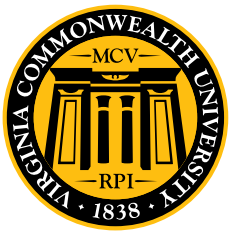


VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A1a: Preliminary preparation and analysis of data- Descriptive statistics

PRAMITT M PATIL

V01104754

Date of Submission: 16-06-2024

CONTENTS

Analysing Consumption in the State of Tamil Nadu Using R & Python

1. Introduction

This study focuses on the state of Tamil Nadu to identify the top and bottom three districts in terms of consumption based on NSSO data. We clean and alter the dataset during the process to obtain the necessary data for analysis. Make this research easier, we have assembled a consumption-related dataset, including information on the rural and urban sectors and district-level variances. The statistical programming language R is known for its versatility in managing and interpreting huge datasets. The dataset has been imported into R.
Some of our goals are identifying missing numbers, dealing with outliers, standardising district and sector names, district- and regional-level summaries of consumption data, and determining the significance of mean differences. The study's conclusions can help legislators and stakeholders by encouraging focused actions and equitable development throughout the state.

2. Results and Interpretation

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

i)Identifying Missing Values

```
Missing Values in Subset:
> print(colSums(is.na(tnnew)))
            state_1            District              Region              Sector
                  0                   0                   0                   0
       State_Region       Meals_At_Home           ricepds_v          Wheatpds_q
                  0                 145                   0                   0
          chicken_q            pulsep_q           wheatos_q No_of_Meals_per_day
                  0                   0                   0                   0
>
```

Most columns in the dataset (state_1, District, Region, Sector, State_Region, ricepds_v, Wheatpds_q, chicken_q, pulsep_q, wheatos_q, and No_of_Meals_per_day) have no missing values. This means that these columns are complete with no NA values.

The column Meals_At_Home has 145 missing values. This indicates that there are 145 rows in the dataset where the value for Meals_At_Home is not available or missing.

In summary, the dataset is mostly complete except for the Meals_At_Home column, which has a significant number of missing values (145). Depending on the analysis, you might need to handle these missing values by imputation, exclusion, or some other method suitable for your data analysis workflow.

b) Check for outliers and describe the outcome of your test and make suitable amendments.

ii)Replacing missing values with mean value

```
# 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)
}
tnnew$Meals_At_Home <- impute_with_mean(tnnew$Meals_At_Home)

# Check for missing values after imputation
cat("Missing Values After Imputation:\n")
issing Values After Imputation:
print(colSums(is.na(tnnew)))
          state_1            District              Region              Sector
                0                   0                   0                   0
     State_Region       Meals_At_Home           ricepds_v          Wheatpds_q
                0                   0                   0                   0
        chicken_q            pulsep_q           wheatos_q No_of_Meals_per_day
                0                   0                   0                   0
```

The missing values in the Meals_At_Home column were successfully replaced with the mean of the non-missing values from that column.
After the imputation, the dataset has no missing values in any of its columns, making it complete and ready for further analysis without missing data issues.

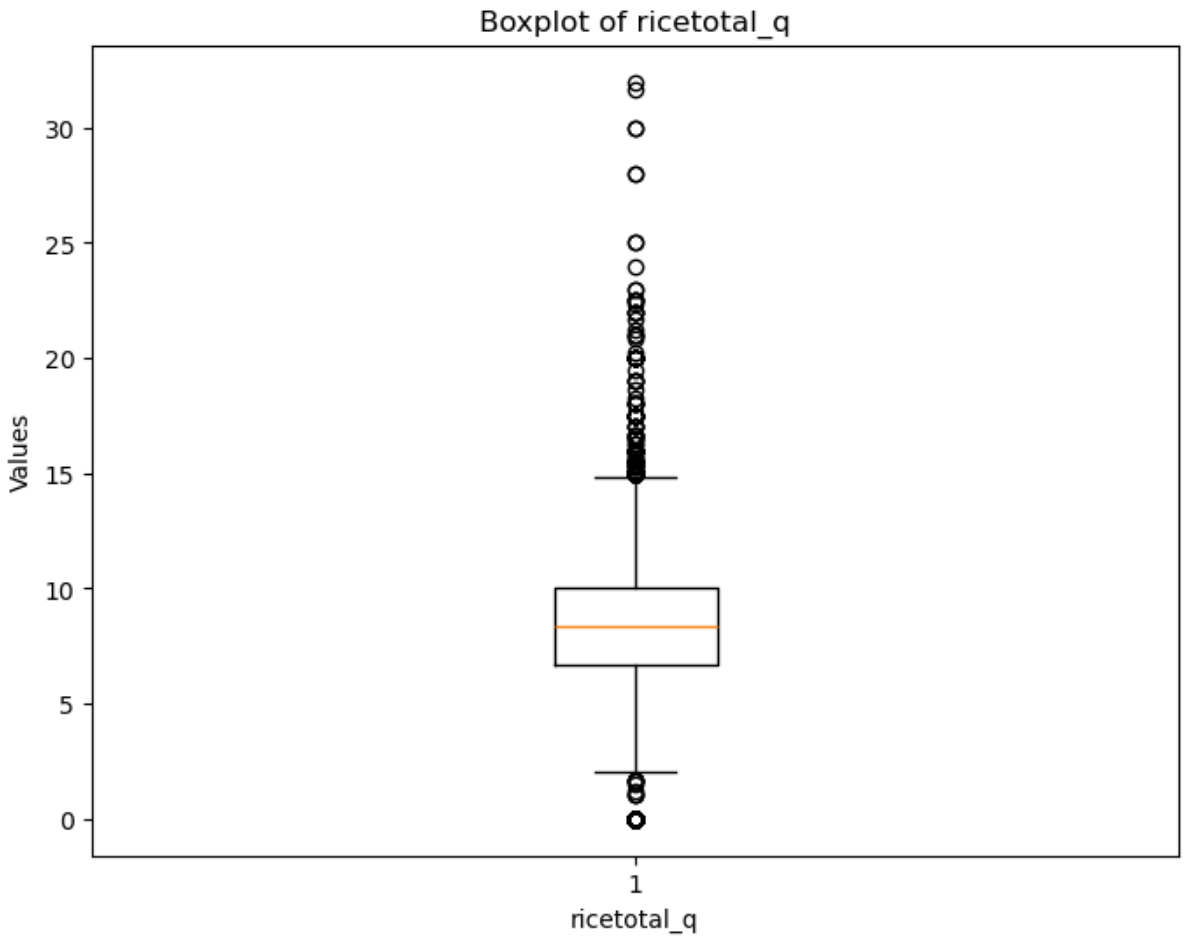
b) Check for outliers and describe the outcome of your test and make suitable amendments.

Plotting the boxplot to visualize outliers.

Code and Result:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.boxplot(TN_clean['ricetotal_q'])
plt.xlabel('ricetotal_q')
plt.ylabel('Values')
plt.title('Boxplot of ricetotal_q')
plt.show()
```

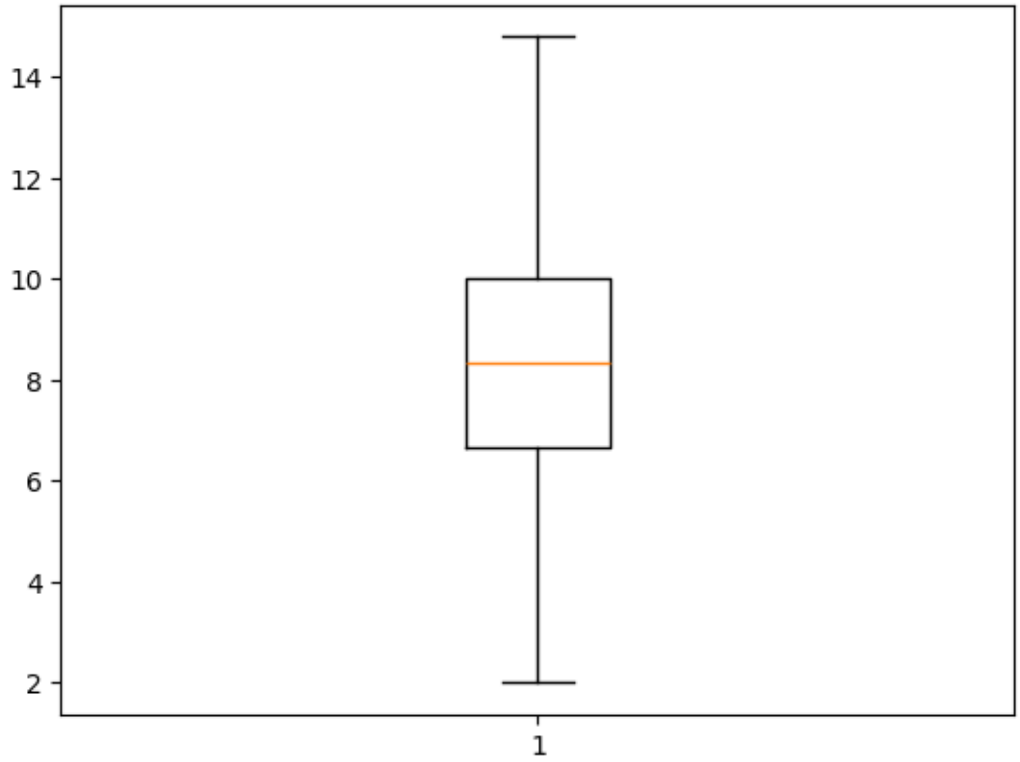


An outlier may be seen in the boxplot above, which represents the variable "ricetotal_q" visually. Outliers can skew statistical analysis and provide false conclusions, which impairs the dependability and accuracy of findings in systems that use data to make decisions. In data-driven decision-making processes, outliers can skew statistical studies and produce false conclusions, which compromises the accuracy and dependability of results. The code below can be used to eliminate the outliers.

```
rice1 = TN_clean['ricetotal_q'].quantile(0.25)
rice2 = TN_clean['ricetotal_q'].quantile(0.75)
iqr_rice = rice2-rice1
up_limit = rice2 + 1.5*iqr_rice
low_limit = rice1 - 1.5*iqr_rice
TN_clean=TN_new[(TN_new['ricetotal_q']<=up_limit)&(TN_new['ricetotal_q'
]>=low_limit)]
```

The result observed after sending this command is as follows

It is possible to identify and eliminate outliers by interpreting quartile ranges. Data points that are more than 1.5 times the interquartile range (IQR) from either quartile are considered outliers and can be removed or handled to maintain the analysis's robustness. The IQR is computed as the difference between the upper and lower quartiles.
The outliers in every other variable can also be eliminated in a similar manner.



It is possible to identify and eliminate outliers by interpreting quartile ranges. Data points that are more than 1.5 times the interquartile range (IQR) from either quartile are considered outliers and can be removed or handled to maintain the analysis's robustness. The IQR is computed as the difference between the upper and lower quartiles.

The outliers in every other variable can also be eliminated in a similar manner.

c) Rename the districts as well as the sector, viz. rural and urban.

In the NSSO68.CSV dataset , a unique number is assigned to each district in a state. The statistics must be accompanied by their individual names to comprehend and identify the state's highest-consuming districts. Likewise, the state's urban and rural areas were assigned to assignments 1 and 2, respectively. Accomplish this, execute the subsequent code.

```
summarize_consumption <- function(group_col) {
  summary <- tnnew %>%
    group_by(across(all_of(group_col))) %>%
    summarise(total = sum(total_consumption)) %>%
    arrange(desc(total))
  return(summary)
}
```

Rename districts and sectors , get codes from appendix of NSSO 68th Round Data

district_mapping <- c("5" = "Dharmapuri", "20" = "thiruvarur", "17" = "ariyalur","12"="coimbatore","8"="salem","18"="cuddalore")

sector_mapping <- c("2" = "URBAN", "1" = "RURAL")

5097	TN	cuddalore	1	RURAL	331	90.00000	40.00000	0.4000000	0.2000000	0.10000000	0.0000000
5098	TN	cuddalore	1	RURAL	331	90.00000	50.00000	0.7500000	0.0000000	0.06250000	0.5000000
5099	TN	cuddalore	1	RURAL	331	90.00000	50.00000	0.0000000	0.0000000	0.00000000	0.0000000
5100	TN	cuddalore	1	RURAL	331	90.00000	50.00000	0.7500000	0.0000000	0.00000000	0.0000000
5102	TN	cuddalore	1	RURAL	331	90.00000	70.00000	0.6000000	0.2000000	0.04000000	0.0000000
5103	TN	cuddalore	1	RURAL	331	90.00000	66.66667	0.0000000	0.3333333	0.00000000	0.0000000
5104	TN	cuddalore	1	RURAL	331	90.00000	100.00000	0.0000000	0.0000000	0.10000000	0.0000000

2265	TN	coimbatore	4	URBAN	334	60.00000	12.000000	0.0000000	0.8000000	0.05000000	0.6000000
2266	TN	coimbatore	4	URBAN	334	90.00000	10.000000	0.5000000	0.3333333	0.08333333	0.0000000
2267	TN	coimbatore	4	URBAN	334	20.00000	15.000000	1.2500000	0.5000000	0.12500000	0.0000000
2268	TN	coimbatore	4	URBAN	334	90.00000	33.600000	1.0000000	0.4000000	0.05000000	0.0000000
2269	TN	coimbatore	4	URBAN	334	90.00000	0.000000	0.0000000	0.5000000	0.06250000	0.5000000
2270	TN	coimbatore	4	URBAN	334	57.00000	24.000000	0.4000000	0.2000000	0.00000000	0.2000000
2271	TN	coimbatore	4	URBAN	334	50.00000	60.000000	0.0000000	0.0000000	0.00000000	0.0000000
2272	TN	coimbatore	4	URBAN	334	90.00000	48.000000	0.0000000	0.5000000	0.00000000	0.0000000

The result as show above has successfully assigned the district names to the given number. Also the sectors 1 and 2 have been replaced as urban and rural sectors respectively.

d) Summarize the critical variables in the data set region wise and district wise and indicate the top three districts and the bottom three districts of consumption.

By summarizing the critical variables as total consumption we can estimate the top 3 and bottom 3 consuming districts.

district_summary <- summarize_consumption("District")

region_summary <- summarize_consumption("Region")

cat("Top 3 Consuming Districts:\n")

print(head(district_summary, 3))

cat("Bottom 3 Consuming Districts:\n")

print(tail(district_summary, 3))

cat("Region Consumption Summary:\n")

print(region summary)

```
> print(head(district_summary, 3))
# A tibble: 3 × 2
  District  total
     <int>  <dbl>
1       12 14154.
2        8 12092.
3       18 11731.
> cat("Bottom 3 Consuming Districts:\n")
Bottom 3 Consuming Districts:
> print(tail(district_summary, 3))
# A tibble: 3 × 2
  District total
     <int> <dbl>
1        5 3955.
2       20 3167.
3       17 3045.
> cat("Region Consumption Summary:\n")
Region Consumption Summary:
> print(region_summary)
# A tibble: 4 × 2
  Region  total
   <int>  <dbl>
1      1 59680.
2      4 55687.
3      3 53938.
4      2 41160.
```

The top three consuming districts are Coimbatore with 14154 units, followed by Salem with 12092 units, and then in the third place Cuddalore with 11731 units

The least consuming district is Dharmapuri with only 3955 units. Followed by Thiruvarur with 3167 units in the second place and Guntur with 3045 points in the last place.

e) Test whether the differences in the means are significant or not.

z_test_result	list [8] (S3: htest)	List of length 8
statistic	double [1]	125.8011
z	double [1]	125.8011
p.value	double [1]	0
conf.int	double [2]	7.68 7.92
estimate	double [2]	37.5 29.7
mean of x	double [1]	37.5096
mean of y	double [1]	29.71146
null.value	double [1]	0
difference in means	double [1]	0
alternative	character [1]	'two.sided'
method	character [1]	'Two-sample z-Test'
data.name	character [1]	'rural and urban'

The two-sample z-test results indicate a highly significant difference between the means of the rural and urban groups. With a z-value of 125.8011 and a p-value effectively 0, we reject the null hypothesis of no difference in means. The confidence interval [7.68, 7.92] suggests that the true difference in means lies within this range, with the rural group having a higher mean (37.5) compared to the urban group (29.7).

CODES

i) R

```
# Set the working directory and verify it
setwd("E:\\SCMA 632\\data")
getwd()

# Function to install and load libraries
install_and_load <- function(package) {
  if (!require(package, character.only = TRUE)) {
    install.packages(package, dependencies = TRUE)
    library(package, character.only = TRUE)
  }
}

# Load required libraries
libraries <- c("dplyr", "readr", "readxl", "tidyr", "ggplot2", "BSDA","glue")
lapply(libraries, install_and_load)

# Reading the file into R
data <- read.csv("NSSO68.csv")

# Filtering for AP
df <- data %>%
  filter(state_1 == "TN")

# Display dataset info
cat("Dataset Information:\n")
print(names(df))
print(head(df))
print(dim(df))

# Finding missing values
missing_info <- colSums(is.na(df))
cat("Missing Values Information:\n")
print(missing_info)

# Sub-setting the data
tnnew <- df %>%
  select(state_1, District, Region, Sector, State_Region, Meals_At_Home, ricepds_v,
Wheatpds_q, chicken_q, pulsep_q, wheatos_q, No_of_Meals_per_day)

# Check for missing values in the subset
```

```
cat("Missing Values in Subset:\n")
print(colSums(is.na(tnnew)))

# 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)
}
tnnew$Meals_At_Home <- impute_with_mean(tnnew$Meals_At_Home)

# Check for missing values after imputation
cat("Missing Values After Imputation:\n")
print(colSums(is.na(tnnew)))

# Finding outliers and removing them
remove_outliers <- function(df, column_name) {
 Q1 <- quantile(df[[column_name]], 0.25)
 Q3 <- quantile(df[[column_name]], 0.75)
 IQR <- Q3 - Q1
 lower_threshold <- Q1 - (1.5 * IQR)
 upper_threshold <- Q3 + (1.5 * IQR)
 df <- subset(df, df[[column_name]] >= lower_threshold & df[[column_name]] <=
upper_threshold)
 return(df)
}
names(tnnew)
#remove outliers in the data set
outlier_columns <- c("ricepds_v", "chicken_q")
for (col in outlier_columns) {
 tnnew <- remove_outliers(tnnew, col)
}

# Summarize consumption
tnnew$total_consumption <- rowSums(tnnew[, c("ricepds_v", "Wheatpds_q", "chicken_q",
"pulsep_q", "wheatos_q")], na.rm = TRUE)

# Summarize and display top and bottom consuming districts and regions
summarize_consumption <- function(group_col) {
 summary <- tnnew %>%
  group_by(across(all_of(group_col))) %>%
  summarise(total = sum(total_consumption)) %>%
  arrange(desc(total))
 return(summary)
}

district_summary <- summarize_consumption("District")
```

```
region_summary <- summarize_consumption("Region")

cat("Top 3 Consuming Districts:\n")
print(head(district_summary, 3))
cat("Bottom 3 Consuming Districts:\n")
print(tail(district_summary, 3))
cat("Region Consumption Summary:\n")
print(region_summary)

# Rename districts and sectors , get codes from appendix of NSSO 68th ROund Data
district_mapping <- c("5" = "Dharmapuri", "20" = "thiruvarur", "17" =
"ariyalur","12"="coimbatore","8"="salem","18"="cuddalore")
sector_mapping <- c("2" = "URBAN", "1" = "RURAL")

tnnew$District <- as.character(tnnew$District)
tnnew$Sector <- as.character(tnnew$Sector)
tnnew$District <- ifelse(tnnew$District %in% names(district_mapping),
district_mapping[tnnew$District], tnnew$District)
tnnew$Sector <- ifelse(tnnew$Sector %in% names(sector_mapping),
sector_mapping[tnnew$Sector], tnnew$Sector)

# Test for differences in mean consumption between urban and rural
rural <- tnnew %>%
  filter(Sector == "RURAL") %>%
  select(total_consumption)

urban <- tnnew %>%
  filter(Sector == "URBAN") %>%
  select(total_consumption)

mean_rural <- mean(rural$total_consumption)
mean_urban <- mean(urban$total_consumption)

# Perform z-test
z_test_result <- z.test(rural, urban, alternative = "two.sided", mu = 0, sigma.x = 2.56, sigma.y
= 2.34, conf.level = 0.95)

# Generate output based on p-value
if (z_test_result$p.value < 0.05) {
  cat(glue::glue("P value is < 0.05 i.e. {round(z_test_result$p.value,5)}, Therefore we reject
the null hypothesis.\n"))
  cat(glue::glue("There is a difference between mean consumptions of urban and rural.\n"))
  cat(glue::glue("The mean consumption in Rural areas is {mean_rural} and in Urban areas its
{mean_urban}\n"))
} else {
  cat(glue::glue("P value is >= 0.05 i.e. {round(z_test_result$p.value,5)}, Therefore we fail to
reject the null hypothesis.\n"))
```

```
  cat(glue::glue("There is no significant difference between mean consumptions of urban and
rural.\n"))
  cat(glue::glue("The mean consumption in Rural area is {mean_rural} and in Urban area its
{mean_urban}\n"))
}
```

ii) Python

python code too long , uploaded on github and canvas