VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS AND MODELLING (SCMA 632)

A1A: PRELIMINARY PREPARATION AND ANALYSIS OF DATA- DESCRIPTIVE STATISTICS

AYONA BIJU

V01151327

DATE OF SUBMISSION: 12-06-2025

CONTENTS

Sl. No.	Title	Page No.
1	INTRODUCTION	3
2	RESULTS AND INTERPRETATION	4-13
3	CODES	13-18
4	References	18

ANALYSIS OF CONSUMPTION IN BIHAR USING PYTHON

INTRODUCTION

The focus of this study is on the state of Bihar, from the NSSO data, to find the top and bottom four consuming districts of Bihar. In the process, we manipulate and clean the dataset to get more suitable data to analyse. To facilitate this analysis, we have gathered a dataset containing consumption-related information, including data on rural and urban sectors, as well as district-wise variations. The dataset has been imported into python , a powerful statistical programming language renowned for its versatility in handling and analysing large datasets.

OBJECTIVES

Our objectives include identifying missing values, addressing outliers, standardizing district and sector names, summarizing consumption data regionally and district-wise, and testing the significance of mean differences. The findings from this study can inform policymakers and stakeholders, fostering targeted interventions and promoting equitable development across the state.

BUSINESS SIGNIFICANCE

The study focuses on the consumption pattern of Bihar based on the NSSO dataset. This is helpful in providing valuable information about market entry, supply chain developments and knowing the top and bottom three consuming districts in Bihar can help businesses and policymakers to focus more on these sectors respectively.

RESULTS AND INTERPRETATION

1) Check if there are any missing values in the data, identify them, and if there are, replace them with the mean of the variable
   Code and Result:
   Identifying missing values

INPUT

```
print ("Missing Values Before Imputation:\n")
print(state_subset.isna().sum())
```

OUTPUT

Missing Values Before Imputation

```
state_1              0
District             0
Region               0
Sector               0
State_Region         0
Meals_At_Home       20
ricetotal_v          0
wheattotal_v         0
Milktotal_v          0
pulsestot_v          0
nonvegtotal_v        0
fruitstt_v           0
No_of_Meals_per_day  4
dtype: int64
```

Interpretation:

From the cleaned data of Bihar from the NSSO dateset, we come to know that Meals_At_Home has 20 and no of meals per day have 4 missing values that that can skew interpretation and hinder decision making process. Thus we replace the missing values with the mean of the variable using the above code.

Imputing missing value with mean of variable

INPUT

```
state_cleaned=state_subset.fillna(state_subset.mean (numeric_only=True))

print("\n missing values after imputation:\n")

print(state_cleaned.isna().sum())
```

OUTPUT

Missing values after imputation:

```
state_1              0
District             0
Region               0
Sector               0
State_Region         0
Meals_At_Home        0
ricetotal_v          0
wheattotal_v         0
Milktotal_v          0
pulsestot_v          0
nonvegtotal_v        0
fruitstt_v           0
No_of_Meals_per_day  0
dtype: int64
```

Interpretation:

Now the missing values have been replaced by the mean of the variable. This will help to get a more accurate analysis of the dataset.

Code and Result:

