Modeling Wheat Yields in India: Describing & Exploiting Spatio-Temporal Variability with Panel Regression

# Introduction

The ability to monitor and predict crop yields in developing countries is critical to the successful adaptation to changes in our climate. Increased temperatures and variability has already been linked to losses in maize and wheat yields (-3.8 and 5.5% respectively)and crop prices globally (Lobell, Schlenker, and Costa-Roberts 2011). Although much effort has been placed on modeling the spatial distribution of these shifts, less effort has been placed on how yields vary across space and time (Ray et al. 2015). Advances in remote sensing provide new avenues to monitor agricultural crop health at high spatial and temporal resolution. However, our ability to monitor changes in plant productivity is still limited in the more complex environments common to many developing countries (Mann and Warner 2015).

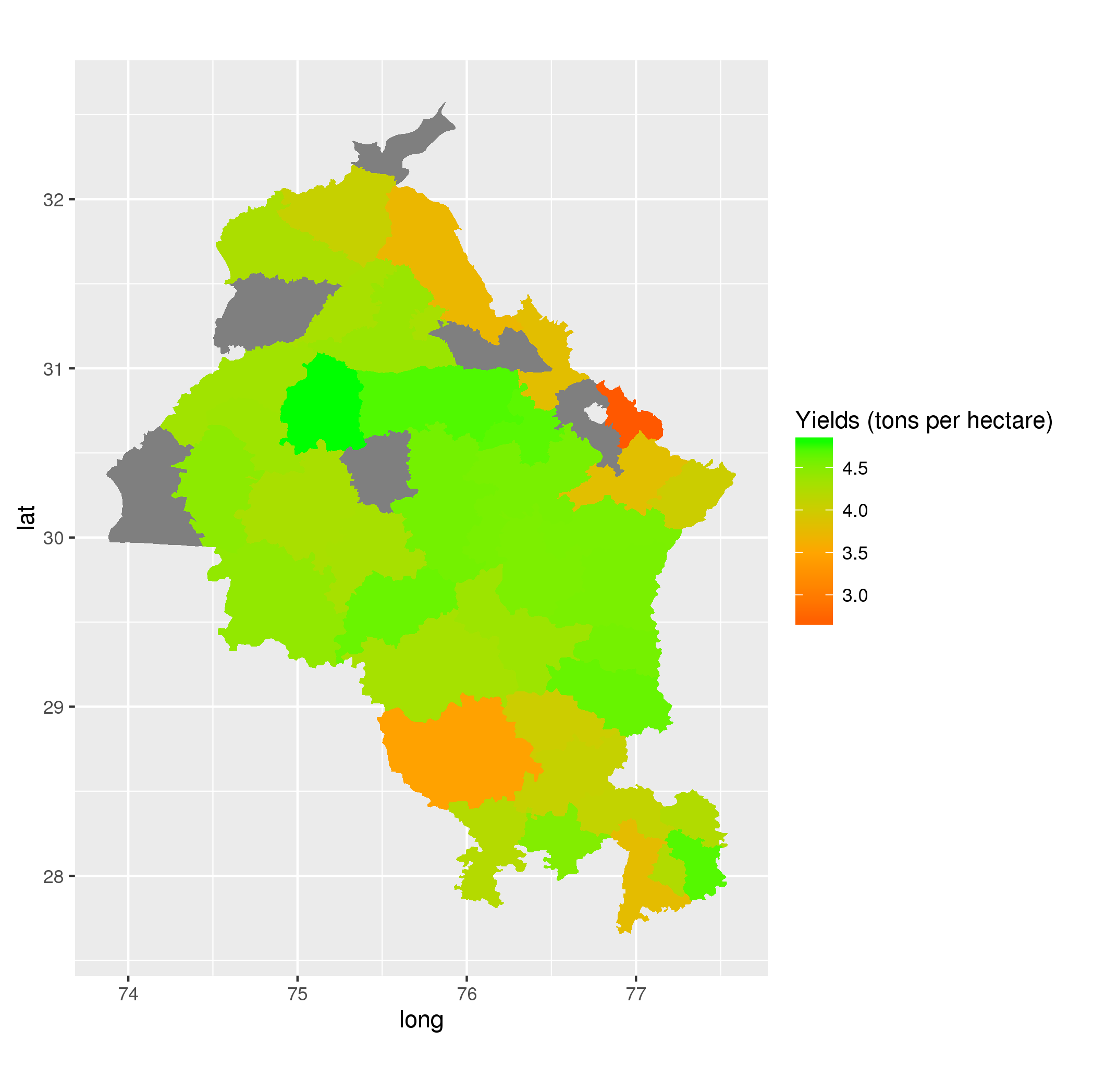
One primary thrust of these efforts has evolved out of the index insurance space. Index insurance is...

The main objectives of this project is to apply and compare statistical methods commonly applied these problems outside of the field of geography. In particular we will focus on the application of spatial panel regression to monitor spatio-temporal variability in wheat yields for Punjab and Haryana India.

# Methods

## Overview and Study Area

We examine wheat yields at the district level for Punjab and Haryana India for Rabi season (roughly Nov-Apr) for the period of 2002 to 2012. Both Punjab and Haryana are extensively cropped but is comprised of a large number of smaller heterogeneous plots. Both states are also extensively double-cropped with rice planting in the Kharif season (roughly May-Oct) and Wheat planted in the Rabi season. Rabi season wheat yield range from 0.33 to 7.46 metric tons per hectare (Table 1, Figure 1).

*Figure 1: Mean Rabi Season Wheat Yields Metric Tons per Hectare by District*  

Here we develop a (non)spatial panel regression model to estimate wheat output per hectare using the open-source programming language R. This model utilizes historical data on plant phenology statistics obtained from the Moderate Resolution Imaging Spectroradiometer (MODIS) satellite. The objective is to develop a handful of metrics that can be used to accurate predict inter-annual variability in wheat yields at the district level.

## Data

The full model is comprised of 25 indicators of plant phenology. District level statistics are then generated from pixel level plant indicators.

### Focus Group Interviews

To help better characterize the physical properties and identify challenges, field visits and focus group interviews were conducted in the winter of 2015. These interviews were conducted in 12 villages with 71 participants in Haryana and Pubjab states, in person with International Food Policy Research Institute (IFPRI) staff. Questions focused on farm characteristics, adopted technologies, Rabi crop calendar dates, and identifying the timing of risks to crops.

### Remote Sensing Data

Considering the relatively small scale of agriculture in this region (median plot sizes of 13.5 acres, range of 2 to 17.2 acres) reported during focus groups (Robles, Ceballos, and Kramer 2015), we utilized 250m vegetation products from the MODIS satellites. Vegetation indices are obtained from two 16 days MODIS products (MOD13Q1, MYD13Q1) from the Aqua and Terra satellites (Didan and Huete 2006). Due to the staggered nature of acquisition these products are treated as partially overlapping windows representing 8 days periods (Doraiswamy, Stern, and Akhmedov 2007).

In particular we examine both the predictive power of the Enhanced Vegetation Index (EVI) and the Normalized Difference Vegetation Index (NDVI) **EVI AND NDVI**. Both are sensitive to the amount of chlorophyll in any given pixel and are commonly used to estimate plant productivity and health in agricultural applications (Mann and Warner 2015). After removal of snow, cloud and other flagged low quality cells, we remove all non-agricultural cells through the use of the 500m MODIS land cover product (MCD12Q1) for the appropriate year (Friedl et al. 2010). The difference in resolutions is expected to have a minimal effect in this case because the extent of rural agriculture in these areas is extremely large. Therefore any cells include or excluded by omission or commission as agriculture should have a minimal effect at the district level.

In addition to cloud cover MODIS data products suffer from four additional sources of error including atmospheric interference, georeferencing, bidirectional reflectance effect and differences in day of the year each pixel is observed (Doraiswamy, Stern, and Akhmedov 2007). While indices such as NDVI minimize the effects of atmospheric distortion but will directly influence indices values. To minimize the effects of the artifacts described above we test the use of temporal smoothing splines and outlier removal (Hastie and Tibshirani 1990) **REFERENCE JOSHUA GRAY**. A visual example of the effects of the cubic smoothing and outlier removal procedure can be seen in Figure 2.

*Figure 2: (Un)smoothed 8-Day NDVI time signature for a dual cropped pixel in Punjab*   *Time series of NDVI unsmoothed (red) and smoothed with outlier removal (blue) for both crop seasons of 2002 to 2016. Wheat growing season highlighted in dark grey, rice growing season in light grey. Date of estimated Rabi season maximum (red vertical line), estimated greenness onset (green vertical line), estimated harvest date (purple veritcal line).*

To test for model improvements obtained through time series cubic smoothing we iteratively run a regression where the model described below (estimating wheat yields) with varying degrees of smoothing of inputs. For each iteration the degree of smoothing is increased and the adjusted and root mean squared error divided by mean yields is reported.

### Agricultural Survey Data

Agricultural survey data at the district level was obtained from \_\_\_\_\_\_\_\_\_. INCLUDE DESCRIPTION OF SWAPPING PUNJAB STATE DATA.

## Summarizing Time: Summarizing Remotely Sensed Data

One of the primary challenges in utilizing an 8 day time series to estimate annual wheat yields is the mismatch in observations. Properties of the time signature must be obtained to characterize and identify important components of the plant phenology time signature correlated with wheat yields in these agricultural systems. Here we utilize 25 metrics to summarize phenology. These measures take two primary forms: first, growing season statistics, spanning the estimated planting date of wheat (mean DOY:315) until harvest date (mean DOY:128); and second annual statistics, spanning a full calendar year including the summer rains of the Kharif growing season. **SWITCH TO RICE GROWING SEASON STATISTICS?** Two classes of statistics are estimated for these two periods: first, summary statistics (e.g. mean, max, variance) and integration (i.e. integration).

### Growing Season Metrics

Planting and harvest dates of are estimated for each growing season of interest. These dates are estimated through an iterative search algorithm finding the date of the global minimum NDVI **or EVI** value nearest to the *a priori* estimated date. A priori values were obtained from the focus group interviews described above. For wheat sowing dates were reported to typically start in the last week of October, and harvest to begin in the 2nd week of April. For details on this function see 1 in the appendix below. Basic growing season summary statics including minimum, maximum, mean, and standard deviation can be calculated using function 2 in the appendix below.

To estimate the cumulative impact of high or low vegetation indices across as season we calculate a variety of integration metrics. These include area under the curve (AUC) of the growing season, the AUC of the increasing portion of the curve (from estimated planting date to growing season maximum), and the AUC of the declining portion of the curve (from growing season maximum to estimated harvest date). For comparision, these values are calculate using two methods, the first using integration using smoothing splines (appendix formula 3 ) and second using trapeziodal estimation (appendix formula 4 ).

We develop a series of metrics to test if modeled yields could be improved through comparisons to 'ideal' years. This includes calculating the 95th percentile (based on sample quantile where the resulting quantile estimates are approximately median-unbiased regardless of the distribution of x (Hyndman and Fan 1996)) of all NDVI **EVI** values, of maximum values, and of the integral (area under the curve) of NDVI **EVI** values. These use the built in functionality of R's base stat function show formula 5 ) in the appedix.

Additional functions were developed to extract the timing of particular phenomena, for instance the date of the maximum value of NDVI **EVI**. Figure 2 visually demonstrates the ability of this function to estimate the timing on greeness onset (referred to henceforth as planting date), seasonal maximums, and harvest dates. For details on these calculations see formula 6 in the appendix. Mutiway ties are handled by perfering the middle most date or if an even number of ties the left middle most date. Another calculates the average value of NDVI **EVI** for each day of the year, which can be used for graphing anomolies (see formula 7 in the appendix). Finally, some of the above codes have improved performance when run on smoothed time series while removing outliers. For this procedure we use a function developed by Joshua Gray at North Carolina State University (see function 8 in the appendix).

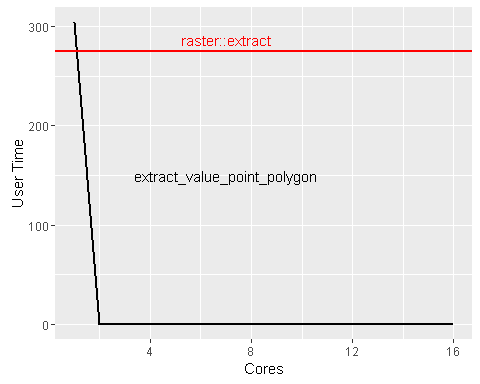
### Annual Metrics

Basic annual summary statics including minimum, maximum, mean, and standard deviation can be calculated using function 9 in the appendix below. Alternatively most functions described above can be used to calculate annual vegetation metrics.

## Summarizing Space & Time: Extraction and Aggregation of Remotely Sensed Data

One major hurdle for this study was the rapid extraction of raster values bases on vector data while maintaining meaningful time stamps. In response we developed the function *extract\_value\_point\_polygon* 10 in the appendix to enhance the performance of the the default raster::extract() function . User processing times were better that 1/6000th that of extract() with the use of a 16-core linux server. Additionally the function can take a list of adjacent rasters to perform data extraction, thereby avoiding vector datasets that span more than the extent of one raster.

*Figure 3: Benchmark test comparing raster::extract() and the new parrallelized extract functions*



## Exploiting Time: (Spatial) Panel Regression Methods and Models

### Panel Regression

|  |  |  |
| --- | --- | --- |
|  | Mean | SD |
| **area** | 153.6 | 86.54 |
| **production\_tonnes** | 665,180 | 396,027 |
| **yield\_tn\_ha** | 4.247 | 0.6061 |
| **plant\_dates** | 314.7 | 9.28 |
| **harvest\_dates** | 128.3 | 27.13 |
| **season\_length** | 178.8 | 28.93 |
| **VEG\_annual\_mean** | 4,570 | 741.7 |
| **VEG\_annual\_min** | 2,369 | 1,117 |
| **VEG\_annual\_max** | 6,978 | 805.1 |
| **VEG\_annual\_AUC** | 1.26e+11 | 3.808e+10 |
| **VEG\_annual\_5th\_prct** | 2,454 | 1,097 |
| **VEG\_annual\_sd** | 1,516 | 479.1 |
| **VEG\_annual\_max\_5th\_prct** | 5,695 | 956.5 |
| **VEG\_annual\_AUC\_5th\_prct** | 2.192e+10 | 2.441e+09 |
| **VEG\_growing\_max\_date** | 43.75 | 21.84 |
| **VEG\_growing\_mean** | 4,523 | 500.8 |
| **VEG\_growing\_min** | 2,185 | 630.6 |
| **VEG\_growing\_max** | 6,826 | 791.4 |
| **VEG\_growing\_AUC** | 101,724 | 20,444 |
| **VEG\_growing\_5th\_prct** | 2,234 | 623.9 |
| **VEG\_growing\_max\_5th\_prct** | 6,309 | 822.8 |
| **VEG\_growing\_AUC\_5th\_prct** | 54,939 | 5,709 |
| **VEG\_growing\_AUC\_v2** | 101,716 | 20,442 |
| **VEG\_growing\_AUC\_leading** | 58,771 | 12,435 |
| **VEG\_growing\_AUC\_trailing** | 45,635 | 10,875 |
| **VEG\_growing\_AUC\_diff\_mn** | 17.47 | 16,825 |
| **VEG\_growing\_AUC\_diff\_90th** | -13,607 | 17,147 |
| **VEG\_all\_growing\_5th\_prct** | 2,149 | 509.3 |
| **VEG\_growing\_sd** | 1,679 | 462 |
| **Whe\_Yeild\_kgha** | 4,308 | 508.6 |
| **yield\_tn\_ha\_dual** | 4.232 | 0.6179 |

### Spatial Panel Regression

#### Defining the neighborhood

# Results

## Focus Group Interviews

## Panel Regression

# Discussion

# Conclusions

# Appendix A

## Yield Data

*Table A1: Rabi Season Wheat Yields Metric Tons per Hectare by State*

## Warning in max(yield\_tn\_ha, na.rm = T): no non-missing arguments to max;  
## returning -Inf

pander(state\_yeilds, justify = c('left', 'left', 'center','center', 'center'))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| State | district | Min | Mean | Max |
| Haryana | AMBALA | 3.24 | 3.817 | 4.86 |
| Haryana | BHIWANI | 0.33 | 3.465 | 4.47 |
| Haryana | FARIDABAD | 3.66 | 4.22 | 4.92 |
| Haryana | FATEHABAD | 4.08 | 4.616 | 5.46 |
| Haryana | GURGAON | 3.52 | 4.094 | 4.92 |
| Haryana | HISAR | 3.7 | 4.301 | 5.1 |
| Haryana | JHAJJAR | 3.59 | 4.067 | 4.89 |
| Haryana | JIND | 3.96 | 4.355 | 5.24 |
| Haryana | KAITHAL | 4.11 | 4.533 | 5.45 |
| Haryana | KARNAL | 4.13 | 4.548 | 5.68 |
| Haryana | KURUKSHETRA | 4.03 | 4.545 | 5.44 |
| Haryana | MAHENDRAGARH | 3.73 | 4.202 | 4.78 |
| Haryana | MEWAT | 3.07 | 3.78 | 4.37 |
| Haryana | PALWAL | 4.15 | 4.688 | 5.09 |
| Haryana | PANCHKULA | 1.88 | 2.625 | 3.57 |
| Haryana | PANIPAT | 4.22 | 4.572 | 5.56 |
| Haryana | REWARI | 4.01 | 4.506 | 4.97 |
| Haryana | ROHTAK | 3.45 | 4.01 | 5 |
| Haryana | SIRSA | 3.43 | 4.425 | 5.36 |
| Haryana | SONIPAT | 4.28 | 4.626 | 5.51 |
| Haryana | YAMUNANAGAR | 3.39 | 4.013 | 5.38 |
| Punjab | AMRITSAR | 4.05 | 4.266 | 4.43 |
| Punjab | BARNALA | Inf | NA | -Inf |
| Punjab | BATHINDA | 3.89 | 4.281 | 4.79 |
| Punjab | FARIDKOT | 4.04 | 4.35 | 4.81 |
| Punjab | FATEHGARH SAHIB | 4.09 | 4.614 | 5.18 |
| Punjab | FAZILKA | Inf | NA | -Inf |
| Punjab | FIROZPUR | 3.98 | 4.332 | 4.92 |
| Punjab | GURDASPUR | 3.57 | 4.056 | 4.48 |
| Punjab | HOSHIARPUR | 3.4 | 3.715 | 4.29 |
| Punjab | JALANDHAR | 4.14 | 4.376 | 4.69 |
| Punjab | KAPURTHALA | 3.97 | 4.283 | 4.64 |
| Punjab | LUDHIANA | 4.39 | 4.7 | 4.96 |
| Punjab | MANSA | 3.75 | 4.293 | 4.88 |
| Punjab | MOGA | 4.14 | 4.81 | 7.46 |
| Punjab | MUKTSAR | 3.94 | 4.45 | 4.98 |
| Punjab | PATHANKOT | Inf | NA | -Inf |
| Punjab | PATIALA | 4.12 | 4.55 | 4.83 |
| Punjab | RUPNAGAR | 3.31 | 3.807 | 4.51 |
| Punjab | SAHIBZADA AJIT SINGH NAGAR | Inf | NA | -Inf |
| Punjab | SANGRUR | 4.23 | 4.574 | 5.13 |
| Punjab | SHAHID BHAGAT SINGH NAGAR | Inf | NA | -Inf |
| Punjab | TARN TARAN | Inf | NA | -Inf |

## Functions

*Functions 1: Planting/Harvest date functions*

PlantHarvestDates = function(start\_date,end\_date,PlantingMonth,PlantingDay,HarvestMonth,HarvestDay){  
 # this function takes in date range and returns planting and harvest date for time series as a data.frame   
 # for all years of interest. Handles growing periods overlaping a new year properly.  
 # NOTE: This is used to create dataframe of planting / harvest dates for many other functions  
 #   
 # e.g. PlantHarvest = PlantHarvestDates('2002-01-01','2016-02-02',PlantingMonth=11, PlantingDay=23,HarvestMonth=4,HarvestDay=30)  
   
 start\_end\_years = c(strptime(start\_date,'%Y-%m-%d'),strptime(end\_date,'%Y-%m-%d'))  
 names(unclass(start\_end\_years[1]))  
 start\_end\_years[1]$mon=PlantingMonth-1  
 start\_end\_years[1]$mday=PlantingDay  
 planting = as.Date(seq(start\_end\_years[1],  
 length=strptime(dates[2],'%Y-%m-%d')$year-strptime(dates[1],'%Y-%m-%d')$year,  
 by='year'))  
 # set harvest  
 start\_end\_years[2]$year=start\_end\_years[1]$year+1 # set year equal to start year +1  
 start\_end\_years[2]$mon=HarvestMonth-1  
 start\_end\_years[2]$mday=HarvestDay  
 harvest = as.Date(seq(start\_end\_years[2],  
 length=strptime(end\_date,'%Y-%m-%d')$year-strptime(start\_date,'%Y-%m-%d')$year,  
 by='year'))  
 return(data.frame(planting=planting,harvest=harvest))  
 }  
  
SearchMinumumBeforeAfterDOY = function(x,dates\_in,DOY\_in,days\_shift,dir){  
 # calculates the global minimum for days before,after,both of expected planting date  
 # best to set DOY as the last expected date of planting  
 # x = vegetation index, dates\_in = dates of observation POSIX, DOY\_in = expected planting or harvest date  
 # days\_shift = # days to search around DOY\_in, dir='before' 'after' 'beforeafter'  
   
 if(days\_shift<=8){print('Using less than 8 days is dangerous, 15-30 stable')}  
   
 # avoid problems with time class  
 if(is.na(DOY\_in[1])){print('ERROR: convert date format to %Y%j');break}  
 if(class(dates\_in)[1]!= 'POSIXlt' ){dates\_in=as.POSIXlt(dates\_in)}  
   
 # limit to fixed # of days before/after DOY  
 DOY\_in = as.POSIXlt(DOY\_in)  
 DOY\_before = DOY\_in  
   
 #names(unclass(DOY\_before[1]))  
 if(dir=='before') DOY\_before$mday=DOY\_before$mday-days\_shift # set days before to doy - days\_before  
 if(dir=='after') DOY\_before$mday=DOY\_before$mday+days\_shift # set days before to doy - days\_before  
 if(dir=='beforeafter'){ DOY\_before$mday=DOY\_before$mday-days\_shift   
 DOY\_in$mday=DOY\_in$mday+days\_shift}  
 DOY\_table = data.frame(DOY\_before=DOY\_before,DOY\_in=DOY\_in) #join start end search dates  
   
 # list all days 'days\_before' DOY\_in  
 if(dir=='before'|dir=='beforeafter'){ DOY\_interest = as.POSIXlt(unlist(lapply(1:dim(DOY\_table)[1],  
 function(h){format(seq(DOY\_table[h,1],  
 DOY\_table[h,2],by='day'),'%Y-%m-%d')})),tz='UTC')}  
 if(dir=='after'){DOY\_interest = as.POSIXlt(unlist(lapply(1:dim(DOY\_table)[1],  
 function(h){format(seq(DOY\_table[h,2],  
 DOY\_table[h,1],by='day'),'%Y-%m-%d')})),tz='UTC')}  
   
 # find all local minima, and match with DOY  
 x\_DOY\_interest = x[dates\_in %in% DOY\_interest]  
 dates\_DOY\_interest = dates\_in[dates\_in %in% DOY\_interest]  
 # get min value for this period for each year  
 sort(AnnualMaxima(x\_DOY\_interest\*-1,as.Date(dates\_DOY\_interest)))  
}

*Functions 2: Flexible growing season vegetation metrics*

PeriodAggregator = function(x,dates\_in,date\_range\_st, date\_range\_end,by\_in='days',FUN){  
 # returns a summary statistic of x for any function FUN, over the period defined by date\_range\_st, date\_range\_end  
 # x = vegetation index data, dates\_in = dates of observation POSIX, dates\_in,date\_range\_st = start end dates of period, FUN = function  
 # E.g. PeriodAggregator(x=plotdatasmoothed$EVI,dates\_in = plotdatasmoothed$dates,date\_range\_st=plotdatasmoothed$dates[1],date\_range\_end=plotdatasmoothed$dates[20], FUN = function(y){mean(y,na.rm=T)})  
 if(class(dates\_in)[1]== "POSIXct"|class(dates\_in)[1]== "POSIXlt" )dates\_in = as.Date(dates\_in)  
 if(class(date\_range\_st)[1]== "POSIXct" ){date\_range\_st = as.Date(date\_range\_st)  
 date\_range\_end = as.Date(date\_range\_end)}  
 #Avoid problems with missing plant or harvest dates  
 if(length(date\_range\_st)!=length(date\_range\_end)){print('number of elements in start end dates dont match'); break}  
 dataout=lapply(1:length(date\_range\_st),function(z){  
 DateRange = seq(date\_range\_st[z],date\_range\_end[z],by=by\_in)  
 x=x[dates\_in %in% DateRange]  
 dates\_in=dates\_in[dates\_in %in% DateRange]  
 FUN(x)})  
 dataout = do.call(c,dataout)  
 names(dataout)=format(date\_range\_st,'%Y')  
 dataout  
 }

*Functions 3: Area under the curve estimation - smoothing splines*

PeriodAUC = function(x\_in,dates\_in,DOY\_start\_in,DOY\_end\_in){  
 # calculate area under the curve by period of the year using spline estimation  
 # x = data, dates\_in=asDate(dates),DOY\_start\_in=asDate(list of start periods),DOY\_end\_in=asDate(list of end per  
 # x = plotdatasmoothed$EVI,dates\_in = plotdatasmoothed$dates , DOY\_start\_in= plant\_dates ,DOY\_end\_in=harvest\_dates)  
 if(class(dates\_in)[1]== "POSIXct"|class(dates\_in)[1]== "POSIXlt" )dates\_in = as.Date(dates\_in)  
 dates\_group = rep(0,length(dates\_in)) # create storage for factors of periods  
 # get sequences of periods of inerest  
 seq\_interest = lapply(1:length(DOY\_start\_in),function(z){seq(DOY\_start\_in[z],DOY\_end\_in[z],by='days')})  
 # switch dates-group to period group  
 years\_avail = sort(as.numeric(unique(unlist(  
 lapply(seq\_interest,function(z) format(z,'%Y'))))))  
 for(z in 1:length(seq\_interest)){ #assigns year for beginging of planting season  
 dates\_group[dates\_in %in% seq\_interest[[z]]]=years\_avail[z]  
 assign('dates\_group',dates\_group,envir = .GlobalEnv) } # assign doesn't work in lapply using for loop instead  
 # calculate AUC for periods of interest  
 FUN = function(q,w){auc(q,w,type='spline')}  
 datesY = format(dates\_in,'%Y')  
 data.split = split(x\_in,dates\_group)  
 d = do.call(c,lapply(2:length(data.split),function(z){ # start at 2 to avoid group=0  
 FUN(q=1:length(data.split[[z]]),w=data.split[[z]]) }))  
 names(d) = names(data.split)[2:length(data.split)]  
 d  
 }

*Functions 4: Area under the curve estimation - trapazoidal estimation*

PeriodAUC\_method2 = function(x\_in,dates\_in,DOY\_start\_in,DOY\_end\_in){  
 #NOTE SPLINE METHOD 1 SEEMS to WORK BETTER  
 # calculate area under the curve by period of the year  
 # x = data, dates\_in=asDate(dates),DOY\_start=asDate(list of start periods),DOY\_end=asDate(list of end per$  
 # x = plotdatasmoothed$EVI,dates\_in = plotdatasmoothed$dates , DOY\_start=annualMinumumBeforeDOY(x = plotd$  
 if(class(dates\_in)[1]== "POSIXct"|class(dates\_in)[1]== "POSIXlt" )dates\_in = as.Date(dates\_in)  
  
 dates\_group = rep(0,length(dates\_in)) # create storage for factors of periods  
 # get sequences of periods of inerest  
 seq\_interest = lapply(1:length(DOY\_start\_in),function(z){seq(DOY\_start\_in[z],DOY\_end\_in[z],by='days')})  
 # switch dates-group to period group  
 years\_avail = sort(as.numeric(unique(unlist(  
 lapply(seq\_interest,function(z) format(z,'%Y'))))))  
 for(z in 1:length(seq\_interest)){ #assigns year for beginging of planting season  
 dates\_group[dates\_in %in% seq\_interest[[z]]]=years\_avail[z]  
 assign('dates\_group',dates\_group,envir = .GlobalEnv) } # assign doesn't work in lapply using for loop instead  
   
 # calculate AUC for periods of interest  
 FUN = function(q,w){ sum(diff(q)\*rollmean(w,2))}  
 datesY = format(dates\_in,'%Y')  
 data.split = split(x\_in,dates\_group)  
 d = do.call(c,lapply(2:length(data.split),function(z){ # start at 2 to avoid group=0  
 FUN(q=1:length(data.split[[z]]),w=data.split[[z]]) }))  
 names(d) = names(data.split)[2:length(data.split)]  
 #print(cbind(names(data.split)[2:length(data.split)], d))  
 d  
 }

*Functions 5: Base function used for estimating sample quantiles*

*Functions 6: Function to return date of any given phenomenon*

*Functions 7: Mean day of the year values*

*Functions 8: Smoothing splines with outlier removal*

# and out\_iterations controls the number of times that outliers are checked and removed w/ subsequent spline refit

*Function 9: Flexible annual vegetation metrics*

AnnualAggregator = function(x,dates\_in,FUN){  
 # returns an annual summary statistic of any function  
 # x = vegetation index data, dates\_in = dates of observation POSIX,  
 # E.g. AnnualAggregator(x= plotdatasmoothed$EVI,dates\_in = plotdatasmoothed$dates, FUN = function(y){mean(y,na.rm=T)})  
 datesY = format(dates\_in,'%Y')  
 do.call(c,lapply(split(x,datesY),FUN))}

*Function 10: Rapid multicore extract raster data by point or polygon*

extract\_value\_point\_polygon = function(point\_or\_polygon, raster\_stack, num\_workers){  
 # Returns list containing values from locations of spatial points or polygons  
 # if polygons are too small reverts to centroid   
 if(class(raster\_stack)!='list'){raster\_stack=list(raster\_stack)}  
 lapply(c('raster','foreach','doParallel'), require, character.only = T)  
 registerDoParallel(num\_workers)  
 ptm <- proc.time()  
 # iterate between points or polygons  
 ply\_result = foreach(j = 1:length(point\_or\_polygon),.inorder=T) %do%{  
 print(paste('Working on feature: ',j,' out of ',length(point\_or\_polygon)))  
 get\_class= class(point\_or\_polygon)[1]  
 # switch rasterstack according to which point or polygon is %over%  
 for(z in 1:length(raster\_stack)){  
 # set raster to use  
 raster\_stack\_use = raster\_stack[[z]]  
 # get cell numbers of point of polygon, repeat if missing  
 if(get\_class=='SpatialPolygons'|get\_class=='SpatialPolygonsDataFrame'){  
 cell = as.numeric(cellFromPolygon(raster\_stack\_use, point\_or\_polygon[j,], weights=F)[[1]])  
 # if polygon is too small to find cells, convert to centroid and get cellfromXY  
 if(length(cell)==0){ #coord(poly) returns centroid  
 cell = as.numeric(na.omit(cellFromXY(raster\_stack\_use, coordinates(point\_or\_polygon[j,]) )))}}  
 if(get\_class=='SpatialPointsDataFrame'|get\_class=='SpatialPoints'){  
 cell = as.numeric(na.omit(cellFromXY(raster\_stack\_use, point\_or\_polygon[j,])))}  
 # if cells found keep raster\_stack\_use = raster\_stack[[z]]  
 if(length(cell)!=0){break}  
 # if cells not found repeat for different stack or return NA  
 if(length(cell)==0 & z!=length(raster\_stack)){next}else{return(NA)}  
 }  
 # create raster mask from cell numbers  
 r = rasterFromCells(raster\_stack\_use, cell,values=F)  
 result = foreach(i = 1:dim(raster\_stack\_use)[3],.packages='raster',.inorder=T) %dopar% {  
 crop(raster\_stack\_use[[i]],r)  
 }  
 result=as.data.frame(getValues(stack(result)))  
 return(result)  
 }  
 print( proc.time() - ptm)  
 endCluster()  
 return(ply\_result)  
 }

# References

Didan, K, and A Huete. 2006. “MODIS Vegetation Index Product Series Collection 5 Change Summary.” Tucson, Arizona.

Doraiswamy, Paul C, Alan J Stern, and Bakhyt Akhmedov. 2007. “Crop Classification in the US Corn Belt Using MODIS Imagery.” In *2007 IEEE International Geoscience and Remote Sensing Symposium*, 809–12. IEEE.

Friedl, Mark A, Damien Sulla-Menashe, Bin Tan, Annemarie Schneider, Navin Ramankutty, Adam Sibley, and Xiaoman Huang. 2010. “MODIS Collection 5 global land cover: Algorithm refinements and characterization of new datasets.” *Remote Sensing of Environment* 114 (1): 168–82. <http://www.sciencedirect.com/science/article/B6V6V-4XFNCW6-1/2/926cb3ceff79aa59819a94aa7fa4d00e>.

Hastie, Trevor J, and Robert J Tibshirani. 1990. *Generalized Additive Models*. Vol. 43. CRC Press.

Hyndman, Rob J, and Yanan Fan. 1996. “Sample Quantiles in Statistical Packages.” *The American Statistician* 50 (4). Taylor & Francis: 361–65.

Lobell, David B., Wolfram Schlenker, and Justin Costa-Roberts. 2011. “Climate Trends and Global Crop Production Since 1980.” *Science* 333 (6042). American Association for the Advancement of Science: 616–20. doi:[10.1126/science.1204531](https://doi.org/10.1126/science.1204531).

Mann, ML L, and J Warner. 2015. “Ethiopian Wheat Yield and Yield Gap Estimation: A Small Area Integrated Data Approach.” Addis Ababa, Ethiopia: International Food Policy Research Institute.

Ray, Deepak K, James S Gerber, Graham K MacDonald, and Paul C West. 2015. “Climate variation explains a third of global crop yield variability.” *Nature Communications* 6 (January). The Author(s): 5989. <http://dx.doi.org/10.1038/ncomms6989 http://10.1038/ncomms6989 http://www.nature.com/articles/ncomms6989{\#}supplementary-information>.

Robles, Miguel, Francisco Ceballos, and Berber Kramer. 2015. “Focus Group Interviews: Haryana and Punjab India.” India: IFPRI.