Text

Description automatically generated with medium confidence

**John O’Brien**

Email:

[John.obrien@henkel.com](mailto:John.obrien@henkel.com)

[Jobrien2.irl@gmail.com](mailto:Jobrien2.irl@gmail.com)

Project Report GitHub URL (insert URL here)

<https://github.com/jobrien2/UCD-project.git>

**Abstract (Short overview of the entire project and features)**

The project looks at the mortality statistics from 1980 to 2017

The project combines datasets on Population, death rate (grouped by age, sex and cause) and some economic statistics.

The project will create a function to allow searchable works in the cause of death category and then subset data to analyse the output to gain insights.

**Introduction (Explain why you chose this project use case)**

Over the past 18 months of a Pandemic, there is a lot of conversation about the efficacy of vaccinations, the risk of suicide due to isolation, deaths due to causes other than Covid that would be assigned an official cause of Death as Covid. The conversations are unavoidable, and I wanted to use the dataset to answer objectively some of the following questions:

What is the trend for deaths in the population up to 2020 and overall is there an increase or otherwise in the death rate?

What has the impact of vaccinations been in general for non-Covid related virus e.g. meningitis?

What is the death rate for suicide? When 2020 data and 2021 data is released on mortality causes, it will help to inform this topic.

Are there any macro-economic indicators for suicide?

**Dataset (Provide a description of your dataset and source. Also justify why you chose this source)**

The project uses dataset mainly from the CSO website <https://www.cso.ie/en/index.html> and additionally a table provided by Wikipedia.

Three datasets are used from CSO

1. Mortality Data – the Data set ‘Age-Specific Mortality 1980 - 2020’ provides the number of deaths by specific cause grouped into age categories by year. The dataset provides data from 1980 to 2017 inclusive. The dataset was downloaded as a CSV file and imported into python and converted to a data frame.

Table

Description automatically generated with medium confidence

1. Ireland’s population – The data set ‘Population Estimates (Persons in April)’ provides the population by age group by sex from the years 1950 to 2020. The dataset fits with the mortality rate insofar as both datasets have comparative age groups to link.

Table

Description automatically generated

1. Date rate by quarter – The data set ‘Deaths Registered Provisional’ includes death numbers for 2018 to 2020 inclusive. There is no cause of death but the data allows us to see any changes to death rate year on year.

Additional data was scraped from Wikipedia, although Wikipedia is not normally a ‘reliable’ source of data, the website was used to practice web scraping.

**Implementation Process (Describe your entire process in detail)**

Before brining in datasets, the appropriate packages were imported to the shell. As the project developed, additional packages were imported and scripted in the same cell.

**Step one – Import data:**

The CSV files were imported with pandas function ‘pd.read\_csv( )’ to convert the CSV directly into a dataframe

BeautifulSoup package was used to scrape Wikipedia to convert a table on the website to a dataframe. Using ‘requests.get(wikiurl)’, the website was checked to ensure the data legally allowed to scrape.

**Step two - Clean the data:**

The first thing to do is to check the Dataframe characteristics, number of columns, data type per column, number of rows etc. by using .info() function for the concise summary. what I want to understand here is if the data is numerical or a text string or other and that the data type is capable of manipulation in the desired ways e.g. numbers are float or int and not stored as string.

The next thing was to check for missing data using .isna() and summing the NaN values to get an idea of the volume of NaN values in the dataframe.

I wanted to preserve as much data as possible in the datasets and to practice different techniques of replacing NaN data.

In the mortality dataset there was only one NaN value out of 7 \* 342,228 rows of data.

I filtered the data by firstly replacing the NaN value with a text ‘check’ using .fillna(‘check’, inplace=True) then using a .query for ‘check’ in the appropriate column. This is a more convoluted way over just filtering NaN but it was done for practice. After filtering the data, the data looked to be overlapped in the columns.

I was able to check the rest of the data and identify the correct values and replace the data in the correct columns and row match by using .at with the index position.

After removing the NaN values, I confirmed the data type of the numerical data to be used for calculations. The number values were of type object. Where there was a record with a 0 value a ‘-‘ was reported. I replaced the ‘-‘ with a 0 using .replace and then converting the column data type to a float type using .map and assigned the float type back through the column i.e. df[‘col’] = df[‘col’].map(float).

I then sliced the data further to have a dataset that could be now manipulated with the desired information.

When cleaning the population data there was more data missing for one of the age groups, 114 data points missing. Here, rather than dropping the data I compared the data for the same age group of the missing value with a similar age group

Data missing - Age group 0 to 4, for years 1980 to 1995 inclusive

Data available - 1. Age group 0 to 4, for years 1996 to 2017 inclusive

2. Age group 5 to 9, for years 1996 to 2017 inclusive

3. Age group 5 to 9, for years 1996 to 2017 inclusive

Objective was to check for correlation of two age groups for the same years, if strongly correlated, run a regression analysis and use the regression model equation to reverse predict the population of the missing data points.

To replace the missing values with the newly created regression equation, I would use iterrows to:

1. Replace the Nan with 'check'
2. If the row value is check, add to a list the index of the check location
3. Identify the year and Sex of the missing value
4. Subset the data to identify the corresponding year and sex of the 5- to 9-year-old of the missing value of 0 to 4 year old
5. Use the regression model equation with the 5- to 9-year-old age and append to a new age list
6. Create a list of the range of the length of the new age list to use as index values
7. Loop through the lists to replace the NaN values in the dataframe with the new age

As iterating through the dataframe is essentially iterating through a copy of the dataframe, I could not replace the NaN values with the new calculated values in place. For this reason, I created lists and appended the values with the corresponding index location in an index list. A more practical way to complete this task would be to use a logical equation:- if age is NaN: complete the regression calculation and add value to a new defined column in the df; if age is not NaN add the value to the newly defined column. This would avoid appending to lists and the iterating through the lists to replace values in the data frame.

The death data was recorded by quarter, a similar process for checking NaN data was undertaking as the previous datasets. The two other activities required for this data set were:

1. Create a year value by taking the data in the Quarter column and slicing the first four digits of the string .str[:4] and saving in a new year column.

Table

Description automatically generated

1. After slicing the data using .loc method, (.loc here used to group multiple row attributes), the values for the qtrs. were grouped by year to get a yearly value as opposed to a quarterly value.

Finally, for the Economic data I backfilled the values of missing data Unemployment rates with .fillna(method = 'bfill')

**Step Three: Join Data**

I mostly joined the data sets with pd.join ensuring I matched on keys for both dataset. Left join was suitable for all events.

Pd.concat was used to concatenate onto the end of a datframe with a second dataframe.

**Step Four: create functions**

I made two main functions

Firstly, I created a function to search through causes of deaths for key words and return a list of any causes that contained the key word

Graphical user interface

Description automatically generated with medium confidence

The function creates a list within the function. The list is created within the function so when the function is re-ran for additional key words searches, I always append to a new list. The function will return a new list every time.

I use wild card before and after the search word so the function will return anything in the searched column that contains the search word. I also uses .lower() on both the search word and on the search column to ensure there will be no mis match because of capitalisation on either search word or column text

The second function is used to subset the dataset so I can review the data based on a number of selected attributes

Results (Include the charts and describe them)

1. Compare death by year with quarterly death grouped by year.

The number of deaths reported in both data sets are strongly correlated and it justifies joining quarterly death statistics for missing data

Chart, histogram

Description automatically generated

1. Population plotted against death count

Chart, histogram

Description automatically generated

1. Meningitis is part of the vaccination program for babies. There is a clear decline in deaths caused by meningitis since 1980.

Chart, bar chart, histogram

Description automatically generated

1. Deaths caused by Meningitis broken out by sex

Chart, bar chart

Description automatically generated

1. To confirm the trend in deaths caused by meningitis it was plotted as a percentage of the population

Chart, line chart

Description automatically generated

1. Plotting the deaths occurred by age group by year

Chart, bar chart

Description automatically generated

1. The deaths caused by suicide by year

Chart, bar chart, histogram

Description automatically generated

1. Plotting GDP and unemployment rate against the death count

Chart, scatter chart

Description automatically generated

**Insights (Point out at least 5 insights in bullet points) References (Include any references if required**

* The number of deaths was in decline up to 2006 and it is on a steady increase since then.

Although there was a Pandemic in 2020, there is no dramatic change in the death rate, how much of this is due to the restriction? When 2020 mortality data is released, a breakdown of the deaths by month to correlate with the restrictions will help to answer that question.

* Deaths caused by meningitis have been in decline since at least 1980.
* Deaths caused by meningitis are more prevalent in the 0 to 4 age group
* There is no discernible difference in deaths caused by meningitis amongst male and females, i.e. meningitis is not biased towards sex.

Although there is no vaccination data to correlate against for meningitis, we can say with a strong degree of confidence that Meningitis is in decline. It will be interesting to see over the next couple of years with ‘Fake news’ and anti-vaccination campaigns, will the positive trend reverse.

* Deaths caused by suicide has increased through the years 1980 up to 2012 and is in decline since then.
* There is no obvious relationship between suicide numbers and GDP or unemployment rate.

**Insights (Point out at least 5 insights in bullet points) References (Include any references if required**

Because I don’t know if the insights are on the project are on the use of python I include both

1. There are a number of ways of cleaning data and dealing with NaNs. Preserving data is important but assuming NaN as other values can be dangerous.
2. The use of functions can be very time efficient. In the project, creating a function to subset data based on key words is useful to save time and rewriting code.
3. As iterating through a dataframe is iterating though a copy of a dataframe, the iterrow function proves more challenging.
4. Overlapping line graphs with bar charts is powerful in comparing two stats.
5. Getting the presentation of a chart is as important as the data itself. The wrong size of the graph, too bright colours used, overuse of data can all distract from the message in the chart.