

Data Structures in R: covid19

Sunsun Chanakarn

2025-08-21

Understanding the Data

Loading the dataset from the covid19.csv CSV file and quick exploration

```
install.packages(c("dplyr", "tibble", "readr"))

## Installing packages into '/cloud/lib/x86_64-pc-linux-gnu-library/4.5'
## (as 'lib' is unspecified)

library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(tibble)
library(readr)

# Loading the dataset
covid_df <- read_csv("covid19.csv")

## Rows: 10903 Columns: 14
## -- Column specification -----
## Delimiter: ","
## chr  (4): Continent_Name, Two_Letter_Country_Code, Country_Region, Province...
## dbl  (9): positive, hospitalized, recovered, death, total_tested, active, ho...
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

# Displaying the dimension of the data
dim(covid_df)

## [1] 10903    14

# Storing the column names in a variable
vector_cols <- colnames(covid_df)
```

```
# Displaying the variable vector_cols
vector_cols
```

```
## [1] "Date" "Continent_Name"
## [3] "Two_Letter_Country_Code" "Country_Region"
## [5] "Province_State" "positive"
## [7] "hospitalized" "recovered"
## [9] "death" "total_tested"
## [11] "active" "hospitalizedCurr"
## [13] "daily_tested" "daily_positive"
```

```
# Showing the first few rows of the dataset
head(covid_df)
```

```
## # A tibble: 6 x 14
##   Date      Continent_Name Two_Letter_Country_C-1 Country_Region Province_State
##   <date>      <chr>          <chr>          <chr>          <chr>
## 1 2020-01-20 Asia          KR              South Korea    All States
## 2 2020-01-22 North America US              United States  All States
## 3 2020-01-22 North America US              United States  Washington
## 4 2020-01-23 North America US              United States  All States
## 5 2020-01-23 North America US              United States  Washington
## 6 2020-01-24 Asia          KR              South Korea    All States
## # i abbreviated name: 1: Two_Letter_Country_Code
## # i 9 more variables: positive <dbl>, hospitalized <dbl>, recovered <dbl>,
## #   death <dbl>, total_tested <dbl>, active <dbl>, hospitalizedCurr <dbl>,
## #   daily_tested <dbl>, daily_positive <dbl>
```

```
# Showing a global view of the dataset.
glimpse(covid_df)
```

```
## Rows: 10,903
## Columns: 14
## $ Date      <date> 2020-01-20, 2020-01-22, 2020-01-22, 2020-01-2~
## $ Continent_Name <chr> "Asia", "North America", "North America", "Nor~
## $ Two_Letter_Country_Code <chr> "KR", "US", "US", "US", "US", "KR", "US", "US"~
## $ Country_Region <chr> "South Korea", "United States", "United States~
## $ Province_State <chr> "All States", "All States", "Washington", "All~
## $ positive     <dbl> 1, 1, 1, 1, 1, 2, 1, 1, 4, 0, 3, 0, 0, 0, 0, 1~
## $ hospitalized <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ recovered    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ death        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ total_tested <dbl> 4, 1, 1, 1, 1, 27, 1, 1, 0, 0, 0, 0, 0, 0, 0, ~
## $ active       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ hospitalizedCurr <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ daily_tested <dbl> 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ daily_positive <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

The dataset contains 14 columns and 10,903 rows. This database provides information on the numbers (per day and cumulatively) of COVID-19 positive cases, deaths, tests performed and hospitalizations for each country through the column's names store in the variable `vector_cols`.

1. This variable contains a character vector.
2. The use of the function `glimpse()` is the very first operation to do because we don't only learn about the dimensions of the database but also about the names of the first columns and their types and

content. It can replace the three previous operations: `dim()`, `colnames()`, and `head()`.

Isolating the Rows We Need

- Selecting only the rows related to "All States" and removing the `Province_State`.

We do this to select only the data that has been consolidated from all states. This makes it easier to analyze the overall national picture, and we no longer need the `Province_State` column which now contains no data.

```
# Filter the "All States" Province states and remove the `Province_State` column
covid_df_all_states <- covid_df %>%
  filter(Province_State == "All States") %>%
  select(-Province_State)

head(covid_df_all_states)
```

```
## # A tibble: 6 x 13
##   Date      Continent_Name Two_Letter_Country_Code Country_Region positive
##   <date>    <chr>           <chr>           <chr>           <dbl>
## 1 2020-01-20 Asia             KR             South Korea       1
## 2 2020-01-22 North America    US             United States     1
## 3 2020-01-23 North America    US             United States     1
## 4 2020-01-24 Asia             KR             South Korea       2
## 5 2020-01-24 North America    US             United States     1
## 6 2020-01-25 Oceania          AU             Australia         4
## # i 8 more variables: hospitalized <dbl>, recovered <dbl>, death <dbl>,
## #   total_tested <dbl>, active <dbl>, hospitalizedCurr <dbl>,
## #   daily_tested <dbl>, daily_positive <dbl>
```

- We can remove `Province_State` without losing information because after the filtering step this column only contains the value "All States".

Isolating the Columns We Need

- Creating a dataset for the daily columns from `covid_df_all_states` dataframe

Let's recall the description of the dataset's columns.

1. `Date`: Date
2. `Continent_Name`: Continent names
3. `Two_Letter_Country_Code`: Country codes
4. `Country_Region`: Country names
5. `Province_State`: States/province names; value is `All States` when state/provincial level data is not available
6. `positive`: Cumulative number of positive cases reported.
7. `active`: Number of actively cases on that `day`.
8. `hospitalized`: Cumulative number of hospitalized cases reported.
9. `hospitalizedCurr`: Number of actively hospitalized cases on that `day`.
10. `recovered`: Cumulative number of recovered cases reported.
11. `death`: Cumulative number of deaths reported.
12. `total_tested`: Cumulative number of tests conducted.
13. `daily_tested`: Number of tests conducted on the `day`; if daily data is unavailable, daily tested is averaged across number of days in between.
14. `daily_positive`: Number of positive cases reported on the `day`; if daily data is unavailable, daily positive is averaged across number of days in.

```
# Selecting the columns with cumulative numbers
covid_df_all_states_daily <- covid_df_all_states %>%
  select(Date, Country_Region, active, hospitalizedCurr, daily_tested, daily_positive)

head(covid_df_all_states_daily)
```

```
## # A tibble: 6 x 6
##   Date      Country_Region active hospitalizedCurr daily_tested daily_positive
##   <date>    <chr>          <dbl>          <dbl>          <dbl>          <dbl>
## 1 2020-01-20 South Korea      0              0              0              0
## 2 2020-01-22 United States  0              0              0              0
## 3 2020-01-23 United States  0              0              0              0
## 4 2020-01-24 South Korea      0              0              5              0
## 5 2020-01-24 United States  0              0              0              0
## 6 2020-01-25 Australia      0              0              0              0
```

Extracting the Top Ten countries in the number of tested cases

Summarizing the data based on the Country_Region column.

```
covid_df_all_states_daily_sum <- covid_df_all_states_daily %>%
  group_by(Country_Region) %>%
  summarise(tested = sum(daily_tested),
            positive = sum(daily_positive),
            active = sum(active),
            hospitalized = sum(hospitalizedCurr)) %>%
  arrange(desc(tested)) #this is equivalent to `arrange(-tested)`

covid_df_all_states_daily_sum
```

```
## # A tibble: 108 x 5
##   Country_Region tested positive active hospitalized
##   <chr>          <dbl>    <dbl>    <dbl>          <dbl>
## 1 United States 17282363 1877179      0           0
## 2 Russia        10542266 406368 6924890      0
## 3 Italy          4091291 251710 6202214 1699003
## 4 India          3692851 60959      0           0
## 5 Turkey         2031192 163941 2980960      0
## 6 Canada         1654779 90873 56454      0
## 7 United Kingdom 1473672 166909      0           0
## 8 Australia      1252900 7200 134586 6655
## 9 Peru           976790 59497      0           0
## 10 Poland         928256 23987 538203      0
## # i 98 more rows
```

```
#Date, Country_Region, active, hospitalizedCurr, daily_tested, daily_positive
```

Taking the top 10

```
covid_top_10 <- head(covid_df_all_states_daily_sum, 10)

covid_top_10
```

```
## # A tibble: 10 x 5
##   Country_Region tested positive active hospitalized
##   <chr>          <dbl>    <dbl>    <dbl>         <dbl>
## 1 United States 17282363 1877179      0           0
## 2 Russia        10542266 406368 6924890      0
## 3 Italy          4091291 251710 6202214    1699003
## 4 India          3692851  60959      0           0
## 5 Turkey         2031192 163941 2980960      0
## 6 Canada         1654779  90873  56454       0
## 7 United Kingdom 1473672 166909      0           0
## 8 Australia      1252900   7200 134586      6655
## 9 Peru           976790  59497      0           0
## 10 Poland        928256  23987  538203      0
```

Identifying the Highest Positive Against Tested Cases

Getting vectors

```
countries <- covid_top_10$Country_Region
tested_cases <- covid_top_10$tested
positive_cases <- covid_top_10$positive
active_cases <- covid_top_10$active
hospitalized_cases <- covid_top_10$hospitalized
```

Naming vectors

```
names(positive_cases) <- countries
names(tested_cases) <- countries
names(active_cases) <- countries
names(hospitalized_cases) <- countries
```

Identifying

```
positive_cases
```

```
## United States      Russia      Italy      India      Turkey
##      1877179      406368      251710      60959      163941
##      Canada United Kingdom      Australia      Peru      Poland
##      90873      166909      7200      59497      23987
```

```
sum(positive_cases)
```

```
## [1] 3108623
```

```
mean(positive_cases)
```

```
## [1] 310862.3
```

```
positive_cases/sum(positive_cases)
```

```
## United States      Russia      Italy      India      Turkey
## 0.603861903 0.130722831 0.080971543 0.019609647 0.052737498
##      Canada United Kingdom      Australia      Peru      Poland
## 0.029232557 0.053692262 0.002316138 0.019139342 0.007716278
```

```
positive_cases/tested_cases
```

```
## United States      Russia      Italy      India      Turkey
## 0.108618191 0.038546552 0.061523368 0.016507300 0.080711720
##      Canada United Kingdom Australia      Peru      Poland
## 0.054915490 0.113260617 0.005746668 0.060910738 0.025840932
```

Conclusion

```
positive_tested_top_3 <- c("United Kingdom" = 0.11, "United States" = 0.10, "Turkey" = 0.08)
positive_tested_top_3
```

```
## United Kingdom United States      Turkey
##           0.11           0.10           0.08
```

Keeping relevant information

```
# Creating vectors
united_kingdom <- c(0.11, 1473672, 166909, 0, 0)
united_states <- c(0.10, 17282363, 1877179, 0, 0)
turkey <- c(0.08, 2031192, 163941, 2980960, 0)

# Creating the matrix covid_mat
covid_mat <- rbind(united_kingdom, united_states, turkey)

# Naming columns
colnames(covid_mat) <- c("Ratio", "tested", "positive", "active", "hospitalized")

# Displaying the matrix
covid_mat
```

```
##           Ratio  tested positive  active hospitalized
## united_kingdom 0.11 1473672 166909      0              0
## united_states 0.10 17282363 1877179      0              0
## turkey        0.08 2031192 163941 2980960      0
```

Putting all together

```
question <- "Which countries have had the highest number of positive cases against the number of tests?"

answer <- c("Positive tested cases" = positive_tested_top_3)

datasets <- list(
  original = covid_df,
  allstates = covid_df_all_states,
  daily = covid_df_all_states_daily,
  top_10 = covid_top_10
)

matrices <- list(covid_mat)
vectors <- list(vector_cols, countries)
```



```

data_structure_list <- list("dataframe" = datasets, "matrix" = matrices, "vector" = vectors)

covid_analysis_list <- list(question, answer, data_structure_list)

covid_analysis_list[[2]]

```

```

## Positive tested cases.United Kingdom Positive tested cases.United States
##                                0.11                                0.10
##      Positive tested cases.Turkey
##                                0.08

```