**NBA 2K Data Analysis & Visualization Report**

**Summary**

This is a visualization for NBA2K20 dataset kindly provided by **SuJian**.

****NBA 2K20**** is a basketball simulation video game developed by Visual Concepts and published by 2K Sports, based on the National Basketball Association (NBA). It is the 21st installment in the NBA 2K franchise, the successor to NBA 2K19, and the predecessor to NBA 2K21. Anthony Davis of the Los Angeles Lakers is the cover athlete for the regular edition of the game, while Dwyane Wade is the cover athlete for the 'Legend Edition'.NBA 2K20 was released on September 6, 2019, for Microsoft Windows, Nintendo Switch, PlayStation 4, and Xbox One, and on November 18, 2019 for Stadia.

The player mainly plays NBA games with real-life or customized players and teams; games follow the rules and objectives of NBA games. Several game modes are present and many settings can be customized. Up to six expansion teams can be created and used in both MyLeague and MyGM Modes, with the possibility of a 36-team league, and any team can be relocated and rebranded.

Publicity pictures

**Import Package**

There are packages you should import.

import numpy as np

import pandas as pd

import plotly.express as px

from datetime import date

import plotly.graph\_objs as go

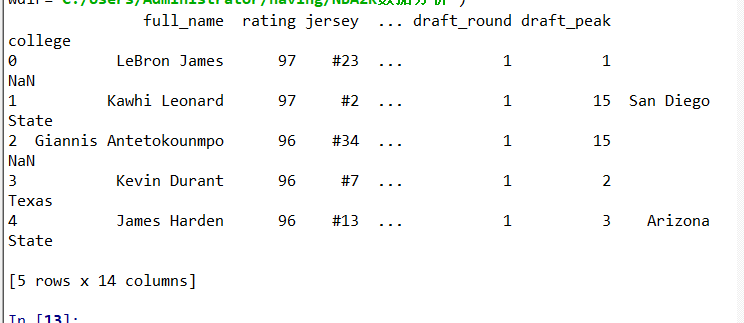
from iso3166 import countries

**Data overview**

We will show you about the first rows data of the dataset.

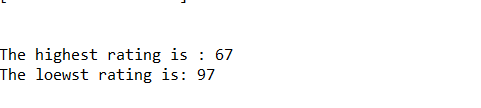
df = pd.read\_csv('r'C:\Users\Administrator\Desktop\python数据分析\nba2k20-full.csv'')

df.head() #展示数据集的前五行



**Show the highest & lowest rating :**

print('Top rating presented in dataset: ', df['rating'].max())print('Low rating presented in dataset: ', df['rating'].min())



**Players rating distribution:**

#统计同类数值的函数，column是列的名字

def same\_values\_column(column):

distribution=df[[column,'full\_name']].groupby(column).count().sort\_values(by='full\_name',ascending=False).reset\_index()

distribution.columns=[column,'number']

print(f"球员中评分最多的数值是{distribution.rating[0]}")

fig, ax = plt.subplots(figsize=(10,5))

sns.barplot(x=distribution.rating, y=distribution.number)

plt.xticks(rotation=90)

plt.xlabel('Rating value')

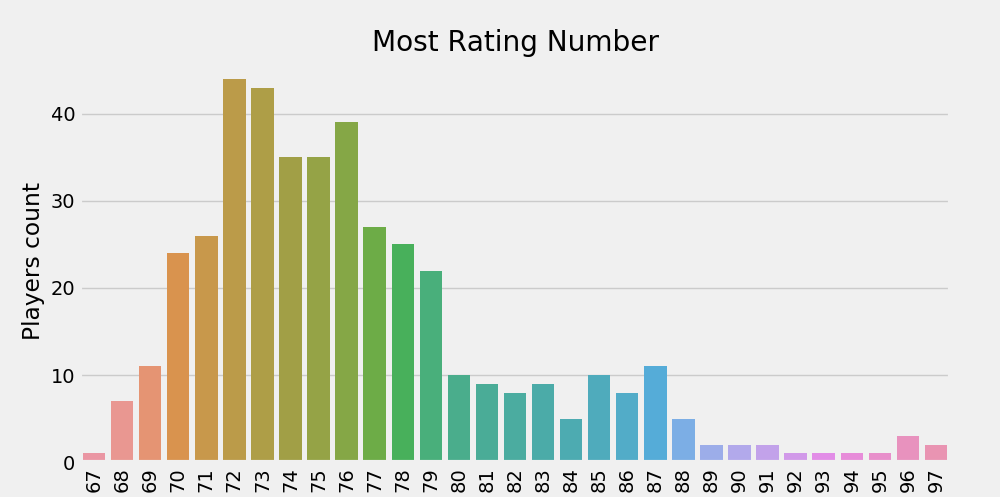
plt.ylabel('Players count')

