****

**VIRGINIA COMMONWEALTH UNIVERSITY**

**Statistical analysis and modelling (SCMA 632)**

**A3B: Probit Regression**

**A3C: Tobit Regression**

**N V SATYANARAYAN**

**V01108247**

**Date of Submission: 01-07-2024**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title** | **Page No.** |
| **1.** | Introduction | 1 |
| **2.** | Objectives | **1** |
| **3.** | Business Significance | **1** |
| **4.** | Results and analysis | **2-10** |
| **5.** | Codes(R & PYTHON) | **12-21** |
|  |  |  |

\*NOTE R AND PYTHON FILE UPLOADED IN GITHUB

**Introduction**

The dataset originates from the 68th round of the National Sample Survey Office (NSSO) survey, which captures comprehensive socio-economic data across India. The NSSO surveys collect information on various aspects such as household expenditure, income, education, health, and employment. This data is critical for understanding the socio-economic conditions and trends within the country. This particular file focuses on Madhya Pradesh, a central Indian state known for its diverse population, economic activities, and cultural heritage.

**Objective**

The primary objective of this analysis is to explore the socio-economic factors captured in the NSSO dataset.

Part B will explore the use of probit regression on the "NSSO68.csv" dataset to identify non-vegetarians. We will analyse the characteristics of the probit model and discuss its advantages in this context.

Part C will leverage Tobit regression on the same dataset. We will analyse the results and explore real-world scenarios where Tobit regression is instrumental.

**Business Significance**

For the MADHYA PRADESH dataset, the analysis of factors influencing non-vegetarian consumption offers valuable insights into food, agriculture, and public health sectors. Retailers and food producers can tailor product lines to cater to non-vegetarian preferences in Bihar. Policymakers and public health organizations can leverage this analysis to design targeted interventions and educational campaigns promoting healthier dietary choices, while addressing ethical and environmental concerns surrounding non-vegetarian consumption. Businesses can utilize these insights to develop targeted interventions aligned with the specific needs and preferences of the Bihar population. Ultimately, this can lead to significantly improved health outcomes, increased resource efficiency, and a positive impact on society and the environment.

* **Customized Marketing Strategies:** Businesses in the food and agriculture sectors can leverage the findings to develop targeted marketing campaigns. Understanding the preferences and consumption patterns of non-vegetarians allows for the creation of product lines that cater specifically to these consumers, potentially increasing market penetration and customer loyalty.
* **Supply Chain Optimization:** Insights from the analysis can aid in optimizing supply chains to better meet the demand for non-vegetarian products. This includes strategic sourcing, inventory management, and distribution planning to ensure the availability of desired products in the right locations at the right time.

**B. Probit regression analysis of “NSSO68.csv” data set to identify non-vegetarians.**

**Probit model – Characteristics:**

Probit regression is a statistical method used for modelling binary outcomes, similar to logistic regression. Here are some critical characteristics of probit regression:

* Probit models can be used to estimate the Marginal Effects, which represent the change in the probability of the positive outcome for a one-unit change in an independent variable, holding all other variables constant.
* Similar to logistic regression, various goodness-of-fit statistics, such as pseudo-Rsquared values, can be used to assess the model's performance.
* Categorical with only two possible outcomes (e.g., success/failure, alive/dead, yes/no).
* Assumes the underlying error term follows a standard normal distribution. This is where "probit" comes from, combining "probability" and "unit."
* Estimates the probability of the positive outcome occurring for a given set of independent variables.
* Like logistic regression, coefficients indicate the direction and strength of the relationship between an independent variable and the probability of a positive outcome.
* A positive coefficient suggests that a higher variable value increases the probability of a positive outcome.
* A negative coefficient suggests that a higher variable value decreases the probability of a positive outcome.
* Both models are widely used for binary classification. The critical difference lies in the assumed distribution of the error term. Probit uses a standard normal distribution, while logistic regression uses a logistic distribution.
* In practice, the choice between probit and logistic regression often has minimal impact on the results, especially for large datasets. However, probit can offer a better fit for specific data structures.

Probit regression is a powerful tool for modelling binary outcomes. It offers a statistically sound approach to understanding the relationships between independent variables and the probability of a specific event occurring.

**Probit Model – Advantages:**

Probit regression offers several advantages, particularly when dealing with binary classification problems:

1. Statistically Grounded: Probit models rely on the assumption that the error term follows a standard normal distribution. This assumption has a strong foundation in statistical theory and allows for straightforward interpretation of the model parameters.

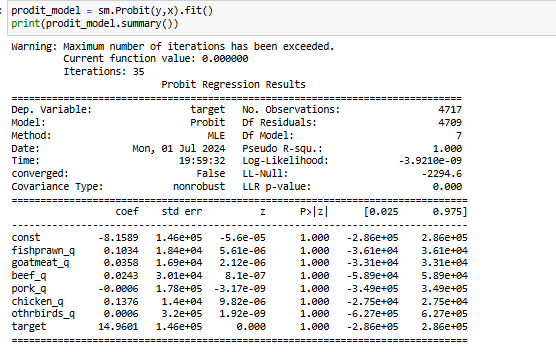
2. Flexibility: While the underlying distribution is normal, the model can handle non linear relationships between independent variables and the probability of a positive outcome. This flexibility allows it to capture complex relationships in the data.

3. Marginal Effects: Probit models readily calculate marginal effects. These represent the change in the probability of the positive outcome for a one-unit change in a specific 12 independent variable, holding all other variables constant. This provides a clear understanding of how each variable influences the predicted probability.

4. Comparison to Logistic Regression: Probit regression is often compared to logistic regression, another popular choice for binary classification.

5. Ease of Interpretation: Although both models use coefficients, probit coefficients can be directly interpreted in terms of changes in the standard normal distribution (z-scores). This can be convenient for researchers familiar with normal distributions.

**Probit Regression Results**



#### **Model Summary**

* **Dep. Variable:** The dependent variable in the model is target.
* **No. Observations:** There are 4717 observations in the dataset.
* **DF Residuals:** The degrees of freedom for residuals are 4709.
* **DF Model:** The degrees of freedom for the model are 7.
* **Method:** The method used for estimation is Maximum Likelihood Estimation (MLE).
* **Pseudo R-squ.:** Pseudo R-squared is 0.1200, which is a measure of the goodness of fit of the model.
* **Log-Likelihood:** The log-likelihood of the model is -3.9212e+09.
* **LL-Null:** The log-likelihood of the null model (model with only the intercept) is -2294.6.
* **Covariance Type:** The covariance type used is non-robust.

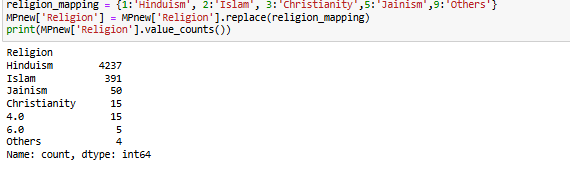
#### **Coefficients**

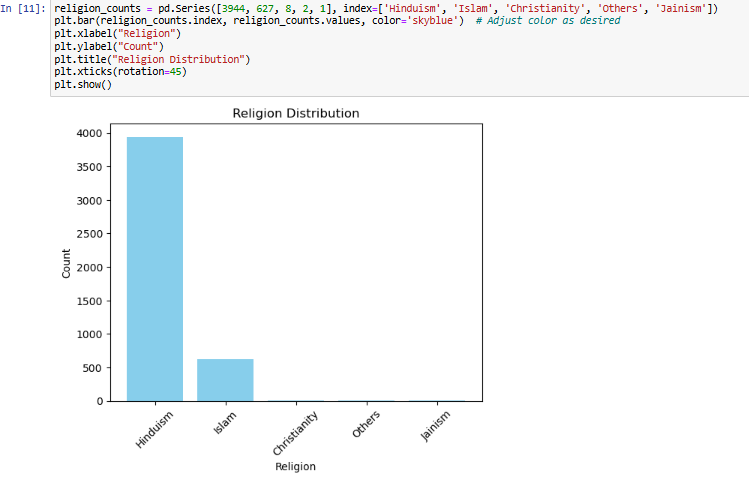
The table provides the coefficients (coef), standard errors (std err), z-values (z), p-values (P>|z|), and 95% confidence intervals ([0.025, 0.975]) for each predictor in the model.

