Practical File

Of

Data Science Laboratory

(LPECS-109)

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

Submitted By :

Aman Chauhan

U.R.N - 1805158

Department of Computer Science and Engineering

Guru Nanak Dev Engineering College

Table Of Contents

Practical 1

Introduction to R

R is an interpreted computer programming language which was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand.

Features of R programming:

- It is a simple and effective programming language which has been well developed.
- It is data analysis software.
- It is a well-designed, easy, and effective language which has the concepts of user-defined, looping, conditional, and various I/O facilities.
- It has a consistent and incorporated set of tools which are used for data analysis.
- For different types of calculation on arrays, lists and vectors, R contains a suite of operators.
- It provides an effective data handling and storage facility.

Why use R:

The important task in data science is the way we deal with the data: clean, feature engineering, feature selection, and import. It should be our primary focus. Data scientist job is to understand the data, manipulate it, and expose the best approach. For machine learning, the best algorithms can be implemented with R. Keras and TensorFlow allow us to create high-end machine learning techniques.

Practical 2

Write a program to implement use of Variables and Data types in R.

Program:

```
TRUE -> varLogical

cat("The data type of ", varLogical, " is ", class(varLogical), "\n")

108 -> varNumeric

cat("The data type of ", varNumeric, " is ", class(varNumeric), "\n")

108L -> varInt

cat("The data type of ", varInt, " is ", class(varInt), "\n")

10 + 8i -> varComplex

cat("The data type of ", varComplex, " is ", class(varComplex), "\n")

"R" -> varChar

cat("The data type of ", varChar, " is ", class(varChar), "\n")
```

Output

```
The data type of  TRUE  is  logical
The data type of  108  is  numeric
The data type of  108  is  integer
The data type of  10+8i  is  complex
The data type of  R  is  character
```

Practical 3

Program to implement Arithmetic, Logical and Matrix operations in R.

Arithmetic Operations Program:

```
33 -> x

19 -> y


cat("The addition of ", x, " and ", y, " is ", x+y, "\n\n")

cat("The subtraction of ", x, " and ", y, " is ", x-y, "\n\n")

cat("The multiplication of ", x, " and ", y, " is ", x*y, "\n\n")

cat("The division of ", x, " and ", y, " is ", x/y, "\n\n")

cat("The modulus of ", x, " and ", y, " is ", x%%y, "\n\n")
```

Output

```
The addition of  33  and  19  is  52

The subtraction of  33  and  19  is  14

The multiplication of  33  and  19  is  627

The division of  33  and  19  is  1.736842

The modulus of  33  and  19  is  14
```

Logical And Matrix Program:

```
matrix(c(0,1,0,1), nrow=4, ) -> matrixA

matrix(c(0,0,1,1), nrow=4, ncol=1) -> matrixB


# Make a And Gate

matrix(matrixA & matrixB, nrow=4, ncol=1) -> andGate

cat("And Gate:", andGate, "\n\n")


# Make a Or Gate

matrix(matrixA | matrixB, nrow=4, ncol=1) -> orGate

cat("Or Gate: ", orGate)
```

Output

```
And Gate: FALSE FALSE FALSE TRUE

Or Gate:  FALSE TRUE TRUE TRUE
```

Practical 4

Write a program to implement the concept of functions.

Program:

```
add <- function(a,b) {
        cat("Sum of ", a, " and ", b, " is ", a+b, "\n") }

subtract <- function(a,b) {
        cat("Subtraction of ", a, " and ", b, " is ", a-b, "\n") }

multiply <- function(a,b) {
        cat("Multiplication of ", a, " and ", b, " is ", a*b, "\n") }

divide <- function(a,b) {
        cat("Division of ", a, " and ", b, " is ", a/b, "\n") }

33 -> a
27 -> b
cat("Two numbers are ", a, " and ", b, "\n")
```

add(a,b)

subtract(a,b)

multiply(a,b)

divide(a,b)

Output

```
Two numbers are  33  and  27
Sum of  33  and  27  is  60
Subtraction of  33  and  27  is  6
Multiplication of  33  and  27  is  891
Division of  33  and  27  is  1.222222
```

Practical 5

Write a program to implement control structures.

Program:

```
20 -> n

while(1) {

        if (n == 11) {

        break }

        else {

        cat(n, "\n") }

        n -1 -> n

}
```

Output

Practical 6

Write a program to read and write data from a dataset.

Program:

```
read.csv("Dataset.csv") -> dataset

print(dataset)
```

Output

```
  ID Name Salary Start_Date Department
1  1 Rick 623.30 2012-01-01         IT
2  2  Dan 515.20 2013-09-23 Operations
3  3 Ryan 729.00 2014-05-11         HR
4  4 Gary 843.25 2015-03-27    Finance
5  5 Nina 578.00 2013-05-21         IT
```

```
read.csv("Dataset.csv") -> dataset

subset(dataset,as.Date(Start_Date)>as.Date("2014-01-01")) -> details

write.csv(details, "output.csv")

read.csv("output.csv") -> output

print(output)
```

Output

```
  X ID Name Salary Start_Date Department
1 3  3 Ryan 729.00 2014-05-11         HR
2 4  4 Gary 843.25 2015-03-27    Finance
```

Practical 7

Write a program to study linear algebra for data science.

Program:

```
matrix(c(5:16), nrow = 4,ncol=3) -> matrix1

matrix(c(1:12), nrow = 4,ncol=3) -> matrix2

matrix1 + matrix2 -> Sum

print("Addition Of Matrices")

print(Sum)
```

Output

```
[1] "Addition Of Matrices"
     [,1] [,2] [,3]
[1,]    6   14   22
[2,]    8   16   24
[3,]   10   18   26
[4,]   12   20   28
```

```
matrix1 - matrix2 -> Subtract

print("Subtraction Of Matrices")

print(Subtract)
```

Output

```
[1] "Subtraction Of Matrices"
     [,1] [,2] [,3]
[1,]    4    4    4
[2,]    4    4    4
[3,]    4    4    4
[4,]    4    4    4
```

matrix1 * matrix2 -> Multiply

print("Multiplication Of Matrices")

print(Multiply)

Output

```
[1] "Multiplication Of Matrices"
     [,1] [,2] [,3]
[1,]    5   45  117
[2,]   12   60  140
[3,]   21   77  165
[4,]   32   96  192
```

matrix1 + matrix2 -> Division

print("Division Of Matrices")

print(Division)

Output

```
[1] "Division Of Matrices"
     [,1] [,2] [,3]
[1,]    6   14   22
[2,]    8   16   24
[3,]   10   18   26
[4,]   12   20   28
```

Practical 8

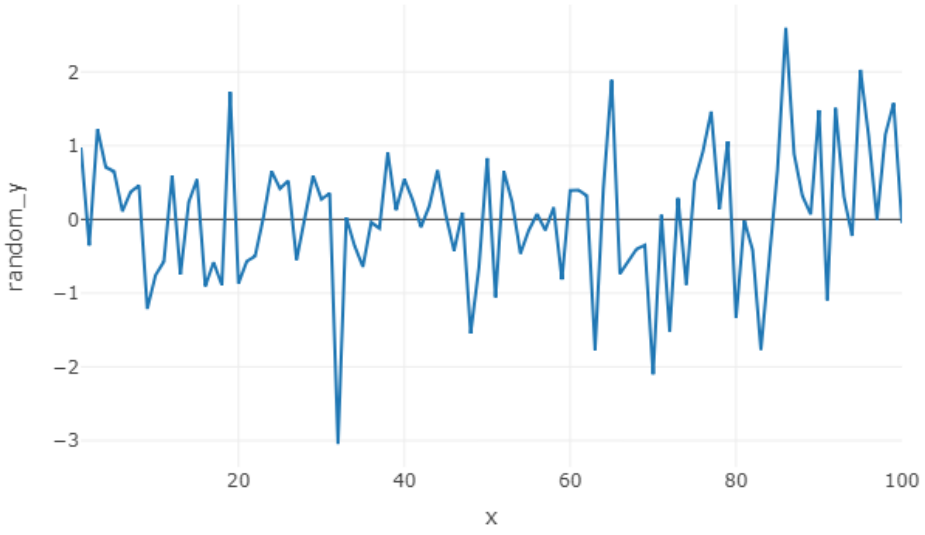
Write a program to study various libraries and packages for Data Visualization in R.

Program:

Module 1 - Plotly

```
library(plotly)

x <- c(1:100)

random_y <- rnorm(100, mean = 0)

data <- data.frame(x, random_y)

fig <- plot_ly(data, x = ~x, y = ~random_y, type = 'scatter', mode = 'lines')
```

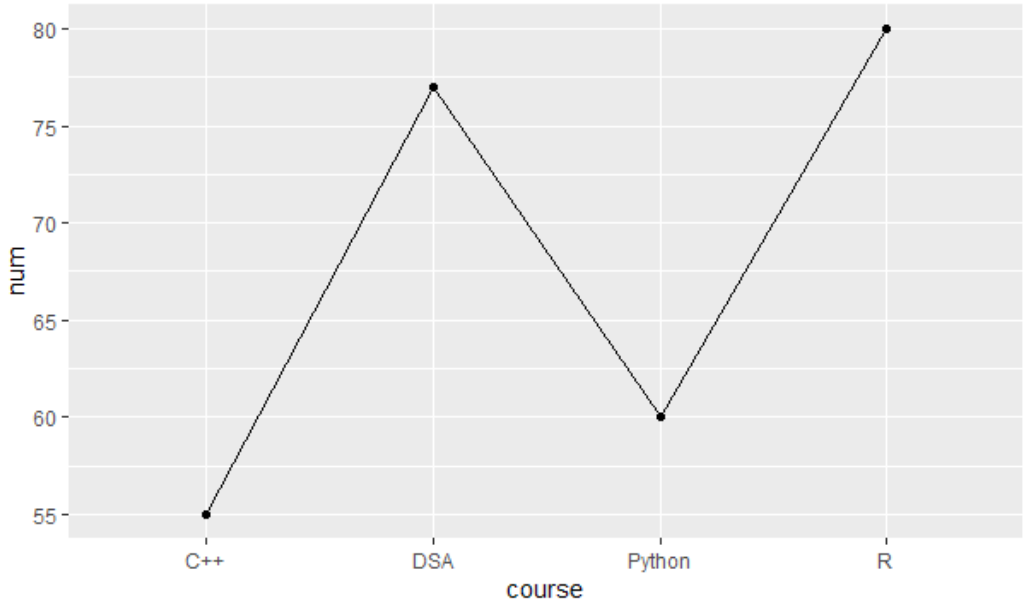
Output



Module 2 - ggplot2

```
library(ggplot2)

val <-data.frame(course=c('DSA','C++','R','Python'), num=c(77,55,80,60))

ggplot(data=val, aes(x=course, y=num, group=1)) + geom_line() + geom_point()
```

