```python
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/ucl-202122-uefa-champions-league/defending.csv
/kaggle/input/ucl-202122-uefa-champions-league/key_stats.csv
/kaggle/input/ucl-202122-uefa-champions-league/attacking.csv
/kaggle/input/ucl-202122-uefa-champions-league/goalkeeping.csv
/kaggle/input/ucl-202122-uefa-champions-league/disciplinary.csv
/kaggle/input/ucl-202122-uefa-champions-league/goals.csv
/kaggle/input/ucl-202122-uefa-champions-league/attempts.csv
/kaggle/input/ucl-202122-uefa-champions-league/distributon.csv
```

Type Markdown and LaTeX: α^2

Viewing the loaded data

```python
df=pd.read_csv('/kaggle/input/ucl-202122-uefa-champions-league/disciplinary.csv')
df.head()
```

[7]:

	serial	player_name	club	position	fouls_committed	fouls_suffered	red	yellow	minutes_played	match_played
0	1	João Palhinha	Sporting CP	Midfielder	23	17	2	0	534	6
1	2	Capoue	Villarreal	Midfielder	19	17	3	1	1046	12
2	2	Rodri	Man. City	Midfielder	19	4	2	0	842	10
3	2	Seiwald	Salzburg	Midfielder	19	3	2	0	649	8
4	5	Balanta	Club Brugge	Midfielder	18	3	4	0	407	5

```python
df.describe()
```

[8]:

	serial	fouls_committed	fouls_suffered	red	yellow	minutes_played	match_played
count	584.000000	584.000000	584.000000	584.000000	584.000000	584.000000	584.000000
mean	261.876712	4.960616	4.525685	0.842466	0.042808	374.214041	5.690068
std	154.692375	3.774995	4.247815	0.933011	0.210895	230.626235	2.467248
min	1.000000	1.000000	0.000000	0.000000	0.000000	4.000000	1.000000
25%	146.000000	2.000000	1.000000	0.000000	0.000000	193.500000	4.000000
50%	263.000000	4.000000	3.000000	1.000000	0.000000	349.500000	5.500000
75%	417.000000	6.000000	7.000000	1.000000	0.000000	520.750000	7.000000
max	491.000000	23.000000	24.000000	4.000000	2.000000	1199.000000	13.000000

Fetching Null values

```python
total_null = df.isnull().sum().sort_values(ascending = False)
percent = ((df.isnull().sum()/df.isnull().count())*100).sort_values(ascending = False)
print("Total records = ", df.shape[0])
missing_data = pd.concat([total_null,percent.round(2)],axis=1,keys=['Total Missing','In Percent'])
missing_data.head(10)
```

```
Total records =  584
```

[9]:

	Total Missing	In Percent
serial	0	0.0
player_name	0	0.0
club	0	0.0
position	0	0.0
fouls_committed	0	0.0
fouls_suffered	0	0.0
red	0	0.0
yellow	0	0.0
minutes_played	0	0.0
match_played	0	0.0

Dropping irrelevant Data

```python
df=df.drop(['player_name'],axis=1)
```

```python
df.head()
```

[11]:

	serial	club	position	fouls_committed	fouls_suffered	red	yellow	minutes_played	match_played
0	1	Sporting CP	Midfielder	23	17	2	0	534	6
1	2	Villarreal	Midfielder	19	17	3	1	1046	12

|									
2	2	Man. City	Midfielder	19	4	2	0	842	10
3	2	Salzburg	Midfielder	19	3	2	0	649	8
4	5	Club Brugge	Midfielder	18	3	4	0	407	5

Necessary Plotting

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

Plot for Fouls committed

```python
[35]:   Q=df['fouls_committed'].value_counts()
        P=df['club'].value_counts()
```
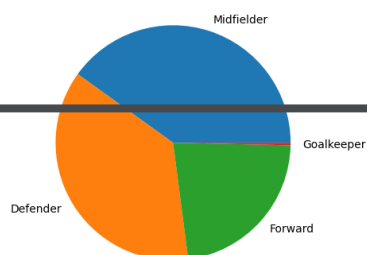
```python
sns.barplot(x=Q.index, y=Q.values)

plt.xlabel('Number of fouls comitted')
plt.ylabel('Number of players')
plt.title('Number of Fouls comitted per team')
plt.savefig('Number of Fouls comitted per team')
plt.show()
```
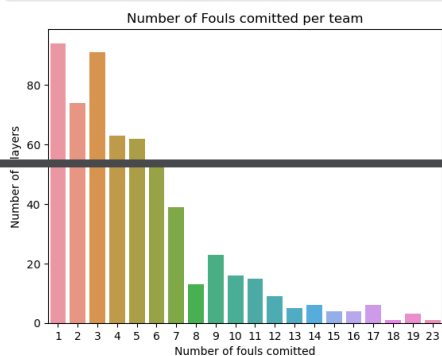

+ Code   + Markdown

Plot for Position

```python
[15]:   plt.pie(df['position'].value_counts().values,labels=df['position'].value_counts().index)
        plt.show()
```


