

VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

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

Akanksha yaramalla

V01108249

Date of Submission: 15-07-2024

CONTENTS

|  |  |  |
| --- | --- | --- |
| Sl. No. | Title | Page No. |
| 1. | Histogram | 1-2 |
| 2. | Barplot | 2-6 |
| 3. | Python and R codes(Histogram and Barplot) | 6-14 |
| 4. | References | 15 |

### **Histogram**

A **histogram** is a type of bar chart that represents the distribution of a dataset by showing the frequency of data points within specified ranges (bins). It helps to visualize the distribution of numerical data and identify patterns such as skewness, peaks, and gaps.

* **X-axis**: Represents the data values (e.g., total consumption values).
* **Y-axis**: Represents the frequency (number of occurrences) of data points within each bin.
* **Bins**: Continuous intervals that divide the range of the data values.

#### **Objectives of Histogram:**

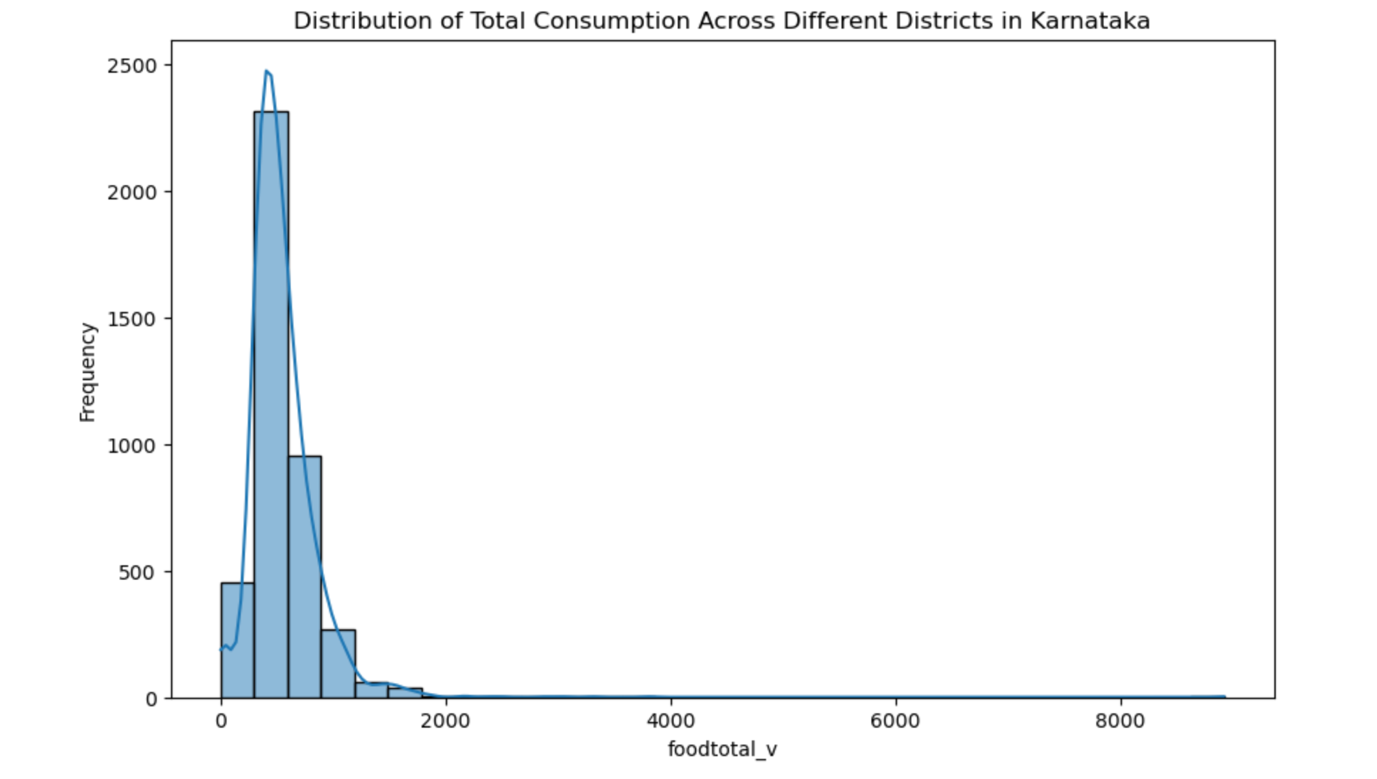
1. **Understand Distribution**: To visualize the distribution of total food consumption values across different districts.
2. **Identify Patterns**: To identify any patterns or anomalies in food consumption, such as skewness or outliers.
3. **Highlight Extremes**: To identify districts with extremely high or low food consumption values.
4. **Support Decision Making**: To provide a visual aid that supports decision-making processes for resource allocation and policy formulation.

#### **Business Significance of Histogram:**

1. **Resource Allocation**: Helps in identifying areas with higher needs for resources, allowing for more efficient distribution.
2. **Market Analysis**: Provides insights into consumption trends, aiding businesses in the food sector to target specific regions.
3. **Policy Formulation**: Assists policymakers in understanding regional disparities in food consumption, leading to more effective interventions.
4. **Identify Target Areas**: Highlights districts with unusual consumption patterns that may need further investigation or support.

**Results:**

**Distribution of Total Consumption Across Different Districts in Karnataka**



A graph of a number of consumption

Description automatically generated

**Histogram Analysis:**

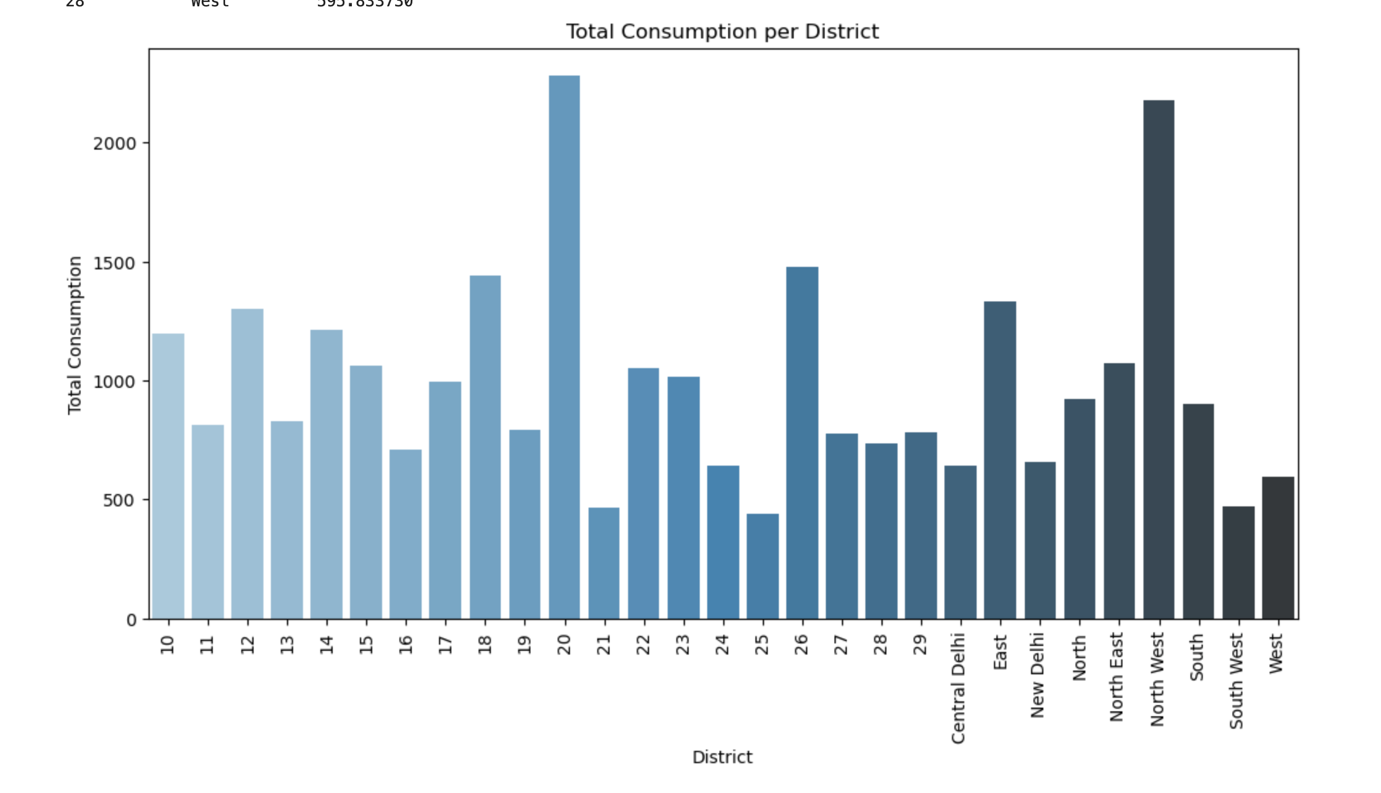
* The histogram shows the distribution of total food consumption values (foodtotal\_v) across different districts in Karnataka.
* The x-axis represents the total consumption values.
* The y-axis represents the frequency (number of occurrences) of those consumption values.
* The distribution is highly skewed to the right, indicating that most districts have lower total consumption values.
* There are very few districts with very high consumption values, creating a long tail towards the right.
* The peak of the histogram is around the lower consumption values (below 1000), indicating that the majority of the districts fall into this range.

**Inference:**

* The consumption of food items (in terms of value) is predominantly lower in most of the districts.
* Only a few districts exhibit higher consumption values, suggesting a disparity in food consumption across the districts in Karnataka.
* This could be due to various socio-economic factors influencing food consumption patterns.

**Results:**

### **Consumption per District in Karnataka**



### **Python code:**

import pandas as pd

# Load the data

file\_path = '/Users/preethireddy/Downloads/NSSO68.csv'

data = pd.read\_csv(file\_path, low\_memory=False)

# Strip whitespace from column names

data.columns = data.columns.str.strip()

# Print the cleaned column names

print(data.columns)

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

# Define the columns to be used

state\_column = 'state'

district\_column = 'State\_Region'

consumption\_column = 'foodtotal\_v' # This can be changed to 'Beveragestotal\_v' or 'fv\_tot' based on your requirement

# Filter the data for Karnataka

karnataka\_state\_code = 29 # Assuming 29 is the state code for Karnataka

karnataka\_data = data[data[state\_column] == karnataka\_state\_code]

# Replace infinite values with NaN

karnataka\_data.replace([np.inf, -np.inf], np.nan, inplace=True)

# Verify if columns exist

if district\_column in karnataka\_data.columns and consumption\_column in karnataka\_data.columns:

print(f"Columns '{district\_column}' and '{consumption\_column}' found in the dataset.")

# Drop rows with NaN values in the specified columns

karnataka\_data = karnataka\_data.dropna(subset=[consumption\_column, district\_column])

# Plotting the histogram

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

sns.histplot(karnataka\_data[consumption\_column], bins=30, kde=True)

plt.title('Distribution of Total Consumption Across Different Districts in Karnataka')

plt.xlabel('foodtotal\_v')

plt.ylabel('Frequency')

plt.show()

# Plotting the barplot

plt.figure(figsize=(14, 8))

sns.barplot(x=district\_column, y=consumption\_column, data=karnataka\_data)

plt.title('Consumption per District in Karnataka')

plt.xlabel('State\_Region')

plt.ylabel('foodtotal\_v')

plt.xticks(rotation=90) # Rotate district names for better readability

plt.show()

else:

print(f"Check the column names. Available columns are: {karnataka\_data.columns}")

------------------------------------------------------------------------------------------------------------

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import geopandas as gpd

# Set the working directory and verify it

os.chdir("/Users/bharathit/Downloads")

print("Current working directory:", os.getcwd())

# Load required libraries

def install\_and\_load(package):

import importlib

try:

importlib.import\_module(package)

except ImportError:

import pip

pip.main(['install', package])

finally:

globals()[package] = importlib.import\_module(package)

# List of required libraries

libraries = ["pandas", "numpy", "matplotlib", "seaborn", "geopandas"]

for lib in libraries:

install\_and\_load(lib)

# Reading the file into Python

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

# Filtering for KA (Karnataka)

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

# Display dataset info

print("Dataset Information:")

print(df.columns)

print(df.head())

print(df.shape)

# Finding missing values

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

print("Missing Values Information:")

print(missing\_info)

# Subsetting the data

ka\_new = 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']]

# Impute missing values with mean for specific columns

ka\_new['Meals\_At\_Home'].fillna(ka\_new['Meals\_At\_Home'].mean(), inplace=True)

# Finding outliers and removing them

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)]

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

for col in outlier\_columns:

ka\_new = remove\_outliers(ka\_new, col)

# Summarize consumption

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

# Summarize and display top consuming districts and regions

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

summary = df.groupby(group\_col)['total\_consumption'].sum().reset\_index().sort\_values(by='total\_consumption', ascending=False)

return summary

district\_summary = summarize\_consumption(ka\_new, 'District')

region\_summary = summarize\_consumption(ka\_new, 'Region')

print("Top Consuming Districts:")

print(district\_summary.head(4))

print("Region Consumption Summary:")

print(region\_summary)

# Rename districts and sectors

district\_mapping = {"1": "North West", "2": "North", "3": "North East", "4": "East", "5": "New Delhi", "6": "Central Delhi", "7": "West", "8": "South West", "9": "South"}

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

ka\_new['District'] = ka\_new['District'].astype(str)

ka\_new['Sector'] = ka\_new['Sector'].astype(str)

ka\_new['District'] = ka\_new['District'].replace(district\_mapping)

ka\_new['Sector'] = ka\_new['Sector'].replace(sector\_mapping)

# Display the updated dataframe

print(ka\_new)

# Plotting the histogram

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

plt.xlabel("Consumption")

plt.ylabel("Frequency")

plt.title("Consumption Distribution in Karnataka State")

plt.show()

# Aggregate total consumption by district

KA\_consumption = ka\_new.groupby('District')['total\_consumption'].sum().reset\_index()

print("KA\_consumption DataFrame:")

print(KA\_consumption)

# Bar plot

plt.figure(figsize=(12, 6))

sns.barplot(x='District', y='total\_consumption', data=KA\_consumption, palette='Blues\_d')

plt.xlabel("District")

plt.ylabel("Total Consumption")

plt.title("Total Consumption per District")

plt.xticks(rotation=90)

plt.show()

**R code:**

# Load necessary libraries

library(dplyr)

library(ggplot2)

library(readr)

# Load the data

file\_path <- '/Users/preethireddy /Downloads/NSSO68.csv'

data <- read\_csv(file\_path)

# Strip whitespace from column names

colnames(data) <- trimws(colnames(data))

# Print the cleaned column names

print(colnames(data))

# Define the columns to be used

state\_column <- 'state'

district\_column <- 'State\_Region'

consumption\_column <- 'foodtotal\_v' # This can be changed to 'Beveragestotal\_v' or 'fv\_tot' based on your requirement

# Filter the data for Karnataka

karnataka\_state\_code <- 29 # Assuming 29 is the state code for Karnataka

karnataka\_data <- data %>% filter(!!sym(state\_column) == karnataka\_state\_code)

# Replace infinite values with NA

karnataka\_data <- karnataka\_data %>% mutate(across(everything(), ~ifelse(. %in% c(Inf, -Inf), NA, .)))

# Verify if columns exist

if (district\_column %in% colnames(karnataka\_data) & consumption\_column %in% colnames(karnataka\_data)) {

print(paste("Columns", district\_column, "and", consumption\_column, "found in the dataset."))

# Drop rows with NA values in the specified columns

karnataka\_data <- karnataka\_data %>% drop\_na(!!sym(consumption\_column), !!sym(district\_column))

# Plotting the histogram

ggplot(karnataka\_data, aes(x = !!sym(consumption\_column))) +

geom\_histogram(bins = 30, fill = "blue", alpha = 0.7, color = "black") +

geom\_density(alpha = 0.2, fill = "red") +

labs(title = 'Distribution of Total Consumption Across Different Districts in Karnataka',

x = consumption\_column,

y = 'Frequency') +

theme\_minimal()

# Plotting the barplot

ggplot(karnataka\_data, aes(x = as.factor(!!sym(district\_column)), y = !!sym(consumption\_column))) +

geom\_bar(stat = "summary", fun = "mean", fill = "skyblue", color = "black") +

labs(title = 'Consumption per District in Karnataka',

x = 'State\_Region',

y = consumption\_column) +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))

} else {

print(paste("Check the column names. Available columns are:", paste(colnames(karnataka\_data), collapse = ", ")))

}

--------------------------------------------------------------------------------------------------

# Set the working directory and verify it

setwd("/Users/bharathit/Downloads")

print(paste("Current working directory:", getwd()))

# Load required libraries

install\_and\_load <- function(package) {

if (!require(package, character.only = TRUE)) {

install.packages(package, dependencies = TRUE)

library(package, character.only = TRUE)

}

}

# List of required libraries

libraries <- c("dplyr", "ggplot2", "sf")

lapply(libraries, install\_and\_load)

# Reading the file into R

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

# Filtering for KA (Karnataka)

df <- subset(data, state\_1 == "KA")

# Display dataset info

print("Dataset Information:")

print(colnames(df))

print(head(df))

print(dim(df))

# Finding missing values

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

print("Missing Values Information:")

print(missing\_info)

# Subsetting the data

ka\_new <- 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

ka\_new$Meals\_At\_Home[is.na(ka\_new$Meals\_At\_Home)] <- mean(ka\_new$Meals\_At\_Home, na.rm = TRUE)

# Finding outliers and removing them

remove\_outliers <- function(df, column\_name) {

Q1 <- quantile(df[[column\_name]], 0.25, na.rm = TRUE)

Q3 <- quantile(df[[column\_name]], 0.75, na.rm = TRUE)

IQR <- Q3 - Q1

lower\_threshold <- Q1 - 1.5 \* IQR

upper\_threshold <- Q3 + 1.5 \* IQR

df[df[[column\_name]] >= lower\_threshold & df[[column\_name]] <= upper\_threshold, ]

}

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

for (col in outlier\_columns) {

ka\_new <- remove\_outliers(ka\_new, col)

}

# Summarize consumption

ka\_new$total\_consumption <- rowSums(ka\_new[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(df, group\_col) {

summary <- df %>% group\_by(!!sym(group\_col)) %>% summarize(total\_consumption = sum(total\_consumption)) %>% arrange(desc(total\_consumption))

return(summary)

}

district\_summary <- summarize\_consumption(ka\_new, 'District')

region\_summary <- summarize\_consumption(ka\_new, 'Region')

print("Top Consuming Districts:")

print(head(district\_summary, 4))

print("Region Consumption Summary:")

print(region\_summary)

# Rename districts and sectors

district\_mapping <- c("1" = "North West", "2" = "North", "3" = "North East", "4" = "East", "5" = "New Delhi", "6" = "Central Delhi", "7" = "West", "8" = "South West", "9" = "South")

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

ka\_new$District <- as.character(ka\_new$District)

ka\_new$Sector <- as.character(ka\_new$Sector)

ka\_new$District <- recode(ka\_new$District, !!!district\_mapping)

ka\_new$Sector <- recode(ka\_new$Sector, !!!sector\_mapping)

# Display the updated dataframe

print(ka\_new)

# Plotting the histogram

ggplot(ka\_new, aes(x = total\_consumption)) +

geom\_histogram(bins = 10, fill = 'blue', color = 'black') +

labs(x = "Consumption", y = "Frequency", title = "Consumption Distribution in Karnataka State")

# Aggregate total consumption by district

KA\_consumption <- ka\_new %>% group\_by(District) %>% summarize(total\_consumption = sum(total\_consumption))

print("KA\_consumption DataFrame:")

print(KA\_consumption)

# Bar plot

ggplot(KA\_consumption, aes(x = District, y = total\_consumption)) +

geom\_bar(stat = "identity", fill = 'blue') +

labs(x = "District", y = "Total Consumption", title = "Total Consumption per District") +

theme(axis.text.x = element\_text(angle = 90, hjust = 1))

Reference (Datasets):

* + NSSO68.csv