In [1]:

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

# Lab 2 : Web scraping and API requests

In this lab you will practice scraping data from a website, as well as doing some priliminary analysis on them.

(Deadline: Thursday ( 11:59 pm

## Part 1: Scraping Data From Wikipedia

We have completed a similar task during lecture. You have to scrap a specific page of Wikipedia and answer some questions regarding the data you have collected. You have to get the data about different countries and their respective populations from the following page:
https://en.wikipedia.org/wiki/List_of_countries_by_past_and_future_population (https://en.wikipedia.org/wiki/List_of_countries_by_past_and_future_population)

This page contains multiple tables for past and future population of countries. For the first part of this lab do the following:

1. Fetch the data from wikipedia with "requests" library
2. Parse html data with BeautifulSoup library
3. Use BeautifulSoup to extract specific tables
4. Combine the tables and convert the data into a dictionary
5. Make a pandas dataframe from the dictionary
6. Answer some questions and do some basic visualization!

### 1.1 Get the data from wikipedia (5 pts)

Use "requests" library.

In [2]:

```
# Your code here
import requests
website_url = requests.get('https://en.wikipedia.org/wiki/List_of_countries_by_past_and_future_population').text
```

## 1.2 Parse html data with BeautifulSoup

Parse the data using BeautifulSoup. Remember that BeautifulSoup has many useful attributes such as prettify(), find(attribute), and find_all(attribute). Check the documentation for more info:
https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (https://www.crummy.com/software/BeautifulSoup/bs4/doc/)

### 1.2.a Find the first title object and extract and print the string stored in it (5 pts)

In [3]:

```
# Your code here
from bs4 import BeautifulSoup
soup = BeautifulSoup(website_url,'lxml')
title = soup.find('title').text.strip()
print(title)
```

```
List of countries by past and future population - Wi
kipedia
```

### 1.2.b Find all the paragrpahs, store them in a list, and print the first 10 (5 pts)

In [4]:

```
# Your code here
paragraph = []
p_tags = soup.find_all('p')
for each in p_tags:
    paragraph.append(str(each.get_text()))
print(paragraph[0:10])
```

['All the figures shown here have been sourced from the International Data Base (IDB) Division of the United States Census Bureau. Every individual value has been rounded to the nearest thousand, to assure data coherence, particularly when adding up (sub)totals. Although data from specific statistical offices may be more accurate, the information provided here has the advantage of being homogeneous.\n', 'Population estimates, as long as they are based on recent censuses, can be more easily projected into the near future than many macroeconomic indicators, such as GDP, which are much more sensitive to political and/or economic crises. This means that demographic estimates for the next five (or even ten) years can be more accurate than the projected evolution of GDP over the same time period (which may also be distorted by inflation).\n', 'However, no projected population figures can be considered exact. As the IDB states, "figures beyond the years 2020-2025 should be taken with caution", as the "census way towards those years has yet to be paved". Thus projections can be said to be looking through a kind of "cloudy glass"[1] or a "misty window": realistically, the projections are "guesstimates".\n', 'To make things complicated, not all countries carry out censuses regularly, especially some of the poorer, faster-growing sub-Saharan African nations (whose evolution may be more interesting, from a demographer\'s point of view, than the "stagnated" populations of countries like Germany or Italy). As is well known from the statistics, the population of many sub-Saharan nations, as well as other nations like Algeria, Bangladesh, Egypt and Pakistan, with their low family planning, are growing much faster than in the aging European nations or Japan. In general, although population growth in the former countries may slow in the future, it is unlikely that it will have stabilized by 2050, as predicted by the IDB data in some cases; they may also stay near the relatively high average level of 1.5% increase per year. Something similar can be said about China, whose population is still growing at an absolute rate of some 10 million per year, despite its government\'s efforts to stabilize it through its one child per couple policy.\n', "On the other hand, some other countries, like the small Asian state of Bhutan, have

```
only recently had a thorough census for the first ti
me: In Bhutan's case in particular, before its natio
nal 2005 population survey,[2][3][4] the IDB estimat
ed its population at over 2 million; this was drasti
cally reduced when the new census results were final
ly included in its database.\n", 'Besides, the IDB u
sually takes some time before including new data, as
happened in the case of Indonesia. That country was
reported by the IDB to have an inflated population o
f some 242 million by mid-2005, because it had not s
till processed the final results of the 2000 Indones
ian census.[5][6][7][8] There was a similar discrepa
ncy with the relatively recent Ethiopian 2007 census
,[9][10] which gave a preliminary result of "only" 7
3,918,505 inhabitants.\n', 'The largest absolute pot
ential discrepancies are naturally related to the mo
st populous nations. However, smaller states, such a
s Tuvalu, can have large relative discrepancies. For
instance, the 2002 census in that Oceanian island, w
hich gave a final population of 9,561[11] shows that
IDB estimates can be significantly off.\n', 'The nat
ional 1 July, mid-year population estimates (usually
based on past national censuses) supplied in these t
ables are given in thousands.\n', 'The table columns
can be sorted by clicking on their respective headin
g.\n', 'The retrospective figures use the present-da
y names and world political division: for example, t
he table gives data for each of the 15 republics of
the former Soviet Union, as if they had already been
independent in 1950. The opposite is the case for Ge
rmany, which had been divided since the end of the S
econd World War but was reunified on October 3, 1990
.\n']
```

## 1.3 Extract the tables (10 pts)

We only care about the tables that contain historical population data. Extract all of them.

In [5]:

```
# Your code here
# You need to  find all objects that include the css class “wiki table” within the soup object.
tables = soup.find_all('table',{'class':'wikitable'})
```

In [6]:

```
# check the tables you extracted

from IPython.core.display import display, HTML
display(HTML(tables[0].prettify()))
```

| Country (or dependent territory) | 1950 | 1955 |
|---|---|---|
| Afghanistan (/wiki/Afghanistan) | 8,151 | 8,892 |
| Albania (/wiki/Albania) | 1,228 | 1,393 |
| Algeria (/wiki/Algeria) | 8,893 | 9,842 |
| American Samoa (/wiki/American_Samoa) | 20 | 20 |
| Andorra (/wiki/Andorra) | 7 | 7 |
| Angola (/wiki/Angola) | 4,118 | 4,424 |
| Anguilla (/wiki/Anguilla) | 6 | 6 |
| Antigua and Barbuda (/wiki/Antigua_and_Barbuda) | 46 | 52 |
| Argentina (/wiki/Argentina) | 17,151 | 18,928 |
| Armenia (/wiki/Armenia) | 1,356 | 1,566 |
| Aruba (/wiki/Aruba) | 50 | 54 |
| Australia (/wiki/Australia) | 8,268 | 9,278 |
| Austria (/wiki/Austria) | 6,936 | 6,947 |
| Azerbaijan (/wiki/Azerbaijan) | 2,886 | 3,314 |
| Bahamas (/wiki/The_Bahamas) | 71 | 88 |
| Bahrain (/wiki/Bahrain) | 115 | 131 |
| Bangladesh (/wiki/Bangladesh) | 45,646 | 49,589 |
| Barbados (/wiki/Barbados) | 211 | 228 |

| | | |
|---|---|---|
| Belarus (/wiki/Belarus) | 7,723 | 7,781 |
| Belgium (/wiki/Belgium) | 8,640 | 8,869 |
| Belize (/wiki/Belize) | 66 | 77 |
| Benin (/wiki/Benin) | 1,673 | 1,847 |
| Bermuda (/wiki/Bermuda) | 39 | 42 |
| Bhutan (/wiki/Bhutan) | 164 | 187 |
| Bolivia (/wiki/Bolivia) | 2,767 | 3,075 |
| Bosnia and Herzegovina (/wiki/Bosnia_and_Herzegovina) | 2,663 | 2,975 |
| Botswana (/wiki/Botswana) | 431 | 462 |
| Brazil (/wiki/Brazil) | 53,444 | 61,652 |
| British Virgin Islands (/wiki/British_Virgin_Islands) | 7 | 7 |
| Brunei (/wiki/Brunei) | 45 | 61 |
| Bulgaria (/wiki/Bulgaria) | 7,251 | 7,500 |
| Burkina Faso (/wiki/Burkina_Faso) | 4,377 | 4,615 |
| Burundi (/wiki/Burundi) | 2,363 | 2,577 |
| Cambodia (/wiki/Cambodia) | 4,472 | 5,049 |
| Cameroon (/wiki/Cameroon) | 4,888 | 5,211 |
| Canada (/wiki/Canada) | 14,012 | 16,051 |
| Cape Verde (/wiki/Cape_Verde) | 147 | 170 |
| Cayman Islands (/wiki/Cayman_Islands) | 7 | 7 |
| Central African Republic (/wiki/Central_African_Republic) | 1,260 | 1,349 |
| Chad (/wiki/Chad) | 2,608 | 2,806 |
| Chile (/wiki/Chile) | 6,091 | 6,744 |
| China (/wiki/China) | 562,580 | 607,047 |
| Colombia (/wiki/Colombia) | 11,592 | 13,589 |
| Comoros (/wiki/Comoros) | 149 | 164 |
| Cook Islands (/wiki/Cook_Islands) | 15 | 17 |
| Costa Rica (/wiki/Costa_Rica) | 867 | 1,032 |
| Croatia (/wiki/Croatia) | 3,838 | 3,956 |

| | | |
|---|---|---|
| Cuba (/wiki/Cuba) | 5,785 | 6,382 |
| Curaçao (/wiki/Cura%C3%A7ao) | 102 | 112 |
| Cyprus (/wiki/Cyprus) | 495 | 533 |
| Czech Republic (/wiki/Czech_Republic) | 8,926 | 9,366 |
| Democratic Republic of the Congo (/wiki/Democratic_Republic_of_the_Congo) | 13,569 | 14,953 |
| Denmark (/wiki/Denmark) | 4,272 | 4,440 |
| Djibouti (/wiki/Djibouti) | 80 | 91 |
| Dominica (/wiki/Dominica) | 52 | 57 |
| Dominican Republic (/wiki/Dominican_Republic) | 2,353 | 2,738 |
| Timor-Leste (/wiki/East_Timor) | 436 | 473 |
| Ecuador (/wiki/Ecuador) | 3,370 | 3,843 |
| Egypt (/wiki/Egypt) | 21,198 | 23,856 |
| El Salvador (/wiki/El_Salvador) | 1,940 | 2,222 |
| Equatorial Guinea (/wiki/Equatorial_Guinea) | 212 | 226 |
| Eritrea (/wiki/Eritrea) | 1,403 | 1,499 |
| Estonia (/wiki/Estonia) | 1,096 | 1,155 |
| Ethiopia (/wiki/Ethiopia) | 20,175 | 21,991 |
| Faroe Islands (/wiki/Faroe_Islands) | 32 | 33 |
| Federated States of Micronesia (/wiki/Federated_States_of_Micronesia) | 31 | 36 |
| Fiji (/wiki/Fiji) | 288 | 333 |
| Finland (/wiki/Finland) | 4,009 | 4,235 |
| France (/wiki/France) | 42,540 | 44,243 |
| French Polynesia (/wiki/French_Polynesia) | 63 | 72 |
| Gabon (/wiki/Gabon) | 416 | 429 |
| Georgia (/wiki/Georgia_(country)) | 3,516 | 3,828 |
| Germany (/wiki/Germany) | 68,375 | 70,196 |
| Ghana (/wiki/Ghana) | 5,298 | 6,049 |
| Gibraltar (/wiki/Gibraltar) | 23 | 24 |

| | | |
|---|---|---|
| Greece (/wiki/Greece) | 7,567 | 7,966 |
| Greenland (/wiki/Greenland) | 23 | 27 |
| Grenada (/wiki/Grenada) | 76 | 85 |
| Guam (/wiki/Guam) | 60 | 69 |
| Guatemala (/wiki/Guatemala) | 2,969 | 3,488 |
| Guernsey (/wiki/Guernsey) | 46 | 47 |
| Guinea (/wiki/Guinea) | 2,586 | 2,787 |
| Guinea-Bissau (/wiki/Guinea-Bissau) | 574 | 592 |
| Guyana (/wiki/Guyana) | 428 | 492 |
| Haiti (/wiki/Haiti) | 3,098 | 3,365 |
| Honduras (/wiki/Honduras) | 1,432 | 1,663 |
| Hong Kong (/wiki/Hong_Kong) | 2,238 | 2,491 |
| Hungary (/wiki/Hungary) | 9,339 | 9,826 |
| Iceland (/wiki/Iceland) | 143 | 159 |
| India (/wiki/India) | 369,881 | 404,268 |
| Indonesia (/wiki/Indonesia) | 82,979 | 90,255 |
| Iran (/wiki/Iran) | 16,358 | 18,739 |
| Iraq (/wiki/Iraq) | 5,164 | 5,904 |
| Ireland (/wiki/Republic_of_Ireland) | 2,964 | 2,917 |
| Isle of Man (/wiki/Isle_of_Man) | 55 | 52 |
| Israel (/wiki/Israel) | 1,287 | 1,771 |
| Italy (/wiki/Italy) | 47,106 | 48,634 |
| Ivory Coast (/wiki/Ivory_Coast) | 2,861 | 3,165 |
| Jamaica (/wiki/Jamaica) | 1,385 | 1,489 |
| Japan (/wiki/Japan) | 83,806 | 89,816 |
| Jersey (/wiki/Jersey) | 57 | 60 |
| Jordan (/wiki/Jordan) | 562 | 689 |
| Kazakhstan (/wiki/Kazakhstan) | 6,694 | 7,977 |
| Kenya (/wiki/Kenya) | 6,122 | 7,034 |

| | | |
|---|---|---|
| Kiribati (/wiki/Kiribati) | 34 | 37 |
| Kosovo (/wiki/Kosovo) | 762 | 854 |
| Kuwait (/wiki/Kuwait) | 145 | 187 |
| Kyrgyzstan (/wiki/Kyrgyzstan) | 1,739 | 1,902 |
| Laos (/wiki/Laos) | 1,886 | 2,078 |
| Latvia (/wiki/Latvia) | 1,937 | 2,003 |
| Lebanon (/wiki/Lebanon) | 1,365 | 1,561 |
| Lesotho (/wiki/Lesotho) | 727 | 787 |
| Liberia (/wiki/Liberia) | 824 | 929 |
| Liechtenstein (/wiki/Liechtenstein) | 14 | 15 |
| Lithuania (/wiki/Lithuania) | 2,554 | 2,615 |
| Luxembourg (/wiki/Luxembourg) | 296 | 305 |
| Libya (/wiki/Libya) | 962 | 1,123 |
| Macau (/wiki/Macau) | 206 | 193 |
| North Macedonia (/wiki/North_Macedonia) | 1,225 | 1,341 |
| Madagascar (/wiki/Madagascar) | 4,621 | 5,003 |
| Malawi (/wiki/Malawi) | 2,817 | 3,089 |
| Malaysia (/wiki/Malaysia) | 6,434 | 7,312 |
| Maldives (/wiki/Maldives) | 80 | 81 |
| Mali (/wiki/Mali) | 3,688 | 4,072 |
| Malta (/wiki/Malta) | 312 | 315 |
| Marshall Islands (/wiki/Marshall_Islands) | 11 | 13 |
| Mauritania (/wiki/Mauritania) | 1,006 | 1,054 |
| Mauritius (/wiki/Mauritius) | 482 | 572 |
| Mexico (/wiki/Mexico) | 28,486 | 32,930 |
| Moldova (/wiki/Moldova) | 2,337 | 2,623 |
| Monaco (/wiki/Monaco) | 19 | 19 |
| Mongolia (/wiki/Mongolia) | 779 | 845 |
| Montenegro (/wiki/Montenegro) | 396 | 432 |

| | | |
|---|---|---|
| Montserrat (/wiki/Montserrat) | 14 | 13 |
| Morocco (/wiki/Morocco) | 9,344 | 10,782 |
| Mozambique (/wiki/Mozambique) | 6,251 | 6,782 |
| Myanmar (/wiki/Myanmar) | 19,488 | 21,050 |
| Namibia (/wiki/Namibia) | 464 | 522 |
| Nauru (/wiki/Nauru) | 4 | 4 |
| Nepal (/wiki/Nepal) | 8,990 | 9,480 |
| Netherlands (/wiki/Netherlands) | 10,121 | 10,759 |
| New Caledonia (/wiki/New_Caledonia) | 56 | 65 |
| New Zealand (/wiki/New_Zealand) | 1,909 | 2,137 |
| Nicaragua (/wiki/Nicaragua) | 1,098 | 1,278 |
| Niger (/wiki/Niger) | 3,272 | 3,560 |
| Nigeria (/wiki/Nigeria) | 31,797 | 35,955 |
| North Korea (/wiki/North_Korea) | 9,472 | 8,864 |
| Northern Mariana Islands (/wiki/Northern_Mariana_Islands) | 7 | 8 |
| Norway (/wiki/Norway) | 3,266 | 3,428 |
| Oman (/wiki/Oman) | 489 | 540 |
| Pakistan (/wiki/Pakistan) | 40,383 | 45,536 |
| Palau (/wiki/Palau) | 8 | 9 |
| Palestine (/wiki/State_of_Palestine) | 1,018 | 1,055 |
| Panama (/wiki/Panama) | 893 | 1,011 |
| Papua New Guinea (/wiki/Papua_New_Guinea) | 1,413 | 1,546 |
| Paraguay (/wiki/Paraguay) | 1,476 | 1,684 |
| Peru (/wiki/Peru) | 7,633 | 8,672 |
| Philippines (/wiki/Philippines) | 21,132 | 24,336 |
| Poland (/wiki/Poland) | 24,825 | 27,221 |
| Portugal (/wiki/Portugal) | 8,443 | 8,693 |
| Puerto Rico (/wiki/Puerto_Rico) | 2,219 | 2,251 |
| Qatar (/wiki/Qatar) | 26 | 36 |

| | | |
|---|---|---|
| Republic of the Congo (/wiki/Republic_of_the_Congo) | 827 | 904 |
| Romania (/wiki/Romania) | 16,312 | 17,326 |
| Russia (/wiki/Russia) | 101,937 | 111,126 |
| Rwanda (/wiki/Rwanda) | 2,440 | 2,699 |
| Saint Barthélemy (/wiki/Saint_Barth%C3%A9lemy) | 3 | 3 |
| Saint Helena, Ascension and Tristan da Cunha (/wiki/Saint_Helena,_Ascension_and_Tristan_da_Cunha) | 6 | 6 |
| Saint Kitts and Nevis (/wiki/Saint_Kitts_and_Nevis) | 45 | 50 |
| Saint Lucia (/wiki/Saint_Lucia) | 80 | 86 |
| Saint Martin (/wiki/Collectivity_of_Saint_Martin) | 3 | 4 |
| Saint Pierre and Miquelon (/wiki/Saint_Pierre_and_Miquelon) | 5 | 5 |
| Saint Vincent and the Grenadines (/wiki/Saint_Vincent_and_the_Grenadines) | 67 | 76 |
| Samoa (/wiki/Samoa) | 82 | 94 |
| San Marino (/wiki/San_Marino) | 13 | 14 |
| São Tomé and Príncipe (/wiki/S%C3%A3o_Tom%C3%A9_and_Pr%C3%ADncipe) | 60 | 61 |
| Saudi Arabia (/wiki/Saudi_Arabia) | 3,860 | 4,244 |
| Senegal (/wiki/Senegal) | 2,654 | 2,927 |
| Serbia (/wiki/Serbia) | 5,957 | 6,314 |
| Seychelles (/wiki/Seychelles) | 33 | 36 |
| Sierra Leone (/wiki/Sierra_Leone) | 2,088 | 2,233 |
| Singapore (/wiki/Singapore) | 1,023 | 1,306 |
| Sint Maarten (/wiki/Sint_Maarten) | 3 | 3 |
| Slovakia (/wiki/Slovakia) | 3,464 | 3,727 |
| Slovenia (/wiki/Slovenia) | 1,468 | 1,518 |
| Solomon Islands (/wiki/Solomon_Islands) | 107 | 115 |
| Somalia (/wiki/Somalia) | 2,438 | 2,674 |
| South Africa (/wiki/South_Africa) | 13,596 | 15,369 |
| South Korea (/wiki/South_Korea) | 20,846 | 21,552 |

| | | |
|---|---|---|
| South Sudan (/wiki/South_Sudan) | 2,707 | 2,757 |
| Spain (/wiki/Spain) | 28,063 | 29,319 |
| Sri Lanka (/wiki/Sri_Lanka) | 7,534 | 8,694 |
| Sudan (/wiki/Sudan) | 6,468 | 7,391 |
| Suriname (/wiki/Suriname) | 209 | 241 |
| Eswatini (/wiki/Eswatini) | 278 | 312 |
| Sweden (/wiki/Sweden) | 7,015 | 7,263 |
| Switzerland (/wiki/Switzerland) | 4,695 | 4,981 |
| Syria (/wiki/Syria) | 3,496 | 3,938 |
| Taiwan (/wiki/Taiwan) | 7,982 | 9,486 |
| Tajikistan (/wiki/Tajikistan) | 1,531 | 1,781 |
| Tanzania (/wiki/Tanzania) | 7,935 | 8,971 |
| Thailand (/wiki/Thailand) | 20,042 | 23,452 |
| Gambia (/wiki/The_Gambia) | 272 | 307 |
| Togo (/wiki/Togo) | 1,172 | 1,299 |
| Tonga (/wiki/Tonga) | 46 | 55 |
| Trinidad and Tobago (/wiki/Trinidad_and_Tobago) | 633 | 721 |
| Tunisia (/wiki/Tunisia) | 3,518 | 3,847 |
| Turkey (/wiki/Turkey) | 21,122 | 24,145 |
| Turkmenistan (/wiki/Turkmenistan) | 1,205 | 1,348 |
| Turks and Caicos Islands (/wiki/Turks_and_Caicos_Islands) | 6 | 6 |
| Tuvalu (/wiki/Tuvalu) | 5 | 5 |
| Uganda (/wiki/Uganda) | 5,522 | 6,318 |
| Ukraine (/wiki/Ukraine) | 36,775 | 39,369 |
| United Arab Emirates (/wiki/United_Arab_Emirates) | 72 | 83 |
| United Kingdom (/wiki/United_Kingdom) | 50,128 | 50,947 |
| United States (/wiki/United_States) | 151,869 | 165,070 |
| United States Virgin Islands (/wiki/United_States_Virgin_Islands) | 27 | 28 |
| Uruguay (/wiki/Uruguay) | 2,195 | 2,354 |

| | | |
|---|---|---|
| Uzbekistan (/wiki/Uzbekistan) | 6,293 | 7,233 |
| Vanuatu (/wiki/Vanuatu) | 53 | 59 |
| Venezuela (/wiki/Venezuela) | 5,010 | 6,171 |
| Vietnam (/wiki/Vietnam) | 25,349 | 27,739 |
| Wallis and Futuna (/wiki/Wallis_and_Futuna) | 7 | 8 |
| Western Sahara (/wiki/Western_Sahara) | 10 | 16 |
| Yemen (/wiki/Yemen) | 4,778 | 5,266 |
| Zambia (/wiki/Zambia) | 2,554 | 2,870 |
| Zimbabwe (/wiki/Zimbabwe) | 2,854 | 3,410 |
| **World (/wiki/World)** | **2,557,629** | **2,782,099** |

## 1.4 Convert the tables into a dictionary (35 pts)

Looking at the tables, we only care about the population number throughout the history. You want to associate each country with a series of population values to make a proper time series table you can use to analyze the population in a given coutnry.

First, you need to clean the tables cells from any footnote, links, commas or any garbage values. Once your data is cleaned, make a dictionary and combine each country with its corresponding year/population values across all three tables. An entry in your final dictionary should look like this:

'Albania': {'1950': 1228, '1955': 1393, '1960': 1624, '1965': 1884, '1970': 2157, '1975': 2402, '1980': 2672, '1985': 2957, '1990': 3245, '1995': 3159, '2000': 3159, '2005': 3025, '2010': 2987, '2015': 3030, '2020': 3075, '2025': 3105, '2030': 3103, '2035': 3063, '2040': 2994, '2045': 2913, '2050': 2825},

One way to do it is:

1. First extract the header
2. From your header only store values that are numeric (you can use isnumeric() function, recall that we only care about year values and we don't want to store columns represented by %
3. Once you have all the relevant column names (column that correspond to a year value), you can go over every row of the table
    - Create a dictionary key with the country name
    - Collect and add values corresponding to one of your column names to the dictionary

In [8]:

```
country = []
```

In [9]:

```
for row in soup.find('table', class_='wikitable').find_all('tr')
[1::1]:
    country.append(row.find_all('a')[0].text)
len(country)
```

Out[9]:

```
228
```

In [10]:

```
x = []
for table in tables:
    table_body = table.find('tbody')
    try:
        rows = table_body.find_all('tr')
        for tr in rows:
            cols = tr.find_all('td')
            for td in cols:
                x.append(td.text)
    except:
        print("no tbody")
```

In [11]:

```
x = [x[i].replace("\xa0", "") for i in range(len(x))]
x = [x[i].replace(",", "") for i in range(len(x))]
output = [ a for a in x if a.isnumeric() ]
x = [output[i:i + 7] for i in range(0, len(output), 7)]
```

In [12]:

```
x1 =  [ x[i] for i in range(int(len(x)/3))]
x2 =  [ x[228+i] for i in range(int(len(x)/3))]
x3 =  [ x[(228*2)+i] for i in range(int(len(x)/3))]
```

In [13]:

```
import pandas as pd
df1 = pd.DataFrame(x1)
df1.columns = ['1950', '1955', '1960', '1965', '1970', '1975', '
1980']
df2 = pd.DataFrame(x2)
df2.columns = ['1985', '1990', '1995', '2000', '2005', '2010', '
2015']
df3 = pd.DataFrame(x3)
df3.columns = ['2020', '2025', '2030', '2035', '2040', '2045', '
2050']
```

In [14]:

```
df = pd.concat([df1,df2, df3], axis=1)
df['Country'] = country
cols = list(df.columns)
cols = [cols[-1]] +cols[:-1]
df = df[cols]
```

In [15]:

```
dict_pop = df.set_index('Country').T.to_dict()
dict_pop['Albania']
```

Out[15]:


```
{'1950': '1228',
 '1955': '1393',
 '1960': '1624',
 '1965': '1884',
 '1970': '2157',
 '1975': '2402',
 '1980': '2672',
 '1985': '2957',
 '1990': '3245',
 '1995': '3159',
 '2000': '3159',
 '2005': '3025',
 '2010': '2987',
 '2015': '3030',
 '2020': '3075',
 '2025': '3105',
 '2030': '3103',
 '2035': '3063',
 '2040': '2994',
 '2045': '2913',
 '2050': '2825'}
```


## 1.5 Create a dataframe from your dictionary (10 pts)

Now that all tables are stored in a dictionary, we can convert the dictionary into a pandas dataframe.

1. Remove the "World" row
2. Replace 'NaN' values with 0
3. Display the first 8 rows

In [59]:

```
# Your code here
import pandas as pd
df = pd.DataFrame(dict_pop)
```

In [60]:

```
# Your code here
df1 = df.drop(['World'], axis=1)
df1.apply(pd.to_numeric)
df2 = df1.fillna(0)
```

Out[60]:

| | Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguill |
|---|---|---|---|---|---|---|---|
| **1950** | 8151 | 1228 | 8893 | 20 | 7 | 4118 | |
| **1955** | 8892 | 1393 | 9842 | 20 | 7 | 4424 | |
| **1960** | 9830 | 1624 | 10910 | 21 | 9 | 4798 | |
| **1965** | 10998 | 1884 | 11964 | 25 | 14 | 5135 | |
| **1970** | 12431 | 2157 | 13932 | 28 | 20 | 5606 | |
| **1975** | 14133 | 2402 | 16141 | 30 | 27 | 6051 | |
| **1980** | 15045 | 2672 | 18807 | 33 | 34 | 7206 | |
| **1985** | 13120 | 2957 | 22009 | 39 | 45 | 8390 | |
| **1990** | 13569 | 3245 | 25191 | 48 | 53 | 9486 | |
| **1995** | 19446 | 3159 | 28322 | 54 | 64 | 11000 | 1 |
| **2000** | 22462 | 3159 | 30639 | 58 | 66 | 12683 | 1 |
| **2005** | 26335 | 3025 | 32918 | 57 | 77 | 14770 | 1 |
| **2010** | 29121 | 2987 | 35950 | 56 | 85 | 17043 | 1 |
| **2015** | 32565 | 3030 | 39543 | 55 | 86 | 19626 | 1 |
| **2020** | 36644 | 3075 | 42973 | 54 | 86 | 22485 | 1 |
| **2025** | 41118 | 3105 | 45842 | 54 | 86 | 25674 | 2 |
| **2030** | 45665 | 3103 | 48149 | 53 | 85 | 29155 | 2 |
| **2035** | 50195 | 3063 | 50118 | 52 | 83 | 32910 | 2 |
| **2040** | 54717 | 2994 | 52030 | 51 | 82 | 36948 | 2 |
| **2045** | 59256 | 2913 | 53894 | 50 | 79 | 41280 | 2 |
| **2050** | 63796 | 2825 | 55445 | 50 | 75 | 45889 | 2 |

21 rows × 227 columns

In [61]:

```
df2.head(8)
```

Out[61]:

| | Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguill |
|---|---|---|---|---|---|---|---|
| **1950** | 8151 | 1228 | 8893 | 20 | 7 | 4118 | |
| **1955** | 8892 | 1393 | 9842 | 20 | 7 | 4424 | |
| **1960** | 9830 | 1624 | 10910 | 21 | 9 | 4798 | |
| **1965** | 10998 | 1884 | 11964 | 25 | 14 | 5135 | |
| **1970** | 12431 | 2157 | 13932 | 28 | 20 | 5606 | |
| **1975** | 14133 | 2402 | 16141 | 30 | 27 | 6051 | |
| **1980** | 15045 | 2672 | 18807 | 33 | 34 | 7206 | |
| **1985** | 13120 | 2957 | 22009 | 39 | 45 | 8390 | |

8 rows × 227 columns

***Part 2. Exploring the data***

Now let's look at the data at hand.

## 2.1 Plotting population (15 pts)

Pick 6 countries of your choice and plot their population growth.

In [73]:

```
# Your code here
import matplotlib.pyplot as plt
plt.figure(figsize=(15,15))
plt.subplot(3,2,1)
plt.plot(df2['Malta'], color='pink')
plt.title('Population Growth in Malta')
plt.xlabel('Years')
plt.ylabel('Population in thousands')
plt.subplot(3,2,2)
plt.plot(df2['India'], color='green')
plt.title('Population Growth in India')
plt.xlabel('Years')
plt.ylabel('Population in thousands')
plt.subplot(3,2,3)
plt.plot(df2['Bhutan'], color='orange')
plt.title('Population Growth in Bhutan')
plt.xlabel('Years')
plt.ylabel('Population in thousands')
plt.subplot(3,2,4)
plt.plot(df2['Greenland'], color='red')
plt.title('Population Growth in Greenland')
plt.xlabel('Years')
plt.ylabel('Population in thousands')
plt.subplot(3,2,5)
plt.plot(df2['United States'], color='black')
plt.title('Population Growth in United States')
plt.xlabel('Years')
plt.ylabel('Population in thousands')
plt.subplot(3,2,6)
plt.plot(df2['Norway'])
plt.title('Population Growth in Norway')
plt.xlabel('Years')
plt.ylabel('Population in thousands')
plt.show()
```

Out[73]:

```
<Figure size 1080x1080 with 0 Axes>
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11752371
0>
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x1179292e8>]
```

Out[73]:

```
Text(0.5, 1.0, 'Population Growth in Malta')
```

Out[73]:

```
Text(0.5, 0, 'Years')
```

Out[73]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x117905be
0>
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x117950d68>]
```

Out[73]:

```
Text(0.5, 1.0, 'Population Growth in India')
```

Out[73]:

```
Text(0.5, 0, 'Years')
```

Out[73]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x117950c1
8>
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x1179814a8>]
```

Out[73]:

```
Text(0.5, 1.0, 'Population Growth in Bhutan')
```

Out[73]:

```
Text(0.5, 0, 'Years')
```

Out[73]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1179815c
0>
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x1179aadd8>]
```

Out[73]:

```
Text(0.5, 1.0, 'Population Growth in Greenland')
```

Out[73]:

```
Text(0.5, 0, 'Years')
```

Out[73]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1179aaef
0>
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x1179dd860>]
```

Out[73]:

```
Text(0.5, 1.0, 'Population Growth in United States')
```

Out[73]:

```
Text(0.5, 0, 'Years')
```

Out[73]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1179dd97
8>
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x117a0e128>]
```

Out[73]:

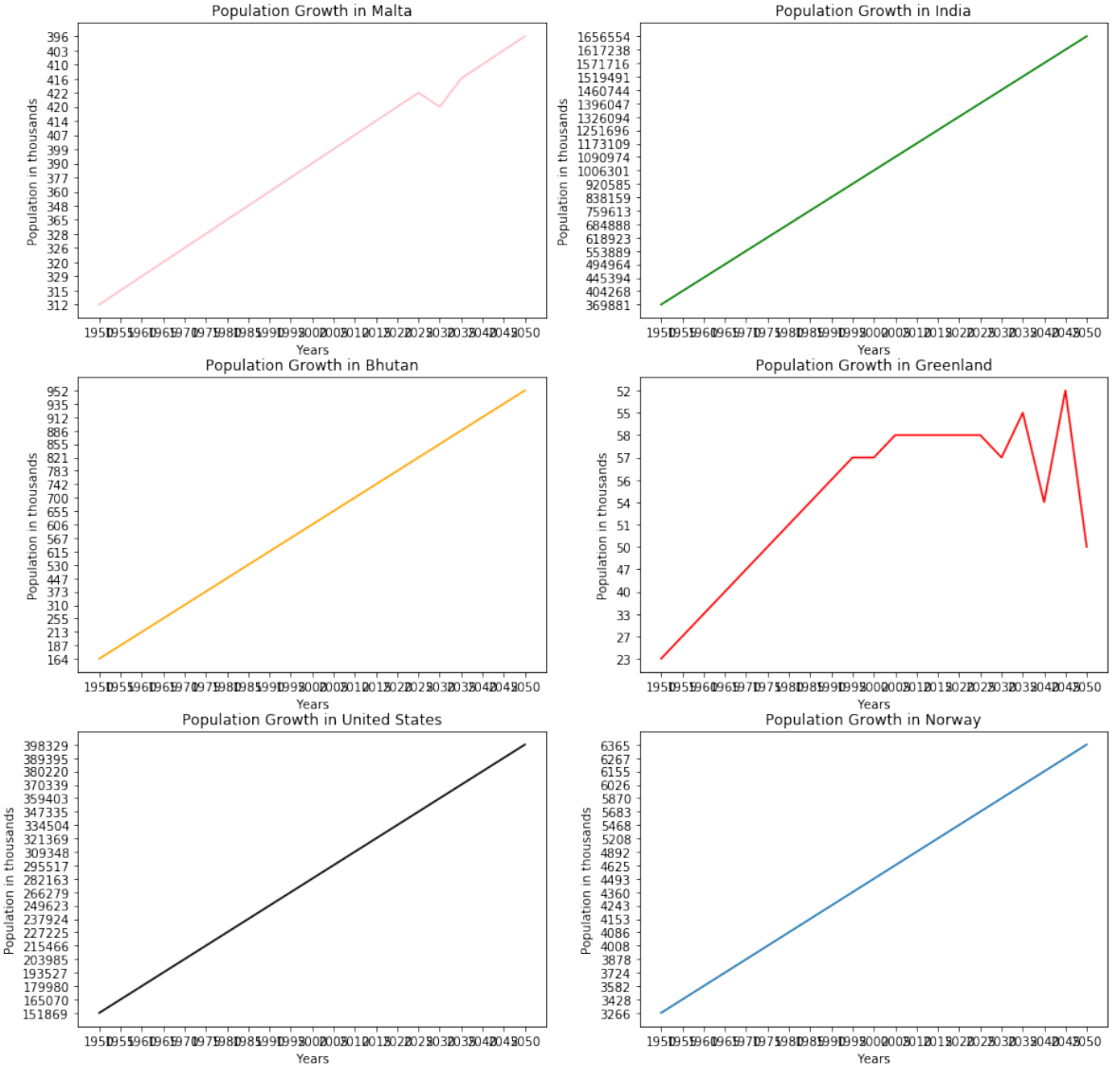
```
Text(0.5, 1.0, 'Population Growth in Norway')
```

Out[73]:

```
Text(0.5, 0, 'Years')
```

Out[73]:

```
Text(0, 0.5, 'Population in thousands')
```



## 2.2 Find 10 most populous countries ( 15 pts)

Find 10 most popoulous coutntries in 1960, 1980, 2000, 2020, and 2040. Plot and compare their population.

In [122]:

```
# Your code here
df2 = df2.transpose()
```

In [134]:

```
y1960 = pd.to_numeric(df2['1960'])
topy1960 = y1960.nlargest(10)
topy1960
```

Out[134]:

```
China             651340
India             445394
United States     179980
Russia            119632
Indonesia         100146
Japan              94092
Germany            72481
Brazil             71412
Bangladesh         54593
United Kingdom     52373
Name: 1960, dtype: int64
```

In [136]:

```
y1980 = pd.to_numeric(df2['1980'])
topy1980 = y1980.nlargest(10)
topy1980
y2000 = pd.to_numeric(df2['2000'])
topy2000= y2000.nlargest(10)
topy2000
y2020 = pd.to_numeric(df2['2020'])
topy2020= y2020.nlargest(10)
topy2020
y2040 = pd.to_numeric(df2['2040'])
topy2040 = y2040.nlargest(10)
topy2040
```

Out[136]:

```
China            987822
India            684888
United States    227225
Indonesia        150322
Russia           139039
Brazil           121064
Japan            116808
Bangladesh        87938
Pakistan          85220
Germany           78298
Name: 1980, dtype: int64
```

Out[136]:

```
China            1268302
India            1006301
United States     282163
Indonesia         214091
Brazil            174316
Pakistan          152430
Russia            147054
Bangladesh        132151
Japan             126776
Nigeria           123946
Name: 2000, dtype: int64
```

Out[136]:

```
China            1397026
India            1326094
United States     334504
Indonesia         279080
Brazil            216016
Pakistan          213720
Nigeria           204950
Bangladesh        169778
Russia            145723
Mexico            128650
Name: 2020, dtype: int64
```

Out[136]:

```
India            1571716
China            1428383
United States     380220
Nigeria           322187
Indonesia         314085
Pakistan          269152
Brazil            236077
Bangladesh        196224
Ethiopia          187611
Mexico            147495
Name: 2040, dtype: int64
```

In [157]:

```
plt.figure(figsize=(15,15))
plt.subplot(5,1,1)
plt.plot(pd.to_numeric(df2['1960']).nlargest(10), marker='o', li
newidth=0, color='red')
plt.title('10 populous countries in 1960')
plt.ylabel('Population in thousands')
plt.xlabel('Countries')
plt.subplot(5,1,2)
plt.plot(pd.to_numeric(df2['1980']).nlargest(10), marker='o', li
newidth=0, color='red')
plt.title('10 populous countries in 1980')
plt.ylabel('Population in thousands')
plt.xlabel('Countries')
plt.subplot(5,1,3)
plt.plot(pd.to_numeric(df2['2000']).nlargest(10), marker='o', li
newidth=0, color='red')
plt.title('10 populous countries in 2000')
plt.ylabel('Population in thousands')
plt.xlabel('Countries')
plt.subplot(5,1,4)
plt.plot(pd.to_numeric(df2['2020']).nlargest(10), marker='o', li
newidth=0, color='red')
plt.title('10 populous countries in 2020')
plt.ylabel('Population in thousands')
plt.xlabel('Countries')
plt.subplot(5,1,5)
plt.plot(pd.to_numeric(df2['2040']).nlargest(10), marker='o', li
newidth=0, color='red')
plt.title('10 populous countries in 2040')
plt.ylabel('Population in thousands')
plt.xlabel('Countries')
plt.subplots_adjust(hspace = 0.5)
plt.show()
```

Out[157]:

```
<Figure size 1080x1080 with 0 Axes>
```

Out[157]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11a9f1da
0>
```

Out[157]:

```
[<matplotlib.lines.Line2D at 0x114b87780>]
```

Out[157]:

```
Text(0.5, 1.0, '10 populous countries in 1960')
```

Out[157]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[157]:

```
Text(0.5, 0, 'Countries')
```

Out[157]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x114b8766
8>
```

Out[157]:

```
[<matplotlib.lines.Line2D at 0x11aa40eb8>]
```

Out[157]:

```
Text(0.5, 1.0, '10 populous countries in 1980')
```

Out[157]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[157]:

```
Text(0.5, 0, 'Countries')
```

Out[157]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11aa3089
8>
```

Out[157]:

```
[<matplotlib.lines.Line2D at 0x11abbc780>]
```

Out[157]:

```
Text(0.5, 1.0, '10 populous countries in 2000')
```

Out[157]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[157]:

```
Text(0.5, 0, 'Countries')
```

Out[157]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11abad19
8>
```

Out[157]:

```
[<matplotlib.lines.Line2D at 0x11abf1128>]
```

Out[157]:

```
Text(0.5, 1.0, '10 populous countries in 2020')
```

Out[157]:

```
Text(0, 0.5, 'Population in thousands')
```

Out[157]:

```
Text(0.5, 0, 'Countries')
```

Out[157]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11abf11d
0>
```

Out[157]:

```
[<matplotlib.lines.Line2D at 0x11ac189b0>]
```

Out[157]:

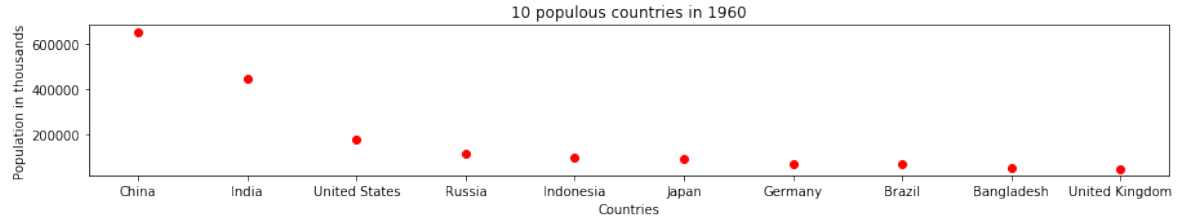
```
Text(0.5, 1.0, '10 populous countries in 2040')
```

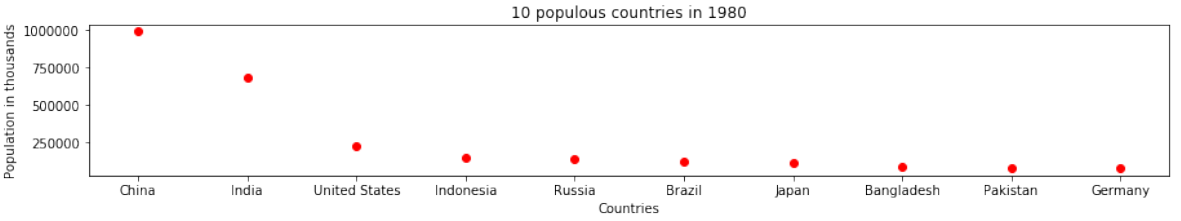
Out[157]:

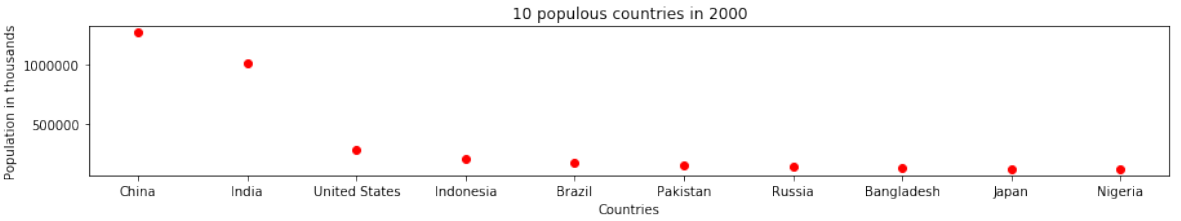
```
Text(0, 0.5, 'Population in thousands')
```

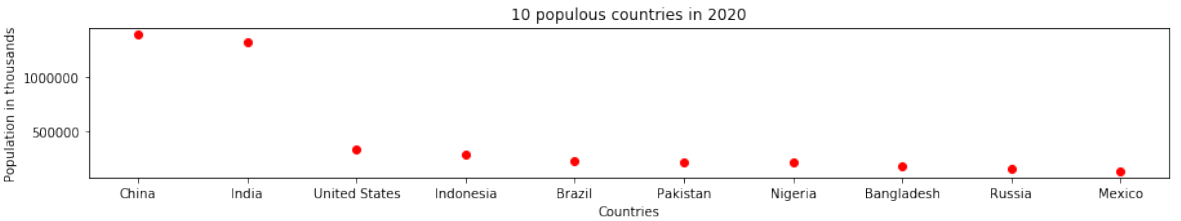
Out[157]:

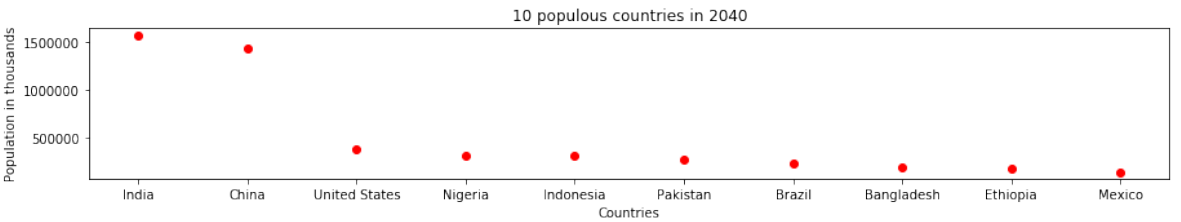
```
Text(0.5, 0, 'Countries')
```




In [159]:

```
print('India, China and United States have been the top 3 counti
res with large population and do not seem to change in the comin
g years')
```

```
India, China and United States have been the top 3 c
ountires with large population and do not seem to ch
ange in the coming years
```

In [160]:

```
print('Of the 10 countries Mexico will be the least populated')
```

```
Of the 10 countries Mexico will be the least populat
ed
```