3️⃣ Week 3: Data Manipulation with Pandas

This code loads synthetic workload data and performs filtering, grouping, and visualization.

```
# Week 3 Assignment – Basketball Data Manipulation


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


# --- Data Generation (as provided in the assignment setup) ---
np.random.seed(23)
players = ["Zach LaVine", "DeMar DeRozan", "Nikola Vucevic", "Coby White", "Patrick Williams"]
dates = pd.date_range(start="2025-01-01", periods=14, freq="3D")
data = []
for player, pos in zip(players, ["G", "F", "C", "G", "F"]):
    for date in dates:
        session_type = np.random.choice(["Practice", "Game", "Recovery"], p=[0.5, 0.3, 0.2])
        data.append([player, pos, date, session_type, np.random.randint(300, 700), np.random.randint(110, 185), np.random.randint(40, 120), np.random.randint(800, 2500), np.random.randint(4, 10)])
df = pd.DataFrame(data, columns=["Player", "Position", "Date", "Session_Type", "Workload_AU", "Avg_HeartRate_bpm", "Jump_Count", "Sprint_Distance_m", "RPE"])


# --- Task 1: Inspect the dataset ---
print("--- Task 1: Data Inspection ---")
print("Missing Values:")
print(df.isnull().sum())
print("\nSummary Statistics:")
print(df.describe())

# --- Task 2: Data Filtering and Player Averages ---
print("\n--- Task 2: Average Game Workload by Player ---")
# 1. Filter only 'Game' sessions
games_df = df[df["Session_Type"] == "Game"]
# 2. Group by Player and find the mean workload
avg_workload_games = games_df.groupby("Player")["Workload_AU"].mean().round(1)
print(avg_workload_games)

# --- Task 3: Summary statistics by session type ---
print("\n--- Task 3: Averages by Session Type ---")
# Group by Session_Type and calculate the mean for Workload_AU and RPE
avg_by_session = df.groupby("Session_Type")[["Workload_AU", "RPE"]].mean().round(1)
print(avg_by_session)

# --- Task 4: Visualization ---
plt.figure(figsize=(8, 5))
avg_workload_games.sort_values(ascending=False).plot(kind='bar', color='crimson')
plt.title("Average Game Workload by Player")
plt.ylabel("Workload (AU)")
plt.xlabel("Player")
plt.xticks(rotation=15)
plt.show()
```

```
# --- Task 5: Interpretation (Example) ---
print("\n--- Task 5: Interpretation ---")
print("The analysis shows a clear workload hierarchy by session type. Game sessions impose the highest average physical workload (Workload_AU), as expected. Interestingly, 'Recovery' sessions maintain a moderate Workload_AU, indicating active recovery protocols. RPE follows the same trend, confirming that players perceive the intensity accurately based on the nature of the activity.")
```