plt.title('Most Rating Number')

plt.show()

same\_values\_column('rating')

Most Rating Distribution Statistical chart:



**Analysis**:We can see that the players mainly focus on the rating of 71 to 79

**Players country distribution:**

#统计同类数值的函数，column是列的名字

def country\_values\_column(column):

distribution=df[[column,'full\_name']].groupby(column).count().sort\_values(by='full\_name',ascending=False).reset\_index()

distribution.columns=[column,'number']

print(f"最多球员的国家是{distribution.country[0]}")

fig, ax = plt.subplots(figsize=(10,5))

sns.barplot(x=distribution.country, y=distribution.number)

plt.xticks(rotation=90)

plt.xlabel('Country name')

plt.ylabel('Players count')

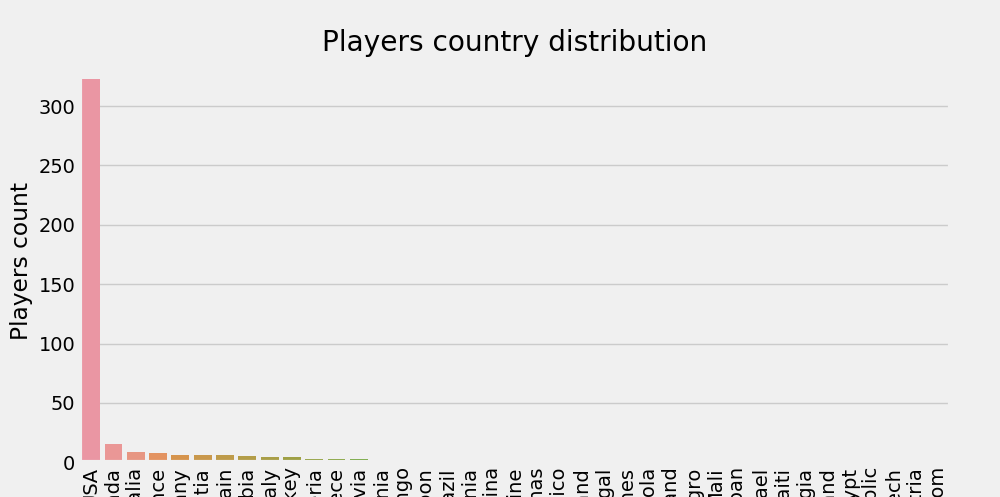
plt.title('Players country distribution')

plt.show()

#统计球员国籍分布

country\_values\_column('country')

Player country Distribution Statistical chart:



**Analysis:**We can see that the nationality of the players is mainly concentrated in the United States, reaching more than 300.

**Players team distribution:**

#统计每支球队人数的函数，column是列的名字,扇形统计图

def team\_values\_pie(column):

distribution=df[[column,'full\_name']].groupby(column).count().sort\_values(by='full\_name',ascending=False).reset\_index()

distribution.columns=[column,'number']

print(f"最多球员的球队是{distribution.team[0]}")

labels =np.array(distribution.team)#讲对象转化为数组

sizes = np.array(distribution.number)

plt.pie(sizes,labels=labels,autopct='%1.1f%%',shadow=False,startangle=150)

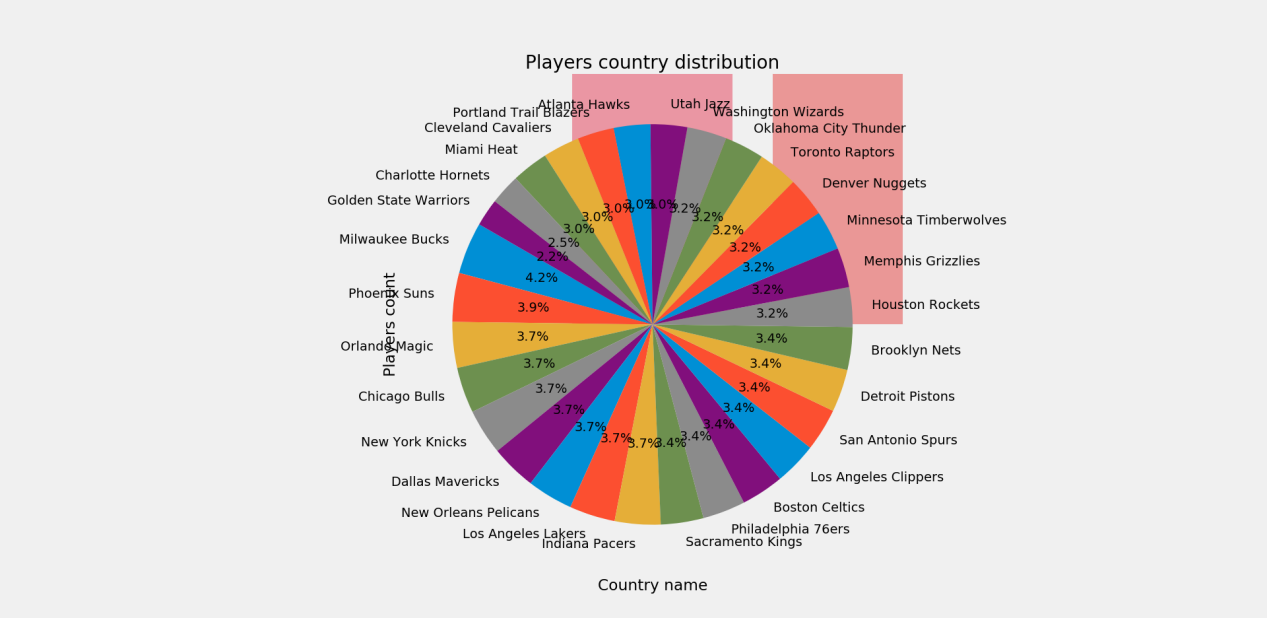
plt.title("Players country distribution")

plt.show()

#调用函数观看各球队人数

team\_values\_pie('team')

Player team Distribution Pie Statistical chart:



**Analysis:**The number of teams accounts for about 3%.

**Players highest rating distribution:**

#各球队的最高评分

def rating\_values\_line(column):

distribution=df[[column,"team"]]

df1=df[[column,'team']].groupby('team').count().sort\_values(by='team',ascending=False).reset\_index()

indexs=np.array(df1.team) #生成球队名字列表

a=[]

for index in indexs:

beat=distribution[distribution.team==index].rating.max()

a.append(beat)

print(f"{index}球队的球员最高评分是{beat}")

fig, ax = plt.subplots(figsize=(20,10))

sns.barplot(x=indexs, y=a)

plt.xticks(rotation=90)

plt.xlabel('Team name')

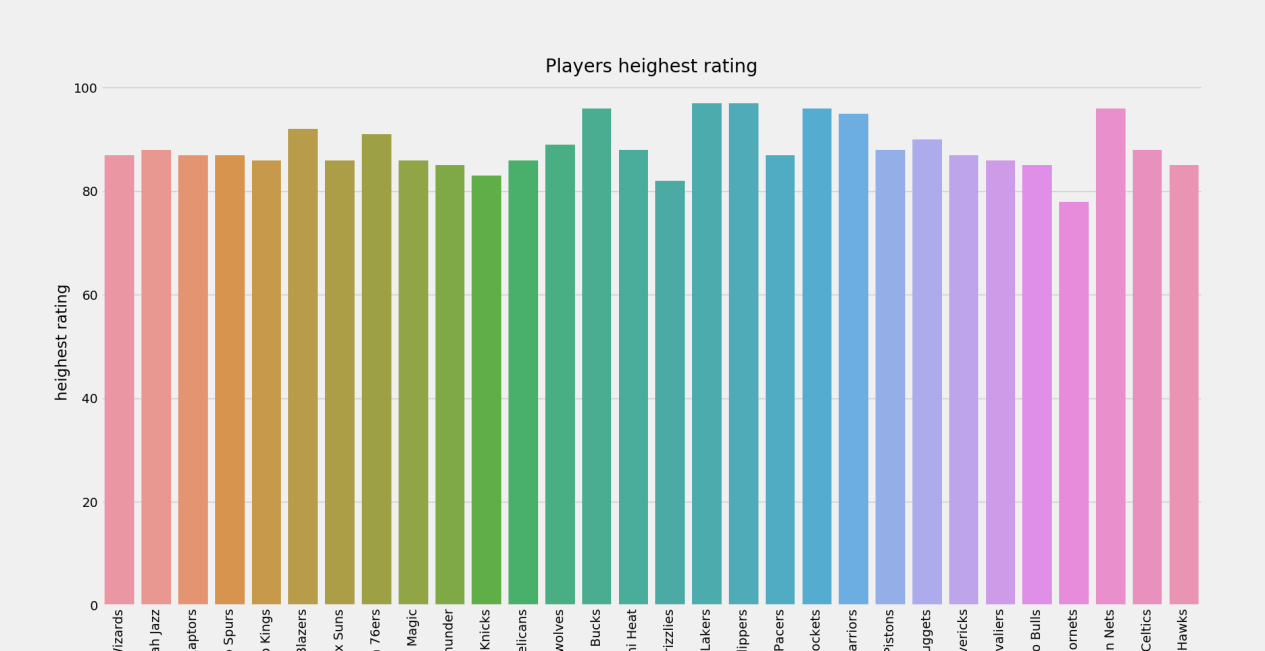
plt.ylabel('heighest rating')

plt.title('Players heighest rating')

plt.show()

rating\_values\_line('rating')

Players Highest Rating Distribution Statistical chart:

****

**Analysis:**The highest score of players in each team is mainly over 80.

**Players draft\_year distribution:**

#统计每年选修的球员人数

def draft\_values\_pie(column):

distribution=df[[column,'full\_name']].groupby(column).count().sort\_values(by='full\_name',ascending=False).reset\_index()

distribution.columns=[column,'number']

print(f"选秀球员最多的年份是{distribution.draft\_year[0]}")

fig, ax = plt.subplots(figsize=(20,10))

labels =np.array(distribution.draft\_year) #讲对象转化为数组

sizes = np.array(distribution.number)

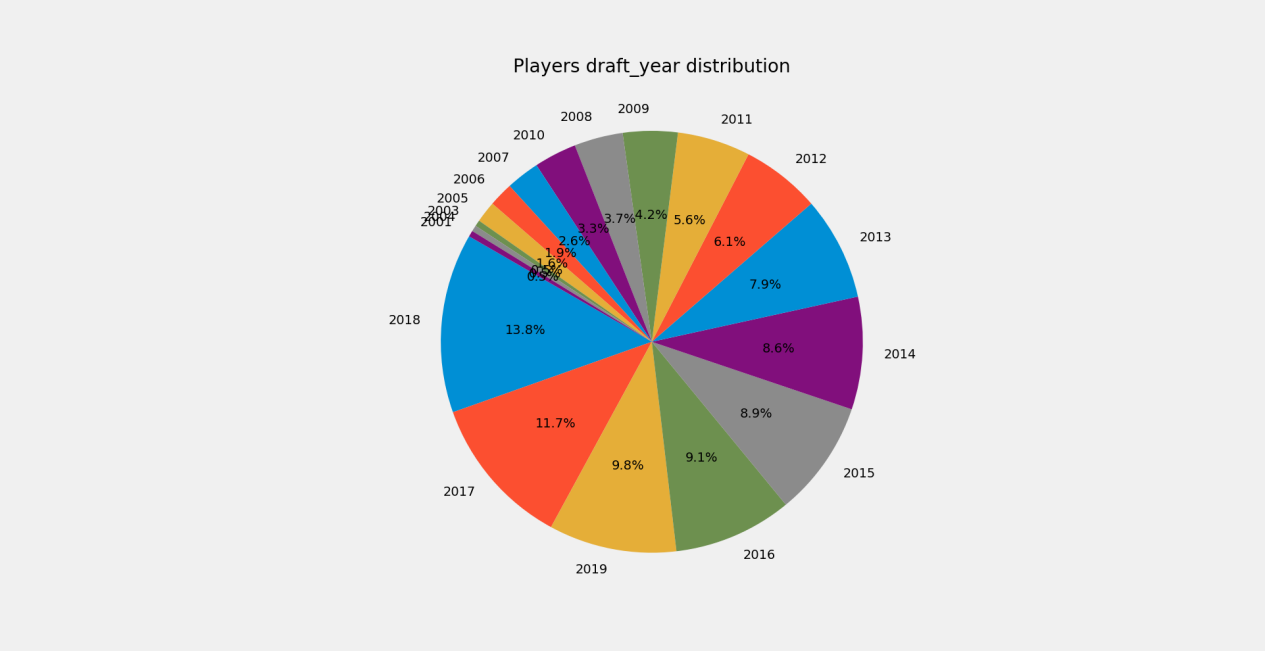
plt.pie(sizes,labels=labels,autopct='%1.1f%%',shadow=False,startangle=150)

plt.title("Players draft\_year distribution")

plt.show()

draft\_values\_pie('draft\_year')

Players Highest Rating Distribution Statistical chart:

****

**Analysis:**The year of the most electives is 2018.

**Players position distribution:**

#打印球员工资

def player\_salary\_column(column):

df1=df[['full\_name',column]]

df1[column]=df1[column].apply(lambda x:get\_salary(x))

df1=df1.sort\_values(by='salary',ascending=False).reset\_index()

print(df1)

print(f"球员中最高薪资的是{df1['salary'].max()},薪资最高的球员是{df1[df1.salary==df1['salary'].max()].full\_name}")

print(f"球员中最低薪资的是{df1['salary'].min()},薪资最高的球员是{df1[df1.salary==df1['salary'].min()].full\_name}")

#打印统计图

fig, ax = plt.subplots(figsize=(20,10))

sns.barplot(x=df1.full\_name, y=df1.salary)

plt.xticks(rotation=90)

plt.xlabel('Players name')

plt.ylabel('Players salary')

plt.title('Players salary distribution')

plt.show()

#去掉球员工资前面的美元符号

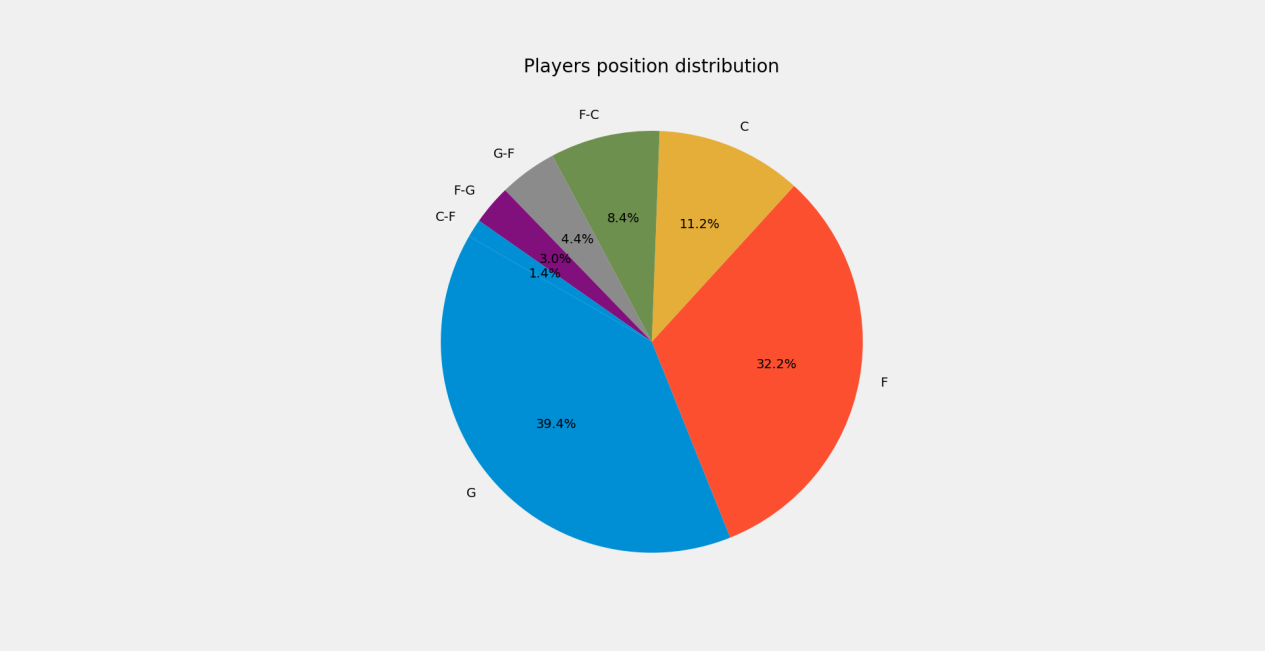
def get\_salary(salary):

salary=salary.replace('$',' ')

return float(salary)

player\_salary\_column('salary')

Players Position Distribution Statistical chart:

****

**Players salary distribution:**

#打印球员工资

def player\_salary\_column(column):

df1=df[['full\_name',column]]

df1[column]=df1[column].apply(lambda x:get\_salary(x))

'''df1=df1.sort\_values(by='salary',ascending=False).reset\_index()'''

print(df1)

print(f"球员中最高薪资的是{df1['salary'].max()},薪资最高的球员是{df1[df1.salary==df1['salary'].max()].full\_name}")

print(f"球员中最低薪资的是{df1['salary'].min()},薪资最高的球员是{df1[df1.salary==df1['salary'].min()].full\_name}")

#打印统计图

fig, ax = plt.subplots(figsize=(20,10))

sns.barplot(x=df1.full\_name, y=df1.salary)

plt.xticks(rotation=90)

plt.xlabel('Players name')

plt.ylabel('Players salary')

plt.title('Players salary distribution')

plt.show()

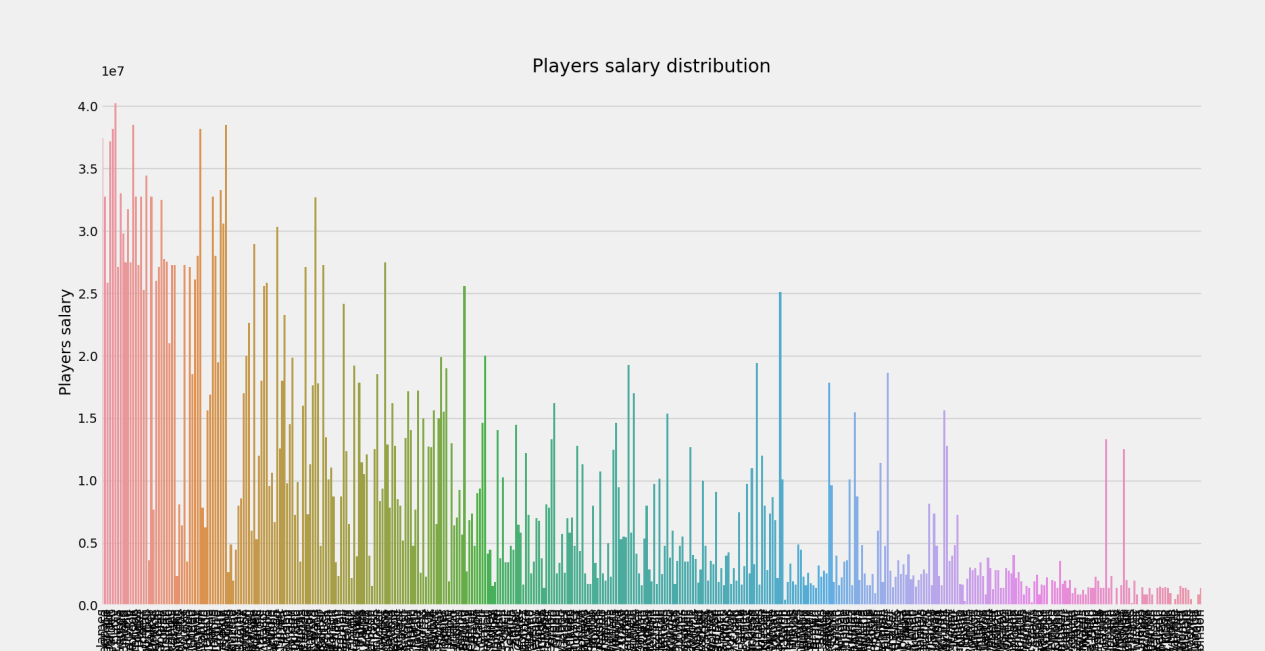
#去掉球员工资前面的美元符号

def get\_salary(salary):

salary=salary.replace('$',' ')

return float(salary)

player\_salary\_column('salary')

****

**Team salary cap:**

#统计球队的工资帽

def team\_salary\_cap(column):

distributions=df[[column,"team"]]

df1=df[[column,'team']].groupby('team').count().sort\_values(by='team',ascending=False).reset\_index()

indexs=np.array(df1.team) #生成球队名字列表

data=[]

team\_salary=0

for index in indexs:

for row in distributions.itertuples():

if(getattr(row,'team')==index):

team\_salary+=get\_salary(getattr(row,'salary'))

data.append([index,team\_salary])

team\_salary=0

df2=pd.DataFrame(data)

df2.columns=['team\_name','salary\_cap']

print(f"工资帽最高的是{df2['salary\_cap'].max()}")

print(f"工资帽最低的是{df2['salary\_cap'].min()}")

print(df2)

#打印统计图

fig, ax = plt.subplots(figsize=(20,10))

sns.barplot(x=df2.team\_name, y=df2.salary\_cap)

plt.xticks(rotation=270)

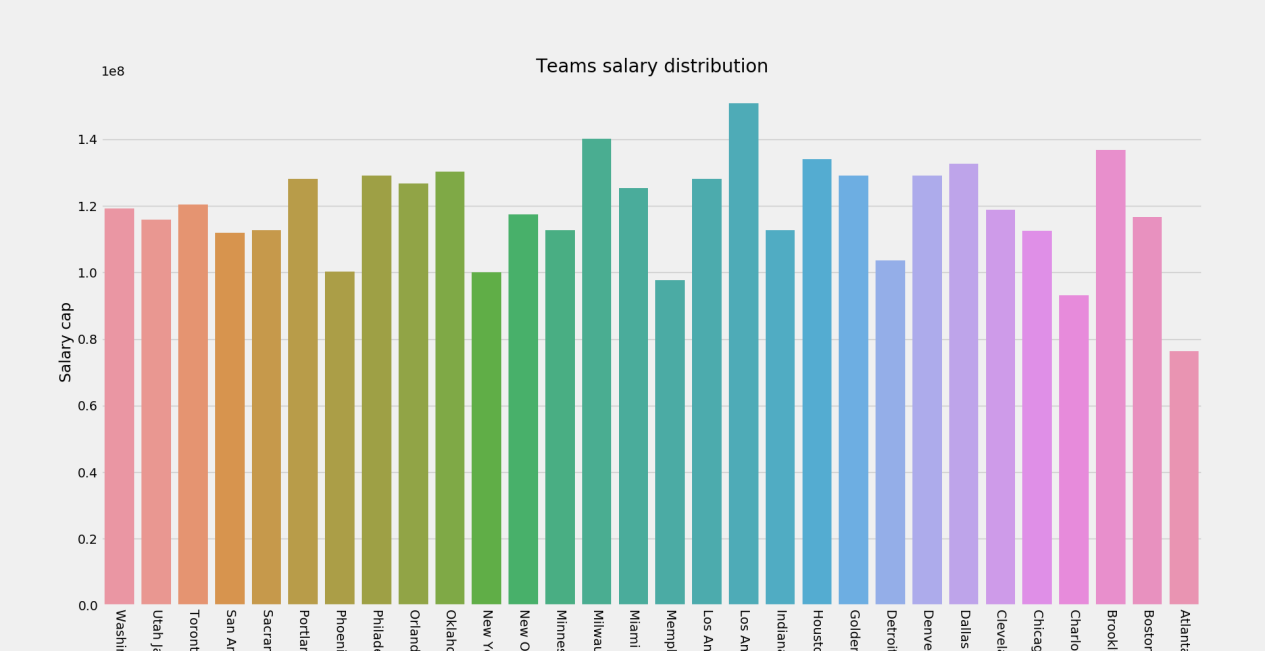
plt.xlabel('Teams name')

plt.ylabel('Salary cap')

plt.title('Teams salary distribution')

plt.show()

team\_salary\_cap('salary')



Operation Result:

工资帽最高的是150836802.0

工资帽最低的是76405008.0

team\_name salary\_cap

0 Washington Wizards 119324102.0

1 Utah Jazz 115926548.0

2 Toronto Raptors 120457739.0

3 San Antonio Spurs 111985972.0

4 Sacramento Kings 112635973.0

5 Portland Trail Blazers 128052735.0

6 Phoenix Suns 100308557.0

7 Philadelphia 76ers 129016567.0

8 Orlando Magic 126644900.0

9 Oklahoma City Thunder 130282723.0

10 New York Knicks 100072993.0

11 New Orleans Pelicans 117526046.0

12 Minnesota Timberwolves 112643846.0

13 Milwaukee Bucks 140102084.0

14 Miami Heat 125260212.0

15 Memphis Grizzlies 97706111.0

16 Los Angeles Lakers 128037562.0

17 Los Angeles Clippers 150836802.0

18 Indiana Pacers 112713124.0

19 Houston Rockets 134127098.0

20 Golden State Warriors 129126051.0

21 Detroit Pistons 103702143.0

22 Denver Nuggets 129184213.0

23 Dallas Mavericks 132671963.0

24 Cleveland Cavaliers 118818083.0

25 Chicago Bulls 112442765.0

26 Charlotte Hornets 93259611.0

27 Brooklyn Nets 136848791.0

28 Boston Celtics 116701886.0

29 Atlanta Hawks 76405008.0

**Draft distribution:**

def draft\_distribution(column):

data=df[['full\_name',column]]

undraft\_sum=0

first\_round=0

second\_round=0

#计算第一轮、第二轮和落选秀

for row in data.itertuples():

if(getattr(row,'draft\_peak')=='Undrafted'):

undraft\_sum+=1

else:

if(int(getattr(row,'draft\_peak'))<=30):

first\_round+=1

else:

second\_round+=1

labels = ['First\_round','Seconde\_round','Undraft']

values = [undraft\_sum,first\_round,second\_round]

fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])

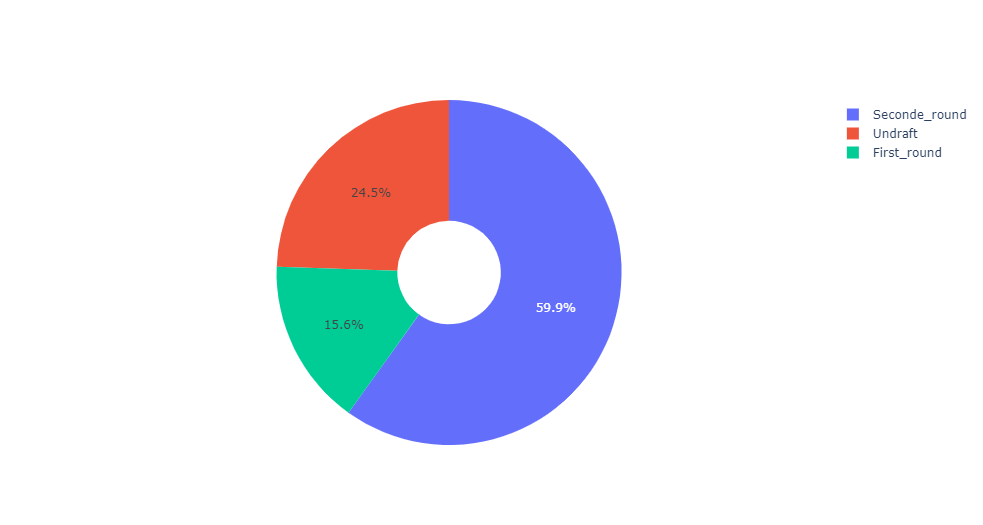
fig.show()

print(undraft\_sum)

print(first\_round)

print(second\_round)

draft\_distribution('draft\_peak')

****