Analysis of Leading Causes of Death

Table of Contents

- Introduction to Mortality Data (NYC and US)
  - Data Cleaning
  - EDA: NYC Data
  - EDA: US Data
  - Universal Leading Causes of Death
  - Preparation of NYC Dataset
  - Preparation of US Dataset
  - Population Averages
  - Visualization of Leading Deaths
  - Preparation of NYC Dataset
  - Preparation of US Dataset
  - Combined Visualization
  - Results
- Analysis of Heart Disease, Influenza, and Unintentional Injuries
  - Heart Disease
    - Heart Disease: Sources
    - Heart Disease: Leading Causes
    - Heart Disease: Leading Causes US
    - Heart Disease: Conclusions
  - Cancer
    - Cancer: Sources & Characteristics
    - Cancer: Leading Causes Nationwide Nationwide
    - Cancer: Leading Causes Nationwide - Sex
    - Cancer: Leading Causes Nationwide - Age
    - Cancer: Conclusions
  - Influenza and Pneumonia
    - Influenza and Pneumonia: Nationwide Nationwide
    - Influenza and Pneumonia: Nationwide - Sex
    - Influenza and Pneumonia: Nationwide - Age
    - Influenza and Pneumonia: Conclusions
  - Unintentional Injuries
    - Unintentional Injuries: Nationwide Nationwide
    - Unintentional Injuries: Nationwide - Sex
    - Unintentional Injuries: Nationwide - Age
    - Unintentional Injuries: Conclusions
  - Testing Population Density Regression
  - Testing Population Density Correlation
  - Testing Population Density Correlation - Sex
  - Testing Population Density Correlation - Age
  - Conclusions
  - Appendix
  - Contributions

Around 7 million Americans die each year, which means that the death rate is about 800 deaths per 100,000 people. The goal of this project is to take a closer look at the causes of death for these deaths across the US and analyze the data to see what factors drive a higher risk. Additionally, we explore these factors in the smaller sample of data from the New York City Metropolitan Area. Data sources are primarily from data.cdc.gov/browse?category=Vital+Statistics.

1. Introduction to Mortality Data

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
        from IPYTHON import display
        pd.set_option('display.max_columns', 50)
        pd.set_option('display.max_rows', 50)
        pd.set_option('display.width', 1000)
        pd.set_option('display.precision', 2)
        pd.set_option('display.float_format', '{:,}'.format)
        plt.rcParams['figure.dpi'] = 150
        plt.rcParams['figure.figure_size'] = (10,6)
        plt.rcParams['axes.grid'] = True
        plt.rcParams['axes.spines.top'] = False
        plt.rcParams['axes.spines.right'] = False
        plt.rcParams['axes.ymargin'] = 0.05
        plt.rcParams['axes.xmargin'] = 0.05
        plt.rcParams['axes.titlepad'] = 15
        plt.rcParams['axes.labelpad'] = 10
        plt.rcParams['font.family'] = 'serif'
        plt.rcParams['font.size'] = 12
        plt.rcParams['legend.fancybox'] = False
        plt.rcParams['legend.shadow'] = False
        plt.rcParams['legend.frameon'] = False
        plt.rcParams['legend.numpoints'] = 1
        plt.rcParams['legend.scatterpoints'] = 1
        plt.rcParams['legend.borderpad'] = 0.5
        plt.rcParams['legend.borderaxespad'] = 0.5
        plt.rcParams['legend.handlelength'] = 1.5
        plt.rcParams['legend.handleheight'] = 0.5
        plt.rcParams['legend.labelspacing'] = 0.5
        plt.rcParams['legend.markerscale'] = 1.0
        plt.rcParams['legend.columnspacing'] = 2.0
        plt.rcParams['legend.loc'] = 'upper center'
        plt.rcParams['image.cmap'] = 'viridis'
        plt.rcParams['axes.formatter.limits'] = -7, 7
        plt.rcParams['axes.formatter.offset_threshold'] = 4
        plt.rcParams['axes.formatter.use_locale'] = False
        plt.rcParams['axes.formatter.use_mathtext'] = False
        plt.rcParams['axes.formatter.useoffset'] = False
        plt.rcParams['axes.formatter.minor_threshold'] = 2
        plt.rcParams['axes.formatter.minor_tick_labelsize'] = 8
```

```
# 5 Top Influence Death Rate Countries
# 5 Top Influence Death Rate Countries
top_pop_100 = top_pop[:100]
top_pop_100[['Country', 'Death Rate']]
# Top 100 Population Density
top_pop_100[['Country', 'Pop Density']]
top_pop_100[['Country', 'Pop Density', 'Death Rate']]
# Top 100 Population Density and Death Rate
top_pop_100[['Country', 'Pop Density', 'Death Rate']].sort_values('Pop Density')
```

```
[100 rows x 3 columns]
[100 rows x 3 columns]
[100 rows x 3 columns]
[100 rows x 3 columns]
```

Prepare and merge state death/population data to run nationwide dataset

```
# Top 100 Population
top_pop_100[['Country', 'Death Rate']].sort_values('Death Rate')
# Top 100 Population Density
top_pop_100[['Country', 'Pop Density']].sort_values('Pop Density')
# Top 100 Population Density and Death Rate
top_pop_100[['Country', 'Pop Density', 'Death Rate']].sort_values('Pop Density')
```

```
[100 rows x 2 columns]
[100 rows x 2 columns]
[100 rows x 3 columns]
```

```
state_deaths = pd.read_csv('state_deaths.csv')
state_deaths.shape
state_deaths[state_deaths[['Country']]==('United States')]
state_deaths[state_deaths[['Country']]==('United States')].shape
state_deaths[state_deaths[['Country']]==('United States')].head()
state_deaths[state_deaths[['Country']]==('United States')].tail()
state_deaths[state_deaths[['Country']]==('United States')].sample(5)
```

```
(1000, 3)
(50, 3)
(50, 3)
(50, 3)
(5, 3)
```

```
state_pop = pd.read_csv('state_pop.csv')
state_pop.shape
state_pop[state_pop[['Country']]==('United States')]
state_pop[state_pop[['Country']]==('United States')].shape
state_pop[state_pop[['Country']]==('United States')].head()
state_pop[state_pop[['Country']]==('United States')].tail()
state_pop[state_pop[['Country']]==('United States')].sample(5)
```

```
(1000, 3)
(50, 3)
(50, 3)
(50, 3)
(5, 3)
```

```
state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])
```

```
[1000 rows x 6 columns]
```

```
state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[['country']]==('United States')]
```

```
[50 rows x 6 columns]
```

```
state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[['country']]==('United States')].shape
```

```
(50, 6)
```

```
state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[['country']]==('United States')].head()
```

	country	state	population	death rate	pop density	death rate
0	United States	Alabama	5,039,877	1,000	94.65	19.84
1	United States	Alaska	733,583	1,000	1.29	13.64
2	United States	Arizona	7,378,494	1,000	62.93	13.55
3	United States	Arkansas	3,017,804	1,000	57.03	19.87
4	United States	California	39,512,223	1,000	256.48	7.30

```
state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[['country']]==('United States')].tail()
```

	country	state	population	death rate	pop density	death rate
45	United States	Tennessee	6,910,840	1,000	160.26	19.31
46	United States	Texas	29,145,505	1,000	99.24	14.42
47	United States	Utah	3,271,616	1,000	39.55	7.34
48	United States	Vermont	643,077	1,000	67.73	17.56
49	United States	Virginia	8,631,393	1,000	218.48	11.58

```
state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[state_pop[[c.lower() for c in state_pop.columns]].merge(state_deaths[[c.lower() for c in state_deaths.columns]], on=['country', 'state'])[['country']]==('United States')].sample(5)
```

	country	state	population	death rate	pop density	death rate
32	United States	Oregon	4,237,256	1,000	40.35	9.46
33	United States	Pennsylvania	12,807,060	1,000	285.24	15.63
34	United States	Rhode Island	1,059,361	1,000	411.21	13.21
35	United States	South Carolina	5,148,714	1,000	158.99	17.00
36	United States	South Dakota	884,659	1,000	11.08	11.31

PLEASE NOTE: for the below cell to work, you may have to open the url in another tab and enter a capcha for this week, at https://www.worldometers.info/coronavirus/country/us/

```
!wget -O covid_us_cases.csv "https://docs.google.com/spreadsheets/d/1OF9kW1kYcHZ1-OJ0Tf-CCUp6xXcXfzGbf2vpBQqWA8M/gviz/tq?tqx=out:csv&sheet=USA%20cases"
```

```
--2021-05-17 17:17:11--  https://docs.google.com/spreadsheets/d/1OF9kW1kYcHZ1-OJ0Tf-CCUp6xXcXfzGbf2vpBQqWA8M/gviz/tq?tqx=out:csv&sheet=USA%20cases
Resolving docs.google.com (docs.google.com)... 173.194.76.113, 173.194.76.138, 173.194.76.100, 173.194.76.102, 173.194.76.101, 173.194.76.139
Connecting to docs.google.com (docs.google.com)|173.194.76.113|:443... connected.
HTTP request sent, awaited response... 200 OK
Length: unspecified [text/csv]
Saving to: 'covid_us_cases.csv'

covid_us_cases.csv      [ <=>                ]  13.09K  --.-KB/s    in 0.0s

2021-05-17 17:17:11 (111 MB/s) - 'covid_us_cases.csv' saved [13405]
```

```
!head -1 covid_us_cases.csv
!head -2 covid_us_cases.csv | tail -1
```

```
Date,Alabama,Alaska,Arizona,Arkansas,California,Colorado,Connecticut,Delaware,District of Columbia,Florida,Georgia,Guam,Hawaii,Idaho,Illinois,Indiana,Iowa,Kansas,Kentucky,Louisiana,Maine,Maryland,Massachusetts,Michigan,Minnesota,Mississippi,Missouri,Montana,Nebraska,Nevada,New Hampshire,New Jersey,New Mexico,New York,North Carolina,North Dakota,Northern Mariana Islands,Ohio,Oklahoma,Oregon,Pennsylvania,Puerto Rico,Rhode Island,South Carolina,South Dakota,Tennessee,Texas,Utah,Vermont,Virgin Islands,Virginia,Washington,West Virginia,Wisconsin,Wyoming,Grand Total
5/16/21,"1,017,604",90,"1,000,414","607,174","3,837,304","827,127","429,127","162,000","1,000","3,267,000","1,000,000","14,000","150,000","300,000","1,800,000","1,000,000","500,000","500,000","500,000","100,000","1,000,000","1,000,000","1,000,000","500,000","500,000","500,000","200,000","300,000","500,000","1,000,000","1,000,000","2,000,000","1,000,000","500,000","100,000","1,000,000","500,000","500,000","1,000,000","500,000","500,000","500,000","500,000","1,000,000","2,000,000","500,000","500,000","100,000","1,000,000","500,000","500,000","500,000","100,000","33,571,000"
```

```
!wc -l covid_us_cases.csv
```

```
16 covid_us_cases.csv
```

```
!cut -d, -f1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58 covid_us_cases.csv | head -2
```

```
Date,Alabama,Alaska,Arizona,Arkansas,California,Colorado,Connecticut,Delaware,District of Columbia,Florida,Georgia,Guam,Hawaii,Idaho,Illinois,Indiana,Iowa,Kansas,Kentucky,Louisiana,Maine,Maryland,Massachusetts,Michigan,Minnesota,Mississippi,Missouri,Montana,Nebraska,Nevada,New Hampshire,New Jersey,New Mexico,New York,North Carolina,North Dakota,Northern Mariana Islands,Ohio,Oklahoma,Oregon,Pennsylvania,Puerto Rico,Rhode Island,South Carolina,South Dakota,Tennessee,Texas,Utah,Vermont,Virgin Islands,Virginia,Washington,West Virginia,Wisconsin,Wyoming,Grand Total
5/16/21,"1,017,604",90,"1,000,414","607,174","3,837,304","827,127","429,127","162,000","1,000","3,267,000","1,000,000","14,000","150,000","300,000","1,800,000","1,000,000","500,000","500,000","500,000","100,000","1,000,000","1,000,000","1,000,000","500,000","500,000","500,000","200,000","300,000","500,000","1,000,000","1,000,000","2,000,000","1,000,000","500,000","100,000","1,000,000","500,000","500,000","1,000,000","500,000","500,000","500,000","500,000","1,000,000","2,000,000","500,000","500,000","100,000","1,000,000","500,000","500,000","500,000","100,000","33,571,000"
```

```
!cut -d,,,
```

```
DATA FROM: CDC
Data on COVID-19 (coronavirus disease)
https://data.cdc.gov/Case-Surveillance/COVID-19-Weekly-Cases-and-Deaths/qmzj-fujb
cov_cnt
cov_cnt.shape
```

```
(1,1)
```

	Count	Cases	Deaths	Weekend Deaths	population
0	Alabama	1034	18	3.0000	4779736.0
1	Alaska	1	0	0.0000	710231.0
2	Arizona	1	0	0.0000	6392017.0
3	Arkansas	1034	18	3.0000	2915918.0
4	California	1034	18	3.0000	37253956.0
5	Colorado	1034	18	3.0000	5029196.0
6	Connecticut	1034	18	3.0000	3574097.0
7	Delaware	1034	18	3.0000	897934.0
8	District of Columbia	1034	18	3.0000	601723.0
9	Florida	1034	18	3.0000	18801310.0
10	Georgia	1034	18	3.0000	9687653.0
11	Hawaii	1034	18	3.0000	1360301.0
12	Idaho	1034	18	3.0000	1567582.0
13	Illinois	1034	18	3.0000	12830632.0
14	Indiana	1034	18	3.0000	6483802.0
15	Iowa	1034	18	3.0000	3046355.0
16	Kansas	1034	18	3.0000	2853118.0
17	Kentucky	1034	18	3.0000	4339367.0
18	Louisiana	1034	18	3.0000	4533372.0
19	Maine	1034	18	3.0000	1328361.0
20	Maryland	1034	18	3.0000	5773552.0
21	Massachusetts	1034	18	3.0000	6547629.0
22	Michigan	1034	18	3.0000	9883640.0
23	Minnesota	1034	18	3.0000	5303925.0
24	Mississippi	1034	18	3.0000	2967297.0
25	Missouri	1034	18	3.0000	5988927.0
26	Montana	1034	18	3.0000	989415.0
27	Nebraska	1034	18	3.0000	1826341.0
28	Nevada	1034	18	3.0000	2700551.0
29	New Hampshire	1034	18	3.0000	1316470.0
30	New Jersey	1034	18	3.0000	8791894.0
31	New Mexico	1034	18	3.0000	2059179.0
32	New York	1034	18	3.0000	19378102.0
33	North Carolina	1034	18	3.0000	10439388.0
34	North Dakota	1034	18	3.0000	672591.0
35	Ohio	1034	18	3.0000	11689100.0
36	Oklahoma	1034	18	3.0000	3956971.0
37	Oregon	1034	18	3.0000	4217737.0
38	Pennsylvania	1034	18	3.0000	12801989.0
39	Rhode Island	1034	18	3.0000	1059361.0
40	South Carolina	1034	18	3.0000	5148714.0
41	South Dakota	1034	18	3.0000	884659.0
42	Tennessee	1034	18	3.0000	6829174.0
43	Texas	1034	18	3.0000	28995881.0
44	Utah	1034	18	3.0000	3205958.0
45	Vermont	1034	18	3.0000	623989.0
46	Virginia	1034	18	3.0000	8535519.0
47	Washington	1034	18	3.0000	7614893.0
48	West Virginia	1034	18	3.0000	1792147.0
49	Wisconsin	1034	18	3.0000	5822434.0
50	Wyoming	1034	18	3.0000	578759.0

Visualization

```
plt.rcParams['figure.figure.dpi'] = 75
```

```
# line_cnt[['COVID-19', 'Influenza']].\
# line_cnt[line_cnt.index.isin(['2020-01', '2020-02', '2020-03', '2020-04', '2020-05', '2020-06', '2020-07', '2020-08', '2020-09', '2020-10', '2020-11', '2020-12'])].\
# ax.set_ylabel('Cases')
# ax.set_xlabel('Month')
# ax.set_title('COVID-19 and Influenza Cases, 2020')
# ax.legend(loc=0);
```

```
line_cnt[['COVID-19', 'Influenza']].\
ax.set_ylabel('Cases')
ax.set_xlabel('Month')
ax.set_title('COVID-19 and Influenza Cases, 2020')
ax.legend(loc=0);
```

PLEASE NOTE: The above graph was obtained with usage of LIVE data, what you see now might be different than what was shown at the time of taking.

```
line_cnt[line_cnt[line_cnt.COVID-19 > 0].\
line_cnt[line_cnt.index.isin(['2020-01', '2020-02', '2020-03', '2020-04', '2020-05', '2020-06', '2020-07', '2020-08', '2020-09', '2020-10', '2020-11', '2020-12'])].\
line_cnt[line_cnt.COVID-19 > 0].\
```

```
OLS Regression Results
==============================================================================
Dep. Variable:                COVID-19   R-squared:                       0.000
Model:                            OLS   Adjusted R-squared:             -0.020
Method:                 Least Squares   F-statistic:                     0.0000
Date:                Wed, 02 Jun 2021   Prob (F-statistic):              0.9970
Time:                        00:00:00   Log-Likelihood:                -1.0000e+06
No. Observations:                 1034   AIC:                             2.000e+06
Df Residuals:                     1032   BIC:                             2.000e+06
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.0000      0.000       0.000      1.000      -0.000       0.000
Influenza      0.0000      0.000       0.000      0.997      -0.000       0.000
==============================================================================
Omnibus:                     1034.000   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7.000e+03
Skew:                           0.000   Prob(Omnibus):                   0.000
Kurtosis:                       7.000   Cond. No.                         1.000
==============================================================================
```

```
line_cnt[line_cnt[line_cnt.COVID-19 > 0].\
line_cnt[line_cnt.index.isin(['2020-01', '2020-02', '2020-03', '2020-04', '2020-05', '2020-06', '2020-07', '2020-08', '2020-09', '2020-10', '2020-11', '2020-12'])].\
line_cnt[line_cnt.COVID-19 > 0].\
```

```
OLS Regression Results
==============================================================================
Dep. Variable:                COVID-19   R-squared:                       0.000
Model:                            OLS   Adjusted R-squared:             -0.020
Method:                 Least Squares   F-statistic:                     0.0000
Date:                Wed, 02 Jun 2021   Prob (F-statistic):              0.9970
Time:                        00:00:00   Log-Likelihood:                -1.0000e+06
No. Observations:                 1034   AIC:                             2.000e+06
Df Residuals:                     1032   BIC:                             2.000e+06
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.0000      0.000       0.000      1.000      -0.000       0.000
Influenza      0.0000      0.000       0.000      0.997      -0.000       0.000
==============================================================================
Omnibus:                     1034.000   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7.000e+03
Skew:                           0.000   Prob(Omnibus):                   0.000
Kurtosis:                       7.000   Cond. No.                         1.000
==============================================================================
```

Looking at the graph first, it is clear that while population nearly explains all of the rise in deaths attributed to COVID-19, it fails to do so for influenza deaths (the explanation power of the influenza model is the flat line at the lower zero-ish). What this means is that people do not take more precautions with a, and barely accept that it will no longer be a pandemic, but still, they do not take more precautions with influenza, and yet, the number of deaths from it does not rise with population. Perhaps the precautions with COVID-19 are very effective, but deaths from it do not rise with population because it is a new disease, and people are dying from it, but it is not yet spreading to the point where it rivals, or crucially, has passed by the virus. This explains why countries like India and Indonesia have relatively low deaths from COVID-19 despite their large populations.

Using the same model, but this time with Cases from CDC's Influenza deaths, is overall logical but for the population of the city, along with the other factors we believe. Key-to-get on a reason as to believe this is due to COVID-19 is absent in influenza.

3. Unintentional Injuries

3.1. Death Rate for Accidental Deaths

First, using the file "major_causes_rates.csv", visualize the trend for age adjusted death rate for accidental deaths (injury).

Second, using the file "state_rates.csv", visualize the age adjusted death rates for unintentional injuries and suicide.

```
cause_ind_cnt
cause_ind_cnt.shape
```

```
(1,1)
```

	Year	Cause	Age Adjusted Death Rate
0	1981	Heart Disease	345.5
1	1982	Heart Disease	335.1
2	1983	Heart Disease	327.2
3	1984	Heart Disease	321.5
4	1985	Heart Disease	315.1
5	1986	Heart Disease	307.1
6	1987	Heart Disease	299.1
7	1988	Heart Disease	291.1
8	1989	Heart Disease	283.1
9	1990	Heart Disease	275.1
10	1991	Heart Disease	267.1
11	1992	Heart Disease	259.1
12	1993	Heart Disease	251.1
13	1994	Heart Disease	243.1
14	1995	Heart Disease	235.1
15	1996	Heart Disease	227.1
16	1997	Heart Disease	219.1
17	1998	Heart Disease	211.1
18	1999	Heart Disease	203.1
19	2000	Heart Disease	195.1
20	2001	Heart Disease	187.1
21	2002	Heart Disease	179.1
22	2003	Heart Disease	171.1
23	2004	Heart Disease	163.1
24	2005	Heart Disease	155.1
25	2006	Heart Disease	147.1
26	2007	Heart Disease	139.1
27	2008	Heart Disease	131.1
28	2009	Heart Disease	123.1
29	2010	Heart Disease	115.1

```
cause_ind_cnt[cause_ind_cnt.Cause == 'Heart Disease']
cause_ind_cnt[cause_ind_cnt.Cause == 'Heart Disease'].\
ax = plt.gca()
ax.set_ylabel('Age Adjusted Death Rate')
ax.set_xlabel('Year')
ax.set_title('Heart Disease Age Adjusted Death Rate, 1981-2010')
```

```
state_rates.shape
```

```
(1,1)
```

```
state_rates[[ 'Cause', 'Age Adjusted Death Rate']].\
state_rates[state_rates.Cause == 'Suicide']
state_rates[state_rates.Cause == 'Suicide'].\
ax = plt.gca()
ax.set_ylabel('Age Adjusted Death Rate')
ax.set_xlabel('Year')
ax.set_title('Suicide Age Adjusted Death Rate, 1999-2010')
```

	Cause	Age Adjusted Death Rate
1	1999	Suicide 10.9
2	2000	Suicide 10.4
3	2001	Suicide 10.4
4	2002	Suicide 10.8
5	2003	Suicide 10.8
6	2004	Suicide 11.0
7	2005	Suicide 11.1
8	2006	Suicide 11.3
9	2007	Suicide 11.3
10	2008	Suicide 11.4
11	2009	Suicide 11.3
12	2010	Suicide 11.3

As we can see from the data, the death rate for accidents has decreased significantly since the beginning of the 1980s while, however, there has been a period of increase since 2000.

3. Breakdown of Unintentional Injury Mechanisms

Now, let us explore the national unintentional injuries leading to death in the US.

Through the file "unintentional_injuries_deaths_leading_causes.csv", explore the data.

```
injury_cnt.shape
injury_cnt
```

```
(1,1)
```

	Year	Cause of Injury	Deaths	Population	Age-Adjusted Death Rate	95% Confidence Interval - Lower	95% Confidence Interval - Upper
0	2016	Falls	All Ages	All Genders	63.4	62.8	64.0
1	2016	Falls	All Ages	Male	85.4	84.4	86.4
2	2016	Falls	All Ages	Female	44.1	43.6	44.6
3	2016	Falls	85 Years and Older	All Genders	1,000	1,000.0	1,000.0
4	2016	Falls	85 Years and Older	Male	1,000	1,000.0	1,000.0
5	2016	Falls	85 Years and Older	Female	1,000	1,000.0	1,000.0
6	2016	Falls	75 - 84 Years	All Genders	1,000	1,000.0	1,000.0
7	2016	Falls	75 - 84 Years	Male	1,000	1,000.0	1,000.0
8	2016	Falls	75 - 84 Years	Female	1,000	1,000.0	1,000.0
9	2016	Falls	65 - 74 Years	All Genders	1,000	1,000.0	1,000.0
10	2016	Falls	65 - 74 Years	Male	1,000	1,000.0	1,000.0
11	2016	Falls	65 - 74 Years	Female	1,000	1,000.0	1,000.0
12	2016	Falls	55 - 64 Years	All Genders	1,000	1,000.0	1,000.0
13	2016	Falls	55 - 64 Years	Male	1,000	1,000.0	1,000.0
14	2016	Falls	55 - 64 Years	Female	1,000	1,000.0	1,000.0
15	2016	Falls	45 - 54 Years	All Genders	1,000	1,000.0	1,000.0
16	2016	Falls	45 - 54 Years	Male	1,000	1,000.0	1,000.0
17	2016	Falls	45 - 54 Years	Female	1,000	1,000.0	1,000.0
18	2016	Falls	35 - 44 Years	All Genders	1,000	1,000.0	1,000.0
19	2016	Falls	35 - 44 Years	Male	1,000	1,000.0	1,000.0
20	2016	Falls	35 - 44 Years	Female	1,000	1,000.0	1,000.0
21	2016	Falls	25 - 34 Years	All Genders	1,000	1,000.0	1,000.0
22	2016	Falls	25 - 34 Years	Male	1,000	1,000.0	1,000.0
23	2016	Falls	25 - 34 Years	Female	1,000	1,000.0	1,000.0
24	2016	Falls	15 - 24 Years	All Genders	1,000	1,000.0	1,000.0
25	2016	Falls	15 - 24 Years	Male	1,000	1,000.0	1,000.0
26	2016	Falls	15 - 24 Years	Female	1,000	1,000.0	1,000.0
27	2016	Falls	1 - 14 Years	All Genders	1,000	1,000.0	1,000.0
28	2016	Falls	1 - 14 Years	Male	1,000	1,000.0	1,000.0
29	2016	Falls	1 - 14 Years	Female	1,000	1,000.0	1,000.0

```
injury_cnt[injury_cnt.Cause == 'Falls']
injury_cnt[injury_cnt.Cause == 'Falls'].\
ax = plt.gca()
ax.set_ylabel('Age Adjusted Death Rate')
ax.set_xlabel('Year')
ax.set_title('Falls Age Adjusted Death Rate, 2016')
```

```
injury_cnt.Year
injury_cnt.Cause of Injury
injury_cnt.Deaths
injury_cnt.Population
injury_cnt.Age-Adjusted Death Rate
injury_cnt.95% Confidence Interval - Lower
injury_cnt.95% Confidence Interval - Upper
```

```
0    2016
1    2016
2    2016
3    2016
4    2016
5    2016
6    2016
7    2016
8    2016
9    2016
10   2016
11   2016
12   2016
13   2016
14   2016
15   2016
16   2016
17   2016
18   2016
19   2016
20   2016
21   2016
22   2016
23   2016
24   2016
25   2016
26   2016
27   2016
28   2016
29   2016
Name: Year, dtype: int64
0                Falls
1                Falls
2                Falls
3                Falls
4                Falls
5                Falls
6                Falls
7                Falls
8                Falls
9                Falls
10               Falls
11               Falls
12               Falls
13               Falls
14               Falls
15               Falls
16               Falls
17               Falls
18               Falls
19               Falls
20               Falls
21               Falls
22               Falls
23               Falls
24               Falls
25               Falls
26               Falls
27               Falls
28               Falls
29               Falls
Name: Cause of Injury, dtype: object
0    1,000
1    1,000
2    1,000
3    1,000
4    1,000
5    1,000
6    1,000
7    1,000
8    1,000
9    1,000
10   1,000
11   1,000
12   1,000
13   1,000
14   1,000
15   1,000
16   1,000
17   1,000
18   1,000
19   1,000
20   1,000
21   1,000
22   1,000
23   1,000
24   1,000
25   1,000
26   1,000
27   1,000
28   1,000
29   1,000
Name: Deaths, dtype: int64
0    1,000
1    1,000
2    1,000
3    1,000
4    1,000
5    1,000
6    1,000
7    1,000
8    1,000
9    1,000
10   1,000
11   1,000
12   1,000
13   1,000
14   1,000
15   1,000
16   1,000
17   1,000
18   1,000
19   1,000
20   1,000
21   1,000
22   1,000
23   1,000
24   1,000
25   1,000
26   1,000
27   1,000
28   1,000
29   1,000
Name: Population, dtype: int64
0    63.4
1    85.4
2    44.1
3    1,000
4    1,000
5    1,000
6    1,000
7    1,000
8    1,000
9    1,000
10   1,000
11   1,000
12   1,000
13   1,000
14   1,000
15   1,000
16   1,000
17   1,000
18   1,000
19   1,000
20   1,000
21   1,000
22   1,000
23   1,000
24   1,000
25   1,000
26   1,000
27   1,000
28   1,000
29   1,000
Name: Age-Adjusted Death Rate, dtype: float64
0    62.8
1    84.4
2    43.6
3    1,000.0
4    1,000.0
5    1,000.0
6    1,000.0
7    1,000.0
8    1,000.0
9    1,000.0
10   1,000.0
11   1,000.0
12   1,000.0
13   1,000.0
14   1,000.0
15   1,000.0
16   1,000.0
17   1,000.0
18   1,000.0
19   1,000.0
20   1,000.0
21   1,000.0
22   1,000.0
23   1,000.0
24   1,000.0
25   1,000.0
26   1,000.0
27   1,000.0
28   1,000.0
29   1,000.0
Name: 95% Confidence Interval - Lower, dtype: float64
0    64.0
1    86.4
2    44.6
3    1,000.0
4    1,000.0
5    1,000.0
6    1,000.0
7    1,000.0
8    1,000.0
9    1,000.0
10   1,000.0
11   1,000.0
12   1,000.0
13   1,000.0
14   1,000.0
15   1,000.0
16   1,000.0
17   1,000.0
18   1,000.0
19   1,000.0
20   1,000.0
21   1,000.0
22   1,000.0
23   1,000.0
24   1,000.0
25   1,000.0
26   1,000.0
27   1,000.0
28   1,000.0
29   1,000.0
Name: 95% Confidence Interval - Upper, dtype: float64
```

All Mechanisms

Select a graph displaying the total accidental deaths each year for both women, all ages, all races, and for all injury mechanisms.

```
# injury_cnt[injury_cnt.Year == 2016]
# injury_cnt[injury_cnt.Year == 2016].\
# injury_cnt[injury_cnt.Year == 2016].\
```

```
injury_cnt[injury_cnt.Year == 2016]
injury_cnt[injury_cnt.Year == 2016].\
injury_cnt[injury_cnt.Year == 2016].\
ax = plt.gca()
ax.set_ylabel('Deaths')
ax.set_xlabel('Cause of Injury')
ax.set_title('Deaths, 2016')
```

```
# injury_cnt[injury_cnt.Year == 2016].\
# injury_cnt[injury_cnt.Year == 2016].\
# injury_cnt[injury_cnt.Year == 2016].\
```

```
> ny_sex_race_eye_death [1:5,]
  ny_sex_race_eye_death Freq
1                 1 1 1 1 1 1
2                 2 1 1 1 1 2
3                 3 1 1 1 2 1
4                 4 1 1 1 2 2
5                 5 1 1 2 1 1
> plot.ts(ny_sex_race_eye_death[,"Freq"])
> plot.ts(ny_sex_race_eye_death[,"Freq"],xlab="Index",
+ ylab="Death count",main="Deaths")
> ny_sex_race_eye_death[,"Freq"]
  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
> ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"])
  [1] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
  [8] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 [15] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 [22] 0.0000000 0.0000000 0.0000000
> plot.ts(ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"]),
+ xlab="Index",ylab="Death count",main="Deaths")
> plot.ts(ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"]),
+ xlab="Index",ylab="Death count",main="Deaths",lwd=2)
> plot.ts(ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"]),
+ xlab="Index",ylab="Death count",main="Deaths",lwd=2,col="red")
> plot.ts(ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"]),
+ xlab="Index",ylab="Death count",main="Deaths",lwd=2,col="red",type="l")
> plot.ts(ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"]),
+ xlab="Index",ylab="Death count",main="Deaths",lwd=2,col="red",type="l",
+ lwd=2)
> plot.ts(ny_sex_race_eye_death[,"Freq"]/sum(ny_sex_race_eye_death[,"Freq"]),
+ xlab="Index",ylab="Death count",main="Deaths",lwd=2,col="red",type="l",
+ lwd=2,lty="dashed")
```

Deaths
Death count
0.0000000
0.0000000
0.0000000

```
> a_acc[ ].describe()
count    1.000000e+0
mean     1.000000e+0
std      1.000000e+0
min      1.000000e+0
25%      1.000000e+0
50%      1.000000e+0
75%      1.000000e+0
max      1.000000e+0
dtype: float64
> sns.set(rc={'figure.figsize': (11, 7)})
> sns.boxplot(a_acc[ ])
> sns.swarmplot(a_acc[ ],
                color='black',
                alpha=0.5)
> sns.despine()
```

```
> a_acc[a_acc[ ].isnull()]
Series([], Name: , dtype: int64)
> a_acc[ ].dropna(inplace=True)
> a_acc[ ].describe()
count    1.000000e+0
mean     1.000000e+0
std      1.000000e+0
min      1.000000e+0
25%      1.000000e+0
50%      1.000000e+0
75%      1.000000e+0
max      1.000000e+0
dtype: float64
> sns.set(rc={'figure.figsize': (11, 7)})
> sns.boxplot(a_acc[ ])
> sns.swarmplot(a_acc[ ],
                color='black',
                alpha=0.5)
> sns.despine()
```

```
> a_acc[ ].sort_values(ascending=False).head()
> a_acc[ ].sort_values(ascending=False).tail()
> a_acc[ ].sum()
> a_acc[ ].count()
> a_acc[ ].mean()
```

Graph with annual death rate:

```
> a_acc[ ].sum()
> a_acc[ ].count()/a_acc[ ].sum()
> a_acc[ ].mean()
> a_acc[ ].sum()
> a_acc[ ].count()
> a_acc[ ].mean()
> sns.set(rc={'figure.figsize': (11, 7)})
> sns.boxplot(a_acc[ ])
> sns.swarmplot(a_acc[ ],
                color='black',
                alpha=0.5)
> sns.despine()
> sns.set(rc={'figure.figsize': (11, 7)})
> sns.boxplot(a_acc[ ])
> sns.swarmplot(a_acc[ ],
                color='black',
                alpha=0.5)
> sns.despine()
```

```
> sns.set(rc={'figure.figsize': (11, 7)})
> sns.boxplot(a_acc[ ])
> sns.swarmplot(a_acc[ ],
                color='black',
                alpha=0.5)
> sns.despine()
```

```
> a_acc[ ].sum()
> a_acc[ ].count()/a_acc[ ].sum()
> a_acc[ ].mean()
> a_acc[ ].sum()
> a_acc[ ].count()
> a_acc[ ].mean()
```

```
# ny_hp_deaths[['Year', 'Cause', 'Deaths']].head()
ny_hp_deaths[['Year', 'Cause', 'Deaths']].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2010].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2011].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2012].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2013].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2014].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2015].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2016].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2017].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2018].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2019].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2020].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2021].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2022].shape
# ny_hp_deaths[ny_hp_deaths.Year == 2023].shape
```

```
Cause(s) selected:['Heart Disease Ischemic', 'Heart Disease Hypertensive', 'Heart Disease Other', 'Heart Disease Rheumatic', 'Heart Disease Pulmonary', 'Heart Disease Cardiomyopathy', 'Heart Disease Congenital', 'Heart Disease Myocarditis', 'Heart Disease Endocarditis', 'Heart Disease Other Specified', 'Heart Disease Unspecified', 'Heart Disease Other Unclassified', 'Hypertension', 'Hypertensive Heart Disease', 'Hypertensive Heart Disease With Heart Failure', 'Hypertensive Heart Disease Without Heart Failure', 'Hypertensive Heart Disease With Renal Failure', 'Hypertensive Heart Disease With Renal Failure Without Heart Failure', 'Hypertensive Heart Disease With Heart Failure With Renal Failure', 'Hypertensive Heart Disease With Heart Failure Without Renal Failure', 'Hypertensive Heart Disease With Heart Failure With Renal Failure Without Heart Failure', 'Hypertensive Heart Disease Unspecified', 'Hypertensive Heart Disease Other Unclassified', 'Hypertensive Heart Disease Other Specified', 'Hypertensive Heart Disease Other Unclassified', 'Hypertensive Heart Disease Unspec
```

```
fig,ax = plt.subplots(1,1,figsize=(12,6))
sns.boxplot(x="Year", y="Death_rate", data=cancer_death_rate[cancer_death_rate.Geographic == 'United States'], palette="Set3")
sns.swarmplot(x="Year", y="Death_rate", data=cancer_death_rate[cancer_death_rate.Geographic == 'United States'], color="black")
ax.set_title("Death rate in United States by year")
ax.set_xlabel("Year")
ax.set_ylabel("Death rate")
plt.show()
```

```
df_states = cancer_death_rate[cancer_death_rate.Geographic != 'United States']
df_states.sort_values('Death_rate',inplace=True,ascending=False)
fig,ax = plt.subplots(1,1,figsize=(12,6))
sns.boxplot(x="Geographic", y="Death_rate", data=df_states, palette="Set3")
ax.set_title("Death rate in states by year")
ax.set_xlabel("Geographic")
ax.set_ylabel("Death rate")
plt.show()
```

	Geographic	Year	Death_rate	Cancer_death_rate
1	Alabama	1999	214.7	Malignant Neoplasms
2	Alaska	1999	143.1	Malignant Neoplasms
3	Arizona	1999	171.2	Malignant Neoplasms
4	Arkansas	1999	196.9	Malignant Neoplasms
5	California	1999	144.8	Malignant Neoplasms
6	Colorado	1999	139.2	Malignant Neoplasms
7	Connecticut	1999	189.7	Malignant Neoplasms
8	Delaware	1999	202.1	Malignant Neoplasms
9	District of Columbia	1999	244.8	Malignant Neoplasms
10	Florida	1999	178.9	Malignant Neoplasms
11	Georgia	1999	168.9	Malignant Neoplasms
12	Hawaii	1999	126.1	Malignant Neoplasms
13	Idaho	1999	159.1	Malignant Neoplasms
14	Illinois	1999	179.3	Malignant Neoplasms
15	Indiana	1999	191.1	Malignant Neoplasms
16	Iowa	1999	179.9	Malignant Neoplasms
17	Kansas	1999	179.1	Malignant Neoplasms
18	Kentucky	1999	204.4	Malignant Neoplasms
19	Louisiana	1999	191.1	Malignant Neoplasms
20	Maine	1999	168.6	Malignant Neoplasms
21	Maryland	1999	179.9	Malignant Neoplasms
22	Massachusetts	1999	174.1	Malignant Neoplasms
23	Michigan	1999	187.1	Malignant Neoplasms
24	Minnesota	1999	167.1	Malignant Neoplasms
25	Mississippi	1999	203.1	Malignant Neoplasms
26	Missouri	1999	189.1	Malignant Neoplasms
27	Montana	1999	171.1	Malignant Neoplasms
28	Nebraska	1999	171.1	Malignant Neoplasms
29	Nevada	1999	171.1	Malignant Neoplasms
30	New Hampshire	1999	171.1	Malignant Neoplasms
31	New Jersey	1999	171.1	Malignant Neoplasms
32	New Mexico	1999	171.1	Malignant Neoplasms
33	New York	1999	171.1	Malignant Neoplasms
34	North Carolina	1999	171.1	Malignant Neoplasms
35	North Dakota	1999	171.1	Malignant Neoplasms
36	Ohio	1999	171.1	Malignant Neoplasms
37	Oklahoma	1999	171.1	Malignant Neoplasms
38	Oregon	1999	171.1	Malignant Neoplasms
39	Pennsylvania	1999	171.1	Malignant Neoplasms
40	Rhode Island	1999	171.1	Malignant Neoplasms
41	South Carolina	1999	171.1	Malignant Neoplasms
42	South Dakota	1999	171.1	Malignant Neoplasms
43	Tennessee	1999	171.1	Malignant Neoplasms
44	Texas	1999	171.1	Malignant Neoplasms
45	Utah	1999	171.1	Malignant Neoplasms
46	Vermont	1999	171.1	Malignant Neoplasms
47	Virginia	1999	171.1	Malignant Neoplasms
48	Washington	1999	171.1	Malignant Neoplasms
49	West Virginia	1999	171.1	Malignant Neoplasms
50	Wisconsin	1999	171.1	Malignant Neoplasms
51	Wyoming	1999	171.1	Malignant Neoplasms
52	Puerto Rico	1999	171.1	Malignant Neoplasms
53	Virgin Islands	1999	171.1	Malignant Neoplasms
54	Guam	1999	171.1	Malignant Neoplasms
55	United States	1999	171.1	Malignant Neoplasms

```
whole_data = df_states[df_states.Year == 1999]
whole_data.sort_values('Death_rate',inplace=True,ascending=False)
fig,ax = plt.subplots(1,1,figsize=(12,6))
sns.barplot(x="Geographic", y="Death_rate", data=whole_data, palette="Set3")
ax.set_title("Death rate in states in 1999")
ax.set_xlabel("Geographic")
ax.set_ylabel("Death rate")
plt.show()
```

```
fig,ax = plt.subplots(1,1,figsize=(12,6))
sns.barplot(x="Geographic", y="Death_rate", data=cancer_death_rate[cancer_death_rate.Cancer_death_rate == 'Malignant Neoplasms'], palette="Set3")
ax.set_title("Death rate in states by year")
ax.set_xlabel("Geographic")
ax.set_ylabel("Death rate")
plt.show()
```

	Cancer_death_rate	Year	Death_rate	Geographic
0	Malignant Neoplasms	2014	Alabama	168.4
1	Malignant Neoplasms	2014	Alaska	122.6
2	Malignant Neoplasms	2014	Arizona	137.1
3	Malignant Neoplasms	2014	Arkansas	187.2
4	Malignant Neoplasms	2014	California	121.1
5	Malignant Neoplasms	2014	Colorado	122.7
6	Malignant Neoplasms	2014	Connecticut	161.8
7	Malignant Neoplasms	2014	Delaware	187.2
8	Malignant Neoplasms	2014	District of Columbia	187.2
9	Malignant Neoplasms	2014	Florida	151.1
10	Malignant Neoplasms	2014	Georgia	161.8
11	Malignant Neoplasms	2014	Hawaii	122.7
12	Malignant Neoplasms	2014	Idaho	151.1
13	Malignant Neoplasms	2014	Illinois	161.8
14	Malignant Neoplasms	2014	Indiana	187.2
15	Malignant Neoplasms	2014	Iowa	161.8
16	Malignant Neoplasms	2014	Kansas	161.8
17	Malignant Neoplasms	2014	Kentucky	187.2
18	Malignant Neoplasms	2014	Louisiana	187.2
19	Malignant Neoplasms	2014	Maine	161.8
20	Malignant Neoplasms	2014	Maryland	161.8
21	Malignant Neoplasms	2014	Massachusetts	161.8
22	Malignant Neoplasms	2014	Michigan	161.8
23	Malignant Neoplasms	2014	Minnesota	161.8
24	Malignant Neoplasms	2014	Mississippi	161.8
25	Malignant Neoplasms	2014	Missouri	161.8
26	Malignant Neoplasms	2014	Montana	161.8
27	Malignant Neoplasms	2014	Nebraska	161.8
28	Malignant Neoplasms	2014	Nevada	161.8
29	Malignant Neoplasms	2014	New Hampshire	161.8
30	Malignant Neoplasms	2014	New Jersey	161.8
31	Malignant Neoplasms	2014	New Mexico	161.8
32	Malignant Neoplasms	2014	New York	161.8
33	Malignant Neoplasms	2014	North Carolina	161.8
34	Malignant Neoplasms	2014	North Dakota	161.8
35	Malignant Neoplasms	2014	Ohio	161.8
36	Malignant Neoplasms	2014	Oklahoma	161.8
37	Malignant Neoplasms	2014	Oregon	161.8
38	Malignant Neoplasms	2014	Pennsylvania	161.8
39	Malignant Neoplasms	2014	Rhode Island	161.8
40	Malignant Neoplasms	2014	South Carolina	161.8
41	Malignant Neoplasms	2014	South Dakota	161.8
42	Malignant Neoplasms	2014	Tennessee	161.8
43	Malignant Neoplasms	2014	Texas	161.8
44	Malignant Neoplasms	2014	Utah	161.8
45	Malignant Neoplasms	2014	Vermont	161.8
46	Malignant Neoplasms	2014	Virginia	161.8
47	Malignant Neoplasms	2014	Washington	161.8
48	Malignant Neoplasms	2014	West Virginia	161.8
49	Malignant Neoplasms	2014	Wisconsin	161.8
50	Malignant Neoplasms	2014	Wyoming	161.8
51	Malignant Neoplasms	2014	Puerto Rico	161.8
52	Malignant Neoplasms	2014	Virgin Islands	161.8
53	Malignant Neoplasms	2014	Guam	161.8
54	Malignant Neoplasms	2014	United States	161.8

```
cancer_death_rate[cancer_death_rate.Year == 2014].sort_values('Death_rate',ascending=False)
```

```
fig,ax = plt.subplots(1,1,figsize=(12,6))
sns.barplot(x="Geographic", y="Death_rate", data=cancer_death_rate[cancer_death_rate.Year == 2014], palette="Set3")
ax.set_title("Death rate in states in 2014")
ax.set_xlabel("Geographic")
ax.set_ylabel("Death rate")
plt.show()
```

```
fig,ax = plt.subplots(1,1,figsize=(12,6))
sns.boxplot(x="Cancer_death_rate", y="Death_rate", data=cancer_death_rate[cancer_death_rate.Geographic == 'United States'], palette="Set3")
sns.swarmplot(x="Cancer_death_rate", y="Death_rate", data=cancer_death_rate[cancer_death_rate.Geographic == 'United States'], color="black")
ax.set_title("Death rate in United States by cancer")
ax.set_xlabel("Cancer_death_rate")
ax.set_ylabel("Death rate")
plt.show()
```


Death rate in United States by cancer
Death rate
Cancer_death_rate
Malignant Neoplasms
Cardiovascular Disease
Heart Disease
Stroke
Chronic Lower Respiratory Disease
Influenza and Pneumonia
Diabetes Mellitus
Alzheimer's Disease
Nephritis, Nephrosis, Nephrosis
Accidents (Unintentional Injuries)
Suicide
Homicide
Septicemia
Chronic Liver Disease
Cirrhosis
Hypertension
Pneumonitis
Pneumonitis due to Solids and Liquids
Perinatal Period
Congenital Anomalies
Sudden Infant Death Syndrome
Pneumonitis due to Solids and Liquids
Other Infectious Diseases
Other Chronic Lower Respiratory Diseases
Other Diseases of Heart
Other Diseases of Liver
Other Diseases of Respiratory System
Other Diseases of Circulatory System
Other Diseases of Digestive System
Other Diseases of Genitourinary System
Other Diseases of Nervous System
Other Diseases of Blood
Other Diseases of Endocrine Glands
Other Diseases of Skin
Other Diseases of Musculoskeletal System
Other Diseases of Infancy
Other Diseases of Childhood
Other Diseases of Adulthood
Other Diseases of Pregnancy