Announcement

P8101S20-Homework_08

Neal Kar

3/31/2020

Announcement

Please do not add code folding (code_folding: hide) to your YAML Header or echo = FALSE to your RMD code chunk options. In order to accurately grade your HTML files we need to be able to see all of your code. Thank you!

Question 1

For this homework we will be exploring and visualizing the Novel Corona Virus (COVID-19) epidemiological data set (stored in `data/time_series_2019-ncov-Confirmed.csv`). This is a real dataset compiled and updated daily by the Johns Hopkins University Center for Systems Science and Engineering (JHU CCSE) from various sources. You can learn more about this dataset at https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases (https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases)

a)

This data is in wide format, reporting cumulative confirmed cases each day per region. In order to graph and analyze it, first transform the data into long format. Once the data is in long format, also use janitor::clean_names() to make the column names easier to use.

```
ncov_df_wide<-read_csv("data/time_series_2019-ncov-Confirmed.cs
v")

ncov_df_long <- pivot_longer(ncov_df_wide,
                             cols = contains("/20"),
                             names_to = "date",
                             values_to = "cumulative_cases")

library(janitor)

ncov_df_long <- ncov_df_long %>%
  clean_names()
```

b)

Look at the data from China, Australia, Canada, and the US. What do you notice about the way the data is organized compared to the data from Italy, Iran, or Japan? If you wanted to create a summary of the total number of cases per day per country,

what would you need to group_by()?.

The data from China, Australia, Canada, and the U.S. is organized by state or province. By contrast, the data from Italy, Iran, and Japan are only reported at the national level (i.e. not organized by state or province). I would need to group by country/region and by date to create a summary of daily cases by country.

c)

After pivoting longer in (a) you should have a dataset with a date variable. These variables are stored as string / character variables in the format "MM/DD/YY". In order to correctly treat them as dates, we need to transform them into a "Date" object. The `lubridate` package from the tidyverse allows us to easily handle this problem. Create a new corrected date variable using `mutate()` and the `mdy()` function from `lubridate`.

```
ncov_df_long <- ncov_df_long %>%
  mutate(date = mdy(date))
```

d)

Now that you have a corrected date variable, create a dataset that summarizes the total number of cumulative cases per day per country.

```
ncov_df_daily_ctry <- ncov_df_long %>%
  group_by(country_region, date) %>%
  summarize(cumulative_cases = sum(cumulative_cases))
```

e)

Create a summary table of total cases per country. This table should only include the country name and the total number of coronavirus cases. Since this is a large table, use the flexible HTML datatable from the DT package to present it. To do this, just run the `datatable()` *function on your table.*

```
ncov_df_by_ctry <- ncov_df_daily_ctry %>%
  group_by(country_region) %>%
  summarize(total_cases = max(cumulative_cases))

datatable(ncov_df_by_ctry)
```

Show 10 entries Search:

	country_region	total_cases
1	Afghanistan	24
2	Albania	70
3	Algeria	90
4	Andorra	75

	country_region	total_cases
5	Angola	1
6	Antigua and Barbuda	1
7	Argentina	128
8	Armenia	136
9	Australia	791
10	Austria	2388

Showing 1 to 10 of 162 entries

Previous 1 2 3 4 5 … 17 Next

f)

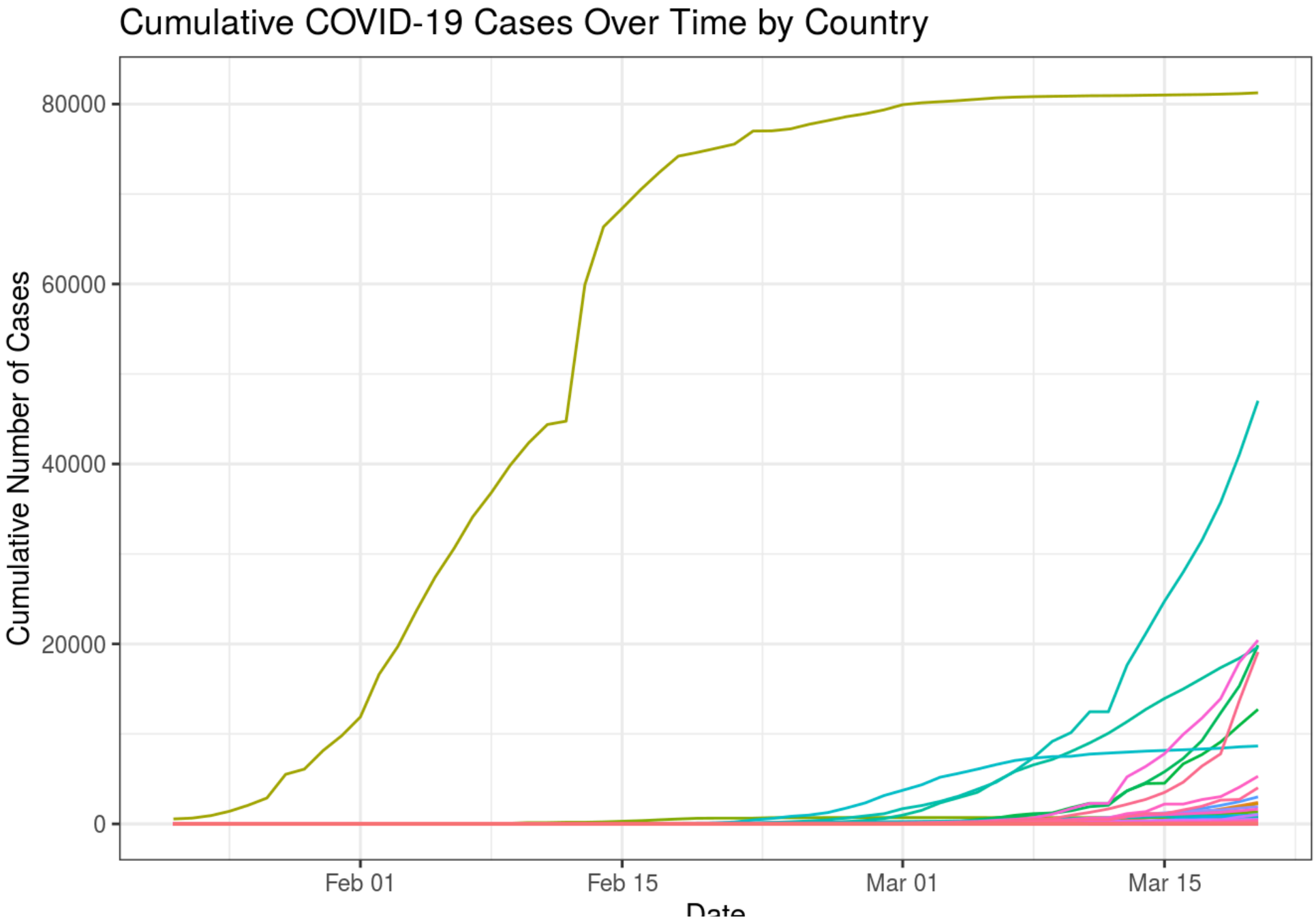
Using the search tool from the datatable you just created, find out how many total confirmed cases there are in Denmark, Italy, Mexico, Iran, and the US. Report these numbers in a sentence (no need for inline coding here).

Denmark: 1,337; Italy: 47,021; Mexico: 164; Iran: 19,644; U.S.: 19,100.

g)

Create a line plot of cumulative cases over time, with different colored lines for each country. There are 162 countries, so remove the legend from your plot (it takes up too much room). Make sure the background of your graph is white and that there is a title for your graph and the x and y axes.

```
ggplot(data = ncov_df_daily_ctry) +
  geom_line(aes(x=date, y=cumulative_cases,
                color=country_region)) +
  theme_bw() +
  theme(legend.position = "none") +
  labs(title="Cumulative COVID-19 Cases Over Time by Country",
       x="Date", y="Cumulative Number of Cases")
```

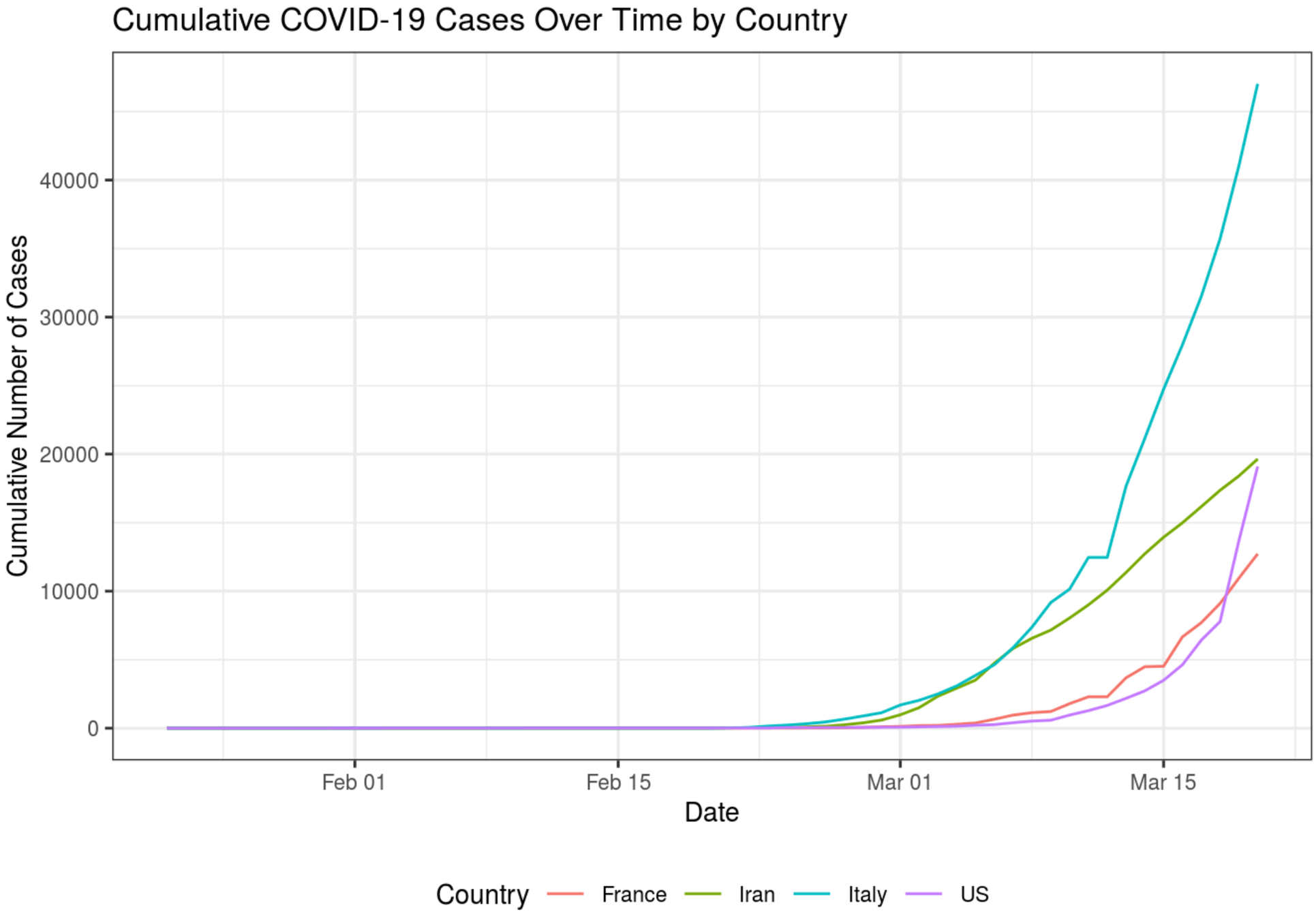

Cumulative COVID-19 Cases Over Time by Country
Cumulative Number of Cases
80000
60000
40000
20000
0
Feb 01
Feb 15
Mar 01
Mar 15
Date

Date

h)

Subset the daily case by country data to just include the following countries: Italy, Iran, the US, and France. Create a new line graph of just these countries. Include a legend at the bottom of the plot, and make sure you have a white background and titles similar to your plot in (g)

```
ncov_df_ctry_filtered <- ncov_df_daily_ctry %>%
  filter(country_region == "Italy" |
           country_region == "Iran" |
           country_region == "US" |
           country_region == "France")

ggplot(data = ncov_df_ctry_filtered) +
  geom_line(aes(x=date, y=cumulative_cases,
                color=country_region)) +
  theme_bw() +
  theme(legend.position = "bottom",
        text = element_text(size=10)) +
  labs(title="Cumulative COVID-19 Cases Over Time by Country",
       x="Date", y="Cumulative Number of Cases",
       color="Country")
```


Cumulative COVID-19 Cases Over Time by Country
Cumulative Number of Cases
40000
30000
20000
10000
0
Feb 01
Feb 15
Mar 01
Mar 15
Date
Country
France
Iran
Italy
US

i)

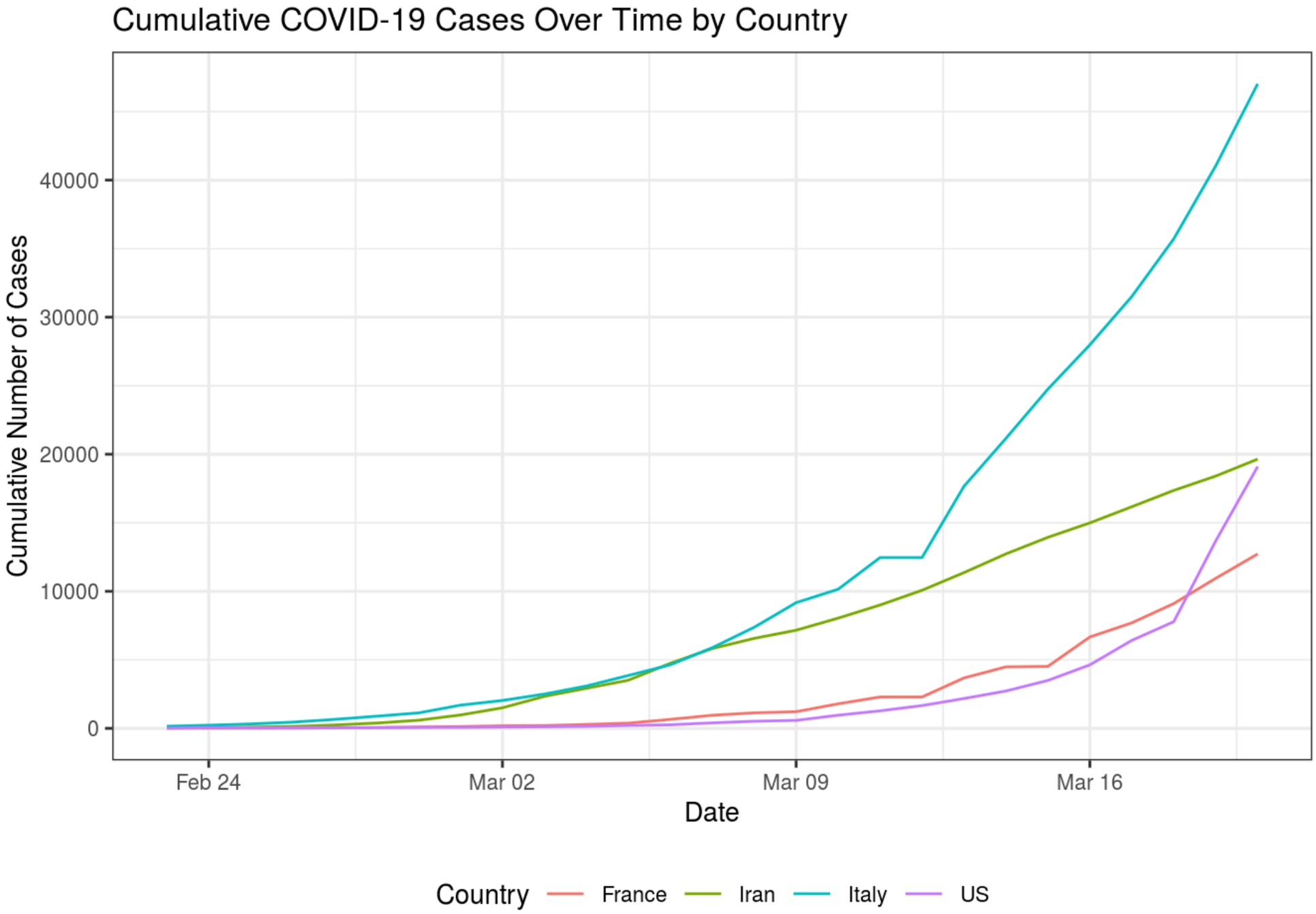
Filter the dataset in (h) so that it only includes dates after February 22, 2020. Hint: you can do this by first creating a variable that represents February 22, 2020 using the mdy() function. Plot this filtered dataset and answer the following questions based on it:

(1) Which of these four countries has the greatest number of confirmed cases?

(2) Which of these four countries do you think has the most consistent increase in cases?

(3) Which of these four countries do you think has the steepest increase in cases?

```
ncov_df_ctry_date_filtered <- ncov_df_ctry_filtered %>%
  filter(date > "2020-02-22")

ggplot(data = ncov_df_ctry_date_filtered) +
  geom_line(aes(x=date, y=cumulative_cases,
                color=country_region)) +
  theme_bw() +
  theme(legend.position = "bottom",
        text = element_text(size=10)) +
  labs(title="Cumulative COVID-19 Cases Over Time by Country",
       x="Date", y="Cumulative Number of Cases",
       color="Country")
```

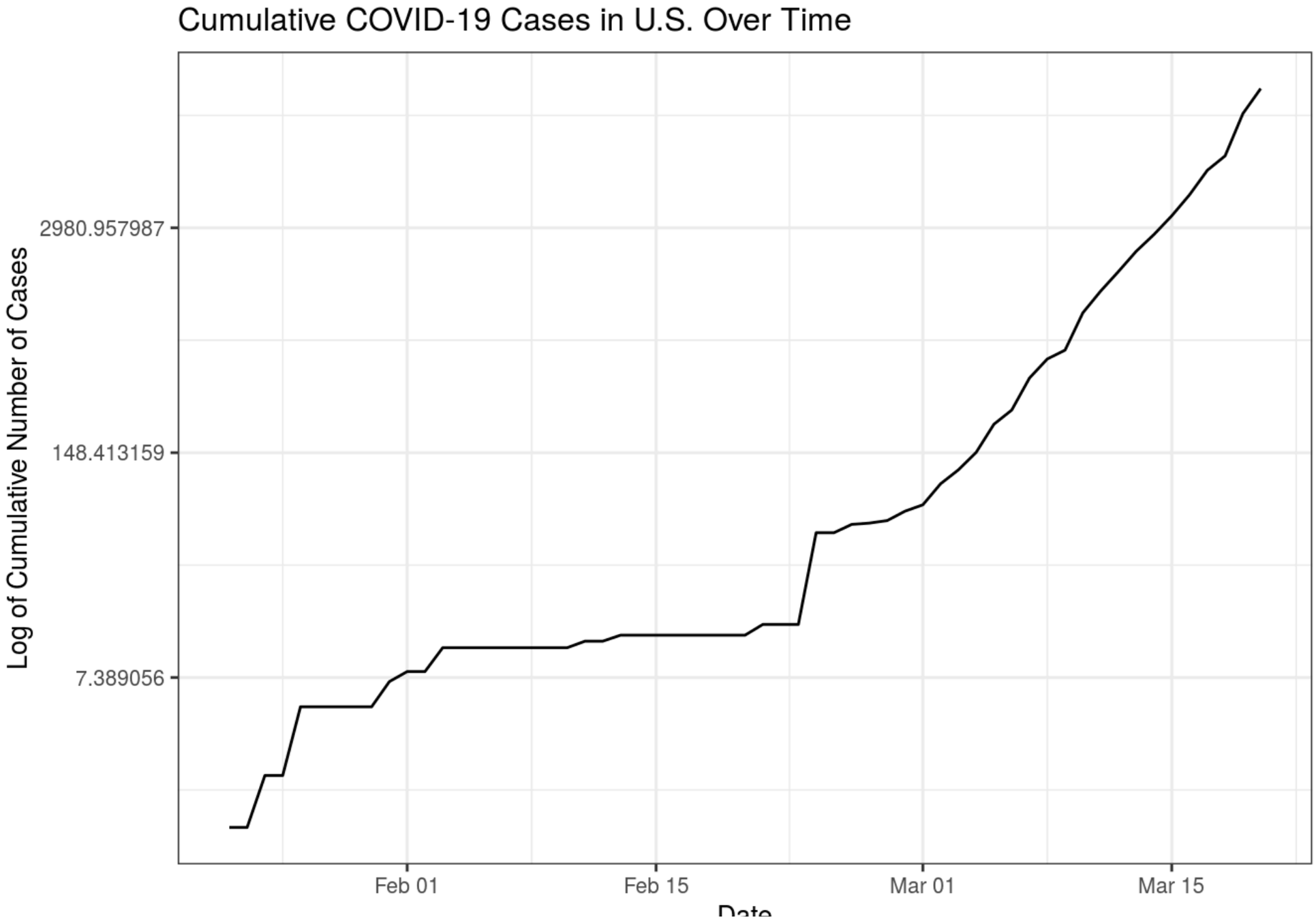

Cumulative COVID-19 Cases Over Time by Country
Cumulative Number of Cases
40000
30000
20000
10000
0
Feb 24
Mar 02
Mar 09
Mar 16
Date
Country
France
Iran
Italy
US

Italy has the greatest number of confirmed cases. Iran seems to have the most consistent increase in cases. Italy seems to have the steepest increase in confirmed cases.

j)

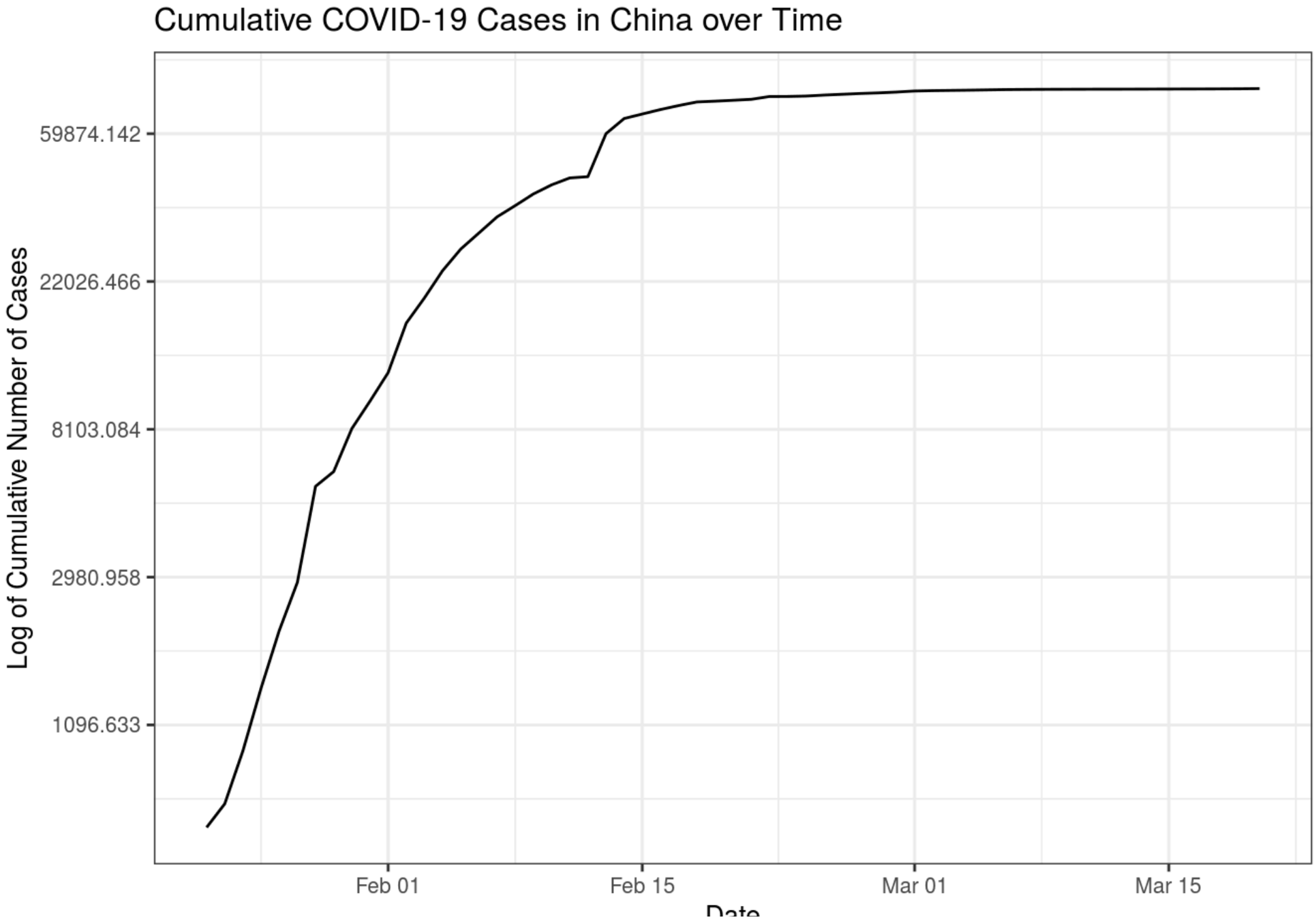
Due to the exponential increase in cases, it is easier to visualize the trend in confirmed coronavirus cases by performing a log transformation on the number of cases. This can be done by adding a `scale_y_continuous(trans = "log")` *statement to your ggplot. Using this method, create a log-transformed confirmed case plot for the United States and a similar plot for China. Make sure your plot has a title indicating the country you are plotting and labels for the x and y axes.*

```
#Create filtered dataframe for U.S.
ncov_df_US <- ncov_df_daily_ctry %>%
  filter(country_region == "US")

#Create filtered dataframe for China
ncov_df_China <- ncov_df_daily_ctry %>%
  filter(country_region == "China")

#ggplot for U.S.
ggplot(data = ncov_df_US) +
  geom_line(aes(x=date, y=cumulative_cases)) +
  theme_bw() +
  theme(text = element_text(size=10)) +
  scale_y_continuous(trans = "log") +
  labs(title="Cumulative COVID-19 Cases in U.S. Over Time",
       x="Date", y="Log of Cumulative Number of Cases")
```


Cumulative COVID-19 Cases in U.S. Over Time
Log of Cumulative Number of Cases
2980.957987
148.413159
7.389056
Feb 01
Feb 15
Mar 01
Mar 15
Date

Date

```
#ggplot for China
ggplot(data = ncov_df_China) +
  geom_line(aes(x=date, y=cumulative_cases)) +
  theme_bw() +
  theme(text = element_text(size=10)) +
  scale_y_continuous(trans = "log") +
  labs(title="Cumulative COVID-19 Cases in China over Time",
       x="Date", y="Log of Cumulative Number of Cases")
```


Cumulative COVID-19 Cases in China over Time
Log of Cumulative Number of Cases
59874.142
22026.466
8103.084
2980.958
1096.633
Feb 01
Feb 15
Mar 01
Mar 15
Date

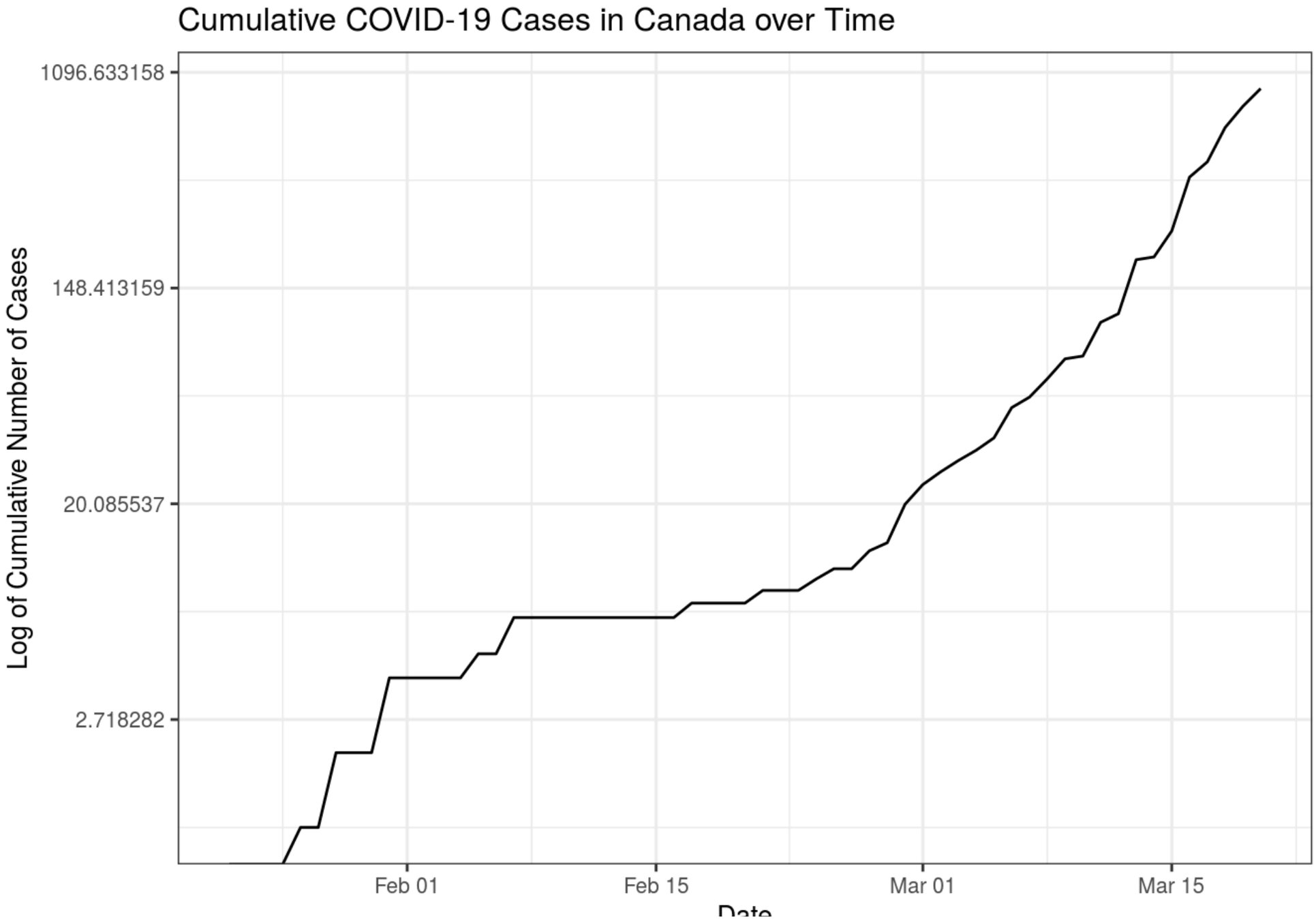
Date

k)

Create a function using your code from part (j). Your function should take a string input that is a country name, and should output a plot of log-transformed confirmed cases over time for that country. Make sure the plot has a title indicating the country you are plotting and labels for the x and y axes. Test your function by creating graphs for 3 countries of your choice.

