Load and read the data

```
# Import required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import shapefile
```

```
# Read the shapefiles
def read_shp(file_path):
    sf = shapefile.Reader(file_path)
    fields = [f[0] for f in sf.fields[1:]]
    records = sf.records()
    df = pd.DataFrame(records, columns=fields)
    return df

district_shp = read_shp("SHAPEFILES/Uganda_Districts.shp")
subcounty_shp = read_shp("SHAPEFILES/Uganda_Subcounties.shp")
maize_shp = read_shp("SHAPEFILES/Crop_Type_Map_Maize.shp")
sorghum_shp = read_shp("SHAPEFILES/Crop_Type_Map_Sorghum.shp")
```

```
# Read the two tables
district_df = pd.read_csv("TABLES/Uganda_Karamoja_District_Crop_Yield_Population.csv")
subcounty_df = pd.read_csv("TABLES/Uganda_Karamoja_Subcounty_Crop_Yield_Population.csv")
```

Data Understanding

```
#Check sample data for districts
district_df.head()
```

	OBJECTID	NAME	POP	Area	S_Yield_Ha	M_Yield_Ha	Crop_Area_Ha	S_Area_Ha	M_Area_Ha	S_Prod_Tot	M_Prod_To
0	92	ABIM	90385	2771977106	449	1040	5470.068394	3277.295971	1848.621855	1471506	19225
1	96	AMUDAT	101790	1643582836	205	1297	5765.443719	2973.423860	2733.661014	609552	35455
2	20	KAABONG	627057	7373606003	279	945	28121.672530	20544.194960	7394.416334	5731830	69877
3	85	KOTIDO	243157	3641539808	331	1148	53032.649450	50247.443900	1751.372284	16631904	20105
4	5	MOROTO	127811	3570160948	128	355	5954.814048	4741.748776	1190.050606	606944	4224

```
#Check sample data for subcounty
subcounty_df.head()
```

	OBJECTID	SUBCOUNTY_NAME	DISTRICT_NAME	POP	Area	Karamoja	S_Yield_Ha	M_Yield_Ha	Crop_Area_Ha	S_Area_Ha	M_
0	263	KACHERI	KOTIDO	17244	1067176155	Y	354.207411	1137.467019	7023.533691	6434.342449	528
1	264	KOTIDO	KOTIDO	52771	597575188	Y	367.890523	1162.996687	13587.990760	12455.592640	824
2	265	KOTIDO TOWN COUNCIL	KOTIDO	27389	23972401	Y	369.314177	1167.005832	1656.531855	1520.322052	8
3	266	NAKAPERIMORU	KOTIDO	38775	419111591	Y	283.324569	852.366578	7087.823334	6761.488901	45
4	267	PANYANGARA	KOTIDO	65704	880955930	Y	373.836926	1283.859882	10398.249390	10111.198130	172

Both columns have similar fields and most of them being numbers.

```
# Check info about the dfs
district_df.info()
# Dataset has 7 districts with 11 columns of crop and population and metadata.No null values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   OBJECTID      7 non-null      int64
```

```
 1   NAME          7 non-null      object
 2   POP           7 non-null      int64
 3   Area          7 non-null      int64
 4   S_Yield_Ha    7 non-null      int64
 5   M_Yield_Ha    7 non-null      int64
 6   Crop_Area_Ha  7 non-null      float64
 7   S_Area_Ha     7 non-null      float64
 8   M_Area_Ha     7 non-null      float64
 9   S_Prod_Tot    7 non-null      int64
 10  M_Prod_Tot    7 non-null      int64
dtypes: float64(3), int64(7), object(1)
memory usage: 744.0+ bytes
```

Data entails 7 districts with 11 columns. No null values.

```
# Check info about the dfs
subcounty_df.info()
# We have 52 rows of sub-counties with 13 columns of crop, population and meta data. No null values.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   OBJECTID        52 non-null     int64
 1   SUBCOUNTY_NAME  52 non-null     object
 2   DISTRICT_NAME   52 non-null     object
 3   POP             52 non-null     int64
 4   Area            52 non-null     int64
 5   Karamoja        52 non-null     object
 6   S_Yield_Ha      52 non-null     float64
 7   M_Yield_Ha      52 non-null     float64
 8   Crop_Area_Ha    52 non-null     float64
 9   S_Area_Ha       52 non-null     float64
 10  M_Area_Ha       52 non-null     float64
 11  S_Prod_Tot      52 non-null     float64
 12  M_Prod_Tot      52 non-null     float64
dtypes: float64(7), int64(3), object(3)
memory usage: 5.4+ KB
```

```
# Read maize shapefile
maize_shp.head()
```

	DN
0	6
1	6
2	6
3	6
4	6

```
# Read sorghum shapefile
sorghum_shp.head()
```

	DN
0	212
1	212
2	212
3	212
4	212

```
sorghum_shp["DN"].nunique()

```

```
1
```

.shp files have 2 columns and the DN is the same throughout the dataframe

Feature Engineering

These new features to help assess the level of food security per subcounty. The per capita productivity of the 2 crops and the land-yield ratio

will inform on land use efficiency hence insight on how much is available to the population relatively.

```
# Create subcounty productivity per capita to gauge how much is available to the population on average
subcounty_df['S_Prod_Per_Capita'] = subcounty_df['S_Prod_Tot']/subcounty_df['POP']
subcounty_df['M_Prod_Per_Capita'] = subcounty_df['M_Prod_Tot']/subcounty_df['POP']
subcounty_df.head()
```

	OBJECTID	SUBCOUNTY_NAME	DISTRICT_NAME	POP	Area	Karamoja	S_Yield_Ha	M_Yield_Ha	Crop_Area_Ha	S_Area_Ha	M_
0	263	KACHERI	KOTIDO	17244	1067176155	Y	354.207411	1137.467019	7023.533691	6434.342449	528
1	264	KOTIDO	KOTIDO	52771	597575188	Y	367.890523	1162.996687	13587.990760	12455.592640	824
2	265	KOTIDO TOWN COUNCIL	KOTIDO	27389	23972401	Y	369.314177	1167.005832	1656.531855	1520.322052	8
3	266	NAKAPERIMORU	KOTIDO	38775	419111591	Y	283.324569	852.366578	7087.823334	6761.488901	45
4	267	PANYANGARA	KOTIDO	65704	880955930	Y	373.836926	1283.859882	10398.249390	10111.198130	172

```
#Create subcounty land-yield ratio to investigate land use efficiency
# Land use efficiency = (Crop Area / Total Area) * 100
subcounty_df['Land_Use_Efficiency'] = (subcounty_df['Crop_Area_Ha'] / subcounty_df['Area']) * 100
subcounty_df.head()
```

	OBJECTID	SUBCOUNTY_NAME	DISTRICT_NAME	POP	Area	Karamoja	S_Yield_Ha	M_Yield_Ha	Crop_Area_Ha	S_Area_Ha	M_
0	263	KACHERI	KOTIDO	17244	1067176155	Y	354.207411	1137.467019	7023.533691	6434.342449	528
1	264	KOTIDO	KOTIDO	52771	597575188	Y	367.890523	1162.996687	13587.990760	12455.592640	824
2	265	KOTIDO TOWN COUNCIL	KOTIDO	27389	23972401	Y	369.314177	1167.005832	1656.531855	1520.322052	8
3	266	NAKAPERIMORU	KOTIDO	38775	419111591	Y	283.324569	852.366578	7087.823334	6761.488901	45
4	267	PANYANGARA	KOTIDO	65704	880955930	Y	373.836926	1283.859882	10398.249390	10111.198130	172

Data Manipulation

We want to retain only what is necessary from our data set for visualization purposes.

```
# Add a prefix D to district_df columns to differentiate them from subcounty_df columns when doing the merge.
district_df.columns = ['D' + col for col in district_df.columns]
district_df.info()

```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   DOBJECTID      7 non-null      int64
 1   DNAME          7 non-null      object
 2   DPOP           7 non-null      int64
 3   DArea          7 non-null      int64
 4   DS_Yield_Ha    7 non-null      int64
 5   DM_Yield_Ha    7 non-null      int64
 6   DCrop_Area_Ha  7 non-null      float64
 7   DS_Area_Ha     7 non-null      float64
 8   DM_Area_Ha     7 non-null      float64
 9   DS_Prod_Tot    7 non-null      int64
 10  DM_Prod_Tot    7 non-null      int64
dtypes: float64(3), int64(7), object(1)
memory usage: 744.0+ bytes
```

```
# Create District Yield,District Crop Area, District Prod by summing the respective crop columns; to see the crop output at
district_df['District Yield'] = district_df['DS_Yield_Ha'] + district_df['DM_Yield_Ha']
district_df['District Crop Area'] = district_df['DS_Area_Ha'] + district_df['DM_Area_Ha']
district_df['District Prod'] = district_df['DS_Prod_Tot'] + district_df['DM_Prod_Tot']

district_df.head()
```

	DOBJECTID	DNAME	DPOP	DArea	DS_Yield_Ha	DM_Yield_Ha	DCrop_Area_Ha	DS_Area_Ha	DM_Area_Ha	DS_Prod_Tot	DM_I
0	92	ABIM	90385	2771977106	449	1040	5470.068394	3277.295971	1848.621855	1471506	
1	96	AMUDAT	101790	1643582836	205	1297	5765.443719	2973.423860	2733.661014	609552	
2	20	KAABONG	627057	7373606003	279	945	28121.672530	20544.194960	7394.416334	5731830	
3	85	KOTIDO	243157	3641539808	331	1148	53032.649450	50247.443900	1751.372284	16631904	
4	5	MOROTO	127811	3570160948	128	355	5954.814048	4741.748776	1190.050606	606944	

```
# Drop the unnecessary columns e.g individual crop columns from which we just summed above as they will be redundant once w
columns_to_drop = ['DS_Yield_Ha', 'DM_Yield_Ha', 'DS_Area_Ha', 'DM_Area_Ha', 'DS_Prod_Tot', 'DM_Prod_Tot', 'DOBJECTID']

district_df = district_df.drop(columns=columns_to_drop)
district_df.head()

```

	DNAME	DPOP	DArea	DCrop_Area_Ha	District Yield	District Crop Area	District Prod
0	ABIM	90385	2771977106	5470.068394	1489	5125.917826	3394073
1	AMUDAT	101790	1643582836	5765.443719	1502	5707.084874	4155110
2	KAABONG	627057	7373606003	28121.672530	1224	27938.611294	12719553
3	KOTIDO	243157	3641539808	53032.649450	1479	51998.816184	18642479
4	MOROTO	127811	3570160948	5954.814048	483	5931.799382	1029412

```
# Merge subcounty_df to district_df to have a full enriched dataset
master_df = pd.merge(left = subcounty_df, right = district_df, how= "left", left_on='DISTRICT_NAME', right_on='DNAME')
```

```
#Drop unnecessary/duplicated columns
master_df.drop(columns=['DNAME', 'Karamoja'], inplace=True)
```

```
# Check master_df
master_df.head()
```

	OBJECTID	SUBCOUNTY_NAME	DISTRICT_NAME	POP	Area	S_Yield_Ha	M_Yield_Ha	Crop_Area_Ha	S_Area_Ha	M_Area_Ha	.
0	263	KACHERI	KOTIDO	17244	1067176155	354.207411	1137.467019	7023.533691	6434.342449	528.124229	
1	264	KOTIDO	KOTIDO	52771	597575188	367.890523	1162.996687	13587.990760	12455.592640	824.767081	
2	265	KOTIDO TOWN COUNCIL	KOTIDO	27389	23972401	369.314177	1167.005832	1656.531855	1520.322052	8.561644	
3	266	NAKAPERIMORU	KOTIDO	38775	419111591	283.324569	852.366578	7087.823334	6761.488901	45.721712	
4	267	PANYANGARA	KOTIDO	65704	880955930	373.836926	1283.859882	10398.249390	10111.198130	172.611914	

5 rows × 21 columns

```
master_df.describe()
```

	OBJECTID	POP	Area	S_Yield_Ha	M_Yield_Ha	Crop_Area_Ha	S_Area_Ha	M_Area_Ha	S_Prod_Tot	M
count	52.000000	52.000000	5.200000e+01	52.000000	52.000000	52.000000	52.000000	52.000000	5.200000e+01	5.2
mean	787.865385	28934.692308	5.331913e+08	274.165405	940.259552	2839.646974	2253.143395	536.300569	6.557443e+05	5.5
std	280.101314	20865.122974	4.913308e+08	118.569907	321.641901	3110.505917	2954.355858	724.092288	9.915839e+05	7.9
min	263.000000	1418.000000	2.121209e+06	108.156411	0.000000	0.171390	0.130941	0.000000	1.728126e+01	0.0
25%	597.750000	16558.500000	1.568923e+08	173.034066	743.075879	964.876031	405.394759	79.821743	1.210555e+05	6.0
50%	810.500000	23053.500000	3.848356e+08	277.255206	1016.684002	1654.265138	1231.824455	326.479336	2.543687e+05	2.8
75%	982.250000	39461.000000	7.749029e+08	368.246437	1203.548665	3267.564651	2429.985069	740.296675	6.040942e+05	8.1
max	1320.000000	100919.000000	2.069555e+09	560.313070	1396.991494	13587.990760	12964.499730	3840.698081	4.582294e+06	4.3

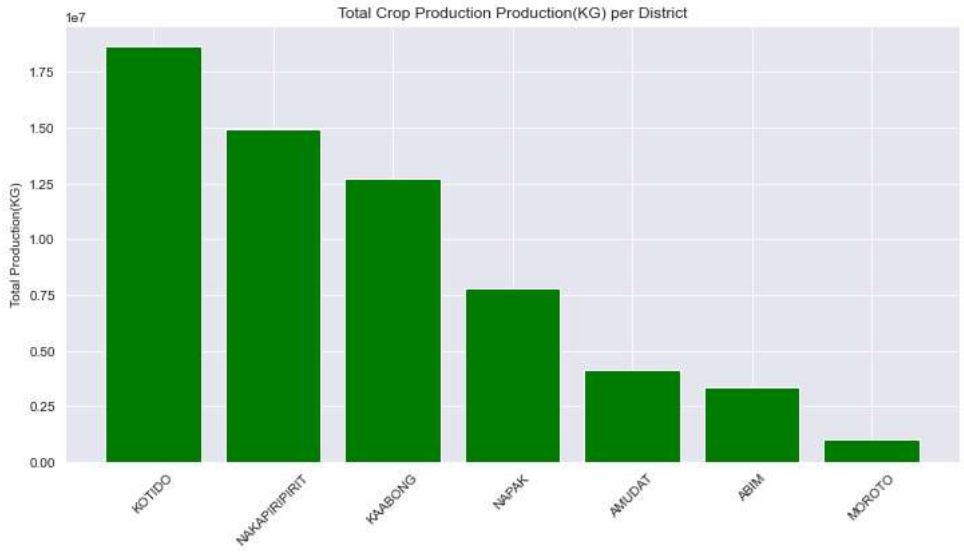
Summary statistics for 52 subcounties showing wide variation in population (1,418 to 100,919) and agricultural production. Mean sorghum yield is 274 kg/ha versus 940 kg/ha for maize, with average total crop area of 2,840 hectares per sub-county. District-level aggregates indicate mean production of 9.9 million tons across an average crop area of 22,675 hectares.

```
master_df.info()
# Since our left table was subcounties, we have 52 rows of data with the enjoined district data.
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 52 entries, 0 to 51
Data columns (total 21 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   OBJECTID             52 non-null     int64  
 1   SUBCOUNTY_NAME       52 non-null     object 
 2   DISTRICT_NAME        52 non-null     object 
 3   POP                  52 non-null     int64  
 4   Area                 52 non-null     int64  
 5   S_Yield_Ha           52 non-null     float64
 6   M_Yield_Ha           52 non-null     float64
 7   Crop_Area_Ha         52 non-null     float64
 8   S_Area_Ha            52 non-null     float64
 9   M_Area_Ha            52 non-null     float64
 10  S_Prod_Tot           52 non-null     float64
 11  M_Prod_Tot           52 non-null     float64
 12  S_Prod_Per_Capita    52 non-null     float64
 13  M_Prod_Per_Capita    52 non-null     float64
 14  Land_Use_Efficiency  52 non-null     float64
 15  DPOP                 52 non-null     int64  
 16  DArea                52 non-null     int64  
 17  DCrop_Area_Ha        52 non-null     float64
 18  District Yield       52 non-null     int64  
 19  District Crop Area   52 non-null     float64
 20  District Prod        52 non-null     int64  
dtypes: float64(12), int64(7), object(2)
memory usage: 8.9+ KB
```

Exploratory Data Analysis

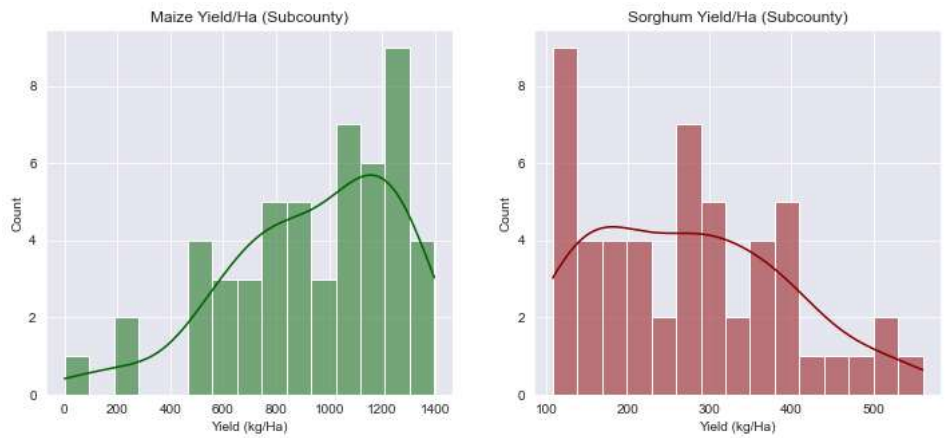
```
#Sort the district productivity from highest to lowest to see the output across.
district_df_sorted = master_df.sort_values(by="District Prod", ascending=False)

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


plt.bar(district_df_sorted["DISTRICT_NAME"], district_df_sorted["District Prod"], label="Maize Productivity",  color="green

#Details
plt.xticks(rotation=45)
plt.ylabel("Total Production(KG)")
plt.title("Total Crop Production Production(KG) per District")
# plt.legend()
plt.show()
```



Observation:

Kotido has the highest crop production at 18642479 Kgs while Moroto is perfoming dismally.

```
#Plot a histogram for each crop to investigate the distribution
fig, axes = plt.subplots(1,2, figsize=(12,5))

sns.set_style("darkgrid")
sns.histplot(master_df["M_Yield_Ha"], bins=15, kde=True, ax=axes[0], color="darkgreen")
axes[0].set_title("Maize Yield/Ha (Subcounty)")
axes[0].set_xlabel("Yield (kg/Ha)")

sns.histplot(master_df["S_Yield_Ha"], bins=15, kde=True, ax=axes[1], color="darkred")
axes[1].set_title("Sorghum Yield/Ha (Subcounty)")
axes[1].set_xlabel("Yield (kg/Ha)")

plt.show()
```



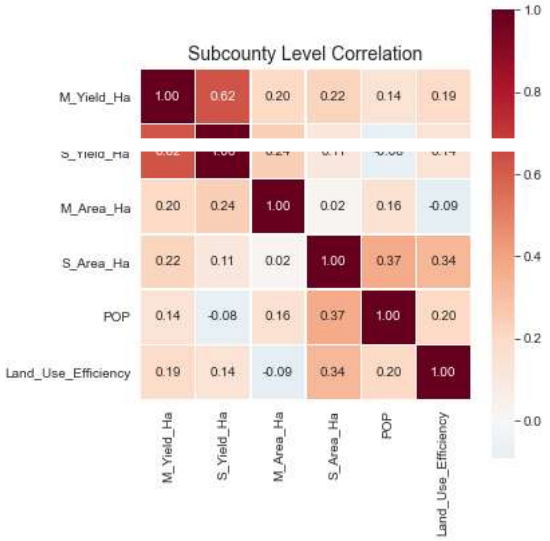
Observation:

Maize yield has a positive distribution with more subcounties producing between 1000-1200Kgs while sorghum has a negative distribution with more subcounties producing between 100-300Kgs.

This implies maize production is well estabished in most areas.

```
# Perform a correlation analysis to analyze the variables' relationship

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

# Include efficiency and per capita metrics in correlation
subcounty_corr = master_df[["M_Yield_Ha", "S_Yield_Ha", "M_Area_Ha", "S_Area_Ha", "POP",
```

```
                          "Land_Use_Efficiency"]].corr()

sns.heatmap(subcounty_corr,
            annot=True,
            cmap="RdBu_r",
            center=0,
            fmt='.2f',
            square=True,
            linewidths=0.5)

#Details
plt.title("Subcounty Level Correlation" , fontsize=14)
plt.tight_layout()
plt.show()
```



Observations

- Maize and sorghum yields have a relatively strong correlation, implying both cereals tend to perform well in the given conditions.
- Population has a weak postive correlation with land use. Hence, a higher population does not imply better land use.
- Maize and sorghum areas have a negligible relationship; there is no relation on the area on which they are cultivated.

Export dataframes as csv

```
# Download the district data
district_df.to_csv('TABLES/karamoja_district.csv', index=False)

# Download the subcounty data
subcounty_df.to_csv('TABLES/karamoja_subcounty.csv', index=False)

# Download master_df
master_df.to_csv('TABLES/karamoja_master.csv', index=False)


```