Full_analysis

May 12, 2023

1 Setup

There are some external libraries necessary for running the notebook. They are listed in `requirements.txt` and can be installed with pip or conda like:

```
pip install -r requirements.txt
```

```
conda install --yes --file requirements.txt
```

As usual, using a virtual environment like venv or conda environment is recommended.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import linregress
import geopandas as gpd
import numpy as np
from math import inf
from sklearn.preprocessing import MinMaxScaler
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=UserWarning)
warnings.simplefilter(action='ignore', category=RuntimeWarning)
%run -i utility_functions.py # running utility_functions.py in IPython's␣
 ↪namespace, importing the functions
```

2 Lab for loading and transforming the data

Responsible: Fabian Pribahsnik (01126693)

2.1 Task

As group 5 we have chosen task 6. Task 6 deals with natural disasters and poses 3 questions to be answered:

1. How did the number of deaths per year from natural disasters change over the last years?
2. How does this vary by country? How does this vary by type of natural disaster?
3. Are there trends visible that could be due to Global Warming?

Based on the questions posed, some requirements for the data set emerge. Question 1 aims at the change in death counts over the last years. Therefore, the dataset must contain death counts as well as a history as long as possible (optimally in the range of 100 years). Question 2 focuses on the development in different countries and whether there are differences between different types of natural disasters. Therefore, the data set must include both geographic characteristics and attributes to distinguish between different types of natural disasters.

2.2 Data Source(s)

Based on these requirements, publicly available datasets were sought. The 2 most relevant of these are presented briefly here and an explanation is given of how the final choice of dataset was made.

2.2.1 WHO data 1 (Explored but not used)

The data was queried from the Mortality Database provided by the WHO. With regard to the origin, completeness and scope of the data, the WHO says:

> The WHO Mortality Database is a compilation of mortality data by country and area, year, sex, age and cause of death, as transmitted annually by national authorities from their civil registration and vital statistics system. It comprises data since 1950 to date. Only data with at least 65% completeness are published here

Requirements met

- Total death counts per year
- Split by country and geographical region
- Split by sex (additional information)
- Split by age (additional information)

Requirements not met

- Long history missing: data for natural disasters starts 1979
- Missing split by type of natural disaster

It can be seen that the WHO data are more detailed in terms of gender and age categories, but they have no distinction of natural disaster categories and also have a limited history of only 50 years.

2.2.2 EM-DAT data 2 (Explored and used)

> In 1988, the Centre for Research on the Epidemiology of Disasters (CRED) launched the Emergency Events Database (EM-DAT). EM-DAT was created with the initial support of the World Health Organisation (WHO) and the Belgian Government.
>
> The main objective of the database is to serve the purposes of humanitarian action at national and international levels. The initiative aims to rationalise decision making for disaster preparedness, as well as provide an objective base for vulnerability assessment and priority setting.
>
> EM-DAT contains essential core data on the occurrence and effects of over 22,000 mass disasters in the world from 1900 to the present day. The database is compiled from var-

ious sources, including UN agencies, non-governmental organisations, insurance companies, research institutes and press agencies.

– Source: https://www.emdat.be/

Requirements met

- Total death counts per event
- Country, region and continent information per event
- Events dated back till 1900
- Disaster type per event

Since the questions do not require an analysis based on gender or age categories, but the EM-DAT dataset covers a much longer period (since 1900) and also assigns a category to each natural disaster, this dataset is perfectly suited for the task and therefore used for the analysis. The database is event based and covers all events for which at least one of the following criteria is fulfilled 3:

- 10 or more people reported killed
- 100 or more people reported affected
- Declaration of a state of emergency
- Call for international assistance

In the following section, a detailed overview of the attributes used in the further analysis will be given.

Data fields

- `Disaster_Group`: Two different types of disasters can be distinguished in the EM-DAT database: natural disasters and technological disasters. Since we are only interested in natural disasters, only they were requested and consequently this field contains only the value *Natural*. No missing values are present for this attribute.
- `Disaster_Subgroup`: Every natural disaster is assigned to one of the following six subgroups: *Biological, Geophysical, Climatological, Hydrological, Meteorological and Extra-terrestrial* to describe the type of natural disaster. No missing values are present for this attribute.
- `Disaster_type`: For every natural disaster event one main disaster type is identified. If two or more disasters are related because they are consequences of each other, then this information is encoded in the attributes `Associated_Dis` and `Associated_Dis2`. No missing values are present for this attribute.
- `Disaster Sub-Type`: Subdivision related to the attribute `Disaster_type` so that a the disaster type Storm can be further classified as tropical, extra-tropical or convective storm.
- `Disaster Sub-Sub Type`: Any appropriate sub-division of the disaster sub-type (not applicable for all disaster sub-types).

  Types of natural disasters could be further broken down using two more categories which would be available in the database. For example, the Disaster type *Storm* could be further subdivided into *Tropical storm*, *Extra-tropical storm* or *Convective storm*. Even a further subdivision of the category *Convective storm* would be possible. Since the analysis is aimed at detecting trends on a high level, the classification of each

event based on the attributes `Disaster_Subgroup` and `Disaster _Type` was considered sufficient and the further subdivisions into `Disaster sub-type` and `Disaster Subsubtype` is only intended to be considered for detailed analysis. The full table is saved in /data/disaster_classification (w/o the technological disaster group) and shown in the Appendix of this notebook.

- `Associated_Dis`: Secondary event triggered by a natural disaster (i.e. Landslide for a flood, explosion after an earthquake, …)
- `Associated_Dis2`: Another secondary event triggered by a natural disaster. (i.e. Landslide for a flood, explosion after an earthquake, …)

Example: If a tsunami is triggered by an earthquake, then the attribute `Disaster_Type` would be *Earthquake*, the attribute `Disaster_Subtype` would be *Ground movement* and the attribute `Associated_Dis` would be *Tsunami/Tidal wave*.

- `Country`: The country in which the disaster has occurred or had an impact. If a disaster has affected more than one country, a seperate entry is created in the database for each country affected. No missing values are present for this attribute.
- `ISO`: Unique 3-letter code for each country defined by ISO 3166. No missing values are present for this attribute.
- `Region`: The region to which the country belongs, based on the UN regional division. No missing values are present for this attribute.
- `Continent`: The continent to which the country belongs. No missing values are present for this attribute.
- `Start_Year`: The year when the disaster occurred. No missing values are present for this attribute.
- `End Year`: The year when the disaster ended. No missing values are present for this attribute.

  For sudden-impact disasters also the month and the day are well defined and available. For disaster situations developing gradually over a longer time period (i.e. drought) with no specific start date the day attribute is empty. For our questions the exact date plays a subordinate role and therefore the year of the beginning of the disaster is completely sufficient for our analysis.
- `Total_Deaths`: Number of people who lost their life because the event happened plus the number of people whose whereabouts since the disaster are unknown, and presumed dead based on official figures. Missing values present for approx. 25% of all events.
- `No_Affected`: Number of people which requiring immediate assistance during an emergency situation. The indicator affected is often reported and is widely used by different actors to convey the extent, impact, or severity of a disaster in non-spatial terms. In case that no values for the attribute `Total_Deaths` are available this attribute could be used as a proxy.

Loading the disaster classification mapping, according to EM-DAT.

[2]:
```
disaster_classification: pd.DataFrame = pd.read_csv("../data/
↪disaster_classification.csv")
```

```
disaster_subgroups = set(disaster_classification.
↪loc[disaster_classification["Disaster Sub-Group"].notna(), "Disaster␣
↪Sub-Group"])
```

Load - Transform - Explore The next steps are to load the data, assign a corresponding data type to each column, and give a brief overview of the data. The detailed analysis of the data set happens in the course of the answering of the questions defined above further down in the notebook. Since the data doesn't cover the whole year 2022, those records are exluded from the analysis in order to have only event data of entire years.

[3]:
```
raw_xlsx = pd.read_excel("../data/emdat.xlsx",
                         skiprows = 5,
                         header = 1,
                         usecols = ['Disaster Group',
                                    'Disaster Subgroup',
                                    'Disaster Type',
                                    'Disaster Subtype',
                                    'Disaster Subsubtype',
                                    'Associated Dis',
                                    'Associated Dis2',
                                    'Country',
                                    'ISO',
                                    'Region',
                                    'Continent',
                                    'Start Year',
                                    'End Year',
                                    'Total Deaths',
                                    'Total Affected'])
raw_xlsx.columns = raw_xlsx.columns.str.replace(' ', '_') # replace all spaces␣
↪in header names
raw_xlsx.head()
```

[3]:

	Disaster_Group	Disaster_Subgroup	Disaster_Type	Disaster_Subtype	\
0	Natural	Climatological	Drought	Drought	
1	Natural	Climatological	Drought	Drought	
2	Natural	Geophysical	Earthquake	Ground movement	
3	Natural	Geophysical	Volcanic activity	Ash fall	
4	Natural	Geophysical	Volcanic activity	Ash fall	

	Disaster_Subsubtype	Country	ISO	Region	Continent	\
0	NaN	Cabo Verde	CPV	Western Africa	Africa	
1	NaN	India	IND	Southern Asia	Asia	
2	NaN	Guatemala	GTM	Central America	Americas	
3	NaN	Guatemala	GTM	Central America	Americas	
4	NaN	Guatemala	GTM	Central America	Americas	

	Associated_Dis	Associated_Dis2	Start_Year	End_Year	Total_Deaths	\

```
0              Famine             NaN        1900      1900       11000.0
1                 NaN             NaN        1900      1900     1250000.0
2  Tsunami/Tidal wave             NaN        1902      1902        2000.0
3                 NaN             NaN        1902      1902        1000.0
4                 NaN             NaN        1902      1902        6000.0

   Total_Affected
0             NaN
1             NaN
2             NaN
3             NaN
4             NaN
```

[4]:
```
df_transformed = pd.DataFrame()
df_transformed = raw_xlsx.convert_dtypes() # assign meaningful data types
print(df_transformed.dtypes) # check data types
df_transformed = df_transformed[(df_transformed['Start_Year'] != 2022)] #␣
 ↪filter out data from 2022
```

```
Disaster_Group         string
Disaster_Subgroup      string
Disaster_Type          string
Disaster_Subtype       string
Disaster_Subsubtype    string
Country                string
ISO                    string
Region                 string
Continent              string
Associated_Dis         string
Associated_Dis2        string
Start_Year              Int64
End_Year                Int64
Total_Deaths            Int64
Total_Affected          Int64
dtype: object
```

[5]:
```
print(df_transformed.shape[0]) # check number of records
print(df_transformed['Total_Deaths'].isna().sum() ) # check number of NA values
print(df_transformed['Start_Year'].min()) # get first year of records
print(df_transformed['Start_Year'].max()) # get last year of records

df_transformed['Disaster_Decade'] = df_transformed['Start_Year']//10*10 #␣
 ↪assign every year to the correspondig decade
print(df_transformed["Total_Deaths"].isnull().groupby(df_transformed.
 ↪Disaster_Decade).sum().astype(int)) # NA values per decade
print(df_transformed["Total_Deaths"].isnull().groupby(df_transformed.Continent).
 ↪sum().astype(int)) # NA values per continent
```

```
16132
4647
1900
2021
Disaster_Decade
1900       8
1910      18
1920      20
1930      16
1940      28
1950      26
1960     142
1970     256
1980     613
1990     897
2000    1301
2010    1033
2020     289
Name: Total_Deaths, dtype: int32
Continent
Africa       947
Americas    1309
Asia        1240
Europe       798
Oceania      353
Name: Total_Deaths, dtype: int32
```

Missing data As discussed earlier, data is collected from multiple sources and therefore there is always the possibility of missing values. The data set contains a total of 16.132 recorded events starting from 1900 up to 2021. For the attribute `Total Deaths` it can be seen that 4.647 records have a missing value which corresponds to a portion of approximately 29% of missing values. Due to the nature of the available data, interpolation of missing values is not appropriate, as disasters can vary significantly in severity, location, and number of people killed. One way to approximate the missing data in terms of deaths is to use the number of affected. However, it must be taken into account that this number is sometimes considerably different from the number of deaths and always overestimates them. Looking at the distribtuion of missing values per decade one can see that the majority of records is from the 1980s or later. When it comes to geografical location of the records with missing data one can observe that the majority of disasters are linked to Americas and Asia.

Data export The transformed data is exported to csv.

[6]:
```
df_transformed.to_csv("../data/emdat_transformed.csv")
```

2.2.3 Population Data

Loading and transformation by Luka Kalezic (12225172)

The population data is necessary only for the question "How does the number of deaths vary by country?", where we will answer that question with using world map with number of deaths per capita. The dataset we got from The World Bank-Population data (link: https://data.worldbank.org/indicator/SP.POP.TOTL) and downloaded as CSV file (link: https://api.worldbank.org/v2/en/indicator/SP.POP.TOTL?downloadformat=csv).

Let's import the data and analyze the structure:

[7]:
```
import numpy as np
from math import inf
population = pd.read_csv('../data/population.csv',skiprows=4)
population
```

[7]:
```
                    Country Name Country Code     Indicator Name  \
0                          Aruba          ABW  Population, total
1    Africa Eastern and Southern          AFE  Population, total
2                    Afghanistan          AFG  Population, total
3     Africa Western and Central          AFW  Population, total
4                         Angola          AGO  Population, total
..                           …           …                …
261                      Kosovo          XKX  Population, total
262                  Yemen, Rep.          YEM  Population, total
263                 South Africa          ZAF  Population, total
264                       Zambia          ZMB  Population, total
265                     Zimbabwe          ZWE  Population, total

