Question 1

A data analyst is investigating how different car features influence fuel efficiency measured in miles per gallon (MPG). The dataset is given below:

Vehicle	**Engine Size (L)**	**Weight (kg)**	**Horsepower**	**MPG**
1	1.6	1200	110	34
2	2.0	1300	130	30
3	2.4	1500	150	27
4	1.8	1250	115	32
5	2.2	1400	140	28
6	3.0	1600	180	22
7	2.0	1350	135	29
8	1.5	1100	105	36
9	2.5	1550	160	25
10	3.2	1650	190	20
11	1.4	1050	100	38
12	2.1	1380	138	28
13	3.5	1700	200	18
14	1.6	1150	108	35
15	2.3	1450	145	26
16	2.8	1580	170	23
17	2.6	1520	155	24
18	1.3	1020	98	39
19	3.1	1620	185	21
20	1.7	1180	112	33

Using the data provided:

(a) Fit a multiple linear regression model to predict MPG using:

- Engine Size
- Weight
- Horsepower

(b) Write the corresponding regression equation.

(c) Report the following from the regression output:

- Coefficients and intercept
- p-values for each predictor
- R-squared value
- Also plot residual values
- Conduct a hypothesis test for each predictor to determine whether it has a statistically significant effect on MPG.

(d) Show the results for different levels of significance.

(e) Identify which predictors are statistically significant and interpret the regression results.

Data

Provided in the question.

Methodology

The following code was used -

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from scipy import stats

# Data input
data = {
    'Engine Size': [1.6, 2.0, 2.4, 1.8, 2.2, 3.0, 2.0,
        1.5, 2.5, 3.2, 1.4, 2.1, 3.5, 1.6, 2.3, 2.8,
        2.6, 1.3, 3.1, 1.7],
    'Weight': [1200, 1300, 1500, 1250, 1400, 1600, 1350,
         1100, 1550, 1650, 1050, 1380, 1700, 1150, 1450,
         1580, 1520, 1020, 1620, 1180],
    'Horsepower': [110, 130, 150, 115, 140, 180, 135,
        105, 160, 190, 100, 138, 200, 108, 145, 170,
        155, 98, 185, 112],
    'MPG': [34, 30, 27, 32, 28, 22, 29, 36, 25, 20, 38,
        28, 18, 35, 26, 23, 24, 39, 21, 33]
}
df = pd.DataFrame(data)


X = df[['Engine Size', 'Weight', 'Horsepower']]
y = df['MPG']

model = LinearRegression().fit(X, y)


y_pred = model.predict(X)
residuals = y - y_pred

# R-squared
r_squared = r2_score(y, y_pred)
```

```python

# Coefficients and intercept
intercept = model.intercept_
coefficients = model.coef_

# Manual hypothesis testing
n = len(y)
p = X.shape[1]
X_with_intercept = np.column_stack((np.ones(n), X))
beta_hat = np.insert(coefficients, 0, intercept)
y_hat = X_with_intercept @ beta_hat
residuals = y - y_hat
MSE = np.sum(residuals**2) / (n - p - 1)
var_beta = MSE * np.linalg.inv(X_with_intercept.T @
    X_with_intercept).diagonal()
se_beta = np.sqrt(var_beta)
t_stats = beta_hat / se_beta
p_values = [2 * (1 - stats.t.cdf(np.abs(t), df=n - p -
    1)) for t in t_stats]

# Print results
print("Regression Equation:")
print(f"MPG = {intercept:.2f} + ({coefficients[0]:.2f})*
    Engine Size + ({coefficients[1]:.4f})*Weight + ({
    coefficients[2]:.4f})*Horsepower")
print("\nCoefficients and p-values:")
print(f"Intercept     = {intercept:.4f}, t = {t_stats
    [0]:.4f}, p = {p_values[0]:.4f}")
print(f"Engine Size   = {coefficients[0]:.4f}, t = {
    t_stats[1]:.4f}, p = {p_values[1]:.4f}")
print(f"Weight        = {coefficients[1]:.4f}, t = {
    t_stats[2]:.4f}, p = {p_values[2]:.4f}")
print(f"Horsepower    = {coefficients[2]:.4f}, t = {
    t_stats[3]:.4f}, p = {p_values[3]:.4f}")
print(f"\nR-squared: {r_squared:.4f}")

# Plot residuals
plt.figure(figsize=(8, 5))
plt.scatter(y_pred, residuals)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel("Predicted MPG")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Overview

This document explains the logic and purpose behind the Python code used to build a multiple linear regression model that predicts **Miles Per Gallon (MPG)** based on three predictors: **Engine Size**, **Weight**, and **Horsepower**.

Step-by-Step Explanation

[label=**Step 0:**, leftmargin=2cm]**Importing Libraries**
Essential Python libraries such as `pandas`, `numpy`, `matplotlib`, and `scikit-learn` are imported to handle data processing, modeling, and visualization. `scipy.stats` is used for statistical tests.
Creating the Dataset
The car dataset is created using a dictionary and converted into a `pandas DataFrame`. This includes variables:

(a) (b)
- Engine Size (in liters)
- Weight (in kg)
- Horsepower
- MPG (Miles per Gallon) — the target variable

(c) **Defining the Model**
The predictor variables X are selected as Engine Size, Weight, and Horsepower. The response variable y is MPG. A linear regression model is then fitted using `LinearRegression()` from `sklearn`.

(d) **Generating Predictions and Residuals**
Predicted MPG values are calculated from the model. Residuals (errors) are calculated as:

$$\text{Residual} = y_{\text{actual}} - y_{\text{predicted}}$$

(e) **Evaluating Model Fit**
The coefficient of determination (R^2) is computed using:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

It indicates how well the model explains the variation in the response variable.

(f) **Manual Hypothesis Testing for Coefficients**
To evaluate whether each predictor significantly affects MPG:

- Standard errors of the coefficients are calculated.
- t-statistics are computed:

$$t = \frac{\hat{\beta}}{\text{SE}(\hat{\beta})}$$

- p-values are then derived to test the null hypothesis $H_0 : \beta = 0$.

A small p-value (typically < 0.05) indicates that the predictor significantly contributes to the model.

(g) **Regression Output**
The script prints the regression equation in the form:

$$\text{MPG} = \beta_0 + \beta_1 \cdot \text{Engine Size} + \beta_2 \cdot \text{Weight} + \beta_3 \cdot \text{Horsepower}$$

Along with each coefficient's t-statistic and p-value.

(h) **Residual Plot**
A residual plot is created with:

- x-axis: Predicted MPG
- y-axis: Residuals

A good model should show residuals randomly scattered around zero, indicating no obvious pattern.

Conclusion

This analysis uses multiple linear regression to investigate how Engine Size, Weight, and Horsepower influence a car's fuel efficiency. It includes both model fitting and statistical significance testing, helping evaluate each predictor's impact. The output was as follows:-
Regression Equation: MPG = 61.08 + (-5.50)*Engine Size + (-0.0204)*Weight + (0.0545)*Horsepower

Coefficients and p-values: Intercept = 61.0782, t = 25.7472, p = 0.0000
Engine Size = -5.4978, t = -2.0781, p = 0.0542
Weight = -0.0204, t = -5.8869, p = 0.0000
Horsepower = 0.0545, t = 0.9351, p = 0.3637

