```
import os, pandas as pd, numpy as np
```

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

```
df.head()
```

```
   slno       grp  Round_Centre  FSU_number  Round  Schedule_Number  Sample  Sector
0     1  4.10E+31             1       41000     68               10       1       2
1     2  4.10E+31             1       41000     68               10       1       2
2     3  4.10E+31             1       41000     68               10       1       2
3     4  4.10E+31             1       41000     68               10       1       2
4     5  4.10E+31             1       41000     68               10       1       2
5 rows × 384 columns
```

```
MP = df[df['state_1']=="MP"]
```

```
MP.isnull().sum().sort_values(ascending = False)
```

```
soyabean_q             4717
soyabean_v             4717
Land_Leased_out        4562
Otherwise_possessed    4552
Meals_Employer         4472
                       ...
cauli_q                   0
parwal_q                  0
bhindi_q                  0
chillig_q                 0
fv_tot                    0
Length: 384, dtype: int64
```

```
df.columns
```

```
Index(['slno', 'grp', 'Round_Centre', 'FSU_number', 'Round', 'Schedule_Number',
       'Sample', 'Sector', 'state', 'State_Region',
       ...
       'pickle_v', 'sauce_jam_v', 'Othrprocessed_v', 'Beveragestotal_v',
       'foodtotal_v', 'foodtotal_q', 'state_1', 'Region', 'fruits_df_tt_v',
       'fv_tot'],
      dtype='object', length=384)
```

```
MP_new = MP[['state_1', 'District', 'Sector','Region','State_Region','ricetotal_q','wheattotal_q','moong_q','Milktotal_q','chicken_q','
```

```
MP_new.isnull().sum().sort_values(ascending = False)
```

```
Meals_At_Home       26
state_1              0
District             0
Sector               0
Region               0
State_Region         0
ricetotal_q          0
wheattotal_q         0
moong_q              0
Milktotal_q          0
chicken_q            0
bread_q              0
foodtotal_q          0
Beveragestotal_v     0
dtype: int64
```

```
MP_clean = MP_new.copy()
```

```
MP_clean.loc[:, 'Meals_At_Home'] = MP_clean['Meals_At_Home'].fillna(MP_new['Meals_At_Home'].mean())
```

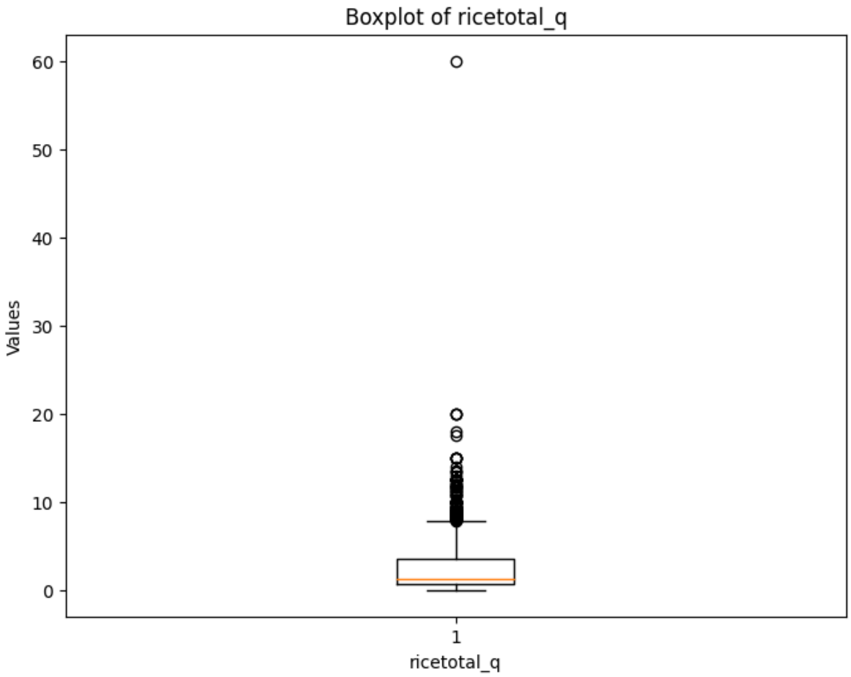
```
MP_clean.isnull().any()
```

```
state_1             False
District            False
Sector              False
Region              False
State_Region        False
```

```
ricetotal_q         False
wheattotal_q        False
moong_q             False
Milktotal_q         False
chicken_q           False
bread_q             False
foodtotal_q         False
Beveragestotal_v    False
Meals_At_Home       False
dtype: bool
```

```
# Outlier Checking
```

```
import matplotlib.pyplot as plt
# Assuming AP_clean is your DataFrame
plt.figure(figsize=(8, 6))
plt.boxplot(MP_clean['ricetotal_q'])
plt.xlabel('ricetotal_q')
plt.ylabel('Values')
plt.title('Boxplot of ricetotal_q')
plt.show()
```