    Indicator Code         1960         1961         1962         1963  \
0      SP.POP.TOTL      54608.0      55811.0      56682.0      57475.0
1      SP.POP.TOTL  130692579.0  134169237.0  137835590.0  141630546.0
2      SP.POP.TOTL    8622466.0    8790140.0    8969047.0    9157465.0
3      SP.POP.TOTL   97256290.0   99314028.0  101445032.0  103667517.0
4      SP.POP.TOTL    5357195.0    5441333.0    5521400.0    5599827.0
..             …            …            …            …            …
261    SP.POP.TOTL     947000.0     966000.0     994000.0    1022000.0
262    SP.POP.TOTL    5542459.0    5646668.0    5753386.0    5860197.0
263    SP.POP.TOTL   16520441.0   16989464.0   17503133.0   18042215.0
264    SP.POP.TOTL    3119430.0    3219451.0    3323427.0    3431381.0
265    SP.POP.TOTL    3806310.0    3925952.0    4049778.0    4177931.0

            1964         1965  …         2013         2014         2015  \
0        58178.0      58782.0  …     102880.0     103594.0     104257.0
1    145605995.0  149742351.0  …  567891875.0  583650827.0  600008150.0
2      9355514.0    9565147.0  …   31541209.0   32716210.0   33753499.0
3    105959979.0  108336203.0  …  387204553.0  397855507.0  408690375.0
4      5673199.0    5736582.0  …   26147002.0   27128337.0   28127721.0
..           …            …  …            …            …            …
261    1050000.0    1078000.0  …    1818117.0    1812771.0    1788196.0
262    5973803.0    6097298.0  …   26984002.0   27753304.0   28516545.0
```

```
263   18603097.0   19187194.0  …   53873616.0   54729551.0   55876504.0
264    3542764.0    3658024.0  …   15234976.0   15737793.0   16248230.0
265    4310332.0    4447149.0  …   13555422.0   13855753.0   14154937.0

            2016         2017         2018         2019         2020  \
0       104874.0     105439.0     105962.0     106442.0     106585.0
1    616377331.0  632746296.0  649756874.0  667242712.0  685112705.0
2     34636207.0   35643418.0   36686784.0   37769499.0   38972230.0
3    419778384.0  431138704.0  442646825.0  454306063.0  466189102.0
4     29154746.0   30208628.0   31273533.0   32353588.0   33428486.0
..           …            …            …            …            …
261    1777557.0    1791003.0    1797085.0    1788878.0    1790133.0
262   29274002.0   30034389.0   30790513.0   31546691.0   32284046.0
263   56422274.0   56641209.0   57339635.0   58087055.0   58801927.0
264   16767761.0   17298054.0   17835893.0   18380477.0   18927715.0
265   14452704.0   14751101.0   15052184.0   15354608.0   15669666.0

            2021  Unnamed: 66
0       106537.0          NaN
1    702976832.0          NaN
2     40099462.0          NaN
3    478185907.0          NaN
4     34503774.0          NaN
..           …          …
261    1786038.0          NaN
262   32981641.0          NaN
263   59392255.0          NaN
264   19473125.0          NaN
265   15993524.0          NaN

[266 rows x 67 columns]
```

Since this data needs to be transposed, then features parsed to right type and names, we need some pre-processing.

[8]:
```
population = population.drop(["Unnamed: 66", "Indicator Name", "Indicator␣
 ↪Code", "Country Name"], axis=1)
population = population.melt(id_vars=["Country Code"],
        var_name="Year",
        value_name="Population")
population["Year"] = population["Year"].astype(int)
population.rename(columns={'Country Code': 'ISO'}, inplace=True)
population.set_index(["Year", "ISO"], inplace=True)
population
```

[8]:
```
           Population
Year ISO
1960 ABW      54608.0
```

```
      AFE  130692579.0
      AFG    8622466.0
      AFW   97256290.0
      AGO    5357195.0
…                   …
2021  XKX    1786038.0
      YEM   32981641.0
      ZAF   59392255.0
      ZMB   19473125.0
      ZWE   15993524.0

[16492 rows x 1 columns]
```

We got the right type of data frame we are going to need for later task, the population of each country by year.

2.2.4 Temperature Data

Loading and transformation by Moritz Renkin (11807211) The temperature data is necessary only for the question regarding Global Warming. We use two different datasets in order to investigate a possible relation between natural disasters and global warming, as outlined below.

Temperature Delta per Country (Berkeley) Berkeley Earth gathers temperature data from different sources all around the globe. An overview over their methodology and data sources is available here and its absolute uncertainty (95% range). The specific dataset for this case study is already aggregated per Country and contains the calculated Warming/Century per Country in degrees Celsius. It is available at: https://berkeleyearth.lbl.gov/country-list/. The main reason for choosing Berkeley Earth's dataset is that is thoroughly bias-corrected, in contrast to other common sources such as the World Bank API.

[9]:
```
country_temp_delta = pd.read_csv("../data/country_temp_delta.csv",␣
↪index_col="Country")
```

The "Warming since…" column needs to be split into a warming and uncertainty column.

[10]:
```
country_temp_delta[["Warming/Century", "Uncertainty (±)"]] =␣
↪country_temp_delta["Warming since 1960 (°C / century)"].str.split("±",␣
↪expand=True).apply(pd.to_numeric, errors="coerce")
country_temp_delta.drop("Warming since 1960 (°C / century)", inplace=True,␣
↪axis=1)
country_temp_delta
```

[10]:

Country	Region	Warming/Century	Uncertainty (±)
Afghanistan	Asia	3.32	0.34
Åland	Europe	3.01	0.24
Albania	Europe	1.97	0.28
Algeria	Africa	2.86	0.28

```
American Samoa     NaN             1.43             0.57
…                  …               …                …
Virgin Islands     NaN             1.88             0.30
Western Sahara  Africa             2.49             0.69
Yemen             Asia             2.50             0.55
Zambia          Africa             1.77             0.27
Zimbabwe        Africa             1.50             0.22

[237 rows x 3 columns]
```

Global temperature delta per year (NASA) In order to identify trends and potential relations in the temporal dimension, the yearly global temperature was sourced from NASA. Source: https://climate.nasa.gov/vital-signs/global-temperature/. It contains the yearly temperature delta to the long-term average from 1951 to 1980 in degrees Celsius.

[11]:
```
yearly_global_temp: pd.DataFrame = pd.read_csv("../data/
↪nasa_yearly_global_temperature.csv", index_col="Year", usecols=["Year",␣
↪"Temp_No_Smoothing"]).rename(columns={"Temp_No_Smoothing": "Temperature␣
↪Delta"})
yearly_global_temp
```

[11]:
```
      Temperature Delta
Year
1880              -0.18
1881              -0.09
1882              -0.11
1883              -0.17
1884              -0.28
…                 …
2017               0.92
2018               0.85
2019               0.98
2020               1.02
2021               0.84

[142 rows x 1 columns]
```

1 WHO data: https://platform.who.int/mortality/themes/theme-details/topics/indicator-groups/indicator-group-details/MDB/natural-disasters

2 EM-dat data: https://public.emdat.be/data Based on the terms of use, the data can be used for this exercise:

> If you are an academic organization, a university, a non-profit research institution and/or an international public organization (UN agencies, multi-lateral banks, other multilateral institution and national governments) and/or part of a Media agency (journalist, press agencies) with the intention to use the EM-DAT database (hereafter 'EM-DAT') for research, teaching or information purposes, you shall, conditional upon the

acceptance of the present conditions of use, be granted free access to EM-DAT (also 'Authorized Use').

3 EM-dat data: https://www.emdat.be/explanatory-notes

3 a) How did the number of deaths per year from natural disasters change over the last years?

Responsible: Birgit Freitag (1125363)

[12]:
```
df_a = df_transformed
df_a = df_a.astype({"Total_Deaths": np.float64, "Total_Affected": np.float64,␣
 ↪"Start_Year": np.int32, "Disaster_Decade":np.int32})
```

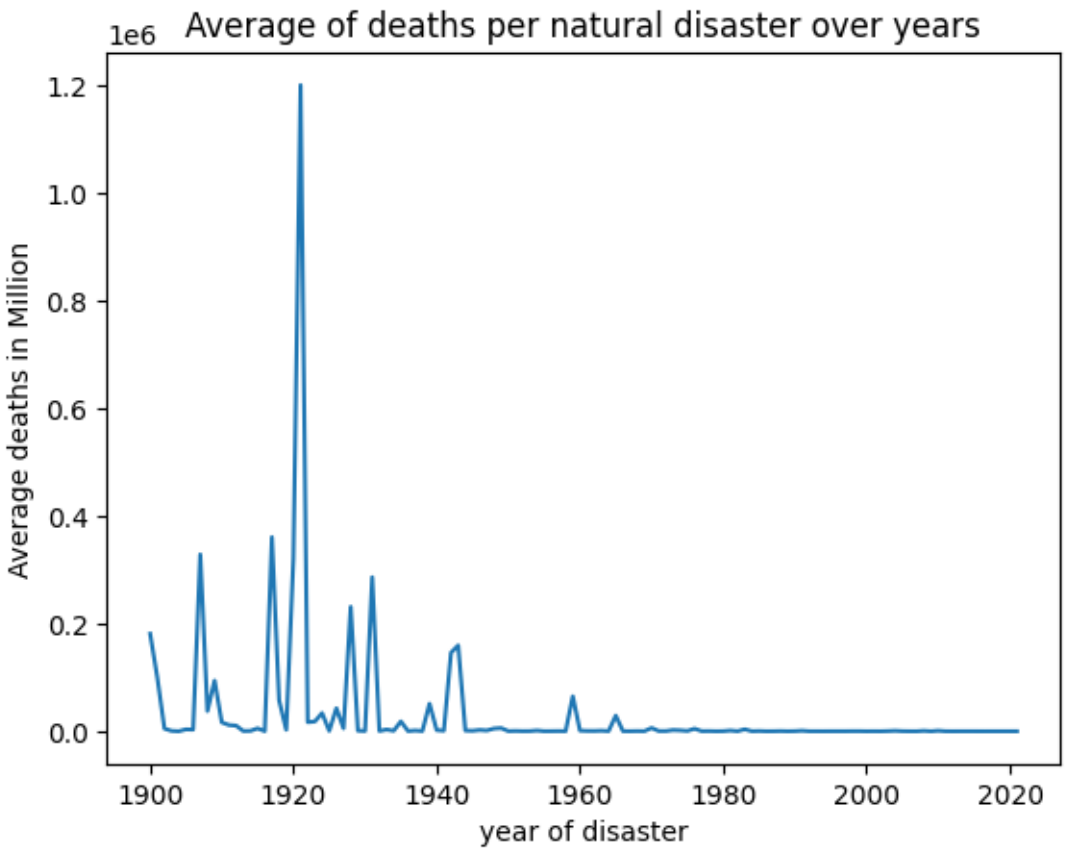
[13]:
```
print(df_a.sum()['Total_Deaths']/(df_a.max()['Start_Year']-df_a.
 ↪min()['Start_Year']+1))
cutoff_date = 1921
df_a_100 = df_a[df_a["Start_Year"] > cutoff_date]
print(df_a_100.sum()['Total_Deaths']/(df_a_100.max()['Start_Year']-df_a_100.
 ↪min()['Start_Year']+1))
cutoff_date = 1971
df_a_50 = df_a[df_a["Start_Year"] > cutoff_date]
print(df_a_50.sum()['Total_Deaths']/(df_a_50.max()['Start_Year']-df_a_50.
 ↪min()['Start_Year']+1))
```

