DVA_lab-5_mahalakshmi18

March 28, 2021

0.0.1 Lab5. Pandas Concatenate, Merge and Join

```
[2]: import pandas as pd            # Import necessary modules
     import matplotlib.pyplot as plt
     %matplotlib inline
```

First column should be used as the row index by passing the argument index_col=0

```
[3]: north_america = pd.read_csv('north_america_2000_2010.csv', index_col=0)
     south_america = pd.read_csv('south_america_2000_2010.csv', index_col=0)
```

```
[4]: north_america        #north_america #(UNCOMMENT AND SEE OUTPUT)
```

```
[4]:           2000    2001    2002    2003    2004  2005    2006    2007  2008  \
     Country
     Canada  1779.0  1771.0  1754.0  1740.0  1760.0  1747  1745.0  1741.0  1735
     Mexico  2311.2  2285.2  2271.2  2276.5  2270.6  2281  2280.6  2261.4  2258
     USA     1836.0  1814.0  1810.0  1800.0  1802.0  1799  1800.0  1798.0  1792

               2009    2010
     Country
     Canada  1701.0  1703.0
     Mexico  2250.2  2242.4
     USA     1767.0  1778.0
```

```
[5]: south_america        #south_america #(UNCOMMENT AND SEE OUTPUT)
```

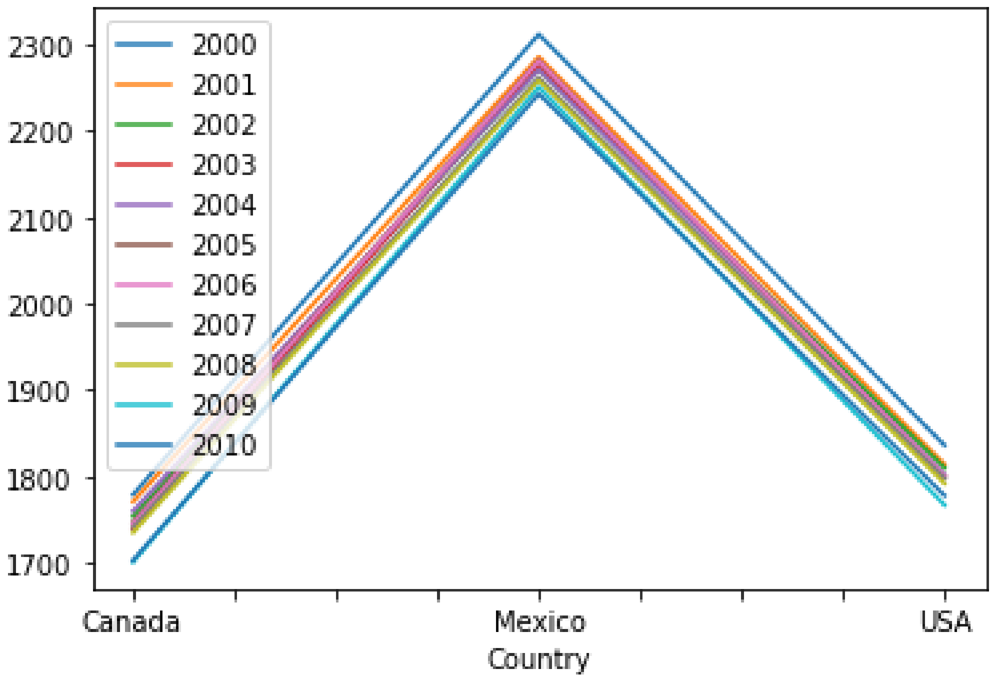
```
[5]:         2000  2001  2002  2003  2004  2005  2006  2007  2008  2009    2010
     Country
     Chile    2263  2242  2250  2235  2232  2157  2165  2128  2095  2074  2069.6
```

Here, rows are countries, columns are years, and cell values are the average annual hours worked per employee.

0.0.2 Create line graphs for our yearly labor trends in north_america

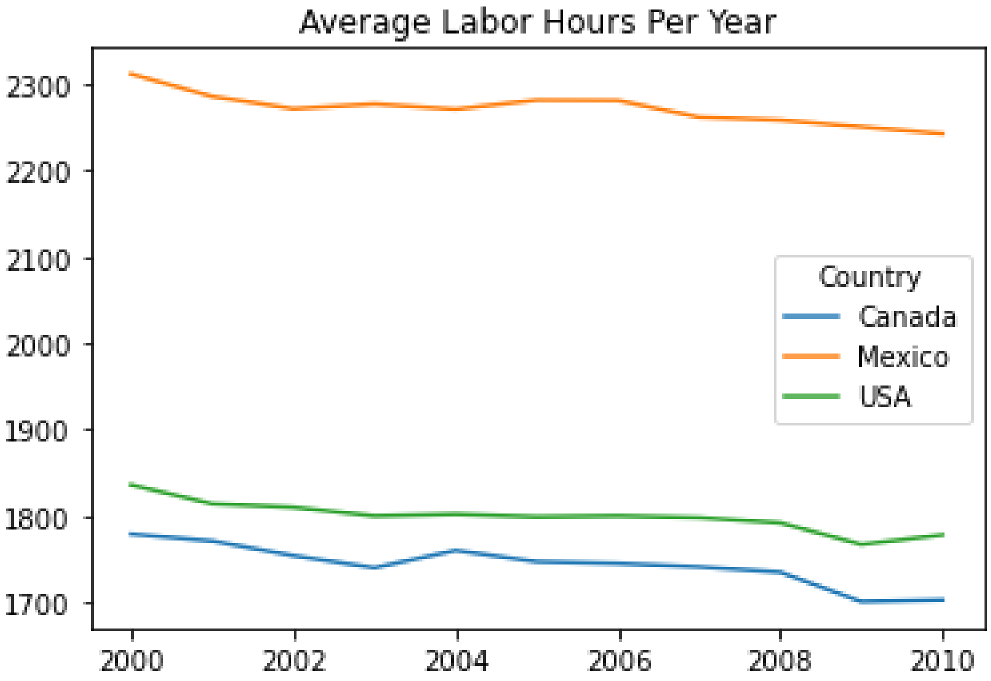
```
[6]: north_america.plot()
```

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



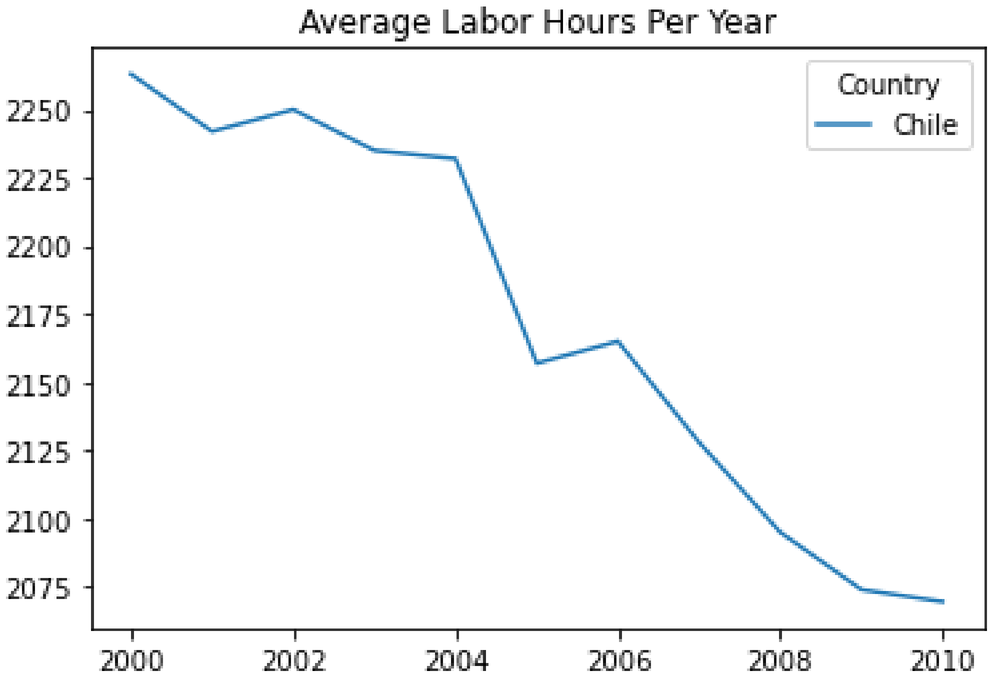
0.0.3 Plot transposed line graph of north_america dataframe, with title "Average Labor Hours Per Year"

