Using Landsat Phenology Curves to Characterize Post-Fire Forest Responses in South Carolina, USA

A Thesis Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Miranda Brooke Rose
August 2019

Abstract

Characterizing forest responses to disturbance over large geographic areas represents one of the most challenging aspects of ecosystem monitoring. Traditional remote sensing methods often assess annual or biennial forest change after a disturbance, selecting one image for every year or two years for the study period. However, by using multiple images per year, researchers can examine intra-annual vegetation patterns, or phenology. Phenology provides information on the timing of vegetation events, such as the onset of greenness and the amplitude of NDVI, which can then be used to classify vegetation communities and characterize land cover change over time. Using all available images collected by Landsat 5, 7, and 8 for the study area in South Carolina, I compared intra-annual fluctuations in various spectral indices in pre- and post-fire Landsat pixels, using nearby unburned pixels as an approximate control group, at varying levels of fire severity. Additionally, this research provides baseline pre- and post-fire phenology estimates for the two dominant forest groups in the study region, loblolly-shortleaf pine and oak-gum-cypress. The methods I developed take advantage of the freely available Landsat archive and can be used to characterize forest recovery following a variety of disturbances in the southeastern U.S. and other regions. Future research could examine the feasibility of using phenology metrics to develop predictive species maps at various timesteps following fire events in the region and to develop successional models for this landscape. Hopefully, this research will add to our understanding of how forests are responding to and recovering from fire in a human-impacted region of the U.S.

Table of Contents

List of Tables

List of Figures

List of Attachments

1. "Phenology sample dataset": tidy_all_data_grouped_phenology.csv, the primary data frame of sample locations used for all data analysis. This data frame includes in patch ID, time, Landsat image date, fire ID, fire type, fire acreas, fire date, burn indicator 1, burn indicator 2, number of days between Landsat observation and fire, forest group, burn status of sample, fire severity, Landsat sensor, NDVI, integrated forest z-score, EVI, and NBR. This file is "grouped_data" in the R code presented in the Appendix.

1 Introduction

One of the most challenging aspects of ecological monitoring over large regions is quantifying and characterizing post-disturbance forest recovery. Forest recovery is a complex process, and forests often have multiple potential successional pathways following a disturbance, depending on disturbance type, pre-disturbance species composition, and landscape topography (Swanson et al., 2011). Additionally, forest disturbances are infrequent, impacting less than 2% of U.S. forestlands per year from 1985-2005 (Masek et al., 2013), which limits the data available for forest recovery assessments. However, patterns and rates of post-disturbance forest recovery are key indicators of forest health and resilience (Lebrija-trejos and Bongers, 2016) and understanding how forests are recovering after disturbances is essential for holistic ecological monitoring.

Images collected by earth-orbiting satellites are powerful tools for understanding ecological change over time and across various geographic scales. These images allow researchers to quantify on-the-ground patterns in a variety of ecosystems, including forests. Remote sensing has been extensively used to better understand multiple types of forest disturbance, including harvest (Cohen et al., 2002), insects (Senf et al., 2015), fire (Lentile et al., 2006). However, characterizing rates of recovery and forest composition following a disturbance with remotely sensed imagery remains challenging. Forest composition is determined by a myriad of factors, including soil composition (van Breemen et al., 2011), invasive species (Hejda et al., 2009), and past land use(Hermy and Verheyen, 2007). Furthermore, changes in forest composition after a disturbance are often more nuanced than can be detected from traditional two-image change detection methods. When assessed annually, common vegetation indices derived from remotely-sensed imagery do not always represent ecological definitions of forest recovery (Buma, 2012; Huang et al., 2010). Vegetation indices collected from a single image during peak growing season do not provide information regarding species composition or the “quality” of forest recovery.

With the increasing availability of free moderate resolution satellite imagery, like Landsat, intra-annual change analyses of vegetation are possible. Assessing intra-annual vegetation patterns, or phenology, allows researchers to compare the intra-annual patterns of vegetation communities, predict vegetation type, and measure forest health. A variety of statistical methods have been developed to construct reliable phenology signals from remote sensing data, including Fourier series-based harmonic models (Bradley et al., 2007; Hermance, 2007) and logistic regression (Melaas et al., 2013; Zhang et al., 2003). Phenological metrics derived from these curves have also been used to characterize different vegetation types in the Sonoran Desert (van Leeuwen et al., 2010).

Forests of the southeastern U.S. have been heavily impacted by humans, through timber harvest and urban expansion (Drummond and Loveland, 2010), making the region an excellent case study to understand how forests are responding to disturbances in novel ecosystems using Landsat-derived phenology curves. The current research compares pre- and post-fire phenology patterns in two forest groups in South Carolina using all available Landsat imagery for the study area.

2 Background

2.1 Forest disturbance and recovery in the southeastern U.S.

Although incidents of forest disturbance are rare nationally, forest disturbance rates in the southeastern U.S. are among the highest in the country (Masek et al., 2008), making the Southeast an excellent region for developing methods for forest recovery assessments. Timber harvest is the dominant disturbance mechanism throughout the Southeast (Masek et al., 2008); however, it is difficult to detect and quantify with publicly available datasets.

Though less widespread than timber harvest, fire is an important disturbance agent in southern forests and has historically shaped the structure and species composition of forest ecosystems in the region, such as longleaf pine-grassland communities (Waldrop et al., 1992). Fire suppression policies of the 20th century fundamentally changed forest composition throughout the eastern U.S. Fire-adapted oak-pine forests transitioned to fire-intolerant forests composed of maple, cherry, and hemlock (Nowacki and Abrams, 2008). Prescribed burning in loblolly, shortleaf, slash, and longleaf pine stands has become a popular tool for restoring pine-savanna ecosystems (South Carolina Forestry Commission, 2010). However, the reintroduction of fire after an extended period of fire suppression can impact forest structure and soil composition in unexpected ways, including unintended tree mortality and root damage (Varner et al., 2005). Large-scale remote sensing assessments of post-fire forest recovery are essential for monitoring the impact of recent prescribed fires and wildfires on forest composition after a century of widespread fire suppression in the area.

Loblolly-shortleaf pine and oak-gum-cypress are two of the primary forest groups in the study area in South Carolina, according to predictive maps produced by the U.S. Forest Service (Ruefenacht et al., 2008). Ecologically, these forest groups occupy different environments. As the dominant forest group in the state, loblolly-shortleaf pine occupied 5.3 million acres as of 2006 (South Carolina Forestry Commission, 2010). Loblolly pine (*Pinus taeda*), a fast-growing timber species, has largely replaced the once widespread longleaf pine (*Pinus palustris*) in South Carolina. Shortleaf pine (*Pinus echinata*) is also widespread throughout the southeastern U.S., second only to loblolly pine in total timber volume. The oak-gum-cypress forest group occupies low-lying wetland areas of the state and covers a much smaller proportion of forested area (19%) than loblolly shortleaf pine (40%) (South Carolina Forestry Commission, 2010). This heterogenous forest group includes 15 tree species: Atlantic white-cedar (*Chamaecyparis thyoides*), bald cypress (*Taxodium distichum*), pond cypress (*Taxodium ascendens*), red maple (*acer rubrum*), water hickory (*Carya aquatica*), water locust (*Gleditsia aquatica*), sweetgum (*Liquidambar*), water tupelo (*Nyssa aquatica*), Ogeechee tupelo (*Nyssa ogeche*), swamp tupelo (*Nyssa biflora*), overcup oak (*Quercus lyrata*), swamp chestnut oak (*Quercus michauxii*), nuttall oak (*Quercus texana*), willow oak (*Quercus phellos*), and American elm (*Ulmus americana*). Because loblolly-shortleaf pine is evergreen, the phenology curves for this forest group are expected to have a lower seasonal amplitude than the mixed oak-gum-cypress.

2.2 Remote sensing of forest disturbance and recovery

With increasing data availability and computing capabilities, remote sensing has become one of the most important tools for analyzing ecological change at various spatial and temporal scales. Data collected by the Landsat satellites are among the most widely used remote sensing data in ecology and environmental science. Beginning with the launch of Landsat 1 in 1972, Landsat imagery makes up the longest running series of satellite-collected data. With the successive launches of Landsat 1-8, improvements in spatial and spectral resolution have occurred. These improvements allow for more sophisticated assessments of on-the-ground forest dynamics, including land cover, species, and successional stage classifications (Hall et al., 1991).

Vegetation indices derived from remote sensing imagery have been used to assess forest recovery following fire across various scales (Buma, 2012; Idris et al., 2005; van Leeuwen, 2008). Common indices used to assess vegetation change include normalized difference vegetation index (NDVI), normalized burn ratio (NBR), enhanced vegetation index (EVI), and forestness index (Table 2.1). NDVI takes advantage of healthy vegetation's tendency to reflect light in the near infrared portion of the electromagnetic spectrum (Reed et al., 2006). Similarly, dry or burned vegetation shows higher reflectance in the shortwave infrared region. The difference in NBR is frequently used to identify burned areas and estimate fire severity, based on the relationship between reflectance in the near-infrared and the shortwave infrared regions of the electromagnetic spectrum(Key and Benson, 2006). The enhanced vegetation index (EVI) is designed to reduce atmospheric and background noise while also being more sensitive to structural changes in vegetation canopy than NDVI (Huete et al., 2002). Huang et al. (2010) developed the integrated forest z-score, commonly referred to as a "forestness index" as an inverse measure of a pixel's likelihood of being forested. The current research will focus on NDVI and NBR phenology curves, though future research could examine the feasibility of constructing phenology curves from other spectral indices to assess forest recovery.

Common methods currently used to detect forest disturbance and recovery rely on spectral indices derived from annual or biennial Landsat time series stacks (LTSS). For example, Masek et al. (2008) used a decade of Landsat imagery to map forest disturbance in North America using annual and biennial image comparison (Figure 2.1).

Table 2.1. Common spectral indices used to quantify vegetation

Index	Equation	Typical Use
Normalized difference vegetation index (NDVI)	$\frac{NIR - R}{NIR + R}$	Measures vegetation "greenness" (Reed et al., 2006)
Normalized burn ratio (NBR)	$\frac{NIR - SWIR}{NIR + SWIR}$	Identify burned areas and estimate fire severity (Key and Benson, 2006)
Enhanced vegetation index (EVI)	$EVI = G * ((NIR - R)/(NIR + C1 * R - C2 * B + L))$	Sensitive to canopy structural changes, reduces atmospheric and background noise (Huete et al., 2002)
Integrated Forest Z-Score (Forestness Index)	$\sqrt{\frac{1}{NB}\sum_{i=1}^{NB}(FZ_i)^2}$	Inverse measure of the likelihood that a given pixel is forest (Huang et al., 2010)

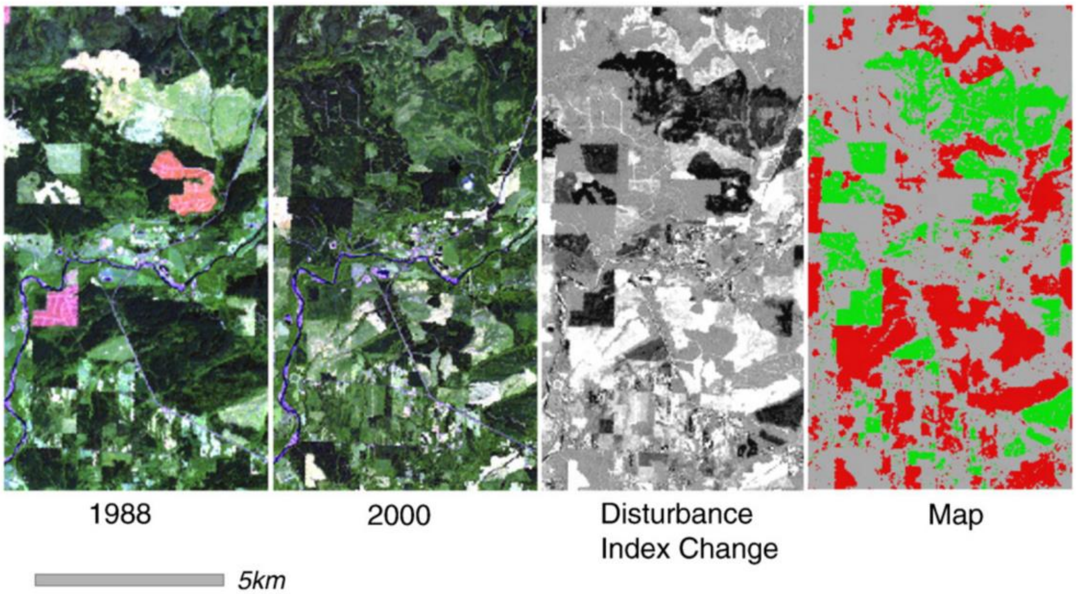


Figure 2.1. Disturbance mapping based on Landsat RGB image-to-image comparison. (Source: Masek et al., 2008). In the final map, green indicates areas of forest regrowth while red represents disturbance. While useful, this approach does not give information about the type of vegetation returning after a disturbance.

In this method, images selected from two different years are chosen, compared, and a disturbance index is used to distinguish between areas of regrowth and disturbance. In image-to-image change analysis, images are often selected from the growing season to reduce the phenological variability between observations and must be free from cloud cover.

Vegetation Change Tracker (VCT) is a popular algorithm that uses spectral-temporal information derived from a LTSS to reconstruct forest disturbance history and track post-disturbance recovery processes in a given study area (Huang et al., 2010). This algorithm is appealing because it requires little user input and can be used across large areas to characterize disturbance and recovery. Similarly, LandTrendr (Landsat-based detection of Trends in Disturbance and Recovery) is a segmentation algorithm that models a Landsat pixel's spectral trajectory over time based on a given yearly LTSS in conjunction with TimeSync, an image visualization and collection tool (Cohen et al., 2010; Kennedy et al., 2010). Another method that was developed as a part of the North American Forest Dynamics project (Goward et al., 2008; Masek et al., 2013), is a shape selection algorithm that fits Landsat spectral trajectories to a specific "shape" based on the pixel's temporal patterns (Moisen et al., 2016). The shapes correspond to potential ecological forest states and include stable or growing forests, rapid disturbance events, slow disturbances.

These methods generally define forest regrowth based on pre-disturbance spectral index thresholds or examine the rate of regrowth as a proxy for forest recovery. Researchers who examined forest disturbance and regrowth in the Pacific Northwest using LandTrendr and TimeSat acknowledge that similar spectral index values before and after a disturbance do not necessarily reflect identical vegetation conditions and that a single spectral index is not sufficient to characterize forest recovery (Kennedy et al., 2012). While these methods are useful for detecting forest disturbance and recovery over large areas, they provide limited information regarding the composition and health of vegetation after a disturbance.

2.3 Assessing intra-annual vs. inter-annual change

Although annual and biennial LTSS methods are generally accurate for disturbance detection and characterization, inferring forest recovery processes using these methods remains challenging. Previous research using VCT and other algorithms across a variety of forest ecosystems, in the U.S. and other regions, indicates that recovery in a spectral index occurs quickly after a disturbance and does not necessarily reflect forest recovery from an ecological perspective (Griffiths et al., 2014; Huang et al., 2010). Similarly, a study conducted in Colorado on post-disturbance forest dynamics demonstrated that Moderate Resolution Imaging Spectroradiometer (MODIS)-derived normalized difference vegetation index (NDVI) was poorly correlated with forest recovery, and high NDVI values in the years following a disturbance were correlated with perennial herbaceous growth, not new tree growth (Buma, 2012).

Assessing intra-annual changes in spectral indices provides an alternative method for examining post-disturbance forest recovery using remote sensing data. Phenology derived remote sensing data can be used to estimate the timing of various vegetation growth events, including length of growing season and onset of greenness in the spring, which are affected by seasonal environmental factors, such as temperature and precipitation, as well as disturbance events, such as fire or land use change (White et al., 1997). With the increasing availability of remote sensing imagery, intra-annual analyses are now possible and allow researchers to use phenology as another remotely-sensed metric to better understand on-the-ground vegetation dynamics. MODIS-derived phenological metrics, including onset of greenness and senescence, were used to characterize post-fire vegetation dynamics and recovery based on differing levels of fire severity and pre-fire fuel treatments in Arizona (van Leeuwen, 2008). Phenology can also be related to land cover classes, as different surfaces have variable annual reflectance patterns. Bradley and Mustard (2008) used Advanced Very-High-Resolution Radiometer (AVHRR)-derived phenology to classify land cover types in the Great Basin and better understand how different vegetation communities respond to different climatic conditions (Figure 2.2).

Due to its relatively coarse temporal resolution (16-day return interval) and previous cost (up to $4,400/scene in the 1980's) – see Reichhardt, Landsat data have not been traditionally used for phenology or intra-annual change research. However, now that all Landsat images are free to download, methods that take advantage of the entire Landsat archive are emerging. Landsat EVI has been successfully used to detect year-to-year phenology changes in the Harvard Forest using a logistic modeling approach (Melaas et al., 2013). The Continuous Change Detection and Classification (CCDC) algorithm uses a sinusoidal approach to detect and classify land cover based on inter-annual and intra-annual change in a given sample pixel (Figure 2.3) (Zhu and Woodcock, 2014). Using this framework, researchers are able to use all available clear pixels in an image as opposed to only completely clear images, which are rare.

The ability to classify vegetation based on remotely-sensed phenological metrics makes it possible to infer or predict vegetation type or community in an area following a disturbance. The current research examines the utility of Landsat-derived NDVI and NBR phenology curves to characterize pre- and post-fire forests for the two dominant forest groups in South Carolina, loblolly-shortleaf pine and oak-gum-cypress.

2.4 Why use phenology to study forest recovery?

An important part of understanding forest recovery is knowing the type of vegetation that returns after a disturbance. Because different types of vegetation often have distinct phenological

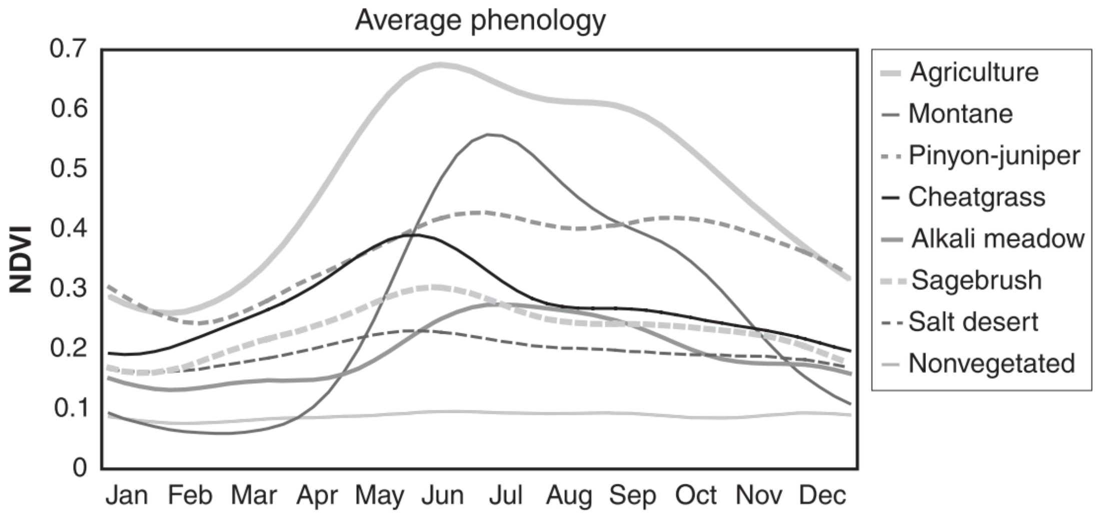


Figure 2.2. Average annual AVHRR NDVI phenology for common land cover types in the Great Basin (Source: Bradley and Mustard, 2008)

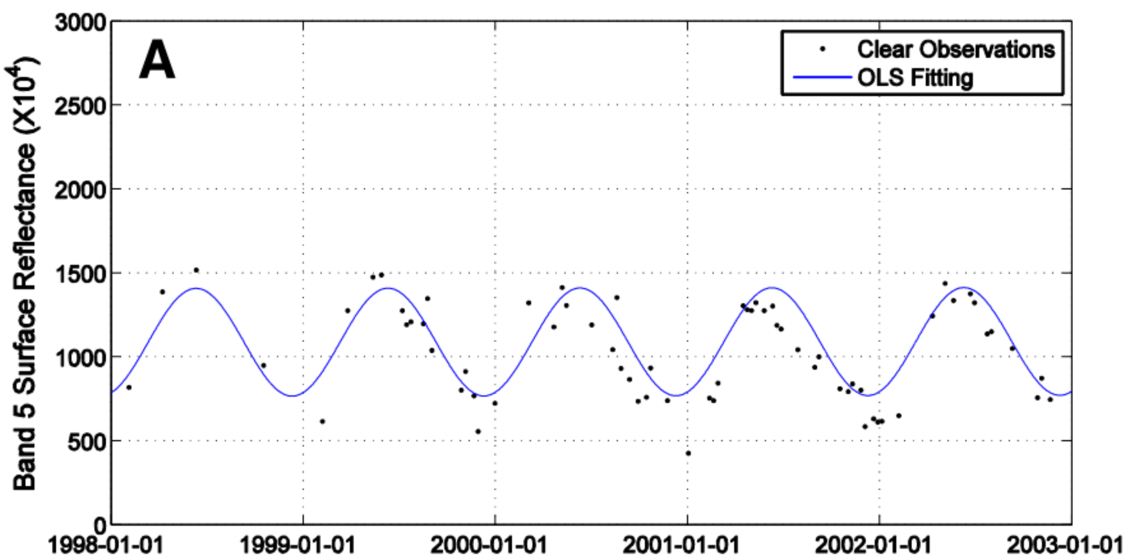


Figure 2.3. Example of Continuous Change Detection and Classification (CCDC) algorithm modeling the inter-annual and intra-annual patterns of a Landsat pixel. The model's coefficients are based on ordinary least squares (OLS) fitting.

signatures (Figure 2.2), phenology curves and phenological metrics can be used as proxies for vegetation type.

Geerken (2009) developed an algorithm to classify MODIS pixels based on the unique intra-annual NDVI shapes of different vegetation communities. Assessing forest response to fire gives insight into the successional patterns of disturbed landscapes and can also provide indicators of forest health. Different vegetation groups have variable responses to seasonal fluctuations in temperature and precipitation, and these responses can be detected using optical remote sensing imagery and then can be used to distinguish between different vegetation types.

Data exploration revealed differences in the NDVI phenology patterns between the forest groups in the study area (Figure 2.4). These Landsat-derived NDVI observations demonstrate discernable differences in the intra-annual patterns for the primary forest groups, loblolly-shortleaf pine and oak-gum-cypress. As expected for evergreen forests that do not lose their leaves in the fall and winter, the loblolly-shortleaf pine has a “flatter”, less variable, annual NDVI pattern. On the other hand, the oak-gum cypress NDVI phenology has a greater amplitude, with lower NDVI values in the fall and winter and elevated NDVI values in the summer.

These preliminary findings are consistent with my assumptions about the forest groups in the study area. If phenology can serve as a remotely-sensed proxy for forest group, we can potentially infer forest composition following a fire or other disturbance based on these metrics. Fires in the southeastern U.S. usually only affect understorey vegetation (Stanturf et al., 2002), making them difficult to detect using remote sensing imagery. The Monitoring Trends in Burn Severity dataset used in this study includes both low severity and prescribed fires, which are designed to reduce undergrowth and promote timber production. Therefore, I expect the impacts of fire on the phenology curves to be small.

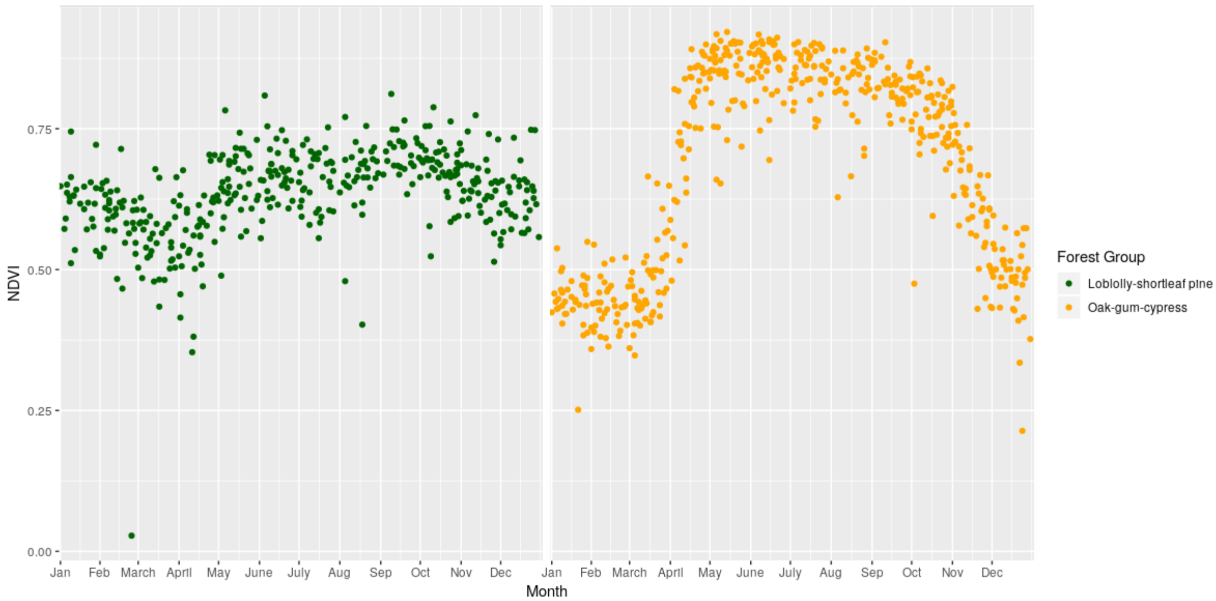


Figure 2.4. Multi-year Landsat observations for two 9-pixel patches in the study area's dominant forest groups. Note the flatter curve for the loblolly-shortleaf pine (evergreen) and the greater amplitude for the oak-gum-cypress (deciduous).

2.5 Research objectives

This work addresses the research question *can Landsat-derived phenology improve our estimation of forest recovery following disturbance*? Hypotheses tested include 1.) The annual Landsat NDVI and NBR phenology in loblolly-shortleaf pine are different than oak-gum-cypress. 2.) Post-fire phenology curves are different than pre-fire curves and differences will vary based on forest group, fire severity, and time since fire. 3.) Pre-fire phenology curves in the burned samples will be similar to phenology curves constructed in unburned control samples; however, post-fire phenology curves will be different in the burned and unburned samples.

Using harmonic regression and spline smoothing, I developed baseline phenology curves for burned and unburned loblolly-shortleaf pine and oak-gum-cypress forests for the study region in South Carolina from the available Landsat data archive (1984-2016). For burned samples, I constructed NDVI and NBR pre- and post-fire models for the two species groups for various degrees of burn severity (low, medium, and high) and also for two time steps following fire (within 2 years vs. more than 2 years). The methods and algorithms developed in this thesis can be customized and applied to other types of disturbance in the southeastern U.S. as well as other forested regions with extensive Landsat coverage.

2.6 Addressing noisy data and other challenges

Landsat data are subject to various types of noise due to atmospheric conditions, sensor issues, ground conditions, and other factors that do not reflect the condition of the vegetation being recorded. When analyzing phenology data, methods such as temporal averaging are sometimes used to reduce noise and error due to satellite inconsistencies (DeFries et al., 1995; Justice et al., 1985). However, Landsat's coarse temporal resolution makes temporal averaging unfeasible for phenological studies. Alternative methods have been applied to intra-annual phenology curves to reduce noise inherent to satellite data. For this research, I selected a non-parametric curve-fitting or spline procedure that uses a least squares approach that handles data gaps and outliers while also preserving essential phenology markers, such as onset of greenness, to generate an average annual phenology (Bradley et al., 2007). While these methods have not, to my knowledge, been used to study forests using Landsat data, they have been applied to AVHRR and MODIS data.

Landsat pixel-based estimates of phenology are currently too noisy for constructing reliable, average intra-annual phenology curves (Figure 2.5). To account for the noisiness inherent in any remotely sensed data, I used a simplified, local form of low-pass spatial filtering for the Landsat NDVI and NBR data for my sample pixels. Typically, spatial filtering is used to enhance remote sensing images, often through high pass filtering, or reducing noisy remote sensing data, through low pass filtering. Instead of using a moving kernel window, I averaged NDVI and NBR values across 3 x 3 Landsat pixel areas, referred to as patches, generating spectral index values with a spatial resolution of 90m x 90m, or nine times the size of a Landsat pixel.

Averaging spectral values across 9-pixel patches generates more stable intra-annual patterns that are less vulnerable to noisy single-pixel values. The 9-pixel patches also include denser temporal data, which is essential for estimating vegetation phenology. For example, if a single pixel is cloudy on a particular image date, the value for that pixel is excluded and the data for that day is lost. However, for a patch, the NDVI for the 8 clear pixels can still be averaged and retained in the dataset. In this case, the center pixels for patch 971 had a clear observation for 307 Julian days for the study period, while at the patch level, there were clear observations for 328 Julian days.

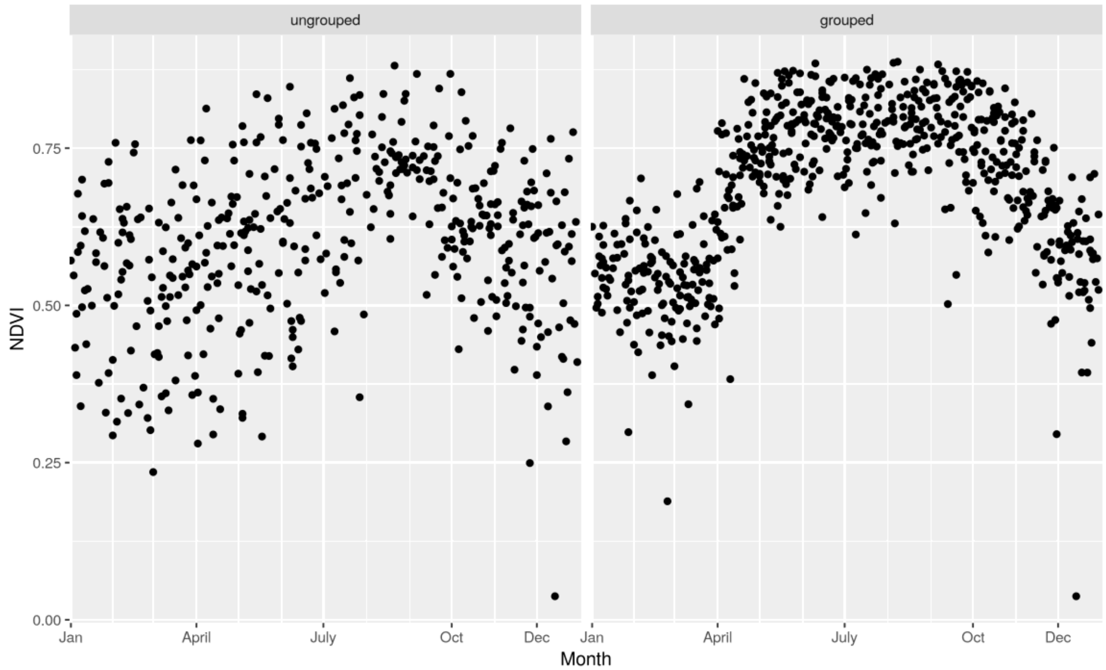


Figure 2.5. Multi-year Landsat NDVI values for the center pixel (left) and the aggregated Landsat NDVI values at the patch-level for patch 971. Note the noisy distribution of the pixel-level NDVI values, which obscures the phenology signal.

Small increases in temporal resolution are beneficial when constructing phenology estimates from remote sensing data. Therefore, the models presented in this research are based on Landsat patch-level averages of NDVI and NBR.

To determine the response of these forests to fire, a baseline or control phenology of similar forests in the absence of fire is needed. However, establishing treatment “disturbed” samples and control “undisturbed” samples in this region is problematic, due to high levels of human activity in the region for thousands of years. Even the control, unburned samples have likely experienced some form of disturbance in the last two to three decades. Many of the pixels sampled are located in the Francis Marion National Forest, which encompasses over 629,000 acres. The USDA Forest Service manages the area for timber production and wildlife habitat, carrying out regular prescribed burns. While there were several wild fires during the study period, the vast majority of the fires examined in the current research were low-intensity prescribed burns designed to clear understory shrubs and grasses. These fires, typical of managed southeastern U.S. forestlands, are difficult to detect using remote sensing imagery because they typically do not affect overstory vegetation. Therefore, I expect to observe relatively small changes following fire, particularly for areas controlled by prescribed burning.

Landsat data have been used to derive both the independent variables (Monitoring Trends in Burn Sevierity, U.S. Forest Service forest group maps) and the dependent variables (Landsat-derived phenology signals) for this research, creating another potential problem. Finally, the U.S. Forest Service forest group maps were generated in 2003. For my analysis, I am assuming that this map is appropriate through the time and that no significant change has occurred to shift the forest groups at a large scale throughout the study region. Ideally, in the future the statistical methods developed here to construct phenology estimates could be combined with independent vegetation metrics collected in the field.

3 Data and Methods

3.1 Study site

South Carolina is dominated by loblolly-shortleaf pine with areas of oak-gum-cypress forest in lower-lying terrain. The study area selected corresponds to the Landsat scene path 16 row 37 (Figure 3.1), which encompasses Charlotte, SC as well as Francis Marion National Forest, providing a variety of land cover classifications as well as the forest groups mentioned above. It falls within the Southeastern Plains and Coastal Plains ecoregions and is characterized by low elevation. This area represents one of the four focal areas in the North American Forest Dynamics (NAFD) project, which aims to better understand forest disturbance and carbon stocks in North American forests (Goward et al., 2008; Huang et al., 2009) .

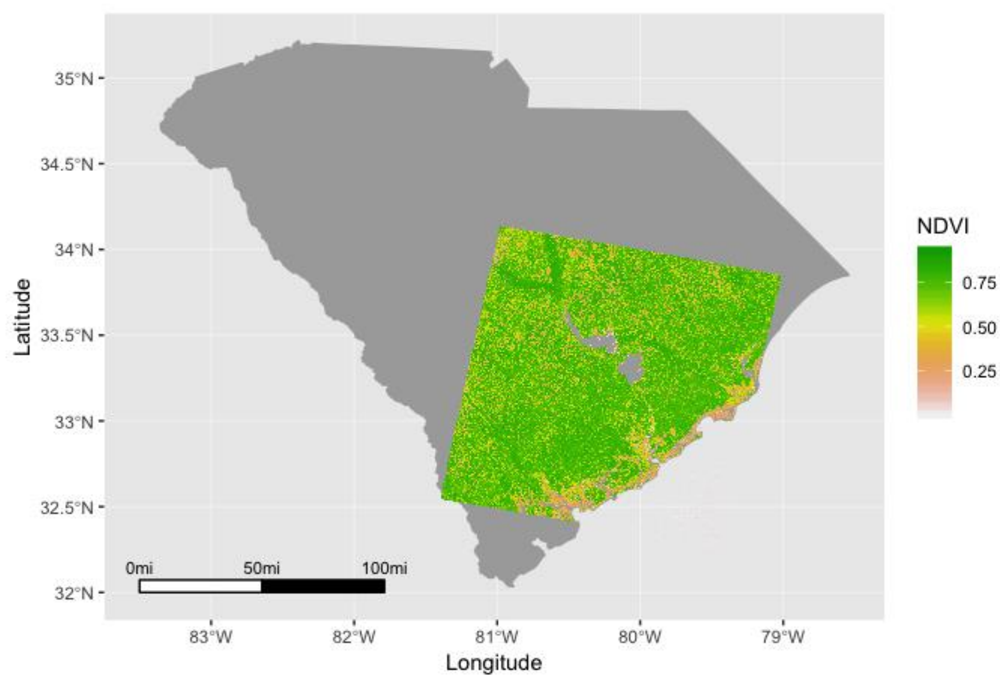


Figure 3.1. Study area, Landsat scene path 16 row 37, normalized difference vegetation index (NDVI) for November 9, 2014.

3.2 Landsat imagery

To construct phenology curves for the study area, I used 660 images collected by Landsat TM, ETM+, and OLI from 1984-2017 for the study area (Figure 3.2). Prior to download from the USGS data archive in November 2017, the Landsat images were atmospherically corrected using the Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS) (Masek et al., 2006; Vermote et al., 1997). Using a random sample function in R and forest group maps produced by the U.S. Forest Service, I generated an equal sample of locations in each of the dominant forest groups, loblolly-shortleaf pine (1000 pixels) and oak-gum-cypress (1000 pixels). These locations and the eight surrounding pixels for each sample pixel were then used to extract band values from every Landsat image in the scene. I used the Quality Assessment (QA) bands included in Landsat Surface Reflectance products to remove pixels affected by instrumental or atmospheric irregularities from the dataset. This method allowed me to retain all available images for the study region, while also removing cloudy and contaminated data.

3.3 Forest group maps

U.S. Forest Service forest group maps, produced in 2003, were used to distinguish between the two dominant forest species groups in the study area (Figure 3.3). These maps were derived from a variety of data, including digital elevation models (DEM), Moderate Resolution Spectroradiometer (MODIS) vegetation indices, National Land Cover Dataset (NLCD), and Forest Inventory and Analysis Data and have a spatial resolution of 250 m (Ruefenacht et al., 2008). According to forest group classifications, loblolly-shortleaf pine and oak-gum cypress represent the dominant forest groups for the study area. These maps were used to differentiate between the two forest groups when generating sample pixels from the burned area data.

.

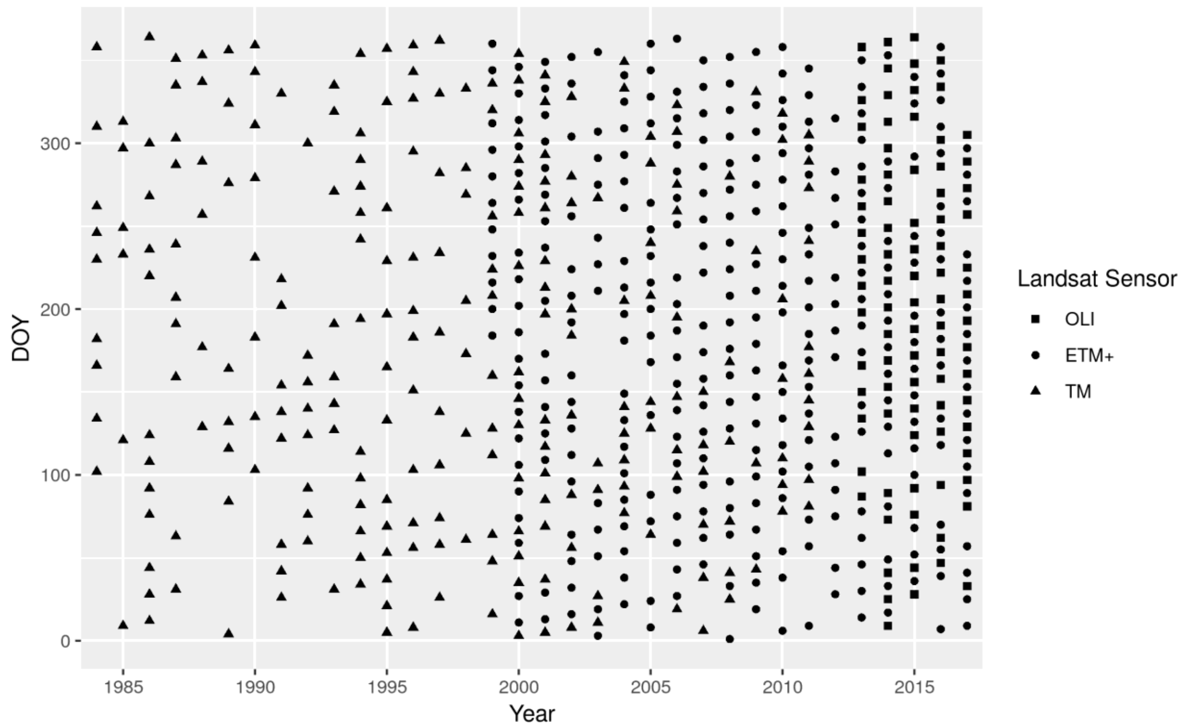


Figure 3.2. Annual distribution and timing of 661 Landsat images used in this analysis. Shapes distinguish between observations collected by the three Landsat sensors: OLI, ETM+, and TM.

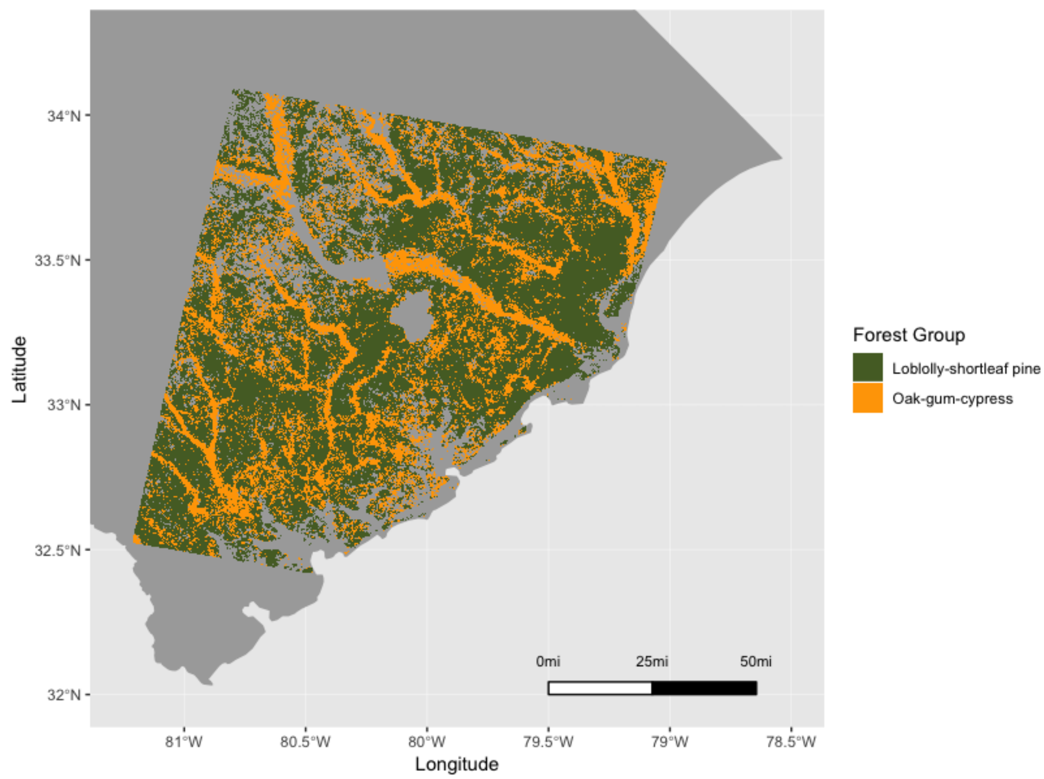


Figure 3.3. U.S. Forest Service forest group map, produced in 2003.

3.4 Fire extent and severity data

To distinguish between burned and unburned pixels within the study area, I used polygons and raster images generated from data collected by the Monitoring Trends in Burn Severity (MTBS) program, sponsored by the Wildland Fire Leadership Council. These data are derived from Landsat TM and ETM+ NBR, a spectral index often used to identify burned vegetation (Eidenshink et al., 2007). MTBS data provide information regarding the extents (Figure 3.4) and severity (Figure 3.5) of fires throughout the U.S. and are freely available to the public. These datasets also include information related to fire identification, fire type, fire extent, fire date, and other variables that were retained through processing and used to model phenology based on different fire metrics. Because the U.S.F.S. forest group maps were generated in 2003 and past species distribution maps are unavailable, I limited my analysis to fires that occurred from 2003 to 2015 to control for changes in forest group composition that occurred prior to the creation of these maps and to ensure that the forest group maps represent pre-fire conditions.

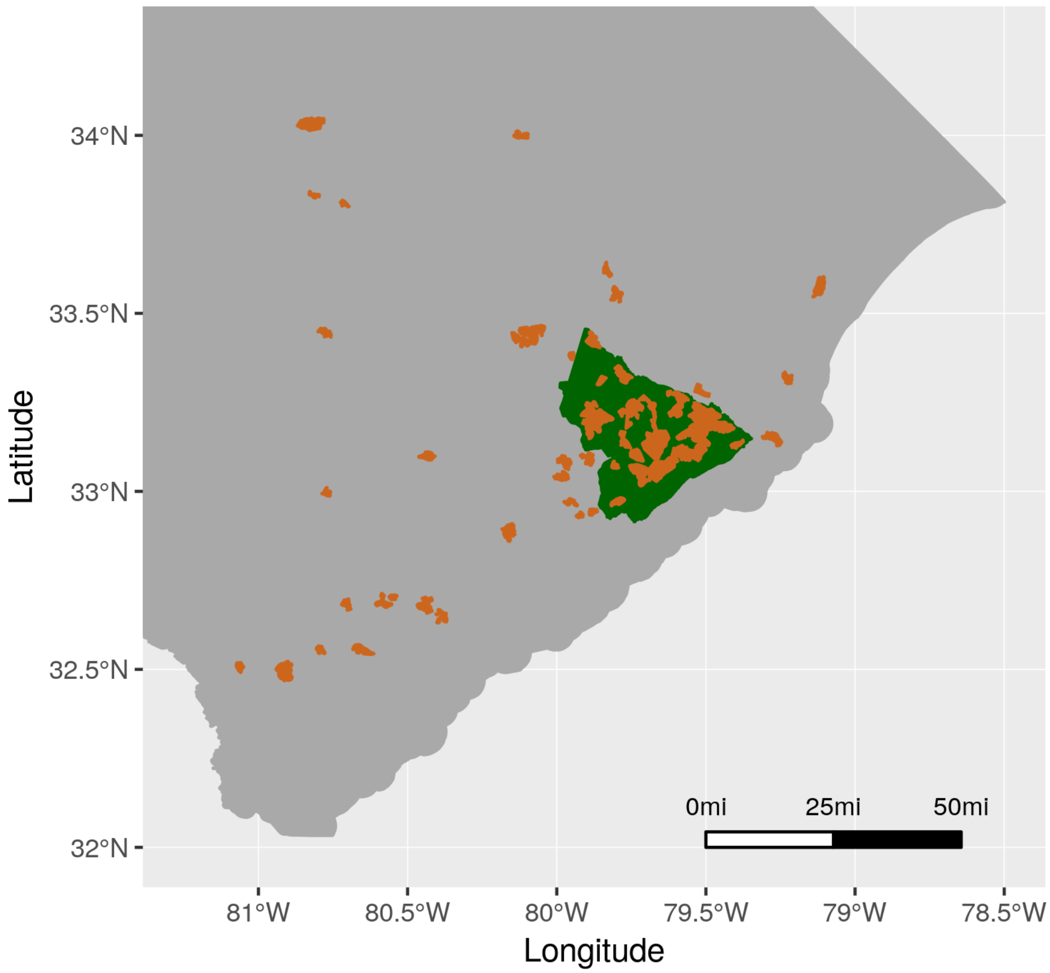


Figure 3.4. Monitoring Trends in Burn Severity fire extents (brown) and Francis Marion National Forest (green). These polygons represent fires from 2003-2015.

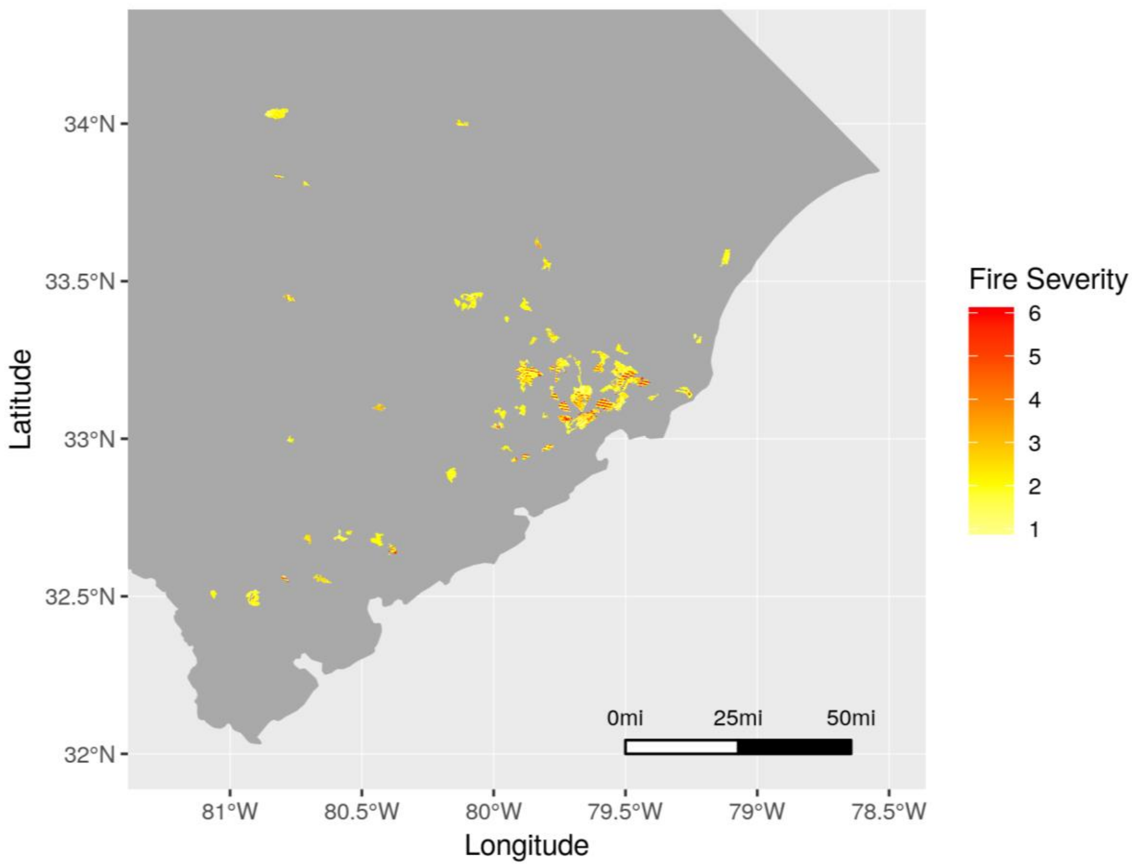


Figure 3.5. Monitoring Trends in Burn Severity raster pixels for fires that occurred from 2003-2015.

3.5 Sample design and data processing

I used a pseudo control-treatment sample design, with the burned forest locations as the treatment and the surrounding unburned forest locations serving as the control. To better understand the differences between pre- and post-fire phenology within the burned samples, I needed to establish a baseline or "normal" set of phenology curves that did not experience fire. This approach assumes that the fires are random and that there are no fundamental environmental differences between the burned and unburned sites. The limitations of this approach are addressed in the discussion. I processed and analyzed the data using R statistical software, specifically the 'tidyverse', 'sf', 'mgcv' and 'raster' packages.

For my burned samples, I removed raster pixels that experienced more than one fire between 1984-2015 using the mask function in R to reduce any potential confounding factors introduced by multiple fires. Using the single-burn pixels from the MTBS raster images from 2003-2015, I sampled an equal number of pixels from the two forest groups, loblolly-shortleaf pine and oak-gum-cypress, for a total of 18,000 burned pixels. For the unburned sample, or control, pixels, I generated 5-kilometer buffers around each of the MTBS polygons and used the burned polygons as a mask to remove any burned areas that overlapped with the buffers. These buffers represent unburned, control areas of similar environmental conditions to the burned samples within the fires (Figure 3.6). I sampled an equal number of pixels from the two forest groups within these buffer areas, for a total of 18,000 unburned pixels. Using the sample pixels' raster row and column information, I included the eight surrounding pixels for each of the original samples to create "patches" for later data aggregation. The processing techniques I used allowed me to retain the original cell ID's and patch ID's for each pixel.

I extracted Landsat image values and quality assurance values at the sample burned and unburned pixel locations using a function created in R. These data were processed to include fire information, forest group, Landsat image date, band number, and Landsat sensor. After calculating NDVI and NBR from the original Landsat values, I averaged the data to the patch level for model construction. To compare phenology curves in the pre-fire, post-fire, and time steps after fire samples, I used a set of indicator variables that corresponded to a patch's burn status on the date of observation. Note that both the burned and unburned samples are designated as either "pre-fire" or "post-fire." For the burned samples, "pre-fire" indicates that the observation occurred before a given fire, while "post-fire" corresponds to observations made after the fire.

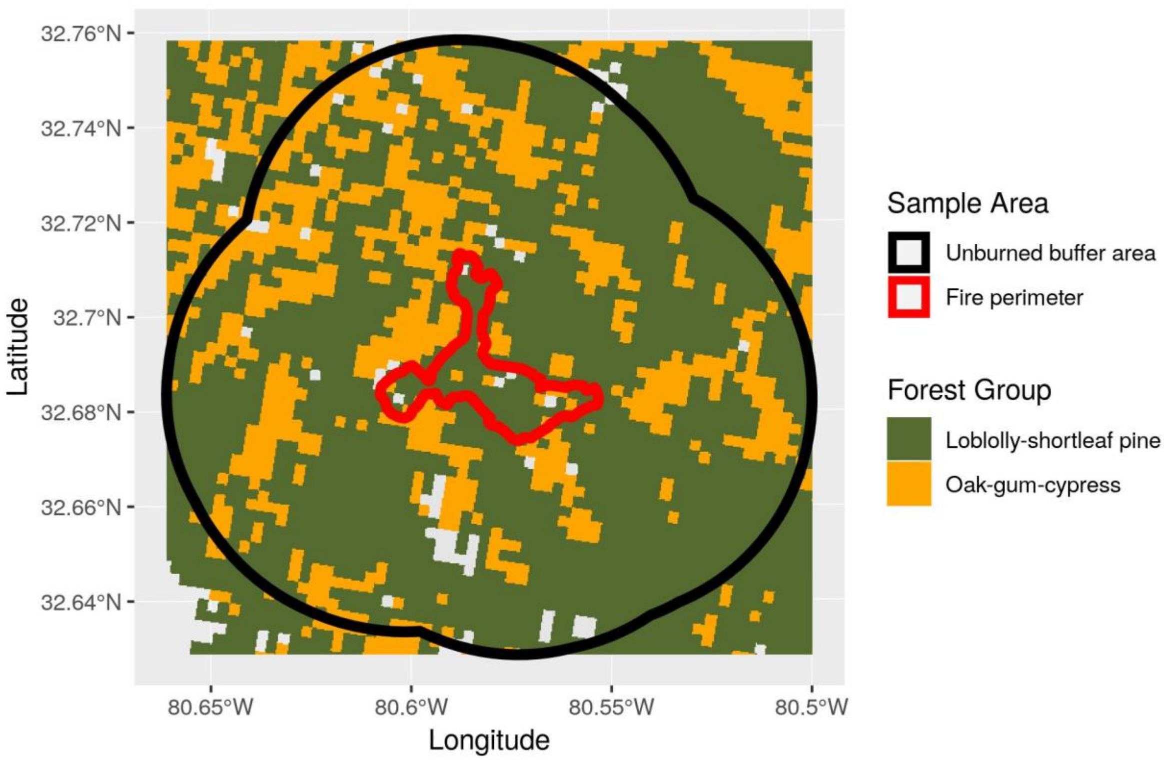


Figure 3.6. Illustration of the control-treatment sampling design. Burned samples were selected from within the fire perimeter (red) and unburned samples were selected from the 5 km buffer region (black). Globally, an equal number of loblolly-shortleaf pine and oak-gum-cypress samples were selected.

Although the unburned samples did not experience fire between 1984-2017, each unburned sample corresponds to an individual fire. Therefore, “pre-fire” unburned samples are observations that were taken from the unburned control area before the corresponding fire occurred. Similarly, “post-fire” unburned samples are all of the observations taken from the unburned control area after a particular fire occurred.

In the framework used to model the pre- and post-fire phenology curves, $S_{pre}(t,\ date_t)$ is the smooth phenology curve before the fire, which is a function of time and Julian Day, $S_{post-pre}(t,\ date_t)$is the change in the phenology curve after the fire, and I() is an indicator variable equal to 0 before fire and 1 for after fire (Equation 3.1).

$$\widehat{NDVI} = S_{pre}(t,\ date_t) + I(t\ after\ fire) * S_{post-pre}(t,\ date_t)$$

Equation 3.1

3.6 Harmonic regression

To model the annual curves of NDVI and NBR derived from Landsat 5 TM, Landsat 7 ETM+, and Landsat 8 OLI pixel samples, I used a non-classical harmonic regression analysis that was developed to construct phenology curves for different land cover types in the Great Basin (Equation 3.2) (Bradley et al., 2007; Hermance, 2007).

$$d_{observation_i} = d_{pred_{i+}} +\ error_i = \sum_{m=0}^{L} c_m t_i^m + \sum_{k=1}^{M} \left(a_k \cos(\frac{2\pi k}{T} t_i) + b_k \sin(\frac{2\pi k}{T} t_i)\right)$$

Equation 3.2

In this equation, $d_{observation_i}$ is the observed NDVI value at time(t), d_{pred_i} represents the predicted NDVI value, and *a* and *b* are coefficients to be estimated. This modeling technique has been applied to AVHRR data, which is sampled daily and has a 1-km spatial resolution. In contrast, Landsat scenes are resampled every sixteen days, making time-sensitive phenology assessments more challenging. Landsat data are collected at a 30m spatial resolution, allowing for more precise characterization of heterogenous landscapes. Harmonic regression analysis using Landsat 5 TM and Landsat 7 ETM+ data has been applied to crop phenology research (Roy and Yan, 2018), however, little work has been done to assess the usefulness of this approach in characterizing forest communities and their recovery following a disturbance using Landsat imagery.

Using the harmonic regression modelling approach, I constructed phenological models for different subsets of the Landsat data, based on forest group, burn severity, and fire ID. In designing my initial models, I decided to prioritize flexibility by allowing for 20 sine and 20 cosine waves every 365 days. These parameters can be adjusted to for different data types or ecological systems. The initial harmonic regression for annual NDVI is highly flexible and captures every fluctuation in NDVI throughout the year but is quite noisy and does not accurately reflect the average phenology of the forests in the region (Figure 3.7).

3.7 Smoothing and standard error

Harmonic regression generates a flexible curve, representing the fluctuation of NDVI for a pixel throughout the year (Figure 3.7). However, these rapid fluctuations do not represent average annual phenology and are quite noisy. To reduce the curve's derivative, I imposed smoothing parameters to penalize the noise and attempt to filter out the "real" NDVI signal, similar to methods developed by Hermance (2007). To construct the phenology models from the Landsat data, I used penalized constrained least squares fitting, which requires a smoothing parameter, a matrix of cosine and sine values at each time step, and a penalty matrix of cosine and sine derivatives at each time step (pcls in the mgcv R package).

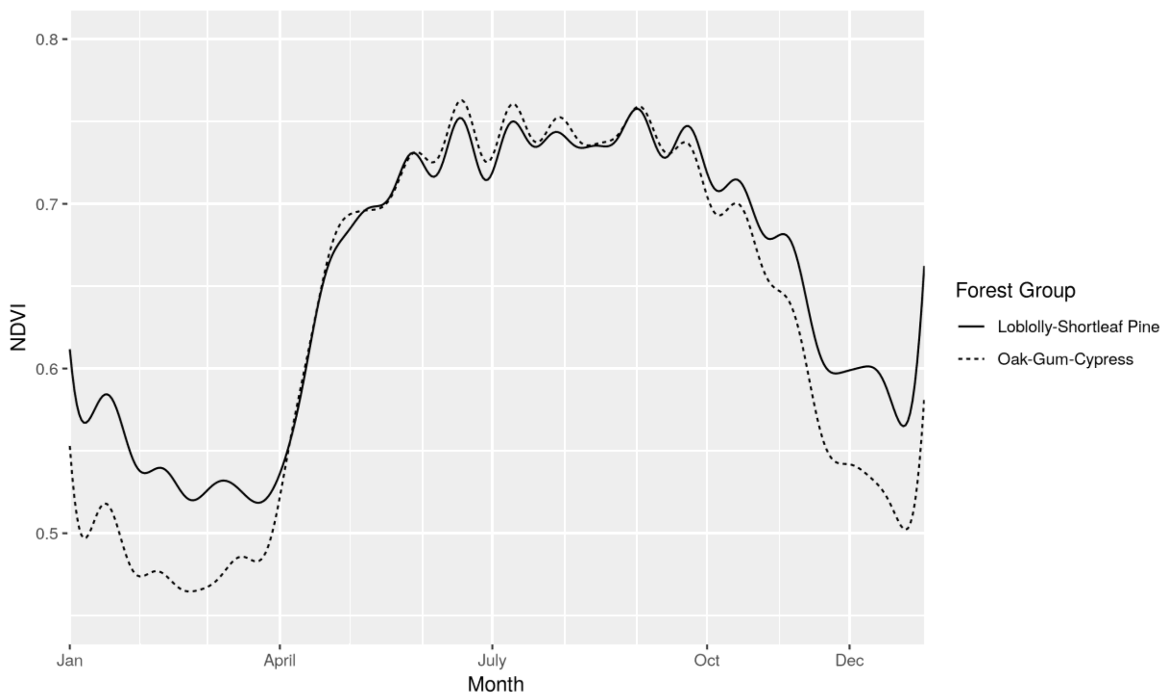


Figure 3.7. Average annual harmonic regression model output for loblolly-shortleaf pine (solid) and oak-gum-cypress (dotted), 1984-2017.

The smoothed harmonic regression model outputs for the loblolly-shortleaf pine are much more representative of the expected vegetation phenology in the region (Figure 3.8, left). It is important to note that applying smoothing parameters that are too strict generates curves that provide little information about the seasonal patterns of the land surface being studied (Figure 3.8, right). To quantify the uncertainty of the penalized least squares regression coefficients, I calculated the standard deviation (Equation 3.3) and the standard error (Equation 3.4) for each of my models. All figures containing phenology curves in the results include confidence intervals for the lower and upper bounds of NDVI and NBR (NDVI or NBR ±2 *standard error).

$$\sigma_e^2 = \sqrt{\frac{\sum(y - \hat{y})^2}{n-k}} \qquad \textbf{\textit{Equation 3.3}}$$

$$min \sum_i (y_i - \hat{y}_i)^2 + \lambda(wiggliness) \qquad \textbf{\textit{Equation 3.4}}$$

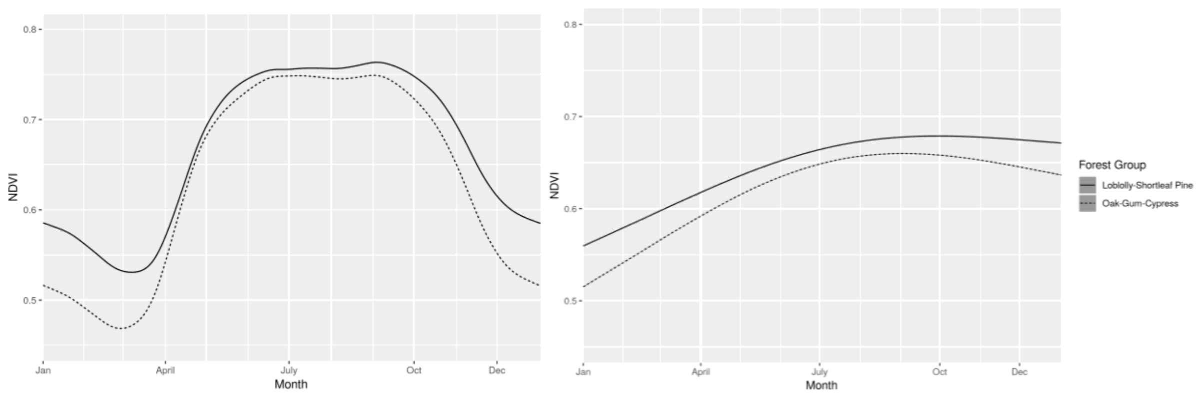


*Figure 3.8. Average annual harmonic regression model outputs for loblolly-shortleaf pine (solid) and oak-gum-cypress (dotted), with penalized least squares smoothing matrix chosen for analysis (left). If the specified smoothing parameter is too high, the output can become too restrictive (right) and does not reflect expected phenology. Both figures include confidence intervals (NDVI ±2 *standard error).*

3.8 Phenology metrics

In addition to constructing phenology curves using a smoothed harmonic regression approach, I derived several phenology metrics from the curves to compare the seasonal differences between the forest groups, pre- and post-fire, and different levels of fire severity. These metrics include the minimum and maximum predicted annual values of NBR and NDVI, the dates on which the minimum and maximum values occur, and the timing of spring and fall senescence based on NDVI or NBR. There is no universally accepted method for defining or deriving phenology metrics, such as start of spring (White et al., 2009). However, I selected a method in which a spectral index threshold value (SI_{ratio}) is used to identify the spring green up and start of growing season (White et al., 1997). White et al. used 50% threshold values to identify spring and onset and cessation, a value that works well with a variety of land cover types. In this method, SI_{ratio} is the percentage of the seasonal amplitude reached at a given point in time (Equation 3.5). To identify the date for spring onset and fall senescence from the phenology curves developed in this thesis, I used 50% threshold values.

$$SI_{ratio} = \frac{SI - SI_{min}}{SI_{max} - SI_{min}} \qquad \textbf{\textit{Equation 3.5}}$$

4 Results

4.1 Fire summary statistics

In total, I compared NDVI and NBR Landsat phenology curves in 85 fires and their unburned 5km buffer areas. The first fire occurred on April 8, 2004 and the last on May 30, 2015. The vast majority of the fires were prescribed (72 fires, 84.7%), followed by wildfires (7 fires, 8.2%), and unknown (6 fires, 7%) (Table 4.1). The number of acres burned by each fire ranged from 508 to 11,023 acres (Figure 4.1).

Table 4.1. Sample fire type based on Monitoring Trends in Burn Severity data.

Fire type	Number of fires	% of total fires
Prescribed	72	84.7
Wildfire	7	8.2
Unknown	6	7.0

Because fire severity is a pixel-level, categorical variable within the MTBS raster dataset, I used the mode severity of the samples collected from each fire to estimate severity at the fire-level (Figure 4.2). Fire severity ranges from 1 (least severe) to 6 (most severe). Of the sample fires, 14 fires had a mode severity of 1 (16.5%), 64 fires had a mode severity of 2 (75.3%), 0 fires had a mode severity of 4, 4 fires had a mode severity of 3, 1 fire had a mode severity of 5, and 2 fires had a mode severity of 6. The vast majority (91.8%) of the sample fires were low severity (mode severity of 1 or 2), while fires with higher mode severities of 3 ,5, or 6 made up the remaining 8.2% of sample fires.

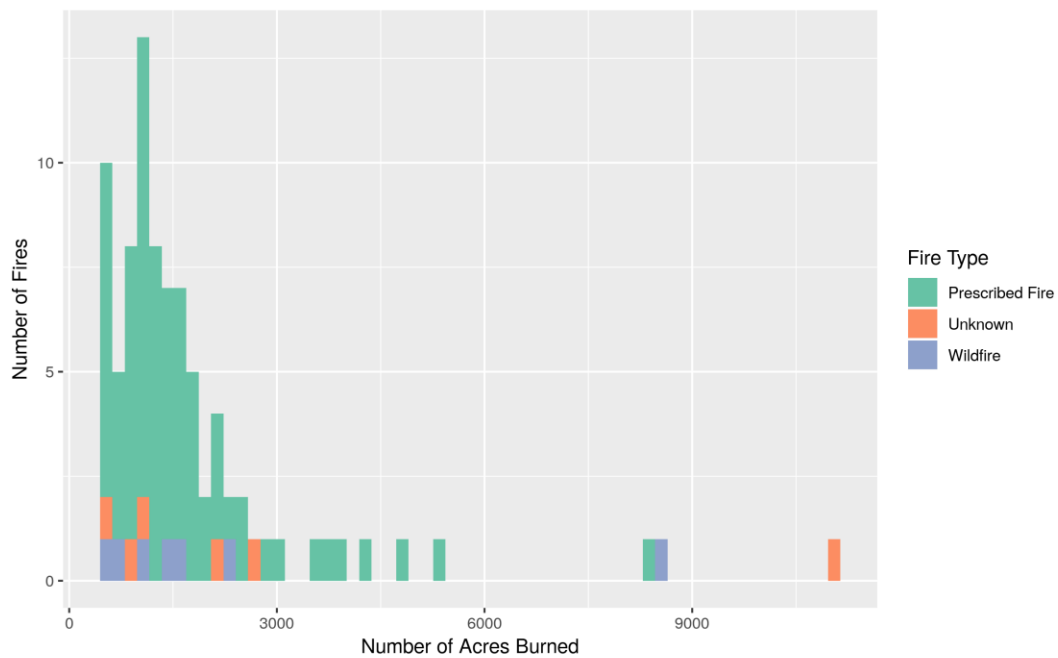


Figure 4.1. Distribution of sample fires and total acres burned.

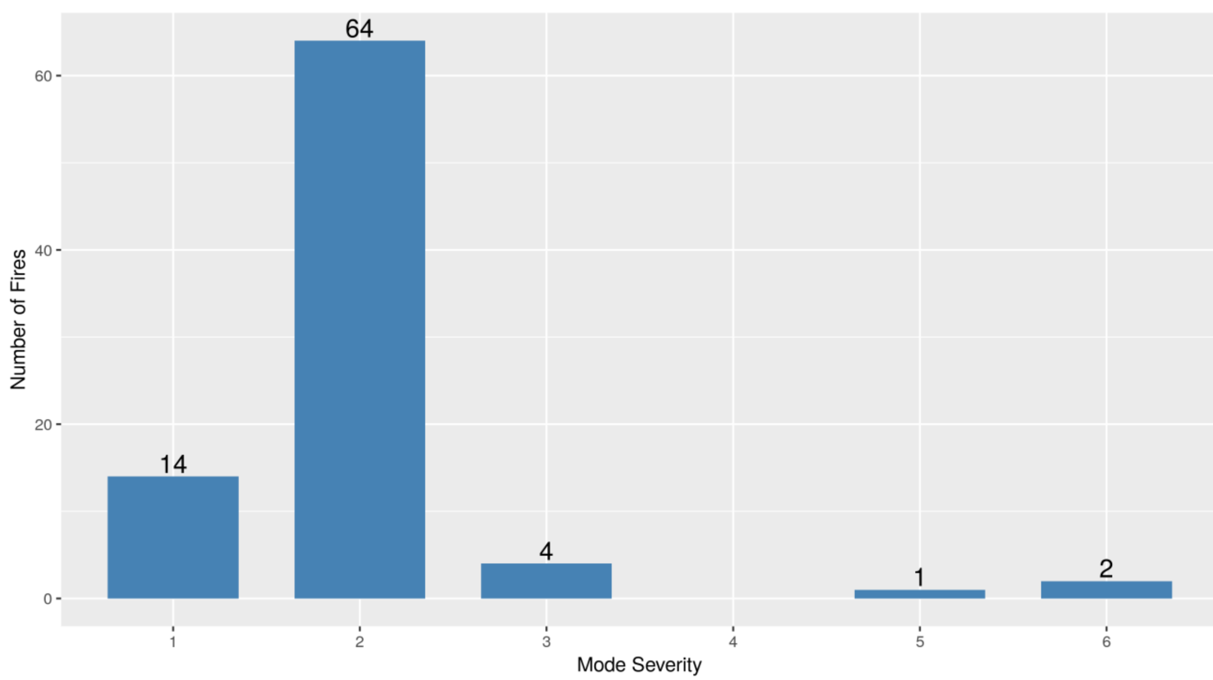


Figure 4.2. Number of fires within each severity category.

Seasonality is a key component for understanding phenological patterns. For this research, I defined spring as March 1 to May 31, summer as June 1 to August 31, autumn/fall as September 1 to November 30, and winter as December 1 through February 28. In terms of the seasonal distribution of the fires, the majority occurred in the spring (73%), followed by the winter months (26%), and autumn (<1%). None of the sample fires occurred during the summer months (Figure 5.3).

4.2 Forest group differences

Based on the raw NDVI observation data, the oak-gum-cypress observations (yellow) do show more seasonality than the loblolly-shortleaf pine (green) (Figure 4.4). However, it's difficult to detect any meaningful differences using only the raw observation data at the patch level. To compare NDVI phenology curves between loblolly-shortleaf pine and oak-gum-cypress, I constructed global models from the pre-fire burned samples for the two forest groups.

Using the global, penalized harmonic regression model, there are distinct differences in the phenology patterns of the two forest groups in the burned samples (Figure 4.5). As expected, the baseline phenology curve of oak-gum-cypress shows a greater intra-annual amplitude than the baseline curve of the loblolly-shortleaf pine. In terms of phenology metrics, NDVI in loblolly-shortleaf pine ranged from .05395 to 0.7751, with a seasonal amplitude of 0.2356 (Table 4.2). NDVI predictions in oak-gum-cypress ranged from 0.465 to 0.746, with a seasonal amplitude of 0.2752. For the dates derived from the phenology curves of the two forest groups, date of maximum NDVI showed the greatest difference, with loblolly-shortleaf pine reaching maximum NDVI at day 244 (September 1 or August 31), and oak-gum-cypress reaching maximum NDVI much earlier at day 190 (July 9 or July 8).

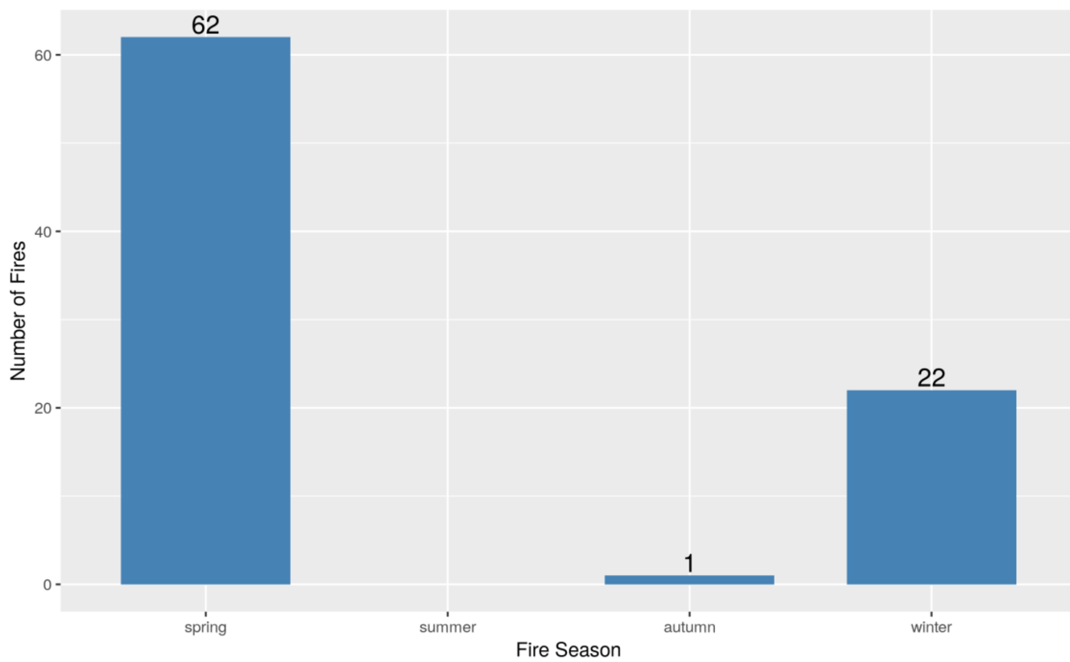


Figure 4.3. Seasonal distribution of sample fires.

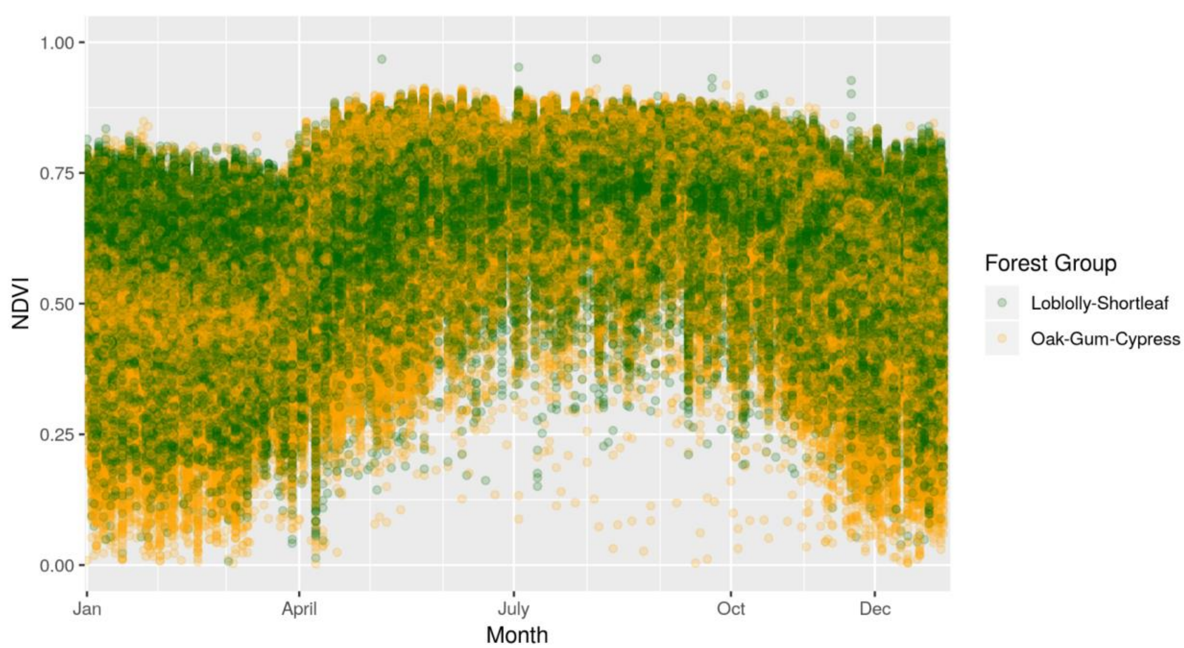


Figure 4.4. Raw NDVI data for all Landsat patches on an annual scale, 1984-2017.

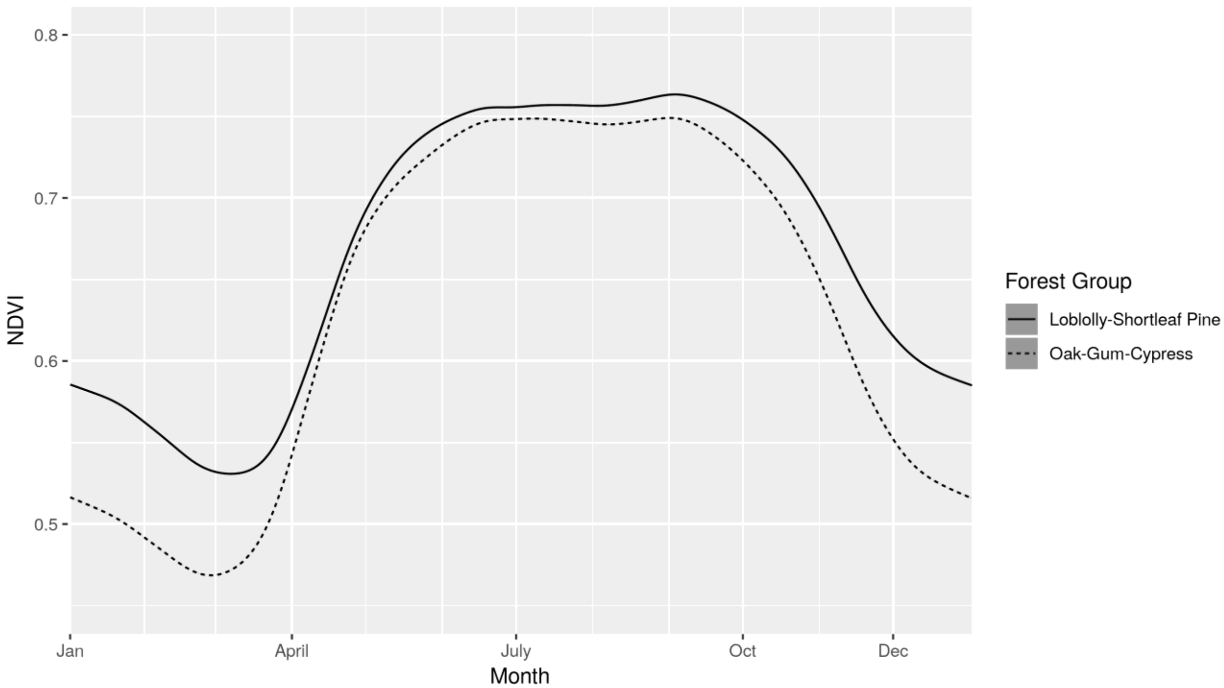


Figure 4.5. Annual NDVI phenology models for the two forest groups, burned samples. Loblolly-shortleaf pine (solid line) and oak-gum-cypress (dotted line), 1984-2017.

Table 4.2. NDVI phenology for loblolly-shortleaf pine and oak-gum-cypress annual predictions, 1984-2017.

	Loblolly-shortleaf pine	**Oak-gum-cypress**
Max NDVI	0.7634	0.7490
Min NDVI	0.5308	0.4686
Seasonal Amplitude of NDVI	0.2327	0.2804
Date of Max NDVI	247	244
Date of Min NDVI	66	58
Date of Spring Onset	108	103
Date of Senescence	322	317

4.3 Pre- and post-fire

For the raw patch observation data, the post-fire samples showed slightly elevated NDVI levels in comparison to the pre-fire samples, particularly in the spring and summer. However, there were no differences in NBR patch observations between pre- and post-fire samples (Figure 4.6). For both loblolly-shortleaf pine and oak-gum-cypress, there was little difference in the NDVI phenology patterns between pre- and post-fire (Figure 4.7). The loblolly-shortleaf pine post-fire predictions do show slightly elevated post-fire levels in the late fall and winter, while the oak-gum-cypress post-fire NDVI is lower than pre-fire levels in the summer. The NDVI phenology metrics did not change substantially for either loblolly-shortleaf pine or oak-gum-cypress after fire (Table 4.3). However, the date for maximum NDVI in post-fire oak-gum-cypress was much later in the year (day 247) than for pre-fire oak-gum-cypress (day 191).

Post-fire NBR for loblolly-shortleaf pine and oak-gum-cypress showed a consistent decrease throughout the year, with the greatest difference occurring from April to October (Figure 4.8). Post-fire NBR maximum and minimum were lower in both loblolly-shortleaf pine and oak-gum-cypress (Table 4.4).

4.4 Post-fire time steps

The time-step models show the difference in phenology pre-fire, less than 2 years after fire, and more than 2 years after fire and demonstrate the different seasonal responses to fire depending on time since fire and forest group (Figure 4.9). The phenology models for loblolly-shortleaf pine less than two years after fire show small increase in NDVI in the fall and winter and a decrease in the spring and summer. The model for more than two years after fire showed a small increase in NDVI during the spring and summer, with NDVI returning to pre-fire levels for the remainder of the year in loblolly-shortleaf pine.

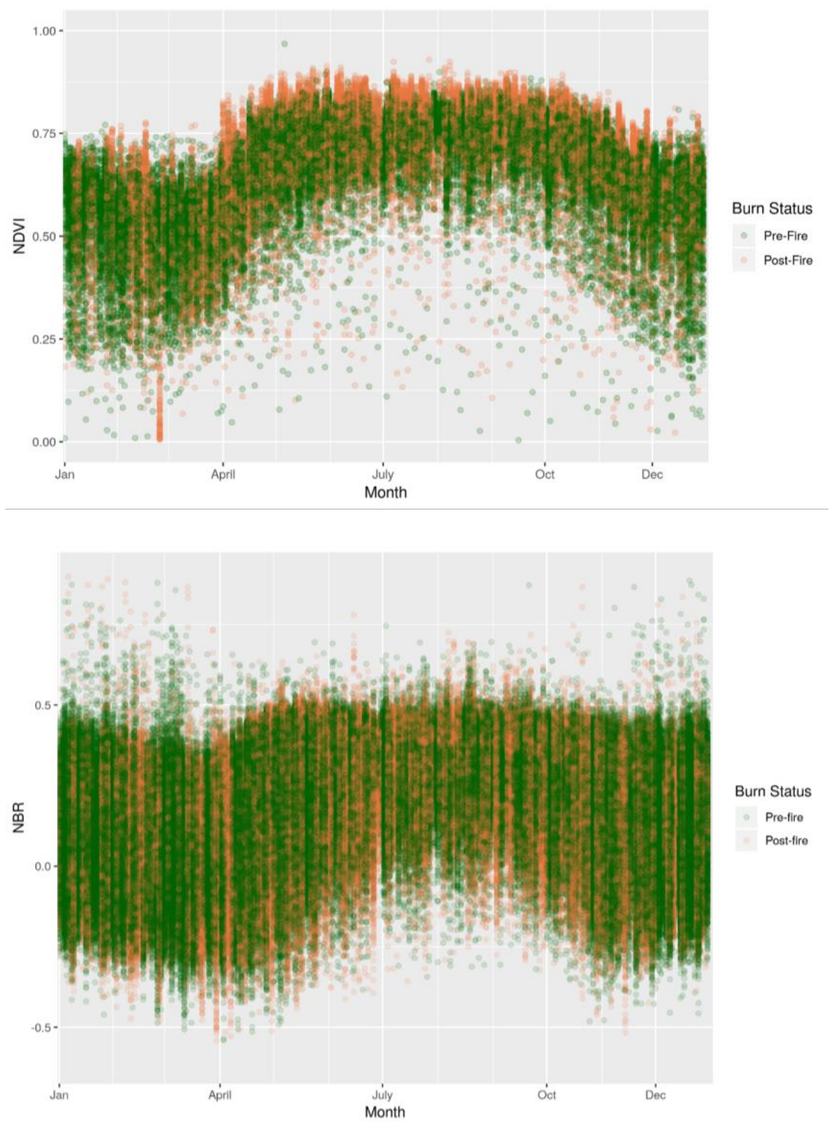
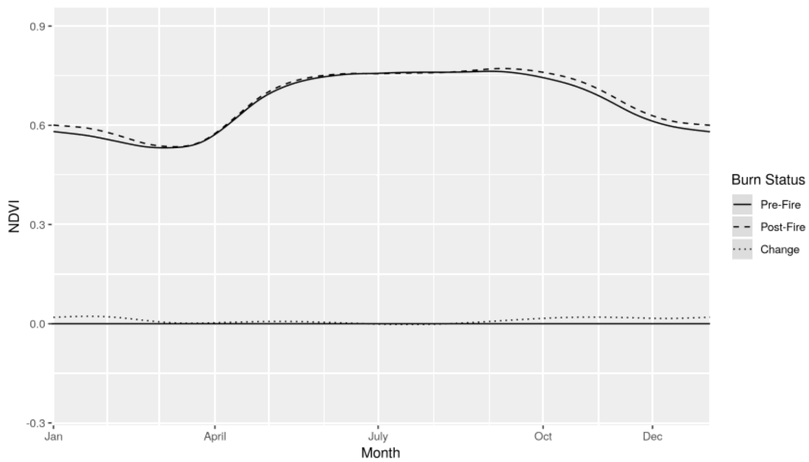


Figure 4.6. Raw NDVI (top) and NBR (bottom) data for pre- (green) and post-fire (orange) sample Landsat patches, compressed to an annual time scale, 1984-2017.



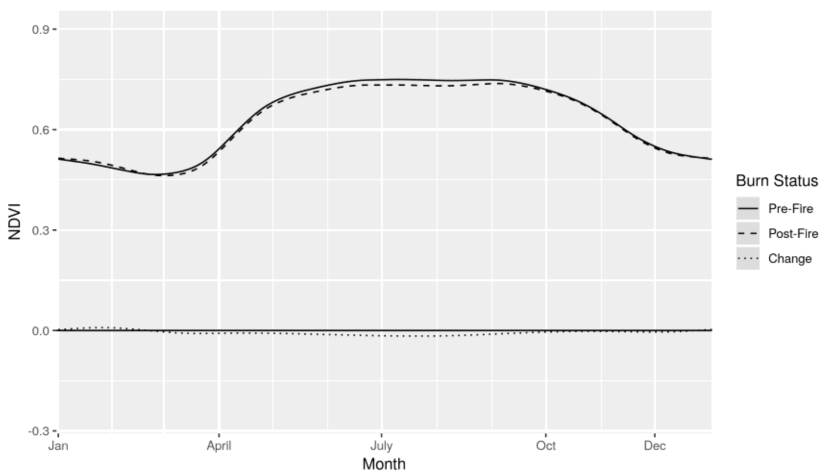
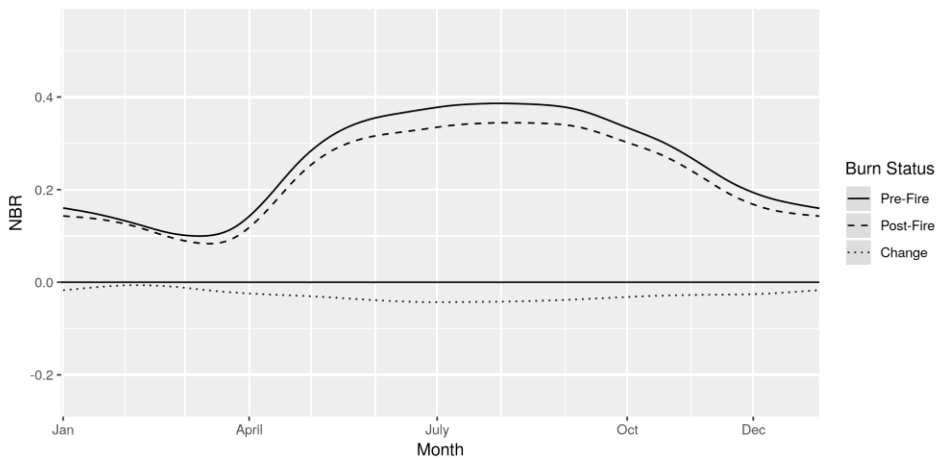


Figure 4.7. Average annual NDVI phenology curves for pre- and post-fire loblolly-shortleaf pine (top) and oak-gum-cypress (bottom), 1984-2017.

Table 4.3. NDVI Phenology metrics for pre- and post-fire annual predictions, 1984-2017.

	Loblolly-shortleaf pine		**Oak-gum-cypress**	
	Pre-fire	**Post-fire**	**Pre-fire**	**Post-fire**
Max NDVI	0.7634	0.7716	0.7497	0.7376
Min NDVI	0.5320	0.5343	0.4659	0.4624
Seasonal Amplitude of NDVI	0.2314	0.2363	0.2838	0.2752
Date of Max NDVI	245	251	191	247
Date of Min NDVI	63	68	56	60
Date of Spring Onset	108	108	103	103
Date of Senescence	320	325	317	318



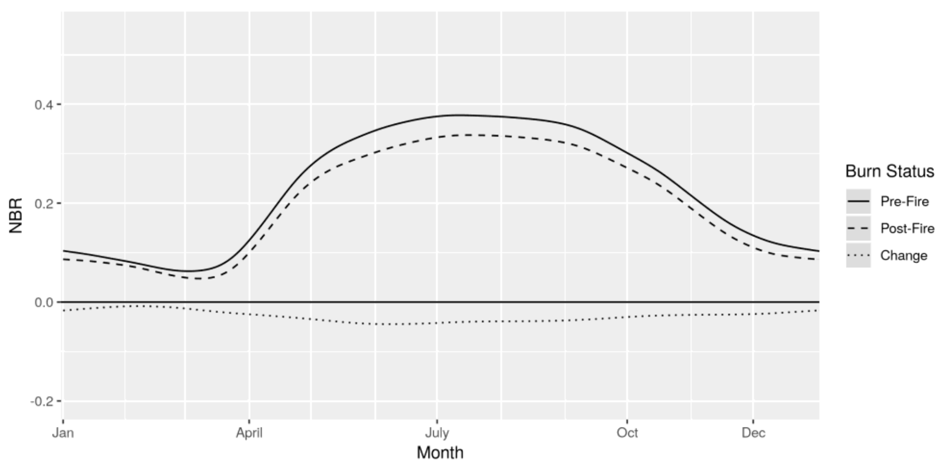


Figure 4.8. Average annual NBR phenology curves for pre- and post-fire loblolly-shortleaf pine (top) and oak-gum-cypress (bottom), 1984-2017.

Table 4.4. Normalized burn ratio (NBR) phenology metrics for pre- and post-fire annual predictions, 1984-2017.

	Loblolly-shortleaf pine		**Oak-gum-cypress**	
	Pre-fire	**Post-fire**	**Pre-fire**	**Post-fire**
Max NBR	0.3863	0.3444	0.3778	0.3376
Min NBR	0.0998	0.0832	0.0624	0.0475
Seasonal Amplitude of NBR	0.2865	0.2611	0.3154	0.2902
Date of Max NBR	212	216	193	197
Date of Min NBR	67	72	62	67
Date of Spring Onset	111	111	108	109
Date of Senescence	315	315	304	304

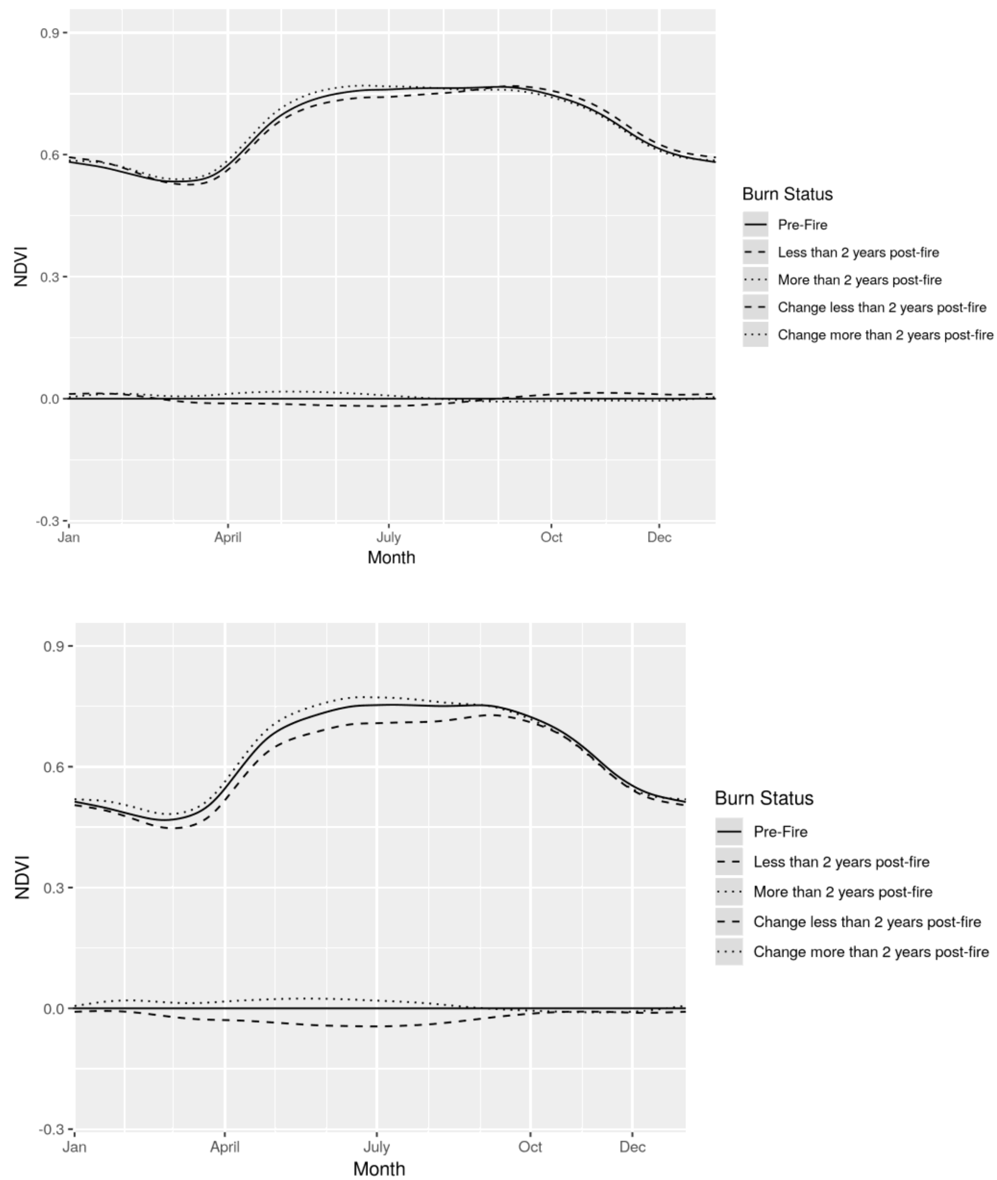


Figure 4.9. Average annual NDVI phenology curves for pre-fire, less than two-years after fire, and more than two years after fire for loblolly-shortleaf pine (top) and oak-gum-cypress (bottom), 1984-2017.

