```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import plotly.express as px
```

```
df = pd.read_csv('/content/china_dept_trap - china_dept_trap.csv')
```

```
df.head()
```

	Expand All | Collapse All	YEAR	AMOUNT	LENDER	BORROWER	SECTOR	SENSITIVE TERRITORY OVERLAP
0	“Lar Patriota" Infrastructure (Phase 1)	2011	$50M	CDB	Government	Transport	None Known
1	10-Year Oil Supply Plan (10Mt/Year)	2009	$7.0B	CDB	Petrobras	Extraction, pipelines	None Known

```
df.tail()
```

	Expand All | Collapse All	YEAR	AMOUNT	LENDER	BORROWER	SECTOR	SENSITIVE TERRITORY OVERLAP	Count
853	Zalingei-El Geneinah Road Construction	2009	$120M	ExImBank	Government	Transport	None Known	Sud
	Zanzibar Airport						None	

```
df.shape
```

```
(858, 8)
```

```
df.columns
```

```
Index(['Expand All | Collapse All', 'YEAR', 'AMOUNT', 'LENDER', 'BORROWER',
       'SECTOR', 'SENSITIVE TERRITORY OVERLAP', 'Country'],
      dtype='object')
```

```
df = df.rename(columns = {'Expand All | Collapse All':'Title'})
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 858 entries, 0 to 857
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Title                        858 non-null    object
 1   YEAR                         858 non-null    int64 
 2   AMOUNT                       858 non-null    object
 3   LENDER                       858 non-null    object
 4   BORROWER                     857 non-null    object
 5   SECTOR                       858 non-null    object
 6   SENSITIVE TERRITORY OVERLAP  858 non-null    object
 7   Country                      858 non-null    object
dtypes: int64(1), object(7)
memory usage: 53.8+ KB
```

```
df.describe()
```

	YEAR
count	858.000000
mean	2013.268065
std	2.894155
min	2008.000000
25%	2011.000000
50%	2013.000000
75%	2016.000000
max	2019.000000

```
df.isnull().sum()
```

```
Title                          0
YEAR                           0
AMOUNT                         0
LENDER                         0
BORROWER                       1
SECTOR                         0
SENSITIVE TERRITORY OVERLAP    0
Country                        0
dtype: int64
```

```
df.dropna(inplace = True)
```

```
df.nunique()
```

```
Title                          827
YEAR                            12
AMOUNT                         346
```

```
LENDER                          7
BORROWER                      120
SECTOR                         10
SENSITIVE TERRITORY OVERLAP     8
Country                        94
dtype: int64
```

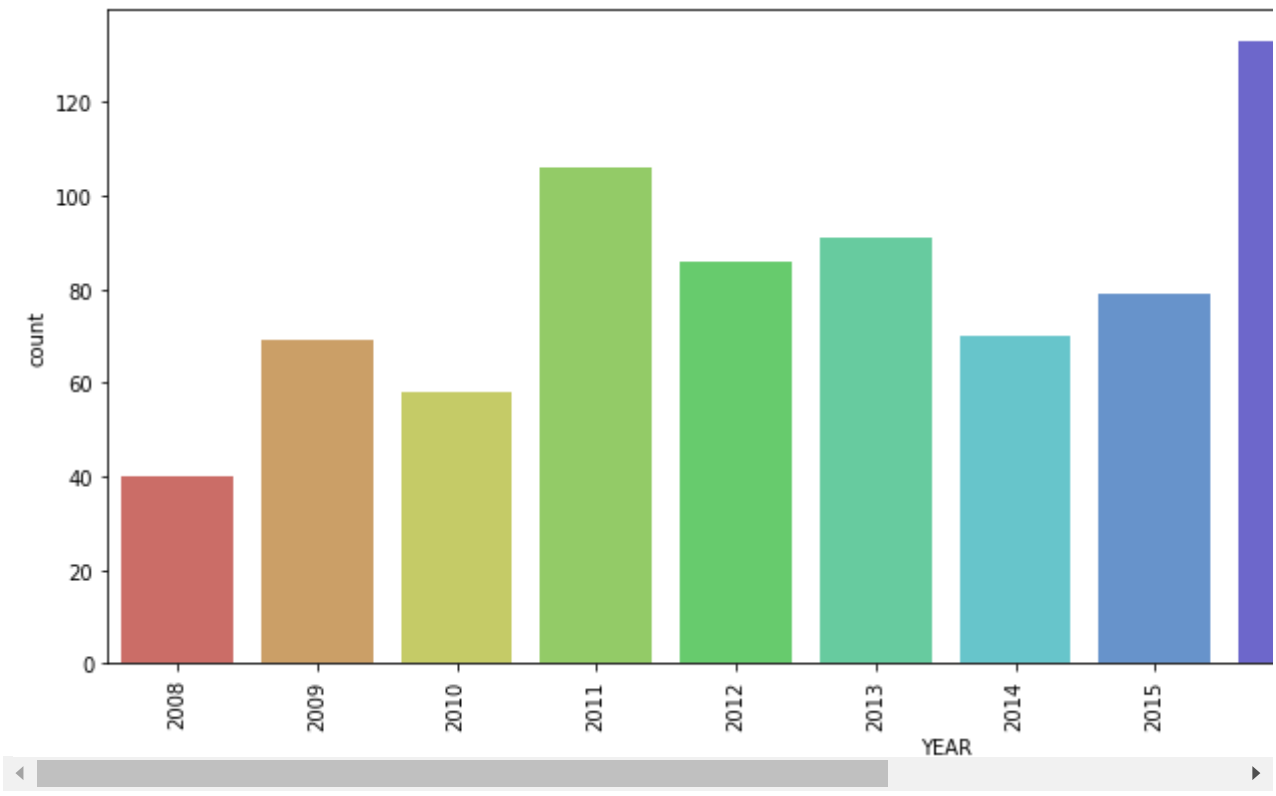
```
df['YEAR'].unique()
```

```
array([2011, 2009, 2012, 2016, 2015, 2017, 2018, 2014, 2008, 2010, 2013,
       2019])
```

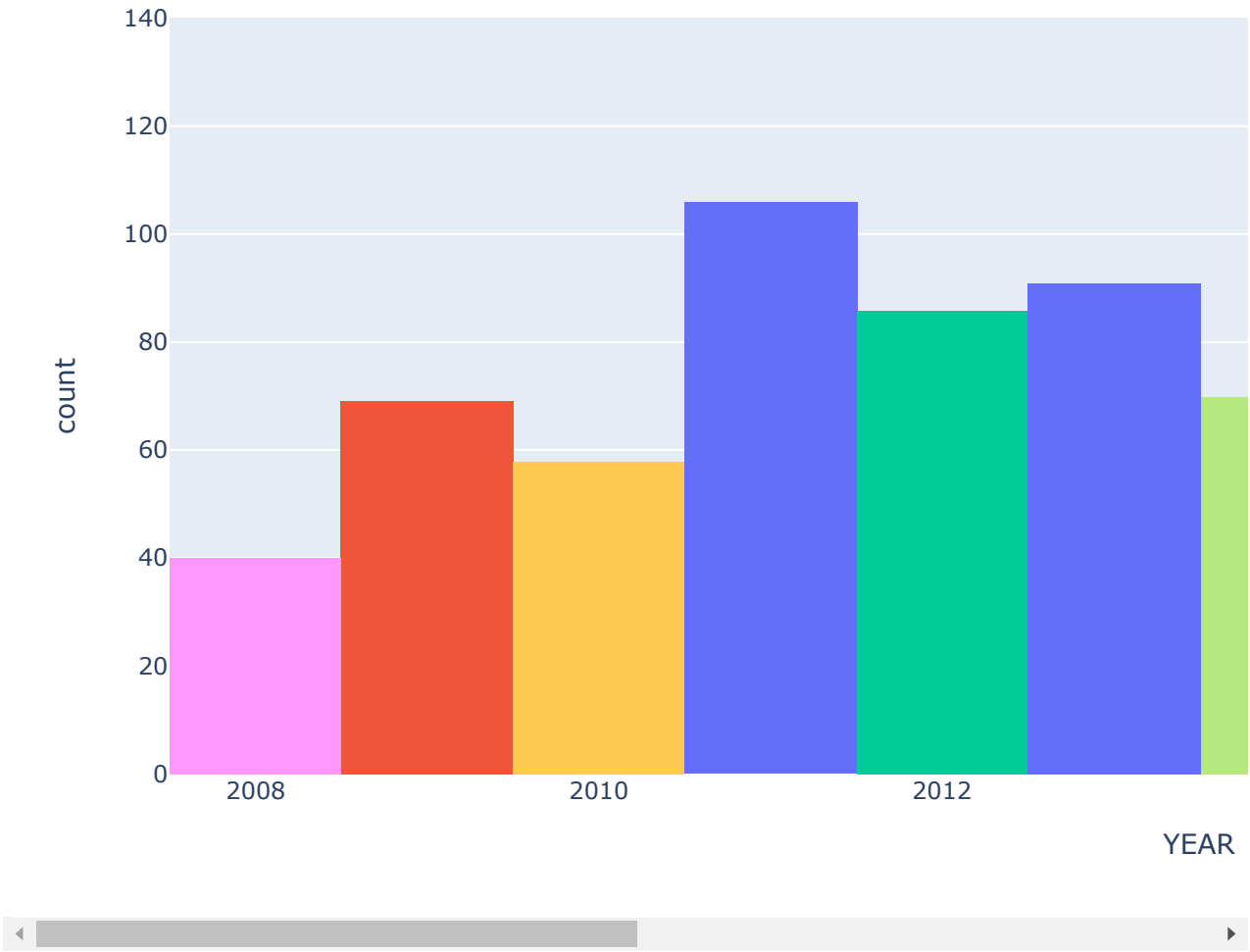
```
df['YEAR'].value_counts()
```

```
2016    133
2011    106
2013     91
2012     86
2015     79
2017     75
2014     70
2009     69
2010     58
2018     45
2008     40
2019      5
Name: YEAR, dtype: int64
```

```
plt.figure(figsize = (15,6))
sns.countplot('YEAR',data = df,palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



```
fig1 = px.histogram(df,x = 'YEAR',color = 'YEAR')
fig1.show()
```



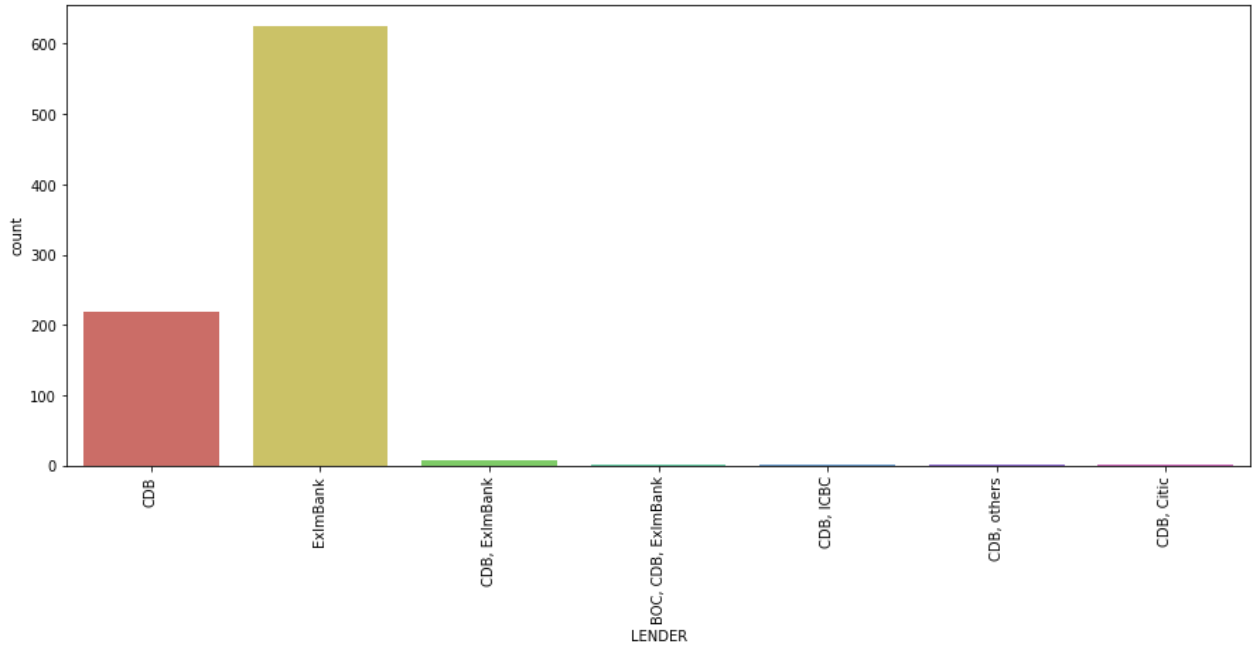

```
df['LENDER'].unique()
```

```
array(['CDB', 'ExImBank', 'CDB, ExImBank', 'BOC, CDB, ExImBank',
       'CDB, ICBC', 'CDB, others', 'CDB, Citic'], dtype=object)
```

```
df['LENDER'].value_counts()
```

```
ExImBank              625
CDB                   219
CDB, ExImBank           8
CDB, ICBC               2
BOC, CDB, ExImBank      1
CDB, others             1
CDB, Citic              1
Name: LENDER, dtype: int64
```

```
plt.figure(figsize = (15,6))
sns.countplot('LENDER',data = df,palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```




```
fig2 = px.histogram(df,x = 'LENDER',color = 'LENDER')
fig2.show()
```

```
df['SECTOR'].unique()
```

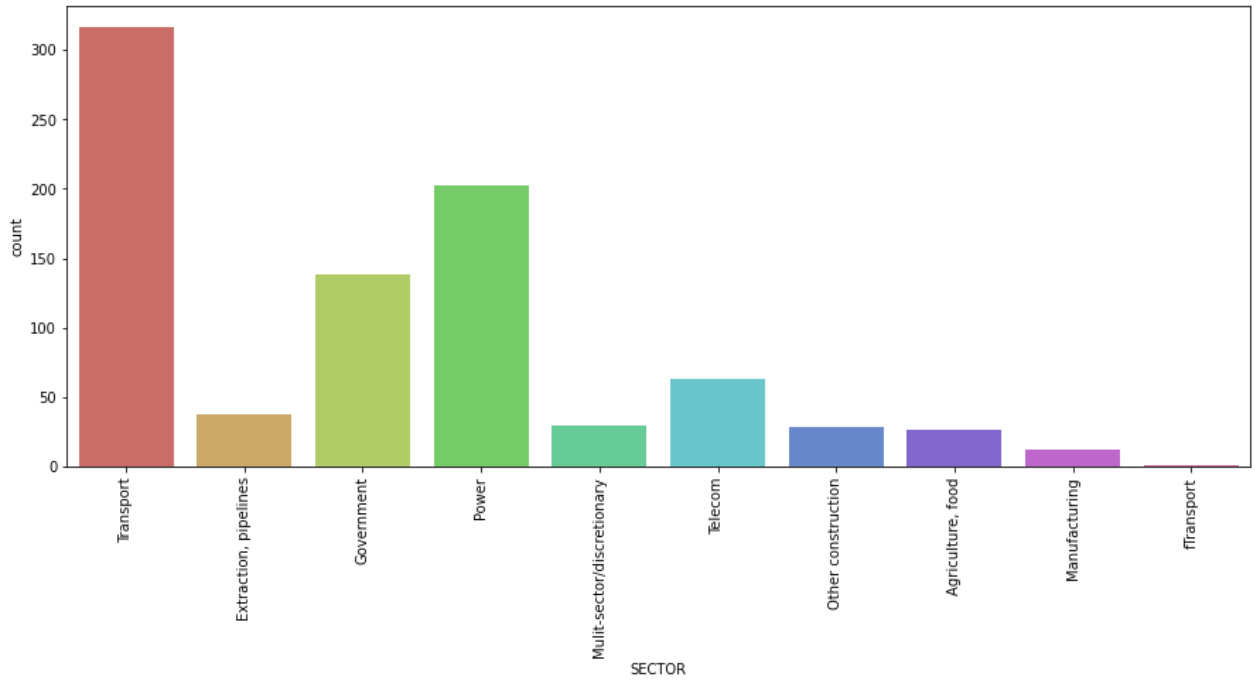
```
array(['Transport', 'Extraction, pipelines', 'Government', 'Power',
       'Mulit-sector/discretionary', 'Telecom', 'Other construction',
       'Agriculture, food', 'Manufacturing', 'fTransport'], dtype=object)
```

```
500
```

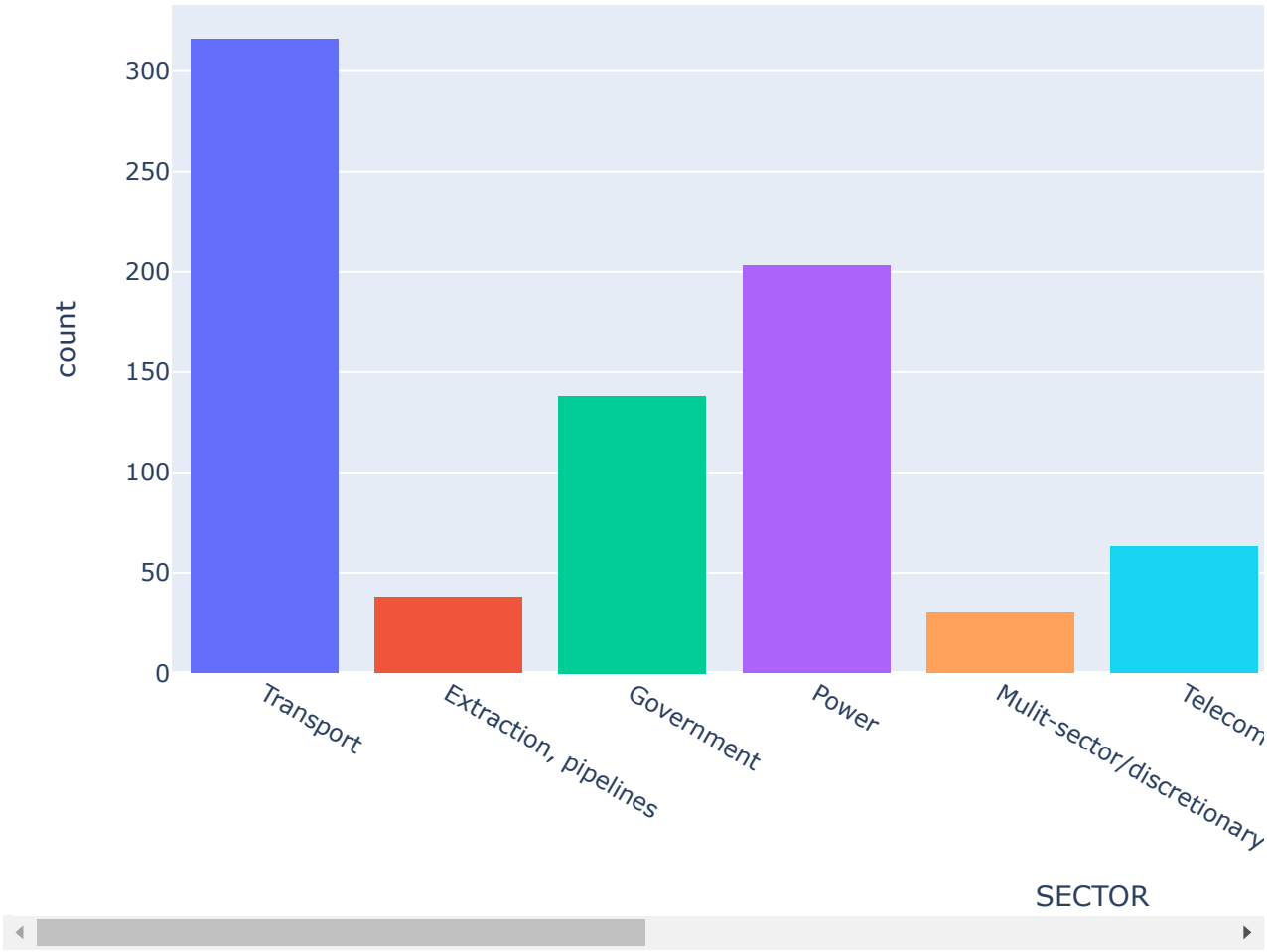
```
df['SECTOR'].value_counts()
```

```
Transport                   316
Power                       203
Government                  138
Telecom                      63
Extraction, pipelines        38
Mulit-sector/discretionary   30
Other construction           29
Agriculture, food            27
Manufacturing                12
fTransport                    1
Name: SECTOR, dtype: int64
```

```
plt.figure(figsize = (15,6))
sns.countplot('SECTOR',data = df,palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



```
fig3 = px.histogram(df,x = 'SECTOR',color = 'SECTOR')
fig3.show()
```



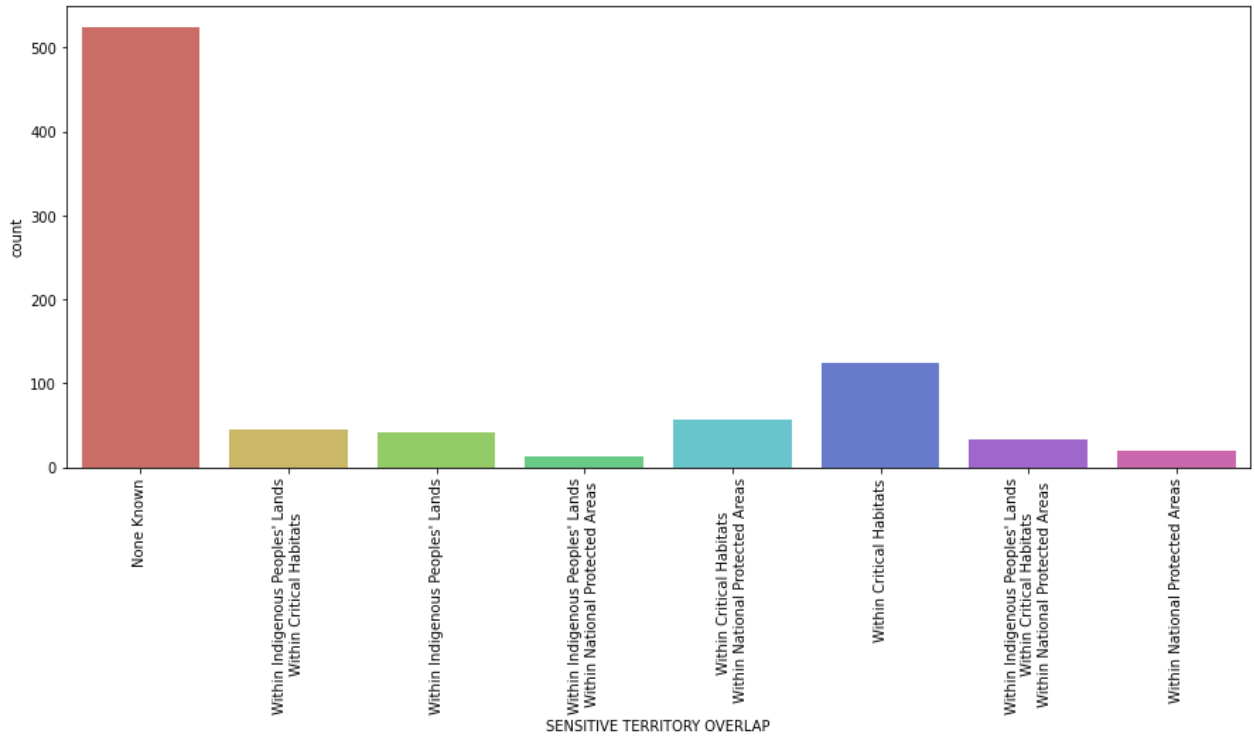

```
df['SENSITIVE TERRITORY OVERLAP'].unique()
```

```
array(['None Known',
       "Within Indigenous Peoples' Lands\n Within Critical Habitats",
       "Within Indigenous Peoples' Lands",
       "Within Indigenous Peoples' Lands\n Within National Protected Areas",
       'Within Critical Habitats\n Within National Protected Areas',
       'Within Critical Habitats',
       "Within Indigenous Peoples' Lands\n Within Critical Habitats\n Within National
       'Within National Protected Areas'], dtype=object)
```

```
df['SENSITIVE TERRITORY OVERLAP'].value_counts()
```

```
None Known
Within Critical Habitats
Within Critical Habitats\n Within National Protected Areas
Within Indigenous Peoples' Lands\n Within Critical Habitats
Within Indigenous Peoples' Lands
Within Indigenous Peoples' Lands\n Within Critical Habitats\n Within National Protect
Within National Protected Areas
Within Indigenous Peoples' Lands\n Within National Protected Areas
Name: SENSITIVE TERRITORY OVERLAP, dtype: int64
```

```
plt.figure(figsize = (15,6))
sns.countplot('SENSITIVE TERRITORY OVERLAP',data = df,palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```




```
df['Country'].unique()
```

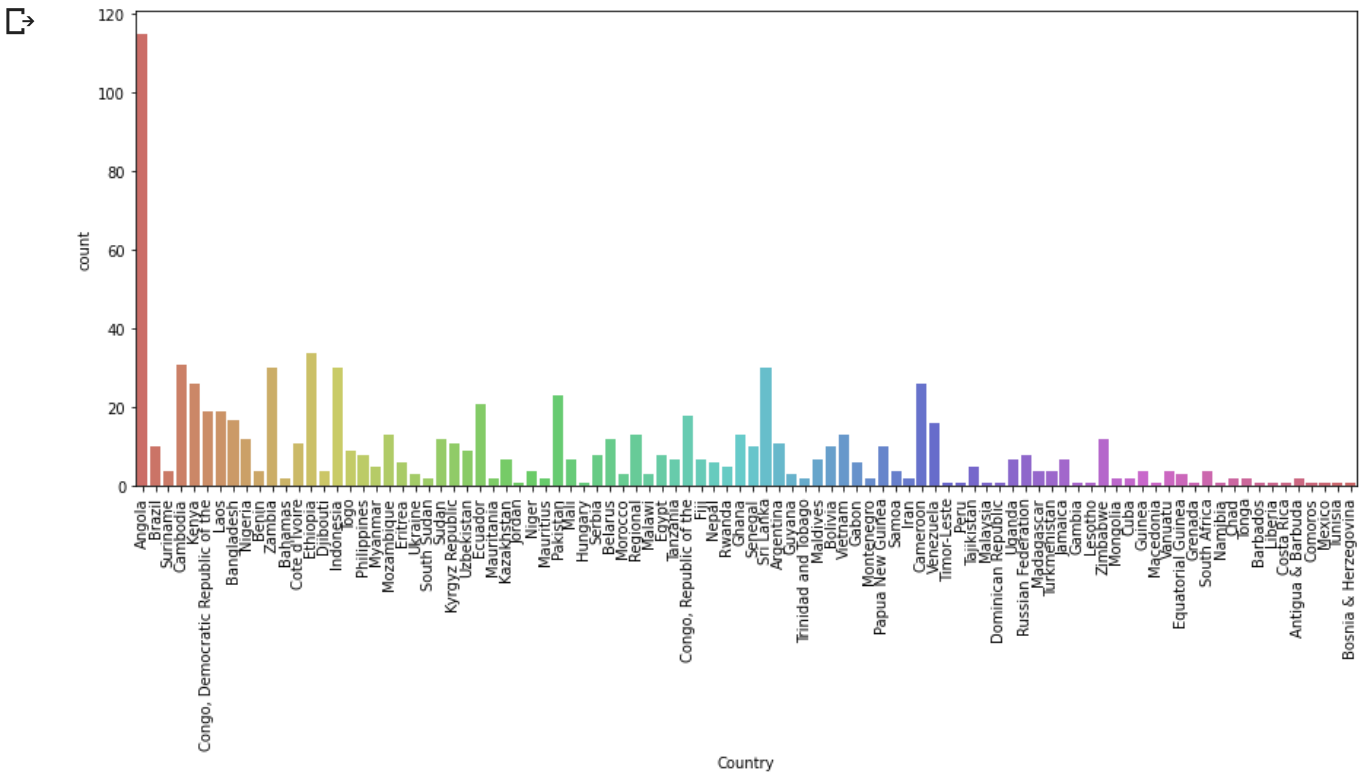
```
array(['Angola', 'Brazil', 'Suriname', 'Cambodia', 'Kenya',
       'Congo, Democratic Republic of the', 'Laos', 'Bangladesh',
       'Nigeria', 'Benin', 'Zambia', 'Bahamas', "Cote d'Ivoire",
       'Ethiopia', 'Djibouti', 'Indonesia', 'Togo', 'Philippines',
       'Myanmar', 'Mozambique', 'Eritrea', 'Ukraine', 'South Sudan',
       'Sudan', 'Kyrgyz Republic', 'Uzbekistan', 'Ecuador', 'Mauritania',
       'Kazakhstan', 'Jordan', 'Niger', 'Mauritius', 'Pakistan', 'Mali',
       'Hungary', 'Serbia', 'Belarus', 'Morocco', 'Regional', 'Malawi',
       'Egypt', 'Tanzania', 'Congo, Republic of the', 'Fiji', 'Nepal',
       'Rwanda', 'Ghana', 'Senegal', 'Sri Lanka', 'Argentina', 'Guyana',
       'Trinidad and Tobago', 'Maldives', 'Bolivia', 'Vietnam', 'Gabon',
       'Montenegro', 'Papua New Guinea', 'Samoa', 'Iran', 'Cameroon',
       'Venezuela', 'Timor-Leste', 'Peru', 'Tajikistan', 'Malaysia',
       'Dominican Republic', 'Uganda', 'Russian Federation', 'Madagascar',
       'Turkmenistan', 'Jamaica', 'Gambia', 'Lesotho', 'Zimbabwe',
       'Mongolia', 'Cuba', 'Guinea', 'Macedonia', 'Vanuatu',
       'Equatorial Guinea', 'Grenada', 'South Africa', 'Namibia', 'Chad',
```

```
       'Tonga', 'Barbados', 'Liberia', 'Costa Rica', 'Antigua & Barbuda',
       'Comoros', 'Mexico', 'Tunisia', 'Bosnia & Herzegovina'],
      dtype=object)
```

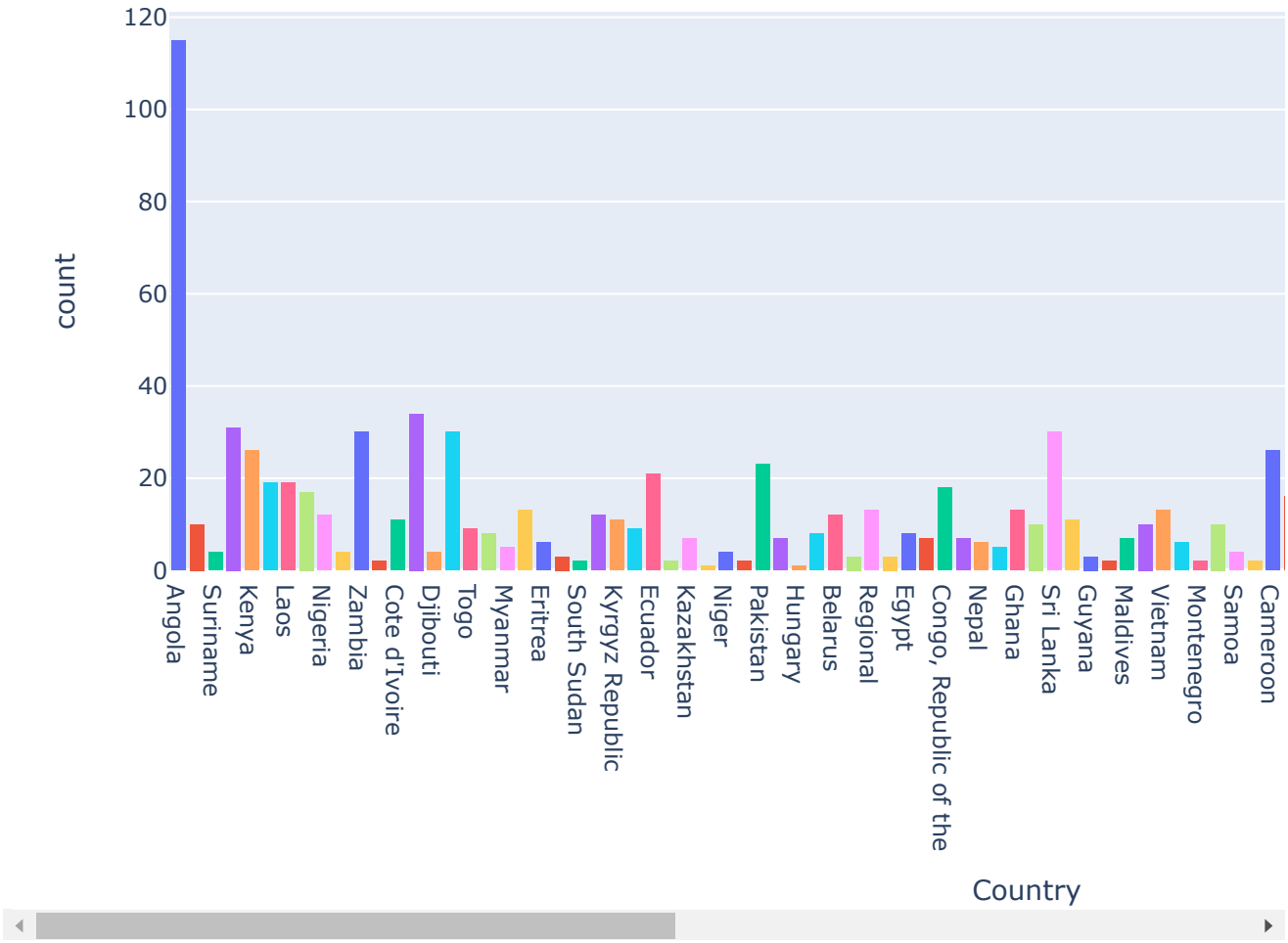
```
df['Country'].value_counts()
```

```
Angola                  115
Ethiopia                 34
Cambodia                 31
Sri Lanka                30
Indonesia                30
                       ...
Malaysia                  1
Peru                      1
Hungary                   1
Jordan                    1
Bosnia & Herzegovina      1
Name: Country, Length: 94, dtype: int64
```

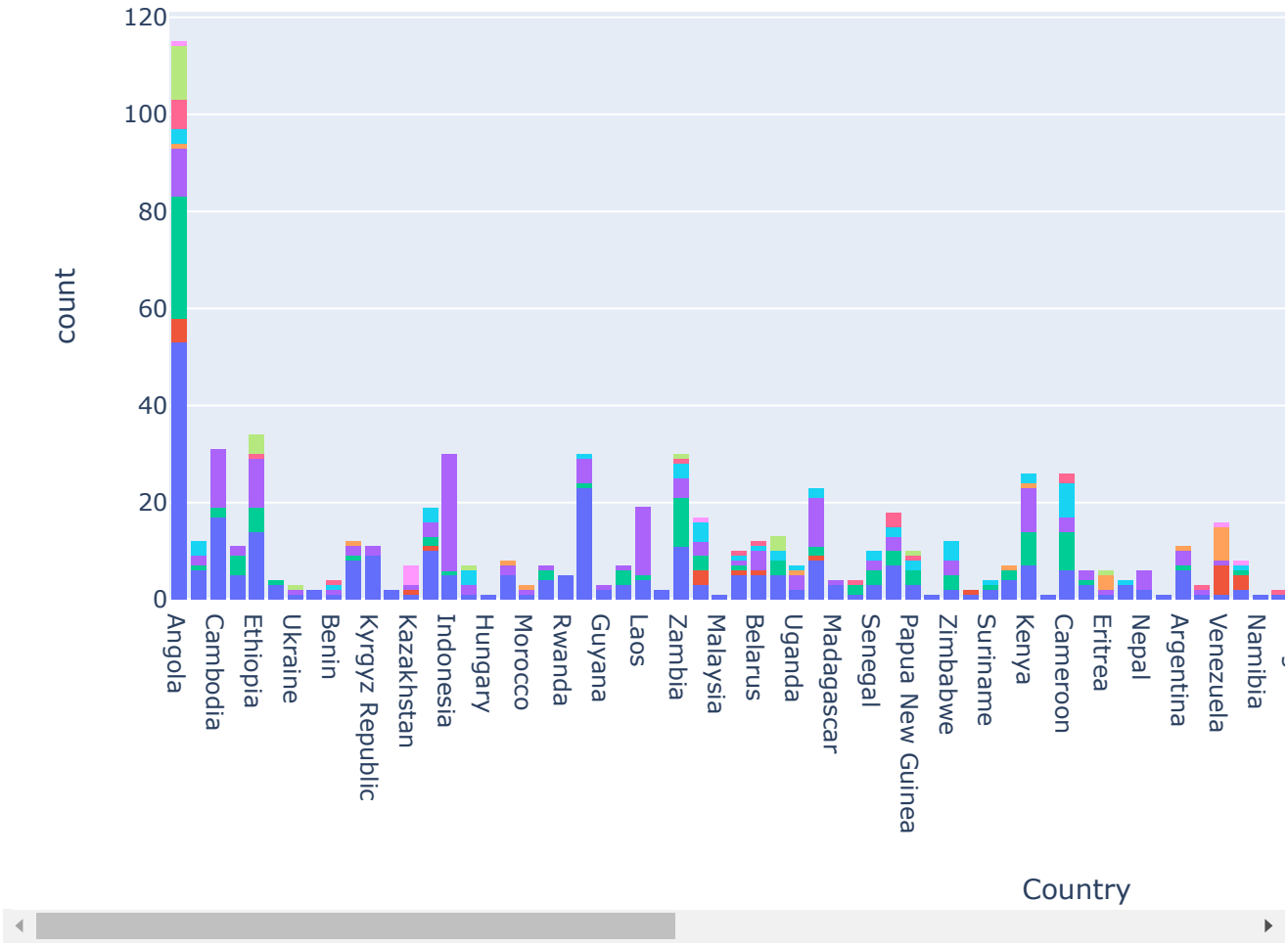
```
plt.figure(figsize = (15,6))
sns.countplot('Country',data = df,palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



```
fig3 = px.histogram(df,x = 'Country',color = 'Country')
fig3.show()
```




```
fig4 = px.histogram(df,x = 'Country',color = 'SECTOR')
fig4.show()
```



✓ 0s completed at 6:15 PM