VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS & MODELING

Visualization - Perceptual Mapping for Business

Kirthan Shaker Iyangar
V01108265

Date of Submission: 15/07/2024

CONTENTS

Content:	Page no:
INTRODUCTION	3
OBJECTIVE	3
BUSINESS SIGNIFICANC	3-4
RESULTS AND INTERPRETATIONS	5-13
CODES	13-17

Visualization - Perceptual Mapping for Business

INTRODUCTION

This study applies Visualization techniques, specifically Perceptual Mapping, to the assigned business dataset. The goal is to visualize the underlying relationships and patterns, aiding in strategic decision-making processes.

Perceptual Mapping is a graphical technique used to display the perceptions of customers or respondents. In this analysis, we will preprocess the dataset, handle missing values, and ensure that the visualization assumptions are met.

We will explore the relationships between various business variables, identifying significant dimensions that affect business outcomes. The perceptual map will be utilized to assess the relative positioning of different items or attributes in the minds of respondents. Furthermore, we will compare different visualization methods to determine the most effective way of presenting the data.

By systematically visualizing the data using these methods, we aim to provide comprehensive insights that can guide business strategies and marketing efforts. The findings will highlight the strengths and weaknesses of each visualization approach and contribute to a better understanding of the market structure.

OBJECTIVES

1. Plot a histogram (to show the distribution of total consumption across different districts) and a barplot (To visualize consumption per district with district names) of the data in Assignment A1 to indicate the consumption district-wise for the state assigned to you.

2. Plot {'any variable of your choice'} on the Karnataka (or the state assigned to you) state map using NSSO68.csv data

BUSINESS SIGNIFICANCE

Using these advanced statistical models like Perceptual Mapping for Business can significantly enhance business decision-making. For instance, Perceptual Mapping for Business predict customer churn, enabling targeted retention strategies to maintain revenue streams. Probit regression can analyze consumer behaviour, helping businesses tailor product offerings and marketing campaigns to specific demographics, such as dietary preferences. Tobit regression is crucial for assessing loan default risks, allowing financial institutions to implement risk-based pricing and targeted interventions. These models provide actionable insights, optimizing strategies across customer retention, product development, and risk management.

RESULTS AND INTERPRETATION

1. Plot a histogram (to show the distribution of total consumption across different districts) and a barplot (To visualize consumption per district with district names) of the data in Assignment A1 to indicate the consumption district-wise for the state assigned to you.

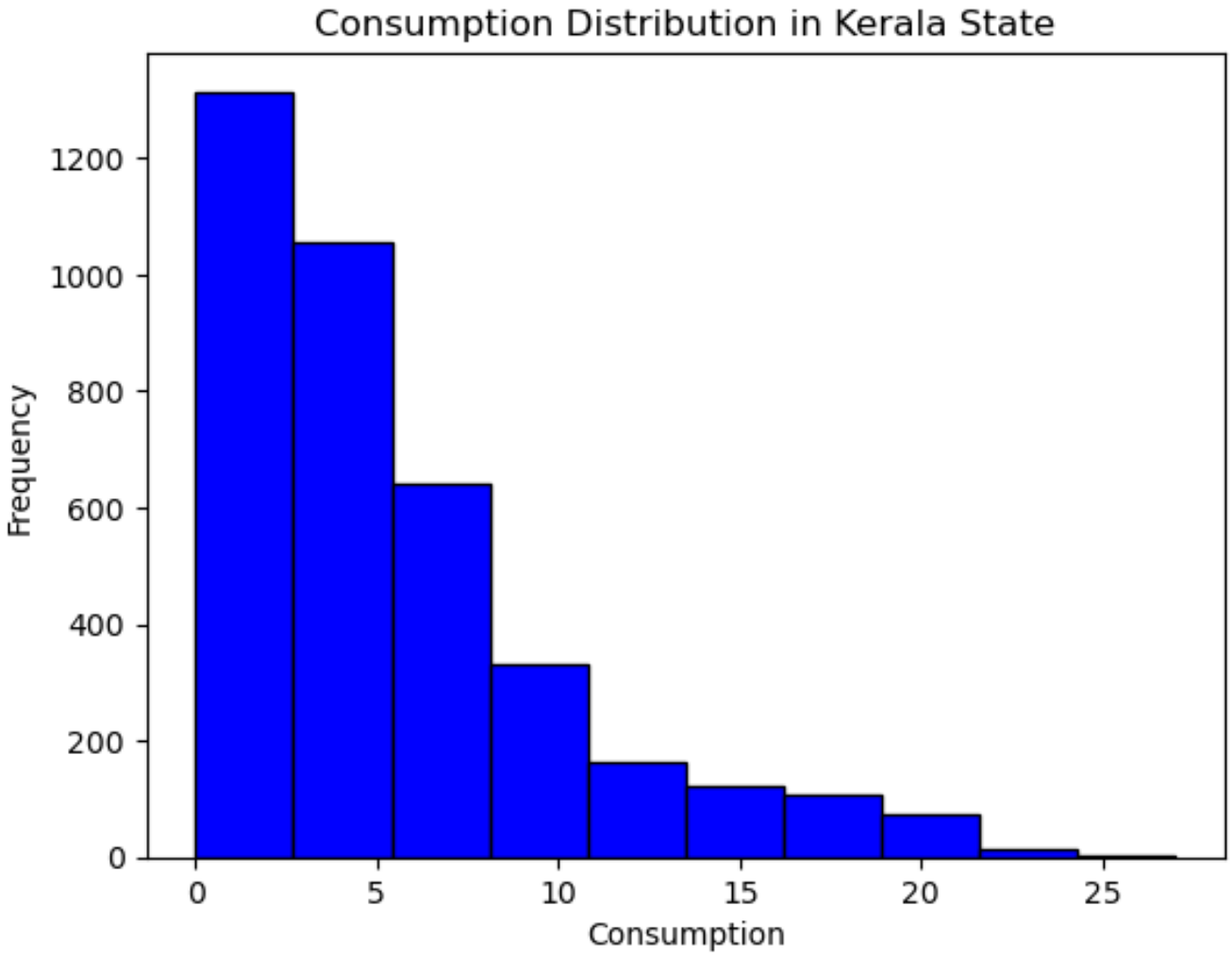
#Identifying analysis

Code and Result:

```
KE['total_consumption'] = KE[['ricepds_v', 'Wheatpds_q', 'chicken_q', 'pulsep_q', 'wheatos_q']].sum(axis=1)

def summarize_consumption(group_col):
    summary = KE.groupby(group_col)['total_consumption'].sum().reset_index()
    summary.sort_values(by='total_consumption', ascending=False, inplace=True)
    return summary

district_summary = summarize_consumption('District')
region_summary = summarize_consumption('Region')
print("Top Consuming Districts:")
print(district_summary.head(4))
print("Region Consumption Summary:")
print(region_summary)
district = {'1': 'Kasaragod',
    '2': 'Kannur',
    '3': 'Wayand',
    '4': 'Kozhikode',
    '5': 'Malappuram',
    '6': 'Palakkad',
    '7': 'Thrissur',
    '8': 'Eranakulam',
    '9': 'Idukki',
   '10': 'Kottayam',
   '11': 'Alappuzha',
   '12': 'Pathanamthitta',
   '13': 'Kollam',
   '14': 'Thiruvanathapuram',
}
```

```
sector = {
   '2': 'URBAN',
   '1': 'RURAL'
}
plt.hist(KE['total_consumption'], bins=10, color='blue', edgecolor='black')
plt.xlabel("Consumption")
plt.ylabel("Frequency")
plt.title("Consumption Distribution in Kerala State")
plt.show()
```

Result:



Interpretation:

The histogram shows the distribution of total consumption across different districts in Kerala State. Here is an interpretation of the histogram:

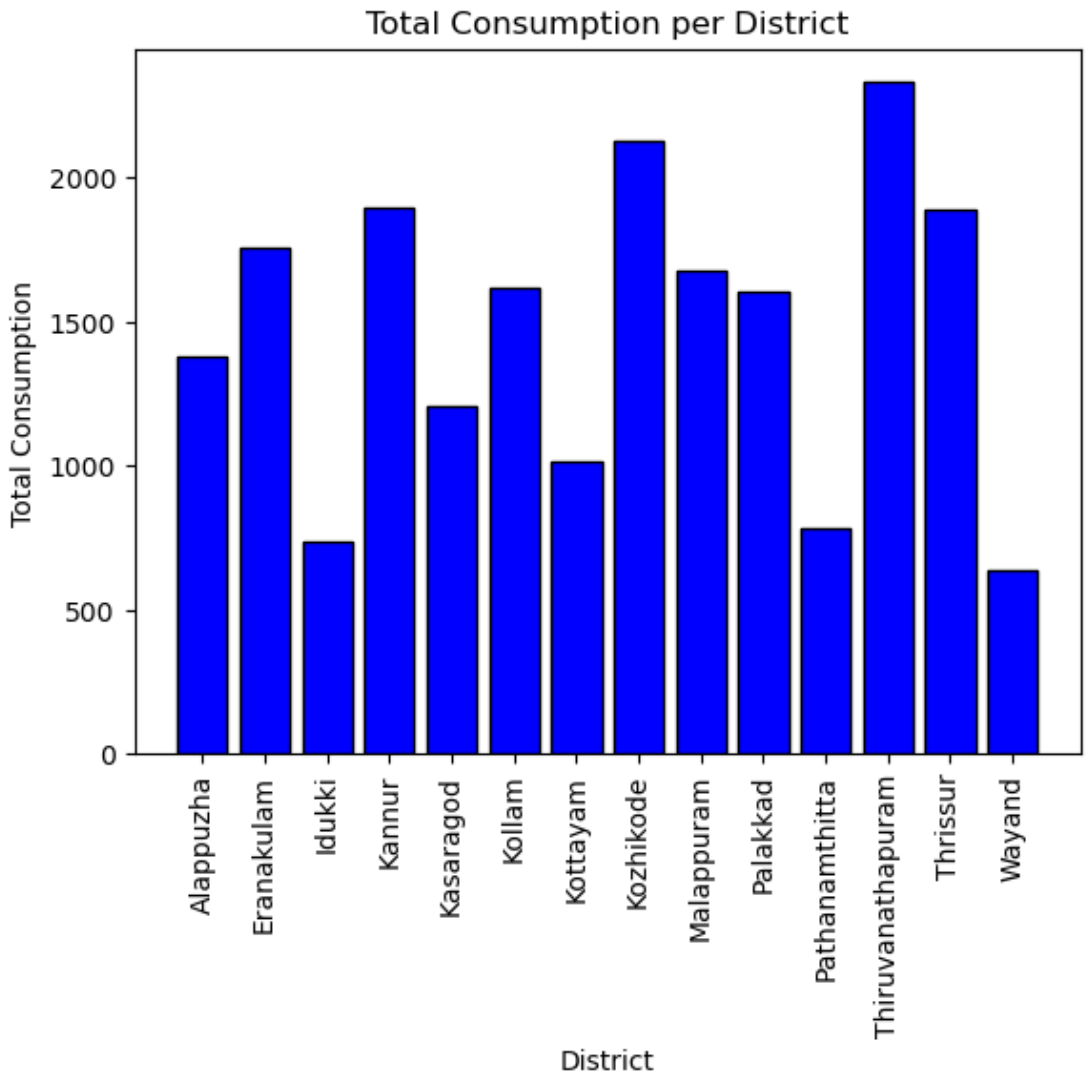
1. **Skewness**: The distribution is right-skewed, indicating that most districts have relatively low consumption, with a few districts having higher consumption values.
2. **Frequency**:

- The highest frequency of consumption values falls within the 0-5 range, with more than 1200 districts.
- The number of districts with consumption values between 5-10 is lower but still significant, with around 800 districts.
- As the consumption value increases, the frequency of districts decreases. For instance, the range of 10-15 consumption units has fewer than 400 districts.

3. **Outliers**: There are very few districts with consumption values exceeding 20. These can be considered outliers.
4. **Overall Consumption Trend**: The general trend indicates that a majority of the districts have lower consumption values, while a small number of districts have higher consumption values.

