JOURNAL

Name– Shubham Shah

Class – Sybscit

Roll No. – 051

Div – A

Subject – Operation Research (OR)

Teacher Incharge – Manisha Ma'am

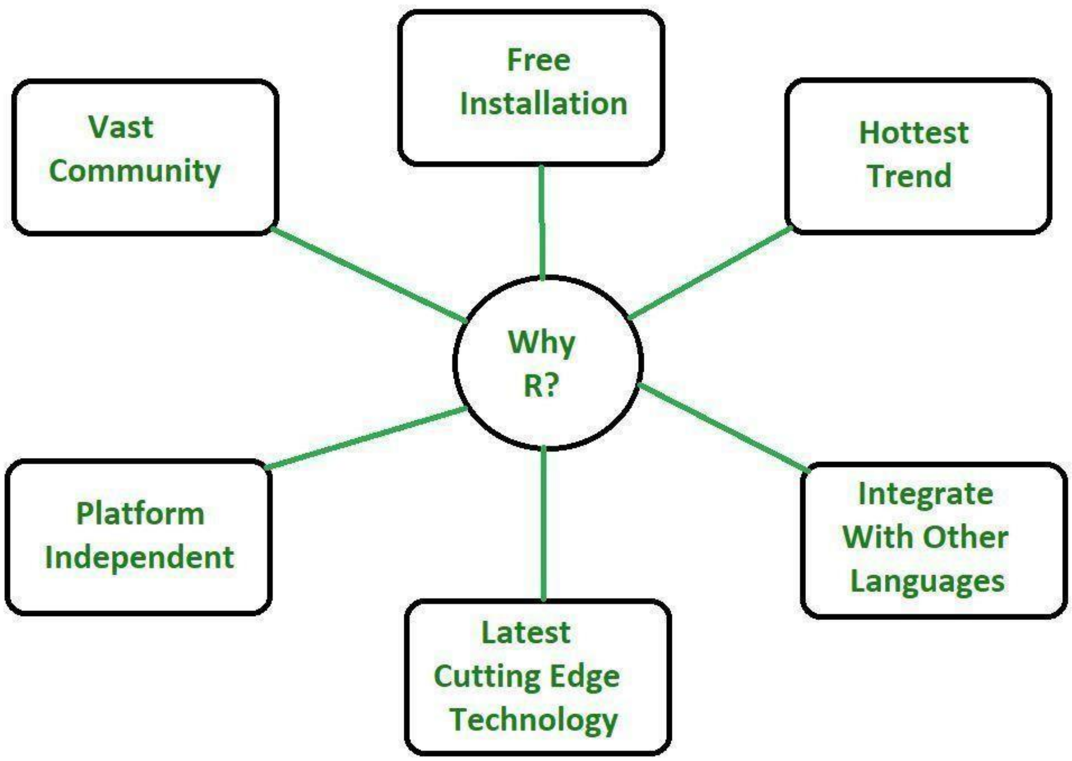
PRACTICAL–1

Introduction to R

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cuttingedge tool.

It was designed by **Ross Ihaka and Robert Gentleman** at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000.

Why R Programming Language?



- R programming is used as a leading tool for machine learning, statistics, and data analysis. Objects, functions, and packages can easily be created by R.
- It’s a platform-independent language. This means it can be applied to all operating system.
- It’s an open-source free language. That means anyone can install it in any organization without purchasing a license.

- R programming language is not only a statistic package but also allows us to integrate with other languages (C, C++). Thus, you can easily interact with many data sources and statistical packages.
- The R programming language has a vast community of users and it’s growing day by day.
- R is currently one of the most requested programming languages in the Data Science job market that makes it the hottest trend nowadays. **Features of R Programming Language**

Statistical Features of R:

- **Basic Statistics:** The most common basic statistics terms are the mean, mode, and median. These are all known as “Measures of Central Tendency.” So using the R language we can measure central tendency very easily.
- **Static graphics:** R is rich with facilities for creating and developing interesting static graphics. R contains functionality for many plot types including graphic maps, mosaic plots, biplots, and the list goes on.
- **Probability distributions:** Probability distributions play a vital role in statistics and by using R we can easily handle various types of probability distribution such as Binomial Distribution, Normal Distribution, Chi-squared Distribution and many more.
- **Data analysis:** It provides a large, coherent and integrated collection of tools for data analysis.

What is R Used For?

Although R is a popular language used by many programmers, it is especially effective when used for

Data analysis

Statistical inference

Machine learning algorithms

R offers a wide variety of statistics-related libraries and provides a favorable environment for statistical computing and design. In addition, the R programming language gets used by many quantitative analysts as a programming tool since it's useful for data importing and cleaning.

Why Do We Use R Programming Language?

There are various reasons why one should learn R programming language. Here are some:

- R is **free** and **open-source**. That means you don't need to pay for licenses.

- It is **platform-independent**. This makes it cost-effective and versatile. You only need to make one program that can run on various platforms.

- Currently, R has over **10,000 packages** available. It is stored in CRAN repositories, and it is continuously updated.

- It's popular for statistics and preferred by programmers for statistical tools.
- It's suitable for machine learning, offering features and packages for tasks like regression and classification.

- R helps with data wrangling. R contains various packages which help transform messy data into structured formats.

- Last but importantly, It has continuously evolved with a supportive community.

However, there are some drawbacks to consider:

- R has a steep learning curve and is better suited for experienced programmers.

- It’s not super secure. It lacks basic security measures and can't be used for web-safe applications.

- Compared to other languages like Python or MATLAB, R may be slower.

- Memory management is also not the strong point of R language. It requires more memory as data is stored in physical memory. Although, cloud-based memory service may eliminate this drawback.

- R does not have consistent package quality. Documentation and package quality can vary due to the community-driven nature of R. Docs, and packages may be patchy or inconsistent as it doesn’t have official dedicated support.

Despite these drawbacks, R remains a powerful and widely used language for data analysis and statistical computing.

