6️⃣ Week 6: NumPy & Statistical Analysis

This code uses NumPy for descriptive statistics, correlation, and efficient calculation of a custom efficiency metric.

```
# Week 6 Assignment – Statistical Basketball Analysis


import numpy as np

import pandas as pd

import matplotlib.pyplot as plt


# --- 1. Create Your Dataset ---

players = np.array(["Player A", "Player B", "Player C", "Player D", "Player E"])

points = np.array([25, 19, 22, 15, 10])

rebounds = np.array([9, 6, 11, 8, 5])

assists = np.array([7, 5, 6, 4, 3])


# --- 2. Calculate Key Metrics ---

avg_points = np.mean(points)

std_points = np.std(points)

median_rebounds = np.median(rebounds)

corr_pr = np.corrcoef(points, rebounds)[0, 1]


print("--- Key Metrics ---")

print(f"Average Points: {avg_points}")

print(f"Std Dev Points: {std_points}")

print(f"Median Rebounds: {median_rebounds}")

print(f"Correlation (Points vs Rebounds): {round(corr_pr, 2)}") # Expected approx. 0.73
```

```
# --- 3. Compute Player Efficiency ---
# Efficiency Formula (example): (Points * 0.5) + (Rebounds * 1.2) + (Assists * 0.8)
efficiency = points * 0.5 + rebounds * 1.2 + assists * 0.8

# Create a DataFrame to display all results
df = pd.DataFrame({
   "Player": players,
   "Points": points,
   "Rebounds": rebounds,
   "Assists": assists,
   "Efficiency": efficiency.round(1) # Round for clean display
})
print("\n--- Player Efficiency Table ---")
print(df)

# --- 4. Visualize Your Findings ---
plt.figure(figsize=(8,5))
plt.bar(df["Player"], df["Efficiency"], color="purple")
plt.title("Player Efficiency Comparison")
plt.ylabel("Efficiency Score")
plt.show()

# --- 5. Reflection (based on output) ---
# Player C (29.0) had the best efficiency.
```

The correlation of 0.73 indicates a strong positive relationship between points and rebounds.