Final Report of Traineeship Program

On

“Classify Suspected Infection in Patients”

MEDTOUREASY

27th June 2024

TABLE OF CONTENTS

ACKNOWLEDGMENT

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 to have 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 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 client satisfaction and also for spending 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.

Abstract

Sepsis is a critical condition responsible for a significant number of in-hospital fatalities. It occurs when an individual's organs shut down in response to a severe infection, posing a major public health challenge and a crucial area for research. This R project aims to identify hospital patients with severe infections by analyzing medical record data. Leveraging the data.table package in R, we will conduct comprehensive data analysis to uncover patterns and insights related to sepsis.

The project will involve tasks such as data loading, cleaning, and manipulation using data.table functions including the := operator, grouped aggregations using by, and the shift function. By identifying new antibiotic administrations and correlating them with blood culture data, we aim to pinpoint instances of severe infection. This detailed analysis will help in understanding the onset and progression of sepsis, providing valuable insights for healthcare professionals.

Through this project, we will create an efficient and effective workflow for analyzing hospital records, ultimately contributing to better identification and management of sepsis cases. The findings from this project will be instrumental in enhancing patient care and developing targeted strategies to reduce the incidence and mortality associated with sepsis.

INTRODUCTION

1.1 About MedTourEasy

MedTourEasy , a global healthcare company , provides you the informational resources needed to evaluate your global options. 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.

1.2 Project Description

Sepsis is a serious and often fatal condition that happens when a severe infection causes a person's organs to shut down. It is responsible for many in-hospital deaths, making it a significant public health concern. This project aims to identify hospital patients with severe infections by analyzing their medical records using the R programming language. Specifically, we will use the data.table package in R, which allows for efficient data manipulation and analysis. Throughout the project, we will perform tasks such as loading data, cleaning it, and conducting various analyses to uncover patterns related to sepsis. Our goal is to improve the identification and understanding of sepsis cases, which can help in developing better treatment and prevention strategies.

1.3 Objective and Deliverables

The main goal of this project is to use R and the data.table package to analyze hospital records and identify patients with severe infections, particularly sepsis. The project involves several key tasks: loading and inspecting antibiotic data (antibioticDT.csv), sorting the data for ease of analysis, identifying new antibiotic administrations by calculating intervals between doses and creating indicators for new antibiotic starts, and investigating blood culture data (blood_cultureDT.csv). This data will be merged with the antibiotic data to form a comprehensive dataset. A temporal analysis will determine the relationship between antibiotic administration and blood culture collection, creating indicators for administration within a specified time window around blood culture collection. Sequential antibiotic administration will be examined by identifying the first day of new antibiotic administration within the window and extracting sequences of antibiotic administration days. These sequences will be validated to ensure no significant gaps, identifying qualifying four-day sequences of antibiotic administration. Patients with suspected severe infections will be flagged based on these sequences, generating a dataset of patients meeting the infection criteria. Finally, the identified infection data will be integrated with a comprehensive patient dataset (all_patients.csv) to calculate the percentage of patients with presumed severe infections.

IMPLEMENTATION

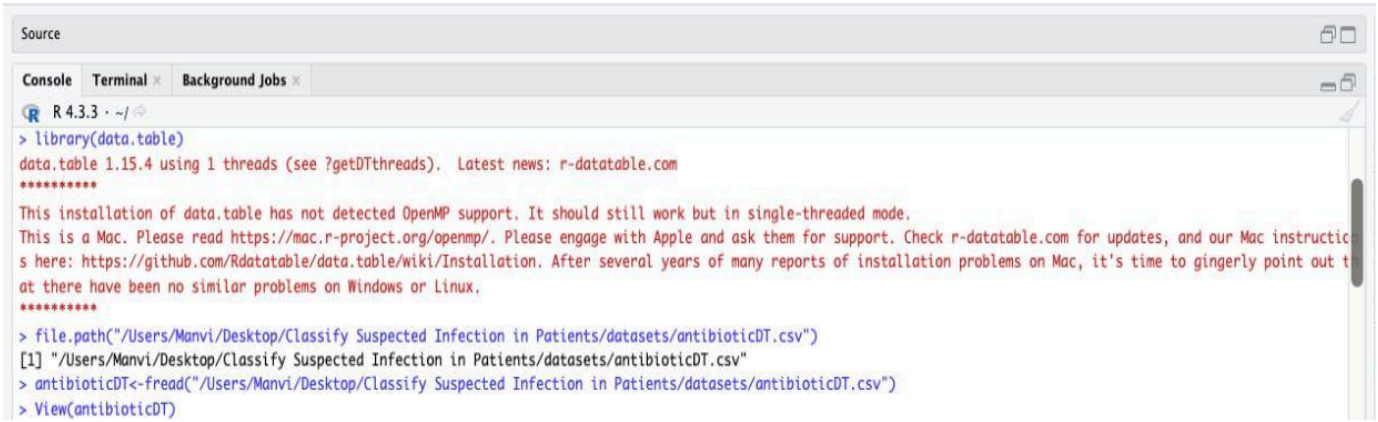
Task 1: Initial Data Inspection

Sepsis is a life-threatening syndrome resulting from a severe infection that causes organ failure. Early treatment significantly improves survival rates, but recognizing sepsis can be challenging. Hospital data could potentially be used to develop machine learning models to flag likely septic patients. Before developing predictive algorithms, we need a reliable method to identify septic patients, with severe infection being a key component of sepsis.

In this project, we will analyze two weeks of hospital electronic health record (EHR) data to identify patients who had a severe infection based on four criteria. Specifically, we will examine whether a doctor ordered a blood culture to test for bacteria and administered a series of intravenous antibiotics.

Steps involved -

- **Load the data.table package**
- **Read in the antibiotic data**
- **Look at the first 30 rows**



```
> head(antibioticDT,30)
    patient_id day_given antibiotic_type
         <int>     <int>          <char>
 1:          1         2   ciprofloxacin
 2:          1         4   ciprofloxacin
 3:          1         6   ciprofloxacin
 4:          1         7     doxycycline
 5:          1         9     doxycycline
 6:          1        15      penicillin
 7:          1        16     doxycycline
 8:          1        18   ciprofloxacin
 9:          8         1     doxycycline
10:          8         2      penicillin
11:          8         3     doxycycline
12:          8         6     doxycycline
13:          8         8      penicillin
14:          8        12      penicillin
15:          9         8     doxycycline
16:          9        12     doxycycline
17:         12         4     doxycycline
18:         12         9     doxycycline
19:         16         1     doxycycline
20:         16         4     amoxicillin
21:         19         3     doxycycline
22:         19         5     amoxicillin
23:         19         6   ciprofloxacin
24:         19        10     doxycycline
25:         19        12      penicillin
26:         23         1     doxycycline
27:         23         1      penicillin
28:         23         3     amoxicillin
29:         23         3   ciprofloxacin
30:         23         3     doxycycline
```

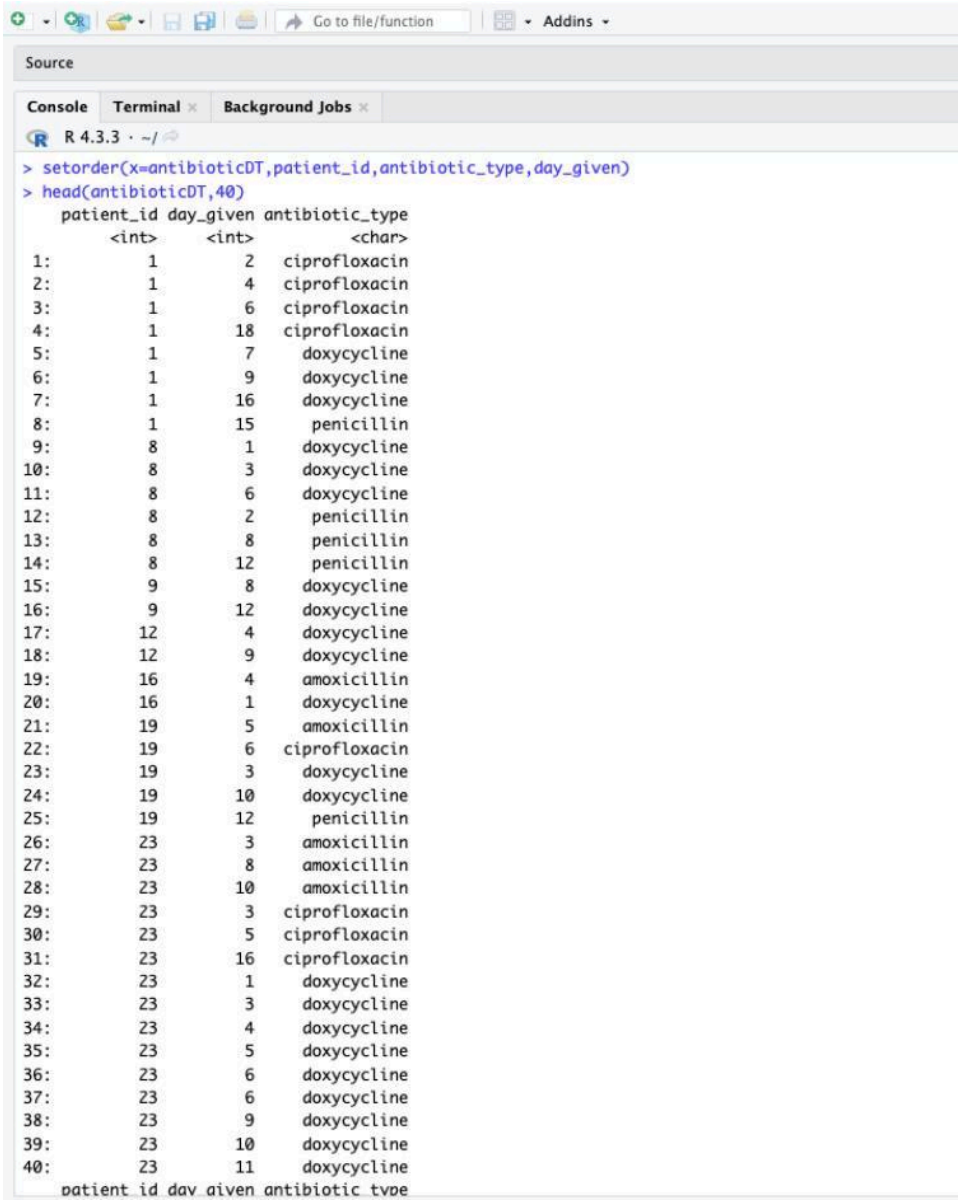
Task 2: Identify Rows Representing "New" Antibiotics

The data represent all antibiotics administered in a hospital over two weeks, with each row indicating a specific antibiotic administration. Variables include patient ID, administration day, antibiotic name, and method of administration. For example, patient "8" received doxycycline orally on their first day.

We will begin by identifying "new antibiotics" by determining if each antibiotic was given to the patient in the prior two days. We'll sort the data by patient ID, antibiotic type, and day for this analysis.

Steps involved -

- **Sort the data**
- **Calculate last_administration_day**
- **Create antibiotic_new variable**



Go to file/function | Addins

Source

Console | Terminal | Background Jobs

R 4.3.3 · ~/

```
> setorder(x=antibioticDT,patient_id,antibiotic_type,day_given)
> head(antibioticDT,40)
    patient_id day_given antibiotic_type
         <int>     <int>          <char>
 1:          1         2   ciprofloxacin
 2:          1         4   ciprofloxacin
 3:          1         6   ciprofloxacin
 4:          1        18   ciprofloxacin
 5:          1         7     doxycycline
 6:          1         9     doxycycline
 7:          1        16     doxycycline
 8:          1        15      penicillin
 9:          8         1     doxycycline
10:          8         3     doxycycline
11:          8         6     doxycycline
12:          8         2      penicillin
13:          8         8      penicillin
14:          8        12      penicillin
15:          9         8     doxycycline
16:          9        12     doxycycline
17:         12         4     doxycycline
18:         12         9     doxycycline
19:         16         4     amoxicillin
20:         16         1     doxycycline
21:         19         5     amoxicillin
22:         19         6   ciprofloxacin
23:         19         3     doxycycline
24:         19        10     doxycycline
25:         19        12      penicillin
26:         23         3     amoxicillin
27:         23         8     amoxicillin
28:         23        10     amoxicillin
29:         23         3   ciprofloxacin
30:         23         5   ciprofloxacin
31:         23        16   ciprofloxacin
32:         23         1     doxycycline
33:         23         3     doxycycline
34:         23         4     doxycycline
35:         23         5     doxycycline
36:         23         6     doxycycline
37:         23         6     doxycycline
38:         23         9     doxycycline
39:         23        10     doxycycline
40:         23        11     doxycycline
    patient_id day_given antibiotic_type
```

```
34:      IV
35:      IV
36:      IV
37:      PO
38:      PO
39:      IV
40:      PO
     route
> antibioticDT[, last_administration_day := shift(day_given), 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]
> head(antibioticDT,10)
    patient_id day_given antibiotic_type
         <int>     <int>          <char>
 1:          1         2   ciprofloxacin
 2:          1         4   ciprofloxacin
 3:          1         6   ciprofloxacin
 4:          1        18   ciprofloxacin
 5:          1         7     doxycycline
 6:          1         9     doxycycline
 7:          1        16     doxycycline
 8:          1        15      penicillin
 9:          8         1     doxycycline
10:          8         3     doxycycline
    route last_administration_day
    <char>                 <int>
 1:     IV                    NA
 2:     IV                     2
 3:     IV                     4
 4:     IV                     6
 5:     IV                    NA
 6:     IV                     7
 7:     IV                     9
 8:     IV                    NA
 9:     PO                    NA
10:     IV                     1
    days_since_last_admin antibiotic_new
                    <int>          <num>
 1:                    NA              1
 2:                     2              0
 3:                     2              0
 4:                    12              1
 5:                    NA              1
```

antibioticDT | blood_cultureDT | combinedDT | necessaryDT | suspected_infection | all_patientsDT | finalDT

Filter

	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
12	8	2	penicillin	IV	NA	NA	1
13	8	8	penicillin	PO	2	6	1
14	8	12	penicillin	IV	8	4	1
15	9	8	doxycycline	IV	NA	NA	1
16	9	12	doxycycline	PO	8	4	1
17	12	4	doxycycline	PO	NA	NA	1
18	12	9	doxycycline	IV	4	5	1
19	16	4	amoxicillin	IV	NA	NA	1
20	16	1	doxycycline	IV	NA	NA	1
21	19	5	amoxicillin	IV	NA	NA	1
22	19	6	ciprofloxacin	IV	NA	NA	1
23	19	3	doxycycline	PO	NA	NA	1
24	19	10	doxycycline	IV	3	7	1
25	19	12	penicillin	IV	NA	NA	1
26	23	3	amoxicillin	IV	NA	NA	1
27	23	8	amoxicillin	IV	3	5	1
28	23	10	amoxicillin	PO	8	2	0
29	23	3	ciprofloxacin	IV	NA	NA	1
30	23	5	ciprofloxacin	PO	3	2	0

Showing 1 to 30 of 6,789 entries, 7 total columns

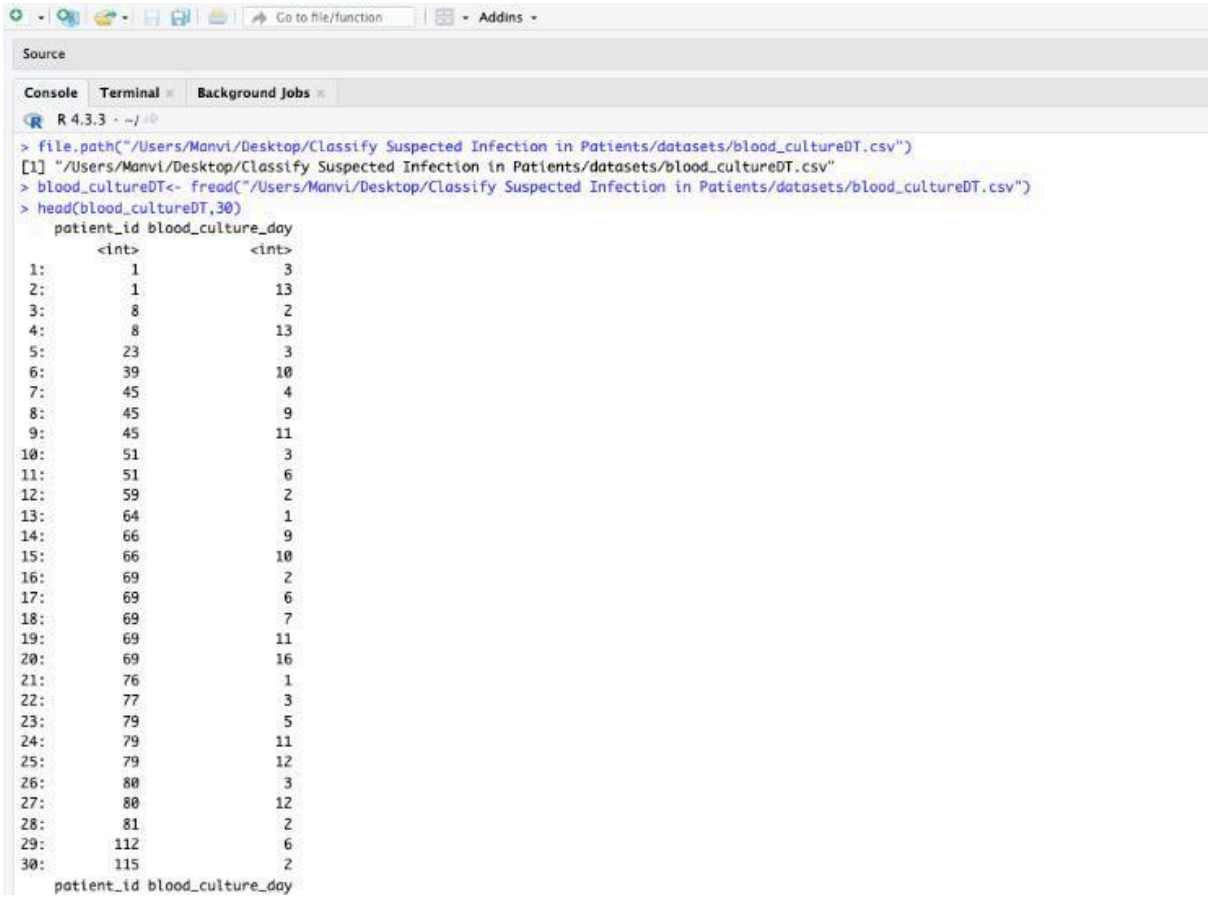
Task 3: Investigate the Blood Culture Data

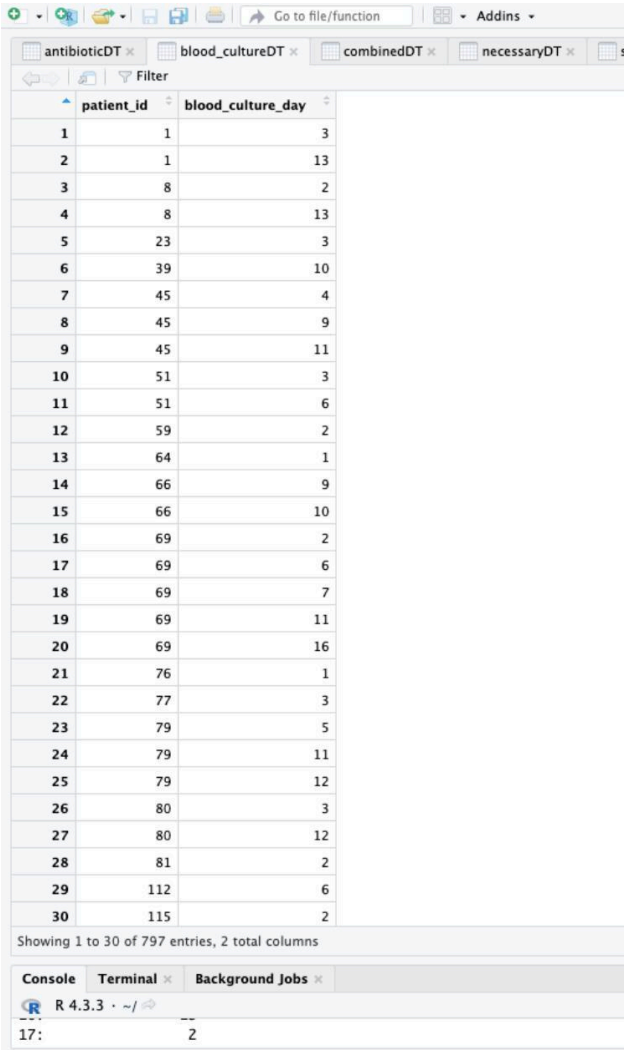
Now let's examine the blood culture data from the same two-week period in this hospital, contained in blood_cultureDT.csv. We will start by reading this data into the workspace and inspecting a few rows.

Each row represents a blood culture, providing the patient's ID and the day the blood culture test was conducted. For example, patient "8" had a blood culture on the second day of their hospitalization and again on the thirteenth day. Note that some patients from the antibiotic dataset are not present in this dataset and vice versa. Additionally, some patients are absent from both datasets as they neither received antibiotics nor had a blood culture.

Steps involved -

- **Read in the blood culture data**
- **Print the first 30 rows**





Task 4: Merge the Antibiotic and Blood Culture Data

To determine which antibiotics were administered close to a blood culture test, we need to combine the drug administration data with the blood culture data, keeping only the patients present in both datasets.

A challenge arises as some patients had blood cultures on multiple days. For each blood culture day, we will check for a sequence of antibiotic days nearby. To achieve this, we will match each blood culture day to each antibiotic day in the merge.

After sorting the merged data, each patient's antibiotic sequence will repeat for each blood culture day. This repetition enables us to examine each blood culture day to see if it is associated with a qualifying sequence of antibiotics.

Steps involved -

- **Merge the datasets**
- **Sort the combined dataset**

```
> combinedDT <- merge(antibioticDT, blood_cultureDT, by = "patient_id")
> View(blood_cultureDT)
> View(combinedDT)
> setorder(x=combinedDT,patient_id,blood_culture_day,day_given,antibiotic_type)
> head(combinedDT,30)
Key: <patient_id>
    patient_id day_given antibiotic_type
         <int>     <int>          <char>
```

Task 5: Determine Antibiotic Administration within 2 Days of Blood Culture

Now that we have combined the antibiotic and blood culture data, we can test each drug administration against each blood culture to determine if it falls "within the window."

Steps involved -

- **Create drug_in_bcx_window**

```
R 4.3.3 · ~/
17:               2
18:               2
19:               2
20:               2
21:               2
22:               2
23:              13
24:              13
25:              13
26:              13
27:              13
28:              13
29:               3
30:               3
    blood_culture_day
> combinedDT[, drug_in_bcx_window := as.numeric(day_given - blood_culture_day<= 2 & day_given - blood_culture_day >= -2)]
```

Task 6: Identify I.V. Antibiotics within +/-2 Days of Blood Culture

Now let's focus on the fourth item in the criteria for suspected infection. To identify a qualifying sequence, the patient must receive antibiotics for four consecutive days, allowing for one-day gaps. The sequence must start with a new antibiotic, defined as one that was not administered in the previous two days. Additionally, the sequence should begin within two days of a blood culture, and there must be at least one intravenous (I.V.) antibiotic administered within the +/-2 day window surrounding the blood culture.

Steps involved -

- **Create any_iv_in_bcx_window**

```
30:                3
    blood_culture_day
> combinedDT[, drug_in_bcx_window := as.numeric(day_given - blood_culture_day<= 2 & day_given - blood_culture_day >= -2)]
> 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]
```

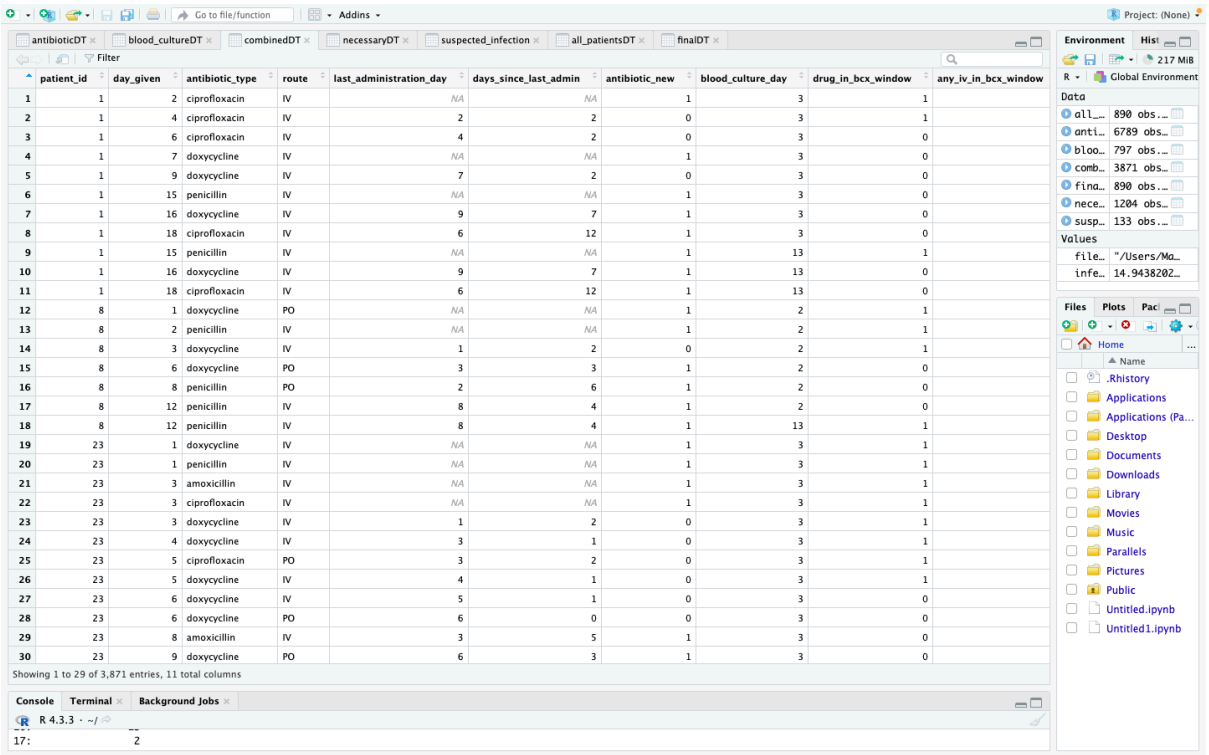
Task 7: Find the First Day of 4-Day Antibiotic Sequences

We aim to implement the first criterion for identifying potential severe infections based on antibiotic usage and blood culture data. To begin, we will find the first day of possible four-day qualifying sequences of antibiotic administrations. This involves creating a variable named day_of_first_new_abx_in_window, which identifies the initial day of new antibiotic administration within the specified window around each blood culture test. This step is crucial as it sets the foundation for subsequent analyses to identify and validate sequences that potentially indicate severe infections in patients.

Steps involved -

- **Create day_of_first_new_abx_in_window**

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



patient_id	day_given	antibiotic_type	route	last_administration_day	days_since_last_admin	antibiotic_new	blood_culture_day	drug_in_bcx_window	any_iv_in_bcx_window
1	2	ciprofloxacin	IV	NA	NA	1	3	1	
1	4	ciprofloxacin	IV	2	2	0	3	1	
1	6	ciprofloxacin	IV	4	2	0	3	0	
1	7	doxycycline	IV	NA	NA	1	3	0	
1	9	doxycycline	IV	7	2	0	3	0	
1	15	penicillin	IV	NA	NA	1	3	0	
1	16	doxycycline	IV	9	7	1	3	0	
1	18	ciprofloxacin	IV	6	12	1	3	0	
1	15	penicillin	IV	NA	NA	1	13	1	
1	16	doxycycline	IV	9	7	1	13	0	
1	18	ciprofloxacin	IV	6	12	1	13	0	
8	1	doxycycline	PO	NA	NA	1	2	1	
8	2	penicillin	IV	NA	NA	1	2	1	
8	3	doxycycline	IV	1	2	0	2	1	
8	6	doxycycline	PO	3	3	1	2	0	
8	8	penicillin	PO	2	6	1	2	0	
8	12	penicillin	IV	8	4	1	2	0	
8	12	penicillin	IV	8	4	1	13	1	
23	1	doxycycline	IV	NA	NA	1	3	1	
23	1	penicillin	IV	NA	NA	1	3	1	
23	3	amoxicillin	IV	NA	NA	1	3	1	
23	3	ciprofloxacin	IV	NA	NA	1	3	1	
23	3	doxycycline	IV	1	2	0	3	1	
23	4	doxycycline	IV	3	1	0	3	1	
23	5	ciprofloxacin	PO	3	2	0	3	1	
23	5	doxycycline	IV	4	1	0	3	1	
23	6	doxycycline	IV	5	1	0	3	0	
23	6	doxycycline	PO	6	0	0	3	0	
23	8	amoxicillin	IV	3	5	1	3	0	
23	9	doxycycline	PO	6	3	1	3	0	

Task 8: Create a Dataset with Required Columns

We will now proceed with implementing the first criterion for identifying potential severe infections based on antibiotic usage and blood culture data. Our objective now is to create a dataset containing essential columns that focus on the days antibiotics were administered. This involves creating a new data.table that retains only the necessary columns relevant to the analysis. These columns will help us to specifically identify and validate four-day sequences of antibiotic administrations, ensuring that gaps of one day are allowed between doses as per the criterion. This step is critical for further analysis to flag patients who may have severe infections, particularly sepsis, based on the temporal patterns of antibiotic use.

Steps involved -

- **Create a new data.table with essential columns**

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

Task 9: Extract the First Four Antibiotic Days

Our goal is to implement the next step in validating the first criterion for identifying potential severe infections based on antibiotic administration and blood culture data. To begin, we will calculate the total number of antibiotic days (num_antibiotic_days) available for each patient and blood culture combination. Patients with fewer than four antibiotic days will be excluded from further analysis. Subsequently, for patients meeting the criterion, we will extract the first four antibiotic days or rows associated with each patient and blood culture combination. This step is crucial for narrowing down potential sequences that align with the criteria of receiving antibiotics for a sequence of four days, allowing for one-day gaps between administrations.

Steps involved -

- **Calculate num_antibiotic_days**

```
> View(necessaryDT)
> necessaryDT <- unique(necessaryDT)
> necessaryDT[, num_antibiotic_days := .N, by = .(patient_id, blood_culture_day)]
> necessaryDT <- necessaryDT[num_antibiotic_days >= 4]
> necessaryDT <- necessaryDT[, .SD[1:4], by = .(patient_id, blood_culture_day)]
```

Task 10: Identify Qualifying Four-Day Sequences

We will create a variable named four_in_seq that will indicate whether each four-day sequence meets the criteria for consecutive antibiotic administrations with gaps of one day or less. This step is essential for validating and selecting sequences that potentially indicate severe infections, particularly sepsis, based on the temporal patterns of antibiotic use in relation to blood culture tests.

Steps involved -

- **Create four_in_seq**

```
> necessaryDT <- unique(necessaryDT)
> necessaryDT[, num_antibiotic_days := .N, by = .(patient_id, blood_culture_day)]
> necessaryDT <- necessaryDT[num_antibiotic_days >= 4]
> necessaryDT <- necessaryDT[, .SD[1:4], by = .(patient_id, blood_culture_day)]
> necessaryDT[, four_in_seq := as.numeric(max(diff(day_given)) <= 2), by = .(patient_id, blood_culture_day)]
```

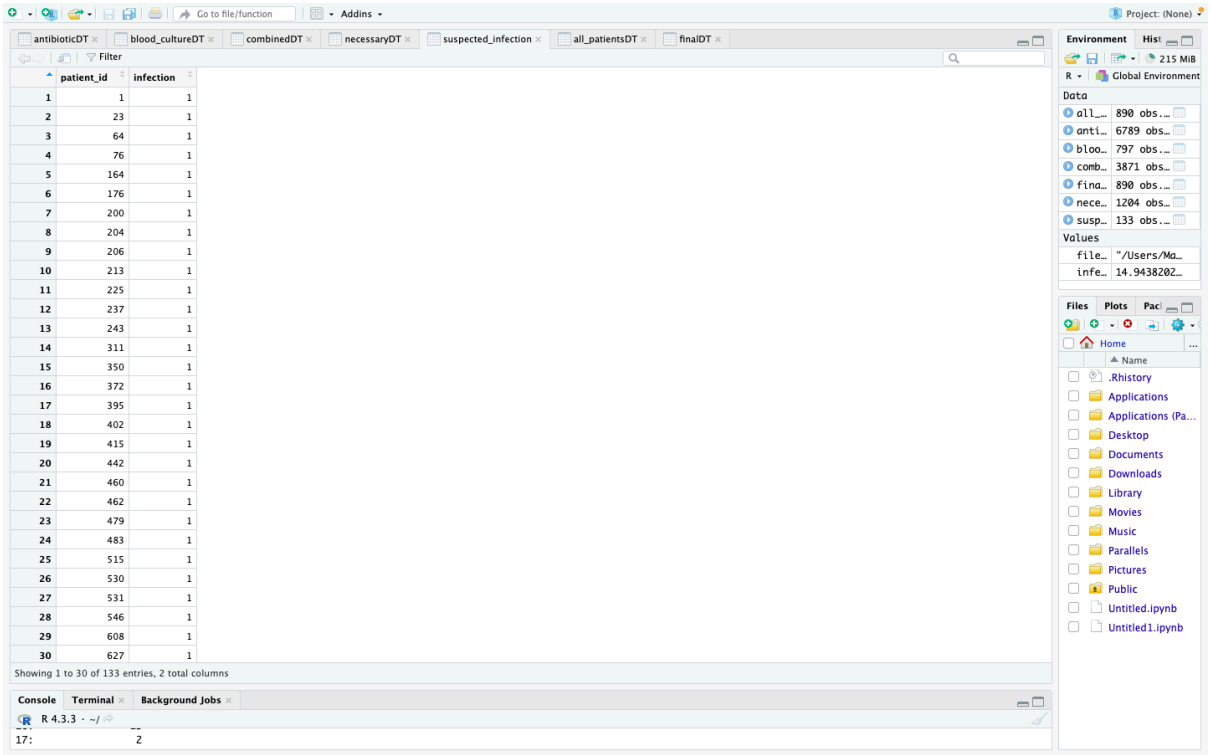
Task 11: Create a Data Frame for Patients with Suspected Infection

After determining which antibiotic sequences qualify for each blood culture test, our next step is to select patients who fulfill these criteria. This involves filtering and compiling a list of patients whose antibiotic administrations align with the defined criteria, indicating a possible severe infection, particularly sepsis. This selection process ensures that we pinpoint individuals who likely require further clinical attention and treatment for severe infections based on the analysis of their hospital records.

Steps involved -

- **Step 1: Select rows with four_in_seq equal to 1**

```
> suspected_infection <- necessaryDT[four_in_seq == 1]
> View(suspected_infection)
> suspected_infection <- suspected_infection[ , .(patient_id)]
> suspected_infection <- unique(suspected_infection)
> suspected_infection[ , infection := 1]
```



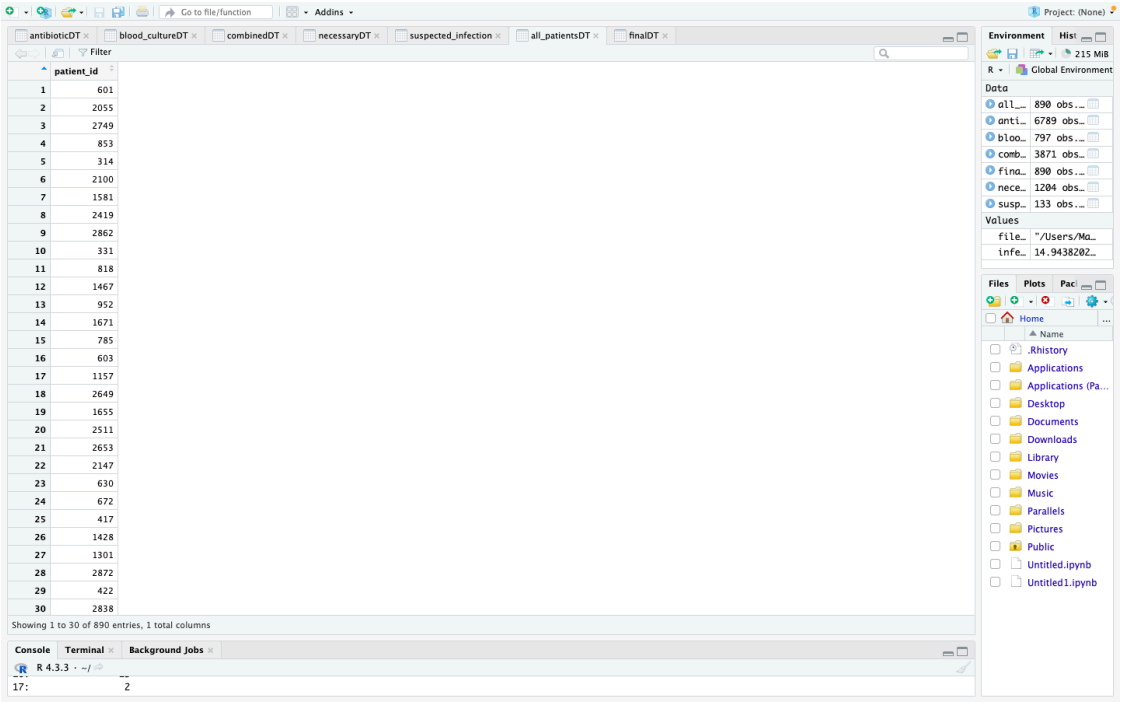
Task 12: Calculate Percentage of Presumed Serious Infections

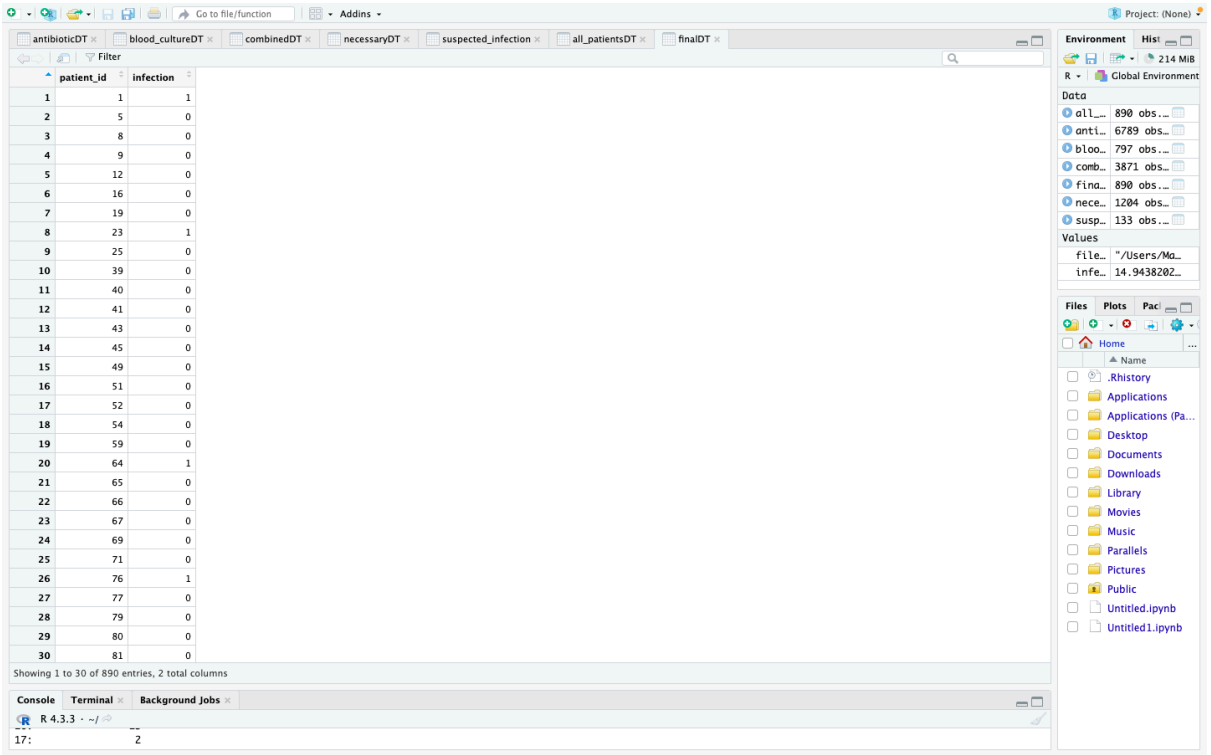
In this phase of the project, we will extend our analysis beyond the antibiotic and blood culture datasets to encompass all hospitalized patients over the two-week period at the hospital. Initially, we will read in additional patient information that includes data on all hospitalized individuals, regardless of whether they received antibiotics or had blood culture tests. Next, we will merge this patient data with our infection flag dataset, which identifies patients suspected of having a serious infection based on our criteria. Finally, we will compute the percentage of presumed serious infections among all hospitalized patients, providing insights into the prevalence of severe infections, particularly sepsis, within the hospital during the specified timeframe. This comprehensive approach ensures a holistic view of infection rates and aids in understanding the impact on patient care and management.

Steps involved -

- **Step 1: Read in the patient data**
- **Step 2: Merge with infection flag data**
- **Step 3: Calculate the percentage of presumed serious infections**

```
> file_path_all <-"/Users/Manvi/Desktop/Classify Suspected Infection in Patients/datasets/all_patients.csv"
> all_patientsDT <- fread("/Users/Manvi/Desktop/Classify Suspected Infection in Patients/datasets/all_patients.csv")
> View(all_patientsDT)
> finalDT <- merge(all_patientsDT, suspected_infection, by = "patient_id", all.x = TRUE)
>
> finalDT[is.na(infection), infection := 0]
> View(finalDT)
> infection_rate <- finalDT[, mean(infection)] * 100
> print(infection_rate)
[1] 14.94382
> save.image("~/Desktop/workspace.RData")
>
```

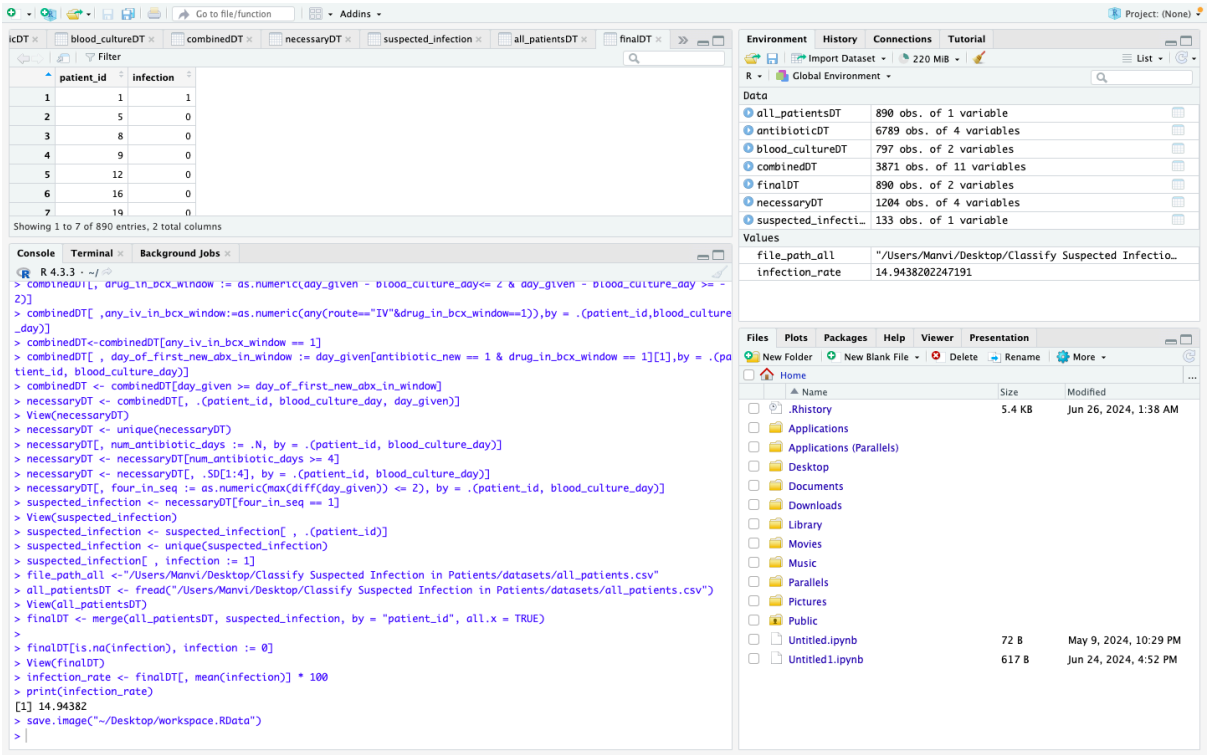




	patient_id	infection
1	1	1
2	5	0
3	8	0
4	9	0
5	12	0
6	16	0
7	19	0
8	23	1
9	25	0
10	39	0
11	40	0
12	41	0
13	43	0
14	45	0
15	49	0
16	51	0
17	52	0
18	54	0
19	59	0
20	64	1
21	65	0
22	66	0
23	67	0
24	69	0
25	71	0
26	76	1
27	77	0
28	79	0
29	80	0
30	81	0

Showing 1 to 30 of 890 entries, 2 total columns

```
17:                 2
```



	patient_id	infection
1	1	1
2	5	0
3	8	0
4	9	0
5	12	0
6	16	0
7	19	0

Showing 1 to 7 of 890 entries, 2 total columns

```
> combinedDT[, drug_in_bcx_window := as.numeric(day_given - blood_culture_day<= 2 & day_given - blood_culture_day >= -2)]
> 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[ , 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]
> necessaryDT <- combinedDT[, .(patient_id, blood_culture_day, day_given)]
> View(necessaryDT)
> necessaryDT <- unique(necessaryDT)
> necessaryDT[, num_antibiotic_days := .N, by = .(patient_id, blood_culture_day)]
> necessaryDT <- necessaryDT[num_antibiotic_days >= 4]
> necessaryDT <- necessaryDT[, .SD[1:4], by = .(patient_id, blood_culture_day)]
> necessaryDT[, four_in_seq := as.numeric(max(diff(day_given)) <= 2), by = .(patient_id, blood_culture_day)]
> suspected_infection <- necessaryDT[four_in_seq == 1]
> View(suspected_infection)
> suspected_infection <- suspected_infection[ , .(patient_id)]
> suspected_infection <- unique(suspected_infection)
> suspected_infection[ , infection := 1]
> file_path_all <-"/Users/Manvi/Desktop/Classify Suspected Infection in Patients/datasets/all_patients.csv"
> all_patientsDT <- fread("/Users/Manvi/Desktop/Classify Suspected Infection in Patients/datasets/all_patients.csv")
> View(all_patientsDT)
> finalDT <- merge(all_patientsDT, suspected_infection, by = "patient_id", all.x = TRUE)
>
> finalDT[is.na(infection), infection := 0]
> View(finalDT)
> infection_rate <- finalDT[, mean(infection)] * 100
> print(infection_rate)
[1] 14.94382
> save.image("~/Desktop/workspace.RData")
>
```

Conclusion and Future Scope

In this project, we aimed to identify hospital patients with severe infections using medical record data. By meticulously following a structured series of data processing and analysis steps, we were able to successfully load, clean, and merge antibiotic and blood culture datasets. We created variables to indicate the relationship between antibiotic administration and blood culture events, calculated days since the last antibiotic administration, and flagged new antibiotic uses. We then filtered and analyzed the data to find sequences of antibiotic administrations that met our criteria for presumed severe infections. Our final analysis revealed that approximately 14.94% of patients in the dataset met these criteria, providing valuable insights into the prevalence of severe infections in the hospital setting.

The future scope of this project includes several key areas for enhancement and further research. Firstly, developing predictive models using machine learning techniques could help identify patients at risk of severe infections earlier, improving patient outcomes. Incorporating real-time data analysis would allow for continuous monitoring and prompt detection of severe infections as new data becomes available. Integration with hospital information systems could provide automated alerts and decision support for healthcare providers. Expanding the dataset to include data from multiple hospitals would validate and generalize the findings, while interactive dashboards and detailed reports could offer visual insights for healthcare administrators. Longitudinal studies tracking patient outcomes over time would evaluate the long-term effectiveness of interventions and treatment strategies. By extending the scope of this project, we can further enhance our understanding of severe infections and contribute to better patient care and management strategies in hospital settings.

THANKYOU