Installing R

- Can be downloaded for free from

  http://www.r-project.org/

- Download the version compatible with your OS

• Simple/Standard installation process

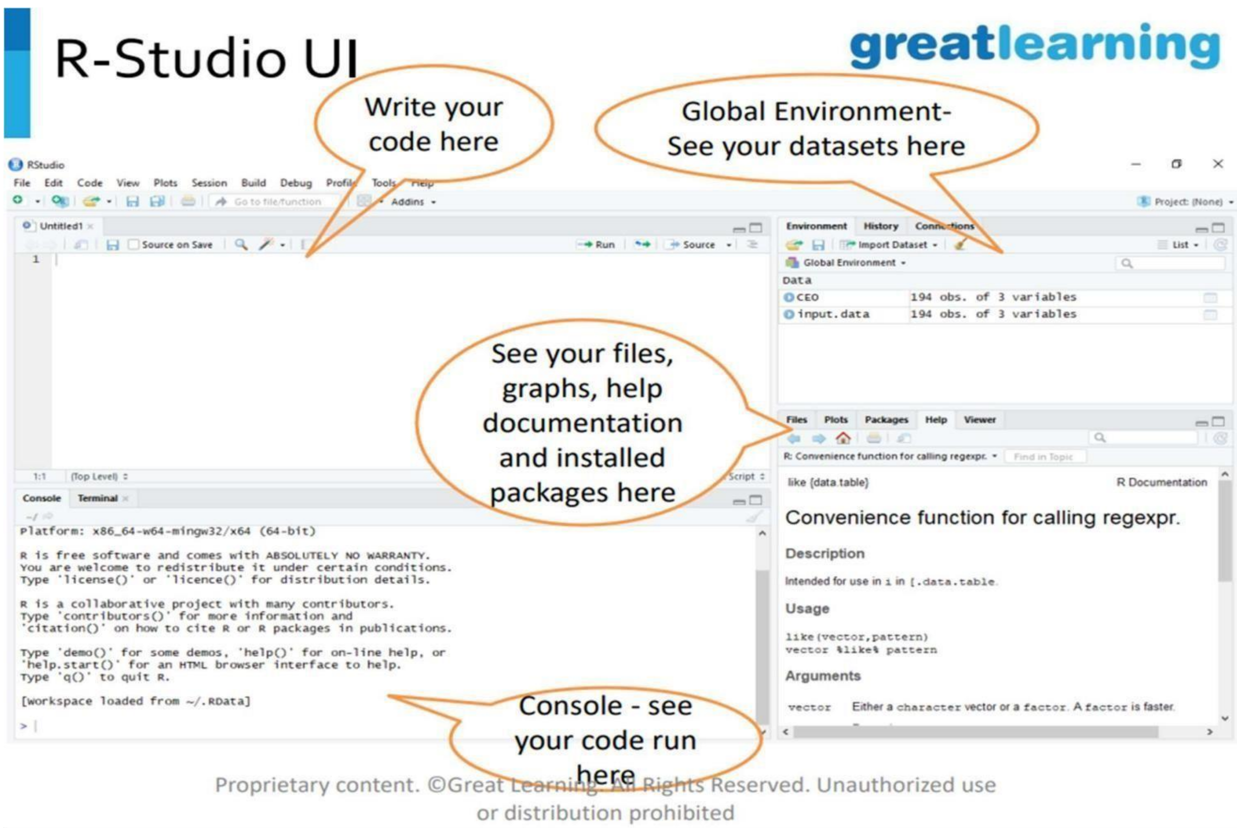
Installing R -Studio

• Can be downloaded for free from:

https://www.rstudio.com/products/rstudio/download/

• Download the free version compatible with your OS

• R needs to be installed before installing R- Studio



R Script File

Usually, you will do your programming by writing your programs in script files and then

you execute those scripts at your command prompt with the help of R interpreter called

Rscript. So let's start with writing following code in a text file called test. R as under –

```
# My first program in R Programming
myString <- "Hello, World!" print (
myString)
```

```
[1] "Hello, World!"
```

R Commands

- Assignments E.g.: x = 1, or x <- 1
- Functions E.g.: print(“Hello World”)
- Computations

E.g.: 17 + 3 ; x + 5

- Mix E.g.: y = sqrt(16); y = 15 + 5
- Assignment queries will update objects in your R environment
- Queries without assignment, as well as ‘call’ of R objects will either generate an output in the console, or in the plot tab

R Basics: Data Types

Variable Assignment in R

- A basic construct in programming is "variable"
- A variable allows you to store a piece of data (‘datum’, e.g. 6, ‘Hello’, etc..) or several pieces of data of a common type, and assign them a unique name
- You can then later ‘call’ this variable's name to easily access the value(s) that is/are stored within this variable.

Careful, R is case sensitive: The variables ‘x’ and ‘X’ can coexist in R environment and have different values.

Basic data types in R

- R works with numerous data types. The most common types are:
- Decimals values like 3.5, called 'numeric'
- Natural numbers like 3 are called 'integers'. Integers are also numeric Boolean variables (TRUE or FALSE) are classified as ‘logical’
- Text (or string) values are classified as 'character’

```
# Create two vectors. v1 <-
c(3,8,4,5,0,11) v2 <-
c(4,11,0,8,1,2)
```

```
# Vector addition. add.result
<- v1+v2 print(add.result)

# Vector subtraction.
sub.result <- v1-v2
print(sub.result)

# Vector multiplication.
multi.result <- v1*v2
print(multi.result)

# Vector division. divi.result
<- v1/v2 print(divi.result)
```

R Basics: Data Structures

R Objects: Vectors

- To assign multiple values to a variable, we can use an R object called a 'vector'
- A vector is a sequence/collection of data elements of the same basic type. Members in a vector are officially called components.

For Example: my_vector = c(14,26,38,30)

- To access a specific element in the vector, we simply need to call variable_name[i], 'i' being the element's position in the vector.

For example: vect[3] would return 38

Vector arithmetic

Two vectors of same length can be added, subtracted, multiplied or divided giving the result as a vector output.

```
[1] 7 19 4 13 1 13
[1] -1 -3 4 -3 -1 9
[1] 12 88 0 40 0 22
[1] 0.7500000 0.7272727 Inf 0.6250000 0.0000000 5.5000000
```

When we execute the above code, it produces the following result –

R Objects: Matrices

• A matrix is a sequence/collection of data elements of the same basic type arranged in a two-dimensional rectangular layout. • Being a 2-dimensional object, in order to obtain a specific value within the matrix, 2 coordinates needs to be entered. For example: my_matrix[i,j] would return the element on the ith row, in the jth column

• my_matrix[i,] would return the entire ith row

• my_matrix[,j] would return the entire jth column

Matrices

A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
# Create a matrix.
```

```
M = matrix( c('a','a','b','c','b','a'), nrow = 2,
ncol = 3, byrow = TRUE) print(M)
```

When we execute the above code, it produces the following result .

