Informatics Practices

Textbook for Class XII

12149

विद्यया ऽमृतमश्नुते

एन सी ई आर टी
NCERT

राष्ट्रीय शैक्षिक अनुसंधान और प्रशिक्षण परिषद्
NATIONAL COUNCIL OF EDUCATIONAL RESEARCH AND TRAINING

Rationalised 2023-24

12149 – Informatics Practices
Textbook for Class XII

ISBN 978-93-5292-361-8

First Edition
December 2020 Agrahayana 1942

Reprinted
January 2023 Pausha 1944

PD 20T BS

₹ 200.00

Printed on 80 GSM paper

Published at the Publication Division by the Secretary, National Council of Educational Research and Training, Sri Aurobindo Marg, New Delhi 110 016 and printed at Arun Packers & Printers, C-36 Lawrence Road Industrial Area, Delhi 110 035

OFFICES OF THE PUBLICATION DIVISION, NCERT

NCERT Campus
Sri Aurobindo Marg
New Delhi 110 016 **Phone : 011-26562708**

108, 100 Feet Road
Hosdakere Halli Extension
Banashankari III Stage
Bengaluru 560 085 **Phone : 080-26725740**

Navjivan Trust Building
P.O.Navjivan
Ahmedabad 380 014 **Phone : 079-27541446**

CWC Campus
Opp. Dhankal Bus Stop
Panihati
Kolkata 700 114 **Phone : 033-25530454**

CWC Complex
Maligaon
Guwahati 781 021 **Phone : 0361-2674869**

Publication Team

Head, Publication Division	:	*Anup Kumar Rajput*
Chief Production Officer	:	*Arun Chitkara*
Chief Business Manager	:	*Vipin Dewan*
Chief Editor (In charge)	:	*Bijnan Sutar*
Production Officer	:	*A.M. Vinod Kumar*

Cover and Layout
DTP Cell, Publication Division

FOREWORD

Information Technology has continuously been crossing the barriers of access and communication and reaching more and more people. The number of internet users in India has been on the rise. The tremendous growth in computer science, telecommunications and information technology has resulted in automation of various tasks and contributed to the ease of living. Technology has made continuous inroads into diverse areas—be it business, commerce, science, sports, health, transportation or education. Today, we are living in an interconnected world where computer based applications influence the way we learn, communicate, commute, or even socialise.

With so many users of Information and Communication Technology (ICT), huge volumes of data are continuously generated at an unprecedented rate. Many innovative business models are being evolved which utilise such data to reach potential customers in a more targeted way. Government agencies are also using data to deliver services and fast track progress of different programmes, strengthen accountability and to make more informed decisions. This has been creating better opportunities for our youth not only to enter the field of technical education but also in the world of work. NCERT, for the first time, has developed a textbook on 'Informatics Practices' to develop skill sets in students to make use of the opportunities provided by ICT.

This book focuses on the fundamental concepts related to handling of data while opening a window to the emerging areas of data processing. It seeks to address the dual challenges of reducing curricular load as well as introducing the latest development in the field of ICT.

As an organisation committed to systemic reforms and continuous improvement in the quality of its curricular material, NCERT welcomes comments and suggestions to enable us to bring about necessary changes in its further publications.

Hrushikesh Senapaty
Director
National Council of Educational
Research and Training

New Delhi
August 2020

Preface

In the present education system of our country, specialised and discipline based courses are introduced at the higher secondary stage. This stage is crucial as well as challenging because of the transition from general to discipline-based curriculum. The syllabus at this stage needs to have sufficient rigour and depth while remaining mindful of the comprehension level of the learners. Further, the textbook should not be heavily loaded with content.

We are living in an era where information drives many of our socio economic decisions. Millions of people are accessing internet round the clock for availing various services and thereby generating vast amount of data. Processing of data is becoming a key skill with applications across the disciplines. Thus, study of basic concepts of data handling and analysis is becoming more and more desirable. There are courses offered in the name of Computer Science, Information and Communication Technology (ICT), Information Technology (IT), etc. by various boards and schools up to secondary stage, as optional. These mainly focus on using computer for word processing, presentation tools and application software.

Informatics Practices (IP) at the higher secondary stage of school education is also offered as an optional subject. At this stage, students can take up IP with the aim of pursuing a career in data science or related areas after going through professional courses at higher levels. Therefore, at higher secondary stage, the curriculum of IP introduces basics of database management systems and data processing. The book has seven chapters covering the following broader themes:

- **SQL Queries:** Querying database using the Structured Query Language by applying SQL functions including aggregate functions.
- **Data Handling:** The popular Python library called Pandas has been introduced. The important data structures of Pandas – Series and DataFrame have been covered in details and basic data handling and data analysis using Pandas are included.
- **Data Visualisation:** The Pandas library called Pyplot is introduced. It demonstrates how to generate high quality graphs and charts from Python using the Pyplot tool.
- **Internet and Web:** Introduction to the concepts of Computer networks are given, followed by a brief overview of Internet, its application are given. The concept of web, website, and its hosting is also included.
- **Societal Impact:** Awareness of digital footprints, data privacy and protection, cyber crime, etiquettes, copyright and plagiarism, E-waste in a digital society and their implications on security, privacy, piracy, ethics, values and health concerns.

Each chapter has two additional components — (i) activities and (ii) think and reflect for self assessment while learning as well as to generate further interest in the learner. A number of hands-on examples are given to gradually explain methodology to solve different types of problems across the Chapters. The programming examples as well as the exercises in the chapters are required to be solved in a computer and verify with the given outputs.

Box items are pinned inside the chapters either to explain related concepts or to describe additional information related to the topic covered in that section. However, these box-items are not to be assessed through examinations.

Project Based Learning given at the end includes exemplar projects related to real-world problems. Teachers are supposed to assign these or similar projects to be developed in groups. Working in such projects may promote peer-learning, team spirit and responsiveness.

The chapters have been written by involving practicing teachers as well as subject experts. Several iterations have resulted into this book. Thanks are due to the authors and reviewers for their valuable contribution. I would like to place on record appreciation for *Professor* Om Vikas for leading the review activities of the book as well as for his guidance and motivation to the development team throughout. Comments and suggestions are welcome.

New Delhi
31 August 2020

Dr. Rejaul Karim Barbhuiya
Assistant Professor
Central Institute of
Educational Technology

Textbook Development Committee

Members

Anamika Gupta, *Assistant Professor*, Shaheed Sukhdev College of Business Studies, University of Delhi

Anju Gupta, *Freelance Educationist*, Delhi

Anuradha Khattar, *Assistant Professor*, Miranda House, University of Delhi

Chetna Khanna, *Freelance Educationist*, Delhi

Harita Ahuja, *Assistant Professor*, Acharya Narendra Dev College, University of Delhi

Mohini Arora, *HOD (Computer Science)*, Air Force Golden Jubilee Institute, Subroto Park, Delhi

Naeem Ahmad, *Assistant Professor*, Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh

Naveen Gupta, *PGT (Computer Science)*, St. Marks's Sr Sec Public School, Meera Bagh, Delhi

Neeru Mittal, *PGT (Computer Science)*, SRDAV Public School, Dayanand Vihar, Delhi

Priti Rai Jain, *Assistant Professor*, Miranda House, University of Delhi

Sangita Chadha, *HOD (Computer Science)*, Ambience Public School, Safdarjung Enclave, Delhi

Sharanjit Kaur, *Associate Professor*, Acharya Narendra Dev College, University of Delhi

Sugandha Gupta, *Assistant Professor*, Sri Guru Gobind Singh College of Commerce, University of Delhi

Vineeta Garg, *PGT (Computer Science)*, SRDAV Public School, Dayanand Vihar, Delhi

Member-Coordinator

Rejaul Karim Barbhuiya, *Assistant Professor*, Central Institute of Educational Technology, NCERT, Delhi

Acknowledgements

The National Council of Educational Research and Training acknowledges the valuable contributions of the individuals and organisations involved in the development of Informatics Practices textbook for Class XII.

The Council expresses its gratitude to the syllabus development team including MPS Bhatia, *Professor*, Netaji Subhas Institute of Technology, Delhi; T. V. Vijay Kumar, *Professor*, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi; Zahid Raza, *Associate Professor*, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi; Vipul Shah, *Principal Scientist*, Tata Consultancy Services, and the CSpathshala team; Aasim Zafar, *Associate Professor*, Department of Computer Science, Aligarh Muslim University, Aligarh; Faisal Anwer, *Assistant Professor*, Department of Computer Science, Aligarh Muslim University, Aligarh; Smruti Ranjan Sarangi, *Associate Professor*, Department of Computer Science and Engineering, Indian Institute of Technology, Delhi; Vikram Goyal, *Associate Professor*, Indraprastha Institute of Information Technology (IIIT), Delhi; and Mamur Ali, *Assistant Professor*, Department of Teacher Training and Non-formal Education (IASE), Faculty of Education, Jamia Millia Islamia, New Delhi.

The Council is thankful to the following resource persons for providing valuable inputs in developing this book — D.N. Sansanwal, *Retd. Professor*, Devi Ahilya Vishwavidyalaya, Indore; Veer Sain Dixit, *Assistant Professor*, Atma Ram Sanatan Dharma College, University of Delhi; Mukesh Kumar, DPS RK Puram, Delhi; Aswin K. Dash, Mother's International School, Delhi; Purvi Kumar, *Co-ordinator*, Computer Science Department, Ganga International School, Rohtak Road, Delhi; Mudasir Wani, *Assistant Professor*, Govt. College for Women, Nawakadal, Srinagar, Jammu and Kashmir; Sajid Yousuf Bhat, *Assistant Professor*, University of Kashmir, Jammu and Kashmir; *Professor* Om Vikas, *Formerly Director*, ABV-IIITM, Gwalior, MP.

The council is grateful to Sunita Farkya, *Professor* and *Head*, Department of Education in Science and Mathematics, NCERT and Amarendra P. Behera, *Professor and Joint Director*, CIET, NCERT for their valuable cooperation and support throughout the development of this book.

The Council also gracefully acknowledges the contributions of Meetu Sharma, *Graphic Designer cum DTP Operator*; Kanika Walecha, *DTP Operator*; Pooja, *Junior Project Fellow* in shaping this book. The contributions of the office of the APC, DESM and Publication Division, NCERT, New Delhi, in bringing out this book are also duly acknowledged.

The Council also acknowledges the contribution of Ankeeta Bezboruah *Assistant Editor* (Contractual), Publication Division, NCERT for copy editing this book. The efforts of Rajshree Saini, *DTP Operator* (Contractual), Publication Division, NCERT are also acknowledged.

CONTENTS

Chapter 1

Querying and SQL Functions

“Any unique image that you desire probably already exists on the internet or in some database... The problem today is no longer how to create the right image, but how to find an already existing one”

— Lev Manovich

12149CH01

In this chapter

1.1 Introduction

In Class XI, we have understood database concepts and learned how to create databases using MySQL. We have also learnt how to populate, manipulate and retrieve data from a database using SQL queries.

In this chapter, we are going to learn more SQL commands which are required to perform various queries in a database. We will understand how to use single row functions, multiple row functions, arranging records in ascending or descending order, grouping records based on some criteria, and working on multiple tables using SQL.

Let us create a database called CARSHOWROOM, having the schema as shown in Figure 1.1. It has the following four relations:

- **INVENTORY:** Stores name, price, model, year of manufacturing, and fuel type for each car in inventory of the showroom,
- **CUSTOMER:** Stores customer Id, name, address, phone number and email for each customer,
- **SALE:** Stores the invoice number, car Id, customer id, sale date, mode of payment, sales person's employee Id, and selling price of the car sold,
- **EMPLOYEE:** Stores employee Id, name, date of birth, date of joining, designation, and salary of each employee in the showroom.

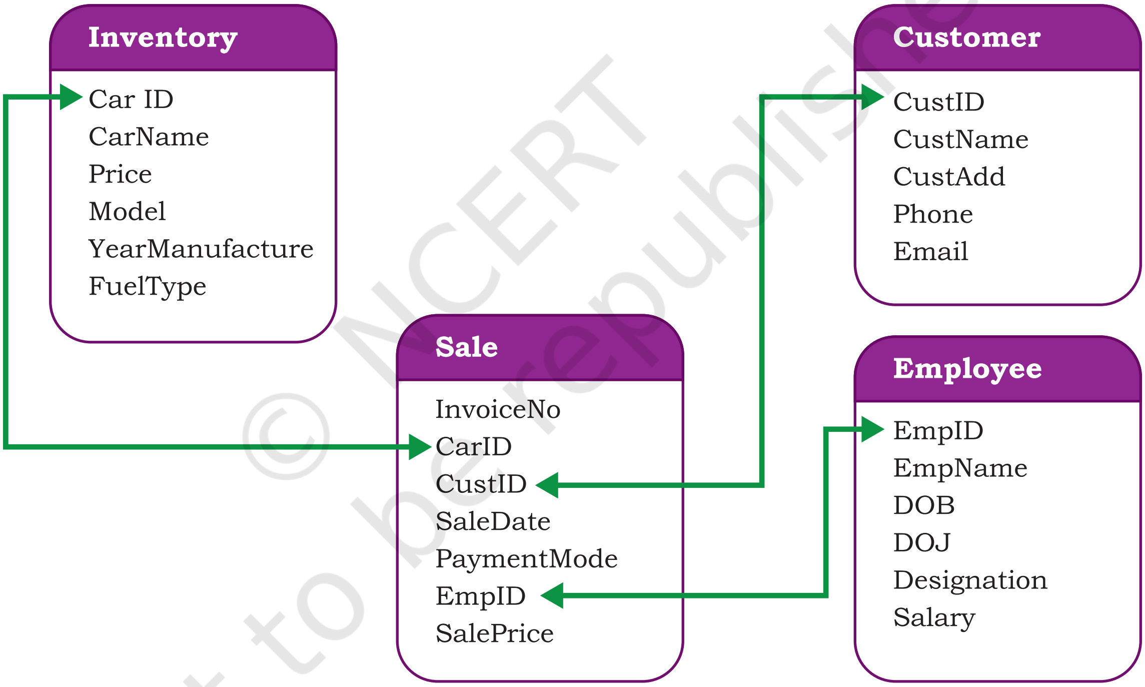


Figure 1.1: Schema diagram of database CARSHOWROOM

The records of the four relations are shown in Tables 1.1, 1.2, 1.3, and 1.4 respectively.

Table 1.1 INVENTORY

```
mysql> SELECT * FROM INVENTORY;
   +-------+--------+-----------+-----------+-----------------+----------+
   | CarId | CarName| Price     | Model     | YearManufacture | Fueltype |
   +-------+--------+-----------+-----------+-----------------+----------+
   | D001  | Car1   | 582613.00 | LXI       |            2017 | Petrol   |
   | D002  | Car1   | 673112.00 | VXI       |            2018 | Petrol   |
   | B001  | Car2   | 567031.00 | Sigma1.2  |            2019 | Petrol   |
   | B002  | Car2   | 647858.00 | Delta1.2  |            2018 | Petrol   |
| E001  | Car3   | 355205.00 | 5 STR STD |              2017 | CNG      |
| E002  | Car3   | 654914.00 | CARE      |              2018 | CNG      |
| S001  | Car4   | 514000.00 | LXI       |              2017 | Petrol   |
| S002  | Car4   | 614000.00 | VXI       |              2018 | Petrol   |
+-------+--------+-----------+-----------+-------------------+----------+
8 rows in set (0.00 sec)
```

Table 1.2 CUSTOMER

```
mysql> SELECT * FROM CUSTOMER;
+-------+-----------+---------------------+-----------+------------------+
|CustId | CustName  | CustAdd             | Phone     | Email            |
+-------+-----------+---------------------+-----------+------------------+
| C0001 |AmitSaha   | L-10, Pitampura     | 4564587852 |amitsaha2@gmail.com|
| C0002 |Rehnuma    | J-12, SAKET         | 5527688761 |rehnuma@hotmail.com|
| C0003 |CharviNayyar| 10/9, FF, Rohini   | 6811635425 |charvi123@yahoo.com|
| C0004 |Gurpreet   | A-10/2, SF, MayurVihar| 3511056125 |gur_singh@yahoo.com|
+-------+-----------+---------------------+-----------+------------------+
4 rows in set (0.00 sec)
```

Table 1.3 SALE

```
mysql> SELECT * FROM SALE;
+-----------+-------+--------+-----------+-------------+------+-----------+
| InvoiceNo | CarId | CustId | SaleDate   | PaymentMode  |EmpID | SalePrice  |
+-----------+-------+--------+-----------+-------------+------+-----------+
| I00001    | D001  | C0001  | 2019-01-24 | Credit Card  | E004 | 613247.00 |
| I00002    | S001  | C0002  | 2018-12-12 | Online       | E001 | 590321.00 |
| I00003    | S002  | C0004  | 2019-01-25 | Cheque       | E010 | 604000.00 |
| I00004    | D002  | C0001  | 2018-10-15 | Bank Finance | E007 | 659982.00 |
| I00005    | E001  | C0003  | 2018-12-20 | Credit Card  | E002 | 369310.00 |
| I00006    | S002  | C0002  | 2019-01-30 | Bank Finance | E007 | 620214.00 |
+-----------+-------+--------+-----------+-------------+------+-----------+
6 rows in set (0.00 sec)
```

Table 1.4 EMPLOYEE

```
mysql> SELECT * FROM EMPLOYEE;
+-------+---------+----------+----------+------------+-------+
| EmpID | EmpName  | DOB        | DOJ        | Designation  | Salary |
+-------+---------+----------+----------+------------+-------+
| E001  |Rushil    | 1994-07-10 | 2017-12-12 | Salesman     |  25550 |
| E002  |Sanjay    | 1990-03-12 | 2016-06-05 | Salesman     |  33100 |
| E003  |Zohar     | 1975-08-30 | 1999-01-08 | Peon         |  20000 |
| E004  |Arpit     | 1989-06-06 | 2010-12-02 | Salesman     |  39100 |
| E006  |Sanjucta  | 1985-11-03 | 2012-07-01 | Receptionist |  27350 |
| E007  |Mayank    | 1993-04-03 | 2017-01-01 | Salesman     |  27352 |
| E010  |Rajkumar  | 1987-02-26 | 2013-10-23 | Salesman     |  31111 |
+-------+---------+----------+----------+------------+-------+
7 rows in set (0.00 sec)
```

1.2 FUNCTIONS IN SQL

We know that a function is used to perform some particular task and it returns zero or more values as a result. Functions are useful while writing SQL queries also. Functions can be applied to work on single or multiple records (rows) of a table. Depending on their application in one or multiple rows, SQL functions are categorised as Single row functions and Aggregate functions.

1.2.1 Single Row Functions

These are also known as Scalar functions. Single row functions are applied on a single value and return a single value. Figure 1.2 lists different single row functions under three categories — Numeric (Math), String, Date and Time.

Math functions accept numeric value as input, and return a numeric value as a result. String functions accept character value as input, and return either character or numeric values as output. Date and time functions accept date and time values as input, and return numeric or string, or date and time values as output.

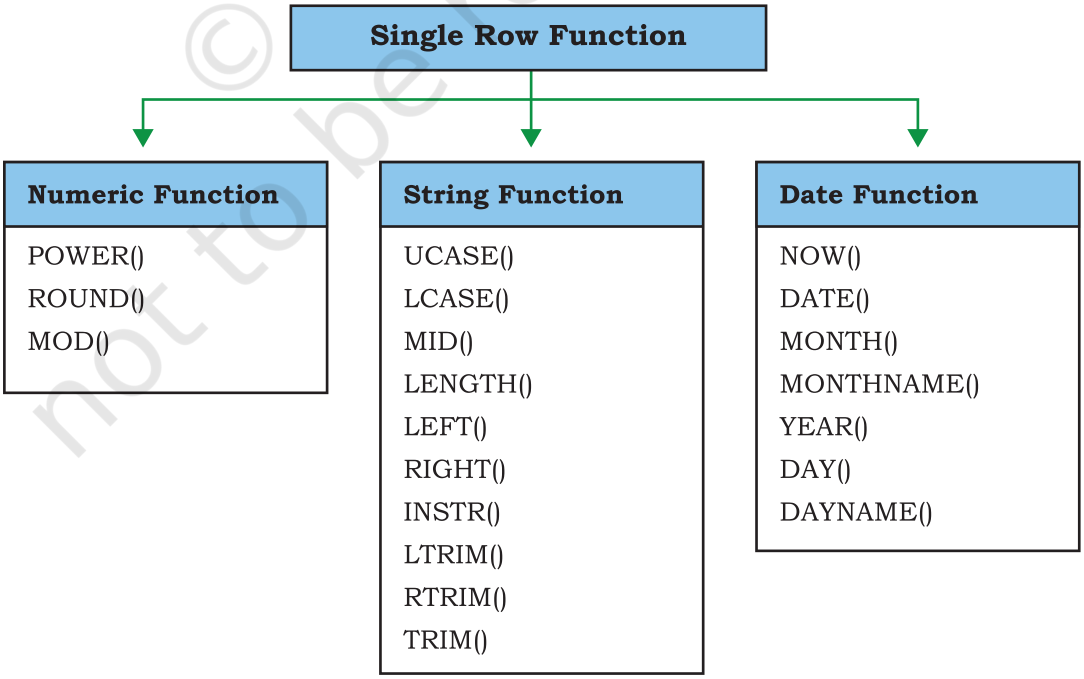


Figure 1.2: Three categories of single row functions in SQL

(A) Numeric Functions

Three commonly used numeric functions are POWER(), ROUND() and MOD(). Their usage along with syntax is given in Table 1.5.

Table 1.5 Math Functions

Function	Description	Example with output

POWER(X,Y)
can also be written as POW(X,Y)
	Calculates X to the power Y.	`mysql> SELECT POWER(2,3);`
		`Output:`
		`8`

ROUND(N,D)	Rounds off number N to D number of decimal places.
	Note: If D=0, then it rounds off the number to the nearest integer.
		`mysql>SELECT ROUND(2912.564, 1);`
		`Output:`
		`2912.6`
		`mysql> SELECT ROUND(283.2);`
		`Output:`
		`283`

MOD(A, B)	Returns the remainder after dividing number A by number B.	`mysql> SELECT MOD(21, 2);`
		`Output:`
		`1`

Example 1.1

In order to increase sales, suppose the car dealer decides to offer his customers to pay the total amount in 10 easy EMIs (equal monthly installments). Assume that EMIs are required to be in multiples of 10,000. For that, the dealer wants to list the CarID and Price along with the following data from the Inventory table:

a) Calculate GST as 12% of Price and display the result after rounding it off to one decimal place.

```
mysql> SELECT ROUND(12/100*Price,1) "GST"
FROM INVENTORY;
+---------+
| GST     |
+---------+
| 69913.6 |
| 80773.4 |
| 68043.7 |
| 77743.0 |
| 42624.6 |
| 78589.7 |
| 61680.0 |
| 73680.0 |
+---------+
 8 rows in set (0.00 sec)
```

b) Add a new column FinalPrice to the table inventory, which will have the value as sum of Price and 12% of the GST.

```
mysql> ALTER TABLE INVENTORY ADD(FinalPrice
Numeric(10,1));
Query OK, 8 rows affected (0.03 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> UPDATE INVENTORY SET
FinalPrice=Price+Round(Price*12/100,1);
Query OK, 8 rows affected (0.01 sec)
Rows matched: 8 Changed: 8  Warnings: 0
```

```
mysql> SELECT * FROM INVENTORY;
+-------+--------+-----------+----------+--------------+----------+-------------+
| CarId |CarName | Price     | Model    |YearManufacture| FuelType |  FinalPric  |
+-------+--------+-----------+----------+--------------+----------+-------------+
| D001  |Car1    | 582613.00 | LXI      |   2017       | Petrol   |   652526.6  |
| D002  |Car1    | 673112.00 | VXI      |   2018       | Petrol   |   753885.4  |
| B001  |Car2    | 567031.00 | Sigma1.2 |   2019       | Petrol   |   635074.7  |
| B002  |Car2    | 647858.00 | Delta1.2 |   2018       | Petrol   |   725601.0  |
| E001  |Car3    | 355205.00 | 5STR STD |   2017       | CNG      |   397829.6  |
| E002  |Car3    | 654914.00 | CARE     |   2018       | CNG      |   733503.7  |
| S001  |Car4    | 514000.00 | LXI      |   2017       | Petrol   |   575680.0  |
| S002  |Car4    | 614000.00 | VXI      |   2018       | Petrol   |   687680.0  |
+-------+--------+-----------+----------+--------------+----------+-------------+
8 rows in set (0.00 sec)
```

c) Calculate and display the amount to be paid each month (in multiples of 1000) which is to be calculated after dividing the FinalPrice of the car into 10 instalments.

d) After dividing the amount into EMIs, find out the remaining amount to be paid immediately, by performing modular division.

Following SQL query can be used to solve the above mentioned problems:

```
mysql> select CarId, FinalPrice, ROUND((FinalPrice-
MOD(FinalPrice,10000))/10,0) "EMI", MOD(FinalPrice,10000) "Remaining Amount"
FROM INVENTORY;
+-------+------------+-------+------------------+
| CarId | FinalPrice | EMI   | Remaining Amount |
+-------+------------+-------+------------------+
| D001  |   652526.6 | 65000 |           2526.6 |
| D002  |   753885.4 | 75000 |           3885.4 |
| B001  |   635074.7 | 63000 |           5074.7 |
| B002  |   725601.0 | 72000 |           5601.0 |
| E001  |   397829.6 | 39000 |           7829.6 |
| E002  |   733503.7 | 73000 |           3503.7 |
| S001  |   575680.0 | 57000 |           5680.0 |
| S002  |   687680.0 | 68000 |           7680.0 |
+-------+------------+-------+------------------+
8 rows in set (0.00 sec)
```

Example 1.2

a) Let us now add a new column Commission to the SALE table. The column Commission should have a total length of 7 in which 2 decimal places to be there.

```
mysql> ALTER TABLE SALE ADD(Commission
Numeric(7,2));
Query OK, 6 rows affected (0.34 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

b) Let us now calculate commission for sales agents as 12 per cent of the SalePrice, insert the values to the newly added column Commission and then display records of the table SALE where commission > 73000.

```
mysql> UPDATE SALE SET
Commission=12/100*SalePrice;
Query OK, 6 rows affected (0.06 sec)
Rows matched: 6 Changed: 6  Warnings: 0
```

```
mysql> SELECT * FROM SALE WHERE Commission > 73000;
+---------------+------+----------+-----------+------+-----------+-----------+
|invoiceno|carid|custid| saledate |paymentmode |empid | saleprice |Commission |
+---------------+------+----------+-----------+------+-----------+-----------+

|I00001   |D001 |C0001 |2019-01-24|Credit Card |E004  | 613247.00 | 73589.64  |
|I0000    |D002 |C0001 |2018-10-15|Bank Finance|E007  | 659982.00 | 79197.84  |
|I00006   |S002 |C0002 |2019-01-30|Bank Finance|E007  | 620214.00 | 74425.68  |
+---------------+------+-----------+-----------+------+----------+-----------+
3 rows in set (0.02 sec)
```

c) Display InvoiceNo, SalePrice and Commission such that commission value is rounded off to 0.

```
mysql> SELECT InvoiceNo, SalePrice,
Round(Commission,0) FROM SALE;
+-----------+-----------+---------------------+
| InvoiceNo | SalePrice | Round(Commission,0) |
+-----------+-----------+---------------------+
| I00001    | 613247.00 |               73590 |
| I00002    | 590321.00 |               70839 |
| I00003    | 604000.00 |               72480 |
| I00004    | 659982.00 |               79198 |
| I00005    | 369310.00 |               44317 |
| I00006    | 620214.00 |               74426 |
+-----------+-----------+---------------------+
```

```
6 rows in set (0.00 sec)
```

(B) String Functions

String functions can perform various operations on alphanumeric data which are stored in a table. They can be used to change the case (uppercase to lowercase

Activity 1.1

Using the table SALE of CARSHOWROOM database, write SQL queries for the following:

a) Display the InvoiceNo and commission value rounded off to zero decimal places.

b) Display the details of SALE where payment mode is credit card..

or vice-versa), extract a substring, calculate the length of a string and so on. String functions and their usage are shown in Table 1.6.

Table 1.6 String Functions

<table>
<tr><th>Function</th><th>Description</th><th>Example with output</th></tr>
<tr><td>UCASE(string)
OR
UPPER(string)</td><td>Converts string into uppercase.</td><td>mysql> SELECT UCASE(“Informatics Practices”);
Output:
INFORMATICS PRACTICES</td></tr>
<tr><td>LOWER(string)
OR
LCASE(string)</td><td>Converts string into lowercase.</td><td>mysql> SELECT LOWER(“Informatics Practices”);
Output:
informatics practices</td></tr>
<tr><td>MID(string, pos, n)
OR
SUBSTRING(string, pos, n)
OR
SUBSTR(string, pos, n)</td><td>Returns a substring of size n starting from the specified position (pos) of the string. If n is not specified, it returns the substring from the position pos till end of the string.</td><td>mysql> SELECT MID(“Informatics”, 3, 4);
Output:
form
mysql> SELECT MID(‘Informatics’,7);
Output:
atics</td></tr>
<tr><td>LENGTH(string)</td><td>Return the number of characters in the specified string.</td><td>mysql> SELECT LENGTH(“Informatics”);
Output:
11</td></tr>
<tr><td>LEFT(string, N)</td><td>Returns N number of characters from the left side of the string.</td><td>mysql> SELECT LEFT(“Computer”, 4);
Output:
Comp</td></tr>
<tr><td>RIGHT(string, N)</td><td>Returns N number of characters from the right side of the string.</td><td>mysql> SELECT RIGHT(“SCIENCE”, 3);
Output:
NCE</td></tr>
<tr><td>INSTR(string, substring)</td><td>Returns the position of the first occurrence of the substring in the given string. Returns 0, if the substring is not present in the string.</td><td>mysql> SELECT INSTR(“Informatics”, “ma”);
Output:
6</td></tr>
<tr><td>LTRIM(string)</td><td>Returns the given string after removing leading white space characters.</td><td>mysql> SELECT LENGTH(“ DELHI”), LENGTH(LTRIM(“ DELHI”));
Output:
+--------+--------+
| 7 | 5 |
+--------+--------+
1 row in set (0.00 sec)</td></tr>
</table>

<table>
<tr><td>RTRIM(string)</td><td>Returns the given string after removing trailing white space characters.</td><td>mysql&gt;SELECT LENGTH(“PEN “)
LENGTH(RTRIM(“PEN “));
Output:
+--------+--------+
| 5 | 3 |
+--------+--------+
1 row in set (0.00 sec)</td></tr>
<tr><td>TRIM(string)</td><td>Returns the given string after removing both leading and trailing white space characters.</td><td>mysql&gt; SELECT LENGTH(“ MADAM “),LENGTH(TRIM(“ MADAM “));
Output:
+--------+--------+
| 9 | 5 |
+--------+--------+
1 row in set (0.00 sec)</td></tr>
</table>

Example 1.3

Let us use CUSTOMER relation shown in Table 1.2 to understand the working of string functions.

a) Display customer name in lower case and customer email in upper case from table CUSTOMER.

```
mysql> SELECT LOWER(CustName), UPPER(Email) FROM
CUSTOMER;
+-----------------+--------------------+
| LOWER(CustName) | UPPER(Email)       |
+-----------------+--------------------+
| amitsaha        | AMITSAHA2@GMAIL.COM |
| rehnuma         | REHNUMA@HOTMAIL.COM |
| charvinayyar    | CHARVI123@YAHOO.COM |
| gurpreet        | GUR_SINGH@YAHOO.COM |
+-----------------+--------------------+
4 rows in set (0.00 sec)
```

b) Display the length of the email and part of the email from the email ID before the character '@'. Note - Do not print '@'.

```
mysql> SELECT LENGTH(Email), LEFT(Email, INSTR(Email,
"@")-1) FROM CUSTOMER;
+--------------+--------------------------------+
| LENGTH(Email) | LEFT(Email, INSTR(Email, "@")-1) |
+--------------+--------------------------------+
|            19 | amitsaha2                        |
|            19 | rehnuma                          |
|            19 | charvi123                        |
|            19 | gur_singh                        |
+--------------+--------------------------------+
4 rows in set (0.03 sec)
```

The function INSTR will return the position of “@” in the email address. So to print email id without “@” we have to use position -1.

Activity 1.2

Using the table INVENTORY from CARSHOWROOM database, write sql queries for the following:

a) Convert the CarMake to uppercase if its value starts with the letter 'B'.

b) If the length of the car's model is greater than 4 then fetch the substring starting from position 3 till the end from attribute Model.

Activity 1.3

Using the table EMPLOYEE from CARSHOWROOM database, write SQL queries for the following:

a) Display employee name and the last 2 characters of his EmpId.

b) Display designation of employee and the position of character 'e' in designation, if present.

c) Let us assume that four digit area code is reflected in the mobile number starting from position number 3. For example, 1851 is the area code of mobile number 9818511338. Now, write the SQL query to display the area code of the customer living in Rohini.

```
mysql> SELECT MID(Phone,3,4) FROM CUSTOMER WHERE
CustAdd like ‘%Rohini%’;
+----------------+
| MID(Phone,3,4) |
+----------------+
| 1163           |
+----------------+
1 row in set (0.00 sec)
```

d) Display emails after removing the domain name extension “.com” from emails of the customers.

```
mysql> SELECT TRIM(“.com” from Email) FROM
CUSTOMER;
+-----------------------+
| TRIM(".com" FROM Email) |
+-----------------------+
| amitsaha2@gmail        |
| rehnuma@hotmail        |
| charvi123@yahoo        |
| gur_singh@yahoo        |
+-----------------------+
4 rows in set (0.00 sec)
```

e) Display details of all the customers having yahoo emails only.

```
mysql> SELECT * FROM CUSTOMER WHERE Email LIKE
"%yahoo%";
+-------+------------+---------------------+----------+-------------------+
|CustID | CustName   | CustAdd             | Phone    | Email             |
+-------+------------+---------------------+----------+-------------------+
|C0003  |CharviNayyar |10/9, FF, Rohini     |6811635425 |charvi123@yahoo.com |
|C0004  |Gurpreet    | A-10/2,SF, MayurVihar|3511056125 | gur_singh@yahoo.com|
+-------+------------+---------------------+----------+-------------------+
2 rows in set (0.00 sec)t
```

(C) Date and Time Functions

There are various functions that are used to perform operations on date and time data. Some of the operations include displaying the current date, extracting each element of a date (day, month and year), displaying day of the week and so on. Table 1.7 explains various date and time functions.

Table 1.7 Date Functions

Function	Description	Example with output

NOW()	It returns the current system date and time.	`mysql> SELECT NOW();`
		`Output:`
		`2019-07-11 19:41:17`

DATE()	It returns the date part from the given date/ time expression.	`mysql> SELECT DATE(NOW());`
		`Output:`
		`2019-07-11`

MONTH(date)	It returns the month in numeric form from the date.	`mysql> SELECT MONTH(NOW());`
		`Output:`
		`7`

MONTHNAME(date)	It returns the month name from the specified date.	`mysql> SELECT MONTHNAME(“2003-11-28”);`
		`Output:`
		`November`

YEAR(date)	It returns the year from the date.	`mysql> SELECT YEAR(“2003-10-03”);`
		`Output:`
		`2003`

DAY(date)	It returns the day part from the date.	`mysql> SELECT DAY(“2003-03-24”);`
		`Output:`
		`24`

DAYNAME(date)	It returns the name of the day from the date.	`mysql> SELECT DAYNAME(“2019-07-11”);`
		`Output:`
		`Thursday`

Example 1.4

Let us use the EMPLOYEE table of CARSHOWROOM database to illustrate the working of some of the date and time functions.

a) Select the day, month number and year of joining of all employees.

```
mysql> SELECT DAY(DOJ), MONTH(DOJ), YEAR(DOJ) FROM
EMPLOYEE;
+----------+------------+-----------+
| DAY(DOJ) | MONTH(DOJ) | YEAR(DOJ) |
+----------+------------+-----------+
|       12 |         12 |      2017 |
|        5 |          6 |      2016 |
|        8 |          1 |      1999 |
|        2 |         12 |      2010 |
|        1 |          7 |      2012 |
|        1 |          1 |      2017 |
|       23 |         10 |      2013 |
+----------+------------+-----------+
7 rows in set (0.03 sec)
```

b) If the date of joining is not a Sunday, then display it in the following format "Wednesday, 26, November, 1979."

Activity 1.4

Using the table EMPLOYEE of CARSHOWROOM database, list the day of birth for all employees whose salary is more than 25000.

Think and Reflect

Can we use arithmetic operators (+, -. *, or /) on date functions?

```
mysql> SELECT DAYNAME(DOJ), DAY(DOJ),
MONTHNAME(DOJ), YEAR(DOJ) FROM EMPLOYEE WHERE
DAYNAME(DOJ)!='Sunday';
+------------+---------+--------------+---------+
|DAYNAME(DOJ)| DAY(DOJ)|MONTHNAME(DOJ) |YEAR(DOJ)|
+------------+---------+--------------+---------+
|Tuesday     |      12 | December     |    2017 |
|Friday      |       8 | January      |    1999 |
|Thursday    |       2 | December     |    2010 |
|Wednesday   |      23 | October           2013 |
+------------+---------+--------------+---------+
4 rows in set (0.00 sec)
```

1.2.2 Aggregate Functions

Aggregate functions are also called multiple row functions. These functions work on a set of records as a whole, and return a single value for each column of the records on which the function is applied. Table 1.8 shows the differences between single row functions and multiple row functions. Table 1.9 describes some of the aggregate functions along with their usage. Note that column must be of numeric type.

Table 1.8 Differences between Single row and Multiple row Functions

Single_row Functions	Multiple_row functions
1. It operates on a single row at a time.	1. It operates on groups of rows.
2. It returns one result per row.	2. It returns one result for a group of rows.
3. It can be used in Select, Where, and Order by clause.	3. It can be used in the select clause only.
4. Math, String and Date functions are examples of single row functions.	4. Max(), Min(), Avg(), Sum(), Count() and Count(*) are examples of multiple row functions.

Table 1.9 Aggregate Functions in SQL

Function	Description	Example with output

MAX(column)	Returns the largest value from the specified column.	`mysql> SELECT MAX(Price) FROM INVENTORY;`
		`Output:`
		`673112.00`

MIN(column)	Returns the smallest value from the specified column.	`mysql> SELECT MIN(Price) FROM INVENTORY;`
		`Output:`
		`355205.00`

AVG(column)	Returns the average of the values in the specified column.	`mysql> SELECT AVG(Price) FROM INVENTORY;`
		`Output:`
		`576091.625000`

SUM(column)	Returns the sum of the values for the specified column.	mysql> SELECT SUM(Price) FROM INVENTORY; Output: 4608733.00
COUNT(column)	Returns the number of values in the specified column ignoring the NULL values. Note: In this example, let us consider a MANAGER table having two attributes and four records.	mysql> SELECT * from MANAGER; Output: +------+---------+ | MNO | MEMNAME | +------+---------+ | 1 | AMIT | | 2 | KAVREET | | 3 | KAVITA | | 4 | NULL | +------+---------+ 4 rows in set (0.00 sec) mysql> SELECT COUNT(MEMNAME) FROM MANAGER; Output: +----------------+ | COUNT(MEMNAME) | +----------------+ | 3 | +----------------+ 1 row in set (0.01 sec)
COUNT(*)	Returns the number of records in a table. Note: In order to display the number of records that matches a particular criteria in the table, we have to use COUNT(*) with WHERE clause.	mysql> SELECT COUNT(*) from MANAGER; Output: +----------+ | count(*) | +----------+ | 4 | +----------+ 1 row in set (0.00 sec)

Example 1.5

a) Display the total number of records from table INVENTORY having a model as VXI.

```
mysql> SELECT COUNT(*) FROM INVENTORY WHERE
Model=”VXI”;
+----------+
| COUNT(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)
```

b) Display the total number of different types of Models available from table INVENTORY.

Activity 1.5

a) Find sum of Sale Price of the cars purchased by the customer having ID C0001 from table SALE.

b) Find the maximum and minimum commission from the SALE table.

```
mysql> SELECT COUNT(DISTINCT Model) FROM
INVENTORY;
+-----------------------+
| COUNT(DISTINCT MODEL) |
+-----------------------+
|                     6 |
+-----------------------+
1 row in set (0.09 sec)
```

c) Display the average price of all the cars with Model LXI from table INVENTORY.

```
mysql> SELECT AVG(Price) FROM INVENTORY WHERE
Model="LXI";
+---------------+
| AVG(Price)    |
+---------------+
| 548306.500000 |
+---------------+
1 row in set (0.03 sec)
```

1.3 GROUP BY IN SQL

At times we need to fetch a group of rows on the basis of common values in a column. This can be done using a GROUP BY clause. It groups the rows together that contain the same values in a specified column. We can use the aggregate functions (COUNT, MAX, MIN, AVG and SUM) to work on the grouped values. HAVING Clause in SQL is used to specify conditions on the rows with GROUP BY clause. Consider the SALE table from the CARSHOWROOM database:

```
mysql> SELECT * FROM SALE;
+-----------+------+-------+------------+-----------------+----
--+------------+------------+
|InvoiceNo|CarId|CustId| SaleDate | PaymentMode  |EmpID| SalePrice|Commission|
+-----------+------+-------+------------+-----------------+------+-----------+-----------+
|I00001   |D001 |C0001 |2019-01-24| Credit Card  | E004| 613247.00| 73589.64 |
|I00002   |S001 |C0002 |2018-12-12| Online       | E001| 590321.00| 70838.52 |
|I00003   |S002 |C0004 |2019-01-25| Cheque       | E010| 604000.00| 72480.00 |
|I00004   |D002 |C0001 |2018-10-15| Bank Finance | E007| 659982.00| 79197.84 |
|I00005   |E001 |C0003 |2018-12-20| Credit Card  | E002| 369310.00| 44317.20 |
|I00006   |S002 |C0002 |2019-01-30| Bank Finance | E007| 620214.00| 74425.68 |
+-----------+------+-------+------------+-----------------+------+-----------+-----------+
6 rows in set (0.11 sec)
```

CarID, CustID, SaleDate, PaymentMode, EmpID, SalePrice are the columns that can have rows with the same values in it. So, GROUP BY clause can be used in these columns to find the number of records of a particular type (column), or to calculate the sum of the price of each car type.

Example 1.6

a) Display the number of cars purchased by each customer from the SALE table.

```
mysql> SELECT CustID, COUNT(*) "Number of Cars"
FROM SALE GROUP BY CustID;
+--------+----------------+
| CustID | Number of Cars |
+--------+----------------+
| C0001  |              2 |
| C0002  |              2 |
| C0003  |              1 |
| C0004  |              1 |
+--------+----------------+
4 rows in set (0.00 sec)
```

b) Display the customer Id and number of cars purchased if the customer purchased more than 1 car from SALE table.

```
mysql> SELECT CustID, COUNT(*) FROM SALE GROUP BY
CustID HAVING Count(*)>1;
+--------+----------+
| CustID | COUNT(*) |
+--------+----------+
| C0001  |        2 |
| C0002  |        2 |
+--------+----------+
2 rows in set (0.30 sec)
```

c) Display the number of people in each category of payment mode from the table SALE.

```
mysql> SELECT PaymentMode, COUNT(PaymentMode) FROM
SALE GROUP BY Paymentmode ORDER BY Paymentmode;
+--------------+--------------------+
| PaymentMode  | Count(PaymentMode) |
+--------------+--------------------+
| Bank Finance |                  2 |
| Cheque       |                  1 |
| Credit Card  |                  2 |
| Online       |                  1 |
+--------------+--------------------+
4 rows in set (0.00 sec)
```

d) Display the PaymentMode and number of payments made using that mode more than once.

```
mysql> SELECT PaymentMode, Count(PaymentMode) FROM
SALE GROUP BY Paymentmode HAVING COUNT(*)>1 ORDER
```

Activity 1.6

a) List the total number of cars sold by each employee.

b) List the maximum sale made by each employee.

NOTES

```
BY Paymentmode;
+--------------+--------------------+
| PaymentMode  | Count(PaymentMode) |
+--------------+--------------------+
| Bank Finance |                  2 |
| Credit Card  |                  2 |
+--------------+--------------------+
2 rows in set (0.00 sec)
```

1.4 Operations on Relations

We can perform certain operations on relations like Union, Intersection, and Set Difference to merge the tuples of two tables. These three operations are binary operations as they work upon two tables. Note here, that these operations can only be applied if both the relations have the same number of attributes, and corresponding attributes in both tables have the same domain.

1.4.1 UNION (U)

This operation is used to combine the selected rows of two tables at a time. If some rows are the same in both the tables, then the result of the Union operation will show those rows only once. Figure 1.3 shows union of two sets.

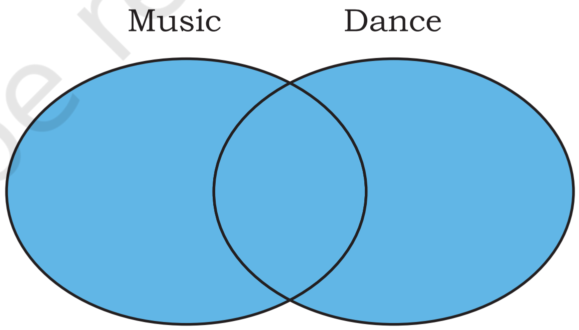


Figure 1.3: Union of two sets

Let us consider two relations DANCE and MUSIC shown in Tables 1.10 and 1.11 respectively.

Table 1.10 DANCE

```
+------+--------+-------+
| SNo  | Name   | Class |
+------+--------+-------+
|     1| Aastha | 7A    |
|     2| Mahira | 6A    |
|     3| Mohit  | 7B    |
|     4| Sanjay | 7A    |
+------+--------+-------+
```

Table 1.11 MUSIC

```
+------+---------+-------+
| SNo  | Name    | Class |
+------+---------+-------+
|     1| Mehak   | 8A    |
|     2| Mahira  | 6A    |
|     3| Lavanya | 7A    |
|     4| Sanjay  | 7A    |
|     5| Abhay   | 8A    |
+------+---------+-------+
```

If we need the list of students participating in either of events, then we have to apply UNION operation (represented by symbol U) on relations DANCE and MUSIC. The output of UNION operation is shown in Table 1.12.

Table 1.12 DANCE ∪ MUSIC

```
+-------+------+------+
|SNo | Name    |Class |
+-------+------+------+
| 1  | Aastha  | 7A   |
| 2  | Mahira  | 6A   |
| 3  | Mohit   | 7B   |
| 4  | Sanjay  | 7A   |
| 1  | Mehak   | 8A   |
| 3  | Lavanya | 7A   |
| 5  | Abhay   | 8A   |
+-------+------+------+
```

1.4.2 INTERSECT (∩)

Intersect operation is used to get the common tuples from two tables and is represented by the symbol ∩. Figure 1.4 shows intersection of two sets.

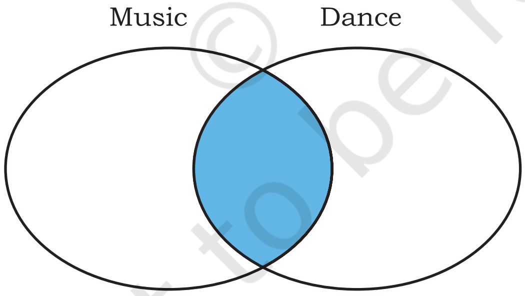


Figure 1.4: Intersection of two sets

Suppose we have to display the list of students who are participating in both the events (DANCE and MUSIC), then intersection operation is to be applied on these two tables. The output of INTERSECT operation is shown in Table 1.13.

Table 1.13 DANCE ∩ MUSIC

```
+------+---------+-------+
| SNo  | Name    | Class |
+------+---------+-------+
|     2| Mahira  | 6A    |
|     4| Sanjay  | 7A    |
+------+---------+-------+
```

Notes

Notes

1.4.3 MINUS (-)

This operation is used to get tuples/rows which are in the first table but not in the second table, and the operation is represented by the symbol - (minus). Figure 1.5 shows minus operation (also called set difference) between two sets.

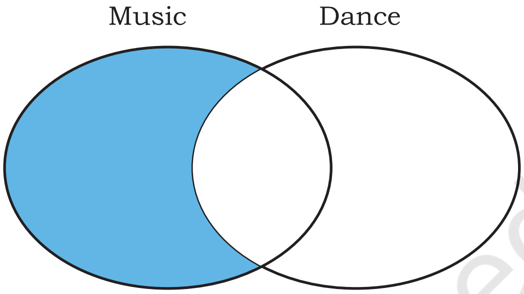


Figure 1.5: Difference of two sets

Suppose, we want the list of students who are only participating in MUSIC and not in DANCE event. Then, we will use the MINUS operation, whose output is given in Table 1.14.

Table 1.14 DANCE - MUSIC

```
+------+---------+-------+
| SNo  | Name    | Class |
+------+---------+-------+
|     1| Mehak   | 8A    |
|     3| Lavanya | 7A    |
|     5| Abhay   | 8A    |
+------+---------+-------+
```

1.4.4 Cartesian Product

Cartesian product operation combines tuples from two relations. It results in all pairs of rows from the two input relations, regardless of whether or not they have the same values on common attributes. It is denoted as 'X'.

The degree of the resulting relation is calculated as the sum of the degrees of both the relations under consideration. The cardinality of the resulting relation is calculated as the product of the cardinality of relations on which cartesian product is applied. Let us use the relations DANCE and MUSIC to show the output of cartesian product. Note that both relations are of degree 3. The cardinality of relations DANCE and MUSIC is 4 and 5 respectively. Applying cartesian product on these two relations will result in a relation of degree 6 and cardinality 20, as shown in Table 1.15.

NOTES

Table 1.15 DANCE X MUSIC

```
+------+--------+-------+------+---------+-------+
| SNo  | Name   | Class| SNo   | Name    | Class|
+------+--------+-------+------+---------+-------+
|    1 | Aastha | 7A    |    1 | Mehak   | 8A    |
|    2 | Mahira | 6A    |    1 | Mehak   | 8A    |
|    3 | Mohit  | 7B    |    1 | Mehak   | 8A    |
|    4 | Sanjay | 7A    |    1 | Mehak   | 8A    |
|    1 | Aastha | 7A    |    2 | Mahira  | 6A    |
|    2 | Mahira | 6A    |    2 | Mahira  | 6A    |
|    3 | Mohit  | 7B    |    2 | Mahira  | 6A    |
|    4 | Sanjay | 7A    |    2 | Mahira  | 6A    |
|    1 | Aastha | 7A    |    3 | Lavanya | 7A    |
|    2 | Mahira | 6A    |    3 | Lavanya | 7A    |
|    3 | Mohit  | 7B    |    3 | Lavanya | 7A    |
|    4 | Sanjay | 7A    |    3 | Lavanya | 7A    |
|    1 | Aastha | 7A    |    4 | Sanjay  | 7A    |
|    2 | Mahira | 6A    |    4 | Sanjay  | 7A    |
|    3 | Mohit  | 7B    |    4 | Sanjay  | 7A    |
|    4 | Sanjay | 7A    |    4 | Sanjay  | 7A    |
|    1 | Aastha | 7A    |    5 | Abhay   | 8A    |
|    2 | Mahira | 6A    |    5 | Abhay   | 8A    |
|    3 | Mohit  | 7B    |    5 | Abhay   | 8A    |
|    4 | Sanjay | 7A    |    5 | Abhay   | 8A    |
+------+--------+-------+------+---------+-------+
20 rows in set (0.03 sec)
```

1.5 USING TWO RELATIONS IN A QUERY

Till now, we have written queries in SQL using a single relation only. In this section, we will learn to write queries using two relations.

1.5.1 Cartesian product on two tables

From the previous section, we learnt that application of operator cartesian product on two tables results in a table having all combinations of tuples from the underlying tables. When more than one table is to be used in a query, then we must specify the table names by separating commas in the FROM clause, as shown in Example 1.7. On execution of such a query, the DBMS (MySql) will first apply cartesian product on specified tables to have a single table. The following query of Example 1.7 applies cartesian product on the two tables DANCE and MUSIC:

Example 1.7

a) Display all possible combinations of tuples of relations DANCE and MUSIC

```
mysql> SELECT * FROM DANCE, MUSIC;
```

As we are using SELECT * in the query, the output will be the Table 1.15 having degree 6 and cardinality 20.

NOTES

b) From the all possible combinations of tuples of relations DANCE and MUSIC, display only those rows such that the attribute name in both have the same value.

```
mysql> SELECT * FROM DANCE D, MUSIC M WHERE D.Name
= M.Name;
```

Table 1.16 Tuples with same name

```
+------+--------+-------+------+--------+-------+
| Sno  | Name   | Class | Sno  | Name   | class |
+------+--------+-------+------+--------+-------+
|    2 | Mahira | 6A    |    2 | Mahira | 6A    |
|    4 | Sanjay | 7A    |    4 | Sanjay | 7A    |
+------+--------+-------+------+--------+-------+
2 rows in set (0.00 sec)
```

Note that in this query we have used table aliases (D for DANCE and M for MUSIC), just like column aliases to refer to tables by shortened names. It is important to note that table alias is valid only for current query and the original table name cannot be used in the query if its alias is given in FROM clause.

1.5.2 JOIN on two tables

JOIN operation combines tuples from two tables on specified conditions. This is unlike cartesian product, which make all possible combinations of tuples. While using the JOIN clause of SQL, we specify conditions on the related attributes of two tables within the FROM clause. Usually, such an attribute is the primary key in one table and foreign key in another table. Let us create two tables UNIFORM (UCode, UName, UColor) and COST (UCode, Size, Price) in the SchoolUniform database. UCode is Primary Key in table UNIFORM. UCode and Size is the Composite Key in table COST. Therefore, Ucode is a common attribute between the two tables which can be used to fetch the common data from both the tables. Hence, we need to define Ucode as foreign key in the Price table while creating this table.

Table 1.17 Uniform table

```
+-------+-------+--------+
| Ucode | Uname | Ucolor |
+-------+-------+--------+
|  1    | Shirt | White  |
|  2    | Pant  | Grey   |
|  3    | Tie   | Blue   |
+-------+-------+--------+
```

Notes

Table 1.18 Cost table

```
+-----+------+-------+
|Ucode| Size | Price |
+-----+------+-------+
| 1   | L    |   580 |
| 1   | M    |   500 |
| 2   | L    |   890 |
| 2   | M    |   810 |
+-------+----+-------+
```

Example 1.7

List the UCode, UName, UColor, Size and Price of related tuples of tables UNIFORM and COST.

The given query may be written in three different ways as given below:

a) Using condition in where clause

```
mysql> SELECT * FROM UNIFORM U, COST C WHERE
U.UCode = C.UCode;
```

Table 1.19 Output of the query

```
+-------+-------+--------+-------+---------+-------+
| UCode | UName | UColor | Ucode | Size    | Price |
+-------+-------+--------+-------+---------+-------+
|   1   | Shirt | White  | 1     | L       |  580  |
|   1   | Shirt | White  | 1     | M       |  500  |
|   2   | Pant  | Grey   | 2     | L       |  890  |
|   2   | Pant  | Grey   | 2     | M       |  810  |
+-------+-------+--------+-------+---------+-------+
4 rows in set (0.08 sec)
```

As the attribute Ucode is in both tables, we need to use table alias to remove ambiguity. Hence, we have used qualifier with attribute UCode in SELECT and FROM clauses to indicate its scope.

b) Explicit use of JOIN clause

```
mysql> SELECT * FROM UNIFORM U JOIN COST C ON
U.Ucode=C.Ucode;
```

The output of the query is the same as shown in Table 1.19. In this query, we have used JOIN clause explicitly along with condition in FROM clause. Hence, no condition needs to be given in WHERE clause.

c) Explicit use of NATURAL JOIN clause

The output of queries (a) and (b) shown in Table 1.19 has a repetitive column Ucode having exactly the same values. This redundant column provides no additional information. There is an extension of JOIN operation called NATURAL JOIN which works similar to JOIN clause in SQL, but removes the redundant attribute. This operator can be used to join the contents of two tables iff there is one common attribute in both the tables. The above SQL query using NATURAL JOIN is shown below:

```
mysql> SELECT * FROM UNIFORM  NATURAL JOIN COST;
+-------+-------+--------+------+-------+
| UCode | UName | UColor | Size | Price |
+-------+-------+--------+------+-------+
| 1     | Shirt | White  | L    |   580 |
| 1     | Shirt | White  | M    |   500 |
| 2     | Pant  | Grey   | L    |   890 |
| 2     | Pant  | Grey   | M    |   810 |
+-------+-------+--------+------+-------+
4 rows in set (0.17 sec)
```

It is clear from the output that the result of this query is same as that of queries written in (a) and (b), except that the attribute Ucode appears only once.

Following are some of the points to be considered while applying JOIN operations on two or more relations:

- If two tables are to be joined on equality condition on the common attribute, then one may use JOIN with ON clause or NATURAL JOIN in FROM clause. If three tables are to be joined on equality condition, then two JOIN or NATURAL JOIN are required.
- In general, N-1 joins are needed to combine N tables on equality condition.
- With JOIN clause, we may use any relational operators to combine tuples of two tables.

Summary

- A Function is used to perform a particular task and return a value as a result.
- Single row functions work on a single row to return a single value.
- Multiple row functions work on a set of records as a whole and return a single value.
- Numeric functions perform operations on numeric values and return numeric values.
- String functions perform operations on character type values and return either character or numeric values.

- Date and time functions allow us to deal with date type data values.
- GROUP BY function is used to group the rows together that contain similar values in a specified column. Some of the group functions are COUNT, MAX, MIN, AVG and SUM.
- Join is an operation which is used to combine rows from two or more tables based on one or more common fields between them.

Exercise

1. Answer the following questions:
   a) Define RDBMS. Name any two RDBMS software.
   b) What is the purpose of the following clauses in a select statement?
      i) ORDER BY
      ii) HAVING
   c) Site any two differences between Single_row functions and Aggregate functions.
   d) What do you understand by Cartesian Product?
   e) Write the name of the functions to perform the following operations:
      i) To display the day like "Monday", "Tuesday", from the date when India got independence.
      ii) To display the specified number of characters from a particular position of the given string.
      iii) To display the name of the month in which you were born.
      iv) To display your name in capital letters.
2. Write the output produced by the following SQL commands:
   a) SELECT POW(2,3);
   b) SELECT ROUND(123.2345, 2), ROUND(342.9234,-1);
   c) SELECT LENGTH("Informatics Practices");
   d) SELECT YEAR("1979/11/26"), MONTH("1979/11/26"), DAY("1979/11/26"),

Notes

MONTHNAME("1979/11/26");

e) SELECT LEFT("INDIA",3), RIGHT("Computer Science",4);

f) SELECT MID("Informatics",3,4), SUBSTR("Practices",3);

3. Consider the following table named "Product", showing details of products being sold in a grocery shop.

PCode	PName	UPrice	Manufacturer
P01	Washing Powder	120	Surf
P02	Tooth Paste	54	Colgate
P03	Soap	25	Lux
P04	Tooth Paste	65	Pepsodant
P05	Soap	38	Dove
P06	Shampoo	245	Dove

a) Write SQL queries for the following:

i. Create the table Product with appropriate data types and constraints.

ii. Identify the primary key in Product.

iii. List the Product Code, Product name and price in descending order of their product name. If PName is the same then display the data in ascending order of price.

iv. Add a new column Discount to the table Product.

v. Calculate the value of the discount in the table Product as 10 per cent of the UPrice for all those products where the UPrice is more than 100, otherwise the discount will be 0.

vi. Increase the price by 12 per cent for all the products manufactured by Dove.

vii. Display the total number of products manufactured by each manufacturer.

b) Write the output(s) produced by executing the following queries on the basis of the information given above in the table Product:

i. SELECT PName, Average(UPrice) FROM Product GROUP BY Pname;

ii. SELECT DISTINCT Manufacturer FROM Product;

Notes

iii. SELECT COUNT(DISTINCT PName) FROM Product;

iv. SELECT PName, MAX(UPrice), MIN(UPrice)
FROM Product GROUP BY PName;

4. Using the CARSHOWROOM database given in the chapter, write the SQL queries for the following:

   a) Add a new column Discount in the INVENTORY table.

   b) Set appropriate discount values for all cars keeping in mind the following:

      (i) No discount is available on the LXI model.

      (ii) VXI model gives a 10% discount.

      (iii) A 12% discount is given on cars other than LXI model and VXI model.

   c) Display the name of the costliest car with fuel type "Petrol".

   d) Calculate the average discount and total discount available on Car4.

   e) List the total number of cars having no discount.

5. Consider the following tables Student and Stream in the Streams_of_Students database. The primary key of the Stream table is StCode (stream code) which is the foreign key in the Student table. The primary key of the Student table is AdmNo (admission number).

AdmNo	Name	StCode
211	Jay	NULL
241	Aditya	S03
290	Diksha	S01
333	Jasqueen	S02
356	Vedika	S01
380	Ashpreet	S03

StCode	Stream
S01	Science
S02	Commerce
S03	Humanities

Write SQL queries for the following:

a) Create the database Streams_Of_Students.

Notes

b) Create the table Student by choosing appropriate data types based on the data given in the table.

c) Identify the Primary keys from tables Student and Stream. Also, identify the foreign key from the table Stream.

d) Jay has now changed his stream to Humanities. Write an appropriate SQL query to reflect this change.

e) Display the names of students whose names end with the character 'a'. Also, arrange the students in alphabetical order.

f) Display the names of students enrolled in Science and Humanities stream, ordered by student name in alphabetical order, then by admission number in ascending order (for duplicating names).

g) List the number of students in each stream having more than 1 student.

h) Display the names of students enrolled in different streams, where students are arranged in descending order of admission number.

i) Show the Cartesian product on the Student and Stream table. Also mention the degree and cardinality produced after applying the Cartesian product.

j) Add a new column 'TeacherIncharge" in the Stream table. Insert appropriate data in each row.

k) List the names of teachers and students.

l) If Cartesian product is again applied on Student and Stream tables, what will be the degree and cardinality of this modified table?

Chapter 2

Data Handling Using Pandas - I

> "If you don't think carefully, you might believe that programming is just typing statements in a programming language."
>
> — W. Cunningham

12149CH02

In this chapter

2.1 Introduction to Python Libraries

Python libraries contain a collection of built-in modules that allow us to perform many actions without writing detailed programs for it. Each library in Python contains a large number of modules that one can import and use.

NumPy, Pandas and Matplotlib are three well-established Python libraries for scientific and analytical use. These libraries allow us to manipulate, transform and visualise data easily and efficiently.

NumPy, which stands for 'Numerical Python', is a library we discussed in class XI. Recall that, it is a package that can be used for numerical data analysis and

NOTES

scientific computing. NumPy uses a multidimensional array object and has functions and tools for working with these arrays. Elements of an array stay together in memory, hence, they can be quickly accessed.

PANDAS (PANel DAta) is a high-level data manipulation tool used for analysing data. It is very easy to import and export data using Pandas library which has a very rich set of functions. It is built on packages like NumPy and Matplotlib and gives us a single, convenient place to do most of our data analysis and visualisation work. Pandas has three important data structures, namely – Series, DataFrame and Panel to make the process of analysing data organised, effective and efficient.

The Matplotlib library in Python is used for plotting graphs and visualisation. Using Matplotlib, with just a few lines of code we can generate publication quality plots, histograms, bar charts, scatterplots, etc. It is also built on Numpy, and is designed to work well with Numpy and Pandas.

You may think what the need for Pandas is when NumPy can be used for data analysis. Following are some of the differences between Pandas and Numpy:

1. A Numpy array requires homogeneous data, while a Pandas DataFrame can have different data types (float, int, string, datetime, etc.).
2. Pandas have a simpler interface for operations like file loading, plotting, selection, joining, GROUP BY, which come very handy in data-processing applications.
3. Pandas DataFrames (with column names) make it very easy to keep track of data.
4. Pandas is used when data is in Tabular Format, whereas Numpy is used for numeric array based data manipulation.

2.1.1. Installing Pandas

Installing Pandas is very similar to installing NumPy. To install Pandas from command line, we need to type in:

```
pip install pandas
```

Note that both NumPy and Pandas can be installed only when Python is already installed on that system. The same is true for other libraries of Python.

2.1.2. Data Structure in Pandas

A data structure is a collection of data values and operations that can be applied to that data. It enables efficient storage, retrieval and modification to the data. For example, we have already worked with a data structure ndarray in NumPy in Class XI. Recall the ease with which we can store, access and update data using a NumPy array. Two commonly used data structures in Pandas that we will cover in this book are:

- Series
- DataFrame

2.2 Series

A Series is a one-dimensional array containing a sequence of values of any data type (int, float, list, string, etc) which by default have numeric data labels starting from zero. The data label associated with a particular value is called its index. We can also assign values of other data types as index. We can imagine a Pandas Series as a column in a spreadsheet. Example of a series containing names of students is given below:

```
Index     Value
0         Arnab
1         Samridhi
2         Ramit
3         Divyam
4         Kritika
```

2.2.1 Creation of Series

There are different ways in which a series can be created in Pandas. To create or use series, we first need to import the Pandas library.

(A) Creation of Series from Scalar Values

A Series can be created using scalar values as shown in the example below:

```
>>> import pandas as pd   #import Pandas with alias pd
>>> series1 = pd.Series([10,20,30])  #create a Series
>>> print(series1)  #Display the series
```

Output:

```
0    10
1    20
2    30
dtype: int64
```

Observe that output is shown in two columns - the index is on the left and the data value is on the right. If we do not explicitly specify an index for the data values while creating a series, then by default indices range from 0 through N – 1. Here N is the number of data elements.

We can also assign user-defined labels to the index and use them to access elements of a Series. The following example has a numeric index in random order.

Activity 2.1

Create a series having names of any five famous monuments of India and assign their States as index values.

```
>>> series2 = pd.Series(["Kavi","Shyam","Ra
vi"], index=[3,5,1])
>>> print(series2)  #Display the series
```

Output:

```
3     Kavi
5    Shyam
1     Ravi
dtype: object
```

Here, data values Kavi, Shyam and Ravi have index values 3, 5 and 1, respectively. We can also use letters or strings as indices, for example:

```
>>> series2 = pd.Series([2,3,4],index=["Feb","M
ar","Apr"])
>>> print(series2) #Display the series
```

Output:

```
Feb    2
Mar    3
Apr    4
dtype: int64
```

Here, data values 2,3,4 have index values Feb, Mar and Apr, respectively.

Think and Reflect

While importing Pandas, is it mandatory to always use pd as an alias name? What would happen if we give any other name?

(B) Creation of Series from NumPy Arrays

We can create a series from a one-dimensional (1D) NumPy array, as shown below:

```
>>> import numpy as np  # import NumPy with alias np
>>> import pandas as pd
>>> array1 = np.array([1,2,3,4])
>>> series3 = pd.Series(array1)
>>> print(series3)
```

Output:

```
0    1
1    2
2    3
3    4
dtype: int32
```

Notes

The following example shows that we can use letters or strings as indices:

```
>>> series4 = pd.Series(array1, index = ["Jan",
"Feb", "Mar", "Apr"])
>>> print(series4)
Jan    1
Feb    2
Mar    3
Apr    4
dtype: int32
```

When index labels are passed with the array, then the length of the index and array must be of the same size, else it will result in a ValueError. In the example shown below, array1 contains 4 values whereas there are only 3 indices, hence ValueError is displayed.

```
>>> series5 = pd.Series(array1, index = ["Jan",
"Feb", "Mar"])
ValueError: Length of passed values is 4, index
implies 3
```

(C) Creation of Series from Dictionary

Recall that Python dictionary has key: value pairs and a value can be quickly retrieved when its key is known. Dictionary keys can be used to construct an index for a Series, as shown in the following example. Here, keys of the dictionary dict1 become indices in the series.

```
>>> dict1 = {'India': 'NewDelhi', 'UK':
'London', 'Japan': 'Tokyo'}
>>> print(dict1)  #Display the dictionary
{'India': 'NewDelhi', 'UK': 'London', 'Japan':
'Tokyo'}
>>> series8 = pd.Series(dict1)
>>> print(series8)  #Display the series
India    NewDelhi
UK         London
Japan       Tokyo
dtype: object
```

2.2.2 Accessing Elements of a Series

There are two common ways for accessing the elements of a series: Indexing and Slicing.

(A) Indexing

Indexing in Series is similar to that for NumPy arrays, and is used to access elements in a series. Indexes are of two types: positional index and labelled index. Positional index takes an integer value that corresponds to its position in the series starting from 0, whereas labelled index takes any user-defined label as index.

- Following example shows usage of the positional index for accessing a value from a Series.

```
>>> seriesNum = pd.Series([10,20,30])
>>> seriesNum[2]
30
```

Here, the value 30 is displayed for the positional index 2.

When labels are specified, we can use labels as indices while selecting values from a Series, as shown below. Here, the value 3 is displayed for the labelled index Mar.

```
>>> seriesMnths = pd.Series([2,3,4],index=["Feb
","Mar","Apr"])
>>> seriesMnths["Mar"]
3
```

In the following example, value NewDelhi is displayed for the labelled index India.

```
>>> seriesCapCntry = pd.Series(['NewDelhi',
     'WashingtonDC', 'London', 'Paris'],
index=['India', 'USA', 'UK', 'France'])
>>> seriesCapCntry['India']
'NewDelhi'
```

Activity 2.2

Write the statement to get NewDelhi as output using positional index.

We can also access an element of the series using the positional index:

```
>>> seriesCapCntry[1]
'WashingtonDC'
```

More than one element of a series can be accessed using a list of positional integers or a list of index labels as shown in the following examples:

```
>>> seriesCapCntry[[3,2]]
France     Paris
UK        London
dtype: object

>>> seriesCapCntry[['UK','USA']]
UK           London
USA    WashingtonDC
dtype: object
```

The index values associated with the series can be altered by assigning new index values as shown in the following example:

```
>>> seriesCapCntry.index=[10,20,30,40]
>>> seriesCapCntry
```

```
10        NewDelhi
20    WashingtonDC
30          London
40           Paris
dtype: object
```

(B) Slicing

Sometimes, we may need to extract a part of a series. This can be done through slicing. This is similar to slicing used with NumPy arrays. We can define which part of the series is to be sliced by specifying the start and end parameters [start :end] with the series name. When we use positional indices for slicing, the value at the endindex position is excluded, i.e., only (end - start) number of data values of the series are extracted. Consider the following series seriesCapCntry:

```
>>> seriesCapCntry = pd.Series(['NewDelhi', 'WashingtonDC', 'London',
'Paris'], index=['India', 'USA', 'UK', 'France'])

 >>> seriesCapCntry[1:3] #excludes the value at index position 3

USA    WashingtonDC
UK           London
dtype: object
```

As we can see that in the above output, only data values at indices 1 and 2 are displayed. If labelled indexes are used for slicing, then value at the end index label is also included in the output, for example:

```
>>> seriesCapCntry['USA' : 'France']

USA       WashingtonDC
UK              London
France           Paris
dtype: object
```

We can also get the series in reverse order, for example:

```
>>> seriesCapCntry[ : : -1]
France           Paris
UK              London
USA       WashingtonDC
India         NewDelhi
dtype: object
```

We can also use slicing to modify the values of series elements as shown in the following example:

```
>>> import numpy as np
>>> seriesAlph = pd.Series(np.arange(10,16,1),
index = ['a', 'b', 'c', 'd', 'e', 'f'])
>>> seriesAlph
a    10
b    11
c    12
d    13
e    14
f    15
dtype: int32

>>> seriesAlph[1:3] = 50
>>> seriesAlph
a    10
b    50
c    50
d    13
e    14
f    15
dtype: int32
```

Observe that updating the values in a series using slicing also excludes the value at the end index position. But, it changes the value at the end index label when slicing is done using labels.

```
>>> seriesAlph['c':'e'] = 500
>>> seriesAlph
a     10
b     50
c    500
d    500
e    500
f     15
dtype: int32
```

2.2.3 Attributes of Series

We can access certain properties called attributes of a series by using that property with the series name. Table 2.1 lists some attributes of Pandas series usingseriesCapCntry as an example:

```
>>> seriesCapCntry
India         NewDelhi
USA       WashingtonDC
UK              London
France           Paris
dtype: object
```

Table 2.1 Attributes of Pandas Series

Attribute Name	Purpose	Example

name	assigns a name to the Series	>>> seriesCapCntry.name = ‘Capitals’
		>>> print(seriesCapCntry)
		India NewDelhi
		USA WashingtonDC
		UK London
		France Paris
		Name: Capitals, dtype: object

index.name	assigns a name to the index of the series	>>>seriesCapCntry.index.name = ‘Countries’ >>> print(seriesCapCntry) Countries India NewDelhi USA WashingtonDC UK London France Paris Name: Capitals, dtype: object
values	prints a list of the values in the series	>>> print(seriesCapCntry.values) [‘NewDelhi’ ‘WashingtonDC’ ‘London’ ‘Paris’]
size	prints the number of values in the Series object	>>> print(seriesCapCntry.size) 4

empty	prints True if the series is empty, and False otherwise	>>> seriesCapCntry.empty
		False
		
		# Create an empty series
		seriesEmpt=pd.Series()
		>>> seriesEmpt.empty
		True

2.2.4 Methods of Series

In this section, we are going to discuss some of the methods that are available for Pandas Series. Let us consider the following series:

```
>>> seriesTenTwenty=pd.Series(np.arange( 10,
20, 1 ))
>>> print(seriesTenTwenty)
0     10
1     11
2     12
3     13
4     14
5     15
6     16
7     17
8     18
9     19
dtype: int32
```

Activity 2.3

Consider the following code:

```
>>>import pandas as pd
>>>import numpy as np
>>>s2=pd.
Series([12,np.nan,10])
>>>print(s2)
```

Find output of the above code and write a Python statement to count and display only non null values in the above series.

Method	Explanation	Example

head(n)	Returns the first n members of the series. If the value for n is not passed, then by default n takes 5 and the first five members are displayed.	>>> seriesTenTwenty.head(2)
		0 10
		1 11
		dtype: int32
		
		>>> seriesTenTwenty.head()
		0 10
		1 11
		2 12
		3 13
		4 14
		dtype: int32

count()	Returns the number of non-NaN values in the Series	>>> seriesTenTwenty.count() 10

tail(n)	Returns the last n members of the series. If the value for n is not passed, then by default n takes 5 and the last five members are displayed.	>>> seriesTenTwenty.tail(2)
		8 18
		9 19
		dtype: int32
		
		>>> seriesTenTwenty.tail()
		5 15
		6 16
		7 17
		8 18
		9 19
		dtype: int32

2.2.5 Mathematical Operations on Series

We have learnt in Class XI that if we perform basic mathematical operations like addition, subtraction, multiplication, division, etc., on two NumPy arrays, the operation is done on each corresponding pair of elements. Similarly, we can perform mathematical operations on two series in Pandas.

While performing mathematical operations on series, index matching is implemented and all missing values are filled in with NaN by default.

Consider the following series: seriesA and seriesB for understanding mathematical operations on series in Pandas.

```
>>> seriesA = pd.Series([1,2,3,4,5], index =
['a', 'b', 'c', 'd', 'e'])

>>> seriesA
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

Notes

```
>>> seriesB = pd.Series([10,20,-10,-50,100],
index = ['z', 'y', 'a', 'c', 'e'])
>>> seriesB
z      10
y      20
a     -10
c     -50
e     100
dtype: int64
```

(A) Addition of two Series

It can be done in two ways. In the first method, two series are simply added together, as shown in the following code. Table 2.2 shows the detailed values that were matched while performing the addition. Note here that the output of addition is NaN if one of the elements or both elements have no value.

```
>>> seriesA + seriesB
a      -9.0
b       NaN
c     -47.0
d       NaN
e     105.0
y       NaN
z       NaN
dtype: float64
```

Table 2.2 Details of addition of two series

index	value from seriesA	value from seriesB	seriesA + seriesB
a	1	-10	-9.0
b	2		NaN
c	3	-50	-47.0
d	4		NaN
e	5	100	105.00
y		20	NaN
z		10	NaN

The second method is applied when we do not want to have NaN values in the output. We can use the series method add() and a parameter fill_value to replace missing value with a specified value. That is, calling seriesA.add(seriesB) is equivalent to calling seriesA+seriesB, but add() allows explicit specification of the fill value for any element in seriesA or seriesB that might be missing, as shown in Table 2.3.

```
>>> seriesA.add(seriesB, fill_value=0)

a      -9.0
b       2.0
c     -47.0
d       4.0
e     105.0
y      20.0
z      10.0
dtype: float64
```

Table 2.3 Details of addition of two series using add() method

index	value from seriesA	value from seriesB	seriesA + seriesB
a	1	-10	-9.0
b	2	0	2.0
c	3	-50	-47.0
d	4	0	4.0
e	5	100	105.00
y	0	20	20.0
z	0	10	10.0

Note that Table 2.2 shows the changes in the series elements and corresponding output without replacing the missing values, while Table 2.3 shows the changes in the series elements and corresponding output after replacing missing values by 0. Just like addition, subtraction, multiplication and division can also be done using corresponding mathematical operators or explicitly calling of the appropriate method.

Activity 2.4

Draw two tables for subtraction similar to tables 2.2 and 2.3 showing the changes in the series elements and corresponding output without replacing the missing values, and after replacing the missing values with 1000.

(B) Subtraction of two Series

Again, it can be done in two different ways, as shown in the following examples:

```
>>> seriesA – seriesB #using subtraction operator
a      11.0
b      NaN
c      53.0
d      NaN
e      -95.0
y      NaN
z      NaN
dtype: float64
```

Let us now replace the missing values with 1000 before subtracting seriesB from seriesA using explicit subtraction method sub().

Activity 2.5

Draw two tables for multiplication similar to Tables 2.2 and 2.3 showing the changes in the series elements and corresponding output without replacing the missing values, and after replacing the missing values with 0.

```
>>> seriesA.sub(seriesB, fill_value=1000)
# using fill value 1000 while making explicit
# call of the method”

a      11.0
b      -998.0
c      53.0
d      -996.0
e      -95.0
y      980.0
z      990.0
dtype: float64
```

(C) Multiplication of two Series

Again, it can be done in two different ways, as shown in the following examples:

```
>>>seriesA * seriesB #using multiplication operator
a     -10.0
b       NaN
c    -150.0
d       NaN
e     500.0
y       NaN
z       NaN
dtype: float64
```

Let us now replace the missing values with 0 before multiplication of seriesB with seriesA using explicit multiplication method mul().

```
>>> seriesA.mul(seriesB, fill_value=0)
# using fill value 0 while making
#explicit call of the method
a     -10.0
b       0.0
c    -150.0
d       0.0
e     500.0
y       0.0
z       0.0
dtype: float64
```

Activity 2.6

Draw two tables for division similar to tables 2.2 and 2.3 showing the changes in the series elements and corresponding output without replacing the missing values, and after replacing the missing values with 0.

(D) Division of two Series

Again, it can be done in two different ways, as shown in the following examples:

```
>>> seriesA/seriesB  # using division operator
a     -0.10
b      NaN
c      -0.06
d      NaN
```

> Explicit call to a mathematical operation is preferred when series may have missing values and we want to replace it by a specific value to have a concrete output in place of NaN.

```
e       0.05
y      NaN
z      NaN
dtype: float64
```

Let us now replace the missing values with 0 before dividing seriesA by seriesB using explicit division method div().

```
# using fill value 0 while making explicit
# call of the method

a    -0.10
b     inf
c    -0.06
d     inf
e     0.05
y     0.00
z     0.00
dtype: float64
```

2.3 DataFrame

Sometimes we need to work on multiple columns at a time, i.e., we need to process the tabular data. For example, the result of a class, items in a restaurant's menu, reservation chart of a train, etc. Pandas store such tabular data using a DataFrame. A DataFrame is a two-dimensional labelled data structure like a table of MySQL. It contains rows and columns, and therefore has both a row and column index. Each column can have a different type of value such as numeric, string, boolean, etc., as in tables of a database.

Column Indexes

	State	Geographical Area (sq Km)	Area under Very Dense Forests (sq Km)
1	Assam	78438	2797
2	Delhi	1483	6.72
3	Kerala	38852	1663

Row Indexes

2.3.1 Creation of DataFrame

There are a number of ways to create a DataFrame. Some of them are listed in this section.

(A) Creation of an empty DataFrame

An empty DataFrame can be created as follows:

```
>>> import pandas as pd
>>> dFrameEmt = pd.DataFrame()
>>> dFrameEmt

Empty DataFrame
Columns: []
Index: []
```

Think and Reflect

What would happen if we pass 3 columns or 5 columns instead of 4 in the above code? What is the reason?

(B) Creation of DataFrame from NumPy ndarrays

Consider the following three NumPy ndarrays. Let us create a simple DataFrame without any column labels, using a single ndarray:

```
>>> import numpy as np
>>> array1 = np.array([10,20,30])
>>> array2 = np.array([100,200,300])
>>> array3 = np.array([-10,-20,-30, -40])

>>> dFrame4 = pd.DataFrame(array1)
>>> dFrame4
    0
0  10
1  20
2  30
```

We can create a DataFrame using more than one ndarrays, as shown in the following example:

```
>>> dFrame5 = pd.DataFrame([array1, array3,
array2], columns=[ 'A', 'B', 'C', 'D'])
>>> dFrame5
     A    B    C     D
0   10   20   30   NaN
1  -10  -20  -30 -40.0
2  100  200  300   NaN
```

(C) Creation of DataFrame from List of Dictionaries

We can create DataFrame from a list of Dictionaries, for example:

```
# Create list of dictionaries
>>> listDict = [{'a':10, 'b':20}, {'a':5,
'b':10, 'c':20}]

>>> dFrameListDict = pd.DataFrame(listDict)
>>> dFrameListDict
    a   b     c
0  10  20   NaN
1   5  10  20.0
```

Here, the dictionary keys are taken as column labels, and the values corresponding to each key are taken as rows. There will be as many rows as the number of dictionaries present in the list. In the above example there are two dictionaries in the list. So, the DataFrame consists of two rows. Number of columns in a DataFrame is equal to the maximum number of keys in any dictionary of the list. Hence, there are three columns as the second dictionary has three elements. Also, note that NaN (Not a Number) is inserted if a corresponding value for a column is missing.

(D) Creation of DataFrame from Dictionary of Lists

DataFrames can also be created from a dictionary of lists. Consider the following dictionary consisting of the keys 'State', 'GArea' (geographical area) and 'VDF' (very dense forest) and the corresponding values as list.

```
>>> dictForest = {'State': ['Assam', 'Delhi',
'Kerala'],
          'GArea': [78438, 1483, 38852] ,
          'VDF' : [2797, 6.72,1663]}
>>> dFrameForest= pd.DataFrame(dictForest)
>>> dFrameForest
    State   GArea    VDF
0   Assam   78438  2797.00
1   Delhi   1483     6.72
2   Kerala  38852  1663.00
```

Note that dictionary keys become column labels by default in a DataFrame, and the lists become the rows. Thus, a DataFrame can be thought of as a dictionary of lists or a dictionary of series.

We can change the sequence of columns in a DataFrame. This can be done by assigning a particular sequence of the dictionary keys as columns parameter, for example:

```
>>> dFrameForest1 = pd.DataFrame(dictForest,
columns = ['State','VDF', 'GArea'])
>>> dFrameForest1
    State      VDF   GArea
0   Assam   2797.00  78438
1   Delhi     6.72   1483
2   Kerala  1663.00  38852
```

In the output, VDF is now displayed as the middle column instead of last.

(E) Creation of DataFrame from Series

Consider the following three Series:

```
seriesA = pd.Series([1,2,3,4,5],
          index = ['a', 'b', 'c', 'd', 'e'])

seriesB = pd.Series ([1000,2000,-1000,-5000,1000],
          index = ['a', 'b', 'c', 'd', 'e'])
```

```
seriesC = pd.Series([10,20,-10,-50,100],
         index = ['z', 'y', 'a', 'c', 'e'])
```

We can create a DataFrame using a single series as shown below:

```
>>> dFrame6 = pd.DataFrame(seriesA)
>>> dFrame6
   0
a  1
b  2
c  3
d  4
e  5
```

Here, the DataFrame dFrame6 has as many numbers of rows as the numbers of elements in the series, but has only one column. To create a DataFrame using more than one series, we need to pass multiple series in the list as shown below:

```
>>> dFrame7 = pd.DataFrame([seriesA, seriesB])
>>> dFrame7
a     b     c     d     e
0     1     2     3     4     5
1  1000  2000 -1000 -5000  1000
```

Observe that the labels in the series object become the column names in the DataFrame object and each series becomes a row in the DataFrame. Now look at the following example:

```
>>> dFrame8 = pd.DataFrame([seriesA, seriesC])
>>> dFrame8
a    b     c    d      e     z     y
0   1.0  2.0   3.0  4.0    5.0   NaN   NaN
1 -10.0  NaN -50.0  NaN  100.0  10.0  20.0
```

Here, different series do not have the same set of labels. But, the number of columns in a DataFrame equals to distinct labels in all the series. So, if a particular series does not have a corresponding value for a label, NaN is inserted in the DataFrame column.

(F) Creation of DataFrame from Dictionary of Series

A dictionary of series can also be used to create a DataFrame. For example, ResultSheet is a dictionary of series containing marks of 5 students in three subjects. The names of the students are the keys to the dictionary, and the index values of the series are the subject names as shown below:

```
>>> ResultSheet={
'Arnab': pd.Series([90, 91, 97],
          index=['Maths','Science','Hindi']),
'Ramit': pd.Series([92, 81, 96],
          index=['Maths','Science','Hindi']),
'Samridhi': pd.Series([89, 91, 88],
          index=['Maths','Science','Hindi']),
'Riya': pd.Series([81, 71, 67],
          index=['Maths','Science','Hindi']),
'Mallika': pd.Series([94, 95, 99],
          index=['Maths','Science','Hindi'])}

>>> ResultDF = pd.DataFrame(ResultSheet)
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya  Mallika
Maths       90     92        89    81       94
Science     91     81        91    71       95
Hindi       97     96        88    67       99
```

Activity 2.7

Use the type function to check the datatypes of ResultSheet and ResultDF. Are they the same?

The following output shows that every column in the DataFrame is a Series:

```
>>> type(ResultDF.Arnab)
<class 'pandas.core.series.Series'>
```

When a DataFrame is created from a Dictionary of Series, the resulting index or row labels are a union of all series indexes used to create the DataFrame. For example:

```
dictForUnion = { 'Series1' :
pd.Series([1,2,3,4,5],
          index = ['a', 'b', 'c', 'd', 'e']) ,
          'Series2' :
pd.Series([10,20,-10,-50,100],
          index = ['z', 'y', 'a', 'c', 'e']),
          'Series3' :
pd.Series([10,20,-10,-50,100],
          index = ['z', 'y', 'a', 'c', 'e']) }

>>> dFrameUnion = pd.DataFrame(dictForUnion)
>>> dFrameUnion

   Series1  Series2  Series3
a      1.0    -10.0    -10.0
b      2.0      NaN      NaN
c      3.0    -50.0    -50.0
d      4.0      NaN      NaN
e      5.0    100.0    100.0
y      NaN     20.0     20.0
z      NaN     10.0     10.0
```

2.3.2 Operations on rows and columns in DataFrames

We can perform some basic operations on rows and columns of a DataFrame like selection, deletion, addition, and renaming, as discussed in this section.

(A) Adding a New Column to a DataFrame

We can easily add a new column to a DataFrame. Let us consider the DataFrame ResultDF defined earlier. In order to add a new column for another student 'Preeti', we can write the following statement:

```
>>> ResultDF['Preeti']=[89,78,76]
>>> ResultDF

          Arnab  Ramit  Samridhi  Riya  Mallika Preeti
Maths        90     92        89    81       94     89
Science      91     81        91    71       95     78
Hindi        97     96        88    67       99     76
```

Assigning values to a new column label that does not exist will create a new column at the end. If the column already exists in the DataFrame then the assignment statement will update the values of the already existing column, for example:

```
>>> ResultDF['Ramit']=[99, 98, 78]
>>> ResultDF
          Arnab  Ramit  Samridhi  Riya  Mallika Preeti
Maths        90     99        89    81       94     89
Science      91     98        91    71       95     78
Hindi        97     78        88    67       99     76
```

We can also change data of an entire column to a particular value in a DataFrame. For example, the following statement sets marks=90 for all subjects for the column name 'Arnab':

```
>>> ResultDF['Arnab']=90
>>> ResultDF

          Arnab  Ramit  Samridhi  Riya  Mallika Preeti
Maths        90     99        89    81       94     89
Science      90     98        91    71       95     78
Hindi        90     78        88    67       99     76
```

(B) Adding a New Row to a DataFrame

We can add a new row to a DataFrame using the DataFrame.loc[ ] method. Consider the DataFrame ResultDF that has three rows for the three subjects – Maths, Science and Hindi. Suppose, we need to add the marks for English subject in ResultDF, we can use the following statement:

```
>>> ResultDF
          Arnab  Ramit  Samridhi  Riya  Mallika  Preeti
Maths        90     92        89    81       94   89
Science      91     81        91    71       95      78
Hindi        97     96        88    67       99      76
```

```
>>> ResultDF.loc['English'] = [85, 86, 83, 80, 90, 89]
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya  Mallika  Preeti
Maths       90     92        89    81       94  89
Science     91     81        91    71       95      78
Hindi       97     96        88    67       99      76
English     85     86        83    80       90      89
```

We cannot use this method to add a row of data with already existing (duplicate) index value (label). In such case, a row with this index label will be updated, for example:

```
>>> ResultDF.loc['English'] = [95, 86, 95, 80, 95,99]
>>> ResultDF

         Arnab  Ramit  Samridhi  Riya  Mallika  Preeti
Maths       90     92        89    81       94  89
Science     91     81        91    71       95      78
Hindi       97     96        88    67       99      76
English     95     86        95    80       95      99
```

DataFRame.loc[] method can also be used to change the data values of a row to a particular value. For example, the following statement sets marks in 'Maths' for all columns to 0:

```
>>> ResultDF.loc['Maths']=0
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya  Mallika  Preeti
Maths        0      0         0     0        0    0
Science     91     81        91    71       95      78
Hindi       97     96        88    67       99      76
English     95     86        95    80       95      99
```

If we try to add a row with lesser values than the number of columns in the DataFrame, it results in a ValueError, with the error message: ValueError: Cannot set a row with mismatched columns.

Think and Reflect

Can you write a program to count the number of rows and columns in a DataFrame?

Similarly, if we try to add a column with lesser values than the number of rows in the DataFrame, it results in a ValueError, with the error message: ValueError: Length of values does not match length of index.

Further, we can set all values of a DataFrame to a particular value, for example:

```
>>> ResultDF[: ] = 0 # Set all values in ResultDF to 0
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya  Mallika  Preeti
Maths        0      0         0     0        0    0
Science      0      0         0     0        0    0
Hindi        0      0         0     0        0    0
English      0      0         0     0        0    0
```

NOTES

(C) Deleting Rows or Columns from a DataFrame

We can use the DataFrame.drop() method to delete rows and columns from a DataFrame. We need to specify the names of the labels to be dropped and the axis from which they need to be dropped. To delete a row, the parameter axis is assigned the value 0 and for deleting a column,the parameter axis is assigned the value 1. Consider the following DataFrame:

```
 >>> ResultDF
         Arnab  Ramit  Samridhi  Riya Mallika
Maths       90     92        89    81      94
Science     91     81        91    71      95
Hindi       97     96        88    67      99
English     95     86        95    80      95
```

The following example shows how to delete the row with label 'Science':

```
>>> ResultDF = ResultDF.drop('Science', axis=0)
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya Mallika
Maths       90     92        89    81      94
Hindi       97     96        88    67      99
English     95     86        95    80      95
```

The following example shows how to delete the columns having labels 'Samridhi', 'Ramit' and 'Riya':

```
 >>> ResultDF = ResultDF.drop(['Samridhi','Rami
t','Riya'], axis=1)
 >>> ResultDF
         Arnab  Mallika
Maths       90       94
Hindi       97       99
English     95       95
```

If the DataFrame has more than one row with the same label, the DataFrame.drop() method will delete all the matching rows from it. For example, consider the following DataFrame:

```
>>> ResultDF

         Arnab  Ramit  Samridhi  Riya  Mallika
Maths     90     92        89    81       94
Science   91     81        91    71       95
Hindi     97     96        88    67       99
Hindi     97     89        78    60       45
```

To remove the duplicate rows labelled 'Hindi', we need to write the following statement:

```
 >>> ResultDF= ResultDF.drop('Hindi', axis=0)
>>> ResultDF
```

```
         Arnab  Ramit  Samridhi  Riya  Mallika
Maths     90     92        89     81       94
Science   91     81        91     71       95
```

(D) Renaming Row Labels of a DataFrame

We can change the labels of rows and columns in a DataFrame using the DataFrame.rename() method. Consider the following DataFrame. To rename the row indices Maths to sub1, Science to sub2, Hindi to sub3 and English to sub4 we can write the following statement:

```
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya  Mallika
Maths     90     92        89     81       94
Science   91     81        91     71       95
English   97     96        88     67       99
Hindi     97     89        78     60       45

>>> ResultDF=ResultDF.rename({'Maths':'Sub1',
‘Science':'Sub2','English':'Sub3',
'Hindi':'Sub4'}, axis='index')
>>> print(ResultDF)

         Arnab  Ramit  Samridhi  Riya  Mallika
Sub1     90     92        89     81       94
Sub2     91     81        91     71       95
Sub3     97     96        88     67       99
Sub4     97     89        78     60       45
```

Think and Reflect

What if in the rename function we pass a value for a row label that does not exist?

The parameter axis='index' is used to specify that the row label is to be changed. If no new label is passed corresponding to an existing label, the existing row label is left as it is, for example:

```
>>> ResultDF=ResultDF.rename({'Maths':'Sub1',‘S
cience':'Sub2','Hindi':'Sub4'}, axis='index')
>>> print(ResultDF)

         Arnab  Ramit  Samridhi  Riya  Mallika
Sub1     90     92        89     81       94
Sub2     91     81        91     71       95
English  97     96        88     67       99
Sub4     97     89        78     60       45
```

(E) Renaming Column Labels of a DataFrame

To alter the column names of ResultDF we can again use the rename() method, as shown below. The parameter axis='columns' implies we want to change the column labels:

```
 >>> ResultDF=ResultDF.rename({'Arnab':'Student1','Ramit':'Student2','
Samridhi':'Student3','Mallika':'Student4'},axis='columns')
>>> print(RsultDF)
```

```
         Student1  Student2  Student3  Riya  Student4
Maths          90        92        89    81        94
Science        91        81        91    71        95
English        97        96        88    67        99
Hindi          97        89        78    60        45
```

Note that the column Riya remains unchanged since we did not pass any new label.

2.3.3 Accessing DataFrames Element through Indexing

Data elements in a DataFrame can be accessed using indexing.There are two ways of indexing Dataframes : Label based indexing and Boolean Indexing.

Think and Reflect

What would happen if the label or row index passed is not present in the DataFrame?

