In [1]:
```
require("httr")
require("rvest")

library(httr)
library(rvest)
```

```
Loading required package: httr

Loading required package: rvest
```

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

In [ ]:
```
# get_wiki_covid19_page <- function() {
  # Our target COVID-19 wiki page URL is: https://en.wikipedia.org/w/index.php?title=T
  # Which has two parts:
    # 1) base URL `https://en.wikipedia.org/w/index.php
    # 2) URL parameter: `title=Template:COVID-19_testing_by_country`, separated by que

  # Wiki page base
  wiki_base_url <- "http://web.archive.org/web/20221025155918/https://en.wikipedia.org

  # You will need to create a List which has an element called `title` to specify whic
  # in our case, it will be `Template:COVID-19_testing_by_country`
  url_param <- list(title = "Template:COVID-19_testing_by_country")

  wiki_response <- GET(wiki_base_url, query = url_param)
  return(wiki_response)
}
```

In [8]:
```
response <- get_wiki_covid19_page()
print(response)
```

```
Response [http://web.archive.org/web/20221025155918/https://en.wikipedia.org/w/index.
php?title=Template%3ACOVID-19_testing_by_country]
  Date: 2023-07-27 22:44
  Status: 200
  Content-Type: text/html; charset=UTF-8
  Size: 464 kB
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head><script type="text/javascript" src="/_static/js/bundle-playback.js?v=1W...
<script type="text/javascript" src="/_static/js/wombat.js?v=txqj7nKC" charset...
<script type="text/javascript">
  __wm.init("http://web.archive.org/web");
  __wm.wombat("https://en.wikipedia.org/w/index.php?title=Template:COVID-19_t...
              "1666713558");
</script>
<link rel="stylesheet" type="text/css" href="/_static/css/banner-styles.css?v...
...
```

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

In [13]:
```
# Get the root html node from the http response in task 1
root_node <- read_html(response)

# Get the table node from the root html node
table_node <- html_node(root_node, "table")

# Read the table node and convert it into a data frame, and print the data frame for r
covid19_df1 <- html_table(table_node)
covid19_df <- as.data.frame(covid19_df1)
covid19_df
```

A data.frame: 173 × 9

Country or region	Date[a]	Tested	Units[b]	Confirmed(cases)	Confirme
<chr>	<chr>	<chr>	<chr>	<chr>	
Afghanistan	17 Dec 2020	154,767	samples	49,621	
Albania	18 Feb 2021	428,654	samples	96,838	
Algeria	2 Nov 2020	230,553	samples	58,574	
Andorra	23 Feb 2022	300,307	samples	37,958	
Angola	2 Feb 2021	399,228	samples	20,981	
Antigua and Barbuda	6 Mar 2021	15,268	samples	832	
Argentina	16 Apr 2022	35,716,069	samples	9,060,495	
Armenia	29 May 2022	3,099,602	samples	422,963	
Australia	9 Sep 2022	78,548,492	samples	10,112,229	
Austria	21 Oct 2022	199,625,374	samples	5,392,347	
Azerbaijan	11 May 2022	6,838,458	samples	792,638	
Bahamas	13 Oct 2022	255,240	samples	37,333	
Bahrain	21 Oct 2022	10,454,512	samples	687,146	
Bangladesh	24 Jul 2021	7,417,714	samples	1,151,644	
Barbados	14 Oct 2022	770,100	samples	103,014	
Belarus	9 May 2022	13,217,569	samples	982,809	
Belgium	17 Oct 2022	35,824,675	samples	4,594,338	
Belize	8 Jun 2022	572,900	samples	60,694	
Benin	4 May 2021	595,112	samples	7,884	
Bhutan	28 Feb 2022	1,736,168	samples	12,702	
Bolivia	5 Jun 2022	4,358,669	cases	910,228	
Bosnia and Herzegovina	6 Sep 2022	1,847,988	samples	310,023	
Botswana	11 Jan 2022	2,026,898		232,432	
Brazil	19 Feb 2021	23,561,497	samples	10,081,676	
Brunei	2 Aug 2021	153,804	samples	338	
Bulgaria	23 Oct 2022	10,668,735	samples	1,274,336	
Burkina Faso	4 Mar 2021	158,777	samples	12,123	
Burundi	5 Jan 2021	90,019		884	
Cambodia	1 Aug 2021	1,812,706		77,914	
Cameroon	18 Feb 2021	942,685	samples	32,681	

Country or region	Date[a]	Tested	Units[b]	Confirmed(cases)	Confirmed
<chr>	<chr>	<chr>	<chr>	<chr>	
⋮	⋮	⋮	⋮	⋮	
Singapore	3 Aug 2021	16,206,203	samples	65,315	
Slovakia	21 Oct 2022	7,325,172	samples	1,851,160	
Slovenia	23 Oct 2022	2,761,572	samples	1,226,117	
South Africa	24 May 2021	11,378,282	cases	1,637,848	
South Korea	1 Mar 2021	6,592,010	samples	90,029	
South Sudan	26 May 2021	164,472		10,688	
Spain	1 Jul 2021	54,128,524	samples	3,821,305	
Sri Lanka	30 Mar 2021	2,384,745	samples	93,128	
Sudan	7 Jan 2021	158,804	samples	23,316	
Sweden	24 May 2021	9,996,795	samples	1,074,751	
Switzerland[l]	10 Oct 2022	22,967,458	samples	4,164,675	
Taiwan[m]	21 Oct 2022	27,331,935	samples	7,414,829	
Tanzania	18 Nov 2020	3,880		509	
Thailand	4 Mar 2021	1,579,597	cases	26,162	
Togo	7 Oct 2022	793,937		39,169	
Trinidad and Tobago	3 Jan 2022	512,730	cases	92,997	
Tunisia	23 Aug 2021	2,893,625	samples	703,732	
Turkey	2 Jul 2021	61,236,294	samples	5,435,831	
Uganda	11 Feb 2021	852,444	samples	39,979	
Ukraine	24 Nov 2021	15,648,456	samples	3,367,461	
United Arab Emirates	21 Oct 2022	193,706,348	samples	1,034,462	
United Kingdom	19 May 2022	522,526,476	samples	22,232,377	
United States	29 Jul 2022	929,349,291	samples	90,749,469	
Uruguay	16 Apr 2022	6,089,116	samples	895,592	
Uzbekistan	7 Sep 2020	2,630,000	samples	43,975	
Venezuela	30 Mar 2021	3,179,074	samples	159,149	
Vietnam	28 Aug 2022	45,772,571	samples	11,403,302	
Zambia	10 Mar 2022	3,301,860	samples	314,850	
Zimbabwe	3 Oct 2022	2,517,861	samples	257,568	

Country or region	Date[a]	Tested	Units[b]	Confirmed(cases)	Confirme
<chr>	<chr>	<chr>	<chr>	<chr>	
.mw-parser-output .reflist{font-size:90%;margin-bottom:0.5em;list-style-type:decimal}.mw-parser-output .reflist .references{font-size:100%;margin-bottom:0;list-style-type:inherit}.mw-parser-output .reflist-columns-2{column-width:30em}.mw-parser-output .reflist-columns-3{column-width:25em}.mw-parser-output .reflist-columns{margin-top:0.3em}.mw-parser-output .reflist-columns ol{margin-top:0}.mw-parser-output .reflist-columns li{page-break-inside:avoid;break-inside:avoid-column}.mw-parser-output .reflist-upper-alpha{list-style-type:upper-alpha}.mw-parser-output .reflist-upper-roman{list-style-type:upper-roman}.mw-parser-output .reflist-lower-alpha{list-style-type:lower-alpha}.mw-parser-output .reflist-lower-greek{list-style-type:lower-greek}.mw-parser-output .reflist-lower-roman{list-	.mw-parser-output .reflist{font-size:90%;margin-bottom:0.5em;list-style-type:decimal}.mw-parser-output .reflist .references{font-size:100%;margin-bottom:0;list-style-type:inherit}.mw-parser-output .reflist-columns-2{column-width:30em}.mw-parser-output .reflist-columns-3{column-width:25em}.mw-parser-output .reflist-columns{margin-top:0.3em}.mw-parser-output .reflist-columns ol{margin-top:0}.mw-parser-output .reflist-columns li{page-break-inside:avoid;break-inside:avoid-column}.mw-parser-output .reflist-upper-alpha{list-style-type:upper-alpha}.mw-parser-output .reflist-upper-roman{list-style-type:upper-roman}.mw-parser-output .reflist-lower-alpha{list-style-type:lower-alpha}.mw-parser-output .reflist-lower-greek{list-style-type:lower-greek}.mw-parser-output .reflist-lower-roman{list-	.mw-parser-output .reflist{font-size:90%;margin-bottom:0.5em;list-style-type:decimal}.mw-parser-output .reflist .references{font-size:100%;margin-bottom:0;list-style-type:inherit}.mw-parser-output .reflist-columns-2{column-width:30em}.mw-parser-output .reflist-columns-3{column-width:25em}.mw-parser-output .reflist-columns{margin-top:0.3em}.mw-parser-output .reflist-columns ol{margin-top:0}.mw-parser-output .reflist-columns li{page-break-inside:avoid;break-inside:avoid-column}.mw-parser-output .reflist-upper-alpha{list-style-type:upper-alpha}.mw-parser-output .reflist-upper-roman{list-style-type:upper-roman}.mw-parser-output .reflist-lower-alpha{list-style-type:lower-alpha}.mw-parser-output .reflist-lower-greek{list-style-type:lower-greek}.mw-parser-output .reflist-lower-roman{list-	.mw-parser-output .reflist{font-size:90%;margin-bottom:0.5em;list-style-type:decimal}.mw-parser-output .reflist .references{font-size:100%;margin-bottom:0;list-style-type:inherit}.mw-parser-output .reflist-columns-2{column-width:30em}.mw-parser-output .reflist-columns-3{column-width:25em}.mw-parser-output .reflist-columns{margin-top:0.3em}.mw-parser-output .reflist-columns ol{margin-top:0}.mw-parser-output .reflist-columns li{page-break-inside:avoid;break-inside:avoid-column}.mw-parser-output .reflist-upper-alpha{list-style-type:upper-alpha}.mw-parser-output .reflist-upper-roman{list-style-type:upper-roman}.mw-parser-output .reflist-lower-alpha{list-style-type:lower-alpha}.mw-parser-output .reflist-lower-greek{list-style-type:lower-greek}.mw-parser-output .reflist-lower-roman{list-	.mw-parser-output .reflist{font-size:90%;margin-bottom:0.5em;list-style-type:decimal}.mw-parser-output .reflist .references{font-size:100%;margin-bottom:0;list-style-type:inherit}.mw-parser-output .reflist-columns-2{column-width:30em}.mw-parser-output .reflist-columns-3{column-width:25em}.mw-parser-output .reflist-columns{margin-top:0.3em}.mw-parser-output .reflist-columns ol{margin-top:0}.mw-parser-output .reflist-columns li{page-break-inside:avoid;break-inside:avoid-column}.mw-parser-output .reflist-upper-alpha{list-style-type:upper-alpha}.mw-parser-output .reflist-upper-roman{list-style-type:upper-roman}.mw-parser-output .reflist-lower-alpha{list-style-type:lower-alpha}.mw-parser-output .reflist-lower-greek{list-style-type:lower-greek}.mw-parser-output .reflist-lower-roman{list-	.mw-pa size:9 bottom type:de parser-ou .refer size:10 bottom type:i parser-ou columns width parser-ou columns width parser-ou colum top: parser-ou columns top:0}. ou colum inside:a in column}. output .re alph alpha}. output .re roma roman}. output .re alph alpha}. output .re gree greek}. output .re roma type:lowe Local some co unclear w report cases. tes recorded and two

Country or region	Date[a]	Tested	Units[b]	Confirmed(cases)	Confirme
<chr>	<chr>	<chr>	<chr>	<chr>	
style-type:lower-roman} ^ Local time. ^ For some countries it is unclear whether they report samples or cases. One person tested twice is recorded as one case and two samples. ^ Excluding Taiwan. ^ Excluding Northern Cyprus. ^ Excluding Greenland and the Faroe Islands. ^ Excluding Overseas France. ^ Testing data from 4 May to 12 May is missing because of the transition to the new reporting system SI-DEP. ^ Excluding Abkhazia and South Ossetia. ^ Data for residents only. ^ Excluding Transnistria. ^ Northern Cyprus is not recognized as a sovereign state by any country except Turkey. ^ Includes data for Liechtenstein. ^ Not a United Nations member	style-type:lower-roman} ^ Local time. ^ For some countries it is unclear whether they report samples or cases. One person tested twice is recorded as one case and two samples. ^ Excluding Taiwan. ^ Excluding Northern Cyprus. ^ Excluding Greenland and the Faroe Islands. ^ Excluding Overseas France. ^ Testing data from 4 May to 12 May is missing because of the transition to the new reporting system SI-DEP. ^ Excluding Abkhazia and South Ossetia. ^ Data for residents only. ^ Excluding Transnistria. ^ Northern Cyprus is not recognized as a sovereign state by any country except Turkey. ^ Includes data for Liechtenstein. ^ Not a United Nations member	style-type:lower-roman} ^ Local time. ^ For some countries it is unclear whether they report samples or cases. One person tested twice is recorded as one case and two samples. ^ Excluding Taiwan. ^ Excluding Northern Cyprus. ^ Excluding Greenland and the Faroe Islands. ^ Excluding Overseas France. ^ Testing data from 4 May to 12 May is missing because of the transition to the new reporting system SI-DEP. ^ Excluding Abkhazia and South Ossetia. ^ Data for residents only. ^ Excluding Transnistria. ^ Northern Cyprus is not recognized as a sovereign state by any country except Turkey. ^ Includes data for Liechtenstein. ^ Not a United Nations member	style-type:lower-roman} ^ Local time. ^ For some countries it is unclear whether they report samples or cases. One person tested twice is recorded as one case and two samples. ^ Excluding Taiwan. ^ Excluding Northern Cyprus. ^ Excluding Greenland and the Faroe Islands. ^ Excluding Overseas France. ^ Testing data from 4 May to 12 May is missing because of the transition to the new reporting system SI-DEP. ^ Excluding Abkhazia and South Ossetia. ^ Data for residents only. ^ Excluding Transnistria. ^ Northern Cyprus is not recognized as a sovereign state by any country except Turkey. ^ Includes data for Liechtenstein. ^ Not a United Nations member	style-type:lower-roman} ^ Local time. ^ For some countries it is unclear whether they report samples or cases. One person tested twice is recorded as one case and two samples. ^ Excluding Taiwan. ^ Excluding Northern Cyprus. ^ Excluding Greenland and the Faroe Islands. ^ Excluding Overseas France. ^ Testing data from 4 May to 12 May is missing because of the transition to the new reporting system SI-DEP. ^ Excluding Abkhazia and South Ossetia. ^ Data for residents only. ^ Excluding Transnistria. ^ Northern Cyprus is not recognized as a sovereign state by any country except Turkey. ^ Includes data for Liechtenstein. ^ Not a United Nations member	Excludin Excludir Cyprus. Greenl Faro Excludir France. ^ T from 4 Ma bec transition reporting DEP. Abkhazia Ossetia resid Excluding ^ Northe not recc sovere any cou Turkey Liechtens a Uni

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

In [14]:
```
# Print the summary of the data frame
summary(covid19_df)
```

```
 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
```

In [ ]:
```
As you can see from the summary, the columns names are little bit different to underst
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 rem
renaming columns, and convert columns into proper data types.
```

In [21]:
```
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

    # 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$'confirmed.tested.ratio' <- as.numeric(gsub(",","",data_frame$`confirme
    data_frame$'tested.population.ratio' <- as.numeric(gsub(",","",data_frame$`tested.
    data_frame$'confirmed.population.ratio' <- as.numeric(gsub(",","",data_frame$`conf

    return(data_frame)
}
```

In [25]:
```
# call `preprocess_covid_data_frame` function and assign it to a new data frame

preprocessed_df <- preprocess_covid_data_frame(covid19_df)

preprocessed_df
```

A data.frame: 172 × 7

	country	date	tested	confirmed	confirmed.tested.ratio	tested.population.ratio	confir
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	
1	Afghanistan	17 Dec 2020	154767	49621	32.10	0.40	
2	Albania	18 Feb 2021	428654	96838	22.60	15.00	
3	Algeria	2 Nov 2020	230553	58574	25.40	0.53	
4	Andorra	23 Feb 2022	300307	37958	12.60	387.00	
5	Angola	2 Feb 2021	399228	20981	5.30	1.30	
6	Antigua and Barbuda	6 Mar 2021	15268	832	5.40	15.90	
7	Argentina	16 Apr 2022	35716069	9060495	25.40	78.30	
8	Armenia	29 May 2022	3099602	422963	13.60	105.00	
9	Australia	9 Sep 2022	78548492	10112229	12.90	313.00	
10	Austria	21 Oct 2022	199625374	5392347	2.70	2242.00	
11	Azerbaijan	11 May 2022	6838458	792638	11.60	69.10	
12	Bahamas	13 Oct 2022	255240	37333	14.60	66.20	
13	Bahrain	21 Oct 2022	10454512	687146	6.60	666.00	
14	Bangladesh	24 Jul 2021	7417714	1151644	15.50	4.50	
15	Barbados	14 Oct 2022	770100	103014	13.40	268.00	
16	Belarus	9 May 2022	13217569	982809	7.40	139.00	

	country	date	tested	confirmed	confirmed.tested.ratio	tested.population.ratio	confir
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	
17	Belgium	17 Oct 2022	35824675	4594338	12.80	311.00	
18	Belize	8 Jun 2022	572900	60694	10.60	140.00	
19	Benin	4 May 2021	595112	7884	1.30	5.10	
20	Bhutan	28 Feb 2022	1736168	12702	0.73	234.00	
21	Bolivia	5 Jun 2022	4358669	910228	20.90	38.10	
22	Bosnia and Herzegovina	6 Sep 2022	1847988	310023	16.80	54.00	
23	Botswana	11 Jan 2022	2026898	232432	11.50	89.90	
24	Brazil	19 Feb 2021	23561497	10081676	42.80	11.20	
25	Brunei	2 Aug 2021	153804	338	0.22	33.50	
26	Bulgaria	23 Oct 2022	10668735	1274336	11.90	154.00	
27	Burkina Faso	4 Mar 2021	158777	12123	7.60	0.76	
28	Burundi	5 Jan 2021	90019	884	0.98	0.76	
29	Cambodia	1 Aug 2021	1812706	77914	4.30	11.20	
30	Cameroon	18 Feb 2021	942685	32681	3.50	3.60	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
143	Serbia	21 Oct 2022	11285164	2395887	21.20	162.0000	
144	Singapore	3 Aug 2021	16206203	65315	0.40	284.0000	
145	Slovakia	21 Oct 2022	7325172	1851160	25.30	134.0000	

	country	date	tested	confirmed	confirmed.tested.ratio	tested.population.ratio	confir
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	
146	Slovenia	23 Oct 2022	2761572	1226117	44.40	132.0000	
147	South Africa	24 May 2021	11378282	1637848	14.40	19.2000	
148	South Korea	1 Mar 2021	6592010	90029	1.40	12.7000	
149	South Sudan	26 May 2021	164472	10688	6.50	1.3000	
150	Spain	1 Jul 2021	54128524	3821305	7.10	116.0000	
151	Sri Lanka	30 Mar 2021	2384745	93128	3.90	10.9000	
152	Sudan	7 Jan 2021	158804	23316	14.70	0.3600	
153	Sweden	24 May 2021	9996795	1074751	10.80	96.8000	
154	Switzerland[l]	10 Oct 2022	22967458	4164675	18.10	267.0000	
155	Taiwan[m]	21 Oct 2022	27331935	7414829	27.13	115.8000	
156	Tanzania	18 Nov 2020	3880	509	13.10	0.0065	
157	Thailand	4 Mar 2021	1579597	26162	1.70	2.3000	
158	Togo	7 Oct 2022	793937	39169	4.90	9.2000	
159	Trinidad and Tobago	3 Jan 2022	512730	92997	18.10	37.6000	
160	Tunisia	23 Aug 2021	2893625	703732	24.30	24.5000	
161	Turkey	2 Jul 2021	61236294	5435831	8.90	73.6000	
162	Uganda	11 Feb 2021	852444	39979	4.70	1.9000	

	country	date	tested	confirmed	confirmed.tested.ratio	tested.population.ratio	confir
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	
163	Ukraine	24 Nov 2021	15648456	3367461	21.50	37.2000	
164	United Arab Emirates	21 Oct 2022	193706348	1034462	0.53	2015.0000	
165	United Kingdom	19 May 2022	522526476	22232377	4.30	774.0000	
166	United States	29 Jul 2022	929349291	90749469	9.80	281.0000	
167	Uruguay	16 Apr 2022	6089116	895592	14.70	175.0000	
168	Uzbekistan	7 Sep 2020	2630000	43975	1.70	7.7000	
169	Venezuela	30 Mar 2021	3179074	159149	5.00	11.0000	
170	Vietnam	28 Aug 2022	45772571	11403302	24.90	46.4000	
171	Zambia	10 Mar 2022	3301860	314850	9.50	19.0000	
172	Zimbabwe	3 Oct 2022	2517861	257568	10.20	16.9000	

In [26]:
```
# Print the summary of the processed data frame again
summary(preprocessed_df)
```

```
                country             date         tested
 Afghanistan        :  1   21 Oct 2022: 13   Min.   :     3880
 Albania            :  1   20 Oct 2022:  5   1st Qu.:   512037
 Algeria            :  1   1 Mar 2021 :  3   Median :  3029859
 Andorra            :  1   15 Oct 2022:  3   Mean   : 31057082
 Angola             :  1   16 Oct 2022:  3   3rd Qu.: 11867328
 Antigua and Barbuda:  1   23 Jul 2021:  3   Max.   :929349291
 (Other)            :166   (Other)    :142
   confirmed        confirmed.tested.ratio tested.population.ratio
 Min.   :       0   Min.   :  0.00         Min.   :   0.0065
 1st Qu.:   37802   1st Qu.:  5.00         1st Qu.:   9.4250
 Median :  281196   Median : 10.05         Median :  46.9500
 Mean   : 2467072   Mean   : 12.15         Mean   : 172.5734
 3rd Qu.: 1249614   3rd Qu.: 15.25         3rd Qu.: 152.5000
 Max.   :90749469   Max.   :185.30         Max.   :3098.0000

 confirmed.population.ratio
 Min.   :  0.000
 1st Qu.:  0.425
 Median :  6.100
 Mean   : 13.011
 3rd Qu.: 16.725
 Max.   :113.900
```

In [ ]:
```
After pre-processing, you can see the columns and columns names are simplified, and co
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

In [27]:
```
# Export the data frame to a csv file
write.csv(preprocessed_df, "covid_data.csv")
```

In [28]:
```
# check file path for csv created
# Get working directory
wd <- getwd()
# Get exported
file_path <- paste(wd, sep="", "/covid_data.csv")
# File path
print(file_path)
file.exists(file_path)
```

```
[1] "C:/Users/nihar/covid_data.csv"
```
TRUE

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 [32]:
```
# Read covid_data_frame_csv from the csv file
df <- read.csv("covid_data.csv")

