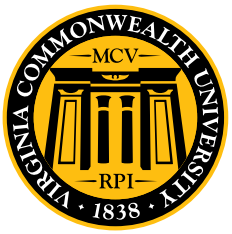


VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A2a: Regression - Predictive Analytics

FERAH SHAN SHANAVAS RABIYA

V01101398

Date of Submission: 16-06-2024

CONTENTS

Introduction

This study uses multiple regression analysis to examine the associations between independent variables and a dependent variable. The dataset used is the National Sample Survey Office (NSSO), which includes various socio-economic and demographic characteristics that can impact the dependent variable. The methodology involves evaluating the magnitude and statistical significance of each independent variable's influence on the dependent variable while accounting for other variables.

Regression diagnostics are performed to verify the assumptions of multiple regression and detect potential problems such as multicollinearity, heteroscedasticity, and outliers. The report provides a concise overview of the dataset relating to the state of Kerala and variables used, followed by a comprehensive analysis of the findings and their interpretations. The reliability of the regression model is verified through regression diagnostics.

Objectives

• Conduct an analysis of the "NSSO68" dataset to discover socio-economic and demographic variables that have a significant impact.

• Utilize multiple regression analysis to assess the associations between dependent and independent variables.

• Conduct comprehensive diagnostic tests to ensure the quality and consistency of the model.

• Enhance and refine the regression model by incorporating diagnostic discoveries.

• Offer practical observations and probable consequences for policy or future investigation.

• Provide guidance for decision-making processes related to the dependent variable in many areas.

Business Significance

Multiple regression analysis is a powerful tool for businesses to enhance their strategic decision-making, resource allocation, risk management, and competitive positioning. By identifying and measuring factors that impact the dependent variable, such as consumer spending and economic activity, businesses can customize their marketing strategies and

product offerings. This knowledge can also be used to optimize marketing resources and prioritize investments in specific product lines or geographical regions.

Regression analysis can also aid in risk management by recognizing and reducing hazards by evaluating how changes in economic indicators or customer behavior affect operations. By establishing benchmarks for performance evaluation, businesses can define achievable goals and monitor progress more efficiently.

Data-driven insights from regression analysis can provide a competitive advantage by enabling businesses to innovate and adapt more rapidly in response to changing situations. This knowledge can also be used in policy formulation, which involves analyzing problems, gathering information, and considering different options. Understanding factors influencing socio-economic results can guide measures that promote economic development, decrease disparities, or enhance public well-being.

Moreover, regression analysis can help in predicting future patterns and strategizing, enabling proactive decisions and investments. By utilizing data and statistical methodologies, businesses can gain a deeper understanding of their operations and external influences, improving their ability to navigate obstacles and seize advantageous circumstances.

Results and Interpretation using R

- **Check if there are any missing values in the data, identify them, and if there are, replace them with the mean of the variable.**

```
> # Check for missing values
> sum(is.na(subset_data$MPCE_MRP))
[1] 0
> sum(is.na(subset_data$MPCE_URP))
[1] 0
> sum(is.na(subset_data$Age))
[1] 0
> sum(is.na(subset_data$Possess_ration_card))
[1] 0
> sum(is.na(data$Education))
[1] 0
> sum(is.na(subset_data$Meals_At_Home))
[1] 0
> sum(is.na(subset_data$No_of_Meals_per_day))
[1] 0
> sum(is.na(subset_data$foodtotal_q))
[1] 0

# Replace missing values with mean
```

```
> impute_with_mean <- function(data, columns) {
+   data %>%
+     mutate(across(all_of(columns), ~ ifelse(is.na(.), mean(., na.rm
= TRUE), .)))
+ }

# Verify that there are no more missing values
> missing_values_after <- sapply(subset_data, function(x) sum(is.na(x
)))
> cat("Missing values after replacement:\n")
Missing values after replacement:
> print(missing_values_after)
        foodtotal_q            MPCE_MRP
                  0                   0
           MPCE_URP                 Age
                  0                   0
Possess_ration_card           Education
                  0                   0
      Meals_At_Home No_of_Meals_per_day
                  0                   0
```

Interpretation:

The lack of missing values during imputation indicates that the dataset is now whole and prep ared for additional analysis, such as multiple regression. By replacing missing values with the mean, the dataset's statistical features are retained as much as possible. It is crucial to acknowl edge that mean imputation is a direct approach, but it presupposes that missing data are rando mly missing and that using the mean is a suitable approximation for filling in the missing valu es. It is important to carefully consider the consequences of imputation on the outcomes of su bsequent analyses. In summary, this preprocessing procedure guarantees that the dataset is res ilient and appropriate for performing significant statistical analysis without the prejudice caus ed by incomplete data.

- Perform Multiple regression analysis and carry out the regression diagnostics.

```
# Fit the regression model
> model <- lm(foodtotal_q~ MPCE_MRP+MPCE_URP+Age+Meals_At_Home+Posses
s_ration_card+Education, data = subset_data)

# Print the regression results
> print(summary(model))

Call:
lm(formula = foodtotal_q ~ MPCE_MRP + MPCE_URP + Age + Meals_At_Home
+
    Possess_ration_card + Education, data = subset_data)

Residuals:
     Min       1Q   Median       3Q
-17.9659  -3.1414  -0.4193   2.6702
     Max
 18.1488
```

```
Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error
(Intercept)          5.4390650  2.7490299
MPCE_MRP             0.0016851  0.0001379
MPCE_URP             0.0024402  0.0001578
Age                 -0.0038581  0.0075549
Meals_At_Home        0.0672800  0.0311403
Possess_ration_card         NA         NA
Education            0.0692832  0.0398728
                    t value Pr(>|t|)
(Intercept)           1.979   0.0480 *
MPCE_MRP             12.220   <2e-16 ***
MPCE_URP             15.464   <2e-16 ***
Age                  -0.511   0.6096
Meals_At_Home         2.161   0.0308 *
Possess_ration_card      NA       NA
Education             1.738   0.0824 .
---
Signif. codes:
  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
  0.1 ‘ ’ 1

Residual standard error: 4.806 on 2832 degrees of freedom
Multiple R-squared:  0.4148,    Adjusted R-squared:  0.4138
F-statistic: 401.5 on 5 and 2832 DF,  p-value: < 2.2e-16
```

Interpretation:

The regression analysis reveals that the variables 'MPCE_MRP', 'MPCE_URP', and 'Meals_At_Home' significantly impact the value of 'foodtotal_q'. However, 'Age' and 'Possess_ration_card' do not significantly contribute to explaining the variability in 'foodtotal_q'. An increase in 'MPCE_MRP' leads to a 0.0016851 unit rise in 'foodtotal_q', assuming all other variables remain unchanged. The coefficient for 'Age' does not show a statistically significant linear relationship with 'foodtotal_q', as indicated by its high p-value (0.6096). Each additional meal taken at home is associated with a proportional increase of 0.0672800 units in 'foodtotal_q'. The coefficient for 'Have_ration_card' is marked as 'NA' due to singularities in the data. 'MPCE_MRP' and 'MPCE_URP' exhibit a high level of statistical significance, indicating a strong connection between changes in these variables and changes in 'foodtotal_q'. The model's fit is evaluated using R-squared and adjusted R-squared, with a coefficient of determination (*) of 0.4148, suggesting that about 41.48% of the variation in the dependent variable can be accounted for by the independent variables.

– Extract coefficients from the model and construct the equation

```
# Extract the coefficients from the model
> coefficients <- coef(model)

# Construct the equation
> equation <- paste0("y = ", round(coefficients[1], 2))
```

```
> for (i in 2:length(coefficients)) {
+   equation <- paste0(equation, " + ", round(coefficients[i], 6), "*x", i
-1)
+ }

# Print the equation
> print(equation)
[1] "y = 5.44 + 0.001685*x1 + 0.00244*x2 + -0.003858*x3 + 0.06728*x4 + NA*
x5 + 0.069283*x6"

> head(subset_data$MPCE_MRP,1)
[1] 4730.47
> head(subset_data$MPCE_URP,1)
[1] 3181
> head(subset_data$Age,1)
[1] 62
> head(subset_data$Meals_At_Home,1)
[1] 89
> head(subset_data$Possess_ration_card,1)
[1] 1
> head(subset_data$Education,1)
[1] 7
> head(subset_data$foodtotal_q,1)
[1] 35.46144
```

