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 [16]:
```
# Print the summary of the data frame
summary(covid_data_frame)
```

```
 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
```

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

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 [10]:
```
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 [17]:
```
# call `preprocess_covid_data_frame` function and assign it to a new data frame
web_covid_data_frame<-preprocess_covid_data_frame(covid_data_frame)
web_covid_data_frame
```

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	23 Feb 2022	300,307	samples	37,958	12.6	387	49.0	[5]
Angola	2 Feb 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	16 Apr 2022	35,716,069	samples	9,060,495	25.4	78.3	20.0	[8]
Armenia	29 May 2022	3,099,602	samples	422,963	13.6	105	14.3	[9]
Australia	5 Jul 2022	74,333,211	samples	8,291,349	11.2	296	33.0	[10]
Austria	10 Jul 2022	191,372,091	samples	4,573,219	2.4	2,150	51.4	[11]
Azerbaijan	11 May 2022	6,838,458	samples	792,638	11.6	69.1	8.0	[12]
Bahamas	29 Jun 2022	242,595	samples	35,975	14.8	62.9	9.3	[13]

Get the summary of the processed data frame again

In [18]:
```
# Print the summary of the processed data frame again
```

Get the summary of the processed data frame again

In [18]:
```
# Print the summary of the processed data frame again
summary(web_covid_data_frame)
```

```
 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
```

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

In [19]:
```
# Export the data frame to a csv file
write.csv(web_covid_data_frame,file="covid.csv",row.names=FALSE)
```

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 [20]:
```
# Get working directory
wd <- getwd()
# Get exported
file_path <- paste(wd, sep="", "/covid.csv")
# File path
print(file_path)
file.exists(file_path)
```

```
[1] "/home/wsuser/work/covid.csv"
```

TRUE

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 [26]:
```
## 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_data_frame_csv <- read.csv("covid.csv", header=TRUE, sep=",")
```