#Climate Change Data Analysis

Climate change is an important issue facing the world in this technological era. The best proof of this situation is the historical temperature change. Our project investigates the reality of the increase in temperatures linked to industrial activities and the greenhouse effect. And before this investigation, the aim of this part enlighten the significant sides of the temperature change data for each area. Before beginning the analysis, firstly, I want to figure out our problem, its cause, and its effects in a short way.

```
!pip install jovian --upgrade --quiet
```

Let's start with the installing required libraries.

```
!pip install pandas seaborn matplotlib plotly --upgrade --quiet
```

Import the libraries.

```
import matplotlib as plt
import pandas as pd
import seaborn as sns
import plotly as px
import numpy as np
%matplotlib inline
plt.rcParams["figure.figsize"] = [10,7]
```

Let's read the csv file and convert it into a dataframe.

```
dataframe = pd.read_csv('Country _Temperature.csv')
```

Analyse the dataframe columns.

```
dataframe.sample(5)
```

	Area Code	Area	Months Code	Months	Element Code	Element	Unit	Y1961	Y1962	Y1963	...	Y2010	Y2011	Y
7511	219	Tonga	7019	Sep–Oct–Nov	6078	Standard Deviation	°C	0.466	0.466	0.466	...	0.466	0.466	
2566	62	Ethiopia PDR	7009	September	7271	Temperature change	°C	-0.266	0.101	0.376	...	NaN	NaN	
8005	228	USSR	7008	August	6078	Standard Deviation	°C	0.497	0.497	0.497	...	NaN	NaN	
2468	63	Estonia	7011	November	7271	Temperature change	°C	NaN	NaN	NaN	...	-0.384	3.640	
7602	223	Turkey	7011	November	7271	Temperature change	°C	0.657	3.256	0.785	...	3.289	-3.738	

5 rows × 66 columns

```
df1 = dataframe.copy()
```

Add the column name country and MonthsCode in the new dataframe, df1 which consists the values of area and Months Code column respectively of original dataframe.

```
df1['Country'] = dataframe['Area']
```

```
df1['MonthsCode'] = dataframe['Months Code']
```

Let's change the values in the rows which have monthscode 7016, 1019, 7018, 7017 with the values Winter, Fall, Summer, Spring repectively for better understanding of data.

```
df1.loc[df1.MonthsCode ==7016, "Months"] = "Winter"
df1.loc[df1.MonthsCode ==7019, "Months"] = "Fall"
df1.loc[df1.MonthsCode ==7018, "Months"] = "Summer"
df1.loc[df1.MonthsCode ==7017, "Months"] = "Spring"
```

```
df1.sample(5)
```

	Area Code	Area	Months Code	Months	Element Code	Element	Unit	Y1961	Y1962	Y1963	...	Y2012	Y2013
5210	151	Netherlands Antilles (former)	7005	May	7271	Temperature change	°C	-0.099	0.058	-0.347	...	0.574	0.813
8724	5204	Central America	7011	November	7271	Temperature change	°C	-0.468	0.091	0.016	...	0.912	0.508
6333	189	Saint Lucia	7005	May	6078	Standard Deviation	°C	0.493	0.493	0.493	...	0.493	0.493
2516	238	Ethiopia	7001	January	7271	Temperature change	°C	NaN	NaN	NaN	...	0.896	1.601
8842	5301	Central Asia	7002	February	7271	Temperature change	°C	NaN	NaN	NaN	...	-4.327	3.457

5 rows × 68 columns

Let's drop the unwanted columns to clean the data.

```
df1 = df1.drop(['Area', 'Area Code', 'Months Code', 'Element Code', 'Unit'], axis=1)
```

Taken the rows which have element value Temperature change.

```
df1 = df1[df1.Element != 'Standard Deviation']
df1 = df1.drop('Element', axis=1)
```

Drop unwanted columns.

```
df1 = df1.drop('MonthsCode', axis=1)
```

```
df1.columns
```

```
Index(['Months', 'Y1961', 'Y1962', 'Y1963', 'Y1964', 'Y1965', 'Y1966', 'Y1967',
       'Y1968', 'Y1969', 'Y1970', 'Y1971', 'Y1972', 'Y1973', 'Y1974', 'Y1975',
       'Y1976', 'Y1977', 'Y1978', 'Y1979', 'Y1980', 'Y1981', 'Y1982', 'Y1983',
       'Y1984', 'Y1985', 'Y1986', 'Y1987', 'Y1988', 'Y1989', 'Y1990', 'Y1991',
       'Y1992', 'Y1993', 'Y1994', 'Y1995', 'Y1996', 'Y1997', 'Y1998', 'Y1999',
       'Y2000', 'Y2001', 'Y2002', 'Y2003', 'Y2004', 'Y2005', 'Y2006', 'Y2007',
       'Y2008', 'Y2009', 'Y2010', 'Y2011', 'Y2012', 'Y2013', 'Y2014', 'Y2015',
       'Y2016', 'Y2017', 'Y2018', 'Y2019', 'Country'],
      dtype='object')
```

Remove the Y from the year columns.

```
for i in range(1961, 2020):
    df1[i] = df1["Y"+str(i)]
```

Drop Unwanted columns.

```
df1 = df1.drop(['Y1961', 'Y1962', 'Y1963', 'Y1964', 'Y1965', 'Y1966', 'Y1967',
       'Y1968', 'Y1969', 'Y1970', 'Y1971', 'Y1972', 'Y1973', 'Y1974', 'Y1975',
       'Y1976', 'Y1977', 'Y1978', 'Y1979', 'Y1980', 'Y1981', 'Y1982', 'Y1983',
       'Y1984', 'Y1985', 'Y1986', 'Y1987', 'Y1988', 'Y1989', 'Y1990', 'Y1991',
       'Y1992', 'Y1993', 'Y1994', 'Y1995', 'Y1996', 'Y1997', 'Y1998', 'Y1999',
       'Y2000', 'Y2001', 'Y2002', 'Y2003', 'Y2004', 'Y2005', 'Y2006', 'Y2007',
       'Y2008', 'Y2009', 'Y2010', 'Y2011', 'Y2012', 'Y2013', 'Y2014', 'Y2015',
       'Y2016', 'Y2017', 'Y2018', 'Y2019'], axis=1)
```

Convert the column header to values of a single column.

```
df2 = df1.melt(id_vars=['Country', 'Months'],
              var_name = 'Year',
              value_name = 'Temp_change')
```

```
df2
```

	Country	Months	Year	Temp_change
0	Afghanistan	January	1961	0.777
1	Afghanistan	February	1961	-1.743
2	Afghanistan	March	1961	0.516
3	Afghanistan	April	1961	-1.709
4	Afghanistan	May	1961	1.412
...	...	...	...	...
284847	OECD	Winter	2019	1.527
284848	OECD	Spring	2019	1.352
284849	OECD	Summer	2019	1.078
284850	OECD	Fall	2019	1.233
284851	OECD	Meteorological year	2019	1.297

284852 rows × 4 columns

Combine the Month and year column to make it date column.

```
cols=["Months", "Year"]
df2['Date'] = df2[cols].apply(lambda x: '-'.join(x.values.astype(str)), axis="columns")
```

Sort values to get the highest temperature change countries.

```
df2= df2.sort_values(by="Temp_change", ascending=False)
df2.head(10)
```

	Country	Months	Year	Temp_change	Date
259472	Svalbard and Jan Mayen Islands	February	2014	11.759	February-2014
220847	Svalbard and Jan Mayen Islands	January	2006	11.331	January-2006
249815	Svalbard and Jan Mayen Islands	January	2012	10.826	January-2012
269127	Svalbard and Jan Mayen Islands	January	2016	10.478	January-2016
220850	Svalbard and Jan Mayen Islands	April	2006	10.049	April-2006
141356	Finland	February	1990	9.730	February-1990
259471	Svalbard and Jan Mayen Islands	January	2014	9.676	January-2014
56695	Svalbard and Jan Mayen Islands	January	1972	9.475	January-1972
114642	Svalbard and Jan Mayen Islands	December	1984	9.303	December-1984
278783	Svalbard and Jan Mayen Islands	January	2018	9.228	January-2018

The highest temperature change in the country Svalbard and Jan Mayen Islands in February-2014.

Group the dataframe by country to calculate the average temperature change over the years and for a better understanding of temperature change in different countries over different years.

```
df2.groupby('Country').mean()
```

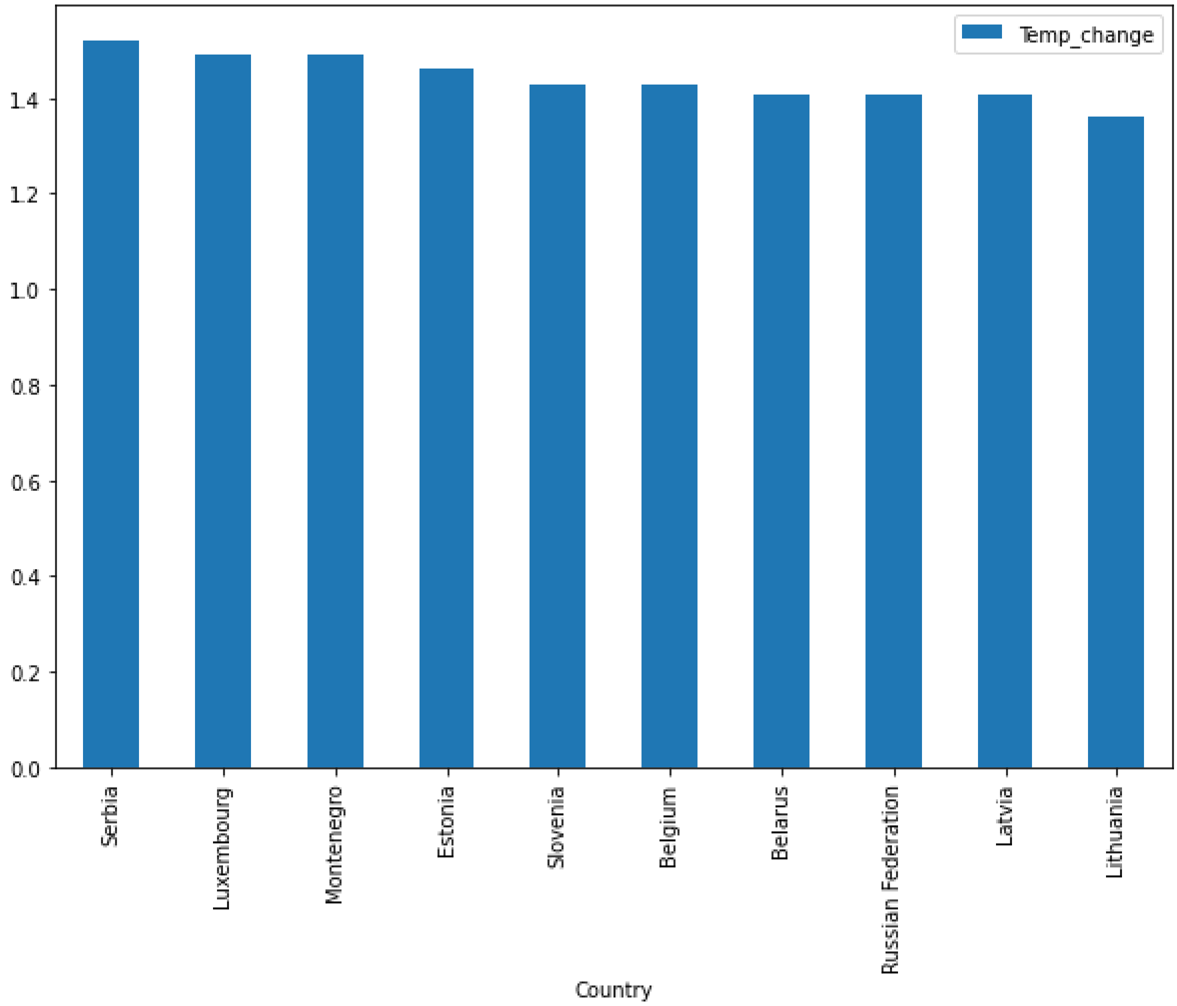
	Temp_change
Country	
Afghanistan	0.434933
Africa	0.491935
Albania	0.485167
Algeria	0.714628
American Samoa	0.434393
...	...
World	0.524768
Yemen	0.148742
Yugoslav SFR	0.023896
Zambia	0.451417
Zimbabwe	0.286078

284 rows × 1 columns

```
df3 = df2.groupby('Country').mean().sort_values(by='Temp_change', ascending=False)
```

```
df3.head(10).plot(kind="bar")
```

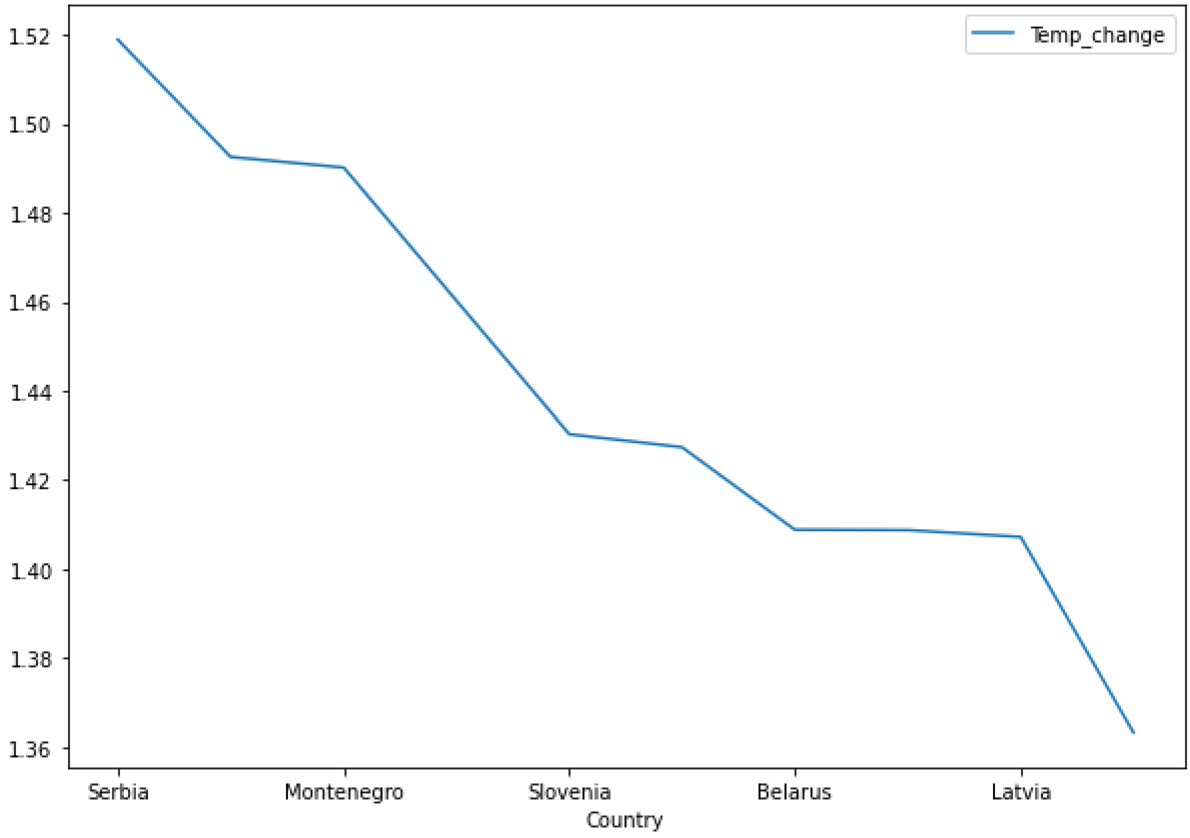
```
<AxesSubplot:xlabel='Country'>
```



As you can see in the bar graph the countries Serbia, Luxembourg, Montenegro, Estonia, Slovenia, Belgium, Belarus, Russian Federation, Latvia, Lithuania undergoes the worst temperature rise over these years.

```
df4.head(10).plot(kind="line")
```

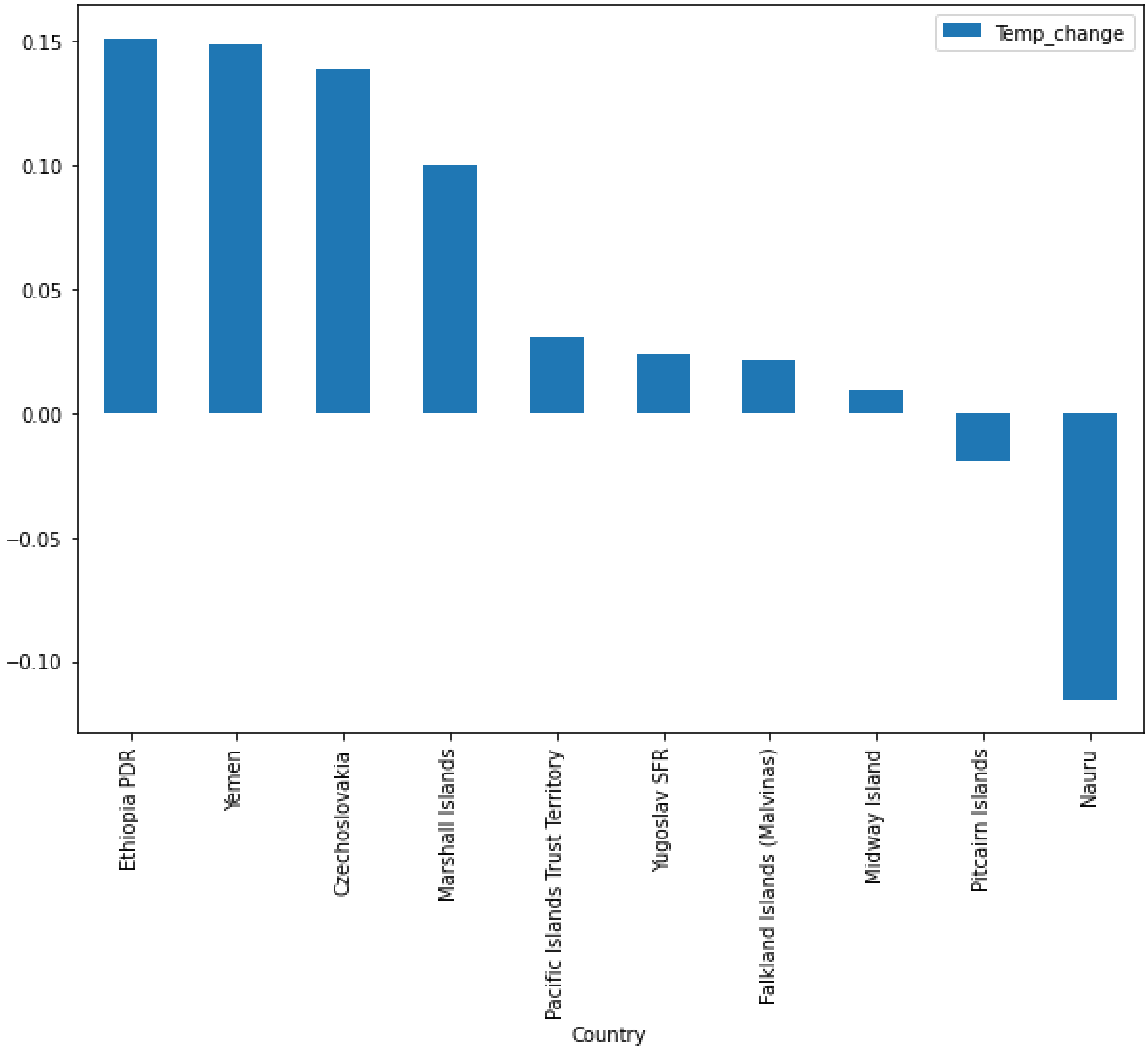
```
<AxesSubplot:xlabel='Country'>
```



The above prediction can also be viewed in the line plot in which it is clearly understandable that Serbia goes the temp rise upto 1.52 degree celcius over these years whereas Montenegro has temperature rise about 1.49 degree celcius and then the graph decreasing with countries Slovenia, Belarus, Latvia and keeps decreasing.

```
df4.tail(10).plot(kind='bar')
```

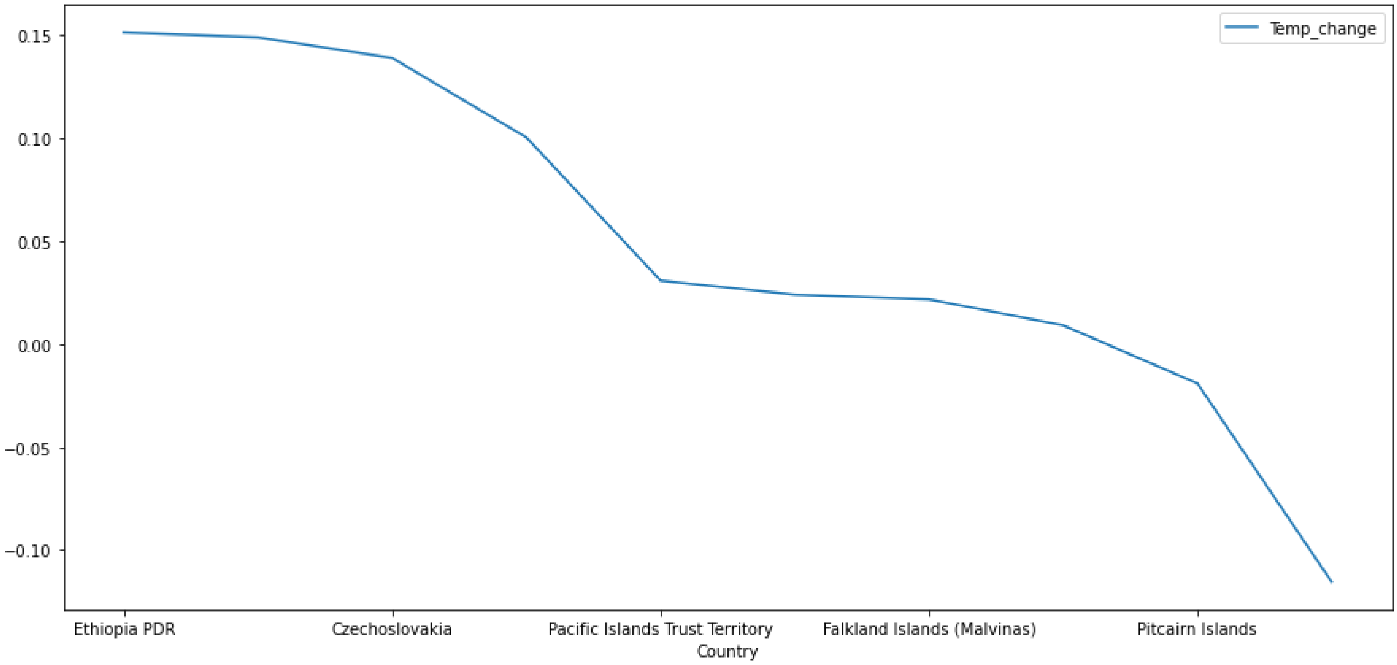
```
<AxesSubplot:xlabel='Country'>
```



As you can see in the bar graph the countries Nauru, Pitcairn Islands, midway islands, falkland island(Malvinas), Yugoslav SFR, Pacific islands trust territory, marshall islands, czechoslovakia, yemen, ethiopia undergoes the major temperature downfall over these years.

```
plt.rcParams['figure.figsize']=[15,7]
df4.tail(10).plot(kind='line')
```

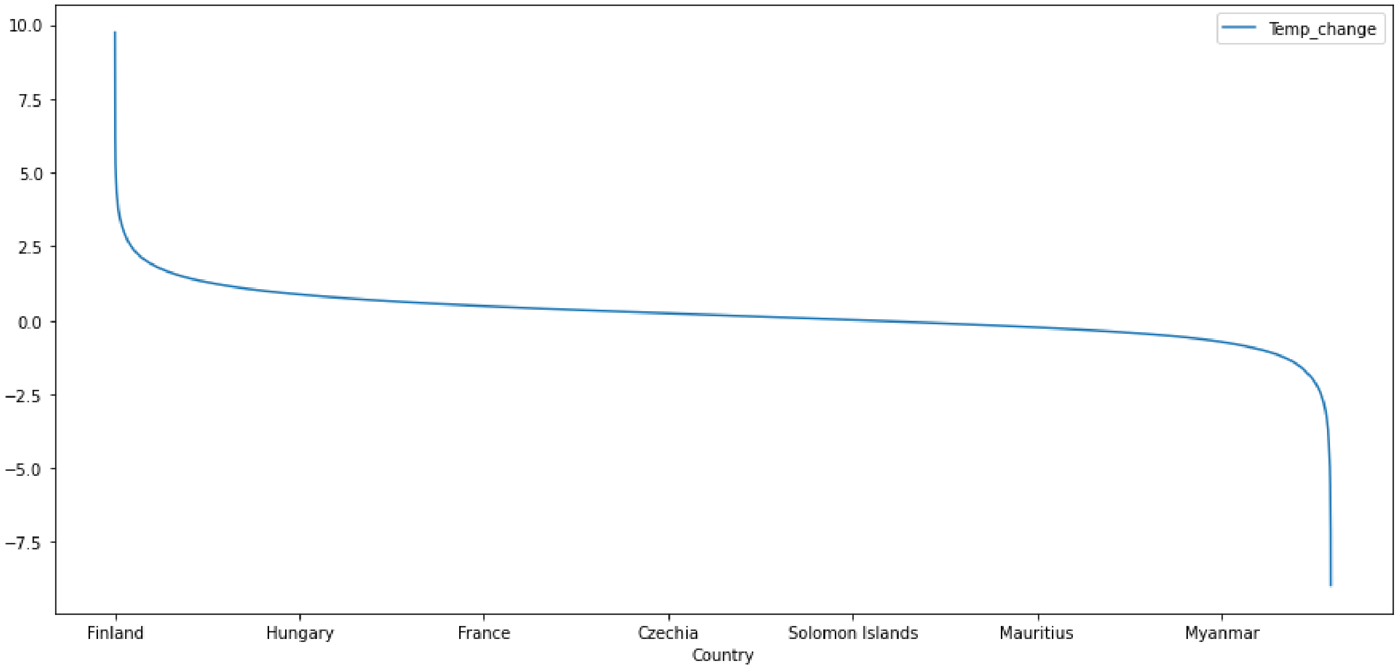
```
<AxesSubplot:xlabel='Country'>
```



The above prediction can also be viewed in the line plot in which it is clearly understandable that Pitcairn Islands experiences the major temp downfall upto -0.8 degree celcius over these years whereas Falkland Islands has temperature downfall about 0.04 degree celcius and then the graph increasing with countries Pacific islands trust territory, Chzechosolvakia, Ethiopia PDR and keeps increasing. We can conclude that mostly the islands experiences the temperature downfall over these years.

```
df3[df3.Year<2000].plot(kind='line', x='Country', y='Temp_change')
```

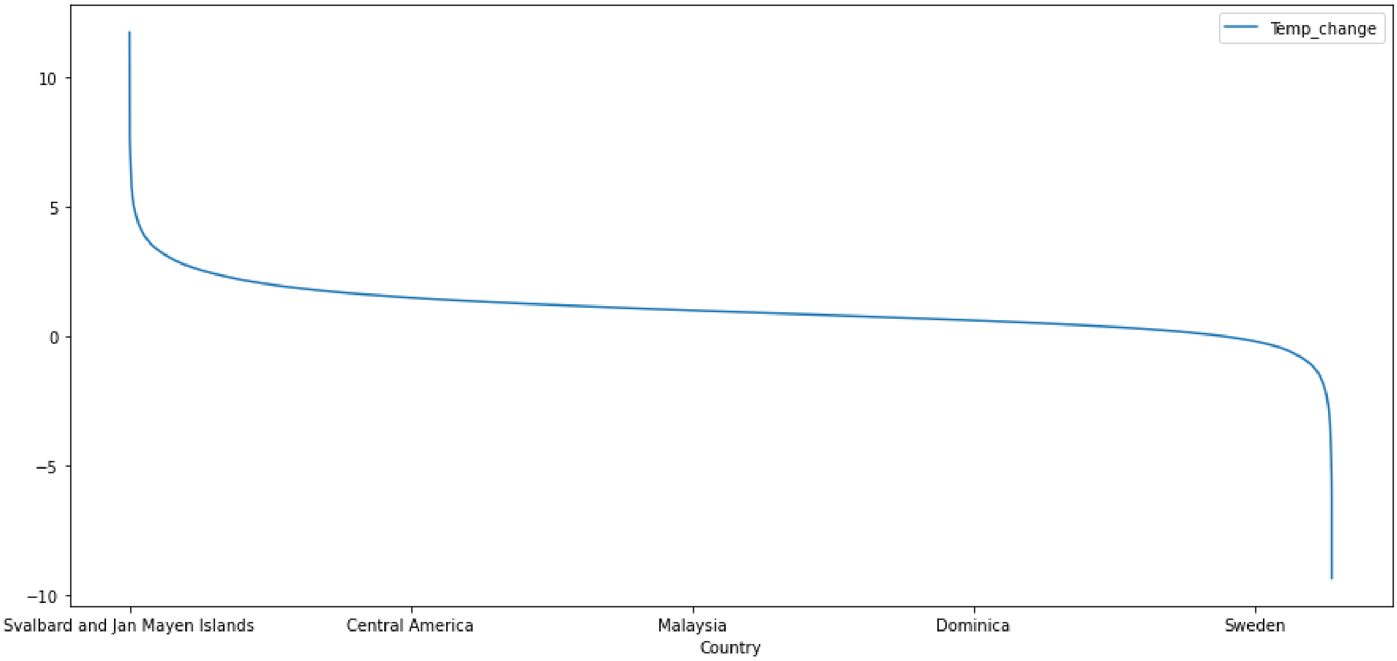
```
<AxesSubplot:xlabel='Country'>
```



Hungary, France, Czechia, Solomon Islands, Mauritis maintains its temperature till 2000s.

```
df3[df3.Year>2000].plot(kind='line', x='Country', y='Temp_change')
```

```
<AxesSubplot:xlabel='Country'>
```



```python
import jovian
```

```python
# Execute this to save new versions of the notebook
jovian.commit(project="data-science-intern-assignment")
```

```
[jovian] Updating notebook "tannu945/data-science-intern-assignment" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/tannu945/data-science-intern-assignment
```

```
'https://jovian.ai/tannu945/data-science-intern-assignment'
```