In [ ]:

```
# 1) Football is one of the few things I am passionate about, hence pl
##     a fun activity!

# 2) Like diwali_sales_analysis_P1 I have continued analysing the data
##   making inference on relationship  between two columns in a datset
```

In [4]:

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as mt
%matplotlib inLine
```

In [5]:

```
df = pd.read_csv('/Users/kaustubhchati/Desktop/player_valuation_projec
# uploading player_valuation data
#'df' variable represents an abbreviation for DataFrame
```

In [8]:

```
df.head(20)
```

Out[8]:

	player_id	date	market_value_in_eur	current_club_id	player_club_domestic_competition_id
0	405973	2000-01-20	150000	3057	BE1
1	342216	2001-07-20	100000	1241	SC1
2	3132	2003-12-09	400000	126	TR1
3	6893	2003-12-15	900000	984	GB1
4	10	2004-10-04	7000000	398	IT1
5	26	2004-10-04	1500000	16	L1
6	65	2004-10-04	8000000	1091	GR1
7	77	2004-10-04	13000000	506	IT1
8	80	2004-10-04	400000	27	L1
9	109	2004-10-04	9500000	825	TR1
10	123	2004-	9500000	33	L1

		10-04			
11	132	2004-10-04	13000000	11	GB1
12	162	2004-10-04	1250000	79	L1
13	215	2004-10-04	7500000	1084	ES1
14	264	2004-10-04	250000	79	L1
15	276	2004-10-04	250000	20100	DK1
16	277	2004-10-04	1250000	3368	ES1
17	299	2004-10-04	9000000	10484	TR1
18	325	2004-10-04	8000000	141	TR1
19	332	2004-10-04	250000	24	L1

In [7]:

```
df.info
```

Out[7]:

```
<bound method DataFrame.info of         player_id        date  market
_value_in_eur  current_club_id  \
0           405973  2000-01-20               150000             3057
1           342216  2001-07-20               100000             1241
2             3132  2003-12-09               400000              126
3             6893  2003-12-15               900000              984
4               10  2004-10-04              7000000              398
...            ...         ...                  ...              ...
478171      493003  2024-07-19             23000000             1184
478172      502842  2024-07-19              1800000            10690
478173      568005  2024-07-19              7000000             1237
478174      661145  2024-07-19              5000000              681
478175      676318  2024-07-19             10000000            16795

       player_club_domestic_competition_id
0                                      BE1
1                                      SC1
2                                      TR1
3                                      GB1
4                                      IT1
...                                    ...
478171                                 BE1
478172                                UKR1
478173                                 GB1
478174                                 ES1
478175                                 ES1

[478176 rows x 5 columns]>
```

In [9]:

```
pd.isnull(df)
```

Out[9]:

	player_id	date	market_value_in_eur	current_club_id	player_club_domestic_competitior
0	False	False	False	False	Fa
1	False	False	False	False	Fa
2	False	False	False	False	Fa
3	False	False	False	False	Fa
4	False	False	False	False	Fa
...	...	...	...	...	
478171	False	False	False	False	Fa
478172	False	False	False	False	Fa
478173	False	False	False	False	Fa
478174	False	False	False	False	Fa
478175	False	False	False	False	Fa

478176 rows × 5 columns

In [10]:

```
pd.isnull(df).sum()
```

Out[10]:

```
player_id                            0
date                                 0
market_value_in_eur                  0
current_club_id                      0
player_club_domestic_competition_id  0
dtype: int64
```

In [11]:

```
# ---> NO null values present in the dataset
```

In [25]:

```
df['market_value_in_eur'].describe()
```

Out[25]:

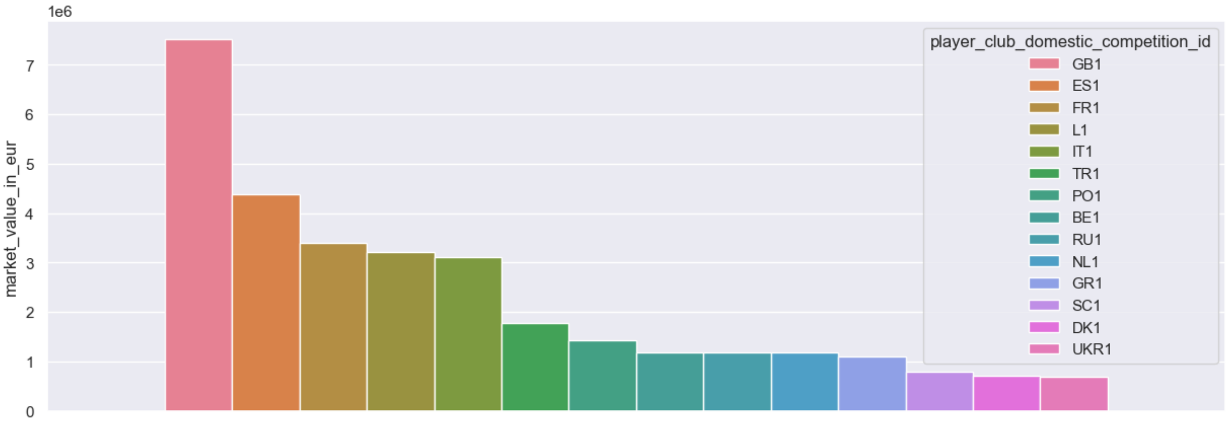
```
count    4.781760e+05
mean     2.410055e+06
std      6.814636e+06
min      0.000000e+00
25%      2.000000e+05
50%      5.000000e+05
75%      1.600000e+06
max      2.000000e+08
Name: market_value_in_eur, dtype: float64
```

In [26]:
```
compwise_data = df.groupby(['player_club_domestic_competition_id'], as
print(compwise_data)
```

```
   player_club_domestic_competition_id  market_value_in_eur
4                                  GB1         7.526280e+06
2                                  ES1         4.391358e+06
3                                  FR1         3.401354e+06
7                                   L1         3.209906e+06
6                                  IT1         3.116126e+06
12                                 TR1         1.766288e+06
9                                  PO1         1.420791e+06
0                                  BE1         1.186822e+06
10                                 RU1         1.181048e+06
8                                  NL1         1.180902e+06
5                                  GR1         1.102735e+06
11                                 SC1         7.875686e+05
1                                  DK1         7.101405e+05
13                                UKR1         6.814589e+05
```

In [58]:
```
sb.set(rc={'figure.figsize':(15,5)})
sb.barplot(data = compwise_data, hue= 'player_club_domestic_competitio
```

Out[58]:
```
<Axes: ylabel='market_value_in_eur'>
```

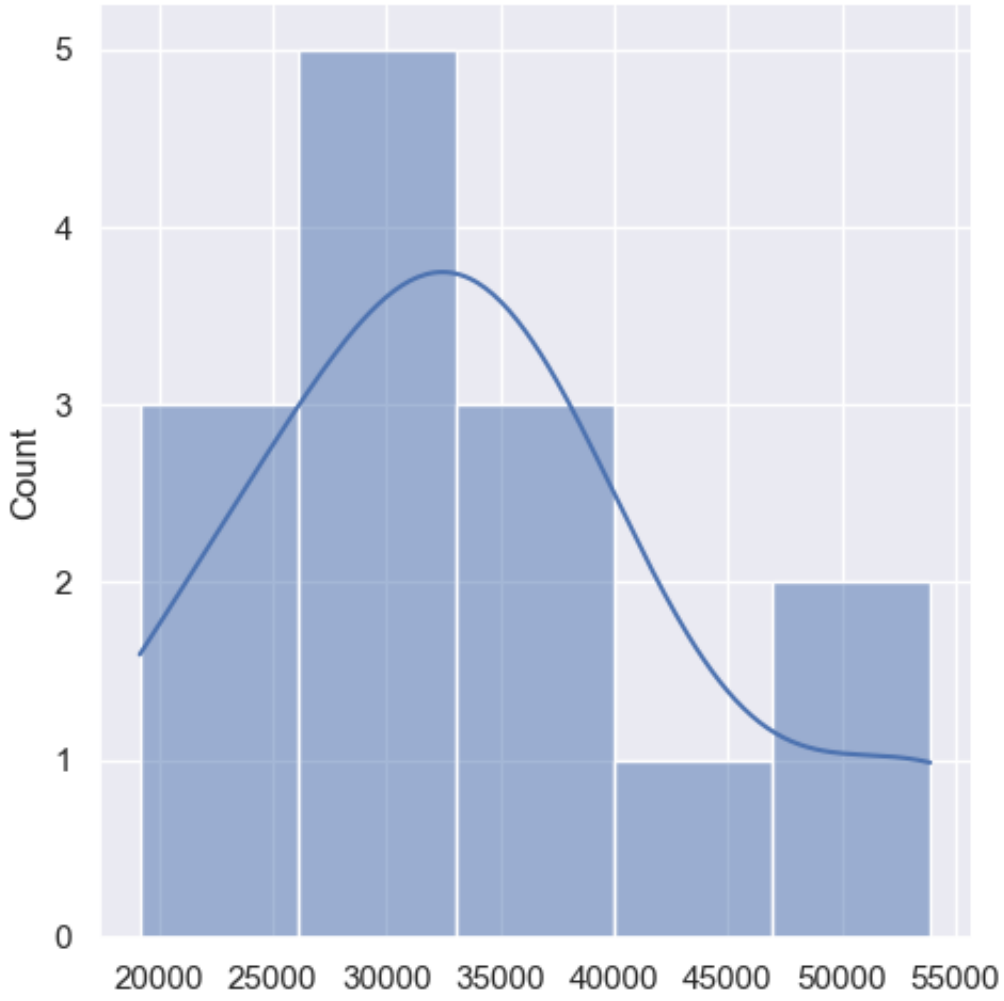


In [24]:
```
# ---> It is observed that  players from club competion:GB1 have signi
#      compared to players from other club competitions.
```

