Market Data Analysis

Abstract:

This project presents how marketing concepts and tools may be applied in investment appraisal studies. The marketing evaluation process begins with a description of the project concept based on the market need the project aims to satisfy.This aids the definition of the project's relevant market and leads to an analysis of the market. The market consists of customer and competing suppliers.

The project must try and match its potential capabilities to existing and potential customer needs. In doing this, the project gains competitive edge and maximizes potential performance.Market performance is a measure of the project's ability to satisfy the key market need factors within its defined target market. It shows how a project analyst may evaluate a project's market performance.

Such a measure may be used as an indicator of competitiveness by which to project market expansion and market share estimates. In economic analysis a market expansion is an outward shift in the demand curve, and takes place when a project achieves a competitiveness rating higher than the level of other market competitors.

1.INTRODUCTION

The challenge for any business is to gain a sufficiently detailed understanding of the fundamentals of a market. Without this insight, it is unlikely that marketing strategies will prove effective or that marketing objectives will be met. Gaining this understanding is the task of market analysis.

By market fundamentals we mean understanding issues such as:

- How big is the market? (for example, measured by sales value, volume sold etc.)
- How fast is the market growing and what is the market growth potential?
- The key social, economic, political/legal and technological factors that drive change in the market
- Who are the existing competitors and what market shares do they have?
- The extent of branding and customer loyalty in the market
- How the market is segmented to meet different customer needs
- Customer preferences in terms of when and where they buy, what prices they pay and which methods of promotion are effective

- The potential for developing a competitive position in a market – either through a USP or through effective price competition

The process of analysing the market should not be considered as a one-off. An effective marketing team is constantly searching and updating their market knowledge.

However, detailed marketing analysis is particularly important for tasks such as:

- Forecasting sales for new products or investments into new markets
- Gathering evidence to support a finance-raising exercise
- To support a new marketing strategy or significant changes to the marketing objectives
- To help make decisions in relation to significant organisational or operational change

There are several methods of analysing a market that are in common use. The following methods are covered in the following sections:

- Test marketing
- Trend analysis (extrapolation, moving averages and correlation)

Scope

The scope of marketing analysis stretches from the identification of consumer wants and needs to the evaluation of consumer satisfaction. It comprises research relating to consumers, products, sales, distribution, advertising, pricing and sales forecasting.Then, we applied some classifications of market research activity.

Objective

Marketing research helps the marketing executive in the formulation of all marketing plans, policies, programmes and procedures, to attain the marketing goals. When the above said plans, policies and programmes are translated into action, marketing research is used for evaluation purposes.Since the main thrust of marketing research is on selling, promotion, advertising and distribution, it is used in minimizing all wasteful and unnecessary marketing cost. Further, the function of marketing research acts as an insurance cover for the survival and growth of the firm. Marketing research finds out for the producers.

(i) Where are his consumers?

(ii) What they want?

(iii) When they want it and?

(iv) Where and how much they are willing to pay for it?

Hence, through marketing research, marketing management can sell the right product through the right channels of distribution to the right consumers at the right places and at right price by formulating right plans, policies and programmes with the help of right personnel.

In short, marketing research helps producers, distributors and advertisers to avoid mistakes either in manufacturing or marketing or to that extent, it can minimize business failures.

2. LITERATURE SURVEY

Girish Punj and David Stewart (1983) have reviewed the applications of cluster analysis in marketing problems and they have recommended a two stage cluster analysis methodology and preliminary identification of clusters via Ward's minimum variance method. The authors have also discussed the issues and problems related to the use and validation of cluster analysis . Agrawal (1993), in their study, described that in the recent past the exploratory analysis in particular of large sets of market basket data has become topic of pertinent research due to various publications on data mining and knowledge in databases and generated association rules from market basket data, which describe relevant 19 interrelations like "If a consumer purchases fruit juice, then, in 40% of the cases they also purchase mineral water". Piatetsky-Shapiro et al. (1996) surveyed a growing number of industrial applications of data mining. The authors have examined the existing data mining tools, described some representative applications like marketing, investment, manufacturing, fraud detection etc. and discussed the issues for deploying successful applications and their adoption by business users. They also highlighted upon the fact of widespread realization of the potential value of data mining and a growing number of researchers and developers in this area . Anand (1996) focused on the organizations' need to invest in data mining solutions because of the phenomenal expansion of the data space and the resulting sharp increase in the size of typical data bases. There is no alternative to heavy reliance on computer programs set to discover patterns for themselves with relatively little human intervention. The authors also proposed a general framework of data mining based on

Evidence Theory that consisted of methods for representing data and knowledge, and methods for data manipulation and knowledge discovery [4]. According to Richard A. Spinello (1997), Wal-Mart captures point-of-sale data from over 2,900 stores in 6 countries and transmits this data to its massive 7.5-terabyte data warehouse and uses it to identify customer-buying patterns, to manage local store inventory and identify new merchandising opportunities using data mining techniques [143]. Peter Spiller and Gerald Lohse (1997), in their paper, present a classification of on-line retail stores based upon a convenience sample of 137 Internet retail stores. Cluster and factor analysis identified five distinct Web catalog interface categories which provide a better understanding of the strategies pursued in Internet-based marketing . Data mining and Knowledge discovery in databases (KDD) are concerned with extracting models and patterns of interest from large databases. Fayyad and Stolorz (1997), in their paper, provide an overview of this growing multi-disciplinary research area, outline the 20 basic techniques, and provide a brief coverage of how they are used in some applications in science data analysis

2.2 About Python

Python is a programming language, which means it'a a language both people and computers can understand. Python was developed by a Dutch software engineer named Guido van Rossum, who created the language to solve some problems he saw in computer languages of the time.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

You Can Use Python for Pretty Much Anything

One significant advantage of learning Python is that it’s a general-purpose language that can be applied in a large variety of projects. Below are just some of the most common fields where Python has found its use:

- Data science
- Scientific and mathematical computing
- Web development
- Computer graphics
- Basic game development
- Mapping and geography (GIS software)

Python Is Widely Used in Data Science

Python’s ecosystem is growing over the years and it’s more and more capable of statistical analysis.

It’s the best compromise between scale and sophistication (in terms of data processing).

Python emphasizes productivity and readability.

Python is used by programmers that want to delve into data analysis or apply statistical techniques (and by devs that turn to data science)

There are plenty of Python scientific packages for data visualization, machine learning, natural language processing, complex data analysis and more. All of these factors make Python a great tool for scientific computing and a solid alternative for commercial packages such as MatLab. The most popular libraries and tools for data science are:

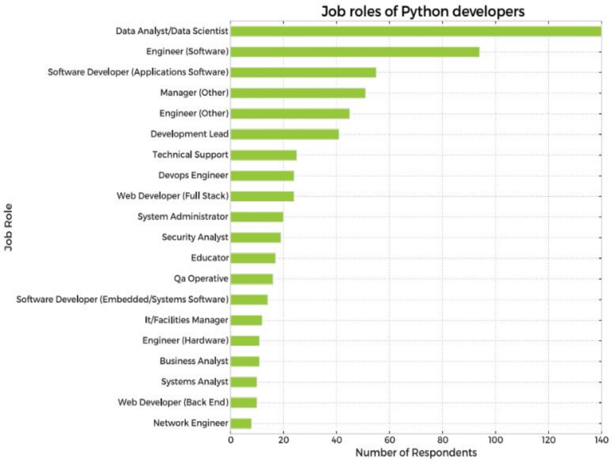
Pandas: a library for data manipulation and analysis. The library provides data structures and operations for manipulating numerical tables and time series.

NumPy: the fundamental package for scientific computing with Python, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

SciPy: a library used by scientists, analysts, and engineers doing scientific computing and technical computing.

Being a free, cross-platform, general-purpose and high-level programming language, Python has been widely adopted by the scientific community. Scientists value Python for its precise and efficient syntax, relatively flat learning curve and the fact that it integrates well with other languages (e.g. C/C++).

As a result of this popularity there are plenty of Python scientific packages for data visualization, machine learning, natural language processing, complex data analysis and more. All of these factors make Python a great tool for scientific computing and a solid alternative for commercial packages such as MatLab.



Here's our list of the most popular Python scientific libraries and tools

Pandas

Pandas is a library for data manipulation and analysis. The library provides data structures and operations for manipulating numerical tables and time series.

Mlpy

Mlpy is a machine learning library built on top of NumPy/SciPy, the GNU Scientific Libraries. Mlpy provides a wide range of machine learning methods for supervised and unsupervised problems and it is aimed at finding a reasonable compromise between modularity, maintainability, reproducibility, usability and efficiency.

matplotlib

Matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib allows you to generate plots, histograms, power spectra, bar charts, error charts, scatterplots, and more.

NumPy

NumPy is the fundamental package for scientific computing with Python, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

SciPy

SciPy is a library used by scientists, analysts, and engineers doing scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.

Anaconda is a free and open source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing,predictive analysis etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as **Anaconda Distribution** or **Anaconda Individual Edition**, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, which are both not free.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface(GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows,macOS,Linux.

Jupyter Notebook:

The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects. First, though: what is a “notebook”? A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. This intuitive workflow promotes iterative and rapid development, making notebooks an increasingly popular choice at the heart of contemporary data science, analysis, and increasingly science at large.The Jupyter project is the successor to the earlier IPython Notebook, which was first published as a prototype in 2010. Although it is possible to use many different programming languages within Jupyter Notebooks, this article will focus on Python as it is the most common use case.To get the most out of this tutorial you should be familiar with programming, specifically Python and pandas specifically. Jupyter Notebooks can also act as a flexible platform for getting to grips with pandas and even Python.

Python Saves Time

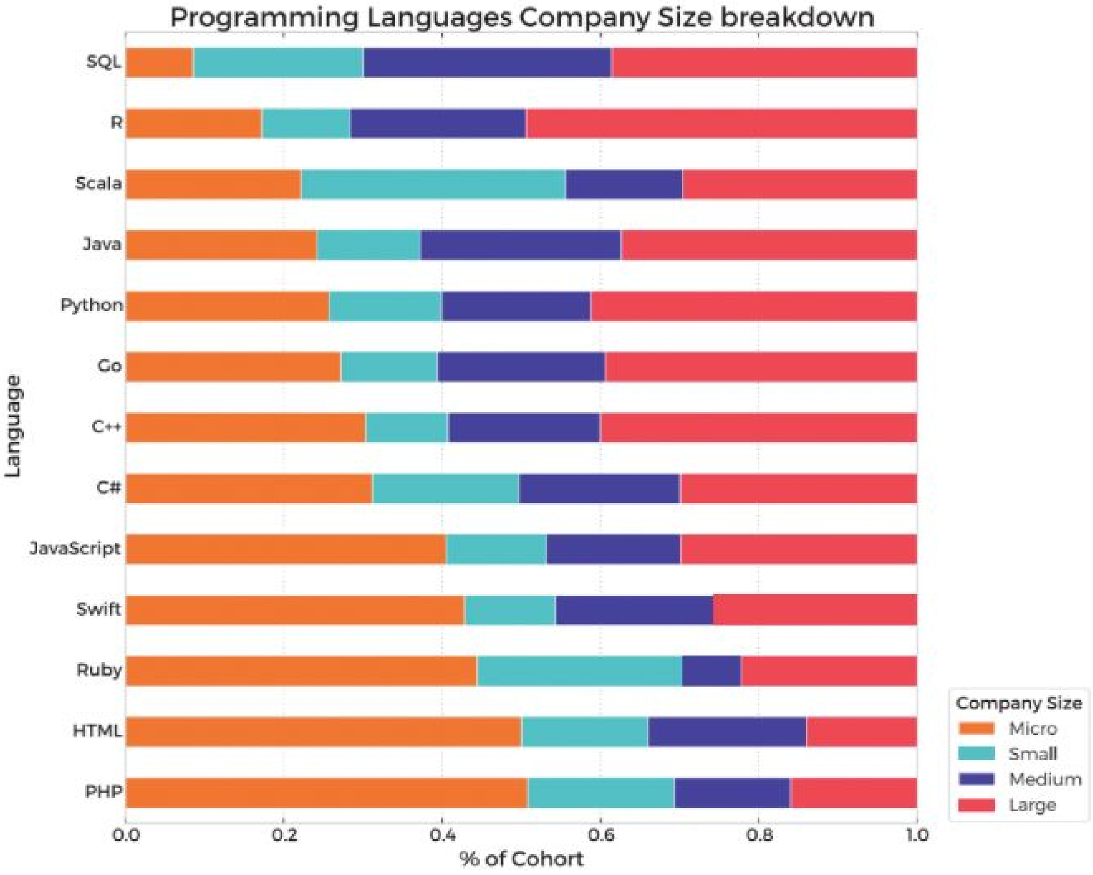
Even the classic “Hello, world” program illustrates this point:

```
print("Hello, world")
```

For comparison, this is what the same program looks like in Java:

```
public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello, world");

    }

}
```

All the Big Names Use Python



Python Keywords and Identifier

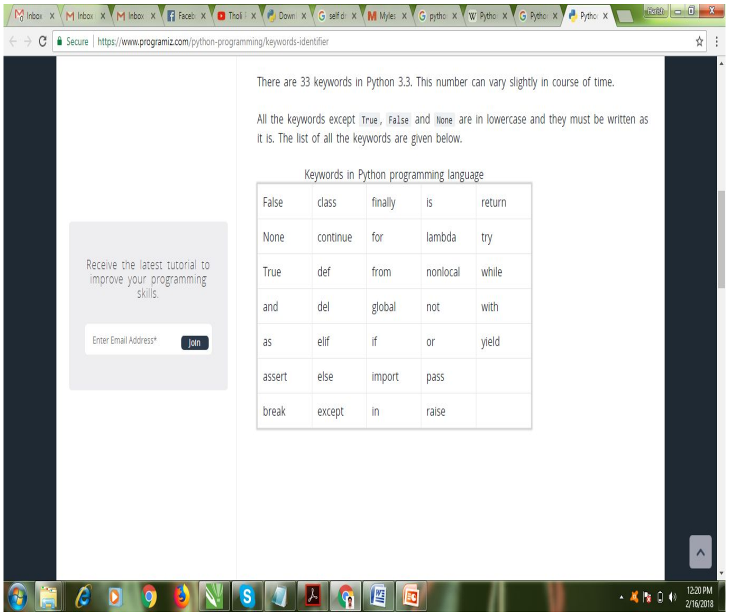
Keywords are the reserved words in Python.

