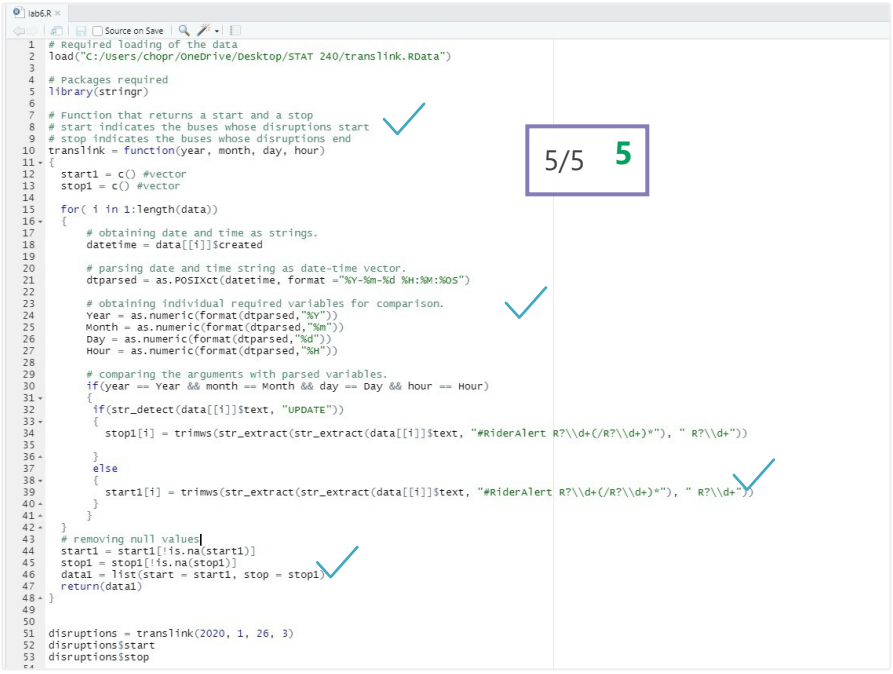

lab6.R

```
# Required loading of the data
load("C:/Users/chopr/OneDrive/Desktop/STAT 240/translink.RData")

# Packages required
library(stringr)

# Function that returns a start and a stop
# start indicates the buses whose disruptions start
# stop indicates the buses whose disruptions end
translink = function(year, month, day, hour)
{
  start1 = c() #vector
  stop1 = c() #vector

  for( i in 1:length(data))
  {
      # obtaining date and time as strings.
      datetime = data[[i]]$created

      # parsing date and time string as date-time vector.
      dtparsed = as.POSIXct(datetime, format ="%Y-%m-%d %H:%M:%OS")

      # obtaining individual required variables for comparison.
      Year = as.numeric(format(dtparsed,"%Y"))
      Month = as.numeric(format(dtparsed,"%m"))
      Day = as.numeric(format(dtparsed,"%d"))
      Hour = as.numeric(format(dtparsed,"%H"))

      # comparing the arguments with parsed variables.
      if(year == Year && month == Month && day == Day && hour == Hour)
      {
       if(str_detect(data[[i]]$text, "UPDATE"))
       {
         stop1[i] = trimws(str_extract(str_extract(data[[i]]$text, "#RiderAlert R?\\d+(/R?\\d+)*"), " R?\\d+"))

       }
       else
       {
         start1[i] = trimws(str_extract(str_extract(data[[i]]$text, "#RiderAlert R?\\d+(/R?\\d+)*"), " R?\\d+"))
       }
      }
  }
  # removing null values
  start1 = start1[!is.na(start1)]
  stop1 = stop1[!is.na(stop1)]
  data1 = list(start = start1, stop = stop1)
  return(data1)
}


disruptions = translink(2020, 1, 26, 3)
disruptions$start
disruptions$stop
```

5/5 **5**

5/5 **5**

REPORT

Anshal Chopra

18th March 2021

We can use APIs of various websites in R to analyse or use the data to our needs. For lab 6 we I had to write an R function named **translink** that takes 4 numeric arguments: a year (a numeric value such as 2020), a month (a numeric value between 1 and 12, inclusive with 1 indicating January), and a day of the month, and an hour of the day (in 24-hour time). The R function should return a list with two elements: 1. An element with the name **start** with a value specifying a character vector enumerating all bus routes that started to have disruptions during the hour indicated by the date and time provided to the function. 2. An element with the name **stop** and with a value specifying a character vector enumerating all bus routes that stopped having a disruption during the hour indicated by the date and time provided to the function.

First, we need to load the file which contains all the data i.e., the **Translink** file. Also, I loaded the necessary packages which is **stringr** in our case. As said in the lab question, I was told to give 4 arguments to the function **translink** which I did namely **year**, **month**, **day**, **hour**. The approach was to extract the date-time from the data **Translink** and compare it with the arguments passed in. So, we parsed the date-time from **Translink** which is of the form of a string and parsed it such as to obtain the data to be compared with year, month, day and hour. To compare the date and time I looped through the entire data file **Translink**. If the date and time matched, I put in another condition i.e., if the text or the tweet in **Translink** contained the word **"UPDATE"** and it started with **#RiderAlert**, then we knew that the disruption had stopped at this point and we were to extract the bus numbers from here and put them in **stop**. If the word **UPDATE** was not there, but **#RiderAlert** was there then the disruption had started, and we had to put the bus numbers in **start**. We extracted the bus numbers using functions from **stringr** package such as **str_extract** and by defining the appropriate regular expressions to put as arguments in the same. The regular expression was such that if it had the letter R in it followed by the digits, it extracted the bus number, which also worked if the bus numbers did not start with the letter R. In the end we removed the null values if any from start and stop and returned the list.

#TEST CASES

1.) translink(2020, 1, 26, 3)

```
> # Required loading of the data
> load("C:/Users/chopr/OneDrive/Desktop/STAT 240/translink.RData")

> # Packages required
> library(stringr)

> # Function that returns a start and a stop
> # start indicates the buses whose disruptions start
> # stop indicates the buses whose disruptions end
 .... [TRUNCATED]

> disruptions = translink(2020, 1, 26, 3)

> disruptions$start
[1] "406" "401"

> disruptions$stop
[1] "23"
```

2.) translink(2020, 2, 23, 1)

```
> # Required loading of the data
> load("C:/Users/chopr/OneDrive/Desktop/STAT 240/translink.RData")

> # Packages required
> library(stringr)

> # Function that returns a start and a stop
> # start indicates the buses whose disruptions start
> # stop indicates the buses whose disruptions end
 .... [TRUNCATED]

> disruptions = translink(2020, 2, 23, 1)

> disruptions$start
NULL

> disruptions$stop
NULL
```

3.) translink(2020, 2, 22, 3)

```
> # Required loading of the data
> load("C:/Users/chopr/OneDrive/Desktop/STAT 240/translink.RData")

> # Packages required
> library(stringr)

> # Function that returns a start and a stop
> # start indicates the buses whose disruptions start
> # stop indicates the buses whose disruptions end
 .... [TRUNCATED]

> disruptions = translink(2020, 2, 22, 3)

> disruptions$start
[1] "239" "239" "403" "403" "430" "403"

> disruptions$stop
NULL
```

4.) translink(2020, 2, 11, 1)

```
> # Required loading of the data
> load("C:/Users/chopr/OneDrive/Desktop/STAT 240/translink.RData")

> # Packages required
> library(stringr)

> # Function that returns a start and a stop
> # start indicates the buses whose disruptions start
> # stop indicates the buses whose disruptions end
 .... [TRUNCATED]

> disruptions = translink(2020, 2, 11, 1)

> disruptions$start
[1] "22"

> disruptions$stop
NULL
```

The above test cases clearly shows that we get the required results. To conclude, the code works perfectly fine and gets me all the results.