

Activity 1 - NBA

You are a data scientist. You work for the NBA and you have the impossible task of answering who is the GOAT of the NBA between Michael Jordan and LeBron James. Use data science to answer this question.

IMPORTANT: There are parts in the code marked as "# TASK: ", you will have to add code after those parts.

Luis Maximiliano López Ramírez - A00833321

Task 1. Load each player data to a Pandas Dataframe (provided)

```
In [30]: import pandas as pd

# Here is a Dictionary with the NBA historical results of LeBron James from 2003 to 2022
lebron_data = {
    'Season': ['2003-04', '2004-05', '2005-06', '2006-07', '2007-08', '2008-09', '2009-10', '2010-11', '2011-12',
    'Team': ['CLE', 'CLE', 'CLE', 'CLE', 'CLE', 'CLE', 'CLE', 'MIA', 'MIA', 'MIA', 'MIA', 'CLE', 'CLE', 'CLE', 'C
    'Games_Played': [79, 80, 79, 78, 75, 81, 76, 79, 62, 76, 77, 69, 76, 74, 82, 55, 67, 45, 56],
    'Points_Per_Game': [20.9, 27.2, 31.4, 27.3, 30.0, 28.4, 29.7, 26.7, 27.1, 26.8, 27.1, 25.3, 25.3, 26.4, 27.5,
}

# TASK: Create a Pandas DataFrame using the lebron_data dictionary

lebron_df = pd.DataFrame(lebron_data)

print(lebron_df)
```

	Season	Team	Games_Played	Points_Per_Game
0	2003-04	CLE	79	20.9
1	2004-05	CLE	80	27.2
2	2005-06	CLE	79	31.4
3	2006-07	CLE	78	27.3
4	2007-08	CLE	75	30.0
5	2008-09	CLE	81	28.4
6	2009-10	CLE	76	29.7
7	2010-11	MIA	79	26.7
8	2011-12	MIA	62	27.1
9	2012-13	MIA	76	26.8
10	2013-14	MIA	77	27.1
11	2014-15	CLE	69	25.3
12	2015-16	CLE	76	25.3
13	2016-17	CLE	74	26.4
14	2017-18	CLE	82	27.5
15	2018-19	LAL	55	27.4
16	2019-20	LAL	67	25.3
17	2020-21	LAL	45	25.0
18	2021-22	LAL	56	30.3

```
In [31]: jordan_data = {
    'Season': ['1984-85', '1985-86', '1986-87', '1987-88', '1988-89', '1989-90', '1990-91', '1991-92', '1992-93',
    'Team': ['CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'NA', 'NA'],
    'Games_Played': [82, 18, 82, 82, 81, 82, 82, 80, 78, 17, 82, 82, 82, 0, 0, 0, 60, 82],
    'Points_Per_Game': [28.2, 22.7, 37.1, 35.0, 32.5, 33.6, 31.5, 30.1, 32.6, 26.9, 30.4, 29.6, 28.7, 0, 0, 0, 22.9, 20.0]
}

# TASK: Create a Pandas DataFrame using the Lebron_data dictionary
jordan_df = pd.DataFrame(jordan_data)

print(jordan_df)
```

	Season	Team	Games_Played	Points_Per_Game
0	1984-85	CHI	82	28.2
1	1985-86	CHI	18	22.7
2	1986-87	CHI	82	37.1
3	1987-88	CHI	82	35.0
4	1988-89	CHI	81	32.5
5	1989-90	CHI	82	33.6
6	1990-91	CHI	82	31.5
7	1991-92	CHI	80	30.1
8	1992-93	CHI	78	32.6
9	1994-95	CHI	17	26.9
10	1995-96	CHI	82	30.4
11	1996-97	CHI	82	29.6
12	1997-98	CHI	82	28.7
13	1998-99	NA	0	0.0
14	1999-00	NA	0	0.0
15	2000-01	NA	0	0.0
16	2001-02	WAS	60	22.9
17	2002-03	WAS	82	20.0

Task 2. First let's see the points per game (ppg) for each player did in their first 15 seasons.

1. Create a Numpy Array with Lebron's first 15 season's points per game named: lebron_ppg. (IMPORTANT: Lebron has 19 seasons so far, you need to take only first 15 seasons)
2. Create a Numpy Array with Jordan's first 15 season's points per game named: jordan_ppg. (IMPORTANT: Jordan did not played for 3 seasons, you need ignore those seasons)
3. Use Matplotlib to graph the points per game per season of both players in a single graph:
 - a) In order to graph this we need another Numpy Array from 1 to 12, named: seasons
 - b) Add required library for Matplotlib --> import matplotlib.pyplot as plt
 - c) Generate a figure of size 10, 6 --> plt.figure(figsize=(10,6))
 - d) Add first plot (plt.plot), and add as parameters: seasons array, lebron ppg array, add label and a marker.
 - e) Add a second plot with Jordans points per game
 - f) Add labels for x axis, y axis and title

```
In [32]: # TASK: Add the required libraries (Numpy and Matplotlib) to create a graph with the points per game of both play
import numpy as np
import matplotlib.pyplot as plt

# TASK: Create a Numpy Array with Lebron's first 15 season's points per game name that array lebron_ppg
lebron_ppg = np.array(lebron_df['Points_Per_Game'][:15])

# TASK: Create a Numpy Array with Jordan's first 15 season's points per game name that array lebron_ppg
# IMPORTANT: Michael Jordan did not play from the 1998 season to the 2000 season, so you need to skip those seasons
# You can identify those seasons in jordan dataframe with the 'NA' value in the 'Team' column
df_jordan_filtered = jordan_df[jordan_df['Team'] != 'NA']

jordan_ppg = np.array(df_jordan_filtered['Points_Per_Game'][:15])

# TASK: Create a Numpy Array with numbers from 1 to 15 to represent the seasons name that array 'seasons'
seasons = np.arange(1, 16)

# TASK: Use Matplotlib to graph the points per game per season of both players in a single graph
# Title of the graph should be 'Points Per Game in First 15 Seasons'
# X Label should be the 'Season'
# Y Label should be 'Points Per Game'

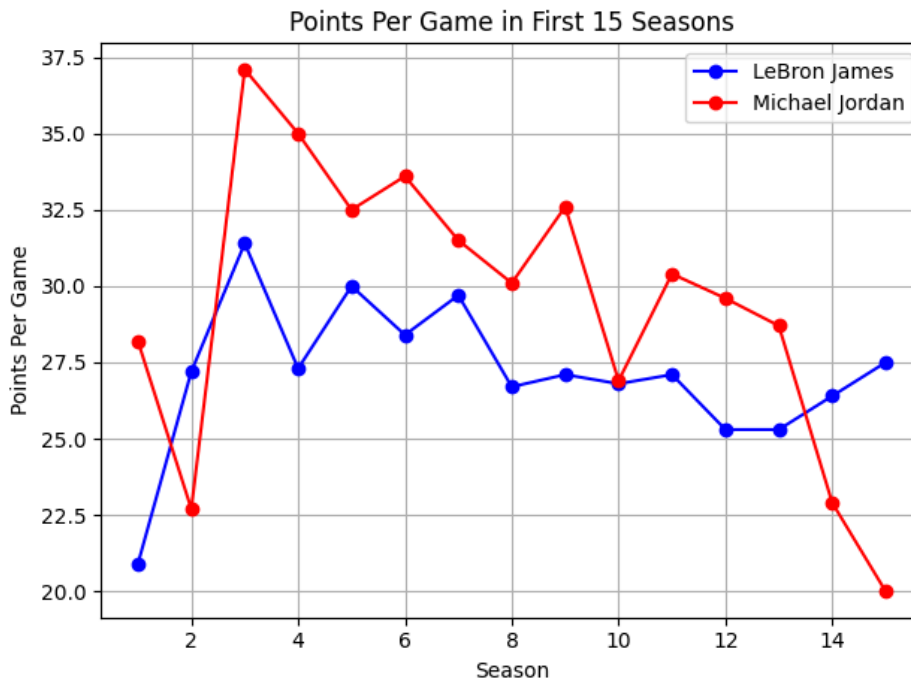
plt.plot(seasons, lebron_ppg, marker='o', label="LeBron James", color='blue')
plt.plot(seasons, jordan_ppg, marker='o', label="Michael Jordan", color='red')

plt.title('Points Per Game in First 15 Seasons')
plt.xlabel('Season')
plt.ylabel('Points Per Game')

plt.legend()

plt.grid(True)

plt.tight_layout()
plt.show()
```



