##### SATELLITE IMAGE ANALYSIS FOR PREDICTING FOOD SUPPLY IN INDIA

##### A MINI PROJECT REPORT

###### ***Submitted by***

##### AKSHAY KUMAR(RA1511008010220)

**UJJWAL VATS(RA1511008010228)**

##### RISHABH PRABHAKAR(RA1511008010246)

**GAURAV BAJPAI(RA1511008010252)**

***in partial fulfillment for the award of the degree***

***of***

##### B.Tech

IN

INFORMATION TECHNOLOGY



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

April,2018

##### SATELLITE IMAGE ANALYSIS FOR PREDICTING FOOD SUPPLY IN INDIA

##### A MINI PROJECT REPORT

###### ***Submitted by***

##### AKSHAY KUMAR(RA1511008010220)

**UJJWAL VATS(RA1511008010228)**

##### RISHABH PRABHAKAR(RA1511008010246)

**GAURAV BAJPAI(RA1511008010252)**

***in partial fulfillment for the award of the degree***

***of***

##### B.Tech

IN

INFORMATION TECHNOLOGY



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

April,2018

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

**BONAFIDE CERTIFICATE**

Certified that this project report **“Satellite Image Analysis for Predicting Food Supply”**is the bonafide work of “**AKSHAY KUMAR,UJJWAL VATS, RISHABH PRABHAKAR and GAURAV BAJPAI”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Guide Name**

**HEAD OF THE DEPARTMENT Information Technology**

INTERNAL EXAMINER

**DECLARATION**

We **AKSHAY KUMAR,UJJWAL VATS, RISHABH PRABHAKAR and GAURAV BAJPAI** studying in III year B.Tech Information Technology program at, SRM Institute of Science and Technology, Kattankulathur, Chennai, hereby declare that this project is an original work of mine and I have not verbatim copied / duplicated any material from sources like internet or from print media, excepting some vital company information / statistics and data that is provided by the company itself.

Signature of the Student

Date:

Place:

**ACKNOWLEDGEMENT**

The success and the final outcome of this project required guidance and assistance from different sources and we feel extremely fortunate to have got this all along the completion of our project. Whatever we have done is largely due to such guidance and assistance and we would not forget to thank them.

We express our sincere thanks to the Head of the Department, Department of Information Technology, Dr.G.Vadivu, for all the help and infrastructure provided to us to complete this project successfully and his valuable guidance.

We owe our profound gratitude to our project guide **Mrs.M.Rajalakshmi**, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the Teaching staff of the Department of Information Technology which helped us in successfully completing our minor project work. Also, we would like to extend our sincere regards to all the non-teaching staff of the department of Information Technology for their timely support.

Ujjwal Vats,RA1511008010228

TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

ABSTRACT i

TABLE OF FIGURES xviii

**1. INTRODUCTION**

1.1 View 1

1.2 R Programming and R Studio 2

1.2.1 R- Packages 5

1.2.1.1 Installing a New Package 19

1.2.1.2 Load Package from Library 25

1.2.2 Reading a file 30

1.3 What is NDVI 45

1.2 N**DVI, Remote Sensing**  2

1.2.1 NDVICalculation

1.2.2 Understanding & Reading Satellite Imagery

**2. REQUIREMENT ANALYSIS 69**

2.1Software/Hardware Requirements 75

* 1. Tech/non-Tech Requirement 99

2.3 Raster data 100

2.3.1 Raster as surface maps 102

2.3.2 Raster as thematic maps 103

2.3.3 Raster as attribute of a feature 105

2.4 Graph plotting in R 106

**3. DESIGN 69**

3.1 Getting the data 75

3.2 Cleaning the data 99

3.3 Overview of data 100

3.4 Getting the resources 106

3.5 Analyzing the Images 108

3.6 Extracting Info. From data 110

3.7 Data Visualization

**4. Implementation 69**

* 1. Initial Analysis 75
  2. Getting the raster image data 99

4.3 Loading Necessary Packages 101

4.4 Reading Basic Image Properties 102

4.5 Subsetting Image Data

4.6 Cropping and Zooming Image Data 104

4.7 Visualizing Image 105

4.8 Calculating NDVI 106

4.9 Making a web report 107

**5. Testing 69**

5.1 Testing with Image Data

5.2 Visualize single and multi-band imagery

5.3 Subset and rename spectral bands

5.4 Extract raster values

5.5 Image classification

5.6 Basic mathematical operations

5.7 Regression Model

5.8 California precipitation

**6. Conclusion**

**Appendix 1**

**Appendix 2**

**References**

**Abstract**

In our country we get to see every year that many people die due to not having proper amount of food and many Farmers commit Suicide since they failed to cultivate crops and get trapped in heavy debt. But what if the food that will produce in coming year is already known by doing analysis based on previous year data. By this we can make a proper storage of food, if it is known that next year not much food will be produced and can save our country from starvation. However this is already being done manually, record is maintained of food produced and the need of country. But this does work till certain extent only and what we are doing is a computer based analysis by which everyone get to know how much supply of food is there and if we share our analysis everybody will be aware about our country agriculture production. Our project Titled Satellite Imagery Analysis for Predicting Food Supply in our country is all about fetching the data in the form of image from various website which provides satellite image for use. After getting the data cleaning of dirty data is done which is followed by initial analysis and then building a predictive model.The whole concept of the project is based on the Data Science and various types of data analytics technique will be used in our project. Tools which we are using for our project is R Studio for doing R programming, Tableau for getting initial overview of our country food supply data and if possible we would like to present our project in the form of web report. In R studio we will analyze different form of the Image data and see the color correlation within different images. We will use Windows environment to run all required software. We will make use of various R packages to do our project. We will analyze the Image data based on vegetation that is greenery. More the vegetation more will be the food produced and vice versa. The Vegetation could definitely contribute the impact to the annual agricultural production. That is why we have chosen vegetation as a parameter to judge greenness and hence agricultural production.

**Table of Figures**

|  |  |
| --- | --- |
| **Figure** | **Description** |
| **1.1** | **NDVI image calculation** |
| **1.2** | **NDVI example for Agriculture** |
| **2.1** | **RASTER Images** |
| **2.2** | **RASTER as Surafce Maps** |
| **2.3** | **RASTER as Thematic Maps** |
| **2.4** | **RASTER Dataset** |
| **2.5** | **RASTER DATA COORDINATES** |
| **4.1** | **Overview of the current scenario** |
| **4.2** | **Croping of image** |
| **4.3** | **Visualization of Data** |
| **4.4** | **Web Report** |
| **5.1** | **Sentinal image data** |
| **5.2** | **Global Regression Model** |

**1.Introduction**

**1.1 View**

This Project is all about getting the satellite Image data from various source and analyzing those data for creating a prediction model which will predict the future food supply and demand. Greenery in an urban environment is an important consideration when studying temperature, and such enquiry can benefit human health. The focus is on biophysical parameters related to these green areas, such as impervious surface percentages, albedo, areal coverage, elevation, and leaf area index . Geographic information systems and remote sensing were used to quantify green spaces using a pixel-based method. It was found that coverage area has little correlation with temperature. Factor analysis was used to determine the minimum number of independent factors, which explained 63% of the variance of that temperature. Only elevation and albedo were significant biophysical factors.

Getting satellite images for a specific project remains a one of the most challenging steps in the workflow. You have to find the data most suitable for you particular objective. A few important properties to consider while searching the remotely sensed (satellite) data include:

1.Spatial resolution (pixel size)

2.Temporal resolution (return time; availability of historical images, and for a particular moment in time)

3.Radiometric resolution (wavelengths)

4.Quality (e.g. cloud-cover or artifacts in data

There are numerous sources of remotely sensed data from satellites. Generally, the very high spatial resolution data is available as (costly) commericial products. Lower spatial resolution data is freely available from NASA and ESA. In this this tutorial we’ll use freely available Sentinel, Landsat 8 andMODIS data.

Read image and display basic image properties.

All these products consist of sensors that measure reflectance of the sun’s radiation in different wavelengths (e.g. in the red, green, or blue wavelengths). Thus one image consists of multiple spectral layers. In remote sensing jargon, such layers are referred to as “bands” (shorthand for “bandwidth”” in the electromagnetic spectrum). From such multi-spectral data you can create a RasterBrick or RasterStack in R (as long as each layer has the same extent and resolution).Another piece of jargon is “pixel”. This is the term that is used for grid cell in remote sensing literature.