Boxplot can be used to find the outliers of the dataset

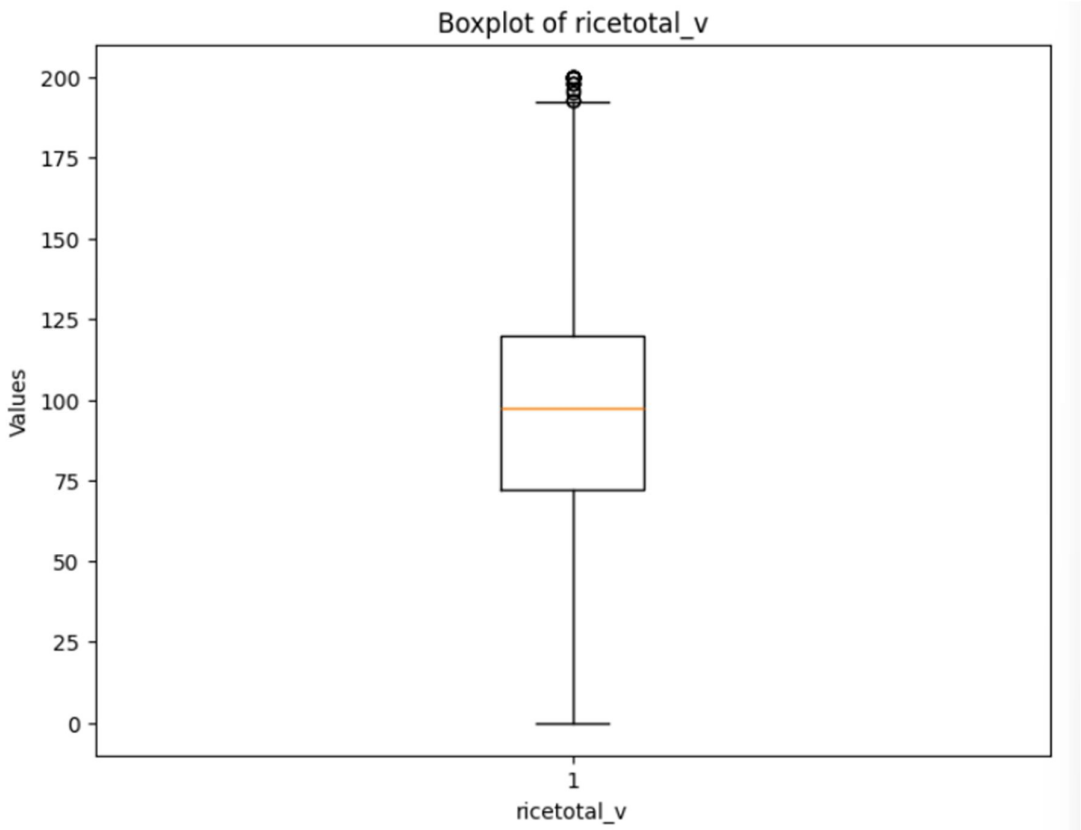
Checking for outliers

Setting quartiles and removing outliers

INPUT

```
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=Q1-1.5IQR upper_threshold=Q3+1.5IQR
return
df[(df[column_name]>=lower_threshold)&(df[column_name] <=upper_threshold)]
outlier_columns=['Meals_At_Home', 'ricetotal_v',
'wheattotal_v', 'Milktotal_v', 'pulsestot_v', 'nonvegtotal_v',
'fruitstt_v', 'No_of_Meals_per_day']
for col in outlier_columns:
state_cleaned=remove_outliers(state_cleaned,col)
print("\n columns in the cleaned dataset:")
print(state_cleaned.columns.tolist())
Plotting on boxplot
import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
plt.boxplot(state_cleaned['ricetotal_v']) plt.xlabel('ricetotal_v')
plt.ylabel('Values') plt.title('Boxplot of ricetotal_v') plt.show()
```

output



Interpretation:

The variable ricetotal_v is moderately concentrated around 100 units with a few high-value outliers above 190. Most of the data falls between 75 and 125, and the presence of

outliers suggests some extreme consumption or measurement cases that deviate from the rest of the dataset.

Renaming the districts and sector, rural and urban.

Based on the NSSO data the districts are numbered and the sectors rural and urban sector are numbered 2 and 1 respectively. In order to find the top and bottom consuming districts and for ease of understanding for the reader the numbers have been replaced by the district and sector names using the following code.

Code and Result:

INPUT

```
state_cleaned['District']= state_cleaned['District'].astype(str)

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

district_mapping={ "28"=" Patna", " 19"=" Samastipur" ,"13"=" Darbhanga",  “35"=" Gaya" , " 34"=" Aurangabad" ," 33"=" Jehanabad" , "3"=" Sheohar ", "38"=" Arwal")


sector_mapping={"2":"URBAN","1":"RURAL"}

state_cleaned['District']=state_cleaned['District'].map(district
_mapping).fillna(state_subset['District'])

state_cleaned['Sector']=state_cleaned['Sector'].map(sector_
mapping).fillna(state_subset['Sector'])

district_summary=

summarize_consumption(state_cleaned,'District')

region_summary=

summarize_consumption(state_cleaned,'Region')

sector_summary=

summarize_consumption(state_cleaned,'Sector')

print("/n updated district summary(after mapping):")

print(district_summary.head(4))
```

```
print("/n region summary:")

print(region_summary)

print("/n sector summary:")

print(sector_summary)
```

Output

```
  sector consumption summary:
   Sector  total_consumption
0  RURAL      773467.494205
1  URBAN      268457.324192
```

Now the districts have been renamed. Also the sectors have been replaced with rural and urban name.

Summarize the critical variables in the data set region-wise and district-wise and indicate the top and bottom three districts of consumption.

Code and Result:

Top four consuming districts

INPUT

```
print(district_summary.head(4))
```

OUTPUT

```
 top 4 consuming districts:
    District  total_consumption
21        28       49935.634789
15        19       42854.125036
11        13       40710.226226
26        35       40233.655844
```

Interpretation:

The top four ṁ consuming districts are Patna(28) at 49935.63. Units, followed by Samastipur(19) at 42854.12 units and in third place is Darbhanga(13) with 40710.22units. and in fourth place is Gaya (35) with 40233.65

Bottom four consuming districts

INPUT

print(district_summary.tail(4))

OUTPUT

bottom 4 consuming districts

	District	total_consumption
33	34	18556.344877
32	33	17727.281471
2	3	15485.817063
37	38	13939.775758

Interpretation:

Similarly when we look at the least consuming district we have Aurangabad(34) at 118556.34 units followed by Jehanabad(33) at 17727.28 units and Sheohar(3) at 15485.81 units and the fourth one is Arwal(38) at 13939.77 units.

