The Efficacy of Body-Mass Index and Child Mortality Rate In Indicating the Causes of Mortality In a Country

By Cao Yueran

Table of Contents

Introduction

Previous studies (https://www.who.int/data/gho/indicator-metadata-registry/imr-details/90) on years of lives lost have shown that in general, the lives of the citizens of more developed countries are cut short due to

degenerative diseases such as stroke and heart disease, while for developing countries, infectious disease still remains a leading cause of death. However, previous measurements of "development" use GDP per capita or qualitative analysis which does not take into account the inequality of a country. **This projects attempts to find a correlation between the causes of mortality and two indexes that are relatively resistant to inequality, BMI and CMR.**

Specific causes of mortality is generally a very difficult type of census data to gather, especially for less financially capable countries. These countries also tend to be the ones most in need of accurate data as to why their citizens are dying young. If BMI and CMR, two widely availabe indexes could accurately indicate a country's developmental status in regards to healthcare (as reflected in causes of mortality), then countries would be able to use them as a basis for targeted healthcare policies instead of causes of mortality.

Methodology

Data Acquisition

Mortality Data

Mortality data is acquired from https://apps.who.int/healthinfo/statistics/mortality/whodpms/ (https://apps.who.int/healthinfo/statistics/mortality/whodpms/). The website is very difficult for a machine to navigate, so 16 .xml files were chosen and downloaded manually, each pertaining to a general cause of death. The .xml files were subsequently converted to .csv files for easy import. Each .xml contains Age-Standardized Death Rates (ASDRs) by country, by year (1979-2016).

The WHO world standard population (https://apps.who.int/healthinfo/statistics/mortality/whodpms/definitions/pop.htm) were used to calibrate the mortality data. The ASDRs were calculated by WHO, so it is using the same formula as the one used in the BMI dataset (also by WHO). This should make the results a bit more indicative.

BMI Data

The WHO BMI dataset (https://apps.who.int/gho/data/view.main.CTRY12461?lang=en) contains 25000 datapoints of the mean male, female and combined BMI of each country, per year (1975-2016). I am only interested in the combined BMI, for the ASDR are combined as well. Within each data cell, it also has the uncertainty bounds, but I will only be using the mean BMI.

CMR Data

The UNICEF child mortality dataset (https://data.unicef.org/wp-content/uploads/2020/09/Infant-mortality-rate_2020.xlsx) contains 20000 datapoints of median child mortality rate by country, by year (1950-2019). It contains both sexes, male and female statistics, but once again we are only interested in both sexes.

Miscellaneous

HDI indexes to determine a country's state of development is taken from
http://hdr.undp.org/sites/default/files/hdro_statistical_data_tables_1_15_d1_d5.xlsxv
(http://hdr.undp.org/sites/default/files/hdro_statistical_data_tables_1_15_d1_d5.xlsxv)

Data Wrangling

Mortality Data

The index for diseases is available locally.

In [1]:

```
1  import numpy as np, pandas as pd
2  import requests
3  from matplotlib import pyplot as plt
4  import seaborn as sns
5  from IPython.display import display, HTML
6
7  pd.options.display.max_rows = 20
8  pd.options.mode.chained_assignment = None
9
10 disease_codes = pd.read_csv("disease_codes.csv", dtype={"disease": str})
11 disease_codes.index = np.arange(1, len(disease_codes) + 1)
12 disease_codes
```

Out[1]:

	disease	name
1	1001	Certain infectious and parasitic diseases
2	1026	Neoplasms
3	1048	Diseases of the blood and blood-forming organs...
4	1051	Endocrine, nutritional and metabolic diseases
5	1055	Mental and behavioural disorders
6	1058	Diseases of the nervous system
7	1062	Diseases of the eye and adnexa
8	1063	Diseases of the ear and mastoid process
9	1064	Diseases of the circulatory system
10	1072	Diseases of the respiratory system

16 .xml files containing ASDRs will now be imported and merged into one DataFrame.

In [2]:

```
forConcat = []
for x in range(1, len(disease_codes) + 1):
    temp = pd.read_csv(str(x) + ".csv", header=1)
    temp.insert(1, "Cause", x)
    forConcat.append(temp)

mdf = pd.concat(forConcat, ignore_index=True)
mdf = mdf.loc[mdf.Countries != "Total reporting countries"]
mdf.rename({"Countries": "Country"}, axis=1, inplace=True)
mdf.reset_index(inplace=True, drop=True)
mdf.replace(" ", np.nan, inplace=True)
mdf
```

Out[2]:

	Country	Cause	2016	2015	2014	2013	2012	2011	2010	2009	...	1988	1987	1986	1985	198
0	Albania	1	NaN	NaN	NaN	NaN	NaN	NaN	1.2	1.1	...	8.9	11.4	NaN	NaN	Nal
1	Anguilla	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	Nal
2	Antigua and Barbuda	1	NaN	28.4	39.5	29.8	38.3	NaN	NaN	21.6	...	15.8	10.2	20.6	11.7	Nal
3	Argentina	1	NaN	23.9	26.0	26.3	25.6	27.0	27.6	27.2	...	25.6	26.5	24.7	23.5	25.
4	Armenia	1	7.1	9.0	8.4	7.4	8.5	8.2	9.8	9.2	...	11.7	15.2	12.8	13.7	Nal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
2251	Uruguay	16	NaN	5.9	5.2	6.3	6.4	NaN	5.1	5.3	...	8.6	9.0	8.2	8.8	9.
2252	Uzbekistan	16	NaN	NaN	2.4	1.7	1.5	1.3	1.3	1.7	...	5.0	6.5	7.4	7.4	Nal

Let us proceed to map the names of causes onto `Cause`.

In [3]:

```
mdf.insert(
    2, "CauseName", mdf.Cause.map(dict(zip(disease_codes.index, disease_codes.name)))
)
mdf.sort_values(by=["Country", "Cause"], inplace=True)
mdf.reset_index(inplace=True, drop=True)
mdf
```

Out[3]:

	Country	Cause	CauseName	2016	2015	2014	2013	2012	2011	2010	...	1988	1987	1986
0	Albania	1	Certain infectious and parasitic diseases	NaN	NaN	NaN	NaN	NaN	NaN	1.2	...	8.9	11.4	NaN
1	Albania	2	Neoplasms	NaN	NaN	NaN	NaN	NaN	NaN	56.4	...	106.9	95.8	NaN
2	Albania	3	Diseases of the blood and blood-forming organs...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
3	Albania	4	Endocrine, nutritional and metabolic diseases	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN

Let us remove the countries with no datapoints from 1979 until 2016. They are of no use to our analysis. The other missing data points would be left in for now, as it may come in handy during the plotting of a choropleth map to show which countries' infrastructure are so bad that they have no mortality data.

In [4]:

```
missingPerCountry = (
    mdf.groupby("Country")
    .apply(lambda x: x.notna().sum())
    .sum(axis=1)
    .sort_values(ascending=False)
)
noDataCountries = list(missingPerCountry[missingPerCountry <= 48].index)
mdf.drop(mdf[mdf.Country.isin(noDataCountries)].index, inplace=True)
mdf.reset_index(inplace=True, drop=True)
mdf
```

Out[4]:

	Country	Cause	CauseName	2016	2015	2014	2013	2012	2011	2010	...	1988	19
0	Albania	1	Certain infectious and parasitic diseases	NaN	NaN	NaN	NaN	NaN	NaN	1.2	...	8.9	1
1	Albania	2	Neoplasms	NaN	NaN	NaN	NaN	NaN	NaN	56.4	...	106.9	9
2	Albania	3	Diseases of the blood and blood-forming organs...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
3	Albania	4	Endocrine, nutritional and metabolic diseases	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
4	Albania	5	Mental and behavioural disorders	NaN	NaN	NaN	NaN	NaN	NaN	1.7	...	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1787	Virgin Islands (USA)	12	Diseases of the skin and subcutaneous tissue	NaN	1.0	NaN	NaN	0.6	1.8	4.1	...	NaN	N
1788	Virgin Islands (USA)	13	Diseases of the musculoskeletal system and con...	NaN	1.6	NaN	NaN	4.3	2.4	2.9	...	NaN	N
1789	Virgin Islands (USA)	14	Diseases of the genitourinary system	NaN	8.3	NaN	NaN	7.7	7.1	13.8	...	NaN	N
1790	Virgin Islands (USA)	15	Pregnancy, childbirth and the puerperium	NaN	0.0	NaN	NaN	0.0	0.0	0.0	...	NaN	N
1791	Virgin Islands (USA)	16	Congenital malformations, deformations and chr...	NaN	0.8	NaN	NaN	2.4	3.5	4.1	...	NaN	N

1792 rows × 41 columns

Reverse the year numbers for easier graph plotting.

In [5]:

```
mdf = mdf[
    ["Country", "Cause", "CauseName"] + list(mdf.columns[:2:-1])
]  # I don't get why this is [:0:-1] instead of [1::-1] but ok.
mdf
```

Out[5]:

	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	1985	...	2007	20
0	Albania	1	Certain infectious and parasitic diseases	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	2.0	2
1	Albania	2	Neoplasms	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	74.2	8(
2	Albania	3	Diseases of the blood and blood-forming organs...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
3	Albania	4	Endocrine, nutritional and metabolic diseases	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
4	Albania	5	Mental and behavioural disorders	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1.8	1
...	...	...	...	...	...	...	...	...	...	...	...	...	
1787	Virgin Islands (USA)	12	Diseases of the skin and subcutaneous tissue	NaN	3.0	NaN	NaN	NaN	NaN	NaN	...	2.5	1
1788	Virgin Islands (USA)	13	Diseases of the musculoskeletal system and con...	NaN	3.2	NaN	NaN	NaN	NaN	NaN	...	3.6	3
1789	Virgin Islands (USA)	14	Diseases of the genitourinary system	NaN	13.9	NaN	NaN	NaN	NaN	NaN	...	8.6	6
1790	Virgin Islands (USA)	15	Pregnancy, childbirth and the puerperium	NaN	14.7	NaN	NaN	NaN	NaN	NaN	...	0.0	(
1791	Virgin Islands (USA)	16	Congenital malformations, deformations and chr...	NaN	8.7	NaN	NaN	NaN	NaN	NaN	...	7.1	4

1792 rows × 41 columns

Let us assign ISO codes for each country and remove the locations that are not countries. We will also remove the rows with not a single data point. I chose to use `get` instead of `search_fuzzy` because the latter is way too slow.

I will also map the codes back onto the name to remove any inconsistencies as pycountry is not a 1-to-1 mapping.

In [6]:

```
import pycountry


def get_country_code(name):
    try:
        return pycountry.countries.lookup(name).alpha_3
    except:
        return None


def get_country_name(code):
    try:
        return pycountry.countries.get(alpha_3=code).name
    except:
        return None


mdf.insert(0, "CountryCode", mdf.Country.apply(get_country_code))
mdf2 = mdf.copy()

mdf.drop(mdf[mdf.CountryCode.isna()].index, inplace=True)
mdf.dropna(thresh=5, inplace=True)
mdf.reset_index(inplace=True, drop=True)
mdf[mdf.CountryCode == "MDK"]
```

Out[6]:

CountryCode	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	...	2007	2

0 rows × 42 columns

We have to check that the countries left out are actually not countries, not just incorrectly named. If they are actually countries, then we should rename them and concatenate them back into `mdf`.

In [7]:

```
leftOut = mdf2[mdf2.CountryCode.isna()]
display(leftOut.Country.value_counts())
countryMapping = {
    "Hong Kong SAR": "Hong Kong",
    "Macau": "Macao",
    "Reunion": "Réunion",
    "Saint Vincent and Grenadines": "Saint Vincent and the Grenadines",
    "Virgin Islands (USA)": "Virgin Islands, U.S.",
    "TFYR Macedonia": "North Macedonia",
    "Iran (Islamic Rep of)": "Iran, Islamic Republic of",
    "Republic of Korea": "Korea, Republic of",
    "Venezuela (Bolivarian Republic of)": "Venezuela, Bolivarian Republic of",
}
leftOut.replace(countryMapping, inplace=True)
leftOut.drop("CountryCode", inplace=True, axis=1)
leftOut.insert(0, "CountryCode", leftOut.Country.apply(get_country_code))
print("number of nan: " + str(leftOut.CountryCode.isna().sum()))

mdf = pd.concat([mdf, leftOut], ignore_index=True)
mdf
```

```
Iran (Islamic Rep of)                16
Venezuela (Bolivarian Republic of)   16
Saint Vincent and Grenadines         16
Macau                                16
Virgin Islands (USA)                 16
TFYR Macedonia                       16
Republic of Korea                    16
Hong Kong SAR                        16
Reunion                              16
Name: Country, dtype: int64

number of nan: 0
```

Out[7]:

	CountryCode	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	.
0	ALB	Albania	1	Certain infectious and parasitic diseases	NaN	NaN	NaN	NaN	NaN	NaN	.
1	ALB	Albania	2	Neoplasms	NaN	NaN	NaN	NaN	NaN	NaN	.
2	ALB	Albania	5	Mental and behavioural disorders	NaN	NaN	NaN	NaN	NaN	NaN	.
3	ALB	Albania	6	Diseases of the nervous system	NaN	NaN	NaN	NaN	NaN	NaN	.
4	ALB	Albania	7	Diseases of the eye and adnexa	NaN	NaN	NaN	NaN	NaN	NaN	.
...	...	...	...	...	...	...	...	...	...	...	.
1776	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	NaN	3.0	NaN	NaN	NaN	NaN	.

	CountryCode	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	...
1777	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	NaN	3.2	NaN	NaN	NaN	NaN	...
1778	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	NaN	13.9	NaN	NaN	NaN	NaN	...
1779	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	NaN	14.7	NaN	NaN	NaN	NaN	...
1780	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	NaN	8.7	NaN	NaN	NaN	NaN	...

In [8]:

```
1 mdf.Country = mdf.CountryCode.apply(get_country_name)
2 mdf.CountryCode.isna().sum()
```

Out[8]:

```
0
```

BMI Data

In [9]:

```
bmidf = pd.read_csv("bmi.csv")
bmidf.drop([0, 1], inplace=True)
bmidf.rename({"Unnamed: 0": "Country"}, axis=1, inplace=True)
bmidf.reset_index(inplace=True, drop=True)
criteria = (bmidf.loc[0] == " Both sexes") | (bmidf.loc[0] == "Country")
bmidf = bmidf[
    criteria.index[criteria]
]  # This is a really good way to boolean index columns
bmidf
```

Out[9]:

	Country	2016	2015	2014	2013	2012	2011	2010	2009	2008	...	1984	19
0	Country	Both sexes	Both sexes	Both sexes	Both sexes	Both sexes	Both sexes	Both sexes	Both sexes	Both sexes	...	Both sexes	Bo sex
1	Afghanistan	23.4 [22.0-24.8]	23.3 [21.9-24.6]	23.2 [21.8-24.5]	23.0 [21.7-24.4]	22.9 [21.6-24.3]	22.8 [21.6-24.1]	22.7 [21.5-24.0]	22.6 [21.4-23.9]	22.5 [21.3-23.8]	...	20.0 [18.3-21.6]	19 [18. 21
2	Albania	26.7 [25.8-27.5]	26.6 [25.8-27.4]	26.5 [25.8-27.2]	26.4 [25.7-27.1]	26.3 [25.6-26.9]	26.2 [25.6-26.8]	26.1 [25.5-26.7]	26.0 [25.5-26.5]	25.9 [25.4-26.4]	...	24.3 [23.1-25.5]	24 [23. 25
3	Algeria	25.5 [24.5-26.5]	25.5 [24.5-26.4]	25.4 [24.5-26.2]	25.3 [24.5-26.1]	25.2 [24.5-26.0]	25.1 [24.4-25.8]	25.1 [24.4-25.7]	25.0 [24.4-25.6]	24.9 [24.3-25.5]	...	22.7 [21.5-23.9]	22 [21. 23
4	Andorra	26.7 [24.6-28.7]	26.7 [24.7-28.7]	26.7 [24.7-28.7]	26.8 [24.8-28.7]	26.8 [24.8-28.7]	26.8 [24.9-28.7]	26.8 [24.9-28.7]	26.8 [25.0-28.7]	26.8 [25.0-28.7]	...	26.0 [24.1-28.0]	26 [24. 27
...	...	...	...	...	...	...	...	...	...	...	...	...	
191	Venezuela (Bolivarian Republic of)	26.7 [26.2-27.2]	26.7 [26.2-27.2]	26.6 [26.2-27.1]	26.6 [26.2-27.1]	26.6 [26.2-27.1]	26.6 [26.1-27.0]	26.6 [26.1-27.0]	26.6 [26.1-27.0]	26.5 [26.1-27.0]	...	24.9 [23.7-26.1]	24 [23. 26
192	Viet Nam	21.9 [21.5-22.3]	21.7 [21.4-22.1]	21.6 [21.3-21.9]	21.5 [21.2-21.7]	21.3 [21.1-21.6]	21.2 [20.9-21.4]	21.0 [20.8-21.3]	20.9 [20.7-21.1]	20.8 [20.6-21.0]	...	18.8 [18.3-19.3]	18 [18. 19
193	Yemen	23.8 [23.1-24.5]	23.7 [23.1-24.3]	23.6 [23.1-24.2]	23.5 [23.1-24.0]	23.4 [23.0-23.9]	23.4 [23.0-23.8]	23.3 [22.9-23.6]	23.2 [22.8-23.5]	23.1 [22.8-23.4]	...	20.6 [19.6-21.7]	20 [19. 21
194	Zambia	22.6 [21.7-23.4]	22.5 [21.6-23.3]	22.4 [21.6-23.2]	22.4 [21.6-23.1]	22.3 [21.6-23.0]	22.2 [21.6-22.9]	22.2 [21.5-22.8]	22.1 [21.5-22.7]	22.0 [21.4-22.6]	...	20.5 [19.6-21.5]	20 [19. 21
195	Zimbabwe	23.8 [23.3-24.3]	23.8 [23.4-24.2]	23.8 [23.4-24.2]	23.7 [23.4-24.1]	23.7 [23.4-24.0]	23.7 [23.3-24.0]	23.6 [23.3-24.0]	23.6 [23.3-23.9]	23.6 [23.3-23.9]	...	22.6 [21.8-23.4]	22 [21. 23

196 rows × 43 columns

As one can see, `bmidf` only contains the BMI values for both sexes, so we can remove that row. We can also remove the uncertainty bounds, keeping only the first number which indicates the mean BMI.

In [10]:

```
bmidf.drop(0, inplace=True)
bmidf.reset_index(inplace=True, drop=True)
bmidf
```

Out[10]:

	Country	2016	2015	2014	2013	2012	2011	2010	2009	2008	...	1984	1983
0	Afghanistan	23.4 [22.0-24.8]	23.3 [21.9-24.6]	23.2 [21.8-24.5]	23.0 [21.7-24.4]	22.9 [21.6-24.3]	22.8 [21.6-24.1]	22.7 [21.5-24.0]	22.6 [21.4-23.9]	22.5 [21.3-23.8]	...	20.0 [18.3-21.6]	19.9 [18.2-21.6]
1	Albania	26.7 [25.8-27.5]	26.6 [25.8-27.4]	26.5 [25.8-27.2]	26.4 [25.7-27.1]	26.3 [25.6-26.9]	26.2 [25.6-26.8]	26.1 [25.5-26.7]	26.0 [25.5-26.5]	25.9 [25.4-26.4]	...	24.3 [23.1-25.5]	24.2 [23.0-25.5]
2	Algeria	25.5 [24.5-26.5]	25.5 [24.5-26.4]	25.4 [24.5-26.2]	25.3 [24.5-26.1]	25.2 [24.5-26.0]	25.1 [24.4-25.8]	25.1 [24.4-25.7]	25.0 [24.4-25.6]	24.9 [24.3-25.5]	...	22.7 [21.5-23.9]	22.6 [21.4-23.9]
3	Andorra	26.7 [24.6-28.7]	26.7 [24.7-28.7]	26.7 [24.7-28.7]	26.8 [24.8-28.7]	26.8 [24.8-28.7]	26.8 [24.9-28.7]	26.8 [24.9-28.7]	26.8 [25.0-28.7]	26.8 [25.0-28.7]	...	26.0 [24.1-28.0]	26.0 [24.0-27.9]
4	Angola	23.3 [21.2-25.6]	23.2 [21.1-25.4]	23.2 [21.1-25.3]	23.1 [21.0-25.2]	23.0 [21.0-25.0]	22.9 [20.9-24.9]	22.8 [20.8-24.8]	22.7 [20.7-24.7]	22.6 [20.6-24.5]	...	19.9 [17.8-21.9]	19.8 [17.6-21.8]
...	...	...	...	...	...	...	...	...	...	...	...	...	...
190	Venezuela (Bolivarian Republic of)	26.7 [26.2-27.2]	26.7 [26.2-27.2]	26.6 [26.2-27.1]	26.6 [26.2-27.1]	26.6 [26.2-27.1]	26.6 [26.1-27.0]	26.6 [26.1-27.0]	26.6 [26.1-27.0]	26.5 [26.1-27.0]	...	24.9 [23.7-26.1]	24.8 [23.6-26.0]
191	Viet Nam	21.9 [21.5-22.3]	21.7 [21.4-22.1]	21.6 [21.3-21.9]	21.5 [21.2-21.7]	21.3 [21.1-21.6]	21.2 [20.9-21.4]	21.0 [20.8-21.3]	20.9 [20.7-21.1]	20.8 [20.6-21.0]	...	18.8 [18.3-19.3]	18.7 [18.2-19.2]
192	Yemen	23.8 [23.1-24.5]	23.7 [23.1-24.3]	23.6 [23.1-24.2]	23.5 [23.1-24.0]	23.4 [23.0-23.9]	23.4 [23.0-23.8]	23.3 [22.9-23.6]	23.2 [22.8-23.5]	23.1 [22.8-23.4]	...	20.6 [19.6-21.7]	20.5 [19.4-21.7]
193	Zambia	22.6 [21.7-23.4]	22.5 [21.6-23.3]	22.4 [21.6-23.2]	22.4 [21.6-23.1]	22.3 [21.6-23.0]	22.2 [21.6-22.9]	22.2 [21.5-22.8]	22.1 [21.5-22.7]	22.0 [21.4-22.6]	...	20.5 [19.6-21.5]	20.5 [19.5-21.5]
194	Zimbabwe	23.8 [23.3-24.3]	23.8 [23.4-24.2]	23.8 [23.4-24.2]	23.7 [23.4-24.1]	23.7 [23.4-24.0]	23.7 [23.3-24.0]	23.6 [23.3-24.0]	23.6 [23.3-23.9]	23.6 [23.3-23.9]	...	22.6 [21.8-23.4]	22.5 [21.6-23.4]

195 rows × 43 columns

In [11]:

```
bmicountries = bmidf.loc[:, "Country"]


def check_float(f):
    try:
        float(f)
        return True
    except ValueError:
        return False


bmidf = bmidf.loc[:, "2016":].applymap(
    lambda x: float(x.split(" ")[0]) if check_float(x.split(" ")[0]) else None
)

bmidf.insert(0, "Country", bmicountries)
bmidf
```

Out[11]:

	Country	2016	2015	2014	2013	2012	2011	2010	2009	2008	...	1984	1983	1982
0	Afghanistan	23.4	23.3	23.2	23.0	22.9	22.8	22.7	22.6	22.5	...	20.0	19.9	19.8
1	Albania	26.7	26.6	26.5	26.4	26.3	26.2	26.1	26.0	25.9	...	24.3	24.2	24.2
2	Algeria	25.5	25.5	25.4	25.3	25.2	25.1	25.1	25.0	24.9	...	22.7	22.6	22.5
3	Andorra	26.7	26.7	26.7	26.8	26.8	26.8	26.8	26.8	26.8	...	26.0	26.0	25.9
4	Angola	23.3	23.2	23.2	23.1	23.0	22.9	22.8	22.7	22.6	...	19.9	19.8	19.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
190	Venezuela (Bolivarian Republic of)	26.7	26.7	26.6	26.6	26.6	26.6	26.6	26.6	26.5	...	24.9	24.8	24.7
191	Viet Nam	21.9	21.7	21.6	21.5	21.3	21.2	21.0	20.9	20.8	...	18.8	18.7	18.6
192	Yemen	23.8	23.7	23.6	23.5	23.4	23.4	23.3	23.2	23.1	...	20.6	20.5	20.4
193	Zambia	22.6	22.5	22.4	22.4	22.3	22.2	22.2	22.1	22.0	...	20.5	20.5	20.4
194	Zimbabwe	23.8	23.8	23.8	23.7	23.7	23.7	23.6	23.6	23.6	...	22.6	22.5	22.4

195 rows × 43 columns

Similar to mortality data, BMI data also has some country names inconsistent with the ISO standard. We should be able to fix it using the same map as the one before.

In [12]:

```
bmidf.replace(countryMapping, inplace=True)
bmidf.insert(0, "CountryCode", bmidf.Country.apply(get_country_code))
bmidf.dropna(inplace=True)
bmidf.Country = bmidf.CountryCode.apply(get_country_name)
display(bmidf.CountryCode.isna().sum())
display(bmidf)
```

0

	CountryCode	Country	2016	2015	2014	2013	2012	2011	2010	2009	...	1984	198
0	AFG	Afghanistan	23.4	23.3	23.2	23.0	22.9	22.8	22.7	22.6	...	20.0	19.
1	ALB	Albania	26.7	26.6	26.5	26.4	26.3	26.2	26.1	26.0	...	24.3	24.
2	DZA	Algeria	25.5	25.5	25.4	25.3	25.2	25.1	25.1	25.0	...	22.7	22.
3	AND	Andorra	26.7	26.7	26.7	26.8	26.8	26.8	26.8	26.8	...	26.0	26.
4	AGO	Angola	23.3	23.2	23.2	23.1	23.0	22.9	22.8	22.7	...	19.9	19.
...	...	...	...	...	...	...	...	...	...	...	...	...	.
190	VEN	Venezuela, Bolivarian Republic of	26.7	26.7	26.6	26.6	26.6	26.6	26.6	26.6	...	24.9	24.
191	VNM	Viet Nam	21.9	21.7	21.6	21.5	21.3	21.2	21.0	20.9	...	18.8	18.
192	YEM	Yemen	23.8	23.7	23.6	23.5	23.4	23.4	23.3	23.2	...	20.6	20.
193	ZMB	Zambia	22.6	22.5	22.4	22.4	22.3	22.2	22.2	22.1	...	20.5	20.
194	ZWE	Zimbabwe	23.8	23.8	23.8	23.7	23.7	23.7	23.6	23.6	...	22.6	22.

186 rows × 44 columns

Also reverse the year numbers to make plotting graphs easier.

In [13]:

```
bmidf = bmidf[["CountryCode", "Country"] + list(bmidf.columns[:1:-1])]
bmidf
```

Out[13]:

	CountryCode	Country	1975	1976	1977	1978	1979	1980	1981	1982	...	2007	200
0	AFG	Afghanistan	18.9	19.0	19.2	19.3	19.4	19.5	19.6	19.8	...	22.4	22.
1	ALB	Albania	23.8	23.8	23.9	23.9	24.0	24.1	24.1	24.2	...	25.8	25.
2	DZA	Algeria	21.9	22.0	22.0	22.1	22.2	22.3	22.4	22.5	...	24.8	24.
3	AND	Andorra	25.4	25.5	25.6	25.6	25.7	25.8	25.9	25.9	...	26.8	26.
4	AGO	Angola	18.8	18.9	19.0	19.2	19.3	19.4	19.5	19.6	...	22.5	22.
...	...	...	...	...	...	...	...	...	...	...	...	...	.
190	VEN	Venezuela, Bolivarian Republic of	23.9	24.0	24.2	24.3	24.4	24.5	24.6	24.7	...	26.5	26.
191	VNM	Viet Nam	18.2	18.2	18.3	18.4	18.4	18.5	18.6	18.6	...	20.7	20.
192	YEM	Yemen	19.7	19.8	19.9	20.0	20.1	20.2	20.3	20.4	...	23.0	23.
193	ZMB	Zambia	19.5	19.6	19.8	19.9	20.0	20.1	20.2	20.4	...	22.0	22.
194	ZWE	Zimbabwe	22.0	22.1	22.1	22.2	22.3	22.3	22.4	22.4	...	23.5	23.

186 rows × 44 columns

Remove data prior to 1979:

In [14]:

```
bmidf.drop([str(x) for x in range(1975, 1979)], axis=1, inplace=True)
bmidf.reset_index(inplace=True, drop=True)
bmidf
```

Out[14]:

	CountryCode	Country	1979	1980	1981	1982	1983	1984	1985	1986	...	2007	200
0	AFG	Afghanistan	19.4	19.5	19.6	19.8	19.9	20.0	20.1	20.2	...	22.4	22.
1	ALB	Albania	24.0	24.1	24.1	24.2	24.2	24.3	24.3	24.4	...	25.8	25.
2	DZA	Algeria	22.2	22.3	22.4	22.5	22.6	22.7	22.8	22.9	...	24.8	24.
3	AND	Andorra	25.7	25.8	25.9	25.9	26.0	26.0	26.1	26.2	...	26.8	26.
4	AGO	Angola	19.3	19.4	19.5	19.6	19.8	19.9	20.0	20.1	...	22.5	22.
...	...	...	...	...	...	...	...	...	...	...	...	...	.
181	VEN	Venezuela, Bolivarian Republic of	24.4	24.5	24.6	24.7	24.8	24.9	25.0	25.1	...	26.5	26.
182	VNM	Viet Nam	18.4	18.5	18.6	18.6	18.7	18.8	18.8	18.9	...	20.7	20.
183	YEM	Yemen	20.1	20.2	20.3	20.4	20.5	20.6	20.7	20.8	...	23.0	23.
184	ZMB	Zambia	20.0	20.1	20.2	20.4	20.5	20.5	20.6	20.7	...	22.0	22.
185	ZWE	Zimbabwe	22.3	22.3	22.4	22.4	22.5	22.6	22.6	22.7	...	23.5	23.

186 rows × 40 columns

Check for missing data:

In [15]:

```
display(bmidf.dtypes.value_counts())
display(bmidf.isna().sum().sum())
```

```
float64    38
object      2
dtype: int64
```

```
0
```

CMR Data

In [16]:

```
cmrdf = pd.read_excel("cmr.xlsx", sheet_name=0, header=14)
cmrdf = cmrdf.loc[:584]
cmrdf
```

Out[16]:

	ISO.Code	Country.Name	Uncertainty.Bounds*	1950.5	1951.5	1952.5	1953.5	195
0	AFG	Afghanistan	Lower	NaN	NaN	NaN	NaN	N
1	AFG	Afghanistan	Median	NaN	NaN	NaN	NaN	N
2	AFG	Afghanistan	Upper	NaN	NaN	NaN	NaN	N
3	ALB	Albania	Lower	NaN	NaN	NaN	NaN	N
4	ALB	Albania	Median	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	
580	ZMB	Zambia	Median	NaN	NaN	NaN	138.522662	135.831
581	ZMB	Zambia	Upper	NaN	NaN	NaN	183.501394	171.798
582	ZWE	Zimbabwe	Lower	NaN	NaN	NaN	NaN	79.678
583	ZWE	Zimbabwe	Median	NaN	NaN	NaN	NaN	104.305
584	ZWE	Zimbabwe	Upper	NaN	NaN	NaN	NaN	139.889

585 rows × 73 columns

Note that we only need the `Median` rows of CMR. I will also standardize the column names.

In [17]:

```
cmrdf = cmrdf[cmrdf["Uncertainty.Bounds*"] == "Median"]
cmrdf.reset_index(inplace=True, drop=True)
cmrdf.drop("Uncertainty.Bounds*", axis=1, inplace=True)
cmrdf.rename(
    {"ISO.Code": "CountryCode", "Country.Name": "Country"}, axis=1, inplace=True
)
cmrdf
```

Out[17]:

	CountryCode	Country	1950.5	1951.5	1952.5	1953.5	1954.5	1955.
0	AFG	Afghanistan	NaN	NaN	NaN	NaN	NaN	Na
1	ALB	Albania	NaN	NaN	NaN	NaN	NaN	Na
2	DZA	Algeria	NaN	NaN	NaN	NaN	146.488512	146.23084
3	AND	Andorra	NaN	NaN	NaN	NaN	NaN	Na
4	AGO	Angola	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	
190	VEN	Venezuela (Bolivarian Republic of)	NaN	81.653269	78.394203	75.444734	72.594348	69.93181
191	VNM	Viet Nam	NaN	NaN	NaN	NaN	NaN	Na
192	YEM	Yemen	NaN	NaN	NaN	NaN	NaN	Na
193	ZMB	Zambia	NaN	NaN	NaN	138.522662	135.831489	133.22854
194	ZWE	Zimbabwe	NaN	NaN	NaN	NaN	104.305814	102.55803

195 rows × 72 columns

Let us standardize the year naming scheme.

In [18]:

```
cmrdf.columns = cmrdf.columns.str.replace("\.5", "")
cmrdf
```

Out[18]:

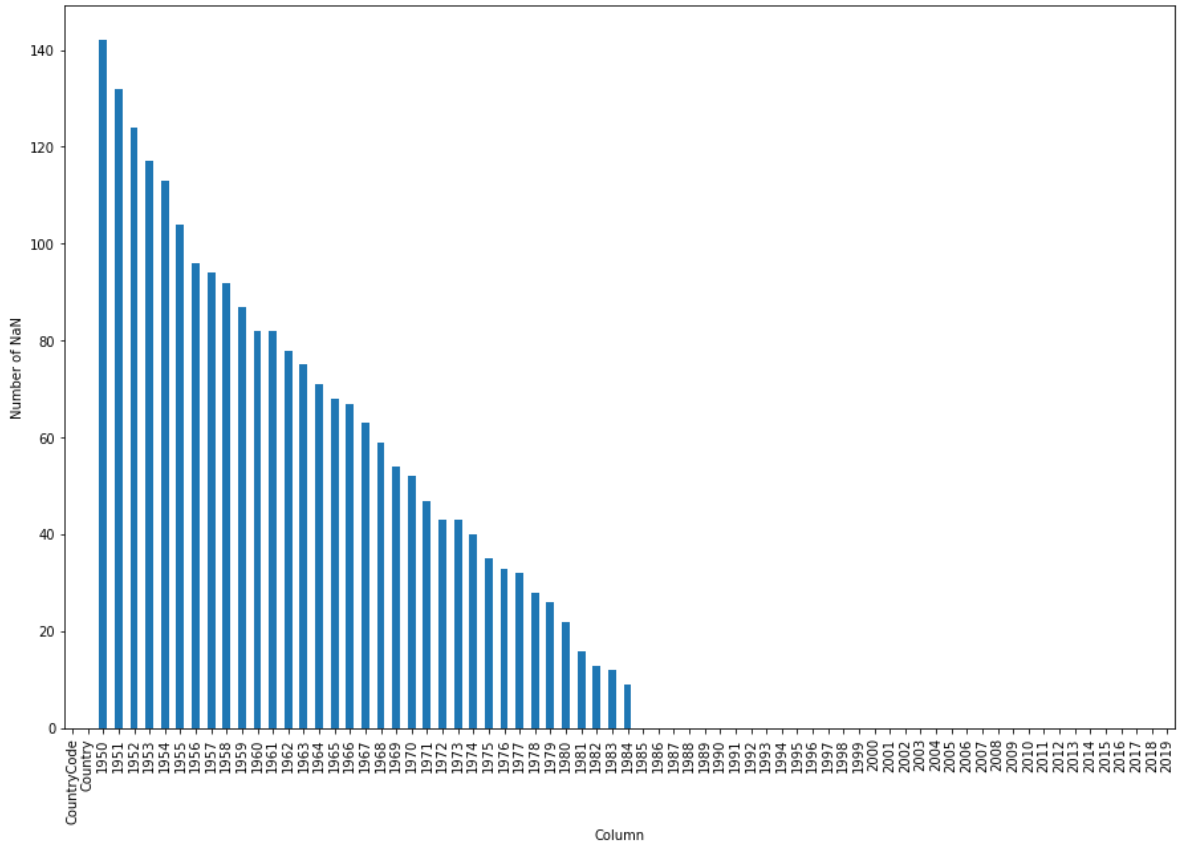
	CountryCode	Country	1950	1951	1952	1953	1954	1955
0	AFG	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN
1	ALB	Albania	NaN	NaN	NaN	NaN	NaN	NaN
2	DZA	Algeria	NaN	NaN	NaN	NaN	146.488512	146.230842
3	AND	Andorra	NaN	NaN	NaN	NaN	NaN	NaN
4	AGO	Angola	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
190	VEN	Venezuela (Bolivarian Republic of)	NaN	81.653269	78.394203	75.444734	72.594348	69.931813
191	VNM	Viet Nam	NaN	NaN	NaN	NaN	NaN	NaN
192	YEM	Yemen	NaN	NaN	NaN	NaN	NaN	NaN
193	ZMB	Zambia	NaN	NaN	NaN	138.522662	135.831489	133.228545
194	ZWE	Zimbabwe	NaN	NaN	NaN	NaN	104.305814	102.558033

195 rows × 72 columns

In [19]:

```
display(
    cmrdf.isna()
    .sum()
    .plot(kind="bar", figsize=(15, 10), xlabel="Column", ylabel="Number of NaN")
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x24405aab4a8>
```



The NaN values seem to be concentrated towards the datapoints taken earlier, mostly before the 1980s. Data before 1979 is useless anyways, so there shouldn't be too much impact leaving a few nans in.

Let us remove the data prior to 1979 and after 2016.

In [20]:

```
cmrdf.drop(
    [str(x) for x in list(range(1950, 1979)) + list(range(2017, 2020))],
    axis=1,
    inplace=True,
)
cmrdf
```

Out[20]:

	CountryCode	Country	1979	1980	1981	1982	1983	
0	AFG	Afghanistan	165.257379	161.028132	156.763436	152.481639	148.354144	14
1	ALB	Albania	71.181438	66.155752	61.475555	57.115470	53.214275	4
2	DZA	Algeria	108.045384	101.794747	94.546163	86.274384	77.023313	6
3	AND	Andorra	NaN	NaN	NaN	NaN	NaN	
4	AGO	Angola	NaN	140.238393	138.636313	137.023220	135.410888	13
...	...	...	...	...	...	...	...	
190	VEN	Venezuela (Bolivarian Republic of)	36.571065	35.186548	33.864139	32.649864	31.545204	3
191	VNM	Viet Nam	47.531900	46.631146	45.737060	44.938104	44.172720	4
192	YEM	Yemen	147.133742	139.228105	131.385962	123.940729	117.077235	11
193	ZMB	Zambia	95.096937	95.596715	96.032758	96.717562	97.831035	9
194	ZWE	Zimbabwe	70.143121	68.574404	66.204777	63.317586	60.084513	5

195 rows × 40 columns

`cmrdf` suffers from the same problem of iso-uncompliant country names, so let's fix that right now.

In [21]:

```
cmrdf.Country = cmrdf.CountryCode.apply(get_country_name)
display(cmrdf)
```

	CountryCode	Country	1979	1980	1981	1982	1983	
0	AFG	Afghanistan	165.257379	161.028132	156.763436	152.481639	148.354144	14
1	ALB	Albania	71.181438	66.155752	61.475555	57.115470	53.214275	4
2	DZA	Algeria	108.045384	101.794747	94.546163	86.274384	77.023313	6
3	AND	Andorra	NaN	NaN	NaN	NaN	NaN	
4	AGO	Angola	NaN	140.238393	138.636313	137.023220	135.410888	13
...	...	...	...	...	...	...	...	
190	VEN	Venezuela, Bolivarian Republic of	36.571065	35.186548	33.864139	32.649864	31.545204	3
191	VNM	Viet Nam	47.531900	46.631146	45.737060	44.938104	44.172720	4
192	YEM	Yemen	147.133742	139.228105	131.385962	123.940729	117.077235	11
193	ZMB	Zambia	95.096937	95.596715	96.032758	96.717562	97.831035	9
194	ZWE	Zimbabwe	70.143121	68.574404	66.204777	63.317586	60.084513	5

195 rows × 40 columns

HDI Data

In [22]:

```
df = pd.read_excel("hdi.xlsx", sheet_name=0)
with pd.option_context("display.max_rows", None):
    display(df)
```

	Unnamed: 0	Table 1. Human Development Index and its components	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unna
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	SDG3	NaN	SDG4.3	NaN	Sl
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	Human development index (HDI)	NaN	Life expectancy at birth	NaN	Expected years of schooling	NaN	ye sch
4	HDI rank	Country	(index value)	NaN	(years)	NaN	(years)	NaN	(
5	NaN	NaN	2018	NaN	2018	NaN	2018	a	

Why doesn't a single database use standardised country codes?????? This is the same thing as before, only this time I was too lazy to type out the entire mapping so I just used pycountry's `fuzzy_search` instead. The country codes and names have been standardised, so it should be the same as that of the `df`s before it.

In [23]:

```
hdidf = df.iloc[7:199, [1, 2]].reset_index(drop=True)
hdidf.columns = ["Country", "HDI"]
hdidf.Country.replace(
    {"Congo (Democratic Republic of the)": "Congo, the Democratic Republic of the"},
    inplace=True,
)
hdidf.insert(0, "CountryCode", hdidf.Country.apply(get_country_code))
not_matched = hdidf[hdidf.CountryCode.isna()]
hdidf = hdidf[~hdidf.CountryCode.isna()]
display(hdidf)
not_matched = not_matched[~not_matched.Country.str.contains("DEVELOPMENT")]
not_matched.Country = not_matched.Country.str.split(",|\(", expand=True)[0]
not_matched.CountryCode = not_matched.Country.apply(
    lambda x: pycountry.countries.search_fuzzy(x)[0].alpha_3
)
hdidf = pd.concat([hdidf, not_matched])
hdidf.Country = hdidf.CountryCode.apply(get_country_name)
hdidf.sort_values(by="CountryCode", inplace=True)
hdidf.reset_index(inplace=True, drop=True)
display(not_matched)
display(hdidf)
```

	CountryCode	Country	HDI
0	NOR	Norway	0.953688
1	CHE	Switzerland	0.945936
2	IRL	Ireland	0.942473
3	DEU	Germany	0.938785
5	AUS	Australia	0.938379
...	...	...	...
187	BDI	Burundi	0.422882
188	SSD	South Sudan	0.41277
189	TCD	Chad	0.401176
190	CAF	Central African Republic	0.380662
191	NER	Niger	0.376591

180 rows × 3 columns

	CountryCode	Country	HDI
4	HKG	Hong Kong	0.938809
22	PRK	Korea	0.905832
65	IRN	Iran	0.797483
97	VEN	Venezuela	0.725773
107	MDA	Moldova	0.711452
114	BOL	Bolivia	0.702842
137	FSM	Micronesia	0.614158
140	SWZ	Eswatini	0.608082

	CountryCode	Country	HDI

	CountryCode	Country	HDI
0	AFG	Afghanistan	0.49596
1	AGO	Angola	0.574488
2	ALB	Albania	0.791406
3	AND	Andorra	0.856781
4	ARE	United Arab Emirates	0.866438
...	...	...	...
184	WSM	Samoa	0.706771
185	YEM	Yemen	0.462717
186	ZAF	South Africa	0.704937
187	ZMB	Zambia	0.591462
188	ZWE	Zimbabwe	0.5631

189 rows × 3 columns

Post-Cleaning

Data cleaning is now complete, and the datasets are as follows:

In [24]:

```
display(HTML("<h5>Mortality Causes DataFrame: </h5>"))
display(mdf)
display(HTML("<h5>BMI DataFrame: </h5>"))
display(bmidf)
display(HTML("<h5>CMR DataFrame: </h5>"))
display(cmrdf)
display(HTML("<h5>HDI DataFrame: </h5>"))
display(hdidf)
```

Mortality Causes DataFrame:

Let us create some long versions of the DataFrames as well.

In [25]:

```
longmdf = pd.melt(
    mdf,
    id_vars=["CountryCode", "Country", "Cause", "CauseName"],
    var_name="Year",
    value_name="DeathsPer100k",
)
longbmidf = pd.melt(
    bmidf, id_vars=["CountryCode", "Country"], var_name="Year", value_name="BMI"
)
longcmrdf = pd.melt(
    cmrdf, id_vars=["CountryCode", "Country"], var_name="Year", value_name="CMR"
)
display(longmdf)
display(longbmidf)
display(longcmrdf)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	NaN
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	NaN
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	NaN
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	NaN
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	NaN

67678 rows × 6 columns

	CountryCode	Country	Year	BMI
0	AFG	Afghanistan	1979	19.4
1	ALB	Albania	1979	24.0
2	DZA	Algeria	1979	22.2
3	AND	Andorra	1979	25.7
4	AGO	Angola	1979	19.3
...	...	...	...	...
7063	VEN	Venezuela, Bolivarian Republic of	2016	26.7
7064	VNM	Viet Nam	2016	21.9

	CountryCode	Country	Year	BMI
7065	YEM	Yemen	2016	23.8
7066	ZMB	Zambia	2016	22.6
7067	ZWE	Zimbabwe	2016	23.8

7068 rows × 4 columns

	CountryCode	Country	Year	CMR
0	AFG	Afghanistan	1979	165.257379
1	ALB	Albania	1979	71.181438
2	DZA	Algeria	1979	108.045384
3	AND	Andorra	1979	NaN
4	AGO	Angola	1979	NaN
...	...	...	...	...
7405	VEN	Venezuela, Bolivarian Republic of	2016	21.044404
7406	VNM	Viet Nam	2016	16.937983
7407	YEM	Yemen	2016	43.310794
7408	ZMB	Zambia	2016	45.789160
7409	ZWE	Zimbabwe	2016	41.392057

7410 rows × 4 columns

Questions

1. How has the leading cause of mortality in a country changed over time?

Results

In general, the leading cause of mortality has shifted from infectious and parasitic diseases to diseases of the circulatory system to neoplasms as time progresses.

Circulatory system illness remain by far the most common cause of death with neoplasms trailing behind. The other causes are far below these two.

Countries that are outliers when it comes to causes of death generally remain outliers for at least a decade. This shows that in order to be an outlier, there must be something habitual that these countries cannot fix quickly.

EDA

Lineplots

Let us find the proportion of deaths caused by each Cause globally. Note that the total number of deaths per year changes, so we need to divide the deaths per 100,000 per Cause by total deaths per 100,000.

In [26]:

```
deathsPerYearByCause = mdf.groupby("CauseName").mean()
display(deathsPerYearByCause)
display(deathsPerYearByCause.sum())
dpybcNormal = (deathsPerYearByCause / deathsPerYearByCause.sum()).drop("Cause", axis=1
dpybcNormal.T.plot(kind="line", figsize=(20, 10), colormap='gist_rainbow').legend(bbox_
dpybcNormal.T.plot(kind="line", figsize=(20, 10), colormap='gist_rainbow', ylim=(0, 0.1
    bbox_to_anchor=(1, 1)
)
```

CauseName	Cause	1980	1981	1982	1983	1984	1985	198(
Certain infectious and parasitic diseases	1	27.803922	26.175385	22.732258	21.658824	27.177358	22.111111	21.35000(
Congenital malformations, deformations and chromosomal abnormalities	16	7.135294	7.244615	7.243548	6.976471	6.639623	6.954167	6.68333
Diseases of the blood and blood-forming organs and certain	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN

The cause with the highest fraction of deaths by far is that of circulatory diseases. This has been decreasing since the 1990s but have recently risen.

The second is neoplasms, which is essentially cancer. This has been gradually increasing since the 1980s, but have recently spiked.

At around 1984, cardiovascular diseases suddenly took a dip while then number 2, 4, 5 (neoplasms, digestive diseases, infectious diseases) spiked.

The third is respiratory diseases. Respiratory diseases in general have been decreasing as the world becomes more developed.

Endocrine, nutritional and metabolic diseases have been on the rise quite rapidly since 1994.

Animated Choropleth Maps

Let us make an animated map of cardiovascular disease deaths across time. The data has to be interpolated based on last recorded number of deaths. This would prevent the countries' color from snapping in between an actual color and gray.

In [27]:

```
import plotly.express as px
from plotly.offline import init_notebook_mode

init_notebook_mode(connected=True)
mdf.loc[:, "1979":"2016"].apply(pd.to_numeric, errors="coerce")
mdf["1979"] = mdf["1979"].astype(float)
display(mdf)

# with pd.option_context('display.max_rows', None):
mdfInterp = pd.concat(
    [mdf.iloc[:, :4], mdf.loc[:, "1979":"2016"].interpolate(method="pad", axis=1)],
    axis=1,
)
display(mdfInterp)
```

	CountryCode	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	...	2
0	ALB	Albania	1	Certain infectious and parasitic diseases	NaN	NaN	NaN	NaN	NaN	NaN	...	
1	ALB	Albania	2	Neoplasms	NaN	NaN	NaN	NaN	NaN	NaN	...	
2	ALB	Albania	5	Mental and behavioural disorders	NaN	NaN	NaN	NaN	NaN	NaN	...	
3	ALB	Albania	6	Diseases of the nervous system	NaN	NaN	NaN	NaN	NaN	NaN	...	
4	ALB	Albania	7	Diseases of the eye and adnexa	NaN	NaN	NaN	NaN	NaN	NaN	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1776	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	NaN	3.0	NaN	NaN	NaN	NaN	...	
1777	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	NaN	3.2	NaN	NaN	NaN	NaN	...	
1778	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	NaN	13.9	NaN	NaN	NaN	NaN	...	
1779	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	NaN	14.7	NaN	NaN	NaN	NaN	...	
1780	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	NaN	8.7	NaN	NaN	NaN	NaN	...	

1781 rows × 42 columns

	CountryCode	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	...	2

	CountryCode	Country	Cause	CauseName	1979	1980	1981	1982	1983	1984	...	2
0	ALB	Albania	1	Certain infectious and parasitic diseases	NaN	NaN	NaN	NaN	NaN	NaN	...	
1	ALB	Albania	2	Neoplasms	NaN	NaN	NaN	NaN	NaN	NaN	...	
2	ALB	Albania	5	Mental and behavioural disorders	NaN	NaN	NaN	NaN	NaN	NaN	...	
3	ALB	Albania	6	Diseases of the nervous system	NaN	NaN	NaN	NaN	NaN	NaN	...	
4	ALB	Albania	7	Diseases of the eye and adnexa	NaN	NaN	NaN	NaN	NaN	NaN	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1776	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	NaN	3.0	3.0	3.0	3.0	3.0	...	
1777	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	NaN	3.2	3.2	3.2	3.2	3.2	...	
1778	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	NaN	13.9	13.9	13.9	13.9	13.9	...	
1779	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	NaN	14.7	14.7	14.7	14.7	14.7	...	
1780	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	NaN	8.7	8.7	8.7	8.7	8.7	...	

1781 rows × 42 columns

We will now convert the DataFrame to a "long" format from the current "wide" format by using the method `melt` . This prepares the data for the choropleth visualization.

In [28]:

```
forMap = pd.melt(
    mdfInterp, id_vars=["CountryCode", "Country", "Cause", "CauseName"], var_name="Yea
)
forMap.reset_index(inplace=True, drop=True)
forMap.rename(columns={"value": "DeathsPer100k"}, inplace=True)
forMap.DeathsPer100k = forMap.DeathsPer100k.astype(float)

display(forMap)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	1.0
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	1.6
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	8.3
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	0.0
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	0.8

67678 rows × 6 columns

We will now divide `DeathsPer100k` by the total number of deaths in that country, in that year. This is because different countries have different total rates of death, and this question aims to analyse the *proportion* of deaths due to a particular cause globally across the years.

In [29]:

```
totalDeathsSeries = forMap.groupby(["CountryCode", "Year"]).DeathsPer100k.sum()


def calculate_death_prop(row):
    if row.DeathsPer100k != np.nan:
        totalDeaths = totalDeathsSeries.loc[(row.CountryCode, row.Year)]
        row.DeathsPer100k = row.DeathsPer100k / totalDeaths
    return row


display(forMap)
forMapCalibed = forMap.apply(
    calculate_death_prop, axis=1
)  # re-calibrated DataFrame for map creation
forMapCalibed.rename(columns={"DeathsPer100k": "DeathsProp"}, inplace=True)
display(forMapCalibed)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	1.0
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	1.6
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	8.3
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	0.0
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	0.8

67678 rows × 6 columns

	CountryCode	Country	Cause	CauseName	Year	DeathsProp
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...

	CountryCode	Country	Cause	CauseName	Year	DeathsProp
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	0.003338
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	0.005340
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	0.027704
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	0.000000
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	0.002670

67678 rows × 6 columns

We will now normalize the data using the min-max method.

In [30]:

```
forMapCalibed.DeathsProp /= (
    forMapCalibed.DeathsProp.max() - forMapCalibed.DeathsProp.min()
)
forMapCalibed
```

Out[30]:

	CountryCode	Country	Cause	CauseName	Year	DeathsProp
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	0.004068
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	0.006509
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	0.033764
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	0.000000
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	0.003254

67678 rows × 6 columns

As one can see, the DeathsPer100k have been successfully converted to DeathsProp and the normalization has also done its job. For instance, the deaths figure for `CountryCode=VIR` and `Cause=14` has been converted from 8.3 originally to 0.027 after calibration to 0.033 after normalization. Now we can make our animated choropleth maps.

Note: Please see individual .html files for the results.

In [31]:

```
for cause in range(1, 17):
    forMapWithCause = forMapCalibed[forMapCalibed.Cause == cause]
    fig = px.choropleth(
        forMapWithCause,  # Input Dataframe
        locations="CountryCode",  # identify country code column
        color="DeathsProp",  # identify representing column
        hover_name="Country",  # identify hover name
        animation_frame="Year",  # identify date column
        projection="natural earth",  # select projection
        color_continuous_scale="Viridis",  # select preferred color scale
        range_color=[0, forMapWithCause.DeathsProp.max()],  # select range of dataset
        title=str(cause) + ". " + disease_codes.loc[cause, "name"],
    )
    fig.write_html("q1choropleth" + format(cause, "02d") + ".html", full_html=False)
```

I will go over some graphs that I find to be interesting.

Graph 9, diseases of the circulatory system: You can see a clear trend of the American countries gradually having circulatory system diseases become a smaller part of their causes of death. Most of Europe is exhibiting the same trend as well, with Eastern Bloc nations lagging behind the First World by about 30 years. Kazakhstan is the exception here, as it has reduced its circulatory disease deaths proportion significant in recent years. Egypt is roughly following the same trend as the Eastern Bloc.

Surprisingly, Mexico and Guatemala has since 1979 shown a very low proportion of deaths due to this cause. This breaks the trend of more developed countries having a lower proportion.

Graph 2, neoplasms: Compared to graph 9, the trend here is much less clear. In recent years, many developed countries have started to have a high proportion of this cause of death. In contrast to cardiovascular diseases, the countries' colors also tend to fluctuate a bit more between light and dark, indicating that the trend is much less obvious.

Graph 5, mental and behavioural disorders: First World countries have been experiencing a higher number of deaths due to this cause in the past 20 years, starting with Finland. I think this may be due to mental health becoming better documented in developed countries. Moreover, in many developing countries, parents would discreetly kill their child if they perceive something to be wrong with their minds early on, which makes this statistic a tad unreliable.

Graph 11, digestive system: Mexico and Guatemala has consistently shown a higher proportion of deaths due to this cause. Egypt and Eastern Bloc countries have also been especially vulnerable in recent years.

Graph 10, respiratory system: Guatemala is exceptionally high here. Most of South America as well as Kazakhstan have seen increases in this cause in the past few years.

Now we shall plot the leading cause of death globally over time. This will be similar to the choropleth plotted above but this time the data is discrete rather than continous. We have to first make a `DataFrame` of leading cause of death and number of deaths (for easy reference) per year by country.

Note from the future Steve:

Plotly is open-source so expect bugs. The following chunk of code creates placeholder datapoints. Without them, the only shown leading causes of deaths would the ones during the first year (1979). In the following years, if another leading cause of death appears, the choropleth map will simply not show it. This persists until

the original leading causes are no longer present in the current year.

This is clearly a bug on their part, so I will take the liberty of using some comparatively inefficient `for` loops to sort out the issue.

In [32]:

```
leadingCauses = list(
    pd.DataFrame(
        forMap.replace(np.nan, 0)
        .groupby(["Country", "Year"])
        .apply(
            lambda x: x.rename(
                {"Country": "ColumnCountry", "Year": "ColumnYear"}, axis=1
            ).loc[x.DeathsPer100k.idxmax()]
        )
    ).Cause.unique()
)
vat = {
    "CountryCode": [],
    "Country": [],
    "Cause": [],
    "CauseName": [],
    "Year": [],
    "DeathsPer100k": [],
}
for x in range(1979, 2017):
    for y in leadingCauses:
        vat["CountryCode"].append("ph" + str(y))
        vat["Country"].append("ph" + str(y))
        vat["Cause"].append(y)
        vat["CauseName"].append(disease_codes.loc[y, "name"])
        vat["Year"].append(str(x))
        vat["DeathsPer100k"].append(0)
vatdf = pd.DataFrame(vat)
forDMap = pd.concat([forMap, vatdf], ignore_index=True)
```

In [33]:

```
display(forDMap)
maxCauseOfDeath = pd.DataFrame(
    forDMap.replace(np.nan, 0)
    .groupby(["Country", "Year"])
    .apply(
        lambda x: x.rename(
            {"Country": "ColumnCountry", "Year": "ColumnYear"}, axis=1
        ).loc[x.DeathsPer100k.idxmax()]
    )
)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...
67863	ph1	ph1	1	Certain infectious and parasitic diseases	2016	0.0
67864	ph9	ph9	9	Diseases of the circulatory system	2016	0.0
67865	ph2	ph2	2	Neoplasms	2016	0.0
67866	ph10	ph10	10	Diseases of the respiratory system	2016	0.0
67867	ph7	ph7	7	Diseases of the eye and adnexa	2016	0.0

67868 rows × 6 columns

Now, we can plot a discrete choropleth map.

In [34]:

```
maxCauseOfDeath.ColumnYear = maxCauseOfDeath.ColumnYear.astype(int)
maxCauseOfDeath.Cause = maxCauseOfDeath.Cause.astype(str)
fig = px.choropleth(
    maxCauseOfDeath,
    color="CauseName",
    locations="CountryCode",
    hover_data=["ColumnCountry", "DeathsPer100k"],
    animation_frame="ColumnYear",
)
fig.update_layout(legend=dict(yanchor="top", y=0.99, xanchor="left", x=0.9))
fig.write_html("q1discretechoro.html")
```

With the exception of Guatemala, which has consistently had respiratory illnesses as it's primary cause of death, the rest of the nations experience the causes of death roughly in this order as they become more developed: infectious, circulatory and then neoplasms.

I believe the reasons are as follows. Infectious diseases used to be commonplace as the healthcare infrastructure of many Second and Third World countries were unsatisfactory. Once this problem has been resolved, the two remaining are both degenerative diseases. Circulatory diseases are partially due to lifestyle decisions such as diet and exercise, and so as the citizens of very developed countries educate themselves in these regards, proportion of deaths due to circulatory diseases have decreased. Lastly, there is still no cure for cancer, so neoplasms is the final major cause of death that even exceptionally developed countries are struggling with today.

Boxplots

Let us also plot some boxplots to identify the outlier countries of each cause of death each year.

In [35]:

```
topCauses = (
    mdf.groupby("Cause").sum().sum(axis=1).sort_values(ascending=False).index[:5]
)
forBoxplot = longmdf.copy()
forBoxplot.reset_index(inplace=True, drop=True)
forBoxplot.DeathsPer100k = forBoxplot.DeathsPer100k.astype(float)

# This is similar to the calibrated data used to create choropleth maps except there is
totalDeathsSeries = forBoxplot.groupby(["CountryCode", "Year"]).DeathsPer100k.sum()
display(forBoxplot)
forBoxplotCalibed = forBoxplot.apply(
    calculate_death_prop, axis=1
)  # re-calibrated DataFrame for Boxplot creation
forBoxplotCalibed.rename(columns={"DeathsPer100k": "DeathsProp"}, inplace=True)
forBoxplotCalibed.DeathsProp /= (
    forBoxplotCalibed.DeathsProp.max() - forBoxplotCalibed.DeathsProp.min()
)
display(forBoxplotCalibed)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN
...	...	...	...	...	...	...
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	NaN
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	NaN
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	NaN
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	NaN
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	NaN

67678 rows × 6 columns

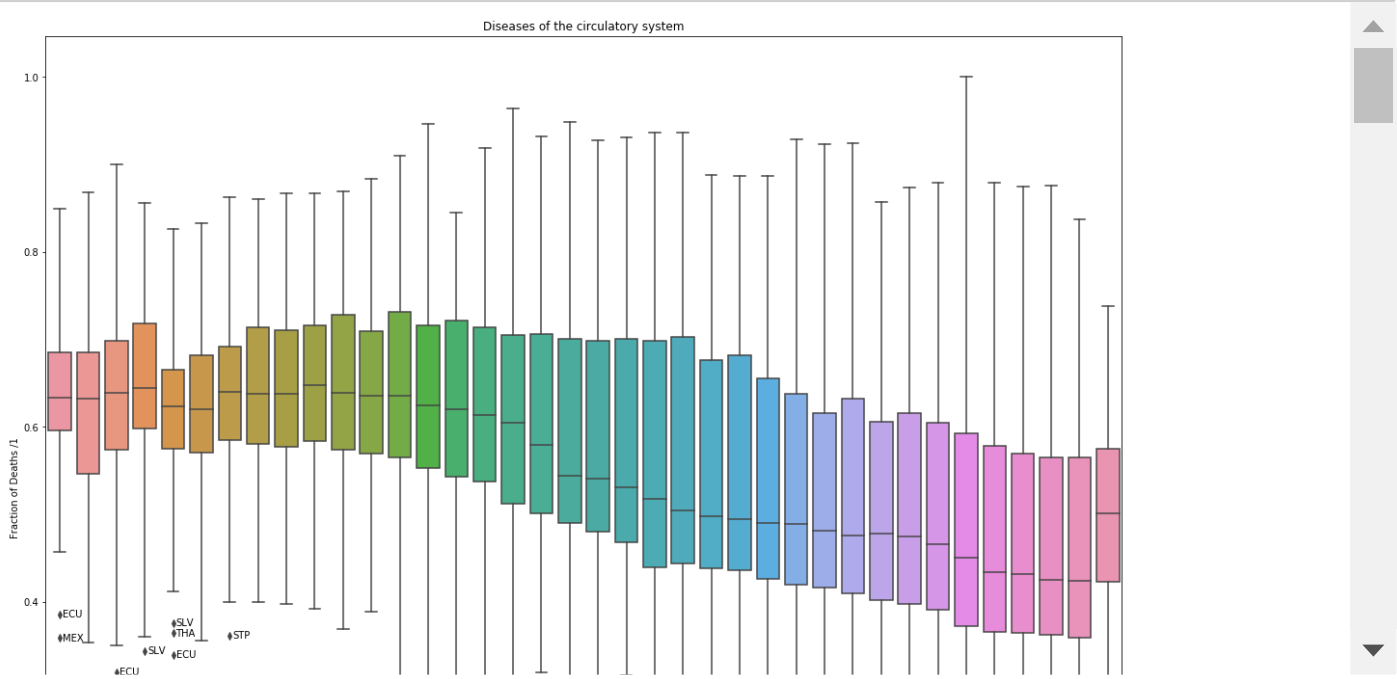
	CountryCode	Country	Cause	CauseName	Year	DeathsProp
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN
1	ALB	Albania	2	Neoplasms	1979	NaN
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN

	CountryCode	Country	Cause	CauseName	Year	DeathsProp
...	...	...	...	...	...	...
67673	VIR	Virgin Islands, U.S.	12	Diseases of the skin and subcutaneous tissue	2016	NaN
67674	VIR	Virgin Islands, U.S.	13	Diseases of the musculoskeletal system and con...	2016	NaN
67675	VIR	Virgin Islands, U.S.	14	Diseases of the genitourinary system	2016	NaN
67676	VIR	Virgin Islands, U.S.	15	Pregnancy, childbirth and the puerperium	2016	NaN
67677	VIR	Virgin Islands, U.S.	16	Congenital malformations, deformations and chr...	2016	NaN

Now, onto plotting the boxplots. I will be using a whisker size of 2 IQR instead of the conventional 1.5 IQR because this dataset has high kurtosis, and too many outliers makes the plot very cluttered. I will also be labelling the outliers so as to get a clear indication of which nations have been outlying throughout the years.

In [36]:

```
fig, axs = plt.subplots(len(topCauses), figsize=(20, 100))
forBoxplotCondensed = forBoxplotCalibed[forBoxplotCalibed.Cause.isin(topCauses)]
for x in range(len(topCauses)):
    # Label outliers:
    # dptc stands for DeathsPerTopCause
    dptc = forBoxplotCondensed[forBoxplotCondensed.Cause == topCauses[x]]
    q1 = dptc.groupby(dptc.Year).quantile(0.25)["DeathsProp"].to_numpy()
    q3 = dptc.groupby(dptc.Year).quantile(0.75)["DeathsProp"].to_numpy()
    outlier_top_lim = q3 + 2 * (q3 - q1)
    outlier_bottom_lim = q1 - 2 * (q3 - q1)
    #     print(outlier_top_lim)
    for row in dptc.itertuples():
        year = int(row.Year) - 1979
        #         print(type(year), year)
        val = row.DeathsProp
        if val > outlier_top_lim[year] or val < outlier_bottom_lim[year]:
            axs[x].text(year + 0.1, val, row.CountryCode, ha="left", va="center")

    sns.boxplot(data=dptc, y="DeathsProp", x="Year", ax=axs[x], whis=2)
    axs[x].set_title(disease_codes.loc[topCauses[x], "name"])
    axs[x].set_ylabel("Fraction of Deaths /1")
```



Circulatory: Guatemala had notably lower proportion of deaths due to this cause prior to 1991.

Neoplasms: Quite a large IQR with few outliers, indicating that no matter how developed a country is, cancer will still take away roughly the same proportion of lives. However, it has been increasing in recent years, as evidenced by the choropleths as well.

Respiratory: Before 2006, Guatemala had exceptionally high proportion of deaths due to this cause. After 2006, Singapore took its position and has maintained this outlier status for the past 10 years. I think this may be due to the high number of smokers after Singapore's GDP per capita rose and people could afford cigarettes.

Digestive: Egypt, North Macedonia and Mexico have all had high proportions of deaths due to this cause, with the latter 2's outlier status persevering from the 1980s until now. Kiribati also experienced a spike in this statistic in the 1990s. Perhaps it is due to the ulcer-inducing spicy food found in the borders of these countries.

Infectious and Parasitic: Guatemala and Kiribati have had notably higher proportions of death due to this cause from the 1980s to the 1990s. In recent years, Thailand and South Africa have dominated the charts, both being significantly outside of the range of 2 IQR. These countries have quite a bit of inequality (e.g. South Africa had apartheid) when it comes to urbanisation, so that may be why.

2. How does the average BMI correlate with the causes of mortality at any point in time?

Result

As BMI increases, the death rate for most causes of death decreases. This is probably because BMI only increases when a country becomes more developed and people can eat more nutritious meals.

The only exceptions are nervous system and behavioural disorders, which are positively correlated with BMI. I will give further explanation below.

On a country level, Russia appears to be an exception as it's number of deaths due to digestive system illnesses is also positively correlated with BMI.

EDA

Correlation and Regplots

Let us find out which causes of death are most correlated with BMI through correlation values and regplots.

In [37]:

```
bmiMort = pd.merge(
    longmdf, longbmidf, on=["CountryCode", "Country", "Year"], how="inner"
)
_bmiMort = pd.merge(longmdf, longbmidf, on=["CountryCode", "Year"], how="inner")
bmiMort.DeathsPer100k = bmiMort.DeathsPer100k.astype(float)
display(bmiMort)
display(_bmiMort)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN	24.0
1	ALB	Albania	2	Neoplasms	1979	NaN	24.0
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN	24.0
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN	24.0
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN	24.0
...	...	...	...	...	...	...	...
60985	VEN	Venezuela, Bolivarian Republic of	12	Diseases of the skin and subcutaneous tissue	2016	NaN	26.7
60986	VEN	Venezuela, Bolivarian Republic of	13	Diseases of the musculoskeletal system and con...	2016	NaN	26.7
60987	VEN	Venezuela, Bolivarian Republic of	14	Diseases of the genitourinary system	2016	NaN	26.7
60988	VEN	Venezuela, Bolivarian Republic of	15	Pregnancy, childbirth and the puerperium	2016	NaN	26.7
60989	VEN	Venezuela, Bolivarian Republic of	16	Congenital malformations, deformations and chr...	2016	NaN	26.7

60990 rows × 7 columns

	CountryCode	Country_x	Cause	CauseName	Year	DeathsPer100k	Country_y	E
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN	Albania	2
1	ALB	Albania	2	Neoplasms	1979	NaN	Albania	2
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN	Albania	2
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN	Albania	2
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN	Albania	2

	CountryCode	Country_x	Cause	CauseName	Year	DeathsPer100k	Country_y	E
...	...	...	...	...	...	...	...	
60985	VEN	Venezuela, Bolivarian Republic of	12	Diseases of the skin and subcutaneous tissue	2016	NaN	Venezuela, Bolivarian Republic of	2
60986	VEN	Venezuela, Bolivarian Republic of	13	Diseases of the musculoskeletal system and con...	2016	NaN	Venezuela, Bolivarian Republic of	2
60987	VEN	Venezuela, Bolivarian Republic of	14	Diseases of the genitourinary system	2016	NaN	Venezuela, Bolivarian Republic of	2
60988	VEN	Venezuela, Bolivarian Republic of	15	Pregnancy, childbirth and the puerperium	2016	NaN	Venezuela, Bolivarian Republic of	2
60989	VEN	Venezuela, Bolivarian Republic of	16	Congenital malformations, deformations and chr...	2016	NaN	Venezuela, Bolivarian Republic of	2

60990 rows × 8 columns

Notice how the number of rows is the same whether `Country` is include in the `on` parameter. This shows that the mapping between `Country` and `CountryCode` is 1-to-1.

We shall now proceed to plot `regplot` s of every countries data throughout the years. We will be removing the outliers (defined arbitrarily based on quantile). If you decrease the quantile range, you would see that the number of data points remaining is decreased, which shows that the code is correct.

In [38]:

```
from scipy import stats
import math


def find_correl(df, a, b):
    dfnona = df.dropna()
    return stats.pearsonr(dfnona[a], dfnona[b])


# append correlation between two variables grouped by Cause
def append_correl(df, a1, b1):
    correls = df.groupby("Cause").apply(find_correl, a=a1, b=b1)
    diseases = df.Cause.unique()
    correldf = pd.DataFrame(
        [[a, correls[a][0], correls[a][1]] for a in diseases],
        columns=["Cause", "Correl", "pValue"],
    )
    return pd.merge(df, correldf, on="Cause")


def remove_outliers(df):
    outlier_boundaries = (
        df.groupby("Cause").DeathsPer100k.quantile([0.05, 0.95]).unstack(level=1)
    )
    nooutlier = df[
        (
            df.DeathsPer100k
            >= outlier_boundaries.iloc[df.Cause - 1, 0].reset_index(drop=True)
        )
        & (
            df.DeathsPer100k
            <= outlier_boundaries.iloc[df.Cause - 1, 1].reset_index(drop=True)
        )
    ]
    return nooutlier


# remove outliers + append_correl
display(bmiMort)
bmiMort_nooutlier = remove_outliers(bmiMort)
display(bmiMort_nooutlier)
bmiMort2 = append_correl(bmiMort_nooutlier.dropna(), "BMI", "DeathsPer100k")
display(bmiMort2)


def plot4x4reg(df, indepvar):
    # To make it work for other number of causes as well
    count = len(disease_codes)
    maxwidth = math.ceil(count**0.5)
    fig, axes = plt.subplots(maxwidth, maxwidth, figsize=(30, 20))
    diseases_sorted = df.sort_values(by="Correl").CauseName.unique()
    for ax, cause in zip(axes.ravel(), diseases_sorted):
        dfs = df.query("CauseName==@cause")
        sns.regplot(
            data=dfs,
            x=indepvar,
            y="DeathsPer100k",
            scatter_kws={"s": 3},
            x_jitter=0.05,
```

```
            ax=ax,
        ).set_title(
            cause + " (" + str(round(dfs.Correl.iloc[0], 2)) + ", p="
            # round p value to 3 d.p as anything less than 0.001 is statistically sign
            + str(round(dfs.pValue.iloc[0], 4)) + ")",
            fontsize=8,
        )


plot4x4reg(
    append_correl(bmiMort.dropna(), "BMI", "DeathsPer100k"), "BMI"
)  # with outliers
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN	24.0
1	ALB	Albania	2	Neoplasms	1979	NaN	24.0
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN	24.0
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN	24.0
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN	24.0
...	...	...	...	...	...	...	...
60985	VEN	Venezuela, Bolivarian Republic of	12	Diseases of the skin and subcutaneous tissue	2016	NaN	26.7
60986	VEN	Venezuela, Bolivarian Republic of	13	Diseases of the musculoskeletal system and con...	2016	NaN	26.7
60987	VEN	Venezuela, Bolivarian Republic of	14	Diseases of the genitourinary system	2016	NaN	26.7
60988	VEN	Venezuela, Bolivarian Republic of	15	Pregnancy, childbirth and the puerperium	2016	NaN	26.7
60989	VEN	Venezuela, Bolivarian Republic of	16	Congenital malformations, deformations and chr...	2016	NaN	26.7

60990 rows × 7 columns

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI
30	ARG	Argentina	1	Certain infectious and parasitic diseases	1979	29.8	24.4
31	ARG	Argentina	2	Neoplasms	1979	155.2	24.4
34	ARG	Argentina	5	Mental and behavioural disorders	1979	3.6	24.4
35	ARG	Argentina	6	Diseases of the nervous system	1979	9.9	24.4

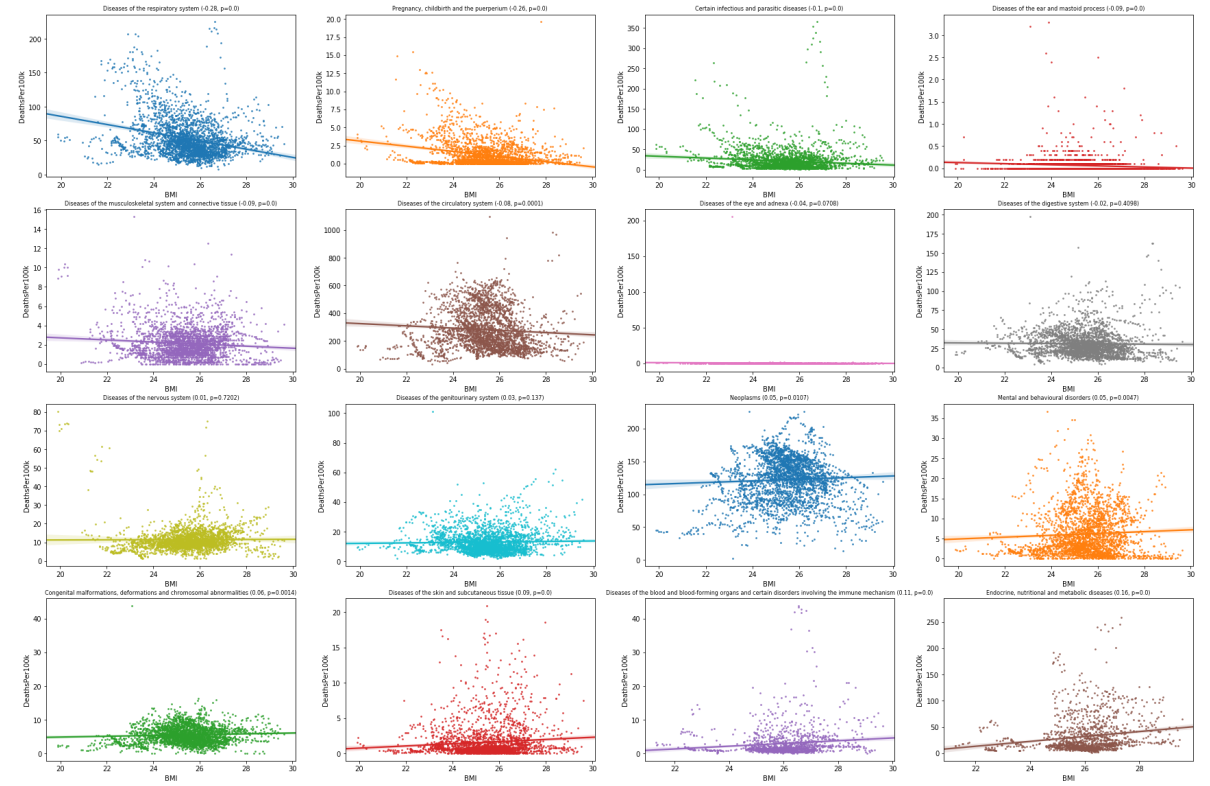
	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI
36	ARG	Argentina	7	Diseases of the eye and adnexa	1979	0.0	24.4
...	...	...	...	...	...	...	...
60889	USA	United States	12	Diseases of the skin and subcutaneous tissue	2016	0.8	28.9
60890	USA	United States	13	Diseases of the musculoskeletal system and con...	2016	2.4	28.9
60891	USA	United States	14	Diseases of the genitourinary system	2016	11.1	28.9
60892	USA	United States	15	Pregnancy, childbirth and the puerperium	2016	0.8	28.9
60893	USA	United States	16	Congenital malformations, deformations and chr...	2016	3.7	28.9

37897 rows × 7 columns

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	
0	ARG	Argentina	1	Certain infectious and parasitic diseases	1979	29.8	24.4	-0.127130	1.40
1	AUS	Australia	1	Certain infectious and parasitic diseases	1979	3.4	24.2	-0.127130	1.40
2	BRB	Barbados	1	Certain infectious and parasitic diseases	1979	17.8	25.3	-0.127130	1.40
3	BEL	Belgium	1	Certain infectious and parasitic diseases	1979	5.4	24.9	-0.127130	1.40
4	CUB	Cuba	1	Certain infectious and parasitic diseases	1979	11.9	23.0	-0.127130	1.40
...	...	...	...	...	...	...	...	...	
37892	MDA	Moldova, Republic of	4	Endocrine, nutritional and metabolic diseases	2016	8.7	27.0	0.217845	1.30
37893	ROU	Romania	4	Endocrine, nutritional and metabolic diseases	2016	7.3	26.9	0.217845	1.30
37894	SWE	Sweden	4	Endocrine, nutritional and metabolic diseases	2016	10.5	26.0	0.217845	1.30

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	
37895	THA	Thailand	4	Endocrine, nutritional and metabolic diseases	2016	18.9	24.4	0.217845	1.30
37896	USA	United States	4	Endocrine, nutritional and metabolic diseases	2016	21.5	28.9	0.217845	1.30

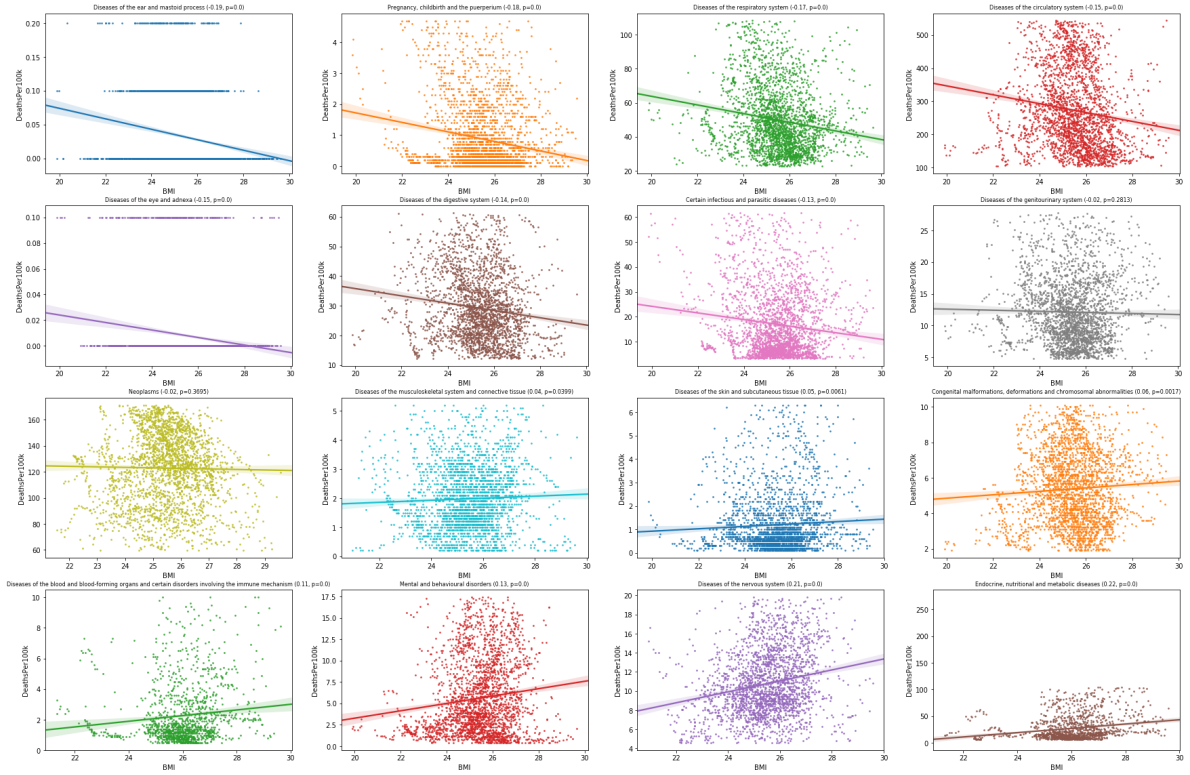
37897 rows × 9 columns



Now, without outliers the regplots are much tighter along the y-axis.

In [39]:

```
1 plot4x4reg(bmiMort2, "BMI")
```



As one can see, there is a huge amount of variability in the data (barring the `jitter` set in place to compensate for decimal point issues) so there is no strong correlation between death originating from major causes and BMI. However, what if we take the average of `DeathsPer100k` for every country per year. This would theoretically fix the massive amount of variation between the developed and developing countries at any point in time.

In [40]:

```
def get_mean_deaths(df, indepvar):
    meanDeaths = df.groupby(["Year", "Cause", "CauseName"]).agg(
        {"DeathsPer100k": ["mean"], indepvar: ["mean"]}
    )
    meanDeaths.reset_index(inplace=True)
    meanDeaths.columns = meanDeaths.columns.droplevel(1)
    return meanDeaths


def reg_mean_deaths(df, indepvar):
    meanDeaths = get_mean_deaths(df, indepvar)
    meanDeaths = append_correl(meanDeaths, indepvar, "DeathsPer100k")
    display(meanDeaths)

    sns.color_palette("bright")
    sns.lmplot(
        data=meanDeaths,
        x=indepvar,
        y="DeathsPer100k",
        hue="CauseName",
        scatter_kws={"s": 5},
        height=8,
        aspect=1.5,
        legend=False,
    )
    legend = (
        meanDeaths.CauseName
        + " ("
        + meanDeaths.Correl.round(2).astype(str)
        + ", p="
        + meanDeaths.pValue.round(4).astype(str)
        + ")"
    ).unique()
    plt.legend(title="Cause", bbox_to_anchor=(1.04, 1), loc="upper left", labels=legen

    plot4x4reg(meanDeaths, indepvar)


display(bmiMort2)
reg_mean_deaths(bmiMort2, "BMI")
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl
0	ARG	Argentina	1	Certain infectious and parasitic diseases	1979	29.8	24.4	-0.127130
1	AUS	Australia	1	Certain infectious and parasitic diseases	1979	3.4	24.2	-0.127130
2	BRB	Barbados	1	Certain infectious and parasitic diseases	1979	17.8	25.3	-0.127130

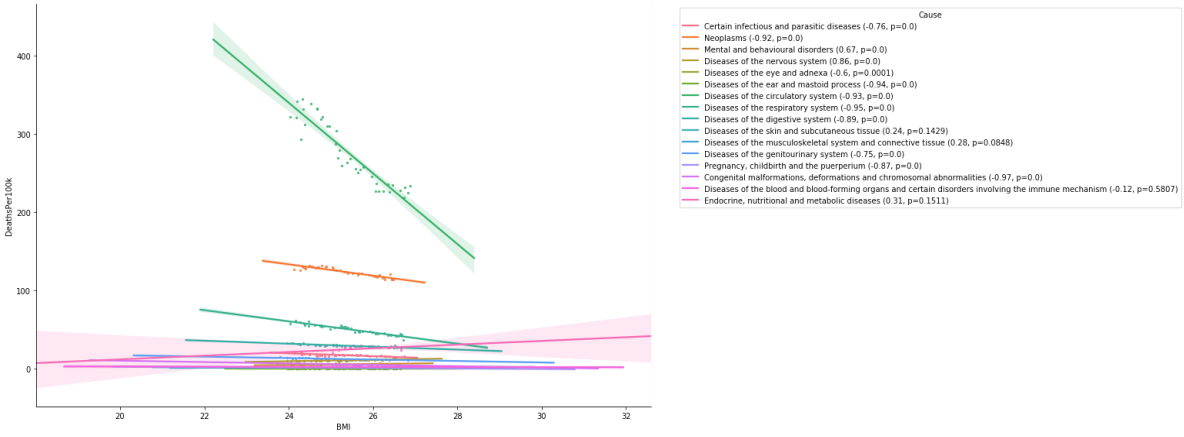
	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl
3	BEL	Belgium	1	Certain infectious and parasitic diseases	1979	5.4	24.9	-0.127130
4	CUB	Cuba	1	Certain infectious and parasitic diseases	1979	11.9	23.0	-0.127130
...	...	...	...	...	...	...	...	...
37892	MDA	Moldova, Republic of	4	Endocrine, nutritional and metabolic diseases	2016	8.7	27.0	0.217845
37893	ROU	Romania	4	Endocrine, nutritional and metabolic diseases	2016	7.3	26.9	0.217845
37894	SWE	Sweden	4	Endocrine, nutritional and metabolic diseases	2016	10.5	26.0	0.217845
37895	THA	Thailand	4	Endocrine, nutritional and metabolic diseases	2016	18.9	24.4	0.217845
37896	USA	United States	4	Endocrine, nutritional and metabolic diseases	2016	21.5	28.9	0.217845

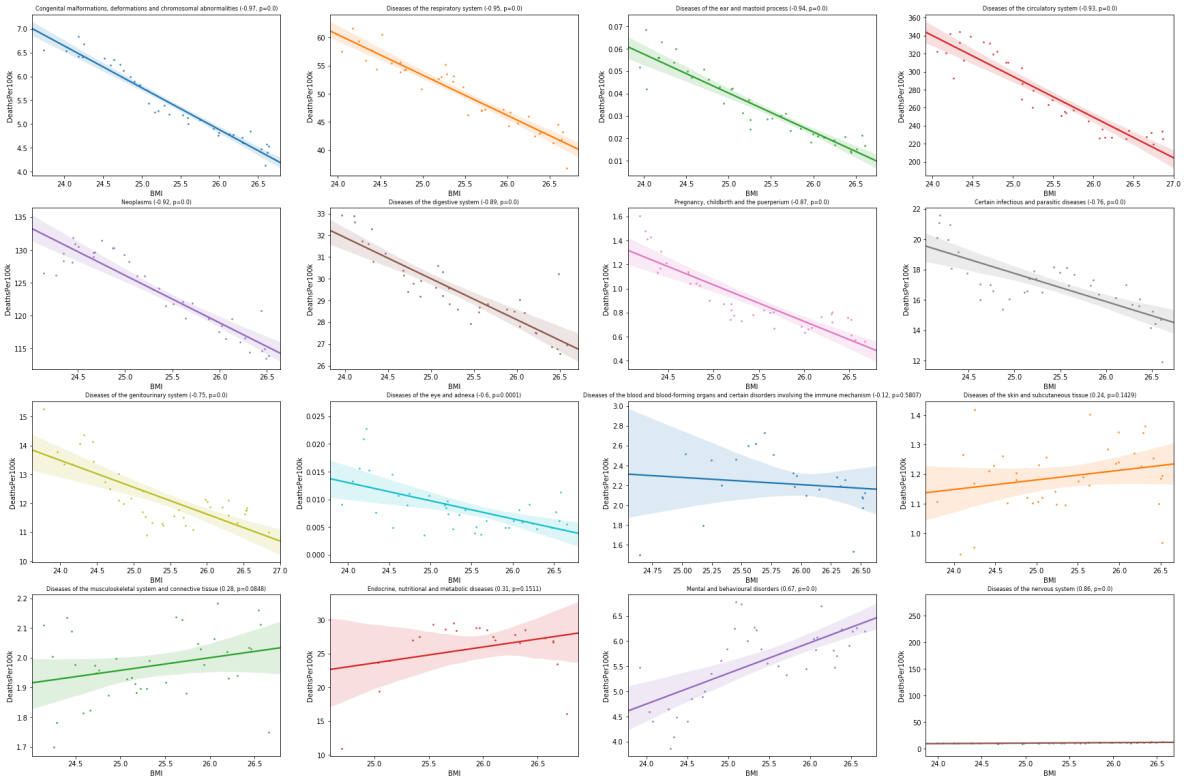
37897 rows × 9 columns

	Year	Cause	CauseName	DeathsPer100k	BMI	Correl	pValue
0	1979	1	Certain infectious and parasitic diseases	20.114815	24.118519	-0.763811	2.425297e-08
1	1980	1	Certain infectious and parasitic diseases	21.067442	24.151163	-0.763811	2.425297e-08
2	1981	1	Certain infectious and parasitic diseases	21.580000	24.167273	-0.763811	2.425297e-08
3	1982	1	Certain infectious and parasitic diseases	20.972222	24.283333	-0.763811	2.425297e-08
4	1983	1	Certain infectious and parasitic diseases	18.088372	24.306977	-0.763811	2.425297e-08
...	...	...	...	...	...	...	...
573	2012	4	Endocrine, nutritional and metabolic diseases	27.324324	26.578378	0.309181	1.511319e-01
574	2013	4	Endocrine, nutritional and metabolic diseases	26.913889	26.665278	0.309181	1.511319e-01

	Year	Cause	CauseName	DeathsPer100k	BMI	Correl	pValue
575	2014	4	Endocrine, nutritional and metabolic diseases	26.654286	26.652857	0.309181	1.511319e-01
576	2015	4	Endocrine, nutritional and metabolic diseases	23.430189	26.664151	0.309181	1.511319e-01
577	2016	4	Endocrine, nutritional and metabolic diseases	16.120000	26.726667	0.309181	1.511319e-01

578 rows × 7 columns





As one can see, the gradient of the regression lines (i.e. whether they are increasing or decreasing) are in line with the correlation values in the legend. This shows that this plot is correct.

Analysis: One would expect that as BMI increases, deaths due to cardiovascular reasons would increase. However this is not the case. There is a very strong *negative* correlation (c=-0.93) between BMI and circulatory system diseases. The same is observed for many other causes of death. I think this is because a country with high BMI is a country that can cater to the nutritional needs of its citizens very well, so it is developed and healthcare standards should be high as well, and deaths due to various illnesses decrease. The only cause that shows a strong positive correlation with BMI is mental and behavioural disorders. Perhaps this is due to the same reason as the choropleth maps above. I.e., in developed countries, people with mental illnesses actually survive long enough to be considered a valid data point.

Let us now try to narrow it down to any particular country. We will keep the outliers in for this one as while the deaths figures may be outliers on the global scale, on a country-by-country basis they could still be statistically significant. NOTE: I will only be drawing conclusions from "important" countries such as the United States, but this model can be used for any country with data.

In [41]:

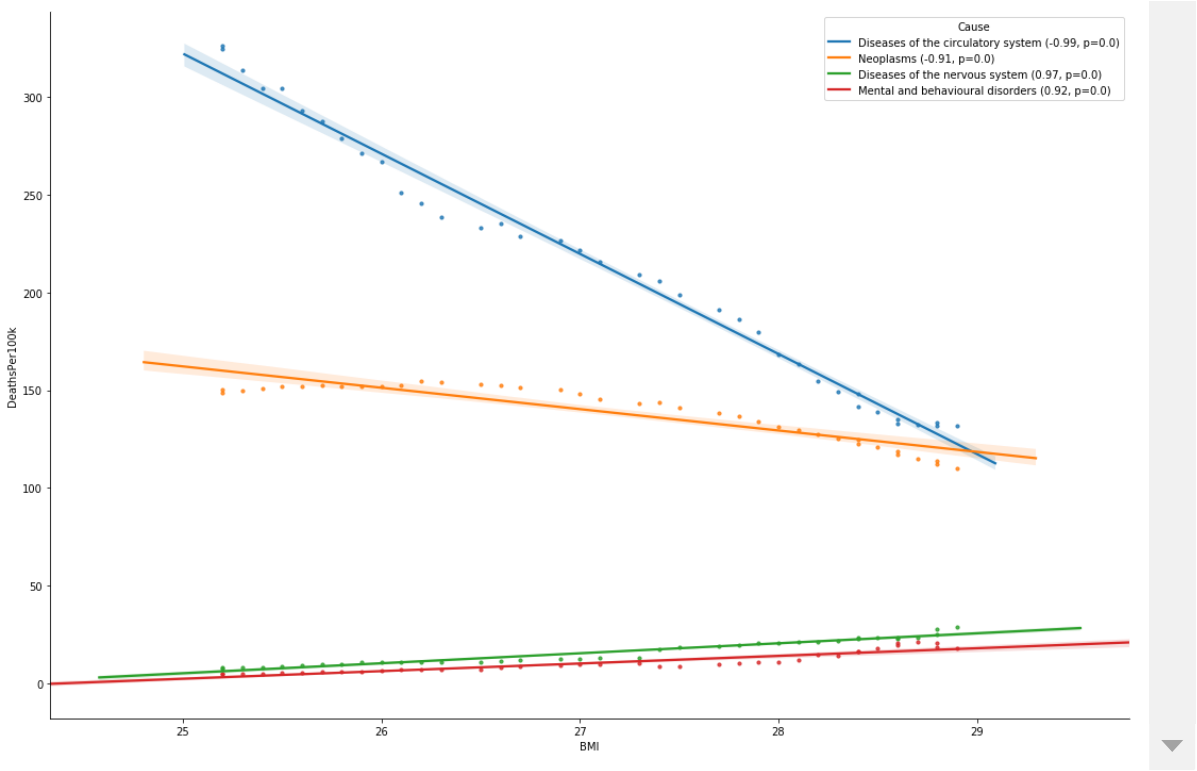
```
def reg_per_country(df, indepvar):
    inp = ""
    while inp.lower() != "stop":
        inp = input("Enter alpha-3 country code: ").upper()
        if inp in df.CountryCode.values:
            country = df[df.CountryCode == inp]
            print(country.Country.iloc[0])

            # Plot only causes that have a variety of number of deaths.
            gbCause = country.groupby("Cause")
            largeDiff = gbCause.DeathsPer100k.max() - gbCause.DeathsPer100k.min() > 10
            country = country[largeDiff[country.Cause].values]
            country.drop(
                ["Correl", "pValue"], axis=1, inplace=True
            )  # recalculate correlation

            country = append_correl(country, indepvar, "DeathsPer100k")
            toPlot = country[
                # Ensure correlation is strong and valid
                (abs(country.Correl) >= 0.5)
                & (country.pValue <= 0.05)
            ]

            # Sort by sum of deaths to make the line colors a bit more in sync for eas
            longmdf.DeathsPer100k = longmdf.DeathsPer100k.astype(float)
            sort_by_this = (
                longmdf.groupby("Cause").DeathsPer100k.sum().sort_values().index
            )
            toPlot.Cause = toPlot.Cause.astype("category")
            toPlot.Cause.cat.set_categories(sort_by_this, inplace=True)
            toPlot.sort_values(["Cause"], inplace=True, ascending=False)
            display(toPlot)

            plot = sns.lmplot(
                data=toPlot,
                x=indepvar,
                y="DeathsPer100k",
                hue="CauseName",
                scatter_kws={"s": 10},
                height=10,
                aspect=1.5,
                legend=False,
            )
            legend = (
                toPlot.CauseName
                + " ("
                + toPlot.Correl.round(2).astype(str)
                + ", p="
                + toPlot.pValue.round(4).astype(str)
                + ")"
            ).unique()
            plt.legend(title="Cause", loc="upper right", labels=legend)
            plt.show(plot)


reg_per_country(append_correl(bmiMort, "BMI", "DeathsPer100k").dropna(), "BMI")
```

```
Enter alpha-3 country code:  usa
```

United States

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	pVa
189	USA	United States	9	Diseases of the circulatory system	2016	131.8	28.9	-0.993215	3.04120
161	USA	United States	9	Diseases of the circulatory system	1988	266.8	26.0	-0.993215	3.04120
168	USA	United States	9	Diseases of the circulatory system	1995	226.7	26.9	-0.993215	3.04120
167	USA	United States	9	Diseases of the circulatory system	1994	228.8	26.7	-0.993215	3.04120
166	USA	United States	9	Diseases of the circulatory system	1993	235.2	26.6	-0.993215	3.04120
...	...	...	...	...	...	...	...	...	
103	USA	United States	5	Mental and behavioural disorders	2006	14.7	28.2	0.921908	2.12903
102	USA	United States	5	Mental and behavioural disorders	2005	12.1	28.1	0.921908	2.12903
101	USA	United States	5	Mental and behavioural disorders	2004	10.9	28.0	0.921908	2.12903
100	USA	United States	5	Mental and behavioural disorders	2003	10.8	27.9	0.921908	2.12903
78	USA	United States	5	Mental and behavioural disorders	1981	5.2	25.3	0.921908	2.12903

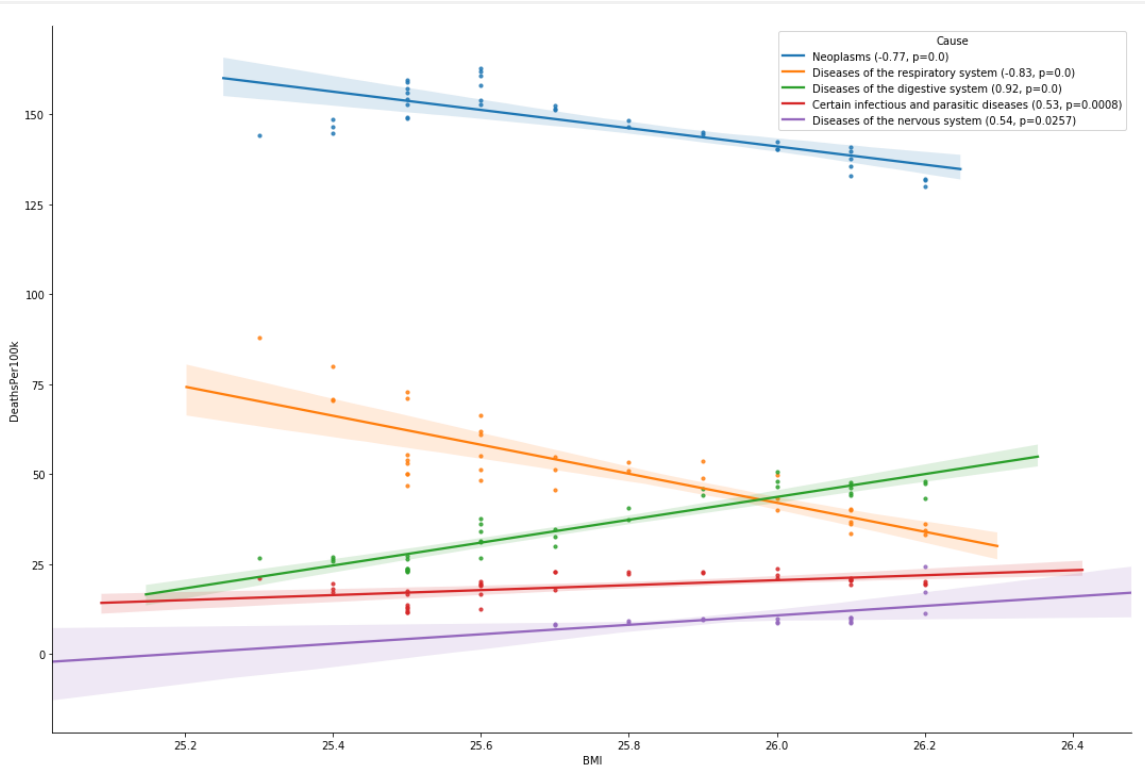
152 rows × 9 columns




```
Enter alpha-3 country code:  rus

Russian Federation
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	pV
54	RUS	Russian Federation	2	Neoplasms	1998	151.7	25.7	-0.768837	4.3299
44	RUS	Russian Federation	2	Neoplasms	1988	156.0	25.5	-0.768837	4.3299
50	RUS	Russian Federation	2	Neoplasms	1994	161.8	25.6	-0.768837	4.3299
49	RUS	Russian Federation	2	Neoplasms	1993	162.9	25.6	-0.768837	4.3299
48	RUS	Russian Federation	2	Neoplasms	1992	160.9	25.6	-0.768837	4.3299
...	...	...	...	...	...	...	...	...	
83	RUS	Russian Federation	6	Diseases of the nervous system	2010	10.2	26.1	0.538560	2.5717
84	RUS	Russian Federation	6	Diseases of the nervous system	2011	10.1	26.1	0.538560	2.5717
85	RUS	Russian Federation	6	Diseases of the nervous system	2012	9.7	26.1	0.538560	2.5717
86	RUS	Russian Federation	6	Diseases of the nervous system	2013	11.3	26.2	0.538560	2.5717
80	RUS	Russian Federation	6	Diseases of the nervous system	2007	8.6	26.0	0.538560	2.5717

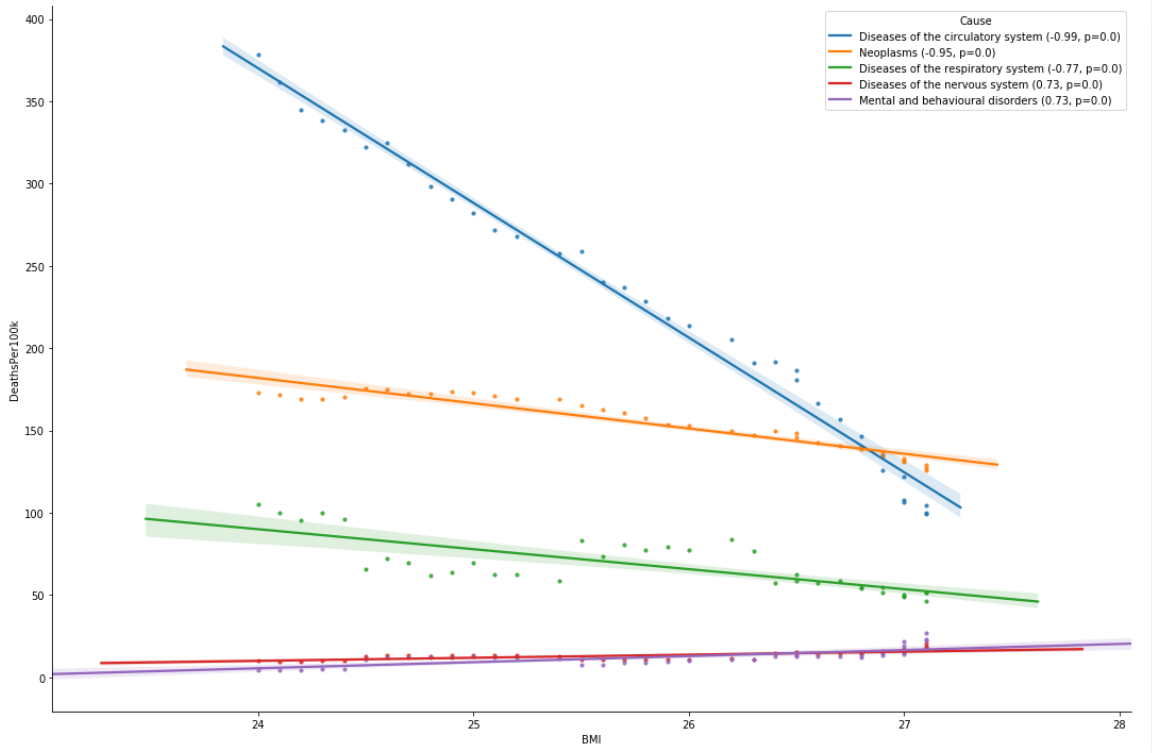
161 rows × 9 columns




```
Enter alpha-3 country code:  gbr

United Kingdom
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	pVa
112	GBR	United Kingdom	9	Diseases of the circulatory system	1980	361.3	24.1	-0.992869	6.3169
138	GBR	United Kingdom	9	Diseases of the circulatory system	2006	146.5	26.8	-0.992869	6.3169
131	GBR	United Kingdom	9	Diseases of the circulatory system	1999	205.2	26.2	-0.992869	6.3169
132	GBR	United Kingdom	9	Diseases of the circulatory system	2000	191.3	26.3	-0.992869	6.3169

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	pVa
133	GBR	United Kingdom	9	Diseases of the circulatory system	2001	191.5	26.4	-0.992869	6.31693
...	...	...	...	...	...	...	...	...	
60	GBR	United Kingdom	5	Mental and behavioural disorders	2002	13.0	26.5	0.730282	2.89415
59	GBR	United Kingdom	5	Mental and behavioural disorders	2001	13.0	26.4	0.730282	2.89415
58	GBR	United Kingdom	5	Mental and behavioural disorders	2000	10.6	26.3	0.730282	2.89415
57	GBR	United Kingdom	5	Mental and behavioural disorders	1999	10.7	26.2	0.730282	2.89415
72	GBR	United Kingdom	5	Mental and behavioural	2014	23.4	27.1	0.730282	2.89415




```
Enter alpha-3 country code:  bra

Brazil
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	
110	BRA	Brazil	9	Diseases of the circulatory system	2015	167.5	26.5	-0.900032	3.5
82	BRA	Brazil	9	Diseases of the circulatory system	1987	237.9	23.7	-0.900032	3.5

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	Correl	
90	BRA	Brazil	9	Diseases of the circulatory system	1995	285.7	24.4	-0.900032	3.5
89	BRA	Brazil	9	Diseases of the circulatory system	1994	292.0	24.3	-0.900032	3.5
88	BRA	Brazil	9	Diseases of the circulatory system	1993	299.4	24.2	-0.900032	3.5
...	...	...	...	...	...	...	...	...	
24	BRA	Brazil	1	Certain infectious and parasitic diseases	2003	31.5	25.1	-0.862668	6.68
23	BRA	Brazil	1	Certain infectious and parasitic diseases	2002	31.1	25.0	-0.862668	6.68
22	BRA	Brazil	1	Certain infectious and parasitic diseases	2001	31.7	24.9	-0.862668	6.68
21	BRA	Brazil	1	Certain infectious and parasitic diseases	2000	32.0	24.8	-0.862668	6.68
0	BRA	Brazil	1	Certain infectious and parasitic diseases	1979	62.6	23.0	-0.862668	6.68

111 rows × 9 columns


```
Enter alpha-3 country code:  stop
```

United States: Nervous system illnesses and mental disorders are related, and they are both positively correlated with BMI. The only explanation I can think of is the one previously mentioned. U.S.'s data is in accordance with the rest of the world.

Russia: The main thing to note here is the strong positive correlation between digestive system diseases and BMI. Perhaps the Russians' diet is quite unhealthy, so as they consume more nutrition, their risk of digestive system illness also increase.

United Kingdom: Similar to U.S., except there is also a negative correlation bewtween respiratory system illnesses and BMI. Perhaps as the U.K. became more developed, people gained the awareness to smoke less and environmental standards went up as well, reducing the number of deaths due to this cause.

Brazil: Nothing out of the ordinary. Developing countries are not that different from developed countries in terms of this correlation.

3. How does the average Child Mortality Rate (CMR) correlate with the causes of mortality at any point in time?

Result

As CMR increases, the death rate for most causes of death increases. This is probably because CMR is only high when a country has poor health infrastructure.

The only exceptions are nervous system and behavioural disorders, which are negatively correlated with CMR.

On a country level, Russia appears to be an exception as it's number of deaths due to digestive system illnesses is also negatively correlated with CMR.

EDA

In [42]:

```
cmrMort = pd.merge(
    longmdf, longcmrdf, on=["CountryCode", "Country", "Year"], how="inner"
)
_cmrMort = pd.merge(longmdf, longcmrdf, on=["CountryCode", "Year"], how="inner")
cmrMort.DeathsPer100k = cmrMort.DeathsPer100k.astype(float)
display(cmrMort)
display(_cmrMort)
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN	71.181438
1	ALB	Albania	2	Neoplasms	1979	NaN	71.181438
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN	71.181438
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN	71.181438
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN	71.181438
...	...	...	...	...	...	...	...
61593	VEN	Venezuela, Bolivarian Republic of	12	Diseases of the skin and subcutaneous tissue	2016	NaN	21.044404
61594	VEN	Venezuela, Bolivarian Republic of	13	Diseases of the musculoskeletal system and con...	2016	NaN	21.044404
61595	VEN	Venezuela, Bolivarian Republic of	14	Diseases of the genitourinary system	2016	NaN	21.044404
61596	VEN	Venezuela, Bolivarian Republic of	15	Pregnancy, childbirth and the puerperium	2016	NaN	21.044404
61597	VEN	Venezuela, Bolivarian Republic of	16	Congenital malformations, deformations and chr...	2016	NaN	21.044404

61598 rows × 7 columns

	CountryCode	Country_x	Cause	CauseName	Year	DeathsPer100k	Country_y	
0	ALB	Albania	1	Certain infectious and parasitic diseases	1979	NaN	Albania	71.18
1	ALB	Albania	2	Neoplasms	1979	NaN	Albania	71.18
2	ALB	Albania	5	Mental and behavioural disorders	1979	NaN	Albania	71.18

	CountryCode	Country_x	Cause	CauseName	Year	DeathsPer100k	Country_y	
3	ALB	Albania	6	Diseases of the nervous system	1979	NaN	Albania	71.18
4	ALB	Albania	7	Diseases of the eye and adnexa	1979	NaN	Albania	71.18
...	...	...	...	...	...	...	...	
61593	VEN	Venezuela, Bolivarian Republic of	12	Diseases of the skin and subcutaneous tissue	2016	NaN	Venezuela, Bolivarian Republic of	21.04
61594	VEN	Venezuela, Bolivarian Republic of	13	Diseases of the musculoskeletal system and con...	2016	NaN	Venezuela, Bolivarian Republic of	21.04
61595	VEN	Venezuela, Bolivarian Republic of	14	Diseases of the genitourinary system	2016	NaN	Venezuela, Bolivarian Republic of	21.04
61596	VEN	Venezuela, Bolivarian Republic of	15	Pregnancy, childbirth and the puerperium	2016	NaN	Venezuela, Bolivarian Republic of	21.04
61597	VEN	Venezuela, Bolivarian Republic of	16	Congenital malformations, deformations and chr...	2016	NaN	Venezuela, Bolivarian Republic of	21.04

61598 rows × 8 columns

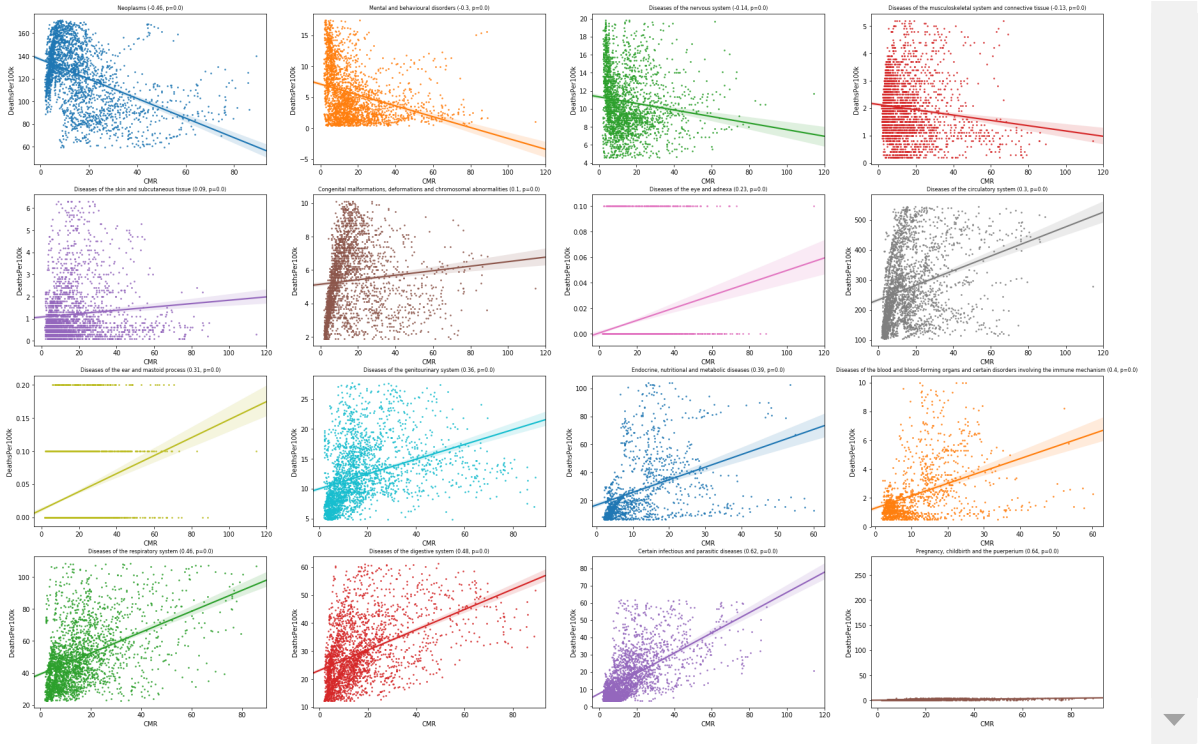
As per BMI, `cmrdf`'s `Country` column is also in accordance with that of `mdf`. Now, let's plot some regplots but without outliers.

In [43]:

```
cmrMort2 = append_correl(remove_outliers(cmrMort).dropna(), "CMR", "DeathsPer100k")
display(cmrMort2)
plot4x4reg(cmrMort2, "CMR")
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl
0	ARG	Argentina	1	Certain infectious and parasitic diseases	1979	29.8	40.094603	0.620171
1	AUS	Australia	1	Certain infectious and parasitic diseases	1979	3.4	11.297138	0.620171
2	BRB	Barbados	1	Certain infectious and parasitic diseases	1979	17.8	24.888679	0.620171
3	BEL	Belgium	1	Certain infectious and parasitic diseases	1979	5.4	12.876635	0.620171
4	CUB	Cuba	1	Certain infectious and parasitic diseases	1979	11.9	19.279995	0.620171
...	...	...	...	...	...	...	...	...
37881	MDA	Moldova, Republic of	4	Endocrine, nutritional and metabolic diseases	2016	8.7	13.149784	0.385809
37882	ROU	Romania	4	Endocrine, nutritional and metabolic diseases	2016	7.3	7.112852	0.385809
37883	SWE	Sweden	4	Endocrine, nutritional and metabolic diseases	2016	10.5	2.287122	0.385809
37884	THA	Thailand	4	Endocrine, nutritional and metabolic diseases	2016	18.9	8.854661	0.385809
37885	USA	United States	4	Endocrine, nutritional and metabolic diseases	2016	21.5	5.764999	0.385809

37886 rows × 9 columns



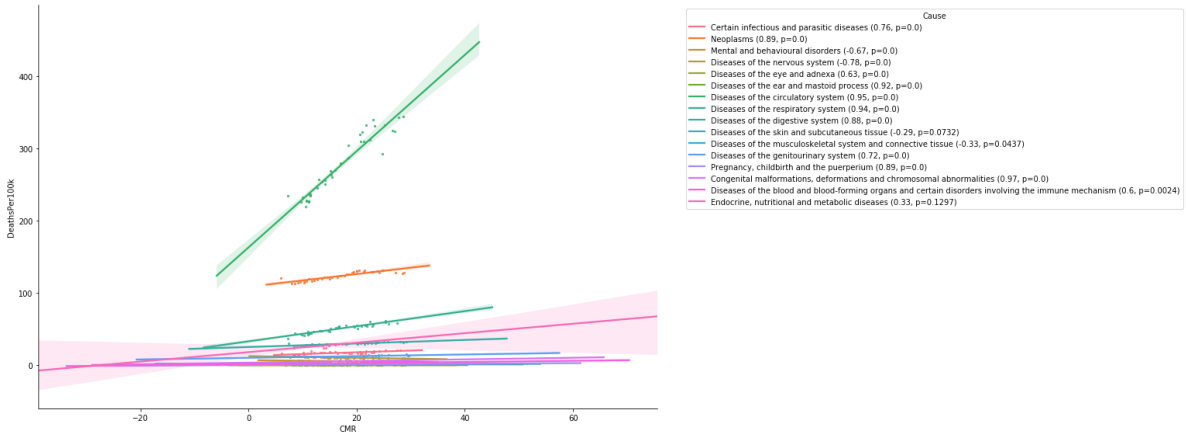
Same as BMI, correlation on a global scale has too much variance to be statistically significant. The cause with the highest correlation is related to childbirth, which makes sense as a high number of the deaths occurring due to childbirth can be classified as "child mortality".

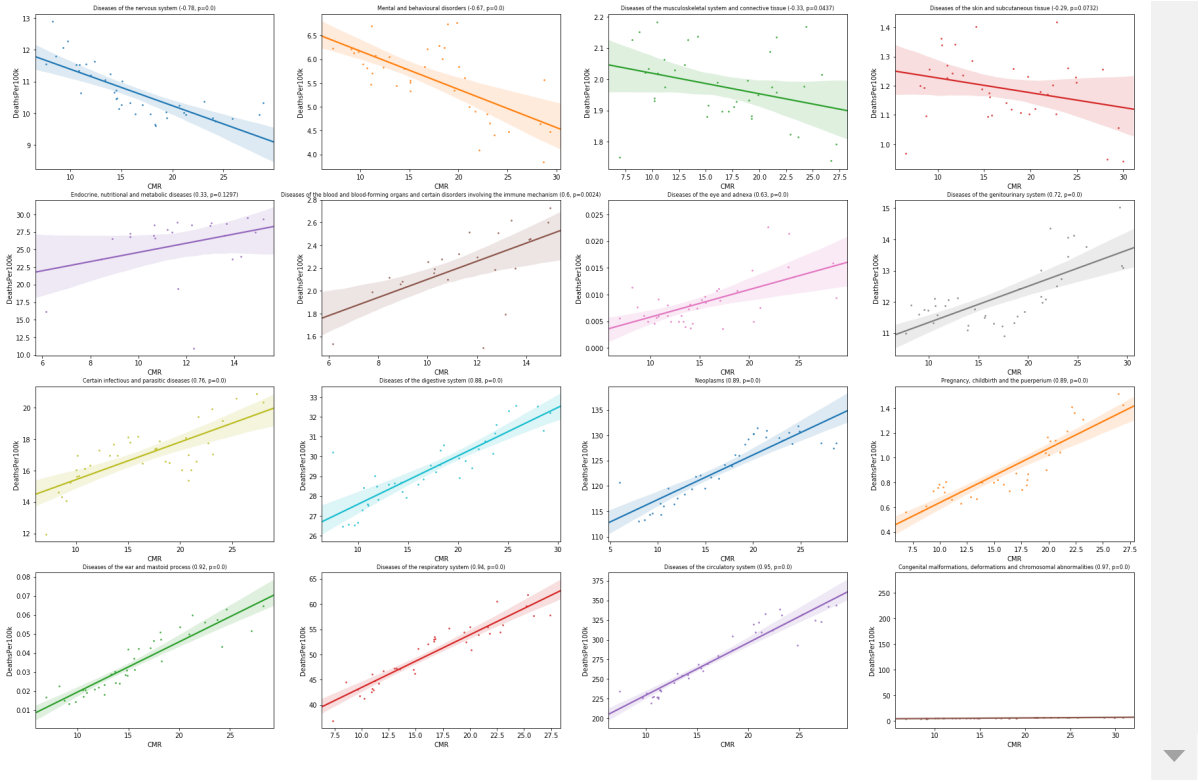
In [44]:

```
1 reg_mean_deaths(cmrMort2, "CMR")
```

	Year	Cause	CauseName	DeathsPer100k	CMR	Correl	pValue
0	1979	1	Certain infectious and parasitic diseases	19.450000	21.749274	0.755045	4.318056e-08
1	1980	1	Certain infectious and parasitic diseases	20.597619	25.321455	0.755045	4.318056e-08
2	1981	1	Certain infectious and parasitic diseases	20.913208	27.414685	0.755045	4.318056e-08
3	1982	1	Certain infectious and parasitic diseases	20.358491	27.987798	0.755045	4.318056e-08
4	1983	1	Certain infectious and parasitic diseases	18.088372	20.594925	0.755045	4.318056e-08
...	...	...	...	...	...	...	...
573	2012	4	Endocrine, nutritional and metabolic diseases	27.324324	9.620851	0.325408	1.297328e-01
574	2013	4	Endocrine, nutritional and metabolic diseases	26.827397	9.601338	0.325408	1.297328e-01
575	2014	4	Endocrine, nutritional and metabolic diseases	26.581690	8.949516	0.325408	1.297328e-01
576	2015	4	Endocrine, nutritional and metabolic diseases	23.637037	8.447201	0.325408	1.297328e-01
577	2016	4	Endocrine, nutritional and metabolic diseases	16.120000	6.185677	0.325408	1.297328e-01

578 rows × 7 columns





The correlations are almost the exact opposite of that of BMI. Diseases of the nervous system and mental disorders are positively correlated with CMR, once again proving my hypothesis that many people with undiagnosed mental disorders are dying young, so they will not be tallied as a death due to those causes.

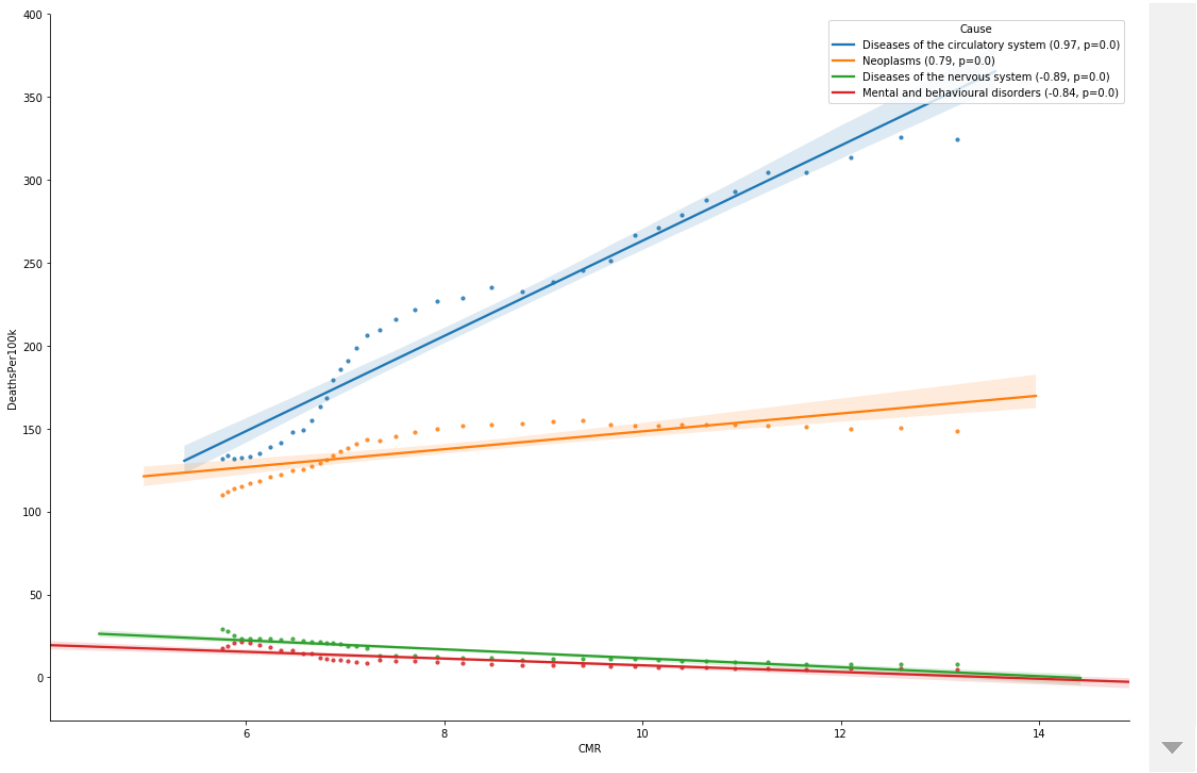
In [45]:

```
1 reg_per_country(append_correl(cmrMort, "CMR", "DeathsPer100k").dropna(), "CMR")
```

```
Enter alpha-3 country code:  usa

United States
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl	
189	USA	United States	9	Diseases of the circulatory system	2016	131.8	5.764999	0.973829	9.
161	USA	United States	9	Diseases of the circulatory system	1988	266.8	9.929447	0.973829	9.
168	USA	United States	9	Diseases of the circulatory system	1995	226.7	7.929688	0.973829	9.
167	USA	United States	9	Diseases of the circulatory system	1994	228.8	8.191276	0.973829	9.
166	USA	United States	9	Diseases of the circulatory system	1993	235.2	8.480499	0.973829	9.
...	...	...	...	...	...	...	...	...	
103	USA	United States	5	Mental and behavioural disorders	2006	14.7	6.666849	-0.840583	4.
102	USA	United States	5	Mental and behavioural disorders	2005	12.1	6.745975	-0.840583	4.
101	USA	United States	5	Mental and behavioural disorders	2004	10.9	6.816018	-0.840583	4.
100	USA	United States	5	Mental and behavioural disorders	2003	10.8	6.880536	-0.840583	4.
78	USA	United States	5	Mental and behavioural disorders	1981	5.2	12.102701	-0.840583	4.

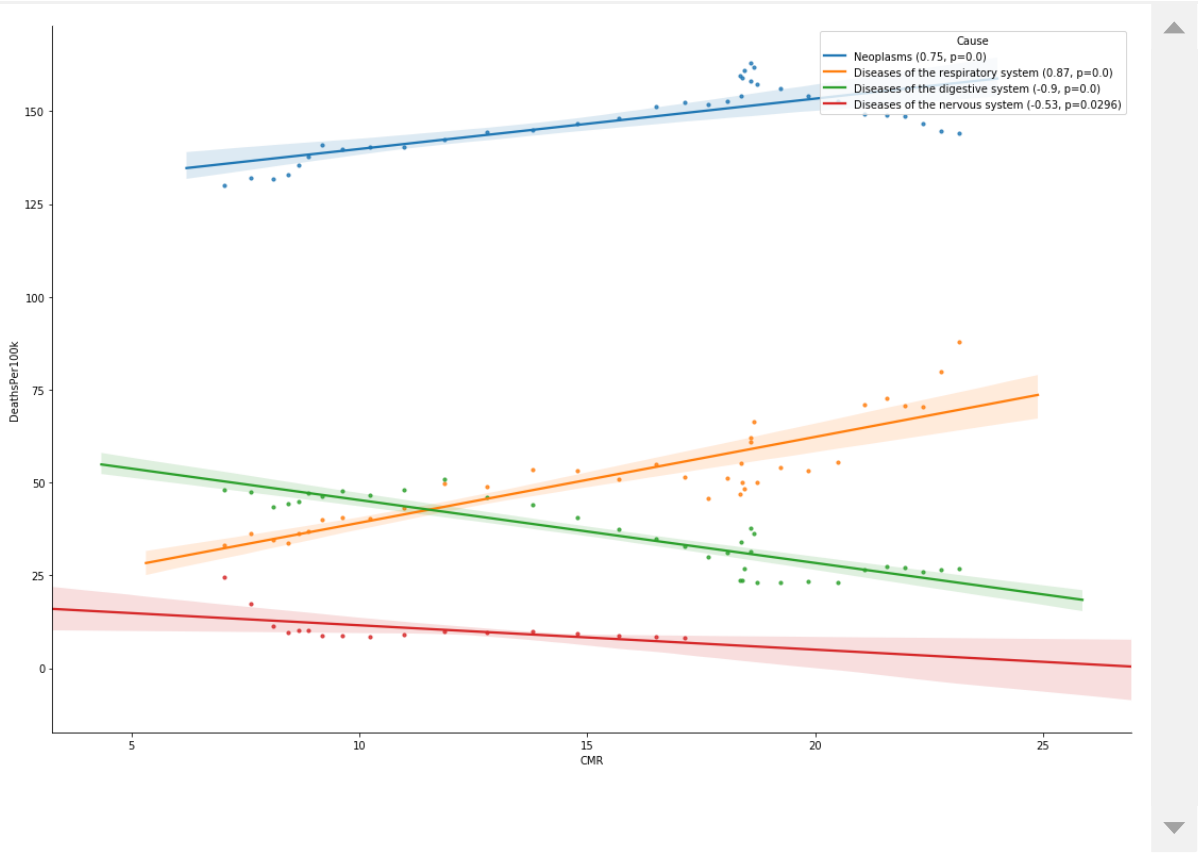
152 rows × 9 columns




```
Enter alpha-3 country code:  rus

Russian Federation
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl
36	RUS	Russian Federation	2	Neoplasms	1980	144.1	23.160614	0.749570
55	RUS	Russian Federation	2	Neoplasms	1999	152.5	17.145943	0.749570
57	RUS	Russian Federation	2	Neoplasms	2001	148.2	15.699803	0.749570
58	RUS	Russian Federation	2	Neoplasms	2002	146.6	14.787114	0.749570
59	RUS	Russian Federation	2	Neoplasms	2003	145.0	13.800184	0.749570
...	...	...	...	...	...	...	...	...
75	RUS	Russian Federation	6	Diseases of the nervous system	2002	9.3	14.787114	-0.527442
74	RUS	Russian Federation	6	Diseases of the nervous system	2001	8.8	15.699803	-0.527442
73	RUS	Russian Federation	6	Diseases of the nervous system	2000	8.4	16.499216	-0.527442
72	RUS	Russian Federation	6	Diseases of the nervous system	1999	8.1	17.145943	-0.527442
77	RUS	Russian Federation	6	Diseases of the nervous system	2004	9.7	12.811291	-0.527442

125 rows × 9 columns

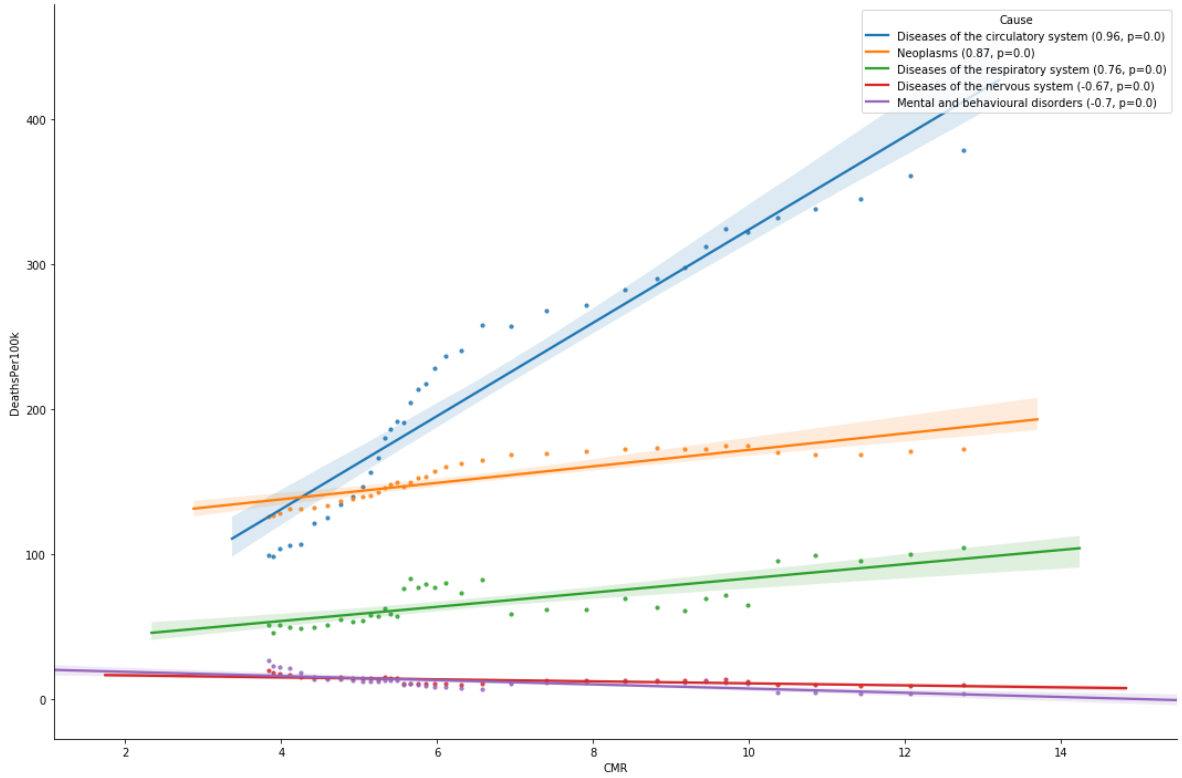



```
Enter alpha-3 country code:  gbr

United Kingdom
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl	
112	GBR	United Kingdom	9	Diseases of the circulatory system	1980	361.3	12.068115	0.961258	3
138	GBR	United Kingdom	9	Diseases of the circulatory system	2006	146.5	5.037681	0.961258	3
131	GBR	United Kingdom	9	Diseases of the circulatory system	1999	205.2	5.659105	0.961258	3
132	GBR	United Kingdom	9	Diseases of the circulatory system	2000	191.3	5.569100	0.961258	3
133	GBR	United Kingdom	9	Diseases of the circulatory system	2001	191.5	5.483651	0.961258	3
...	...	...	...	...	...	...	...	...	
60	GBR	United Kingdom	5	Mental and behavioural disorders	2002	13.0	5.401056	-0.704435	1
59	GBR	United Kingdom	5	Mental and behavioural disorders	2001	13.0	5.483651	-0.704435	1

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl	
58	GBR	United Kingdom	5	Mental and behavioural disorders	2000	10.6	5.569100	-0.704435	1
57	GBR	United Kingdom	5	Mental and behavioural disorders	1999	10.7	5.659105	-0.704435	1
72	GBR	United Kingdom	5	Mental and behavioural disorders	2014	23.4	3.892211	-0.704435	1

185 rows × 9 columns

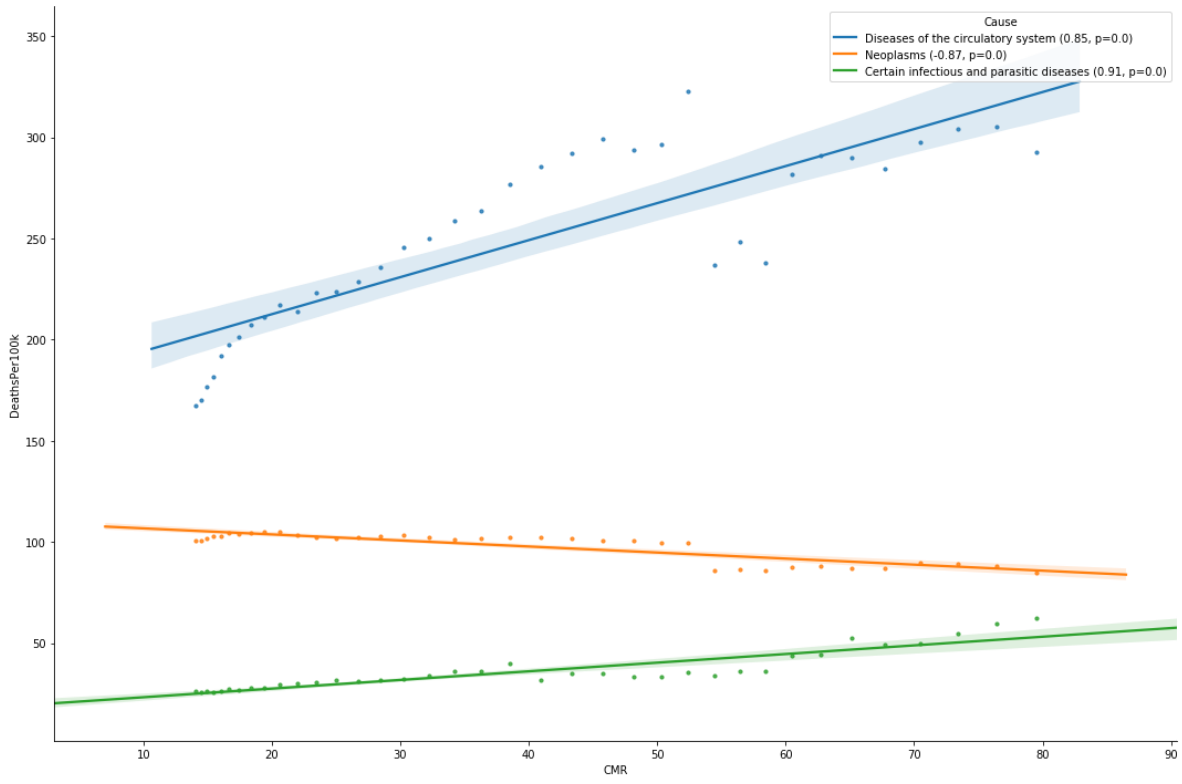



```
Enter alpha-3 country code:  bra

Brazil
```

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl
110	BRA	Brazil	9	Diseases of the circulatory system	2015	167.5	14.046389	0.846563
82	BRA	Brazil	9	Diseases of the circulatory system	1987	237.9	58.420730	0.846563
90	BRA	Brazil	9	Diseases of the circulatory system	1995	285.7	40.903623	0.846563
89	BRA	Brazil	9	Diseases of the circulatory system	1994	292.0	43.342598	0.846563

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	CMR	Correl
88	BRA	Brazil	9	Diseases of the circulatory system	1993	299.4	45.764745	0.846563
...	...	...	...	...	...	...	...	...
24	BRA	Brazil	1	Certain infectious and parasitic diseases	2003	31.5	25.023282	0.906555
23	BRA	Brazil	1	Certain infectious and parasitic diseases	2002	31.1	26.703441	0.906555
22	BRA	Brazil	1	Certain infectious and parasitic diseases	2001	31.7	28.459729	0.906555
21	BRA	Brazil	1	Certain infectious and parasitic diseases	2000	32.0	30.300841	0.906555
0	BRA	Brazil	1	Certain infectious and parasitic diseases	1979	62.6	79.486312	0.906555

111 rows × 9 columns




```
Enter alpha-3 country code:  stop
```

Almost the reverse correlations as BMI. Once again, Russia is the anomaly here, with deaths due to digestive diseases correlating negatively with CMR. I cannot explain this with a direct reason. Perhaps it is due to the same reason as BMI, where as Russia gets more developed, its citizens eat unhealthily enough to develop digestive disorders.

4. Using an index consisting of BMI and CMR, is it possible to predict the trend of some major causes of death globally?

Result

The only cause of death that could be predicted to some extent from these two statistics is illnesses of the circulatory system. The model that has been created works best when the country is in the transition phase between developing and developed.

EDA

We will use the data for every developed country (HDI > 0.9) as over the course of 38 years, most of them have went through several stages of development and can be used as a predictor for the rest of the world in the future. This would be free from the variance (perhaps due to war and instabilities) that's present in developing countries. Feel free to adjust the HDI threshold to something other than 0.9, but from my experience doing so, the resulting model is really all over the place. The scatterplots look to be randomly generated in that case. **The failed graphs have been omitted for the sake of saliency.**

I will not be taking the mean of all countries throughout the years as that produces too few data points and is prone to overfitting.

In [46]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

altogether = pd.merge(
    bmiMort2.drop(['Correl', 'pValue'], axis=1),
    cmrMort2[["CountryCode", "Cause", "Year", "CMR"]],
    on=["CountryCode", "Cause", "Year"],
)

# to show that there is an hdi value for every country code in altogether
# notice number of rows is the same
display(len(altogether))
altogether = pd.merge(altogether, hdidf, on=["CountryCode", "Country"], how="left")
display(len(altogether))

# feel free to uncomment these lines for mean deaths per year,
# but there are too few data points, leading to underfitting
# when it comes to a country by country basis.

# it will not be plotted for the sake of saliency.

# for_mlm = pd.concat([get_mean_deaths(altogether,'BMI'),
#                      get_mean_deaths(altogether,'CMR')['CMR']],
#                     axis=1)
for_mlm = altogether[altogether.HDI > 0.9]
for_mlm
```

37836

37836

Out[46]:

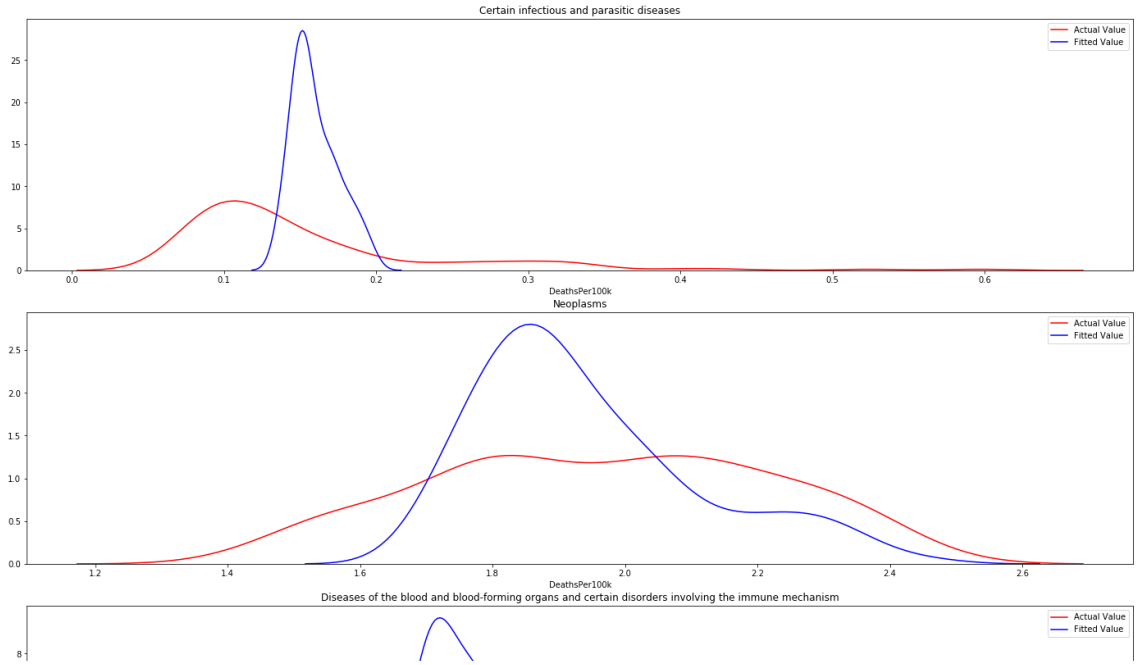
	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	CMR
1	AUS	Australia	1	Certain infectious and parasitic diseases	1979	3.4	24.2	11.297138
3	BEL	Belgium	1	Certain infectious and parasitic diseases	1979	5.4	24.9	12.876635
8	IRL	Ireland	1	Certain infectious and parasitic diseases	1979	8.4	24.0	13.043885
9	ISR	Israel	1	Certain infectious and parasitic diseases	1979	12.1	24.6	16.275730
11	JPN	Japan	1	Certain infectious and parasitic diseases	1979	10.4	22.2	7.832282
...	...	...	...	...	...	...	...	...

	CountryCode	Country	Cause	CauseName	Year	DeathsPer100k	BMI	CMR
37822	AUT	Austria	4	Endocrine, nutritional and metabolic diseases	2016	19.3	25.6	2.993266
37827	ISL	Iceland	4	Endocrine, nutritional and metabolic diseases	2016	8.2	26.2	1.707952
37829	NLD	Netherlands	4	Endocrine, nutritional and metabolic diseases	2016	9.4	25.6	3.428635
37833	SWE	Sweden	4	Endocrine, nutritional and metabolic diseases	2016	10.5	26.0	2.287122
37835	USA	United States	4	Endocrine, nutritional and metabolic diseases	2016	21.5	28.9	5.764999

With the combined dataset manufactured, we can now proceed to create a multiple linear regression model using BMI and CMR to predict any cause of death we so choose. This data is free from outliers. `distplot` s will be used to determine roughly how good a model they are.

In [47]:

```
def build_model(cause):
    mlm = LinearRegression()
    by_cause = for_mlm.query("Cause==@cause")
    # normalize
    by_cause.DeathsPer100k /= (
        by_cause.DeathsPer100k.max() - by_cause.DeathsPer100k.min()
    )
    x_train, x_test, y_train, y_test = train_test_split(
        by_cause[["CMR", "BMI"]],
        by_cause["DeathsPer100k"],
        test_size=0.2,
        random_state=4132,
    )
    mlm.fit(x_train, y_train)
    return (mlm.intercept_, mlm.coef_, mlm.predict(x_test), y_test)


fig, axes = plt.subplots(len(disease_codes), 1, figsize=(20, 80))
fig.tight_layout(pad=3.0)
models = []
for cause in range(1, len(disease_codes) + 1):
    model = build_model(cause)
    name = disease_codes.loc[cause, "name"]
    models.append([name, mean_squared_error(model[3], model[2]), model])
    ax1 = sns.distplot(
        model[3], hist=False, color="r", label="Actual Value", ax=axes[cause - 1]
    )
    axes[cause - 1].set_title(name)
    sns.distplot(model[2], hist=False, color="b", label="Fitted Value", ax=ax1)
```

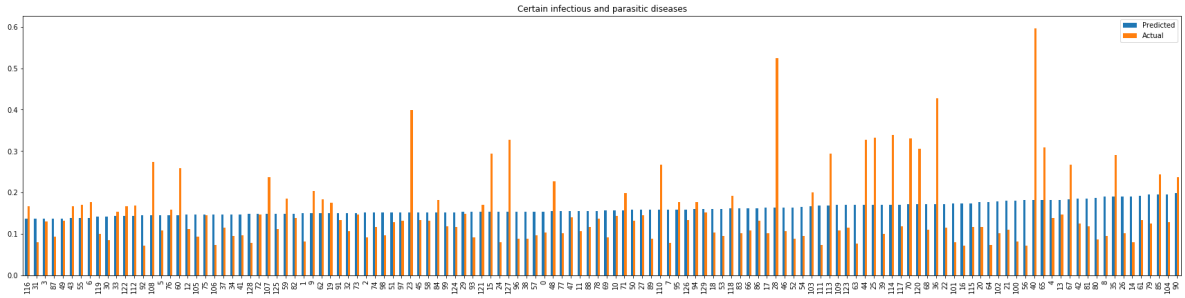


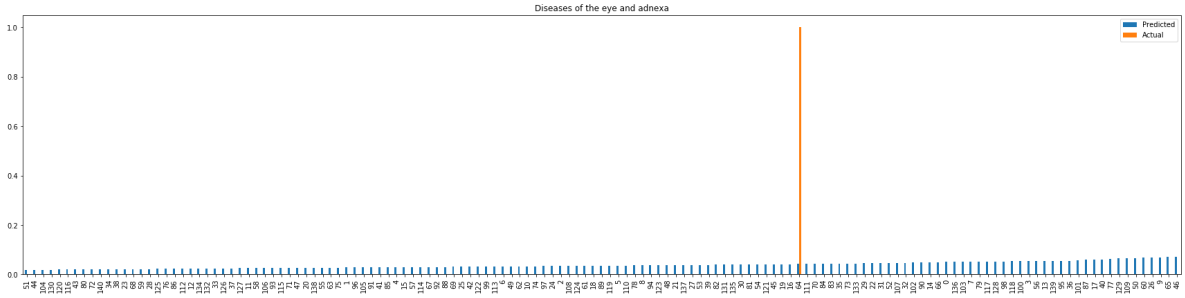
The model is really bad at predictions for most causes of death, so let's narrow it down a bit.

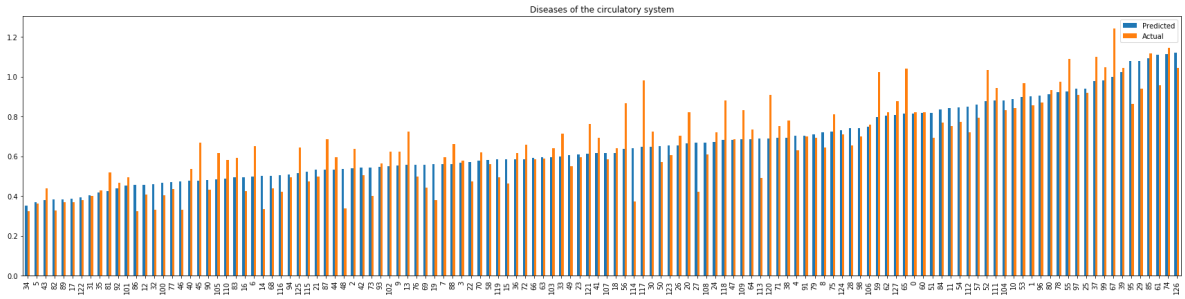
Now, let's take the top five models ranked by mean-squared-error and see how they fare using some bar plots.

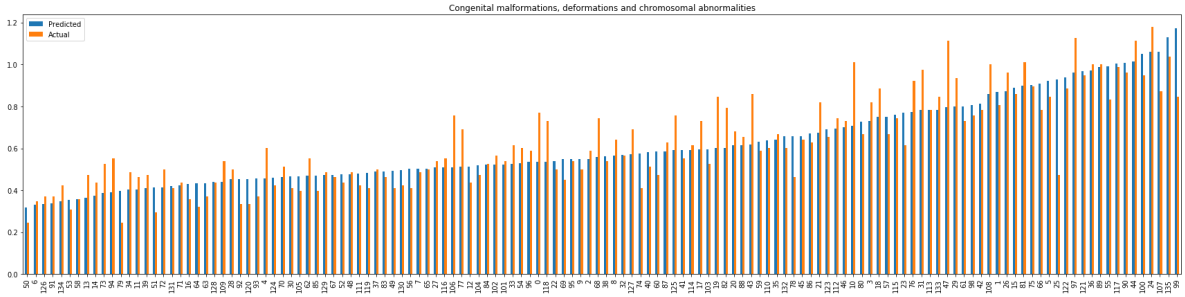
In [48]:

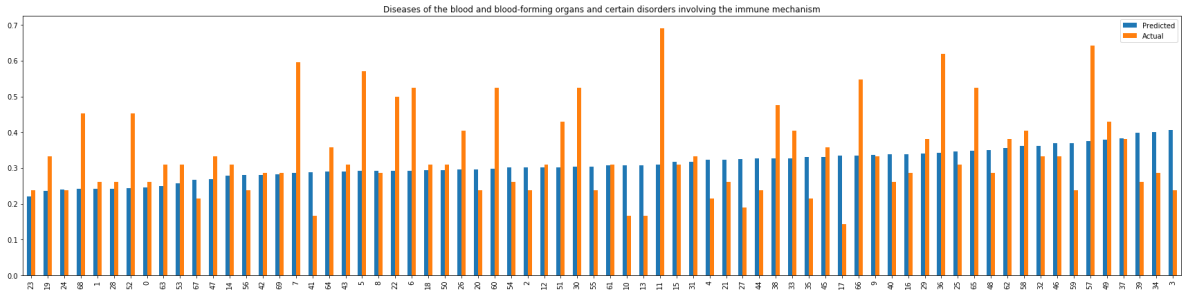
```
models.sort(key=lambda x: x[1])
fig, axes = plt.subplots(5, figsize=(30, 40))
for x in range(5):
    name = models[x][0]
    mse = models[x][1]
    actual_vs_predicted = pd.DataFrame([models[x][2][2], models[x][2][3]]).T
    axes[x].set_title(name)
    actual_vs_predicted.columns = ["Predicted", "Actual"]
    actual_vs_predicted.sort_values(by="Predicted", inplace=True)
    actual_vs_predicted.plot(kind="bar", ax=axes[x])
```





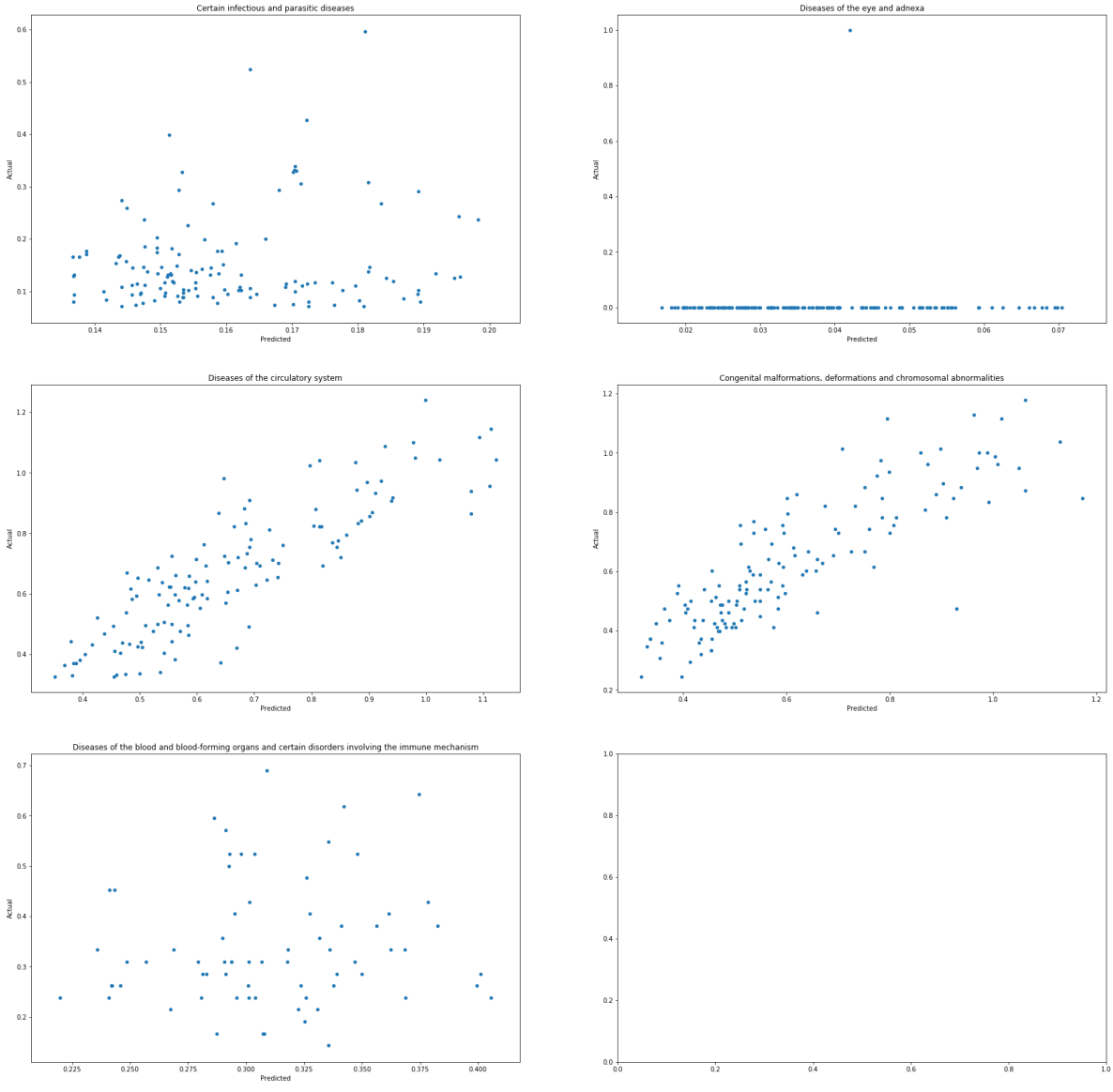






In [49]:

```
fig, axes = plt.subplots(3,2, figsize=(30, 30))
for x in range(5):
    name = models[x][0]
    mse = models[x][1]
    actual_vs_predicted = pd.DataFrame([models[x][2][2], models[x][2][3]]).T
    axes.ravel()[x].set_title(name)
    actual_vs_predicted.columns = ["Predicted", "Actual"]
    actual_vs_predicted.sort_values(by="Predicted", inplace=True)
    actual_vs_predicted.plot(kind="scatter", x='Predicted', y='Actual',ax=axes.ravel()
```



The model for eye disease is clearly inaccurate. Diseases relating to blood has too few data points and is also too inaccurate. The only reason the model for infectious and parasitic diseases has low MSE is because few people die from infectious diseases in developed countries. That is clearly not a good model either.

Congenital malformations is directly correlated with CMR for obvious reasons, so it is not interesting. The only one left that is somewhat interesting is diseases of the circulatory system. Let's do some more analysis on that.

In [50]:

```
1 display(models[2])
```

```
['Diseases of the circulatory system',
 0.01158372364476246,
 (1.4572018205979171,
  array([ 0.06097839, -0.04670699]),
  array([0.81425328, 0.90133845, 0.53914445, 0.56840262, 0.7025814 ,
         0.36899801, 0.49704923, 0.56184976, 0.72200838, 0.55433993,
         0.8862405 , 0.84420291, 0.4563368 , 0.55605172, 0.50027008,
         0.58473738, 0.49651417, 0.38878218, 0.61824463, 0.56133623,
         0.66455559, 0.53147336, 0.57068838, 0.60881048, 0.67198644,
         0.9411238 , 0.65503264, 0.6698129 , 0.74133971, 1.07797976,
         0.6476085 , 0.40456543, 0.45968747, 0.59838638, 0.35249979,
         0.41716189, 0.58513929, 0.97761413, 0.6942383 , 1.02330925,
         0.4762497 , 0.61602216, 0.54211881, 0.37986249, 0.53374805,
         0.47761484, 0.47570019, 0.68432329, 0.53618502, 0.60568221,
         0.64979416, 0.81959285, 0.87629052, 0.8966906 , 0.84603105,
         0.92771282, 0.63774494, 0.86043782, 0.58243312, 0.79754763,
         0.81711635, 1.11109557, 0.80379768, 0.59582106, 0.68669485,
         0.8140288 , 0.59257648, 0.99923773, 0.50242586, 0.55659363,
         0.5785617 , 0.69160118, 0.58649357, 0.54296214, 1.1131506 ,
         0.72593756, 0.55614739, 0.46982978, 0.92167412, 0.70961587,
         0.91136883, 0.42583398, 0.38228499, 0.49471431, 0.83630508,
         1.09303275, 0.45501612, 0.53162798, 0.56239656, 0.38489047,
         0.48247593, 0.70361165, 0.43843871, 0.5488142 , 0.50971251,
         1.07787174, 0.90601686, 0.93966934, 0.74241416, 0.98067893,
         0.46690957, 0.45453409, 0.55178391, 0.59710476, 0.8811169 ,
         0.48438655, 0.75006767, 0.61750098, 0.6702431 , 0.68500385,
         0.4863754 , 0.87932318, 0.85128903, 0.69048756, 0.64120953,
         0.52387482, 0.5048359 , 0.64662072, 0.68286349, 0.58394156,
         0.69114843, 0.61173977, 0.39537349, 0.65404497, 0.73192293,
         0.51520287, 1.12203386, 0.80713025]),
  15215    0.821342
  14919    0.855200
  15827    0.636941
  16225    0.578295
  15634    0.629686
             ...
  15840    0.605804
  15720    0.711004
  16378    0.645405
  14862    1.044135
  15451    0.879081
  Name: DeathsPer100k, Length: 128, dtype: float64)]
```

In [51]:

```
model_data = models[2][2]
by_cause = for_mlm.query("Cause==9")
scale_factor = by_cause.DeathsPer100k.max() - by_cause.DeathsPer100k.min()
values = np.array([model_data[0], model_data[1][0], model_data[1][1]])
values *= scale_factor


def predict_point(row):
#     display(row)
    bmi, cmr = row.BMI, row.CMR
    return values[0] + values[1]*cmr + values[2]*bmi


print(
    "Equation for Number of Deaths/100k: y = "
    + str(values[0])
    + " + "
    + str(values[1])
    + "a + "
    + str(values[2])
    + "b"
)
```

```
Equation for Number of Deaths/100k: y = 482.042362253791 + 20.17164976806169
5a + -15.450673344144477b
```

Where a is CMR and b is BMI.

Let's try to use this model to predict a country's diseases of the circulatory system deaths per 100k as time progresses.

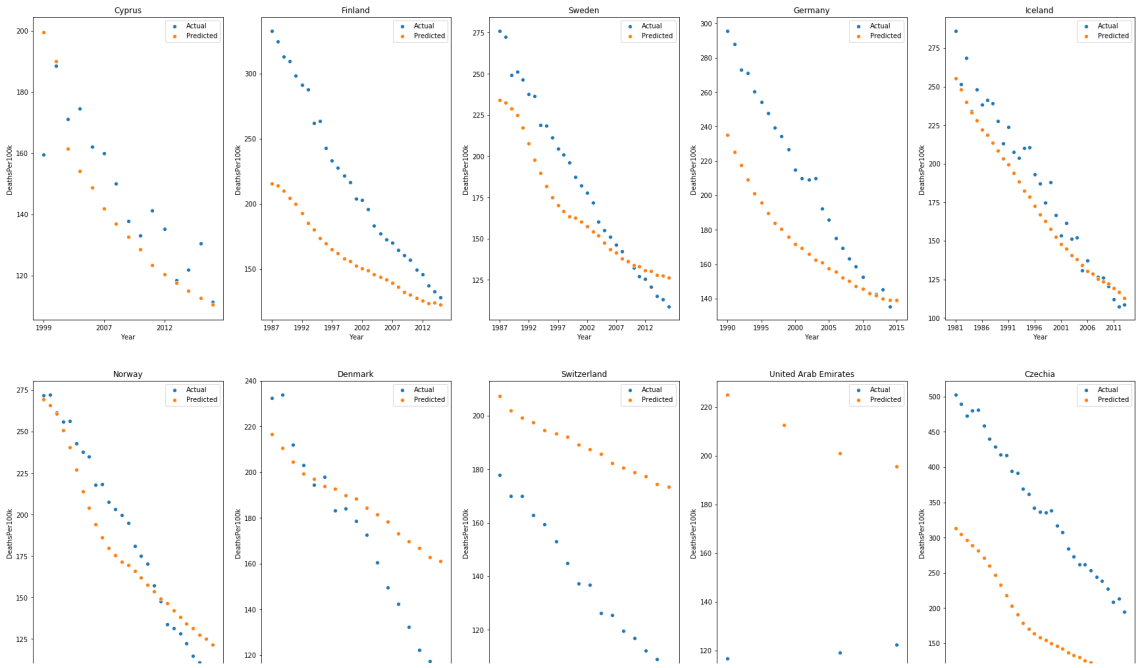
In [52]:

```
mses = []
for country in altogether.CountryCode.unique():
    df = altogether[(altogether.CountryCode==country)&(altogether.Cause==cause)]
    df['Predicted'] = df.apply(predict_point,axis=1)
    mses.append([country, mean_squared_error(df.Predicted, df.DeathsPer100k)])
mses.sort(key=lambda x: x[1])
mses = pd.DataFrame(mses,columns=['CountryCode', 'MSE'])
with pd.option_context("display.max_rows", None):
    display(mses)
```

	CountryCode	MSE
0	CYP	1.756630e+04
1	FIN	2.493592e+04
2	SWE	2.763274e+04
3	DEU	2.895980e+04
4	ISL	3.039817e+04
5	NOR	3.090837e+04
6	DNK	3.157552e+04
7	CHE	3.257105e+04
8	ARE	4.147307e+04
9	CZE	4.209307e+04
10	ESP	4.216831e+04

In [53]:

```
fig, axes = plt.subplots(10,5,figsize=(30,100))
def predict_countries(countries, cause):
    for x in range(len(countries)):
        country = countries[x]
        ax = axes.ravel()[x]
        df = altogether[(altogether.CountryCode==country)&(altogether.Cause==cause)]
        df['Predicted'] = df.apply(predict_point,axis=1)
        sns.scatterplot(data=df, x='Year', y='DeathsPer100k',ax=ax)
        sns.scatterplot(data=df, x='Year', y='Predicted',ax=ax)
        ax.set_xticks(ax.get_xticks()[::5])
        ax.set_title(df.iloc[0]['Country'])
        ax.set_ylabel('DeathsPer100k')
        ax.legend(['Actual','Predicted'])
predict_countries(mses.loc[:49,'CountryCode'],9)
```

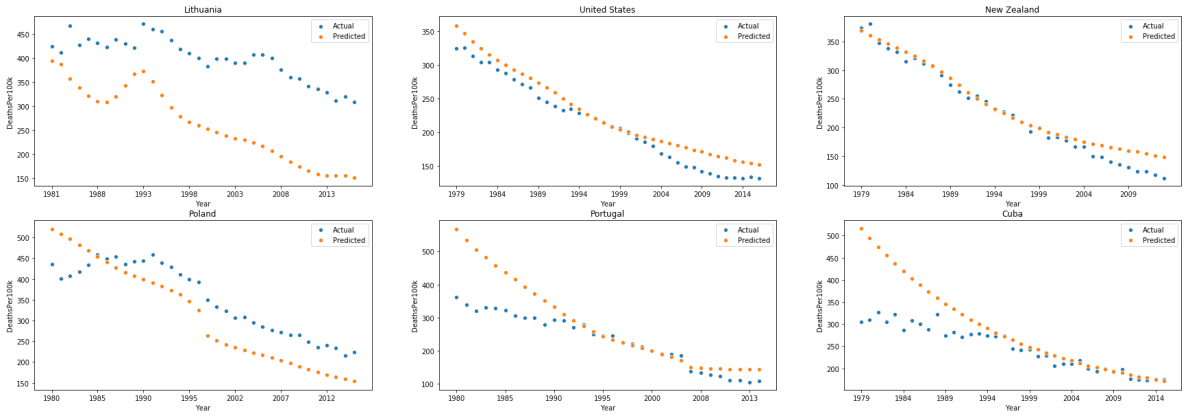


Nice! This model appears to work for most developed and in-transition countries. A few countries look to be off by a simple translation parallel to the y-axis. This may be perhaps due to a genetic predisposition to die less from heart-related disorders or some other external factor. Countries can calibrate this model for their own usage with a few years of mortality, CMR and BMI data.

I will now show you in detail some curated graphs that display this model's effectiveness.

In [54]:

```
curated = ['LTU','USA','NZL','POL','PRT','CUB']
fig, axes = plt.subplots(2,3,figsize=(30,10))
predict_countries(curated,9)
```



Lithuania: The prediction is off by a simple translation, but I would like to draw attention to around the 1993 mark. When the actual figures unexpectedly (at least against the year) increased, the predicted values also increased. This shows that something happened in Lithuania that year which shifted deaths due to cardiovascular causes and BMI or CMR.

Poland: Same as Lithuania but at the 1996 mark. I wonder if it's due to the Soviet Union disintegrating.

United States: Pretty good prediction for one of the world's most important countries, starts deviating a bit towards the end, perhaps as the country reaches stages of late-development.

New Zealand: This model in general works well for developed countries, which is expected as the model is based off their data. The other option (use all data) would have way too much variability due to developing countries, and is much worse than this.

Cuba: As Cuba transitions towards being less of a developing country and in general becoming more stable, the actual figures become more in line with the predicted ones.

Portugal: Same as Cuba. This model is quite effective during the transition stage between developing and developed.

In [55]:

```
def predict_country(cause):
    country = ""
    while country != "STOP":
        country = input("Enter alpha-3 country code: ").upper()
        if country in altogether.CountryCode.values:
            df = altogether[(altogether.CountryCode==country)&(altogether.Cause==cause
            df['Predicted'] = df.apply(predict_point,axis=1)
            display(df)
            plt.figure(figsize=(20,10))
            ax = sns.scatterplot(data=df, x='Year', y='DeathsPer100k')
            sns.scatterplot(data=df, x='Year', y='Predicted',ax=ax)
            ax.set_title(df.iloc[0]['Country'])
            ax.set_ylabel('DeathsPer100k')
            ax.legend(['Actual','Predicted'])
            plt.show()
# uncomment this to choose any country you want to be displayed
# predict_country(9)
```

Conclusion and Recommendations

Countries with exceptionally low rates of death due to mental disorders should monitor the mental health of children more closely, and diagnose mental disorders early on. This would ensure that appropriate care is given to them but the truth is that in the end many of them will die later on life nonetheless due to their condition. Therefore, when a country's rate of death due to mental disorders is high, it means that it has succeeded in given it's disabled children a few extra years of life.

Countries can use their leading cause of death to judge their current state of development. If it is infectious diseases, then they are far behind the rest of the world and should work on sanitation. If it is illnesses of the circulatory system, then they are average and should implement measures such as advocating for a healthy lifestyle. If it is neoplasms, then they are very developed and there is not much they can do about it unless cancer can be cured.

Countries in the transition phase of their development can use the model in question 4 to roughly determine the number of deaths due to illnesses of the circulatory system. They can then compare the predicted value with the actual value to see if they have a disproportionately high or low number of deaths due to that cause at their current stage of development. If it is overly high, then the country should work on equipping its public spaces with facilities necessary (e.g. AEDs) to prevent deaths due to heart disease from occurring.

Russia should really advocate for a healthier diet among its citizens.