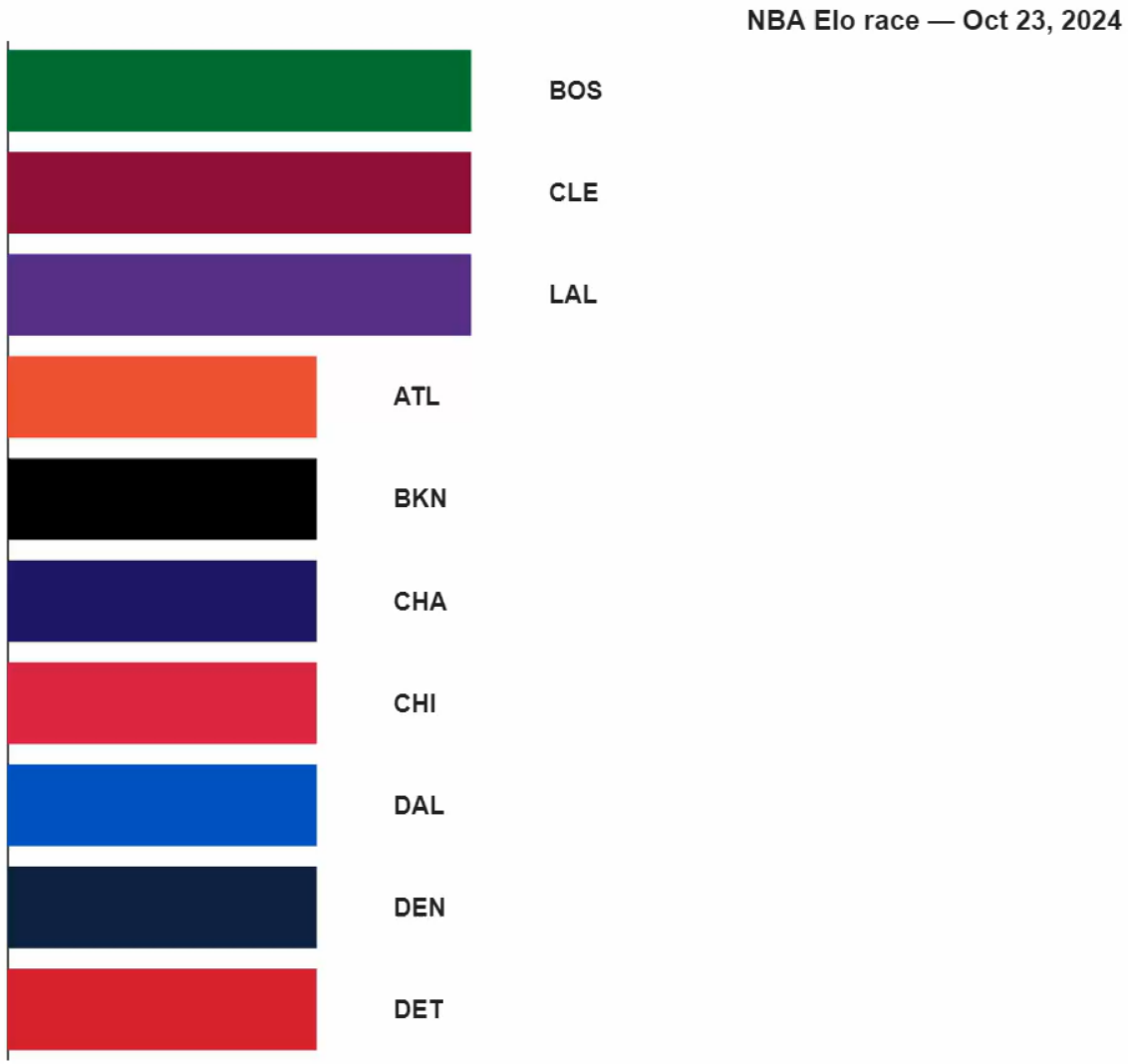
MATLAB WITH PYTHON NBA API


NBA Elo race — Oct 23, 2024
BOS
CLE
LAL
ATL
BKN
CHA
CHI
DAL
DEN
DET

DATASET

Find a dataset to analyze the NBA championship

PLAYERS

See All Player Stats

Daily Leaders | SEASON LEADERS

06/08/2025

POINTS PER GAME

1.	**Giannis Antetokounmpo** MIL	**33.0**
2.	Shai Gilgeous-Alexander OKC	30.4
3.	Luka Dončić LAL	30.2
4.	Donovan Mitchell CLE	29.6
5.	Jalen Brunson NYK	29.4

REBOUNDS PER GAME

1.	**Giannis Antetokounmpo** MIL	**15.4**
2.	Nikola Jokić DEN	12.7
3.	Alperen Sengun HOU	11.9
4.	Karl-Anthony Towns NYK	11.6
5.	Jayson Tatum BOS	11.5

ASSISTS PER GAME

1.	**Tyrese Haliburton** IND	**9.3**
2.	James Harden LAC	9.1
3.	Cade Cunningham DET	8.7
4.	Nikola Jokić DEN	8.0
5.	Jalen Brunson NYK	7.0

BLOCKS PER GAME

1.	**Zach Edey** MEM	**2.5**
2.	Myles Turner IND	2.2
3.	Chet Holmgren OKC	1.9
4.	LeBron James LAL	1.8
5.	Nicolas Batum LAC	1.7

STEALS PER GAME

1.	**Gary Trent Jr.** MIL	**2.6**
2.	Jayson Tatum BOS	2.1
3.	OG Anunoby NYK	2.0
3.	LeBron James LAL	2.0
3.	Nikola Jokić DEN	2.0

FIELD GOAL PERCENTAGE

1.	**Jarrett Allen** CLE	**72.1**
2.	Ivica Zubac LAC	65.9
3.	Jalen Duren DET	65.0
4.	Isaiah Hartenstein OKC	61.9
5.	Davion Mitchell MIA	61.0

THREE POINTERS MADE

1.	**Aaron Nesmith** IND	**50**
2.	Jalen Brunson NYK	48
3.	Anthony Edwards MIN	46
4.	Tyrese Haliburton IND	43
5.	OG Anunoby NYK	41

THREE POINT PERCENTAGE

1.	**Vince Williams Jr.** MEM	**57.1**
2.	AJ Green MIL	51.4
3.	Davion Mitchell MIA	50.0
3.	Gary Trent Jr. MIL	50.0
5.	Aaron Nesmith IND	49.5

FANTASY POINTS PER GAME

1.	**Giannis Antetokounmpo** MIL	**62.8**
2.	Nikola Jokić DEN	58.1
3.	Jayson Tatum BOS	55.4
4.	LeBron James LAL	53.4
5.	Cade Cunningham DET	52.2

BETA

DAILY DUNK SCORE LEADERS

06/08/2025

Dunk Score ranks in-game dunks by calculating and modelling their physical and stylistic traits.

1.	**Myles Turner**	**71.5**
2.	Chet Holmgren	60.1
3.	Tyrese Haliburton	44.7
4.	Chet Holmgren	22.6

MORE STATS: PLAYOFFS

TOTAL POINTS

Shai Gilgeous-Alexander	548
Jalen Brunson	530
Karl-Anthony Towns	386

TOTAL REBOUNDS

Karl-Anthony Towns	209
Nikola Jokić	178
Josh Hart	158

TOTAL ASSISTS

Tyrese Haliburton	168

README MIT license

nba_api

An API Client Package to Access the APIs of NBA.com

`nba_api` is an API Client for `www.nba.com`. This package intends to make the APIs of NBA.com easily accessible and provide extensive documentation about them.

Getting Started

`nba_api` requires Python 3.7+ along with the `requests` and `numpy` packages. While `pandas` is not required, it is required to work with Pandas DataFrames.

```
pip install nba_api
```

NBA Official Stats

```
from nba_api.stats.endpoints import playercareerstats

# Nikola Jokić
career = playercareerstats.PlayerCareerStats(player_id='203999')
```

Languages

- Python 100.0%

DATASET 📈

games = *10749x8 table*

	GAME_ID	GAME_DATE	MATCHUP	WL	PTS	PLUS_MINUS	TEAM	OPPONENT
1	21500002	2015-10-27	'CHI vs. CLE'	'W'	97	2	'CHI'	'CLE'
2	21500001	2015-10-27	'ATL vs. DET'	'L'	94	-12	'ATL'	'DET'
3	21500003	2015-10-27	'GSW vs. NOP'	'W'	111	16	'GSW'	'NOP'
4	21500014	2015-10-28	'PHX vs. DAL'	'L'	95	-16	'PHX'	'DAL'
5	21500015	2015-10-28	'POR vs. NOP'	'W'	112	18	'POR'	'NOP'
6	21500013	2015-10-28	'OKC vs. SAS'	'W'	112	6	'OKC'	'SAS'
7	21500005	2015-10-28	'BOS vs. PHI'	'W'	112	17	'BOS'	'PHI'
8	21500016	2015-10-28	'SAC vs. LAC'	'L'	104	-7	'SAC'	'LAC'
9	21500010	2015-10-28	'HOU vs. DEN'	'L'	85	-20	'HOU'	'DEN'

