

Out [2]:

	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 [3]: jordan_data = {
    'Season': ['1984-85', '1985-86', '1986-87', '1987-88', '1988-89', '1989-
    'Team': ['CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI', 'CHI',
    'Games_Played': [82, 18, 82, 82, 81, 82, 82, 80, 78, 17, 82, 82, 82, 0,
    'Points_Per_Game': [28.2, 22.7, 37.1, 35.0, 32.5, 33.6, 31.5, 30.1, 32.6
  }

# TASK: Create a Pandas DataFrame using the lebron_data dictionary
# Remember to import the Pandas library
jordan_df = pd.DataFrame(jordan_data)
jordan_df
```

Out [3]:

	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 [4]: # TASK: Create a Numpy Array with Lebron's first 15 season's points per game
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
# IMPORTANT: Michael Jordan did not play from the 1998 season to the 2000 se
# You can identify those seasons in jordan dataframe with the 'NA' value in
jordan_first_15_seasons = jordan_df[jordan_df['Team'] != 'NA'][:15]
jordan_ppg = np.array(jordan_first_15_seasons['Points_Per_Game'])

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

# TASK: Use Matplotlib to graph the points per game per season of both playe
# 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.figure(figsize=(10, 6))

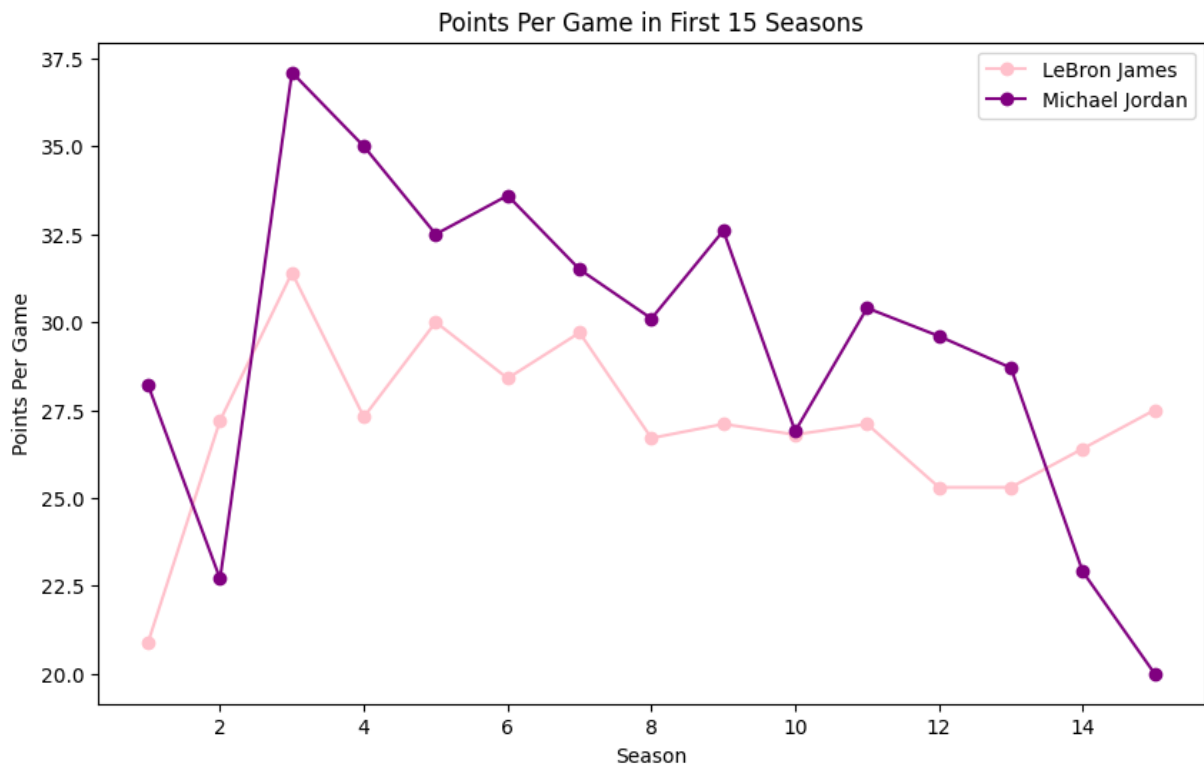
# Plot LeBron's PPG
plt.plot(seasons, lebron_ppg, marker='o', label="LeBron James", color='pink')

# Plot Jordan's PPG
plt.plot(seasons, jordan_ppg, marker='o', label="Michael Jordan", color='purple')

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

# Add a legend
plt.legend()

# Display the graph
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 [5]: # Here we are adding a new field to our Python Dictionary:

# lebron_data dictionary will now have a new key called 'Assists_Per_Game' w
lebron_data['Assists_Per_Game'] = [5.9, 7.2, 6.6, 6.0, 7.2, 7.2, 8.6, 7.0, 6

# jordan_data dictionary will now have a new key called 'Assists_Per_Game' w
jordan_data['Assists_Per_Game'] = [5.9, 2.9, 4.6, 5.9, 8.0, 6.3, 5.5, 6.1, 5

# TASK: Create a Pandas DataFrame using the updated lebron_data and jordan_d
# 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 James DataFrame:")
print(lebron_df)
```

```
print("\nMichael Jordan DataFrame:")
print(jordan_df)
```

