



# Python for Basketball Data Science: Week 1 Student Guide

This guide is designed for students in your 'Python for Basketball Data Science' course. It provides a structured walkthrough of the concepts and assignments from **Week 1**, focusing on Python basics and the calculation of key basketball metrics: **Pace**, **(Possessions)** and **Offensive Efficiency**.

---

## Part 1: Reviewing Python & Basketball Context

This section reviews the core concepts and code from the Week1\_Getting\_Started\_with\_Python\_and\_Basketball.ipynb notebook.

### 1. Python Basics: Variables and Data Types

In data science, we use variables to store data and perform calculations. Python has several **data types** that are essential for handling basketball statistics:

Data Type	Example	Use in Basketball Data
Integer (int)	shots = 85	Whole numbers like <b>Field Goals Made (FGM)</b> , <b>Turnovers (TO)</b> , or <b>Points (PTS)</b> .
Float (float)	free_throw_rate = 0.75	Numbers with decimal points, used for <b>percentages</b> or <b>rates</b> like 0.44 in the possessions formula.
String (str)	player = "Zach LaVine"	Text data, used for <b>team names</b> or <b>player names</b> .

**Code Example (Section C: Data Types):**

```
Python
# Integer
shots = 85

# Float
free_throw_rate = 0.75

# String
player = "Zach LaVine"
```

### 2. Basketball Application: Pace and Efficiency

The two most fundamental efficiency metrics in modern basketball analytics are **Possessions** and **Offensive Efficiency**.

Metric	Purpose	Formula (in Python)
<b>Possessions</b>	Estimates the number of opportunities a team had to score.	Possessions = FGA + (0.44 * FTA) - OREB + TO
<b>Offensive Efficiency</b>	Measures how many points a team scores per 100 possessions.	(PTS / Possessions) * 100

### Code Example (Section D: Basketball Application):

```

Python
# Input Stats
FGA = 85 # Field Goal Attempts
FTA = 20 # Free Throw Attempts
OREB = 10 # Offensive Rebounds
TO = 15 # Turnovers
points = 112 # Points Scored

# Calculation
possessions = FGA + (0.44 * FTA) - OREB + TO
print("Estimated possessions:", possessions)
# Output: Estimated possessions: 93.8

off_efficiency = (points / possessions) * 100
print("Offensive Efficiency:", off_efficiency)
# Output: Offensive Efficiency: 119.403... (or ~119.4)

```

---

## Part 2: Week 1 Assignment Walkthrough

The **Week 1 Assignment** challenges you to apply the formulas from the review using real-world data.

### 1. Assignment Objective

Your goal is to use a real NBA box score, collect five statistics, and then write a Python script to calculate the team's possessions and Offensive Efficiency.

### 2. Required Stats

You must collect these stats for at least **one team** from a recent NBA game:

- **FGA:** Field Goal Attempts
- **FTA:** Free Throw Attempts
- **OREB:** Offensive Rebounds
- **TO:** Turnovers
- **PTS:** Points Scored

### 3. Solution Guide (Example)

Let's use a sample box score for the **Boston Celtics** from a hypothetical game as an example to guide your work.

### Sample Data for Boston Celtics:

- **FGA** = 90
- **FTA** = 25
- **OREB** = 12
- **TO** = 14
- **PTS** = 115

### Step 1: Enter Game Stats (in your notebook's first code cell)

You need to update the initial variable assignments with the data you collected.

#### Python

```
# Step 1: Enter game stats here  
FGA = 90 # Field Goal Attempts  
FTA = 25 # Free Throw Attempts  
OREB = 12 # Offensive Rebounds  
TO = 14 # Turnovers  
PTS = 115 # Points Scored
```

### Step 2: Calculate and Print Results (in your notebook's second code cell)

Use the defined formulas, then use the `print()` function to display a clear, complete sentence as required.

#### Python

```
# Step 2: Calculate possessions and offensive efficiency  
possessions = FGA + (0.44 * FTA) - OREB + TO  
  
# Calculate offensive efficiency, ensuring you handle division by zero (safety check)  
off_efficiency = (PTS / possessions) * 100 if possessions > 0 else 0  
  
# Print the results in a clear sentence (as required by the instructions)  
print("The team scored", PTS, "points on", round(possessions, 2), "possessions.")  
# The round() function is used here to make the possessions value cleaner.  
  
print("Offensive Efficiency:", round(off_efficiency, 2))
```

### Expected Output for Celtics Example:

Metric	Calculation	Result
Possessions	$90 + (0.44 \times 25) - 12 + 14$	93.0
Offensive Efficiency	$(115 / 93.0) \times 100$	123.66

### Final Print Output:

The team scored 115 points on 93.0 possessions.  
Offensive Efficiency: 123.66

---

## Part 3: Extra Challenge: Comparing Two Teams

For the **Extra Challenge**, you must repeat the process for the opposing team and compare the efficiencies.

### 1. Opposing Team Data

Let's use sample data for the **Los Angeles Lakers** as the opposing team:

**Sample Data for Los Angeles Lakers:**

- **FGA** = 88
- **FTA** = 30
- **OREB** = 8
- **TO** = 17
- **PTS** = 108

### 2. Solution Hint

The best way to handle the two teams is to use **distinct variable names** for each team's statistics.

Python

```
# Celtics Stats (Team 1)
FGA_1 = 90
FTA_1 = 25
OREB_1 = 12
TO_1 = 14
PTS_1 = 115

# Lakers Stats (Team 2)
FGA_2 = 88
FTA_2 = 30
OREB_2 = 8
TO_2 = 17
PTS_2 = 108

# Calculate Possessions
possessions_1 = FGA_1 + (0.44 * FTA_1) - OREB_1 + TO_1
possessions_2 = FGA_2 + (0.44 * FTA_2) - OREB_2 + TO_2

# Calculate Offensive Efficiency
off_efficiency_1 = (PTS_1 / possessions_1) * 100
off_efficiency_2 = (PTS_2 / possessions_2) * 100

# Print and Compare
print("Celtics Efficiency:", round(off_efficiency_1, 2)) # 123.66
print("Lakers Efficiency:", round(off_efficiency_2, 2)) # 118.04
```

```
if off_efficiency_1 > off_efficiency_2:  
    print("The Celtics were more efficient.")  
else:  
    print("The Lakers were more efficient.")
```

By completing this assignment, you've successfully used Python variables, arithmetic operations, and conditional logic to perform a foundational task in basketball data science!