Analysis of Global COVID-19 Pandemic Data

Estimated time needed: **90** minutes

Overview:

There are 10 tasks in this final project. All tasks will be graded by your peers who are also completing this assignment within the same session.

You need to submit the following the screenshot for the code and output for each task for review.

If you need to refresh your memories about specific coding details, you may refer to previous hands-on labs for code examples.

In [ ]:
```
# This lab requires 'httr' and 'rvest'packages, which are already pre-loaded into this lab environment.
# However, if you are working on your local RStudio, please uncomment the below codes and install the packages.

#install.packages("httr")
#install.packages("rvest")
```

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

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

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.

Before you write the function, you can open this public page from this

URL 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)

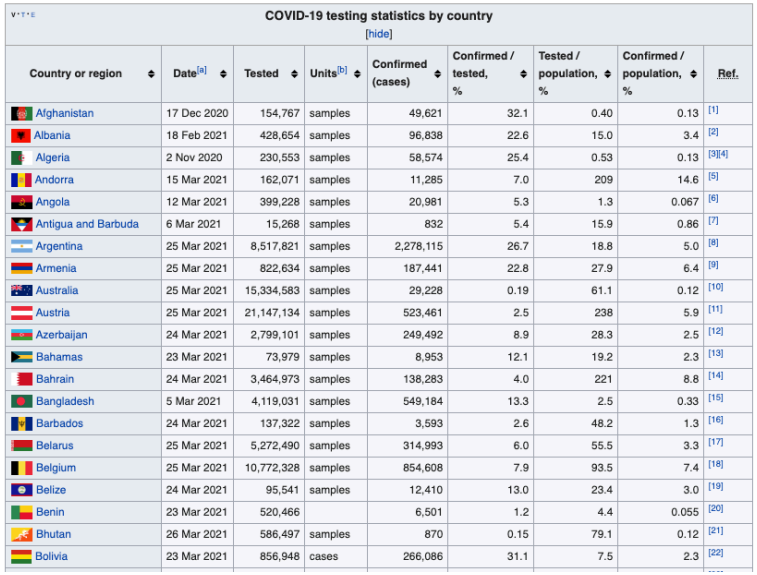
In [ ]:
```
get_wiki_covid19_page <- function() {

  # 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
  # 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

  # Use the `return` function to return the response

}
```

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

In [ ]:
```
# Call the get_wiki_covid19_page function and print the response
```

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

On the COVID-19 testing wiki page, you should see a 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]

Note the numbers you actually see on your page may be different from above because it is still an on-going pandemic when creating this notebook.

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

In [ ]:
```
# Get the root html node from the http response in task 1
```

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

In [ ]:
```
# Get the table node from the root html node
```

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`

Hint:- Please read the `table_node` with index 2(ex:- table_node[2]).

In [ ]:
```
# Read the table node and convert it into a data frame, and print the data frame for review
```

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

Let's get a summary of the data frame

In [ ]:
```
# Print the summary of the data frame
```

As you can see 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.

We have prepared a pre-processing function for you to conver the data frame but you can also try to write one by yourself

In [ ]:
```
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", "con

    # 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 [ ]:
```
# call `preprocess_covid_data_frame` function and assign it to a new data frame
```

Get the summary of the processed data frame again

In [ ]:
```
# Print the summary of the processed data frame again
```

After pre-processing, you can see 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

OK, we can call `write.csv()` function to save the csv file into a file.

In [ ]:
```
# Export the data frame to a csv file
```

Note for IBM Waston Studio, there is no traditional "hard disk" associated with a R workspace.

Even if you call `write.csv()` method to save the data frame as a csv file, it won't be shown in IBM Cloud Object Storage asset UI automatically.

However, you may still check if the `covid.csv` exists using following code snippet:

In [ ]:
```
# Get working directory
wd <- getwd()
# Get exported
file_path <- paste(wd, sep="", "/covid.csv")
# File path
print(file_path)
file.exists(file_path)
```

Optional Step: If you have difficulties finishing above webscraping tasks, you may still continue with next tasks by downloading a provided csv file from here:

In [ ]:
```
## Download a sample csv file
# covid_csv_file <- download.file("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwor
# 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 [ ]:
```
# Read covid_data_frame_csv from the csv file

# Get the 5th to 10th rows, with two "country" "confirmed" columns
```

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 [ ]:
```
# Get the total confirmed cases worldwide

# Get the total tested cases worldwide

# Get the positive ratio (confirmed / tested)
```

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 [ ]:
```
# Get the `country` column

# Check its class (should be Factor)

# Conver the country column into character so that you can easily sort them

# Sort the countries AtoZ

# Sort the countries ZtoA

# Print the sorted ZtoA list
```

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 [ ]:
```
# Use a regular expression `United.+` to find matches

# Print the matched country names
```

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, you will need to select two rows from the dataframe, and select `country`, `confirmed`, `confirmed-population-ratio` columns

In [ ]:
```
# Select a subset (should be only one row) of data frame based on a selected country name and columns


# Select a subset (should be only one row) of data frame based on a selected country name and columns
```

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 you have selected before has larger ratio of confirmed cases to population, which may indicate that country has higher COVID-19 infection risk

In [ ]:
```
# Use if-else statement
# if (check which confirmed.population value is greater) {
#    print()
# } else {
#    print()
# }
```

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 [ ]:
```
# Get a subset of any countries with `confirmed.population.ratio` less than the threshold
```