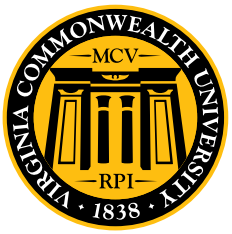


VIRGINIA COMMONWEALTH UNIVERSITY

Statistical Analysis and Modelling (SCMA 632)

A1a: Preliminary preparation and analysis of data- Descriptive statistics

NIHARIHA KAMALANATHAN

V01108259

Date of Submission: 16-06-2024

CONTENTS

Introduction

The focus of this study is on the state of Andhra Pradesh, from the NSSO data, to find the top and bottom three consuming districts of Andhra Pradesh. In the process, we manipulate and clean the dataset to get the required data to analyse. To facilitate this analysis, we have gathered a dataset containing consumption-related information, including data on rural and urban sectors, as well as district-wise variations. The dataset has been imported into R, a powerful statistical programming language renowned for its versatility in handling and analysing 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 and 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 Maharashtra'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 Maharashtra's economic growth.

Using R

Input:

```
#set the working directory
setwd('C:\\Users\\nihar\\OneDrive\\Desktop\\Bootcamp\\SCMA632\\Assignments\\A1a\\Data
')
getwd()


#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)
  }
}

# List of required libraries
libraries <- c("dplyr", "readr", "readxl", "tidyr", "ggplot2", "BSDA", "glue")
# Apply the function to the list of libraries
lapply(libraries, install_and_load)

# Load the dataset into R
data <- read.csv("NSSO68.csv")

#Filtering for Maharasthra
df <- data %>%
  filter(state_1 == "MH")

#Dataset Information Display
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 - Set 1

mhgrains <- df %>%

  select(state_1, District, Region, Sector, ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v)


# Check for missing values in the subset

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

print(colSums(is.na(mhgrains)))


# Finding outliers and removing them

remove_outliers <- function(df,ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v) {

  Q1 <- quantile(df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]], 0.25)

  Q3 <- quantile(df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]], 0.75)

  IQR <- Q3 - Q1

  lower_threshold <- Q1 - (1.5 * IQR)

  upper_threshold <- Q3 + (1.5 * IQR)

  df <- subset(df, df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]] >= lower_threshold & df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]] <= upper_threshold)

  return(df)

}
```

```
outlier_columns <- c("ricetotal_v", "wheattotal_v", "jowarp_v", "barleyp_v", "maizep_v", "maida_v", "suji_v", "bajrap_v", "milletp_v")
for (col in outlier_columns) {
  mhgrains <- remove_outliers(mhgrains, col)
}

# Summarize consumption
 mhgrains$total_consumption <- rowSums(mhgrains[, c("ricetotal_v", "wheattotal_v", "jowarp_v", "barleyp_v", "maizep_v", "maida_v", "suji_v", "bajrap_v", "milletp_v")], na.rm = TRUE)

# Summarize and display top and bottom consuming districts and regions
 summarize_consumption <- function(group_col) {
   summary <- mhgrains %>%
     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
district_mapping <- c("21" = "Thane", "22" = "Mumbai (Suburban) an", "25" = "Pune")
sector_mapping <- c("2" = "URBAN", "1" = "RURAL")
district_mapping <- c("10" = "Bhandara", "22" = "Gadchiroli", "6" = "Washim")

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

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

urban <- mhgrains %>%
  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"))
 }

 #Sub-setting the data - set 2
 mhfruits <- df %>%
  select(state_1, District, Region, Sector, bananano_v, jackfruit_v, watermel_v, pineaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, berries_v, leechi_v, apple_v, grapes_v)

 # Check for missing values in the subset
 cat("Missing Values in Subset:\n")
 print(colSums(is.na(mhfruits)))

 # Finding outliers and removing them
 remove_outliers <- function(df,bananano_v, jackfruit_v, watermel_v, pineaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, berries_v, leechi_v, apple_v, grapes_v) {
  Q1 <- quantile(df[[bananano_v, jackfruit_v, watermel_v, pineaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, berries_v, leechi_v, apple_v, grapes_v]], 0.25)
  Q3 <- quantile(df[[bananano_v, jackfruit_v, watermel_v, pineaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, berries_v, leechi_v, apple_v, grapes_v]], 0.75)
  IQR <- Q3 - Q1
  lower_threshold <- Q1 - (1.5 * IQR)
  upper_threshold <- Q3 + (1.5 * IQR)
```

```
  df <- subset(df, df[[bananano_v, jackfruit_v, watermel_v, pineaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, berries_v, leechi_v, apple_v, grapes_v]] >= lower_threshold & df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]] <= upper_threshold)
  return(df)
}

outlier_columns <- c("bananano_v", "jackfruit_v", "watermel_v", "pineaplno_v", "guava_v", "papayar_v", "sighara_v", "cocogno_v", "mango_v", "kharbooz_v", "pears_v", "berries_v", "leechi_v", "apple_v", "grapes_v")
for (col in outlier_columns) {
  mhfruits <- remove_outliers(mhfruits, col)
}

# Summarize consumption
mhfruits$tot_consumption <- rowSums(mhfruits[, c("bananano_v", "jackfruit_v", "watermel_v", "pineaplno_v", "guava_v", "papayar_v", "sighara_v", "cocogno_v", "mango_v", "kharbooz_v", "pears_v", "berries_v", "leechi_v", "apple_v", "grapes_v")], na.rm = TRUE)

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

district_summary <- summarize_consumptionfruits("District")
region_summary <- summarize_consumptionfruits("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
district_mapping <- c("1" = "Manudurbar", "2" = "Dhule", "3" = "Jalgaon")
sector_mapping <- c("2" = "URBAN", "1" = "RURAL")
district_mapping <- c("33" = "Sindhudurg", "34" = "Kolhapur", "35" = "Sangli")

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

# Test for differences in mean consumption between urban and rural
ruralf <- mhfruits1 %>%
  filter(Sector == "RURAL") %>%
  select(total_consumption)

urbanf <- mhfruits %>%
  filter(Sector == "URBAN") %>%
  select(total_consumption)

mean_ruralf <- mean(ruralf$total_consumption)
mean_urbanf <- mean(urbanf$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_ruralf} and in Urban areas its {mean_urbanf}\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_ruralf} and in Urban area its {mean_urbanf}\n"))

}


#Sub-setting the data - Set 3

mhmeat <- df %>%

  select(state_1, District, Region, Sector, eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v, chicken_v, othrbirds_v )


# Check for missing values in the subset

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

print(colSums(is.na(mhmeat)))


# Finding outliers and removing them

remove_outliers <- function(df,eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v, chicken_v, othrbirds_v) {

  Q1 <- quantile(df[[eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v, chicken_v, othrbirds_v]], 0.25)
```

```
  Q3 <- quantile(df[[eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v, chicken_v, othrbirds_v]], 0.75)
  IQR <- Q3 - Q1
  lower_threshold <- Q1 - (1.5 * IQR)
  upper_threshold <- Q3 + (1.5 * IQR)
  df <- subset(df, df[[eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v, chicken_v, othrbirds_v]] >= lower_threshold & df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]] <= upper_threshold)
  return(df)
 }

 outlier_columns <- c("eggsno_v", "fishprawn_v", "goatmeat_v", "beef_v", "pork_v", "chicken_v", "othrbirds_v")
 for (col in outlier_columns) {
  mhmeat <- remove_outliers(mhmeat, col)
 }

 # Summarize consumption
 mhmeat$total_cons <- rowSums(mhmeat[, c("eggsno_v", "fishprawn_v", "goatmeat_v", "beef_v", "pork_v", "chicken_v", "othrbirds_v")], na.rm = TRUE)

 # Summarize and display top and bottom consuming districts and regions
 summarize_consumption1 <- function(group_col) {
  summary <- mhmeat %>%
    group_by(across(all_of(group_col))) %>%
    summarise(total = sum(total_cons)) %>%
    arrange(desc(total))
  return(summary)
 }

 district_summary <- summarize_consumption1("District")
 region_summary <- summarize_consumption1("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
district_mapping <- c("21" = "Thane", "22" = "Mumbai (Suburban) an", "25" = "Pune")
sector_mapping <- c("2" = "URBAN", "1" = "RURAL")
district_mapping <- c("10" = "Bhandara", "5" = "Akola", "16" = "Hingoli")

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

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

urbanm <- mhmeat %>%
  filter(Sector == "URBAN") %>%
  select(total_cons)

mean_ruralm <- mean(rural$total_cons)
mean_urbanm <- mean(urban$total_cons)
```

```
# 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_ruralm} and in Urban areas its {mean_urbanm}\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_ruralm} and in Urban area its {mean_urbanm}\n"))

}
```

Output

```
#Finding missing values
> missing_info <- colSums(is.na(df))
> cat("Missing Values Information: \n")
Missing Values Information:
> print (missing_info)
                          slno                           grp
                             0                             0
                 Round_Centre                    FSU_number
                             0                             0
                        Round                Schedule_Number
                             0                             0
                       Sample                        Sector
                             0                             0
                        state                  State_Region
                             0                             0
                     District                 Stratum_Number
                             0                             0
                   Sub_Stratum                  Schedule_type
                             0                             0
                    Sub_Round                    Sub_Sample
                             0                             0
               FOD_Sub_Region          Hamlet_Group_Sub_Block
                             0                             0
                             t                 X_Stage_Stratum
```

0	0
HHS_No	Level
0	0
Filler	hhdsz
0	0
NIC_2008	NCO_2004
438	419
HH_type	Religion
0	0
Social_Group	Whether_owns_any_land
0	0
Type_of_land_owned	Land_Owned
1331	1368
Land_Leased_in	Otherwise_possessed
6583	7780
Land_Leased_out	Land_Total_possessed
7880	3
During_July_June_Cultivated	During_July_June_Irrigated
5453	6698
NSS	NSC
0	0
MLT	land_tt
0	3
Cooking_code	Lighting_code
0	0
Dwelling_unit_code	Regular_salary_earner
1	0
Perform_Ceremony	Meals_seved_to_non_hhld_members
1	1058
Possess_ration_card	Type_of_ration_card
0	1727
MPCE_URP	MPCE_MRP
0	0
Person_Srl_No	Relation
0	0
Sex	Age
0	0
Marital_Status	Education
0	0
Days_Stayed_away	No_of_Meals_per_day
6091	0
Meals_School	Meals_Employer
7953	7899
Meals_Others	Meals_Payment
7345	7297
Meals_At_Home	Item_Code
184	0
Source_Code	ricepds_q
78	0
riceos_q	ricetotal_q
0	0
chira_q	khoi_q
0	0
muri_q	ricepro_q
0	0
riceGT_q	Wheatpds_q
0	0
wheatos_q	wheattotal_q
0	0
maida_q	suji_q
0	0
sewai_q	bread_q
0	0
wheatp_q	wheatGT_q
0	0
jowarp_q	bajrap_q
0	0
maizep_q	barleyp_q
0	0

milletp_q	ragip_q
0	0
cerealot_q	cerealtot_q
0	0
cerealsub_q	cerealstt_q
0	0
arhar_q	gramdal_q
0	0
gramwholep_q	gramGT_q
0	0
moong_q	masur_q
0	0
urd_q	peasdal_q
0	0
khesari_q	otpulse_q
0	0
gramp_q	besan_q
0	0
pulsep_q	pulsestot_q
0	0
pulsestt_q	soyabean_q
0	8043
milk_q	babyfood_q
0	0
milkcond_q	curd_q
0	0
ghee_q	butter_q
0	0
icecream_q	otmilkp_q
0	0
Milktotal_q	milkprott_q
0	0
vanas_q	musoil_q
0	0
gnoil_q	cocooil_q
0	0
edioilothr_q	edibletotal_q
0	0
ediblest_q	eggsno_q
0	0
fishprawn_q	goatmeat_q
0	0
beef_q	pork_q
0	0
chicken_q	othrbirds_q
0	0
nonvegtotal_q	emftt_q
0	0
potato_q	onion_q
0	0
tamato_q	brinjal_q
0	0
radish_q	carrot_q
0	0
palak_q	chillig_q
0	0
bhindi_q	parwal_q
0	0
cauli_q	cabbage_q
0	0
pumpkin_q	peas_q
0	0
fbeans_q	lemonno_q
0	0
otveg_q	vegtt_q
0	0
bananano_q	jackfruit_q
0	0
watermel_q	pineaplno_q

```
                0                 0
        cocono_q          cocogno_q
                0                 0
         guava_q           sighara_q
                0                 0
      orangeno_q           papayar_q
                0                 0
         mango_q          kharbooz_q
                0                 0
         pears_q           berries_q
                0                 0
        leechi_q             apple_q
                0                 0
        grapes_q           otfruits_q
                0                 0
      fruitstt_q        fruitt_total
                0                 0
        cocodf_q             gnutdf_q
                0                 0
       datesdf_q           cashewdf_q
                0                 0
      walnutdf_q           otnutsdf_q
                0                 0
      kishmish_q            otherdf_q
                0                 0
dryfruitstotal_q               dftt_q
                0                 0
      sugarpds_q           sugaros_q
                0                 0
       sugarst_q               gur_q
                0                 0
         misri_q             honey_q
                0                 0
    sugartotal_q           sugartt_q
                0                 0
          salt_q            ginger_q
                0                 0
        garlic_q             jeera_q
                0                 0
        dhania_q          turnmeric_q
                0                 0
   blackpepper_q          drychilly_q
                0                 0
      tamarind_q        currypowder_q
                0                 0
      oilseeds_q         spicesothr_q
                0                 0
      spicetot_q        spicestotal_q
                0                 0
      teacupno_q            tealeaf_q
                0                 0
      teatotal_q            cofeeno_q
                0                 0
    coffeepwdr_q         cofeetotal_q
                0                 0
           ice_q           coldbvrg_q
                0                 0
         juice_q          othrbevrg_q
                0                 0
     bevergest_q           Biscuits_q
                0                 0
 preparedsweet_q              pickle_q
                0                 0
     sauce_jam_q     Othrprocessed_q
                0                 0
Beveragestotal_q             ricepds_v
                0                 0
        riceos_v          ricetotal_v
                0                 0
```

