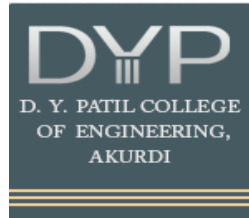


D. Y. Patil College of Engineering, Akurdi, Pune-44

Department of Information Technology

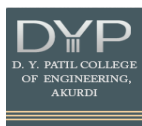


LABORATORY MANUAL

TE (INFORMATION TECHNOLOGY) (SEMESTER – II)

314457: DS & BDA Lab

2019 course



D. Y. Patil College of Engineering, Akurdi, Pune

Department of Information Technology

Vision of Institute:

“Empowerment through Knowledge”.

Mission of Institute:

M1- To educate the students to transform them as professionally competent and quality conscious engineers.

M2-To provide Conducive Environment for Teaching-Learning and overall personality development.

M3-To culminate the institute into an International seat of excellence.

Vision of Department

Developing globally competent IT professional for sustainable growth of humanity.

Mission of Department

M1 Build a strong foundation and techniques for problem-solving and inculcate communication skills as an integral component of Information Technology

M2 Develop competency skills in the faculty members and students to serve the societal challenges and needs in lieu of its multidisciplinary applications in the field of Information Technology

M3 Encourage development of strong technical skills and knowledge and to encourage students to undergo research in the field of Information Technology

M4 Nurture students to become ethical and committed lifelong IT professionals

M5 Empower students with strong decision-making skills and technical competency to accomplish start-up ideas in the field of IT Engineering

The Program Educational Objectives (PEOs):

The Program Educational Objectives (PEOs) are as follows:

- 1) **Core Competency:** To provide graduates with a solid foundation in Mathematics, Science, Engineering fundamentals required to solve complex Software Engineering Problem.
- 2) **Breadth:** To impart the knowledge and skills in the field of Information Technology; and to comprehend, analyze, design and create novel products and solutions for the real-time and Complex Engineering problems of any domain with innovative approaches.
- 3) **Professionalism:** To inculcate in graduates, professional and ethical values, effective communication skills, teamwork, multidisciplinary approach, and ability to relate engineering issues in broader social context.
- 4) **Learning Environment:** To provide graduates with an academic environment that makes them aware of excellence in leadership, presentation, time management and ethics leading them to become responsible and competent professionals prepared to address challenges in the field of IT at global level.
- 5) **Attainment:** To empower graduates with an attitude and skills of Research, Entrepreneur and Higher education in the field of Information Technology.

Program Specific Outcomes:

- 1) Apply design methodologies, application development tools, engineering skills in Software Engineering Domains and IT Application areas like Cloud Computing, Software Testing, Mobile App Development, etc.
- 2) Aspire to pursue Higher Education in the specialized fields of IT Engineering and management programs like Data science, Cyber Security, Artificial Intelligence, etc.
- 3) Formulate decision-making skills, IT Engineering skills, and knowledge to implement start-up ideas as an entrepreneur in the fields such as Cyber Security, Mobile Application Development, etc.
- 4) Devise to design, implement, and evaluate IT based software systems to serve the needs of society or IT industries at large.

Program Outcomes :

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



D. Y. Patil College of Engineering, Akurdi, Pune-44
Department of Information Technology

Course Code: 314457

Course Name: DS & BDA Lab (2019 Course)

Teaching Scheme:

Credits: 01

Examination Scheme:

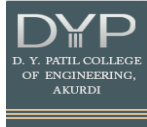
Practical: 2hrs/week

TW: 25 Marks

Practical: 25 Marks

Course Outcomes:

Course Outcome	Statement
	On completion of the course, students will be able to–
CO1	Apply Big data primitives and fundamentals for application development.
CO2	Explore different Big data processing techniques with use cases.
CO3	Apply the Analytical concept of Big data using Python.
CO4	Visualize the Big Data using Tableau
CO5	Design algorithms and techniques for Big data analytics.
CO6	Design and develop Big data analytic application for emerging trends.



D. Y. Patil College of Engineering, Akurdi, Pune-
44 Department of Information Technology

Sr.No.	Title of Assignment
Group A: Assignments based on the Hadoop	
1.	Single node/Multiple node Hadoop Installation.
2.	Design a distributed application using MapReduce(Using Java) which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.
3.	Write an application using HiveQL for flight information system which will include <ol style="list-style-type: none">Creating, Dropping, and altering Database tables.Creating an external Hive table.Load table with data, insert new values and field in the table, Join tables with HiveCreate index on Flight Information TableFind the average departure delay per day in 2008.
Group B: Assignments based on Data Analytics using Python	
1.	Perform the following operations using Python on the Facebook metrics data sets <ol style="list-style-type: none">Create data subsetsMerge DataSort DataTransposing DataShape and reshape Data
2.	Perform the following operations using Python on the Air quality and Heart Diseases data sets <ol style="list-style-type: none">Data cleaningData integrationData transformationError correctingData model building
3.	Integrate Python and Hadoop and perform the following operations on forest fire dataset <ol style="list-style-type: none">Data analysis using the Map Reduce in PyHadoopData mining in Hive
4.	Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3 (Group B).

5.	<p>Perform the following data visualization operations using Tableau on Adult and Iris datasets.</p> <ul style="list-style-type: none"> a. 1D (Linear) Data visualization b. 2D (Planar) Data Visualization c. 3D (Volumetric) Data Visualization d. Temporal Data Visualization e. Multidimensional Data Visualization f. Tree/ Hierarchical Data visualization
Group C: Model Implementation	
1.	Create a review scrapper for any ecommerce website to fetch real time comments, reviews, ratings, comment tags, customer name using Python.
2.	Develop a mini project in a group using different predictive models techniques to solve any real life problem. (Refer link dataset- https://www.kaggle.com/tanmoyie/us-graduate-schools-admission-parameters)

GUIDELINES FOR STUDENTS

Guidelines for Student's Lab Journal

1. Students should submit term work in the form of a handwritten journal based on a specified list of assignments.
2. Practical Examination will be based on the term work.
3. Candidate is expected to know the theory involved in the experiment.
4. The practical examination should be conducted if and only if the journal of the candidate is complete in all respects.

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.
3. Appropriate knowledge of usage of software and hardware related to the respective laboratory should be checked by the concerned faculty member.

Guidelines for Laboratory Conduction

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers of the program in a journal may be avoided. There must be hand-written write-ups for every assignment in the journal. The DVD/CD containing student's programs should be attached to the journal by every student and the same to be maintained by the department/lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Practical Examination

1. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement.
2. Student's understanding of the fundamentals, effective and efficient implementation can be evaluated by asking relevant questions based on implementation of experiments he/she has carried out.

RUBRICS FOR LABORATORY ASSESSMENT

1. Attendance

Assessment Outcome Dimensions	Poor (1)	Satisfactory(2)	Good (3)	Very Good (4)	Excellent (5)
1.Attendance with Involvement of Student (5M)	Passive observer	Very little involvement	Good Involvement in performing experiment	Individual Involvement in performing experiment	Individual and self - Involvement in performing experiment

2. Viva

Assessment Outcome Dimensions	Poor (1)	Satisfactory(2)	Good (3)	Very Good (4)	Excellent (5)
1.Preparation and Basic Knowledge (5M)	No preparation	Little Knowledge	Prepared Well	Very well prepared	Advance Knowledge
2.Program development and execution (5M)	Not Executed	Partially executed	Executed	Executed without additional modification	Executed with additional modification
3.Punctuality and Ethics (5M)	Attendance Below 50% and not following the lab instructions	Attendance 50% to 75% And sometimes copies the program	Regular attendance 75-00% and follows the instruction and try to perform on his own	Regular attendance 80-90% and follows the instruction and try to perform on his own	90-100 % attendance, follows all instructions and execute the program on his own

3. Presentation

Assessment Outcome Dimensions	Poor (1)	Satisfactory(2)	Good (3)	Very Good (4)	Excellent (5)
Journal Presentation (5M)	Not Prepared	Incomplete	Completed documentation	well documented	Very well documented

Outcome: Student will be able to

- Apply knowledge to real life examples and develop practical approach
- Design Basic Application.

Note: Students with poor marks should repeat the assignment

ProblemStatement:

**Aim: To Perform Hadoop Installation (Configuration) on a)Single Node
b)Multiple Node**

Tools/Environment:

Ubuntu

RelatedTheory:

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. To process data, Hadoop transfers packaged code for nodes to process in parallel based on the data that needs to be processed. This approach takes advantage of data locality— nodes manipulating the data they have access to— to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking. The base Apache Hadoop framework is composed of the following modules:

- Hadoop Common – contains libraries and utilities needed by other Hadoop modules;
 - Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
 - Hadoop YARN – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications; and
 - HadoopMapReduce – an implementation of the MapReduce programming model for large scale data processing.
-

a)Single Node:

Steps for Compilation & Execution

```
sudo apt-get update
sudo apt-get install openjdk-7-jre-headless sudo
apt-get install openjdk-7-jdk sudo apt-get install
sshsudo apt-get install rsync
# Download hadoop from :
http://www.eu.apache.org/dist/hadoop/common/stable/hadoop-
2.9.0.tar.gz
# copy and extract hadoop-2.9.0.tar.gz in home folder
# rename the name of the extracted folder from hadoop-2.9.0 to hadoopreadlink -f
/usr/bin/javac
# find whether ubuntu is 32 bit (i686) or 64 bit (x86_64) uname -i
gedit ~/hadoop/etc/hadoop/hadoop-env.sh
# add following line in it # for 32
bit ubuntu
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
# for 64 bit ubuntu
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
# save and exit the file
# to display the usage documentation for the hadoop script try next command ~/hadoop/bin/hadoop

1. For standalone mode
mkdir input
cp ~/hadoop/etc/hadoop/*.xml input
~/hadoop/bin/hadoop jar ~/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar grep
input output 'us[a-z.]+' cat output/*
# Our task is done, so remove input and output folders rm -r
input output

# 2. Pseudo-Distributed mode
# get your user name
whoami
# remember your user name, we'll use it in the next step gedit
~/hadoop/etc/hadoop/core-site.xml

<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:1234</value>
</property>
</configuration>

gedit ~/hadoop/etc/hadoop/hdfs-site.xml
<configuration>
<property>
<name>dfs.replication</name>
```

```

<value>1</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>file:///home/your_user_name/hadoop/name_dir</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>file:///home/your_user_name/hadoop/data_dir</value>
</property>
</configuration>

```

```

#Setup passphraseless/passwordlesssshssh-keygen -t dsa -P " -f
~/.ssh/id_dsa cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys export
HADOOP_PREFIX=/home/your_user_name/hadoopsshlocalhost
# typeexit in the terminal to close the ssh connection (very important) Exit

```

The following instructions are to run a MapReduce job locally.

```

#Format the filesystem:( Do it only once )
~/hadoop/bin/hdfsnamenode -format
#Start NameNode daemon and DataNode daemon:
~/hadoop/sbin/start-dfs.sh
#Browse the web interface for the NameNode; by default it is available at: http://localhost:50070/
#Make the HDFS directories required to execute MapReduce jobs:

~/hadoop/bin/hdfsdfs -mkdir /user
~/hadoop/bin/hdfsdfs -mkdir /user/your_user_name
#Copy the sample files (from ~/hadoop/etc/hadoop) into the distributed filesystemfolder(input)
~/hadoop/bin/hdfsdfs -put ~/hadoop/etc/hadoop input
#Run the example map-reduce job
~/hadoop/bin/hadoop jar ~/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep
input output 'us[a-z.]+'
#View the output files on the distributed filesystem
~/hadoop/bin/hdfsdfs -cat output/*
#Copy the output files from the distributed filesystem to the local filesystem and examine them:
~/hadoop/bin/hdfsdfs -get output output
#ignore warnings (if any) cat
output/*
# remove local output folder rm -r
output
# remove distributed folders (input & output)
~/hadoop/bin/hdfsdfs -rm -r input output #When you're
done, stop the daemons with
~/hadoop/sbin/stop-dfs.sh

```

Conclusion: In this way the Hadoop was installed & configured on Ubuntu for BigData.

b)Multiple Node

Install a Multi Node Hadoop Cluster on Ubuntu 14.04

You would need minimum of 2 ubuntu machines or virtual images to complete a multi-node installation. If you want to just try out a single node cluster, follow this article on [Installing Hadoop on Ubuntu 14.04](#).

Installing Java on Master and Slaves

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
```

```
$ sudo apt-get install oracle-java7-installer
# Update Java runtime
$ sudo update-java-alternatives -s java-7-oracle
```

Disable IPv6

As of now Hadoop does not support IPv6, and is tested to work only on IPv4 networks. If you are using IPv6, you need to switch Hadoop host machines to use IPv4. [The Hadoop Wiki](#) link provides a one liner command to disable the IPv6. If you are not using IPv6, skip this step:

```
sudo sed -i 's/net.ipv6.bindv6only\ =\ 1/net.ipv6.bindv6only\ =\ 0/' /etc/sysctl.d/bindv6only.conf
&&sudo invoke-rc.dprocps restart
```

Setting up a Hadoop User

Hadoop talks to other nodes in the cluster using no-password ssh. By having Hadoop run under a specific user context, it will be easy to distribute the ssh keys around in the Hadoop cluster. Lets's create a user **hadoopuser** on **master** as well as **slave** nodes.

