FINAL REPORT OF INTERNSHIP PROGRAM 2021

ON

"CLASSIFICATION OF SUSPECTED INFECTION IN PATIENTS"

SUBMITTED BY

THISHONIA PREETHI.S

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES

MEDTOUREASY

27TH MARCH 2021

ACKNOWLDEGMENTS

The internship opportunity that I had with MedTourEasy was a great chance for learning and understanding the intricacies of the subject of Business 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 internship 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 the opportunity to carry out my internship at their esteemed organization. Also, I express my thanks to the team for making me understand the details of the Business 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 sparing his valuable time despite 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.

TABLE OF CONTENTS

ABSTRACT

Sepsis is a life-threatening organ dysfunction caused by a dysregulated host response to infection. If not recognized early and managed promptly, it can lead to septic shock, multiple organ failure and death. It is most frequently a serious complication of infection, particularly in low- and middle-income countries where it represents a major cause of maternal and neonatal morbidity and mortality. In the community setting, sepsis often presents as the clinical deterioration of common and preventable infections. Sepsis also frequently results from infections acquired in health care settings, which are one of the most frequent adverse events during care delivery and affect hundreds of millions of patients worldwide every year. Healthcare-associated infections are often resistant to antibiotics and can rapidly lead to deteriorating clinical conditions. Antimicrobial resistance is a major factor determining clinical unresponsiveness to treatment and rapid evolution to sepsis and septic shock. Sepsis patients with resistant pathogens have been found to have a higher risk of hospital mortality. Implementing preventive measures against infections, such as good hygiene practices, ensuring access to vaccination programmes, improved sanitation and water quality and availability, and other infection prevention and control best practices both in the community and health care settings, are key steps in reducing the occurrence of sepsis. Early diagnosis and timely and appropriate clinical management of sepsis, such as optimal antimicrobial use and fluid resuscitation, are crucial to increase the likelihood of survival. Even though the onset of sepsis can be acute and poses a short-term mortality burden, it can also be the cause of significant long-term morbidity requiring treatment and support. Thus, this project aims at calculating the percentage of patients prone to sepsis.

1. INTRODUCTION

1.1 ABOUT THE COMPANY

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 ABOUT THE PROJECT

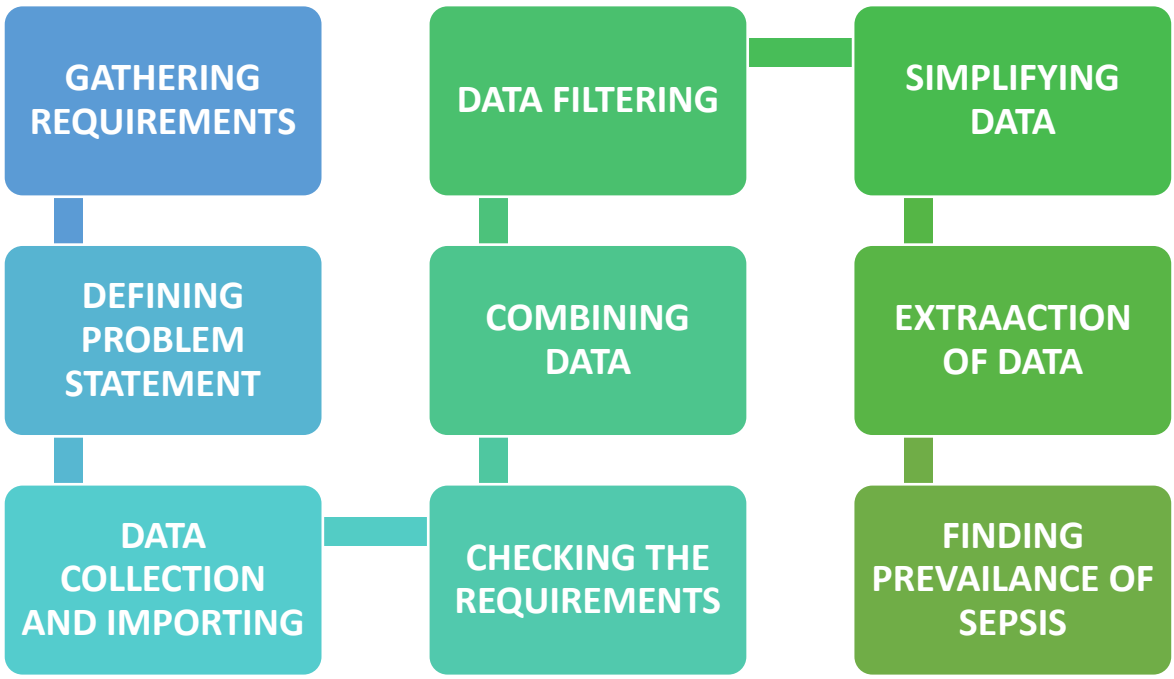
Sepsis is a deadly illness that accounts for a large portion of in-hospital deaths. It occurs when a person's organs shut down in response to a severe infection. This public health problem is a major target for research, and hospital records can help us investigate the problem. In 2017, the largest contributors to sepsis cases and sepsis-related mortality across all ages were diarrhoeal diseases (9.2 to 15 million annual cases) and lower respiratory infections (1.8-2.8 million annually). However, non-communicable diseases are on the rise; one-third of sepsis cases and nearly half of all sepsis-related deaths in 2017 were due to an underlying injury or chronic disease. Maternal disorders were the most common non-communicable disease complicated by sepsis. In this R project, you will identify hospital patients with severe infection using medical record data.

1.3 OBJECTIVE

This project focuses on finding out the rate of suspected infection among patients in hospital. It is a deadly disease and timely treatment will help saves lives.

2. METHODOLOGY

2.1 FLOW OF THE PROJECT



2.2 LANGUAGE AND PLATFORM USED

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. As noted above, 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.

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 DATA.TABLE

Data manipulation operations such as subset, group, update, join etc., are all inherently related. Keeping these related operations together allows for:

- Concise and consistent syntax irrespective of the set of operations you would like to perform to achieve your end goal.

- Performing analysis fluidly without the cognitive burden of having to map each operation to a particular function from a potentially huge set of functions available before performing the analysis.
- Automatically optimising operations internally, and very effectively, by knowing precisely the data required for each operation, leading to very fast and memory efficient code.

Briefly, if you are interested in reducing programming and compute time tremendously, then this package is for you. The philosophy that data.table adheres to makes this possible. Our goal is to illustrate it through this series of vignettes.

3. IMPLEMENTATION

3.1 GATHERING REQUIREMENTS

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.

3.2 DEFINING PROBLEM STATEMENT

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. In this R project, the hospital patients with severe infections are identified using medical record data.

3.3 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.

3.4 DATA FILTERING

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.

3.5 COMBINING DATA

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

3.6 CHECKING THE REQUIREMENTS

Investigate the blood culture data.

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

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.

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.

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.

3.7 SIMPLIFYING DATA

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.

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.

3.8 EXTRACTION OF DATA

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.

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.

- diff() takes a vector of numbers and calculates the difference between each pair of adjacent numbers. If there is a gap of one day, the difference will be two. max() of the diff() would be useful here too.

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.

3.9 FINDING PREVAILANCE OF SEPSIS

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.

4. SOURCE CODE AND OUTPUT

```
#TASK 1

# Load packages

library(data.table)

# Read in the data

antibioticDT <- fread("D:\\antibioticDT.csv")

# Look at the first 30 rows

antibioticDT[1:30]

#TASK 2

# Sort the data by id, antibiotic type, day

setorder(antibioticDT, patient_id, antibiotic_type, day_given)

#antibioticDT[1:40]

# Use shift to calculate the last day a particular drug was administered

antibioticDT[ , last_administration_day := shift(day_given, 1),

            by = .(patient_id, antibiotic_type)]

# Calculate the number of days since the drug was last administered

antibioticDT[ , days_since_last_admin := day_given - last_administration_day]
```

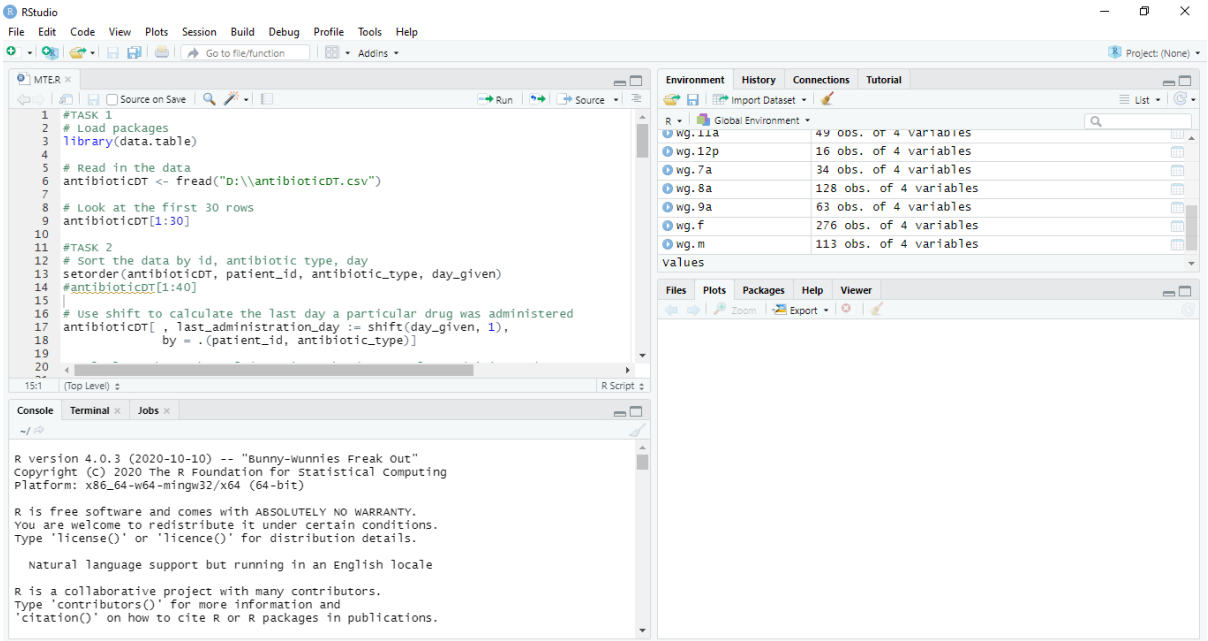
```
# Create antibiotic_new with an initial value of one, then reset it to zero as needed

antibioticDT[ , antibiotic_new := 1]

antibioticDT[days_since_last_admin <= 2, antibiotic_new := 0]

#TASK 3

# Read in blood_cultureDT.csv

blood_cultureDT <- fread("D:\\blood_cultureDT.csv")

# Print the first 30 rows

blood_cultureDT[1:30]

#TASK 4

# Merge antibioticDT with blood_cultureDT

combinedDT <- merge(

  blood_cultureDT,

  antibioticDT,

  all = FALSE,

  by = 'patient_id')

# Sort by patient_id, blood_culture_day, day_given, and antibiotic_type

setorder(combinedDT, patient_id, blood_culture_day, day_given, antibiotic_type)
```

```
# Print and examine the first 30 rows

combinedDT[1:40]

#TASK 5

# Make a new variable called drug_in_bcx_window

combinedDT[ ,

        drug_in_bcx_window :=

         as.numeric(

           day_given - blood_culture_day <= 2

           &

             day_given - blood_culture_day >= -2)]

#TASK 6

# Create a variable indicating if there was at least one I.V. drug given in the window

combinedDT[ ,

        any_iv_in_bcx_window := as.numeric(any(route == 'IV' & drug_in_bcx_window ==

1)),

        by = .(patient_id, blood_culture_day)]

# Exclude rows in which the blood_culture_day does not have any I.V. drugs in window
```

```
combinedDT <- combinedDT[any_iv_in_bcx_window == 1]

#TASK 7

# Create a new variable called day_of_first_new_abx_in_window

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)]

# Remove rows where the day is before this first qualifying day

combinedDT <- combinedDT[day_given >= day_of_first_new_abx_in_window]

#TASK 8

# Create a new data.table containing only patient_id, blood_culture_day, and day_given

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

# Remove duplicate rows

simplified_data <- unique(simplified_data)

#TASK 9

#Count the antibiotic days within each patient/blood culture day combination

simplified_data[ , num_antibiotic_days := .N, by = .(patient_id, blood_culture_day)]
```

```
# Remove blood culture days with less than four rows

simplified_data <- simplified_data[num_antibiotic_days >= 4]

# Select the first four days for each blood culture

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

#TASK 10

# Make the indicator for consecutive sequence

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

#TASK 11

# Select the rows which have four_in_seq equal to 1

suspected_infection <- first_four_days[four_in_seq == 1]

# Retain only the patient_id column

suspected_infection <- suspected_infection[ , .(patient_id)]

# Remove duplicates

suspected_infection <- unique(suspected_infection)

# Make an infection indicator

suspected_infection[ , infection := 1]
```

```
suspected_infection

#TASK 12

# Read in "all_patients.csv"

all_patientsDT <- fread("D:\\all_patients.csv")

# Merge this with the infection flag data

all_patientsDT <- merge(

  all_patientsDT,

  suspected_infection,

  by = "patient_id",

  all = TRUE)

# Set any missing values of the infection flag to 0

all_patientsDT[is.na(infection) , infection := 0]

# Calculate the percentage of patients who met the criteria for presumed infection

ans <- all_patientsDT[ , 100*mean(infection == 1)]

ans
```




```
> library(data.table)
data.table 1.13.6 using 2 threads (see ?getDTthreads).  Latest news: r-datatable.com
> antibioticDT <- fread("D:\\antibioticDT.csv")
> antibioticDT[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:          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
17:         12         4     doxycycline    PO
18:         12         9     doxycycline    IV
19:         16         1     doxycycline    IV
20:         16         4     amoxicillin    IV
21:         19         3     doxycycline    PO
22:         19         5     amoxicillin    IV
23:         19         6   ciprofloxacin    IV
24:         19        10     doxycycline    IV
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(antibioticDT, patient_id, antibiotic_type, day_given)
> #antibioticDT[1:40]
> antibioticDT[ , last_administration_day := shift(day_given, 1),
+               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]
> blood_cultureDT <- fread("D:\\blood_cultureDT.csv")
> blood_cultureDT[1:30]
    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
18:         69                 7
19:         69                11
20:         69                16
21:         76                 1
22:         77                 3
23:         79                 5
24:         79                11
25:         79                12
26:         80                 3
27:         80                12
```

```
10:         51                  3
11:         51                  6
12:         59                  2
13:         64                  1
14:         66                  9
15:         66                 10
16:         69                  2
17:         69                  6
18:         69                  7
19:         69                 11
20:         69                 16
21:         76                  1
22:         77                  3
23:         79                  5
24:         79                 11
25:         79                 12
26:         80                  3
27:         80                 12
28:         81                  2
29:        112                  6
30:        115                  2
    patient_id blood_culture_day
> combinedDT <- merge(
+   blood_cultureDT,
+   antibioticDT,
+   all = FALSE,
+   by = 'patient_id')
> setorder(combinedDT, patient_id, blood_culture_day, day_given, antibiotic_type)
```

```
> combinedDT[1:40]
    patient_id blood_culture_day day_given antibiotic_type route
 1:          1                 3         2   ciprofloxacin    IV
 2:          1                 3         4   ciprofloxacin    IV
 3:          1                 3         6   ciprofloxacin    IV
 4:          1                 3         7     doxycycline    IV
 5:          1                 3         9     doxycycline    IV
 6:          1                 3        15      penicillin    IV
 7:          1                 3        16     doxycycline    IV
 8:          1                 3        18   ciprofloxacin    IV
 9:          1                13         2   ciprofloxacin    IV
10:          1                13         4   ciprofloxacin    IV
11:          1                13         6   ciprofloxacin    IV
12:          1                13         7     doxycycline    IV
13:          1                13         9     doxycycline    IV
14:          1                13        15      penicillin    IV
15:          1                13        16     doxycycline    IV
16:          1                13        18   ciprofloxacin    IV
17:          8                 2         1     doxycycline    PO
18:          8                 2         2      penicillin    IV
19:          8                 2         3     doxycycline    IV
20:          8                 2         6     doxycycline    PO
21:          8                 2         8      penicillin    PO
22:          8                 2        12      penicillin    IV
23:          8                13         1     doxycycline    PO
24:          8                13         2      penicillin    IV
25:          8                13         3     doxycycline    IV
26:          8                13         6     doxycycline    PO
27:          8                13         8      penicillin    PO
28:          8                13        12      penicillin    IV
29:         23                 3         1     doxycycline    IV
30:         23                 3         1      penicillin    IV
31:         23                 3         3     amoxicillin    IV
32:         23                 3         3   ciprofloxacin    IV
33:         23                 3         3     doxycycline    IV
```

	patient_id	blood_culture_day	day_given	antibiotic_type	route	last_administration_day	days_since_last_admin	antibiotic_new
36:	23	3	5	doxycycline	IV			
37:	23	3	6	doxycycline	IV			
38:	23	3	6	doxycycline	PO			
39:	23	3	8	amoxicillin	IV			
40:	23	3	9	doxycycline	PO			
1:		NA				NA	NA	1
2:		2				2	2	0
3:		4				2	2	0
4:		NA				NA	NA	1
5:		7				2	2	0
6:		NA				NA	NA	1
7:		9				7	7	1
8:		6				12	12	1
9:		NA				NA	NA	1
10:		2				2	2	0
11:		4				2	2	0
12:		NA				NA	NA	1
13:		7				2	2	0
14:		NA				NA	NA	1
15:		9				7	7	1
16:		6				12	12	1
17:		NA				NA	NA	1
18:		NA				NA	NA	1
19:		1				2	2	0
20:		3				3	3	1
21:		2				6	6	1
22:		8				4	4	1
23:		NA				NA	NA	1
24:		NA				NA	NA	1
25:		1				2	2	0
26:		3				3	3	1
27:		2				6	6	1
28:		8				4	4	1
29:		NA				NA	NA	1

```
> 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]
> simplified_data <- combinedDT[ , .(patient_id, blood_culture_day, day_given)]
> simplified_data <- unique(simplified_data)
> 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_da
y)]
> first_four_days[ , four_in_seq := as.numeric(max(diff(day_given)) < 3), by = .(patien
t_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
     patient_id infection
  1:          1         1
  2:         23         1
  3:         64         1
  4:         76         1
  5:        164         1
 ---
129:       2957         1
130:       2958         1
131:       2961         1
132:       2975         1
133:       2998         1
> all_patientsDT <- fread("D:\\all_patients.csv")
> all_patientsDT <- merge(
+   all_patientsDT,
+   suspected_infection,
+   by = "patient_id",
+   all = TRUE
+ )
> all_patientsDT[is.na(infection) , infection := 0]
> ans <- all_patientsDT[ , 100*mean(infection == 1)]
> ans
[1] 14.94382
```

5. CONCLUSION

Successful management of the critically ill patient with sepsis crucially depends on the appropriate treatment of the underlying infectious focus. A detailed and structured history and a systematic patient examination identify the infectious focus in the majority of patient and help to guide further diagnostic work-up and early treatment decisions. Only if the physical examination has been performed systematically, the clinician can be sure that no potential focus has been missed, a primary bloodstream infection is present, or the patient suffers from a condition other than an infection.

6. FUTURE SCOPE

It's important that patients, their families and caregivers, and healthcare professionals think about sepsis as a possibility. Get Ahead of Sepsis reminds us all of the importance of early recognition, timely treatment, and preventing infections.

- Sepsis is a medical emergency. Time matters. If you or your loved one suspects sepsis or has an infection that's not getting better or is getting worse, ask your healthcare professional, "Could this infection be leading to sepsis?"
- Anyone can get an infection, and almost any infection can lead to sepsis. Certain people are at higher risk, including adults 65 or older; people with chronic medical conditions, such as diabetes, lung disease, cancer, and kidney disease; people with weakened immune systems; sepsis survivors; and children younger than one.
- A patient with sepsis might have one or more of the following signs or symptoms:
    - High heart rate or low blood pressure
    - Fever, shivering, or feeling very cold
    - Confusion or disorientation

- Shortness of breath
- Extreme pain or discomfort
- Clammy or sweaty skin

Getting ahead of Sepsis encourages healthcare professionals to know sepsis signs and symptoms, identify and treat patients early, act fast if they suspect sepsis, know their facility's existing guidance for diagnosing and managing sepsis, prevent infections, and educate patients and their families. If healthcare professional's suspect sepsis, they should:

- Know their facility's existing guidance for diagnosing and managing sepsis.
- Immediately alert the clinician in charge if it is not them.
- Start antibiotics as soon as possible in addition to other therapies appropriate for the patient. Once the specific cause of sepsis is known, such as a positive test for COVID-19, therapy can be targeted, and empiric broad-spectrum antibiotics might not be needed.
- Check patient progress frequently. Always remember to prescribe the right antibiotic, at the right dose, for the right duration, and at the right time. Reassess antibiotic therapy to stop or tailor treatment based on the patient's or resident's clinical condition and diagnostic test results as appropriate.

Sepsis is a medical emergency. Healthcare professionals should protect their patients by acting fast. Their fast recognition and treatment can increase their patients' chances of survival. Infections can put your patients at a risk for a life-threatening condition called sepsis. Hence "Prevention is better than cure"

7. REFRENCES

1. https://drive.google.com/file/d/162KBAQEUU6b84LIgj8MU3a7E6Yrel_bg/view

Dataset

2. https://learn.datacamp.com/

R tutorial

3.https://www.who.int/news-room/fact-sheets/detail/sepsis#:~:text=Sepsis%20is%20a%20life%2Dthreatening,multiple%20organ%20failure%20and%20death.

Information on Sepsis

4. https://www.cdc.gov/patientsafety/features/get-ahead-of-sepsis.html

5. https://cran.r-project.org/web/packages/data.table/vignettes/datatable-intro.html

6. https://en.wikipedia.org/wiki/RStudio

7.https://oxfordmedicine.com/view/10.1093/med/9780199204854.001.1/med-9780199204854-miscMatter-1001