PLAYERS

```
career = py.nba_api.stats.endpoints.playercareerstats.PlayerCareerStats(player_id=string(player_id));
c = career.get_data_frames();
table(c{1})
```

ans = *8×27 table*

	PLAYER_AGE	GP	GS	MIN	FGM	FGA	FG_PCT	FG3M	FG3A
1	20	80	80	2443	397	835	0.4750	105	24
2	21	79	79	2455	466	1036	0.4500	116	31
3	22	66	66	2265	552	1226	0.4500	189	46
4	23	64	64	2290	605	1318	0.4590	187	48
5	24	76	76	2731	708	1564	0.4530	230	65
6	25	74	74	2732	727	1559	0.4660	240	68
7	26	74	74	2645	672	1426	0.4710	229	60
8	27	72	72	2624	662	1465	0.4520	250	72

TEAMS 👥

```
team = BOS ;
team_id = T{teams == team,"id"};
% team_id = 1610612738 % celtics https://www.nba.com/stats/team/1610612738
game_log = py.nba_api.stats.endpoints.teamgamelog.TeamGameLog(team_id=team_id, season='2024-25').get_data_frames();
table(game_log{1})
```

ans = *82×27 table*

	Team_ID	Game_ID	GAME_DATE	MATCHUP	WL	W	L	W_PCT	M
3	1610612738	"0022401156"	"APR 09, 2025"	"BOS @ ORL"	"L"	59	21	0.7360	
4	1610612738	"0022401151"	"APR 08, 2025"	"BOS @ NYK"	"W"	59	20	0.7470	
5	1610612738	"0022401137"	"APR 06, 2025"	"BOS vs. WAS"	"W"	58	20	0.7440	
6	1610612738	"0022401120"	"APR 04, 2025"	"BOS vs. PHX"	"W"	57	20	0.7400	
7	1610612738	"0022401106"	"APR 02, 2025"	"BOS vs. MIA"	"L"	56	20	0.7370	
8	1610612738	"0022401093"	"MAR 31, 2025"	"BOS @ MEM"	"W"	56	19	0.7470	
9	1610612738	"0022401080"	"MAR 29, 2025"	"BOS @ SAS"	"W"	55	19	0.7430	
10	1610612738	"0022401058"	"MAR 26, 2025"	"BOS @ PHX"	"W"	54	19	0.7400	
11	1610612738	"0022401044"	"MAR 24, 2025"	"BOS @ SAC"	"W"	53	19	0.7360	

GAMES 🏀

```
season = "2024-25";
```

Run Python Code

Autorun

`games2` = Run Python code

Select input type

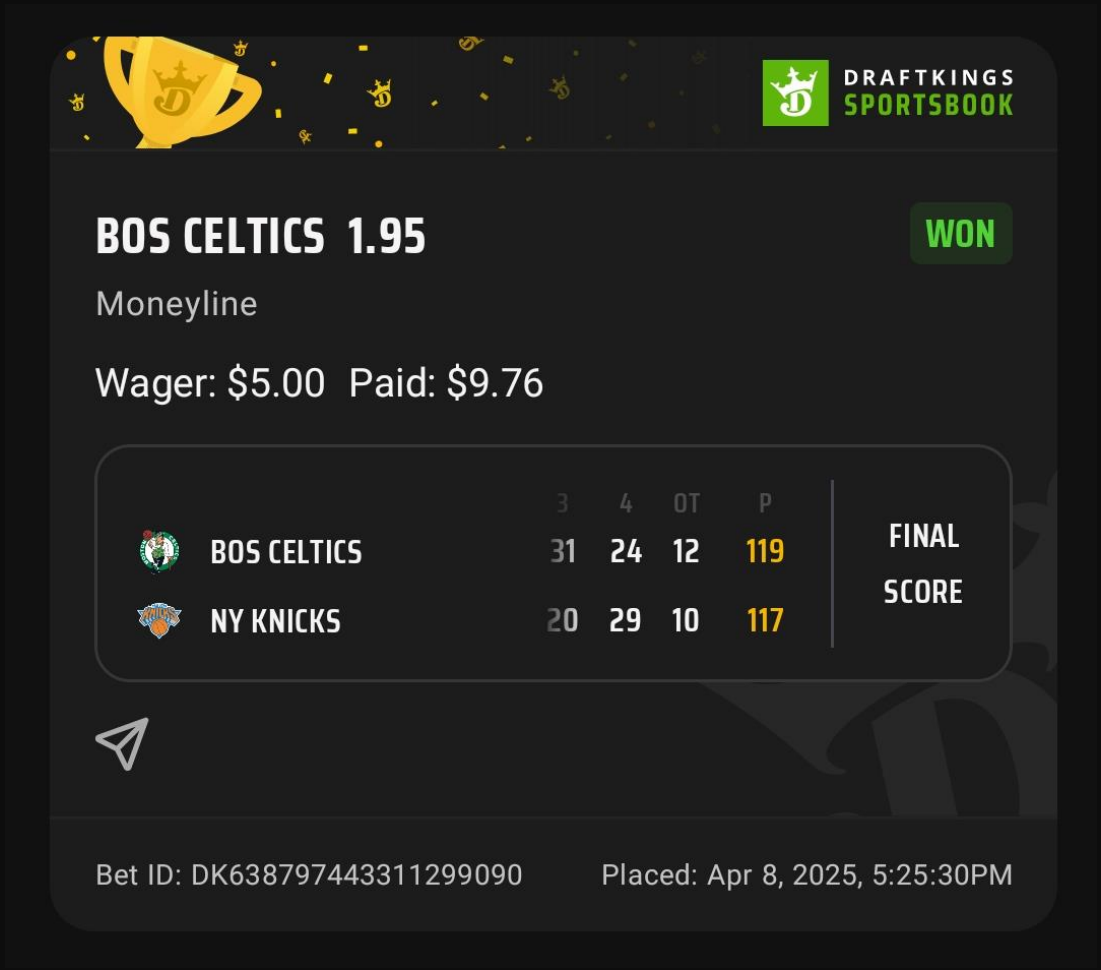
- (•) Code
- ( ) File

Enter Python code

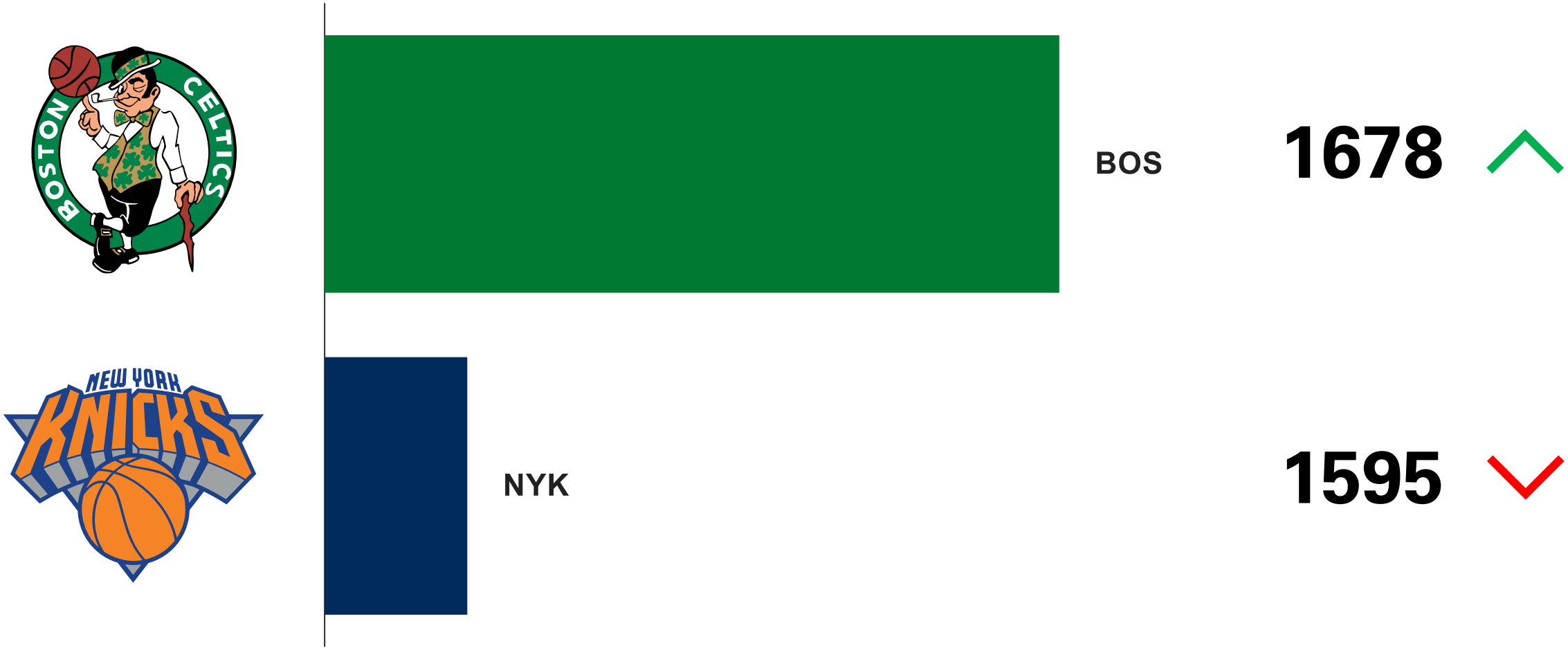
```
from nba_api.stats.endpoints import LeagueGameLog
gl = LeagueGameLog(season=season)
games = gl.get_data_frames()[0]
```

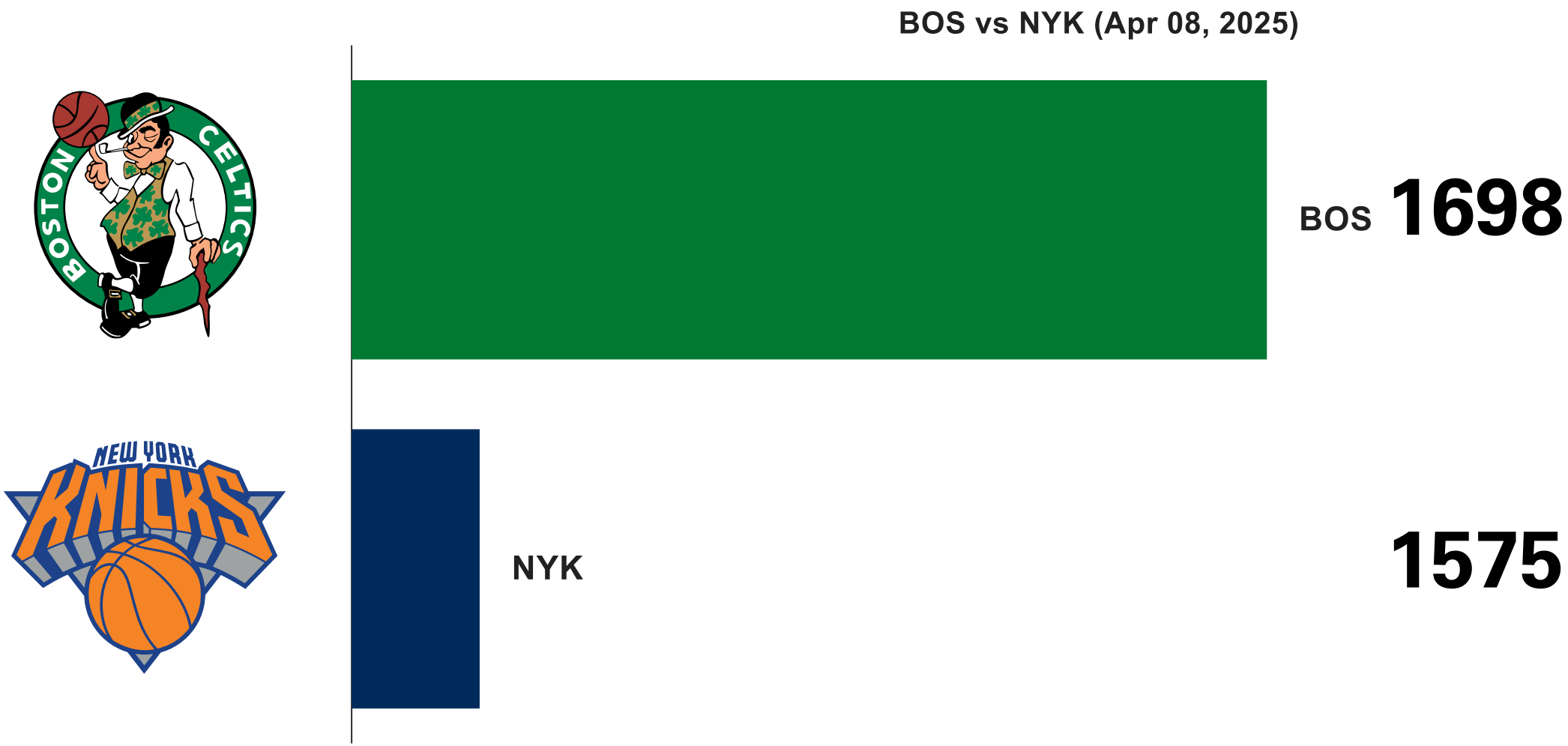
Output options

WHAT ARE THE ODDS?



BOS vs NYK (Apr 08, 2025)




BOS vs NYK (Apr 08, 2025)
BOS 1698
NYK 1575

ELO RATING

Elo rating is a method for calculating the *relative skill levels* of players and teams