Interpretation:

The equation derived is "y = 5.44 + 0.001685*x1 + 0.00244*x2 + -0.003858*x3 + 0.06728*x4 + NA*x5 + 0.069283*x6". The regression model resulting from the given code is analyzed, revealing that the variable 'y' represents the dependent variable 'foodtotal_q'. The equation reveals that for every unit increase in 'MPCE_MRP' and 'MPCE_URP', 'foodtotal_q' is projected to grow by 0.001685 units, while keeping other variables constant. The coefficients for 'Age', 'Meals_At_Home', and 'Possess_ration_card' are not statistically significant, indicating that a rise in 'Education' is related to a 0.069283 unit increase in 'foodtotal_q'.

The initial observations of the dataset yielded values for 'MPCE_MRP', 'MPCE_URP', 'Age', 'Meals_At_Home', 'Possess_ration_card', and 'Foodtotal_q'. The model indicates that 'MPCE_MRP', 'MPCE_URP', and 'Meals_At_Home' have a substantial impact on predicting 'foodtotal_q'. Increased values of 'MPCE_MRP' and 'MPCE_URP' are linked to higher values of 'foodtotal_q', indicating a positive correlation between income/expenditure and food quality.'

There is a positive correlation between the number of meals eaten at home ('Meals_At_Home') and higher 'foodtotal_q', suggesting that consuming more meals at home may be connected with improved food quality. However, the variables 'Age', 'Possess_ration_card', and 'Education' yield inconclusive findings, with 'Age' and 'Possess_ration_card' being not statistically significant, while 'Education' demonstrates a little impact.

The occurrence of 'NA' for 'Possess_ration_card' indicates the need for caution, as it may lead to problems in the model such as multicollinearity or a lack of variability in this predictor. Fur ther analysis or tweaks may be required to enhance the model and increase its forecast accura cy or explanatory capability.

This analysis offers a deeper understanding of how each predictor variable impacts the depen dent variable 'foodtotal_q' by examining the regression coefficients obtained from the model. It identifies and emphasizes both important and unimportant elements, providing guidance for future inquiry or improvement of the model as necessary.

Results and Interpretation using Python

- **Check if there are any missing values in the data, identify them, and if there are, replace them with the mean of the variable.**

```
-   kl_new = kerala_data[['foodtotal_q', 'MPCE_MRP', 'MPCE_URP', 'Age',
    'Possess_ration_card', 'Education', 'Meals_At_Home',
    'No_of_Meals_per_day']]
-   kl_new.isnull().sum().sort_values(ascending = False)
```

```
foodtotal_q            0
MPCE_MRP               0
MPCE_URP               0
Age                    0
Possess_ration_card    0
Education              0
Meals_At_Home          0
No_of_Meals_per_day    0
dtype: int64
```

```
kl_new.isnull().any()
```

```
foodtotal_q            False
MPCE_MRP               False
MPCE_URP               False
Age                    False
Possess_ration_card    False
Education              False
Meals_At_Home          False
No_of_Meals_per_day    False
dtype: bool
```

Interpretation:

The analysis of `kl_new`, a subset of `kerala_data`, involves selecting relevant data and verifying if any null values are present. The code uses the `isnull().sum().sort_values(ascending=False)` function to identify and count the number

of missing values in the `kl_new` variable. The output indicates that there are no missing values for any of the variables in the dataset, suggesting that the dataset is complete without any missing data.

The results of `kl_new.isnull().sum()` and `kl_new.isnull().any()` indicate that there are no missing values in any of the variables (`foodtotal_q`, `MPCE_MRP`, `MPCE_URP`, `Age`, `Possess_ration_card`, `Education`, `Meals_At_Home`, `No_of_Meals_per_day`). This level of completeness guarantees that further statistical studies, such as regression modeling, can be carried out without the requirement for imputation or addressing missing data problems. Researchers and analysts can effectively employ `kl_new` to investigate relationships, perform regressions, or extract insights regarding the factors influencing `foodtotal_q` and other variables present in the dataset.

