

# How has the Arab Spring Affected Different Economies across the MENA Region?

Principal Investigators: Mohannad Alramlawi - Zeyad Dahab

Emails: ma4402@nyu.edu - zmd221@nyu.edu

The Arab spring has caused a lot of instability within the MENA region. This is expected, considering that the series of revolts led to changes in the political, economic, and social dynamics of the countries that were directly affected by the Arab spring, and even those that were not. This project will focus on the economic effects of the Arab Spring on different countries in the MENA region. In order to study those effects, several variables will be utilized, including GDP, Total Investment, Inflation, Volume of Exports, Unemployment Rate, and Foreign Debt. Those variables will be analyzed between the years 2004 and 2018, paying special attention to the year 2011 (when the Arab spring started).

First, a series of graphs -each with the variable "year" on the x-axis and the different economic indicators listed above on the y-axis- will be generated for the region as a whole. This would give us a general overview of the entire region. Then, the same graphs will be generated, only this time each country will be explored separately. We will first start with the countries that experienced revolts, including Egypt, Tunisia, and Libya. Afterwards, we will look at the countries that were indirectly affected by the Arab Spring, including Saudi Arabia, Jordan, and Lebanon.

Different methods of visualization will be used throughout the project, including graphing and mapping wherever possible.

## Importing data

This section imports the data and assigns it to a pandas dataframe.

```
In [120]: import pandas as pd  
import matplotlib.pyplot as plt  
import numpy
```

```
In [121]: url = "/Users/Ramlawi/Desktop/DB_DATA.xlsx"
```

```
In [122]: data = pd.read_excel(url)
```

```
In [123]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 168 entries, 0 to 167
Data columns (total 21 columns):
Country           168 non-null object
Subject Descriptor 168 non-null object
Units             168 non-null object
Scale              21 non-null object
Country/Series-specific Notes 139 non-null object
2004               139 non-null object
2005               139 non-null object
2006               139 non-null object
2007               139 non-null object
2008               139 non-null object
2009               139 non-null object
2010               139 non-null object
2011               139 non-null object
2012               139 non-null object
2013               139 non-null object
2014               139 non-null object
2015               139 non-null object
```

In [124]: `data.head(5)`

Out[124]:

	Country	Subject Descriptor	Units	Scale	Country/Series-specific Notes	2004	2005	2006	2007	2008
0	Algeria	Gross domestic product, constant prices	Percent change	NaN	See notes for: Gross domestic product, consta...	4.3	5.9	1.684	3.1	3.1
1	Algeria	Gross domestic product per capita, constant pr...	Purchasing power parity; 2011 international do...	Units	See notes for: Gross domestic product, consta...	11977.7	12495.6	12517.1	12517.1	12517.1
2	Algeria	Total investment	Percent of GDP	NaN	Source: National Statistics Office Latest actu...	33.305	31.334	29.837	32.2	32.2
3	Algeria	Gross national savings	Percent of GDP	NaN	Source: National Statistics Office Latest actu...	46.333	51.861	54.575	56.5	56.5
4	Algeria	Inflation, average consumer prices	Index	NaN	Source: National Statistics Office Latest actu...	109.95	111.47	114.05	114.05	114.05

5 rows × 21 columns

```
In [125]: data.tail(5)
```

Out[125]:

	Country	Subject Descriptor	Units	Scale	Country/Series-specific Notes	2004	2005	2006
163	Yemen	Gross national savings	Percent of GDP	NaN	Source: IMF Staff Estimates Latest actual data...	21.901	22.311	17.598
164	Yemen	Inflation, average consumer prices	Index	NaN	Source: National Statistics Office. Central Ba...	156.607	172.059	190.719
165	Yemen	Volume of exports of goods and services	Percent change	NaN	Base year: 2005 Primary domestic currency: Yem...	-9.004	-2.029	-3.882
166	Yemen	Unemployment rate	Percent of total labor force	NaN	NaN	NaN	NaN	NaN
167	Yemen	General government net debt	Percent of GDP	NaN	See notes for: General government net debt (N...	45.466	37.065	32.999

5 rows × 21 columns

## Cleaning data

As it can be seen from the tables above, there are some cells and entire rows with NaN values. This is because some countries have missing data.

In this section, we deal with this problem by eliminating the indicators that are not reported for most countries. We also eliminate the data for Syria and Somalia, since both countries have a lot of missing values.

```
In [126]: data.rename(columns={'Subject Descriptor':'Subject'}, inplace=True)
```

```
In [127]: data.drop( data[ data['Subject'] == "Unemployment rate" ].index , inplace=True)
```

```
In [128]: data.drop( data[ data['Subject'] == "General government net debt" ].index , inplace=True)
```

```
In [129]: data.drop( data[ data['Subject'] == "Total investment" ].index , inplace=True)
```

```
In [130]: data.drop( data[ data['Country'] == "Syria" ].index , inplace=True)
```

```
In [131]: data.drop( data[ data['Country'] == "Somalia" ].index , inplace=True)
```

```
In [132]: data.head(5)
```

Out[132]:

	Country	Subject	Units	Scale	Country/Series-specific Notes	2004	2005	2006	2007
0	Algeria	Gross domestic product, constant prices	Percent change	NaN	See notes for: Gross domestic product, consta...	4.3	5.9	1.684	3.373
1	Algeria	Gross domestic product per capita, constant pr...	Purchasing power parity; 2011 international do...	Units	See notes for: Gross domestic product, consta...	11977.7	12495.6	12517.1	1274!

## Visualizations for the entire Middle East region

We start off the project by looking at the Middle East region as a whole.

We do not have values for the indicators of the entire region. Because that is the case, we will calculate those values by aggregating the values for each indicator over all the countries in the dataset.

## Aggregating the indicators

We aggregate the indicators by calculating the mean of each indicator for every year over the

different countries we have.

This is done below.

We start by creating a different dataframe for each indicator:

```
In [133]: gdp = data[data['Subject'] == "Gross domestic product, constant prices"]
```

```
In [134]: gdp_capita = data[data['Subject'] == "Gross domestic product per capita,
```

```
In [135]: gns = data[data['Subject'] == "Gross national savings"]
```

```
In [136]: gns.drop( gns[ gns['Country'] == "Libya" ].index , inplace=True)
```

```
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/_main_.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
```

```
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
```

```
if __name__ == '__main__':
```

Libya is dropped from the gross national savings dataframe because it does not have data for this indicator.

```
In [137]: inf = data[data['Subject'] == "Inflation, average consumer prices"]
```

```
In [138]: exp = data[data['Subject'] == "Volume of exports of goods and services"]
```

```
In [139]: exp.drop( exp[ exp['Country'] == "Iraq" ].index , inplace=True)
```

```
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/_main_.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
```

```
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
```

```
if __name__ == '__main__':
```

Iraq is dropped from the exports dataframe because it does not have data for this indicator.

In this next step, we create one dataframe that has the aggregated values for each indicator over time.

```
In [140]: ME_df = pd.DataFrame({2004: [gdp.loc[:, 2004].mean(), gdp_capita.loc[:, 2004].mean()], 2005: [gdp.loc[:, 2005].mean(), gdp_capita.loc[:, 2005].mean()], 2006: [gdp.loc[:, 2006].mean(), gdp_capita.loc[:, 2006].mean()], 2007: [gdp.loc[:, 2007].mean(), gdp_capita.loc[:, 2007].mean()], 2008: [gdp.loc[:, 2008].mean(), gdp_capita.loc[:, 2008].mean()], 2009: [gdp.loc[:, 2009].mean(), gdp_capita.loc[:, 2009].mean()], 2010: [gdp.loc[:, 2010].mean(), gdp_capita.loc[:, 2010].mean()], 2011: [gdp.loc[:, 2011].mean(), gdp_capita.loc[:, 2011].mean()], 2012: [gdp.loc[:, 2012].mean(), gdp_capita.loc[:, 2012].mean()], 2013: [gdp.loc[:, 2013].mean(), gdp_capita.loc[:, 2013].mean()], 2014: [gdp.loc[:, 2014].mean(), gdp_capita.loc[:, 2014].mean()], 2015: [gdp.loc[:, 2015].mean(), gdp_capita.loc[:, 2015].mean()]})
```

```
In [141]: years = [2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015]
```

```
In [142]: x=1
for number in years:
    ME_df.insert(x, number, [gdp.loc[:, number].mean(), gdp_capita.loc[:, number].mean()])
    x = x + 1
```

```
In [143]: ME_df['Economic Indicator'] = ["GDP", "GDP/capita", "GNS", "Inflation", "Exports"]
```

This is the dataframe we get:

```
In [144]: ME_df.set_index('Economic Indicator')
```

	2004	2005	2006	2007	2008	2009
Economic Indicator						
<b>GDP</b>	8.956316	5.617789	7.284684	5.889632	5.547895	1.9
<b>GDP/capita</b>	29064.483579	28820.365000	29004.904211	28390.535421	27142.672368	26800.0
<b>GNS</b>	27.259056	30.848389	32.693278	32.809389	34.328778	27.0
<b>Inflation</b>	88.859789	93.759947	100.049105	108.212368	119.229842	120.0
<b>Exports</b>	9.837944	6.791500	6.281056	6.788944	5.643944	-4.0

## Plotting the aggregated values

Now that we have the dataframe with all the aggregated values, all we have to do is to do is plot them.

This is done below:

```
In [145]: %%javascript
IPython.OutputArea.prototype._should_scroll = function(lines) {
    return false;}
```

```
In [146]: def regionalplot(indicator):
    fig, ax = plt.subplots(figsize = (14,8))

    years_1 = [2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013

    d = {"GDP": 0, "GDP/capita": 1, "GNS": 2, "Inflation": 3, "Exports": 4

    for element in d:
        if element == indicator:
            m = d[element]

    a = ME_df.loc[m]
    a = a.tolist()
    del a[-1]

    ax.plot(years_1, a, color = "orange", linewidth = 3)

    ax.set_title("%s over time (for the entire region)"%indicator, loc ='center')
    ax.set_xlabel("Year", fontsize = 20, fontname = "Arial", color = "navy")
    if indicator == "GDP" or indicator == "Exports":
        ax.set_ylabel("%s (percent change)"%indicator, fontsize = 20, fontname = "Arial", color = "navy")
    elif indicator == "GDP/capita":
        ax.set_ylabel("%s (2011 international dollars)"%indicator, fontsize = 20, fontname = "Arial", color = "navy")
    elif indicator == "GNS":
        ax.set_ylabel("%s (as a percentage of GDP)"%indicator, fontsize = 20, fontname = "Arial", color = "navy")
    elif indicator == "Inflation":
        ax.set_ylabel("%s (index)"%indicator, fontsize = 20, fontname = "Arial", color = "navy")

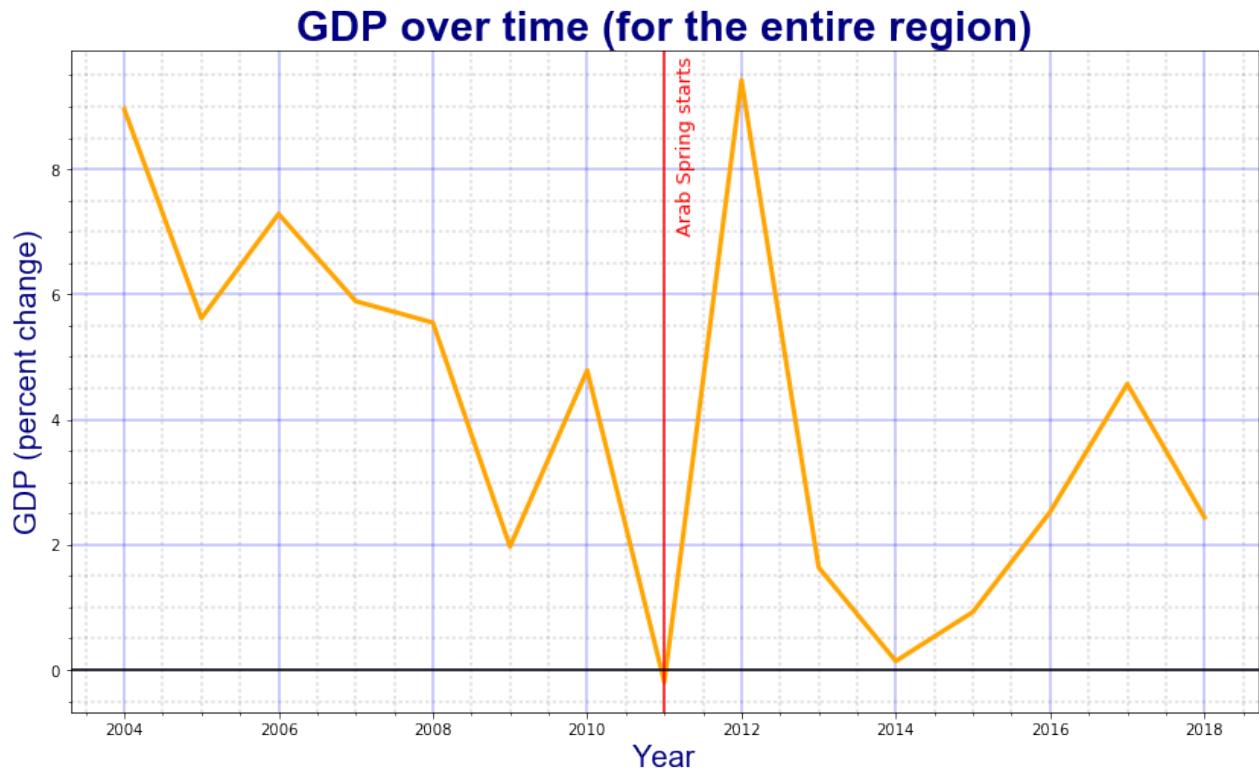
    ax.grid(which='major', linestyle='-', linewidth='0.4', color='blue')
    ax.minorticks_on()
    ax.grid(which='minor', linestyle=':', linewidth='0.2', color='black')

    plt.axvline(x=2011, color='red', linestyle='--')
    plt.text(0.51, 0.97, 'Arab Spring starts', transform=ax.transAxes, rotation=0)

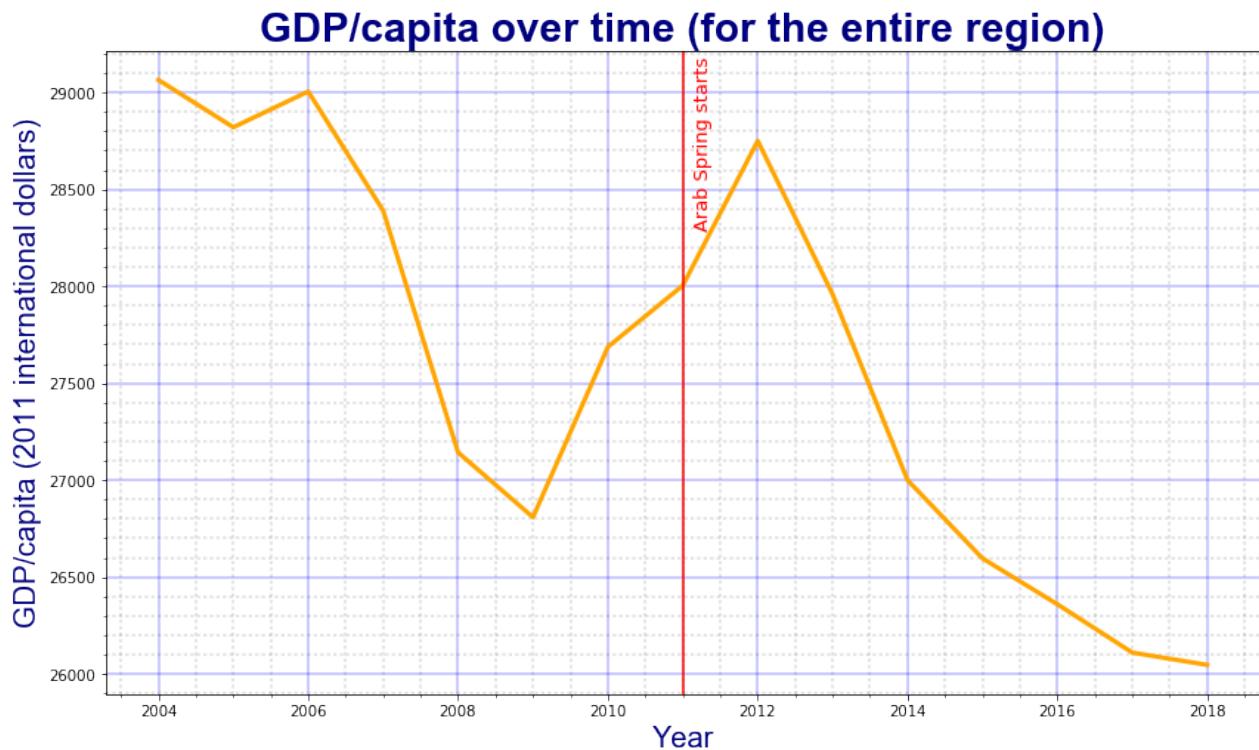
    if indicator == "GDP" or indicator == "Exports":
        plt.axhline(y=0, color='black', linestyle='--')

    plt.show()
```

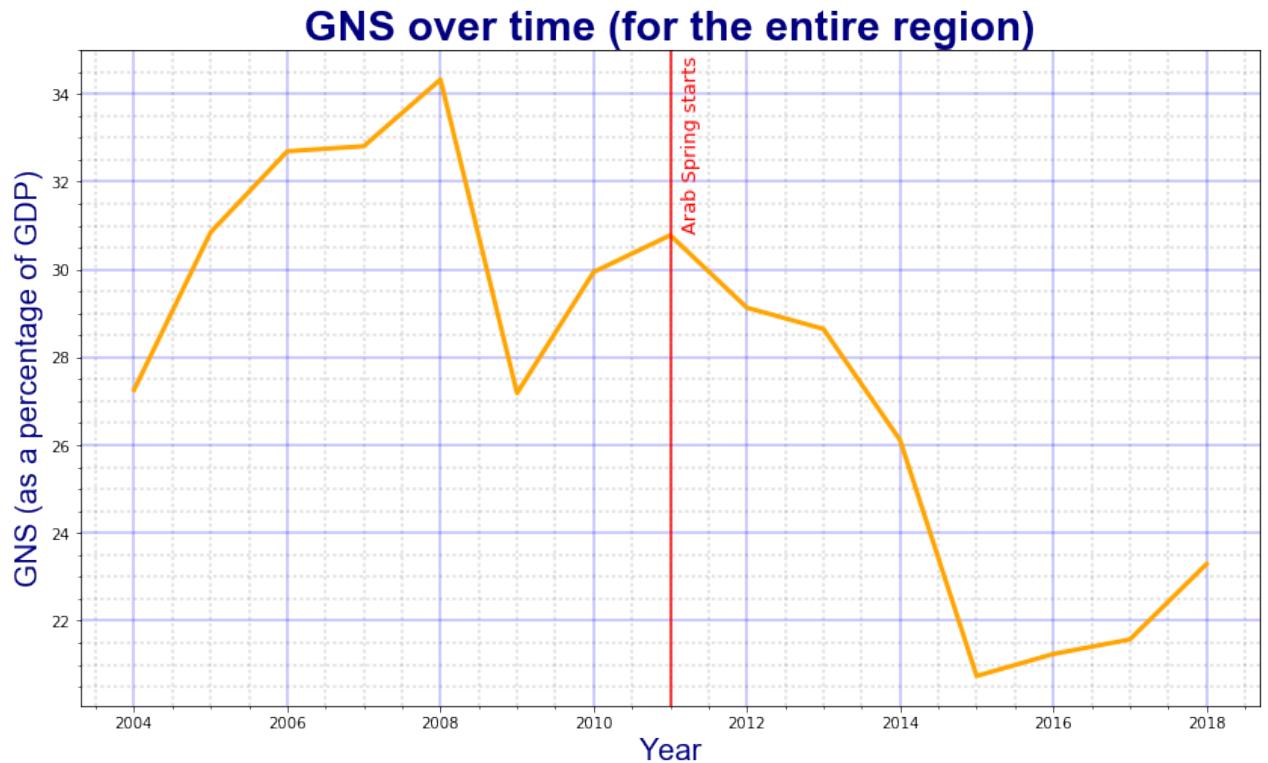
```
In [147]: regionalplot("GDP")
```



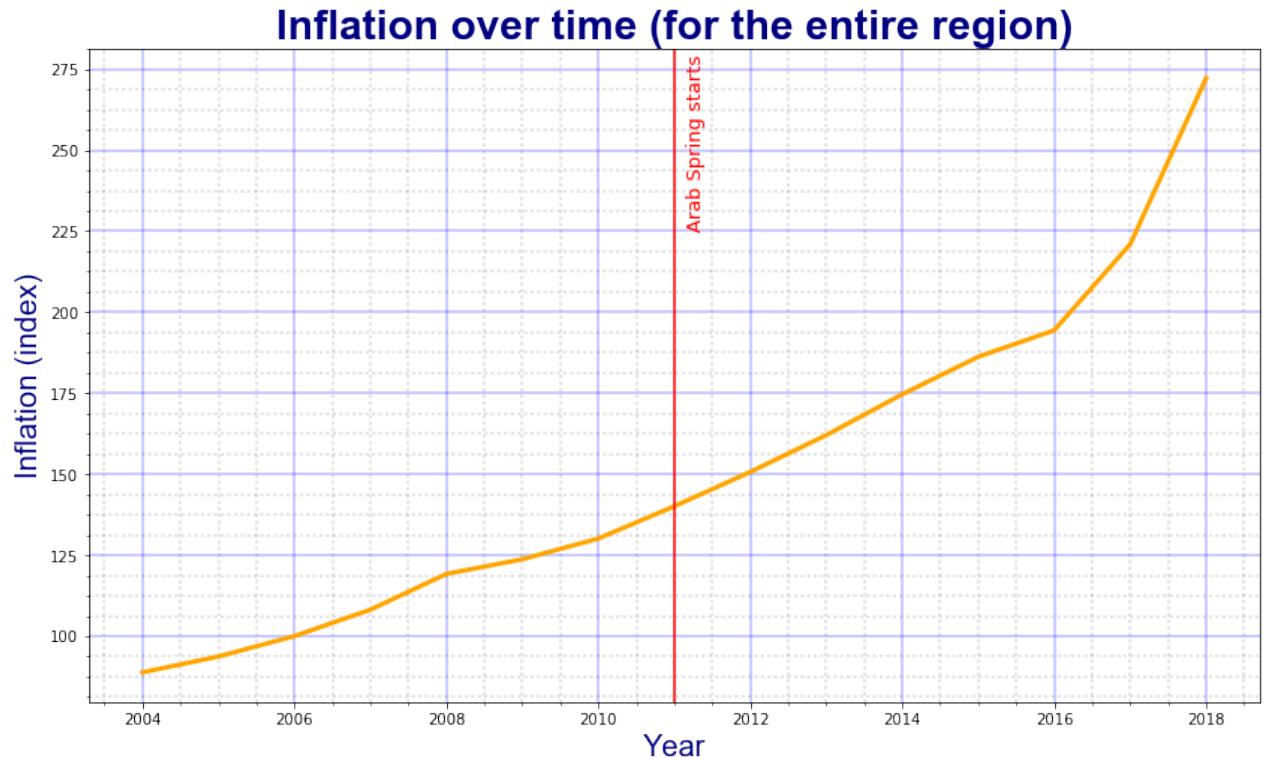
```
In [148]: regionalplot("GDP/capita")
```



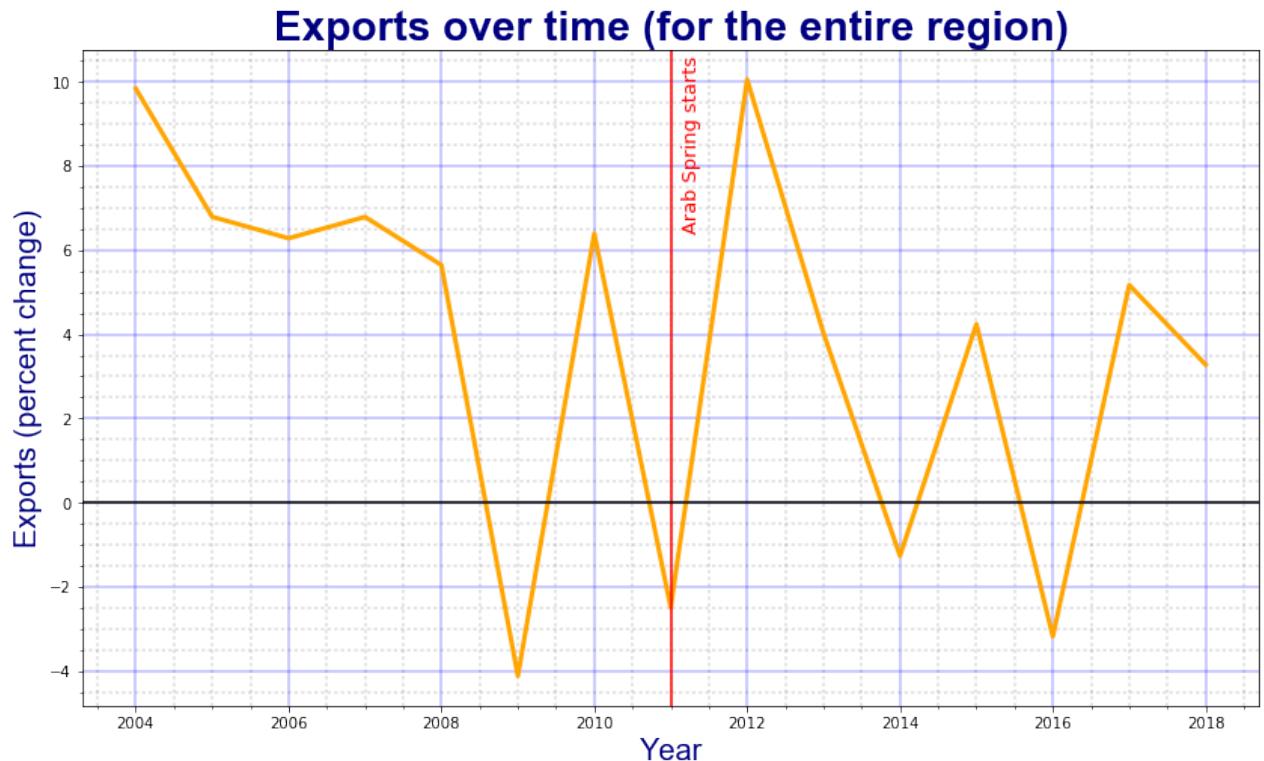
```
In [149]: regionalplot("GNS")
```



```
In [150]: regionalplot("Inflation")
```



```
In [151]: regionalplot("Exports")
```



Type *Markdown* and *LaTeX*:  $\alpha^2$

### Comments on the graphs:

GDP: Interestingly, it seems that the percentage change in GDP is almost always positive in the region except in 2011, the year when the Arab Spring started.

GDP/capita: The GDP/capita faces a decrease after 2012. There is a fall of almost 3000 dollars from 2004 till 2018.

GNS: It is striking to see that GNS starts making up a lower and lower percentage of GDP after the Arab Spring started in 2011.

Inflation: Inflation level is constantly increasing. Nothing is particularly special about the year 2011.

Exports: Constant fluctuation in the percentage change of exports. Nothing is particularly special about the year 2011.

## Mapping

As it can be seen, the graphs above are very interesting. They also give us valuable information. One problem however is that those graphs are for the MENA region as a whole. This is problematic, as countries in the Middle East are very diverse.

In this section, we aim to generate a map of the Middle East, and color code each country based on the indicators we have. The aim of the mapping is to show how different each indicator will be for each country, and therefore prove the necessity to make separate graphs for each country.

Mapping comprises of two main parts. First, for each country, we have to calculate the average of each indicator across years. This is important as mapping does not show changes over time like graphs do. We then use that data to generate the maps.

```
In [152]: from IPython.display import display, Image
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import os

!pip install census
from census import Census

!pip install geopandas
import geopandas as gpd
from shapely.geometry import Point, Polygon

!pip install descartes
from descartes import PolygonPatch

from mpl_toolkits.axes_grid1.inset_locator import zoomed_inset_axes
from mpl_toolkits.axes_grid1.inset_locator import mark_inset
```

Requirement already satisfied: census in /Users/Ramlawi/anaconda/lib/python3.6/site-packages

Requirement already satisfied: requests>=1.1.0 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from census)

Requirement already satisfied: future in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from census)

You are using pip version 9.0.1, however version 19.1.1 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.

Requirement already satisfied: geopandas in /Users/Ramlawi/anaconda/lib/python3.6/site-packages

Requirement already satisfied: fiona in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from geopandas)

Requirement already satisfied: shapely in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from geopandas)

Requirement already satisfied: pyproj in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from geopandas)

```
Requirement already satisfied: pandas in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from geopandas)
Requirement already satisfied: cligj>=0.5 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from fiona->geopandas)
Requirement already satisfied: click-plugins>=1.0 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from fiona->geopandas)
Requirement already satisfied: click<8,>=4.0 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from fiona->geopandas)
Requirement already satisfied: six>=1.7 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from fiona->geopandas)
Requirement already satisfied: munch in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from fiona->geopandas)
Requirement already satisfied: attrs>=17 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from fiona->geopandas)
Requirement already satisfied: python-dateutil>=2 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from pandas->geopandas)
Requirement already satisfied: pytz>=2011k in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from pandas->geopandas)
Requirement already satisfied: numpy>=1.7.0 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from pandas->geopandas)
You are using pip version 9.0.1, however version 19.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

Requirement already satisfied: descartes in /Users/Ramlawi/anaconda/lib/python3.6/site-packages
Requirement already satisfied: matplotlib in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from descartes)
Requirement already satisfied: numpy>=1.7.1 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from matplotlib->descartes)
Requirement already satisfied: six>=1.10 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from matplotlib->descartes)
Requirement already satisfied: python-dateutil in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from matplotlib->descartes)
Requirement already satisfied: pytz in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from matplotlib->descartes)
Requirement already satisfied: cycler>=0.10 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from matplotlib->descartes)
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=1.5.6 in /Users/Ramlawi/anaconda/lib/python3.6/site-packages (from matplotlib->descartes)
You are using pip version 9.0.1, however version 19.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

```
In [153]: world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))  
world.head()
```

Out[153]:

	pop_est	continent	name	iso_a3	gdp_md_est	geometry
0	920938	Oceania	Fiji	FJI	8374.0	(POLYGON ((180 -16.06713266364245, 180 -16.555...))
1	53950935	Africa	Tanzania	TZA	150600.0	POLYGON ((33.90371119710453 -0.9500000000000000...))
2	603253	Africa	W. Sahara	ESH	906.5	POLYGON ((-8.665589565454809 27.65642588959236...))
3	35623680	North America	Canada	CAN	1674000.0	(POLYGON ((-122.84 49.00000000000011, -122.974...)))
4	326625791	North America	United States of America	USA	18560000.0	(POLYGON ((-122.84 49.00000000000011, -120 49....)))

```
In [154]: fig, gax = plt.subplots(figsize=(16,20))

countries = data.Country.unique()
countries = countries.tolist()

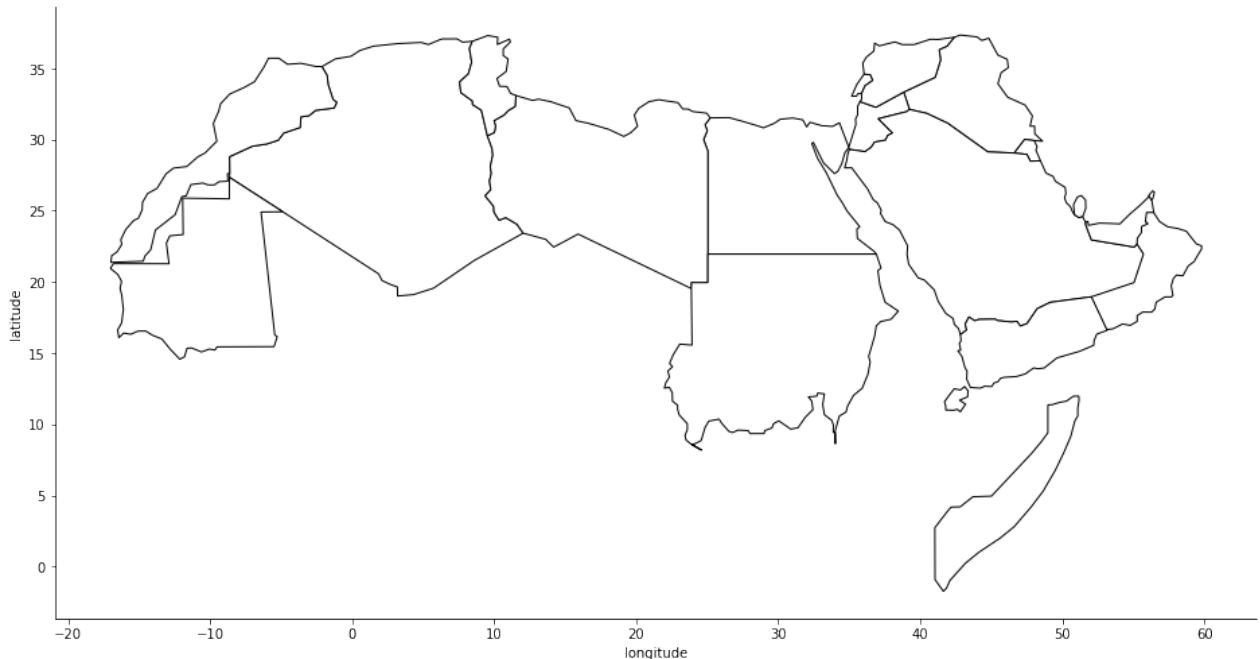
for item in countries:
    world[world['name'] == item].plot(ax = gax, edgecolor='black',color='white')
    world[world['name'] == 'Syria'].plot(ax = gax, edgecolor='black',color='white')
    world[world['name'] == 'Somalia'].plot(ax = gax, edgecolor='black',color='white')

gax.set_xlabel('longitude')
gax.set_ylabel('latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()

/Users/Ramlawi/anaconda/lib/python3.6/site-packages/geopandas/plotting
.py:418: UserWarning: The GeoDataFrame you are attempting to plot is empty. Nothing has been displayed.
"empty. Nothing has been displayed.", UserWarning)
```



**Calculating the average across years for each indicator, for each country**

```
In [155]: gdp['Average GDP'] = (gdp[2004] + gdp[2005] + gdp[2006] + gdp[2007] + gdp  
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/__main__.py: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: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>  
(<http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>)

```
if __name__ == '__main__':
```

```
In [156]: gdp_capita['Average GDP/capita'] = (gdp_capita[2004] + gdp_capita[2005] +  
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/__main__.py: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: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>  
(<http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>)

```
if __name__ == '__main__':
```

```
In [157]: gns['Average GNS'] = (gns[2004] + gns[2005] + gns[2006] + gns[2007] + gns  
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/__main__.py: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: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>  
(<http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>)

```
if __name__ == '__main__':
```

```
In [158]: inf['Average Inflation'] = (inf[2004] + inf[2005] + inf[2006] + inf[2007] +  
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/__main__.py: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: http://pandas.pydata.org/pandas-  
docs/stable/indexing.html#indexing-view-versus-copy  
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-vi  
ew-versus-copy)  
if __name__ == '__main__':
```

```
In [159]: exp['Average Exports'] = (exp[2004] + exp[2005] + exp[2006] + exp[2007] +  
/Users/Ramlawi/anaconda/lib/python3.6/site-packages/ipykernel/__main__.py: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: http://pandas.pydata.org/pandas-  
docs/stable/indexing.html#indexing-view-versus-copy  
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-vi  
ew-versus-copy)  
if __name__ == '__main__':
```

```
In [160]: world = world.set_index('name')
```

```
In [161]: rows = world.loc[countries]
```

```
In [162]: gdp = gdp.set_index('Country')
```

```
In [163]: gdp_capita = gdp_capita.set_index('Country')  
gns = gns.set_index('Country')  
inf = inf.set_index('Country')  
exp = exp.set_index('Country')
```

```
In [164]: rows['Average GDP'] = gdp['Average GDP']  
rows['Average GDP/capita'] = gdp_capita['Average GDP/capita']  
rows['Average GNS'] = gns['Average GNS']  
rows['Average Inflation'] = inf['Average Inflation']  
rows['Average Exports'] = exp['Average Exports']
```

In [165]: `rows.head(3)`

Out[165]:

	<code>pop_est</code>	<code>continent</code>	<code>iso_a3</code>	<code>gdp_md_est</code>	<code>geometry</code>	<code>Average GDP</code>	<code>Ave GD</code>
<code>name</code>							
<b>Algeria</b>	40969443.0	Africa	DZA	609400.0	POLYGON ((-8.684399786809053 27.39574412689601...))	3.0692	130
<b>Bahrain</b>	NaN	NaN	NaN	NaN	NaN	4.60527	433
<b>Djibouti</b>	865267.0	Africa	DJI	3345.0	POLYGON ((42.351560000000012 12.542230000000013,...))	5.12247	276

## Generating the maps

In [166]: `def mapp(indc):`

```

fig, gax = plt.subplots(figsize=(16,20))

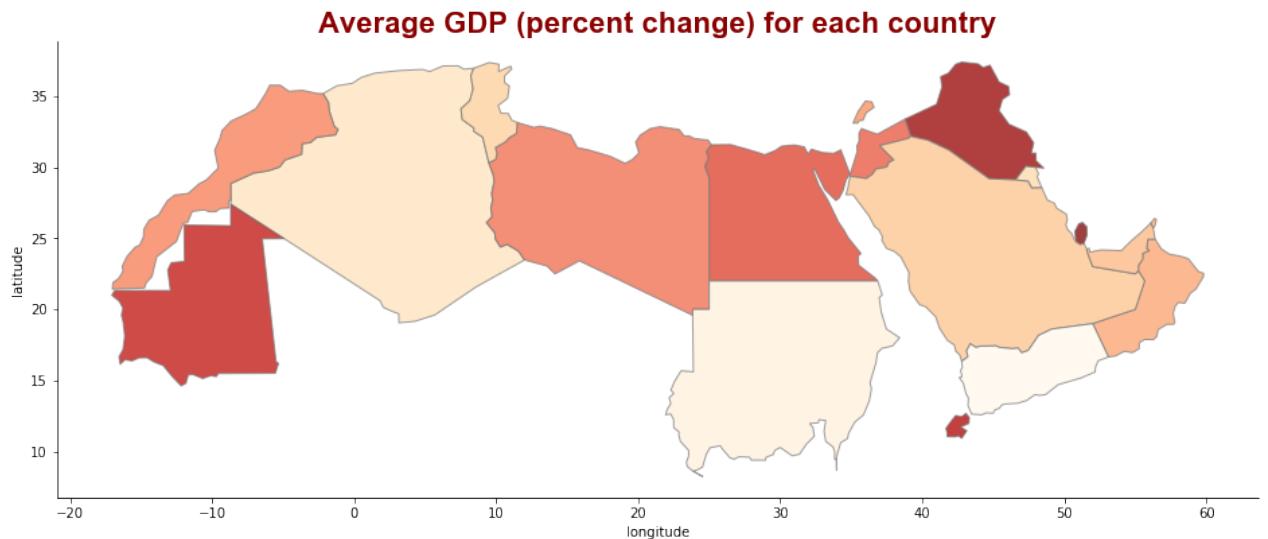
rows.plot(ax = gax, edgecolor='tab:grey', column= indc, cmap='OrRd',
          gax.set_xlabel('longitude')
          gax.set_ylabel('latitude')

          gax.spines['top'].set_visible(False)
          gax.spines['right'].set_visible(False)

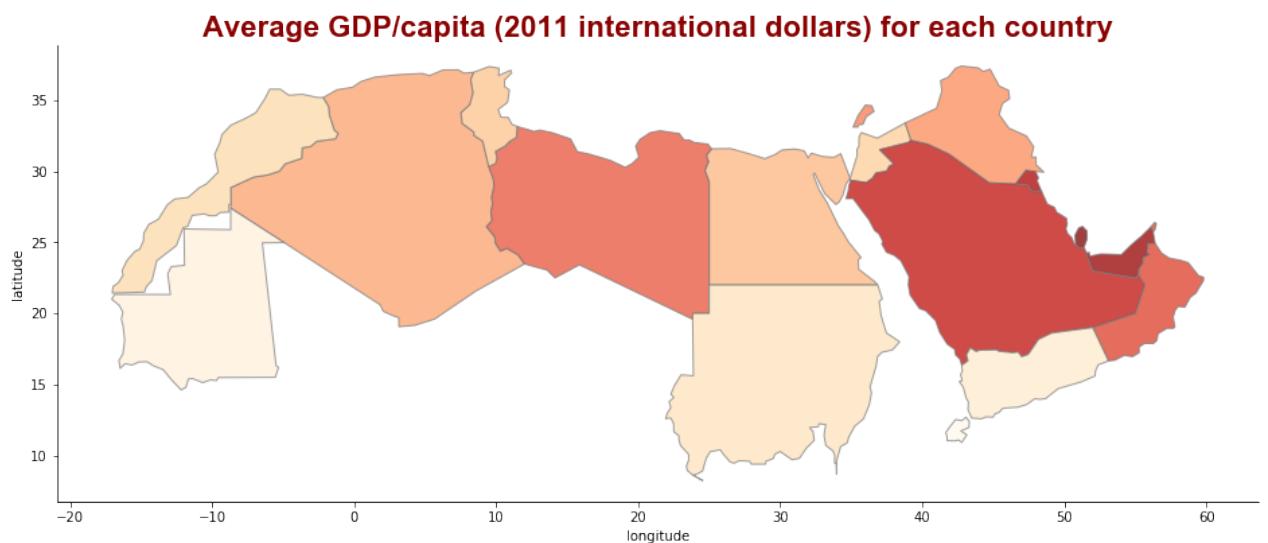
          if indc == "Average GDP" or indc == "Average Exports":
              gax.set_title("%s (percent change) for each country"%indc, loc ='center')
          elif indc == "Average GDP/capita":
              gax.set_title("%s (2011 international dollars) for each country"%indc, loc ='center')
          elif indc == "Average GNS":
              gax.set_title("%s (as a percentage of GDP) for each country"%indc, loc ='center')
          elif indc == "Average Inflation":
              gax.set_title("%s (index) for each country"%indc, loc ='center',)

          plt.show()
    
```

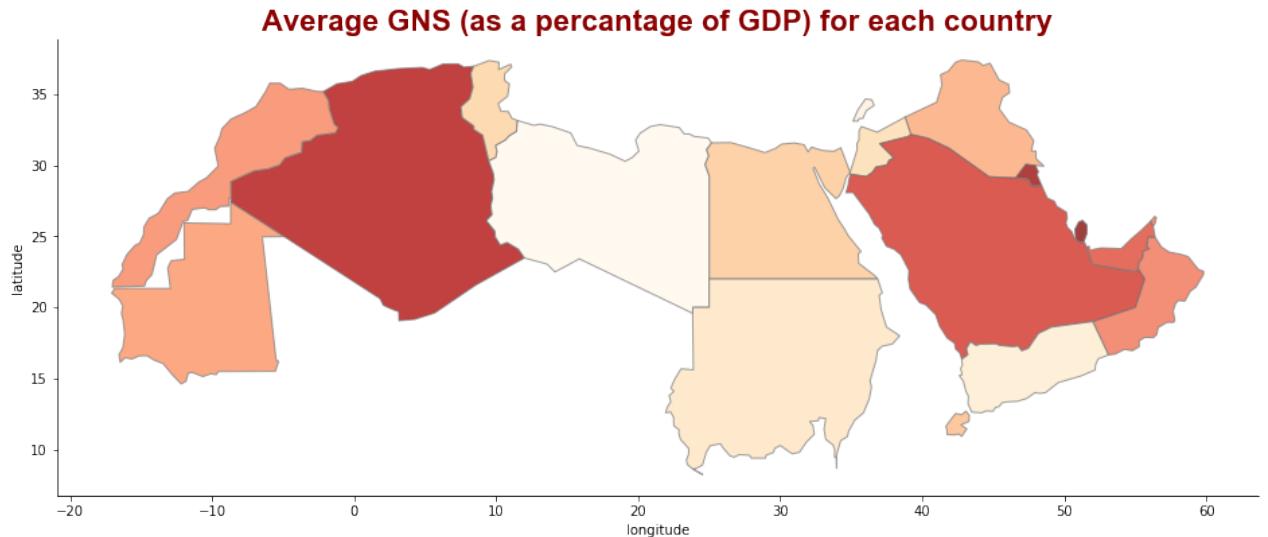
```
In [167]: mapp("Average GDP")
```



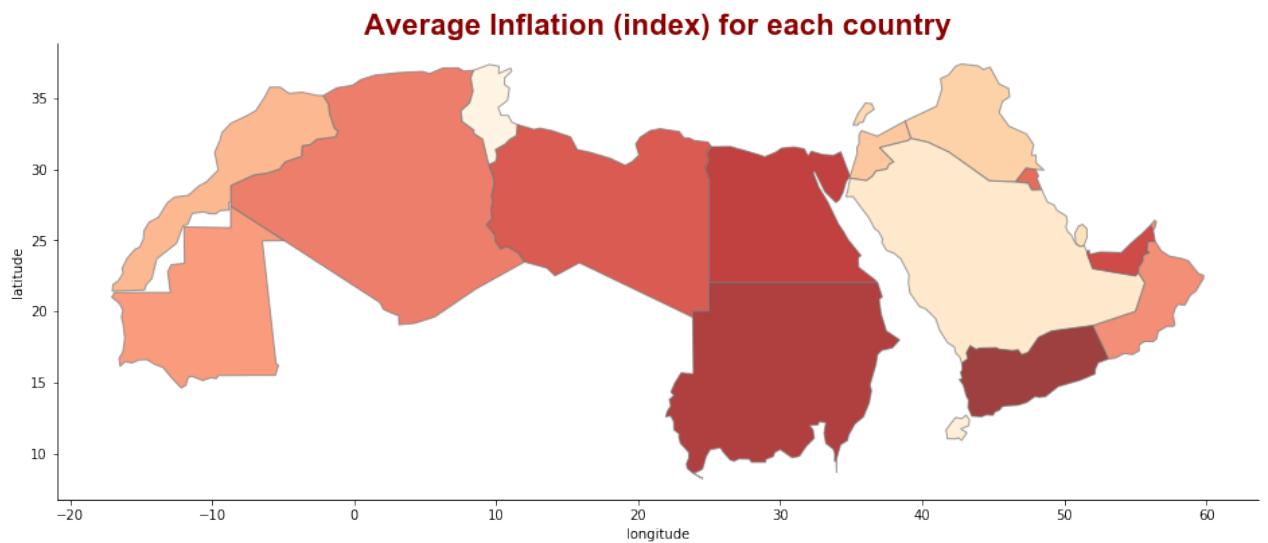
```
In [168]: mapp("Average GDP/capita")
```



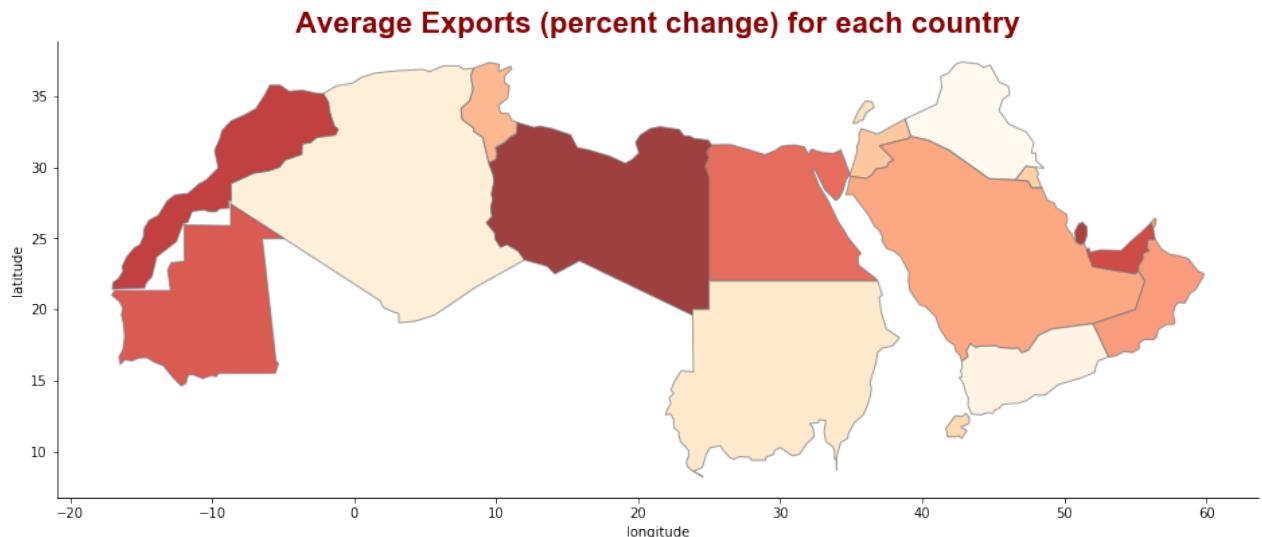
```
In [169]: mapp("Average GNS")
```



```
In [170]: mapp("Average Inflation")
```



```
In [171]: mapp("Average Exports")
```



As the maps above show, there is a lot of variation in the indicators across different countries. This is why in the following section, we will create separate graphs for each country in the MENA region.

## Graphing the indicators for each country

```
In [172]: a = data.Subject.unique()
a = a.tolist()
a
```

```
Out[172]: ['Gross domestic product, constant prices',
 'Gross domestic product per capita, constant prices',
 'Gross national savings',
 'Inflation, average consumer prices',
 'Volume of exports of goods and services']
```

```
In [173]: def plots(metric,arab_spring):
    if arab_spring == False:
        metric_data = data.drop( data[ data['Subject'] != metric ].index)
        metric_data = metric_data.drop(["Subject", "Estimates Start After"
        for item in ["Egypt", "Tunisia", "Yemen", "Bahrain", "Libya"]:
            metric_data.drop( metric_data[ metric_data['Country'] == item
        metric_data["ind"] = [0,1,2,3,4,5,6,7,8,9,10,11,12,13]
        metric_data = metric_data.set_index("ind")
        fig, ax = plt.subplots(nrows = 7, ncols = 2,figsize = (15,35))
        fig.suptitle("%s for non Arab Spring Countries" %metric, fontsize
        fig.tight_layout()
        fig.subplots_adjust(top=0.95)
        plt.subplots_adjust(hspace=0.5      wspace = 0.2)
```

```
    plt.subplots_adjust( hspace = 0.5 , wspace = 0.2 ,
x = [0,0,1,1,2,2,3,3,4,4,5,5,6,6,7]
y = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]
for item in range(14):
    a = metric_data.loc[item][4:19]
    Country_Metric = a.tolist()
    years = list(metric_data.columns.values[4:19])
    if metric_data["Country"][item] in ["Egypt", "Yemen", "Tunisi
        ax[x[item],y[item]].plot(years, Country_Metric, color =
    else:
        ax[x[item],y[item]].plot(years, Country_Metric, color =
    ax[x[item],y[item]].set_xlabel("Year", fontsize = 12, color =
    ax[x[item],y[item]].set_ylabel("%s"%(metric), fontsize = 12,
    ax[x[item],y[item]].spines["right"].set_visible(False)
    ax[x[item],y[item]].spines["top"].set_visible(False)
    ax[x[item],y[item]].set_title("%s"%(metric_data["Country"])[it
    ax[x[item],y[item]].axvline(x=2011, color = "black")
    ax[x[item],y[item]].text(0.51,0.95,'Arab Spring Starts',trans
    ax[x[item],y[item]].grid(which='major', linestyle='-', linewidth=
    ax[x[item],y[item]].minorticks_on()
    ax[x[item],y[item]].grid(which='minor', linestyle=':', linewidth=
else:
    metric_data = data.drop( data[ data['Subject'] != metric ].index)
    metric_data = metric_data.drop(["Subject", "Estimates Start After
    for item in ["Jordan", "Algeria", "Djibouti", "Islamic Republic o
        metric_data.drop( metric_data[ metric_data['Country'] == item
    metric_data["ind"] = [0,1,2,3,4]
    metric_data = metric_data.set_index("ind")
    fig, ax = plt.subplots(nrows = 3, ncols = 2, figsize = (15,20))
    fig.suptitle("%s for Arab Spring Countries" %metric, fontsize = 2
    fig.tight_layout()
    fig.subplots_adjust(top=0.93)
    plt.subplots_adjust( hspace=0.5 , wspace = 0.2)
x = [0,0,1,1,2]
y = [0,1,0,1,0]

for item in range(5):
    a = metric_data.loc[item][4:19]
    Country_Metric = a.tolist()
    years = list(metric_data.columns.values[4:19])
    if metric_data["Country"][item] in ["Egypt", "Yemen", "Tunisi
        ax[x[item],y[item]].plot(years, Country_Metric, color =
    else:
        ax[x[item],y[item]].plot(years, Country_Metric, color =
    ax[x[item],y[item]].set_xlabel("Year", fontsize = 12, color =
    ax[x[item],y[item]].set_ylabel("%s"%(metric), fontsize = 12,
    ax[x[item],y[item]].spines["right"].set_visible(False)
    ax[x[item],y[item]].spines["top"].set_visible(False)
    ax[x[item],y[item]].set_title("%s"%(metric_data["Country"])[it
    ax[x[item],y[item]].axvline(x=2011, color = "black")
    ax[x[item],y[item]].text(0.51,0.95,'Arab Spring Starts',trans
```

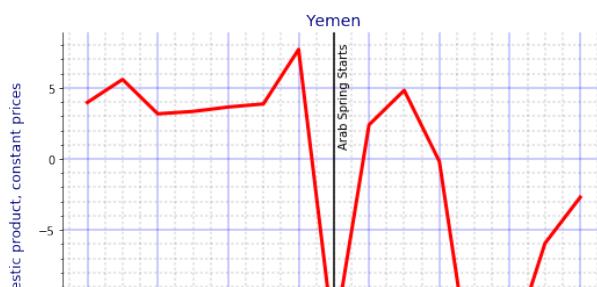
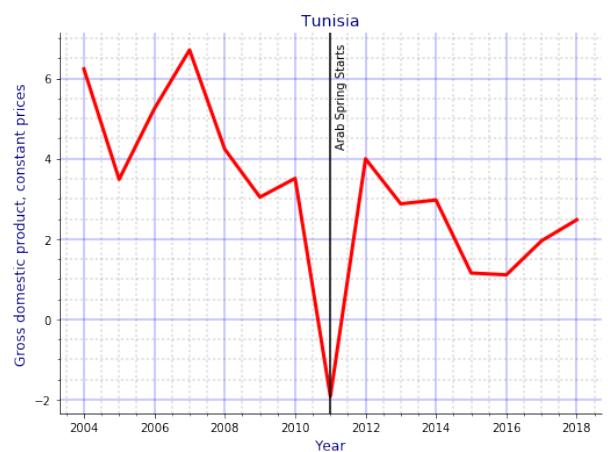
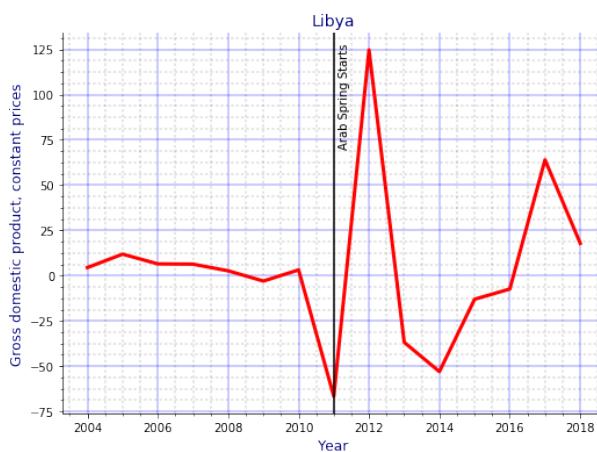
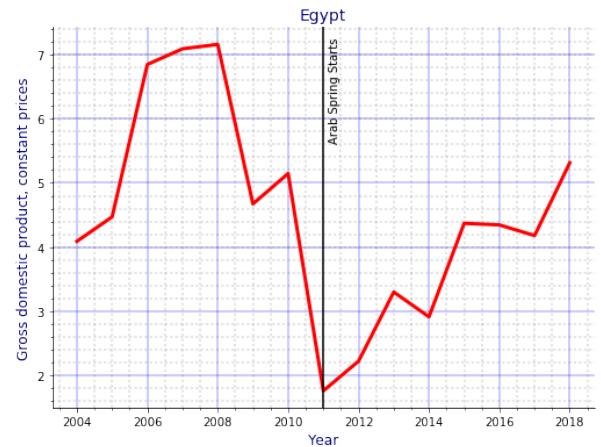
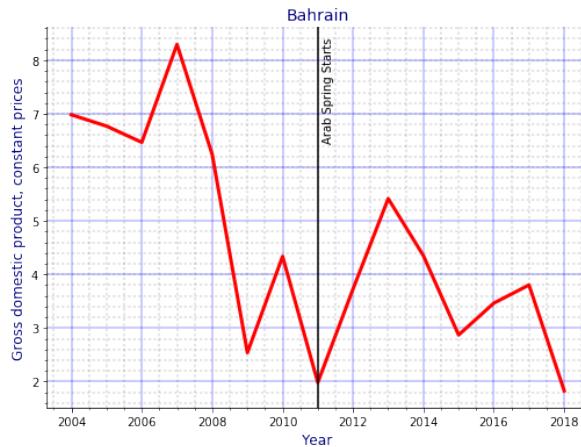
```

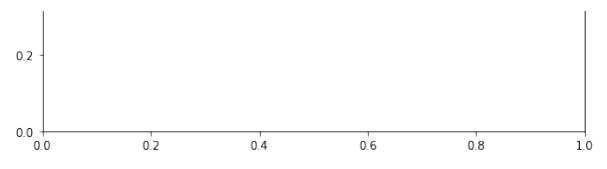
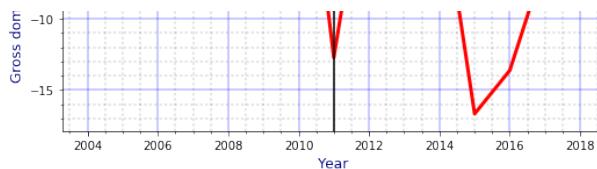
    ax[x[item],y[item]].grid(which='major', linestyle='-', linewidth=1)
    ax[x[item],y[item]].minorticks_on()
    ax[x[item],y[item]].grid(which='minor', linestyle=':', linewidth=1)
plt.show()

```

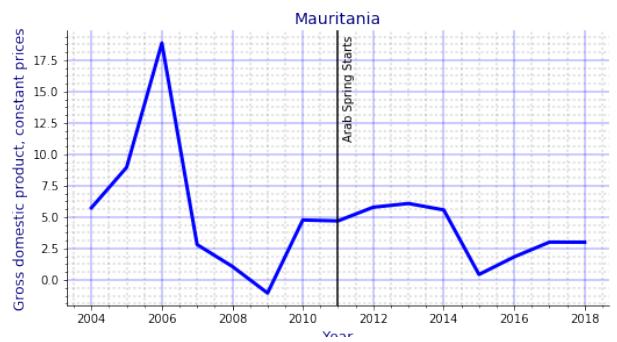
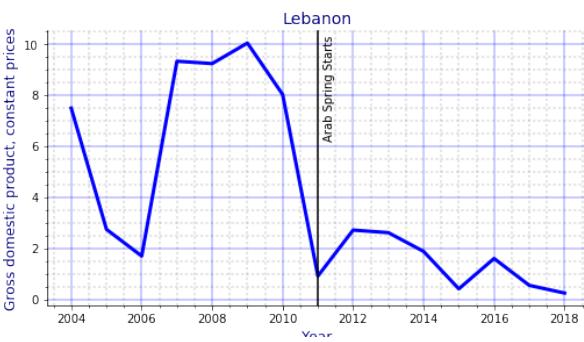
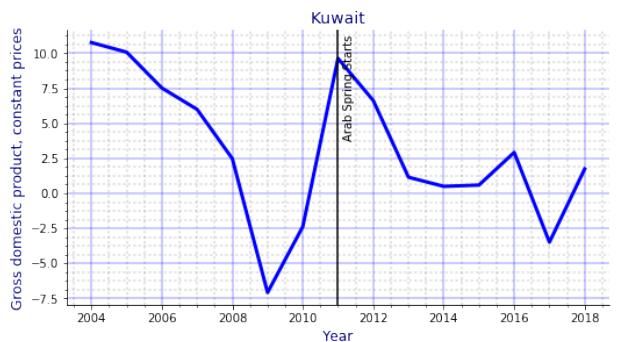
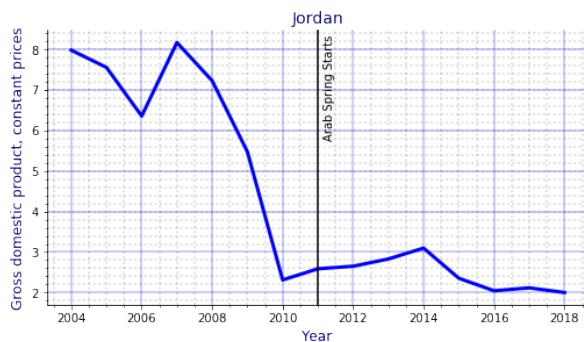
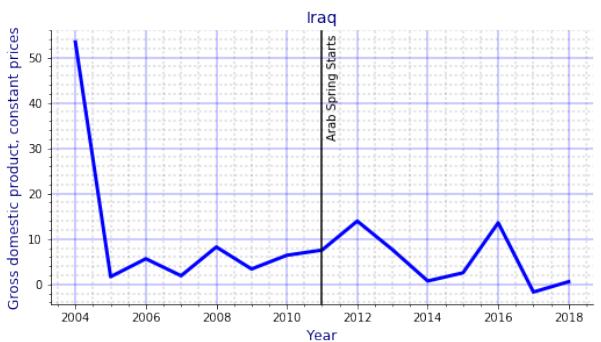
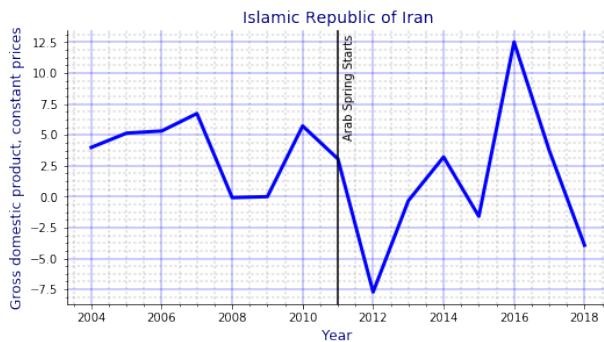
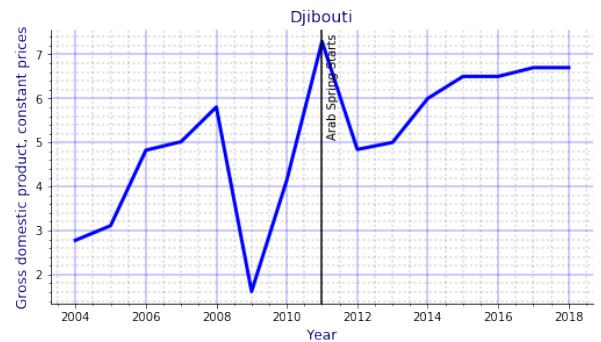
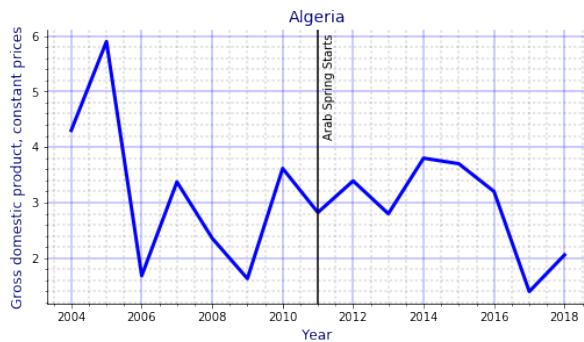
```
In [174]: for item in [True, False]:
    plots(a[0], item)
plt.show()
```

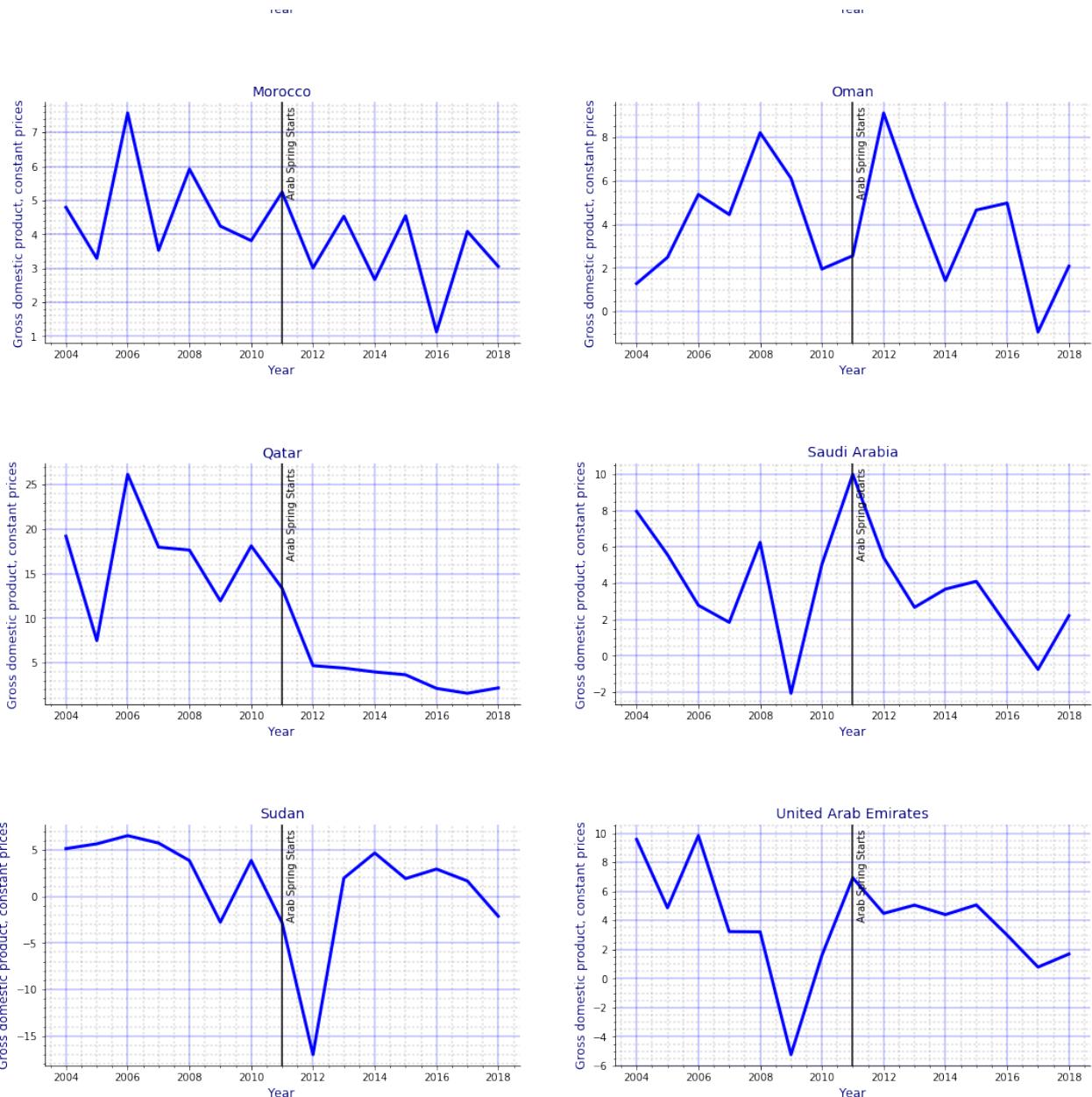
### Gross domestic product, constant prices for Arab Spring Countries





### Gross domestic product, constant prices for non Arab Spring Countries

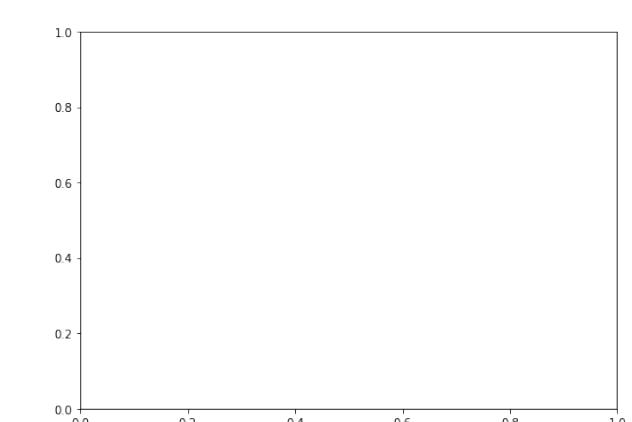
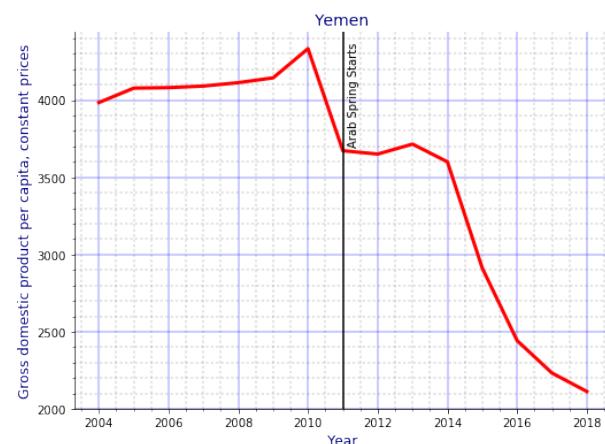
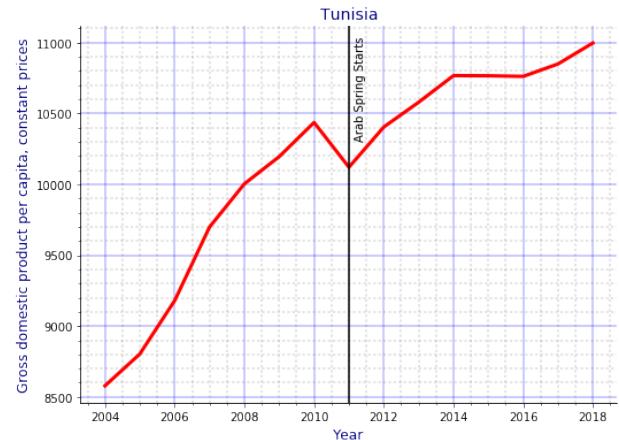
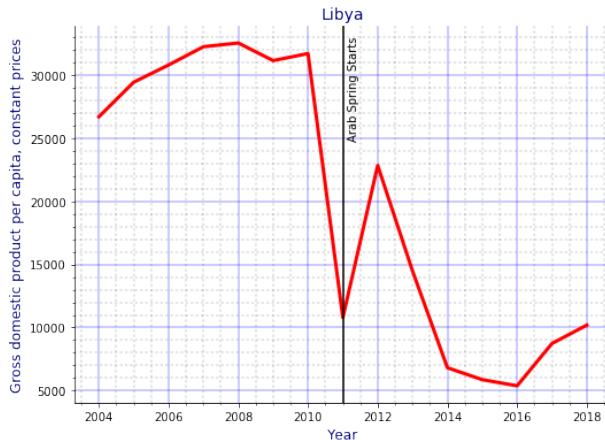
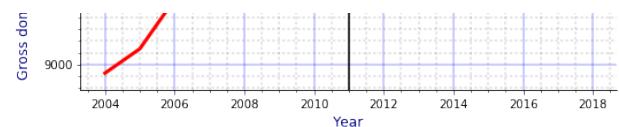
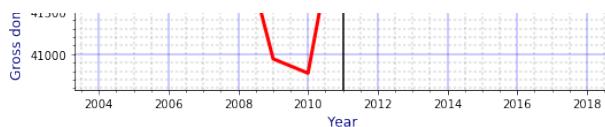




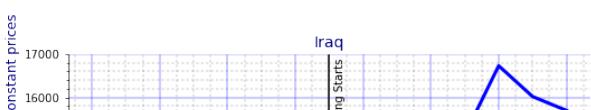
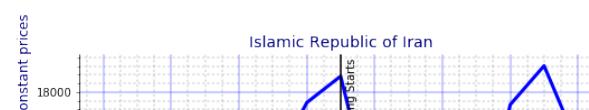
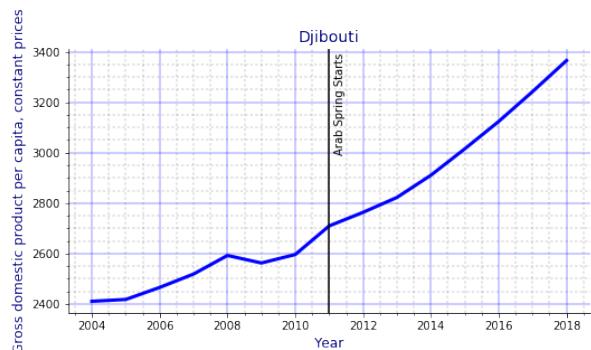
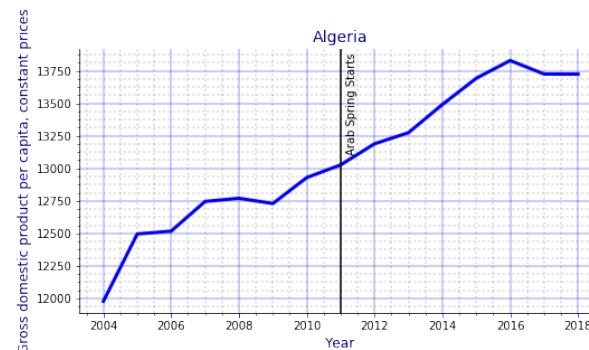
```
In [175]: for item in [True, False]:
    plots(a[1], item)
plt.show()
```

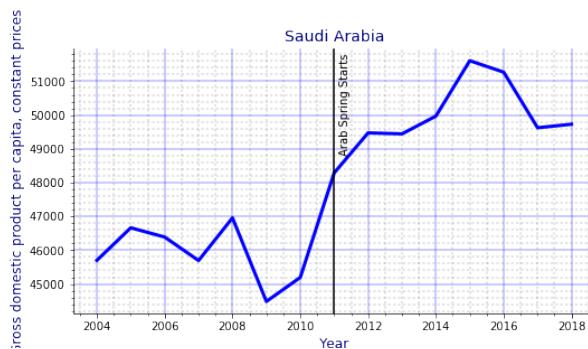
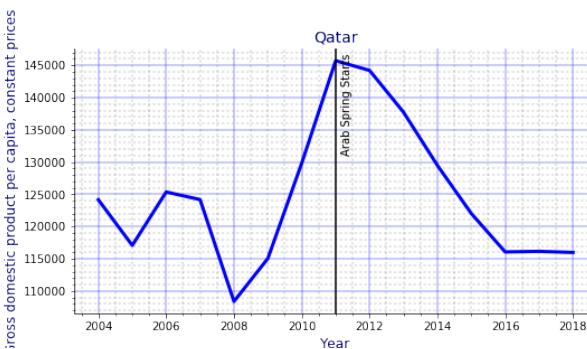
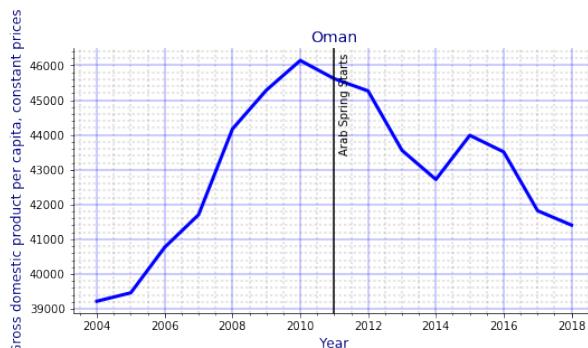
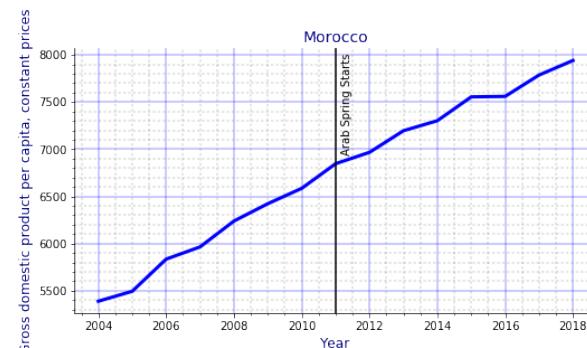
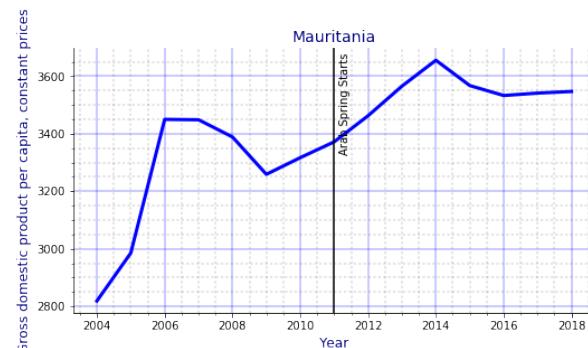
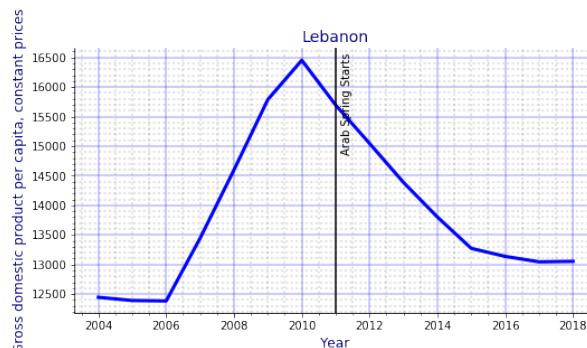
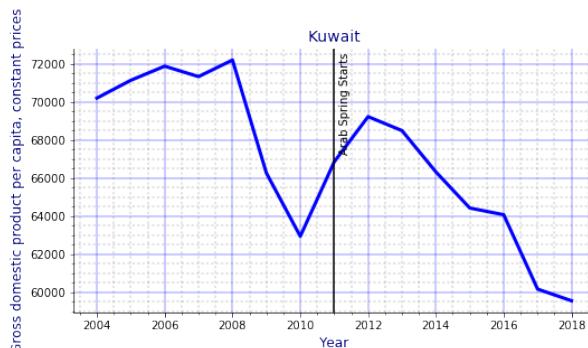
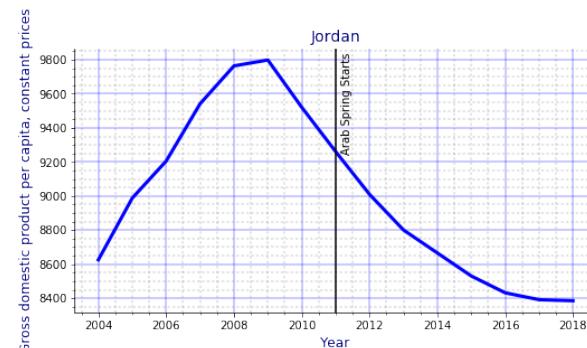
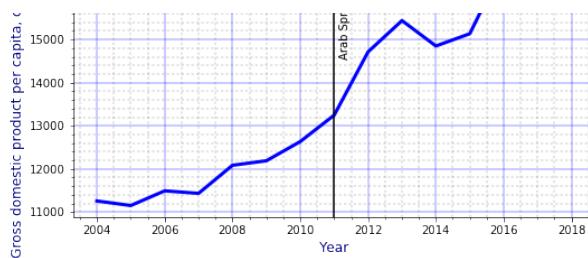
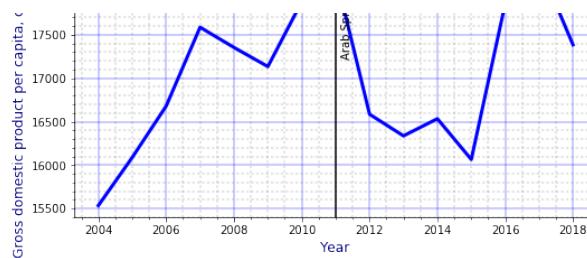
### Gross domestic product per capita, constant prices for Arab Spring Countries



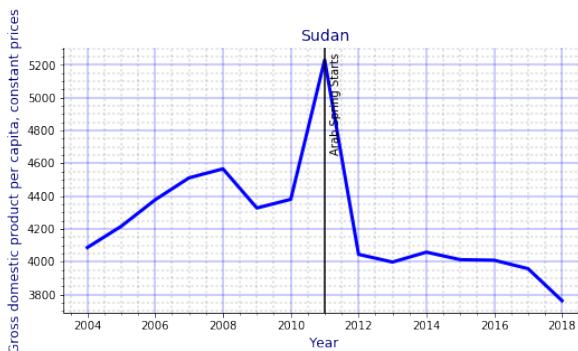


### Gross domestic product per capita, constant prices for non Arab Spring Countries

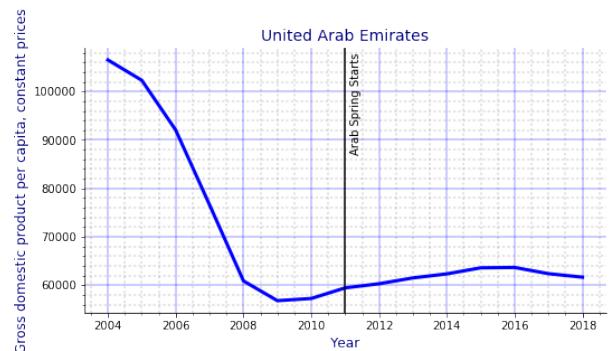




C

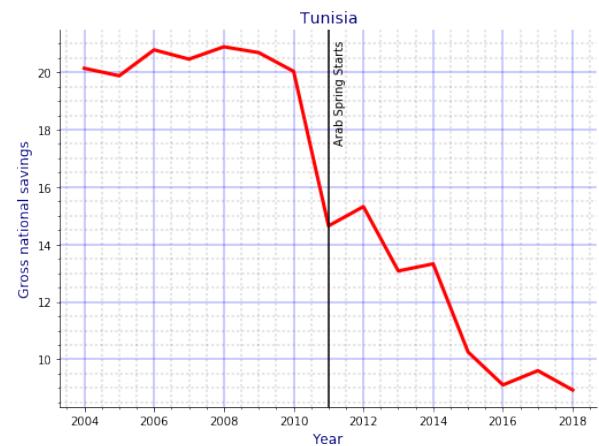
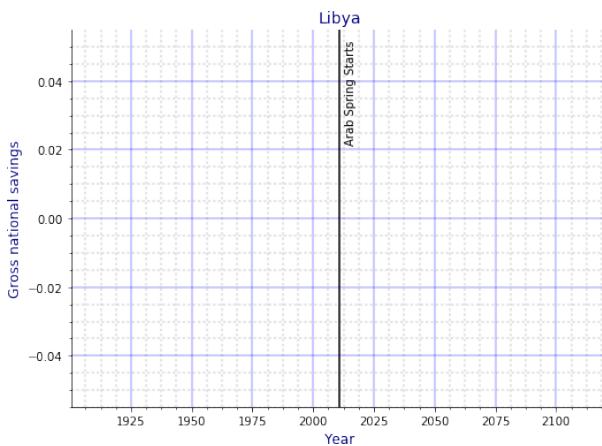
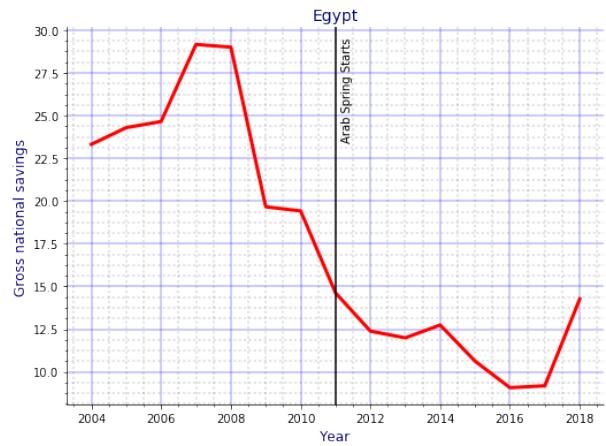
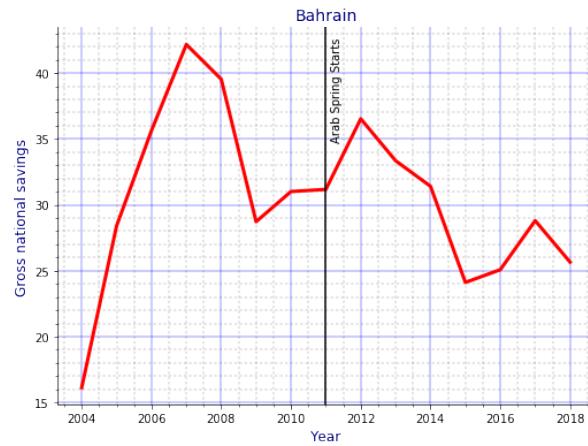


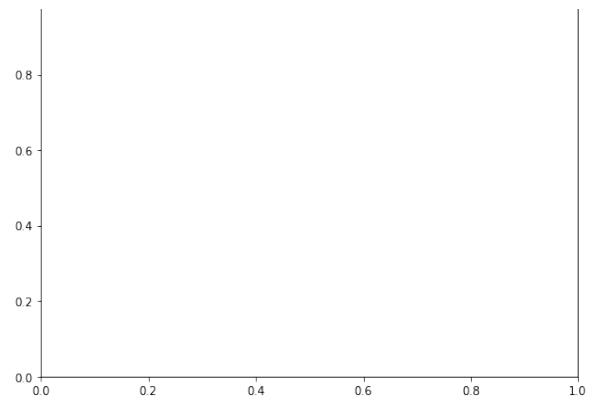
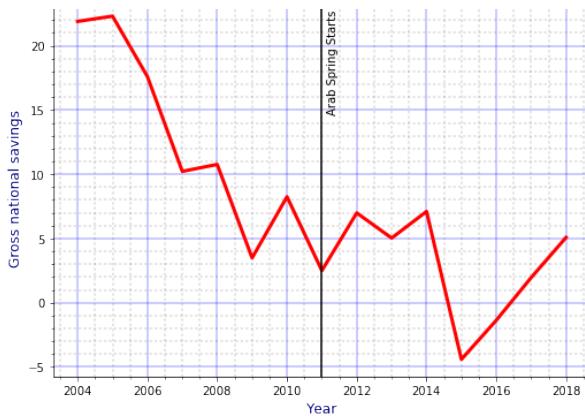
C



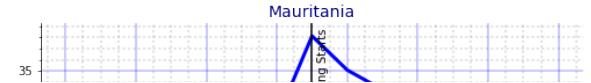
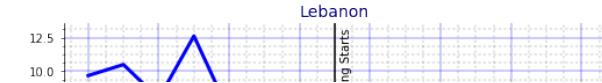
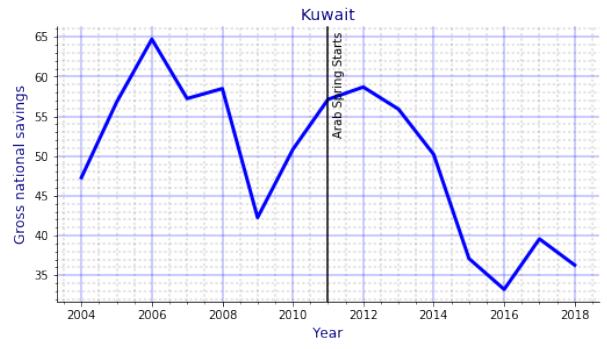
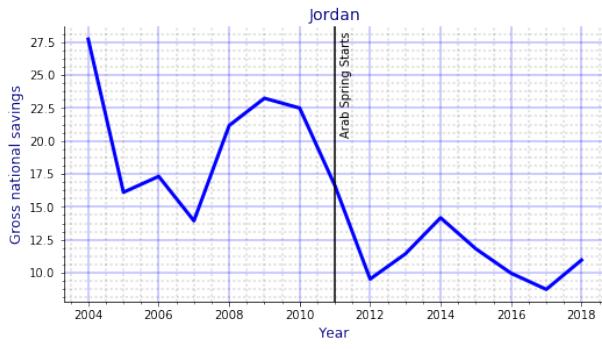
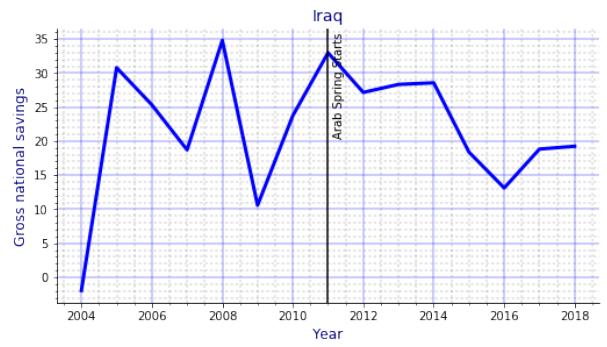
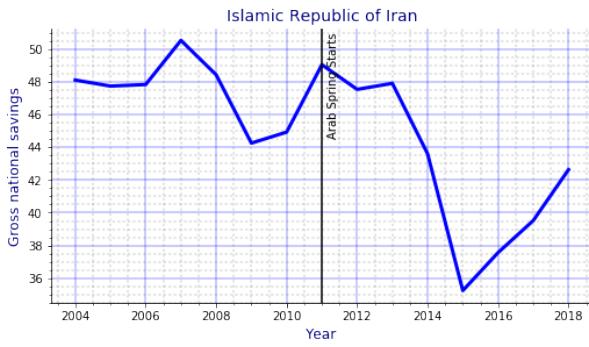
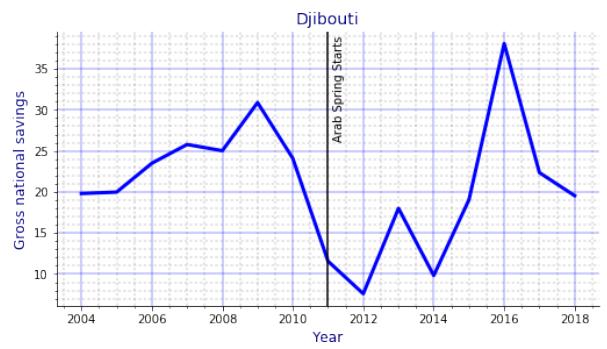
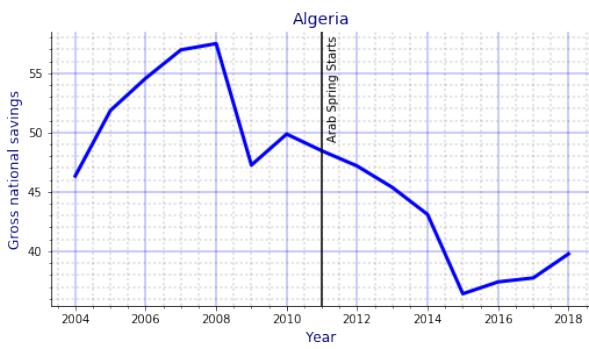
```
In [176]: for item in [True, False]:
    plots(a[2], item)
plt.show()
```

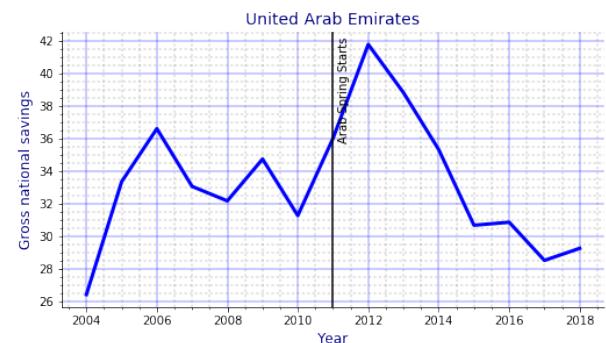
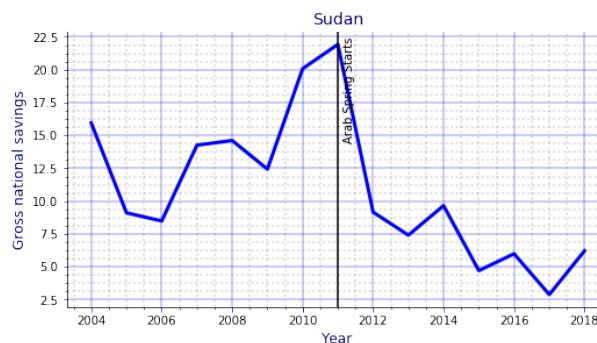
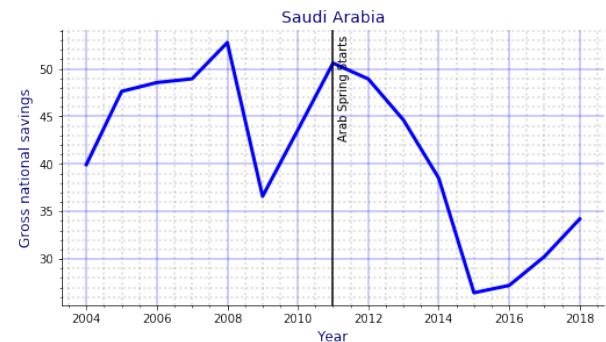
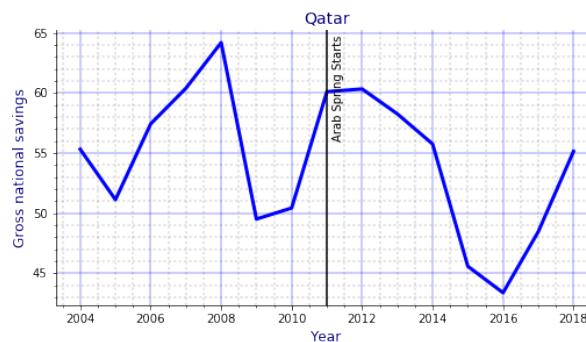
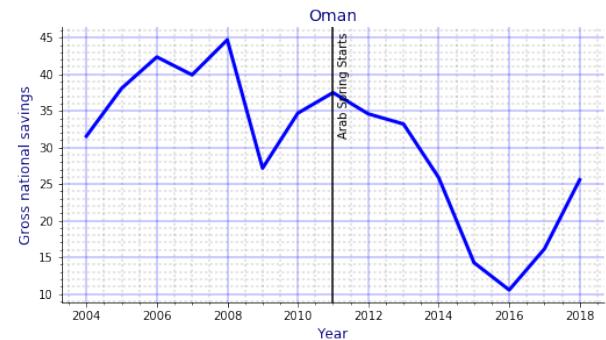
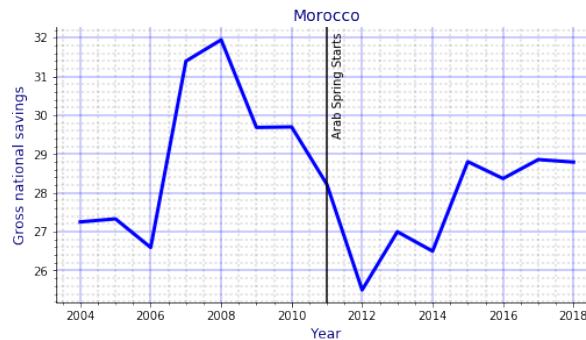
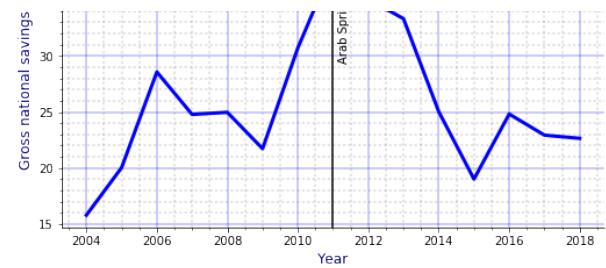
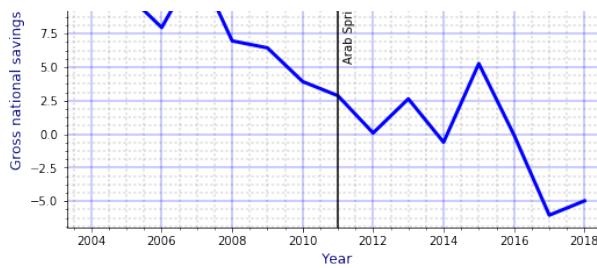
### Gross national savings for Arab Spring Countries





### Gross national savings for non Arab Spring Countries

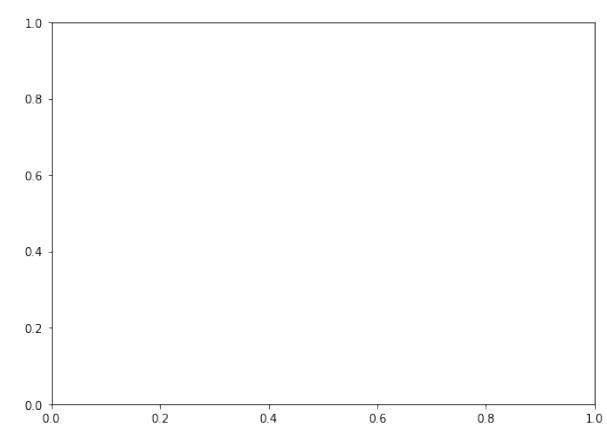
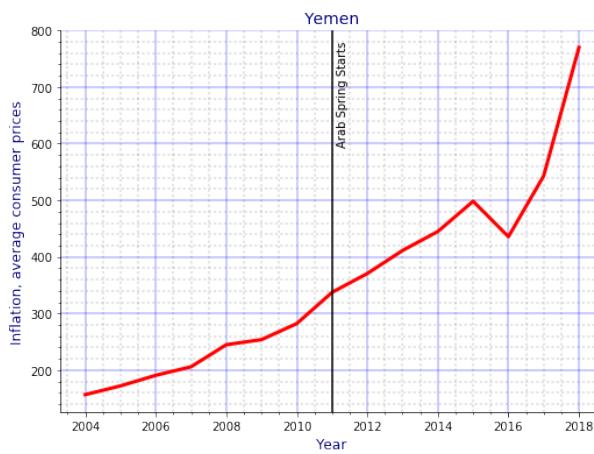
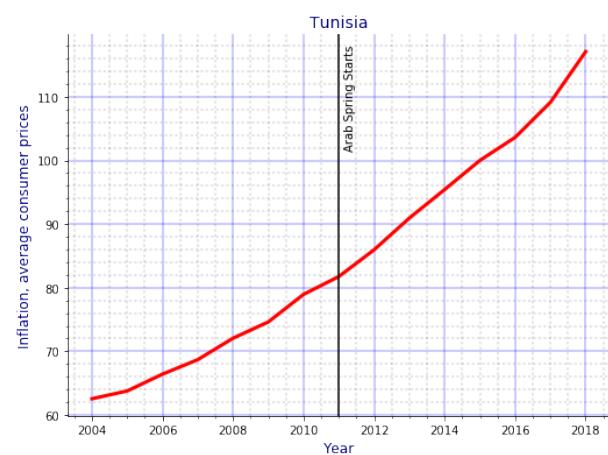
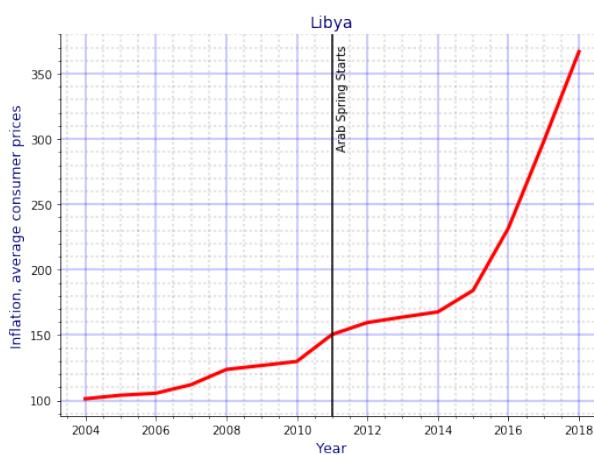
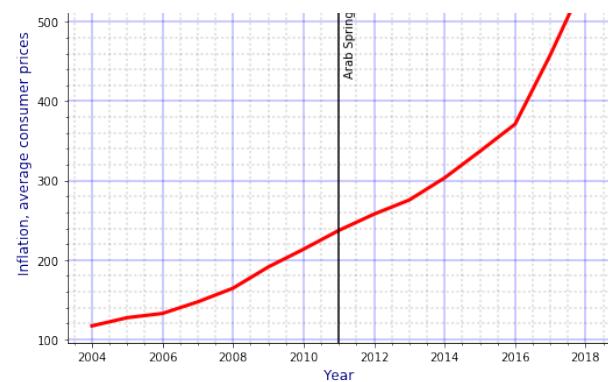
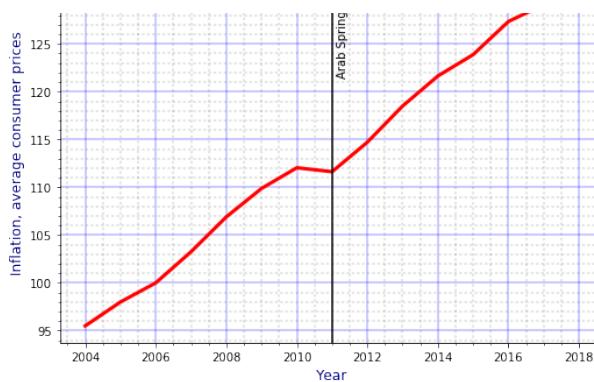




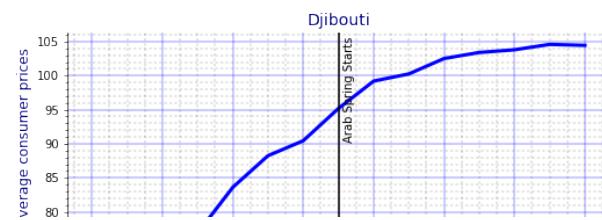
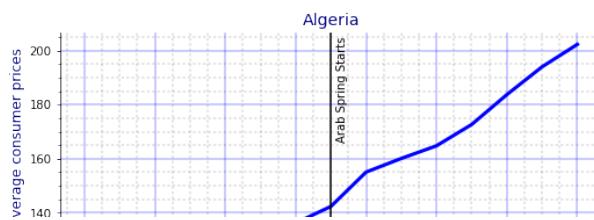
```
In [177]: for item in [True, False]:
    plots(a[3], item)
plt.show()
```

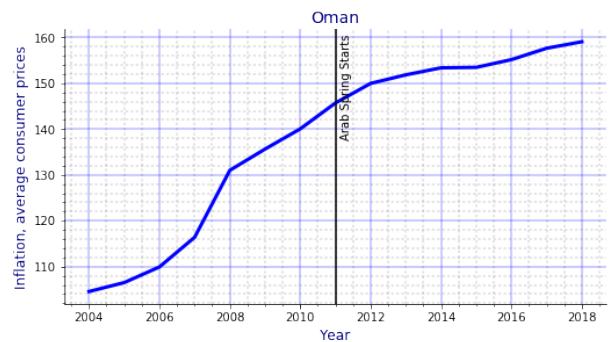
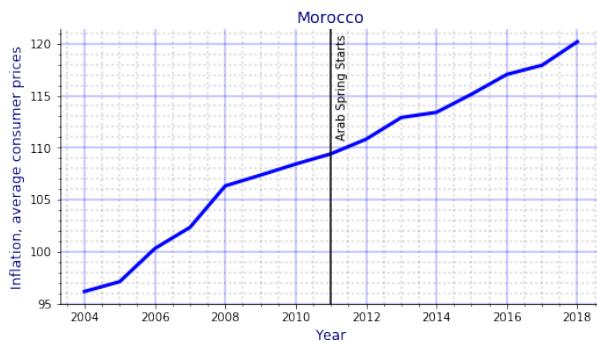
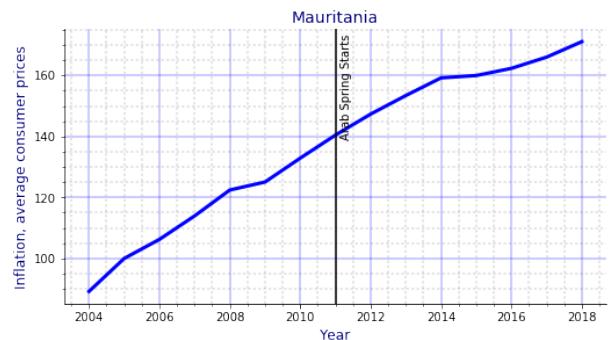
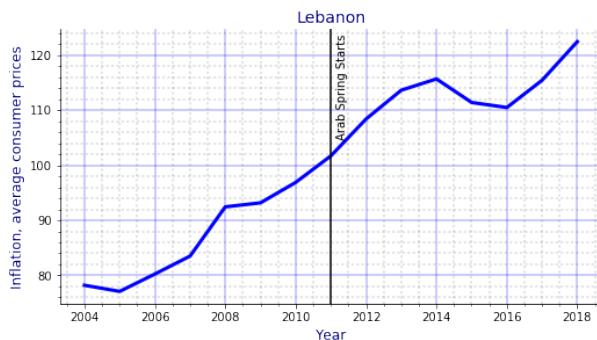
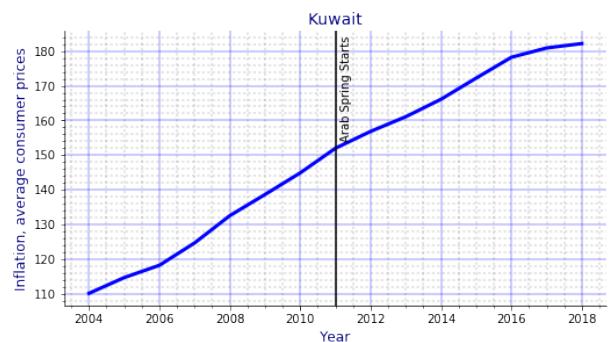
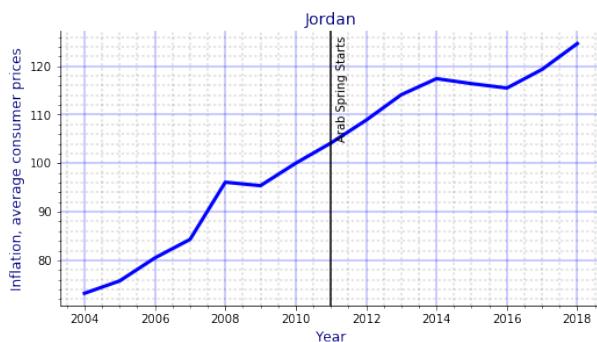
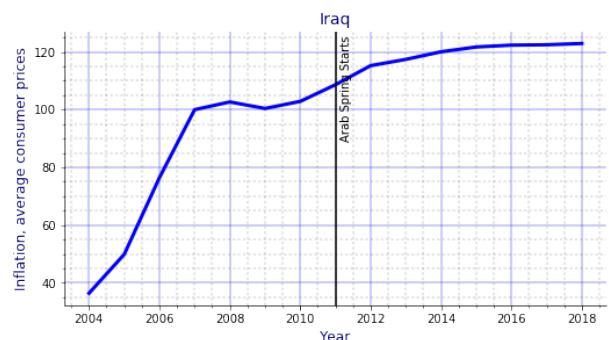
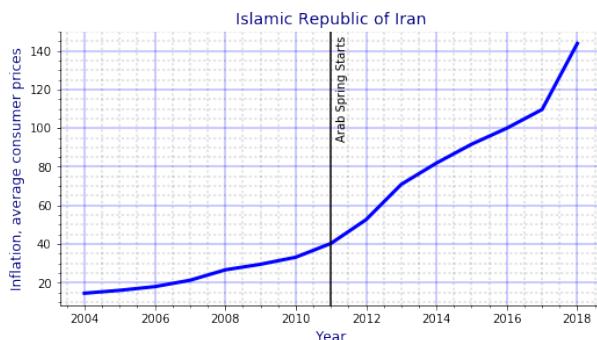
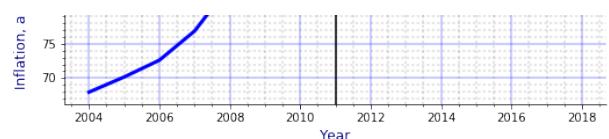
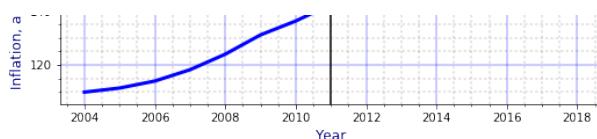
### Inflation, average consumer prices for Arab Spring Countries

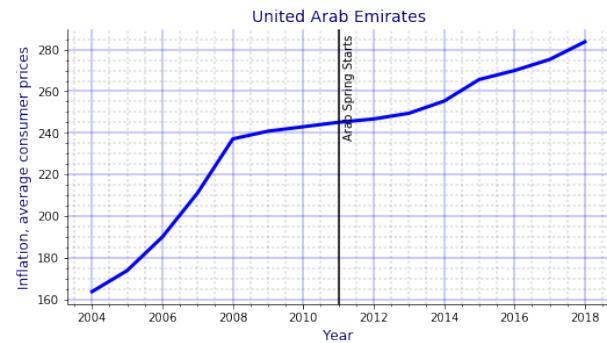
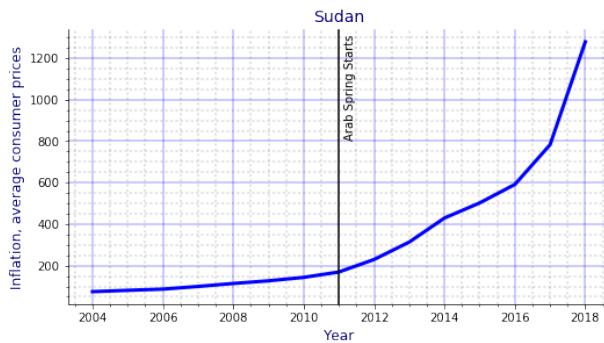
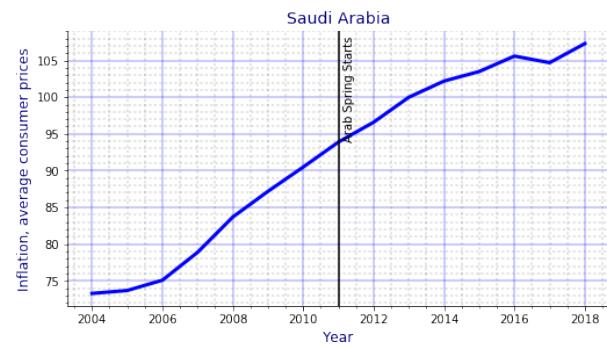
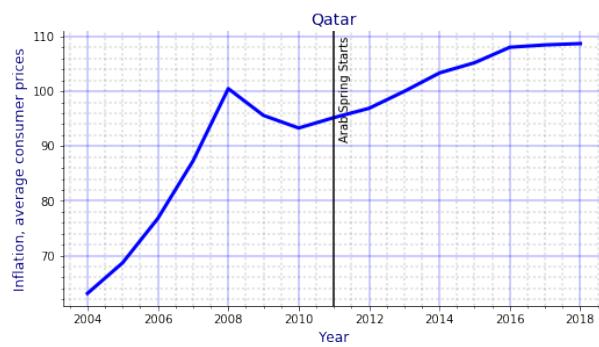




### Inflation, average consumer prices for non Arab Spring Countries

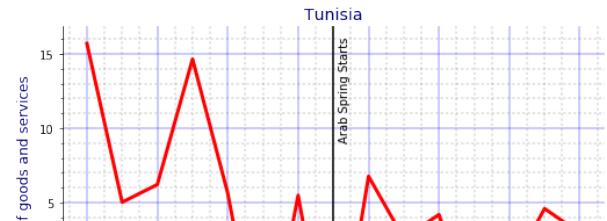
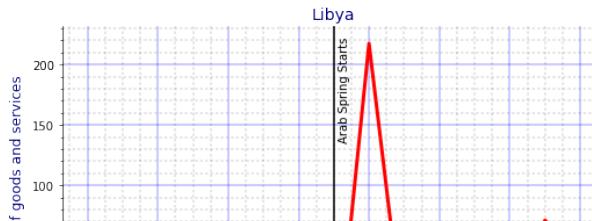
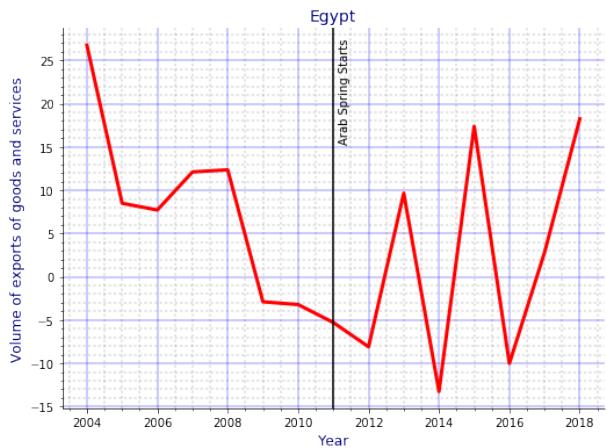
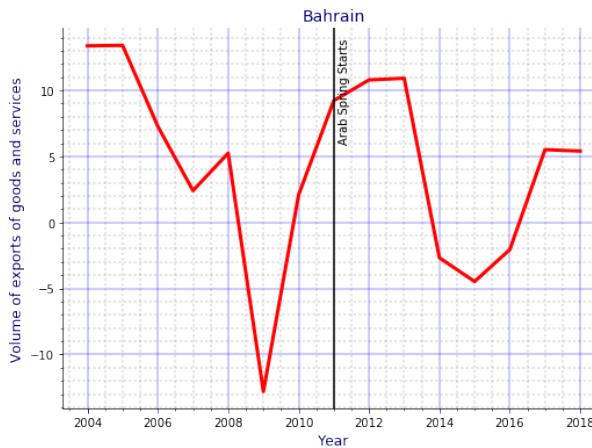


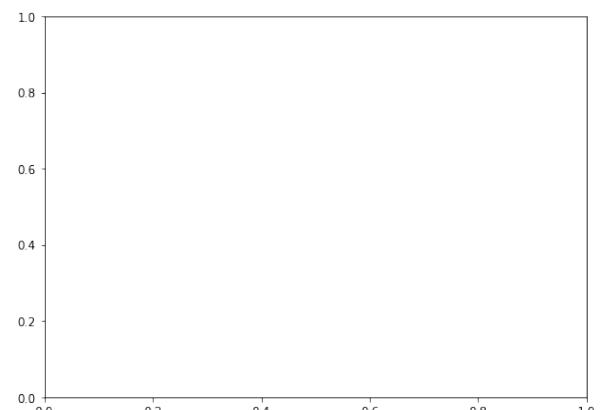
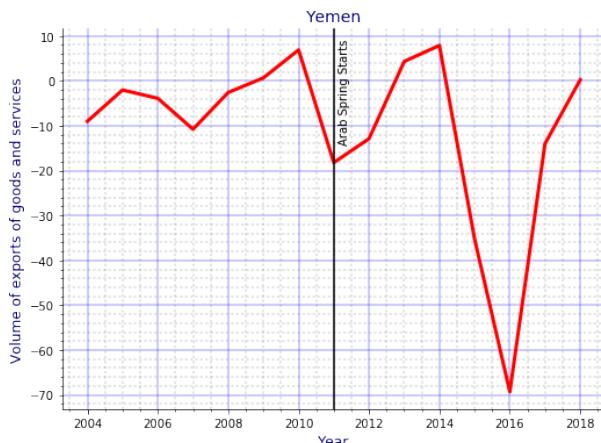
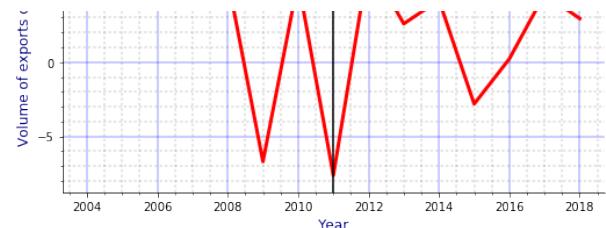
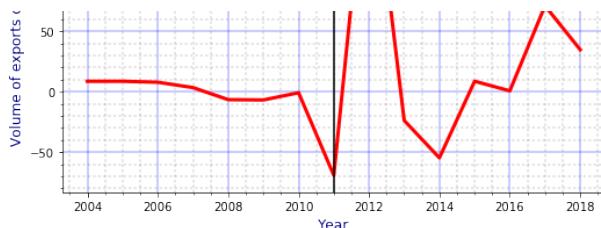




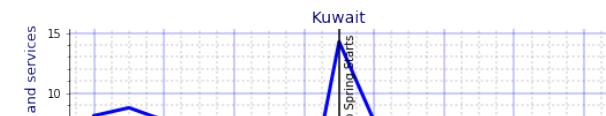
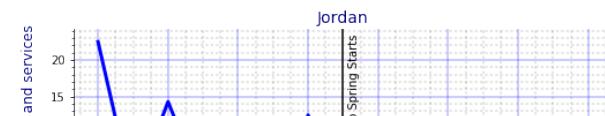
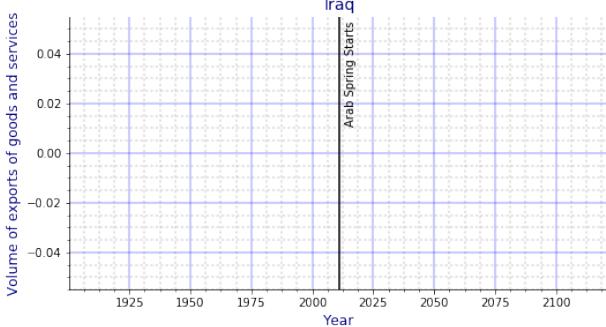
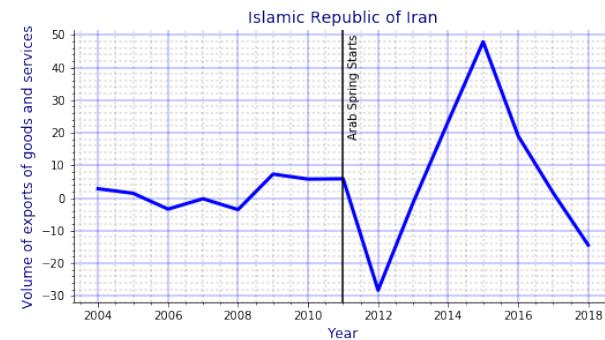
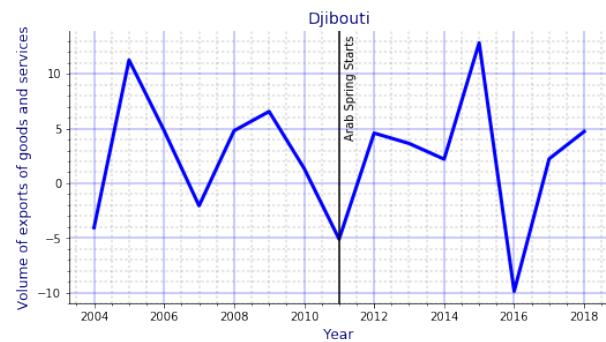
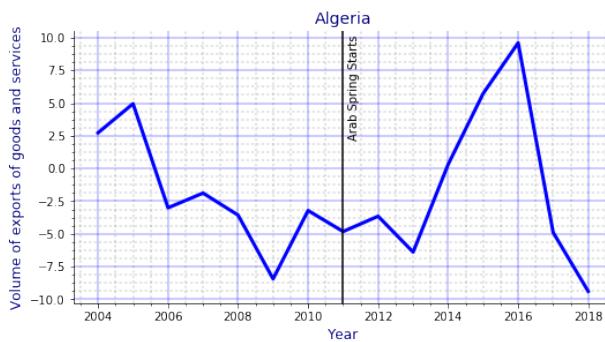
```
In [178]: for item in [True, False]:
    plots(a[4], item)
plt.show()
```

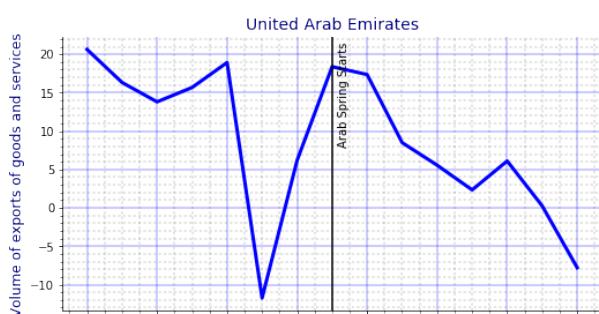
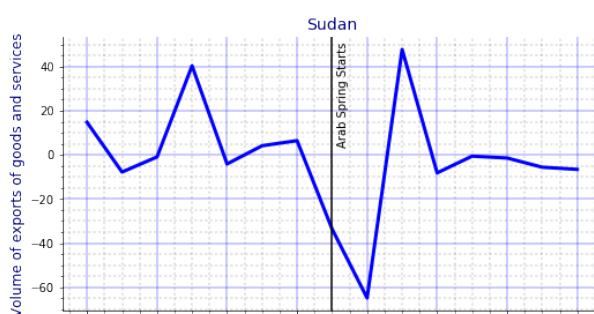
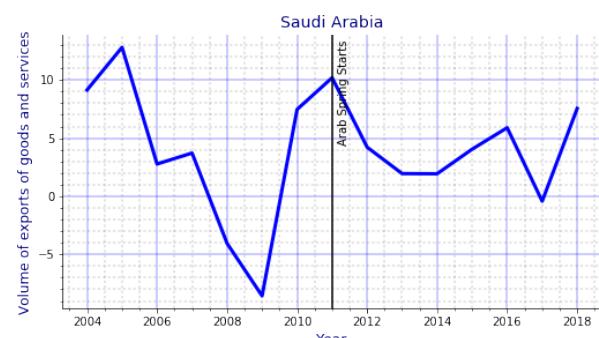
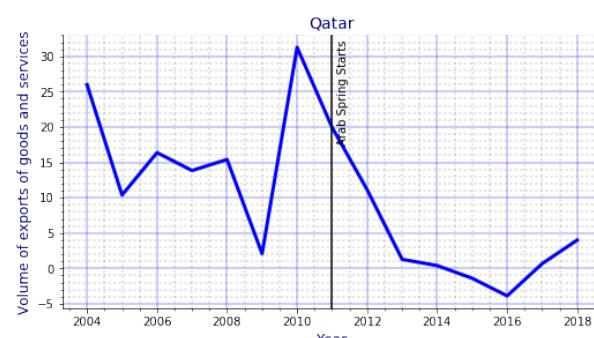
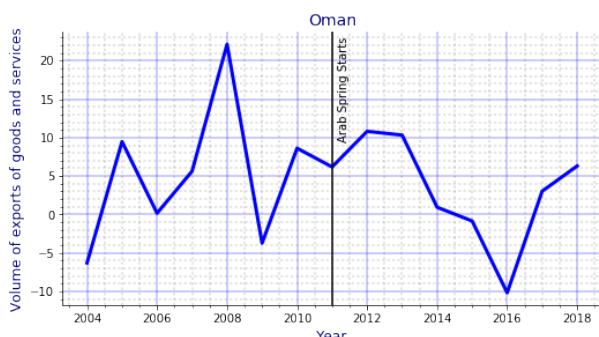
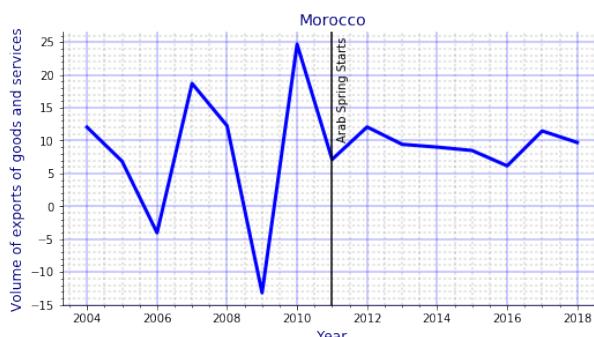
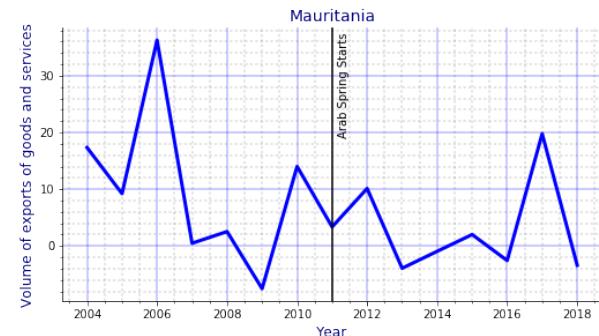
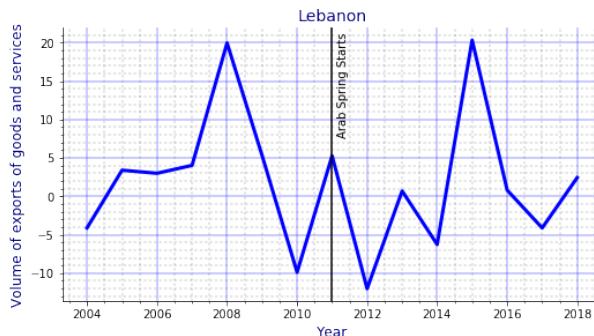
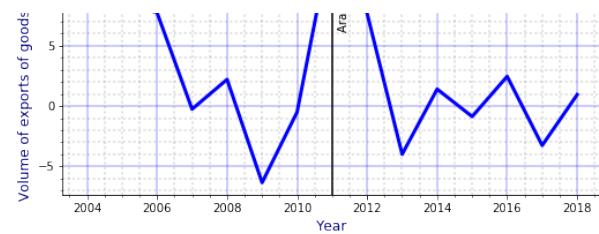
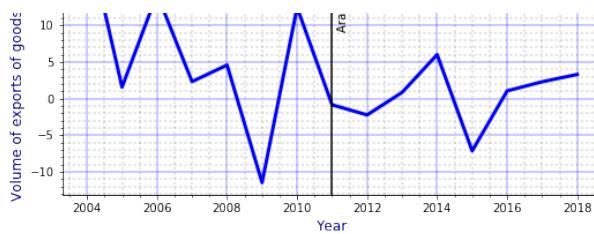
### Volume of exports of goods and services for Arab Spring Countries





### Volume of exports of goods and services for non Arab Spring Countries







### Comments on the graphs:

GDP: The data shows that there was a sizable decrease in GDP in the year leading up to the Arab Spring for all countries that had protests. This drop in GDP is then followed by a similarly sizable increase in GDP in 2012. There doesn't seem to be any obvious trends for all non Arab Spring countries. However, It is interesting to note that there was a fall in GDP between 2011 and 2012 for all GCC countries.

GDP/capita: GDP/capita tells a somewhat similar story for Arab Spring and GCC countries. The changes, however, seem to be less drastic. The only exception seems to be Bahrain, where GDP/Capita was actually on the rise before.

GNS: It seems that GNS as a percentage of GDP has been falling in the years leading up to the Arab Spring in the countries that had major protests. It seems to be a general trend, however, in all Arab countries that GNS as a percentage of GDP has been on the decline, with a few exceptions (e.g Iraq and Djibouti).

Inflation: Inflation level is constantly increasing. Nothing is particularly special about the year 2011.

Exports: There is constant fluctuation in the percentage change of exports. However, it seems that exports fell for all Arab Spring Countries between 2010 and 2011, with the exception of Bahrain. It also seems like exports fell for a lot of non-Arab-Spring countries right after 2011. However, given how exports have been fluctuating before and after 2011, it is hard to say that this was a direct result of the Arab Spring.

## Conclusion

The project aims to show the different ways in which the MENA region has been affected by the Arab Spring.

The visualizations show us that no generalizations can be made regarding the Arab Spring, as it had very different effects on different countries. They also show how the Arab Spring has had major effects on the countries that did not have revolts, which is quite interesting.