



File Edit View Run Add-ons Help

+ Run All | Code ▾

● Draft Session (1h:33m)

```
[1]: # 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/distribution.csv
```

+ Code

+ Markdown

```
[2]: df = pd.read_csv('/kaggle/input/ucl-202122-uefa-champions-league/disciplinary.csv')
```

```
[3]: df.head()
```

	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	Sewald	Salzburg	Midfielder	19	3	2	0	649	8
4	5	Balanta	Club Brugge	Midfielder	18	3	4	0	407	5

+ Code

+ Markdown

```
[4]: df.shape
```

```
[4]: (584, 10)
```

```
[5]: df.describe()
```

	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

```
[6]: df.describe(include = 'all')
```

	serial	player_name	club	position	fouls_committed	fouls_suffered	red	yellow	minutes_played	match_played
count	584.000000	584	584	584	584.000000	584.000000	584.000000	584.000000	584.000000	584.000000
unique		NaN	574	32	4	NaN	NaN	NaN	NaN	NaN
top		NaN	Peña	Liverpool	Midfielder	NaN	NaN	NaN	NaN	NaN
freq		NaN	2	23	234	NaN	NaN	NaN	NaN	NaN
mean	261.876712	NaN	NaN	NaN	4.960616	4.525685	0.842466	0.042808	374.214041	5.690068
std	154.692375	NaN	NaN	NaN	3.774995	4.247815	0.933011	0.210895	230.626235	2.467248
min	1.000000	NaN	NaN	NaN	1.000000	0.000000	0.000000	0.000000	4.000000	1.000000
25%	146.000000	NaN	NaN	NaN	2.000000	1.000000	0.000000	0.000000	193.500000	4.000000
50%	263.000000	NaN	NaN	NaN	4.000000	3.000000	1.000000	0.000000	349.500000	5.500000
75%	417.000000	NaN	NaN	NaN	6.000000	7.000000	1.000000	0.000000	520.750000	7.000000
max	491.000000	NaN	NaN	NaN	23.000000	24.000000	4.000000	2.000000	1199.000000	13.000000

+ Code

+ Markdown

```
[7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 584 entries, 0 to 583
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
 --- 
 0   serial          584 non-null   int64  
 1   player_name     584 non-null   object  
 2   club            584 non-null   object  
 3   position         584 non-null   object  
 4   fouls_committed 584 non-null   int64  
 5   fouls_suffered   584 non-null   int64  
 6   red              584 non-null   int64  
 7   yellow           584 non-null   int64  
 8   minutes_played  584 non-null   int64  
 9   match_played    584 non-null   int64  
dtypes: int64(7), object(3)
memory usage: 45.8+ kb
```

```
[8]: df.nunique()
```

```
[8]: serial      39
player_name  574
club        32
position      4
```

```
fouls_committed    20  
fouls_suffered     23  
red                 5  
yellow                3  
minutes_played   387  
match_played      13  
dtype: int64
```

+ Code + Markdown

Checking for NULL Values

```
[102]: 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	Missing	In Percent
[102]			
serial	0	0	0.0%
player_name	0	0	0.0%
club	0	0	0.0%
position	0	0	0.0%
fouls_committed	0	0	0.0%
fouls_suffered	0	0	0.0%
red	0	0	0.0%
yellow	0	0	0.0%
minutes_played	0	0	0.0%
match_played	0	0	0.0%

NO NULL VALUES PRESENT So we dont have to replace it

Data Visualization

```
[103]: import matplotlib.pyplot as plt  
import seaborn as sns
```

FOULS COMMITTED

```
[104]: fouls_committed_count = df['fouls_committed'].value_counts()  
fouls committed count
```

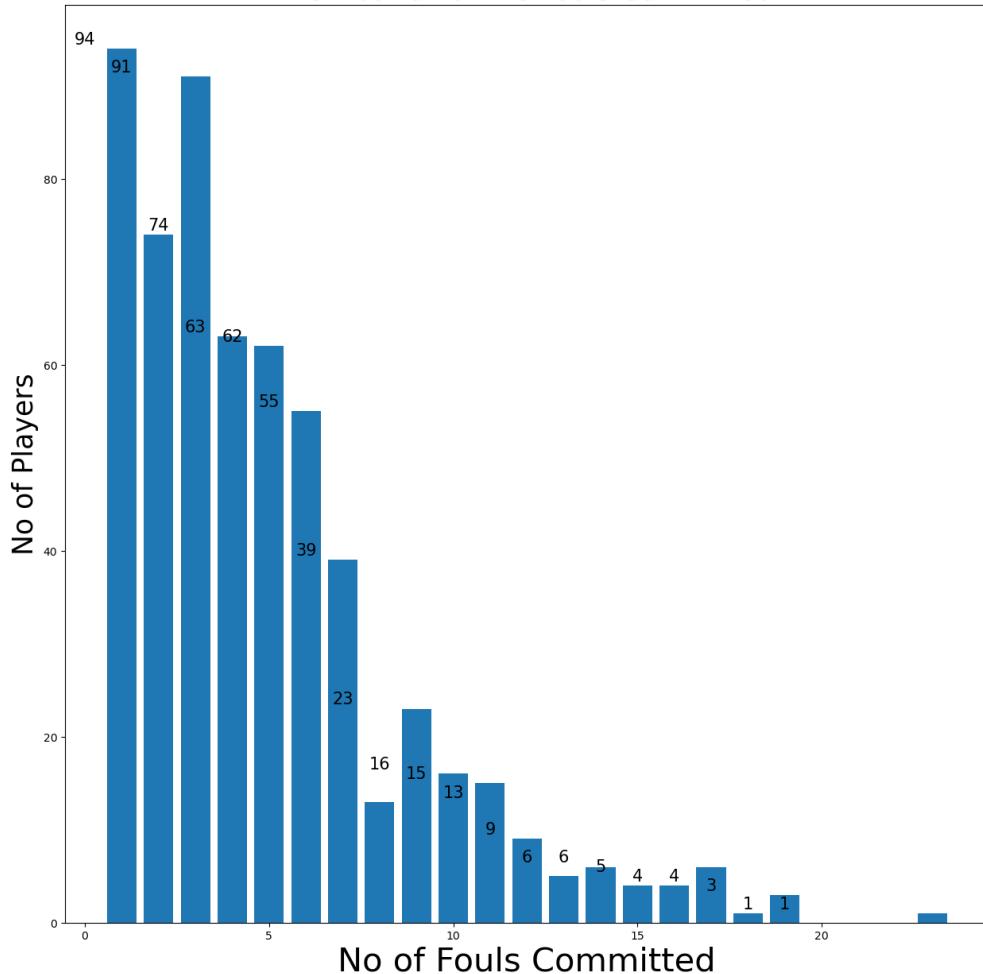
```
[104__] 1 94
      3 91
      2 74
      4 63
      5 62
      6 55
      7 39
      9 23
     10 16
     11 15
     8 13
     12 9
     14 6
     17 6
     13 5
     15 4
     16 4
     19 3
     18 1
     23 1
Name: faults committed, dtype: int64
```

