

# DADOS RELATIVOS AOS 50 MELHORES JOGADORES DE NBA NA TEMPORADA 2021/2022



Como base e principal ideia deste trabalho escolhemos um tema em comum aos dois elementos do grupo. Quer seja por gosto ou por cansaço de ouvir falar do mesmo, escolhemos como tema o basquetebol. Mais propriamente, devido à facilidade na colheita de dados, os 50 melhores jogadores de NBA na temporada 2021/2022 e alguns dos seus feitos (equipa onde os jogadores jogam, partidas jogadas pelos mesmos, minutos e pontos por jogo, etc).

Utilizamos dados do site oficial do MSN (<https://www.msn.com/pt-br/esportes/basquete/nba/estatisticas-do-jogador>). Consideramos esta como uma fonte confiável devido à sua frequente utilização e popularidade. É importante destacar que poderá haver uma nova atualização na tabela do site, portanto, decidimos utilizar os dados iniciais até ao fim do projeto (12/06/2022), sem alterá-los conforme o MSN.

Na recolha de dados tivemos o cuidado de verificar se os que se encontravam presentes no site oficial da MSN correspondiam aos apresentados pela aplicação da NBA.

Para uma melhor utilização, achamos por bem realizar uma “limpeza” através do excel que consistiu na substituição de vírgulas por pontos e na remoção de pontos nos números inteiros. Existiu também a remoção de “valores ausentes” que não ocorriam em campos importantes para os objetivos do projeto e não existem valores duplicados ou a precisar de correção.

Os campos mais importantes para os objetivos deste projeto são a equipa onde os jogadores jogam, as partidas jogadas pelos mesmos, os minutos e pontos por jogo.

Apesar de não ter existido nenhum problema de tipo de dados tivemos de rejeitar alguns. Rejeitamos duas colunas (FG% e PF), pois consideramos que as mesmas não afetariam as análises que pretendíamos realizar nem afetarão as conclusões a que chegaremos.



# BIBLIOGRAFIA

- [https://www.msn.com/pt-br/esportes/basquete/nba/estatisticas-do-jogador.](https://www.msn.com/pt-br/esportes/basquete/nba/estatisticas-do-jogador)
- NBA App.
- Programação em Python-Fundamentos e Resolução de Problemas, de Ernesto Costa.

# ESTRUTURA

Melhores jogadores NBA.csv: dados relativos aos 50 melhores jogadores de NBA na temporada 2021/2022. Estes dados foram recolhidos 12/06/2022.



# API'S USADOS - NOTAS TÉCNICAS

Primeiramente apresentamos o HTML (MSN). Este apenas tem a tabela que pretendemos analisar por isso, transferimo-la para csv.

Apresentação do HTML: #df = pd.read\_html ('<https://www.msn.com/pt-br/esportes/basquete/nba/estatisticas-do-jogador?ocid=StripeOCID>').

HTML para csv: código feito através de pesquisa no youtube: #df[0].to\_csv('Jogadores em destaque NBA') Baseado no video: <https://www.youtube.com/watch?v=ODNMNwgtehk&t=213s> ('Scrape HTML tables easily with Pandas and Python').

Após a obtenção do código em csv passamos para a análise, ou seja, averiguamos se faltavam dados na tabela ( df.info() ) e comprovamos que a mesma se encontrava corretamente preenchida, sem dados em falta. Assim ficamos a saber com que tipos de dados iríamos trabalhar.

## DICIONÁRIO DE DADOS

NOME	SIGNIFICADO	TIPO DE DADOS
Posição	Classificação dos melhores jogadores.	int64
Jogador	Nome do jogador (primeiro e último).	object
Equipa	Nome da equipa de onde cada jogador faz parte.	object
Partidas jogadas	Número de jogos que cada jogador fez ao longo da época.	int64
Minutos por jogo	Média de minutos que cada atleta jogou por partida.	float64
Pontos por jogo	Média de pontos que cada atleta marcou por partida.	float64
Pontos totais	Pontos que cada atleta marcou no total de jogos.	int64

Cesta de Campo	Arremessos convertidos durante as jogadas normais de jogo.	int64
Arremessos de 3 pontos convertidos	Arremesso certo realizado além da linha de 3 pontos.	int64
Turn overs	Acontece quando o jogador perde a posse de bola para a equipa adversária.	int64

# ANÁLISE DE DADOS

```
In [1]: import pandas as pd      #importar o pandas para ler o documento csv
         import numpy as np     #importar o numpy pois este é uma biblioteca para operações matemáticas, possuindo assim muitas funções que facilitam as operações com arrays.
```

```
In [2]: import matplotlib.pyplot as plt  #importar o matplotlib.pyplot para a elaboração dos gráficos
```

```
In [3]: #criação de uma variável que guarda os dados do csv "Melhores jogadores NBA".
#o pandas é usado para ler o ficheiro csv

df = pd.read_csv('Melhores jogadores NBA.csv')
```

```
In [4]: #para sabermos com que tipo dados vamos trabalhar, usamos a função info().
#assim esta dá-nos o tipo de dados, sejam eles int64, float64, object, etc.

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype  
--- 
 0   Player     50 non-null    int64  
 1   Unnamed: 1  50 non-null    object  
 2   Team       50 non-null    object  
 3   GP          50 non-null    int64  
 4   MPG         50 non-null    float64 
 5   PPG         50 non-null    float64 
 6   TP          50 non-null    int64  
 7   FGM         50 non-null    int64  
 8   FG%         50 non-null    float64 
 9   3PM         50 non-null    int64  
 10  TO          50 non-null    int64  
 11  PF          50 non-null    int64  
dtypes: float64(3), int64(7), object(2)
memory usage: 4.8+ KB
```

Para dar inicio ao código deste projeto importamos o pandas para ler o documento csv, também importamos o numpy e o matplotlib.pyplot para a elaboração dos gráficos.

Criamos então uma variável que guarda todos os dados em csv “df”. **df = pd.read\_csv( )**

Para saber com que tipo de dados iamos trabalhar, usamos a função **info( )** pois a mesma dá-nos o tipo de dados, sejam eles int64, float64, object, etc. **df.info( )**

```
In [5]: #a função head(), permite-nos visualizar em formato tabela as 5 primeiras linhas da tabela
#no nosso caso, esta permite-nos ver quem são os 5 melhores jogadores da NBA da temporada 21/22

df.head()
```

Out[5]:

	Player	Unnamed: 1	Team	GP	MPG	PPG	TP	FGM	FG%	3PM	TO	PF
0	1	Joel Embiid	76ers	68	33.8	30.6	2079	666	49.9	93	214	181
1	2	LeBron James	Lakers	56	37.2	30.3	1695	640	52.4	161	196	121
2	3	Giannis Antetokounmpo	Bucks	67	32.9	29.9	2002	689	55.3	71	219	212
3	4	Kevin Durant	NETS	55	37.2	29.9	1643	578	51.8	115	191	113
4	5	Luka Doncic	Dallas	65	35.4	28.4	1847	641	45.7	201	292	145

Com a função head( ) podemos visualizar, no formato tabela, as 5 primeiras linhas da tabela que corresponde aos melhores jogadores da NBA 21/22.  
df.head( )

```
In [7]: #para uma melhor análise dos dados, vamos retirar duas colunas, pois não achamos pertinentes para as conclusões e as
#a função drop permite-nos retirar da tabela as colunas que desejamos, ou seja, a "FG%" e a "PF"

df = df.drop(columns=['FG%', 'PF'])
```

In [7]: #usamos novamente a função head para percebermos como ficam os dados dispostos na tabela sem as colunas retiradas anteriormente  
df.head()

Out[7]:

	Player	Unnamed: 1	Team	GP	MPG	PPG	TP	FGM	3PM	TO
0	1	Joel Embiid	76ers	68	33.8	30.6	2079	666	93	214
1	2	LeBron James	Lakers	56	37.2	30.3	1695	640	161	196
2	3	Giannis Antetokounmpo	Bucks	67	32.9	29.9	2002	689	71	219
3	4	Kevin Durant	NETS	55	37.2	29.9	1643	578	115	191
4	5	Luka Doncic	Dallas	65	35.4	28.4	1847	641	201	292

```
In [8]: #mesmo tendo retirado as colunas, existem ainda elementos que precisam de ser modificados para que a realização deste possa ser feita
#para isso vamos renomear as colunas da tabela
#depois de alteradas, usamos novamente a função head para ver as alterações feitas
df.columns = ['Posição', 'Jogador', 'Equipa', 'Partidas jogadas', 'Minutos por jogo', 'Pontos por jogo', 'Pontos totais', 'Cesta de Campo', 'Arremessos de 3 pontos convertidos', 'Turn overs']

df.head()
```

Out[8]:

	Posição	Jogador	Equipa	Partidas jogadas	Minutos por jogo	Pontos por jogo	Pontos totais	Cesta de Campo	Arremessos de 3 pontos convertidos	Turn overs
0	1	Joel Embiid	76ers	68	33.8	30.6	2079	666	93	214
1	2	LeBron James	Lakers	56	37.2	30.3	1695	640	161	196
2	3	Giannis Antetokounmpo	Bucks	67	32.9	29.9	2002	689	71	219
3	4	Kevin Durant	NETS	55	37.2	29.9	1643	578	115	191
4	5	Luka Doncic	Dallas	65	35.4	28.4	1847	641	201	292

```
In [9]: #para iniciar a compreensão da tabela, e para termos uma maior noção dos dados, usamos a função describe que nos permite analisar os dados

df.describe()
```

Out[9]:

	Posição	Partidas jogadas	Minutos por jogo	Pontos por jogo	Pontos totais	Cesta de Campo	Arremessos de 3 pontos convertidos	Turn overs
count	50.00000	50.00000	50.00000	50.00000	50.00000	50.00000	50.00000	50.00000
mean	25.50000	60.54000	34.22600	23.05800	1398.48000	498.08000	130.34000	169.20000
std	14.57738	14.271722	2.156037	3.656869	407.849974	144.829082	68.60791	67.020862
min	1.00000	29.00000	27.70000	18.30000	590.000000	222.000000	0.00000	43.000000
25%	13.25000	55.25000	33.50000	20.10000	1183.000000	406.250000	78.25000	126.250000
50%	25.50000	65.50000	34.50000	22.40000	1420.000000	517.000000	114.50000	176.000000
75%	37.75000	70.00000	35.37500	25.80000	1633.750000	592.000000	182.75000	204.500000
max	50.00000	80.00000	37.90000	30.60000	2155.000000	774.000000	285.00000	303.000000

Para conseguirmos analisar melhor os dados da tabela, tivemos de remover que não achamos pertinentes para as conclusões. Usamos então a função drop para retirar as colunas indesejadas. **df = df.dropcolumns = ['FG%', 'PF']**

Voltamos a usar a função head( ) para ver as alterações feitas na tabela.

Mesmo depois de retirar as colunas ainda queríamos fazer algumas alterações de forma a tornar a tabela mais simples de perceber, com esse objetivo renomeamos as colunas da tabela e depois voltamos a usar a função head( ).

Para iniciar a compreensão da tabela usamos a função describe( ) **df.describe( )**

```
In [10]: #Qual a equipa com mais jogadores no top 50 de melhores jogadores de NBA da temporada 21/22?  
#usando a função value_counts() apenas na coluna "Equipa", esta conta quantos jogadores da mesma equipa estão presentes  
df['Equipa'].value_counts()
```

```
Out[10]: Warriors      4  
Miami          3  
Lakers         3  
Bulls          3  
76ers          2  
Pelicans       2  
Kings          2  
Clippers       2  
Timberwolves   2  
Hornets         2  
Celtics         2  
Knicks          2  
Grizzlies      2  
Dallas          2  
NETS            2  
Bucks            2  
Raptors         2  
Spurs           1  
Pistons         1  
Cavaliers       1  
Oklc             1  
Wizards          1  
Portland        1  
Jazz             1  
Suns             1  
Nuggets          1  
Atlanta          1  
Pacers           1  
Name: Equipa, dtype: int64
```

```
In [11]: #reparamos anteriormente que a equipa com mais jogadores no ranking que estamos a estudar é a Warriors pois esta tem 4  
#no entanto, isto não quer dizer que esta equipa tenha mais jogadores no top 10, e por isso?
```

```
#Qual a equipa com mais jogadores no top 10 melhores jogadores?
```

```
df.head(10)[ 'Equipa' ].value_counts()
```

```
Out[11]: NETS      2  
76ers     1  
Lakers    1  
Bucks     1  
Dallas    1  
Atlanta   1  
Bulls     1  
Grizzlies 1  
Nuggets   1  
Name: Equipa, dtype: int64
```

Qual a equipa com mais jogadores no top 50 de melhores jogadores de NBA da temporada 21/22? Usando a função `value_counts()` na coluna da “Equipa” conseguimos ver o nº de jogadores que estam na mesma equipa no TOP50 melhores jogadores 21/22. Reparamos que a quipa com mais jogadores era o Warriors que continha 4 jogadores no TOP50 mas isso queria dizer que também tinha mais jogadores no TOP10? A resposta era não, chegamos a essa conclusão depois de usar a função `df.head(10) ['Equipa'].value_counts()` e perceber que o Nets eram quem mais tinha jogadores no TOP10.

In [12]: #Qual o jogador com mais arremessos de 3 pontos convertidos?

```
#para vermos qual é o jogador com mais arremessos de 3 pontos convertidos, começamos por criar uma variavel que vai guardar o valor máximo de arremessos é nos dado através do uso da função .max()
#seguidamente criamos uma variavel que armazena o jogador a que equivale o número máximo de arremessos guardado anteriormente
#finalmente fazemos print da linha em que se encontra o jogador na tabela dos 50 melhores e o seu nome

mais_arremessosconvertidos= df['Arremessos de 3 pontos convertidos'].max()
jogador_maisarremessosconvertidos = df.loc[(df['Arremessos de 3 pontos convertidos'] == mais_arremessosconvertidos)]
jogador_maisarremessosconvertidos
```

```
print('A linha e o nome do jogador com mais arremessos convertidos são: ', jogador_maisarremessosconvertidos['Jogador'])
```

A linha e o nome do jogador com mais arremessos convertidos são: 13 Stephen Curry  
Name: Jogador, dtype: object

In [13]: #os arremessos de 3 pontos convertidos são, no basket os pontos mais altos que se podem fazer num só lance.  
#Mas será que o jogador com mais arremessos de 3 pontos convertidos, é tambem o jogador com mais pontos em toda a temporda?

```
#para averiguar começamos por criar uma variável que guarda o numero máximo de pontos totais do ranking de 50 melhores
#o numero máximo é nos dado através do uso da função max()
##seguidamente criamos uma variavel que armazena o jogador a que equivale o número máximo de pontos totais guardado anteriormente

#finalmente fazemos print da linha em que se encontra o jogador na tabela dos 50 melhores e o seu nome

maisPontos= df['Pontos totais'].max()
jogadorMaisPontos = df.loc[(df['Pontos totais'] == maisPontos)]
jogadorMaisPontos
```

```
print('A linha e nome do jogador com mais pontos são: ', jogadorMaisPontos['Jogador'])
```

A linha e nome do jogador com mais pontos são: 5 Trae Young  
Name: Jogador, dtype: object

In [14]: #Com o que vimos anteriormente podemos concluir que não é por ter mais arremessos de 3 pontos convertidos que significa que o jogador tem mais pontos totais por ter jogado mais partidas que os outros?  
#vamos averiguar:

```
#Qual o jogador com mais partidas jogadas?
#realizamos o mesmo processo que fizemos anteriormente.
maisJogos = df['Partidas jogadas'].max()
jogadorMax = df.loc[(df['Partidas jogadas'] == maisJogos)]
jogadorMax
```

Out[14]:

Posição	Jogador	Equipa	Partidas jogadas	Minutos por jogo	Pontos por jogo	Pontos totais	Cesta de Campo	Arremessos de 3 pontos convertidos	Turn overs
35	36	Miles Bridges	Hornets	80	35.5	20.2	1613	596	154

Como podemos ver nestes resultamos chegamos a conclusão que cada dado vai corresponder a um jogador diferente e todos estes dados só foram possiveis devido ao codigo usado. `mais_...= df['...'].max()`  
`jogador_... = df.loc[(df['...']== ...)]`  
`jogador_...`

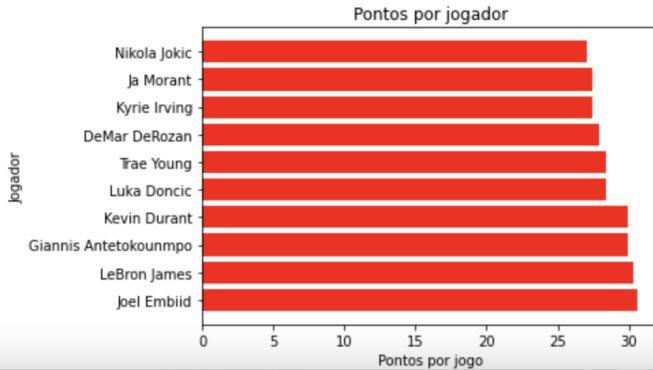
```
In [16]: #A média de pontos marcados por jogo ao longo desta temporada é muito diferente entre os 10 melhores jogadores?

#%matplotlib notebook
%matplotlib inline

#neste gráfico vamos fazer uma relação entre os Jogadores e a média de pontos que estes fizeram ao longo da época
#criar as variáveis que vão ser relacionadas: Jogador e Pontos
#a função head(10) vai à tabela buscar os dados dos 10 melhores jogadores.
#neste caso, a função vai buscar os nomes dos jogadores e a média de pontos por jogo que estes fizeram
#estes dados são adicionados em forma de lista (devido ao uso da função .tolist()) às variaveis criadas ("Jogador" e "Pontos")

Jogador = df ['Jogador'].head(10).tolist()
Pontos = df ['Pontos por jogo'].head(10).tolist()

#Criar o gráfico:
#No eixo do x encontram-se os pontos, no eixo do y os jogadores, e escolhemos a cor vermelha para as barras
plt.barh(Jogador, Pontos, color='red')
plt.ylabel("Jogador") #nome do eixo vertical, eixo do y
plt.xlabel("Pontos por jogo") #nome do eixo horizontal, eixo do x
plt.title('Pontos por jogador') #título do gráfico
plt.show()
```



Fizemos um grafico para descobrir se a média de pontos marcados por jogo era muito diferente entre os 10 melhores jogadores. Usamos as variaveis relacionadas aos jogadores e pontos para conseguirmos elaborar o gráfico.

**Jogador = df ['Jogador'].head(10).tolist()**

**Pontos = df ['Pontos por jogo'].head(10).tolist()**

Criamos um gráfico com o nome dos jogadores no eixo y e a média de pontos no eixo x.

**plt.barh(Jogador, Pontos, color='red')**

**plt.ylabel("Jogador") #nome do eixo vertical, eixo do y**

**plt.xlabel("Pontos por jogo") #nome do eixo horizontal, eixo do x**

**plt.title('Pontos por jogador') #título do gráfico**

**plt.show()**

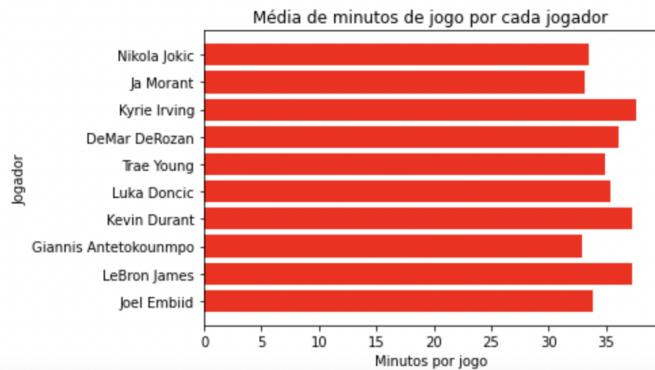
```
In [18]: #A média de minutos por jogo ao longo desta temporada é muito diferente entre os 10 melhores jogadores?
```

```
#%matplotlib notebook
%matplotlib inline

#neste gráfico vamos fazer uma relação entre os Jogadores e a média de minutos que estes fizeram por jogo ao longo da época
#criar as variáveis que vão ser relacionadas: Jogador e Minutos
Jogador = df ['Jogador'].head(10).tolist()
Minutos = df ['Minutos por jogo'].head(10).tolist()

#Criar o gráfico:
#No eixo do x encontram-se os jogadores, no eixo do y os minutos, e escolhemos a cor azul para as barras

plt.barh(Jogador, Minutos, color='red')
plt.ylabel("Jogador") #nome do eixo vertical, eixo do y
plt.xlabel("Minutos por jogo") #nome do eixo horizontal, eixo do x
plt.title('Média de minutos de jogo por cada jogador') #título do gráfico
plt.show()
```



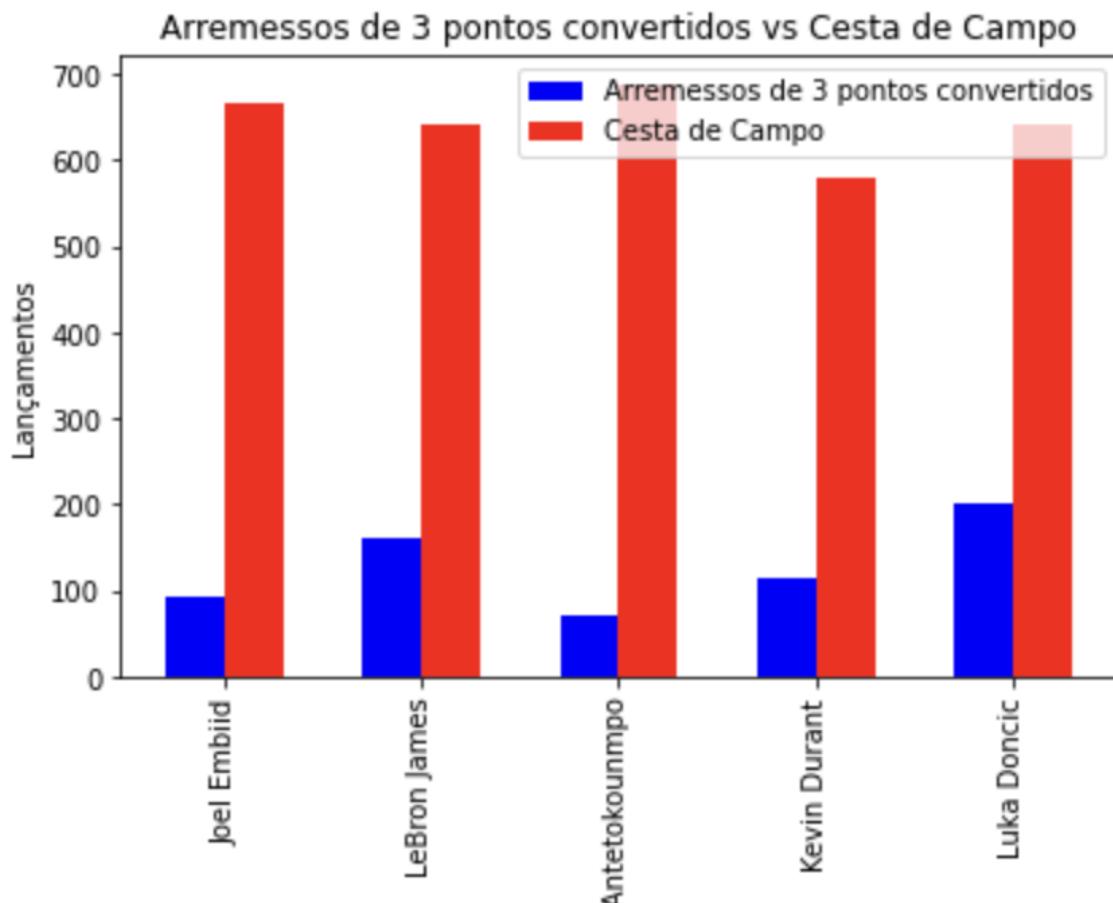
Da mesma forma que criamos o gráfico anterior também criamos outro gráfico mas desta vez para a média de minutos por jogo.

```
In [20]: #dados que vão ser estudados: tentativas de arremessos e os arremessos convertidos
#relacionar as tentativas de arremessos com os arremessos convertidos dos 5 melhores jogadores
jogador = df ['Jogador'].head().tolist()
arremessos = df ['Arremessos de 3 pontos convertidos'].head().tolist()
cesta = df ['Cesta de Campo'].head().tolist()

x = np.arange(len(jogador)) #localizações dos jogadores no gráfico
largura = 0.30 #largura das barras

#criar duas barras com valores diferentes
fig, ax = plt.subplots()
rects1 = ax.bar(x - largura/2, arremessos, largura, label='Arremessos de 3 pontos convertidos', color='blue')
rects2 = ax.bar(x + largura/2, cesta, largura, label='Cesta de Campo', color='red')
#x - largura/2 : coluna do lado esquerdo, que representa as tentativas de lançamentos.
#x + largura/2 : coluna do lado direito, que representa os lançamentos convertidos
ax.set_ylabel('Lançamentos')
ax.set_title('Arremessos de 3 pontos convertidos vs Cesta de Campo')
ax.set_xticks(x)
ax.set_xticklabels(jogador)
ax.legend()

fig.tight_layout() #espaço entre nome dos jogadores no gráfico para não existir sobreposição
plt.xticks(rotation='vertical')
plt.show()
```



O objetivo deste gráfico é ver o número de arremessos de 3 pontos convertidos e o número de cesta de campo dos 5 melhores jogadores da temporada 21/22. Dicidimos fazer um tabela vertical do mesmo estilo das anteriores. Fizemos a barra de 3 pontos a azul e a cesta de campo a vermelha para podermos distinguir as duas.

```
jogador = df ['Jogador'].head().tolist()
arremessos = df ['Arremessos de 3 pontos convertidos'].head().tolist()
cesta = df ['Cesta de Campo'].head().tolist()

x = np.arange(len(jogador)) #localizações dos jogadores no grafico
largura = 0.30 #largura das barras

fig, ax = plt.subplots()
rects1 = ax.bar(x - largura/2, arremessos, largura, label='Arremessos de 3
pontos convertidos', color='blue')
rects2 = ax.bar(x + largura/2, cesta, largura, label='Cesta de Campo', co-
lor='red')
```

```

ax.set_ylabel('Lançamentos')
ax.set_title('Arremessos de 3 pontos convertidos vs Cesta de Campo')
ax.set_xticks(x)
ax.set_xticklabels(jogador)
ax.legend()

fig.tight_layout()
plt.xticks(rotation='vertical')
plt.show()

```

```

In [21]: #objetivo deste gráfico: podemos comparar os resultados do melhor e do pior jogador do top 50
#vamos comparar usando vários dados:
    ##número de partidas jogadas
    ##média de minutos que fez por jogo
    ##média de pontos por jogo
    ##cesta de campo
    ##Arremessos de 3 pontos convertidos
    ##LTurn overs

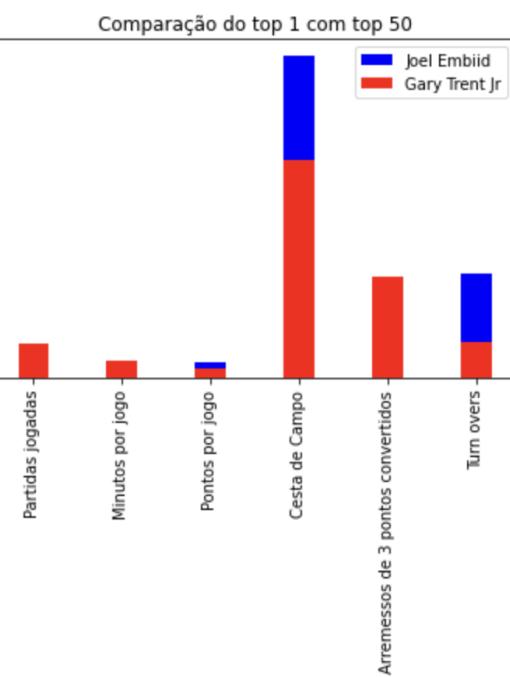
labels = ['Partidas jogadas', 'Minutos por jogo', 'Pontos por jogo', 'Cesta de Campo', 'Arremessos de 3 pontos convertidos']

#dados usados para comparação, extraídos da tabela
Joel_Embiid = [68, 33.8, 30.6, 666, 93, 214]
Gary_Trent_jr = [70, 35, 18.3, 450, 209, 73]
largura = 0.35 #largura das barras
fig, ax = plt.subplots()

#Criação das barras correspondentes aos valores dos jogadores
ax.bar(labels, Joel_Embiid, largura, label= 'Joel Embiid', color='blue')
ax.bar(labels, Gary_Trent_jr, largura, label= 'Gary Trent Jr', color='red')

ax.set_ylabel('')
ax.set_title('Comparação do top 1 com top 50')
ax.legend()
plt.xticks(rotation='vertical') #colocar o nome dos dados que estamos a comparar na vertical, para tornar mais percertivo
plt.show()

```



De forma semelhante ao ultimo gráfico e para completar o nosso projeto elaboramos um gráfico para comparar os dados do 1º jogador e do ultimo da tabela.

# Poster

## OS 50 MELHORES JOGADORES DE NBA NA ÉPOCA 2021/2022



### Equipas com jogadores do top 10 melhores jogadores

2	NETS
1	Atlanta
1	Grizzlies
1	Lakers
1	76ers
1	Bulls
1	Nuggets
1	Dallas
1	Bucks

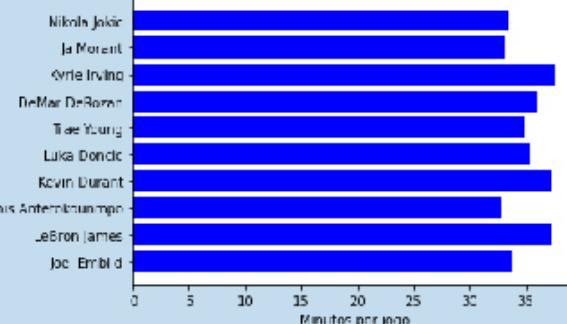
### Jogador com mais pontos marcados



Trae Young

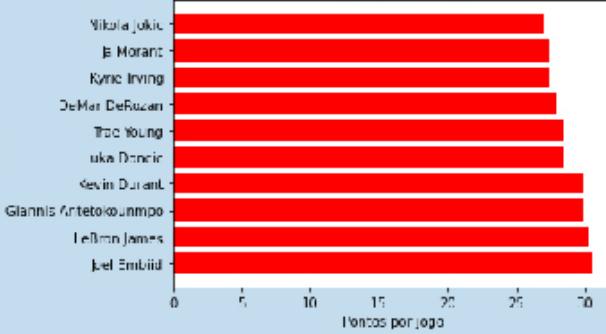
### Jogadores e a média de minutos que estes fizeram por jogo ao longo da época

Média de minutos de jogo per cada jogador



### Jogadores e a média de pontos que estes fizeram ao longo da época

Pontos por jogador



# OS 50 MELHORES JOGADORES DE NBA NA ÉPOCA 2021/2022



## TOP 3



Joel Embiid LeBron James Giannis Antetokounmpo

## Equipas com jogadores do top 10 melhores jogadores

2	NETS
1	Atlanta
1	Grizzlies
1	Lakers
1	76ers
1	Bulls
1	Nuggets
1	Dallas
1	Bucks

## Jogador com mais pontos marcados



Trae Young

Jogadores e a média de minutos que estes fizeram por jogo ao longo da época



Jogadores e a média de pontos que estes fizeram ao longo da época

