Contents

[1. Creating the database and importing the data 1](#_Toc97221867)

[2. Checks 2](#_Toc97221868)

[3. Exploratory Analysis 2](#_Toc97221869)

[3.1 Calculating Death Rate 2](#_Toc97221870)

[3.2 Percentage of the Population that have tested positive 3](#_Toc97221871)

[3.3 Countries with highest death counts 3](#_Toc97221872)

[3.4 Checking manual calculation of total world deaths vs the value in the data 3](#_Toc97221873)

[3.5 Inner Join 4](#_Toc97221874)

[3.6 Investigating the effect of vaccinations on deaths 4](#_Toc97221875)

[Comparing the total number of vaccinations vs the number of new deaths 4](#_Toc97221876)

[Comparing the percentage vaccination vs death rate. 5](#_Toc97221877)

[Smoothing the data to make it less misleading 6](#_Toc97221878)

# 1. Creating the database and importing the data

-- Creating the database in MySQL

CREATE DATABASE covid19\_data;

USE covid19\_data;

-- Creating a table for the covid deaths csv

CREATE TABLE covid\_deaths (

    iso\_code CHAR(3), continent TINYTEXT,

    location TEXT, date\_ DATE, population INT, total\_cases INT,

    new\_cases INT, new\_cases\_smoothed FLOAT(3), total\_deaths INT,

    new\_deaths INT, new\_deaths\_smoothed FLOAT(3), total\_cases\_per\_million FLOAT(3),

    new\_cases\_per\_million FLOAT(3), new\_cases\_smoothed\_per\_million FLOAT(3),

    total\_deaths\_per\_million FLOAT(3), new\_deaths\_per\_million FLOAT(3),

    new\_deaths\_smoothed\_per\_million FLOAT(3), reproduction\_rate FLOAT(3),

    icu\_patients INT, icu\_patients\_per\_million FLOAT(3), hosp\_patients INT,

    hosp\_patients\_per\_million FLOAT(3), weekly\_icu\_admissions INT,

    weekly\_icu\_admissions\_per\_million FLOAT(3), weekly\_hosp\_admissions INT,

    weekly\_hosp\_admissions\_per\_million FLOAT(3)

    );

CREATE TABLE covid\_vaccinations (

    iso\_code CHAR(3), continent TINYTEXT, location TEXT, date\_ DATE,

    new\_tests INT, total\_tests INT,

    total\_tests\_per\_thousand FLOAT(3),

    new\_tests\_per\_thousand FLOAT(3),

    new\_tests\_smoothed INT,

    new\_tests\_smoothed\_per\_thousand FLOAT(3),

    positive\_rate FLOAT(3), tests\_per\_case FLOAT(3),

    tests\_units TEXT, total\_vaccinations INT,

    people\_vaccinated INT, people\_fully\_vaccinated INT,

    total\_boosters INT, new\_vaccinations INT,

    new\_vaccinations\_smoothed INT,

    total\_vaccinations\_per\_hundred FLOAT(3),

    people\_vaccinated\_per\_hundred FLOAT(3),

    people\_fully\_vaccinated\_per\_hundred FLOAT(3),

    total\_boosters\_per\_hundred FLOAT(3),

    new\_vaccinations\_smoothed\_per\_million INT,

    new\_people\_vaccinated\_smoothed INT,

    new\_people\_vaccinated\_smoothed\_per\_hundred FLOAT(3),

    stringency\_index FLOAT(3), population\_density FLOAT(3),

    median\_age INT, aged\_65\_older FLOAT(3), aged\_70\_older FLOAT(3),

    gdp\_per\_capita FLOAT(3), extreme\_poverty FLOAT(3),

    cardiovasc\_death\_rate FLOAT(3), diabetes\_prevalence FLOAT(3),

    female\_smokers FLOAT(3), male\_smokers FLOAT(3),

    handwashing\_facilities FLOAT(3),

    hospital\_beds\_per\_thousand FLOAT(3),

    life\_expectancy FLOAT(3), human\_development\_index FLOAT(3),

    excess\_mortality\_cumulative\_absolute FLOAT(3),

    excess\_mortality\_cumulative FLOAT(3),

    excess\_mortality FLOAT(3),

    excess\_mortality\_cumulative\_per\_million FLOAT(3)

    );

-- Loading data from my CSVs in the repository into my MySQL DB

LOAD DATA LOCAL INFILE 'C:/Users/Dillon/Documents/Coding Learning/Projects/covid19/CovidDeaths.csv'

INTO TABLE covid\_deaths

COLUMNS TERMINATED BY ','

IGNORE 1 LINES;

LOAD DATA LOCAL INFILE 'C:/Users/Dillon/Documents/Coding Learning/Projects/covid19/CovidVaccinations.csv'

INTO TABLE covid\_vaccinations

COLUMNS TERMINATED BY ','

IGNORE 1 LINES;

# 2. Checks

-- Checking the table has populated

SELECT \*

FROM covid19\_data.covid\_deaths

LIMIT 100;

-- Checking the table has populated

SELECT \*

FROM covid19\_data.covid\_vaccinations

LIMIT 100

# 3. Exploratory Analysis

## 3.1 Calculating Death Rate

-- Total Cases vs Total Deaths

-- Shows death rate (Likelihood of death after contraction)

SELECT location, date\_, total\_cases, total\_deaths, (total\_deaths/total\_cases)\*100 AS prop\_death

FROM covid19\_data.covid\_deaths

WHERE  date\_ = '2022-02-26'

order by 5 DESC;

## 3.2 Percentage of the Population that have tested positive

-- Total Cases vs Population

--Shows percentage of population that have tested positive for the virus (ordered from low to high)

SELECT location, date\_, total\_cases, population, (total\_cases/population)\*100 AS perc\_pop\_pos

FROM covid19\_data.covid\_deaths

WHERE  date\_ = '2022-02-26'

order by 5 DESC;

## 3.3 Countries with highest death counts

-- Finding countries with highest death counts

-- Needed to add the where clause due to data in table for continents and the world. Clause excludes the data to show only countries

-- Change the <> to = to filter by continents & the world instead of countries

SELECT location, MAX(total\_deaths) as totaldeathcount

FROM covid19\_data.covid\_deaths

WHERE continent <> ""

GROUP BY location

ORDER BY totaldeathcount DESC;

The results from this were interesting, the results for deaths for income ranges were almost opposite of what was expected. Low income showed the least deaths vs upper-middle at the top. Obviously lower income areas have less access to healthcare or at least less frequent access and access to lower quality healthcare.

It’s worth checking if this is due to less testing in lower income areas.

## 3.4 Checking manual calculation of total world deaths vs the value in the data

-- Running a check on the "World" count, comparing the datas value vs my calculated value

-- Pulling the value from the data

SELECT location, MAX(total\_deaths) as totaldeathcount

FROM covid19\_data.covid\_deaths

WHERE location = "world"

GROUP BY location

ORDER BY totaldeathcount DESC;

-- Calculating it myself

SELECT SUM(totals)

FROM

    (SELECT MAX(total\_deaths) AS totals

    FROM covid\_deaths

    WHERE continent <> ""

    GROUP BY location)

AS totalworlddeaths;

The calculated value was 5,944,420 vs the real value of 5,944,313. A difference of 107. It might be possible that a country was double counted or the WHERE continent <> "" clause didn’t filter out all non-countries.

After a check in excel, it seems that all locations from the manual calc are real countries so there might be an issue with the dataset’s calculation of the world value.

## 3.5 Inner Join

-- Joining the vaccinations and deaths table

-- This will inner join all columns (join where both tables match in the equations below) both tables

SELECT \*

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

## 3.6 Investigating the effect of vaccinations on deaths

### Comparing the total number of vaccinations vs the number of new deaths

-- Pulling columnns of interest from the joined table

-- Filtering for Irish data

SELECT dea.location, dea.date\_, total\_vaccinations, people\_vaccinated, people\_fully\_vaccinated, total\_deaths, new\_deaths

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

WHERE dea.location = "Ireland"

Exporting this to a file to then import into Tableau.

INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vaccs\_vs\_deaths\_ireland.csv'

    FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n'

Some data were missing from the table.

Chart, line chart

Description automatically generated

SQL was used to clean the data.

The query above was modified to remove the rows where data was missing. The action to remove rather than interpolate between the rows was chosen because there is a lot of data to work with and it wouldn’t have a significant effect on the data.

-- Pulling columnns of interest from the joined table

-- Filtering for Irish data

SELECT dea.location, dea.date\_, total\_vaccinations, people\_vaccinated, people\_fully\_vaccinated, total\_deaths, new\_deaths

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

WHERE dea.location = "Ireland" AND (dea.date\_ < '2020-12-28' OR (dea.date\_ >= 2020-12-28 AND people\_fully\_vaccinated <>0 ))

Chart, line chart

Description automatically generated

Final Result; [Link to tableau sheet here](https://public.tableau.com/app/profile/dillon.o.rourke/viz/Covid19PeopleVaccinatedvsNewDeathsIreland/Sheet1)

### Comparing the percentage vaccination vs death rate.

Calculation of % vaccination has to be done manually. First the total number of people eligible for the vaccine must be calculated.

According to <https://covid19ireland-geohive.hub.arcgis.com/> , 84.39% of people have been fully vaccinated as of 2nd March 2022.

Total 2nd dose vaccines administered as of 2nd Mar 2022 = 3,734,734.

3734734/0.8439 = 4425564.64036

Cant have half a person so round down to 4,425,564 eligible for vaccination.

But we’ll simply use the population of the country in our data and ignore the eligibility of people for vaccines to calculate our percentages.

-- Looking at the effect of vac on death rate in Ireland

SELECT dea.location, population, dea.date\_, (people\_fully\_vaccinated/population)\*100 AS PercentageVac, (total\_deaths/total\_cases)\*100 AS PercDeath

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

WHERE dea.location = "Ireland" AND (dea.date\_ < '2020-12-28' OR (dea.date\_ >= 2020-12-28 AND people\_fully\_vaccinated <>0 ))

INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/%vaccs\_vs\_%deaths\_ireland.csv'

    FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n'

;

[Link to Tableau Dashboard](https://public.tableau.com/app/profile/dillon.o.rourke/viz/EffectsofVaccinesonCovid-19inIreland/Dashboard1) that shows the effect of vaccines on deaths and also compares the effect of increasing percentage of vaccinated people vs death rate.

Graphical user interface, chart, histogram

Description automatically generated

### Smoothing the data to make it less misleading

The reporting or gathering on deaths in the later stages of the pandemic after the 17th of June 2021 changed from daily reports to weekly. This change in frequency makes the latter half of the pandemic a bit misleading in terms of frequency of deaths.

Chart, histogram

Description automatically generated

This highlights the importance of reviewing what exactly is shown by visualisations. In our case we are graphing both daily data and weekly data, this could easily lead someone to misinterpret and assume we are showing how many people die per day from Covid-19 when in reality we are simply graphing the reported deaths.

Using a 7-day rolling average of new deaths we obtain the following variation of the graph.

Chart, line chart

Description automatically generated

This graph is more appropriate as it is less misleading and approximates daily new deaths rather than graphing the reports.

## 3.7 Calculating the Cumulative Sum of Vaccinations as days go by

-- Calculating the Cumulative Sum of Vaccinations as days go by

SELECT dea.date\_, dea.location, population, vac.new\_vaccinations,

SUM(vac.new\_vaccinations) OVER (partition by location ORDER BY location, dea.date\_) AS cumsum\_vac

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

#WHERE dea.location = "Ireland"

;

Table

Description automatically generated

Using CTEs and Temp Tables to

USE covid19\_data;

USE covid19\_data;

-- Calculating the Cumulative Sum of Vaccinations as days go by

-- Common Table Expression

WITH pop\_vs\_vac (Date, Continent, Location, Population, NewVaccs, CumSumVac)

as

(

-- Calculating the Cumulative Sum of Vaccinations as days go by

SELECT dea.date\_, dea.continent, dea.location, population, new\_people\_vaccinated\_smoothed,

SUM(new\_people\_vaccinated\_smoothed) OVER (partition by location ORDER BY location, dea.date\_) AS cumsum\_vac

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

WHERE dea.continent IS NOT NULL

)

-- Calculating the cumulative % pop vaccinated

SELECT \* , (CumSumVac/Population)\*100 AS cum\_perc\_vac

FROM pop\_vs\_vac;

-- Using a Temp table

DROP TABLE IF EXISTS PercentPopulationVaccinated;

CREATE TEMPORARY TABLE PercentPopulationVaccinated

(

Date\_ date,

Continent nvarchar(255),

Location nvarchar(255),

Population numeric,

New\_Vaccinations numeric,

CumSumVac numeric

);

INSERT INTO PercentPopulationVaccinated

-- Calculating the Cumulative Sum of Vaccinations as days go by

(SELECT dea.date\_, dea.continent, dea.location, population, new\_people\_vaccinated\_smoothed,

SUM(new\_people\_vaccinated\_smoothed) OVER (partition by location ORDER BY location, dea.date\_) AS cumsum\_vac

FROM covid19\_data.covid\_deaths dea

JOIN covid19\_data.covid\_vaccinations vac

    ON dea.location = vac.location

    AND dea.date\_ = vac.date\_

WHERE dea.continent IS NOT NULL

);

-- Calculating the cumulative % pop vaccinated

SELECT \* , (CumSumVac/Population)\*100 AS cum\_perc\_vac

FROM PercentPopulationVaccinated