Output



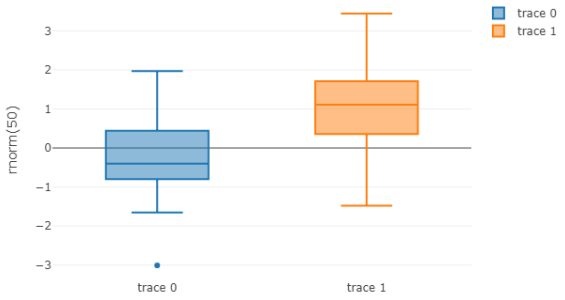
Practical 9

Write a program to find data distribution using a box and scatter plot.

Program:

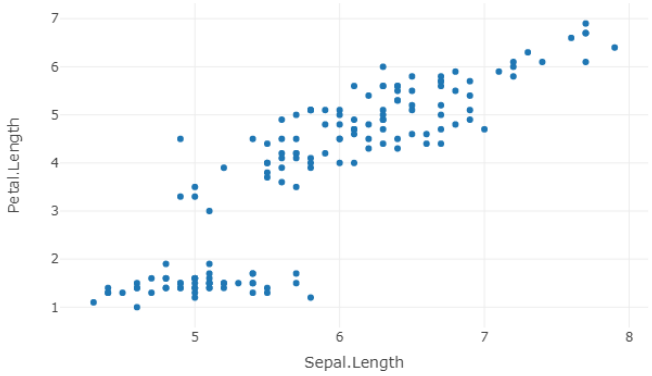
```
library(plotly)

fig <- plot_ly(y = ~rnorm(50), type = "box")

fig <- fig %>% add_trace(y = ~rnorm(50, 1))
```

Output




```
fig <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length)
```

Output



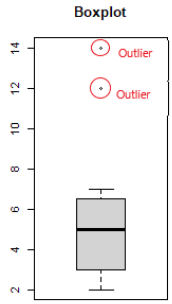
Practical 10

Write a program to find outliers using plot.

Program:

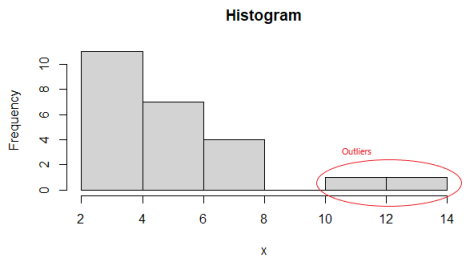
```
x = c(5,2,3,4,5,4,3,3,6,7,5,4,4,2,2,5,7,6,7,3,5,7,12,14)

boxplot(x, main = "Boxplot")
```

Output




```
x = c(5,2,3,4,5,4,3,3,6,7,5,4,4,2,2,5,7,6,7,3,5,7,12,14)

hist(x, main = "Histogram")
```

Output



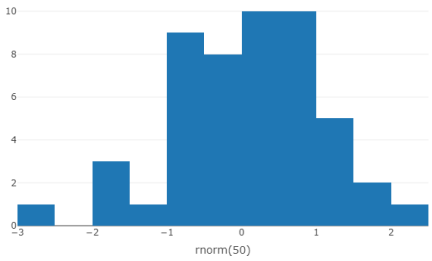
Practical 11

Write a program to plot Histogram and Bar chart on sample data.

Program:

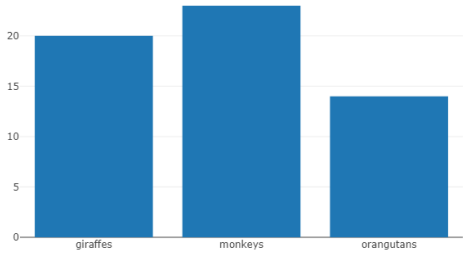
```
library(plotly)

fig <- plot_ly(x = ~rnorm(50), type = "histogram")

fig
```

Output




```
fig <- plot_ly(x = c("giraffes", "orangutans", "monkeys"), y = c(20, 14, 23), name
= "SF Zoo", type = "bar")

fig
```

Output



Mini Project

To develop a project to use various Data Science constructs like box, scatter plot, Histogram, Dimensionality, Transformation to visualize a sample dataset.

Introduction

Coronaviruses are a large family of viruses that are known to cause illness ranging from the common cold to more severe diseases such as Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS). A novel coronavirus (COVID-19) was identified in 2019 in Wuhan, China. This is a new coronavirus that has not been previously identified in humans.

In this mini-project, we will visualize the data of covid in the form of charts. Also we will see the trends based on that data.

Data Source

Kaggle : https://www.kaggle.com/datasets/anandhuh/latest-covid19-india-statewise-data

Steps To Follow :

- Data Collection
- Import required modules
- Import Dataset
- Data Visualization

Objectives:

- To visualize the tabular dataset in the form of charts and graphs.
- To find the relationship between 2 variables.

Step 1 : Data Collection

Source : https://www.kaggle.com/datasets/anandhuh/latest-covid19-india-statewise-data

Step 2 : Importing Modules

```
import pandas as pd
import seaborn as sns
```

Step 3 : Importing Dataset

```
data = pd.read_csv("/content/drive/MyDrive/Colab
Notebooks/Dataset.csv")
```

```
data.head()
```

```
             State/UTs  Total Cases  Active  Discharged  Deaths
Active Ratio  \
0  Andaman and Nicobar        10034       0        9905     129
0.00
1       Andhra Pradesh      2319645      43     2304872   14730
0.00
2    Arunachal Pradesh        64495       8       64191     296
0.01
3                Assam       724200    1349      716212    6639
0.19
4                Bihar       830506      12      818238   12256
0.00

   Discharge Ratio  Death Ratio  Population
0            98.71         1.29   100896618
1            99.36         0.64   128500364
2            99.53         0.46      658019
3            98.90         0.92      290492
4            98.52         1.48    40100376
```

```
data.describe()
```

```
        Total Cases       Active    Discharged         Deaths  Active
Ratio  \
count  3.600000e+01    36.000000  3.600000e+01      36.000000
36.000000
mean   1.195674e+06   320.611111  1.180855e+06   14499.027778
0.030833
std    1.768788e+06   564.396329  1.743084e+06   26911.351564
0.054110
min    1.003400e+04     0.000000  9.905000e+03       4.000000
0.000000
25%    9.910525e+04     8.000000  9.801450e+04    1104.250000
0.000000
50%    5.890705e+05    56.000000  5.826635e+05    5977.000000
0.010000
75%    1.284374e+06   329.250000  1.274812e+06   14208.000000
0.030000
max    7.875845e+06  2466.000000  7.727372e+06  147827.000000
0.260000

       Discharge Ratio  Death Ratio    Population
count        36.000000    36.000000  3.600000e+01
mean         98.845000     1.125000  3.971861e+07
```

```
std           0.486207     0.493069  5.050913e+07
min          97.660000     0.030000  6.600100e+04
25%          98.527500     0.870000  1.695473e+06
50%          98.865000     1.090000  2.410088e+07
75%          99.115000     1.412500  6.979986e+07
max          99.970000     2.340000  2.315026e+08
```

```
data.isnull().sum()
```

```
State/UTs          0
Total Cases        0
Active             0
Discharged         0
Deaths             0
Active Ratio       0
Discharge Ratio    0
Death Ratio        0
Population         0
dtype: int64
```

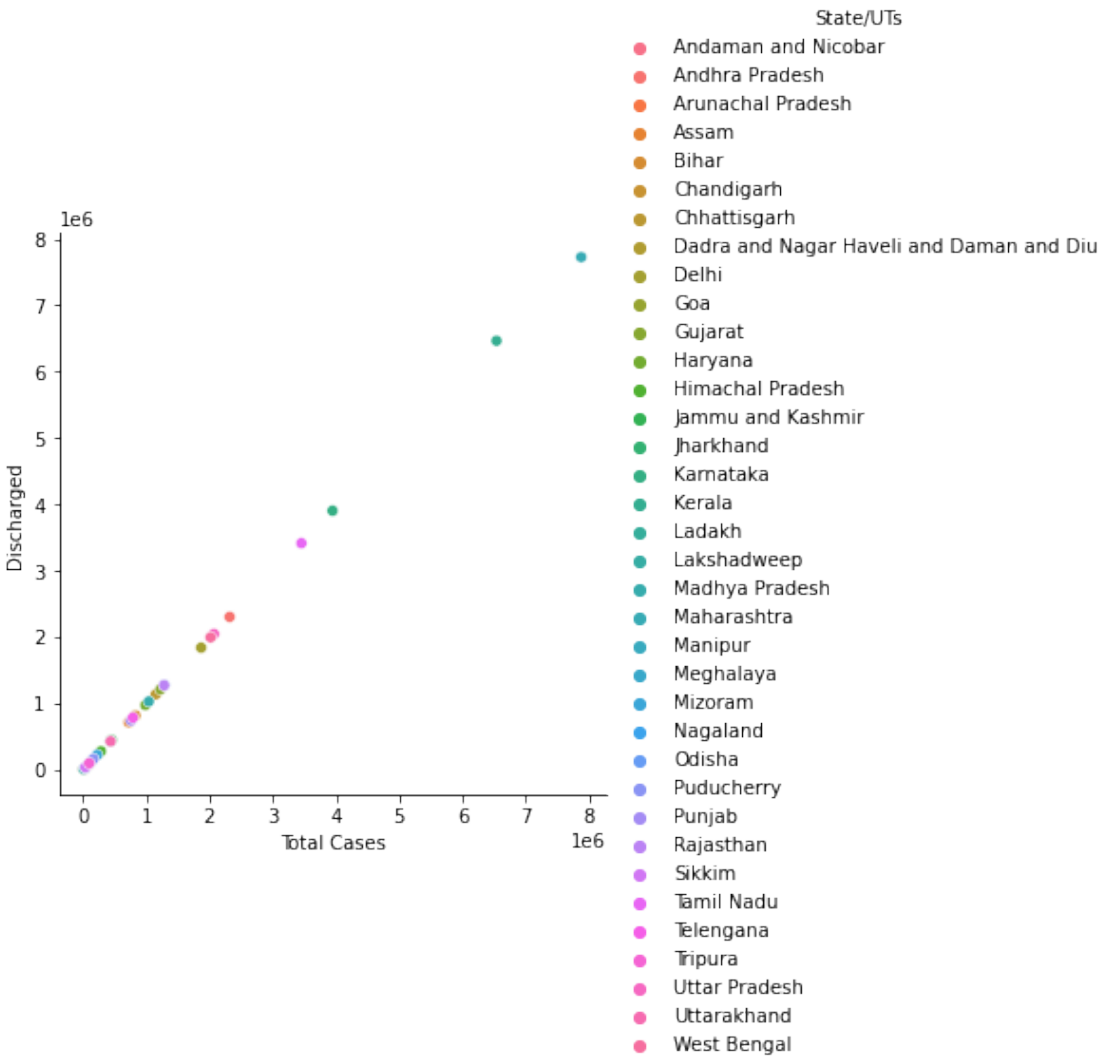
Step 4 : Data Visualization

1.Scatter Plot:

i) Relationship between Total Cases and Discharged on the basis of state.

```
sns.relplot(x="Total Cases", y="Discharged", data=data,
hue="State/UTs")
```

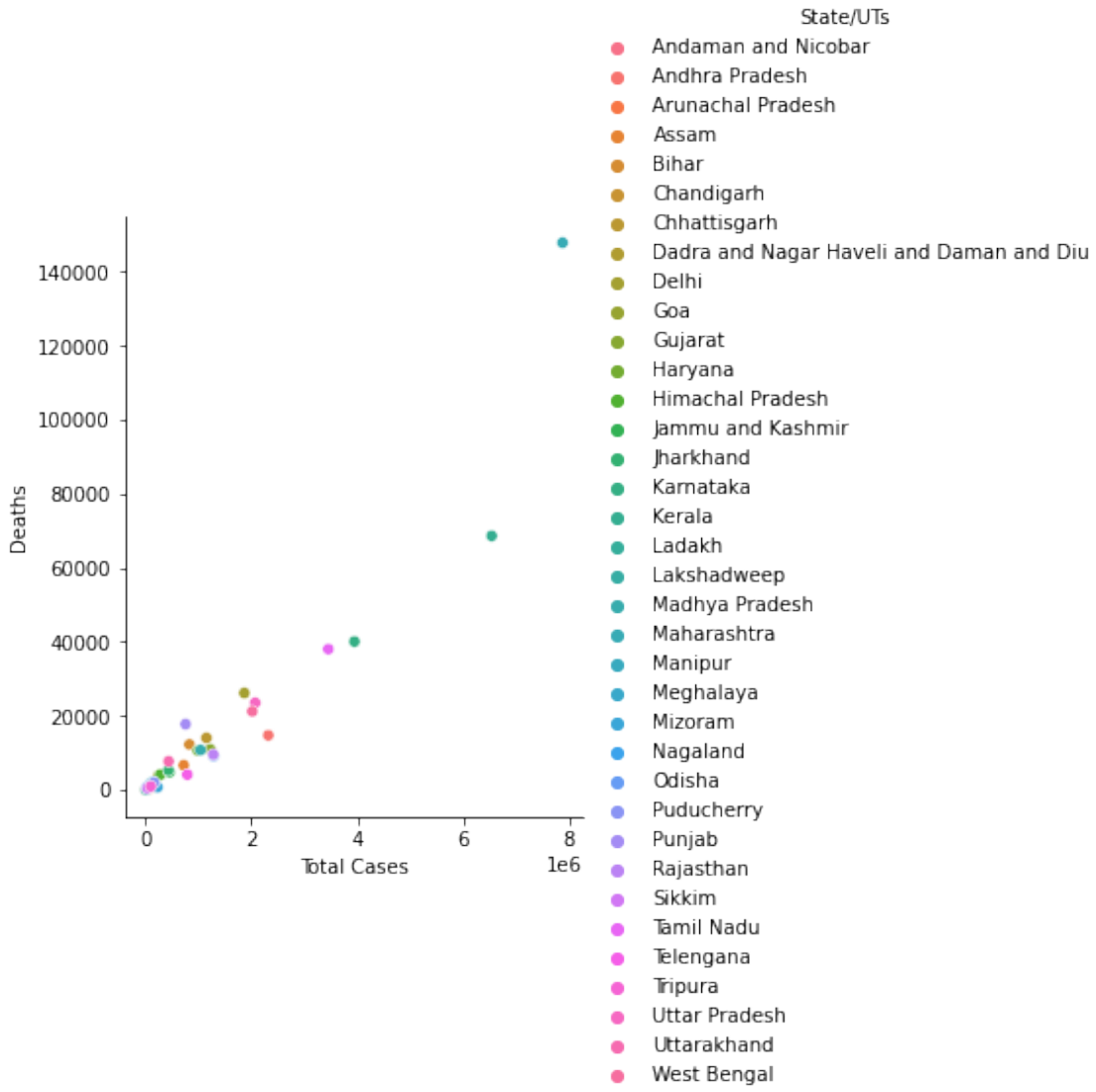
```
<seaborn.axisgrid.FacetGrid at 0x7fdab6275a10>
```



ii) Relationship between Total Cases and Deaths on the basis of state.

```
sns.relplot(x="Total Cases", y="Deaths", data=data, hue="State/UTs")
```

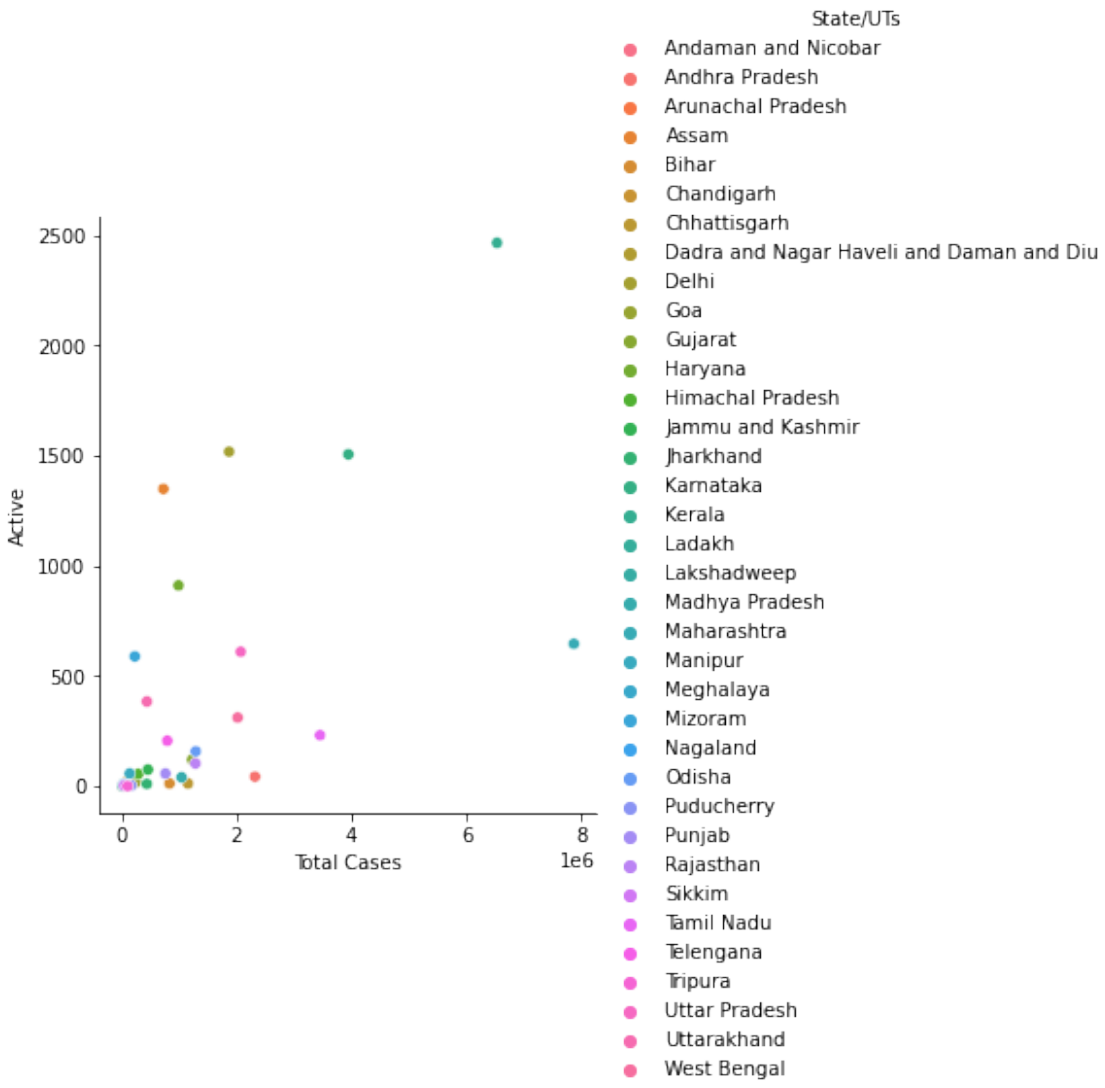
```
<seaborn.axisgrid.FacetGrid at 0x7fdab5a6abd0>
```



iii) Relationship between Total Cases and Active on the basis of state.

```
sns.relplot(x="Total Cases", y="Active", data=data, hue="State/UTs")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fdab2aed0d0>
```

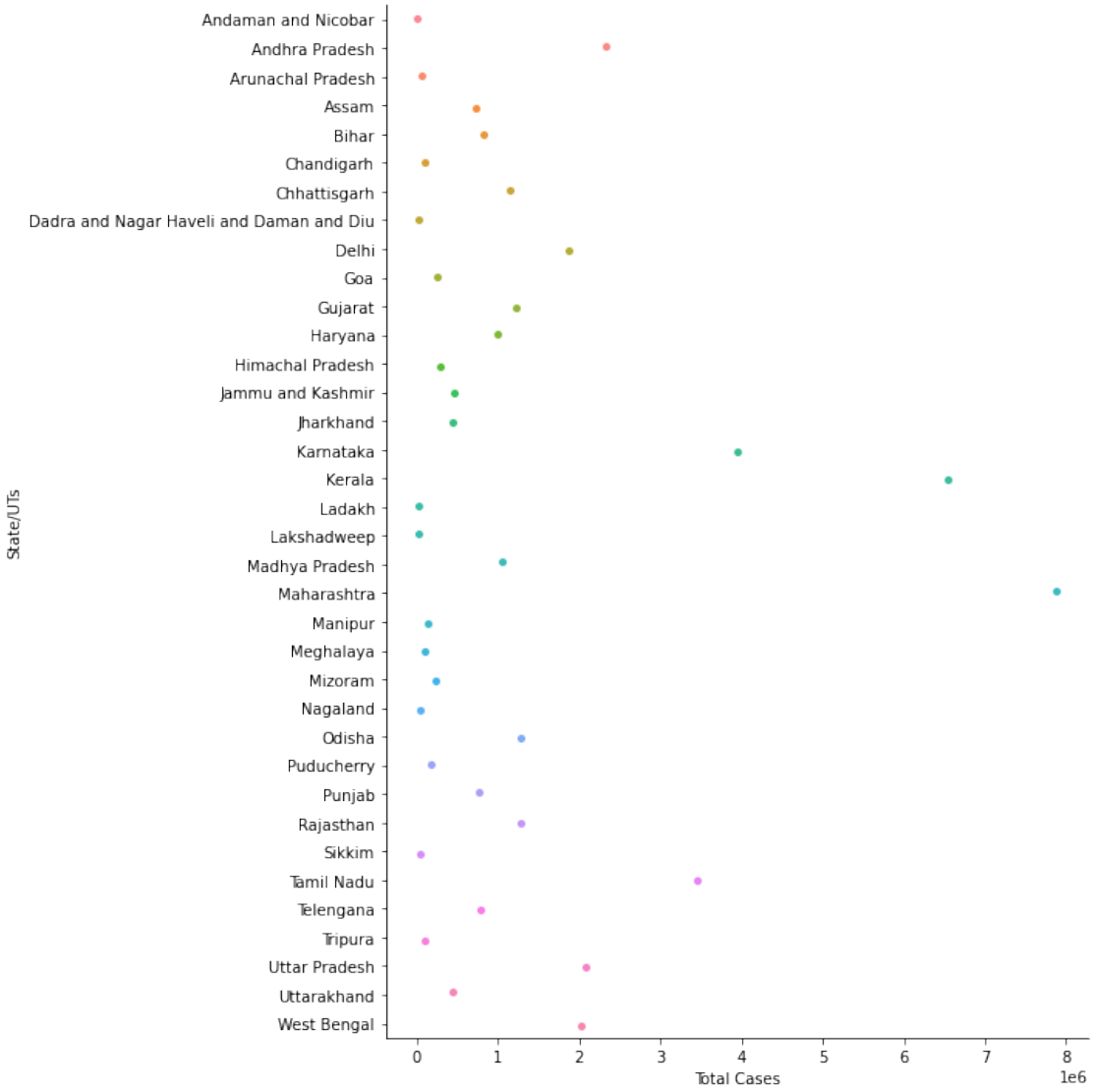


2. Category Plot

Relationship between Total Cases and States.

```
sns.catplot(x="Total Cases", y="State/UTs", data=data, height=10)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fdab2996250>
```

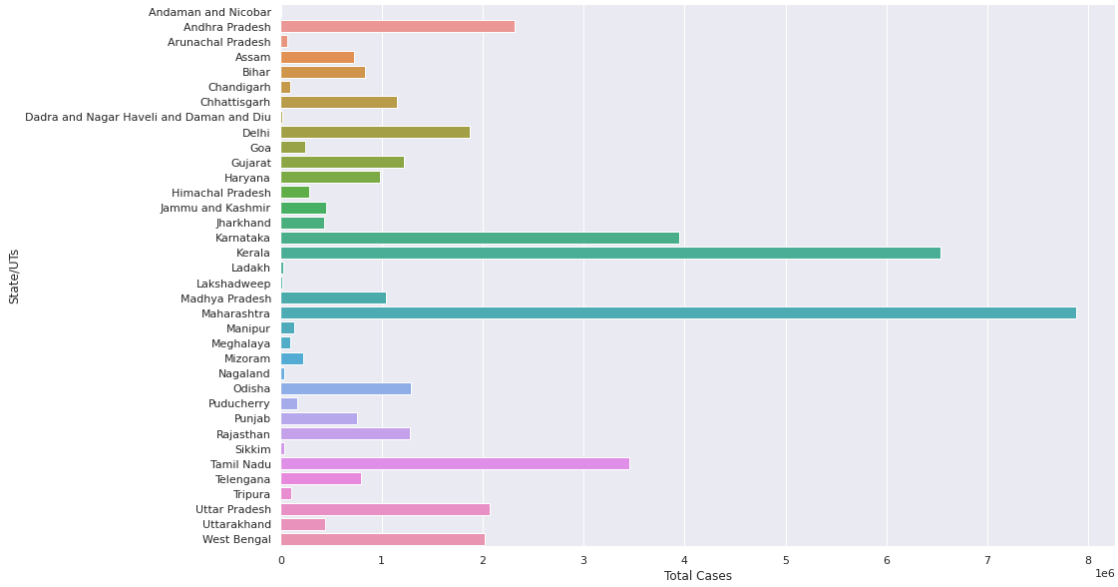


3. Bar Plot

i) Relationship between Total Cases and States.

```
sns.set(rc = {'figure.figsize':(15,10)})
sns.barplot(x="Total Cases", y="State/UTs", data=data)
```

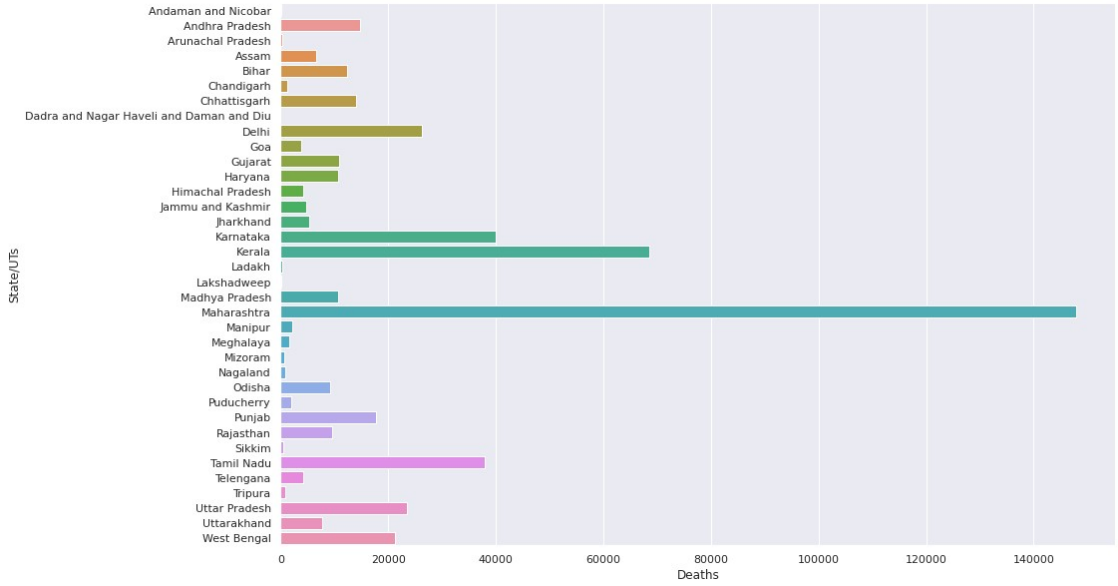
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdab28f8c90>
```



ii) Relationship between Deaths and States.

```
sns.barplot(x="Deaths", y="State/UTs", data=data)
```

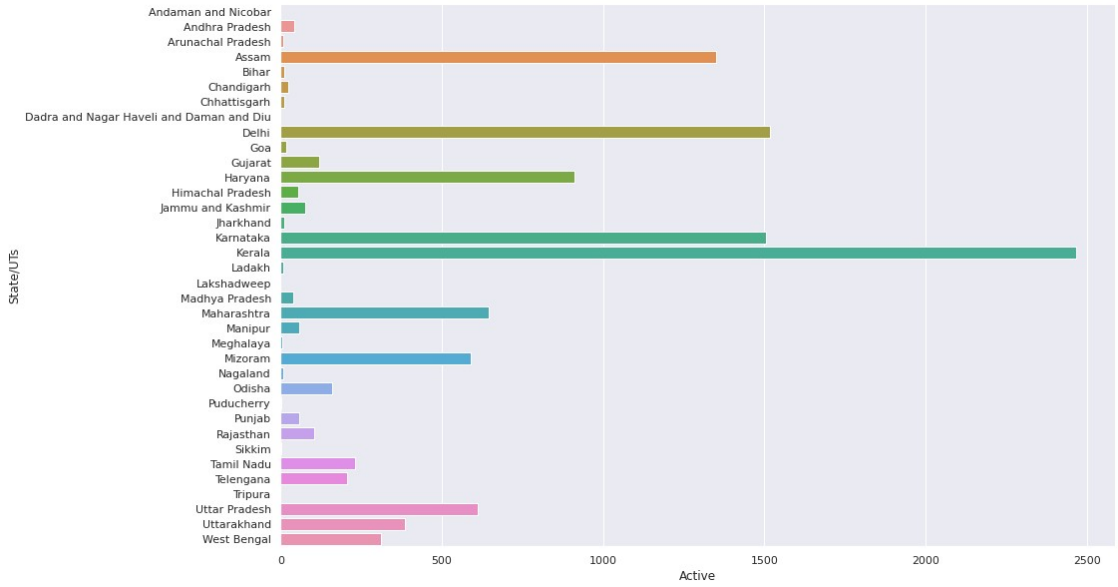
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdab26dcdd0>
```



iii) Relationship between Active and States.

```
sns.barplot(x="Active", y="State/UTs", data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdab24f19d0>
```



4. Histogram

Relationship between Deaths and States.

```
sns.histplot(x="Deaths", y="State/UTs", data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdab24064d0>
```