Step 1: Import Libraries

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.impute import SimpleImputer
from statsmodels.stats.outliers_influence import variance_inflation_factor
# Step 1: Import Libraries
print("Step 1: Importing Libraries")
```

```
Step 1: Importing Libraries
```

Step 2: Load the Dataset

```
# Step 2: Load the Dataset
print("Step 2: Loading the Dataset")
data = pd.read_csv(r"NSSO68.csv")
data
```

```
Step 2: Loading the Dataset
<ipython-input-21-dc85d80149cc>:3: DtypeWarning: Columns (1) have mixed types. Speci
  data = pd.read_csv(r"NSSO68.csv")
```

	slno	grp	Round_Centre	FSU_number	Round
0	1	4099999999999999926524952937758 72.0	1	41000	68
1	2	4099999999999999926524952937758 72.0	1	41000	68
2	3	4099999999999999926524952937758 72.0	1	41000	68
3	4	4099999999999999926524952937758 72.0	1	41000	68
4	5	4099999999999999926524952937758 72.0	1	41000	68
...	...	...	...	...	...
101657	101658	7999999999999999708717035972 1984.0	1	79998	68
101658	101659	7999999999999999708717035972 1984.0	1	79998	68
101659	101660	7999999999999999708717035972 1984.0	1	79998	68
101660	101661	7999999999999999708717035972 1984.0	1	79998	68
101661	101662	7999999999999999708717035972 1984.0	1	79998	68

101662 rows × 384 columns

Step 3: Display Unique Values in a Column

```
# Step 3: Display Unique Values in a Column
print("Step 3: Displaying Unique Values in 'state_1' Column")
print(data['state_1'].unique())
```

```
Step 3: Displaying Unique Values in 'state_1' Column
['GUJ' 'ORI' 'CHTSD' 'MP' 'JRKD' 'WB' 'AP' 'MH' 'D&D' 'D&NH' 'MIZ' 'TRPR'
 'MANPR' 'ASSM' 'MEG' 'NAG' 'A&N' 'PNDCRY' 'TN' 'GOA' 'KA' 'KE' 'LKSDP'
 'SKM' 'Bhr' 'UP' 'RJ' 'ARP' 'DL' 'HR' 'Pun' 'HP' 'UT' 'Chandr' 'J$K']
```

Step 4: Subset Data and Select Columns

```
# Step 4: Subset Data and Select Columns
print("Step 4: Subsetting Data and Selecting Columns")
subset_data = data[data['state_1'] == 'AP'][['foodtotal_q', 'MPCE_MRP', 'MPCE_URP', 'Age', 'Meals_At_Home', 'Possess_ration_card', 'Edu
print(subset_data)
```

```
Step 4: Subsetting Data and Selecting Columns
       foodtotal_q  MPCE_MRP  MPCE_URP  Age  Meals_At_Home  \
6777     18.308732   3818.86   3780.50   35           60.0
6778     29.781670   4100.08   4322.00   24           60.0
6779     18.412530   2333.55   1769.40   45           60.0
6780     24.025527   2284.85   1986.25   36           60.0
6781     22.070518   1952.06   1224.20   36           84.0
...            ...       ...       ...  ...            ...
79719    17.800203   1304.09   1344.50   26           90.0
79720    47.825207   2581.20   2613.50   50           60.0
79721    13.250154   1374.35   1820.50   32           60.0
79722    21.725427   1391.12   1900.50   18           90.0
79723     0.000000   1536.62   1348.00   14            0.0

       Possess_ration_card  Education  No_of_Meals_per_day
```

```
6777                  1.0        7.0                 2.0
6778                  2.0       10.0                 2.0
6779                  1.0        8.0                 2.0
6780                  1.0       12.0                 2.0
6781                  1.0        1.0                 3.0
...                   ...        ...                 ...
79719                 1.0        6.0                 3.0
79720                 2.0        8.0                 2.0
79721                 2.0       12.0                 2.0
79722                 1.0        1.0                 3.0
79723                 2.0        6.0                 2.0

[6899 rows x 8 columns]
```

Step 5: Check for Missing Values

```
# Step 5: Check for Missing Values
print("Step 5: Checking for Missing Values")
print(subset_data.isna().sum())
```

```
Step 5: Checking for Missing Values
foodtotal_q              0
MPCE_MRP                 0
MPCE_URP                 0
Age                      0
Meals_At_Home          122
Possess_ration_card      0
Education                0
No_of_Meals_per_day      0
dtype: int64
```

Step 6: Impute Missing Values

```
# Step 6: Impute Missing Values
print("Step 6: Imputing Missing Values")
subset_data = subset_data.dropna()
```

```
Step 6: Imputing Missing Values
```

Step 7: Fit the Regression Model

```
# Step 7: Fit the Regression Model
print("Step 7: Fitting the Regression Model")
model = LinearRegression()
X = subset_data[['MPCE_MRP', 'MPCE_URP', 'Age', 'Meals_At_Home', 'Possess_ration_card', 'Education']]
y = subset_data['foodtotal_q']
model.fit(X, y)
```

```
Step 7: Fitting the Regression Model
▾ LinearRegression
LinearRegression()
```

Step 8: Print Regression Results

```
# Step 8: Print Regression Results
print("Step 8: Printing Regression Results")
print("Intercept:", model.intercept_)
print("Coefficients:", model.coef_)
```

```
Step 8: Printing Regression Results
Intercept: 7.6648416046440175
Coefficients: [ 2.16935253e-03  6.59504317e-04  1.10990344e-01  9.25106872e-02
 -1.57629375e+00  2.56726961e-03]
```

Step 9: Check for Multicollinearity (VIF)

```
# Step 9: Check for Multicollinearity (VIF)
print("Step 9: Checking for Multicollinearity (VIF)")
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
print(vif_data)
```

```
Step 9: Checking for Multicollinearity (VIF)
                feature        VIF
0             MPCE_MRP   6.576028
```

```
1             MPCE_URP   4.770529
2                  Age   9.723373
3         Meals_At_Home  10.504278
4  Possess_ration_card   9.006736
5            Education   3.836287
```

Step 10: Construct and Print the Regression Equation

```
# Step 10: Construct and Print the Regression Equation
print("Step 10: Constructing and Printing the Regression Equation")
equation = f"y = {model.intercept_:.2f}"
for i, coef in enumerate(model.coef_):
    equation += f" + {coef:.6f} * x{i+1}"
print(equation)
```

```
Step 10: Constructing and Printing the Regression Equation
y = 7.66 + 0.002169 * x1 + 0.000660 * x2 + 0.110990 * x3 + 0.092511 * x4 + -1.576294 * x5 + 0.002567 * x6
```

Step 11: Display Head of Selected Columns

```
# Step 11: Display Head of Selected Columns
print("Step 11: Displaying Head of Selected Columns")
print(subset_data[['MPCE_MRP', 'MPCE_URP', 'Age', 'Meals_At_Home', 'Possess_ration_card', 'Education', 'foodtotal_q']].head(1))
```

```
Step 11: Displaying Head of Selected Columns
      MPCE_MRP  MPCE_URP  Age  Meals_At_Home  Possess_ration_card  Education  \
6777   3818.86    3780.5   35           60.0                  1.0        7.0

      foodtotal_q
6777    18.308732
```

Start coding or generate with AI.