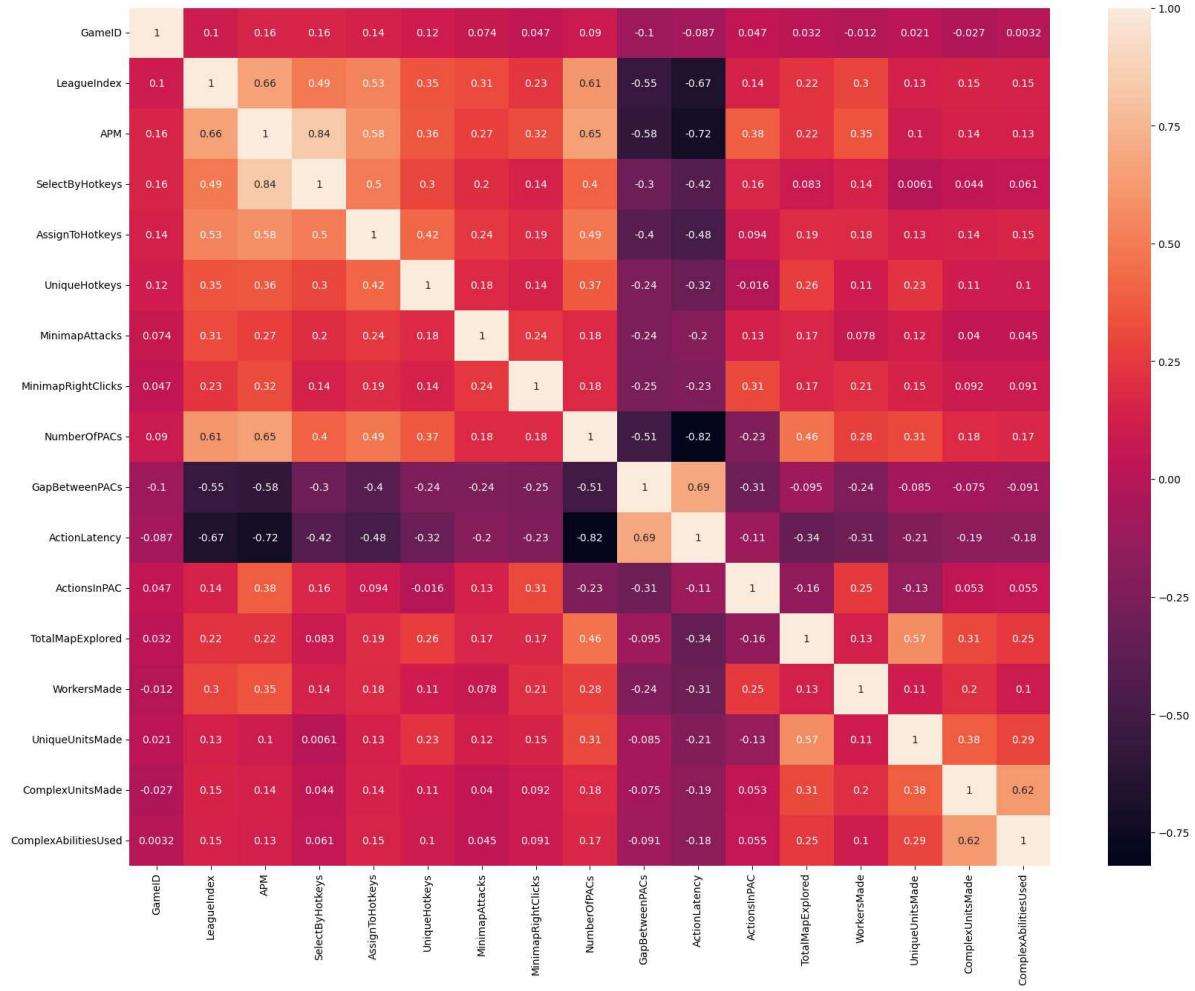
In [334]: `data.tail()`

	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	NumberOfPACs	TotalMapsExplored	WintersMade	ComplexityAttained	SelectedByHotkeys	AssignHotkeys
3335	9261	4	20.0	8.0	158.1390	0.000217	0.003583					
3336	9264	5	16.0	56.0	186.1320	0.000083	0.005414					
3337	9265	4	21.0	8.0	121.6992	0.000055	0.003690					
3338	9270	3	20.0	28.0	134.2848	0.000000	0.003205					
3339	9271	4	22.0	6.0	88.8246	0.000000	0.003099					

In [335]: `plt.figure(figsize=(20, 15))
corr = df.corr()
sns.heatmap(corr, annot=True)`

Out[335]: <AxesSubplot:>

Evil_Geniuses_Submit



In [337...]

```
rows_with_league_8 = data[data['LeagueIndex'] == 8]
print(rows_with_league_8)
```

Empty DataFrame

Columns: [GameID, LeagueIndex, Age, HoursPerWeek, APM, MinimapAttacks, NumberOfPACs, TotalMapExplored, WorkersMade, ComplexAbilitiesUsed, SelectByHotkeys, AssignToHotkeys, UniqueHotkeys, MinimapRightClicks, NumberOfPACs, GapBetweenPACs, ActionLatency, ActionsInPAC, TotalMapExplored, WorkersMade, UniqueUnitsMade, ComplexUnitsMade]

Index: []

In [338...]

```
rows_with_league_8 = df[df['LeagueIndex'] == 8]
print(rows_with_league_8)
```

GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	\
3340	10001	8	?	?	?	189.7404
3341	10005	8	?	?	?	287.8128
3342	10006	8	?	?	?	294.0996
3343	10015	8	?	?	?	274.2552
3344	10016	8	?	?	?	274.3404
3345	10017	8	?	?	?	245.8188
3346	10018	8	?	?	?	211.0722
3347	10021	8	?	?	?	189.5778
3348	10022	8	?	?	?	210.5088
3349	10023	8	?	?	?	248.0118
3350	10024	8	?	?	?	299.2290
3351	10025	8	?	?	?	179.9982
3352	10026	8	?	?	?	340.1982
3353	10028	8	?	?	?	319.7148
3354	10029	8	?	?	?	290.5914
3355	10030	8	?	?	?	275.8632
3356	10035	8	?	?	?	298.7916
3357	10036	8	?	?	?	325.1154
3358	10038	8	?	?	?	146.3892
3359	10039	8	?	?	?	192.4554
3360	10041	8	?	?	?	315.6936
3361	10045	8	?	?	?	203.7726
3362	10046	8	?	?	?	334.5240
3363	10047	8	?	?	?	175.5936
3364	10049	8	?	?	?	252.7206
3365	10050	8	?	?	?	211.9188
3366	10051	8	?	?	?	269.8998
3367	10052	8	?	?	?	190.2396
3368	10055	8	?	?	?	212.4972
3369	10059	8	?	?	?	219.3894
3370	10060	8	?	?	?	230.6694
3371	10061	8	?	?	?	284.2296
3372	10062	8	?	?	?	355.3518
3373	10063	8	?	?	?	364.8504
3374	10064	8	?	?	?	256.5888
3375	10065	8	?	?	?	248.4012
3376	10066	8	?	?	?	251.2284
3377	10067	8	?	?	?	318.3000
3378	10068	8	?	?	?	288.9198
3379	10069	8	?	?	?	313.9080
3380	10072	8	?	?	?	243.7134
3381	10073	8	?	?	?	312.9804
3382	10074	8	?	?	?	313.5762
3383	10075	8	?	?	?	274.6194
3384	10076	8	?	?	?	225.0678
3385	10079	8	?	?	?	254.2188
3386	10081	8	?	?	?	339.1524
3387	10082	8	?	?	?	310.0416
3388	10083	8	?	?	?	288.7608
3389	10084	8	?	?	?	151.4046
3390	10089	8	?	?	?	259.6296
3391	10090	8	?	?	?	314.6700
3392	10092	8	?	?	?	299.4282
3393	10094	8	?	?	?	375.8664
3394	10095	8	?	?	?	348.3576

SelectByHotkeys	AssignToHotkeys	UniqueHotkeys	MinimapAttacks	\
3340	0.004582	0.000655	4	0.000073
3341	0.029040	0.001041	9	0.000231
3342	0.029640	0.001076	6	0.000302
3343	0.018121	0.001264	8	0.000053
3344	0.023131	0.000739	8	0.000622
3345	0.010471	0.000841	10	0.000657

3346	0.013049	0.000940	10	0.000366
3347	0.007559	0.000487	10	0.000606
3348	0.007974	0.000867	7	0.000548
3349	0.014722	0.001752	7	0.000375
3350	0.026428	0.000951	10	0.000155
3351	0.009524	0.001052	6	0.000000
3352	0.028214	0.001242	8	0.000519
3353	0.037130	0.000820	5	0.000403
3354	0.027561	0.001750	6	0.000022
3355	0.019502	0.001449	10	0.000306
3356	0.023253	0.000659	4	0.000433
3357	0.029790	0.001338	10	0.000059
3358	0.006701	0.000400	10	0.000883
3359	0.014277	0.000466	4	0.000000
3360	0.028311	0.001160	10	0.001242
3361	0.008337	0.000573	5	0.000614
3362	0.017742	0.001548	6	0.000384
3363	0.012680	0.000934	9	0.000098
3364	0.019097	0.001522	6	0.000384
3365	0.019817	0.000633	4	0.000201
3366	0.024645	0.000642	10	0.000415
3367	0.008720	0.000879	10	0.000171
3368	0.014917	0.000767	10	0.000599
3369	0.005926	0.000741	6	0.000440
3370	0.010383	0.001242	10	0.000375
3371	0.016069	0.000711	9	0.000355
3372	0.037526	0.000600	7	0.001242
3373	0.042576	0.000996	8	0.000176
3374	0.019592	0.000580	8	0.000416
3375	0.016018	0.000874	9	0.000388
3376	0.022910	0.000946	5	0.001097
3377	0.034851	0.000933	7	0.000187
3378	0.029322	0.001569	6	0.000118
3379	0.019537	0.001214	4	0.000318
3380	0.017195	0.000711	6	0.000666
3381	0.026327	0.000266	6	0.000000
3382	0.030550	0.000560	5	0.000000
3383	0.022497	0.000707	6	0.000163
3384	0.014339	0.001627	7	0.000291
3385	0.016608	0.000788	6	0.000926
3386	0.033058	0.001017	10	0.000477
3387	0.026873	0.001278	10	0.000319
3388	0.024022	0.000628	6	0.000350
3389	0.009732	0.000949	6	0.000028
3390	0.020425	0.000743	9	0.000621
3391	0.028043	0.001157	10	0.000246
3392	0.028341	0.000860	7	0.000338
3393	0.036436	0.000594	5	0.000204
3394	0.029855	0.000811	4	0.000224

	MinimapRightClicks	NumberOfPACs	GapBetweenPACs	ActionLatency	\
3340	0.000618	0.006291	23.5130	32.5665	
3341	0.000656	0.005399	31.6416	36.1143	
3342	0.002374	0.006294	16.6393	36.8192	
3343	0.000975	0.007111	10.6419	24.3556	
3344	0.003552	0.005355	19.1568	36.3098	
3345	0.001314	0.005031	14.5518	36.7134	
3346	0.000909	0.003719	19.6169	38.9326	
3347	0.000566	0.005821	22.0317	36.7330	
3348	0.000638	0.006518	15.7856	30.7156	
3349	0.000110	0.004115	17.4656	34.2357	
3350	0.000929	0.005443	17.0835	33.7398	
3351	0.000125	0.003567	32.5628	39.5600	
3352	0.001163	0.006898	15.2852	26.6907	

3353	0.000619	0.005208	35.4127	44.0552
3354	0.001949	0.005293	22.0126	36.0669
3355	0.000386	0.007569	18.1407	24.0936
3356	0.000330	0.005561	16.0743	29.2593
3357	0.000357	0.005381	15.4571	40.3646
3358	0.002384	0.003617	18.4444	47.3364
3359	0.001591	0.003142	29.7500	35.7531
3360	0.000628	0.005076	17.7035	32.6344
3361	0.000757	0.005954	11.3597	31.1615
3362	0.004041	0.007780	13.5401	28.2243
3363	0.001010	0.005265	27.1322	43.7278
3364	0.000569	0.004090	21.6151	38.2256
3365	0.000201	0.003912	31.8222	54.5588
3366	0.000491	0.004015	25.6352	43.3856
3367	0.000342	0.004971	17.9901	35.9509
3368	0.000273	0.005648	21.6687	41.2231
3369	0.000709	0.005185	17.0456	30.5342
3370	0.003328	0.006375	13.5028	31.4044
3371	0.000548	0.006680	9.4756	29.6851
3372	0.000514	0.004541	9.2871	41.9497
3373	0.000146	0.004687	19.9499	41.1417
3374	0.000357	0.005812	17.0462	34.3734
3375	0.000372	0.005987	16.3144	30.2486
3376	0.001173	0.005411	13.7404	35.7203
3377	0.000023	0.005225	26.0987	32.4464
3378	0.000219	0.005213	23.2857	32.8026
3379	0.000607	0.005879	8.1642	26.0918
3380	0.000426	0.005594	21.8795	30.5722
3381	0.000207	0.005053	14.6118	30.7836
3382	0.000206	0.004390	19.5405	35.4094
3383	0.000082	0.004053	20.6757	32.7785
3384	0.000911	0.005281	16.3502	33.2874
3385	0.000330	0.005408	14.9191	35.9921
3386	0.000509	0.004609	21.6389	37.1862
3387	0.000479	0.005517	16.5446	33.8174
3388	0.001051	0.005580	19.0108	30.0866
3389	0.000156	0.004363	27.4658	43.8052
3390	0.000146	0.004555	18.6059	42.8342
3391	0.001083	0.004259	14.3023	36.1156
3392	0.000169	0.004439	12.4028	39.5156
3393	0.000780	0.004346	11.6910	34.8547
3394	0.001315	0.005566	20.0537	33.5142

	ActionsInPAC	TotalMapExplored	WorkersMade	UniqueUnitsMade	\
3340	4.4451	25	0.002218		6
3341	4.5893	34	0.001138		6
3342	4.1850	26	0.000987		6
3343	4.3870	28	0.001106		6
3344	5.2811	28	0.000739		6
3345	7.1943	33	0.001474		11
3346	7.1320	23	0.000898		9
3347	4.9050	28	0.000540		5
3348	4.8058	34	0.000817		6
3349	7.8973	20	0.001111		8
3350	5.2703	16	0.000697		6
3351	7.0050	13	0.000999		6
3352	5.1293	26	0.001535		8
3353	4.4282	26	0.000892		6
3354	4.9540	19	0.000642		6
3355	4.1723	15	0.001031		5
3356	5.8444	19	0.000783		6
3357	5.7652	22	0.000907		7
3358	5.8341	17	0.000950		8
3359	7.1975	11	0.001280		3

Evil_Geniuses_Submit

3360	6.2231	24	0.000791	6
3361	5.1082	23	0.000859	7
3362	5.6862	29	0.002161	9
3363	3.8371	24	0.000575	5
3364	6.8534	23	0.000523	5
3365	5.0294	14	0.001409	3
3366	6.4922	21	0.000478	6
3367	5.5872	21	0.000904	5
3368	4.4680	28	0.001119	9
3369	6.6749	35	0.002072	9
3370	5.0533	32	0.001512	6
3371	5.3326	25	0.002459	7
3372	6.5063	22	0.001228	8
3373	5.6167	18	0.000674	7
3374	5.0563	19	0.001308	7
3375	5.0973	21	0.001197	6
3376	4.5524	22	0.000738	5
3377	4.8705	13	0.000933	4
3378	4.7540	24	0.001130	5
3379	6.7885	26	0.001321	8
3380	5.1136	25	0.000576	8
3381	6.1930	10	0.001802	4
3382	6.4228	12	0.001296	3
3383	6.9262	15	0.000626	3
3384	5.4713	26	0.000898	8
3385	5.7205	28	0.001128	6
3386	6.7103	16	0.001049	3
3387	5.7350	22	0.000922	8
3388	5.3831	30	0.000761	5
3389	4.3312	23	0.000949	6
3390	6.2754	46	0.000877	5
3391	7.1965	16	0.000788	4
3392	6.3979	19	0.001260	4
3393	7.9615	15	0.000613	6
3394	6.3719	27	0.001566	7

	ComplexUnitsMade	ComplexAbilitiesUsed
3340	0.000000	0.000000
3341	0.000058	0.000000
3342	0.000000	0.000000
3343	0.000000	0.000000
3344	0.000000	0.000000
3345	0.000040	0.000048
3346	0.000000	0.000000
3347	0.000000	0.000000
3348	0.000000	0.000000
3349	0.000000	0.000000
3350	0.000033	0.000011
3351	0.000000	0.000000
3352	0.000000	0.000113
3353	0.000245	0.000144
3354	0.000044	0.000078
3355	0.000000	0.000000
3356	0.000000	0.000309
3357	0.000000	0.000000
3358	0.000017	0.000167
3359	0.000000	0.000000
3360	0.000000	0.000150
3361	0.000000	0.000000
3362	0.000145	0.000073
3363	0.000000	0.000000
3364	0.000000	0.000323
3365	0.000000	0.000000
3366	0.000000	0.000579

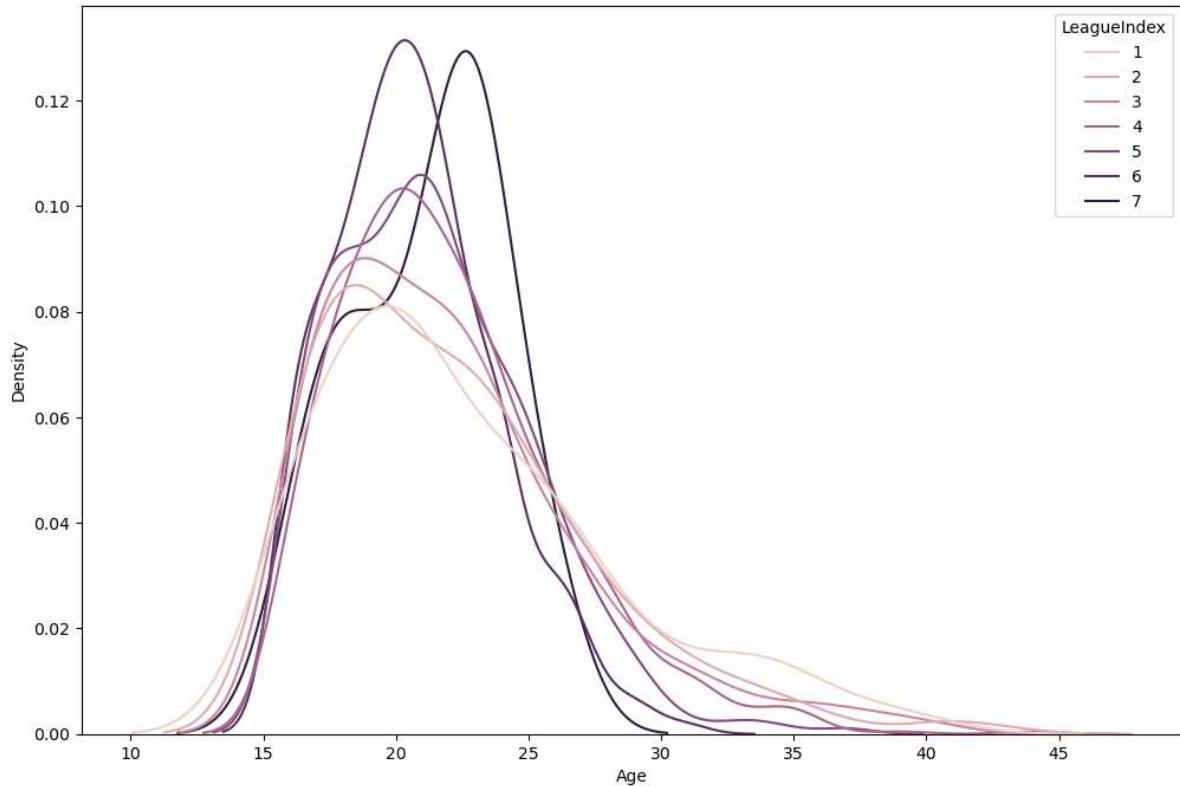
3367	0.000000	0.000000
3368	0.000035	0.000062
3369	0.000225	0.000064
3370	0.000035	0.000047
3371	0.000000	0.000000
3372	0.000000	0.000614
3373	0.000000	0.000000
3374	0.000000	0.000000
3375	0.000000	0.000000
3376	0.000000	0.000662
3377	0.000000	0.000233
3378	0.000000	0.000000
3379	0.000106	0.000443
3380	0.000000	0.000000
3381	0.000000	0.000030
3382	0.000000	0.000059
3383	0.000000	0.000000
3384	0.000000	0.000959
3385	0.000000	0.000000
3386	0.000000	0.000000
3387	0.000000	0.000000
3388	0.000000	0.000652
3389	0.000000	0.000099
3390	0.000000	0.000000
3391	0.000000	0.000000
3392	0.000000	0.000000
3393	0.000000	0.000631
3394	0.000457	0.000895

All the LeagueIndex with value 8 has entries with '?' value

Probability Distribution Plots of variables

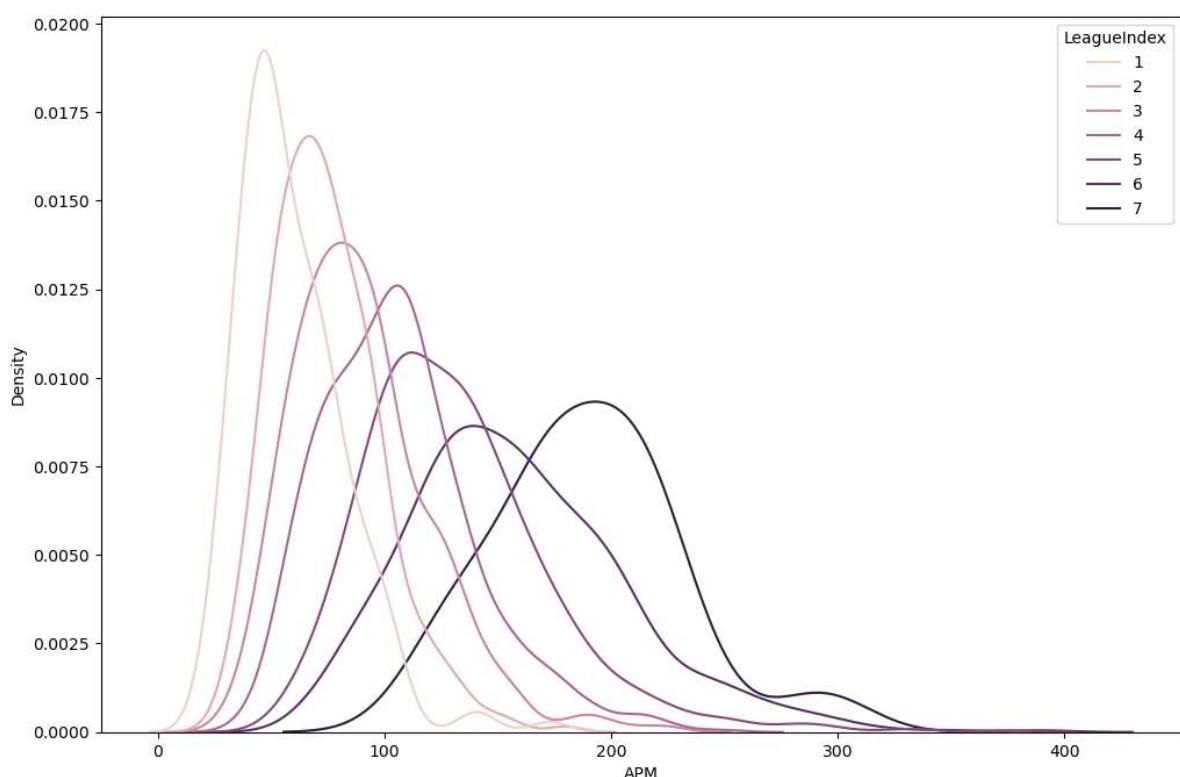
```
In [339...]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="Age", hue = 'LeagueIndex', common_norm = False)

Out[339]: <AxesSubplot:xlabel='Age', ylabel='Density'>
```



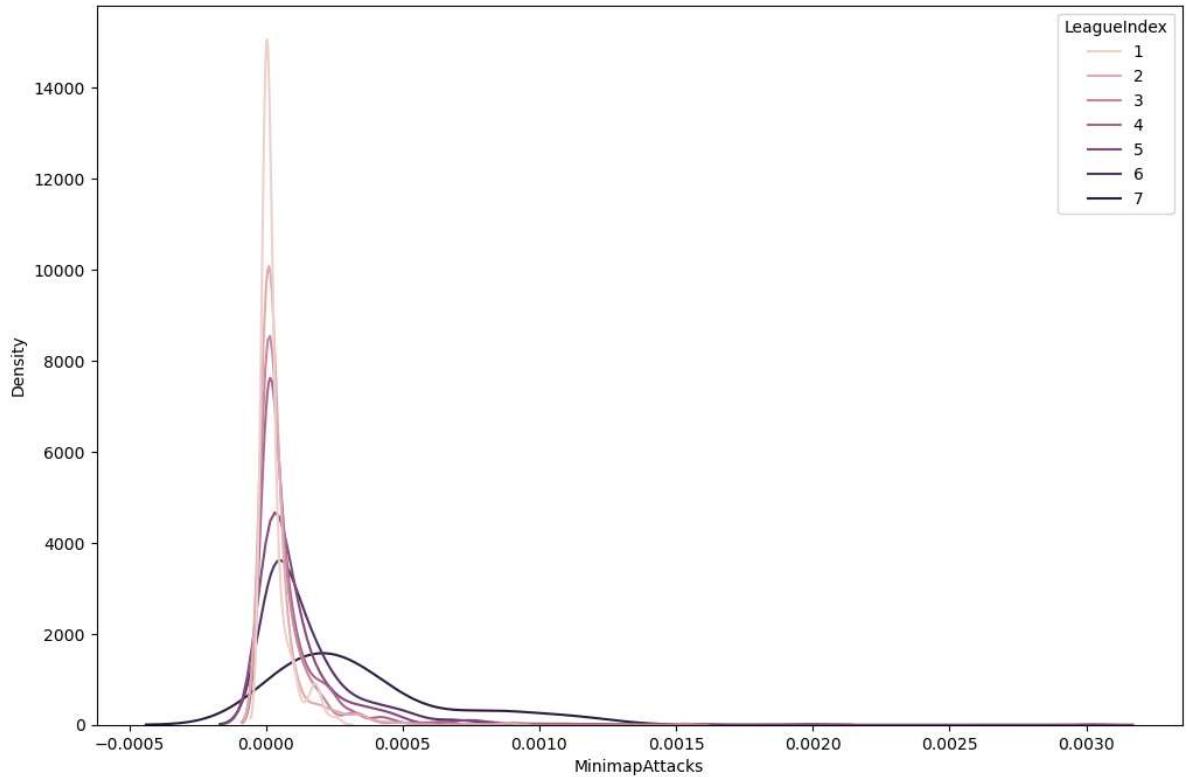
```
In [340]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="APM", hue = 'LeagueIndex', common_norm = False)
```

Out[340]: <AxesSubplot:xlabel='APM', ylabel='Density'>

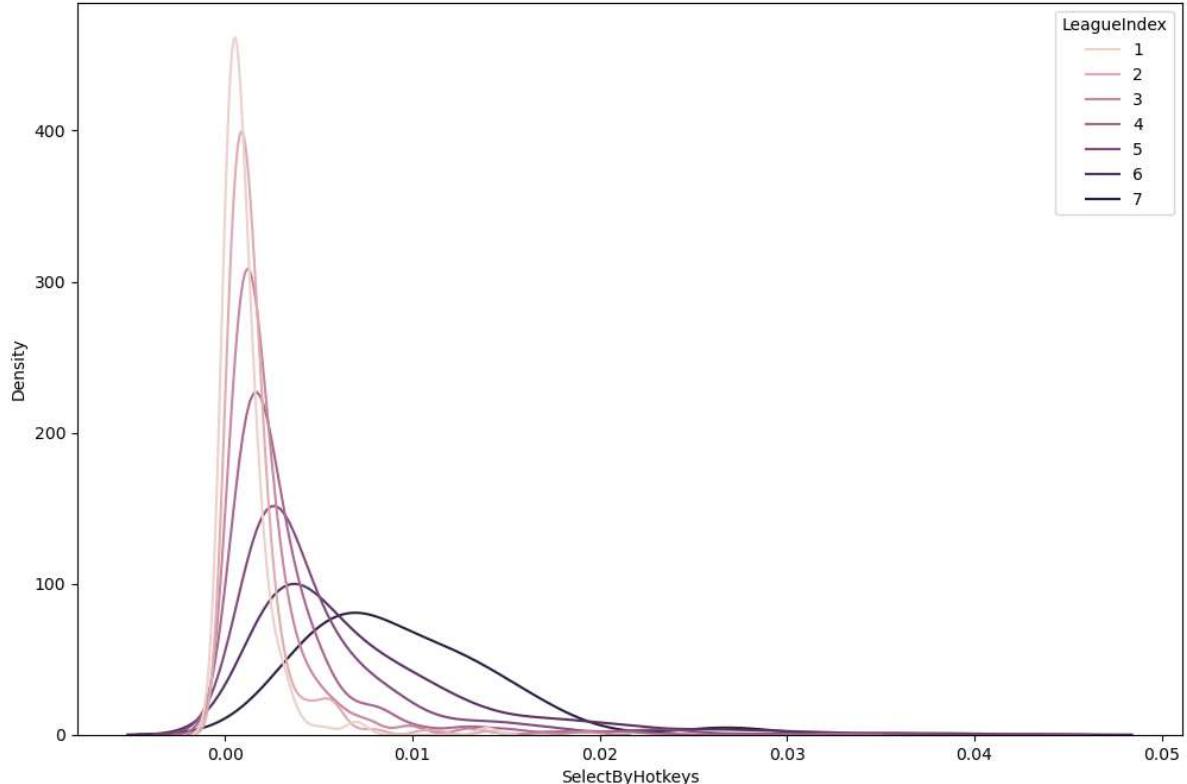


```
In [341]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="MinimapAttacks", hue = 'LeagueIndex', common_norm = False)
```

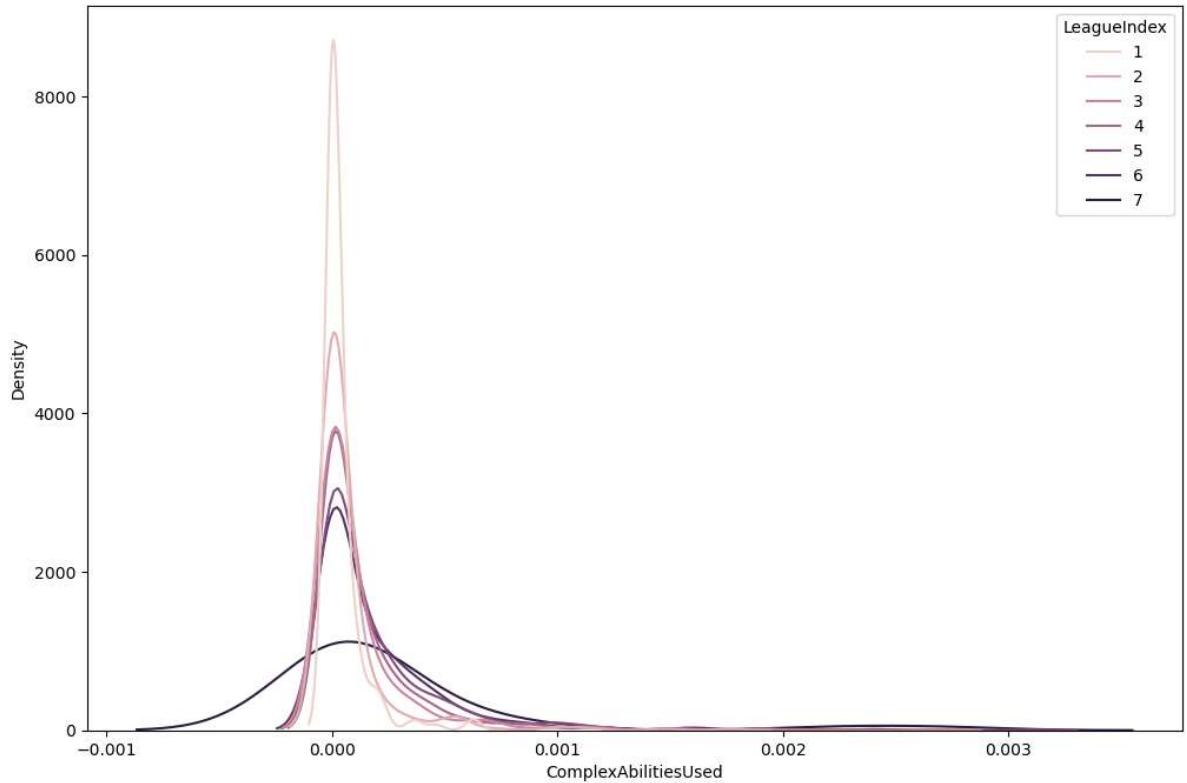
Out[341]: <AxesSubplot:xlabel='MinimapAttacks', ylabel='Density'>



```
In [342]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="SelectByHotkeys", hue = 'LeagueIndex', common_norm = False)
Out[342]: <AxesSubplot:xlabel='SelectByHotkeys', ylabel='Density'>
```

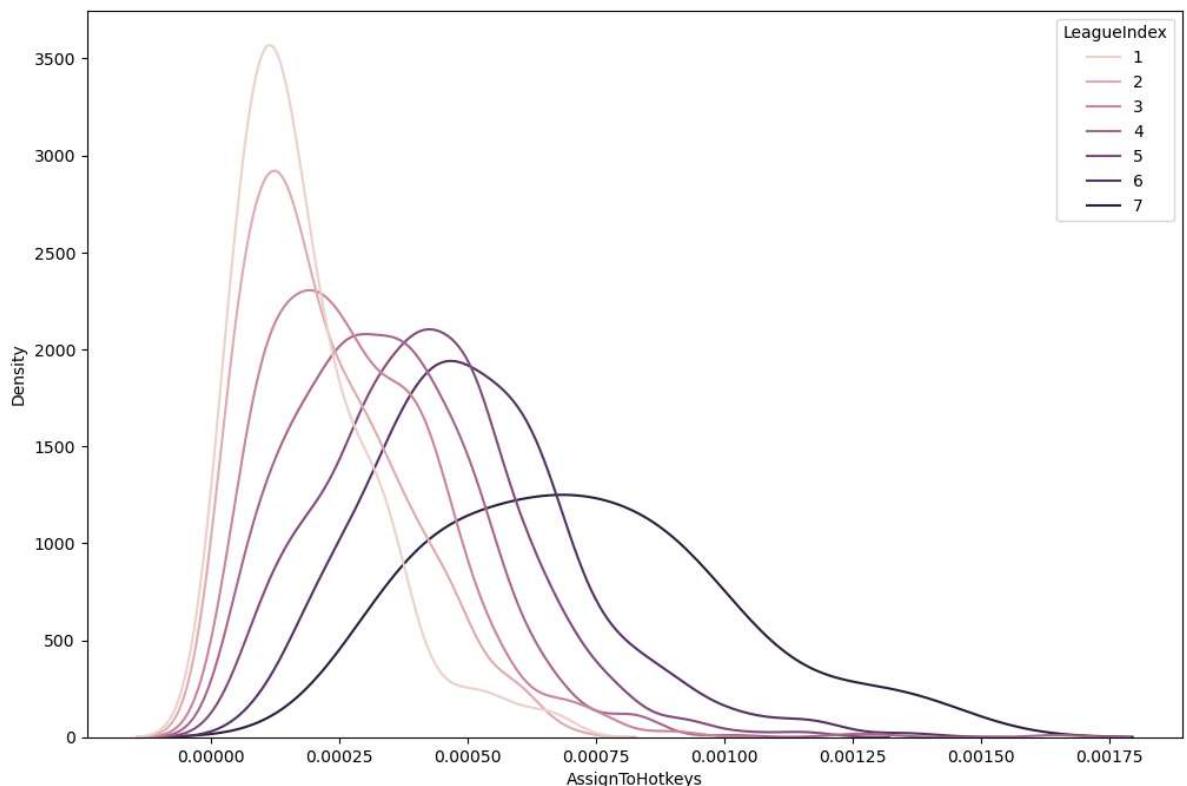


```
In [343]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="ComplexAbilitiesUsed", hue = 'LeagueIndex', common_norm = False)
Out[343]: <AxesSubplot:xlabel='ComplexAbilitiesUsed', ylabel='Density'>
```



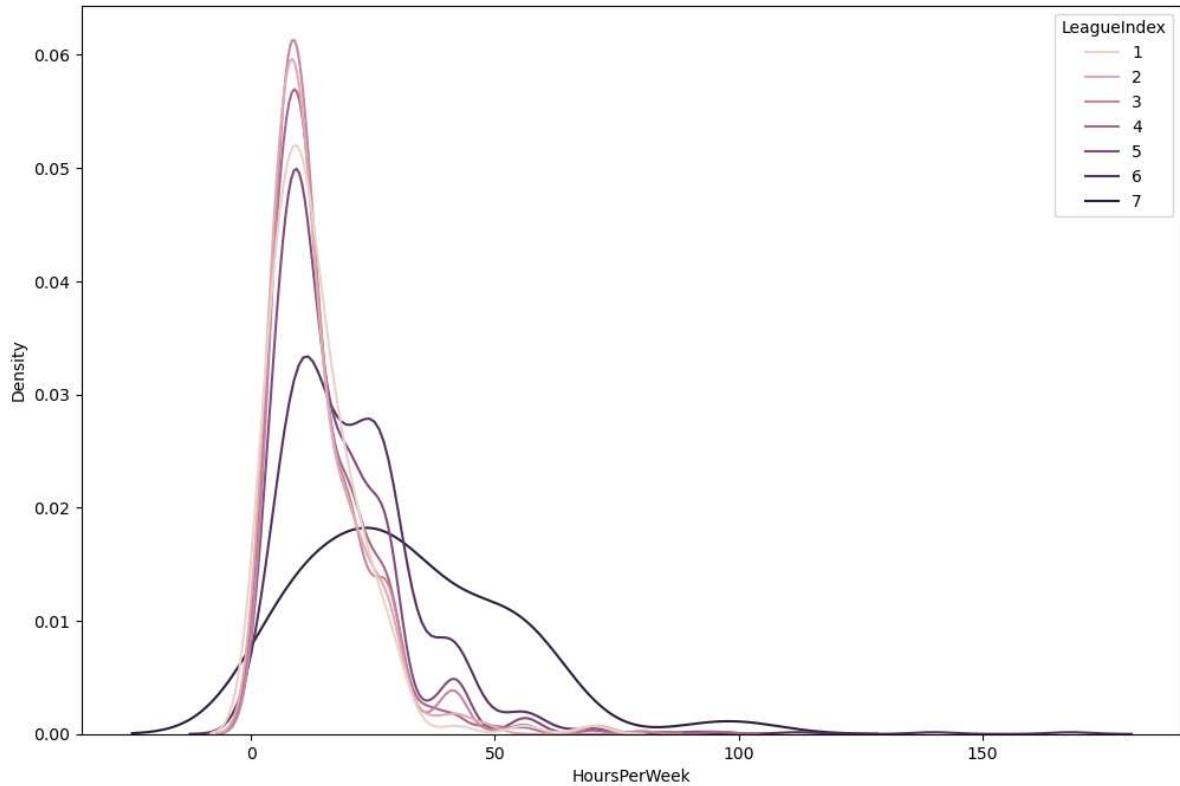
```
In [344]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="AssignToHotkeys", hue = 'LeagueIndex', common_norm = False)
```

```
Out[344]: <AxesSubplot:xlabel='AssignToHotkeys', ylabel='Density'>
```



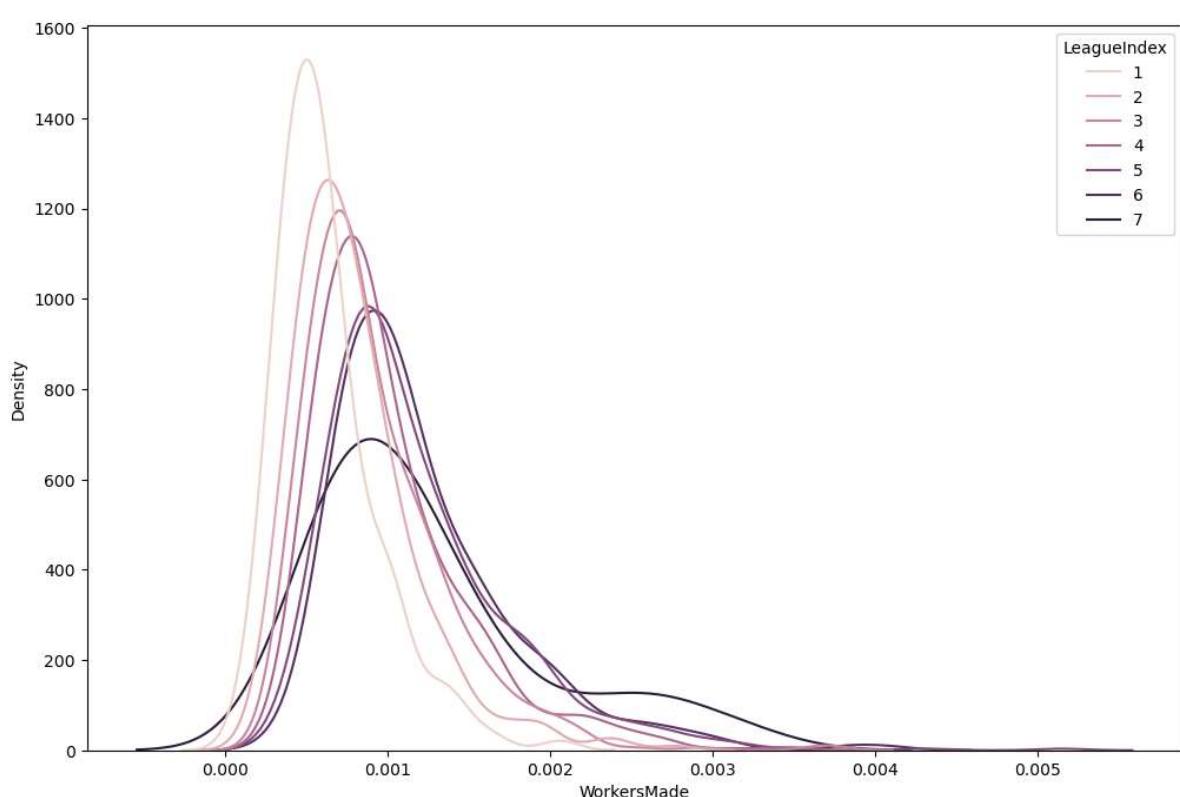
```
In [345]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="HoursPerWeek", hue = 'LeagueIndex', common_norm = False)
```

```
Out[345]: <AxesSubplot:xlabel='HoursPerWeek', ylabel='Density'>
```



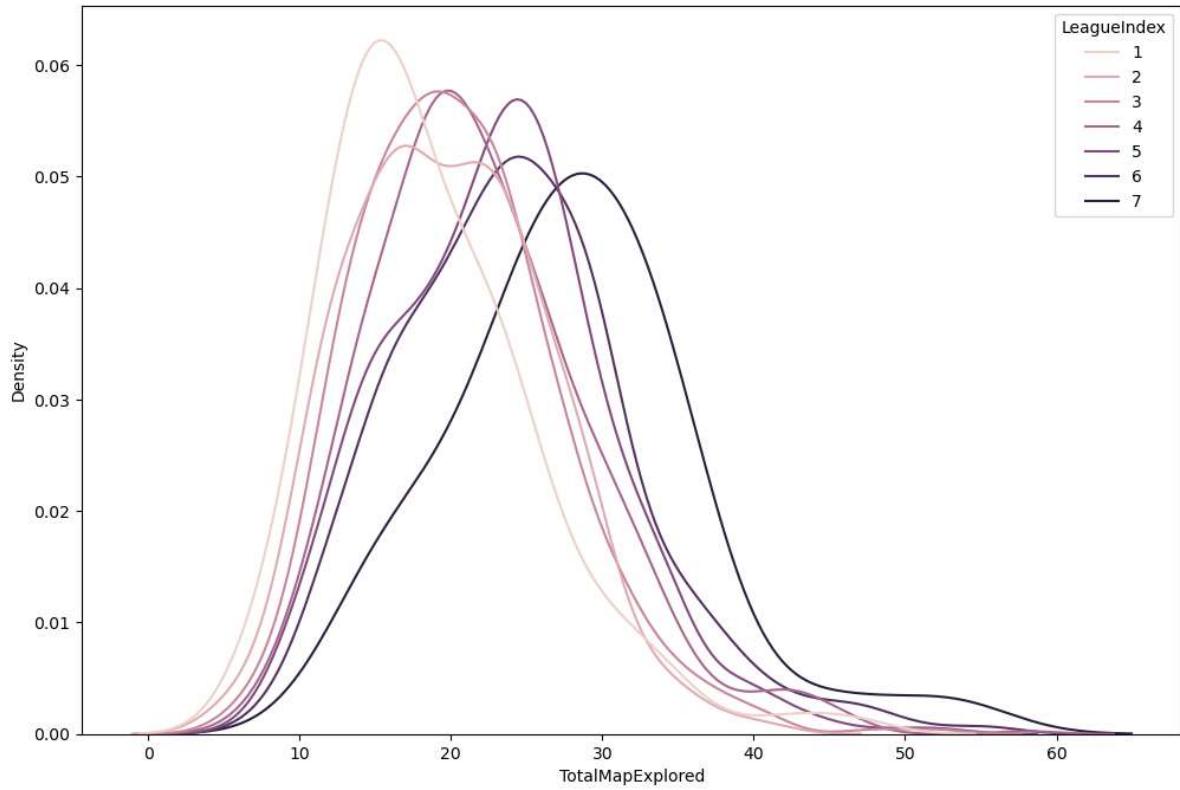
```
In [346]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="WorkersMade", hue = 'LeagueIndex', common_norm = False)

Out[346]: <AxesSubplot:xlabel='WorkersMade', ylabel='Density'>
```



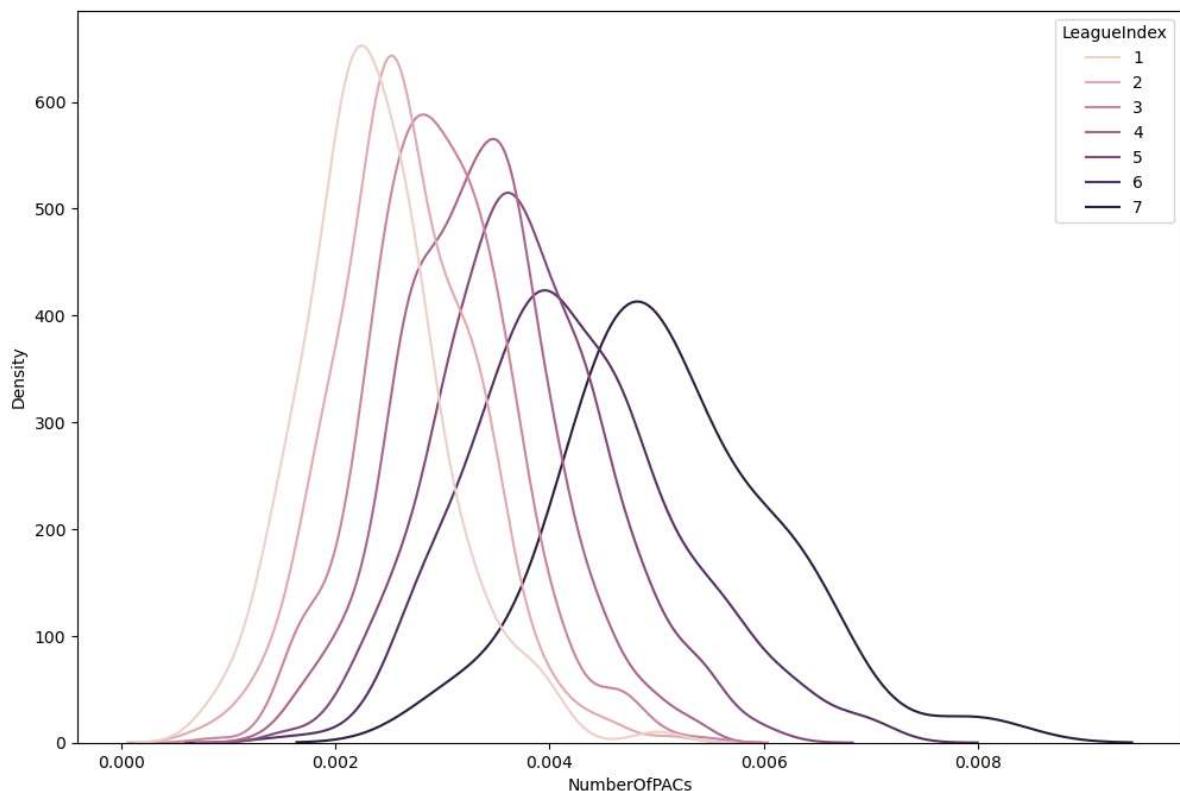
```
In [373]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="TotalMapExplored", hue = 'LeagueIndex', common_norm = False)

Out[373]: <AxesSubplot:xlabel='TotalMapExplored', ylabel='Density'>
```



```
In [374]: plt.figure(figsize=(12, 8))
sns.kdeplot(data=data, x="NumberOfPACs", hue = 'LeagueIndex', common_norm = False)
```

```
Out[374]: <AxesSubplot:xlabel='NumberOfPACs', ylabel='Density'>
```



Features were selected based on Probability Distribution Plot

```
In [376]: features = ['APM', 'TotalMapExplored', 'WorkersMade', 'NumberOfPACs',
 'ComplexAbilitiesUsed', 'SelectByHotkeys', 'AssignToHotkeys']
```

Calculate the median and MAD using numpy functions. The MAD is multiplied by a constant factor of 1.4826 to make it consistent with the definition of the MAD for normally distributed data

```
In [377...]: #data['Rank'] = data[features].median(axis=1)
```

```
In [394...]: median = np.median(data[features], axis=1)

mad = np.median(np.abs(data[features] - median.reshape(-1, 1)), axis=1) * 1.4826

winsorized_data = np.where(np.abs(data[features] - median.reshape(-1, 1)) > 3 * mad,
                           median.reshape(-1, 1), data[features])

data['Rank'] = np.mean(winsorized_data, axis=1)
```

```
In [395...]: data['PlayerRank'] = rankdata(data['Rank'])
```

```
In [380...]: #top_10_players = data.sort_values('PlayerRank').head(10)
#print(top_10_players[['PlayerRank']])
```

	PlayerRank
1485	1.0
394	2.0
1332	3.0
1453	4.0
1030	5.0
1053	6.0
639	7.0
924	8.0
385	9.0
2921	10.0

```
In [381...]: data.head(5)
```

```
Out[381]:
```

	PlayerRank	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	NumberOf
0	2423.0	52	5	27.0	10.0	143.7180	0.000110	0.00
1	2280.0	55	5	23.0	10.0	129.2322	0.000294	0.00
2	695.0	56	4	30.0	10.0	69.9612	0.000294	0.00
3	787.0	57	3	19.0	20.0	107.6016	0.000053	0.00
4	672.0	58	3	32.0	10.0	122.8908	0.000000	0.00

```
In [382...]: #target_rank = 17
#rows_with_target_rank = data[data['PlayerRank'] == target_rank]
#print(rows_with_target_rank)
```

```

PlayerRank GameID LeagueIndex Age HoursPerWeek APM \
2732      17.0    7623        2   22.0          6.0  71.088

MinimapAttacks NumberOfPACs TotalMapExplored WorkersMade \
2732       0.0      0.002446           16     0.000194

ComplexAbilitiesUsed SelectByHotkeys AssignToHotkeys Rank
2732       0.0      0.000298     0.000239  0.000232

```

In [383...]: #data['Rank'].value_counts()

```

Out[383]: 0.001497    2
0.002430    1
0.001025    1
0.003025    1
0.001365    1
..
0.002041    1
0.001737    1
0.001669    1
0.002158    1
0.000896    1
Name: Rank, Length: 3338, dtype: int64

```

In [384...]: #data['PlayerRank'].value_counts()

```

Out[384]: 1321.5    2
2423.0    1
619.0     1
2848.0    1
1144.0    1
..
2040.0    1
1646.0    1
1545.0    1
2156.0    1
428.0     1
Name: PlayerRank, Length: 3338, dtype: int64

```

In [385...]: #target_player_rank = 1
#rows_with_target_player_rank = data[data['PlayerRank'] == target_player_rank]
#print(rows_with_target_player_rank)

```

PlayerRank GameID LeagueIndex Age HoursPerWeek APM \
1485      1.0    4273        2   16.0          14.0  36.24774

MinimapAttacks NumberOfPACs TotalMapExplored WorkersMade \
1485       0.0      0.00227           15     0.000165

ComplexAbilitiesUsed SelectByHotkeys AssignToHotkeys Rank
1485       0.000011     0.000077     0.000022  0.00011

```

In [386...]: #target_player_rank = 2
#rows_with_target_player_rank = data[data['PlayerRank'] == target_player_rank]
#print(rows_with_target_player_rank)

```

      PlayerRank  GameID  LeagueIndex  Age  HoursPerWeek      APM  \
394        2.0      1173           1  26.0          8.0  31.33314

      MinimapAttacks  NumberOfPACs  TotalMapExplored  WorkersMade  \
394            0.0       0.002531             25     0.000166

      ComplexAbilitiesUsed  SelectByHotkeys  AssignToHotkeys      Rank
394            0.000108       0.000149      0.000066  0.000141

```

In [387]: `data['PlayerRank'].min()`

Out[387]: `1.0`

In [388]: `data['PlayerRank'].max()`

Out[388]: `3339.0`

In [389]: `data`

Out[389]:

	PlayerRank	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	Number
0	2423.0	52	5	27.0	10.0	143.7180	0.000110	
1	2280.0	55	5	23.0	10.0	129.2322	0.000294	
2	695.0	56	4	30.0	10.0	69.9612	0.000294	
3	787.0	57	3	19.0	20.0	107.6016	0.000053	
4	672.0	58	3	32.0	10.0	122.8908	0.000000	
...
3335	3158.0	9261	4	20.0	8.0	158.1390	0.000217	
3336	3120.0	9264	5	16.0	56.0	186.1320	0.000083	
3337	2115.0	9265	4	21.0	8.0	121.6992	0.000055	
3338	2318.0	9270	3	20.0	28.0	134.2848	0.000000	
3339	428.0	9271	4	22.0	6.0	88.8246	0.000000	

3339 rows × 14 columns

◀ ▶

In [390]: `sorted_data = data.sort_values('PlayerRank', ascending=True)`

In [391]: `sorted_data`

Out[391]:

	PlayerRank	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	Num
1485	1.0	4273		2	16.0	14.0	36.24774	0.000000
394	2.0	1173		1	26.0	8.0	31.33314	0.000000
1332	3.0	3843		1	20.0	24.0	65.79480	0.000000
1453	4.0	4177		1	16.0	20.0	51.44520	0.000011
1030	5.0	2970		2	26.0	20.0	88.59540	0.000349
...
1685	3335.0	4839		6	24.0	36.0	283.27560	0.000101
2046	3336.0	5768		6	20.0	30.0	240.51960	0.000000
464	3337.0	1352		6	18.0	28.0	286.55280	0.000133
3072	3338.0	8526		6	19.0	20.0	301.76820	0.000143
26	3339.0	144		6	18.0	70.0	267.55860	0.000000

3339 rows × 14 columns

◀	▶
---	---

In [392]:

```
shifted_column = sorted_data.pop('PlayerRank')
sorted_data.insert(0, 'PlayerRank', shifted_column)
```

In [393]:

sorted_data

Out[393]:

	PlayerRank	GameID	LeagueIndex	Age	HoursPerWeek	APM	MinimapAttacks	Num
1485	1.0	4273		2	16.0	14.0	36.24774	0.000000
394	2.0	1173		1	26.0	8.0	31.33314	0.000000
1332	3.0	3843		1	20.0	24.0	65.79480	0.000000
1453	4.0	4177		1	16.0	20.0	51.44520	0.000011
1030	5.0	2970		2	26.0	20.0	88.59540	0.000349
...
1685	3335.0	4839		6	24.0	36.0	283.27560	0.000101
2046	3336.0	5768		6	20.0	30.0	240.51960	0.000000
464	3337.0	1352		6	18.0	28.0	286.55280	0.000133
3072	3338.0	8526		6	19.0	20.0	301.76820	0.000143
26	3339.0	144		6	18.0	70.0	267.55860	0.000000

3339 rows × 14 columns

◀	▶
---	---

In []:

--