In [ ]:

```
!pip install openpyxl
!pip install xlrd
```

```
Requirement already satisfied: openpyxl in c:\users\teste\appdata\local\programs\pyt
hon\python311\lib\site-packages (3.1.2)
Requirement already satisfied: et-xmlfile in c:\users\teste\appdata\local\programs\p
ython\python311\lib\site-packages (from openpyxl) (1.1.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: xlrd in c:\users\teste\appdata\local\programs\python
\python311\lib\site-packages (2.0.1)

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [ ]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

pd.read_excel('../files/titanic3.xls')
excel_file_path = '../files/food-twentieth-century-crop-statistics-1900-2017-xlsx.x
sheet_name = 'Maize_Hectares'
header_row = 1
df = pd.read_excel(excel_file_path, sheet_name=sheet_name, header=header_row)
df = df.rename(columns={'admin0':'Country','admin1':'Location'})
df
```

Out[ ]:

	Country	Location	crop	notes	1861	1866	1867	1868	18
0	Argentina	NaN	maize	NaN	NaN	NaN	NaN	NaN	Na
1	Australia	Australian Capital Territory	maize	NaN	NaN	NaN	NaN	NaN	Na
2	Australia	New South Wales(b)	maize	NaN	20800.0	45900.0	NaN	NaN	Na
3	Australia	Northern Territory	maize	NaN	NaN	NaN	NaN	NaN	Na
4	Australia	Queensland	maize	NaN	800.0	4000.0	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	
115	United States	WASHINGTON	maize	NaN	NaN	NaN	NaN	NaN	Na
116	United States	WEST VIRGINIA	maize	NaN	NaN	121404.0	133544.4	141638.0	152969.
117	United States	WISCONSIN	maize	NaN	NaN	202340.0	206386.8	228644.2	232691.
118	United States	WYOMING	maize	NaN	NaN	NaN	NaN	NaN	Na
119	Uruguay	NaN	maize	NaN	NaN	NaN	NaN	NaN	Na

120 rows × 159 columns

The columns from "crop" to "1899" are not in my field of focus so I drop them

In [ ]:

```
start_column = 'crop'
end_column = 1899

# Create a list of column names to drop
columns_to_drop = df.columns[df.columns.get_loc(start_column): df.columns.get_loc(e

# Drop columns
df = df.drop(columns=columns_to_drop)
df
```

Out[ ]:

	Country	Location	1900	1901	1902	1903	1904
0	Argentina	NaN	NaN	1255346.00	1405796.00	1801644.00	2106819.00
1	Australia	Australian Capital Territory	NaN	NaN	NaN	NaN	NaN
2	Australia	New South Wales(b)	86900.00	83400.00	67700.00	81900.00	91800.00
3	Australia	Northern Territory	NaN	NaN	NaN	NaN	NaN
4	Australia	Queensland	51800.00	47300.00	36400.00	53900.00	48200.00
...	...	...	...	...	...	...	...
115	United States	WASHINGTON	6070.20	6474.88	6879.56	7284.24	7688.92
116	United States	WEST VIRGINIA	297439.80	291369.60	295416.40	279229.20	279229.20
117	United States	WISCONSIN	637371.00	659628.40	679862.40	667722.00	659628.40
118	United States	WYOMING	1618.72	2023.40	2428.08	2428.08	2428.08
119	Uruguay	NaN	NaN	145668.00	181558.00	178238.00	162467.00

120 rows × 122 columns

Removing rows with 75% or more null values

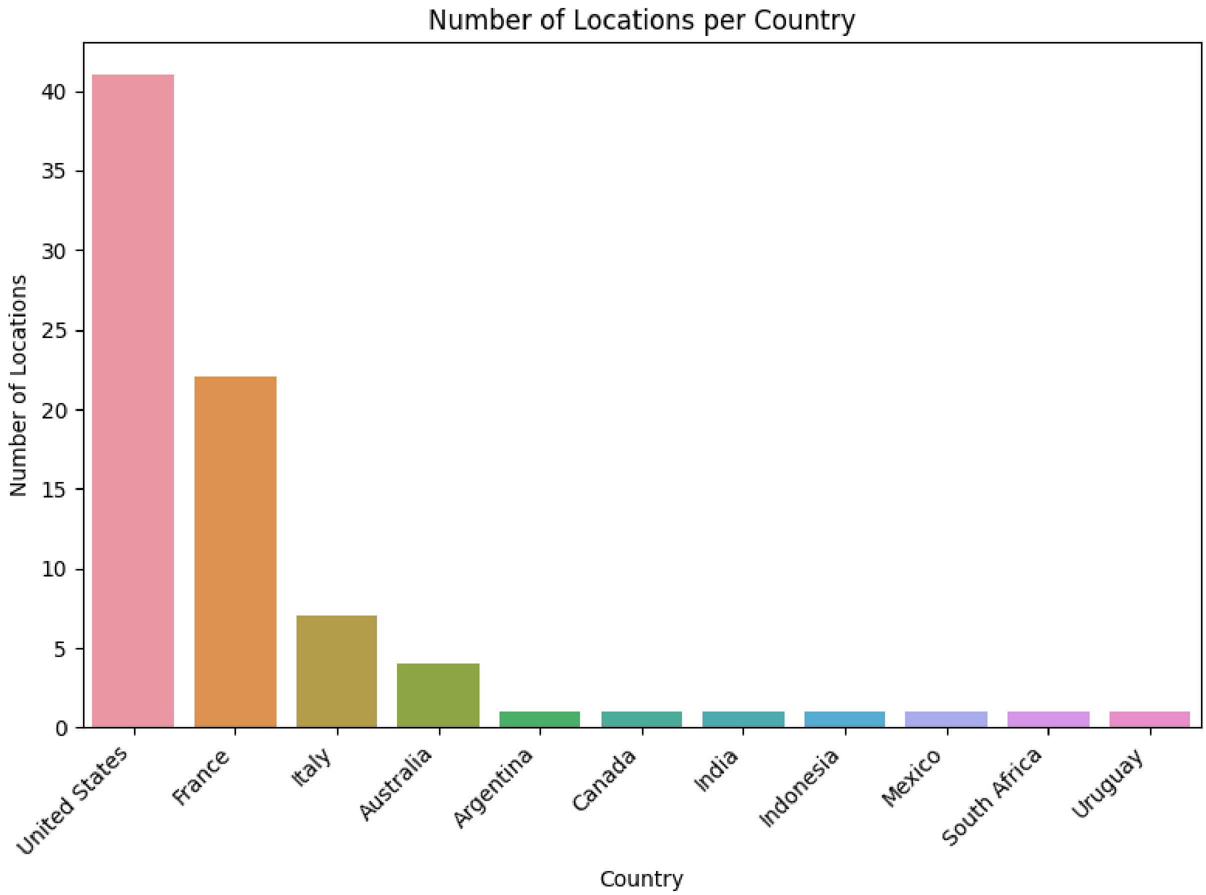
In [ ]:

```
threshold = int(0.75 * len(df.columns))

df = df.dropna(thresh=threshold)
```

In the barchart below we see which countries has the most locations for their maize from the most to least

In [ ]:

```
country_location = df[['Country', 'Location']]


# Count plot showing the number of locations in each country
plt.figure(figsize=(8, 6))
sns.countplot(data=country_location, x='Country', order=country_location['Country']
plt.xlabel('Country')
plt.ylabel('Number of Locations')
plt.title('Number of Locations per Country')
plt.xticks(rotation=45, ha='right')  # Rotate labels for better readability
plt.tight_layout()
plt.show()
```



In [ ]:

```
# Grouping by 'Country' and counting unique locations
locations_per_country = df.groupby('Country')['Location'].nunique()
print("Number of locations per country:")
print(locations_per_country)
```

```
Number of locations per country:
Country
Argentina         0
Australia         4
Canada            0
France           22
India             0
Indonesia         1
Italy             7
Mexico            0
South Africa      0
United States    41
Uruguay           0
Name: Location, dtype: int64
```

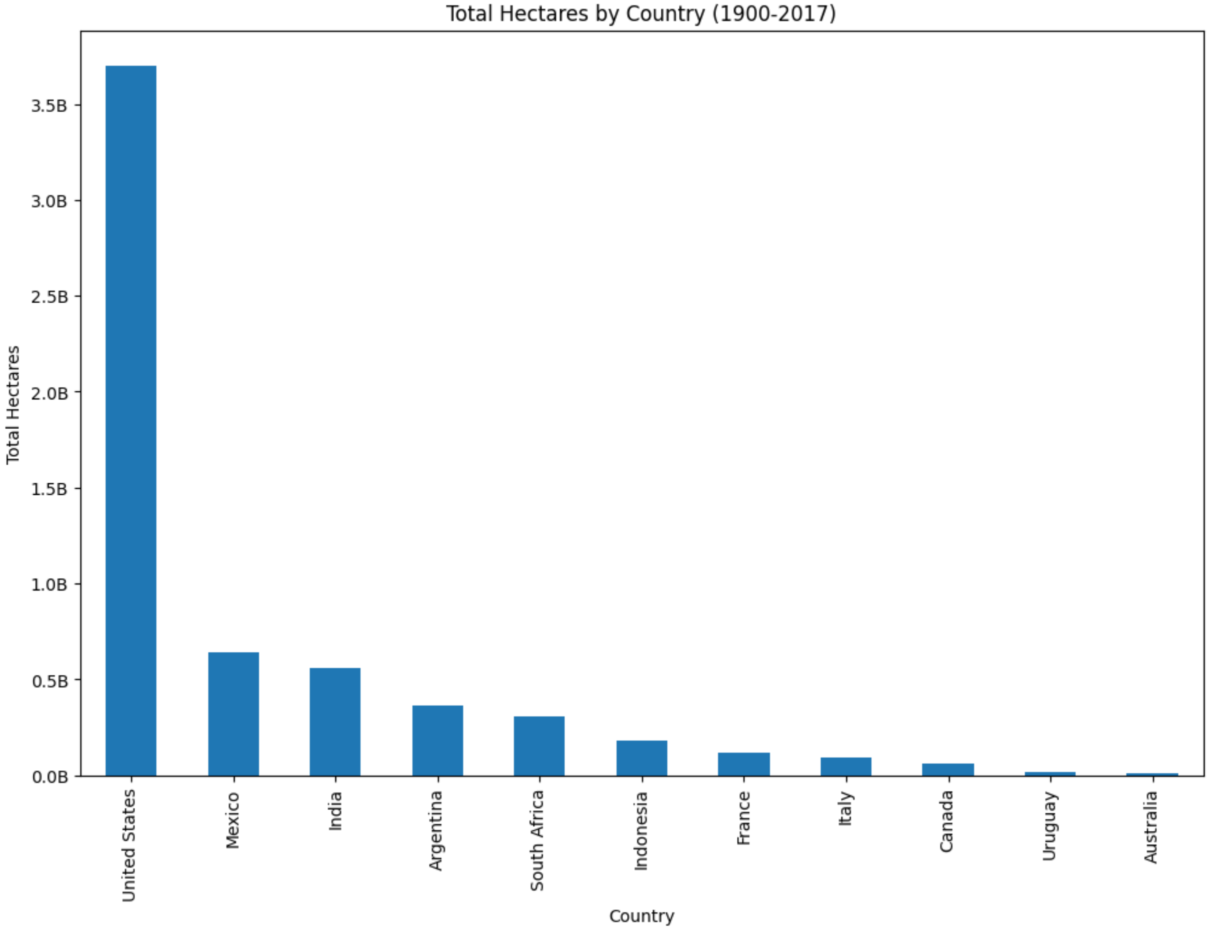
From 1900 to 2017 I will sum the total number of Hectares and compare against each country

In [ ]:

```
import matplotlib.ticker as ticker
```

```
df['Total_Hectares'] = df.loc[:, 1900:2017].sum(axis=1)

# Grouping by 'Country' and finding the total hectares for each country
country_totals = df.groupby('Country')['Total_Hectares'].sum()

country_totals = country_totals.sort_values(ascending=False)


# Plotting the total hectares for each country
country_totals.plot(kind='bar', figsize=(12, 8))
plt.title('Total Hectares by Country (1900-2017)')
plt.xlabel('Country')
plt.ylabel('Total Hectares')
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: '{:.1f}B'.for

plt.show()
```

```
C:\Users\teste\AppData\Local\Temp\ipykernel_21044\2824303973.py:3: SettingWithCopyWa
rning:
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/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  df['Total_Hectares'] = df.loc[:, 1900:2017].sum(axis=1)
```



We can see that the United States has the highest number of hectares for maize for the accumulated from 1900 to 2017 so I will be focusing on the United States fro this dataset

In [ ]:

```
import pandas as pd


us_data = df[df['Country'] == 'United States']

us_data
```

Out[ ]:

	Country	Location	1900	1901	1902	1903	1904
70	United States	ALABAMA	1064308.40	1044074.40	1084542.40	1088589.20	1052168.00
71	United States	ARIZONA	4451.48	4856.16	4046.80	4856.16	4046.80
72	United States	ARKANSAS	922670.40	886249.20	906483.20	870062.00	837687.60
73	United States	CALIFORNIA	25090.16	25494.84	25899.52	25090.16	23876.12
74	United States	COLORADO	38039.92	44110.12	53013.08	57869.24	66367.52
76	United States	DELAWARE	78912.60	78103.24	77698.56	80126.64	78912.60
77	United States	FLORIDA	232691.00	230667.60	238761.20	240784.60	240784.60
78	United States	GEORGIA	1444707.60	1388052.40	1388052.40	1363771.60	1339490.80
79	United States	IDAHO	4046.80	4046.80	3642.12	4046.80	3237.44
80	United States	ILLINOIS	4232952.80	4330076.00	4390778.00	4269374.00	4249140.00
81	United States	INDIANA	2033517.00	2043634.00	2114453.00	2084102.00	2205506.00
82	United States	IOWA	3682588.00	3828272.80	3868740.80	3403358.80	3864694.00
83	United States	KANSAS	3023769.00	2766392.50	2889415.20	2714188.80	2714188.80
84	United States	KENTUCKY	1363771.60	1375912.00	1416380.00	1386029.00	1497316.00
85	United States	LOUISIANA	526084.00	509896.80	505850.00	489662.80	473475.60
87	United States	MARYLAND	270326.24	271135.60	273159.00	267088.80	269112.20
89	United States	MICHIGAN	635347.60	647488.00	655581.60	655581.60	647488.00
90	United States	MINNESOTA	598926.40	651534.80	728424.00	724377.20	789126.00
91	United States	MISSISSIPPI	849828.00	857921.60	870062.00	882202.40	829594.00

	Country	Location	1900	1901	1902	1903	1904
92	United States	MISSOURI	3116036.00	3055334.00	3055334.00	2852994.00	2691122.00
93	United States	MONTANA	1214.04	1618.72	2023.40	2023.40	2428.08
94	United States	NEBRASKA	2974398.00	2903579.00	2974398.00	2863111.00	2944047.00
97	United States	NEW JERSEY	127878.88	120189.96	125046.12	114929.12	114929.12
98	United States	NEW MEXICO	18210.60	19829.32	21448.04	23876.12	22662.08
99	United States	NEW YORK	341954.60	319697.20	327790.80	307556.80	311603.60
100	United States	NORTH CAROLINA	1092636.00	1040027.60	1080495.60	1027887.20	1027887.20
101	United States	NORTH DAKOTA	35611.84	55845.84	50989.68	55036.48	60702.00
102	United States	OHIO	1618720.00	1568135.00	1618720.00	1578252.00	1608603.00
103	United States	OKLAHOMA	1064308.40	1116916.80	1278788.80	1250461.20	1456848.00
104	United States	OREGON	9712.32	10926.36	11331.04	11735.72	11735.72
106	United States	PENNSYLVANIA	635347.60	629277.40	635347.60	617137.00	598926.40
108	United States	SOUTH CAROLINA	704143.20	667722.00	692002.80	663675.20	639394.40
109	United States	SOUTH DAKOTA	517990.40	586786.00	671768.80	687956.00	708190.00
110	United States	TENNESSEE	1311163.20	1343537.60	1428520.40	1392099.20	1384005.60
111	United States	TEXAS	1954604.40	1938417.20	1974838.40	1974838.40	1954604.40
112	United States	UTAH	4451.48	4451.48	4451.48	4451.48	4046.80
114	United States	VIRGINIA	768892.00	768892.00	776985.60	748658.00	748658.00
115	United States	WASHINGTON	6070.20	6474.88	6879.56	7284.24	7688.92

	Country	Location	1900	1901	1902	1903	1904
116	United States	WEST VIRGINIA	297439.80	291369.60	295416.40	279229.20	279229.20
117	United States	WISCONSIN	637371.00	659628.40	679862.40	667722.00	659628.40
118	United States	WYOMING	1618.72	2023.40	2428.08	2428.08	2428.08

41 rows × 123 columns

I will take 2018 and 2019 out of the data set, but the column 2018 will be saved to compare the model's prediction for 2018 with the actual data in 2018 but first I will remove rows that has all null values since it will be useless in this dataset

In [ ]:

```
null_count = us_data[2018].isnull().sum()
print("Null values count in '2018':", null_count)

non_null_count = us_data[2018].notnull().sum()
print("Non-null values count in '2018':", non_null_count)

null_count = us_data[2019].isnull().sum()
print("Null values count in '2019':", null_count)

non_null_count = us_data[2019].notnull().sum()
print("Non-null values count in '2019':", non_null_count)
```

```
Null values count in '2018': 9
Non-null values count in '2018': 32
Null values count in '2019': 41
Non-null values count in '2019': 0
```

In [ ]:

```
us_data.dropna(how='all', inplace=True)

crop2018_df = us_data.iloc[:, df.columns.get_loc('Location'): df.columns.get_loc(20
crop2019_df = us_data.iloc[:, df.columns.get_loc('Location'): df.columns.get_loc(20


crop2018_df.to_csv(f'{2018}_Cropdata.csv', index=False, header=True)
crop2019_df.to_csv(f'{2019}_Cropdata.csv', index=False, header=True)



start_column = 2018
end_column = 2019

# Create a list of column names to drop
columns_to_drop = us_data.columns[us_data.columns.get_loc(start_column): us_data.co

# Drop columns
```

```
us_data = us_data.drop(columns=columns_to_drop)
us_data
```

```
C:\Users\teste\AppData\Local\Temp\ipykernel_21044\3486512010.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  us_data.dropna(how='all', inplace=True)
```

Out[ ]:

	Country	Location	1900	1901	1902	1903	1904
70	United States	ALABAMA	1064308.40	1044074.40	1084542.40	1088589.20	1052168.00
71	United States	ARIZONA	4451.48	4856.16	4046.80	4856.16	4046.80
72	United States	ARKANSAS	922670.40	886249.20	906483.20	870062.00	837687.60
73	United States	CALIFORNIA	25090.16	25494.84	25899.52	25090.16	23876.12
74	United States	COLORADO	38039.92	44110.12	53013.08	57869.24	66367.52
76	United States	DELAWARE	78912.60	78103.24	77698.56	80126.64	78912.60
77	United States	FLORIDA	232691.00	230667.60	238761.20	240784.60	240784.60
78	United States	GEORGIA	1444707.60	1388052.40	1388052.40	1363771.60	1339490.80
79	United States	IDAHO	4046.80	4046.80	3642.12	4046.80	3237.44
80	United States	ILLINOIS	4232952.80	4330076.00	4390778.00	4269374.00	4249140.00
81	United States	INDIANA	2033517.00	2043634.00	2114453.00	2084102.00	2205506.00
82	United States	IOWA	3682588.00	3828272.80	3868740.80	3403358.80	3864694.00
83	United States	KANSAS	3023769.00	2766392.50	2889415.20	2714188.80	2714188.80
84	United States	KENTUCKY	1363771.60	1375912.00	1416380.00	1386029.00	1497316.00
85	United States	LOUISIANA	526084.00	509896.80	505850.00	489662.80	473475.60
87	United States	MARYLAND	270326.24	271135.60	273159.00	267088.80	269112.20
89	United States	MICHIGAN	635347.60	647488.00	655581.60	655581.60	647488.00
90	United States	MINNESOTA	598926.40	651534.80	728424.00	724377.20	789126.00
91	United States	MISSISSIPPI	849828.00	857921.60	870062.00	882202.40	829594.00

	Country	Location	1900	1901	1902	1903	1904
92	United States	MISSOURI	3116036.00	3055334.00	3055334.00	2852994.00	2691122.00
93	United States	MONTANA	1214.04	1618.72	2023.40	2023.40	2428.08
94	United States	NEBRASKA	2974398.00	2903579.00	2974398.00	2863111.00	2944047.00
97	United States	NEW JERSEY	127878.88	120189.96	125046.12	114929.12	114929.12
98	United States	NEW MEXICO	18210.60	19829.32	21448.04	23876.12	22662.08
99	United States	NEW YORK	341954.60	319697.20	327790.80	307556.80	311603.60
100	United States	NORTH CAROLINA	1092636.00	1040027.60	1080495.60	1027887.20	1027887.20
101	United States	NORTH DAKOTA	35611.84	55845.84	50989.68	55036.48	60702.00
102	United States	OHIO	1618720.00	1568135.00	1618720.00	1578252.00	1608603.00
103	United States	OKLAHOMA	1064308.40	1116916.80	1278788.80	1250461.20	1456848.00
104	United States	OREGON	9712.32	10926.36	11331.04	11735.72	11735.72
106	United States	PENNSYLVANIA	635347.60	629277.40	635347.60	617137.00	598926.40
108	United States	SOUTH CAROLINA	704143.20	667722.00	692002.80	663675.20	639394.40
109	United States	SOUTH DAKOTA	517990.40	586786.00	671768.80	687956.00	708190.00
110	United States	TENNESSEE	1311163.20	1343537.60	1428520.40	1392099.20	1384005.60
111	United States	TEXAS	1954604.40	1938417.20	1974838.40	1974838.40	1954604.40
112	United States	UTAH	4451.48	4451.48	4451.48	4451.48	4046.80
114	United States	VIRGINIA	768892.00	768892.00	776985.60	748658.00	748658.00
115	United States	WASHINGTON	6070.20	6474.88	6879.56	7284.24	7688.92

	Country	Location	1900	1901	1902	1903	1904
116	United States	WEST VIRGINIA	297439.80	291369.60	295416.40	279229.20	279229.20
117	United States	WISCONSIN	637371.00	659628.40	679862.40	667722.00	659628.40
118	United States	WYOMING	1618.72	2023.40	2428.08	2428.08	2428.08

41 rows × 121 columns

```
In [ ]: #Saving the data frame as a csv
        us_data.to_csv('cropStats.csv',index=False)
```