```
# Createhadoopgroup
$ sudoaddgrouphadoopgroup
# Createhadoopuser user
$ sudoadduser --ingrouphadoopgrouphadoopuser
```

Our next step will be to generate a ssh key for password-less login between master and slave nodes. Run the following commands only on **master** node. Run the last two commands for each slave node. Password less ssh should be working before you can proceed with further steps.

```
# Login as hadoopuser
$ su - hadoopuser
#Generate a ssh key for the user
$ ssh-keygen -t rsa -P ""
#Authorize the key to enable password less ssh
$ cat /home/hadoopuser/.ssh/id_rsa.pub >> /home/hadoopuser/.ssh/authorized_keys $ chmod 600
authorized_keys
#Copy this key to slave-1 to enable password less ssh
$ ssh-copy-id -i ~/.ssh/id_rsa.pub slave-1
#Make sure you can do a password less ssh using following command. $ ssh slave-1
```

Download and Install Hadoop binaries on Master and Slave nodes

Pick the best mirror site to download the binaries from [Apache Hadoop](http://www.apache.org/hadoop/), and download the stable/hadoop-2.6.0.tar.gz for your installation. **Do this step on master and every slave node.** You can download the file once and then distribute to each slave node using scp command.

```
$ cd /home/hadoopuser
```

```
$ wget http://www.webhostingjams.com/mirror/apache/hadoop/core/stable/hadoop-2.2.0.tar.gz $ tar
xvf hadoop-2.2.0.tar.gz
$ mv hadoop-2.2.0 hadoop
```

Setup Hadoop Environment on Master and Slave Nodes

Copy and paste following lines into your .bashrc file under /home/hadoopuser. **Do this step on master and every slave node.**

```
# Set HADOOP_HOME
export HADOOP_HOME=/home/hduser/hadoop
# Set JAVA_HOME
export JAVA_HOME=/usr/lib/jvm/java-7-oracle
# AddHadoop bin and sbin directory to PATH
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

Update hadoop-env.sh on Master and Slave Nodes

Update JAVA_HOME in /home/hadoopuser/hadoop/etc/hadoop/hadoop_env.sh to following. **Do this step on master and every slave node.**

```
export JAVA_HOME=/usr/lib/jvm/java-7-oracle
```

Common Terminologies

Before we start getting into configuration details, let's discuss some of the basic terminologies used in Hadoop.

- **Hadoop Distributed File System:** A distributed file system that provides high-throughput access to application data. A HDFS cluster primarily consists of a NameNode that manages the file system metadata and DataNodes that store the actual data. If you compare HDFS to a traditional storage structures (e.g. FAT, NTFS), then NameNode is analogous to a Directory Node structure, and DataNode is analogous to actual file storage blocks.
- **Hadoop YARN:** A framework for job scheduling and cluster resource management.

- **HadoopMapReduce:** A YARN-based system for parallel processing of large data sets.

Update Configuration Files

Add/update core-site.xml on **Master and Slave nodes** with following options. Master and slave nodes should all be using the same value for this property **fs.defaultFS**, and should be pointing to master node only.

/home/hadoopuser/hadoop/etc/hadoop/core-site.xml (Other Options)

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hadoopuser/tmp</value>
  <description>Temporary Directory.</description>
</property>

<property>
  <name>fs.defaultFS</name>
  <value>hdfs://master:54310</value>
  <description>Use HDFS as file storage engine</description></property>
```

Add yarn-site.xml on **Master and Slave Nodes**. This file is required for a Node to work as a Yarn Node. Master and slave nodes should all be using the same value for the following properties, and should be pointing to master node only.

/home/hadoopuser/hadoop/etc/hadoop/yarn-site.xml

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.resourcemanager.scheduler.address</name>
  <value>master:8030</value>
</property>
<property>
  <name>yarn.resourcemanager.address</name>
  <value>master:8032</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address</name>
  <value>master:8088</value>
</property>
<property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>master:8031</value>
</property>
<property>
  <name>yarn.resourcemanager.admin.address</name>
  <value>master:8033</value>
</property>
```


Add/update **slaves** file on Master node only. Add just name, or ip addresses of master and all slave node. If file has an entry for localhost, you can remove that. This file is just helper file that are used by hadoop scripts to start appropriate services on master and slave nodes.

```
/home/hadoopuser/hadoop/etc/hadoop/slave
```

```
master slave-1  
slave-2
```

Format the Namenode

Before starting the cluster, we need to format the Namenode. Use the following command only on **master node**:

```
$ hdfsnamenode -format
```

Start the Distributed Format System

Run the following on **master node** command to start the DFS.

```
$ ./home/hadoopuser/hadoop/sbin/start-dfs.sh
```

You should observe the output to ascertain that it tries to start datanode on slave nodes one by one. To validate the success, run following command on master nodes, and slave node.

```
$ su - hadoopuser  
$ jps
```

The output of this command should list *NameNode*, *SecondaryNameNode*, *DataNode* on **master** node, and *DataNode* on all slave nodes. If you don't see the expected output, review the log files listed in Troubleshooting section.

Start the Yarn MapReduce Job tracker

Run the following command to start the Yarn mapreduce framework.

```
$ ./home/hadoopuser/hadoop/sbin/start-yarn.sh
```

To validate the success, run **jps** command again on master nodes, and slave node. The output of this command should list *NodeManager*, *ResourceManager* on **master** node, and *NodeManager*, on all **slave** nodes. If you don't see the expected output, review the log files listed in Troubleshooting section.

Review Yarn Web console

If all the services started successfully on all nodes, then you should see all of your nodes listed under Yarn nodes. You can hit the following url on your browser and verify that: <http://master:8088/cluster/nodes>

Lets's execute a MapReduce example now

You should be all set to run a MapReduce example now. Run the following command

```
$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.2.0.jar pi 30 100
```

Once the job is submitted you can validate that its running on the cluster by accessing following url. <http://master:8088/cluster/apps>

Conclusion: Thus we have tested successfully multimode cluster.

ASSIGNMENTNO.2

ProblemStatement:

Aim: Design a distributed application using MapReduce(Using Java) which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.

Tools/Environment:

Ubuntu

RelatedTheory:

MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner. MapReduce is a processing technique and a program model for distributed computing based on java.

The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).

Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job. The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes.

Under the MapReduce model, the data processing primitives are called mappers and reducers.

Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

Mapstage : The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

Reduce stage : This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

Inserting Data into HDFS:

- The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.
- The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework.
- Input and Output types of a MapReduce job: (Input) <k1,v1> -> map -><k2, v2>-> reduce -><k3, v3> (Output).

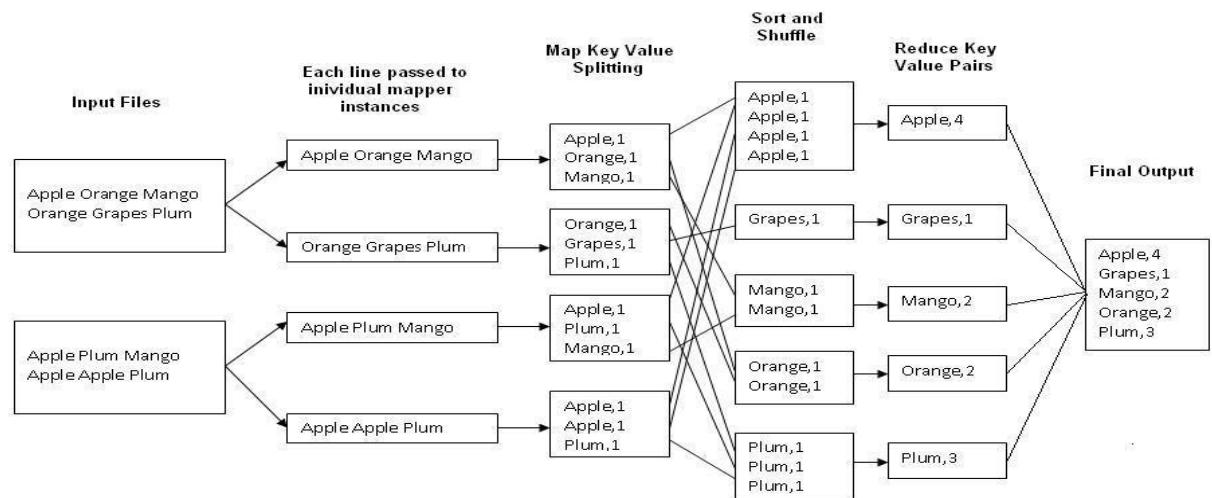


Fig.1 : An Example Program to Understand working of MapReduce Program.

Steps for Compilation & Execution of Program:

Su hadoopuser

#sudomkdiranalyzelogs

ls

#sudochmod -R 777 analyzelogs/

cd

ls cd .. (to
move to home directory)

pwd

ls

cd

pwd

#sudochown -R hadoop1 analyzelogs/

cd

ls

#cd analyzelogs/

ls cd ..

Copy the Files (Mapper.java,Reduce.java,Driver.java to Analyzelogs Folder)

#sudocp /home/priyanka/Desktop/assignment3/* -/analyzelogs/

(Convert access_log_short.txt into access_log_short.csv)

Start HADOOP

#start-dfs.sh

#start-yarn.sh

#jps

cd

cdanalyzelogs

ls pwd

ls

#ls -ltr

```
#ls -al
#sudochmod +r *.*
    pwd
#export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduceclient-core-
2.9.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-clientcommon-
2.9.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-common-
2.9.0.jar:~/analyzlogs/SalesCountry/*:$HADOOP_HOME/lib/*"
(This should be PWD)
```

Compile Java Files

```
# javac -d . SalesMapper.java SalesCountryReducer.java SalesCountryDriver.java    ls
#cd SalesCountry/
ls    cd ..
#sudogedit Manifest.txt
```

Main-class:SalesCountry.SalesCountryDriver

(Press enter)

```
#jar -cfm analyzlogs.jar Manifest.txt SalesCountry/*.class    ls    cd
```

```
#cd analyzlogs/
```

Create Directory on Hadoop

```
#sudomkdir ~/input2000
ls
pwd
#sudocp access_log_short.csv ~/input2000/
# $HADOOP_HOME/bin/hdfsdfs -put ~/input2000 /
# $HADOOP_HOME/bin/hadoop jar analyzlogs.jar /input2000 /output2000

# $HADOOP_HOME/bin/hdfsdfs -cat /output2000/part-00000
```

```
# stop-all.sh
```

```
# jps
```

For GUI

Go to browser(localhost:50070)

Go to utilities(browse directory)

Conclusion: Thus we have learnt how to design a distributed application using MapReduce and process a log file of a system.

ASSIGNMENTNO.3

ProblemStatement:

Aim: Write an application using HiveQL for flight information system which will include

- Creating, Dropping, and altering Database tables.
- Creating an external Hive table.
- Load table with data, insert new values and field in the table, Join tables with Hive
- Create index on Flight Information Table
- Find the average departure delay per day in 2008.

Tools/Environment:

Ubuntu

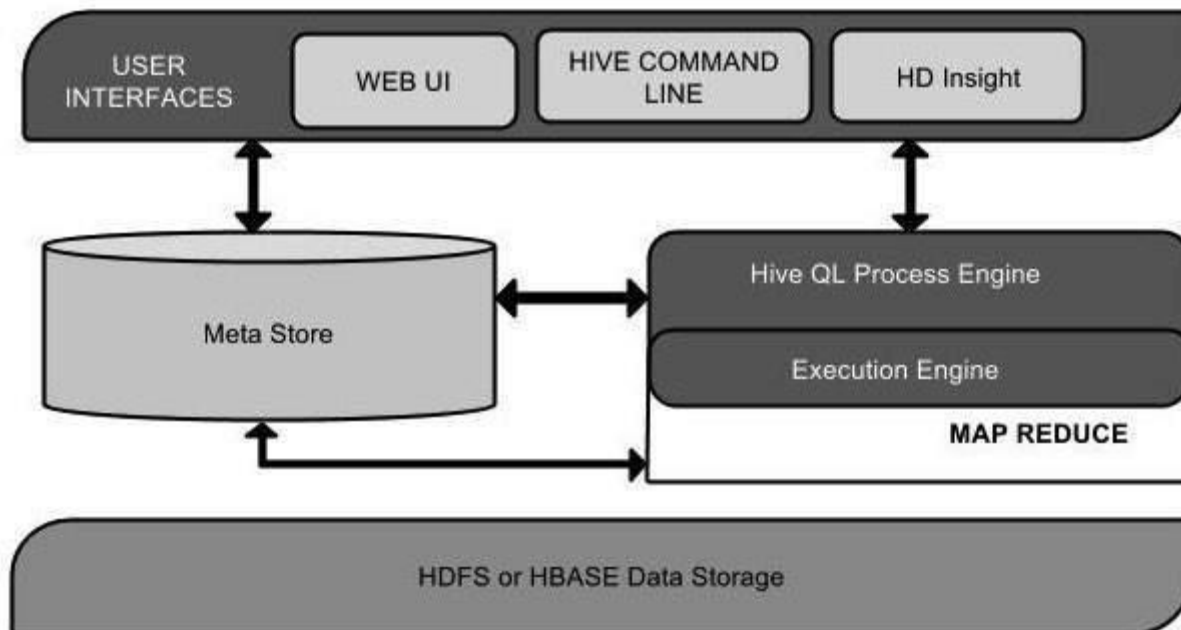
RelatedTheory:

Hive:

Apache Hive is a data warehouse system developed by Facebook to process a huge amount of structure data in Hadoop. We know that to process the data using Hadoop, we need to right complex map-reduce functions which is not an easy task for most of the developers. Hive makes this work very easy for us.

It uses a scripting language called HiveQL which is almost similar to the SQL. So now, we just have to write SQL-like commands and at the backend of Hive will automatically convert them into the map-reduce jobs.

Hive architecture:

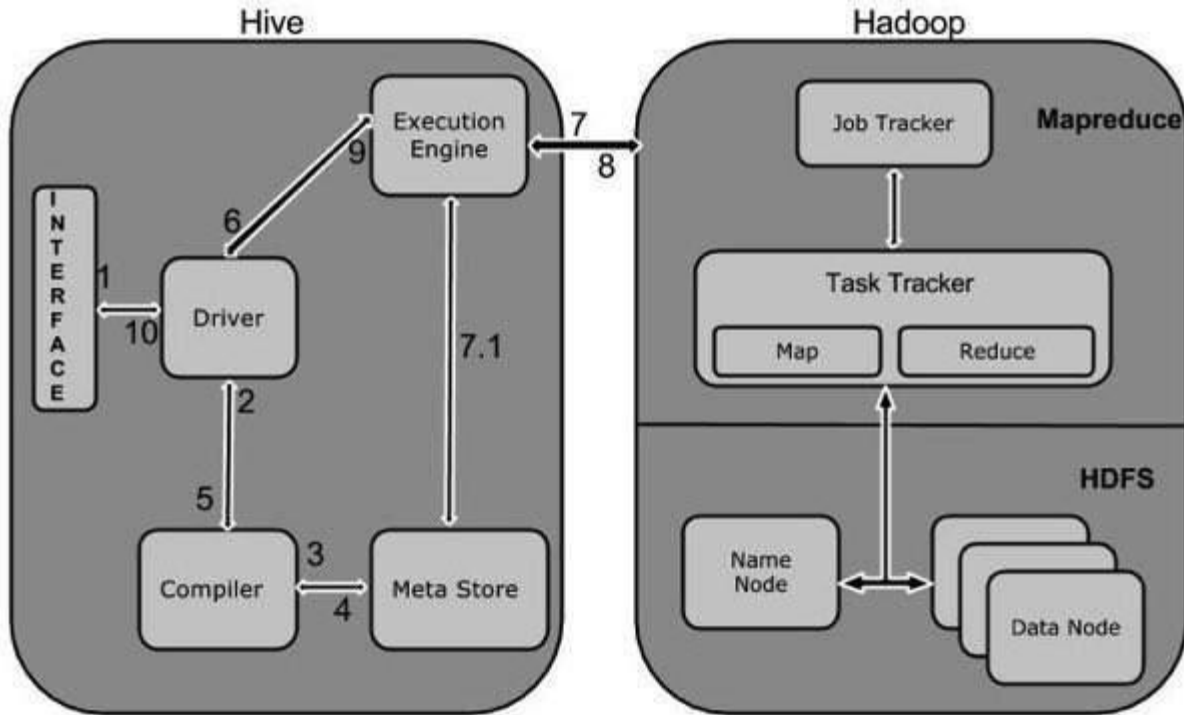


Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping. HiveQL is similar to SQL for querying on schema info on the Metastore.

Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of

MapReduce. Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

Working of Hive:



Data Types in Apache Hive

Hive data types are divided into the following 5 different categories:

1. Numeric Type: TINYINT, SMALLINT, INT, BIGINT
2. Date/Time Types: TIMESTAMP, DATE, INTERVAL
3. String Types: STRING, VARCHAR, CHAR
4. Complex Types: STRUCT, MAP, UNION, ARRAY
5. Misc Types: BOOLEAN, BINARY

Download hive tar

```
Unzip tar, rename
Edit bashrc- add export HIVE_HOME=/home/stud/Downloads/hive
starthadoop services
go to hadoop/bin and make 2 dir
hdfsdfs -mkdir -p /user/hive/warehouse
hdfsdfs -mkdir -p /tmp/hive
hdfsdfs -chmod 777 /tmp
hdfsdfs -chmod 777 /user/hive/warehouse
hdfsdfs -chmod 777 /tmp/hive
go to hive/bin
cd..(come out from hadoop bin)
cd..(come out from hadoop)
cd hive/bin
ls
./schematool -initSchema -dbtype derby
```

output-

```
Metastore connection URL:
jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :      org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:      APP
Starting metastore schema initialization to 1.2.0
Initialization script hive-schema-1.2.0.derby.sql
Initialization script completed
schemaTool completed
```

```
ls
start hive terminal
./hive
```

```
1.show databases;
2.create database d1;
```

```
3.create table emp(enamestring,esalint) row format delimited fields terminated by
'_'stored as textfile;
```

```
4.load data local inpath '/home/stud/Desktop/sample.txt' into table emp1;
```

```
5.create table flight(Year int, Month int, DayofMonthint, DayOfWeekint, DepTime int,
CRSDepTimeint, ArrTimeint, CRSArrTimeint, UniqueCarrier string, FlightNumint, TailNum
string, ActualElapsedTimeint, CRSElapsedTimeint, AirTimeint, ArrDelayint, DepDelayint,
Origin string, Dest string, Distance int, TaxiInint, TaxiOutint, Cancelled int,
CancellationCode string, Diverted string, CarrierDelayint, WeatherDelayint,
NASDelayint, SecurityDelayint, LateAircraftDelayint) row format delimited fields
terminated by ',';
```

```
6.load data local inpath '/home/stud/Downloads/flight_data.csv' into table flight;
```

```
####for External Table:
```

7. Open another terminal and execute following commands to copy csv file to hdfs

```
hdfsdfs -mkdir /user/hive/flight
hdfsdfs -put /home/stud/Downloads/flight_ext.csv /user/hive/flight
hdfsdfs -ls /user/hive/flight
```

8. Go to hive terminal and create external table by specifying path:

```
create external table flight_ext(Year int, Month int, DayofMonthint, DayOfWeekint,
DepTime int, CRSDepTimeint, ArrTimeint, CRSArrTimeint, UniqueCarrier string,
FlightNumint, TailNum string, ActualElapsedTimeint, CRSElapsedTimeint, AirTimeint,
ArrDelayint, DepDelayint, Origin string, Dest string, Distance int, TaxiInint,
TaxiOutint, Cancelled int, CancellationCode string, Diverted string, CarrierDelayint,
WeatherDelayint, NASDelayint, SecurityDelayint, LateAircraftDelayint) row format
delimited fields terminated by ',' location '/user/hive/flight';
```

***Command to find the details of table in hive:

```
describe formatted table_name;
```

Count the number of rows in table:

```
select count(*) from flight_ext;
```

****Join tables:

```
create table flight_fare( FlightNumint, Fare int) row format delimited fields
terminated by ',';
```

```
load data local inpath '/home/stud/Downloads/flight_fare.csv' into table flight_fare;
```

```
selectflight_csv.Origin, flight_csv.Dest, flight_fare.fare from flight_csv join
flight_fare on flight_csv.FlightNum=flight_fare.FlightNum;
```


Group-B

Assignments based on Data Analytics using Python

Assignment 1

ProblemStatement

Perform the following operations using Python on the Facebook metrics data sets

- a. Create data subsets
- b. Merge Data
- c. Sort Data
- d. Transposing Data
- e. Shape and reshape Data

Theory

Python

It is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python Features

Python's features

- Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read – Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain – Python's source code is fairly easy-to-maintain.
- A broad standard library – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases – Python provides interfaces to all major commercial databases.
- GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems

Installation steps -

Python is available on a wide variety of platforms including Linux and Mac OS.

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

Unix and Linux Installation

Here are the simple steps to install Python on Unix/Linux machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.

- Follow the link to download zipped source code available for Unix/Linux.
- Download and extract files.
- Editing the *Modules/Setup* file if you want to customize some options.
- run `./configure` script
- `make`
- `make install`

This installs Python at standard location `/usr/local/bin` and its libraries at `/usr/local/lib/pythonXX` where XX is the version of Python.

Note- You can install Python on Windows or any other operating system.

Jupyter notebook:

With your virtual environment active, install Jupyter with the local instance of pip.

`pip install jupyter`

Run your notebook-

`jupyter notebook`

Dataset:

Download Facebook metrics data set.

Facebook Metrics

Data	Code (1)	Discussion (0)	Metadata	5	New Notebook	Download (15 kB)	
Detail	Compact	Column	10 of 19 columns				
# Page total likes	Type	# Category	# Post Month	# P			
0 total values	[null] 100%	500 total values	500 total values				
139441	Photo	2	12	2			
139441	Photo	2	12	2			
139441	Status	2	12	1			
139441	Photo	3	12	1			
139441	Photo	3	12	7			
139441	Status	2	12	7			
139441	Photo	3	12	6			
139441	Status	2	12	5			

Import require libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

- Read dataset and create dataframe in notebook.

Dataframe is a data structure that holds the data in the form of a matrix i.e. it contains the data in the value-form of rows and columns.

Create data subsets –

- Using Indexing operator

Syntax-

```
dataframe[['col1','col2','colN']]
subset1= df[['Type','Category']]
```

- Using loc() function

Syntax-

```
pandas.dataframe.loc[]
subset2=df.loc[0:3]
subset3= df.loc[0:2,['Type','Category']]
```

- Using iloc() function-

It enables us to create subset choosing specific values from rows and columns based on indexes.

Syntax-

```
pandas.dataframe.iloc[]
Subset4= df.iloc[[0,1,3,6],[0,2]]
```

Merge Data

Syntax-

```
m1= Pd.concat(df1,df2)
m2= df1.merge(df2)
m3= df1.merge(df2, on='Type')
m4= df1.merge(df2, left_on='Type',right_on= 'Category')
```

Sort Data –

Syntax-

```
st1=df.sort_values(by='Type')
```

Sort Dataframe rows based on a multiple columns.

```
st2=df.sort_values(by=['Type', 'Category'])
```

Sort Dataframe rows based on columns in Descending Order.

```
St3= df.sort_values(by =Type, ascending =False)
```

Sort columns of a Dataframe based on a single row.

Transposing Data

Reflect the DataFrame over its main diagonal by writing rows as columns and vice-versa.

```
tp=df.transpose()
```

Shape and reshape Data

```
Dataset.shape()
```

Melt: The `.melt()` function is used to reshape a DataFrame from a wide to a long format. It is useful to get a DataFrame where one or more columns are identifier variables, and the other columns are unpivoted to the row axis leaving only two non-identifier columns named `variable` and `value` by default.

Wide To Long: `.wide_to_long()` is another function that can help us transform the data from wide to long. This function is less flexible but more user-friendly than melt.

Pivot: The `.pivot()` method allows us to reshape the data from a long to a wide format. It returns a reshaped DataFrame organized by given unique index or column values.

Sample operation statements-

1. Create a data subset having 5 columns and 50 rows.(Type, Post Weekday, Post Hour,like, share)
2. Create a data subset having 4 columns and 25 rows. {Type, Total Interactions,like, share)
3. Create a subset having Post Hour>3hr.
4. Sort the subset 1 on like and share column in descending order.
5. Merge first two subset on Type and sort them on no of shares(share column)
6. Merge first two subsets on different left(like) and right columns(share).
7. Transpose first two subsets and sort them in descending order.
8. Show all dataframes in wide and long formats. Convert wide to long and vice versa.

Conclusion

Thus we have learnt different operations using Python on the Facebook metrics data sets

Assignment 2

ProblemStatement

Perform the following operations using Python on the Air quality and Heart Diseases data sets

- Data cleaning
- Data integration
- Data transformation
- Error correcting
- Data model building

Theory –

Download the datasets Air Quality and heart diseases available at kaggle.com.

Air_ quality dataset:

```
In [4]: df.describe()
```

Out[4]:

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3
count	24933.000000	18391.000000	25949.000000	25946.000000	25346.000000	19203.000000	27472.000000	25677.000000	25
mean	67.450578	118.127103	17.574730	28.560659	32.309123	23.483476	2.248598	14.531977	34
std	64.661449	90.605110	22.785846	24.474746	31.646011	25.684275	6.962884	18.133775	21
min	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.0
25%	28.820000	56.255000	5.630000	11.750000	12.820000	8.580000	0.510000	5.670000	18
50%	48.570000	95.680000	9.890000	21.690000	23.520000	15.850000	0.890000	9.160000	30
75%	80.590000	149.745000	19.950000	37.620000	40.127500	30.020000	1.450000	15.220000	45
max	949.990000	1000.000000	390.680000	362.210000	467.630000	352.890000	175.810000	193.860000	25

Data cleaning

Data cleaning means fixing bad data in your data set. Bad data could be:

- Empty cells
- Data in wrong format
- Wrong data
- Duplicates

When working with multiple data sources, there are many chances for data to be incorrect, duplicated, or mislabeled. If data is wrong, outcomes and algorithms are unreliable, even though they may look correct. *Data cleaning* is the process of changing or eliminating garbage, incorrect, duplicate, corrupted, or incomplete data in a dataset. There's no such absolute way to describe the precise steps in the data cleaning process because the processes may vary from dataset to dataset. Data cleansing, data cleansing, or data scrub is the initiative among the general data preparation process. Data cleaning plays an important part in developing reliable answers and within the analytical process and is observed to be a basic feature of the info science basics. The motive of data cleaning services is to construct uniform and standardized data sets that enable data analytical tools and business intelligence easy access and perceive accurate data for each problem.

Data cleaning is the most important task that should be done as a data science professional. Having wrong or bad quality data can be detrimental to processes and analysis. Having clean data will ultimately increase overall productivity and permit the very best quality information in your decision-making.

- Error-Free Data

- Data Quality
- Accurate and Efficient
- Complete Data
- Maintains Data Consistency

Data cleaning with Pandas-

Import pandas library and required dataset.

- Dropping columns-
- `df.drop(['NH3', 'PM10'], inplace=True, axis=1)`
- Find null values-
- `df.isnull()`

- `print df[PM10].isnull()`

- Cleaning / Filling Missing Data

- Pandas provides various methods for cleaning the missing values. The `fillna` function can “fill in” NA values with non-null data in a couple of ways, which we have illustrated in the following sections.

- `df.isna()---`5 cells NA
- `df.isnull()`

- `df.fillna(12.45)----`2 NA cloumn1

Drop Missing Values

If you want to simply exclude the missing values, then use the **dropna** function along with the **axis** argument. By default, `axis=0`, i.e., along row, which means that if any value within a row is NA then the whole row is excluded.

```
df.dropna()- 3 NA column 3
```

Convert Into a Correct Format

In our Data Frame, we have two cells with the wrong format.

Converting date to datetime-

```
df['Date'] = pd.to_datetime(df['Date'])
```

Converting datetime to date-

```
df['Datetime'] = pd.to_datetime(df['Datetime']).dt.date
```

Replacing Values

One way to fix wrong values is to replace them with something else.

```
df.loc[7, 'PM10'] = 45
```

```
Replace out of range values-  
for x in df.index:  
    if df.loc[x, "PM10"] < 100:  
        df.loc[x, "PM10"] = 100
```

Data Integration -

So far, we've made sure to remove the impurities in data and make it clean. Now, the next step is to combine data from different sources to get a unified structure with more meaningful and valuable information. This is mostly used if the data is segregated into different sources.

To do data integration, we can merge multiple pandas DataFrames using the merge function.

```
df = pd.merge(df1, df2, on = 'Id')
```

Data Transformation-

Now, we have a lot of columns that have different types of data. Our goal is to transform the data into a machine-learning-digestible format. All machine learning algorithms are based on mathematics. So, we need to convert all the columns into numerical format.

- Numerical: As the name suggests, this is numeric data that is quantifiable.
- Categorical: The data is a string or non-numeric data that is qualitative in nature.

Handling Categorical Data

There are some algorithms that can work well with categorical data, such as decision trees. But most machine learning algorithms cannot operate directly with categorical data. These algorithms require the input and output both to be in numerical form. If the output to be predicted is categorical, then after prediction we convert them back to categorical data from numerical data. Let's discuss some key challenges that we face while dealing with categorical data:

Encoding

To address the problems associated with categorical data, we can use encoding. This is the process by which we convert a categorical variable into a numerical form. Here, we will look at three simple methods of encoding categorical data.

Replacing

This is a technique in which we replace the categorical data with a number. This is a simple replacement and does not involve much logical processing. Let's look at an exercise to get a better idea of this.

Find the categorical column and separate it out with a different dataframe. To do so, use the `select_dtypes()` function from pandas:

```
df_categorical = df.select_dtypes(exclude=[np.number])
```

Find the distinct unique values in the Grade column. To do so, use the unique() function from pandas with the column name:

```
df_categorical[Type].unique()
```

Find the frequency distribution of each categorical column. To do so, use the value_counts() function on each column. This function returns the counts of unique values in an object:

```
df_categorical.Grade.value_counts()
```

For the Gender column, write the following code:

```
df_categorical.Gender.value_counts()
```

1. Replace the entries in the Gender column. Replace Male with 0 and Female with 1. To do so, use the replace() function:

```
df_categorical.Gender.replace({"Male":0,"Female":1}, inplace= True)
```

2. Error Correction

There are many reasons such as noise, cross-talk etc., which may help data to get corrupted during transmission. Most of the applications would not function expectedly if they receive erroneous data. Thus error correction is important to do before any analysis.

- Gauge min and max values: For continuous variables, checking the minimum and maximum values for each column can give you a quick idea of whether your values are falling within the correct range.
- Look for missing values: The easiest way to find missings is to perform a count or sorting your columns. It helps in finding missing values which can be replaced/removed to get expected analysis.

Model Building :

In this phase, the data science team needs to develop data sets for training, testing, and production purposes.

These data sets enable data scientists to develop analytical methods and train it, while holding aside some data for testing the model.

- Normalization
- Simple and Multiple Linear Regression
- Model Evaluation Using Visualization
- Polynomial Regression and Pipelines
- R-squared and MSE for In-Sample Evaluation
- Prediction and Decision Making

#Grouping the AQI by city and calculating the average AQI per city

```
x=pd.DataFrame(df.groupby(['City'])['AQI'].mean().sort_values(by='AQI',ascending=False).head(10))  
x=x.reset_index('City')
```



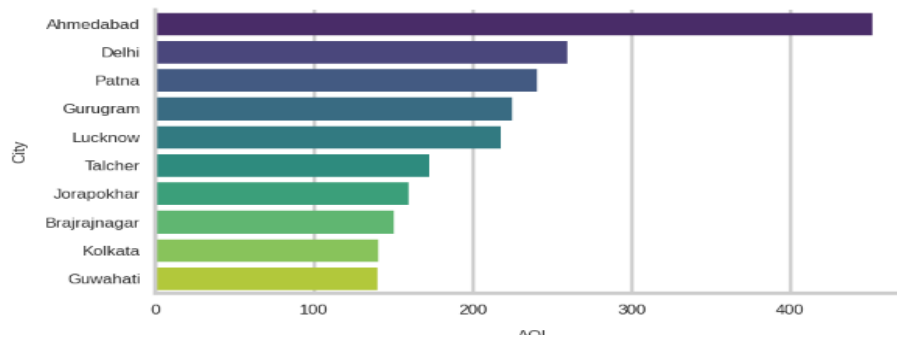
```
plt.style.use('seaborn-whitegrid')
```

```
plt.figure(figsize=(3,1.5))
```

```
sns.barplot(data=x,x='AQI',y='City',orient='h',palette='viridis')
```

```
#plotting the average AQI per city
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(3,1.5))
sns.barplot(data=x,x='AQI',y='City',orient='h',palette='viridis')

Out[7]:
<AxesSubplot:xlabel='AQI', ylabel='City'>
```



```
from sklearn import linear_model
```

```
X = dff['PM2.5'].values.reshape(-1,1)
```

```
y = dff['AQI'].values
```

```
ols = linear_model.LinearRegression()
```

```
model = ols.fit(X, y)
```

```
response = model.predict(X)
```

```
r2 = model.score(X, y)
```

```
plt.style.use('default')
```

```
plt.style.use('ggplot')
```

```
fig, ax = plt.subplots(figsize=(8, 4))
```

```
ax.plot(X, response, color='k', label='Regression model')
```

```
ax.scatter(X, y, edgecolor='k', facecolor='grey', alpha=0.7, label='Data-Jan')
```

```
ax.set_ylabel('AQI', fontsize=14)
```

```
ax.set_xlabel('PM2.5', fontsize=14)
```

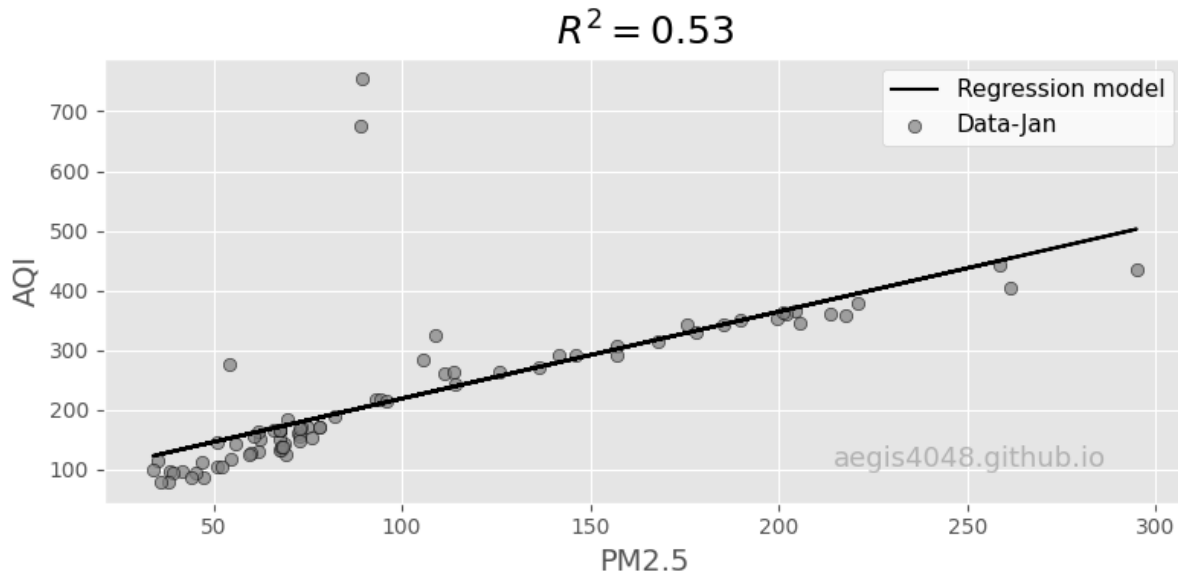
```
ax.text(0.8, 0.1, 'aegis4048.github.io', fontsize=13, ha='center', va='center', transform=ax.transAxes, color='grey')
```

```
alpha=0.5)
```

```
ax.legend(facecolor='white', fontsize=11)
```

```
ax.set_title('$R^2= %.2f$' % r2, fontsize=18)
```

```
fig.tight_layout()
```



Conclusion:

Thus we have learnt different operations using Python on the Airquality data sets

Assignment 4

ProblemStatement

Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3 .

Theory

Visualization:

It may sometimes seem easier to go through a set of data points and build insights from it but usually this process may not yield good results. There could be a lot of things left undiscovered as a result of this process. Additionally, most of the data sets used in real life are too big to do any analysis manually.

Data visualization is an easier way of presenting the data, however complex it is, to analyze trends and relationships amongst variables with the help of pictorial representation.

The following are the advantages of Data Visualization

- Easier representation of compels data
- Highlights good and bad performing areas
- Explores relationship between data points
- Identifies data patterns even for larger data points

Visualization should have:

- Appropriate usage of shapes, colors, and size while building visualization
- Plots/graphs using a co-ordinate system are more pronounced
- Knowledge of suitable plot with respect to the data types brings more clarity to the information
- Usage of labels, titles, legends and pointers passes seamless information the wider audience

Visualization libraries in python:

There are a lot of python libraries which could be used to build visualization like *matplotlib*, *vispy*, *bokeh*, *seaborn*, *pygal*, *folium*, *plotly*, *cufflinks*, and *networkx*. Of the many, *matplotlib* and *seaborn* seems to be very widely used for basic to intermediate level of visualizations.

1.Matplotlib

It is an library in Python for 2D plots of arrays, It is a multi-platform data visualization library built on *NumPy* arrays and designed to work with the broader *SciPy* stack.

It is well maintained visualization output with high quality graphics draws a lot of users to it. Basic as well as advanced charts could be very easily built from the users/developers point of view, since it has a large community support, resolving issues and debugging becomes much easier.

2.Seaborn

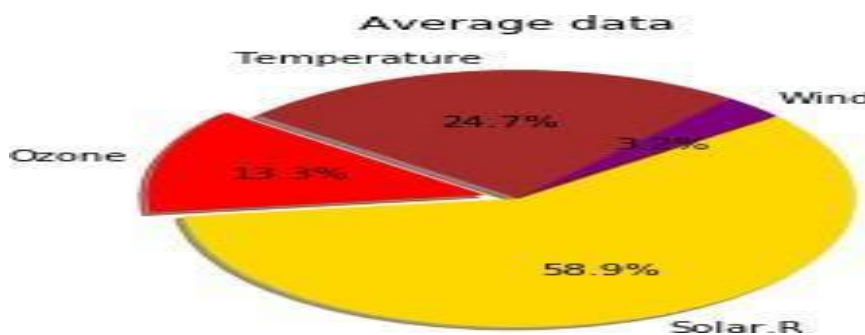
This library sits on top of *matplotlib*.Means, it has some flavors of *matplotlib* while from the visualization point, its is much better than *matplotlib* and has added features as well.

Benefits:

- Built-in themes aid better visualization
- Statistical functions aiding better data insights
- Better aesthetics and built-in plots
- Helpful documentation with effective examples

1. PIE CHART

```
In [50]: import matplotlib.pyplot as plt
In [51]: import pandas as pd
In [52]: data = pd.read_csv("airquality.csv")
In [53]: labels = 'Ozone','Solar.R','Wind','Temperature'
In [54]: sizes = [data['Ozone'].mean(),data['Solar.R'].mean(),data['Wind'].mean(),data['Temp'].mean()]
In [55]: colors = ['red','gold','purple','brown']
In [56]: explode = (0.1, 0, 0, 0)
In [57]: plt.pie(sizes, explode=explode, labels=labels, colors=colors,
...: autopct='%1.1f%%', shadow=True, startangle=140)
...: plt.title('Average data')
Out[57]: Text(0.5, 1.0, 'Average data')
In [58]: plt.savefig('plot1.png')
<Figure size 432x288 with 0 Axes>
```



2. BAR PLOT

```
import matplotlib.pyplot as plt
import numpy as np

import pandas as pd

data = pd.read_csv("airquality.csv")
h = data.iloc[1:21,4]

y_pos = np.arange(len(h))
v = range(1,21)

plt.bar(y_pos,h,align = 'center', alpha = 0.5)

plt.xticks(y_pos,v)
```

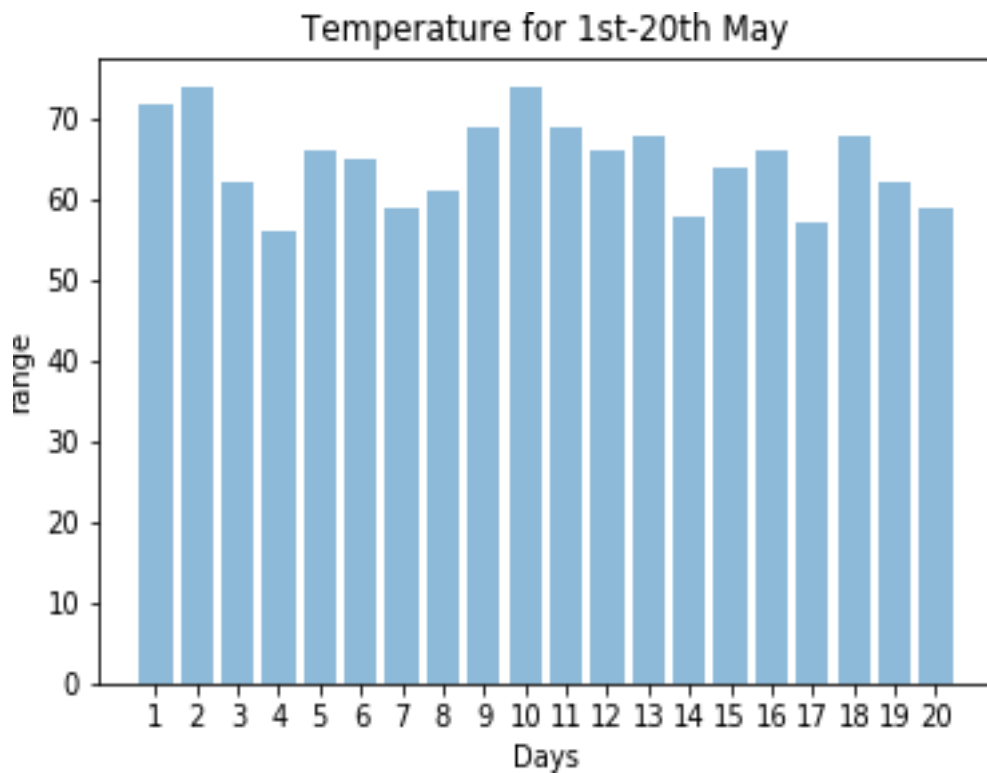
```
plt.ylabel('range')

plt.xlabel("Days")

plt.title('Temperature for 1st-20th May')

plt.savefig('plot2.png')

plt.show()
```



3.BOXPLOT

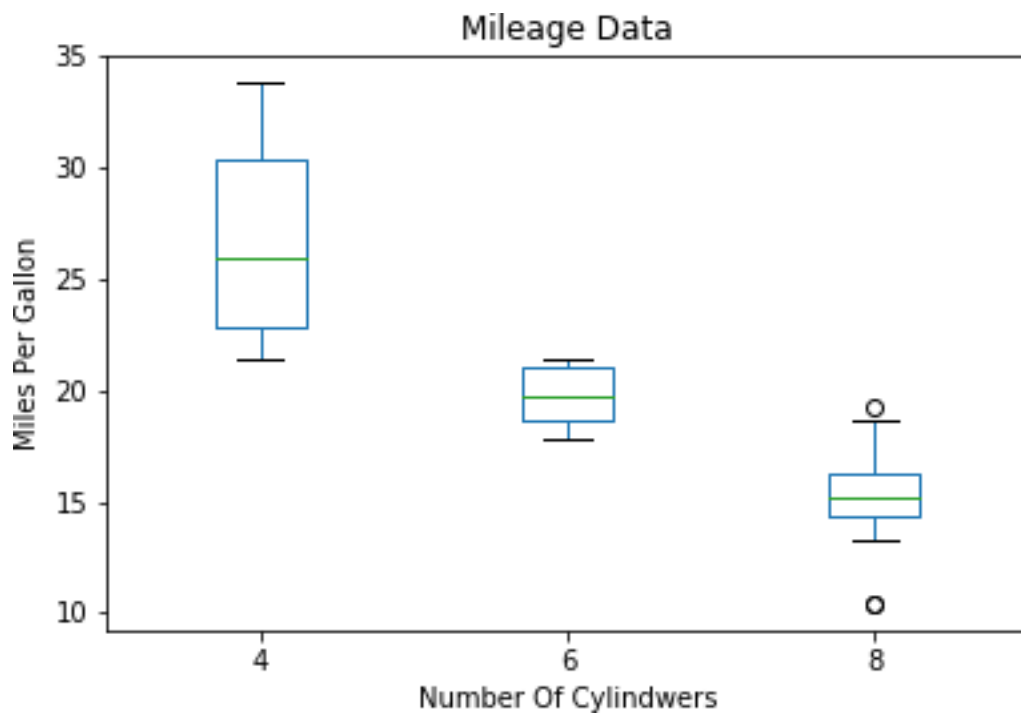
```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv("mtcars.csv")
data.head()

data.boxplot(by = 'cyl',column = ['mpg'],grid = False)
plt.ylabel("Miles Per Gallon")

plt.xlabel("Number Of Cylinders")
plt.title("Mileage Data")
plt.suptitle("")
```

```
plt.savefig('plot3.png')
```



4.HISTOGRAM

```
import matplotlib.pyplot as plt import pandas as
```

```
pd
```

```
data = pd.read_csv("mtcars.csv") h = data.iloc[:,-1]
```

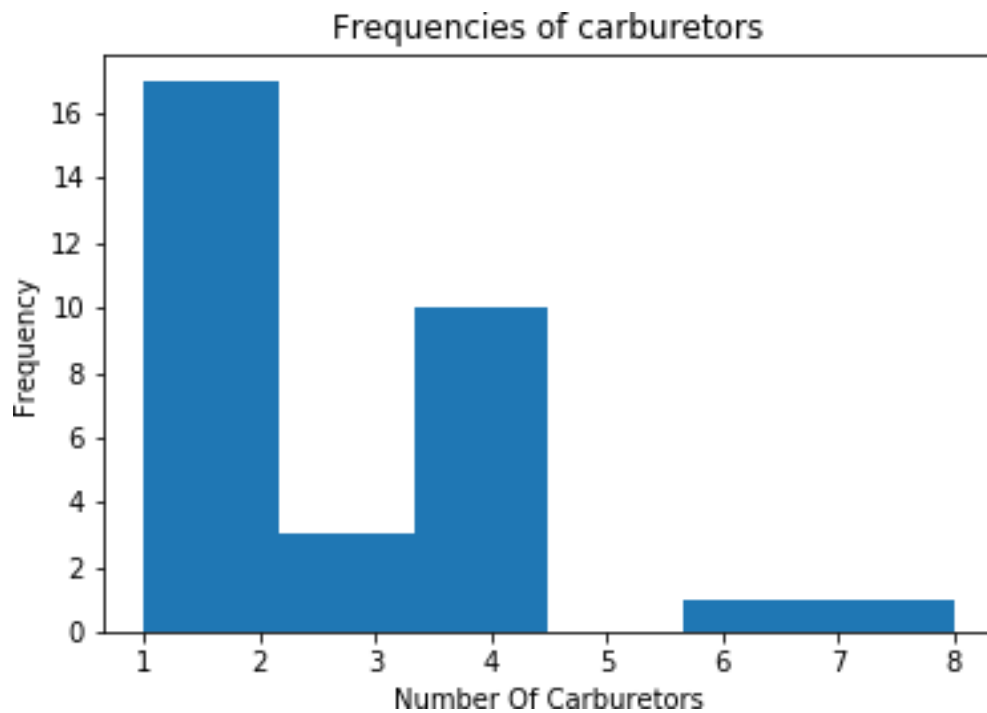
```
plt.hist(h, bins= 'auto') plt.title("Frequencies of
```

```
carburetors") plt.ylabel("Frequency")
```

```
plt.xlabel("Number Of Carburetors")
```

```
plt.savefig("plot4.png")
```

```
plt.show()
```



5.LINE GRAPH

import matplotlib.pyplot as plt import pandas as

pd

data = pd.read_csv("airquality.csv")

data["Ozone"].fillna(data['Ozone'].median(),inplace = True) h =

data.iloc[1:16,1]

v = data.iloc[1:16,3]

plt.plot(h,label = 'ozone',marker = 'o',linestyle = "dotted") plt.plot(v,label =

'wind',marker = 'o',linestyle = "dashed") plt.ylim(0,60)

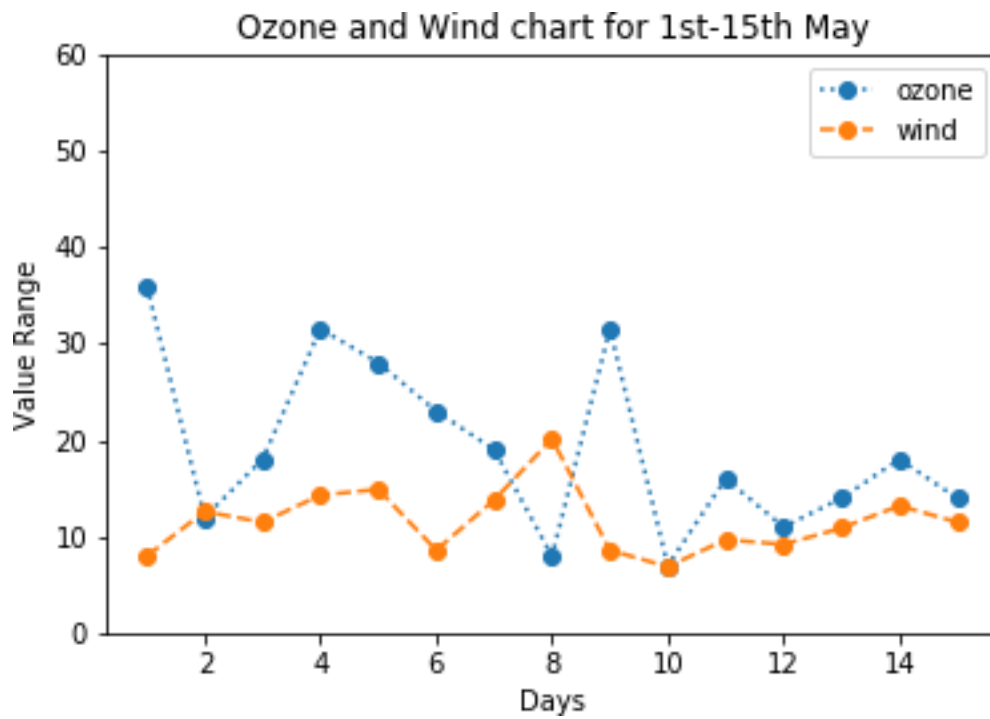
plt.legend()

plt.title("Ozone and Wind chart for 1st-15th May") plt.ylabel("Value

Range")

plt.xlabel("Days") plt.savefig("plot5.png")

plt.show()



6. SCATTER PLOT

```
import matplotlib.pyplot as plt
import pandas as
```

```
pd
```

```
data = pd.read_csv("forestfires.csv")
h =
```

```
data.iloc[1:91,8]
```

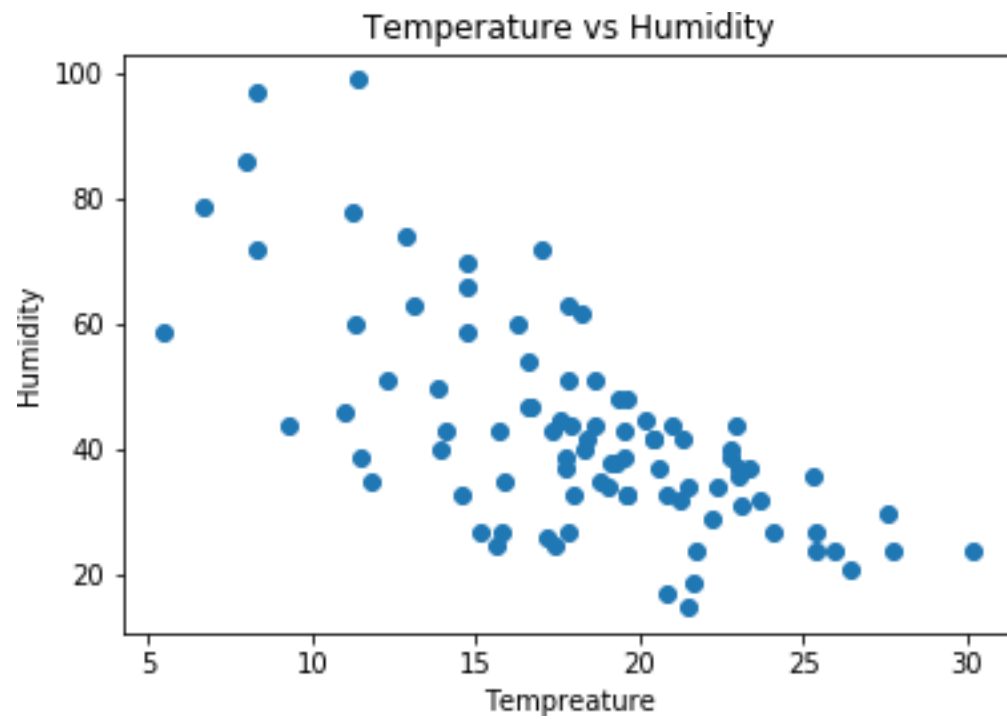
```
v = data.iloc[1:91,9]
plt.scatter(h,v)
```

```
plt.title("Temperature vs Humidity")
```

```
plt.xlabel("Temperature")
plt.ylabel("Humidity")
```

```
plt.savefig("plot6.png")
```

```
plt.show()
```

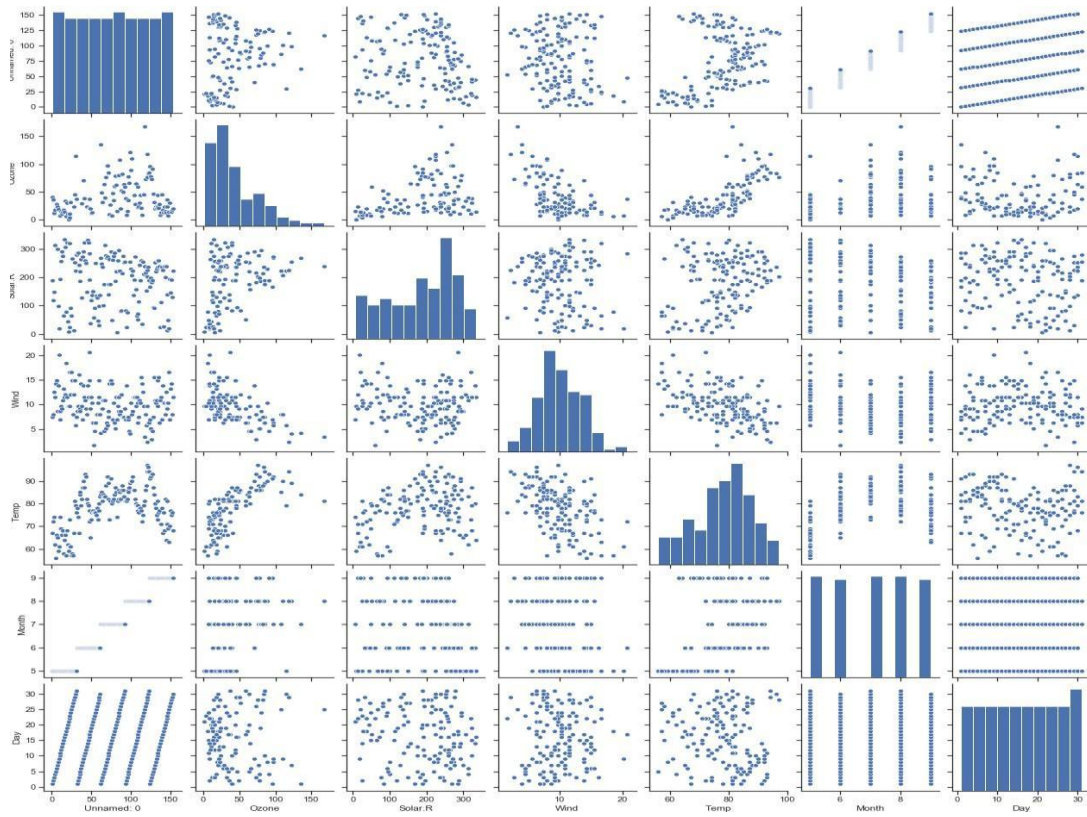



```
import seaborn as sns
import pandas as pd

import matplotlib.pyplot as plt
data = pd.read_csv("airquality.csv")
sns.set(style = "ticks")

sns.pairplot(data)
plt.savefig("plot7.png")

plt.show()
```



7.HEAT MAP

```
import matplotlib.pyplot as plt
import seaborn
```

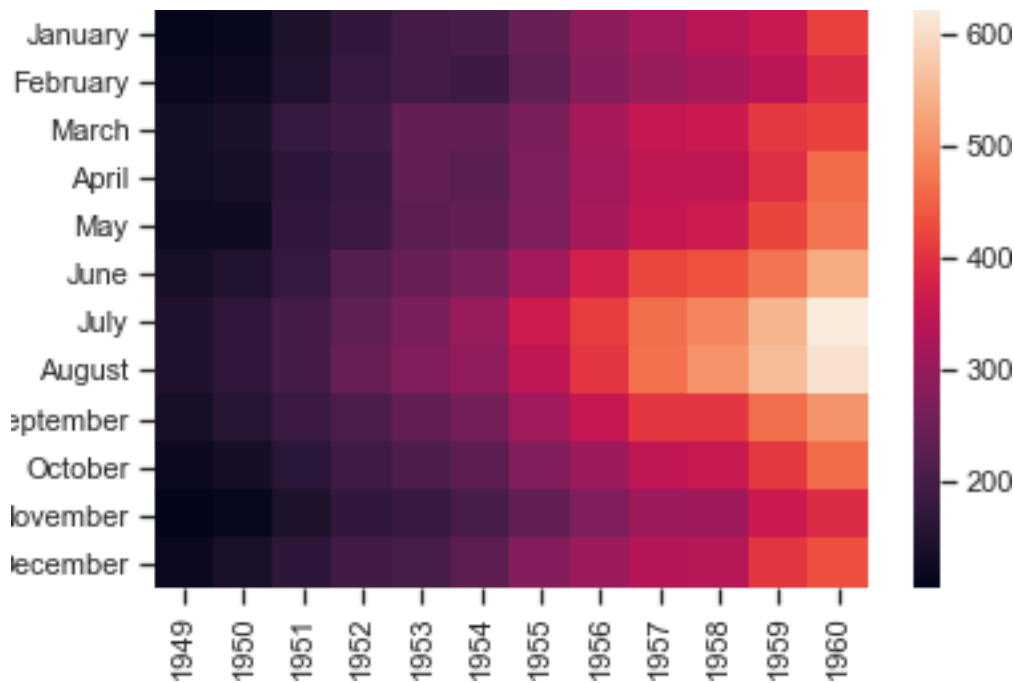
```
as sb
```

```
flights = sb.load_dataset("flights")
```

```
flights = flights.pivot("month", "year", "passengers") ax =
```

```
sb.heatmap(flights)
```

```
plt.savefig("plot8.png")
```



8.WORDCLOUD

```
import matplotlib.pyplot as pPlot

from wordcloud import WordCloud, STOPWORDS import numpy as

np

from PIL import Image

dataset = open("sampleWords.txt", "r").read() def

create_word_cloud(string):

    cloud = WordCloud(background_color = "white", max_words = 200, stopwords = set(STOPWORDS))

    cloud.generate(string)

    cloud.to_file("wordCloud.png") dataset =

dataset.lower() create_word_cloud(dataset)
```

level oriented late english institute
mathematics high
python
van scripting
readable computer fewer syntactical
language
use now guido rosum
interactive designed national highly punctuation derived
interpreted eighties
frequently
algorithms
constructing
constructions
early use
modules
netherlands
including
unix
science
research
object
keywords
nineties

ProblemStatement

Perform the following data visualization operations using Tableau on Adult and Iris datasets

1. 1D (Linear) Data visualization
2. 2D (Planar) Data Visualization
3. 3D (Volumetric) Data Visualization
4. Temporal Data Visualization
5. Multidimensional Data Visualization
6. Tree/ Hierarchical Data visualization
7. Network Data visualization

Theory

Introduction

Data visualization or data visualization is viewed by many disciplines as a modern equivalent of visual communication. It involves the creation and study of the visual representation of data, meaning "information that has been abstracted in some schematic form, including attributes or variables for the units of information".

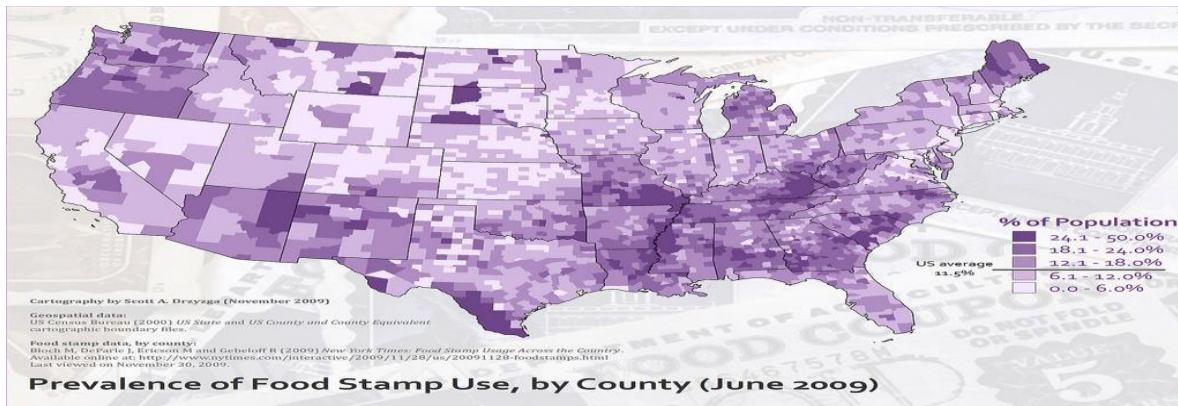
Data visualization refers to the techniques used to communicate data or information by encoding it as visual objects (e.g., points, lines or bars) contained in graphics. The goal is to communicate information clearly and efficiently to users. It is one of the steps in data analysis or data science

Examples:

- lists of data items, organized by a single feature (e.g., alphabetical order) (not commonly visualized)

Examples (geospatial):

- choropleth



3D/Volumetric

Broadly, examples of scientific visualization:

- 3D computer models

In 3D computer graphics, **3D modeling** (or **three-dimensional modeling**) is the process of developing a mathematical representation of any surface of an object (either inanimate or living) in three dimensions via specialized software. The product is called a **3D model**. Someone who works with 3D models may be referred to as a **3D artist**. It can be displayed as a two-dimensional image through a process called 3D rendering or used in a computer simulation of physical phenomena. The model can also be physically created using 3D printing devices.

- surface and volume rendering

Rendering is the process of generating an image from a model, by means of computer programs. The model is a description of three-dimensional objects in a strictly defined language or data structure. It would contain geometry, viewpoint, texture, lighting, and shading information. The image is a digital image or raster graphics image. The term may be by analogy with an "artist's rendering" of a scene. 'Rendering' is also used to describe the process of calculating effects in a video editing file to produce final video output.

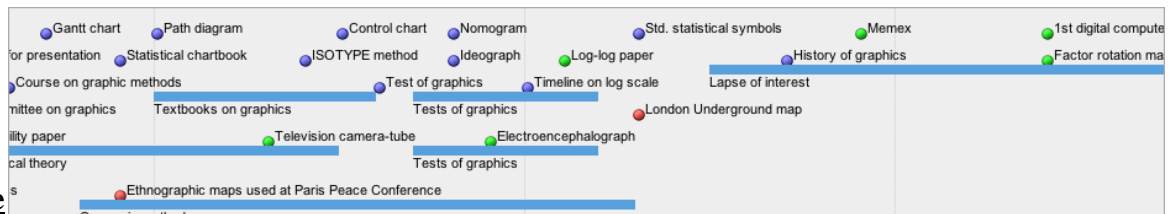
Volume rendering is a technique used to display a 2D projection of a 3D discretely sampled data set. A typical 3D data set is a group of 2D slice images acquired by a CT or MRI scanner. Usually these are acquired in a regular pattern (e.g., one slice every millimeter) and usually have a regular number of image pixels in a regular pattern. This is an example of a regular volumetric grid, with each volume element, or voxel represented by a single value that is obtained by sampling the immediate area surrounding the voxel.

- computer simulations

Computer simulation is a computer program, or network of computers, that attempts to simulate an abstract model of a particular system. Computer simulations have become a useful part of mathematical modeling of many natural systems in physics, and computational physics, chemistry and biology; human systems in economics, psychology, and social science; and in the process of engineering and new technology, to gain insight into the operation of those systems, or to observe their behavior.^[6] The simultaneous visualization and simulation of a system is called visulation.

Examples:

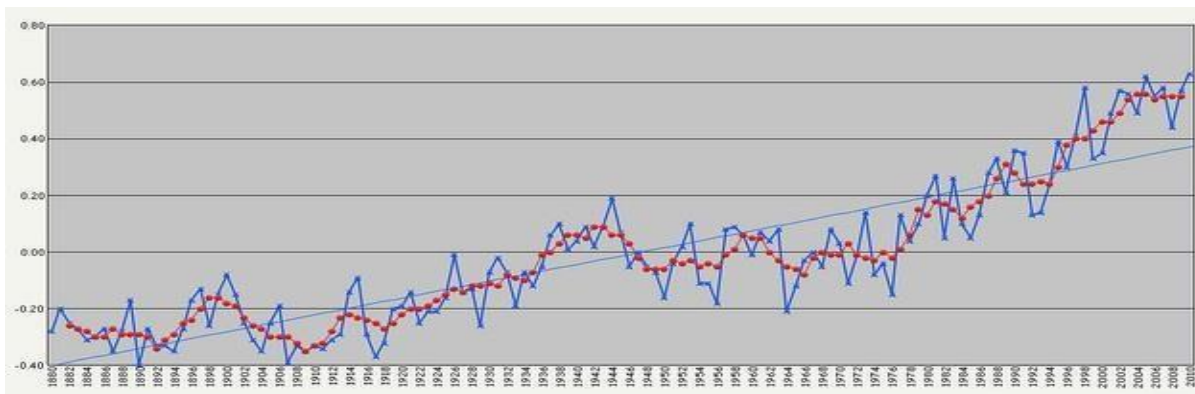
- **timeline**



Tools: SIMILE Timeline, TimeFlow, Timeline JS, Excel Image:

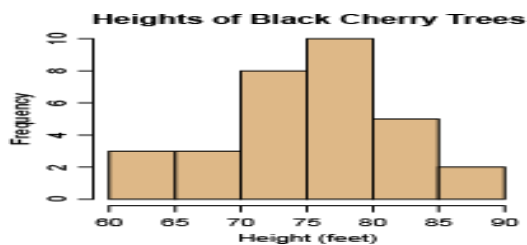
Friendly, M. & Denis, D. J. (2001). Milestones in the history of thematic cartography, statistical graphics, and data visualization. Web document, <http://www.datavis.ca/milestones/>. Accessed: August 30, 2012.

- **time series**

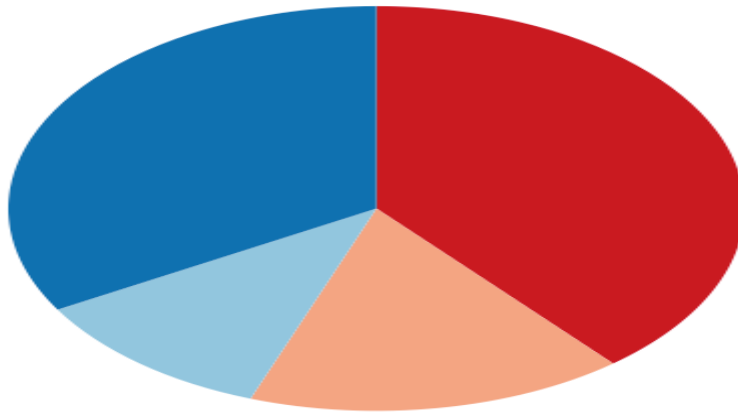


Examples (category proportions, counts):

- **histogram**



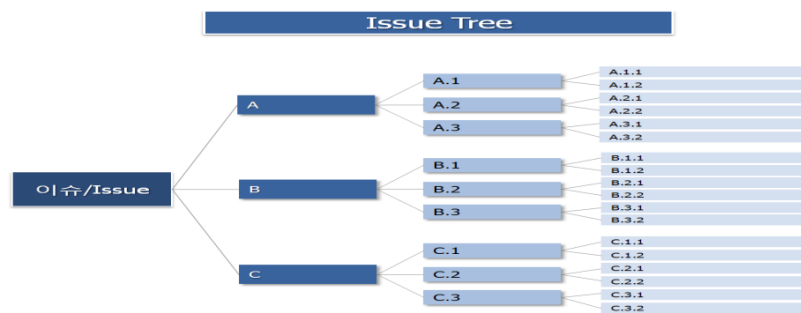
- **pie chart**



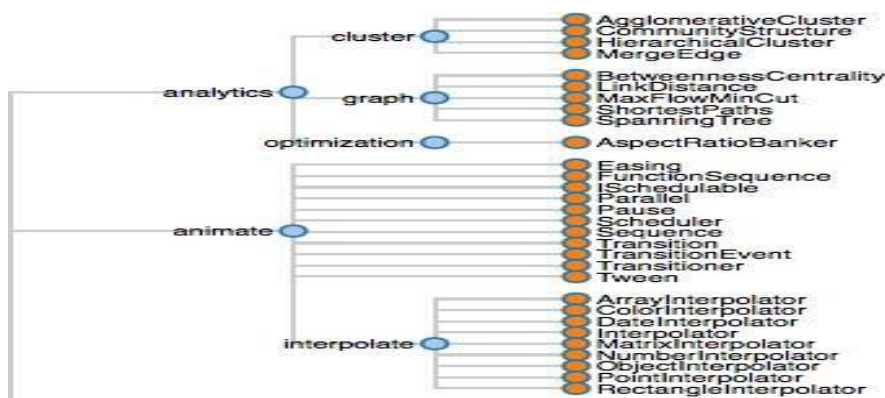
Colors from www.ColorBrewer.org by
Cynthia A. Brewer, Penn State Univ.

Examples:

- general tree visualization

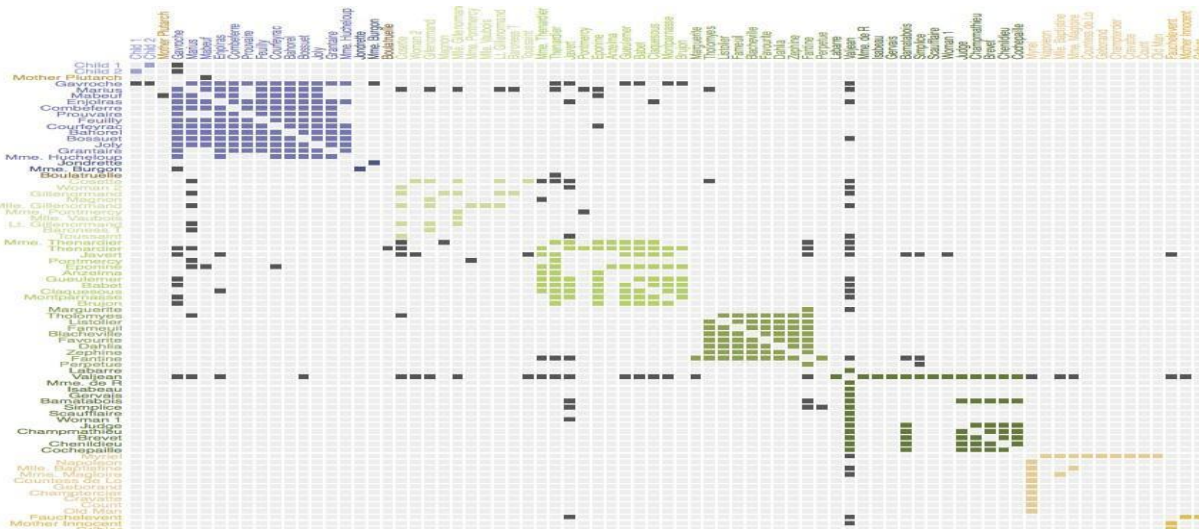


- dendrogram



Examples:

- matrix



- **node-link diagram (link-based layout algorithm)**



Tableau:

Tableau is a Business Intelligence tool for visually analyzing the data. Users can create and distribute an interactive and shareable dashboard, which depict the trends, variations, and density of the data in the form of graphs and charts. Tableau can connect to files, relational and Big Data sources to acquire and process data. The software allows data blending and real-time collaboration, which makes it very unique. It is used by businesses, academic researchers, and many government organizations for visual data analysis. It is also positioned as a leader Business Intelligence and Analytics Platform in Gartner Magic Quadrant.

Tableau Features:

Tableau provides solutions for all kinds of industries, departments, and data environments. Following are some unique features which enable Tableau to handle diverse scenarios.

- **Speed of Analysis** – As it does not require high level of programming expertise, any user with access to data can start using it to derive value from the data.

- **Self-Reliant** – Tableau does not need a complex software setup. The desktop version which is used by most users is easily installed and contains all the features needed to start and complete data analysis.
- **Visual Discovery** – The user explores and analyzes the data by using visual tools like colors, trend lines, charts, and graphs. There is very little script to be written as nearly everything is done by drag and drop.
- **Blend Diverse Data Sets** – Tableau allows you to blend different relational, semi structured and raw data sources in real time, without expensive up-front integration costs. The users don't need to know the details of how data is stored.
- **Architecture Agnostic** – Tableau works in all kinds of devices where data flows. Hence, the user need not worry about specific hardware or software requirements to use Tableau.
- **Real-Time Collaboration** – Tableau can filter, sort, and discuss data on the fly and embed a live dashboard in portals like SharePoint site or Salesforce. You can save your view of data and allow colleagues to subscribe to your interactive dashboards so they see the very latest data just by refreshing their web browser.
- **Centralized Data** – Tableau server provides a centralized location to manage all of the organization's published data sources. You can delete, change permissions, add tags, and manage schedules in one convenient location. It's easy to schedule extract refreshes and manage them in the data server. Administrators can centrally define a schedule for extracts on the server for both incremental and full refreshes.

There are three basic steps involved in creating any Tableau data analysis report.

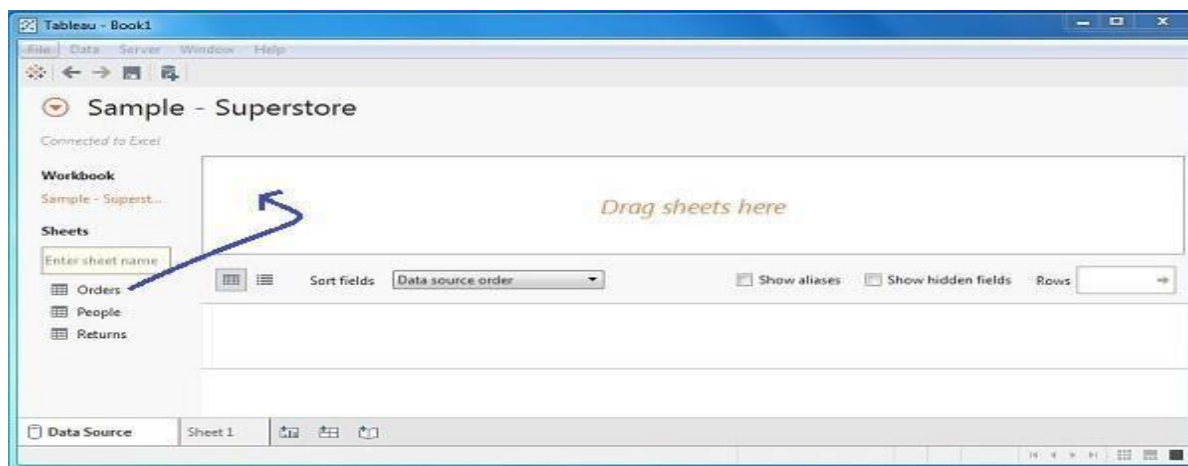
These three steps are –

- **Connect to a data source** – It involves locating the data and using an appropriate type of connection to read the data.
- **Choose dimensions and measures** – This involves selecting the required columns from the source data for analysis.
- **Apply visualization technique** – This involves applying required visualization methods, such as a specific chart or graph type to the data being analyzed.

For convenience, let's use the sample data set that comes with Tableau installation named sample – superstore.xls. Locate the installation folder of Tableau and go to **My Tableau Repository**. Under it, you will find the above file at **Datasources\9.2\en_US-US**.

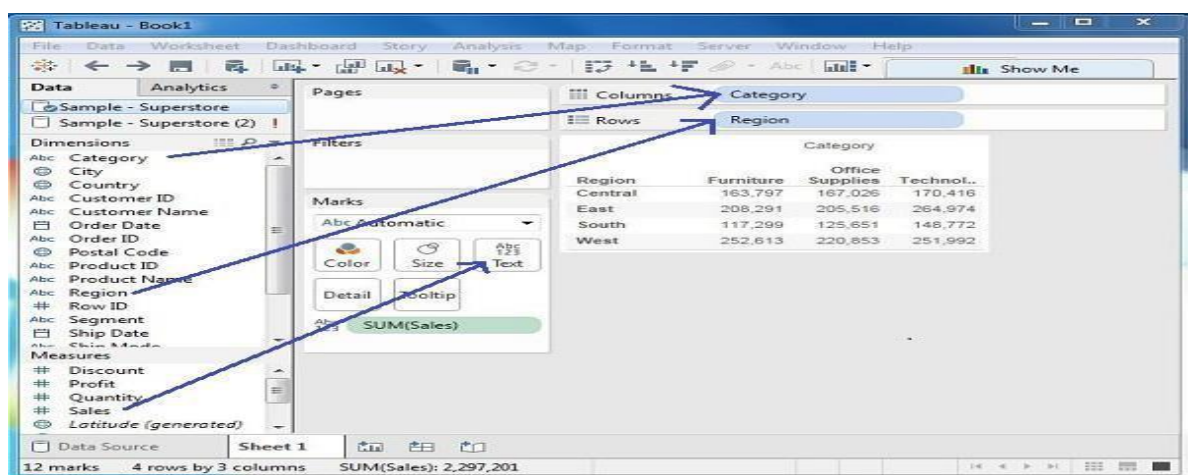
Connect to a Data Source

On opening Tableau, you will get the start page showing various data sources. Under the header “**Connect**”, you have options to choose a file or server or saved data source. Under Files, choose excel. Then navigate to the file “**Sample – Superstore.xls**” as mentioned above. The excel file has three sheets named Orders, People and Returns. Choose **Orders**.



Choose the Dimensions and Measures

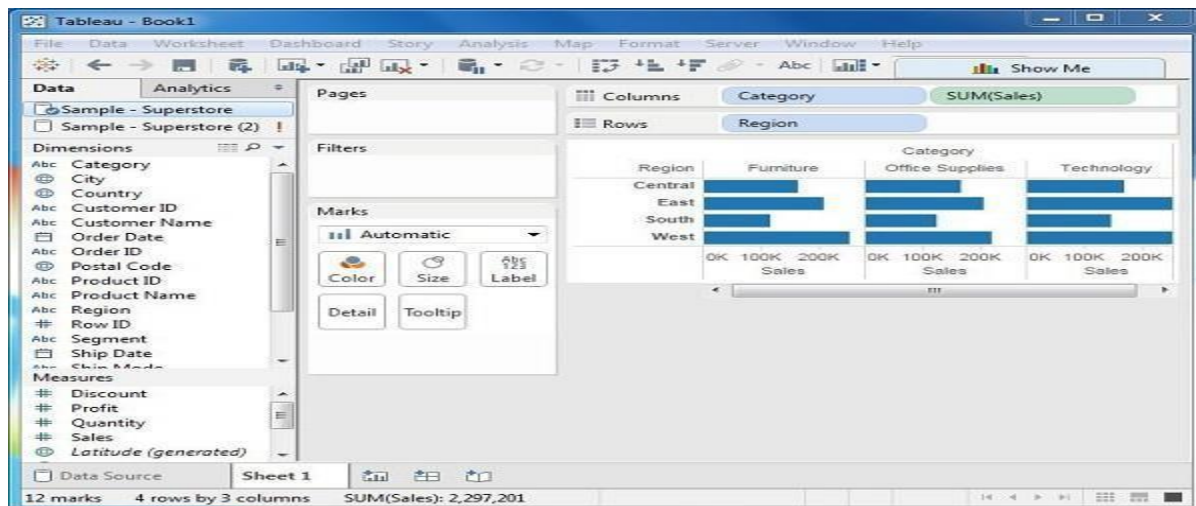
Next, choose the data to be analyzed by deciding on the dimensions and measures. Dimensions are the descriptive data while measures are numeric data. When put together, they help visualize the performance of the dimensional data with respect to the data which are measures. Choose **Category** and **Region** as the dimensions and **Sales** as the measure. Drag and drop them as shown in the following screenshot. The result shows the total sales in each category for each region.



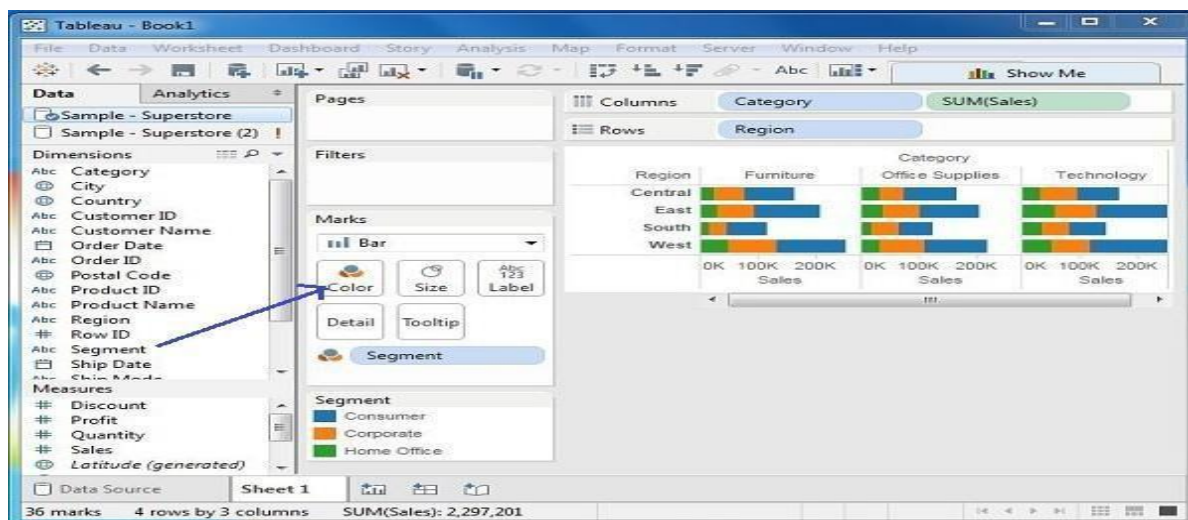
Apply Visualization Technique

In the previous step, you can see that the data is available only as numbers. You have to read and calculate each of the values to judge the performance. However, you can see them as graphs or charts with different colors to make a quicker judgment.

We drag and drop the sum (sales) column from the Marks tab to the Columns shelf. The table showing the numeric values of sales now turns into a bar chart automatically.



You can apply a technique of adding another dimension to the existing data. This will add more colors to the existing bar chart as shown in the following screenshot.



Conclusion: Thus we have learnt how to Visualize the data in different types (1D (Linear) Data visualization, 2D (Planar) Data Visualization, 3D (Volumetric) Data Visualization, Temporal Data Visualization, Multidimensional Data Visualization, Tree/ Hierarchical Data visualization, Network Data visualization) by using Tableau Software.

Group C: Model Implementation

Model Implementation Assignment 1

Create a review scraper for any ecommerce website to fetch real time comments, reviews, ratings, comment tags, customer name using Python.

Web Scrapping:

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API's that allow you to access their data in a structured format.

Web Scrapers can extract all the data on particular sites or the specific data that a user wants. Ideally, it's best if you specify the data you want so that the web scraper only extracts that data quickly. For example, you might want to scrape an Amazon page for the types of juicers available, but you might only want the data about the models of different juicers and not the customer reviews.

So, when a web scraper needs to scrape a site, first the URLs are provided. Then it loads all the HTML code for those sites and a more advanced scraper might even extract all the CSS and Javascript elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Mostly, this is in the form of an Excel spreadsheet or a CSV file, but the data can also be saved in other formats, such as a JSON file.

Types:

- **Self-built Web Scrapers**
- **Browser extensions Web Scrapers**
- **Software Web Scrapers**
- **Cloud Web Scrapers**
- **Local Web Scrapers**

Web scraping requires two parts, namely the **crawler** and the **scraper**. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website.

Python for web scraping:

It is the most popular language for web scraping as it can handle most of the processes easily. It also has a variety of libraries that were created specifically for Web Scraping. **Scrapy** is a very popular open-source web crawling framework that is written in Python. It is ideal for web scraping as well as extracting data using APIs. **Beautiful soup** is another Python library that is highly suitable for Web Scraping. It creates a parse tree that can be used to extract data from HTML on a website. Beautiful soup also has multiple features for navigation, searching, and modifying these parse trees.

Sample Code:

Below code scraps the customer name, Similarly implement code for scrapping real time comments, reviews, ratings, comment tags

```

# import module
import requests
from bs4 import BeautifulSoup

HEADERS = ({'User-Agent':
            'Mozilla/5.0 (Windows NT 10.0; Win64; x64) \
            AppleWebKit/537.36 (KHTML, like Gecko) \
            Chrome/90.0.4430.212 Safari/537.36',
            'Accept-Language': 'en-US, en;q=0.5'})

# Scrape the data
def getdata(site_url):
    res = requests.get(url, headers=HEADERS)
    return res.text

def gethtml(site_url):

    # pass the url
    # into getdata function
    data = getdata(site_url)
    soup = BeautifulSoup(data, 'html.parser')

    # display html code
    return (soup)

site_url = "https://www.amazon.in/Columbia-Mens-wind-\
resistant-Glove/dp/B0772WVHPS/?_encoding=UTF8&pd_rd\
_w=d9RS9&pf_rd_p=3d2ae0df-d986-4d1d-8c95-aa25d2ade606&pf\
_rd_r=7MP3ZDYBBV88PYJ7KEMJ&pd_rd_r=550bec4d-5268-41d5-\
87cb-8af40554a01e&pd_rd_wg=oy8v8&ref_=pd_gw_cr_cartx&th=1"

soup = gethtml(site_url)
#print(soup)

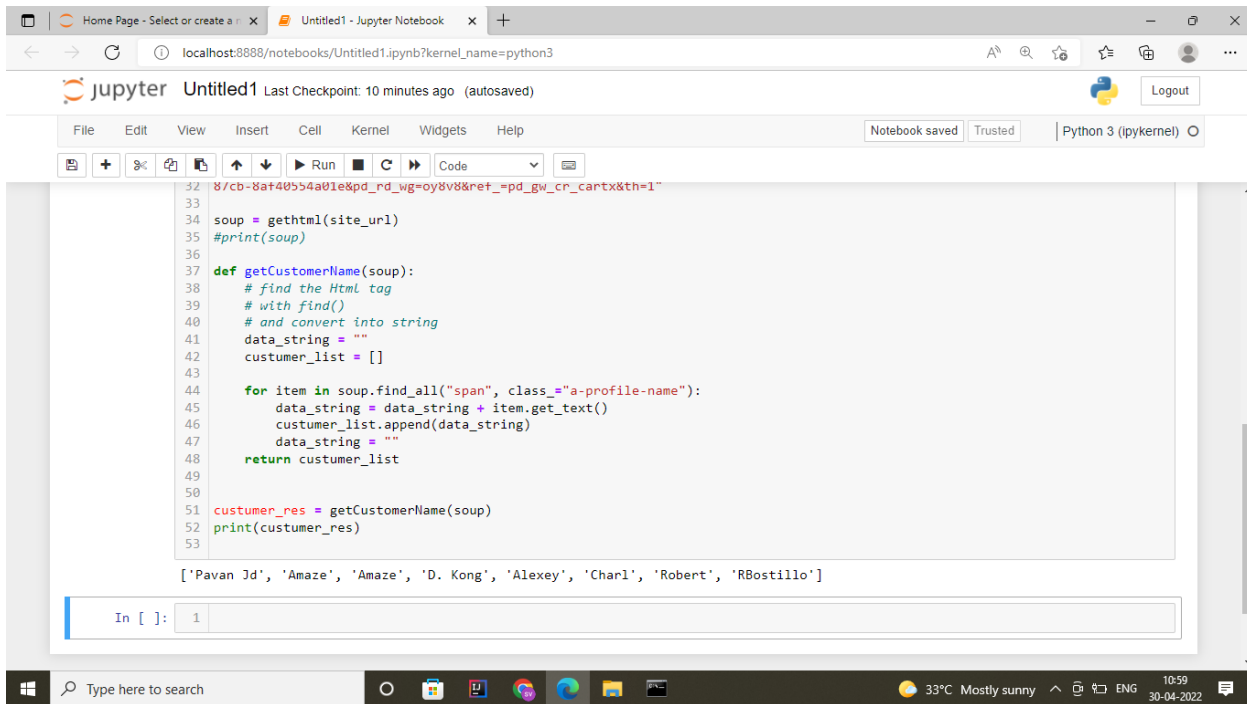
def getCustomerName(soup):
    # find the Html tag
    # with find()
    # and convert into string
    data_string = ""
    customer_list = []

    for item in soup.find_all("span", class_="a-profile-name"):
        data_string = data_string + item.get_text()
        customer_list.append(data_string)
        data_string = ""
    return customer_list

```



```
customer_res = getCustomerName(soup)
print(customer_res)
```



The screenshot shows a Jupyter Notebook titled 'Untitled1' running on a local host. The notebook contains a Python script that uses BeautifulSoup to scrape customer names from a website. The script defines a function `getCustomerName` that takes a `soup` object and returns a list of customer names. The output of the script is displayed as a list: `['Pavan Jd', 'Amaze', 'Amaze', 'D. Kong', 'Alexey', 'Charl', 'Robert', 'RBostillo']`.

```
32 8/cb-8af40554a01e&pd_rd_wg=oy8v8&ref_=pd_gw_cr_crtx&th=1"
33
34 soup = gethtml(site_url)
35 #print(soup)
36
37 def getCustomerName(soup):
38     # find the Html tag
39     # with find()
40     # and convert into string
41     data_string = ""
42     customer_list = []
43
44     for item in soup.find_all("span", class_="a-profile-name"):
45         data_string = data_string + item.get_text()
46         customer_list.append(data_string)
47         data_string = ""
48     return customer_list
49
50
51 customer_res = getCustomerName(soup)
52 print(customer_res)
53
```

['Pavan Jd', 'Amaze', 'Amaze', 'D. Kong', 'Alexey', 'Charl', 'Robert', 'RBostillo']

Conclusion: Thus we learnt how web scrapping to fetch real time comments, reviews, ratings, comment tags, customer name using Python.





