Analysis of Global COVID-19 Pandemic Data

In [1]:
```
#install.packages("httr")
#install.packages("rvest")
```

In [7]:
```
library(httr)
library(rvest)
```

```
Loading required package: xml2
```

Note: if you can import above libraries, please use install.packages() to install them first.

TASK 1: Get a `COVID-19 pandemic` Wiki page using HTTP request

First, let's write a function to use HTTP request to get a public COVID-19 Wiki page.

URL https://en.wikipedia.org/w/index.php?title=Template:COVID-19_testing_by_country (https://en.wikipedia.org/w/index.php?title=Template:COVID-19_testing_by_country) using a web browser.

The goal of task 1 is to get the html page using HTTP request (`httr` library)

In [8]:
```
get_wiki_covid19_page <- function(url, prm) {

  # Our target COVID-19 wiki page URL is: https://en.wikipedia.org/w/index.php?title=Template:COVID-19_testing_by_country
  # Which has two parts:
    # 1) base URL `https://en.wikipedia.org/w/index.php
    # 2) URL parameter: `title=Template:COVID-19_testing_by_country`, seperated by question mark ?

  # Wiki page base
  wiki_base_url <- "https://en.wikipedia.org/w/index.php"
  # You will need to create a list which has an element called `title` to specify which page you want to get from Wiki
  URL_parameter <- list(title = "Template:COVID-19_testing_by_country")

  # in our case, it will be `Template:COVID-19_testing_by_country`

  # - Use the `GET` function in httr library with a `url` argument and a `query` arugment to get a HTTP response
    response <- GET(url = wiki_base_url, query = URL_parameter)

  # Use the `return` function to return the response
    return(response)
}
```

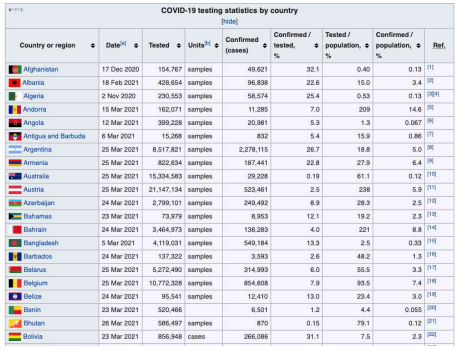
Call the `get_wiki_covid19_page` function to get a http response with the target html page

In [9]:
```
# Call the get_wiki_covid19_page function and print the response
get_wiki_covid19_page("https://en.wikipedia.org/w/index.php","Template:COVID-19_testing_by_country")
```

```
Response [https://en.wikipedia.org/w/index.php?title=Template%3ACOVID-19_testing_by_country]
  Date: 2024-06-04 12:40
  Status: 200
  Content-Type: text/html; charset=UTF-8
  Size: 449 kB
<!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-fea...
<head>
<meta charset="UTF-8">
<title>Template:COVID-19 testing by country - Wikipedia</title>
<script>(function(){var className="client-js vector-feature-language-in-heade...
["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","...
"CS1 uses Russian-language script (ru)","CS1 Russian-language sources (ru)","...
,"CS1 Lithuanian-language sources (lt)","CS1 Malagasy-language sources (mg)",...
"wgRelevantArticleId":63303421,"wgIsProbablyEditable":false,"wgRelevantPageIs...
...
```

TASK 2: Extract COVID-19 testing data table from the wiki HTML page

On the COVID-19 testing wiki page, data table `<table>` node contains COVID-19 testing data by country on the page:



COVID-19 testing statistics by country

[hide]

Country or region	Date[a]	Tested	Units[b]	Confirmed (cases)	Confirmed / tested, %	Tested / population, %	Confirmed / population, %	Ref.
Afghanistan	17 Dec 2020	154,767	samples	49,621	32.1	0.40	0.13	[1]
Albania	18 Feb 2021	428,654	samples	96,838	22.6	15.0	3.4	[2]
Algeria	2 Nov 2020	230,553	samples	58,574	25.4	0.53	0.13	[3][4]
Andorra	15 Mar 2021	162,071	samples	11,285	7.0	209	14.6	[5]
Angola	12 Mar 2021	399,228	samples	20,981	5.3	1.3	0.067	[6]
Antigua and Barbuda	6 Mar 2021	15,268	samples	832	5.4	15.9	0.86	[7]
Argentina	25 Mar 2021	8,517,821	samples	2,278,115	26.7	18.8	5.0	[8]
Armenia	25 Mar 2021	822,634	samples	187,441	22.8	27.9	6.4	[9]
Australia	25 Mar 2021	15,334,583	samples	29,228	0.19	61.1	0.12	[10]
Austria	25 Mar 2021	21,147,134	samples	523,461	2.5	238	5.9	[11]
Azerbaijan	24 Mar 2021	2,799,101	samples	249,492	8.9	28.3	2.5	[12]
Bahamas	23 Mar 2021	73,979	samples	8,953	12.1	19.2	2.3	[13]
Bahrain	24 Mar 2021	3,464,973	samples	138,283	4.0	221	8.8	[14]
Bangladesh	5 Mar 2021	4,119,031	samples	549,184	13.3	2.5	0.33	[15]
Barbados	24 Mar 2021	137,322	samples	3,593	2.6	48.2	1.3	[16]
Belarus	25 Mar 2021	5,272,490	samples	314,993	6.0	55.5	3.3	[17]
Belgium	25 Mar 2021	10,772,328	samples	854,608	7.9	93.5	7.4	[18]
Belize	24 Mar 2021	95,541	samples	12,410	13.0	23.4	3.0	[19]
Benin	23 Mar 2021	520,466		6,501	1.2	4.4	0.055	[20]
Bhutan	26 Mar 2021	586,497	samples	870	0.15	79.1	0.12	[21]
Bolivia	23 Mar 2021	856,948	cases	266,086	31.1	7.5	2.3	[22]

(https://cognitiveclass.ai/)

The goal of task 2 is to extract above data table and convert it into a data frame

Now use the `read_html` function in rvest library to get the root html node from response

Get the tables in the HTML root node using `html_nodes` function.

In [20]:
```
# Get the root html node from the http response in task 1
url <- get_wiki_covid19_page("https://en.wikipedia.org/w/index.php","Template:COVID-19_testing_by_country")
root_node <- read_html(url)
```

In [24]:
```
# Get the table node from the root html node
table_nodes <- html_nodes(root_node, "table")
```

Read the specific table from the multiple tables in the `table_node` using the `html_table` function and convert it into dataframe using `as.data.frame`

In [31]:
```
# Read the table node and convert it into a data frame, and print the data frame for review
covid_data <- html_table(table_nodes[2], fill = TRUE)
covid_data <- as.data.frame(covid_data)
head(covid_data)
```

A data.frame: 6 × 9

	Country.or.region	Date.a.	Tested	Units.b.	Confirmed.cases.	Confirmed..tested..	Tested..population..	Confirmed..population..	Ref.
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	Afghanistan	17 Dec 2020	154,767	samples	49,621	32.1	0.40	0.13	[1]
2	Albania	18 Feb 2021	428,654	samples	96,838	22.6	15.0	3.4	[2]
3	Algeria	2 Nov 2020	230,553	samples	58,574	25.4	0.53	0.13	[3][4]
4	Andorra	23 Feb 2022	300,307	samples	37,958	12.6	387	49.0	[5]
5	Angola	2 Feb 2021	399,228	samples	20,981	5.3	1.3	0.067	[6]
6	Antigua and Barbuda	6 Mar 2021	15,268	samples	832	5.4	15.9	0.86	[7]

TASK 3: Pre-process and export the extracted data frame

The goal of task 3 is to pre-process the extracted data frame from the previous step, and export it as a csv file

Summary of the data frame

In [32]:
```
# Print the summary of the data frame
summary(covid_data)
```

```
 Country.or.region    Date.a.             Tested            Units.b.
 Length:173         Length:173         Length:173         Length:173
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Confirmed.cases.   Confirmed..tested.. Tested..population..
 Length:173         Length:173          Length:173
 Class :character   Class :character    Class :character
 Mode  :character   Mode  :character    Mode  :character
 Confirmed..population..     Ref.
 Length:173              Length:173
 Class :character        Class :character
 Mode  :character        Mode  :character
```

From the summary, the columns names are little bit different to understand and some column data types are not correct. For example, the `Tested` column shows as `character`.

