DEPARTMENT OF COMPUTER SCIENCE
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING
AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI
NEW DELHI – 110063
AUGUST 2018

SUMMER INTERNSHIP PROJECT REPORT

AT

Delhi e-Governance Society

8th Level, B-Wing, Delhi Secretariat,
IP Estate, New Delhi 110013
Email: info@degs.org.in
Phone: 011-23392311

BY

ANSH MITTAL
00611502715

MENTOR

Santosh Kumar
Tech-Lead
Delhi e-Governance Society
Department of Information Technology
Govt. of NCT of Delhi
8th Level, B Wing, Delhi Secretariat

INDEX

CANDIDATE'S DECLARATION

It is hereby certified that the work which is being presented in the B. Tech Industrial Training Report entitled **"DATA VISUALIZER"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** and submitted in the **Department of Computer Science** of **Bharati Vidyapeeth's College of Engineering, New Delhi (Affiliated to Guru Gobind Singh Indraprastha University, Delhi)** is an authentic record of my own work carried out during the period from **June 2018 to July 2018** under the guidance of **Santosh Kumar, Tech-Lead** at **Delhi e-Governance Society** under **Department of Information Technology.**

The matter presented in the B. Tech Industrial Training Report has not been submitted by me for the award of any other degree of this or any other Institute.

ANSH MITTAL
ENROLLMENT NUMBER: 00611502715

ACKNOWLEDGEMENT

It is a great pleasure to present this report of Summer Industrial Training at **Delhi e-Governance Society** under **Department of Information Technology,** in partial fulfillment of B.tech programme under **Bharati Vidyapeeth's College of Engineering, Paschim Vihar, New Delhi** affiliated to **Guru Gobind Singh Indraprastha University, Dwarka, New Delhi.**

I express my deep gratitude to **Santosh Kumar, Tech-Lead** at **Delhi e-Governance Society** under **Department of Information Technology,** for his valuable guidance and suggestion throughout my project work. I am also thankful to **ANUSURYA,** my co-workers in the project.

At the outset, I would like to express my immense gratitude to technical team, guiding right from the inception till the successful completion of training and project.

ANSH MITTAL

ENROLLMENT NUMBER: 00611502715

ABSTRACT

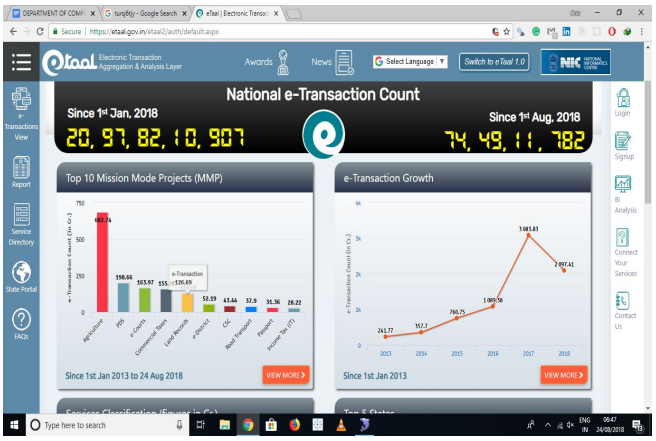
Working at DeGS(Delhi e Governance Society) gave us the opportunity to brush our skills in various fields like Data Science, Web development using asp.net and Machine Learning. It also provided us the opportunity to understand the DBMS Techniques at Industrial level. In addition to all this, we also gained some interpersonal skills.

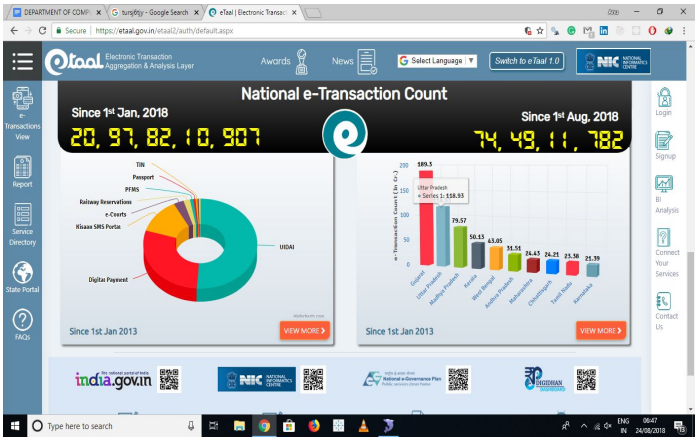
We made a tool called “Data Visualizer” which takes data directly from government database and performs analysis to display it on web which made analysis of various data easier for the a lot of departments. It required asp.net along with SQLYOG for querying and processing data and then designing of web page in order to give a proper visualization space. This tool has the feature to incorporate various kind of filters based on department and dataset type. As per current statistics, it will prove helpful for at least 80 organisations in their analysis work.

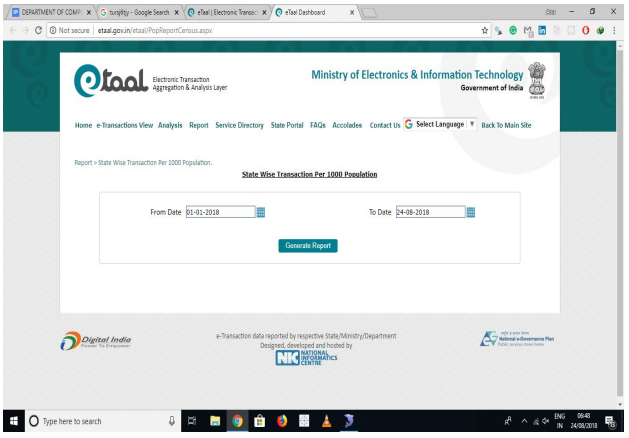
Additionally, we used RStudio and R-language for doing sentimental analysis on various data provided also using Web Scraping. Web Scraping is a technique with which we can access the data of a particular website using selector gadget and use it to form datasets for further analysis. Sentimental Analysis is type of analysis for text or any such form of information where the true sentiments can be judged based on certain scores example for positive and negative sentiments also score points for anger, joy, happiness and anticipation. Based on these scores, sentiments are calculated.

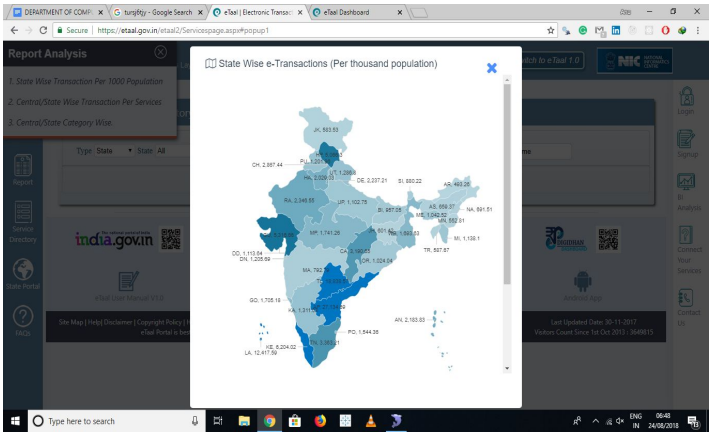
Finally, in Machine Learning we were asked to create a regressor for number of transactions under a certain services of different department. This helped us calculate the number of transactions for the data given to us. As we have mentioned earlier, We created A dummy Dataset as these values were not officially disclosable. We used certain Regressors, and also learnt the flow of Regressors in Microsoft Azure which helped us construct a flowchart which has been depicted in later pages.

ORGANIZATION'S PROFILE









About Delhi e-Governance Society

The primary objective of the Delhi e-Governance Society (DeGS) is to administer the implementation of e-Governance projects for the overall benefit of the citizens and public by setting up the necessary administrative, financial, legal and technical framework implementation mechanism and resources in the National Capital Territory of Delhi. It will facilitate the establishment of service centre through the society as an innovative way of providing public facilitation and citizen services. To provide the economic, scientific technological social and cultural development by promoting the utilization of Information Technology Computer Communication Networks, Informatics etc. To promote further development of services, technologies, infrastructure and value added computer and computer communication services it is considered expedient to provide the Society a legal entity by getting it registered under the Societies Registration Act. 1860.

FUNCTIONS OF THE SOCIETY

1. To take all necessary steps to promote efficiency, reduce delays enhance accountability, transparency and objectivity in the functioning of the Government of National Capital Territory of Delhi (for Short called "Government")

2. To assist the Department of IT in formulating and implementation policies procedure guidelines and e-Governance through various Government departments and agencies for the improvement of citizen services. To promote and disseminate Information Technology culture in the National Capital Territory of Delhi so that the common man can avail the benefit of information technology and e-Governance.

3. To administrate the implementation of e-Governance projects for ensuring use of Information Technology of masses. To lay down the necessary administrative, financial, legal and technical framework and resources for the It enabled Citizen Services.

4. List and priorities the areas for Citizen Services in consultation with the concerned Departments and take all steps for improving Citizen Services to the use of IT. To facilitate implementation of Citizen Charters framed by the other departments through the use of e-Governance and IT as a tool.

5. To workout revenue models and modalities for providing Citizen Services through the use of IT on a PPP (Public Private Partnership) model for its self-sustainability and to encourage private sector initiative in IT related infrastructure and services. This would include working out all, commercial modalities and revenue model including Franchise Model in citizen services.

6. To collect revenue and to issue receipts on behalf of the various Departments. Such receipts shall have the same legal validity as if it were an actual receipt issued by the concerned department and transfer the revenue collected to the concerned departments or to authorize a suitable authority, committee, sub-committee or society for these purposes on behalf of the e-Governance Society.

7. To establish and make available connectivity and access to Information through Internet, Intranet, LAN, MAN, WAN, E-Mail, Web Servers and Web sites, VSAT and other modes of Communication or many other form of connectivity and regulate their use in the State. To facilitate a comprehensive State Wide Area Network (SWAN) and a State Telecom Network for al the Government Departments, Institutions, and agencies.

8. To buy, sell, let on hire, repair, import, export, lease, trade and otherwise deal and provide all types of IT resources and support, including hiring of professional, consultancy services for e-Governance projects, procurement of hardware and software, development of software projects on turnkey basis.

9. To enter into collaborations, MOUs, partnerships, agreements and contracts with individuals, companies or other organizations for transfer, sale, purchase of equipment and for technical, financial or any other assistance for carrying our al or any of the objects of the Society.

10. To enter into any agreement with any Government or any corporations, companies, or persons which may seem conducive to the Societies objects or any of them and to obtain from any such Government authorities, corporations, companies, societies or persons any contracts, rights and concessions, which the society may think desirable and to carry out exercise and with any such contracts rights, privileges and concessions.

11. To hire professionals, consultants and other specialized agencies found necessary for efficient handling and conduct of the business of the society.

12. To provide and or arrange to provide all technical assistance or help a create organize, and maintain centralized data-warehouse, departmental database information repository, and software-library to be shared by all Departments, Institutions, Agencies. To facilitate inter-departmental coordination in al IT related matters and to devise modalities for information sharing so as to avoid duplication of data entry and development of software. To enable, simplify and modernize the storage, retrieval, transmission, distribution and exchange of information in electronic format. To work out procedures for data entry and its verification, validation, security, storage and updation on a regular basis.

13. All IPR of the software created in any Government Department would be jointly held by the Department of IT through the Society.

14. To workout and notify the Standard of Service and Service Level Agreements in e-Governance and IT for citizen services.

15. To take all means for the safety and security of data and to lay policies procedures guidelines and rules for achieving the objectives of security and authentication. This would include use of digital signatures and public key infrastructure.

16. To take all the steps necessary to fulfill the objectives of the Department of IT, good governance for the overall benefit of the citizens and public in the State of Delhi. To do all such other lawful things as may be necessary, incidental or conducive to the attainment of the above objects.

17. To obtain licenses, certificates and privileges for all purposes form all persons, local authorities and the Central and State Governments, to renew the same and transfer the same in favour of any person or authorities.

18. To lend or deposit moneys belonging to or entrusted to or at the disposal of the Society or franchisees and other having dealings with the Society with or without security and upon such terms as may be thought proper and to guarantee shall not carry on banking business as defined in the Banking Regulation Act 1949 (10 of 1949).

19. To borrow and raise money with or without security or to receive money and deposit on interest or otherwise in such manner as the Society may deem fit.

20. To draw, issue, accept and to endorse discount and negotiate promissory notes certificates and other negotiable or commercial or mercantile instruments connected with the business of the Society.

21. To establish and maintain any agencies and franchises in the state for the conduct of the business of the Society.

22. To apply for tender, purchase or otherwise require any contracts and concessions for or in relation to the construction, execution carrying out, equipment, improvement, management, administration or contract of improvement, management, administration or control of works and conveniences and to undertake, execute, carry out, dispose of or otherwise turn to account the same.

23. To improve, manage, work, develop, alter, exchange, lease, mortgage, turn account, abandon or otherwise deal with all or any part of the property right concessions of the Society.

24. To let out on hire all or any of the properties of the Society including every description of apparatus appliances of the Society.

25. To open account or accounts with any individual firm or company or with any bank or banks and to pay into and to withdraw moneys from such account or accounts.

26. To invest apply for and acquire, or otherwise employ moneys belonging to or entrusted to or at the disposal of the Society upon Government securities and FDRs upon such terms as may be thought proper and from time to time and vary such transactions in such manner as the Society may think fit.

27. To create any depreciation fund, reserve fund, sinking fund, insurance fund or any special or other fund whether for depreciation or for repairing, improving, extending or maintaining any of the properties of the Society ad to transfer any such fund or part thereof to any of the other funds herein mentioned.

Board of Governors

1. The Society shall have its Board of Governors as the Supreme Authority source of all powers, functions and activities.

2. The general superintendence, direction and control of the affairs of to Society and of its income and property shall be vested in the Board of Governors of the Society (hereafter referred to as "The Board" or "The Board of Governance")

Executive Committee

1. The Empowered Committee shall be empowered to take all administrative decisions where no creation of posts in involved.

2. The Executive Committee shall issue Administrative Approvals (AA), Financial Sanctions (FS) and Technical Sanction (TS) for all IT and e-Governance projects of value less that Rs. One Crores. Any project of more than Rs. 1 crores would require the approval of the Board of Governors.

3. The Executive Committee will be competent to handle all IT resource acquisition and disposal cases of value less than Rs. 1 Crore per tender. Any acquisition of more than Rs. 1 crore per tender would require the approval of the Board of Governors.

4. In case of urgency, the Executive Committee will take decisions and get the same ratified from the Board of Governors in its next meeting.

5. The Executive Committee, after approval of the Chairman, may also further delegate any of its powers to Deputy Secretary(IT), GNCTD of the Society.

6. The Executive Committee shall exercise such other powers as delegated to it by the Board of Governors.

DATASETS TO WORK UPON

We worked upon multiple SQL Databases and Datasets which were given to us by the Delhi e-Governance Society. Most of these Datasets are not disclosable and hence have not been included in this report or have partially been included as we describe our work with the prestigious organization.

We used SQLYog as it is one of the most used MySQL management tool nowadays and has been in popular use in government department in which we worked. Snaps of which are given below as a reference:

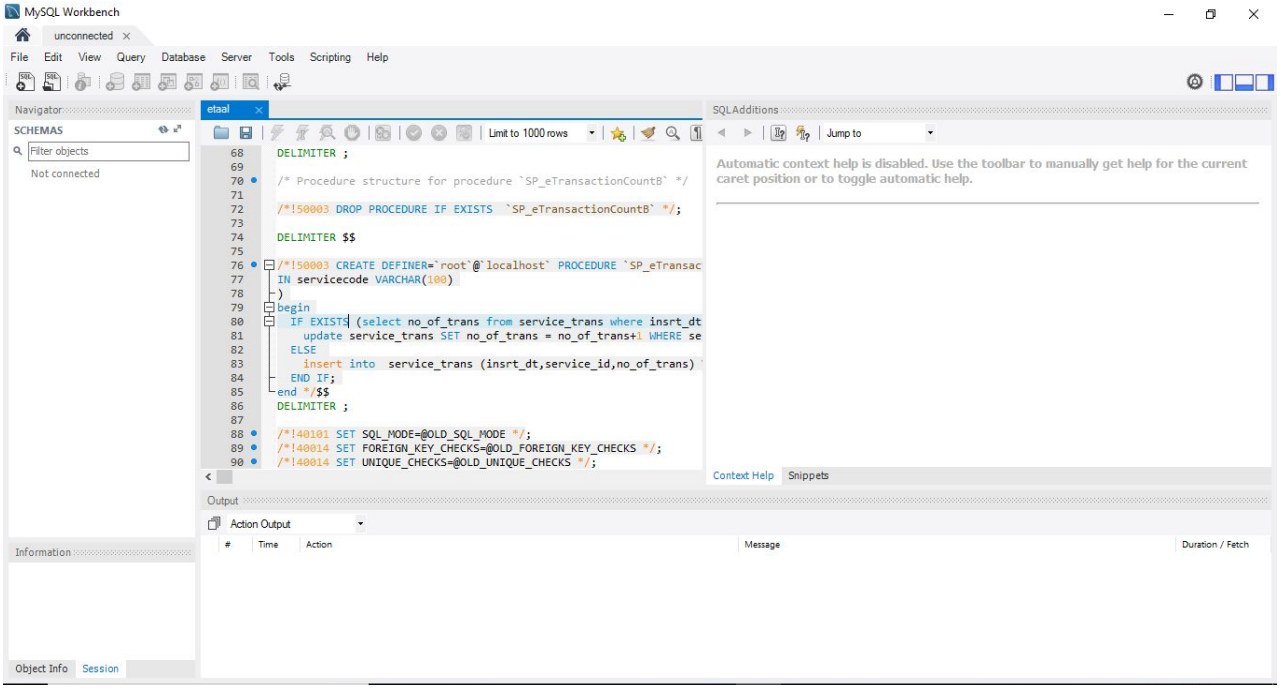
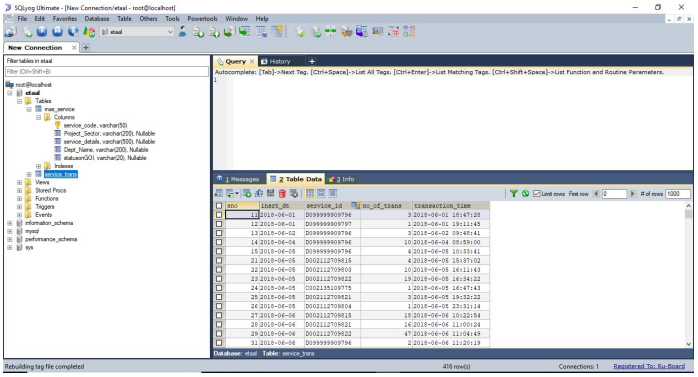
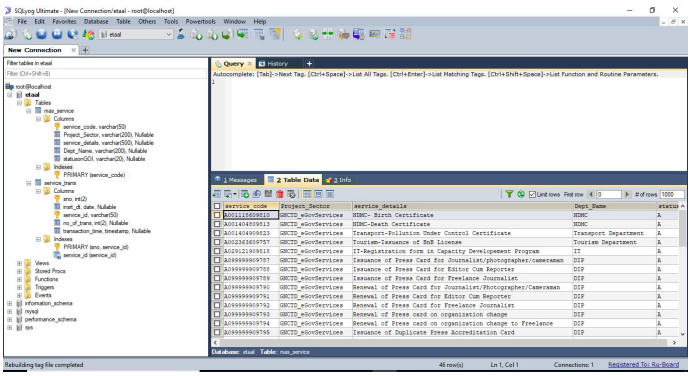
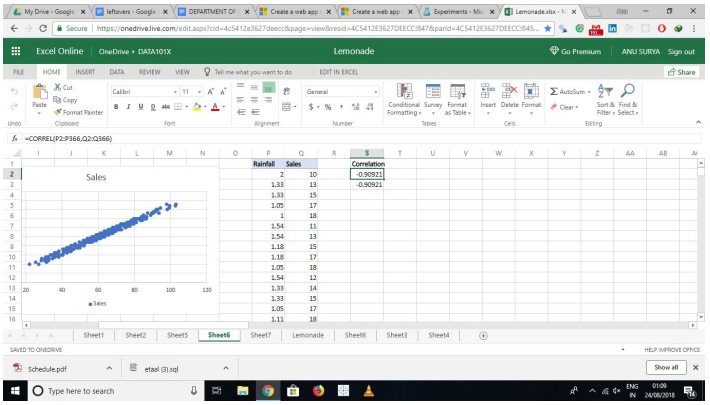


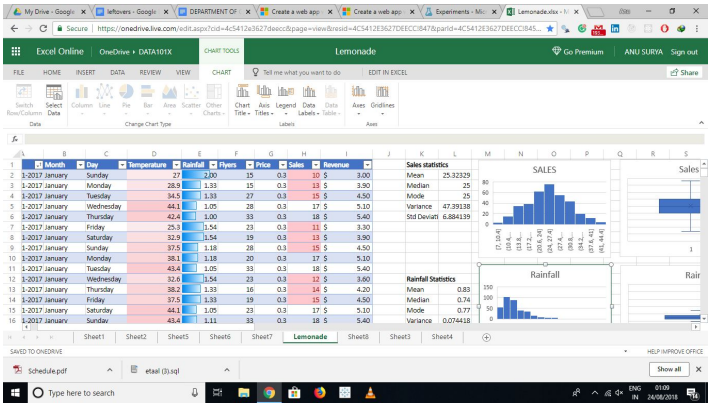
Figure 1: MySQL Window Pane having Queries for our Database that we worked upon

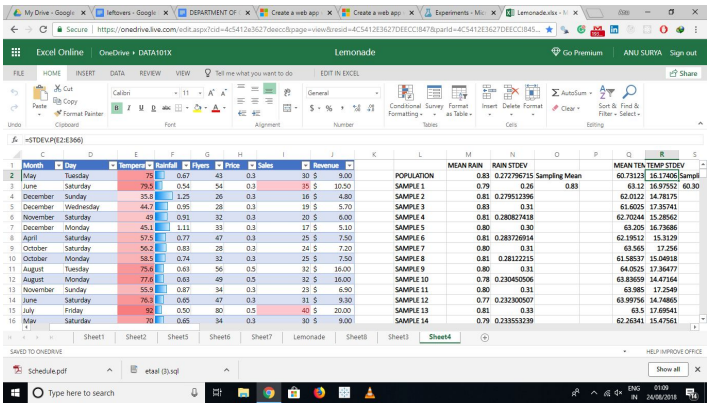


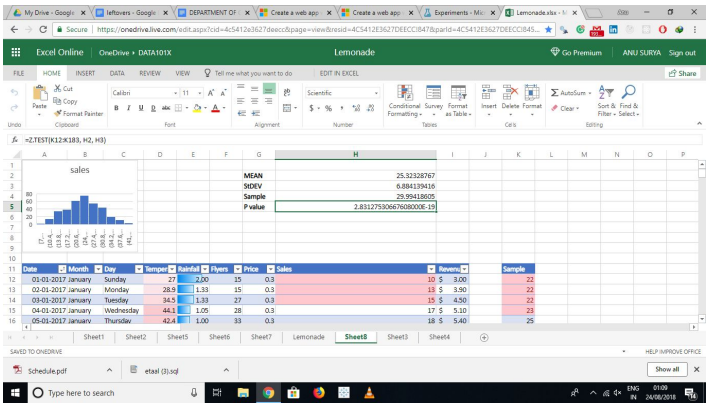


While starting our work with Machine Learning during our 6-week Internship, we did try a lot of platforms for getting comfortable with the new work environment, and also stumbled upon Microsoft Azure which helped us understand the flow of the Machine Learning Algorithm on this yet another Microsoft Platform. The below mentioned images are some of the Datasets that we used along with some of the Analysis that helped us get through the flow of the Machine Learning Models that were Implemented on Microsoft Azure Platform.









ML DataSet

DOMAIN OF INPUTS

There were various tables in the database. These fields have been used as an input and were a part of sample database that has been shared over here. Inputs were the various fields in the table mentioned below:

Field	Type
service_code	varchar(50) NOT NULL
Project_Sector	varchar(200) NULL
service_details	varchar(500) NULL
Dept_Name	varchar(200) NULL
statusonGOI	varchar(20) NULL

We considered service_code as our primary key.

Indexes	Columns	Index Type
PRIMARY	**service_code**	**Unique**

These are the entities from a similar table which was subjected to various operations during our coding and hence we could essentially accomplish our goal of creating a ASP.NET Web Form.

Field	Type
sno	int(2) NOT NULL
insrt_dt	date NULL
service_id	varchar(50) NOT NULL
no_of_trans	int(2) NULL
transaction_time	timestamp NULL

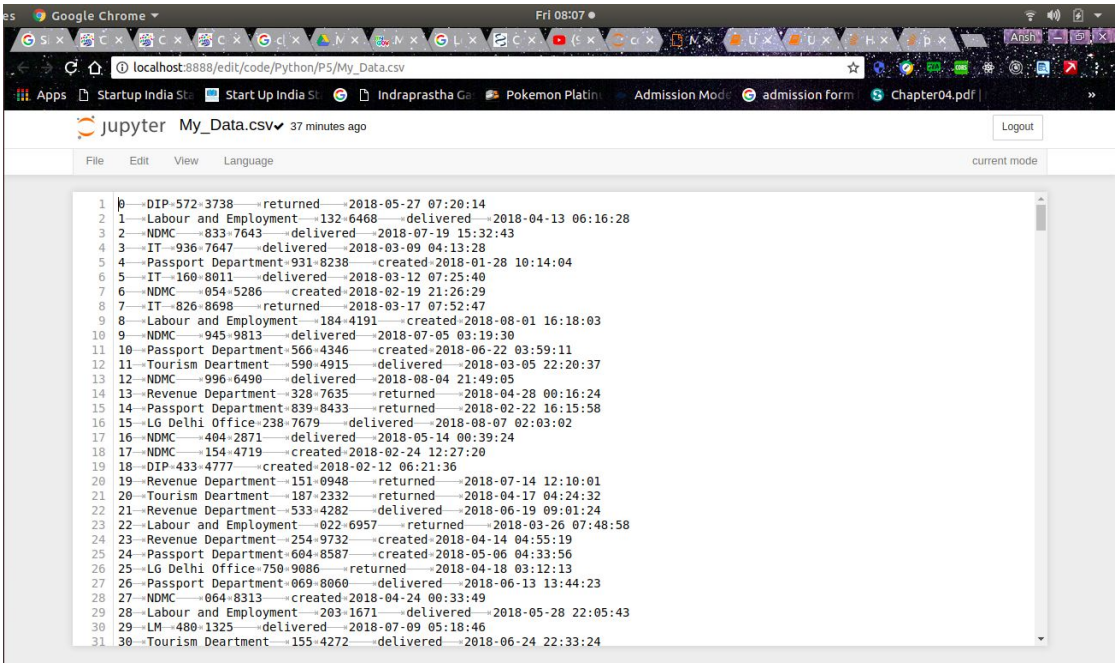
As discussed in the last table, we have used Primary Key which has been considered here as sno and service_id as they belong to unique index type.

