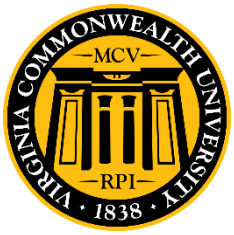


VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A5: Visualization - Perceptual Mapping for Business

JESIN KANDATHY JOY

V01110163

Date of Submission: 15-07-2024

CONTENTS

Introduction:

The analysis focuses on comparing food consumption patterns across districts in Uttar Pradesh (UP) and Karnataka (KA) using data from the NSSO 68th Round. The study examines how consumption varies geographically within each state, aiming to identify regions with higher and lower food consumption levels. This comparative analysis provides insights into regional disparities in food consumption, highlighting potential factors influencing these patterns.

Objective:

- Analyze district-level food consumption patterns within Uttar Pradesh (UP) and Karnataka (KA) using data from the NSSO 68th Round.
- Identify the top consuming districts and regions within each state to understand variations in food consumption.
- Plot a histogram and a barplot of the NSSO data to indicate the district-wise consumption of the state of Uttar Pradesh.
- Plot total consumption levels on the Karnataka state map.

Business Significance:

- Policy Formulation: Insights into consumption patterns can guide policymakers in targeting interventions for improving food security and nutrition outcomes at the regional level.
- Market Strategy: Businesses can leverage insights to tailor marketing strategies and distribution networks based on regional consumption trends, thereby enhancing market penetration and operational efficiency.
- Resource Allocation: Governments and NGOs can allocate resources more effectively to regions with higher food consumption needs, ensuring equitable distribution of food-related resources and services.

R code results:

```
# Reading the file into R
data <- read.csv("NSSO68.csv")


# a)Plotting a histogram and a barplot of the data to indicate the consumption district-wise for the Uttar Pradesh


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

# Display dataset info
cat("Dataset Information:\n")
```

```
# 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)
}
upnew$Meals_At_Home <- impute_with_mean(upnew$Meals_At_Home)

# Check for missing values after imputation
cat("Missing Values After Imputation:\n")
```

```
# 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) {
  upnew <- remove_outliers(upnew, col)
}

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

# Summarize and display top and bottom consuming districts and regions
summarize_consumption <- function(group_col) {
  summary <- upnew %>%
    group_by(across(all_of(group_col))) %>%
    summarise(total = sum(total_consumption)) %>%
    arrange(desc(total))
  return(summary)
}
district_summary <- summarize_consumption("District")
region_summary <- summarize_consumption("Region")

cat("Top 3 Consuming Districts:\n")
```

```
print(head(district_summary, 3))
```

```
## # A tibble: 3 × 2
##   District total
##      <int> <dbl>
## 1       15 1323.
## 2       11 1229.
## 3       12 1124.
```

```
cat("Bottom 3 Consuming Districts:\n")
```

```
## Bottom 3 Consuming Districts:
```