in competitor-vs-competitor games like chess, and it has been adapted for many sports, including **basketball**.

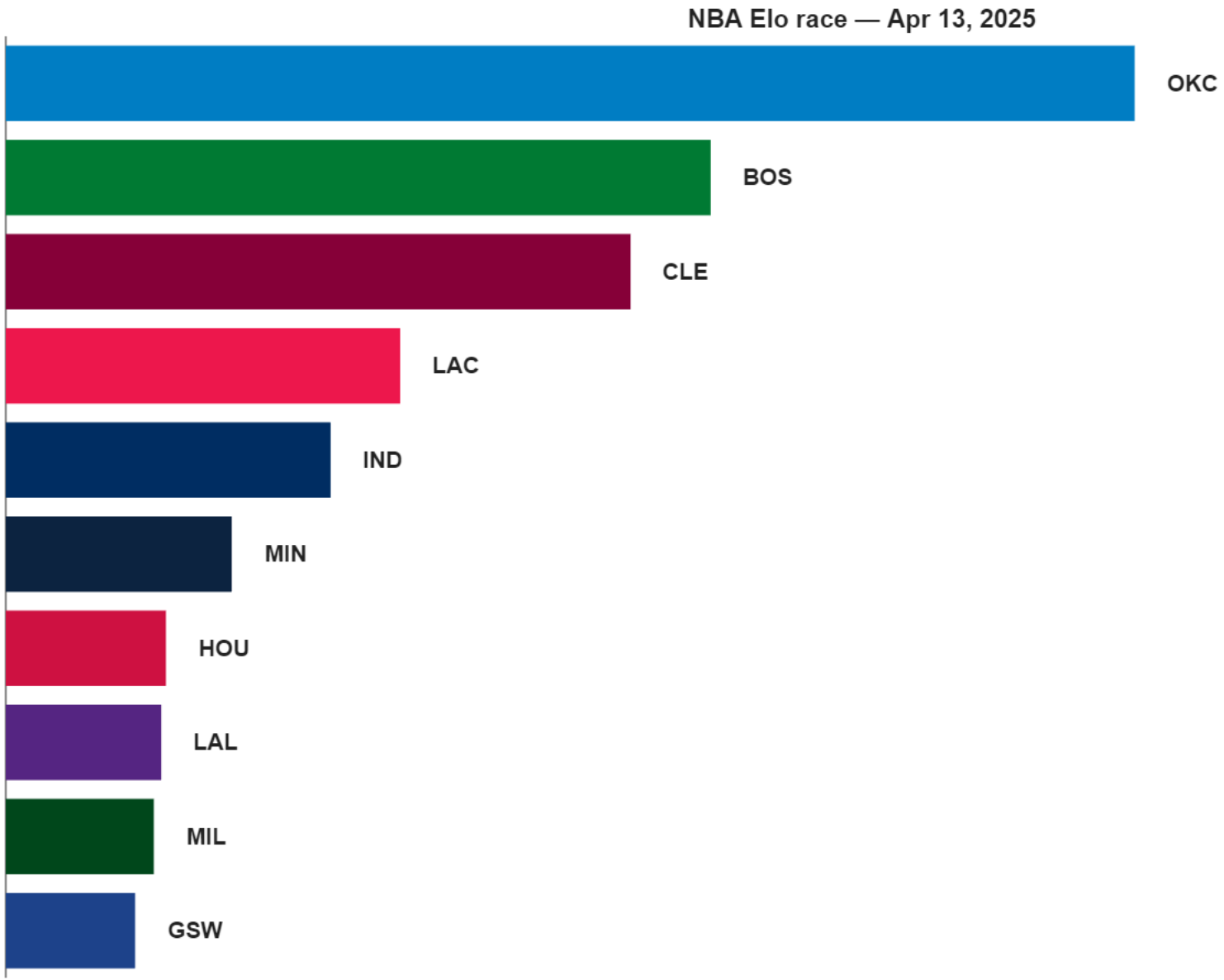
$$R' = R + K \times \left(S - \frac{1}{1 + 10^{\frac{(R_{\text{opponent}} - R)}{400}}} \right)$$

🎲 ELO RANK, PROBABILITIES & ODDS

```
function p = calculate_win_probability(elo_a, elo_b)
    % Compute win probability for team A given two Elo ratings
    p = 1 / (1 + 10^((elo_b - elo_a) / 400));
end

function odds = probability_to_odds(probability)
    % Convert probability to decimal odds
    odds = 1 / probability;
end
```

THANKS FOR WATCHING


NBA Elo race — Apr 13, 2025
OKC
BOS
CLE
LAC
IND
MIN
HOU
LAL
MIL
GSW

Elo rating system

From Wikipedia, the free encyclopedia

The **Elo**[a] **rating system** is a method for calculating the relative skill levels of players in zero-sum games such as chess or esports. It is named after its creator Arpad Elo, a Hungarian-American chess master and physics professor.

Arpad Elo, the inventor of the Elo rating system

The Elo system was invented as an improved chess-rating system over the previously used Harkness system,[1] but is also used as a rating system in association football (soccer), American football, baseball, basketball, pool, various board games and esports, and, more recently, large language models.

The difference in the ratings between two players serves as a predictor of the outcome of a match. Two players with equal ratings who play against each other are expected to score an equal number of wins. A player whose rating is 100 points greater than their opponent's is expected to score 64%; if the difference is 200 points, then the expected score for the stronger player is 76%.[2]

A player's Elo rating is a number that may change depending on the outcome of rated games played. After every game, the winning player takes points from the losing one. The difference between the ratings of the winner and loser determines the total number of points gained or lost after a game. If the higher-rated player wins, then only a few rating points will be taken from the lower-rated player. However, if the lower-rated player scores an upset win, many rating points will be transferred. The lower-rated player will also gain a few points from the higher rated player in the event of a draw. This means that this rating system is self-correcting. Players whose ratings are too low or too high should, in the long run, do better or worse

🔢 WHAT IS ELO RATING?

The **Elo system** updates a team's rating based on the expected outcome of a match and the actual outcome. The key ideas:

- Every team starts with a default rating (e.g., 1500).
- A win increases your rating, and a loss decreases it.
- Beating a highly rated team earns you more points than beating a lower-rated one.

🧮 ELO FORMULA

When Team A plays against Team B:

1.Expected Score (E_A):

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}$$

2.**Updated Rating**:

$$R'_A = R_A + K \cdot (S_A - E_A)$$

- R_A: Team A's current rating
- R_B : Team B's current rating
- S_A : Actual score (1 if A wins, 0 if A loses)
- K : A factor controlling sensitivity (commonly 20 in basketball Elo systems)

🏀 HOW ELO IS APPLIED TO BASKET

1. Sort the dataset by *date*
2. Initialize ratings (e.g., all teams start at 1500)
3. For each row:
   - Get ratings for *winner_name* and *loser_name*
   - Compute expected scores
   - Update both ratings based on the match outcome

📋 Example Dataset Columns

records = *10749x8 table*

	date	game_id	team	opponent	elo_pre	elo_opp_pre	elo_diff	win
1	2015-10-27	21500002	'CHI'	'CLE'	1500	1500	100	1
2	2015-10-27	21500001	'ATL'	'DET'	1.5072e+03	1.4928e+03	114.3974	0
3	2015-10-27	21500003	'GSW'	'NOP'	1.4940e+03	1.5060e+03	88.0402	1
4	2015-10-28	21500014	'PHX'	'DAL'	1.5015e+03	1.4985e+03	103.0780	0