Exploration of Covid Data in SQL

James Wheeler

2022-01-13

My intent with this exploration was to see what in the data set interests me and casually consider the validity of the data.

This data was found via *Our World in Data*.

Organizing the Data

I split the data into more manageable chunks to import into *MySQL*. One being death related data and the other being vaccination related data.

```
colnames(covid_raw)
# population is col 47, place it col 5 between date and total_cases
covid_raw_pop <- covid_raw %>%
  relocate(population, .before = total_cases)
colnames(covid_raw_pop)

# Splitting at 27, new_tests
covid_deaths <- covid_raw_pop[-c(27:65)]
# Using the other half, keeping 1:4
covid_vaccinations <- covid_raw_pop[-c(5:26)]

# Moving them over to MySQL to work on them there
write.csv(covid_deaths,
          "C:\\Users\\15037\\Desktop\\DA\\covid_project\\covid_deaths.csv",
          row.names = FALSE)

write.csv(covid_vaccinations,
          "C:\\Users\\15037\\Desktop\\DA\\covid_project\\covid_vaccinations.csv",
          row.names = FALSE)
# Both exported without issue
```

Having no luck with a fast import to *MySQL*, I returned to R and used *tidyquery* and *queryparser*. These packages interpret most *SQL* queries and rebuild them as *dplyr* pipes, run them and return a data frame. Example of the translation from *SQL* to *dplyr*:

```
show_dplyr('SELECT total_deaths, location FROM covid_deaths WHERE date = "2022-01-12"')

covid_deaths %>%
  filter(date == "2022-01-12") %>%
  select(total_deaths, location) %>%
  arrange(desc(total_deaths))
```

Explore the data

Looked over entire data sets and isolating cols that drew my interest

```
output0 <- query('SELECT * FROM covid_deaths ORDER BY location, date')
view(output0)

output1 <- query('SELECT * FROM covid_vaccinations ORDER BY location, date')
view(output1)


output2 <- query('SELECT location, date, total_cases, new_cases, total_deaths,
                   population
                  FROM covid_deaths
                  ORDER BY location, date')
view(output2)
```

Covid death data

Percentage of daily deaths/daily new cases in the United States.

```
output3 <- query('SELECT location, date, total_cases, total_deaths,
                    ROUND((total_deaths/total_cases)*100, 2) as death_percentage
                  FROM covid_deaths
                  WHERE location = "United States"
                  ORDER BY date DESC')
```

```
## # A tibble: 6 x 5
##   location      date       total_cases total_deaths death_percentage
##   <chr>         <date>           <dbl>        <dbl>            <dbl>
## 1 United States 2022-01-12    63203443       844562             1.34
## 2 United States 2022-01-11    62308472       842141             1.35
## 3 United States 2022-01-10    61556085       839500             1.36
## 4 United States 2022-01-09    60090560       837665             1.39
## 5 United States 2022-01-08    59767418       837266             1.4
## 6 United States 2022-01-07    59388686       836605             1.41
```

Percentage of the population who have had reported Covid cases in the United States. This could be skewed by individuals having multiple cases (double report) or unreported cases.

```
output4 <- query('SELECT location, date, total_cases, population,
                    ROUND((total_cases/population)*100, 2) as infection_percentage
                  FROM covid_deaths
                  WHERE location = "United States"
                  ORDER BY date DESC')
```

```
## # A tibble: 722 x 5
##    location      date       total_cases population infection_percentage
##    <chr>         <date>           <dbl>      <dbl>                <dbl>
##  1 United States 2022-01-12    63203443  332915074                 19.0
##  2 United States 2022-01-11    62308472  332915074                 18.7
##  3 United States 2022-01-10    61556085  332915074                 18.5
```

```
##  4 United States 2022-01-09    60090560  332915074                18.0
##  5 United States 2022-01-08    59767418  332915074                18.0
##  6 United States 2022-01-07    59388686  332915074                17.8
##  7 United States 2022-01-06    58487854  332915074                17.6
##  8 United States 2022-01-05    57700993  332915074                17.3
##  9 United States 2022-01-04    57077603  332915074                17.1
## 10 United States 2022-01-03    56278376  332915074                16.9
## # ... with 712 more rows
```

Countries with highest reported percentage of population infected

```
output5 <- query('SELECT location, MAX(total_cases) as highest_infection_count, population,
                   ROUND(MAX((total_cases/population))*100, 2) as percent_pop_infected
                 FROM covid_deaths
                 GROUP BY population, location
                 ORDER BY percent_pop_infected desc')
```

```
## # A tibble: 238 x 4
##    population location   highest_infection_count percent_pop_infected
##         <dbl> <chr>                        <dbl>                <dbl>
##  1      77354 Andorra                      28899                 37.4
##  2     628051 Montenegro                  196640                 31.3
##  3      33691 Gibraltar                    10248                 30.4
##  4      34010 San Marino                   10009                 29.4
##  5      98910 Seychelles                   29030                 29.4
##  6     107195 Aruba                        29176                 27.2
##  7    5460726 Slovakia                    1405854                 25.7
##  8    3979773 Georgia                      979235                 24.6
##  9    2078723 Slovenia                     505929                 24.3
## 10     896005 Cyprus                       215271                 24.0
## # ... with 228 more rows
```

Dropping continent grouped data, `LIKE` practice

```
output6 <- query('SELECT *
                 FROM covid_deaths
                 WHERE iso_code not like "%OWID_%"')
view(output6)
```

Countries sorted by highest death count

```
output7 <- query('SELECT location, MAX(total_deaths) as total_death_count
                 FROM covid_deaths
                 WHERE iso_code not like "%OWID_%"
                 GROUP BY location
                 ORDER BY total_death_count desc')
```

```
## # A tibble: 6 x 2
##   location      total_death_count
##   <chr>                     <dbl>
## 1 United States            844562
## 2 Brazil                   620641
```

```
## 3 India                    485035
## 4 Russia                   312010
## 5 Mexico                   300764
## 6 Peru                     203157
```

Exclusively continent data

```
output8 <- query('SELECT continent, MAX(total_deaths) as total_death_count
                  FROM covid_deaths
                  WHERE iso_code not like "%OWID_%"
                  AND total_deaths >= 1
                  GROUP BY continent
                  ORDER BY total_death_count desc')
```

This gave back the highest total death count for a single location in the continent.

```
## # A tibble: 6 x 2
##   continent     total_death_count
##   <chr>                     <dbl>
## 1 North America            844562
## 2 South America            620641
## 3 Asia                     485035
## 4 Europe                   312010
## 5 Africa                    92830
## 6 Oceania                    2522
```

Using location and finding the correct numbers for continent totals

```
output9 <- query('SELECT location, MAX(total_deaths) as total_death_count
                  FROM covid_deaths
                  WHERE iso_code like "%OWID_%"
                  AND location not like "%income"
                  AND total_deaths >= 1
                  GROUP BY location
                  ORDER BY total_death_count desc')
```

```
## # A tibble: 10 x 2
##    location       total_death_count
##    <chr>                      <dbl>
##  1 World                    5513550
##  2 Europe                   1564969
##  3 Asia                     1272248
##  4 North America            1243952
##  5 South America            1195610
##  6 European Union            923710
##  7 Africa                    231941
##  8 Oceania                     4815
##  9 Kosovo                      2992
## 10 International                15
```

Daily global death percentage (progressive sum)

```
output10 <- query('SELECT date, SUM(total_cases) as total_cases, SUM(total_deaths) as
                    total_deaths, ROUND((SUM(total_deaths)/SUM(total_cases))*100, 2) as
                    death_percentage
                  FROM covid_deaths
                  WHERE iso_code not like "%OWID_%"
                  GROUP BY date
                  ORDER BY total_cases DESC')
```

```
## # A tibble: 6 x 4
##   date       total_cases total_deaths death_percentage
##   <date>           <dbl>        <dbl>            <dbl>
## 1 2022-01-12   317000858      5510543             1.74
## 2 2022-01-11   313333325      5501381             1.76
## 3 2022-01-10   310465012      5492707             1.77
## 4 2022-01-09   307178903      5486281             1.79
## 5 2022-01-08   305170771      5482324             1.8
## 6 2022-01-07   303106684      5477294             1.81
```

Daily global new death per new case

```
output11 <- query('SELECT date, SUM(new_cases) as new_daily_cases, SUM(new_deaths) as
                    new_daily_deaths, ROUND((SUM(new_deaths)/SUM(new_cases)), 4) as
                    new_death_per_new_case
                  FROM covid_deaths
                  WHERE iso_code not like "%OWID_%"
                  GROUP BY date
                  ORDER BY date DESC')
```

```
## # A tibble: 743 x 4
##    date       new_daily_cases new_daily_deaths new_death_per_new_case
##    <date>               <dbl>            <dbl>                  <dbl>
##  1 2022-01-12         3667533             9162                 0.0025
##  2 2022-01-11         2868313             8839                 0.0031
##  3 2022-01-10         3286109             6426                 0.002
##  4 2022-01-09         2008132             3957                 0.002
##  5 2022-01-08         2064087             5030                 0.0024
##  6 2022-01-07         2914673             7269                 0.0025
##  7 2022-01-06         2557067             6707                 0.0026
##  8 2022-01-05         2513125             7774                 0.0031
##  9 2022-01-04         2543116             7777                 0.0031
## 10 2022-01-03         2524202             5897                 0.0023
## # ... with 733 more rows
```

Total global new deaths per new cases

```
output12 <- query('SELECT SUM(new_cases) as new_daily_cases, SUM(new_deaths) as
                    new_daily_deaths, ROUND((SUM(new_deaths)/SUM(new_cases))*100, 2)
                    as new_death_per_new_case
                  FROM covid_deaths
                  WHERE iso_code not like "%OWID_%"')
```

```
## # A tibble: 1 x 3
```

```
##   new_daily_cases new_daily_deaths new_death_per_new_case
##             <dbl>            <dbl>                  <dbl>
## 1       316428071          5485211                   1.73
```

Rejoining vaccination data

Checking vaccination data frame

```
output13 <- query('SELECT *
                  FROM covid_vaccinations')
view(output13)
```

Testing SQL join

```
covid_test <- query('SELECT *
                    FROM covid_deaths dea
                    JOIN covid_vaccinations vac
                      ON dea.location = vac.location
                      AND dea.date = vac.date')
# Checking if the join worked
view(covid_test)
colnames(covid_test) # all present + joining cols
view(covid_test) # checking entry count
view(covid_deaths) # covid_test and covid_death have the same number of entries
covid_full <- covid_test
```

Adding a summative column of vaccinations day over day

```
output15 <- query('SELECT dea.continent, location, date, population, new_vaccinations,
                    SUM(new_vaccinations) OVER (PARTITION BY location) as
                    rolling_people_vaccinated
                  FROM covid_full
                  WHERE dea.iso_code not like "%OWID_%"
                  ORDER BY location, date')
view(output15)
```

tidyquery doesn't support OVER clauses, rewriting in *dplyr*

```
output_test <- query('SELECT dea.continent, location, date, population, new_vaccinations
                     FROM covid_full
                     WHERE dea.iso_code not like "%OWID_%"')

output_test[is.na(output_test)] <- 0 # NA issue, will not stick to covid_full

output_works <- output_test %>%
  select(dea.continent, location, date, population, new_vaccinations) %>%
  group_by(location) %>%
  arrange(location, date) %>%
  mutate(rolling_people_vaccinated = cumsum(new_vaccinations)) %>%
  arrange(location, date) %>%
  select(everything())
view(output_works)
```

```
output15 <- output_works
view(output15) # desired outcome using dplyr

rm(output_test) # Clean up
rm(output_works)
```

Calculated total_vaccinations differ from 'true' total_vaccinations as seen in the original data. Searching shows that 'true' total_vaccinations changed without any documented new vaccinations.

Checking over the data also showed that many locations were well over 100% vaccinated. I isolated and ordered those cols to show locations with the highest percent vaccinated as of the most recent data. It's likely that second and third vaccinations are being counted in total_vaccinations.

```
output16 <- query('SELECT location, date, population, total_vaccinations,
                   ROUND((total_vaccinations/population)*100, 2) as percent_vaccinated
                  FROM covid_full
                  WHERE date == "2022-01-12"
                  ORDER BY percent_vaccinated DESC')
```

```
## # A tibble: 6 x 5
##   location             date       population total_vaccinations percent_vaccina~
##   <chr>                <date>          <dbl>              <dbl>             <dbl>
## 1 Gibraltar            2022-01-12      33691             109110              324.
## 2 United Arab Emirates 2022-01-12    9991083           22929333              230.
## 3 Malta                2022-01-12     516100            1152980              223.
## 4 Denmark              2022-01-12    5813302           12706992              219.
## 5 Isle of Man          2022-01-12      85410             182880              214.
## 6 South Korea          2022-01-12   51305184          108323390              211.
```

Reflection

This data set was interesting to check out. I was particularly interested in deaths relative to cases, both in total and daily. I initially did this exploration in early October of 2021 and the change from then to now (January 2022) was staggering. When isolating the United States, watching the percentage of deaths per case fluctuate was particularly interesting. Almost like wave pattern the percentage would drop when infection rate was escalating, then raise when death rate started to escalate. Another staggering statistic was the 37.4% of Andorra having been infected. Though to see how long ago it took for the United States move from 10% to 19%, we may reach the upper 30%.