AFRICAN-CRISIS

In [2]:
```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [3]:
```
# upload dataset:
df= pd.read_csv("F:\\archive (12)\\african_crises.csv")
df.head(2)
```

Out[3]:

	case	cc3	country	year	systemic_crisis	exch_usd	domestic_debt_in_default	sovereign_external_c
0	1	DZA	Algeria	1870	1	0.052264	0	
1	1	DZA	Algeria	1871	0	0.052798	0	

In [6]:
```
# give the spaces at the end
print(df.columns, end=" ")
```

```
Index(['case', 'cc3', 'country', 'year', 'systemic_crisis', 'exch_usd',
       'domestic_debt_in_default', 'sovereign_external_debt_default',
       'gdp_weighted_default', 'inflation_annual_cpi', 'independence',
       'currency_crises', 'inflation_crises', 'banking_crisis'],
      dtype='object')
```

In [7]:
```
# see the first five rows of data:
df.head(5)
```

Out[7]:

	case	cc3	country	year	systemic_crisis	exch_usd	domestic_debt_in_default	sovereign_external_c
0	1	DZA	Algeria	1870	1	0.052264	0	
1	1	DZA	Algeria	1871	0	0.052798	0	
2	1	DZA	Algeria	1872	0	0.052274	0	
3	1	DZA	Algeria	1873	0	0.051680	0	
4	1	DZA	Algeria	1874	0	0.051308	0	

In [8]:
```
# To see the bottm five rows of data
df.tail(5)
```

Out[8]:

	case	cc3	country	year	systemic_crisis	exch_usd	domestic_debt_in_default	sovereign_ext
1054	70	ZWE	Zimbabwe	2009	1	354.8	1	
1055	70	ZWE	Zimbabwe	2010	0	378.2	1	
1056	70	ZWE	Zimbabwe	2011	0	361.9	1	

	case	cc3	country	year	systemic_crisis	exch_usd	domestic_debt_in_default	sovereign_ext
1057	70	ZWE	Zimbabwe	2012	0	361.9	1	
1058	70	ZWE	Zimbabwe	2013	0	361.9	1	

In [9]:

```
# To see the shape of data:
df.shape
```

Out[9]:

```
(1059, 14)
```

In [10]:

```
# To see the size of data:
df.size
```

Out[10]:

```
14826
```

In [11]:

```
# To see the columns of data :
df.columns
```

Out[11]:

```
Index(['case', 'cc3', 'country', 'year', 'systemic_crisis', 'exch_usd',
       'domestic_debt_in_default', 'sovereign_external_debt_default',
       'gdp_weighted_default', 'inflation_annual_cpi', 'independence',
       'currency_crises', 'inflation_crises', 'banking_crisis'],
      dtype='object')
```

In [12]:

```
# give the infomation of data :
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 14 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   case                             1059 non-null   int64  
 1   cc3                              1059 non-null   object 
 2   country                          1059 non-null   object 
 3   year                             1059 non-null   int64  
 4   systemic_crisis                  1059 non-null   int64  
 5   exch_usd                         1059 non-null   float64
 6   domestic_debt_in_default         1059 non-null   int64  
 7   sovereign_external_debt_default  1059 non-null   int64  
 8   gdp_weighted_default             1059 non-null   float64
 9   inflation_annual_cpi             1059 non-null   float64
 10  independence                     1059 non-null   int64  
 11  currency_crises                  1059 non-null   int64  
 12  inflation_crises                 1059 non-null   int64  
 13  banking_crisis                   1059 non-null   object 
dtypes: float64(3), int64(8), object(3)
memory usage: 116.0+ KB
```

In [13]:

```
# To see the statiscal summary or descrption of data:
df.describe
```

Out[13]:

```
<bound method NDFrame.describe of       case  cc3   country  year  systemic_crisis
exch_usd  \
0        1  DZA   Algeria  1870               1    0.052264
1        1  DZA   Algeria  1871               0    0.052798
```

```
2        1  DZA   Algeria  1872           0    0.052274
3        1  DZA   Algeria  1873           0    0.051680
4        1  DZA   Algeria  1874           0    0.051308
...    ...  ...       ...   ...         ...         ...
1054    70  ZWE  Zimbabwe  2009           1  354.800000
1055    70  ZWE  Zimbabwe  2010           0  378.200000
1056    70  ZWE  Zimbabwe  2011           0  361.900000
1057    70  ZWE  Zimbabwe  2012           0  361.900000
1058    70  ZWE  Zimbabwe  2013           0  361.900000

      domestic_debt_in_default  sovereign_external_debt_default  \
0                            0                                0
1                            0                                0
2                            0                                0
3                            0                                0
4                            0                                0
...                        ...                              ...
1054                         1                                1
1055                         1                                1
1056                         1                                1
1057                         1                                1
1058                         1                                1

      gdp_weighted_default  inflation_annual_cpi  independence  \
0                      0.0              3.441456             0
1                      0.0             14.149140             0
2                      0.0             -3.718593             0
3                      0.0             11.203897             0
4                      0.0             -3.848561             0
...                    ...                   ...           ...
1054                   0.0             -7.670000             1
1055                   0.0              3.217000             1
1056                   0.0              4.920000             1
1057                   0.0              3.720000             1
1058                   0.0              1.632000             1

      currency_crises  inflation_crises banking_crisis
0                   0                 0         crisis
1                   0                 0      no_crisis
2                   0                 0      no_crisis
3                   0                 0      no_crisis
4                   0                 0      no_crisis
...               ...               ...            ...
1054                1                 0         crisis
1055                0                 0      no_crisis
1056                0                 0      no_crisis
1057                0                 0      no_crisis
1058                0                 0      no_crisis

[1059 rows x 14 columns]>
```

In [14]:

```
# see the null values whether it is present in data or not, there is no null values so no further null
# values treatment is required :
df.isnull().sum()
```

Out[14]:

```
case                               0
cc3                                0
country                            0
year                               0
systemic_crisis                    0
exch_usd                           0
domestic_debt_in_default           0
sovereign_external_debt_default    0
gdp_weighted_default               0
inflation_annual_cpi               0
```

```
independence                  0
currency_crises               0
inflation_crises              0
banking_crisis                0
dtype: int64
```

In [15]:
```
# To see the data types in the data , whether integer, float or an object:
df.dtypes
```

Out[15]:
```
case                               int64
cc3                               object
country                           object
year                               int64
systemic_crisis                    int64
exch_usd                         float64
domestic_debt_in_default           int64
sovereign_external_debt_default    int64
gdp_weighted_default             float64
inflation_annual_cpi             float64
independence                       int64
currency_crises                    int64
inflation_crises                   int64
banking_crisis                    object
dtype: object
```

In [16]:
```
# To finding the unique values of country column in data , it makes an array of listing countries in
# the data:
df.country.unique()
```

Out[16]:
```
array(['Algeria', 'Angola', 'Central African Republic', 'Ivory Coast',
       'Egypt', 'Kenya', 'Mauritius', 'Morocco', 'Nigeria',
       'South Africa', 'Tunisia', 'Zambia', 'Zimbabwe'], dtype=object)
```

In [17]:
```
# To see the two important columns together for comparison:
df[['country', 'year']]
```

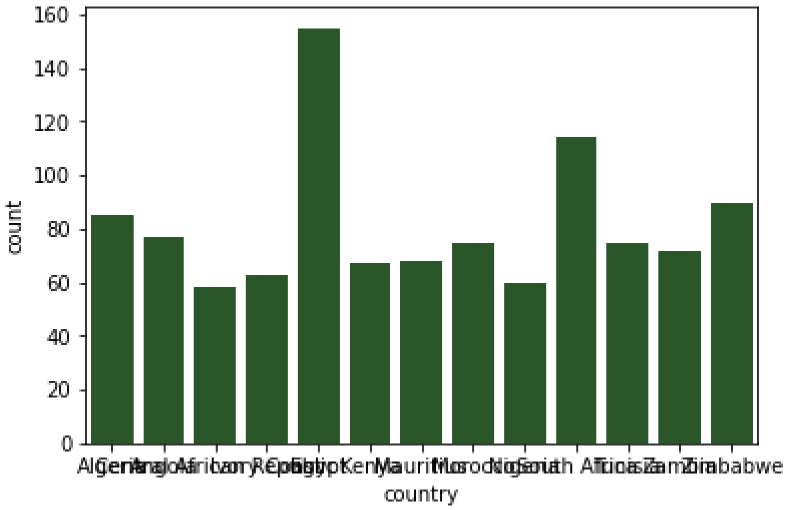
Out[17]:

	country	year
0	Algeria	1870
1	Algeria	1871
2	Algeria	1872
3	Algeria	1873
4	Algeria	1874
...	...	...
1054	Zimbabwe	2009
1055	Zimbabwe	2010
1056	Zimbabwe	2011
1057	Zimbabwe	2012
1058	Zimbabwe	2013

1059 rows × 2 columns

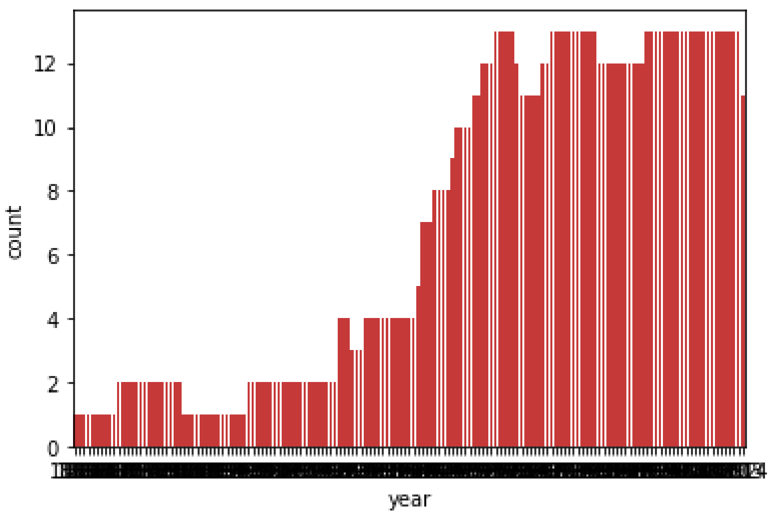
In [18]:
```python
# Make a countplot of important variable like country is here...
sns.countplot(x='country',data=df, color='green', saturation=0.35)
```

Out[18]: <AxesSubplot:xlabel='country', ylabel='count'>



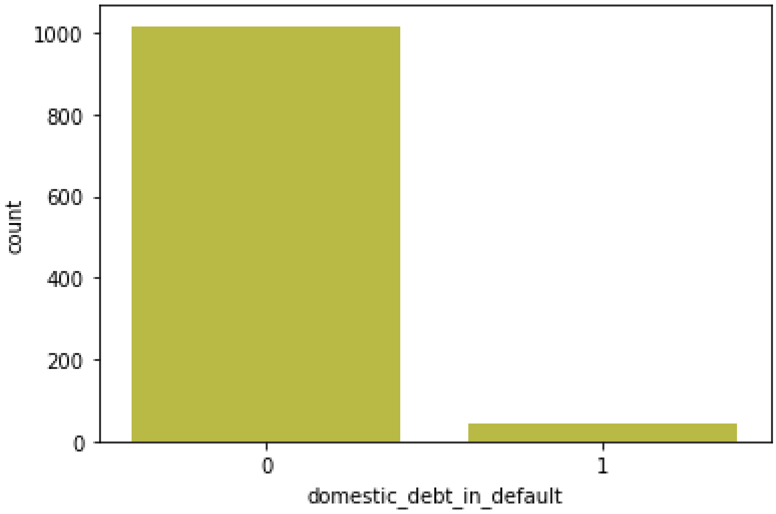
In [19]:
```python
# Make a countplot of year, one of the important variable of data...
sns.countplot(x='year',data=df, color='red', saturation=0.55)
```

Out[19]: <AxesSubplot:xlabel='year', ylabel='count'>



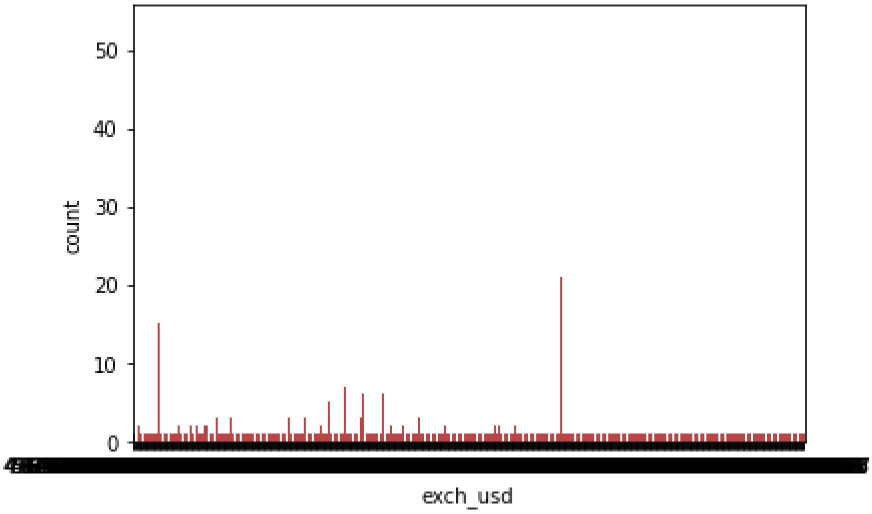
In [20]:
```python
# Make a countplot of domestic debt in default,how much domestic debt
here, debt is higher, one of the important variable of data...
sns.countplot(x='domestic_debt_in_default',data=df, color='yellow',
saturation=0.45)
```

Out[20]: <AxesSubplot:xlabel='domestic_debt_in_default', ylabel='count'>



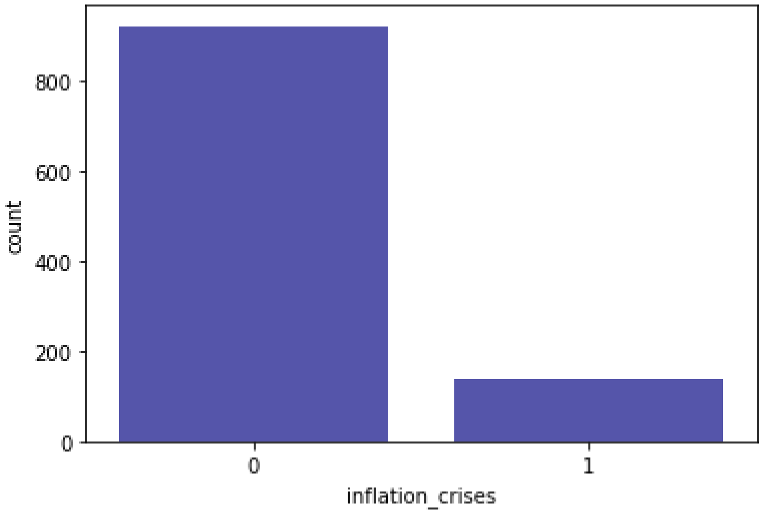
In [40]:
```
# Make a countplot of exch_usd which one is the important variable of data and it is fluctuating
sns.countplot(x='exch_usd',data=df, color='red', saturation=0.45)
```

Out[40]: <AxesSubplot:xlabel='exch_usd', ylabel='count'>



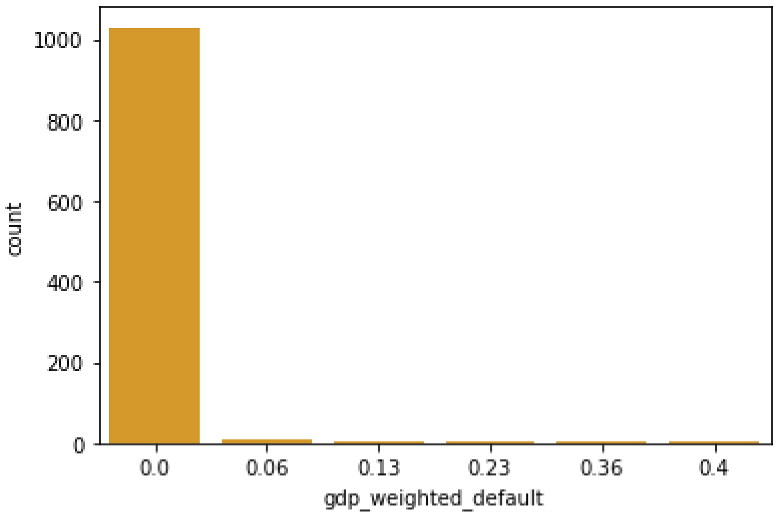
In [22]:
```
# Make a countplot of infaltion crisis,it shows inflation crisis is higher, one of the important variable of data...
sns.countplot(x='inflation_crises',data=df, color='blue', saturation=0.33)
```

Out[22]: <AxesSubplot:xlabel='inflation_crises', ylabel='count'>



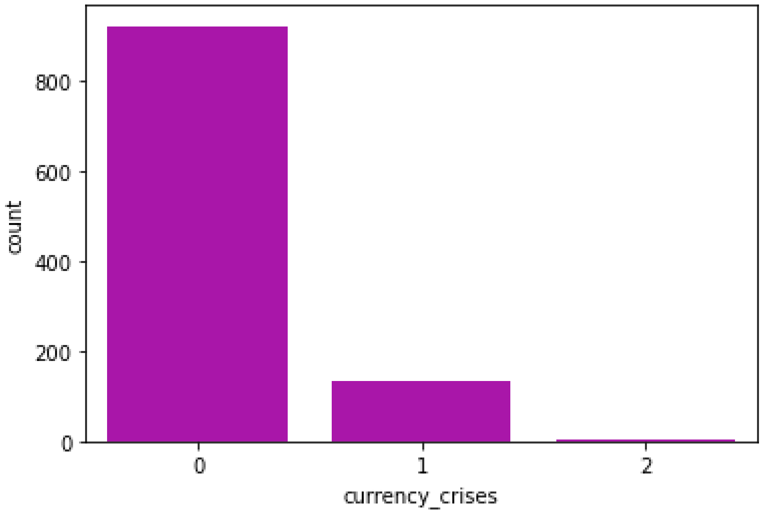
In [23]:
```
# Make a countplot of gdp weighted default which tells us how much gdp,
one of the important variable of data...
sns.countplot(x='gdp_weighted_default',data=df, color='orange',
saturation=0.66)
```

Out[23]: <AxesSubplot:xlabel='gdp_weighted_default', ylabel='count'>



In [24]:
```
# Make a countplot of card no of data, one of the important variable of
data, it shows most of the
# countries have currency crises..very less countries facing currency
crisis as compare to others..
sns.countplot(x='currency_crises',data=df, color='m', saturation=0.77)
```

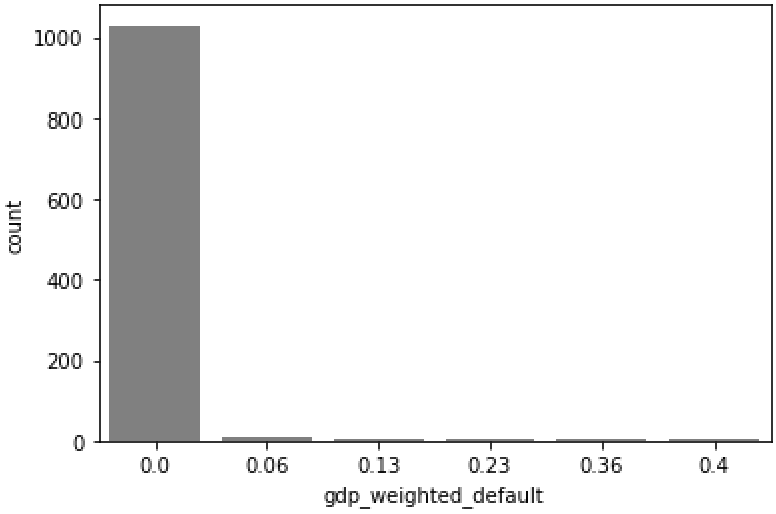
Out[24]: <AxesSubplot:xlabel='currency_crises', ylabel='count'>



In [25]:

```
# Make a countplot of card no of data, one of the important variable of data...it shows most of the countrie
# going throgh with 0 gdp weighted default.., other are sumthing more than zero.
sns.countplot(x='gdp_weighted_default',data=df, color='grey', saturation=0.88)
```

Out[25]: <AxesSubplot:xlabel='gdp_weighted_default', ylabel='count'>



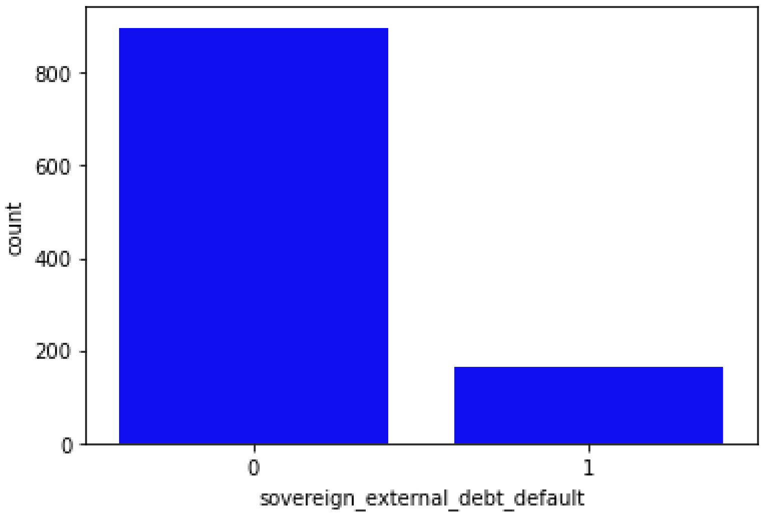
In [ ]:

```
# BLANK CELL
```

In [27]:

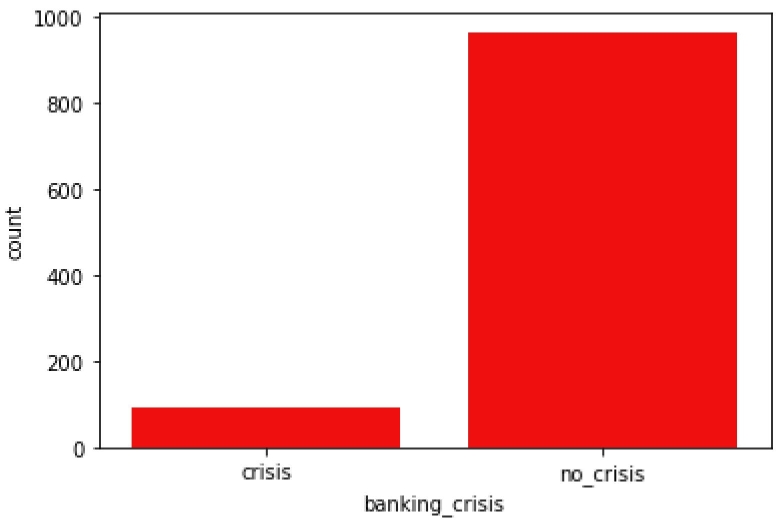
```
# Make a countplot of card no of data, one of the important variable of data...it shows most of the countries
# fails to pay back its loan to domestic or international creditors, so there is most of huge sovereign
# external debt default as comparare to other countries

sns.countplot(x='sovereign_external_debt_default',data=df, color='blue', saturation=0.88)
```

Out[27]: <AxesSubplot:xlabel='sovereign_external_debt_default', ylabel='count'>



In [28]:
```
# Make a countplot of card no of data, one of the important variable of data...its shows most of the
# countries have less or No banking crisis but the others have some.
sns.countplot(x='banking_crisis',data=df, color='red', saturation=0.88)
```

Out[28]: <AxesSubplot:xlabel='banking_crisis', ylabel='count'>



In [29]:
```
# find the correlation between the dependent and independent variables .
corr = df.corr()
```

In [30]:
```
corr
```

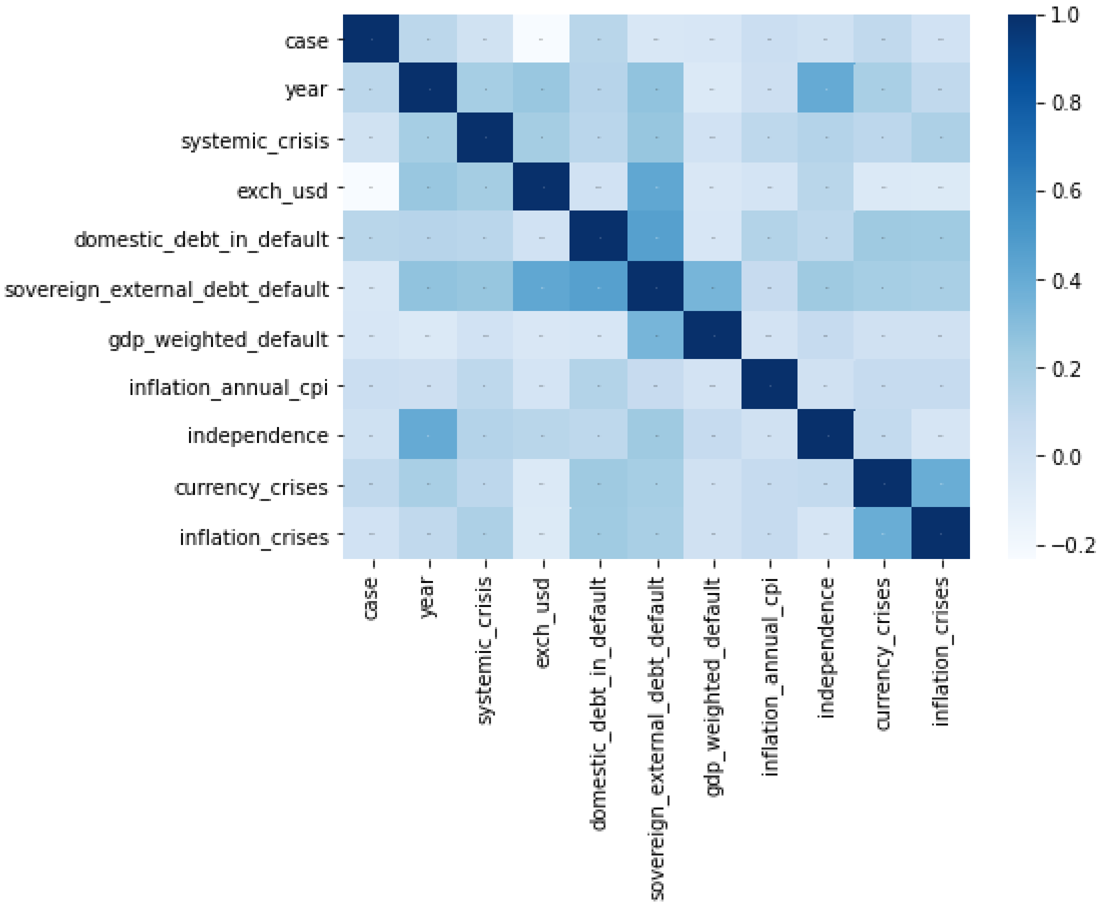
Out[30]:

	case	year	systemic_crisis	exch_usd	domestic_debt_in_defau
case	1.000000	0.115574	0.010991	-0.231976	0.1283
year	0.115574	1.000000	0.197450	0.248757	0.1368
systemic_crisis	0.010991	0.197450	1.000000	0.202687	0.1221
exch_usd	-0.231976	0.248757	0.202687	1.000000	0.0052
domestic_debt_in_default	0.128358	0.136828	0.122158	0.005253	1.0000

	case	year	systemic_crisis	exch_usd	domestic_debt_in_defau
sovereign_external_debt_default	-0.039262	0.271890	0.249850	0.422890	0.4647
gdp_weighted_default	-0.032981	-0.054670	0.005274	-0.040726	-0.0298
inflation_annual_cpi	0.044762	0.037035	0.106452	-0.011947	0.1518
independence	0.021858	0.407360	0.147083	0.126034	0.1091
currency_crises	0.095339	0.189390	0.112751	-0.056472	0.2275
inflation_crises	0.006405	0.098630	0.172562	-0.063783	0.2244

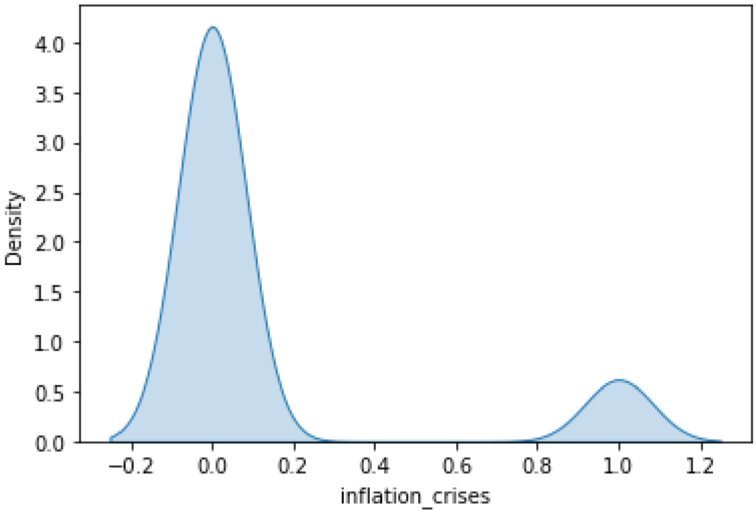
In [31]:
```
# here, we are drawing a heat map to see the relationship between nthe variables , how much they are
# correlated to each other, darker values have good relation however, lighter have less compatible .
plt.figure(figsize=(7,5))
sns.heatmap(corr, annot=True, annot_kws={'size': 0.01}, cmap="Blues")
```

Out[31]: <AxesSubplot:>



In [32]:
```
 # Draw a kde plot to see fluctuation in inflation crises of the countries, intially inflation crisis
 # was high and faltten down with stability later on they goes little arised.
sns.kdeplot(df['inflation_crises'], shade = True)
```

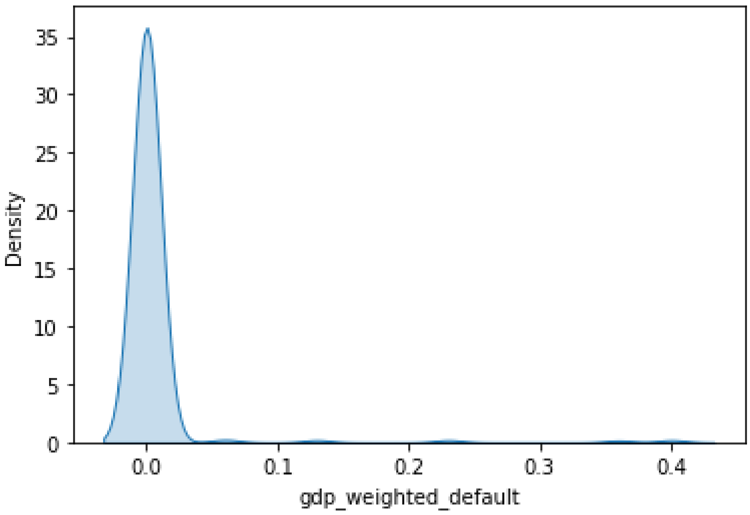
Out[32]: <AxesSubplot:xlabel='inflation_crises', ylabel='Density'>



In [33]:

```
# this kde plot showing how gdp goes , gdp goes super high with zero then it comes flatten down with
# good stability..
sns.kdeplot(df['gdp_weighted_default'], shade = True)
```

Out[33]: <AxesSubplot:xlabel='gdp_weighted_default', ylabel='Density'>



In [34]:

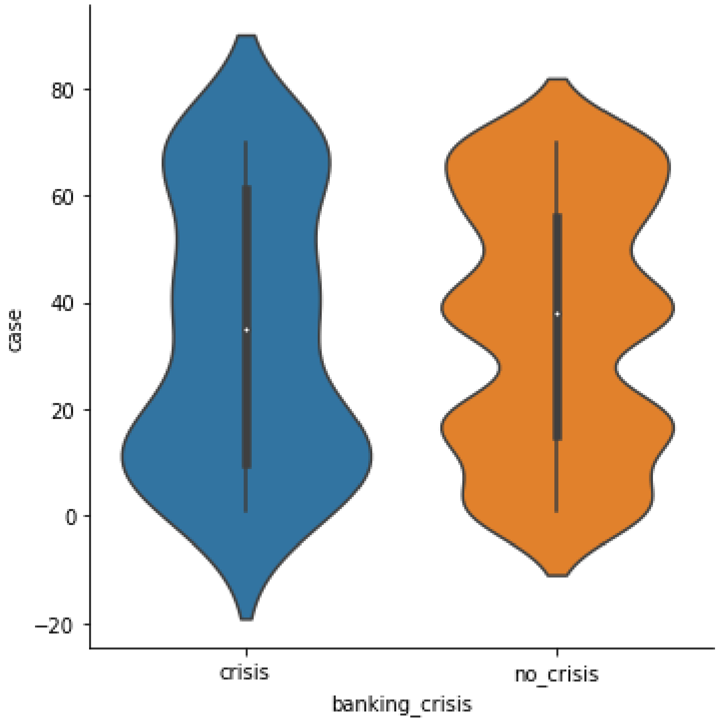
```
#This give all details about number of rows and columns registered in case "n"

df[df.case==10].shape
```

Out[34]: (58, 14)

In [35]:

```
#sns.violinplot(df.banking_crisis, df.case)
#OR
sns.catplot(data=df, x= 'banking_crisis', y="case",kind="violin")
```

Out[35]: <seaborn.axisgrid.FacetGrid at 0x1d7a781bb80>



In [36]:
```
# groupby for banking crisis count with their crisis or no crisis
categories:
df.groupby('banking_crisis').country.count()
```

Out[36]:
```
banking_crisis
crisis         94
no_crisis     965
Name: country, dtype: int64
```

In [37]:
```
# groupby of columns or variables for comparison..
df.groupby(['banking_crisis','country']).describe().unstack().T
```

Out[37]:

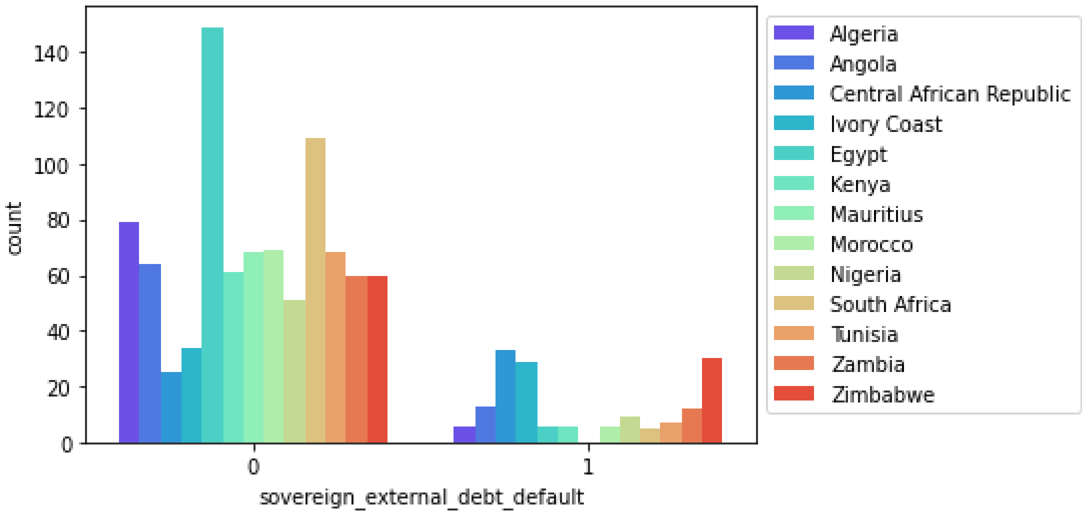
		banking_crisis	crisis	no_crisis
		country		
case	**count**	**Algeria**	4.0	81.0
		Angola	7.0	70.0
		Central African Republic	19.0	39.0
		Egypt	11.0	144.0
		Ivory Coast	4.0	59.0
...	**...**	**...**	...	...
inflation_crises	**max**	**Nigeria**	1.0	1.0
		South Africa	0.0	1.0
		Tunisia	0.0	1.0
		Zambia	1.0	1.0
		Zimbabwe	1.0	1.0

1144 rows × 2 columns

In [78]:

```
# #this representation shows that "Egypt" remains the african country with the least sovereign_external_debt_default within the complete year of study
#Also, this representation shows that "Central African Republic" remains the african country with the most sovereign_external_debt_default within the complete year of study
# both bars are different but in one place right and left, right bar chart for high sovereignity where,
# Central african republic is high and left chart for least sovereignity where egypt has least sovereign debt.
sns.countplot(x = 'sovereign_external_debt_default', hue= 'country', data= df, palette= 'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

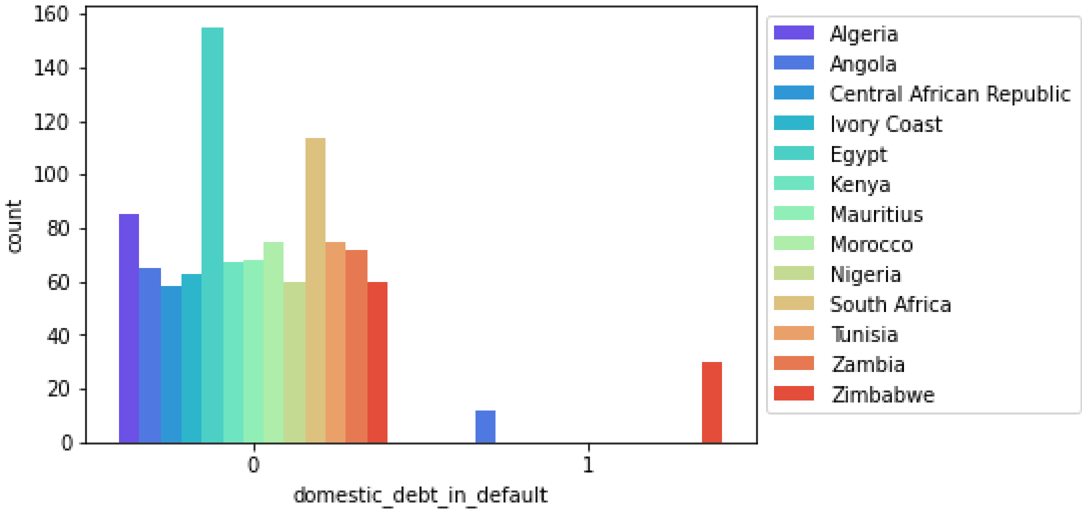
Out[78]: <matplotlib.legend.Legend at 0x2216b956c40>



In [81]:

```
# #this representation reveals that about very less amount of african countries have a domestic_debt_in_default within the year of observation.
sns.countplot(x = 'domestic_debt_in_default', hue= 'country', data= df, palette= 'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

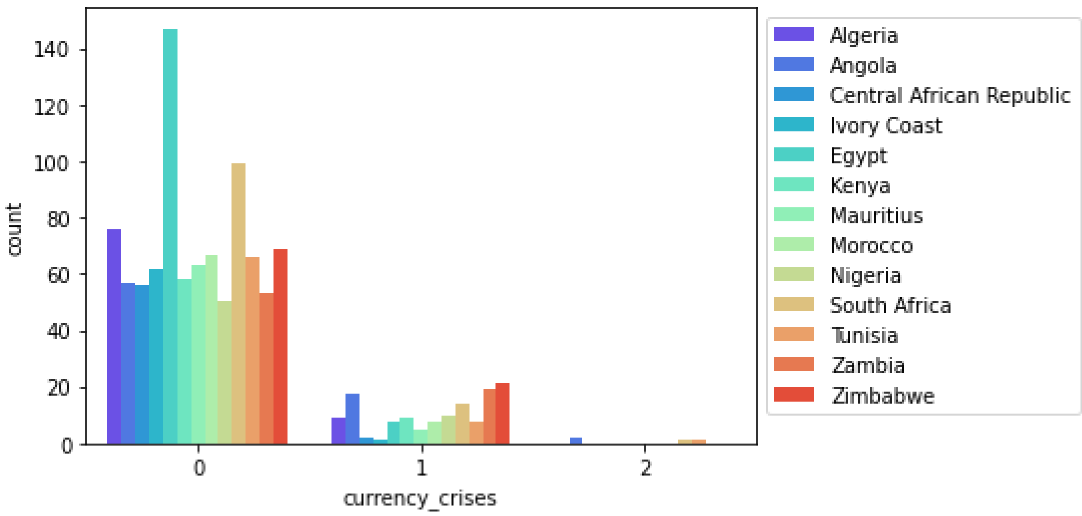
Out[81]: <matplotlib.legend.Legend at 0x2216b9a9a90>



In [82]:

```
#this representation shows that "Egypt" remains the african country with the least currency_crises within the complete year of stud
sns.countplot(x = 'currency_crises', hue= 'country', data= df, palette= 'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

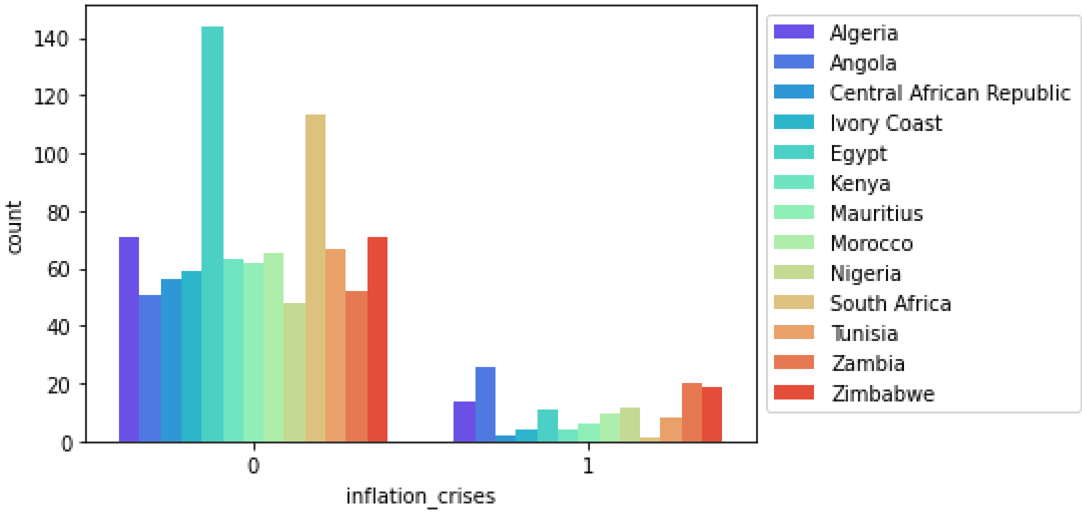
Out[82]: <matplotlib.legend.Legend at 0x2216b997c70>



In [83]:

```
#this representation shows that "Egypt" remains the african country with the least inflation_crises within the complete year of
sns.countplot(x = 'inflation_crises', hue= 'country', data= df, palette= 'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

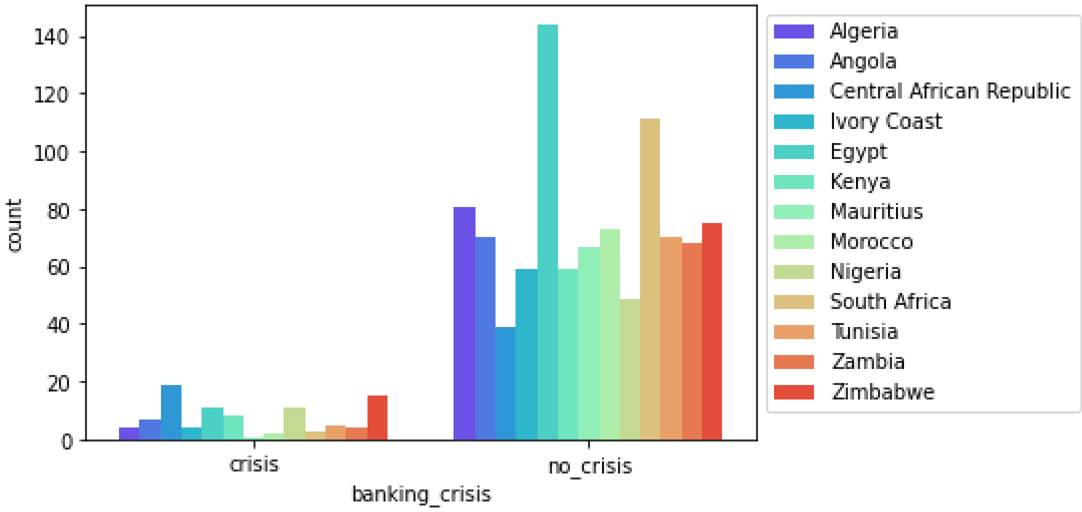
Out[83]: <matplotlib.legend.Legend at 0x2216a8bb940>



In [84]:

```
#this representation shows that "Egypt" remains the african country with
the least banking_crises within the complete year of study

sns.countplot(x = 'banking_crisis', hue= 'country', data= df, palette=
'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

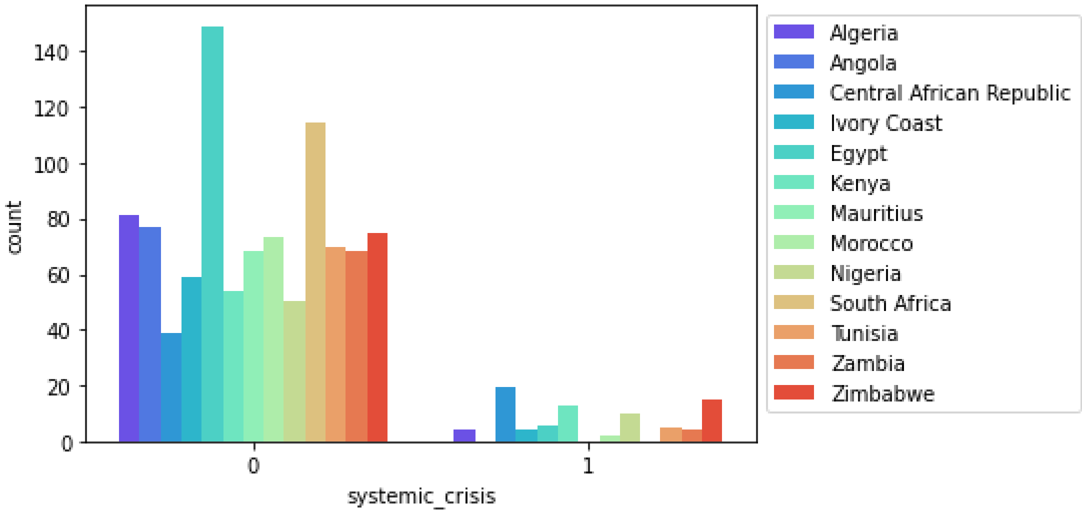
Out[84]: <matplotlib.legend.Legend at 0x22169b11700>



In [85]:

```
#this representation shows that "Egypt" remains the african country with
the least systemic_crises within the complete year of study

sns.countplot(x = 'systemic_crisis', hue= 'country', data= df, palette=
'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

Out[85]: <matplotlib.legend.Legend at 0x2216a8cf040>



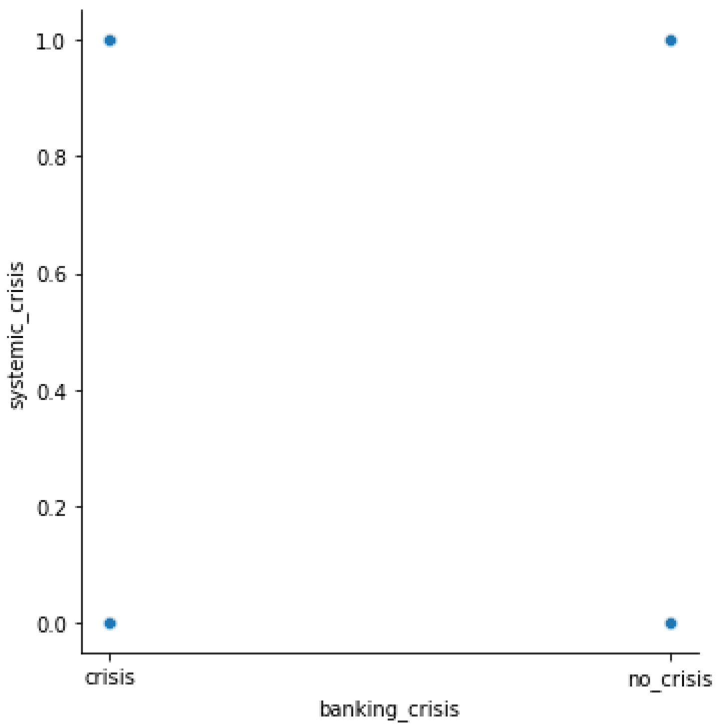
In [ ]:

```
# BLANK CELL--------
```

In [88]:

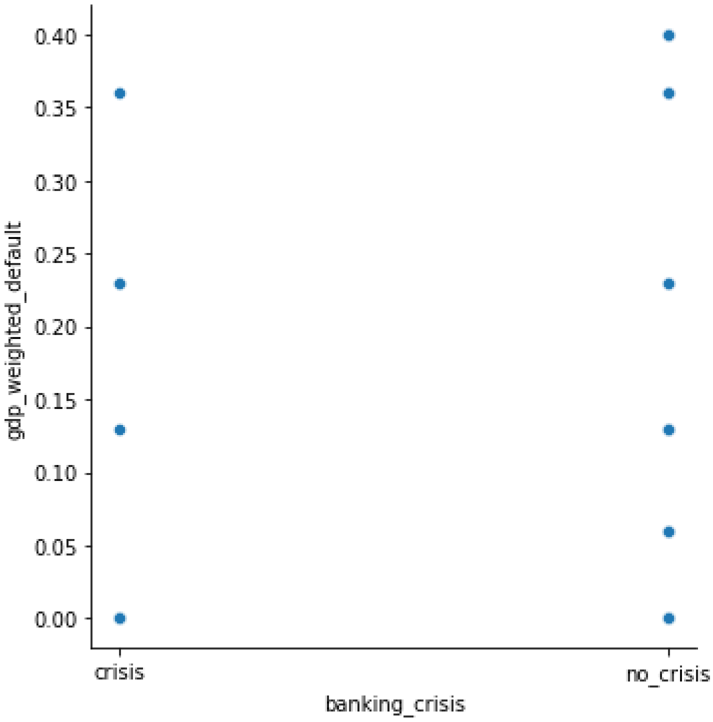
```
# relational plots ,#this relationship graph between banking_crisis and systemic_crisis

#The graph shows that the banking_crisis is directly proportional to the systemic_crisis
# In other words, the higher the banking crisis, the lower the systematic crisis. and vice versa

sns.relplot(x='banking_crisis', y='systemic_crisis', data=df)
```

Out[88]: <seaborn.axisgrid.FacetGrid at 0x2216be142e0>



In [92]:
```
# relational plot:
sns.relplot(x='banking_crisis', y='gdp_weighted_default', data=df)
```

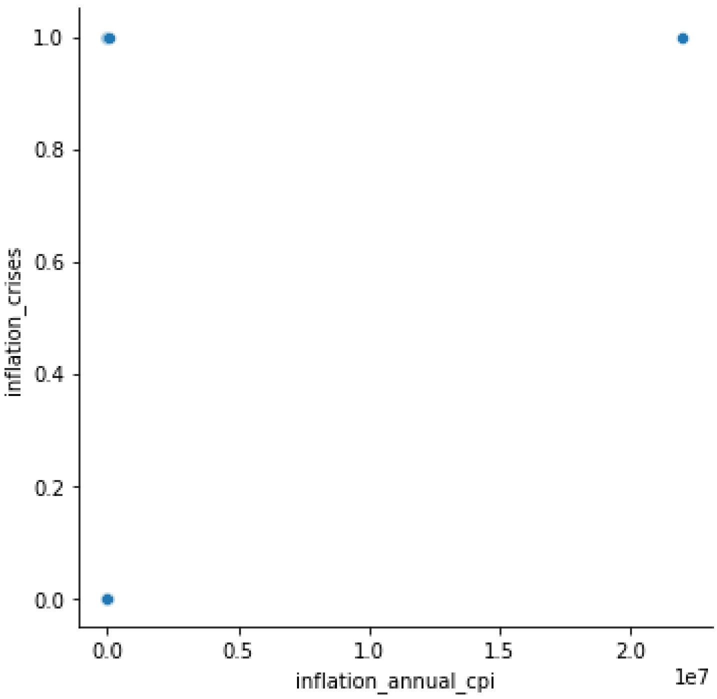
Out[92]: <seaborn.axisgrid.FacetGrid at 0x2216bce6820>



In [19]:
```
# BLANK CELL----
```

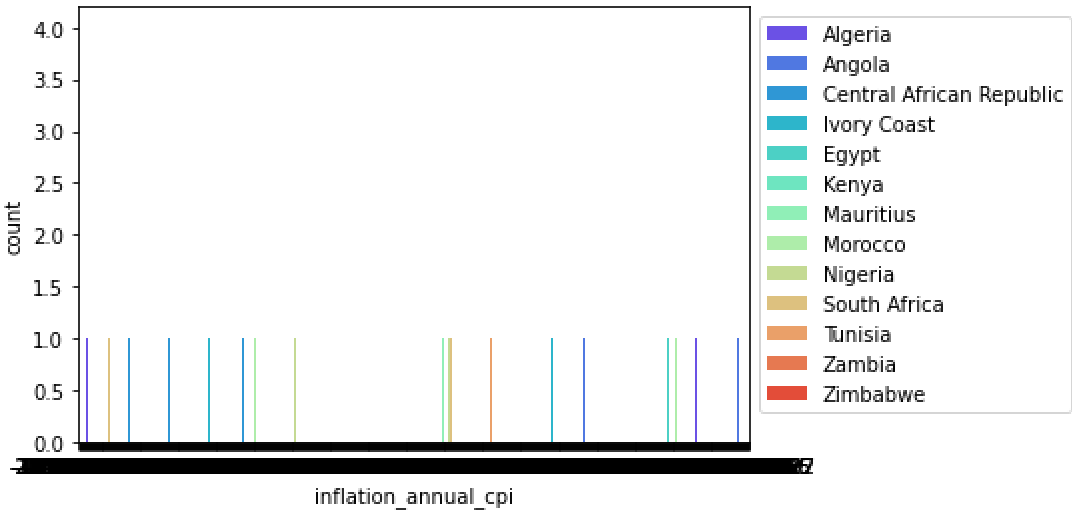
In [93]:
```
# rel plot of inflation and infaltion annual inflation cpi
sns.relplot(x='inflation_annual_cpi', y='inflation_crises', data=df)
```

Out[93]: <seaborn.axisgrid.FacetGrid at 0x2216bd9ddc0>



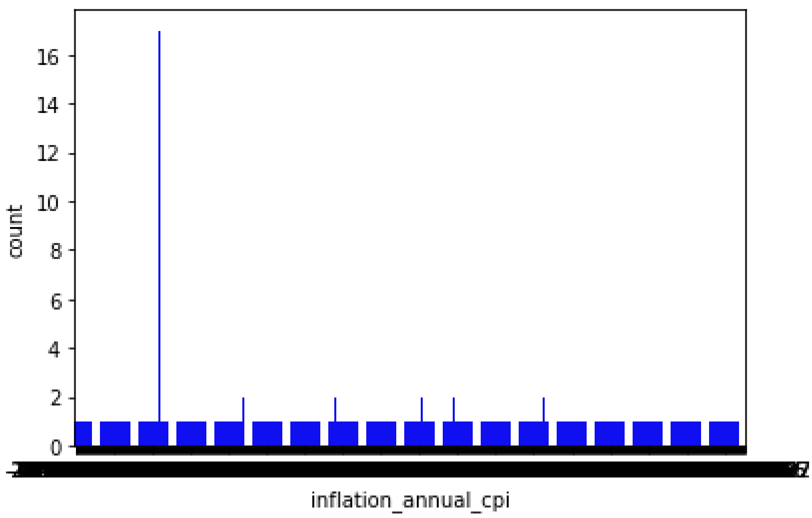
In [44]:
```
# cpi is equal for all countries
sns.countplot(x = 'inflation_annual_cpi', hue= 'country', data= df,
palette= 'rainbow')
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

Out[44]: <matplotlib.legend.Legend at 0x1d7ac9b5730>



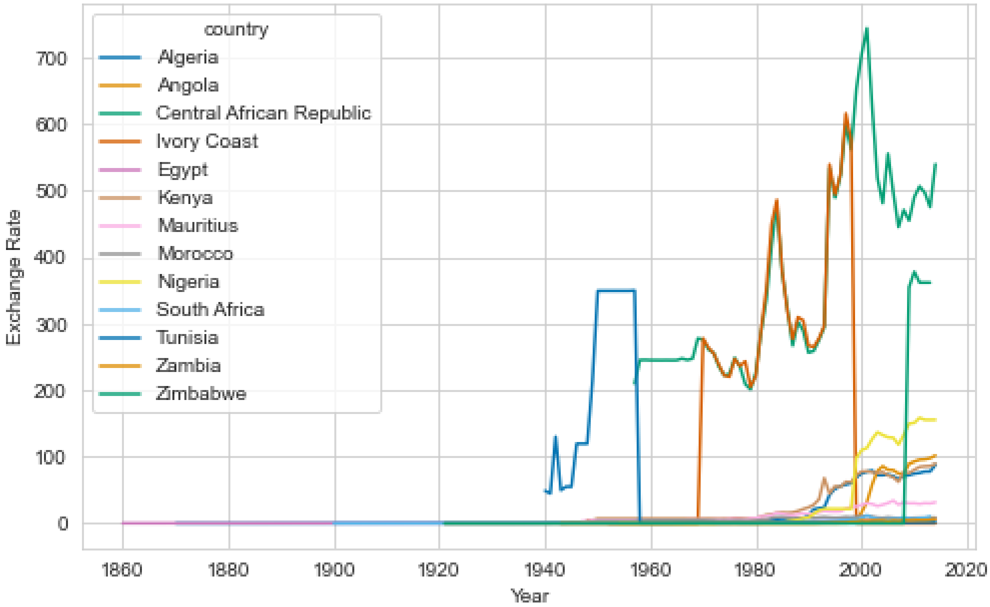
In [43]:
```
sns.countplot(x='inflation_annual_cpi',data=df, color='blue',
saturation=0.88)
```

Out[43]: <AxesSubplot:xlabel='inflation_annual_cpi', ylabel='count'>



In [54]:
```
#Some countries have relatively lower exchange rate than other countries.
Countries like South Africa, Zambia, Egypt and Morocco has relatively
lower exchange rate (It is hard to interpret with the above graph, Let's
break it down the exchange rate for each country in the next graph)
#The exchange rate is almost zero for all the countries before 1940. This
might be because the value is not recorded or a new currency had been
adopted by the countries. (Further analysis required)
#There are tremendous spikes in the exchange rate Angola and Zimbabwe.
```

```
sns.set_style('whitegrid')
plt.figure(figsize = (8,5))

sns.lineplot(
    x = 'year', y = 'exch_usd',
    hue = 'country',
    data = df, palette = 'colorblind'
)

plt.xlabel('Year')
plt.ylabel('Exchange Rate')
display()
```



In [ ]:
```
# given below is for my reference only:
```

In [20]:
```
#sns.catplot(data=df, x= "inflation_annual_cpi",y="inflation_crises",
hue= "country",kind="box")
```

In [ ]:
```
#this gives the average exchange rate value, Mininimum excgange rate
value and Maximum exchange rate value for each country.

#df = df.groupby('country').exch_usd.agg(['count','mean', 'max', 'min'])
#df

#This give the different categories of "case" attribute
#df['case'].unique()
```

```
#This give all details about number of rows and columns registered in case "n"

#df[df.case==10].shape
```

In [ ]:

```
```

In [ ]:

```
```