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).

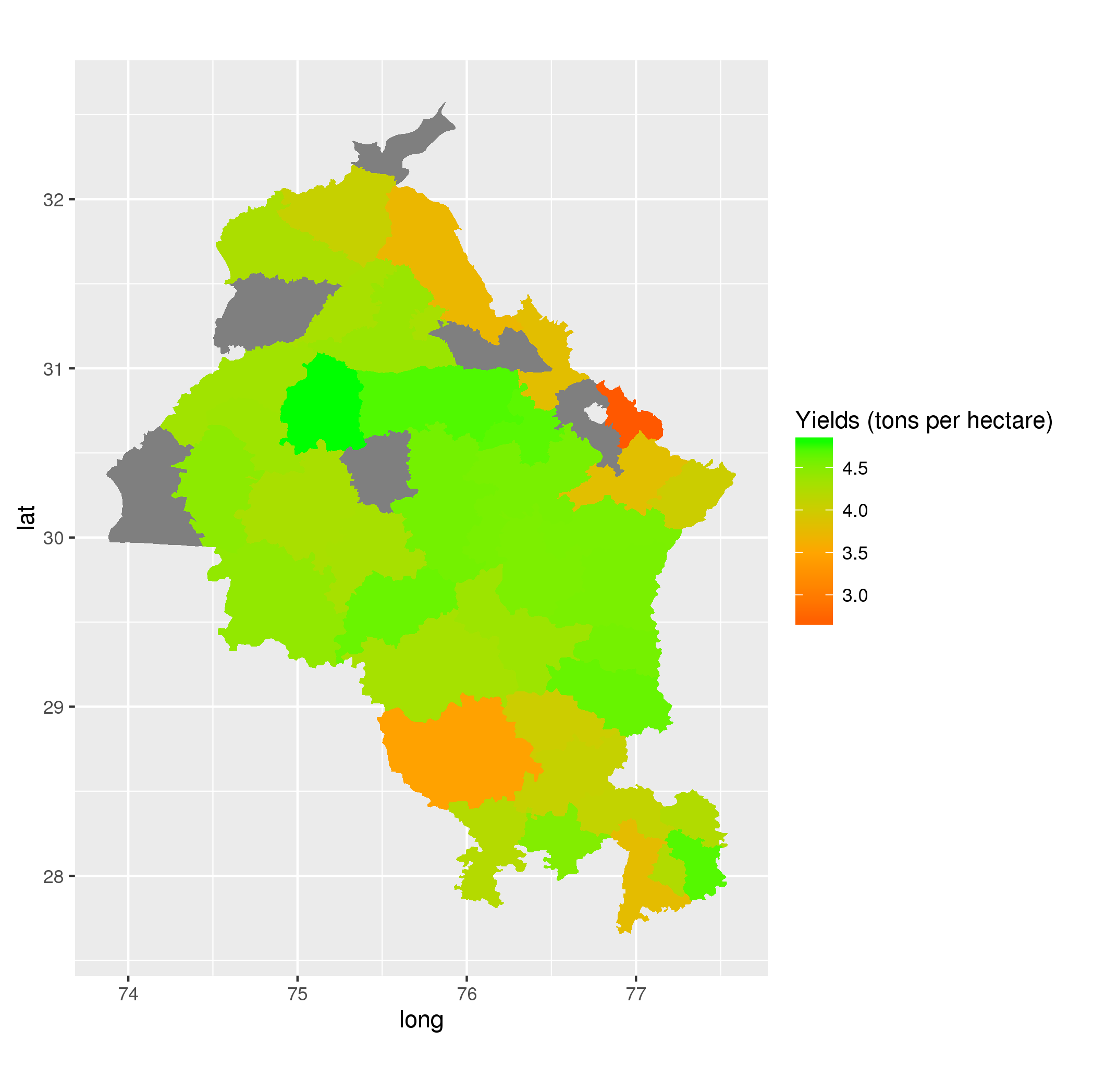
One primary thrust of these efforts has evolved out of the index insurance space. Index insurance is...

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 1.88 to 5.68 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 5 indicators of plant phenology. District level statistics are then generated from pixel level plant indicators.

### Focus Group Interviews

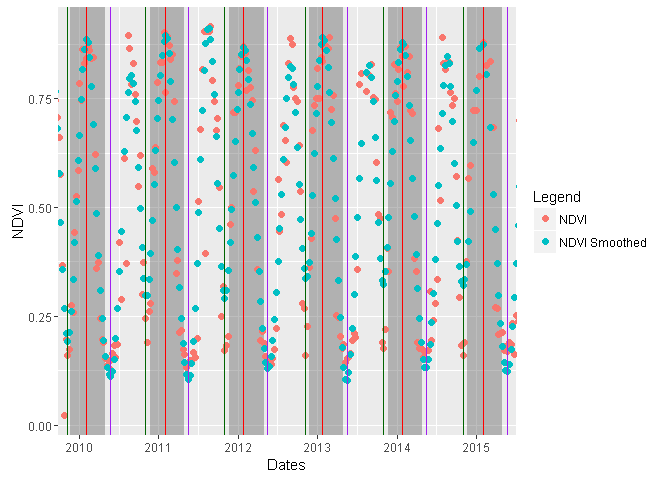
To help better characterize the physical properties and identify challenges, field visits and focus group interviews were conducted in the winter of 2015. These interviews were conducted in 12 villages with 71 participants in Haryana and Pubjab states, in person with International Food Policy Research 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. Moreover agricultural patterns are generally uniform over broad areas of Punjab and Haryana.

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**. Where outliers above three standard deviations are removed before applying a cubic smoothing spline. 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 5 metrics to summarize phenology. These measures take two primary forms: first, growing season statistics, spanning the estimated planting date of wheat (mean DOY:NA) until harvest date (mean DOY:NA); and second monsoon season statistics, spanning end of the Rabi wheat season to end of the Kharif growing season. Two classes of statistics are estimated for these two periods: first, summary statistics (e.g. mean, max, variance) and integration (i.e. integration).

Pixels with in area of interest (AOI), district boundaries in this case, can be evaluated on a pixel by pixel basis in utilizing parrallel processing or summarized by the AOI's mean value for each image. In this study, district level mean values of EVI are used to represent agricultural productivity for each 8 day period.

### 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.

### Variable Definitions

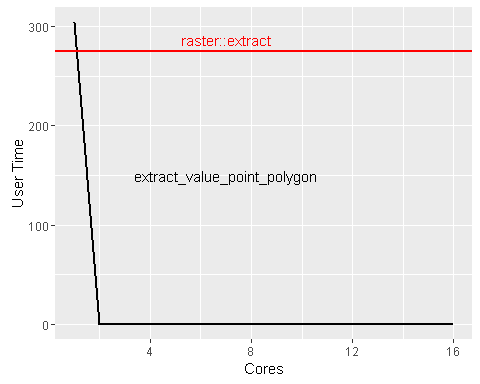
## Name  
## 1 Basic Summary Statistics  
## 2 Rabi(Kharif)\_mean  
## 3 Rabi(Kharif)\_min  
## 4 Rabi(Kharif)\_max  
## 5 Rabi(Kharif)\_sd  
## 6 Rabi(Kharif)\_mean  
## 7 Rabi(Kharif)\_max\_date  
## 8 Rabi(Kharif)\_plant\_date  
## 9 Rabi(Kharif)\_harvest\_date  
## 10 All\_95th\_prct  
## 11 Integrated Summary Statistics  
## 12 Rabi(Kharif)\_AUC  
## 13 Rabi(Kharif)\_AUC\_v2  
## 14 Rabi(Kharif)\_AUC\_leading  
## 15 Rabi(Kharif)\_AUC\_trailing  
## 16 Rabi\_season\_length  
## 17 Comparison to Norms  
## 18 Rabi(Kharif)\_95th\_diff\_mn  
## 19 Rabi(Kharif)\_95th\_diff\_mx  
## 20 Rabi(Kharif)\_95th\_diff\_AUC  
## 21 Rabi(Kharif)\_AUC\_diff\_mn  
## Description  
## 1   
## 2 Rabi(Kharif) mean values of NDVI  
## 3 Rabi(Kharif) min values of NDVI  
## 4 Rabi(Kharif) maximum values of NDVI  
## 5 Rabi(Kharif) standard devIation in values of NDVI  
## 6 Rabi(Kharif) mean values of NDVI  
## 7 Date of Rabi(Kharif) maximum values of NDVI  
## 8 Estimated date of Rabi(Kharif) planting based on NDVI phenology  
## 9 Estimated date of Rabi(Kharif) harvest based on NDVI phenology  
## 10 The estimated 95th percentile values of all seasons in the NDVI historical record  
## 11   
## 12 Rabi(Kharif) area under the curve of NDVI  
## 13 Rabi(Kharif) area under the curve of NDVI estimated by splines  
## 14 Area under the curve of NDVI for the ascending part of the curve during the Rabi(Kharif) season  
## 15 Area under the curve of NDVI for the decreasing part of the curve during the Rabi(Kharif) season  
## 16 Difference between Rabi plant\_dates and harvest\_dates in days  
## 17   
## 18 Difference between Rabi(Kharif) mean NDVI and Rabi(Kharif) estimated 95th percentile of historical values  
## 19 Difference between Rabi(Kharif) max NDVI and Rabi(Kharif) estimated 95th percentile of historical max values  
## 20 Difference between Rabi(Kharif) AUC NDVI and Rabi(Kharif) estimated 95th percentile of historical AUC values  
## 21 Difference between the Rabi(Kharif) mean area under the curve and Rabi(Kharif) area under the curve for NDVI

|  |  |
| --- | --- |
| Name | Description |
| Basic Summary Statistics |  |
| Rabi(Kharif)\_mean | Rabi(Kharif) mean values of NDVI |
| Rabi(Kharif)\_min | Rabi(Kharif) min values of NDVI |
| Rabi(Kharif)\_max | Rabi(Kharif) maximum values of NDVI |
| Rabi(Kharif)\_sd | Rabi(Kharif) standard devIation in values of NDVI |
| Rabi(Kharif)\_mean | Rabi(Kharif) mean values of NDVI |
| Rabi(Kharif)\_max\_date | Date of Rabi(Kharif) maximum values of NDVI |
| Rabi(Kharif)\_plant\_date | Estimated date of Rabi(Kharif) planting based on NDVI phenology |
| Rabi(Kharif)\_harvest\_date | Estimated date of Rabi(Kharif) harvest based on NDVI phenology |
| All\_95th\_prct | The estimated 95th percentile values of all seasons in the NDVI historical record |
| Integrated Summary Statistics |  |
| Rabi(Kharif)\_AUC | Rabi(Kharif) area under the curve of NDVI |
| Rabi(Kharif)\_AUC\_v2 | Rabi(Kharif) area under the curve of NDVI estimated by splines |
| Rabi(Kharif)\_AUC\_leading | Area under the curve of NDVI for the ascending part of the curve during the Rabi(Kharif) season |
| Rabi(Kharif)\_AUC\_trailing | Area under the curve of NDVI for the decreasing part of the curve during the Rabi(Kharif) season |
| Rabi\_season\_length | Difference between Rabi plant\_dates and harvest\_dates in days |
| Comparison to Norms |  |
| Rabi(Kharif)\_95th\_diff\_mn | Difference between Rabi(Kharif) mean NDVI and Rabi(Kharif) estimated 95th percentile of historical values |
| Rabi(Kharif)\_95th\_diff\_mx | Difference between Rabi(Kharif) max NDVI and Rabi(Kharif) estimated 95th percentile of historical max values |
| Rabi(Kharif)\_95th\_diff\_AUC | Difference between Rabi(Kharif) AUC NDVI and Rabi(Kharif) estimated 95th percentile of historical AUC values |
| Rabi(Kharif)\_AUC\_diff\_mn | Difference between the Rabi(Kharif) mean area under the curve and Rabi(Kharif) area under the curve for NDVI |

## 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

### Input Data

A summary table of the value of the dependent variable and all independent variables can be found below:

*Table 1: Variable summary statistics and descriptions*

|  |  |  |
| --- | --- | --- |
|  | Mean | SD |
| **yield\_tn\_ha** | 4.249 | 0.5465 |
| **Rabi\_plant\_dates** | 314.6 | 9.729 |
| **Rabi\_harvest\_dates** | 134.5 | 6.07 |
| **Rabi\_season\_length** | 185.2 | 11.07 |
| **Rabi\_max\_date** | 43.3 | 20.27 |
| **Rabi\_mean** | 4,441 | 480.8 |
| **Rabi\_min** | 2,044 | 545.1 |
| **Rabi\_max** | 6,775 | 829.1 |
| **Rabi\_AUC** | 106,331 | 13,640 |
| **Rabi\_95th\_prct** | 6,722 | 827.3 |
| **Rabi\_max\_95th\_prct** | 7,217 | 686.5 |
| **Rabi\_AUC\_95th\_prct** | 115,735 | 13,789 |
| **Rabi\_AUC\_v2** | 106,324 | 13,638 |
| **Rabi\_AUC\_leading** | 59,924 | 11,199 |
| **Rabi\_AUC\_trailing** | 46,365 | 9,367 |
| **Rabi\_AUC\_diff\_mn** | 5,524 | 5,556 |
| **Rabi\_AUC\_diff\_90th** | -7,673 | 6,386 |
| **Rabi\_sd** | 1,711 | 442.2 |
| **Whe\_Yeild\_kgha** | 4,308 | 508.6 |
| **yield\_tn\_ha\_dual** | 4.232 | 0.6179 |
| **Kharif\_plant\_dates** | 134.4 | 6.506 |
| **Kharif\_harvest\_dates** | 315.1 | 8.877 |
| **R\_mx\_dates** | 235.1 | 12.31 |
| **Kharif\_mean** | 4,185 | 621.2 |
| **Kharif\_min** | 2,023 | 478 |
| **Kharif\_max** | 6,549 | 898.7 |
| **Kharif\_AUC** | 96,032 | 14,959 |
| **Kharif\_95th\_prct** | 6,477 | 889.8 |
| **Kharif\_max\_95th\_prct** | 7,229 | 694.7 |
| **Kharif\_AUC\_95th\_prct** | 105,651 | 13,845 |
| **Kharif\_AUC\_v2** | 96,029 | 14,959 |
| **Kharif\_AUC\_leading** | 48,912 | 10,894 |
| **Kharif\_AUC\_trailing** | 47,053 | 10,536 |
| **Kharif\_95th\_diff\_mn** | -2,293 | 461.2 |
| **Kharif\_AUC\_diff\_mn** | 1.424e-12 | 4,530 |
| **Rabi\_95th\_diff\_mn** | -2,281 | 623.3 |
| **Rabi\_95th\_diff\_mx** | -441.7 | 313.1 |
| **Rabi\_95th\_diff\_AUC** | -9,404 | 6,709 |
| **All\_95th\_prct** | 6,868 | 733 |

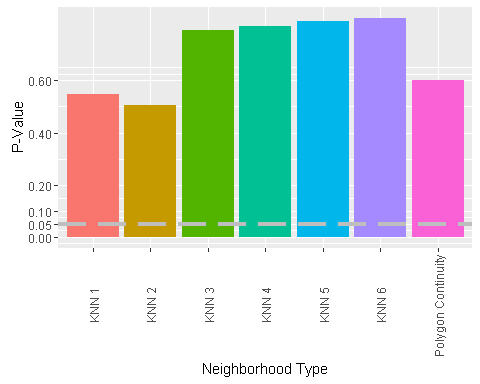
### Panel Regression Methods and Diagnostic results

## OGR data source with driver: ESRI Shapefile   
## Source: "H:/Projects/India\_Index\_Insurance/Data/Admin Boundaries", layer: "PunjabHaryanaDistricts"  
## with 43 features  
## It has 14 fields

To avoid overstating the statistical significance of regression coefficients we test the residuals from a pooled linear for spatial autocorrelation (Fotheringham, Brunsdon, and Charlton 2002). Here we estimate equation 1:

Where is a vector of our dependent variable, district yields in tons per hectare (*yield\_tn\_ha*), for each district *i* for each year *t*, is an intercept term, is a vector of *k* coefficients, corresponds to the *k* independent variables describing NDVI **EVI**, and is pooled error term.

Because a better implementation of the Moran's I for panel regression residuals is not available, tests for spatial autocorrelation are run on the mean regression residuals for each district (*i*) (Cliff and Ord 1981). Tests were carried out for seven potential neighborhood definitions including six different measures of K-nearest Neighbor, and a queen's polygon continuity (Figure 4). Moran's I test statistics are less than -0.57, which fails to reject (p <=0.5) the null hypothesis of spatial independence.

*Figure 4: Statistical significance of Moran's I on pooled regression residuals*  

#### Tests: Pooled, Fixed or Random

Testing is required to choose the proper estimation method for panel regression. We must choose between pooled, fixed effect (FE), and random effect (RE) models. First we can test for poolability of our model. Pooled regression assumes a constant intercept and slopes between different districts and time periods. We can test if variance across districts is equal to zero using an F-test. Here we reject the use of pooled OLS (p <= 4.654403210^{-16}) in favor of a fixed effects model with unique intercepts for each district. We can then compare the use of FE and RE models. The hausman test checks for exogeniety of the unobserved error component, tf the null hypothesis is rejected, the RE model is inconsistent, and the FE model will be preferred. If individual effects are exogenous both fixed (FE) and random effects are asymptotically equivalent. Here we test if , where are coefficient vectors of time-varying explanitory variables. Here we reject the null hypothesis (p >= 9.44e-72) and choose to use the fixed effects estimator as it will be more efficient.

#### Multicolinearity & Principal Components Analysis Transform

Multicolinearity, high correlations between independent variables, can increase estimates of a variable's estimated variance. This can have the adverse effect of creating models in which the is high and no variables are statistically significant. Multicolinearity can also produce coefficients of the 'wrong sign' and of unreasonable magnitude (O’Brien 2007; Greene, n.d.). Here we use a variance inflation factor (VIF) to quantify how much the variance is inflated for each coefficient[[1]](#footnote-42). VIF values over 4 for any variable are generally considered problematic and require further examination (Greene, n.d.). Here we present summary statistics for VIFs on a ordinary least squares estimation of all model variable in Table (2) below:

*Table 2: Variance inflation factor summary table*

##   
## =====================================================  
## Statistic N Mean St. Dev. Min Max   
## -----------------------------------------------------  
## vif(fit) 32 77,772.790 179,885.300 1.361 634,503.100  
## -----------------------------------------------------

To avoid problems with multicolinearity between independent variables we apply a principal components analysis transformation (PCA) to a centered and scaled matrix of independent variables . PCA allows for the replacement of with a new matrix whos variables are orthoganal to each other but span the multidimensional space of (Geladi and Kowalski 1986). In this case we generate 36 principal components for inclusion in a random effects panel regression. A table of PCA component imporance can be found in the appendix, Table(A1).

#### OLS Regression Estimation

#### Panel Regression Estimation

We use panel data to model wheat yields over time at the district level. The use of **(spatial)** panel data in this study helps to alleviate two key problems, unobserved spatial and temporal dynamics, and homogeneity (lack of variance). When pooled together, the integration of two statewide data sets (N = 37) over the 2003-2012 period (T = 10) provides 370 observations. Compared to cross-sectional approaches this amounts to a substantial increase in the degree of observed variance over both space and time.

Here we estimate equation 2:

Where is a vector of our dependent variable, district yields in tons per hectare (*yield\_tn\_ha*), for each district *i* for each year *t*, is an intercept term, is a one-period lag of the first pricipal component, is a vector of *k* coefficients, corresponding to the *H* principal components, is the between entity error term, and is the within entity error term. To capture non-linear effects here we apply natural cubic splines to the first 4 principal components (Devlin, Weeks, and others 1986).[[2]](#footnote-45) Natural cubic splines use cubic terms in the center of the data and restrict the ends to straight lines, preventing distortion at extreme values. When reported splines are noted with rcs(variable\_name, number\_of\_knots) with order noted by ' and ''.

# Results

## Focus Group Interviews

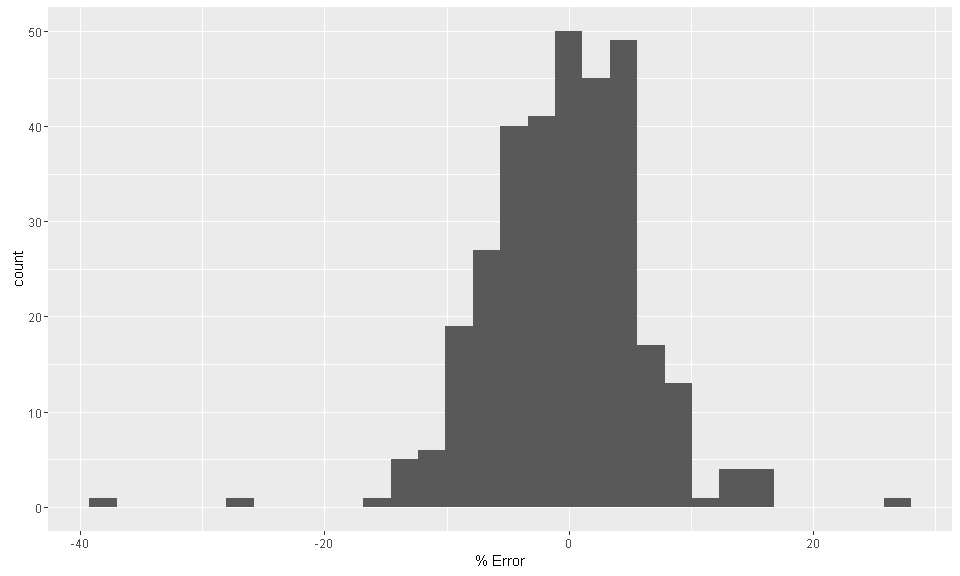
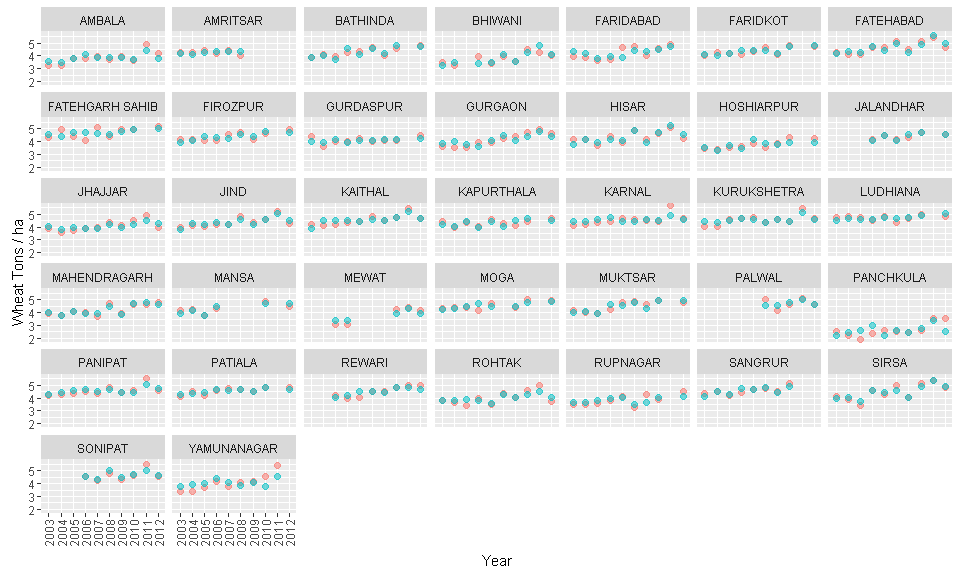
## Panel Regression

Estimates of equation (2) are provide below:

*District level random effects estimation of wheat yields in tons per hectare*

## Oneway (individual) effect Within Model  
##   
## Call:  
## plm(formula = PCA\_formula\_regression, data = pca\_pred, model = "within",   
## index = c("district", "years"))  
##   
## Unbalanced Panel: n=37, T=5-10, N=325  
##   
## Residuals :  
## Min. 1st Qu. Median 3rd Qu. Max.   
## -7.12e-01 -1.69e-01 -7.33e-05 1.60e-01 9.54e-01   
##   
## Coefficients :  
## Estimate Std. Error t-value Pr(>|t|)   
## rcs(PC1, 4)PC1 73.496731 45.838396 1.6034 0.11007   
## rcs(PC1, 4)PC1' -0.140365 0.392360 -0.3577 0.72083   
## rcs(PC1, 4)PC1'' 0.191723 0.767077 0.2499 0.80283   
## rcs(PC2, 4)PC2 2.332585 6.263697 0.3724 0.70990   
## rcs(PC2, 4)PC2' 0.851204 0.357265 2.3826 0.01791 \*  
## rcs(PC2, 4)PC2'' -2.036087 0.928025 -2.1940 0.02912 \*  
## rcs(PC3, 4)PC3 -5.787391 9.040725 -0.6401 0.52264   
## rcs(PC3, 4)PC3' -0.022944 0.076666 -0.2993 0.76497   
## rcs(PC3, 4)PC3'' -0.329891 0.341536 -0.9659 0.33499   
## rcs(PC4, 4)PC4 -62.616586 36.267156 -1.7265 0.08544 .  
## rcs(PC4, 4)PC4' -0.152705 0.133995 -1.1396 0.25549   
## rcs(PC4, 4)PC4'' 0.865224 0.475665 1.8190 0.07007 .  
## PC5 33.971845 19.567201 1.7362 0.08372 .  
## PC6 4.249612 4.749590 0.8947 0.37176   
## PC7 36.344069 20.223361 1.7971 0.07348 .  
## PC8 5.556272 5.301196 1.0481 0.29556   
## PC9 -29.582268 23.305627 -1.2693 0.20547   
## PC10 2.214074 5.412826 0.4090 0.68285   
## PC11 -35.667843 21.425701 -1.6647 0.09718 .  
## PC12 -64.612457 36.543076 -1.7681 0.07822 .  
## PC13 169.223703 96.057708 1.7617 0.07930 .  
## PC14 194.973513 107.618222 1.8117 0.07119 .  
## PC15 -77.344037 66.903814 -1.1560 0.24873   
## PC16 -197.344649 109.905147 -1.7956 0.07373 .  
## PC17 61.328072 34.619333 1.7715 0.07765 .  
## PC18 -30.563499 36.423618 -0.8391 0.40218   
## PC19 200.679512 127.089977 1.5790 0.11555   
## PC20 -52.709635 30.384905 -1.7347 0.08398 .  
## PC21 137.774575 78.027850 1.7657 0.07862 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Total Sum of Squares: 47.797  
## Residual Sum of Squares: 19.924  
## R-Squared: 0.58316  
## Adj. R-Squared: 0.46473  
## F-statistic: 12.4944 on 29 and 259 DF, p-value: < 2.22e-16

## [1] "Within R2 0.58"

*Figure 5: Fitted vs actual for estimation of equation (3)*  

## Spatial Panel Regression

*Actual values (red), predicted values (blue)*

## OGR data source with driver: ESRI Shapefile   
## Source: "H:/Projects/India\_Index\_Insurance/Data/Admin Boundaries", layer: "PunjabHaryanaDistricts"  
## with 43 features  
## It has 14 fields

## OGR data source with driver: ESRI Shapefile   
## Source: "H:/Projects/India\_Index\_Insurance/Data/Admin Boundaries", layer: "PunjabHaryanaDistricts"  
## with 43 features  
## It has 14 fields

##   
## 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 14 14 14 14 14 14 14 14 14 14

## Note: method with signature 'sparseMatrix#ANY' chosen for function 'kronecker',  
## target signature 'dgCMatrix#dgeMatrix'.  
## "ANY#Matrix" would also be valid

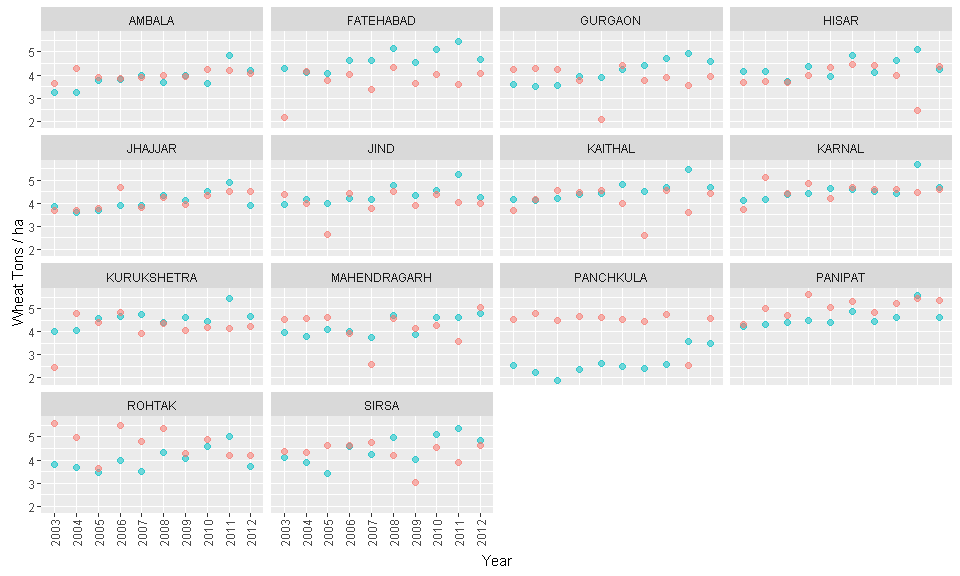
## Note: method with signature 'dsparseMatrix#dsparseMatrix' chosen for function 'kronecker',  
## target signature 'dgCMatrix#dgTMatrix'.  
## "sparseMatrix#TsparseMatrix" would also be valid

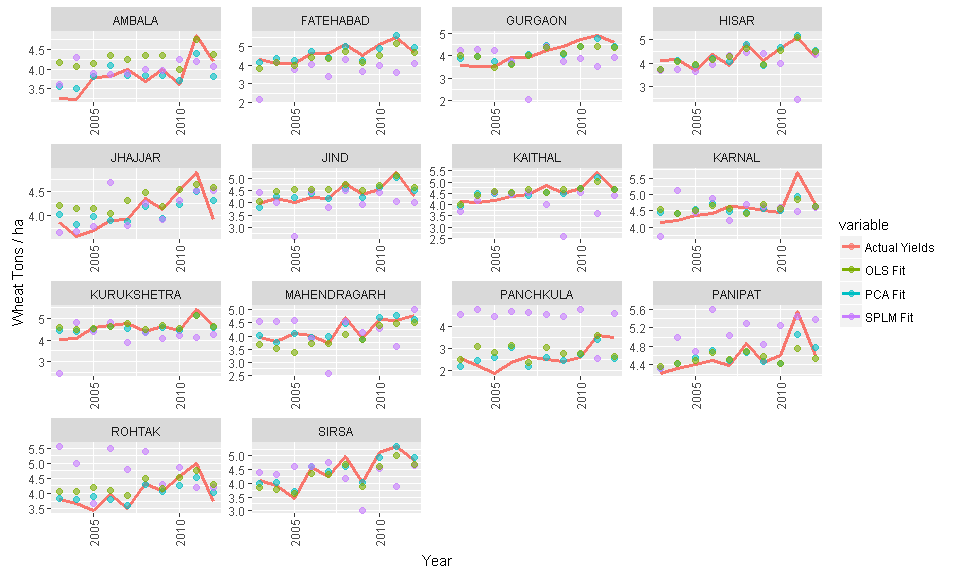
## Note: method with signature 'sparseMatrix#matrix' chosen for function '%\*%',  
## target signature 'dgTMatrix#matrix'.  
## "TsparseMatrix#ANY" would also be valid

## Spatial panel fixed effects sarar model  
##   
## Call:  
## spml(formula = PCA\_formula\_regression, data = pca\_pred\_balanced,   
## index = c("district", "years"), listw = districts\_polyListw,   
## model = "within", lag = TRUE, spatial.error = "b")  
##   
## Residuals:  
## Min. 1st Qu. Median 3rd Qu. Max.   
## -0.58200 -0.11100 -0.00262 0.09520 0.59600   
##   
## Coefficients:  
## Estimate Std. Error t-value Pr(>|t|)   
## lambda 0.760458 0.046803 16.2481 < 2.2e-16 \*\*\*  
## rho -0.536668 0.095173 -5.6389 1.711e-08 \*\*\*  
## rcs(PC1, 4)PC1 55.112468 37.365241 1.4750 0.1402   
## rcs(PC1, 4)PC1' 0.114151 0.223637 0.5104 0.6098   
## rcs(PC1, 4)PC1'' -0.212104 0.437880 -0.4844 0.6281   
## rcs(PC2, 4)PC2 3.435757 6.803356 0.5050 0.6136   
## rcs(PC2, 4)PC2' 0.695700 0.429066 1.6214 0.1049   
## rcs(PC2, 4)PC2'' -1.823459 1.193849 -1.5274 0.1267   
## rcs(PC3, 4)PC3 7.276776 9.979433 0.7292 0.4659   
## rcs(PC3, 4)PC3' 0.096971 0.084746 1.1443 0.2525   
## rcs(PC3, 4)PC3'' -0.511711 0.314702 -1.6260 0.1039   
## rcs(PC4, 4)PC4 -36.281540 27.220124 -1.3329 0.1826   
## rcs(PC4, 4)PC4' -0.084174 0.118399 -0.7109 0.4771   
## rcs(PC4, 4)PC4'' 0.500813 0.489080 1.0240 0.3058   
## PC5 18.892554 14.503815 1.3026 0.1927   
## PC6 -1.173909 4.856978 -0.2417 0.8090   
## PC7 19.115398 14.599834 1.3093 0.1904   
## PC8 -3.402317 5.300585 -0.6419 0.5210   
## PC9 -35.160277 22.871581 -1.5373 0.1242   
## PC10 7.199923 6.288244 1.1450 0.2522   
## PC11 -11.274268 15.801429 -0.7135 0.4755   
## PC12 -44.805718 27.787062 -1.6125 0.1069   
## PC13 111.875758 71.766894 1.5589 0.1190   
## PC14 99.770826 75.849789 1.3154 0.1884   
## PC15 -91.484883 69.796544 -1.3107 0.1899   
## PC16 -116.876983 80.730758 -1.4477 0.1477   
## PC17 19.114389 25.117634 0.7610 0.4467   
## PC18 14.802264 39.499980 0.3747 0.7079   
## PC19 145.845156 107.364881 1.3584 0.1743   
## PC20 -14.426790 23.214024 -0.6215 0.5343   
## PC21 80.861089 58.713445 1.3772 0.1684   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## [1] 0.9322996

## [1] 0.9929734





# Discussion

# Conclusions

# Appendix A

## Yield Data

*Table A2: Rabi Season Wheat Yields Metric Tons per Hectare by State*

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 | 3.28 | 3.779 | 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 | BATHINDA | 3.89 | 4.281 | 4.79 |
| PUNJAB | FARIDKOT | 4.04 | 4.35 | 4.81 |
| PUNJAB | FATEHGARH SAHIB | 4.09 | 4.666 | 5.18 |
| 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 | 4.516 | 5.01 |
| PUNJAB | MUKTSAR | 3.94 | 4.45 | 4.98 |
| PUNJAB | PATIALA | 4.12 | 4.55 | 4.83 |
| PUNJAB | RUPNAGAR | 3.31 | 3.807 | 4.51 |
| PUNJAB | SANGRUR | 4.23 | 4.574 | 5.13 |

## Principal Components Analysis

*Table A1: Importance of PCA components*

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 3.4705 2.8295 2.2751 1.56211 1.27376 1.22646  
## Proportion of Variance 0.3346 0.2224 0.1438 0.06778 0.04507 0.04178  
## Cumulative Proportion 0.3346 0.5569 0.7007 0.76851 0.81358 0.85537  
## PC7 PC8 PC9 PC10 PC11 PC12  
## Standard deviation 1.12703 1.04822 0.88901 0.76207 0.73851 0.57710  
## Proportion of Variance 0.03528 0.03052 0.02195 0.01613 0.01515 0.00925  
## Cumulative Proportion 0.89065 0.92117 0.94312 0.95926 0.97441 0.98366  
## PC13 PC14 PC15 PC16 PC17 PC18  
## Standard deviation 0.41604 0.36745 0.27546 0.21815 0.1992 0.17711  
## Proportion of Variance 0.00481 0.00375 0.00211 0.00132 0.0011 0.00087  
## Cumulative Proportion 0.98847 0.99222 0.99432 0.99565 0.9968 0.99762  
## PC19 PC20 PC21 PC22 PC23 PC24  
## Standard deviation 0.15783 0.13625 0.12338 0.10855 0.06597 0.06181  
## Proportion of Variance 0.00069 0.00052 0.00042 0.00033 0.00012 0.00011  
## Cumulative Proportion 0.99831 0.99883 0.99925 0.99958 0.99970 0.99980  
## PC25 PC26 PC27 PC28 PC29 PC30  
## Standard deviation 0.05294 0.04926 0.03593 0.02273 0.002726 0.001797  
## Proportion of Variance 0.00008 0.00007 0.00004 0.00001 0.000000 0.000000  
## Cumulative Proportion 0.99988 0.99995 0.99999 1.00000 1.000000 1.000000  
## PC31 PC32 PC33 PC34 PC35  
## Standard deviation 0.001187 0.0009753 1.001e-15 6.899e-16 3.002e-16  
## Proportion of Variance 0.000000 0.0000000 0.000e+00 0.000e+00 0.000e+00  
## Cumulative Proportion 1.000000 1.0000000 1.000e+00 1.000e+00 1.000e+00  
## PC36  
## Standard deviation 2.846e-16  
## Proportion of Variance 0.000e+00  
## Cumulative Proportion 1.000e+00

## 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*

quantile\_type8 = function(x){  
 quantile(x ,p=Quant\_percentile,type=8,na.rm=T)  
}

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

PeriodAggregatorDates = function(x,dates\_in,date\_range\_st, date\_range\_end,by\_in='days',FUN){  
 # returns a date of summary statistic defined by FUN  
 # like the date of the maximum value of x for the period defined by date\_range\_st, date\_range\_end  
 # other parameters identical to other functions show above  
 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){  
 DateRange2 = seq(date\_range\_st[z],date\_range\_end[z],by=by\_in)  
 x2 = x[dates\_in %in% DateRange2]  
 dates\_in2 = dates\_in[dates\_in %in% DateRange2]  
 which\_max = which(FUN(x2) == x2)  
 if(length(which\_max)>1){  
 which\_max = c(which\_max[1],which\_max[length(which\_max)]) # limit to only 2   
 if((which\_max[2]-which\_max[1])==1){  
 which\_max=which\_max[1] # favor the first instance of maximum  
 } else if((which\_max[2]-which\_max[1])==2){  
 which\_max=which\_max[1]+1 # is seperated by 2 choose middle left  
 } else if((which\_max[2]-which\_max[1])==3){  
 which\_max=which\_max[1]+2} # is seperated by 3 choose middle  
 }  
 max\_dates = dates\_in2[which\_max]  
 })  
 dataout = do.call(c,dataout)  
 names(dataout)=format(date\_range\_st,'%Y')  
 dataout  
 }

*Functions 7: Mean day of the year values*

AnnualAverageDOYvalues = function(x,dates\_in){  
 # calculates the average value for DOY for the whole series  
 datesj = format(dates\_in,'%j')  
 do.call(c,lapply(split(x,datesj),function(y){mean(y,na.rm=T)}))}

*Functions 8: Smoothing splines with outlier removal*

#---------------------------------------------------------------------  
# This function takes a time series w/ dates (x, dates) and returns a spline smoothed time series with outliers removed.  
# Outliers are identified as points with absolute value more than out\_sigma \* sd, where sd is the residual  
# standard deviation between the input data and the initial spline fit, and out\_sigma is a variable  
# coefficient. The spline smoothing parameter spline\_spar controls the smoothness of the fit (see spline.smooth help)  
# and out\_iterations controls the number of times that outliers are checked and removed w/ subsequent spline refit  
# pred\_dates is a vector of dates where spline smoothed predictions of x are desired. If NA, then a daily series spanning  
# min(dates)-max(dates) is returned  
SplineAndOutlierRemoval <- function(x, dates, out\_sigma=3, spline\_spar=0.3, out\_iterations=1,pred\_dates){  
 dates <- as.numeric(dates) # spline doesn't work with dates  
 pred\_dates = as.numeric(pred\_dates)  
 # if prediction dates aren't provided, we assume we want daily ones  
 if(is.na(pred\_dates[1])){  
 pred\_dates <- min(dates, na.rm=T):max(dates, na.rm=T)}  
 # eliminate outliers and respline  
 for(i in 1:out\_iterations){  
 # fit a smoothing spline to non-missing data  
 spl <- try(smooth.spline(dates[!is.na(x)], x[!is.na(x)], spar=spline\_spar), silent=T)  
 if(inherits(spl, 'try-error')){  
 print("Failed to fit smoothing spline")  
 return(NA)  
 }  
 smooth\_x <- try(predict(spl, dates)$y, silent=T) # calculate spline smoothed values  
 if(inherits(smooth\_x, 'try-error')){  
 print("Failed to predict with spline")  
 return(NA)  
 }  
 smooth\_x\_resid <- x - smooth\_x # calculate residuals from spline  
 smooth\_x\_resid\_sd <- try(sd(smooth\_x\_resid, na.rm=T), silent=T) # standard dev of absolute value of residuals  
 if(inherits(smooth\_x\_resid\_sd, 'try-error')){  
 print("Failed to get sd of residuals")  
 return(NA)  
 }  
 outliers <- abs(smooth\_x\_resid) > out\_sigma \* smooth\_x\_resid\_sd  
 outliers[is.na(outliers)] <- F  
 if(sum(outliers) > 0){  
 # if we found outliers, eliminate them in x and refit up to iterations  
 x[outliers] <- NA  
 }else{  
 # if we didn't find any outliers, we abandon the iteration and return the smoothed values  
 smooth\_x\_return <- try(predict(spl, pred\_dates)$y, silent=T)  
 if(inherits(smooth\_x\_return, 'try-error')){  
 print("No outliers, but failed to predict with final spline")  
 return(NA)  
 }else{  
 return(smooth\_x\_return)  
 }  
 }  
 }  
 # fit the spline to the outlier screened data, then return the predicted series  
 spl <- try(smooth.spline(dates[!is.na(x)], x[!is.na(x)], spar=spline\_spar), silent=T)  
 if(inherits(spl, 'try-error')){  
 print("Failed to predict with final spline")  
 return(NA)  
 }else{  
 smooth\_x\_return <- try(predict(spl, pred\_dates)$y, silent=T)  
 if(inherits(smooth\_x\_return, 'try-error')){  
 return(NA)  
 }else{  
 return(smooth\_x\_return)  
 }  
 }  
}

*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

Cliff, Andrew, and John Ord. 1981. *Spatial Processes: models and applications*. Edited by Anonymous. London: Pion.

Devlin, TF, BJ Weeks, and others. 1986. “Spline Functions for Logistic Regression Modeling.” In *Proceedings of the 11th Annual SAS Users Group International Conference*, 646:51. SAS Institute Inc., Cary, NC.

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.

Fotheringham, Stewart, Chris Brunsdon, and Martin Charlton. 2002. *Geographically Weighted Regression‚ The analysis of spatially varying relationships*. Edited by Anonymous. Hoboken: Wiley.

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>.

Geladi, Paul, and Bruce R. Kowalski. 1986. “Partial Least-Squares Regression: A Tutorial.” *Analytica Chimica Acta* 185: 1–17. doi:[http://dx.doi.org/10.1016/0003-2670(86)80028-9](https://doi.org/http://dx.doi.org/10.1016/0003-2670(86)80028-9).

Greene, William. n.d. “H.(1997): Econometric Analysis.” New York: Prentice Hall.

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.

O’Brien, Robert M. 2007. “A Caution Regarding Rules of Thumb for Variance Inflation Factors.” *Quality & Quantity* 41 (5). Springer: 673–90.

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.

State, Penn. 2016. “Detecting Multicollinearity Using Variance Inflation Factors.” <https://onlinecourses.science.psu.edu/stat501/node/347>.

1. A nice synopsis of VIF can be found here (State 2016). [↑](#footnote-ref-42)
2. The first four principal components in this PCA comprises 77 % of total variance in [↑](#footnote-ref-45)