📊 Week 3 Student Guide – Basketball Data Manipulation with Pandas

This guide is designed to support your work on the **Week 3 Assignment: Basketball Data Manipulation**. You will be using the **pandas** library to load, inspect, filter, group, and visualize a synthetic dataset of Chicago Bulls player performance metrics.

💻 Part 1: Pandas Fundamentals Review

The core of Week 3 is working with **DataFrames**—the primary data structure in the pandas library.

1. The DataFrame

A DataFrame is a two-dimensional labeled data structure with columns of potentially different types, similar to a spreadsheet or SQL table.

DataFrame Action	Python Code Example	Purpose
Inspection	`df.head()`	Displays the first 5 rows.
Information	`df.info()`	Shows the index, column names, data types, and non-null values.
Statistics	`df.describe()`	Provides summary statistics (mean, min, max, std, quartiles) for numerical columns.

2. Filtering Data (Conditional Selection)

Filtering allows you to select a subset of rows based on specific conditions. This is essential for isolating data like "Game" sessions or a specific player.

Goal	Python Code Example	Explanation
Filter by Column Value	`games_df = df[df["Session_Type"] == "Game"]`	Creates a new DataFrame containing only rows where the `Session_Type` is 'Game'.
Select a Column	`df["Workload_AU"]`	Selects a specific Series (column) from the DataFrame.

3. Grouping and Aggregation

The `groupby()` function splits the data into groups based on some criterion (e.g., player or session type) and then applies an aggregation function (like `mean()`, `sum()`, or `max()`) to each group.

Goal	Python Code Example	Explanation
Group by a Column	`df.groupby("Session_Type")`	Groups all rows that have the same value in the `Session_Type` column.
Calculate Mean of Groups	`.mean()`	Calculates the average of all numerical columns within each group.
Select Columns for Aggregation	`[["Workload_AU", "RPE"]]`	Selects only these two columns for aggregation after grouping.

🛠️ Part 2: Week 3 Assignment Tasks

The assignment uses a synthetic dataset of player performance data containing columns like `Player`, `Position`, `Session_Type`, `Workload_AU`, and `RPE`.

🧩 Task 1: Inspect the Dataset

Use the inspection methods above to understand your DataFrame.

- **Display dataset info and summary statistics**: Use `df.info()` and `df.describe()`.
- **Check for missing values**: You can chain `.isnull()` and `.sum()` on your DataFrame: `df.isnull().sum()`

🧩 Task 2: Data Filtering and Workload per Player

This task requires two steps: **filtering** for a specific condition, then **grouping** the resulting data.

1. **Filter only Game sessions**:

```
Python
games_df = df[df["Session_Type"] == "Game"]
```

2. **Show the average workload per player**:

```
Python
avg_workload_games = games_df.groupby("Player")["Workload_AU"].mean()
print(avg_workload_games)
```

🧩 Task 3: Summary Statistics by Session Type

This task requires you to group by `Session_Type` and calculate the mean for two specified columns.

- **Group the dataset by `Session_Type` and find the mean `Workload_AU` and `RPE`:**

```
Python
avg_by_session = df.groupby("Session_Type")[["Workload_AU",
"RPE"]].mean().round(1)
print(avg_by_session)
```

🧩 Task 4: Visualization

Create a visual representation of player workloads. You can use the `avg_workload_games` from Task 2.

- **Create a bar or line chart comparing workloads by player**: Use the built-in plotting function on your grouped Series/DataFrame, for example: `avg_workload_games.plot(kind='bar')`.
- **Add appropriate labels and titles**: Use `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` after importing `matplotlib.pyplot as plt`.

🧩 Task 5: Interpretation

Review the outputs from Task 3, which shows the mean **Workload_AU** and **RPE** by session type.

- **Write 2–3 sentences summarizing what your data analysis reveals about workload and RPE patterns among players.**

Consider: Which `Session_Type` has the highest average `Workload_AU`? How does this compare to the average `RPE` for that session type? (e.g., Do Game sessions have higher workload than Practice sessions?)