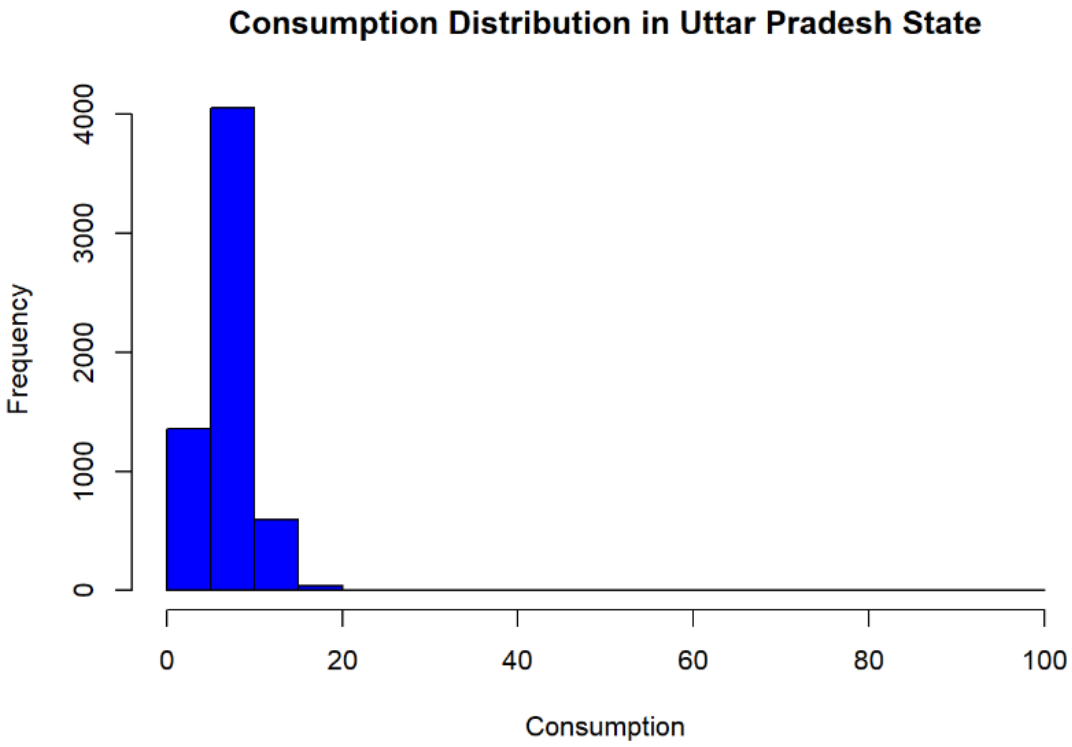
```
print(tail(district_summary, 3))
```

```
## # A tibble: 3 × 2
##   District total
##      <int> <dbl>
## 1       41  290.
## 2       48  222.
## 3       56  191.
```

```
print(region_summary)
```

```
## # A tibble: 5 × 2
##   Region  total
##    <int>  <dbl>
## 1      5 13449.
## 2      3 12949.
## 3      1  7396.
## 4      2  6495.
## 5      4  3667.
```

```
# Rename districts and sectors , get codes from appendix of NSSO 68th ROund Data
district_mapping <- c("15" = "Agra", "11" = "Bulandshahar", "12" = "Aligarh")
sector_mapping <- c("2" = "URBAN", "1" = "RURAL")

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

# up_consumption stores the aggregate of the consumption district wise
up_consumption <- aggregate(total_consumption ~ District, data = upnew, sum)
View(up_consumption)

# histogram to show the distribution of total consumption across different districts
hist(upnew$total_consumption, breaks = 15, col = 'blue', border = 'black',
     xlab = "Consumption", ylab = "Frequency", main = "Consumption Distribution in Uttar Pradesh State")
```

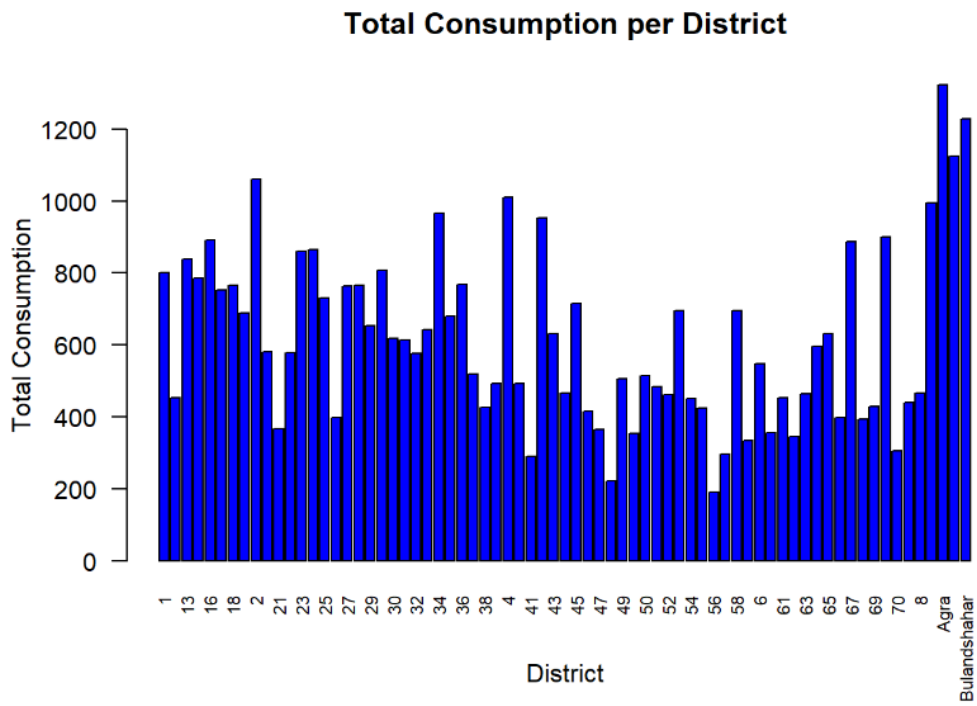



```
# barplot to visualize consumption per district with district names
??barplot
```

```
## starting httpd help server ...
```

```
##  done
```

```
barplot(up_consumption$total_consumption,
        names.arg = up_consumption$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)
```



```
# b) Plotting total consumption on the Karnataka state map


# Filtering for Karnataka
df_ka <- data %>%
  filter(state_1 == "KA")

# Sub-setting the data
ka_new <- df_ka %>%
  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)

# Check for missing values in the subset
cat("Missing Values in Subset:\n")
```

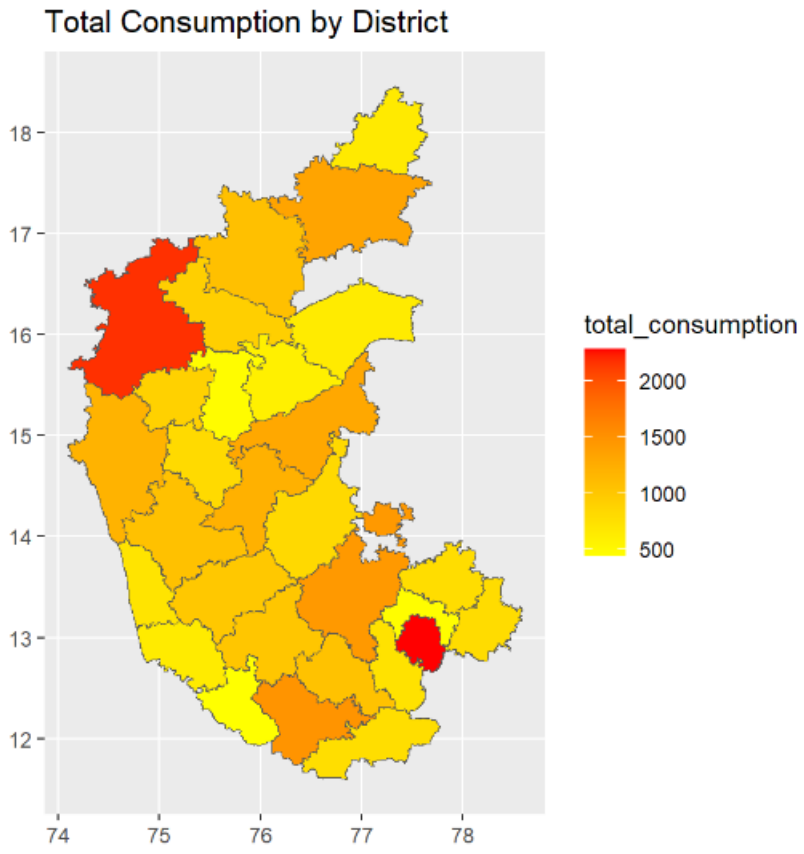
```
# Impute missing values with mean for specific columns
ka_new$Meals_At_Home <- impute_with_mean(ka_new$Meals_At_Home)

# Check for missing values after imputation
cat("Missing Values After Imputation:\n")
```

```
# Finding outliers and removing them
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 = TR
UE)

district_summary <- summarize_consumption("District")
cat("District Consumption Summary:\n")
```

```r
# mapping districts so that meging of the tables will be easier
district_mapping <- c(
  "1" = "Belagavi",
  "2" = "Bagalkote",
  "3" = "Vijayapura",
  "4" = "Kalaburagi",
  "5" = "Bidar",
  "6" = "Raichur",
  "7" = "Koppal",
  "8" = "Gadag",
  "9" = "Dharwad",
  "10" = "Uttara Kannada",
  "11" = "Haveri",
  "12" = "Ballari",
  "13" = "Chitradurga",
  "14" = "Davanagere",
  "15" = "Shivamogga",
  "16" = "Udupi",
  "17" = "Chikkamagaluru",
  "18" = "Tumakuru",
  "19" = "Kolar",
  "20" = "Bangalore",
  "21" = "Bengaluru Rural",
  "22" = "Mandya",
  "23" = "Hassan",
  "24" = "Dakshina Kannada",
  "25" = "Kodagu",
  "26" = "Mysuru",
  "27" = "Chamarajanagara",
  "28" = "Ramanagara",
  "29" = "Chikkaballapura"
)
```

