Final Report of Traineeship Program 2023

On

"Classify Suspected Infection in Patients"

MEDTOUREASY

28th May 2023

ACKNOWLDEGMENTS

- The traineeship opportunity that I had with MedTourEasy was a great change for learning and understanding the intricacies of the subject of Data Visualizations in Data Analytics; and also, for personal as well as professional development. I am very obliged for having a chance to interact with so many professionals who guided me throughout the traineeship project and made it a great learning curve for me.
- Firstly, I express my deepest gratitude and special thanks to the Training & Development Team of MedTourEasy who gave me an opportunity to carry out my traineeship at their esteemed organization. Also, I express my thanks to the team and specially my mentor **Mr. Ankit Hasija** for making me understand the details of the Data Analytics profile and training me in the same so that I can carry out the project properly and with maximum satisfaction and also for spearing his valuable time in spite of his busy schedule.
- I would also like to thank the team of MedTourEasy and my colleagues who made the working environment productive and very conducive.

Contents

- Acknowledgements ………………………………………….02
- Abstract…………………………………………………………..04
- About the company…………………………………………07
- About the Project……………………………………………..08
- Objectives and Deliverables ……………………….09-10
- Methodology………………………………………………….11
- Implementation……………………………………………..16
- Workflow………………………………………………………..20
- Data Tables……………………………………………………..32
- R commands and output………………………………..37
- Conclusion and Future scope…………………………..50

ABSTRACT

- Sepsis is a deadly illness that accounts for a large portion of in-hospital deaths. It occurs when a person's organs shut down in response to a severe infection. This public health problem is a major target for research, and hospital records can help us investigate the problem.
- Sepsis is a potentially life-threatening condition that occurs when the body's response to an infection triggers a widespread inflammatory reaction. It is often referred to as blood poisoning. Sepsis can occur as a result of various infections, including bacterial, viral, or fungal infections.
- When the immune system detects an infection, it releases chemicals into the bloodstream to fight off the invading organisms. However, in sepsis, the immune system response becomes uncontrolled and can cause widespread inflammation throughout the body. This inflammation can lead to organ damage and failure, and in severe cases, it can result in septic shock, which is a medical emergency.

- Sepsis requires immediate medical attention, as it can progress rapidly and become life-threatening. If you suspect sepsis, it is important to seek medical help right away. Doctors typically diagnose sepsis based on a combination of symptoms, physical examination findings, and laboratory tests.
- The treatment for sepsis usually involves hospitalization and includes:

1. Antibiotics: Broad-spectrum antibiotics are administered intravenously to target the infection.
2. Intravenous fluids: Fluids are given to maintain blood pressure and prevent dehydration.
3. Vasopressors: Medications may be used to increase blood pressure if it drops too low.
4. Oxygen therapy: Supplemental oxygen may be provided to ensure adequate oxygenation.

5. Source control: If there is a localized infection, such as an abscess, drainage or surgical intervention may be necessary to remove the source of infection.

- In this R project, we will identify hospital patients with severe infection using medical record data.
- Therefore, this project aims at collecting and analyzing wide variety of large data sets, create intuitive and interactive dashboards for representing hospital patients with severe infection in order to gain meaningful insights.

About the Company:

- MedTourEasy, a global healthcare company, provides you the informational resources needed to evaluate your global options. It provides analytical solutions to our partner healthcare providers globally. It helps you find the right healthcare solution based on specific health needs, affordable care while meeting the quality standards that you expect to have in healthcare. MedTourEasy improves access to healthcare for people everywhere. It is an easy to use platform and service that helps patients to get medical second opinions and to schedule affordable, high-quality medical treatment abroad.

About the Project:

- The early identification and classification of infections in patients are critical for timely and effective treatment. This project aims to develop a machine learning-based model to classify suspected infections in patients. By leveraging the power of artificial intelligence and data analysis techniques, the model will assist healthcare professionals in making accurate and efficient decisions, leading to improved patient outcomes and reduced mortality rates.
- *Analysis of the problem*: This is done to assess the suspected patients with the infection. It contains statistics and data representing the problem – patients, blood culture, antibiotic given and infection rate . Also, it contains many comparative statistics with respect to parameters like last drug administration ,days since last administration, new antibiotic given etc.

Objectives of Project :

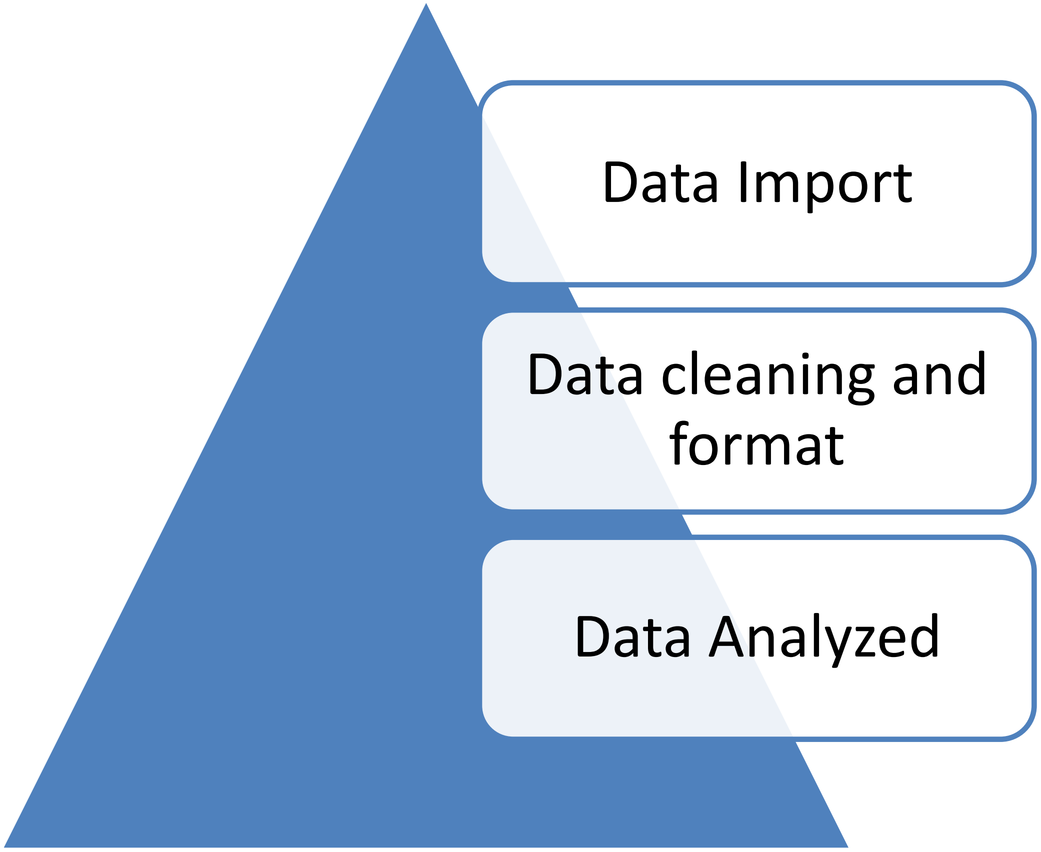
- ***Develop a reliable and accurate machine learning model***: The primary objective of the project is to develop a machine learning-based model that can effectively classify suspected infections in patients. The model should be capable of accurately differentiating between infected and non-infected patients based on relevant data.
- ***Enhance early identification of infections:*** The project aims to improve the early identification of infections in patients, enabling healthcare professionals to initiate timely interventions and treatment. By providing a reliable classification system, the project intends to reduce the risk of delayed diagnosis and potential complications associated with untreated infections.
- ***Optimize model performance and generalization:*** The project aims to optimize the developed model's performance by fine-tuning its parameters, employing feature selection techniques, and addressing overfitting or underfitting issues. The objective is to ensure the model's ability to generalize well to unseen data, increasing its reliability and applicability in real-world scenarios.

Deliverables of Project:

- ***Machine learning-based infection classification model:*** The primary deliverable is a developed and trained machine learning model capable of classifying suspected infections in patients accurately. The model should demonstrate robust performance and high accuracy in differentiating between infected and non-infected patients.
- ***Documentation and code:*** Detailed documentation of the project, including explanations of the methodology, data preprocessing steps, feature selection techniques, model development, and evaluation procedures. Additionally, the project should provide well-documented code that can be used to replicate the results and further enhance the developed model.
- ***Recommendations for implementation:*** A set of recommendations and guidelines for the implementation and integration of the infection classification model into real-world healthcare systems. The deliverable should address practical considerations, potential challenges, and suggestions for effectively deploying the model to assist healthcare professionals in the classification of suspected infections in patients.

METHODOLOGY:

- **<u>Flow of the Project</u> :** The project followed the following steps to accomplish the desired objectives and deliverables.



Language and Platform Used :

Language: R

- It is a programming language and software environment for statistical analysis, representation of graphics, and reports. R was developed in the University of Auckland, New Zealand by Ross Ihaka and Robert Gentleman, and is currently being developed by the R Technology Core Team. R is a programming language and software environment for statistical analysis, representation of graphics, and reporting. The important features of R are:
  - **R is a well-developed, simple, and effective programming language that includes conditionals, loops, recursive functions defined by the user, and input and output facilities.**
  - **R has efficient data processing and storage facilities.**
  - **R includes a set of operators for arrays, lists, vectors, and matrix calculations.**
  - **R offers a detailed, coherent and organized data analysis tool set.**
  - **R provides graphical data analysis facilities and displays either directly on the computer or printing on papers.**

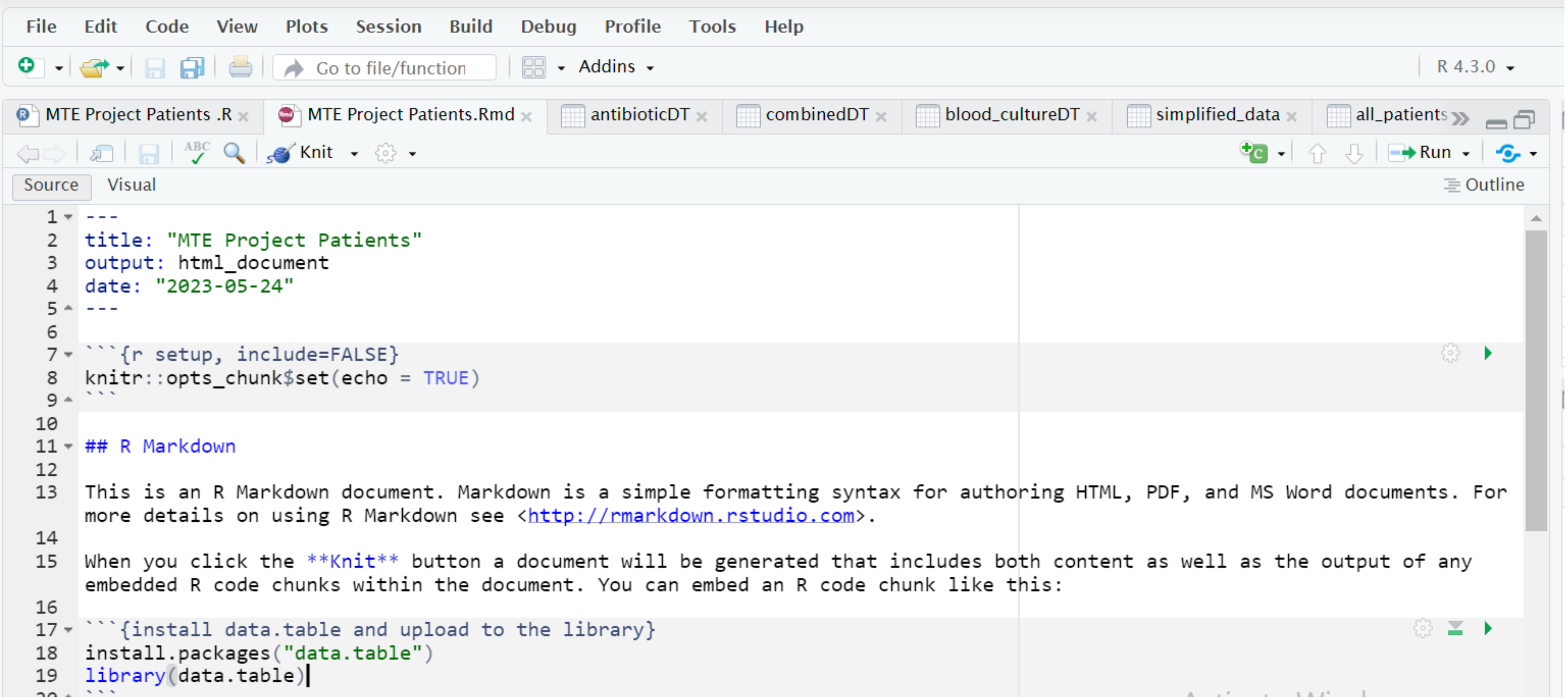
IDE: RStudio

- RStudio is an integrated development environment for R (IDE). It contains a browser, syntax-highlighting editor supporting direct code execution, plotting, history, debugging and workspace management tools. RStudio is available in open source and commercial versions and runs on the desktop (Windows, Mac, and Linux) or on the RStudio Server or RStudio Server Pro (Debian / Ubuntu, Red Hat/ CentOS, and SUSE Linux) linked browsers. Major features are:
  - RStudio runs on most desktops or on a server and accessed over the web.
  - It integrates the tools you use with R into a single environment.
  - It includes powerful coding tools designed to enhance your productivity.
  - It enables rapid navigation to files and functions.
  - It has integrated support for Git and Subversion.
  - It supports authoring HTML, PDF, Word Documents, and slide shows.
  - It supports interactive graphics with Shiny and ggvis.

Package: RMarkdown

- R Markdown provides a data science authoring framework (.Rmd files). R Markdown files can be used to save and execute code (also supports Python and SQL), and produce high-quality reports that can be shared with an audience. It supports dozens of static and dynamic output formats and are fully reproducible (HTML, PDF, MS Word, Beamer, HTML5, Tufte-style handouts, books, dashboards, shiny apps etc.)
- RMarkdown file contains the following parts:
  - An (optional) YAML header surrounded by --- : The title, output format, orientation and layout of the file are defined.
  - R code chunks surrounded by ```{r} : It is in the arranged in the form of rows and columns where the coding is to be done.
  - Text mixed with simple text formatting.
  - Knit button: To run the file and display output.

R Markdown file:



~~~
---
title: "MTE Project Patients"
output: html_document
date: "2023-05-24"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{install data.table and upload to the library}
install.packages("data.table")
library(data.table)
```
```
```

IMPLEMENTATION:

- Gathering Requirements and Defining Problem Statement :

- This is the first step wherein the requirements are collected from the clients to understand the deliverables and goals to be achieved after which a problem statement is defined which has to be adhered to while development of the project.

– Data Collection and Importing:

- Data collection is a systematic approach for gathering and measuring information from a variety of sources in order to obtain a complete and accurate picture of an interest area. It helps an individual or organization to address specific questions, determine outcomes and forecast future probabilities and patterns.
- The data classify suspected infection in patients has been collected through the resources: https://drive.google.com/file/d/162KBAQEUU6b84LIgj8MU3a7E6Yrel_bg/view?usp=sharing
- Data importing is referred to as uploading the required data into the coding environment from internal sources (computer) or external sources (online websites and data repositories). This data can then be manipulated, aggregated, filtered according to the requirements and needs of the project.

Packages Used:

- **Readr**: The goal of readr is to provide a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. To accurately read a rectangular dataset with readr, one needs to combine two pieces: a function that parses the overall file, and a column specification.
- **Readxl**: The readxl package is used to get data out of Excel and into R. Compared to many of the existing packages (e.g. gdata, xlsx, xlsReadWrite) readxl has no external dependencies, so it's easy to install and use on all operating systems. It isdesigned to work with tabular data. readxl supports both the legacy .xls format and the modern xml-based .xlsx format.

Functions Used:

- **read.csv ()**: It is a wrapper function for read.table() that mandates a comma as separator and uses the input file's first line as header that specifies the table's column names. Thus, it is an ideal candidate to read CSV files. It has an additional parameter of url() which is used to pull live data directly from GitHub repository.
- **read_excel ():** It calls excel_format() to determine if path is xls or xlsx, based on the file extension and the file itself, in that order.

– Designing Databases

- Once the data has been collected and imported into the R environment, it is important to design the structure of the database tables so as to identify the constraints in the data, keys, dependencies and relations between various tables.
- Once the data is imported in the environment, it is converted into a data frame (data type in R) which makes it easy to maintain the data in form of tables. The various tables which have been created are mentioned as follows:

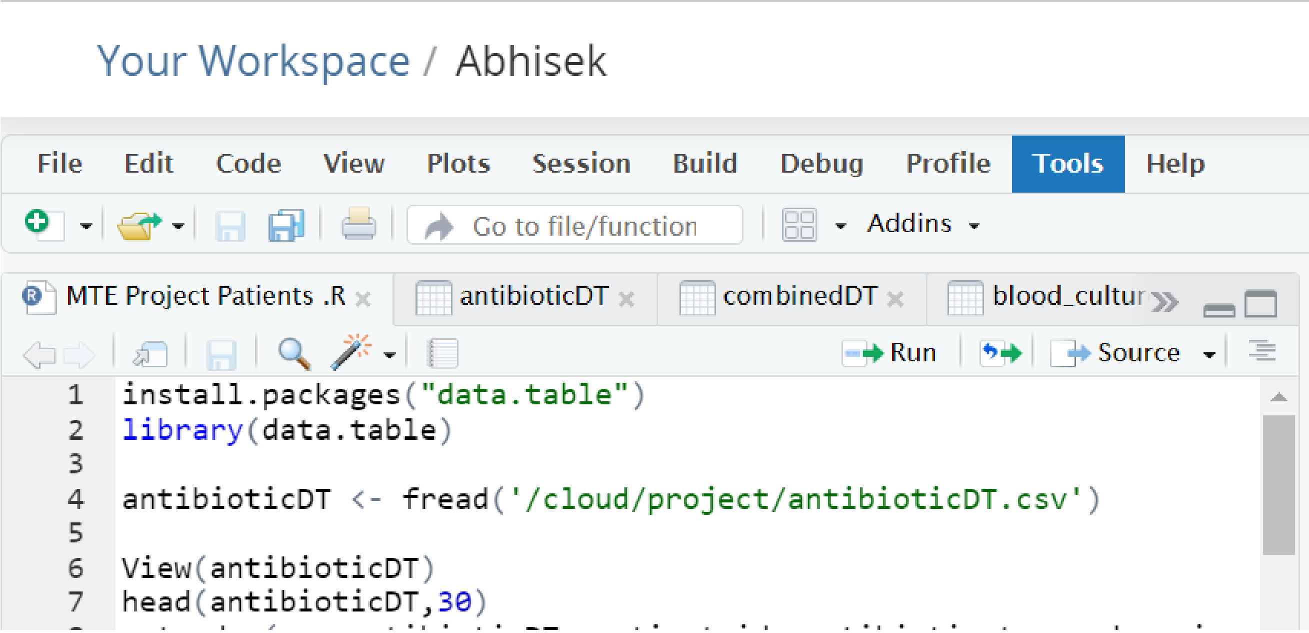
Attribute	Data type	Size
antibioticDT	VARCHAR	7
combinedDT	VARCHAR	11
blood_culture_DT	INT	2
Simplified_data	INT	4
All_patientsDT	INT	2

“WORKFLOW/ WORKPROCESS”

Task 1: Instructions

First, let's take a look at the antibiotic data.

- Load the `data.table` package using `library()`.
- Read in `datasets/antibioticDT.csv` using the `data.table` function `fread()`.
- Look at the first 30 rows.

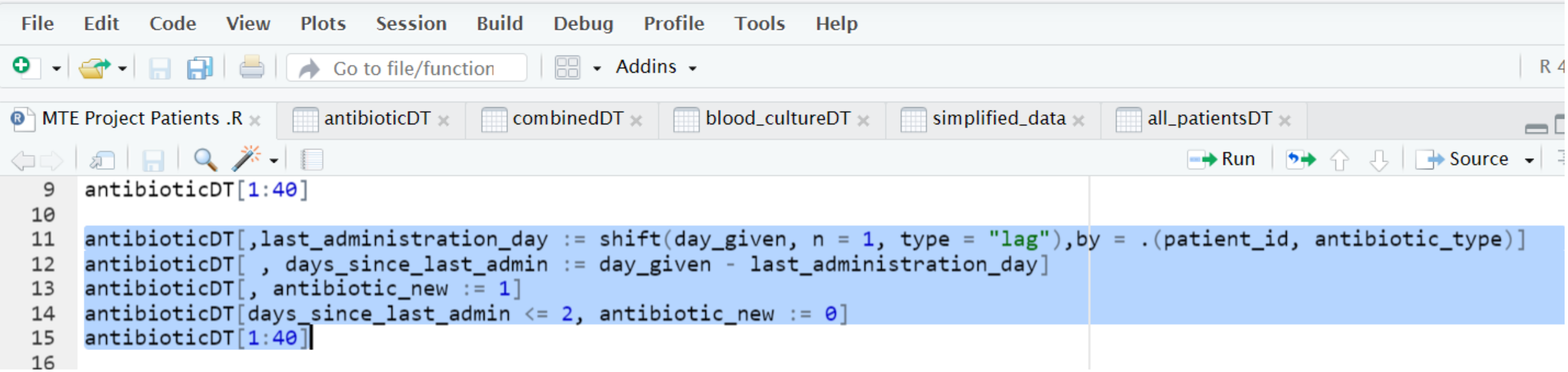


```r
install.packages("data.table")
library(data.table)

antibioticDT <- fread('/cloud/project/antibioticDT.csv')

View(antibioticDT)
head(antibioticDT,30)
```

Task 2: Instructions

Identify rows representing "new" antibiotics.

- Use `setorder()` to sort the data by `patient_id`, `antibiotic_type`, and `day_given`. Print and examine the first 40 rows.
- Use `shift` to calculate the last day the antibiotic was given to a patient. Call the new variable, `last_administration_day`.
- Calculate the number of days since the antibiotic was administered to a patient. Call the new variable, `days_since_last_admin`.
- In a two-step process, create a new variable called `antibiotic_new` that is initialized to one, then reset it to zero in rows where it has only been one or two days since the antibiotic was given.

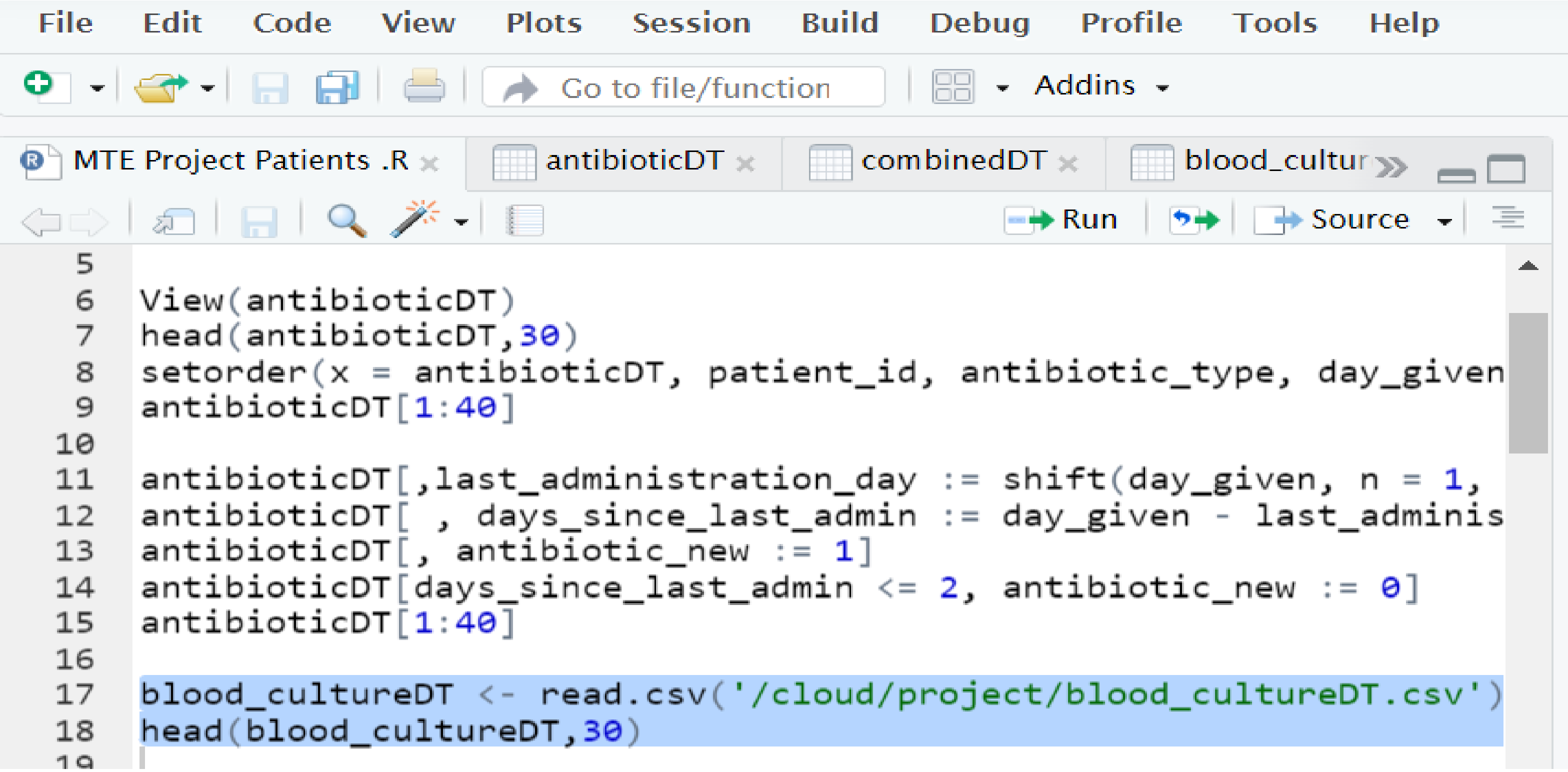


```
antibioticDT[1:40]

antibioticDT[,last_administration_day := shift(day_given, n = 1, type = "lag"),by = .(patient_id, antibiotic_type)]
antibioticDT[ , days_since_last_admin := day_given - last_administration_day]
antibioticDT[, antibiotic_new := 1]
antibioticDT[days_since_last_admin <= 2, antibiotic_new := 0]
antibioticDT[1:40]

```

Task 3: Instructions

Investigate the blood culture data.

- Read in `"datasets/blood_cultureDT.csv"`.
- Print the first 30 rows.

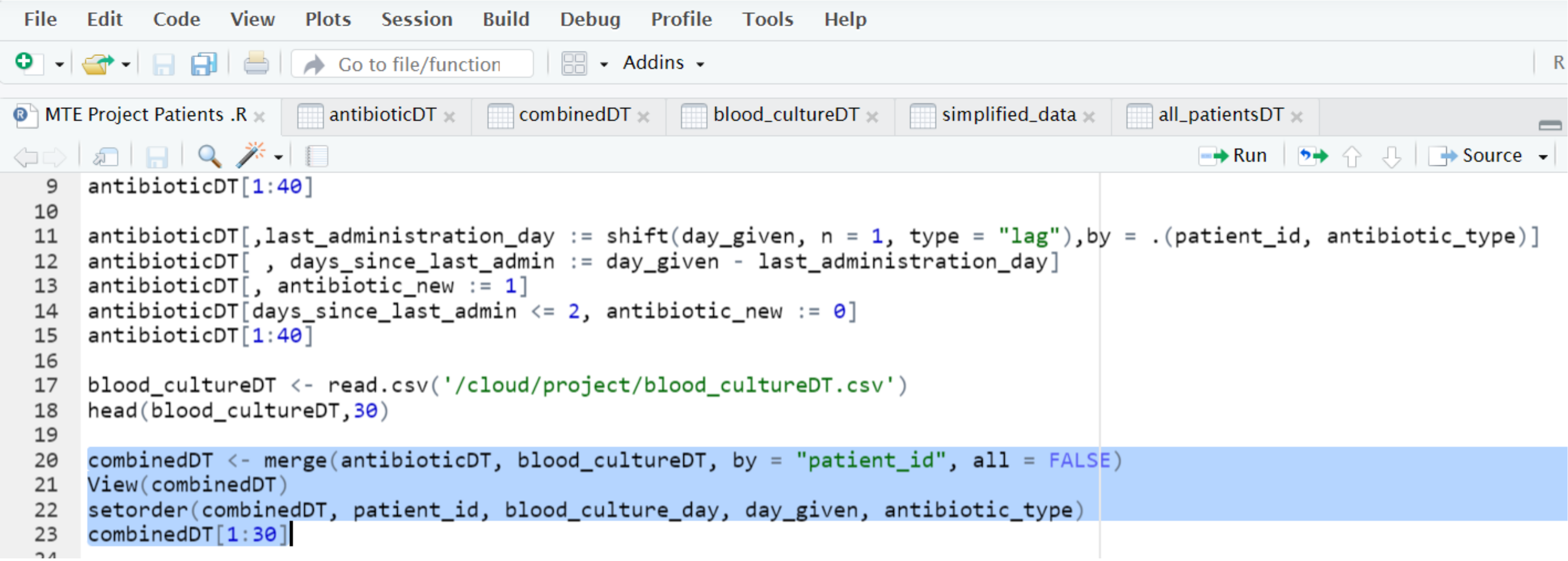


```
View(antibioticDT)
head(antibioticDT,30)
setorder(x = antibioticDT, patient_id, antibiotic_type, day_given
antibioticDT[1:40]

antibioticDT[,last_administration_day := shift(day_given, n = 1,
antibioticDT[ , days_since_last_admin := day_given - last_adminis
antibioticDT[, antibiotic_new := 1]
antibioticDT[days_since_last_admin <= 2, antibiotic_new := 0]
antibioticDT[1:40]

blood_cultureDT <- read.csv('/cloud/project/blood_cultureDT.csv')
head(blood_cultureDT,30)
```

Task 4: Instructions

Merge the antibiotic data with the blood culture data.

- Make a combined dataset by merging `antibioticDT` with `blood_cultureDT`.
- Sort by `patient_id`, `blood_culture_day`, `day_given`, and `antibiotic_type`.
- Print and examine the first 30 rows.

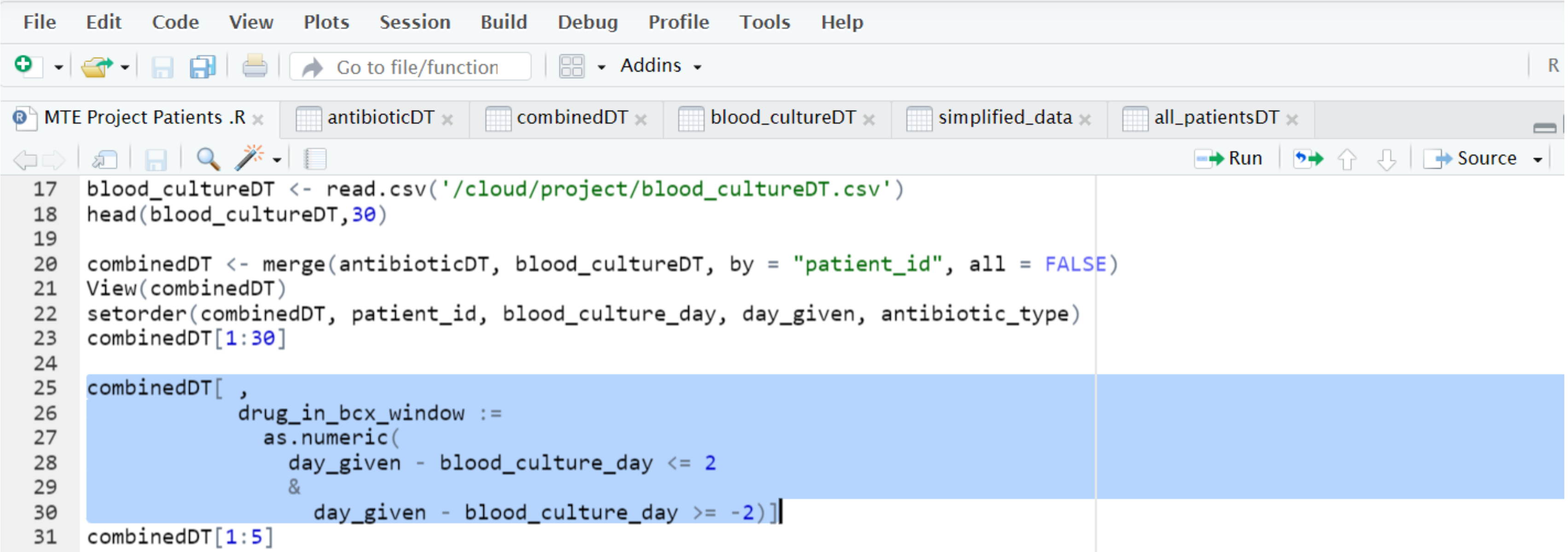


```r
antibioticDT[1:40]

antibioticDT[,last_administration_day := shift(day_given, n = 1, type = "lag"),by = .(patient_id, antibiotic_type)]
antibioticDT[ , days_since_last_admin := day_given - last_administration_day]
antibioticDT[, antibiotic_new := 1]
antibioticDT[days_since_last_admin <= 2, antibiotic_new := 0]
antibioticDT[1:40]

blood_cultureDT <- read.csv('/cloud/project/blood_cultureDT.csv')
head(blood_cultureDT,30)

combinedDT <- merge(antibioticDT, blood_cultureDT, by = "patient_id", all = FALSE)
View(combinedDT)
setorder(combinedDT, patient_id, blood_culture_day, day_given, antibiotic_type)
combinedDT[1:30]
```

Task 5: Instructions

Make a new variable indicating whether or not the antibiotic administration and blood culture are within two days of each other.

- Make a new variable called `drug_in_bcx_window` which is `1` if the drug was given in the 2-day window and `0` otherwise.

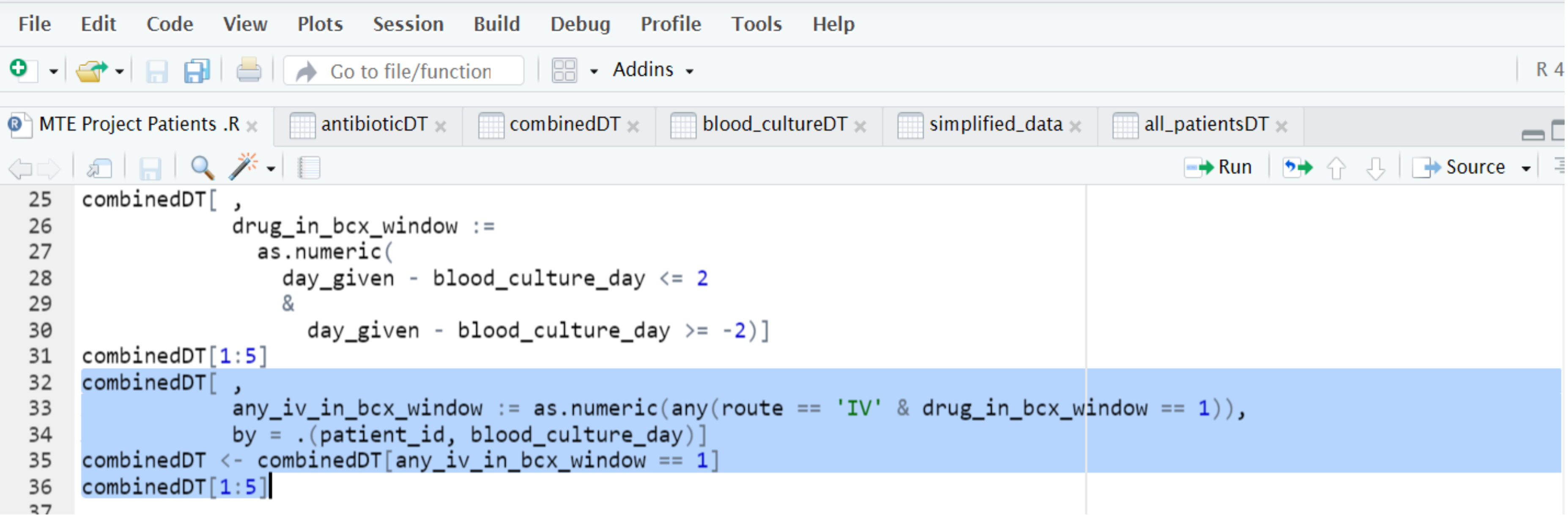


```
blood_cultureDT <- read.csv('/cloud/project/blood_cultureDT.csv')
head(blood_cultureDT,30)

combinedDT <- merge(antibioticDT, blood_cultureDT, by = "patient_id", all = FALSE)
View(combinedDT)
setorder(combinedDT, patient_id, blood_culture_day, day_given, antibiotic_type)
combinedDT[1:30]

combinedDT[ ,
            drug_in_bcx_window :=
              as.numeric(
                day_given - blood_culture_day <= 2
                &
                  day_given - blood_culture_day >= -2)]
combinedDT[1:5]
```

Task 6: Instructions

For each patient/blood culture day combination, determine if at least one I.V. antibiotic was given in the +/-2 day window.

- Create a new variable, `any_iv_in_bcx_window`, indicating whether or not an I.V. drug was given within a +/-2 day window of a blood culture day.
- Exclude rows in which the `blood_culture_day` does not have any I.V. drugs in the window.

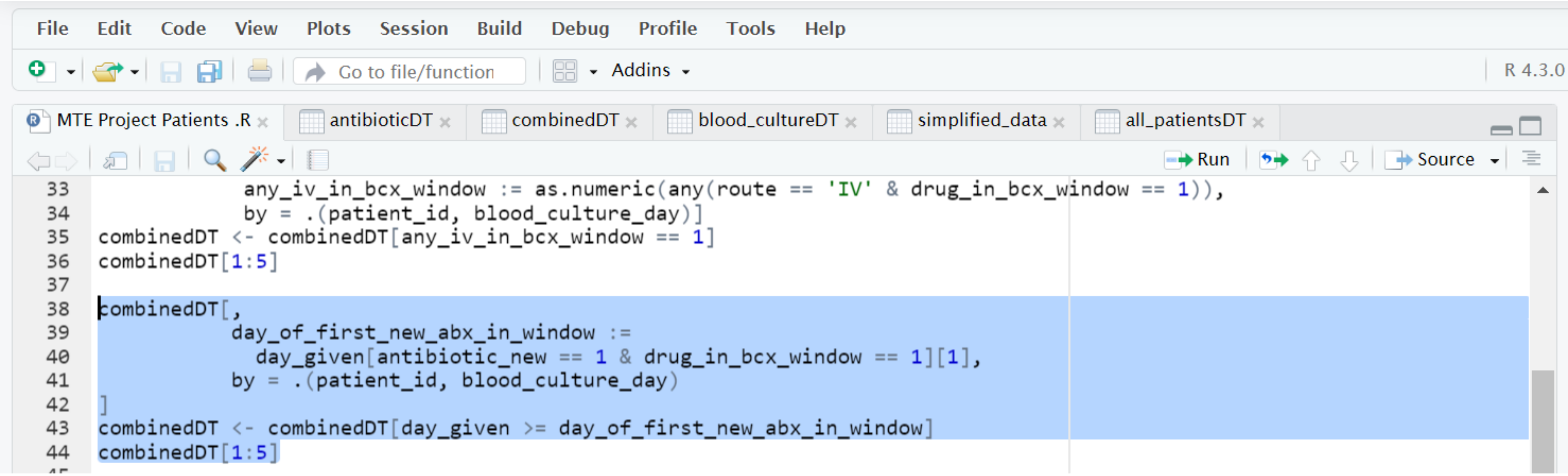


```
combinedDT[ ,
            drug_in_bcx_window :=
              as.numeric(
                day_given - blood_culture_day <= 2
                &
                  day_given - blood_culture_day >= -2)]
combinedDT[1:5]
combinedDT[ ,
            any_iv_in_bcx_window := as.numeric(any(route == 'IV' & drug_in_bcx_window == 1)),
            by = .(patient_id, blood_culture_day)]
combinedDT <- combinedDT[any_iv_in_bcx_window == 1]
combinedDT[1:5]
```

Task 7: Instructions

For each blood culture, find the **first day** of potential 4-day antibiotic sequences. This day will be the first day that is both in the window, and a new antibiotic was given.

- Create a new variable called `day_of_first_new_abx_in_window`.
- Remove rows where the day is before this first qualifying day.

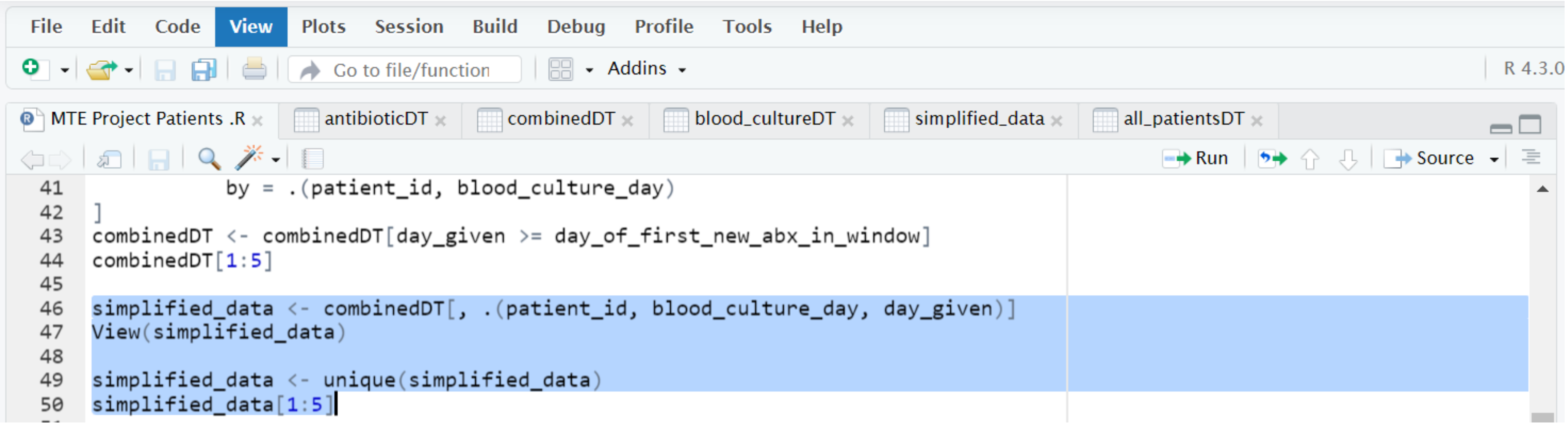


```
            any_iv_in_bcx_window := as.numeric(any(route == 'IV' & drug_in_bcx_window == 1)),
            by = .(patient_id, blood_culture_day)]
combinedDT <- combinedDT[any_iv_in_bcx_window == 1]
combinedDT[1:5]

combinedDT[,
           day_of_first_new_abx_in_window :=
             day_given[antibiotic_new == 1 & drug_in_bcx_window == 1][1],
           by = .(patient_id, blood_culture_day)
]
combinedDT <- combinedDT[day_given >= day_of_first_new_abx_in_window]
combinedDT[1:5]
```

Task 8: Instructions

Make a new dataset that only contains what we need to check the remaining criteria.

- Create a new data.table containing only `patient_id`, `blood_culture_day`, and `day_given`.
- Remove duplicate rows.

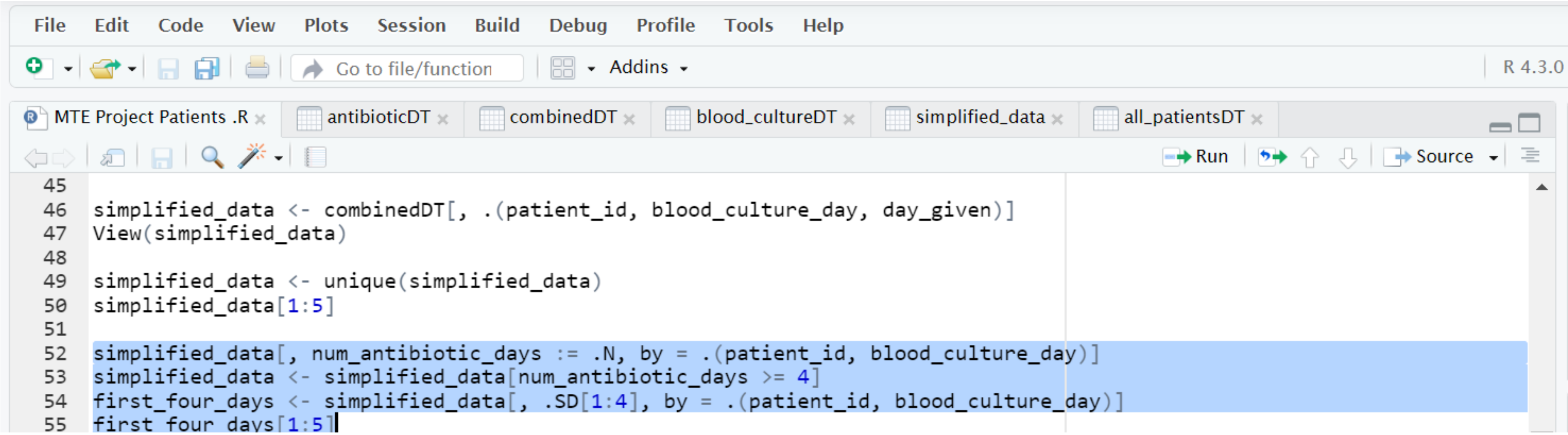


```
           by = .(patient_id, blood_culture_day)
]
combinedDT <- combinedDT[day_given >= day_of_first_new_abx_in_window]
combinedDT[1:5]

simplified_data <- combinedDT[, .(patient_id, blood_culture_day, day_given)]
View(simplified_data)

simplified_data <- unique(simplified_data)
simplified_data[1:5]
```

Task 9: Instructions

Extract the first four antibiotic days.

- Make a new variable, `num_antibiotic_days`, showing the number of antibiotic days each patient/blood culture day combination had.
- Remove blood culture days with less than four antibiotic days (rows).
- Select the first four days (rows) for each blood culture.

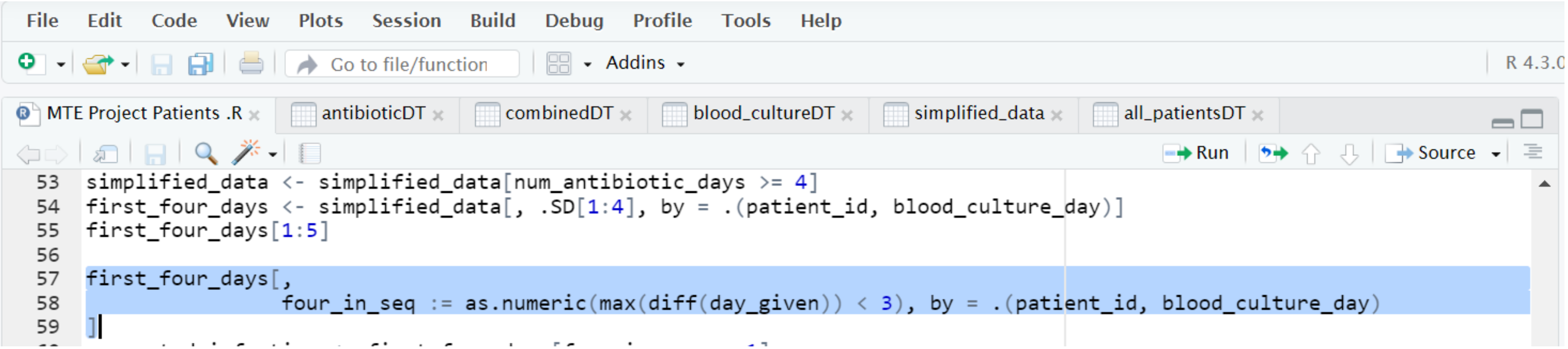


```

simplified_data <- combinedDT[, .(patient_id, blood_culture_day, day_given)]
View(simplified_data)

simplified_data <- unique(simplified_data)
simplified_data[1:5]

simplified_data[, num_antibiotic_days := .N, by = .(patient_id, blood_culture_day)]
simplified_data <- simplified_data[num_antibiotic_days >= 4]
first_four_days <- simplified_data[, .SD[1:4], by = .(patient_id, blood_culture_day)]
first_four_days[1:5]
```

Task 10: Instructions

Find which four-day sequences qualify.

- Make a new 0/1 variable, `four_in_seq`, indicating whether or not the antibiotic sequence has no skips of more than one day.



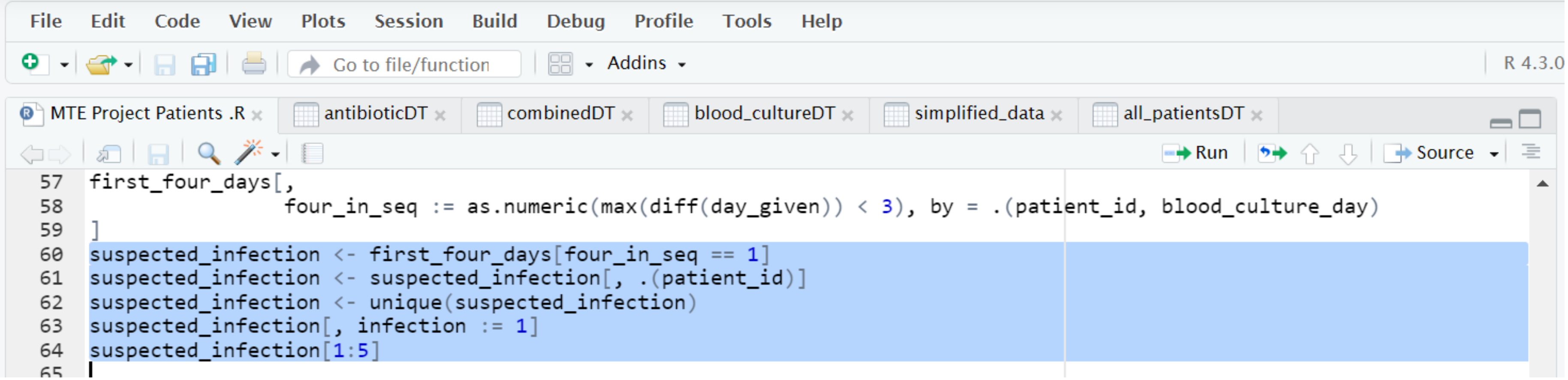
```
simplified_data <- simplified_data[num_antibiotic_days >= 4]
first_four_days <- simplified_data[, .SD[1:4], by = .(patient_id, blood_culture_day)]
first_four_days[1:5]

first_four_days[,
                four_in_seq := as.numeric(max(diff(day_given)) < 3), by = .(patient_id, blood_culture_day)
]
```

Task 11: Instructions

Create a new data frame with one row for each `patient_id` with suspected infection.

- Select the rows which have `four_in_seq` equal to `1`.
- Retain only the `patient_id` column.
- Get rid of duplicates.
- Make a new indicator,`infection`, setting it to `1` for everyone.

To select one column of a data table as a new data table, use `.()` with the column name inside the paratheses.

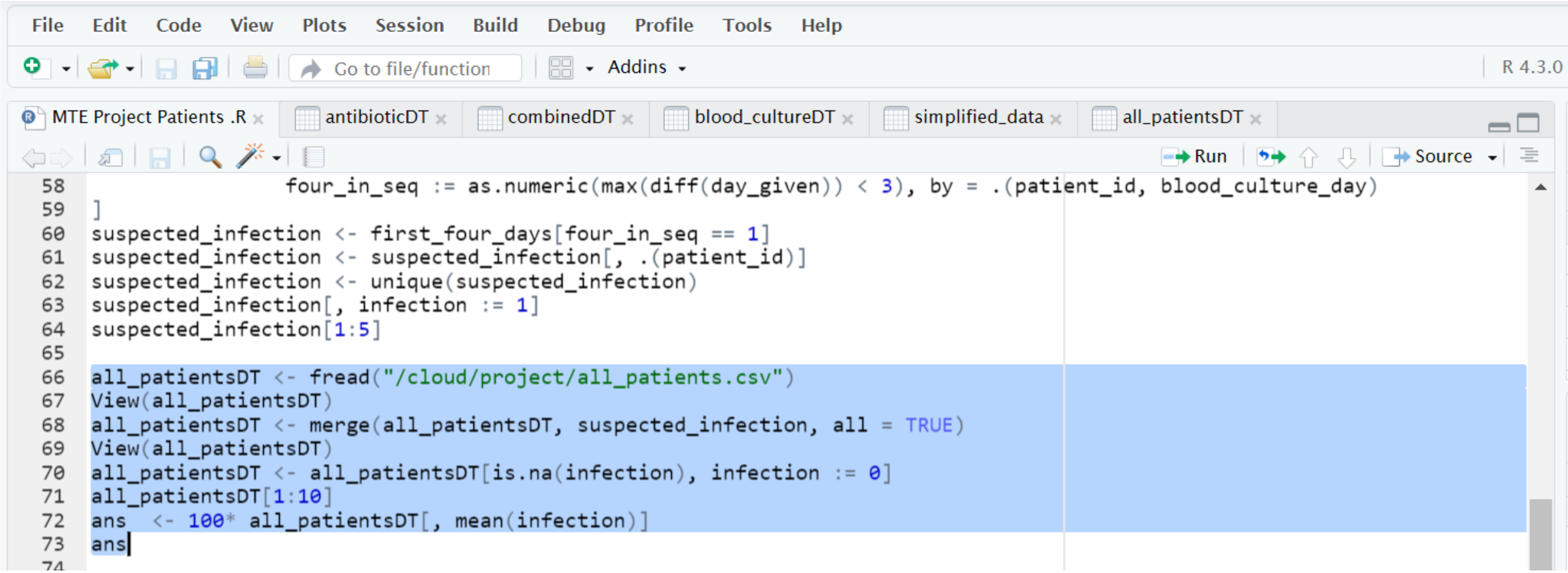


```r
first_four_days[,
                four_in_seq := as.numeric(max(diff(day_given)) < 3), by = .(patient_id, blood_culture_day)
]
suspected_infection <- first_four_days[four_in_seq == 1]
suspected_infection <- suspected_infection[, .(patient_id)]
suspected_infection <- unique(suspected_infection)
suspected_infection[, infection := 1]
suspected_infection[1:5]
```

Task 12: Instructions

Find the percentage of presumed serious infections in the data.

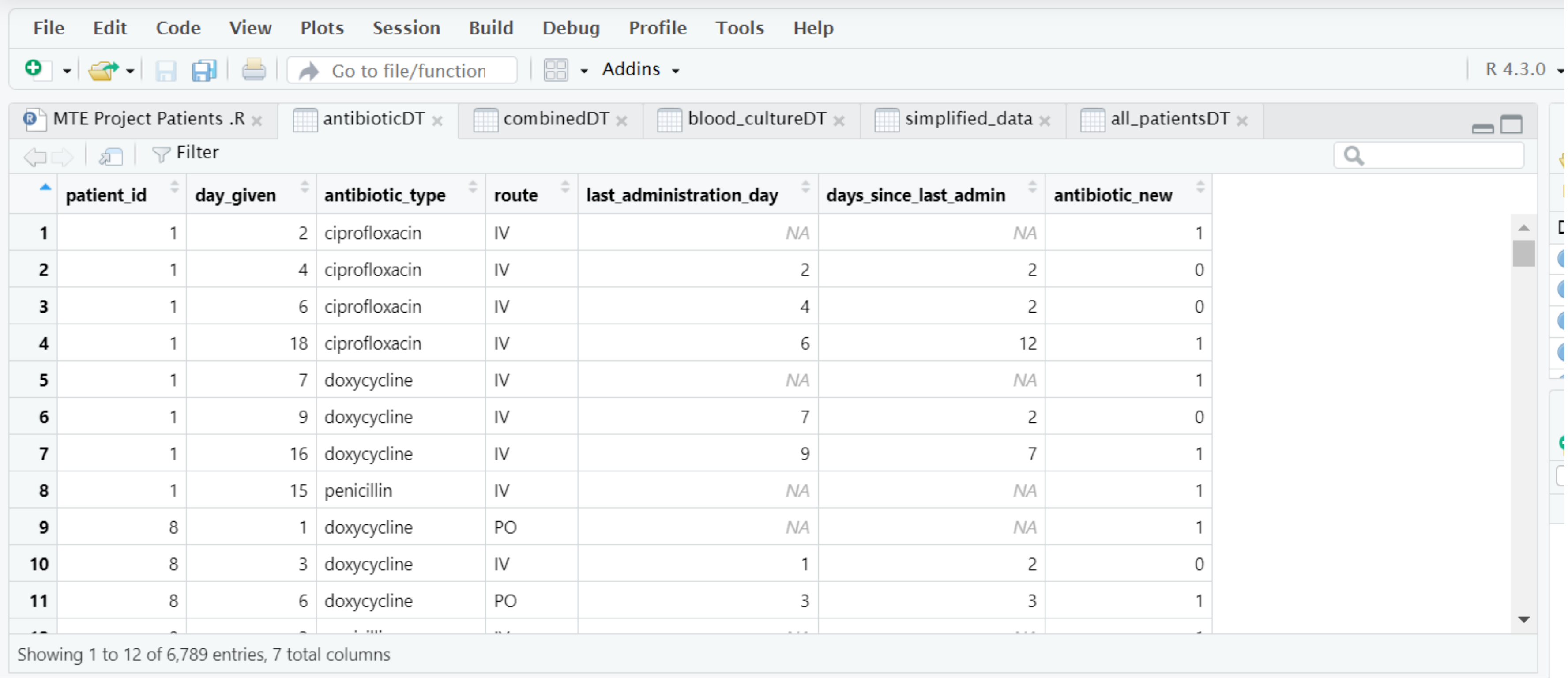
- Use `fread()` to read in `"datasets/all_patients.csv"`, which contains a record of all patients who were in the hospital during the same two-week timeframe.
- Merge this dataset with the infection flag data. Make sure to retain all patients.
- The patients who were not in the antibiotic and blood culture data will have missing values for the infection flag. Set these to 0.
- Calculate the percentage of patients who met the criteria for presumed infection.



```r
                four_in_seq := as.numeric(max(diff(day_given)) < 3), by = .(patient_id, blood_culture_day)
]
suspected_infection <- first_four_days[four_in_seq == 1]
suspected_infection <- suspected_infection[, .(patient_id)]
suspected_infection <- unique(suspected_infection)
suspected_infection[, infection := 1]
suspected_infection[1:5]

all_patientsDT <- fread("/cloud/project/all_patients.csv")
View(all_patientsDT)
all_patientsDT <- merge(all_patientsDT, suspected_infection, all = TRUE)
View(all_patientsDT)
all_patientsDT <- all_patientsDT[is.na(infection), infection := 0]
all_patientsDT[1:10]
ans  <- 100* all_patientsDT[, mean(infection)]
ans
```

Data Tables Formed:

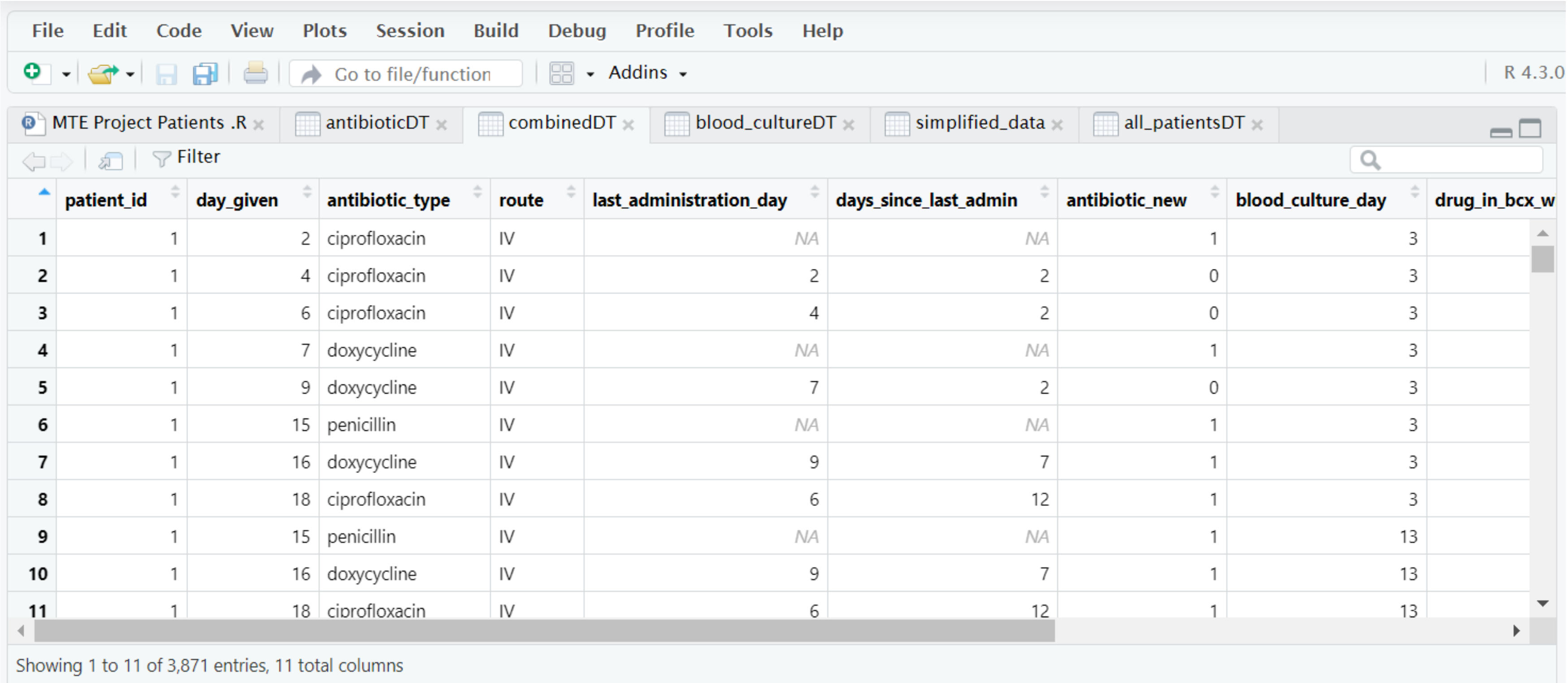
- antibioticDT



	patient_id	day_given	antibiotic_type	route	last_administration_day	days_since_last_admin	antibiotic_new
1	1	2	ciprofloxacin	IV	NA	NA	1
2	1	4	ciprofloxacin	IV	2	2	0
3	1	6	ciprofloxacin	IV	4	2	0
4	1	18	ciprofloxacin	IV	6	12	1
5	1	7	doxycycline	IV	NA	NA	1
6	1	9	doxycycline	IV	7	2	0
7	1	16	doxycycline	IV	9	7	1
8	1	15	penicillin	IV	NA	NA	1
9	8	1	doxycycline	PO	NA	NA	1
10	8	3	doxycycline	IV	1	2	0
11	8	6	doxycycline	PO	3	3	1

Showing 1 to 12 of 6,789 entries, 7 total columns

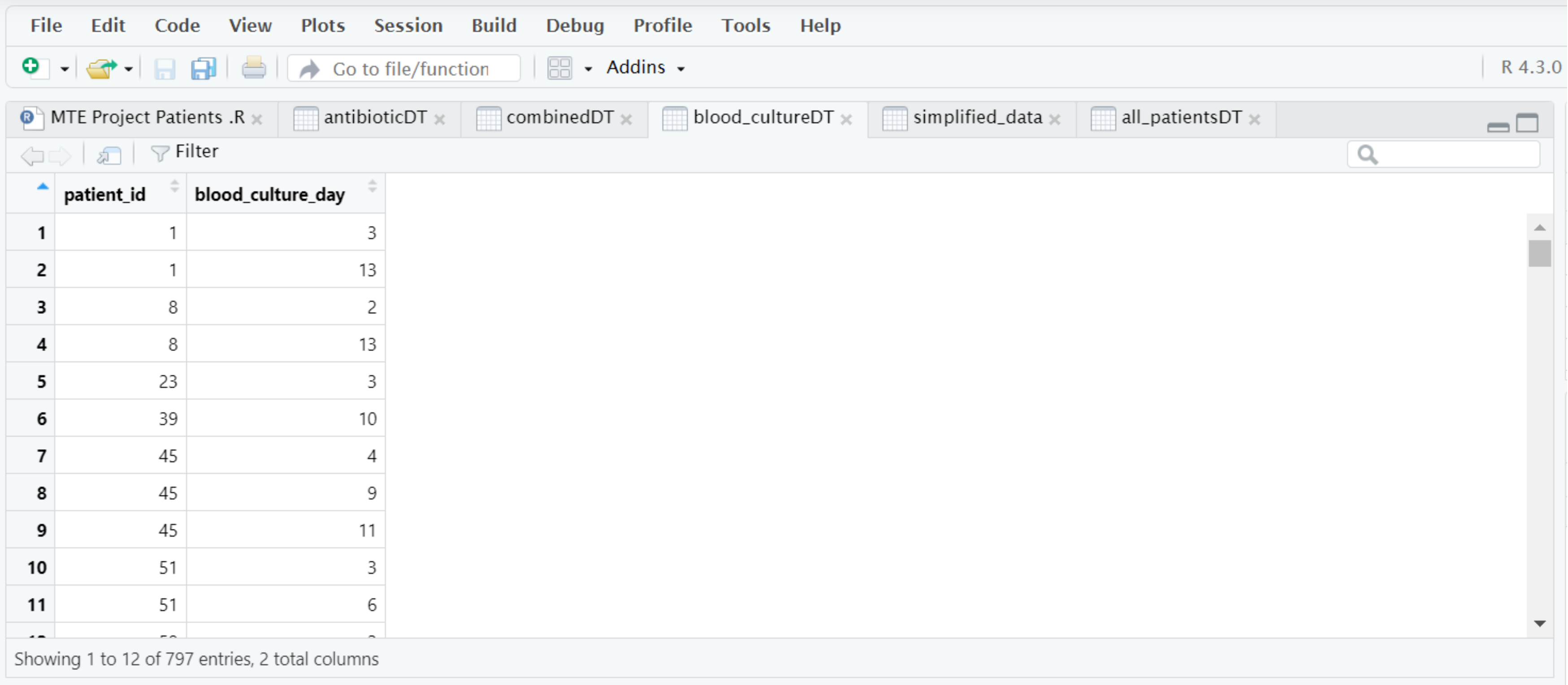
- combinedDT



	patient_id	day_given	antibiotic_type	route	last_administration_day	days_since_last_admin	antibiotic_new	blood_culture_day	drug_in_bcx_w
1	1	2	ciprofloxacin	IV	NA	NA	1	3	
2	1	4	ciprofloxacin	IV	2	2	0	3	
3	1	6	ciprofloxacin	IV	4	2	0	3	
4	1	7	doxycycline	IV	NA	NA	1	3	
5	1	9	doxycycline	IV	7	2	0	3	
6	1	15	penicillin	IV	NA	NA	1	3	
7	1	16	doxycycline	IV	9	7	1	3	
8	1	18	ciprofloxacin	IV	6	12	1	3	
9	1	15	penicillin	IV	NA	NA	1	13	
10	1	16	doxycycline	IV	9	7	1	13	
11	1	18	ciprofloxacin	IV	6	12	1	13	

Showing 1 to 11 of 3,871 entries, 11 total columns

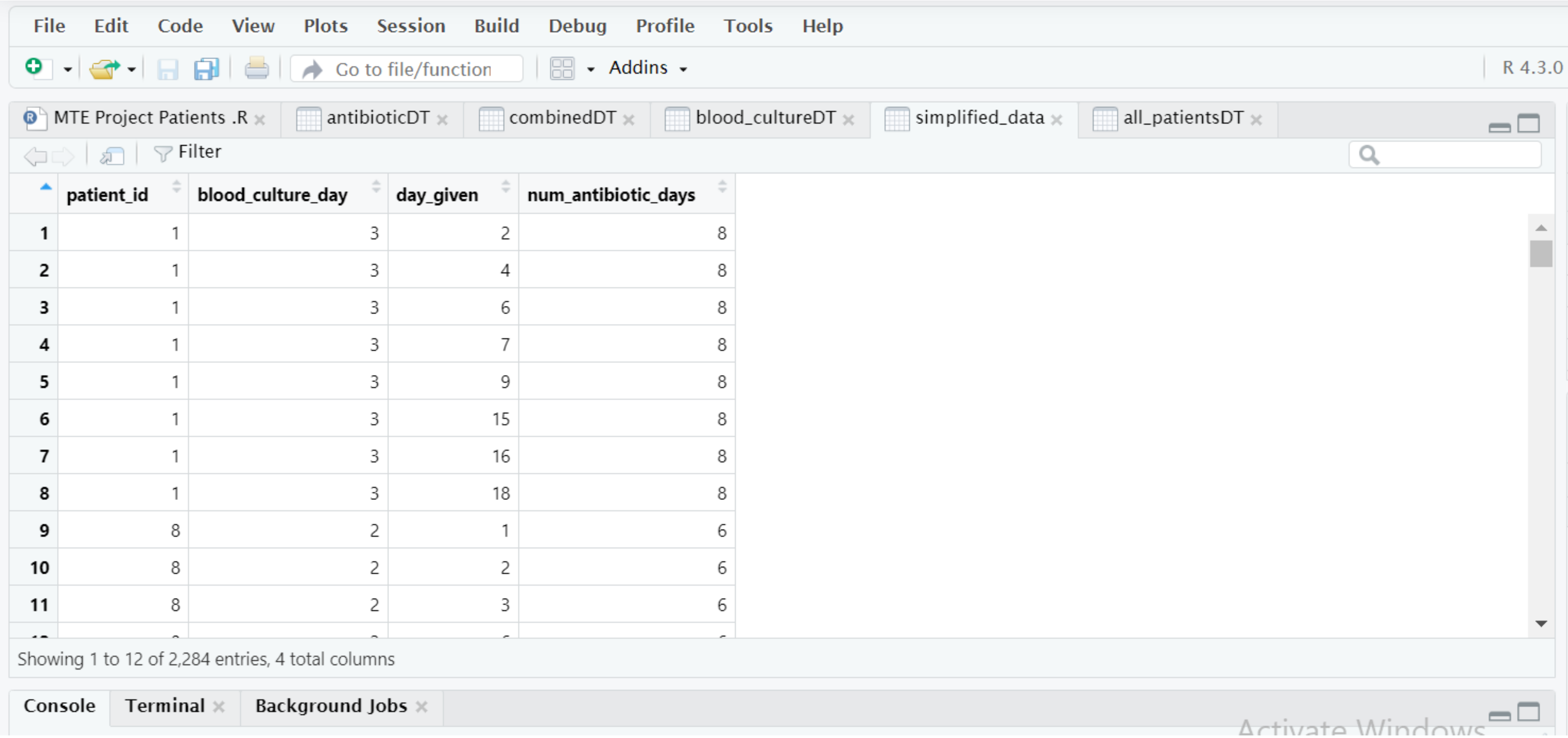
- bloodcultureDT



	patient_id	blood_culture_day
1	1	3
2	1	13
3	8	2
4	8	13
5	23	3
6	39	10
7	45	4
8	45	9
9	45	11
10	51	3
11	51	6

Showing 1 to 12 of 797 entries, 2 total columns

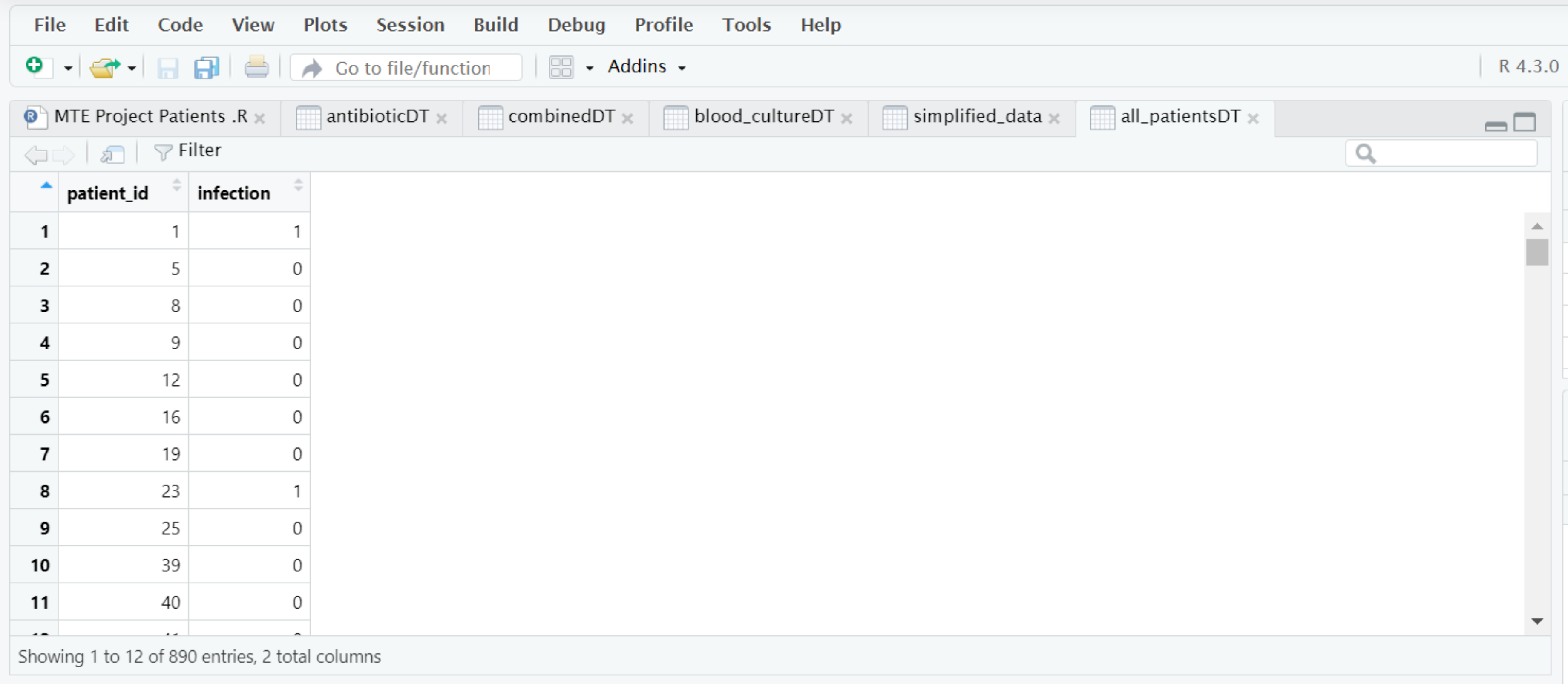
- Simplified_data



	patient_id	blood_culture_day	day_given	num_antibiotic_days
1	1	3	2	8
2	1	3	4	8
3	1	3	6	8
4	1	3	7	8
5	1	3	9	8
6	1	3	15	8
7	1	3	16	8
8	1	3	18	8
9	8	2	1	6
10	8	2	2	6
11	8	2	3	6

Showing 1 to 12 of 2,284 entries, 4 total columns

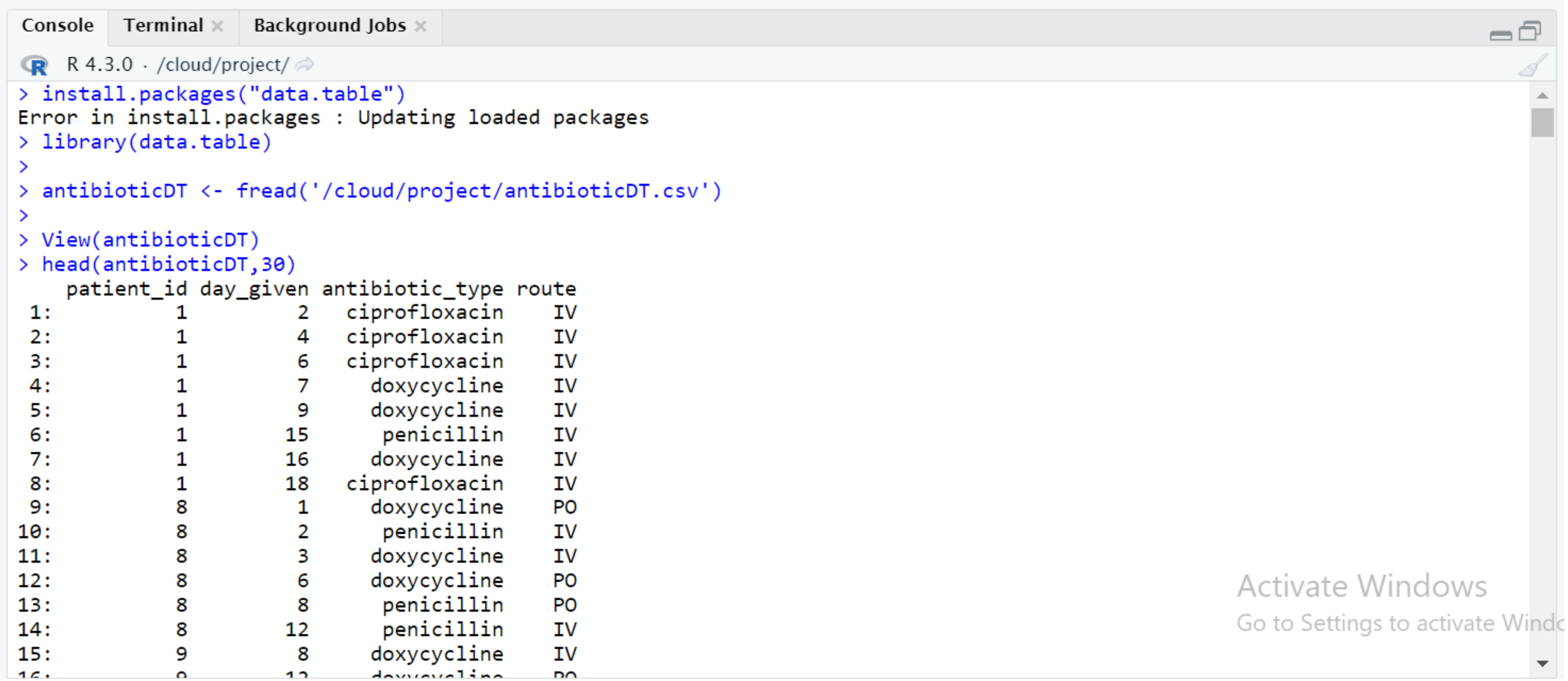
- All_patients



	patient_id	infection
1	1	1
2	5	0
3	8	0
4	9	0
5	12	0
6	16	0
7	19	0
8	23	1
9	25	0
10	39	0
11	40	0

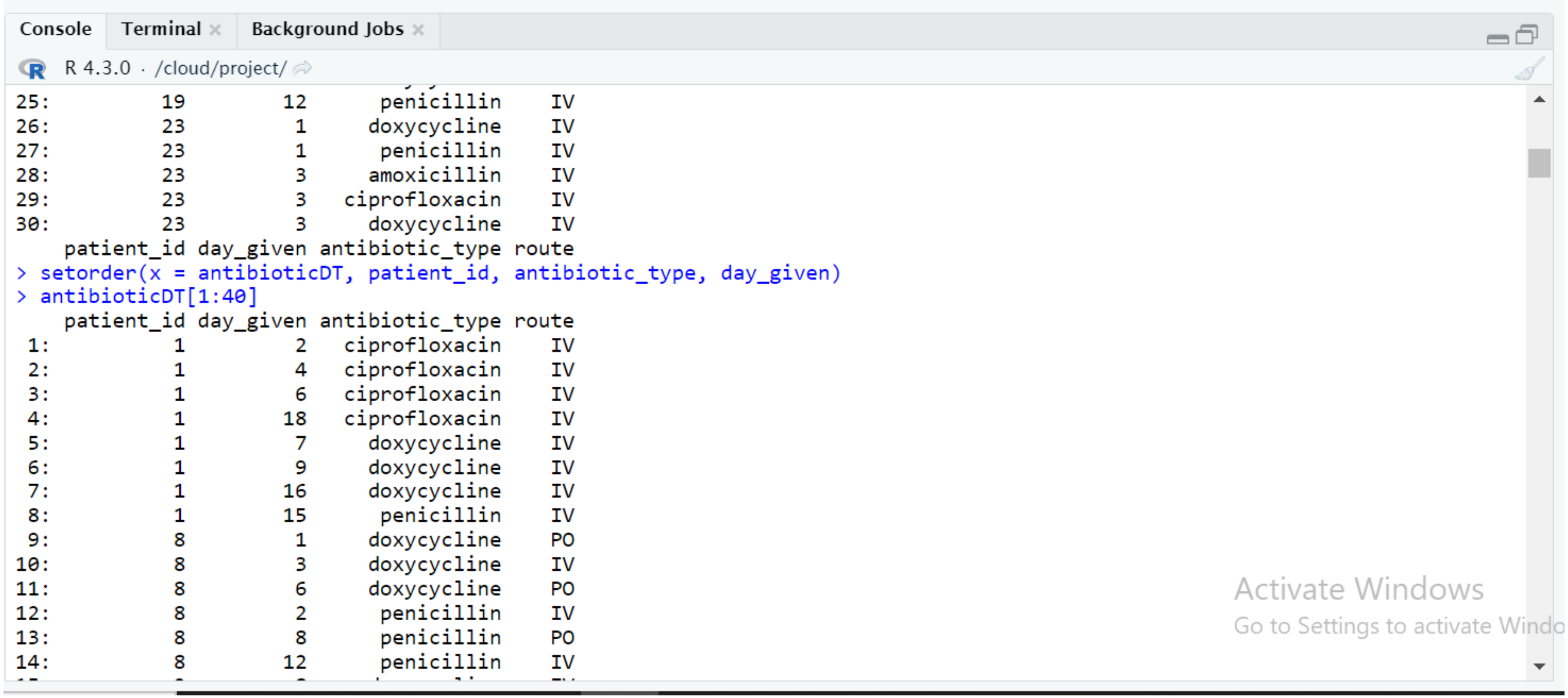
Showing 1 to 12 of 890 entries, 2 total columns

R Console Commands and Outputs(Task1 to 12):



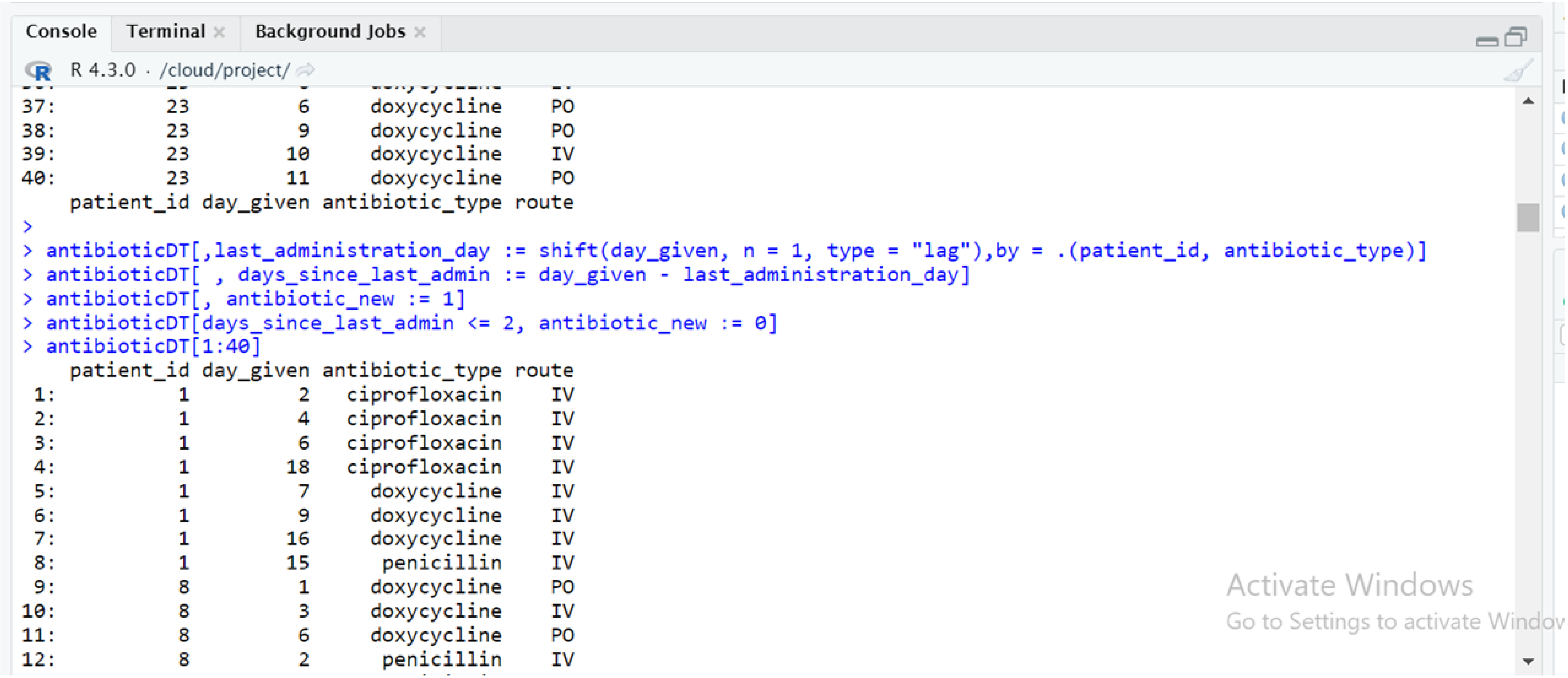
```
> install.packages("data.table")
Error in install.packages : Updating loaded packages
> library(data.table)
>
> antibioticDT <- fread('/cloud/project/antibioticDT.csv')
>
> View(antibioticDT)
> head(antibioticDT,30)
    patient_id day_given antibiotic_type route
 1:          1         2   ciprofloxacin    IV
 2:          1         4   ciprofloxacin    IV
 3:          1         6   ciprofloxacin    IV
 4:          1         7     doxycycline    IV
 5:          1         9     doxycycline    IV
 6:          1        15      penicillin    IV
 7:          1        16     doxycycline    IV
 8:          1        18   ciprofloxacin    IV
 9:          8         1     doxycycline    PO
10:          8         2      penicillin    IV
11:          8         3     doxycycline    IV
12:          8         6     doxycycline    PO
13:          8         8      penicillin    PO
14:          8        12      penicillin    IV
15:          9         8     doxycycline    IV
16:          9        12     doxycycline    PO
```

Activate Windows
Go to Settings to activate Windc

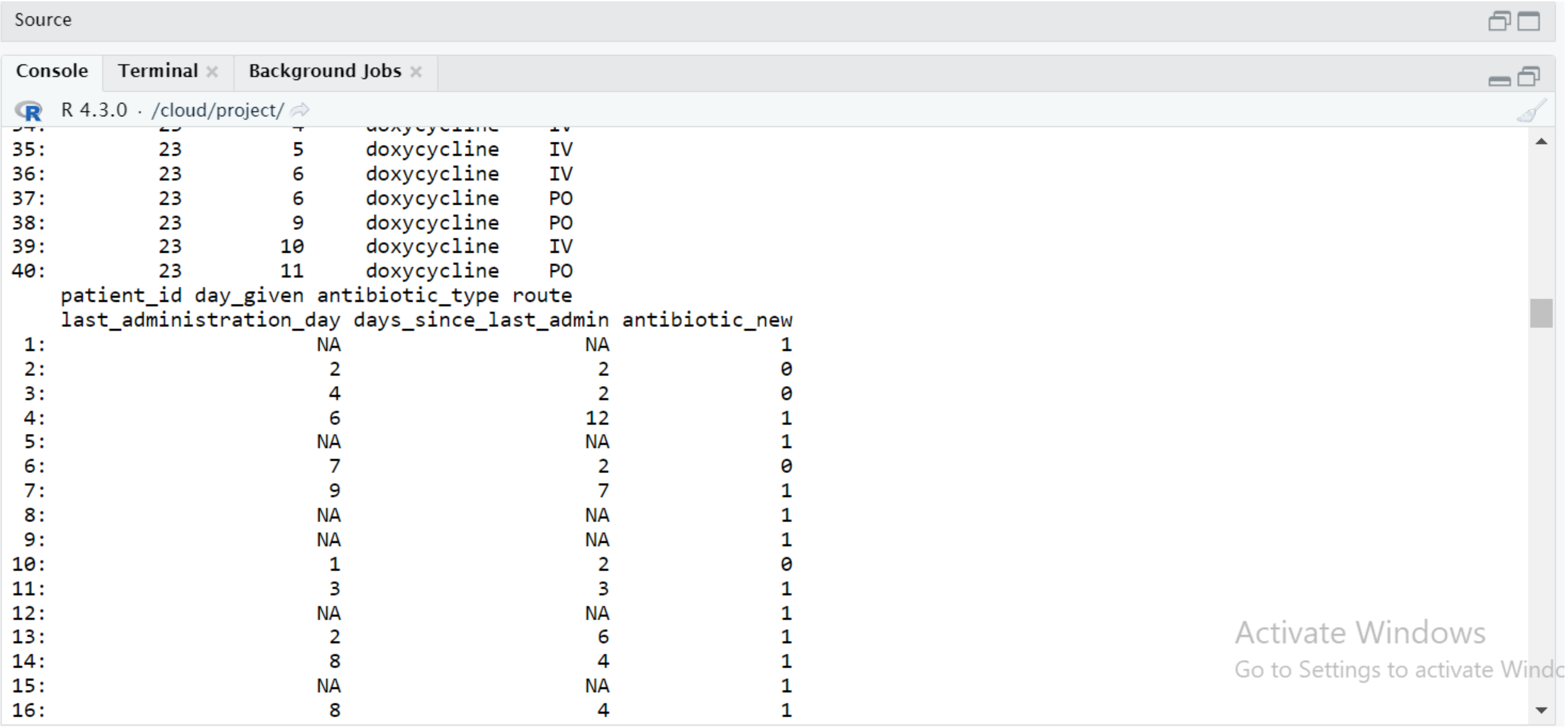


```
25:         19        12      penicillin    IV
26:         23         1     doxycycline    IV
27:         23         1      penicillin    IV
28:         23         3     amoxicillin    IV
29:         23         3   ciprofloxacin    IV
30:         23         3     doxycycline    IV
    patient_id day_given antibiotic_type route
> setorder(x = antibioticDT, patient_id, antibiotic_type, day_given)
> antibioticDT[1:40]
    patient_id day_given antibiotic_type route
 1:          1         2   ciprofloxacin    IV
 2:          1         4   ciprofloxacin    IV
 3:          1         6   ciprofloxacin    IV
 4:          1        18   ciprofloxacin    IV
 5:          1         7     doxycycline    IV
 6:          1         9     doxycycline    IV
 7:          1        16     doxycycline    IV
 8:          1        15      penicillin    IV
 9:          8         1     doxycycline    PO
10:          8         3     doxycycline    IV
11:          8         6     doxycycline    PO
12:          8         2      penicillin    IV
13:          8         8      penicillin    PO
14:          8        12      penicillin    IV
```

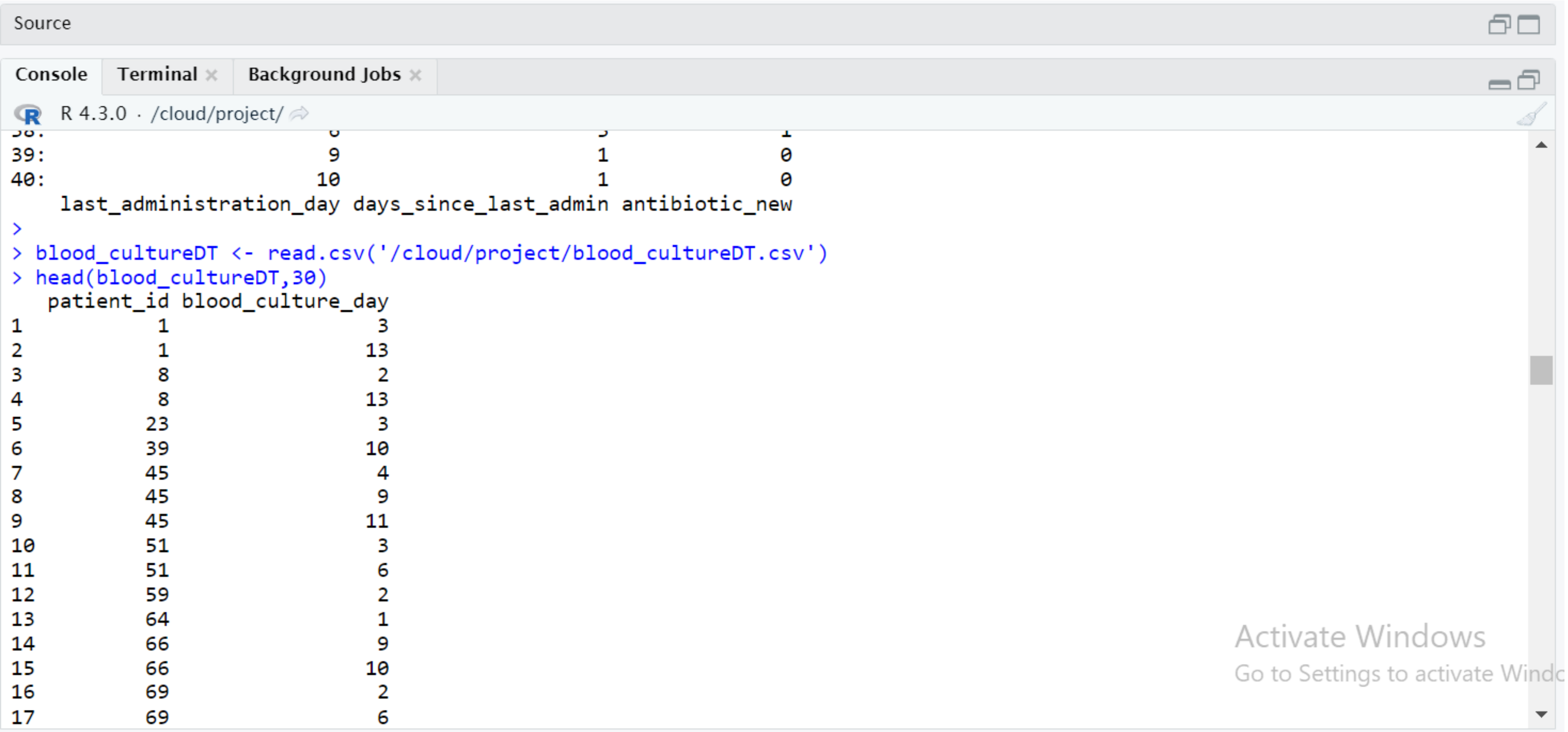
Activate Windows
Go to Settings to activate Windo



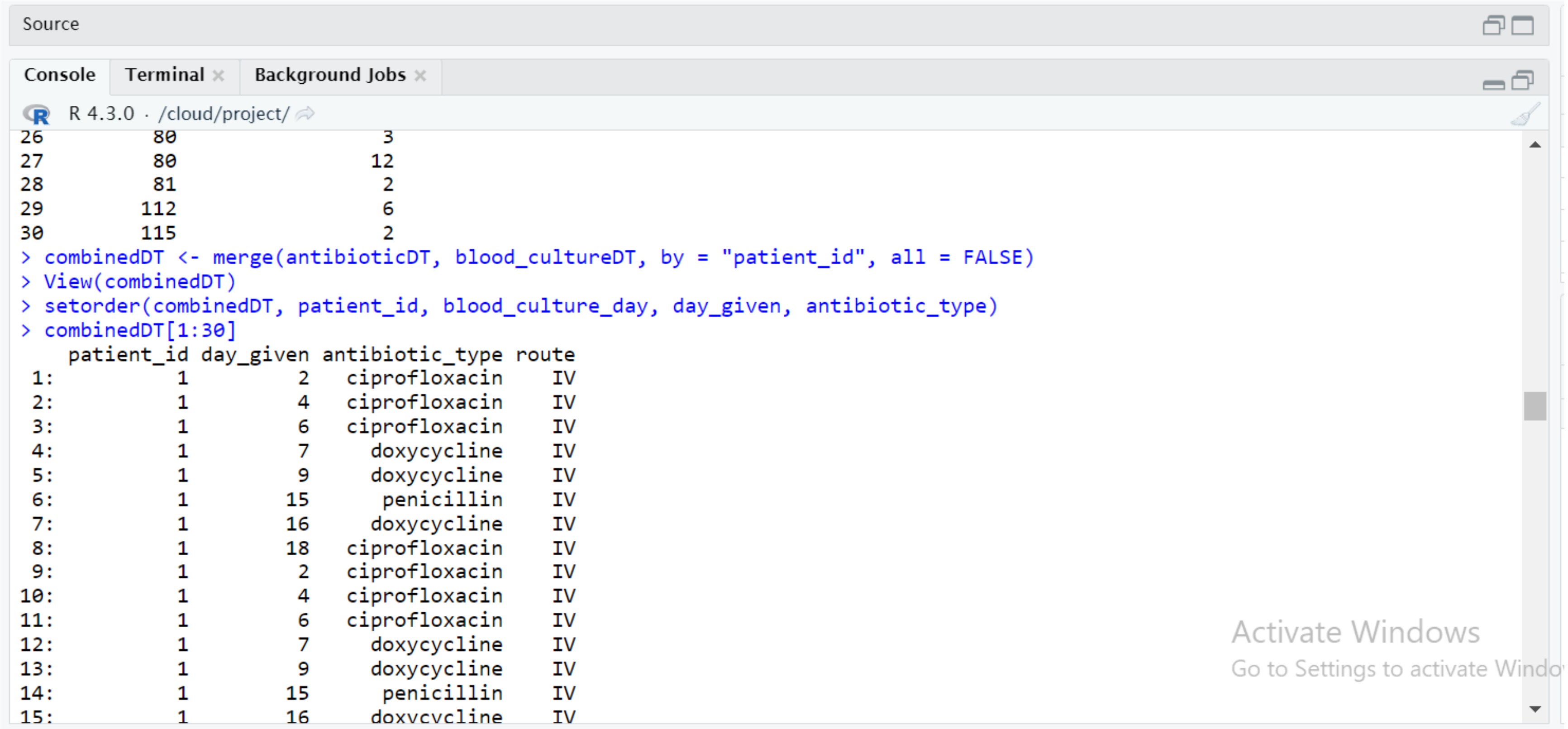
```
37:         23         6     doxycycline    PO
38:         23         9     doxycycline    PO
39:         23        10     doxycycline    IV
40:         23        11     doxycycline    PO
    patient_id day_given antibiotic_type route
>
> antibioticDT[,last_administration_day := shift(day_given, n = 1, type = "lag"),by = .(patient_id, antibiotic_type)]
> antibioticDT[ , days_since_last_admin := day_given - last_administration_day]
> antibioticDT[, antibiotic_new := 1]
> antibioticDT[days_since_last_admin <= 2, antibiotic_new := 0]
> antibioticDT[1:40]
    patient_id day_given antibiotic_type route
 1:          1         2   ciprofloxacin    IV
 2:          1         4   ciprofloxacin    IV
 3:          1         6   ciprofloxacin    IV
 4:          1        18   ciprofloxacin    IV
 5:          1         7     doxycycline    IV
 6:          1         9     doxycycline    IV
 7:          1        16     doxycycline    IV
 8:          1        15      penicillin    IV
 9:          8         1     doxycycline    PO
10:          8         3     doxycycline    IV
11:          8         6     doxycycline    PO
12:          8         2      penicillin    IV
```



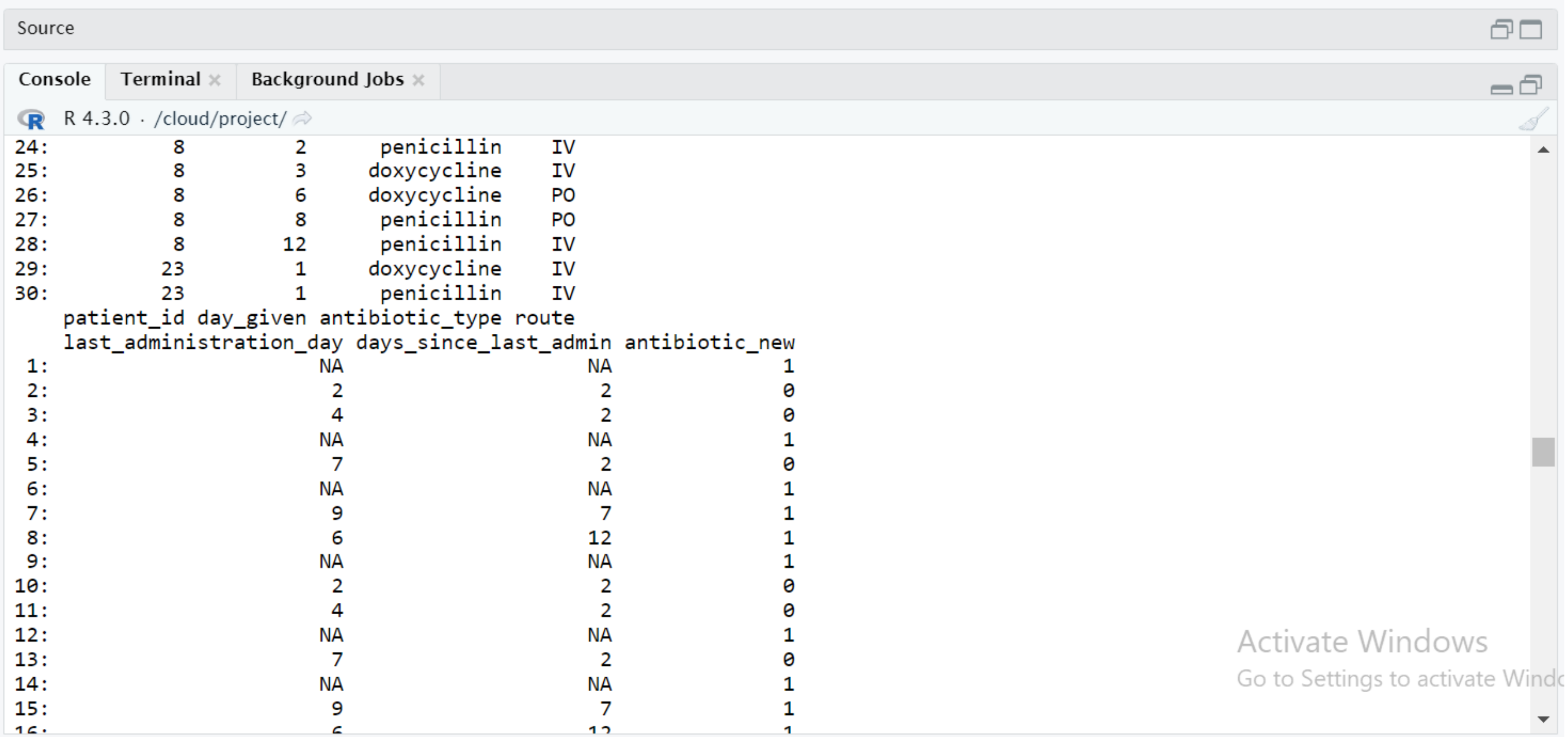
```
34:         23         4     doxycycline    IV
35:         23         5     doxycycline    IV
36:         23         6     doxycycline    IV
37:         23         6     doxycycline    PO
38:         23         9     doxycycline    PO
39:         23        10     doxycycline    IV
40:         23        11     doxycycline    PO
    patient_id day_given antibiotic_type route
    last_administration_day days_since_last_admin antibiotic_new
 1:                      NA                    NA              1
 2:                       2                     2              0
 3:                       4                     2              0
 4:                       6                    12              1
 5:                      NA                    NA              1
 6:                       7                     2              0
 7:                       9                     7              1
 8:                      NA                    NA              1
 9:                      NA                    NA              1
10:                       1                     2              0
11:                       3                     3              1
12:                      NA                    NA              1
13:                       2                     6              1
14:                       8                     4              1
15:                      NA                    NA              1
16:                       8                     4              1
```



```
39:                        9                     1              0
40:                       10                     1              0
    last_administration_day days_since_last_admin antibiotic_new
>
> blood_cultureDT <- read.csv('/cloud/project/blood_cultureDT.csv')
> head(blood_cultureDT,30)
   patient_id blood_culture_day
1           1                 3
2           1                13
3           8                 2
4           8                13
5          23                 3
6          39                10
7          45                 4
8          45                 9
9          45                11
10         51                 3
11         51                 6
12         59                 2
13         64                 1
14         66                 9
15         66                10
16         69                 2
17         69                 6
```

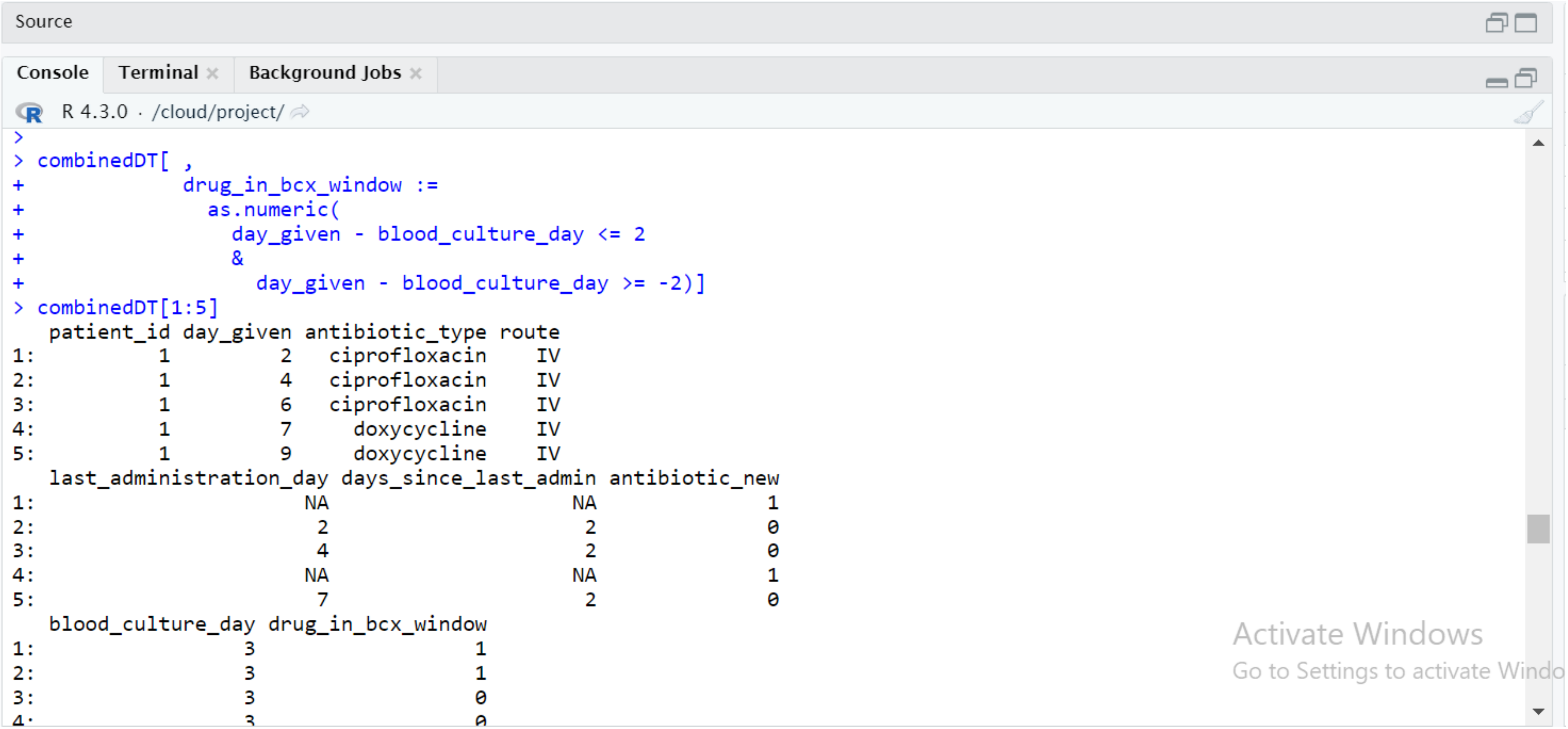


```
26          80               3
27          80              12
28          81               2
29         112               6
30         115               2
> combinedDT <- merge(antibioticDT, blood_cultureDT, by = "patient_id", all = FALSE)
> View(combinedDT)
> setorder(combinedDT, patient_id, blood_culture_day, day_given, antibiotic_type)
> combinedDT[1:30]
    patient_id day_given antibiotic_type route
 1:          1         2   ciprofloxacin    IV
 2:          1         4   ciprofloxacin    IV
 3:          1         6   ciprofloxacin    IV
 4:          1         7     doxycycline    IV
 5:          1         9     doxycycline    IV
 6:          1        15      penicillin    IV
 7:          1        16     doxycycline    IV
 8:          1        18   ciprofloxacin    IV
 9:          1         2   ciprofloxacin    IV
10:          1         4   ciprofloxacin    IV
11:          1         6   ciprofloxacin    IV
12:          1         7     doxycycline    IV
13:          1         9     doxycycline    IV
14:          1        15      penicillin    IV
15:          1        16     doxycycline    IV
```

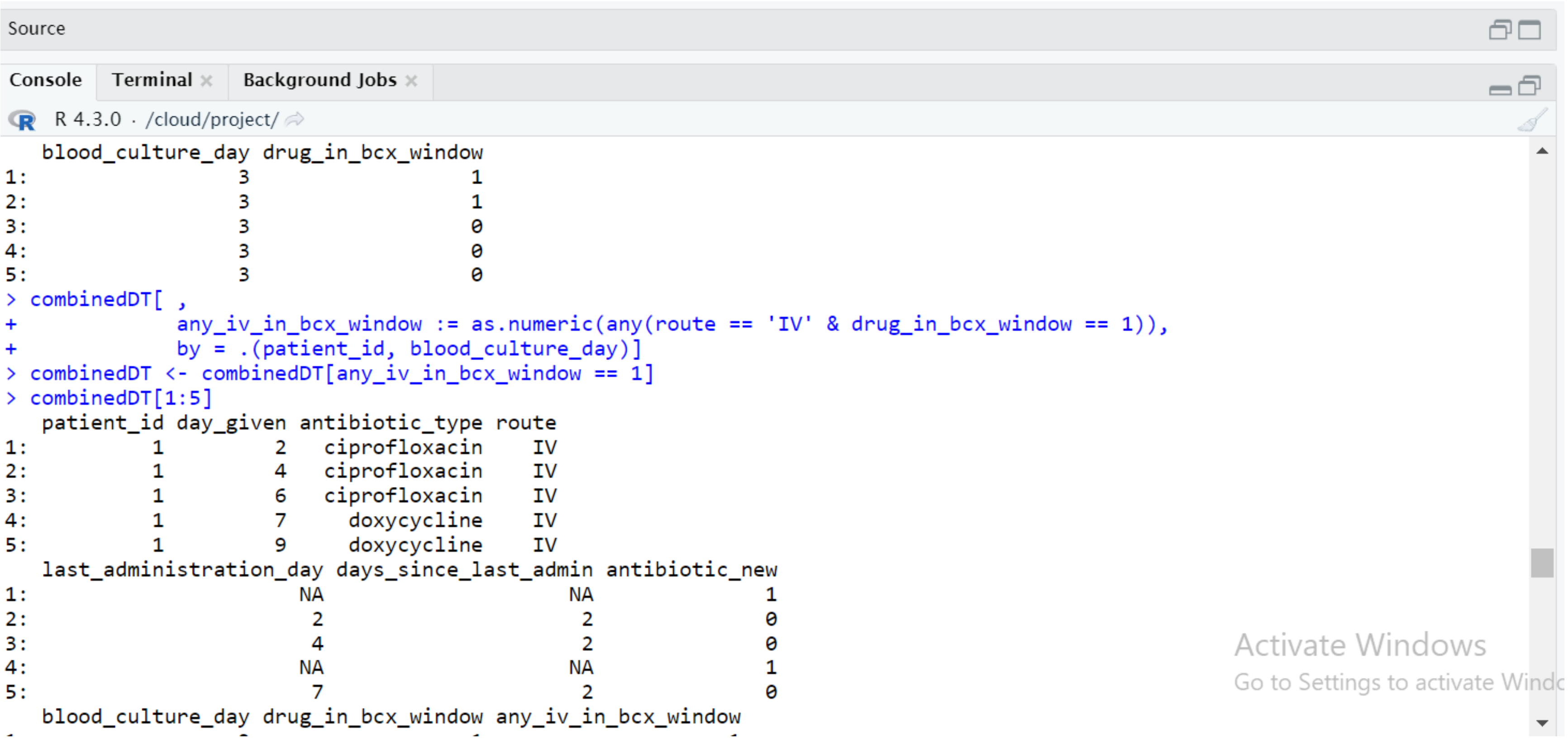


```
24:          8         2      penicillin    IV
25:          8         3     doxycycline    IV
26:          8         6     doxycycline    PO
27:          8         8      penicillin    PO
28:          8        12      penicillin    IV
29:         23         1     doxycycline    IV
30:         23         1      penicillin    IV
    patient_id day_given antibiotic_type route
    last_administration_day days_since_last_admin antibiotic_new
 1:                      NA                    NA              1
 2:                       2                     2              0
 3:                       4                     2              0
 4:                      NA                    NA              1
 5:                       7                     2              0
 6:                      NA                    NA              1
 7:                       9                     7              1
 8:                       6                    12              1
 9:                      NA                    NA              1
10:                       2                     2              0
11:                       4                     2              0
12:                      NA                    NA              1
13:                       7                     2              0
14:                      NA                    NA              1
15:                       9                     7              1
16:                       6                    12              1
```

Activate Windows
Go to Settings to activate Windo

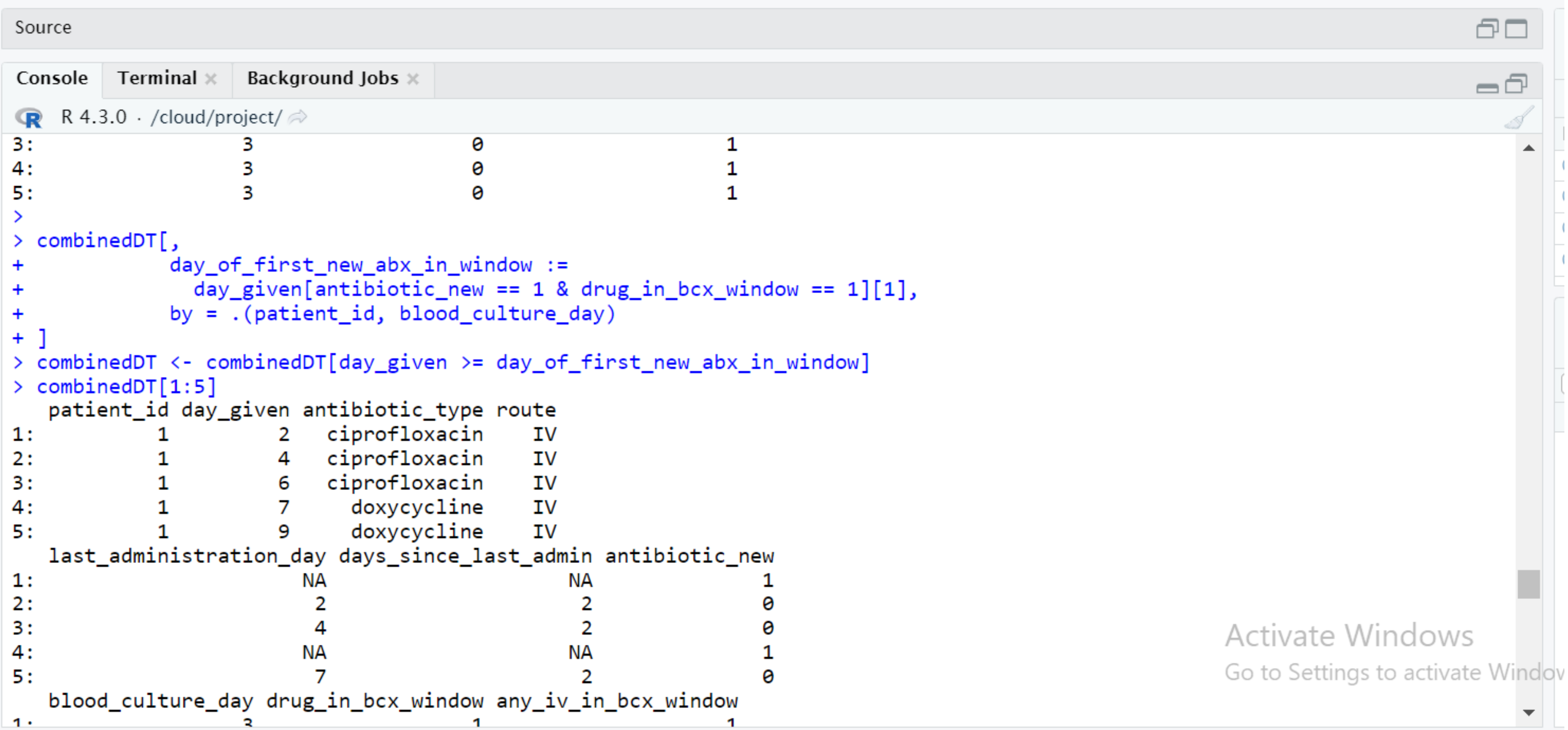


```
>
> combinedDT[ ,
+            drug_in_bcx_window :=
+              as.numeric(
+                day_given - blood_culture_day <= 2
+                &
+                  day_given - blood_culture_day >= -2)]
> combinedDT[1:5]
   patient_id day_given antibiotic_type route
1:          1         2   ciprofloxacin    IV
2:          1         4   ciprofloxacin    IV
3:          1         6   ciprofloxacin    IV
4:          1         7     doxycycline    IV
5:          1         9     doxycycline    IV
   last_administration_day days_since_last_admin antibiotic_new
1:                      NA                    NA              1
2:                       2                     2              0
3:                       4                     2              0
4:                      NA                    NA              1
5:                       7                     2              0
   blood_culture_day drug_in_bcx_window
1:                 3                  1
2:                 3                  1
3:                 3                  0
4:                 3                  0
```

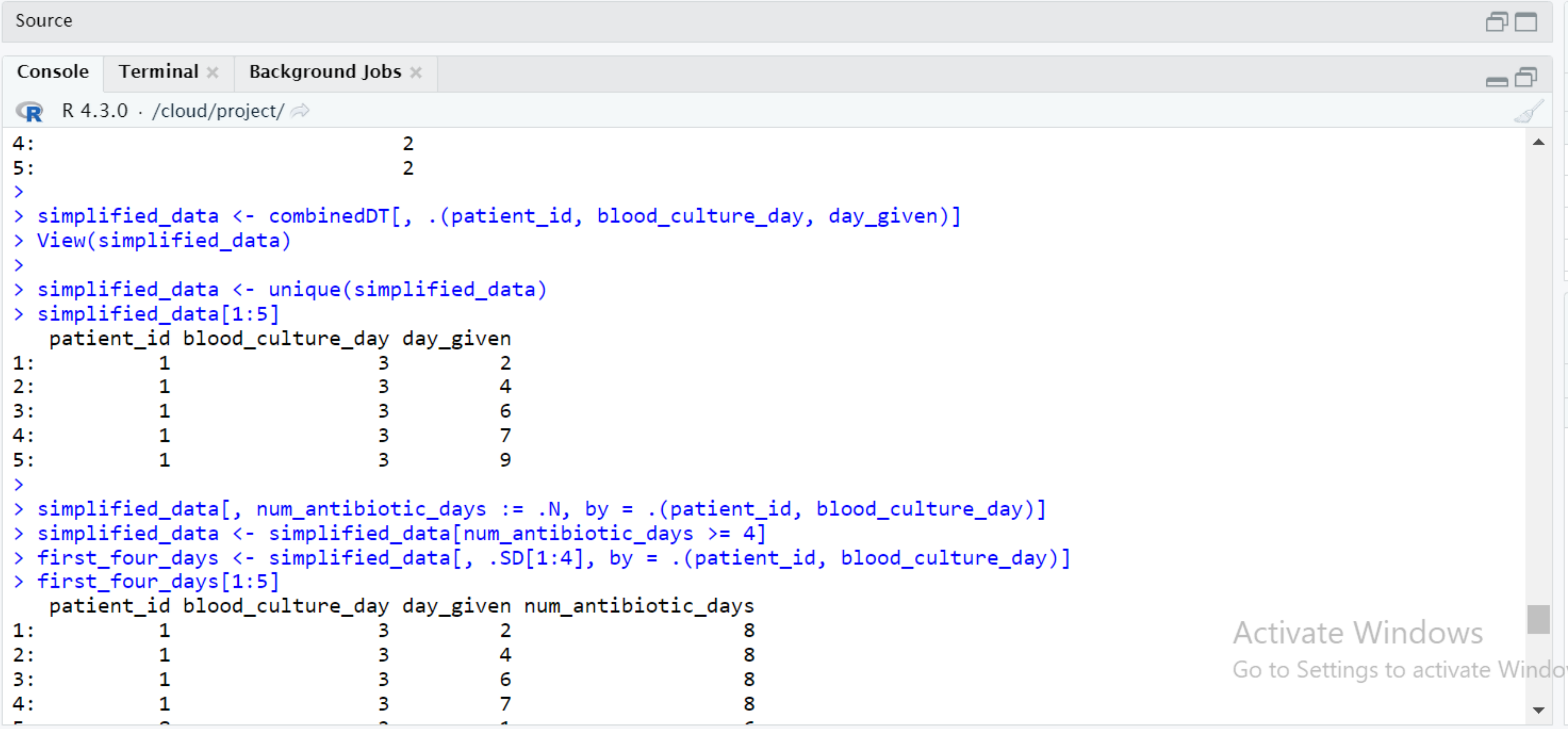


```
   blood_culture_day drug_in_bcx_window
1:                 3                  1
2:                 3                  1
3:                 3                  0
4:                 3                  0
5:                 3                  0
> combinedDT[ ,
+             any_iv_in_bcx_window := as.numeric(any(route == 'IV' & drug_in_bcx_window == 1)),
+             by = .(patient_id, blood_culture_day)]
> combinedDT <- combinedDT[any_iv_in_bcx_window == 1]
> combinedDT[1:5]
   patient_id day_given antibiotic_type route
1:          1         2   ciprofloxacin    IV
2:          1         4   ciprofloxacin    IV
3:          1         6   ciprofloxacin    IV
4:          1         7     doxycycline    IV
5:          1         9     doxycycline    IV
   last_administration_day days_since_last_admin antibiotic_new
1:                      NA                    NA              1
2:                       2                     2              0
3:                       4                     2              0
4:                      NA                    NA              1
5:                       7                     2              0
   blood_culture_day drug_in_bcx_window any_iv_in_bcx_window
```

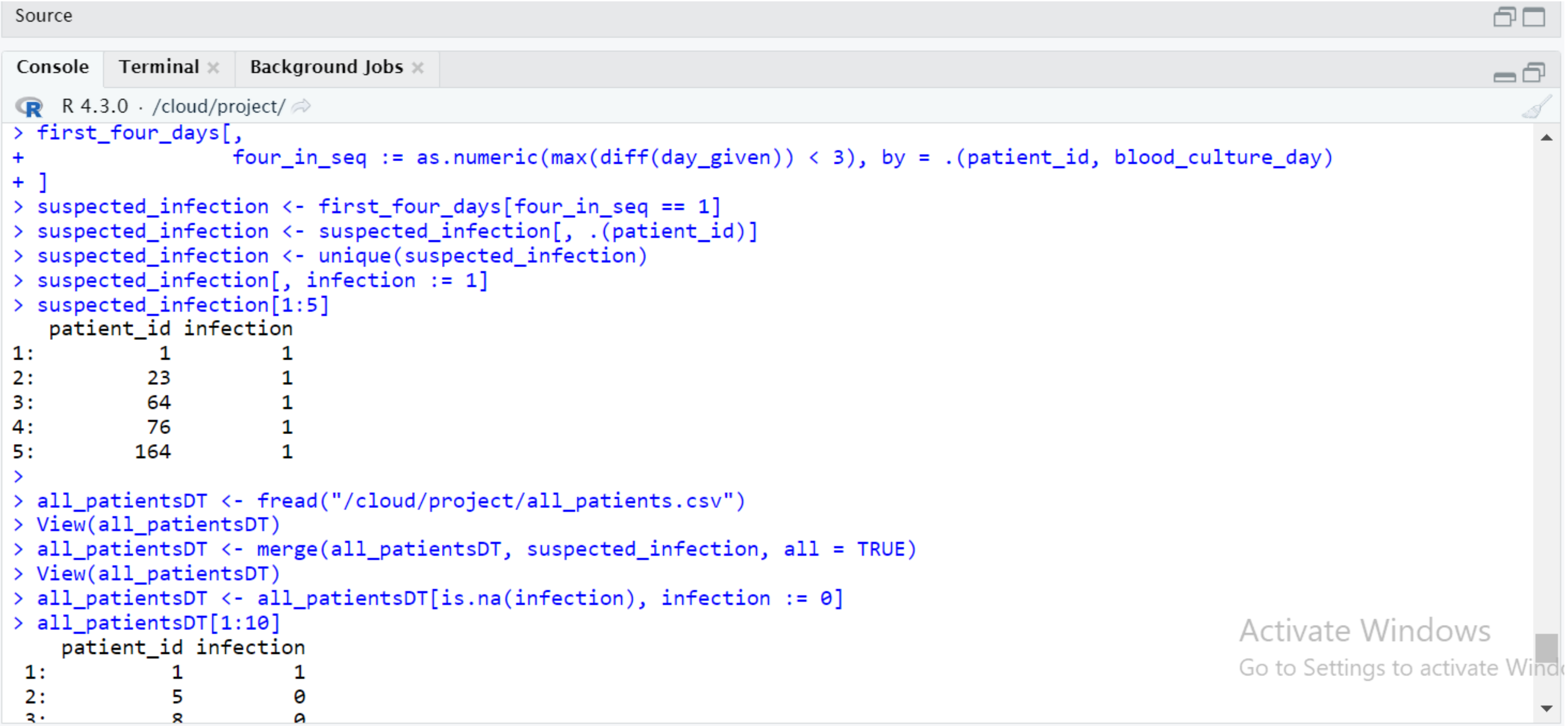
Activate Windows
Go to Settings to activate Windc



```
3:                 3                  0                    1
4:                 3                  0                    1
5:                 3                  0                    1
>
> combinedDT[,
+            day_of_first_new_abx_in_window :=
+              day_given[antibiotic_new == 1 & drug_in_bcx_window == 1][1],
+            by = .(patient_id, blood_culture_day)
+ ]
> combinedDT <- combinedDT[day_given >= day_of_first_new_abx_in_window]
> combinedDT[1:5]
   patient_id day_given antibiotic_type route
1:          1         2   ciprofloxacin    IV
2:          1         4   ciprofloxacin    IV
3:          1         6   ciprofloxacin    IV
4:          1         7     doxycycline    IV
5:          1         9     doxycycline    IV
   last_administration_day days_since_last_admin antibiotic_new
1:                      NA                    NA              1
2:                       2                     2              0
3:                       4                     2              0
4:                      NA                    NA              1
5:                       7                     2              0
   blood_culture_day drug_in_bcx_window any_iv_in_bcx_window
1:                 3                  1                    1
```

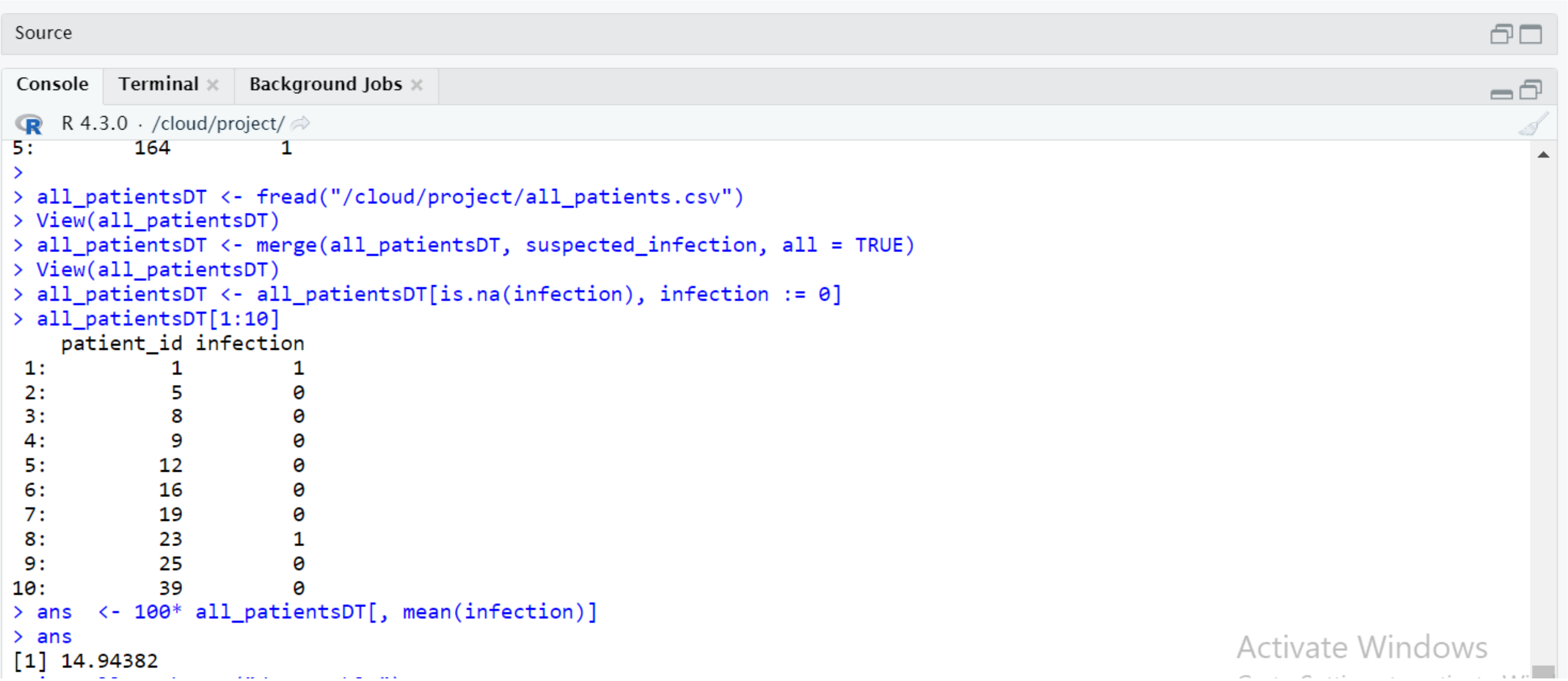


```
4:                              2
5:                              2
>
> simplified_data <- combinedDT[, .(patient_id, blood_culture_day, day_given)]
> View(simplified_data)
>
> simplified_data <- unique(simplified_data)
> simplified_data[1:5]
   patient_id blood_culture_day day_given
1:          1                 3         2
2:          1                 3         4
3:          1                 3         6
4:          1                 3         7
5:          1                 3         9
>
> simplified_data[, num_antibiotic_days := .N, by = .(patient_id, blood_culture_day)]
> simplified_data <- simplified_data[num_antibiotic_days >= 4]
> first_four_days <- simplified_data[, .SD[1:4], by = .(patient_id, blood_culture_day)]
> first_four_days[1:5]
   patient_id blood_culture_day day_given num_antibiotic_days
1:          1                 3         2                   8
2:          1                 3         4                   8
3:          1                 3         6                   8
4:          1                 3         7                   8
```



```
> first_four_days[,
+                 four_in_seq := as.numeric(max(diff(day_given)) < 3), by = .(patient_id, blood_culture_day)
+ ]
> suspected_infection <- first_four_days[four_in_seq == 1]
> suspected_infection <- suspected_infection[, .(patient_id)]
> suspected_infection <- unique(suspected_infection)
> suspected_infection[, infection := 1]
> suspected_infection[1:5]
   patient_id infection
1:          1         1
2:         23         1
3:         64         1
4:         76         1
5:        164         1
>
> all_patientsDT <- fread("/cloud/project/all_patients.csv")
> View(all_patientsDT)
> all_patientsDT <- merge(all_patientsDT, suspected_infection, all = TRUE)
> View(all_patientsDT)
> all_patientsDT <- all_patientsDT[is.na(infection), infection := 0]
> all_patientsDT[1:10]
    patient_id infection
 1:          1         1
 2:          5         0
 3:          8         0
```

Activate Windows
Go to Settings to activate Wind



```
5:        164          1
>
> all_patientsDT <- fread("/cloud/project/all_patients.csv")
> View(all_patientsDT)
> all_patientsDT <- merge(all_patientsDT, suspected_infection, all = TRUE)
> View(all_patientsDT)
> all_patientsDT <- all_patientsDT[is.na(infection), infection := 0]
> all_patientsDT[1:10]
    patient_id infection
 1:          1         1
 2:          5         0
 3:          8         0
 4:          9         0
 5:         12         0
 6:         16         0
 7:         19         0
 8:         23         1
 9:         25         0
10:         39         0
> ans  <- 100* all_patientsDT[, mean(infection)]
> ans
[1] 14.94382
```

Activate Windows

CONCLUSION AND FUTURE SCOPE:

- In conclusion, the project "Classify Suspected Infection in Patients" has successfully developed a machine learning-based model for accurately classifying suspected infections in patients. Through rigorous data analysis, model training, and evaluation, we have achieved the objective of improving early identification of infections, enabling timely interventions and treatment.
- The developed model has demonstrated robust performance, achieving high accuracy, precision, recall, and F1-score in classifying infected and non-infected patients. By leveraging machine learning algorithms and utilizing relevant features, we have created a reliable tool that can assist healthcare professionals in making accurate and efficient decisions, ultimately leading to improved patient outcomes.
- The project's findings highlight the effectiveness of machine learning techniques in infection classification and emphasize the importance of early detection for prompt treatment and prevention of complications. By integrating the developed model into healthcare systems, we can enhance the diagnostic process, reduce the risk of misdiagnosis, and provide timely interventions to patients suspected of having infections.

Future Scope:

- While the project has achieved its immediate objectives, there are several avenues for future exploration and enhancement in the field of classifying suspected infections in patients:
- Integration with real-time data sources: Incorporating real-time data, such as vital signs, laboratory results, and clinical notes, can enhance the model's performance and enable dynamic and accurate infection classification.
- Expansion to different infection types: The current project primarily focuses on classifying suspected infections in a general sense. Future research can delve into specific infection types, such as respiratory tract infections, urinary tract infections, or bloodstream infections, to develop specialized models for each type.
- Multimodal data analysis: Integrating multiple data sources, including imaging data, genetic information, and patient history, can provide a comprehensive view of infections and improve classification accuracy. Exploring multimodal data analysis techniques can unlock new insights and diagnostic capabilities.

- Deployment in healthcare settings: Implementing the developed model in real healthcare environments and assessing its impact on clinical decision-making, patient outcomes, and healthcare resource utilization will be valuable for validating its effectiveness and utility in real-world scenarios.
- Continuous model refinement and updates: The field of machine learning and healthcare is constantly evolving. It is essential to continually update and refine the developed model by incorporating new data, adopting state-of-the-art algorithms, and addressing emerging challenges to ensure its relevance and reliability over time.
- Ethical considerations and fairness: Further exploration of ethical considerations, such as privacy, bias, and interpretability, is necessary to ensure that the developed model adheres to ethical guidelines and provides fair and equitable healthcare services to diverse patient populations.
- By pursuing these future avenues, we can advance the field of infection classification, improve patient care, and contribute to the development of intelligent healthcare systems that aid in early diagnosis and treatment of infections, ultimately saving lives and improving overall healthcare outcomes.

REFERENCES

- **Data Collection**

The following website has been referred to obtain the input data and statistics:

❖ https://drive.google.com/file/d/162KBAQEUU6b84LIgj8MU3a7E6Yrel_bg/view?usp=sharing

- **Programming References**

The following websites have been referred for R coding and Shiny tutorials:

- https://datascienceplus.com/category/programming
- https://rstudio.com/resources/webinars/
- https://bookdown.org/yihui/rmarkdown/document-templates.html
- https://datascienceplus.com/map-visualization-of-covid19-across-world/
- https://rmarkdown.rstudio.com/lesson-12.html
- http://www.htmlwidgets.org/showcase_leaflet.html
- http://jeffgoldsmith.com/p8105_f2017/shiny.html
- https://rmarkdown.rstudio.com/flexdashboard/using.html#page_icons