```
[111]: fouls_committed_values = df['fouls_committed'].val  
fouls_committed_values
```

```
[105]:  
plt.figure(figsize=(15,15))  
plt.bar( fouls_committed_count.index, fouls_committed_count.values, width = 0.8)  
plt.title('Distribution of the Fouls Committed', fontsize = 25)  
# plt.xticks([0,1],['Not Survived','Survived'])  
plt.ylabel('No of Players', fontsize = 25)  
plt.xlabel('No of Fouls Committed',fontsize = 30)  
for i, value in enumerate(fouls_committed_count.values):  
    plt.text(i, value+1, str(value), fontsize=15, color='black', horizontalalignment='center', verticalalignment='center')
```

```
plt.show()
```

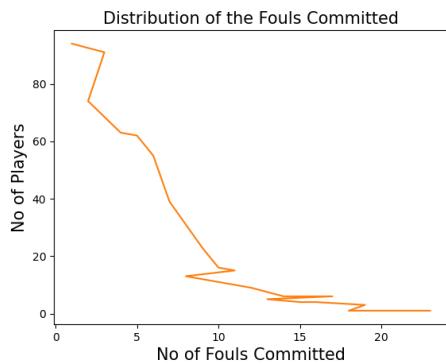
Distribution of the Fouls Committed



+ Code + Markdown

```
[106]:  
fig, ax = plt.subplots()  
ax.plot(fouls_committed_count.index,fouls_committed_count.values,color='C1')  
plt.title('Distribution of the Fouls Committed', fontsize = 15)  
plt.ylabel('No of Players', fontsize = 15)  
plt.xlabel('No of Fouls Committed', fontsize = 15)
```

```
[106... Text(0.5, 0, 'No of Fouls Committed')
```



FOULS SUFFERED

```
[107]:  
fouls_suffered_count = df['fouls_suffered'].value_counts()  
fouls_suffered_count
```

```
[107... 2    83  
0    79  
3    73  
1    70  
4    63  
6    41  
7    32  
8    28  
5    26  
11   18  
9    17  
10   15  
13   9  
17   6  
12   6  
14   5  
..   ..
```

```

18    4
15    4
16    2
20    2
24    1
23    1
19    2
Name: fouls_suffered, dtype: int64

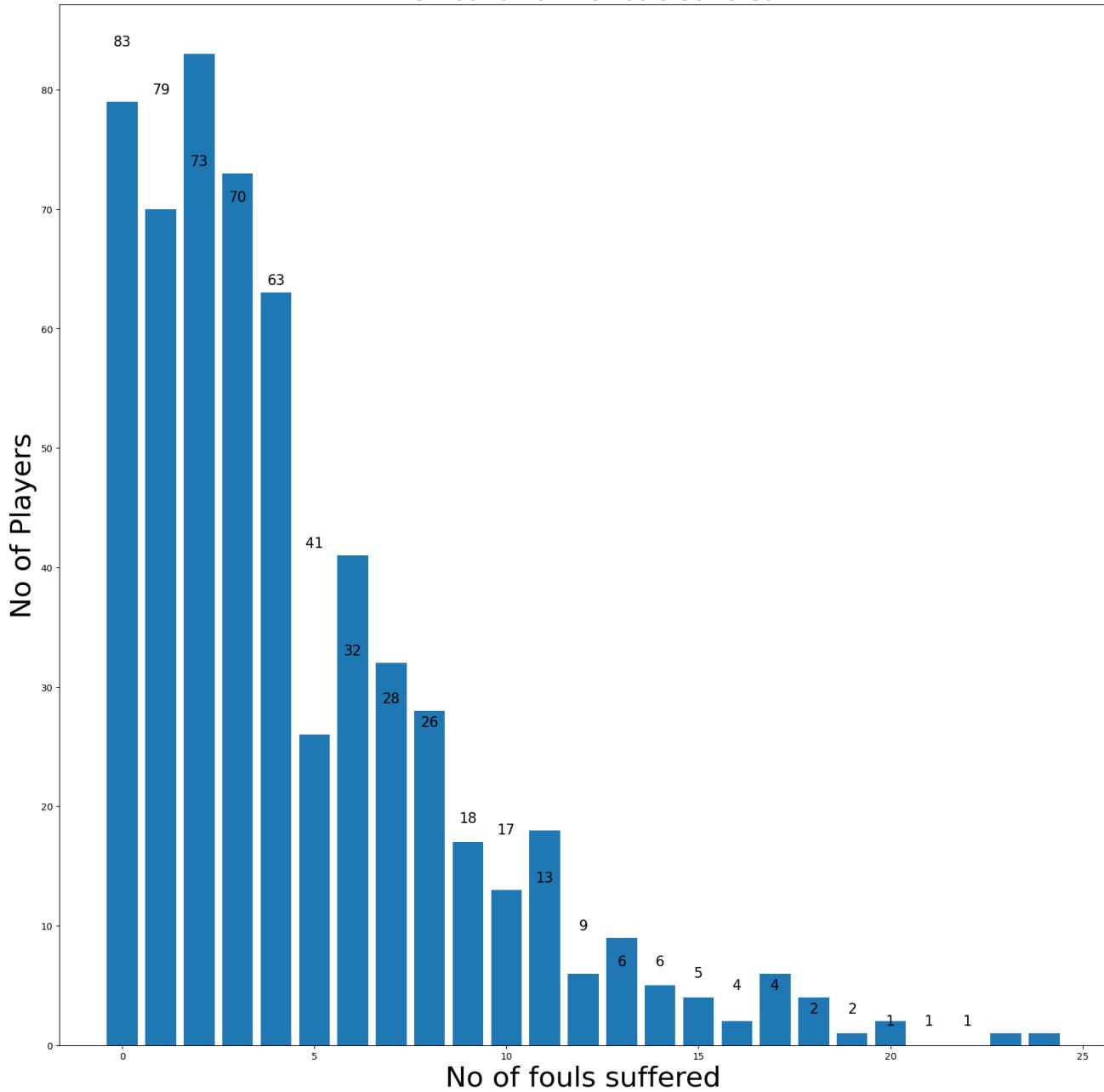
```

```

[15]:
plt.figure(figsize=(20,20))
plt.bar(fouls_suffered_count.index, fouls_suffered_count.values)
plt.title('Distribution of the Fouls Suffered', fontsize = 25)
# plt.xticks([0,1],['Not Survived','Survived'])
plt.ylabel('No of Players',fontsize = 30)
plt.xlabel('No of fouls suffered',fontsize = 30)
for i, value in enumerate(fouls_suffered_count.values):
    plt.text(i, value+1, str(value), fontsize=15, color='black', horizontalalignment='center',verticalalignment='center')
plt.show()

```

Distribution of the Fouls Suffered

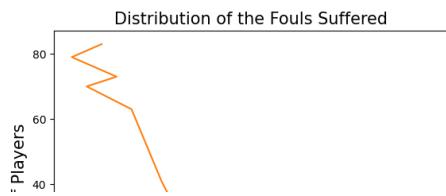


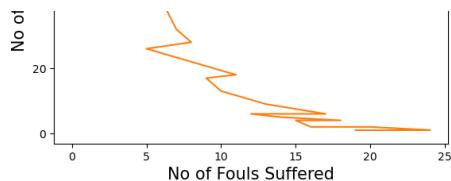
```

[16]:
fig, ax = plt.subplots()
ax.plot(fouls_suffered_count.index,fouls_suffered_count.values,color='C1')
plt.title('Distribution of the Fouls Suffered', fontsize = 15)
plt.ylabel('No of Players',fontsize = 15)
plt.xlabel('No of Fouls Suffered',fontsize = 15)

```

```
[16]: Text(0.5, 0, 'No of Fouls Suffered')
```



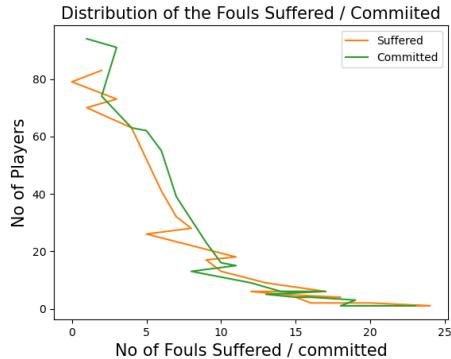


Fouls Committed VS Fouls Suffered

```
[17]: fig, ax = plt.subplots()
ax.plot(fouls_suffered_count.index,fouls_suffered_count.values,color='C1')
ax.plot(fouls_committed_count.index,fouls_committed_count.values,color='C2')
plt.legend(['Suffered','Committed'])

plt.title('Distribution of the Fouls Suffered / Committed' , fontsize = 15)
plt.ylabel('No of Players',fontsize = 15)
plt.xlabel('No of Fouls Suffered / committed',fontsize = 15)
```

[17]: Text(0.5, 0, 'No of Fouls Suffered / committed')

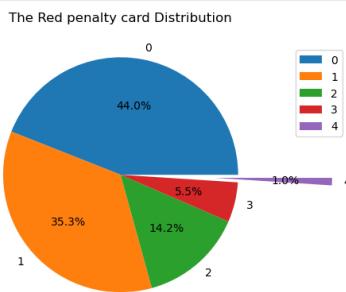


RED COUNT

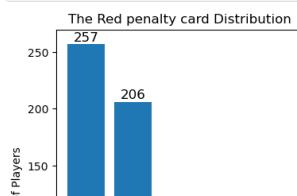
```
[18]: red_count = df['red'].value_counts()
red_count
```

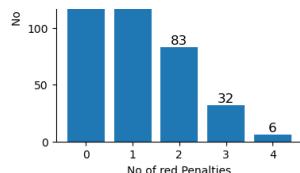
```
[18]: 0    257
1    206
2     83
3     32
4      6
Name: red, dtype: int64
```

```
[19]: plt.pie(red_count.values ,labels=red_count.index,autopct = '%.1f%%' ,explode=[0,0,0,0,0.8])
plt.title('The Red penalty card Distribution')
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=2)
plt.show();
```



```
[20]: plt.figure(figsize=(4,5))
plt.bar(red_count.index, red_count.values)
plt.title('The Red penalty card Distribution')
plt.ylabel('No of Players')
plt.xlabel('No of red Penalties')
for i, value in enumerate(red_count.values):
    plt.text(i, value+6, str(value), fontsize=12, color='black', horizontalalignment='center',verticalalignment='center')
plt.show()
```





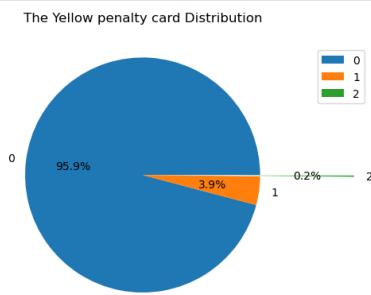
YELLOW COUNT

```
[21]: yellow_count = df['yellow'].value_counts()  
yellow_count
```

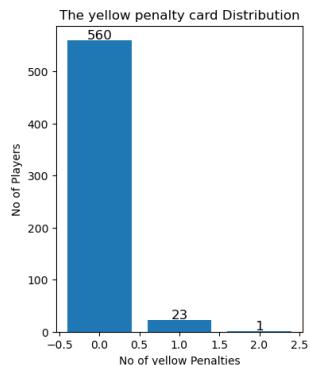
```
[21]:   0      560
        1      23
        2      1
Name: yellow, dtype: int64
```

```
[22]: yellow_values = df['yellow'].values  
yellow_values
```

```
plt.pie(x=yellow_count.values ,labels=yellow_count.index, autopct = '%.1f%%' ,explode=[0,0,0,0.8])
plt.title('The Yellow penalty card Distribution')
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=2)
plt.show();
```



```
[24]: plt.figure(figsize=(4,5))
plt.bar(yellow_count.index, yellow_count.values)
plt.title('The yellow penalty card Distribution')
plt.ylabel('No of Players')
plt.xlabel('No of yellow Penalties')
for i, value in enumerate(yellow_count.values):
    plt.text(i, value+9, str(value), fontsize=12, color='black', horizontalalignment='center',verticalalignment='center')
plt.show()
```



MINUTES PLAYED

```
[25]: minutes_played_count = df['minutes_played'].value_counts()
minutes_played_count
```

```
[25]: 540    15
450    12
720     6
180     6
484     5
      .
350     1
383     1
591     1
116     1
    7     1
Name: minutes_played, Length: 387, dtype: int64
```

```
[26]: minutes_played_values = df['minutes_played'].values
minutes_played_values
```

```
[26]: array([ 534, 1046, 842, 649, 407, 914, 1076, 819, 703, 668, 558,
       822, 715, 648, 398, 407, 281, 487, 672, 562, 581,
       473, 728, 421, 493, 446, 446, 716, 677, 672, 609,
       467, 273, 379, 231, 304, 427, 325, 876, 630, 666, 548,
       500, 496, 578, 572, 495, 849, 840, 720, 778, 429, 532,
       540, 401, 531, 400, 570, 398, 409, 878, 538, 660, 810,
       327, 666, 231, 484, 607, 461, 313, 487, 630, 734, 661,
       561, 484, 407, 891, 321, 1045, 494, 463, 320, 764, 352,
       510, 505, 1045, 425, 362, 489, 325, 229, 522, 553,
       418, 530, 337, 369, 393, 334, 903, 466, 354, 355,
       528, 587, 486, 460, 376, 205, 867, 431, 574, 720, 444,
       347, 464, 315, 397, 539, 312, 418, 420, 400, 603, 477,
       466, 442, 645, 328, 388, 573, 341, 472, 484, 444, 131,
       219, 225, 986, 632, 529, 405, 526, 373, 540, 543, 301,
       540, 618, 515, 986, 534, 540, 487, 268, 520, 435, 757,
       532, 592, 536, 621, 520, 520, 489, 362, 527, 521,
       383, 350, 276, 247, 142, 297, 292, 226, 212, 273, 154,
       247, 289, 358, 313, 630, 484, 664, 531, 272, 413, 80,
       450, 167, 582, 520, 578, 505, 336, 431, 658, 581, 296,
       863, 450, 358, 577, 528, 275, 540, 533, 540, 720, 604,
       509, 794, 540, 228, 335, 437, 448, 357, 343, 258, 471,
       509, 162, 162, 35, 563, 563, 563, 563, 563, 563, 48,
       261, 345, 276, 204, 487, 382, 289, 286, 232, 540, 426,
       300, 473, 333, 240, 572, 450, 233, 295, 168, 468, 463,
       1055, 573, 1008, 679, 511, 433, 509, 484, 349, 344, 660,
       523, 219, 389, 345, 119, 450, 444, 187, 804, 261, 264,
       180, 367, 180, 306, 685, 540, 499, 315, 289, 405, 97,
       452, 794, 362, 260, 210, 370, 290, 204, 419, 540, 516,
       523, 732, 262, 262, 262, 262, 262, 262, 262, 262, 277,
       328, 234, 165, 126, 78, 60, 702, 434, 715, 351, 510,
       673, 630, 541, 472, 1077, 252, 718, 524, 496, 435, 349,
       131, 525, 333, 330, 204, 490, 540, 450, 185, 450, 536,
       347, 332, 324, 257, 212, 179, 162, 91, 450, 438, 329,
       257, 88, 637, 328, 304, 281, 256, 230, 125, 125, 720,
       360, 260, 260, 260, 260, 210, 170, 170, 160, 50, 353,
       345, 283, 388, 330, 172, 178, 171, 20, 377, 462, 123,
       391, 360, 303, 260, 182, 177, 159, 218, 357, 216, 204,
       175, 133, 52, 222, 203, 170, 156, 54, 38, 436, 514,
       528, 243, 293, 261, 115, 255, 171, 798, 236, 194, 1106, 184,
       417, 296, 292, 192, 115, 497, 160, 156, 130, 80, 360,
       315, 211, 144, 105, 548, 93, 60, 47, 395, 322, 320,
       315, 250, 250, 250, 250, 180, 180, 81, 490, 370, 370,
       220, 75, 270, 180, 188, 158, 114, 98, 69, 49, 315,
       299, 269, 146, 139, 117, 104, 97, 38, 20, 207, 195,
       109, 101, 91, 79, 71, 6, 347, 500, 249, 356, 405,
       472, 155, 367, 322, 139, 56, 355, 351, 220, 180, 168,
       109, 93, 539, 540, 418, 113, 22, 277, 257, 209, 154,
       84, 196, 166, 166, 119, 253, 230, 230, 135, 122, 113,
       106, 95, 84, 65, 68, 56, 45, 540, 20, 227, 277,
       66, 22, 74, 293, 261, 180, 179, 129, 101, 101, 88,
       84, 81, 75, 68, 56, 36, 33, 315, 90, 70, 45,
       37, 18, 15, 14, 4, 540, 245, 192, 180, 170, 93,
       90, 18, 90, 73, 71, 51, 31, 19, 16, 13, 10, 9,
```

```
[27]: from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df['minutes_played'] = le.fit_transform(df['minutes_played'].astype(float))
```

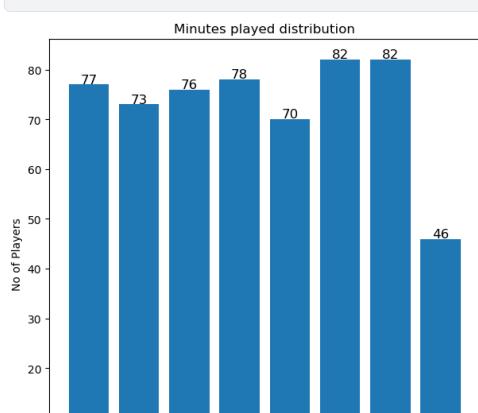
```
[35]: minutes_played_values_hist = np.histogram(minutes_played_values,bins=[0,50,100,150,200,250,300,350,400])
```

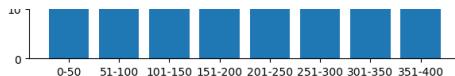
```
[29]: minutes_played_values_hist
```

```
[29]: (array([ 77, 73, 76, 78, 82, 82, 46, 0]), array([ 0, 50, 100, 150, 200, 250, 300, 350, 400]))
```

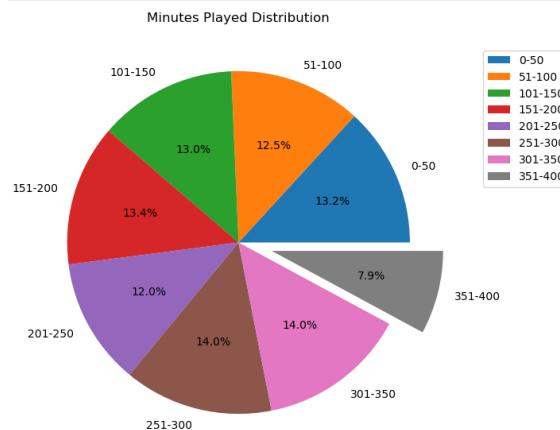
```
[36]: minutes_played_values_hist_labels = ['0-50', '51-100', '101-150', '151-200', '201-250', '251-300', '301-350', '351-400']
```

```
[37]: plt.figure(figsize=(7,7))
plt.title('Minutes played distribution')
plt.bar(minutes_played_values_hist_labels,minutes_played_values_hist[0])
plt.xlabel('Minutes Played',fontsize=0.2)
plt.ylabel('No of Players')
for i,value in enumerate(minutes_played_values_hist[0]):
    plt.text(i, value*1, str(value), fontsize=12, color='black', horizontalalignment='center',verticalalignment='center')
plt.show()
```





```
[48]: plt.figure(figsize=(7,7))
plt.title('Minutes Played Distribution')
plt.pie(minutes_played_values_hist[0], labels=minutes_played_values_hist_labels, autopct='%1.1f%%', explode=[0,0,0,0,0,0,0.2])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=2)
plt.show()
```



```
minutes_played_values_hist_labels = [
```

MATCH PLAYED

```
[61]: match_played_count = df['match_played'].value_counts()
match_played_count
```

```
[61]: 5    118
4    110
3     88
5      5
6     48
2     48
1     37
8     24
9     22
0     17
11    11
10     8
12     6
Name: match_played, dtype: int64
```

```
[62]: match_played_values = df['match_played'].values
match_played_values
```

```
[62]: array([ 5, 11,  9,  7,  4, 10, 11,  8,  7,  7,  6, 12,  8, 12,  5,  9,  6,
 4, 10,  7, 12, 11,  9,  5,  7, 12,  7,  8,  8,  7,  8,  8,  5,  6,
 5,  4,  7,  4,  5,  5,  9,  6,  7,  7,  5,  5, 10,  7,  5,  8,  8,
 7,  9,  4,  5,  5,  9,  5,  6,  7,  4,  6, 10,  6,  7,  8,  5, 11,
 7,  5,  7,  6,  3,  5,  6,  9,  7,  7,  5,  4,  9,  4, 11,  5,  7,
 5,  9,  7,  8,  5, 11,  5,  6,  5,  8,  6,  5,  6,  5,  5,  3,
 5,  6,  7, 11,  5,  2,  8,  7,  5,  4,  5,  5,  5,  4,  8,
 7,  5,  5,  5,  4,  4,  6,  5,  4,  4,  4,  7,  6,  5,  4,  9,
 7,  6,  5,  6,  5,  4,  3,  4,  3, 11,  8,  6,  5,  5,  5,  5,  5,
 4,  5,  7,  9, 10,  5,  5,  5,  3,  7,  6,  8,  5,  5,  6,  5,  8,
 5,  5,  3,  6,  3,  6,  4,  4,  4,  3,  4,  4,  4,  3,  3,  3,
 3,  3,  4,  3,  6,  5,  7,  7,  5,  4,  4,  2,  2,  7,  6,  8, 10,
 3,  3, 10,  7,  4,  7,  4,  6,  6,  5,  5,  5,  5,  5,  5,  5,
 8,  5,  5,  7,  4,  4,  7,  4,  8,  5,  9,  5,  4,  1,  7,  6,  9,
 9,  3,  3,  4,  4,  5,  2,  3,  5,  4,  4,  3,  6,  5,  4,  5,
 3,  4,  7,  4,  2,  2,  5,  5,  5, 11,  7, 12,  7,  6,  5,  5,
 7,  4,  9,  5,  3,  5,  4,  3,  4,  4,  6, 10,  4,  5,  1,  6,
 3,  6,  5,  5,  3,  4,  4,  4,  4,  8,  5,  3,  8,  5,  3,  3,
 5,  7,  6,  4,  4,  4,  4,  4,  3,  4,  4,  4,  4,  4,  4,  1,
 5,  7,  5,  7,  4,  4,  4,  4,  6,  6,  6, 12,  3,  7,  4,  4,
 4,  4,  5,  5,  8,  3,  5,  5,  4,  3,  4,  5,  6,  3,  5,  4,
 5,  4,  2,  4,  4,  4,  3,  1,  7,  3,  3,  4,  4,  3,  4,  3,
 7,  4,  5,  7,  3,  3,  4,  4,  6, 11,  6,  4,  3,  6,  5,  3,
 5,  2,  6,  5,  5,  7,  3,  3,  4,  4,  3,  3,  4,  5,  4,  4,
 2,  2,  4,  5,  4,  4,  3,  1,  2,  4,  9,  5,  3,  3,  2,  4,
 4,  3,  5,  4,  4,  3,  1,  2,  4,  9,  5,  3,  3,  2,  4,
 5,  2,  6,  5,  5,  7,  3,  3,  4,  4,  3,  3,  4,  5,  4,  4,
 5,  5,  5,  1,  2,  3,  4,  4,  5,  5,  5,  3,  2,  1,  4,
 1,  4,  2,  4,  3,  2,  2,  3,  1,  5,  2,  2,  3,  3,  2,  2,
 2,  1,  1,  1,  3,  0,  7,  3,  4,  1,  1,  2,  4,  0,  7,  5,
 5,  5,  6,  2,  4,  4,  2,  2,  4,  3,  2,  1,  3,  3,  2,  5,
 4,  2,  0,  5,  7,  3,  5,  2,  4,  3,  3,  4,  2,  6,  2,  1,
 2,  2,  4,  1,  1,  2,  1,  1,  4,  4,  1,  1,  1,  1,  3,  2,
 2,  2,  4,  2,  1,  1,  1,  1,  4,  4,  1,  1,  1,  1,  3,  0,
 1,  1,  1,  1,  0,  5,  6,  2,  1,  4,  2,  0,  0,  2,  0,  3,
 1,  1,  0,  0,  0,  0])
```

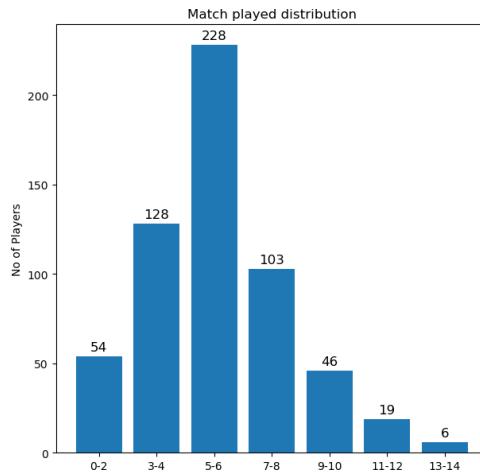
```
[49]: from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df['match_played'] = le.fit_transform(df['match_played'].astype(float))
```

