9️⃣ Week 9: Feature Engineering in Practice

This code loads the data and creates 5 complex engineered features, performing necessary checks and visualizations.

```
# Week 9 Assignment: Feature Engineering in Practice


import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


# --- Data Loading (Assumes 'clean_box_scores.csv' was created in the setup cell) ---

df = pd.read_csv('clean_box_scores.csv')

print("Initial Data Head:")

print(df.head())


# --- 1. Engineer Features ---


# 1. Player Efficiency (EFF) - Advanced Box Score Metric

df['Efficiency'] = ((df['PTS'] + df['REB'] + df['AST'] + df['STL'] + df['BLK']) - (df['FGA'] - df['FGM']) - (df['FTA'] - df['FTM']) - df['TO']) / df['GP']


# 2. True Shooting % (TS%)

df['TS%'] = df['PTS'] / (2 * (df['FGA'] + 0.44 * df['FTA']))


# 3. Usage Rate (USG%) - Requires per-minute normalization

df['USG%'] = 100 * ((df['FGA'] + 0.44 * df['FTA'] + df['TO']) * df['MIN']) / df['Team_MIN']
```

```
# 4. Workload Ratio (Minutes as fraction of total game time)
df['Workload_Ratio'] = df['MIN'] / df['Team_MIN']

# 5. Rolling 5-Game Efficiency (Contextual Feature)
df['Rolling_Efficiency'] = df.groupby('Player')['Efficiency'].transform(lambda x: x.rolling(5, min_periods=1).mean())

print("\nEngineered Features Added:")
print(df[['Player', 'Efficiency', 'TS%', 'USG%', 'Rolling_Efficiency']].head())


# --- 2. Visualize New Features ---
# Visualization 1: Efficiency by Position
plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='Position', y='Efficiency', palette='viridis')
plt.title('EFF Score Distribution by Position')
plt.show()

# Visualization 2: True Shooting vs Usage Rate (Efficiency vs Volume)
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='USG%', y='TS%', hue='Player')
plt.title('True Shooting vs Usage Rate')
plt.show()
```

```
# --- 3. Correlation Analysis (Redundancy Check) ---
plt.figure(figsize=(12, 8))
# Select only numerical columns for correlation calculation
sns.heatmap(df.select_dtypes(include=np.number).corr(),
        cmap='coolwarm',
        annot=False)
plt.title('Feature Correlation Matrix (Original + Engineered)')
plt.show()



# --- 4. Interpretation (Example) ---
print("\n--- Interpretation ---")
print("The correlation matrix shows that our engineered 'Efficiency' is highly correlated with basic scoring metrics, confirming its role as a summary stat. However, 'TS%' and 'USG%' are less correlated with each other, making them valuable in a model as they decouple a player's *efficiency* from their *volume*. The 'Rolling_Efficiency' feature is crucial because it captures a player's recent form, offering a better prediction of immediate impact than a simple season average.")
```