In [8]:
```
#Question 3

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
import itertools
from scipy import stats
```

In [9]:
```
#Load the new dataset

df = pd.read_csv("C:\\Users\OkechPC\Downloads\FE-GWP1_model_selection_2.csv")
print("Data shape:", df.shape)
print(df.head(), "\n")

# Identify dependent and independent variables
y = df["Y"]
predictors = [c for c in df.columns if c != "Y"]
print("Candidate predictors:", predictors, "\n")
```

```
Data shape: (100, 6)
          Y       Z1         Z2        Z3        Z4        Z5
0  2.172296  0.121634 -0.051562  0.570616  1.279931  0.075233
1  0.502380  0.025446 -0.093062  0.304875 -0.582292  0.377388
2  0.711362 -0.136716 -0.082229 -0.191680 -0.647970  1.230986
3 -0.557168 -0.284459 -0.170922 -0.853670 -1.256146 -0.991686
4  1.500199  0.105205 -0.169141  0.826558  0.640945  1.099873

Candidate predictors: ['Z1 ', 'Z2', 'Z3', 'Z4', 'Z5']
```

In [10]:
```
#All-subsets OLS regressions

results = []
for k in range(1, len(predictors)+1):
    for subset in itertools.combinations(predictors, k):
        X = sm.add_constant(df[list(subset)])
        model = sm.OLS(y, X).fit()
        results.append({
            "subset": subset,
            "k": k,
            "adj_R2": model.rsquared_adj,
            "AIC": model.aic,
            "BIC": model.bic
        })

res_df = pd.DataFrame(results)
print("Top 5 models by Adjusted R²")
print(res_df.sort_values("adj_R2", ascending=False).head(), "\n")
print("Top 5 models by AIC")
print(res_df.sort_values("AIC").head(), "\n")
print("Top 5 models by BIC")
print(res_df.sort_values("BIC").head(), "\n")
```

```
Top 5 models by Adjusted R²
                  subset  k    adj_R2         AIC         BIC
30  (Z1 , Z2, Z3, Z4, Z5)  5  0.993573 -165.902248 -150.271227
29       (Z2, Z3, Z4, Z5)  4  0.989545 -118.185565 -105.159714
28      (Z1 , Z3, Z4, Z5)  4  0.985698  -86.850250  -73.824399
24          (Z3, Z4, Z5)  3  0.982814  -69.435042  -59.014362
25      (Z1 , Z2, Z3, Z4)  4  0.965353    1.630684   14.656535

Top 5 models by AIC
                  subset  k    adj_R2         AIC         BIC
30  (Z1 , Z2, Z3, Z4, Z5)  5  0.993573 -165.902248 -150.271227
29       (Z2, Z3, Z4, Z5)  4  0.989545 -118.185565 -105.159714
28      (Z1 , Z3, Z4, Z5)  4  0.985698  -86.850250  -73.824399
24          (Z3, Z4, Z5)  3  0.982814  -69.435042  -59.014362
25      (Z1 , Z2, Z3, Z4)  4  0.965353    1.630684   14.656535

Top 5 models by BIC
                  subset  k    adj_R2         AIC         BIC
30  (Z1 , Z2, Z3, Z4, Z5)  5  0.993573 -165.902248 -150.271227
29       (Z2, Z3, Z4, Z5)  4  0.989545 -118.185565 -105.159714
28      (Z1 , Z3, Z4, Z5)  4  0.985698  -86.850250  -73.824399
24          (Z3, Z4, Z5)  3  0.982814  -69.435042  -59.014362
25      (Z1 , Z2, Z3, Z4)  4  0.965353    1.630684   14.656535
```

In [11]:

```python
#Fit the best model (highest Adjusted R²)

best = res_df.sort_values("adj_R2", ascending=False).iloc[0]
best_predictors = list(best["subset"])
print(f"Best model predictors: {best_predictors}\n")

X_best = sm.add_constant(df[best_predictors])
best_model = sm.OLS(y, X_best).fit()
print(best_model.summary())
```

```
Best model predictors: ['Z1 ', 'Z2', 'Z3', 'Z4', 'Z5']

                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.994
Model:                            OLS   Adj. R-squared:                  0.994
Method:                 Least Squares   F-statistic:                     3062.
Date:                Tue, 16 Sep 2025   Prob (F-statistic):          2.07e-102
Time:                        18:23:50   Log-Likelihood:                 88.951
No. Observations:                 100   AIC:                            -165.9
Df Residuals:                      94   BIC:                            -150.3
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.0097      0.013     77.496      0.000       0.984       1.036
Z1             0.4487      0.058      7.781      0.000       0.334       0.563
Z2             0.2987      0.028     10.836      0.000       0.244       0.353
Z3            -0.4065      0.010    -39.578      0.000      -0.427      -0.386
Z4             1.0082      0.009    114.106      0.000       0.991       1.026
Z5             0.2572      0.013     20.449      0.000       0.232       0.282
==============================================================================
Omnibus:                        0.377   Durbin-Watson:                   2.046
Prob(Omnibus):                  0.828   Jarque-Bera (JB):                0.126
Skew:                           0.072   Prob(JB):                        0.939
Kurtosis:                       3.097   Cond. No.                         9.13
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

In [12]:
```python
#Multicollinearity check

vif_data = pd.DataFrame({
    "Variable": X_best.columns,
    "VIF": [variance_inflation_factor(X_best.values, i)
            for i in range(X_best.shape[1])]
})
print("\nVariance Inflation Factors")
print(vif_data, "\n")
```

```
Variance Inflation Factors
  Variable       VIF
0    const  1.614501
1       Z1  1.026366
2       Z2  1.022433
3       Z3  1.013241
4       Z4  1.003538
5       Z5  1.014190
```

In [13]:
```python
#Basic residual diagnostics

resid = best_model.resid
print("Residual mean:", resid.mean())
print("Residual std:", resid.std(ddof=1))
print("Skewness:", resid.skew())
print("Kurtosis:", resid.kurtosis())
```

```
Residual mean: 9.769962616701378e-17
Residual std: 0.09991608383209873
Skewness: 0.07338360732705755
Kurtosis: 0.164680748635607
```