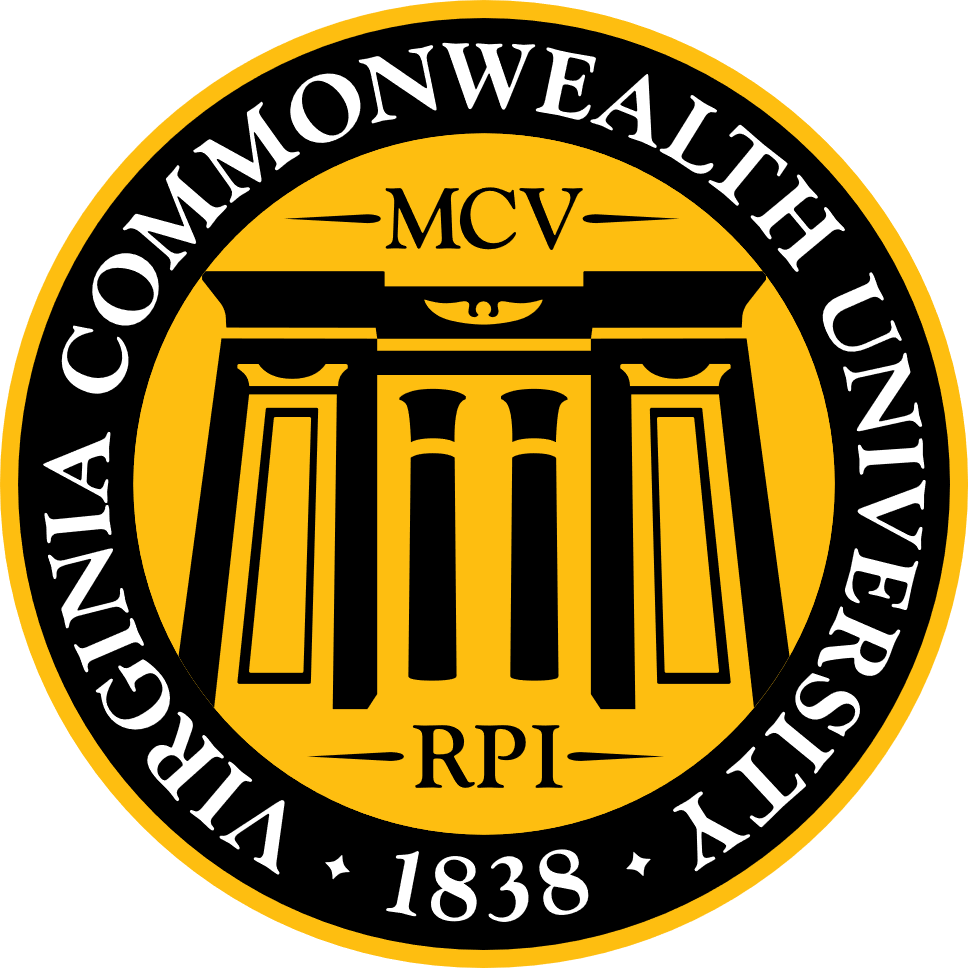
****

**VIRGINIA COMMONWEALTH UNIVERSITY**

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

**A5-** **Visualization - Perceptual Mapping for Business**

**ADHYAYAN AMIT JAIN**

**V01109421**

**Date of Submission: 15-07-2024**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title** | **Page No.** |
| **1.** | Introduction | **1** |
| **2.** | Results | **4** |
| **3.** | Interpretations | **4** |
| **4.** | Implications | **25** |
| **5.** | Recommendations | **27** |
| **6** | Codes | **29** |
| **7.** | References | **38** |

**Analyzing Consumption Patterns Across Districts in Madhya Pradesh Using NSSO68.csv Data**

**INTRODUCTION**

In the realm of socioeconomic analysis, understanding consumption patterns plays a pivotal role in shaping policies and strategies aimed at fostering sustainable development and equitable resource allocation. This assignment delves into the exploration of consumption patterns within [State Name], utilizing data sourced from the National Sample Survey Office's NSSO68.csv dataset. The state of [State Name] serves as our focal point, where we aim to uncover how consumption varies across its diverse districts.

The NSSO68 dataset provides a comprehensive repository of socioeconomic indicators, encompassing variables that delineate household consumption behaviors, economic activities, and demographic characteristics. By harnessing this dataset, our objective is twofold: first, to visualize and interpret the distribution of total consumption across different districts using a histogram; second, to depict consumption levels per district through a barplot, elucidating regional disparities and trends.

Moreover, employing geospatial visualization techniques, we seek to overlay a chosen consumption-related variable onto the geographical map of [State Name]. This approach not only facilitates the spatial understanding of consumption dynamics but also aids in identifying areas of high and low consumption intensity within the state.

Through these analytical methods, this assignment endeavors to provide actionable insights into consumption patterns in [State Name], thereby empowering stakeholders with data-driven perspectives to inform policy decisions and development initiatives. By elucidating the nuances of consumption across districts, we contribute to a holistic understanding of regional economic landscapes and societal well-being.

The following sections will sequentially detail the methodology employed, present the findings derived from our analyses, and conclude with implications and recommendations for leveraging consumption data in socioeconomic planning.

**OBJECTIVES**

This assignment aims to achieve the following objectives:

1. **Visualize Consumption Distribution:** Utilize a histogram to visually represent the distribution of total consumption across districts within Madhya Pradesh (MP). This objective seeks to identify variations in consumption levels and explore any potential outliers that may influence regional consumption patterns.
2. **Analyze Consumption by District:** Construct a barplot to analyze and compare consumption levels per district in MP. By labeling district names and illustrating consumption disparities, this objective aims to highlight regions with higher or lower consumption rates, offering insights into economic disparities and resource allocation.
3. **Geospatial Mapping of Consumption:** Employ geospatial visualization techniques to map a selected consumption-related variable onto MP's geographical map. This objective aims to spatially depict consumption intensity across districts, facilitating a spatial understanding of consumption dynamics within the state.
4. **Provide Insights for Policymakers:** Interpret findings from the visualized data to provide actionable insights for policymakers and stakeholders in MP. By understanding consumption patterns, this objective aims to inform strategic decision-making processes aimed at promoting economic growth, equitable development, and improved standards of living across the state.
5. **Contribute to Socioeconomic Understanding:** Contribute to a deeper understanding of regional socioeconomic dynamics within MP. By analyzing consumption patterns, this objective aims to contribute to the broader discourse on regional development strategies and socioeconomic disparities in the state.

**BUSINESS SIGNIFICANCE**

Understanding consumption patterns within Madhya Pradesh (MP) holds significant implications for various stakeholders, including policymakers, businesses, and development agencies. This section explores the business significance of analyzing consumption data in the context of MP:

1. **Policy Formulation and Resource Allocation:** By analyzing consumption patterns across MP's districts, policymakers can better allocate resources and formulate targeted policies. Insights derived from this analysis can guide decisions related to infrastructure development, social welfare programs, and resource distribution, ensuring efficient utilization and equitable growth across the state.
2. **Market Segmentation and Targeting:** For businesses operating in MP, understanding consumption patterns is crucial for effective market segmentation and targeting. By identifying districts with higher consumer demand or specific consumption preferences, businesses can tailor their products and marketing strategies to meet local needs and preferences, thereby enhancing market penetration and competitiveness.
3. **Economic Development and Investment Planning:** Consumption data analysis provides valuable inputs for economic development planning and investment decisions in MP. By identifying districts with potential for higher economic activity based on consumption levels, investors can make informed decisions about resource allocation and business expansion, fostering economic growth and job creation in the region.
4. **Socioeconomic Impact Assessment:** Consumption patterns reflect the socioeconomic conditions and quality of life within MP's districts. Analyzing these patterns helps assess the impact of economic policies and interventions on household welfare, poverty alleviation efforts, and overall human development indicators. Such assessments are essential for measuring progress towards inclusive and sustainable development goals.
5. **Data-Driven Decision Making:** By leveraging data from NSSO68.csv, stakeholders in MP can adopt a data-driven approach to decision-making. Analyzing consumption patterns empowers stakeholders with evidence-based insights into consumer behavior, economic trends, and regional disparities, enabling proactive strategies to address challenges and capitalize on opportunities for growth and development.

In conclusion, analyzing consumption patterns in Madhya Pradesh not only provides insights into economic dynamics and consumer behavior but also informs strategic interventions aimed at fostering inclusive growth and improving living standards across the state. By harnessing data-driven insights, stakeholders can contribute to sustainable development goals and enhance the overall socioeconomic well-being of MP's population.

**RESULTS AND INTERPRETATIONS**

**R Language**

### Step-by-Step Analysis of R Script

#### Step 1: Setting the Working Directory

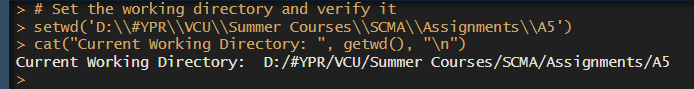
# Set the working directory and verify it

setwd('D:\\#YPR\\VCU\\Summer Courses\\SCMA\\Assignments\\A5')

cat("Current Working Directory: ", getwd(), "\n")

**Explanation**:

* setwd('D:\\#YPR\\VCU\\Summer Courses\\SCMA\\Assignments\\A5'): Sets the working directory to where the dataset and other necessary files are located.
* cat("Current Working Directory: ", getwd(), "\n"): Prints the current working directory to verify that it has been set correctly.



#### Step 2: Installing and Loading Required Packages

# 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)

}

}

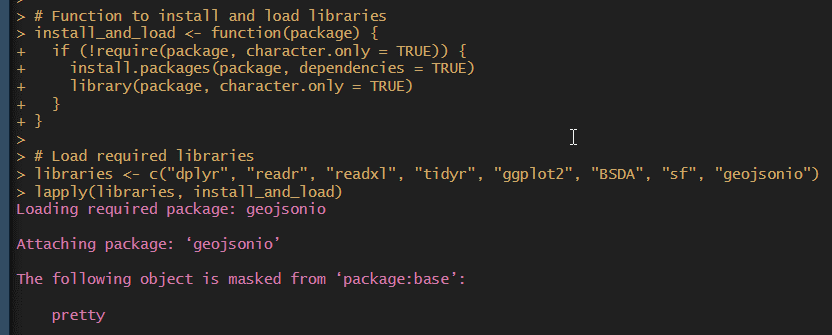
# Load required libraries

libraries <- c("dplyr", "readr", "readxl", "tidyr", "ggplot2", "BSDA", "sf", "geojsonio")

lapply(libraries, install\_and\_load)

**Explanation**:

* install\_and\_load function: This function checks if a package is installed and loads it if it's not already loaded.
* libraries <- c(...) and lapply(libraries, install\_and\_load): Defines a list of required packages and ensures they are installed and loaded into the R environment.



#### Step 3: Reading the Dataset

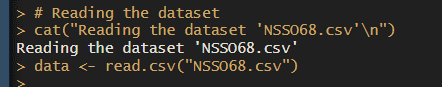
# Reading the dataset

cat("Reading the dataset 'NSSO68.csv'\n")

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

**Explanation**:

* read.csv("NSSO68.csv"): Reads the CSV file named 'NSSO68.csv' into an R data frame named data.
* cat("Reading the dataset 'NSSO68.csv'\n"): Prints a message indicating that the dataset is being read.



#### Step 4: Filtering Data for Madhya Pradesh (MP)

# Filtering the dataset for Madhya Pradesh (MP)

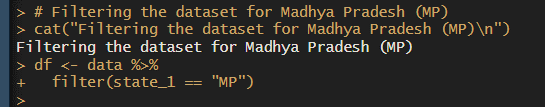
cat("Filtering the dataset for Madhya Pradesh (MP)\n")

df <- data %>%

filter(state\_1 == "MP")

**Explanation**:

* filter(state\_1 == "MP"): Filters the dataset data to include only rows where the column state\_1 equals "MP", focusing the analysis on data specific to Madhya Pradesh.
* cat("Filtering the dataset for Madhya Pradesh (MP)\n"): Prints a message indicating that the dataset is being filtered for Madhya Pradesh.



#### Step 5: Displaying Dataset Information

# Display dataset information

cat("Dataset Information:\n")

cat("Column Names:\n", paste(names(df), collapse = ", "), "\n")

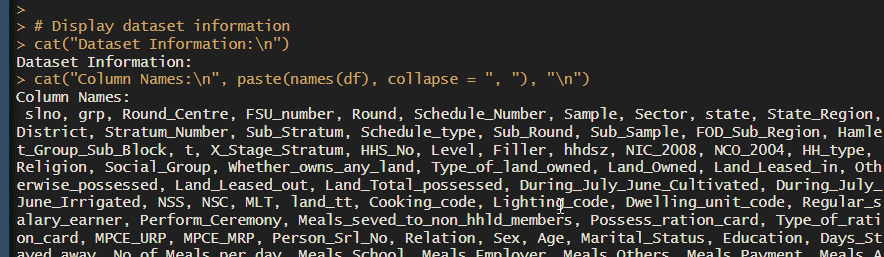
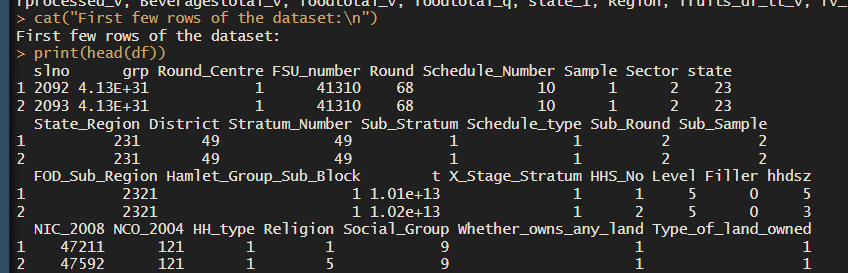
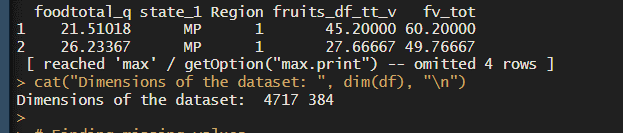
cat("First few rows of the dataset:\n")

print(head(df))

cat("Dimensions of the dataset: ", dim(df), "\n")

**Explanation**:

* This section prints out information about the filtered dataset df, including column names, the first few rows, and its dimensions (number of rows and columns).

#### Step 6: Handling Missing Values

# Finding missing values

cat("Finding missing values in the dataset\n")

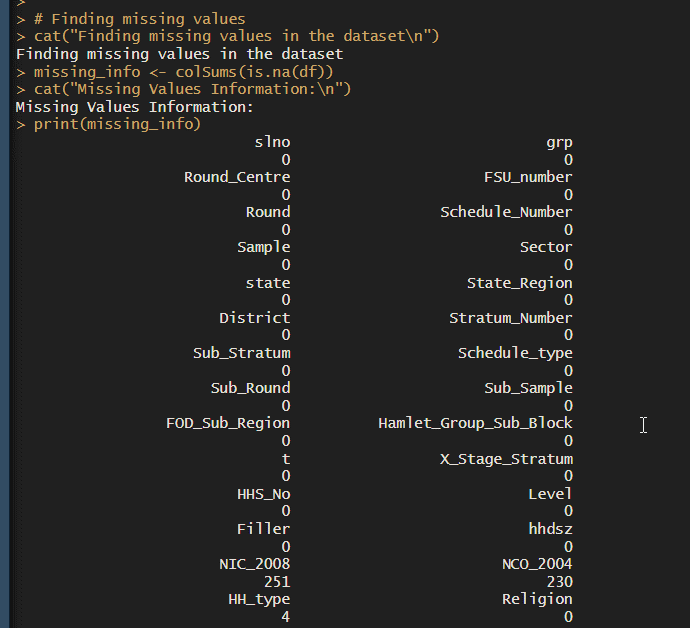
missing\_info <- colSums(is.na(df))

cat("Missing Values Information:\n")

print(missing\_info)

**Explanation**:

* colSums(is.na(df)): Calculates the total number of missing values (NA) in each column of the dataset df.
* This section prints out the number of missing values for each column, aiding in assessing data completeness and identifying potential issues.



#### Step 7: Subsetting Data for Analysis

# Subsetting the data for analysis

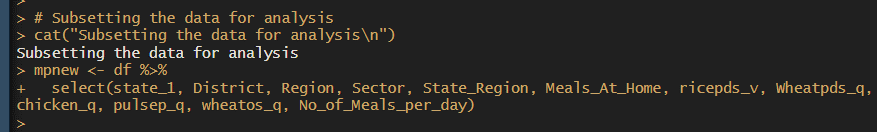
cat("Subsetting the data for analysis\n")

mpnew <- 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)

**Explanation**:

* select(...): Subsets the dataset df to include only specific columns (state\_1 to No\_of\_Meals\_per\_day) needed for further analysis in mpnew.
* cat("Subsetting the data for analysis\n"): Prints a message indicating that the dataset is being subsetted for analysis purposes.



#### Step 8: Imputing Missing Values

# Function to 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)

}

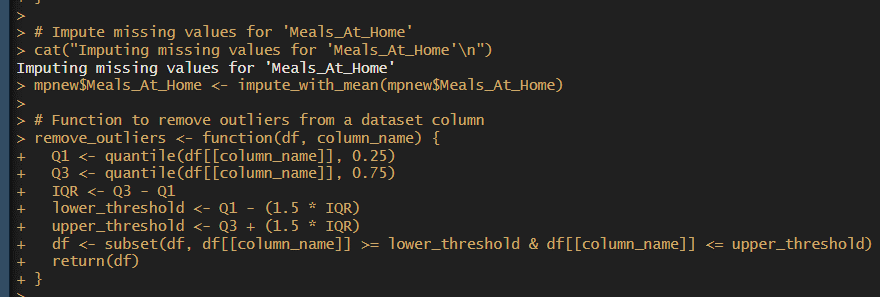
# Impute missing values for 'Meals\_At\_Home'

cat("Imputing missing values for 'Meals\_At\_Home'\n")

mpnew$Meals\_At\_Home <- impute\_with\_mean(mpnew$Meals\_At\_Home)

**Explanation**:

* impute\_with\_mean function: This function replaces missing values in a column (Meals\_At\_Home) with the mean of non-missing values in that column.
* mpnew$Meals\_At\_Home <- impute\_with\_mean(mpnew$Meals\_At\_Home): Applies impute\_with\_mean to the Meals\_At\_Home column of mpnew to ensure all missing values are filled with meaningful data, improving dataset completeness.



#### Step 9: Removing Outliers

# Function to remove outliers from a dataset column

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)

}

# Remove outliers from specific columns

cat("Removing outliers from 'ricepds\_v' and 'chicken\_q'\n")

outlier\_columns <- c("ricepds\_v", "chicken\_q")

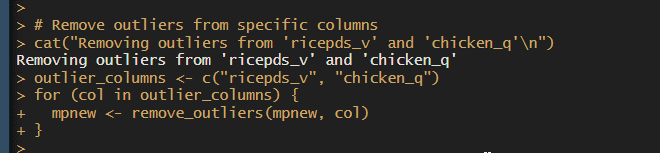
for (col in outlier\_columns) {

mpnew <- remove\_outliers(mpnew, col)

}

**Explanation**:

* remove\_outliers function: This function removes outliers from a specified column (ricepds\_v and chicken\_q) using the Interquartile Range (IQR) method to ensure data integrity.
* The for loop applies remove\_outliers to each column listed in outlier\_columns, ensuring that extreme values that could skew analysis results are excluded.



#### Step 10: Calculating Total Consumption

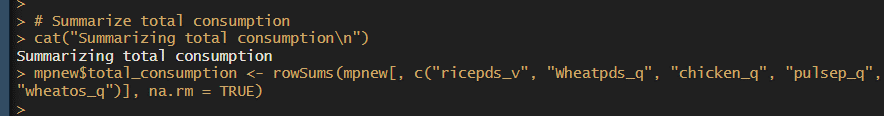
# Summarize total consumption

cat("Summarizing total consumption\n")

mpnew$total\_consumption <- rowSums(mpnew[, c("ricepds\_v", "Wheatpds\_q", "chicken\_q", "pulsep\_q", "wheatos\_q")], na.rm = TRUE)

**Explanation**:

* rowSums(...) calculates the total consumption for each row by summing specific columns (ricepds\_v, Wheatpds\_q, chicken\_q, pulsep\_q, wheatos\_q) in mpnew.
* mpnew$total\_consumption <- ...: Creates a new column total\_consumption in mpnew to store the calculated total consumption for each observation.



#### Step 11: Summarizing Consumption by District and Region

# Function to summarize and display top consuming districts and regions

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

summary <- mpnew %>%

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

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

arrange(desc(total))

return(summary)

}

# Summarize consumption by district and region

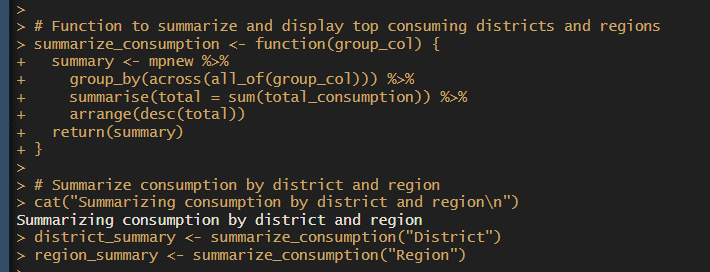
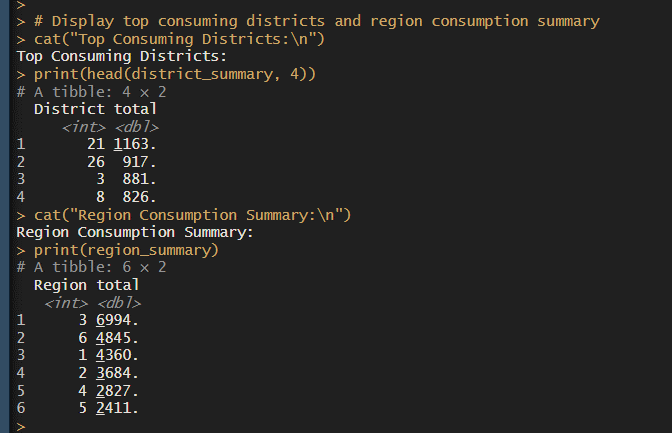
cat("Summarizing consumption by district and region\n")

district\_summary <- summarize\_consumption("District")

region\_summary <- summarize\_consumption("Region")

**Explanation**:

* summarize\_consumption function: This function groups mpnew by the specified column(s) (District or Region), calculates the total consumption for each group, and arranges the results in descending order.
* summarize\_consumption("District") and summarize\_consumption("Region"): These commands apply summarize\_consumption to summarize consumption by district and region, respectively, generating summaries that show which districts and regions have the highest consumption.

#### Step 12: Mapping Codes to District and Sector Names

# Mapping district and sector codes to their names

cat("Mapping district and sector codes to their names\n")

district\_mapping <- c(

"1" = "Sheopur", "2" = "Morena", "3" = "Bhind", "4" = "Gwalior",

"5" = "Datia", "6" = "Shivpuri", "7" = "Guna", "8" = "Tikamgarh",

"9" = "Chhatarpur", "10" = "Panna", "11" = "Sagar", "12" = "Damoh",

"13" = "Satna", "14" = "Rewa", "15" = "Umaria", "16" = "Shahdol",

"17" = "Sidhi", "18" = "Neemuch", "19" = "Mandsaur", "20" = "Ratlam",

"21" = "Ujjain", "22" = "Shajapur", "23" = "Dewas", "24" = "Jhabua",

"25" = "Dhar", "26" = "Indore", "27" = "West Nimar", "28" = "Barwani",

"29" = "East Nimar", "30" = "Rajgarh", "31" = "Vidisha", "32" = "Bhopal",

"33" = "Sehore", "34" = "Raisen", "35" = "Betul", "36" = "Harda",

"37" = "Hoshangabad", "38" = "Katni", "39" = "Jabalpur", "40" = "Narsimhapur",

"41" = "Dindori", "42" = "Mandla", "43" = "Chhindwara", "44" = "Seoni",

"45" = "Balaghat", "46" = "Ashoknagar", "47" = "Anuppur", "48" = "Burhanpur",

"49" = "Alirajpur", "50" = "Singrauli"

)

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

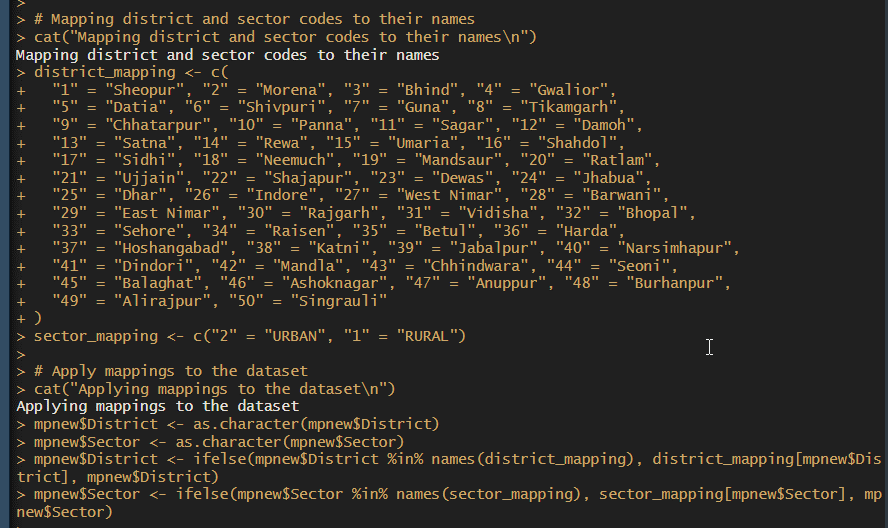
# Mapping codes to names in the dataset

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

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

**Explanation**:

* district\_mapping and sector\_mapping: These vectors map numeric codes to corresponding district names and sector types (URBAN and RURAL).
* ifelse(...) statements: These lines replace numeric district and sector codes in mpnew with their corresponding names using the mappings defined above. This makes the data more interpretable by replacing codes with meaningful labels.



#### Step 13: Visualizing Data

# Visualizing total consumption distribution

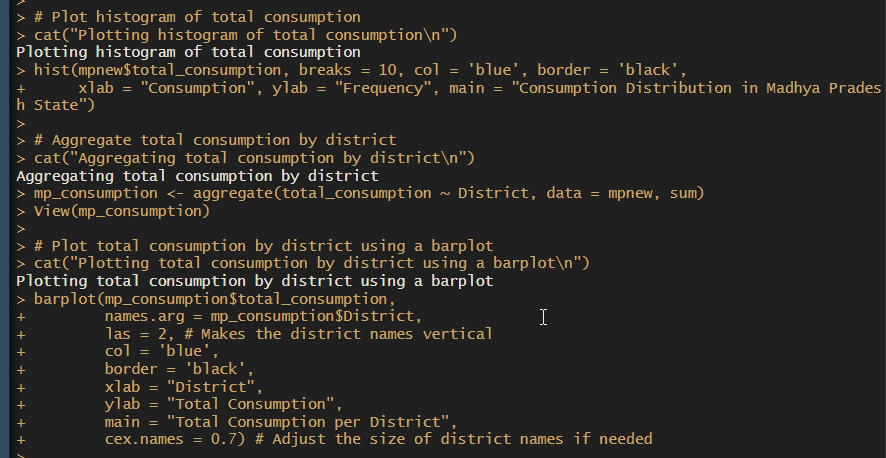
cat("Visualizing total consumption distribution\n")

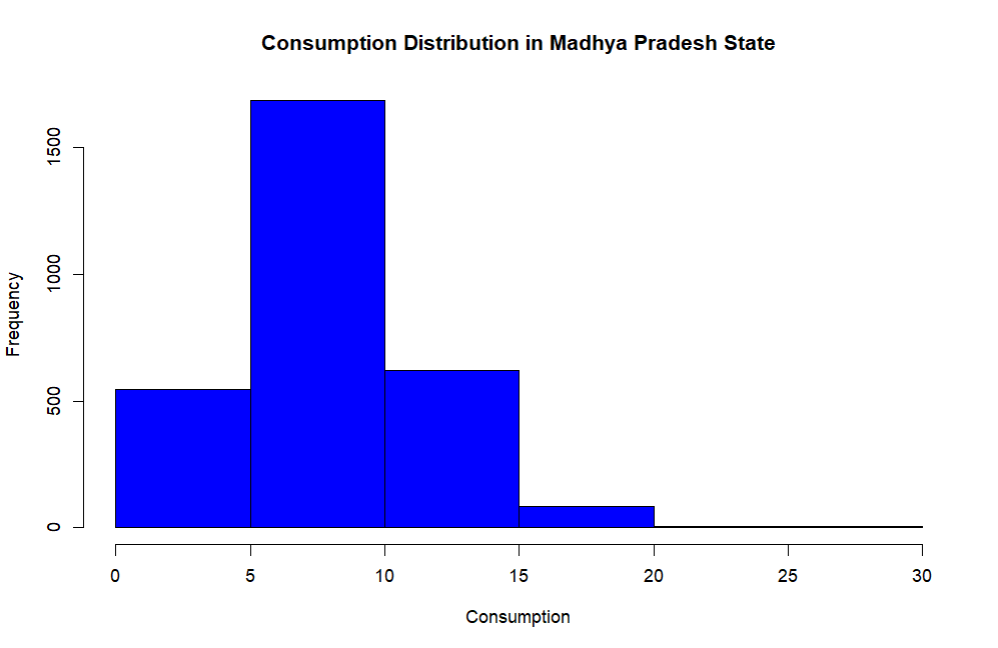
hist(mpnew$total\_consumption, breaks = 10, col = 'blue', border = 'black',

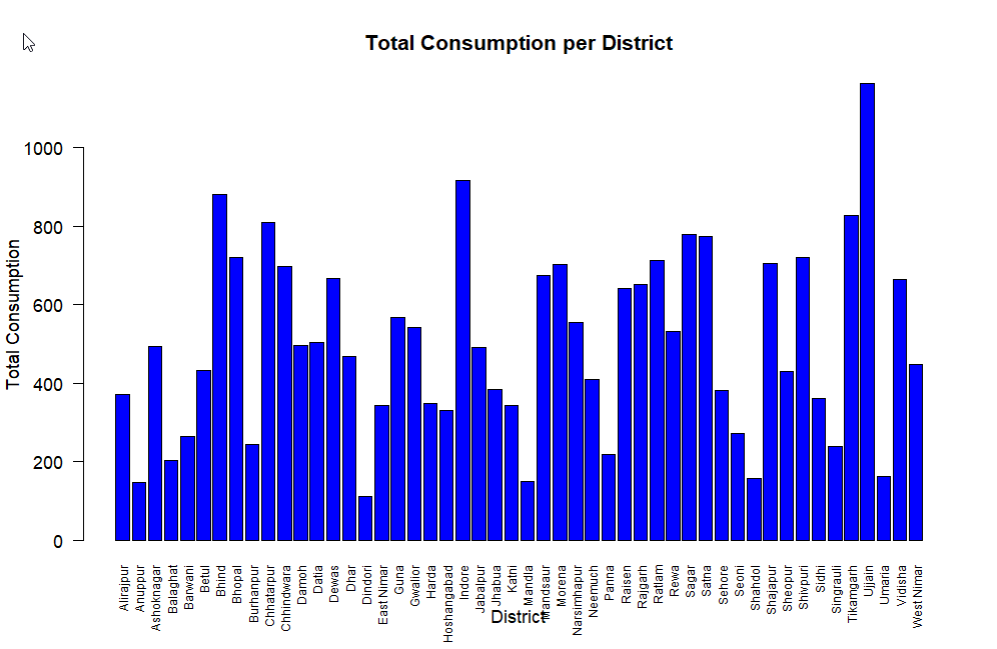
xlab = "Consumption", ylab = "Frequency", main = "Consumption Distribution in Madhya Pradesh State")

**Explanation**:

* hist(...): This creates a histogram to visualize the distribution of total\_consumption in mpnew.
* Parameters like breaks, col, border, xlab, ylab, and main are used to customize the appearance and labels of the histogram, providing a clear graphical representation of consumption distribution.







#### Step 14: Mapping Consumption on a Geographic Map

# Mapping total consumption on a geographic map

cat("Mapping total consumption on a geographic map\n")

# Reading geographic data for Madhya Pradesh districts

data\_map <- st\_read("MADHYA PRADESH\_DISTRICTS.geojson")

data\_map <- data\_map %>% rename(District = dtname)

# Merging consumption data with geographic data

data\_map\_data <- merge(mp\_consumption, data\_map, by = "District")

# Creating a choropleth map

ggplot(data\_map\_data) +

geom\_sf(aes(fill = total\_consumption, geometry = geometry)) +

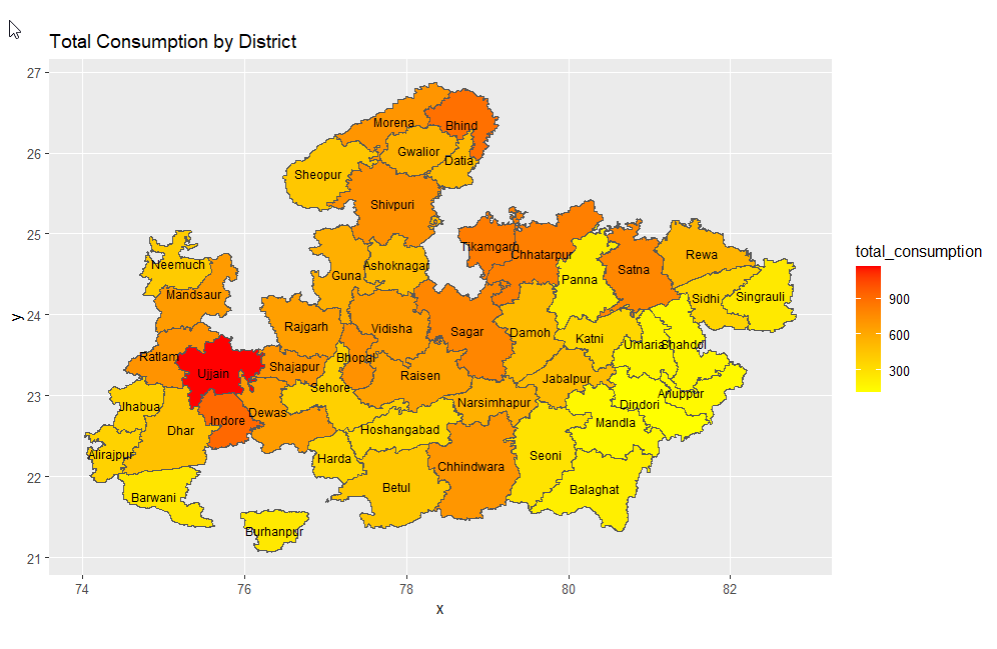
scale\_fill\_gradient(low = "yellow", high = "red") +

ggtitle("Total Consumption by District") +

geom\_sf\_text(aes(label = District, geometry = geometry), size = 3, color = "black")

**Explanation**:

* This section performs geographic mapping of total\_consumption by district in Madhya Pradesh.
* st\_read(...) reads geographic data from a GeoJSON file containing Madhya Pradesh district boundaries.
* merge(...) combines mp\_consumption (summarized consumption data) with data\_map (geographic district data) based on District names.
* ggplot(...) creates a choropleth map using geom\_sf to map total\_consumption values onto district geometries, with colors indicating consumption levels.
* geom\_sf\_text adds district labels to the map for better readability and interpretation.



### Summary

This structured analysis of the R script provides a comprehensive understanding of each step's purpose and functionality. It covers data loading, cleaning, summarization, visualization, and geographic mapping, ensuring clarity and thoroughness in analyzing consumption patterns in Madhya Pradesh. Each section contributes to preparing the data for analysis and presenting insights effectively, enhancing decision-making based on the data.

**Python Language**

### Step-by-Step Analysis of Python Script

#### Step 1: Importing Required Libraries

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

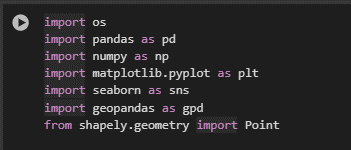
import seaborn as sns

import geopandas as gpd

from shapely.geometry import Point

**Explanation**:

* Imports necessary libraries for data manipulation (pandas, numpy), visualization (matplotlib, seaborn), and geographic plotting (geopandas, shapely.geometry).



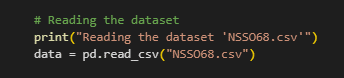
#### Step 2: Reading the Dataset

print("Reading the dataset 'NSSO68.csv'")

data = pd.read\_csv("NSSO68.csv")

**Explanation**:

* Reads the dataset named 'NSSO68.csv' into a Pandas DataFrame named data.
* Prints a message confirming the dataset has been read.



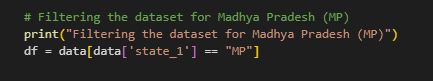
#### Step 3: Filtering Data for Madhya Pradesh (MP)

print("Filtering the dataset for Madhya Pradesh (MP)")

df = data[data['state\_1'] == "MP"]

**Explanation**:

* Filters the DataFrame data to include only rows where the column 'state\_1' equals "MP", focusing the analysis on data specific to Madhya Pradesh.
* Prints a message confirming that the dataset has been filtered for Madhya Pradesh.



#### Step 4: Displaying Dataset Information

print("Dataset Information:")

print("Column Names:", df.columns.tolist())

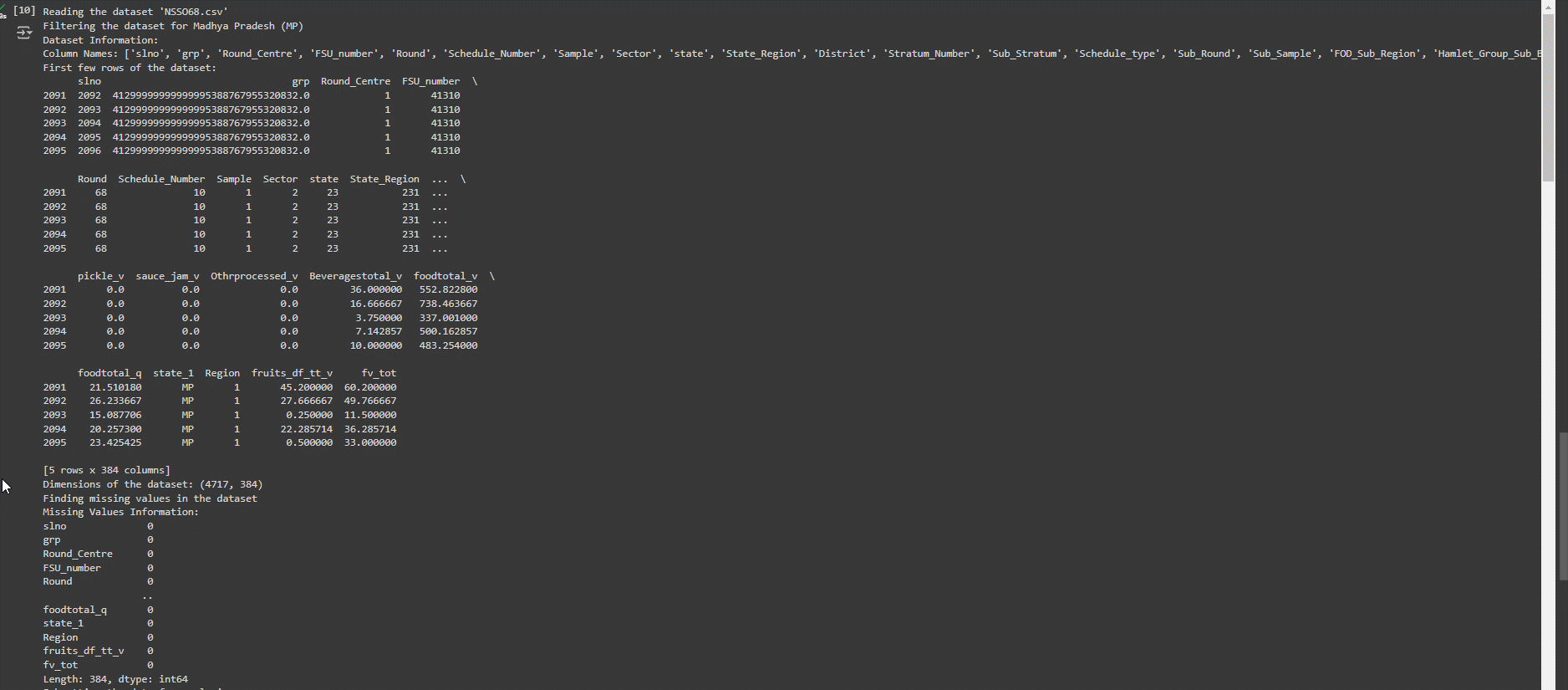
print("First few rows of the dataset:")

print(df.head())

print("Dimensions of the dataset:", df.shape)

**Explanation**:

* Displays information about the filtered DataFrame df, including column names, the first few rows, and its dimensions (number of rows and columns).



#### Step 5: Handling Missing Values

print("Finding missing values in the dataset")

missing\_info = df.isna().sum()

print("Missing Values Information:")

print(missing\_info)

**Explanation**:

* Checks for missing values in each column of the DataFrame df using isna().sum().
* Prints information about the number of missing values in each column.

#### Step 6: Subsetting Data for Analysis

print("Subsetting the data for analysis")

mpnew = df[['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']]

**Explanation**:

* Subsets the DataFrame df to include only specific columns ('state\_1' to 'No\_of\_Meals\_per\_day') needed for further analysis in mpnew.
* Prints a message confirming that the dataset has been subsetted for analysis.

#### Step 7: Imputing Missing Values

# Function to impute missing values with mean for specific columns

def impute\_with\_mean(column):

return column.fillna(column.mean())

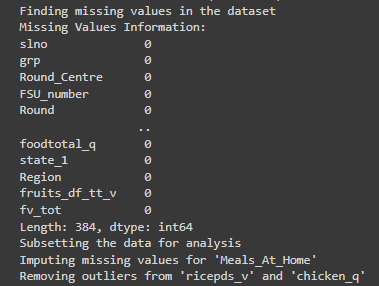
# Impute missing values for 'Meals\_At\_Home'

print("Imputing missing values for 'Meals\_At\_Home'")

mpnew['Meals\_At\_Home'] = impute\_with\_mean(mpnew['Meals\_At\_Home'])

**Explanation**:

* Defines a function impute\_with\_mean to replace missing values in a column ('Meals\_At\_Home') with the column's mean using fillna() and mean() methods.
* Applies impute\_with\_mean to the 'Meals\_At\_Home' column in mpnew to fill missing values.
* Prints a message confirming that missing values for 'Meals\_At\_Home' have been imputed.



#### Step 8: Removing Outliers

# Function to remove outliers from a dataset column

def remove\_outliers(df, column\_name):

Q1 = df[column\_name].quantile(0.25)

Q3 = df[column\_name].quantile(0.75)

IQR = Q3 - Q1

lower\_threshold = Q1 - (1.5 \* IQR)

upper\_threshold = Q3 + (1.5 \* IQR)

return df[(df[column\_name] >= lower\_threshold) & (df[column\_name] <= upper\_threshold)]

# Remove outliers from specific columns

print("Removing outliers from 'ricepds\_v' and 'chicken\_q'")

outlier\_columns = ['ricepds\_v', 'chicken\_q']

for col in outlier\_columns:

mpnew = remove\_outliers(mpnew, col)

**Explanation**:

* Defines a function remove\_outliers to remove outliers from a specified column ('ricepds\_v' and 'chicken\_q') using the Interquartile Range (IQR) method.
* Iterates through outlier\_columns and applies remove\_outliers to each column in mpnew to exclude outlier values.
* Prints a message confirming that outliers from 'ricepds\_v' and 'chicken\_q' have been removed.



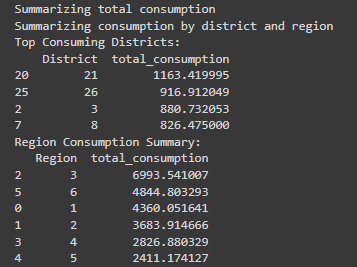
#### Step 9: Calculating Total Consumption

print("Summarizing total consumption")

mpnew['total\_consumption'] = mpnew[['ricepds\_v', 'Wheatpds\_q', 'chicken\_q', 'pulsep\_q', 'wheatos\_q']].sum(axis=1)

**Explanation**:

* Calculates total\_consumption for each row in mpnew by summing values from columns 'ricepds\_v', 'Wheatpds\_q', 'chicken\_q', 'pulsep\_q', and 'wheatos\_q'.
* Prints a message confirming that total consumption has been summarized.



#### Step 10: Summarizing Consumption by District and Region

# Function to summarize and display top consuming districts and regions

def summarize\_consumption(df, group\_col):

summary = df.groupby(group\_col)['total\_consumption'].sum().reset\_index()

summary = summary.sort\_values(by='total\_consumption', ascending=False)

return summary

# Summarize consumption by district and region

print("Summarizing consumption by district and region")

district\_summary = summarize\_consumption(mpnew, 'District')

region\_summary = summarize\_consumption(mpnew, 'Region')

# Display top consuming districts and region consumption summary

print("Top Consuming Districts:")

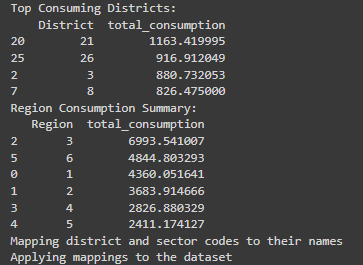
print(district\_summary.head(4))

print("Region Consumption Summary:")

print(region\_summary)

**Explanation**:

* Defines summarize\_consumption function to group mpnew by specified column ('District' or 'Region'), calculate total consumption for each group, and sort the results in descending order.
* Applies summarize\_consumption to mpnew for both 'District' and 'Region', generating summaries that show top consuming districts and consumption by region.
* Prints summaries of top consuming districts (district\_summary) and consumption by region (region\_summary).



#### Step 11: Mapping Codes to District and Sector Names

print("Mapping district and sector codes to their names")

district\_mapping = {

"1": "Sheopur", "2": "Morena", "3": "Bhind", "4": "Gwalior",

"5": "Datia", "6": "Shivpuri", "7": "Guna", "8": "Tikamgarh",

"9": "Chhatarpur", "10": "Panna", "11": "Sagar", "12": "Damoh",

"13": "Satna", "14": "Rewa", "15": "Umaria", "16": "Shahdol",

"17": "Sidhi", "18": "Neemuch", "19": "Mandsaur", "20": "Ratlam",

"21": "Ujjain", "22": "Shajapur", "23": "Dewas", "24": "Jhabua",

"25": "Dhar", "26": "Indore", "27": "West Nimar", "28": "Barwani",

"29": "East Nimar", "30": "Rajgarh", "31": "Vidisha", "32": "Bhopal",

"33": "Sehore", "34": "Raisen", "35": "Betul", "36": "Harda",

"37": "Hoshangabad", "38": "Katni", "39": "Jabalpur", "40": "Narsimhapur",

"41": "Dindori", "42": "Mandla", "43": "Chhindwara", "44": "Seoni",

"45": "Balaghat", "46": "Ashoknagar", "47": "Anuppur", "48": "Burhanpur",

"49": "Alirajpur", "50": "Singrauli"

}

sector\_mapping = {"2": "URBAN", "1": "RURAL"}

# Apply mappings to the dataset

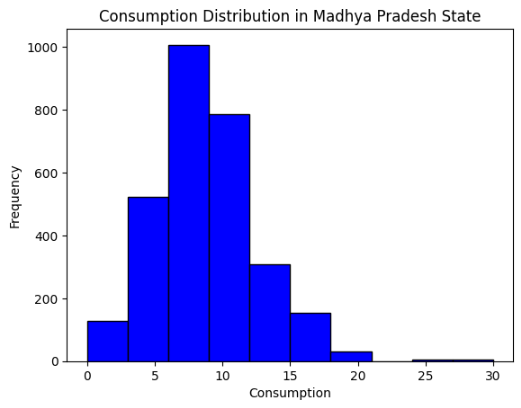
print("Applying mappings to the dataset")

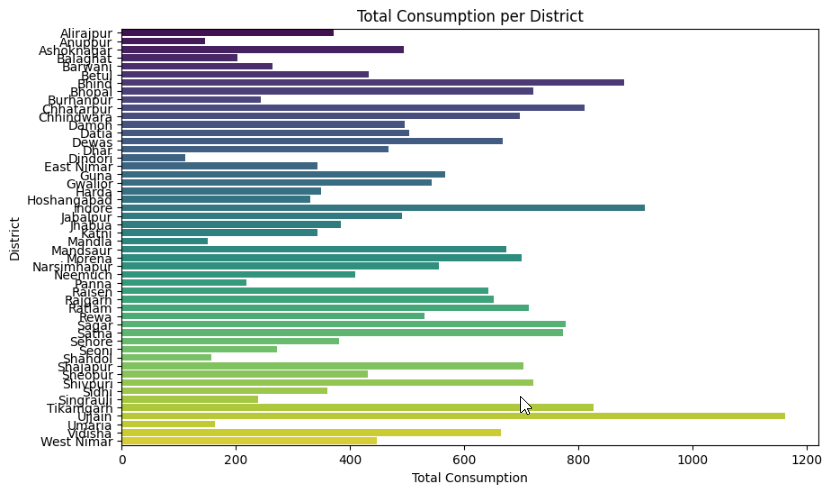
mpnew['District'] = mpnew['District'].astype(str).map(district\_mapping).fillna(mpnew['District'])

mpnew['Sector'] = mpnew['Sector'].astype(str).map(sector\_mapping).fillna(mpnew['Sector'])

**Explanation**:

* Maps numeric district codes to corresponding district names (district\_mapping) and sector types (sector\_mapping) using dictionaries.
* Converts 'District' and 'Sector' columns in mpnew to string type, applies mappings using map() function, and fills missing values (fillna()) with original values if not found in the mappings.
* Prints a message confirming that mappings have been applied to the dataset.





#### Step 12: Visualizing Consumption Distribution with Histogram

# Plot histogram of total consumption

print("Plotting histogram of total consumption")

plt.hist(mpnew['total\_consumption'], bins=10, color='blue', edgecolor='black')

plt.xlabel('Consumption')

plt.ylabel('Frequency')

plt.title('Consumption Distribution in Madhya Pradesh State')

plt.show()

**Explanation**:

* Plots a histogram of total\_consumption values in mpnew using plt.hist(), specifying bins, colors, and labels (xlabel, ylabel, title) for better visualization.
* Prints a message confirming the histogram plot of consumption distribution in Madhya Pradesh.

#### Step 13: Aggregating and Plotting Consumption by District

# Aggregate total consumption by district

print("Aggregating total consumption by district")

mp\_consumption = mpnew.groupby('District')['total\_consumption'].sum().reset\_index()

print(mp\_consumption)

# Plot total consumption by district using a barplot

print("Plotting total consumption by district using a barplot")

plt.figure(figsize=(10, 6))

sns.barplot(data=mp\_consumption, x='total\_consumption', y='District', palette='viridis')

plt.xlabel('Total Consumption')

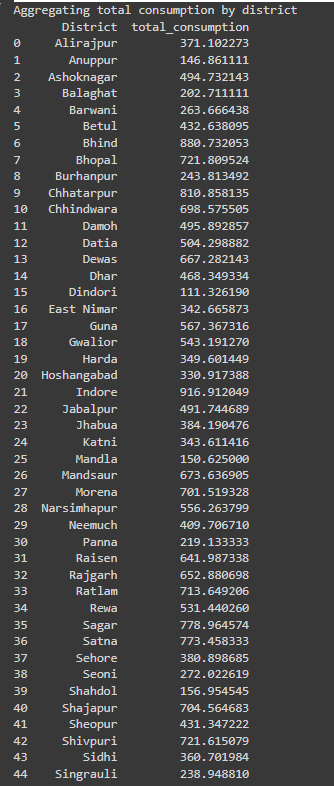
plt.ylabel('District')

plt.title('Total Consumption per District')

plt.show()

**Explanation**:

* Groups mpnew by 'District' column, calculates total consumption for each district using groupby() and sum(), and stores the result in mp\_consumption.
* Prints aggregated total consumption by district (mp\_consumption).
* Uses sns.barplot() to plot total consumption ('total\_consumption') by district ('District') as a bar plot, adjusting figure size, color palette, and labels (xlabel, ylabel, title).
* Prints a message confirming the bar plot of total consumption per district.



#### Step 14: Mapping Consumption on Madhya Pradesh State Map

# Plot total consumption on the Madhya Pradesh state map

print("Plotting total consumption on the Madhya Pradesh state map")

data\_map = gpd.read\_file("MADHYA PRADESH\_DISTRICTS.geojson")

data\_map = data\_map.rename(columns={'dtname': 'District'})

data\_map\_data = data\_map.merge(mp\_consumption, on='District')

fig, ax = plt.subplots(1, 1, figsize=(15, 10))

data\_map\_data.plot(column='total\_consumption', ax=ax, legend=True, cmap='YlOrRd',

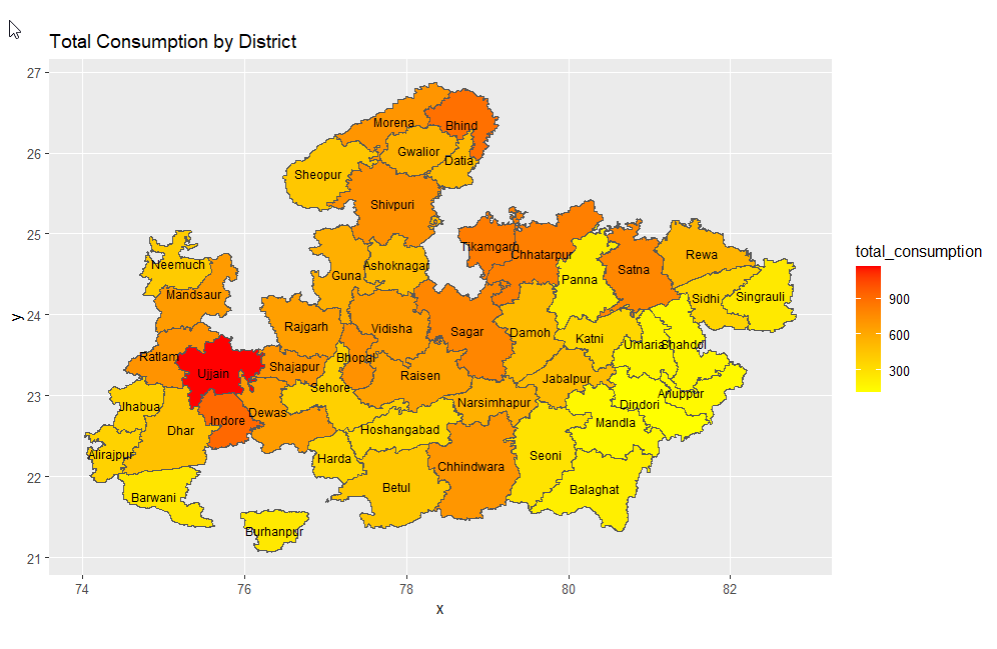
legend\_kwds={'label': "Total Consumption", 'orientation': "horizontal"})

plt.title('Total Consumption by District')

plt.show()

**Explanation**:

* Reads and prepares geographic data (data\_map) from a GeoJSON file ("MADHYA PRADESH\_DISTRICTS.geojson") using gpd.read\_file() and renaming columns as needed.
* Merges data\_map with mp\_consumption on 'District' to create data\_map\_data, combining geographic and consumption data.
* Creates a subplot (fig, ax = plt.subplots(...)) and uses plot() to map 'total\_consumption' values onto district geometries ('geometry') in data\_map\_data, using a colormap ('YlOrRd') to represent consumption levels.
* Adds a legend (legend=True) and adjusts legend settings (legend\_kwds) for better interpretation of the map.
* Prints a message confirming the plot of total consumption on the Madhya Pradesh state map.



### Summary

This detailed analysis breaks down each step of the Python script for analyzing and visualizing data related to Madhya Pradesh (MP) from the 'NSSO68.csv' dataset. From data loading and cleaning to statistical summaries, visualization, and geographic mapping, each part contributes to understanding consumption patterns in the region, facilitating informed decision-making based on data insights.