LeBron James DataFrame:

	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

Michael Jordan DataFrame:

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

# TASK: Create a Numpy Array with Jordan's first 15 season's assists per gam
jordan_first_15_seasons = jordan_df[jordan_df['Team'] != 'NA'][:15]
jordan_apg = np.array(jordan_first_15_seasons['Assists_Per_Game'])
```

```

# TASK: Use Matplotlib to graph the assists per game per season of both play
# 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 Ja
# Plot marker for Michael Jordan should be 'o' and for LeBron James should b
plt.figure(figsize=(10, 6))

# Plot LeBron's Assists Per Game
plt.plot(seasons, lebron_apg, marker='x', label="LeBron James", color='pink')

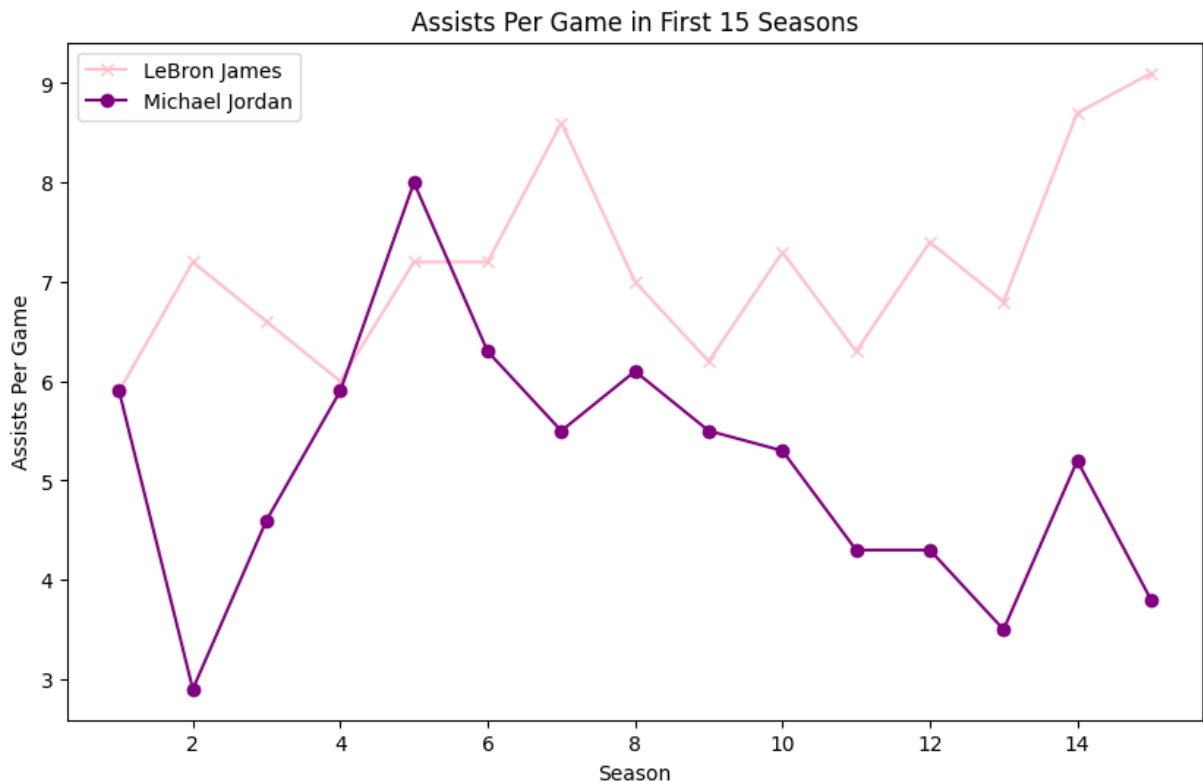
# Plot Jordan's Assists Per Game
plt.plot(seasons, jordan_apg, marker='o', label="Michael Jordan", color='purple')