```
In [66]: comp_data = df.groupby('player_club_domestic_competition_id').size()
         mt.figure(figsize=(10, 30))
         sb.displot(data= comp_data,kde=True)
```

```
Out[66]: <seaborn.axisgrid.FacetGrid at 0x11777d010>

         <Figure size 1000x3000 with 0 Axes>
```



In [64]:

```
comp_data = df.groupby('player_club_domestic_competition_id').size()
print(comp_data)
```

```
player_club_domestic_competition_id
BE1     29216
DK1     23134
ES1     40408
FR1     31937
GB1     31923
GR1     38380
IT1     53677
L1      36361
NL1     29590
PO1     32850
RU1     35056
SC1     19142
TR1     53890
UKR1    22612
dtype: int64
```

In [55]:

```
#  ---> The competition sample data is approximately normal (slightly 
##      therefore the inference about market valuation based on compet

# ==> Here I analysed the competition population sample distribution, 
##    and experimented using 'distplot' for plotting the histogram set
###   for Kernel Density Estimation to get an idea of the normality of
```

In [49]:
```
clubwise_data = df.groupby(['current_club_id'], as_index=False)['market
clubwise_data.head(20)
```

Out[49]:

	current_club_id	market_value_in_eur
128	583	26429474999
45	131	26044950000
180	985	24664749999
106	418	23329450000
15	27	22693499999
7	13	22583125000
75	281	20523830000
119	506	17942975000
27	46	17303645000
17	31	17204675000
5	11	16394650000
138	631	16163400000
6	12	15647980000
50	148	14257000000
101	405	13897650000
48	141	12755615000
2	5	12731035000
90	368	11714550000
9	16	11295900000
146	683	10817025001

In [54]:
```
# ---> Club with club_id 583 has the most valuable set of players i.e.
##     by club_id 131 and 985.

#==> Here I recognized the club with highest net market value using th
```

In [53]:
```
best_data = df.groupby(['current_club_id','player_id'], as_index=False
print(best_data)
```

```
       current_club_id  player_id  market_value_in_eur
9960               583     342229            200000000
8232               418     581678            180000000
5716               281     418560            180000000
9941               583      68290            180000000
9939               583      28003            180000000
...                ...        ...                  ...
2969               126     667428                10000
7924               415     237466                10000
7888               410     942099                10000
17038             1245     667966                10000
30052            83678    1143804                10000

[30053 rows x 3 columns]
```

In [ ]:
```
# -->Club with club_id 583 has the most valuable player in the sample

# ==> Here I experimented with 'groupby' method with multiple groups a
##       more specifically the player of the club with the highest mar
```

In [ ]:
```
```