Indexes	Columns	Index Type
PRIMARY	sno, service_id	Unique

Also, for sentimental analysis we used various government sites and operated on then with the help of web scrapping. https://www.glassdoor.co.in/Reviews/Government-of-India-Reviews-E335550.htm

For Machine Learning, as we weren't allowed to disclose the actual dataset we created a dummy dataset to show our algorithm (for prototyping) worked over there:

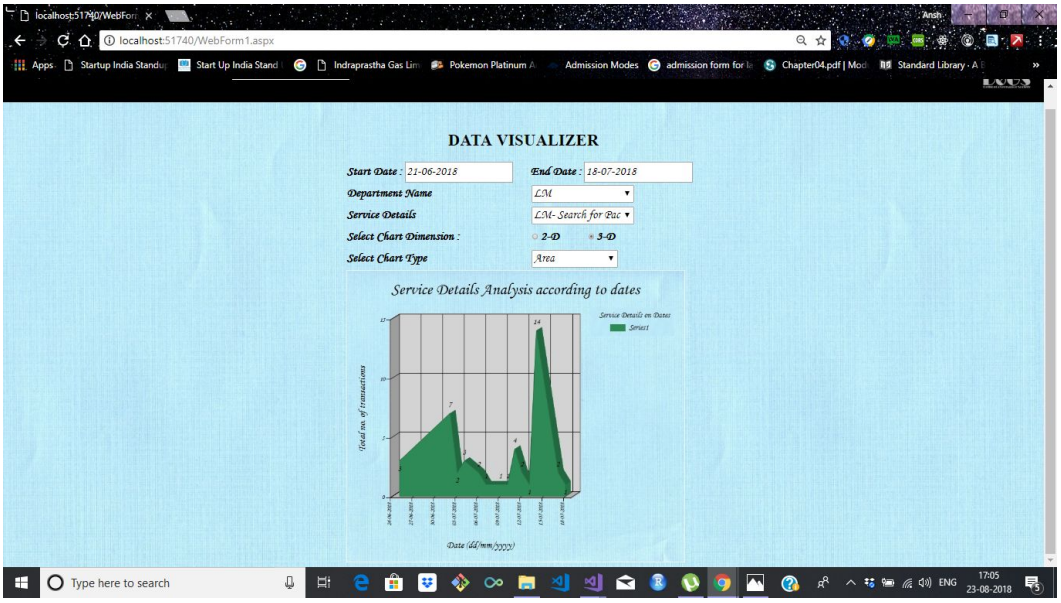
```
In [17]: import pandas
         from faker import Factory
         import random
         from datetime import datetime
         import csv

         faker = Factory.create()
         status = 'created,delivered,returned'.split(',')
         Dept_name = 'LM,Labour and Employment,Revenue Department,Tourism Deartment,NDMC,DIP,IT,Passport Department,LG Delhi

         def date_between(d1, d2):
             f = '%b%d-%Y'
             return faker.date_time_between_dates(datetime.strptime(d1, f), datetime.strptime(d2, f))


         def fakerecord():
             return {'Number of People visited': faker.numerify('####'),
                     'insrt_dt': date_between('jan01-2018', 'aug20-2018'),
                     'final_transaction status_status': random.choice(status),
                     'Department Name': random.choice(Dept_name),
                     'No_of_trans':faker.numerify('###')
                     }

         example_dummy_data = pandas.DataFrame([fakerecord() for _ in range(1000)])

         print(example_dummy_data)

         example_dummy_data.to_csv('My_Data.csv', sep='\t')
```



EXPECTED OUTPUT

The expected output to the Data Visualizer that we created is something that will be given below. This Expected Output consists of various field which are taken for input pertaining to the different domains on which work was undertaken.

ASP.NET Output :



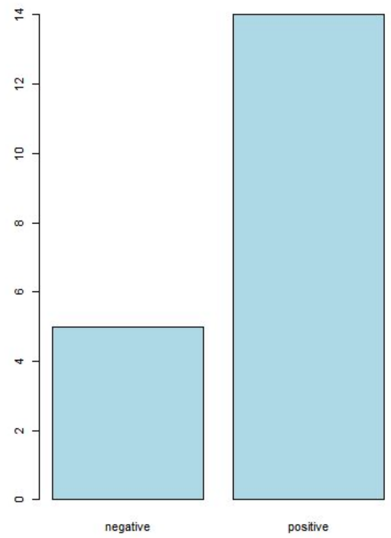
ML Output :

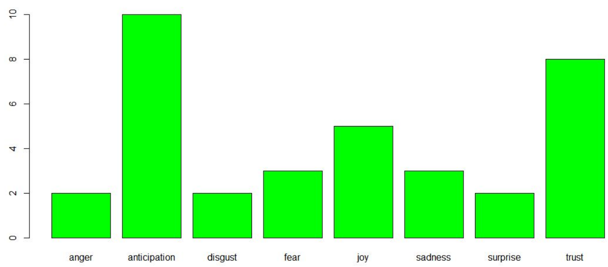
```
# Avoiding the Dummy Variable Trap
X = X[:, 1:]

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)"""

# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)


print(y_pred)
```

```
[103015.20159796 132582.27760815 132447.73845175  71976.09851258
 178537.48221056 116161.24230166  67851.69209676  98791.73374687
 113969.43533013 167921.06569551]
```

R Outputs :


government
india
whatsappshare
working
response
emailcopy
ago
flag
via
time
pdti
linklink
full
copied
years
edit
delete
twittershare
facebookshare
seconds
inappropriateflag
inappropriatehelpful
outlookapproves
share
ceoi
new
work
good
salary
organization
timeprosgreat
securityshare
managementgreat
structure
power
hierarchy
augumentation
revenueconsno
servicecomfort
support
nothing
pressure
systemstabilityjustice
recommendspositive
timeprosvery

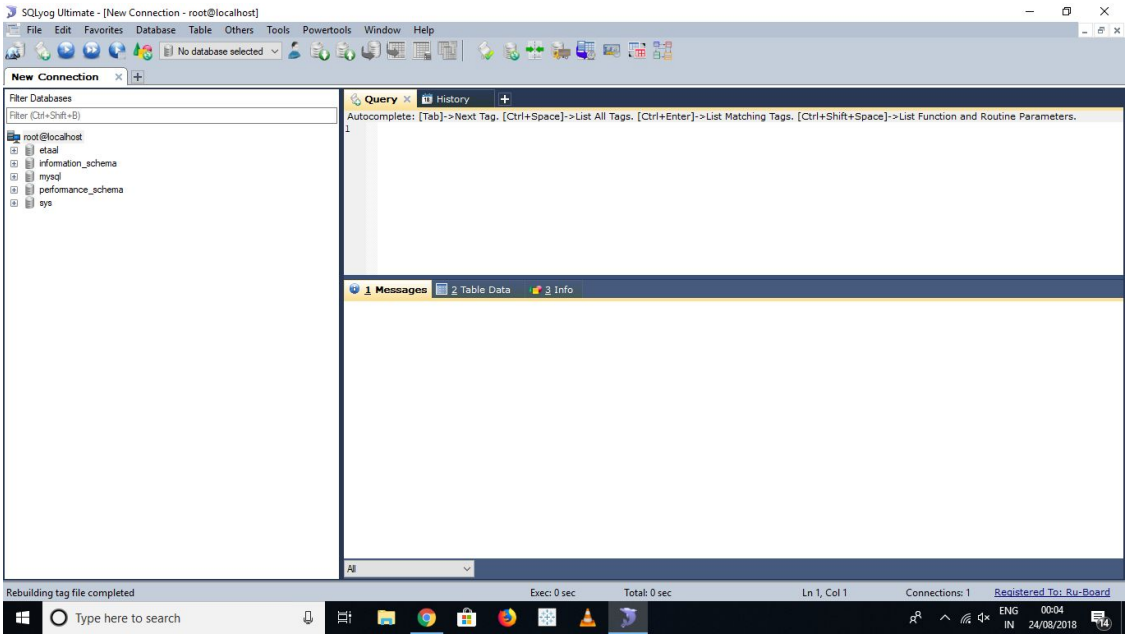

0
2
4
6
8
10
12
14
negative
positive


0
2
4
6
8
10
anger
anticipation
disgust
fear
joy
sadness
surprise
trust

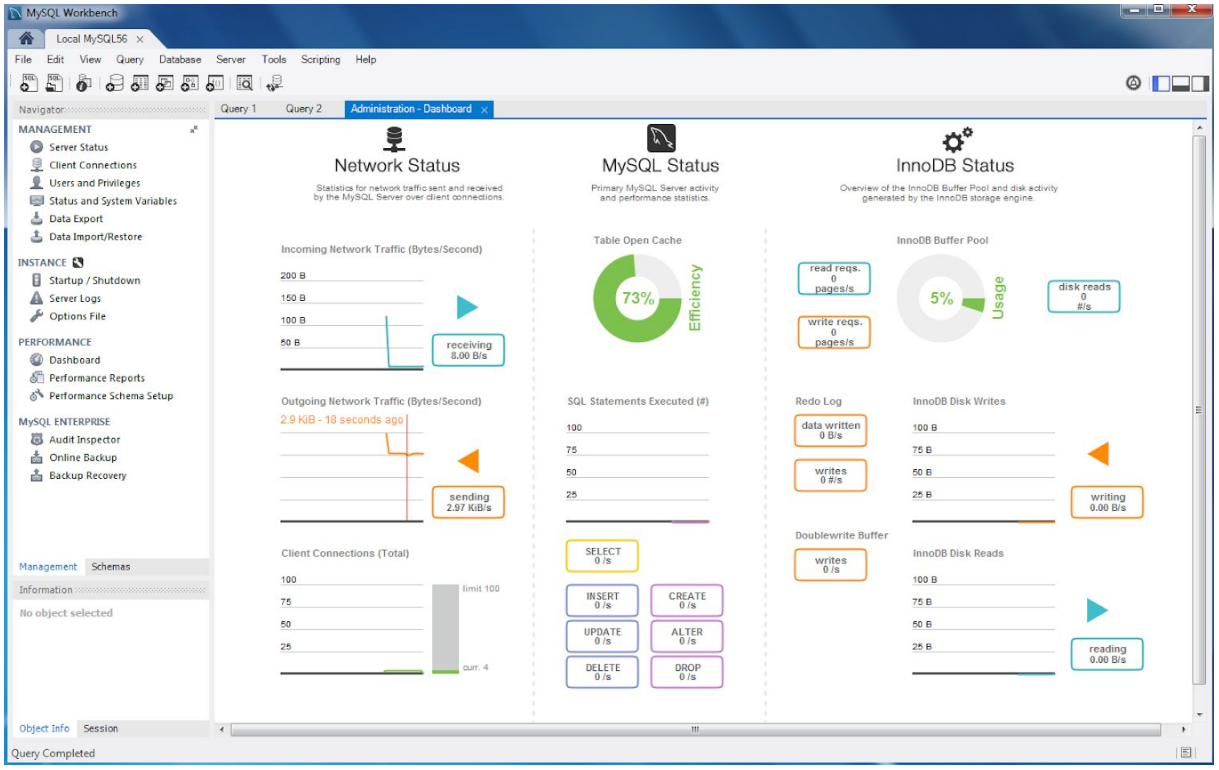
BRIEF DESCRIPTION OF PLATFORMS USED

1. For our Data Visualizer, we have used certain softwares which have been mentioned below:
    a. For our web development, we used the platform of Visual Studio Which has been discussed in detail along with its Installation and many of its features which helped us throughout our work. **Visual Studio** is one of the most renowned platform developed by Microsoft for working with ASP.NET using C# and VB.NET. This platform has been used for many years and still its utility remains as of now.

    b. **SQLyog** is a GUI tool for the RDBMS MySQL. It is developed by Webyog, Inc. based in Bangalore, India and Santa Clara California. SQLyog is being used by more than 30,000 customers worldwide and has been downloaded more than 2,000,000 times. SQLyog v0.9 was first released to the public in 2001 as after eight months of development. SQLyog was available free of charge, but with closed source code, until v3.0 when it was made a fully commercial software. Nowadays SQLyog is distributed both as free software as well as several paid, proprietary, versions. The free software version is known as Community Edition at GitHub. Paid versions are sold as Professional, Enterprise and Ultimate Editions.
        i. Programmed and developed in C++ using Win32 API . No dependencies on runtimes (.NET, Java etc.).
        ii. Uses MySQL C API to communicate with MySQL servers. No dependencies on 'database abstraction layers' (like ODBC/JDBC).
        iii. Uses SQLite to store internal data like Grid settings. Consequently, these settings are persistent across sessions on a per-table basis.



c. **MySQL** is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.
MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.



2. Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

OVERVIEW

Anaconda distribution comes with more than 1,000 data packages as well as the Conda package and virtual environment manager, called Anaconda Navigator, so it eliminates the need to learn to install each library independently.The open source data packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together.

You can also make your own custom packages using the conda build command, and you can share them with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.6. However, you can create new environments that include any version of Python packaged with conda

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 and Linux.

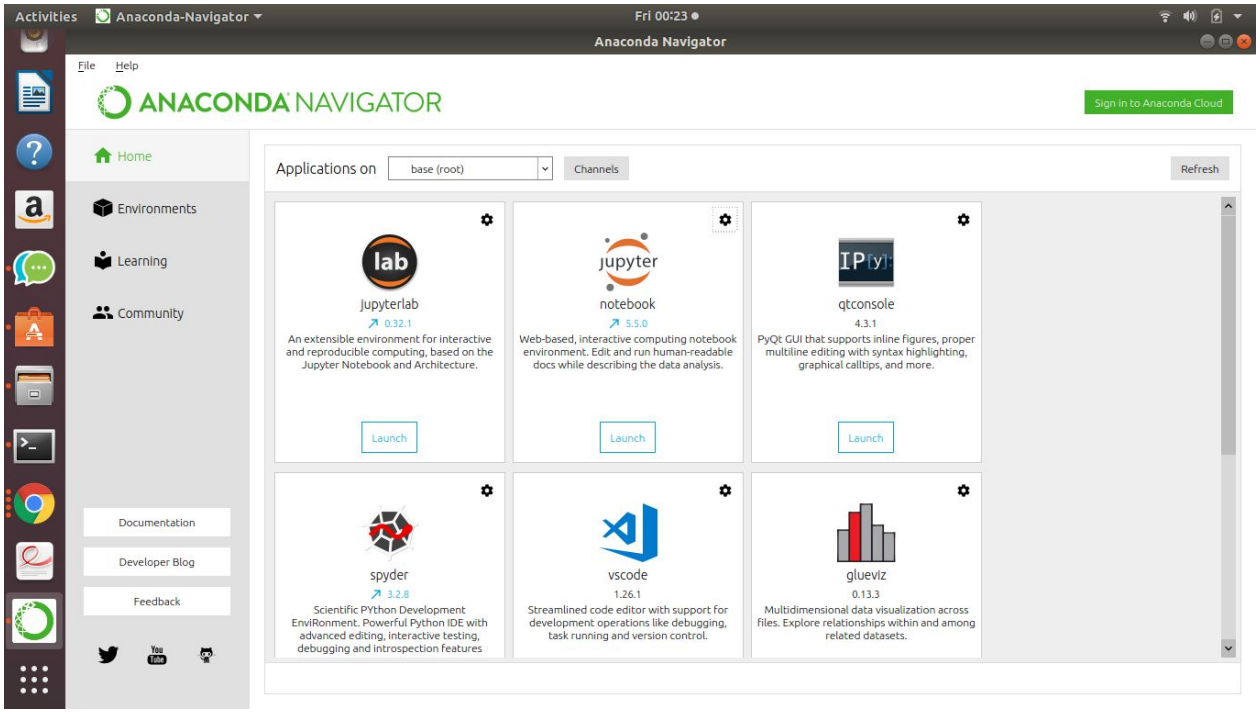
Navigator is automatically included with Anaconda version 4.0.0 or higher.

The following applications are available by default in Navigator:

1. JupyterLab
2. Jupyter Notebook
3. QtConsole
4. Spyder
5. Glueviz
6. Orange
7. Rstudio
8. Visual Studio Code

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

Anaconda Cloud is a package management service by Anaconda where you can find, access, store and share public and private notebooks, environments, and conda and PyPI packages. Cloud hosts useful Python packages, notebooks and environments for a wide variety of applications. You do not need to log in or to have a Cloud account, to search for public packages, download and install them. You can build new packages using the Anaconda Client command line interface (CLI), then manually or automatically upload the packages to Cloud.



3. **RStudio** is a free and open-source **integrated development environment (IDE) for R**, a programming language for statistical computing and graphics. RStudio was founded by JJ Allaire, creator of the programming language ColdFusion. Hadley Wickham is the Chief Scientist at RStudio. It is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server. Prepackaged distributions of RStudio Desktop are available for Windows, macOS, and Linux. It is available in open source and commercial editions and runs on the desktop (Windows, macOS, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian, Ubuntu, Red Hat Linux, CentOS, openSUSE and SLES).
RStudio is written in the C++ programming language and uses the Qt framework for its graphical user interface. Work on RStudio started around December 2010, and the first public beta version (v0.92) was officially announced in February 2011. Version 1.0 was released on 1 November 2016. Version 1.1 was released on 9 October 2017. In April 2018 it was announced RStudio will be providing operational and infrastructure support for Ursa Labs. Ursa Labs will focus on building a new data science runtime powered by Apache Arrow.

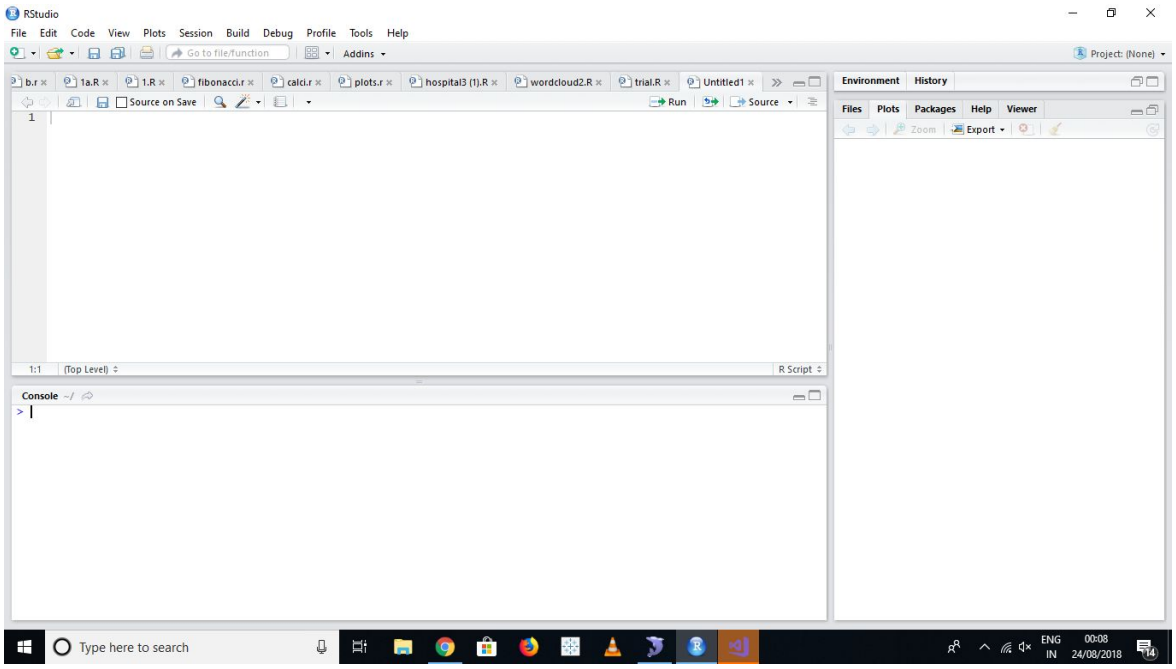


TABLE OF CONTENTS I
(for ASP.NET Web Development)

ABOUT VISUAL STUDIO

Welcome to the Visual Studio IDE

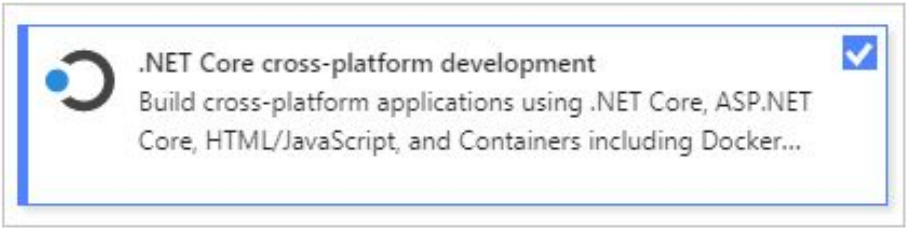
The Visual Studio integrated development environment is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. An integrated development environment (IDE) is a feature-rich program that can be used for many aspects of software development. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphic designers, and many more features to ease the software development process.

Visual Studio is available for Windows and Mac. Visual Studio for Mac has many of the same features as Visual Studio 2017, and is optimized for developing cross-platform and mobile apps.

This overview article focuses on Visual Studio 2017 for Windows. It introduces you to the basic features of the IDE. We'll walk through some things you can do with Visual Studio, including creating a simple project, using IntelliSense as a coding aid, and debugging an app to see the value of a variable during the program's execution. We'll also take a tour of the various tool windows.

Install the Visual Studio IDE

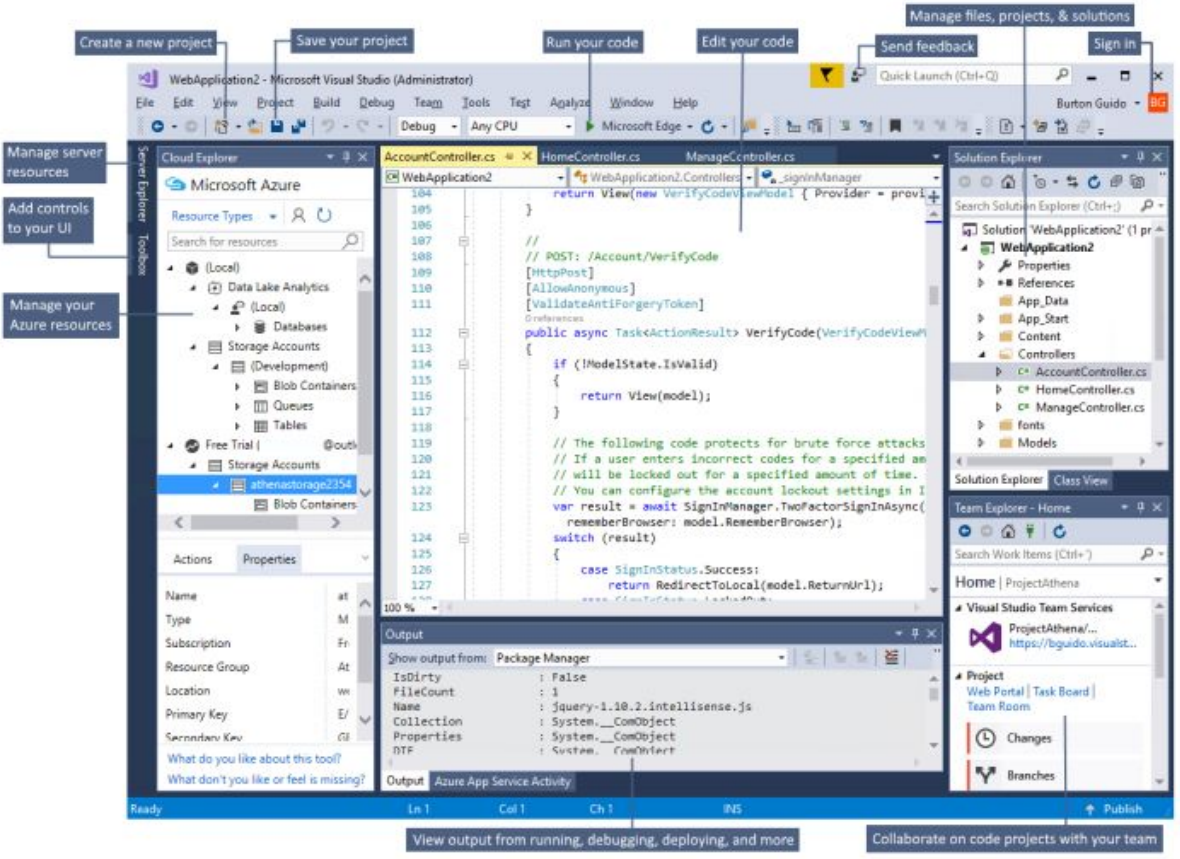
To get started, download Visual Studio 2017 and install it on your system.
The modular installer enables you to choose and install workloads, which are groups of features needed for the programming language or platform you prefer. To follow the steps for creating a program, be sure to select the .NET Core cross-platform development workload during installation.



When you start Visual Studio for the first time, you can optionally sign in using your Microsoft account, or your work or school account.

Tour of the IDE

To give you a high-level visual overview of Visual Studio, the following image shows Visual Studio with an open project and several key tool windows you'll likely use:

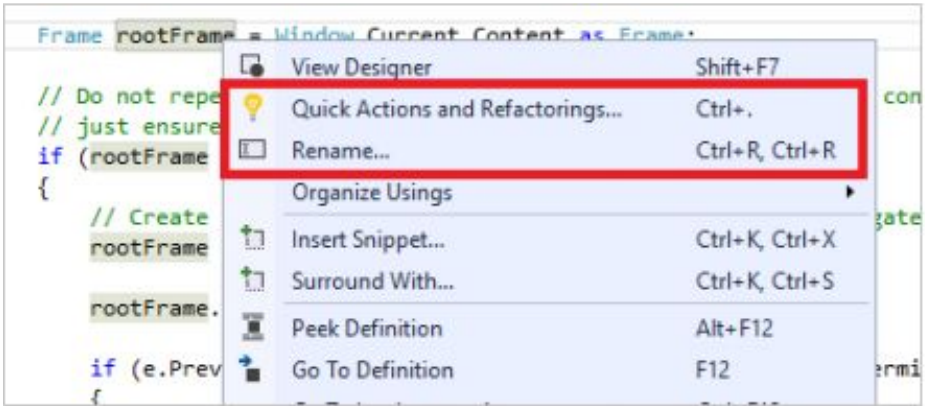


- Solution Explorer (top right) lets you view, navigate, and manage your code files. Solution Explorer can help organize your code by grouping the files into solutions and projects.
- The editor window (center), where you'll likely spend a majority of your time, displays file contents. This is where you can edit code or design a user interface such as a window with buttons and text boxes.
- The Output window (bottom center) is where Visual Studio sends notifications such as debugging and error messages, compiler warnings, publishing status messages, and more. Each message source has its own tab.
- Team Explorer (bottom right) lets you track work items and share code with others using version control technologies such as Git and Team Foundation Version Control (TFVC).