chira_v	khoi_v
0	0
muri_v	ricepro_v
0	0
riceGT_v	Wheatpds_v
0	0
wheatos_v	wheattotal_v
0	0
maida_v	suji_v
0	0
sewai_v	bread_v
0	0
wheatp_v	wheatGT_v
0	0
jowarp_v	bajrap_v
0	0
maizep_v	barleyp_v
0	0
milletp_v	ragip_v
0	0
cerealot_v	cerealtot_v
0	0
cerealsub_v	cerealstt_v
0	0
arhar_v	gramdal_v
0	0
gramwholep_v	gramGT_v
0	0
moong_v	masur_v
0	0
urd_v	peasdal_v
0	0
khesari_v	otpulse_v
0	0
gramp_v	besan_v
0	0
pulsep_v	pulsestot_v
0	0
pulsestt_v	soyabean_v
0	8043
milk_v	babyfood_v
0	0
milkcond_v	curd_v
0	0
ghee_v	butter_v
0	0
icecream_v	otmilkp_v
0	0
Milktotal_v	milkprott_v
0	0
vanas_v	musoil_v
0	0
gnoil_v	cocooil_v
0	0
edioilothr_v	edibletotal_v
0	0
ediblest_v	eggsno_v
0	0
fishprawn_v	goatmeat_v
0	0
beef_v	pork_v
0	0
chicken_v	othrbirds_v
0	0
nonvegtotal_v	emftt_v
0	0
potato_v	onion_v
0	0
tamato_v	brinjal_v

0	0
radish_v	carrot_v
0	0
palak_v	chillig_v
0	0
bhindi_v	parwal_v
0	0
cauli_v	cabbage_v
0	0
pumpkin_v	peas_v
0	0
fbeans_v	lemonno_v
0	0
otveg_v	vegtt_v
0	0
bananano_v	jackfruit_v
0	0
watermel_v	pineaplno_v
0	0
cocono_v	cocogno_v
0	0
guava_v	sighara_v
0	0
orangeno_v	papayar_v
0	0
mango_v	kharbooz_v
0	0
pears_v	berries_v
0	0
leechi_v	apple_v
0	0
grapes_v	otfruits_v
0	0
fruitstt_v	cocodf_v
0	0
gnutdf_v	datesdf_v
0	0
cashewdf_v	walnutdf_v
0	0
otnutsdf_v	kishmish_v
0	0
otherdf_v	dryfruitstotal_v
0	0
dftt_v	sugarpds_v
0	0
sugaros_v	sugarst_v
0	0
gur_v	misri_v
0	0
honey_v	sugartotal_v
0	0
sugartt_v	salt_v
0	0
ginger_v	garlic_v
0	0
jeera_v	dhania_v
0	0
turnmeric_v	blackpepper_v
0	0
drychilly_v	tamarind_v
0	0
currypowder_v	oilseeds_v
0	0
spicesothr_v	spicetot_v
0	0
spicestotal_v	teacupno_v
0	0
tealeaf_v	teatotal_v
0	0

```
                    cofeeno_v                     coffeepwdr_v
                            0                                0
                 cofeetotal_v                            ice_v
                            0                                0
                   coldbvrg_v                         juice_v
                            0                                0
                  othrbevrg_v                      bevergest_v
                            0                                0
                   Biscuits_v                  preparedsweet_v
                            0                                0
                     pickle_v                      sauce_jam_v
                            0                                0
               Othrprocessed_v                 Beveragestotal_v
                            0                                0
                  foodtotal_v                      foodtotal_q
                            0                                0
                      state_1                           Region
                            0                                0
                fruits_df_tt_v                         fv_tot
                            0                                0
> #Sub-setting the data - Set 1
> mhgrains <- df %>%
+   select(state_1, District, Region, Sector, ricetotal_v, wheattotal_v, j
owarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v)
>
> # Check for missing values in the subset
> cat("Missing Values in Subset:\n")
Missing Values in Subset:
> print(colSums(is.na(mhgrains)))
     state_1     District       Region       Sector  ricetotal_v
           0            0            0            0            0
wheattotal_v     jowarp_v    barleyp_v     maizep_v      maida_v
           0            0            0            0            0
      suji_v     bajrap_v    milletp_v
           0            0            0
> # Finding outliers and removing them
> remove_outliers <- function(df,ricetotal_v, wheattotal_v, jowarp_v, barl
eyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v) {
+   Q1 <- quantile(df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, mai
zep_v, maida_v, suji_v, bajrap_v, milletp_v]], 0.25)
+   Q3 <- quantile(df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, mai
zep_v, maida_v, suji_v, bajrap_v, milletp_v]], 0.75)
+   IQR <- Q3 - Q1
+   lower_threshold <- Q1 - (1.5 * IQR)
+   upper_threshold <- Q3 + (1.5 * IQR)
+   df <- subset(df, df[[ricetotal_v, wheattotal_v, jowarp_v, barleyp_v, m
aizep_v, maida_v, suji_v, bajrap_v, milletp_v]] >= lower_threshold & df[[r
icetotal_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v,
bajrap_v, milletp_v]] <= upper_threshold)
+   return(df)
+ }
>
> outlier_columns <- c("ricetotal_v", "wheattotal_v", "jowarp_v", "barleyp
_v", "maizep_v", "maida_v", "suji_v", "bajrap_v", "milletp_v")
> for (col in outlier_columns) {
+   mhgrains <- remove_outliers(mhgrains, col)
+ }

> # Summarize consumption
>  mhgrains$total_consumption <- rowSums(mhgrains[, c("ricetotal_v", "whea
ttotal_v", "jowarp_v", "barleyp_v", "maizep_v", "maida_v", "suji_v", "bajr
ap_v", "milletp_v")], na.rm = TRUE)
>
> # Summarize and display top and bottom consuming districts and regions
>  summarize_consumption <- function(group_col) {
+    summary <- mhgrains %>%
+      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")
Top 3 Consuming Districts:
>  print(head(district_summary, 3))
# A tibble: 3 × 2
  District   total
     <int>   <dbl>
1       22 136733.
2       21 118477.
3       25 105528.
>  cat("Bottom 3 Consuming Districts:\n")
Bottom 3 Consuming Districts:
>  print(tail(district_summary, 3))
# A tibble: 3 × 2
  District  total
     <int>  <dbl>
1       10 13998.
2       12 12803.
3        6 12433.
>  cat("Region Consumption Summary:\n")
Region Consumption Summary:
>  print(region_summary)
# A tibble: 6 × 2
  Region   total
   <int>   <dbl>
1      2 356906.
2      1 325819.
3      4 199984.
4      5 196667.
5      3 115717.
6      6  67265.
> # Rename districts and sectors
>  district_mapping <- c("21" = "Thane", "22" = "Mumbai (Suburban) an", "2
5" = "Pune")
>  sector_mapping <- c("2" = "URBAN", "1" = "RURAL")
>  district_mapping <- c("10" = "Bhandara", "22" = "Gadchiroli", "6" = "Wa
shim")
>
>  mhgrains$District <- as.character(mhgrains$District)
>  mhgrains$Sector <- as.character(mhgrains$Sector)
>  mhgrains$District <- ifelse(mhgrains$District %in% names(district_mappi
ng), district_mapping[mhgrains$District], mhgrains$District)
>  mhgrains$Sector <- ifelse(mhgrains$Sector %in% names(sector_mapping), s
ector_mapping[mhgrains$Sector], mhgrains$Sector)
>
>  # Test for differences in mean consumption between urban and rural
>  rural <- mhgrains %>%
+    filter(Sector == "RURAL") %>%
+    select(total_consumption)
>
>  urban <- mhgrains %>%
+    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 ur
ban and rural.\n"))
+    cat(glue::glue("The mean consumption in Rural areas is {mean_rural} a
nd 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 consu
mptions of urban and rural.\n"))
+    cat(glue::glue("The mean consumption in Rural area is {mean_rural} an
d in Urban area its {mean_urban}\n"))
+  }
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 consump
tion in Rural areas is 148.579142894145 and in Urban areas its 165.3674990
23822
>  #Sub-setting the data - set 2
>  mhfruits <- df %>%
+    select(state_1, District, Region, Sector, bananano_v, jackfruit_v, wa
termel_v, pineaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v,
kharbooz_v, pears_v, berries_v, leechi_v, apple_v, grapes_v)
>
>  # Check for missing values in the subset
>  cat("Missing Values in Subset:\n")
Missing Values in Subset:
>  print(colSums(is.na(mhfruits)))
    state_1    District      Region      Sector  bananano_v jackfruit_v
          0           0           0           0           0           0
 watermel_v pineaplno_v     guava_v   papayar_v   sighara_v   cocogno_v
          0           0           0           0           0           0
    mango_v  kharbooz_v     pears_v   berries_v    leechi_v     apple_v
          0           0           0           0           0           0
   grapes_v
          0
>
>  # Finding outliers and removing them
>  remove_outliers <- function(df,bananano_v, jackfruit_v, watermel_v, pin
eaplno_v, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, p
ears_v, berries_v, leechi_v, apple_v, grapes_v) {
+    Q1 <- quantile(df[[bananano_v, jackfruit_v, watermel_v, pineaplno_v,
guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, be
rries_v, leechi_v, apple_v, grapes_v]], 0.25)
+    Q3 <- quantile(df[[bananano_v, jackfruit_v, watermel_v, pineaplno_v,
guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v, be
rries_v, leechi_v, apple_v, grapes_v]], 0.75)
+    IQR <- Q3 - Q1
+    lower_threshold <- Q1 - (1.5 * IQR)
+    upper_threshold <- Q3 + (1.5 * IQR)
+    df <- subset(df, df[[bananano_v, jackfruit_v, watermel_v, pineaplno_v
, guava_v, papayar_v, sighara_v, cocogno_v, mango_v, kharbooz_v, pears_v,
berries_v, leechi_v, apple_v, grapes_v]] >= lower_threshold & df[[ricetota
l_v, wheattotal_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_
v, milletp_v]] <= upper_threshold)
+    return(df)
+  }
>
>  outlier_columns <- c("bananano_v", "jackfruit_v", "watermel_v", "pineap
lno_v", "guava_v", "papayar_v", "sighara_v", "cocogno_v", "mango_v", "khar
booz_v", "pears_v", "berries_v", "leechi_v", "apple_v", "grapes_v")
>  for (col in outlier_columns) {
+    mhfruits <- remove_outliers(mhfruits, col)
+  }

>  # Summarize consumption
>  mhfruits$tot_consumption <- rowSums(mhfruits1[, c("bananano_v", "jackfr
uit_v", "watermel_v", "pineaplno_v", "guava_v", "papayar_v", "sighara_v",
"cocogno_v", "mango_v", "kharbooz_v", "pears_v", "berries_v", "leechi_v",
"apple_v", "grapes_v")], na.rm = TRUE)
Error: object 'mhfruits1' not found
```

```
>  # Summarize consumption
>  mhfruits$tot_consumption <- rowSums(mhfruits[, c("bananano_v", "jackfru
it_v", "watermel_v", "pineaplno_v", "guava_v", "papayar_v", "sighara_v", "
cocogno_v", "mango_v", "kharbooz_v", "pears_v", "berries_v", "leechi_v", "
apple_v", "grapes_v")], na.rm = TRUE)
>
>  # Summarize and display top and bottom consuming districts and regions
>  summarize_consumptionfruits <- function(group_col) {
+    summary <- mhfruits %>%
+      group_by(across(all_of(group_col))) %>%
+      summarise(total = sum(mhfruits$tot_consumption)) %>%
+      arrange(desc(total))
+    return(summary)
+  }
>  district_summary <- summarize_consumptionfruits("District")
>  region_summary <- summarize_consumptionfruits("Region")
>
>  cat("Top 3 Consuming Districts:\n")
Top 3 Consuming Districts:
>  print(head(district_summary, 3))
# A tibble: 3 × 2
  District   total
     <int>   <dbl>
1        1 206260.
2        2 206260.
3        3 206260.
>  cat("Bottom 3 Consuming Districts:\n")
Bottom 3 Consuming Districts:
>  print(tail(district_summary, 3))
# A tibble: 3 × 2
  District   total
     <int>   <dbl>
1       33 206260.
2       34 206260.
3       35 206260.
>  cat("Region Consumption Summary:\n")
Region Consumption Summary:
>  print(region_summary)
# A tibble: 6 × 2
  Region   total
   <int>   <dbl>
1      1 206260.
2      2 206260.
3      3 206260.
4      4 206260.
5      5 206260.
6      6 206260.
>
>  # Rename districts and sectors
>  district_mapping <- c("1" = "Manudurbar", "2" = "Dhule", "3" = "Jalgaon
")
>  sector_mapping <- c("2" = "URBAN", "1" = "RURAL")
>  district_mapping <- c("33" = "Sindhudurg", "34" = "Kolhapur", "35" = "S
angli")
>
>  mhfruits$District <- as.character(mhfruits$District)
>  mhfruits$Sector <- as.character(mhfruits$Sector)
>  mhfruits$District <- ifelse(mhfruits$District %in% names(district_mappi
ng), district_mapping[mhfruits$District], mhfruits$District)
>  mhfruits$Sector <- ifelse(mhfruits$Sector %in% names(sector_mapping), s
ector_mapping[mhfruits$Sector], mhfruits$Sector)
> # Test for differences in mean consumption between urban and rural
>  ruralf <- mhfruits1 %>%
+    filter(Sector == "RURAL") %>%
+    select(total_consumption)
>  #Sub-setting the data - Set 3
>  mhmeat <- df %>%
+    select(state_1, District, Region, Sector, eggsno_v, fishprawn_v, goat
meat_v, beef_v, pork_v, chicken_v, othrbirds_v )
```

```
>
>  # Check for missing values in the subset
>  cat("Missing Values in Subset:\n")
Missing Values in Subset:
>  print(colSums(is.na(mhmeat)))
    state_1    District      Region      Sector    eggsno_v fishprawn_v
          0           0           0           0           0           0
 goatmeat_v      beef_v      pork_v   chicken_v othrbirds_v
          0           0           0           0           0
>
>  # Finding outliers and removing them
>  remove_outliers <- function(df,eggsno_v, fishprawn_v, goatmeat_v, beef_
v, pork_v, chicken_v, othrbirds_v) {
+    Q1 <- quantile(df[[eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v,
chicken_v, othrbirds_v]], 0.25)
+    Q3 <- quantile(df[[eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_v,
chicken_v, othrbirds_v]], 0.75)
+    IQR <- Q3 - Q1
+    lower_threshold <- Q1 - (1.5 * IQR)
+    upper_threshold <- Q3 + (1.5 * IQR)
+    df <- subset(df, df[[eggsno_v, fishprawn_v, goatmeat_v, beef_v, pork_
v, chicken_v, othrbirds_v]] >= lower_threshold & df[[ricetotal_v, wheattot
al_v, jowarp_v, barleyp_v, maizep_v, maida_v, suji_v, bajrap_v, milletp_v]
] <= upper_threshold)
+    return(df)
+  }
>
>  outlier_columns <- c("eggsno_v", "fishprawn_v", "goatmeat_v", "beef_v",
"pork_v", "chicken_v", "othrbirds_v")
>  for (col in outlier_columns) {
+    mhmeat <- remove_outliers(mhmeat, col)
+  }

> # Summarize consumption
>  mhmeat$total_cons <- rowSums(mhmeat[, c("eggsno_v", "fishprawn_v", "goa
tmeat_v", "beef_v", "pork_v", "chicken_v", "othrbirds_v")], na.rm = TRUE)
>
>  # Summarize and display top and bottom consuming districts and regions
>  summarize_consumption1 <- function(group_col) {
+    summary <- mhmeat %>%
+      group_by(across(all_of(group_col))) %>%
+      summarise(total = sum(total_cons)) %>%
+      arrange(desc(total))
+    return(summary)
+  }
> district_summary <- summarize_consumption1("District")
>  region_summary <- summarize_consumption1("Region")
>
>  cat("Top 3 Consuming Districts:\n")
Top 3 Consuming Districts:
>  print(head(district_summary, 3))
# A tibble: 3 × 2
  District  total
     <int>  <dbl>
1       22 69079.
2       21 53515.
3       25 24383.
>  cat("Bottom 3 Consuming Districts:\n")
Bottom 3 Consuming Districts:
>  print(tail(district_summary, 3))
# A tibble: 3 × 2
  District total
     <int> <dbl>
1       16 3725.
2        5 2976.
3       10 2535.
>  cat("Region Consumption Summary:\n")
Region Consumption Summary:
>  print(region_summary)
```

```
# A tibble: 6 × 2
  Region   total
   <int>   <dbl>
1      1 155462.
2      2  74889.
3      4  50926.
4      5  46547.
5      3  33295.
6      6  19053.
>
>  # Rename districts and sectors
>  district_mapping <- c("21" = "Thane", "22" = "Mumbai (Suburban) an", "2
5" = "Pune")
>  sector_mapping <- c("2" = "URBAN", "1" = "RURAL")
>  district_mapping <- c("10" = "Bhandara", "5" = "Akola", "16" = "Hingoli
")
>
>  mhmeat$District <- as.character(mhmeat$District)
>  mhmeat$Sector <- as.character(mhmeat$Sector)
>  mhmeat$District <- ifelse(mhmeat$District %in% names(district_mapping),
district_mapping[mhmeat$District], mhmeat$District)
>  mhmeat$Sector <- ifelse(mhmeat$Sector %in% names(sector_mapping), secto
r_mapping[mhmeat$Sector], mhmeat$Sector)
>  # Test for differences in mean consumption between urban and rural
>  ruralm <- mhmeat %>%
+    filter(Sector == "RURAL") %>%
+    select(total_consumption)
>  # Test for differences in mean consumption between urban and rural
>  ruralm <- mhmeat %>%
+    filter(Sector == "RURAL") %>%
+    select(total_cons)
>
>  urbanm <- mhmeat %>%
+    filter(Sector == "URBAN") %>%
+    select(total_cons)
>
>  mean_ruralm <- mean(rural$total_cons)
>  mean_urbanm <- mean(urban$total_cons)
> # 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 ur
ban and rural.\n"))
+    cat(glue::glue("The mean consumption in Rural areas is {mean_ruralm}
and in Urban areas its {mean_urbanm}\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 consu
mptions of urban and rural.\n"))
+    cat(glue::glue("The mean consumption in Rural area is {mean_ruralm} a
nd in Urban area its {mean_urbanm}\n"))
+  }
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 consump
tion in Rural areas is 148.579142894145 and in Urban areas its 165.3674990
23822
```

a1a-v01108259

June 16, 2024

[98]:
```
import os, pandas as pd, numpy as np
```

[99]:
```
os.chdir("C:\\Users\\nihar\\OneDrive\\Desktop\\Bootcamp\\SCMA␣
↪632\\Assignments\\A1a\\Data")
```

[100]:
```
df=pd.read_csv("NSSO68.csv",encoding="Latin-1", low_memory=False)
```

[101]:
```
df.head()
```

[101]:

	slno	grp	Round_Centre	FSU_number	Round	Schedule_Number	Sample	\
0	1	4.10E+31	1	41000	68	10	1	
1	2	4.10E+31	1	41000	68	10	1	
2	3	4.10E+31	1	41000	68	10	1	
3	4	4.10E+31	1	41000	68	10	1	
4	5	4.10E+31	1	41000	68	10	1	

	Sector	state	State_Region	…	pickle_v	sauce_jam_v	Othrprocessed_v	\
0	2	24	242	…	0.0	0.0	0.0	
1	2	24	242	…	0.0	0.0	0.0	
2	2	24	242	…	0.0	0.0	0.0	
3	2	24	242	…	0.0	0.0	0.0	
4	2	24	242	…	0.0	0.0	0.0	

	Beveragestotal_v	foodtotal_v	foodtotal_q	state_1	Region	\
0	0.000000	1141.492400	30.942394	GUJ	2	
1	17.500000	1244.553500	29.286153	GUJ	2	
2	0.000000	1050.315400	31.527046	GUJ	2	
3	33.333333	1142.591667	27.834607	GUJ	2	
4	75.000000	945.249500	27.600713	GUJ	2	

	fruits_df_tt_v	fv_tot
0	12.000000	154.18
1	333.000000	484.95
2	35.000000	214.84
3	168.333333	302.30
4	15.000000	148.00

```
[5 rows x 384 columns]
```

[102]:
```
MH = df[df['state_1']=="MH"]
```

[103]:
```
MH.isnull().sum().sort_values(ascending = False)
```

[103]:
```
soyabean_q          8043
soyabean_v          8043
Meals_School        7953
Meals_Employer      7899
Land_Leased_out     7880
                    …
palak_q                0
carrot_q               0
radish_q               0
brinjal_q              0
fv_tot                 0
Length: 384, dtype: int64
```

[104]:
```
df.columns
```

[104]:
```
Index(['slno', 'grp', 'Round_Centre', 'FSU_number', 'Round', 'Schedule_Number',
       'Sample', 'Sector', 'state', 'State_Region',
       …
       'pickle_v', 'sauce_jam_v', 'Othrprocessed_v', 'Beveragestotal_v',
       'foodtotal_v', 'foodtotal_q', 'state_1', 'Region', 'fruits_df_tt_v',
       'fv_tot'],
      dtype='object', length=384)
```

[105]:
```
MH_new = MH[['state_1', 'District', 'Sector','Region', 'ricetotal_v',␣
 ↪'wheattotal_v', 'jowarp_v', 'barleyp_v', 'maizep_v', 'maida_v', 'suji_v',␣
 ↪'bajrap_v', 'milletp_v', 'wheattotal_v', 'jowarp_v', 'barleyp_v',␣
 ↪'maizep_v', 'maida_v', 'suji_v', 'bajrap_v', 'milletp_v']]
```

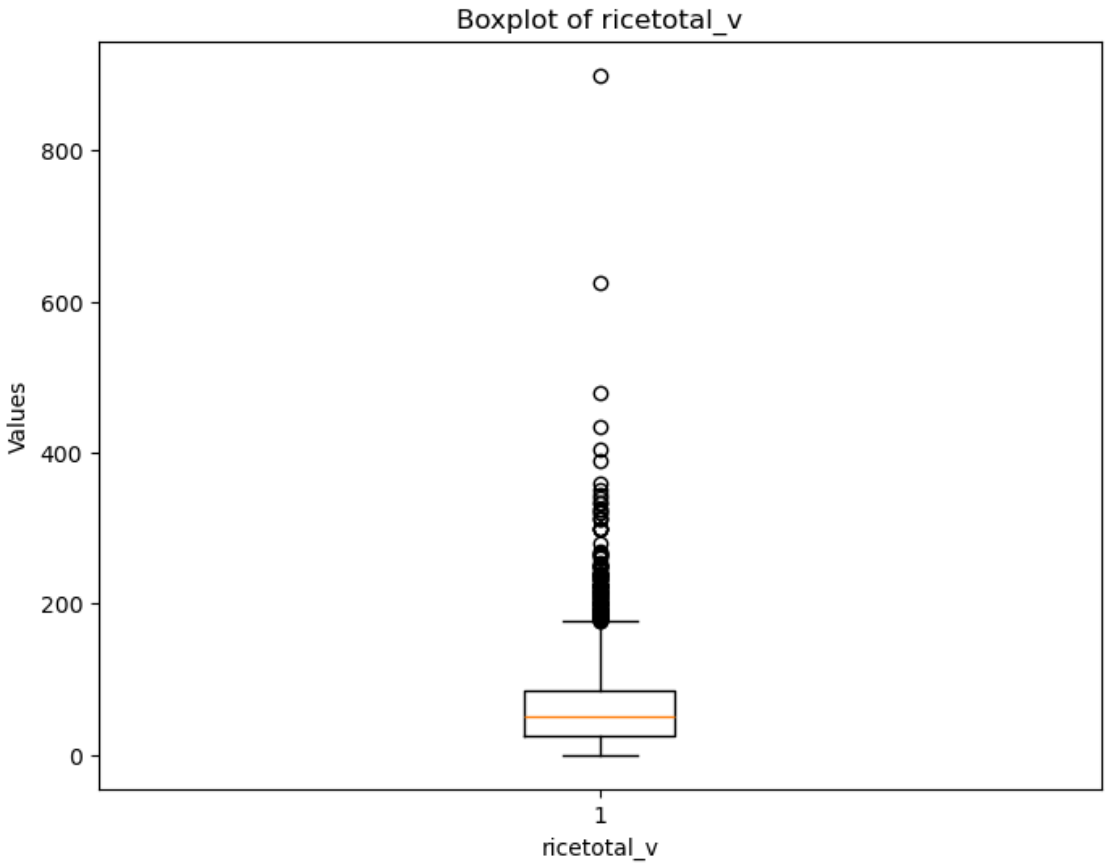
[106]:
```
MH_new.isnull().sum().sort_values(ascending = False)
```

[106]:
```
state_1         0
bajrap_v        0
bajrap_v        0
suji_v          0
maida_v         0
maizep_v        0
barleyp_v       0
jowarp_v        0
wheattotal_v    0
milletp_v       0
suji_v          0
```

```
District        0
maida_v         0
maizep_v        0
barleyp_v       0
jowarp_v        0
wheattotal_v    0
ricetotal_v     0
Region          0
Sector          0
milletp_v       0
dtype: int64
```

[107]:
```
# Outlier Checking
```

[108]:
```
import matplotlib.pyplot as plt
# Assuming MH_clean is your DataFrame
plt.figure(figsize=(8, 6))
plt.boxplot(MH_new['ricetotal_v'])
plt.xlabel('ricetotal_v')
plt.ylabel('Values')
plt.title('Boxplot of ricetotal_v')
plt.show()
```