Task 3. If we only look at ppg we might have a clear winner, let's add information about assists:

- i) Bellow you will see how we are adding the assists per game (apg) to the original Dictionary
- ii) Create a pandas dataframe with the updated python dictionary
- iii) Show the pandas dataframe of each player, you should see these columns: 'Season', 'Team', 'Games_Played', 'Points_Per_Game', 'Assists_Per_Game'. (you can use print)

```
In [33]: # Here we are adding a new field to our Python Dictionary:

# lebron_data dictionary will now have a new key called 'Assists_Per_Game' with the following values
lebron_data['Assists_Per_Game'] = [5.9, 7.2, 6.6, 6.0, 7.2, 7.2, 8.6, 7.0, 6.2, 7.3, 6.3, 7.4, 6.8, 8.7, 9.1, 8.3]

# jordan_data dictionary will now have a new key called 'Assists_Per_Game' with the following values
jordan_data['Assists_Per_Game'] = [5.9, 2.9, 4.6, 5.9, 8.0, 6.3, 5.5, 6.1, 5.5, 5.3, 4.3, 4.3, 3.5, 0, 0, 0, 5.2,

# TASK: Create a Pandas DataFrame using the updated lebron_data and jordan_data dictionaries
# Call them lebron_df and jordan_df respectively
lebron_df = pd.DataFrame(lebron_data)
jordan_df = pd.DataFrame(jordan_data)

# TASK: Print both DataFrames to see the updated data
print(lebron_df)
print('-' * 100)
print('')
print(jordan_df)
```

	Season	Team	Games_Played	Points_Per_Game	Assists_Per_Game
0	2003-04	CLE	79	20.9	5.9
1	2004-05	CLE	80	27.2	7.2
2	2005-06	CLE	79	31.4	6.6
3	2006-07	CLE	78	27.3	6.0
4	2007-08	CLE	75	30.0	7.2
5	2008-09	CLE	81	28.4	7.2
6	2009-10	CLE	76	29.7	8.6
7	2010-11	MIA	79	26.7	7.0
8	2011-12	MIA	62	27.1	6.2
9	2012-13	MIA	76	26.8	7.3
10	2013-14	MIA	77	27.1	6.3
11	2014-15	CLE	69	25.3	7.4
12	2015-16	CLE	76	25.3	6.8
13	2016-17	CLE	74	26.4	8.7
14	2017-18	CLE	82	27.5	9.1
15	2018-19	LAL	55	27.4	8.3
16	2019-20	LAL	67	25.3	10.2
17	2020-21	LAL	45	25.0	7.8
18	2021-22	LAL	56	30.3	6.2

	Season	Team	Games_Played	Points_Per_Game	Assists_Per_Game
0	1984-85	CHI	82	28.2	5.9
1	1985-86	CHI	18	22.7	2.9
2	1986-87	CHI	82	37.1	4.6
3	1987-88	CHI	82	35.0	5.9
4	1988-89	CHI	81	32.5	8.0
5	1989-90	CHI	82	33.6	6.3
6	1990-91	CHI	82	31.5	5.5
7	1991-92	CHI	80	30.1	6.1
8	1992-93	CHI	78	32.6	5.5
9	1994-95	CHI	17	26.9	5.3
10	1995-96	CHI	82	30.4	4.3
11	1996-97	CHI	82	29.6	4.3
12	1997-98	CHI	82	28.7	3.5
13	1998-99	NA	0	0.0	0.0
14	1999-00	NA	0	0.0	0.0
15	2000-01	NA	0	0.0	0.0
16	2001-02	WAS	60	22.9	5.2
17	2002-03	WAS	82	20.0	3.8