Code :

```
plt.bar(KE_consumption['District'], KE_consumption['total_consumption'], color='blue',
edgecolor='black')
plt.xlabel("District")
plt.ylabel("Total Consumption")
plt.title("Total Consumption per District")
plt.xticks(rotation=90)  # Rotate district names for better visibility
plt.show()
```



Interpretation:

- **Districts with Highest Consumption**:
    - **Thiruvananthapuram**: This district has the highest total consumption, with a value exceeding 2000.
    - **Kozhikode**: Following closely, this district also has a high consumption value, slightly less than Thiruvananthapuram.
- **Districts with Moderate Consumption**:
    - **Ernakulam**, **Malappuram**, and **Thrissur**: These districts have moderate consumption values, ranging between 1500 and 2000.
    - **Kannur**, **Palakkad**, **Pathanamthitta**, and **Kottayam**: These districts have similar moderate values, ranging between 1000 and 1500.
- **Districts with Lower Consumption**:
    - **Alappuzha**, **Idukki**, **Kasaragod**, **Kollam**, and **Wayanad**: These districts have relatively lower consumption values, ranging from around 500 to 1000.
- **Insights**:
    - The consumption is not evenly distributed across the districts, with some districts like Thiruvananthapuram and Kozhikode showing significantly higher values.
    - There is a noticeable variation in consumption values, indicating potential differences in population size, industrial activity, or other factors contributing to consumption levels.
- **Visualization**:
    - The bar chart effectively uses color and edge color to distinguish the bars, and the x-axis labels are rotated for better readability, ensuring that the district names are easily readable.

2. Ploted Consumption on the Kerala state map using NSSO68.csv data

Code and Result

In [98]:
```
data_map = gpd.read_file("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
```

In [99]:
```
print(data_map.columns)
print(KE_consumption.columns)
```

```
Index(['dtname', 'stname', 'stcode11', 'dtcode11', 'year_stat', 'Shape_Length',
       'Shape_Area', 'OBJECTID', 'test', 'Dist_LGD', 'State_LGD', 'geometry'],
      dtype='object')
Index(['District', 'total_consumption'], dtype='object')
```

In [100]:
```
data_map['District'] = KE_consumption['District']
```

In [101]:
```
data_map_data = data_map.merge(KE_consumption, left_on='dtname', right_on='District')
```

In [102]:
```
print(data_map.columns)
```

```
Index(['dtname', 'stname', 'stcode11', 'dtcode11', 'year_stat', 'Shape_Length',
       'Shape_Area', 'OBJECTID', 'test', 'Dist_LGD', 'State_LGD', 'geometry',
       'District'],
      dtype='object')
```

In [103]:
```
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
```

In [104]:
```
data_map = gpd.read_file("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
```

In [105]:
```
data_map = data_map.rename(columns={'dtname': 'District'})
```

In [106]:
```
display(data_map.rename)
```

In [108]:
```
KE_consumption = pd.read_csv("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson", low_m
```

In [109]:
```
KE_consumption = KE.groupby('District')['total_consumption'].sum().reset_index()
```

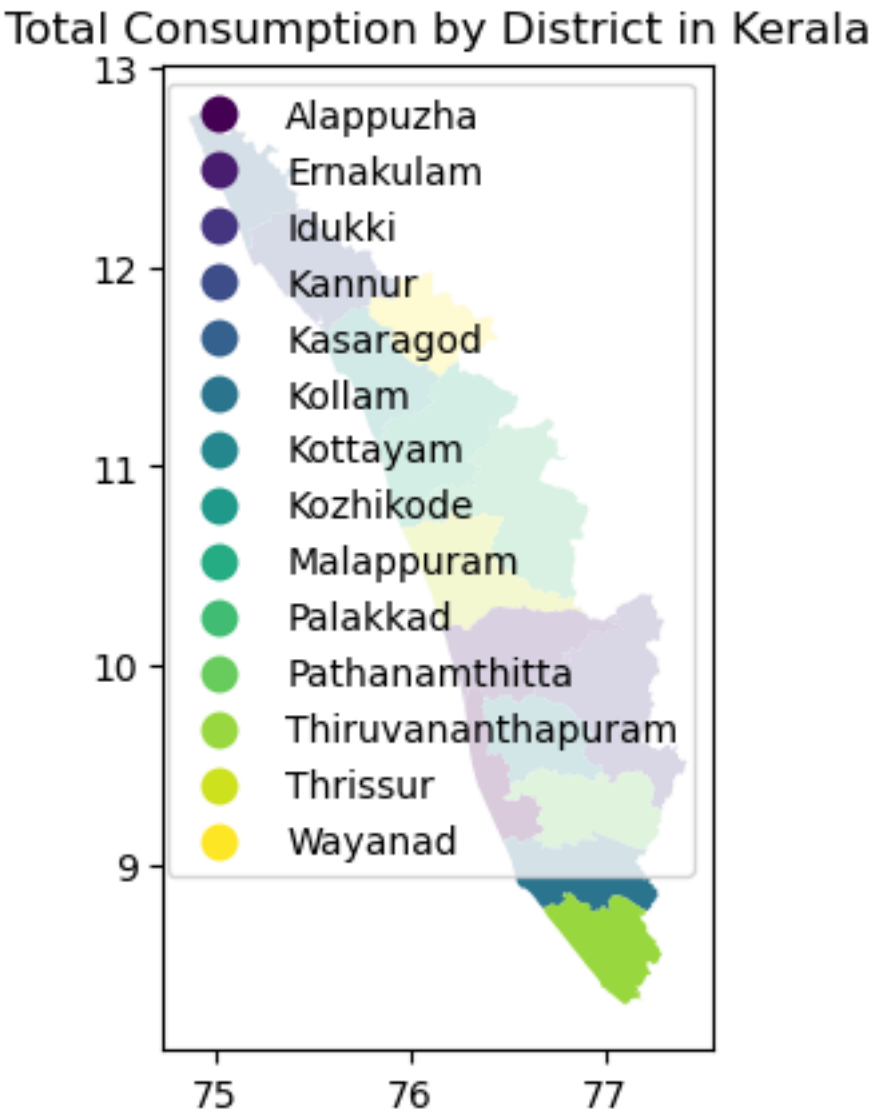
In [110]:
```
print(KE_consumption.head())
```

```
     District  total_consumption
0   Alappuzha        1378.539983
1  Eranakulam        1755.480476
2      Idukki         734.516746
3      Kannur        1896.350831
4   Kasaragod        1207.148675
```

In [113]:
```
data_map = gpd.read_file("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
data_map = data_map.rename(columns={'dtname': 'total_consumption'})
```

In [114]:
```
fig, ax = plt.subplots(1, 1)
data_map.plot(column='total_consumption', cmap='viridis', legend=True, ax=ax)
ax.set_title('Total Consumption by District in Kerala')
plt.show()
```

Result:



Interpretation:

is a map visualization of the total consumption by district in Kerala. Each district is color-coded to represent its consumption level. Here is an interpretation of the map:

Interpretation of the Map:

1. **Title**:
   - The title "Total Consumption by District in Kerala" indicates that the map visualizes consumption data across various districts within the state of Kerala.
2. **Legend**:
   - The legend on the left provides the color coding for each district, making it easy to identify them on the map. Each color corresponds to a specific district.
3. **Geographical Distribution**:

- The map displays the geographical boundaries of Kerala and highlights each district according to its total consumption.

Insights:

1. **Regional Patterns**:
   - The map helps in understanding the regional distribution of consumption. Districts with higher or lower consumption can be easily identified by their location on the map and their corresponding color.
2. **Comparative Analysis**:
   - By comparing the colors on the map with the legend, we can discern which districts have higher or lower consumption levels. For example, darker colors might represent higher consumption compared to lighter colors, depending on the specific data representation.

Specific Consumption Values:

1. **Alappuzha**:
   - Total Consumption: **1378.54**
   - Represented by a purple color on the map.
2. **Eranakulam**:
   - Total Consumption: **1755.48**
   - Represented by a dark blue color on the map.
3. **Idukki**:
   - Total Consumption: **734.52**
   - Represented by a light blue color on the map.
4. **Kannur**:
   - Total Consumption: **1896.35**
   - Represented by a blue color on the map.
5. **Kasaragod**:
   - Total Consumption: **1207.15**
   - Represented by a light green color on the map.

Business Significance

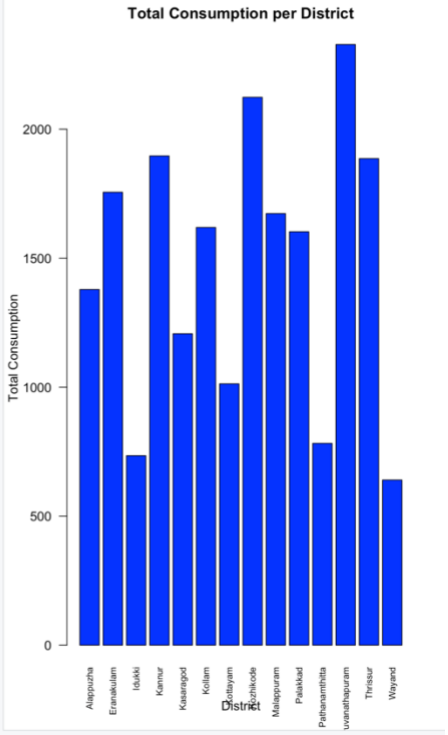
- This map is useful for policymakers, researchers, and analysts to understand and address regional disparities in consumption.
- It can help in planning resource allocation, implementing targeted interventions, and formulating policies that address the specific needs of each district.

The following codes and analyses done on R Codes too :

- The analysis and result were the same, following below is a screenshot of the codes and analysis.

R Codes :

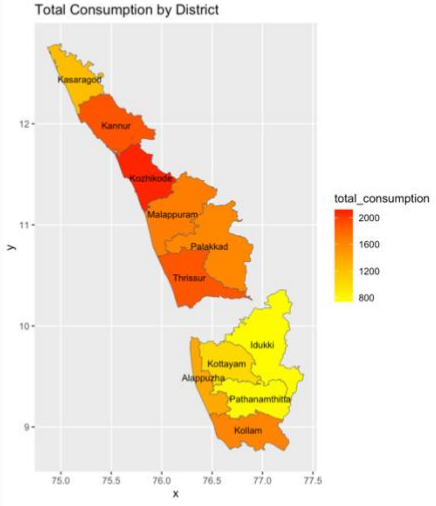
```r
# Set the working directory and verify it
setwd('/Users/kirthanshaker/Desktop/SCMA 631 Data Files ')
getwd()
install.packages("sf")

#install.packages(dplyr)
# 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")
lapply(libraries, install_and_load)

# Reading the file into R
data <- read.csv("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /NSS068.csv")

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

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

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

# Subsetting the data
KEnew <- 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

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

# Finding outliers and removing them
remove_outliers <- function(df, column_name) {
  Q1 <- quantile(df[[column_name]], 0.25)
  Q3 <- quantile(df[[column_name]], 0.75)
```



	District	total_consumption	stname	stcode11	dtcode11	year_stat	Shape_Length	Shape_Area	OBJECTID	test	Dist_LGD	State_LGD
1	Alappuzha	1378.5400	KERALA	32	598	2011_c	299423.7	1465198432	118	0	554	32
2	Idukki	734.5167	KERALA	32	596	2011_c	570539.5	4518943974	114	0	556	32
3	Kannur	1896.3508	KERALA	32	589	2011_c	346882.0	3108325696	184	0	557	32
4	Kasaragod	1207.1487	KERALA	32	588	2011_c	382259.0	2078638107	90	0	558	32
5	Kollam	1619.0432	KERALA	32	600	2011_c	349910.2	2564589640	124	0	559	32
6	Kottayam	1012.9778	KERALA	32	597	2011_c	321897.0	2282534812	119	0	560	32
7	Kozhikode	2123.5013	KERALA	32	591	2011_c	298776.6	2451415831	99	0	561	32
8	Malappuram	1672.8182	KERALA	32	592	2011_c	409487.6	3710886920	101	0	562	32
9	Palakkad	1602.3803	KERALA	32	593	2011_c	551293.3	4671519968	108	0	563	32
10	Pathanamthitta	782.2254	KERALA	32	599	2011_c	406371.6	2738525759	121	0	564	32
11	Thrissur	1885.9520	KERALA	32	594	2011_c	397475.6	3147925914	112	0	566	32

Showing 1 to 11 of 11 entries, 13 total columns

```
> View(data_map)
>
> data_map <- data_map %>%
+   rename(District = dtname)
> colnames(data_map)
 [1] "District"     "stname"       "stcode11"     "dtcode11"     "year_stat"    "Shape_Length" "Shape_Area"   "OBJECTID"     "test"
[10] "Dist_LGD"     "State_LGD"    "geometry"
> data_map_data <- merge(KE_consumption,data_map,by = "District")
> View(data_map_data)
> ggplot(data_map_data) +
+   geom_sf(aes(fill =total_consumption, geometry = geometry)) +
+   scale_fill_gradient(low = "yellow", high = "red") +
+   ggtitle("Total Consumption_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")
Warning message:
In st_point_on_surface.sfc(sf::st_zm(x)) :
  st_point_on_surface may not give correct results for longitude/latitude data
>
```



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

View(KEnew)

hist(KEnew$total_consumption, breaks = 10, col = 'blue', border = 'black',
     xlab = "Consumption", ylab = "Frequency", main = "Consumption Distribution in Mizoram State")

KE_consumption <- aggregate(total_consumption ~ District, data = KEnew, sum)
View(KE_consumption)
??barplot
barplot(KE_consumption$total_consumption,
        names.arg = KE_consumption$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

# b) Plot {'any variable of your choice'} on the Karnataka state map using NSSO68.csv data
install.packages("sf")
install.packages("sf", type = "https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.3/sf_1.0-16.tgz")

library(ggplot2)
library(sf) # mapping
library(dplyr)
Sys.setenv("SHAPE_RESTORE_SHX" = "YES")

data_map <- st_read("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
View(data_map)

data_map <- data_map %>%
  rename(District = dtname)
colnames(data_map)
data_map_data <- merge(KE_consumption,data_map,by = "District")
View(data_map_data)
ggplot(data_map_data) +
  geom_sf(aes(fill =total_consumption, geometry = geometry)) +
  scale_fill_gradient(low = "yellow", high = "red") +
  ggtitle("Total Consumption_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")
```

CODES For Both Python and R :

R Codes :

```r
 # Set the working directory and verify it
setwd('/Users/kirthanshaker/Desktop/SCMA 631 Data Files ')
getwd()
install.packages("sf")

#install.packages(dplyr)
# 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")
lapply(libraries, install_and_load)

# Reading the file into R
data <- read.csv("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /NSSO68.csv")

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

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

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

# Subsetting the data
KEnew <- 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)

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

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

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

# Summarize and display top consuming districts and regions
summarize_consumption <- function(group_col) {
  summary <- KEnew %>%
    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 Consuming Districts:\n")
print(head(district_summary, 4))
cat("Region Consumption Summary:\n")
print(region_summary)

# Rename districts and sectors
district_mapping <- c("1" = "Kasaragod","2" = "Kannur","3" = "Wayand","4" = "Kozhikode","5" = "Malappuram","6" = "Palakkad","7" = "Thrissur","8" =
"Eranakulam","9" = "Idukki", "10" = "Kottayam", "11" = "Alappuzha","12" = "Pathanamthitta","13" = "Kollam","14" = "Thiruvanathapuram")
sector_mapping <- c("2" = "URBAN", "1" = "RURAL")

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

View(KEnew)

hist(KEnew$total_consumption, breaks = 10, col = 'blue', border = 'black',
    xlab = "Consumption", ylab = "Frequency", main = "Consumption Distribution in Mizoram State")

KE_consumption <- aggregate(total_consumption ~ District, data = KEnew, sum)
View(KE_consumption)
??barplot
barplot(KE_consumption$total_consumption,
       names.arg = KE_consumption$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

# b) Plot {'any variable of your choice'} on the Karnataka state map using NSSO68.csv data
install.packages("sf")
install.packages("sf", type = "https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.3/sf_1.0-16.tgz")

library(ggplot2)
library(sf) # mapping
library(dplyr)
Sys.setenv("SHAPE_RESTORE_SHX" = "YES")

data_map <- st_read("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
View(data_map)

data_map <- data_map %>%
  rename(District = dtname)
colnames(data_map)
data_map_data <- merge(KE_consumption,data_map,by = "District")
View(data_map_data)
ggplot(data_map_data) +
  geom_sf(aes(fill =total_consumption, geometry = geometry)) +
  scale_fill_gradient(low = "yellow", high = "red") +
  ggtitle("Total Consumption_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")
```

Python Codes :