Popular productivity features

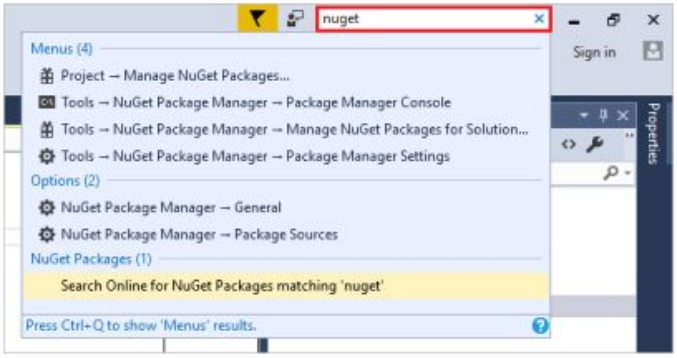
Some of the popular features in Visual Studio that help you to be more productive as you develop software include:

- **Refactoring**
  Refactoring includes operations such as intelligent renaming of variables, extracting one or more lines of code into a new method, changing the order of method parameters, and much more.



- **IntelliSense**
  IntelliSense is a term for a set of features that displays information about your code directly in the editor and, in some cases, write small bits of code for you. It's like having basic documentation inline in the editor, which saves you from having to look up type information elsewhere. IntelliSense features vary by language. For more information, see C# IntelliSense, Visual C++ IntelliSense, JavaScript IntelliSense, and
  Visual Basic IntelliSense.The following illustration shows how IntelliSense displays a member list for a type:

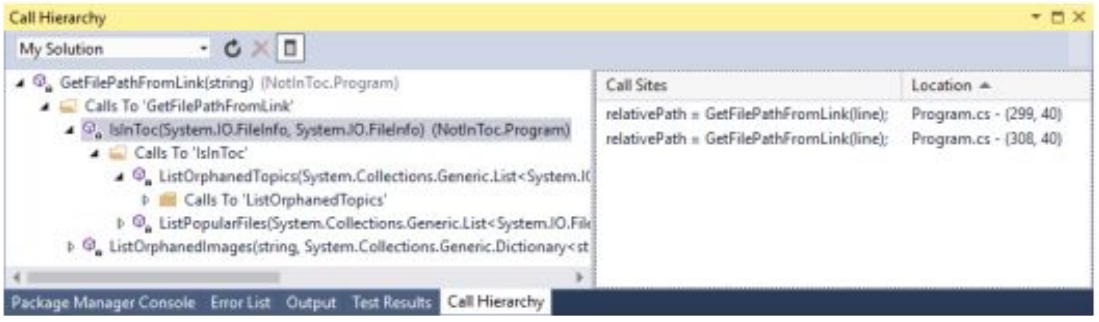
- **Quick Launch**
  Visual Studio can seem overwhelming at times with so many menus, options,and properties.The Quick Launch search box is a great way to rapidly find what you need in Visual Studio. When you start typing the name of something you're looking for, Visual Studio lists results that take you exactly where you need to go. If you need to add functionality to Visual Studio, for example to add support for an additional programming language, Quick Launch provides results that open Visual Studio Installer to install a workload or individual component.



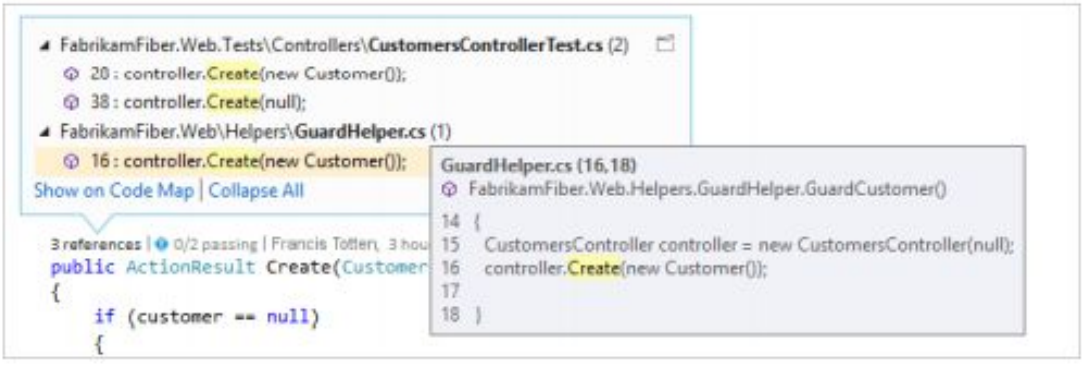
- **Squiggles and Quick Actions**

Squiggles are wavy underlines that alert you to errors or potential problems in your code as you type.These visual clues enable you to fix problems immediately without waiting for the error to be discovered during build or when you run the program. If you hover over a squiggle, you see additional information about the error. A light bulb may also appear in the left margin with actions,known as Quick Actions, to fix the error.

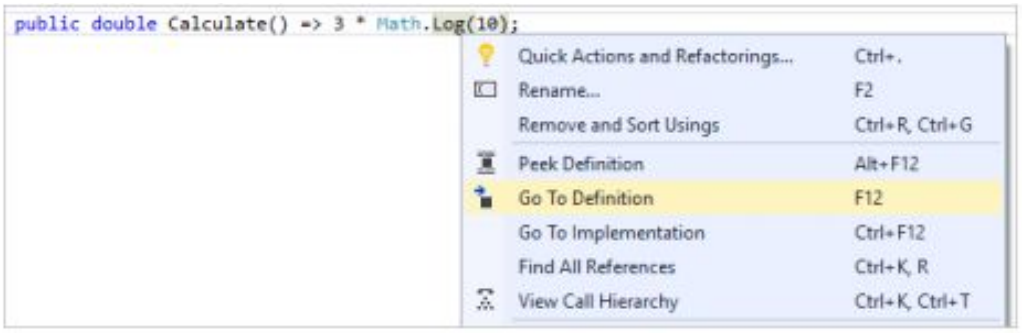
- **Call Hierarchy**
  The Call Hierarchy window shows the methods that call a selected method.This can be useful information when you're thinking about changing or removing the method, or when you're trying to track down a bug.



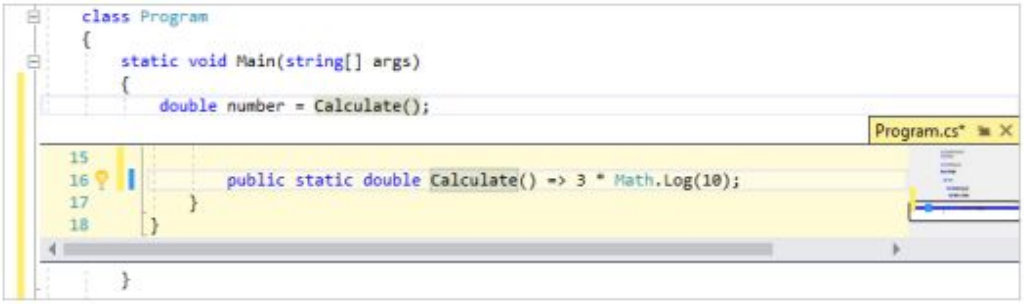
- **CodeLens**
  CodeLens helps you find references to your code,changes to your code, linked bugs, work items, code reviews, and unit tests,all without leaving the editor.



- **Go To Definition**
  The Go To Definition feature takes you directly to the location where a function or type is defined.



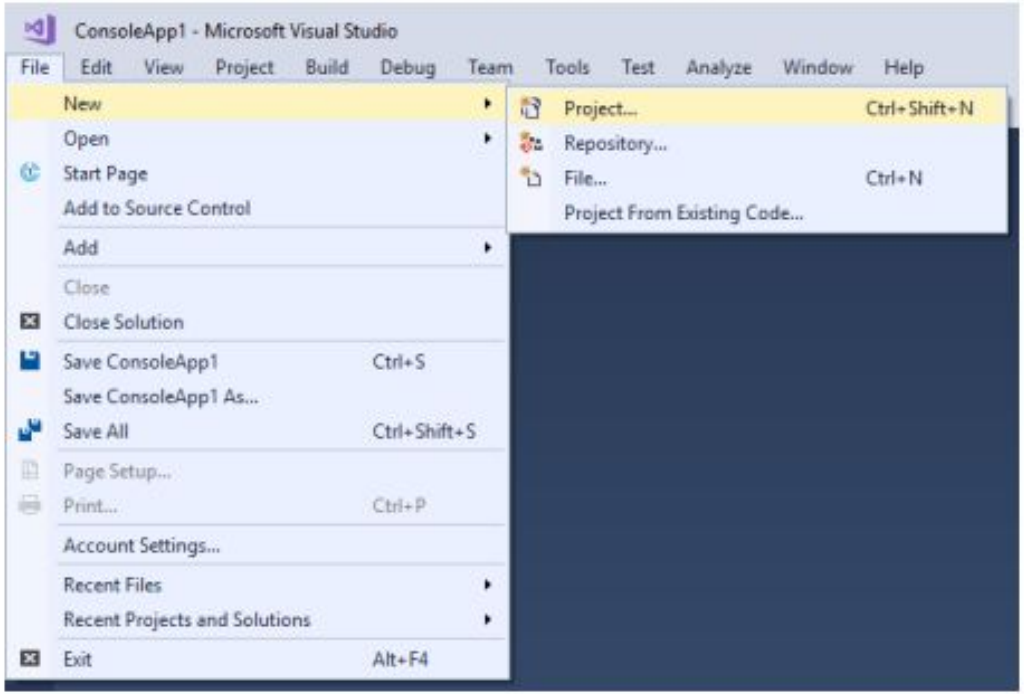
- **Peek To Definition**
  The Peek Definition window shows the definition of a method or type without actually opening a separate file.



Create a program

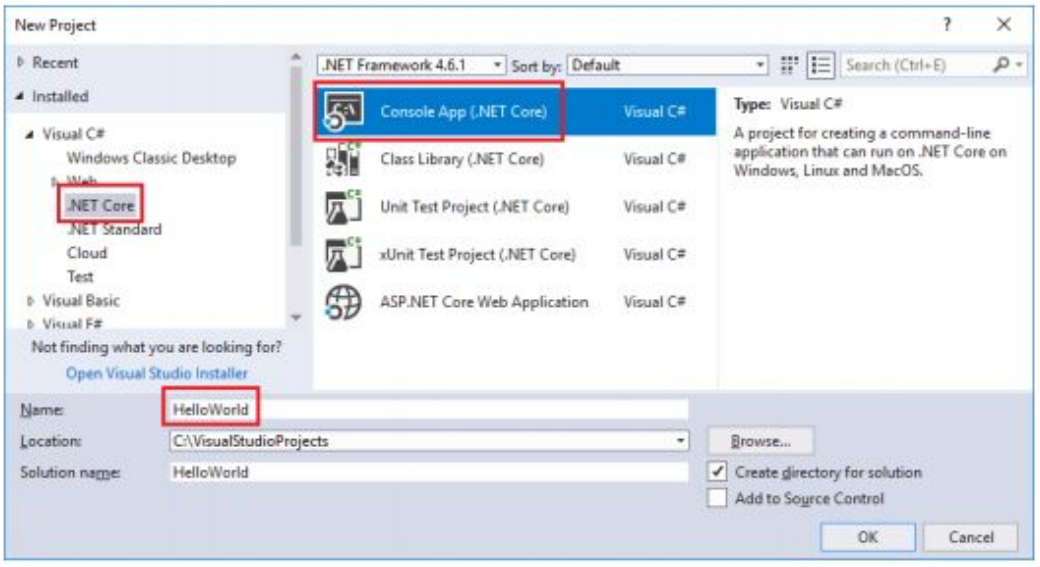
Let's dive in and create a new, simple program.

1. Open Visual Studio. On the menu, choose **File > New > Project**.



2. The **New Project** dialog box shows several project *templates*. A template contains the basic files and settings needed for a given project type. Choose the **.NET Core** category under **Visual C#**,

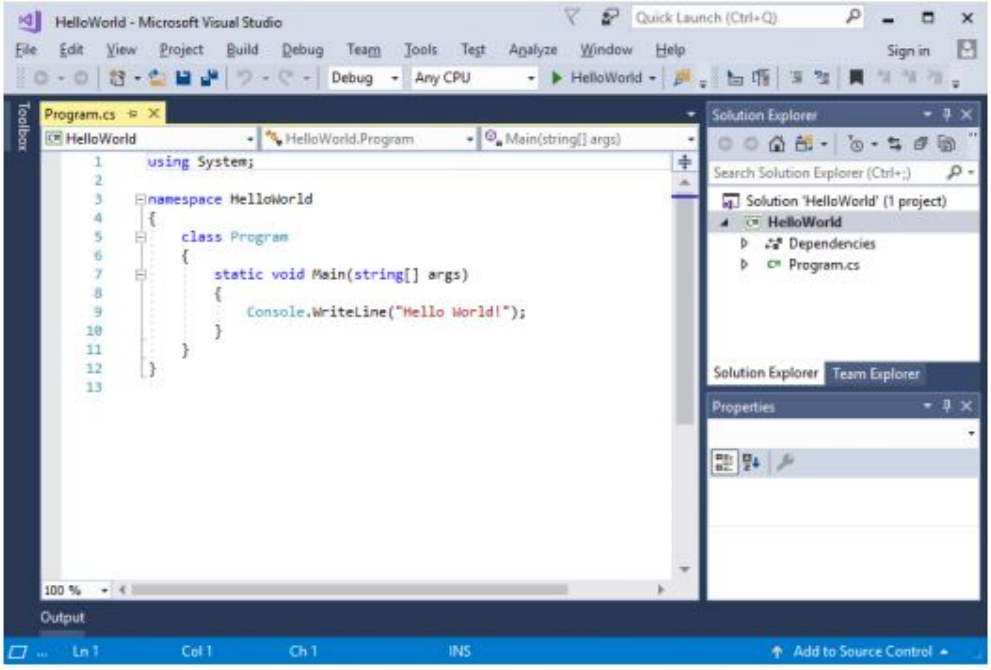
and then choose the **Console App (.NET Core)** template. In the **Name** text box, type **HelloWorld**, and then select the **OK** button.



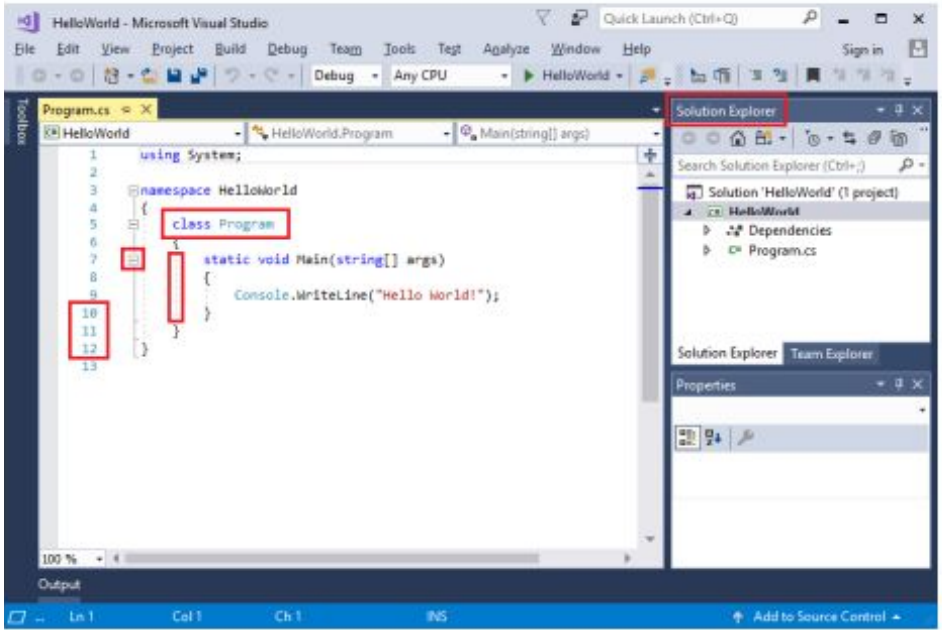
Visual Studio creates the project. It's a simple "Hello World" application that calls the Console.WriteLine() method to display the literal string "Hello World!" in the console (program output) window.

NOTE :
If you don't see the **.NET** Core category, you need to install the **.NET Core cross-platform development** workload. To do this, choose the **Open Visual Studio Installer** link on the bottom left of the **New Project** dialog. After Visual Studio Installer opens, scroll down and select the **.NET Core cross-platform development** workload, and then select **Modify**.

Shortly,you should see something like the following:

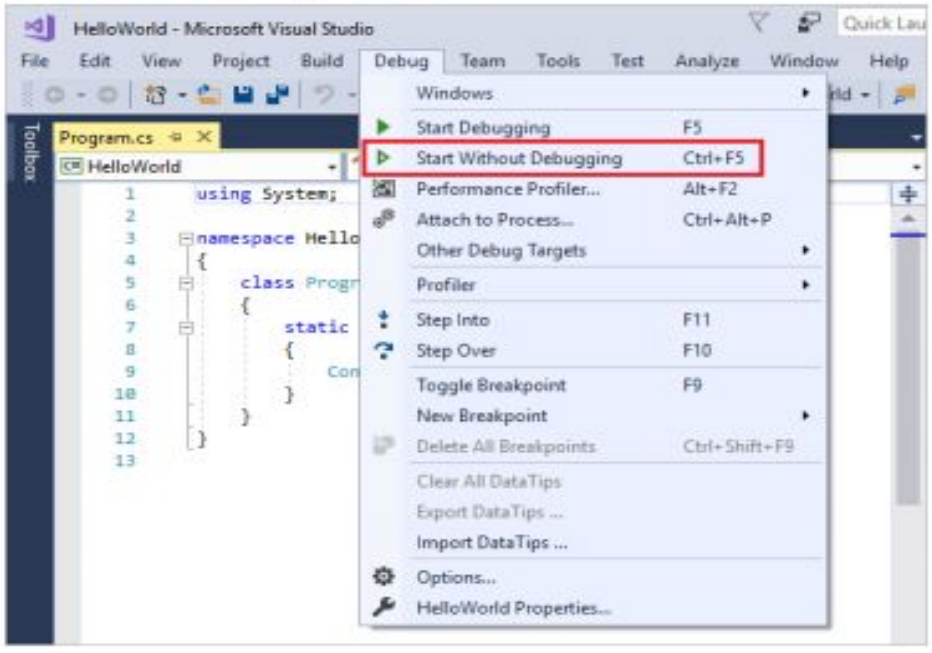


The C# code for your application shows in the editor window, which takes up most of the space. Notice that the text is automatically colorized to indicate different parts of the code, such as keywords and types. In addition, small,vertical dashed lines in the code indicate which braces match one another, and line numbers help you locate code later. You can choose the small, boxed minus signs to collapse or expand blocks of code.This code outlining feature lets you hide code you don't need, helping to minimize on screen clutter. The project files are listed on the right side in a window called **Solution Explorer**.

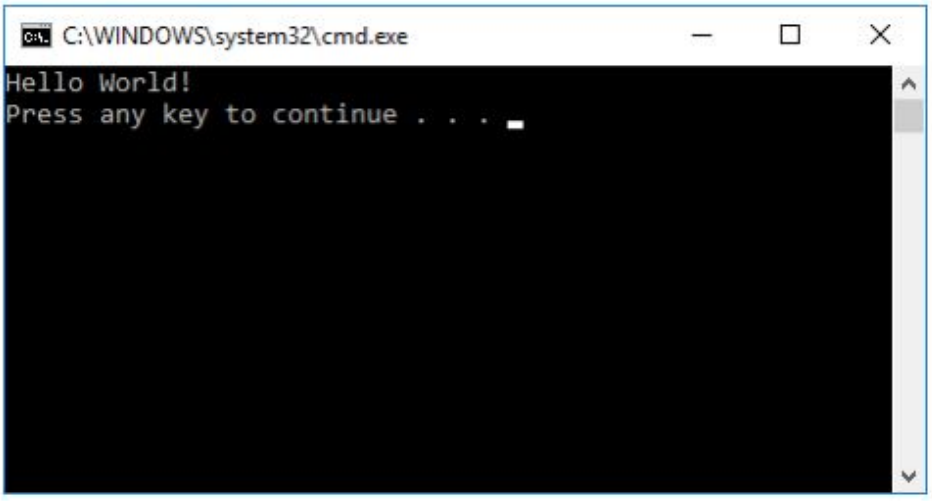


There are other menus and tool windows available, but let's move on for now.

3. Now, start the app. You can do this by choosing Start Without Debugging from the Debug menu on the menu bar. You can also press Ctrl+F5.



Visual Studio builds the app,and a console window opens with the message Hello World!. You now have a running app!



4. To close the console window, press any key on your keyboard.
5. Let's add some additional code to the app. Add the following C# code before the line that says

```
Console.WriteLine("Hello World!");
```

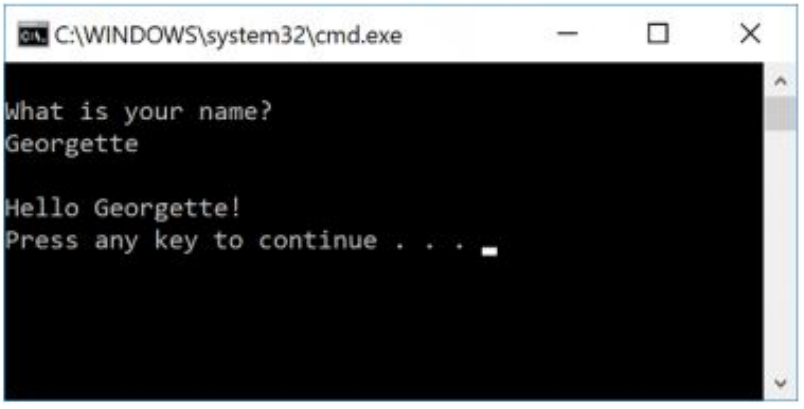
```
Console.WriteLine("\nWhat is your name?");
var name = Console.ReadLine();
```

This code displays **What is your name?** in the console window,and then waits until the user enters some text followed by the **Enter** key.

6. Change the line that says Console.WriteLine("Hello World!"); to the following code:

```
Console.WriteLine($"\nHello {name}!");
```

7. Run the app again by selecting **Debug > Start Without Debugging** or by pressing **Ctrl+F5.** Visual Studio rebuilds the app, and a console window opens and prompts you for your name.
8. Enter your name in the console window and press **Enter**.



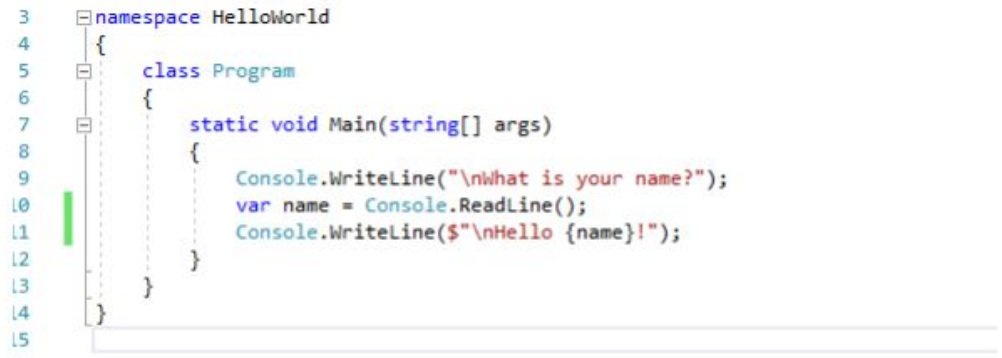
9. Press any key to close the console window and stop the running program.

Use refactoring and IntelliSense

Let's look at a couple of the ways that refactoring and IntelliSense can help you code more efficiently. First, let's rename the name variable:

1. Double-click the name variable to select it.
2. Type in the new name for the variable, **username**.
   Notice that a gray box appears around the variable,and a light bulb appears in the margin.
3. Select the light bulb icon to show the available Quick Actions.Select Rename 'name' to 'username'.

The variable is renamed across the project, which in our case is only two places.



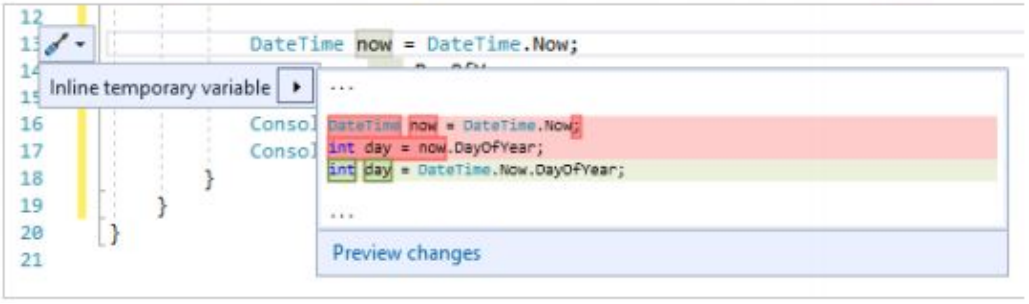
4. Now let's take a look at IntelliSense. Below the line that says Console.WriteLine($"\nHello {username}!"); ,
   type **DateTime now = DateTime.**.
   A box displays the members of the DateTime class. In addition, the description of the currently selected member displays in a separate box.

5. Select the member named **Now**, which is a property of the class, by double-clicking on it or pressing **Tab**. Complete the line of code by adding a semicolon ;
6. Below that, type in or copy the following lines of code:

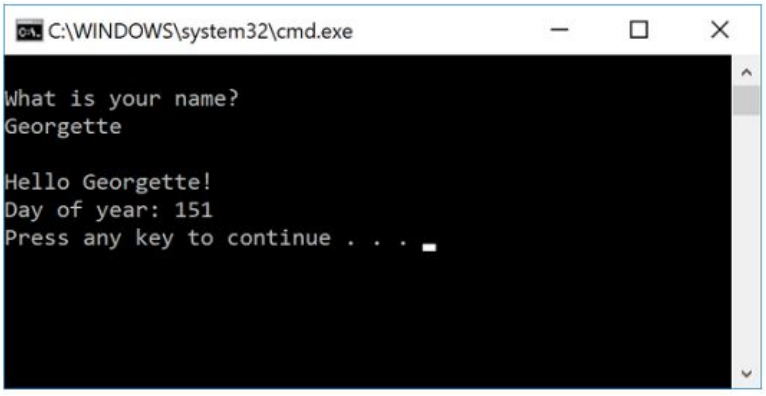
```
int dayOfYear = now.DayOfYear;

Console.Write("Day of year: ");
Console.WriteLine(dayOfYear);
```

TIP
Console.Write is a little different to Console.WriteLine in that it doesn't add a line terminator after it prints. That means that the next piece of text that's sent to the output will print on the same line. You can hover over each of these methods in your code to see their description.