**1.2 R Programming and R Studio**

Once R is installed on your computer, the software is executed by launching the corresponding executable. The prompt, by default ‘>’, indicates that R is waiting for your commands. Under Windows using the program Rgui.exe, some commands (accessing the on-line help, opening files.) can be executed via the pull-down menus. At this stage, a new user is likely to wonder “What do I do now?” It is indeed very useful to have a few ideas on how R works when it is used for the first time, and this is what we will see now. We shall see first briefly how R works. Then, I will describe the “assign” operator which allows creating objects, how to manage objects in memory, and finally how to use the on-line help which is very useful when running R.

The fact that R is a language may deter some users who think “I can’t program”. This should not be the case for two reasons. First, R is an interpreted language, not a compiled one, meaning that all commands typed on the keyboard are directly executed without requiring to build a complete program like in most computer languages (C, Fortran, Pascal, . . .). Second, R’s syntax is very simple and intuitive. For instance, a linear regression can be done with the command lm(y ~ x) which means “fitting a linear model with y as response and x as predictor”. In R, in order to be executed, a function always needs to be written with parentheses, even if there is nothing within them (e.g., ls()). If one just types the name of a function without parentheses, R will display the content of the function. In this document, the names of the functions are generally written with parentheses in order to distinguish them from other objects, unless the text indicates clearly so.

**1.2.1 R- Packages**

R packages are a collection of R functions, complied code and sample data. They are stored under a directory called "library" in the R environment. By default, R installs a set of packages during installation. More packages are added later, when they are needed for some specific purpose. When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.

All the packages available in R language are listed at [R Packages.](https://cran.r-project.org/web/packages/available_packages_by_name.html)

**1.2.1.1 Installing a New Package**

There are two ways to add new R packages. One is installing directly from the CRAN directory and another is downloading the package to your local system and installing it manually.

The following command gets the packages directly from CRAN webpage and installs the package in the R environment. You may be prompted to choose a nearest mirror. Choose the one appropriate to your location.

install.packages("Package Name")

# Install the package named "XML".

install.packages("XML")

We can run the following command to install this package in the R environment.

install.packages(file\_name\_with\_path, repos = NULL, type = "source")

# Install the package named "XML"

install.packages("E:/XML\_3.98-1.3.zip", repos = NULL, type = "source")

## 1.2.1.2 Load Package from Library

Before a package can be used in the code, it must be loaded to the current R environment. You also need to load a package that is already installed previously but not available in the current environment.

## 1.2.2 Reading a file

Reading data in a file For reading and writing in files, R uses the working directory. To find this directory, the command getwd() (get working directory) can be used, and the working directory can be changed with setwd("C:/data") or setwd("/home/- paradis/R"). It is necessary to give the path to a file if it is not in the working directory. 8 R can read data stored in text (ASCII) files with the following functions:

read.table (which has several variants, see below), scan and read.fwf.

R can also read files in other formats (Excel, SAS, SPSS, . . .), and access SQLtype databases, but the functions needed for this are not in the package base. These functionalities are very useful for a more advanced use of R, but we will restrict here to reading files in ASCII format. The function read.table has for effect to create a data frame, and so is the main way to read data in tabular form. For instance, if one has a file named data.dat, the command: > mydata <- read.table("data.dat") will create a data frame named mydata, and each variable will be named, by default, V1, V2, . . . and can be accessed individually by mydata$V1, mydata$V2, . . . , or by mydata["V1"], mydata["V2"], . . . , or, still another solution, by mydata[, 1], mydata[,2 ], . . . 9 There are several options whose default values (i.e. those used by R if they are omitted by the user) are detailed in the following table: read.table(file, header = FALSE, sep = "", quote = "\"’", dec = ".",

**1.3 What is NDVI (Normalized Difference Vegetation Index)?**

Normalized Difference Vegetation Index (NDVI) quantifies vegetation by measuring the difference between near-infrared (which vegetation strongly reflects) and red light (which vegetation absorbs).NDVI always ranges from -1 to +1. But there isn’t a distinct boundary for [each type of land cover](https://gisgeography.com/free-global-land-cover-land-use-data/).

For example, when you have negative values, it’s highly likely that it’s water. On the other hand, if you have a NDVI value close to +1, there’s a high possibility that it’s dense green leaves.But when NDVI is close to zero, there isn’t green leaves and it could even be an urbanized area.

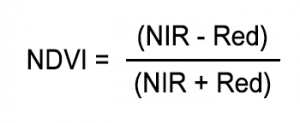
**1.3.1 NDVI, the Foundation for Remote Sensing Phenology**  
Remote sensing phenology studies use data gathered by satellite sensors that measure wavelengths of light absorbed and reflected by green plants. Certain pigments in plant leaves strongly absorb wavelengths of visible (red) light. The leaves themselves strongly reflect wavelengths of near-infrared light, which is invisible to human eyes. As a plant canopy changes from early spring growth to late-season maturity and senescence, these reflectance properties also change.

Many sensors carried aboard satellites measure red and near-infrared light waves reflected by land surfaces. Using mathematical formulas (algorithms), scientists transform raw satellite data about these light waves into vegetation indices. A vegetation index is an indicator that describes the greenness — the relative density and health of vegetation — for each picture element, or pixel, in a satellite image. 

Although there are several vegetation indices, one of the most widely used is the Normalized Difference Vegetation Index (NDVI). NDVI values range from +1.0 to -1.0. Areas of barren rock, sand, or snow usually show very low NDVI values (for example, 0.1 or less). Sparse vegetation such as shrubs and grasslands or senescing crops may result in moderate NDVI values (approximately 0.2 to 0.5). High NDVI values (approximately 0.6 to 0.9) correspond to dense vegetation such as that found in temperate and tropical forests or crops at their peak growth stage.   
By transforming raw satellite data into NDVI values, researchers can create images and other products that give a rough measure of vegetation type, amount, and condition on land surfaces around the world. NDVI is especially useful for continental- to global-scale vegetation monitoring because it can compensate for changing illumination conditions, surface slope, and viewing angle. That said, NDVI does tend to saturate over dense vegetation and is sensitive to underlying soilcolor.   
  
NDVI values can be averaged over time to establish "normal" growing conditions in a region for a given time of year. Further analysis can then characterize the health of vegetation in that place relative to the norm. When analyzed through time, NDVI can reveal where vegetation is thriving and where it is under stress, as well as changes in vegetation due to human activities such as deforestation, natural disturbances such as wild fires, or changes in plants' phenological stage.

**1.3.2 NDVI Calculation**

As shown below, Normalized Difference Vegetation Index (NDVI) uses the NIR and red channels in its formula.



Healthy vegetation (chlorophyll) reflects more near-infrared (NIR) and green light compared to other wavelengths. But it absorbs more red and blue light.This is why our eyes see vegetation as the **color green**. If you could see near-infrared, then it would be strong for vegetation too. Satellite sensors like [**Landsat**](https://gisgeography.com/landsat-program-satellite-imagery-bands/) and [**Sentinel-2**](https://gisgeography.com/how-to-download-sentinel-satellite-data/) both have the necessary bands with NIR and red.

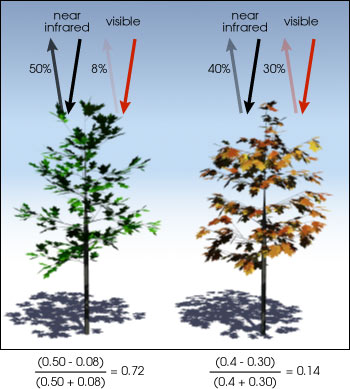


Figure 1.1

The result of this formula generates a value between -1 and +1. If you have low reflectance (or low values) in the red channel and high reflectance in the NIR channel, this will yield a high NDVI value. And vice versa.

Overall, NDVI is a standardized way to measure healthy vegetation. When you have high NDVI values, you have healthier vegetation. When you have low NDVI, you have less or no vegetation. Generally, if you want to see vegetation change over time, then you will have to perform [atmospheric correction](https://gisgeography.com/atmospheric-correction/).

### 1.3.3 NDVI Example for Agriculture

Let’s examine NDVI for an agricultural area with center pivot irrigation. Pivot irrigation rotates on a point creating a circular crop pattern.

In **true color**, here’s how it looks for red, green and blue bands. We say true color because it is the same as how our eyes see.



Figure 1.2

In the formula, you can see how NDVI leverages near-infrared (NIR). So when we put NIR band to display as red, we get **color infrared**. We say color infrared because near infrared is in the red channel. As you can see below, the pivot irrigation vegetation should already be shouting out at you in bright red!

When you apply the formula, bright green indicates high NDVI. Whereas red has low NDVI. So it’s quantifying vegetation by measuring the difference between near-infrared (which vegetation strongly reflects) and red light (which vegetation absorbs).

### 1.3.4 Use of NDVI

### In particular, there are several sectors that use NDVI.For example, in agriculture, farmers use NDVI for precision farming and to measure biomass. Whereas, in forestry, foresters use NDVI to quantify forest supply and leaf area index.

Furthermore, NASA states that [**NDVI is a good indicator of drought**](http://earthobservatory.nasa.gov/Features/MeasuringVegetation/). When water limits vegetation growth, it has a lower relative NDVI and density of vegetation.In reality, there are hundreds of applications where NDVI and other [**remote sensing applications**](https://gisgeography.com/100-earth-remote-sensing-applications-uses/)is being applied to in the real world.As mentioned before, satellites like [**Sentinel-2**](https://gisgeography.com/how-to-download-sentinel-satellite-data/), [**Landsat**](https://gisgeography.com/landsat-program-satellite-imagery-bands/) and [**SPOT**](https://gisgeography.com/spot-satellite-pour-observation-terre/) produce red and near infrared images.

### 1.4 Understanding & Reading Satellite Imagery

Any satellite essentialy captures radiance/reflectance in different wavelengths of the spectra ranging from visible to microwaves. What we obtain as data in the image much like a photograph is reflectance of sunlight/optical rays in these different spectral bands reflected by land features like buildings, water body, vegetation , barren surface etc. Except that in photographs, reflectance is captured only in the visible (RGB) spectra.

Landsat 8 imagery consists of nine spectral bands with a spatial resolution of 30 meters for Bands 1 to 7 and 9. New band 1 (ultra-blue) is useful for coastal and aerosol studies. Band 9 is useful for cirrus cloud detection. The resolution for Band 8 (panchromatic) is 15 meters. Thermal bands 10 and 11 are useful in providing more accurate surface temperatures and are collected at 100 meters

When we worked with flat files or non-spatial data, imagine an image as a 2D matrix with values representing radiance and each cell representing an area on the earth having dimensions equivalent to the resolution of the satellite in this case 30m X 30m. An Image is a continuous grid of equal dimensions representing some area on the earth’s surface and its feature.

**1.5 Frequency of Coverage**

Neither the NDVI nor the EVI product will eliminate all obstacles. Clouds and aerosols can often block the satellites’ view of the surface entirely, glare from the sun can saturate certain pixels, and temporary malfunctions in the satellite instruments themselves can distort an image. Consequently, many of the pixels in a day’s worth of images are indecipherable, and maps made from the daily Vegetation Indices are patchy at best.

**2. Requirement Analysis**

**2.1 Software/Hardware Requirements**

* RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. Click here to see more RStudio features. RStudio is available in open source
* Tableau is groundbreaking data visualization software created by Tableau Software. Tableau allows for instantaneous insight by transforming data into visually appealing, interactive visualizations called dashboards.
* Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.
* Windows 10 is a personal computer operating system developed and released by Microsoft, as part of the Windows NT family of operating systems.

**2.2 Technical/Non-Technical Requirements**

* Knowledge of R programming-Familiar with the packages used for spatial Data Analysis.
* Knowledge of Markup Language (HTML) and Styling sheet language (CSS) which describes the appearance of the Markup language.
* Raw Data which is the High Resolution Vegetation image of our country.
* This projects actually requires lots of research work since there are very less online resource available related to the project.

**2.3 Raster Data**

In its simplest form, a raster consists of a matrix of cells (or pixels) organized into rows and columns (or a grid) where each cell contains a value representing information, such as temperature. Rasters are digital aerial photographs, imagery from satellites, digital pictures, or even scanned maps.

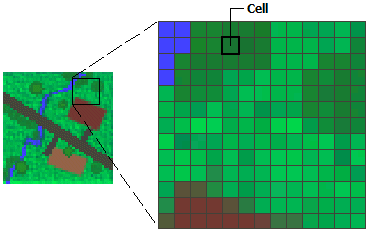


Figure 2.1

Data stored in a raster format represents real-world phenomena:

Thematic data (also known as discrete) represents features such as land-use or soils data.

Continuous data represents phenomena such as temperature, elevation, or spectral data such as satellite images and aerial photographs.Pictures include scanned maps or drawings and building photographs.Thematic and continuous rasters may be displayed as data layers along with other geographic data on your map but are often used as the source data for spatial analysis with the ArcGIS Spatial Analyst extension. Picture rasters are often used as attributes in tables—they can be displayed with your geographic data and are used to convey additional information about map features.

While the structure of raster data is simple, it is exceptionally useful for a wide range of applications. Within a GIS, the uses of raster data fall under four main categories:

Rasters as basemaps

A common use of raster data in a GIS is as a background display for other feature layers. For example, orthophotographs displayed underneath other layers provide the map user with confidence that map layers are spatially aligned and represent real objects, as well as additional information. Three main sources of raster basemaps are orthophotos from aerial photography, satellite imagery, and scanned maps. Below is a raster used as a basemap for road data.

**2.3.1 Rasters as surface maps**

Rasters are well suited for representing data that changes continuously across a landscape (surface). They provide an effective method of storing the continuity as a surface. They also provide a regularly spaced representation of surfaces. Elevation values measured from the earth's surface are the most common application of surface maps, but other values, such as rainfall, temperature, concentration, and population density, can also define surfaces that can be spatially analyzed. The raster below displays elevation—using green to show lower elevation and red, pink, and white cells to show higher elevations.

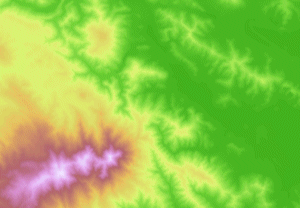


Figure 2.2

**2.3.2 Rasters as thematic maps**

Rasters representing thematic data can be derived from analyzing other data. A common analysis application is classifying a satellite image by land-cover categories. Basically, this activity groups the values of multispectral data into classes (such as vegetation type) and assigns a categorical value. Thematic maps can also result from geoprocessing operations that combine data from various sources, such as vector, raster, and terrain data. For example, you can process data through a geoprocessing model to create a raster dataset that maps suitability for a specific activity. Below is an example of a classified raster dataset showing land use.

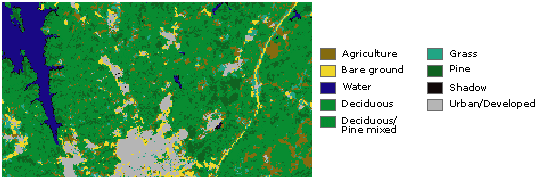


Figure 2.3

**2.3.3 Rasters as attributes of a feature**

Rasters used as attributes of a feature may be digital photographs, scanned documents, or scanned drawings related to a geographic object or location. A parcel layer may have scanned legal documents identifying the latest transaction for that parcel, or a layer representing cave openings may have pictures of the actual cave openings associated with the point features. Below is a digital picture of a large, old tree that could be used as an attribute to a landscape layer that a city may maintain.

**2.3.4 Why store data as a raster?**

Sometimes you don't have the choice of storing your data as a raster; for example, imagery is only available as a raster. However, there are many other features (such as points) and measurements (such as rainfall) that could be stored as either a raster or a feature (vector) data type.

The advantages of storing your data as a raster are as follows:

A simple data structure—A matrix of cells with values representing a coordinate and sometimes linked to an attribute table

* A powerful format for advanced spatial and statistical analysis.
* The ability to represent continuous surfaces and perform surface analysis.
* The ability to uniformly store points, lines, polygons, and surfaces.
* The ability to perform fast overlays with complex datasets.
* There are other considerations for storing your data as a raster that may convince you to use a vector-based storage option.
* There can be spatial inaccuracies due to the limits imposed by the raster dataset cell dimensions.

Raster datasets are potentially very large. Resolution increases as the size of the cell decreases; however, normally cost also increases in both disk space and processing speeds. For a given area, changing cells to one-half the current size requires as much as four times the storage space, depending on the type of data and storage techniques used.

**2.3.5 General characteristics of raster data**

In raster datasets, each cell (which is also known as a pixel) has a value. The cell values represent the phenomenon portrayed by the raster dataset such as a category, magnitude, height, or spectral value. The category could be a land-use class such as grassland, forest, or road. A magnitude might represent gravity, noise pollution, or percent rainfall. Height (distance) could represent surface elevation above mean sea level, which can be used to derive slope, aspect, and watershed properties. Spectral values are used in satellite imagery and aerial photography to represent light reflectance and color.

Cell values can be either positive or negative, integer, or floating point. Integer values are best used to represent categorical (discrete) data and floating-point values to represent continuous surfaces. For additional information on discrete and continuous data, see [Discrete and continuous data](http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/discrete-and-continuous-data.htm). Cells can also have a NoData value to represent the absence of data. For information on NoData, see [NoData in raster datasets](http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/nodata-in-raster-datasets.htm).