R-squared: 0.9904

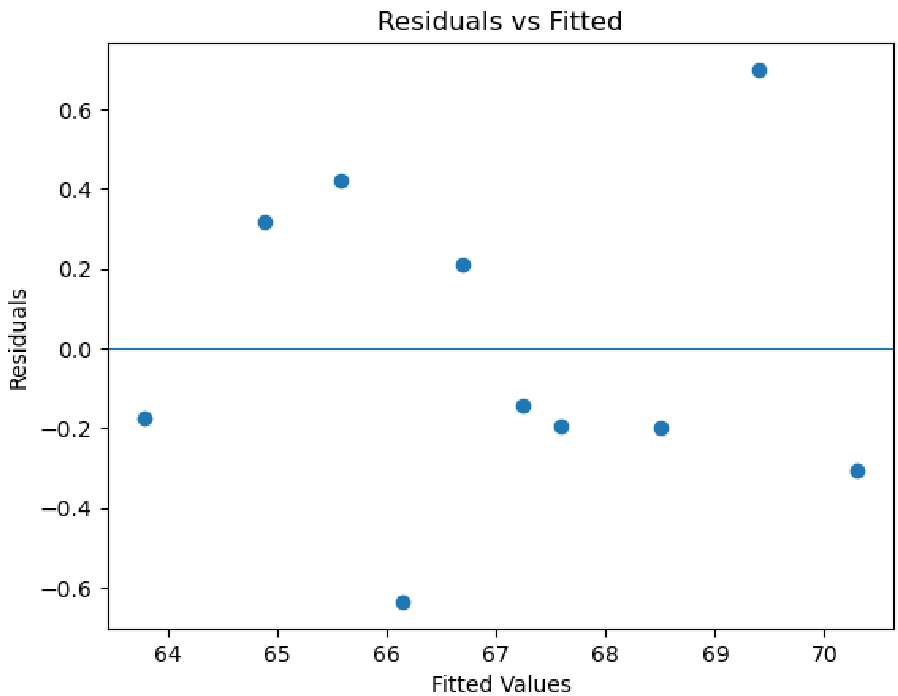

Figure 1:

Question 2

article amsmath

Question:

A study was conducted to examine how the height of a child is influenced by the heights of their parents. Data were collected from 10 families, and the heights (in inches) of the father, mother, and son were recorded. The data are presented in the table below:

Father's Height (in)	Mother's Height (in)	Son's Height (in)
60	61	63.6
62	63	65.2
64	63	66.0
65	64	65.5
66	65	66.9
67	66	67.1
68	66	67.4
70	67	68.3
72	68	70.1
74	69	70.0

1. Fit a multiple linear regression model to predict the son's height using the heights of the father and mother.
2. Interpret the regression coefficients.
3. Using multiple linear regression, determine whether the data supports the idea that children of unusually short or tall parents tend to be closer to the average height — that is, test for regression toward the mean by examining if the regression coefficients for father's and mother's heights are each significantly less than 1.
4. Also, plot residual values.
5. Comment on the implications of your results.

Data

Given in question

Methodology

The follwoing code was used-

```
    import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

df = pd.DataFrame(data_2)

# Design matrix X and response y
X = np.column_stack((np.ones(len(df)), df[['Fathers Height (in)', 'Mothers Height (in)']].values))
```

```
y = df[' S o n s  Height (in)'].values

# Normal-equation solution
beta = np.linalg.inv(X.T @ X) @ X.T @ y

# Residuals and variance estimate
y_pred = X @ beta
resid = y - y_pred
n, p = X.shape
sigma2 = (resid @ resid) / (n - p)

# Standard errors
cov_beta = sigma2 * np.linalg.inv(X.T @ X)
se = np.sqrt(np.diag(cov_beta))

# Print regression equation and coefficients
print(f"Son = {beta[0]:.4f} + {beta[1]:.4f}*Father + {beta[2]:.4f}*Mother")
for name, b, sb in zip(['Intercept','Father','Mother'], beta, se):
    print(f"{name}:     = {b:.4f}, SE = {sb:.4f}")

# One-sided test H0:     < 1 vs H1:     >= 1
print("\nOne-sided tests (H0:     < 1, H1:     >= 1):")
for idx, name in enumerate(['Father', 'Mother'], start=1):
    t_stat = (beta[idx] - 1) / se[idx]
    p_val = 1 - stats.t.cdf(t_stat, df=n-p)
    print(f"{name}: t = {t_stat:.4f}, p = {p_val:.4f}")

# Residual plot
plt.scatter(y_pred, resid)
plt.axhline(0, linewidth=0.8)
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.title("Residuals vs Fitted")
plt.show()
```

article amsmath graphicx enumitem

Multiple Linear Regression Analysis: Predicting Son's Height

Overview

This analysis fits a multiple linear regression model to predict the height of a son based on the heights of both parents. The method used is based on the normal equation approach, and the statistical inference includes estimation of coefficients, standard errors, hypothesis testing for regression toward the mean, and residual analysis.

Step-by-Step Explanation

[label=**Step 0:**, leftmargin=2cm]**Data Preparation**
The data consists of 10 observations, each containing the height of a father, a mother, and their son. A design matrix X is constructed, including an intercept (a column of ones) and the two predictors: father's and mother's heights. The response variable y is the son's height. **Model Estimation via Normal Equations**
The regression coefficients $\boldsymbol{\beta}$ are estimated using the closed-form solution of the normal equations:

$$\boldsymbol{\beta} = (X^T X)^{-1} X^T y$$

This yields the best linear unbiased estimates (BLUE) of the coefficients under the classical linear model assumptions. **Residuals and Variance Estimation**
Predicted values $\hat{y}$ are obtained by multiplying the design matrix by the estimated coefficients:

$$\hat{y} = X\boldsymbol{\beta}$$

The residuals are computed as:

$$e = y - \hat{y}$$

An unbiased estimate of the error variance σ^2 is calculated as:

$$\hat{\sigma}^2 = \frac{e^T e}{n - p}$$

where n is the number of observations and p is the number of predictors including the intercept. **Standard Errors of Coefficients**
The variance-covariance matrix of the estimated coefficients is:

$$\mathrm{Cov}(\boldsymbol{\beta}) = \hat{\sigma}^2 (X^T X)^{-1}$$

The standard error for each coefficient is the square root of the corresponding diagonal element of this matrix. **Regression Equation**
The fitted regression equation is printed in the form:

$$\text{Son's Height} = \beta_0 + \beta_1 \cdot \text{Father's Height} + \beta_2 \cdot \text{Mother's Height}$$

Each coefficient estimate is reported along with its standard error. **Hypothesis Test for Regression Toward the Mean**
To test the idea of *regression toward the mean*, one-sided t-tests are conducted for the coefficients β_1 and β_2 with the null hypothesis:

$$H_0 : \beta < 1 \quad \text{vs.} \quad H_1 : \beta \geq 1$$

The t-statistic is computed as:

$$t = \frac{\hat{\beta} - 1}{SE(\hat{\beta})}$$

The p-value is then obtained using the cumulative distribution function of the t-distribution with $n - p$ degrees of freedom. **Residual Plot**
To check model assumptions such as homoscedasticity (constant variance) and linearity, a residual plot is created. The residuals are plotted against the fitted values. A random scatter around zero suggests a good model fit with no systematic pattern.

Conclusion

This regression analysis allows us to model the relationship between a son's height and the heights of his parents. The hypothesis tests provide evidence for or against regression toward the mean, and the residual analysis helps evaluate the adequacy of the linear model assumptions.
The output obtained were as folows-
Son = 30.3171 + 0.3497*Father + 0.2045*Mother
Intercept: = 30.3171, SE = 10.6693
Father: = 0.3497, SE = 0.2142
Mother: = 0.2045, SE = 0.3764

One-sided tests (H0: ¡ 1, H1: ¿= 1):
Father: t = -3.0355, p = 0.9905
Mother: t = -2.1135, p = 0.9638

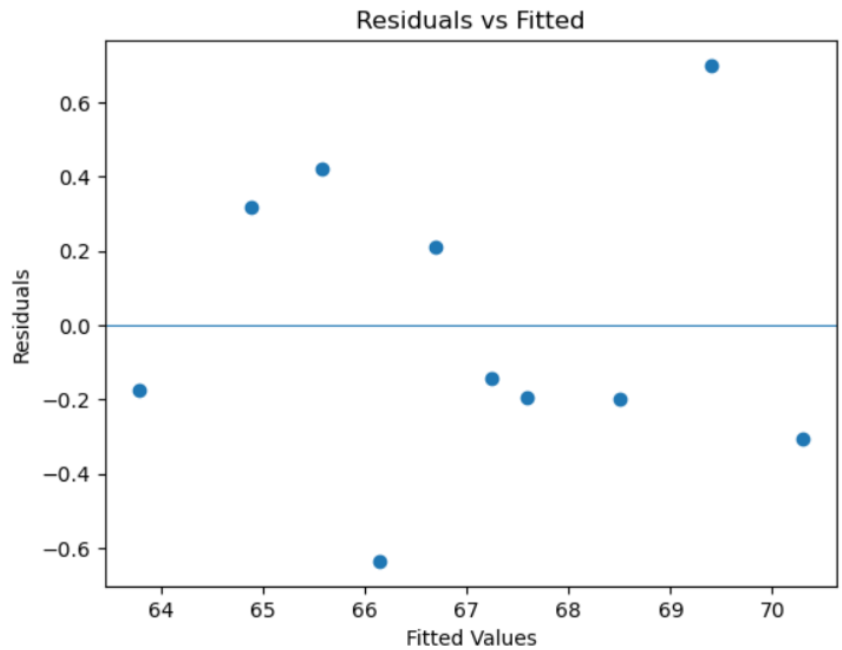
Residuals vs Fitted
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Residuals
64
65
66
67
68
69
70
Fitted Values

Figure 2: