SENTINELS OF THE HARVEST

GEOSPATIAL MONITORING OF CROPLAND HEALTH FOR A CHANGING CLIMATE

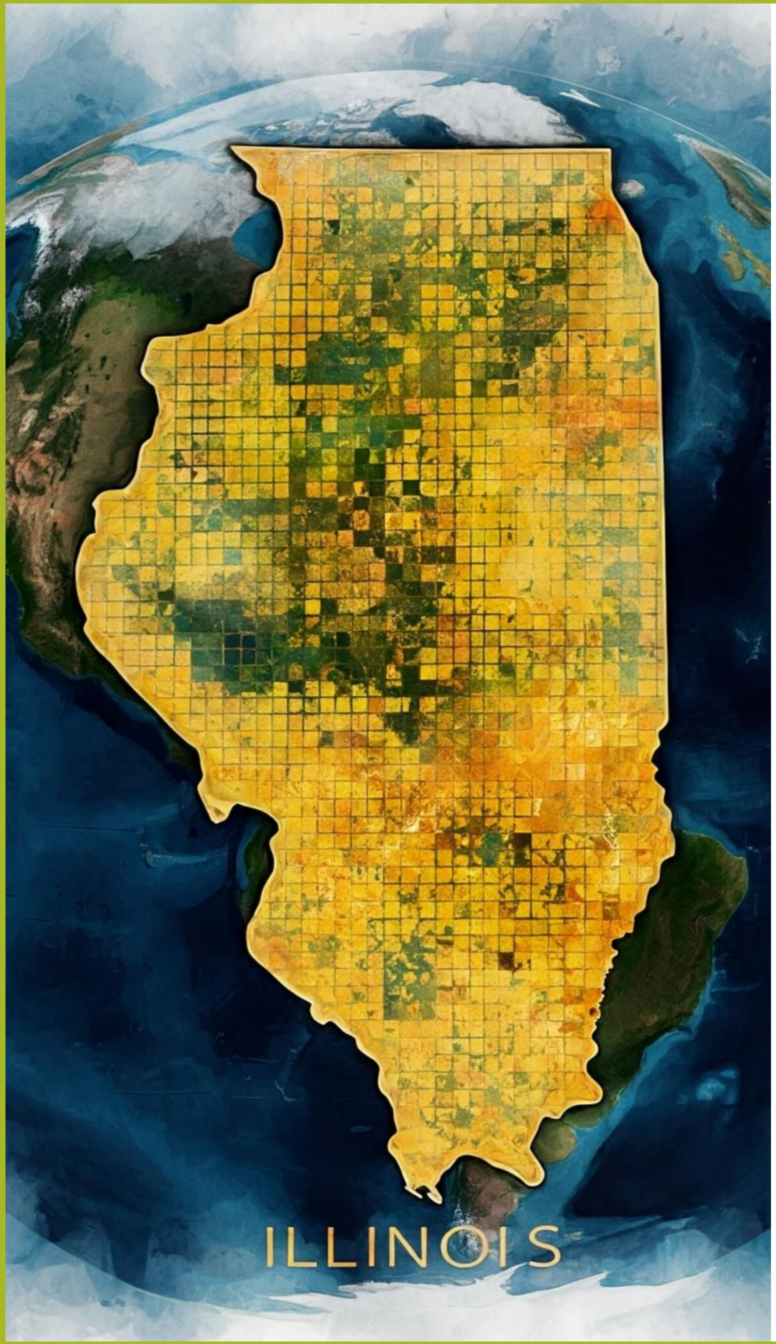
Team SaFe MeMe

Sara Hunsberger, Meucci Ilunga, Fernanda Montoya, Meklit Yimenu

140.777 | Q2 2024

Why do we care?

- Cropland is an important thing to look at because of the US dependence on it
- Climate change presents a threat to cropland health
- Previous Research:
  - Studies on climate change and vegetation in general



Phase 1: Data Collection and Preparation

Objective:

- Analyze the impact of climate change on cropland health and productivity in Illinois.

Methodology:

- Utilize Google Earth Engine API to access and analyze geospatial data.
- Focus on Illinois croplands.
- Analyze data at a 1km x 1km sptial resolution.
- Use data from 02/2000 to 12/2023.
- Use weekly resolution (1244 weeks in total)

Datasets:

- USDA NASS Cropland Data Layers
- MODIS (Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), Terra MODIS Vegetation Continuous Fields)
- PRISM (temperature and precipitation)

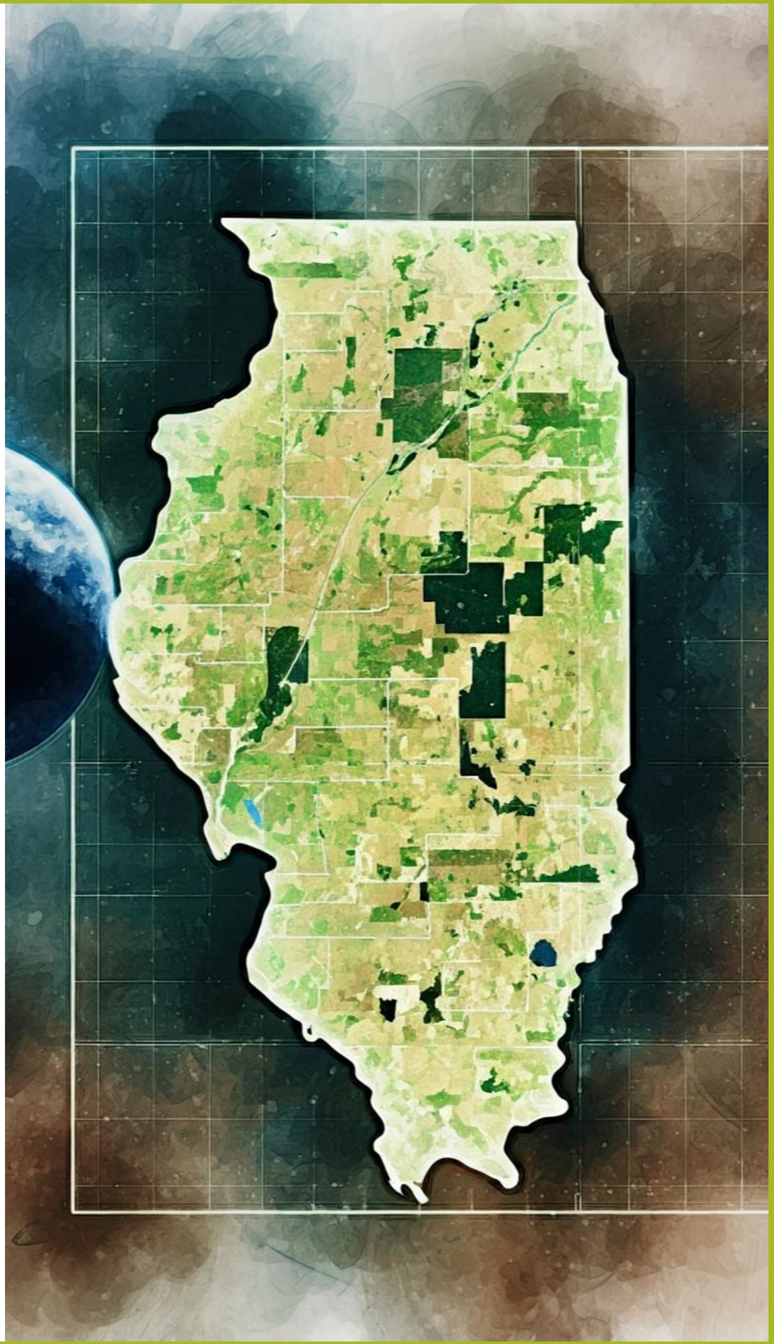
Parameters Obtained:

- Daymet: dayl, prep, srad, swe, tmax, tmin, vp
- MODIS: EVI, NDVI, ET, LE, PET, PLE, FPAR, LAI

Phase 1: Data Collection and Preparation

- **Pipeline Overview**:
  - Functional programming paradigms were essential to efficiently process large datasets from Google Earth Engine (GEE).

- **Automation and Preprocessing**:
  - **Bash Scripts**: Automated the retrieval of GeoTIFFs via GEE over an 8-hour runtime to manage large-scale data pulls. Downloaded 1244 GeoTIFF files (11.2GB) directly via API.
  - **Object-Oriented Programming**: Leveraged a custom GeoTIFFImage class in Python for preprocessing, including metadata extraction, band visualization, and flattening to pandas DataFrames.
  - Preprocessing ensured data quality by clipping to Illinois, aligning temporal spans, and removing non-relevant pixels.
  - Produced **205,151,722** worth of 20-dimensional data-points



Examples of Shell Scripting / Functional Programming

```bash
#!/bin/bash

# Run the Python script in the background via nohup; log terminal prints to home dir file
nohup python3 -u /home/katavga/code/safe-meme/gee_acquire_data.py > ~/gee_acquire_data.log 2>&1 &

# Confirm successful script instantiation
echo "Script is running in the background. Logs are being written to ~/gee_acquire_data.log"

# Verify that the process is running
ps -ef | grep gee_acquire_data.py | grep -v grep
```

```python
# Ensure all images across all bands have consistent, common projection
# Since focusing on CONUS, use EPSG:3347 as default
target_projection_crs = ee.Projection('EPSG:3347')
target_scale = 1000  # meters

def reproject_image(image):
    return image.reproject(crs=target_projection_crs, scale=target_scale)

merged_dataset = merged_dataset.map(reproject_image)
```

Example of OOP Paradigm Usage

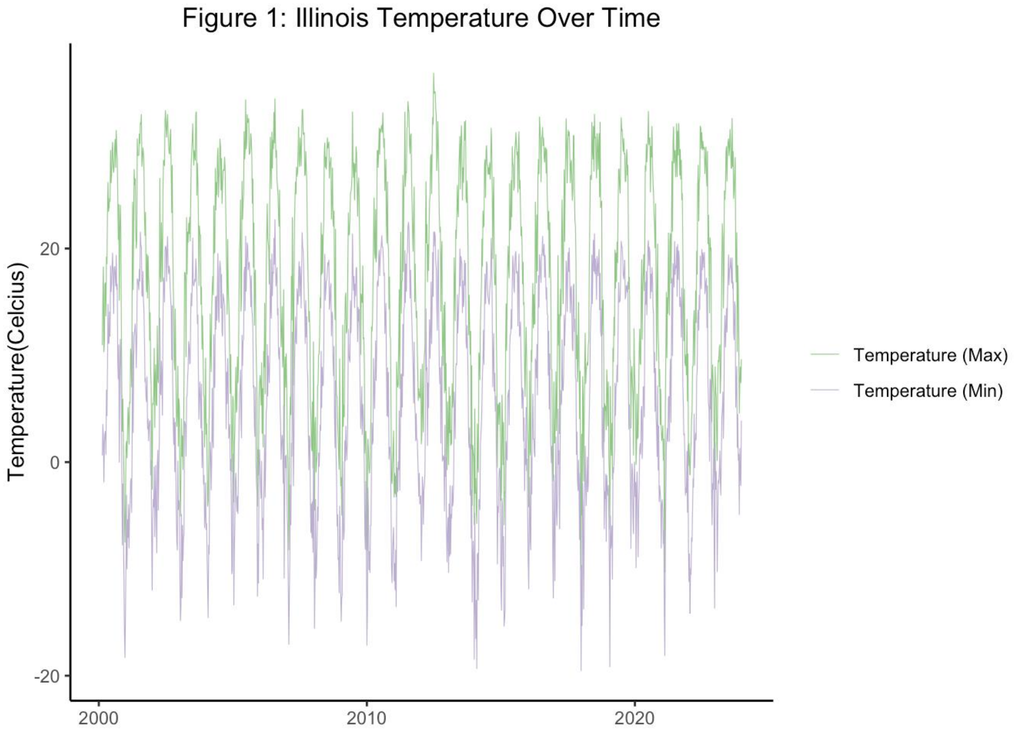
```python
class GeoTIFFImage:
    """
    Represents a GeoTIFF image with core attributes and data.
    """

    def __init__(self, filepath):
        """
        Initializes a GeoTIFFImage object.

        Args:
          filepath: Path to the GeoTIFF file.
        """

        with rasterio.open(filepath) as src:
            self.data = src.read()
            self.crs = src.crs
            self.transform = src.transform
            self.width = src.width
            self.height = src.height
            self.count = src.count
            self.band_names = src.descriptions
            self.nodata = src.nodata

        # Extract week number and date metadata directly from filename
        basename = os.path.basename(filepath)
        parts = basename.split('_')
        self.week_number = int(parts[2])
        self.date = parts[3].split('.')[0]  # Remove .tif extension
```

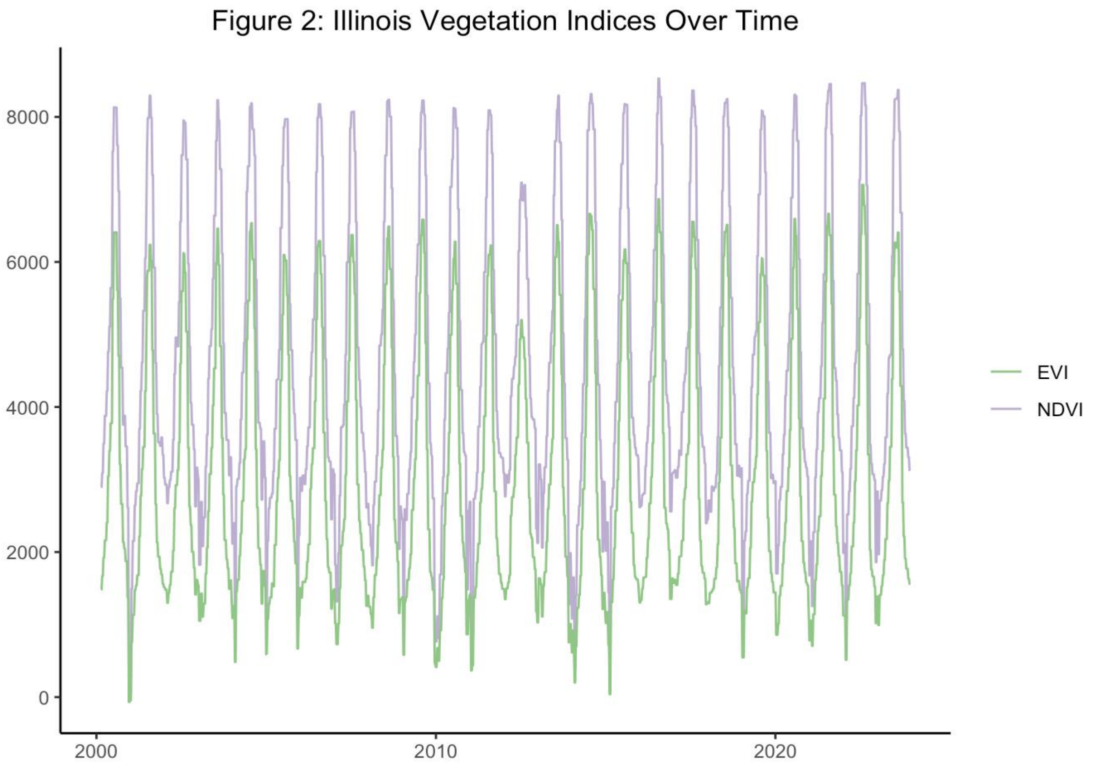
```python
def flatten(self):
    """
    Flattens the GeoTIFF data into a pandas DataFrame with specified columns.

    Returns:
      A pandas DataFrame with columns: week, date, lat, long, crop, dayl,
      prcp, srad, swe, tmax, tmin, vp, EVI, NDVI, ET, LE, PET, PLE,
      FPAR, LAI.
    """

    # Create coordinate arrays
    rows, cols = np.meshgrid(np.arange(self.height), np.arange(self.width))
    xs, ys = rasterio.transform.xy(self.transform, rows, cols)

    # Reproject coordinates to lat/lon in degrees (epsg:4326)
    transformer = Transformer.from_crs(self.crs, "epsg:4326")
    lats, lons = transformer.transform(xs, ys)

    # Flatten the data arrays
    data = {
        'week': np.full(lons.size, self.week_number),
        'date': np.full(lons.size, self.date),
        'lat': lats.flatten(),
        'long': lons.flatten(),
    }

    # Assuming band names correspond to the remaining columns
    for i, band_name in enumerate(self.band_names):
        data[band_name] = self.data[i].flatten()

    df = pd.DataFrame(data)
    return df
```

Phase 2: Exploratory Data Analysis

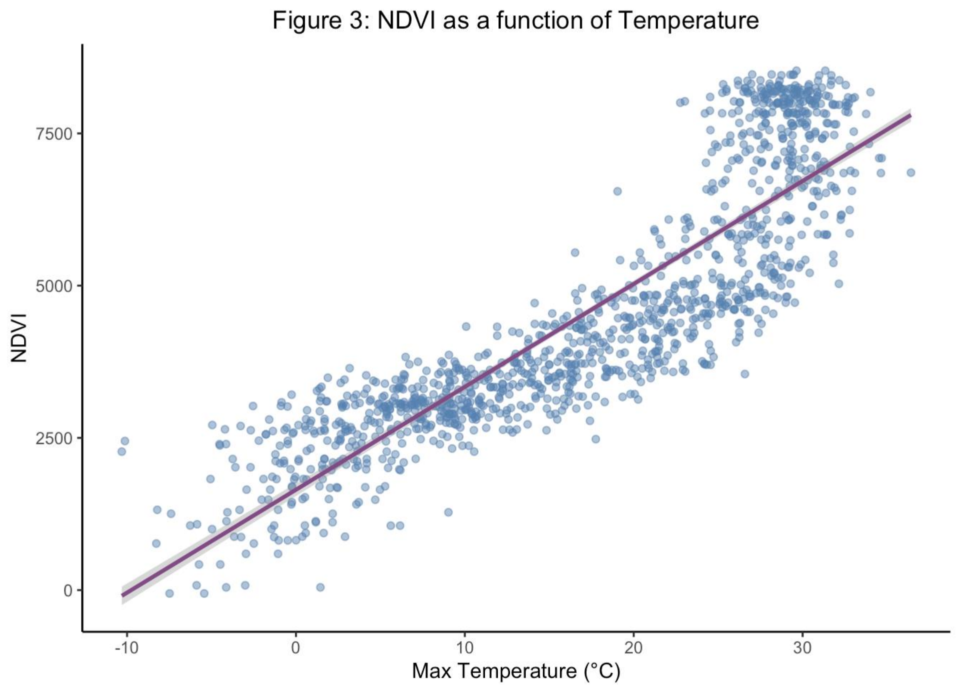
- **Data Quality Assessment:** checked for data completeness , removed duplicate rows , verified range validity for numeric columns, ensured consistency in units and data types and detected and addressed outliers where necessary.
- **Data Imputation:** Imputed missing values using group means and combined preprocessed data into a single cohesive dataset for downstream analysis.
- **Exploratory Data Analysis (EDA)**
  - Analyzed relationships between various features in the dataset.
  - Visualized the relationship between NDVI (the outcome of interest) and other variables to identify patterns.
- **Correlation and Feature Selection**
  - Computed correlation coefficients between predictors.
  - Removed highly correlated variables to reduce redundancy and avoid multicollinearity.
  - selected key features relevant to the analysis.
- **Clustering for Feature Importance**
  - Applied K-means clustering on scaled data to group observations and identify important features contributing to the outcome.

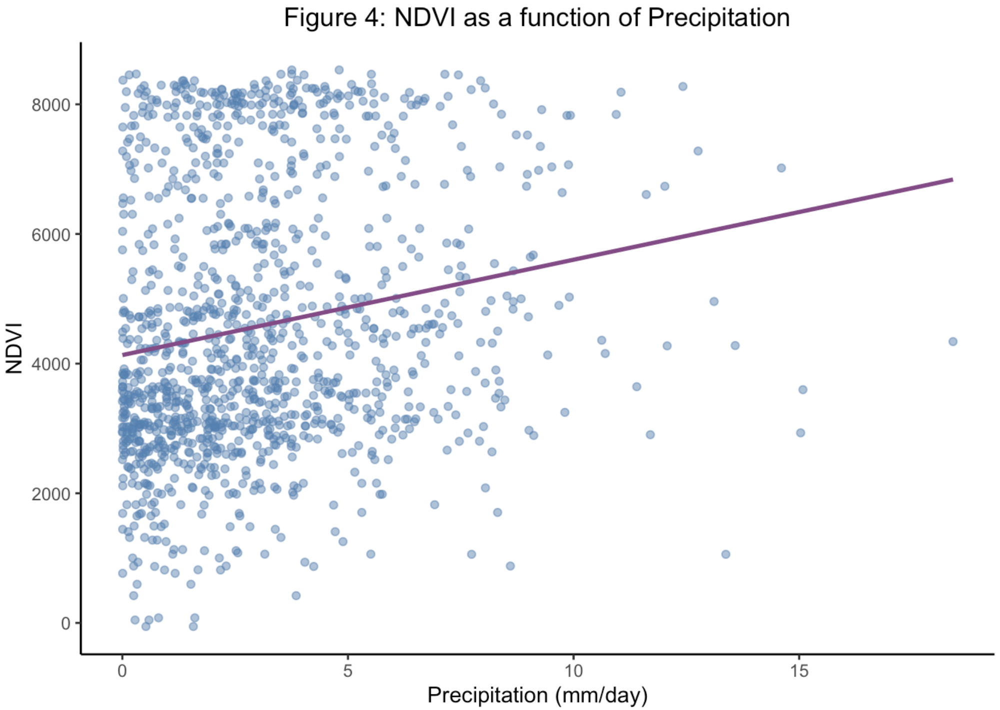
Phase 2:Examples





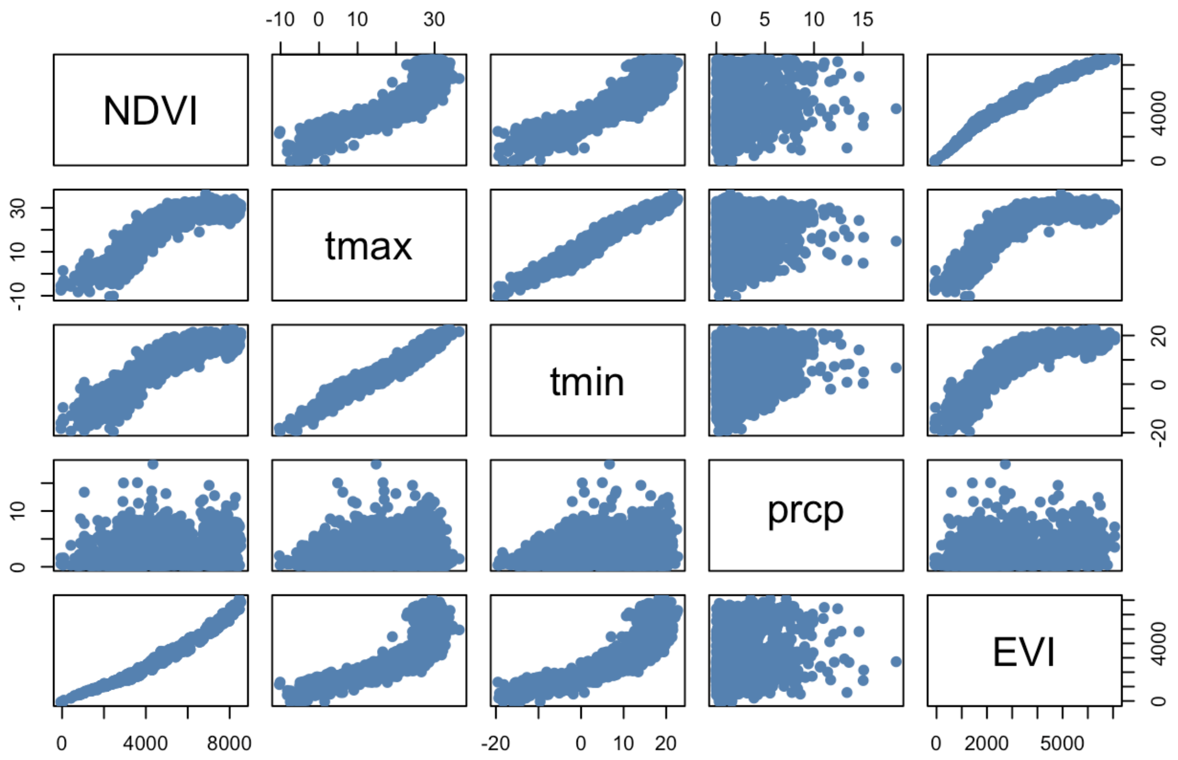
Phase 2: Examples





Phase 2: Examples

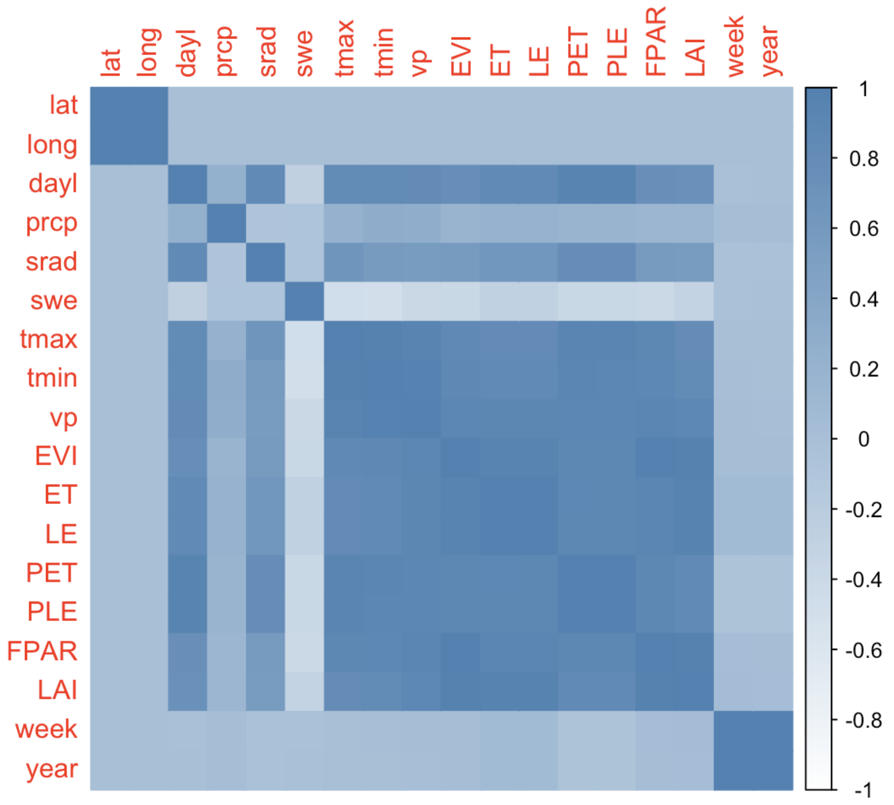
Figure 6: Pairwise Scatter Plots of NDVI and Climate/Veg Variables



Phase 2: Examples

```r
#highly correlated features (> 0.85)
high_corr <- findCorrelation(corr_matrix, cutoff = 0.85, names = TRUE)
#features that are not highly correlated
selected_features_corr <- setdiff(colnames(numeric_cols), high_corr)
selected_features_corr
```

```
[1] "lat"  "dayl" "prcp" "srad" "swe"  "tmax" "LAI"  "week"
```



Phase 2: Examples

```r
# variables with high variance within clusters
feature_variances <- apply(scaled_data, 2, var)
selected_features_kmeans <- names(feature_variances)[order(-feature_variances)[1:15]]
final_selected_features <- selected_features_kmeans
# add NDVI
final_selected_features <- unique(c(final_selected_features, "NDVI"))
cat("Selected Features for Predictive Modeling:\n")
```

```
Selected Features for Predictive Modeling:
```

```r
print(final_selected_features)
```

```
 [1] "swe"  "year" "EVI"  "LAI"  "prcp" "NDVI" "PET"  "tmax" "vp"   "week"
[11] "FPAR" "ET"   "PLE"  "tmin" "dayl"
```

Phase 3: Predictive Modeling

- **Predictive modeling to be used for future cropland health**
  - **Outcomes of interest:** Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), Fraction of Photosynthetically Active Radiation (FPAR), Leaf Area Index (LAI), Evapotranspiration (ET)
  - **Predictors:** snow water equivalent, precipitation, minimum and maximum temperature, water vapor pressure, and daylength

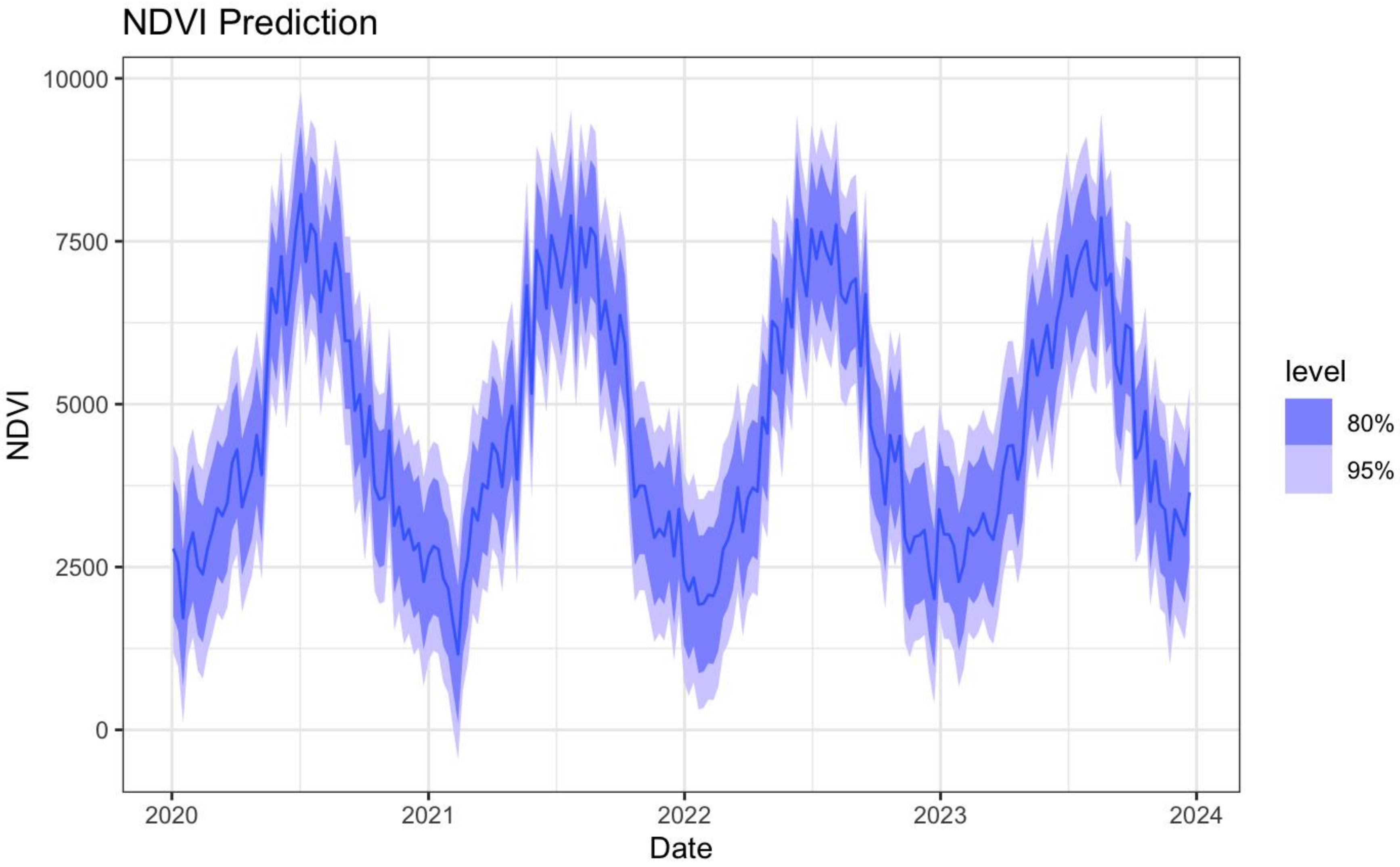
- **Method of Analysis:** Time series linear models using different outcomes of interest and predictor variables
  - Use of R packages tsibble and fable

modified_ts

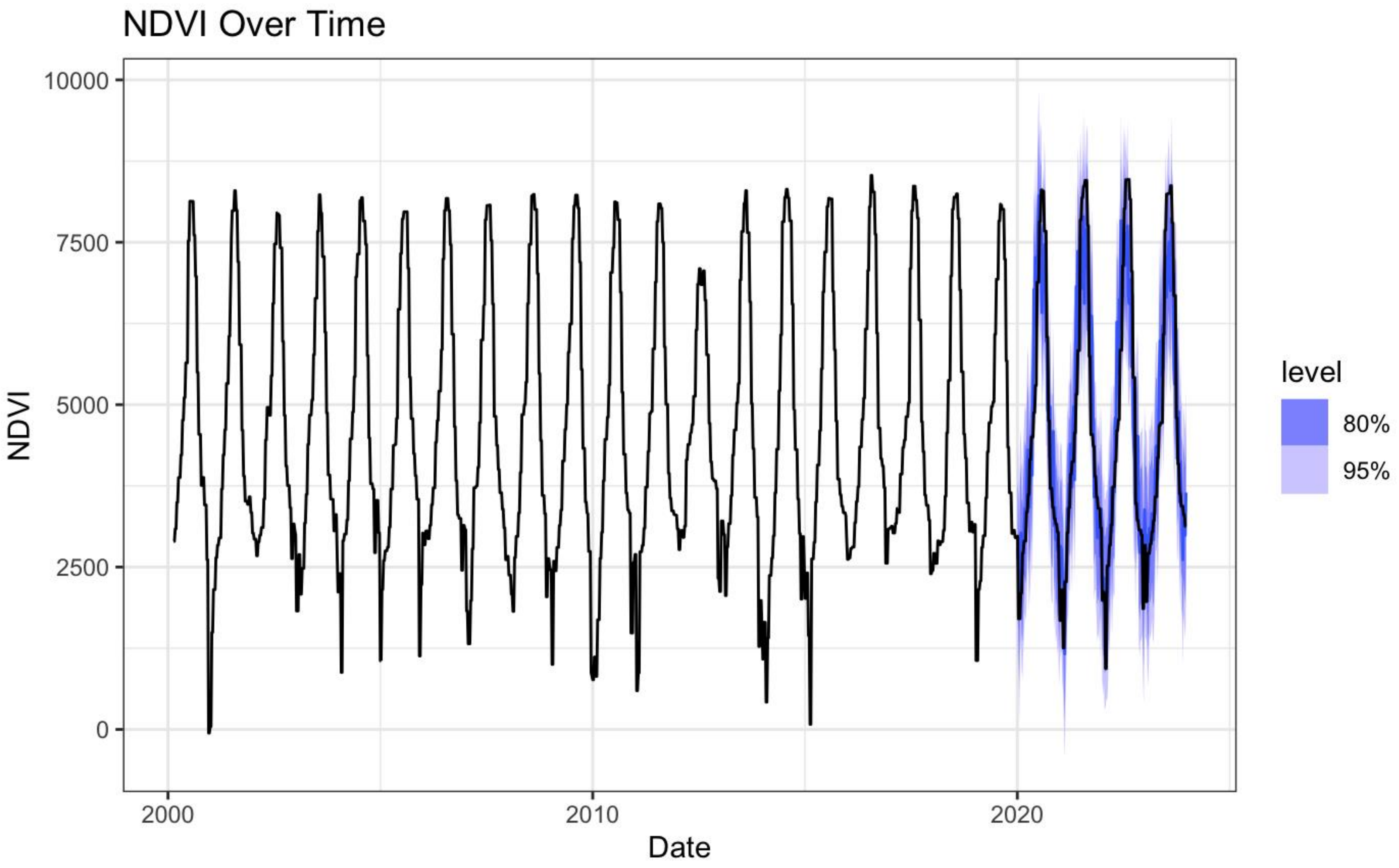
Filter

	date	NDVI	swe	EVI	LAI	prcp	PET	tmax	tmin	vp	FPAR	ET	PLE	dayl	year
1	2000-02-18	2896.30970	2.406506e+01	1485.26821	1.1580127	2.804398120	166.92109	10.9576271	0.62298475	687.5415	7.740970	88.99755	516.84551	38584.12	2000
2	2000-02-25	2896.30970	1.169511e+01	1485.26821	2.1307585	3.235776960	198.97766	15.3579969	3.53793170	821.1993	14.126238	74.74936	616.47492	39660.58	2000
3	2000-03-03	3088.81233	6.360253e+00	1670.27977	3.2218412	0.265447582	248.91784	18.2818345	2.53921192	751.1127	20.859638	64.59298	770.36094	40768.75	2000
4	2000-03-10	3088.81233	3.036499e+00	1670.27977	2.9247518	2.239660289	228.95670	10.3255311	-1.86208597	535.7038	18.620513	75.57951	708.89910	41897.46	2000
5	2000-03-17	3496.38155	1.778706e+00	1923.04278	3.5355389	3.649601825	249.87981	11.3417067	1.85614702	712.4573	21.576501	83.92383	773.28448	43036.75	2000
6	2000-03-24	3496.38155	3.346059e-01	1923.04278	4.5240166	1.105931257	314.07506	16.6702755	2.64017126	746.7513	26.615206	78.12142	971.18071	44177.34	2000
7	2000-03-31	3877.53019	4.654858e-02	2161.71640	4.9234406	0.574266015	315.99465	16.7879134	2.88645156	737.4229	26.480712	76.71338	978.48357	45310.05	2000
8	2000-04-07	3877.53019	2.138790e-04	2161.71640	5.5512963	2.178058838	315.03263	13.6820261	0.71771295	635.0285	27.534031	98.49374	972.95048	46425.18	2000
9	2000-04-14	3877.53019	0.000000e+00	2161.71640	5.5512963	6.426472937	315.03263	19.2053338	6.74465252	953.8332	27.534031	98.49374	972.95048	47511.94	2000
10	2000-04-21	4224.10848	0.000000e+00	2406.29228	5.8785902	2.086088690	371.93544	18.5615462	4.33897382	824.4788	28.153176	89.72835	1145.97223	48557.98	2000
11	2000-04-28	4224.10848	0.000000e+00	2406.29228	6.8048743	1.047790930	433.24339	23.5893910	8.76248958	1120.6192	31.273950	97.08368	1331.52647	49549.05	2000

Phase 3: Analysis Results



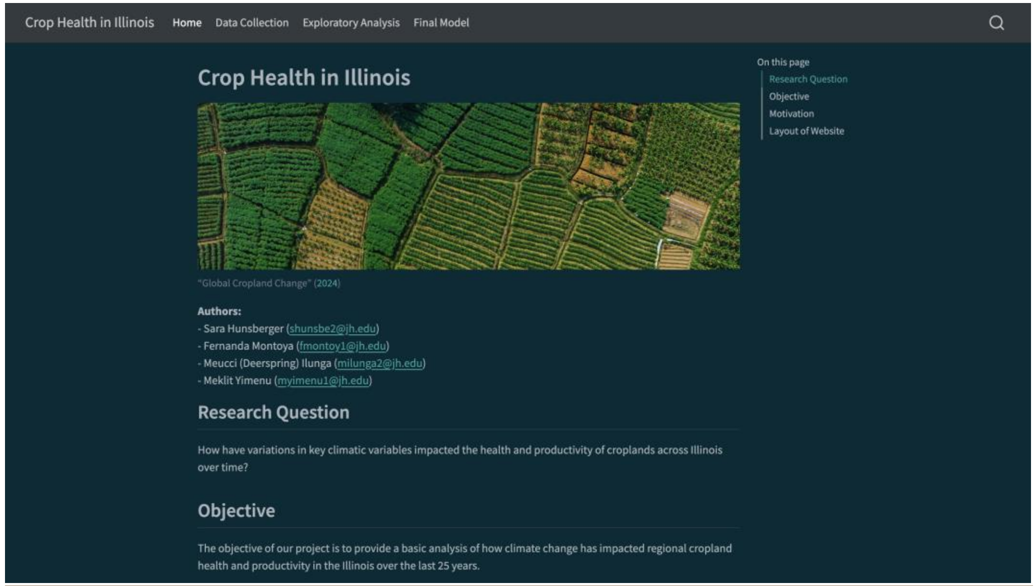
Phase 3: Analysis Results



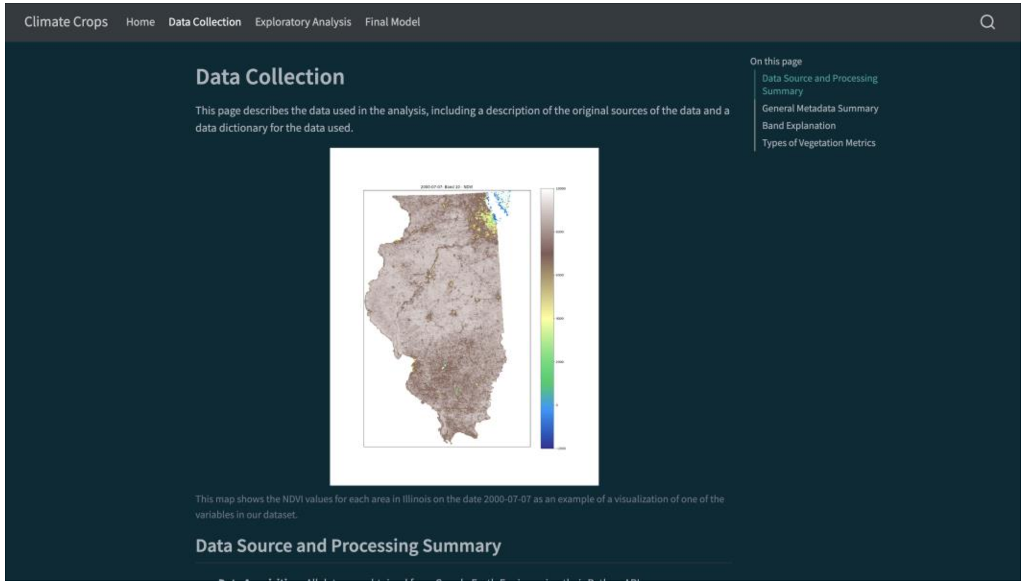
Website

- Created a website to explain our analysis and findings.
- Short demo of the website: WEBSITE

Website

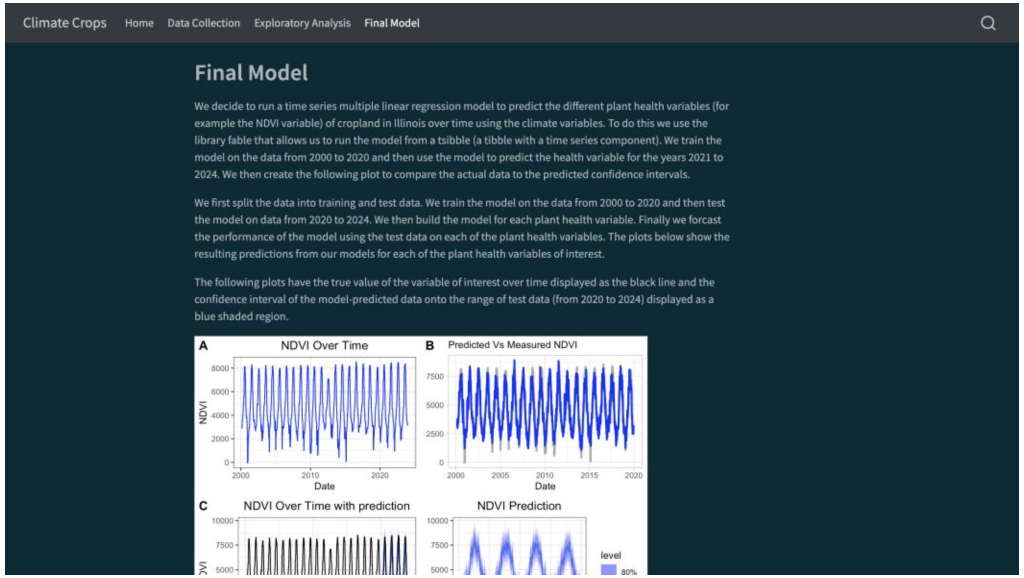


Crop Health in Illinois
Home
Data Collection
Exploratory Analysis
Final Model
Crop Health in Illinois
"Global Cropland Change" (2024)
Authors:
- Sara Hunsberger (shunsbe2@jh.edu)
- Fernanda Montoya (fmontoy1@jh.edu)
- Meucci (Deerspring) Ilunga (milunga2@jh.edu)
- Meklit Yimenu (myimenu1@jh.edu)
Research Question
How have variations in key climatic variables impacted the health and productivity of croplands across Illinois over time?
Objective
The objective of our project is to provide a basic analysis of how climate change has impacted regional cropland health and productivity in the Illinois over the last 25 years.
On this page
Research Question
Objective
Motivation
Layout of Website



Climate Crops
Home
Data Collection
Exploratory Analysis
Final Model
Data Collection
This page describes the data used in the analysis, including a description of the original sources of the data and a data dictionary for the data used.
This map shows the NDVI values for each area in Illinois on the date 2000-07-07 as an example of a visualization of one of the variables in our dataset.
Data Source and Processing Summary
On this page
Data Source and Processing Summary
General Metadata Summary
Band Explanation
Types of Vegetation Metrics

Climate Crops
Home
Data Collection
Exploratory Analysis
Final Model
Summary plots
We first explore the NDVI variable that tells us how well crops are doing and compare the health of crops over time, and across different variables.
As shown in Figure 1, the temperature in Illinois fluctuates seasonally in the same way between the years 2000 and 2023.
We now plot the NDVI and EVI variables over time, we can see a cyclic pattern, which have one peak and valley each year, this makes sense because the seasons likely strongly affect NDVI and EVI, so we expect to see this cyclic pattern.
Figure 2: Illinois Vegetation Indices Over Time
8000
6000
4000
2000
0
2000
2010
2020
EVI
NDVI
On this page
Summary plots
Feature Selection



Climate Crops
Home
Data Collection
Exploratory Analysis
Final Model
Final Model
We decide to run a time series multiple linear regression model to predict the different plant health variables (for example the NDVI variable) of cropland in Illinois over time using the climate variables. To do this we use the library fable that allows us to run the model from a tsibble (a tibble with a time series component). We train the model on the data from 2000 to 2020 and then use the model to predict the health variable for the years 2021 to 2024. We then create the following plot to compare the actual data to the predicted confidence intervals.
We first split the data into training and test data. We train the model on the data from 2000 to 2020 and then test the model on data from 2020 to 2024. We then build the model for each plant health variable. Finally we forcast the performance of the model using the test data on each of the plant health variables. The plots below show the resulting predictions from our models for each of the plant health variables of interest.
The following plots have the true value of the variable of interest over time displayed as the black line and the confidence interval of the model-predicted data onto the range of test data (from 2020 to 2024) displayed as a blue shaded region.
A
NDVI Over Time
NDVI
Date
B
Predicted Vs Measured NDVI
Date
C
NDVI Over Time with prediction
NDVI Prediction
level
80%