```
[,1] [,2] [,3]
[1,] "a" "a" "b"
[2,] "c" "b" "a"
```

PRACTICAL–2

Importing Packages R:

Packages

- R Packages are collections of R functions and data sets
- Some standard ones come with R installation
- Others can be installed in a few clicks in R studio, or using install. Packages (“package name”) function. You can choose the CRAN Mirror closest to your location, but the default R studio is consistently good all over the world.
- Some have to be downloaded

(from http://cran.rproject.org/), or through Google and manually installed

- Once installed we need to call the package in when needed using “library(“package name”)

Binding

- Binding columns: If 2 datasets, a dataset and a vector, or 2 vectors have the same number of values (rows in the case of datasets), they can be placed together into one same dataset using cbind()
- This is different from « merging » (see later chapter), hence there is no row matching system: rows need to be in the exact same order for the data to make sense.
- See example:- Binding rows: If 2 datasets have the same columns(order, data types, names), one can be appended under the other using rbind()

PRACTICAL – 3

PROBLMS ON LPP USING R

PRACTICAL – 3(A)

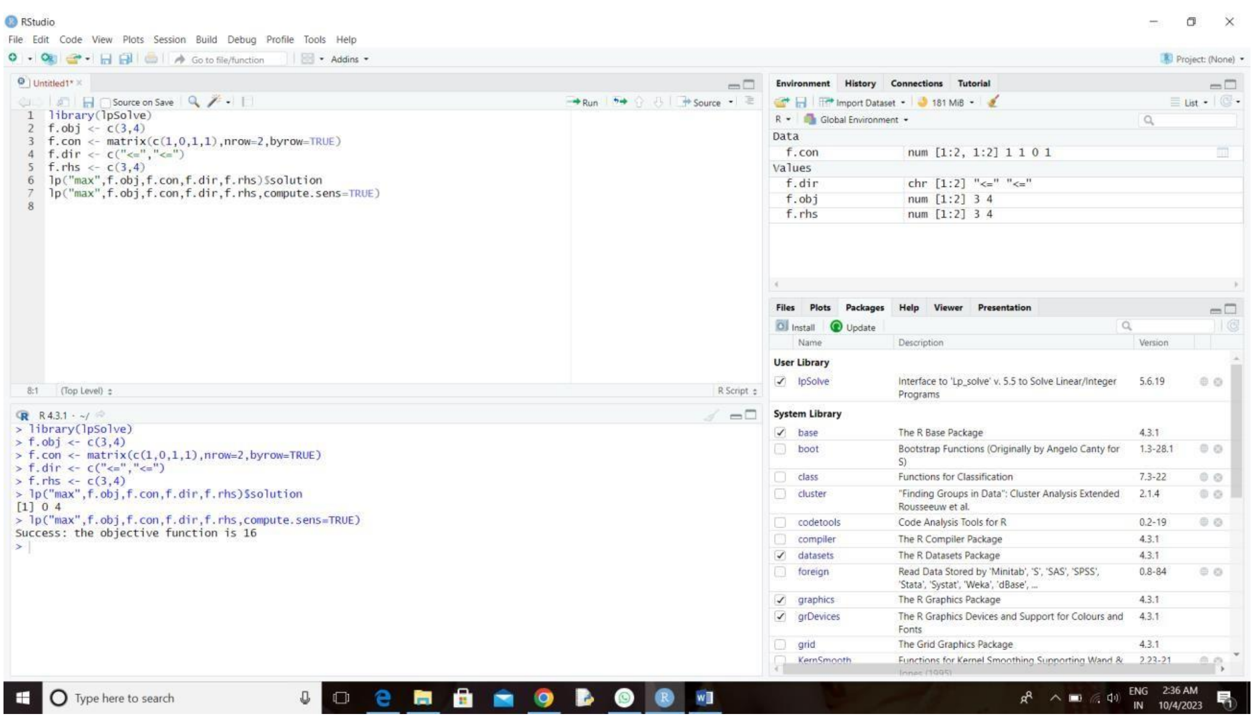
1. Maximize Z = 3X1+4X2

SUBJECT TO

X1+0X2<=3

X1+X2 <=4

X1, X2 >= 0



PRACTICAL – 3(B)

2. Minimize Z = 4x+8y

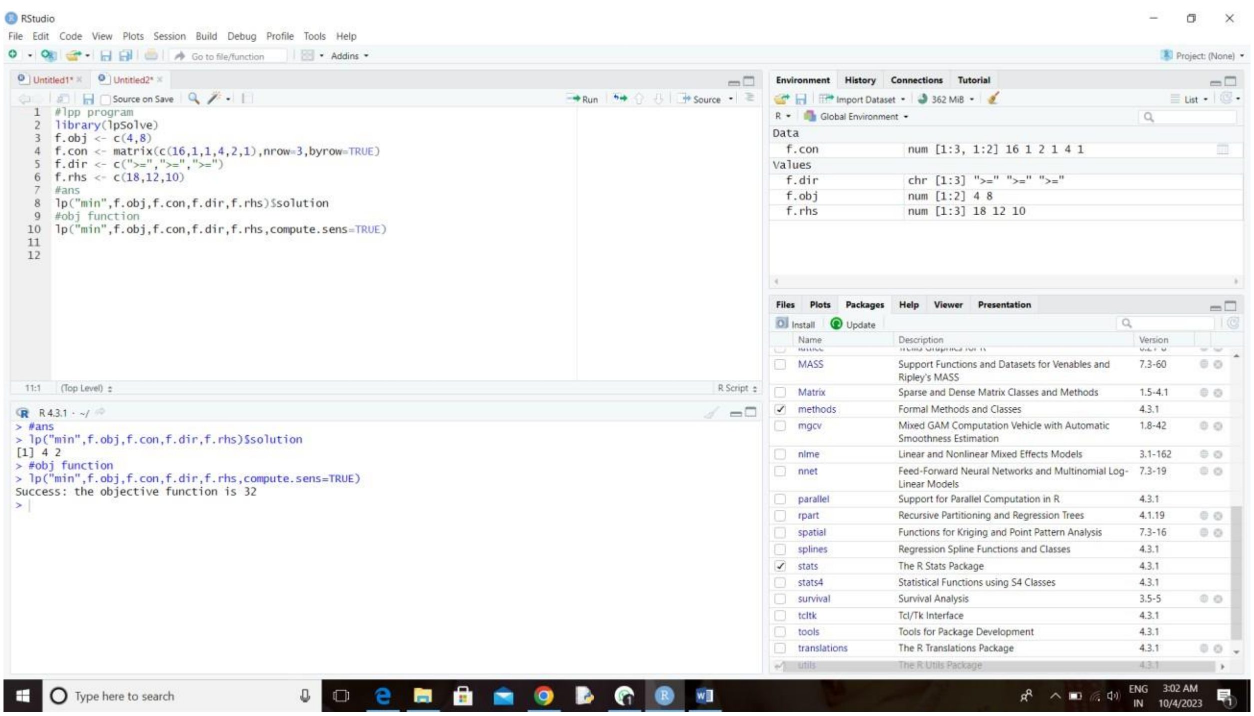
Subject to

16X+y>=18

X+4y>=12

2X+Y>=10

X,Y>= 0



PRACTICAL – 3(C)

Solve

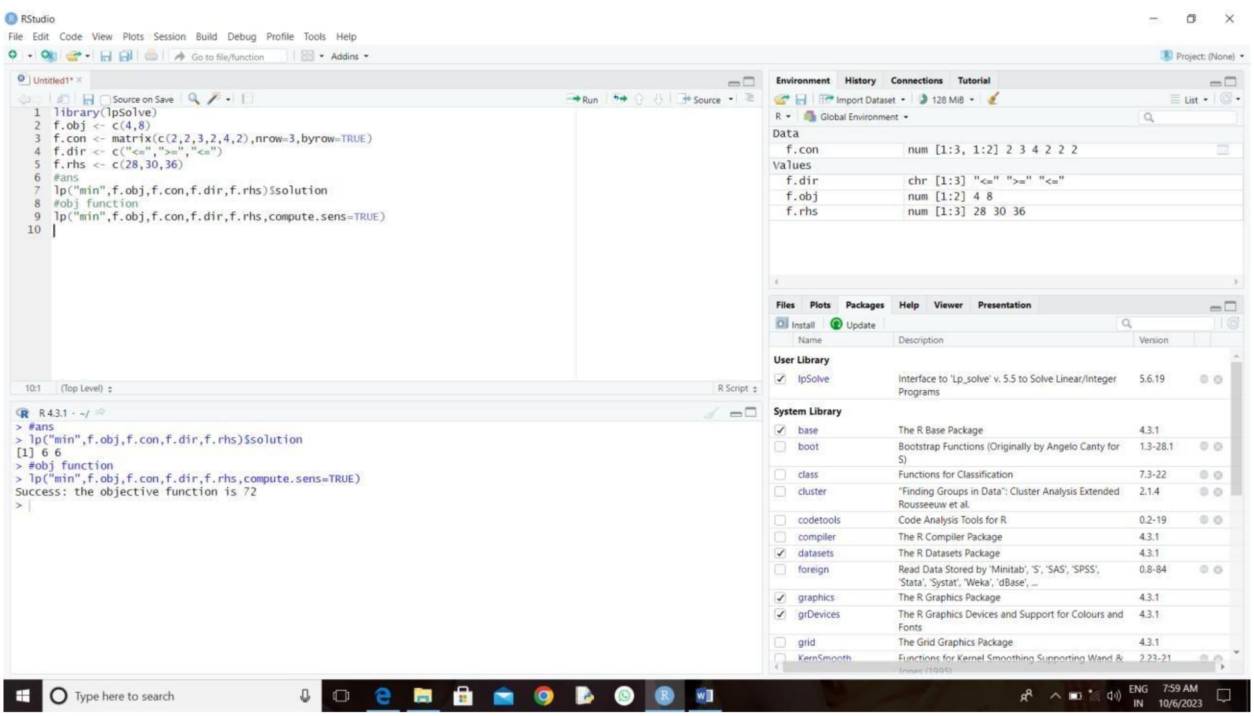
Minimize Z = 4X+8Y

Subject to

2X+2Y <= 28

3X+2Y>=30

4X+2Y <=36 X,Y>= 0



PRACTICAL – 4

SOLVE PROBLEMS ON LPP USING SIMPLEX METHOD USIN R

PRACTICAL – 4(A)

Maximization

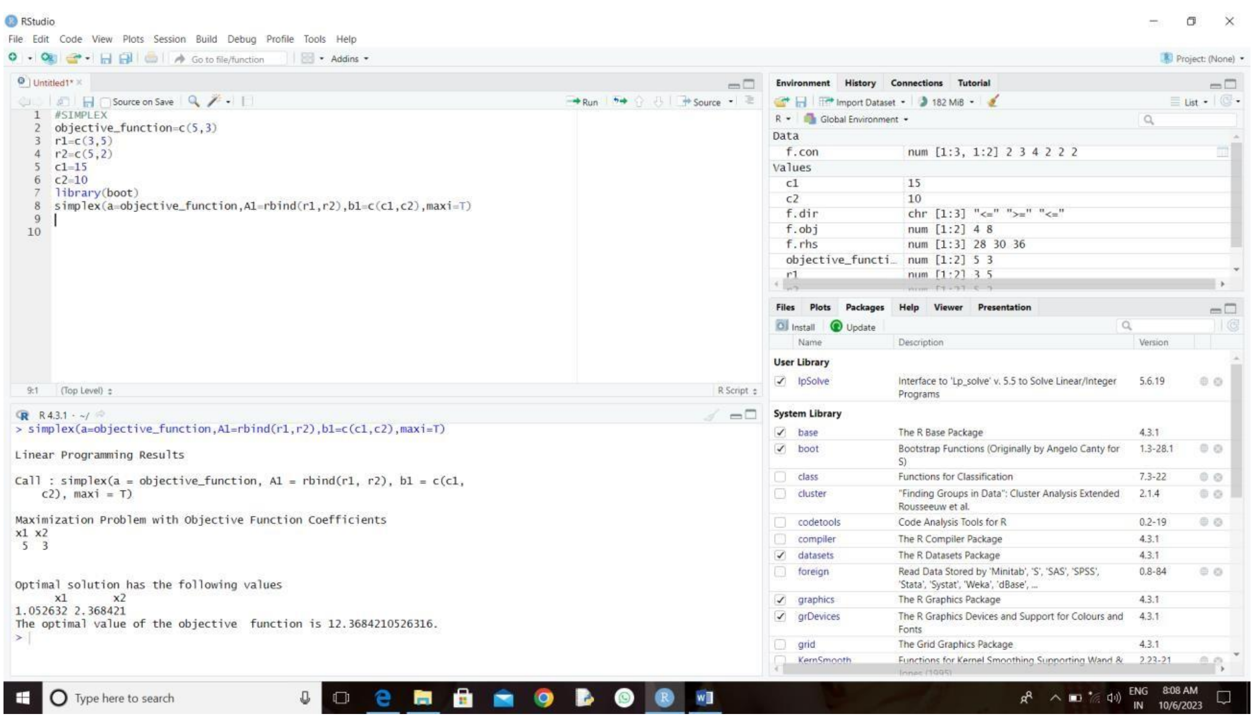
Z = 5X1 +3X2

SUBJECT TO

3X1+5X2 <=15

5X1+2X2 <= 10

X1,X2>= 0



PRACTICAL – 4(B)

Maximize

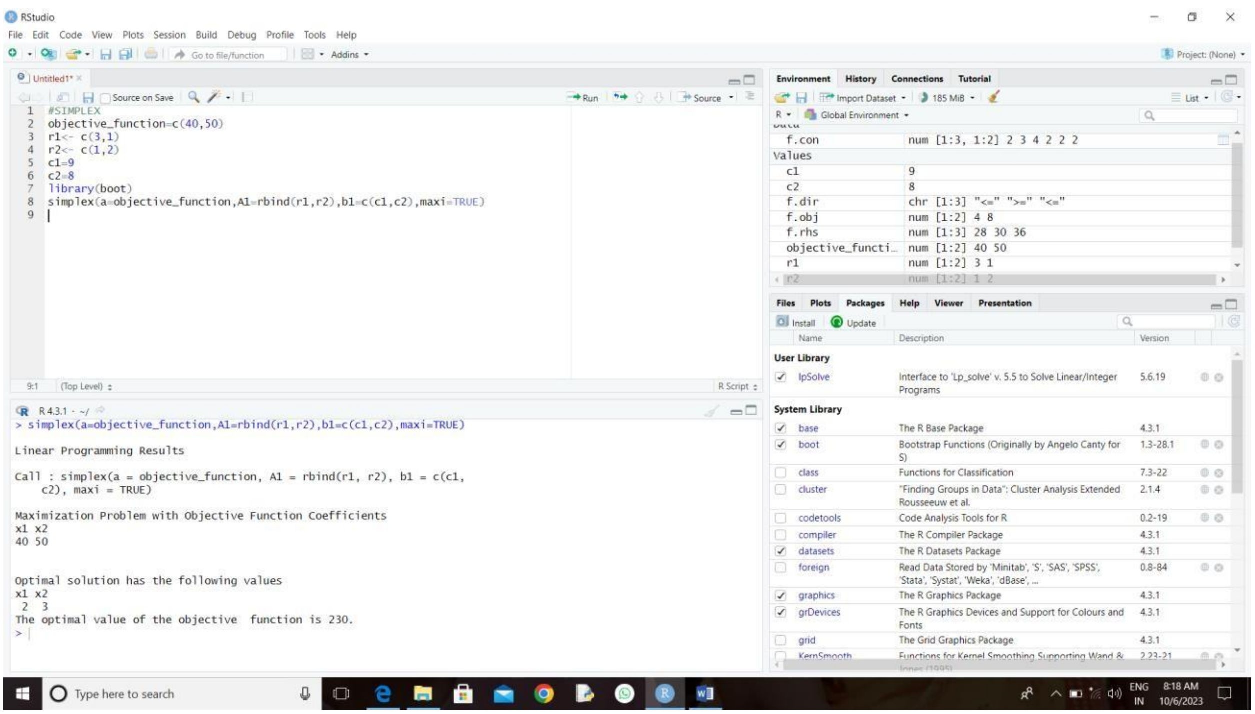
Z= 40X1 + 50X2

SUBJECT TO

3 X1 + X2 <=9

X1 +2X2 <=8

X1,X2>=0



PRACTICAL – 4(C)

Maximize

Z = 10X1 +6 X2 +4X3

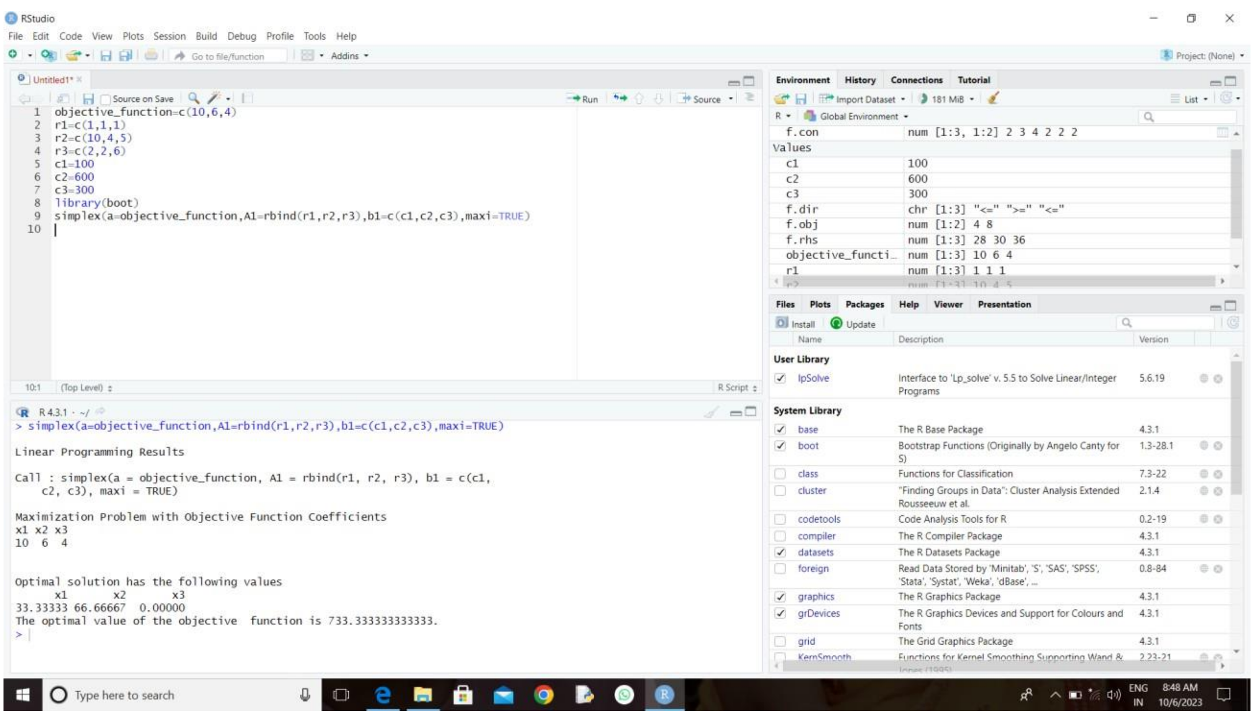
SUBJECT TO

X1+X2+X3 <= 100

10X1 + 4 X2 +5X3 <=600

2X1 +2X2 + 6X3 <= 300

X1,X2,X3>= 0

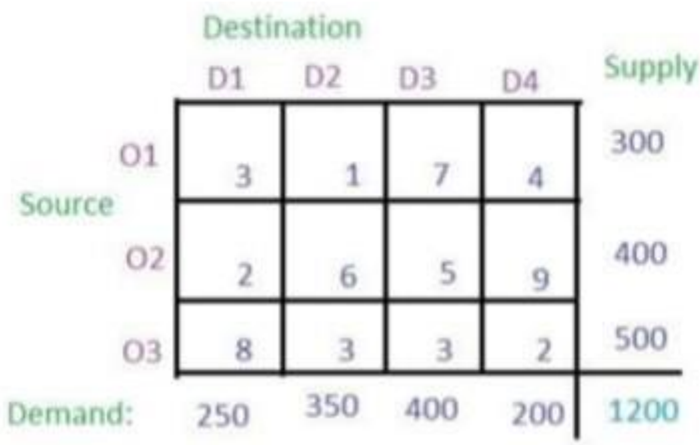


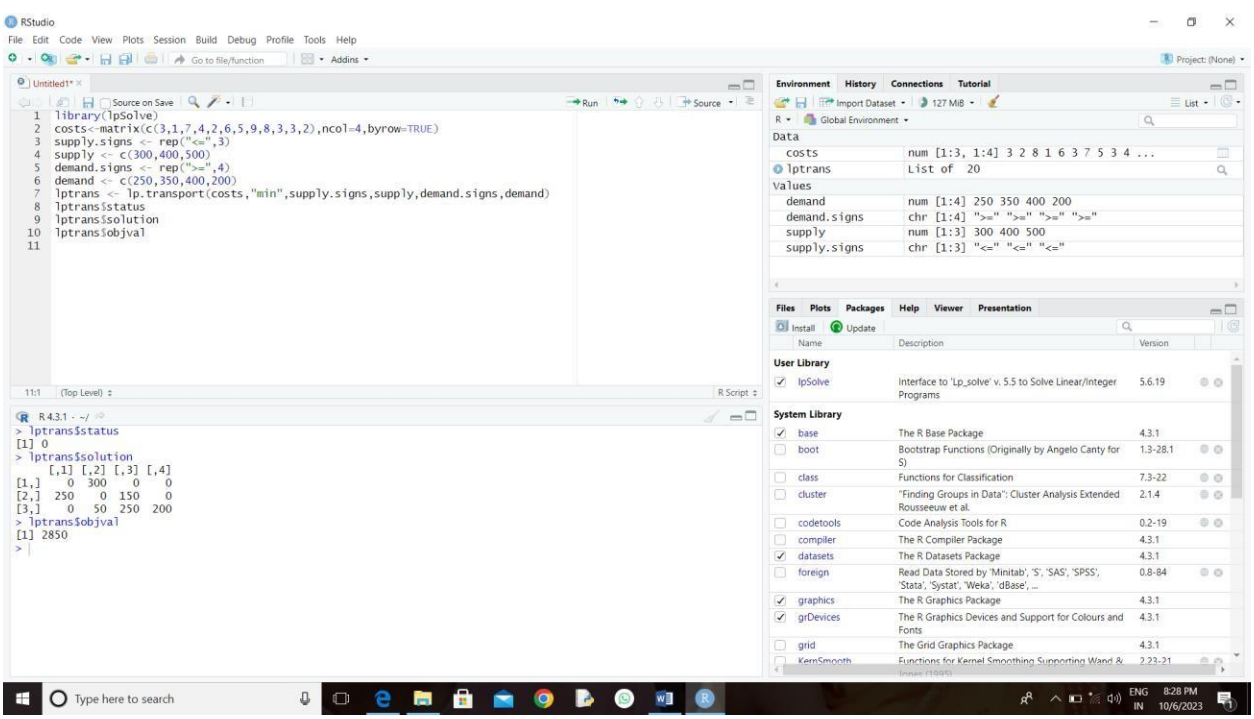
PRACTICAL – 5

SOLVING PROBLEMS ON TRANSPORTATION PROBLE USING R

PRACTICAL - 5(A)

SOLVE THE FOLLOWING TRANSPORTATION PROBLEM FOR MINIMAL COST





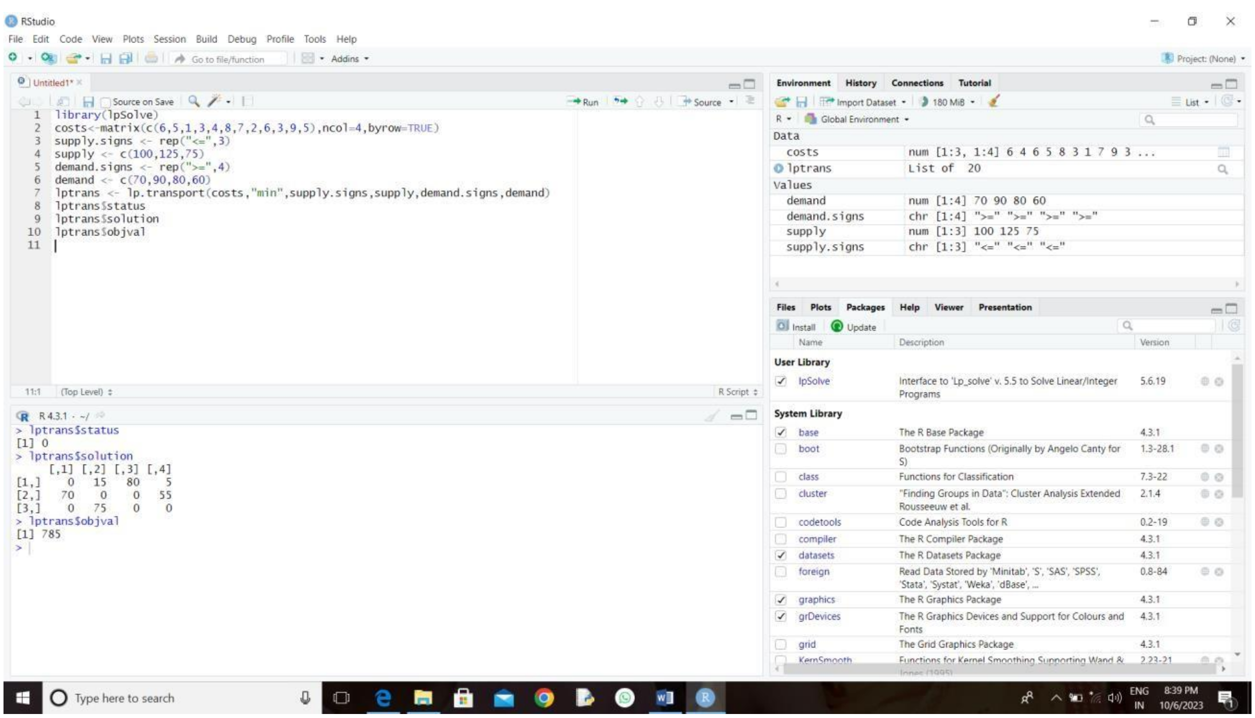
PRACTICAL – 5(B)

SOLVED EXAMPLES

Example 1 :

Solve the following transportation problem for the optimum cost

	D_1	D_2	D_3	D_4	Capacity
O_1	6	5	1	3	100
O_2	4	8	7	2	125
O_3	6	3	9	3	75
Demand	70	90	80	60	300

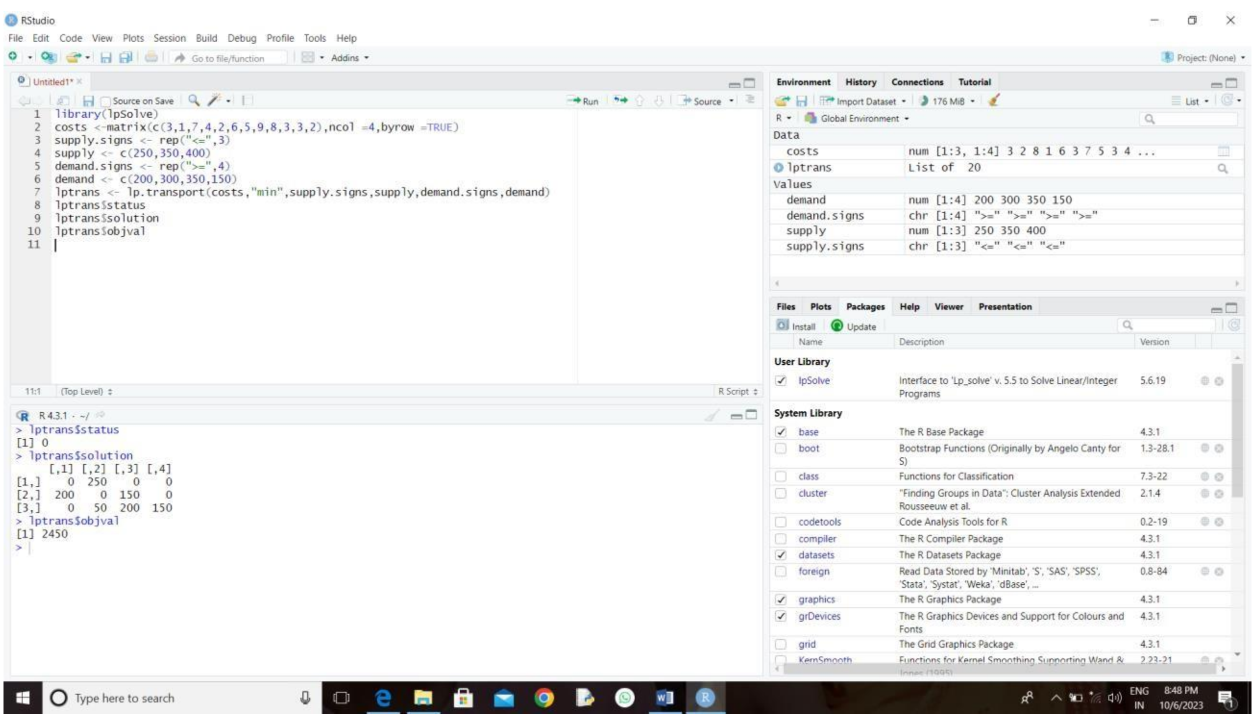


PRACTICAL - 5(C)

SOLVE FOR OPTIMUM VALUE

Destination

Source	D1	D2	D3	D4	Supply(S_i)
O1	3	1	7	4	250
O2	2	6	5	9	350
O3	8	3	3	2	400
Demand(D_j):	200	300	350	150	



PRACTICAL 6

SOLVE ASSIGNMENT PROBLM USING R

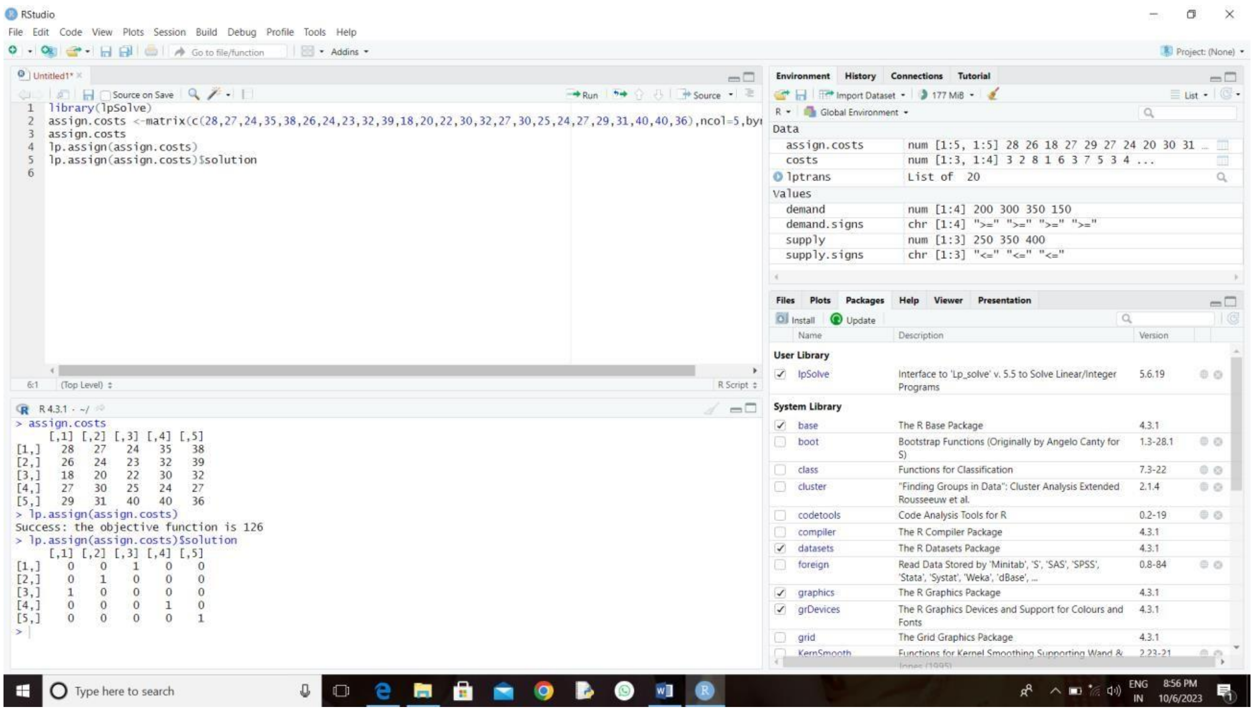
PRACTICAL – 6(A)

Example 1 :

M/s. Global Pvt. Ltd. has five jobs A, B, C, D & E to be done by five Employees Vishesh, Dhruv, Nilanj, Aditya, and Shivam. Each Employee is assigned one and only one job. The number of hours each Employee would take to complete each job is given by the following table.

Employees	A	B	C	D	E
			Jobs		
Vishesh	28	27	24	35	38
Dhruv	26	24	23	32	39
Nilanj	18	20	22	30	32
Aditya	27	30	25	24	27
Shivam	29	31	40	40	36

Find the assignment of jobs to Employee in such a way that the total time taken to perform 5 jobs is minimum.



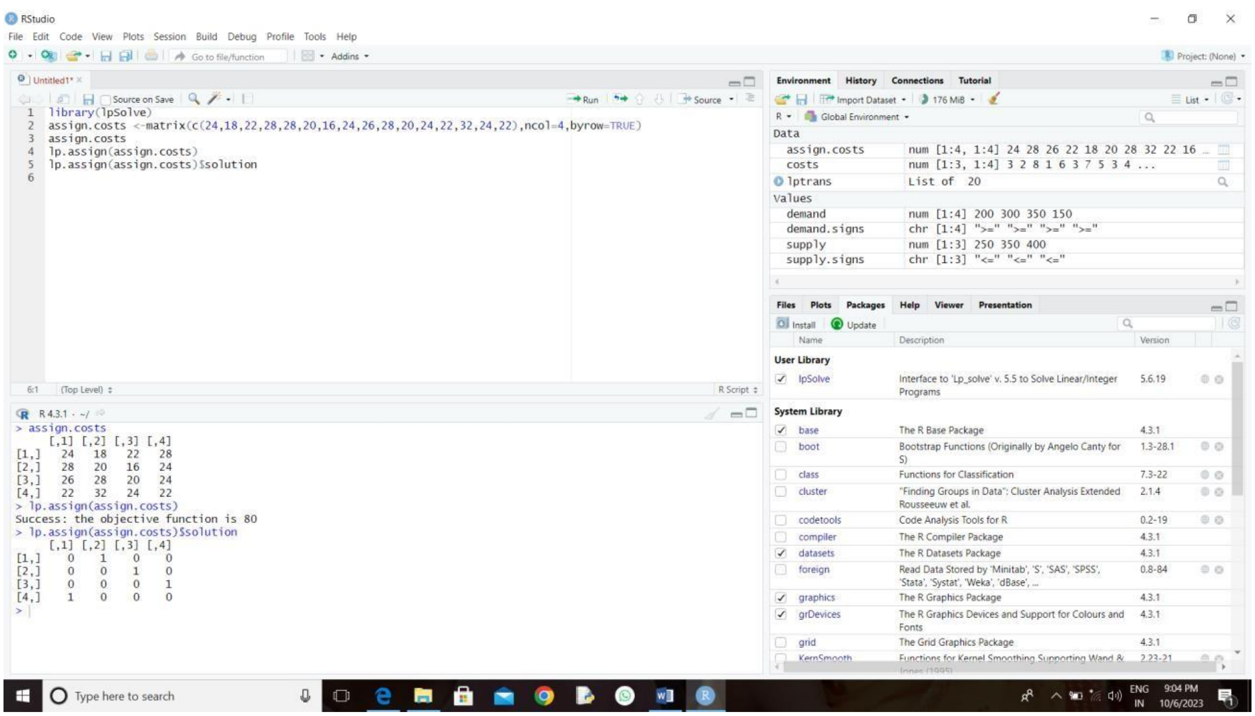
PRACTICAL - 6(B)

Example 3 :

The MBI manufacturing company plans to manufacturer 4 types of ne minicomputers. Each of the plant has manufacturing capacity for one product only. Th unit manufacturing cost for producing the different minicomputers at the four plants a shown in table below. What is the lowest total manufacturing cost?

Minicomputers Type

Plants	1	2	3	4
1	24	18	22	28
2	28	20	16	24
3	26	28	20	24
4	22	32	24	22



PRACTICAL – 6(C)

Example 4 :

Case Study

i) A departmental store agency runs five stores located at different parts of grea[ter] Bombay. Each store is administered by the manager appointed by the agency. Th[e] manager resides in the different parts of the city. The agency reimburses the car tra[vel] expenses incurred by the managers in commuting of the work from their residence [to] the stores to which they are assigned. The basis of reimbursements is

A fixed sum of ₹ 300/- per month for repair and maintenance. A variable amount [at] the rate of ₹ 1.60 per Km. of travel incurred during the month. All stores work [for] 25 days in a month.

The distance in kilometers from a manager's residence to stores is displayed in [the] following table. Find the optimal assignment of managers to stores so that the mon[thly] expenditure to be incurred by the agency on car travel of managers is minimum.

ii) After having operated optimal assignment for some time, Mr. Vansh request the agen[cy] to assign him store S1, which is the closest to his residence, as he has been advised [not] to take long journeys. If the agency agrees to his request, how should the pres[ent] assignment be changed and how much extra will it cost to the agency?

Managers	Stores (Distance in Km.)				
	S_1	S_2	S_3	S_4	S_5
Vansh	4	10	12	18	17
Aayushi	7	16	16	22	18
Dhvani	8	6	9	19	21
Arth	11	12	15	12	13
Kripesh	9	14	19	18	14