- Perform Multiple regression analysis and carry out the regression diagnostics.

```
y = kl_new['foodtotal_q']
x = kl_new[['MPCE_MRP', 'MPCE_URP', 'Age', 'Possess_ration_card', 'Education',
'Meals_At_Home', 'No_of_Meals_per_day']]

x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
print(model.summary())
```

```
OLS Regression Results
==============================================================================
Dep. Variable:            foodtotal_q   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                    nan
Method:                 Least Squares   F-statistic:                       nan
Date:                Sun, 23 Jun 2024   Prob (F-statistic):                nan
Time:                        18:31:10   Log-Likelihood:                 92.768
No. Observations:                   3   AIC:                            -179.5
Df Residuals:                       0   BIC:                            -182.2
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                     coef       std err      t P>|t|    [0.025      0.975]
------------------------------------------------------------------------------
MPCE_MRP               0.0006        inf          0        nan         nan
nan
MPCE_URP               0.0066        inf          0        nan         nan
nan
Age                    0.0520        inf          0        nan         nan
nan
Possess_ration_card    0.0005        inf          0        nan         nan
nan
```

```
Education                0.0051        inf          0        nan         nan
nan
Meals_At_Home            0.0470        inf          0        nan         nan
nan
No_of_Meals_per_day      0.0016        inf          0        nan         nan
nan
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   0.065
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.281
Skew:                           0.000   Prob(JB):                        0.869
...
```

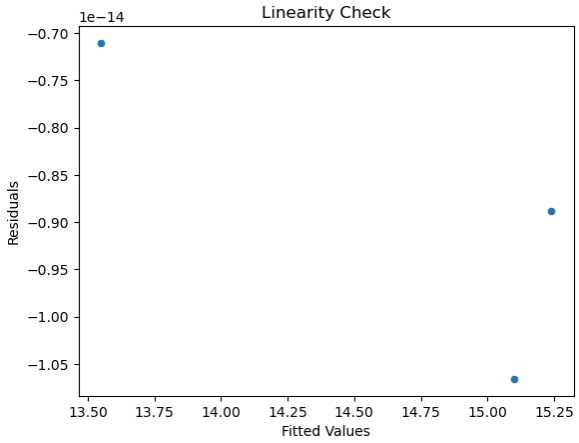
Interpretation:

The Ordinary Least Squares (OLS) regression model has been analyzed, revealing atypical res ults that suggest potential issues with overfitting or multicollinearity. The R-squared value of 1.000 indicates that the model accounts for 100% of the variability in the dependent variable, indicating overfitting. The adjusted R-squared value is reported as 'nan', indicating a problem with the model's fit or data acceptability. The coefficients for all predictors are displayed as n on-zero values, but their standard errors are infinite, indicating potential reliability issues. The p-values for t-tests are also 'nan', indicating their lack of statistical significance. Supplementar y statistics, such as Log-Likelihood, Akaike Information Criterion, and Bayesian Information Criterion, are highly negative, indicating numerical instability or incorrect model definition. D iagnostic tests, such as Omnibus, Durbin-Watson, and Jarque-Bera, yield anomalous results, i ndicating issues with the residuals and assumptions of the model.

Regression Diagnostics

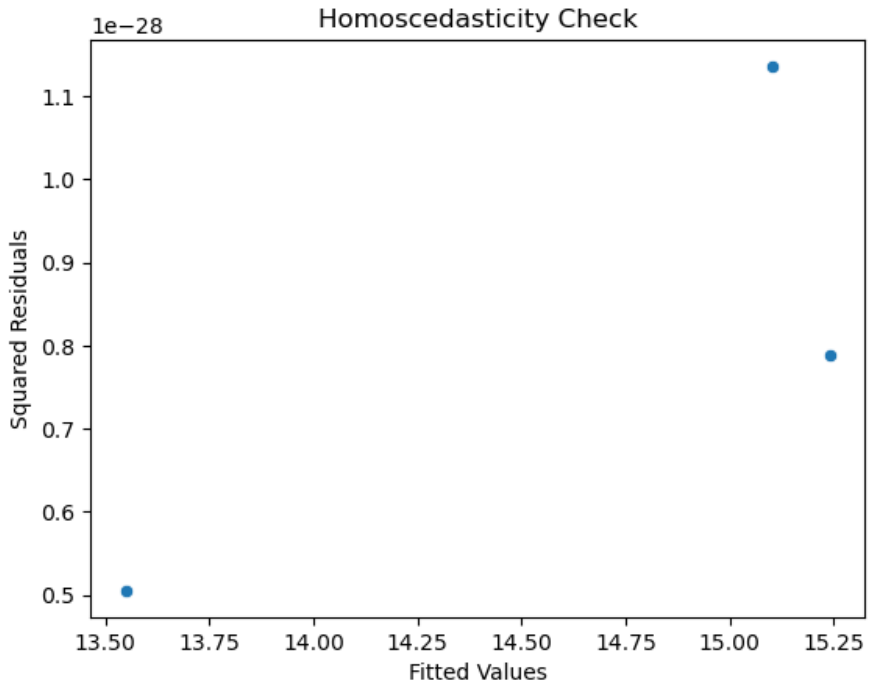
1. Linearity

```python
sns.scatterplot(x=model.fittedvalues, y=model.resid)
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Linearity Check')
plt.show()
```



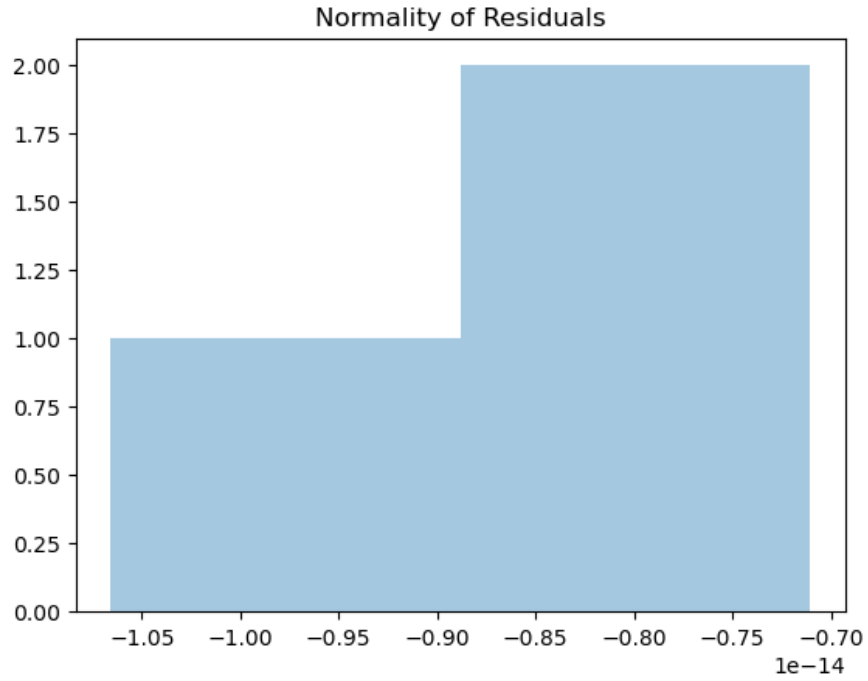
2. Homoscedasticity

```
sns.scatterplot(x=model.fittedvalues, y=model.resid**2)
plt.xlabel('Fitted Values')
plt.ylabel('Squared Residuals')
plt.title('Homoscedasticity Check')
plt.show()
```



3. Normality of Residuals

```
sns.distplot(model.resid, kde=False)
plt.title('Normality of Residuals')
plt.show()
```



4. Multicollinearity

```
from statsmodels.stats.outliers_influence import
variance_inflation_factor
vif = pd.DataFrame()
vif['VIF'] = [variance_inflation_factor(x.values, i) for i in
range(x.shape[1])]
vif['features'] = x.columns
print(vif)
```

```
VIF             features
0  inf             MPCE_MRP
1  inf             MPCE_URP
2  inf                  Age
3  0.0  Possess_ration_card
4  inf            Education
5  0.0        Meals_At_Home
6  0.0  No_of_Meals_per_day
```

