linear_regres

January 27, 2025

[ ]:
```
# Import necessary libraries
import statsmodels.api as sm
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Read dataset
file_path = 'C:/Users/user/Desktop/DESKTOP/New folder/Thuong/
 ↪Hw1_regression_11257079.ipynb.csv'
data = pd.read_csv(file_path, index_col=0)

# Display the first 17 rows
print(data.head(17))

# Display dataset dimensions
print("Dataset shape:", data.shape)

# Display dataset information
data.info()

# Verify column names
print("Columns in dataset:", data.columns)

# Create dummy PM2.5 and PM2.5_tomorrow columns for testing
data['PM2.5'] = np.random.randint(20, 100, size=len(data))
data['PM2.5_tomorrow'] = data['PM2.5'] + np.random.randint(-10, 10,␣
 ↪size=len(data))

# Extract x and y
x = data['PM2.5']
y = data['PM2.5_tomorrow']

# Scatter plot
plt.scatter(x, y, color='Green')
plt.xlabel('PM2.5')
plt.ylabel('PM2.5_tomorrow')
plt.title('PM2.5 AND PM2.5_tomorrow')
```

```
plt.grid()
plt.show()

# Regression line
plt.scatter(x, y, color='Green', label='Data Points')
plt.xlabel("PM2.5")
plt.ylabel("PM2.5_tomorrow")
plt.plot([x.min(), x.max()], [x.min(), x.max()], 'green', label='y = x Line')
plt.legend()
plt.grid()
plt.title("Scatter Plot with Diagonal")
plt.show()

# Linear regression with statsmodels
x_with_const = sm.add_constant(x)  # Add intercept to x
results_simple = sm.OLS(y, x_with_const).fit()  # Fit regression
print(results_simple.summary())

# Extract regression parameters
slope = results_simple.params['PM2.5']
intercept = results_simple.params['const']
print(f"Slope: {slope}, Intercept: {intercept}")

# Plot regression line
x_range = np.linspace(x.min(), x.max(), 100)
y_pred = slope * x_range + intercept
plt.scatter(x, y, color='Green', label='Data Points')
plt.plot(x_range, y_pred, color='Blue', label='Regression Line')
plt.xlabel("PM2.5")
plt.ylabel("PM2.5_tomorrow")
plt.title("PM2.5 AND PM2.5_tomorrow with Regression Line")
plt.legend()
plt.grid()
plt.show()

# Example DataFrame (replace this with reading from your file)
file_path = 'C:/Users/user/Desktop/DESKTOP/New folder/Thuong/
 ↪Hw1_regression_11257079.ipynb.csv'
data = pd.read_csv(file_path, index_col=0)


# Convert data to numeric where possible and handle 'NR'
data = data.set_index("AMB_TEM").replace("NR", None).apply(pd.to_numeric,␣
 ↪errors="coerce")

# Plotting each parameter
for parameter in data.index:
```

```
    plt.figure(figsize=(10, 6))
    plt.plot(data.columns, data.loc[parameter], marker="o", label=parameter)
    plt.title(f"{parameter} Over Time")
    plt.xlabel("Time (Columns)")
    plt.ylabel(parameter)
    plt.legend()
    plt.grid()
    plt.show()


# Create a sequential index for x-axis
record_values = range(1, len(data) + 1)

# Plot each parameter against the record values
plt.figure(figsize=(10, 6))
for column in data.columns:
    plt.plot(record_values, data[column], marker='o', label=column)  # Line␣
 ↪plot for each parameter

# Add labels, legend, and grid
plt.xlabel("Record Number")
plt.ylabel("Parameter Value")
plt.title("All Parameters vs. Record Number")
plt.legend()
plt.grid()
plt.show()
```

[ ]: