

```

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

# Read the data from the dataset file
data = pd.read_csv('Assignment_Dataset.csv', parse_dates=['Date'])

# Convert dates to the desired format
data['Date'] = data['Date'].dt.strftime("%b/%y")

# Calculate the moving average of PR
data['PR_MA'] = data['PR'].rolling(window=30, min_periods=1).mean()

# Calculate the target budget PR values
start_value = 73.9
budget_reduction_rate = 0.008 # 0.8% per year
min_year = data['Date'].str[-2:].astype(int).min()

data['Budget_PR'] = start_value - (budget_reduction_rate *
(data['Date'].str[-2:].astype(int) - min_year))

# Plot the scatter plot
fig, ax = plt.subplots(figsize=(12, 8))

# Scatter plot with color-coded points based on GHI value
colors = ['navy', 'lightblue', 'orange', 'brown']
conditions = [data['GHI'] < 2, (data['GHI'] >= 2) & (data['GHI'] < 4),
(data['GHI'] >= 4) & (data['GHI'] < 6), data['GHI'] >= 6]
labels = ['< 2', '2-4', '4-6', '>= 6']
for i, cond in enumerate(conditions):
    scatter_data = data[cond & (data['PR'] > data['Budget_PR'])]
    sns.scatterplot(data=scatter_data, x='Date', y='PR', ax=ax,
color=colors[i], label=f'GHI: {labels[i]}', alpha=0.5)

# Line plots for PR moving average, target budget PR, and average PR
sns.lineplot(data=data, x='Date', y='PR_MA', ax=ax, color='red',
label='Moving Average PR')
sns.lineplot(data=data, x='Date', y='Budget_PR', ax=ax,
color='darkgreen', label='Target Budget PR')

# Calculate and plot the average PR for different periods
periods = [7, 30, 60, 90] # Adjust if needed
for period in periods:
    average_PR = data['PR'].rolling(window=period,
min_periods=1).mean()
    sns.lineplot(data=data, x='Date', y=average_PR, ax=ax,
label=f'Average PR ({period} days)')

# Set the labels, title, and legend
ax.set_xlabel('Date')
ax.set_ylabel('PR')

```

```
ax.set_title('Performance Ratio (PR) and Irradiation (GHI)')
ax.legend()
```

```
# Rotate x-axis labels for better readability
plt.xticks(rotation=45)
```

```
# Display the graph
plt.tight_layout()
plt.show()
```