As such, the data frame read from HTML table will need some pre-processing such as removing irrelvant columns, renaming columns, and convert columns into proper data types.

In [33]:
```
preprocess_covid_data_frame <- function(data_frame) {

    shape <- dim(data_frame)

    # Remove the World row
    data_frame<-data_frame[!(data_frame$`Country.or.region`=="World"),]
    # Remove the last row
    data_frame <- data_frame[1:172, ]

    # We dont need the Units and Ref columns, so can be removed
    data_frame["Ref."] <- NULL
    data_frame["Units.b."] <- NULL

    # Renaming the columns
    names(data_frame) <- c("country", "date", "tested", "confirmed", "confirmed.tested.ratio", "tested.population.ratio", "confirmed.population.ratio")

    # Convert column data types
    data_frame$country <- as.factor(data_frame$country)
    data_frame$date <- as.factor(data_frame$date)
    data_frame$tested <- as.numeric(gsub(",","",data_frame$tested))
    data_frame$confirmed <- as.numeric(gsub(",","",data_frame$confirmed))
    data_frame$'confirmed.tested.ratio' <- as.numeric(gsub(",","",data_frame$`confirmed.tested.ratio`))
    data_frame$'tested.population.ratio' <- as.numeric(gsub(",","",data_frame$`tested.population.ratio`))
    data_frame$'confirmed.population.ratio' <- as.numeric(gsub(",","",data_frame$`confirmed.population.ratio`))

    return(data_frame)
}
```

Call the `preprocess_covid_data_frame` function

In [34]:
```
# call `preprocess_covid_data_frame` function and assign it to a new data frame
covid_data_2 <- preprocess_covid_data_frame(covid_data)
```

Summary of the processed data frame

In [35]:
```
# Print the summary of the processed data frame again
summary(covid_data_2)
```

```
                country             date         tested
 Afghanistan        :  1   2 Feb 2023 :  6   Min.   :     3880
 Albania            :  1   1 Feb 2023 :  4   1st Qu.:   512037
 Algeria            :  1   31 Jan 2023:  4   Median :  3029859
 Andorra            :  1   1 Mar 2021 :  3   Mean   : 31377219
 Angola             :  1   23 Jul 2021:  3   3rd Qu.: 12386725
 Antigua and Barbuda:  1   29 Jan 2023:  3   Max.   :929349291
 (Other)            :166   (Other)    :149
   confirmed        confirmed.tested.ratio tested.population.ratio
 Min.   :       0   Min.   : 0.00          Min.   :   0.006
 1st Qu.:   37839   1st Qu.: 5.00          1st Qu.:   9.475
 Median :  281196   Median :10.05          Median :  46.950
 Mean   : 2508340   Mean   :11.25          Mean   : 175.504
 3rd Qu.: 1278105   3rd Qu.:15.25          3rd Qu.: 156.500
 Max.   :90749469   Max.   :46.80          Max.   :3223.000

 confirmed.population.ratio
 Min.   : 0.000
 1st Qu.: 0.425
 Median : 6.100
 Mean   :12.769
 3rd Qu.:16.250
 Max.   :74.400
```

After pre-processing, the columns and columns names are simplified, and columns types are converted into correct types.

The data frame has following columns:

- **country** - The name of the country
- **date** - Reported date
- **tested** - Total tested cases by the reported date
- **confirmed** - Total confirmed cases by the reported date
- **confirmed.tested.ratio** - The ratio of confirmed cases to the tested cases
- **tested.population.ratio** - The ratio of tested cases to the population of the country
- **confirmed.population.ratio** - The ratio of confirmed cases to the population of the country

Call `write.csv()` function to save the csv file into a file.

In [36]:
```
# Export the data frame to a csv file
write.csv(covid_data_2, file='Covid.csv')
```

In [38]:
```
# Get working directory
wd <- getwd()
# Get exported
file_path <- paste(wd, sep="", "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-RP0101EN-Coursera/v2/dataset/covid.csv")
# File path
print(file_path)
file.exists(file_path)
```

```
[1] "/resources/labs/authoride/IBMSkillsNetwork+RP0101EN/v2/M5_Finalhttps://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-RP0101EN-Coursera/v2/dataset/covid.csv"
```

FALSE

In [39]:
```
## Download a sample csv file
 covid_csv_file <- download.file("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-RP0101EN-Coursera/v2/dataset/covid.csv", destfile="covid.csv")
 covid_data_frame_csv <- read.csv("covid.csv", header=TRUE, sep=",")
```

TASK 4: Get a subset of the extracted data frame

The goal of task 4 is to get the 5th to 10th rows from the data frame with only `country` and `confirmed` columns selected

In [40]:
```
# Read covid_data_frame_csv from the csv file
covid_subset <- read.csv('covid.csv')

# Get the 5th to 10th rows, with two "country" "confirmed" columns
covid_subset[5:10,c('country','confirmed')]
```

A data.frame: 6 × 2

	country	confirmed
	<fct>	<int>
5	Angola	20981
6	Antigua and Barbuda	832
7	Argentina	2195722
8	Armenia	177104
9	Australia	29130
10	Austria	488007

TASK 5: Calculate worldwide COVID testing positive ratio

The goal of task 5 is to get the total confirmed and tested cases worldwide, and try to figure the overall positive ratio using `confirmed cases / tested cases`

In [41]:
```
# Get the total confirmed cases worldwide
sum(covid_subset$confirmed)

# Get the total tested cases worldwide
sum(covid_subset$tested)

# Get the positive ratio (confirmed / tested)
positive_ratio <- sum(covid_subset$confirmed)/sum(covid_subset$tested)
positive_ratio
```

117313932

1698581244

0.0690658350399164

TASK 6: Get a country list which reported their testing data

The goal of task 6 is to get a catalog or sorted list of countries who have reported their COVID-19 testing data

In [45]:

```r
# Get the `country` column
countries <- covid_subset$country

# Check its class (should be Factor)
class(countries)

# Conver the country column into character so that you can easily sort them
countries <-  as.character(countries)

# Sort the countries AtoZ
AZ <- sort(countries)

# Sort the countries ZtoA
ZA <- sort(countries, decreasing = TRUE)

# Print the sorted ZtoA list
print(ZA)
```

'factor'

```
  [1] "Zimbabwe"                "Zambia"                  "Vietnam"
  [4] "Venezuela"               "Uzbekistan"              "Uruguay"
  [7] "United States"           "United Kingdom"          "United Arab Emirates"
 [10] "Ukraine"                 "Uganda"                  "Turkey"
 [13] "Tunisia"                 "Trinidad and Tobago"     "Togo"
 [16] "Thailand"                "Tanzania"                "Taiwan[m]"
 [19] "Switzerland[l]"          "Sweden"                  "Sudan"
 [22] "Sri Lanka"               "Spain"                   "South Sudan"
 [25] "South Korea"             "South Africa"            "Slovenia"
 [28] "Slovakia"                "Singapore"               "Serbia"
 [31] "Senegal"                 "Saudi Arabia"            "San Marino"
 [34] "Saint Vincent"           "Saint Lucia"             "Saint Kitts and Nevis"
 [37] "Rwanda"                  "Russia"                  "Romania"
 [40] "Qatar"                   "Portugal"                "Poland"
 [43] "Philippines"             "Peru"                    "Paraguay"
 [46] "Papua New Guinea"        "Panama"                  "Palestine"
 [49] "Pakistan"                "Oman"                    "Norway"
 [52] "Northern Cyprus[k]"      "North Macedonia"         "North Korea"
 [55] "Nigeria"                 "Niger"                   "New Zealand"
 [58] "New Caledonia"           "Netherlands"             "Nepal"
 [61] "Namibia"                 "Myanmar"                 "Mozambique"
 [64] "Morocco"                 "Montenegro"              "Mongolia"
 [67] "Moldova[j]"              "Mexico"                  "Mauritius"
 [70] "Mauritania"              "Malta"                   "Mali"
 [73] "Maldives"                "Malaysia"                "Malawi"
 [76] "Madagascar"              "Luxembourg[i]"           "Lithuania"
 [79] "Libya"                   "Liberia"                 "Lesotho"
 [82] "Lebanon"                 "Latvia"                  "Laos"
 [85] "Kyrgyzstan"              "Kuwait"                  "Kosovo"
 [88] "Kenya"                   "Kazakhstan"              "Jordan"
 [91] "Japan"                   "Jamaica"                 "Ivory Coast"
 [94] "Italy"                   "Israel"                  "Ireland"
 [97] "Iraq"                    "Iran"                    "Indonesia"
[100] "India"                   "Iceland"                 "Hungary"
[103] "Honduras"                "Haiti"                   "Guyana"
[106] "Guinea-Bissau"           "Guinea"                  "Guatemala"
[109] "Grenada"                 "Greenland"               "Greece"
[112] "Ghana"                   "Germany"                 "Georgia[h]"
[115] "Gambia"                  "Gabon"                   "France[f][g]"
[118] "Finland"                 "Fiji"                    "Faroe Islands"
[121] "Ethiopia"                "Eswatini"                "Estonia"
[124] "Equatorial Guinea"       "El Salvador"             "Egypt"
[127] "Ecuador"                 "DR Congo"                "Dominican Republic"
[130] "Dominica"                "Djibouti"                "Denmark[e]"
[133] "Czechia"                 "Cyprus[d]"               "Cuba"
[136] "Croatia"                 "Costa Rica"              "Colombia"
[139] "China[c]"                "Chile"                   "Chad"
[142] "Canada"                  "Cameroon"                "Cambodia"
[145] "Burundi"                 "Burkina Faso"            "Bulgaria"
[148] "Brunei"                  "Brazil"                  "Botswana"
[151] "Bosnia and Herzegovina"  "Bolivia"                 "Bhutan"
[154] "Benin"                   "Belize"                  "Belgium"
[157] "Belarus"                 "Barbados"                "Bangladesh"
[160] "Bahrain"                 "Bahamas"                 "Azerbaijan"
[163] "Austria"                 "Australia"               "Armenia"
[166] "Argentina"               "Antigua and Barbuda"     "Angola"
[169] "Andorra"                 "Algeria"                 "Albania"
[172] "Afghanistan"
```

TASK 7: Identify countries names with a specific pattern

The goal of task 7 is using a regular expression to find any countires start with `United`

In [46]:
```
# Use a regular expression `United.+` to find matches
utd <- grep('United.+', countries)

# Print the matched country names
countries[utd]
```

'United Arab Emirates' 'United Kingdom' 'United States'

TASK 8: Pick two countries you are interested, and then review their testing data

The goal of task 8 is to compare the COVID-19 test data between two countires, select two rows from the dataframe, and select `country` , `confirmed` , `confirmed-population-ratio` columns

In [47]:
```
# Select a subset (should be only one row) of data frame based on a selected country name and columns
covid_subset[covid_subset$country == 'United Kingdom', c('confirmed','country','confirmed.population.ratio')]

# Select a subset (should be only one row) of data frame based on a selected country name and columns
covid_subset[covid_subset$country == 'Germany', c('confirmed','country','confirmed.population.ratio')]
```

A data.frame: 1 × 3

	confirmed	country	confirmed.population.ratio
	<int>	<fct>	<dbl>
165	4248286	United Kingdom	6.3

A data.frame: 1 × 3

	confirmed	country	confirmed.population.ratio
	<int>	<fct>	<dbl>
60	2532947	Germany	3

TASK 9: Compare which one of the selected countries has a larger ratio of confirmed cases to population

The goal of task 9 is to find out which country has larger ratio of confirmed cases to population, which may indicate that country has higher COVID-19 infection risk

In [48]:
```
# Use if-else statement
# if (check which confirmed.population value is greater) {
#    print()
# } else {
#    print()
# }
if (covid_subset[165,'confirmed.population.ratio'] > covid_subset[60, 'confirmed.population.ratio']) {
    print('United Kingdom')
} else {
    print("Germany")
}
```

```
[1] "United Kingdom"
```

TASK 10: Find countries with confirmed to population ratio rate less than a threshold

The goal of task 10 is to find out which countries have the confirmed to population ratio less than 1%, it may indicate the risk of those countries are relatively low

In [49]:
```
# Get a subset of any countries with `confirmed.population.ratio` less than the threshold
low_countries <- subset(covid_subset, covid_subset$confirmed.population.ratio < 0.01)
low_countries[,2]
```

5 Jan 2021 2 Mar 2021 31 Jul 2020 1 Mar 2021 1 Mar 2021 25 Nov 2020 3 Mar 2021 18 Nov 2020 7 Mar 2021

▶ **Levels**: