Manual on

Climate data generation, mapping and climate-disease relation modelling using R

ICAR - National Institute of Veterinary Epidemiology and Disease Informatics,

Post Box No. 6450, Ramagondanahalli, Yelahanka, Bengaluru, Karnataka 560064

(An ISO 9001:2015 Certified Institute)

Prepared By

Dr. K. P. Suresh

Dr. P. Krishnamoorthy

Dr. Siju Susan Jacob

Technical Assistance

Rashmi R. Kurli

Dheeraj R.

Sudhagar S.

Kiran Kumar S.

Published by

Director,

ICAR-NIVEDI, Bengaluru

ಭಾ.ಕೃ.ಸಂ.ಪ – ರಾಷ್ಟ್ರೀಯ ಪಶುರೋಗ ಸೋಂಕುಶಾಸ್ತ್ರ ಮತ್ತು ಮಾಹಿತಿ ವಿಜ್ಞಾನ ಸಂಸ್ಥೆ
भाकृअनुप–राष्ट्रीय पशुरोग जानपदिक एवं सूचना विज्ञान संस्थान
ICAR–National Institute of Veterinary Epidemiology and Disease Informatics
ರಾಮಗೊಂಡನಹಳ್ಳಿ, ಯಲಹಂಕ, ಬೆಂಗಳೂರು – ೫೬೦೦೬೪, ಭಾರತ.
रामगोंडनहल्लि, येलहंका, बेंगलूरू – ५६० ०६४, भारत.
Ramagondanahalli, Post Box No: 6450, Yelahanka, Bengaluru – 560064, INDIA.

Dr. Parimal Roy, M.V.Sc.,Ph.D
Director

Foreword

Environmental condition prevailing in the world is indistinguishably linked to human and animal health. It is the environment which helps the pathogen for transmitting to the susceptible host and to cause the disease. The environmental parameters that influencing the occurrence of the disease can be extracted from various data sources. The manual on Climate data generation, mapping and modeling the impact of climate change on livestock diseases using R is intended to describe the installation of software, data generation, data representation and address the various steps for performing desired analysis. The practical knowledge presented in this booklet is up to date and has been organized in a very practical terms, keeping in mind the facilities available and difficulties encountered in handling the software. The analysis of climate data will help in geographical surveillance of livestock diseases and their influence on disease occurrence.

This manual provides how the basic information on remote sensing and disease-climate modeling can be generated and also help in better correlation of disease occurrence and climate change. I am happy that the manual developed by our team of scientists and project staff under National Innovation in Climate Resilient Agriculture (NICRA) project will help the scientists working on generation of climate data and its analysis using open source software. I congratulate the team for preparing this manual and providing a useful source of resources for climate change studies, which will go a great way in the research on climate change. This manual is unique and useful for the researchers working in the field of animal health and livestock diseases.

(Parimal Roy)

Preface

The environmental variables are very crucial in determining the epidemiology and transmission pattern of infectious diseases and analyzing these variables will enhance the accuracy of disease prediction that ultimately helps to implement timely and effective control measures. The emergence of Remote sensing satellites provided a wide array of environmental variables at different spatial and temporal scales which created an avenue to increase our understanding about the correlation between diseases and a variety of environmental/climatic variables. Remote sensing offers the ability to observe and collect data for large area as relatively quickly, and is an important source of data for Geographical Information System (GIS). The generation of remote sensing data manually is a difficult task since it demands handling of huge data that require more time, manpower, storage space etc. The extraction of these variables from the remote sensing data products and databases are made easy by the use of software like R, an open source software available to all the researchers. R is designed for statistical computing, with thousands of packages that contain the implementations of almost every available statistical methods. R is an integrated suite of software facilities for data management, calculation and graphical display.

This manual states the importance of understanding the climate variables influencing the livestock disease incidence and the change in the climate profoundly alters disease dynamics. It provides guidelines on installation of R software and RStudio. A skeleton R codes are written to generate the remote sensing parameters and meteorological parameters from the satellite datasets available from different sources. It also includes the making of point maps and intensity maps for different geographical boundaries using R software. Protocol to develop climate-disease models and risk maps to predict the disease occurrence developed using R software.

It is hoped that this publication will be useful in achieving its objective to provide a practical, easy-to-use instructions for the data and geostatistical analysis, representation and simplified interpretations. This manual will help the researchers in the field of climate change studies and its relationship to livestock diseases and animal health. We thank the Deputy Director General (AS), ICAR, New Delhi and our Director for their support and encouragement in bringing out this manual. The editors thank the National Innovations in Climate Resilient Agriculture (NICRA) project for providing funds and support in preparing this manual.

K. P. Suresh

P. Krishnamoorthy

Siju Susan Jacob

Contents

I. Introduction:

In the era of global climate change, the burden of infectious diseases of livestock is alarmingly increasing. As the scientific evidence of climate change is unequivocal, the challenge of reducing the incidence and transmission of infection diseases in livestock is a growing concern especially in developing countries. As it is envisaged that climate change is profoundly modulating the survivability and transmissibility of pathogens, the situation of emergence of new species or strains of pathogen with increased virulence and shift in the geographic distribution of diseases especially vector borne diseases may be anticipated in the near future. A better understanding of disease incidence with respect to climatic variables is essentially required for speculating the future incidence trends with respect to climate change. In order to establish the link between climate change and infectious diseases, it demands (i) examining the evidence for association between climate variability and incidence of disease from recent past, (ii) determining early indicators of already emerging infectious diseases and (iii) use of above mentioned evidences to create predictive models to estimate future burden of infectious disease under projected climate change scenarios. In order to understand the effect of climate change on infectious diseases both statistical and mathematical models have imperative roles to play; statistical models generally use descriptive correlations between explanatory (climate change) and response (disease incidence) variables to predict the future trend. Such models may not be suitable for underpinning the biological mechanisms behind the predictive changes. On the other hand, mathematical models following a process based approach by combining different environmental, epidemiological and biological process to formulate assumptions that characterize models. Further, model calibration and validation may provide reliable means to predict short term and long term disease dynamics.

The use of remotely-sensed data in mapping of disease incidence and as source of input data for environmental processes modeling has become popular in recent years. With the availability of remotely-sensed data from different sensors of various platforms with a wide range of spatiotemporal, radiometric and spectral resolutions has made remote sensing as one of the best source of data for disease mapping. The extraction of environmental variables remotely sensed data using software like R will help to correlate these environmental variables with disease incidence that ultimately will help to understand the pattern of disease distribution.

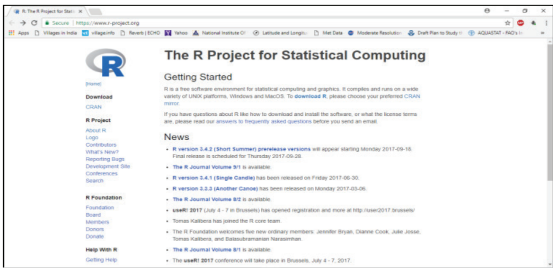
II. About R software

Ross Ihaka and Robert Gentleman created a programming at the University of Auckland, New Zealand, they named it 'R' based on their first letter of first names. R is free online software with pre-compiled binary versions for various operating systems like Windows, Linux and Mac. It is an integrated suite of software for data management, calculation, statistical analysis, graphical representation and reporting, it also has an effective data handling and storage facility. R is world's most widely used statistics programming language. As of June 2018, R ranks 10th in the TIOBE index, a measure of popularity of programming languages. RStudio is the most commonly used graphical integrated development environment.

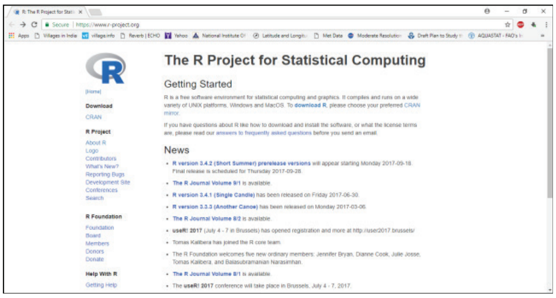
A. Installation Protocol

i. R Software installation

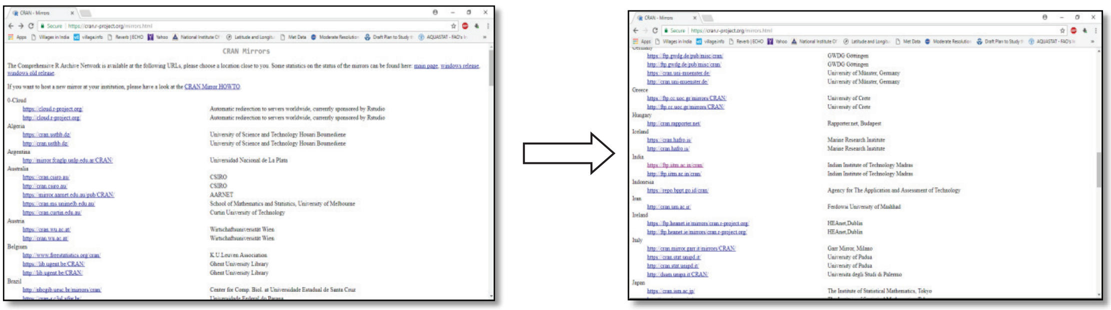
1. Open an internet browser and go to www.r-project.org. and Click the "download R" link in the middle of the page under "Getting Started."



2. Select a CRAN location (a mirror site) and click the corresponding link.

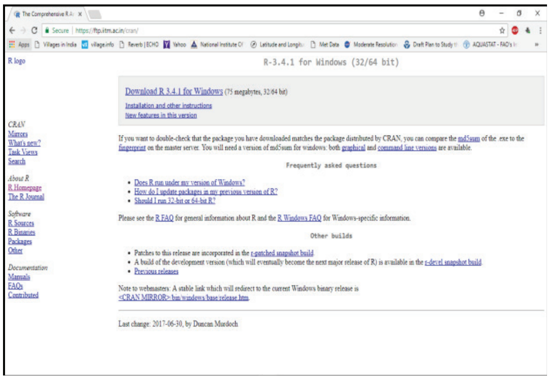


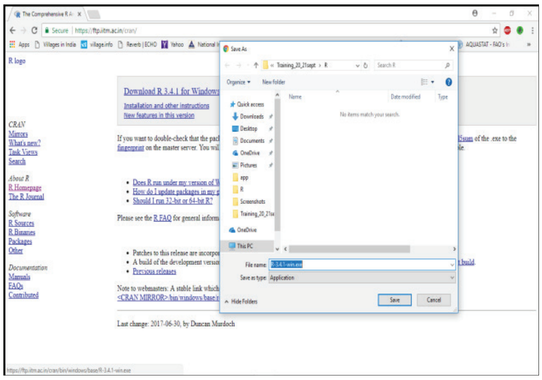
3. Scroll down to select the closest place for you



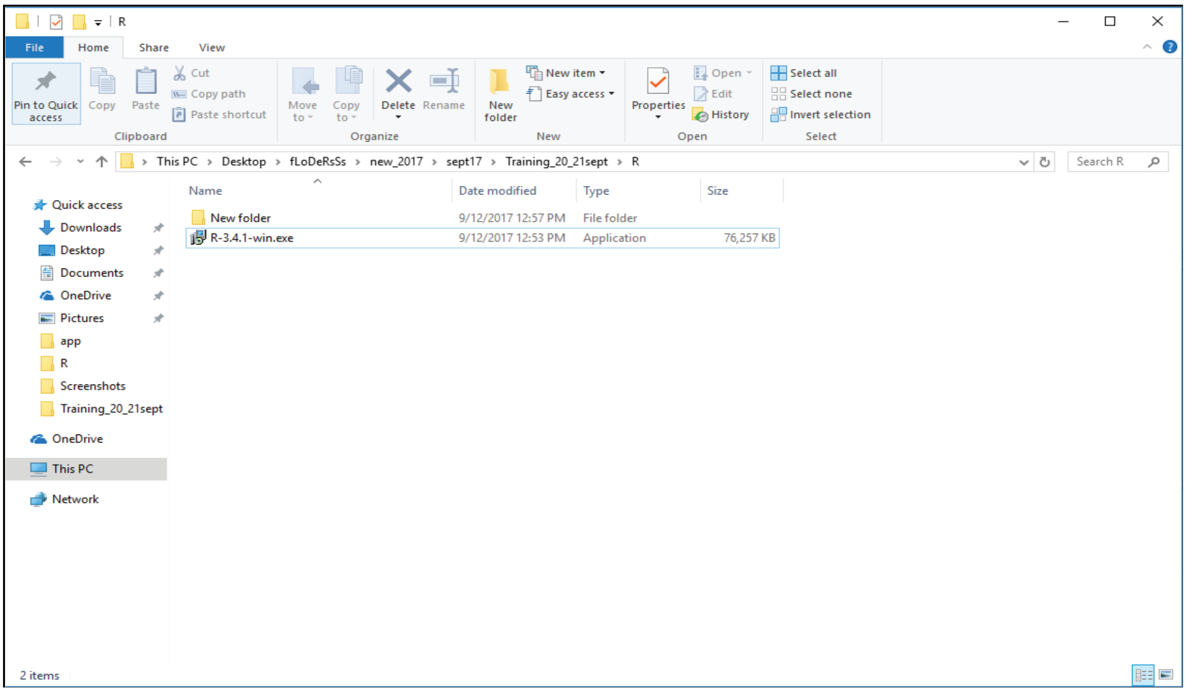
4. Click on the link "Download R for Windows" for suitable Operating System.

5. Click on install R software and Save the file in the desired folder

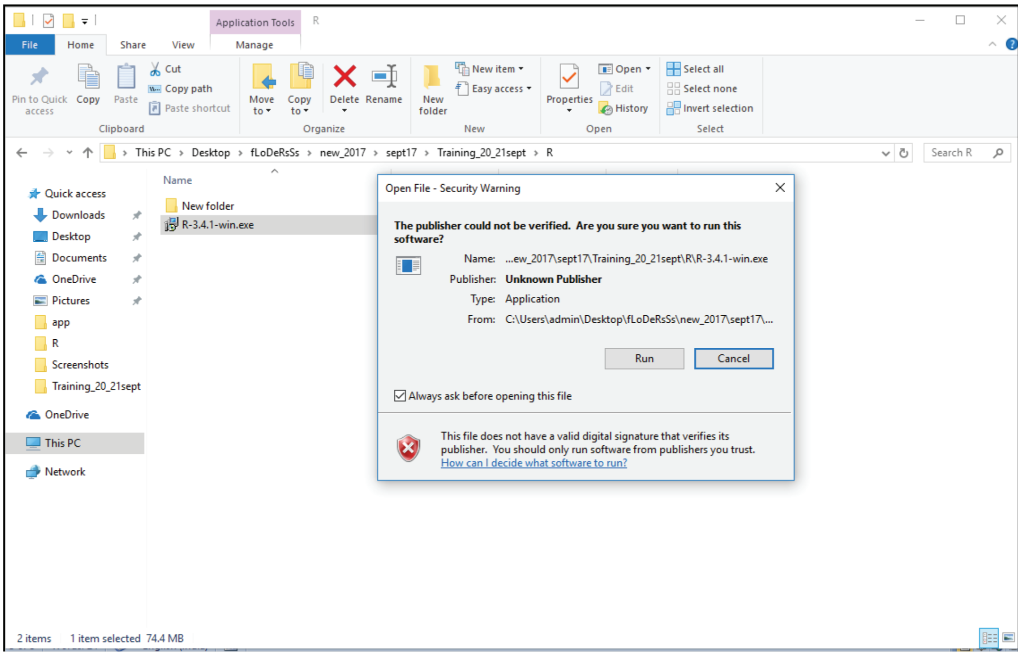




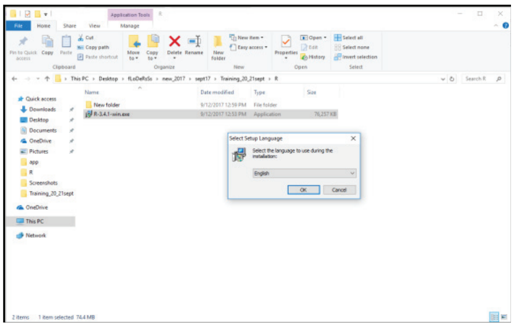
6. Click on the file

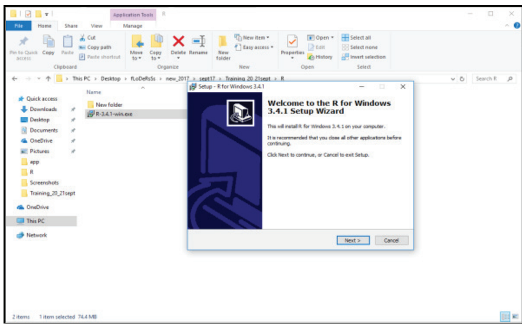


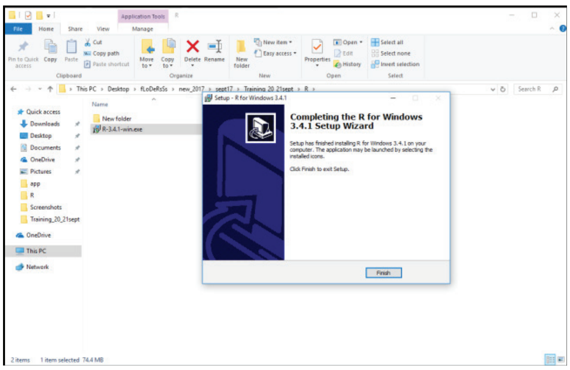
7. Run it

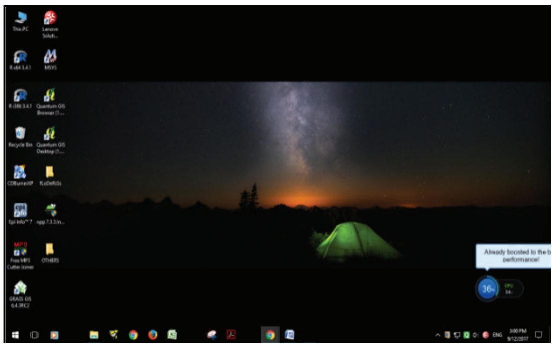


8. Follow the instructions to install the software on your PC



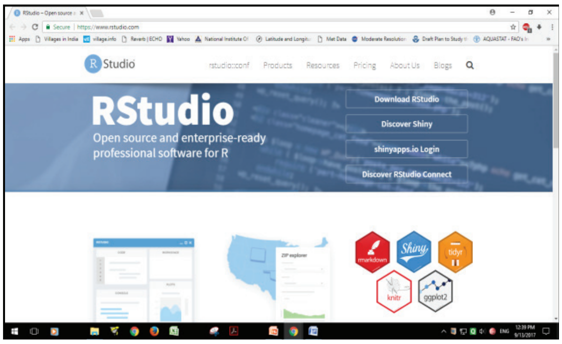


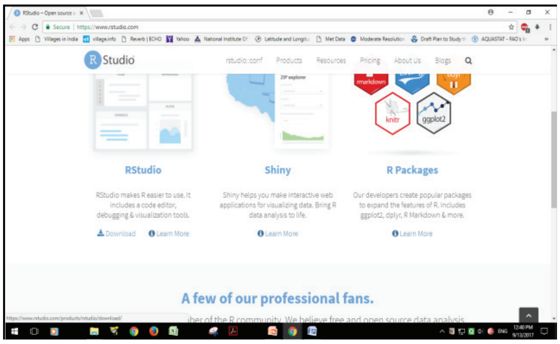




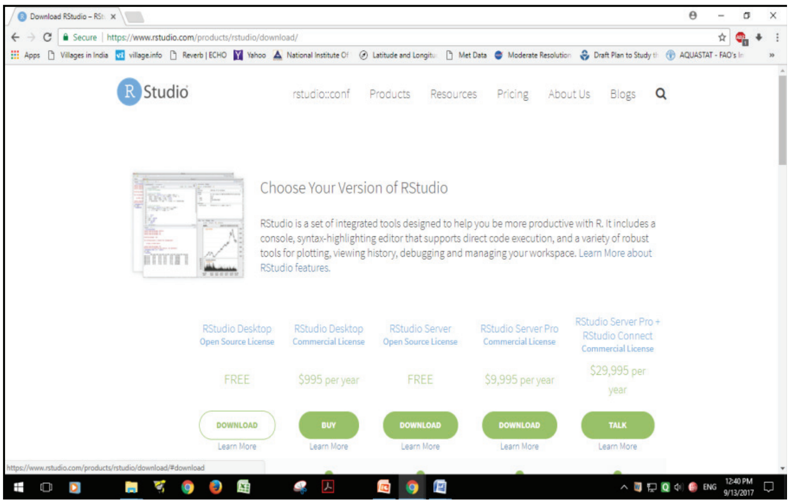
ii. R Studio Installation

1. Go to www.rstudio.com and Click on the "Download RStudio" button.

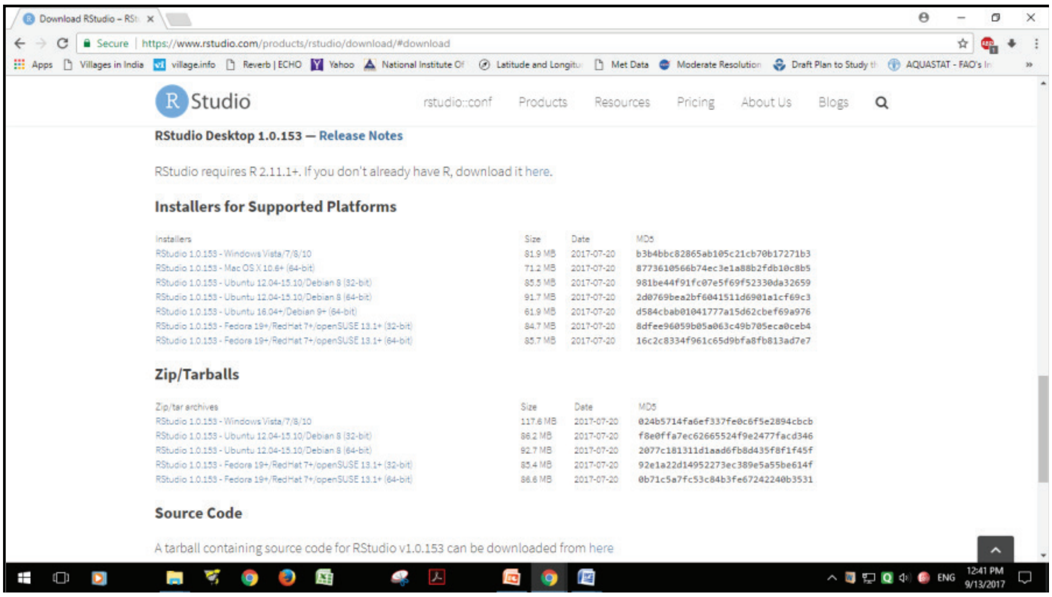




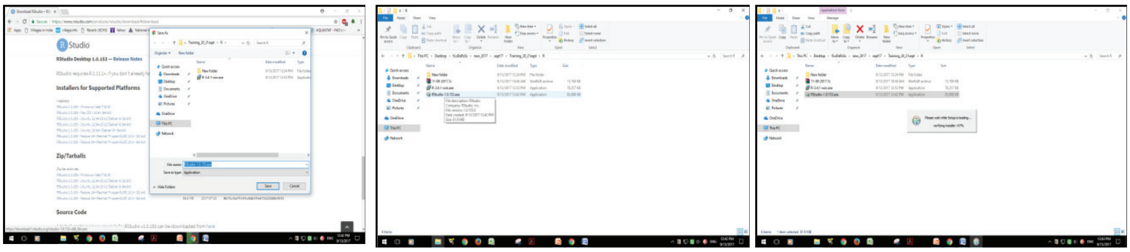
2. Click on "Download RStudio Desktop Open Source."



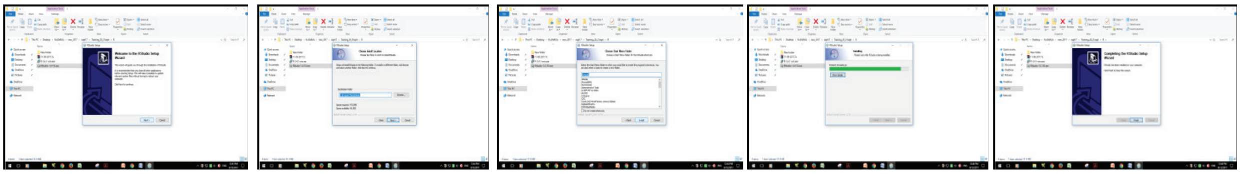
3. Click on the version recommended for your system, or the latest Windows version



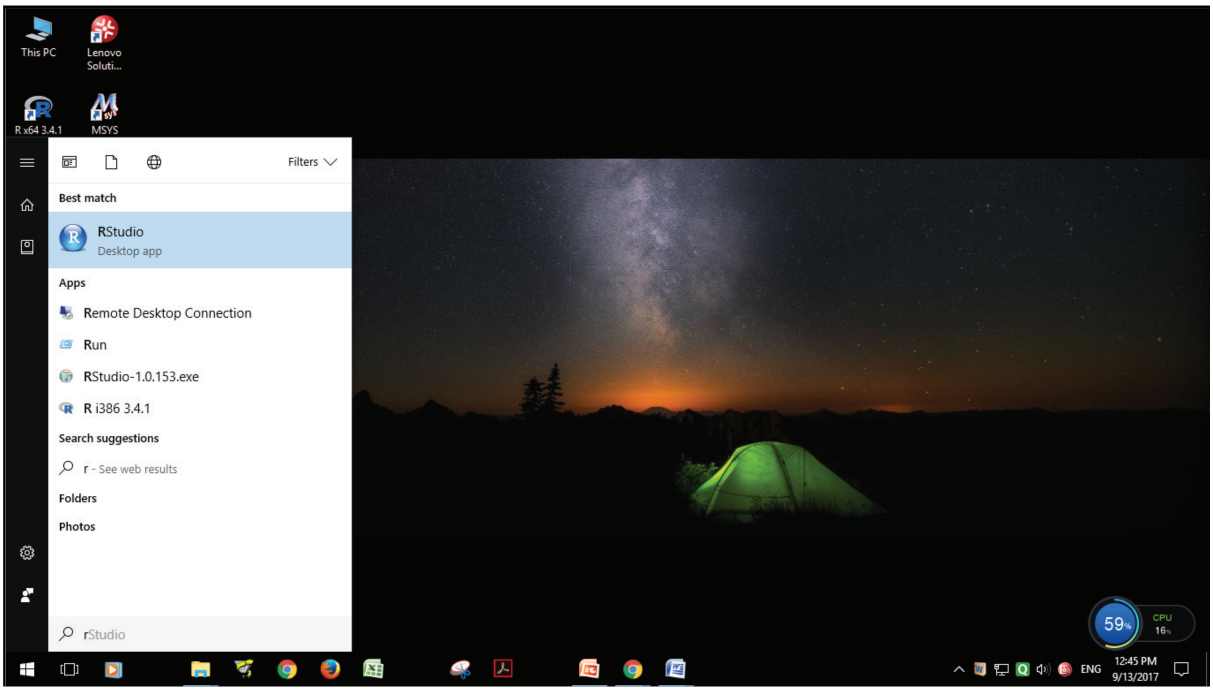
4. Save the executable file and Run the .exe file



5. Follow the installation instructions and click on finish option when the window pops up



6. Now the RStudio is installed on your PC



III. Remote sensing and GIS

Introduction

Remote sensing is a multi-disciplinary science which forms a complete integrated system to detect, monitor and obtain information or physical characteristics of objects or areas at the earth's surface by measuring its reflected and emitted radiation using instruments which are remote to the surface typically from satellites. Remote sensing enables to observe and collect data for a broad area at a time and to observe the area for a long period by which time series data is easily obtained and change detection is effortless. This substitutes the costly and slow data collection on the ground making it possible to collect data of dangerous or inaccessible areas. Remote sensing has made it easy for scientist necessitating continuous data requirements for atmospheric, ocean, and land studies at a variety of spatial and temporal scales to identify environmental attributes relevant to global change research by monitor physical and biological processes.

Applications of Remote Sensing widespread in many fields Oceanography, Glaciology, Geology, Topography and cartography , Agriculture, Hydrology, Oil and mineral exploration and Climate.

NDVI (Normalised Difference Vegetative Index)

Remote sensing phenology studies use data gathered by satellite sensors that measure wavelengths of light absorbed and reflected by green plants. To determine the density of green on a patch of land, researchers must observe the distinct colours (wavelengths) of visible and near-infrared sunlight reflected by the plants. When sunlight strikes objects, certain wavelengths of this spectrum are absorbed and other wavelengths are reflected. The pigment in plant leaves, chlorophyll, strongly absorbs visible light (from 0.4 to 0.7 µm) for use in photosynthesis. The cell structure of the leaves, on the other hand, strongly reflects near-infrared light. Many sensors carried aboard satellites measure red and near-infrared light waves reflected by land surfaces. Using mathematical formulas (algorithms), the raw satellite data about these light waves is transformed 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.

Calculations of NDVI

NDVI is calculated from the visible and near-infrared light reflected by vegetation. Healthy vegetation (left) absorbs most of the visible light that hits it, and reflects a large portion of the near-infrared light. The Normalized Difference Vegetation Index(NDVI) can be calculated by

$$NDVI = (NIR — VIS)/ (NIR + VIS)$$

Calculations of NDVI for a given pixel always result in a number that ranges from minus one (-1) to plus one (+1); however, no green leaves gives a value close to zero. A zero means no vegetation and close to +1 (0.8 - 0.9) indicates the highest possible density of green leaves. MODIS product **MOD13A1** is used to obtain NDVI values of 500m resolution. The data is captured for every 16 days.

Land Surface Temperature

Land surface temperature is how hot the "surface" of the Earth would feel to the touch in a particular location from a satellite's point of view, the "surface" is whatever it sees when it looks through the atmosphere to the ground. It could be snow and ice, the grass on a lawn, the roof of a building, or the leaves in the canopy of a forest. Thus, land surface temperature is not the same as the air temperature that is included in the daily weather report

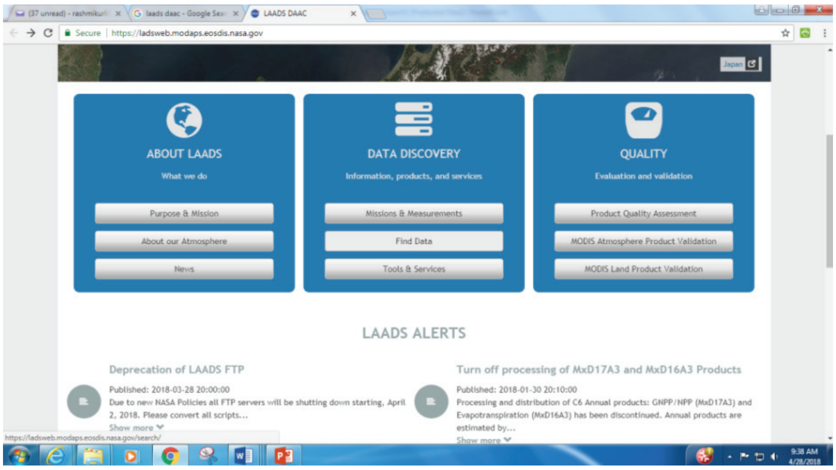
The data is collected by the Moderate Resolution Imaging Spectroradiometer (MODIS) on NASA's Terra satellite. Temperatures range from -25 degrees Celsius (deep blue) to 45 degrees Celsius (pinkish yellow). At mid-to-high latitudes, land surface temperatures can vary throughout the year, but equatorial regions tend to remain consistently warm, and Antarctica and Greenland remain consistently cold. MODIS product **MOD11A2** is used to obtain LST values of 1km resolution. The data is captured for every 8 days.

Protocol to Generate NDVI and LST

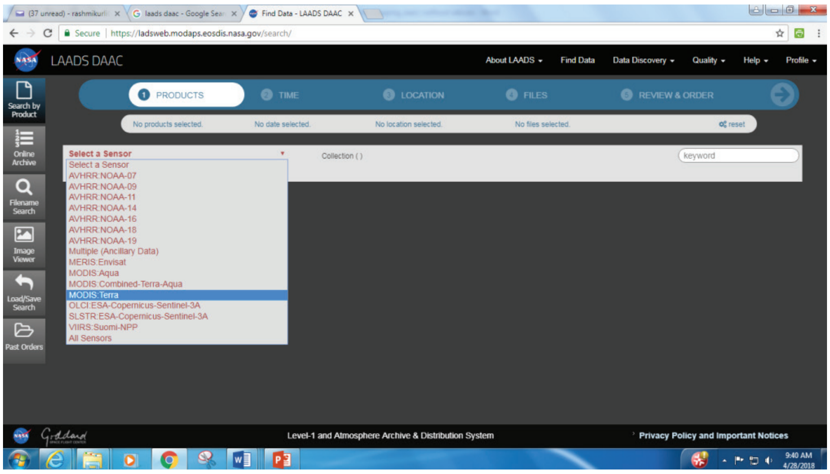
The HDF files for LST (^oC), NDVI were downloaded from the MODIS website using the MOD11A2 and MOD13A1 products respectively by specifying the coordinates and time period (dates). HDF files were then converted to TIF files using gdalUtils package of R software. The pixel values were converted to index value for NDVI (pixel value X 0.0001) and LST were converted to degree centigrade (pixel value X 0.002- 273.15 Kelvin). In NDVI the index values were considered negative for water, 0-0.1 for rock, soil and barren land. 0.2- 0.4 was taken as low vegetation, 0.41-0.6 as moderate and 0.6-0.8 as high vegetation.

1. Arrange the excel sheet (CSV/XLSX) comprising latitude & longitude and its address.
2. Install the following Packages in R software"rjava", "raster" ,"Rcurl" and "gdalUtils"

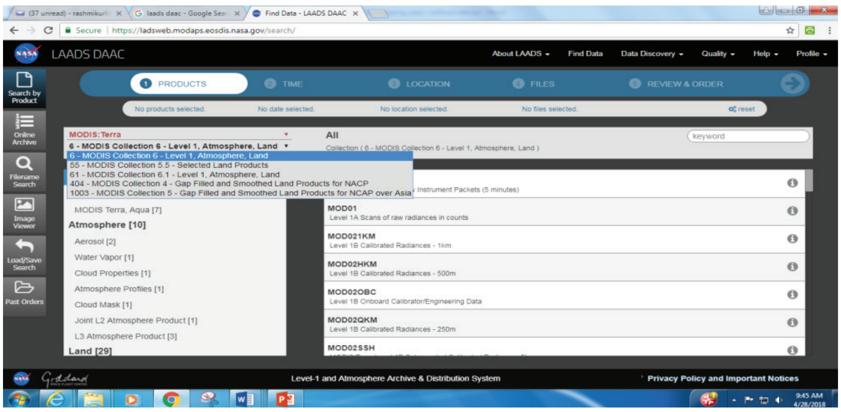
3. Download the Data from an Online Source LAADS DAAC- Click on Find data



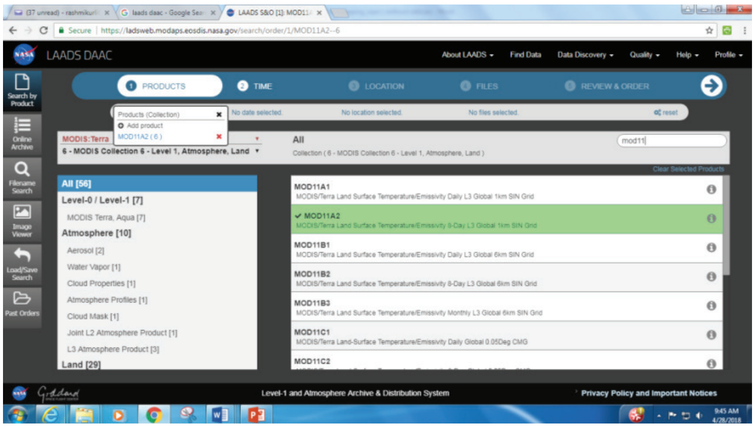
4. Select MODIS: TERRA Sensor.



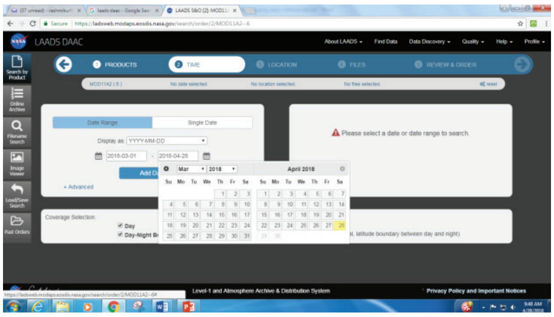
5. Select the collection- MODIS collection 6 Atmosphere, Land

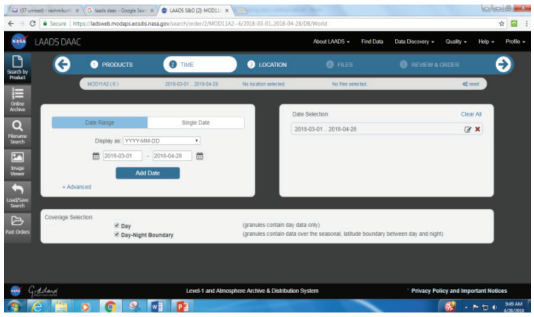


6. Select the product MOD11A2 for LST and MOD13A1 for NDVI

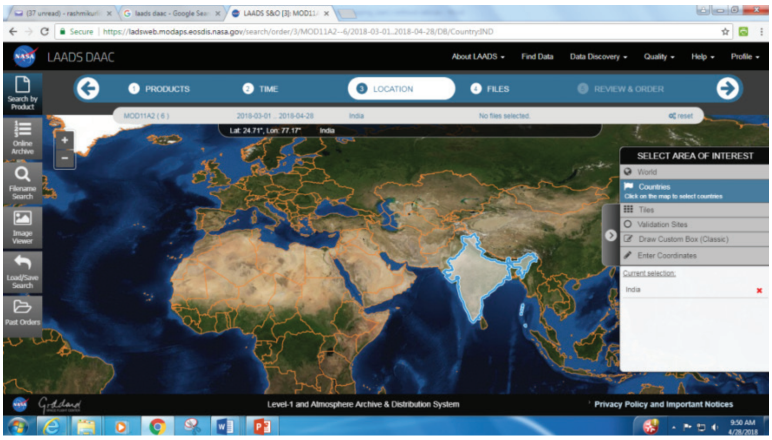


7. Specify the time period to download the values and click on add date

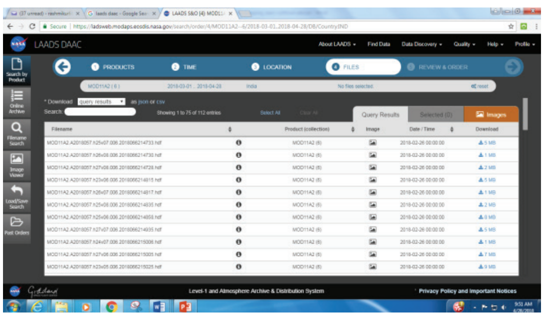


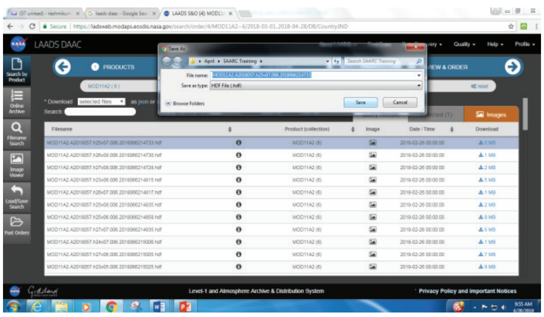


8. Select the location by specifying the area or selecting the country boundaries



9. The files will appear in the Files menu which can be downloaded by clicking on it

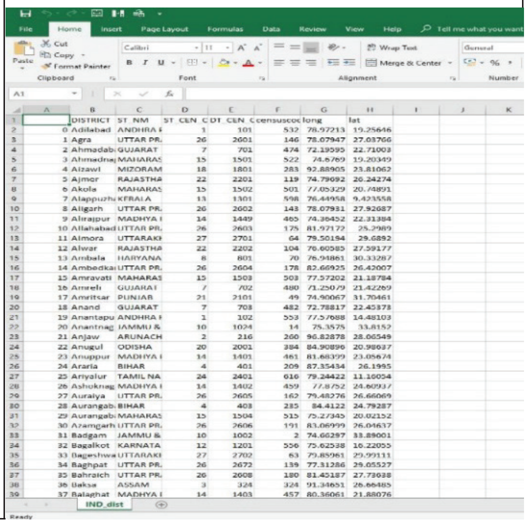




10. R code to generate input .csv file containing geo coordinates.

```
library(rgdal)
library(raster)
#Set directory by clicking on Session in the menu bar and on set working directory and select choose directory. Now choose the folder containing input files. The output files will be saved in the selected folder
s=readOGR("shp/2011_Dist.shp") # shapefile of India (District level)
s1=cbind(s@data,coordinates(s))
colnames(s1)
colnames(s1)[6:7]=c("long","lat")
write.csv(s1,"IND_dist.csv")
```

Data format



	DISTRICT	ST_NM	ST_CEN_C	DT_CEN_C	censuscoc	long	lat
0	Adilabad	ANDHRA F	1	101	532	78.97213	19.25646
1	Agra	UTTAR PR.	26	2601	146	78.07947	27.03766
2	Ahmadab	GUJARAT	7	701	474	72.19595	22.71003
3	Ahmadnag	MAHARAS	15	1501	522	74.6769	19.20349
4	Aizawl	MIZORAM	18	1801	283	92.88905	23.81062
5	Ajmer	RAJASTHA	22	2201	119	74.79092	26.24274
6	Akola	MAHARAS	15	1502	501	77.05329	20.74891
7	Alappuzh	KERALA	13	1301	598	76.44958	9.423558
8	Aligarh	UTTAR PR.	26	2602	143	78.07931	27.92687
9	Alirajpur	MADHYA I	14	1449	465	74.36452	22.31384
10	Allahabad	UTTAR PR.	26	2603	175	81.97172	25.2989
11	Almora	UTTARAKH	27	2701	64	79.50194	29.6892
12	Alwar	RAJASTHA	22	2202	104	76.60585	27.59177
13	Ambala	HARYANA	8	801	70	76.94861	30.33287
14	Ambedkar	UTTAR PR.	26	2604	178	82.66925	26.42007
15	Amravati	MAHARAS	15	1503	503	77.57202	21.18784
16	Amreli	GUJARAT	7	702	480	71.25079	21.42269
17	Amritsar	PUNJAB	21	2101	49	74.90067	31.70461
18	Anand	GUJARAT	7	703	482	72.78817	22.45378
19	Anantapu	ANDHRA F	1	102	553	77.57688	14.48103
20	Anantnag	JAMMU &	10	1024	14	75.3575	33.8152
21	Anjaw	ARUNACH	2	216	260	96.82878	28.06549
22	Anugul	ODISHA	20	2001	384	84.90896	20.98637
23	Anuppur	MADHYA I	14	1401	461	81.68399	23.05674
24	Araria	BIHAR	4	401	209	87.35434	26.1995
25	Ariyalur	TAMIL NA	24	2401	616	79.24422	11.10054
26	Ashoknag	MADHYA I	14	1402	459	77.8752	24.60937
27	Auraiya	UTTAR PR.	26	2605	162	79.48276	26.66069
28	Aurangab	BIHAR	4	403	285	84.4122	24.79287
29	Aurangab	MAHARAS	15	1504	515	75.27345	20.02152
30	Azamgarh	UTTAR PR.	26	2606	191	83.06999	26.04637
31	Badgam	JAMMU &	10	1002	2	74.66297	33.89001
32	Bagalkot	KARNATA	12	1201	556	75.62538	16.22055
33	Bageshwa	UTTARAKI	27	2702	63	79.85961	29.99111
34	Baghpat	UTTAR PR.	26	2672	139	77.31286	29.05527
35	Bahraich	UTTAR PR.	26	2608	180	81.45187	27.73638
36	Baksa	ASSAM	3	324	324	91.34651	26.66485
37	Balaghat	MADHYA I	14	1403	457	80.36061	21.88076

11. TIF file generation

```
#LST
library(gdalUtils)
library(MODIS)
files <- list.files(path="LST",pattern = glob2rx("*.hdf"),full.names = T)
#specify the directory name containing HDF files
j<-length(files)
date=extractDate(files,asDate = T)
filename <- paste0("LST/", substr(files,23,28),date$inputLayerDates,".tif")
# specify the directory name to store TIF files
i <-1
while(i<=j){
  sds <- get_subdatasets(files[i]);
  gdal_translate(sds[1], dst_dataset = filename[i]); # sds[1] LST
  i<-i+1;
}
```

```
#NDVI
library(gdalUtils)
library(MODIS)
```

```
files <- list.files(path="NDVI",pattern = glob2rx("*.hdf"),full.names = T)
#specify the directory name containing HDF files
j<-length(files)
date=extractDate(files,asDate = T)
filename <- paste0("NDVI/", substr(files,24,28),date$inputLayerDates,".tif")
# specify the directory name to store TIF files
i <-1
while(i<=j){
 sds <- get_subdatasets(files[i]);
 gdal_translate(sds[1], dst_dataset = filename[i]); # sds[1] LST
 i<-i+1;
}
```

12. **Remote sensing variable extraction**

```
#LST Measurements
ss<-read.csv("IND_dist.csv",sep=",",header=T,check.names = F) # Specify the
geocoordinates file
filename <- list.files(path="LST/",pattern = ".tif",full.names = T) #specify the
directory name containing TIF files

x<-ss$lat
y<-ss$long
data<-data.frame(y,x)
latlon1<-CRS('+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0')
coordinates1 = SpatialPoints(data,latlon1)
sinus1 = CRS("+proj=sinu +lon_0=0 +x_0=0 +y_0=0 +a=6371007.181
+b=6371007.181 +units=m +no_defs")
coordinates_sinus1 = spTransform(coordinates1,sinus1)
df_total<-NULL
i=1
for(i in 1:length(filename))
{
 my<-raster(filename[i])
```

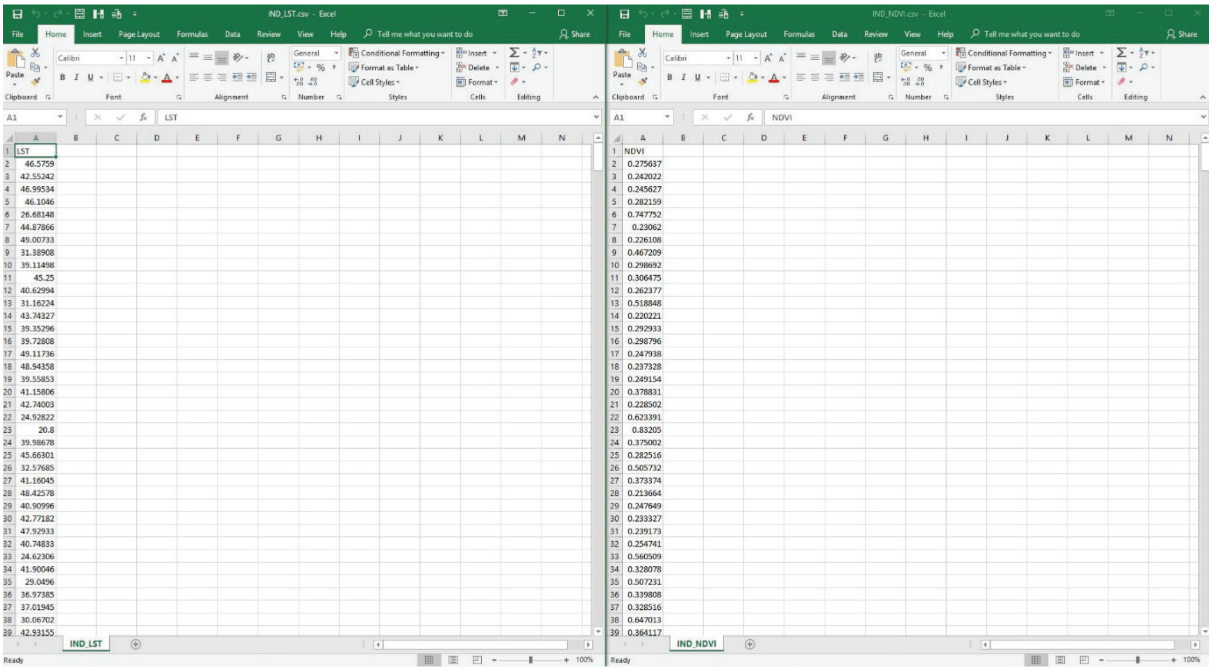
```
  my<-stack(my)
  dd<-extract(my,coordinates_sinus1)
  tempf = which(is.na(dd))
  if(length(tempf) != nrow(dd))
  {
    df_total<-cbind(df_total,dd)
  }
}

df_total = as.matrix(df_total)
write.csv(df_total,"IND_LST.csv",row.names = F) # output filename
```

```
#NDVI Measuements
ss<-read.csv("IND_dist.csv",sep=",",header=T,check.names = F) # Specify the geocoordinates file
filename <- list.files(path="NDVI/",pattern = ".tif",full.names = T)#specify the directory name containing TIF files

x<-ss$lat
y<-ss$long
data<-data.frame(y,x)
latlon1<-CRS('+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0')
coordinates1 = SpatialPoints(data,latlon1)
sinus1 = CRS("+proj=sinu +lon_0=0 +x_0=0 +y_0=0 +a=6371007.181 +b=6371007.181 +units=m +no_defs")
coordinates_sinus1 = spTransform(coordinates1,sinus1)
df_total<-NULL
i=1
for(i in 1:length(filename))
{
  my<-raster(filename[i])
  my<-stack(my)
```

```
  dd<-extract(my,coordinates_sinus1)
  tempf = which(is.na(dd))
  if(length(tempf) != nrow(dd))
  {
   df_total<-cbind(df_total,dd)
  }
 }

 df_total = as.matrix(df_total)
 write.csv(df_total,"IND_NDVI.csv",row.names = F) # output filename
```

13. Preview of output files

NDVI and LST values.



	A
1	LST
2	46.5759
3	42.55242
4	46.99534
5	46.1046
6	26.68148
7	44.87866
8	49.00733
9	31.38908
10	39.11498
11	45.25
12	40.62994
13	31.16224
14	43.74327
15	39.35296
16	39.72808
17	49.11736
18	48.94358
19	39.55853
20	41.15806
21	42.74003
22	24.92822
23	20.8
24	39.98678
25	45.66301
26	32.57685
27	41.16045
28	48.42578
29	40.90996
30	42.77182
31	47.92933
32	40.74833
33	24.62306
34	41.90046
35	29.0496
36	36.97385
37	37.01945
38	30.06702
39	42.93155

IND_LST

	A
1	NDVI
2	0.275637
3	0.242022
4	0.245627
5	0.282159
6	0.747752
7	0.23062
8	0.226108
9	0.467209
10	0.298692
11	0.306475
12	0.262377
13	0.518848
14	0.220221
15	0.292933
16	0.298796
17	0.247938
18	0.237328
19	0.249154
20	0.378831
21	0.228502
22	0.623391
23	0.83205
24	0.375002
25	0.282516
26	0.505732
27	0.373374
28	0.213664
29	0.247649
30	0.233327
31	0.239173
32	0.254741
33	0.560509
34	0.328078
35	0.507231
36	0.339808
37	0.328516
38	0.647013
39	0.364117

IND_NDVI

IV. Meteorological Parameters

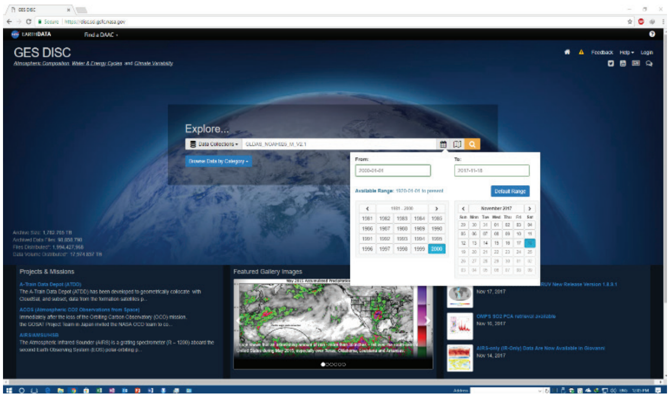
Introduction

All meteorological parameters are subject to short-term variations, normally caused by turbulences within the atmosphere. They are influenced by solar radiation, directly or indirectly, and this results in typical daily or yearly trends.The main meteorological parameters in this field are: Temperature, Pressure, Sea level pressure, Precipitation, Perceptible water, Zonal wind, Meridional wind, Relative Humidity etc.

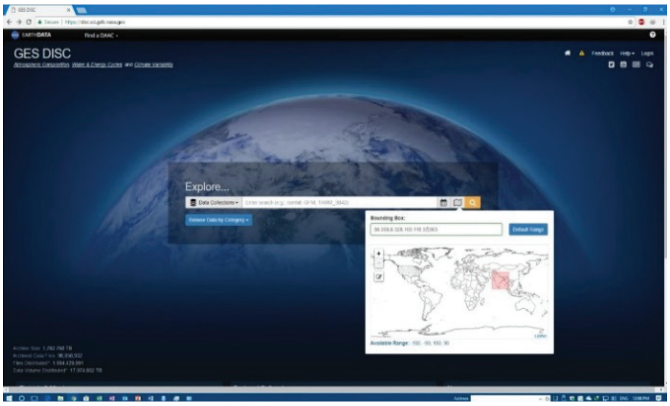
Protocol for generating Environmental parameters

GLDAS Noah Land Surface Model containing the environmental parameters such as Potential evaporation rate (W m-2), Pressure (Pa), Specific humidity (kg/kg), Total precipitation rate (kg m-2 s-1), Soil moisture (kg m-2), Temperature (K), Wind speed (m/s) were downloaded and data was extracted.

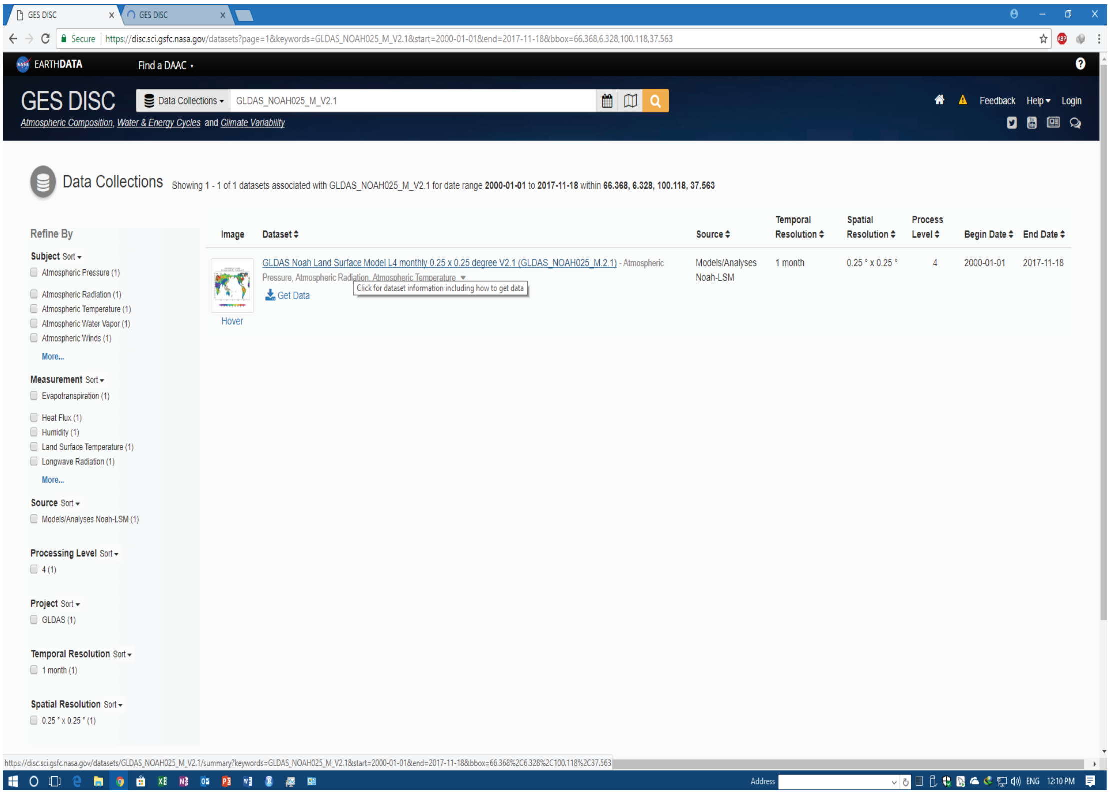
1. Arrange excel sheet (CSV/XLSX) comprising latitude, longitude and address.
2. Data was downloaded from the "GLDAS_NOAH025_M_V2.1" Dataset (https://disc.sci.gsfc.nasa.gov/) by setting the start and end dates.



3. The extent of map was set to the boundary of India map by drawing rectangular box.



4. The dataset was downloaded from search results.



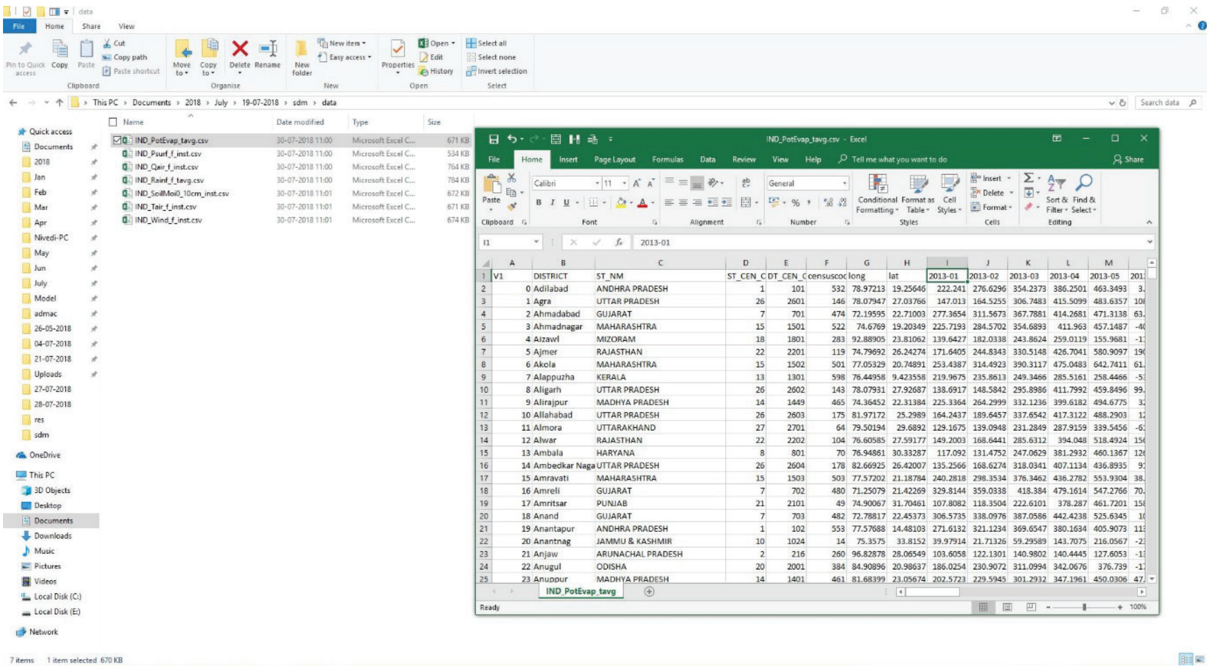

```
#GLDAS data extraction code
library(ncdf4)
library(raster)
library(rgdal)
library(data.table)
library(qdap)

pars=c("PotEvap_tavg","Psurf_f_inst","Qair_f_inst","Rainf_f_tavg","SoilMoi0_10cm_inst","Tair_f_inst","Wind_f_inst")
files_nc=list.files(path="E:/NC4/GLDAS_NOAH025_M.2.1/LSM/",pattern=glob2rx("*.nc4"),full.names = T,recursive = T) #specify the directory where the GLDAS dataset is downloaded.
files_nc=grep(paste0(2013:2017,collapse = "|"),files_nc,value = T) # To filter data between particular years
k=1
fs="IND_dist.csv" # input file containing  lat long
ss<-fread(fs[k],header=T,check.names=F,data.table = F)
```

```
#Change column names of geo-coordinates to long,lat
# names(ss)[6]="long"
# names(ss)[7]="lat"
for (j in 1:length(pars)) {
  for (k  in 1:length(fs)) {
    x<-ss$lat
    y<-ss$long
    data<-data.frame(y,x)
    df_total<-data.frame(c(1:(nrow(data)+1)),stringsAsFactors = F)
    for(i in 1:length(files_nc))
    {
          mydata<- raster(files_nc[i],varname=pars[j])
      crs(mydata)    <-    "+proj=longlat    +datum=WGS84    +no_defs    +ellps=WGS84
+towgs84=0,0,0"
      ff<-extract(mydata,data)
      ff[which(ff=="-9999")]=""
      if(i==6)
      {
        ff= as.numeric(ff) -273.15
      }
      dd<- substr(basename(files_nc[i]),18,23)
      dd=paste0(substr(dd,1,4),"-",substr(dd,5,6))
      ff<-c(as.character(dd),ff)
      df_total<-cbind(df_total,ff)
    }
    v=as.character.Date(data.frame(df_total[1,]))
    v=substr(v,nchar(v)-6,nchar(v))
    v=gsub("\\.","-",v)
    df_total= setNames(df_total,c(v))
    df_total=df_total[2:nrow(df_total),2:ncol(df_total)]
    final_df=data.frame(stringsAsFactors = F)
    final_df=cbind(ss,df_total)
```

```
    filename= paste0(beg2char(basename(fs[k]),"_"),"_",pars[j],".csv",sep="")
    fwrite(final_df,filename,row.names=F,col.names=T,sep=",")
  }
}
```

The input file containing latitude and longitude were read and corresponding parameter values were extracted from the downloaded datasets.

5. Preview of output .csv/xlsx files containing remote environmental values.



V. Generation of maps using R

Introduction

A map is a graphic representation or scale model of spatial concepts. It is a means for conveying geographic information. Mapping of disease events is one of the best method for better visualization for exploring the complex structure of data. Data visualization is not only creating interest and also attract the attention of viewer and provide the way for discovering the pattern. Disease mapping is one of the tools of geographical epidemiology, fulfilling the need to generate accurate and precise maps of disease events. For Example, dot or dot density maps are used display point data, while areal data were presented by Choropleth (Intensity) maps and for continuous surface data. In the veterinary epidemiology, the presentation of maps is established as a basic tool for analysis and interpretation.

Maps are useful visual tools, from displaying sample sites to performing spatial analyses,

i. Shape files:

A shapefile stores geo-spatial data and attribute information related to the dataset. Shapefiles can support points, lines, and polygon features. Areas are represented as closed shape, double-digitized polygons.

An ESRI shapefile consists of a main file (.shp), an index file (.shx), and a dBase (.dbf) table. The main file is a variable-record-length file each record representing a geometry shape with a list of its vertices. Index file record contains the offset which indicates beginning of the record in main file. The dBase has attributes with one record per feature. The order of records in the dBase must same as records in .shp file.

- Main file: 2011_Dist.shp
- Index file: 2011_Dist.shx
- dBase table: 2011_Dist.dbf

ii. Link to download the shape files

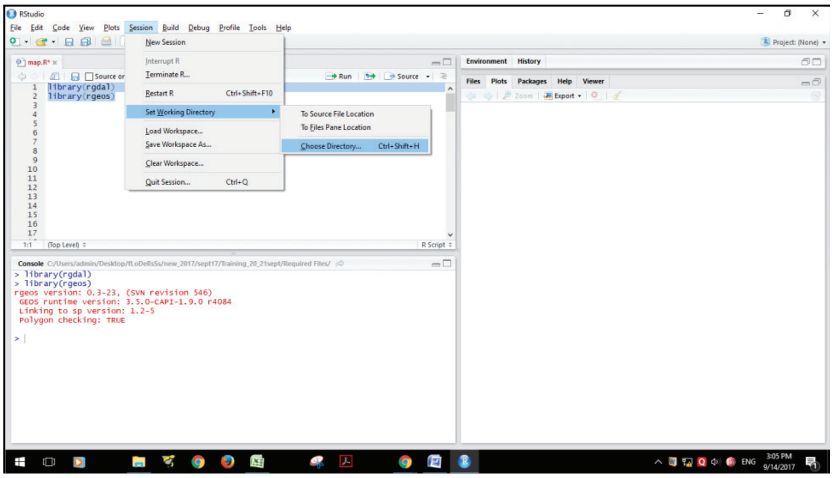
https://drive.google.com/file/d/1RoyV52W08weGPHp-ZEa0nfETn4GYrjhU/view?usp=sharing

Generating Point maps when input data contains Geo coordinates.

1. Install packages (rgdal) and load library

```
install.packages("rgdal") #Don't run  if already installed
library(rgdal)
```

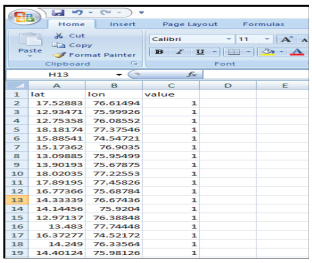
2. Set directory(mention the path of the folder) or go to session in the menu bar and click on set working directory and select choose directory.



3. Read the file containing lat long and output data

```
df=read.csv("latlong.csv")
#filename containing geo-coordinates
ka=readOGR("shp/2011_Dist.shp")
 #Shapefile of India (District level)
```

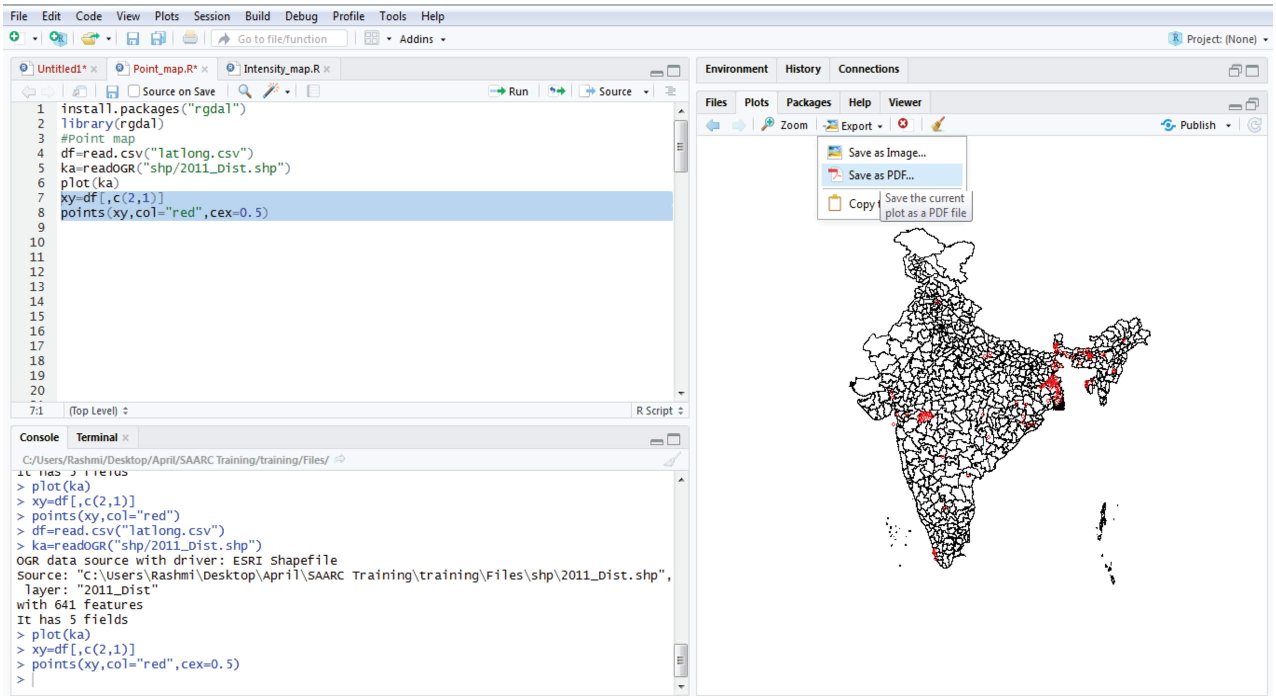
Data Format



4. R code for plotting the point map

```
plot(ka, main= "Name of the map") #plots the outline of the map, Main =“the title of the map”
xy=df[,c(2,1)]
points(xy,col="red",cex=0.5)Col represents the colourof the dots displaying on map, cex=“size of the dot”
```

5. Exporting the map in an image or PDF format.



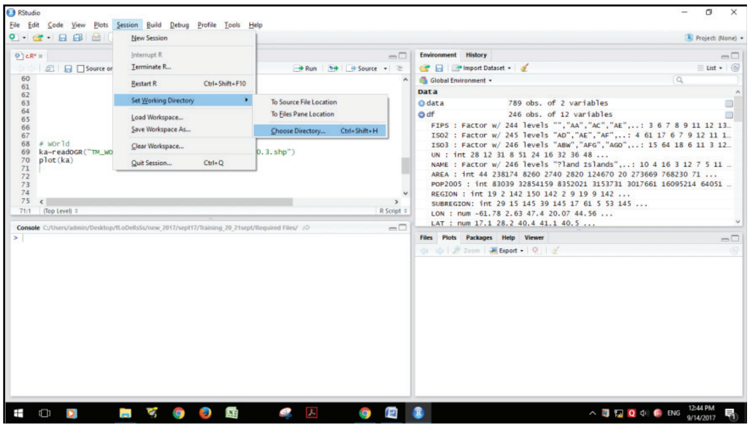
Intensity map

iii. Intensity World map

1. Install and load packages rgdal, plyr and dplyr

```
install.packages(c("plyr","dplyr","rgdal")) #Don't run if already installed
library(rgdal)
library(plyr)
library(dplyr)
```

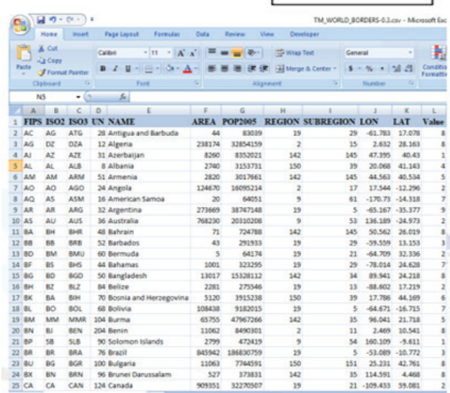
2. Set directory (mention the path of the folder) by clicking on **Session** in the menu bar and on **set working directory** and select **choose directory**. Now choose the folder containing input files. The output files will be saved in the selected folder



3. Feed the shape file and input file.

```
ka=readOGR("TM_WORLD_BORDERS-.3/TM_WORLD_BORDERS-0.3.shp")
# read the world shape file
plot(ka) #Plots the outline of the world map
df=read.csv("TM_WORLD_BORDERS-0.3.csv")
#Read the .csv file containing outbreak data
```

Data format



4. R code for world map

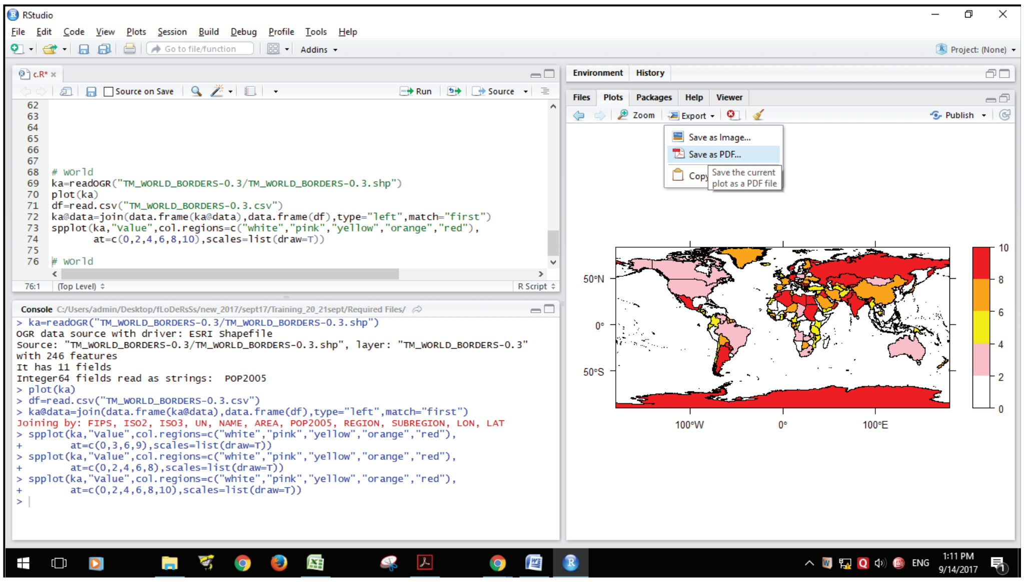
ka@data=join(data.frame(ka@data),data.frame(df),type="left",match="first")

#Joining the shape file data with input data

spplot(ka,main="**World Intensity Map**", "**Value**",col.regions=c("white","pink", "yellow","orange","red") ,at=c(0,2,4,6,8,10),scales=list(draw=T))

*#Plotting the map with colouring the output data, **Main** =“the title of the map”, **Value** indicates the column name of data to be presented on the map, **Col.regions** will represent the colours for the range of values, **scale** indicates lat long of the world map, **at** divides the output data into 5 intervals.*

5. Export the plotted map (Save as image or PDF)

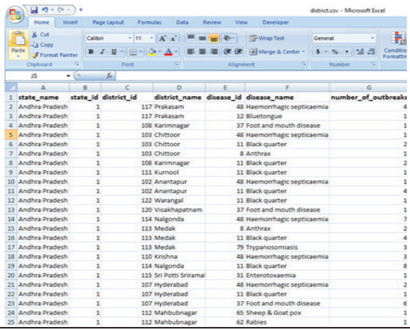


iv. Intensity map- State wise India Map

1. Install and load packages rgdal, plyr and dplyr
2. Set directory
3. Feed the shape file and input file.

```
ka=readOGR("state/state.shp")
# read the State wise India shape file
plot(ka) #Plots the outline of the map
df=read.csv("district.csv") #Read the .csv file containing outbreak data
```

Data Format

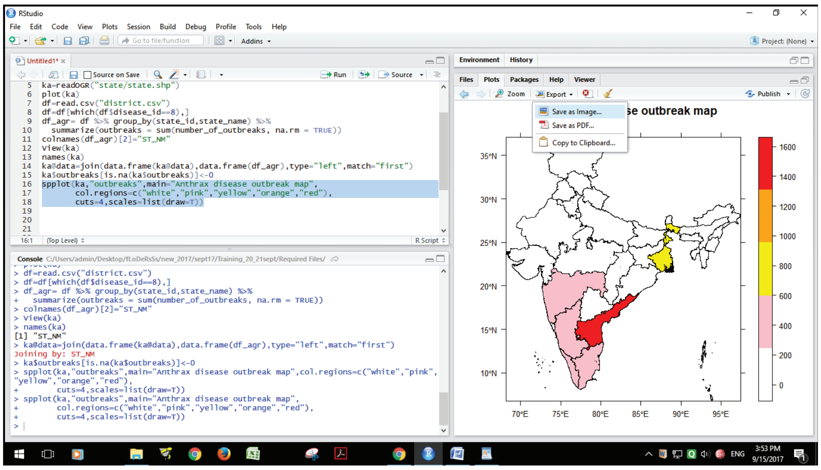


	state_name	state_id	district_id	district_name	disease_id	disease_name	number_of_outbreaks
2	Andhra Pradesh	1	117	Prakasam	48	Haemorrhagic septicaemia	4
3	Andhra Pradesh	1	117	Prakasam	12	Bluetongue	1
4	Andhra Pradesh	1	108	Karimnagar	37	Foot and mouth disease	1
5	Andhra Pradesh	1	103	Chittoor	48	Haemorrhagic septicaemia	1
6	Andhra Pradesh	1	103	Chittoor	11	Black quarter	2
7	Andhra Pradesh	1	103	Chittoor	8	Anthrax	1
8	Andhra Pradesh	1	108	Karimnagar	11	Black quarter	2
9	Andhra Pradesh	1	111	Kurnool	11	Black quarter	1
10	Andhra Pradesh	1	102	Anantapur	48	Haemorrhagic septicaemia	1
11	Andhra Pradesh	1	102	Anantapur	11	Black quarter	4
12	Andhra Pradesh	1	122	Warangal	11	Black quarter	1
13	Andhra Pradesh	1	120	Visakhapatnam	37	Foot and mouth disease	1
14	Andhra Pradesh	1	114	Nalgonda	48	Haemorrhagic septicaemia	7
15	Andhra Pradesh	1	113	Medak	8	Anthrax	2
16	Andhra Pradesh	1	113	Medak	11	Black quarter	4
17	Andhra Pradesh	1	113	Medak	79	Trypanosomiasis	3
18	Andhra Pradesh	1	110	Krishna	48	Haemorrhagic septicaemia	3
19	Andhra Pradesh	1	114	Nalgonda	11	Black quarter	8
20	Andhra Pradesh	1	115	Sri Potti Sriramal	31	Enterotoxaemia	1
21	Andhra Pradesh	1	107	Hyderabad	48	Haemorrhagic septicaemia	2
22	Andhra Pradesh	1	107	Hyderabad	11	Black quarter	1
23	Andhra Pradesh	1	107	Hyderabad	37	Foot and mouth disease	6
24	Andhra Pradesh	1	112	Mahbubnagar	65	Sheep & Goat pox	1
25	Andhra Pradesh	1	112	Mahbubnagar	62	Rabies	1

4. R code for State wise India map

```
df=df[which(df$disease_id==8),] #specify the disease id
df_agr= df %>% group_by(state_id,state_name) %>% summarize(outbreaks = sum(number_of_outbreaks, na.rm = TRUE)) #to extract the disease outbreak data and aggregate it to the state level
colnames(df_agr)[2]="ST_NM" # to change the column names of state as present in the shape file.
View(ka@data) #To view the column names of the shape file data
names(ka)
ka@data=join(data.frame(ka@data),data.frame(df_agr),type="left",match="first")
ka$outbreaks[is.na(ka$outbreaks)]<-0 # To join the shape file and outbreak data, and to set 0 at the missing data
spplot(ka,"outbreaks",main="Anthrax disease outbreak map", col.regions=c("white","pink","yellow","orange","red"),cuts=4, scales=list(draw=T)) # to Plot the disease outbreak data map. Main =“the title of the map”, outbreaks indicates the column name of data to be presented on the map, cuts divides the output data into intervals.if cuts=4, five colours are to be entered.
colors() #colors can be chosen using the color names available.
```

5. Export the plotted map (Save as image or PDF)



v. Intensity map- District wise India Map

1 Install and load packages rgdal, plyr and dplyr

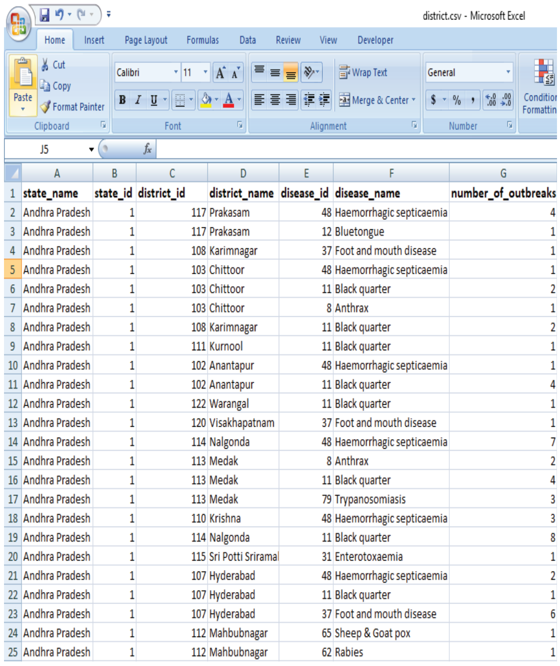
2 Set directory

3 Feed the shape file and input file.

4 R code for District wise India map & extracting the state for separate state map

```
ka=readOGR("District_shp/2011_Dist.shp")
# Read the district level India map shape file
plot(ka) # plot the map
View(ka@data)
# extracting state, district wise map#
ka=ka[which(ka$ST_CEN_CD==15),]
#specify the state code for separate state map
plot(ka) #plots the state map
# extracting state, district wise map#
df=read.csv("district.csv") # Read the .csv file containing outbreak data(all india/ specific state)
```

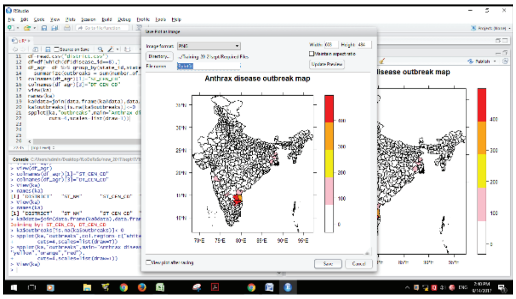
Data Format



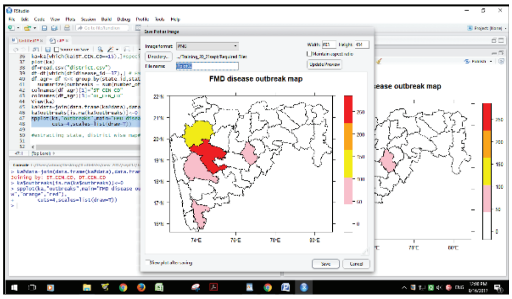
state_name	state_id	district_id	district_name	disease_id	disease_name	number_of_outbreaks
Andhra Pradesh	1	117	Prakasam	48	Haemorrhagic septicaemia	4
Andhra Pradesh	1	117	Prakasam	12	Bluetongue	1
Andhra Pradesh	1	108	Karimnagar	37	Foot and mouth disease	1
Andhra Pradesh	1	103	Chittoor	48	Haemorrhagic septicaemia	1
Andhra Pradesh	1	103	Chittoor	11	Black quarter	2
Andhra Pradesh	1	103	Chittoor	8	Anthrax	1
Andhra Pradesh	1	108	Karimnagar	11	Black quarter	2
Andhra Pradesh	1	111	Kurnool	11	Black quarter	1
Andhra Pradesh	1	102	Anantapur	48	Haemorrhagic septicaemia	1
Andhra Pradesh	1	102	Anantapur	11	Black quarter	4
Andhra Pradesh	1	122	Warangal	11	Black quarter	1
Andhra Pradesh	1	120	Visakhapatnam	37	Foot and mouth disease	1
Andhra Pradesh	1	114	Nalgonda	48	Haemorrhagic septicaemia	7
Andhra Pradesh	1	113	Medak	8	Anthrax	2
Andhra Pradesh	1	113	Medak	11	Black quarter	4
Andhra Pradesh	1	113	Medak	79	Trypanosomiasis	3
Andhra Pradesh	1	110	Krishna	48	Haemorrhagic septicaemia	3
Andhra Pradesh	1	114	Nalgonda	11	Black quarter	8
Andhra Pradesh	1	115	Sri Potti Sriramal	31	Enterotoxaemia	1
Andhra Pradesh	1	107	Hyderabad	48	Haemorrhagic septicaemia	2
Andhra Pradesh	1	107	Hyderabad	11	Black quarter	1
Andhra Pradesh	1	107	Hyderabad	37	Foot and mouth disease	6
Andhra Pradesh	1	112	Mahbubnagar	65	Sheep & Goat pox	1
Andhra Pradesh	1	112	Mahbubnagar	62	Rabies	1

```
df=df[which(df$disease_id==8),] #Extract the disease outbreak data by specifying the disease id
df_agr= df %>% group_by(state_id,state_name,district_id,district_name) %>%
  summarize(outbreaks = sum(number_of_outbreaks, na.rm = TRUE)) #aggregate it to the district level.
colnames(df_agr)[1]="ST_CEN_CD"
colnames(df_agr)[3]="DT_CEN_CD" #Change the column names of state and district as present in the shape file.
View(ka@data) #check and view the column names of the shape file data
names(ka)
ka@data=join(data.frame(ka@data),data.frame(df_agr),type="left",match="first")
ka$outbreaks[is.na(ka$outbreaks)]<-0 # To join the shape file and outbreak data, and to set 0 at the missing data
spplot(ka,"outbreaks",main="Anthrax disease outbreak map",
col.regions=c("white","pink","yellow","orange","red"),cuts=4,scales=list(draw=T))
# Plot the disease outbreak data map Main =“the title of the map”,
```

5 Export the plotted map (Save as image or PDF)



District wise Intensity map of India



District wise Intensity map of Maharashtra State

VI. Generation of Risk Maps

Risk map

A risk map is a data visualization tool for communicating level of risk for specific disease. A risk map is used to assist in identifying, prioritizing, and quantifying risks associated with the disease occurrence.

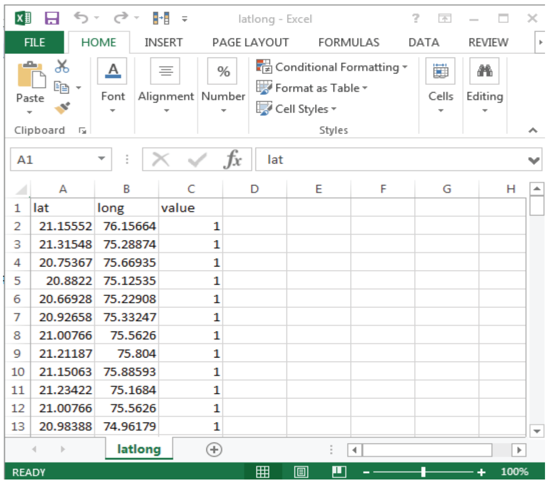
Protocol to Generate risk maps

Disease Climate Modelling also known as Species distributions models (SDMs), bioclimatic envelope models, ecological niche models and habitat suitability models, explore the relationship between geographical occurrences of species or disease occurrence and corresponding environmental variables.

i. Data preparation

- The livestock disease outbreak data has to be collected for a particular place and time period. The geo coordinates of the outbreak locations can be obtained by Google API.

Data format



lat	long	value
21.15552	76.15664	1
21.31548	75.28874	1
20.75367	75.66935	1
20.8822	75.12535	1
20.66928	75.22908	1
20.92658	75.33247	1
21.00766	75.5626	1
21.21187	75.804	1
21.15063	75.88593	1
21.23422	75.1684	1
21.00766	75.5626	1
20.98388	74.96179	1

- The district wise remote sensing data obtained using the R code mentioned under heading III B is merged with the geo coordinates of Indian districts. (code given below)

```
fl=c("IND_NDVI.csv","IND_LST.csv") # filenames containing generated NDVI and LST values

fdf=read.csv("IND_dist.csv") #read input file containing geo-coordinates
```

```
for (i in 1:length(fl)) {
  d=read.csv(fl[i])
  fdf=cbind(fdf,d)
}
write.csv(fdf,"Model_IND_pars.csv",row.names = F) #filename used for modelling
```

ii. Disease climate modelling

Geo coordinates of livestock disease outbreak are compiled. The values of environmental and remote sensing variables were extracted. The extracted valuesand outbreak data are subjected to disease climate modelling. The model estimates similarity of disease occurrencewith the climate providing the predicted risk maps, which can be used as the basis for resources allocation and provide control measures.The different bio modelling data models were used to predict the risk probability across the region.

R code for Disease-climate modelling

```
library(biomod2)
library(randomForest)
library(raster)
library(e1071)
library(rgdal)
library(plyr)

set.seed(4)
s=readOGR("shp/2011_Dist.shp") #read India district level shapefile contained in shp directory
d <- read.csv("Model_IND_pars.csv",sep=",",header=T) #read csv file containing remote sensing parameters along with state and district names
```

```
colnames(d)[c(2,4)]=c("ST_NM","DISTRICT") #rename columns of state and district so as to match with shapefile column names

s@data<-join(data.frame(s@data), data.frame(d), by =c("ST_NM","DISTRICT"), type = "left", match = "first")  # join the shapefile data frame and remote sensing data using similar columns.
View(s@data) #Check whether data got merged or not by viewing shapefile data.

r <- raster(extent(s)) #create empty rectangular raster object of shapefile
projection(r) <- proj4string(s) #set projection system of raster to that of shapefile
res(r)=0.01745 #set resolution of raster object
cols=names(d) # obtain column names of data frame d containing remote sensing data.
cols=cols[5:ncol(d)] # obtain only parameter column names from it.
# convert each remote sensing data stored in shapefile to raster object
 for(i in 1:length(cols))
 {
  resize <- rasterize(s, field=cols[i], r) #Fill empty raster object with parameter values
  filename= paste(cols[i],".tif",sep="") # filename for raster object
  writeRaster(resize, filename, format="GTiff",overwrite=T) #write raster to a tif file
 }
#create a formula object of dependent and independent variables.
 formula=paste(cols,collapse = "+",sep="")
 formula=paste("pb ~",formula)
 formula=as.formula(formula)

 DataSpecies <- read.csv("latlong.csv",sep=",",header=T) # read outbreak data file containing lat,long
 DataSpecies=na.omit(DataSpecies) #Exclude blank values
 myRespName <- "AI" # Set a response name
 myRespXY <- DataSpecies[c("long","lat")] # store geo coordinates in a variable
 myRespXY=na.omit(myRespXY) # Exclude blank values
```

```
myResp<-rep(1,nrow(myRespXY)) # store 1 equal to number of rows in myRespXY to indicate presence of outbreak
raster_data<-list.files(path = ".",pattern=glob2rx("*.tif$"),full.names = T) #obtain path of tif files used for modelling
myExpl<-stack(raster_data) # store it in a single variable
plot(myExpl) # plot the stack data

myBiomodData <- BIOMOD_FormatingData(resp.var = myResp,expl.var = myExpl,resp.xy=myRespXY,resp.name=myRespName,PA.nb.rep= 2,PA.nb.absences =200,PA.strategy = "random") #BIOMOD_FormatingData generates 200 random geo coordinate points within India, # number of random points can be adjusted
plot(myBiomodData)
coor<-myBiomodData@coord #obtain random geo coordinate points
temp<-length(myResp)+1 #obtain length of myResp variable
presvals <- extract(myExpl,myRespXY) # Extract parameter values for outbreak geo coordinates
lat<-coor$lat[temp:nrow(coor)] #store latitude of randomly generated points
lon<-coor$lon[temp:nrow(coor)] #store longitude of randomly generated points
latlon<-cbind(lat,lon) # column bind lat, long objects
backgr <- latlon # store it in variable
absvals <- extract(myExpl, backgr[,c(2,1)]) # Extract parameter values for randomly generated geo coordinates
pb <- c(rep(1, nrow(presvals)), rep(0, nrow(absvals))) # repeat 1 equal to rows of presvals dataframe and repeat 0 equal to rows of absvals dataframe
sdmdata <- data.frame(cbind(coor,pb, rbind(presvals, absvals))) #combine randomly generated geo coordinates, binary data and presence absence data frame
count_one<-nrow(sdmdata[pb==1,]) #obtain number of rows where pb is 1.
count_one1<-count_one+1 #increment value
nr<-nrow(sdmdata) #obtain number of rows in sdmdata
cut_abs<-sdmdata[count_one1:nr,] # Extract rows that contain 0 in pb.
```

```
ss<-replicate(10,sample(nrow(cut_abs),500,replace=T)) # create random 500 values using row numbers of cut_abs variable
pp<-round(rowMeans(ss),0) #Make it integer
gg<-cut_abs[pp,] #Extract the rows
pre_abs<-rbind(sdmdata[1:count_one,],gg) #bind rows of data frames.
pre_abs<-data.frame(impute(pre_abs)) #Fill empty values.

#GLM Model
m1 <- glm( formula,family="binomial", data=pre_abs) # provide formula object and data containing geo coordinates of presence and absence locations along with parameter values
p1 <- predict(myExpl, m1,type="response") #predict the probability values for geo coordinates of raster file.
plot(p1) # plot the predicted raster object
glmresult<- extract(p1, coordinates(s)) # extract probability values for all districts of India.

rf2 <- randomForest(formula, data=pre_abs,ntree=500,mtry=1,importance=T) # provide formula, presence absence data, number of trees to grow for classification, mtry is square root of number of variables used in model.
pr1 <-predict(myExpl, rf2) #predict the probability values for geo coordinates of raster file.
plot(pr1) # plot the predicted raster data
rfresult<- extract(pr1,coordinates(s)) # extract probability values for all districts of India.
res=cbind(s@data,glmresult,rfresult) # combine shapefile data and model results
write.csv(res,"Results.csv") # write results to the csv file, Change the outputfile name
```

Note:

Red colour: Change the file name

Purple Colour: comment explaining the changes to make in that line (Red colour)

Blue colour: comment explaining the code line execution

ICAR - National Institute of Veterinary Epidemiology and Disease Informatics,

Post Box No. 6450, Ramagondanahalli, Yelahanka, Bengaluru, Karnataka 560064

(An ISO 9001:2015 Certified Institute)