# NAME : RETHINAGIRI G

# ROLL NO :225229130

## COURSE TITTLE : DATA AND VISUAL ANALYTICS LAB

## Lab.05 Pandas Concatenate, Merge and Join

In [1]:
```
import pandas as pd
```

In [2]:
```
north_america=pd.read_csv("C:/Users/user/Downloads/oecd/north_america_2000_2010.csv",index_col=0)
```

In [3]:
```
south_america=pd.read_csv("C:/Users/user/Downloads/oecd/south_america_2000_2010.csv",index_col=0)
```

In [4]:
```
north_america
```

Out[4]:

|  | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Country** |  |  |  |  |  |  |  |  |  |  |  |
| **Canada** | 1779.0 | 1771.0 | 1754.0 | 1740.0 | 1760.0 | 1747 | 1745.0 | 1741.0 | 1735 | 1701.0 | 1703.0 |
| **Mexico** | 2311.2 | 2285.2 | 2271.2 | 2276.5 | 2270.6 | 2281 | 2280.6 | 2261.4 | 2258 | 2250.2 | 2242.4 |
| **USA** | 1836.0 | 1814.0 | 1810.0 | 1800.0 | 1802.0 | 1799 | 1800.0 | 1798.0 | 1792 | 1767.0 | 1778.0 |

```
In [5]: south_america
```

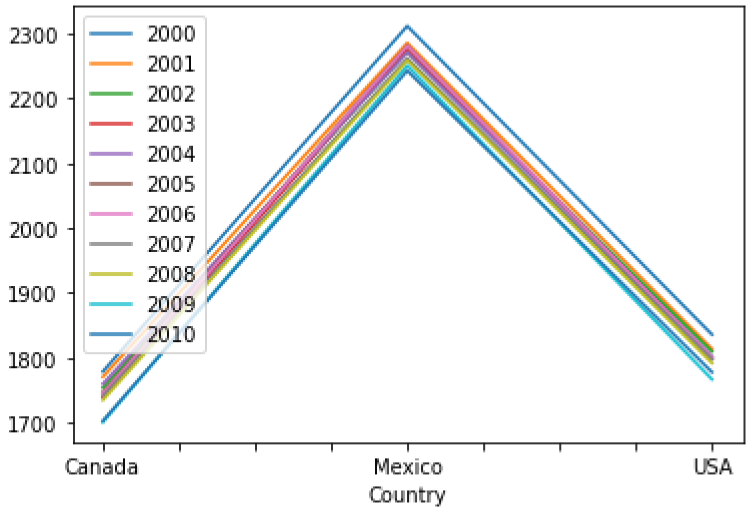
Out[5]:

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | | | | | | | | | | | |
| **Chile** | 2263 | 2242 | 2250 | 2235 | 2232 | 2157 | 2165 | 2128 | 2095 | 2074 | 2069.6 |

**create line graphs for our yearly labor trends in north_america¶**

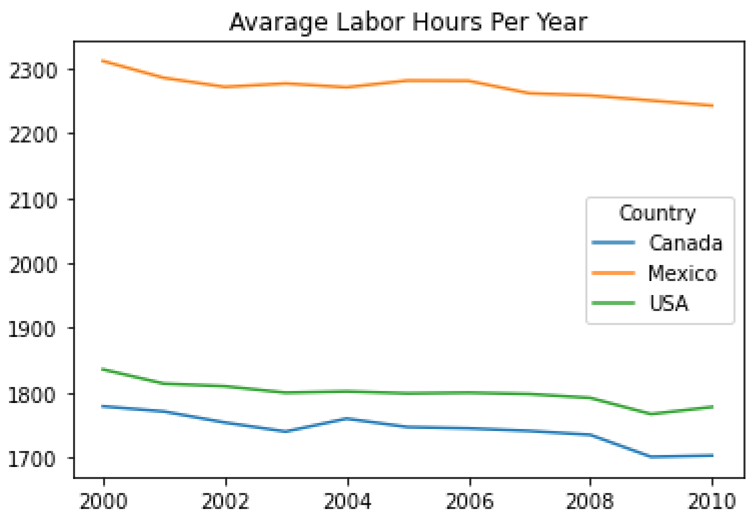
```
In [6]: north_america.plot()
```

```
Out[6]: <AxesSubplot:xlabel='Country'>
```



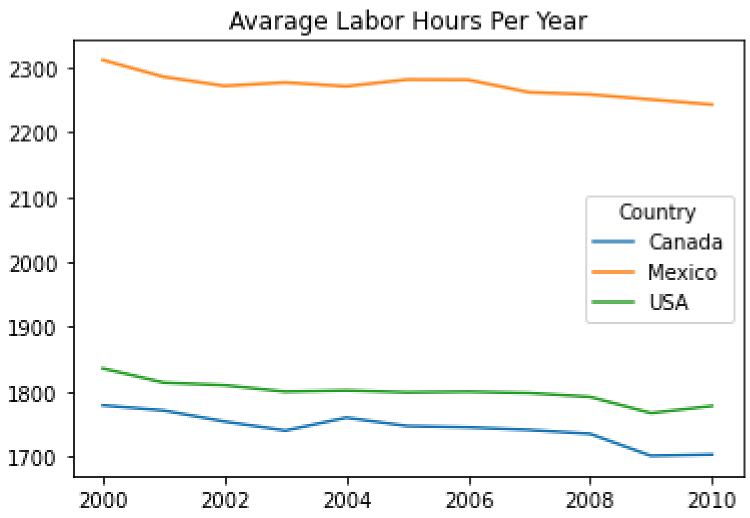
```
In [7]: north_america.transpose().plot(title="Avarage Labor Hours Per Year")
```

```
Out[7]: <AxesSubplot:title={'center':'Avarage Labor Hours Per Year'}>
```



```
In [8]: north_america.transpose().plot(title="Avarage Labor Hours Per Year")
```

```
Out[8]: <AxesSubplot:title={'center':'Avarage Labor Hours Per Year'}>
```



## Concatenate America Data

It’s hard to compare the average labor hours in South America versus North America. If we were able to get all the countries into the same data frame, it would be much easier to do this camparison.

#### Concatenate north_america and south_america dataframes and store result in a dataframe,americas

In [9]:

```
Americas=pd.concat([north_america,south_america])
Americas
```

Out[9]:

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | | | | | | | | | | | |
| **Canada** | 1779.0 | 1771.0 | 1754.0 | 1740.0 | 1760.0 | 1747 | 1745.0 | 1741.0 | 1735 | 1701.0 | 1703.0 |
| **Mexico** | 2311.2 | 2285.2 | 2271.2 | 2276.5 | 2270.6 | 2281 | 2280.6 | 2261.4 | 2258 | 2250.2 | 2242.4 |
| **USA** | 1836.0 | 1814.0 | 1810.0 | 1800.0 | 1802.0 | 1799 | 1800.0 | 1798.0 | 1792 | 1767.0 | 1778.0 |
| **Chile** | 2263.0 | 2242.0 | 2250.0 | 2235.0 | 2232.0 | 2157 | 2165.0 | 2128.0 | 2095 | 2074.0 | 2069.6 |

Now, our data collection team has sent us data files for each year from 2011 to 2015 in separate CSV files. They are americas_2011.csv , americas_2012.csv, americas_2014.csv and americas_2015.csv

#### Load the additional files

In [10]:

```
americas_dfs=[Americas]

for year in range(2011,2016):
    file_name="C:\\Users\\user\\Downloads\\oecd\\americas_{}.csv".format(year)
    df=pd.read_csv(file_name,index_col=0)
    americas_dfs.append(df)
```

```
In [11]: americas_dfs[1]
```

```
Out[11]:
                2011
         Country
          Canada   1700.0
           Chile   2047.4
          Mexico   2250.2
             USA   1786.0
```

```
In [12]: americas_dfs[2]
```

```
Out[12]:
                2012
         Country
          Canada   1713.0
           Chile   2024.0
          Mexico   2225.8
             USA   1789.0
```

### Concatenate americas and americas_dfs dataframes and store result in americas

```
In [13]: americas=pd.concat(americas_dfs,axis=1)
```

In [14]:
```
americas.index.names=["country"]
americas
```

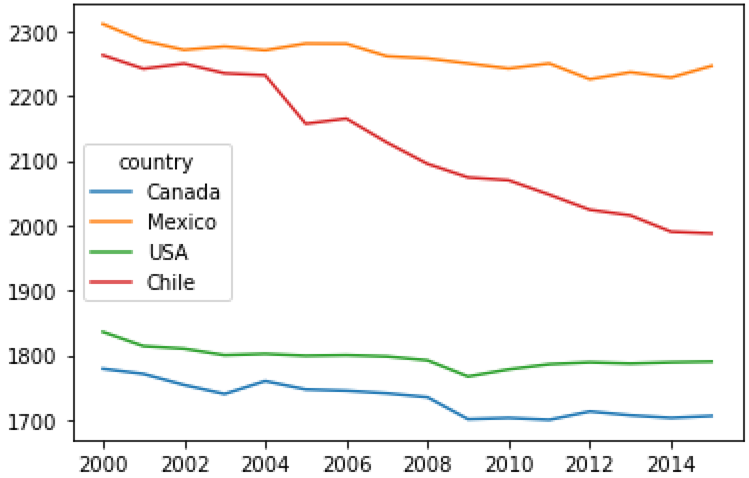
Out[14]:

| country | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Canada** | 1779.0 | 1771.0 | 1754.0 | 1740.0 | 1760.0 | 1747 | 1745.0 | 1741.0 | 1735 | 1701.0 | 1703.0 | 1700.0 | 1713.0 | 1707.0 | 1703.0 | 1706.0 |
| **Mexico** | 2311.2 | 2285.2 | 2271.2 | 2276.5 | 2270.6 | 2281 | 2280.6 | 2261.4 | 2258 | 2250.2 | 2242.4 | 2250.2 | 2225.8 | 2236.6 | 2228.4 | 2246.4 |
| **USA** | 1836.0 | 1814.0 | 1810.0 | 1800.0 | 1802.0 | 1799 | 1800.0 | 1798.0 | 1792 | 1767.0 | 1778.0 | 1786.0 | 1789.0 | 1787.0 | 1789.0 | 1790.0 |
| **Chile** | 2263.0 | 2242.0 | 2250.0 | 2235.0 | 2232.0 | 2157 | 2165.0 | 2128.0 | 2095 | 2074.0 | 2069.6 | 2047.4 | 2024.0 | 2015.3 | 1990.1 | 1987.5 |

#### Now, plot transposed americas dataframe¶

In [15]:
```
americas.transpose().plot()
```

Out[15]:
```
<AxesSubplot:>
```



#### Appending data from other Continents

The data collection team has provided CSV files for Asia, Europe, and the South Pacific for 2000 through 2015. Let’s load these files in and have a preview

In [16]:

```
asia=pd.read_csv("C:\\Users\\user\\Downloads\\oecd\\asia_2000_2015.csv",index_col=0)
asia
```

Out[16]:

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | | | | | | | | | | | | | | | | |
| **Israel** | 2017 | 1979 | 1993 | 1974 | 1942 | 1931 | 1919 | 1931 | 1929 | 1927 | 1918 | 1920 | 1910 | 1867 | 1853 | 1858 |
| **Japan** | 1821 | 1809 | 1798 | 1799 | 1787 | 1775 | 1784 | 1785 | 1771 | 1714 | 1733 | 1728 | 1745 | 1734 | 1729 | 1719 |
| **Korea** | 2512 | 2499 | 2464 | 2424 | 2392 | 2351 | 2346 | 2306 | 2246 | 2232 | 2187 | 2090 | 2163 | 2079 | 2124 | 2113 |
| **Russia** | 1982 | 1980 | 1982 | 1993 | 1993 | 1989 | 1998 | 1999 | 1997 | 1974 | 1976 | 1979 | 1982 | 1980 | 1985 | 1978 |

In [17]:

```
europe=pd.read_csv("C:\\Users\\user\\Downloads\\oecd\\europe_2000_2015.csv",index_col=0)
europe.head()
```

Out[17]:

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | | | | | | | | | | | | | | | | |
| **Austria** | 1807.4 | 1794.6 | 1792.2 | 1783.8 | 1786.8 | 1764.0 | 1746.2 | 1736.0 | 1728.5 | 1673.0 | 1668.6 | 1675.9 | 1652.9 | 1636.7 | 1629.4 | 1624.9 |
| **Belgium** | 1595.0 | 1588.0 | 1583.0 | 1578.0 | 1573.0 | 1565.0 | 1572.0 | 1577.0 | 1570.0 | 1548.0 | 1546.0 | 1560.0 | 1560.0 | 1558.0 | 1560.0 | 1541.0 |
| **Switzerland** | 1673.6 | 1635.0 | 1614.0 | 1626.8 | 1656.5 | 1651.7 | 1643.2 | 1632.7 | 1623.1 | 1614.9 | 1612.4 | 1605.4 | 1590.9 | 1572.9 | 1568.3 | 1589.7 |
| **Czech Republic** | 1896.0 | 1818.0 | 1816.0 | 1806.0 | 1817.0 | 1817.0 | 1799.0 | 1784.0 | 1790.0 | 1779.0 | 1800.0 | 1806.0 | 1776.0 | 1763.0 | 1771.0 | 1779.0 |
| **Germany** | 1452.0 | 1441.9 | 1430.9 | 1424.8 | 1422.2 | 1411.3 | 1424.7 | 1424.4 | 1418.4 | 1372.7 | 1389.9 | 1392.8 | 1375.3 | 1361.7 | 1366.4 | 1371.0 |

In [18]:

```
south_pacific=pd.read_csv("C:\\Users\\user\\Downloads\\oecd\\south_pacific_2000_2015.csv",index_col=0)
south_pacific.head()
```

Out[18]:

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Country** | | | | | | | | | | | | | | | | |
| **Australia** | 1778.7 | 1736.7 | 1731.7 | 1735.8 | 1734.5 | 1729.2 | 1720.5 | 1712.5 | 1717.2 | 1690 | 1691.5 | 1699.5 | 1678.6 | 1662.7 | 1663.6 | 1665 |
| **New Zealand** | 1836.0 | 1825.0 | 1826.0 | 1823.0 | 1830.0 | 1815.0 | 1795.0 | 1774.0 | 1761.0 | 1740 | 1755.0 | 1746.0 | 1734.0 | 1752.0 | 1762.0 | 1757 |

**Append asia, europe and south_pacific to americas dataframe and assign to new dataframe world**

In [19]:

```
world=americas.append([asia,europe,south_pacific])
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_26408\2482502302.py:1: FutureWarning: The frame.append method is deprecated
and will be removed from pandas in a future version. Use pandas.concat instead.
  world=americas.append([asia,europe,south_pacific])
```

In [20]:

```
world.index
```

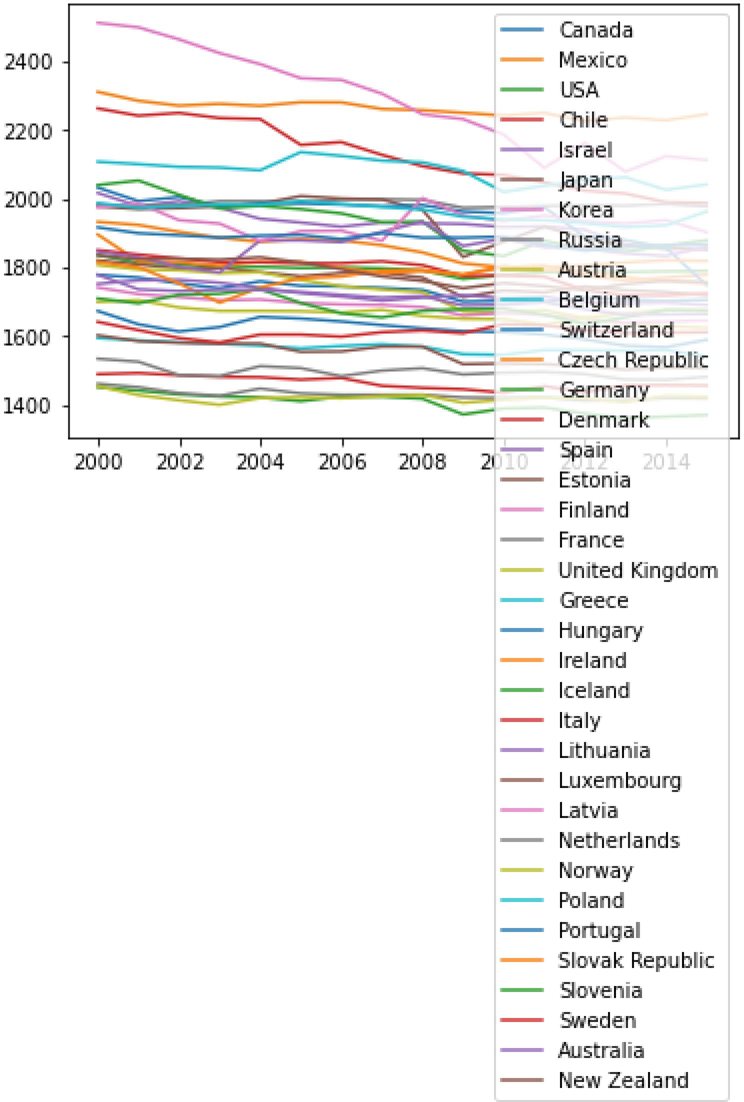
Out[20]:

```
Index(['Canada', 'Mexico', 'USA', 'Chile', 'Israel', 'Japan', 'Korea',
       'Russia', 'Austria', 'Belgium', 'Switzerland', 'Czech Republic',
       'Germany', 'Denmark', 'Spain', 'Estonia', 'Finland', 'France',
       'United Kingdom', 'Greece', 'Hungary', 'Ireland', 'Iceland', 'Italy',
       'Lithuania', 'Luxembourg', 'Latvia', 'Netherlands', 'Norway', 'Poland',
       'Portugal', 'Slovak Republic', 'Slovenia', 'Sweden', 'Australia',
       'New Zealand'],
      dtype='object')
```

### Plot, transposed world dataframe¶

In [21]:
```
world.transpose().plot()
```

Out[21]:
```
<AxesSubplot:>
```



**let us customize this plot, so that country names appear outside the chart**

Update plot() with the following features

figsize=(10,10),

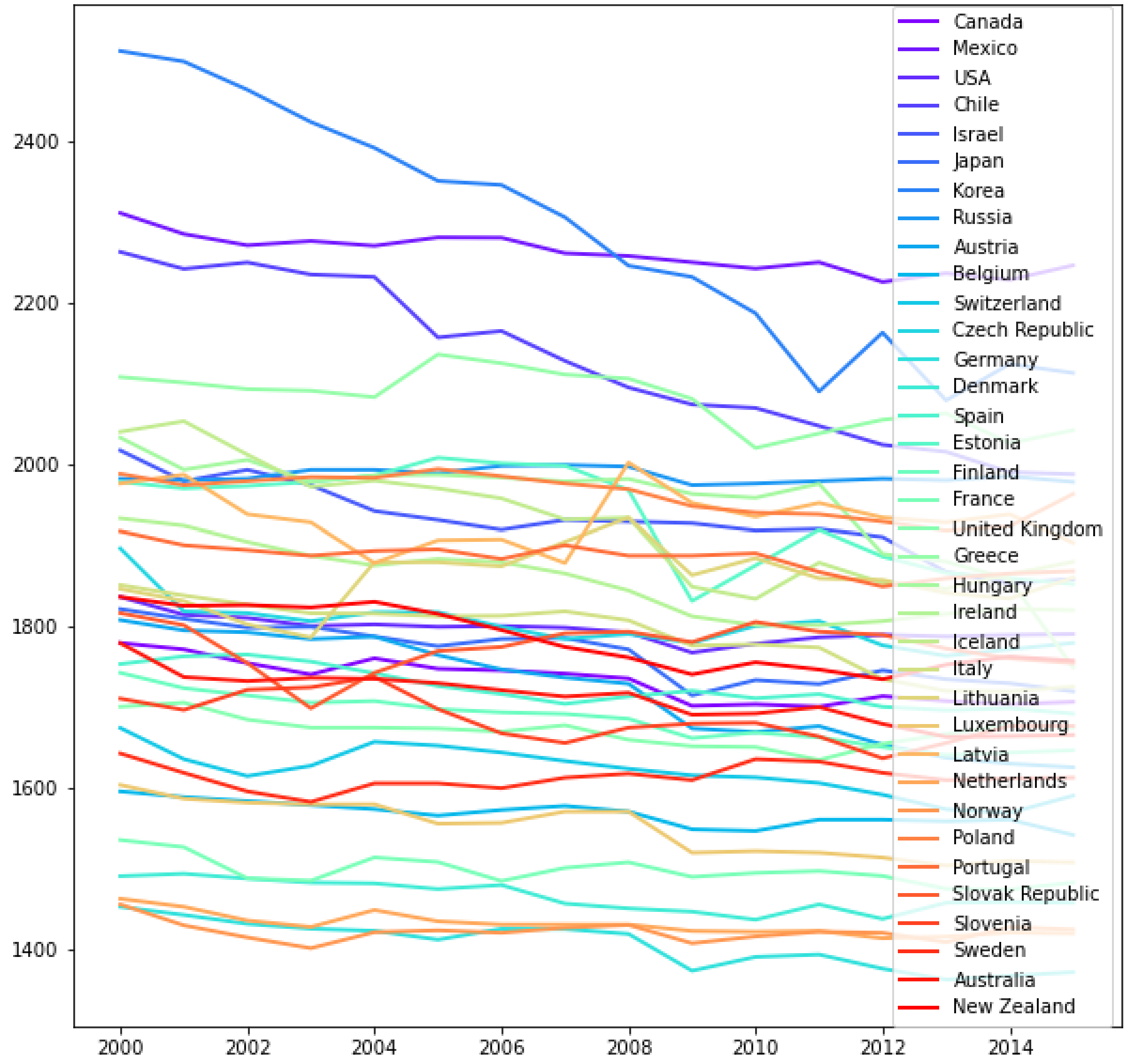
colormap='rainbow',

linewidth=2,

loc='right'

```
In [22]: world.transpose().plot(figsize=(10,10),colormap="rainbow",linewidth=2).legend(loc="right")
```

```
Out[22]: <matplotlib.legend.Legend at 0x117b9231190>
```



### Merging Historical Labor Data

It's nice being able to see how the labor hours have shifted since 2000, but in order to see real trends emerge, we want to be able to see as much historical data as possible. The data collection team was kind enough to send data from 1950 to 2000, let's load it in and take a look.

In [23]:

```
historical=pd.read_csv("C:\\Users\\user\\Downloads\\oecd\\historical.csv")
historical.head()
```

Out[23]:

| | Country | 1950 | 1951 | 1952 | 1953 | 1954 | 1955 | 1956 | 1957 | 1958 | ... | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Australia | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1779.5 | 1774.90 | 1773.70 | 1786.50 | 1797.60 | 1793.400 | 1782.700 | 1783.600 |
| **1** | Austria | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | 1619.200 | 1637.150 | 1648.500 |
| **2** | Belgium | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1662.9 | 1625.79 | 1602.72 | 1558.59 | 1558.59 | 1515.835 | 1500.295 | 1510.315 |
| **3** | Canada | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1789.5 | 1767.50 | 1766.00 | 1764.50 | 1773.00 | 1771.500 | 1786.500 | 1782.500 |
| **4** | Switzerland | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | 1673.10 | 1684.80 | 1685.80 | 1706.20 | 1685.500 | 1658.900 | 1648.600 |

5 rows × 51 columns

In [24]:

```
print(world.shape,"world rows & columns")
print(historical.shape,"Historical rows & columns")
```

```
(36, 16) world rows & columns
(39, 51) Historical rows & columns
```

Note that the historical table has 39 rows, even though we are only analyzing 36 countries in our world table. Dropping the three extra rows can be automatically taken care of with some proper DataFrame merging. We will treat world as our primary table and want this to be on the right side of the resulting DataFrame and historical on the left, so the years (columns) stay in chronological order. The columns in these two tables are all distinct, that means we will have to find a key to join on. In this case, the key will be the row indexes (countries). We will want to do a right join using the pd.merge() function and use the indexes as keys to join on.

The right join will ensure we only keep the 36 rows from the right table and discard the extra 3 from the historical table. Let's print the shape of the resulting DataFrame and display the head to make sure everything turned out correct.

**Merge historical dataframe with world dataframe and store in a new variable, world_historical**

In [25]:
```
world_historical=pd.merge(world,historical,left_index=True,right_index=True,how="right")
```

**Print size of world_historical dataframe¶**

In [26]:
```
world_historical.shape
```

Out[26]:
```
(39, 67)
```

**Print top-5 of world_historical dataframe¶**

In [27]:
```
world_historical.head(5)
```

Out[27]:

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | ... | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1779.5 | 1774.90 | 1773.70 | 1786.50 | 1797.60 | 1793.400 | 1782.700 | 1783.600 | 1768. |
| **1** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | 1619.200 | 1637.150 | 1648.500 | 1641. |
| **2** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1662.9 | 1625.79 | 1602.72 | 1558.59 | 1558.59 | 1515.835 | 1500.295 | 1510.315 | 1513. |
| **3** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 1789.5 | 1767.50 | 1766.00 | 1764.50 | 1773.00 | 1771.500 | 1786.500 | 1782.500 | 1778. |
| **4** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | 1673.10 | 1684.80 | 1685.80 | 1706.20 | 1685.500 | 1658.900 | 1648.600 | 1656. |

5 rows × 67 columns

## Joining Historical Data

Now that we’ve done it the hard way and understand table merging conceptually, let’s try a more elegant technique. Pandas has a clean method to join on indexes which is perfect for our situation.

**Use join method to join historical dataframe and world dataframe and store result in world_historical dataframe**

In [28]:
```
world_historical=historical.join(world,how='right')
```

```
In [29]: world_historical.head()
```

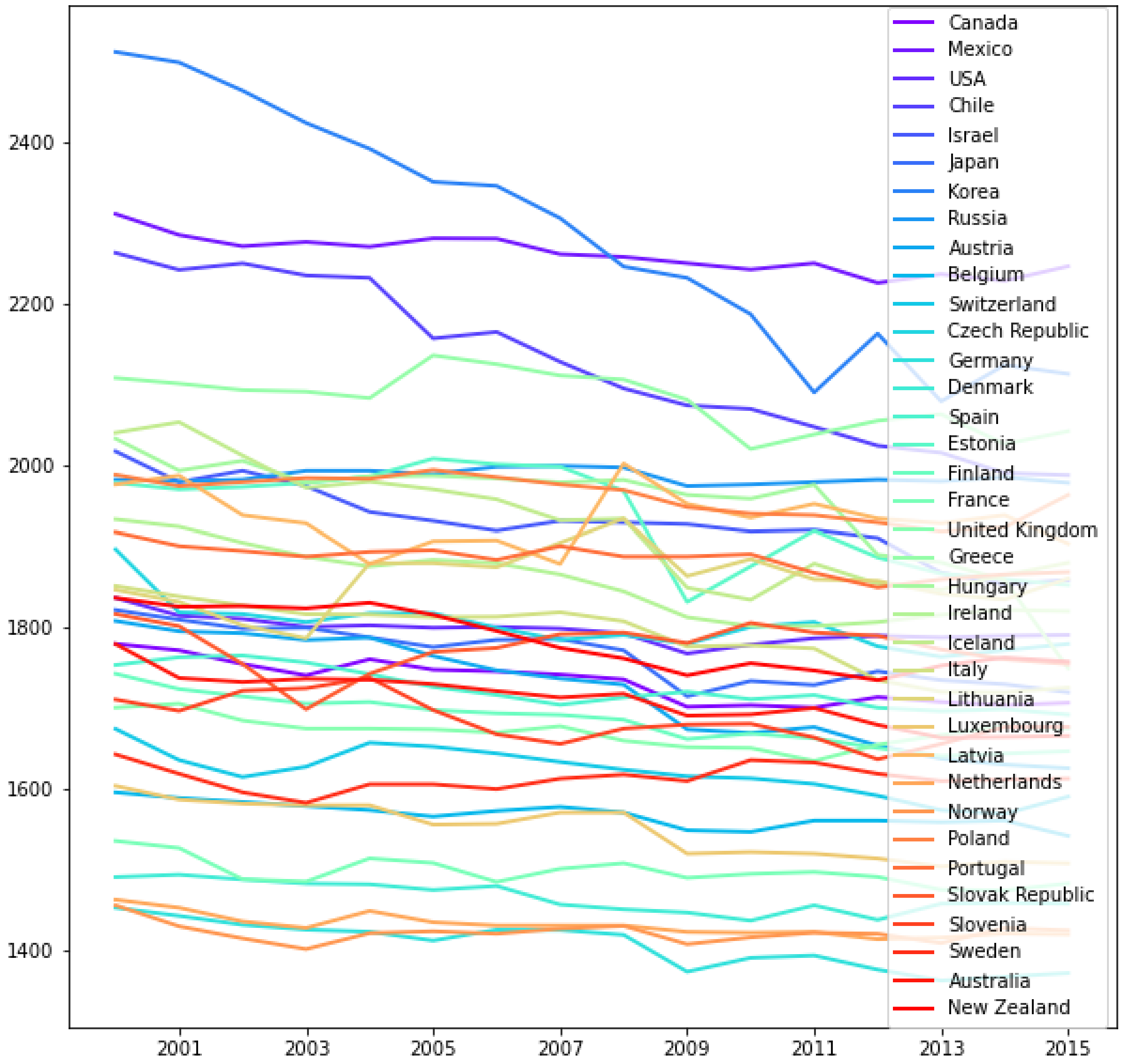
```
Out[29]:
         Country 1950 1951 1952 1953 1954 1955 1956 1957 1958 ...   2006   2007   2008   2009   2010   2011   2012   2013   2014
  Canada          NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN ... 1745.0 1741.0 1735.0 1701.0 1703.0 1700.0 1713.0 1707.0 1703.0
  Mexico          NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN ... 2280.6 2261.4 2258.0 2250.2 2242.4 2250.2 2225.8 2236.6 2228.4
     USA          NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN ... 1800.0 1798.0 1792.0 1767.0 1778.0 1786.0 1789.0 1787.0 1789.0
   Chile          NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN ... 2165.0 2128.0 2095.0 2074.0 2069.6 2047.4 2024.0 2015.3 1990.1
  Israel          NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN ... 1919.0 1931.0 1929.0 1927.0 1918.0 1920.0 1910.0 1867.0 1853.0

5 rows × 67 columns
```

### Plot our world labor data

Before plotting the final line graph, it’s a good idea to sort our rows alphabetically to make the legend more easy to read for our viewers. This can be executed with the DataFrame.sort_index() method. We can pass in the parameter inplace=True to avoid having to reassign our world_historical variable.

```
In [30]: world_historical.transpose().plot(figsize=(10,10),colormap="rainbow",linewidth=2).legend(loc="right")
```

```
Out[30]: <matplotlib.legend.Legend at 0x117b75b1d90>
```



**Which country worked longer hours per year?**

```
In [31]: world_historical.index.max()
```

```
Out[31]: 'United Kingdom'
```

**Which country worked shorter hours per year?**

```
In [32]: world_historical.index.min()
```

```
Out[32]: 'Australia'
```