⚙️ Week 4 Student Guide – Box Score Analysis and Data Export

This guide walks you through the concepts and steps in the **Week 4 Assignment**, which focuses on using the **pandas** library for calculating team averages, identifying statistical leaders, creating a new efficiency metric, and exporting your final results.

💻 Part 1: Pandas Box Score Operations Review

This week builds on your DataFrame knowledge to perform common analytical tasks on a simple basketball box score.

1. Creating and Inspecting DataFrames

You start by manually creating a DataFrame from a Python dictionary, which is a fundamental skill for inputting small, structured datasets into pandas.

Concept	**Description**	**Python Code Example**
DataFrame Creation	Building a DataFrame from a dictionary where keys are column names and values are lists of data.	df = pd.DataFrame(data)
Inspection	Viewing the entire DataFrame.	print(df)

2. Aggregation: Team Averages

The .mean() function is a quick way to calculate the average of all numerical columns in a DataFrame, giving you the **team average** for a specific box score.

Function	**Purpose**	**Python Code Example**
Calculate Mean	Finds the arithmetic mean (average) of each column.	df.mean(numeric_only=True)
numeric_only=True	*Important:* Ensures the operation only runs on numerical columns (PTS, REB, AST, etc.) and ignores text columns (Player).	N/A

3. Finding Leaders: Max Value Location

To identify the player who led the team in a given statistical category (like PTS or REB), you need to find the **index** of the maximum value in that column and use that index to look up the player's name.

Step	Purpose	Python Code Example
Find Max Index	Finds the index (row label) of the maximum value in a column.	df["PTS"].idxmax()
Locate Player Name	Uses the index found above to return the value from the **"Player"** column in that row.	df.loc[index, "Player"]

📝 Part 2: Week 4 Assignment Tasks

Your assignment is a five-step sequence that moves from creating the data to exporting the results, as demonstrated in the assignment notebook.

1️⃣ Create Your Data (Code Cell 1)

This step is provided for you. It creates the initial DataFrame `df` with basic box score stats (PTS, REB, AST, STL, BLK) for five players.

2️⃣ Team Averages (Code Cell 2)

Calculate and print the average for all available numerical statistics to get the overall team performance snapshot.

Python

```
# Team Average Stats: Use .mean() with numeric_only=True
print("Team Average Stats:\n", df.mean(numeric_only=True))
```

3️⃣ Identify Leaders (Code Cell 3)

Use the combination of `.idxmax()` and `.loc` to dynamically identify the leaders for Points, Rebounds, and Assists.

Python

```
# Identify leaders for the three main categories
print("Top Scorer:", df.loc[df["PTS"].idxmax(), "Player"])
print("Best Rebounder:", df.loc[df["REB"].idxmax(), "Player"])
print("Top Assister:", df.loc[df["AST"].idxmax(), "Player"])
```

4️⃣ Add Efficiency (Code Cell 4)

Create a new column called **"EFF"** using a standard, basic formula often used in fantasy basketball and basic box score analysis: $\text{EFF} = \text{PTS} + \text{REB} + \text{AST} + \text{STL} + \text{BLK}$. Then, sort the DataFrame by this new metric.

1. **Calculate EFF:**

   Python

```
df["EFF"] = df["PTS"] + df["REB"] + df["AST"] + df["STL"] + df["BLK"]
```

2. **Sort the DataFrame:** Use the .sort_values() function on the new "EFF" column in descending order (ascending=False).

   Python

   ```
   df_sorted = df.sort_values(by="EFF", ascending=False)
   print(df_sorted)
   ```

5️⃣ Export Results (Code Cell 5)

Save the sorted DataFrame to a CSV file. This is a critical step in a data science workflow for sharing or future analysis.

- **Export to CSV:** Use the .to_csv() function. Remember to set index=False to prevent writing the DataFrame's index as an unnecessary column in your output file.

Python

```
df_sorted.to_csv("team_performance.csv", index=False)
print("Saved team_performance.csv")
```