The area (or surface) represented by each cell consists of the same width and height and is an equal portion of the entire surface represented by the raster. For example, a raster representing elevation (that is, digital elevation model) may cover an area of 100 square kilometers. If there were 100 cells in this raster, each cell would represent 1 square kilometer of equal width and height (that is, 1 km x 1 km).

The dimension of the cells can be as large or as small as needed to represent the surface conveyed by the raster dataset and the features within the surface, such as a square kilometer, square foot, or even square centimeter. The cell size determines how coarse or fine the patterns or features in the raster will appear. The smaller the cell size, the smoother or more detailed the raster will be. However, the greater the number of cells, the longer it will take to process, and it will increase the demand for storage space. If a cell size is too large, information may be lost or subtle patterns may be obscured. For example, if the cell size is larger than the width of a road, the road may not exist within the raster dataset. In the diagram below, you can see how this simple polygon feature will be represented by a raster dataset at various cell sizes.

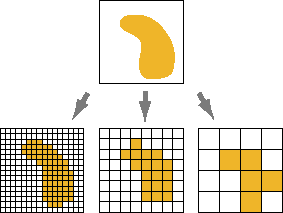


Figure 2.4

The location of each cell is defined by the row or column where it is located within the raster matrix. Essentially, the matrix is represented by a Cartesian coordinate system, in which the rows of the matrix are parallel to the x-axis and the columns to the y-axis of the Cartesian plane. Row and column values begin with 0. In the example below, if the raster is in a Universal Transverse Mercator (UTM) projected coordinate system and has a cell size of 100, the cell location at 5,1 would be 300,500 East, 5,900,600 North.

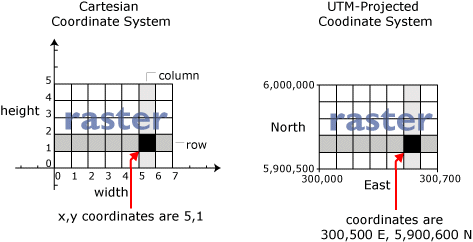


Figure 2.5

Often you need to specify the extent of a raster. The extent is defined by the top, bottom, left, and right coordinates of the rectangular area covered by a raster, as shown below.

**2.4 Graph plotting in R**

ggplot2 is a plotting package that makes it simple to create complex plots from data in a data frame. It provides a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties. Therefore, we only need minimal changes if the underlying data change or if we decide to change from a bar plot to a scatterplot. This helps in creating publication quality plots with minimal amounts of adjustments and tweaking.

ggplot2 functions like data in the ‘long’ format, i.e., a column for every dimension, and a row for every observation. Well-structured data will save you lots of time when making figures with ggplot2

ggplot graphics are built step by step by adding new elements. Adding layers in this fashion allows for extensive flexibility and customization of plots.

* use the ggplot() function and bind the plot to a specific data frame using the data argument
* define a mapping (using the aesthetic (aes) function), by selecting the variables to be plotted and specifying how to present them in the graph, e.g. as x/y positions or characteristics such as size, shape, color, etc.
* add ‘geoms’ – graphical representations of the data in the plot (points, lines, bars). **ggplot2** offers many different geoms; we will use some common ones today, including:
  + geom\_point() for scatter plots, dot plots, etc.
  + geom\_boxplot() for, well, boxplots!
  + geom\_line() for trend lines, time series, etc.

## 2.4.1 Grammar Of Graphics

The basic idea: independently specify plot building blocks and combine them to create just about any kind of graphical display you want. Building blocks of a graph include:

* data
* aesthetic mapping
* geometric object
* statistical transformations
* scales
* coordinate system
* position adjustments
* faceting

**3.Design**

**3.1 Getting Data**

The initial design of the project starts with collecting no noise NDVI images which can be provided by USGS Earth Explorer. The USGS Earth Explorer is a similar tool to the USGS Global Visualization Viewer (GloVis) in that users search catalogs of satellite and aerial imagery. The USGS Earth Explorer is the new and improved version. The USGS Earth Explorer gives some extra capabilities:

* Downloading data over chronological timelines.
* Wide range of specifying search criteria.
* A long list of satellite and aerial imagery to choose.

**3.2 Cleaning the data**

Cleaning of Data is important :

* You are joining two data sets that don't use the same naming convention (country code vs. country name). How do you effectively join these?
* Some data is missing. For example, incomplete survey responses, measurement issues, lack of complete data from an external source, etc. How do you handle these?
* You have a handful of outliers. Do you modify or remove the data or simply use a model less sensitive to outliers?
* Your data is in multiple units. Will this skew the results of your model? If so, should you standardize or normalize?

Once you "clean" your data, you can effectively start building more interesting features, and eventually build a solid model.

**3.3 Overview of Data**

In this stage of system design the overview of data is done. Data Analysis process involves following components:

* Extract data
* Hypothesis building (Data visualization & Insights building)
* Transform data (Aggregate/Join Data)
* Statistical Data Analysis (Bivariate,univariate analysis of variables)
* Model Development (Create statistical models) / Hypothesis testing
* Recommendations / Story Boarding

**3.4 Getting Resources**

In this step we collect resources for performing different tasks. In this project the different resources can be R studio and other data analysis tool.

**3.5 Analysing Images**

Image analysis is the extraction of meaningful information from images mainly from digital images by means of digital image processing techniques.Image analysis tasks can be as simple as reading bar coded tags or as sophisticated as identifying a person from their face.

Computers are indispensable for the analysis of large amounts of data, for tasks that require complex computation, or for the extraction of quantitative information. On the other hand, the human visual cortex is an excellent image analysis apparatus, especially for extracting higher-level information, and for many applications — including medicine, security, and remote sensing — human analysts still cannot be replaced by computers. For this reason, many important image analysis tools such as edge detectors and neural networks are inspired by human visual perception models.

**3.6 Extracting Information from Data**

It is difficult to extracting data from the unstructured source. But, it depends on the type of data. If it comprises lots of variations like sometimes query based data, accounting base data etc then it's hard to rectify.

* If data is unstructured with 3–4 groups then it's fine to structured out it. As we just need to make a categorized list including item categories.
* There are lots of software and tools which are introduced into structured data in chronological order. You need to just admit data in such software and it result out it category wise to view data.
* For a structured process, statistical modeling and execution, relevant topics should be analysis first to track the data in good logic.
* New information is crucial for an organization, so it is important to capture it first on basis of relevant facts and information or ideas.

**3.7 Data Visualization**

Data visualization enables decision-makers to see analytics presented visually so that they can grasp difficult concepts or identify new patterns. Data visualization software plays an important role in big data and advanced analytics projects. Many business organizations implement data visualization software to track their own initiatives. For example, a marketing team might implement a data visualization software to monitor the performance of an email campaign and track metrics like open rate, click-through rate, and conversion rate

**3.8 Result**

At last the last and the most important is getting desired result on the basis of analyzed data .

**4.Implementation**

**4.1 Initial Analysis**

The Project begin with the initial Analysis of CSV data about food demand and supply from the year 2005 to 2015, provided by the Indian Government. Such Initial Analysis is very much important to get the overview of the current scenario, and also it provides a clear picture of what to do next.

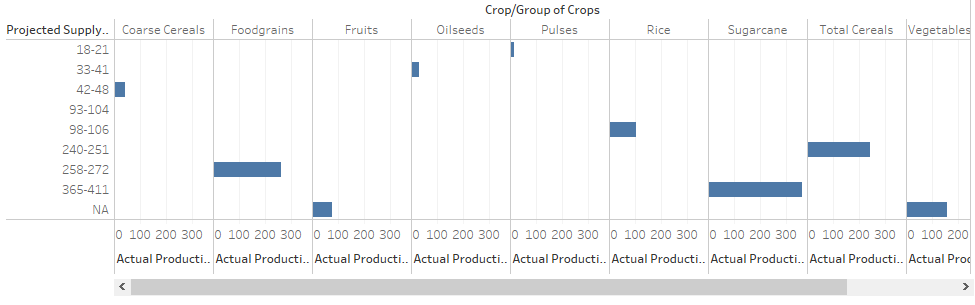


Figure 4.1

The above figure shows the result of initial analysis. It is a graph of Actual Production vs Projected Supply with respect to variety of major crops in India. After the Initial analysis of the data what we got, we found that for maximum year, the Demand is always greater than supply.