(A) Label Based Indexing

There are several methods in Pandas to implement label based indexing. DataFrame.loc[ ] is an important method that is used for label based indexing with DataFrames. Let us continue to use the ResultDF created earlier. As shown in the following example, a single row label returns the row as a Series.

```
>>> ResultDF
         Arnab  Ramit  Samridhi  Riya  Mallika
Maths       90     92        89    81       94
Science     91     81        91    71       95
Hindi       97     96        88    67       99

>>> ResultDF.loc['Science']

Arnab     91
Ramit     81
Samridhi  91
Riya      71
Mallika   95
Name: Science, dtype: int64
```

Also, note that when the row label is passed as an integer value, it is interpreted as a label of the index and not as an integer position along the index, for example:

```
>>> dFrame10Multiples = pd.DataFrame([10,20,30,40,50])

>>> dFrame10Multiples.loc[2]
0    30
Name: 2, dtype: int64
```

When a single column label is passed, it returns the column as a Series.

```
>>> ResultDF.loc[:,'Arnab']
```

NOTES

```
Maths     90
Science   91
Hindi     97
Name: Arnab, dtype: int64
```

Also, we can obtain the same result that is the marks of 'Arnab' in all the subjects by using the command:

```
>>> print(df['Arnab'])

Maths       56
Science     91
English     97 Hindi  97
Name: Arnab, dtype: int64
```

To read more than one row from a DataFrame, a list of row labels is used as shown below. Note that using [[]] returns a DataFrame.

```
>>> ResultDF.loc[['Science', 'Hindi']]
        Arnab  Ramit  Samridhi  Riya  Mallika
Science    91     81        91    71       95
Hindi      97     96        88    67       99
```

(B) Boolean Indexing

Boolean means a binary variable that can represent either of the two states - True (indicated by 1) or False (indicated by 0). In Boolean indexing, we can select the subsets of data based on the actual values in the DataFrame rather than their row/column labels. Thus, we can use conditions on column names to filter data values. Consider the DataFrame ResultDF, the following statement displays True or False depending on whether the data value satisfies the given condition or not.

```
>>> ResultDF.loc['Maths'] > 90
Arnab        False
Ramit         True
Samridhi     False
Riya         False
Mallika       True
Name: Maths, dtype: bool
```

To check in which subjects 'Arnab' has scored more than 90, we can write:

```
>>> ResultDF.loc[:,'Arnab']>90
Maths       False
Science      True
Hindi        True
Name: Arnab, dtype: bool
```

2.3.4 Accessing DataFrames Element through Slicing

We can use slicing to select a subset of rows and/or columns from a DataFrame. To retrieve a set of rows, slicing can be used with row labels. For example:

```
>>> ResultDF.loc['Maths': 'Science']
        Arnab  Ramit  Samridhi  Riya  Mallika
Maths      90     92        89    81       94
Science    91     81        91    71       95
```

Here, the rows with labels Maths and Science are displayed. Note that in DataFrames slicing is inclusive of the end values. We may use a slice of labels with a column name to access values of those rows in that column only. For example, the following statement displays the rows with label Maths and Science, and column with label Arnab:

```
>>> ResultDF.loc['Maths': 'Science', ‘Arnab’]
Maths         90
Science       91
Name: Arnab, dtype: int64
```

We may use a slice of labels with a slice of column names to access values of those rows and columns:

```
>>> ResultDF.loc['Maths': 'Science', ‘Arnab’:’Samridhi’]
              Arnab   Ramit    Samridhi
Maths            90      92          89
Science          91      81          91
```

Alternatively, we may use a slice of labels with a list of column names to access values of those rows and columns:

```
>>> ResultDF.loc['Maths': 'Science',[‘Arnab’,’Samridhi’]]
             Arnab     Samridhi
Maths           90           89
Science         91           91
```

Activity 2.8

a) Using the DataFrame ResultDF, write the statement to access Marks of Arnab in Maths.

b) Create a DataFrame having 5 rows and write the statement to get the first 4 rows of it.

Filtering Rows in DataFrames

In DataFrames, Boolean values like True (1) and False (0) can be associated with indices. They can also be used to filter the records using the DataFrmae.loc[] method.

In order to select or omit particular row(s), we can use a Boolean list specifying ‘True’ for the rows to be shown and ‘False’ for the ones to be omitted in the output. For example, in the following statement, row having index as Science is omitted:

```
>>> ResultDF.loc[[True, False, True]]
         Arnab  Ramit  Samridhi  Riya  Mallika
Maths       90     92        89    81       94
Hindi       97     96        88    67       99
```

NOTES

2.3.5 Joining, Merging and Concatenation of DataFrames

(A) Joining

We can use the pandas.DataFrame.append() method to merge two DataFrames. It appends rowsof the second DataFrame at the end of the first DataFrame. Columns not present in the first DataFrame are added as new columns. For example, consider the two DataFrames—dFrame1 and dFrame2described below. Let us use theappend() method to append dFrame2 to dFrame1:

```
>>> dFrame1=pd.DataFrame([[1, 2, 3], [4, 5],
[6]], columns=['C1', 'C2', 'C3'], index=['R1',
'R2', 'R3'])
>>> dFrame1
    C1   C2   C3
R1   1  2.0  3.0
R2   4  5.0  NaN
R3   6  NaN  NaN

>>> dFrame2=pd.DataFrame([[10, 20], [30], [40,
50]], columns=['C2', 'C5'], index=['R4', 'R2',
'R5'])
>>> dFrame2
    C2    C5
R4  10  20.0
R2  30   NaN
R5  40  50.0

>>> dFrame1=dFrame1.append(dFrame2)
>>> dFrame1
     C1    C2   C3    C5
R1  1.0   2.0  3.0   NaN
R2  4.0   5.0  NaN   NaN
R3  6.0   NaN  NaN   NaN
R4  NaN  10.0  NaN  20.0
R2  NaN  30.0  NaN   NaN
R5  NaN  40.0  NaN  50.0
```

Alternatively, if we append dFrame1 to dFrame2, the rows of dFrame2 precede the rows of dFrame1. To get the column labels appear in sorted order we can set the parameter sort=True. The column labels shall appear in unsorted order when the parameter sort = False.

```
# append dFrame1 to dFrame2
>>> dFrame2 =dFrame2.append(dFrame1,
sort=’True’)
>>> dFrame2
     C1    C2   C3    C5
R4  NaN  10.0  NaN  20.0
R2  NaN  30.0  NaN   NaN
```

```
R5  NaN  40.0  NaN  50.0
R1  1.0   2.0  3.0   NaN
R2  4.0   5.0  NaN   NaN
R3  6.0   NaN  NaN   NaN
# append dFrame1 to dFrame2 with sort=False
>>> dFrame2 = dFrame2.append(dFrame1,
sort=’False’)
>>> dFrame2
       C2    C5   C1   C3
R4  10.0  20.0  NaN  NaN
R2  30.0   NaN  NaN  NaN
R5  40.0  50.0  NaN  NaN
R1   2.0   NaN  1.0  3.0
R2   5.0   NaN  4.0  NaN
R3   NaN   NaN  6.0  NaN
```

The parameter verify_integrity of append()method may be set to True when we want to raise an error if the row labels are duplicate. By default, verify_integrity = False. That is why we could append the duplicate row with label R2 when appending the two DataFrames, as shown above.

The parameter ignore_index of append()method may be set to True, when we do not want to use row index labels. By default, ignore_index = False.

```
>>> dFrame1 = dFrame1.append(dFrame2, ignore_
index=True)
>>> dFrame1
     C1    C2   C3    C5
0  1.0   2.0  3.0   NaN
1  4.0   5.0  NaN   NaN
2  6.0   NaN  NaN   NaN
3  NaN  10.0  NaN  20.0
4  NaN  30.0  NaN   NaN
5  NaN  40.0  NaN  50.0
```

Think and Reflect

How can you check whether a given DataFrame has any missing value or not?

The append()method can also be used to append a series or a dictionary to a DataFrame.

2.3.6 Attributes of DataFrames

Like Series, we can access certain properties called attributes of a DataFrame by using that property with the DataFrame name. Table 2.4 lists some attributes of Pandas DataFrame. We are going to use a part of the data from a report called “STATE OF FOREST REPORT 2017”, Published by Forest Survey of India, accessible at http://fsi.nic.in/forest-report-2017, as our example data in this section.

As per this report, the geographical area, the area under very dense forests, the area under moderately dense forests, and the area under open forests (all in sq km), in three States of India - Assam, Delhi and Kerala are as shown in the following DataFrame ForestAreaDF:

```
>>> ForestArea = {
         'Assam' :pd.Series([78438, 2797,
10192, 15116], index = ['GeoArea', 'VeryDense',
'ModeratelyDense', 'OpenForest']),
         'Kerala' :pd.Series([ 38852, 1663,
9407, 9251],   index = ['GeoArea' ,'VeryDense',
'ModeratelyDense', 'OpenForest']),
         'Delhi' :pd.Series([1483, 6.72, 56.24,
129.45],  index = ['GeoArea', 'VeryDense',
'ModeratelyDense', 'OpenForest'])}

>>> ForestAreaDF = pd.DataFrame(ForestArea)
>>> ForestAreaDF
                 Assam  Kerala    Delhi
GeoArea          78438   38852  1483.00
VeryDense         2797    1663     6.72
ModeratelyDense  10192    9407    56.24
OpenForest       15116    9251   129.45
```

Table 2.4 Some Attributes of Pandas DataFrame

Attribute Name	Purpose	Example

DataFrame.index	to display row labels	`>>> ForestAreaDF.index`
		`Index([‘GeoArea’, ‘VeryDense’, ‘ModeratelyDense’, ‘OpenForest’], dtype =’object’)`

DataFrame.columns	to display column labels	`>>> ForestAreaDF.columns`
		`Index([‘Assam’, ‘Kerala’, ‘Delhi’], dtype=’object’)`

DataFrame.dtypes	to display data type of each column in the DataFrame	`>>> ForestAreaDF.dtypes`
		`Assam       int64`
		`Kerala      int64`
		`Delhi     float64`
		`dtype: object`

DataFrame.values	to display a NumPy ndarray having all the values in the DataFrame, without the axes labels	`>>> ForestAreaDF.values`
		`array([[7.8438e+04, 3.8852e+04, 1.4830e+03],`
		`[2.7970e+03, 1.6630e+03, 6.7200e+00],`
		`[1.0192e+04, 9.4070e+03, 5.6240e+01],`
		`[1.5116e+04, 9.2510e+03, 1.2945e+02]])`

DataFrame.shape	to display a tuple representing the dimensionality of the DataFrame	`>>> ForestAreaDF.shape`
		`(4, 3)`
		`It means ForestAreaDF has 4 rows and 3 columns.`

DataFrame.size	to display a tuple representing the dimensionality of the DataFrame	`>>> ForestAreaDF.size`
		`12`
		`This means the ForestAreaDF has 12 values in it.`

DataFrame.T	to transpose the DataFrame. Means, row indices and column labels of the DataFrame replace each other's position	`>>> ForestAreaDF.T`
		`      GeoArea  VeryDense  ModeratelyDense OpenForest`
		`Assam 78438.0   2797.00         10192.00    15116.00`
		`Kerala38852.0   1663.00          9407.00     9251.00`
		`Delhi  1483.0      6.72            56.24      129.45`

DataFrame.head(n)	to display the first n rows in the DataFrame	`>>> ForestAreaDF.head(2)`
		`              Assam  Kerala    Delhi`
		`GeoArea       78438   38852  1483.00`
		`VeryDense      2797    1663     6.72`
		
		displays the first 2 rows of the DataFrame ForestAreaDF.If the parameter n is not specified by default it gives the first 5 rows of the DataFrame.

DataFrame.tail(n)	to display the last n rows in the DataFrame	`>>> ForestAreaDF.tail(2)`
		`                 Assam  Kerala   Delhi`
		`ModeratelyDense  10192    9407   56.24`
		`OpenForest       15116    9251  129.45`
		
		displays the last 2 rows of the DataFrame ForestAreaDF.If the parameter n is not specified by default it gives the last 5 rows of the DataFrame.

	to returns the value True if DataFrame is empty and False otherwise	`>>> ForestAreaDF.empty`
		`False`
		`>>> df=pd.DataFrame() #Create an empty dataFrame`
		`>>> df.empty`
		`True`

2.4 Importing and Exporting Data between CSV Files and DataFrames

We can create a DataFrame by importing data from CSV files where values are separated by commas. Similarly, we can also store or export data in a DataFrame as a .csv file.

2.4.1 Importing a CSV file to a DataFrame

Let us assume that we have the following data in a csv file named ResultData.csv stored in the folder C:/NCERT. In order to practice the code while we progress, you are suggested to create this csv file using a spreadsheet and save in your computer.

```
RollNo     Name        Eco   Maths
1          Arnab       18    57
2          Kritika     23    45
3          Divyam      51    37
4          Vivaan      40    60
5          Aaroosh     18    27
```

NOTES

We can load the data from the ResultData.csv file into a DataFrame, say marks using Pandas read_csv() function as shown below:

```
>>> marks = pd.read_csv("C:/NCERT/ResultData.
csv",sep =",", header=0)
>>> marks
   RollNo      Name   Eco        Maths
0       1     Arnab    18        57
1       2   Kritika    23        45
2       3    Divyam    51        37
3       4    Vivaan    40        60
4       5   Aaroosh    18        27
```

- The first parameter to the read_csv() is the name of the comma separated data file along with its path.
- The parameter sep specifies whether the values are separated by comma, semicolon, tab, or any other character. The default value for sepis a space.
- The parameter header specifies the number of the row whose values are to be used as the column names. It also marks the start of the data to be fetched. header=0 implies that column names are inferred from the first line of the file. By default, header=0.

We can exclusively specify column names using the parameter names while creating the DataFrame using the read_csv() function. For example, in the following statement, names parameter is used to specify the labels for columns of the DataFrame marks1:

```
>>> marks1 = pd.read_csv("C:/NCERT/ResultData1.
csv",sep=",",
         names=['RNo','StudentName', 'Sub1',
'Sub2'])
>>> marks1
   RNo  StudentName  Sub1  Sub2
0    1        Arnab    18    57
1    2      Kritika    23    45
2    3       Divyam    51    37
3    4       Vivaan    40    60
4    5      Aaroosh    18    27
```

2.4.2 Exporting a DataFrame to a CSV file

We can use the to_csv() function to save a DataFrame to a text or csv file. For example, to save the DataFrame ResultDF created in the previous section; we can use the following statement:

```
>>> ResultDF
```

```
        Arnab  Ramit  Samridhi  Riya  Mallika
Maths      90     92        89    81       94
Science    91     81        91    71       95
Hindi      97     96        88    67       99

>>> ResultDF.to_csv(path_or_buf='C:/NCERT/
resultout.csv', sep=',')
```

This creates a file by the name resultout.csv in the folder C:/NCERT on the hard disk. When we open this file in any text editor or a spreadsheet, we will find the above data along with the row labels and the column headers, separated by comma.

A Comma-Separated Value (CSV) file is a text file where values are separated by comma. Each line represents a record (row). Each row consists of one or more fields (columns). They can be easily handled through a spreadsheet application.

In case we do not want the column names to be saved to the file we may use the parameter header=False. Another parameter index=False is used when we do not want the row labels to be written to the file on disk. For example:

```
>>> ResultDF.to_csv( 'C:/NCERT/resultonly.txt',
sep = '@', header = False, index= False)

If we open the file resultonly.txt, we will find
the following contents:

90@92@89@81@94
91@81@91@71@95
97@96@88@67@99
```

2.5 Pandas Series Vs NumPy ndarray

Pandas supports non-unique index values. If an operation that does not support duplicate index values is attempted, an exception will be raised at that time.

A basic difference between Series and ndarray is that operations between Series automatically align the data based on the label. Thus, we can write computations without considering whether all Series involved have the same label or not.

The result of an operation between unaligned Series (i.e. where the corresponding labels of the series are not the same or are not in the same order) will have the union of the indexes involved. If a label is not found in one Series or the other, the result will be marked as missing NaN. Being able to write code without doing any explicit data alignment grants immense freedom and flexibility in interactive data analysis and research.

Think and Reflect

What are the other parameters that can be used with read_csv() function? You may explore from https://pandas.pydata.org.

Think and Reflect

Besides comma, what are the other allowed characters that can be used as a separator while creating a CSV file frmo a DataFrame?

Table 2.5 Difference between Pandas Series and NumPy Arrays

Pandas Series	NumPy Arrays
In series we can define our own labeled index to access elements of an array. These can be numbers or letters.	NumPy arrays are accessed by their integer position using numbers only.
The elements can be indexed in descending order also.	The indexing starts with zero for the first element and the index is fixed.
If two series are not aligned, NaN or missing values are generated.	There is no concept of NaN values and if there are no matching values in arrays, alignment fails.
Series require more memory.	NumPy occupies lesser memory.

Summary

- NumPy, Pandas and Matplotlib are Python libraries for scientific and analytical use.
- pip install pandas is the command to install Pandas library.
- A data structure is a collection of data values and the operations that can be applied to that data. It enables efficient storage, retrieval and modification to the data.
- Two main data structures in Pandas library are Series and DataFrame. To use these data structures, we first need to import the Pandas library.
- A Series is a one-dimensional array containing a sequence of values. Each value has a data label associated with it also called its index.
- The two common ways of accessing the elements of a series are Indexing and Slicing.
- There are two types of indexes: positional index and labelled index. Positional index takes an integer value that corresponds to its position in the series starting from 0, whereas labelled index takes any user-defined label as index.
- When positional indices are used for slicing, the value at end index position is excluded, i.e., only (end - start) number of data values of the series are extracted. However with labelled indexes the

value at the end index label is also included in the output.

- All basic mathematical operations can be performed on Series either by using the operator or by using appropriate methods of the Series object.
- While performing mathematical operations index matching is implemented and if no matching indexes are found during alignment, Pandas returns NaN so that the operation does not fail.
- A DataFrame is a two-dimensional labeled data structure like a spreadsheet. It contains rows and columns and therefore has both a row and column index.
- When using a dictionary to create a DataFrame, keys of the Dictionary become the column labels of the DataFrame. A DataFrame can be thought of as a dictionary of lists/ Series (all Series/columns sharing the same index label for a row).
- Data can be loaded in a DataFrame from a file on the disk by using Pandas read_csv function.
- Data in a DataFrame can be written to a text file on disk by using the pandas.DataFrame.to_csv() function.
- DataFrame.T gives the transpose of a DataFrame.
- Pandas haves a number of methods that support label based indexing but every label asked for must be in the index, or a KeyError will be raised.
- DataFrame.loc[ ] is used for label based indexing of rows in DataFrames.
- Pandas.DataFrame.append() method is used to merge two DataFrames.
- Pandas supports non-unique index values. Only if a particular operation that does not support duplicate index values is attempted, an exception is raised at that time.
- The basic difference between Pandas Series and NumPy ndarray is that operations between Series automatically align the data based on labels. Thus, we can write computations without considering whether all Series involved have the same label or not whereas in case of ndarrays it raises an error.

Exercise

1. What is a Series and how is it different from a 1-D array, a list and a dictionary?
2. What is a DataFrame and how is it different from a 2-D array?
3. How are DataFrames related to Series?
4. What do you understand by the size of (i) a Series, (ii) a DataFrame?
5. Create the following Series and do the specified operations:
   a) EngAlph, having 26 elements with the alphabets as values and default index values.
   b) Vowels, having 5 elements with index labels 'a', 'e', 'i', 'o' and 'u' and all the five values set to zero. Check if it is an empty series.
   c) Friends, from a dictionary having roll numbers of five of your friends as data and their first name as keys.
   d) MTseries, an empty Series. Check if it is an empty series.
   e) MonthDays, from a numpy array having the number of days in the 12 months of a year. The labels should be the month numbers from 1 to 12.
6. Using the Series created in Question 5, write commands for the following:
   a) Set all the values of Vowels to 10 and display the Series.
   b) Divide all values of Vowels by 2 and display the Series.
   c) Create another series Vowels1 having 5 elements with index labels 'a', 'e', 'i', 'o' and 'u' having values [2,5,6,3,8] respectively.
   d) Add Vowels and Vowels1 and assign the result to Vowels3.
   e) Subtract, Multiply and Divide Vowels by Vowels1.
   f) Alter the labels of Vowels1 to ['A', 'E', 'I', 'O', 'U'].
7. Using the Series created in Question 5, write commands for the following:
   a) Find the dimensions, size and values of the Series EngAlph, Vowels, Friends, MTseries, MonthDays.
   b) Rename the Series MTseries as SeriesEmpty.
   c) Name the index of the Series MonthDays as monthno and that of Series Friends as Fname.

d) Display the 3rd and 2nd value of the Series Friends, in that order.

e) Display the alphabets 'e' to 'p' from the Series EngAlph.

f) Display the first 10 values in the Series EngAlph.

g) Display the last 10 values in the Series EngAlph.

h) Display the MTseries.

8. Using the Series created in Question 5, write commands for the following:

a) Display the names of the months 3 through 7 from the Series MonthDays.

b) Display the Series MonthDays in reverse order.

9. Create the following DataFrame Sales containing year wise sales figures for five sales persons in INR. Use the years as column labels, and sales person names as row labels.

	2014	2015	2016	2017
Madhu	100.5	12000	20000	50000
Kusum	150.8	18000	50000	60000
Kinshuk	200.9	22000	70000	70000
Ankit	30000	30000	100000	80000
Shruti	40000	45000	125000	90000

10. Use the DataFrame created in Question 9 above to do the following:

a) Display the row labels of Sales.

b) Display the column labels of Sales.

c) Display the data types of each column of Sales.

d) Display the dimensions, shape, size and values of Sales.

e) Display the last two rows of Sales.

f) Display the first two columns of Sales.

g) Create a dictionary using the following data. Use this dictionary to create a DataFrame Sales2.

	2018
Madhu	160000
Kusum	110000
Kinshuk	500000
Ankit	340000
Shruti	900000

h) Check if Sales2 is empty or it contains data.

NOTES

11. Use the DataFrame created in Question 9 above to do the following:

    a) Append the DataFrame Sales2 to the DataFrame Sales.

    b) Change the DataFrame Sales such that it becomes its transpose.

    c) Display the sales made by all sales persons in the year 2017.

    d) Display the sales made by Madhu and Ankit in the year 2017 and 2018.

    e) Display the sales made by Shruti 2016.

    f) Add data to Sales for salesman Sumeet where the sales made are [196.2, 37800, 52000, 78438, 38852] in the years [2014, 2015, 2016, 2017, 2018] respectively.

    g) Delete the data for the year 2014 from the DataFrame Sales.

    h) Delete the data for sales man Kinshuk from the DataFrame Sales.

    i) Change the name of the salesperson Ankit to Vivaan and Madhu to Shailesh.

    j) Update the sale made by Shailesh in 2018 to 100000.

    k) Write the values of DataFrame Sales to a comma separated file SalesFigures.csv on the disk. Do not write the row labels and column labels.

    l) Read the data in the file SalesFigures.csv into a DataFrame SalesRetrieved and Display it. Now update the row labels and column labels of SalesRetrieved to be the same as that of Sales.

Chapter 3

Data Handling using Pandas - II

> “We owe a lot to the Indians, who taught us how to count, without which no worthwhile scientific discovery could have been made.”
>
> — Albert Einstein

12149CH03

3.1 Introduction

As discussed in the previous chapter, Pandas is a well established Python Library used for manipulation, processing and analysis of data. We have already discussed the basic operations on Series and DataFrame like creating them and then accessing data from them. Pandas provides more powerful and useful functions for data analysis.

In this chapter, we will be working with more advanced features of DataFrame like sorting data, answering analytical questions using the data, cleaning data and applying different useful functions on the data. Below is the example data on which we will be applying the advanced features of Pandas.

Case Study

Let us consider the data of marks scored in unit tests held in school. For each unit test, the marks scored by all students of the class is recorded. Maximum marks are 25 in each subject. The subjects are Maths, Science. Social Studies (S.St.), Hindi, and English. For simplicity, we assume there are 4 students in the class and the table below shows their marks in Unit Test 1, Unit Test 2 and Unit Test 3. Table 3.1 shows this data.

Table 3.1 Case Study

Result						
Name/ Subjects	**Unit Test**	**Maths**	**Science**	**S.St.**	**Hindi**	**Eng**
Raman	1	22	21	18	20	21
Raman	2	21	20	17	22	24
Raman	3	14	19	15	24	23
Zuhaire	1	20	17	22	24	19
Zuhaire	2	23	15	21	25	15
Zuhaire	3	22	18	19	23	13
Aashravy	1	23	19	20	15	22
Aashravy	2	24	22	24	17	21
Aashravy	3	12	25	19	21	23
Mishti	1	15	22	25	22	22
Mishti	2	18	21	25	24	23
Mishti	3	17	18	20	25	20

Let us store the data in a DataFrame, as shown in Program 3.1:

Program 3-1 Store the Result data in a DataFrame called marksUT.

```
>>> import pandas as pd
>>> marksUT= {'Name':['Raman','Raman','Raman','Zuhaire','Zuhaire','Zu
haire', 'Ashravy','Ashravy','Ashravy','Mishti','Mishti','Mishti'],
            'UT':[1,2,3,1,2,3,1,2,3,1,2,3],
            'Maths':[22,21,14,20,23,22,23,24,12,15,18,17],
            'Science':[21,20,19,17,15,18,19,22,25,22,21,18],
            'S.St':[18,17,15,22,21,19,20,24,19,25,25,20],
            'Hindi':[20,22,24,24,25,23,15,17,21,22,24,25],
            'Eng':[21,24,23,19,15,13,22,21,23,22,23,20]
            }
>>> df=pd.DataFrame(marksUT)
>>> print(df)
```

```
         Name  UT  Maths  Science  S.St  Hindi  Eng
0       Raman   1     22       21    18     20   21
1       Raman   2     21       20    17     22   24
2       Raman   3     14       19    15     24   23
3     Zuhaire   1     20       17    22     24   19
4     Zuhaire   2     23       15    21     25   15
5     Zuhaire   3     22       18    19     23   13
6     Ashravy   1     23       19    20     15   22
7     Ashravy   2     24       22    24     17   21
8     Ashravy   3     12       25    19     21   23
9      Mishti   1     15       22    25     22   22
10     Mishti   2     18       21    25     24   23
11     Mishti   3     17       18    20     25   20
```

3.2 DESCRIPTIVE STATISTICS

Descriptive Statistics are used to summarise the given data. In other words, they refer to the methods which are used to get some basic idea about the data.

In this section, we will be discussing descriptive statistical methods that can be applied to a DataFrame. These are max, min, count, sum, mean, median, mode, quartiles, variance. In each case, we will consider the above created DataFrame df.

3.2.1 Calculating Maximum Values

DataFrame.max() is used to calculate the maximum values from the DataFrame, regardless of its data types. The following statement outputs the maximum value of each column of the DataFrame:

```
>>> print(df.max())
Name       Zuhaire      #Maximum value in name column
                        #(alphabetically)
UT               3      #Maximum value in column UT
Maths           24      #Maximum value in column Maths
Science         25      #Maximum value in column Science
S.St            25      #Maximum value in column S.St
Hindi           25      #Maximum value in column Hindi
Eng             24      #Maximum value in column Eng
dtype: object
```

If we want to output maximum value for the columns having only numeric values, then we can set the parameter numeric_only=True in the max() method, as shown below:

```
>>> print(df.max(numeric_only=True))
UT           3
Maths       24
Science     25
S.St        25
Hindi       25
Eng         24
dtype: int64
```

Program 3-2 Write the statements to output the maximum marks obtained in each subject in Unit Test 2.

```
>>> dfUT2 = df[df.UT == 2]
>>> print('\nResult of Unit Test 2:
\n\n',dfUT2)

Result of Unit Test 2:
       Name  UT  Maths  Science  S.St  Hindi  Eng
1     Raman   2     21       20    17     22   24
4   Zuhaire   2     23       15    21     25   15
7   Ashravy   2     24       22    24     17   21
10   Mishti   2     18       21    25     24   23
```

The output of Program 3.2 can also be achieved using the following statements

```
>>> dfUT2=df[df
['UT']==2].max
(numeric_only=True)
>>> print(dfUT2)
```

```
>>> print('\nMaximum Mark obtained in
Each Subject in Unit Test 2: \n\n',dfUT2.
max(numeric_only=True))

Maximum Mark obtained in Each Subject in Unit
Test 2:

UT           2
Maths       24
Science     22
S.St        25
Hindi       25
Eng         24
dtype: int64
```

By default, the max() method finds the maximum value of each column (which means, axis=0). However, to find the maximum value of each row, we have to specify axis = 1 as its argument.

```
#maximum marks for each student in each unit
test among all the subjects
```

```
>>> df.max(axis=1)

0     22
1     24
2     24
3     24
4     25
5     23
6     23
7     24
8     25
9     25
10    25
11    25
dtype: int64
```

Note: In most of the python function calls, axis = 0 refers to row wise operations and axis = 1 refers to column wise operations. But in the call of max(), axis = 1 gives row wise output and axis = 0 (default case) gives column-wise output. Similar is the case with all statistical operations discussed in this chapter.

3.2.2 Calculating Minimum Values

DataFrame.min() is used to display the minimum values from the DataFrame, regardless of the data types. That is, it shows the minimum value of each column or row. The following line of code output the minimum value of each column of the DataFrame:

```
>>> print(df.min())
Name        Ashravy
UT                1
Maths            12
Science          15
S.St             15
Hindi            15
Eng              13
dtype: object
```

Program 3-3 Write the statements to display the minimum marks obtained by a particular student 'Mishti' in all the unit tests for each subject.

```
>>> dfMishti = df.loc[df.Name == 'Mishti']
```

Notes

```
>>> print('\nMarks obtained by Mishti in all
the Unit Tests \n\n',dfMishti)

Marks obtained by Mishti in all the Unit Tests
       Name    UT  Maths  Science  S.St  Hindi  Eng
9    Mishti    1     15       22    25     22   22
10   Mishti    2     18       21    25     24   23
11   Mishti    3     17       18    20     25   20

>>> print('\nMinimum Marks obtained by
Mishti in each subject across the unit
tests\n\n', dfMishti[['Maths','Science','S.
St','Hindi','Eng']].min())
```

Minimum Marks obtained by Mishti in each subject across the unit tests:

```
Maths      15
Science    18
S.St       20
Hindi      22
Eng        20
dtype: int64
```

Note: Since we did not want to output the min value of column UT, we mentioned all the other column names for which minimum is to be calculated.

The output of Program 3.3 can also be achieved using the following statements
>>> dfMishti=df[['Maths','Science','S.St','Hindi','Eng']][df.Name == 'Mishti'].min()
>>> print(dfMishti)

3.2.3 Calculating Sum of Values

DataFrame.sum() will display the sum of the values from the DataFrame regardless of its datatype. The following line of code outputs the sum of each column of the DataFrame:

```
>>> print(df.sum())
Name
RamanRamanRamanZuhaireZuhaireZuhaireAshravyAsh...
UT                                              24
Maths                                          231
Science                                        237
S.St                                           245
Hindi                                          262
Eng                                            246
dtype: object
```

We may not be interested to sum text values. So, to print the sum of a particular column, we need to specify the column name in the call to function sum. The following statement prints the total marks of subject mathematics:

```
>>> print(df['Maths'].sum())
231
```

To calculate total marks of a particular student, the name of the student needs to be specified.

Program 3-4 Write the python statement to print the total marks secured by raman in each subject.

```
>>> dfRaman=df[df['Name']=='Raman']
>>> print(“Marks obtained by Raman in each test
are:\n”, dfRaman)
Marks obtained by Raman in each test are:
    Name  UT  Maths  Science  S.St  Hindi  Eng
0  Raman   1     22       21    18     20   21
1  Raman   2     21       20    17     22   24
2  Raman   3     14       19    15     24   23

>>> dfRaman[['Maths','Science','S.
St','Hindi','Eng']].sum()
Maths      57
Science    60
S.St       50
Hindi      66
Eng        68
dtype: int64
#To print total marks scored by Raman in all
subjects in each Unit Test
>>> dfRaman[['Maths','Science','S.
St','Hindi','Eng']].sum(axis=1)
0    102
1    104
2     95
dtype: int64
```

3.2.4 Calculating Number of Values

DataFrame.count() will display the total number of values for each column or row of a DataFrame. To count the rows we need to use the argument axis=1 as shown in the Program 3.5 below.

Think and Reflect

Can you write a shortened code to get the output of Program 3.4?

Activity 3.1

Write the python statements to print the sum of the english marks scored by Mishti.

```
>>> print(df.count())

Name       12
UT         12
Maths      12
Science    12
S.St       12
Hindi      12
Eng        12
dtype: int64
```

Program 3-5 Write a statement to count the number of values in a row.

```
>>> df.count(axis=1)
0     7
1     7
2     7
3     7
4     7
5     7
6     7
7     7
8     7
9     7
10    7
11    7
dtype: int64
```

3.2.5 Calculating Mean

DataFrame.mean() will display the mean (average) of the values of each column of a DataFrame. It is only applicable for numeric values.

```
>>> df.mean()
UT          2.5000
Maths      18.6000
Science    19.8000
S.St       20.0000
Hindi      21.3125
Eng        19.8000
dtype: float64
```

Program 3-6 Write the statements to get an average of marks obtained by Zuhaire in all the Unit Tests.

```
>>> dfZuhaireMarks = dfZuhaire.
loc[:,'Maths':'Eng']
>>> print("Slicing of the DataFrame to get only
the marks\n", dfZuhaireMarks)

Slicing of the DataFrame to get only the marks
    Maths  Science  S.St  Hindi  Eng
3     20       17     22     24   19
4     23       15     21     25   15
5     22       18     19     23   13

>>> print("Average of marks obtained by
Zuhaire in all Unit Tests \n", dfZuhaireMarks.
mean(axis=1))

Average of marks obtained by Zuhaire in all
Unit Tests
3    20.4
4    19.8
5    19.0
dtype: float64
```

In the above output, 20.4 is the average of marks obtained by Zuhaire in Unit Test 1. Similarly, 19.8 and 19.0 are the average of marks in Unit Test 2 and 3 respectively.

Think and Reflect

Try to write a short code to get the above output. Remember to print the relevant headings of the output.

3.2.6 Calculating Median

DataFrame.Median() will display the middle value of the data. This function will display the median of the values of each column of a DataFrame. It is only applicable for numeric values.

```
>>> print(df.median())

UT          2.5
Maths      19.0
Science    20.0
S.St       19.5
Hindi      21.5
Eng        21.0
dtype: float64
```

Program 3-7 Write the statements to print the median marks of mathematics in UT1.

```
>>> dfMaths=df['Maths']
>>> dfMathsUT1=dfMaths[df.UT==1]
>>> print("Displaying the marks scored in
Mathematics in UT1\n",dfMathsUT1)

Displaying the marks of UT1, subject
Mathematics
0    22
3    20
6    23
9    15
Name: Maths, dtype: int64

>>> dfMathMedian=dfMathsUT1.median()
>>> print("Displaying the median of Mathematics
in UT1\n”,dfMathMedian)

Displaying the median of Mathematics in UT1
21.0
```

Here, the number of values are even in number so two middle values are there i.e. 20 and 22. Hence, Median is the average of 20 and 22.

Activity 3.2

Find the median of the values of the rows of the DataFrame.

3.2.7 Calculating Mode

DateFrame.mode() will display the mode. The mode is defined as the value that appears the most number of times in a data. This function will display the mode of each column or row of the DataFrame. To get the mode of Hindi marks, the following statement can be used.

```
>>> df['Hindi']
0     20
1     22
2     24
3     24
4     25
5     23
6     15
7     17
8     21
9     22
10    24
11    25
Name: Hindi, dtype: int64
>>> df['Hindi'].mode()
```

Activity 3.3

Calculate the mode of marks scored in Maths.

```
0    24
dtype: int64
```

Note that three students have got 24 marks in Hindi subject while two students got 25 marks, one student got 23 marks, two students got 22 marks, one student each got 21, 20, 15, 17 marks.

3.2.8 Calculating Quartile

Dataframe.quantile() is used to get the quartiles. It will output the quartile of each column or row of the DataFrame in four parts i.e. the first quartile is 25% (parameter q = .25), the second quartile is 50% (Median), the third quartile is 75% (parameter q = .75). By default, it will display the second quantile (median) of all numeric values.

```
>>> df.quantile() # by default, median is the
output
UT          2.0
Maths      20.5
Science    19.5
S.St       20.0
Hindi      22.5
Eng        21.5
Name: 0.5, dtype: float64

>>> df.quantile(q=.25)
UT          1.00
Maths      16.50
Science    18.00
S.St       18.75
Hindi      20.75
Eng        19.75
Name: 0.25, dtype: float64

>>> df.quantile(q=.75)
UT          3.00
Maths      22.25
Science    21.25
S.St       22.50
Hindi      24.00
Eng        23.00
Name: 0.75, dtype: float64
```

NOTES

Program 3-8 Write the statement to display the first and third quartiles of all subjects.

```
>>> dfSubject=df[['Maths','Science','S.
St','Hindi','Eng']]
>>> print("Marks of all the subjects:\
n",dfSubject)

Marks of all the subjects:
    Maths  Science  S.St  Hindi  Eng
0      22       21    18     20   21
1      21       20    17     22   24
2      14       19    15     24   23
3      20       17    22     24   19
4      23       15    21     25   15
5      22       18    19     23   13
6      23       19    20     15   22
7      24       22    24     17   21
8      12       25    19     21   23
9      15       22    25     22   22
10     18       21    25     24   23
11     17       18    20     25   20

>>> dfQ=dfSubject.quantile([.25,.75])
>>> print("First and third quartiles of all the
subjects:\n",dfQ)

First and third quartiles of all the subjects:
      Maths  Science   S.St  Hindi    Eng
0.25  16.50    18.00  18.75  20.75  19.75

0.75  22.25    21.25  22.50  24.00  23.00
```

3.2.9 Calculating Variance

DataFrame.var() is used to display the variance. It is the average of squared differences from the mean.

Activity 3.4

Find the variance and standard deviation of the following scores on an exam: 92, 95, 85, 80, 75, 50.

```
>>> df[['Maths','Science','S.
St','Hindi','Eng']].var()


Maths      15.840909
Science     7.113636
S.St        9.901515
```

```
Hindi       9.969697
Eng        11.363636
dtype: float64
```

3.2.10 Calculating Standard Deviation

DataFrame.std() returns the standard deviation of the values. Standard deviation is calculated as the square root of the variance.

```
>>> df[['Maths','Science','S.
St','Hindi','Eng']].std()

Maths      3.980064
Science    2.667140
S.St       3.146667
Hindi      3.157483
Eng        3.370999
dtype: float64
```

DataFrame.describe() function displays the descriptive statistical values in a single command. These values help us describe a set of data in a DataFrame.

```
>>> df.describe()
           UT      Maths     Science      S.St      Hindi        Eng
count 12.000000 12.000000 12.00000 12.000000 12.000000 12.000000
mean   2.000000 19.250000 19.75000 20.416667 21.833333 20.500000
std    0.852803  3.980064  2.66714  3.146667  3.157483  3.370999
min    1.000000 12.000000 15.00000 15.000000 15.000000 13.000000
25%    1.000000 16.500000 18.00000 18.750000 20.750000 19.750000
50%    2.000000 20.500000 19.50000 20.000000 22.500000 21.500000
75%    3.000000 22.250000 21.25000 22.500000 24.000000 23.000000
max    3.000000 24.000000 25.00000 25.000000 25.000000 24.000000
```

3.3 Data Aggregations

Aggregation means to transform the dataset and produce a single numeric value from an array. Aggregation can be applied to one or more columns together. Aggregate functions are max(),min(), sum(), count(), std(), var().

```
>>> df.aggregate('max')

Name    Zuhaire # displaying the maximum of Name
as well
UT            3
Maths        24
```

NOTES

```
Science         25
S.St            25
Hindi           25
Eng             24
dtype: object

#To use multiple aggregate functions in a
single statement
>>> df.aggregate(['max','count'])

     Name    UT   Maths Science S.St  Hindi  Eng
max  Zuhaire  3     24      25   25     25   24
count  12    12     12      12   12     12   12

>>> df['Maths'].aggregate(['max','min'])
max    24
min    12
Name: Maths, dtype: int64
```

Note: We can also use the parameter axis with aggregate function. By default, the value of axis is zero, means columns.

```
#Using the above statement with axis=0 gives
the same result
>>> df['Maths'].aggregate(['max','min'],axis=0)
max    24
min    12
Name: Maths, dtype: int64

#Total marks of Maths and Science obtained by
each student.
#Use sum() with axis=1 (Row-wise summation)
>>> df[['Maths','Science']].
aggregate('sum',axis=1)
0     43
1     41
2     33
3     37
4     38
5     40
6     42
7     46
8     37
9     37
10    39
11    35
dtype: int64
```

3.4 Sorting a DataFrame

Sorting refers to the arrangement of data elements in a specified order, which can either be ascending or descending. Pandas provide sort_values() function to sort the data values of a DataFrame. The syntax of the function is as follows:

```
DataFrame.sort_values(by, axis=0, ascending=True)
```

Here, a column list (by), axis arguments (0 for rows and 1 for columns) and the order of sorting (ascending = False or True) are passed as arguments. By default, sorting is done on row indexes in ascending order.

Consider a scenario, where the teacher is interested in arranging a list according to the names of the students or according to marks obtained in a particular subject. In such cases, sorting can be used to obtain the desired results. Following is the python code for sorting the data in the DataFrame created at program 3.1.

To sort the entire data on the basis of attribute 'Name', we use the following command:

```
#By default, sorting is done in ascending order.
>>> print(df.sort_values(by=['Name']))

         Name  UT  Maths  Science  S.St  Hindi  Eng
6     Ashravy   1     23       19    20     15   22
7     Ashravy   2     24       22    24     17   21
8     Ashravy   3     12       25    19     21   23
9      Mishti   1     15       22    25     22   22
10     Mishti   2     18       21    25     24   23
11     Mishti   3     17       18    20     25   20
0       Raman   1     22       21    18     20   21
1       Raman   2     21       20    17     22   24
2       Raman   3     14       19    15     24   23
3     Zuhaire   1     20       17    22     24   19
4     Zuhaire   2     23       15    21     25   15
5     Zuhaire   3     22       18    19     23   13
```

Now, to obtain sorted list of marks scored by all students in Science in Unit Test 2, the following code can be used:

```
# Get the data corresponding to Unit Test 2
>>> dfUT2 = df[df.UT == 2]
# Sort according to ascending order of marks in
Science
```

```
>>> print(dfUT2.sort_values(by=['Science']))

       Name  UT  Maths  Science  S.St  Hindi  Eng
4   Zuhaire   2     23       15    21     25   15
1     Raman   2     21       20    17     22   24
10   Mishti   2     18       21    25     24   23
7   Ashravy   2     24       22    24     17   21
```

Program 3-9 Write the statement which will sort the marks in English in the DataFrame df based on Unit Test 3, in descending order.

```
# Get the data corresponding to Unit Test 3
>>> dfUT3 = df[df.UT == 3]
# Sort according to descending order of marks in
Science
>>> print(dfUT3.sort_values(by=['Eng'],ascending=F
alse))

      Name  UT  Maths  Science  S.St  Hindi  Eng
2     Raman  3     14       19    15     24   23
8   Ashravy  3     12       25    19     21   23
11   Mishti  3     17       18    20     25   20
5   Zuhaire  3     22       18    19     23   13
```

A DataFrame can be sorted based on multiple columns. Following is the code of sorting the DataFrame df based on marks in Science in Unit Test 3 in ascending order. If marks in Science are the same, then sorting will be done on the basis of marks in Hindi.

```
# Get the data corresponding to marks in Unit Test
3
>>> dfUT3 = df[df.UT == 3]
# Sort the data according to Science and then
according to Hindi
>>> print(dfUT3.sort_
values(by=['Science','Hindi']))

       Name  UT  Maths  Science  S.St  Hindi  Eng
5   Zuhaire   3     22       18    19     23   13
11   Mishti   3     17       18    20     25   20
2     Raman   3     14       19    15     24   23
8   Ashravy   3     12       25    19     21   23
```

Here, we can see that the list is sorted on the basis of marks in Science. Two students namely, Zuhaire and Mishti have equal marks (18) in Science. Therefore for them, sorting is done on the basis of marks in Hindi.

Notes

3.5 GROUP BY Functions

In pandas, DataFrame.GROUP BY() function is used to split the data into groups based on some criteria. Pandas objects like a DataFrame can be split on any of their axes. The GROUP BY function works based on a split-apply-combine strategy which is shown below using a 3-step process:

Step 1: Split the data into groups by creating a GROUP BY object from the original DataFrame.

Step 2: Apply the required function.

Step 3: Combine the results to form a new DataFrame.

To understand this better, let us consider the data shown in the diagram given below. Here, we have a two-column DataFrame (key, data). We need to find the sum of the data column for a particular key, i.e. sum of all the data elements with key A, B and C, respectively. To do so, we first split the entire DataFrame into groups by key column. Then, we apply the sum function on the respective groups. Finally, we combine the results to form a new DataFrame that contains the desired result.

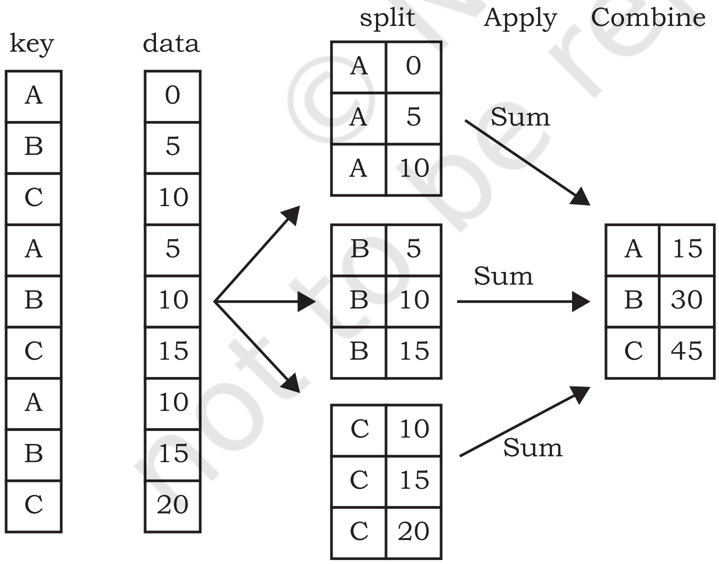


Figure 3.1: A DataFrame with two columns

The following statements show how to apply GROUP BY() function on our DataFrame df created at Program 3.1:

```
#Create a GROUP BY Name of the student from
DataFrame df
>>> g1=df.GROUP BY('Name')
```

NOTES

```
#Displaying the first entry from each group
>>> g1.first()
          UT  Maths  Science  S.St  Hindi  Eng
Name
Ashravy    1     23       19    20     15   22
Mishti     1     15       22    25     22   22
Raman      1     22       21    18     20   21
Zuhaire    1     20       17    22     24   19

#Displaying the size of each group
>>> g1.size()
Name
Ashravy    3
Mishti     3
Raman      3
Zuhaire    3
dtype: int64

#Displaying group data, i.e., group_name, row
indexes corresponding to the group and their
data type
>>> g1.groups
{'Ashravy': Int64Index([6, 7, 8],
dtype='int64'),
 'Mishti': Int64Index([9, 10, 11],
dtype='int64'),
 'Raman': Int64Index([0, 1, 2], dtype='int64'),
 'Zuhaire': Int64Index([3, 4, 5],
dtype='int64')}


#Printing data of a single group
>>> g1.get_group('Raman')
   UT  Maths  Science  S.St  Hindi  Eng
0   1     22       21    18     20   21
1   2     21       20    17     22   24
2   3     14       19    15     24   23

#Grouping with respect to multiple attributes
#Creating a GROUP BY Name and UT

>>> g2=df.GROUP BY(['Name', 'UT'])

>>> g2.first()
```

```
                 Maths  Science  S.St  Hindi  Eng
Name     UT
Ashravy  1          23       19    20     15   22
         2          24       22    24     17   21
         3          12       25    19     21   23
Mishti   1          15       22    25     22   22
         2          18       21    25     24   23
         3          17       18    20     25   20
Raman    1          22       21    18     20   21
         2          21       20    17     22   24
         3          14       19    15     24   23
Zuhaire  1          20       17    22     24   19
         2          23       15    21     25   15
         3          22       18    19     23   13
```

The above statements show how we create groups by splitting a DataFrame using GROUP BY(). Next step is to apply functions over the groups just created. This is done using Aggregation.

Aggregation is a process in which an aggregate function is applied on each group created by GROUP BY(). It returns a single aggregated statistical value corresponding to each group. It can be used to apply multiple functions over an axis. Be default, functions are applied over columns. Aggregation can be performed using agg() or aggregate() function.

```
#Calculating average marks scored by all
students in each subject for each UT
>>> df.GROUP BY(['UT']).aggregate('mean')

    Maths  Science   S.St  Hindi    Eng
UT
1   20.00    19.75  21.25  20.25  21.00
2   21.50    19.50  21.75  22.00  20.75
3   16.25    20.00  18.25  23.25  19.75

#Calculate average marks scored in Maths in
each UT
>>> group1=df.GROUP BY(['UT'])
>>> group1['Maths'].aggregate('mean')
UT
1    20.00
2    21.50
3    16.25
Name: Maths, dtype: float64
```

Program 3-10 Write the python statements to print the mean, variance, standard deviation and quartile of the marks scored in Mathematics by each student across the UTs.

```
>>> df.GROUP BY(by='Name')['Maths'].agg(['mean','v
ar','std','quantile'])

                mean         var       std       quantile
Name
Ashravy 19.666667   44.333333   6.658328      23.0
Mishti  16.666667    2.333333   1.527525      17.0
Raman   19.000000   19.000000   4.358899      21.0
Zuhaire21.666667   2.333333   1.527525      22.0
```

Activity 3.5

Write the python statements to print average marks in Science by all the students in each UT.

3.6 Altering the Index

We use indexing to access the elements of a DataFrame. It is used for fast retrieval of data. By default, a numeric index starting from 0 is created as a row index, as shown below:

```
>>> df                        #With default Index
        Name   UT  Maths  Science  S.St  Hindi  Eng
0      Raman    1     22       21    18     20   21
1      Raman    2     21       20    17     22   24
2      Raman    3     14       19    15     24   23
3    Zuhaire    1     20       17    22     24   19
4    Zuhaire    2     23       15    21     25   15
5    Zuhaire    3     22       18    19     23   13
6    Ashravy    1     23       19    20     15   22
7    Ashravy    2     24       22    24     17   21
8    Ashravy    3     12       25    19     21   23
9     Mishti    1     15       22    25     22   22
10    Mishti    2     18       21    25     24   23
11    Mishti    3     17       18    20     25   20
```

Here, the integer number in the first column starting from 0 is the index. However, depending on our requirements, we can select some other column to be the index or we can add another index column.

When we slice the data, we get the original index which is not continuous, e.g. when we select marks of all students in Unit Test 1, we get the following result:

```
>>> dfUT1 = df[df.UT == 1]
>>> print(dfUT1)
```

```
      Name  UT  Maths  Science  S.St  Hindi  Eng
0    Raman   1     22       21    18     20   21
3  Zuhaire   1     20       17    22     24   19
6  Ashravy   1     23       19    20     15   22
9   Mishti   1     15       22    25     22   22
```

Notice that the first column is a non-continuous index since it is slicing of original data. We create a new continuous index alongside this using the reset_index() function, as shown below:

```
>>> dfUT1.reset_index(inplace=True)
>>> print(dfUT1)
   index     Name  UT  Maths  Science  S.St  Hindi  Eng
0      0    Raman   1     22       21    18     20   21
1      3  Zuhaire   1     20       17    22     24   19
2      6  Ashravy   1     23       19    20     15   22
3      9   Mishti   1     15       22    25     22   22
```

A new continuous index is created while the original one is also intact. We can drop the original index by using the drop function, as shown below:

```
>>> dfUT1.drop(columns=[‘index’],inplace=True)
>>> print(dfUT1)

      Name  UT  Maths  Science  S.St  Hindi  Eng
0    Raman   1     22       21    18     20   21
1  Zuhaire   1     20       17    22     24   19
2  Ashravy   1     23       19    20     15   22
3   Mishti   1     15       22    25     22   22
```

We can change the index to some other column of the data.

```
>>> dfUT1.set_index('Name',inplace=True)
>>> print(dfUT1)
         UT  Maths  Science  S.St  Hindi  Eng
Name
Raman     1     22       21    18     20   21
Zuhaire   1     20       17    22     24   19
Ashravy   1     23       19    20     15   22
Mishti    1     15       22    25     22   22
```

We can revert back to previous index by using following statement:

```
>>> dfUT1.reset_index('Name', inplace = True)
>>> print(dfUT1)
```

```
       Name  UT  Maths  Science  S.St  Hindi  Eng
0     Raman   1     22       21    18     20   21
1   Zuhaire   1     20       17    22     24   19
2   Ashravy   1     23       19    20     15   22
3    Mishti   1     15       22    25     22   22
```

3.7 Other DataFrame Operations

In this section, we will learn more techniques and functions that can be used to manipulate and analyse data in a DataFrame.

3.7.1 Reshaping Data

The way a dataset is arranged into rows and columns is referred to as the shape of data. Reshaping data refers to the process of changing the shape of the dataset to make it suitable for some analysis problems. The example given in the below section explains the utility of reshaping the data.

For reshaping data, two basic functions are available in Pandas, pivot and pivot_table. This section covers them in detail.

(A) Pivot

The pivot function is used to reshape and create a new DataFrame from the original one. Consider the following example of sales and profit data of four stores: S1, S2, S3 and S4 for the years 2016, 2017 and 2018.

Example 3.1

```
>>> import pandas as pd

>>> data={'Store':['S1','S4','S3','S1','S2','S3
','S1','S2','S3'],  'Year':[2016,2016,2016,2017
,2017,2017,2018,2018,2018],
'Total_sales(Rs)':[12000,330000,420000,
20000,10000,450000,30000, 11000,89000],
'Total_profit(
Rs)':[1100,5500,21000,32000,9000,45000,3000,
1900,23000]
}

>>> df=pd.DataFrame(data)
>>> print(df)
```

```
   Store    Year   Total_sales(Rs)    Total_profit(Rs)
0    S1     2016         12000                1100
1    S4     2016        330000                5500
2    S3     2016        420000               21000
```

```
3    S1     2017            20000               32000
4    S2     2017            10000                9000
5    S3     2017           450000               45000
6    S1     2018            30000                3000
7    S2     2018            11000                1900
8    S3     2018            89000               23000
```

Let us try to answer the following queries on the above data.

1) What was the total sale of store S1 in all the years? Python statements to perform this task will be as follows:

```
    # will get the data related to store S1
>>> S1df = df[df.Store==’S1’]
#find the total of sales for Store S1
>>> S1df[‘Total_sales(Rs)’].sum()
62000
```

2) What is the maximum sale value by store S3 in any year?

```
#will get the data related to store S3
>>> S3df = df[df.Store==’S3’]
#find the maximum sale for Store S3
>>> S3df[‘Total_sales(Rs)’].max()
450000
```

3) Which store had the maximum total sale in all the years?

```
>>> S1df = df[df.Store=='S1']
>>> S2df=df[df.Store == 'S2']
>>> S3df = df[df.Store=='S3']
>>> S4df = df[df.Store=='S4']
>>> S1total = S1df['Total_sales(Rs)'].sum()
>>> S2total = S2df['Total_sales(Rs)'].sum()
   >>> S3total = S3df['Total_sales(Rs)'].sum()
   >>> S4total = S4df['Total_sales(Rs)'].sum()
 >>> max(S1total,S2total,S3total,S4total)
 959000
```

Notice that we have to slice the data corresponding to a particular store and then answer the query. Now, let us reshape the data using pivot and see the difference.

```
>>>
pivot1=df.pivot(index='Store',columns='Year',va
lues='Total_sales(Rs)')
```

Here, Index specifies the columns that will be acting as an index in the pivot table, columns specifies the new columns for the pivoted data and values specifies columns whose values will be displayed. In this particular case, store names will act as index, year will be the headers for columns and sales value will be displayed as values of the pivot table.

```
>>> print(pivot1)

Year        2016      2017     2018
Store
S1       12000.0   20000.0  30000.0
S2           NaN   10000.0  11000.0
S3      420000.0  450000.0  89000.0
S4      330000.0       NaN      NaN
```

As can be seen above, the value of Total_sales (Rs) for every row in the original table has been transferred to the new table: pivot1, where each row has data of a store and each column has data of a year. Those cells in the new pivot table which do not have a matching entry in the original one are filled with NaN. For instance, we did not have values corresponding to sales of Store S2 in 2016, thus the appropriate cell in pivot1 is filled with NaN.

Now the python statements for the above queries will be as follows:

1) What was the total sale of store S1 in all the years?

```
>>> pivot1.loc[‘S1’].sum()
```

2) What is the maximum sale value by store S3 in any year?

```
>>> pivot1.loc[‘S3’].max()
```

3) Which store had the maximum total sale?

```
>>> S1total = pivot1.loc['S1'].sum()
>>> S2total = pivot1.loc['S2'].sum()
>>> S3total = pivot1.loc['S3'].sum()
>>> S4total = pivot1.loc['S4'].sum()
>>> max(S1total,S2total,S3total,S4total)
```

We can notice that reshaping has transformed the structure of the data, which makes it more readable and easy to analyse the data.

Activity 3.6

Consider the data of unit test marks given at program 3.1, write the python statements to print name wise UT marks in mathematics.

(B) Pivoting by Multiple Columns

For pivoting by multiple columns, we need to specify multiple column names in the values parameter of pivot() function. If we omit the values parameter, it will display the pivoting for all the numeric values.

```
>>> pivot2=df.pivot(index='Store',columns='Year
',values=['Total_sales(Rs)','Total_profit(Rs)'])

>>> print(pivot2)
                Total_sales(Rs)                          Total_profit(Rs)
Year        2016       2017      2018               2016       2017      2018
Store
S1       12000.0    20000.0   30000.0             1100.0    32000.0    3000.0
S2           NaN    10000.0   11000.0                NaN     9000.0    1900.0
S3      330000.0        NaN       NaN             5500.0        NaN       NaN
```

Let us consider another example, where suppose we have stock data corresponding to a store as:

```
>>> data={'Item':['Pen','Pen','Pencil','Pencil'
,'Pen','Pen'],
'Color':['Red','Red','Black','Black','Blue','B
lue'],
'Price(Rs)':[10,25,7,5,50,20],
'Units_in_stock':[50,10,47,34,55,14]
}
>>> df=pd.DataFrame(data)
>>> print(df)

     Item  Color  Price(Rs)  Units_in_stock
0     Pen    Red         10              50
1     Pen    Red         25              10
2  Pencil  Black          7              47
3  Pencil  Black          5              34
4     Pen   Blue         50              55
5     Pen   Blue         20              14
```

Now, let us assume, we have to reshape the above table with Item as the index and Color as the column. We will use pivot function as given below:

```
>>> pivot3=df.pivot(index='Item',columns='Color
',values='Units_in_stock')
```

But this statement results in an error: “ValueError: Index contains duplicate entries, cannot reshape”. This is because duplicate data can’t be reshaped using pivot function. Hence, before calling the pivot() function, we need to ensure that our data do not have rows with duplicate values for the specified columns. If we can’t ensure this, we may have to use pivot_table function instead.

(C) Pivot Table

It works like a pivot function, but aggregates the values from rows with duplicate entries for the specified columns. In other words, we can use aggregate functions like min, max, mean etc, wherever we have duplicate entries. The default aggregate function is mean.

Syntax:

```
pandas.pivot_table(data, values=None,
index=None, columns=None, aggfunc='mean')
```

The parameter aggfunc can have values among sum, max, min, len, np.mean, np.median.

We can apply index to multiple columns if we don't have any unique column to act as index.

```
>>> df1 = df.pivot_
table(index=['Item','Color'])
>>> print(df1)
              Price(Rs)  Units_in_stock
Item   Color
Pen    Blue        35.0            34.5
       Red         17.5            30.0
Pencil Black        6.0            40.5
```

Please note that mean has been used as the default aggregate function. Price of the blue pen in the original data is 50 and 20. Mean has been used as aggregate and the price of the blue pen is 35 in df1.

We can use multiple aggregate functions on the data. Below example shows the use of the sum, max and np.mean function.

```
>>> pivot_table1=df.pivot_table(index='
Item',columns='Color',values='Units_in_
stock',aggfunc=[sum,max,np.mean])

>>> pivot_table1
```

```
                 sum                  max                  mean
Color     Black  Blue  Red     Black  Blue  Red     Black  Blue   Red
Item
Pen         NaN  69.0  60.0     NaN   55.0  50.0     NaN   34.5  30.0
Pencil     81.0   NaN  NaN     47.0   NaN   NaN     40.5   NaN   NaN
```

Pivoting can also be done on multiple columns. Further, different aggregate functions can be applied on different columns. The following example demonstrates pivoting on two columns - Price(Rs) and Units_in_stock. Also, the application of len() function on the column

Notes

Price(Rs) and mean() function of column Units_in_stock is shown in the example. Note that the aggregate function len returns the number of rows corresponding to that entry.

```
>>> pivot_table1=df.pivot_table(index='Item'
,columns='Color',values=['Price(Rs)','Units_
in_stock'],aggfunc={"Price(Rs)":len,"Units_in_
stock":np.mean})

>>> pivot_table1
               Price(Rs)        Units_in_stock
Color        Black Blue  Red    Black  Blue   Red
Item
Pen            NaN  2.0  2.0      NaN  34.5  30.0
Pencil         2.0  NaN  NaN     40.5   NaN   NaN
```

Program 3-11 Write the statement to print the maximum price of pen of each color.

```
>>> dfpen=df[df.Item=='Pen']
>>> pivot_redpen=dfpen.pivot_table(index='Item'
,columns=['Color'],values=['Price(Rs)'],aggfun
c=[max])
>>> print(pivot_redpen)

                max
             Price(Rs)
Color        Blue  Red
Item
Pen            50   25
```

3.8 Handling Missing Values

As we know that a DataFrame can consist of many rows (objects) where each row can have values for various columns (attributes). If a value corresponding to a column is not present, it is considered to be a missing value. A missing value is denoted by NaN.

In the real world dataset, it is common for an object to have some missing attributes. There may be several reasons for that. In some cases, data was not collected properly resulting in missing data e.g some people did not fill all the fields while taking the survey. Sometimes, some attributes are not relevant to all. For example, if a person is unemployed then salary attribute will be irrelevant and hence may not have been filled up.

NOTES

Missing values create a lot of problems during data analysis and have to be handled properly. The two most common strategies for handling missing values explained in this section are:

i) drop the object having missing values,
ii) fill or estimate the missing value

Let us refer to the previous case study given at table 3.1. Suppose, the students have now appeared for Unit Test 4 also. But, Raman could not appear for the Science, Maths and English tests, and suppose there is no possibility of a re-test. Therefore, marks obtained by him corresponding to these subjects will be missing. The dataset after Unit Test 4 is as shown at Table 3.2. Note that the attributes 'Science, 'Maths' and 'English' have missing values in Unit Test 4 for Raman.

Table 3.2 Case study data after UT4

Result						
Name/ Subjects	Unit Test	Maths	Science	S.St.	Hindi	Eng
Raman	1	22	21	18	20	21
Raman	2	21	20	17	22	24
Raman	3	14	19	15	24	23
Raman	4			19	18	
Zuhaire	1	20	17	22	24	19
Zuhaire	2	23	15	21	25	15
Zuhaire	3	22	18	19	23	13
Zuhaire	4	19	20	17	19	16
Aashravy	1	23	19	20	15	22
Aashravy	2	24	22	24	17	21
Aashravy	3	12	25	19	21	23
Aashravy	4	15	20	20	20	17
Mishti	1	15	22	25	22	22
Mishti	2	18	21	25	24	23
Mishti	3	17	18	20	25	20
Mishti	4	14	20	19	20	18

To calculate the final result, teachers are asked to submit the percentage of marks obtained by all students. In the case of Raman, the Maths teacher decides to compute the marks obtained in 3 tests and then find the percentage of marks from the total score of 75 marks. In a way, she decides to drop the marks of Unit Test 4. However, the English teacher decides to give the same marks to Raman in the 4th test as scored in the 3rd test. Science teacher decides to give Raman zero marks in the 4th test and then computes the percentage of marks obtained. Following sections explain the code for checking missing values and the code for replacing those missing values with appropriate values.

3.8.1 Checking Missing Values

Pandas provide a function isnull() to check whether any value is missing or not in the DataFrame. This function checks all attributes and returns True in case that attribute has missing values, otherwise returns False.

The following code stores the data of marks of all the Unit Tests in a DataFrame and checks whether the DataFrame has missing values or not.

```
>>> marksUT = {
'Name':['Raman','Raman','Raman','Raman','Zuhaire','Zuhaire','Zuhaire'
,'Zuhaire','Ashravy','Ashravy','Ashravy','Ashravy','Mishti','Mishti',
'Mishti','Mishti'],
'UT':[1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4],
'Maths':[22,21,14,np.NaN,20,23,22,19,23,24,12,15,15,18,17,14],
'Science':[21,20,19,np.NaN,17,15,18,20,19,22,25,20,22,21,18,20],
'S.St':[18,17,15,19,22,21,19,17,20,24,19,20,25,25,20,19],
'Hindi':[20,22,24,18,24,25,23,21, 15,17,21,20,22,24,25,20],
'Eng':[21,24,23,np.NaN,19,15,13,16,22,21,23,17,22,23,20,18]       }
>>> df = pd.DataFrame(marksUT)
>>> print(df.isnull())
```

Output of the above code will be

```
     Name     UT  Maths  Science   S.St  Hindi    Eng
0   False  False  False    False  False  False  False
1   False  False  False    False  False  False  False
2   False  False  False    False  False  False  False
3   False  False   True     True  False  False   True
4   False  False  False    False  False  False  False
5   False  False  False    False  False  False  False
6   False  False  False    False  False  False  False
7   False  False  False    False  False  False  False
8   False  False  False    False  False  False  False
9   False  False  False    False  False  False  False
10  False  False  False    False  False  False  False
11  False  False  False    False  False  False  False
12  False  False  False    False  False  False  False
13  False  False  False    False  False  False  False
14  False  False  False    False  False  False  False
15  False  False  False    False  False  False  False
```

NOTES

One can check for each individual attribute also, e.g. the following statement checks whether attribute ‘Science’ has a missing value or not. It returns True for each row where there is a missing value for attribute ‘Science’, and False otherwise.

```
>>> print(df['Science'].isnull())
0     False
1     False
2     False
3      True
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
Name: Science, dtype: bool
```

To check whether a column (attribute) has a missing value in the entire dataset, any() function is used. It returns True in case of missing value else returns False.

```
>>> print(df.isnull().any())
Name        False
UT          False
Maths        True
Science      True
S.St        False
Hindi       False
Eng          True
dtype: bool
```

The function any() can be used for a particular attribute also. The following statements) returns True in case an attribute has a missing value else it returns False.

```
>>> print(df['Science'].isnull().any())
True
```

NOTES

```
>>> print(df['Hindi'].isnull().any())
False
```

To find the number of NaN values corresponding to each attribute, one can use the sum() function along with isnull() function, as shown below:

```
>>> print(df.isnull().sum())
Name       0
UT         0
Maths      1
Science    1
S.St       0
Hindi      0
Eng        1
dtype: int64
```

To find the total number of NaN in the whole dataset, one can use df.isnull().sum().sum().

```
>>> print(df.isnull().sum().sum())
3
```

Program 3-12 Write a program to find the percentage of marks scored by Raman in hindi.

```
>>> dfRaman = df[df['Name']=='Raman']
>>> print('Marks Scored by Raman \n\n',dfRaman)

Marks Scored by Raman
    Name  UT  Maths  Science  S.St  Hindi   Eng
0  Raman   1   22.0     21.0    18     20  21.0
1  Raman   2   21.0     20.0    17     22  24.0
2  Raman   3   14.0     19.0    15     24  23.0
3  Raman   4    NaN      NaN    19     18   NaN

>>> dfHindi = dfRaman['Hindi']
>>> print("Marks Scored by Raman in Hindi
\n\n",dfHindi)
```

Marks Scored by Raman in Hindi

```
0    20
1    22
2    24
3    18
Name: Hindi, dtype: int64

>>>  row = len(dfHindi)  # Number of Unit Tests
held. Here row will be 4
```

NOTES

```
>>> print("Percentage of Marks Scored by Raman
in Hindi\n\n",(dfHindi.sum()*100)/(25*row),"%")

# denominator in the above formula represents
the aggregate of marks of all tests. Here row
is 4 tests and 25 is maximum marks for one test
```

```
Percentage of Marks Scored by Raman in Hindi
 84.0 %
```

Program 3-13 Write a python program to find the percentage of marks obtained by Raman in Maths subject.

```
>>> dfMaths = dfRaman['Maths']
>>> print("Marks Scored by Raman in Maths
\n\n",dfMaths)
Marks Scored by Raman in Maths
0    22.0
1    21.0
2    14.0
3     NaN
Name: Maths, dtype: float64
>>> row = len(dfMaths) # here, row will be 4,
the number of Unit Tests
>>> print("Percentage of Marks Scored by Raman
in Maths\n\n", dfMaths.sum()*100/(25*row),"%")
```

```
Percentage of Marks Scored by Raman in Maths
 57%
```

Here, notice that Raman was absent in Unit Test 4 in Maths Subject. While computing the percentage, marks of the fourth test have been considered as 0.

3.8.2 Dropping Missing Values

Missing values can be handled by either dropping the entire row having missing value or replacing it with appropriate value.

Dropping will remove the entire row (object) having the missing value(s). This strategy reduces the size of the dataset used in data analysis, hence should be used in case of missing values on few objects. The dropna() function can be used to drop an entire row from the DataFrame. For example, calling dropna() function on the previous example will remove the 4th row having NaN value.

```
>>> df1 = df.dropna()
>>> print(df1)
       Name  UT  Maths  Science  S.St  Hindi   Eng
0     Raman   1   22.0     21.0    18     20  21.0
1     Raman   2   21.0     20.0    17     22  24.0
2     Raman   3   14.0     19.0    15     24  23.0
4   Zuhaire   1   20.0     17.0    22     24  19.0
5   Zuhaire   2   23.0     15.0    21     25  15.0
6   Zuhaire   3   22.0     18.0    19     23  13.0
7   Zuhaire   4   19.0     20.0    17     21  16.0
8   Ashravy   1   23.0     19.0    20     15  22.0
9   Ashravy   2   24.0     22.0    24     17  21.0
10  Ashravy   3   12.0     25.0    19     21  23.0
11  Ashravy   4   15.0     20.0    20     20  17.0
12   Mishti   1   15.0     22.0    25     22  22.0
13   Mishti   2   18.0     21.0    25     24  23.0
14   Mishti   3   17.0     18.0    20     25  20.0
15   Mishti   4   14.0     20.0    19     20  18.0
```

Now, let us consider the following code:

```
# marks obtained by Raman in all the unit tests
>>> dfRaman=df[df.Name=='Raman']

# inplace=true makes changes in the #original
DataFrame i.e. dfRaman #here
>>> dfRaman.dropna(inplace=True,how='any')
>>> dfMaths = dfRaman['Maths'] # get the marks
scored in Maths
>>> print("\nMarks Scored by Raman in Maths
\n",dfMaths)
Marks Scored by Raman in Maths
0    22.0
1    21.0
2    14.0
3     NaN
Name: Maths, dtype: float64

>>> row = len(dfMaths)
>>> print("\nPercentage of Marks Scored by
Raman in Maths\n")
>>> print(dfMaths.sum()*100/(25*row),"%")
```

Percentage of Marks Scored by Raman in Maths

```
76.0 %
```

Note that the number of rows in dfRaman is 3 after using dropna. Hence percentage is computed from marks obtained in 3 Unit Tests.

3.8.3 Estimating Missing Values

Missing values can be filled by using estimations or approximations e.g a value just before (or after) the missing value, average/minimum/maximum of the values of that attribute, etc. In some cases, missing values are replaced by zeros (or ones).

The fillna(num) function can be used to replace missing value(s) by the value specified in num. For example, fillna(0) replaces missing value by 0. Similarly fillna(1) replaces missing value by 1. Following code replaces missing values by 0 and computes the percentage of marks scored by Raman in Science.

```
#Marks Scored by Raman in all the subjects
across the tests
>>> dfRaman = df.loc[df['Name']=='Raman']

>>> (row,col) = dfRaman.shape
>>> dfScience = dfRaman.loc[:,'Science']
>>> print("Marks Scored by Raman in Science
\n\n",dfScience)

Marks Scored by Raman in Science

0    21.0
1    20.0
2    19.0
3     NaN
Name: Science, dtype: float64

>>> dfFillZeroScience = dfScience.fillna(0)
>>> print('\nMarks Scored by Raman in Science
with Missing Values Replaced with Zero\
n',dfFillZeroScience)
```

Marks Scored by Raman in Science with Missing Values Replaced with Zero

```
0    21.0
1    20.0
```

```
2    19.0
3     0.0
Name: Science, dtype: float64

>>> print("Percentage of Marks Scored by Raman
in Science\n\n",dfFillZeroScience.sum()*100/
(25*row),"%")
```

Percentage of Marks Scored by Raman in Science

```
60.0 %
```

df.fillna(method='pad') replaces the missing value by the value before the missing value while df.fillna(method='bfill') replaces the missing value by the value after the missing value. Following code replaces the missing value in Unit Test 4 of English test by the marks of Unit Test 3 and then computes the percentage of marks obtained by Raman.

```
>>> dfEng = dfRaman.loc[:,'Eng']
>>> print("Marks Scored by Raman in English
\n\n",dfEng)
```

Marks Scored by Raman in English

```
0    21.0
1    24.0
2    23.0
3     NaN
Name: Eng, dtype: float64

>>>  dfFillPadEng = dfEng.fillna(method='pad')
>>> print('\nMarks Scored by Raman in English
with Missing Values Replaced by Previous Test
Marks\n',dfFillPadEng)
```

Marks Scored by Raman in English with Missing Values Replaced by Previous Test Marks

```
0    21.0
1    24.0
2    23.0
3    23.0
Name: Eng, dtype: float64
>>> print("Percentage of Marks Scored by Raman
in English\n\n")
>>> print(dfFillPadEng.sum()*100/(25*row),"%")
```

Percentage of Marks Scored by Raman in English

```
91.0 %
```

In this section, we have discussed various ways of handling missing values. Missing value is loss of

Notes

information and replacing missing values by some estimation will surely change the dataset. In all cases, data analysis results will not be actual results but will be a good approximation of actual results.

3.9 Import and Export of Data between Pandas and MySQL

So far, we have directly entered data and created a DataFrame and learned how to analyse data in a DataFrame. However, in actual scenarios, data need not be typed or copy pasted everytime. Rather, data is available most of the time in a file (text or csv) or in a database. Thus, in real-world scenarios, we will be required to bring data directly from a database and load to a DataFrame. This is called importing data from a database. Likewise, after analysis, we will be required to store data back to a database. This is called exporting data to a database.

Data from DataFrame can be read from and written to MySQL database. To do this, a connection is required with the MySQL database using the pymysql database driver. And for this, the driver should be installed in the python environment using the following command:

```
pip install pymysql
```

sqlalchemy is a library used to interact with the MySQL database by providing the required credentials. This library can be installed using the following command:

```
pip install sqlalchemy
```

Once it is installed, sqlalchemy provides a function create_engine() that enables this connection to be established. The string inside the function is known as connection string. The connection string is composed of multiple parameters like the name of the database with which we want to establish the connection, username, password, host, port number and finally the name of the database. And, this function returns an engine object based on this connection string. The syntax for the same is discussed below:

```
engine=create_engine('driver://
username:password@host:port/name_of_
database',index=false)
```

where,
Driver = mysql+pymysql
username=User name of the mysql (normally it is root)
password= Password of the MySql
port = usually we connect to localhost with port number 3306 (Default port number)
Name of the Database = Your database

In the following subsections, importing and exporting data between Pandas and MySQL applications are demonstrated. For this, we will use the same database CARSHOWROOM and Table INVENTORY created in Chapter 1 of this book.

```
mysql> use CARSHOWROOM ;
Database changed
mysql> select * from INVENTORY;
+-------+--------+-----------+-----------+-----------------+----------+
| CarId | CarName| Price     | Model     | YearManufacture | Fueltype |
+-------+--------+-----------+-----------+-----------------+----------+
| D001  | Car1   | 582613.00 | LXI       |            2017 | Petrol   |
| D002  | Car1   | 673112.00 | VXI       |            2018 | Petrol   |
| B001  | Car2   | 567031.00 | Sigma1.2  |            2019 | Petrol   |
| B002  | Car2   | 647858.00 | Delta1.2  |            2018 | Petrol   |
| E001  | Car3   | 355205.00 | 5 STR STD |            2017 | CNG      |
| E002  | Car3   | 654914.00 | CARE      |            2018 | CNG      |
| S001  | Car4   | 514000.00 | LXI       |            2017 | Petrol   |
| S002  | Car4   | 614000.00 | VXI       |            2018 | Petrol   |
+-------+--------+-----------+-----------+-----------------+----------+
8 rows in set (0.00 sec)
```

3.9.1 Importing Data from MySQL to Pandas

Importing data from MySQL to pandas basically refers to the process of reading a table from MySQL database and loading it to a pandas DataFrame. After establishing the connection, in order to fetch data from the table of the database we have the following three functions:

1) `pandas.read_sql_query(query,sql_conn)`

   It is used to read an sql query (query) into a DataFrame using the connection identifier (sql_conn) returned from the create_engine ().

2) `pandas.read_sql_table(table_name,sql_conn)`

   It is used to read an sql table (table_name) into a DataFrame using the connection identifier (sql_conn).

3) `pandas.read_sql(sql, sql_conn)`

   It is used to read either an sql query or an sql table (sql) into a DataFrame using the connection identifier (sql_conn).

```
>>> import pandas as pd
>>> import pymysql as py
>>> import sqlalchemy
>>> engine=create_engine('mysql+pymysql://
root:smsmb@localhost:3306/CARSHOWROOM')
>>> df = pd.read_sql_query('SELECT * FROM
INVENTORY', engine)
>>> print(df)
   CarId  CarName Price      Model    YearManufacture    Fueltype
0  D001   Car1    582613.00  LXI        2017             Petrol
1  D002   Car1    673112.00  VXI        2018             Petrol
2  B001   Car2    567031.00  Sigma1.2   2019             Petrol
3  B002   Car2    647858.00  Delta1.2   2018             Petrol
4  E001   Car3    355205.00  5STR STD   2017             CNG
5  E002   Car3    654914.00  CARE       2018             CNG
6  S001   Car4    514000.00  LXI        2017             Petrol
7  S002   Car4    614000.00  VXI        2018``           Petrol
```

3.9.2 Exporting Data from Pandas to MySQL

Exporting data from Pandas to MySQL basically refers to the process of writing a pandas DataFrame to a table of MySQL database. For this purpose, we have the following function:

```
pandas.DataFrame.to_sql(table,sql_conn,if_
exists=”fail”,index=False/True)
```

- Table specifies the name of the table in which we want to create or append DataFrame values. It is used to write the specified DataFrame to the table the connection identifier (sql_conn) returned from the create_engine ().
- The parameter if_exists specifies “the way data from the DataFrame should be entered in the table. It can have the following three values: “fail”, “replace”, “append”.
    - “fail” is the default value that indicates a ValueError if the table already exists in the database.
    - “replace” specifies that the previous content of the table should be updated by the contents of the DataFrame.
    - “append” specifies that the contents of the DataFrame should be appended to the existing table and when updated the format must be the same (column name sequences).

- Index — By default index is True means DataFrame index will be copied to MySQL table. If False, then it will ignore the DataFrame indexing.

```
#Code to write DataFrame df to database

>>> import pandas as pd
>>> import pymysql as py
>>> import sqlalchemy
>>> engine=create_engine('mysql+pymysql://
root:smsmb@localhost:3306/CARSHOWROOM')
>>> data={
'ShowRoomId':[1,2,3,4,5],
‘Location':[‘Delhi','Bangalore','Mumbai','Chand
igarh','Kerala']}

>>> df=pd.DataFrame(data)
>>> df.to_sql('showroom_info',engine,if_
exists="replace",index=False)
```

After running this python script, a mysql table with the name “showroom_info” will be created in the database.

Notes

Summary

- Descriptive Statistics are used to quantitatively summarise the given data.
- Pandas provide many statistical functions for analysis of data. Some of the functions are max(), min(), mean(), median(), mode(), std(), var() etc.
- Sorting is used to arrange data in a specified order, i.e. either ascending or descending.
- Indexes or labels of a row or column can be changed in a DataFrame. This process is known as Altering the index. Two functions reset_index and set_index are used for that purpose.
- Missing values are a hindrance in data analysis and must be handled properly.
- There are primarily two main strategies for handling missing data. Either the row (or column) having missing value is removed completely from analysis or missing value is replaced by some

NOTES

appropriate value (which may be zero or one or average etc.)

- Process of changing the structure of the DataFrame is known as Reshaping. Pandas provide two basic functions for this, pivot() and pivot_table().
- pymysql and sqlalchemy are two mandatory libraries for facilitating import and export of data between Pandas and MySQL. Before import and export, a connection needs to be established from python script to MySQL database.
- Importing data from MySQL to Panda refers to the process of fetching data from a MySQL table or database to a pandas DataFrame.
- Exporting data from Pandas to MySQL refers to the process of storing data from a pandas DataFrame to a MySQL table or database.

Exercise

1. Write the statement to install the python connector to connect MySQL i.e. pymysql.
2. Explain the difference between pivot() and pivot_table() function?
3. What is sqlalchemy?
4. Can you sort a DataFrame with respect to multiple columns?
5. What are missing values? What are the strategies to handle them?
6. Define the following terms: Median, Standard Deviation and variance.
7. What do you understand by the term MODE? Name the function which is used to calculate it.
8. Write the purpose of Data aggregation.
9. Explain the concept of GROUP BY with help on an example.
10. Write the steps required to read data from a MySQL database to a DataFrame.
11. Explain the importance of reshaping of data with an example.

12. Why estimation is an important concept in data analysis?

13. Assuming the given table: Product. Write the python code for the following:

Item	Company	Rupees	USD
TV	LG	12000	700
TV	VIDEOCON	10000	650
TV	LG	15000	800
AC	SONY	14000	750

a) To create the data frame for the above table.

b) To add the new rows in the data frame.

c) To display the maximum price of LG TV.

d) To display the Sum of all products.

e) To display the median of the USD of Sony products.

f) To sort the data according to the Rupees and transfer the data to MySQL.

g) To transfer the new dataframe into the MySQL with new values.

14. Write the python statement for the following question on the basis of given dataset:

```
          Name   Degree  Score
0      Aparna      MBA   90.0
1      Pankaj      BCA    NaN
2         Ram   M.Tech   80.0
3      Ramesh      MBA   98.0
4      Naveen      NaN   97.0
5   Krrishnav      BCA   78.0
6      Bhawna      MBA   89.0
```

a) To create the above DataFrame.

b) To print the Degree and maximum marks in each stream.

c) To fill the NaN with 76.

d) To set the index to Name.

e) To display the name and degree wise average marks of each student.

f) To count the number of students in MBA.

g) To print the mode marks BCA.

NOTES

SOLVED CASE STUDY BASED ON OPEN DATASETS

UCI dataset is a collection of open datasets, available to the public for experimentation and research purposes. 'auto-mpg' is one such open dataset.

It contains data related to fuel consumption by automobiles in a city. Consumption is measured in miles per gallon (mpg), hence the name of the dataset is auto-mpg. The data has 398 rows (also known as items or instances or objects) and nine columns (also known as attributes).

The attributes are: mpg, cylinders, displacement, horsepower, weight, acceleration, model year, origin, car name. Three attributes, cylinders, model year and origin have categorical values, car name is a string with a unique value for every row, while the remaining five attributes have numeric value.

The data has been downloaded from the UCI data repository available at http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/.

Following are the exercises to analyse the data.

1) Load auto-mpg.data into a DataFrame autodf.
2) Give description of the generated DataFrame autodf.
3) Display the first 10 rows of the DataFrame autodf.
4) Find the attributes which have missing values. Handle the missing values using following two ways:
    i. Replace the missing values by a value before that.
    ii. Remove the rows having missing values from the original dataset
5) Print the details of the car which gave the maximum mileage.
6) Find the average displacement of the car given the number of cylinders.
7) What is the average number of cylinders in a car?
8) Determine the no. of cars with weight greater than the average weight.

Chapter 4 Plotting Data using Matplotlib

“Human visual perception is the “most powerful of data interfaces between computers and Humans”

— M. McIntyre

12149CH04

4.1 Introduction

We have learned how to organise and analyse data and perform various statistical operations on Pandas DataFrames. Likewise, in Class XI, we have learned how to analyse numerical data using NumPy. The results obtained after analysis is used to make inferences or draw conclusions about data as well as to make important business decisions. Sometimes, it is not easy to infer by merely looking at the results. In such cases, visualisation helps in better understanding of results of the analysis.

Data visualisation means graphical or pictorial representation of the data using graph, chart, etc. The purpose of plotting data is to visualise variation or show relationships between variables.

Notes

Visualisation also helps to effectively communicate information to intended users. Traffic symbols, ultrasound reports, Atlas book of maps, speedometer of a vehicle, tuners of instruments are few examples of visualisation that we come across in our daily lives. Visualisation of data is effectively used in fields like health, finance, science, mathematics, engineering, etc. In this chapter, we will learn how to visualise data using Matplotlib library of Python by plotting charts such as line, bar, scatter with respect to the various types of data.

4.2 Plotting using Matplotlib

Matplotlib library is used for creating static, animated, and interactive 2D- plots or figures in Python. It can be installed using the following pip command from the command prompt:

```
pip install matplotlib
```

For plotting using Matplotlib, we need to import its Pyplot module using the following command:

```
import matplotlib.pyplot as plt
```

Here, plt is an alias or an alternative name for matplotlib.pyplot. We can use any other alias also.

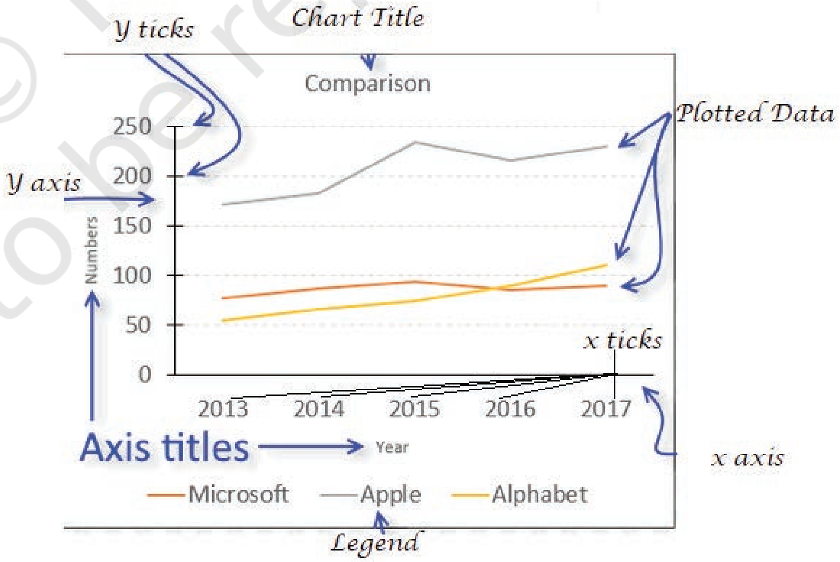


Figure 4.1: Components of a plot

The pyplot module of matplotlib contains a collection of functions that can be used to work on a plot. The plot() function of the pyplot module is used to create a figure. A figure is the overall window where the outputs of pyplot functions are plotted. A figure contains a plotting area, legend, axis labels, ticks, title, etc. (Figure 4.1). Each function makes some change to a figure: example, creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

It is always expected that the data presented through charts easily understood. Hence, while presenting data we should always give a chart title, label the axis of the chart and provide legend in case we have more than one plotted data.

To plot x versus y, we can write plt.plot(x,y). The show() function is used to display the figure created using the plot() function.

Let us consider that in a city, the maximum temperature of a day is recorded for three consecutive days. Program 4-1 demonstrates how to plot temperature values for the given dates. The output generated is a line chart.

Program 4-1 Plotting Temperature against Height

```
import matplotlib.pyplot as plt
#list storing date in string format
date=["25/12","26/12","27/12"]
#list storing temperature values
temp=[8.5,10.5,6.8]
#create a figure plotting temp versus date
plt.plot(date, temp)
#show the figure
plt.show()
```

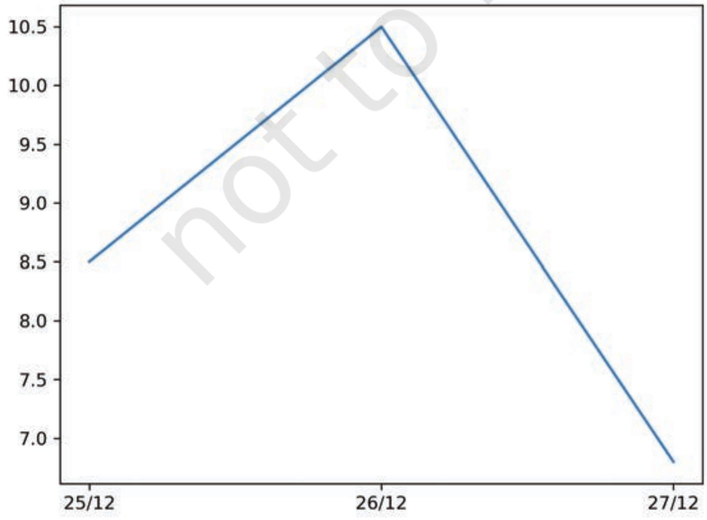


Figure 4.2: Line chart as output of Program 4-1

In program 4-1, plot() is provided with two parameters, which indicates values for x-axis and y-axis, respectively. The x and y ticks are displayed accordingly. As shown in Figure 4.2, the plot() function by default plots a line chart. We can click on the save button on the output window and save the plot as an image. A figure can also be saved by using savefig() function. The name of the figure is passed to the function as parameter.

For example: plt.savefig('x.png').

In the previous example, we used plot() function to plot a line graph. There are different types of data available for analysis. The plotting methods allow for a handful of plot types other than the default line plot, as listed in Table 4.1. Choice of plot is determined by the type of data we have.

Table 4.1 List of Pyplot functions to plot different charts

plot(*args[, scalex, scaley, data])	Plot x versus y as lines and/or markers.
bar(x, height[, width, bottom, align, data])	Make a bar plot.
boxplot(x[, notch, sym, vert, whis, ...])	Make a box and whisker plot.
hist(x[, bins, range, density, weights, ...])	Plot a histogram.
pie(x[, explode, labels, colors, autopct, ...])	Plot a pie chart.
scatter(x, y[, s, c, marker, cmap, norm, ...])	A scatter plot of x versus y.

4.3 Customisation of Plots

Pyplot library gives us numerous functions, which can be used to customise charts such as adding titles or legends. Some of the customisation options are listed in Table 4.2:

Table 4.2 List of Pyplot functions to customise plots

grid([b, which, axis])	Configure the grid lines.
legend(*args, **kwargs)	Place a legend on the axes.
savefig(*args, **kwargs)	Save the current figure.
show(*args, **kw)	Display all figures.
title(label[, fontdict, loc, pad])	Set a title for the axes.
xlabel(xlabel[, fontdict, labelpad])	Set the label for the x-axis.
xticks([ticks, labels])	Get or set the current tick locations and labels of the x-axis.
ylabel(ylabel[, fontdict, labelpad])	Set the label for the y-axis.
yticks([ticks, labels])	Get or set the current tick locations and labels of the y-axis.

Program 4-2 Plotting a line chart of date versus temperature by adding Label on X and Y axis, and adding a Title and Grids to the chart.

```
import matplotlib.pyplot as plt
date=["25/12","26/12","27/12"]
temp=[8.5,10.5,6.8]
plt.plot(date, temp)
plt.xlabel("Date")                      #add the Label on x-axis
plt.ylabel("Temperature")               #add the Label on y-axis
plt.title("Date wise Temperature")       #add the title to the chart
plt.grid(True)                     #add gridlines to the background
plt.yticks(temp)
plt.show()
```

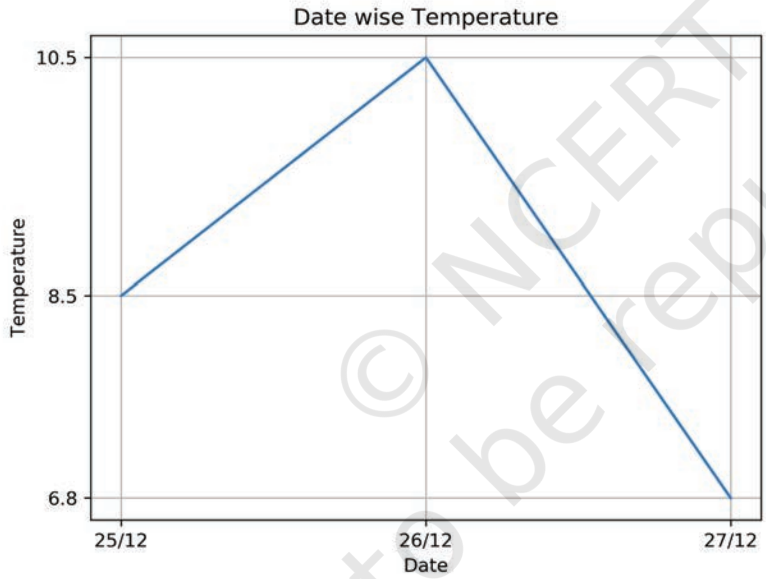


Figure 4.3: Line chart as output of Program 4-2

Think and Reflect

On providing a single list or array to the plot() function, can matplotlib generate values for both the x and y axis?

In the above example, we have used the xlabel, ylabel, title and yticks functions. We can see that compared to Figure 4.2, the Figure 4.3 conveys more meaning, easily. We will learn about customisation of other plots in later sections.

4.3.1 Marker

We can make certain other changes to plots by passing various parameters to the plot() function. In Figure 4.3, we plot temperatures day-wise. It is also possible to specify each point in the line through a marker.

A marker is any symbol that represents a data value in a line chart or a scatter plot. Table 4.3 shows a list of markers along with their corresponding symbol and description. These markers can be used in program codes:

Table 4.3 Some of the Matplotlib Markers

Marker	Symbol	Description	Marker	Symbol	Description
“.”	•	Point	“8”	⯃	octagon
“,”	·	Pixel	“s”	■	square
“o”	●	Circle	“p”	⬟	pentagon
“v”	▼	triangle_down	“P”	✚	plus (filled)
“^”	▲	triangle_up	“*”	★	star
“<”	◀	triangle_left	“h”	⬢	hexagon1
“>”	▶	triangle_right	“H”	⬣	hexagon2
“1”	Y	tri_down	“+”	+	plus
“2”	⅄	tri_up	“x”	✕	x
“3”	≺	tri_left	“X”	✖	x (filled)
“4”	≻	tri_right	“D”	◆	diamond

4.3.2 Colour

It is also possible to format the plot further by changing the colour of the plotted data. Table 4.4 shows the list of colours that are supported. We can either use character codes or the color names as values to the parameter color in the plot().

Table 4.4 Colour abbreviations for plotting

Character	Colour
‘b’	blue
‘g’	green
‘r’	red
‘c’	cyan
‘m’	magenta
‘y’	yellow
‘k’	black
‘w’	white

4.3.3 Linewidth and Line Style

The linewidth and linestyle property can be used to change the width and the style of the line chart. Linewidth is specified in pixels. The default line width is 1 pixel showing a thin line. Thus, a number greater than 1 will output a thicker line depending on the value provided.

We can also set the line style of a line chart using the linestyle parameter. It can take a string such as "solid", "dotted", "dashed" or "dashdot". Let us write the Program 4-3 applying some of the customisations.

Program 4-3 Consider the average heights and weights of persons aged 8 to 16 stored in the following two lists:

height = [121.9,124.5,129.5,134.6,139.7,147.3, 152.4, 157.5,162.6]

weight= [19.7,21.3,23.5,25.9,28.5,32.1,35.7,39.6, 43.2]

Let us plot a line chart where:

i. x axis will represent weight
ii. y axis will represent height
iii. x axis label should be “Weight in kg”
iv. y axis label should be “Height in cm”
v. colour of the line should be green
vi. use * as marker
vii. Marker size as10
viii. The title of the chart should be “Average weight with respect to average height”.
ix. Line style should be dashed
x. Linewidth should be 2.

```
import matplotlib.pyplot as plt
import pandas as pd
height=[121.9,124.5,129.5,134.6,139.7,147.3,152.4,157.5,162.6]
weight=[19.7,21.3,23.5,25.9,28.5,32.1,35.7,39.6,43.2]
df=pd.DataFrame({"height":height,"weight":weight})
#Set xlabel for the plot
plt.xlabel('Weight in kg')
#Set ylabel for the plot
```

```
plt.ylabel('Height in cm')
#Set chart title:
plt.title('Average weight with respect to average height')
#plot using marker'-*' and line colour as green
plt.plot(df.weight,df.height,marker='*',markersize=10,color='green
',linewidth=2, linestyle='dashdot')
plt.show()
```

In the above we created the DataFrame using 2 lists, and in the plot function we have passed the height and weight columns of the DataFrame. The output is shown in Figure 4.4.

> Continuous data are measured while discrete data are obtained by counting. Height, weight are examples of continuous data. It can be in decimals. Total number of students in a class is discrete. It can never be in decimals.

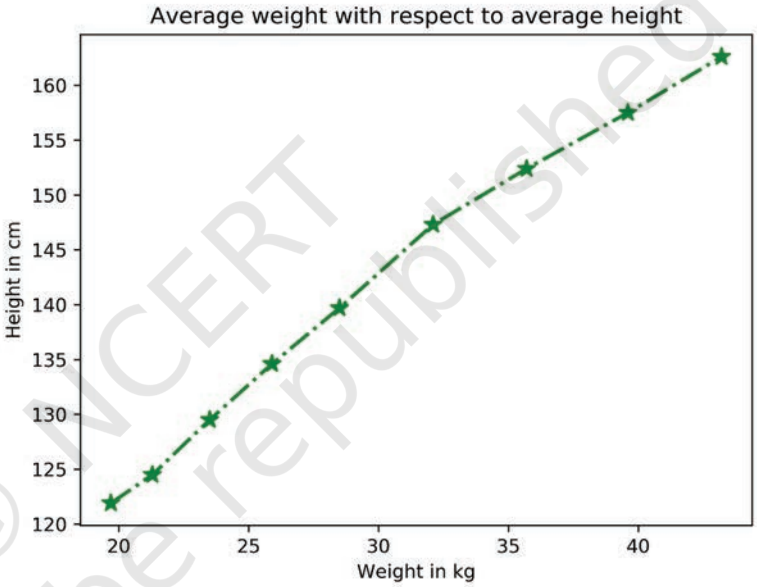


Figure 4.4: Line chart showing average weight against average height

4.4 The Pandas Plot function (Pandas Visualisation)

In Programs 4-1 and 4-2, we learnt that the plot() function of the pyplot module of matplotlib can be used to plot a chart. However, starting from version 0.17.0, Pandas objects Series and DataFrame come equipped with their own .plot() methods. This plot() method is just a simple wrapper around the plot() function of pyplot. Thus, if we have a Series or DataFrame type object (let's say 's' or 'df') we can call the plot method by writing:

```
s.plot() or df.plot()
```

The plot() method of Pandas accepts a considerable number of arguments that can be used to plot a variety of graphs. It allows customising different plot types by supplying the kind keyword arguments. The general syntax is: plt.plot(kind),where kind accepts a string indicating the type of .plot, as listed in Table 4.5. In addition, we can use the matplotlib.pyplot methods and functions also along with the plt() method of Pandas objects.

Table 4.5 Arguments accepted by kind for different plots

kind =	Plot type
line	Line plot (default)
bar	Vertical bar plot
barh	Horizontal bar plot
hist	Histogram
box	Boxplot
area	Area plot
pie	Pie plot
scatter	Scatter plot

In the previous chapters, we have learned to store different types of data in a two dimensional format using DataFrame. In the subsequent sections we will learn to use plot() function to create various types of charts with respect to the type of data stored in DataFrames.

4.4.1 Plotting a Line chart

A line plot is a graph that shows the frequency of data along a number line. It is used to show continuous dataset. A line plot is used to visualise growth or decline in data over a time interval. We have already plotted line charts through Programs 4-1 and 4-2. In this section, we will learn to plot a line chart for data stored in a DataFrame.

Activity 4.1

Create the MelaSale.csv using Python Pandas containing data as shown in Table 4.6.

Program 4-4 Smile NGO has participated in a three week cultural mela. Using Pandas, they have stored the sales (in Rs) made day wise for every week in a CSV file named “MelaSales.csv”, as shown in Table 4.6.

Table 4.6 Day-wise mela sales data

Week 1	Week 2	Week 3
5000	4000	4000
5900	3000	5800
6500	5000	3500
3500	5500	2500
4000	3000	3000
5300	4300	5300
7900	5900	6000

Depict the sales for the three weeks using a Line chart. It should have the following:

i. Chart title as “Mela Sales Report”.
ii. axis label as Days.
iii. axis label as “Sales in Rs”.

Line colours are red for week 1, blue for week 2 and brown for week 3.

```
import pandas as pd
import matplotlib.pyplot as plt
# reads "MelaSales.csv" to df by giving path to the file
df=pd.read_csv("MelaSales.csv")
#create a line plot of different color for each week
df.plot(kind='line', color=['red','blue','brown'])
# Set title to "Mela Sales Report"
plt.title('Mela Sales Report')
# Label x axis as "Days"
plt.xlabel('Days')
# Label y axis as "Sales in Rs"
plt.ylabel('Sales in Rs')
#Display the figure
plt.show()
```

The Figure 4.5 displays a line plot as output for Program 4-4. Note that the legend is displayed by default associating the colours with the plotted data.

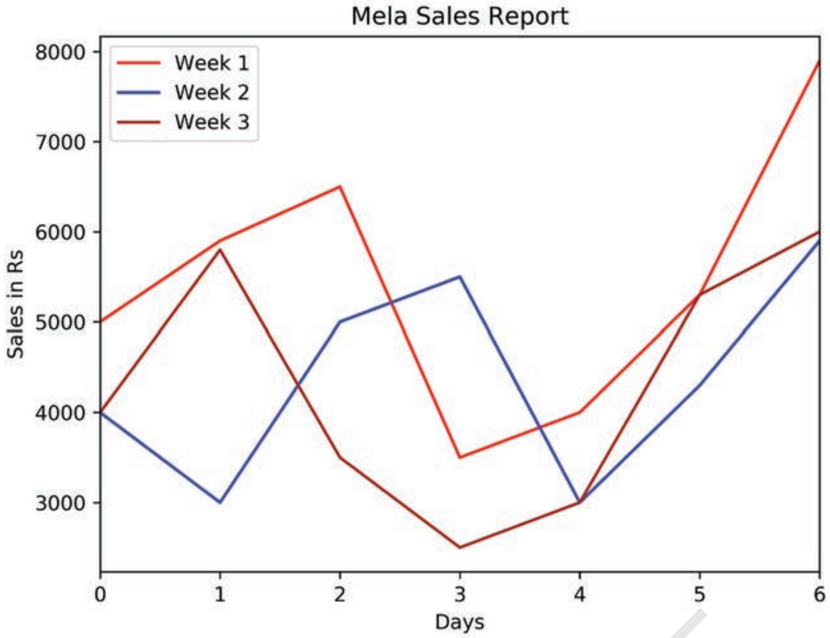


Figure 4.5: Line plot showing mela sales figures

The line plot takes a numeric value to display on the x axis and hence uses the index (row labels) of the DataFrame in the above example. Thus, x tick values are the index of the DataFramedf that contains data stored in MelaSales.CSV.

Customising Line Plot

We can substitute the ticks at x axis with a list of values of our choice by using plt.xticks(ticks,label) where ticks is a list of locations(locs) on x axis at which ticks should be placed, label is a list of items to place at the given ticks.

Program 4-5 Assuming the same CSV file, i.e., MelaSales. CSV, plot the line chart with following customisations:

```
Maker ="*"
Marker size=10
linestyle="--"
Linewidth =3
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("MelaSales.csv")
#creates plot of different color for each week
df.plot(kind='line', color=['red','blue','brown'],marker="*",markersize=10,linewidth=3,linestyle="--")
```

```
plt.title('Mela Sales Report')
plt.xlabel('Days')
plt.ylabel('Sales in Rs')
#store converted index of DataFrame to a list
ticks = df.index.tolist()
#displays corresponding day on x axis
plt.xticks(ticks,df.Day)
plt.show()
```

Figure 4.6 is generated as output of Program 4-5 with xticks as Day names.

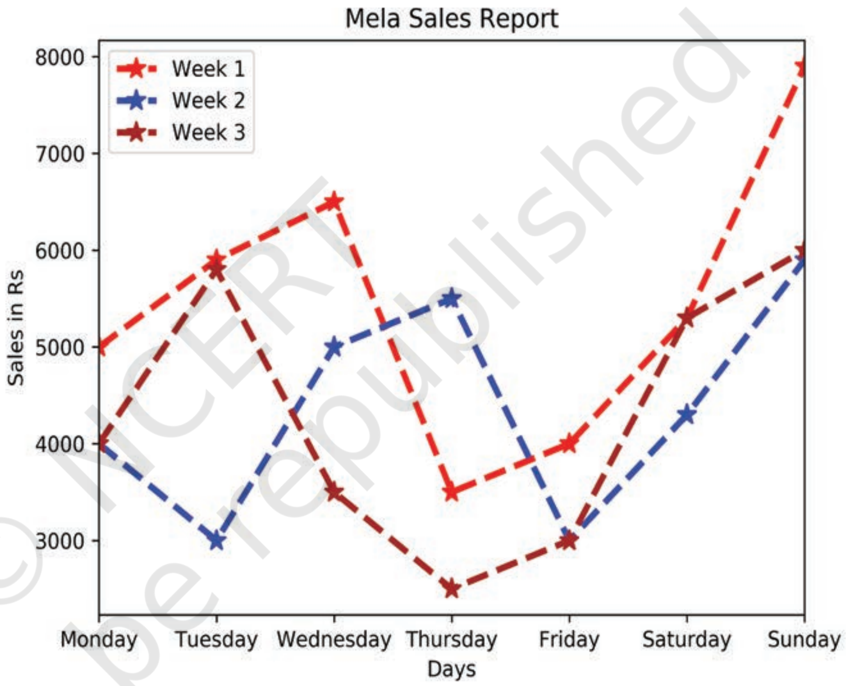


Figure 4.6: Mela sales figures with day names

4.4.2 Plotting Bar Chart

The line plot in Figure 4.6 shows that the sales for all the weeks increased during the weekend. Other than weekends, it also shows that the sales increased on Wednesday for Week 1, on Thursday for Week 2 and on Tuesday for Week 3.

But, the lines are unable to efficiently depict comparison between the weeks for which the sales data is plotted. In order to show comparisons, we prefer Bar charts. Unlike line plots, bar charts can plot strings on the x axis. To plot a bar chart, we will specify kind='bar'. We can also specify the DataFrame columns to be used as x and y axes.

Let us now add a column "Days" consisting of day names to "MelaSales.csv" as shown in Table 4.7.

Table 4.7 Day-wise sales data along with Day's names

Week 1	Week 2	Week 3	Day
5000	4000	4000	Monday
5900	3000	5800	Tuesday
6500	5000	3500	Wednesday
3500	5500	2500	Thursday
4000	3000	3000	Friday
5300	4300	5300	Saturday
7900	5900	6000	Sunday

> If we do not specify the column name for the x parameter in the plot(), the bar plot will plot all the columns of the DataFrame with the index (row label) of DataFrame at x axis which is a numeric starting from 0.

Program 4-6 This program displays the Python script to display Bar plot for the "MelaSales.csv" file with column Day on x axis as shown below in Figure 4.7

```
import pandas as pd
df= pd.read_csv('MelaSales.csv')
import matplotlib.pyplot as plt
# plots a bar chart with the column "Days" as x axis
df.plot(kind='bar',x='Day',title='Mela Sales Report')
#set title and set ylabel
plt.ylabel('Sales in Rs')
plt.show()
```

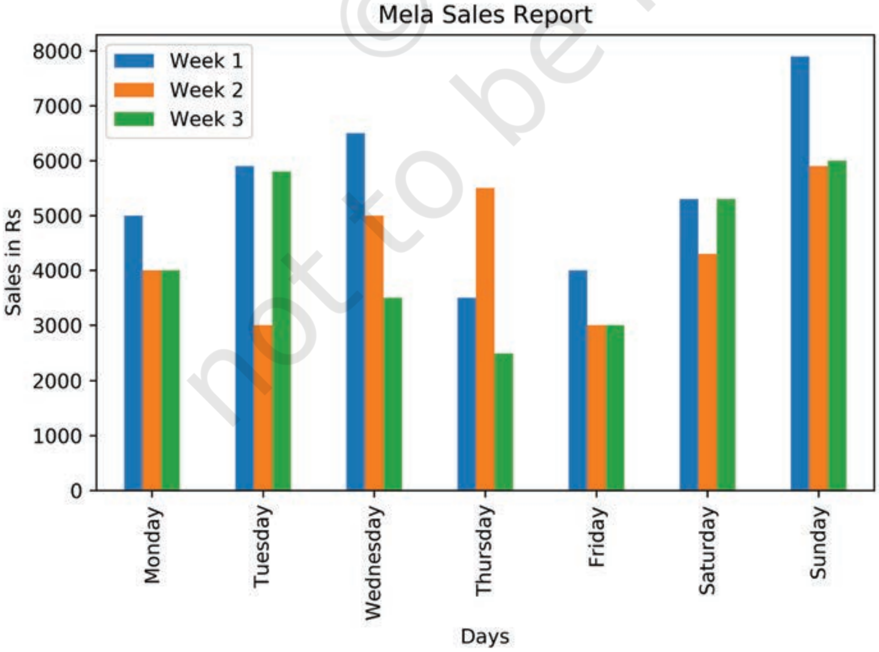


Figure 4.7: A bar chart as output of Program 4-6

Customising Bar Chart

We can also customise the bar chart by adding certain parameters to the plot function. We can control the edgecolor of the bar, linestyle and linewidth. We can also control the color of the lines. The following example shows various customisations on the bar chart of Figure 4.8

Program 4-7 Let us write a Python script to display Bar plot for the "MelaSales.csv" file with column Day on x axis, and having the following customisation:

- Changing the color of each bar to red, yellow and purple.
- Edgecolor to green
- Linewidth as 2
- Line style as "--"

```
import pandas as pd
import matplotlib.pyplot as plt
df= pd.read_csv('MelaSales.csv')
# plots a bar chart with the column "Days" as x axis
df.plot(kind='bar',x='Day',title='Mela Sales Report',color=['red',
'yellow','purple'],edgecolor='Green',linewidth=2,linestyle='--')
#set title and set ylabel
plt.ylabel('Sales in Rs')
plt.show()
```

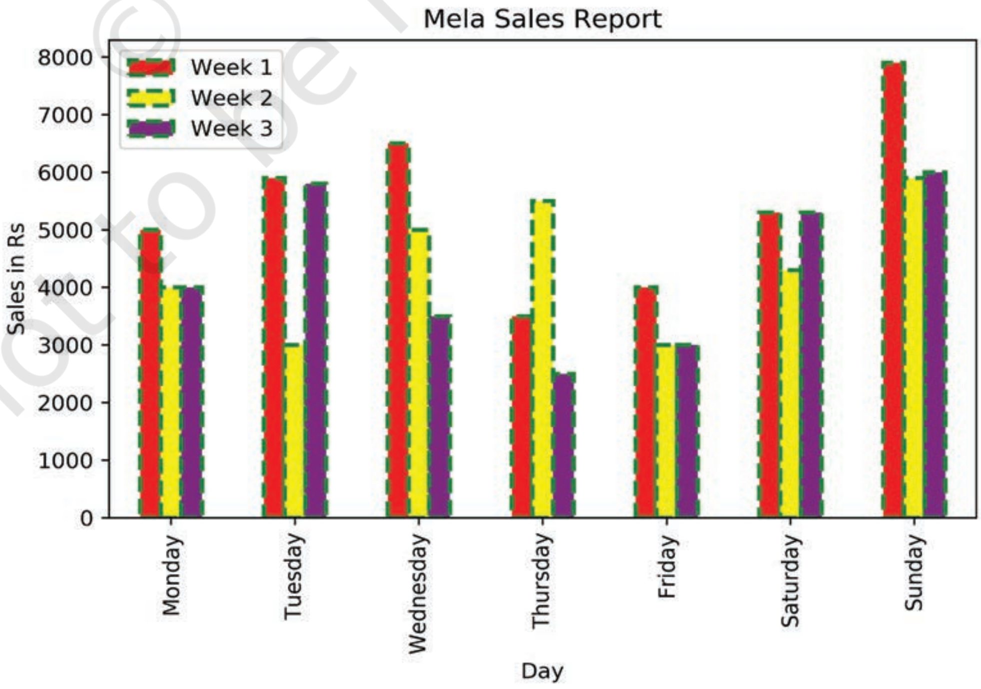


Figure 4.8: A bar chart as output of Program 4-7

4.4.3 Plotting Histogram

Histograms are column-charts, where each column represents a range of values, and the height of a column corresponds to how many values are in that range.

To make a histogram, the data is sorted into "bins" and the number of data points in each bin is counted. The height of each column in the histogram is then proportional to the number of data points its bin contains.

If we do not specify Bins are the number of intervals you want to divide all of your data into, such that it can be displayed as bars on a histogram.

The df.plot(kind='hist') function automatically selects the size of the bins based on the spread of values in the data.

Program 4-8

```
import pandas as pd
import  matplotlib.pyplot as plt
data = {'Name':['Arnav', 'Sheela', 'Azhar', 'Bincy', 'Yash',
'Nazar'],
'Height' : [60,61,63,65,61,60],
'Weight' : [47,89,52,58,50,47]}
      }
df=pd.DataFrame(data)
df.plot(kind='hist')
plt.show()
```

Think and Reflect

How can we make the bar chart of Figure 4.8 horizontal?

The Program 4-9 displays the histogram corresponding to all attributes having numeric values, i.e., 'Height' and 'Weight' attributes as shown in Figure 4.9. On the basis of the height and weight values provided in the DataFrame, the plot() calculated the bin values.

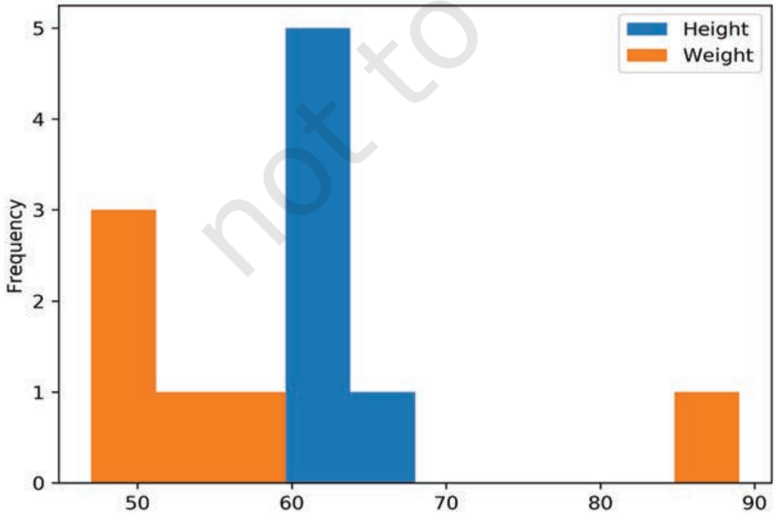


Figure 4.9: A histogram as output of Program 4-8

It is also possible to set value for the bins parameter, for example,

```
df.plot(kind=’hist’,bins=20)
df.plot(kind='hist',bins=[18,19,20,21,22])
df.plot(kind='hist',bins=range(18,25))
```

Customising Histogram

Taking the same data as above, now let see how the histogram can be customised. Let us change the edgecolor, which is the border of each hist, to green. Also, let us change the line style to ":" and line width to 2. Let us try another property called fill, which takes boolean values. The default True means each hist will be filled with color and False means each hist will be empty. Another property called hatch can be used to fill to each hist with pattern ('-', '+', 'x', '\\', '*', 'o', 'O', '.'). In the Program 4-10, we have used the hatch value as "o".

Program 4-9

```
import pandas as pd
import matplotlib.pyplot as plt
data = {'Name':['Arnav', 'Sheela', 'Azhar','Bincy','Yash',
'Nazar'],
'Height' : [60,61,63,65,61,60],
'Weight' : [47,89,52,58,50,47]}
df=pd.DataFrame(data)
df.plot(kind='hist',edgecolor='Green',linewidth=2,linestyle=':',fil
l=False,hatch='o')
plt.show()
```

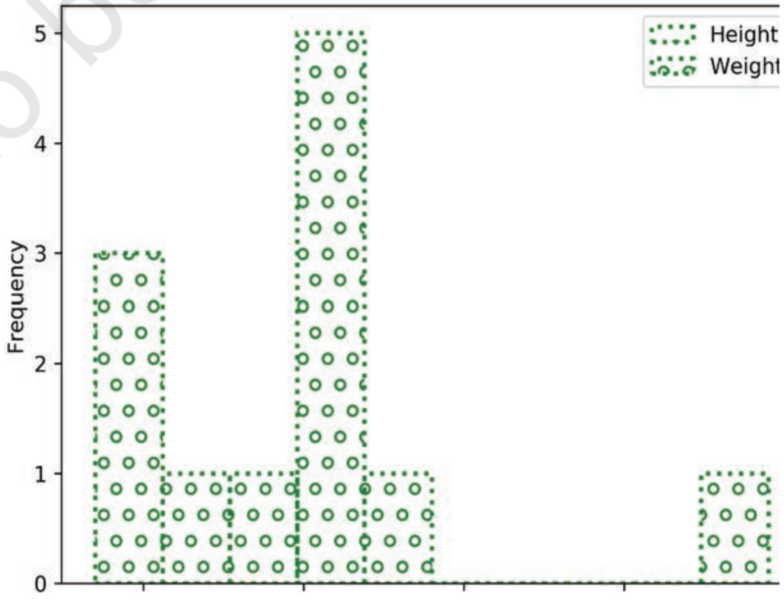


Figure 4.10: Customised histogram as output of Program 4-9

Using Open Data

There are many websites that provide data freely for anyone to download and do analysis, primarily for educational purposes. These are called Open Data as the data source is open to the public. Availability of data for access and use promotes further analysis and innovation. A lot of emphasis is being given to open data to ensure transparency, accessibility and innovation. “Open Government Data (OGD) Platform India” (data.gov.in) is a platform for supporting the Open Data initiative of the Government of India. Large datasets on different projects and parameters are available on the platform.

Let us consider a dataset called “Seasonal and Annual Min/Max Temp Series - India from 1901 to 2017” from the URL https://data.gov.in/resources/seasonal-and-annual-minmax-temp-series-india-1901-2017.

Our aim is to plot the minimum and maximum temperature and observe the number of times (frequency) a particular temperature has occurred. We only need to extract the 'ANNUAL - MIN' and 'ANNUAL - MAX' columns from the file. Also, let us aim to display two Histogram plots:

i) Only for 'ANNUAL - MIN'

ii) For both 'ANNUAL - MIN' and 'ANNUAL - MAX'

Program 4-10

```
import pandas as pd
import matplotlib.pyplot as plt
#read the CSV file with specified columns
#usecols parameter to extract only two required columns
data=pd.read_csv("Min_Max_Seasonal_IMD_2017.csv",
     usecols=['ANNUAL - MIN','ANNUAL - MAX'])
df=pd.DataFrame(data)
#plot histogram for 'ANNUAL - MIN'
df.plot(kind='hist',y='ANNUAL - MIN',title='Annual Minimum
Temperature (1901-2017)')
plt.xlabel('Temperature')
plt.ylabel('Number of times')
#plot histogram for both 'ANNUAL - MIN' and 'ANNUAL - MAX'
df.plot(kind='hist',
     title='Annual Min and Max Temperature (1901-2017)',color=['b
lue','red'])
plt.xlabel('Temperature')
plt.ylabel('Number of times')
plt.show()
```

The Figures 4.11 and 4.12 are produced as output of Program 4-10.

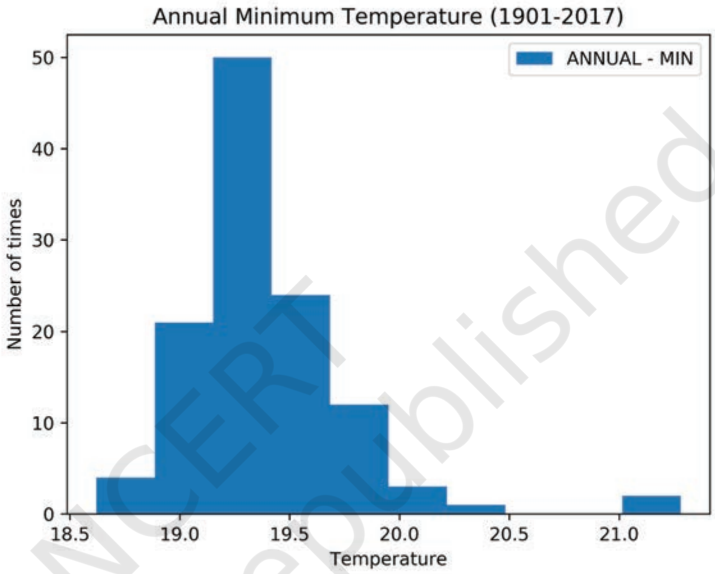


Figure 4.11: Histogram for 'ANNUAL – MIN' and 'ANNUAL – MAX'

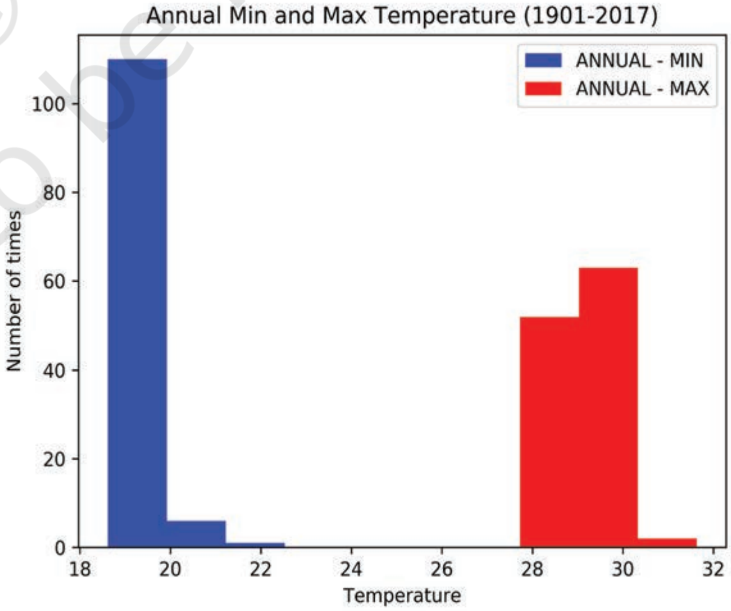


Figure 4.12: Histogram for 'ANNUAL – MIN'

Program 4-11 Plot a frequency polygon for the 'ANNUAL – MIN' column of the "Min/Max Temp" data over the histogram depicting it.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data=pd.read_csv("Min_Max_Seasonal_IMD_2017.csv",
     usecols=['ANNUAL - MIN'])
     df=pd.DataFrame(data)
     #convert the 'ANNUAL - MIN' column into a numpy 1D array
     minarray=np.array([df['ANNUAL - MIN']])
     # Extract y (frequency) and edges (bins)
     y,edges = np.histogram(minarray)
     #calculate the midpoint for each bar on the histogram
     mid = 0.5*(edges[1:]+ edges[:-1])
     df.plot(kind='hist',y='ANNUAL - MIN'
     plt.plot(mid,y,'-^')
     plt.title('Annual Min Temperature plot(1901 - 2017)')
     plt.xlabel('Temperature')
     plt.show()
```

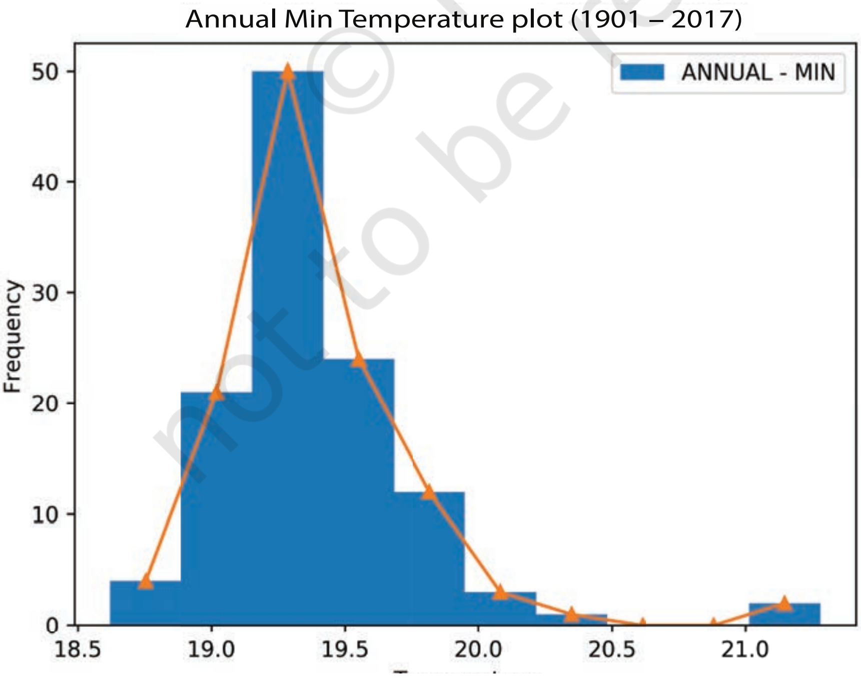


Figure 4.13: Output of Program 4-11

4.4.4 Plotting Scatter Chart

A scatter chart is a two-dimensional data visualisation method that uses dots to represent the values obtained for two different variables —one plotted along the x-axis and the other plotted along the y-axis.
Scatter plots are used when you want to show the relationship between two variables. Scatter plots are sometimes called correlation plots because they show how two variables are correlated. Additionally, the size, shape or color of the dot could represent a third (or even fourth variable).

Program 4-12 Prayatna sells designer bags and wallets. During the sales season, he gave discounts ranging from 10% to 50% over a period of 5 weeks. He recorded his sales for each type of discount in an array. Draw a scatter plot to show a relationship between the discount offered and sales made.

```
import numpy as np
import matplotlib.pyplot as plt
discount= np.array([10,20,30,40,50])
saleInRs=np.array([40000,45000,48000,50000,100000])
plt.scatter(x=discount,y=saleInRs)
plt.title('Sales Vs Discount')
plt.xlabel('Discount offered')
plt.ylabel('Sales in Rs')
plt.show()
```

Activity 4.2

What value does each bubble on the plot at Figure 4.14 represent?

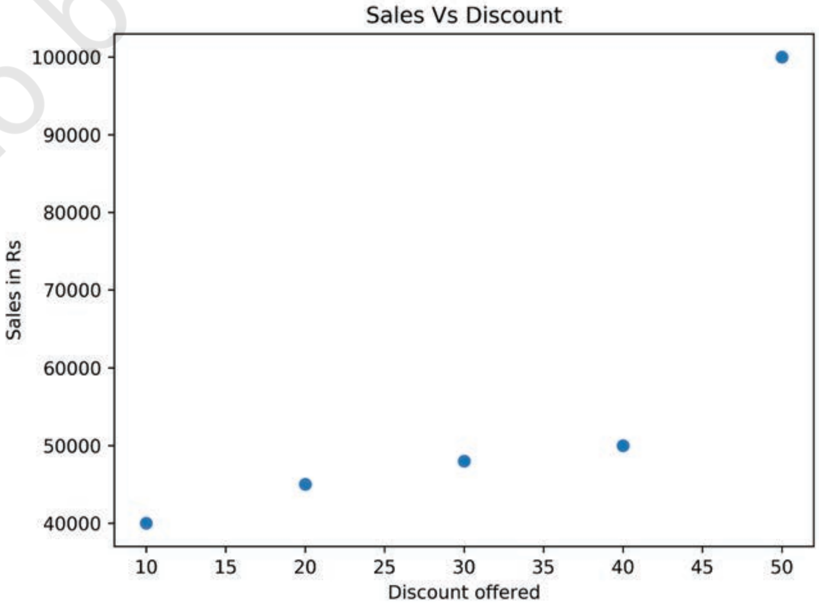


Figure 4.14: Output of Program 4-12

Customising Scatter chart

The size of the bubble can also be used to reflect a value. For example, in program 4-14, we have opted for displaying the size of the bubble as 10 times the discount, as shown in Figure 4.15. The colour and markers can also be changed in the above plot by adding the following statements:

Program 4-13

```
import numpy as np
import  matplotlib.pyplot as plt
discount= np.array([10,20,30,40,50])
saleInRs=np.array([40000,45000,48000,50000,100000])
size=discount*10
plt.scatter(x=discount,y=saleInRs,s=size,color='red',linewidth=3,m
arker='*',edgecolor='blue')
plt.title('Sales Vs Discount')
plt.xlabel('Discount offered')
plt.ylabel('Sales in Rs')
plt.show()
```

Think and Reflect

What would happen if we use df.plot(kind='scatter') instead of plt.scatter() in Program 4-13?

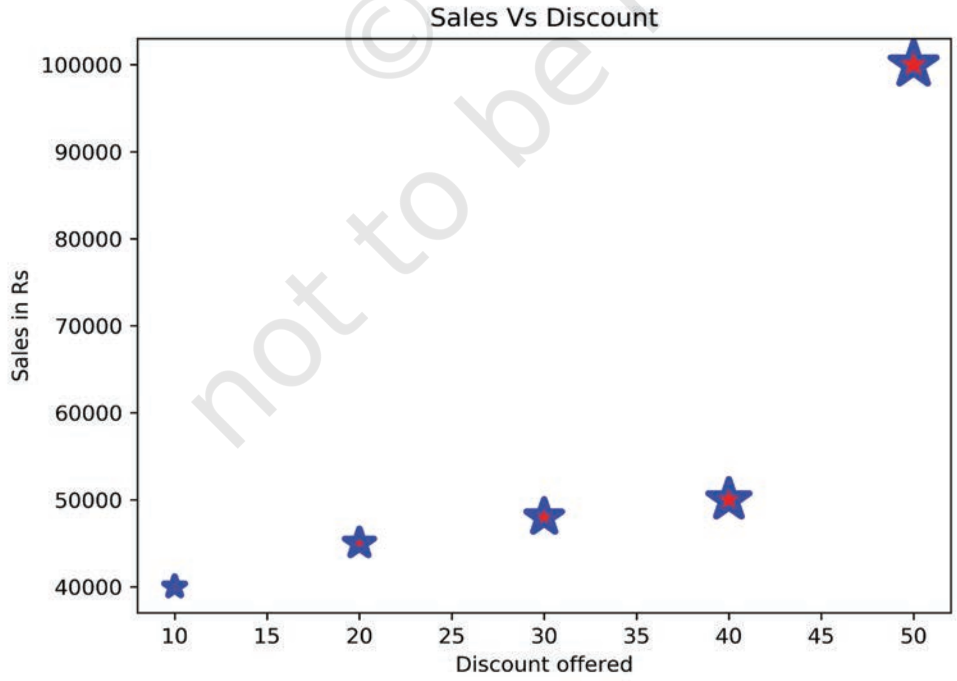


Figure 4.15: Scatter plot based on modified Program 4-13

NOTES

4.4.5 Plotting Quartiles and Box plot

Suppose an entrance examination of 200 marks is conducted at the national level, and Mahi has topped the exam by scoring 120 marks. The result shows 100 percentile against Mahi's name, which means all the candidates excluding Mahi have scored less than Mahi. To visualise this kind of data, we use quartiles.

Quartiles are the measures which divide the data into four equal parts, and each part contains an equal number of observations. Calculating quartiles requires calculation of median. Quartiles are often used in educational achievement data, sales and survey data to divide populations into groups. For example, you can use Quartile to find the top 25 percent of students in that examination.

A Box Plot is the visual representation of the statistical summary of a given data set. The summary includes Minimum value, Quartile 1, Quartile 2, Median, Quartile 4 and Maximum value. The whiskers are the two lines outside the box that extend to the highest and lowest values. It also helps in identifying the outliers. An outlier is an observation that is numerically distant from the rest of the data, as shown in Figure 4.16:

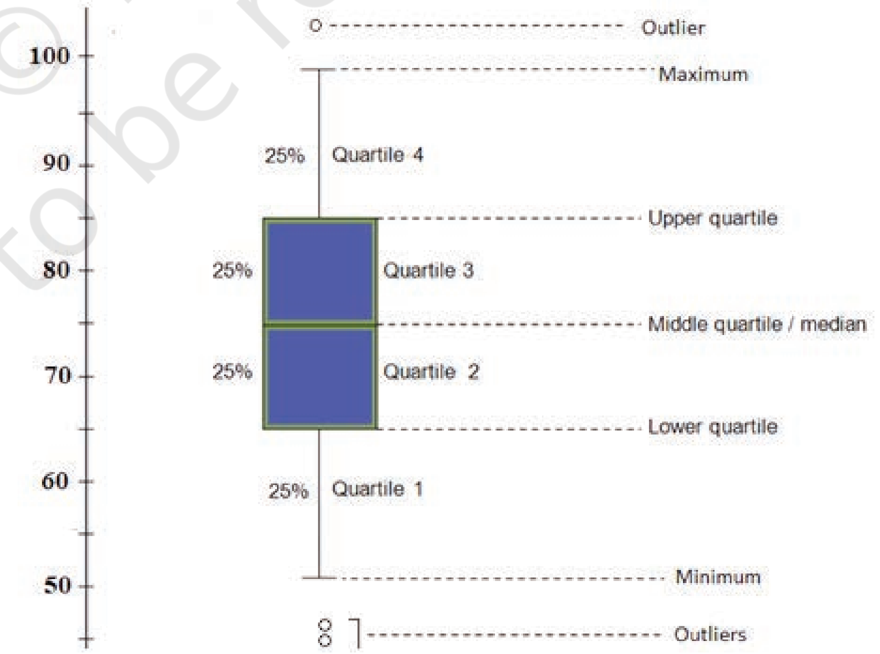


Figure 4.16: A Box Plot

Program 4-14 In order to assess the performance of students of a class in the annual examination, the class teacher stored marks of the students in all the 5 subjects in a CSV “Marks.csv” file as shown in Table 4.8. Plot the data using boxplot and perform a comparative analysis of performance in each subject.

Table 4.8 Marks obtained by students in five subjects

Name	English	Maths	Hindi	Science	Social_Studies
Rishika Batra	95	95	90	94	95
Waseem Ali	95	76	79	77	89
Kulpreet Singh	78	81	75	76	88
Annie Mathews	88	63	67	77	80
Shiksha	95	55	51	59	80
Naveen Gupta	82	55	63	56	74
Taleem Ahmed	73	49	54	60	77
Pragati Nigam	80	50	51	54	76
Usman Abbas	92	43	51	48	69
Gurpreet Kaur	60	43	55	52	71
Sameer Murthy	60	43	55	52	71
Angelina	78	33	39	48	68
Angad Bedi	62	43	51	48	54

```
Program 4-14
     import numpy as np
     import pandas as pd
     import  matplotlib.pyplot as plt
     data= pd.read_csv('Marks.csv')
     df= pd.DataFrame(data)
     df.plot(kind='box')
     #set title,xlabel,ylabel
     plt.title('Performance Analysis')
     plt.xlabel('Subjects')
     plt.ylabel('Marks')
     plt.show()
```

Think and Reflect

What would happen if the label or row index passed is not present in the DataFrame?

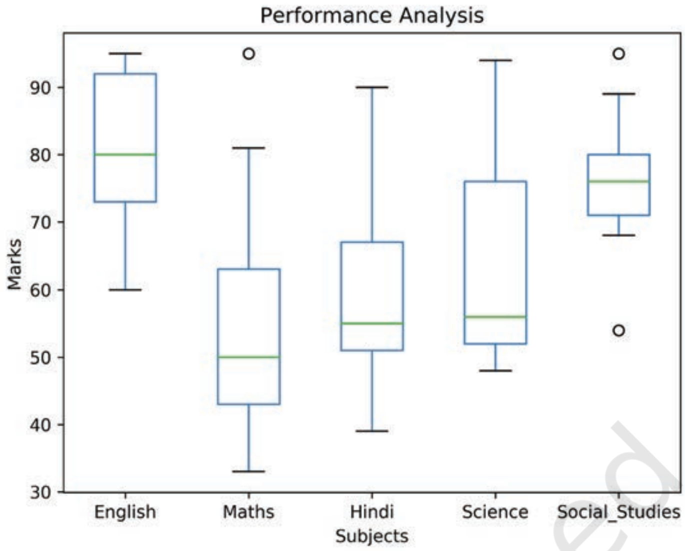


Figure 4.17: A boxplot of "Marks.csv"

The distance between the box and lower or upper whiskers in some boxplots are more, and in some less. Shorter distance indicates small variation in data, and longer distance indicates spread in data to mean larger variation.

Program 4-15 To keep improving their services, XYZ group of hotels have asked all the three hotels to get feedback form filled by their customers at the time of checkout. After getting ratings on a scale of (1–5) on factors such as Food, Service, Ambience, Activities, Distance from tourist spots they calculate the average rating and store it in a CSV file. The data are given in Table 4.9.

Table 4.9 Year-wise average ratings on five parameters

Year	Sunny Bunny Resort	Happy Lucky Resort	Breezy WIndy Resort
2014	4.75	3	4.5
2015	2.5	4	2
2016	3.5	2.5	3
2017	4	2	3.5
2018	1.5	4.5	1

This year, to award the best hotel they have decided to analyse the ratings of the past 5 years for each of the hotels. Plot the data using Boxplot.

```
Program 4-15
import pandas as pd
import  matplotlib.pyplot as plt
#read the CSV file in 'data'
data= pd.read_csv('compareresort.csv')
#convert 'data' into a DataFrame 'df'
df= pd.DataFrame(data)
#plot a box plot for the DataFrame 'df'
with a title
df.plot(kind='box',title='Compare Resorts')
#set xlabel,ylabel
plt.xlabel('Resorts')
plt.ylabel('Rating (5 years)')
#display the plot
plt.show()
```

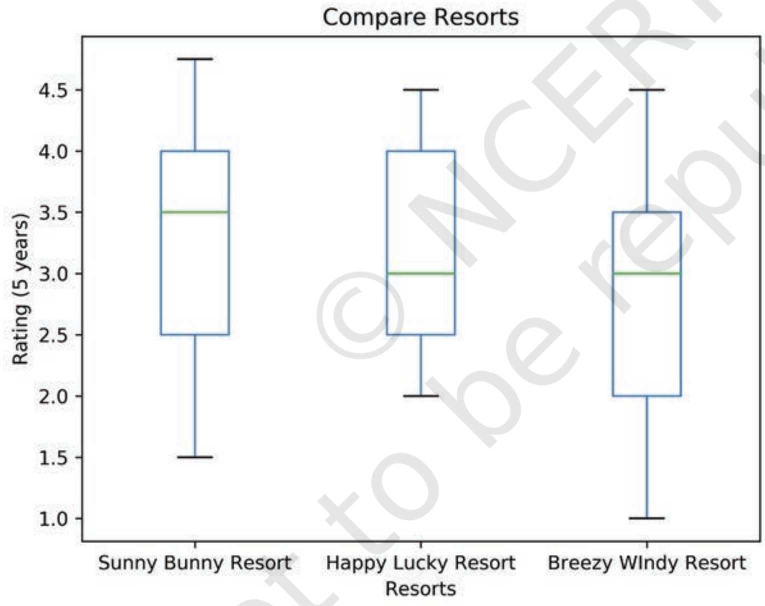


Figure 4.18: A boxplot as output of Program 4.15.

Customising Box plot

We can display the whisker in horizontal direction by adding a parameter vert=False in the Program 4-15, as shown in the following line of code. We can change the color of the whisker as well. The output of the modified Program is shown in Figure 4.19.

```
df.plot(kind='box',title='Compare Resorts',
color='red', vert=False)
```

Think and Reflect

Which of the three resorts should be awarded? Give reasons.

Activity 4.3

Plot a pie to display the radius of the planets and also give an appropriate title to the plot.

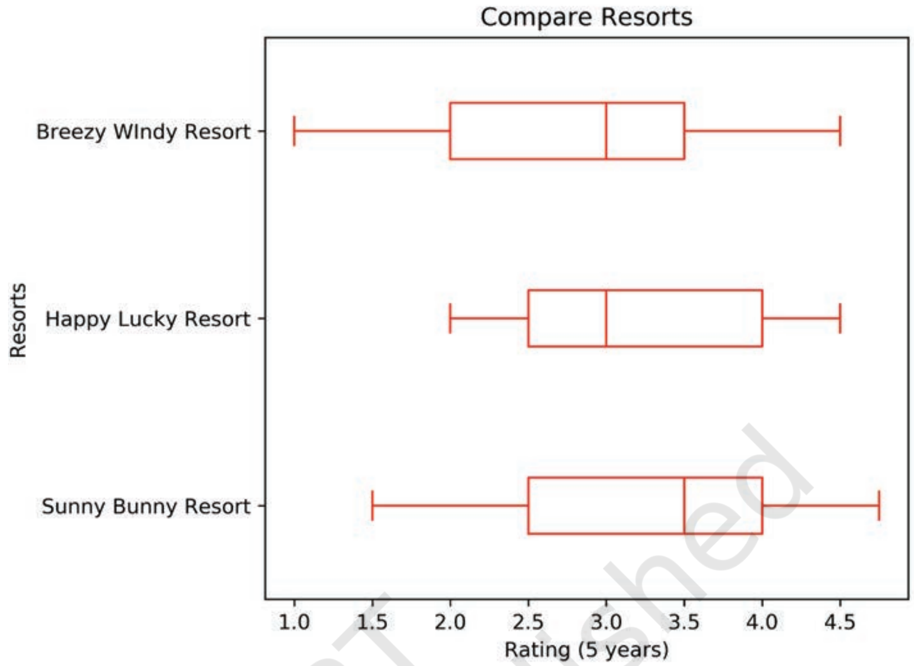


Figure 4.19: The horizontal boxplot after modifying Program 4.15.

4.4.6 Plotting Pie Chart

Pie is a type of graph in which a circle is divided into different sectors and each sector represents a part of the whole. A pie plot is used to represent numerical data proportionally. To plot a pie chart, either column label y or 'subplots=True' should be set while using df.plot(kind='pie') . If no column reference is passed and subplots=True, a 'pie' plot is drawn for each numerical column independently.

In the Program 4.16, we have a DataFrame with information about the planet's mass and radius. The 'mass' column is passed to the plot() function to get a pie plot as shown in Figure 4.20.

Program 4-16

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.DataFrame({'mass': [0.330, 4.87 , 5.97],
      'radius': [2439.7, 6051.8, 6378.1]},
      index=['Mercury', 'Venus', 'Earth'])
df.plot(kind='pie',y='mass')
plt.show()
```

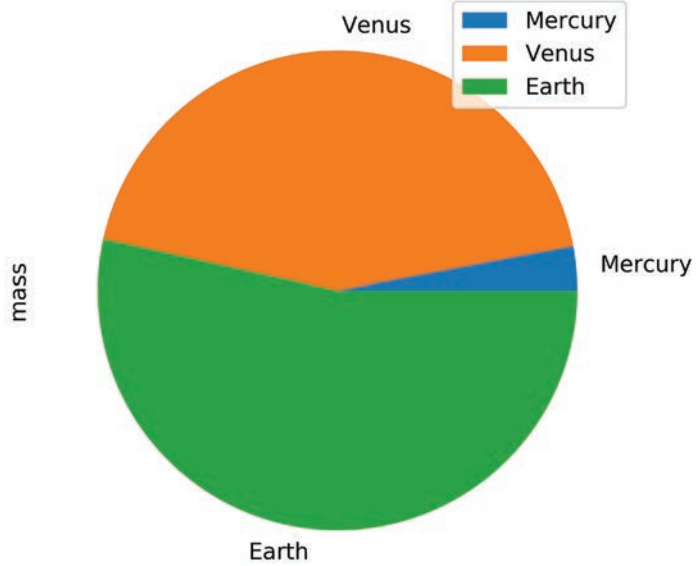


Figure 4.20: Pie chart as output of Program 4-16.

It is important to note that the default label names are the index value of the DataFrame. The labels as shown in Figure 4.20 are the names of the planet which are the index values as shown in Program 4.16.

Program 4-17 Let us consider the dataset of Table 4.10 showing the forest cover of north eastern states that contains geographical area and corresponding forest cover in sq km along with the names of the corresponding states.

Table 4.10 Forest cover of north eastern states

State	GeoArea	ForestCover
Arunachal Pradesh	83743	67353
Assam	78438	27692
Manipur	22327	17280
Meghalaya	22429	17321
Mizoram	21081	19240
Nagaland	16579	13464
Tripura	10486	8073

```
Program 4-17
import pandas as pd
import matplotlib.pyplot as plt
df=pd.DataFrame({'GeoArea':[83743,78438,22327,22429,21081,16579,10
486],'ForestCover':[67353,27692,17280,17321,19240,13464,8073]},
     index=['Arunachal Pradesh','Assam','Manipur','Meghalaya',
     'Mizoram','Nagaland','Tripura'])
```

```
df.plot(kind='pie',y='ForestCover',
     title='Forest cover of North Eastern
states',legend=False)
plt.show()
```

Think and Reflect

What effect did 'legend= False' in Program 4.17 have on the output?

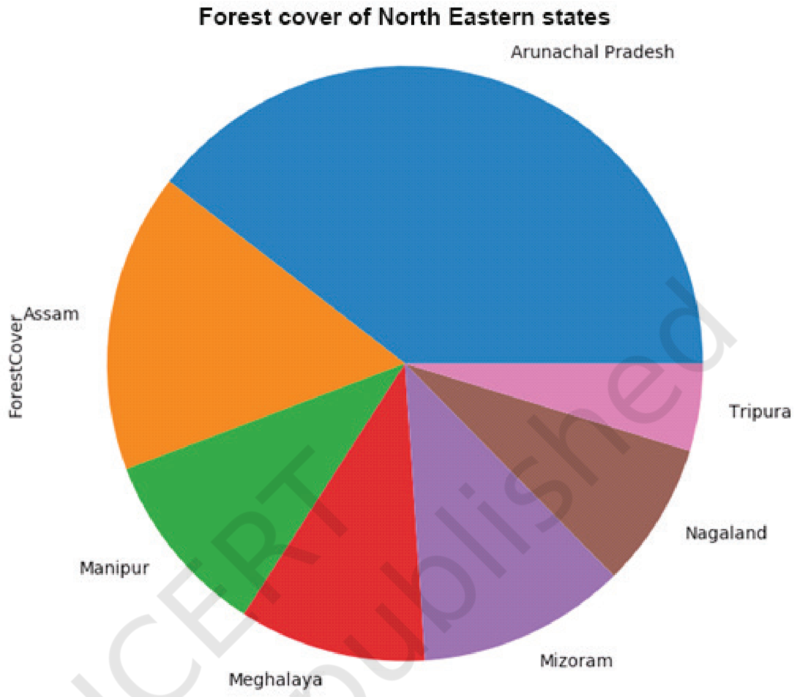


Figure 4.21: Pie chart as output of Program 4.17

Customisation of pie chart

To customise the pie plot of Figure 4.21, we have added the following two properties of pie chart in program 4-18:

- Explode—it specifies the fraction of the radius with which to explode or expand each slot.
- Autopct—to display the percentage of that part as a label.

Program 4-18

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.DataFrame({'GeoArea':[83743,78438,22327,22429,21081,16579,1
0486],'ForestCover':[67353,27692,17280,17321,19240,13464,8073]},
index=['Arunachal Pradesh','Assam','Manipur','Meghalaya', 'Mizoram
','Nagaland','Tripura'])
exp=[0.1,0,0,0,0.2,0,0]
#explode the first wedge to .1 level and fifth to level 2.
c=['r','g','m','c','brown','pink','purple']
```

```
#change the color of each wedge
df.plot(kind='pie',y='ForestCover',title='Forest cover of North
Eastern states', legend=False, explode=exp, autopct="%.2f",
colors=c)
plt.show()
```

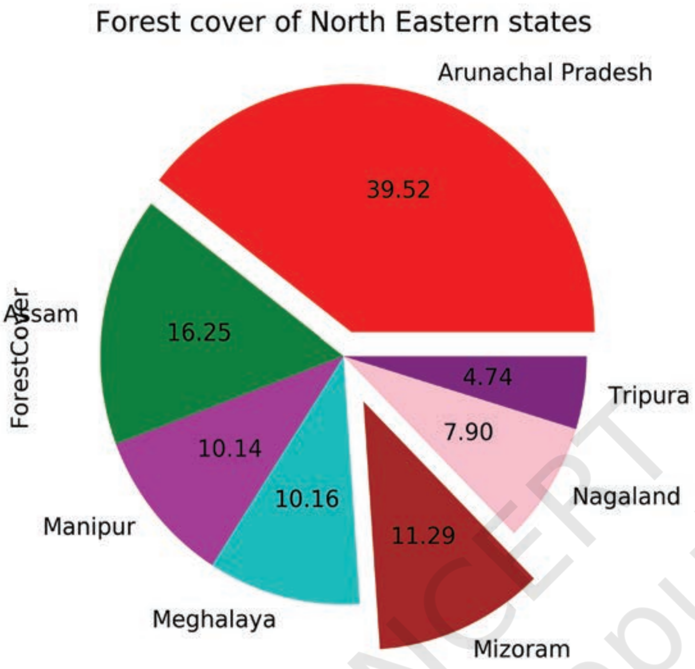


Figure 4.22: Pie chart as output of Program 4.18

Summary

- A plot is a graphical representation of a data set which is also interchangeably known as a graph or chart. It is used to show the relationship between two or more variables.
- In order to be able to use Python's Data Visualisation library, we need to import the pyplot module from Matplotlib library using the following statement: import matplotlib.pyplot as plt, where plt is an alias or an alternative name for matplotlib.pyplot. You can keep any alias of your choice.
- The pyplot module houses functions to create a figure(plot), create a plotting area in a figure, plot lines, bars, hist. etc., in a plotting area, decorate the plot with labels, etc.

NOTES

- The various components of a plot are: Title, Legend, Ticks, x label, ylabel
- plt.plot() is used to build a plot, where plt is an alias.
- plt.show() is used to display the figure, where plt is an alias.
- plt.xlabel() and plt.ylabel() are used to set the x and y label of the plot.
- plt.title() can be used to display the title of a plot.
- It is possible to plot data directly from the DataFrame.
- Pandas has a built-in .plot() function as part of the DataFrame class.
- The general format of plotting a DataFrame is df.plot(kind = ' ') where df is the name of the DataFrame and kind can be line, bar, hist, scatter, box depending upon the type of plot to be displayed.

Exercise

1. What is the purpose of the Matplotlib library?
2. What are some of the major components of any graphs or plot?
3. Name the function which is used to save the plot.
4. Write short notes on different customisation options available with any plot.
5. What is the purpose of a legend?
6. Define Pandas visualisation.
7. What is open data? Name any two websites from which we can download open data.
8. Give an example of data comparison where we can use the scatter plot.
9. Name the plot which displays the statistical summary.

Note: Give appropriate title, set xlabel and ylabel while attempting the following questions.

10. Plot the following data using a line plot:

Day	1	2	3	4	5	6	7
Tickets sold	2000	2800	3000	2500	2300	2500	1000

- Before displaying the plot display "Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday" in place of Day 1, 2, 3, 4, 5, 6, 7
- Change the color of the line to 'Magenta'.

11. Collect data about colleges in Delhi University or any other university of your choice and number of courses they run for Science, Commerce and Humanities, store it in a CSV file and present it using a bar plot.

12. Collect and store data related to the screen time of students in your class separately for boys and girls and present it using a boxplot.

13. Explain the findings of the boxplot of Figure 4.18 by filling the following blanks:

   a) The median for the five subjects is _____ , ______, _______, ______, ______

   b) The highest value for the five subjects is : _____ , ______, _______, ______, ______

   c) The lowest value for the five subjects is : _____ , ______, _______, ______, ______

   d) _____________ subject has two outliers with the value ________ and ________

   e) _____________ subject shows minimum variation

14. Collect the minimum and maximum temperature of your city for a month and present it using a histogram plot.

15. Conduct a class census by preparing a questionnaire. The questionnaire should contain a minimum of five questions. Questions should relate to students, their family members, their class performance, their health etc. Each student is required to fill up the questionnaire. Compile the information in numerical terms (in terms of percentage). Present the information through a bar, scatter–diagram. (NCERT Geography class IX, Page 60)

NOTES

16. Visit data.gov.in , search for the following in “catalogs” option of the website:

- Final population Totals, India and states
- State Wise literacy rate

Download them and create a CSV file containing population data and literacy rate of the respective state. Also add a column Region to the CSV file that should contain the values East, West, North and South. Plot a scatter plot for each region where X axis should be population and Y axis should be Literacy rate. Change the marker to a diamond and size as the square root of the literacy rate.

Group the data on the column region and display a bar chart depicting average literacy rate for each region.

Chapter 5 Internet and Web

> “The internet could be a very positive step towards education, organisation and participation in a meaningful society.”
>
> — Noam Chomsky

12149CH05

In this chapter

5.1 Introduction to Computer Networks

We are living in a connected world. Information is being produced, exchanged, and traced across the globe in real time. It's possible as almost everyone and everything in the digital world is interconnected through one way or the other.

A group of two or more similar things or people interconnected with each other is called network (Figure 5.1). Some of the examples of network in our everyday life include:

- Social network
- Mobile network
- Network of computers
- Airlines, railway, banks, hospitals networks.

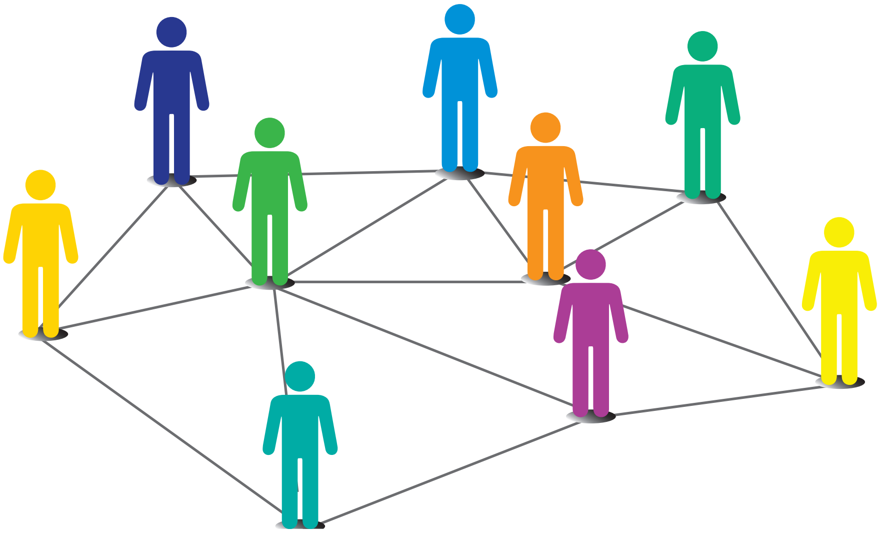


Figure 5.1: Interconnection forming a social network

A computer network (Figure 5.2) is an interconnection among two or more computers or computing devices. Such interconnection allows computers to share data and resources among each other. A basic network may connect a few computers placed in a room.

The network size may vary from small to large depending on the number of computers it connects. A computer network can include different types of hosts (also called nodes) like server, desktop, laptop, cellular phones.

Activity 5.1

Identify some other networks in the real world.

A computer network (Figure 5.2) is an interconnection among two or more computers or computing devices. Such interconnection allows computers to share data and resources among each other. A basic network may connect a few computers placed in a room.

The network size may vary from small to large depending on the number of computers it connects. A computer network can include different types of hosts (also called nodes) like server, desktop, laptop, cellular phones.

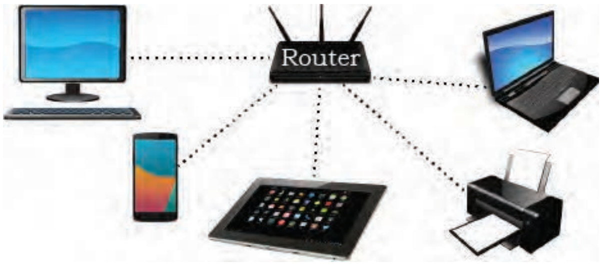


Figure 5.2: A computer network

Apart from computers, networks include networking devices like switch, router, modem, etc. Networking devices are used to connect multiple computers in different settings. For communication, data in a network is divided into smaller chunks called packets. These packets are then carried over a network. Devices in a network can be connected either through wired media like cables or wireless media like air.

In a communication network, each device that is a part of a network and that can receive, create, store or send data to different network routes is called a node. In the context of data communication, a node can be a device such as a modem, hub, bridge, switch, router, digital telephone handset, a printer, a computer or a server.

Interconnectivity of computing devices in a network allows us to exchange information simultaneously with many parties through email, websites, audio/video calls, etc. Network allows sharing of resources. For example, a printer can be made available to multiple computers through a network; a networked storage can be accessed by multiple computers. People often connect their devices through hotspot, thus forming a small personal network.

Activity 5.2

Create a hotspot using a smartphone and connect other devices to it.

5.2 Types of Networks

There are various types of computer networks ranging from network of handheld devices (like mobile phones or tablets) connected through Wi-Fi or Bluetooth within a single room to the millions of computers spread across the globe. Some are connected wireless while others are connected through wires.

Based on the geographical area covered and data transfer rate, computer networks are broadly categorised as:

- LAN (Local Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

5.2.1 Local Area Network (LAN)

It is a network that connects computers, mobile phones, tablet, mouse, printer, etc., placed at a limited distance. The geographical area covered by a LAN can range from a single room, a floor, an office having one or more buildings in the same premise, laboratory, a school, college, or university campus. The connectivity is done by means of wires, Ethernet cables, fibre optics, or Wi-Fi. A Local Area Network (LAN) is shown in Figure 5.3.

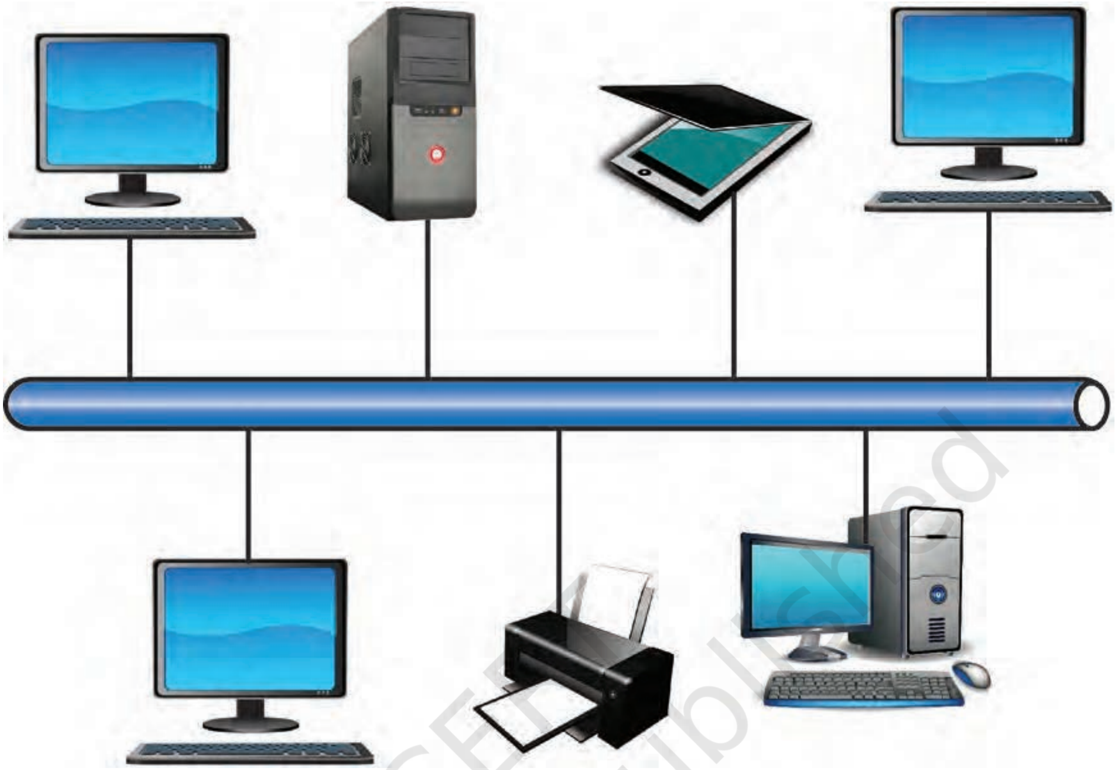


Figure 5.3: A Local Area Network

LAN is comparatively secure as only authentic users in the network can access other computers or shared resources. Users can print documents using a connected printer, upload or download documents and software to and from the local server. Such LANs provide the short range communication with the high speed data transfer rates. These types of networks can be extended up to 1 km. Data transfer in LAN is quite high, and usually varies from 10 Mbps (called Ethernet) to 1000 Mbps (called Gigabit Ethernet), where Mbps stands for Megabits per second. Ethernet is a set of rules that decides how computers and other devices connect with each other through cables in a local area network or LAN.

Think and Reflect

Explore and find out the minimum internet speed required to make a video call.

5.2.2 Metropolitan Area Network (MAN)

Metropolitan Area Network (MAN) is an extended form of LAN which covers a larger geographical area like a city or a town. Data transfer rate in MAN also ranges in Mbps, but it is considerably less as compared to LAN. Cable TV network or cable based broadband internet services are examples of MAN. This kind of network can be extended up to 30–40 km. Sometimes, many LANs are connected together to form MAN, as shown in Figure 5.4.

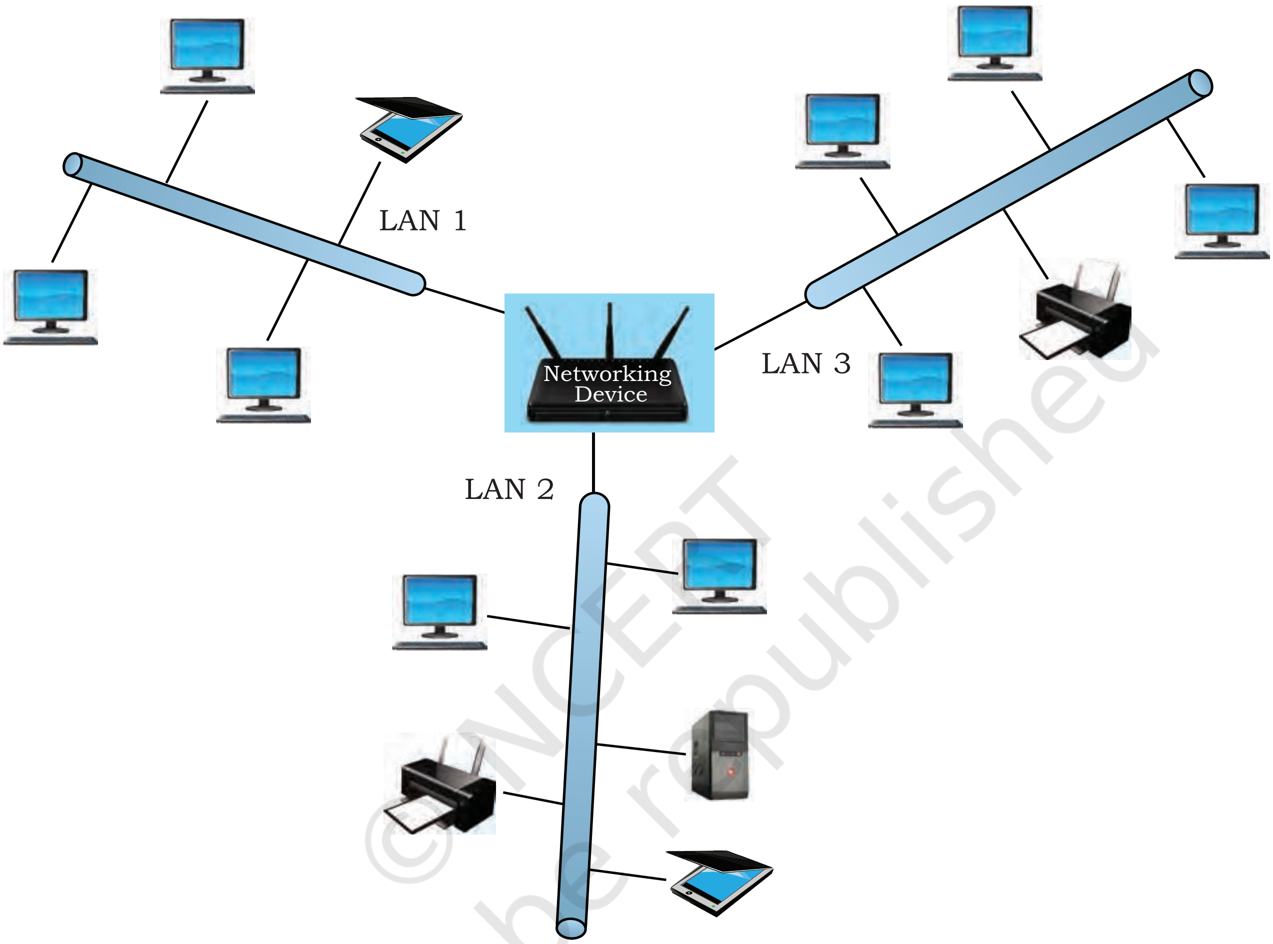


Figure 5.4: A Metropolitan Area Network

5.2.3 Wide Area Network (WAN)

Wide Area Network (WAN) connects computers and others LANs and MANs, which are spread across different geographical locations of a country or in different countries or continents. A WAN could be formed by connecting a LAN to other LANs (Figure 5.5) via wired or wireless media. Large business, educational and government organisations connect their different branches in different locations across the world through WAN. The Internet is the largest WAN that connects billions of computers, smartphones and millions of LANs from different continents.

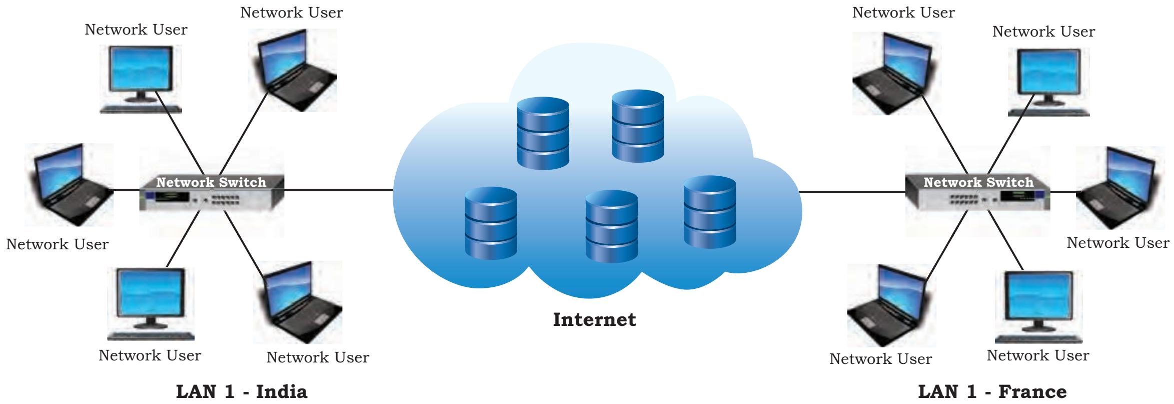


Figure 5.5: A Wide Area Network

5.3 Network Devices

To communicate data through different transmission media and to configure networks with different functionality, we require different devices like Modem, Hub, Switch, Repeater, Router, Gateway, etc. Let us explore them in detail.

Think and Reflect

It is possible to access your bank account from any part of the world. Find if the bank's network is a LAN, MAN, WAN or any other type.

5.3.1 Modem

Modem stands for 'MOdulator DEMolulator'. It refers to a device used for conversion between analog signals and digital bits. We know computers store and process data in terms of 0s and 1s. However, to transmit data from a sender to a receiver, or while browsing the internet, digital data are converted to an analog signal and the medium (be it free-space or a physical media) carries the signal to the receiver. There are modems connected to both the source and destination nodes. The modem at the sender's end acts as a modulator that converts the digital data into analog signals. The modem at the receiver's end acts as a demodulator that converts the analog signals into digital data for the destination node to understand. Figure 5.6 shows connectivity using a modem.

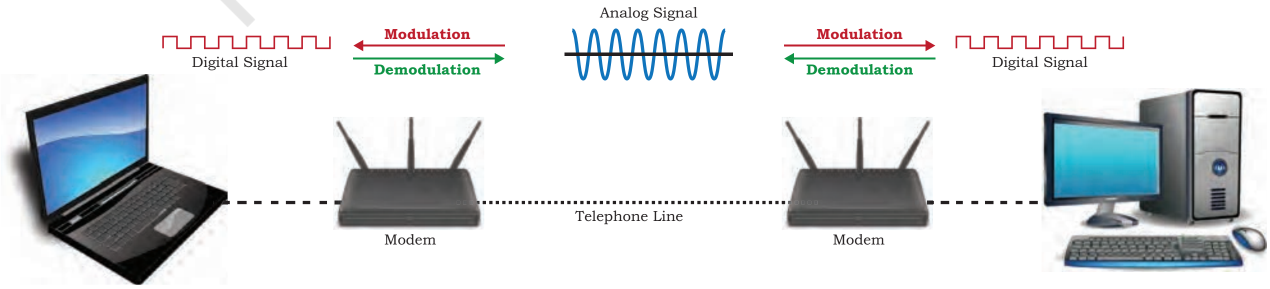


Figure 5.6: Use of modem

5.3.2 Ethernet Card

Ethernet card, also known as Network Interface Card (NIC card in short) is a network adaptor used to set up a wired network. It acts as an interface between computer and the network. It is a circuit board mounted on the motherboard of a computer as shown in Figure 5.7. The Ethernet cable connects the computer to the network through NIC. Ethernet cards can support data transfer between 10 Mbps and 1 Gbps (1000 Mbps). Each NIC has a MAC address, which helps in uniquely identifying the computer on the network.

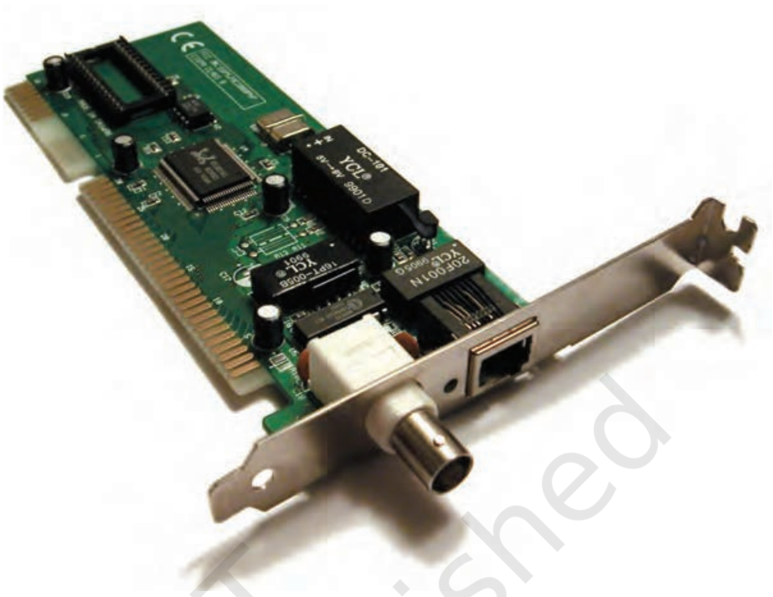


Figure 5.7: A Network Interface Card

5.3.3 Repeater

Data are carried in the form of signals over the cable. These signals can travel a specified distance (usually about 100 m). Signals lose their strength beyond this limit and become weak. In such conditions, original signals need to be regenerated.

A repeater is an analog device that works with signals on the cables to which it is connected. The weakened signal appearing on the cable is regenerated and put back on the cable by a repeater.

5.3.4 Hub

An Ethernet hub (Figure 5.8) is a network device used to connect different devices through wires. Data arriving on any of the lines are sent out on all the others. The limitation of hub is that if data from two devices come at the same time, they will collide.

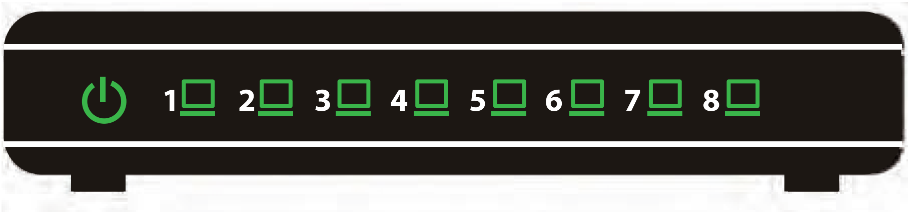


Figure 5.8: A network hub with 8 ports

NOTES

5.3.5 Switch

A switch is a networking device (Figure 5.9) that plays a central role in a Local Area Network (LAN). Like a hub, a network switch is used to connect multiple computers or communicating devices. When data arrives, the switch extracts the destination address from the data packet and looks it up in a table to see where to send the packet. Thus it sends signals to only selected devices instead of sending to all. It can forward multiple packets at the same time. A switch does not forward the signals which are noisy or corrupted. It drops such signals and asks the sender to resend it.

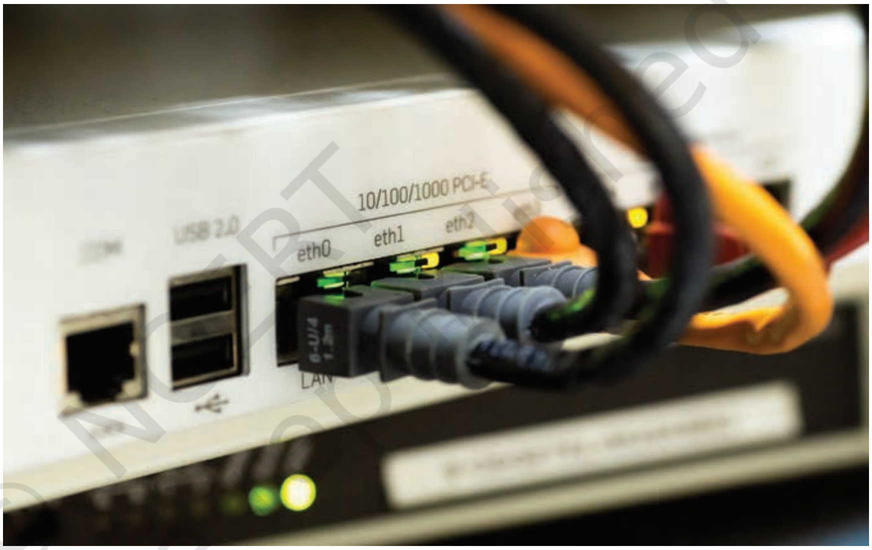


Figure 5.9: Cables connected to a network switch

Ethernet switches are common in homes and offices to connect multiple devices, thus creating LANs or to access the Internet.

5.3.6 Router

A router (Figure 5.10) is a network device that can receive the data, analyse it and transmit it to other networks. A router connects a local area network to the internet. Compared to a hub or a switch, a router has advanced capabilities as it can analyse the data being carried over a network, decide or alter how it is packaged, and send it to another network of a different type. For example, data has been divided into packets of a certain size. Suppose, these packets are to be carried over a different type of network which cannot handle bigger packets, in such a case, the data is to be repackaged as smaller packets and then sent over the network by a router.

Figure 5.10: A Router

A router can be wired or wireless. A wireless router can provide Wi-Fi access to smartphones and other devices. Usually, such routers also contain some ports to provide wired Internet access. These days, home Wi-Fi routers perform the dual task of a router and a modem or switch. These routers connect to incoming broadband lines, from ISP (Internet Service Provider), and convert them to digital data for computing devices to process.

An Internet service provider (ISP) is any organisation that provides services for accessing the Internet.

5.3.7 Gateway

As the term "Gateway" suggests, it is a key access point that acts as a "gate" between an organisation's network and the outside world of the Internet (Figure 5.11). Gateway serves as the entry and exit point of a network, as all data coming in or going out of a network must first pass through the gateway in order to use routing paths. Besides routing data packets, gateways also maintain information about the host network's internal connection paths and the identified paths of other remote networks. If a node from one network wants to communicate with a node of a foreign network, it will pass the data packet to the gateway, which then routes it to the destination using the best possible route.

For simple Internet connectivity at homes, the gateway is usually the Internet Service Provider that provides access to the entire Internet. Generally, a router is configured to work as a gateway device in computer networks. A gateway can be implemented as software, hardware, or a combination of both. This is because a

Activity 5.3

Find and list a few ISPs in your region.

network gateway is placed at the edge of a network and the firewall is usually integrated with it.

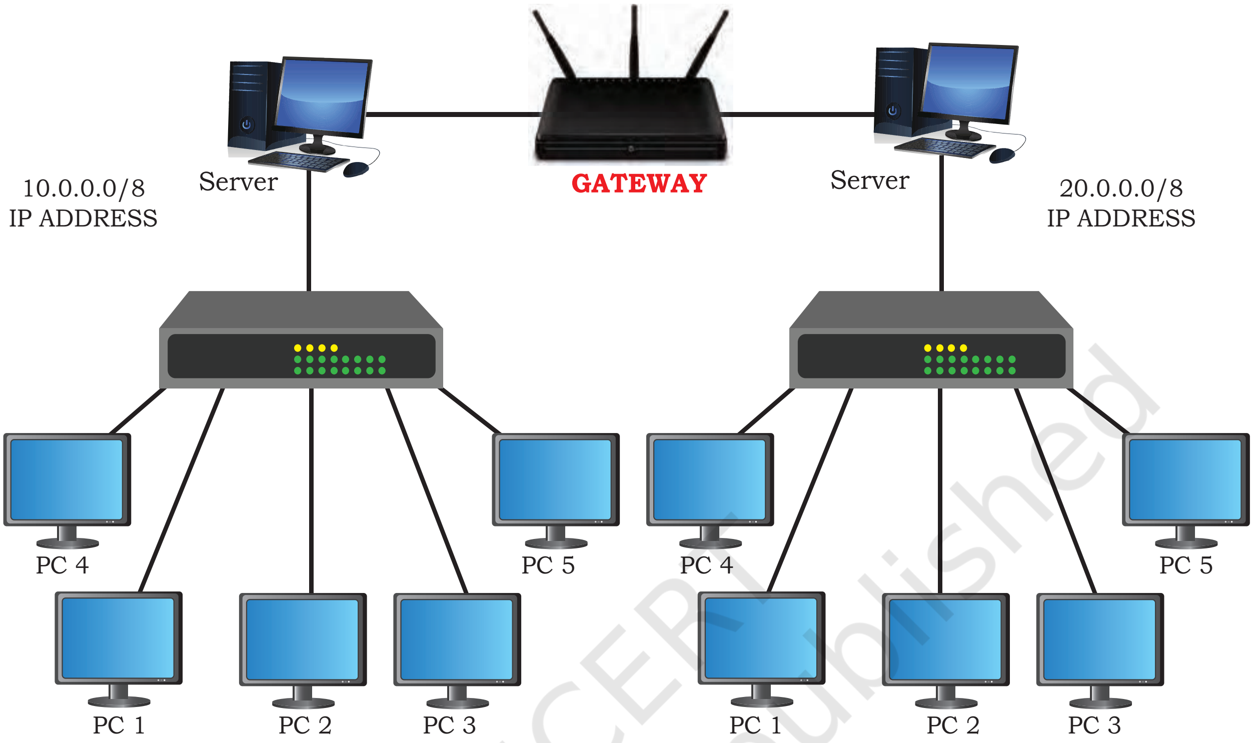


Figure 5.11: A network gateway

5.4 NETWORKING TOPOLOGIES

We have already discussed that a number of computing devices are connected together to form a Local Area Network (LAN), and interconnections among millions of LANs forms the Internet. The arrangement of computers and other peripherals in a network is called its topology. Common network topologies are mesh, ring, bus, star and tree.

5.4.1 Mesh Topology

In this networking topology, each communicating device is connected with every other device in the network as shown in Figure 5.12. Such a network can handle large amounts of traffic since multiple nodes can transmit data simultaneously. Also, such networks are more reliable in the sense that even if a node gets down, it does not cause any break in the transmission of data between other nodes. This topology is also more secure as compared to other topologies because each cable between two nodes carries different data. However, wiring is complex and cabling cost is high in creating such networks, and there are many redundant or unutilised connections.

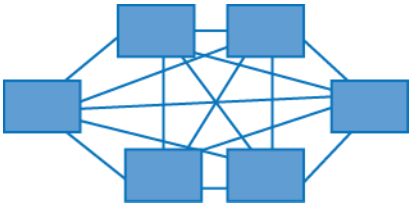


Figure 5.12: A mesh topology

5.4.2 Ring Topology

In ring topology, each node is connected to two other devices, one each on either side, as shown in Figure 5.13. The nodes connected with each other thus form a ring. The link in a ring topology is unidirectional. Thus, data can be transmitted in one direction only (clockwise or counterclockwise).

To build a fully-connected mesh topology of n nodes, it requires n(n-1)/2 wires.

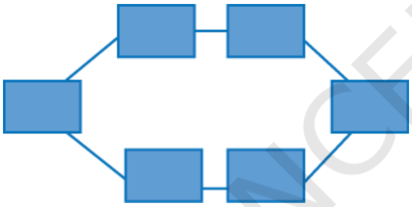


Figure 5.13: A ring topology

5.4.3 Bus Topology

In bus topology (Figure 5.14), each communicating device connects to a transmission medium, known as bus. Data sent from a node are passed on to the bus and hence are transmitted to the length of the bus in both directions. That means data can be received by any of the nodes connected to the bus.

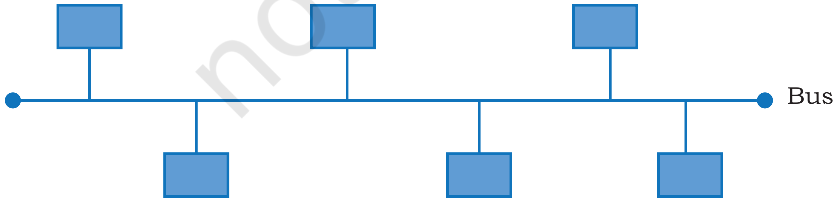


Figure 5.14: A bus topology

In this topology, a single backbone wire called bus is shared among the nodes, which makes it cheaper and easy to maintain. Both ring and bus topologies are considered to be less secure and less reliable.

Think and Reflect

How will a bus and ring topology behave in case a node is down?

5.4.4 Star Topology

In star topology, each communicating device is connected to a central node, which is a networking device like a hub or a switch, as shown in Figure 5.15.

Star topology is considered very effective, efficient and fast as each device is directly connected with the central device. Although disturbance in one device will not affect the rest of the network, any failure in the central networking device may lead to the failure of complete network.

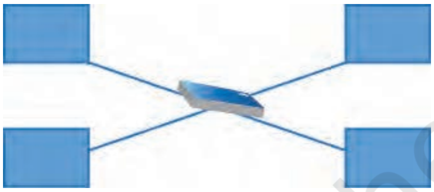


Figure 5.15: A star topology

The central node can be either a broadcasting device means data will be transmitted to all the nodes in the network, or a unicast device means the node can identify the destination and forward data to that node only.

5.4.5 Tree or Hybrid Topology

It is a hierarchical topology, in which there are multiple branches and each branch can have one or more basic topologies like star, ring and bus. Such topologies are usually realised in WANs where multiple LANs are connected. Those LANs may be in the form of ring, bus or star. In Figure 5.16, a hybrid topology is shown connecting 4 star topologies in bus.

In this type of network, data transmitted from source first reaches the centralised device and from there the data passes through every branch where each branch can have link for more nodes.

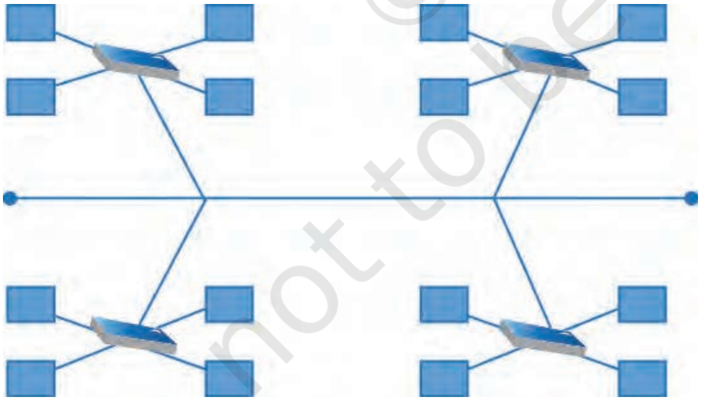


Figure 5.16: A hybrid topology

5.5 THE INTERNET

The Internet is the global network of computing devices including desktop, laptop, servers, tablets, mobile phones, other handheld devices as well as peripheral devices such as printers, scanners, etc. In addition, it also consists of networking devices such as routers, switches, gateways, etc. Today, smart electronic appliances like TV, AC, refrigerator, fan, light, etc., can also communicate through the Internet. The list of such smart devices are always increasing e.g., drones, vehicles, door lock, security camera, etc.

The Internet is evolving everyday. Computers are either connected to a modem through a cable or wirelessly (Wi-Fi). A modem, be it wired or wireless, is connected to a local Internet Service Provider (ISP) who then connects to a national network. Many such ISPs connect together forming a regional network and regional networks connect together forming a national network, and such country-wise networks form the Internet backbone.

The Internet today is a widespread network, and its influence is no longer limited to the technical fields of computer communications. It is being used by everyone in the society as is evident from the increasing use of online tools for education, creativity, entertainment, socialisation and e-commerce.

5.6 APPLICATIONS OF INTERNET

Following are some of the broad areas or services provided through Internet:

- The World Wide Web (WWW)
- Electronic mail (Email)
- Chat
- Voice Over Internet Protocol (VoIP)

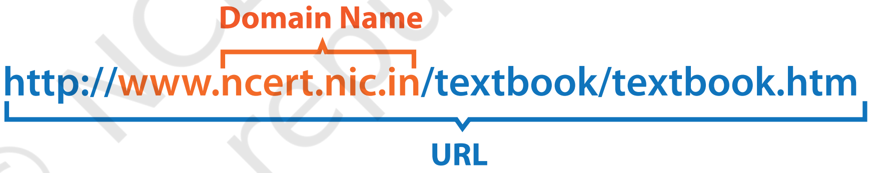
5.6.1 The World Wide Web (WWW)

The World Wide Web (WWW) or web in short, is an ocean of information, stored in the form of trillions of interlinked web pages and web resources. The resources on the web can be shared or accessed through the Internet.

Earlier, to access files residing in different computers, one had to login individually to each computer through the Internet. Besides, files in different computers were sometimes in different formats, and it was difficult to understand each other's files and documents. Sir Tim Berners-Lee — a British computer scientist invented the revolutionary World Wide Web in 1990 by defining three fundamental technologies that lead to creation of web:

- HTML — HyperText Markup Language or HTML is a language which is used to design standardised Web Pages so that the Web contents can be read and understood from any computer across the globe. It uses tags to define the way page content should be displayed by the web browser. Basic structure of every webpage is designed using HTML.
- URI — Uniform Resource Identifier or URI is a unique identifier to identify a resource located on the web. URI identifies a resource (hardware or software) either by its location or by its name or by both.

  URL is Uniform Resource Locator and provides the location and mechanism (protocol) to access the resource. Examples of URI identifying resources using location (i.e., URL) are: https://www.mhrd.gov.in, http://www.ncert.nic.in, http://www.airindia.in, etc. URL is sometimes also called a web address. However, it is not only the domain name, but contains other information that completes a web address, as depicted below:



In the above URL, http is the protocol name, it can be https, http, FTP, Telnet, etc. www is a subdomain. ncert.nic.in is the domain name.

Search Engine(s) like google.co.in, bing.com, duckduckgo.com, in.yahoo.com, etc., can be used to search and retrieve information when the address of the web page is not known.

Note: These days it is not mandatory to mention protocol and subdomain while entering a URL. The browser automatically prefixes it.

- HTTP — The HyperText Transfer Protocol is a set of rules which is used to retrieve linked web pages across the web. It's more secure and advanced version is HTTPS.

  Many people confuse the web with the Internet. The Internet as we know is the huge global network of interconnected computers, which may or may not have any file or webpage to share with the world. The web on the other hand is the interlinking of a collection of WebPages on these computers which are accessible over the Internet. WWW today gives users access to a vast collection of information created and shared by people across the world. It is today the most popular information retrieval system.

5.6.2 Electronic Mail (Email)

Email is the short form of electronic mail. It is one of the ways of sending and receiving message(s) using the Internet. An email can be sent anytime to any number of recipients at anywhere. The message can be either text entered directly onto the email application or an attached file (text, image audio, video, etc.) stored on a secondary storage. An existing file can be sent as an attachment with the email, so no need to type it again.

To use email service, one needs to register with an email service provider by creating a mail account. These services may be free or paid. Some of the popular email service providers are Google (gmail), Yahoo (yahoo mail), Microsoft (outlook), etc. However, many organisations nowadays get customised business email addresses for their staff using their own domain name. For example, username@companyname.com.

Following are some of the common facilities available for an email user:

1. Creating an email, attaching files with an email, saving an email as draft for mailing later. Creating email is also termed as composing.
2. Sending and receiving mail. Same email can be sent to multiple email addresses, simultaneously.
3. Sending the copy of mail, as carbon copy (cc) or blind carbon copy (bcc).
4. Forwarding a received email to other user(s)
5. Filtering spam emails
6. Organising email in folders and sub folders
7. Creating and managing email ids of the people you know.
8. Setting signature/footer to be inserted automatically at the end of each email
9. Printing emails using a printer or saving as files.
10. Searching emails using email address or email subject text

5.6.3 Chat

Chatting or Instant Messaging (IM) over the Internet means communicating to people at different geographic locations in real time through text message(s). It is a

NOTES

NOTES

forum where multiple people connect to each other, to discuss their common interests. Two individuals can also send messages instantly. The sender types a message and sends it; the receiver immediately receives the message and can read and revert through text message. All this happens in real time, as if the sender and receiver were sitting in the same place. For a successful chat session, the communicating parties should be online simultaneously, and use the same chat application.

With ever increasing internet speed, it is now possible to send image, document, audio, video as well through instant messengers. It means, the communicating parties can talk to each other through an audio call or through a video call. Moreover, it is also possible to chat through text, audio and video in a group. Thus, we can have group chat or group calls.

Applications such as WhatsApp, Slack, Skype, Yahoo Messenger, Google Talk, Facebook Messenger, Google Hangout, etc., are examples of instant messengers. Some of these applications support instant messaging through all the modes — text, audio and video.

5.6.4 VoIP

Voice over Internet Protocol or VoIP, allows us to have voice call (telephone service) over the Internet, i.e., the voice transmission over a computer network rather than through the regular telephone network. It is also known as Internet Telephony or Broadband Telephony. But to avail the phone service over the Internet, we need to have an Internet connection with reasonably good speed.

VoIP works on the simple principle of converting the analogue voice signals into digital and then transmitting them over the broadband line. There are two major advantages of a VoIP—

- These services are either free or very economical, so people use them to save on cost. That is why these days even international calls are being made using VoIP.
- VoIP call(s) can be received and made using IP phones from any place having Internet access. Hence, VoIP has increased the portability and functionality of the voice calling system. Incoming phone calls can be

automatically routed to the VoIP phone as soon as it is connected to the Internet.

The only disadvantage of VoIP is that its call quality is dependent on Internet connection speed. Slow Internet connection will lead to poor quality voice calls.

5.7 Website

Each one of us might have visited one or the other website. A website in general contains information organised in multiple pages about an organisation. A website can also be created for a particular purpose, theme or to provide a service.

A website (usually referred to as a site in short) is a collection of web pages related through hyperlinks, and saved on a web server. A visitor navigates from one page to another by clicking on hyperlinks. Also, all the pages of a website are integrated under one domain name and have a common theme and template. For example, the website of NCERT will have all the pages related to NCERT, viz., textbooks, syllabus, events, resource materials, etc., under one domain name and having a common design theme. To access a website, one has to type the address of the website (URL) in the address bar of a browser, and press enter. The home page of the website will be displayed.

5.7.1 Purpose of a Website

We are living in an Internet era where the whole world is connected. A website's purpose is to make the information available to people at large. For example, a company might like to advertise or sell its products, a government organisation may like to publish circulars, float tenders, invite applications or get feedback from various stakeholders. A website is a means that helps to communicate with people in a specific, transparent and user friendly manner. Therefore, while developing a website, the first question to ask is why the website is being created, and what should be its pages so that it serves the required purpose.

Basically, a website should be user friendly and provide information to users with minimum efforts. A website should be designed keeping in mind different categories of people that will be visiting the site. Some of the common purposes for which websites are designed are listed below:

- Selling products and delivering services
- Posting and finding information on the internet
- Communicating with each other
- Entertainment purposes
- Disseminating contents and software

Activity 5.4

Visit NCERT, SWAYAM or any other website and note down URLs of some of the specific pages of that website.

5.8 Web Page

A web page (also referred to as a page) is a document on the WWW that is viewed in a web browser. Basic structure of a web page is created using HTML (HyperText Markup Language) and CSS (Cascaded Style Sheet). A web page is usually a part of a website and may contain information in different forms, such as:

- text in the form of paragraphs, lists, tables, etc.
- images
- audio
- video
- software application
- other interactive contents

Additionally, various styling and formatting are applied on a web page to make it attractive and organised. Further, program codes called scripts are used to define the manner in which the page will behave on different actions. Scripts make a web page interactive. JavaScript is the most popular and commonly used scripting language. However, Python and PHP are also used to apply scripting on a web page.

The first page of the website is called a home page. It generally contains information and links to all the related web pages. Each web page has a unique address that is visible on the address bar. Hence if we want to view a particular web page, its address has to be typed in the address bar of the browser. The web pages that are linked to form a website share a unique domain name. For example, https://swayam.gov.in/ is a website by the Government of India to deliver online courses for School, College and University students and teachers. It is a collection of multiple web pages that link to different courses related information.

5.8.1 Static and Dynamic Web Pages

A web page can be static or dynamic. A static webpage is one whose content always remains static, i.e., does not change for person to person. When a web server receives a request (from browser) for a static web page, it just locates the page on its storage media and sends it to the browser of the client. No additional processing is performed on the page. Hence, a static web page remains the same for all users until someone changes its code manually.

Static web pages are generally written in HTML, JavaScript and/or CSS and have the extension .htm or .html.

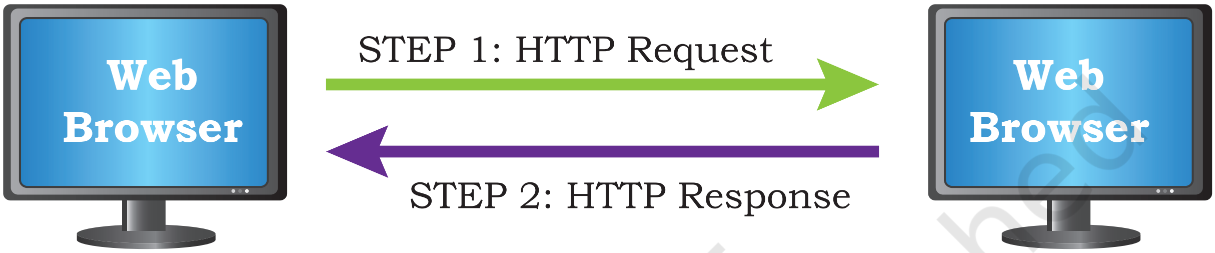


Figure 5.17: Working of a static web page

On the other hand, a dynamic web page is one in which the content of the web page can be different for different users. The difference in content may be because of different choices made by the user. When a request for a dynamic web page is made to the web server, it does not simply retrieve the page and send. Before sending the requested web page, the server may perform some additional processes like getting information from the database, updating date and time, updating weather information, etc. The content of such pages changes frequently. They are more complex and thus take more time to load than static web pages.

Dynamic web pages can be created using various languages such as JavaScript, PHP, ASP.NET, Python, Java, Ruby, etc. These are complex to construct and design, as the code to perform the additional operations has to be added. Such server side code allows the server to change its content each time the page is loaded. Further, most dynamic pages are linked to databases so that each time the page is uploaded, the required information from the databases is retrieved to update the web page. Few common examples of dynamic web pages are those web pages displaying the date, time, and weather report or having e-commerce applications.

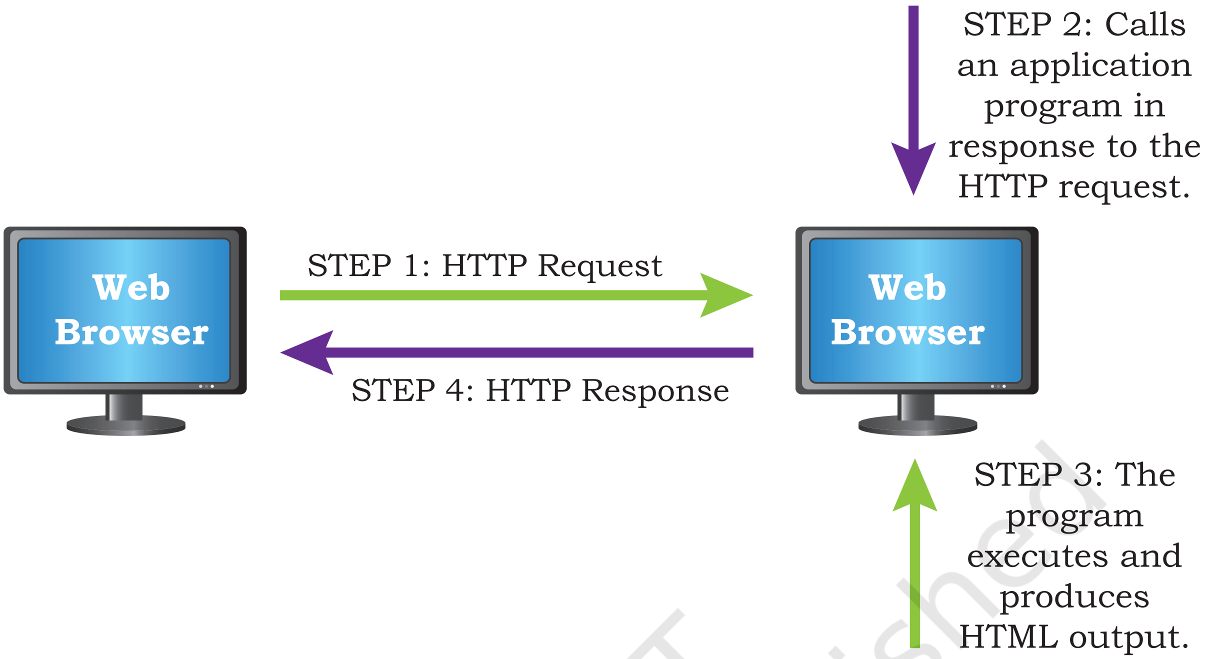

STEP 2: Calls an application program in response to the HTTP request.
Web Browser
STEP 1: HTTP Request
STEP 4: HTTP Response
Web Browser
STEP 3: The program executes and produces HTML output.

Figure 5.18: Working of a dynamic web page

5.9 WEB SERVER

A web server is used to store and deliver the contents of a website to clients such as a browser that request it. A web server can be software or hardware.

When talking about a web server as computer hardware, it stores web server software and a website's contents (HTML pages, images, CSS stylesheets, and JavaScript files). The server needs to be connected to the Internet so that its contents can be made accessible to others.

When talking about a web server as a software, it is a specialised program that understands URLs or web addresses coming as requests from browsers, and responds to those requests. The server is assigned a unique domain name so that it can be accessed from anywhere using the domain name. To develop and test a website using a personal computer, we need to first install a web server on that computer.

The web browser from the client computer sends a request (HTTP request) for a page containing the desired data or service. The web server then accepts, interprets, searches and responds (HTTP response) to the request made by the web browser. The requested web page is then displayed in the browser of the client. If the server is not able to locate the page, it sends a page containing the error message (Error 404 – page not found) to the client's browser.

5.10 Hosting of a Website

Web hosting is a service that allows us to put a website or a web page onto the Internet, and make it a part of the World Wide Web. Once a website is created using a hardware server, we need to connect it to the Internet so that users across the globe can access. On the other hand, we can rent server resources (CPU, RAM, and storage) from a cloud service provider and host our locally created website there. This is done by uploading the files constituting the website (HTML, CSS, JavaScript, images, databases, etc.) from the local computer onto the space allocated on the server. For this, we have to avail the services of a web hosting service provider. These services for using the server's resources such as RAM, hard disk, bandwidth, etc., are usually paid and these resources can be increased or decreased as per the loads on the website.

A web server whether it is a local server or a cloud server when connected to the Internet is assigned a unique numeric address on the Internet called IP address. This IP address needs to be mapped to a textual name called domain name of the website. This is because it is not convenient for users to remember a numeric IP address. Thus, for accessing a website, the user enters the domain through a browser (URL). The domain name has to be registered (purchased) with an authorised agency.

Activity 5.5

Find out some of the Web hosting service providers from both categories — free and paid.

5.10.1 How to host a website?

To host a website, follow the steps given below:

- Select the web hosting service provider that will provide the web server space as well as related technologies and services such as database, bandwidth, data backup, firewall support, email service, etc. This has to be done keeping in mind the features and services that we want to offer through our website.
- Identify a domain name, which best suits our requirement, and get it registered through domain name Registrar.
- Once we get web space, create logins with appropriate rights and note down IP address to manage web space.

- Upload the files in properly organised folders on the allocated space.
- Get domain name mapped to the IP address of the web server.

The domain name system (DNS) is a service that does the mapping between domain name and IP address. When the address of a website is entered in a browser, the DNS finds out the IP address of the server corresponding to the requested domain name and sends the request to that server.

5.11 Browser

A browser is a software application that helps us to view the web page(s). In other words, it helps us to view the data or information that is retrieved from various web servers on the Internet. Some of the commonly used web browsers are Google Chrome, Internet Explorer, Mozilla Firefox, Opera, etc. A web browser essentially displays the HTML documents which may include text, images, audio, video and hyperlinks that help to navigate from one web page to another.

Figure 5.19: Some commonly used browsers

Mosaic was the first web browser developed by the National Centre for Supercomputing Application (NCSA).

The initial web browsers like Mosaic used to support HTML documents containing plain text (static website) only, but nowadays with the advancement of technology, modern web browsers allow us to view interactive and dynamic websites. In addition to this, most modern browsers allow a wide range of visual effects, use encryption for advanced security and also have cookies that can store the browser settings and data.

5.11.1 Browser Settings

Every web browser has got certain settings that define the manner in which the browser will behave. These settings may be with respect to privacy, search engine preferences, download options, auto signature, autofill and autocomplete feature, theme and much more. Each browser application allows us to change or customise its settings in a user friendly manner. Let's learn how to change the browser settings using the open source browser, Mozilla Firefox.

Open Mozilla Firefox, and on the top right corner of the browser window, click the Menu button.

Mozilla Firefox is an open source web browser which is available free of cost and can be easily downloaded from the Internet.

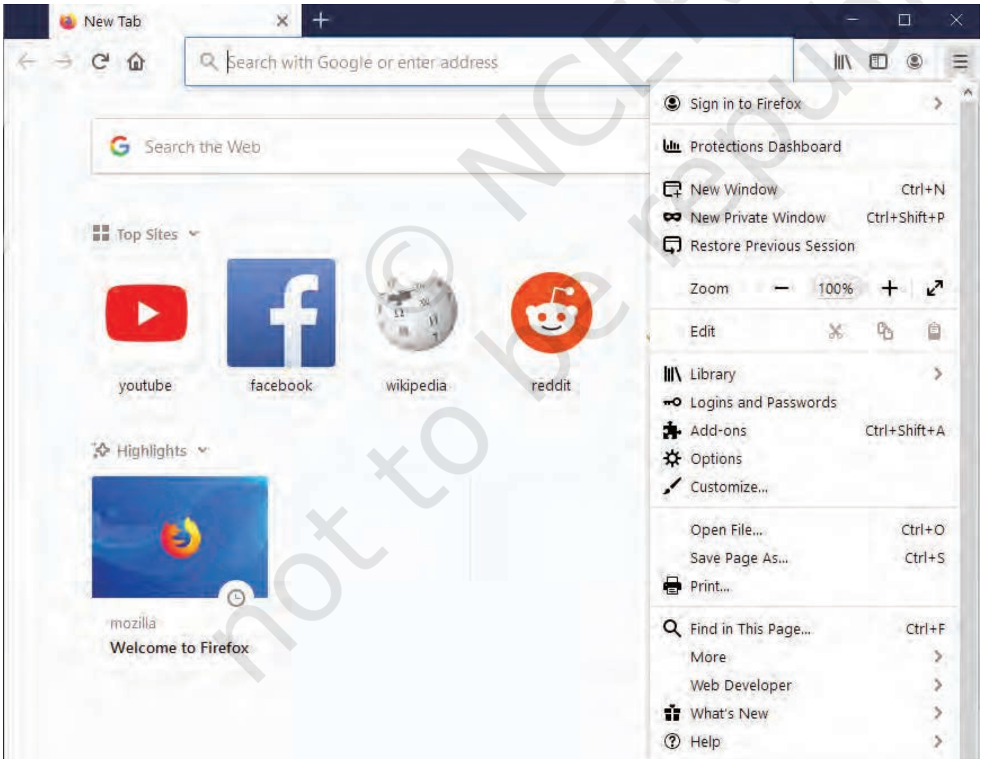
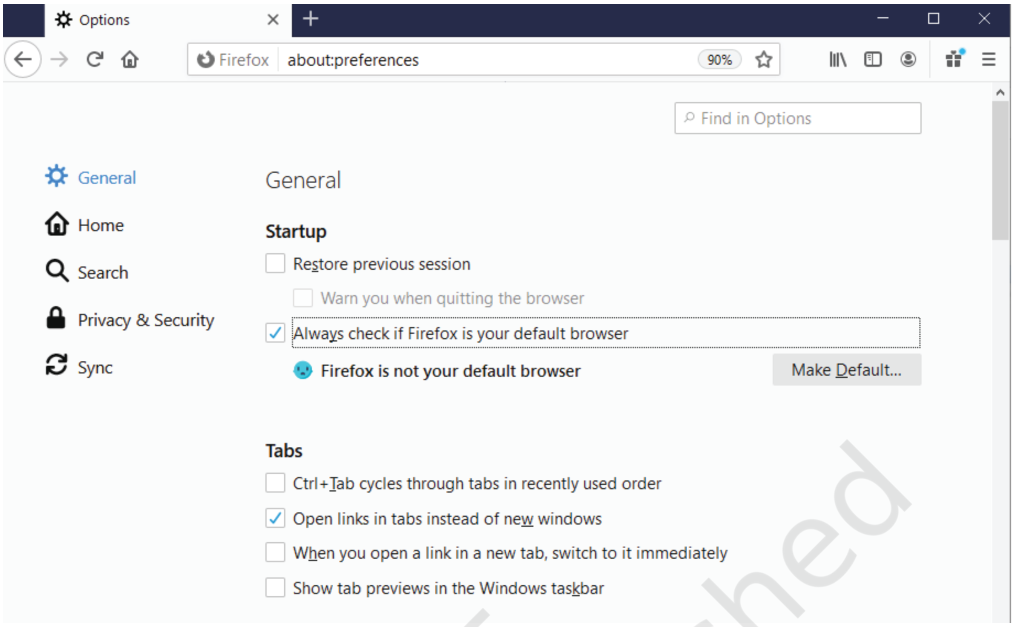


Figure 5.20: Mozilla Firefox Menu button

From the drop down button, select Options. The preferences and Options window will be displayed in the browser.



General

Startup

□ Restore previous session

□ Warn you when quitting the browser

☑ Always check if Firefox is your default browser

Firefox is not your default browser — Make Default...

Tabs

□ Ctrl+Tab cycles through tabs in recently used order

☑ Open links in tabs instead of new windows

□ When you open a link in a new tab, switch to it immediately

□ Show tab previews in the Windows taskbar

Figure 5.21: Preference and options page

On the left side, there are multiple Panels to choose from: General, Home, Search, Privacy and Security and Sync.

General Panel: Some of the options that the panel contains are as follows:

- setting the default browser
- language and appearance of text
- downloading files and applications
- firefox update settings
- browsing and network settings

Home Panel: This panel contains options to set the home page of the browser, browser window and tab settings.

Search Panel: This panel contains options to edit the settings of the search engine used by Firefox.

Privacy and Security Panel: This panel contains options to secure the browser and data. It includes the following:

- enhanced tracking protection
- forms and passwords
- history and address bar
- cookies and site data
- permission to view pop ups windows and install add-ons

Sync Panel: This panel contains options to set up and manage a Firefox account which is needed to access all services given by Mozilla.

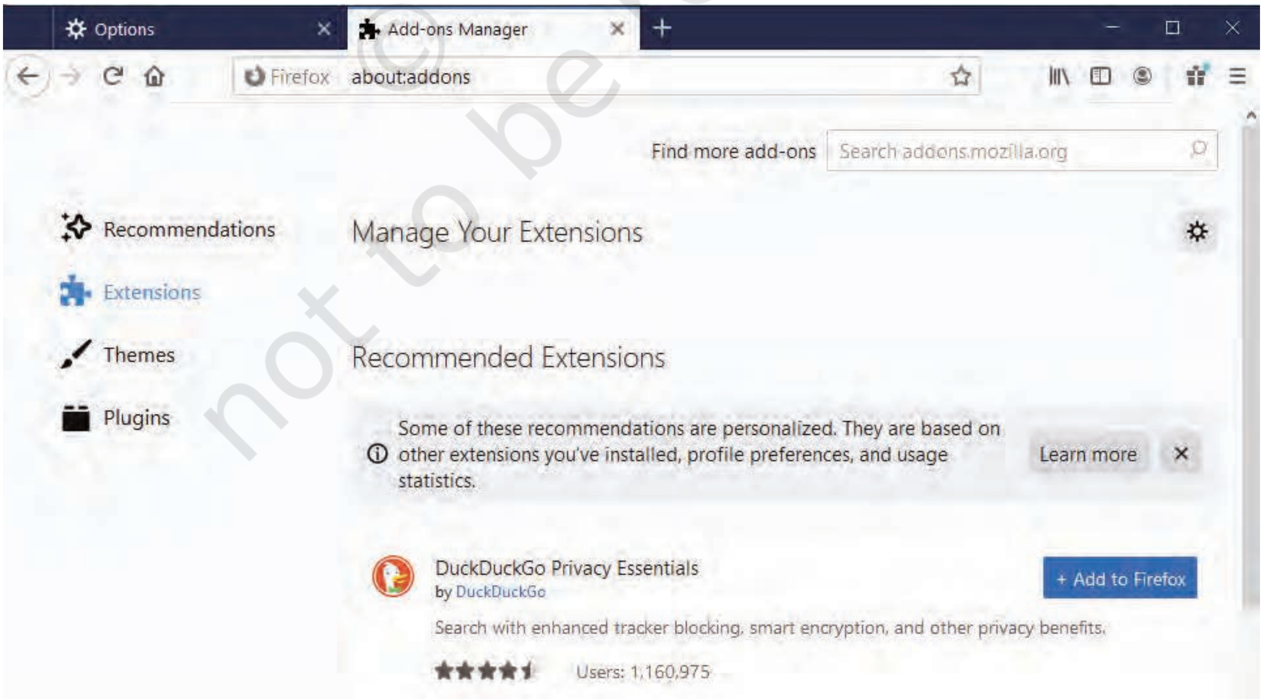
Make the desired settings and close the browser settings window. The changes made in the browser settings will be applied.

5.11.2 Add-Ons and Plug-ins

Add-ons and plug-ins are the tools that help to extend and modify the functionality of the browser. Both the tools boost the performance of the browser, but are different from each other.

A plug-in is a complete program or may be a third-party software. For example, Flash and Java are plug-ins. A Flash player is required to play a video in the browser. A plug-in is a software that is installed on the host computer and can be used by the browser for multiple functionalities and can even be used by other applications as well.

On the other hand, an add-on is not a complete program and so is used to add only a particular functionality to the browser. An add-on is also referred to as extension in some browsers. Adding the functionality of a sound and graphics card is an example of an add-on.



Some of these recommendations are personalized. They are based on other extensions you've installed, profile preferences, and usage statistics.

Search with enhanced tracker blocking, smart encryption, and other privacy benefits.

Figure 5.22: Add-ons and plug-ins

To add an extension, click the Options button on the top right corner of the browser and select the Add-ons option. Click the Extensions Panel option on the left. On the right, options to Manage your Extensions will appear. There will be a list of enabled, disabled and recommended extensions. Make the desired selections and close the add-ons window.

Similarly, to add plug-ins, click Plug-ins options on the left side of the browser window. Make the desired selections to enable or disable the required plug-ins.

Think and Reflect

Can we compare Add-ons and Plug-ins with utility software?

5.11.3 Cookies

A cookie is a text file, containing a string of information, which is transferred by the website to the browser when we browse it. This string of information gets stored in the form of a text file in the browser. The information stored is retransmitted to the server to recognise the user, by identifying pages that were visited, choices that were made while browsing various menu(s) on a particular website. It helps in customising the information that will be displayed, for example the choice of language for browsing, allowing the user to auto login, remembering the shopping preference, displaying advertisements of one's interest, etc.

First cookie software was created in 1994 at Netscape, for determining whether the person is a first time visitor or a re-visitor of their site.

Cookies are usually harmless and they can't access information from the hard disk of a user or transmit virus or malware. It is the browser on our computer which stores and manages the cookies. However, viruses can also be tricked as cookies and cause harm to a computer. One can disable cookies by changing the Privacy and Security settings of our browser.

SUMMARY

- A group of two or more similar things or people interconnected with each other is called network
- A computer network is an interconnection among two or more computers to share data and resources.
- Devices in a network can be connected either through wired or wireless media.

Notes

- Based on the geographical area covered and data transfer rate, computer networks are broadly categorised as LAN, MAN and WAN.
- The protocol or the set of rules that decide functioning of a LAN is called Ethernet.
- Local Area Network (LAN) is a network that connects digital devices placed at a limited distance of upto 1 km.
- Metropolitan Area Network (MAN) is an extended form of LAN which covers a larger geographical area like a city or a town.
- Wide Area Network (WAN) connects computers and other LANs and MANs, which are spread across different geographical locations of a country or in different countries or continents.
- A repeater is an electronic device that receives a weak signal and regenerates it.
- Modem (MOdulator DEMolulator) refers to any such device used for conversion between analog signals and digital bits.
- A hub is a network device used to connect multiple devices to form a network or to connect segment(s) of LAN.
- A switch is a networking device that filters network traffic while connecting multiple computers or communicating devices.
- A router is a network device that can receive the data, analyse it and transmit to other networks.
- A gateway is a device that connects the organisation's network with the outside world of the Internet.
- The physical organisation of computers, cables and other peripherals in a network is called its topology. Common network topologies are Bus, Star, Tree, Mesh, etc.
- In bus topology, each communicating device connects to a common central transmission medium, known as bus.
- In star topology, each communicating device is connected to a central node, which is a networking device like a hub or a switch, through separate cables.

Notes

- In tree topology, multiple star and bus topologies are connected to a central cable, also called the backbone of the network.
- In mesh topology, each communicating device is connected with every other device in the network.
- The Internet is the largest WAN that connects millions of computers across the globe.
- Some of the services provided through the Internet are information sharing, communication, data transfer, social networking, e-commerce, etc.
- A Uniform Resource Locator (URL) is a standard naming convention used for accessing resources over the Internet.
- Electronic mail is a means of sending and receiving message(s) through the Internet.
- Chatting is communicating in real time using text message(s).
- Voice over Internet Protocol (VoIP) allows you to have voice calls over digital networks.
- A website is a collection of related web pages.
- A web page is a document that is viewed in a web browser such as Google Chrome, Mozilla Firefox, Opera, Internet Explorer, etc. It can be static or dynamic.
- A static web page is one whose content does not change for requests made by different people.
- A dynamic web page is one in which the content of the web page displayed is different for different users.
- A web server is a program or a computer that provides services to other programs or computers called clients.
- Web hosting is a service that allows you to post the website created locally so that it is available for all internet users across the globe.
- Every browser has got certain settings that define the manner in which the browser will behave. These settings may be with respect to privacy, search engine preferences, download options, auto signature, autofill and autocomplete feature and much more.

- Add-ons and plug-ins are the tools that help to extend and modify the functionality of the browser.
- A cookie is a text file containing a string of information which stores browsing information on the hard disk of your computer.

Exercise

1. Fill in the blanks:

   a) To transmit data for sharing on a network, it has to be divided into smaller chunks called ________________.

   b) The set of rules that decide the functioning of a network is called ______________.

   c) A LAN can be extended up to a distance of _________ km.

   d) The _________________ connects a local area network to the internet.

   e) The ____________ topology is of hierarchical nature.

   f) __________________ is a standard naming convention used for accessing resources over the Internet.

   g) _____________ is a collection of related web pages.

   h) A ____________ is a computer that provides services to other programs or computers.

2. Expand the following:

   a) ARPANET

   b) ISP

   c) URL

3. Name the device for the following:

   a) It stands for Modulator Demodulator

   b) It regenerates the signals.

4. Differentiate between:

   a) MAN and WAN

   b) Website and web page

   c) Router and Gateway

NOTES

   d) Bus and Star topology
   e) Static and Dynamic web pages

5. Define a network. What is the need of forming a network?
6. Give any two examples of networks.
7. Give any three applications on the Internet.
8. Name any two mail service providers.
9. Explain VoIP.
10. What is DNS?
11. Identify the type of topology from the following:
    a) Each node is connected with the help of a single cable.
    b) Each node is connected with central switching through independent cables.
12. Sahil, a Class X student, has just started understanding the basics of Internet and web technologies. He is a bit confused in between the terms "World Wide Web" and "Internet". Help him in understanding both the terms with the help of suitable examples of each.
13. Murugan wants to send a report on his trip to the North East to his mentor. The report contains images and videos. How can he accomplish his task through the Internet?
14. Mampi is planning to open a company that deals with rural handicrafts. She wants to advertise about handicrafts on a social platform. Which Internet service she should use and why?
15. Ruhani wants to edit some privacy settings of her browser. How can she accomplish her task?
16. Shubham wants to play a video in his browser but he is not able to do so. A message on the screen instructs him to install the Adobe Flash Player plug-in. Help him to add it in his browser.
17. When Joe typed a URL in the address bar of his browser, Error 404 was displayed? Why did this happen? What can be done to avoid it?

Chapter 6 Societal Impacts

> "I think computer viruses should count as life. I think it says something about human nature that the only form of life we have created so far is purely destructive. We've created life in our own image."
>
> — Stephen Hawking

12149CH06

In this chapter

6.1 Introduction

In recent years, the world around us has seen a lot of changes due to use of 'Digital Technologies'. These changes have made a dramatic impact on our lives, making things more convenient, faster, and easier to handle. In the past, a letter would take days to reach, and every recipient would get his or her own copy and respond separately. Today, one can send and receive emails to more than one person at a time. The instantaneous nature of electronic communications has made us more efficient and productive.

From the banking industry to aviation, industrial production to e-commerce, especially with regard to the delivery of their goods and services, all are now dependent on the use of computers and digital technologies. Applications of digital technologies have redefined and evolved all spheres of human activities. Today more and more people are using digital technologies through smartphones, computers, etc., with the help of high speed Internet.

Why did the digital technologies become so widespread? The introduction of personal computers (PCs) and Internet followed by smartphones has brought these technologies to the common man.

While we reap the benefits of digital technologies, these technologies can also be misused. Let's look at the impact of these technologies on our society and the best practices that can ensure a productive and safe digital environment for us.

6.2 Digital Footprints

Have you ever searched online for any information? Have you ever purchased an online ticket, or responded to your friend's email, or checked the score of a game online? Whenever we surf the Internet using smartphones, tablets, computers, etc., we leave a trail of data reflecting the activities performed by us online, which is our *digital footprint*.

Figure 6.1: Exemplar web applications that result in digital footprints

Our digital footprint can be created and used with or without our knowledge. It includes websites we visit, emails we send, and any information we submit online, etc., along with the computer's IP address, location, and other device specific details. Such data could be used for targeted advertisement or could also be misused or exploited. Thus, it is good to be aware of the data trail we might be leaving behind. This awareness should make us cautious about what we write, upload or download or even browse online.

There are two kinds of digital footprints we leave behind. Active digital footprints which includes data that we intentionally submit online. This would include emails we write, or responses or posts we make on different websites or mobile Apps, etc. The digital data trail we leave online unintentionally is called passive digital footprints. This includes the data generated when we visit a website, use a mobile App, browse Internet, etc. as shown in Figure 6.1

Everyone who is connected to the Internet may have a digital footprint. With more usage, the trail grows. On examining the browser settings, we can find out how it stores our browsing history, cookies, passwords, auto fills, and many other types of data.

Besides browser, most of our digital footprints are stored in servers where the applications are hosted. We may not have access to remove or erase that data, neither do we have any control on how that data will be used. Therefore, once a data trail is generated, even if we later try to erase data about our online activities, the digital footprints still remain. There is no guarantee that digital footprints will be fully eliminated from the Internet. Therefore, we need to be more cautious while being online! All our online activities leave a data trace on the Internet as well as on the computing device that we use. This can be used to trace the user, their location, device and other usage details.

Think and Reflect

Can your digital footprints be used to judge your attitude and work ethics?

6.3 Digital Society and Netizen

As our society is inclined towards using more and more digital technologies, we end up managing most of our tasks digitally. In this era of digital society, our daily activities like communication, social networking, banking, shopping, entertainment, education, transportation, etc., are increasingly being driven by online transactions.

Digital society thus reflects the growing trend of using digital technologies in all spheres of human activities. But while online, all of us need to be aware of how to conduct ourselves, how best to relate with others and what ethics, morals and values to maintain. Anyone who uses digital technology along with Internet is a digital citizen or a netizen. Being a good netizen means practicing safe, ethical and legal use of digital technology. A responsible netizen must abide by net etiquettes, communication etiquettes and social media etiquettes.

Activity 6.1

As a digital citizen, list various services that you avail online.

6.3.1 Net Etiquettes

We follow certain etiquettes during our social interactions. Similarly, we need to exhibit proper manners and etiquettes while being online as shown in Figure 6.2. One should be ethical, respectful and responsible while surfing the Internet.

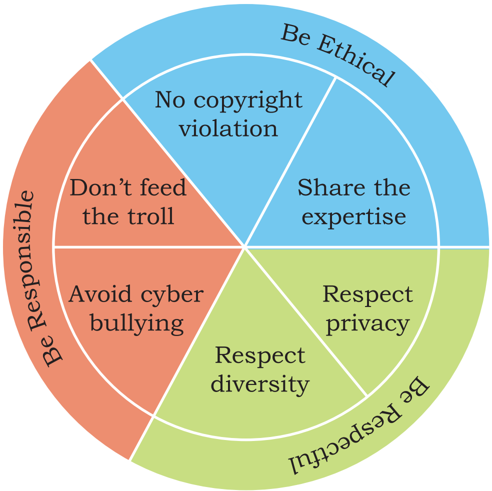


Figure 6.2: Net etiquettes

(A) *Be Ethical*

- No copyright violation: we should not use copyrighted materials without the permission of the creator or owner. As an ethical digital citizen, we need to be careful while streaming audio or video or downloading images and files from the Internet. We will learn more about copyright in Section 6.4.
- Share the expertise: it is good to share information and knowledge on Internet so that others can access it. However, prior to sharing information, we need to be sure that we have sufficient knowledge on that topic. The information shared should be true and unambiguous. Also, in order to avoid redundant information, we should verify that the information is not available already on Internet.

(B) *Be Respectful*

- Respect privacy: as good digital citizens we have the right to privacy and the freedom of personal expression. At the same time, we have to understand that other digital citizens also have the same rights and freedoms. Our personal communication with a digital citizen may include images, documents, files, etc., that are private to both. We should respect this privacy and should not share those images, documents, files, etc., with any other digital citizen without each others' consent.
- Respect diversity: in a group or public forum, we should respect the diversity of the people in terms of knowledge, experience, culture and other aspects.

While surfing the Internet, we should be cautious about our personal and confidential data.

√ Think before sharing credentials with others on an online platform.

√ Keep personal information safe and protected through passwords.

(C) *Be Responsible*

- Avoid cyber bullying: any insulting, degrading or intimidating online behaviour like repeated posting of rumours, giving threats online, posting the victim's personal information, sexual harassment or comments aimed to publicly ridicule a victim is termed as cyber bullying. It implies repeatedly targeting someone with intentions to hurt or embarrass. Perhaps new or non-frequent users of the Internet feel that things done online have no effect in the real world. We need to realise that bullying online can have very serious implications on the other person (victim). Also, remember our actions can be traced back using our digital footprints.

- Don't feed the troll: an Internet troll is a person who deliberately sows discord on the Internet by starting quarrels or upsetting people, by posting inflammatory or off topic messages in an online community, just for amusement. Since trolls thrive on attention, the best way to discourage trolls is not to pay any attention to their comments.

Activity 6.2

Find out how to report about an abusive or inappropriate post or about a sender in a social network.

6.3.2 Communication Etiquettes

Digital communication includes email, texting, instant messaging, talking on the cell phone, audio or video conferencing, posting on forums, social networking sites, etc. All these are great ways to connect with people in order to exchange ideas, share data and knowledge. Good communication over email, chat room and other such forums require a digital citizen to abide by the communication etiquettes as shown in Figure 6.3.

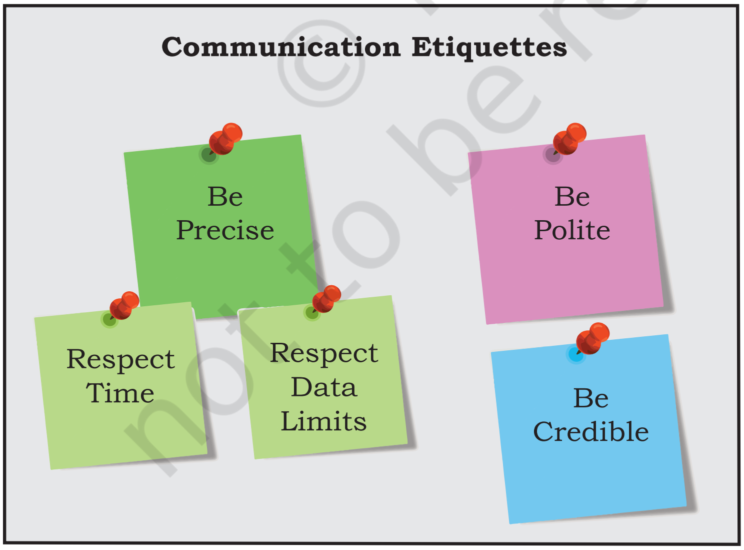


Figure 6.3: Communication etiquettes

(A) Be Precise

- Respect time: we should not waste precious time in responding to unnecessary emails or comments

Avoid Spam!!
On receiving junk email (called Spam), neither reply nor open any attachment in such email.

No Permanent Deletion!!

We can post or comment anything on Internet, and delete it later.

√ But remember, it cannot be permanently deleted. It is recorded in our Digital Footprint.

√ This is how many culprits who spread hate, bully others or engage in criminal activities are traced and apprehended.

unless they have some relevance for us. Also, we should not always expect an instant response as the recipient may have other priorities.

- Respect data limits: For concerns related to data and bandwidth, very large attachments may be avoided. Rather send compressed files or link of the files through cloud shared storage like Google Drive, Microsoft OneDrive, Yahoo Dropbox, etc.

(B) Be Polite

Whether the communication is synchronous (happening in real time like chat, audio/video calls) or asynchronous (like email, forum post or comments), we should be polite and non-aggressive in our communication. We should avoid being abusive even if we don't agree with others' point of view.

(C) Be Credible

We should be cautious while making a comment, replying or writing an email or forum post as such acts decide our credibility over a period of time. That is how we decide to follow some particular person's forum posts while ignoring posts of other members of the forum. On various discussion forums, we usually try to go through the previous comments of a person and judge their credibility before relying on that person's comments.

6.3.3 Social Media Etiquettes

In the current digital era, we are familiar with different kinds *social media* and we may have an account on Facebook, Google+, Twitter, Instagram, Pinterest, or YouTube channel. Social media are websites or applications that enable their users to participate in social networking by creating and sharing content with others in the community. These platforms encourage users to share their thoughts and experiences through posts or pictures. In this way users can interact with other online users of those social media apps or channels. This is why the impact and outreach of social media has grown exponentially. It has begun to shape the outcome of politics, business, culture, education and more. In social media too, there are certain etiquettes we need to follow as shown in Figure 6.4.

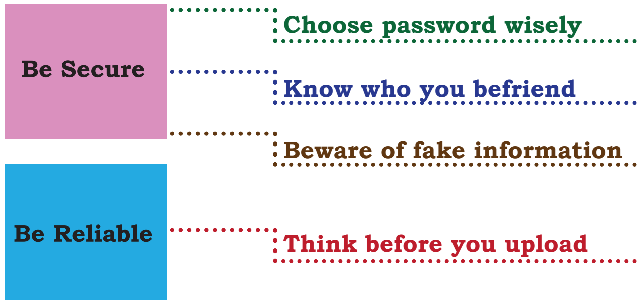


Figure 6.4: Social media etiquettes

(A) Be Secure

- Choose password wisely: it is vital for social network users. News of breaching or leakage of user data from social network often attracts headlines. Users should be wary of such possibilities and must know how to safeguard themselves and their accounts. The minimum one can do is to have strong and frequently changed password. Never share personal credentials like username and password with others.
- Know who you befriend: social networks usually encourage connecting with users (making friends), sometime even those whom we don't know or have not met. However, we need to be careful while befriending unknown people as their intentions possibly could be malicious and unsafe.
- Beware of fake information: fake news, messages and posts are common in social networks. As a user, we should be aware of them. With experience, we should be able to figure out whether a news, message or post is genuine or fake. Thus, we should not blindly believe in everything that we come across on such platforms, we should apply our knowledge and experience to validate such news, message or post.

(B) Be Reliable

- Think before uploading: we can upload almost anything on social network. However, remember that once uploaded, it is always there in the remote server even if we delete the files. Hence we need to be cautious while uploading or sending sensitive or confidential files which have a bearing on our privacy.

Don't Meet Up!!

√ Never arrange to meet an online friend because it may not be safe.

√ No matter how genuine someone is appearing online,
they might be pretending and hiding their real identity.

Think and Reflect

Is having the same password for all your accounts on different websites safe?

Play Safe!!

Think carefully before sharing personal photos.

6.4 Data Protection

In this digital age, data or information protection is mainly about the privacy of data stored digitally. Elements of data that can cause substantial harm, embarrassment, inconvenience and unfairness to an individual, if breached or compromised, is called sensitive data. Examples of sensitive data include biometric information, health information, financial information, or other personal documents, images or audios or videos. Privacy of sensitive data can be implemented by encryption, authentication, and other secure methods to ensure that such data is accessible only to the authorised user and is for a legitimate purpose.

Activity 6.3

Suppose someone's email password is 'technology', which is weak. Can you suggest a stronger password?

All over the world, each country has its own data protection policies (laws). These policies are legal documents that provide guidelines to the user on processing, storage and transmission of sensitive information. The motive behind implementation of these policies is to ensure that sensitive information is appropriately protected from modification or disclosure.

Think and Reflect

Why should we always mention the source from which we got an idea or used resources (text, image, audio, video, etc.) to prepare a project or a writeup?

6.4.1 Intellectual Property Right (IPR)

When someone owns a house or a motorcycle, we say that the person owns that property. Similarly, if someone comes out with a new idea, this original idea is that person’s intellectual property. Intellectual Property refers to the inventions, literary and artistic expressions, designs and symbols, names and logos. The ownership of such concepts lies with the creator, or the holder of the intellectual property. This enables the creator or copyright owner to earn recognition or financial benefit by using their creation or invention. Intellectual Property is legally protected through copyrights, patents, trademarks,etc.

Executing IPR: say for a software

- √ Code of the software will be protected by a copyright
- √ Functional expression of the idea will be protected by a patent
- √ The name and logo of the software will come under a registered trademark

(A) Copyright

Copyright grants legal rights to creators for their original works like writing, photograph, audio recordings, video, sculptures, architectural works, computer software, and other creative works like literary and artistic work. Copyrights are automatically granted to creators and authors. Copyright law gives the copyright holder a set of rights that they alone can avail legally. The rights include right to copy (reproduce) a work, right to create derivative works based upon it, right to distribute copies of the work to the public, and right to publicly display or perform the work. It prevents others from copying, using or selling the work. For example, writer Rudyard Kipling holds the copyright to his novel, 'The Jungle Book', which tells the story of Mowgli, the jungle boy. It would be an infringement of the writer's copyright if someone used parts of the novel without permission. To use other's copyrighted material, one needs to obtain a license from them.

Activity 6.4

Explore the following websites to know about open/public licensing:

(i) creativecommons.org for CC, and

(ii) gnu.org for GNU GPL.

(B) Patent

A patent is usually granted for inventions. Unlike copyright, the inventor needs to apply (file) for patenting the invention. When a patent is granted, the owner gets an exclusive right to prevent others from using, selling, or distributing the protected invention. Patent gives full control to the patentee to decide whether or how the invention can be used by others. Thus it encourages inventors to share their scientific or technological findings with others. A patent protects an invention for 20 years, after which it can be freely used. Recognition and/or financial benefit foster the right environment, and provide motivation for more creativity and innovation.

Only the copyright owner of a work can enter into a license agreement.

(C) Trademark

Trademark includes any visual symbol, word, name, design, slogan, label, etc., that distinguishes the brand or commercial enterprise, from other brands or commercial enterprises. For example, no company other than Nike can use the Nike brand to sell shoes or clothes. It also prevents others from using a confusingly similar mark, including words or phrases. For example, confusing brands like "Nikke" cannot be used. However, it may be possible to apply for the Nike trademark for unrelated goods like notebooks.

6.4.2 Licensing

We have studied about copyright in the previous section. Licensing and copyrights are two sides of the same coin. A license is a type of contract or a permission agreement between the creator of an original work permitting someone to use their work, generally for some price; whereas copyright is the legal rights of the creator for the protection of original work of different types. Licensing

End User License Agreement (EULA) contains the dos and don'ts with respect to the software being purchased. It covers all clauses of software purchase, viz., how many copies can be installed, whether source is available, whether it can be modified and redistributed and so on.

is the legal term used to describe the terms under which people are allowed to use the copyrighted material. We will limit our study to software licensing in this chapter.

A software license is an agreement that provides legally binding guidelines pertaining to the authorised use of digital material. The digital material may include any software or any form of art, literature, photos, etc., in digital form. Any such resource posted on the Internet constitutes intellectual property and must be downloaded, used or distributed according to the guidelines given in the license agreement. Failure to follow such guidelines is considered as an infringement of Intellectual Property Rights (IPR), and is a criminal offence.

Beware!!

√ Plagiarism means using other's work and not giving adequate citation for use.

√ Copyright infringement means using another person's work, without permission or without paying for it, if it is being sold.

6.4.3 Violation of IPR

Violation of intellectual property right may happen in one of the following ways:

(A) Plagiarism

With the availability of Internet, we can instantly copy or share text, pictures and videos. Presenting someone else's idea or work as one's own idea or work is called plagiarism. If we copy some contents from Internet, but do not mention the source or the original creator, then it is considered as an act of plagiarism. Further, if someone derives an idea or a product from an already existing idea or product, but instead presents it as a new idea, then also it is plagiarism. It is a serious ethical offense and sometimes considered as an act of fraud. Even if we take contents that are open for public use, we should cite the author or source to avoid plagiarism.

(B) Copyright Infringement

Copyright infringement is when we use other person's work without obtaining their permission to use or we have not paid for it, if it is being sold. Suppose we download an image from the Internet and use it in our project. But if the owner of the copyright of the image does not permit its free usage, then using such an image even after giving reference of the image in our project is a violation of copyright. Just because it is on the Internet, does not mean that it is free for use. Hence, check the copyright status of writer's work before using it to avoid copyright infringement.

(C) Trademark Infringement

Trademark Infringement means unauthorised use of other's trademark on products and services. An owner of a trademark may commence legal proceedings against someone who infringes its registered trademark.

6.4.4 Public Access and Open Source Software

Copyright sometimes put restriction on the usage of the copyrighted works by anyone else. If others are allowed to use and built upon the existing work, it will encourage collaboration and would result in new innovations in the same direction. Licenses provide rules and guidelines for others to use the existing work. When authors share their copyrighted works with others under public license, it allows others to use and even modify the content. Open source licenses help others to contribute to existing work or project without seeking special individual permission to do so.

The GNU General Public License (GPL) and the Creative Commons (CC) are two popular categories of public licenses. CC is used for all kind of creative works like websites, music, film, literature, etc. CC enables the free distribution of an otherwise copyrighted work. It is used when an author wants to give people the right to share, use and build upon a work that they have created. GPL is primarily designed for providing public licence to a software. GNU GPL is another free software license, which provides end users the freedom to run, study, share and modify the software, besides getting regular updates.

Users or companies who distribute GPL licensed works may charge a fee for copies or give them free of charge. This distinguishes the GPL license from freeware software licenses like Skype, Adobe Acrobat reader, etc. that allow copying for personal use but prohibit commercial distribution, or proprietary licenses where copying is prohibited by copyright law.

Many of the proprietary software that we use are sold commercially and their program code (source code) are not shared or distributed. However, there are certain software available freely for anyone and their source code is also open for anyone to access, modify, correct and improve. Free and open source software (FOSS) has a large community of users and developers who are

Remember

√ CC licenses are a set of copyright licenses that give the recipients, rights to copy, modify and redistribute the creative material, but giving the authors, the liberty to decide the conditions of licensing.

√ GPL is the most widely used free software license which grants the recipients, rights to copy, modify and redistribute the software and that the same rights are preserved in all derivative works.

NOTES

contributing continuously towards adding new features or improving the existing features. For example, Linux kernel-based operating systems like Ubuntu and Fedora come under FOSS. Some of the popular FOSS tools are office packages, like Libre Office, browser like Mozilla Firefox, etc.

Software piracy is the unauthorised use or distribution of software. Those who purchase a license for a copy of the software do not have the rights to make additional copies without the permission of the copyright owner. It amounts to copyright infringement regardless of whether it is done for sale, for free distribution or for copier's own use. One should avoid software piracy. Using a pirated software not only degrades the performance of a computer system, but also affects the software industry which in turn affects the economy of a country.

6.5 CREATIVE COMMONS

Creative Commons is a non-profit organisation (https://creativecommons.org/) that aims to build a publically accessible global platform where a range of creative and academic works are shared freely. Any one across the globe can access them, share them, and even use them for creating their own work out of it without infringing the copyright or Intellectual Property rights of the owners. In fact, it gives proper attribution to the owners.

The Creative Commons organisation provides Creative Commons (CC) licenses free of charge. It allows owners of a work to grant copyright permissions for their creative and/or academic works in a free, simple and standardised way. A CC license is a type of copyright license that enables the free distribution of anybody's copyrighted work. This license is used when an author wants to give others the right to share, use and extend the work done by them. The work licensed under CC is governed by the Copyright law and so applies to all types of work including art, music, literature, dramatics, movies, images, educational resources, photographs and software. The CC Search feature of the online platform makes the licensed material easier to find. The author of the content is given full freedom to set up conditions to use their work. The owner of a work can combine these conditions to create six different types of CC licenses, as listed in Table 6.1.

Table 6.1 Creative Commons (CC) Licenses

License Name	Symbolic name	License icon	Description
Attribution	CC BY	cc BY	This license lets others distribute, remix, tweak, and build upon your work, even commercially, as long as they credit you for the original creation.
Attribution-ShareAlike	CC BY-SA	cc BY SA	This license lets others remix, tweak, and build upon your work even for commercial purposes, as long as they credit you and license their new creations under the identical terms.
Attribution-NoDerivs	CC BY-ND	cc BY ND	This license lets others reuse the work for any purpose, including commercially; however, it cannot be shared with others in adapted form, and credit must be provided to you.
Attribution-NonCommercial	CC BY-NC	cc BY NC	This license lets others remix, tweak, and build upon your work non-commercially, and although their new works must also acknowledge you and be non-commercial.
Attribution-NonCommercial-ShareAlike	CC BY-NC-SA	cc BY NC SA	This license lets others remix, tweak, and build upon your work non-commercially, as long as they credit you and license their new creations under the identical terms.
Attribution-NonCommercial-NoDerivs	CC BY-NC-ND	cc BY NC ND	This license is the most restrictive of our six main licenses, only allowing others to download your works and share them with others as long as they credit you, but they can't change them in any way or use them commercially.

6.6 Cyber Crime

Criminal activities or offences carried out in a digital environment can be considered as cyber crime. In such crimes, either the computer itself is the target or the computer is used as a tool to commit a crime. Cyber crimes are carried out against either an individual, or a group, or an organisation or even against a country, with the intent to directly or indirectly cause physical harm, financial loss or mental harassment. A cyber criminal attacks a computer or a network to reach other computers in order to disable or damage data or services. Apart from this, a cyber criminal may spread viruses and other malwares in order to steal private and confidential data for blackmailing and extortion. A computer virus is some lines of malicious code that can copy itself and can have detrimental effect on the computers, by destroying data or corrupting the system. Similarly, malware is

Remember!!

Cyber crime is defined as a crime in which computer is the medium of crime (hacking, phishing, spamming), or the computer is used as a tool to commit crimes (extortion, data breaches, theft).

a software designed to specifically gain unauthorised access to computer systems. The nature of criminal activities are alarmingly increasing day-by-day, with frequent reports of hacking, ransomware attacks, denial-of-service, phishing, email fraud, banking fraud and identity theft.

Activity 6.5

How can you unsubscribe from a mail group or block an email sender?

6.6.1 Hacking

Hacking is the act of unauthorised access to a computer, computer network or any digital system. Hackers usually have technical expertise of the hardware and software. They look for bugs to exploit and break into the system.

Hacking, when done with a positive intent, is called ethical hacking. Such ethical hackers are known as white hat hackers. They are specialists in exploring any vulnerability or loophole by during testing of the software. Thus, they help in improving the security of a software. An ethical hacker may exploit a website in order to discover its security loopholes or vulnerabilities. He then reports his findings to the website owner. Thus, ethical hacking is actually preparing the owner against any cyber attack.

A non-ethical hacker is the one who tries to gain unauthorised access to computers or networks in order to steal sensitive data with the intent to damage or bring down systems. They are called black hat hackers or crackers. Their primary focus is on security cracking and data stealing. They use their skill for illegal or malicious purposes. Such hackers try to break through system securities for identity theft, monetary gain, to bring a competitor or rival site down, to leak sensitive information, etc.

Beware !!

Accepting links from untrusted emails can be hazardous, as they may potentially contain a virus or link to malicious website. We should ensure to open any email link or attachment only when it is from a trusted source and doesn't look doubtful.

6.6.2 Phishing and Fraud Emails

Phishing is an unlawful activity where fake websites or emails that look original or authentic are presented to the user to fraudulently collect sensitive and personal details, particularly usernames, passwords, banking and credit card details. The most common phishing method is through email spoofing where a fake or forged email address is used and the user presumes it to be from an authentic source. So you might get an email from an address that looks similar to your bank or educational institution, asking for your information, but if you look carefully you will see their URL address is fake. They will often use logo's of the original, making them difficult to detect from the real! Phishing attempts through phone calls or text messages are also common these days.

(A) Identity Theft

Identity thieves increasingly use personal information stolen from computers or computer networks, to commit fraud by using the data gained unlawfully. A user's identifiable personal data like demographic details, email ID, banking credentials, passport, PAN, Aadhaar number and various such personal data are stolen and misused by the hacker on behalf of the victim. This is one type of phishing attack where the intention is largely for monetary gain. There can be many ways in which the criminal takes advantage of an individual's stolen identity. Given below are a few examples:

- Financial identity theft: when the stolen identity is used for financial gain.
- Criminal identity theft: criminals use a victim's stolen identity to avoid detection of their true identity.
- Medical identity theft: criminals can seek medical drugs or treatment using a stolen identity.

Activity 6.6

Explore and find out how to file a complaint with the cyber cell in your area.

6.6.3 Ransomware

This is another kind of cyber crime where the attacker gains access to the computer and blocks the user from accessing, usually by encrypting the data. The attacker blackmails the victim to pay for getting access to the data, or sometimes threatens to publish personal and sensitive information or photographs unless a ransom is paid.

Ransomware can get downloaded when the users visit any malicious or unsecure websites or download software from doubtful repositories. Some ransomware are sent as email attachments in spam mails. It can also reach our system when we click on a malicious advertisement on the Internet.

6.6.4 Combatting and Preventing Cyber Crime

The challenges of cyber crime can be mitigated with the twin approach of being alert and taking legal help.

Digital signatures are the digital equivalent of a paper certificate. Digital signatures work on a unique digital ID issued by an Certificate Authority (CA) to the user. Signing a document digitally means attaching that user's identify, which can be used to authenticate.

A licensed Certifying Authority (CA) who has been granted a license to issue it under Section 24 of the Indian IT-Act 2000, can issue the digital signature.

Following points can be considered as safety measures to reduce the risk of cyber crime:

- Take regular backup of important data.
- Use an antivirus software and keep it updated always.
- Avoid installing pirated software. Always download software from known and secure (HTTPS) sites.
- Always update the system software which include the Internet browser and other application software
- Do not visit or download anything from untrusted websites.
- Usually the browser alerts users about doubtful websites whose security certificate could not be verified; avoid visiting such sites.
- Use strong password for web login, and change it periodically. Do not use same password for all the websites. Use different combinations of alphanumeric characters including special characters. Ignore common words or names in password.
- While using someone else's computer, don't allow browser to save password or auto fill data, and try to browse in your private browser window.
- For an unknown site, do not agree to use cookies when asked for through a Yes/No option.
- Perform online transaction like shopping, ticketing, and other such services only through well-known and secure sites.
- Always secure wireless network at home with strong password and regularly change it.

6.7 INDIAN INFORMATION TECHNOLOGY ACT (IT ACT)

With the growth of Internet, many cases of cyber crimes, frauds, cyber attacks and cyber bullying are reported. The nature of fraudulent activities and crimes keeps changing. To deal with such menaces, many countries have come up with legal measures for protection of sensitive personal data and to safeguard the rights of Internet users. The Government of India's The Information Technology Act, 2000 (also known as IT Act), amended in 2008, provides guidelines to the user on the processing, storage and transmission of sensitive information. In many Indian states, there are cyber cells in police stations where one can report any cyber crime. The act provides legal framework for electronic governance by giving recognition to electronic records and digital signatures. The act outlines cyber crimes and penalties for them.

Cyber Appellate Tribunal has been established to resolve disputes arising from cyber crime, such as tampering with computer source documents, hacking the computer system, using password of another person, publishing sensitive personal data of others without their consent, etc. The act is needed so that people can perform transactions over the Internet through credit cards without fear of misuse. Not only people, the act empowers government departments also to accept filing, creation and storage of official documents in the digital format.

California Law University has identified non-functioning cathode ray tubes (CRTs) from televisions and computer monitors as hazardous.

6.8 E-waste: Hazards and Management

E-waste or Electronic waste includes electric or electronic gadgets and devices that are no longer in use. Hence, discarded computers, laptops, mobile phones, televisions, tablets, music systems, speakers, printers, scanners etc. constitute e-waste when they are near or end of their useful life.

E-waste is becoming one of the fastest growing environmental hazards in the world today. The increased use of electronic equipment has also caused an exponential increase in the number of discarded products. Lack of awareness and appropriate skill to manage it has further worsened the problem. So, Waste Electrical and Electronic Equipment (WEEE) is becoming a major concern for all countries across the world. Globally, e-waste constitutes more than 5 per cent of the municipal solid waste. Therefore, it is very important that e-waste is disposed of in such a manner that it causes minimum damage to the environment and society.

Leaching is the process of removing a substance from another substance by passing water through it.

6.8.1 Impact of e-waste on environment

To some extent, e-waste is responsible for the degradation of our environment. Whether it is emission of gases and fumes into the atmosphere, discharge of liquid waste into drains or disposal of solid e-waste materials, all of this contributes to environmental pollution in some way or the other.

When e-waste is carelessly thrown or dumped in landfills or dumping grounds, certain elements or metals used in production of electronic products cause air, water and soil pollution. This is because when these products come in contact with air and moisture, they tend to leach. As a result, the harmful chemicals seep into the soil, causing soil pollution. Further, when these chemicals reach and contaminate the natural ground water, it causes water pollution as the water becomes unfit for humans, animals and even for agricultural use. When dust particles loaded with heavy metals enters the atmosphere, it causes air pollution as well.

6.8.2 Impact of e-waste on humans

As mentioned before, the electrical or electronic devices are manufactured using certain metals and elements like lead, beryllium, cadmium, plastics, etc. Most of these materials are difficult to recycle and are considered to be toxic and carcinogenic. If e-waste is not disposed of in proper manner, it can be extremely harmful to humans, plants, animals and the environment as discussed below:

Carcinogenic: May cause cancer

- One of the most widely used metals in electronic devices (such as monitors and batteries) is lead. When lead enters the human body through contaminated food, water, air or soil, it causes lead poisoning which affects the kidneys, brain and central nervous system. Children are particularly vulnerable to lead poisoning.
- When e-waste such as electronic circuit boards are burnt for disposal, the elements contained in them create a harmful chemical called beryllium which causes skin diseases, allergies and an increased risk of lung cancer. Burning of insulated wires to extract copper can cause neurological disorders.
- Some of the electronic devices contain mercury which causes respiratory disorders and brain damage.
- The cadmium found in semiconductors and resistors can damage kidneys, liver and bones.
- None of the electronic devices are manufactured without using plastics. When this plastic reacts

with air and moisture, it passes harmful chemicals into the soil and water resources. When consumed, it damages the immune system of the body and also causes various psychological problems like stress and anxiety.

6.8.3 Management of e-waste

E-waste management is the efficient disposal of e-waste. Although we cannot completely destroy e-waste, still certain steps and measures have to be taken to reduce harm to the humans and environment. Some of the feasible methods of e-waste management are reduce, reuse and recycle.

- **Reduce:** We should try to reduce the generation of e-waste by purchasing the electronic or electrical devices only according to our need. Also, they should be used to their maximum capacity and discarded only after their useful life has ended. Good maintenance of electronics devices also increases the life of the devices.
- **Reuse:** It is the process of re-using the electronic or electric waste after slight modification. The electronic equipment that is still functioning should be donated or sold to someone who is still willing to use it. The process of re-selling old electronic goods at lower prices is called refurbishing.
- **Recycle:** Recycling is the process of conversion of electronic devices into something that can be used again and again in some or the other manner. Only those products should be recycled that cannot be repaired, refurbished or re-used. To promote recycling of e-waste many companies and NGOs are providing door-to-door pick up facilities for collecting the e-waste from homes and offices.

6.8.4 E-waste Management in India

In India, the Environmental Protection Act, 1986, has been enacted to punish people responsible for causing any form of pollution by paying for the damage done to the natural environment. According to this act, "Polluter pays Principle", any one causing any form of pollution will pay for the damage caused. Any violation of the provisions of this act is liable for punishment.

Think and Reflect

Do you follow precautions to stay healthy - physically, mentally as well as emotionally while using digital technologies?

The Central Pollution Control Board (CPCB) has issued a formal set of guidelines for proper handling and disposal of e-waste. According to these guidelines, the manufacturer of any electronic equipment will be "personally" responsible for the final safe disposal of the product when it becomes an e-waste.

The Department of Information Technology (DIT), Ministry of Communication and Information Technology, has also issued a comprehensive technical guide on "Environmental Management for Information Technology Industry in India." The industries have to follow these guidelines for recycling and reuse of e-waste. In order to make the consumers aware of the recycling of e-waste, prominent smartphone and computer manufacturing companies have started various recycling programs.

Device Safety: Ensures Good Health of a Computer System

√ Regularly clean it to keep the dust off. Use a liquid solution specifically formulated for the cleaning of electronic screens.

√ Wipe monitor's screen often using the regular microfibre soft cloth (the one used for spectacles).

√ Keep it away from direct heat, sunlight and put it in a room with enough ventilation for air circulation.

√ Do not eat food or drink over the keyboard. Food crumbs that fall into the gaps between the keys or spilled over liquid can cause issues to the devices.

6.9 Impact on Health

As digital technologies have penetrated into different fields, we are spending more time in front of screens, be it mobile, laptop, desktop, television, gaming console, music or sound device. But interacting in an improper posture can be bad for us — both physically, and mentally. Besides, spending too much time on Internet can be addictive and can have a negative impact on our physical and psychological well being.

However, these health concerns can be addressed to some extent by taking care of the way we position such devices and the way we position our posture. Ergonomics is a branch of science that deals with designing or arranging workplaces including the furniture, equipments and systems so that it becomes safe and comfortable for the user. Ergonomics helps us in reducing the strain on our bodies — including the fatigue and injuries due to prolonged use.

When we continuously look at the screen for watching, typing, chatting or playing games, our eyes are continuously exposed to the glare coming from the screens. Looking at small handheld devices makes it worse. Eye strain is a symptom commonly complained by users of digital devices. Ergonomically maintaining the viewing distance and angle, along with the position can be of some help. Figure 6.5 shows the posture to be maintained in order to avoid fatigue caused due to prolonged use of computer system and other digital devices. However, to get rid of dry, watering, or itchy eyes, it is better to periodically focus on distant objects, and take a break for outdoor activities.

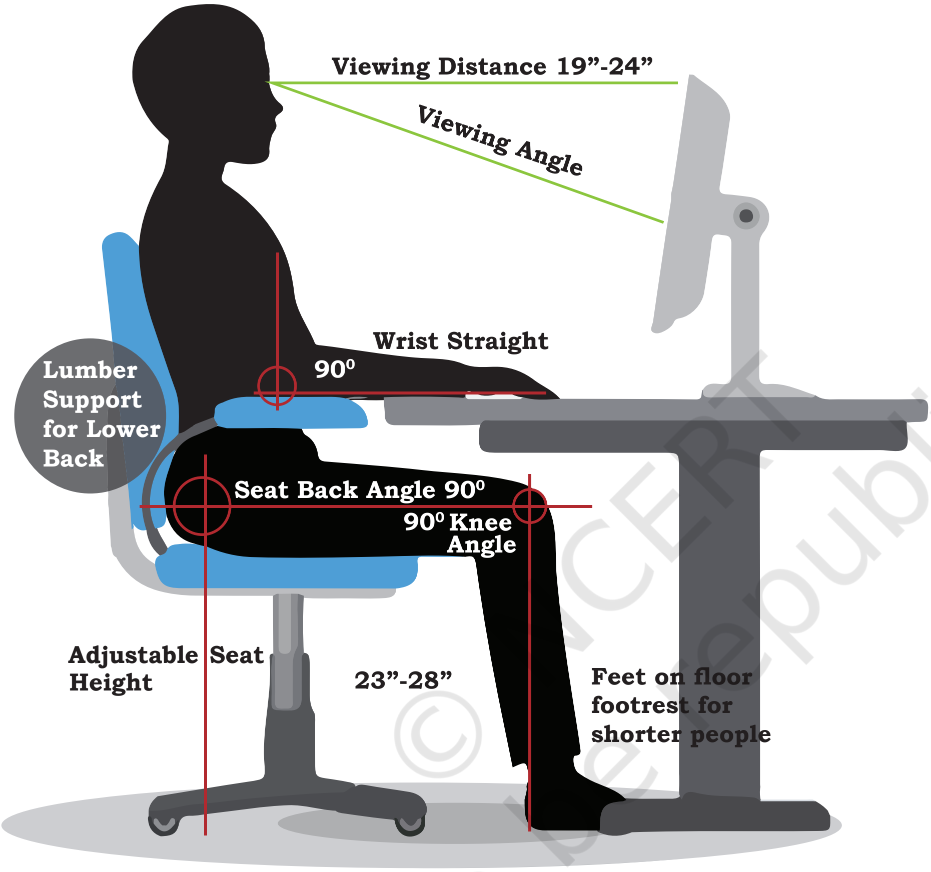


Figure 6.5: Correct posture while sitting in front of a computer

Maintain a Balance!!

Enjoy the exciting world of digital devices in tandem with other pursuits of thrilling sports and hobbies. Online friends are good, but spending time with friends in real life is very fulfilling. Often the wholesome nature of real interactions cannot be compared to just online social networking.

Bad posture, backaches, neck and shoulder pains can be prevented by arranging the workspace as recommended by ergonomics. Overuse of keyboards (be it physical keyboard or touchscreen-based virtual keyboard) not aligned ergonomically, can give rise to a painful condition of wrists and fingers, and may require medical help in the long run.

Stress, physical fatigue and obesity are the other related impacts the body may face if one spends too much time using digital devices.

NOTES

SUMMARY

- Digital footprint is the trail of data we leave behind when we visit any website (or use any online application or portal) to fill-in data or perform any transaction.
- A user of digital technology needs to follow certain etiquettes like net-etiquettes, communication-etiquettes and social media-etiquettes.
- Net-etiquette includes avoiding copyright violations, respecting privacy and diversity of users, and avoiding cyber bullies and cyber trolls, besides sharing of expertise.
- Communication-etiquette requires us to be precise and polite in our conversation so that we remain credible through our remarks and comments.
- While using social media, one needs to take care of security through password, be aware of fake information and be careful while befriending unknowns. Care must be taken while sharing anything on social media as it may create havoc if being mishandled, particularly our personal, sensitive information.
- Intellectual Property Rights (IPR) help in data protection through copyrights, patents and trademarks. There are both ethical and legal aspects of violating IPR. A good digital citizen should avoid plagiarism, copyright infringement and trademark infringement.
- Certain software are made available for free public access. Free and Open Source Software (FOSS) allow users to not only access but also to modify (or improve) them.
- Cyber crimes include various criminal activities carried out to steal data or to break down important services. These include hacking, spreading viruses or malware, sending phishing or fraudulent emails, ransomware, etc.
- Excessive usage of digital devices has a negative impact on our physical as well as psychological well-being. Ergonomic positioning of devices as well as our posture are important.

Exercise

Notes

1. After practicals, Atharv left the computer laboratory but forgot to sign off from his email account. Later, his classmate Revaan started using the same computer. He is now logged in as Atharv. He sends inflammatory email messages to few of his classmates using Atharv's email account. Revaan's activity is an example of which of the following cyber crime? Justify your answer.
   a) Hacking
   b) Identity theft
   c) Cyber bullying
   d) Plagiarism
2. Rishika found a crumpled paper under her desk. She picked it up and opened it. It contained some text which was struck off thrice. But she could still figure out easily that the struck off text was the email ID and password of Garvit, her classmate. What is ethically correct for Rishika to do?
   a) Inform Garvit so that he may change his password.
   b) Give the password of Garvit's email ID to all other classmates.
   c) Use Garvit's password to access his account.
3. Suhana is down with fever. So, she decided not to go to school tomorrow. Next day, in the evening she called up her classmate, Shaurya and enquired about the computer class. She also requested him to explain the concept. Shaurya said, "Mam taught us how to use tuples in python". Further, he generously said, "Give me some time, I will email you the material which will help you to understand tuples in python". Shaurya quickly downloaded a 2-minute clip from the Internet explaining the concept of tuples in python. Using video editor, he added the text "Prepared by Shaurya" in the downloaded video clip. Then, he emailed the modified video clip to Suhana. This act of Shaurya is an example of —
   a) Fair use
   b) Hacking
   c) Copyright infringement
   d) Cyber crime
4. After a fight with your friend, you did the following activities. Which of these activities is not an example of cyber bullying?

NOTES

a) You sent an email to your friend with a message saying that "I am sorry".

b) You sent a threatening message to your friend saying "Do not try to call or talk to me".

c) You created an embarrassing picture of your friend and uploaded on your account on a social networking site.

5. Sourabh has to prepare a project on "Digital India Initiatives". He decides to get information from the Internet. He downloads three web pages (webpage 1, webpage 2, webpage 3) containing information on Digital India Initiatives. Which of the following steps taken by Sourabh is an example of plagiarism or copyright infringement? Give justification in support of your answer.

   a) He read a paragraph on " Digital India Initiatives" from webpage 1 and rephrased it in his own words. He finally pasted the rephrased paragraph in his project.

   b) He downloaded three images of " Digital India Initiatives" from webpage 2. He made a collage for his project using these images.

   c) He downloaded "Digital India Initiative" icon from web page 3 and pasted it on the front page of his project report.

6. Match the following:

Column A	Column B
Plagiarism	Fakers, by offering special rewards or money prize asked for personal information, such as bank account information
Hacking	Copy and paste information from the Internet into your report and then organise it
Credit card fraud	The trail that is created when a person uses the Internet.
Digital Foot Print	Breaking into computers to read private emails and other files

7. You got the below shown SMS from your bank querying a recent transaction. Answer the following —

   a) Will you SMS your pin number to the given contact number?

   b) Will you call the bank helpline number to recheck the validity of the SMS received?

8. Preeti celebrated her birthday with her family. She was excited to share the moments with her friend Himanshu. She uploaded selected images of her

birthday party on a social networking site so that Himanshu can see them. After few days, Preeti had a fight with Himanshu. Next morning, she deleted her birthday photographs from that social networking site, so that Himanshu cannot access them. Later in the evening, to her surprise, she saw that one of the images which she had already deleted from the social networking site was available with their common friend Gayatri. She hurriedly enquired Gayatri "Where did you get this picture from?". Gayatri replied "Himanshu forwarded this image few minutes back".

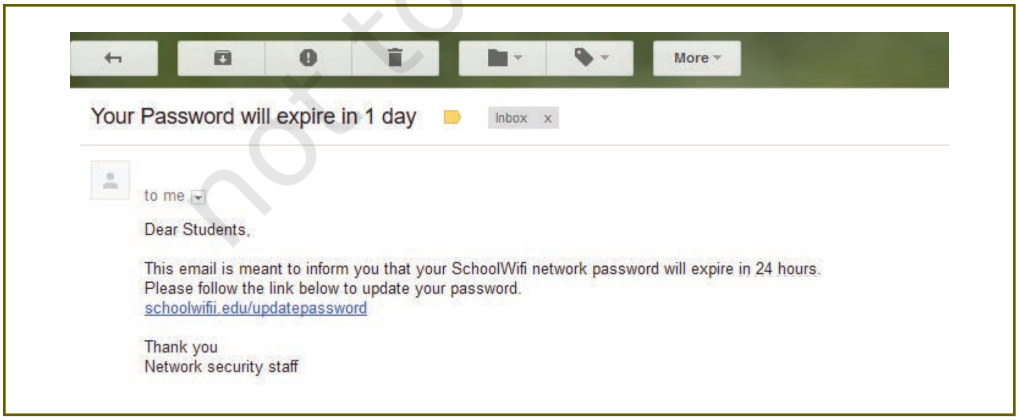
Help Preeti to get answers for the following questions. Give justification for your answers so that Preeti can understand it clearly.

a) How could Himanshu access an image which I had already deleted?

b) Can anybody else also access these deleted images?

c) Had these images not been deleted from my digital footprint?

9. The school offers wireless facility (wifi) to the Computer Science students of Class XI. For communication, the network security staff of the school have a registered URL schoolwifi.edu. On 17 September 2017, the following email was mass distributed to all the Computer Science students of Class XI. The email claimed that the password of the students was about to expire. Instructions were given to go to URL to renew their password within 24 hours.



Your Password will expire in 1 day

to me

Dear Students,

This email is meant to inform you that your SchoolWifi network password will expire in 24 hours.
Please follow the link below to update your password.
schoolwifii.edu/updatepassword

Thank you
Network security staff

a) Do you find any discrepancy in this email?

b) What will happen if the student will click on the given URL?

NOTES

c) Is the email an example of cyber crime? If yes, then specify which type of cyber crime is it. Justify your answer.

10. You are planning to go for a vacation. You surfed the Internet to get answers for the following queries —

    a) Weather conditions
    b) Availability of air tickets and fares
    c) Places to visit
    d) Best hotel deals

    Which of your above mentioned actions might have created a digital footprint?

11. How would you recognise if one of your friends is being cyber bullied?

    a) Cite the online activities which would help you detect that your friend is being cyber bullied?
    b) What provisions are in IT Act 2000, (amended in 2008) to combact such situations.

12. Write the differences between the following —

    a) Copyrights and Patents
    b) Plagiarism and Copyright infringement
    c) Non-ethical hacking and Ethical hacking
    d) Active and Passive footprints
    e) Free software and Free and open source software

13. If you plan to use a short text from an article on the web, what steps must you take in order to credit the sources used?

14. When you search online for pictures, how will you find pictures that are available in the free public domain. How can those pictures be used in your project without copyright violations?

15. Describe why it is important to secure your wireless router at home. Search the Internet to find the rules to create a reasonably secure password. Create an imaginary password for your home router. Will you share your password for home router with following people. Justify your answer.

    a) Parents
    b) Friends
    c) Neighbours
    d) Home tutors

16. List down the steps you need to take in order to ensure —

a) your computer is in good working condition for a longer time.

b) smart and safe Internet surfing.

17. What is data privacy? Websites that you visit collect what type of information about you?

18. In the computer science class, Sunil and Jagdish were assigned the following task by their teacher.

    a) Sunil was asked to find information about "India, a Nuclear power". He was asked to use Google Chrome browser and prepare his report using Google Docs.

    b) Jagdish was asked to find information about "Digital India". He was asked to use Mozilla Firefox browser and prepare his report using Libre Office Writer.

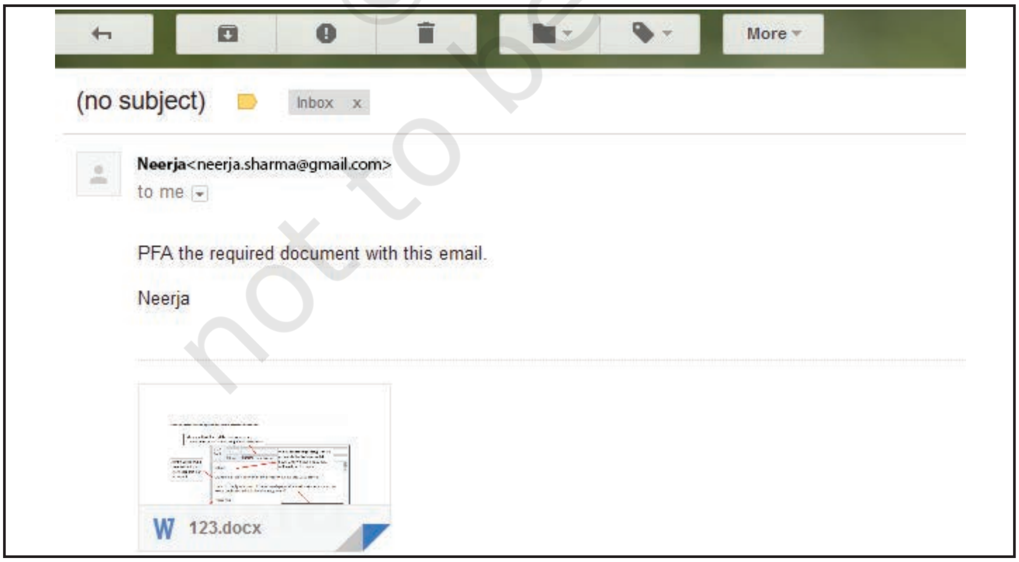
    What is the difference between technologies used by Sunil and Jagdish?

19. Cite examples depicting that you were a victim of following cyber crime. Also, cite provisions in IT Act to deal with such a cyber crime.

    a) Identity theft

    b) Credit card account theft

20. Neerja is a student of Class XI. She has opted for Computer Science. Neerja prepared the project assigned to her. She mailed it to her teacher. The snapshot of that email is shown below.



(no subject)

Neerja<neerja.sharma@gmail.com>
to me

PFA the required document with this email.

Neerja

Find out which of the following email etiquettes are missing in it. Justify your answer.

NOTES

a) Subject of the mail
b) Formal greeting
c) Self-explanatory terms
d) Identity of the sender
e) Regards

21. Sumit got good marks in all the subjects. His father gifted him a laptop. He would like to make Sumit aware of health hazards associated with inappropriate and excessive use of laptop. Help his father to list the points which he should discuss with Sumit.

Chapter 7 Project Based Learning

> “An idea that is developed and put into action is more important than idea that exists only as an idea.”
>
> — Gautam Buddha

12149CH07

In this chapter

- » *Introduction*
- » *Approaches for Solving Projects*
- » *Teamwork*
- » *Project Descriptions*

7.1 Introduction

Project based learning gives a thorough practical exposure to students regarding a problem upon which the project is based. Through project based learning, students learn to organise their project and use their time effectively for successful completion of the project. Projects are developed generally in groups where students can learn various skills such as working together, problem solving, decision making, and investigating activities. Project based learning involves the steps such as analysing the problem, formulating the problem into small modules, applying the mechanism or method to solve each module and then integrating the solution

NOTES

of all the modules to arrive at the complete solution of the problem. To solve a problem it is required that those who work on it gather the relevant data and process it by applying a particular method. Data may be collected as per the requirement of the project in a particular format. All the team members should associate themselves to accomplish the task. After collecting the data, it should be processed to solve the problem. The results should be reported in a predetermined format.

7.2 Approaches for Solving Projects

The approach followed for the development and completion of a project plays a pivotal role in project-based learning. There are several approaches to execute a project such as modular approach, top down approach and bottom up approach. A structured or a modular approach to a project means that a project is divided into various manageable modules, and each of the modules has a well-defined task to be performed with a set of inputs. This would lead to a set of outputs which when integrated leads to the desired outcome.

Different steps involved in Project Based Learning (Figure 7.1) are:

(1) Identification of a project: The project idea may come through any real life situation. For example, one could think of doing a project for organising a seminar. One needs to understand the usefulness of the project and its impact. Students must be encouraged to undertake interdisciplinary projects.

(2) Defining a plan: Normally for any kind of project, there are several project members involved in it. One project leader has to be identified. The roles of project leader and each project member have to be clearly defined. Students who are performing a project must be assigned with specific activities. The various tools for executing these activities must be known. To obtain a better solution, one should always think of the extreme situations.

(3) Fixing of a time frame and processing: Every project is a time relevance project. A student must understand the importance of time frame for completion of the project. All the activities which are performed in the projects require a certain amount of time. Every project must be well structured and at the same time it must be flexible in its time frame.

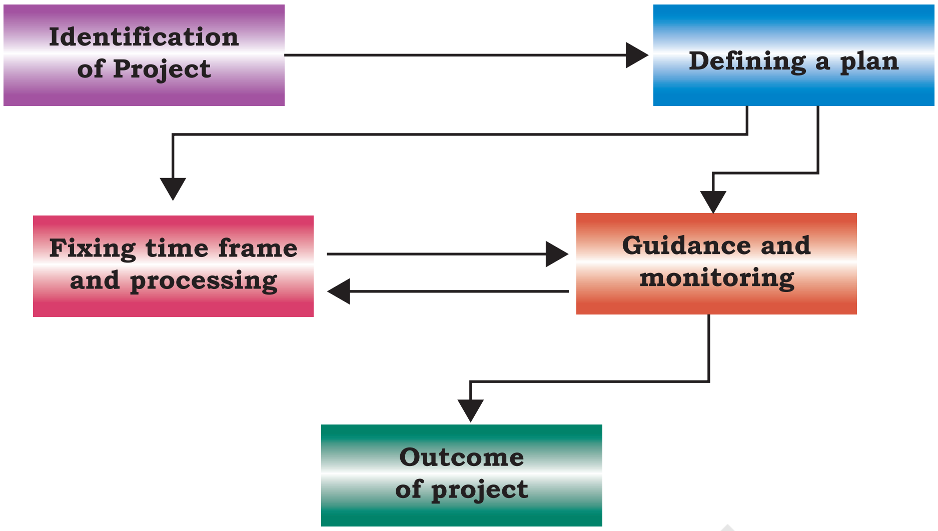


Figure 7.1: Steps in project based learning

(4) Providing guidance and monitoring a project: Many times, the participants in the project get stuck up with a particular process and it becomes impossible to proceed further. In such a case, they need guidance, which can be obtained from various resources such as books, websites and experts in the field. While it is essential that the project leader should ensure monitoring of the project, the guide teacher also helps in monitoring the project.

(5) Outcome of a project: One needs to understand thoroughly the outcome of a project. The outcome can be single, or it can be multiple. The output of a project can be peer reviewed and can be modified as per the feedback from the guide teacher or other users.

7.3 TEAMWORK

Many real life tasks are very complex and require a lot of individuals to contribute in achieving them. An effort made collectively by individuals to accomplish a task is called teamwork.

For example, in many sports, there is a team of players. These players play together to win a match. Take an example of a cricket team. We find that even if a bowler bowls a good ball, but if the fielder cannot take a catch then a wicket cannot be taken. So, in order to take a catch, efforts of a bowler as well as of the fielders

NOTES

are needed. To win a cricket match, contributions from all the team members in all the three areas — batting, bowling and fielding are required.

7.3.1 Components of Teamwork

Apart from technical proficiency, a wide variety of other components make a successful teamwork. It comprises skilled team members with specific roles to achieve the goal.

(A) Communicate with Others

When a group of individuals perform one job, it is necessary to have effective communication between the members of the team. Such communication can be done via e-mails, telephones, or by arranging group meetings. This helps the team members to understand each other and sort out their problems to achieve the goal effectively.

(B) Listen to Others

It is necessary to understand the ideas of others while executing a job together. This can be achieved when the team members listen to each other in group meetings, and follow steps that are agreed upon.

(C) Share with Others

Ideas, images and tools need to be shared with each other in order to perform a job. Sharing is an important component of teamwork. Any member of the team who is well versed in a certain area should share the expertise and experience with others to effectively achieve the goal within the time frame.

(D) Respect for Others

Every member of the team must be treated respectfully. All the thoughts and ideas that are put forth in the group meetings may be respected and duly considered. Not respecting the views of a particular member may cause problems and that particular team member may not give his best.

(E) <L3>Help Others

A helping hand from every member is a key to success. Sometimes, help from people who are not a part of the team is also obtained in order to accomplish a job.

(F) Participate

All the team members must be encouraged by each other to participate in completing the project and also in discussions in group meetings. Also, every member should take an active participation so that they feel their importance in the team.

7.4 PROJECT DESCRIPTIONS

In this section, some examples of project works are given, which can be taken up in groups under project based learning. However, a group may choose any other project in consultation with the guide teacher.

7.4.1 Project I : Online Shopping Platform

Description

Murugan plans to launch an online shopping platform—'APPAREL EASY'. He plans to have two broad categories of merchandise—Men, Women. Under both the categories— Clothing, Footwear and Accessories will be the sub-categories. Also, on his shopping platform, he is planning to launch two mega events— Festive Sale (a month before Diwali to Christmas), End of Season Sale (February and August). Murugan also wants to keep a record of his monthly revenue generation sales and category wise sales, with special focus on mega events. A record should also be kept on discounts being offered by the manufacturers, payment sites or any discount offered as a promotional campaign by the APPAREL EASY portal.

Specification

The details of the Men and Women apparels should be stored in a data file with fields as Apparel Code, Name, Category, Size, Price, Customer Name, Payment Mode, Discount Code, etc.

If the Category is Men, then apparels can be Men's Trousers, Men's Shirt, Men's Jeans, Men's T-shirt. If the Category is Women, then the apparels can be Skirts, Top, Pants, Jeans, Kurta, etc.

If the Payment Mode is Credit or Debit Card, then the Credit card number, name, CVV and validity should be entered.

If the Payment Mode is Cash on Delivery(COD), then no details to be asked.

Randomly select the merchandise to be put on sale. The selected merchandise should not be more than 70 per cent of the total merchandise.

NOTES

The discount code can be either FEST (for Festive) or EOS (for End of Season). The discounts for FEST will be 10 per cent and for EOS will be 15 per cent.

You need to visualise the data structure, keeping all the requirements of Murugan in mind, and then implement it using Python Pandas. Thereafter, you need to design a software to store details of the merchandise to put them online for sale. At the same time, records of customers visiting the e-commerce site and the number of customers placing the order also have to be maintained. The data collected should be plotted appropriately to help Murugan make decisions for future marketing and promotion strategies.

7.4.2 Project II: Automating a Books Donation Camp

Description

Realising the importance of Reduce, Reuse and Recycle, the Bookworm club every year organises a Book Donation Camp. The Book Donation camp collects books and notebooks. The volunteers assess the condition of the books and categorise them as Fit, Needs mending, or Unfit. The unfit books' pages are used to create paper bags and envelopes. The other categories of books are resold at half the price. They accept notebooks that have pages left in them. The pages are torn from the notebooks, and are attractively bound to create a new notebook and sold. They create a variety of recycled objects and sell them. They want to create a software for this purpose and store details about the camp. To be able to efficiently store, retrieve and visualise data, they need to implement the following using Pandas.

Specification

The details of collections are stored in a CSV file with column headings as Item category, Item ID, Item name, Item type, Condition.

If the Item Category is Book, then the Item Type can be either Academic or Non Academic, and Item Id shall be prefixed with a 'B'. In case of Academic, class shall be entered.

If the Item Category is Notebook then the Item can be Single line, Four Line, Five Line, and Item Id shall be prefixed with an 'N'.

Condition can only be Fit, Needs Mending or Unfit. After the items are refurbished, the data are stored in another CSV file containing the following column headings: Item id, Item name, Item Category, Quantity, Price. Item Category can be Paper bags, Notebook, Books. In case of books Class is also to be entered.

Another CSV file to store orders is created that stores Item Category, Item name, Quantity and Price. In case of an order, the refurbished CSV shall update the quantity.

To ensure effective decision making, it is required that different data are plotted using appropriate plots to show sales, items refurbished, and items collected.

7.4.3 Project III: A survey of the effect of social networking sites on behaviour of teenagers

Description

With the Internet revolution everyone today is now connected. Teenagersspend a good amount of time on social networking sites, and it plays a vital role in their behaviour. It is considered that excessive use of social networking sites has sometimes a serious impact on the mental health of individuals. A well-crafted survey questionnaire can help in exploring and finding many facts.

Specifications

1. Create a survey questionnaire using any of the freely available online tools (such as google forms) and store the responses in a CSV file.
2. Prepare some data analysis questions that you expect them to answer
3. Import the CSV file in Pandas DataFrame
4. Perform statistical computation such as mean, median, etc., with respect to the identified questions
5. Visualise the findings of the survey using appropriate charts.

7.4.4 Project IV: Utilising an open data source to use a national, state or district level Dataset

Description

Open Government Data (OGD) Platform India www.data.gov.in is a platform for supporting open data initiative of Government of India. From this platform, let us consider the dataset “Special Tabulation on Adolescent and youth population classified by various

Notes

parameters for India, States and Union Territories, 2011". The dataset was contributed by the Ministry of Home Affairs, Government of India, and released under National Data Sharing and Accessibility Policy (NDSAP). The dataset was published on portal on 07/09/2015.

Statistics of the Data Set:

Number of rows: 12168

Number of columns: 123

Descriptions of some of the columns are given below:

State: Serial numbers given to states

Area Name: Name of the states and union territories

Total/Rural/Urban: Data about the total, rural or urban areas of a state or UT.

Adolescent and youth: Data for different age groups

Total Male: Total number of males

Total Female: Total number of females

SC-M: Total number of males of Scheduled Castes(SC)

SC-F: Total number of females of Scheduled Castes(SC)

ST-M: total number of males of Scheduled Tribes(ST)

ST-F: total number of females of Scheduled Tribes(ST)

Literates-M: total number of literate males

Literates-F: total number of literate females

LiteratesSC-M: total number of literate males of Scheduled Castes(SC)

LiteratesSC-F: total number of literate females of Scheduled Castes(SC)

LiteratesST-M: total number of literate males of Scheduled Tribes(ST)

LiteratesST-F: total number of literate females of Scheduled Tribes(ST)

Illiterates-M: total number of illiterate males

Illiterates-F: total number of illiterate females

IlliteratesSC-M: total number of illiterate males of Scheduled Castes(SC)

IlliteratesSC-F: total number of illiterate females of Scheduled Castes(SC)

IlliteratesST-M: total number of illiterate males of Scheduled Tribes(ST)

IlliteratesST-F: total number of illiterate females of Scheduled Tribes(ST)

*MainWorker-M:*total number of main worker males

*MainWorker-F:*total number of main worker females

MainWorkerSC-M: total number of main worker males of Scheduled Castes(SC)

MainWorkerSC-F: total number of main worker females of Scheduled Castes(SC)

MainWorkerST-M: total number of main worker males of Scheduled Tribes(ST)

MainWorkerST-F: total number of main worker females of Scheduled Tribes(ST)

MarginalWorker-M: total number of marginal worker males

MarginalWorker-F: total number of marginal worker females

MarginalWorkerSC-M: total number of marginal worker males of Scheduled Castes(SC)

MarginalWorkerSC-F: total number of marginal worker females of Scheduled Castes(SC)

MarginalWorkerST-M: total number of marginal worker males of Scheduled Tribes(ST)

MarginalWorkerST-F: total number of marginal worker females of Scheduled Tribes(ST)

Specifications

On such a large dataset, various types of questions can be answered by doing different analysis of data. Following is a list of some of the possible queries that can be answered by analysing the dataset:

1. What is the total population, total male population and total female population aged 10 to 24 in India?
2. Which State or Union Territory in India has the maximum number of illiterates in the youth ages?
3. What is the percentage of people working as a marginal worker?
4. List the top 5 states or union territories which have the maximum population working as a marginal worker.
5. Compare the sex ratio of urban areas and rural areas using appropriate graph.
6. Which state has the highest and the lowest percentage of literate Scheduled Tribes and Scheduled Castes?
7. For each state, compare the no. of female marginal workers with no. of male marginal workers. Use appropriate graphs.
8. What percentage of Scheduled Tribes lives in urban areas? Draw a pie chart showing the proportion of literate and illiterates scheduled tribes living in urban areas.
9. What is the state wise ratio of literates vs. illiterates in all age groups?

NOTES

10. Which state is home to the maximum no. of ST in India? Which state has the minimum no. of ST in India?
11. For each state, find the no. of literate females and no. of literate males. Draw a bar graph for the same. Which state has the highest ratio of literate female vs literate male and which state has the minimum?

A project work can be carried out by taking any 4–5 of the above questions and any other similar questions, and solving them step-by-step, with detailed explanation and documentation. As an example, in the following pages, we will solve the first question. This will give us an idea about how the other questions are to be answered.

Task 1: What is the total population, total male population and total female population aged 10 to 24 in India?

Solution:

Prerequisite: we need to first download the CSV file through the QR code given at the beginning of this chapter.

Step 1: Read the CSV file in a DataFrame

Step 2: Check the shape of the DataFrame

Step 3: View the columns

Step 4: Filter data

a. Identify the columns that you wants to use for plotting

b. Identify the number of rows required for plotting

Step 5: Create a new DataFrame containing the filtered data

Step 6: Rename the columns for ease of use

Step 7: Group data as per the requirement

Step 8: Plot data as a barchart for the DataFrame obtained in Step 7.

Let us now write the code for the above identified steps:

Step 0: Import required libraries.

```
import pandas as pd
importmatplotlib.pyplot as plt
```

Step 1: Read the CSV file in a DataFrame.

```
# Add path to the CSV file in your computer
data=pd.read_csv("PCA_AY_2011_Revised.csv")
df=pd.DataFrame(data)
```

Step 2: Check shape of the DataFrame.

```
print(df.shape)
```

We get the output showing the dataset contains 12168 rows and 123 columns.

Step 3: Display the columns.

```
print(df.columns.values)
```

A part of the output produced for the 123 columns is shown below:

```
['Table No.' 'State Code' 'District Code' 'Area Name'
     'Total/ Rural/ Urban' 'Adolescent and youth categories'
     'Total Population - Persons' 'Total Population - Males'
     'Total Population - Females' 'Scheduled Caste - Persons'
     'Scheduled Caste - Males' 'Scheduled Caste - Females'
                              ...
     'Scheduled Tribe Marginal Worker - Household Industry - Males'
     'Scheduled Tribe Marginal Worker - Household Industry - Females'
     'Scheduled Tribe Marginal Worker - Other Workers - Persons'
     'Scheduled Tribe Marginal Worker - Other Workers - Males'
     'Scheduled Tribe Marginal Worker - Other Workers - Females']
```

Step 4: Filter Data.

a. Identify the columns that you want to use for plotting.

   For our analysis, we will consider only the columns ‘Area Name’; ’Total/Rural/Urban’, 'Adolescent and youth categories', and 'Total Population - Persons' .

b. Identify the number of rows required for plotting.

   In order to decide the number of rows, we needs to check the values in the column ‘Area Name

```
print(df[‘Area Name’]
```

The following is the output:

```
0                              INDIA
1                              INDIA
2                              INDIA
3                              INDIA
4                              INDIA
                    ...
```

```
12163    District - South Andaman (03)
12164    District - South Andaman (03)
12165    District - South Andaman (03)
12166    District - South Andaman (03)
12167    District - South Andaman (03)
Name: Area Name, Length: 12168, dtype: object
```

Step 5: Create a new DataFrame containing the filtered data.

Suppose, we want to consider data for ‘Area Name’ = ‘INDIA’ only, Therefore, we shall create a new DataFrame df1 containing only the filtered data, using the following syntax:

```
df.loc[row selection, column selection]
df1=df.loc[(df['Area Name'] == 'INDIA'),
'Area Name':'Total Population - Females']
```

In the above statement, df [‘Area Name’] is used to select the required rows.We apply slicing on column labels to select the columns starting from ‘Area Name’ till ‘Total Population — Females’

Step 6: The names of the columns in the DataFrame are too long. The following statement can be used to rename the columns.

```
df1.columns = ['Area', 'Class', 'Category',
'TotalPop', 'MalePop', 'FemalePop']
```

Step 7: Group data as per the requirement.

We decided to plot TotalPop, MalePop, FemalePopwith respect to Category. But, on inspecting the DataFrame df1 we have noticed that the Category column contains data under six different categories — '10-14', '15-19', '20-24', 'Adolescent (10-19)', 'All Ages', 'Youth (15-24)'. print(df1)

```
Area  Class                  Category    TotalPop MalePop   FemalePop
0   INDIA  Total           All Ages  1210854977  623270258  587584719
1   INDIA  Total              10-14   132709212   69418835   63290377
2   INDIA  Total              15-19   120526449   63982396   56544053
3   INDIA  Total              20-24   111424222   57584693   53839529
4   INDIA  Total  Adolescent (10-19)   253235661  133401231  119834430
5   INDIA  Total       Youth (15-24)   231950671  121567089  110383582
6   INDIA  Rural           All Ages   833748852  427781058  405967794
7   INDIA  Rural              10-14    96804494   50488158   46316336
8   INDIA  Rural              15-19    83902472   44570557   39331915
```

```
9   INDIA  Rural                 20-24   73835046   38138662   35696384
10  INDIA  Rural  Adolescent (10-19)  180706966   95058715   85648251
11  INDIA  Rural         Youth (15-24)  157737518   82709219   75028299
12  INDIA  Urban              All Ages  377106125  195489200  181616925
13  INDIA  Urban                 10-14   35904718   18930677   16974041
14  INDIA  Urban                 15-19   36623977   19411839   17212138
15  INDIA  Urban                 20-24   37589176   19446031   18143145
16  INDIA  Urban  Adolescent (10-19)   72528695   38342516   34186179
17  INDIA  Urban         Youth (15-24)   74213153   38857870   35355283
```

Therefore, to plot TotalPop, MalePop, FemalePop, we should do grouping of these six categories and find the sum for each type of population. This will help to provide a complete picture. The GROUP BY() function when applied on the column 'Category' on our DataFrame df1, gives us the following result:

```
d = df1.GROUP BY('Category').sum()
TotalPopMalePopFemalePop
Category
10-14                 265418424   138837670   126580754
15-19                 241052898   127964792   113088106
20-24                 222848444   115169386   107679058
Adolescent (10-19)    506471322   266802462   239668860
All Ages             2421709954  1246540516  1175169438
Youth (15-24)         463901342   243134178   220767164
```

We are interested only in the categories '10-14', '15-19' and '20-24'. So, let us drop the remaining rows using the following Python statement:

```
d= d.drop(['Adolescent (10-19)','All
Ages','Youth (15-24)'],axis= 0)
TotalPopMalePopFemalePop
Category
10-14    265418424  138837670  126580754
15-19    241052898  127964792  113088106
20-24    222848444  115169386  107679058
```

Step 8: Plot the data as a barchart for the DataFrame obtained in Step 7.

```
d.plot(kind='bar')
plt.show()
```

The barchart shown at Figure 7.2 is produced as the output. The value (1e8) marked at the top is offset that is being displayed for the y axes which corresponds to scientific notation which is used for numbers outside a specified range.

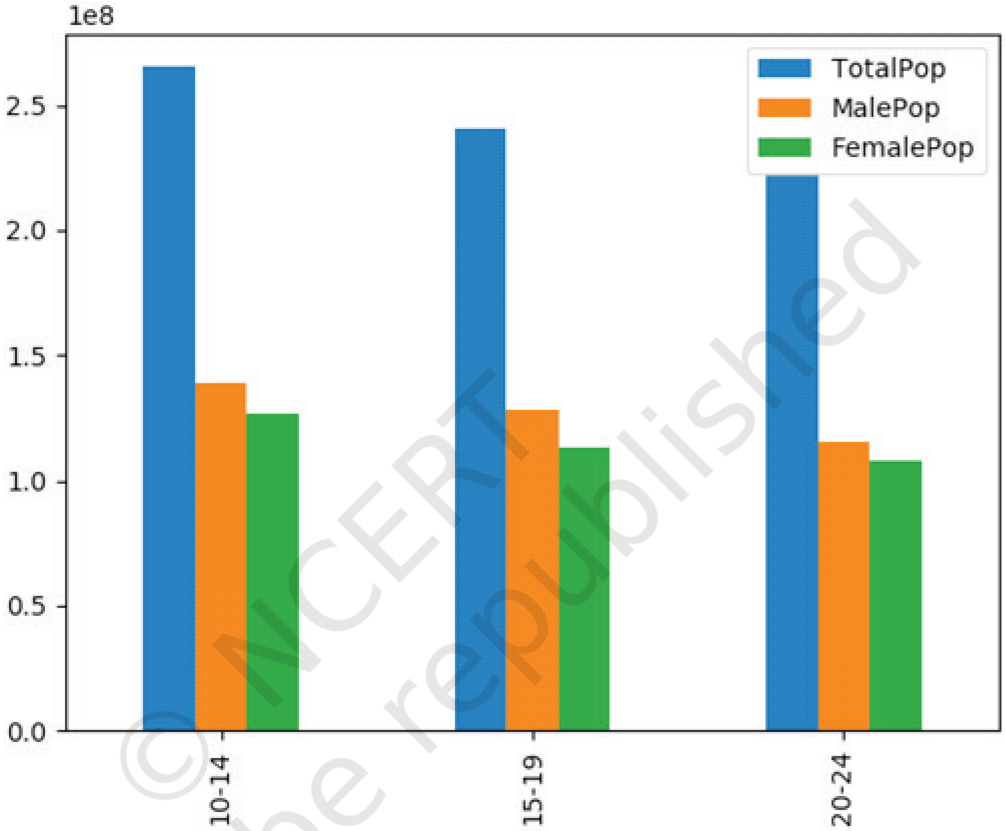


Figure 7.2: Barchart showing population in different categories

NOTES

NOTES