[7]:
```
north_america.transpose().plot(title='Average Labor Hours Per Year')
plt.show()
```



0.0.4 Similarly, plot transposed south_america dataframe with title "Average Labor Hours Per Year". Output chart is shown below

[8]:
```
south_america.transpose().plot(title='Average Labor Hours Per Year')
plt.show()
```



0.0.5 Concatenate America DataIt'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.

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

```
[38]: americas = pd.concat([north_america, south_america])
      americas
```

```
[38]:           2000    2001    2002    2003    2004  2005    2006    2007  2008  \
      Country
      Canada  1779.0  1771.0  1754.0  1740.0  1760.0  1747  1745.0  1741.0  1735
      Mexico  2311.2  2285.2  2271.2  2276.5  2270.6  2281  2280.6  2261.4  2258
      USA     1836.0  1814.0  1810.0  1800.0  1802.0  1799  1800.0  1798.0  1792
      Chile   2263.0  2242.0  2250.0  2235.0  2232.0  2157  2165.0  2128.0  2095

                2009    2010
      Country
      Canada  1701.0  1703.0
      Mexico  2250.2  2242.4
      USA     1767.0  1778.0
      Chile   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

0.0.7 Load the additional files

```
[39]: americas_11 = pd.read_csv('americas_2011.csv', index_col=0)
      americas_12 = pd.read_csv('americas_2012.csv', index_col=0)
      americas_13 = pd.read_csv('americas_2013.csv', index_col=0)
      americas_14 = pd.read_csv('americas_2014.csv', index_col=0)
      americas_15 = pd.read_csv('americas_2015.csv', index_col=0)
```

```
[40]: t=americas_11.join(americas_12)
```

```
[41]: t=t.join(americas_13)
```

```
[42]: t=t.join(americas_14)
```

```
[43]: t=t.join(americas_15)
```

```
[44]: t
```

```
[44]:           2011    2012    2013    2014    2015
      Country
      Canada  1700.0  1713.0  1707.0  1703.0  1706.0
      Chile   2047.4  2024.0  2015.3  1990.1  1987.5
      Mexico  2250.2  2225.8  2236.6  2228.4  2246.4
      USA     1786.0  1789.0  1787.0  1789.0  1790.0
```

```
[45]: americas = americas.join(t)
```

```
[46]: americas.index.names = ['Country']
```

```
[47]: americas
```

```
[47]:           2000    2001    2002    2003    2004  2005    2006    2007  2008  \
      Country
      Canada  1779.0  1771.0  1754.0  1740.0  1760.0  1747  1745.0  1741.0  1735
      Mexico  2311.2  2285.2  2271.2  2276.5  2270.6  2281  2280.6  2261.4  2258
      USA     1836.0  1814.0  1810.0  1800.0  1802.0  1799  1800.0  1798.0  1792
      Chile   2263.0  2242.0  2250.0  2235.0  2232.0  2157  2165.0  2128.0  2095

                2009    2010    2011    2012    2013    2014    2015
      Country
      Canada  1701.0  1703.0  1700.0  1713.0  1707.0  1703.0  1706.0
      Mexico  2250.2  2242.4  2250.2  2225.8  2236.6  2228.4  2246.4
      USA     1767.0  1778.0  1786.0  1789.0  1787.0  1789.0  1790.0
      Chile   2074.0  2069.6  2047.4  2024.0  2015.3  1990.1  1987.5
```

0.0.8 Concatenate americas and americas_dfs dataframes and store result in americas

[48]:
```
americas_dfs = [americas]
americas = pd.concat(americas_dfs, axis=1)
```

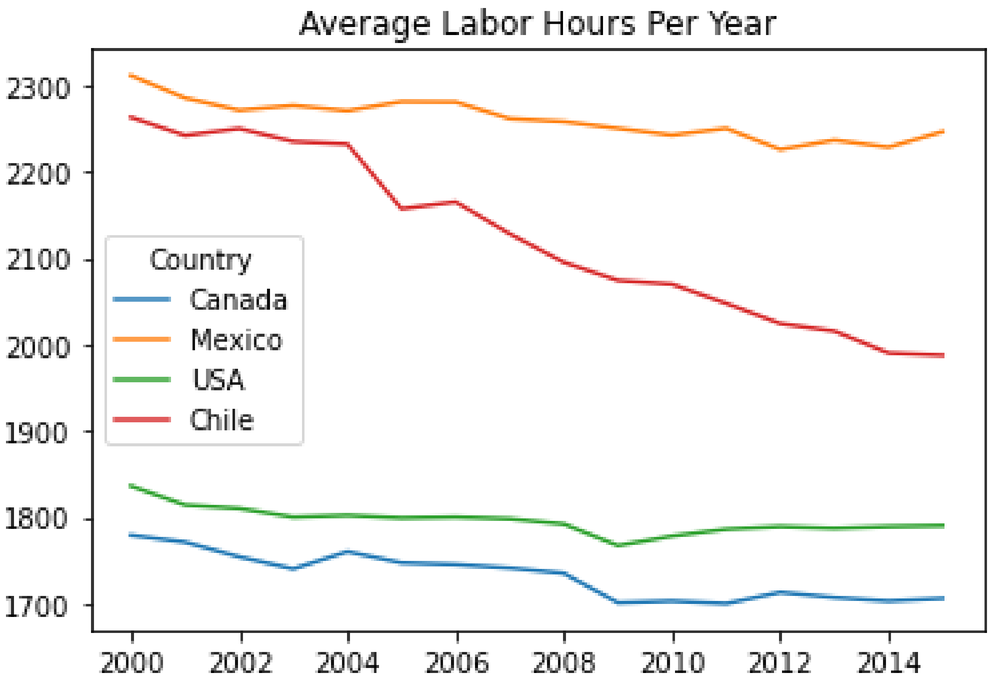
[49]:
```
americas.index.names = ['Country']
americas
```

[49]:
```
            2000    2001    2002    2003    2004  2005    2006    2007  2008  \
Country
Canada    1779.0  1771.0  1754.0  1740.0  1760.0  1747  1745.0  1741.0  1735
Mexico    2311.2  2285.2  2271.2  2276.5  2270.6  2281  2280.6  2261.4  2258
USA       1836.0  1814.0  1810.0  1800.0  1802.0  1799  1800.0  1798.0  1792
Chile     2263.0  2242.0  2250.0  2235.0  2232.0  2157  2165.0  2128.0  2095

            2009    2010    2011    2012    2013    2014    2015
Country
Canada    1701.0  1703.0  1700.0  1713.0  1707.0  1703.0  1706.0
Mexico    2250.2  2242.4  2250.2  2225.8  2236.6  2228.4  2246.4
USA       1767.0  1778.0  1786.0  1789.0  1787.0  1789.0  1790.0
Chile     2074.0  2069.6  2047.4  2024.0  2015.3  1990.1  1987.5
```

0.0.9 Now, plot transposed americas dataframe

[50]:
```
americas.transpose().plot(title='Average Labor Hours Per Year')
plt.show()
```



0.0.10 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

[51]:
```
asia = pd.read_csv('asia_2000_2015.csv', index_col=0)
asia
```

[51]:
```
         2000  2001  2002  2003  2004  2005  2006  2007  2008  2009  2010  \
Country
Israel   2017  1979  1993  1974  1942  1931  1919  1931  1929  1927  1918
Japan    1821  1809  1798  1799  1787  1775  1784  1785  1771  1714  1733
Korea    2512  2499  2464  2424  2392  2351  2346  2306  2246  2232  2187
Russia   1982  1980  1982  1993  1993  1989  1998  1999  1997  1974  1976

         2011  2012  2013  2014  2015
Country
Israel   1920  1910  1867  1853  1858
Japan    1728  1745  1734  1729  1719
Korea    2090  2163  2079  2124  2113
Russia   1979  1982  1980  1985  1978
```

[52]:
```
europe = pd.read_csv('europe_2000_2015.csv', index_col=0)
europe.head()
```

[52]:
```
                  2000    2001    2002    2003    2004    2005    2006  \
Country
Austria         1807.4  1794.6  1792.2  1783.8  1786.8  1764.0  1746.2
Belgium         1595.0  1588.0  1583.0  1578.0  1573.0  1565.0  1572.0
Switzerland     1673.6  1635.0  1614.0  1626.8  1656.5  1651.7  1643.2
Czech Republic  1896.0  1818.0  1816.0  1806.0  1817.0  1817.0  1799.0
Germany         1452.0  1441.9  1430.9  1424.8  1422.2  1411.3  1424.7

                  2007    2008    2009    2010    2011    2012    2013  \
Country
Austria         1736.0  1728.5  1673.0  1668.6  1675.9  1652.9  1636.7
Belgium         1577.0  1570.0  1548.0  1546.0  1560.0  1560.0  1558.0
Switzerland     1632.7  1623.1  1614.9  1612.4  1605.4  1590.9  1572.9
Czech Republic  1784.0  1790.0  1779.0  1800.0  1806.0  1776.0  1763.0
Germany         1424.4  1418.4  1372.7  1389.9  1392.8  1375.3  1361.7

                  2014    2015
Country
Austria         1629.4  1624.9
Belgium         1560.0  1541.0
Switzerland     1568.3  1589.7
```

```
Czech Republic  1771.0  1779.0
Germany         1366.4  1371.0
```

[53]:

```
south_pacific = pd.read_csv('south_pacific_2000_2015.csv', index_col=0)
south_pacific
```

[53]:

```
               2000    2001    2002    2003    2004    2005    2006    2007  \
Country
Australia    1778.7  1736.7  1731.7  1735.8  1734.5  1729.2  1720.5  1712.5
New Zealand  1836.0  1825.0  1826.0  1823.0  1830.0  1815.0  1795.0  1774.0

               2008  2009    2010    2011    2012    2013    2014  2015
Country
Australia    1717.2  1690  1691.5  1699.5  1678.6  1662.7  1663.6  1665
New Zealand  1761.0  1740  1755.0  1746.0  1734.0  1752.0  1762.0  1757
```

If any columns were missing from the data we are trying to append, they would result in those rows having NaN values in the cells falling under the missing year columns. Let's run the append method and verify that all the countries have been sucesfully appended by printing DataFrame.index.

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

[54]:

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

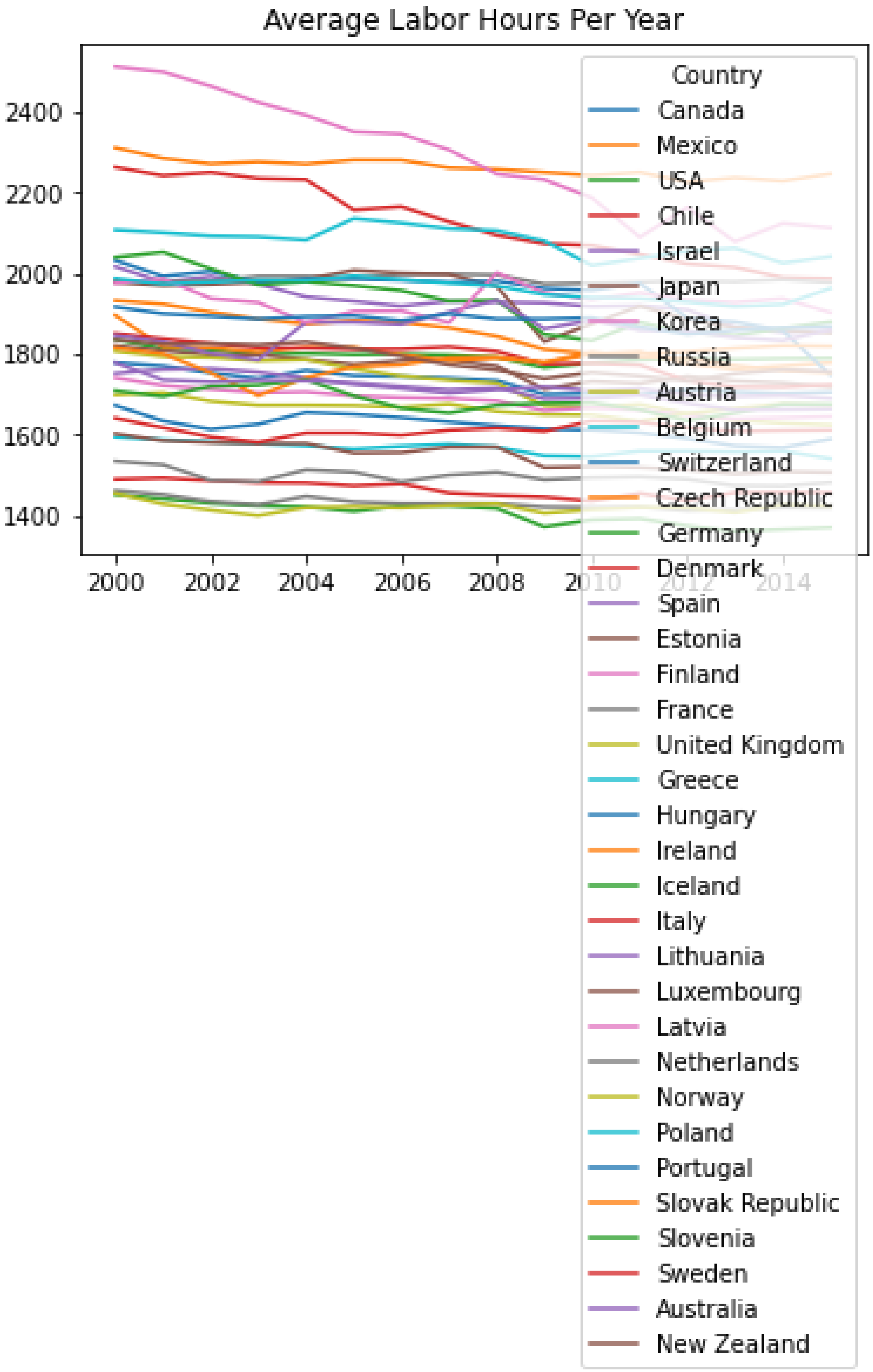
[54]:

```
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', name='Country')
```

0.0.12 Plot, transposed world dataframe

[55]:

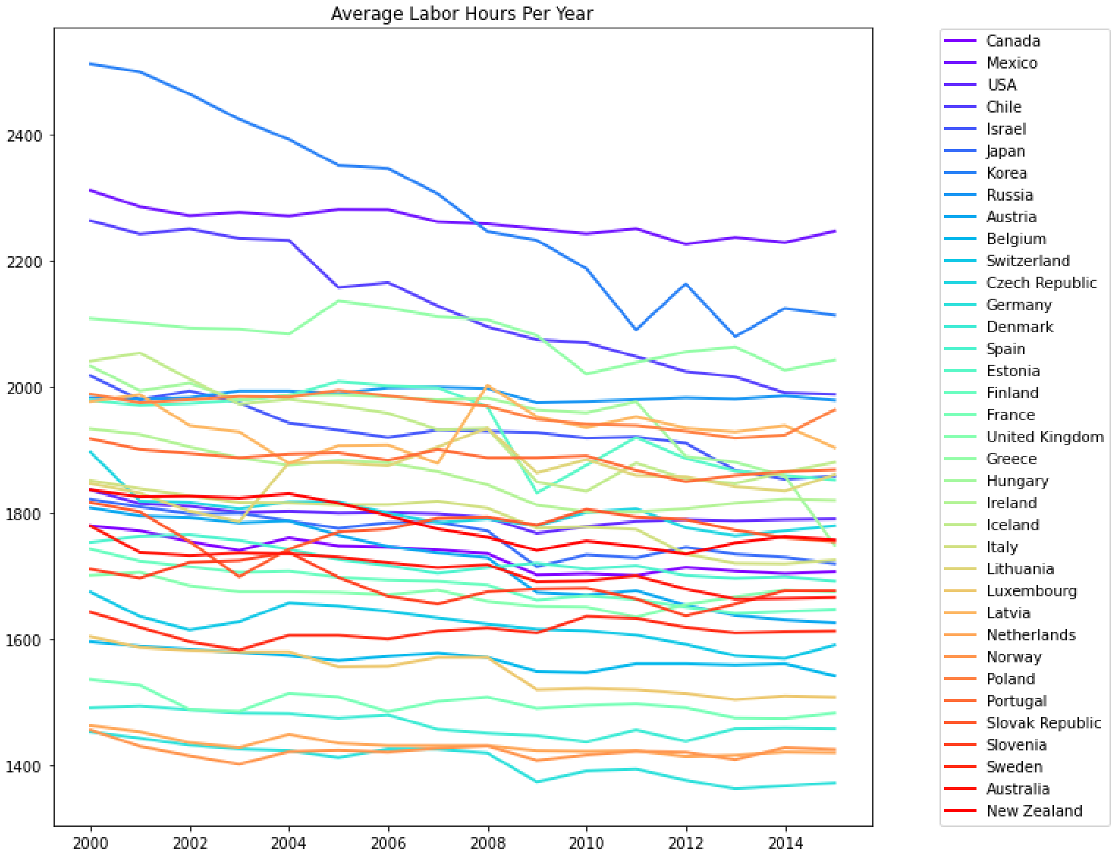
```
world.transpose().plot(title='Average Labor Hours Per Year')
plt.show()
```


Average Labor Hours Per Year
2400
2200
2000
1800
1600
1400
2000
2002
2004
2006
2008
2010
2012
2014
Country
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

0.0.13 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'

```
[56]: world.transpose().plot(figsize=(10,10), colormap='rainbow', linewidth=2,␣
       ↪title='Average Labor Hours Per Year')
      plt.legend(loc='right', bbox_to_anchor=(1.3, 0.5))
      plt.show()
```



0.0.14 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.

```
[57]: historical = pd.read_csv('historical.csv', index_col=0)
      historical.head()
```

```
[57]:                1950   1951   1952   1953   1954   1955   1956   1957   1958   1959   …  \
      Country                                                                             …
```

```
Australia      NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN  …
Austria        NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN  …
Belgium        NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN  …
Canada         NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN  …
Switzerland    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN  …

                 1990     1991     1992     1993     1994      1995      1996  \
Country
Australia      1779.5  1774.90  1773.70  1786.50  1797.60  1793.400  1782.700
Austria           NaN      NaN      NaN      NaN      NaN  1619.200  1637.150
Belgium        1662.9  1625.79  1602.72  1558.59  1558.59  1515.835  1500.295
Canada         1789.5  1767.50  1766.00  1764.50  1773.00  1771.500  1786.500
Switzerland       NaN  1673.10  1684.80  1685.80  1706.20  1685.500  1658.900

                   1997     1998    1999
Country
Australia      1783.600  1768.40  1778.8
Austria        1648.500  1641.65  1654.0
Belgium        1510.315  1513.33  1514.5
Canada         1782.500  1778.50  1778.5
Switzerland    1648.600  1656.60  1678.4

[5 rows x 50 columns]
```

[58]:

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

```
World rows & columns:  (36, 16)
Historical rows & columns:  (39, 50)
```

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

[59]:

```
world_historical = pd.merge(historical, world, left_index=True,␣
 ↪right_index=True, how='right')
```

0.0.16 Print size of world_historical dataframe

[60]:

```
print(world_historical.shape)
```

```
(36, 66)
```

0.0.17 Print top-5 of world_historical dataframe

[61]:

```
world_historical.head()
```

[61]:

```
             1950     1951     1952     1953     1954     1955     1956     1957  \
Country
```

```
Canada      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
Mexico      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
USA      1960.0  1975.5  1978.0  1980.0  1970.5  1992.5  1990.0  1962.0
Chile       NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
Israel      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN

           1958    1959  …    2006    2007    2008    2009    2010    2011  \
Country                  …
Canada      NaN     NaN  …  1745.0  1741.0  1735.0  1701.0  1703.0  1700.0
Mexico      NaN     NaN  …  2280.6  2261.4  2258.0  2250.2  2242.4  2250.2
USA      1936.5  1947.0  …  1800.0  1798.0  1792.0  1767.0  1778.0  1786.0
Chile       NaN     NaN  …  2165.0  2128.0  2095.0  2074.0  2069.6  2047.4
Israel      NaN     NaN  …  1919.0  1931.0  1929.0  1927.0  1918.0  1920.0

           2012    2013    2014    2015
Country
Canada   1713.0  1707.0  1703.0  1706.0
Mexico   2225.8  2236.6  2228.4  2246.4
USA      1789.0  1787.0  1789.0  1790.0
Chile    2024.0  2015.3  1990.1  1987.5
Israel   1910.0  1867.0  1853.0  1858.0

[5 rows x 66 columns]
```

0.0.18 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

[62]:
```
world_historical = historical.join(world, how='right')
world_historical.head()       # Print head of world_historical dataframe
```

[62]:
```
           1950    1951    1952    1953    1954    1955    1956    1957  \
Country
Canada      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
Mexico      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
USA      1960.0  1975.5  1978.0  1980.0  1970.5  1992.5  1990.0  1962.0
Chile       NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
Israel      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN

           1958    1959  …    2006    2007    2008    2009    2010    2011  \
Country                  …
Canada      NaN     NaN  …  1745.0  1741.0  1735.0  1701.0  1703.0  1700.0
Mexico      NaN     NaN  …  2280.6  2261.4  2258.0  2250.2  2242.4  2250.2
USA      1936.5  1947.0  …  1800.0  1798.0  1792.0  1767.0  1778.0  1786.0
Chile       NaN     NaN  …  2165.0  2128.0  2095.0  2074.0  2069.6  2047.4
```

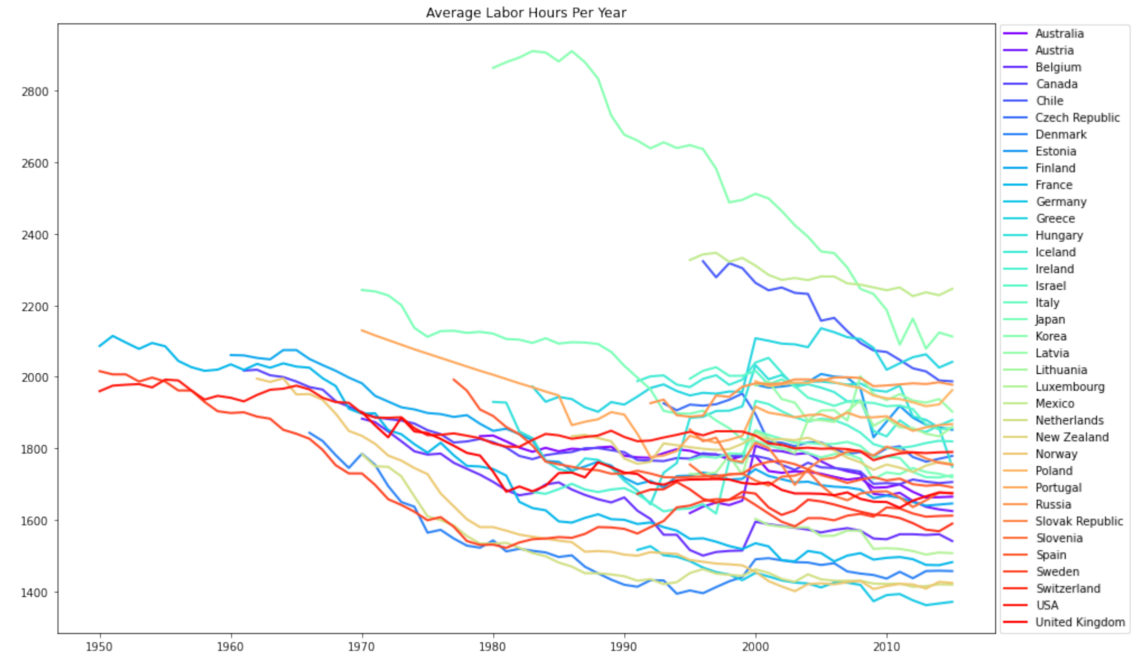
```
Israel      NaN     NaN  …  1919.0  1931.0  1929.0  1927.0  1918.0  1920.0

           2012    2013    2014    2015
Country
Canada   1713.0  1707.0  1703.0  1706.0
Mexico   2225.8  2236.6  2228.4  2246.4
USA      1789.0  1787.0  1789.0  1790.0
Chile    2024.0  2015.3  1990.1  1987.5
Israel   1910.0  1867.0  1853.0  1858.0

[5 rows x 66 columns]
```

0.0.19 Plot, transposed world_historical dataframe

[63]:
```
world_historical.sort_index(inplace=True)
world_historical.transpose().plot(figsize=(15,10), colormap='rainbow',␣
 ↪linewidth=2, title='Average Labor Hours Per Year')
plt.legend(loc='right', bbox_to_anchor=(1.15, 0.5))
plt.show()
```



0.0.20 Which country worked longer hours per year?

[65]:
```
work=world.mean(axis=1)

long=max(world.mean(axis=1))
```

```

short=min(world.mean(axis=1))
```

```
[66]: print("country worked longer hours per year : ",work[work == long].index[0])
```

```
country worked longer hours per year :  Korea
```

0.0.21 Which country worked shorter hours per year?

```
[67]: print("country worked longer hours per year : ",work[work==short].index[0])
```

```
country worked longer hours per year :  Germany
```