```
[97]: match_played_values_hist = np.histogram(match_played_values,bins=[0,2,4,6,8,10,12,14])
match_played_values_hist
```

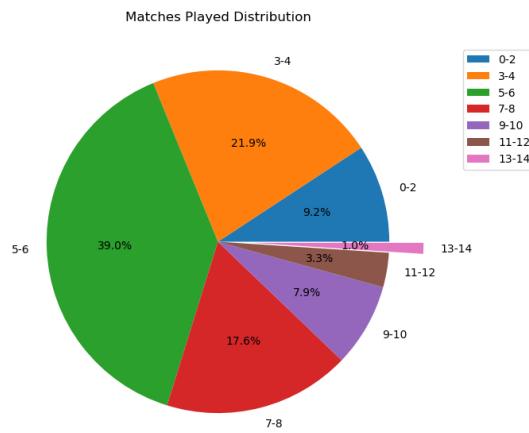
```
[97]: (array([ 54, 128, 228, 103,  46,  19,   6]),
 array([ 0,  2,  4,  6,  8, 10, 12, 14]))
```

```
[98]: match_played_values_hist_labels = ['0-2','3-4','5-6','7-8','9-10','11-12','13-14']
```

```
[99]:  
plt.figure(figsize=(7,7))  
plt.title('Match played distribution')  
plt.bar(match_played_values_hist_labels,match_played_values_hist[0])  
plt.xlabel('Match Played', fontsize=0.2)  
plt.ylabel('No of Players')  
for i,value in enumerate(match_played_values_hist[0]):  
    plt.text(i, value*5, str(value), fontsize=12, color='black', horizontalalignment='center', verticalalignment='center')  
plt.show()
```



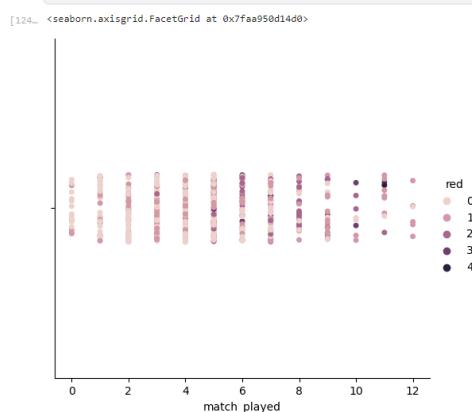
```
[100]:  
plt.figure(figsize=(7,7))  
plt.title('Matches Played Distribution')  
plt.pie(match_played_values_hist[0], labels=match_played_values_hist_labels, autopct="%1.1f%%", explode=[0,0,0,0,0,0.2])  
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=2)  
plt.show()
```



CROSS ATTRIBUTES

MATCH PLAYED VS RED CARD

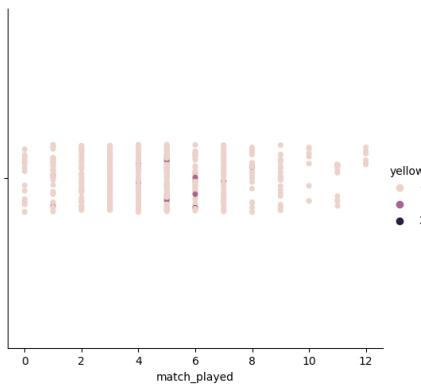
```
[124]:  
sns.catplot(x="match_played",hue="red",data=df)
```



MATCH PLAYED VS YELLOW CARD