Test whether the differences in the means are significant or not.

Z test is performed for the large set of data to check whether the differences in mean are significant or not.

Code and Result:

INPUT

```
consumption_rural = state_cleaned[state_cleaned['Sector'] ==
'RURAL']['total_consumption']

consumption_urban = state_cleaned[state_cleaned['Sector'] ==
'URBAN']['total_consumption']

z_statistic, p_value = stests.ztest(consumption_rural, consumption_urban, alternative='two-
sided')

print(" Z-Test for Rural vs Urban Consumption")

print("Z-Score:", round(z_statistic, 4))

print("P-Value:", round(p_value, 4))

if p_value < 0.05:

print(" Significant difference between Rural and Urban mean consumption (Reject Ho)")

or

print(" No significant difference between Rural and Urban mean consumption (Fail to reject
Ho)")
```

OUTPUT

```
 Z-Test for Rural vs Urban Consumption
Z-Score: -2.2701
P-Value: 0.0232
 Significant difference between Rural and Urban mean consumption (Reject Ho)
```

Interpretation:

The Z-test compares the mean consumption between rural and urban areas. The calculated Z-score of -2.2701 with a p-value of 0.0232 indicates a significant difference between the two groups. Since the p value is less than 0.05, we reject the null hypothesis (H_o). This suggests that rural and urban consumption patterns are significantly different.

Z test for finding significance of difference in mean of top and bottom districts consumption

INPUT

```
top_district = state_cleaned[state_cleaned['District'] ==
district_summary.head(1).iloc[0]['District']]['total_consumpti on']


bottom_district = state_cleaned[state_cleaned['District'] ==
district_summary.tail(1).iloc[0]['District']]['total_consumption' ]


z_statistic, p_value = stests.ztest(consumption_rural, consumption_urban, alternative='two-
sided')

print("Z test for top and bottom Consumption")

print("Z-Score:", round(z_statistic, 4))

print("P-Value:", round(p_value, 4))

if p_value < 0.05:

i print(f" Significant difference between {district_summary.head(1).iloc[0]['District']} and
{district_summary.tail(1).iloc[0]['District']} mean consumption (Reject Ho)")

or

print(f" No significant difference between {district_summary.head(1).iloc[0]['District']} and
{district_summary.tail(1).iloc[0]['District']} mean consumption (Fail to reject Ho)")
```

OUTPUT

```
Z test for top and bottom Consumption
Z-Score: -2.2701
P-Value: 0.0232
 Significant difference between 28 and 38 mean consumption (Reject Ho)
```

Interpretation:

The Z-test was conducted to compare the mean consumption between Patna (top) and Aurangabad (bottom) districts. The Z-score of -2.2701 and a p-value of 0.0232 indicate a significant difference between their consumption levels. Since the p-value is well below 0.05, we reject the null hypothesis (H_o). This confirms that Patna and Aurangabad have Significantly different mean consumption levels.

RECOMMENDATIONS

The data filtered from the NSSO data gives a comprehensive look into the household consumption patterns and sizes within the state of Bihar. The statistical analysis performed has brought out significant incites into the outliers present in the data, the top and bottom districts and sectors consumption as well as whether a significance difference in mean exists or not between the sectors and the top and bottom districts. This information can be used by businesses and policymakers for making appropriate changes to their business and well as helps with decision making processes.

CODES

Setting up directory and installing packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.stats import weightstats as stests
```

Reading the file into python

```
df=pd.read_csv("/content/NSSO68 (1).csv", encoding="Latin-1", low_memory=False)
df.head()
```

Filtering for Bihar

```
state_data=df[df['state_1']=="Bhr"]
state_data.to_csv("../Filtered_state_data.csv", index=False)
print("Dataset Information:\n")
print("Column Names:")
print(state_data.columns.tolist())
print("\nFirst 5 rows:")
print(state_data.head())
print("\nDimensions(rows,columns):")
print(state_data.shape)
```

```
print("\nTotal Missing Values:")
print(state_data.isna().sum().sum())
```

Finding issuing values , sub setting it and replacing it with mean of the variables

```
state_data.isnull().sum().sort_values(ascending=False)
state_subset = state_data[[ 'state_1', 'District', 'Region',
                    'Sector', 'State_Region', 'Meals_At_Home', 'ricetotal_v',
                     'wheattotal_v', 'Milktotal_v', 'pulsestot_v', 'nonvegtotal_v',
                     'fruitstt_v', 'No_of_Meals_per_day' ]]
print("Missing Values Before Imputation:\n")
print(state_subset.isna().sum())
state_cleaned=state_subset.fillna(state_subset.mean(numeric_only=True))
print("\n missing values after imputation:\n")
print(state_cleaned.isna().sum())
```

Finding outlier and making amendments(boxplot)

```
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=['Meals_At_Home', 'ricetotal_v', 'wheattotal_v',
          'Milktotal_v', 'pulsestot_v', 'nonvegtotal_v', 'fruitstt_v',
          'No_of_Meals_per_day']
for col in outlier_columns:
```

```
  state_cleaned=remove_outliers(state_cleaned,col)
  print("\n columns in the cleaned dataset:")
  print(state_cleaned.columns.tolist())
import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
plt.boxplot(state_cleaned['ricetotal_v'])
plt.xlabel('ricetotal_v')
plt.ylabel('Values')
plt.title('Boxplot of ricetotal_v')
plt.show()
```

Creating new column called total consumption

```
state_cleaned['total_consumption']=state_cleaned[['ricetotal_v',
'wheattotal_v', 'Milktotal_v', 'pulsestot_v', 'nonvegtotal_v',
'fruitstt_v']].sum(axis=1)
```

Summarising top and bottom consuming districts, region and sector

```
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
district_summary= summarize_consumption(state_cleaned,'District')
region_summary= summarize_consumption(state_cleaned,'Region')
sector_summary= summarize_consumption(state_cleaned,'Sector')
print("\n top 4 consuming districts:")
print(district_summary.head(4))
```

```
print("\n region consumption summary:")

print(region_summary)

print("\n sector consumption summary:")

print(sector_summary)

print("\n bottom 4 consuming districts")

print(district_summary.tail(4))
```

Renaming district s and sector, viz ,rural and urban

```
state_cleaned['District']= state_cleaned['District'].astype(str)

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

district_mapping={ "28"=" Patna", " 19"=" Samastipur" ,"13"=" Darbhanga",  “35"=" Gaya" , " 34"=" Aurangabad" ," 33"=" Jehanabad" , "3"=" Sheohar ", "38"=" Arwal")

sector_mapping={"2":"URBAN","1":"RURAL"}

state_cleaned['District']=state_cleaned['District'].map(district_mapping).fillna(state_subset['District'])

state_cleaned['Sector']=state_cleaned['Sector'].map(sector_mapping).fillna(state_subset['Sector'])

district_summary= summarize_consumption(state_cleaned,'District')

region_summary= summarize_consumption(state_cleaned,'Region')

sector_summary= summarize_consumption(state_cleaned,'Sector')

print("\n top 4 consuming districts:")

print(district_summary.head(4))

print("\n region consumption summary:")

print(region_summary)

print("\n sector consumption summary:")

print(sector_summary.head(4))

print("\n bottom 4 consuming districts")

print(district_summary.head(4))

first_district=district_summary.head(2).iloc[0]['District']
```

```
last_district=district_summary.head(2).iloc[0]['District']

last_district

consumption_rural = state_cleaned[state_cleaned['Sector'] ==
'RURAL']['total_consumption']

consumption_urban = state_cleaned[state_cleaned['Sector'] ==
'URBAN']['total_consumption']
```

z test to find if any significance difference in mean exists between rural and urban consumption

```
z_statistic, p_value = stests.ztest(consumption_rural, consumption_urban,
alternative='two-sided')

print(" Z-Test for Rural vs Urban Consumption")

print("Z-Score:", round(z_statistic, 4))

print("P-Value:", round(p_value, 4))

  print(" Significant difference between Rural and Urban mean consumption (Reject
Ho)")

else: print(" No significant difference between Rural and Urban mean consumption
(Fail to reject Ho)")
```

z test to find if any significance difference in mean exists between top and bottom district consumption

```
top_district = state_cleaned[state_cleaned['District'] ==
district_summary.head(1).iloc[0]['District']]['total_consumption']

bottom_district = state_cleaned[state_cleaned['District'] ==
district_summary.tail(1).iloc[0]['District']]['total_consumption']

z_statistic, p_value = stests.ztest(consumption_rural,

consumption_urban, alternative='two-sided')

print("Z test for top and bottom Consumption")

print("Z-Score:", round(z_statistic, 4))

print("P-Value:", round(p_value, 4))

if p_value < 0.05:
```

```
        print(f" Significant difference between {district_summary.head(1).iloc[0]['District']} and {district_summary.tail(1).iloc[0]['District']} mean consumption (Reject $H_o$)")
    else:
        print(f" No significant difference between {district_summary.head(1).iloc[0]['District']} and {district_summary.tail(1).iloc[0]['District']} mean consumption (Fail to reject $H_o$)")
```

References

https://github.com/scma-632/scma632 - A1 /tree/ main