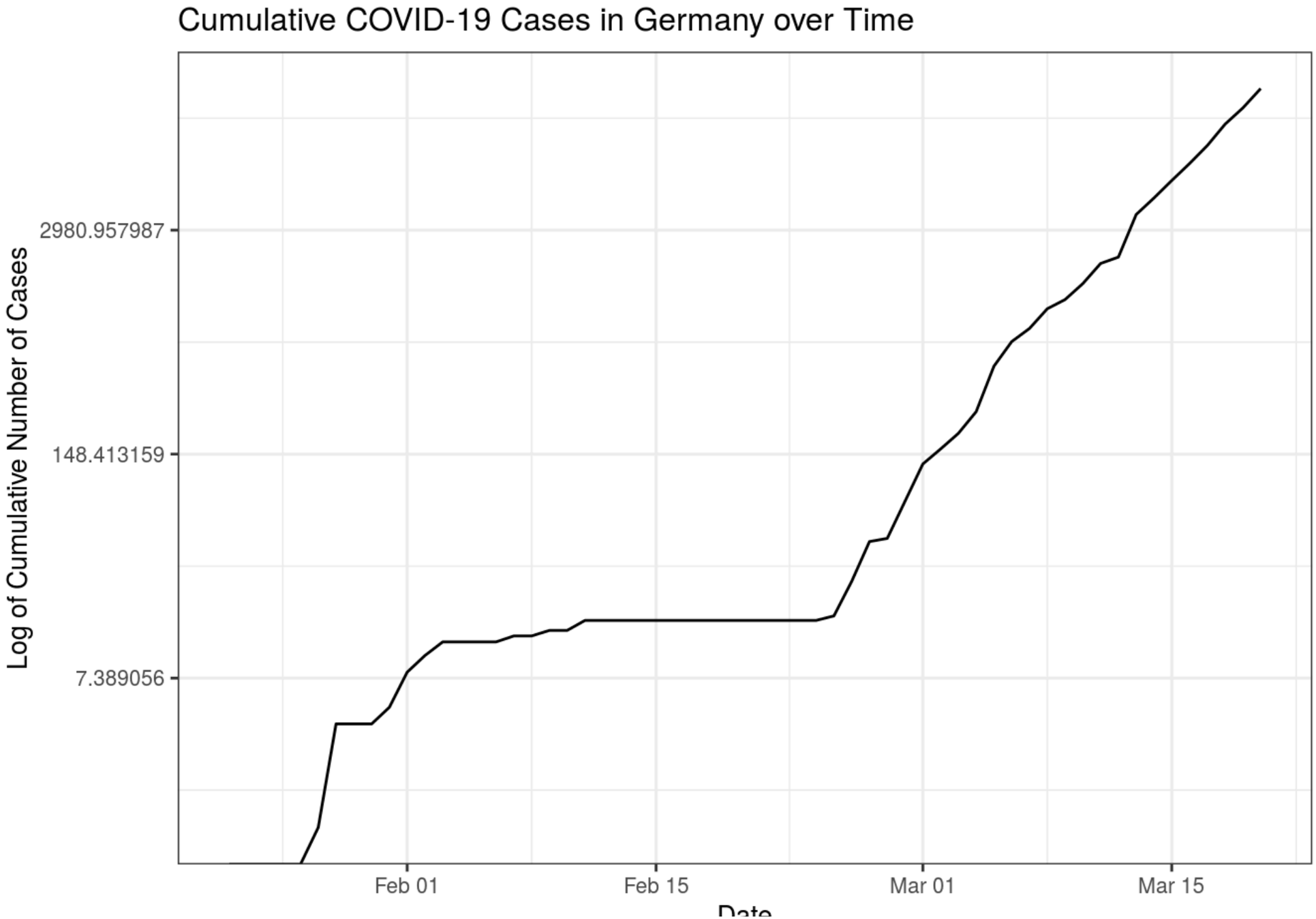
```r
### Create the function
plot_ncov_ctry <- function(country) {

  #Temporarily create filtered dataframe for country
  df <- ncov_df_daily_ctry %>%
    filter(country_region == country)

  #Create ggplot (store it as an object)
  plot <- ggplot(data = df) +
    geom_line(aes(x=date, y=cumulative_cases)) +
    theme_bw() +
    theme(text = element_text(size=10)) +
    scale_y_continuous(trans = "log") +
    labs(title = paste("Cumulative COVID-19 Cases in", country,
    "over Time"), x="Date",
    y="Log of Cumulative Number of Cases")

  #Return the plot so that the graph is displayed
  return(plot)
```

```
}

#Test the function

#Country 1 = Canada
plot_ncov_ctry("Canada")
```


Cumulative COVID-19 Cases in Canada over Time
1096.633158
148.413159
20.085537
2.718282
Log of Cumulative Number of Cases
Feb 01
Feb 15
Mar 01
Mar 15
Date

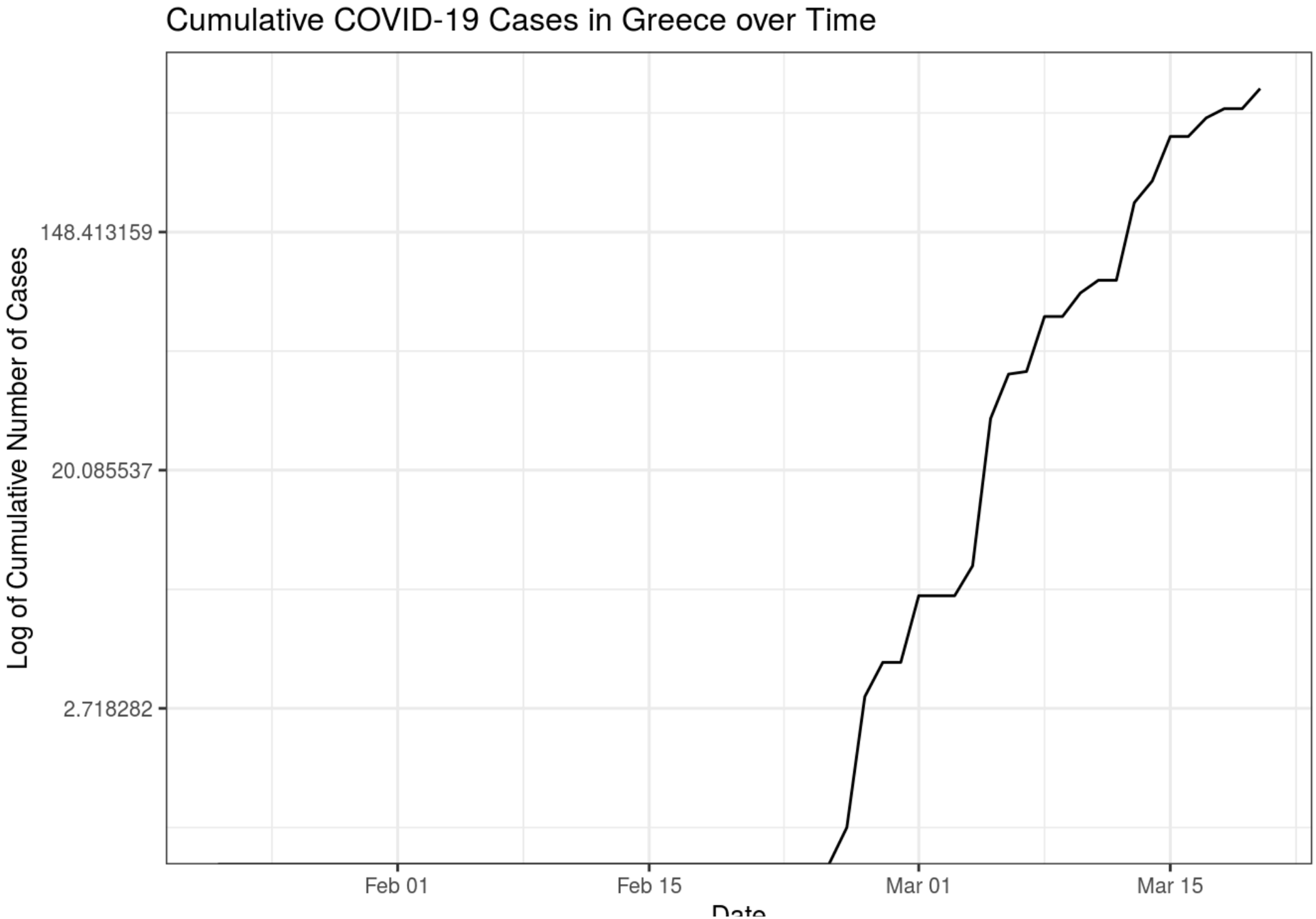
Date

```
#Country 2 = Germany
plot_ncov_ctry("Germany")
```


Cumulative COVID-19 Cases in Germany over Time
Log of Cumulative Number of Cases
2980.957987
148.413159
7.389056
Feb 01
Feb 15
Mar 01
Mar 15
Date

Date

```
#Country 3 = Greece
plot_ncov_ctry("Greece")
```


Cumulative COVID-19 Cases in Greece over Time
Log of Cumulative Number of Cases
148.413159
20.085537
2.718282
Feb 01
Feb 15
Mar 01
Mar 15
Date

Date

Question 2

We obtained the confirmed case dataset on 03/21/20 from https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases (https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases). For this question obtain a new confirmed case dataset from the website. The file you should download should be called `time_series_covid19_confirmed_global.csv` .

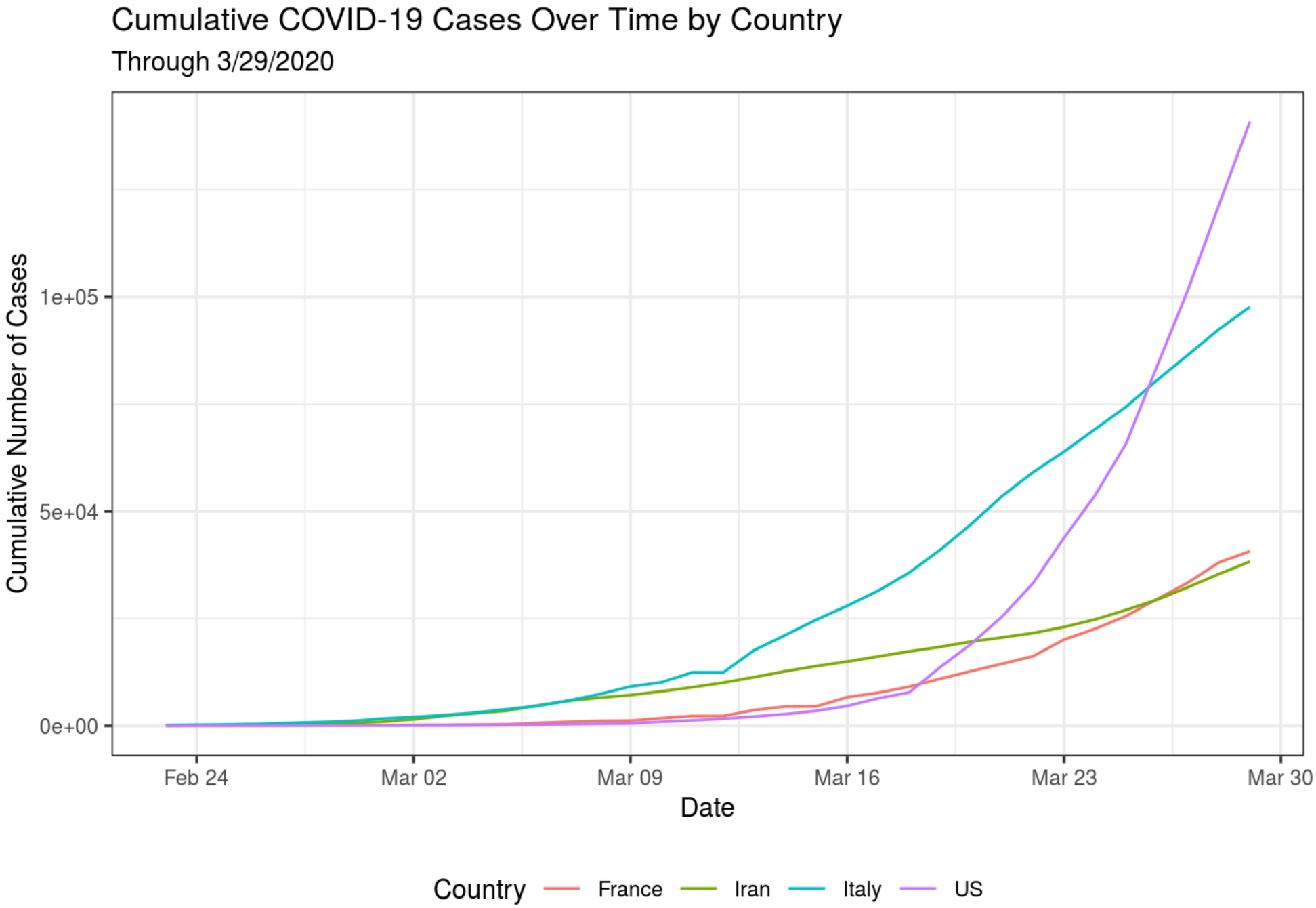
a)

Using your updated case dataset `time_series_covid19_confirmed_global.csv` *, re-create your figure from Question 1 part (i). Note: to do this correctly you will need to manipulate and summarize the data as you did in Question 1 parts (a)-(d). Be sure to include the date you downloaded the data – a good way to do this is to add a subtitle using labs(subtitle = "Your Date Here").*

```
#Import wide format dataframe
confirmed_df_wide<-read_csv("data/time_series_covid19_confirmed_
global.csv")

#Convert to long format
confirmed_df_long <- pivot_longer(confirmed_df_wide,
                                  cols = contains("/20"),
                                  names_to = "date",
                                  values_to = "cumulative_cases")

#Clean variable names
confirmed_df_long <- confirmed_df_long %>%
  clean_names()

#Convert date variable into date format
confirmed_df_long <- confirmed_df_long %>%
  mutate(date = mdy(date))

#Cumulative cases per day
```

```
confirmed_df_daily_ctry <- confirmed_df_long %>%
  group_by(country_region, date) %>%
  summarize(cumulative_cases = sum(cumulative_cases))

#Filter by desired countries
confirmed_df_ctry_filtered <- confirmed_df_daily_ctry %>%
  filter(country_region == "Italy" |
           country_region == "Iran" |
           country_region == "US" |
           country_region == "France")

#Create graph from Feb. 22nd onwards for selected countries
confirmed_df_ctry_date_filtered <- confirmed_df_ctry_filtered %
>%
  filter(date > "2020-02-22")

ggplot(data = confirmed_df_ctry_date_filtered) +
  geom_line(aes(x=date, y=cumulative_cases,
                color=country_region)) +
```

```
theme_bw() +
theme(legend.position = "bottom",
      text = element_text(size=10)) +
labs(title="Cumulative COVID-19 Cases Over Time by Country",
     x="Date", y="Cumulative Number of Cases",
     color="Country", subtitle = "Through 3/29/2020")
```


Cumulative COVID-19 Cases Over Time by Country
Through 3/29/2020
Cumulative Number of Cases
1e+05
5e+04
0e+00
Feb 24
Mar 02
Mar 09
Mar 16
Mar 23
Mar 30
Date
Country
France
Iran
Italy
US

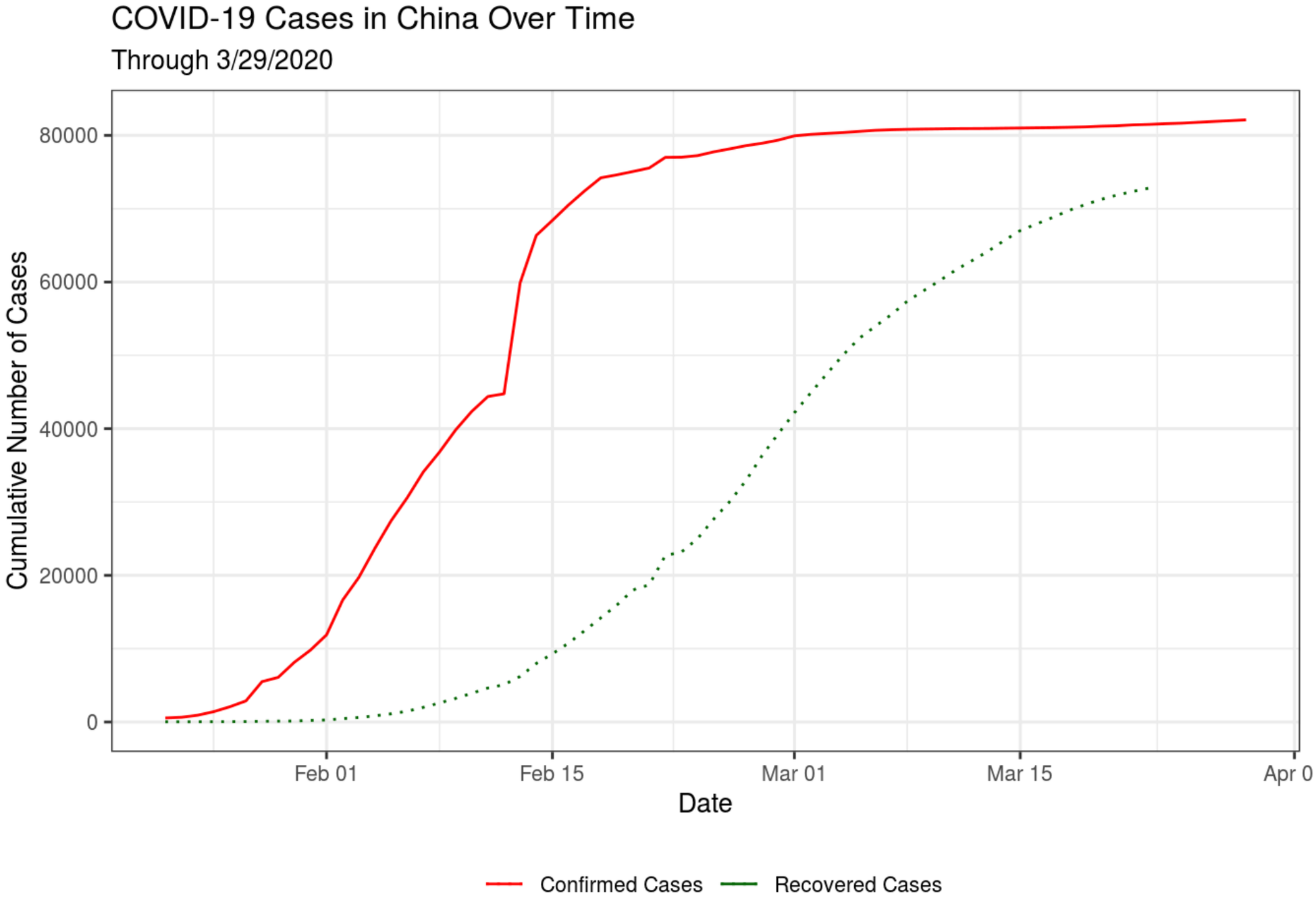
b)

Using your new confirmed case dataset and the recovery dataset located in `data/time_series_2019-ncov-Recovered.csv` *, create a graph that presents log-transformed confirmed cases and log-transformed recoveries for China. The cases should be presented as a solid line red line, and the recoveries should be a dark green dotted line. Make sure the plot has a title indicating the country you are plotting and labels for the x and y axes. Note: Johns Hopkins stopped compiling data on recoveries on 03/23/20, so your recoveries line will stop there.*

```
#Import dataframe in wide format
recovery_df_wide <- read_csv("data/time_series_2019-ncov-Recover
ed.csv")

#Convert into long format and clean variable names
recovery_df_long <- pivot_longer(data = recovery_df_wide,
                                 cols = contains("/20"),
                                 names_to = "date",
                                 values_to = "cases_recovered")

recovery_df_long <- recovery_df_long %>%
  clean_names() %>%
  mutate(date = mdy(date))

#Filter recovery dataframe to show only China
recovery_df_China <- recovery_df_long %>%
  group_by(country_region, date) %>%
  summarize(cumulative_recoveries = sum(cases_recovered)) %>%
  filter(country_region == "China")
```

```
#Filter new confirmed cases dataframe to show only China
confirmed_df_China <- confirmed_df_daily_ctry %>%
  filter(country_region == "China")

#Graph recoveries and confirmed cases in China
ggplot(data = confirmed_df_China) +
  geom_line(aes(x=date, y=cumulative_cases,
                color="Confirmed Cases")) +
  geom_line(data = recovery_df_China,
            aes(x=date, y=cumulative_recoveries,
                color="Recovered Cases"), linetype="dotted") +
  scale_color_manual(values = c("Confirmed Cases" = "red",
                                "Recovered Cases" = "dark green"))
+
  theme_bw() +
  theme(legend.position = "bottom", text=element_text(size=10))
+
  labs(title="COVID-19 Cases in China Over Time", x="Date",
```

```
       y="Cumulative Number of Cases", color=NULL,
       subtitle="Through 3/29/2020")
```


COVID-19 Cases in China Over Time
Through 3/29/2020
Cumulative Number of Cases
80000
60000
40000
20000
0
Feb 01
Feb 15
Mar 01
Mar 15
Apr 0
Date
Confirmed Cases
Recovered Cases

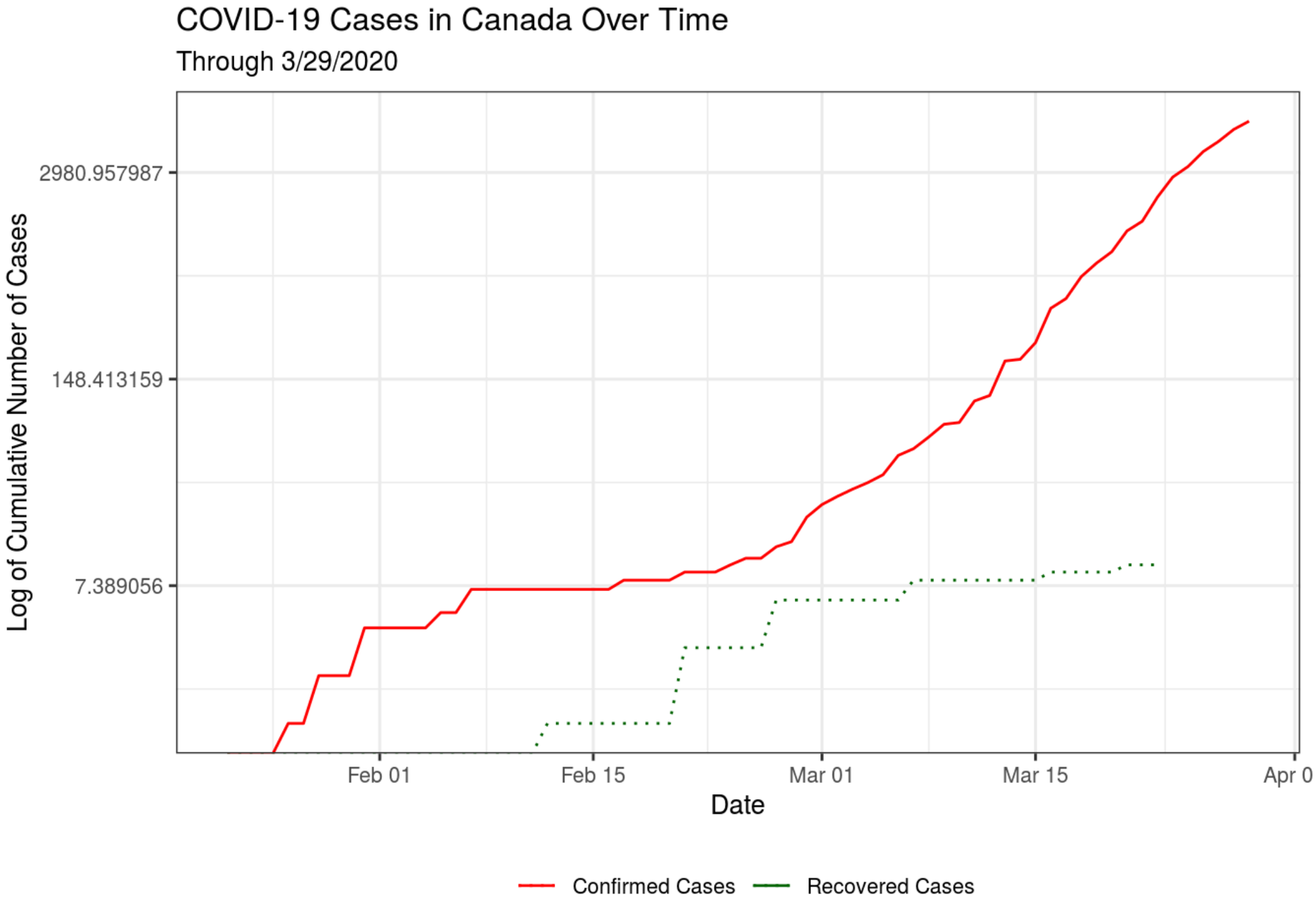
c)

Create a function using your code from part (b). Your function should take a string input that is a country name, and should output a plot of log-transformed confirmed cases over time and log-transformed recoveries over time for that country.The cases should be presented as a solid line red line, and the recoveries should be a dark green dotted line. Make sure the plot has a title indicating the country you are plotting and labels for the x and y axes. Test your function by creating graphs for 3 countries of your choice.

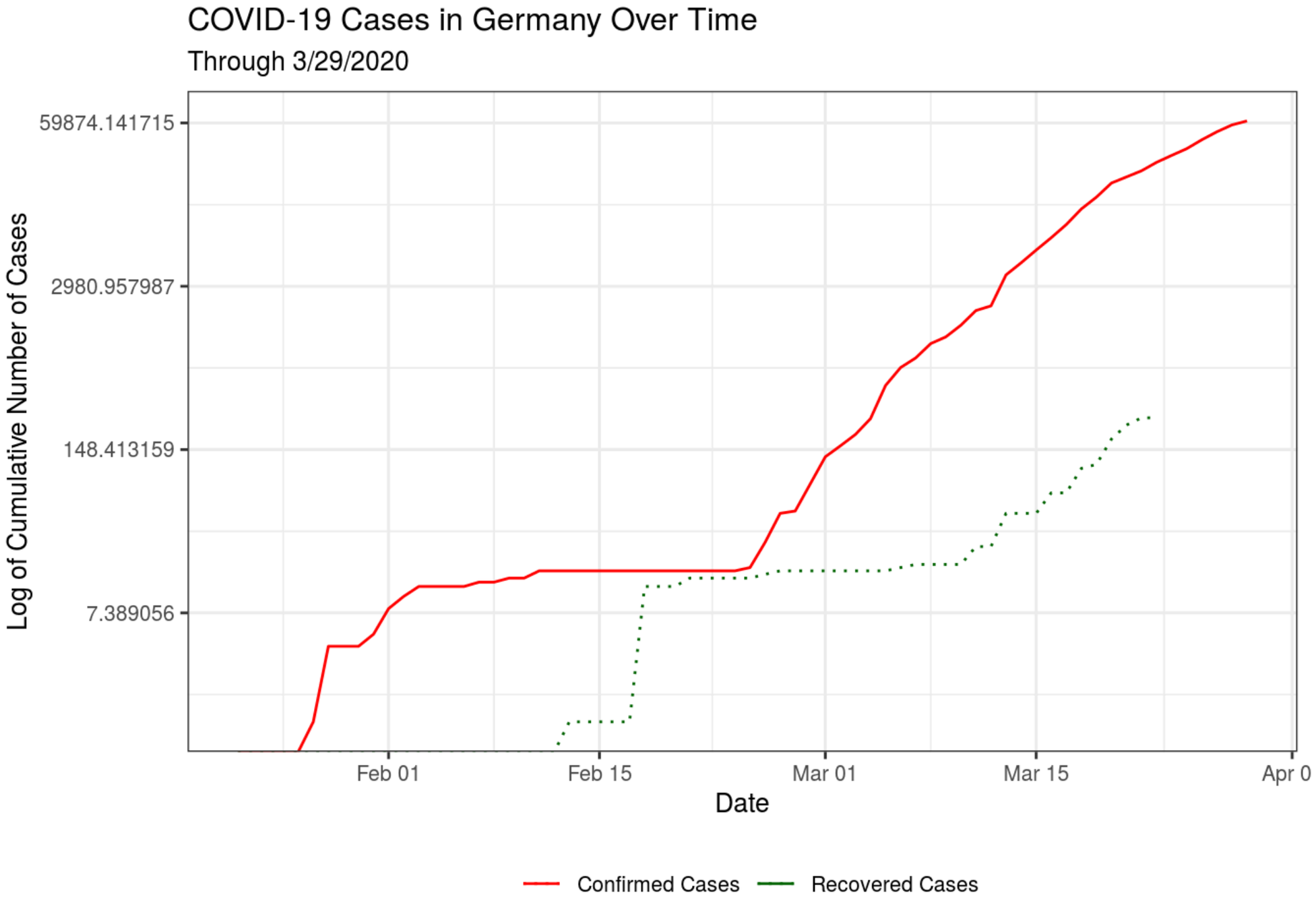
```
### Create the function
plot_cases_ctry <- function(country) {

  #Create filtered confirmed cases dataframe for country
  confirmed_df <- confirmed_df_daily_ctry %>%
    filter(country_region == country)

  #Create filtered recovered cases dataframe for country
  recovered_df <- recovery_df_long %>%
    group_by(country_region, date) %>%
    summarize(cumulative_recoveries = sum(cases_recovered)) %>%
    filter(country_region == country)

  #Create ggplot (store it as an object)
  plot <- ggplot(data = confirmed_df) +
    geom_line(aes(x=date, y=cumulative_cases,
                  color="Confirmed Cases")) +
    geom_line(data = recovered_df,
              aes(x=date, y=cumulative_recoveries,
```

```
                    color="Recovered Cases"), linetype="dotted") +
    scale_color_manual(values = c("Confirmed Cases" = "red",
                                  "Recovered Cases" = "dark gree
n")) +
    scale_y_continuous(trans = "log") +
    theme_bw() +
    theme(legend.position = "bottom", text=element_text(size=1
0)) +
    labs(title=paste("COVID-19 Cases in", country, "Over Time"),
         x="Date", y="Log of Cumulative Number of Cases",
         color=NULL, subtitle="Through 3/29/2020")

  #Return the plot so that the graph is displayed
  return(plot)

}



#Test the function
```

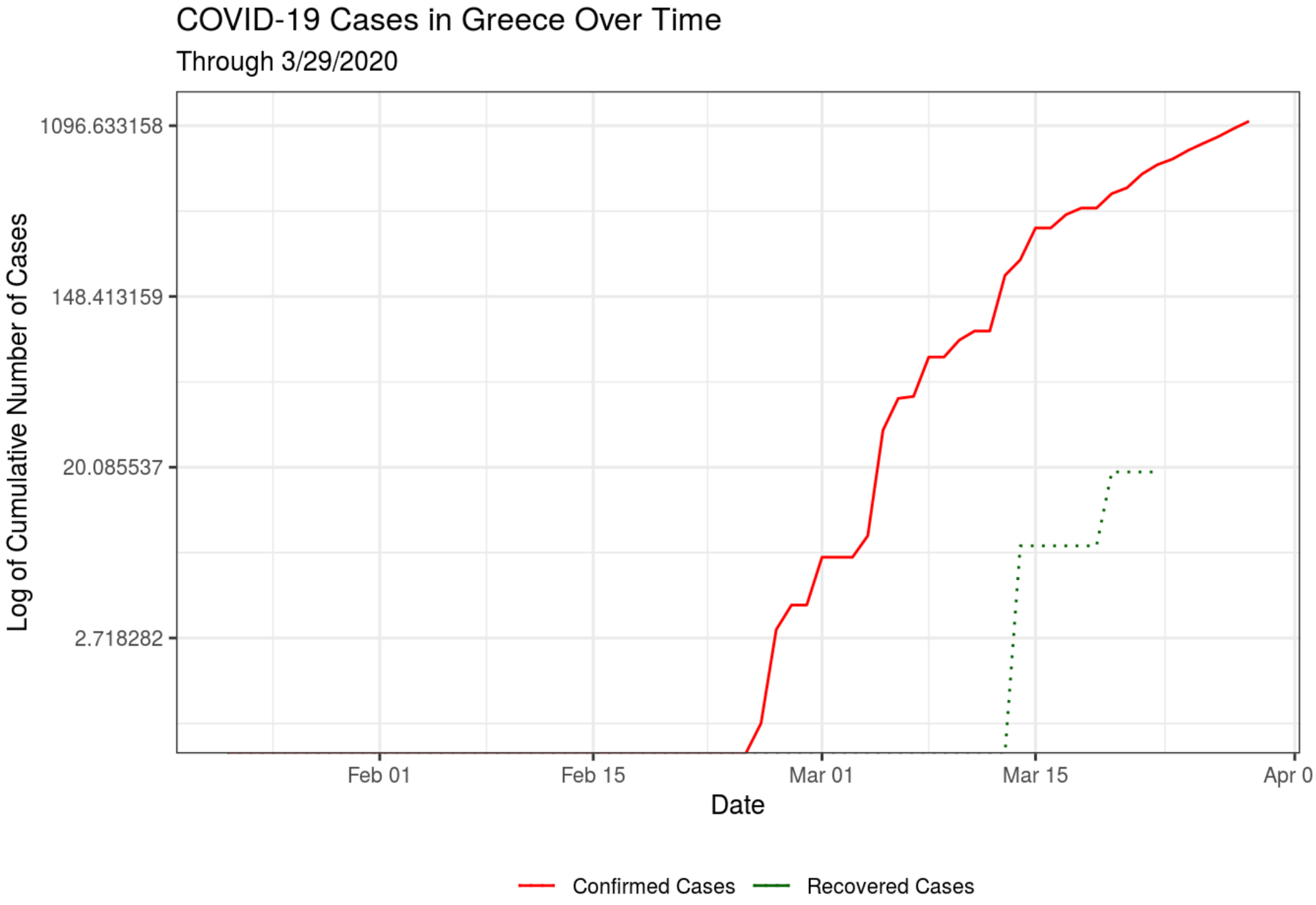
```
#Country 1 = Canada
plot_cases_ctry("Canada")
```


COVID-19 Cases in Canada Over Time
Through 3/29/2020
Log of Cumulative Number of Cases
2980.957987
148.413159
7.389056
Feb 01
Feb 15
Mar 01
Mar 15
Apr 0
Date
Confirmed Cases
Recovered Cases

```
#Country 2 = Germany
plot_cases_ctry("Germany")
```


COVID-19 Cases in Germany Over Time
Through 3/29/2020
59874.141715
2980.957987
148.413159
7.389056
Log of Cumulative Number of Cases
Feb 01
Feb 15
Mar 01
Mar 15
Apr 0
Date
Confirmed Cases
Recovered Cases

```
#Country 3 = Greece
plot_cases_ctry("Greece")
```


COVID-19 Cases in Greece Over Time
Through 3/29/2020
Log of Cumulative Number of Cases
1096.633158
148.413159
20.085537
2.718282
Feb 01
Feb 15
Mar 01
Mar 15
Apr 0
Date
Confirmed Cases
Recovered Cases