7. Next, we'll use refactoring again to make the code a little more concise. Click on the variable now in the line DateTime now = DateTime.Now;
   Notice that a little screwdriver icon appears in the margin on that line.
8. Click the screwdriver icon to see what suggestions Visual Studio has available. In this case, it's showing the Inline temporary variable refactoring to remove a line of code without changing the overall behavior:



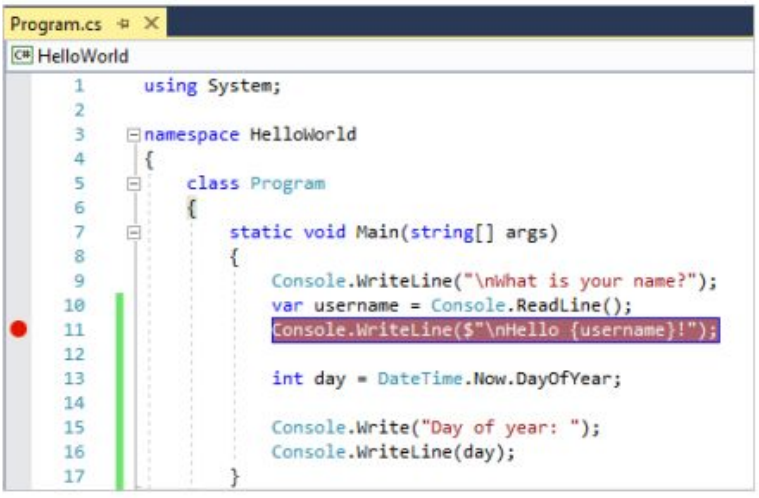
9. Click **Inline temporary variable** to refactor the code.
10. Run the program again by pressing **Ctrl+F5**.The output looks something like this:



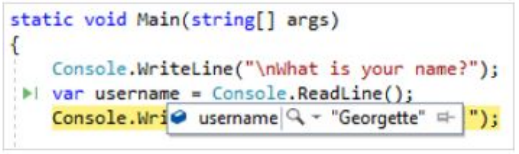
Debug code

When you write code,you need to run it and test it for bugs. Visual Studio's debugging system lets you step through code onestatementata time and inspect variables as you go. You can set breakpoints that stop execution of the code at a particular line. You can observe how the value of a variable changes as the code runs, and more.
Let's set a breakpoint to see the value of the username variable while the program is "in flight".

1. Find the line of code that says Console.WriteLine($"\nHello {username}!"); .To set a breakpoint on this line of code, that is, to make the program pause execution at this line,click in the far left margin of the editor.
   You can also click anywhere on the line of code and then press **F9**.
   A red circle appears in the far left margin,and the code is highlighted in red.



2. Start debugging by selecting **Debug > Start Debugging** or by pressing **F5**.
3. When the console window appears and asks for your name, type it in and press **Enter**. Notice that the focus returns to the Visual Studio code editor and the line of code with the breakpoint is highlighted in yellow.This signifies that it's the next line of code that the program will execute.
4. Hover your mouse over the username variable to see its value. Alternatively,you can right-click on username and select **Add Watch** to add the variable to the **Watch** window, where you can also see its value.

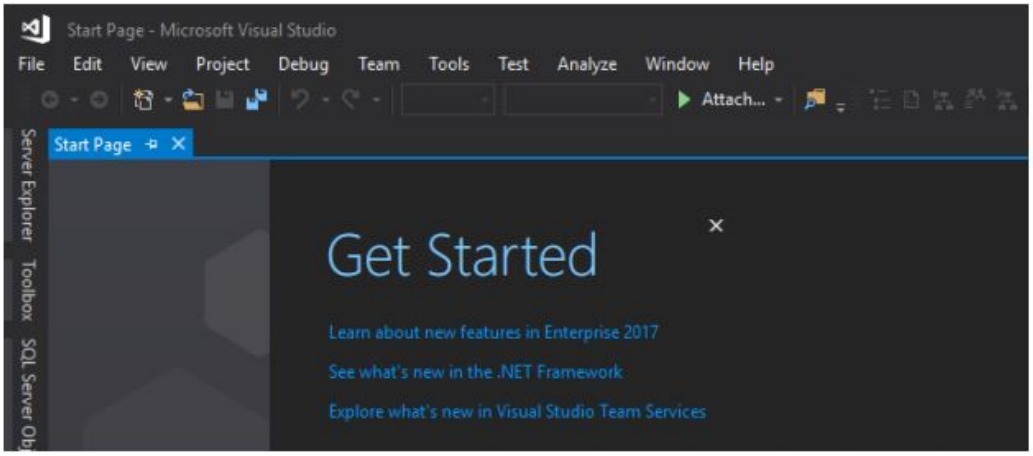


5. To let the program run to completion, press **F5** again.

Customize Visual Studio

You can personalize the Visual Studio user interface, including change the default color theme.To change to the Dark theme:

1. On the menu bar, choose **Tools > Options** to open the **Options** dialog.
2. On the **Environment > General** options page, change the **Color theme** selection to Dark, and then choose **OK**.
   The color theme for the entire IDE changes to **Dark**.



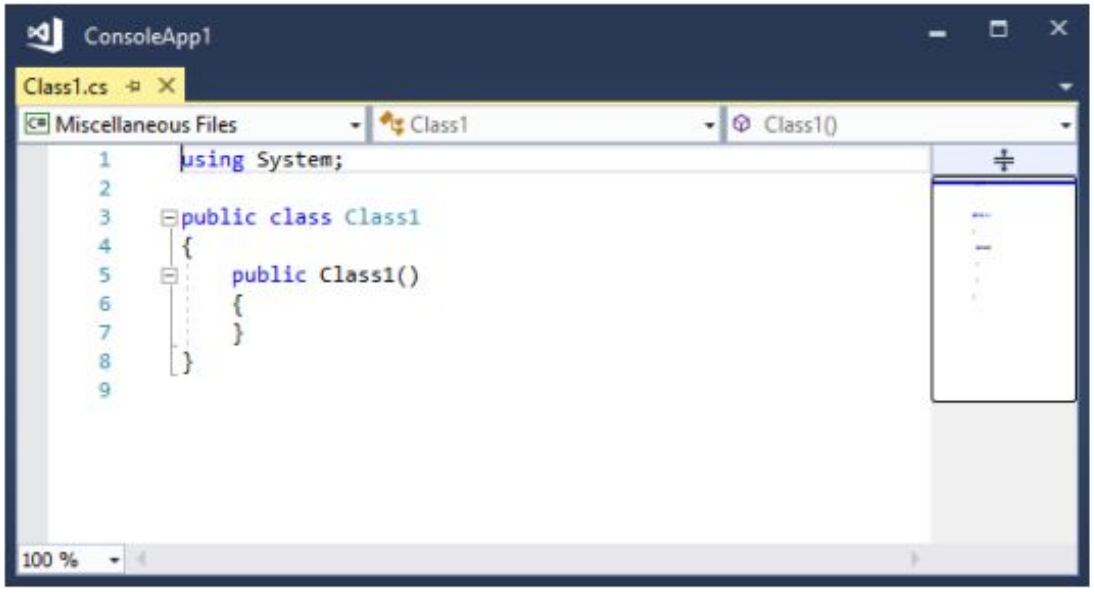
ABOUT CODE EDITOR

In this 10-minute introduction to the code editor in Visual Studio, we'll add code to a file to look at some of the ways that Visual Studio makes writing, navigating, and understanding code easier.
If you haven't already installed Visual Studio, go to the Visual Studio downloads page to install it for free. This article assumes you're already familiar with a programming language. If you aren't, we suggest you look at one of the programming quickstarts first, such as create a web app with Python or C#, or create a console app with Visual Basic or C++.

Create a new code file

Start by creating a new file and adding some code to it.

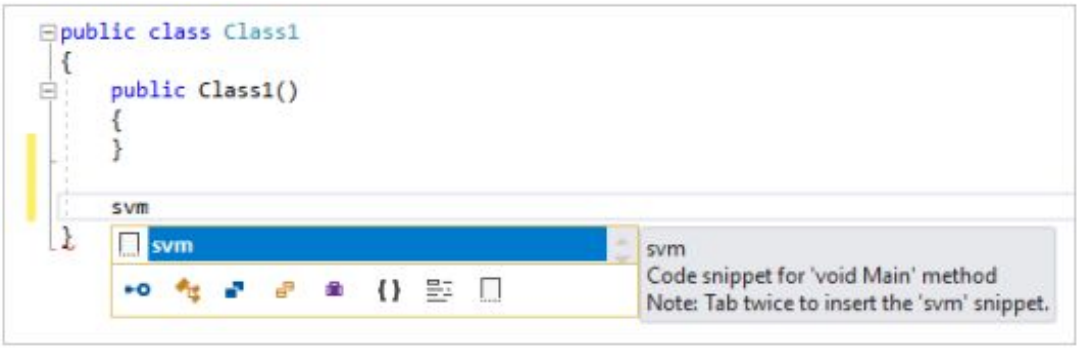
1. Open Visual Studio,and from the **File** menu on the menu bar, choose **New > File**
2. In the **New File** dialog box, under the **General** category,choose **Visual C# Class**, and then choose **Open**.
   A new file opens in the editor with the skeleton of a C# class. (Notice that we don't have to create a full Visual Studio project to gain some of the benefits that the code editor offers;all you need is a codefile!)



Use code snippets

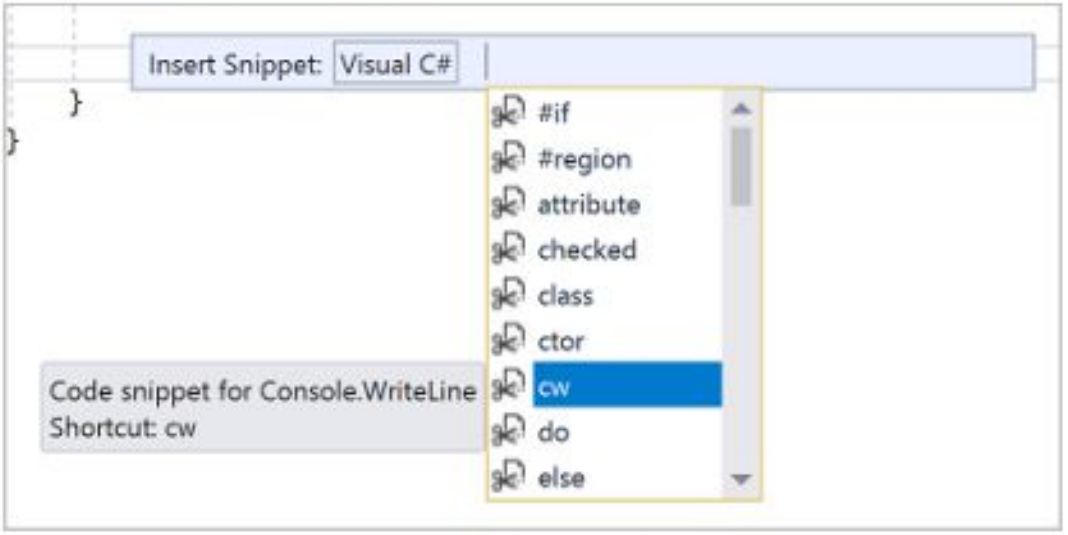
Visual Studio provides useful code snippets that you can use to quickly and easily generate commonly used code blocks. Code snippets are available for different programming languages including C#, Visual Basic, and C++. Let's add the C# void Main snippet to our file.

1. Place your cursor just above the final closing brace} in the file,and type the characters svm (which stands for static void Main —don't worry too much if you don't know what that means).
   A pop-up dialog box appears with information about the svm code snippet.



2. Press **Tab** twice to insert the code snippet. You seethe static void Main() method signature get added to the file. The Main() method is the entry point for C# applications.

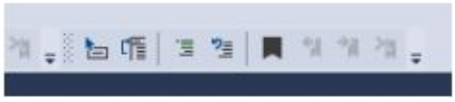
The available code snippets vary for different programming languages. You can look at the available code snippets for your language by choosing **Edit > IntelliSense > Insert** Snippet,and then choosing your language's folder. For C#, the list looks like this:



The list includes snippets for creating a class, a constructor, a for loop, an if or switch statement, and more.

Comment out code

The toolbar, which is the row of buttons under the menu bar in Visual Studio,can help make you more productive as you code. For example,you can toggle IntelliSense completion mode(IntelliSense is a coding aid that displays a list of matching methods,amongst other things), increase or decrease a line indent, or comment out code that you don't want to compile. In this section, we'll comment out some code.



1. Paste the following code into the Main() method body.

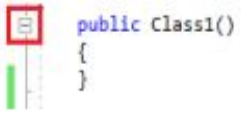
```
// _words is a string array that we'll sort alphabetically
string[] _words = {
    "the",
    "quick",
    "brown",
    "fox",
    "jumps"
};

string[] morewords = {
    "over",
    "the",
    "lazy",
    "dog"
};

IEnumerable<string> query = from word in _words
                            orderby word.Length
                            select word;
```

2. We're not using the morewords variable, but we may use it later so we don't want to completely delete it. Instead, let's comment out thoselines.Select the entire definition of morewords to the closing semicolon, and then choose the Comment out the selected lines button on the toolbar. If you prefer to use the keyboard, press Ctrl+K, Ctrl+C.

The C# comment characters // are added to the beginning of each selected line to comment out the code.

Collapse code blocks

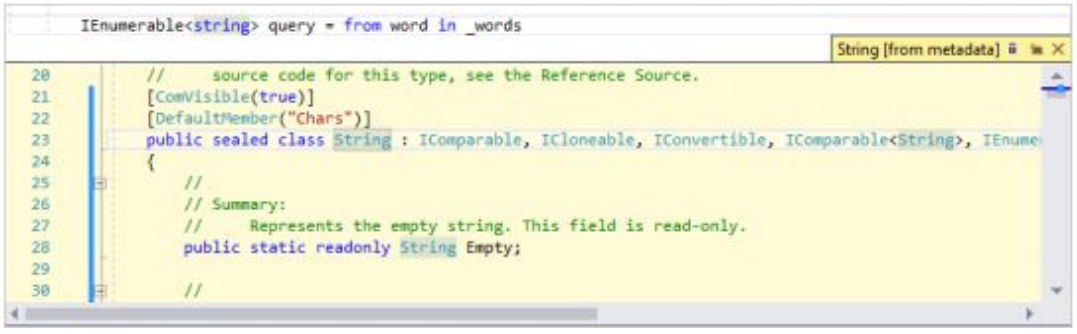
We don't want to see the empty constructor for Class 1 that was generated, so to declutter our view of the code, let's collapse it. Choose the small gray box with the minus sign inside it in the margin of the first line of the constructor. Or, if you're a keyboard user, place the cursor anywhere in the constructor code and press **Ctrl+M, Ctrl+M.**



The code block collapses to just the first line, followed by an ellipsis (...).To expand the code block again, click the same gray box that now has a plus sign in it, or press **Ctrl+M, Ctrl+M** again.This feature is called Outlining and is especially useful when you're collapsing long methods or entire classes.

View symbol definitions

The Visual Studio editor makes it easy to inspect the definition of a type, method,etc. One way is to navigate to the file that contains the definition, for example by choosing **Go to Definition** anywhere the symbol is referenced. An even quicker way that doesn't move your focus away from the file you're working in is to use Peek Definition. Let's peek at the definition of the string type.

1. Right-click on any occurrence of string and choose **Peek Definition** from the content menu. Or, press **Alt+F12**.
2. A pop-up window appears with the definition of the String class. You can scroll within the pop-up window, or even peek at the definition of another type from the peeked code.
3. 

4. Close the peeked definition window by choosing the small box with an "x"at the top right of the pop-up window.

Use IntelliSense to complete words

IntelliSense is an invaluable resource when you're coding. It can show you information about available members of a type, or parameter details for different overloads of a method. You can also use IntelliSense to complete a word after you type enough characters to disambiguate it. Let's add a line of code to print out the ordered strings to the console window, which is the standard place for output from the program to go.

1. Below the query variable, start typing the following code:

```
foreach (string str in qu
```

   You see IntelliSense show you **Quick Info** about the query symbol.

2. To insert the rest of the word query by using IntelliSense's word completion functionality, press **Tab**.

3. Finish off the code block to look like the following code. You can even practice using code snippets again by entering cw and then pressing **Tab** twice to generate the Console.WriteLine code.

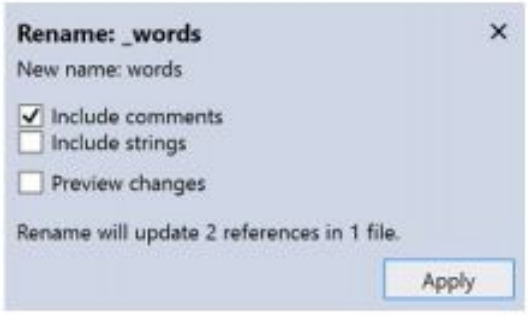
```
foreach (string str in query)
{
   Console.WriteLine(str);
}
```

Refactor a name

Nobody gets code right the first time,and one of the things you might have to change is the name of a variable or method. Let's try out Visual Studio's refactor functionality to rename the _words variable to words.

1. Place your cursor over the definition of the _words variable,and choose Rename from the right-click or context menu, or press **Ctrl+R, Ctrl+R**.
   A pop-up Rename dialog box appears at the top right of the editor.

2. Enter the desired name **words**. Notice that thereferenceto words in the query is also automatically Next steps See also renamed. Before you press **Enter**, select the **Include comments** checkbox in the **Rename** pop-up box.



3. Press **Enter**.
   Both occurrences of words have been renamed, as well as thereferenceto words in the code comment.

A Basic Overview Of ASP.NET

ASP.NET has been an impressive web development platform, which has provided a programming model, a comprehensive software infrastructure and numerous number of services which were used to build up robust web applications for PC, as well as mobile devices. It works on top of the HTTP protocol, and uses the HTTP commands and policies to set-up browser-to-server bilateral communication and cooperation.

ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework. The ASP.NET application codes can be written in any of the following languages:

- C#
- Visual Basic.Net
- Jscript
- J#

This framework is used to produce data-driven and interactive web applications across the World Wide Web (commonly referred to as the "Internet"). It consists of a large number of controls such as text boxes, buttons, control charts and labels for assembling, configuring, and manipulating code to create HTML pages. These controls are nothing but Drag & Drops which can be added to our ASP Web Application (as we are working on the Design view corresponding to the Application) as we continue on our voyage to add functionalities to this Application.

Web Forms Model of ASP.NET

ASP.NET web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for stateful processing. The server processes the output of the client actions and triggers the reactions.
Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:

- Page state
- Session state

The **page state** denotes the state of the client, i.e., the content of various input fields in the web form. Meanwhile, **session state** is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take an example of a shopping cart.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown on a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

The ASP.NET runtime carries the page state to and from the server across page requests while generating ASP.NET runtime codes, and incorporates the state of the server side components in hidden fields.

This way, the server becomes aware of the overall application state and operates in a two-tiered connected way.

Component Model of ASP.NET

The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:

Server side counterparts of almost all HTML elements or tags, such as <form> and <input>.
Server controls, which help in developing complex user-interface. For example, the Calendar control or the Gridview control.

ASP.NET is a technology, which works on the .Net framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.NET web application is made of pages. When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system.

The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework. Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

Components of .Net Framework

Before going to the next session on Visual Studio.Net, let us go through at the various components of the .Net framework. The following list describes the components of the .Net framework and the job they perform:

Components and their Description

1. **Common Language Runtime or CLR**
   It performs memory management, exception handling, debugging, security checking, thread execution, code execution, code safety, verification, and compilation. The code that is directly managed by the CLR is called the managed code. When the managed code is compiled, the compiler converts the source code into a CPU independent intermediate language (IL) code. A Just In Time(JIT) compiler compiles the IL code into native code, which is CPU specific.

2. **.Net Framework Class Library**
   It contains a huge library of reusable types. classes, interfaces, structures, and enumerated values, which are collectively called types.

3. **Common Language Specification**
   It contains the specifications for the .Net supported languages and implementation of language integration.

4. **Common Type System**

It provides guidelines for declaring, using, and managing types at runtime, and cross-language communication.

5. **Metadata and Assemblies**
   Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory. Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code, and a set of resources like image files.

6. **Windows Forms**
   Windows Forms contain the graphical representation of any window displayed in the application.

7. **ASP.NET and ASP.NET AJAX**
   ASP.NET is the web development model and AJAX is an extension of ASP.NET for developing and implementing AJAX functionality. ASP.NET AJAX contains the components that allow the developer to update data on a website without a complete reload of the page.

8. **ADO.NET**
   It is the technology used for working with data and databases. It provides access to data sources like SQL server, OLE DB, XML etc. The ADO.NET allows connection to data sources for retrieving, manipulating, and updating data.

9. **Windows Workflow Foundation (WF)**
   It helps in building workflow-based applications in Windows. It contains activities, workflow runtime, workflow designer, and a rules engine.

10. **Windows Presentation Foundation**
    It provides a separation between the user interface and the business logic. It helps in developing visually stunning interfaces using documents, media, two and three dimensional graphics, animations, and more.

11. **Windows Communication Foundation (WCF)**
    It is the technology used for building and executing connected systems.

12. **Windows CardSpace**
    It provides safety for accessing resources and sharing personal information on the internet.

13. **LINQ**
    It imparts data querying capabilities to .Net languages using a syntax which is similar to the tradition query language SQL.

Introducing ASP.NET

What Is ASP.NET ?

Introduction

ASP.NET is a programming Framework built on common language runtime that can be used on a server to build powerful Web Applications. ASP.NET offers several important advantages over previous Web Development Models :

- **Enhanced Performance :** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
- **World-Class Tool Support:** The ASP.NET framework is complimented with a rich-toolbox and designer in the Visual Studio integrated development Environment (IDE), WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few features this powerful tool provides.
- **Power and Flexibility:** Because ASP.NET is based on the common language runtime, the power and flexibility of the entire platform is available to the Web application developers. The .NET framework class library, Messaging, and Data Access Solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in in COM-based development is preserved when migrating to ASP.NET.
- **Simplicity:** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that clearly separate application logic from presentation code and to handle events in a simple, Visual Basic-like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.
- **Manageability:** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and the Web Applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. The “zero local administration” philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to server. No server restart is required, even to deploy or replace running compiled code.
- **Scalability & Availability:** ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one

misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

- **Customizability & Extensibility:** ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the asp.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.
- **Security:** With built-in Windows authentication and per-application configuration, you can be assured that your application are secure.

What Is ASP.NET Web Forms?

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate web pages. Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular in provides:

- The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that the page developer has to write.
- The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti" code)
- The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP.NET is opaque to tools)

ABOUT CONTROL CHARTS

Start Date & End Date

We used a starting date and an ending date to make it easy for the person using the Data Visualizer to Visualize the data according to their needs. Since, this was a major functionality of the Data Visualizer that it ought to have.

```
<%-- Start date and End Date Inputs --%>
<tr>
    <td class="auto-style17">
        <strong>
        <asp:Label ID="Label3" runat="server" Text="Start Date  :  " CssClass="auto-style8"></asp:Label>
        <asp:TextBox ID="TextStartDate" runat="server" TextMode="Date" CssClass="auto-style8"></asp:TextBox>
        </strong>
    </td>
    <td class="auto-style16">
        <strong>
        <asp:Label ID="Label2" runat="server" Text="End Date  :  " CssClass="auto-style8"></asp:Label>
        <asp:TextBox ID="TextEndDate" runat="server" TextMode="Date" CssClass="auto-style8"></asp:TextBox>
        </strong>
    </td>
</tr>
```

Populating the DropDownList1

Inside the Page Load event, the Drop Down List (which can be called DropDownList1) is populated with Department Name. When a Department Name is selected in the DropDownList, the statistical records of Service Details are fetched from the Database and Populates the other DropDownList. Finally, using these values the Chart is populated and displayed.

```
<%-- Department Name --%>
<tr>

    <td class="auto-style17">
        <strong>
        <asp:Label ID="Label5" runat="server" Text="Department Name" CssClass="auto-style8"></asp:Label>
        </strong>
    </td>

    <td class="auto-style16">

        <strong>

        <asp:DropDownList
            ID="DDDeptName"
            runat="server"
            CssClass="auto-style6"
            Height="34px"
            Width="200px"
            AutoPostBack="true"
            OnSelectedIndexChanged="DDDeptName_SelectedIndexChanged">
        </asp:DropDownList>

        </strong>

    </td>

</tr>
```

Populate the DropDownList2 from another DropDownList1

This Drop Down List was populated from another list in order to get selected Details for a certain Department (which can be called DropDownList2). This helped in sorting the queries of each Department which they were having trouble dealing with. This List has some Service Details which have been kept from due to Confidentiality purposes of the organization.

```
<%-- Service Details --%>
<tr>

    <td class="auto-style17">
        <strong>
        <asp:Label ID="Label7" runat="server" Text="Service Details" CssClass="auto-style8" >
        </asp:Label>
        </strong>
    </td>

    <td class="auto-style16">
        <strong>
        <asp:DropDownList
            ID="DDServiceDetails"
            runat="server"
            CssClass="auto-style6"
            Height="34px"
            Width="200px"
            AutoPostBack = "true"
            DataTextField="service_details" >
        </asp:DropDownList>

        </strong>

    </td>

</tr>
```

Using Radio Buttons

We used 2 radio buttons as a set of field. This was done to select the graph that we wanted to be in the 2D or 3D view which worked out as an advantage for people working as they could acquire better graphics overall, at the same time also see their graph plotted in 2D view.

```
<%-- Select Dimensionality of the Chart --%>
<tr class="auto-style5">

    <td class="auto-style18">
        <strong>
        <asp:Label ID="Label6" runat="server" Text="Select Chart Dimension : " CssClass="auto-style7"></asp:Label>
        </strong>
    </td>

    <td class="auto-style10">
          <strong>
          <asp:RadioButton
              ID="TwoDim"
              Text="2-D"
              runat="server"
              style ="padding-right:50px"
              AutoPostBack="True"
              GroupName="Dimension"
              OnCheckedChanged="TwoDim_CheckedChanged"
              CssClass="auto-style7">
          </asp:RadioButton>
          <asp:RadioButton
              ID="ThreeDim"
              Text="3-D"
              runat="server"
              AutoPostBack="True"
              GroupName="Dimension"
              OnCheckedChanged="ThreeDim_CheckedChanged"
              CssClass="auto-style7">
          </asp:RadioButton>
          </strong>
    </td>

</tr>
```

Using DropDownList to select Chart Types

We populated a Drop Down List (let's call it DropDownList3) just for the sake of showing the different Chart Types that our Data can be visualized in some of which required prerequisites for them to be 3D to get a better visualization (done by ADO.NET). These included some Chart Types such as Funnel, Box Plot.

```
<%-- Chart Type input --%>
<tr>

    <td class="auto-style17">
        <strong>
        <asp:Label ID="Label4" runat="server" Text="Select Chart Type" CssClass="auto-style8"></asp:Label>
        </strong>
    </td>

    <td class="auto-style16">
        <strong>
        <asp:DropDownList
            ID="DropDownList1"
            AutoPostBack="true"
            runat="server"
            OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged"
            Height="34px"
            Width="168px"
            CssClass="auto-style6">
        </asp:DropDownList>
        </strong>
    </td>

</tr>
```