```
265986.7950819672
202370.5
65795.8
```

Natural disasters killed globally on average 266 thousand people per year over the last 122 years. This average lowers to 202 thousand when only having a look at the last 100 years. There died on average 66 thousand people per year in the last 50 years.

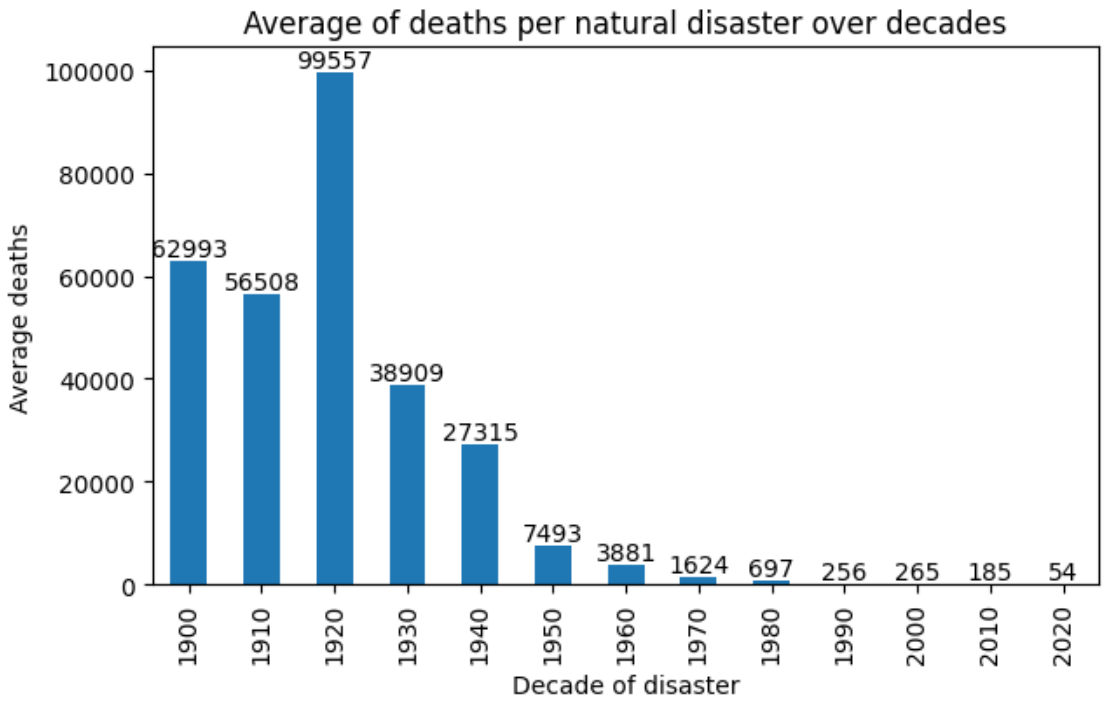
[14]:
```
# calc mean deaths per disaster over the years

means = df_a.groupby(["Start_Year"]).mean()['Total_Deaths'].round()
ax = means.plot(kind = 'line')
ax.set_ylabel('Average deaths in Million')
ax.set_xlabel ('year of disaster')
ax.set_title('Average of deaths per natural disaster over years')
```

[14]:
```
Text(0.5, 1.0, 'Average of deaths per natural disaster over years')
```



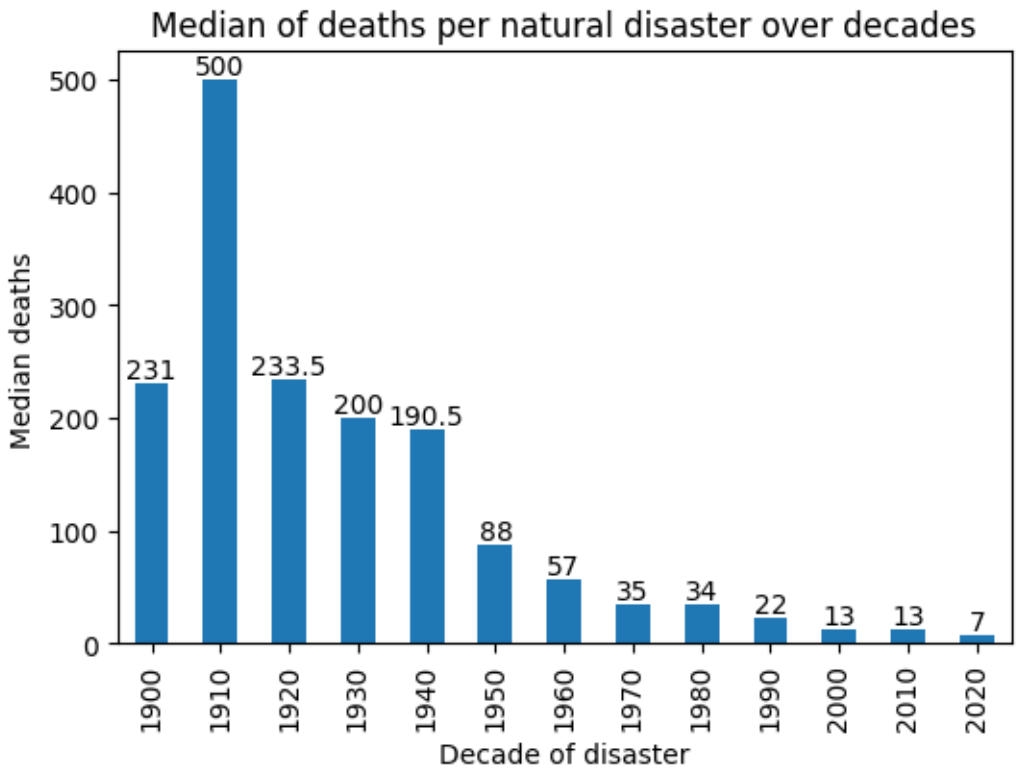
The graphic shows how many people died on average per natural disaster per year. As there is only one event with 1.2 million death in 1921 it leads to that big spike. Therefor we decided to use decades instead of years.

[15]:
```
# calc mean deaths, per Decade and plot them in a bar plot
means = df_a.groupby(["Disaster_Decade"]).mean()['Total_Deaths'].round()
ax = means.plot(kind = 'bar', figsize = (7,4))
ax.set_ylabel('Average deaths')
ax.set_xlabel ('Decade of disaster')
ax.set_title('Average of deaths per natural disaster over decades')
ax.bar_label(ax.containers[0])
plt.show()
```



When having a look at the average deaths per decade per natural disaster, it can be seen that there was a peek in the 1920th and the number of deaths decreased over the years. However, this does not automatically mean that the number of natural disasters is decreasing.

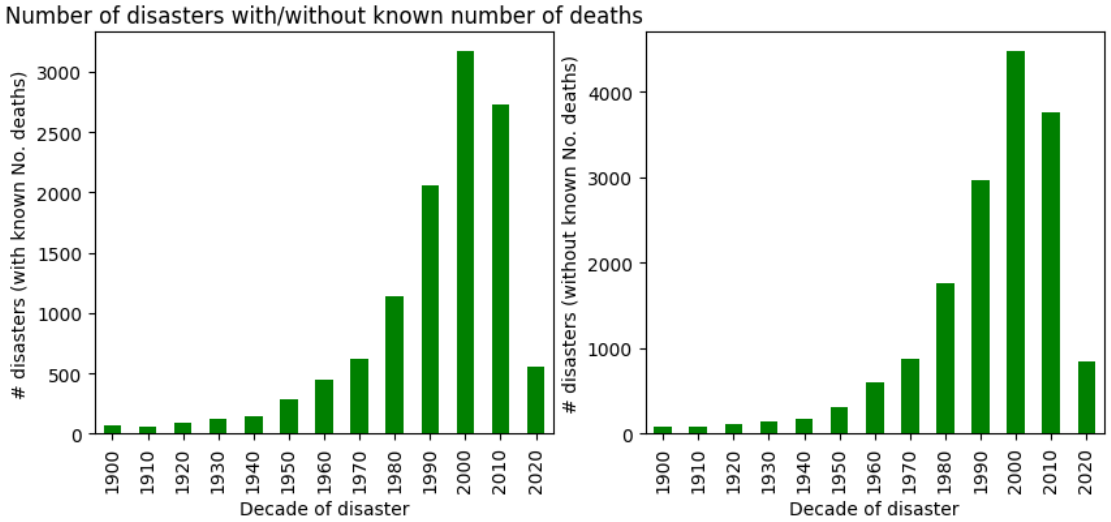
[16]:
```
# calc median deaths, per Decade and plot them in a bar plot
medians = df_a.groupby(["Disaster_Decade"]).median()['Total_Deaths']
ax = medians.plot(kind = 'bar', figsize = (6,4))
ax.set_ylabel('Median deaths')
ax.set_xlabel ('Decade of disaster')
ax.set_title('Median of deaths per natural disaster over decades')
ax.bar_label(ax.containers[0])
plt.show()
```



The median compared to the mean shows that there seem to have been some events in 1920 leading to a really high number of deaths. Still we can see, that the number of deaths have significantly decreased in the last 90 years.

[17]:
```
# calc count, per Decade and plot them in a bar plot
counts = df_a.groupby(["Disaster_Decade"]).count()['Total_Deaths']
plt.subplot(1, 2,1)
ax = counts.plot(kind = 'bar', figsize = (6,4), color ="green")
ax.set_ylabel('# disasters (with known No. deaths)')
ax.set_xlabel ('Decade of disaster')
ax.set_title('Number of disasters with/without known number of deaths')
#ax.bar_label(ax.containers[0])

counts = df_a.groupby(["Disaster_Decade"]).count()['Continent'] # continent␣
 ↪just used, because then we to not have a look at na values
plt.subplot(1, 2, 2)
ax = counts.plot(kind = 'bar', figsize = (10,4), color ="green")
ax.set_ylabel('# disasters (without known No. deaths)')
ax.set_xlabel ('Decade of disaster')
#ax.bar_label(ax.containers[0])
plt.show()
```

```
print(counts)
print("Last Start_year of data : " +str(df_a["Start_Year"].max()))
print( "Hochgerechnete Number of natural disasters 2020-2029: " + str(1204*3.5))
```



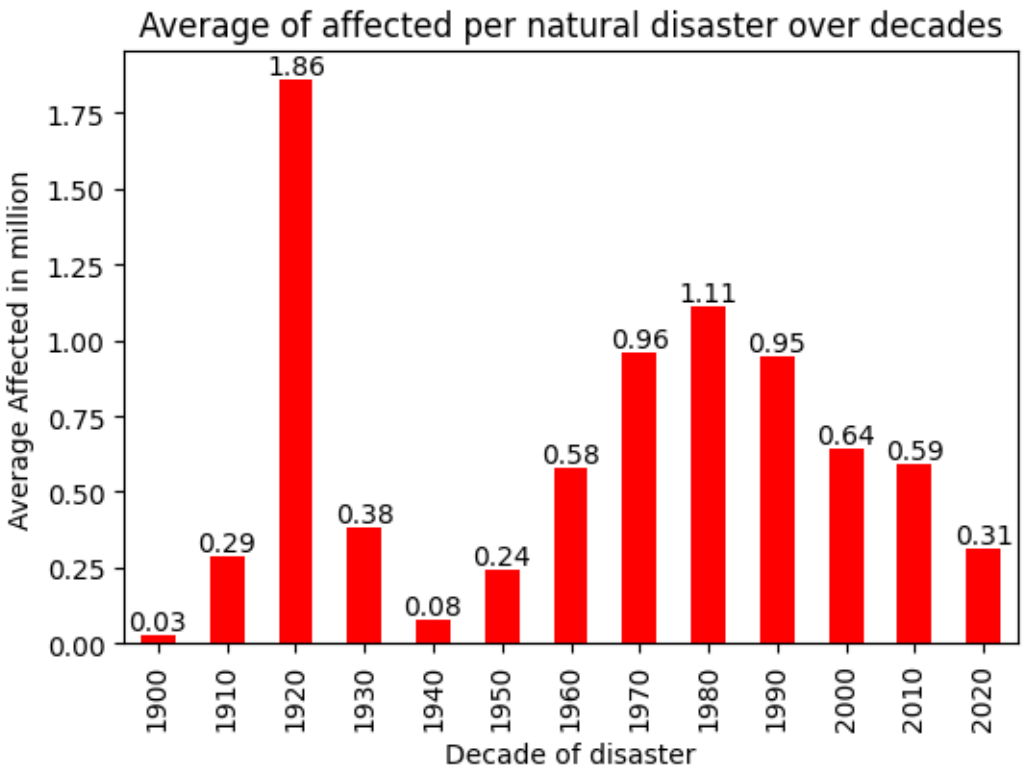

```
Disaster_Decade
1900       79
1910       77
1920      106
1930      135
1940      170
1950      310
1960      593
1970      871
1980     1755
1990     2957
2000     4473
2010     3758
2020      848
Name: Continent, dtype: int64
Last Start_year of data : 2021
Hochgerechnete Number of natural disasters 2020-2029: 4214.0
```

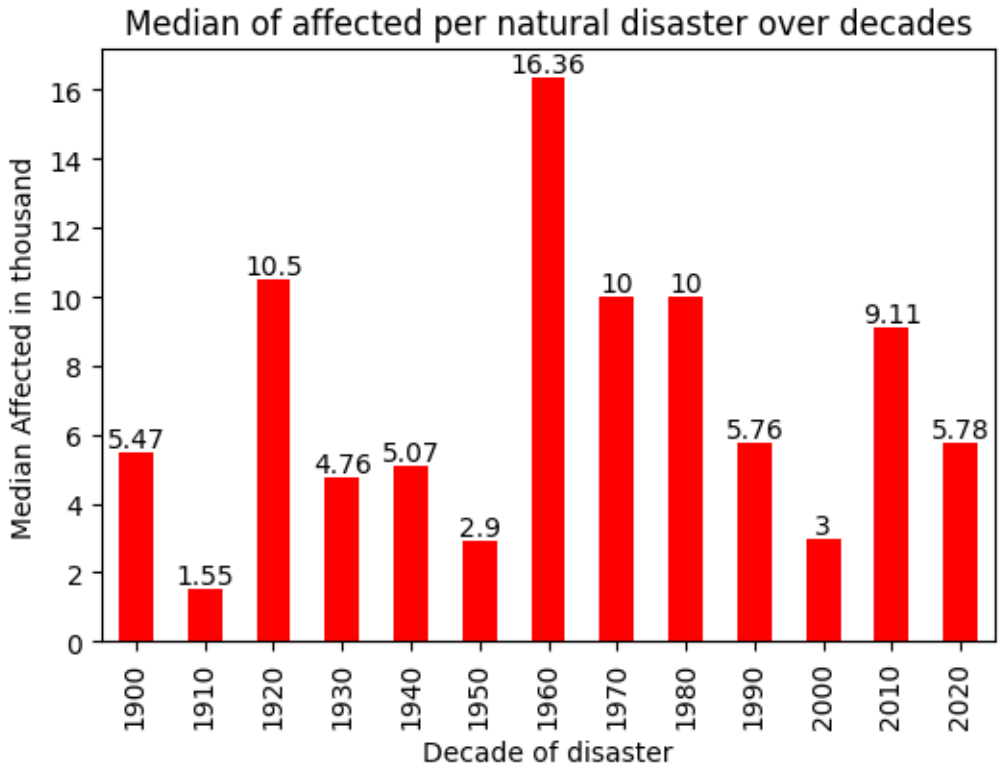
In the last decade, the number of deaths/disasters is of course much smaller, since the data only contain information up to the year 2021. So – assuming that the first 2 years of the 1920s are representative of the decade - this value would have to be multiplied by 5 to get a correct ratio. Additionally it can be seen, that (as the graphics above have shown) the number of deaths per natural disaster decreased in the last century, but the number of natural disasters has definitifly constantly increased. This could either mean, that there are more natural disasters with a lower impact or that e.g. due to better health care or faster help there are just less people dying and the intensity of the disasters did not change that much.

To look a bit more into that, we decided to have a look at the number of affected.

[18]:
```
# calc mean affected, per Decade and plot them in a bar plot
means_aff = round(df_a.groupby(["Disaster_Decade"]).mean()['Total_Affected'].
 ↪round()/1_000_000,2)
#plt.subplot(1, 2,1)
ax = means_aff.plot(kind = 'bar', figsize = (6,4), color ="red")
ax.set_ylabel('Average Affected in million')
ax.set_xlabel ('Decade of disaster')
ax.set_title('Average of affected per natural disaster over decades')
print(means_aff)
ax.bar_label(ax.containers[0])
plt.show()

# calc median deaths, per Decade and plot them in a bar plot
medians_aff = round(df_a.groupby(["Disaster_Decade"]).
 ↪median()['Total_Affected']/1_000,2)
#plt.subplot(1, 2,2)
plt.tight_layout()
ax = medians_aff.plot(kind = 'bar', figsize = (6,4), color ="red")
ax.set_ylabel('Median Affected in thousand')
ax.set_xlabel ('Decade of disaster')
ax.set_title('Median of affected per natural disaster over decades')
ax.bar_label(ax.containers[0])
plt.show()

print(medians_aff)
```

```
Disaster_Decade
1900    0.03
1910    0.29
1920    1.86
1930    0.38
1940    0.08
1950    0.24
1960    0.58
1970    0.96
1980    1.11
1990    0.95
2000    0.64
2010    0.59
2020    0.31
Name: Total_Affected, dtype: float64
```


Average of affected per natural disaster over decades
Average Affected in million
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0.03
0.29
1.86
0.38
0.08
0.24
0.58
0.96
1.11
0.95
0.64
0.59
0.31
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
Decade of disaster


Median of affected per natural disaster over decades
Median Affected in thousand
0
2
4
6
8
10
12
14
16
5.47
1.55
10.5
4.76
5.07
2.9
16.36
10
10
5.76
3
9.11
5.78
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
Decade of disaster

```
Disaster_Decade
1900     5.47
1910     1.55
1920    10.50
1930     4.76
1940     5.07
1950     2.90
1960    16.36
1970    10.00
1980    10.00
1990     5.76
2000     3.00
2010     9.11
2020     5.78
Name: Total_Affected, dtype: float64
```

The number of affected shows a different picture than the one from the number of deaths. This supports our supposition, that less people die but that there are more and more people affected from natural disasters.

We take one quick look why the 1920 decade has so many deaths.

[19]:
```
data_1920 = df_a.loc[df_a["Disaster_Decade"]==1920]
data_1920.nlargest(6,"Total_Deaths")
#df_a.nlargest(15,"Total_Deaths")
```

[19]:

	Disaster_Group	Disaster_Subgroup	Disaster_Type	Disaster_Subtype	\
96	Natural	Climatological	Drought	Drought	
58	Natural	Biological	Epidemic	Bacterial disease	
906	Natural	Climatological	Drought	Drought	
56	Natural	Climatological	Drought	Drought	
59	Natural	Biological	Epidemic	Bacterial disease	
85	Natural	Biological	Epidemic	Viral disease	

	Disaster_Subsubtype	Country	ISO	Region	Continent	\
96	<NA>	China	CHN	Eastern Asia	Asia	
58	<NA>	India	IND	Southern Asia	Asia	
906	<NA>	Soviet Union	SUN	Russian Federation	Europe	
56	<NA>	China	CHN	Eastern Asia	Asia	
59	<NA>	India	IND	Southern Asia	Asia	
85	<NA>	India	IND	Southern Asia	Asia	

	Associated_Dis	Associated_Dis2	Start_Year	End_Year	Total_Deaths	\
96	<NA>	<NA>	1928	1928	3000000.0	
58	<NA>	<NA>	1920	1920	2000000.0	
906	Crop failure	Famine	1921	1921	1200000.0	

```
56            <NA>            <NA>        1920        1920       500000.0
59            <NA>            <NA>        1920        1920       500000.0
85            <NA>            <NA>        1926        1926       423000.0

     Total_Affected  Disaster_Decade
96              NaN             1920
58              NaN             1920
906       5000000.0             1920
56       20000000.0             1920
59              NaN             1920
85              NaN             1920
```

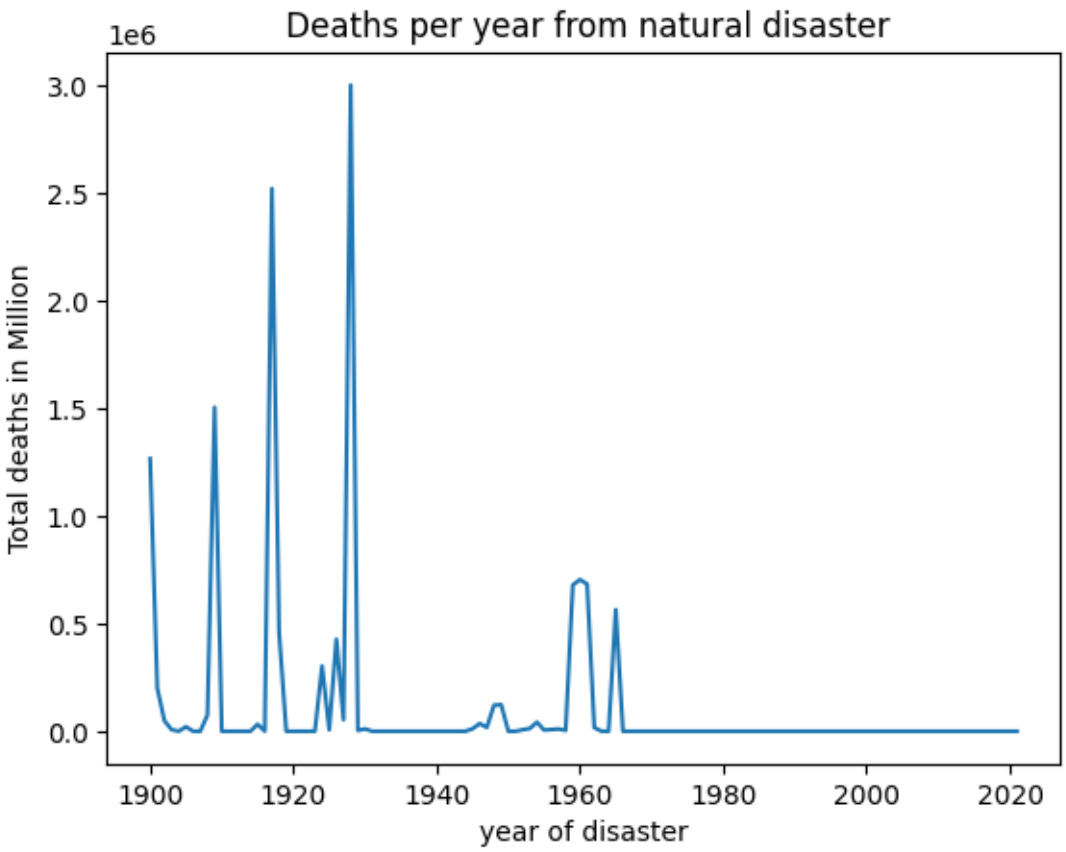
Up until now we have used "Start_Year" as the relevant year. Now let's look at how the results change if we distribute the deaths per event equally over all years (from "Start_Year" to "End_Year)

[20]:
```
df_a_deaths = get_yearly_deaths(df_a, include_zero=True).to_frame().
 ↪reset_index()
df_a_deaths = df_a_deaths[(df_a_deaths['Year'] != 2022)] # filter out data from␣
 ↪2022
# new column (decade)
df_a_deaths['Disaster_Decade'] = df_a_deaths['Year']//10*10
df_a_deaths = df_a_deaths.astype({"Total_Deaths": np.float64, "Year": np.int32,␣
 ↪"Disaster_Decade":np.int32})
df_a_deaths.sum()['Total_Deaths']/(df_a_deaths.max()['Year']-df_a_deaths.
 ↪min()['Year']+1)# check, daran verändert sich nichts
#df_a_deaths
```

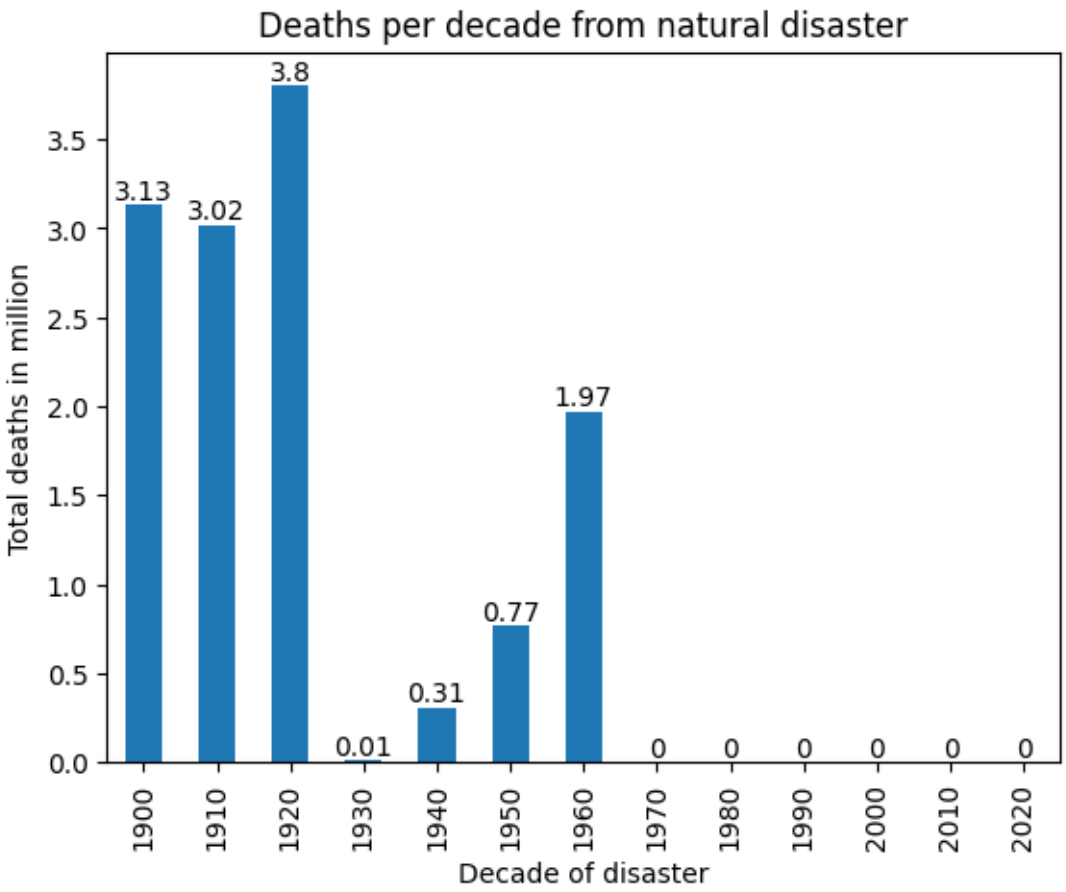
[20]:
```
106593.0081967213
```

[21]:
```
# calc mean deaths, per year and plot them in a bar plot

dth_distr = df_a_deaths.groupby(["Year"]).sum()['Total_Deaths'].round()
ax = dth_distr.plot(kind = 'line')
ax.set_ylabel('Total deaths in Million')
ax.set_xlabel ('year of disaster')
ax.set_title(' Deaths per year from natural disaster')
```

[21]:
```
Text(0.5, 1.0, ' Deaths per year from natural disaster')
```



[22]:
```
# calc deaths per Decade and plot them in a bar plot
sum_distr = round(df_a_deaths.groupby(["Disaster_Decade"]).
 ↪sum()['Total_Deaths']/1_000_000,2)
ax = sum_distr.plot(kind = 'bar')
ax.set_ylabel('Total deaths in million')
ax.set_xlabel ('Decade of disaster')
ax.set_title('Deaths per decade from natural disaster')
ax.bar_label(ax.containers[0])
plt.show()
```



3.1 Conclusions:

The average/median number of deaths per natural disaster over the last century has definitly decreased. The number of deaths per decade has also decreased over the last 122 years. (Always keepining in mind, that the 2020 decade only contains 2 years, not ten as the others. The number of natural disasters itself and also the average/median number of people affected per natural disasters however has increased.

Trend Analysis: Have a look, why the number of deaths in the 1920th is so high (5 from the 15 biggest disasters happend in the 1920th)

- Drought in China (1928): more political than natural (warlords using grain for themselves, less production due to opium plantation); Reference: https://disasterhistory.org/the-northwest-china-famine-1928-1930
- Epidemic in India (1920): Encephalitis lethargica; Reference https://simplifiedupsc.in/epidemics-that-have-hit-india-since-1900/
- Drought in Soviet Union (1921): natural and human caused - (Civil War, Russian Revolution: confiscation of stored grain) ; Reference: https://www.norkarussia.info/famine-1921-1924.html

- Viral disease (1926): Spanish flu brought back from soldiers; Reference: https://simplifiedupsc.in/epidemics-that-have-hit-india-since-1900/
- Drought in China (1920): rainless 12 months - total failure of Harvest; Reference: http://disasterhistory.org/north-china-famine-1920-21

Natural disasters in the early 1920s are partly also consequences of the first world war.

4 b) How does this vary by country? How does this vary by type of natural disaster?

4.1 How does this vary by country?

To find answers for this question, we are going to have two types of world map plots! For both of them, we will have separate maps for each decade.

4.1.1 Plot country's average number of deaths through decades

For the first type of plot, we are plotting each country's number of deaths through decades. The main idea behind this type of plotting is to see, for which decade most of the countries had on average most deaths by natural disasters. Does the concept of Global Warming relates only to frequencies of disasters or also the amount of deaths?

[23]:
```
df_b = df_transformed.copy()
df_b = df_b[df_b["Total_Deaths"].isna() == False]
shapefile = '../data/countries_110m/ne_110m_admin_0_countries.shp'
gdf = gpd.read_file(shapefile)[['ADMIN', 'ADM0_A3', 'geometry']]
gdf.columns = ['country', 'country_code', 'geometry']

df_new = get_yearly_deaths(df_b, custom_index=["ISO"], include_zero=True).
 ↪to_frame()

df_grouped = df_new.groupby([(df_new.index.get_level_values("Year")//
 ↪10)*10+2,"ISO"]).sum()
df_grouped = df_grouped.loc[df_grouped.index.get_level_values("Year") != 2022]
scaler = MinMaxScaler()

df_scaled = df_grouped.groupby(level=1, group_keys=False).apply(lambda x : pd.
 ↪DataFrame(scaler.fit_transform(x), columns=x.columns, index=x.index).
 ↪round(5))

for i in df_scaled.index.get_level_values("Year").unique():
    df_yearly = df_scaled.loc[df_grouped.index.get_level_values("Year") == i]
    merged = gdf.merge(df_yearly.reset_index("ISO"), left_on = 'country_code',␣
 ↪right_on = 'ISO')
    plot_world_map(merged,f"Normalized number of deaths by country for {i} -␣
 ↪{i+10} decade")
    display(merged)
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO

                                              geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.00000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.00000
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00219
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       1.00000
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
..                                                 …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.00000

[163 rows x 5 columns]
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO

                                              geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.00000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.00000
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       1.00000
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.21481
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       1.00000
..                                                 …  …           …
```

```
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.00000

[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO

                                              geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.00000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.00000
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00106
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.30717
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
..                                                 …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.00000

[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO
```

```
                                                geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       1.00000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.00000
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00962
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.35571
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
..                                                 …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.54167

[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                             …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO
```

```
                                                geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI          0.03
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA          0.00
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN          0.00
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA          0.00
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ          0.00
..                                                 …  …          …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH          0.00
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD          0.00
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB          0.00
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE          0.00
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO          0.00

[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
```

```
..                            …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                   Montenegro          MNE
162          Trinidad and Tobago          TTO

                                                  geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.11500
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.00000
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.01110
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.22553
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
..                                                 …  …          …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.00000

[163 rows x 5 columns]
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                            …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                   Montenegro          MNE
162          Trinidad and Tobago          TTO

                                                  geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.01500
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.00532
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00084
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.33822
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
..                                                 …  …          …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       1.00000
```

```
[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO

                                              geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.57500
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.06344
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00076
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.17006
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
..                                                 …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.08333
```

```
[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                    Montenegro          MNE
162          Trinidad and Tobago          TTO

                                              geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.45000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.03444
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00050
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.36990
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.00000
```

```
..                                                   …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.00000

[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                   Montenegro          MNE
162          Trinidad and Tobago          TTO

                                              geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.29500
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       1.00000
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00161
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.33586
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.28667
..                                                   …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       0.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.20833

[163 rows x 5 columns]
```

```
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                   Montenegro          MNE
162          Trinidad and Tobago          TTO
```

```
                                                 geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.35000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.05546
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00141
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.44523
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.12222
..                                                 …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       0.30556
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       1.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       0.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       0.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.12500

[163 rows x 5 columns]
                        country country_code  \
0                          Fiji          FJI
1    United Republic of Tanzania          TZA
2                        Canada          CAN
3       United States of America          USA
4                     Kazakhstan          KAZ
..                           …          …
158       Bosnia and Herzegovina          BIH
159              North Macedonia          MKD
160           Republic of Serbia          SRB
161                   Montenegro          MNE
162          Trinidad and Tobago          TTO

                                                 geometry  ISO  Total_Deaths
0    MULTIPOLYGON (((180.00000 -16.06713, 180.00000…  FJI       0.37000
1    POLYGON ((33.90371 -0.95000, 34.07262 -1.05982…  TZA       0.10065
2    MULTIPOLYGON (((-122.84000 49.00000, -122.9742…  CAN       0.00157
3    MULTIPOLYGON (((-122.84000 49.00000, -120.0000…  USA       0.25106
4    POLYGON ((87.35997 49.21498, 86.59878 48.54918…  KAZ       0.11778
..                                                 …  …           …
158  POLYGON ((18.56000 42.65000, 17.67492 43.02856…  BIH       1.00000
159  POLYGON ((22.38053 42.32026, 22.88137 41.99930…  MKD       1.00000
160  POLYGON ((18.82982 45.90887, 18.82984 45.90888…  SRB       1.00000
161  POLYGON ((20.07070 42.58863, 19.80161 42.50009…  MNE       1.00000
162  POLYGON ((-61.68000 10.76000, -61.10500 10.890…  TTO       0.00000

[163 rows x 5 columns]
```

This map show us perfectly the increasing trend for total number of deaths for each country. When we look at earlier decades, most of the countries were plain yellow. Of course, for some countries there was a problem for gathering the data at that time, or maybe they didn't even exist. Nevertheless, the reality of increasing number of deaths for each country is real.

4.1.2 Plot country's normalized number of deaths through decades

For the second type of plot, we want to see if there are some countries (or even regions) in the world which are proportionally hit the heaviest by the number of deaths. Naturally, for comparing countries between themselves, we are going to have to normalize the number of deaths by the population of each country! For general research, can we recognize some changes in the world by looking beyond country view, are there also for continents some trends?

[24]:
```
df_grouped = df_new.groupby([(df_new.index.get_level_values("Year")//
 ↪10)*10+2,"ISO"]).sum()
df_grouped = df_grouped.loc[df_grouped.index.get_level_values("Year") >= 1962]
df_grouped = df_grouped.loc[df_grouped.index.get_level_values("Year") != 2022]

df_pop = population.groupby([(population.index.get_level_values("Year")//
 ↪10)*10+2,"ISO"]).mean()
df_grouped["Total_Deaths"] = df_grouped["Total_Deaths"]/df_pop["Population"] *␣
 ↪100000000
df_grouped["Total_Deaths"] = np.log(df_grouped["Total_Deaths"])
df_grouped["Total_Deaths"] = pd.to_numeric(df_grouped["Total_Deaths"], errors␣
 ↪='coerce').fillna(0).astype('float64')
df_grouped["Total_Deaths"].replace([np.inf, -np.inf], 0, inplace=True)

df_scaled = df_grouped.groupby(level=0, group_keys=False).apply(lambda x : pd.
 ↪DataFrame(scaler.fit_transform(x), columns=x.columns, index=x.index).
 ↪round(5))

for i in df_scaled.index.get_level_values("Year").unique():
    df_yearly = df_scaled.loc[df_scaled.index.get_level_values("Year") == i]
    merged = gdf.merge(df_yearly.reset_index("ISO"), left_on = 'country_code',␣
 ↪right_on = 'ISO')
    plot_world_map(merged,f"Normalized number of deaths between countries for␣
 ↪{i} - {i+10} decade")
```

By looking at these plots, the first thing we notice is how there are less countries with proportionally small number of deaths, and this is definitely alarming. One of the key takeaways will be not only that but also changes based on different geographical position of countries.

4.2 Conclusion

These types of plots offer us various ways of interpreting them and can also serve as a good specimen for setting hypothesis for testing.

From the first plot we found out that the average number of deaths for each country is slowly increasing, hence, the consequences of GW (Global Warming) span not only increasing number of natural disasters but also total number of deaths. We are stating this because there was at the beginning our assumption, which stated that as the more we looked into the deaths from natural disasters of later years, the less there would be because the world standard is also increasing (new more quality buildings, better healthcare...). We notice that is not the case. From geographical perspective, there is no trend that goes for one specific region. Countries regardless of their position

are getting maxed number of deaths, which also implies that GW is not predominately increasing hits for countries close to the equator but also from all parts of the world.

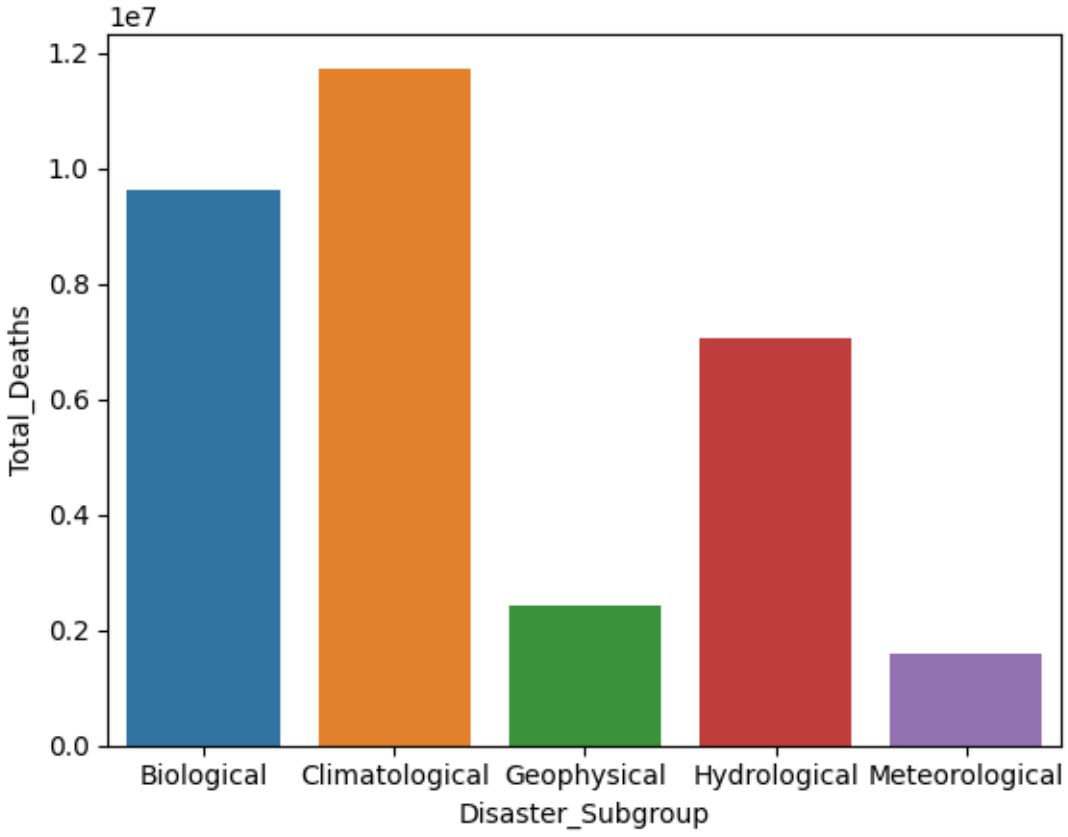
From the second plot, we are trying to see which countries suffer the most proportionally. The most important trend which we can notice is how the countries far away from equator change their color over time. For example, take Europe. In the first decades we see that it is mostly "untouched" but the furthermore we look we can see it changes colors. Also goes for Russia, South Africa... All the countries from the world are experience proportionally more deaths by each decade and to state even worse, the map is getting similar colors for all countries. This is definitely a fight we are all included in!

4.3 How does this vary by type of natural disaster?

To answer this question, we are going to look into various types of visualizations. First, let's see which type of disaster is the deadliest. We are going to check this using bar plots.

[25]:
```
groupby_base = df_b.astype({"Total_Deaths": np.float64, "Total_Affected": np.
 ↪float64, "Start_Year": np.int32})

bars = groupby_base[["Disaster_Subgroup", "Total_Deaths"]].
 ↪groupby(["Disaster_Subgroup"]).sum()
sns.barplot(data=bars.reset_index(), x="Disaster_Subgroup",y="Total_Deaths")
```

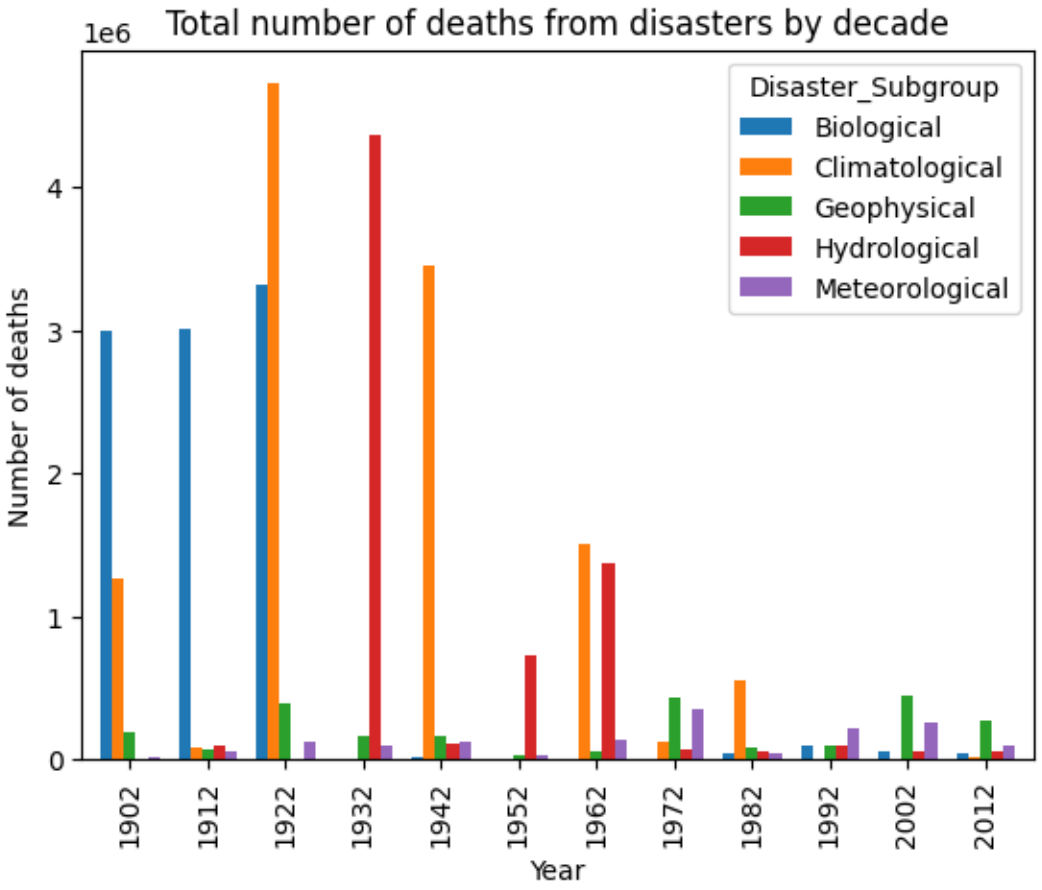
[25]:
```
<AxesSubplot: xlabel='Disaster_Subgroup', ylabel='Total_Deaths'>
```



For the second plot we are going to see for each year total number of deaths by disaster.

[26]:
```
df_new = get_yearly_deaths(df_b, custom_index=["Disaster_Subgroup"],␣
↪include_zero=True).to_frame() # Deaths can be grouped by year and each of␣
↪custom_index
df_grouped = df_new.groupby([(df_new.index.get_level_values("Year")//
↪10)*10+2,"Disaster_Subgroup"]).sum()
df_grouped = df_grouped.loc[df_grouped.index.get_level_values("Year") != 2022]
df_grouped.unstack(level=1)
df_plt = df_grouped.reset_index()
df_plt = df_plt.set_index("Year")
df_plt = df_plt.pivot_table('Total_Deaths', ["Year"], 'Disaster_Subgroup')
df_plt.plot(kind="bar", width=0.75)
plt.title("Total number of deaths from disasters by decade")
plt.ylabel("Number of deaths")
```

[26]:
```
Text(0, 0.5, 'Number of deaths')
```



4.4 Conclusion

From the first visualization, we can see that the deadliest disaster is Climatological, and the least deadly is Geophysical. This is a bit ironic because if we look at the second plot, we can see that the number of deaths from Climatological disasters was rapidly decreasing for decades while Geophysical stayed on average the same for all years. Without a second plot, this would definitely mislead us in our conclusions.

Two of the most common types of Climatological disaster are droughts and wildfires. This information certainly explains why the number of deaths have been decreasing over the years. It certainly is connected to better water supply for all household over the world's and better wildfire localization. Biological is also in decline (pandemics and diseases), and it can be also explained by overall better healthcare worldwide. Hydrological is very oscillating, which can infer that there were very deadly floods back in the time. The good news is that it is also in decline. Meteorological (cyclones and storms) is in a little incline. Alongside Geophysical disaster (earthquakes, landslides and volcanic activity), Meteorological disaster represent the main type of disaster problem these days. The fact that Geophysical's number of deaths is very constant, says that humankind is still struggling to find solution on how to deal with this type of disaster. If we can find common ground between these two disasters, it would for sure be very poor type of constructions where the people live.

5 c) Are there trends visible that could be due to Global Warming?

Responsible: Moritz Renkin (11807211)

Dumping all data before 1950 as it is not relevant to Global Warming.

[27]:
```
cutoff_year = 1950 # TODO check
df_climate = df_transformed[df_transformed["Start_Year"] >= cutoff_year]
min_year = df_climate["Start_Year"].min()
max_year = df_climate["Start_Year"].max()

yearly_global_temp = yearly_global_temp[yearly_global_temp.index >= cutoff_year]
```

Dumping extra-terrestrial and geophysical disasters can be assumed to be indifferent to Global Warming

[28]:
```
df_climate = df_climate[(df_climate["Disaster_Subgroup"] !=␣
 ↪"Extra-terrestrial") & (df_climate["Disaster_Subgroup"] != "Geophysical")]
yearly_disaster_deaths = get_yearly_deaths(df_climate, include_zero=True) #␣
 ↪remove
```

5.1 Regarding Groupby and Pandas Datatypes

Pandas usually uses numpy datatypes. However, **numpy integer arrays do not allow for null values (np.nan)**. That's why pandas introduces their own Integer array, which can inlcude null values (pd.NA). However, for groupby operations these pd.NA values can cause problems, so it is

advisable to convert back to a numpy float array (which does allow for np.nan) before applying the groupby.

[29]:
```
groupby_base = df_climate.astype({"Total_Deaths": np.float64, "Total_Affected":␣
↪np.float64, "Start_Year": np.int32, "Disaster_Decade": np.int32})
groupby_base = groupby_base[groupby_base["Disaster_Decade"] < 2020]
groupby_base[["Disaster_Subgroup", "Total_Deaths"]].
↪groupby(["Disaster_Subgroup"]).mean()
```

[29]:

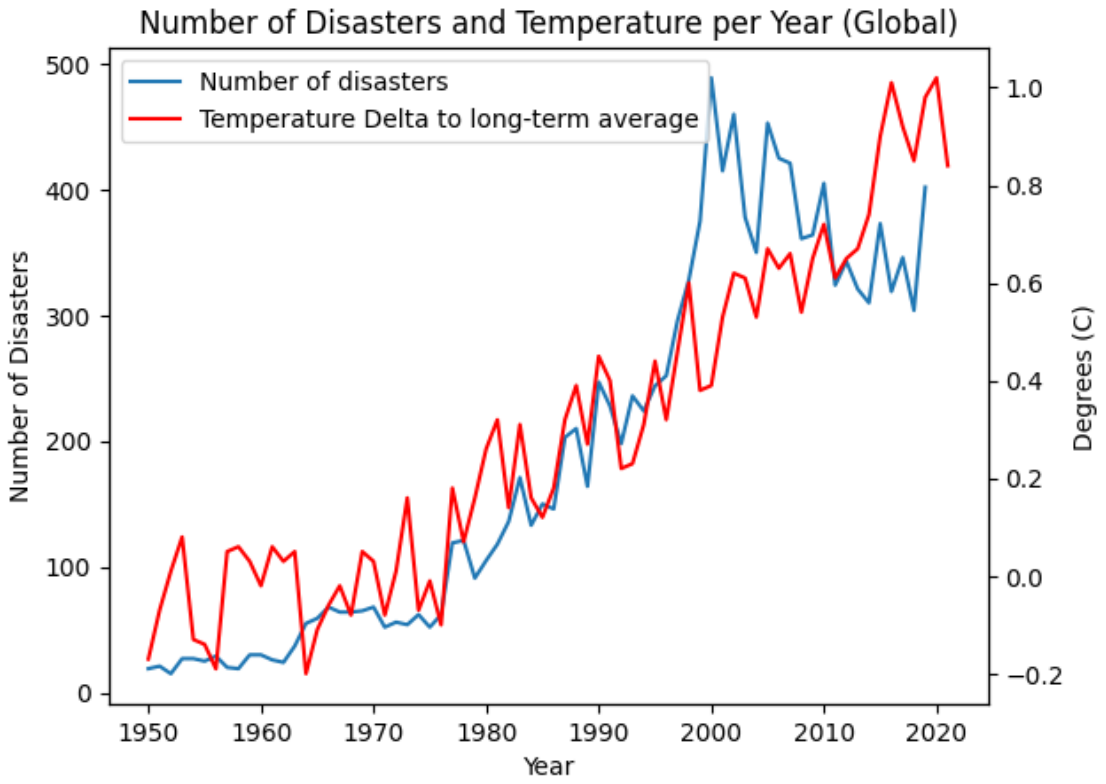
Disaster_Subgroup	Total_Deaths
Biological	215.214689
Climatological	9342.864979
Hydrological	554.114710
Meteorological	332.690490

5.2 Data Exploration

[30]:
```
year_range = range(min_year, max_year)

def fill_missing_year():
    set(groupby_base.groupby("Start_Year").size().rename("No_Disasters").index).
↪difference(year_range)
    # TODO
```

[31]:
```
total_disaster_per_year = get_yearly_disaster_count(groupby_base,␣
↪include_zero=True)

fig, ax_dis = plt.subplots()

plot_dis = ax_dis.plot(total_disaster_per_year.index, total_disaster_per_year,␣
↪label="Number of disasters")
ax_dis.set_ylabel("Number of Disasters")
ax_dis.set_xlabel("Year")
ax_temp = ax_dis.twinx()

plot_temp = ax_temp.plot(yearly_global_temp.index,␣
↪yearly_global_temp["Temperature Delta"], color="r", label="Temperature Delta␣
↪to long-term average")
ax_temp.set_ylabel("Degrees (C)")

plots = plot_dis + plot_temp
plt.legend(plots, [plot.get_label() for plot in plots])
plt.title("Number of Disasters and Temperature per Year (Global)")

plt.show()
```

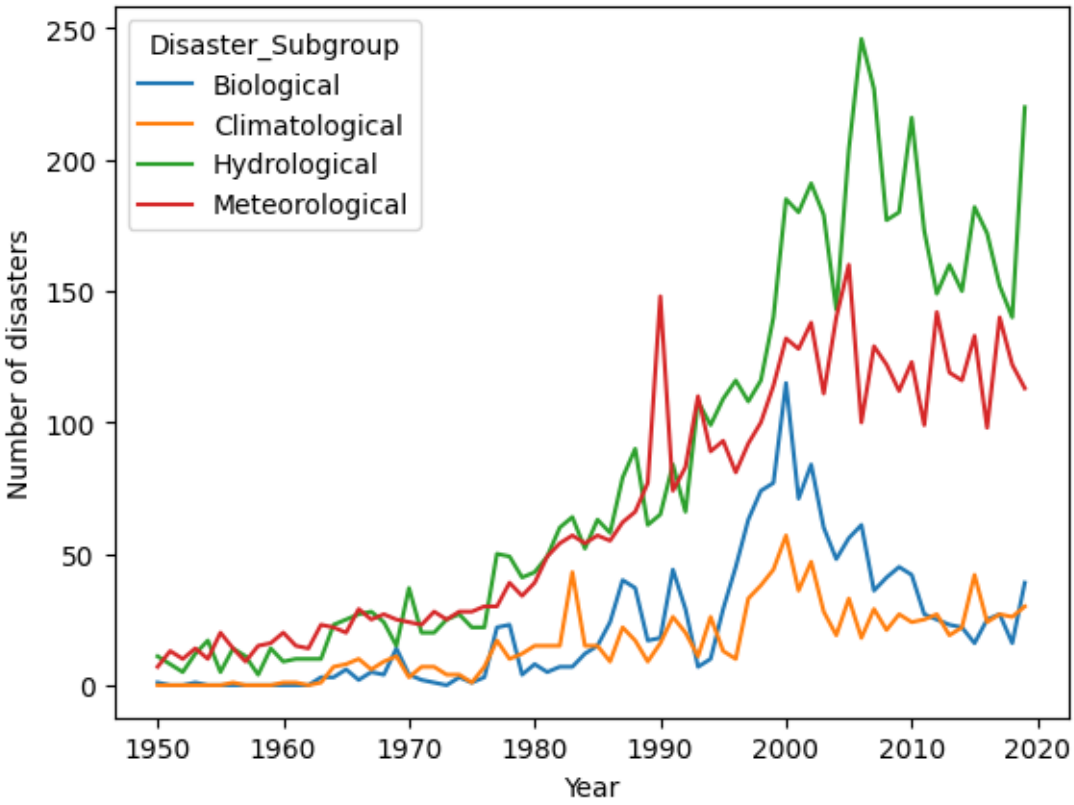
```
print(f"Correlation Coefficient: {total_disaster_per_year.
  ↪corr(yearly_global_temp['Temperature Delta'])}")
```




```
Correlation Coefficient: 0.8886333511376328
```

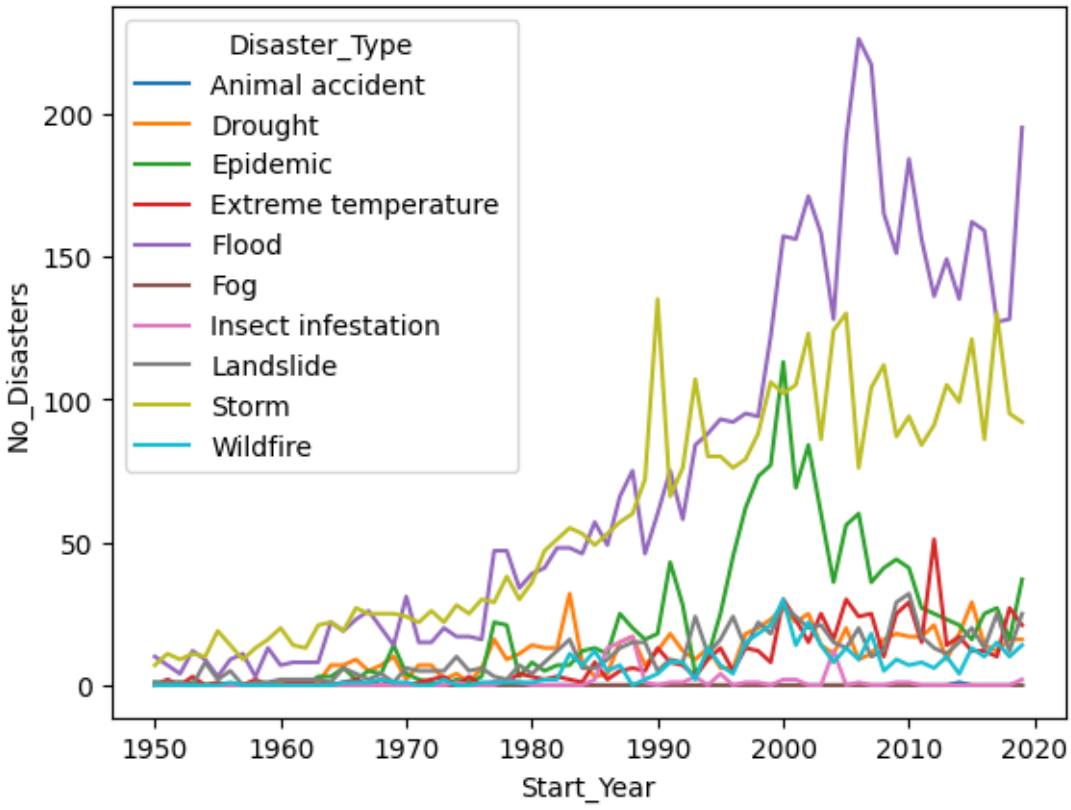
The plot above shows that there has been an overall increase in the number of yearly disaster occurrences in the considered time frame. At the same time, there is a trend of increasing global temperature. The correlation between the two values over the considered time frame is quite high, but this correlation does not prove a causal influence. We will be looking exploring general trends in the different disaster subgroups/types before trying to relate them to Global Warming.

[32]:
```
sns.lineplot(data=get_yearly_disaster_count(groupby_base,␣
  ↪index_cols=["Start_Year","Disaster_Subgroup"]).reset_index(),
             x="Start_Year",
             y="No_Disasters",
             hue="Disaster_Subgroup")
plt.ylabel("Number of disasters")
plt.xlabel("Year")
plt.title("") # TODO
plt.show()
```



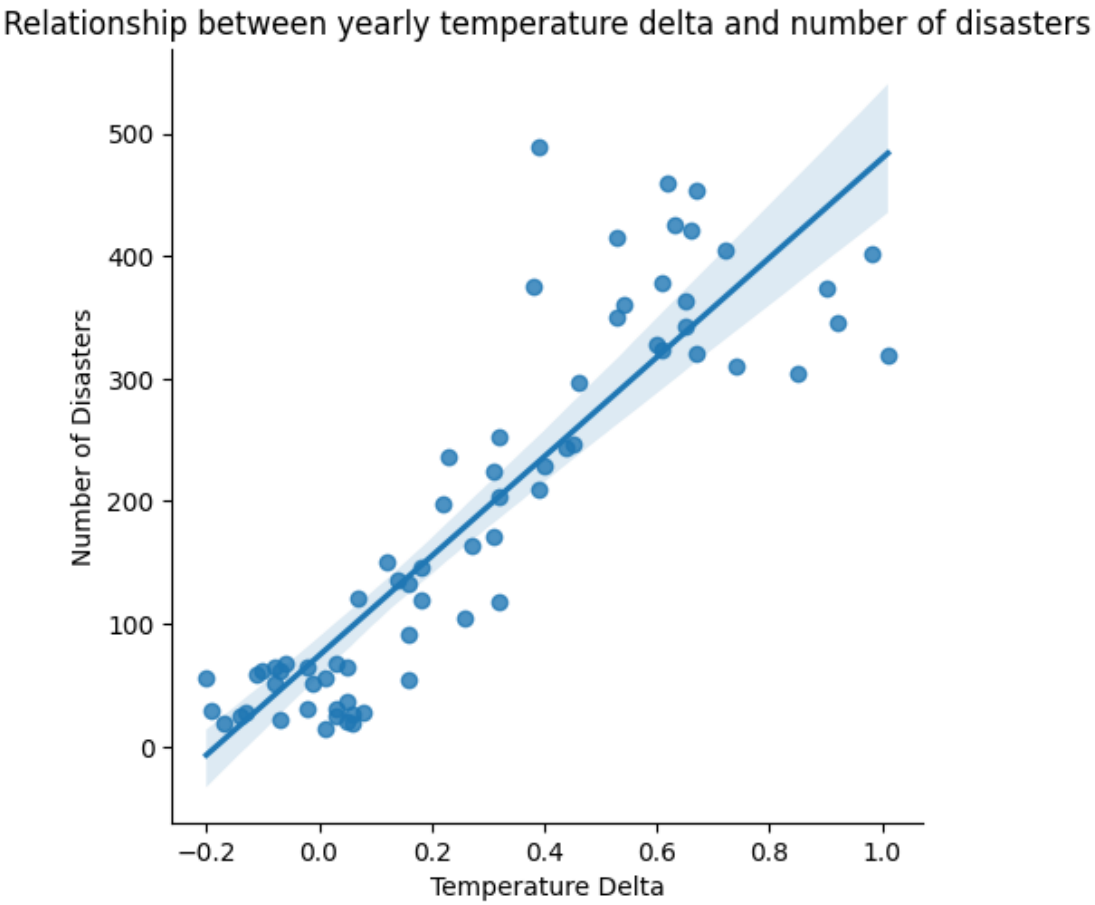
[33]:
```
sns.lineplot(data=get_yearly_disaster_count(groupby_base,␣
 ↪index_cols=["Start_Year","Disaster_Type"]).reset_index(),
             x="Start_Year",
             y="No_Disasters",
             hue="Disaster_Type")
```

[33]: <AxesSubplot: xlabel='Start_Year', ylabel='No_Disasters'>



5.2.1 Global trends related to rising temperature

[34]:
```
yearly_temp_disasters = pd.concat((yearly_global_temp,␣
 ↪get_yearly_disaster_count(groupby_base)), axis="columns")#.
 ↪rename_axis("Year")
yearly_temp_disasters["Disaster_Decade"] = yearly_temp_disasters.index//10*10
sns.lmplot(data=yearly_temp_disasters,
           x="Temperature Delta",
           y="No_Disasters")
plt.xlabel("Temperature Delta")
plt.ylabel("Number of Disasters")
plt.title("Relationship between yearly temperature delta and number of␣
 ↪disasters")
plt.show()
print(f"Correlation coefficient: {yearly_temp_disasters['Temperature Delta'].
 ↪corr(yearly_temp_disasters['No_Disasters'])}")
```



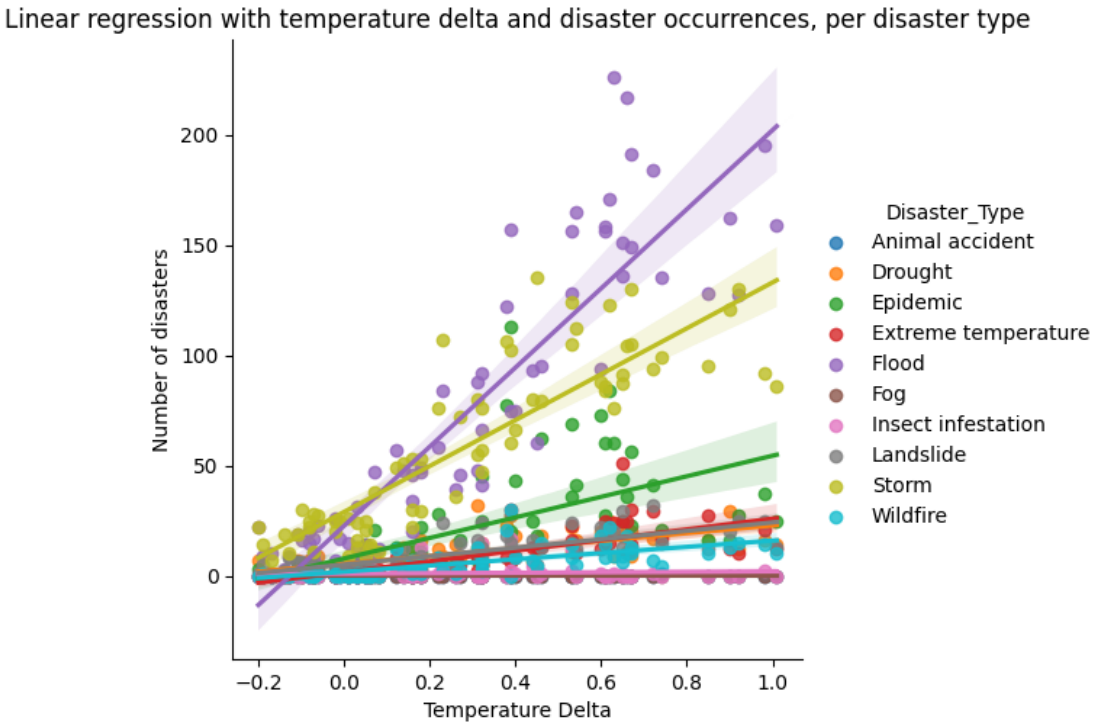

```
Correlation coefficient: 0.8886333511376328
```

In the graph above, a linear regression is depicted to show the supposed impact of rising temperatures on the number of natural disaster occurrences. Specially, the linear regression is performed on the global temperature delta of a given year to long-term average (x-axis) and the Number of natural disasters in that year (y-axis).

[35]:
```
yearly_disaster_temp_by_subgroup: pd.DataFrame =␣
  ↪get_yearly_disaster_count(groupby_base,␣
  ↪index_cols=["Start_Year","Disaster_Type"]).to_frame()
yearly_disaster_temp_by_subgroup["Temperature Delta"] = 0
yearly_disaster_temp_by_subgroup["Temperature Delta"] =␣
  ↪yearly_disaster_temp_by_subgroup["Temperature Delta"].
  ↪add(yearly_temp_disasters["Temperature Delta"].
  ↪rename_axis(index="Start_Year"))#.groupby("Disaster_Decade")["Temperature␣
  ↪Delta"].mean().drop(2020))
yearly_disaster_temp_by_subgroup.reset_index(inplace=True)

sns.lmplot(data=yearly_disaster_temp_by_subgroup,
```

```
            x="Temperature Delta",
            y="No_Disasters",
            hue="Disaster_Type")
plt.title("Linear regression with temperature delta and disaster occurrences,␣
 ↪per disaster type")
plt.ylabel("Number of disasters")
```

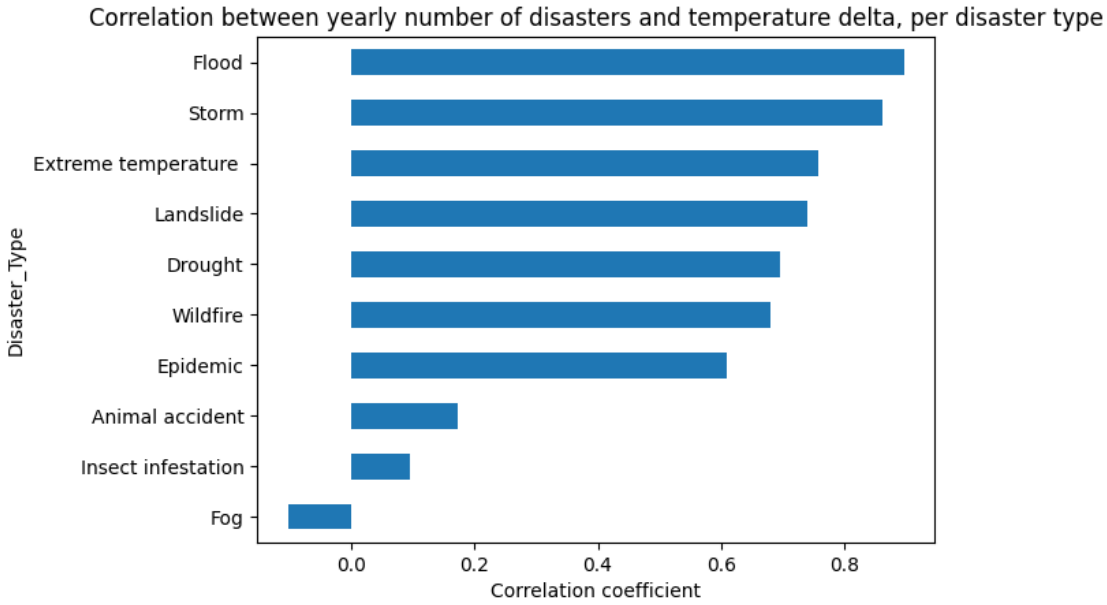
[35]:
```
Text(58.13052343750002, 0.5, 'Number of disasters')
```



Similar to the previous graph, this one tries to show the relationship between increasing temperature and the number of disaster occurrences. For this graphic, though, the regression is performed for each disaster type independently. It has to be noted that the natural disaster occurrences here are in **absolute** numbers, not relative to a long-term average or similar. Disaster type "Flood" and "Storm" have the steepest slope which corresponds to their linear regression coefficients. Note that this does not mean that they also are the most strongly correlated to temperature. We will investigate the correlation per disaster type next.

[36]:
```
subgroup_correlations = yearly_disaster_temp_by_subgroup.
 ↪groupby("Disaster_Type", as_index=True, group_keys=True).apply(lambda␣
 ↪dis_type: dis_type["No_Disasters"].corr(dis_type["Temperature Delta"])).
 ↪rename("Correlation").sort_values(ascending=True)
subgroup_correlations.plot.barh()
plt.xlabel("Correlation coefficient")
```

```
plt.title("Correlation between yearly number of disasters and temperature␣
  ↪delta, per disaster type")
plt.show()
```



Similar to our linear regression models from before, flood and storm occurrences show the highest correlation to the yearly temperature, with both disaster types having a correlation coefficient higher than 0.8.

As noted before, **no causal influence can be assumed due to a strong correlation alone**. We are merely looking at trends here.

5.2.2 Country-specific trends due to temperature increase

In this section, we try to investigate a potential trend of specific countries and their respective frequency of disaster occurrences. The hypothesis is that Countries which experienced a relatively strong warming also have a stronger trend of more frequent natural disaster occurrences. We calculate the linear regression coefficient for Year (X) and number disaster occurrences (y) as a measure of this trend. The result of this calculation is saved in the "Disaster Occurrence Trend" column shown below. The values of this column can be interpreted as an estimator for absolute yearly increase in disaster occurrences.

[37]:
```
yearly_country_disasters = get_yearly_disaster_count(groupby_base,␣
  ↪index_cols=["Start_Year", "Country"], include_zero=True)
disaster_occurrence_trend = yearly_country_disasters.groupby(level="Country",␣
  ↪sort=False).apply(lambda country: linregress(country, country.index.
  ↪get_level_values("Start_Year")).slope).rename("Disaster Occurrence Trend")
```

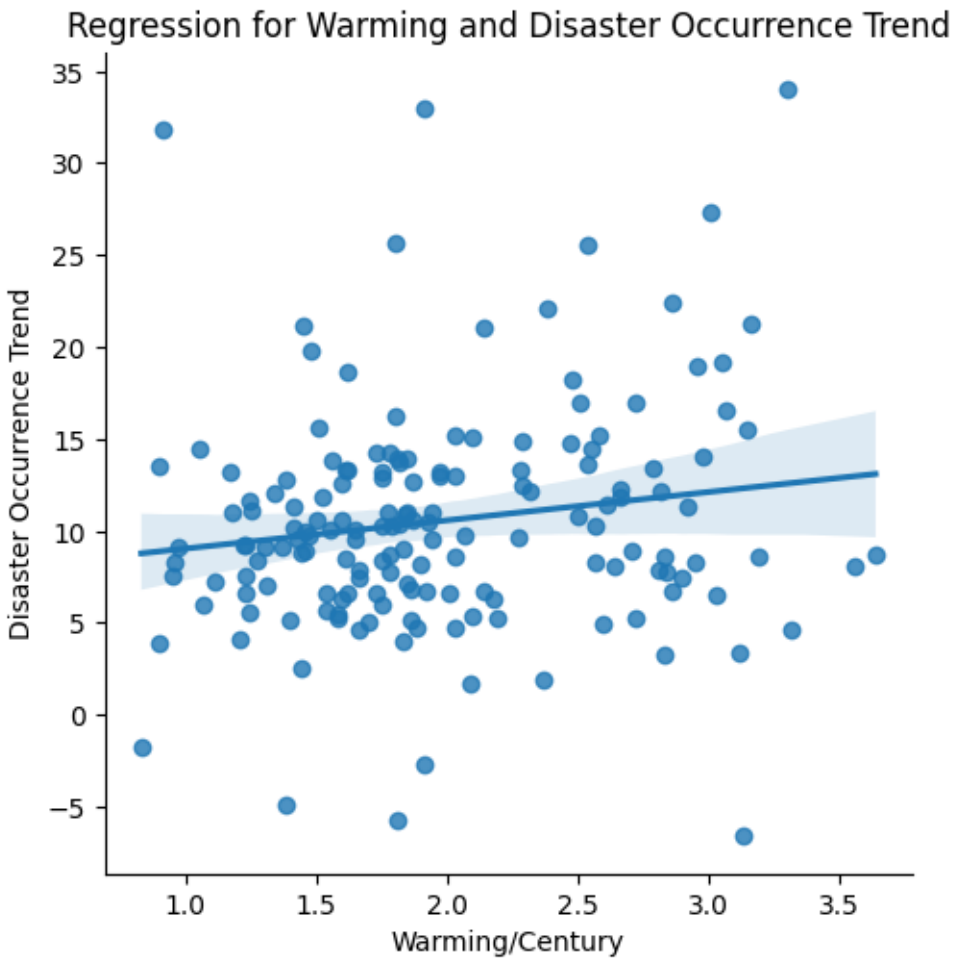
```
yearly_country_deaths = get_yearly_deaths(df_climate, custom_index=["Country"],␣
 ↪include_zero=True)
disaster_death_trend = yearly_country_deaths.groupby(level="Country").
 ↪apply(lambda country: linregress(country.index.get_level_values("Year"),␣
 ↪country).slope).rename("Disaster Deaths Trend")

countries_complete = pd.concat((country_temp_delta, disaster_occurrence_trend,␣
 ↪disaster_death_trend), axis="columns", join="inner")
countries_complete
```

[37]:

Country	Region	Warming/Century	Uncertainty (±)	Disaster Occurrence Trend	Disaster Deaths Trend
Afghanistan	Asia	3.32	0.34	4.642134	7.521014
Albania	Europe	1.97	0.28	13.169659	0.026770
Algeria	Africa	2.86	0.28	6.726465	0.861824
American Samoa	NaN	1.43	0.57	9.584615	-0.053058
Angola	Africa	1.61	0.34	8.512264	3.708808
…	…	…	…	…	…
Uruguay	South America	1.56	0.45	13.841310	0.020194
Uzbekistan	Asia	2.72	0.29	16.969697	0.040067
Yemen	Asia	2.50	0.55	10.784651	1.871374
Zambia	Africa	1.77	0.27	10.966807	0.755016
Zimbabwe	Africa	1.50	0.22	10.571221	5.168773

[163 rows x 5 columns]

[38]:
```
sns.lmplot(data=countries_complete,
           x="Warming/Century",
           y="Disaster Occurrence Trend")
plt.title("Regression for Warming and Disaster Occurrence Trend")
plt.show()
print(f"Correlation coefficient: {countries_complete['Warming/Century'].
 ↪corr(countries_complete['Disaster Occurrence Trend'])}")
```




```
Correlation coefficient: 0.15749030996600916
```

It is pretty obvious that from the scatter plot and regression above that with our chosen method, no influence of a countries warming and disaster occurrence trend can be shown. The correlation coefficient of these two measures is low with ~0.15. Our initial hypothesis is not supported by the data.

5.3 Conclusion

As already discussed when answering the previous two questions, several trends with respect to disaster occurrences are identifiable in the dataset. In this section we delved deeper into trends within the different disaster subgroups and (sub-)types as well as their potential relationship to global warming.

It was shown that yearly flood and storm occurrences correlate most strongly with the global temperature, whereas occurrences of animal accidents, insect infestation and fog have only a slight positive or in the latter case even a negative correlation. For each of these disaster subtypes, a linear regression model was devised, with the results suggesting a relationship between global temperatures and disaster occurrences. However, a causal link between disaster occurrences and

global temperature could not be conclusively demonstrated, since the results could be due to the hardly eliminable survivorship bias in the disaster dataset.

Additionally, the relation between total disaster occurrences and temperature was investigated per country. A separate comprehensive dataset was acquired and transformed but no the result no relationship could be demonstrated, as visibly clear in the last scatter/regression figure.

Although a causal connection between global warming has been shown to causally influence natural disaster occurrences in academic climate research, our research project could only hint at this relationship. We conclude that our research question and the underlying climatological processes are, in hindsight, too complex achieve any conclusively answers within our limited project scope.

6 Appendix

6.1 Disaster Classification according to EM-DAT

Sourced from https://public.emdat.be/about.

Disaster

Group

Disaster

Sub-Group

Disaster

Type

Disaster

Sub-Type

Disaster

Sub-Sub Type

Natural

Geophysical

Earthquake

Ground movement

Tsunami

Volcanic activity

Ash fall

Lahar

Pyroclastic flow

Lava flow

Mass Movement

Meteorological

Storm

Tropical storm

Extra-tropical storm

Convective storm

Derecho

Hail

Lightning/thunderstorm

Rain

Tornado

Sand/dust storm

Winter storm/blizzard

Storm/surge

Wind

Severe Storm

Extreme Temperature

Cold wave

Heat Wave

Severe winter conditions

Snow/ice

Frost/freeze

Fog

Hydrological

Flood

Coastal flood

Riverine flood

Flash flood

Ice jam flood

Landslide

Avalanche (snow, debris, mudflow, rock fall)

Wave action

Rogue wave

Seiche

Climatological

Drought

Drought

Glacial Lake outburst

Wildfire

Forest fires

Land fire: Brush, bush, pasture

Biological

Epidemic

Viral diseases

Bacterial diseases

Parasitic diseases

Fungal diseases

Prion diseases

Insect Infestation

Locust

Grasshopper

Animal accident

Extra-terrestrial

Impact

Airburst

Space weather

Energic particles

Geomagnetic storm

Shockwave