Task 4. Similar to the points per game, create a graph that now shows the Assists Per Game.

```
In [37]: # TASK: Create a Numpy Array with LeBron's first 15 season's assists per game
lebron_apg = np.array(lebron_df['Assists_Per_Game'][:15])

# TASK: Create a Numpy Array with Jordan's first 15 season's assists per game
df_jordan_filtered = jordan_df[jordan_df['Team'] != 'NA']
jordan_apg = np.array(df_jordan_filtered['Assists_Per_Game'][:15])

# TASK: Use Matplotlib to graph the assists per game per season of both players in a single graph
# Call the x-axis 'Season' and the y-axis 'Assists Per Game'
# Graph title should be 'Assists Per Game in First 15 Seasons'
# Plot label for Michael Jordan should be 'Michael Jordan' and for LeBron James should be 'LeBron James'
# Plot marker for Michael Jordan should be 'o' and for LeBron James should be 'x'
plt.plot(seasons, lebron_apg, marker='x', label="LeBron James", color='blue')
plt.plot(seasons, jordan_apg, marker='o', label="Michael Jordan", color='red')

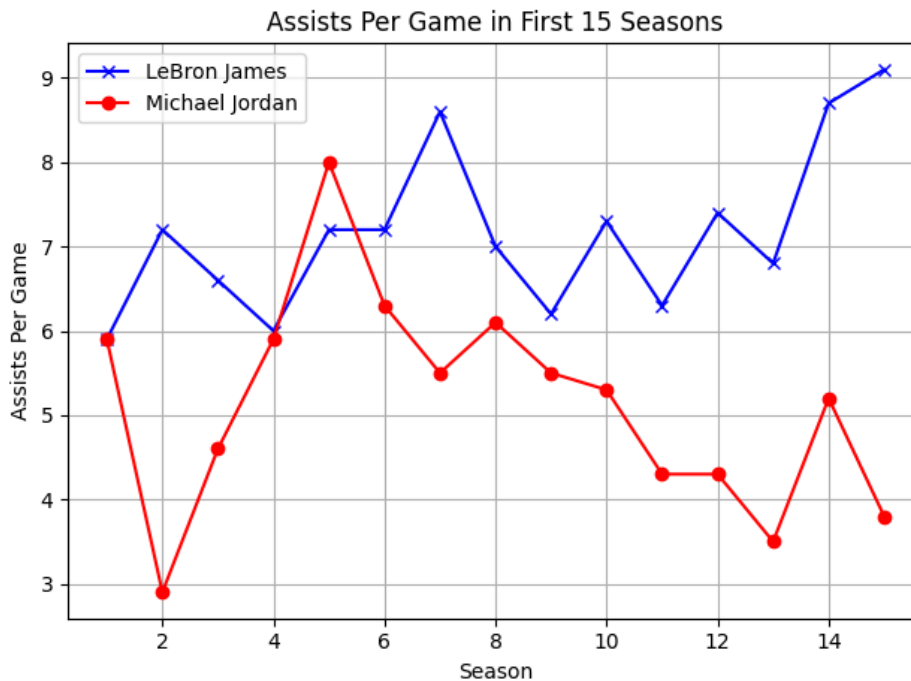
plt.title('Assists Per Game in First 15 Seasons')
plt.xlabel('Season')
plt.ylabel('Assists Per Game')

plt.legend()

plt.grid(True)

plt.tight_layout()
plt.show()

print(jordan_apg)
```



```
[5.9 2.9 4.6 5.9 8. 6.3 5.5 6.1 5.5 5.3 4.3 4.3 3.5 5.2 3.8]
```

Task 5. Now it is getting complicated to know who is the best. Michael Jordan has better ppg (points per game) but LeBron more apg (assists per game).

We could use a know formula that calculated how effective a player is, called the Player Efficiency Rating Formula:

$$PER = ppg + rpg + apg + spg + bpg - mfg - mft - tpg$$

Where:

ppg = point per game

rpg = rebounds per game

apg = assists per game

bpg = blocks per game

mfg = missed field goals per game

mft = missed free throws per game

tpg = turnovers per game

```

In [35]: # Data for all required statistics
# IMPORTANT: These are Python Lists, consider create a Numpy Array for each one for the PER calculation
# Remember you already have the points per game and assists per game for both players as Numpy Arrays
# These next Lists have already filtered the seasons that Michael Jordan did not play

lebron_ppg = np.array(lebron_df['Points_Per_Game'])
lebron_apg = np.array(lebron_df['Assists_Per_Game'])

jordan_ppg = np.array(jordan_df['Points_Per_Game'])
jordan_apg = np.array(jordan_df['Assists_Per_Game'])

# Rebounds Per Game Per Season for both players (Python List)
lebron_rpg = np.array([5.5, 7.4, 7.0, 6.7, 7.9, 7.6, 7.3, 7.5, 7.9, 8.0, 6.9, 6.0, 7.4, 8.6, 8.6, 8.5, 7.8, 7.7,
jordan_rpg = np.array([6.5, 3.6, 5.2, 5.5, 8.0, 6.9, 6.0, 6.4, 6.7, 6.9, 6.6, 5.9, 5.8, 0, 0, 0, 5.7, 6.1])

# Steals Per Game Per Season for both players (Python List)
lebron_spg = np.array([1.6, 2.2, 1.6, 1.6, 1.8, 1.7, 1.6, 1.6, 1.9, 1.7, 1.6, 1.6, 1.4, 1.2, 1.4, 1.3, 1.2, 1.1,
jordan_spg = [2.4, 2.1, 2.9, 3.2, 2.9, 2.8, 2.7, 2.3, 2.8, 1.8, 2.2, 1.7, 1.7, 0, 0, 0, 1.4, 1.5]

# Blocks Per Game Per Season for both players (Python List)
lebron_bpg = np.array([0.7, 0.7, 0.8, 0.7, 1.1, 1.1, 1.0, 0.6, 0.8, 0.9, 0.3, 0.7, 0.6, 0.6, 0.9, 0.6, 0.5, 0.6,
jordan_bpg = np.array([0.8, 1.2, 1.5, 1.6, 0.8, 0.7, 1.0, 0.9, 0.8, 0.8, 0.8, 0.5, 0.5, 0.0, 0.0, 0.0, 0.4, 0.5])

# Turnovers Per Game Per Season for both players (Python List)
lebron_tpg = np.array([3.5, 3.3, 3.3, 3.2, 3.4, 3.0, 3.4, 3.6, 3.4, 3.0, 3.5, 3.9, 3.3, 4.1, 4.2, 3.6, 3.9, 3.7,
jordan_tpg = np.array([3.5, 2.5, 3.3, 3.1, 3.6, 3.0, 2.5, 2.5, 2.7, 2.1, 2.4, 2.3, 2.1, 0, 0, 0, 2.7, 2.4])

# Field Goals Missed Per Game (Attempted - Made / Games) Per Season for both players (Python List)
lebron_fgm = np.array([11.0, 11.2, 12.0, 11.8, 11.9, 10.1, 8.8, 8.1, 9.3, 7.7, 7.6, 9.4, 7.4, 9.8, 8.9, 12.0, 9.8
jordan_fgm = np.array([9.6, 25.4, 14.4, 11.2, 10.2, 12.0, 10.8, 10.9, 12.9, 27.6, 11.3, 11.9, 12.3, 0, 0, 0, 10.8

# Free Throws Missed Per Game (Attempted - Made / Games) Per Season for both players (Python List)
lebron_ftm = np.array([1.4, 1.6, 2.7, 2.2, 3.0, 2.1, 2.4, 2.0, 1.9, 1.7, 1.9, 1.9, 1.7, 1.8, 1.8, 2.3, 1.9, 2.6,
jordan_ftm = np.array([1.4, 2.6, 1.7, 1.9, 1.1, 1.2, 1.6, 1.6, 1.6, 3.5, 1.9, 1.5, 2.1, 0, 0, 0, 0.9, 0.8])

# TASK: Generate a Numpy array with Player Efficiency Rating (PER) for each player
# Formula for PER:
# PER = ppg + rpg + apg + spg + bpg - mfg - mft - tpg
#
# Call the Numpy array for LeBron: lebron_per
# Call the Numpy array for Jordan: jordan_per
lebron_per = lebron_ppg + lebron_rpg + lebron_apg + lebron_spg + lebron_bpg - lebron_fgm - lebron_ftm - lebron_tp
jordan_per = jordan_ppg + jordan_rpg + jordan_apg + jordan_spg + jordan_bpg - jordan_fgm - jordan_ftm - jordan_tp

# TASK: Add PER to the dataframes
lebron_df['PER'] = np.concatenate([lebron_per])
jordan_df['PER'] = np.concatenate([jordan_per])

# TASK: Print the updated DataFrames
print(lebron_df)
print('-' * 100)
print('')
print(jordan_df)

```

	Season	Team	Games_Played	Points_Per_Game	Assists_Per_Game	PER
0	2003-04	CLE	79	20.9	5.9	18.7
1	2004-05	CLE	80	27.2	7.2	28.6
2	2005-06	CLE	79	31.4	6.6	29.4
3	2006-07	CLE	78	27.3	6.0	25.1
4	2007-08	CLE	75	30.0	7.2	29.7
5	2008-09	CLE	81	28.4	7.2	30.8
6	2009-10	CLE	76	29.7	8.6	33.6
7	2010-11	MIA	79	26.7	7.0	29.7
8	2011-12	MIA	62	27.1	6.2	29.3
9	2012-13	MIA	76	26.8	7.3	32.3
10	2013-14	MIA	77	27.1	6.3	29.2
11	2014-15	CLE	69	25.3	7.4	25.8
12	2015-16	CLE	76	25.3	6.8	29.1
13	2016-17	CLE	74	26.4	8.7	29.8
14	2017-18	CLE	82	27.5	9.1	32.6
15	2018-19	LAL	55	27.4	8.3	28.2
16	2019-20	LAL	67	25.3	10.2	29.4
17	2020-21	LAL	45	25.0	7.8	23.7
18	2021-22	LAL	56	30.3	6.2	29.4

