VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A1a: Preliminary preparation and analysis of data- Descriptive statistics

POLAMREDDY MADHUMITHA

V01107517

Date of Submission: 16-06-2024

CONTENTS

Analysing Consumption in the State of TAMIL NADU Using R

Introduction

This study focuses on the state of Tamil Nadu, using NSSO data to identify the top and bottom three consuming districts of Tamil Nadu. We will manipulate and clean the dataset to obtain the necessary data for analysis. The dataset contains consumption-related information, including data on rural and urban sectors, as well as district-wise variations. This data has been imported into R, a powerful statistical programming language renowned for its versatility in handling and analyzing large datasets.

Our objectives include identifying missing values, addressing outliers, standardizing district and sector names, summarizing consumption data regionally and district-wise, and testing the significance of mean differences. The findings from this study can inform policymakers and stakeholders, fostering targeted interventions and promoting equitable development across the state.

OBJECTIVES

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.

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

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

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.

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

BUSINESS SIGNIFICANCE

The focus of this study on Tamil Nadu's consumption patterns from NSSO data holds significant implications for businesses and policymakers. By identifying the top and bottom three consuming districts, the study provides valuable insights for market entry, resource allocation, supply chain optimization, and targeted interventions. Through data cleaning, outlier detection, and significance testing, the findings facilitate informed decision-making, fostering equitable development and promoting Madhya Pradesh's economic growth.

A)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.

#Identifying the missing values.

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

Interpretation: "colSums" is used to calculate and display the number of missing entries within each column. Examining the output, we see that every column except "Meals_At_Home" has zero missing values. This indicates complete data for factors like state, district, region, and various food staples. However, the "Meals_At_Home" column holds a value of 145, signifying that data for 145 entries is absent in this specific category. This substantial absence of information in "Meals_At_Home" could potentially skew any analysis that relies on this data.

#Imputing the values, i.e. replacing the missing values with mean.

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

Interpretation: The code provided effectively filled in the missing values in the dataset by replacing them with the mean value of each respective variable. As shown in the result, there are no longer any missing entries in the selected data.

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

```
> # 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)
+ }
> outlier_columns <- c("ricepds_v", "chicken_q")
> for (col in outlier_columns) {
+   TNnew <- remove_outliers(TNnew, col)
+ }
```

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

Data preparation in NSSO helps to identify the state's top consuming districts. It addresses two limitations: First, it replaces numerical district identifiers with their actual names for better understanding. Second, it likely translates coded sector labels (presumably 1 for urban and 2 for rural) into meaningful terms like "Urban" and "Rural," making the data more user-friendly. This transformation allows for a clearer analysis to pinpoint the districts with the highest consumption levels within the state.

```
# Rename districts and sectors , get codes from appendix of NSSO 68th ROund Data
district_mapping <- c("12" = "Coimbatore", "5" = "Dharmapuri", "8" = "Salem")
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)
```

Output

	state_1	District	Sector	Region	State_Region	ricetotal_q	wheattotal_q	moong_q	Milktotal_q	chicken_q	bread_q	foodtotal_c
20869	TN	29	RURAL	3	333	7.5	1.0	0.000	0	0.5	0.125	25.150588
20870	TN	29	RURAL	3	333	5.0	0.5	0.125	0	0.0	0.000	28.400400
20871	TN	29	RURAL	3	333	7.0	0.5	0.250	0	0.5	0.000	24.595815
20872	TN	29	RURAL	3	333	7.5	0.5	0.000	0	0.0	0.000	22.189005
20873	TN	29	RURAL	3	333	7.0	0.4	0.050	0	0.2	0.000	19.280350

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

```
# 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))
```

Output:

```
Top 3 Consuming Districts:
> print(head(district_summary, 3))
# A tibble: 3 x 2
  District    total
  <chr>       <dbl>
1 Coimbatore 14154.
2 Salem      12092.
3 Cuddalore  11731.
```

Interpretation: 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

```
cat("Bottom 3 Consuming Districts:\n")
print(tail(district_summary, 3))
```

Output:

```
> 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.
```

Interpretation: The bottom three consuming districts are Dharmapuri with 3955 units, followed by Thiruvarur with 3167 units, and then in the third place Ariyalur with 3045 units

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

The first step to this is to have a Hypotheses Statement.

#H0: There is no difference in consumption between urban and rural.

#H1: There is difference in consumption between urban and rural.

```
mean_rural <- mean(rural$total_consumption)

mean_urban <- mean(urban$total_consumption)
```

Result:

P value is < 0.05 i.e. 0, Therefore we reject the null hypothesis.There is a difference between mean consumptions of urban and rural.The mean consumption in Rural areas is 37.5095976533813 and in Urban areas its 29.7114562651036

CODES

```
# Set the working directory and verify it
setwd('/Users/aravi/OneDrive/Desktop/VCU')
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
IPL1 <- read.csv("assignment.csv")

# Filtering for TN
df <- IPL1 %>%
  filter(state == "33")

# 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 IPL1
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)
TNnew$No_of_Meals_per_day <- impute_with_mean(TNnew$No_of_Meals_per_day)

# 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)
}

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("12" = "Coimbatore", "18" = "Cuddalore", "8" = "Salem")
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"))

}
```