Simple Linear Regression

Gujarati, D. N. (2010). Econometrics (5th ed.). New York: McGraw-Hill Interamericana.

The methodology for simple linear regression involves the following steps:

Formulation of the theory or hypothesis Specification of the mathematical model of the theory Specification of the econometric or statistical model of the theory Data collection Estimation of the econometric model parameters Hypothesis testing Forecasting Application of the model for control or policy purposes

1. Formulation of the Theory or Hypothesis

John Maynard Keynes, the economist, posits in his work:

> "The fundamental psychological law...is that men are disposed, as a rule and on average, to increase their consumption as their income increases, but not by as much as the increase in their income."

In essence, Keynes suggests that the marginal propensity to consume (MPC) is greater than zero but less than one.

2. Specification of the Mathematical Model of the Theory

Although Keynes postulated a positive relationship between consumption and income, he did not specify the exact functional form of this relationship. A mathematical economist might propose the following form for the Keynesian consumption function:

$$Y = \beta_1 + \beta_2 X \quad \text{with} \quad 0 < \beta_2 < 1$$

where:

- (Y) represents consumption expenditure,
- (X) denotes income,
- (\beta_1) and (\beta_2) are the parameters of the model, representing the intercept and the slope, respectively.

The slope coefficient (\beta_2) measures the MPC. This equation suggests that consumption is linearly related to income. This is an example of a single-equation model, as it involves only one equation.

3. Specification of the Econometric or Statistical Model of the Theory

The mathematical model in the previous equation assumes an exact relationship between consumption and income. However, in reality, the relationships between economic variables are imprecise. To reflect this, an error term or disturbance, (u), is introduced into the equation:

$$Y = \beta_1 + \beta_2 X + u$$

where (u) is a random variable that accounts for factors not explicitly considered in the model.

4. Data

To estimate the parameters of the econometric model, we need data. In Gujarati's book, data up to 2005 from the United States is used. In this example, we will use private consumption and GDP data from the World Bank, connecting to their API.

In [ ]:
```
# consumption
import pandas as pd
import world_bank_data as wb
pd.set_option('display.max_rows', 6)
c = wb.get_series("NE.CON.PRVT.CD", id_or_value='id', simplify_index=False)
c = c.reset_index(name = "C")


## GDP
pib = wb.get_series("NY.GDP.MKTP.CD", id_or_value='id', simplify_index=False)
pib = pib.reset_index(name = "PIB")



# Filter data
cmex = c[c["Country"] == "MEX"]
pibmex = pib[pib["Country"] == "MEX"]

df = pd.merge(cmex, pibmex, on  = "Year")
df = df[["Year","C","PIB"]]
df.head(10)
```

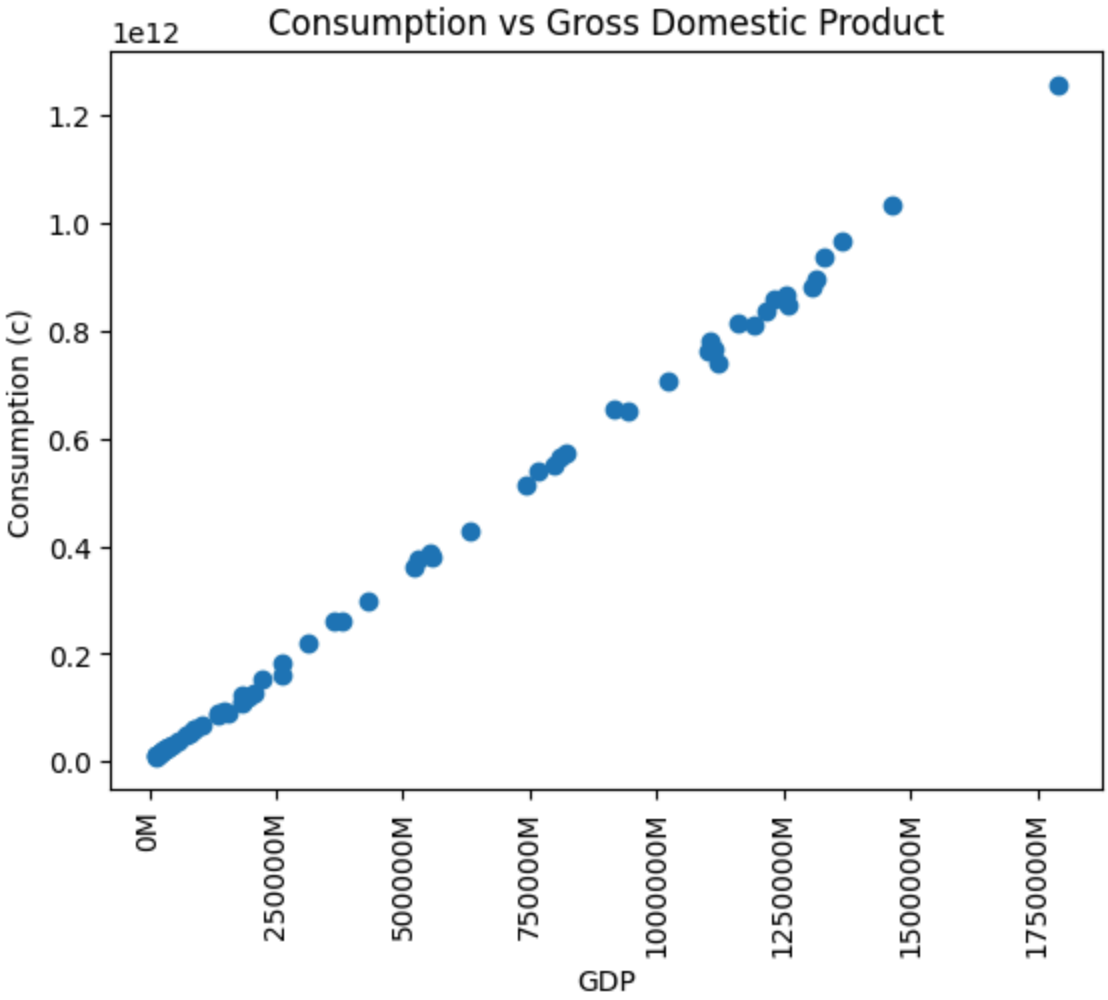
Out[ ]:

	Year	C	PIB
0	1960	1.006864e+10	1.304000e+10
1	1961	1.088552e+10	1.416000e+10
2	1962	1.172704e+10	1.520000e+10
...	...	...	...
7	1967	1.945720e+10	2.656000e+10
8	1968	2.175656e+10	2.936000e+10
9	1969	2.370496e+10	3.248000e+10

10 rows × 3 columns

In [ ]:
```
import matplotlib.pyplot as plt
plt.scatter(df["PIB"], df["C"])
plt.ticklabel_format(style='plain', axis='x')
plt.gca().xaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: '{:.0f}M'.format
plt.xticks(rotation=90)
plt.title('Consumption vs Gross Domestic Product')
plt.xlabel('GDP')
plt.ylabel('Consumption (c)')
```

Out[ ]:
```
Text(0, 0.5, 'Consumption (c)')
```



5. Estimation of Econometric Model Parameters

Using statistical techniques such as regression analysis, we can estimate the values of $\beta 1$ and $\beta 2$. We will use the statsmodels library and sklearn to split the dataset into training and testing sets.

In [ ]:
```
from sklearn.model_selection import train_test_split
import statsmodels.api as sm

# Train and test
x = df["PIB"].values
X = x[:, None]
y = df["C"].values
X = sm.add_constant(x)  # statsmodels constant
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

#Train Model
model = sm.OLS(y_train, X_train).fit()
```

6. Hypothesis Testing

After executing the model, it is necessary to validate that the parameters are significant, along with interpreting other instances.

In [ ]:
```
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                 5.604e+04
Date:                Tue, 02 Jul 2024   Prob (F-statistic):           1.31e-76
Time:                        13:28:50   Log-Likelihood:                -1249.6
No. Observations:                  51   AIC:                             2503.
Df Residuals:                      49   BIC:                             2507.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -3.903e+09   2.18e+09     -1.789      0.080   -8.29e+09    4.82e+08
x1             0.6960      0.003    236.730      0.000       0.690       0.702
==============================================================================
Omnibus:                       11.603   Durbin-Watson:                   2.207
Prob(Omnibus):                  0.003   Jarque-Bera (JB):               12.199
Skew:                          -0.937   Prob(JB):                      0.00224
Kurtosis:                       4.494   Cond. No.                     1.07e+12
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 1.07e+12. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Interpretation:

- R-squared and Adjusted R-squared: Both R-squared and adjusted R-squared are very high (0.999), indicating that the model explains nearly all of the variability in the dependent variable (y).
- F-statistic and Prob (F-statistic): The F-statistic is 5.604e+04 with a very low probability (1.31e-76), suggesting that the overall model is statistically significant.
- Coefficients (const and x1):
  - The coefficient for the constant (const) is -3.903e+09 with a standard error of 2.18e+09. It is not statistically significant at the conventional levels (p-value = 0.080).
  - The coefficient for x1 (the independent variable) is 0.6960 with a standard error of 0.003. It is highly statistically significant (p-value < 0.001).
- Omnibus, Jarque-Bera, Skewness, and Kurtosis: These tests provide information about the normality and distribution of residuals. A significant p-value in Omnibus (0.003) and

Jarque-Bera (0.00224) indicates that residuals may not be normally distributed.

- Durbin-Watson: The value of 2.207 suggests that there is little autocorrelation in the residuals.
- Condition Number: The large condition number (1.07e+12) indicates potential multicollinearity or numerical issues in the model.

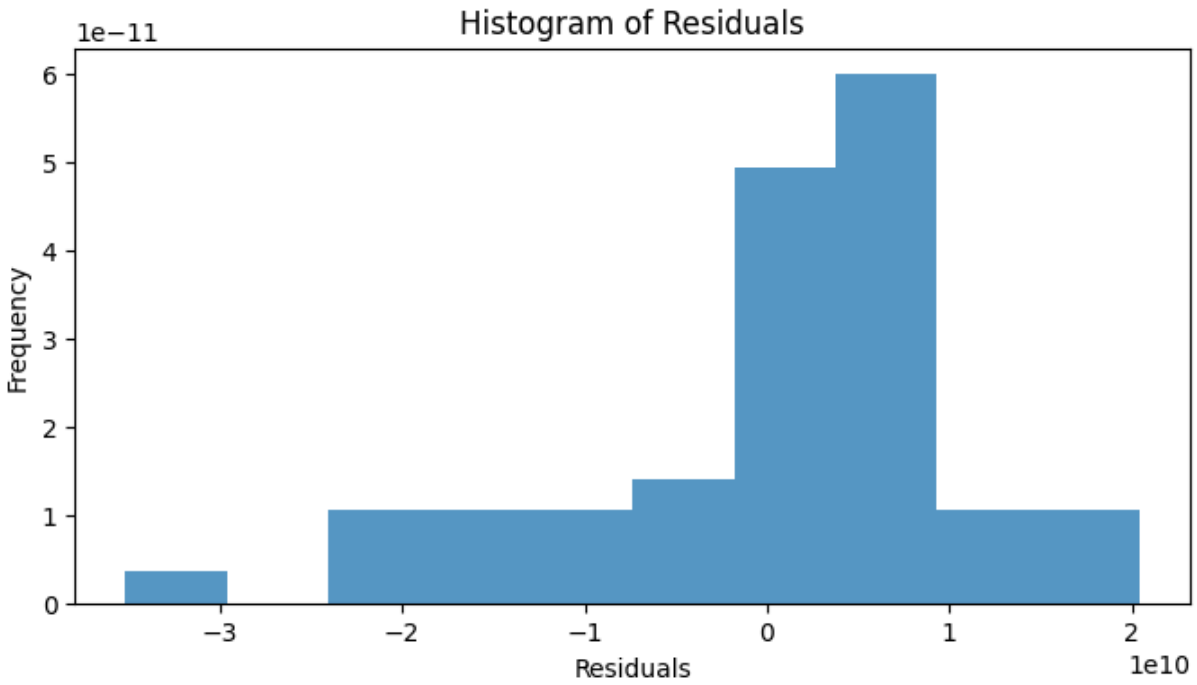
6.1 Residuals Histogram

According to the results from visualizing the residuals, they do not follow a normal distribution. Additionally, the Shapiro-Wilk test suggests that they are not normally distributed. This may lead us to reconsider the assumptions of the model.

In [ ]:
```
import scipy.stats as stats
import matplotlib.pyplot as plt
res = model.resid
shapiro_test = stats.shapiro(res)
print(f"Shapiro-Wilk Test: p-value = {shapiro_test[1]}")
plt.figure(figsize=(8, 4))
plt.hist(res, bins=10, density=True, alpha=0.75)
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.title('Histogram of Residuals')
```

```
Shapiro-Wilk Test: p-value = 0.0012528270278368136
```

Out[ ]:
```
Text(0.5, 1.0, 'Histogram of Residuals')
```



7. Forecast

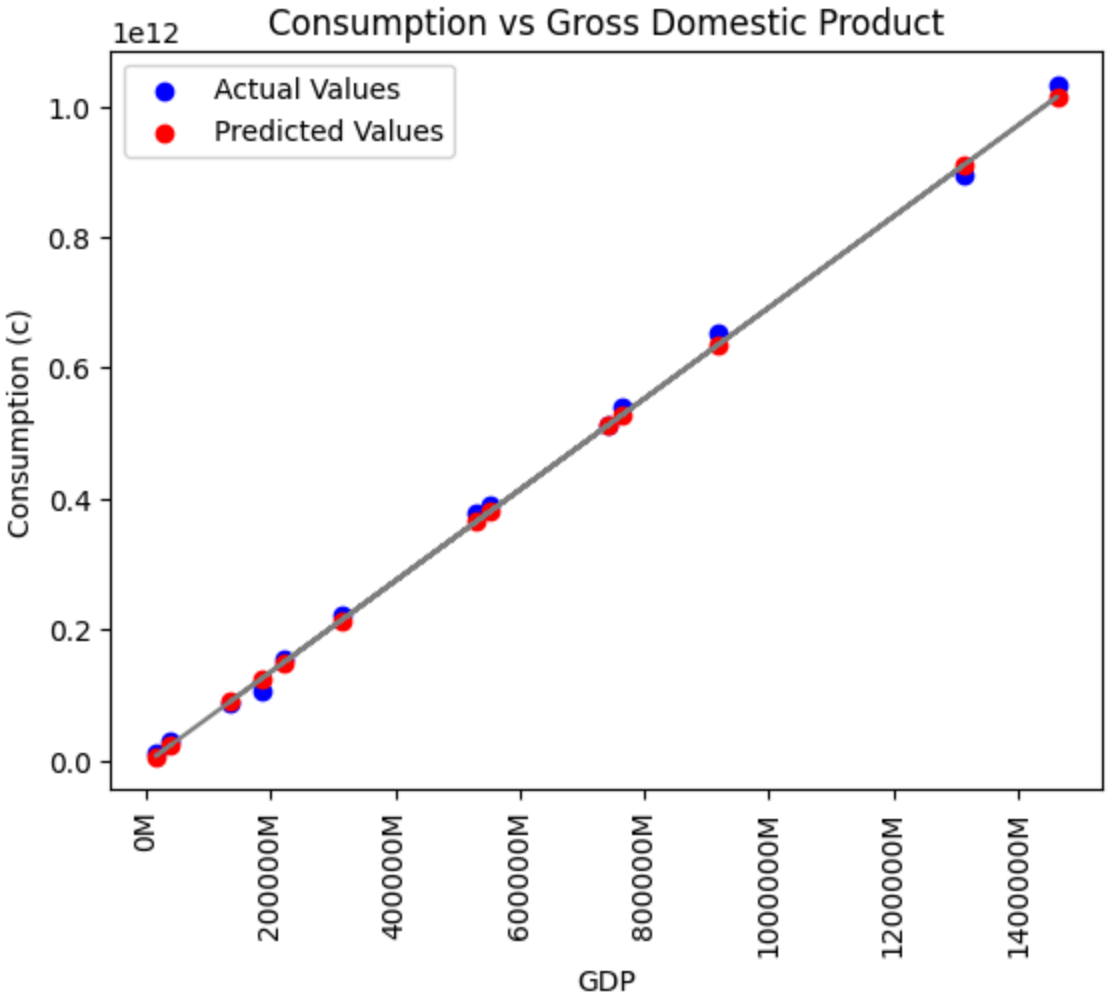
Using the estimated model, we can predict future values of the dependent variable Y based on known or expected values of the explanatory variable X. For example, to predict consumption expenditure:

$$\hat{Y} = -3.903e + 0.6960 * GDP$$

In [ ]:
```
y_pred_test = model.predict(X_test)
plt.scatter(X_test[:,1], y_test, color='blue', label='Actual Values')  # Blue for a
plt.scatter(X_test[:,1], y_pred_test, color='red', label='Predicted Values')  # Rea


plt.plot(X_test[:,1], y_pred_test, color='gray')  # Line reg
plt.ticklabel_format(style='plain', axis='x')
plt.gca().xaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: '{:.0f}M'.format
plt.xticks(rotation=90)
plt.title('Consumption vs Gross Domestic Product')
plt.xlabel('GDP')
plt.ylabel('Consumption (c)')
plt.legend(loc='upper left')  # Specify location for the legend
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x2686285a4b0>
```



8. Using the model for control or policy purposes

The estimated model can be used for control purposes or public policy. For instance, it can help determine the necessary GDP to achieve adequate expenditure. The model enables us to make predictions, although it is clear that certain parameters may need adjustment. This example demonstrates the value of linear regression.