Advanced Data Analysis

- Managing large sets of data
- Merging multiple datasets with multi-indexing
- Panel regression

Double indexing panel data

- data is often identified by mutiple indeces

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# index_col = [0,2] will select  countrycode as the primary index and year as
# the secondary index
data = pd.read_excel("mpd2018.xlsx", sheet_name="Full data", index_col=[0, 2])
```

In [2]:
```
data  #prints dataframe
```

Out[2]:

		country	cgdppc	rgdpnapc	pop	i_cig	i_bm
countrycode	**year**						
AFG	**1820**	Afghanistan	NaN	NaN	3280.0	NaN	NaN
	1870	Afghanistan	NaN	NaN	4207.0	NaN	NaN
	1913	Afghanistan	NaN	NaN	5730.0	NaN	NaN
	1950	Afghanistan	2392.0	2392.0	8150.0	Extrapolated	NaN
	1951	Afghanistan	2422.0	2422.0	8284.0	Extrapolated	NaN
...	**...**	...	...	...	...	...	...
ZWE	**2012**	Zimbabwe	1623.0	1604.0	12620.0	Extrapolated	NaN
	2013	Zimbabwe	1801.0	1604.0	13183.0	Extrapolated	NaN
	2014	Zimbabwe	1797.0	1594.0	13772.0	Extrapolated	NaN
	2015	Zimbabwe	1759.0	1560.0	14230.0	Extrapolated	NaN
	2016	Zimbabwe	1729.0	1534.0	14547.0	Extrapolated	NaN

19873 rows × 6 columns

In [3]:
```
data.index  #displays list of tuples including ISO Code and year
```

Out[3]:
```
MultiIndex([('AFG', 1820),
            ('AFG', 1870),
            ('AFG', 1913),
            ('AFG', 1950),
```

```
                ('AFG', 1951),
                ('AFG', 1952),
                ('AFG', 1953),
                ('AFG', 1954),
                ('AFG', 1955),
                ('AFG', 1956),
                ...
                ('ZWE', 2007),
                ('ZWE', 2008),
                ('ZWE', 2009),
                ('ZWE', 2010),
                ('ZWE', 2011),
                ('ZWE', 2012),
                ('ZWE', 2013),
                ('ZWE', 2014),
                ('ZWE', 2015),
                ('ZWE', 2016)],
               names=['countrycode', 'year'], length=19873)
```

In [4]:

```
years = list(set(data.index.get_level_values('year')))
years
```

Out[4]:

```
[1,
 730,
 1000,
 1150,
 1280,
 1281,
 1282,
 1283,
 1284,
 1285,
 1286,
 1287,
 1288,
 1289,
 1290,
 1291,
 1292,
 1293,
 1294,
 1295,
 1296,
 1297,
 1298,
 1299,
 1300,
 1301,
 1302,
 1303,
 1304,
 1305,
 1306,
 1307,
 1308,
 1309,
 1310,
 1311,
 1312,
 1313,
 1314,
 1315,
 1316,
```

```
            1317,
            1318,
            1319,
            1320,
            1321,
            1322,
            1323,
            1324,
            1325,
            1326,
            1327,
            1328,
            1329,
            1330,
            1331,
            1332,
            1333,
            1334,
            1335,
            1336,
            1337,
            1338,
            1339,
            1340,
            1341,
            1342,
            1343,
            1344,
            1345,
            1346,
            1347,
            1348,
            1349,
            1350,
            1351,
            1352,
            1353,
            1354,
            1355,
            1356,
            1357,
            1358,
            1359,
            1360,
            1361,
            1362,
            1363,
            1364,
            1365,
            1366,
            1367,
            1368,
            1369,
            1370,
            1371,
            1372,
            1373,
            1374,
            1375,
            1376,
            1377,
            1378,
            1379,
            1380,
            1381,
```

```
            1382,
            1383,
            1384,
            1385,
            1386,
            1387,
            1388,
            1389,
            1390,
            1391,
            1392,
            1393,
            1394,
            1395,
            1396,
            1397,
            1398,
            1399,
            1400,
            1401,
            1402,
            1403,
            1404,
            1405,
            1406,
            1407,
            1408,
            1409,
            1410,
            1411,
            1412,
            1413,
            1414,
            1415,
            1416,
            1417,
            1418,
            1419,
            1420,
            1421,
            1422,
            1423,
            1424,
            1425,
            1426,
            1427,
            1428,
            1429,
            1430,
            1431,
            1432,
            1433,
            1434,
            1435,
            1436,
            1437,
            1438,
            1439,
            1440,
            1441,
            1442,
            1443,
            1444,
            1445,
            1446,
```

```
            1447,
            1448,
            1449,
            1450,
            1451,
            1452,
            1453,
            1454,
            1455,
            1456,
            1457,
            1458,
            1459,
            1460,
            1461,
            1462,
            1463,
            1464,
            1465,
            1466,
            1467,
            1468,
            1469,
            1470,
            1471,
            1472,
            1473,
            1474,
            1475,
            1476,
            1477,
            1478,
            1479,
            1480,
            1481,
            1482,
            1483,
            1484,
            1485,
            1486,
            1487,
            1488,
            1489,
            1490,
            1491,
            1492,
            1493,
            1494,
            1495,
            1496,
            1497,
            1498,
            1499,
            1500,
            1501,
            1502,
            1503,
            1504,
            1505,
            1506,
            1507,
            1508,
            1509,
            1510,
            1511,
```

```
            1512,
            1513,
            1514,
            1515,
            1516,
            1517,
            1518,
            1519,
            1520,
            1521,
            1522,
            1523,
            1524,
            1525,
            1526,
            1527,
            1528,
            1529,
            1530,
            1531,
            1532,
            1533,
            1534,
            1535,
            1536,
            1537,
            1538,
            1539,
            1540,
            1541,
            1542,
            1543,
            1544,
            1545,
            1546,
            1547,
            1548,
            1549,
            1550,
            1551,
            1552,
            1553,
            1554,
            1555,
            1556,
            1557,
            1558,
            1559,
            1560,
            1561,
            1562,
            1563,
            1564,
            1565,
            1566,
            1567,
            1568,
            1569,
            1570,
            1571,
            1572,
            1573,
            1574,
            1575,
            1576,
```

```
            1577,
            1578,
            1579,
            1580,
            1581,
            1582,
            1583,
            1584,
            1585,
            1586,
            1587,
            1588,
            1589,
            1590,
            1591,
            1592,
            1593,
            1594,
            1595,
            1596,
            1597,
            1598,
            1599,
            1600,
            1601,
            1602,
            1603,
            1604,
            1605,
            1606,
            1607,
            1608,
            1609,
            1610,
            1611,
            1612,
            1613,
            1614,
            1615,
            1616,
            1617,
            1618,
            1619,
            1620,
            1621,
            1622,
            1623,
            1624,
            1625,
            1626,
            1627,
            1628,
            1629,
            1630,
            1631,
            1632,
            1633,
            1634,
            1635,
            1636,
            1637,
            1638,
            1639,
            1640,
            1641,
```

```
            1642,
            1643,
            1644,
            1645,
            1646,
            1647,
            1648,
            1649,
            1650,
            1651,
            1652,
            1653,
            1654,
            1655,
            1656,
            1657,
            1658,
            1659,
            1660,
            1661,
            1662,
            1663,
            1664,
            1665,
            1666,
            1667,
            1668,
            1669,
            1670,
            1671,
            1672,
            1673,
            1674,
            1675,
            1676,
            1677,
            1678,
            1679,
            1680,
            1681,
            1682,
            1683,
            1684,
            1685,
            1686,
            1687,
            1688,
            1689,
            1690,
            1691,
            1692,
            1693,
            1694,
            1695,
            1696,
            1697,
            1698,
            1699,
            1700,
            1701,
            1702,
            1703,
            1704,
            1705,
            1706,
```

```
1707,
1708,
1709,
1710,
1711,
1712,
1713,
1714,
1715,
1716,
1717,
1718,
1719,
1720,
1721,
1722,
1723,
1724,
1725,
1726,
1727,
1728,
1729,
1730,
1731,
1732,
1733,
1734,
1735,
1736,
1737,
1738,
1739,
1740,
1741,
1742,
1743,
1744,
1745,
1746,
1747,
1748,
1749,
1750,
1751,
1752,
1753,
1754,
1755,
1756,
1757,
1758,
1759,
1760,
1761,
1762,
1763,
1764,
1765,
1766,
1767,
1768,
1769,
1770,
1771,
```

```
1772,
1773,
1774,
1775,
1776,
1777,
1778,
1779,
1780,
1781,
1782,
1783,
1784,
1785,
1786,
1787,
1788,
1789,
1790,
1791,
1792,
1793,
1794,
1795,
1796,
1797,
1798,
1799,
1800,
1801,
1802,
1803,
1804,
1805,
1806,
1807,
1808,
1809,
1810,
1811,
1812,
1813,
1814,
1815,
1816,
1817,
1818,
1819,
1820,
1821,
1822,
1823,
1824,
1825,
1826,
1827,
1828,
1829,
1830,
1831,
1832,
1833,
1834,
1835,
1836,
```

```
            1837,
            1838,
            1839,
            1840,
            1841,
            1842,
            1843,
            1844,
            1845,
            1846,
            1847,
            1848,
            1849,
            1850,
            1851,
            1852,
            1853,
            1854,
            1855,
            1856,
            1857,
            1858,
            1859,
            1860,
            1861,
            1862,
            1863,
            1864,
            1865,
            1866,
            1867,
            1868,
            1869,
            1870,
            1871,
            1872,
            1873,
            1874,
            1875,
            1876,
            1877,
            1878,
            1879,
            1880,
            1881,
            1882,
            1883,
            1884,
            1885,
            1886,
            1887,
            1888,
            1889,
            1890,
            1891,
            1892,
            1893,
            1894,
            1895,
            1896,
            1897,
            1898,
            1899,
            1900,
            1901,
```

```
            1902,
            1903,
            1904,
            1905,
            1906,
            1907,
            1908,
            1909,
            1910,
            1911,
            1912,
            1913,
            1914,
            1915,
            1916,
            1917,
            1918,
            1919,
            1920,
            1921,
            1922,
            1923,
            1924,
            1925,
            1926,
            1927,
            1928,
            1929,
            1930,
            1931,
            1932,
            1933,
            1934,
            1935,
            1936,
            1937,
            1938,
            1939,
            1940,
            1941,
            1942,
            1943,
            1944,
            1945,
            1946,
            1947,
            1948,
            1949,
            1950,
            1951,
            1952,
            1953,
            1954,
            1955,
            1956,
            1957,
            1958,
            1959,
            1960,
            1961,
            1962,
            1963,
            1964,
            1965,
            1966,
```

```
1967,
1968,
1969,
1970,
1971,
1972,
1973,
1974,
1975,
1976,
1977,
1978,
1979,
1980,
1981,
1982,
1983,
1984,
1985,
1986,
1987,
1988,
1989,
1990,
1991,
1992,
1993,
1994,
1995,
1996,
1997,
1998,
1999,
2000,
2001,
2002,
2003,
2004,
2005,
2006,
2007,
2008,
2009,
2010,
2011,
2012,
2013,
2014,
2015,
2016]
```

In [5]:

```
countries = list(data.groupby("countrycode").mean().index)
countries
```

Out[5]:

```
['AFG',
 'AGO',
 'ALB',
 'ARE',
 'ARG',
 'ARM',
 'AUS',
 'AUT',
 'AZE',
```

```
                'BDI',
                'BEL',
                'BEN',
                'BFA',
                'BGD',
                'BGR',
                'BHR',
                'BIH',
                'BLR',
                'BOL',
                'BRA',
                'BRB',
                'BWA',
                'CAF',
                'CAN',
                'CHE',
                'CHL',
                'CHN',
                'CIV',
                'CMR',
                'COD',
                'COG',
                'COL',
                'COM',
                'CPV',
                'CRI',
                'CSK',
                'CUB',
                'CYP',
                'CZE',
                'DEU',
                'DJI',
                'DMA',
                'DNK',
                'DOM',
                'DZA',
                'ECU',
                'EGY',
                'ESP',
                'EST',
                'ETH',
                'FIN',
                'FRA',
                'GAB',
                'GBR',
                'GEO',
                'GHA',
                'GIN',
                'GMB',
                'GNB',
                'GNQ',
                'GRC',
                'GTM',
                'HKG',
                'HND',
                'HRV',
                'HTI',
                'HUN',
                'IDN',
                'IND',
                'IRL',
                'IRN',
                'IRQ',
                'ISL',
                'ISR',
```

```
                'ITA',
                'JAM',
                'JOR',
                'JPN',
                'KAZ',
                'KEN',
                'KGZ',
                'KHM',
                'KOR',
                'KWT',
                'LAO',
                'LBN',
                'LBR',
                'LBY',
                'LCA',
                'LKA',
                'LSO',
                'LTU',
                'LUX',
                'LVA',
                'MAR',
                'MDA',
                'MDG',
                'MEX',
                'MKD',
                'MLI',
                'MLT',
                'MMR',
                'MNE',
                'MNG',
                'MOZ',
                'MRT',
                'MUS',
                'MWI',
                'MYS',
                'NAM',
                'NER',
                'NGA',
                'NIC',
                'NLD',
                'NOR',
                'NPL',
                'NZL',
                'OMN',
                'PAK',
                'PAN',
                'PER',
                'PHL',
                'POL',
                'PRI',
                'PRK',
                'PRT',
                'PRY',
                'PSE',
                'QAT',
                'ROU',
                'RUS',
                'RWA',
                'SAU',
                'SDN',
                'SEN',
                'SGP',
                'SLE',
                'SLV',
                'SRB',
```

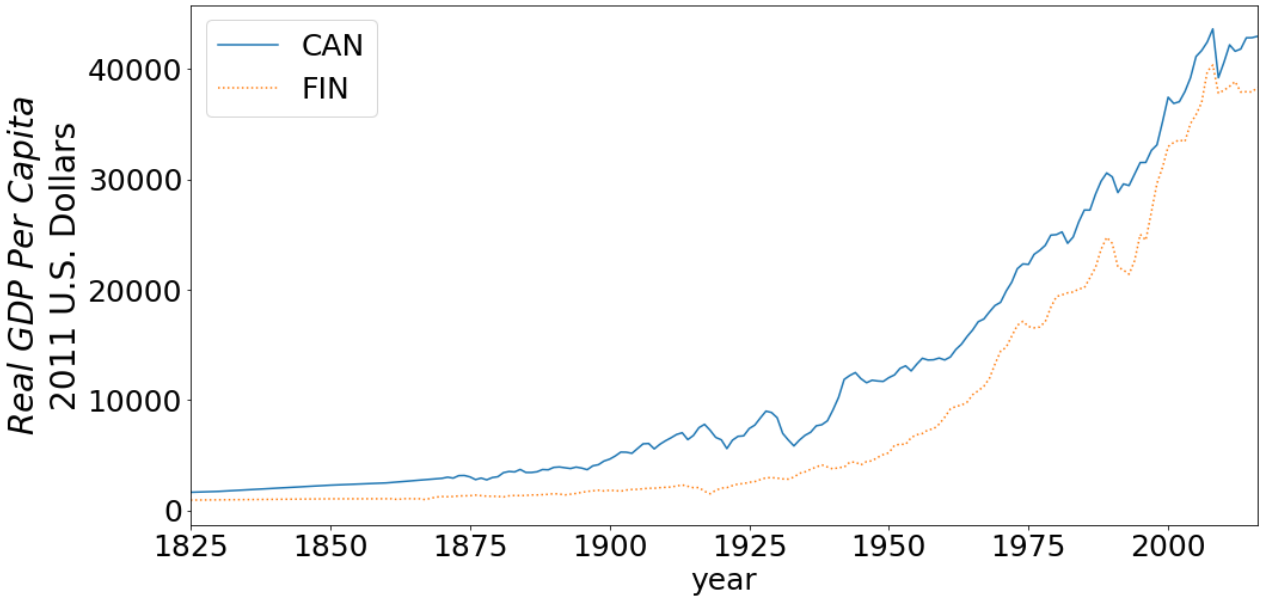
```
'STP',
'SUN',
'SVK',
'SVN',
'SWE',
'SWZ',
'SYC',
'SYR',
'TCD',
'TGO',
'THA',
'TJK',
'TKM',
'TTO',
'TUN',
'TUR',
'TWN',
'TZA',
'UGA',
'UKR',
'URY',
'USA',
'UZB',
'VEN',
'VNM',
'YEM',
'YUG',
'ZAF',
'ZMB',
'ZWE']
```

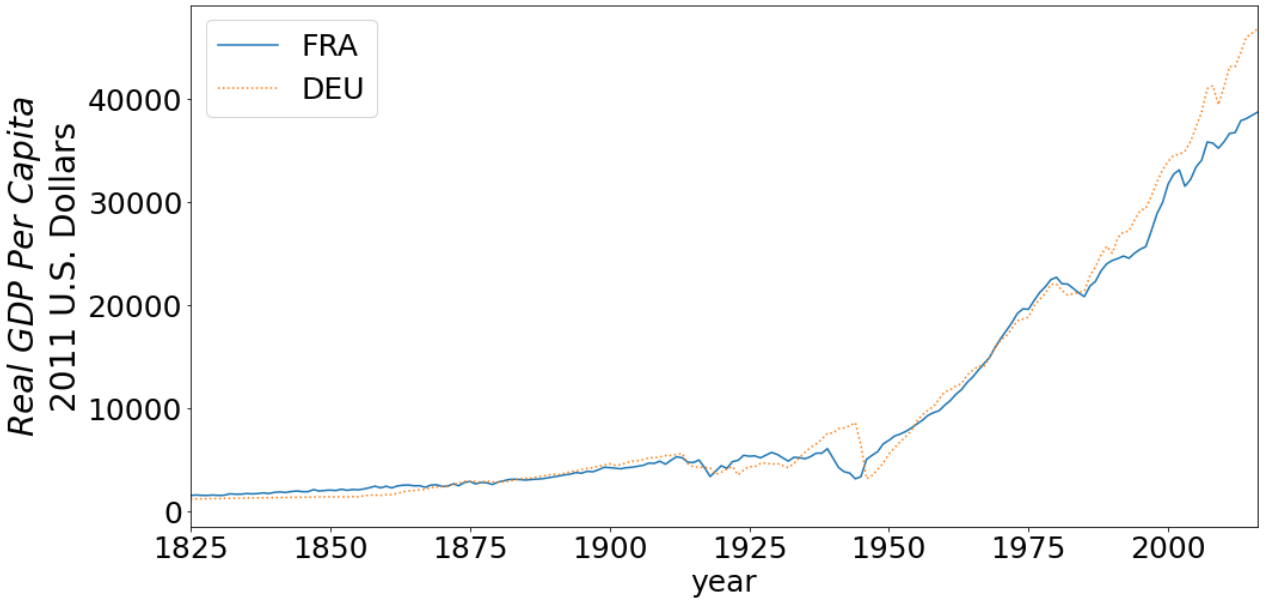
Plotting multi-index data

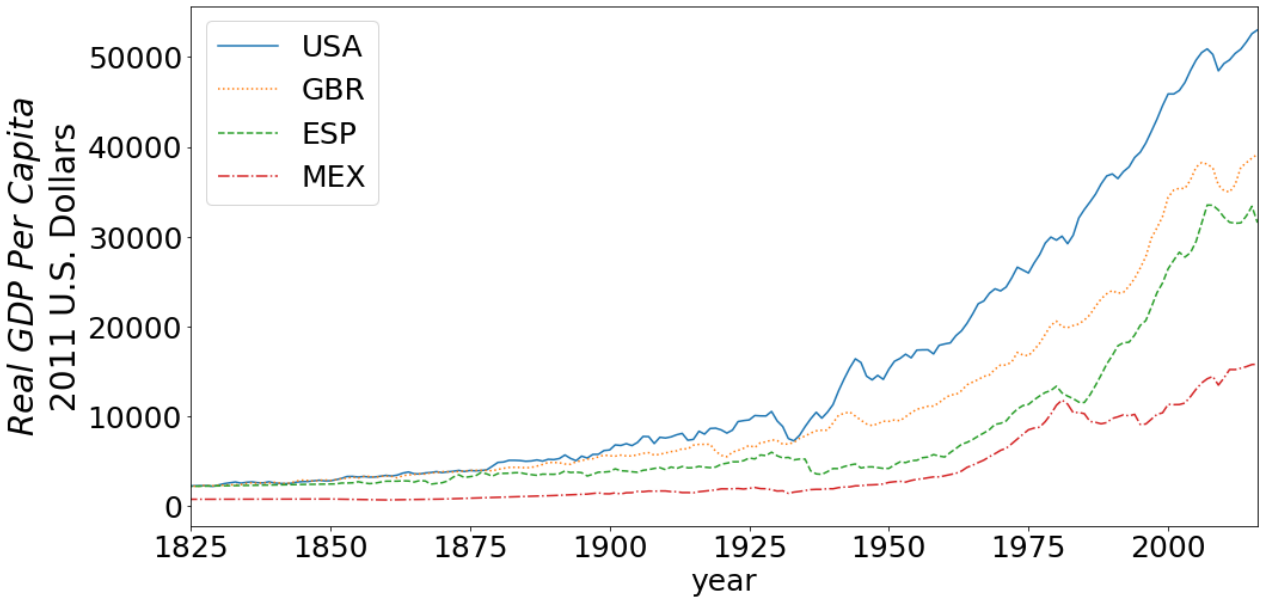
- plot data by country to start

In [6]:
```
pairs = [("CAN", "FIN"), ("FRA", "DEU"), ("USA", "GBR", "ESP", "MEX")]
linestyles = ["-", ":", "--", "-."]
```

In [7]:
```
plt.rcParams.update({"legend.fontsize": 25, "legend.handlelength": 2})
plt.rcParams.update({"font.size": 25})
for pair in pairs:
    fig, ax = plt.subplots(figsize=(16, 8))
    for i in range(len(pair)):
        country = pair[i]
        linestyle = linestyles[i]
        data.loc[country, :]["cgdppc"].dropna().plot.line(ax=ax,
                                                          label=country,
                                                          linestyle=linestyle)
    plt.xlim([1825, max(years)])
    plt.ylabel("$Real$ $GDP$ $Per$ $Capita$\n2011 U.S. Dollars", fontsize=28)
    plt.legend()
    plt.show()
    plt.close()
```







Merging multiple datasets and creating common index

In [8]:
```
import pandas as pd

fraser_data = pd.read_csv("cleanedEconFreedomData (1).csv",  # dropna twice to clean ro
                          index_col=[0, 1],
                          parse_dates=False).dropna(axis=0, thresh=1).dropna(
                              axis=1, thresh=1)

maddison_data = data
```

In [9]:
```
fraser_data["RGDP Per Capita"] = maddison_data["rgdpnapc"]
fraser_data
```

Out[9]:

		EFW	Size of Government	Legal System & Property Rights	Sound Money	Freedom to trade internationally	Regulation	RGDP Per Capita
ISO_Code	**Year**							
ALB	**2017**	7.673511	7.528167	5.064907	9.648271	8.343863	7.782349	NaN
	2016	7.637742	7.875862	5.071814	9.553657	8.214900	7.472476	10342.0
	2015	7.639666	7.904257	5.003489	9.585625	8.109118	7.595838	10032.0
	2014	7.586769	7.882037	4.666740	9.629320	8.208630	7.547119	9808.0
	2013	7.389525	7.807904	4.543782	9.690942	7.705771	7.199224	9660.0
...	**...**	...	...	...	...	...	...	...
ZWE	**2000**	4.299839	5.365058	4.662445	2.891166	3.224735	5.355792	2249.0
	1995	5.518614	6.418859	5.138131	4.915293	5.839664	5.423290	2156.0
	1990	4.516140	5.108843	3.439437	5.664840	4.689623	3.953668	2232.0
	1985	4.226841	5.026250	2.633492	6.305850	3.161743	4.207229	2198.0
	1980	4.054740	6.322625	1.379602	6.343342	3.277015	3.280637	2133.0

3030 rows × 7 columns

In [10]:
```
fraser_data.to_csv("fraserDataWithRGDPPC.csv")
```

Creating indicator variables

- Countries residing in North America would be indicated with a 1 (i.e., True), and those outside of North America would receive a zero.

In [12]:
```
def create_indicator_variables(data, indicator_name, index_name,
                               target_index_list):
    data[indicator_name] = 0
    data.loc[target_index_list, [indicator_name]] = int(1)
```

```
data = fraser_data

index_name = data.index.names[0]
indicator_name = "North America"

countries_in_north_america = [
    "BHS", "BLZ", "CAN", "CRI", "DOM", "SLV", "GTM", "HTI", "HND",
    "JAM", "MEX", "NIC", "PAN", "TTO", "USA"
]
create_indicator_variables(data=data,
                           indicator_name=indicator_name,
                           index_name=index_name,
                           target_index_list=countries_in_north_america)
```

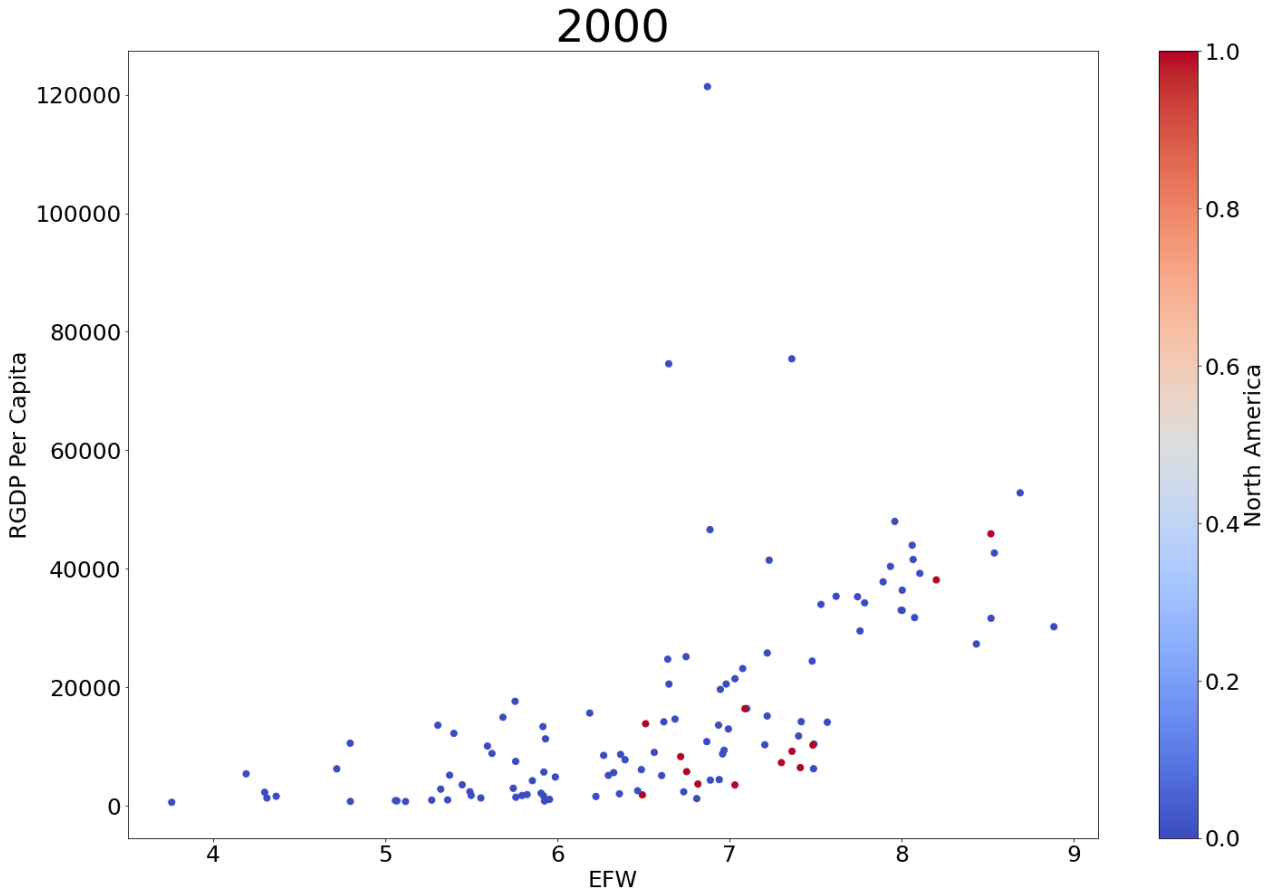
In [13]:
```
data.loc[countries_in_north_america, "North America"]
```

Out[13]:
```
ISO_Code  Year
BHS       2017    1
          2016    1
          2015    1
          2014    1
          2013    1
                 ..
USA       1990    1
          1985    1
          1980    1
          1975    1
          1970    1
Name: North America, Length: 322, dtype: int64
```

Creating quintiles(or any percentage subdivision)

In [14]:
```
from matplotlib import cm
import datetime

year = 2000

plot_data = data[data.index.get_level_values("Year") == 2000]

norm = cm.colors.Normalize()
cmap = cm.get_cmap('coolwarm', 2)
plt.cm.ScalarMappable(cmap=cmap, norm=norm)

fig, ax = plt.subplots(figsize=(24, 16))
plot_data.plot.scatter(x="EFW",
                       y="RGDP Per Capita",
                       c="North America",
                       cmap="coolwarm",
                       ax=ax,
                       s=50)
ax.set_title(str(year), fontsize=50)
```

Out[14]:
```
Text(0.5, 1.0, '2000')
```



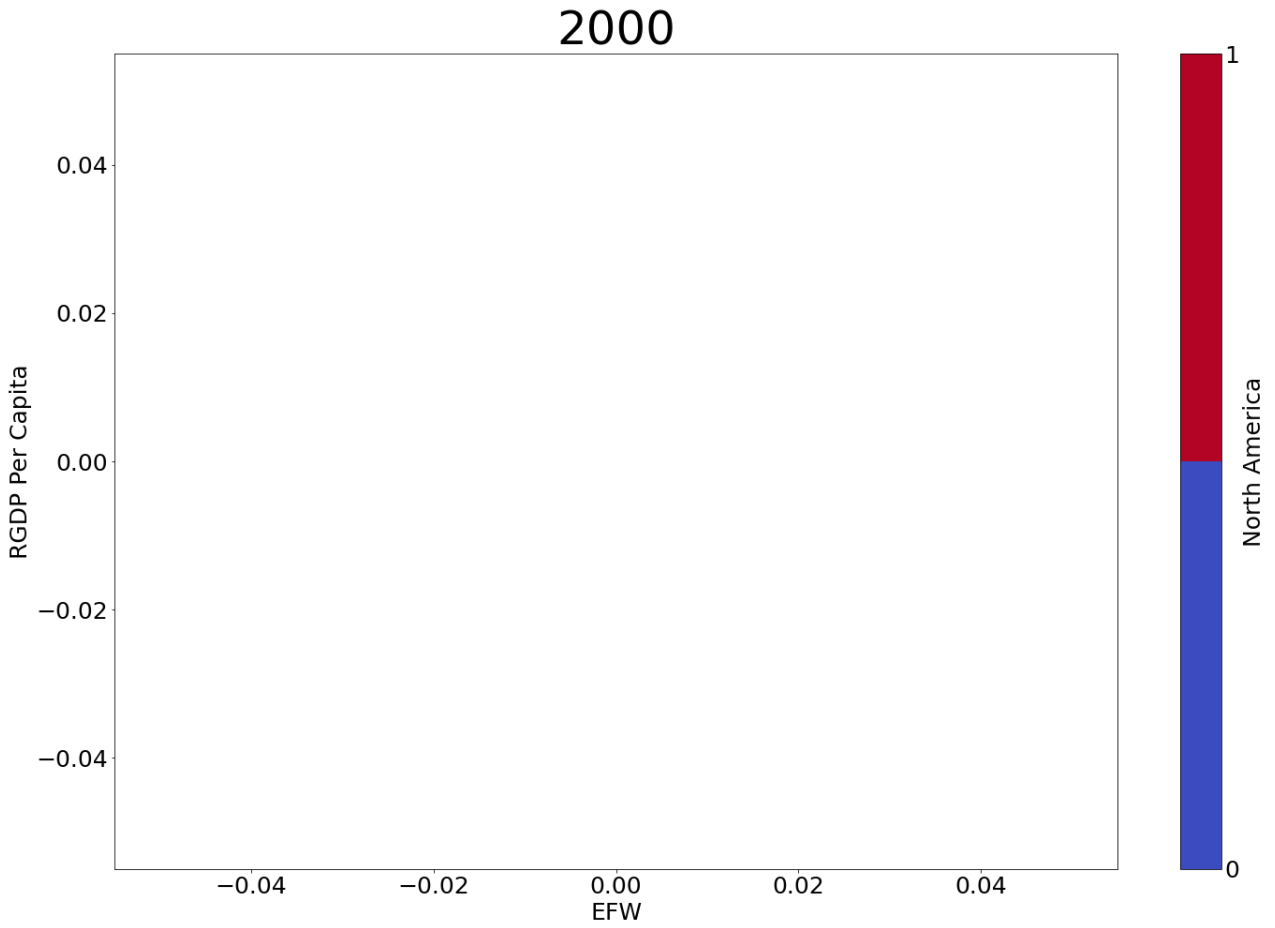
In [16]:

```
import datetime
from matplotlib import cm

year = 2000
# change colors, divide into 4 distinct colors
norm = cm.colors.Normalize()
cmap = cm.get_cmap('coolwarm', 2)
plt.cm.ScalarMappable(cmap=cmap, norm=norm)

plot_data = data[data.index.get_level_values("Year")== datetime.datetime(year,1,1)]
fig, ax = plt.subplots(figsize = (24, 16))
plot_data.plot.scatter(x = "EFW",
                       y = "RGDP Per Capita",
                       c = "North America",
                      cmap = cmap, ax = ax, norm = norm, s = 50)

# to remove numbers between 0 and 1, access the color axis through plt.gcf()
f = plt.gcf()
cax  = f.get_axes()[1]
# access colorbar values
vals = cax.get_yticks()
print(vals)
# only include 0 or 1
cax.set_yticklabels([int(val) if val % 1 == 0 else "" for val in vals ])
# remove tick lines from color axis
cax.tick_params(length = 0)

ax.set_title(str(year), fontsize = 50)
```

```
[0.  0.2 0.4 0.6 0.8 1. ]
```

```
<ipython-input-16-0b5121bf11b2>:24: UserWarning: FixedFormatter should only be used toge
ther with FixedLocator
  cax.set_yticklabels([int(val) if val % 1 == 0 else "" for val in vals ])
```

Out[16]: Text(0.5, 1.0, '2000')



In [71]:

```
#quantile.py
import pandas as pd
import numpy as np

# choose numbers of divisions
n = 8
# import data
data = pd.read_csv("fraserDataWithRGDPPC.csv",
                   index_col = ["ISO_Code", "Year"],
                   parse_dates = True)
#create column identifying n-tile rank
quantile_var = "RGDP Per Capita"
quantile_name = quantile_var + " " + str(n) + "-tile"
data[quantile_name] = np.nan
```

Creating quantiles of our data:

we can find the points for each variable at which the ith Ntile occurs

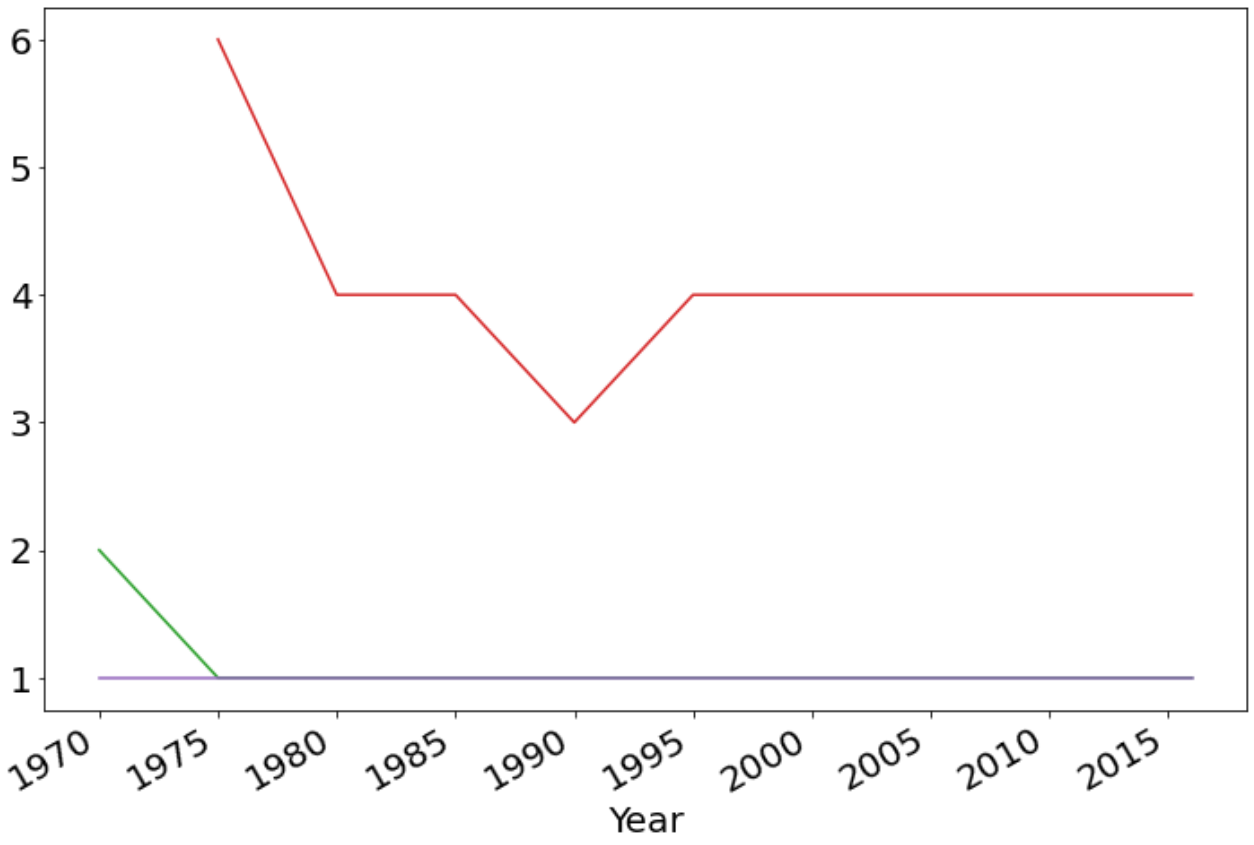
In [72]:

```
#quantile.py
import pandas as pd
import numpy as np
```

```
def create_quantile(n, data, year, quantile_var, quantile_name):
    # index that indentifies countries for a given year
    year_index = data.index.get_level_values("Year") == year
    quantile_values_dict = {i:data[year_index][quantile_var]\
                            .quantile(i/n) for i in range(1, n + 1)}
    # cycle through each country for a given year
    for index in data[year_index].index:
        # identtify value of the variable of interest
        val = data.loc[index][quantile_var]
        # compare that value to the values that divide each quantile
        for i in range(1, n + 1):
            # if the value is less than the highest in the quantile identified,
            # save quantile as i
            if val <= quantile_values_dict[i]:
                data.loc[index,[quantile_name]]=int((n + 1) - i)
                break
            else:
                 continue


years = data.groupby("Year").mean().index
for year in years:
    create_quantile(n, data, year, quantile_var, quantile_name)
```

In [73]:

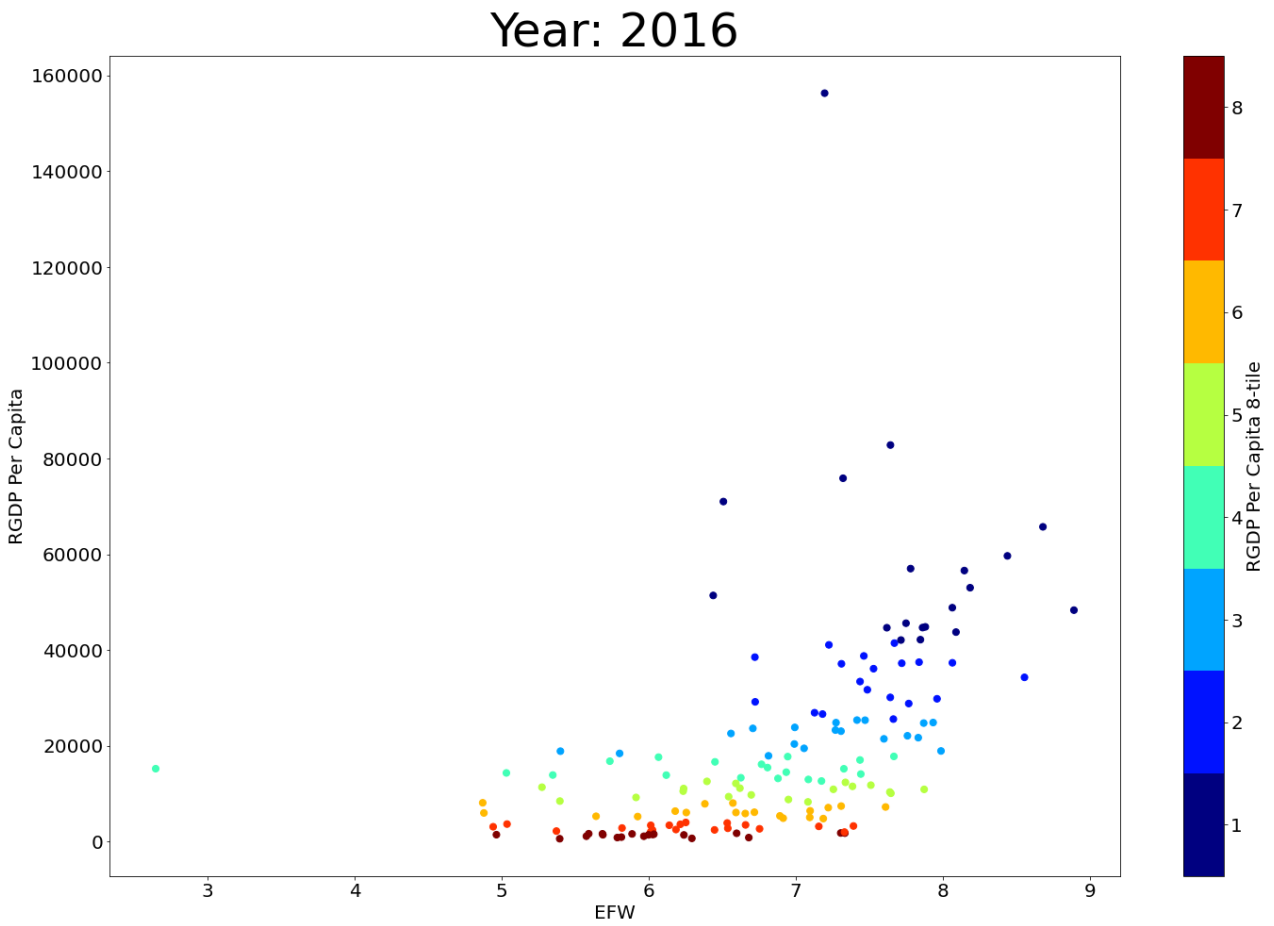
```
countries = ["BHS", "BLZ", "CAN", "MEX", "USA"]
fig, ax = plt.subplots(figsize = (12, 8))
for country in countries:
    data.loc[country, (f"RGDP Per Capita {n}-tile")].plot.line(ax = ax, label = country
```



In [75]:
```
year = 2016
plt.rcParams.update({"font.size":20})
plot_data = data[data.index.get_level_values("Year")== datetime.datetime(year,1,1)]
fig, ax = plt.subplots(figsize = (24, 16))

norm = cm.colors.Normalize()
cmap = cm.get_cmap('jet', n)
plt.cm.ScalarMappable(cmap=cmap, norm=norm)

plot_data.plot.scatter(x = "EFW", y = "RGDP Per Capita",
                       c = "RGDP Per Capita " + str(n) + "-tile",
                      cmap = cmap, norm = norm,  ax = ax, s = 50,
                      legend = False, vmin = 0.5, vmax = n+.5)
ax.set_title("Year: " + str(year), fontsize = 50)
```

```
C:\Users\jzach\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:1041: Mat
plotlibDeprecationWarning: Passing parameters norm and vmin/vmax simultaneously is depre
cated since 3.3 and will become an error two minor releases later. Please pass vmin/vmax
directly to the norm when creating it.
  scatter = ax.scatter(
```

Out[75]:
```
Text(0.5, 1.0, 'Year: 2016')
```



OLS Indicator Variables

In [82]:
```
data["RGDP Per Capita Lag"] = data.groupby(level="ISO_Code")["RGDP Per Capita"].shift(-
data.loc["USA"].index["2001"]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-82-d3fb3c361845> in <module>
      1 data["RGDP Per Capita Lag"] = data.groupby(level="ISO_Code")["RGDP Per Capita"]
.shift(-1)
----> 2 data.loc["USA"].index["2001"]

~\anaconda3\lib\site-packages\pandas\core\indexes\extension.py in __getitem__(self, key)
    236
    237     def __getitem__(self, key):
--> 238         result = self._data[key]
    239         if isinstance(result, type(self._data)):
    240             if result.ndim == 1:

~\anaconda3\lib\site-packages\pandas\core\arrays\datetimelike.py in __getitem__(self, ke
y)
    279         only handle list-likes, slices, and integer scalars
    280         """
--> 281         result = super().__getitem__(key)
    282         if lib.is_scalar(result):
    283             return result

~\anaconda3\lib\site-packages\pandas\core\arrays\_mixins.py in __getitem__(self, key)
    227         key = extract_array(key, extract_numpy=True)
    228         key = check_array_indexer(self, key)
--> 229         result = self._ndarray[key]
    230         if lib.is_scalar(result):
    231             return self._box_func(result)

IndexError: only integers, slices (`:`), ellipsis (`...`), numpy.newaxis (`None`) and in
teger or boolean arrays are valid indices
```