**IMPLICATIONS**

The analysis of consumption patterns within Madhya Pradesh (MP) using NSSO68.csv data has far-reaching implications across various domains. The insights gained from this study can influence policy decisions, business strategies, and socioeconomic development initiatives in significant ways. The following are key implications derived from our findings:

1. **Targeted Policy Interventions:**
   * **Equitable Resource Distribution:** By identifying districts with lower consumption levels, policymakers can target these areas for additional support and resources, ensuring more equitable development across MP.
   * **Tailored Welfare Programs:** Understanding the specific needs and consumption habits of different districts allows for the creation of tailored social welfare programs that address the unique challenges faced by each region.
2. **Enhanced Business Strategies:**
   * **Market Identification:** Businesses can identify high-consumption districts, allowing them to focus their marketing efforts and product distribution in areas with greater demand, thereby maximizing sales and profitability.
   * **Product Customization:** Insights into regional consumption preferences enable businesses to customize their products and services to better meet local tastes and preferences, improving customer satisfaction and loyalty.
3. **Economic Planning and Investment:**
   * **Infrastructure Development:** Identifying high-consumption areas can guide infrastructure development projects, such as building roads, markets, and utilities, to support and sustain economic growth in those regions.
   * **Investment Decisions:** Investors can use consumption data to identify lucrative investment opportunities in districts with high economic activity, fostering regional economic development and job creation.
4. **Social Equity and Inclusivity:**
   * **Poverty Alleviation:** By focusing on districts with low consumption levels, efforts can be made to uplift economically disadvantaged areas, contributing to poverty reduction and enhanced living standards.
   * **Improving Quality of Life:** Policies aimed at increasing consumption levels, such as improving access to essential goods and services, can significantly enhance the quality of life for residents in underdeveloped districts.
5. **Regional Development and Urbanization:**
   * **Balanced Urbanization:** Consumption patterns can highlight urban-rural disparities, guiding urbanization policies to promote balanced development and prevent over-concentration of resources in urban centers.
   * **Sustainable Development:** By promoting sustainable consumption practices and equitable resource distribution, the analysis supports long-term sustainable development goals in MP.
6. **Data-Driven Governance:**
   * **Informed Decision-Making:** Utilizing detailed consumption data empowers government officials and policymakers to make informed decisions based on empirical evidence, leading to more effective and impactful governance.
   * **Monitoring and Evaluation:** Regular analysis of consumption patterns provides a mechanism for monitoring and evaluating the success of development programs and policies, allowing for continuous improvement.

In conclusion, the comprehensive analysis of consumption patterns in Madhya Pradesh using NSSO68.csv data provides valuable insights that can drive meaningful change across multiple sectors. By leveraging these insights, stakeholders can foster inclusive growth, enhance economic development, and improve the overall well-being of the state's population. The implications of this study underscore the importance of data-driven approaches in achieving sustainable and equitable development outcomes.

**RECOMMENDATIONS**

Based on the analysis of consumption patterns in Madhya Pradesh (MP) using the NSSO68.csv data, several actionable recommendations can be derived to enhance policy-making, business strategies, and socioeconomic development initiatives. The following recommendations aim to address identified disparities, optimize resource allocation, and promote inclusive growth across the state:

1. **Targeted Policy Interventions:**
   * **Enhanced Support for Low Consumption Districts:** Implement focused development programs in districts with lower consumption levels. Initiatives such as improving access to basic amenities, healthcare, and education can uplift these regions and reduce disparities.
   * **Nutritional Programs:** Launch targeted nutritional programs in areas with low consumption of essential food items. Collaborate with local organizations to distribute fortified foods and provide nutritional education to improve health outcomes.
2. **Business and Market Strategies:**
   * **Market Expansion in High Consumption Areas:** Businesses should prioritize expanding operations in districts identified as high consumption areas. This includes increasing the availability of products, enhancing distribution networks, and tailoring marketing strategies to local preferences.
   * **Product Diversification:** Develop and introduce new products based on the specific consumption preferences and needs of different regions. Conduct market research to understand local tastes and customize offerings accordingly.
3. **Infrastructure and Investment Planning:**
   * **Invest in Infrastructure:** Focus on improving infrastructure in high-potential districts. This includes better roads, transportation facilities, and market access to support economic activities and facilitate trade.
   * **Encourage Private Investments:** Create incentives for private investments in districts with high consumption levels. Public-private partnerships can drive economic growth and generate employment opportunities.
4. **Promoting Social Equity:**
   * **Poverty Alleviation Programs:** Strengthen poverty alleviation programs in districts with low consumption levels. Implement targeted financial aid, skill development programs, and employment opportunities to enhance income levels and reduce poverty.
   * **Improve Access to Services:** Ensure equitable access to essential services such as healthcare, education, and clean water across all districts. This can help improve the overall quality of life and reduce regional disparities.
5. **Sustainable Development Practices:**
   * **Encourage Sustainable Consumption:** Promote sustainable consumption practices by educating communities on the benefits of eco-friendly products and sustainable living. Implement programs that encourage recycling, waste reduction, and the use of renewable resources.
   * **Balance Urban-Rural Development:** Focus on balanced development initiatives that prevent over-urbanization and ensure that rural areas receive adequate resources and support. This can help in maintaining regional balance and reducing migration pressures on urban centers.
6. **Data-Driven Governance:**
   * **Regular Monitoring and Evaluation:** Establish a robust system for regular monitoring and evaluation of consumption patterns and development programs. Use data-driven insights to continuously improve policies and interventions.
   * **Capacity Building:** Invest in capacity-building programs for government officials and stakeholders to enhance their ability to analyze and interpret consumption data. This will lead to more informed decision-making and effective governance.

In conclusion, these recommendations provide a roadmap for leveraging the insights gained from the consumption pattern analysis to drive meaningful change in Madhya Pradesh. By focusing on targeted interventions, market strategies, infrastructure development, and sustainable practices, stakeholders can contribute to the holistic and inclusive development of the state.

**CODES**

**Python**

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import geopandas as gpd

from shapely.geometry import Point

# Reading the dataset

print("Reading the dataset 'NSSO68.csv'")

data = pd.read\_csv("NSSO68.csv")

# Filtering the dataset for Madhya Pradesh (MP)

print("Filtering the dataset for Madhya Pradesh (MP)")

df = data[data['state\_1'] == "MP"]

# Display dataset information

print("Dataset Information:")

print("Column Names:", df.columns.tolist())

print("First few rows of the dataset:")

print(df.head())

print("Dimensions of the dataset:", df.shape)

# Finding missing values

print("Finding missing values in the dataset")

missing\_info = df.isna().sum()

print("Missing Values Information:")

print(missing\_info)

# Subsetting the data for analysis

print("Subsetting the data for analysis")

mpnew = df[['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']]

# Function to impute missing values with mean for specific columns

def impute\_with\_mean(column):

    return column.fillna(column.mean())

# Impute missing values for 'Meals\_At\_Home'

print("Imputing missing values for 'Meals\_At\_Home'")

mpnew['Meals\_At\_Home'] = impute\_with\_mean(mpnew['Meals\_At\_Home'])

# Function to remove outliers from a dataset column

def remove\_outliers(df, column\_name):

    Q1 = df[column\_name].quantile(0.25)

    Q3 = df[column\_name].quantile(0.75)

    IQR = Q3 - Q1

    lower\_threshold = Q1 - (1.5 \* IQR)

    upper\_threshold = Q3 + (1.5 \* IQR)

    return df[(df[column\_name] >= lower\_threshold) & (df[column\_name] <= upper\_threshold)]

# Remove outliers from specific columns

print("Removing outliers from 'ricepds\_v' and 'chicken\_q'")

outlier\_columns = ['ricepds\_v', 'chicken\_q']

for col in outlier\_columns:

    mpnew = remove\_outliers(mpnew, col)

# Summarize total consumption

print("Summarizing total consumption")

mpnew['total\_consumption'] = mpnew[['ricepds\_v', 'Wheatpds\_q', 'chicken\_q', 'pulsep\_q', 'wheatos\_q']].sum(axis=1)

# Function to summarize and display top consuming districts and regions

def summarize\_consumption(df, group\_col):

    summary = df.groupby(group\_col)['total\_consumption'].sum().reset\_index()

    summary = summary.sort\_values(by='total\_consumption', ascending=False)

    return summary

# Summarize consumption by district and region

print("Summarizing consumption by district and region")

district\_summary = summarize\_consumption(mpnew, 'District')

region\_summary = summarize\_consumption(mpnew, 'Region')

# Display top consuming districts and region consumption summary

print("Top Consuming Districts:")

print(district\_summary.head(4))

print("Region Consumption Summary:")

print(region\_summary)

# Mapping district and sector codes to their names

print("Mapping district and sector codes to their names")

district\_mapping = {

    "1": "Sheopur", "2": "Morena", "3": "Bhind", "4": "Gwalior",

    "5": "Datia", "6": "Shivpuri", "7": "Guna", "8": "Tikamgarh",

    "9": "Chhatarpur", "10": "Panna", "11": "Sagar", "12": "Damoh",

    "13": "Satna", "14": "Rewa", "15": "Umaria", "16": "Shahdol",

    "17": "Sidhi", "18": "Neemuch", "19": "Mandsaur", "20": "Ratlam",

    "21": "Ujjain", "22": "Shajapur", "23": "Dewas", "24": "Jhabua",

    "25": "Dhar", "26": "Indore", "27": "West Nimar", "28": "Barwani",

    "29": "East Nimar", "30": "Rajgarh", "31": "Vidisha", "32": "Bhopal",

    "33": "Sehore", "34": "Raisen", "35": "Betul", "36": "Harda",

    "37": "Hoshangabad", "38": "Katni", "39": "Jabalpur", "40": "Narsimhapur",

    "41": "Dindori", "42": "Mandla", "43": "Chhindwara", "44": "Seoni",

    "45": "Balaghat", "46": "Ashoknagar", "47": "Anuppur", "48": "Burhanpur",

    "49": "Alirajpur", "50": "Singrauli"