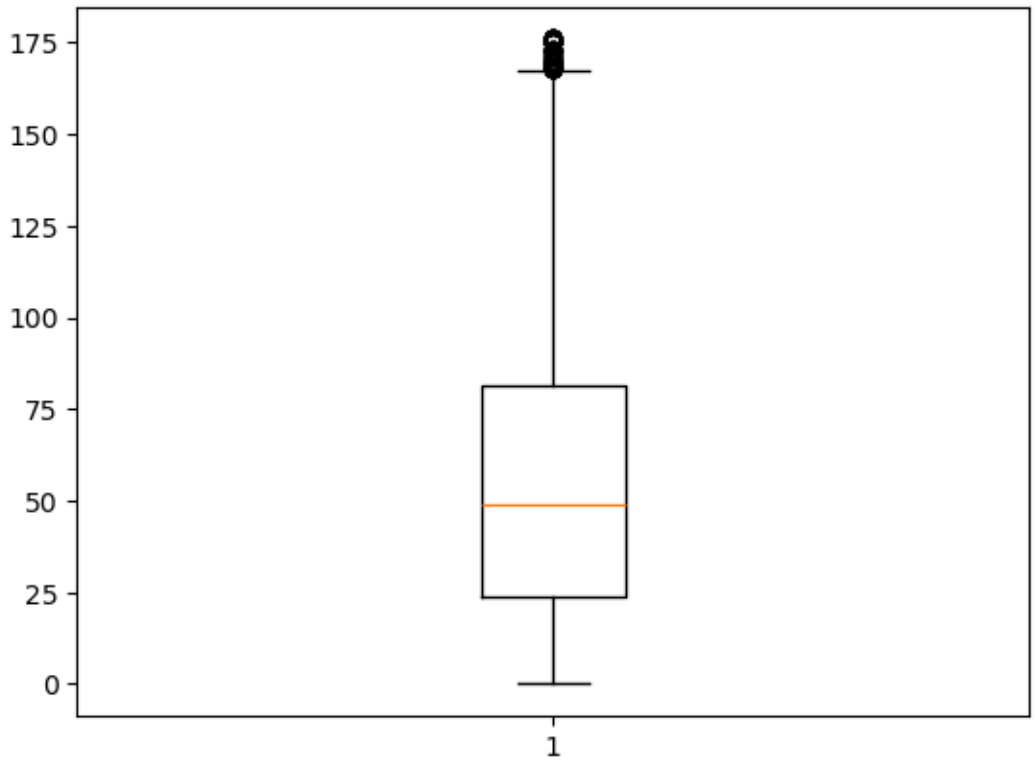


[109]:
```
rice1 = MH_new['ricetotal_v'].quantile(0.25)
rice2 = MH_new ['ricetotal_v'].quantile(0.75)
iqr_rice = rice2-rice1
up_limit = rice2 + 1.5*iqr_rice
low_limit = rice1 - 1.5*iqr_rice
```

[110]:
```
MH_new␣
↪=MH_new[(MH_new['ricetotal_v']<=up_limit)&(MH_new['ricetotal_v']>=low_limit)]
```

[111]:
```
plt.boxplot(MH_new['ricetotal_v'])
```

[111]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x165fa9ca410>,
  <matplotlib.lines.Line2D at 0x165f95e38d0>],
 'caps': [<matplotlib.lines.Line2D at 0x1659021fe90>,
  <matplotlib.lines.Line2D at 0x1659021e710>],
 'boxes': [<matplotlib.lines.Line2D at 0x16590107410>],
 'medians': [<matplotlib.lines.Line2D at 0x1659021c250>],
 'fliers': [<matplotlib.lines.Line2D at 0x1659021e0d0>],
 'means': []}
```



[112]:
```
MH_new['District'].unique()
```

[112]:
```
array([21, 24, 22,  9, 13, 14, 12, 11,  7,  4,  5,  6,  8, 10, 28, 20, 27,
       18, 19, 17, 15,  2,  3,  1, 16, 25, 34, 35, 33, 31, 30, 29, 26, 32],
      dtype=int64)
```

[113]:
```
# Replace values in the 'Sector' column
MH_new.loc[:,'Sector'] = MH_new['Sector'].replace([1, 2], ['URBAN', 'RURAL'])
```

[114]:
```
#total consumption
```

[115]:
```
MH_new.columns
```

[115]:
```
Index(['state_1', 'District', 'Sector', 'Region', 'ricetotal_v',
       'wheattotal_v', 'jowarp_v', 'barleyp_v', 'maizep_v', 'maida_v',
       'suji_v', 'bajrap_v', 'milletp_v', 'wheattotal_v', 'jowarp_v',
       'barleyp_v', 'maizep_v', 'maida_v', 'suji_v', 'bajrap_v', 'milletp_v'],
      dtype='object')
```

[116]:
```
MH_new.loc[MH_new.index, 'total_consumption'] = MH_new[['ricetotal_v',␣
↪'wheattotal_v', 'jowarp_v', 'barleyp_v', 'maizep_v', 'maida_v', 'suji_v',␣
↪'bajrap_v', 'milletp_v', 'wheattotal_v', 'jowarp_v', 'barleyp_v',␣
↪'maizep_v', 'maida_v', 'suji_v', 'bajrap_v', 'milletp_v']].sum(axis=1)
```

[117]:
```
MH_new.head()
```

[117]:

	state_1	District	Sector	Region	ricetotal_v	wheattotal_v	jowarp_v	\
7577	MH	21	RURAL	1	91.0	100.0	0.0	
7579	MH	21	RURAL	1	0.0	0.0	0.0	
7580	MH	21	RURAL	1	84.0	120.0	0.0	
7581	MH	21	RURAL	1	75.0	100.0	0.0	
7582	MH	21	RURAL	1	70.0	100.0	0.0	

	barleyp_v	maizep_v	maida_v	…	milletp_v	wheattotal_v	jowarp_v	\
7577	0.0	0.0	0.0	…	0.0	100.0	0.0	
7579	0.0	0.0	0.0	…	0.0	0.0	0.0	
7580	0.0	0.0	0.0	…	0.0	120.0	0.0	
7581	0.0	0.0	0.0	…	0.0	100.0	0.0	
7582	0.0	0.0	0.0	…	0.0	100.0	0.0	

	barleyp_v	maizep_v	maida_v	suji_v	bajrap_v	milletp_v	\
7577	0.0	0.0	0.0	0.0	0.0	0.0	
7579	0.0	0.0	0.0	0.0	0.0	0.0	
7580	0.0	0.0	0.0	0.0	0.0	0.0	
7581	0.0	0.0	0.0	8.0	0.0	0.0	
7582	0.0	0.0	0.0	0.0	0.0	0.0	

```
      total_consumption
7577              491.0
7579                0.0
7580              564.0
7581              507.0
7582              470.0

[5 rows x 22 columns]
```

[118]:
```
MH_new.groupby('Region').agg({'total_consumption':['std','mean','max','min']})
```

[118]:
```
       total_consumption
                     std        mean          max  min
Region
1             213.205301  379.859149  1536.000000  0.0
2             237.839270  559.018854  3914.000000  0.0
3             164.926829  373.196319  1062.000000  0.0
4             227.981142  481.248561  1897.333333  0.0
5             176.564713  437.841208  1810.000000  0.0
6             141.509527  263.623876   718.666667  0.0
```

[119]:
```
MH_new.groupby('District').agg({'total_consumption':['std','mean','max','min']})
```

[119]:
```
         total_consumption
                       std        mean          max         min
District
1                128.878427  339.352914   771.000000    0.000000
2                168.676345  361.767752  1062.000000    0.000000
3                180.368018  394.218173  1030.000000    0.000000
4                194.829357  432.566989  1618.750000   94.000000
5                195.035406  487.482292  1040.000000    0.000000
6                223.577856  427.903250  1810.000000   98.666667
7                159.565486  496.414703  1288.000000   70.000000
8                164.613606  369.612720  1085.000000    0.000000
9                156.410786  397.143424   865.333333    0.000000
10               145.450646  289.645695   604.000000    0.000000
11               132.721521  237.402968   684.000000    0.000000
12               134.306589  235.164001   615.000000    0.000000
13               144.449919  284.660926   718.666667    0.000000
14               148.269703  461.327282   855.000000   55.000000
15               231.507821  481.874432  1198.000000    0.000000
16               240.818388  464.477156  1685.666667    0.000000
17               200.951678  537.950080  1392.000000  100.833333
18               139.056245  389.471848   852.500000    0.000000
19               136.982517  379.977732   880.000000    0.000000
20               160.104166  373.474416   797.000000    0.000000
21               220.588450  364.382038  1536.000000    0.000000
```

```
22              213.234795  429.310403  1446.500000    0.000000
24              169.962282  248.111343   740.000000    0.000000
25              249.552792  509.617017  2247.000000    0.000000
26              201.065678  518.928595  1050.000000    0.000000
27              183.975002  441.276854  1490.000000    0.000000
28              294.460217  584.160451  1897.333333    0.000000
29              243.994401  633.051415  1400.000000   95.000000
30              212.449008  617.613736  1260.000000    0.000000
31              237.623347  548.412093  1096.000000    0.000000
32              161.173572  335.454468   798.000000    0.000000
33              162.639707  408.446672   810.000000    0.000000
34              189.431861  598.046630  1341.000000    0.000000
35              295.035633  619.014127  3914.000000    0.000000
```

[120]:
```
total_consumption_by_districtcode=MH_new.
↪groupby('District')['total_consumption'].sum()
```

[121]:
```
total_consumption_by_districtcode.sort_values(ascending=False).head(3)
```

[121]:
```
District
22    321553.491539
25    302202.890802
21    248508.550184
Name: total_consumption, dtype: float64
```

[122]:
```
MH_new.loc[:,"District"] = MH_new.loc[:,"District"].replace({22: "Mumbai␣
↪Suburban", 25: "Pune", 21: "Thane"})
```

[123]:
```
total_consumption_by_districtname=MH_new.
↪groupby('District')['total_consumption'].sum()
```

[124]:
```
total_consumption_by_districtname.sort_values(ascending=False).head(3)
```

[124]:
```
District
Mumbai Suburban    321553.491539
Pune               302202.890802
Thane              248508.550184
Name: total_consumption, dtype: float64
```

[125]:
```
from statsmodels.stats import weightstats as stests
```

[126]:
```
rural=MH_new[MH_new['Sector']=="RURAL"]
urban=MH_new[MH_new['Sector']=="URBAN"]
```

[127]:
```
rural.head()
```

[127]:
```
      state_1 District Sector  Region  ricetotal_v  wheattotal_v  jowarp_v  \
7577      MH    Thane  RURAL       1         91.0         100.0       0.0
7579      MH    Thane  RURAL       1          0.0           0.0       0.0
7580      MH    Thane  RURAL       1         84.0         120.0       0.0
7581      MH    Thane  RURAL       1         75.0         100.0       0.0
7582      MH    Thane  RURAL       1         70.0         100.0       0.0

      barleyp_v  maizep_v  maida_v  …  milletp_v  wheattotal_v  jowarp_v  \
7577        0.0       0.0      0.0  …        0.0         100.0       0.0
7579        0.0       0.0      0.0  …        0.0           0.0       0.0
7580        0.0       0.0      0.0  …        0.0         120.0       0.0
7581        0.0       0.0      0.0  …        0.0         100.0       0.0
7582        0.0       0.0      0.0  …        0.0         100.0       0.0

      barleyp_v  maizep_v  maida_v  suji_v  bajrap_v  milletp_v  \
7577        0.0       0.0      0.0     0.0       0.0        0.0
7579        0.0       0.0      0.0     0.0       0.0        0.0
7580        0.0       0.0      0.0     0.0       0.0        0.0
7581        0.0       0.0      0.0     8.0       0.0        0.0
7582        0.0       0.0      0.0     0.0       0.0        0.0

      total_consumption
7577              491.0
7579                0.0
7580              564.0
7581              507.0
7582              470.0

[5 rows x 22 columns]
```

[128]:
```
urban.head()
```

[128]:
```
      state_1 District Sector  Region  ricetotal_v  wheattotal_v  jowarp_v  \
74284     MH       24  URBAN       1        122.0    100.000000       0.0
74285     MH       24  URBAN       1        125.0     58.333333       0.0
74286     MH       24  URBAN       1        120.0     21.250000       0.0
74287     MH       24  URBAN       1         20.0      3.333333       0.0
74288     MH       24  URBAN       1        144.0     28.000000       0.0

       barleyp_v  maizep_v  maida_v  …  milletp_v  wheattotal_v  jowarp_v  \
74284        0.0       0.0      4.4  …        0.0    100.000000       0.0
74285        0.0       0.0      0.0  …        0.0     58.333333       0.0
74286        0.0       0.0      0.0  …        0.0     21.250000       0.0
74287        0.0       0.0      0.0  …        0.0      3.333333       0.0
74288        0.0       0.0      2.2  …        0.0     28.000000       0.0

       barleyp_v  maizep_v  maida_v  suji_v  bajrap_v  milletp_v  \
```

```
74284        0.0        0.0        4.4        4.8        0.0        0.0
74285        0.0        0.0        0.0        0.0        0.0        0.0
74286        0.0        0.0        0.0        0.0        0.0        0.0
74287        0.0        0.0        0.0        2.0        0.0        0.0
74288        0.0        0.0        2.2        0.0        0.0        0.0

       total_consumption
74284         558.800000
74285         358.333333
74286         205.000000
74287          41.333333
74288         264.800000

[5 rows x 22 columns]
```

[129]:
```
cons_rural=rural['total_consumption']
cons_urban=urban['total_consumption']
```

[130]:
```
z_statistic, p_value = stests.ztest(cons_rural, cons_urban)
# Print the z-score and p-value
print("Z-Score:", z_statistic)
print("P-Value:", p_value)
```

```
Z-Score: 2.767739911233802
P-Value: 0.005644648277714505
```

[131]:
```
#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 148.579142894145 and in Urban areas its 165.367499023822
```

[ ]:
```

```