```
[125]: sns.catplot(x="match_played", hue="yellow", data=df)
```

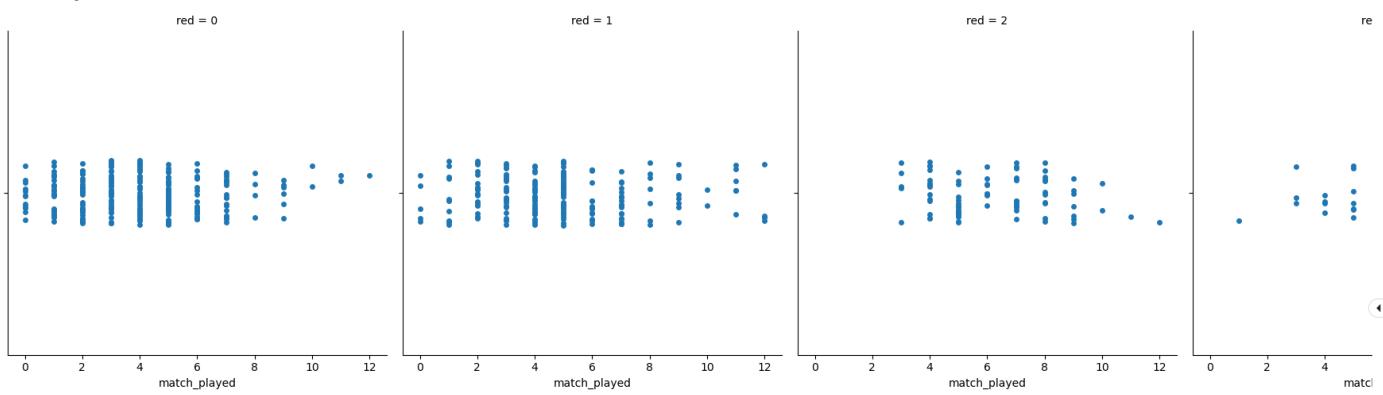
```
[125]: <seaborn.axisgrid.FacetGrid at 0x7faa952c8b50>
```



MATCHES PLAYED VS DIFFERENT VALUES OF RED CARD

```
⇒ sns.catplot(x="match_played", col="red", data=df)
```

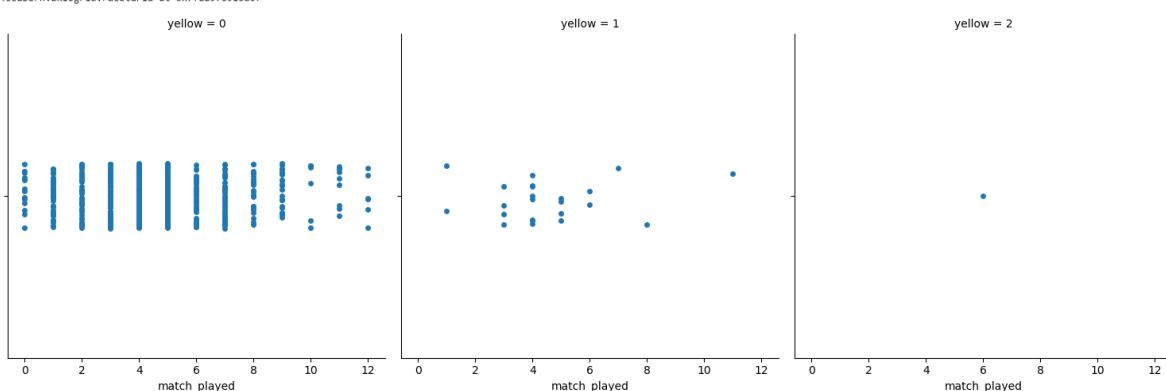
```
[126]: <seaborn.axisgrid.FacetGrid at 0x7faa96e50f10>
```



MATCHES PLAYED VS DIFFERENT VALUES OF YELLOW CARD

```
⇒ sns.catplot(x="match_played", col="yellow", data=df)
```

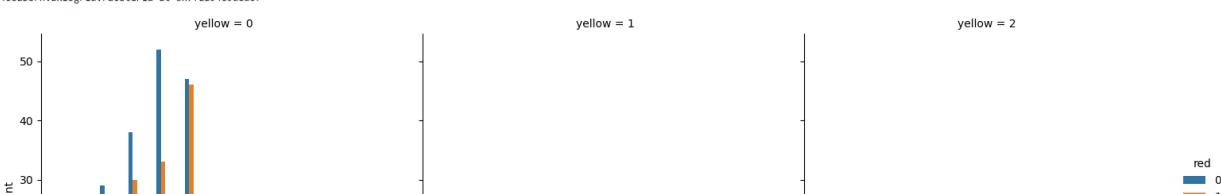
```
[127]: <seaborn.axisgrid.FacetGrid at 0x7faa970913d0>
```

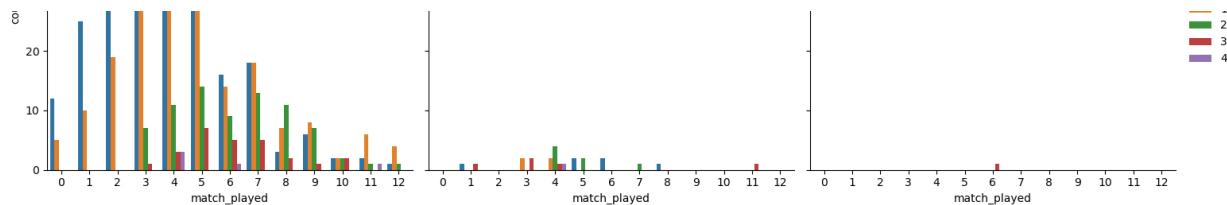


MATCH PLAYED VS RED VS YELLOW

```
⇒ sns.catplot(x="match_played", hue="red", kind="count", col="yellow", data=df)
```

```
[130]: <seaborn.axisgrid.FacetGrid at 0x7faa9469d8d0>
```





+ Code + Markdown

MATCH PLAYED VS YELLOW VS RED

```
[131]: sns.catplot(x="match_played", hue="yellow", kind="count", col="red", data=df)
```