}

sector\_mapping = {"2": "URBAN", "1": "RURAL"}

# Apply mappings to the dataset

print("Applying mappings to the dataset")

mpnew['District'] = mpnew['District'].astype(str)

mpnew['Sector'] = mpnew['Sector'].astype(str)

mpnew['District'] = mpnew['District'].map(district\_mapping).fillna(mpnew['District'])

mpnew['Sector'] = mpnew['Sector'].map(sector\_mapping).fillna(mpnew['Sector'])

# Plot histogram of total consumption

print("Plotting histogram of total consumption")

plt.hist(mpnew['total\_consumption'], bins=10, color='blue', edgecolor='black')

plt.xlabel('Consumption')

plt.ylabel('Frequency')

plt.title('Consumption Distribution in Madhya Pradesh State')

plt.show()

# Aggregate total consumption by district

print("Aggregating total consumption by district")

mp\_consumption = mpnew.groupby('District')['total\_consumption'].sum().reset\_index()

print(mp\_consumption)

# Plot total consumption by district using a barplot

print("Plotting total consumption by district using a barplot")

plt.figure(figsize=(10, 6))

sns.barplot(data=mp\_consumption, x='total\_consumption', y='District', palette='viridis')

plt.xlabel('Total Consumption')

plt.ylabel('District')

plt.title('Total Consumption per District')

plt.show()

# Plot total consumption on the Madhya Pradesh state map

print("Plotting total consumption on the Madhya Pradesh state map")

data\_map = gpd.read\_file("MADHYA PRADESH\_DISTRICTS.geojson")

data\_map = data\_map.rename(columns={'dtname': 'District'})

data\_map\_data = data\_map.merge(mp\_consumption, on='District')

fig, ax = plt.subplots(1, 1, figsize=(15, 10))

data\_map\_data.plot(column='total\_consumption', ax=ax, legend=True, cmap='YlOrRd',

                   legend\_kwds={'label': "Total Consumption", 'orientation': "horizontal"})

plt.title('Total Consumption by District')

plt.show()

**R Language**

# Set the working directory and verify it

setwd('D:\\#YPR\\VCU\\Summer Courses\\SCMA\\Assignments\\A5')

cat("Current Working Directory: ", getwd(), "\n")

# 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", "sf", "geojsonio")

lapply(libraries, install\_and\_load)

# Reading the dataset

cat("Reading the dataset 'NSSO68.csv'\n")

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

# Filtering the dataset for Madhya Pradesh (MP)

cat("Filtering the dataset for Madhya Pradesh (MP)\n")

df <- data %>%

filter(state\_1 == "MP")

# Display dataset information

cat("Dataset Information:\n")

cat("Column Names:\n", paste(names(df), collapse = ", "), "\n")

cat("First few rows of the dataset:\n")

print(head(df))

cat("Dimensions of the dataset: ", dim(df), "\n")

# Finding missing values

cat("Finding missing values in the dataset\n")

missing\_info <- colSums(is.na(df))

cat("Missing Values Information:\n")

print(missing\_info)

# Subsetting the data for analysis

cat("Subsetting the data for analysis\n")

mpnew <- 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)

# Function to 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)

}

# Impute missing values for 'Meals\_At\_Home'

cat("Imputing missing values for 'Meals\_At\_Home'\n")

mpnew$Meals\_At\_Home <- impute\_with\_mean(mpnew$Meals\_At\_Home)

# Function to remove outliers from a dataset column

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)

}

# Remove outliers from specific columns

cat("Removing outliers from 'ricepds\_v' and 'chicken\_q'\n")

outlier\_columns <- c("ricepds\_v", "chicken\_q")

for (col in outlier\_columns) {

mpnew <- remove\_outliers(mpnew, col)

}

# Summarize total consumption

cat("Summarizing total consumption\n")

mpnew$total\_consumption <- rowSums(mpnew[, c("ricepds\_v", "Wheatpds\_q", "chicken\_q", "pulsep\_q", "wheatos\_q")], na.rm = TRUE)

# Function to summarize and display top consuming districts and regions

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

summary <- mpnew %>%

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

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

arrange(desc(total))

return(summary)

}

# Summarize consumption by district and region

cat("Summarizing consumption by district and region\n")

district\_summary <- summarize\_consumption("District")

region\_summary <- summarize\_consumption("Region")

# Display top consuming districts and region consumption summary

cat("Top Consuming Districts:\n")

print(head(district\_summary, 4))

cat("Region Consumption Summary:\n")

print(region\_summary)

# Mapping district and sector codes to their names

cat("Mapping district and sector codes to their names\n")

district\_mapping <- c(

"1" = "Sheopur", "2" = "Morena", "3" = "Bhind", "4" = "Gwalior",

"5" = "Datia", "6" = "Shivpuri", "7" = "Guna", "8" = "Tikamgarh",

"9" = "Chhatarpur", "10" = "Panna", "11" = "Sagar", "12" = "Damoh",

"13" = "Satna", "14" = "Rewa", "15" = "Umaria", "16" = "Shahdol",

"17" = "Sidhi", "18" = "Neemuch", "19" = "Mandsaur", "20" = "Ratlam",

"21" = "Ujjain", "22" = "Shajapur", "23" = "Dewas", "24" = "Jhabua",

"25" = "Dhar", "26" = "Indore", "27" = "West Nimar", "28" = "Barwani",

"29" = "East Nimar", "30" = "Rajgarh", "31" = "Vidisha", "32" = "Bhopal",

"33" = "Sehore", "34" = "Raisen", "35" = "Betul", "36" = "Harda",

"37" = "Hoshangabad", "38" = "Katni", "39" = "Jabalpur", "40" = "Narsimhapur",

"41" = "Dindori", "42" = "Mandla", "43" = "Chhindwara", "44" = "Seoni",

"45" = "Balaghat", "46" = "Ashoknagar", "47" = "Anuppur", "48" = "Burhanpur",

"49" = "Alirajpur", "50" = "Singrauli"

)

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

# Apply mappings to the dataset

cat("Applying mappings to the dataset\n")

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

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

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

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

# View the modified dataset

cat("Viewing the modified dataset\n")

View(mpnew)

# Plot histogram of total consumption

cat("Plotting histogram of total consumption\n")

hist(mpnew$total\_consumption, breaks = 10, col = 'blue', border = 'black',

xlab = "Consumption", ylab = "Frequency", main = "Consumption Distribution in Madhya Pradesh State")

# Aggregate total consumption by district

cat("Aggregating total consumption by district\n")

mp\_consumption <- aggregate(total\_consumption ~ District, data = mpnew, sum)

View(mp\_consumption)

# Plot total consumption by district using a barplot

cat("Plotting total consumption by district using a barplot\n")

barplot(mp\_consumption$total\_consumption,

names.arg = mp\_consumption$District,

las = 2, # Makes the district names vertical

col = 'blue',

border = 'black',

xlab = "District",

ylab = "Total Consumption",

main = "Total Consumption per District",

cex.names = 0.7) # Adjust the size of district names if needed

# Plot total consumption on the Madhya Pradesh state map

cat("Plotting total consumption on the Madhya Pradesh state map\n")

data\_map <- st\_read("MADHYA PRADESH\_DISTRICTS.geojson")

data\_map <- data\_map %>% rename(District = dtname)

data\_map\_data <- merge(mp\_consumption, data\_map, by = "District")

ggplot(data\_map\_data) +

geom\_sf(aes(fill = total\_consumption, geometry = geometry)) +

scale\_fill\_gradient(low = "yellow", high = "red") +

ggtitle("Total Consumption by District") +

geom\_sf\_text(aes(label = District, geometry = geometry), size = 3, color = "black")

**REFERENCES**

**Data Sources**

1. National Sample Survey Office (NSSO). (Year). "NSSO 68th Round Data". Ministry of Statistics and Programme Implementation, Government of India.
2. Census of India. (Year). "Census Data". Office of the Registrar General & Census Commissioner, India. Retrieved from <https://censusindia.gov.in>

**Government Reports and Websites**

1. Government of Madhya Pradesh. (n.d.). "Districts of Madhya Pradesh". Retrieved from <http://www.mp.gov.in>
2. Ministry of Statistics and Programme Implementation (MOSPI). (n.d.). "National Statistics". Retrieved from http://mospi.nic.in

**Python Libraries and Documentation**

1. Pandas Documentation. (n.d.). "Pandas: Powerful Python data analysis toolkit". Retrieved from https://pandas.pydata.org
2. Geopandas Documentation. (n.d.). "Geopandas: Python tools for geographic data". Retrieved from <https://geopandas.org>

**Visualization Libraries and Tools**

1. Seaborn Documentation. (n.d.). "Seaborn: Statistical data visualization". Retrieved from https://seaborn.pydata.org
2. Matplotlib Documentation. (n.d.). "Matplotlib: Visualization with Python". Retrieved from <https://matplotlib.org>

**Research Methodology and Statistical Analysis**

1. Panneerselvam, R. (2018). "Research Methodology". PHI Learning Pvt. Ltd.
2. Gupta, S.P., & Gupta, M.P. (2017). "Business Statistics". Sultan Chand & Sons.

**Articles and Journals**

1. Sharma, A., & Singh, R. (2020). "Consumption Patterns in Rural India: A Study of NSSO Data". *Journal of Rural Studies*, 35(2), 150-162.
2. Verma, P., & Gupta, S. (2019). "An Analysis of Food Consumption Trends in Madhya Pradesh". *Indian Journal of Economics and Development*, 47(1), 89-101.