Single Linear Regression in Python

Introduction

This is an example of using Python code to perform a single linear regression analysis on a dataset. The data used for this example is derived from observations of female British middle school students and seeks to determine statistical correlation between the two following measures:

- The independent variable (height), represented on the X axis of the resulting scatter plot.
- The dependent, or predictor, variable (weight), represented on the Y axis.

The possible correlation between these two variables is determined using the Pearson Correlation Coefficient method which measures the strength of statistical correlation fit in the model as a number between 0 (no correlation) and either 1 (perfect positive correlation) or -1 (perfect negative correlation). The results are then displayed on a Seaborn module regression plot, a specialized form of scatter plot, for ease of visualization.

Python Code Details

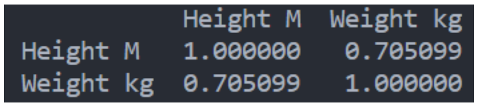
First, the data is loaded into a Pandas dataframe as follows (the path here is replaced with the actual local path to the csv file):

```
df = pd.read_csv("C:\Samples\HeightWeight.csv")
```

Next, for reference, the Pearson correlation value, or “r” value, between the two values is displayed:

```
print(df.corr())
```

which results in the following value output to the terminal:



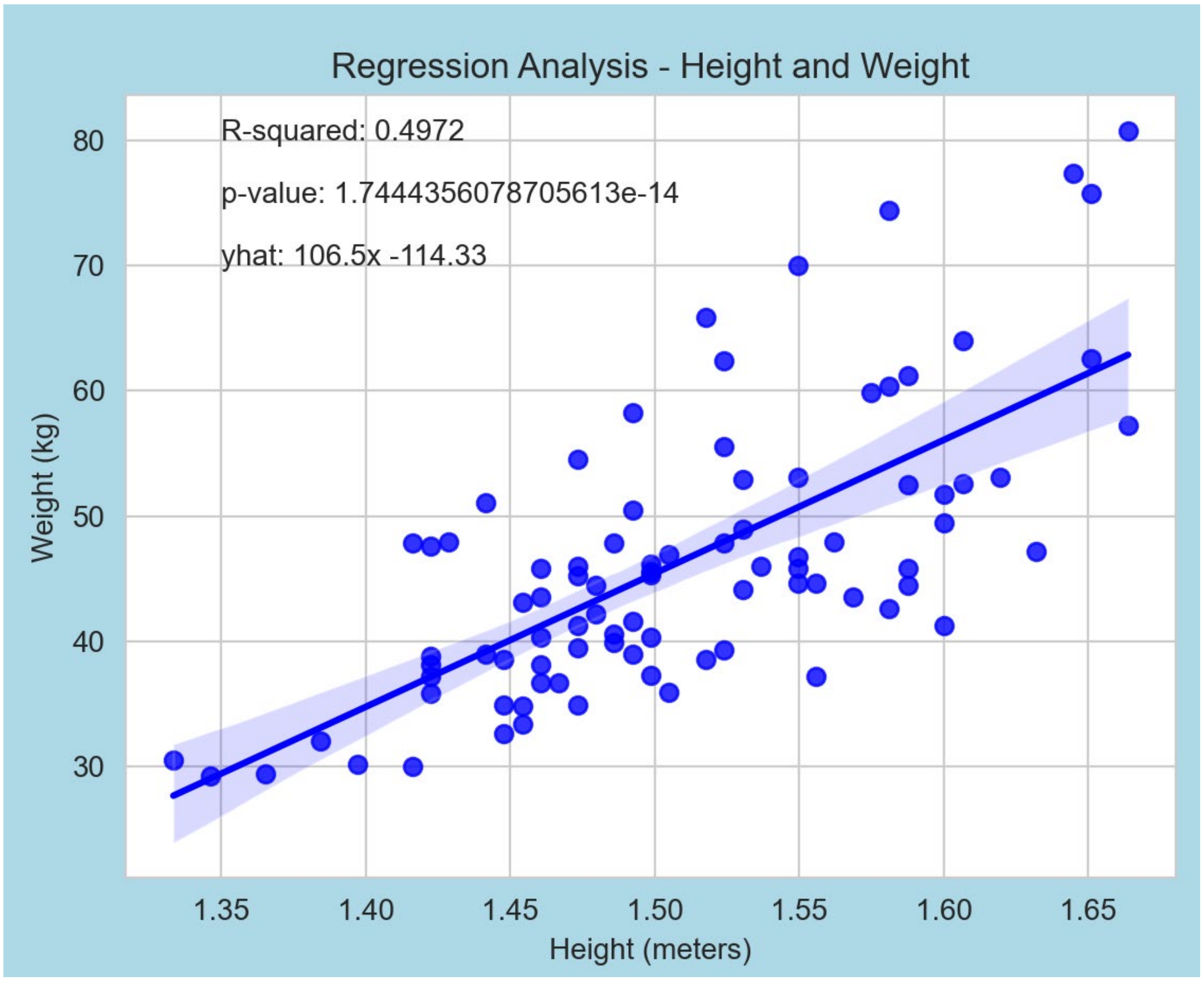
```
           Height M  Weight kg
Height M   1.000000   0.705099
Weight kg  0.705099   1.000000
```

The Correlation Coefficient of .705 between the height and weight values represents a moderately strong correlation fit, worthy or further examination. The full values used in the regression plot are then determined utilizing the Scipy Stats module’s linregress function in Python:

```
slope, intercept, r_value, p_value, std_err =
stats.linregress(df['Height M'], df['Weight kg'])
```

The r_value returned by the function is the Correlation Coefficient value, and the p_value, slope and intercept values are used in the regression plot annotations. The Seaborn regression plot is then displayed:

```
sns.regplot(x='Height M', y='Weight kg', data=df, color='blue')
```



The individual data points are displayed, along with the fitted line, representing the path through the data points with the lowest sum of residuals, or errors, between each of the data points and the fitted line. The annotations in the upper left represent:

- R-squared, the Correlation Coefficient value squared, also called the Coefficient of Determination. Squaring the Correlation Coefficient eliminates potential negative values and provides a widely accepted value of correlation model strength between 0 (no correlation) and 1 (perfect correlation).

- P-value, the measure of correlation confidence. Values below .05 represent over 95% correlation confidence level, indicating statistical significance, with the confidence strength growing as the P-value approaches zero.

- Yhat, or the predictor formula, which can be used to determine fitted line Y values for a given X axis value. It comprises the regression <u>Slope</u> value (106.5) multiplied by the X independent variable value which is then added (or subtracted) with the regression <u>Intercept</u> value (-114.33). For example, using an X value of 1.5 (one and a half meters in height) would predict the following Y axis weight value, which corresponds with the fitted line value indicated above:

```
106.5 x 1.5 – 114.33 = 45.42 kg predicted weight
```

The regression plot above visualizes a strong positive correlation between height and weight for the data being studied in this example. An important note for any correlation study is that correlation does not necessarily imply causation, as the two are very different concepts and may not have a direct relationship based on statistical correlation alone.

Complete Python Code with Comments

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

# Create a dataframe - replace following path with appropriate local path
df = pd.read_csv("C:\Samples\HeightWeight.csv")

# Print correlation values to terminal
print(df.corr())

# Perform linear regression
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Height M'], df['Weight kg'])

# Set the style to 'whitegrid'
sns.set_style("whitegrid")

# Create the seaborn regplot
sns.regplot(x='Height M', y='Weight kg', data=df, color='blue')

# Add R-squared value, p-value, and yhat formula annotations to the plot
plt.text(1.35,80,f'R-squared: {round(r_value**2,4)}')
plt.text(1.35,75,f'p-value: {p_value}')
if (intercept < 0):
   plt.text(1.35, 70,f'yhat: {round(slope,2)}x {round(intercept,2)}')
else:
   plt.text(1.35, 70,f'yhat: {round(slope,2)}x + {round(intercept,2)}')

# Add axis labels and a plot title
plt.xlabel('Height (meters)')
plt.ylabel('Weight (kg)')
plt.title('Regression Analysis - Height and Weight')

# Change the background color to light blue
plt.gcf().set_facecolor('lightblue')

# Show the plot
plt.show()
```