🧹 Week 7 Student Guide – Cleaning and Preprocessing Basketball Data

This guide covers the crucial first step in any data science project: **Data Cleaning and Preprocessing**. You will learn how to use the **pandas** library to ensure your basketball data is accurate, complete, and formatted correctly before analysis.

💻 Part 1: Data Cleaning Fundamentals

Raw datasets often contain errors, missing entries, and inconsistent formatting. Data cleaning fixes these issues using specific pandas methods.

1. Identifying and Handling Missing Data (NaN / None)

Missing values are represented in pandas as `NaN` (Not a Number) or `None`. Analyzing data with these values will lead to errors or inaccurate results.

Goal	Pandas Method	Description & Example
Locate Missing Values	`.isna().sum()`	Returns the count of missing values per column.
Remove Rows with Missing Data	`.dropna()`	*Generally avoided* for small datasets as it removes entire rows, leading to data loss.
Fill with Mean (Imputation)	`.fillna(df['COL'].mean())`	Fills missing numeric values with the **average** of the existing column data. *Best for roughly symmetrical data*.
Fill with Median (Imputation)	`.fillna(df['COL'].median())`	Fills missing numeric values with the **middle** value. *Best for skewed data or when outliers might distort the mean*.

2. Handling Duplicates and Inconsistent Types

Data Issue	Pandas Method	Example
Duplicate Entries	`.drop_duplicates()`	Removes rows that are identical across all columns (e.g., the duplicate LeBron James row was removed).
Incorrect Data Type	`.astype(dtype)`	Converts a column's data type, e.g., changing a floating-point number (like `32.75`) into an integer (`32`).
Inconsistent Labels	`.rename(columns={...})`	Changes column names to be standardized and readable (e.g., `PTS` to `Points`).

Data Issue	Pandas Method	Example
Column Order	df = df[['Col1', 'Col2', ...]]	Specifies the final, desired order of columns.

🏀 Part 2: Week 7 Assignment Tasks

The assignment uses a raw dataset with **missing values** in the PTS, REB, AST, and MIN columns.

1. Identify Missing Values (Step 2)

Before filling, you must know what you are dealing with:

Python
```
# Check how many missing values are in each column
print(df.isnull().sum())
# Expected: PTS: 1, REB: 1, AST: 1, MIN: 1
```
2. Fill or Drop Missing Data (Step 3)

In the assignment, you have the choice of method. The example code uses a mix of techniques appropriate for different column types:

- **Points (PTS):** Filled with the **Mean** (Average) of the remaining points.
- **Assists (AST):** Filled with the **Median** of the remaining assists.
- **Rebounds (REB):** Filled with the **Mean** (Average) of the remaining rebounds.
- **Minutes (MIN):** Filled with the **Mean** (Average) of the remaining minutes.

Python
```
# Impute using Mean for PTS and REB, and Median for AST.
# Impute MIN using Mean.
df["PTS"].fillna(df["PTS"].mean(), inplace=True)
df["AST"].fillna(df["AST"].median(), inplace=True)
df["REB"].fillna(df["REB"].mean(), inplace=True)
df["MIN"].fillna(df["MIN"].mean(), inplace=True)
```
3. Fix Data Types (Step 4)

Since minutes can only be whole numbers in a box score, the MIN column should be converted from a floating-point number to an integer to remove the decimal (e.g., $32.75 \to 32$).

Python
```
# Convert the MIN column to integers
df["MIN"] = df["MIN"].astype(int)
```
4. Rename Columns (Step 5)

Rename the shorthand column names to full, readable versions.

Python
```
df.rename(columns={"PTS": "Points", "REB": "Rebounds", "AST": "Assists"}, inplace=True)
```
5. Detect Outliers (Step 6)

Although optional, detecting outliers is a good practice. The assignment gives an example of flagging minutes played greater than 45.

Python
```
# Create a new column to flag any minutes considered an outlier (> 45)
df["Outlier_MIN"] = df["MIN"] > 45
```
6. Export Clean Data (Step 9)

Your final, clean DataFrame should be saved as a CSV file.

Python
```
# Save the final cleaned data
df.to_csv("cleaned_week7_assignment.csv", index=False)
```

Reflection Questions

When completing the reflection (Step 8), use the information you gained during the cleaning process:

- **How many missing values did you handle?** You handled 4 missing values (1 in PTS, 1 in REB, 1 in AST, 1 in MIN).
- **Which cleaning method (mean, median, or drop) worked best?** Imputation (mean/median) worked best because it preserved all 5 player records, preventing data loss.
- **Why is data cleaning crucial before performing analytics?** It prevents errors in calculations (e.g., dividing by NaN or skewed averages) and ensures that statistical results (like correlation) are based on reliable and complete data.