In [2]:

```
import pandas as pd
df = pd.read_csv (r"C:\Users\Anusha V\Desktop\csv file.csv")
print(df.head(100))
```

```
    GameID  LeagueIndex   Age  HoursPerWeek  TotalHours       APM  \
0     52.0          5.0  27.0          10.0      3000.0  143.7180
1     55.0          5.0  23.0          10.0      5000.0  129.2322
2     56.0          4.0  30.0          10.0       200.0   69.9612
3     57.0          3.0  19.0          20.0       400.0  107.6016
4     58.0          3.0  32.0          10.0       500.0  122.8908
..     ...          ...   ...           ...         ...       ...
95   321.0          3.0  30.0          12.0       500.0   62.8980
96   329.0          6.0  24.0          24.0      1000.0  146.6856
97   333.0          2.0  16.0          28.0       300.0   76.8504
98   336.0          4.0  23.0           4.0       415.0   59.1810
99   338.0          4.0  31.0           4.0       500.0  100.4322

    SelectByHotkeys  AssignToHotkeys  UniqueHotkeys  MinimapAttacks  \
0          0.003515         0.000220            7.0        0.000110
1          0.003304         0.000259            4.0        0.000294
2          0.001101         0.000336            4.0        0.000294
3          0.001034         0.000213            1.0        0.000053
4          0.001136         0.000327            2.0        0.000000
```

In [3]:
```
1 df.head(50)
```

Out[3]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	Assig
0	52.0	5.0	27.0	10.0	3000.0	143.7180	0.003515	
1	55.0	5.0	23.0	10.0	5000.0	129.2322	0.003304	
2	56.0	4.0	30.0	10.0	200.0	69.9612	0.001101	
3	57.0	3.0	19.0	20.0	400.0	107.6016	0.001034	
4	58.0	3.0	32.0	10.0	500.0	122.8908	0.001136	
5	60.0	2.0	27.0	6.0	70.0	44.4570	0.000978	
6	61.0	1.0	21.0	8.0	240.0	46.9962	0.000820	
7	72.0	7.0	17.0	42.0	10000.0	212.6022	0.009040	
8	77.0	4.0	20.0	14.0	2708.0	117.4884	0.002944	
9	81.0	4.0	18.0	24.0	800.0	155.9856	0.005054	
10	83.0	3.0	16.0	16.0	6000.0	153.8010	0.001677	
11	93.0	4.0	26.0	4.0	190.0	79.2948	0.000379	
12	97.0	3.0	18.0	12.0	350.0	67.4754	0.000423	
13	98.0	3.0	38.0	6.0	1000.0	119.4366	0.004952	
14	100.0	5.0	16.0	30.0	5000.0	160.4754	0.004254	
15	102.0	5.0	17.0	16.0	1500.0	81.7722	0.002333	
16	105.0	4.0	28.0	8.0	2000.0	50.8374	0.000664	
17	106.0	5.0	20.0	10.0	120.0	160.6464	0.003430	
18	118.0	5.0	16.0	14.0	350.0	107.9118	0.006701	
19	127.0	4.0	26.0	28.0	1100.0	114.7806	0.002630	
20	132.0	5.0	21.0	10.0	800.0	115.1274	0.002651	
21	138.0	6.0	21.0	6.0	500.0	133.7016	0.004500	
22	139.0	5.0	18.0	20.0	800.0	99.5088	0.000734	
23	140.0	5.0	26.0	10.0	500.0	83.9172	0.002854	
24	141.0	4.0	17.0	14.0	500.0	216.6936	0.012495	
25	142.0	4.0	23.0	20.0	800.0	129.8598	0.002315	
26	144.0	6.0	18.0	70.0	2520.0	267.5586	0.027815	
27	149.0	5.0	25.0	6.0	800.0	74.1174	0.000875	
28	154.0	5.0	25.0	20.0	700.0	101.6796	0.001406	
29	158.0	6.0	18.0	10.0	160.0	150.5004	0.005667	
30	160.0	3.0	19.0	6.0	150.0	64.6416	0.000672	
31	161.0	3.0	18.0	8.0	250.0	41.9094	0.000276	
32	162.0	5.0	16.0	16.0	1000.0	128.0784	0.002754	
33	163.0	6.0	16.0	28.0	500.0	161.3466	0.006110	
34	168.0	4.0	22.0	4.0	400.0	90.7686	0.001610	
35	169.0	6.0	20.0	14.0	730.0	162.0876	0.007597	
36	171.0	1.0	18.0	6.0	230.0	69.5076	0.000175	
37	175.0	3.0	16.0	24.0	300.0	81.0702	0.001747	
38	178.0	5.0	22.0	8.0	300.0	157.3320	0.007743	

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	Assi
39	180.0	4.0	23.0	4.0	100.0	67.6194	0.001566	
40	181.0	2.0	25.0	10.0	200.0	75.9642	0.000610	
41	182.0	5.0	25.0	12.0	200.0	164.1114	0.003372	
42	184.0	4.0	20.0	14.0	270.0	101.7864	0.001453	
43	187.0	2.0	25.0	8.0	100.0	79.1280	0.000501	
44	193.0	4.0	22.0	20.0	1200.0	120.7014	0.005496	
45	194.0	6.0	18.0	20.0	800.0	108.5424	0.002101	
46	196.0	5.0	18.0	28.0	800.0	84.1578	0.002368	
47	201.0	3.0	23.0	2.0	30.0	66.7818	0.000686	
48	203.0	6.0	18.0	28.0	2000.0	129.1464	0.002631	
49	204.0	3.0	18.0	8.0	600.0	62.2194	0.001099	

In [4]:
```
1 df.tail(10)
```

Out[4]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	Assigr
3380	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3381	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3382	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3383	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3384	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3385	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3386	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3387	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3388	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3389	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [5]:
```
pivot_table = df.pivot_table(index = 'GameID', aggfunc='mean')
pivot_table
```

Out[5]:

GameID	APM	ActionLatency	ActionsInPAC	Age	AssignToHotkeys	ComplexAbilitiesUse
52.0	143.7180	40.8673	4.7508	27.0	0.000220	0.00000
55.0	129.2322	42.3454	4.8434	23.0	0.000259	0.00020
56.0	69.9612	75.3548	4.0430	30.0	0.000336	0.00018
57.0	107.6016	53.7352	4.9155	19.0	0.000213	0.00038
58.0	122.8908	62.0813	9.3740	32.0	0.000327	0.00001
...	...	...	...	...	...	.
849.0	116.1894	56.2857	3.8789	20.0	0.000560	0.00000
851.0	130.2162	50.3720	5.4582	22.0	0.000648	0.00036
865.0	69.9006	76.6039	4.0613	21.0	0.000184	0.00000
866.0	61.4670	85.5431	3.9345	20.0	0.000029	0.00005
872.0	132.0822	65.2500	11.7266	25.0	0.000681	0.00006

298 rows × 19 columns

In [6]:
```
melt_df = pd.melt(df,id_vars=['GameID'], var_name='UniqueUnitMade')
melt_df
```

Out[6]:

	GameID	UniqueUnitMade	value
0	52.0	LeagueIndex	5.0
1	55.0	LeagueIndex	5.0
2	56.0	LeagueIndex	4.0
3	57.0	LeagueIndex	3.0
4	58.0	LeagueIndex	3.0
...	...	...	...
64405	NaN	ComplexAbilitiesUsed	NaN
64406	NaN	ComplexAbilitiesUsed	NaN
64407	NaN	ComplexAbilitiesUsed	NaN
64408	NaN	ComplexAbilitiesUsed	NaN
64409	NaN	ComplexAbilitiesUsed	NaN

64410 rows × 3 columns

In [7]:

```
from functools import reduce
def multiply(x, y):
    return x * y
column_to_reduce = 'LeagueIndex'
result = reduce(multiply, df['LeagueIndex'])
print(result)
```

```
nan
```

In [8]:

```
def map_grades(grade):
    grade_mapping = {
        'A':90,
        'B':80,
        'C':70,
        'D':60,
        'E':50,
        'F':40
}
    return grade_mapping.get(grade, 0)
LeagueIndex = 'Grade'
df['Numeric_Score'] = df['HoursPerWeek'].map(map_grades)
print(df)
```

```
      GameID  LeagueIndex   Age  HoursPerWeek  TotalHours       APM  \
0       52.0          5.0  27.0          10.0      3000.0  143.7180
1       55.0          5.0  23.0          10.0      5000.0  129.2322
2       56.0          4.0  30.0          10.0       200.0   69.9612
3       57.0          3.0  19.0          20.0       400.0  107.6016
4       58.0          3.0  32.0          10.0       500.0  122.8908
...      ...          ...   ...           ...         ...       ...
3385     NaN          NaN   NaN           NaN         NaN       NaN
3386     NaN          NaN   NaN           NaN         NaN       NaN
3387     NaN          NaN   NaN           NaN         NaN       NaN
3388     NaN          NaN   NaN           NaN         NaN       NaN
3389     NaN          NaN   NaN           NaN         NaN       NaN

      SelectByHotkeys  AssignToHotkeys  UniqueHotkeys  MinimapAttacks  ...
\
0            0.003515         0.000220            7.0        0.000110  ...
1            0.003304         0.000259            4.0        0.000294  ...
2            0.001101         0.000336            4.0        0.000294  ...
3            0.001034         0.000213            1.0        0.000053  ...
4            0.001136         0.000327            2.0        0.000000  ...
...               ...              ...            ...             ...  ...
3385              NaN              NaN            NaN             NaN  ...
3386              NaN              NaN            NaN             NaN  ...
3387              NaN              NaN            NaN             NaN  ...
3388              NaN              NaN            NaN             NaN  ...
3389              NaN              NaN            NaN             NaN  ...

      NumberOfPACs  GapBetweenPACs  ActionLatency  ActionsInPAC  \
0         0.004849         32.6677        40.8673        4.7508
1         0.004307         32.9194        42.3454        4.8434
2         0.002926         44.6475        75.3548        4.0430
3         0.003783         29.2203        53.7352        4.9155
4         0.002368         22.6885        62.0813        9.3740
...            ...             ...            ...           ...
3385           NaN             NaN            NaN           NaN
3386           NaN             NaN            NaN           NaN
3387           NaN             NaN            NaN           NaN
3388           NaN             NaN            NaN           NaN
3389           NaN             NaN            NaN           NaN

      TotalMapExplored  WorkersMade  UniqueUnitsMade  ComplexUnitsMade  \
0                 28.0     0.001397              6.0               0.0
1                 22.0     0.001193              5.0               0.0
2                 22.0     0.000745              6.0               0.0
3                 19.0     0.000426              7.0               0.0
4                 15.0     0.001174              4.0               0.0
...                ...          ...              ...               ...
3385               NaN          NaN              NaN               NaN
3386               NaN          NaN              NaN               NaN
3387               NaN          NaN              NaN               NaN
3388               NaN          NaN              NaN               NaN
3389               NaN          NaN              NaN               NaN

      ComplexAbilitiesUsed  Numeric_Score
0                 0.000000              0
1                 0.000208              0
2                 0.000189              0
3                 0.000384              0
4                 0.000019              0
...                    ...            ...
3385                   NaN              0
```

```
3386                  NaN              0
3387                  NaN              0
3388                  NaN              0
3389                  NaN              0

[3390 rows x 21 columns]
```

In [9]:

```
def score_condition(score):
    return score > 5
LeagueIndex = 'score'
filtered_df = df[df['LeagueIndex'].apply(score_condition)]
print(filtered_df)
```

```
     GameID  LeagueIndex   Age  HoursPerWeek  TotalHours       APM  \
7      72.0          7.0  17.0          42.0     10000.0  212.6022
21    138.0          6.0  21.0           6.0       500.0  133.7016
26    144.0          6.0  18.0          70.0      2520.0  267.5586
29    158.0          6.0  18.0          10.0       160.0  150.5004
33    163.0          6.0  16.0          28.0       500.0  161.3466
..      ...          ...   ...           ...         ...       ...
276   796.0          6.0  23.0          28.0      1260.0  240.7638
280   813.0          6.0  24.0          20.0      1000.0   94.0662
287   834.0          6.0  22.0          10.0       600.0  151.8984
291   842.0          6.0  18.0          12.0       420.0  193.3734
293   849.0          6.0  20.0           6.0       800.0  116.1894

     SelectByHotkeys  AssignToHotkeys  UniqueHotkeys  MinimapAttacks  ...
\
7           0.009040         0.000676            6.0        0.001164  ...
21          0.004500         0.000420            3.0        0.000019  ...
26          0.027815         0.000708           10.0        0.000000  ...
29          0.005667         0.000632            6.0        0.000376  ...
33          0.006110         0.000577            9.0        0.000328  ...
..               ...              ...            ...             ...  ...
276         0.022020         0.000458            5.0        0.000086  ...
280         0.002651         0.000409            6.0        0.000149  ...
287         0.004397         0.000156            4.0        0.000035  ...
291         0.014781         0.000626            5.0        0.000068  ...
293         0.003470         0.000560            5.0        0.000042  ...

     NumberOfPACs  GapBetweenPACs  ActionLatency  ActionsInPAC  \
7        0.004952         24.6117        41.7671        6.6104
21       0.003874         21.4686        50.5253        5.4892
26       0.005616         34.6035        40.6025        4.1629
29       0.004626         30.2222        45.5941        4.9077
33       0.004135         34.1209        47.1034        5.8466
..            ...             ...            ...           ...
276      0.004004         33.2086        66.8857        5.7857
280      0.003345         41.7695        57.5407        4.6667
287      0.003618         44.9808        58.4306        6.6077
291      0.002757         21.4815        68.8589        8.2945
293      0.004506         42.4798        56.2857        3.8789

     TotalMapExplored  WorkersMade  UniqueUnitsMade  ComplexUnitsMade  \
7                45.0     0.002277              9.0          0.000129
21               29.0     0.001008             10.0          0.000000
26               36.0     0.000856             12.0          0.000088
29               14.0     0.001007              6.0          0.000000
33               48.0     0.001347              9.0          0.000378
..                ...          ...              ...               ...
276              14.0     0.002946              4.0          0.000000
280              26.0     0.000818             10.0          0.000087
287              17.0     0.000537              5.0          0.000000
291              12.0     0.001235              6.0          0.000000
293              28.0     0.001511              7.0          0.000000

     ComplexAbilitiesUsed  Numeric_Score
7                0.000249              0
21               0.000560              0
26               0.000197              0
29               0.000000              0
33               0.000171              0
..                    ...            ...
276              0.000000              0
```

```
280              0.000334              0
287              0.000000              0
291              0.000000              0
293              0.000000              0

[64 rows x 21 columns]
```

In [10]:

```
modify_score = lambda x: x * 2
LeagueIndex = 'score'
df['modified-score'] = df['LeagueIndex'].apply(modify_score)
print(df)
```

```
      GameID  LeagueIndex   Age  HoursPerWeek  TotalHours       APM  \
0       52.0          5.0  27.0          10.0      3000.0  143.7180
1       55.0          5.0  23.0          10.0      5000.0  129.2322
2       56.0          4.0  30.0          10.0       200.0   69.9612
3       57.0          3.0  19.0          20.0       400.0  107.6016
4       58.0          3.0  32.0          10.0       500.0  122.8908
...      ...          ...   ...           ...         ...       ...
3385     NaN          NaN   NaN           NaN         NaN       NaN
3386     NaN          NaN   NaN           NaN         NaN       NaN
3387     NaN          NaN   NaN           NaN         NaN       NaN
3388     NaN          NaN   NaN           NaN         NaN       NaN
3389     NaN          NaN   NaN           NaN         NaN       NaN

      SelectByHotkeys  AssignToHotkeys  UniqueHotkeys  MinimapAttacks  ...
\
0            0.003515         0.000220            7.0        0.000110  ...
1            0.003304         0.000259            4.0        0.000294  ...
2            0.001101         0.000336            4.0        0.000294  ...
3            0.001034         0.000213            1.0        0.000053  ...
4            0.001136         0.000327            2.0        0.000000  ...
...               ...              ...            ...             ...  ...
3385             NaN              NaN            NaN             NaN  ...
3386             NaN              NaN            NaN             NaN  ...
3387             NaN              NaN            NaN             NaN  ...
3388             NaN              NaN            NaN             NaN  ...
3389             NaN              NaN            NaN             NaN  ...

      GapBetweenPACs  ActionLatency  ActionsInPAC  TotalMapExplored  \
0            32.6677        40.8673        4.7508              28.0
1            32.9194        42.3454        4.8434              22.0
2            44.6475        75.3548        4.0430              22.0
3            29.2203        53.7352        4.9155              19.0
4            22.6885        62.0813        9.3740              15.0
...              ...            ...           ...               ...
3385             NaN            NaN           NaN               NaN
3386             NaN            NaN           NaN               NaN
3387             NaN            NaN           NaN               NaN
3388             NaN            NaN           NaN               NaN
3389             NaN            NaN           NaN               NaN

      WorkersMade  UniqueUnitsMade  ComplexUnitsMade  ComplexAbilitiesUsed
\
0        0.001397              6.0               0.0              0.000000
1        0.001193              5.0               0.0              0.000208
2        0.000745              6.0               0.0              0.000189
3        0.000426              7.0               0.0              0.000384
4        0.001174              4.0               0.0              0.000019
...           ...              ...               ...                   ...
3385          NaN              NaN               NaN                   NaN
3386          NaN              NaN               NaN                   NaN
3387          NaN              NaN               NaN                   NaN
3388          NaN              NaN               NaN                   NaN
3389          NaN              NaN               NaN                   NaN

      Numeric_Score  modified-score
0                 0            10.0
1                 0            10.0
2                 0             8.0
3                 0             6.0
4                 0             6.0
...             ...             ...
```

```
3385              0             NaN
3386              0             NaN
3387              0             NaN
3388              0             NaN
3389              0             NaN

[3390 rows x 22 columns]
```

In [14]:

```
sorted_df = df.sort_values(by='GameID', ascending=False)
print(sorted_df)
```

```
      GameID  LeagueIndex   Age  HoursPerWeek  TotalHours       APM  \
297    872.0          5.0  25.0          28.0      1300.0  132.0822
296    866.0          3.0  20.0           6.0      1000.0   61.4670
295    865.0          4.0  21.0          12.0       350.0   69.9006
294    851.0          4.0  22.0          10.0       350.0  130.2162
293    849.0          6.0  20.0           6.0       800.0  116.1894
...      ...          ...   ...           ...         ...       ...
3385     NaN          NaN   NaN           NaN         NaN       NaN
3386     NaN          NaN   NaN           NaN         NaN       NaN
3387     NaN          NaN   NaN           NaN         NaN       NaN
3388     NaN          NaN   NaN           NaN         NaN       NaN
3389     NaN          NaN   NaN           NaN         NaN       NaN

      SelectByHotkeys  AssignToHotkeys  UniqueHotkeys  MinimapAttacks  ...
\
297          0.002486         0.000681            6.0        0.000269  ...
296          0.000040         0.000029            4.0        0.000057  ...
295          0.001390         0.000184            5.0        0.000033  ...
294          0.004411         0.000648            2.0        0.000011  ...
293          0.003470         0.000560            5.0        0.000042  ...
...               ...              ...            ...             ...  ...
3385              NaN              NaN            NaN             NaN  ...
3386              NaN              NaN            NaN             NaN  ...
3387              NaN              NaN            NaN             NaN  ...
3388              NaN              NaN            NaN             NaN  ...
3389              NaN              NaN            NaN             NaN  ...

      GapBetweenPACs  ActionLatency  ActionsInPAC  TotalMapExplored  \
297          31.0236        65.2500       11.7266              14.0
296          47.2495        85.5431        3.9345              24.0
295          55.8947        76.6039        4.0613              25.0
294          34.7568        50.3720        5.4582              26.0
293          42.4798        56.2857        3.8789              28.0
...              ...            ...           ...               ...
3385             NaN            NaN           NaN               NaN
3386             NaN            NaN           NaN               NaN
3387             NaN            NaN           NaN               NaN
3388             NaN            NaN           NaN               NaN
3389             NaN            NaN           NaN               NaN

      WorkersMade  UniqueUnitsMade  ComplexUnitsMade  ComplexAbilitiesUsed
\
297      0.003198              8.0          0.000000              0.000063
296      0.000388              9.0          0.000000              0.000057
295      0.000728              9.0          0.000000              0.000000
294      0.000872              8.0          0.000415              0.000361
293      0.001511              7.0          0.000000              0.000000
...           ...              ...               ...                   ...
3385          NaN              NaN               NaN                   NaN
3386          NaN              NaN               NaN                   NaN
3387          NaN              NaN               NaN                   NaN
3388          NaN              NaN               NaN                   NaN
3389          NaN              NaN               NaN                   NaN

      Numeric_Score  modified-score
297               0            10.0
296               0             6.0
295               0             8.0
294               0             8.0
293               0            12.0
...             ...             ...
```

```
3385              0              NaN
3386              0              NaN
3387              0              NaN
3388              0              NaN
3389              0              NaN

[3390 rows x 22 columns]
```

Type *Markdown* and LaTeX: α^2

In [15]:
```
sum_result = df['LeagueIndex'].sum()
sum_result
```

Out[15]: 1265.0

In [16]:
```
sum_result = df['LeagueIndex'].mean()
sum_result
```

Out[16]: 4.24496644295302

In [17]:
```
sum_result = df['LeagueIndex'].min()
sum_result
```

Out[17]: 1.0

In [18]:
```
sum_result = df['LeagueIndex'].max()
sum_result
```

Out[18]: 7.0

In [19]:
```
sum_result = df['LeagueIndex'].count()
sum_result
```

Out[19]: 298

In [20]:
```
sum_result = df['LeagueIndex'].median()
sum_result
```

Out[20]: 4.0

In [21]:
```
sum_result = df['LeagueIndex'].std()
sum_result
```

Out[21]: 1.4059965966696235

In [22]:
```
sum_result = df['LeagueIndex'].var()
sum_result
```

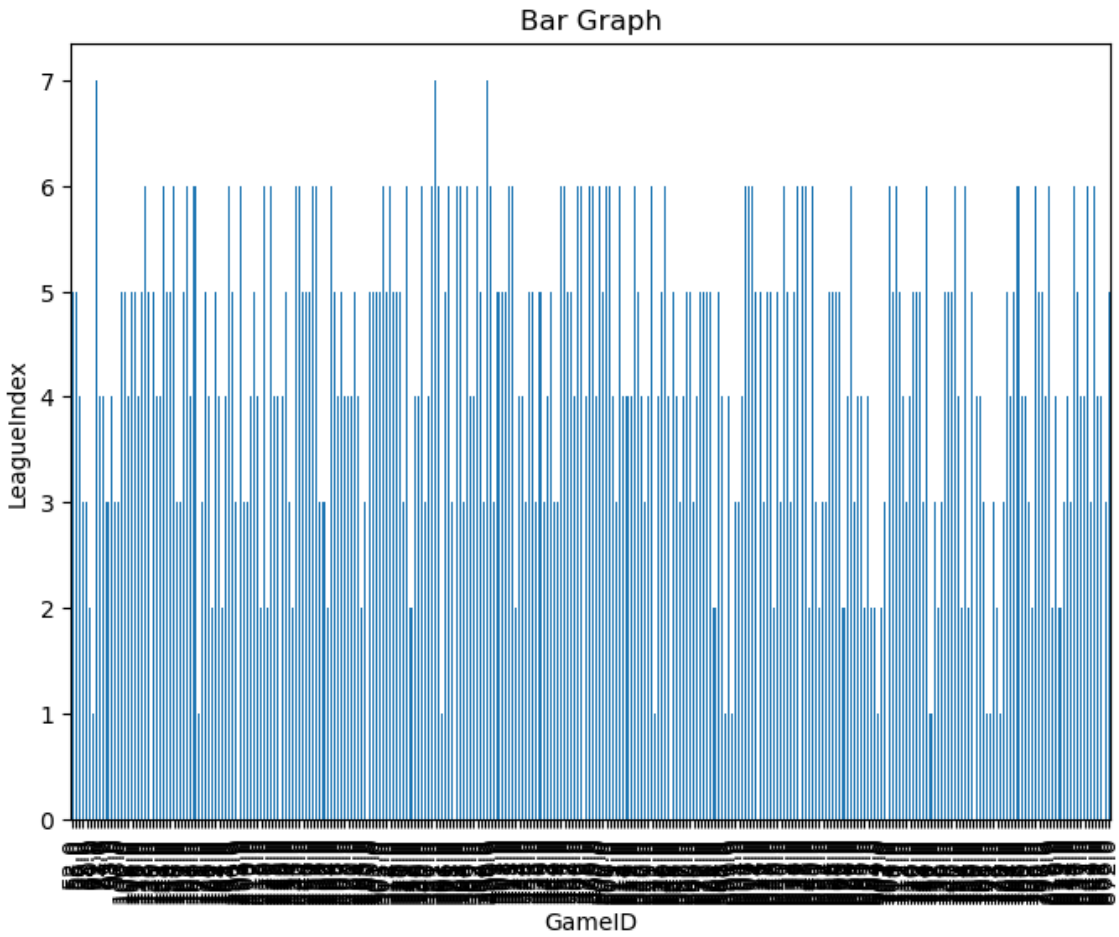
Out[22]: 1.9768264298465639

```
1
```

In [23]:
```
1 grouped = df.groupby('LeagueIndex'), aggfunc='min'
2 print(grouped)
```

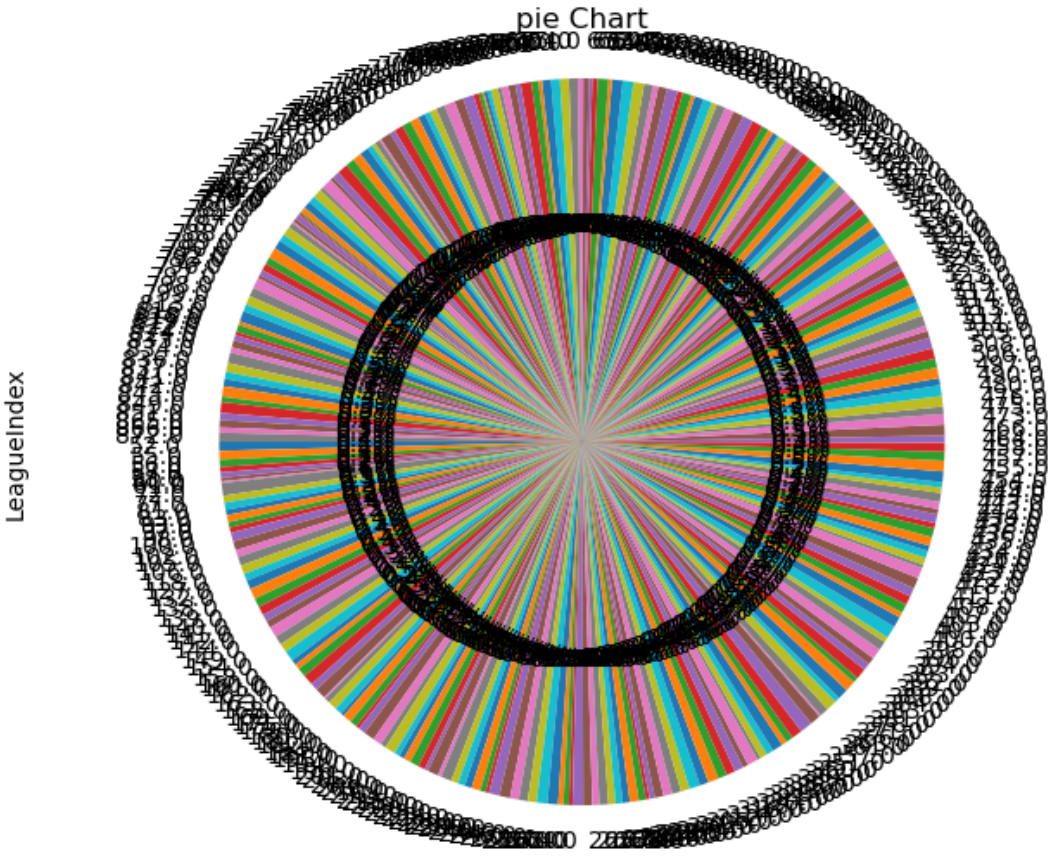
```
  File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_19012\1907097214.p
y", line 1
    grouped = df.groupby('LeagueIndex'), aggfunc='min'
              ^
SyntaxError: cannot assign to function call
```

In [24]:
```
1  import pandas as pd
2  df = pd.read_csv (r"C:\Users\Anusha V\Desktop\csv file.csv")
3  import matplotlib.pyplot as plt
4  bar_data = df.groupby('GameID')['LeagueIndex'].sum()
5  plt.figure(figsize=(8, 6))
6  bar_data.plot(kind='bar')
7  plt.xlabel('GameID')
8  plt.ylabel('LeagueIndex')
9  plt.title('Bar Graph')
10 plt.show()
```



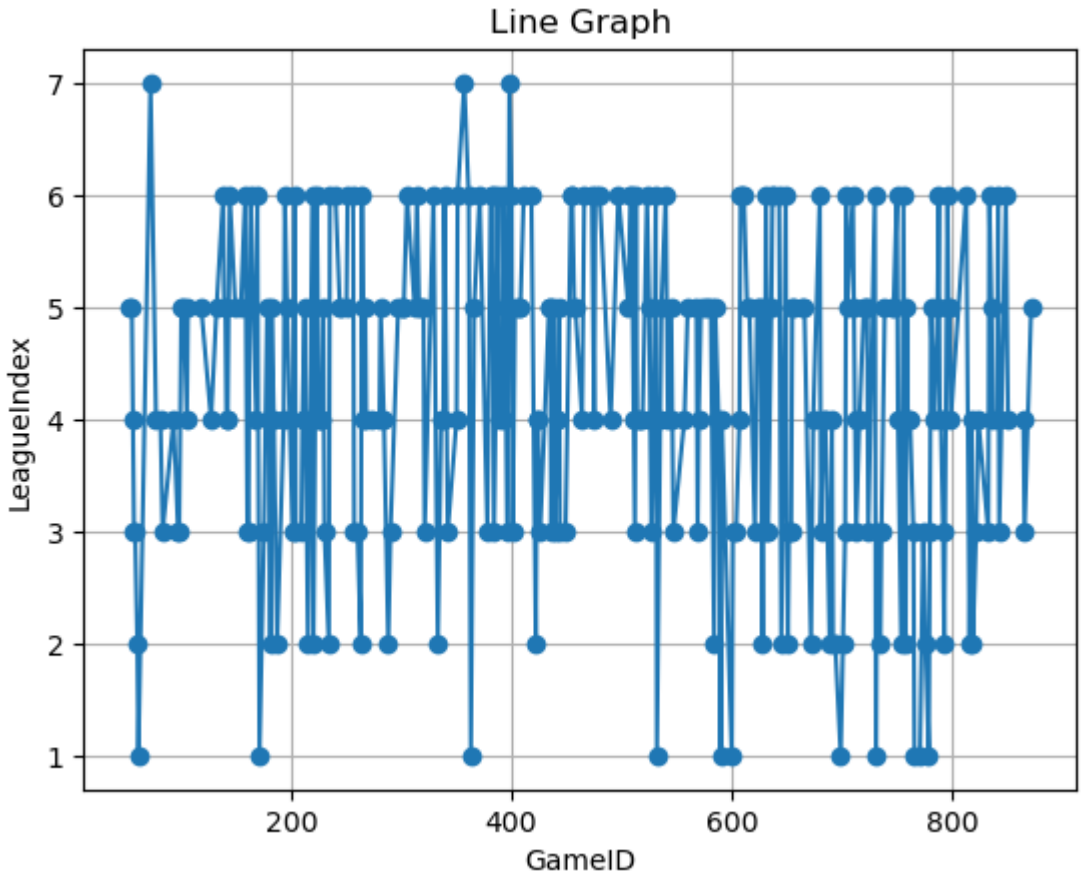
In [25]:

```
pie_data = df.groupby('GameID')['LeagueIndex'].sum()
plt.figure(figsize=(8, 6))
pie_data.plot(kind='pie', autopct='%1.1f%%', startangle=180)
plt.title('pie Chart')
plt.axis('equal')
plt.show()
```



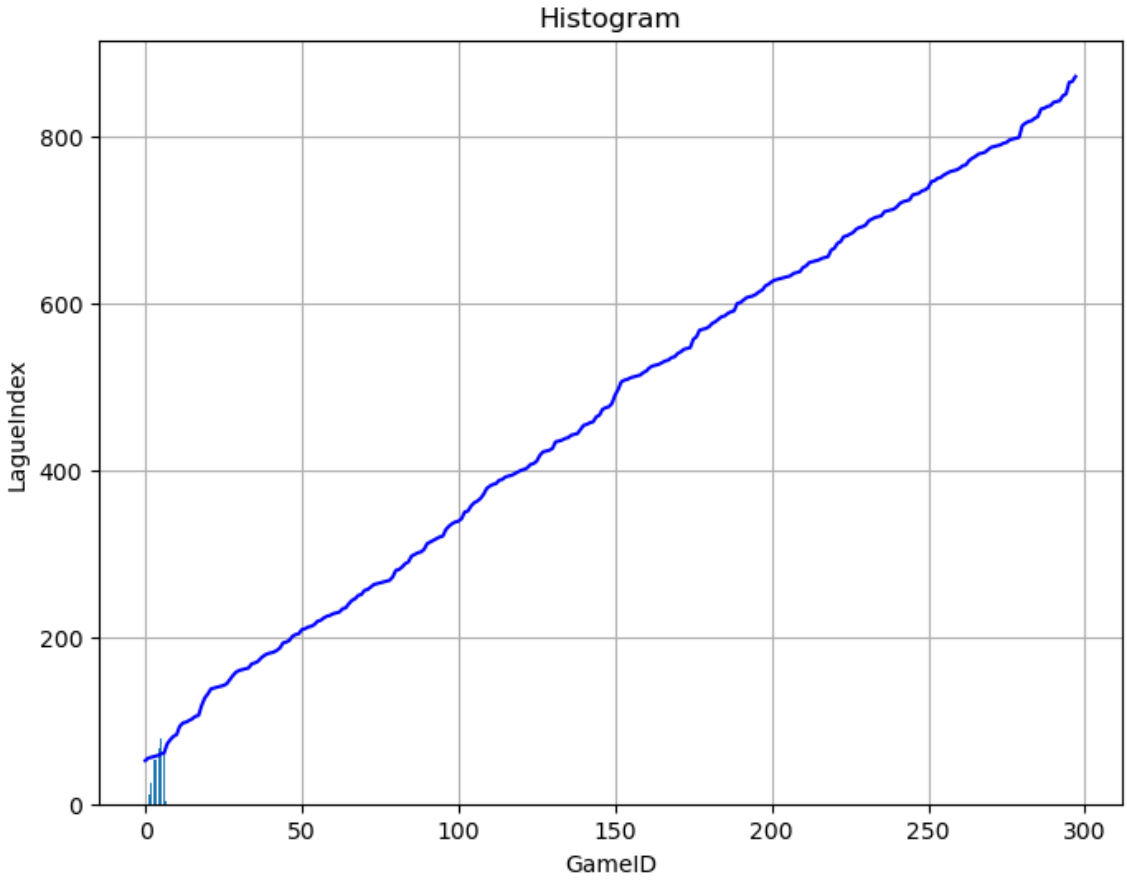
In [26]:

```
x_line = df['GameID']
y_line = df['LeagueIndex']
plt.plot(x_line, y_line, marker='o', linestyle='-')
plt.xlabel('GameID')
plt.ylabel('LeagueIndex')
plt.title('Line Graph')
plt.grid(True)
plt.show()
```



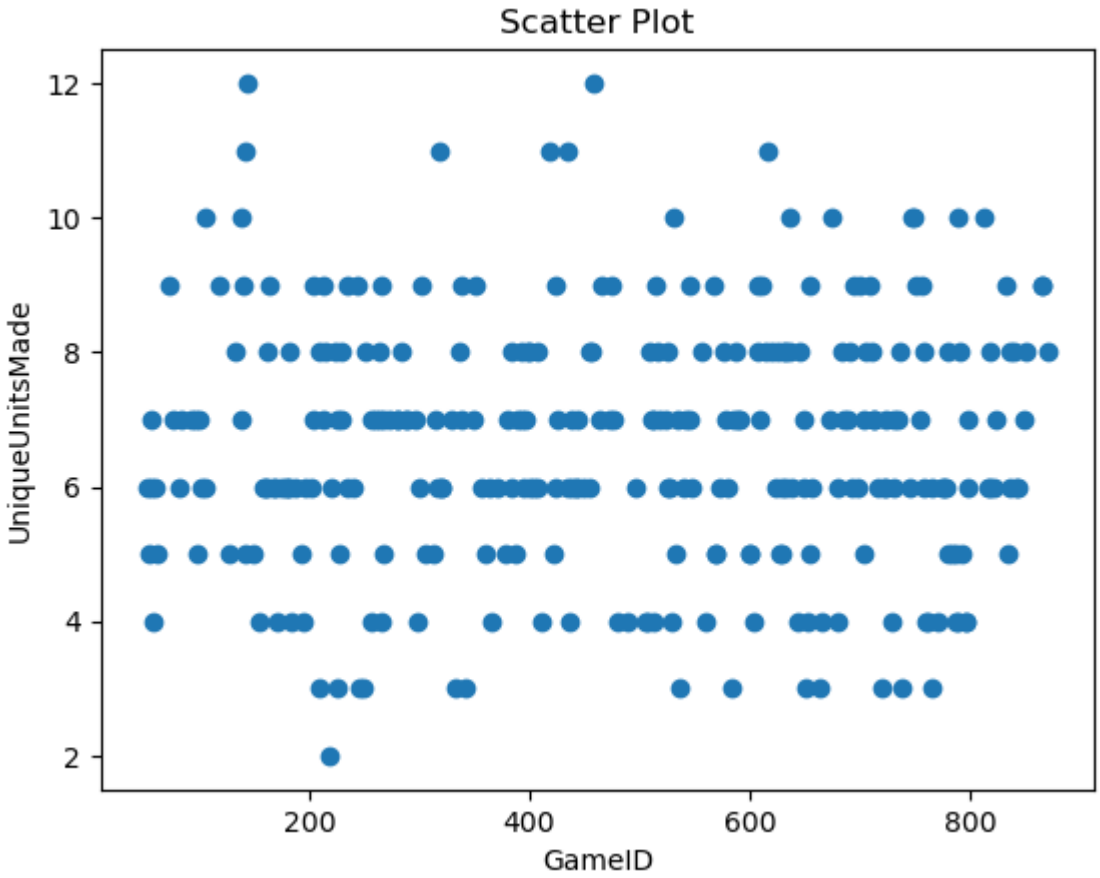
In [29]:

```
data_hist =df['LeagueIndex']
plt.figure(figsize=(8, 6))
plt.hist(data_hist, bins=10)
plt.xlabel('GameID')
plt.ylabel('LagueIndex')
plt.plot(df.index, df['GameID'], label='LagueIndex', color='b')

# Optional: Customize other plot properties

# Display the plot
plt.grid(True)  # Optional: Add grid lin')
plt.title('Histogram')
plt.show()


```



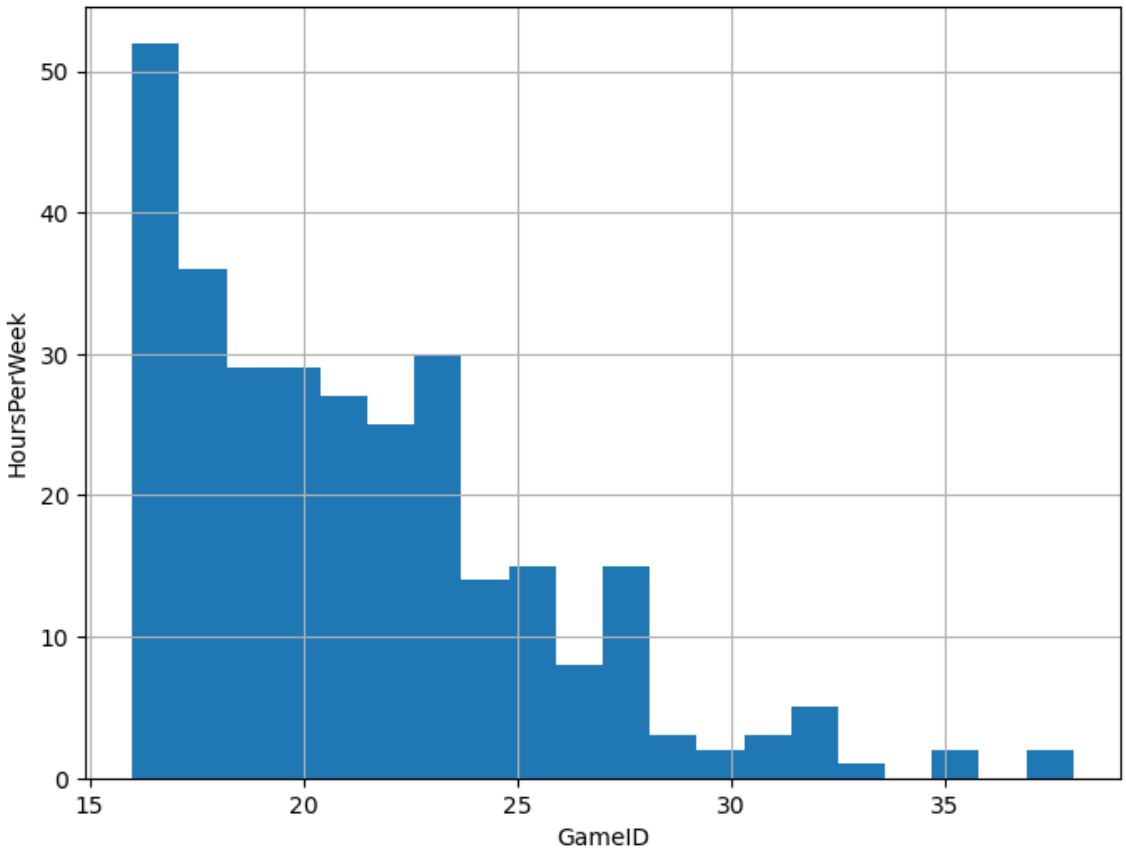
In [30]:

```
import pandas as pd
import matplotlib.pyplot as plt
x_column = 'GameID'
y_column = 'LagueIndex'
# Create a scatter plot
plt.scatter(df['GameID'], df['UniqueUnitsMade'])

# Add labels and title
plt.xlabel('GameID')
plt.ylabel('UniqueUnitsMade')
plt.xlabel('GameID')
plt.title('Scatter Plot')

# Show the plot
plt.show()


```



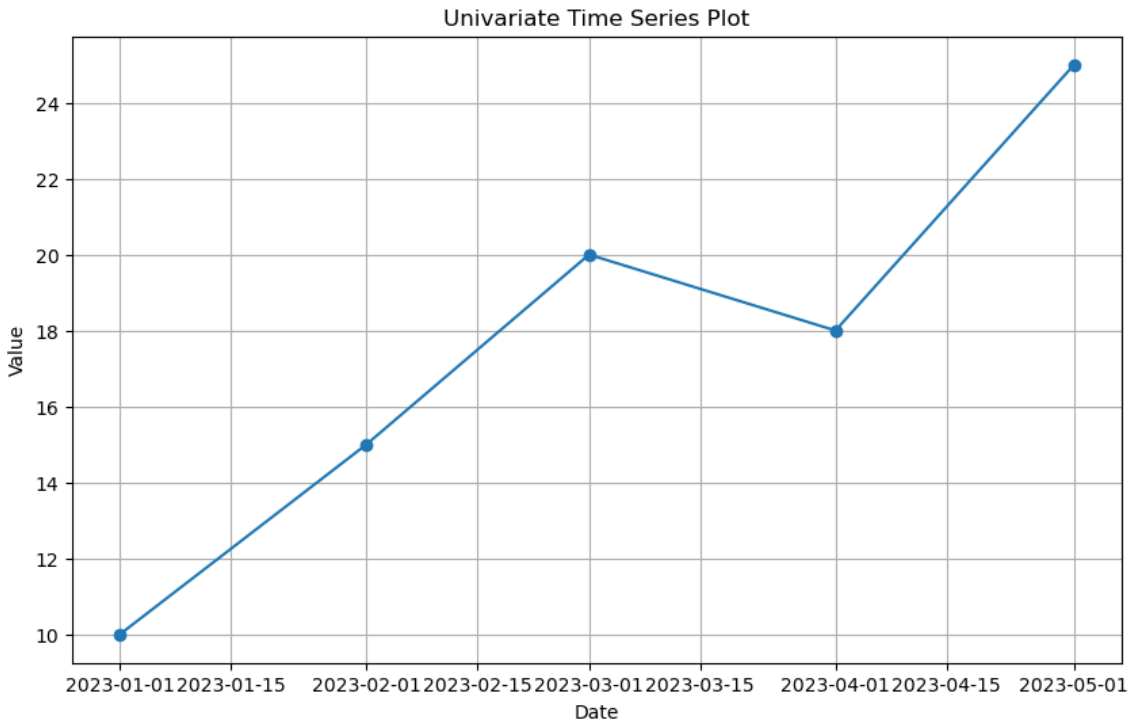
In [31]:

```
passenger_age = df['Age'].dropna()
plt.figure(figsize=(8, 6))
plt.hist(passenger_age, bins=20)
plt.xlabel("GameID")
plt.ylabel("HoursPerWeek")
plt.grid(True)
plt.show()
```



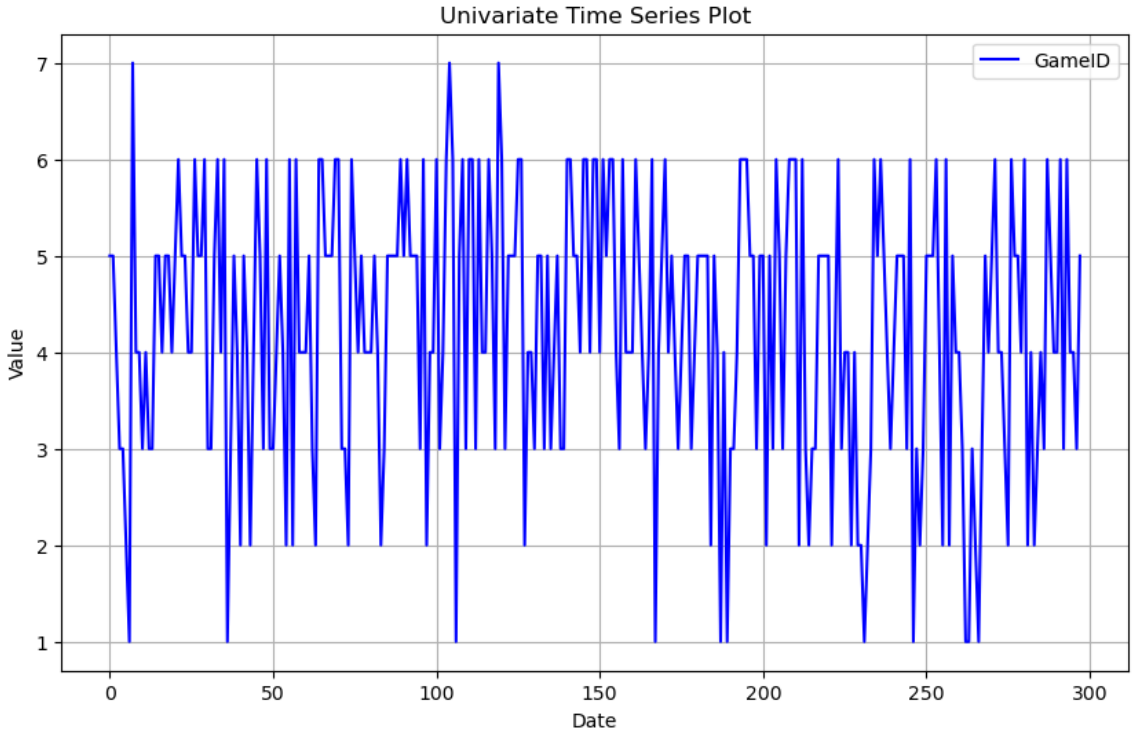
In [8]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data: Date and corresponding values
data = {
    'Date': ['2023-01-01', '2023-02-01', '2023-03-01', '2023-04-01', '2
    'Value': [10, 15, 20, 18, 25]
}

# Create a DataFrame from the sample data
df = pd.DataFrame(data)

# Convert the 'Date' column to a datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Set the 'Date' column as the index of the DataFrame
df.set_index('Date', inplace=True)

# Create a univariate time series plot
plt.figure(figsize=(10, 6))  # Optional: Set the figure size
plt.plot(df.index, df['Value'], marker='o', linestyle='-')

# Optional: Add labels and a title
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Univariate Time Series Plot')

# Display the plot
plt.grid(True)  # Optional: Add grid lines
plt.show()
```



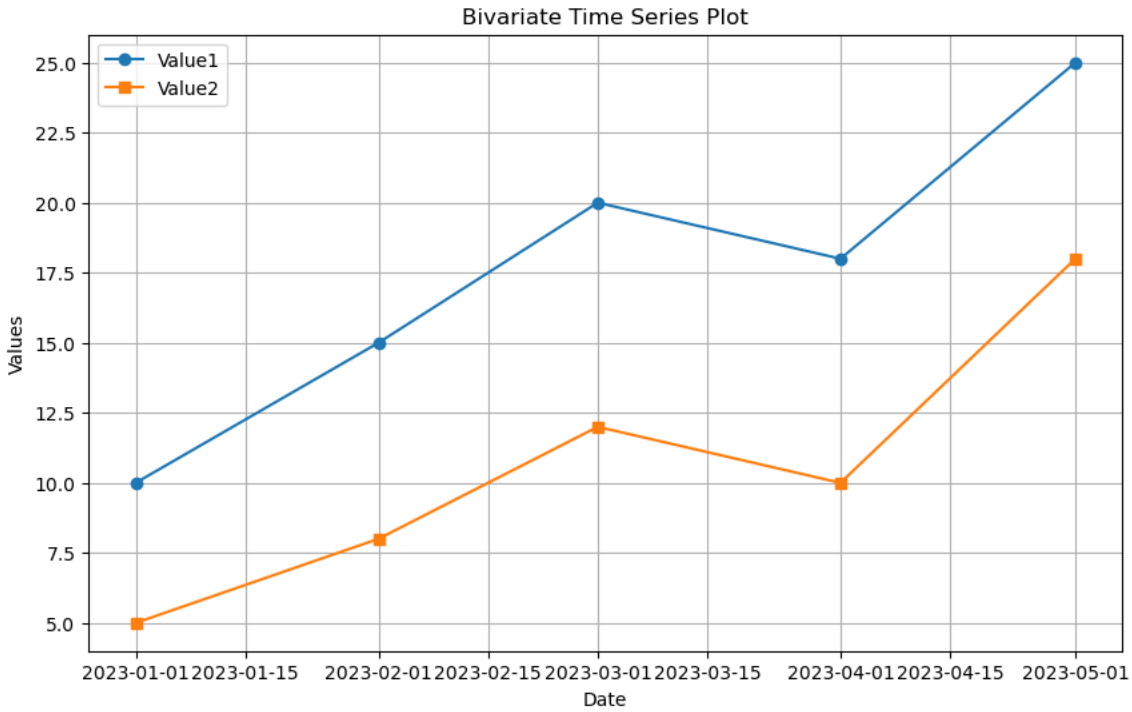
In [32]:

```
plt.figure(figsize=(10, 6))  # Optional: Set the figure size

# Plot the time series data
plt.plot(df.index, df['LeagueIndex'], label='GameID', color='b')

# Optional: Add labels and a title
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Univariate Time Series Plot')

# Optional: Customize other plot properties

# Display the plot
plt.grid(True)  # Optional: Add grid lines
plt.legend()  # Optional: Show the legend if you have multiple series
plt.show()
```



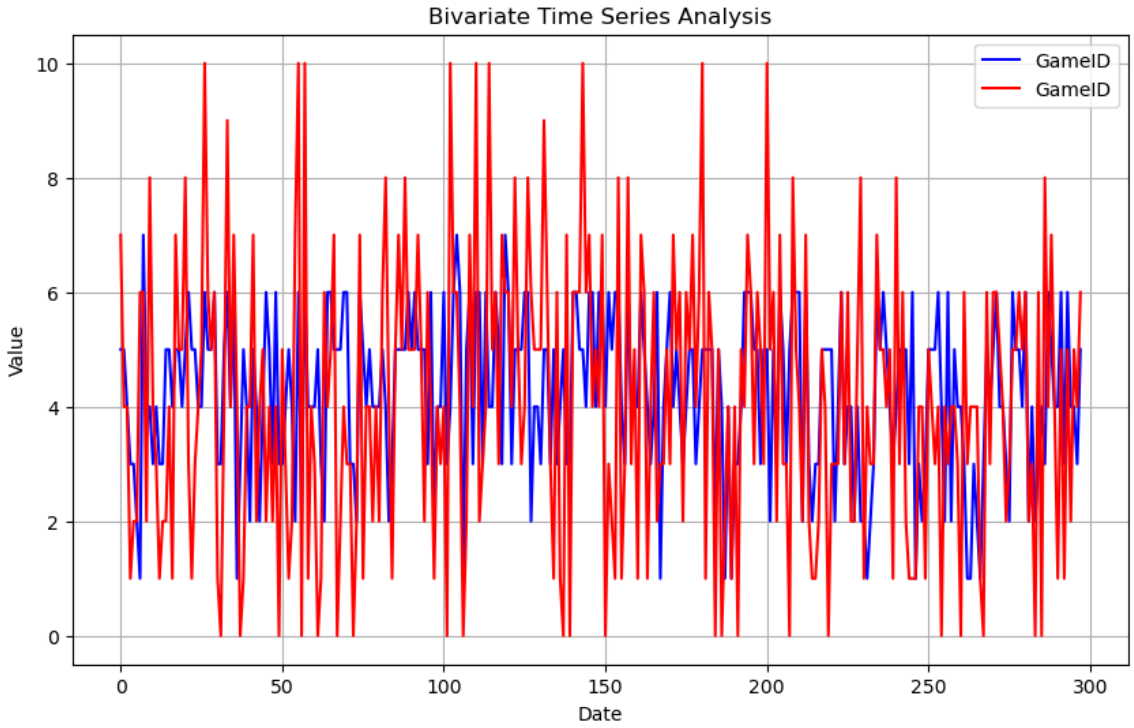
In [9]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data: Date and corresponding values for two variables
data = {
    'Date': ['2023-01-01', '2023-02-01', '2023-03-01', '2023-04-01', '2
    'Value1': [10, 15, 20, 18, 25],
    'Value2': [5, 8, 12, 10, 18]
}

# Create a DataFrame from the sample data
df = pd.DataFrame(data)

# Convert the 'Date' column to a datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Set the 'Date' column as the index of the DataFrame
df.set_index('Date', inplace=True)

# Create a bivariate time series plot
plt.figure(figsize=(10, 6))  # Optional: Set the figure size

# Plot the first variable
plt.plot(df.index, df['Value1'], marker='o', linestyle='-', label='Valu

# Plot the second variable on the same graph
plt.plot(df.index, df['Value2'], marker='s', linestyle='-', label='Valu

# Optional: Add labels and a legend
plt.xlabel('Date')
plt.ylabel('Values')
plt.title('Bivariate Time Series Plot')
plt.legend()

# Display the plot
plt.grid(True)  # Optional: Add grid lines
plt.show()
```



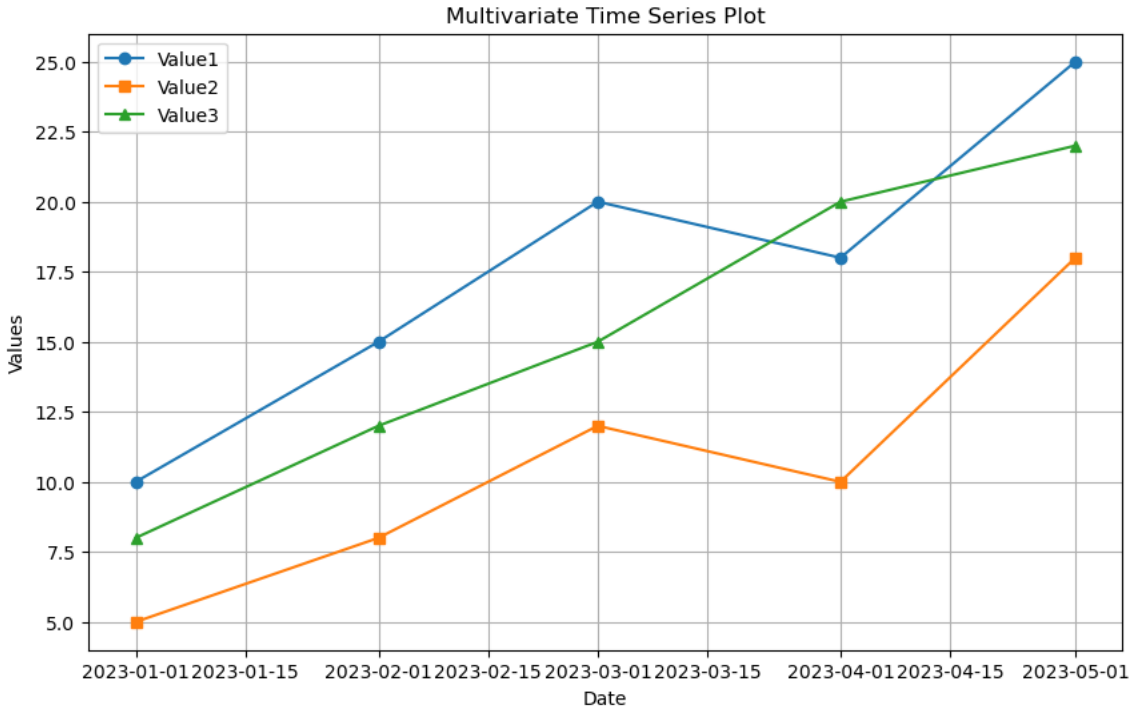
In [33]:

```
import pandas as pd
import matplotlib.pyplot as plt


# Plot the time series data
plt.figure(figsize=(10, 6))
plt.plot(df.index, df['LeagueIndex'], label='GameID', color='blue')
plt.plot(df.index, df['UniqueHotkeys'], label='GameID', color='red')

# Customize the plot
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Bivariate Time Series Analysis')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```



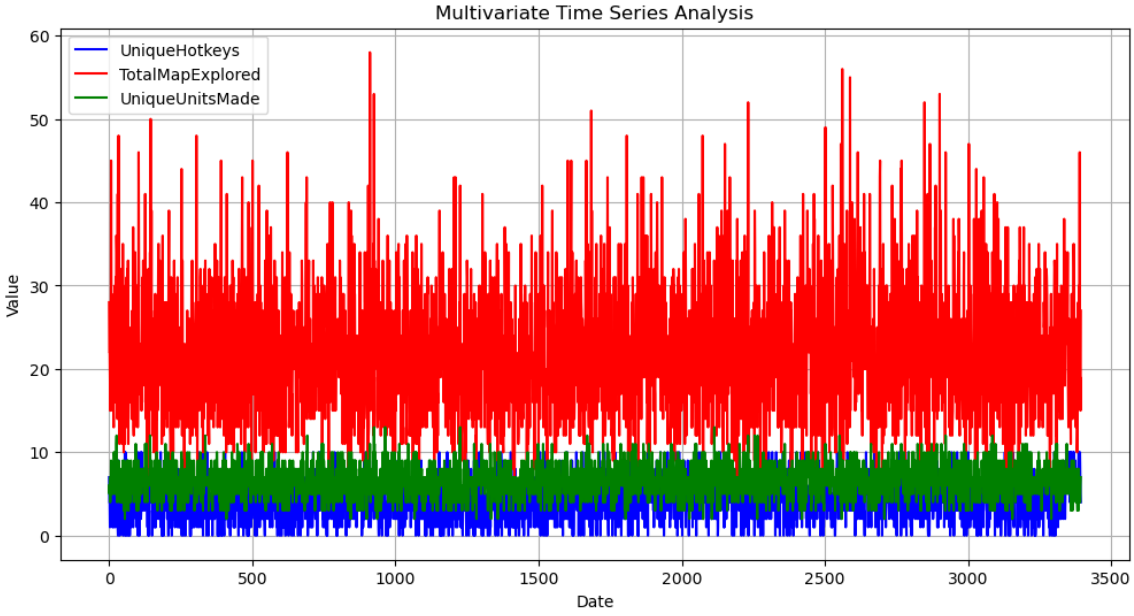
In [34]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data: Date and corresponding values for multiple variables
data = {
    'Date': ['2023-01-01', '2023-02-01', '2023-03-01', '2023-04-01', '2
    'Value1': [10, 15, 20, 18, 25],
    'Value2': [5, 8, 12, 10, 18],
    'Value3': [8, 12, 15, 20, 22]
}

# Create a DataFrame from the sample data
df = pd.DataFrame(data)

# Convert the 'Date' column to a datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Set the 'Date' column as the index of the DataFrame
df.set_index('Date', inplace=True)

# Create a multivariate time series plot
plt.figure(figsize=(10, 6))  # Optional: Set the figure size

# Plot the first variable
plt.plot(df.index, df['Value1'], marker='o', linestyle='-', label='Valu

# Plot the second variable on the same graph
plt.plot(df.index, df['Value2'], marker='s', linestyle='-', label='Valu

# Plot the third variable on the same graph
plt.plot(df.index, df['Value3'], marker='^', linestyle='-', label='Valu

# Optional: Add labels and a legend
plt.xlabel('Date')
plt.ylabel('Values')
plt.title('Multivariate Time Series Plot')
plt.legend()

# Display the plot
plt.grid(True)  # Optional: Add grid lines
plt.show()
```


Multivariate Time Series Plot
Value1
Value2
Value3
Values
Date
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2023-01-01
2023-01-15
2023-02-01
2023-02-15
2023-03-01
2023-03-15
2023-04-01
2023-04-15
2023-05-01

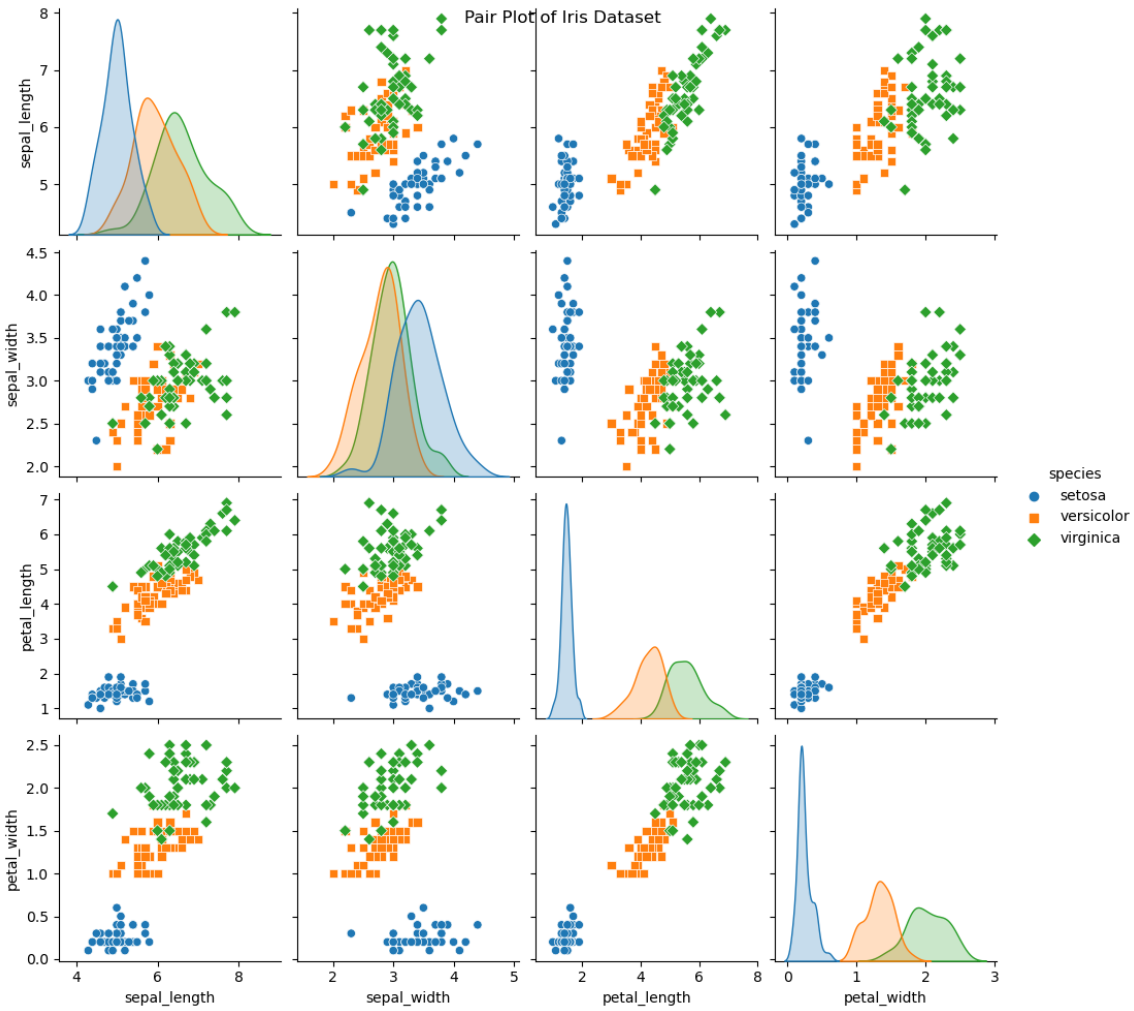
In [41]:

```
import pandas as pd
import matplotlib.pyplot as plt


# Plot the multivariate time series data
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['UniqueHotkeys'], label='UniqueHotkeys', color='b
plt.plot(df.index, df['TotalMapExplored'], label='TotalMapExplored', co
plt.plot(df.index, df['UniqueUnitsMade'], label='UniqueUnitsMade', colo

# Customize the plot
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Multivariate Time Series Analysis')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```



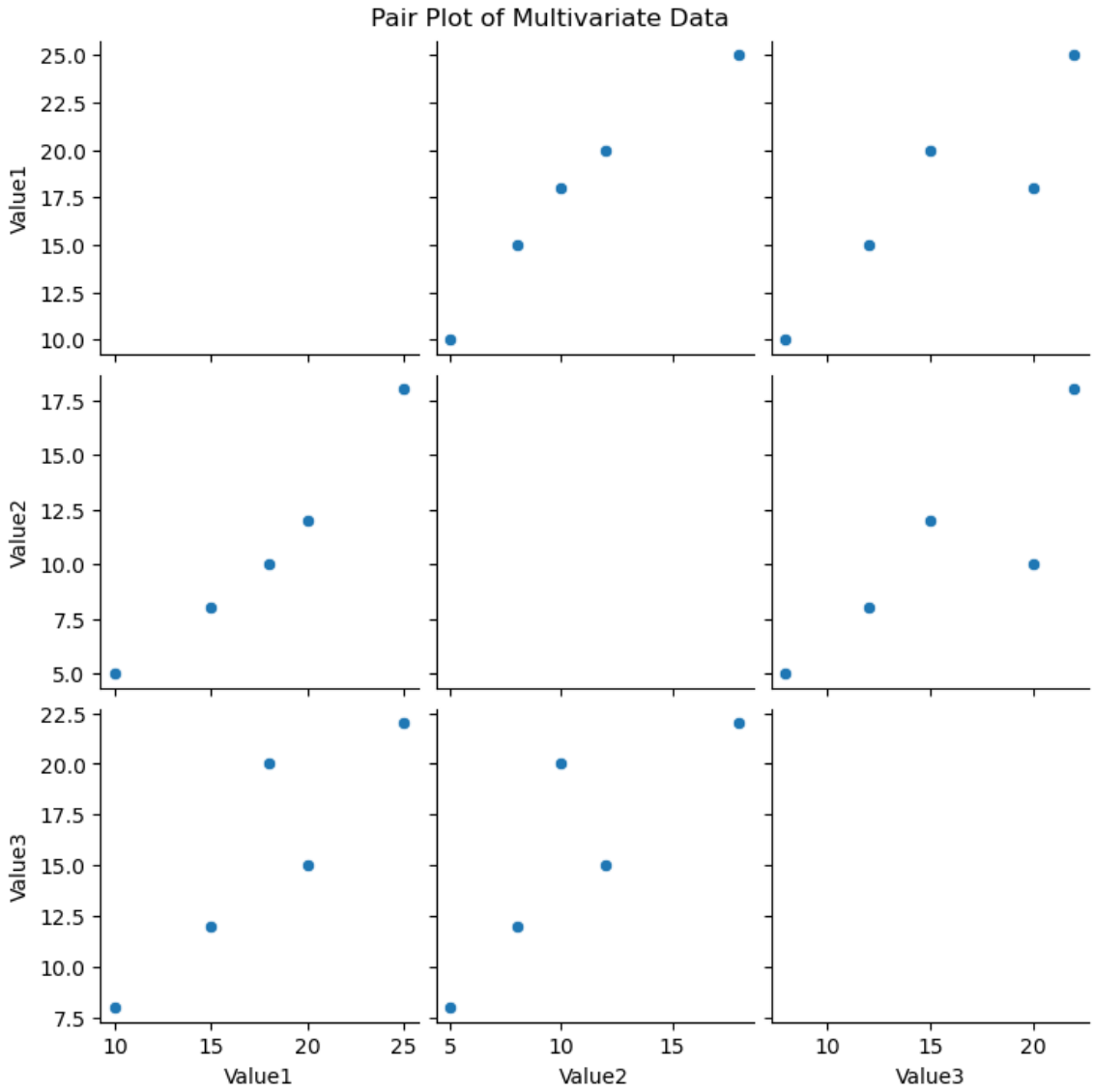
In [35]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load a sample dataset from Seaborn (you can replace this with your ow
iris = sns.load_dataset("iris")

# Create a pair plot
sns.pairplot(iris, hue="species", markers=["o", "s", "D"])

# Optional: Customize plot properties
plt.suptitle("Pair Plot of Iris Dataset")
plt.show()
```



In [36]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Create a pair plot using Seaborn
# Use 'scatter' for individual plots
sns.pairplot(df, diag_kind='UniqueHotkeys')  # 'kde' adds kernel densit

# Customize plot properties (titles, labels, etc.)
plt.suptitle('Pair Plot of Multivariate Data')
plt.subplots_adjust(top=0.95)  # Adjust the top position of the title

# Display the plot
plt.show()
```



In [37]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Extract the variable for which you want to create a distribution plot
variable_to_plot = 'UniqueHotkeys'  # Replace 'ColumnName' with the act

# Create a distribution plot using Seaborn
plt.figure(figsize=(8, 6))  # Optional: Set the figure size
sns.histplot(data=df, x=variable_to_plot, kde=True)  # 'kde=True' adds

# Customize the plot (labels, title, etc.)
plt.xlabel('X-Axis Label')
plt.ylabel('UniqueHotkeys')
plt.title('Distribution Plot of ' + variable_to_plot)

# Show the distribution plot
plt.show()
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_19012\2634274016.py in <module>
      8 # Create a distribution plot using Seaborn
      9 plt.figure(figsize=(8, 6))  # Optional: Set the figure size
---> 10 sns.histplot(data=df, x=variable_to_plot, kde=True)  # 'kde=True'
adds a kernel density estimate plot
     11
     12 # Customize the plot (labels, title, etc.)

~\anaconda3\lib\site-packages\seaborn\distributions.py in histplot(data,
x, y, hue, weights, stat, bins, binwidth, binrange, discrete, cumulative,
common_bins, common_norm, multiple, element, fill, shrink, kde, kde_kws, l
ine_kws, thresh, pthresh, pmax, cbar, cbar_ax, cbar_kws, palette, hue_orde
r, hue_norm, color, log_scale, legend, ax, **kwargs)
   1428 ):
   1429
-> 1430     p = _DistributionPlotter(
   1431         data=data,
   1432         variables=_DistributionPlotter.get_semantics(locals())

~\anaconda3\lib\site-packages\seaborn\distributions.py in __init__(self, d
ata, variables)
    109     ):
    110
--> 111         super().__init__(data=data, variables=variables)
    112
    113     @property

~\anaconda3\lib\site-packages\seaborn\_core.py in __init__(self, data, var
iables)
    603     def __init__(self, data=None, variables={}):
    604
--> 605         self.assign_variables(data, variables)
    606
    607         for var, cls in self._semantic_mappings.items():

~\anaconda3\lib\site-packages\seaborn\_core.py in assign_variables(self, d
ata, variables)
    666         else:
    667             self.input_format = "long"
--> 668             plot_data, variables = self._assign_variables_longform
(
    669                 data, **variables,
    670             )

~\anaconda3\lib\site-packages\seaborn\_core.py in _assign_variables_longfo
rm(self, data, **kwargs)
    901
    902                 err = f"Could not interpret value `{val}` for para
meter `{key}`"
--> 903                 raise ValueError(err)
    904
    905             else:

ValueError: Could not interpret value `UniqueHotkeys` for parameter `x`
```

```
<Figure size 800x600 with 0 Axes>
```

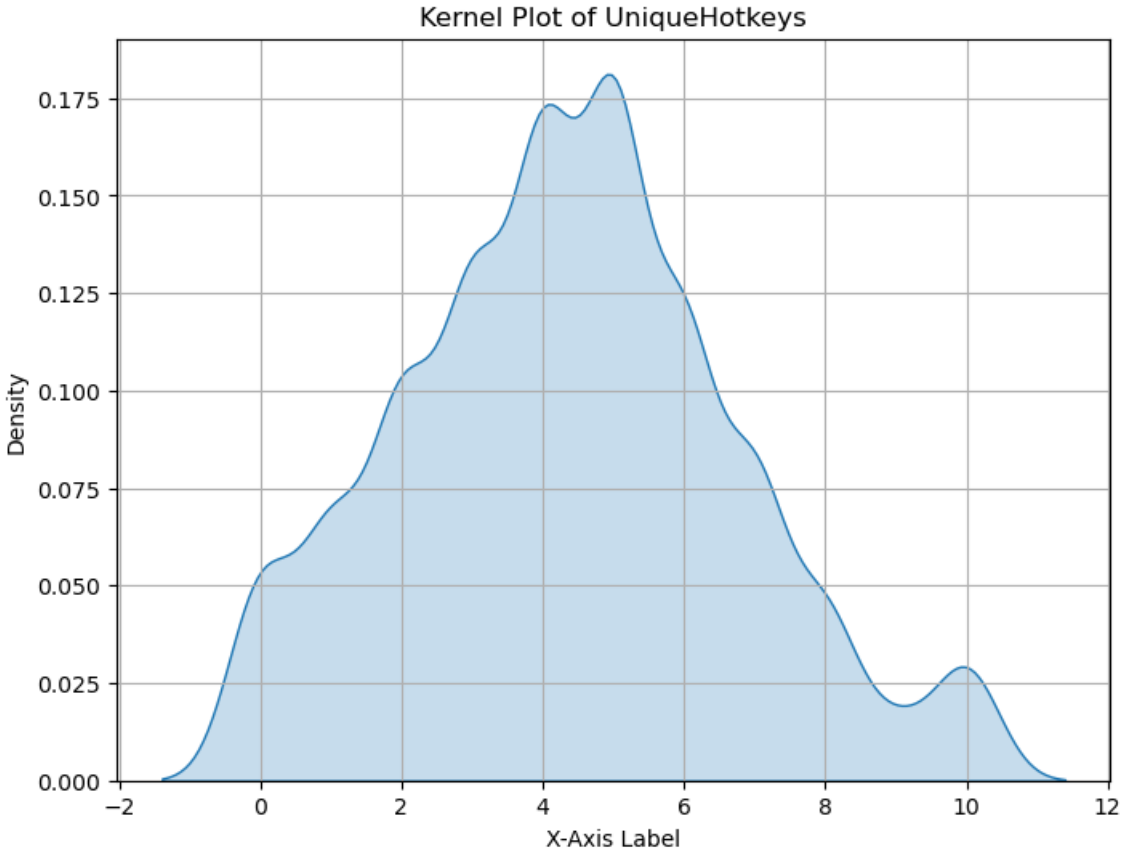
In [38]:

```
import pandas as pd
import matplotlib.pyplot as plt


# Extract the variable for which you want to create a histogram
variable_to_plot = 'UniqueHotkeys'  # Replace 'ColumnName' with the act
  # Replace 'ColumnName' with the actual column name

# Create a histogram
plt.figure(figsize=(8, 6))
plt.hist(df[variable_to_plot], bins=20, edgecolor='k')
plt.xlabel('X-Axis Label')
plt.ylabel('Frequency')
plt.title('Histogram of ' + variable_to_plot)
plt.grid(True)
plt.show()
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self,
key, method, tolerance)
   3628             try:
-> 3629                 return self._engine.get_loc(casted_key)
   3630             except KeyError as err:

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.inde
x.IndexEngine.get_loc()

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.inde
x.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObject
HashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObject
HashTable.get_item()

KeyError: 'UniqueHotkeys'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_19012\4018736219.py in <module>
      9 # Create a histogram
     10 plt.figure(figsize=(8, 6))
---> 11 plt.hist(df[variable_to_plot], bins=20, edgecolor='k')
     12 plt.xlabel('X-Axis Label')
     13 plt.ylabel('Frequency')

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, ke
y)
   3503             if self.columns.nlevels > 1:
   3504                 return self._getitem_multilevel(key)
-> 3505             indexer = self.columns.get_loc(key)
   3506             if is_integer(indexer):
   3507                 indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self,
key, method, tolerance)
   3629                 return self._engine.get_loc(casted_key)
   3630             except KeyError as err:
-> 3631                 raise KeyError(key) from err
   3632             except TypeError:
   3633                 # If we have a listlike key, _check_indexing_error
will raise

KeyError: 'UniqueHotkeys'


<Figure size 800x600 with 0 Axes>
```

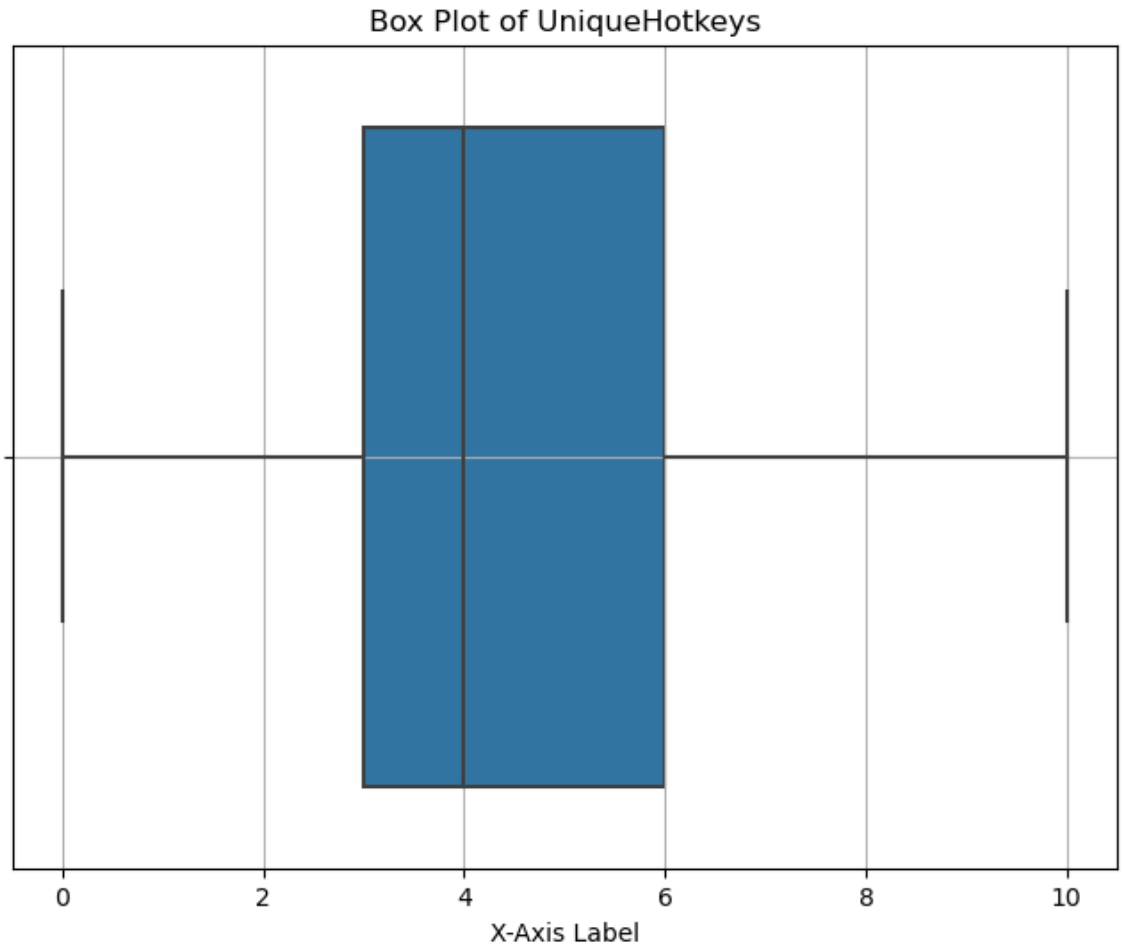
In [9]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Extract the variable for which you want to create a KDE plot
variable_to_plot = 'UniqueHotkeys'  # Replace 'ColumnName' with the act

# Create a KDE plot
plt.figure(figsize=(8, 6))
sns.kdeplot(df[variable_to_plot], shade=True)
plt.xlabel('X-Axis Label')
plt.ylabel('Density')
plt.title('Kernel Plot of ' + variable_to_plot)
plt.grid(True)
plt.show()
```



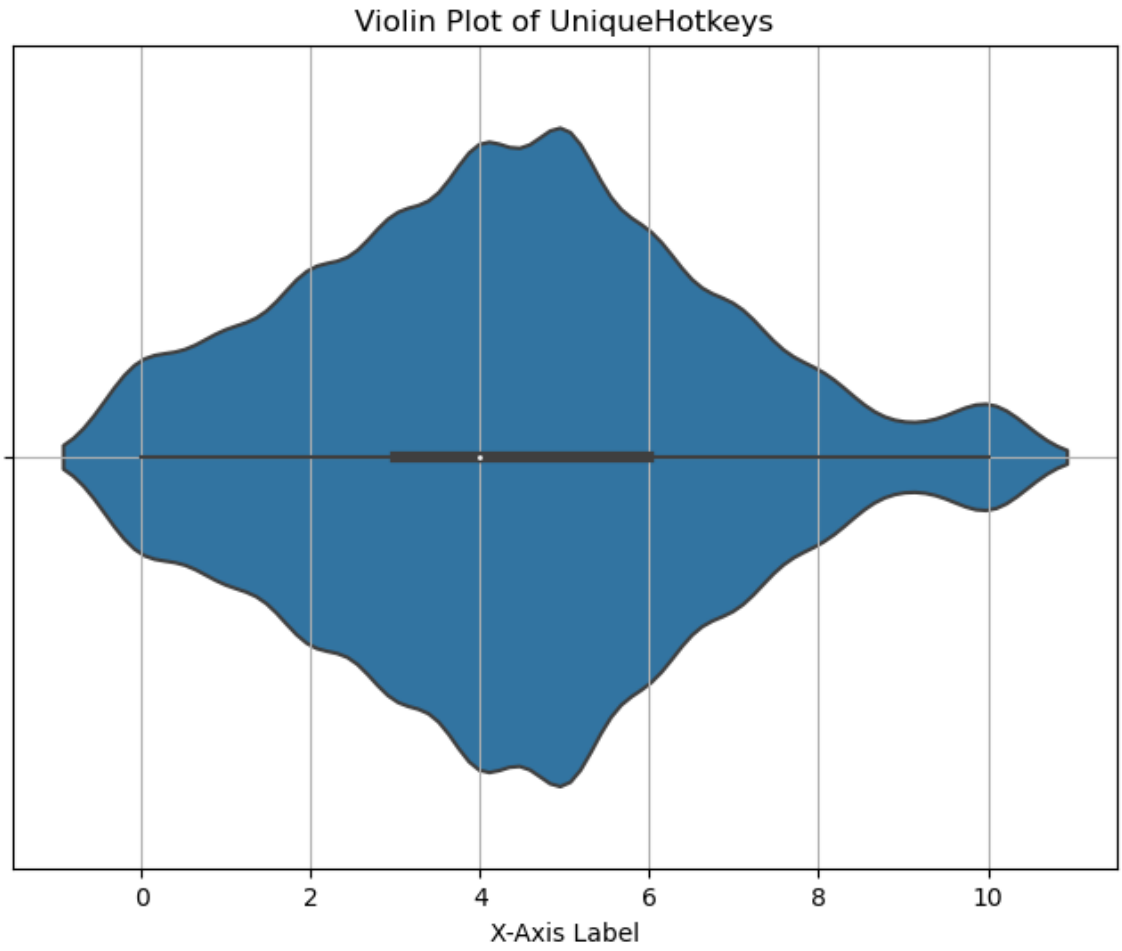
In [8]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a box plot
plt.figure(figsize=(8, 6))
sns.boxplot(x='UniqueHotkeys', data=df)
plt.xlabel('X-Axis Label')
plt.title('Box Plot of ' + variable_to_plot)
plt.grid(True)
plt.show()
```



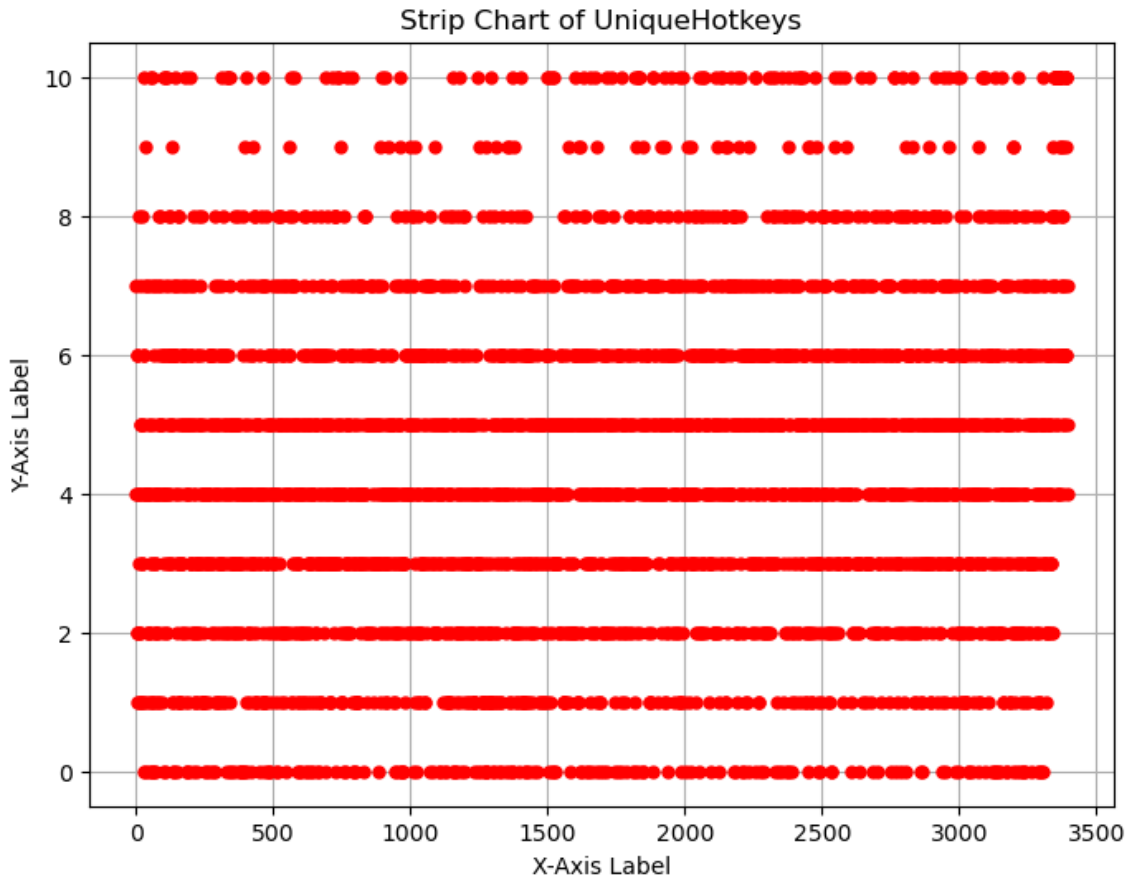
In [11]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a violin plot
plt.figure(figsize=(8, 6))
sns.violinplot(x='UniqueHotkeys', data=df)
plt.xlabel('X-Axis Label')
plt.title('Violin Plot of ' + variable_to_plot)
plt.grid(True)
plt.show()
```



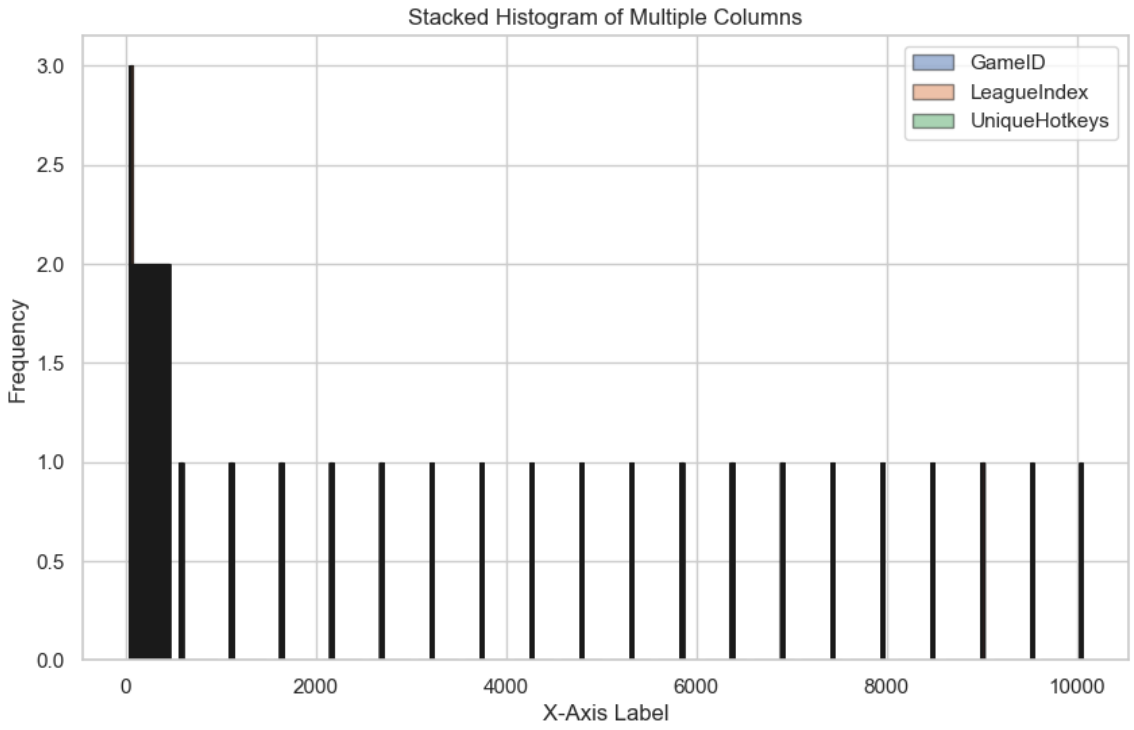
In [12]:

```
 import pandas as pd
import matplotlib.pyplot as plt
# Extract the variable for the strip chart
variable_to_plot = 'UniqueHotkeys'  # Replace 'ColumnName' with the act

# Create a strip chart
plt.figure(figsize=(8, 6))
plt.plot(df[variable_to_plot], 'ro', markersize=5)  # 'ro' means red ci
plt.xlabel('X-Axis Label')
plt.ylabel('Y-Axis Label')
plt.title('Strip Chart of ' + variable_to_plot)
plt.grid(True)
plt.show()
```



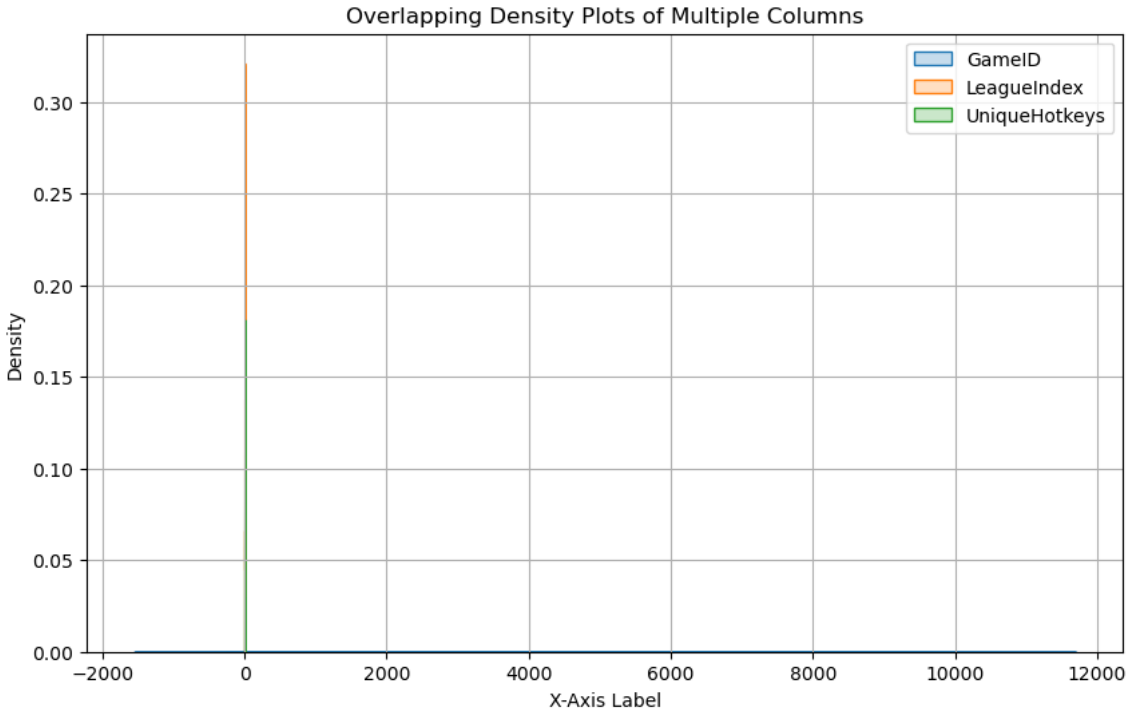
In [20]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Define the columns you want to create a stacked histogram for
columns_to_plot = ['GameID', 'LeagueIndex', 'UniqueHotkeys']  # Replace

# Create a stacked histogram
plt.figure(figsize=(10, 6))
plt.hist(df[columns_to_plot].values.T, bins=20, edgecolor='k', alpha=0.
plt.xlabel('X-Axis Label')
plt.ylabel('Frequency')
plt.title('Stacked Histogram of Multiple Columns')
plt.legend()
plt.grid(True)
plt.show()
```



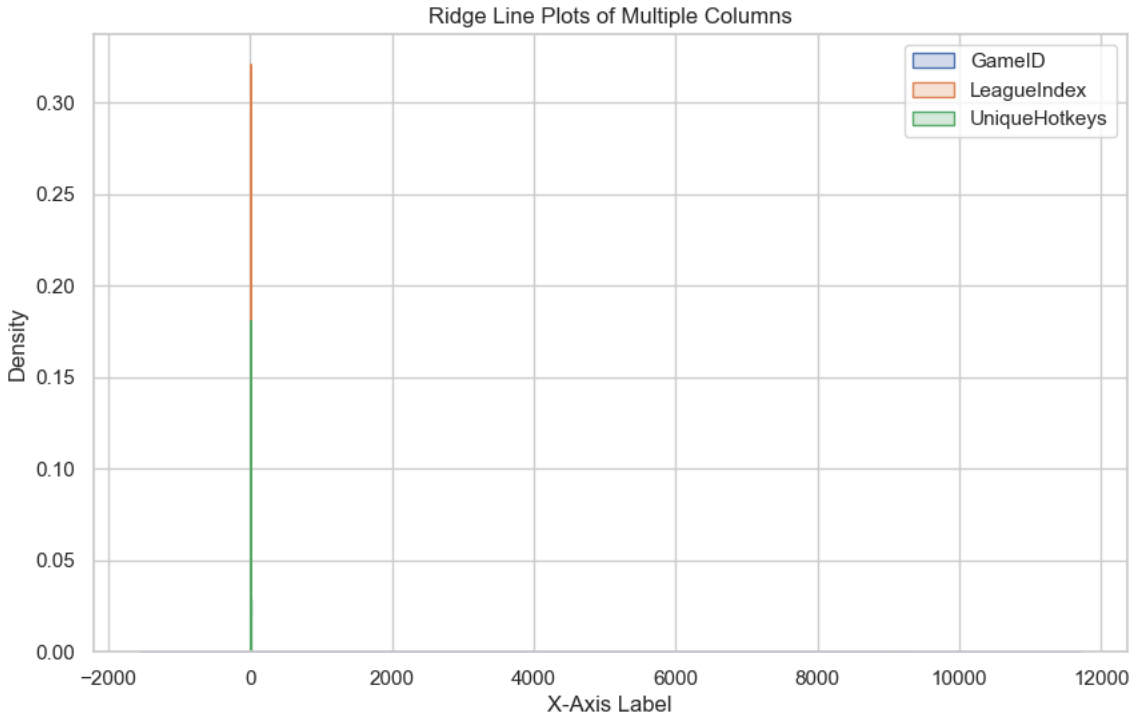
In [17]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Define the columns you want to create overlapping density plots for
columns_to_plot = ['GameID', 'LeagueIndex', 'UniqueHotkeys']  # Replace

# Create overlapping density plots
plt.figure(figsize=(10, 6))
for column in columns_to_plot:
    sns.kdeplot(df[column], label=column, shade=True)

plt.xlabel('X-Axis Label')
plt.ylabel('Density')
plt.title('Overlapping Density Plots of Multiple Columns')
plt.legend()
plt.grid(True)
plt.show()
```



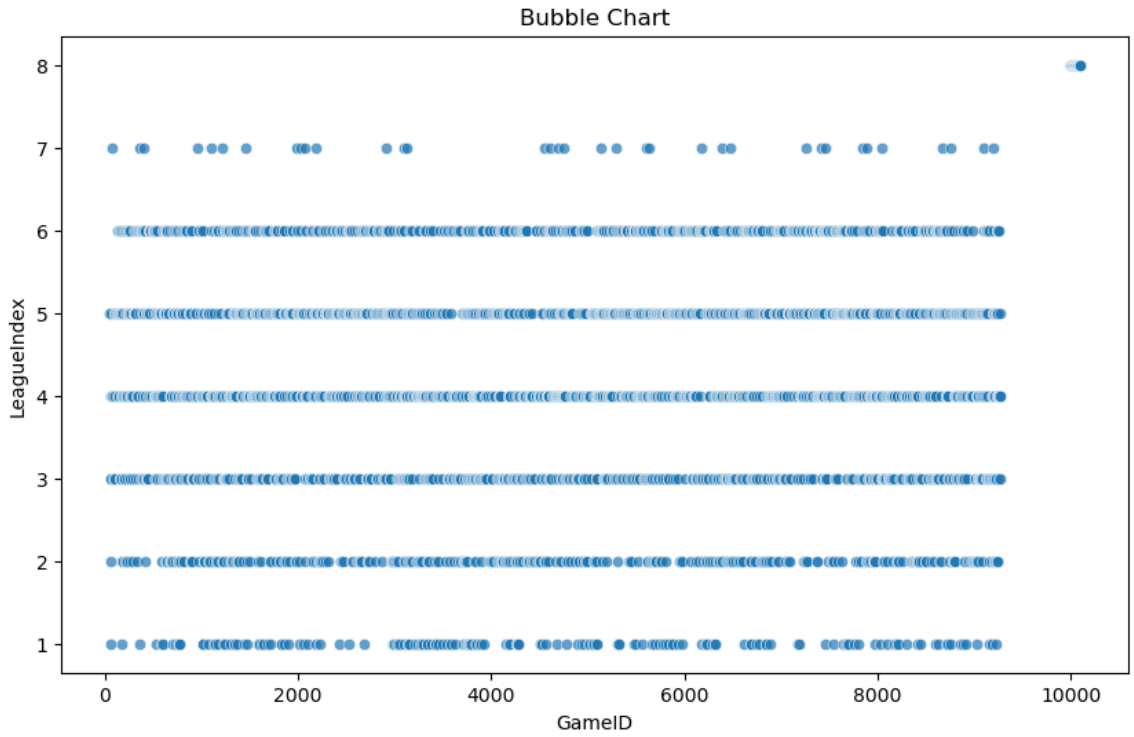
In [18]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Define the columns you want to create ridge line plots for
columns_to_plot = ['GameID', 'LeagueIndex', 'UniqueHotkeys']  # Replace

# Create ridge line plots
plt.figure(figsize=(10, 6))
sns.set(style='whitegrid')
for column in columns_to_plot:
    sns.kdeplot(df[column], label=column, shade=True)

plt.xlabel('X-Axis Label')
plt.ylabel('Density')
plt.title('Ridge Line Plots of Multiple Columns')
plt.legend()
plt.grid(True)
plt.show()
```



In [6]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Define the columns for x, y, and bubble size
x_column = 'GameID'  # Replace with the actual column name
y_column = 'LeagueIndex'  # Replace with the actual column name
size_column = 'Size_Column'  # Replace with the actual column name

# Create a bubble chart
plt.figure(figsize=(10, 6))
sns.scatterplot(x=x_column, y=y_column, data=df, sizes=(20, 200), alpha

# Customize the plot properties (title, labels, etc.)
plt.title('Bubble Chart')
plt.xlabel('GameID')
plt.ylabel('LeagueIndex')

# Show the bubble chart
plt.show()
```



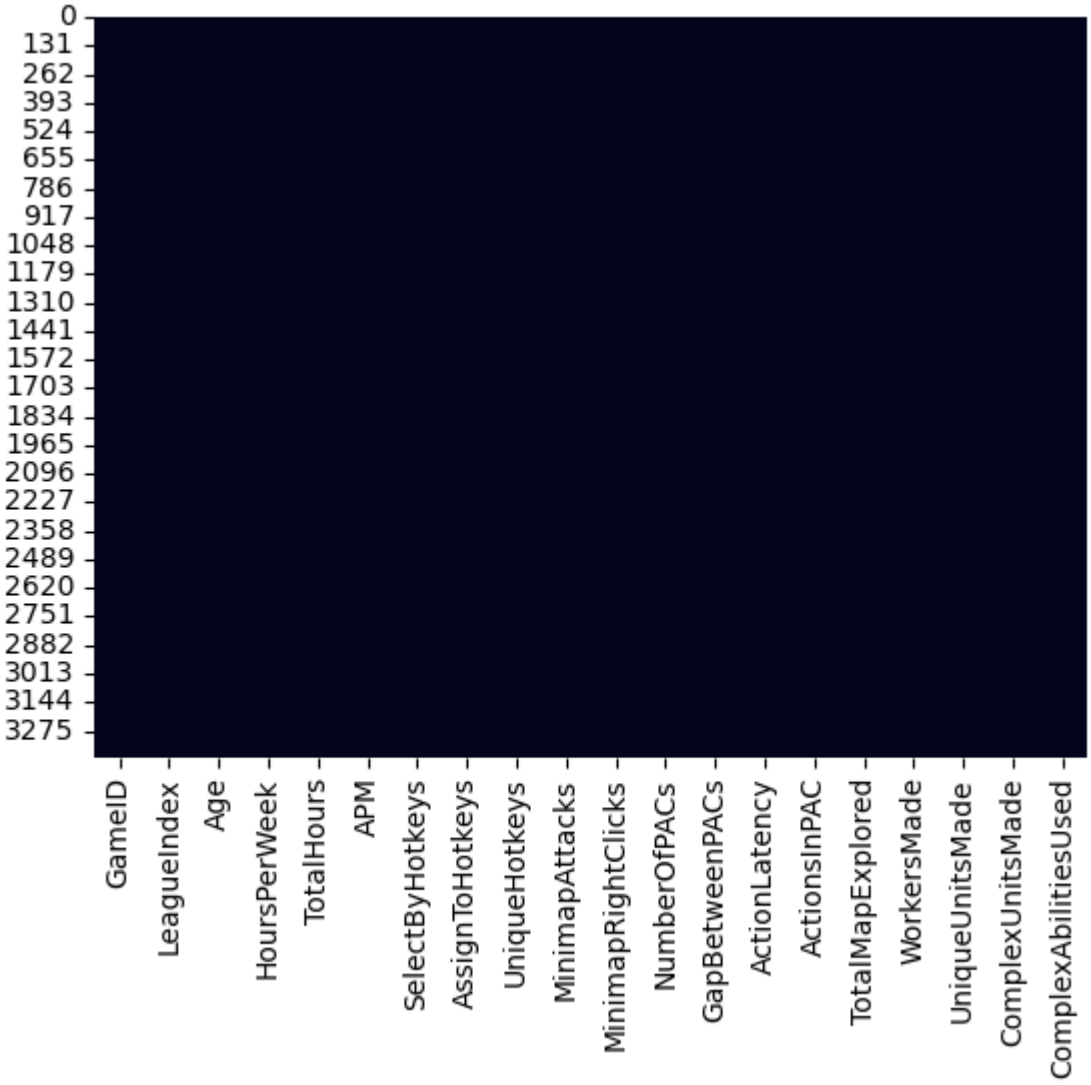
In [2]:

```
 import pandas as pd


# Identify missing values
missing_data = df.isnull()

# Count missing values per column
missing_count = missing_data.sum()

# Count missing values per row
missing_count_per_row = missing_data.sum(axis=1)

# Print or analyze the missing data
print("Missing Data Count per Column:")
print(missing_count)

print("Missing Data Count per Row:")
print(missing_count_per_row)
```

```
Missing Data Count per Column:
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
Missing Data Count per Row:
0       0
1       0
2       0
3       0
4       0
       ..
3390    0
3391    0
3392    0
3393    0
3394    0
Length: 3395, dtype: int64
```

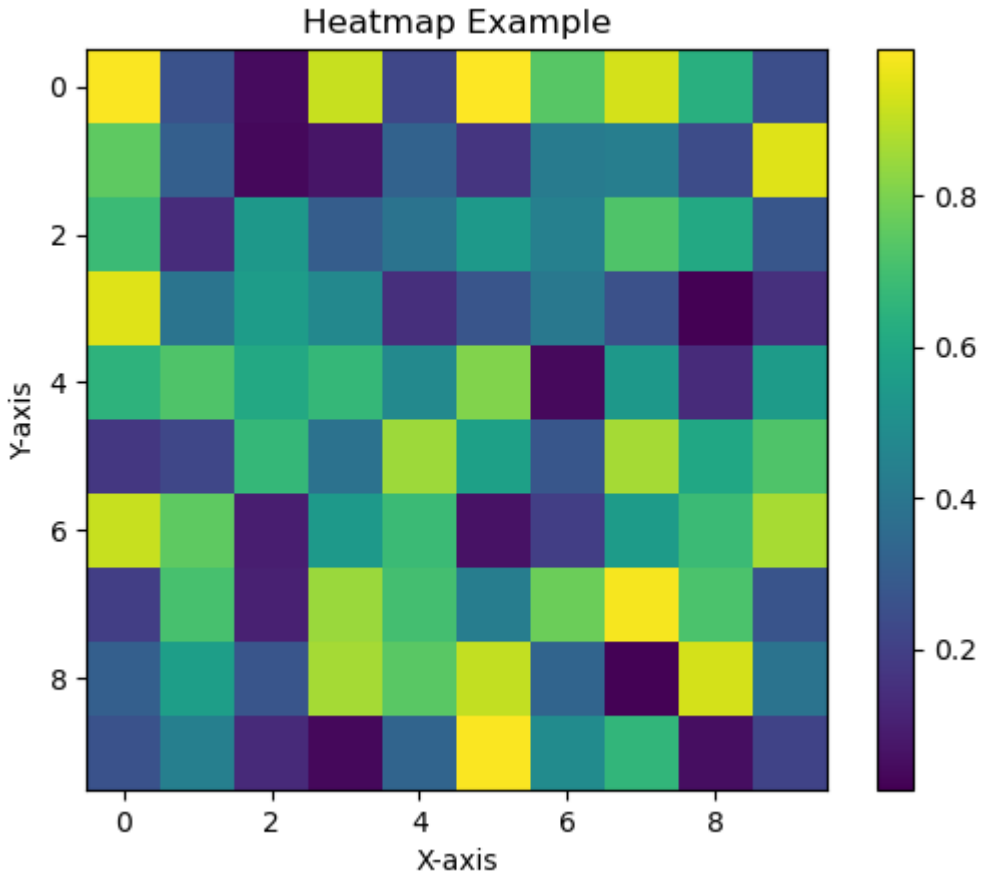
In [9]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv(r"C:\Users\Anusha V\Downloads\skillcraft1+master+table
sns.heatmap(df.isnull(),cbar=False)

```

Out[9]: <AxesSubplot:>



In [6]:

```
import numpy as np
import matplotlib.pyplot as plt

# Sample data for the heatmap
data = np.random.random((10, 10))  # Replace this with your own data

# Create a heatmap
plt.imshow(data, cmap='viridis')  # You can choose a different colormap
plt.colorbar()  # Add a colorbar for reference

plt.title('Heatmap Example')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Show the heatmap
plt.show()
```



In [12]:
```
d_f = df.fillna(5)
d_f
```

Out[12]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	As
0	52	5	27	10	3000	143.7180	0.003515	
1	55	5	23	10	5000	129.2322	0.003304	
2	56	4	30	10	200	69.9612	0.001101	
3	57	3	19	20	400	107.6016	0.001034	
4	58	3	32	10	500	122.8908	0.001136	
...	...	...	...	...	...	...	...	
3390	10089	8	?	?	?	259.6296	0.020425	
3391	10090	8	?	?	?	314.6700	0.028043	
3392	10092	8	?	?	?	299.4282	0.028341	
3393	10094	8	?	?	?	375.8664	0.036436	
3394	10095	8	?	?	?	348.3576	0.029855	

3395 rows × 20 columns

In [16]:
```
d_f = df.dropna(0)
d_f
```

```
C:\Users\Anusha V\AppData\Local\Temp\ipykernel_3584\1675708418.py:1: FutureWarning: In a future version of pandas all arguments of DataFrame.dropna will be keyword-only.
  d_f = df.dropna(0)
```

Out[16]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	As
0	52	5	27	10	3000	143.7180	0.003515	
1	55	5	23	10	5000	129.2322	0.003304	
2	56	4	30	10	200	69.9612	0.001101	
3	57	3	19	20	400	107.6016	0.001034	
4	58	3	32	10	500	122.8908	0.001136	
...	...	...	...	...	...	...	...	
3390	10089	8	?	?	?	259.6296	0.020425	
3391	10090	8	?	?	?	314.6700	0.028043	
3392	10092	8	?	?	?	299.4282	0.028341	
3393	10094	8	?	?	?	375.8664	0.036436	
3394	10095	8	?	?	?	348.3576	0.029855	

3395 rows × 20 columns

In [17]:
```
1 df.info
```

Out[17]:
```
<bound method DataFrame.info of       GameID  LeagueIndex Age HoursPerWeek
TotalHours       APM  \
0         52            5  27           10       3000  143.7180
1         55            5  23           10       5000  129.2322
2         56            4  30           10        200   69.9612
3         57            3  19           20        400  107.6016
4         58            3  32           10        500  122.8908
...      ...          ...  ..          ...        ...       ...
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  \
0            0.003515         0.000220              7        0.000110
1            0.003304         0.000259              4        0.000294
2            0.001101         0.000336              4        0.000294
3            0.001034         0.000213              1        0.000053
4            0.001136         0.000327              2        0.000000
...               ...              ...            ...             ...
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  \
0               0.000392      0.004849         32.6677        40.8673
1               0.000432      0.004307         32.9194        42.3454
2               0.000461      0.002926         44.6475        75.3548
3               0.000543      0.003783         29.2203        53.7352
4               0.001329      0.002368         22.6885        62.0813
...                  ...           ...             ...            ...
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  \
0           4.7508                28     0.001397                6
1           4.8434                22     0.001193                5
2           4.0430                22     0.000745                6
3           4.9155                19     0.000426                7
4           9.3740                15     0.001174                4
...            ...               ...          ...              ...
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
0             0.000000              0.000000
1             0.000000              0.000208
2             0.000000              0.000189
3             0.000000              0.000384
4             0.000000              0.000019
...                ...                   ...
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

[3395 rows x 20 columns]>
```

In [18]:
```
1 df.isnull()
```

Out[18]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	Assig
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
3390	False	False	False	False	False	False	False	
3391	False	False	False	False	False	False	False	
3392	False	False	False	False	False	False	False	
3393	False	False	False	False	False	False	False	
3394	False	False	False	False	False	False	False	

3395 rows × 20 columns

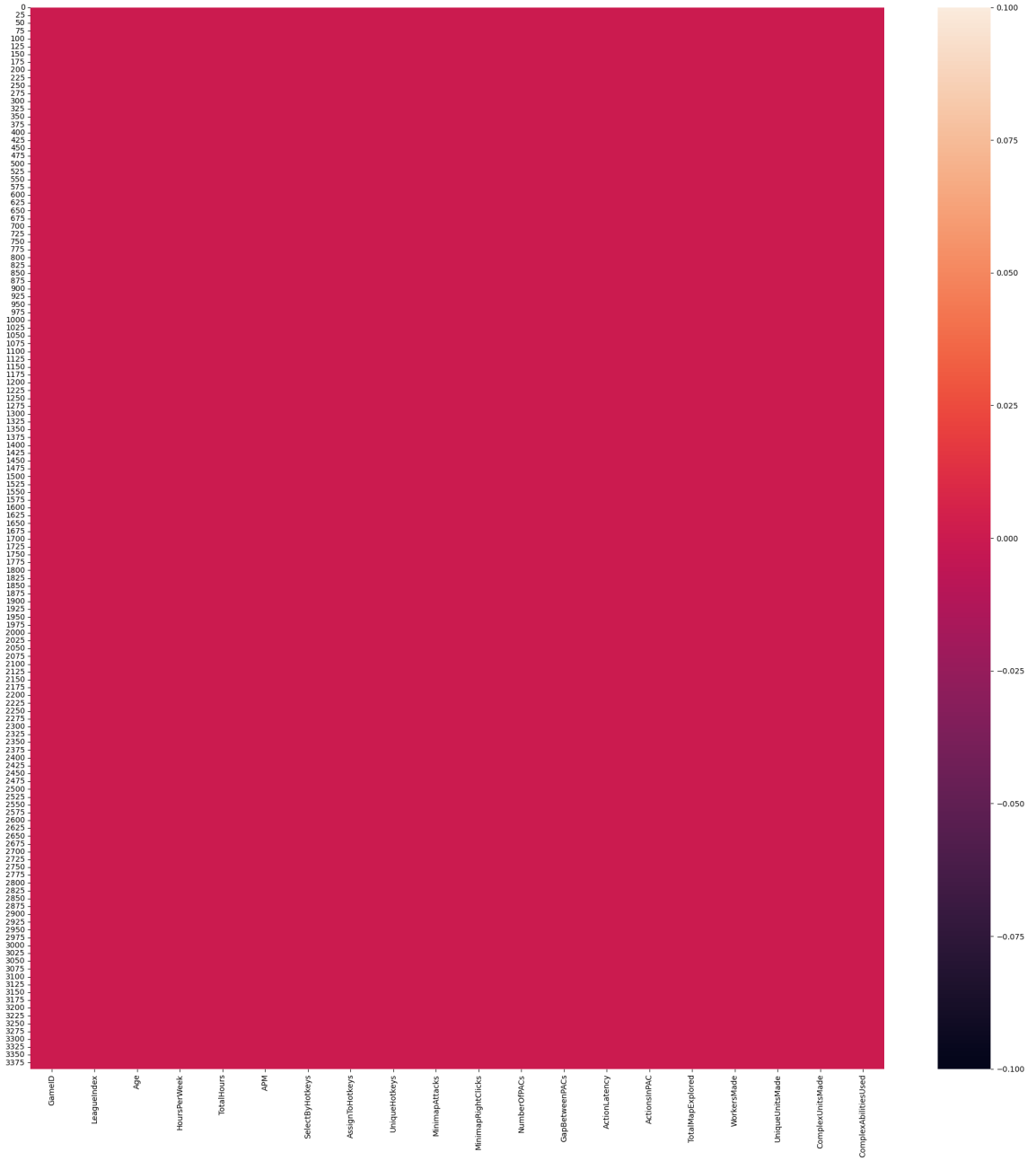
In [19]:
```
1 df.isnull().sum
```

Out[19]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of       GameID
LeagueIndex    Age  HoursPerWeek  TotalHours    APM  \
0      False        False  False         False       False  False
1      False        False  False         False       False  False
2      False        False  False         False       False  False
3      False        False  False         False       False  False
4      False        False  False         False       False  False
...      ...          ...    ...           ...         ...    ...
3390   False        False  False         False       False  False
3391   False        False  False         False       False  False
3392   False        False  False         False       False  False
3393   False        False  False         False       False  False
3394   False        False  False         False       False  False

      SelectByHotkeys  AssignToHotkeys  UniqueHotkeys  MinimapAttacks  \
0               False            False          False           False
1               False            False          False           False
2               False            False          False           False
3               False            False          False           False
4               False            False          False           False
...               ...              ...            ...             ...
3390            False            False          False           False
3391            False            False          False           False
3392            False            False          False           False
3393            False            False          False           False
3394            False            False          False           False

      MinimapRightClicks  NumberOfPACs  GapBetweenPACs  ActionLatency  \
0                  False         False           False          False
1                  False         False           False          False
2                  False         False           False          False
3                  False         False           False          False
4                  False         False           False          False
...                  ...           ...             ...            ...
3390               False         False           False          False
3391               False         False           False          False
3392               False         False           False          False
3393               False         False           False          False
3394               False         False           False          False

      ActionsInPAC  TotalMapExplored  WorkersMade  UniqueUnitsMade  \
0            False             False        False            False
1            False             False        False            False
2            False             False        False            False
3            False             False        False            False
4            False             False        False            False
...            ...               ...          ...              ...
3390         False             False        False            False
3391         False             False        False            False
3392         False             False        False            False
3393         False             False        False            False
3394         False             False        False            False

      ComplexUnitsMade  ComplexAbilitiesUsed
0                False                 False
1                False                 False
2                False                 False
3                False                 False
4                False                 False
...                ...                   ...
3390             False                 False
```

```
3391            False                False
3392            False                False
3393            False                False
3394            False                False

[3395 rows x 20 columns]>
```

In [20]:
```
1 plt.figure(figsize=(25, 25))
2 sns.heatmap(df.isnull())
```

Out[20]: <AxesSubplot:>



In [24]:
```
1 df.isnull().sum()/df.shape[0] = 100
```

```
  File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_3584\1699131357.p
y", line 1
    df.isnull().sum()/df.shape[0] = 100
    ^
SyntaxError: cannot assign to operator
```

In [36]:
```
1 drop_columns = df['GameID' > 200].keys()
2 deop_columns
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_3584\1647178008.py in <module>
----> 1 drop_columns = df['GameID' > 200].keys()
      2 deop_columns

TypeError: '>' not supported between instances of 'str' and 'int'
```

In [ ]:
```
1
```