```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
df = pd.read_csv("/content/drive/MyDrive/환경/climate_change/FAOSTAT_data_11-24-2020.csv")
df1 = pd.read_csv("/content/drive/MyDrive/환경/climate_change/FAOSTAT_data_1-10-2022.csv")
df3 = pd.read_csv("/content/drive/MyDrive/환경/climate_change/Environment_Temperature_change_E_All_Data_NOFLAG.csv", encoding='cp1252')
```

EDA

```
temp_df =  df3.copy()
```

```
temp_df.info()
```

```
 10  Y1964         8252 non-null   float64
 11  Y1965         8281 non-null   float64
 12  Y1966         8364 non-null   float64
 13  Y1967         8347 non-null   float64
 14  Y1968         8345 non-null   float64
 15  Y1969         8326 non-null   float64
 16  Y1970         8308 non-null   float64
 17  Y1971         8303 non-null   float64
 18  Y1972         8323 non-null   float64
 19  Y1973         8394 non-null   float64
 20  Y1974         8374 non-null   float64
 21  Y1975         8280 non-null   float64
 22  Y1976         8209 non-null   float64
 23  Y1977         8257 non-null   float64
 24  Y1978         8327 non-null   float64
 25  Y1979         8290 non-null   float64
 26  Y1980         8283 non-null   float64
 27  Y1981         8276 non-null   float64
 28  Y1982         8237 non-null   float64
 29  Y1983         8205 non-null   float64
 30  Y1984         8259 non-null   float64
 31  Y1985         8216 non-null   float64
 32  Y1986         8268 non-null   float64
 33  Y1987         8284 non-null   float64
 34  Y1988         8273 non-null   float64
 35  Y1989         8257 non-null   float64
 36  Y1990         8239 non-null   float64
 52  Y2006         8503 non-null   float64
 53  Y2007         8534 non-null   float64
 54  Y2008         8475 non-null   float64
 55  Y2009         8419 non-null   float64
 56  Y2010         8435 non-null   float64
 57  Y2011         8437 non-null   float64
 58  Y2012         8350 non-null   float64
 59  Y2013         8427 non-null   float64
 60  Y2014         8377 non-null   float64
 61  Y2015         8361 non-null   float64
 62  Y2016         8348 non-null   float64
 63  Y2017         8366 non-null   float64
 64  Y2018         8349 non-null   float64
 65  Y2019         8365 non-null   float64
dtypes: float64(59), int64(3), object(4)
memory usage: 4.9+ MB
```

```
temp_df.isnull().sum()
```

	0
Area Code	0
Area	0
Months Code	0
Months	0
Element Code	0
...	...
Y2015	1295
Y2016	1308
Y2017	1290
Y2018	1307
Y2019	1291

66 rows × 1 columns

dtype: int64

```
len(temp_df.Area.unique())
```

```
284
```

```
temp_df = temp_df.rename(columns = {'Area Code':'area_code', 'Area':'area', 'Months Code':'months_code','Months':'months','Element Code':'element_cod
```

```
temp_df.drop(temp_df[temp_df.element_code == 6078].index , inplace = True)
```

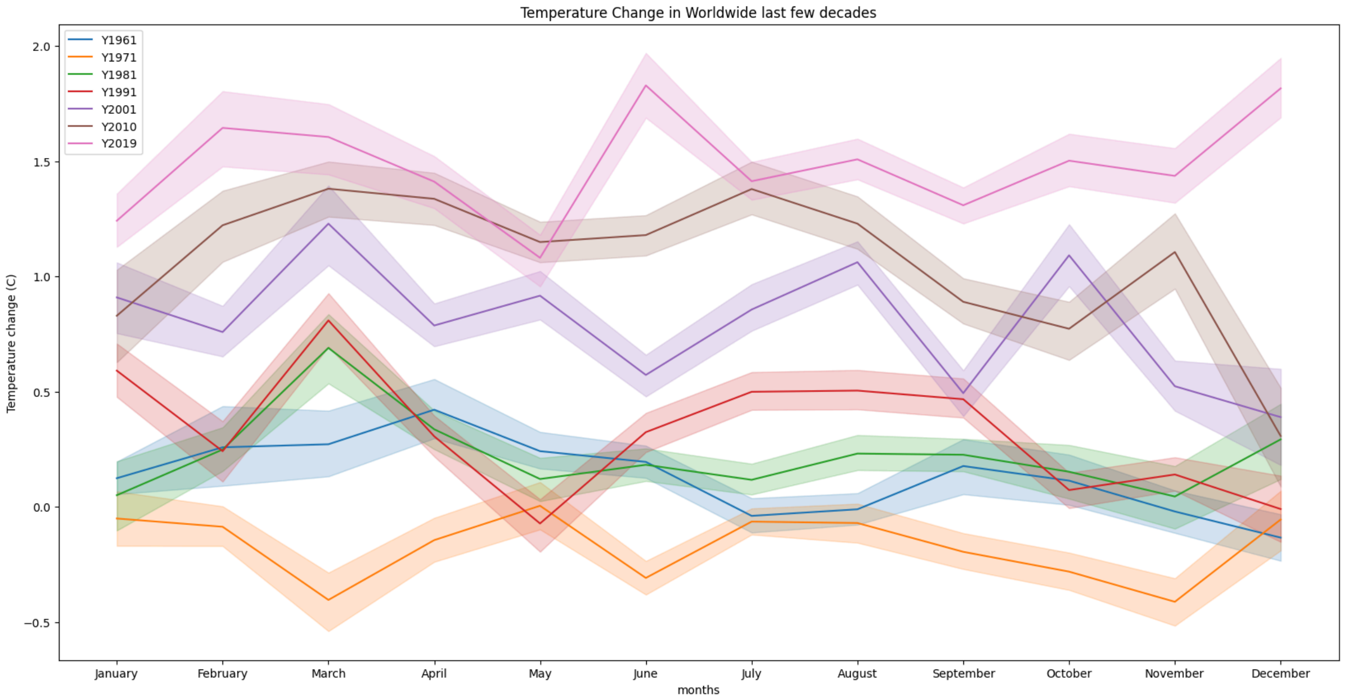
```
temp_df
```

	area_code	area	months_code	months	element_code	element	unit	Y1961	Y1962	Y1963	...	Y2010	Y2011	Y2
0	2	Afghanistan	7001	January	7271	Temperature change	°C	0.777	0.062	2.744	...	3.601	1.179	-0.
2	2	Afghanistan	7002	February	7271	Temperature change	°C	-1.743	2.465	3.919	...	1.212	0.321	-3.
4	2	Afghanistan	7003	March	7271	Temperature change	°C	0.516	1.336	0.403	...	3.390	0.748	-0.
6	2	Afghanistan	7004	April	7271	Temperature change	°C	-1.709	0.117	0.919	...	2.591	1.712	1.
8	2	Afghanistan	7005	May	7271	Temperature change	°C	1.412	-0.092	-0.690	...	1.419	3.643	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9646	5873	OECD	7016	Dec–Jan–Feb	7271	Temperature change	°C	0.561	-0.362	-0.222	...	1.272	0.770	1.
9648	5873	OECD	7017	Mar–Apr–May	7271	Temperature change	°C	-0.038	-0.189	0.141	...	1.742	0.390	1.

```
temp_df.drop(temp_df[temp_df.element_code == 6078].index , inplace = True)
```

```
months = ['January','February','March','April','May','June','July','August','September','October','November','December']
temp_months = temp_df[temp_df['months'].isin(months)]


plt.figure(figsize=(20,10))
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y1961.loc[temp_months.element=='Temperature change'],
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y1971.loc[temp_months.element=='Temperature change'],
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y1981.loc[temp_months.element=='Temperature change'],
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y1991.loc[temp_months.element=='Temperature change'],
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y2001.loc[temp_months.element=='Temperature change'],
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y2010.loc[temp_months.element=='Temperature change'],
sns.lineplot(x=temp_months.months.loc[temp_months.element=='Temperature change'],y=temp_months.Y2019.loc[temp_months.element=='Temperature change'],
plt.xlabel('months')
plt.ylabel('Temperature change (C)')
plt.title('Temperature Change in Worldwide last few decades')
plt.show()
```

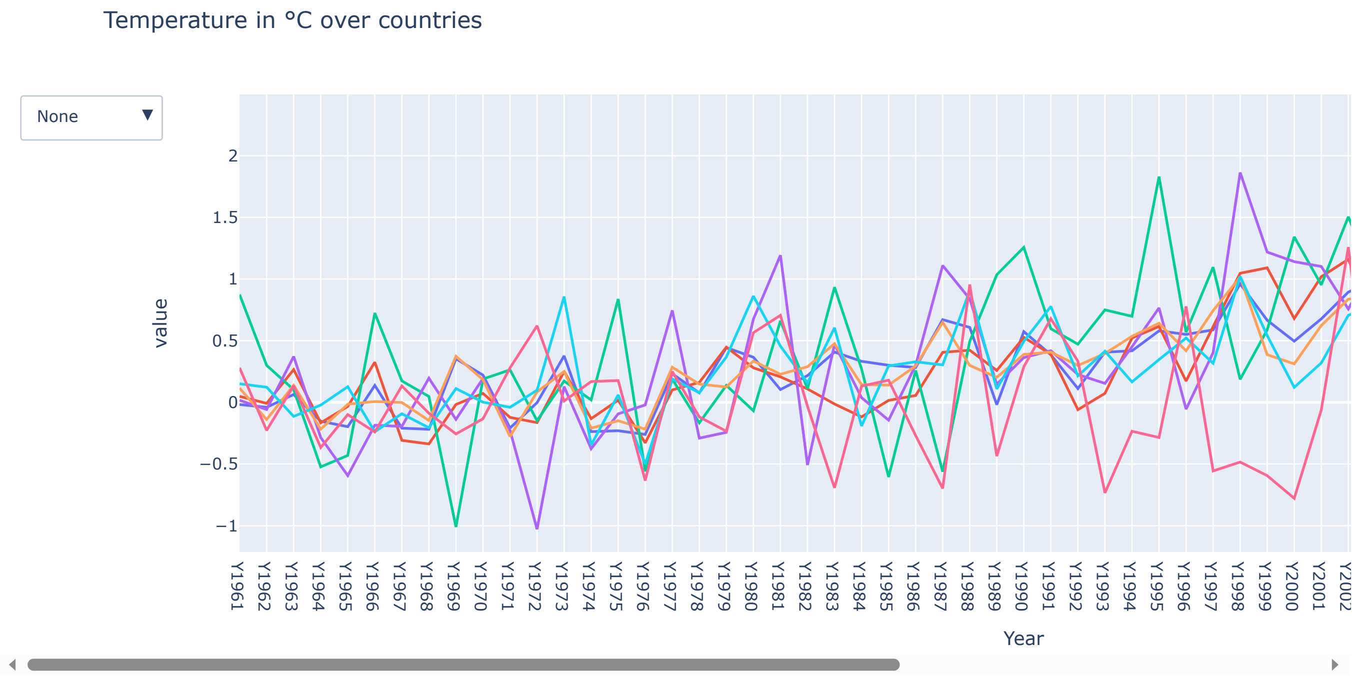


```
temp_year = temp_df[(temp_df["months"]=="Meteorological year")]
temp_year = temp_year.drop(["area_code","months_code","months","element","element_code","unit"],axis=1)
temp_year = temp_year.T
temp_year.columns = temp_year.loc['area']
temp_year.drop('area', inplace = True)
Continents = temp_year[["Africa","Asia","Europe","Northern America","South America","Australia","Antarctica"]]
Continents=Continents.rename(columns={"Northern America":"N_America","South America":"S_America"})
Continents.reset_index(level=0, inplace=True)
Continents=Continents.rename(columns={"index":"Year"})
idx = Continents.columns[1:].tolist()
```

```
fig = px.line(Continents, x=Continents.Year, y=Continents.columns[1:],
              title='Temperature in ° C over countries', width=1500)

fig.update_layout(
    updatemenus=[
        dict(
            active=0,
            buttons=list([dict(label="None",
                     method="update",
                     args=[{"visible": [True for _ in range(186)]},
                           {"title": "Temperature in ° C over continents",
                            "annotations": []}])]) + list([
                dict(label=f"{j}",
                     method="update",
                     args=[{"visible": [True if i==idx else False for i in range(186)]},
                           {"title": f"{j}",
                            "annotations": []}]) for idx,j in enumerate(Continents.columns[1:])])
            )])

fig.show()
```



```
temp = temp_df.melt(id_vars=['area_code','area','months','months_code', 'element','element_code',
       'unit'],var_name='Year', value_name='temperature')
temp = temp.drop(columns = ['area_code', 'months_code',  'element_code'])
temp = temp[temp['months'] == 'Meteorological year']
```

```
Top_countries = temp.groupby('area',).sum().sort_values('temperature', ascending=False)[:10].reset_index()['area']
```

```
Top_countries
```

	area
0	Mongolia
1	Svalbard and Jan Mayen Islands
2	Gambia
3	Morocco
4	Guinea-Bissau
5	Tunisia
6	Mauritania
7	Eastern Europe
8	Austria
9	Senegal

dtype: object

코딩을 시작하거나 AI로 코드를 생성하세요

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.