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

# Add a legend
plt.legend()

# Display the graph
plt.show()

```



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:

$$\text{PER} = \text{ppg} + \text{rpg} + \text{apg} + \text{spg} + \text{bpg} - \text{mfg} - \text{mft} - \text{tpg}$$

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

You will be provided with these values, you will have to:

1. Use these metrics to calculate the PER
2. Create another column in your dataframe called PER (player efficiency rating) with the calculated value based on the formula
3. Generate a graph that has both players PER
4. Generate your conclusions on who is the best player:

```
In [7]: # Data for all required statistics
# IMPORTANT: These are Python Lists, consider create a Numpy Array for each
# Remember you already have the points per game and assists per game for both
# These next lists have already filtered the seasons that Michael Jordan did

def remove_zeros(data_list):
    return [x for x in data_list if x != 0]

# Rebounds Per Game Per Season for both players (Python List)
lebron_rpg = remove_zeros([5.5, 7.4, 7.0, 6.7, 7.9, 7.6, 7.3, 7.5, 7.9, 8.0,
jordan_rpg = remove_zeros([6.5, 3.6, 5.2, 5.5, 8.0, 6.9, 6.0, 6.4, 6.7, 6.9,

# Steals Per Game Per Season for both players (Python List)
lebron_spg = remove_zeros([1.6, 2.2, 1.6, 1.6, 1.8, 1.7, 1.6, 1.6, 1.9, 1.7,
jordan_spg = remove_zeros([2.4, 2.1, 2.9, 3.2, 2.9, 2.8, 2.7, 2.3, 2.8, 1.8,

# Blocks Per Game Per Season for both players (Python List)
lebron_bpg = remove_zeros([0.7, 0.7, 0.8, 0.7, 1.1, 1.1, 1.0, 0.6, 0.8, 0.9,
jordan_bpg = remove_zeros([0.8, 1.2, 1.5, 1.6, 0.8, 0.7, 1.0, 0.9, 0.8, 0.8,

# Turnovers Per Game Per Season for both players (Python List)
lebron_tpg = remove_zeros([3.5, 3.3, 3.3, 3.2, 3.4, 3.0, 3.4, 3.6, 3.4, 3.0,
jordan_tpg = remove_zeros([3.5, 2.5, 3.3, 3.1, 3.6, 3.0, 2.5, 2.5, 2.7, 2.1,

# Field Goals Missed Per Game (Attempted - Made / Games) Per Season for both
lebron_fgm = remove_zeros([11.0, 11.2, 12.0, 11.8, 11.9, 10.1, 8.8, 8.1, 9.3
jordan_fgm = remove_zeros([9.6, 25.4, 14.4, 11.2, 10.2, 12.0, 10.8, 10.9, 12

# Free Throws Missed Per Game (Attempted - Made / Games) Per Season for both
lebron_ftm = remove_zeros([1.4, 1.6, 2.7, 2.2, 3.0, 2.1, 2.4, 2.0, 1.9, 1.7,
jordan_ftm = remove_zeros([1.4, 2.6, 1.7, 1.9, 1.1, 1.2, 1.6, 1.6, 1.6, 3.5,

# TASK: Generate a Numpy array with Player Efficiency Rating (PER) for each
py arrays
```



```

lebron_rpg = np.array(lebron_rpg)
jordan_rpg = np.array(jordan_rpg)
lebron_spg = np.array(lebron_spg)
jordan_spg = np.array(jordan_spg)
lebron_bpg = np.array(lebron_bpg)
jordan_bpg = np.array(jordan_bpg)
lebron_tpg = np.array(lebron_tpg)
jordan_tpg = np.array(jordan_tpg)
lebron_fgm = np.array(lebron_fgm)
jordan_fgm = np.array(jordan_fgm)
lebron_ftm = np.array(lebron_ftm)
jordan_ftm = np.array(jordan_ftm)

# Assuming lebron_ppg and lebron_apg are already defined as Numpy arrays
# If not, you should define them here

# Trim all LeBron's arrays to match Jordan's 15 seasons
lebron_ppg = lebron_ppg[:15]
lebron_apg = lebron_apg[: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]

# Trim Jordan's arrays to 15 seasons (assuming they're not already)
jordan_ppg = jordan_ppg[:15]
jordan_apg = jordan_apg[: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]

# 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
jordan_per = (jordan_ppg + jordan_rpg + jordan_apg + jordan_spg + jordan_bpg

# TASK: Add PER to the dataframes
lebron_df = pd.DataFrame({
    'PPG': lebron_ppg,
    'APG': lebron_apg,
    'RPG': lebron_rpg,
    'SPG': lebron_spg,
    'BPG': lebron_bpg,
    'TPG': lebron_tpg,
    'FGM': lebron_fgm,
    'FTM': lebron_ftm,
    'PER': lebron_per
})

```

```

jordan_df = pd.DataFrame({
    'PPG': jordan_ppg,
    'APG': jordan_apg,
    'RPG': jordan_rpg,
    'SPG': jordan_spg,
    'BPG': jordan_bpg,
    'TPG': jordan_tpg,
    'FGM': jordan_fgm,
    'FTM': jordan_ftm,
    'PER': jordan_per
})

# TASK: Print the updated DataFrames
print("LeBron James DataFrame with PER:")
print(lebron_df)
print("\nMichael Jordan DataFrame with PER:")
print(jordan_df)

```

LeBron James DataFrame with PER:

	PPG	APG	RPG	SPG	BPG	TPG	FGM	FTM	PER
0	20.9	5.9	5.5	1.6	0.7	3.5	11.0	1.4	18.7
1	27.2	7.2	7.4	2.2	0.7	3.3	11.2	1.6	28.6
2	31.4	6.6	7.0	1.6	0.8	3.3	12.0	2.7	29.4
3	27.3	6.0	6.7	1.6	0.7	3.2	11.8	2.2	25.1
4	30.0	7.2	7.9	1.8	1.1	3.4	11.9	3.0	29.7
5	28.4	7.2	7.6	1.7	1.1	3.0	10.1	2.1	30.8
6	29.7	8.6	7.3	1.6	1.0	3.4	8.8	2.4	33.6
7	26.7	7.0	7.5	1.6	0.6	3.6	8.1	2.0	29.7
8	27.1	6.2	7.9	1.9	0.8	3.4	9.3	1.9	29.3
9	26.8	7.3	8.0	1.7	0.9	3.0	7.7	1.7	32.3
10	27.1	6.3	6.9	1.6	0.3	3.5	7.6	1.9	29.2
11	25.3	7.4	6.0	1.6	0.7	3.9	9.4	1.9	25.8
12	25.3	6.8	7.4	1.4	0.6	3.3	7.4	1.7	29.1
13	26.4	8.7	8.6	1.2	0.6	4.1	9.8	1.8	29.8
14	27.5	9.1	8.6	1.4	0.9	4.2	8.9	1.8	32.6

Michael Jordan DataFrame with PER:

	PPG	APG	RPG	SPG	BPG	TPG	FGM	FTM	PER
0	28.2	5.9	6.5	2.4	0.8	3.5	9.6	1.4	29.3
1	22.7	2.9	3.6	2.1	1.2	2.5	25.4	2.6	2.0
2	37.1	4.6	5.2	2.9	1.5	3.3	14.4	1.7	31.9
3	35.0	5.9	5.5	3.2	1.6	3.1	11.2	1.9	35.0
4	32.5	8.0	8.0	2.9	0.8	3.6	10.2	1.1	37.3
5	33.6	6.3	6.9	2.8	0.7	3.0	12.0	1.2	34.1
6	31.5	5.5	6.0	2.7	1.0	2.5	10.8	1.6	31.8
7	30.1	6.1	6.4	2.3	0.9	2.5	10.9	1.6	30.8
8	32.6	5.5	6.7	2.8	0.8	2.7	12.9	1.6	31.2
9	26.9	5.3	6.9	1.8	0.8	2.1	27.6	3.5	8.5
10	30.4	4.3	6.6	2.2	0.8	2.4	11.3	1.9	28.7
11	29.6	4.3	5.9	1.7	0.5	2.3	11.9	1.5	26.3
12	28.7	3.5	5.8	1.7	0.5	2.1	12.3	2.1	23.7
13	22.9	5.2	5.7	1.4	0.4	2.7	10.8	0.9	21.2
14	20.0	3.8	6.1	1.5	0.5	2.4	9.2	0.8	19.5

```

In [8]: # TASK: Get the Numpy array of LeBron and Jordan PER for the first 15 season
# IMPORTANT: Michael Jordan did not play from the 1998 season to the 2000 se
jordan_per_first_15 = jordan_df['PER'][:15]
lebron_per_first_15 = lebron_df['PER'][:15]

# Use Matplotlib to graph the Player Efficiency Rating (PER) per season of b
# 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
# - Plot the PER of LeBron James first 15 seasons and label it as 'LeBron Ja
# - Make sure the graph has grid lines

seasons = np.arange(1, 16) # Season array from 1 to 15
plt.figure(figsize=(12, 8)) # Plot

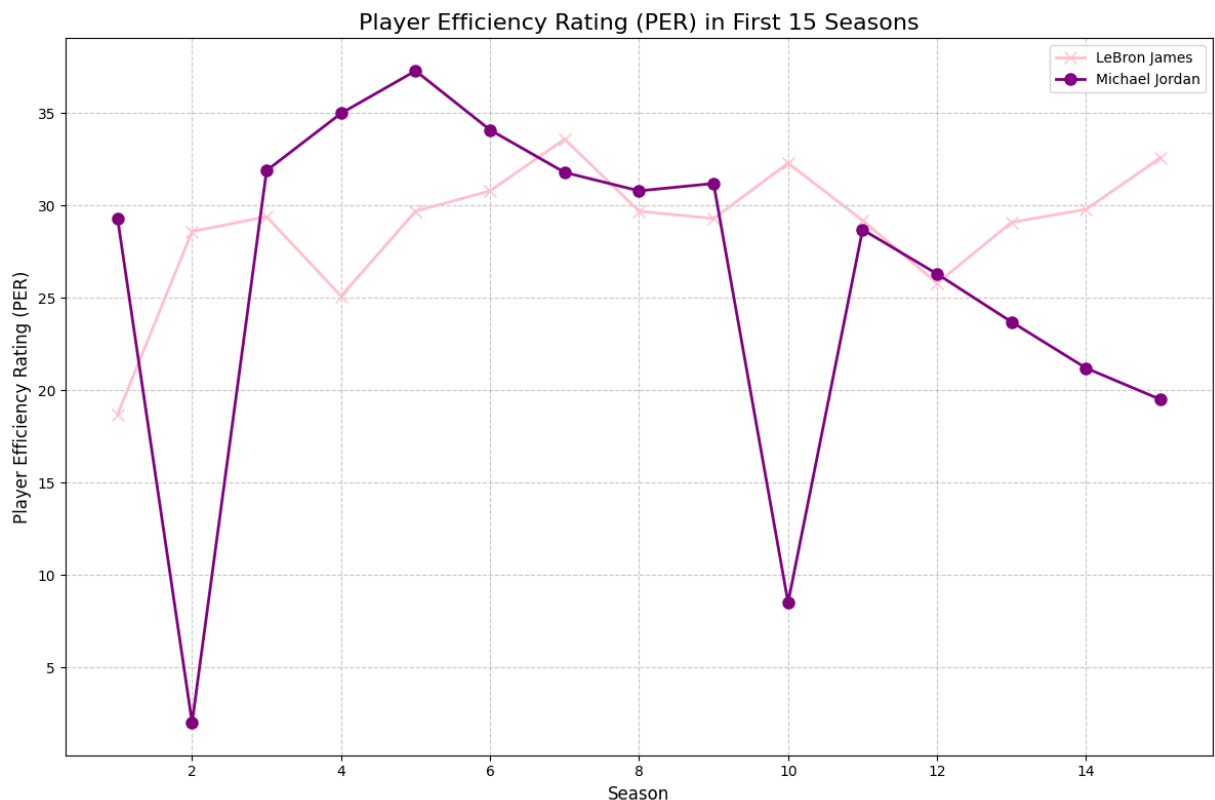
plt.plot(seasons, lebron_per_first_15, marker='x', label='LeBron James', col
plt.plot(seasons, jordan_per_first_15, marker='o', label='Michael Jordan', c

# Title and labels
plt.title('Player Efficiency Rating (PER) in First 15 Seasons', fontsize=16)
plt.xlabel('Season', fontsize=12)
plt.ylabel('Player Efficiency Rating (PER)', fontsize=12)

plt.grid(True, linestyle='--', alpha=0.7) # Grid lines
plt.legend(fontsize=10) # Legend

plt.tight_layout()
plt.show()

```



The best player between both is: LeBron James

This player was chosen because of his consistent performance. Michel Jordan, despite surpassing him in certain numbers, his performance went downhill as the seasons went on.