	Season	Team	Games_Played	Points_Per_Game	Assists_Per_Game	PER
0	1984-85	CHI	82	28.2	5.9	29.3
1	1985-86	CHI	18	22.7	2.9	2.0
2	1986-87	CHI	82	37.1	4.6	31.9
3	1987-88	CHI	82	35.0	5.9	35.0
4	1988-89	CHI	81	32.5	8.0	37.3
5	1989-90	CHI	82	33.6	6.3	34.1
6	1990-91	CHI	82	31.5	5.5	31.8
7	1991-92	CHI	80	30.1	6.1	30.8
8	1992-93	CHI	78	32.6	5.5	31.2
9	1994-95	CHI	17	26.9	5.3	8.5
10	1995-96	CHI	82	30.4	4.3	28.7
11	1996-97	CHI	82	29.6	4.3	26.3
12	1997-98	CHI	82	28.7	3.5	23.7
13	1998-99	NA	0	0.0	0.0	0.0
14	1999-00	NA	0	0.0	0.0	0.0
15	2000-01	NA	0	0.0	0.0	0.0
16	2001-02	WAS	60	22.9	5.2	21.2
17	2002-03	WAS	82	20.0	3.8	19.5


```

In [36]: # TASK: Get the Numpy array of LeBron and Jordan PER for the first 15 seasons:
# IMPORTANT: Michael Jordan did not play from the 1998 season to the 2000 season, so you need to skip those seasons
positions_to_remove = [13, 14, 15]

jordan_rpg = np.delete(jordan_rpg, positions_to_remove)
jordan_spg = np.delete(jordan_spg, positions_to_remove)
jordan_bpg = np.delete(jordan_bpg, positions_to_remove)
jordan_tpg = np.delete(jordan_tpg, positions_to_remove)
jordan_fgm = np.delete(jordan_fgm, positions_to_remove)
jordan_ftm = np.delete(jordan_ftm, positions_to_remove)

lebron_ppg = np.array(lebron_df['Points_Per_Game'][:15])
lebron_apg = np.array(lebron_df['Assists_Per_Game'][:15])

df_jordan_filtered = jordan_df[jordan_df['Team'] != 'NA']
jordan_ppg = np.array(df_jordan_filtered['Points_Per_Game'][:15])
jordan_apg = np.array(df_jordan_filtered['Assists_Per_Game'][:15])

lebron_rpg = lebron_rpg[:15]
lebron_spg = lebron_spg[:15]
lebron_bpg = lebron_bpg[:15]
lebron_tpg = lebron_tpg[:15]
lebron_fgm = lebron_fgm[:15]
lebron_ftm = lebron_ftm[:15]

jordan_rpg = jordan_rpg[:15]
jordan_spg = jordan_spg[:15]
jordan_bpg = jordan_bpg[:15]
jordan_tpg = jordan_tpg[:15]
jordan_fgm = jordan_fgm[:15]
jordan_ftm = jordan_ftm[:15]

lebron_per = lebron_ppg + lebron_rpg + lebron_apg + lebron_spg + lebron_bpg - lebron_fgm - lebron_ftm - lebron_tpg
jordan_per = jordan_ppg + jordan_rpg + jordan_apg + jordan_spg + jordan_bpg - jordan_fgm - jordan_ftm - jordan_tpg

# Use Matplotlib to graph the Player Efficiency Rating (PER) per season of both players in a single graph
# The graph should have the following:
# - Title: Player Efficiency Rating (PER) in First 15 Seasons
# - X-axis: Season
# - Y-axis: Player Efficiency Rating (PER)
# - Plot the PER of Michael Jordan first 15 seasons and Label it as 'Michael Jordan', marker='o'
# - Plot the PER of LeBron James first 15 seasons and Label it as 'LeBron James', marker='x'
# - Make sure the graph has grid lines
plt.plot(seasons, lebron_per, marker='x', label="LeBron James", color='blue')
plt.plot(seasons, jordan_per, marker='o', label="Michael Jordan", color='red')

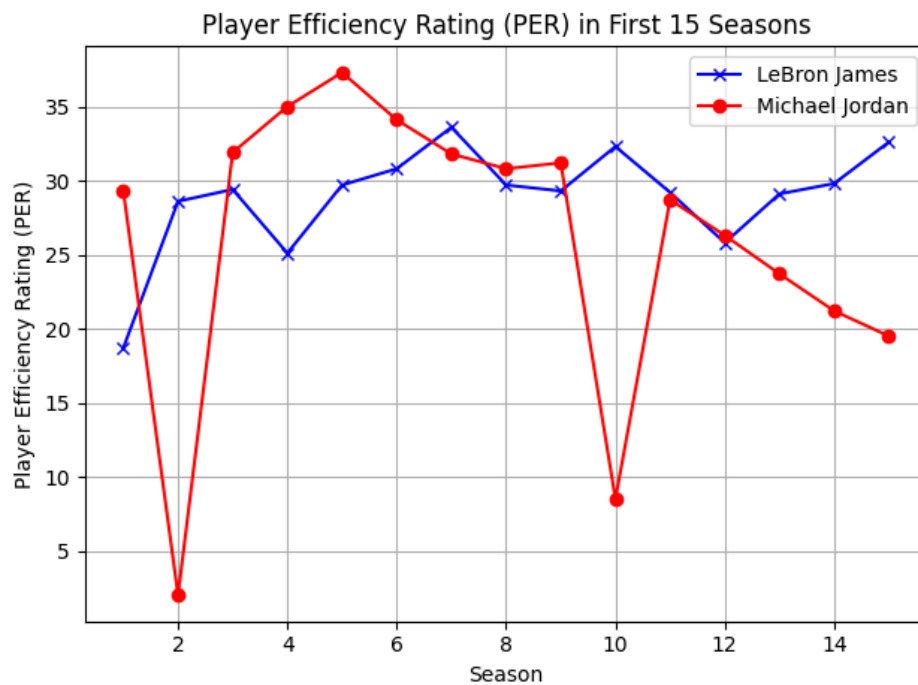
plt.title('Player Efficiency Rating (PER) in First 15 Seasons')
plt.xlabel('Season')
plt.ylabel('Player Efficiency Rating (PER)')

plt.legend()

plt.grid(True)

plt.tight_layout()
plt.show()

```



The best player between both is: ?

Michael Jordan fue más dominante en términos de puntos por partido y tuvo picos más altos en su rendimiento y eficiencia (PER) durante sus mejores años, destacándose como un anotador imparable. Sin embargo, LeBron James muestra una consistencia notable a lo largo de sus primeras 15 temporadas, con una eficiencia más estable y superior en asistencias, lo que lo convierte en un jugador más completo en cuanto a involucrar a sus compañeros.

Personalmente, me gusta más Jordan por su dominio ofensivo en cuanto a puntos y en su mejor momento tenía un PER demasiado alto.