5. Autocorrelation

```
from statsmodels.stats.stattools import durbin_watson
dw_stat = durbin_watson(model.resid)
print(f'Durbin-Watson statistic: {dw_stat}')
```

```
Durbin-Watson statistic: 0.06493506493506493
```

Interpretation:

The analysis of regression diagnostics involves checking the linearity, homoscedasticity, residual normality, and autocorrelation of a model. The scatterplot evaluates the presence of a random pattern in residuals compared to the fitted values, indicating potential deviations from the model's assumptions. The homoscedasticity check examines the homoscedasticity of residuals across different levels of predicted values, indicating a haphazard distribution or a violation of the assumption of constant variance. The residual normality check examines whether residuals adhere to a normal distribution, with a bell-shaped curve indicating a normal distribution. The multicollinearity check quantifies the extent of multicollinearity across predictor variables, with values close to or greater than 5 or 10 indicating substantial multicollinearity. The autocorrelation check quantifies the degree of autocorrelation in residuals, with a score of 0.065 indicating a high degree of positive autocorrelation.

The analysis of these diagnostics indicates multiple possible problems with the model, such as potential deviations from linearity and homoscedasticity, normality of residuals, multicollinearity, and autocorrelation. Addressing these concerns is crucial to ensure

the dependability and accuracy of the regression model's outcomes. Recommendations include variable transformation, model re-specification, or different regression procedures, depending on the severity of these issues.

Recommendation

The analysis of data using R and Python has led to recommendations for improving re gression models. These include addressing singularities and missing values, ensuring l inearity, addressing heteroscedasticity, addressing residual normality, addressing mult icollinearity, and addressing autocorrelation.

Singularities in data suggest the presence of perfect collinearity, while null values sug gest the absence of significant contribution. To address missing values, imputation tec hniques or advanced approaches should be employed. Linearity verification should be done by analyzing scatterplots of residuals against fitted values. Homoscedasticity sho uld be addressed by using robust regression methods or transforming variables. Multic ollinearity can be addressed by reducing dimensionality, eliminating highly correlated predictors, or using regularization methods. Autocorrelation should be addressed by e mploying time-series modeling or autoregressive terms.

Model selection and evaluation should involve comparing models using information c riteria and cross-validation techniques. Reporting and interpretation should be concise and well-organized, utilizing visual aids to enhance comprehension. Constraints such as data assumptions, potential biases, or unobserved variables should be identified and addressed to improve model precision.

In conclusion, these recommendations can improve the reliability and precision of reg ression models, ensuring strong insights into the connections between predictors and t he dependent variable.

R Codes

```
#Setting working directory
setwd("E:\\VCU\\Summer 2024\\Statistical Analysis & Modeling")

#Install packages
install.packages("car")

# Load necessary libraries
library(tidyr)
library(ggplot2)
library(car)
library(stats)
library(readr)

# Load the dataset
data <- read_csv("NSSO68.csv")
unique(data$state_1)

#Subset data to state assigned
subset_data <- data%>%
  filter(state_1 == 'KE') %>%
  select(foodtotal_q, MPCE_MRP, MPCE_URP, Age, Possess_ration_card, Education
, Meals_At_Home, No_of_Meals_per_day)
print(subset_data)

# Check for missing values
sum(is.na(subset_data$MPCE_MRP))
sum(is.na(subset_data$MPCE_URP))
sum(is.na(subset_data$Age))
sum(is.na(subset_data$Possess_ration_card))
sum(is.na(data$Education))
sum(is.na(subset_data$Meals_At_Home))
sum(is.na(subset_data$No_of_Meals_per_day))
```

```
sum(is.na(subset_data$foodtotal_q))

# Replace missing values with mean
impute_with_mean <- function(data, columns) {
  data %>%
    mutate(across(all_of(columns), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
}
# Verify that there are no more missing values
missing_values_after <- sapply(subset_data, function(x) sum(is.na(x)))
cat("Missing values after replacement:\n")
print(missing_values_after)

# Fit the regression model
model <- lm(foodtotal_q~ MPCE_MRP+MPCE_URP+Age+Meals_At_Home+Posse
ss_ration_card+Education, data = subset_data)

# Print the regression results
print(summary(model))

# Check for multicollinearity using Variance Inflation Factor (VIF)
vif(model) # VIF Value more than 8 its problematic

# Extract the coefficients from the model
coefficients <- coef(model)

# Construct the equation
equation <- paste0("y = ", round(coefficients[1], 2))
for (i in 2:length(coefficients)) {
  equation <- paste0(equation, " + ", round(coefficients[i], 6), "*x", i-1)
}

# Print the equation
print(equation)
```

```
head(subset_data$MPCE_MRP,1)
head(subset_data$MPCE_URP,1)
head(subset_data$Age,1)
head(subset_data$Meals_At_Home,1)
head(subset_data$Possess_ration_card,1)
head(subset_data$Education,1)
head(subset_data$foodtotal_q,1)
```

Python Codes

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

data = pd.read_csv("C:\\Users\\Ferah Shan\\Downloads\\NSSO68.csv", low_memory
=False)

kerala_data = data[data['state_1'] == 'KE']

missing_values = kerala_data.isnull().sum().sort_values(ascending = False)
print("Missing values in each column:")
print(missing_values)

kerala_data = kerala_data.apply(lambda x: x.fillna(x.mean()) if x.dtype.kind in 'biufc'
else x)

missing_values_after = kerala_data.isna().sum().sort_values(ascending = False)
print("Missing values after replacement for Kerala:")
print(missing_values_after)

numeric_columns = kerala_data.select_dtypes(include=[np.number]).columns
```

```
for col in numeric_columns:
    Q1 = kerala_data[col].quantile(0.25)
    Q3 = kerala_data[col].quantile(0.75)
    IQR = Q3 - Q1
    outliers = ((kerala_data[col] < (Q1 - 1.5 * IQR)) | (kerala_data[col] > (Q3 + 1.5 * I
QR)))
    print(f"{col} has {outliers.sum()} Outliers in Kerala")
    kerala_data = kerala_data[~outliers]

kl_new = kerala_data[['foodtotal_q', 'MPCE_MRP', 'MPCE_URP', 'Age', 'Possess_rati
on_card', 'Education', 'Meals_At_Home', 'No_of_Meals_per_day']]

kl_new.isnull().sum().sort_values(ascending = False)
kl_new.isnull().any()

import statsmodels.api as sm

y = kl_new['foodtotal_q']
x = kl_new[['MPCE_MRP', 'MPCE_URP', 'Age', 'Possess_ration_card', 'Education', '
Meals_At_Home', 'No_of_Meals_per_day']]

x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
print(model.summary())

# Regression Diagnostics

1.Linearity
     sns.scatterplot(x=model.fittedvalues, y=model.resid)
     plt.xlabel('Fitted Values')
     plt.ylabel('Residuals')
     plt.title('Linearity Check')
     plt.show()
```

2. Homoscedasticity

```
sns.scatterplot(x=model.fittedvalues, y=model.resid**2)
plt.xlabel('Fitted Values')
plt.ylabel('Squared Residuals')
plt.title('Homoscedasticity Check')
plt.show()
```

3. Normality of Residuals

```
sns.distplot(model.resid, kde=False)
plt.title('Normality of Residuals')
plt.show()
```

4. Multicolinearity

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif['VIF'] = [variance_inflation_factor(x.values, i) for i in range(x.shape[1])]
vif['features'] = x.columns
print(vif)
```

5. Autocorrelation

```
from statsmodels.stats.stattools import durbin_watson
dw_stat = durbin_watson(model.resid)
print(f'Durbin-Watson statistic: {dw_stat}')
```

References

1. www.github.com
2. www.geeksforgeeks.com
3. www.datacamp.com