We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.3. This number can vary slightly in course of time.

All the keywords except True, False and None are in lowercase and they must be written as it is. The list of all the keywords is given below.


Secure
https://www.programiz.com/python-programming/keywords-identifier
Receive the latest tutorial to improve your programming skills.
Enter Email Address*
Join
12:20 PM
2/16/2018

There are 33 keywords in Python 3.3. This number can vary slightly in course of time.

All the keywords except `True`, `False` and `None` are in lowercase and they must be written as it is. The list of all the keywords are given below.

Keywords in Python programming language

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Identifier is the name given to entities like class, functions, variables etc. in Python. It helps differentiate one entity from another.

Rules for writing identifiers

Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore (_). Names like myClass, var_1 and print_this_to_screen, all are valid examples.

An identifier cannot start with a digit. 1variable is invalid, but variable1 is perfectly fine.

Keywords cannot be used as identifiers.

```
>>> global = 1
  File "<interactive input>", line 1
    global = 1
           ^
```

SyntaxError: invalid syntax

We cannot use special symbols like !, @, #, $, % etc. in our identifier.

```
>>> a@ = 0
  File "<interactive input>", line 1
    a@ = 0
     ^
SyntaxError: invalid syntax
```

Identifiers can be of any length.

Python

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

Python Logo

3. SYSTEM ANALYSIS

3.1 Existing System:

The existing system uses A query based approach for analysing the data. This analysis provides a lot of computational challenges for analysing the data without a distributed mechanism. The data analysis is performed on a simple server environment without parallel processing.

Disadvantages of Existing System:

- When data sizes increase , the computational process becomes a challenging thing.
- No perfect visualisation.
- Parallel processing is not possible in clustering Analysis
- Mining techniques do not work well with big data in analysis when we occur with multiple column analysis.

3.2 Proposed System:

We are analyzing a Data set by using an object based approach within the Python programming Language. Python programming Language provides libraries which also focus on the implementation of data analysis and in representing the data in different visualizations.

Advantages:

- We propose a system where we use statistical analysis with sampling data in the analysis.
- Considering the data visualization which is not done in the big data analysis.
- No maximum skill is required by the analysers to perform the analysis.
- Python has good graphical libraries.
- The output is more effective using graphical libraries in Python.

4. SOFTWARE REQUIREMENT SPECIFICATION

The reason for this SRS record is to distinguish the necessities and functionalities for Intelligent Network Backup Tool. The SRS will characterize how our group and the customer consider the last item and the attributes or usefulness it must have. This record additionally makes a note of the discretionary prerequisites which we intend to execute yet are not required for the working of the venture.

This stage assesses the required necessities for the Images Processing for an orderly method for assessing the prerequisites a few procedures are included. The initial step associated with dissecting the prerequisites of the framework is perceiving the idea of framework for a solid examination and all the cases are defined to better comprehend the investigation of the dataset.

INTENDED AUDIENCE AND READING SUGGESTIONS

This record is proposed for extended engineers, directors, clients, analyzers and documentation journalists. This report goes for examining plan and execution imperatives, conditions, framework highlights, outside interface prerequisites and other non utilitarian necessities.

IDENTIFICATION OF NEEDS

The first and imperative need for a business firm or an association is to know how they are performing in the market and parallelly they have to know how to conquer their rivals in the market.

To do as such we have to investigation our information in view of all the accessible variables

4.1 FEASIBILITY STUDY

A credible contemplate expects to fairly-mindedly and soundly uncover the qualities and inadequacies of a present business or proposed meander, openings and threats present in nature, the benefits required to bring through, and in the long run the prospects for advance. In its most clear terms, the two criteria to judge believability are incurred significant injury required and motivator to be fulfilled.

An inside and out arranged feasibility ponder should give a recorded establishment of the business or wander, a delineation of the thing or organization, accounting explanations, purposes of enthusiasm of the operations and organization, publicizing examination and game plans, budgetary data, authentic necessities and cost duties. All things considered, plausibility looks at go before specific change and wander utilization. There are three sorts of attainability

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

Economical feasibility

The electronic structure manages the present existing system's data stream and technique absolutely and should make each one of the reports of the manual structure other than a substantial gathering of the other organization reports. It should be filled in as an electronic application with specific web server and database server. Advance a segment of the associated trades happen in different ranges. Open source programming like TOMCAT, JAVA, MySQL and Linux is used to restrict the cost for the Customer. No extraordinary wander need to manage the instrument.

Technical feasibility

Surveying the particular probability is the trickiest bit of a believability consider. This is in light of the fact that, starting at the present moment, not a lot of point by point layout of the system, making it difficult to get to issues like execution, costs on (by excellence of the kind of development to be passed on) et cetera.

Different issues must be considered while doing a particular examination. Grasp the differing progressions required in the proposed system. Before starting the wander, we should be clear about what are the advances that are to be required for the change of the new system.

Check whether the affiliation by and by has the required advancements. Is the required development open with the affiliation?

In case so is the utmost sufficient?

For instance – "Will the present printer have the ability to manage the new reports and structures required for the new system?"

Operational feasibility

Proposed wanders are profitable just if they can be changed into information systems that will meet the affiliations working necessities. Simply communicated, this trial of probability asks with reference to whether the structure will work when it is made and presented. Are there genuine obstacles to Implementation? Here are questions that will help test the operational achievability of a wander.

- Is there sufficient help for the wander from organization from customers? In case the present structure is particularly cherished and used to the extent that individuals won't have the ability to see purposes behind change, there may be resistance.
- Are the present business methodologies qualified to the customer? If they are not, Users may welcome a change that will accomplish a more operational and supportive system.
- Have the customers been locked in with the orchestrating and change of the wander? Early commitment decreases the chances of impenetrability to the structure.

4.2 SOFTWARE REQUIREMENTS

Operating System : Windows 7 , Windows 8, (or higher versions)

Language : Python 3.5

Mozilla Firefox(or any browser)

4.3 HARDWARE REQUIREMENTS

Processor : Pentium 3,Pentium 4 and higher

RAM : 2GB/4GB RAM and higher

Hard disk : 40GB and higher

Processor	:	Pentium
RAM	:	2GB
Hard disk	:	80GB

Figure 4.1

5. SYSTEM DESIGN:

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements traceability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

This section describes any constraints in the system design (reference any trade-off analyses conducted such as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.

The organization code and title of the key points of contact (and alternates if appropriate) for the information system development effort. These points of contact should include the Project Manager, System Proponent, User Organization, Quality Assurance (QA) Manager, Security Manager, and Configuration Manager, as appropriate.

5.1 SYSTEM ARCHITECTURE

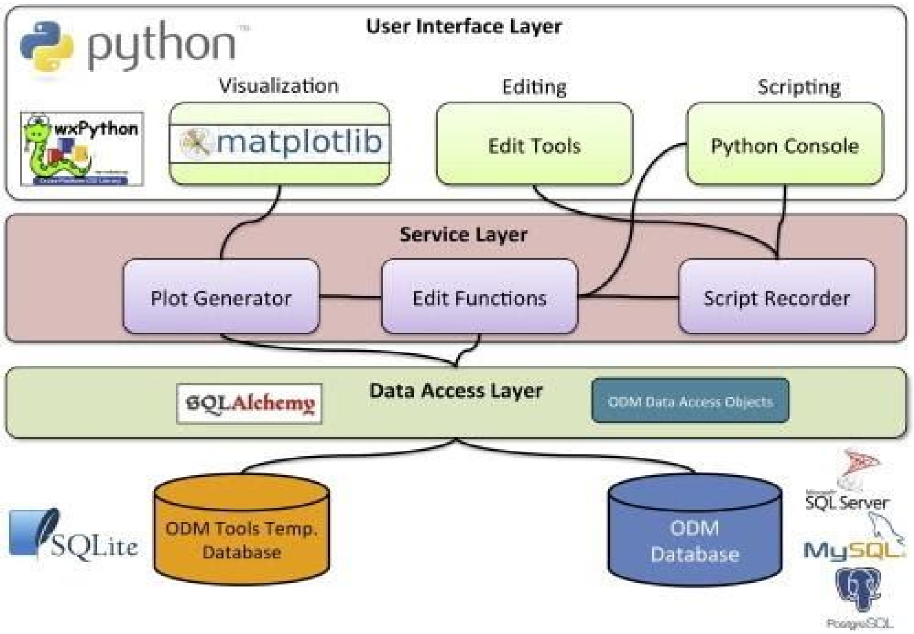


Fig 5.1 System architecture

5.2 Modules

- **Import Data**
- **Data Processing**
- **Results Evaluation**

5.3 UML Diagrams

UML (Unified Modeling Language) is a standard vernacular for choosing, envisioning, making, and specifying the collectibles of programming structures. UML is a pictorial vernacular used to make programming blueprints. It is in like way used to exhibit non programming structures similarly like process streams in a gathering unit and so forth.

UML is not a programming vernacular yet rather instruments can be utilized to make code in different tongues utilizing UML graphs. UML has an incite relationship with question composed examination and outline. UML expects a fundamental part in portraying trade viewpoints of a structure.

Use case Diagram:

The use case graph is for demonstrating the directness of the structure. This chart contains the course of action of use cases, performing pros and their relationship. This chart might be utilized to address the static perspective of the structure.

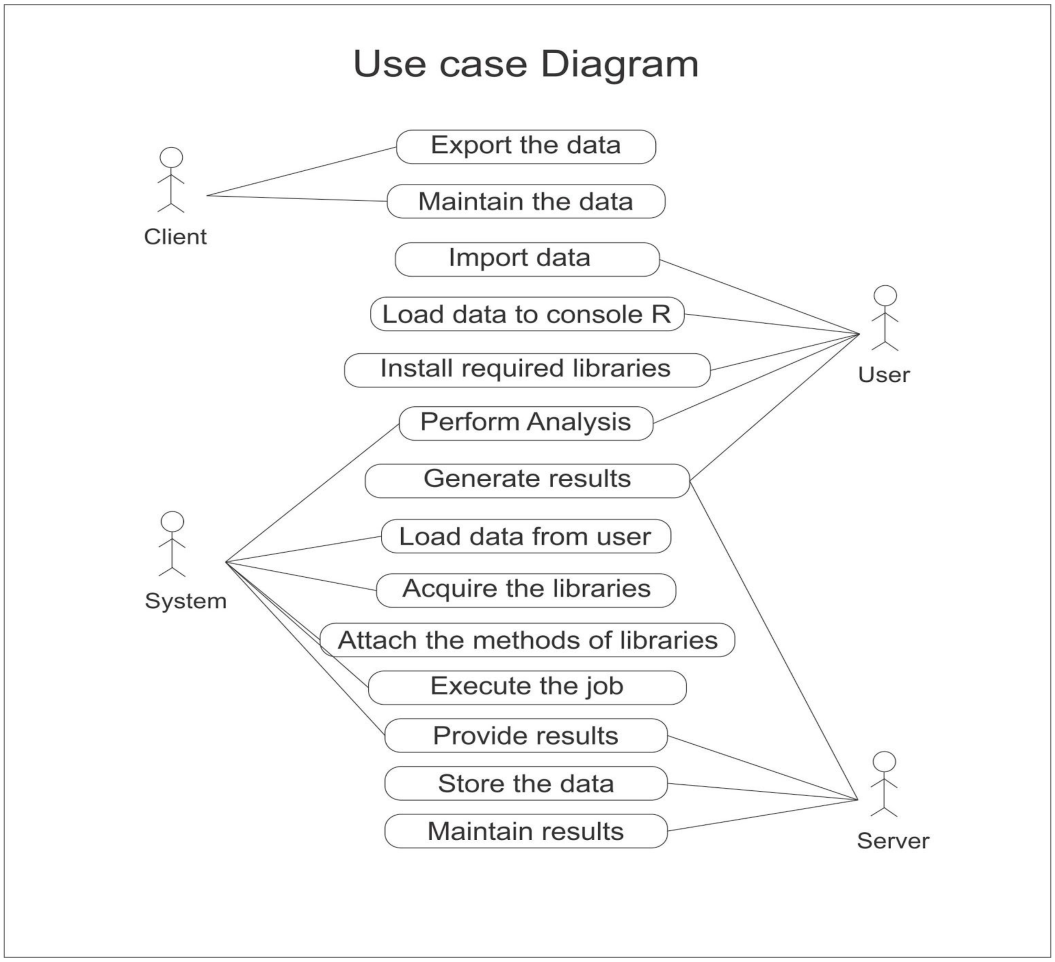


Fig: 5.2 Use Case Diagram

In the above diagram, the performing specialists are customer, structure, client, server, Python and data cleaning. The client exchanges the data to the system which disengages the data into squares and gives the data to Python. By then Python does the data cleaning which is just performing data connection and data repairing, by then the results will be secured. These results can be seen using Python and can be secured on the server for future reasons. The gained results can be created as reports by the customer.

Class Diagram: The class graph is the most normally pulled in UML layout. It addresses the static course of action perspective of the structure. It solidifies the strategy of classes, interfaces, joint attempts and their affiliations.

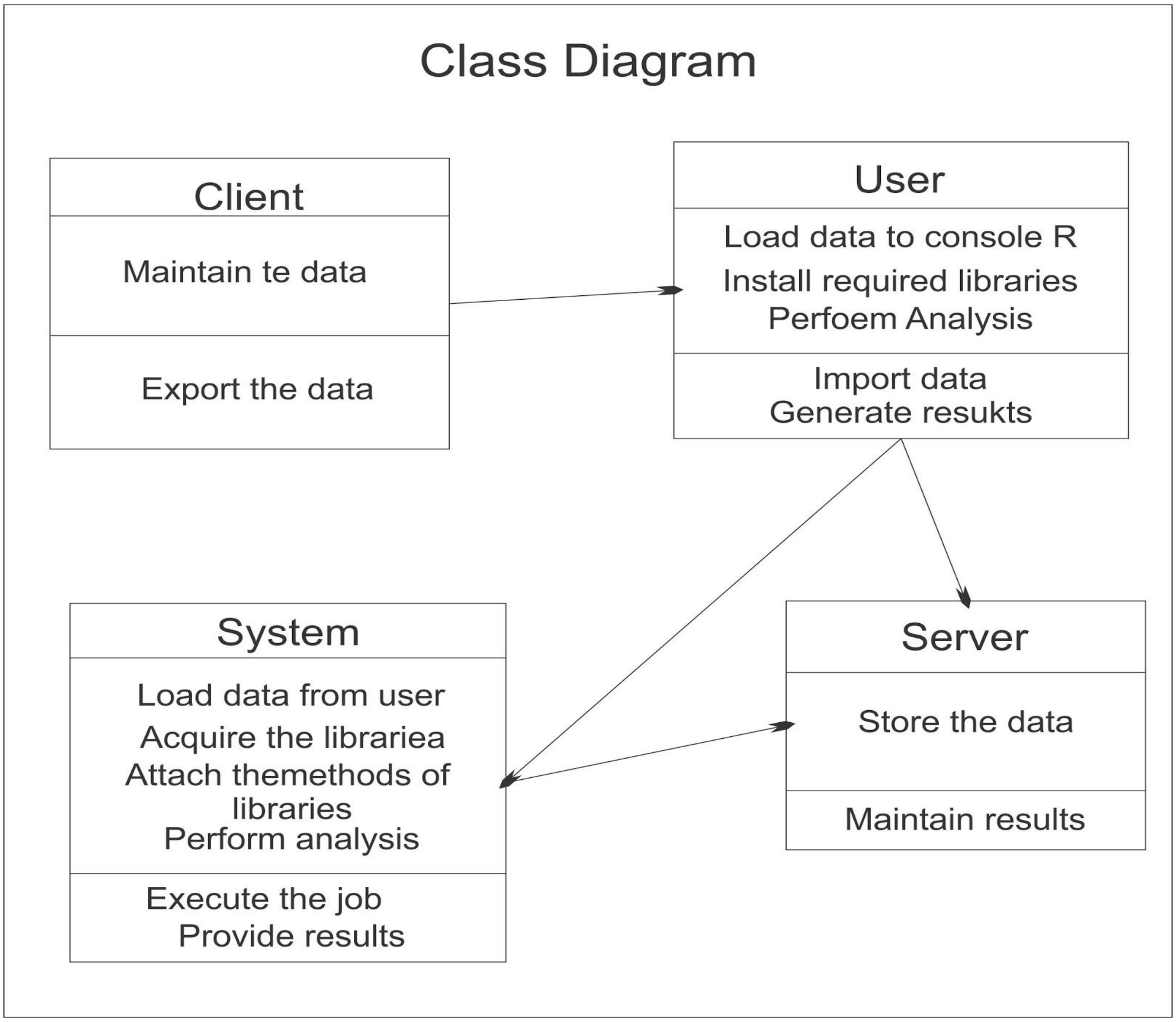


Fig: 5.3 Class Diagram

In the above class diagram, the relationship that is the dependence between each one of the classes is sketched out. Additionally, even the operations performed in each and every class similarly appeared.

Sequence Diagram:

This is a cooperation design which tends to the time requesting messages. It includes a set of parts and the messages sent and gotten by the instance of parts. This chart is utilized to address the dynamic perspective of the structure.

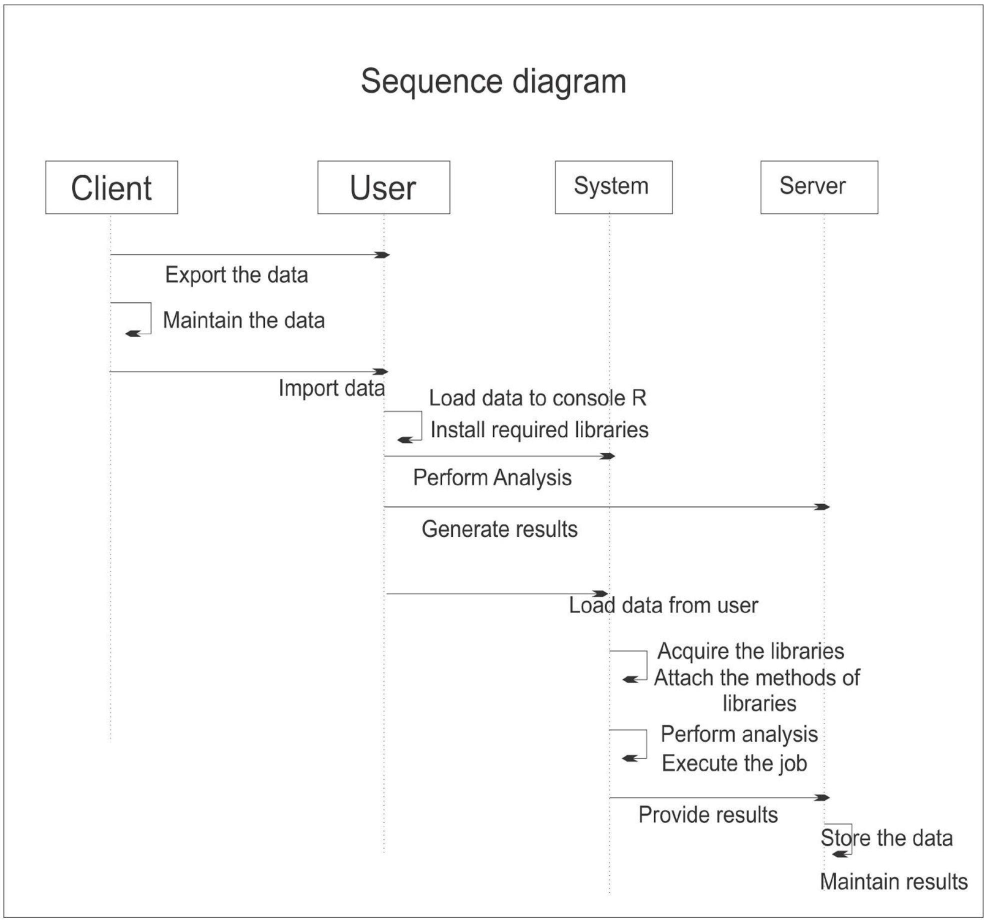


Fig: 5.4 Sequence Diagram

A succession outline indicates question communications masterminded in time arrangement. In the above graph, there are five articles cooperating with each other. Each protest has a vertical dashed line which speaks to the presence of a question over some undefined time frame. This graph has additionally a tall, thin rectangle which is called center of control that demonstrates the timeframe amid which a protest is playing out an activity, either specifically or through a subordinate system.

Collaboration Diagram:

This is a support format, which tends to the principal relationship of articles that send and get messages. It incorporates a set of parts, connectors that interface the parts and the messages sent and get by those parts. This graph is utilized to address the dynamic perspective of the framework.

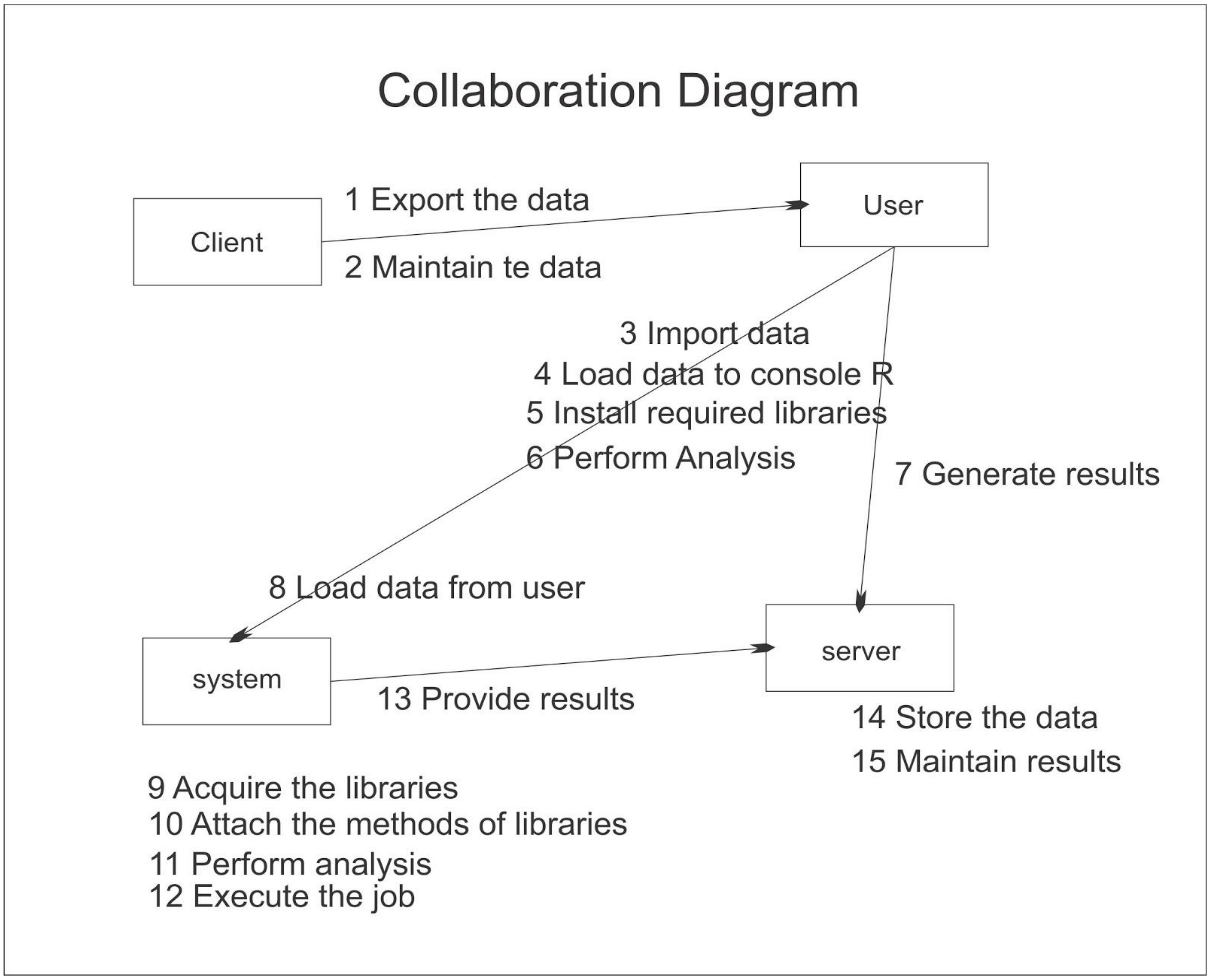


Fig: 5.5 Collaboration Diagram

The joint effort outline contains articles, ways and arrangement numbers. In the above graph, there are five questions specifically customer, client, framework, Python and server. These items are connected to each other utilizing a way. A succession number shows the time request of a message.

State Chart Diagram:

The state graph contains the game-plan of states, occasions and exercises. This graph is noteworthy for tending to the lead of the interface, class and made effort. The key centralization of state outline is to show the occasion to sort out the lead of the request. The state follows a diagram the dynamic perspective of the framework.

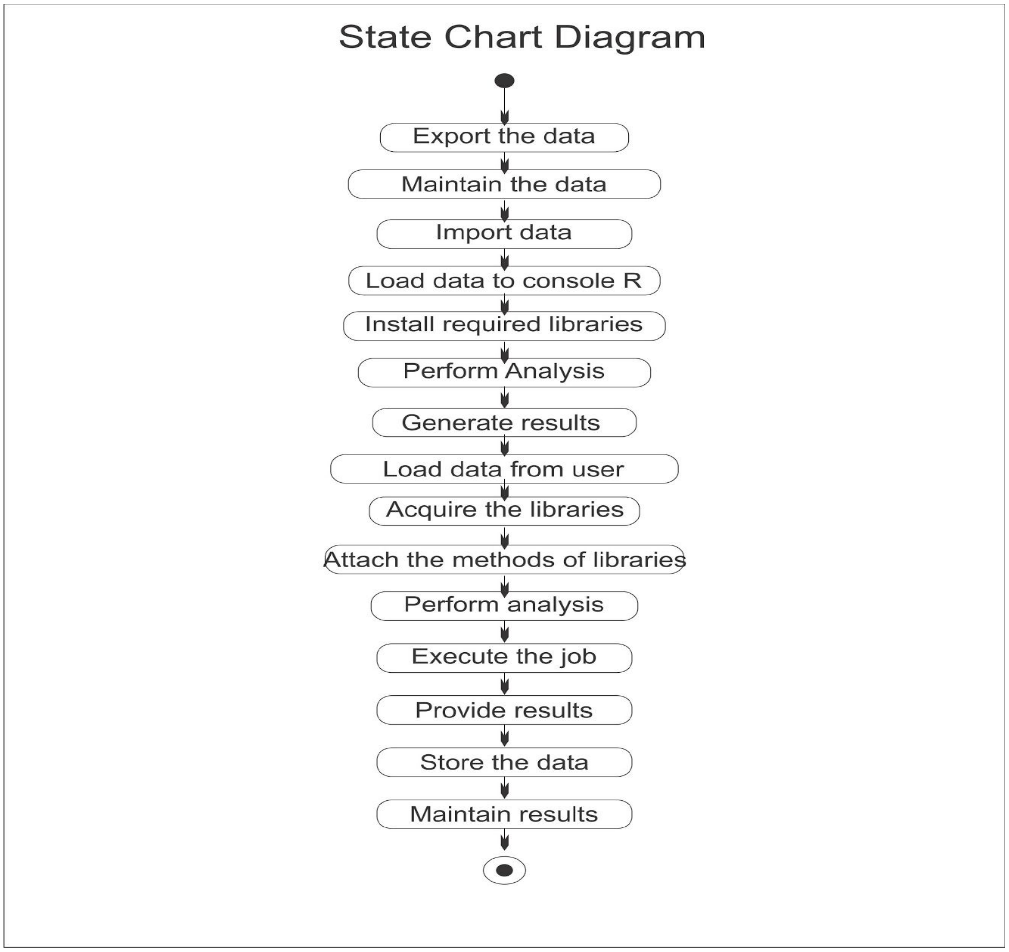


Fig: 5.6 State Chart Diagram

A state outline graph contains two components called states and progress. States speak to circumstances amid the life of a question. We can without much of a stretch outline a state in SmartDraw by utilizing a rectangle with adjusted corners. Change is a strong bolt that speaks to the way between various conditions of a question. Name the change with the occasion that activated it and the activity those outcomes from it.

Component Diagram:

The imperative portion of part format is segment. This diagram demonstrates within parts, connectors and ports that understand the piece. Precisely when a section is instantiated, duplicates of inside parts are besides instantiated.

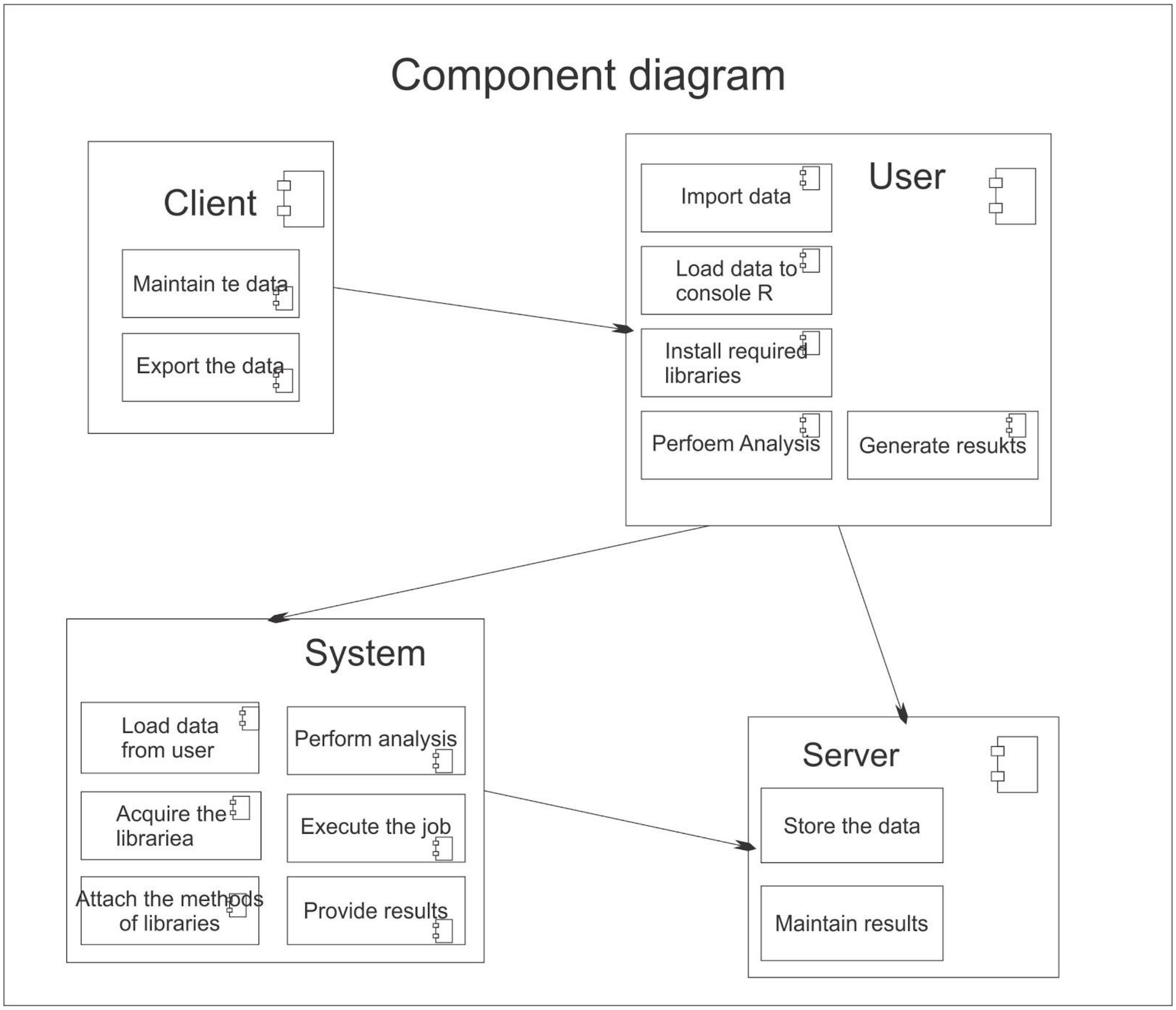


Fig: 5. 7 Component Diagram

A part outline is spoken to the utilizing segment. A part is a physical building piece of the framework. It is spoken as a rectangle with tab. Part outline portrays the inward handling of the venture. The information is sent to Python where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

Deployment Diagram:

The fundamental fragment in game-plan layout is a middle point. The strategy of focus and their relationship with others tended to utilize sending plot. The sending outline is identified with the area diagram, that is one focus purpose obviously of activity format frequently includes no short of what one section. This outline is in like way critical for tending to the static perspective of the framework.

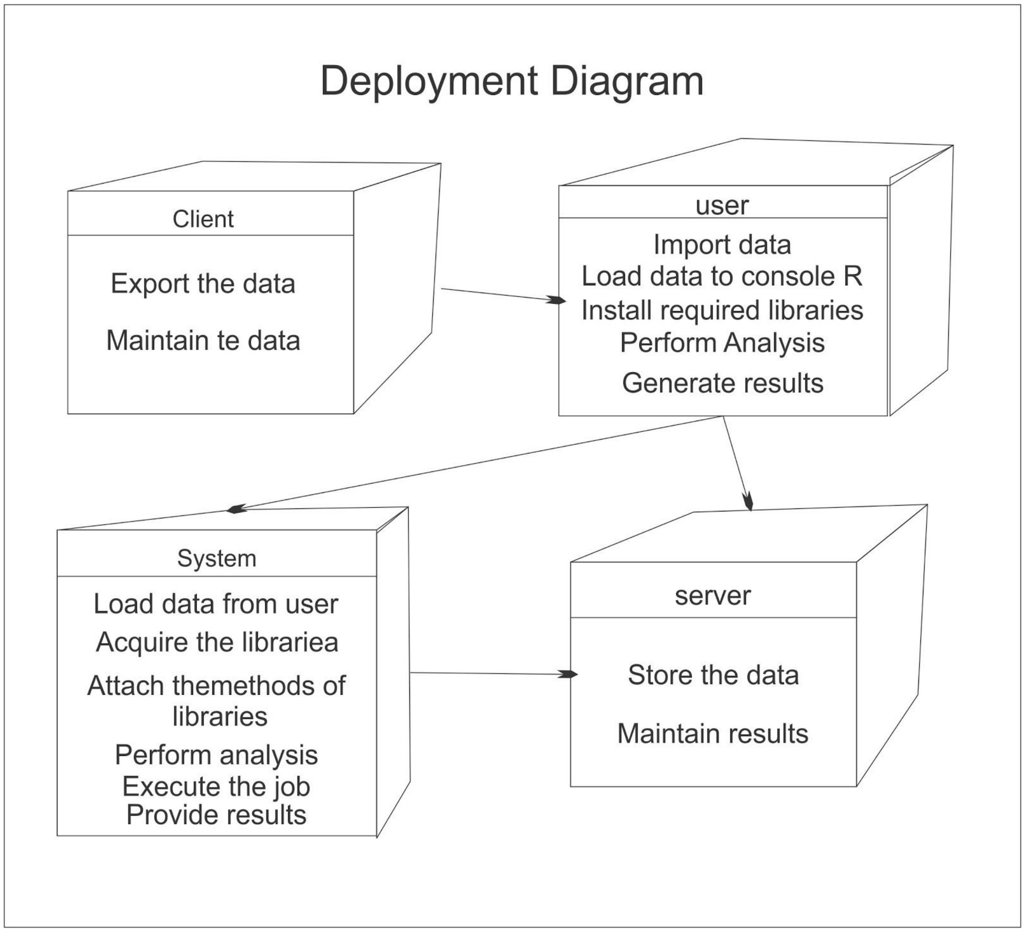


Fig: 5.8 Deployment Diagram

An arrangement graph is spoken of utilizing hubs. A hub is a physical asset that executes code parts. They are likewise used to portray run time handling of hubs. The information is sent to Python where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

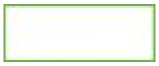
5.4 DATA FLOW DIAGRAMS

An information stream design (DFD) is a graphical portrayal of the "stream" of information through a data framework, demonstrating its strategy edges. A DFD is a significant part of the time utilized as a preparatory stroll to make an overview of the framework, which can later be cleared up. DFDs can in like way be utilized for the depiction of information prepared. A DFD indicates what sort of data will be sense of duty regarding and yield from the structure, where the information will begin from and go to, and where the information will be secured. It doesn't demonstrate data about the organizing of process or data about whether strategy will work in game-plan or in parallel.

DFD Symbols:

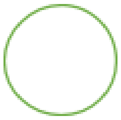
In the DFD, there are four symbols

- A square defines a source or destination of system data.

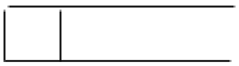


- An arrow identifies data flow. It is the pipeline through which the information flows.

- A circle represents a process that transforms incoming data flow into outgoing data flow.



- An open rectangle is a data store, data at rest or a temporary repository of data.



Level 0: System input/ output level

A level 0 DFD describes the system wide boundaries, dealing input to and output flow from the system and major processes.

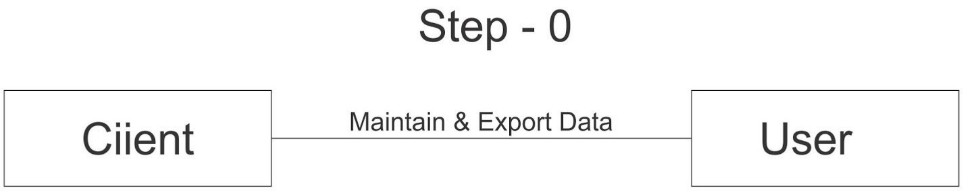


Fig 5.5 Level 0 DFD

DFD Level 0 is in a way called a Context Diagram. It's an urgent review of the entire structure or process being bankrupt or appeared. It's required to be an at first watch, demonstrating the framework as a particular surprising state handle, with its relationship to outside substances.

Level 1: Sub system level data flow

Level 1 DFD delineates the accompanying level of purposes of enthusiasm with the data stream between subsystems. The Level 1 DFD exhibits how the system is secluded into sub-structures (shapes), each of which oversees no less than one of the data streams to or from an outside pro, and which together give most of the helpfulness of the system as a rule.

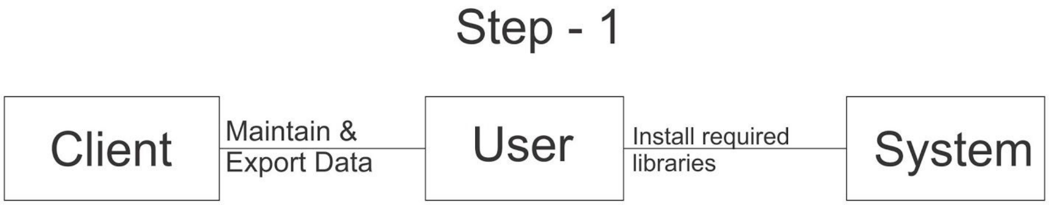


Fig 5.6 Level 1 DFD

Level 2: File level detail data flow

Plausibility and danger examination are connected here from various perspectives. The level 2 DFD elucidates the fundamental level of understanding about the system's working.

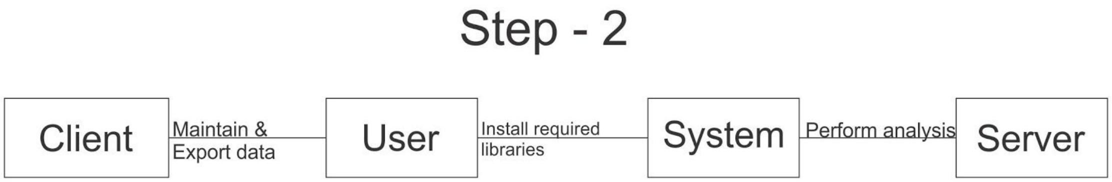


Fig 5.7 Level 2 DFD

Level 3:

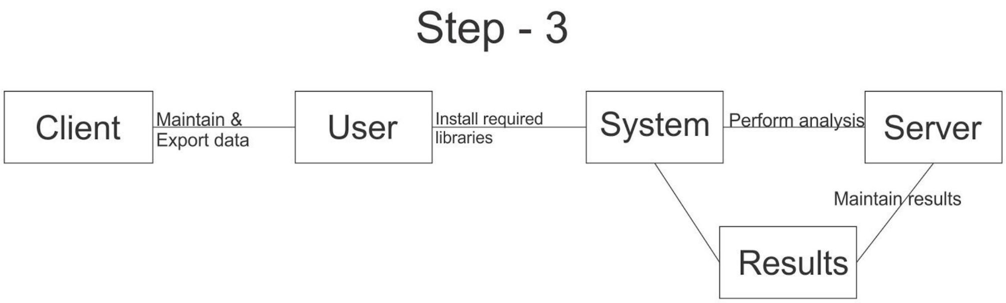
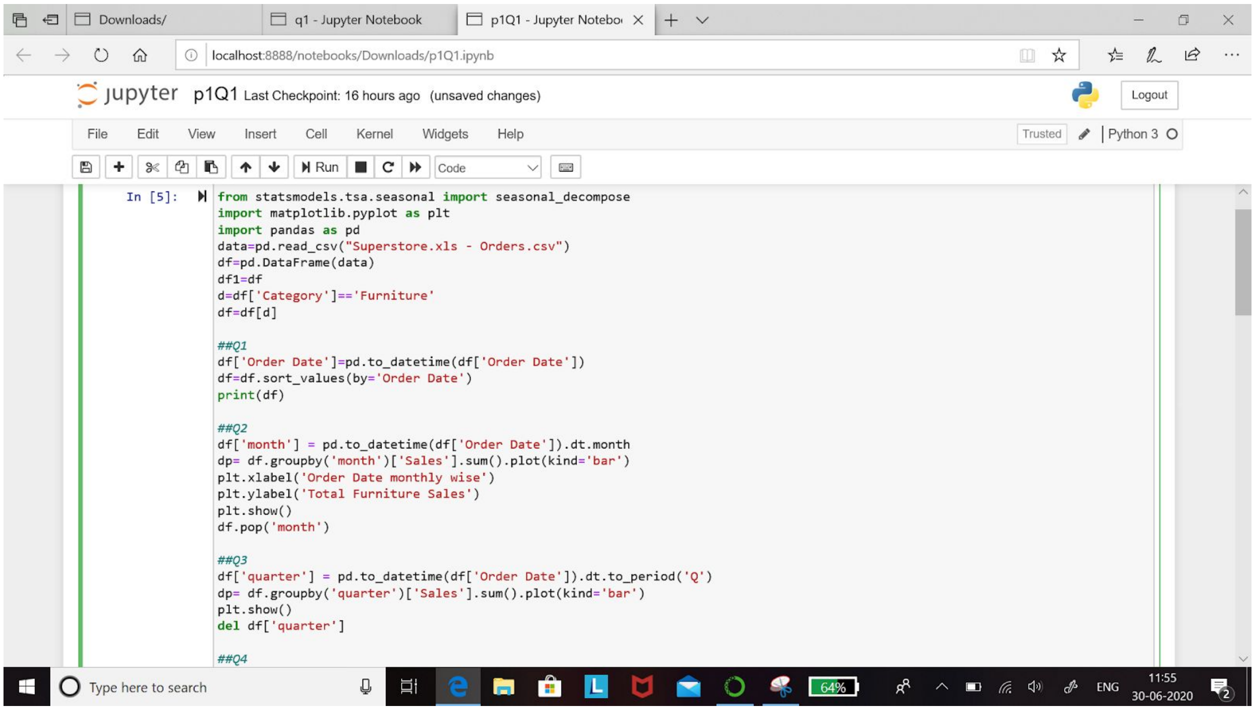


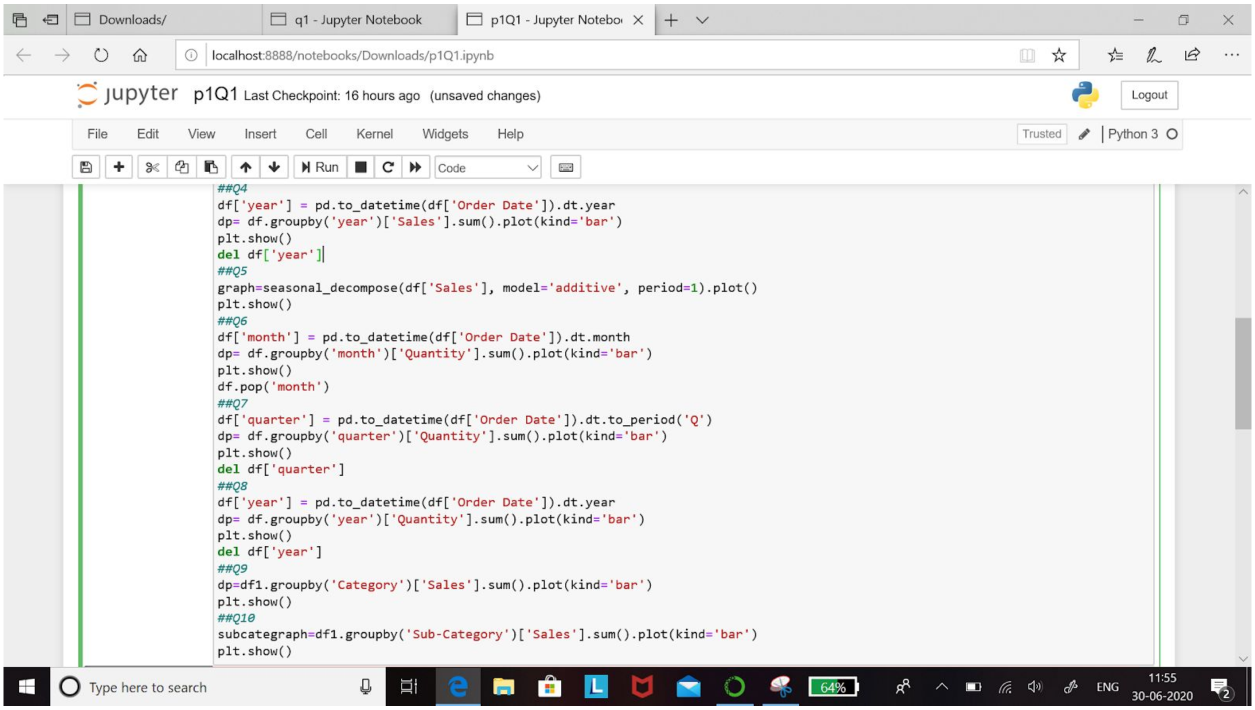
Fig 5.8 Level 3 DFD

6. Implementation

6.1 Sample Code


Downloads/
q1 - Jupyter Notebook
p1Q1 - Jupyter Notebo
localhost:8888/notebooks/Downloads/p1Q1.ipynb
jupyter p1Q1 Last Checkpoint: 16 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [5]:
Type here to search
64%
ENG
11:55
30-06-2020

```
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
df1=df
d=df['Category']=='Furniture'
df=df[d]

##Q1
df['Order Date']=pd.to_datetime(df['Order Date'])
df=df.sort_values(by='Order Date')
print(df)

##Q2
df['month'] = pd.to_datetime(df['Order Date']).dt.month
dp= df.groupby('month')['Sales'].sum().plot(kind='bar')
plt.xlabel('Order Date monthly wise')
plt.ylabel('Total Furniture Sales')
plt.show()
df.pop('month')

##Q3
df['quarter'] = pd.to_datetime(df['Order Date']).dt.to_period('Q')
dp= df.groupby('quarter')['Sales'].sum().plot(kind='bar')
plt.show()
del df['quarter']

##Q4
```


Downloads/
q1 - Jupyter Notebook
p1Q1 - Jupyter Notebo
localhost:8888/notebooks/Downloads/p1Q1.ipynb
jupyter p1Q1 Last Checkpoint: 16 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
64%
ENG
11:55
30-06-2020

```
##Q4
df['year'] = pd.to_datetime(df['Order Date']).dt.year
dp= df.groupby('year')['Sales'].sum().plot(kind='bar')
plt.show()
del df['year']
##Q5
graph=seasonal_decompose(df['Sales'], model='additive', period=1).plot()
plt.show()
##Q6
df['month'] = pd.to_datetime(df['Order Date']).dt.month
dp= df.groupby('month')['Quantity'].sum().plot(kind='bar')
plt.show()
df.pop('month')
##Q7
df['quarter'] = pd.to_datetime(df['Order Date']).dt.to_period('Q')
dp= df.groupby('quarter')['Quantity'].sum().plot(kind='bar')
plt.show()
del df['quarter']
##Q8
df['year'] = pd.to_datetime(df['Order Date']).dt.year
dp= df.groupby('year')['Quantity'].sum().plot(kind='bar')
plt.show()
del df['year']
##Q9
dp=df1.groupby('Category')['Sales'].sum().plot(kind='bar')
plt.show()
##Q10
subcategraph=df1.groupby('Sub-Category')['Sales'].sum().plot(kind='bar')
plt.show()
```

7. TESTING

7.1 INTRODUCTION TO TESTING

Testing is a procedure, which uncovers blunders in the program. Programming testing is a basic component of programming quality affirmation and speaks to a definitive audit of determination, outline and coding. The expanding perceivability of programming as a framework component and chaperon costs related with a product disappointment are propelling variables for us arranged, through testing. Testing is the way toward executing a program with the plan of finding a mistake. The plan of tests for programming and other built items can be as trying as the underlying outline of the item itself It is the significant quality measure utilized amid programming improvement. Amid testing, the program is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing as it is relied upon to perform.

7.2 TESTING STRATEGIES

A technique for programming testing coordinates the outline of programming experiments into an all around arranged arrangement of steps that result in fruitful improvement of the product. The procedure gives a guide that portrays the means to be taken, when, and how much exertion, time, and assets will be required. The procedure joins test arranging, experiment configuration, test execution, and test outcome gathering and assessment. The procedure gives direction to the specialist and an arrangement of points of reference for the chief. Due to time weights, advance must be quantifiable and issues must surface as ahead of schedule as would be prudent .Keeping in mind the end goal to ensure that the framework does not have blunders, the distinctive levels of testing techniques that are connected at varying periods of programming improvement are:

- Unit Testing
- Testing
- White Box testing
- Integration Testing
- System Testing
- Acceptance Testing

7.3 TEST APPROACH

A Test approach is the test system usage of a venture, characterizing how testing would be done. The decision of test methodologies or test technique is a standout amongst the most intense factors in the achievement of the test exertion and the precision of the test designs and gauges.

Testing should be possible in two ways

- Bottom up approach
- Top down approach

7.4 VALIDATION

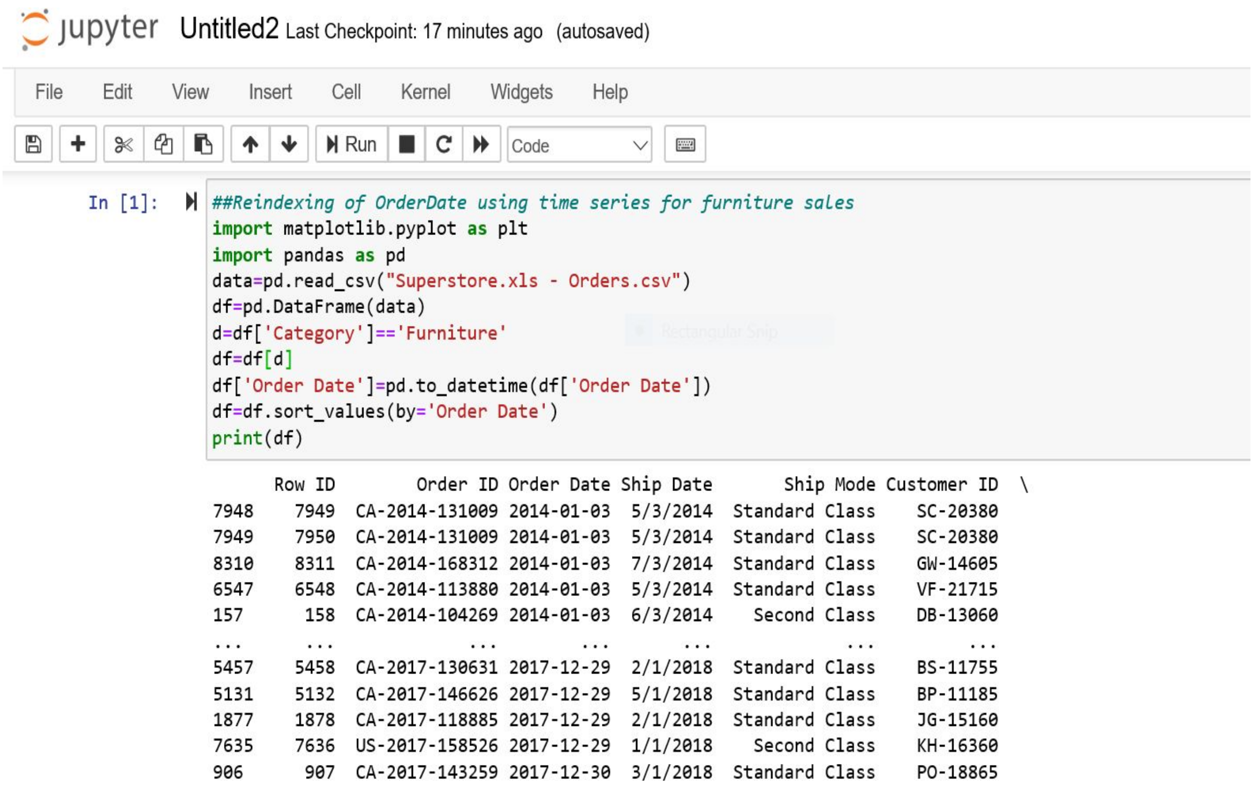
The way toward assessing programming amid the improvement procedure or toward the finish of the advancement procedure to decide if it fulfills determined business prerequisites. Approval Testing guarantees that the item really addresses the customer's issues. It can likewise be characterized as to exhibit that the item satisfies its proposed utilize when sent on proper condition.

The framework has been tried and actualized effectively and along these lines guaranteed that every one of the prerequisites as recorded in the product necessities determination are totally satisfied.

7.5 Test Cases

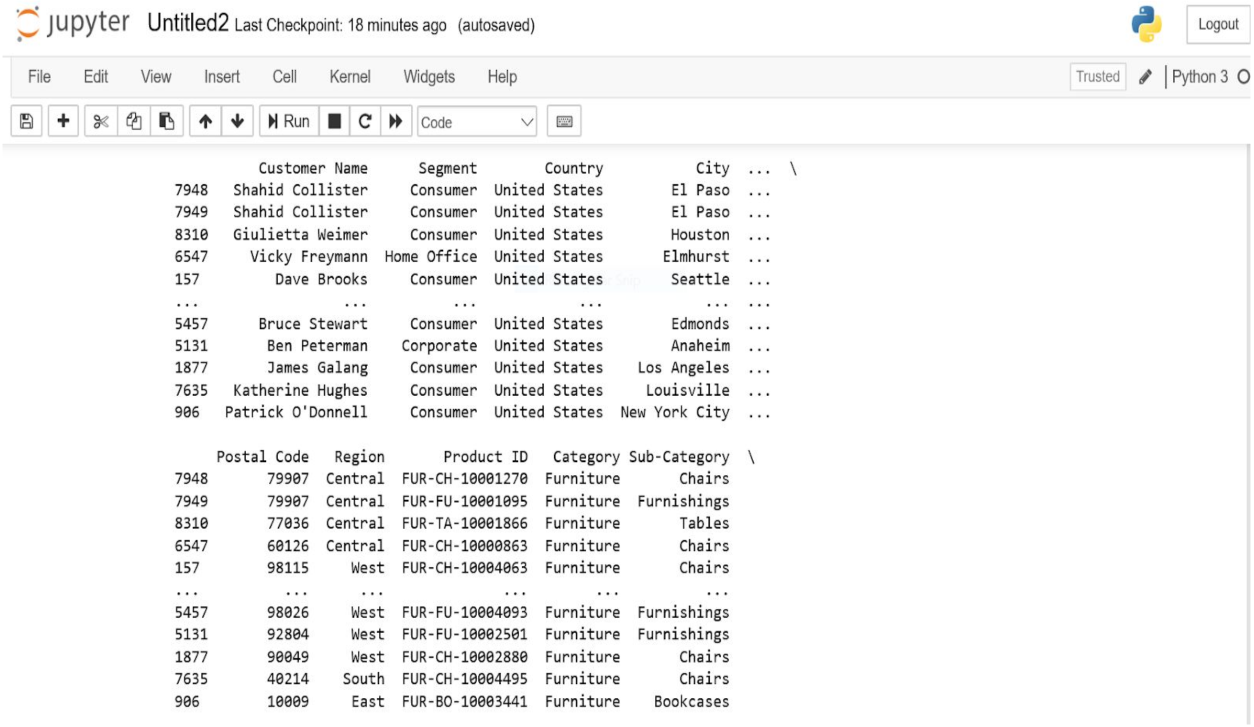
Experiments include an arrangement of steps, conditions and sources of info that can be utilized while performing testing undertakings. The principle expectation of this action is to guarantee whether a product passes or bombs as far as usefulness and different perspectives. The way toward creating experiments can likewise help discover issues in the prerequisites or plan of an application. Experiment goes about as the beginning stage for the test execution, and in the wake of applying an arrangement of information esteems, the application has a conclusive result and leaves the framework at some end point or otherwise called execution post condition.

8. Screenshots

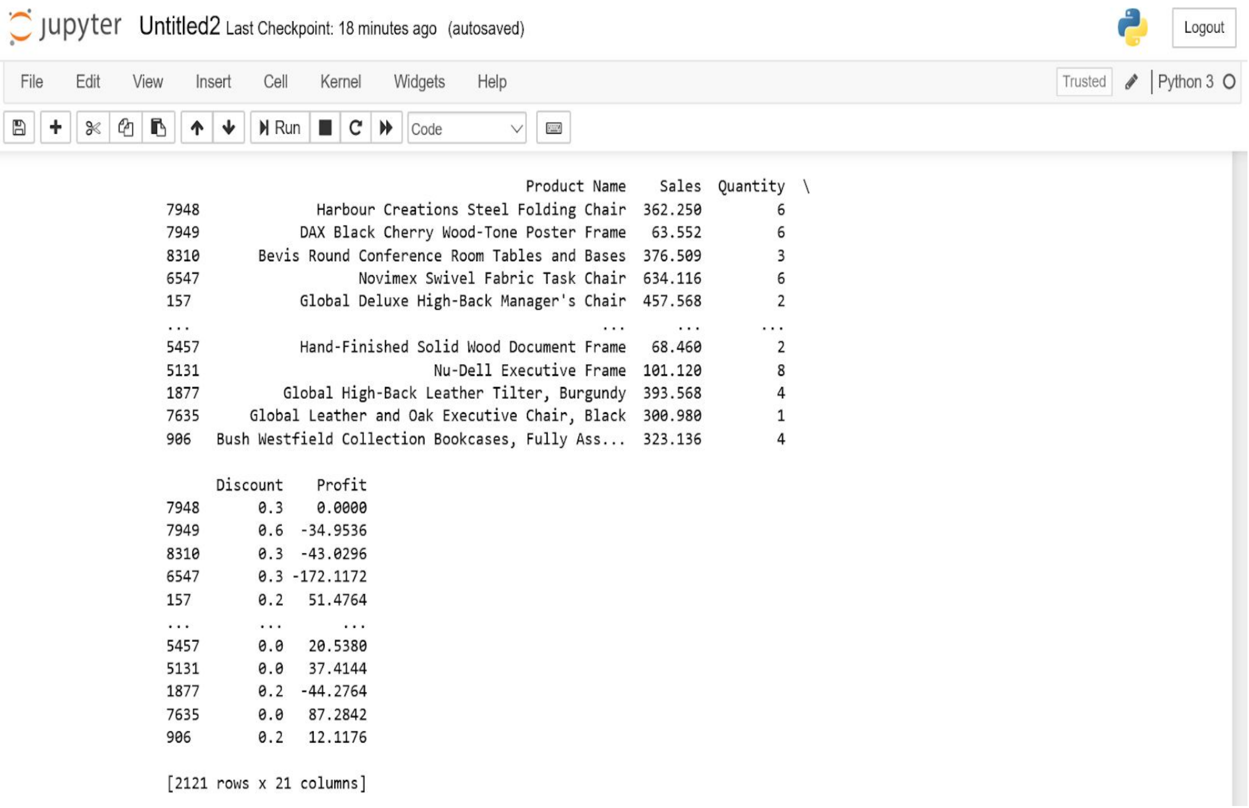


```
##Reindexing of OrderDate using time series for furniture sales
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['Order Date']=pd.to_datetime(df['Order Date'])
df=df.sort_values(by='Order Date')
print(df)
```

```
      Row ID        Order ID Order Date Ship Date       Ship Mode Customer ID  \
7948    7949  CA-2014-131009 2014-01-03  5/3/2014  Standard Class    SC-20380
7949    7950  CA-2014-131009 2014-01-03  5/3/2014  Standard Class    SC-20380
8310    8311  CA-2014-168312 2014-01-03  7/3/2014  Standard Class    GW-14605
6547    6548  CA-2014-113880 2014-01-03  5/3/2014  Standard Class    VF-21715
157      158  CA-2014-104269 2014-01-03  6/3/2014    Second Class    DB-13060
...      ...             ...        ...       ...             ...         ...
5457    5458  CA-2017-130631 2017-12-29  2/1/2018  Standard Class    BS-11755
5131    5132  CA-2017-146626 2017-12-29  5/1/2018  Standard Class    BP-11185
1877    1878  CA-2017-118885 2017-12-29  2/1/2018  Standard Class    JG-15160
7635    7636  US-2017-158526 2017-12-29  1/1/2018    Second Class    KH-16360
906      907  CA-2017-143259 2017-12-30  3/1/2018  Standard Class    PO-18865
```

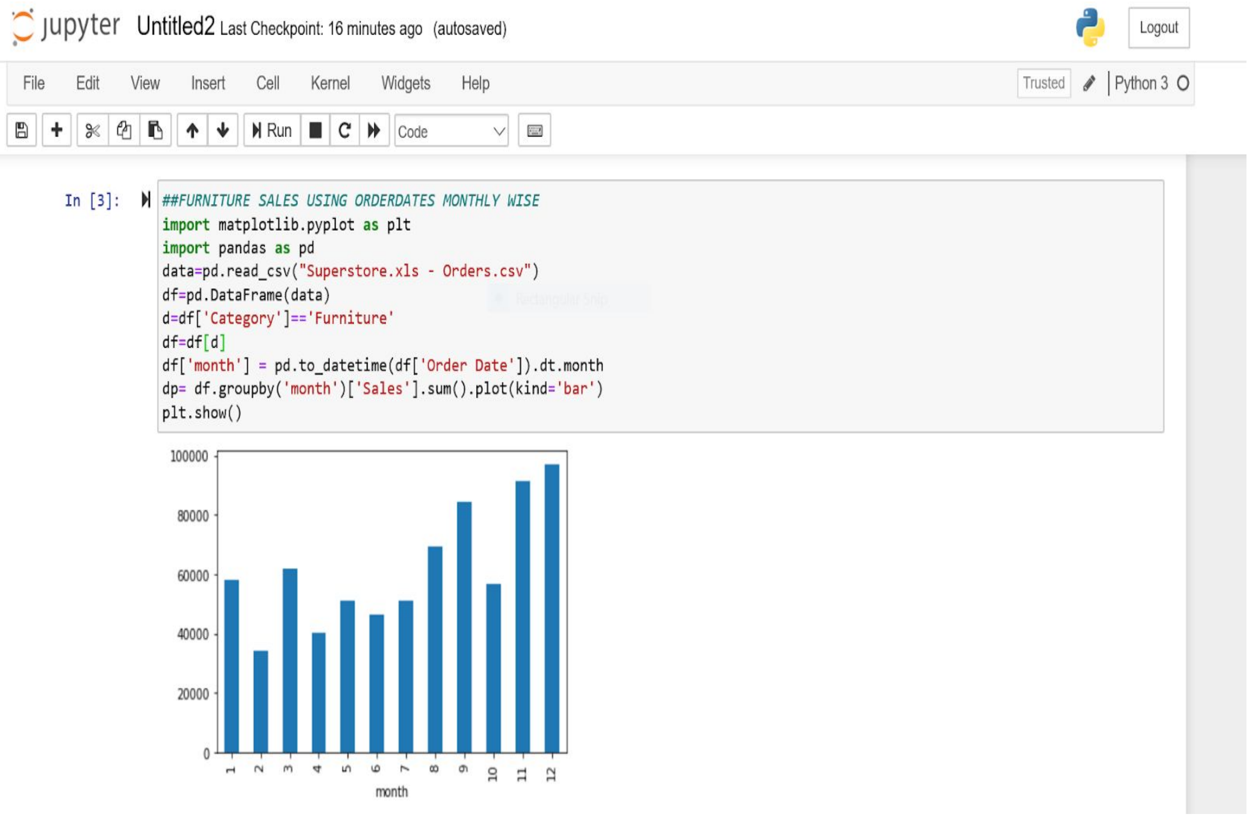


```
         Customer Name    Segment        Country          City  ...  \
7948  Shahid Collister   Consumer  United States       El Paso  ...
7949  Shahid Collister   Consumer  United States       El Paso  ...
8310  Giulietta Weimer   Consumer  United States       Houston  ...
6547    Vicky Freymann  Home Office  United States     Elmhurst  ...
157        Dave Brooks   Consumer  United States       Seattle  ...
...                ...        ...            ...           ...  ...
5457     Bruce Stewart   Consumer  United States       Edmonds  ...
5131      Ben Peterman  Corporate  United States       Anaheim  ...
1877      James Galang   Consumer  United States   Los Angeles  ...
7635  Katherine Hughes   Consumer  United States    Louisville  ...
906   Patrick O'Donnell  Consumer  United States  New York City  ...

      Postal Code   Region       Product ID   Category Sub-Category  \
7948        79907  Central  FUR-CH-10001270  Furniture       Chairs
7949        79907  Central  FUR-FU-10001095  Furniture  Furnishings
8310        77036  Central  FUR-TA-10001866  Furniture       Tables
6547        60126  Central  FUR-CH-10000863  Furniture       Chairs
157         98115     West  FUR-CH-10004063  Furniture       Chairs
...           ...      ...              ...        ...          ...
5457        98026     West  FUR-FU-10004093  Furniture  Furnishings
5131        92804     West  FUR-FU-10002501  Furniture  Furnishings
1877        90049     West  FUR-CH-10002880  Furniture       Chairs
7635        40214    South  FUR-CH-10004495  Furniture       Chairs
906         10009     East  FUR-BO-10003441  Furniture    Bookcases
```



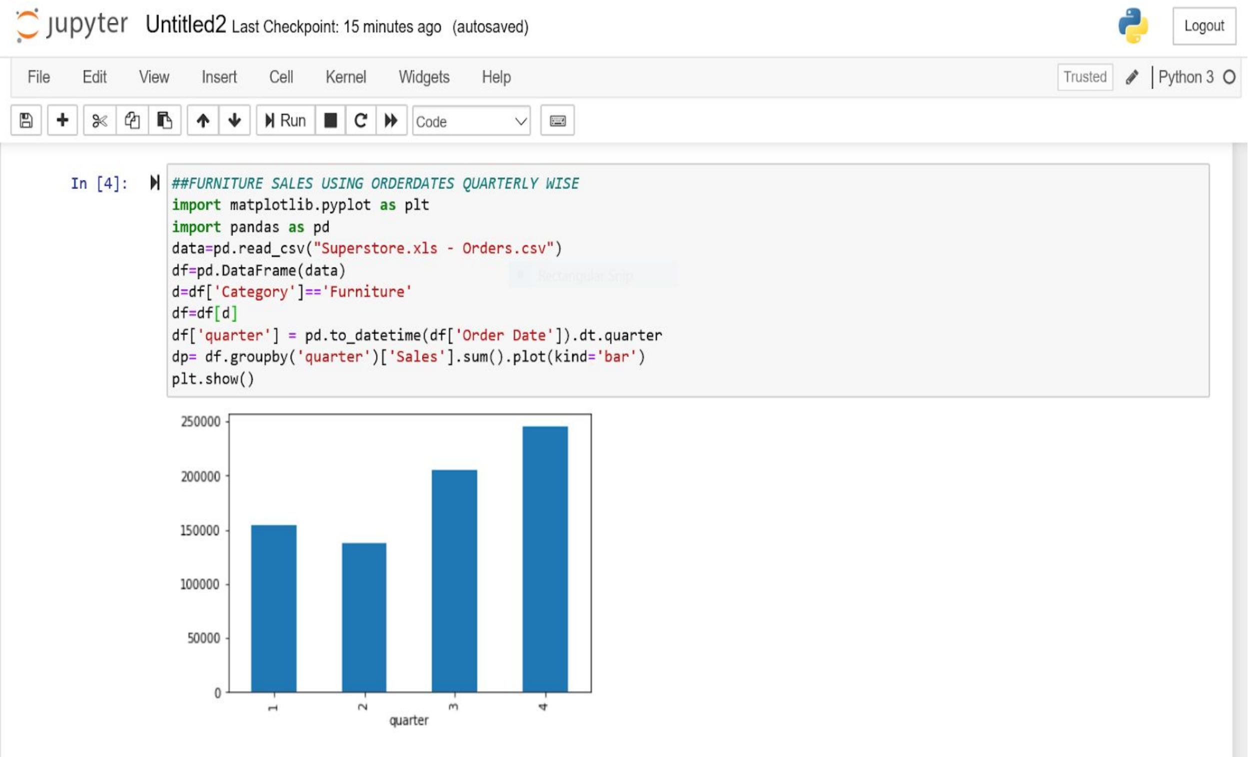
jupyter Untitled2 Last Checkpoint: 18 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
                                         Product Name    Sales  Quantity  \
7948            Harbour Creations Steel Folding Chair  362.250         6
7949           DAX Black Cherry Wood-Tone Poster Frame   63.552         6
8310        Bevis Round Conference Room Tables and Bases  376.509         3
6547                Novimex Swivel Fabric Task Chair  634.116         6
157            Global Deluxe High-Back Manager's Chair  457.568         2
...                                               ...      ...       ...
5457           Hand-Finished Solid Wood Document Frame   68.460         2
5131                         Nu-Dell Executive Frame  101.120         8
1877          Global High-Back Leather Tilter, Burgundy  393.568         4
7635       Global Leather and Oak Executive Chair, Black  300.980         1
906   Bush Westfield Collection Bookcases, Fully Ass...  323.136         4

      Discount    Profit
7948       0.3    0.0000
7949       0.6  -34.9536
8310       0.3  -43.0296
6547       0.3 -172.1172
157        0.2   51.4764
...        ...       ...
5457       0.0   20.5380
5131       0.0   37.4144
1877       0.2  -44.2764
7635       0.0   87.2842
906        0.2   12.1176

[2121 rows x 21 columns]
```



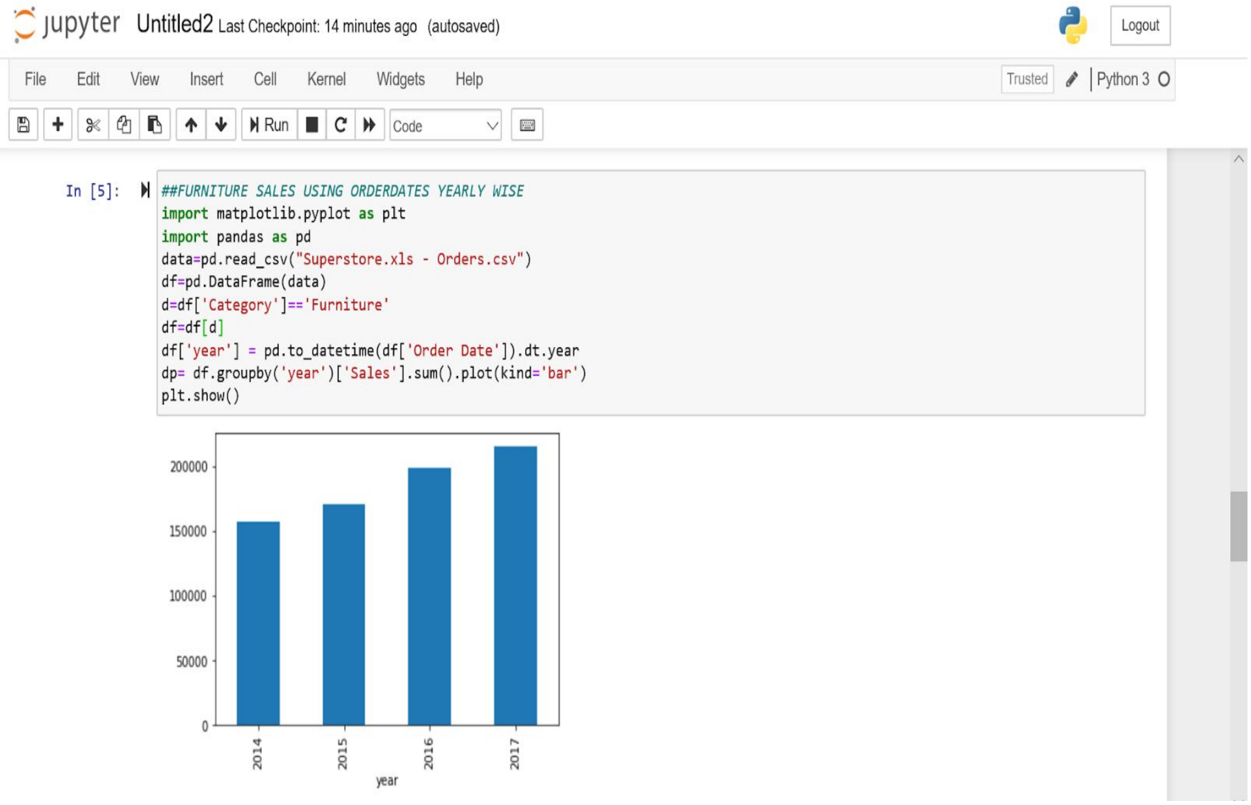
jupyter Untitled2 Last Checkpoint: 16 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [3]:
100000
80000
60000
40000
20000
0
1
2
3
4
5
6
7
8
9
10
11
12
month

```
##FURNITURE SALES USING ORDERDATES MONTHLY WISE
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['month'] = pd.to_datetime(df['Order Date']).dt.month
dp= df.groupby('month')['Sales'].sum().plot(kind='bar')
plt.show()
```



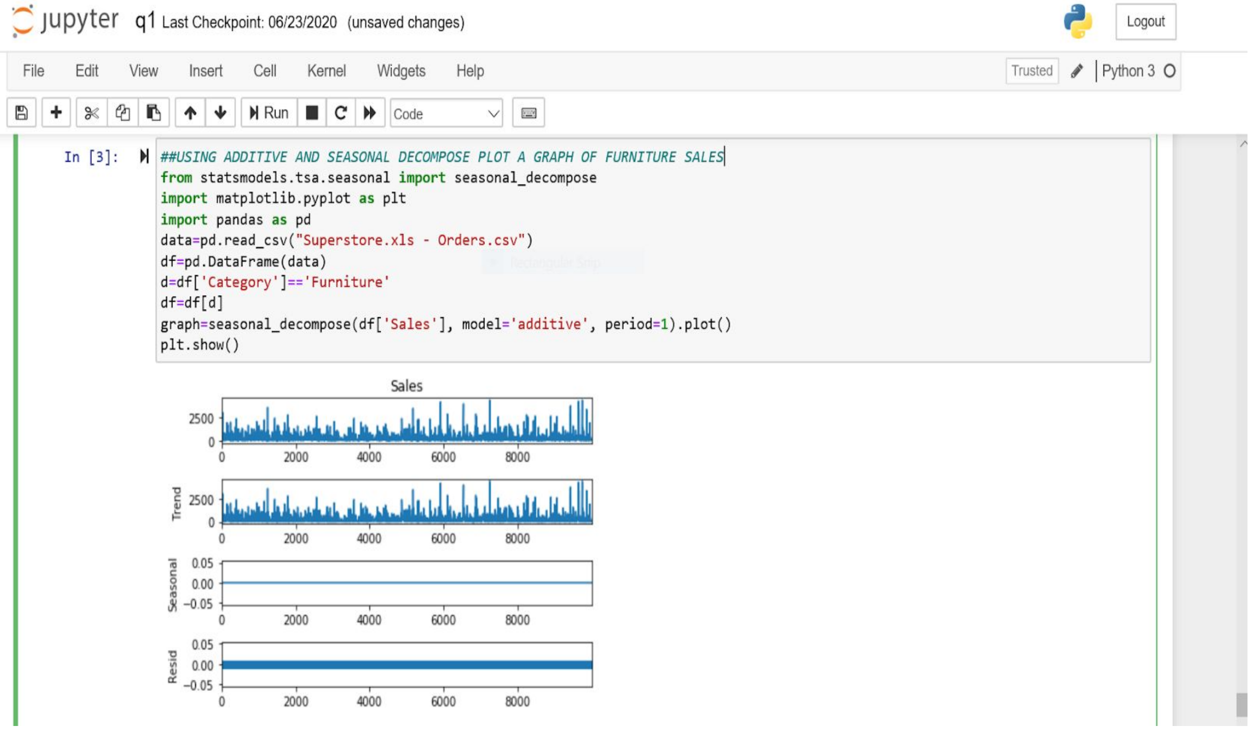
jupyter Untitled2 Last Checkpoint: 15 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [4]:
250000
200000
150000
100000
50000
0
1
2
3
4
quarter

```
##FURNITURE SALES USING ORDERDATES QUARTERLY WISE
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['quarter'] = pd.to_datetime(df['Order Date']).dt.quarter
dp= df.groupby('quarter')['Sales'].sum().plot(kind='bar')
plt.show()
```

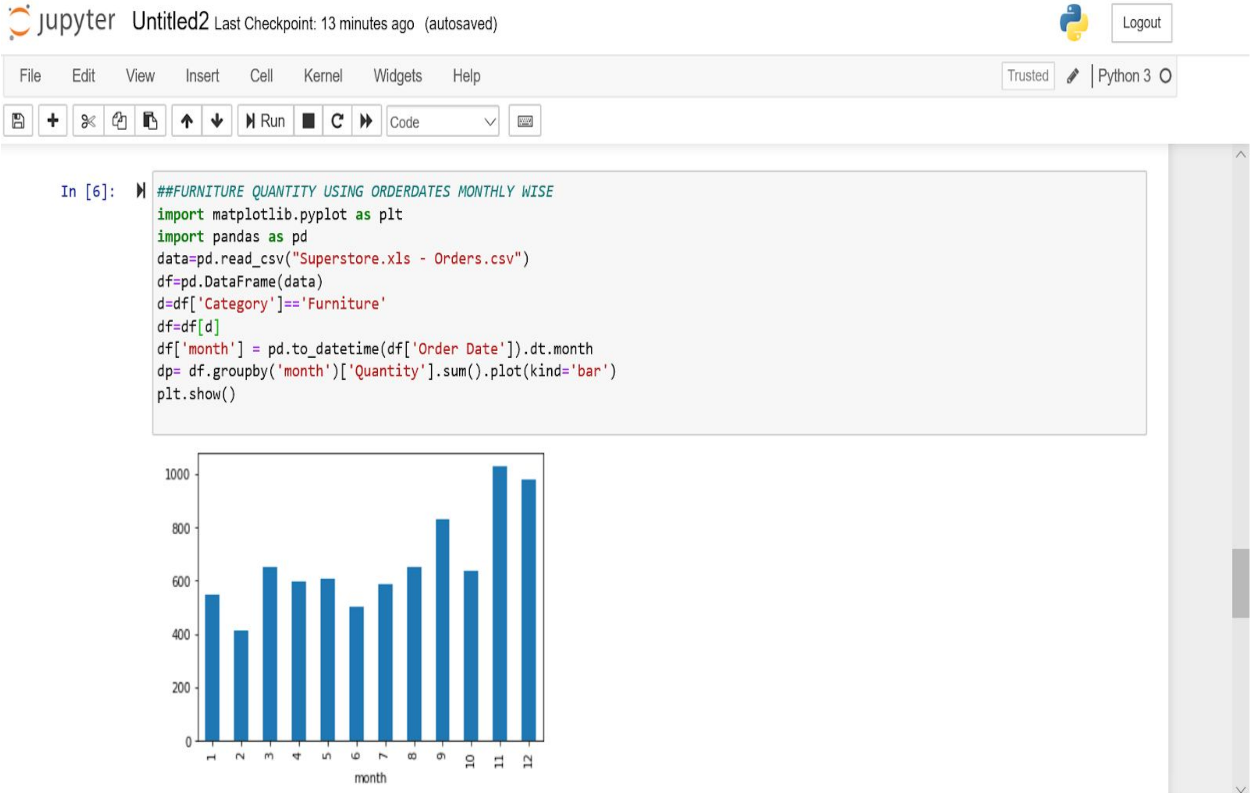


jupyter Untitled2 Last Checkpoint: 14 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [5]:
200000
150000
100000
50000
0
2014
2015
2016
2017
year

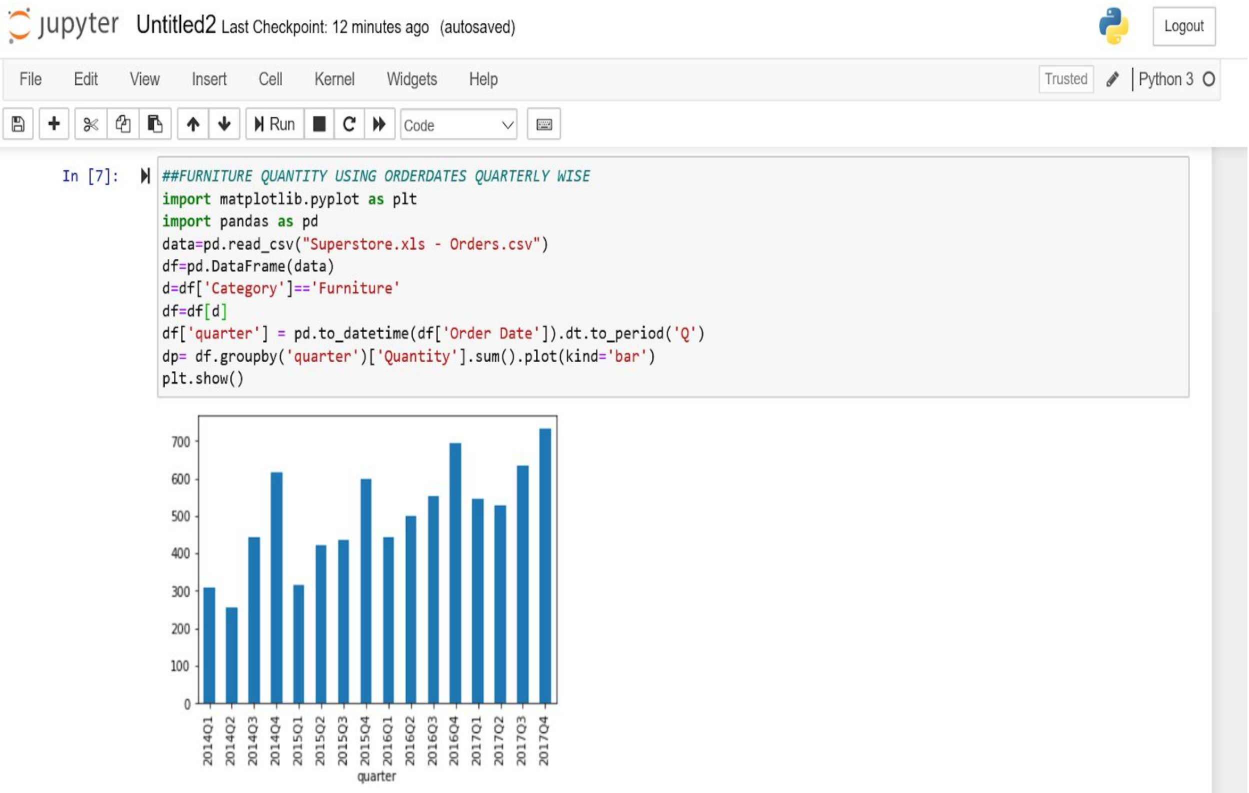
```
##FURNITURE SALES USING ORDERDATES YEARLY WISE
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['year'] = pd.to_datetime(df['Order Date']).dt.year
dp= df.groupby('year')['Sales'].sum().plot(kind='bar')
plt.show()
```


jupyter q1 Last Checkpoint: 06/23/2020 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [3]:
Sales
2500
0
2000
4000
6000
8000
Trend
Seasonal
0.05
0.00
−0.05
Resid

```
##USING ADDITIVE AND SEASONAL DECOMPOSE PLOT A GRAPH OF FURNITURE SALES
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
graph=seasonal_decompose(df['Sales'], model='additive', period=1).plot()
plt.show()
```

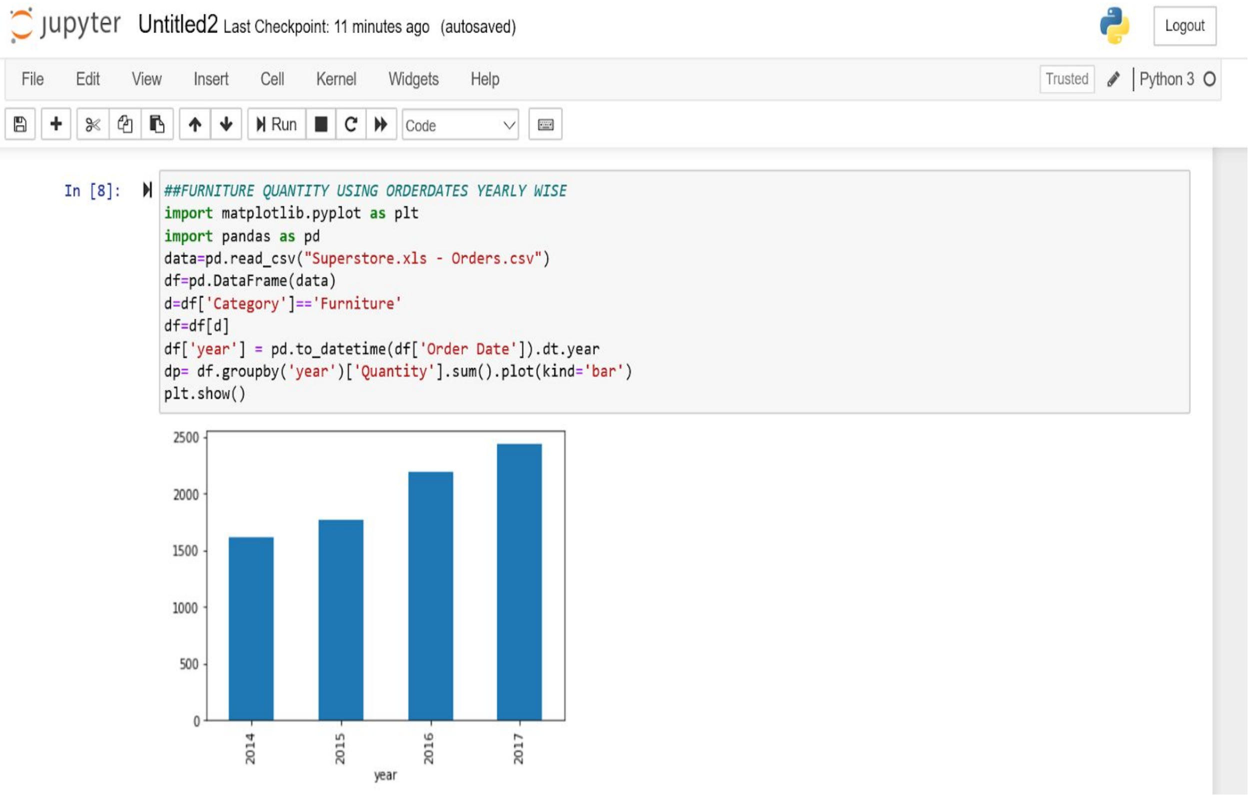

jupyter Untitled2 Last Checkpoint: 13 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [6]:
1000
800
600
400
200
0
1
2
3
4
5
6
7
8
9
10
11
12
month

```
##FURNITURE QUANTITY USING ORDERDATES MONTHLY WISE
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['month'] = pd.to_datetime(df['Order Date']).dt.month
dp= df.groupby('month')['Quantity'].sum().plot(kind='bar')
plt.show()
```



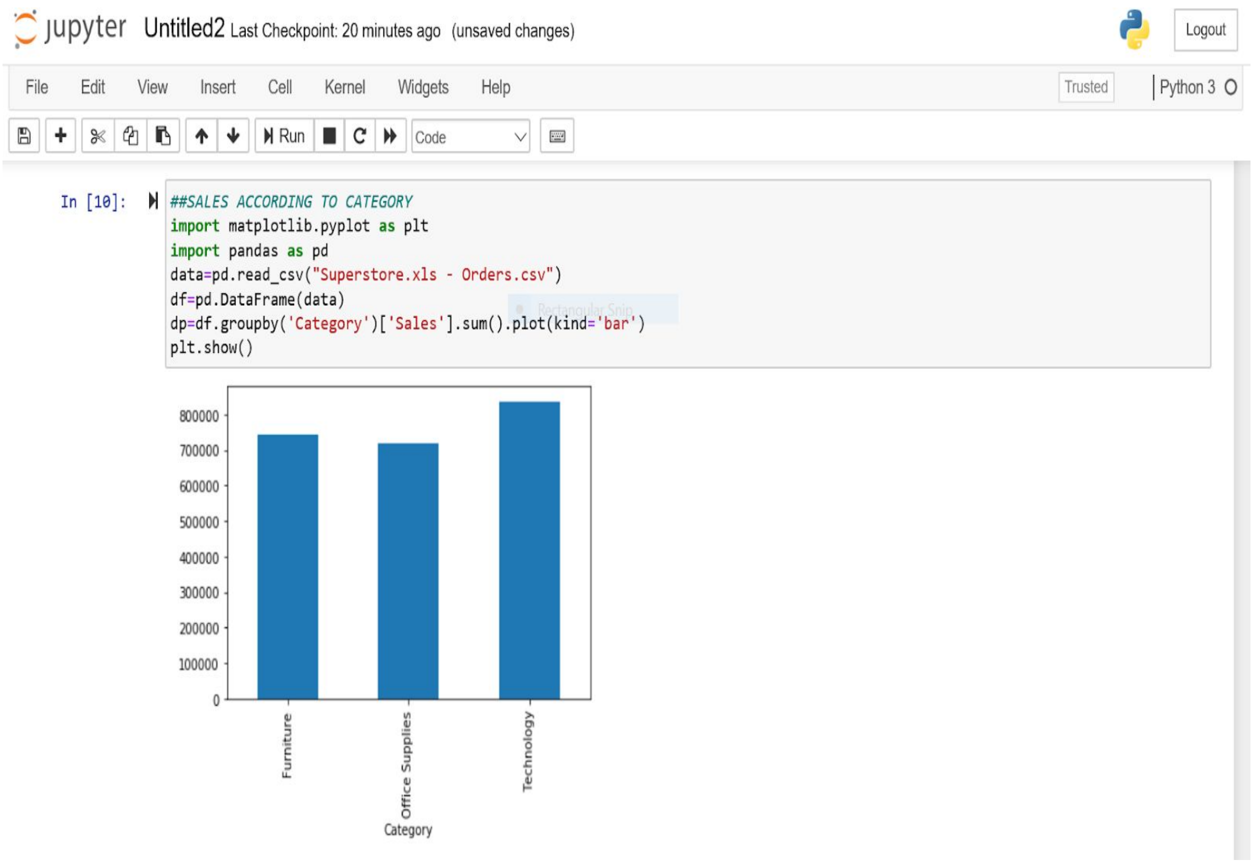
In [7]:

```
##FURNITURE QUANTITY USING ORDERDATES QUARTERLY WISE
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['quarter'] = pd.to_datetime(df['Order Date']).dt.to_period('Q')
dp= df.groupby('quarter')['Quantity'].sum().plot(kind='bar')
plt.show()
```

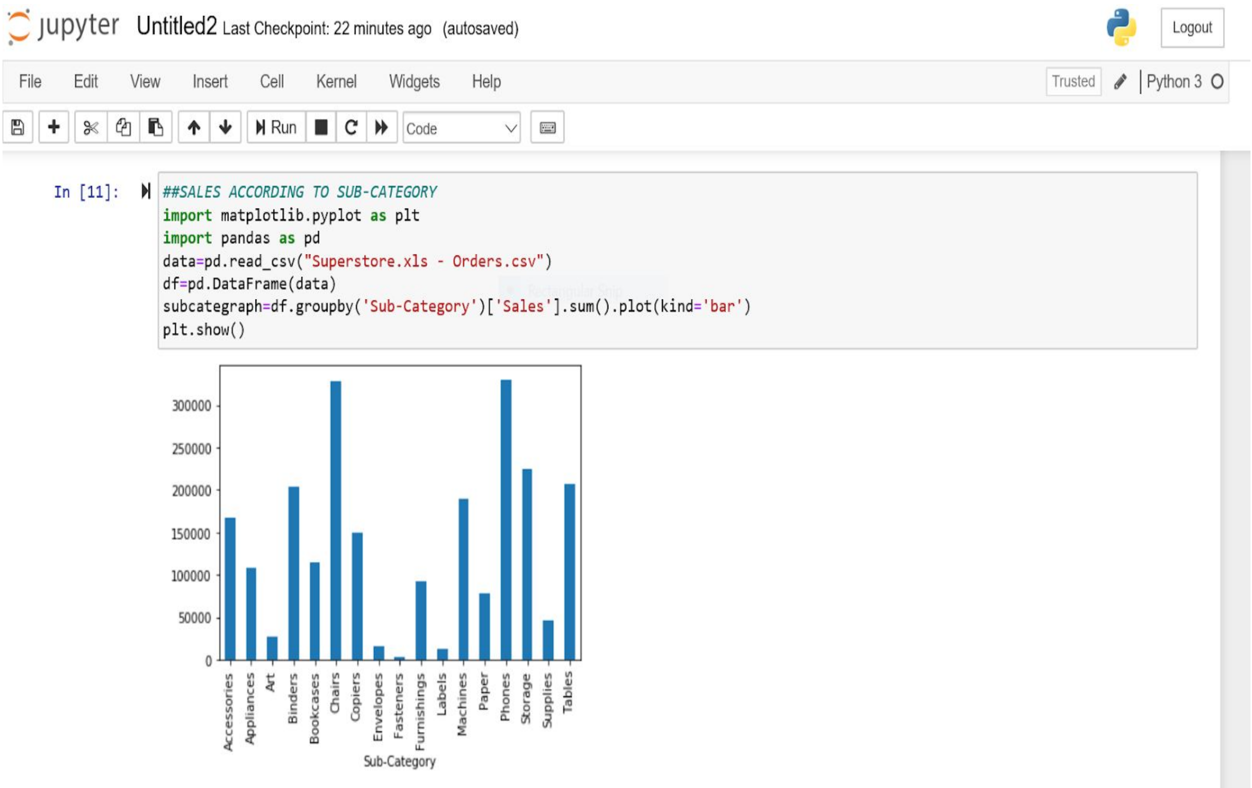


In [8]:

```
##FURNITURE QUANTITY USING ORDERDATES YEARLY WISE
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv("Superstore.xls - Orders.csv")
df=pd.DataFrame(data)
d=df['Category']=='Furniture'
df=df[d]
df['year'] = pd.to_datetime(df['Order Date']).dt.year
dp= df.groupby('year')['Quantity'].sum().plot(kind='bar')
plt.show()
```



```
In [10]:  ##SALES ACCORDING TO CATEGORY
          import matplotlib.pyplot as plt
          import pandas as pd
          data=pd.read_csv("Superstore.xls - Orders.csv")
          df=pd.DataFrame(data)
          dp=df.groupby('Category')['Sales'].sum().plot(kind='bar')
          plt.show()
```



```
In [11]:  ##SALES ACCORDING TO SUB-CATEGORY
          import matplotlib.pyplot as plt
          import pandas as pd
          data=pd.read_csv("Superstore.xls - Orders.csv")
          df=pd.DataFrame(data)
          subcategraph=df.groupby('Sub-Category')['Sales'].sum().plot(kind='bar')
          plt.show()
```

9. Conclusion

The market analysis is useful for the development of any business. In the case study given, there are some of the details of the sales according to the category of the product and the datasets needed to the analysis are downloaded and imported to the python.In this phase the training and testing phases are done while building the model.

The purpose of the Market analysis section is to describe the current state of the industry specific to your particular location and how your product or service will have relevance due to current or upcoming trends and also the current state of the industry by studying it through Monthly,Yearly,Seasonally in the quantity and quality aspects.And this market analysis section includes information about the industry, The target market, And the competition, and how you intend to make a place for your own product and service by studying the market through timeseries.In this way,We can conclude that market analysis is so important for the gradual growth of the industry.

REFERENCE

Hartpury College of Further Education, Gloucestershire-”Introduction to market analysis”-https://www.tutor2u.net/business/reference/introduction-to-market-analysis.

Harvard institute for international development - ”abstract,scopeandobjectives” - https://www.researchgate.net/publication/228150316_Marketing_Analysis_in_Project_Evaluation.

Rafael A. Irizarry, (2020) - “Introduction to Data Science” - https://rafalab.github.io/dsbook/

Microsoft World shodhganga-”Literature survey”-https://shodhganga.inflibnet.ac.in/bitstream/10603/11075/6/06_chapter2.pdf