Created Chart

```
<%-- Chart Create  --%>
<tr>
    <td colspan="2" class="auto-style9">

        <strong resource="~/images/background.png">

        <asp:Chart
            ID="Chart1"
            CanResize="true"
            runat="server"
            EnableTheming = "true"
            Height = "590px"
            Width ="688px"
            BorderlineColor="Transparent"
            BackImage="~/images/background.png"
            CssClass="auto-style7"
            ImageType="Jpeg"
            >
            <Series>
                <asp:Series BackImage="~/images/background.png">
                    <SmartLabelStyle AllowOutsidePlotArea="No" />
                </asp:Series>
            </Series>
            <ChartAreas>
                <asp:ChartArea Name="ChartArea1"  BackImage="~/images/background.png">
                    <AxisX Title="Date (dd/mm/yyyy)" />
                    <AxisY Title="Total no. of transactions" />
                    <Area3DStyle />
                </asp:ChartArea>
            </ChartAreas>
            <Legends>
                <asp:Legend Name="Legend1" BackImage="~/images/background.png"/>
            </Legends>
            <Titles>
                <asp:Title
                    Name="Title1"
                    >
                </asp:Title>
            </Titles>
        </asp:Chart>
        <asp:Label ID="Label1" runat="server" CssClass="auto-style7"></asp:Label>
        </strong>
    </td>
</tr>
```

INPUT (CODE BEHIND (.aspx.cs))

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MySql.Data.MySqlClient;
using System.Data;
using System.Web.UI.DataVisualization.Charting;
using System.Configuration;


namespace WebApplication2
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (TwoDim.Checked)
         {
            Chart1.Palette = ChartColorPalette.Grayscale;
            Chart1.ChartAreas["ChartArea1"].Area3DStyle.Enable3D = false;
         }

         else if (ThreeDim.Checked)
         {
            Chart1.ChartAreas["ChartArea1"].Area3DStyle.Enable3D = true;
            Chart1.ChartAreas["ChartArea1"].Area3DStyle.WallWidth = 2;
            Chart1.ChartAreas["ChartArea1"].Area3DStyle.Rotation = 40;
         }

         if (!IsPostBack)
         {

            {
               MySqlConnection con = new
MySqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].
ConnectionString);
               string MySQLQuery3 = "SELECT DISTINCT " +
```

```
                "Dept_Name " +
                "FROM mas_service";


            MySqlCommand cmd2 = new MySqlCommand(MySQLQuery3, con);
            con.Open();
            DDDeptName.DataSource = cmd2.ExecuteReader();

            DDDeptName.DataTextField = "Dept_Name";
            //DDDeptName.DataValueField = "COUNT(Dept_Name)";
            //DDDeptName.DataValueField = "AutoID";
            DDDeptName.DataValueField = DDDeptName.DataSourceID;
            DDDeptName.DataBind();
            DDDeptName.Items.Insert(0, new ListItem("-------Select-------", "NA"));
            DDServiceDetails.Items.Insert(0, new ListItem("-------Select-------", "NA"));
            con.Close();
        }
        GetChartData();
        GetChartTypes();
    }
}

    private void GetChartData()
    {
        MySqlConnection con = new
MySqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].
ConnectionString);

     string MySQLQuery2 = "SELECT st.sno, st.insrt_dt, st.no_of_trans, st.transaction_time,
ms.service_details, ms.Dept_name " +
            "FROM service_trans AS st " +
            "INNER JOIN mas_service AS ms " +
            "ON st.service_id = ms.service_code " +
            "WHERE ms.Dept_Name='" + DDDeptName.SelectedValue + "' " +
            "AND ms.service_code='" + DDServiceDetails.SelectedValue + "' " +
            "AND st.insrt_dt " +
            "BETWEEN'" + TextStartDate.Text +
            "'AND'" + TextEndDate.Text + "'";

        MySqlCommand cmd2 = new MySqlCommand(MySQLQuery2, con);
        con.Open();
        MySqlDataAdapter adp1 = new MySqlDataAdapter(cmd2);
```

```
        DataTable ds = new DataTable();
        adp1.Fill(ds);

        MySqlDataReader rdr1 = cmd2.ExecuteReader();

        if (rdr1.Read())
        {

            //Chart Properties
            Chart1.Width = 659;
            Chart1.Height = 567;

            //Chart1.BorderlineColor = System.Drawing.Color.Transparent;
            Chart1.AlternateText = "Sorry!! We couldn't load the requested Graph";
            //Chart1.BackImageTransparentColor = System.Drawing.Color.Transparent;
            Chart1.BackImageWrapMode = ChartImageWrapMode.Scaled;
            Chart1.CssClass = "bcol";
            Chart1.BackImage = "~/images/background.png";



            //Chart Series Defined and its Properties

            Chart1.Series["Series1"].ChartArea = "ChartArea1";
            Chart1.Series["Series1"].Legend = "Legend1";
            Chart1.Series["Series1"].IsValueShownAsLabel = true;

            Chart1.Series["Series1"].XValueMember = "insrt_dt";
            Chart1.Series["Series1"].YValueMembers = "no_of_trans";
            Chart1.Series["Series1"].Font = new System.Drawing.Font("Monotype Corsiva", 12.0f);

            Chart1.Series["Series1"].SmartLabelStyle.AllowOutsidePlotArea =
LabelOutsidePlotAreaStyle.No;

            //Chart Series ChartAreas Properties
            Chart1.ChartAreas["ChartArea1"].BackImage = "~/images/background.png";

            Chart1.ChartAreas["ChartArea1"].AxisX.Title = "Date (dd/mm/yyyy)";
            Chart1.ChartAreas["ChartArea1"].AxisX.TitleFont = new System.Drawing.Font("Monotype
Corsiva", 14.0f);
            //Chart1.ChartAreas["ChartArea1"].AxisX.LabelAutoFitStyle = LabelAutoFitStyles.None;
```

```
            Chart1.ChartAreas["ChartArea1"].AxisX.LabelStyle.Font = new
System.Drawing.Font("Monotype Corsiva", 9.0f);
            Chart1.ChartAreas["ChartArea1"].AxisX.LabelStyle.Angle = -90;

            Chart1.ChartAreas["ChartArea1"].AxisY.Title = "Total no. of transactions";
            Chart1.ChartAreas["ChartArea1"].AxisY.TitleFont = new System.Drawing.Font("Monotype
Corsiva", 14.0f);
            //Chart1.ChartAreas["ChartArea1"].AxisY.LabelAutoFitStyle = LabelAutoFitStyles.None;
            Chart1.ChartAreas["ChartArea1"].AxisY.LabelStyle.Font = new
System.Drawing.Font("Monotype Corsiva", 9.0f);

            //Chart Series Legend Properties
            Chart1.Series["Series1"].Legend = "Legend1";

            Chart1.Legends["Legend1"].Title = "Service Details on Dates";
            Chart1.Legends["Legend1"].TitleFont = new System.Drawing.Font("Monotype Corsiva",
12.0f);

            Chart1.Legends["Legend1"].BackImage = "~/images/background.png";

            Chart1.Legends["Legend1"].Font = new System.Drawing.Font("Monotype Corsiva", 12.0f);

            //Chart Title Properties
            Chart1.Titles["Title1"].Text = "Service Details Analysis according to dates";
            Chart1.Titles["Title1"].Font = new System.Drawing.Font("Monotype Corsiva", 24.0f);
            Chart1.Palette = ChartColorPalette.SeaGreen;

            //Chart Data binding
            Chart1.DataSource = rdr1;
            Chart1.DataBind();
        }
    }
    private void GetChartTypes()
    {
        foreach (int chartType in Enum.GetValues(typeof(SeriesChartType)))
        {
            if (chartType != 9 && chartType != 12 && chartType != 16 && chartType != 25)
            {
                ListItem li = new ListItem(Enum.GetName(typeof(SeriesChartType), chartType),
chartType.ToString());
                DropDownList1.Items.Add(li);
            }
```

```
        }
    }
    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        GetChartData();
        Chart1.Series["Series1"].ChartType = (SeriesChartType)Enum.Parse(typeof(SeriesChartType),
DropDownList1.SelectedValue);
    }
    protected void ThreeDim_CheckedChanged(object sender, EventArgs e)
    {
        Chart1.ChartAreas["ChartArea1"].Area3DStyle.Enable3D = false;
        Chart1.Palette = ChartColorPalette.Grayscale;
    }
    protected void TwoDim_CheckedChanged(object sender, EventArgs e)
    {
        Chart1.ChartAreas["ChartArea1"].Area3DStyle.Enable3D = true;
        Chart1.ChartAreas["ChartArea1"].Area3DStyle.WallWidth = 2;
        Chart1.ChartAreas["ChartArea1"].Area3DStyle.Rotation = 35;
    }
    protected void DDDeptName_SelectedIndexChanged(object sender, EventArgs e)
    {
        DDServiceDetails.DataBind();
        DDServiceDetails.Items.Clear();
        string MySQLQuery4 = "SELECT service_details, service_code " +
            "FROM mas_service " +
            "WHERE Dept_Name= '" + DDDeptName.SelectedItem.Value + "'";
        MySqlConnection con = new
MySqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].
ConnectionString);
        MySqlCommand cmnd = new MySqlCommand(MySQLQuery4, con);
        MySqlDataAdapter sda = new MySqlDataAdapter(cmnd);
        DataTable dt = new DataTable();
        sda.Fill(dt);
        DDServiceDetails.DataSource = dt;
        DDServiceDetails.DataTextField = "service_details";
        DDServiceDetails.DataValueField = "service_code";
        DDServiceDetails.DataBind();
        DDServiceDetails.Items.Insert(0, new ListItem("-------Select-------", "NA"));

    }
  }
}
```

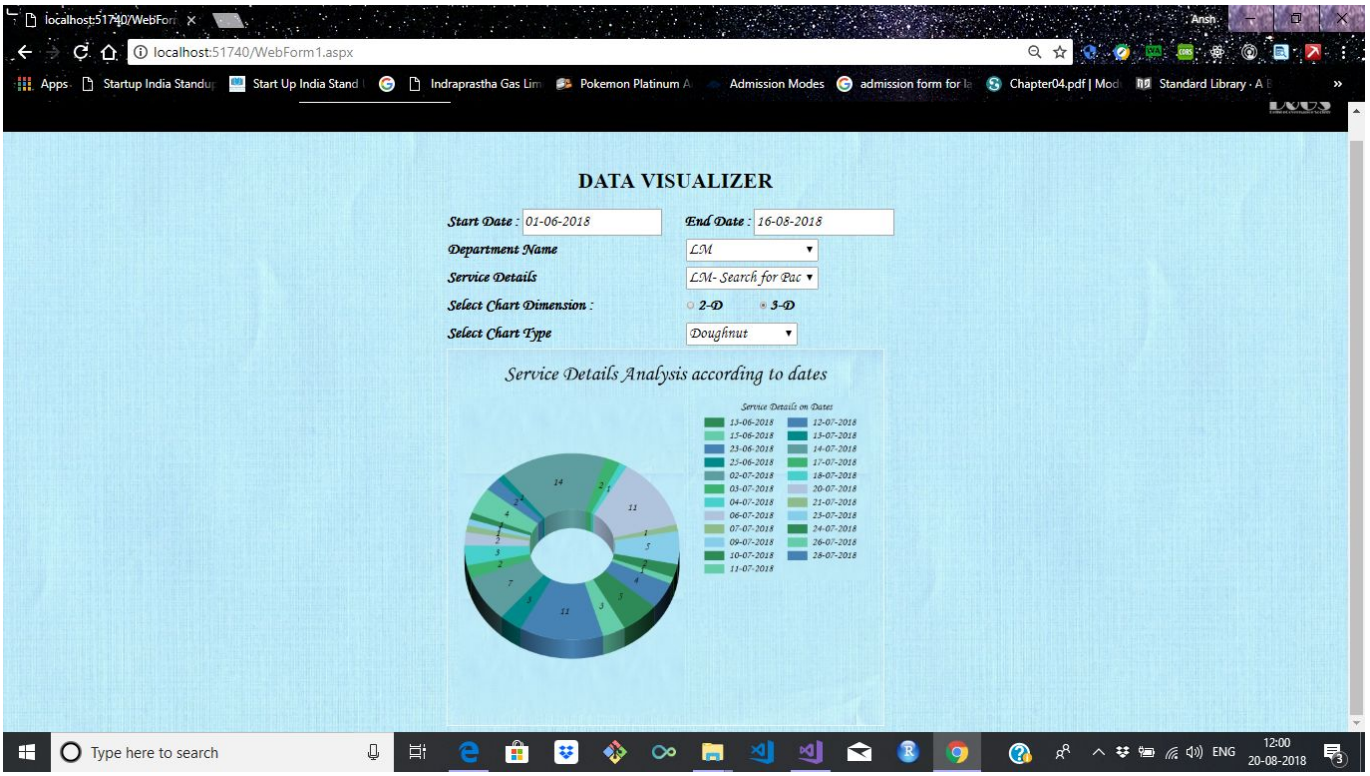
STYLESHEET

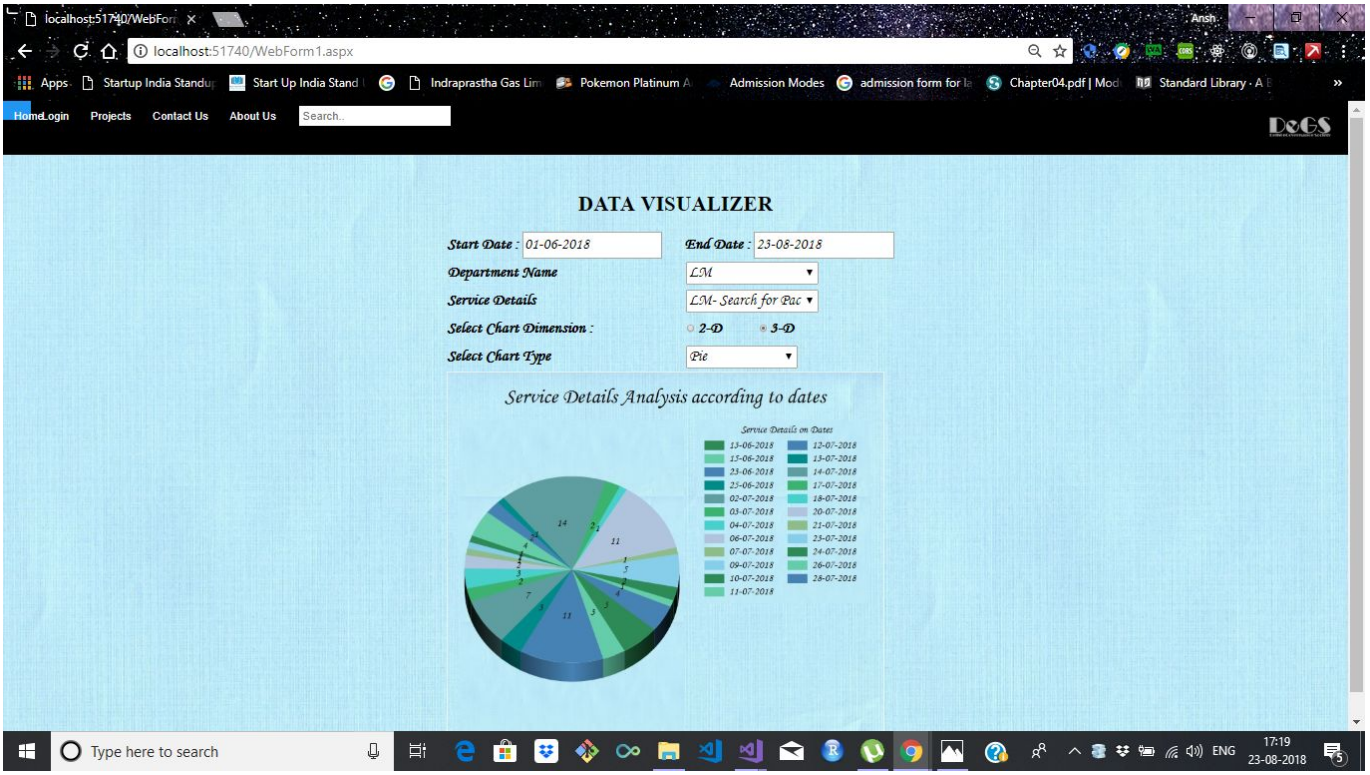
```
.bcol {
   background-size: 100px 50px;
}

.auto-style1 {
   height: 40px;
}

#topbarlogo {
   float: right; padding: 10px;
}

.auto-style2 {
   width: 318px; height: 340px;margin-top: 3px;
}

*{
   box-sizing: border-box;
}

body {
   margin: 0;

   font-family: Arial, Helvetica, sans-serif;
   /*background:
url('./images/background.png');*/
   /*background-repeat: no-repeat;*/
}

.topnav {
   overflow: hidden;
   background-color: black;
}

   .topnav a {
      float: left;
      display: block;
      color: white;
      text-align: center;
      padding: 14px 16px;
      text-decoration: none;
      font-size: 17px;
   }

      .topnav a:hover {
         background-color: #ddd;
         color: black;
      }

      .topnav .active {
         background-color: #2196F3;
         color: white;
      }

   .topnav input[type=text] {
      /*float: right;*/
      padding: 6px;
      margin-top: 8px;
      margin-right: 16px;
      margin-left: 18px;
      border: none;
      font-size: 17px;
   }

@media screen and (max-width: 600px) {
   .topnav a, .topnav input[type=text] {
      float: none;
      display: block;
      text-align: left;
      width: 100%;
      margin: 0;
      padding: 14px;
   }

   .topnav input[type=text] {
      border: 1px solid #ccc;
```

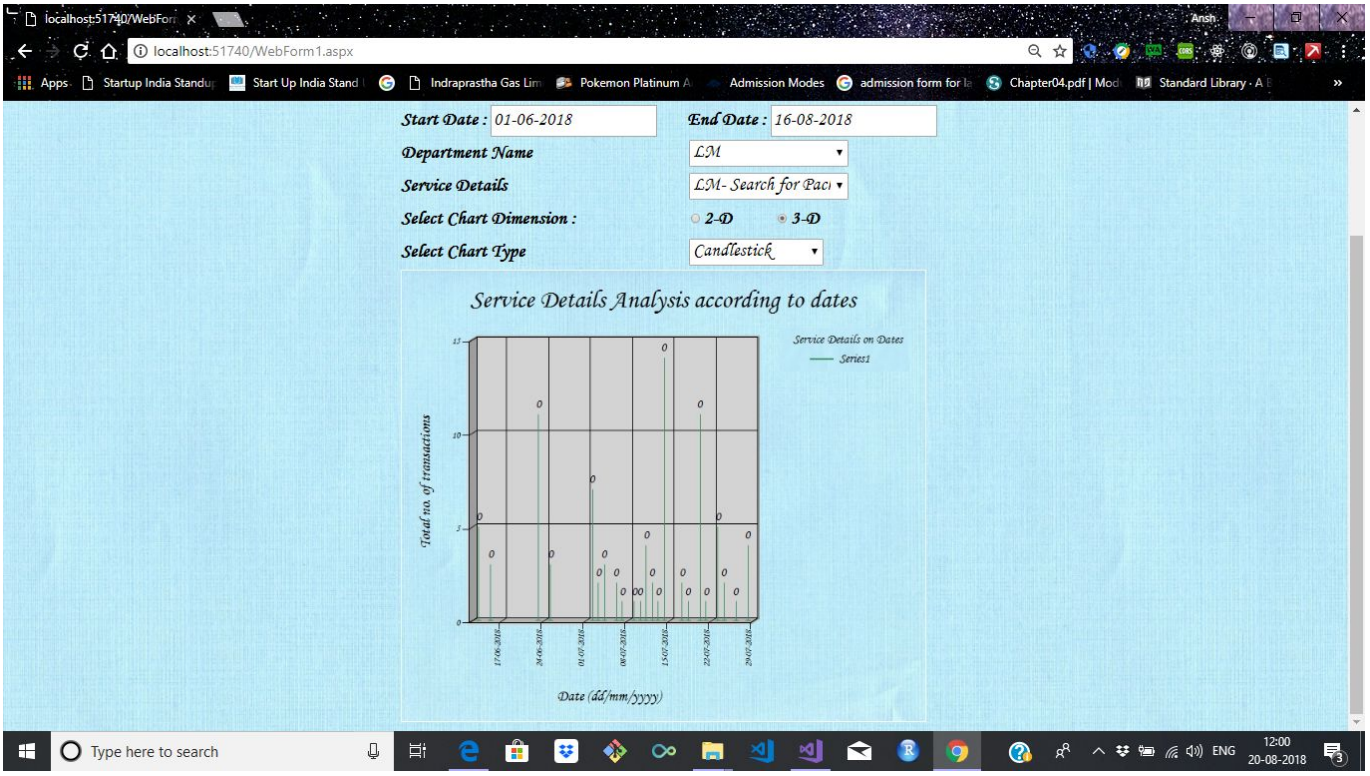
```
    }
}