Plot for Teams played

```python
fig = plt.figure(figsize = (10, 10))
plt.pie(df['club'].value_counts().values,labels=df['club'].value_counts().index)
plt.show()

#fig = plt.figure(figsize = (10, 5))
#plt.bar(df['fouls_committed'].value_counts().values,df['position'].value_counts().values,width=0.5)
```
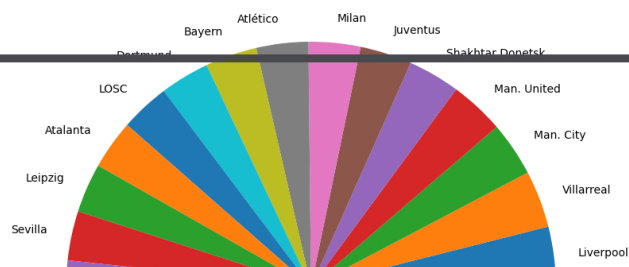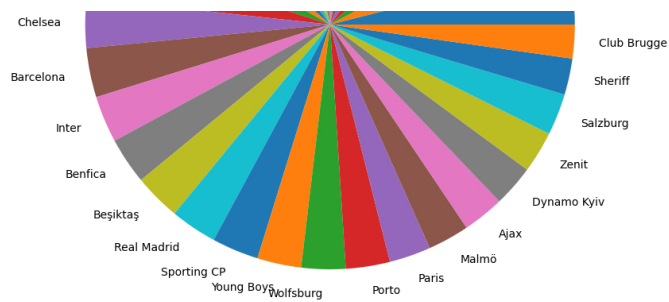
Cross Plot for type of cards

```
[13]:    A=df['red'].value_counts()
         B=df['yellow'].value_counts()
```

```
[13]:  0    560
       1     23
       2      1
       Name: yellow, dtype: int64
```

```
[25]:    fig, cards=plt.subplots()
         cards.plot(A.index,A.values,color='red')
         cards.plot(B.index,B.values,color='yellow')
         plt.legend(["Red Cards", "Yellow cards"],loc=0, frameon=True)
         plt.xlabel('Number of Instances per match')
         plt.ylabel('number of players')
```

```
[25]:  Text(0, 0.5, 'number of players')
```
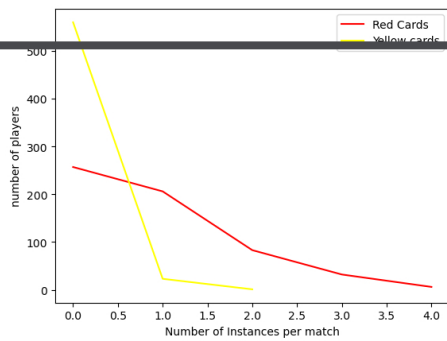

Cross Plotting betweeen Position and Yellow Cards

```
[28]:    sns.catplot(x="position",hue="yellow",kind="count", data=df)
```

```
[28]:  <seaborn.axisgrid.FacetGrid at 0x7fec1eff43d0>
```
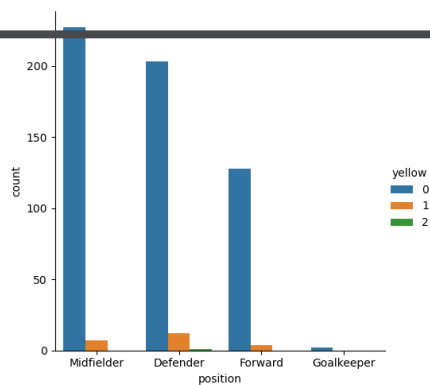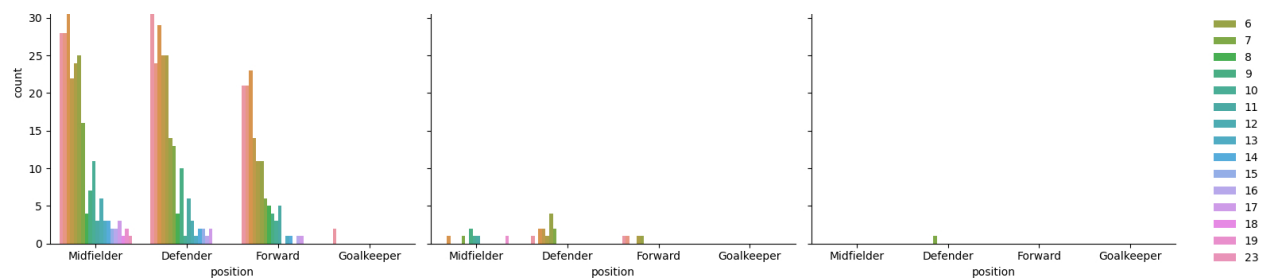

Cross Plot betweeen Position played and number of fouls committed

```
[32]:    sns.catplot(x="position",hue="fouls_committed",kind="count", col='yellow', data=df)
```

```
[32]:  <seaborn.axisgrid.FacetGrid at 0x7fec2a53b990>
```

matplotlib

Cross Plot betweeen Position played and number of fouls suffered

```
[33]: sns.catplot(x="position",hue="fouls_suffered",kind="count", col='red', data=df)
```

[33]: <seaborn.axisgrid.FacetGrid at 0x7fec29a1f7d0>