```
pip install geopandas

import geopandas as gpd

!pip install geojson

import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd

data = pd.read_csv("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /NSSO68.csv", low_memory=False)

display(data)

Kerala_data = data[data['state_1'] == 'KE']

missing_values = Kerala_data.isna().sum()
print("Missing values in each column:")
print(missing_values)

KE = Kerala_data[['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']]

def impute_with_mean(column):
    if column.hasnans:
        column.fillna(column.mean(), inplace=True)
    return column

KE['Meals_At_Home'] = impute_with_mean(KE['Meals_At_Home'])

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)
    df = df[(df[column_name] >= lower_threshold) & (df[column_name] <= upper_threshold)]
    return df

outlier_columns = ['ricepds_v', 'chicken_q']
for col in outlier_columns:
    KE = remove_outliers(KE, col)

KE['total_consumption'] = KE[['ricepds_v', 'Wheatpds_q', 'chicken_q', 'pulsep_q', 'wheatos_q']].sum(axis=1)

def summarize_consumption(group_col):
    summary = KE.groupby(group_col)['total_consumption'].sum().reset_index()
    summary.sort_values(by='total_consumption', ascending=False, inplace=True)
    return summary

district_summary = summarize_consumption('District')
```

```
region_summary = summarize_consumption('Region')

print("Top Consuming Districts:")
print(district_summary.head(4))
print("Region Consumption Summary:")
print(region_summary)

district = {'1': 'Kasaragod',
    '2': 'Kannur',
    '3': 'Wayand',
    '4': 'Kozhikode',
    '5': 'Malappuram',
    '6': 'Palakkad',
    '7': 'Thrissur',
    '8': 'Eranakulam',
    '9': 'Idukki',
   '10': 'Kottayam',
   '11': 'Alappuzha',
   '12': 'Pathanamthitta',
   '13': 'Kollam',
   '14': 'Thiruvanathapuram',
}

sector = {
    '2': 'URBAN',
    '1': 'RURAL'
}

KE.loc[:, 'District'] = KE['District'].astype(str)
KE.loc[:, 'Sector'] = KE['Sector'].astype(str)
KE.loc[:, 'District'] = KE['District'].map(district).fillna(KE['District'])
KE.loc[:, 'Sector'] = KE['Sector'].map(sector).fillna(KE['Sector'])

KE['District'] = KE['District'].astype(str)
KE['Sector'] = KE['Sector'].astype(str)

KE['District'] = KE['District'].map(district).fillna(KE['District'])
KE['Sector'] = KE['Sector'].map(sector).fillna(KE['Sector'])

print(KE.head())

print(KE.columns)

plt.hist(KE['total_consumption'], bins=10, color='blue', edgecolor='black')
plt.xlabel("Consumption")
plt.ylabel("Frequency")
plt.title("Consumption Distribution in Kerala State")
plt.show()

KE_consumption = KE.groupby('District')['total_consumption'].sum().reset_index()

print(KE_consumption.head())

plt.bar(KE_consumption['District'], KE_consumption['total_consumption'], color='blue', edgecolor='black')
plt.xlabel("District")
plt.ylabel("Total Consumption")
plt.title("Total Consumption per District")
plt.xticks(rotation=90)  # Rotate district names for better visibility
plt.show()

data_map = gpd.read_file("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
```

```
print(data_map.columns)
print(KE_consumption.columns)

data_map['District'] = KE_consumption['District']

data_map_data = data_map.merge(KE_consumption, left_on='dtname', right_on='District')

print(data_map.columns)

import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt

data_map = gpd.read_file("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")

data_map = data_map.rename(columns={'dtname': 'District'})

display(data_map.rename)

KE_consumption = pd.read_csv("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson", low_memory=False)

KE_consumption = KE.groupby('District')['total_consumption'].sum().reset_index()

print(KE_consumption.head())

data_map = gpd.read_file("/Users/kirthanshaker/Desktop/SCMA 631 Data Files /KERALA_DISTRICTS (1).geojson")
data_map = data_map.rename(columns={'dtname': 'total_consumption'})
```