**4.2 Getting the Satellite Image data**

Getting satellite images for a specific project remains a one of the most challenging steps in the workflow. You have to find the data most suitable for you particular objective. A few important properties to consider while searching the remotely sensed (satellite) data include:

1. Spatial resolution (pixel size)
2. Temporal resolution (return time; availability of historical images, and for a particular moment in time)
3. Radiometric resolution (wavelengths)
4. Quality (e.g. cloud-cover in data)

There are numerous sources of remotely sensed data from satellites. Generally, the very high spatial resolution data is available as (costly) commercial products. Lower spatial resolution data is freely available from NASA and ESA. In this this tutorial we’ll use freely available Sentinel and MODIS data. Some of the Image data source for this project is listed below:

1. <http://earthexplorer.usgs.gov/>
2. <https://lpdaacsvc.cr.usgs.gov/appeears/>
3. <https://search.earthdata.nasa.gov/search>
4. <https://lpdaac.usgs.gov/data_access/data_pool>
5. <https://scihub.copernicus.eu/>
6. <https://aws.amazon.com/public-data-sets/landsat/>

**4.2.1 Cleaning the Image Data**

Cleaning of Image Data or any Data must be cleaned before feeding into the system for processing. This becomes really important because even few invalid or data error can badly impact your overall result of the operation. In case of Image data, it can be cleaned manually by completely removing the invalid image or by processing those Image using any software, Earlier one is done in this project.

**4.3 Loading Necessary Packages**

R provides a variety of packages for performing different kind of operation which really makes our work easy. Syntax for loading any package in R is “library(Package Name)”. In this project we require packages for raster data, spatial analysis, data manipulation, etc. The way required packages are loaded is given below.

> library(raster)

> library(sp)

> library(dplyr)

> library(tidyr)

> library(rpart)

**4.3.1 Raster**

The raster package provides classes and functions to manipulate geographic (spatial) data in 'raster' format. Raster data divides space into cells (rectangles; pixels) of equal size (in units of the coordinate reference system). Such continuous spatial data are also referred to as 'grid' data, and be contrasted with discrete (object based) spatial data (points, lines, polygons).

**4.3.2 Spatial**

Classes and methods for spatial data; the classes document where the spatial location information resides, for 2D or 3D data. Utility functions are provided, e.g. for plotting data as maps, spatial selection, as well as methods for retrieving coordinates, for subsetting, print, summary, etc.

**4.3.3 Dplyr**

A fast, consistent tool for working with data frame like objects, both in memory and out of memory. dplyr is designed to abstract over how the data is stored. That means as well as working with local data frames, you can also work with remote database tables, using exactly the same R code.

**4.3.4 Rpart**

The R package [rpart](https://cran.r-project.org/web/packages/rpart/index.html) implements recursive partitioning. It is very easy to use. Classification and regression trees (as described by Brieman, Freidman, Olshen, and Stone) can be generated through the rpart package.

**4.4 Reading Basic Image Properties**

All the Satellites consist of sensors that measure reflectance of the sun’s radiation in different wavelengths (e.g. in the red, green, or blue wavelengths)and captures the images. Thus one image consists of multiple spectral layers. In remote sensing jargon, such layers are referred to as “bands”. From such multi-spectral data we have create a RasterBrick in R.

Img\_Data<- brick('Raster\_Img\_Ind.tif')

Img\_Data

> Img\_Data<- brick('Raster\_Img\_Ind.tif')

>

> Img\_Data

class : RasterBrick

dimensions : 1934, 1786, 3454124, 10 (nrow, ncol, ncell, nlayers)

resolution : 10, 10 (x, y)

extent : 251900.9, 269760.9, -382182.3, -362842.3 (xmin, xmax, ymin, ymax)

coord. ref. : +proj=utm +zone=37 +datum=WGS84 +units=m +no\_defs +ellps=WGS84 +towgs84=0,0,0

data source : C:\Users\Ujjwal\Documents\Raster\_Img\_Ind.tif

names : Raster\_Img\_Ind.1, Raster\_Img\_Ind.2, Raster\_Img\_Ind.3, Raster\_Img\_Ind.4, Raster\_Img\_Ind.5, Raster\_Img\_Ind.6, Raster\_Img\_Ind.7, Raster\_Img\_Ind.8, Raster\_Img\_Ind.9, Raster\_Img\_Ind.10

min values : 576, 401, 235, 307, 464, 489, 471, 371, 149, 89

max values : 6398, 6279, 6372, 5482, 6586, 7242, 11990, 7095, 7738, 5612

**4.4.1 Detailed Information and Statistics**

crs(Img\_Data) #Coordinate reference system

nlayers(Img\_Data) #no. of rows,columns

res(Img\_Data) #Spatial resolution of Image data

names(Img\_Data) #Names associated with different bands

> crs(Img\_Data)

CRS arguments:

+proj=utm +zone=37 +datum=WGS84 +units=m +no\_defs +ellps=WGS84 +towgs84=0,0,0

> nlayers(Img\_Data)

[1] 10

> res(Img\_Data)

[1] 10 10

> names(Img\_Data)

[1] "Raster\_Img\_Ind.1" "Raster\_Img\_Ind.2" "Raster\_Img\_Ind.3" "Raster\_Img\_Ind.4"

[5] "Raster\_Img\_Ind.5" "Raster\_Img\_Ind.6" "Raster\_Img\_Ind.7" "Raster\_Img\_Ind.8"

[9] "Raster\_Img\_Ind.9" "Raster\_Img\_Ind.10"

**4.5 Subsetting Image Data**

R has powerful indexing features for accessing object elements. These features can be used to select and exclude variables and observations. Here, subsetting is used to exclude the unwanted data and focus on the selected data only.

subset\_data<-subset(Img\_Data, 1:8)

names(subset\_data)

names(subset\_data)<-c(1:8)

names(subset\_data)

> subset\_data<-subset(Img\_Data, 1:8)

> names(subset\_data)

[1] "Raster\_Img\_Ind.1" "Raster\_Img\_Ind.2" "Raster\_Img\_Ind.3" "Raster\_Img\_Ind.4"

[5] "Raster\_Img\_Ind.5" "Raster\_Img\_Ind.6" "Raster\_Img\_Ind.7" "Raster\_Img\_Ind.8"

>

> names(subset\_data)<-c(1:8)

> names(subset\_data)

[1] "X1" "X2" "X3" "X4" "X5" "X6" "X7" "X8"

> subset\_data<-subset(Img\_Data, 1:8)

> names(subset\_data)

[1] "Raster\_Img\_Ind.1" "Raster\_Img\_Ind.2" "Raster\_Img\_Ind.3" "Raster\_Img\_Ind.4"

[5] "Raster\_Img\_Ind.5" "Raster\_Img\_Ind.6" "Raster\_Img\_Ind.7" "Raster\_Img\_Ind.8"

>

> names(subset\_data)<-c(1:8)

> names(subset\_data)

[1] "X1" "X2" "X3" "X4" "X5" "X6" "X7" "X8"

**4.6 Cropping and Zooming Image Data**

crop returns a geographic subset of an object as specified by an Extent object (or object from which an extent object can be extracted/created). If x is a Raster object, the Extent is aligned to x. Areas included in y but outside the extent of x are ignored.

new\_extent<-extent(235000, 260000, -375000, -370000)

croped\_Img<-crop(subset\_data,new\_extent)

> (new\_extent<-extent(235000, 260000, -375000, -370000))

class : Extent

xmin : 235000

xmax : 260000

ymin : -375000

ymax : -370000

>

> (croped\_Img<-crop(subset\_data,new\_extent))

class : RasterBrick

dimensions : 500, 810, 405000, 8 (nrow, ncol, ncell, nlayers)

resolution : 10, 10 (x, y)

extent : 251900.9, 260000.9, -375002.3, -370002.3 (xmin, xmax, ymin, ymax)

coord. ref. : +proj=utm +zone=37 +datum=WGS84 +units=m +no\_defs +ellps=WGS84 +towgs84=0,0,0

data source : in memory

names : X1, X2, X3, X4, X5, X6, X7, X8

min values : 727, 542, 351, 435, 550, 563, 470, 430

max values : 3504, 3718, 4332, 3834, 3928, 4944, 4792, 5255

**4.7 Visualizing Image**

plotRGB(croped\_Img, r= 2, g = 8, b = 2,stretch ="lin",main="Vegetation")

|  |
| --- |
| > plotRGB(croped\_Img, r= 2, g = 8, b = 2,stretch ="lin",main="Vegetation") |
|  |
| |  | | --- | |  | |

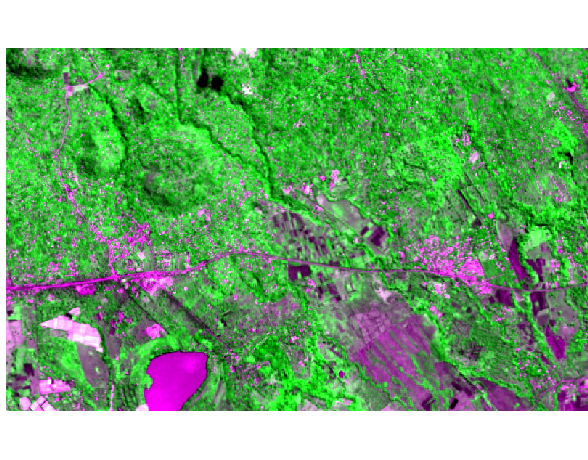
****

Figure 4.2

**4.8 Calculating NDVI**

The Normalized Difference Vegetation Index (NDVI) is a quantitative index of greenness ranging from 0-1 where 0 represents minimal or no greenness and 1 represents maximum greenness. NDVI is often used for a quantitate proxy measure of vegetation health, cover and phenology (life cycle stage) over large areas.

NDVI is calculated from the visible and near-infrared light reflected by vegetation. Healthy vegetation absorbs most of the visible light that hits it, and reflects a large portion of near-infrared light. Unhealthy or sparse vegetation reflects more visible light and less near-infrared light.

 Either scenario results in an NDVI value that, over time, can be averaged to establish the "normal" growing conditions for the vegetation in a given region for a given time of the year. In short, a region’s absorption and reflection of photosynthetically active radiation over a given period of time can be used to characterize the health of the vegetation there, relative to the norm. Sometimes we can download already calculated NDVI data products. In this case, we need to calculate NDVI using NDVI formula which is explained below.

The normalized difference vegetation index (NDVI) uses a ratio between near infrared and red light within the electromagnetic spectrum. To calculate NDVI you use the following formula where NIR is near infrared light and red represents red light. For your raster data, you will take the reflectance value in the red and near infrared bands to calculate the index.

NDVI = { ( NIR – RED ) + ( NIR + RED ) }

Incase of our Image Data, NIR value is 7 and RED band value is 3.

ndvi<-(Img\_Data[[7]] -Img\_Data[[3]]) / (Img\_Data[[7]] + Img\_Data[[3]])

ndvi

> ndvi<-(Img\_Data[[7]] -Img\_Data[[3]]) / (Img\_Data[[7]] + Img\_Data[[3]])

> ndvi

class : RasterLayer

dimensions : 1934, 1786, 3454124 (nrow, ncol, ncell)

resolution : 10, 10 (x, y)

extent : 251900.9, 269760.9, -382182.3, -362842.3 (xmin, xmax, ymin, ymax)

coord. ref. : +proj=utm +zone=37 +datum=WGS84 +units=m +no\_defs +ellps=WGS84 +towgs84=0,0,0

data source : in memory

names : layer

values : -0.1530886, 0.8304381 (min, max)

**4.8.1 Visualizing with NDVI range**

plot(ndvi, col = rev(terrain.colors(6)), main = 'NDVI')

|  |
| --- |
| > plot(ndvi, col = rev(terrain.colors(6)), main = 'NDVI') |
|  |
| |  | | --- | |  | |

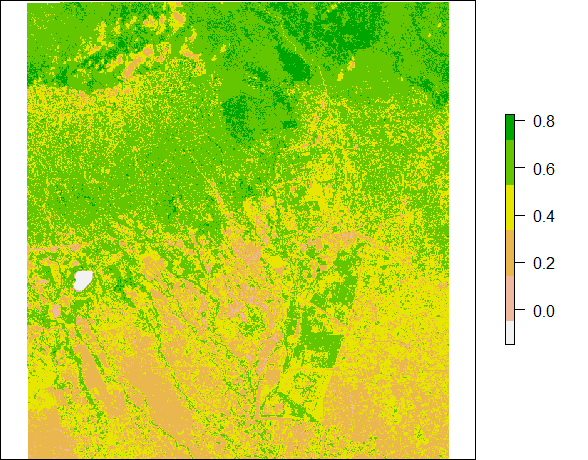
****

Figure 4.3

**4.9 Prediction**

Make a Raster object with predictions from a fitted model object (for example, obtained with lm, glm). The first argument is a Raster object with the independent (predictor) variables. The [names](https://www.rdocumentation.org/link/names?package=raster&version=2.6-7) in the Raster object should exactly match those expected by the model. This will be the case if the same Raster object was used (via extract) to obtain the values to fit the model. Any type of model (e.g. glm, gam, randomForest) for which a predict method has been implemented (or can be implemented) can be used.

sr <- sampleRandom(croped\_Img, 10000)

plot(sr[,c(3,7)])

pca <- prcomp(sr, scale = TRUE)

pci <- predict(croped\_Img, pca, index = 1:2)

pci

> pci

class : RasterBrick

dimensions : 500, 810, 405000, 2 (nrow, ncol, ncell, nlayers)

resolution : 10, 10 (x, y)

extent : 251900.9, 260000.9, -375002.3, -370002.3 (xmin, xmax, ymin, ymax)

coord. ref. : +proj=utm +zone=37 +datum=WGS84 +units=m +no\_defs +ellps=WGS84 +towgs84=0,0,0

data source : in memory

names : layer.1, layer.2

min values : -8.35063, -37.44528

max values : 19.515004, 7.323271

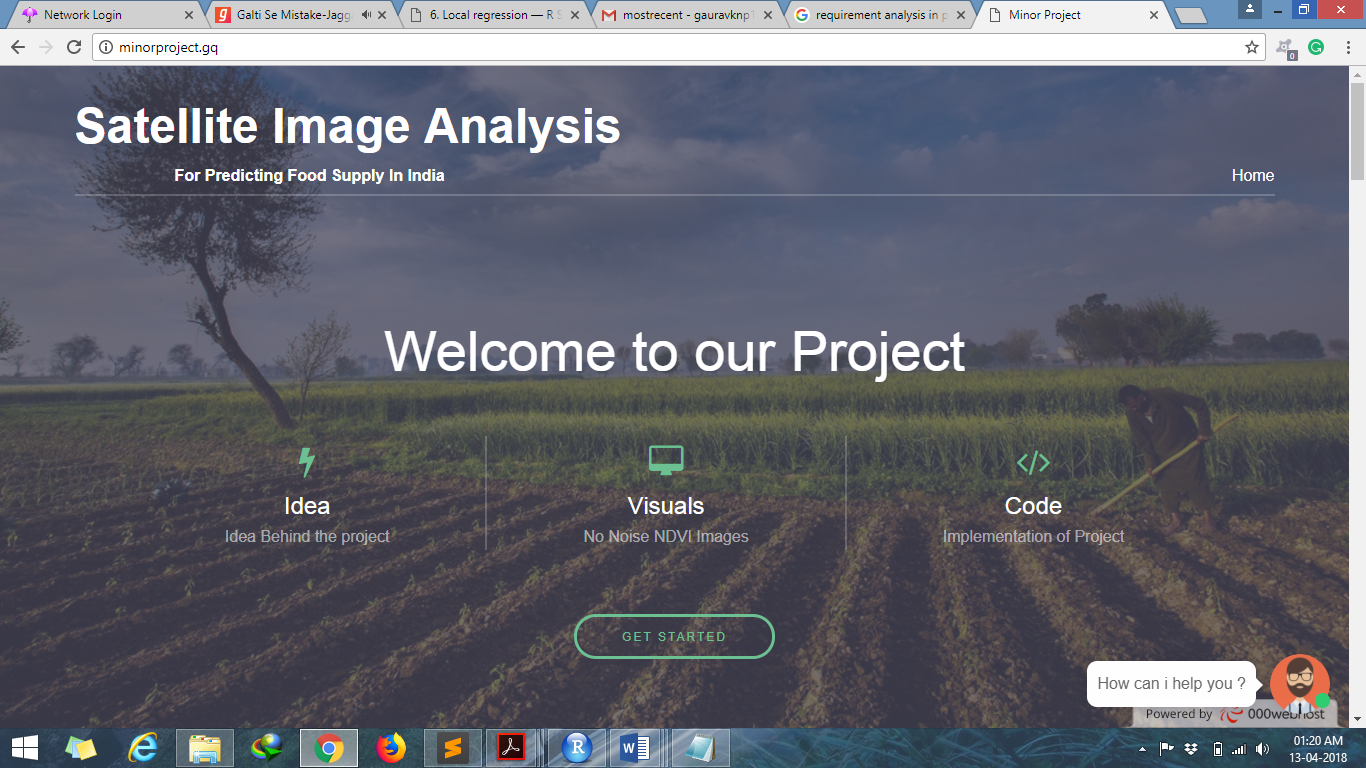
**4.10 Making Web Report**

Figure 4.4

In order to make our presentation look creative and catchy, we thought of making a web report presentation. This will bring a different type in showcasing the report other than ordinary power-point presentations. Also it could be accessed anytime, anywhere and by anyone thus providing mobility and comfort.

The report is divided into 3 categories-

* Idea- It contains the basic idea of our approach with the tools and resources required and why is it needed for use.
* Visuals- It contains the raster images which were used for analysis and predicting the results.
* Code- This section contains the basic or snippet code which we used to calculate NDVI values and used it for the prediction model.

The link to the web report is- <http://minorproject.gq/>

**5.Testing**

**5.1 Testing with Image Data**

Efficient methodologies for mapping croplands are an essential condition for the implementation of sustainable agricultural practices and for monitoring crops periodically. The increasing spatial and temporal resolution of globally available satellite images, such as those provided by Sentinel-2, creates new possibilities for generating accurate datasets on available crop types, in ready-to-use vector data format. Existing solutions dedicated to cropland mapping, based on high resolution remote sensing data, are mainly focused on pixel-based analysis of time series data. This paper evaluates how a time-weighted dynamic time warping (TWDTW) method that uses Sentinel-2 time series performs when applied to pixel-based and object-based classifications of various crop types in three different study areas (in Romania, Italy and the USA). The classification outputs were compared to those produced by Random Forest (RF) for both pixel- and object-based image analysis units. The sensitivity of these two methods to the training samples was also evaluated. Object-based TWDTW outperformed pixel-based TWDTW in all three study areas, with overall accuracies ranging between 78.05% and 96.19%; it also proved to be more efficient in terms of computational time. TWDTW achieved comparable classification results to RF in Romania and Italy, but RF achieved better results in the USA, where the classified crops present high intra-class spectral variability. Additionally, TWDTW proved to be less sensitive in relation to the training samples. This is an important asset in areas where inputs for training samples are limited.

### 5.2 Visualize single and multi-band imagery

You can plot indivudal layers of a multi-spectral image, but they are often combined to create more interesting plots. To combine three bands, we can use plotRGB. To make a “true color” image (that is, something that looks like a normal photograph), we need to select the bands that we want to render in the red, green and blue regions. For this Landsat image, r = 3 (red), g = 2(green), b = 1(blue) will plot the true color composite (vegetation in green, water blue etc). You can also supply additional arguments to plotRGB to improve the visualization (e.g. a linear stretch of the values, using strecth = "lin").

plotRGB(r, r **=** 3, g **=** 2, b **=** 1, axes **=** TRUE, stretch **=** "lin",

main **=** "Landsat True Color Composite")

Selecting r = 4 (NIR), g = 3 (red), b = 2(green) will plot a “false color”” composite. This representation is popular as it makes it easy to see the vegetation (in red). You can find more about the visualization[here](http://www.crisp.nus.edu.sg/~research/tutorial/opt_int.htm).

plotRGB(r, r **=** 4, g **=** 3, b **=** 2, axes **=** TRUE, stretch **=** "lin",

main **=** "Landsat False Color Composite")

The same for the Sentinel image, using the layout function to both reprentations next to each other:

nf **<-** layout(matrix(c(1,0,2), 1, 3, byrow **=** TRUE), width **=** c(1,0.2,1), respect **=** TRUE)

plotRGB(s, r **=** 3, g **=** 2, b **=** 1, axes **=** TRUE, stretch **=** "lin",

main **=** "Senitnel True Color Composite")

plotRGB(s, r **=** 7, g **=** 3, b **=** 2, axes **=** TRUE, stretch **=** "lin",

main **=** "Sentinel False Color Composite")

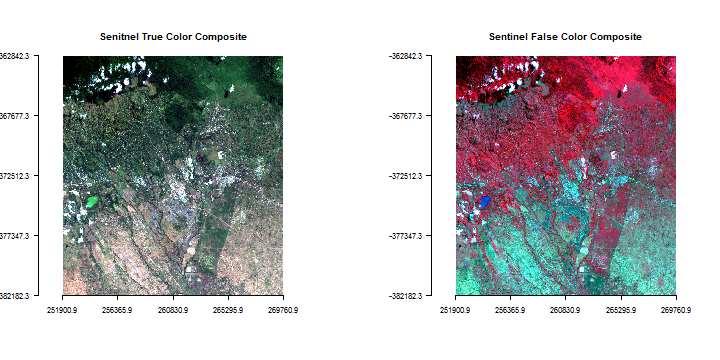


Figure 5.1

### 5.3 Subset and rename spectral bands

You can select specific layers (bands) using subset function, or via indexing.

# select first 3 bands only

rsub <- subset(r, 1:3)

# same result

rsub <- r[[1:3]]

# Check the number of bands in orginal and new data

nlayers(r)

## [1] 6

nlayers(rsub)

## [1] 3

Set the names of the bands using the following:

*# For LANDSAT*

names(r)

*## [1] "landsat8.2016march.1" "landsat8.2016march.2" "landsat8.2016march.3"*

*## [4] "landsat8.2016march.4" "landsat8.2016march.5" "landsat8.2016march.6"*

names(r) **<-** c('blue','green','red','NIR','SWIR1','SWIR2')

names(r)

*## [1] "blue" "green" "red" "NIR" "SWIR1" "SWIR2"*

*# For SENTINEL*

names(s)

*## [1] "sentinel.1" "sentinel.2" "sentinel.3" "sentinel.4" "sentinel.5"*

*## [6] "sentinel.6" "sentinel.7" "sentinel.8" "sentinel.9" "sentinel.10"*

names(s) **<-** c('blue','green','red','rededge1','rededge2',

'rededge3','NIR','rededge4','SWIR1','SWIR2')

### 5.4 Extract raster values

Often we require value(s) of raster cell(s) for a geographic location/area. The extract function is used to get raster values at the locations of other spatial data. You can use points, lines, polygons or an Extent (rectangle) object. You can also use cell numbers to extract values.If using points, extract returns the values of a Raster\* object for the cells in which a set of points fall.

Below, we extract values of Sentinel data for a set of samples. These land cover at the sample locations have been classified into different classes including cloud, forest, crop, fallow, built-up, open-soil, water and grassland.

*# extract values with points*

samp **<-** readRDS('rsdata/samples.rds')

df **<-** extract(s, samp)

*# To see some of the reflectance values*

head(df)

*## blue green red rededge1 rededge2 rededge3 NIR rededge4 SWIR1 SWIR2*

*## [1,] 1531 1466 1399 1428 2236 2477 2626 2698 1831 1083*

*## [2,] 2735 2643 2737 2908 3597 3957 3979 4163 3667 2560*

*## [3,] 2487 2424 2505 2699 3583 3971 3929 4230 3371 2221*

*## [4,] 2262 2195 2176 2424 3416 3811 3733 4067 3254 2198*

*## [5,] 2299 2219 2194 2381 3268 3622 3632 3860 2988 1980*

*## [6,] 1962 2012 2078 2366 3553 4056 3810 4398 3109 1949*

## 5.5 Image classification

There are two classification methods: unsupervised and supervised. Various unsupervised and supervised classification algorithms exist, and the choice of algoritm can affect the results. We will explore two k-means (unsupervised) and decision tree (supervised) algorithms.

### 5.5.1 Supervised classification

In a supervised classification, we have prior knowledge about some of the land-cover types through, for example, fieldwork or interpretation of high resolution imagery (such as avaialble on Google maps). Specific sites in the study area that represent homogeneous examples of these known land-cover types are identified. These areas are commonly referred to as training sites because the spectral properties of these sites are used to train the classification algorithm. After that, the entire image is classified using the trained algorithm.

## 5.6Basic mathematical operations

The raster package supports many mathematical operations. Math operations are generally performed per pixel. First we will learn about basic arithmetic operations on bands. First example is a custom math function that calculates the Normalized Difference Vegetation Index (NDVI).

### 5.6.1Compute vegetation indices

Let’s define a general function for ratio based vegetation index.

# i and k are the index of bands to be used for the indices computation

NDVI <- function(img, i, k) {

bi <- img[[i]]

bk <- img[[k]]

vi <- (bk - bi) / (bk + bi)

return(vi)

}

# For Sentinel NIR = 7, red = 3.

ndvi <- NDVI(ss, 3, 7)

plot(ndvi, col = rev(terrain.colors(30)), main = 'NDVI from Sentinel')

**5.7 Regression Model**

Regression models are typically “global”. That is, all date are used simultaneously to fit a single model. In some cases it can make sense to fit more flexible “local” models. Such models exist in a general regression framework (e.g. generalized additive models), where “local” refers to the values of the predictor values. In a spatial context local refers to location. Rather than fitting a single regression model, it is possible to fit several models, one for each location (out of possibly very many) locations. This technique is sometimes called “geographically weighted regression” (GWR). GWR is a data exploration technique that allows to understand changes in importance of different variables over space (which may indicate that the model used is misspecified and can be improved).

There are two examples here. One short example with California precipitation data, and than a more elaborate example with house price data.

5.7 California precipitation

Here is an example of GWR with California precipitation data. Get the data ([precipitation data](http://rspatial.org/analysis/data/precipitation.csv) and[counties](http://rspatial.org/analysis/data/counties.rds)).

cts <- readRDS('data/counties.rds')

p <- read.csv('data/precipitation.csv')

head(p)

## ID NAME LAT LONG ALT JAN FEB MAR APR MAY JUN

## 1 ID741 DEATH VALLEY 36.47 -116.87 -59 7.4 9.5 7.5 3.4 1.7 1.0

## 2 ID743 THERMAL/FAA AIRPORT 33.63 -116.17 -34 9.2 6.9 7.9 1.8 1.6 0.4

## 3 ID744 BRAWLEY 2SW 32.96 -115.55 -31 11.3 8.3 7.6 2.0 0.8 0.1

## 4 ID753 IMPERIAL/FAA AIRPORT 32.83 -115.57 -18 10.6 7.0 6.1 2.5 0.2 0.0

## 5 ID754 NILAND 33.28 -115.51 -18 9.0 8.0 9.0 3.0 0.0 1.0

## 6 ID758 EL CENTRO/NAF 32.82 -115.67 -13 9.8 1.6 3.7 3.0 0.4 0.0

## JUL AUG SEP OCT NOV DEC

## 1 3.7 2.8 4.3 2.2 4.7 3.9

## 2 1.9 3.4 5.3 2.0 6.3 5.5

## 3 1.9 9.2 6.5 5.0 4.8 9.7

## 4 2.4 2.6 8.3 5.4 7.7 7.3

## 5 8.0 9.0 7.0 8.0 7.0 9.0

## 6 3.0 10.8 0.2 0.0 3.3 1.4

plot(cts)

points(p[,c('LONG', 'LAT')], col='red', pch=20)

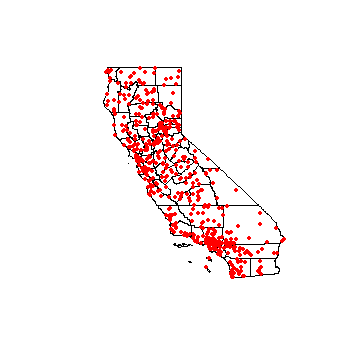


Figure 5.2

Global regression model

m <- lm(pan ~ ALT, data=p)

m

##

## Call:

## lm(formula = pan ~ ALT, data = p)

##

## Coefficients:

## (Intercept) ALT

## 523.60 0.17

Create Spatial\* objects with a planar crs.

alb **<-** CRS("+proj=aea +lat\_1=34 +lat\_2=40.5 +lat\_0=0 +lon\_0=-120 +x\_0=0 +y\_0=-4000000 +ellps=GRS80 +datum=NAD83 +units=m +no\_defs")

sp **<-** p

coordinates(sp) **=** **~** LONG **+** LAT

crs(sp) **<-** "+proj=longlat +datum=NAD83"

spt **<-** spTransform(sp, alb)

ctst **<-** spTransform(cts, alb)

Get the optimal bandwidth

library( spgwr )

## Error in library(spgwr): there is no package called 'spgwr'

bw <- gwr.sel(pan ~ ALT, data=spt)

## Error in gwr.sel(pan ~ ALT, data = spt): could not find function "gwr.sel"

bw

## Error in eval(expr, envir, enclos): object 'bw' not found

Create a regular set of points to estimate parameters for.

r **<-** raster(ctst, res**=**10000)

r **<-** rasterize(ctst, r)

newpts **<-** rasterToPoints(r)

Run the gwr function

g <- gwr(pan ~ ALT, data=spt, bandwidth=bw, fit.points=newpts[, 1:2])

## Error in gwr(pan ~ ALT, data = spt, bandwidth = bw, fit.points = newpts[, : could not find function "gwr”

## Error in eval(expr, envir, enclos): object 'g' not found

Link the results back to the raster

slope <- r

intercept <- r

slope[!is.na(slope)] <- g$SDF$ALT

## Error in eval(expr, envir, enclos): object 'g' not found

intercept[!is.na(intercept)] <- g$SDF$'(Intercept)'

## Error in eval(expr, envir, enclos): object 'g' not found

s <- stack(intercept, slope)

names(s) <- c('intercept', 'slope')

plot(s)

**6 .Conclusion**

What we have done is just a prediction and not a judgement. We are not assuring that the value which we predicted will be the value of next band, It’s just a prediction. That means the probability of happening that thing increases by much more.

This Project really helped us to get expose to real working environment as an IT undergraduate. It developed our skills and gave us brief picture about R programming and Data Analysis, specially Spatial Data Analysis. It increased chances and interest for our carrier in Data Analysis.

This project not only strengthen our technical skills but also enhanced our Interpersonal skills, creativity, critical thinking and problem solving approach. It helped us to use our theoretical knowledge and implement practically to solve problems. Interacting with team mates also increased our social skills. We are glad that our University has included this Minor Project as a part of our curriculum so that we can have some experience about work environment and new and demanding technologies in the IT industry.

##### APPENDIX 1

# Satellite Imagery Aids in Prediction and Prevention of Food Crises in USA

Commercial satellite imaging has penetrated the earth’s layers and made the concept of location-based services popular across industries. Now the technology which has empowered so much commercial progress is being called on to address some of the world’s most significant humanitarian crises.

Food supply predictions and forecasts such as those produced by satellite imaging companies could go a long way in mitigating the effects of a food shortage, assuming proper responsive measures are taken. Satellite imaging solutions, along with the integration of remote sensing technologies and high-resolution cameras, could mean a whole new scenario for food shortage prediction. Images captured by satellites could aid in crop assessment, map irrigated landscape, give a precise soil and environmental analysis, and predict crop yields. By careful assessment of the local food environments, and changes in land use and land cover, satellite imaging can help researchers develop plans for sustaining our food resources. Proper planning, better management of existing agricultural lands, and improved production practices can lead to a secure food supply at a global, as well as local, scale.

Geospatial information systems and remote sensing technologies, when used in sync with satellite imaging, are capable of precise food production forecasts. Due to these advanced applications of GIS and remote sensing technologies in the immediate future, the market for satellite images is expected to witness a huge boost in size. Expert analysts at Allied Market Research have predicted that the [commercial satellite imaging market](https://www.alliedmarketresearch.com/commercial-satellite-imaging-market) is expected to bring in $5,275 million by 2022.

##### APPENDIX 2

# Investigation of Drought prediction using remote-sensing vegetation indices for different time spans in Iran

Iran is a country in a dry part of the world and extensively suffers from drought. Drought is a natural and repeatable phenomenon definable at specified time and area. In addition, social and economic issues can be affected by drought. Information such as intensity, duration, and spatial coverage of drought can help decision makers to reduce the vulnerability of the drought-affected areas, therefore lessen the risks associated with drought episodes. Lack of long-term meteorological data for many parts of the country is one of the most important problems for drought monitoring in Iran. One of the useful ways for gathering information about soil and vegetation conditions is using satellite-based imagery. In this study, remotely sensed image data were applied in order to forecast and model the drought. To this end, SPI (standardized precipitation index) drought indicator was used to represent the drought and its intensity in different time spans (1, 3, 6, 9, 12, and 24 months). Some vegetation indices (VIs) including normalized difference vegetation index, temperature condition index, vegetation condition index, and normalized difference vegetation index deviation were extracted using Advanced Very High Resolution Radiometer sensor imagery. These indices were plugged into the model to calculate the SPI. A unique Support Vector Machine classifier improved for all types of the SPI by applying various remotely sensed VIs. The best vegetation index for each kind of SPI was determined. In this framework, meteorological stations were clustered based on their land cover extracted from satellite-based indices before insertion to the model.

**References**

1. [A. A. Gitelson](https://www.tandfonline.com/author/Gitelson%2C+A+A),[R. Stark](https://www.tandfonline.com/author/Stark%2C+R),[U. Grits](https://www.tandfonline.com/author/Grits%2C+U),[D. Rundquist](https://www.tandfonline.com/author/Rundquist%2C+D),[Y. Kaufman](https://www.tandfonline.com/author/Kaufman%2C+Y) & [D. Derry](https://www.tandfonline.com/author/Derry%2C+D) (2010) ‘Vegetation and soil lines in visible spectral space: A concept and technique for remote estimation of vegetation fraction’ pp.2537-2562.

2. [Mushtaq Ahmad Ganie](https://www.researchgate.net/profile/Mushtaq_Ganie3) & [Dr Asima Nusrath](https://www.researchgate.net/scientific-contributions/2105828513_Dr_Asima_Nusrath) (2016) ‘Determining the Vegetation Indices (NDVI) from Landsat 8 Satellite Data.’ pp1459-1463.