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

A data.frame: 6 × 2

	country	confirmed
	<chr>	<int>
5	Angola	20981
6	Antigua and Barbuda	832
7	Argentina	9060495
8	Armenia	422963
9	Australia	10112229
10	Austria	5392347

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 [40]:
```
# Get the total confirmed cases worldwide
Total_confirmed  <- sum(df[5])
Total_confirmed
```

424336298

In [41]:
```
# Get the total tested cases worldwide
Total_tested <- sum(df$tested)
Total_tested
```

5341818173

In [43]:
```
# Get the positive ratio (confirmed / tested)
positive_ratio = Total_confirmed/Total_tested
positive_ratio
```

0.0794366794708196

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 [46]:
```r
# Get the `country` column

country_col <- df$country

# Check its class
class(country_col)

# Convert the country column into character so that you can easily sort them
country_col <- as.character(country_col)
```

'character'

In [47]:
```r
# Sort the countries AtoZ

sorted_countries <- sort(country_col)
```

In [48]:
```r
# Sort the countries ZtoA
sorted_countries_desc <- sort(country_col, decreasing = TRUE )

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

```
  [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 [51]:
```
# Use a regular expression `United.+` to find matches


matches <- grep("United.+", df$country)
matches
```

164 · 165 · 166

In [54]:
```
# Print the matched country names

for (i in matches)
{print(df$country[i])}
```

```
[1] "United Arab Emirates"
[1] "United Kingdom"
[1] "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, you will need to select two rows from the dataframe, and select `country`, `confirmed`, `confirmed-population-ratio` columns

In [59]:
```
# Select a subset (should be only one row) of data frame based on a selected country r

US_cases <- df[df$country == 'United States', c('confirmed','country','confirmed.popul
US_cases
```

A data.frame: 1 × 3

	confirmed	country	confirmed.population.ratio
	<int>	<chr>	<dbl>
166	90749469	United States	27.4

In [60]:
```
# Select a subset (should be only one row) of data frame based on a selected country r

Japan_cases <- df[df$country == 'Japan', c('confirmed','country','confirmed.populatior
Japan_cases
```

A data.frame: 1 × 3

	confirmed	country	confirmed.population.ratio
	<int>	<chr>	<dbl>
82	432773	Japan	0.34

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 [65]:
```
# Use if-else statement

if (US_cases$confirmed.population.ratio > Japan_cases$confirmed.population.ratio)
   {
    print("Unites States has larger ratio of COVID confirmed cases than Japan" )
   } else {
    print("Japan has larger ratio of COVID confirmed casesthan United States")
   }
```

```
[1] "Unites States has larger ratio of COVID confirmed cases than Japan"
```

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 [79]:
```
# Get a subset of any countries with `confirmed.population.ratio` less than the thresh

low_risk <- df[df$confirmed.population.ratio < 1]

low_risk_countries <- low_risk$country
low_risk_countries
```

'Afghanistan' · 'Algeria' · 'Angola' · 'Antigua and Barbuda' · 'Bangladesh' · 'Benin' · 'Brunei' · 'Burkina Faso' · 'Burundi' · 'Cambodia' · 'Cameroon' · 'Chad' · 'China[c]' · 'DR Congo' · 'Egypt' · 'Ethiopia' · 'Gabon' · 'Gambia' · 'Ghana' · 'Grenada' · 'Guinea' · 'Guinea-Bissau' · 'Haiti' · 'Ivory Coast' · 'Japan' · 'Kenya' · 'Laos' · 'Liberia' · 'Madagascar' · 'Malawi' · 'Mali' · 'Mauritania' · 'Mauritius' · 'Mozambique' · 'Myanmar' · 'New Caledonia' · 'Niger' · 'Nigeria' · 'North Korea' · 'Pakistan' · 'Papua New Guinea' · 'Rwanda' · 'Senegal' · 'South Korea' · 'South Sudan' · 'Sri Lanka' · 'Sudan' · 'Tanzania' · 'Thailand' · 'Togo' · 'Uganda' · 'Uzbekistan' · 'Venezuela'

In [ ]: