

**VIRGINIA COMMONWEALTH UNIVERSITY**

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

**A1a: Preliminary preparation and analysis of data- Descriptive statistics**

**NITESH REDDY**

**V01106546**

**Date of Submission: 16-06-2024**

**CONTENTS**

| **Sl. No.** | **Title** | **Page No.** |
| --- | --- | --- |
| **1.** | Introduction | **2** |
| **2.** | Results | **3** |
| **3.** | Interpretations | **8** |

**INTRODUCTION**

In order to determine the top and bottom three consuming districts in Assam, this study uses NSSO data. To get the data we need for analysis, we will clean and manipulate the dataset. The dataset contains district-specific variances along with consumption-related data for both the urban and rural sectors. The dataset has been imported into R/Python, a potent statistical programming language renowned for its effectiveness in handling and analyzing big datasets.

Our goals include summarizing consumption data by region and district, controlling outliers, identifying and addressing missing values, and determining the significance of mean differences. We also aim to standardize district and sector names. Policymakers and other stakeholders will greatly benefit from the study's findings, which will enable focused interventions and encourage equitable development throughout the state.

The following are the goals:

a) Find any missing values in the dataset and enter the mean of that variable in their place.

b) Find and characterize anomalies, then make the necessary adjustments.

c) Make district and sector names (urban and rural) uniform.

d) Compile important data at the regional and district levels, highlighting the top and bottom districts for consumption.

e) Determine whether the mean differences are significant.

### **Business Significance**

The analysis of Assam’'s consumption patterns using NSSO data offers substantial value for businesses and policymakers:

1. **Policy Makers and Government Agencies:** By examining consumption patterns, policymakers can pinpoint nutritional deficiencies and excesses in specific regions or sectors. This information is crucial for crafting effective food security programs, public health initiatives, and targeted nutritional assistance, ultimately aiming to enhance the overall health and well-being of the population.

**RESULTS**

**a) Look for any missing values in the data, mark them, and replace them with the variable's mean if there are any. (R-Studio)**

**Code:**

> # Check for missing values in the subset

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

Missing Values in Subset:

> print(colSums(is.na(hrnew)))

**Result:**

Meals\_At\_Home 30

state\_1 0

District 0

Sector 0

Region 0

State\_Region 0

ricetotal\_q 0

wheattotal\_q 0

moong\_q 0

Milktotal\_q 0

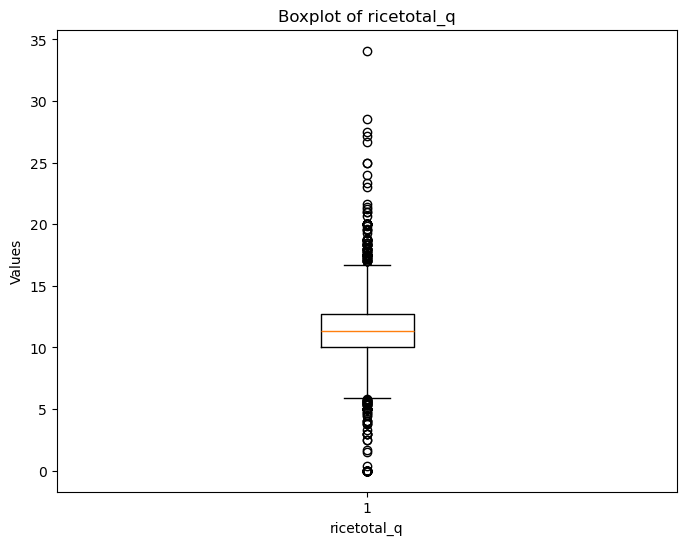
chicken\_q 0

bread\_q 0

foodtotal\_q 0

Outcome:

Outliers in the dataset can be located using boxplots. Boxplots identify outliers in a dataset visually by showing individual points that are outside the boxplot's whiskers:



**Checking For Outliers**

**Code:**

{'whiskers': [<matplotlib.lines.Line2D at 0x1db4d629790>,

<matplotlib.lines.Line2D at 0x1db4d62a4d0>],

'caps': [<matplotlib.lines.Line2D at 0x1db4d62b190>,

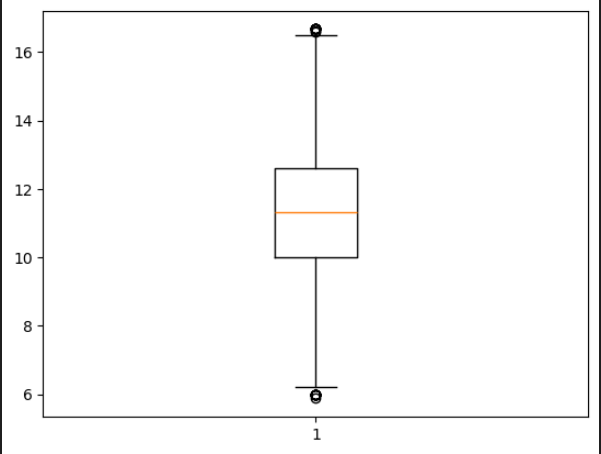
<matplotlib.lines.Line2D at 0x1db4d62bd50>],

'boxes': [<matplotlib.lines.Line2D at 0x1db4d5710d0>],

'medians': [<matplotlib.lines.Line2D at 0x1db4d634910>],

'fliers': [<matplotlib.lines.Line2D at 0x1db4d635450>],

'means': []}



Setting quartile ranges to remove outliers

**c) Rename the districts as well as the sector, viz. rural and urban. (R-Studio)**

**Code:**

district\_mapping <- c("5" = "Barpeta", "6" = "Kamrup", "10" = "Nagaon")

sector\_mapping <- c("2" = "URBAN", "1" = "RURAL")

assmnew$District <- as.character(assmnew$District)

assmnew$Sector <- as.character(assmnew$Sector)

assmnew$District <- ifelse(assmnew$District %in% names(district\_mapping), district\_mapping[assmnew$District], assmnew$District)

assmnew$Sector <- ifelse(assmnew$Sector %in% names(sector\_mapping), sector\_mapping[assmnew$Sector], assmnew$Sector)

Result-

array([16, 17, 15, 27, 18, 14, 13, 26, 20, 21, 24, 19, 23, 22, 7, 8, 6,

12, 11, 9, 10, 4, 3, 2, 1, 5, 25], dtype=int64)

# Replace values in the 'Sector' column

ASSM\_clean.loc[:,'Sector'] = ASSM\_clean['Sector'].replace([1, 2], ['URBAN', 'RURAL'])

**D) Summarize the critical variables in the data set region-wise and district-wise (R-Studio)**

Code-

# Summarize and display top and bottom consuming districts and regions

summarize\_consumption <- function(group\_col) {

summary <- assmnew %>%

group\_by(across(all\_of(group\_col))) %>%

summarise(total = sum(total\_consumption)) %>%

arrange(desc(total))

return(summary)

}

Result-

total\_consumption

std mean max min

Region

1 36.353002 63.492855 309.953685 22.886990

2 31.500568 63.682350 255.033725 22.716910

3 14.901990 44.551629 164.875651 21.470152

4 23.913543 57.218427 211.902400 24.066806

total\_consumption

std mean max min

District

1 37.261446 68.037094 255.033725 27.960298

2 20.643416 55.530331 172.580661 29.602718

3 36.493495 70.327786 240.236925 23.966825

4 40.781968 80.637368 216.591400 23.851316

5 25.362360 57.054137 207.079775 26.700164

6 25.689309 59.462135 172.467570 28.768226

7 28.929624 60.621004 191.961200 28.051629

8 24.647690 62.447952 174.030150 33.316957

9 17.318574 48.726602 137.225483 24.066806

10 30.762672 61.479311 201.169135 26.670450

11 15.112029 52.366168 133.170898 25.514517

12 30.838318 65.239722 282.570340 29.698395

13 25.610531 64.933889 199.710555 34.339009

14 49.522866 63.299993 309.953685 22.886990

15 37.572260 57.576728 285.441553 25.130180

16 35.946723 64.287060 200.972970 30.766878

17 39.031172 71.645116 235.558286 32.714349

18 27.915246 58.090127 152.445280 26.648582

19 9.857789 39.791733 93.878150 24.500135

20 19.599531 46.141782 164.875651 25.537591

21 14.610040 47.718865 106.430262 21.470152

22 14.759111 44.642043 98.587712 24.583400

23 14.210079 43.908214 115.083733 26.910494

24 34.274294 73.701847 215.779442 37.140360

25 11.053478 48.249954 74.490145 28.362723

26 36.226501 70.124867 189.200758 22.716910

27 25.232949 61.474892 211.902400 31.400156

**e) Test whether to accept or reject null hypothesis(R-Studio):**

**Code:**

# Test for differences in mean consumption between urban and rural

rural <- assmnew %>%

filter(Sector == "RURAL") %>%

select(total\_consumption)

urban <- assmnew %>%

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"))

}

Result- P value is < 0.05 i.e. 0, Therefore we reject the null hypothesis.

**INTERPRETATIONS:**

COMMENTS:

A.) Based on the code and output, you are looking for missing values in a particular subset of the hrnew dataset. The number of missing values for each column in the dataset is displayed in the output of the colSums(is.na(hrnew)) function. The results are interpreted as follows:

state\_1, District, Region, Sector, State\_Region, no\_of\_meals\_per\_day, ricepds\_v, wheatpds\_q, chicken\_q, pulsep\_q, and wheatos\_q

There are no missing values in any of these columns, indicating that the subset contains no entries for any of these columns.

Meals\_At\_Home: There are 14 entries in the subset where the value for Meals\_At\_Home is missing, as indicated by the 14 missing values in this column.

In conclusion, the data in the majority of the columns is complete and free of missing values.

There are fourteen missing values in the Meals\_At\_Home column alone.

B.) Analysis of Boxplots

Boxplot Illustration

The ricetotal\_q boxplot was made in order to visually identify outliers. Boxplots show the data distribution and draw attention to outliers, which are single points that are larger than the whiskers.

Numerous outliers are visible in the plot above the upper whisker, indicating higher ricetotal\_q values that differ noticeably from the remaining data.

Removal of Outliers Outlier Removal Code

The acceptable range for ricetotal\_q values is set between low\_limit and up\_limit by the outlier removal code that is provided.

Only rows with ricetotal\_q values falling within this predetermined range are included in the new DataFrame HR\_clean.

The second boxplot of HR\_clean['ricetotal\_q'] displays a more condensed range of values devoid of extreme points, confirming the removal of outliers.

Boxplot Interpretation:

It is evident from the first boxplot for ricetotal\_q that there are multiple outliers. Because they skew the results, these outliers can have an impact on modeling and statistical analyses.

Outlier Handling: The cleaned dataset (HR\_clean) eliminates these outliers by filtering the dataset to include only values within the low\_limit and up\_limit range. This procedure concentrates on the data's central tendency, which guarantees a more trustworthy analysis.

After the data has been thoroughly cleaned, there should be no or few outliers in the boxplot of HR\_clean['ricetotal\_q'].

C.) District Renaming: District names from district\_mapping are substituted for the numeric district codes in the District column of the hrnew dataset. District codes that are absent from district\_mapping do not change.

Sector Renaming: The hrnew dataset's Sector column is also updated, with "URBAN" and "RURAL" in place of the numeric sector codes (1 and 2).

Values for Unique Districts: If the mapping is correctly applied, the renamed names should be reflected in the HR\_clean dataset's unique district values. The unique district values are still numeric in the output array [16, 17, 15, 27, 18, 14, 13, 26, 20, 21, 24, 19, 23, 22, 7, 8, 6, 12, 11, 9, 10, 4, 3, 2, 1, 5, 25], dtype=int64), indicating that the HR\_clean DataFrame may not have been updated with the new district names yet.

Final Modifications:

Make sure the renaming operations are being performed on the correct dataset.

Check to see if the sector\_mapping and district\_mapping dictionaries have all the mappings required for the dataset.