.auto-style4 {
    margin-left: 3px;
}
.auto-style5 {
        font-family: "Times New Roman",
Times, serif;
    }
    .auto-style6 {
        margin-left: 3px;
        font-family: "Monotype Corsiva";
        font-size: x-large;
    }
    .auto-style7 {
        font-size: x-large;
    }
    .auto-style8 {
        font-family: "Monotype Corsiva";
        font-size: x-large;
    }
    .auto-style9 {
        font-family: "Monotype Corsiva";
    }
    .auto-style10 {
        height: 40px;
        font-family: "Monotype Corsiva";
        width: 395px;
    }
    .auto-style11 {
        width: 694px;
        height: 340px;
        margin-top: 3px;
    }
    .auto-style16 {
        height: 40px;
        width: 395px;
    }
    .auto-style17 {
        height: 40px;
        width: 427px;
    }
    .auto-style18 {
        height: 40px;
        font-family: "Monotype Corsiva";
        width: 427px;
    }
    .auto-style19 {
        font-family: "Times New Roman",
Times, serif;
        font-size: xx-large;
```

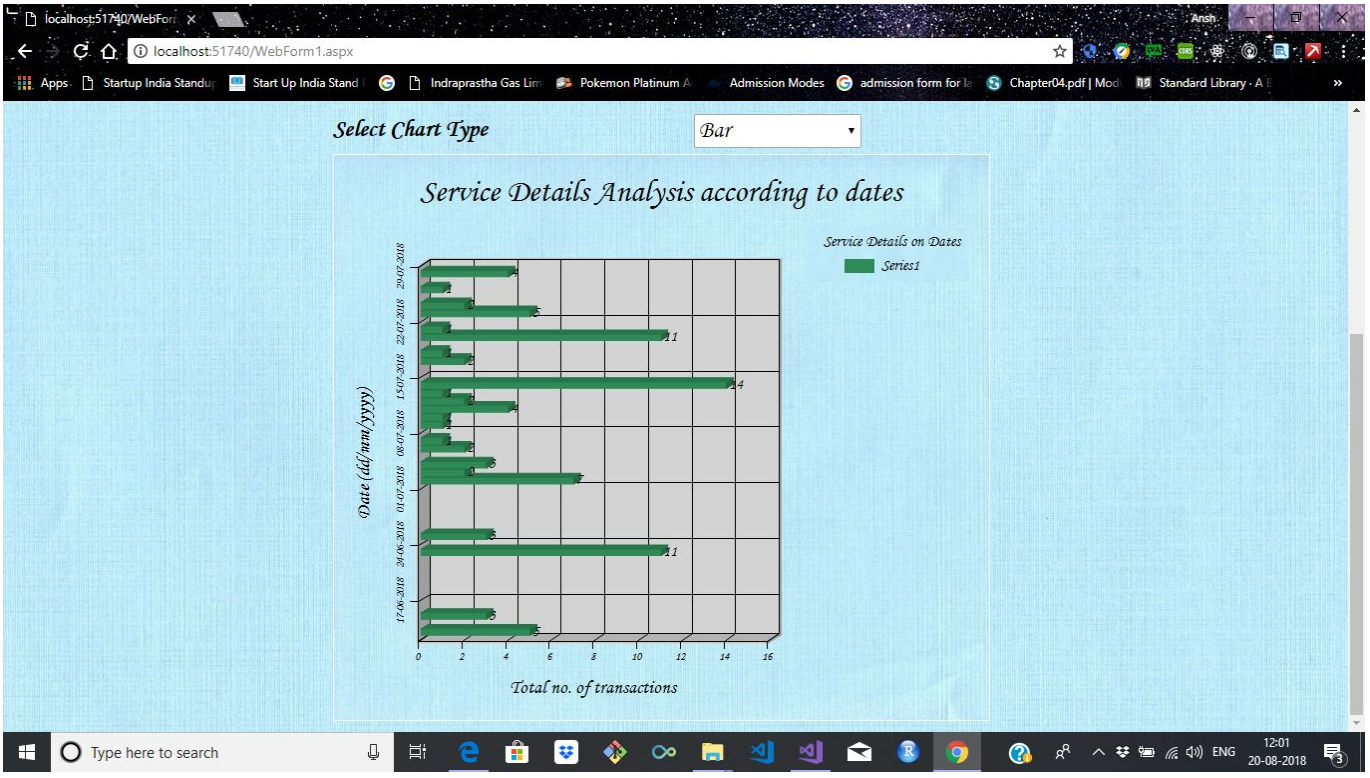
OUTPUTS

There were certain screenshots that we recorded in the early phases of our work which have been presented as Outputs. Some of these Outputs of our Data Visualizer have been displayed below :






localhost:51740/WebForm1.aspx
Start Date : 01-06-2018
End Date : 16-08-2018
Department Name
LM
Service Details
LM- Search for Pacl
Select Chart Dimension :
2-D
3-D
Select Chart Type
Candlestick
Service Details Analysis according to dates
Service Details on Dates
Series1
Total no. of transactions
Date (dd/mm/yyyy)


localhost:51740/WebForm1.aspx
Select Chart Type
Bar
Service Details Analysis according to dates
Service Details on Dates
Series1
Total no. of transactions
Date (dd/mm/yyyy)

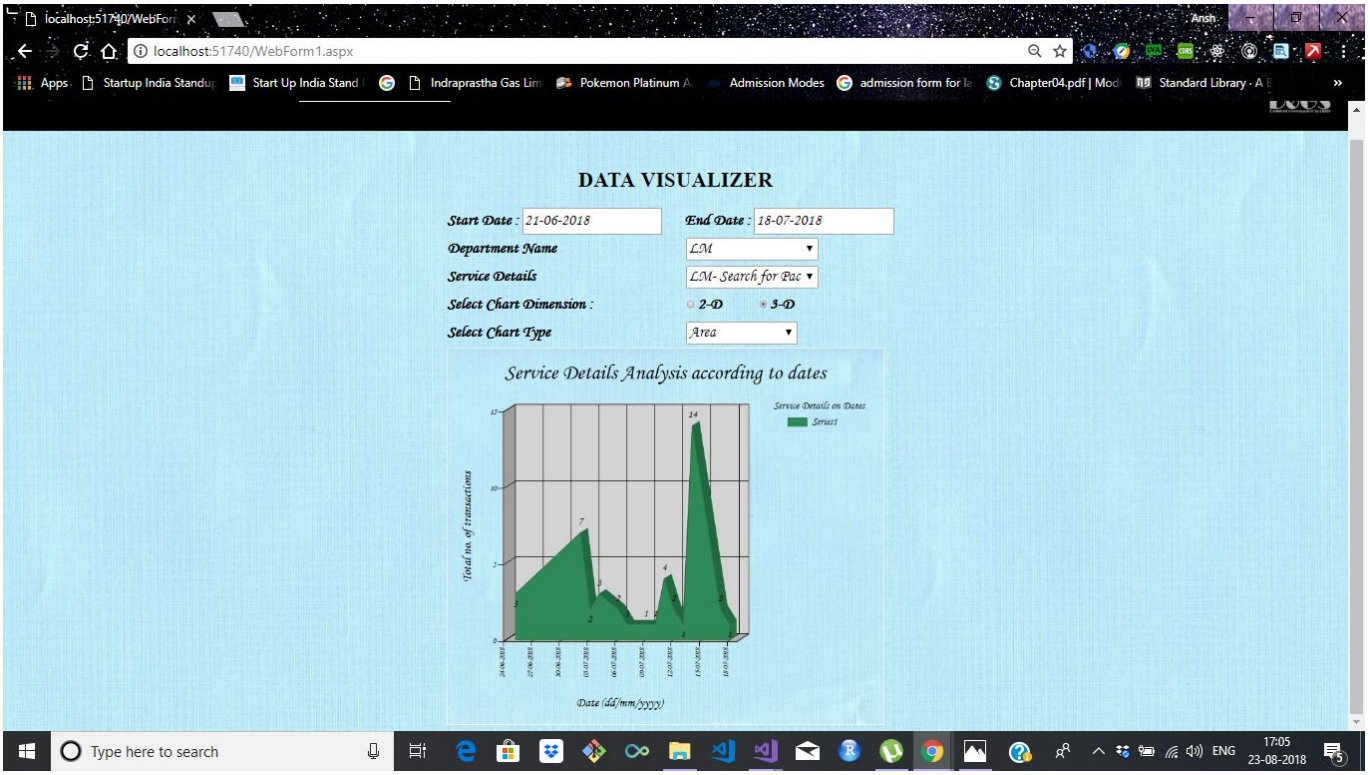

localhost:51740/WebForm1.aspx
DATA VISUALIZER
Start Date : 21-06-2018
End Date : 18-07-2018
Department Name
LM
Service Details
LM- Search for Pac
Select Chart Dimension :
2-D
3-D
Select Chart Type
Area
Service Details Analysis according to dates
Service Details on Dates
Series1
Total no. of transactions
Date (dd/mm/yyyy)
Type here to search
ENG
17:05
23-08-2018

TABLE OF CONTENTS II
(for Machine Learning)

Prototyping and Understanding Microsoft Azure

What is Azure?

Azure is a complete cloud platform that can host our existing applications, streamline the development of new applications, and even enhance on-premises applications. Azure integrates the cloud services that we need to develop, test, deploy, and manage our applications—while taking advantage of the efficiencies of cloud computing.

By hosting our applications in Azure, we can start small and easily scale our application as our customer demand grows. Azure also offers the reliability that's needed for high-availability applications, even including failover between different regions. The Azure portal lets we easily manage all our Azure services. we can also manage our services programmatically by using service-specific APIs and templates.

Who should read this: This guide is an introduction to the Azure platform for application developers. It provides guidance and direction that we need to start building new applications in Azure or migrating existing applications to Azure.

Where do I start?

With all the services that Azure offers, it can be a daunting task to figure out which services we need to support our solution architecture. This section highlights the Azure services that developers commonly use. For a list of all Azure services, see the Azure documentation.

First, we must decide on how to host our application in Azure. Do we need to manage our entire infrastructure as a virtual machine (VM). Can we use the platform management facilities that Azure provides? Maybe we need a serverless framework to host code execution only?

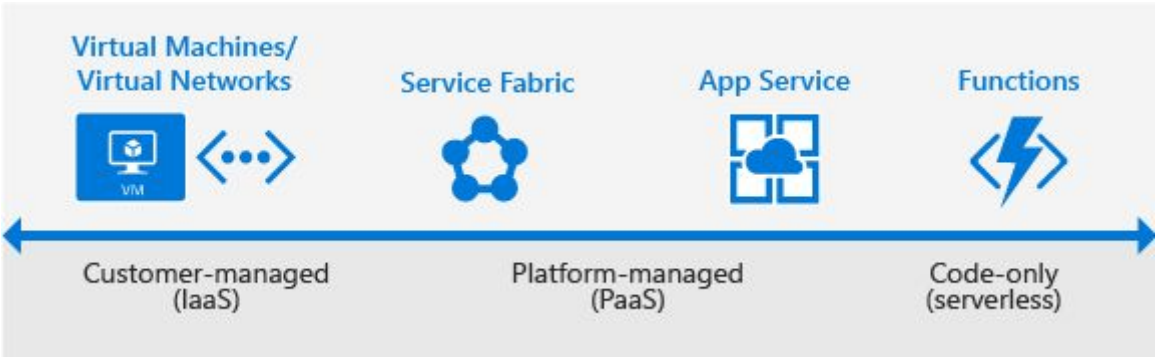
our application needs cloud storage, which Azure provides several options for. we can take advantage of Azure's enterprise authentication. There are also tools for cloud-based development and monitoring, and most hosting services offer DevOps integration.

Now, let's look at some of the specific services that we recommend investigating for our applications.

Application hosting

Azure provides several cloud-based compute offerings to run our application so that we don't have to worry about the infrastructure details. We can easily scale up or scale out our resources as our application usage grows.

Azure offers services that support our application development and hosting needs. Azure provides Infrastructure as a Service (IaaS) to give we full control over our application hosting. Azure's Platform as a Service (PaaS) offerings provide the fully managed services needed to power our apps. There is even true serverless hosting in Azure where all we need to do is write our code.



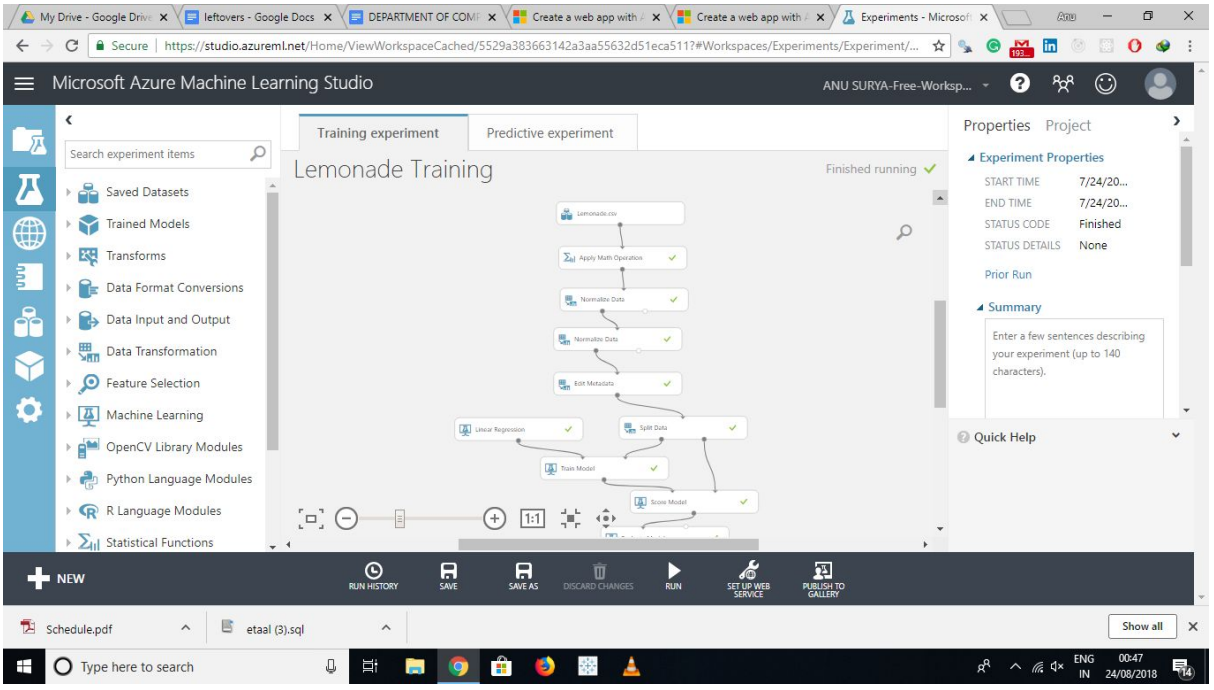
Azure App Service

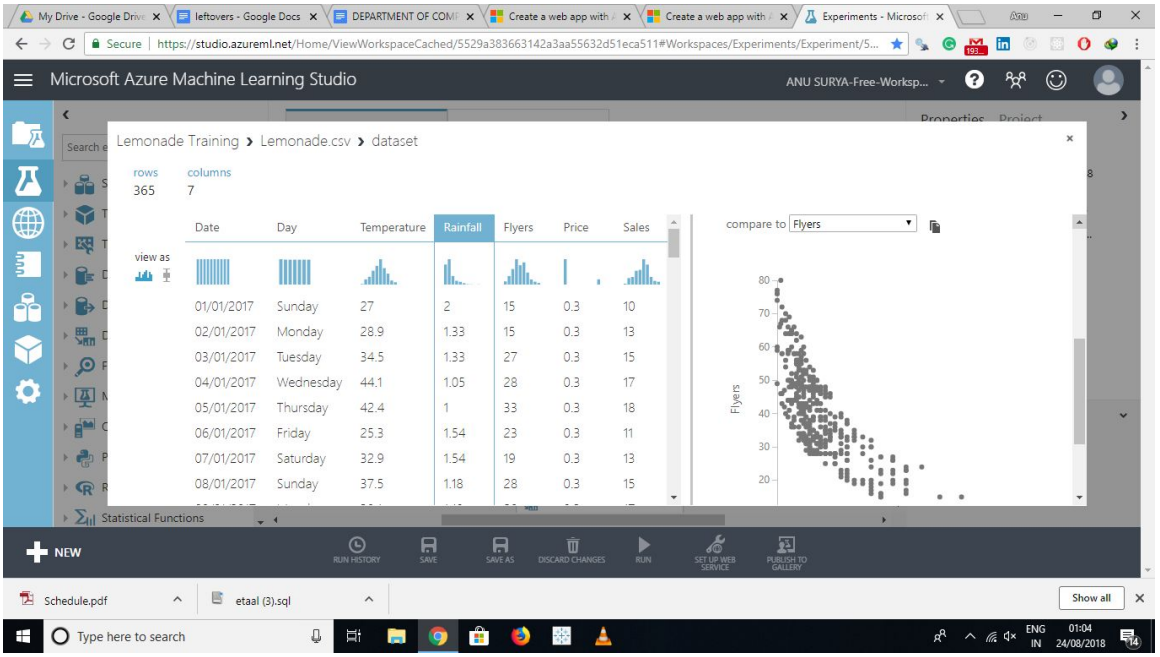
When we want the quickest path to publish our web-based projects, consider Azure App Service. App Service makes it easy to extend our web apps to support our mobile clients and publish easily consumed REST APIs. This platform provides authentication by using social providers, traffic-based auto-scaling, testing in production, and continuous and container-based deployments.

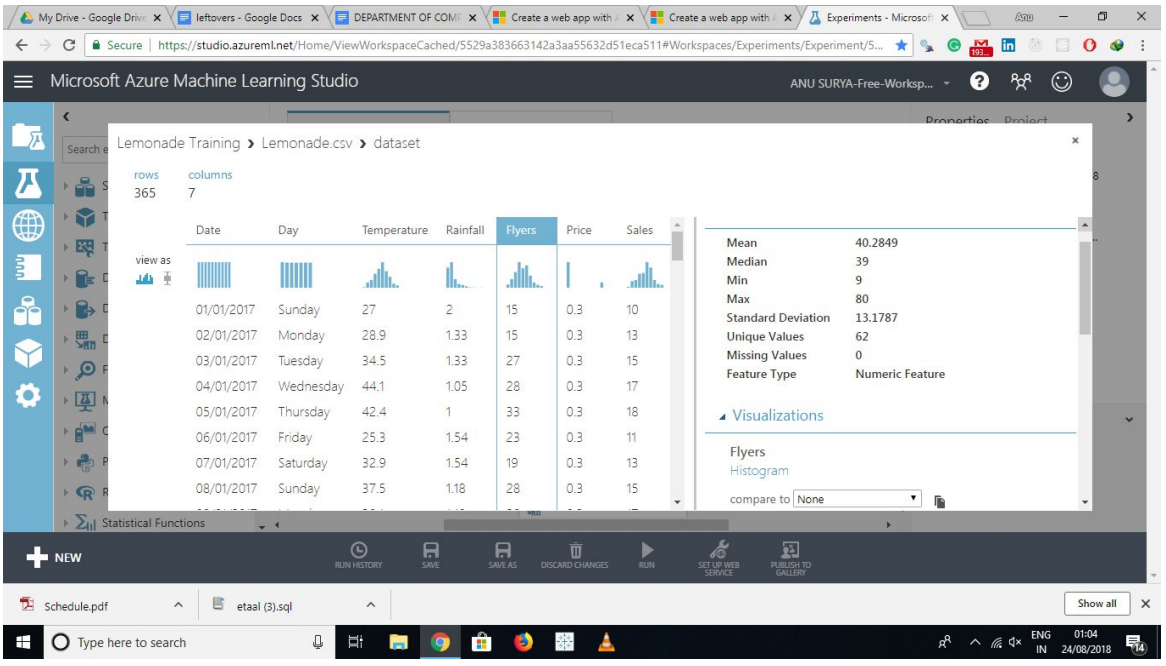
We can create web apps, mobile app back ends, and API apps.

Because all three app types share the App Service runtime, we can host a website, support mobile clients, and expose our APIs in Azure, all from the same project or solution. To learn more about App Service.

App Service has been designed with DevOps in mind. It supports various tools for publishing and continuous integration deployments, including GitHub webhooks, Jenkins, Visual Studio Team Services, TeamCity, and others.




Microsoft Azure Machine Learning Studio
Lemonade Training › Lemonade.csv › dataset
rows 365
columns 7
Date Day Temperature Rainfall Flyers Price Sales
01/01/2017 Sunday 27 2 15 0.3 10
02/01/2017 Monday 28.9 1.33 15 0.3 13
03/01/2017 Tuesday 34.5 1.33 27 0.3 15
04/01/2017 Wednesday 44.1 1.05 28 0.3 17
05/01/2017 Thursday 42.4 1 33 0.3 18
06/01/2017 Friday 25.3 1.54 23 0.3 11
07/01/2017 Saturday 32.9 1.54 19 0.3 13
08/01/2017 Sunday 37.5 1.18 28 0.3 15
compare to Flyers
Flyers
NEW
Schedule.pdf
etaal (3).sql
Show all
Type here to search


Microsoft Azure Machine Learning Studio
Lemonade Training › Lemonade.csv › dataset
rows 365
columns 7
Date Day Temperature Rainfall Flyers Price Sales
01/01/2017 Sunday 27 2 15 0.3 10
02/01/2017 Monday 28.9 1.33 15 0.3 13
03/01/2017 Tuesday 34.5 1.33 27 0.3 15
04/01/2017 Wednesday 44.1 1.05 28 0.3 17
05/01/2017 Thursday 42.4 1 33 0.3 18
06/01/2017 Friday 25.3 1.54 23 0.3 11
07/01/2017 Saturday 32.9 1.54 19 0.3 13
08/01/2017 Sunday 37.5 1.18 28 0.3 15
Mean 40.2849
Median 39
Min 9
Max 80
Standard Deviation 13.1787
Unique Values 62
Missing Values 0
Feature Type Numeric Feature
Visualizations
Flyers
Histogram
compare to None
NEW
Schedule.pdf
etaal (3).sql
Show all
Type here to search

ANACONDA

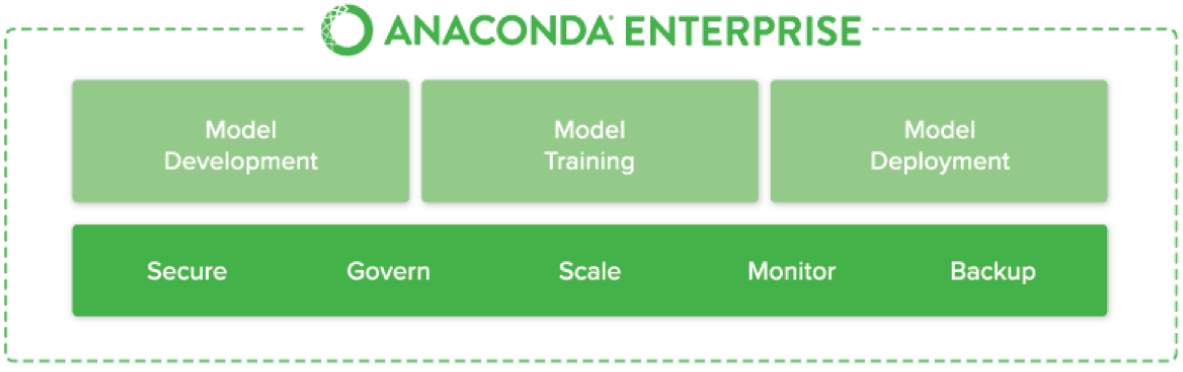
Anaconda Distribution 5 is a free, easy-to-install package manager, environment manager and Python distribution with a collection of 1,000+ open source packages with free community support. Anaconda is platform-agnostic.Anaconda Enterprise is an enterprise-ready, secure and scalable data science platform that empowers teams to govern data science assets, collaborate and deploy data science projects.
Enterprise 5 includes these capabilities:

- Easily deploy our projects into interactive data applications, live notebooks and machine learning models with APIs.
- Share those applications with colleagues and collaborators.
- Manage our data science assets: notebooks, packages, environments and projects in an integrated data science experience.

Anaconda Enterprise 5

Anaconda Enterprise is an enterprise-ready, secure and scalable data science platform that empowers teams to govern data science assets, collaborate and deploy data science projects.
With Anaconda Enterprise, we can:

- Develop: ML/AI pipelines in a central development environment that scales from laptops to thousands of nodes
- Govern: Complete reproducibility from laptop to cluster with the ability to configure access control
- Automate: Model training and deployment on scalable, container-based infrastructure



ML.NET

ML.NET is a free, open-source, and cross-platform machine learning framework that enables we to build custom machine learning solutions and integrate them into our .NET applications. This guide provides many resources about working with ML.NET.
ML.NET will allow .NET developers to develop their own models and infuse custom ML into their applications without prior expertise in developing or tuning machine learning models.

ML.NET was originally developed in Microsoft Research and evolved into a significant framework over the last decade; it is used across many product groups in Microsoft like Windows, Bing, Azure, and more. With this first preview release, ML.NET enables ML tasks like classification (e.g. text categorization and sentiment analysis) and regression (e.g. forecasting and price prediction). Along with these ML capabilities, this first release of ML.NET also brings the first draft of .NET APIs for training models, using models for predictions, as well as the core components of this framework, such as learning algorithms, transforms, and core ML data structures.

ML.NET is first and foremost a framework, which means that it can be extended to add popular ML Libraries like TensorFlow, Accord.NET, and CNTK. We are committed to bringing the full experience of ML.NET's internal capabilities to ML.NET in open source.

To sum it all up, ML.NET is our commitment to make ML great in .NET.

ML.NET Core Components

ML.NET is being launched as a part of the .NET Foundation and the repo today contains the .NET C# API(s) for both model training and consumption, along with a variety of transforms and learners required for many popular ML tasks like regression and classification.

ML.NET is aimed at providing the E2E workflow for infusing ML into .NET apps across pre-processing, feature engineering, modeling, evaluation, and operationalization.

ML.NET comes with support for the types and runtime needed for all aspects of machine learning, including core data types, extensible pipelines, high performance math, data structures for heterogeneous data, tooling support, and more.

The table below describes the entire list of components that are being released as a part of ML.NET 0.1. We aim to make ML.NET's APIs generic, such that other frameworks like CNTK, Accord.NET, TensorFlow and other libraries can become usable through one shared API.

Getting Started Installation

To get started with ML.NET, install the ML.NET NuGet from the CLI using:

dotnet add package Microsoft.ML

From package manager:

Install-Package Microsoft.ML

Sentiment Classification with ML.NET

Train our own model

Here is a simple snippet to train a model for sentiment classification (full snippet of code can be found here).

```
var pipeline = new LearningPipeline();
pipeline.Add(new TextLoader<SentimentData>(dataPath, separator: ","));
pipeline.Add(new TextFeaturizer("Features", "SentimentText"));
pipeline.Add(new FastTreeBinaryClassifier());
pipeline.Add(new PredictedLabelColumnOriginalValueConverter(PredictedLabelColumn =
"PredictedLabel"));

var model = pipeline.Train<SentimentData, SentimentPrediction>();
```

We create a Learning Pipeline which will encapsulate the data loading, data processing/featurization, and learning algorithm. These are the steps required to train a machine learning model which allows us to take the input data and output a prediction.

- The first part of the pipeline is the TextLoader, which loads the data from our training file into our pipeline. We then apply a Text Featurizer to convert the Sentiment Text column into a numeric vector called Features which can be used by the machine learning algorithm (as it cannot take text input). This is our preprocessing/featurization step.
- FastTreeBinary Classifier is a decision tree learner we will use in this pipeline. Like the featurization step, trying out different learners available in ML.NET and changing their parameters may enable identifying better results. PredictedLabelColumnOriginalValueConverter converts the model's predicted labels back to their original value/format.
- pipeline.Train<SentimentData, SentimentPrediction>() trains the pipeline (loads the data, trains the featurizer and learner). The experiment is not executed until this happens.

Use the trained model for predictions

```
SentimentData data = new SentimentData
{
   SentimentText = "Today is a great day!"
};
SentimentPrediction prediction = model.Predict(data);
Console.WriteLine("prediction: " + prediction.Sentiment);
```

To get a prediction, we use model.Predict() on new data. Note that the input data is a string and the model includes the featurization, so our pipeline stays in sync during both training and prediction. We didn't have to write preprocessing/featurization code specifically for predictions.
For more scenarios for getting started please refer to the documentation walkthroughs which go over sentiment analysis and taxi fare prediction in more detail.
There are many capabilities we aspire to add to ML.NET, but we would love to understand what will best fit our needs. The current areas we are exploring are:

Additional ML Tasks and Scenarios

- Deep Learning with TensorFlow & CNTK
- ONNX support
- Scale-out on Azure
- Better GUI to simplify ML tasks
- Integration with VS Tools for AI
- Language Innovation for .NET

Code for Creating a Dummy Dataset

We created a Dummy Dataset as we weren't allowed to disclose the dataset that we were working on and hence are left with the only option to create one for performing the algorithm we used there.

Python Code

```
import pandas
from faker import Factory
import random
from datetime import datetime
import csv

faker = Factory.create()
status = 'created,delivered,returned'.split(',')
Dept_name = 'LM,Labour and Employment,Revenue Department,Tourism
Deartment,NDMC,DIP,IT,Passport Department,LG Delhi Office'.split(',')

def date_between(d1, d2):
   f = '%b%d-%Y'
   return faker.date_time_between_dates(datetime.strptime(d1, f), datetime.strptime(d2, f))



def fakerecord():
   return {'Number of People visited': faker.numerify('####'),
         'insrt_dt': date_between('jan01-2018', 'aug20-2018'),
         'final_transaction status_status': random.choice(status),
         'Department Name': random.choice(Dept_name),
         'No_of_trans':faker.numerify('###')
         }

example_dummy_data = pandas.DataFrame([fakerecord() for _ in range(1000)])

print(example_dummy_data)

example_dummy_data.to_csv('My_Data.csv', sep='\t')
```

ML Code (prototyped)

```
# Multiple Linear Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('My_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values

# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categorical_features = [3])
X = onehotencoder.fit_transform(X).toarray()

# Avoiding the Dummy Variable Trap
X = X[:, 1:]

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)"""

# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)


print(y_pred)
```

DataSet Used

```
12→NDMC→996→6490→delivered→2018-08-04 21:49:05
13→Revenue Department→328→7635→returned→2018-04-28 00:16:24
14→Passport Department→839→8433→returned→2018-02-22 16:15:58
15→LG Delhi Office→238→7679→delivered→2018-08-07 02:03:02
16→NDMC→404→2871→delivered→2018-05-14 00:39:24
17→NDMC→154→4719→created→2018-02-24 12:27:20
18→DIP→433→4777→created→2018-02-12 06:21:36
19→Revenue Department→151→0948→returned→2018-07-14 12:10:01
20→Tourism Deartment→187→2332→returned→2018-04-17 04:24:32
21→Revenue Department→533→4282→delivered→2018-06-19 09:01:24
22→Labour and Employment→022→6957→returned→2018-03-26 07:48:58
23→Revenue Department→254→9732→created→2018-04-14 04:55:19
24→Passport Department→604→8587→created→2018-05-06 04:33:56
25→LG Delhi Office→750→9086→returned→2018-04-18 03:12:13
26→Passport Department→069→8060→delivered→2018-06-13 13:44:23
27→NDMC→064→8313→created→2018-04-24 00:33:49
28→Labour and Employment→203→1671→delivered→2018-05-28 22:05:43
29→LM→480→1325→delivered→2018-07-09 05:18:46
30→Tourism Deartment→155→4272→delivered→2018-06-24 22:33:24
31→LG Delhi Office→590→1485→delivered→2018-03-17 04:37:58
32→Revenue Department→262→5129→returned→2018-06-30 03:20:50
33→NDMC→508→6121→returned→2018-03-31 15:57:30
34→Passport Department→173→0383→created→2018-04-27 07:01:47
35→DIP→453→6233→created→2018-03-30 10:02:07
36→LG Delhi Office→712→9877→delivered→2018-02-16 14:32:03
37→Tourism Deartment→086→6861→returned→2018-04-30 08:02:47
38→IT→476→4369→delivered→2018-01-30 16:42:46
39→NDMC→654→6956→returned→2018-05-25 19:42:38
40→NDMC→690→8960→created→2018-04-05 21:31:34
41→Revenue Department→476→2063→created→2018-07-13 22:38:23
42→LM→189→5165→delivered→2018-04-06 06:21:52
```

OUTPUT

```
# Avoiding the Dummy Variable Trap
X = X[:, 1:]

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)"""

# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)


print(y_pred)
```

```
[103015.20159796 132582.27760815 132447.73845175  71976.09851258
 178537.48221056 116161.24230166  67851.69209676  98791.73374687
 113969.43533013 167921.06569551]
```

TABLE OF CONTENTS III
(for Data Science)

Data Science

Data science is a multidisciplinary blend of data inference, algorithm development, and technology in order to solve analytically complex problems.

Data science – discovery of data insight

This aspect of data science is all about uncovering findings from data. Diving in at a granular level to mine and understand complex behaviors, trends, and inferences. It's about surfacing hidden insight that can help enable companies to make smarter business decisions.

How do data scientists mine out insights?

It starts with data exploration. When given a challenging question, data scientists become detectives. They investigate leads and try to understand pattern or characteristics within the data. This requires a big dose of analytical creativity.Then as needed, data scientists may apply quantitative technique in order to get a level deeper – e.g. inferential models, segmentation analysis, time series forecasting, synthetic control experiments, etc. The intent is to scientifically piece together a forensic view of what the data is really saying.

This data-driven insight is central to providing strategic guidance. In this sense, data scientists act as consultants, guiding business stakeholders on how to act on findings.

Data science – development of data product

A "data product" is a technical asset that:

(1) utilizes data as input

(2) processes that data to return algorithmically-generated results.

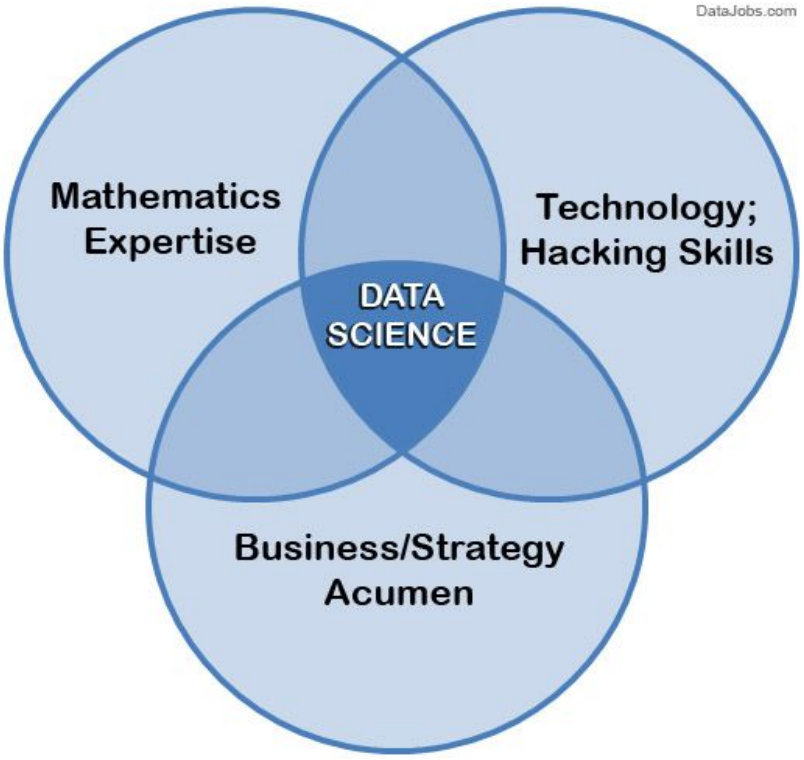
The classic example of a data product is a recommendation engine, which ingests user data, and makes personalized recommendations based on that data.

This is different from the "data insights" section above, where the outcome to that is to perhaps provide advice to an executive to make a smarter business decision. In contrast, a data product is technical functionality that encapsulates an algorithm, and is designed to integrate directly into core applications. Respective examples of applications that incorporate data product behind the scenes: Amazon's homepage, Gmail's inbox, and autonomous driving software.

Data scientists play a central role in developing data product. This involves building out algorithms, as well as testing, refinement, and technical deployment into production systems. In this sense, data scientists serve as technical developers, building assets that can be leveraged at wide scale.

The requisite skill set

Data science is a blend of skills in three major areas:



Mathematics Expertise

At the heart of mining data insight and building data product is the ability to view the data through a quantitative lens. There are textures, dimensions, and correlations in data that can be expressed mathematically. Finding solutions utilizing data becomes a brain teaser of heuristics and quantitative technique. Solutions to many business problems involve building analytic models grounded in the hard math, where being able to understand the underlying mechanics of those models is key to success in building them.

Also, a misconception is that data science all about statistics. While statistics is important, it is not the only type of math utilized.

First, there are two branches of statistics –

1. Classical statistics
2. Bayesian statistics.

When most people refer to *stats* they are generally referring to *classical stats*, but knowledge of both types is helpful. Furthermore, many inferential techniques and machine learning algorithms lean on knowledge of linear algebra.

What is a data scientist – curiosity and training

The Mindset

A common personality trait of data scientists is they are deep thinkers with *intense intellectual curiosity*. Data science is all about being inquisitive – asking new questions, making new discoveries, and learning new things. Ask data scientists most obsessed with their work what drives them in their job, and they will not say "money". The real motivator is being able to use their creativity and ingenuity to solve hard problems and constantly indulge in their curiosity. Deriving complex reads from data is beyond just

making an observation, it is about uncovering "truth" that lies hidden beneath the surface. Problem solving is not a task, but an intellectually-stimulating journey to a solution. Data scientists are passionate about what they do, and reap great satisfaction in taking on challenge.

Training

There is a glaring misconception out there that you need a sciences or math Ph.D to become a legitimate data scientist. That view misses the point that data science is multidisciplinary. Highly-focused study in academia is certainly helpful, but doesn't guarantee that graduates have the full set of experiences and abilities to succeed.

What is Analytics?

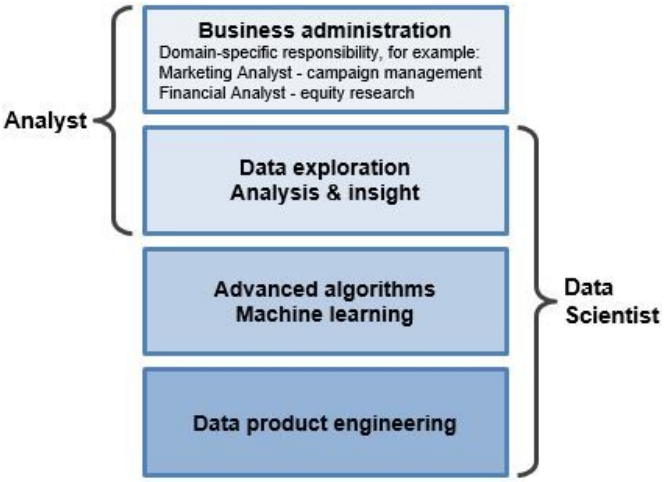
Analytics has risen quickly in popular business lingo over the past several years; the term is used loosely, but generally meant to describe critical thinking that is quantitative in nature. Technically, analytics is the "science of analysis" — put another way, the practice of analyzing information to make decisions.
Is "analytics" the same thing as data science? Depends on context. Sometimes it is synonymous with the definition of data science that we have described, and sometimes it represents something else. A data scientist using raw data to build a predictive algorithm falls into the scope of analytics. At the same time, a non-technical business user interpreting pre-built dashboard reports is also in the realm of analytics, but does not cross into the skill set needed in data science. Analytics has come to have fairly broad meaning. At the end of the day, as long as you understand beyond the buzzword level, the exact semantics don't matter much.

What is the difference between an analyst and a data scientist?

"Analyst" is somewhat of an ambiguous job title that can represent many different types of roles (data analyst, marketing analyst, operations analyst, financial analyst, etc). What does this mean in comparison to data scientist?

- ***Data Scientist****:* Specialty role with abilities in math, technology, and business acumen. Data scientists work at the raw database level to derive insights and build data product.
- ***Analyst****:* This can mean a lot of things. Common thread is that analysts look at data to try to gain insights. Analysts may interact with data at both the database level or the summarized report level.

Thus, "analyst" and "data scientist" is not exactly synonymous, but also not mutually exclusive. Here is our interpretation of how these job titles map to skills and scope of responsibilities:



Machine Learning

Machine learning is a term closely associated with data science. It refers to a broad class of methods that revolve around data modeling to
(1) Algorithmically make prediction
(2) Algorithmically decipher patterns in data.

- **Machine learning for making predictions —** Core concept is to use tagged data to train predictive models. *Tagged data* means observations where ground truth is already known. *Training models* means automatically characterizing tagged data in ways to predict tags for unknown data points. E.g. a credit card fraud detection model can be trained using a historical record of tagged fraud purchases. The resultant model estimates the likelihood that any new purchase is fraudulent.
- **Machine learning for pattern discovery —** Another modeling paradigm known as *unsupervised learning* tries to surface underlying patterns and associations in data when no existing ground truth is known (i.e. no observations are tagged). Within this broad category of methods, the most commonly used are clustering techniques, which algorithmically detect what are the natural groupings that exist in a data set.

Not all machine learning methods fit neatly into the above two categories.
This wide-ranging breadth of machine learning techniques comprise an important part of the data science toolbox. It is up to the data scientist to figure out which tool to use in different circumstances (as well as how to use the tool correctly) in order to solve analytically open-ended problems.

Data Munging

Raw data can be unstructured and messy, with information coming from disparate data sources, mismatched or missing records, and a slew of other tricky issues. Data munging is a term to describe the data wrangling to bring together data into cohesive views, as well as the janitorial work of cleaning up data so that it is polished and ready for downstream usage. This requires good pattern-recognition sense and clever hacking skills to merge and transform masses of database-level information. If not properly done, dirty data can obfuscate the 'truth' hidden in the data set and completely mislead results. Thus, any data scientist must be skillful and nimble at data munging in order to have accurate, usable data before applying more sophisticated analytical tactics.

Lifecycle of Data Science Project

1) Data Acquisition

For doing Data Science, you need data. The primary step in the lifecycle of data science projects is to first identify the person who knows what data to acquire and when to acquire based on the question to be answered. The person need not necessarily be a data scientist but anyone who knows the real difference between the various available data sets and making hard-hitting decisions about the data investment strategy of an organization – will be the right person for the job.

Data science project begins with identifying various data sources which could be –logs from web servers, social media data, data from online repositories like US Census datasets, data streamed from online sources via APIs, web scraping or data could be present in an excel or can come from any other source. Data acquisition involves acquiring data from all the identified internal and external sources that can help answer the business question.

A major challenge that data professionals often encounter in data acquisition step is tracking where each data slice comes from and whether the data slice acquired is up-to-date or not. It is important to track this information during the entire lifecycle of a data science project as data might have to be re-acquired to test other hypothesis or run any other updated experiments.

2) Data Preparation

Often referred as data cleaning or data wrangling phase. Data scientists often complain that this is the most boring and time consuming task involving identification of various data quality issues. Data acquired in the first step of a data science project is usually not in a usable format to run the required analysis and might contain missing entries, inconsistencies and semantic errors.

Having acquired the data, data scientists have to clean and reformat the data by manually editing it in the spreadsheet or by writing code. This step of the data science project lifecycle does not produce any meaningful insights. However, through regular data cleaning, data scientists can easily identify what foibles exists in the data acquisition process, what assumptions they should make and what models they can apply to produce analysis results. Data after reformatting can be converted to JSON, CSV or any other format that makes it easy to load into one of the data science tools.

Exploratory data analysis forms an integral part at this stage as summarization of the clean data can help identify outliers, anomalies and patterns that can be usable in the subsequent steps. This is the step that helps data scientists answer the question on as to what do they actually want to do with this data.

“Exploratory data analysis” is an attitude, a state of flexibility, a willingness to look for those things that we believe are not there, as well as those we believe to be there. — said John Tukey, an American Mathematician

3) Hypothesis and Modelling

This is the core activity of a data science project that requires writing, running and refining the programs to analyse and derive meaningful business insights from data. Often these programs are written in

languages like Python, R, MATLAB or Perl. Diverse machine learning techniques are applied to the data to identify the machine learning model that best fits the business needs. All the contending machine learning models are trained with the training data sets.

4) Evaluation and Interpretation

There are different evaluation metrics for different performance metrics. For instance, if the machine learning model aims to predict the daily stock then the RMSE (root mean squared error) will have to be considered for evaluation. If the model aims to classify spam emails then performance metrics like average accuracy, AUC and log loss have to be considered. A common question that professionals often have when evaluating the performance of a machine learning model is that which dataset they should use to measure the performance of the machine learning model. Looking at the performance metrics on the trained dataset is helpful but is not always right because the numbers obtained might be overly optimistic as the model is already adapted to the training dataset. Machine learning model performances should be measured and compared using validation and test sets to identify the best model based on model accuracy and over-fitting.

All the above steps from 1 to 4 are iterated as data is acquired continuously and business understanding become much clearer.

5) Deployment

Machine learning models might have to be recoded before deployment because data scientists might favour Python programming language but the production environment supports Java. After this, the machine learning models are first deployed in a pre-production or test environment before actually deploying them into production.

6) Operations/Maintenance

This step involves developing a plan for monitoring and maintaining the data science project in the long run. The model performance is monitored and performance downgrade is clearly monitored in this phase. Data scientists can archive their learnings from a specific data science projects for shared learning and to speed up similar data science projects in near future.

7) Optimization

This is the final phase of any data science project that involves retraining the machine learning model in production whenever there are new data sources coming in or taking necessary steps to keep up with the performance of the machine learning model.

Having a well-defined workflow for any data science project is less frustrating for any data professional to work on. The lifecycle of a data science project mentioned above is not definitive and can be altered accordingly to improve the efficiency of a specific data science project as per the business requirements.

INTRODUCTION TO R

R is a programming language and software environment for statistical analysis, graphics representation and reporting. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team.

The core of R is an interpreted computer language which allows branching and looping as well as modular programming using functions. R allows integration with the procedures written in the C, C++, .Net, Python or FORTRAN languages for efficiency.

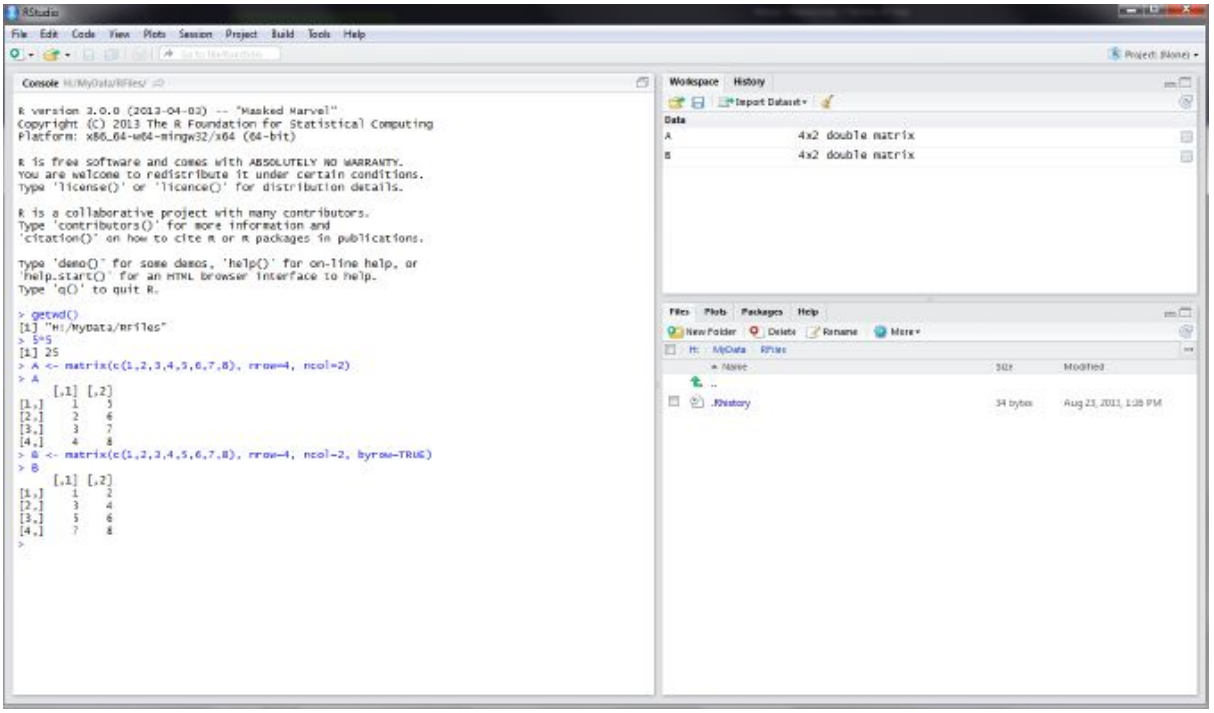
R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac.

Introduction to RStudio (v 1.3)

Introduction

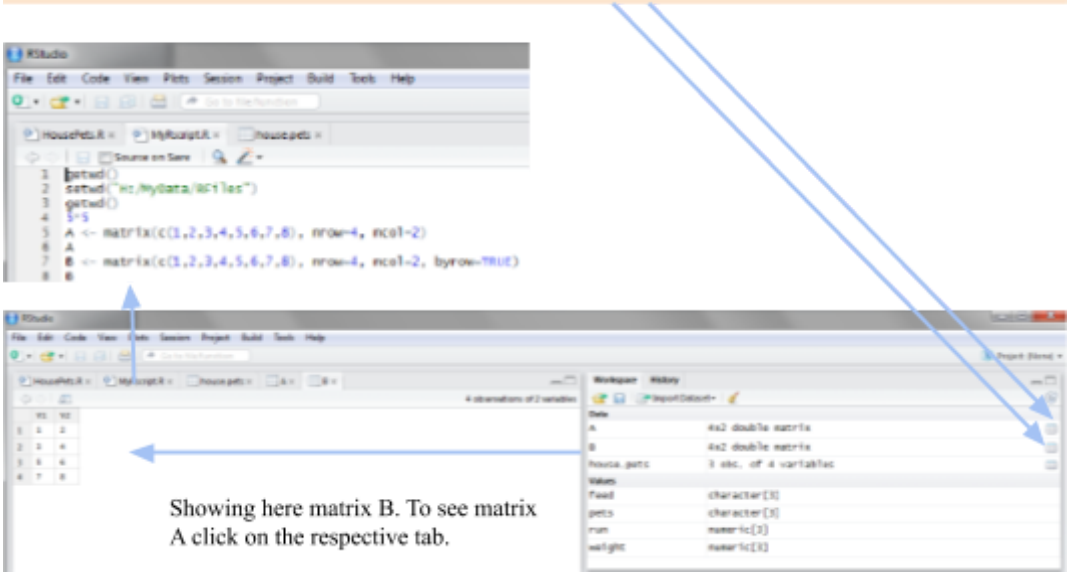
RStudio allows the user to run R in a more user-friendly environment. It is open-source (i.e. free) and available at http://www.rstudio.com/

RStudio screen



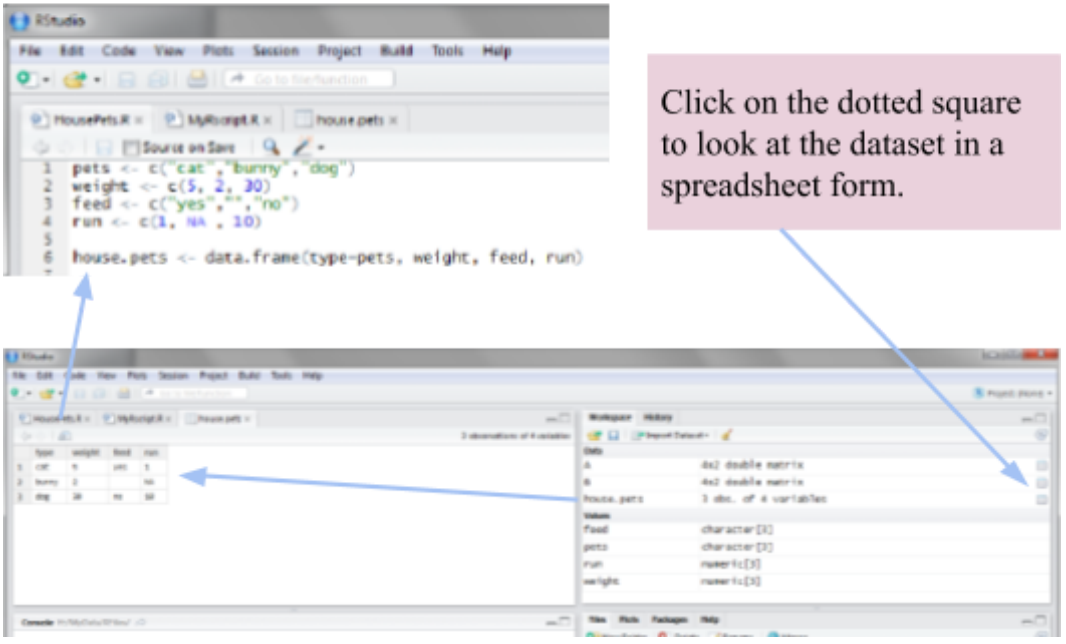
Workspace tab (1)

The workspace tab stores any object, value, function or anything you create during your R session. In the example below, if you click on the dotted squares you can see the data on a screen to the left.



Workspace tab (2)

Here is another example on how the workspace looks like when more objects are added. Notice that the data frame house.pets is formed from different individual values or vectors.

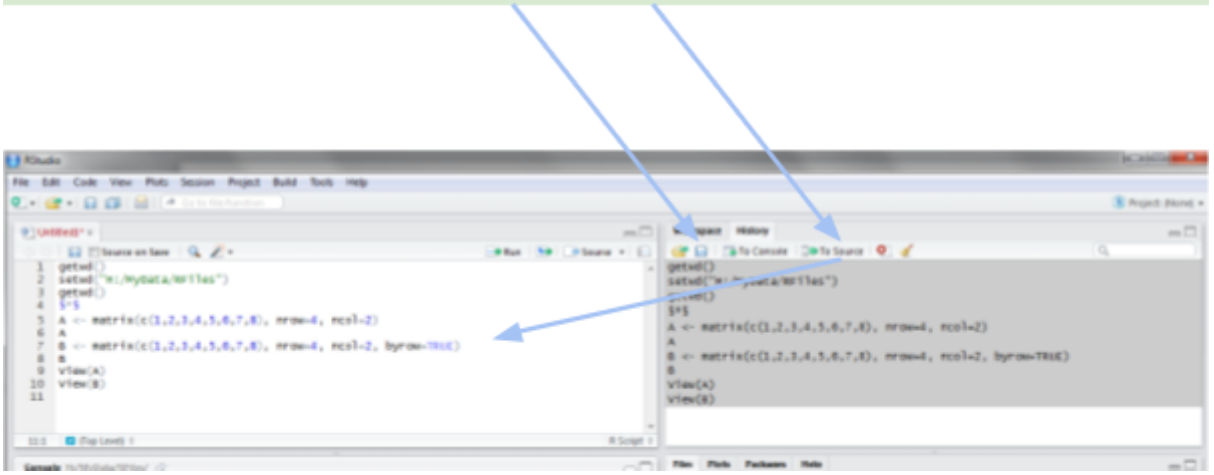


History tab

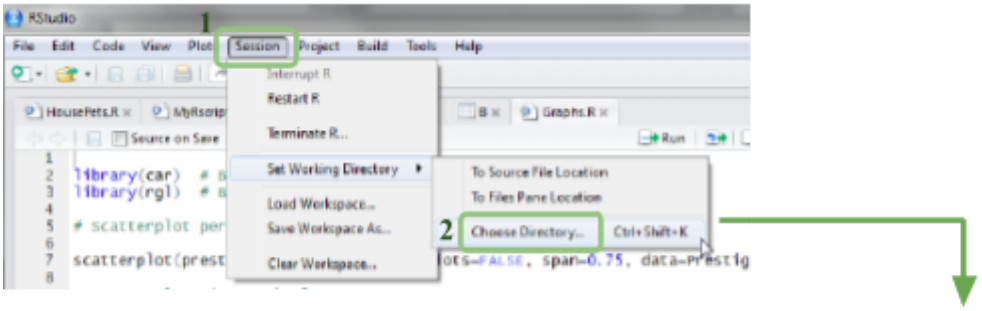
History tab

The history tab keeps a record of all previous commands. It helps when testing and running processes. Here you can either save the whole list or you can select the commands you want and send them to an R script to keep track of your work.

In this example, we select all and click on the "To Source" icon, a window on the left will open with the list of commands. Make sure to save the 'untitled1' file as an *.R script.



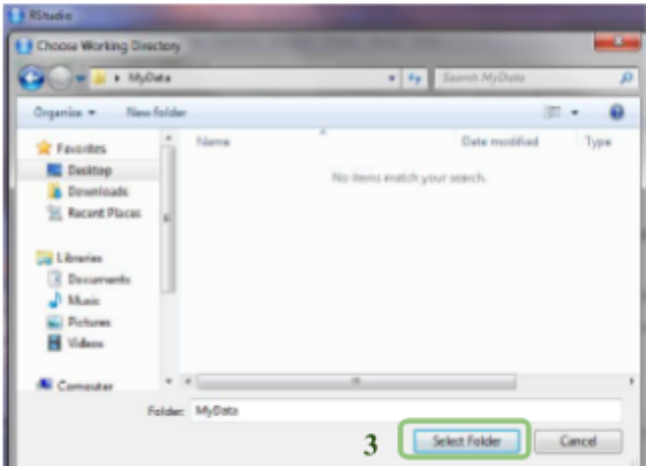
Changing the working directory



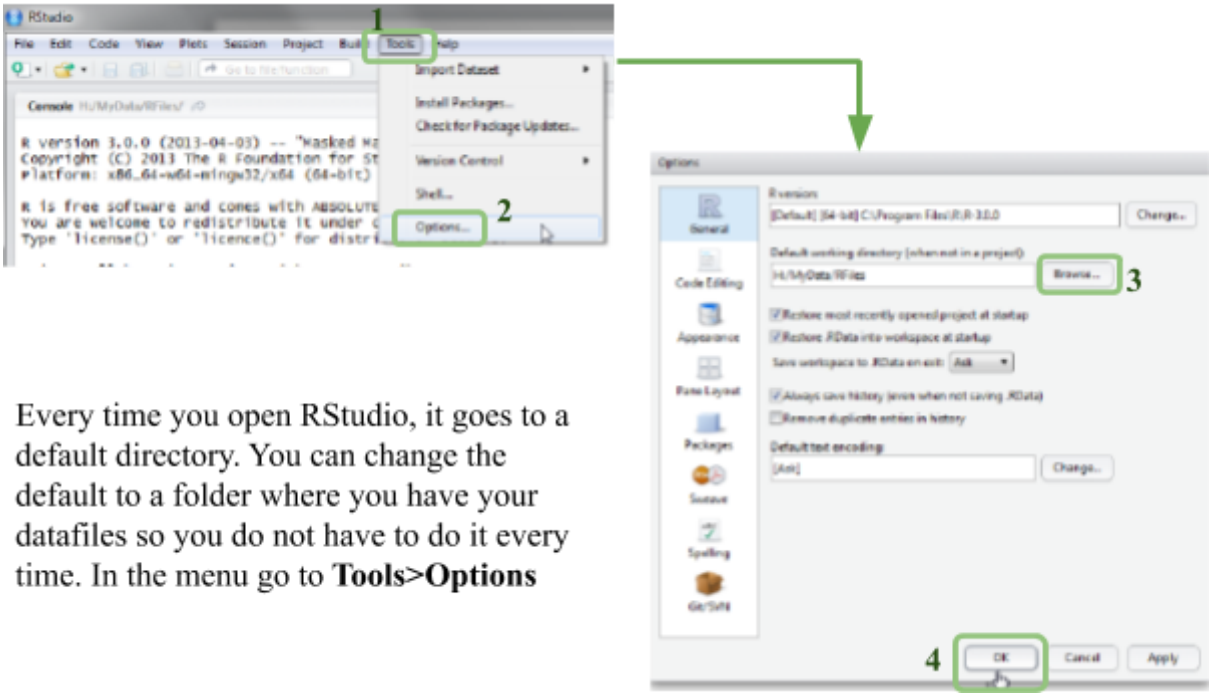
If you have different projects you can change the working directory for that session, see above. Or you can type:

```
# Shows the working directory (wd)
getwd()
# Changes the wd
setwd("C:/myfolder/data")
```

More info see the following document:
http://dss.princeton.edu/training/RStata.pdf



Setting a default working directory



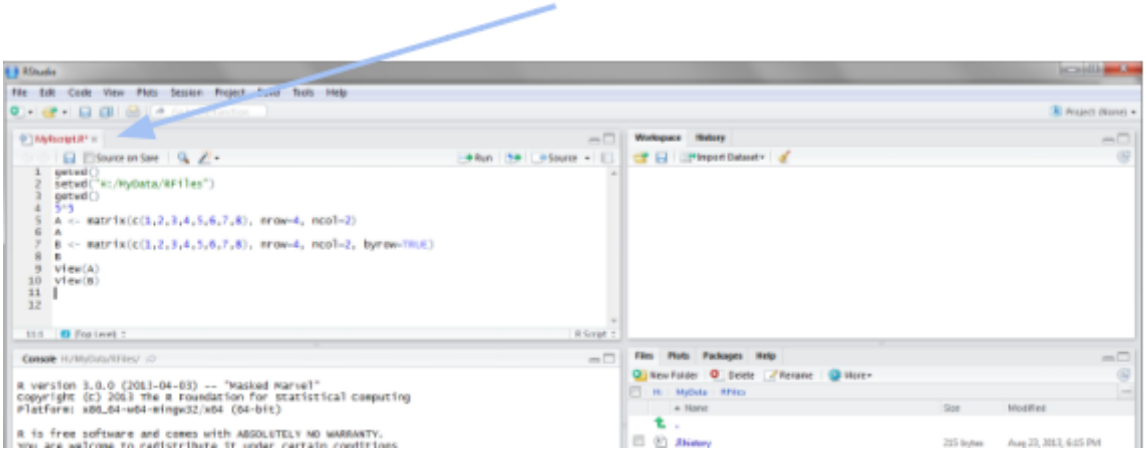
Every time you open RStudio, it goes to a default directory. You can change the default to a folder where you have your datafiles so you do not have to do it every time. In the menu go to **Tools>Options**

R script (1)

The usual Rstudio screen has four windows:

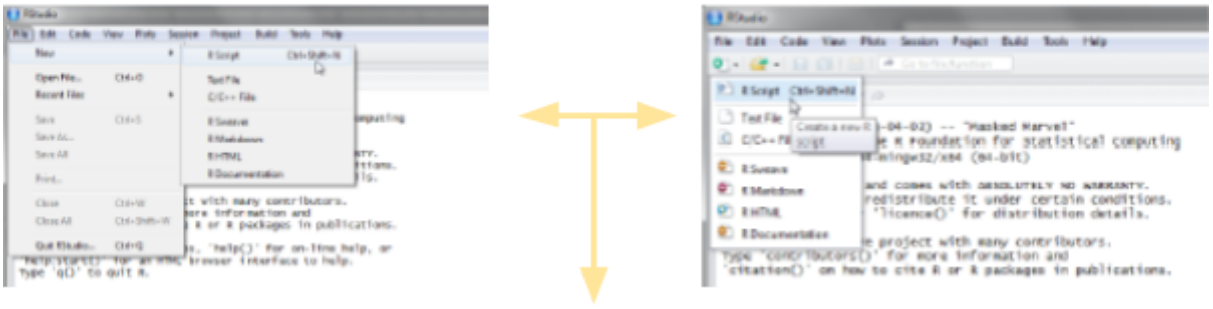
1. Console.
2. Workspace and history.
3. Files, plots, packages and help.
4. The R script(s) and data view.

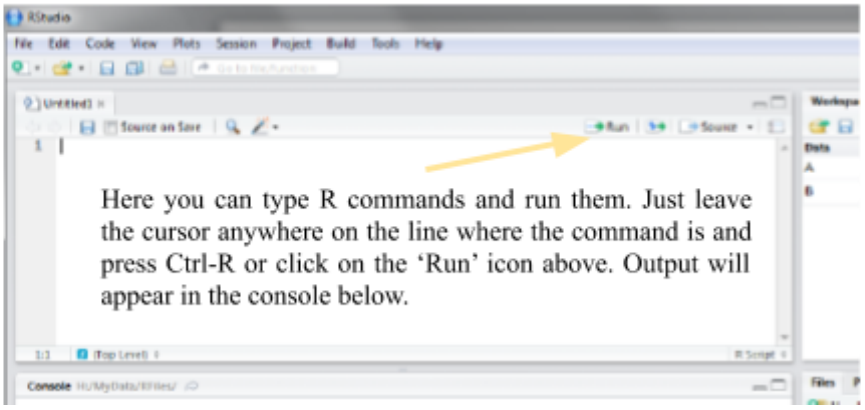
The R script is where you keep a record of your work. For Stata users this would be like the do-file, for SPSS users is like the syntax and for SAS users the SAS program.



R script (2)

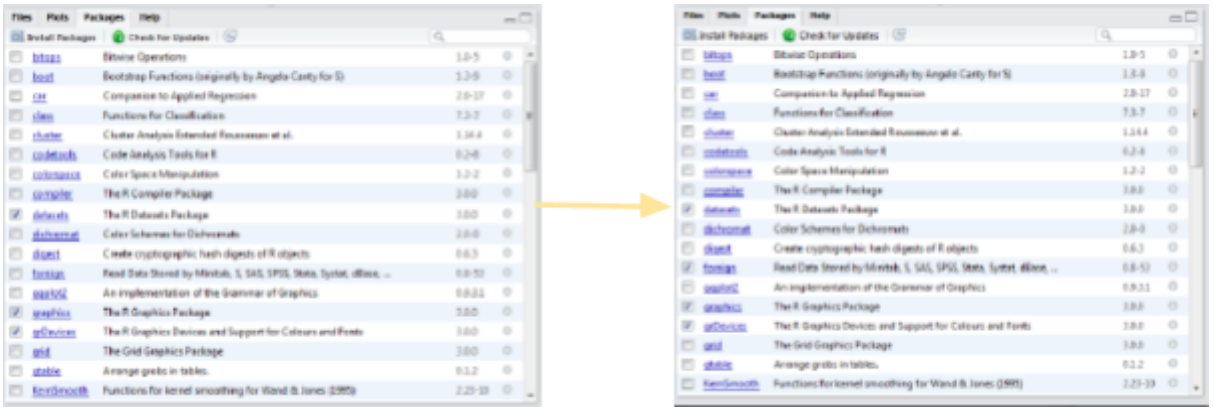
To create a new R script you can either go to File -> New -> R Script, or click on the icon with the "+" sign and select "R Script", or simply press Ctrl+Shift+N. Make sure to save the script.



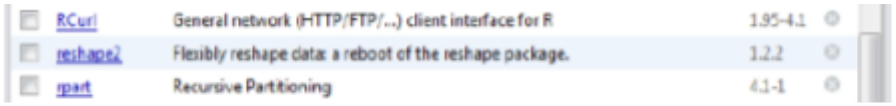


Packages tab

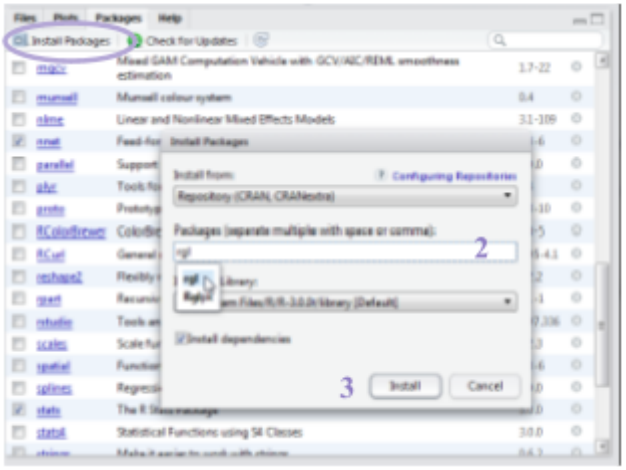
The package tab shows the list of add-ons included in the installation of RStudio. If checked, the package is loaded into R, if not, any command related to that package won't work, you will need select it. You can also install other add-ons by clicking on the 'Install Packages' icon. Another way to activate a package is by typing, for example, library(foreign). This will automatically check the --foreign package (it helps bring data from proprietary formats like Stata, SAS or SPSS).



Installing a package

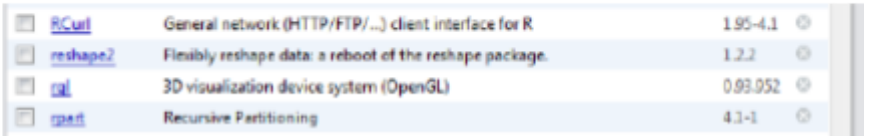


Before

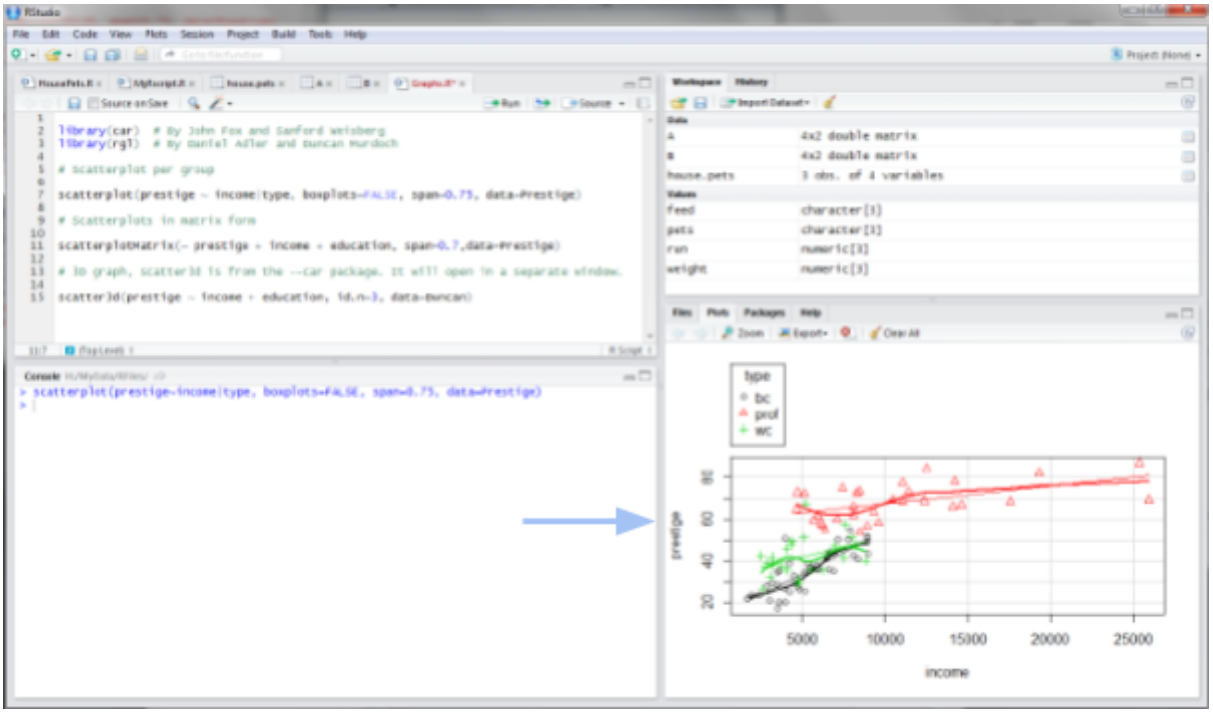


We are going to install the package –rgl (useful to plot 3D images). It does not come with the original R install.
Click on “Install Packages”, write the name in the pop-up window and click on “Install”.

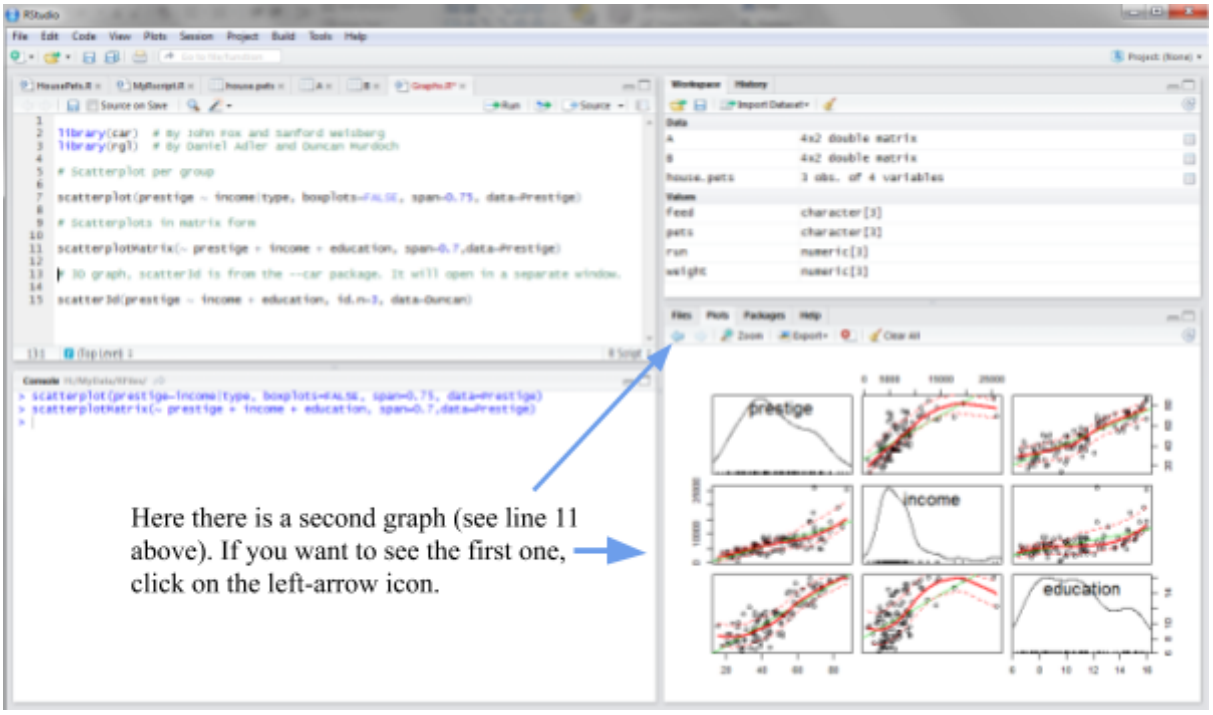
After



Plots tab (1)

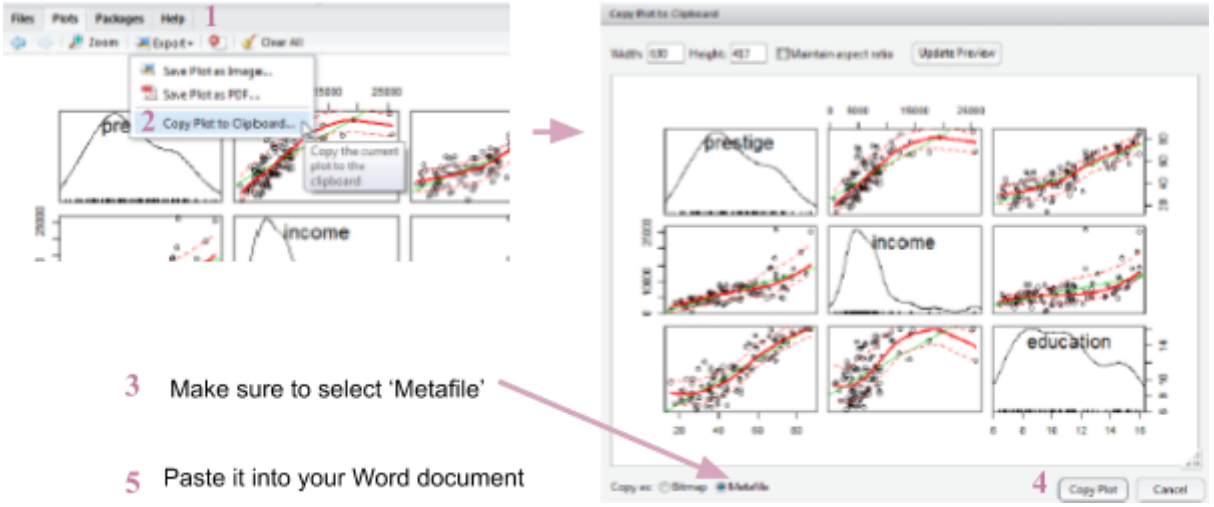


Plots tab (2)

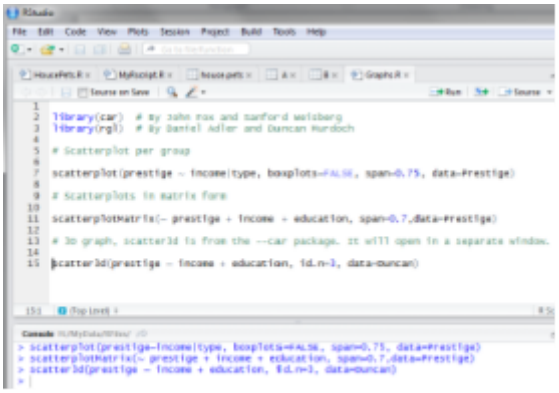


Plots tab (3) – Graphs export

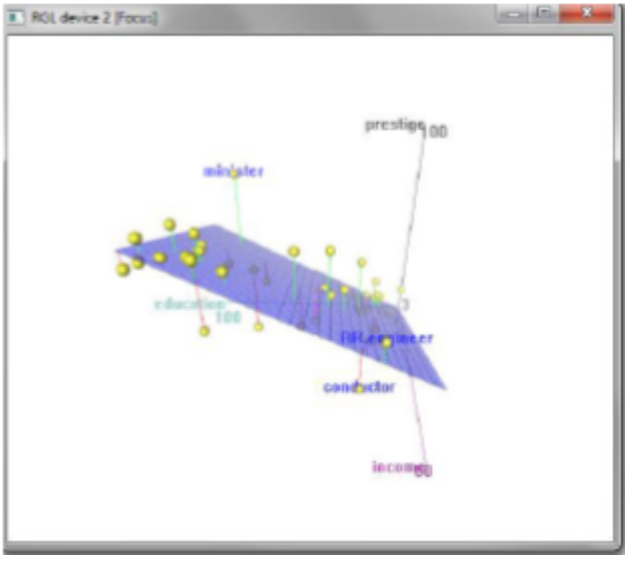
To extract the graph, click on "Export" where you can save the file as an image (PNG, JPG, etc.) or as PDF, these options are useful when you only want to share the graph or use it in a LaTeX document. Probably, the easiest way to export a graph is by copying it to the clipboard and then paste it directly into your Word document.



3D graphs



3D graphs will display on a separate screen (see line 15 above). You won't be able to save it, but after moving it around, once you find the angle you want, you can screenshot it and paste it to you Word document.



Installation & Commands

Installing R on Your Windows Computer

Install R for windows from https://cran.r-project.org/bin/windows/base/

Installing R-Studio on Your Windows Computer

Install R-Studio from https://www.rstudio.com/products/rstudio/download/

The Help Command in R

R contains a lot of built-in help, and how this is displayed varies according to which OS you are using and the options (if any) that you set.

The basic command to bring up help is:

help(topic)

Simply replace topic with the name of the item you want help on. You can also save a bit of typing by prefacing the topic with a question mark, like so:

?topic

Search Command

The R program is built from a series of modules, called packages. These packages are libraries of commands that undertake various functions. When you first start R several packages are loaded on your computer and become ready for use. You can see what is available by using the search() command like so:

>search()

[1] ".GlobalEnv" "tools:RGUI" "package:stats" "package:graphics"

[5] "package:grDevices" "package:utils" "package:datasets" "package:methods"

[9] "Autoloads" "package:base"

Installing Extra Packages for Windows Users

In Windows you can use the Packages menu. You have several options, but Install Package(s) is the one you will want most often.

install.packages("rattle")

Loading Packages

It is simple to load packages as required. Start by issuing the following command:

library(package)

The library() command retrieves the appropriate package and makes its contents available for you (think of it as adding to your library of available commands).

Removing or Unloading Packages

If you have loaded some packages and want to remove one, perhaps to free up an overwritten command, you can use the detach() command like so:

detach(package:name)

You simply replace the name part with the name of the package that you want to remove. Once removed, a package is not totally gone; you can still use the library() command to get it back when required.

Getting the directory for R

getwd()

Setting the directory for R

setwd(path)

eg:- path-E:/

Variables

A variable provides us with named storage that our programs can manipulate. A variable in R can store an atomic vector, group of atomic vectors or a combination of many Robjects. A valid variable name consists of letters, numbers and the dot or underline characters. The variable name starts with a letter or the dot not followed by a number.

Variable Assignment

The variables can be assigned values using leftward, rightward and equal to operator. The values of the variables can be printed using **print()** or **cat()**function. The **cat()** function combines multiple items into a continuous print output.

```
# Assignment using equal operator.
var.1=c(0,1,2,3)
 # Assignment using leftward operator.
var.2<- c("learn","R")
 # Assignment using rightward operator.
c(TRUE,1)->var.3
```

Deleting Variables

Variables can be deleted by using the **rm()** function. Below we delete the variable var.3. On printing the value of the variable error is thrown.

```
rm(var.3)
print(var.3)
```

Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. R language is rich in built-in operators and provides following types of operators.

Types of Operators

We have the following types of operators in R programming −

- Arithmetic Operators(+,-,*,/,%%,%/%,^)
- Relational Operators(<,>,<=,>=,==,!=)
- Logical Operators(&,!,|,&&,||)
- Assignment Operators(<-,=,->)
- Miscellaneous Operators(%in%,:,%*%)

SENTIMENTAL ANALYSIS

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral.It's also known as opinion mining, deriving the opinion or attitude of a speaker. A common use case for this technology is to discover how people feel about a particular topic.

CODE FOR SENTIMENTAL ANALYSIS

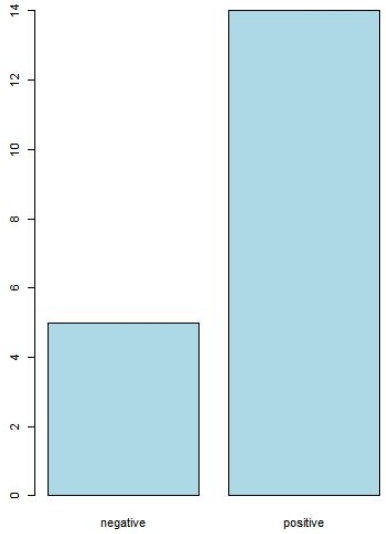
Here, we analysed the people reviews about government project and analysed it closely.

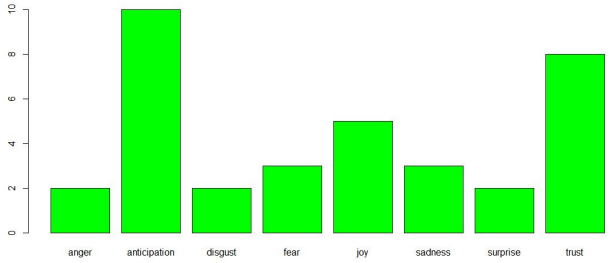
R CODE:

```
library(rvest)
library(stringr)
library(readr)
library(tm)
library(RSentiment)
library(wordcloud)
library(jpeg)
library(syuzhet)
library(XML)
library(ggplot2)

data_cnv<-function(t)
{
 #Convert multiple Lines into One Text
 text1 = paste(t,collapse = "")
 #1. Remove Puntuations
 text1 = gsub(pattern="\\W", replace = " ", text1)
 #2. Remove Numbers or Digits
 text1 = gsub(pattern="\\d", replace = " ", text1)
 #3. Convert entire text to lowercase
 text1 = tolower(text1)
 #5. Removing Certain Words(StopWords)
 #stopwords()
 text1 = removeWords(text1,stopwords())
 #6. Removing Words of Certain Length
 text1 = gsub(pattern="\\b[A-z]\\b{1}", replace=" ", text1)
 #7. Cleaning up all White Space
 text1 = stripWhitespace(text1)
 #Sentiment Analysis

 #S+ One or More Space
 tb = str_split(text1, pattern="\\s+")
 tb1 = unlist(tb)
 cat("\n \n")
 #Creating WordCloud
```

```
  wordcloud(tb1,min.freq = 1,random.order = FALSE,scale=c(3,0.5),color = rainbow(3))
  #to lower
  text1=tolower(text1)

    #Calculating Polarity and Sentiment Score
  g_score = get_nrc_sentiment(text1)

  g_polarity = g_score[1,9:10]
  g_polarity = data.matrix(g_polarity, rownames.force = TRUE)
  print(g_polarity)

  g_sentiment = g_score[1,1:8]
  g_sentiment = data.matrix(g_sentiment,rownames.force = TRUE)
  print(g_sentiment)

#Visualizing a WordCloud
  wordcloud(tb1,min.freq = 1,random.order = FALSE,scale=c(3,0.5),color = rainbow(3))
  #Visualize Polarity and Sentiment
  barplot(g_polarity,col ='lightblue')
  barplot(g_sentiment,col = 'green')
}

public_rev<-function()
{
cat("Public Review On Government Project")

Public_review<-read_html("https://www.glassdoor.co.in/Reviews/Government-of-India-Reviews-E3355
50.htm")
reviews_1_5<-Public_review%>%html_nodes(".reviewBodyCell")%>%html_text()
write(reviews_1_5, " pub.txt", sep=" ")
Public_review2 =readLines(" pub.txt ")
#DATA CLEANING
data_cnv( Public_review2) #calling data cleaning function
}
public_rev()
```

OUTPUTS:


0
2
4
6
8
10
12
14
negative
positive


0
2
4
6
8
10
anger
anticipation
disgust
fear
joy
sadness
surprise
trust


timeprosgreat
securityshare
salaryless
responsibilitiesconsprolonged
managementgreat
peopleconslow
lowstagnation
structure
power
hierarchy
consperformance
challenges
playnetworking
lateral
average
augmentation
positions
top
future
prosadvice
see
revenueconsno
charge
working
india
whatsappshare
prosgood
bosses
access
meritrigid
interaction
interactworked
duty
servicecomfort
concept
response
curve
prosyour
support
almost
work
balanceconsi
based
prosyou
delay
done
pdfi
cons
emailcopy
consa
prospresting
doesn
ceoi
time
ago
flag
via
daily
phasesconsodd
amazing
new
basis
nil
take
lovesanyone
individual
directly
government
level
downside
timings
gst
entryget
linklink
full
copied
years
job
terms
much
delete
edit
twittershare
bad
maintain
office
facebookshare
seconds
keen
nothing
pressure
inappropriateflag
salary
govt
leaves
inappropriatehelpful
good
organization
travelshare
hygieneshare
outlookapproves
share
career
learner
salaryshare
associated
postingdomain
clearanceadvice
hereadvice
progress
timeproseasy
experienceconsplace
learnshare
pressureshare
systemstability
opportunity
pressuresecurityconslearning
postingrecognition
recommendspositive
timeprosvery