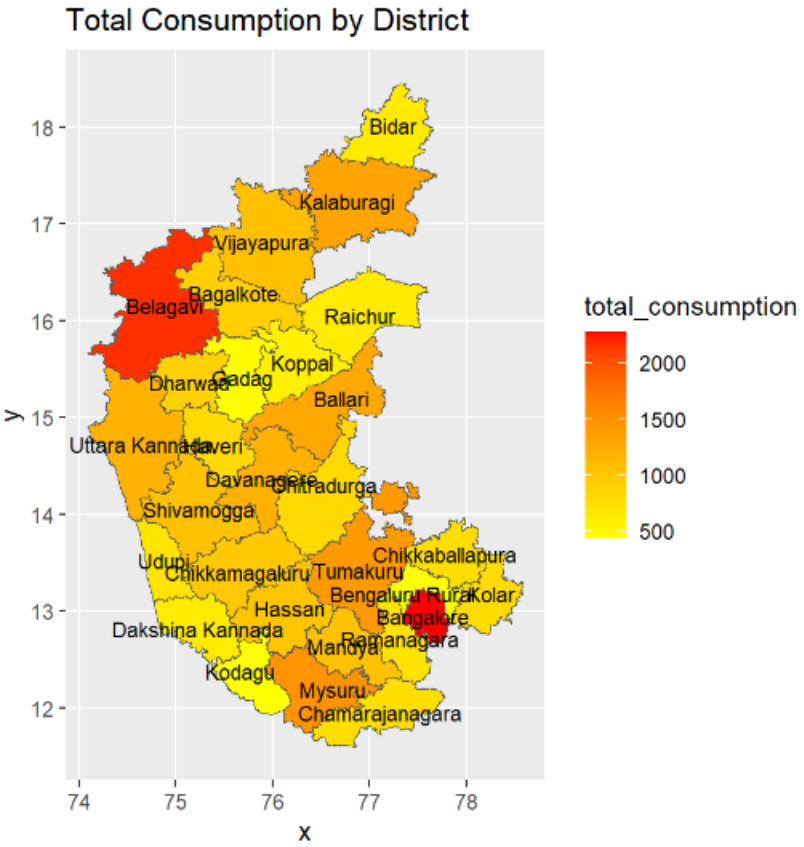
```r
ka_new$District <- as.character(ka_new$District)
ka_new$District <- district_mapping[ka_new$District]
#ka_new$District <- ifelse(ka_new$District %in% names(district_mapping), district_mapping[ka_new$District], ka_new$District)
View(ka_new)

# ka_consumption stores aggregate of total consumption district wise
ka_consumption <- aggregate(total_consumption ~ District, data = ka_new, sum)
View(ka_consumption)

#Plotting total consumption on the Karnataka state

Sys.setenv("SHkaE_RESTORE_SHX" = "YES")

data_map <- st_read("E:\\JESIN\\DOCUMENTS\\scma\\A5\\KARNATAKA_DISTRICTS.geojson")
```

```r
View(data_map)

data_map <- data_map %>%
  rename(District = dtname)

# merging ka_consumption and data_map tables
data_map_data <- merge(ka_consumption,data_map,by = "District")
View(data_map_data)

# Plot without labeling district names
ggplot(data_map_data) +
  geom_sf(aes(fill =total_consumption, geometry = geometry)) +
  scale_fill_gradient(low = "yellow", high = "red") +
  ggtitle("Total Consumption by District")
```



```r
# Plot with labelled district names
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 in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```



Python code results:

```python
# a) Plotting a histogram and a barplot of the data to indicate the consumption district-wise for Uttar Pradesh

# Filtering for UP
df = data[data['state_1'] == "UP"]

# Display dataset info
print("Dataset Information:")
print(df.columns)
print(df.head())
print(df.shape)
```

```
# Sub-setting the data
upnew = 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']]

# Check for missing values in the subset
print("Missing Values in Subset:")
print(upnew.isnull().sum())

# Impute missing values with mean for specific columns
upnew['Meals_At_Home'].fillna(upnew['Meals_At_Home'].mean(), inplace=True)

# Check for missing values after imputation
print("Missing Values After Imputation:")
print(upnew.isnull().sum())
```

```
# Function to remove outliers
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:
    upnew = remove_outliers(upnew, col)
```

```
# Summarize consumption
upnew['total_consumption'] = upnew[['ricepds_v', 'Wheatpds_q', 'chicken_q', 'pulsep_q', 'wheatos_q']].sum(axis=1)

# Summarize and display top and bottom consuming districts and regions
district_summary = upnew.groupby('District')['total_consumption'].sum().reset_index().sort_values(by='total_consumption', ascending=False)
region_summary = upnew.groupby('Region')['total_consumption'].sum().reset_index().sort_values(by='total_consumption', ascending=False)

print("Top 3 Consuming Districts:")
print(district_summary.head(3))
print("Bottom 3 Consuming Districts:")
print(district_summary.tail(3))

print("Region Consumption Summary:")
print(region_summary)
```

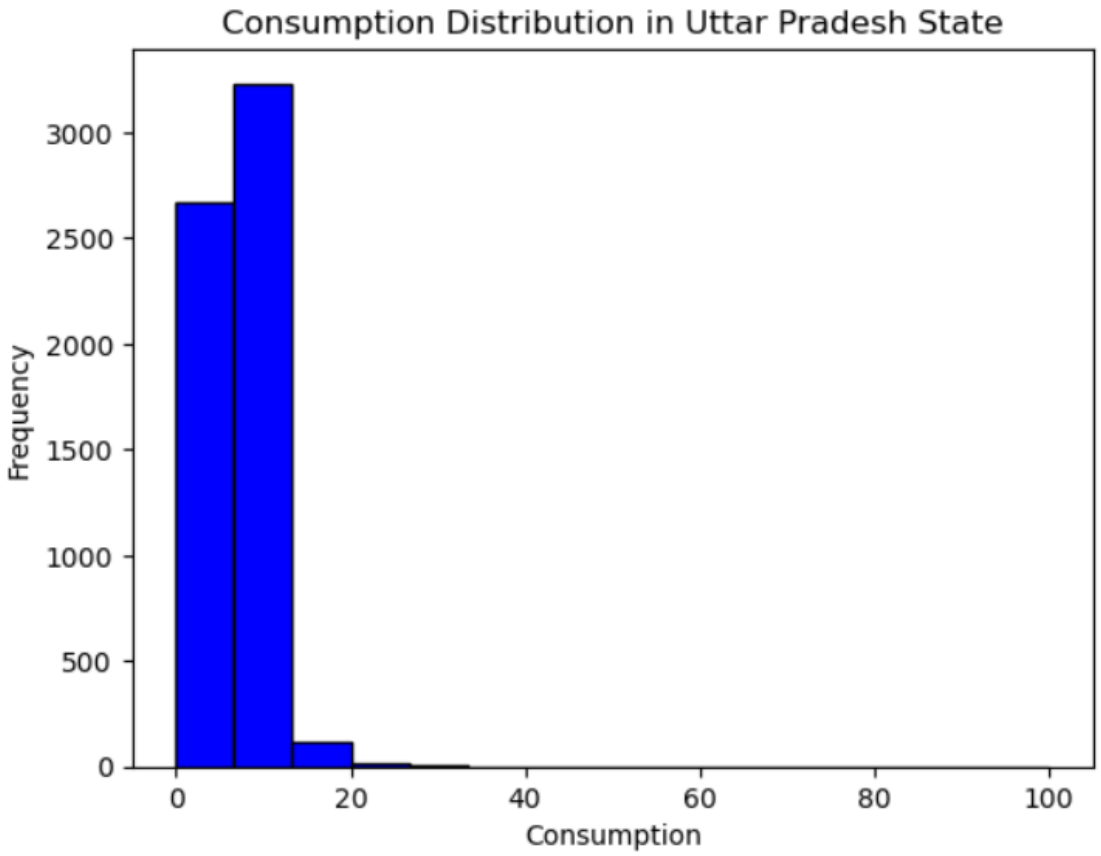
```
Top 3 Consuming Districts:
    District  total_consumption
14        15        1323.015188
10        11        1228.852129
11        12        1124.361810
Bottom 3 Consuming Districts:
    District  total_consumption
40        41         289.548810
47        48         221.714015
55        56         190.502381
Region Consumption Summary:
   Region  total_consumption
4       5       13449.489887
2       3       12949.177897
0       1        7396.093142
1       2        6494.791811
3       4        3667.097532
```

```
# Rename districts and sectors
district_mapping = {"15": "Agra", "11": "Bulandshahar", "12": "Aligarh"}
sector_mapping = {"2": "URBAN", "1": "RURAL"}

upnew['District'] = upnew['District'].astype(str).map(district_mapping).fillna(upnew['District'])
upnew['Sector'] = upnew['Sector'].astype(str).map(sector_mapping).fillna(upnew['Sector'])
print(upnew)
```

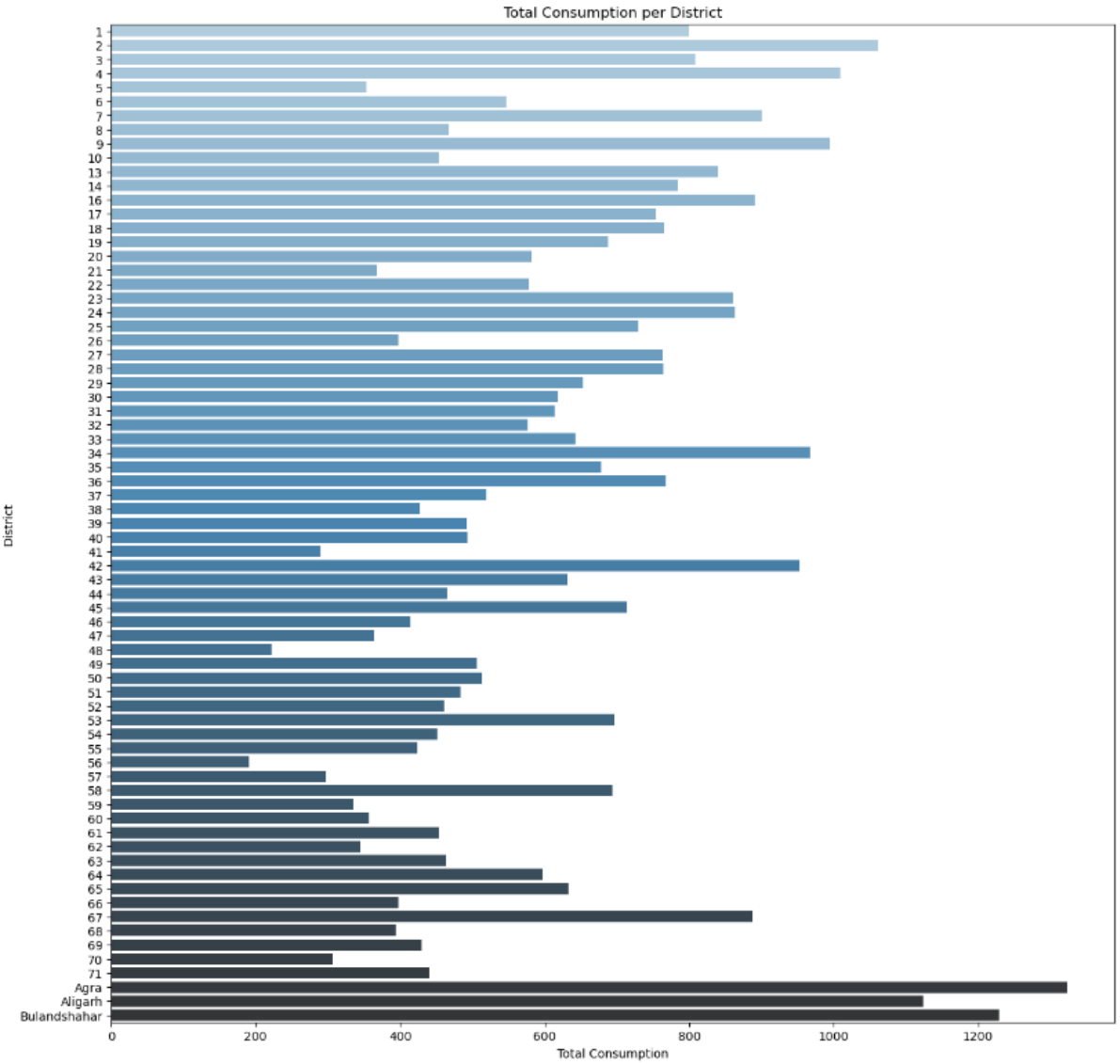
```
# up_consumption stores the aggregate of the consumption district-wise
up_consumption = upnew.groupby('District')['total_consumption'].sum().reset_index()
print(up_consumption)
```

```
        District  total_consumption
0              1         800.083145
1              2        1061.463651
2              3         808.415093
3              4        1009.743272
4              5         353.268452
..           ...                ...
66            70         306.074747
67            71         440.595996
68          Agra        1323.015188
69       Aligarh        1124.361810
70  Bulandshahar        1228.852129
```

```
# Histogram to show the distribution of total consumption across different districts
plt.hist(upnew['total_consumption'], bins=15, color='blue', edgecolor='black')
plt.xlabel('Consumption')
plt.ylabel('Frequency')
plt.title('Consumption Distribution in Uttar Pradesh State')
plt.show()
```



```
# Barplot to visualize consumption per district with district names
plt.figure(figsize=(15, 15))
sns.barplot(x='total_consumption', y='District', data=up_consumption, palette='Blues_d')
plt.xlabel('Total Consumption')
plt.ylabel('District')
plt.title('Total Consumption per District')
plt.show()
```




```
# b) Plotting total consumption on the Karnataka state map

# Filtering for Karnataka
df_ka = data[data['state_1'] == "KA"]

# Sub-setting the data
ka_new = df_ka[['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']]
```

```
# Check for missing values in the subset
print("Missing Values in Subset:")
print(ka_new.isnull().sum())

# Impute missing values with mean for specific columns
ka_new['Meals_At_Home'].fillna(ka_new['Meals_At_Home'].mean(), inplace=True)

# Check for missing values after imputation
print("Missing Values After Imputation:")
print(ka_new.isnull().sum())
```

```
# Remove outliers
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)

district_summary = ka_new.groupby('District')['total_consumption'].sum().reset_index().sort_values(by='total_consumption', ascending=False)
print("District Consumption Summary:")
print(district_summary)
```

```
# Mapping districts so that merging of the tables will be easier
district_mapping = {
    "1": "Belagavi", "2": "Bagalkote", "3": "Vijayapura", "4": "Kalaburagi", "5": "Bidar",
    "6": "Raichur", "7": "Koppal", "8": "Gadag", "9": "Dharwad", "10": "Uttara Kannada",
    "11": "Haveri", "12": "Ballari", "13": "Chitradurga", "14": "Davanagere", "15": "Shivamogga",
    "16": "Udupi", "17": "Chikkamagaluru", "18": "Tumakuru", "19": "Kolar", "20": "Bangalore",
    "21": "Bengaluru Rural", "22": "Mandya", "23": "Hassan", "24": "Dakshina Kannada",
    "25": "Kodagu", "26": "Mysuru", "27": "Chamarajanagara", "28": "Ramanagara", "29": "Chikkaballapura"
}

ka_new['District'] = ka_new['District'].astype(str).map(district_mapping).fillna(ka_new['District'])
print(ka_new)
```

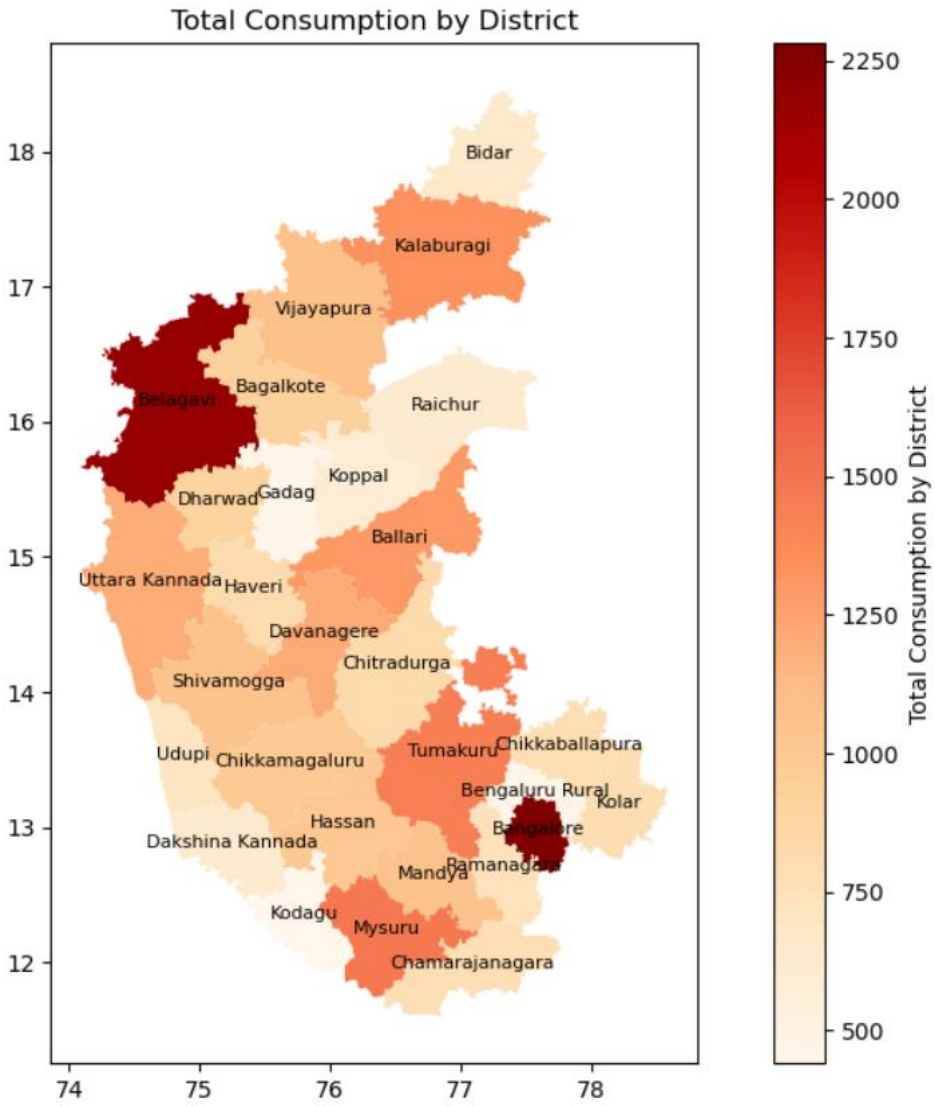
```
# ka_consumption stores aggregate of total consumption district-wise
ka_consumption = ka_new.groupby('District')['total_consumption'].sum().reset_index()
print(ka_consumption)
```

	District	total_consumption
0	Bagalkote	923.939246
1	Ballari	1302.404203
2	Bangalore	2281.357870
3	Belagavi	2174.372053
4	Bengaluru Rural	465.970635
5	Bidar	657.904545
6	Chamarajanagara	777.135595
7	Chikkaballapura	781.763333
8	Chikkamagaluru	992.455833
9	Chitradurga	827.296829
10	Dakshina Kannada	641.593523
11	Davanagere	1214.228730
12	Dharwad	901.403968
13	Gadag	468.564448
14	Hassan	1015.792560
15	Haveri	812.777516
16	Kalaburagi	1332.916755
17	Kodagu	440.578030
18	Kolar	792.061729
19	Koppal	595.833730
20	Mandya	1053.904167
21	Mysuru	1479.373753
22	Raichur	641.353694
23	Ramanagara	736.295310
24	Shivamogga	1059.634816
25	Tumakuru	1441.823070
26	Udupi	709.974567
27	Uttara Kannada	1198.843083
28	Vijayapura	1074.834615

```
# Load and plot Karnataka state map
data_map = gpd.read_file("E:\\JESIN\\DOCUMENTS\\scma\\A5\\KARNATAKA_DISTRICTS.geojson")

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

```
# Merging ka_consumption and data_map tables
data_map_data = data_map.merge(ka_consumption, on='District')
print(data_map_data)
```

```
# Plot with labeled district names
fig, ax = plt.subplots(1, 1, figsize=(12, 8))
data_map_data.plot(column='total_consumption', cmap='OrRd', legend=True, ax=ax, legend_kwds={'label': "Total Consumption by District"})
data_map_data.apply(lambda x: ax.annotate(text=x['District'], xy=x.geometry.centroid.coords[0], ha='center', fontsize=8, color='black'), axis=1)
plt.title('Total Consumption by District')
plt.show()
```



Interpretations:

1. Uttar Pradesh (UP) Consumption Analysis:

- Top Consuming Districts: Districts with codes 15, 11, and 12 (mapped to Agra, Bulandshahar, and Aligarh) have the highest total consumption. These districts likely have higher population densities or greater access to food resources.
- Bottom Consuming Districts: Districts with codes 41, 48, and 56 have the lowest consumption, indicating potential issues such as lower population, limited access to food, or economic constraints affecting consumption patterns.
- Region 5 and 3: These regions have significantly higher total consumption compared to others. This suggests they are more resource-rich or have larger populations.
- Region 4: Has the lowest consumption, possibly indicating areas that may need more support in terms of food supply and resource allocation.
- Histogram: Shows the frequency distribution of total consumption across districts, highlighting the variability in consumption levels.
- Barplot: Visualizes total consumption per district, making it easy to identify districts with exceptionally high or low consumption.

2. Karnataka (KA) Consumption Analysis:

- Top Consuming Districts: Bangalore and Belagavi have the highest total consumption. Bangalore, being a major urban center, likely has a higher population and greater economic activity, leading to higher consumption.
- Lower Consuming Districts: Bengaluru Rural and Kodagu have lower total consumption. These areas might be more rural or less densely populated, impacting overall consumption levels.
- Balanced Consumption: Districts like Mysuru, Tumakuru, and Ballari have moderate to high consumption, indicating a relatively balanced distribution of resources.
- State Map: Visualizes total consumption across districts, providing a clear spatial understanding of consumption patterns.

Recommendations:

- Resource Distribution: Implement targeted food assistance and infrastructure improvements in low-consumption districts to ensure a consistent food supply.
- Support in High Consumption Areas: Maintain and monitor robust supply chains in high-consumption districts to prevent shortages and manage demand effectively.
- Nutritional Education: Educate communities on balanced nutrition and integrate nutritional education into schools to promote healthy eating habits.
- Data Monitoring: Regularly collect and analyze food consumption data to inform policy decisions and address disparities.