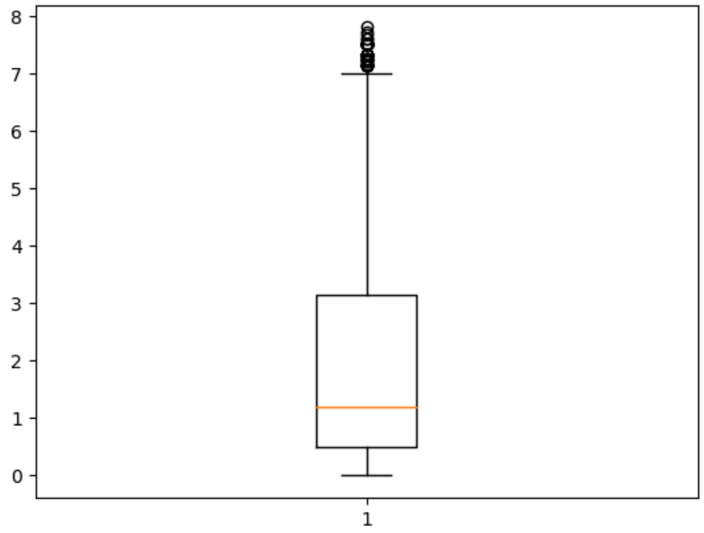


```
rice1 = MP_clean['ricetotal_q'].quantile(0.25)
rice2 = MP_clean['ricetotal_q'].quantile(0.75)
iqr_rice = rice2-rice1
up_limit = rice2 + 1.5*iqr_rice
low_limit = rice1 - 1.5*iqr_rice
```

```
MP_clean=MP_new[(MP_new['ricetotal_q']<=up_limit)&(MP_new['ricetotal_q']>=low_limit)]
```

```
plt.boxplot(MP_clean['ricetotal_q'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7807e0a4a950>,
  <matplotlib.lines.Line2D at 0x7807e16cd7e0>],
 'caps': [<matplotlib.lines.Line2D at 0x7807e16ced40>,
  <matplotlib.lines.Line2D at 0x7807e16cc370>],
 'boxes': [<matplotlib.lines.Line2D at 0x7807d49d90c0>],
 'medians': [<matplotlib.lines.Line2D at 0x7807e0837790>],
 'fliers': [<matplotlib.lines.Line2D at 0x7807e0eec2e0>],
 'means': []}
```



```
MP_clean['District'].unique()
```

```
array([49, 40, 41, 47, 39, 44, 42, 10, 38, 17, 13, 16, 12, 15, 14, 11, 50,
        8,  4,  7,  3,  6,  5,  2,  9,  1, 23, 24, 22, 46, 30, 21, 18, 19,
       20, 26, 32, 48, 43, 34, 35, 33, 37, 36, 45, 31, 29, 28, 27, 25])
```

```
# Replace values in the 'Sector' column
MP_clean.loc[:,'Sector'] = MP_clean['Sector'].replace([1, 2], ['URBAN', 'RURAL'])
```

```
#total consumption
```

```
MP_clean.columns
```

```
Index(['state_1', 'District', 'Sector', 'Region', 'State_Region',
       'ricetotal_q', 'wheattotal_q', 'moong_q', 'Milktotal_q', 'chicken_q',
       'bread_q', 'foodtotal_q', 'Beveragestotal_v', 'Meals_At_Home'],
      dtype='object')
```

```
MP_clean.loc[:, 'total_consumption'] = MP_clean[['ricetotal_q', 'wheattotal_q', 'moong_q', 'Milktotal_q', 'chicken_q', 'bread_q', 'food
```

```
<ipython-input-66-80d49cf8525b>:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
```

```
MP_clean.head()
```

	state_1	District	Sector	Region	State_Region	ricetotal_q	wheattotal_q	mo
2091	MP	49	RURAL	1	231	1.000000	8.000000	0.1
2092	MP	49	RURAL	1	231	1.000000	10.000000	0.0
2093	MP	49	RURAL	1	231	0.500000	8.125000	0.0
2094	MP	49	RURAL	1	231	1.428571	9.285714	0.1
2095	MP	49	RURAL	1	231	0.500000	10.000000	0.0

Next steps: Generate code with MP_clean View recommended plots

```
MP_clean.groupby('Region').agg({'total_consumption':['std','mean','max','min']})
```

	total_consumption			
	std	mean	max	min
Region				
1	29.560237	46.640378	532.920880	0.0
2	38.829794	54.058974	551.007358	0.0
3	36.706541	55.662767	368.150555	0.0
4	23.742635	43.056724	228.551375	0.0
5	13.588121	38.945954	140.886029	0.0
6	31.381611	50.936510	361.951650	0.0

```
MP_clean.groupby('District').agg({'total_consumption':['std','mean','max','min']})
```

	total_consumption			
	std	mean	max	min
District				
1	16.775819	39.746085	102.950500	17.940200
2	19.073353	44.085860	122.815275	18.262662
3	16.968680	42.753640	119.150300	21.950169
4	37.374055	50.190857	235.688827	17.257750
5	22.757348	47.536850	138.277767	20.528800
6	45.287017	65.215649	361.951650	0.000000
7	38.497496	58.312801	299.831165	23.387795
8	21.852111	54.327756	143.701710	18.300000
9	53.524958	64.259850	532.920880	18.728716
10	23.481478	49.247331	138.950163	14.283386
11	36.857306	54.414838	340.839137	12.608433
12	36.850612	57.929265	296.460550	18.935871
13	19.197056	43.727410	153.575265	21.718437
14	39.902524	49.481078	423.875520	0.000000
15	16.523679	41.945661	98.400473	26.370070
16	12.689767	38.899458	79.925275	21.420040
17	16.718969	37.721699	103.307825	17.875053
18	37.465217	52.226480	210.733977	20.360160
19	28.003673	51.176768	159.725825	18.783460
20	21.951832	42.260451	139.851975	14.666850
21	39.690816	64.011533	236.045783	6.200000
22	36.275602	48.298919	368.150555	0.000000
23	33.855594	53.693723	238.000275	13.900585
24	18.701657	39.959696	126.017220	18.310280
25	40.250016	64.869301	214.600400	0.000000
26	47.764086	69.874306	282.860450	16.014621
27	10.166227	40.517103	69.760136	16.107271
28	14.084628	38.593005	131.900800	18.200549
29	10.940196	37.999688	71.139261	0.000000
30	31.594859	55.898809	193.375348	6.880000
31	58.686453	58.446243	551.007358	15.791918
32	36.957188	54.191858	335.480256	0.000000
33	27.034604	43.908925	170.626250	0.000000
34	30.520705	55.282438	175.801020	18.700380
35	9.738820	34.849389	79.060165	0.000000
36	14.390718	38.462336	71.467048	0.000000
37	20.166315	41.184527	140.886029	10.000000
38	26.203667	46.447708	139.176310	7.448220
39	24.994098	45.469218	175.433373	8.620000
40	25.979579	53.048309	199.640492	15.700110
41	12.958814	33.770509	79.500533	12.228715
42	31.237234	39.575799	228.551375	15.000000
43	20.502409	40.108454	212.100000	19.812693
44	20.626990	42.666259	101.275600	0.000000
45	13.173145	39.160220	91.050400	23.505155
46	27.769475	60.623462	207.025908	31.225184
47	14.939470	36.971601	127.575130	24.450108

48	15.200552	42.819423	108.350470	25.417386
49	17.099732	37.697130	89.450175	10.000000
50	13.531332	40.836403	84.908130	19.666743

```
total_consumption_by_districtcode=MP_clean.groupby('District')['total_consumption'].sum()
```

```
total_consumption_by_districtcode.sort_values(ascending=False).head(3)
```

```
District
26    11179.888996
11     8651.959181
32     8562.313518
Name: total_consumption, dtype: float64
```

```
total_consumption_by_districtcode.sort_values(ascending=False).tail(3)
```

```
District
50    2041.820134
45    1958.010997
47    1700.693654
Name: total_consumption, dtype: float64
```

```
MP_clean.loc[:,"District"] = MP_clean.loc[:,"District"].replace({11: "Sagar", 26: "Indore", 32: "Bhopal"})
```

```
MP_clean.loc[:,"District"] = MP_clean.loc[:,"District"].replace({15: "Umaria", 45: "Balaghat", 50: "Singrauli"})
```

```
total_consumption_by_districtname=MP_clean.groupby('District')['total_consumption'].sum()
```

```
total_consumption_by_districtname.sort_values(ascending=False).head(3)
```

```
District
Indore    11179.888996
Sagar      8651.959181
Bhopal     8562.313518
Name: total_consumption, dtype: float64
```

```
total_consumption_by_districtname.sort_values(ascending=False).tail(3)
```

```
District
Singrauli    2041.820134
Balaghat     1958.010997
Anuppur      1700.693654
```