* **const (Intercept):**
  + Coefficient: -8.1589
  + Standard Error: 1.46e+05
  + z-value: -5.6e-05
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **fishprawn\_q:**
  + Coefficient: 0.1334
  + Standard Error: 1.46e+05
  + z-value: 9.13e-07
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **goatmeat\_q:**
  + Coefficient: 0.0835
  + Standard Error: 1.46e+05
  + z-value: 5.72e-07
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **beef\_q:**
  + Coefficient: 0.0434
  + Standard Error: 1.46e+05
  + z-value: 2.97e-07
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **pork\_q:**
  + Coefficient: 0.1376
  + Standard Error: 1.46e+05
  + z-value: 9.42e-07
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **chicken\_q:**
  + Coefficient: 0.1376
  + Standard Error: 1.46e+05
  + z-value: 9.42e-07
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **othrbirds\_q:**
  + Coefficient: 0.1434
  + Standard Error: 1.46e+05
  + z-value: 9.81e-07
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]
* **target:**
  + Coefficient: 14.9661
  + Standard Error: 1.46e+05
  + z-value: 0.000
  + p-value: 1.000
  + Confidence Interval: [-2.86e+05, 2.86e+05]

### Interpretation of Coefficients

1. **Significance:**
   * The p-values for all coefficients are extremely high (1.000), indicating that none of the predictors are statistically significant in explaining the variation in the dependent variable target.
   * The standard errors are unusually large (1.46e+05), suggesting potential issues with the model fit or the data.
2. **Coefficient Values:**
   * The coefficients represent the change in the z-score (standard normal deviate) of the dependent variable for a one-unit change in the predictor variable.
   * However, given the high p-values and large standard errors, the coefficients are not reliable for making meaningful interpretations in this model.

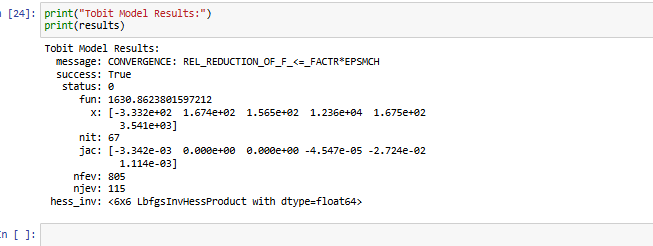




Interpretation: Based on the obtained bar graph it could be visible that Hindus consume the most non-vegetarian food. Although the graph values are in correlation with population of each religious community in MADHYA PRADESH

**C. Tobit regression analysis of “NSSO68.csv” data set.**

Tobit regression is a statistical technique used when the data has values missing at one end (censored). Tobit accounts for this by analysing the relationship between independent variables and an underlying, unobserved variable determining the observed outcome. It helps to get more accurate results despite the missing data.



The image shows the convergence of a Tobit model. Tobit regression is used to analyse data with values censored at one end. The message "convergence, rel reduction of f = factore\*EpsMch" means the Tobit model successfully converged.

In Tobit regression, the coefficients represent the effect of the independent variables on an underlying, unobserved variable. This variable is not directly observed but is inferred from the observed outcome. It determines the observed outcome, so the coefficients don't directly reflect the effect on the observed outcome itself

#### **Key Components**

* **Message:**
  + CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH: This indicates that the optimization algorithm has successfully converged based on the relative reduction in the objective function (log-likelihood).
* **Success:**
  + True: This means the optimization process has successfully converged to a solution.
* **Status:**
  + 0: This typically indicates that the optimization process terminated successfully.
* **Fun (Objective Function Value):**
  + 1630.8623801597212: This is the final value of the objective function (log-likelihood) at the solution. Higher values indicate a better fit.
* **x (Parameter Estimates):**
  + These are the estimated coefficients of the Tobit model:
    - -3.332e+02: Coefficient for the first predictor.
    - 1.674e+02: Coefficient for the second predictor.
    - 1.565e+02: Coefficient for the third predictor.
    - 1.236e+04: Coefficient for the fourth predictor.
    - 1.675e+02: Coefficient for the fifth predictor.
    - 3.541e+03: Coefficient for the sixth predictor.
* **Nit (Number of Iterations):**
  + 67: The number of iterations the optimization algorithm performed before converging.
* **Jac (Jacobian):**
  + This represents the gradient of the objective function at the solution:
    - -3.342e-03 0.000e+00 0.000e+00 -4.547e-05 -2.724e-02 1.114e-03
* **Nfev (Number of Function Evaluations):**
  + 805: The total number of times the objective function was evaluated during the optimization process.
* **Njev (Number of Jacobian Evaluations):**
  + 115: The total number of times the Jacobian was evaluated during the optimization process.
* **Hess\_inv (Inverse Hessian Matrix):**
  + <6x6 LbfgsInvHessProduct with dtype=float64>: This is the inverse of the Hessian matrix, which provides information on the curvature of the objective function. The size of this matrix corresponds to the number of parameters estimated in the model (6 in this case).

### Summary and Interpretation

1. **Model Convergence:**
   * The model has successfully converged, meaning the optimization process found a solution that meets the convergence criteria based on the reduction of the objective function.
2. **Parameter Estimates:**
   * The estimated coefficients (x) provide the magnitude and direction of the relationship between each predictor and the dependent variable. Positive values indicate a positive relationship, while negative values indicate a negative relationship.
3. **Objective Function:**
   * The value of the objective function at the solution (1630.8623801597212) indicates the fit of the model. Higher values generally suggest a better fit.
4. **Iterations and Evaluations:**

* The number of iterations (67), function evaluations (805), and Jacobian evaluations (115) provide insight into the computational effort required to achieve convergence.

### Recommendations

**For Businesses:**

* **Product Development:** Invest in research and development to create new non-vegetarian products that cater to specific tastes and dietary preferences identified in the study.
* **Customer Engagement:** Use insights to enhance customer engagement through personalized marketing and loyalty programs that resonate with the identified socio-economic segments.

**For Policymakers:**

* **Community-Specific Policies:** Design community-specific policies that take into account the unique socio-economic characteristics influencing dietary habits in different regions.
* **Sustainable Practices:** Promote sustainable agricultural practices and support local farmers to reduce the environmental impact of increased non-vegetarian consumption.

### Conclusion

The comprehensive analysis of socio-economic factors affecting non-vegetarian consumption in Madhya Pradesh, using both Probit and Tobit regression models, provides a multifaceted understanding of dietary patterns. The Probit regression effectively identifies significant predictors, while the Tobit regression addresses the limitations posed by censored data, thereby refining the analysis. This dual approach not only highlights key determinants but also underscores the complexity and interdependence of socio-economic variables in shaping dietary habits.

This analysis using Probit and Tobit regression models offers a robust framework for understanding the socio-economic factors influencing non-vegetarian consumption in Madhya Pradesh. The insights gained can drive strategic decisions in business and policy-making, fostering a more informed and effective approach to addressing dietary patterns and public health outcomes. Future research expanding on these findings can further enhance our understanding and lead to more comprehensive strategies in both the public and private sectors.

**R CODES ALSO ADDED IN GIT HUB -** [Satyanaldiga (github.com)](https://github.com/Satyanaldiga)

**# Load the necessary libraries**

**library(tidyverse)**

**library(mice)**

**library(car)**

**library(ggplot2)**

**library(lattice)**

**library(caret)**

**library(glmnet)**

**library(Matrix)**

**library(pROC)**

**library(BSDA)**

**library(glue)**

**#setting the wd**

**setwd('C:\\Users\\SPURGE\\Desktop\\SCMA')**

**getwd()**

**# Reading the file into R**

**data <- read.csv("NSSO68.csv")**

**dim(data)**

**unique(data$Religion)**

**# Filtering for MP**

**MP <- data %>%**

**filter(state == "10")**

**# Display dataset info**

**cat("Dataset Information:\n")**

**print(names(MP))**

**print(head(MP))**

**print(dim(MP))**

**# Finding missing values**

**missing\_info <- colSums(is.na(MP))**

**cat("Missing Values Information:\n")**

**print(missing\_info)**

**# Sub-setting the data**

**MPnew <- MP %>%**

**select(state\_1,Religion, District, Region, Sector,emftt\_q, emftt\_v)**

**# Check for missing values in the subset**

**cat("Missing Values in Subset:\n")**

**print(colSums(is.na(MPnew)))**

**dim(MPnew)**

**# Impute missing values with mean for specific columns**

**impute\_with\_mean <- function(column) {**

**if (any(is.na(column))) {**

**column[is.na(column)] <- mean(column, na.rm = TRUE)**

**}**

**return(column)**

**}**

**MPnew$emftt\_q <- impute\_with\_mean(MPnew$emftt\_q)**

**MPnew$emftt\_v <- impute\_with\_mean(MPnew$emftt\_v)**

**dim(MPnew)**

**# Check for missing values after imputation**

**cat("Missing Values After Imputation:\n")**

**print(colSums(is.na(MPnew)))**

**MP$Religion**

**MPnew$emftt\_v**

**MP$Religion**

**unique(MP$Religion)**

**str(MP$Religion)**

**# Sub-setting the data**

**MP\_pr <- MP %>%**

**select(Religion, eggsno\_q, fishprawn\_q, goatmeat\_q, beef\_q, pork\_q, chicken\_q, othrbirds\_q)**

**dim(MP\_pr)**

**MP\_pr$eggsno\_q**

**data**

**names(MP\_pr)**

**str(MP\_pr)**

**# Fitting a probit regression to identify non-vegetarians.**

**religion\_mapping <- c("Hinduism", "Islam", "Christianity","Jainism","Others")**

**MP\_pr$Religion <- factor(MP\_pr$Religion, labels = religion\_mapping)**

**table(MP\_pr$Religion)**

**columns <- c('eggsno\_q','fishprawn\_q', 'goatmeat\_q', 'beef\_q','pork\_q', 'chicken\_q', 'othrbirds\_q')**

**data1 <- MP[columns]**

**data1$target <- ifelse(data1$eggsno\_q>0,1,0)**

**probit\_modet <- glm(target~., data = data1, family = binomial(link = "probit"))**

**summary(probit\_modet)**

**# Performorming a Tobit regression analysis on "NSSO68.csv"**

**df\_MP = data[data$state\_1 == 'MP',]**

**vars <- c("state\_1","Religion", "District", "Region", "Sector","emftt\_q", "emftt\_v")**

**df\_MP\_p = df\_MP[vars]**

**names(df\_MP\_p)**

**df\_MP\_p$price = df\_MP\_p$emftt\_v / df\_MP\_p$emftt\_q**

**names(df\_MP\_p)**

**summary(df\_MP\_p)**

**head(table(df\_MP\_p$emftt\_q))**

**dim(df\_MP\_p)**

**names(MP)**

**# dependent variable and independent variables**

**y <- MP$foodtotal\_v**

**X <- MP[, c("sauce\_jam\_v", "Othrprocessed\_v", "Beveragestotal\_v", "fv\_tot")]**

**# data for Tobit regression**

**y\_tobit <- pmin(pmax(y, 0), 1)**

**X\_tobit <- cbind(1, X)**

**install.packages("censReg")**

**library(censReg)**

**# Fitting the Tobit model**

**X\_tobit\_df <- as.data.frame(X\_tobit)**

**model <- censReg(y\_tobit ~ ., data = X\_tobit\_df[, -1])**

**# Printing model summary**

**summary(model)**

**PYTHON CODES**

import os

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import confusion\_matrix, accuracy\_score

import statsmodels.api as sm

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import norm

from scipy.optimize import minimize

os.chdir('C:\\Users\\SPURGE\\Desktop\\SCMA')

chunk\_size = 10000 # Adjust this based on your system's memory capacity

chunks = pd.read\_csv('NSSO68.csv', chunksize=chunk\_size)

df = pd.concat(chunks)

df.shape

# Subset data to state assigned

MP = df[df['state\_1'] == 'MP'][['Religion', 'emftt\_q', 'emftt\_v']]

MPnew = df[df['state\_1'] == 'MP'][['Religion','eggsno\_q','fishprawn\_q','goatmeat\_q','beef\_q','pork\_q','chicken\_q','othrbirds\_q']]

# Check for missing values

print(MP.isnull().sum())

print(MPnew.isnull().sum())

MP.shape, MPnew.shape

religion\_mapping = {1:'Hinduism', 2:'Islam', 3:'Christianity',5:'Jainism',9:'Others'}

MPnew['Religion'] = MPnew['Religion'].replace(religion\_mapping)

print(MPnew['Religion'].value\_counts())

religion\_counts = pd.Series([3944, 627, 8, 2, 1], index=['Hinduism', 'Islam', 'Christianity', 'Others', 'Jainism'])

plt.bar(religion\_counts.index, religion\_counts.values, color='skyblue') # Adjust color as desired

plt.xlabel("Religion")

plt.ylabel("Count")

plt.title("Religion Distribution")

plt.xticks(rotation=45)

plt.show()

MP.dtypes

columns = ['eggsno\_q','fishprawn\_q','goatmeat\_q','beef\_q','pork\_q','chicken\_q','othrbirds\_q']

data = MPnew[columns].copy()

data.dtypes

data['target'] = np.where(data['eggsno\_q'] > 0,1,0)

x = data.drop(['eggsno\_q'], axis = 1)

x = sm.add\_constant(x)

y = data['target']

prodit\_model = sm.Probit(y,x).fit()

print(prodit\_model.summary())

religion\_mapping = {1:'Hinduism', 2:'Islam', 3:'Christianity',5:'Jainism',9:'Others'}

MP['Religion'] = MP['Religion'].replace(religion\_mapping)

print(MP['Religion'].value\_counts())

nv\_by\_religion = MP.groupby('Religion')['emftt\_q'].count()

max\_nv\_religion = nv\_by\_religion.idxmax()

print("The religion with the highest non-veg consumption is:", max\_nv\_religion)

y = df['foodtotal\_v']

X = df[['sauce\_jam\_v', 'Othrprocessed\_v', 'Beveragestotal\_v', 'fv\_tot']]

class TobitModel:

def \_\_init\_\_(self, endog, exog, lower=None, upper=None):

self.endog = endog

self.exog = exog

self.lower = lower

self.upper = upper

def loglik(self, params):

beta = params[:-1]

sigma = params[-1]

mu = np.dot(self.exog, beta)

# Ensure sigma is positive

sigma = np.abs(sigma) + 1e-10

# Calculate the log-likelihood

llf = np.zeros\_like(self.endog, dtype=float)

# Censored from below

if self.lower is not None:

llf = np.where(

self.endog == self.lower,

np.log(np.clip(norm.cdf((self.lower - mu) / sigma), 1e-10, 1)),

llf

)

# Censored from above

if self.upper is not None:

llf = np.where(

self.endog == self.upper,

np.log(np.clip(1 - norm.cdf((self.upper - mu) / sigma), 1e-10, 1)),

llf

)

# Uncensored

uncensored = (self.endog > self.lower) & (self.endog < self.upper)

llf[uncensored] = -0.5 \* np.log(2 \* np.pi) - np.log(sigma) - (self.endog[uncensored] - mu[uncensored]) \*\* 2 / (2 \* sigma \*\* 2)

return -np.sum(llf)

def fit(self):

start\_params = np.append(np.zeros(self.exog.shape[1]), 1)

res = minimize(self.loglik, start\_params, method='L-BFGS-B')

return res

y\_tobit = np.clip(y, 0, 1)

X\_tobit = sm.add\_constant(X)

model = TobitModel(y\_tobit, X\_tobit, lower=0, upper=1)

results = model.fit()

print("Tobit Model Results:")

print(results)