Oak-gum-cypress NDVI phenology curves for less than 2 years after fire decrease from pre-fire levels in the early spring through the summer, while the model for more than 2 years after fire shows a small increase in the early spring and summer. For NDVI phenology metrics, the biggest change occurred for date of maximum NDVI, which shifted later by two months for both forest groups more than 2 years after fire (Table 4.5).

The NBR phenology curves revealed different seasonal patterns than the NDVI predictions (Figure 4.10). For loblolly-shortleaf pine less than 2 years after fire, NDVI remains below pre-fire levels in the spring, summer, and fall, before returning to pre-fire levels in the winter. The NBR phenology curve for loblolly-shortleaf pine more than 2 years after fire declines in the early summer and falls below even the less than 2 years after fire predictions in the late summer. The oak-gum-cypress NBR phenology curves show a similar pattern, with the less than 2 years after fire curve decreasing more in the early spring and summer. Similar to the phenology metrics observed for NDVI, the greatest change between less than 2 years after fire and more than 2 years after fire occurred for the date of maximum NBR, which shifted to a later date more than 2 years after fire (Table 4.6).

4.5 Fire severity

In addition to comparing the pre- and post-fire phenology curves globally, I also compared phenology curves between differing levels of fire severity, derived from the MTBS dataset. For post-fire loblolly-shortleaf pine, NDVI increased throughout the year for low severity fires (ranking 1), remained the same in fires with a ranking of 2, decreased slightly in the winter and fall in fires with a 3 ranking, and decreased the most in fires with a 6 ranking (Figure 4.11). For loblolly-shortleaf pine samples in fires with a 6 ranking, the decrease was most pronounced in the winter, spring, and early summer.

Table 4.5. NDVI phenology metrics for time steps after fire, 1984-2017.

	Loblolly-shortleaf pine		**Oak-gum-cypress**	
	< 2 years post-fire	**>2 years post-fire**	**< 2 years post-fire**	**> 2years post fire**
Max NDVI	0.7687	0.7698	0.7281	0.7727
Min NDVI	0.5262	0.5396	0.4473	0.4825
Seasonal Amplitude of NDVI	.2424	0.2302	0.2807	0.2909
Date of Max NDVI	252	168	250	174
Date of Min NDVI	69	63	60	57
Date of Spring Onset	110	106	104	102
Date of Senescence	326	317	321	309

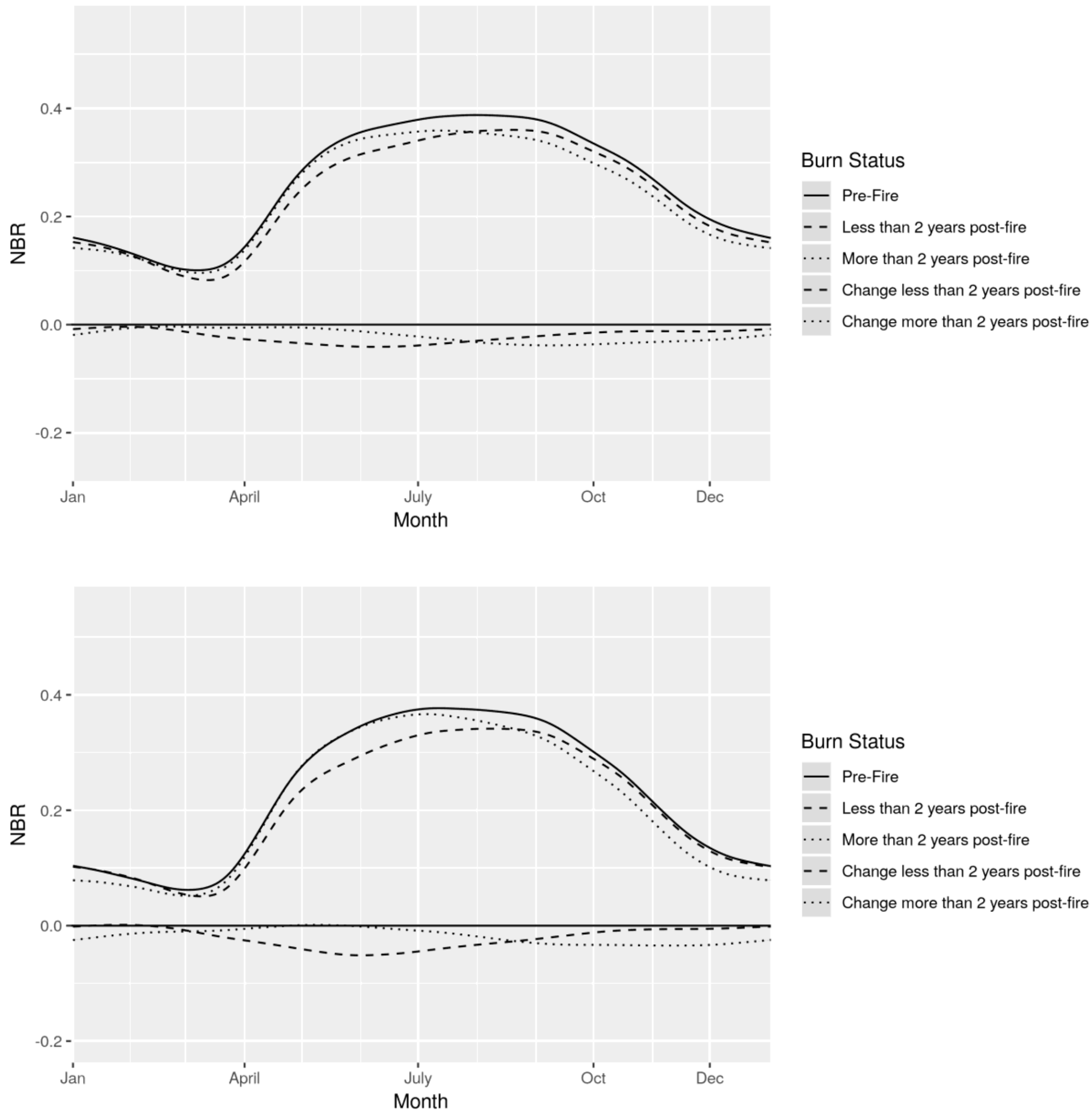


Figure 4.10. Average annual NBR phenology curves for pre-fire, less than two-years after fire, and more than two years after fire for loblolly-shortleaf pine (top) and oak-gum-cypress (bottom), 1984-2017.

Table 4.6. NBR phenology metrics for time steps after fire, 1984-2017.

	Loblolly-shortleaf pine		**Oak-gum-cypress**	
	< 2 years post-fire	**> 2 years post-fire**	**<2 years post-fire**	**>2 years post-fire**
Max NBR	0.3602	0.3590	0.3413	0.3666
Min NBR	0.0824	0.0957	0.0508	0.052
Seasonal Amplitude of NBR	0.2779	0.2633	0.2906	0.3147
Date of Max NBR	232	193	219	187
Date of Min NBR	72	68	69	61
Date of Spring Onset	113	109	110	106
Date of Senescence	318	309	309	297

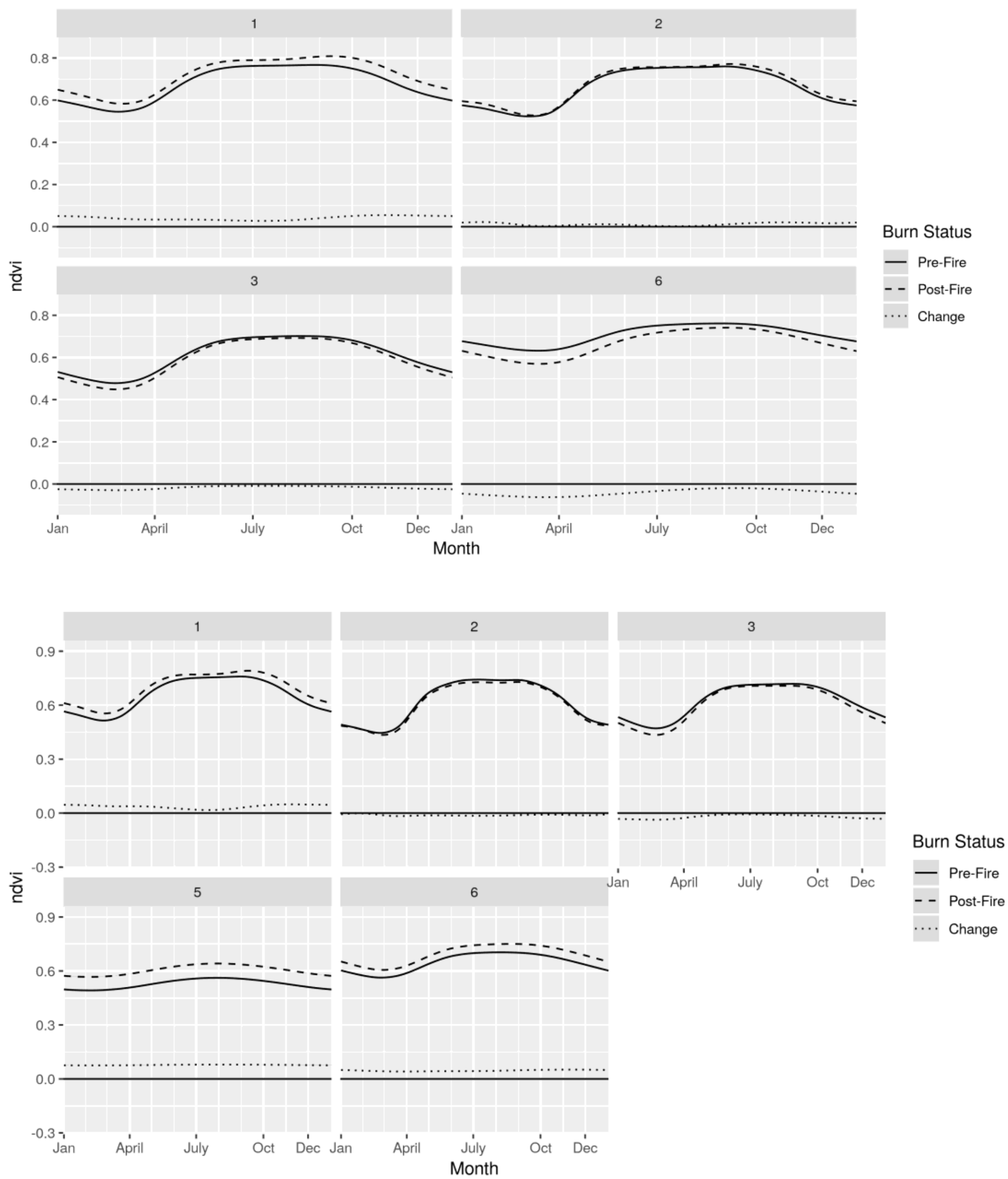


Figure 4.11, Average annual NDVI phenology curves for pre- and post-fire, modelled by fire severity, loblolly-shortleaf pine (top)– (severity-Number of fires: 1-13 fires, 2-57 fires, 3-4 fires, 6-2 fires) and oak-gum-cypress (bottom) – (1-12 fires, 2-56 fires, 3-4 fires, 5-1 fire, 6-1 fire), 1984-2017.

Patterns in oak-gum-cypress were similar to the patterns observed in loblolly-shortleaf pine samples, with the exception of severe fires (Figure 4.11). Post-fire NDVI phenology curves in oak-gum-cypress with fire severity rankings of 5 and 6 showed an increase throughout the year; however, the sample size for severe fires was small. The fire severity-level NBR phenology curves revealed similar patterns to those observed in the NDVI curves for both forest groups (Figure 4.12). However, in contrast to the stable patterns observed for NDVI, post-fire NBR showed a decrease for fires ranking 2 and 3 in both loblolly-shortleaf pine and oak-gum-cypress.

Examining post-fire time steps at the fire-severity level revealed different responses than observed for pre- and post-fire (Figure 4.13). In fires with a severity ranking of 1, both post-fire time step NDVI phenology curves in loblolly-shortleaf pine showed slight increases from pre-fire levels throughout the year. For fires with severity rankings of 3 and 6, NDVI phenology curves in loblolly-shortleaf pine less than 2 years after fire showed a decrease throughout the year. The phenology curves for more than 2 years after fire showed a consistent increase in NDVI. This pattern was also observed for oak-gum-cypress in fire severity 3. The fire with a severity ranking 5 only contained oak-gum-cypress samples and both post-fire NDVI phenology curves increased from pre-fire levels. For the fire with a 6-severity ranking, the phenology prediction for less than 2 years after fire was not different than the pre-fire phenology and the curve for more than 2 years after fire showed an increase in NDVI. NBR phenology curves in both forest groups showed similar patterns to the NDVI phenology predictions at varying time steps following fire (Figure 4.14).

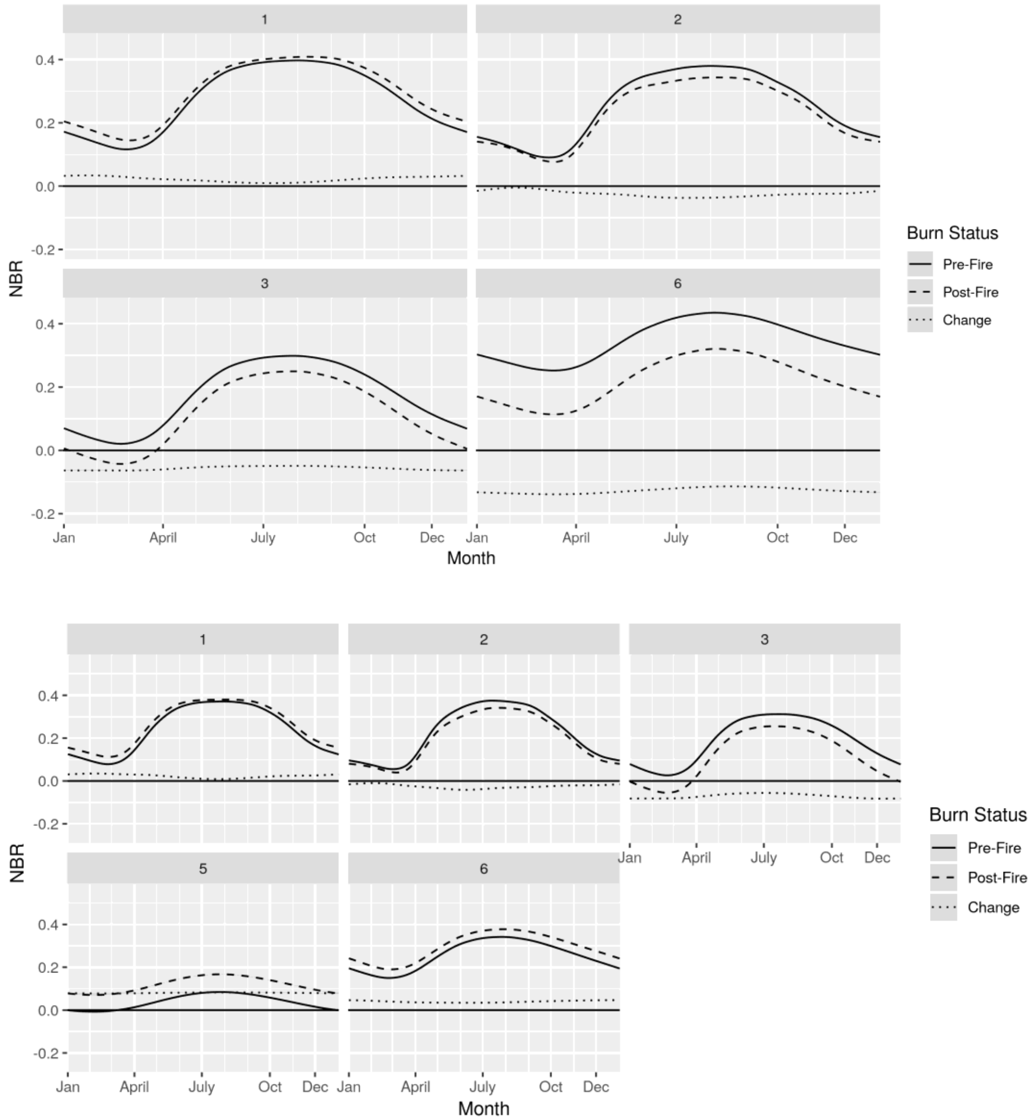


Figure 4.12. Average annual NBR phenology curves for pre- and post-fire, modelled by fire severity, loblolly-shortleaf pine (top) – (severity-Number of fires: 1-13 fires, 2-57 fires, 3-4 fires, 6-2 fires) and oak-gum-cypress (bottom) – (1-12 fires, 2-56 fires, 3-4 fires, 5-1 fire, 6-1 fire), 1984-2017.

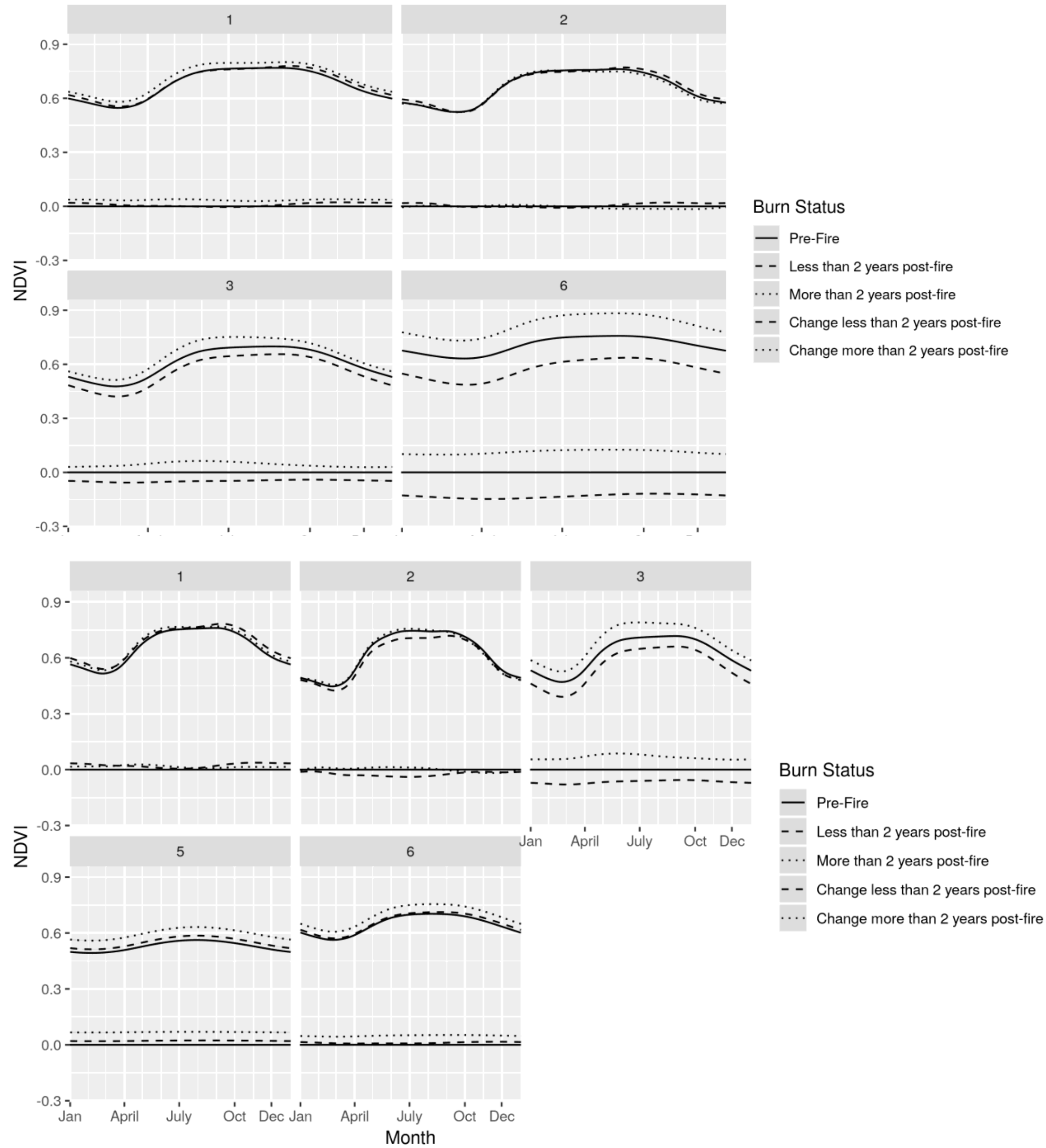
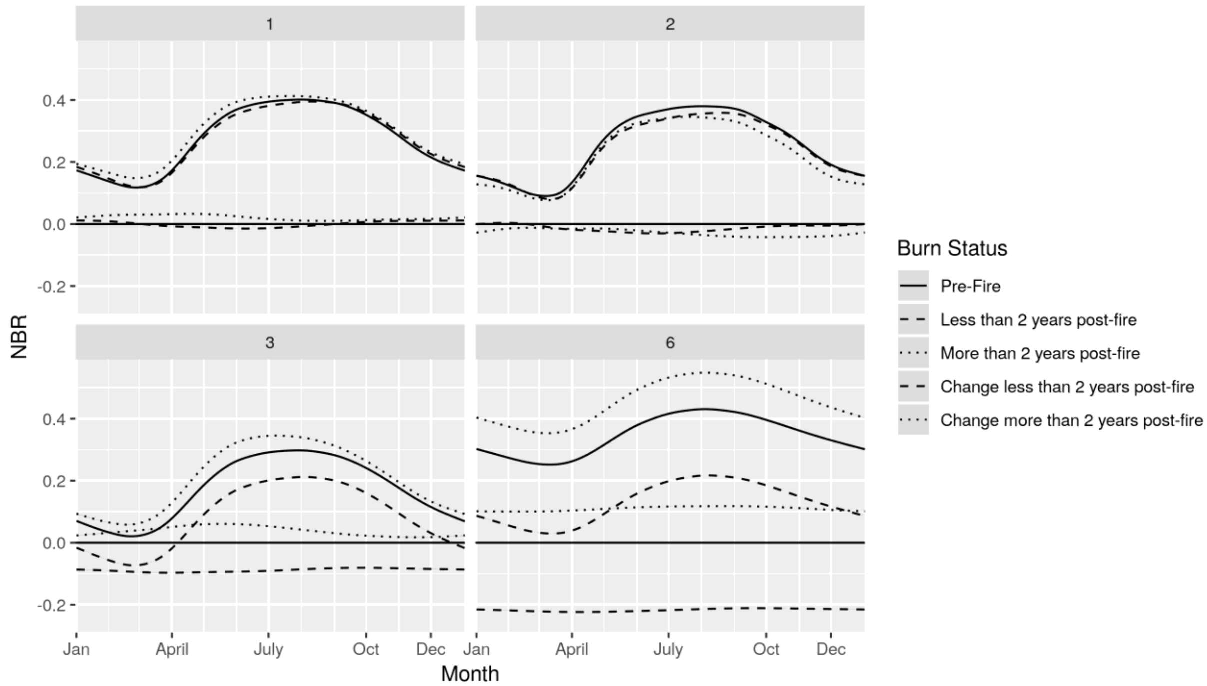


Figure 4.13. Annual average NDVI phenology curves for post-fire time steps, modelled by fire severity, loblolly-shortleaf pine (top), 1984-2017.



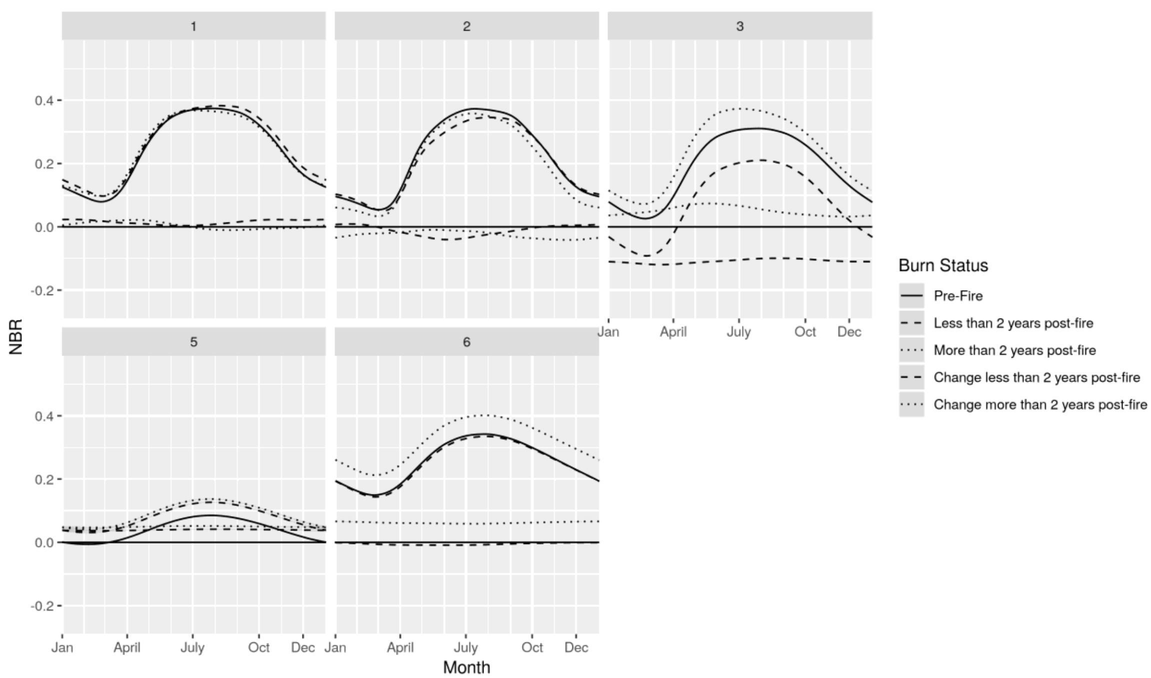


Figure 4.14. Annual average NBR phenology curves for post-fire time steps, modelled by fire severity, loblolly-shortleaf pine (top), 1984-2017.

4.6 Comparing burned and unburned samples

Unlike the patch-level NDVI observation for the burned samples (Figure 4.4), there were no detectable differences between forest groups in the unburned samples observations (Figure 4.15). Similarly, the phenology curves constructed from the unburned samples revealed no differences between loblolly-shortleaf pine and oak-gum-cypress (Figure 4.16). The NDVI phenology metrics extracted from the curves constructed using the unburned samples revealed no differences between forest groups (Table 4.7).

Phenology curves in the unburned samples showed distinct increases in NDVI post-fire when compared to pre-fire curves for both forest groups throughout the year (Figure 4.17). There were only small changes in the NBR phenology curves for pre- and post-fire samples in the control group (Figure 4.18). Note that the "pre-fire" and "post-fire" designations refer to when the corresponding burned samples experienced fire. Post-fire NDVI phenology curves for loblolly-shortleaf pine and oak-gum-cypress showed an increase in both minimum and maximum NDVI, with no distinct changes in the DOY on which these events occurred (Table 4.8). The seasonal amplitude for NDVI and NBR phenology increased for post-fire loblolly-shortleaf pine and oak-gum-cypress (Table 4.8, Table 4.9).

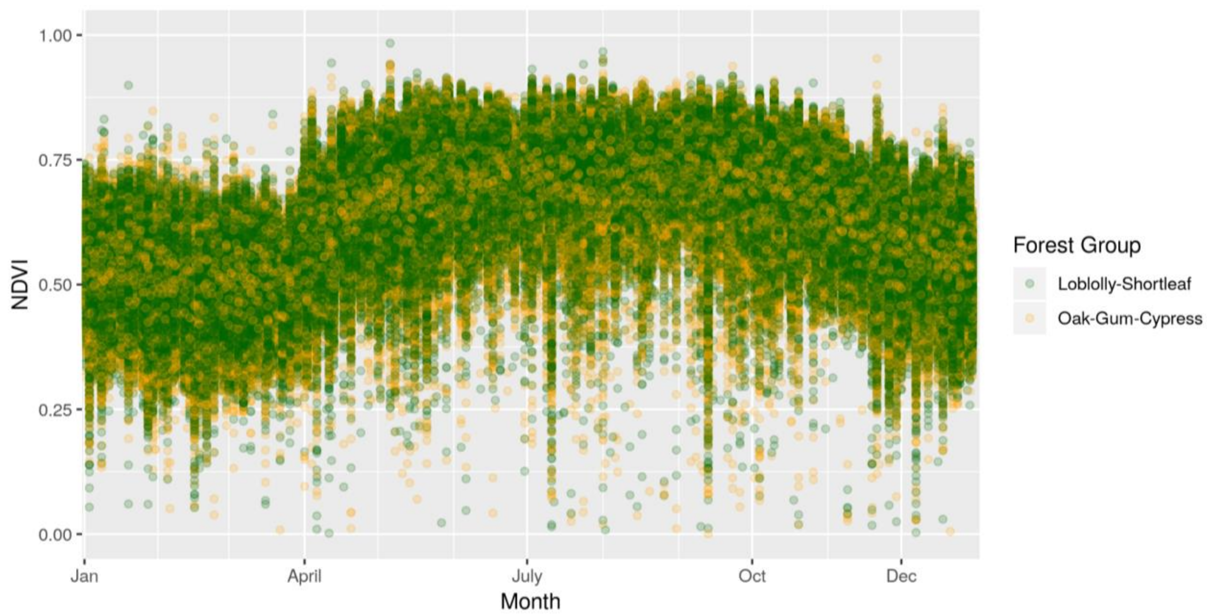


Figure 4.15. Raw NDVI data for unburned samples Landsat patches, compressed to an annual time scale, 1984-2017.

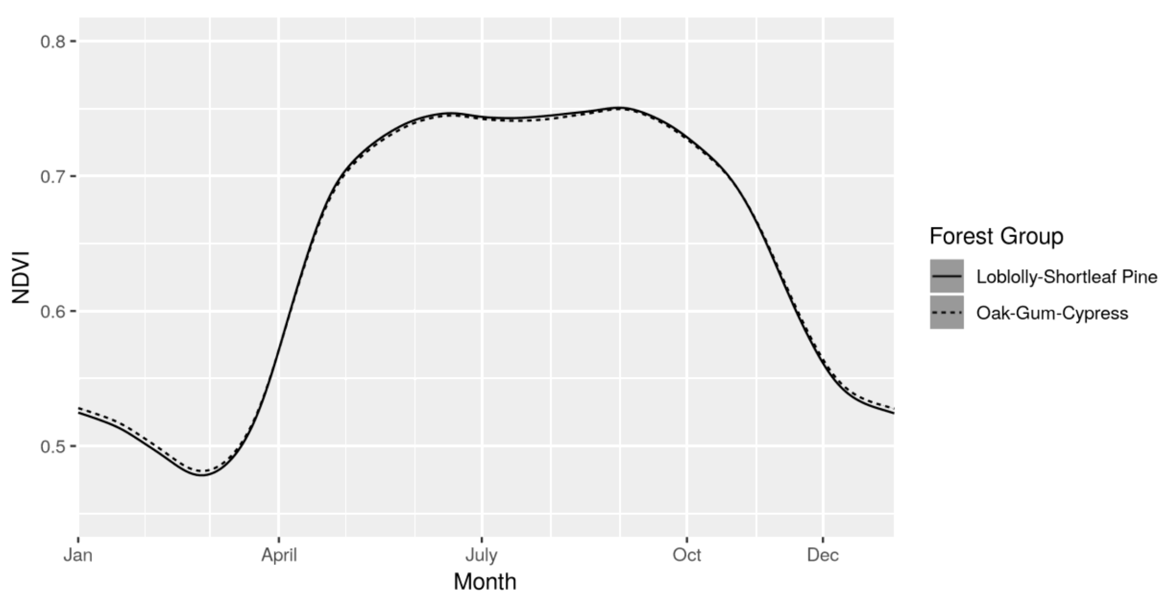


Figure 4.16. Annual NDVI phenology models for the two forest groups in unburned samples. Loblolly-shortleaf pine (solid line) and oak-gum-cypress (dotted line), 1984-2017.

Table 4.7. NDVI phenology metrics for loblolly-shortleaf pine and oak-gum-cypress, unburned samples, 1984-2017.

	Loblolly-shortleaf pine	**Oak-gum-cypress**
Max NDVI	0.7319	0.7307
Min NDVI	0.4649	0.4697
Seasonal Amplitude of NDVI	0.2670	0.2610
Date of Max NDVI	244	244
Date of Min NDVI	51	52
Date of Spring Onset	100	100
Date of Senescence	321	321

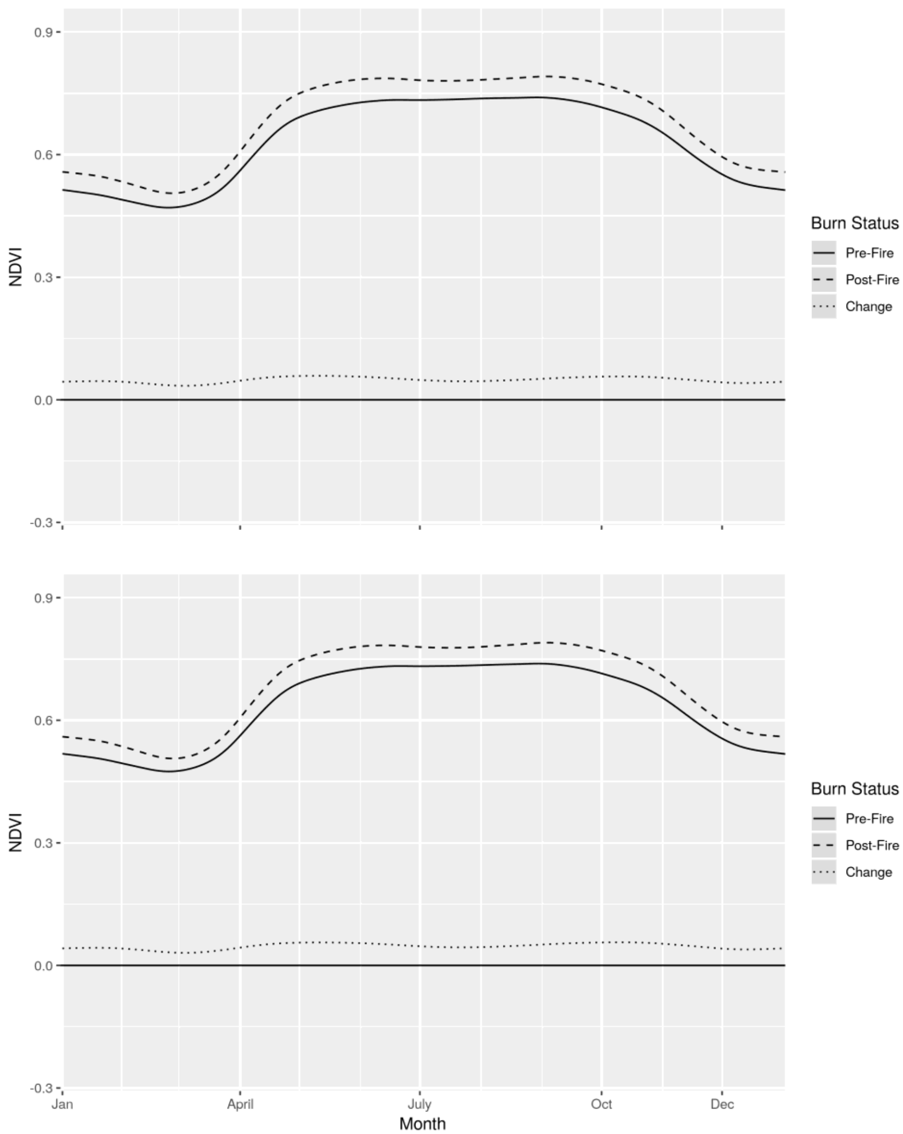


Figure 4.17. Average annual NDVI phenology curves for pre- and post-fire for loblolly-shortleaf pine (top) and oak-gum-cypress (bottom) for the unburned samples, 1984-2017.

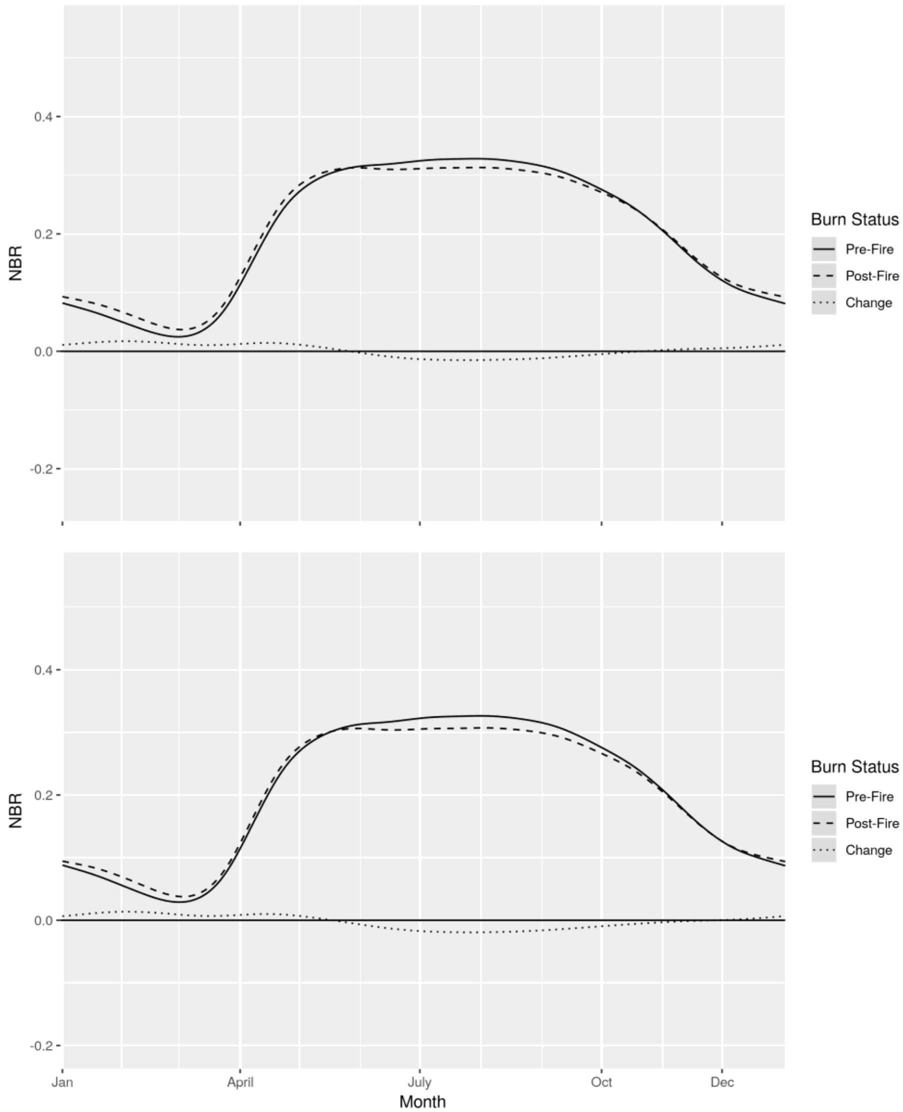


Figure 4.18. Average annual NBR phenology curves for pre- and post-fire for loblolly-shortleaf pine (top) and oak-gum-cypress (bottom) for the unburned samples.

Table 4.8. NDVI phenology metrics for pre- and post-fire annual predictions in unburned samples, 1984-2017.

	Loblolly-shortleaf pine		**Oak-gum-cypress**	
	Pre-fire	**Post-fire**	**Pre-fire**	**Post-fire**
Max NDVI	0.7398	0.7911	0.7388	0.7900
Min NDVI	0.4702	0.5054	0.4747	0.5064
Seasonal Amplitude of NDVI	0.2696	0.2857	0.2641	0.2836
Date of Max NDVI	241	246	242	246
Date of Min NDVI	55	57	55	57
Date of Spring Onset	98	97	99	97
Date of Senescence	319	320	319	321

Table 4.9. NBR metrics for pre- and post-fire oak-gum-cypress in unburned samples.

	Loblolly-shortleaf pine		**Oak-gum-cypress**	
	Pre-fire	**Post-fire**	**Pre-fire**	**Post-fire**
Max NBR	0.3281	0.3131	0.3262	0.3070
Min NBR	0.0247	0.0368	0.0289	0.0377
Seasonal Amplitude of NBR	0.3034	0.2762	0.2973	0.2694
Date of Max NBR	210	212	212	214
Date of Min NBR	60	62	60	62
Date of Spring Onset	100	98	101	98
Date of Senescence	315	316	316	317

5 Discussion

5.1 Forest group differences

The phenology models for oak-gum-cypress curve had a greater seasonal amplitude than the loblolly-shortleaf pine model. These results support previous findings that phenology is a useful remotely-sensed metric for distinguishing between different vegetation classifications, particularly between deciduous and evergreen forest types. However, the loblolly-shortleaf pine model demonstrated more seasonality than expected for an evergreen forest group, likely due to the heterogeneity of the forests in this region.

One potential issue with these findings is that the original forest group map classifications were developed in 2003 and were assumed to remain constant throughout the study period. In the future, post-fire metrics collected in the field could be related to phenology curves over time for a more dynamic understanding of forest group composition change. While the forest group phenology curves constructed from the burned sample data revealed distinct differences, the curves constructed from the unburned samples did not differ by forest group. The discrepancies between the forest group differences found in the burned samples vs. the unburned samples is addressed later in the discussion.

5.2 Pre- and post-fire

Pre- and post-fire phenology differences were highly dependent on the spectral index chosen for analysis. In the original pre- and post-fire phenology models, both loblolly-shortleaf pine and oak-gum-cypress demonstrated almost no change in NDVI. However, NBR, which is commonly used to identify and assess burned vegetation, decreased from pre- to post-fire. A potential explanation for the lack of difference in NDVI is that the fires carried out in the study area are designed to maintain the forests. This explanation is further supported by the sharp increase in NDVI observed in the unburned control samples after the corresponding burned samples experienced fire.

Time since fire also played a role in determining the shape and fluctuations of the phenology curves, depending on forest group. There were differences in NBR when the data were divided into the time steps, within 2 years of fire and more than 2 years after fire. A potential explanation for this observation could be that prescribed fires carried out in the southeast rarely kill overstory trees, and that any overstory fire effects are not observable more than 2 years after a fire event.

5.3 Fire severity

The results presented in this research support differential phenology responses to varying degrees of fire severity. However, a larger sample size of severe fires (ranked 5 and 6) would be needed to determine if the observed differences are “real.” The vast majority of the study fires were low in severity, with mode pixel severity rankings of 1 or 2. For low severity fires, both forest groups demonstrated an increase or no change in NDVI after fire. When considering time since fire, oak-gum-cypress NDVI phenology did show a small decrease in the summer less than 2 years post-fire. For severe fires, loblolly-shortleaf pine and oak-gum-cypress had contrasting responses. The post-fire NDVI phenology curve in loblolly-shortleaf pine was lower, particularly in the spring, early summer, and winter, than pre-fire levels. The time step results suggest that loblolly-shortleaf pine show a decrease in NDVI throughout the year less than 2 years post-fire but an increase more than 2 years post-fire. However, for fires with severity rankings of 5 and 6, oak-gum-cypress NDVI phenology curves showed either stable patterns or a consistent increase after fire, regardless of time since fire. These contrasting results are likely due to the small sample size of severe fires.

5.4 Differences between the burned and unburned samples

Among the most puzzling findings of this research were the results of the burned to unburned sample comparisons. While the phenology curves and metrics derived from the burned samples showed differences between loblolly-shortleaf pine and oak-gum-cypress, there were essentially no forest group differences in the unburned samples. While pre- and post-fire NDVI phenology in the burned samples showed very small differences, the unburned samples showed a distinct upward shift in NDVI during the time period after the corresponding burned sites experienced fire. NBR phenology curves were more representative of the expected results, with a downward shift from pre- to post-fire in burned samples and no change between pre- and post-fire phenology in the corresponding unburned samples. Because the majority of the fires carried out in the study area are prescribed and not random, it is possible that the unburned samples are not as environmentally similar to the burned samples as assumed in the research design.

There are some potential ecological explanations for the similar phenological patterns observed for unburned loblolly-shortleaf pine and oak-gum-cypress. One of the primary goals of prescribed burning in the Coastal Plains ecosystem is to maintain pine-grassland forest types. Because the phenology curves and metrics for the unburned loblolly-shortleaf pine more closely matched the seasonal patterns of a deciduous forest, it is possible that in the absence of fire for the study period, these pine-grasslands have not been maintained in the unburned samples, leading to loblolly-shortleaf pine transitioning to deciduous forest. This observation is supported by research on the widespread transition from fire-tolerant pine to fire-intolerant hardwood species in the southeastern U.S. under fire suppression (Nowacki and Abrams, 2008). Nevertheless, more analysis is needed to confirm these patterns.

5.5 Future considerations

The current research successfully developed preliminary baseline phenology curves for burned forest Landsat patches for loblolly-shortleaf pine and oak-gum-cypress at varying time steps following fires of varying levels of severity. However, future work is needed to implement these methods at broader spatial scales, in different ecosystems, and after various disturbance types. For my analysis, I selected smoothing parameters based on the seasonal nature of phenology and my expectation for the seasonal patterns in the study area. However, to identify the mathematically optimal smoothing parameters, I could have applied k-fold cross validation based on Landsat image date. K-fold cross validation partitions data into equal subsets and then fits the model to all of the subsets except one, which is reserved as the testing set. Error values are then produced based on the model's performance using the withheld testing data. This process continues iteratively until each k fold subset has served as the testing data once, generating the corresponding error values. These error values can then help identify the optimal penalty terms in penalized least squares regression.

In this study, I removed areas that burned multiple times over the course of the study period. However, the management practices implemented in the loblolly-shortleaf pine forests of South Carolina includes repeat burning. Therefore, a potential future study could compare the recovery characteristics in forests that are burned multiple times over the course of a few decades to those that are only burned once. A key management goal in Francis Marion National Forest is to maintain or, in some cases, reintroduce the pine-grassland ecosystems native to the region, which requires frequent, low-intensity burning. Understanding how these prescribed burns are affecting the vegetation composition is essential to making informed management decisions.

The ultimate goal of this research was to move towards developing methods that could be applied to a large region or regions to characterize forest recovery after a variety of disturbances, not just fire. Future research could examine the impacts of harvest, insects, disease, and hurricanes on the phenological signatures of vegetation.

6 Conclusion

The results of the current research support the original hypothesis that the Landsat NDVI and NBR phenology curves in loblolly-shortleaf pine differed from the phenology curves of oak-gum-cypress. Loblolly-shortleaf pine curves consistently demonstrated lower seasonal amplitude than the oak-gum-cypress curves. Pre- and post-fire NDVI phenology curves were not different in either forest group. However, examining the Landsat phenology curves different time steps after fire revealed forest group level differences, with oak-gum-cypress showing a decrease in summer and spring NDVI within two years after fire. On the other hand, post-fire NBR phenology curves in both forest groups showed a decrease from pre-fire levels, with some variability in the magnitude of that decrease throughout the year. The effects of fire severity differed between the forest groups, with loblolly-shortleaf pine showing lowered NDVI and NBR and oak-gum-cypress experiencing increases in both spectral indices following severe fires. However, these findings could be an artifact of the small sample size for severe fires.

There was no difference between the forest group phenology curves constructed from the unburned samples. Additional analysis is necessary to understand the drivers behind the differences observed between phenology curves constructed from the burned and unburned samples. However, a potential ecological explanation for this pattern could be that the lack of fire for the study period has led to the homogenization of forest groups, with the loblolly-shortleaf pine transitioning into a deciduous forest type. Although more work is needed to refine the uncertainty measures of these phenology curves, the current results suggest that deriving phenology curves from Landsat data is a feasible approach for better understanding forest recovery after fire.

References

Bradley, B.A., Jacob, R.W., Hermance, J.F., Mustard, J.F., 2007. A curve fitting procedure to derive inter-annual phenologies from time series of noisy satellite NDVI data. Remote Sens. Environ. 106, 137–145. https://doi.org/10.1016/j.rse.2006.08.002

Buma, B., 2012. Evaluating the utility and seasonality of NDVI values for assessing post-disturbance recovery in a subalpine forest. Environ. Monit. Assess. 184, 3849–3860. https://doi.org/10.1007/s10661-011-2228-y

Cohen, W.B., Spies, T.A., Alig, R.J., Oetter, D.R., Maiersperger, T.K., Fiorella, M., 2002. Characterizing 23 years (1972-95) of stand replacement disturbance in western Oregon forests with landsat imagery. Ecosystems 5, 122–137. https://doi.org/10.1007/s10021-001-0060-X

Cohen, W.B., Yang, Z., Kennedy, R., 2010. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 2. TimeSync - Tools for calibration and validation. Remote Sens. Environ. 114, 2911–2924. https://doi.org/10.1016/j.rse.2010.07.010

DeFries, R., Hansen, M., Townshend, J., 1995. Global discrimination of land cover types from metrics derived from AVHRR pathfinder data. Remote Sens. Environ. 54, 209–222. https://doi.org/10.1016/0034-4257(95)00142-5

Drummond, M.A., Loveland, T.R., 2010. Land-use Pressure and a Transition to Forest-cover Loss in the Eastern United States 60, 286–298. https://doi.org/10.1525/bio.2010.60.4.7

Eidenshink, J., Schwind, B., Brewer, K., Zhu, Z., Quayle, B., Howard, S., Falls, S., Falls, S., 2007. A project for monitoring trends in burn severity. Fire Ecol. Spec. Issue 3, 3–21.

Goward, S.N., Masek, J.G., Cohen, W., Moisen, G., Collatz, G.J., Healey, S., Houghton, R.A., Huang, C., Kennedy, R., Law, B., Powell, S., Turner, D., Wulder, M.A., 2008. Forest Disturbance and North American Carbon Flux. Earth Obs. Syst. 89, 28–31.

https://doi.org/10.1029/2008EO110001

Griffiths, P., Kuemmerle, T., Baumann, M., Radeloff, V.C., Abrudan, I. V., Lieskovsky, J., Munteanu, C., Ostapowicz, K., Hostert, P., 2014. Forest disturbances, forest recovery, and changes in forest types across the carpathian ecoregion from 1985 to 2010 based on landsat image composites. Remote Sens. Environ. 151, 72–88. https://doi.org/10.1016/j.rse.2013.04.022

Hall, F.G., Botkin, D.B., Strebel, D.E., Woods, K.D., Goetz, J., Hall, F.G., Botkin, D.B., Strebel, D.E., Woods, K.D., 1991. Large-Scale Patterns of Forest Succession as Determined by Remote Sensing Published by : Wiley Stable URL : http://www.jstor.org/stable/2937203 REFERENCES Linked references are available on JSTOR for this article : You may need to log in to JSTOR to acces 72, 628–640.

Hejda, M., Pyšek, P., Jarošík, V., 2009. Impact of invasive plants on the species richness, diversity and composition of invaded communities. J. Ecol. 97, 393–403. https://doi.org/10.1111/j.1365-2745.2009.01480.x

Hermance, J.F., 2007. Stabilizing high-order, non-classical harmonic analysis of NDVI data for average annual models by damping model roughness. Int. J. Remote Sens. 28, 2801–2819. https://doi.org/10.1080/01431160600967128

Hermy, M., Verheyen, K., 2007. Legacies of the past in the present-day forest biodiversity: A review of past land-use effects on forest plant species composition and diversity. Sustain. Divers. For. Ecosyst. An Interdiscip. Approach 361–371. https://doi.org/10.1007/978-4-431-73238-9_1

Huang, C., Goward, S.N., Masek, J.G., Gao, F., Vermote, E.F., Thomas, N., Schleeweis, K., Kennedy, R.E., Zhu, Z., Eidenshink, J.C., Townshend, J.R.G., 2009. Development of time series stacks of landsat images for reconstructing forest disturbance history. Int. J. Digit. Earth 2, 195–218. https://doi.org/10.1080/17538940902801614

Huang, C., Goward, S.N., Masek, J.G., Thomas, N., Zhu, Z., Vogelmann, J.E., 2010. An automated approach for reconstructing recent forest disturbance history using dense Landsat time series stacks. Remote Sens. Environ. 114, 183–198. https://doi.org/10.1016/j.rse.2009.08.017

Huete, A., Didan, K., Miura, T., Rodriguez, E.., Gao, X., Ferreira, L.., 2002. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. Remote Sens. Environ. 83, 195–213. https://doi.org/10.1016/S0034-4257(02)00096-2

Idris, M.H., Kuraji, K., Suzuki, M., 2005. Evaluating vegetation recovery following large-scale forest fires in Borneo and northeastern China using multi-temporal NOAA/AVHRR images. J. For. Res. 10, 101–111. https://doi.org/10.1007/s10310-004-0106-y

Justice, C.O., Townshend, J.R.G., Holben, A.N., Tucker, C.J., 1985. Analysis of the phenology of global vegetation using meteorological satellite data. Int. J. Remote Sens. 6, 1271–1318. https://doi.org/10.1080/01431168508948281

Kennedy, R.E., Yang, Z., Cohen, W.B., 2010. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr - Temporal segmentation algorithms. Remote Sens. Environ. 114, 2897–2910. https://doi.org/10.1016/j.rse.2010.07.008

Kennedy, R.E., Yang, Z., Cohen, W.B., Pfaff, E., Braaten, J., Nelson, P., 2012. Spatial and temporal patterns of forest disturbance and regrowth within the area of the Northwest Forest Plan. Remote Sens. Environ. 122, 117–133. https://doi.org/10.1016/j.rse.2011.09.024

Key, C.H., Benson, N.C., 2006. Landscape Assessment (LA) Sampling and Analysis Methods.

FIREMON Fire Eff. Monit. Invent. Syst. Integr. Stand. F. Data Collect. Tech. Sampl. Des. With Remote Sens. to Assess Fire Eff. 1–51.

Lebrija-trejos, E., Bongers, F., 2016. Successional Change and Resilience of a Very Dry Tropical Deciduous Forest following Shifting Agriculture Author (s): Edwin Lebrija-Trejos , Frans Bongers , Eduardo A . Pérez-García and Jorge A . Published by : Association for Tropical Biology and Conse 40, 422–431.

Lentile, L.B., Holden, Z. a, Smith, A.M.S., Falkowski, M.J., Hudak, A.T., 2006. Remote sensing techniques to assess active fire characteristics and post fire effects. USDA For. Serv. / UNL Fac. Publ. 194, 319–345.

Masek, J.G., Goward, S.N., Kennedy, R.E., Cohen, W.B., Moisen, G.G., Schleeweis, K., Huang, C., 2013. United States Forest Disturbance Trends Observed Using Landsat Time Series. Ecosystems 16, 1087–1104. https://doi.org/10.1007/s10021-013-9669-9

Masek, J.G., Huang, C., Wolfe, R., Cohen, W., Hall, F., Kutler, J., Nelson, P., 2008. North American forest disturbance mapped from a decadal Landsat record. Remote Sens. Environ. 112, 2914–2926. https://doi.org/10.1016/j.rse.2008.02.010

Masek, J.G., Vermote, E.F., Saleous, N.E., Wolfe, R., Hall, F.G., Huemmrich, K.F., Gao, F., Kutler, J., Lim, T., 2006. A Landsat Surface Reflectance Dataset for North America, 1999-2000. IEEE Geosci. Remote Sens. Lett. 3, 68–72. https://doi.org/10.1016/S0168-1273(87)10004-9

Melaas, E.K., Friedl, M.A., Zhu, Z., 2013. Detecting interannual variation in deciduous broadleaf forest phenology using Landsat TM/ETM+ data. Remote Sens. Environ. 132, 176–185. https://doi.org/10.1016/j.rse.2013.01.011

Moisen, G.G., Meyer, M.C., Schroeder, T.A., Liao, X., Schleeweis, K.G., Freeman, E.A., Toney, C., 2016. Shape selection in Landsat time series: a tool for monitoring forest dynamics. Glob. Chang. Biol. 22, 3518–3528. https://doi.org/10.1111/gcb.13358

Nowacki, G.J., Abrams, M.D., 2008. The Demise of Fire and " Mesophication " of Forests in the Eastern United States. Source Biosci. 58, 123–138. https://doi.org/10.1641/b580207

Reed, B.C., Brown, J.F., VanderZee, D., Loveland, T.R., Merchant, J.W., Ohlen, D.O., 2006. Measuring phenological variability from satellite imagery. J. Veg. Sci. 5, 703–714. https://doi.org/10.2307/3235884

Roy, D.P., Yan, L., 2018. Robust Landsat-based crop time series modelling. Remote Sens. Environ. 0–1. https://doi.org/10.1016/j.rse.2018.06.038

Ruefenacht, B., Finco, M.V., Nelson, M.D., Czaplewski, R., Helmer, E.H., Blackard, J.A., Holden, G.R., Lister, A.J., Salajanu, D., Weyermann, D., Winterberger, K., 2008. Conterminous U.S. and Alaska Forest Type Mapping Using Forest Inventory and Analysis Data. Photogramm. Eng. Remote Sens. 74, 1379–1388. https://doi.org/10.14358/PERS.74.11.1379

Senf, C., Pflugmacher, D., Wulder, M.A., Hostert, P., 2015. Characterizing spectral-temporal patterns of defoliator and bark beetle disturbances using Landsat time series. Remote Sens. Environ. 170, 166–177. https://doi.org/10.1016/j.rse.2015.09.019

South Carolina Forestry Commission, 2010. South Carolina's Statewide Forest Resource Assessment and Strategy: Conditions, Trends, Threats, Benefits, and Issues 271.

Stanturf, J.A., Wade, D.D., Waldrop, T.A., Kennard, D.K., Achtemeiser, G.L., 2002. Fire in Southern Forest Landscape, in Southern forest resource assessment. In David N. Wear & John G. Greis (Eds.), Gen. Tech. Rep. SRS-53 (pp. 635). Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station.

Swanson, M.E., Franklin, J.F., Beschta, R.L., Crisafulli, C.M., DellaSala, D. a, Hutto, R.L., Lindenmayer, D.B., Swanson, F.J., 2011. The forgotten stage of forest succession: Early-successional ecosystems on forest sites. Front. Ecol. Environ. 9, 117–125. https://doi.org/10.1890/090157

van Breemen, N., Finzi, A.C., Canham, C.D., 2011. Canopy tree-soil interactions within temperate forests: effects of soil elemental composition and texture on species distributions. Can. J. For. Res. https://doi.org/10.1139/x97-061

van Leeuwen, W.J.D., 2008. Monitoring the Effects of Forest Restoration Treatments on Post-Fire Vegetation Recovery with MODIS Multitemporal Data. Sensors 8, 2017–2042. https://doi.org/10.3390/s8032017

van Leeuwen, W.J.D., Davison, J.E., Casady, G.M., Marsh, S.E., 2010. Phenological characterization of desert sky island vegetation communities with remotely sensed and climate time series data. Remote Sens. 2, 388–415. https://doi.org/10.3390/rs2020388

Varner, J.M., Gordon, D.R., Putz, F.E., Hiers, J.K., 2005. Restoring Fire to Long-Unburned Pinus palustris Ecosystems : Novel Fire Effects and Consequences for Long-Unburned Ecosystems 13, 536–544.

Vermote, E.F., El Saleous, N., Justice, C.O., Kaufman, Y.J., Privette, J.L., Remer, L., Roger, J.C., Tanré, D., 1997. Atmospheric correction of visible to middle-infrared EOS-MODIS data over land surfaces: Background, operational algorithm and validation. J. Geophys. Res. Atmos. 102, 17131–17141. https://doi.org/10.1029/97JD00201

Waldrop, T.A., White, D.L., Jones, S.M., 1992. Fire regimes for pine grassland communities in the southeastern United States. For. Ecol. Manage. 47, 195–210.

https://doi.org/10.1016/0378-1127(92)90274-d

White, M.A., de Beurs, K.M., Didan, K., Inouye, D.W., Richardson, A.D., Jensen, O.P., O'Keefe, J., Zhang, G., Nemani, R.R., van Leeuwen, W.J.D., Brown, J.F., de Wit, A., Schaepman, M., Lin, X., Dettinger, M., Bailey, A.S., Kimball, J., Schwartz, M.D., Baldocchi, D.D., Lee, J.T., Lauenroth, W.K., 2009. Intercomparison, interpretation, and assessment of spring phenology in North America estimated from remote sensing for 1982-2006. Glob. Chang. Biol. 15, 2335–2359. https://doi.org/10.1111/j.1365-2486.2009.01910.x

White, M.A., Thornton, P.E., Running, S.W., 1997. A continental phenology model for monitoring vegetation responses to interannual climatic variability. Global Biogeochem. Cycles 11, 217–234. https://doi.org/10.1029/97GB00330

Zhang, X., Friedl, M.A., Schaaf, C.B., Strahler, A.H., Hodges, J.C.F., Gao, F., Reed, B.C., Huete, A., 2003. Monitoring vegetation phenology using MODIS. Remote Sens. Environ. 84, 471–475. https://doi.org/10.1016/S0034-4257(02)00135-9

Zhu, Z., Woodcock, C.E., 2014. Continuous change detection and classification of land cover using all available Landsat data. Remote Sens. Environ. 144, 152–171. https://doi.org/10.1016/j.rse.2014.01.011

Appendix

Script workflow

1. clean_mtbs_code.R: This script creates a single burn shapefile for the MTBS polygons in Landsat path 16 row 37, 1984-2015
2. phenology_mtbs_05.R: This script includes the first steps for data processing of the MTBS polygons (downloaded from mtbs.gov) and creates 1.) MTBS polygons for the Landsat scene path 16 row 37 only, 1984-2015: *mtbs_poly,* 2). Rasterized version of MTBS polygons: *mtbs_rasterize_poly,* 3.) A mask of areas that were burned multiple times: *mtbs_mask,* 4.) Shapefile for the clean MTBS polygons after 2003: *clean_mtbs_2003*
3. phenology_mtbs_07.R: This script calls the MTBS raster files (downloaded from mtbs.gov) and creates a data frame of burned locations with fire attributes derived from the MTBS polygons and fire severity, 2003-2015, and forest group designation from the U.S. Forest Service raster maps.
4. phenology_mtbs_10.R: This script generates the control samples from the 5 kilometer buffer regions surrounding the MTBS polygons, 1000 samples in loblolly-shortleaf pine and 1000 samples in oak-gum-cypress. Forest designations are based on the U.S. Forest Service raster maps (data.fs.usda.gov). All burned areas are masked out using the mtbs_rasterize_poly created in phenology_mtbs_05.R.
5. adding_pixels.R.: This script 1.) Creates a common extent for the Landsat rasters based on the extent of the smallest image. 2.) Samples 1000 burned pixels from each forest group (2000 burned samples). 3.) Calls the burned and unburned sample locations and calculates the cell id's of the 8 surrounding pixels for patch construction. The final outputs are two CSV files for the burned and unburned sample pixels that include raster

cell id, a pixel id, row, column, fire attributes, and forest group classification.

6. Landsat_extraction_functions.R: This script calls the CSV files created in adding_pixels.R for the burned and unburned samples. This code loops through each Landsat image and extracts the value at each sample location. The function also ensures that the fire attribute and forest information are retained in the processing. The outputs are two CSV files (16GB/file) that include all of the raw Landsat, forest group, and fire attribute data needed for analysis.
7. Tidy_landsat_dfs.R: This script calls the Landsat value extraction files from the previous script. Using the 'tidyverse' package and Landsat information, cloudy pixels are removed from the dataset and spectral indices are calculated from the raw data. The data are also organized by sensor (Landsat 5, 7 and 8). The final output is the dataset used for the rest of the analyses and includes NDVI, NBR, fire information, and forest designation aggregated to the patch level.
8. Phen_funs.R: This script includes all of the necessary functions for constructing phenology curves using penalized least squares regression ('pcls' within the 'mgv' package). Functions included: 1.) matrix_fun: takes arguments for time, k (number of sine/cosine cycles allowed within 365 days). 2.) xmat_fun: takes argument t (time) and generates sine and cosine X-matrices based on given time (xmat_mat_fun and xmat_burn_fun are variations of this function that also take burn status variables as arguments). 3.) penalty_fun: takes argument t and constructs penalty matrices based on the sine/cosine derivatives (pen_mat_fun and pen_burn_fun are variations of this function that also take burn status variables as arguments). 4.) pen_mat_sd, pen_mat_sd2, pen_matsd3: outputs penalty matrices similar to the pervious penalty_fun series but

multiplies the end product by the selected smoothing parameters. The output is used to estimate the standard deviation of the phenology curves. 5.) pred_sd and pred_se: functions that calculate the predicted standard deviation and standard error (respectively) of a given prediction X matrix of sine and cosine values. In this case, the prediction matrix covers an annual time period. 6.) beta_fun2 and beta_fun3: inputs include beta, the coefficients produced by pcls, prediction X matrix, and t (time). The first function, beta_fun2, is used for pre- and post-fire phenology estimates and outputs a data frame of predicted pre-fire, post-fire, and change for a spectral index. The second function, beta_fun3, is used for constructing time step based phenology estimates (pre-fire, within 2 years post fire, more than 2 years post fire, and change metrics) and outputs a data frame similar to the first function. This script serves as the source code for each modelling script.

9. Thirteen separate scripts were developed to construct phenology curves for various subset of data, based on forest group, fire severity, and burn status. These scripts use the functions developed in phen_funs.R. The general flow includes: 1.) constructing the X-matrix for the focal data set, 2.) constructing the penalty matrices for the focal data set, 3.) generating a list of inputs for 'pcls' based on the focal data set which include the response variable (nbr or ndvi), a weight matrix, the global X-matrix, a matrix of linear equality constraints (0 for all models), a list of penalty matrices, offset values (0 for all models), smoothing parameters, matrix for inequality constraints based on the dimensions of the X-matrix, and a vector for inequality constraints (0 for all models). From the coefficients produced by 'pcls', predictions for the given spectral index (NDVI or NBR) are added to the original dataset. A unique penalty matrix is constructed for calculating

the standard deviation and standard errors of the prediction matrix outputs. The standard errors are added to the prediction data frames. Finally, the figures used in the thesis document are displayed using ‘ggplot’. For length purposes, only one of these scripts is included in the appendix, lob_models_ndvi.R which generates the figures for pre- and post-fire loblolly-shortleaf pine samples. The script file names: lob_models_nbr.R, lob_models_nbr_unburn.R, lob_model_ndvi.R, lob_models_ndvi_unburn.R, lob_severity_models_nbr.R, lob_severity_models_ndvi.R, oak_models_nbr.R, oak_models_nbr_unburn.R, oak_models_ndvi.R, oak_models_ndvi_unburn.R, oak_severity_models_nbr.R, oak_severity_models_ndvi.R, species_difference_models_ndvi.R, and species_difference_models_ndvi_unburned.R

10. Phenometrics.R: This script calls all of the phenology curves datasets generated in the model scripts for each subset of data. The function phen_metric_fun1 takes a phenology data frame as an input and outputs the following metrics for the given spectral index: maximum value, minimum, date of maximum, date of minimum, date of spring onset, and date of senescence. The function is applied to each data set to extract phenology metrics.

R Code

Clean_mtbs_code.R

```
# This code creates a "clean" mtbs SpatialPolygonsDataFrame file that includes
# areas only burned once in Landsat scene path 16 row 37
library(rgdal)
library(sp)
library(sf)
setwd('/data/landsat/SC')
##list of all landsat files scene path 16 row 37
file.list = list.files('/data/landsat/SC/', pattern = "(sr_band|bt_band|pixel_qa|radsat_qa)")
# calling in a sample Landsat scene to transform the projection of mtbs
landsat_raster <- raster(file.list[[1]])
setwd("/data/SC/")
# calling polygon for mtbs in landsat scene path 16 row 37
mtbs_polygon <- readOGR("MTBS_polygon/p16r37_mtbs_poly.shp")
# transforming mtbs polygon projection to match landsat raster
mtbs_poly <- spTransform(mtbs_polygon, proj4string(landsat_raster))
# buffering the mtbs polygon to redraw the lines
mtbs_buffer <- rgeos::gBuffer(mtbs_poly, byid=TRUE, width=0)
# checking the geometry of mtbs_buffer
rgeos::gIsValid(mtbs_buffer)
# Nagle code for determing the intersecting polygons with mtbs
# outputs a SpatialPolygonsDataFrame for "clean" burns: areas only burned once
mtbs_sf <- as(mtbs_buffer, 'sf')
test_sf <- lapply(1:nrow(mtbs_sf), function(x) st_difference(mtbs_sf[x,], st_union(mtbs_sf[-
x,])))
test2_sf <- do.call(rbind, test_sf)
st_write(test2_sf, "MTBS_polygon/clean_mtbs_poly_sf.shp")
#clean_data <- as(test2_sf, 'Spatial')
#writeOGR(clean_data, "MTBS_polygon", layer = "clean_mtbs_poly.shp", driver = "ESRI
Shapefile")
```

Phenology_mtbs_05.R

```
# This script creates
# 1. MTBS polygons for the landsat scene path 16 row 37 only, 1984-2015: mtbs_poly
# 2. Raster file from full polygon file (for masking out burned areas): mtbs_rasterize_poly
# 3. A mask for areas that were burned multiple times: the difference of the original burn
polygons
# and the clean mtbs polygons created in the clean_mtbs_code in R: mtbs_mask
# This will be used for no-go areas for our random samples
# 4. Spatial Polygons Data Frame for the clean MTBS polygons 2003 and after:
clean_mtbs_2003
```

5. Polygon for 20 largest fires that occurred after 2003

NOTE THE CLEAN POLYGON GENERATED IN THIS FILE CONTAINS 191 FIRES, HOWEVER THE FINAL DATASET
INCLUDES DATA FROM 125 FIRES AFTER PROCESSING AND CLEANING

```
library(tidyverse)
library(dplyr)
library(gdalUtils)
library(rgdal)
library(raster)
library(rgeos)

# working directory for calling landsat files
setwd('/data/landsat/SC')

##list of all landsat files scene path 16 row 37
file.list = list.files('/data/landsat/SC/', pattern = "(sr_band|bt_band|pixel_qa|radsat_qa)")

# calling in a sample Landsat scene to convert the projection of mtbs and forest group files
# to that of the landsat raster file
landsat_raster <- raster(file.list[[1]])

# setting to working directory for other files
setwd('/data/brooke_thesis')

# cropping mtbs polygon to extent of landsat path 16 row 37
if(!file.exists("MTBS_created/polygons/p16r37_mtbs_poly.shp")){
# calling in mtbs shapefile for 1984-2015, entire US
# downloaded on March 21, 2018
  mtbs_poly <- readOGR("original_mtbs_files/mtbs_polygon/mtbs_perims_1984-
2015_DD_20170815.shp")
  mtbs_poly <- spTransform(mtbs_poly, proj4string(landsat_raster))
# cropping mtbs_poly to landsat_raster extent
  mtbs_crop <- crop(mtbs_poly, landsat_raster)
  mtbs_crop <- rgeos::gBuffer(mtbs_crop, byid=TRUE, width=0)
  mtbs_crop <- spTransform(mtbs_crop, proj4string(landsat_raster))
  writeOGR(mtbs_crop, "MTBS_created/polygons/", layer = "p16r37_mtbs_poly", driver =
"ESRI Shapefile",
          overwrite_layer=TRUE)
}
# calling the full mtbs polygon generated from code above
# extent is landsat path 16 row 37
mtbs_poly <- readOGR("MTBS_created/polygons/p16r37_mtbs_poly.shp")

# creating a raster of the cropped mtbs polygon to filter
```

```
# out any burned areas from the unburned sample areas
if(!file.exists("MTBS_created/rasters/mtbs_rasterize_poly")){
  mtbs_full_rast <- rasterize(mtbs_poly, landsat_raster)
  writeRaster(mtbs_full_rast, "MTBS_created/rasters/mtbs_rasterize_poly.img")
}
mtbs_full_rast <- raster("MTBS_created/rasters/mtbs_rasterize_poly.img")

# calling in clean mtbs polygon from R file "clean_mtbs_code"
# contains only areas that were burned once in path 16 row 37
clean_mtbs <- readOGR("MTBS_created/polygons/clean_mtbs_poly_sf.shp")
clean_mtbs <- rgeos::gBuffer(clean_mtbs, byid=TRUE, width=0)

# mask of the clean polygons on the original mtbs to determine
# areas that were burned more than once
if(!file.exists("MTBS_created/polygons/mtbs_doubleburn_mask.shp")){
  mtbs_mask <- mtbs_poly - clean_mtbs
  mtbs_mask <- spTransform(mtbs_mask, proj4string(landsat_raster))
  writeOGR(mtbs_mask, "MTBS_created/polygons/", layer ="mtbs_doubleburn_mask", driver =
"ESRI Shapefile",
           overwrite_layer=TRUE)
}
# don't actually need this for anything other than to validate the clean mtbs
mtbs_mask <- readOGR("MTBS_created/polygons/mtbs_doubleburn_mask.shp")

#creating a raster for the mtbs_mask
if(!file.exists("MTBS_created/rasters/mtbs_doubleburn_mask.img")){
  mtbs_mask_rast <- rasterize(mtbs_mask, landsat_raster)
  writeRaster(mtbs_mask_rast, "MTBS_created/rasters/mtbs_doubleburn_mask.img")
}
mtbs_mask_rast <- raster("MTBS_created/rasters/mtbs_doubleburn_mask.img")

# selecting fires that occurred 2003 or after
# and top 20 largest fires for this time period

clean_mtbs_data <- clean_mtbs@data
clean_mtbs_data$Acres <- as.numeric(paste(clean_mtbs_data$Acres))
# ordering fire info based on size
fire_df <- clean_mtbs_data[order(-clean_mtbs_data$Acres),]
# selecting the 20 largest fires
big_fires <- head(fire_df, 25) %>%
  dplyr::select(Fire_ID)
fire.list <- as.vector(big_fires)
fire.list2 <- as.character(fire.list$Fire_ID)
if(!file.exists("MTBS_created/polygons/clean_mtbs_2003.shp")){
  clean_mtbs_2003 <- clean_mtbs[clean_mtbs$Year>=2003,]
  big_fire_2003 <- clean_mtbs_2003[clean_mtbs_2003$Fire_ID %in% fire.list2,]
```

```
  writeOGR(clean_mtbs_2003, "MTBS_created/polygons/", layer = "clean_mtbs_2003", driver =
"ESRI Shapefile",
         overwrite_layer = TRUE)
  writeOGR(big_fire_2003, "MTBS_created/polygons", layer = "big_fire_post2003", driver =
"ESRI Shapefile",
         overwrite_layer = TRUE)
}
clean_mtbs_2003 <- readOGR("MTBS_created/polygons/clean_mtbs_2003.shp")
big_fire_2003 <- readOGR("MTBS_created/big_fire_post2003.shp")
```

Phenology_mtbs_07.R

```
# This file loops through the mtbs raster files and creates a data frame of burned points with fire
# attributes using st_join,
# also includes burn severity (cellvalue) from the raster files using as_tibble
## in final data frame, attribute "cellvalue" is burn severity, others are self-explanatory

library(tidyverse)
library(dplyr)
library(gdalUtils)
library(rgdal)
library(raster)
library(rgeos)
library(tabularaster)
library(sf)

setwd("/data/brooke_thesis")

# test mtbs raster
mtbs_2004_rast <- raster("original_mtbs_files/mtbs_rasters/mtbs_SC_2004.tif")

# clean mtbs polygons 2003-2015
mtbs_clean <- st_read("MTBS_created/polygons/clean_mtbs_2003.shp")
mtbs_clean <- st_transform(mtbs_clean, proj4string(mtbs_2004_rast))

# testing sf method for joining polygon attributes to a sf points object
# filtering for 2004 fires
#mtbs_clean_2004 <- mtbs_clean %>%
  #filter(Year == 2004)
# test sf format on mtbs 2004 raster to locs
#mtbs_2004_df <- as_tibble(mtbs_2004_rast, xy=TRUE) %>% drop_na() %>%
mutate(year='2004')
# sfc_POINT object generated from mtbs 2004 raster tibble
#locs <- st_as_sf(mtbs_2004_df, coords = c('x','y'), crs=proj4string(mtbs_2004_rast),
agr="constant")
# st_join joins based on geometry and attaches attributes of y to x
```

```
#joined <- st_join(x = locs, y = mtbs_clean_2004) %>%
  #na.omit()
#plot(joined)

# testing extract with sf locs and mtbs_2004_rast
# extract returns a SpatialPointsDataFrame
#test_ex <- raster::extract(mtbs_2004_rast, joined, df=TRUE, sp=TRUE)
# converting back to an sf object
#test_ex_sf <- st_as_sf(test_ex)

# where do the extract and original cell value agree?
# confirms that these are points and they can be used to extract values from a raster
#agree <- test_ex_sf %>%
  #filter(cellvalue == mtbs_SC_2004)

# preparing data for the loop

# vector with study years
years <- c('2003', '2004', '2005', '2006', '2007', '2008', '2009',
           '2010', '2011', '2012', '2013', '2014', '2015')
# vector for the tibbles of the x,y points from the mtbs raster images
burned_pixels <- vector('list')
# mtbs polygons for each year 2003-2015, vector
mtbs_loop <- vector('list')
# joins of the burned pixel coordinates (locs_sf) to mtbs loop
join <- vector('list')

for (yr in as.list(years)){
  raster_file <-  paste0('original_mtbs_files/mtbs_rasters/mtbs_SC_', yr, '.tif' )
  mtbs <- raster(raster_file)
  burned_pixels[[yr]] <- as_tibble(mtbs, xy =TRUE) %>%
    drop_na() %>%
    mutate(year = yr)
  locs_sf <- st_as_sf(burned_pixels[[yr]], coords = c('x','y'), crs=proj4string(mtbs_2004_rast),
agr="constant")
  mtbs_loop[[yr]] <- mtbs_clean %>%
    filter(Year == yr)
  join[[yr]] <- st_join(x=locs_sf, y = mtbs_loop[[yr]]) %>% na.omit()
}

# method for visualizing sf data
plot(st_geometry(join$`2005`))

# converts the join list of sf locations to a single data frame
join_df <- do.call(rbind.data.frame, join) %>%
  dplyr::select(-year)
```

```
# forest group raster
forest_gp <- raster("predictors/p16r37_forest_group.img")
join_df_tf <- st_transform(join_df, proj4string(forest_gp))

# nlcd rasters
nlcd2001 <- raster("predictors/p16r37_nlcd_2001.img")
nlcd2006 <- raster("predictors/p16r37_nlcd_2006.img")
nlcd2011 <- raster("predictors/p16r37_nlcd_2011.img")

# extracting forest group and nlcd classifications at the burned locations
extract1 <- raster::extract(forest_gp, join_df_tf, sp =TRUE)
extract2 <- raster::extract(nlcd2001, extract1, sp=TRUE)
extract3 <- raster::extract(nlcd2006, extract2, sp=TRUE)
extract4 <- raster::extract(nlcd2011, extract3, sp=TRUE)

extract_final <- st_as_sf(extract4)
names(extract_final) <- c("severity", "cellindex", "Fire_ID", "Fire_Name", "Year", "ig_date",
                   "Acres", "FireType", "forest", "nlcd2001", "nlcd2006", "nlcd2011",
                   "geometry")


# writes burn locations from join loop to csv file
# layer_options = "GEOMETRY=AS_XY" writes x and y as columns
st_write(extract_final, "MTBS_created/burned_points_from_mtbs_rasters.csv",
      layer_options = "GEOMETRY=AS_XY", delete_dsn=TRUE)

st_write(extract_final, "MTBS_created/burned_points_from_mtbs_rasters.shp",
      delete_layer = TRUE)

mtbs_raster_points <- st_read("MTBS_created/burned_points_from_mtbs_rasters.shp")
```

Phenology_mtbs_10.R

```
# 1. Buffer file for the clean mtbs polygons 2003 and after
# 2. Raster for the buffers created in #1., then inverse masked to burn areas,
# creating a file that only includes the unburned buffer areas around each fire
# 3. A raster file of unburned buffer areas masked to loblolly forests
# 4. Random sample of ~800 points from unburned buffer areas of loblolly
# 5. A raster file of unburned buffer areas masked to oak forests
# 6. Random sample of ~800 points from unburned buffer areas of oak

library(tidyverse)
library(gdalUtils)
library(rgdal)
```

```
library(raster)
library(rgeos)
library(sf)

# working directory for calling landsat files
setwd('/data/landsat/SC')
##list of all landsat files scene path 16 row 37
file.list = list.files('/data/landsat/SC/', pattern = "(sr_band|bt_band|pixel_qa|radsat_qa)")
# calling in a sample Landsat scene to convert the projection of mtbs and forest group files
# to that of the landsat raster file
landsat_raster <- raster(file.list[[1]])

# setting to working directory for other files
setwd('/data/brooke_thesis')

# common fire id's for comparing burned and unburned samples
study_fires <- read_csv("MTBS_created/study_fire_ids.csv")

# single burn mtbs fires after 2003
clean_mtbs_2003 <- st_read("MTBS_created/polygons/clean_mtbs_2003.shp") %>%
  filter(Fire_ID %in% study_fires$Fire_ID)
# top ~20 biggest fires, single burn post 2003
big_fires_2003 <- st_read("MTBS_created/polygons/big_fire_post2003.shp")

# creating a 5 km buffer for each clean mtbs polygon, 2003 and after
if(!file.exists("MTBS_created/polygons/5km_mtbs_clean_buff.shp")){
  clean_buff <- st_buffer(clean_mtbs_2003, dist = 5000)
  st_write(clean_buff, "MTBS_created/polygons/5km_mtbs_clean_buff.shp",
           delete_layer = TRUE)
}
clean_buff <- st_read("MTBS_created/polygons/5km_mtbs_clean_buff.shp")

# creating a 5 km buffer for each clean mtbs polygon, post-2003, for the largest 20 fires
if(!file.exists("MTBS_created/polygons/5km_mtbs_clean_buff_big_fires.shp")){
  clean_buff_bf <- st_buffer(big_fires_2003, dist=5000)
  st_write(clean_buff_bf, "MTBS_created/polygons/5km_mtbs_clean_buff_big_fires.shp",
           delete_layer = TRUE)
}
big_fires_buff <- st_read("MTBS_created/polygons/5km_mtbs_clean_buff_big_fires.shp")

# raster of all mtbs fires, to mask out any previously burned areas
# from the unburned buffers
mtbs_full_rast <- raster("MTBS_created/rasters/mtbs_rasterize_poly.img")

# rasterizing the clean buffer file
buff_rast <- rasterize(clean_buff, landsat_raster)
```

```
unburn_attributes <- as.data.frame(levels(buff_rast)[[1]])
write_csv(unburn_attributes, "MTBS_created/locations/unburn_sample_fire_attributes.csv")
writeRaster(buff_rast, "MTBS_created/rasters/5km_clean_buffer.img", overwrite =TRUE)

if(!file.exists("MTBS_created/rasters/5km_mtbs_unburn_mask.img")){
  # removing all burned areas using the full mtbs raster
  unburn_masked <- mask(buff_rast, mtbs_full_rast, inverse= TRUE)
  writeRaster(unburn_masked, "MTBS_created/rasters/5km_mtbs_unburn_mask.img",
overwrite=TRUE)
}
unburn_mask <- raster("MTBS_created/rasters/5km_mtbs_unburn_mask.img")

# buff_rast_bf = rasterized 5km buffer area for 20 largest fires
# unburn_mask_bf = rasterized area of 5km buffers around the 20 largest fires, unburned
buff_rast_bf <- rasterize(big_fires_buff, landsat_raster)
unburn_attributes_bf <- as.data.frame(levels(buff_rast_bf)[[1]])
writeRaster(buff_rast_bf, "MTBS_created/rasters/5km_clean_buffer_big_fires.img")

if(!file.exists("MTBS_created/rasters/5km_mtbs_unburn_mask_big_fires.img")){
  # removing all burned areas using the full mtbs raster
  unburn_masked_bf <- mask(buff_rast_bf, mtbs_full_rast, inverse= TRUE)
  writeRaster(unburn_masked_bf,
"MTBS_created/rasters/5km_mtbs_unburn_mask_big_fires.img", overwrite=TRUE)
}
unburn_mask_bf <- raster("MTBS_created/rasters/5km_mtbs_unburn_mask_big_fires.img")

# loblolly and oak rasters created in phenology_mtbs_05.R
# from the USFS forest group maps
loblolly_rast <- raster("Forest_groups/loblolly_rast.img")
oak_rast <- raster("Forest_groups/oak_gum_cypress.img")

if(!file.exists("MTBS_created/rasters/unburned_loblolly.img")){
  # cropping unburned mask to loblolly raster extent
  unburn_crop_lob <- crop(unburn_mask, loblolly_rast)
  unburn_loblolly <- mask(unburn_crop_lob, loblolly_rast)
  writeRaster(unburn_loblolly, "MTBS_created/rasters/unburned_loblolly.img", overwrite =
TRUE)
}
unburn_loblolly <- raster("MTBS_created/rasters/unburned_loblolly.img")

#loblolly unburn mask for big fires post 2003
if(!file.exists("MTBS_created/rasters/unburned_loblolly_big_fires.img")){
  # cropping unburned mask to loblolly raster extent
  unburn_crop_bf <- crop(unburn_mask_bf, loblolly_rast)
  unburn_loblolly_bf <- mask(unburn_crop_bf, loblolly_rast)
  writeRaster(unburn_loblolly_bf, "MTBS_created/rasters/unburned_loblolly_big_fires.img")
```

```
}
unburn_loblolly_bf <- raster("MTBS_created/rasters/unburned_loblolly_big_fires.img")

# generating random sample points for unburned loblolly forests
if(!file.exists("MTBS_created/locations/loblolly_unburn_points.shp")){
  # the size parameter gives enough locations for roughly 1000 non NA samples
  unburn_loblolly_sp <- raster::sampleRandom(unburn_loblolly, size=1000, sp = TRUE,
na.rm=TRUE, xy=TRUE)
  unburn_loblolly_sf <- as(unburn_loblolly_sp, "sf")
  st_write(unburn_loblolly_sf, "MTBS_created/locations/loblolly_unburned_locs.shp",
           delete_layer = TRUE)
}

unburn_lob_sf <- st_read("MTBS_created/locations/loblolly_unburned_locs.shp")

# fire_id is the simplified id that is introudced when the buffer files are
# rasterized
if(!file.exists("MTBS_created/locations/loblolly_unburn_locs_attributes.csv")){
  unburn_loblolly_sf <- st_read("MTBS_created/locations/loblolly_unburned_locs.shp")
  names(unburn_loblolly_sf)[3] <- paste("fire_id")
  lob_unburn_locs <- left_join(unburn_loblolly_sf, unburn_attributes, by = c("fire_id"="ID"))
  st_write(lob_unburn_locs, "MTBS_created/locations/loblolly_unburn_locs_attributes.shp",
delete_layer = TRUE)
}

# unburned loblolly forest post 2003, largest fires
# generating random sample points for unburned loblolly forests
if(!file.exists("MTBS_created/locations/loblolly_unburn_points_big_fires.shp")){
  unburn_loblolly_sp_bf <- raster::sampleRandom(unburn_loblolly_bf, size=1000, sp = TRUE,
na.rm=TRUE, xy=TRUE)
  unburn_loblolly_sf_bf <- as(unburn_loblolly_sp_bf, "sf")
  st_write(unburn_loblolly_sf_bf,
"MTBS_created/locations/loblolly_unburn_points_big_fires.shp",
           delete_layer = TRUE)
}

# joining the data from the unburned loblolly samples to the mtbs attribute information
# and then outputing as a csv file
# for big fires
if(!file.exists("MTBS_created/locations/loblolly_unburn_locs_attributes_big_fires.csv")){
  unburn_loblolly_sf_bf <-
st_read("MTBS_created/locations/loblolly_unburn_points_big_fires.shp")
  names(unburn_loblolly_sf_bf)[3] <- paste("fire_id")
  lob_unburn_locs_bf <- left_join(unburn_loblolly_sf_bf, unburn_attributes_bf, by =
c("fire_id"="ID"))
  st_write(lob_unburn_locs_bf,
```

```
"MTBS_created/locations/loblolly_unburn_locs_attributes_big_fires.shp")
}

unburn_lob <-
read_csv("MTBS_created/locations/loblolly_unburn_locs_attributes_big_fires.csv")

# unburned oak forest
if(!file.exists("MTBS_created/rasters/unburned_oak.img")){
  unburn_oak_crop <- crop(unburn_mask, oak_rast)
  unburn_oak <- mask(unburn_oak_crop, oak_rast)
  writeRaster(unburn_oak, "MTBS_created/rasters/unburned_oak.img", overwrite=TRUE)
}
unburn_oak <- raster("MTBS_created/rasters/unburned_oak.img")

# generating random sample points for unburned oak forests for all fires
if(!file.exists("MTBS_created/locations/oak_unburn_points.shp")){
  # the size parameter gives enough locations for 1000 non NA samples
  unburn_oak_sp <- raster::sampleRandom(unburn_oak, size=1000, sp = TRUE, na.rm=TRUE,
xy=TRUE)
  unburn_oak_sf <- as(unburn_oak_sp, "sf")
  st_write(unburn_oak_sf, "MTBS_created/locations/oak_unburn_points.shp",
           delete_layer = TRUE)
}

if(!file.exists("MTBS_created/locations/oak_unburn_locs_attributes.csv")){
  unburn_oak_sf <- st_read("MTBS_created/locations/oak_unburn_points.shp")
  names(unburn_oak_sf)[3] <- paste("fire_id")
  oak_unburn_locs <- left_join(unburn_oak_sf, unburn_attributes, by = c("fire_id"="ID"))
  st_write(oak_unburn_locs, "MTBS_created/locations/oak_unburn_locs_attributes.shp",
           delete_layer = TRUE)
}

oak_unburn <- read_csv("MTBS_created/locations/oak_unburn_locs_attributes.csv")

#oak unburn mask for big fires post 2003
if(!file.exists("MTBS_created/rasters/unburned_oak_big_fires.img")){
  # cropping unburned mask to loblolly raster extent
  unburn_crop_bf <- crop(unburn_mask_bf, oak_rast)
  unburn_oak_bf <- mask(unburn_crop_bf, oak_rast)
  writeRaster(unburn_oak_bf, "MTBS_created/rasters/unburned_oak_big_fires.img")
}
unburn_oak_bf <- raster("MTBS_created/rasters/unburned_oak_big_fires.img")

#unburned oak points for 20 largest fires buffer areas
# generating random sample points for unburned oak forests
if(!file.exists("MTBS_created/locations/oak_unburn_points_big_fires.shp")){
```

```
  # the size parameter gives enough locations for roughly 1000 non NA samples
  unburn_oak_sp_bf <- raster::sampleRandom(unburn_oak_bf, size=1000, sp = TRUE,
na.rm=TRUE, xy=TRUE)
  unburn_oak_sf_bf <- as(unburn_oak_sp_bf, "sf")
  st_write(unburn_oak_sf_bf, "MTBS_created/locations/oak_unburn_points_big_fires.shp",
          delete_layer = TRUE)
}

# joining the data from the unburned oak samples to the mtbs attribute information
# and then outputing as a csv file
# for big fires
if(!file.exists("MTBS_created/locations/oak_unburn_locs_attributes_big_fires.csv")){
  unburn_oak_sf_bf <- st_read("MTBS_created/oak_unburn_points_big_fires.shp")
  names(unburn_oak_sf_bf)[3] <- paste("fire_id")
  oak_unburn_locs_bf <- left_join(unburn_oak_sf_bf, unburn_attributes_bf, by =
c("fire_id"="ID"))
  st_write(oak_unburn_locs_bf, "MTBS_created/oak_unburn_locs_attributes_big_fires.shp")
}
```

adding_pixels.R

```
# Part 1: Creates a common extent for the landsat rasters
# Part 2: uses the burned and unburned sample points to
# identify the 9 surrounding pixels for each location
# outputs: dataframes for the burned and unburned locations + the 8 extra pixels
# based on cell numbers + row and column, determined using cellFromXY and
# rowRolFromCell using cropped test raster from landsat file.list)

setwd('/data/landsat/SC')
file.list = list.files('/data/landsat/SC/',
                       pattern = "(sr_band|bt_band|pixel_qa|radsat_qa)")
landsat_rasters <- lapply(file.list, raster)
landsat_extents <- lapply(landsat_rasters, extent)

ext2df <- function(x){
  return(data_frame(xmin = x@xmin,
                    xmax = x@xmax,
                    ymin = x@ymin,
                    ymax = x@ymax))
}
landsat_extents <- lapply(landsat_extents, ext2df)
bind_extents <- bind_rows(landsat_extents)
# extents of all raster files and file names
bind_extents$ID <- file.list
```

```
# creating base extent for the landsat
xmin <- max(bind_extents[,1])
xmax <- min(bind_extents[,2])
ymin <- max(bind_extents[,3])
ymax <- min(bind_extents[,4])
bbox <- c(xmin, xmax, ymin, ymax)
base_ext <- extent(bbox)

# cropped test raster
crop_test <- crop(landsat_rasters[[1]], base_ext)

# Part 2

setwd('/data/brooke_thesis')
# burned locations, extracted from mtbs rasters
burned_points <- st_read("MTBS_created/burned_points_from_mtbs_rasters.shp")
# taking a sample of 1000 points for burned oak forests
oak_burn <- burned_points %>%
  filter(forest == 600)
# taking sample of 1000 points for burned loblolly forests
oak_burn_sample <- sample_n(oak_burn, size = 1000)
loblolly_burn <- burned_points %>%
  filter(forest == 160)
loblolly_burn_sample <- sample_n(loblolly_burn, size = 1000)
# binding the oak and loblolly burned samples
burned_samples <- rbind(oak_burn_sample, loblolly_burn_sample)
burn_sp <- as(burned_samples, 'Spatial')
# using crop raster to output a cell id for the locations
burn_cells <- burned_samples %>%
  mutate(cell_id = cellFromXY(crop_test, burn_sp))
# row and column from cell id
burn_df <- as_data_frame(rowColFromCell(crop_test, burn_cells$cell_id)) %>%
  mutate(cell_id = burn_cells$cell_id)

# 8 surrounding pixels added (rows)
burn_row_df <- burn_df %>%
  dplyr::select('row', 'cell_id') %>%
  mutate(P1 = row-1,
         P2 = row-1,
         P3 = row-1,
         P4 = row,
         P5 = row,
         P6 = row,
         P7 = row + 1,
         P8 = row + 1,
         P9 = row + 1)
```

```
# two new keys = pixel_id (P1, P2, etc) and row (the values for P1, P2, etc. above)
burn_rows <- gather(burn_row_df, pixel_id, row, P1:P9)

# columns
# 8 surrounding pixels (columns)
burn_col_df <- burn_df %>%
  dplyr::select('col', 'cell_id') %>%
  mutate(P1 = col-1,
         P2 = col,
         P3 = col+1,
         P4 = col-1,
         P5 = col,
         P6 = col+1,
         P7 = col-1,
         P8 = col,
         P9 = col+1)

# same method used for rows above, now for columns
#adding the cell ID for the surrounding pixels
burn_cols <- gather(burn_col_df, pixel_id, col, P1:P9)

# joining row and column data frames
burn_row_col <- left_join(burn_rows, burn_cols, by = c('cell_id', 'pixel_id'))
## adding sf information back to the col row locations, this info only corresponds to pixel 5
burn_cells_rc <- left_join(burn_row_col, burn_cells, by = 'cell_id')
burn_locs_df <- burn_cells_rc[order(burn_cells_rc$cell_id),] %>%
  mutate(status = 'burned',
         cell_id2 = cellFromRowCol(crop_test, burn_locs_df$row, burn_locs_df$col))

write_csv(burn_locs_df, "MTBS_created/locations/burn_nine_pixel_df.csv")


# at some point, add nlcd classifications to this data set
# created in phenology_mtbs_10.R
unburned_oak_points <- st_read("MTBS_created/locations/oak_unburn_locs_attributes.shp")
%>%
  mutate(forest = 600)
unburned_loblolly_points <-
st_read("MTBS_created/locations/loblolly_unburn_locs_attributes.shp") %>%
  mutate(forest = 160)
unburned_samples <- rbind(unburned_oak_points, unburned_loblolly_points)
unburn_sp <- as(unburned_samples, 'Spatial')


# using raster functions to convert the location files above to cells
```

```
# and then row, column from the test cropped landsat
unburn_cells <- unburned_samples %>%
  mutate(cell_id = cellFromXY(crop_test, unburn_sp))
unburn_row_col <- rowColFromCell(crop_test, unburn_cells$cell_id)
# data frame of unburned row and columns and cell number id
unburn_df <- as_data_frame(unburn_row_col) %>%
  mutate(cell_id = unburn_cells$cell_id)

# unburned points, same method as for burn points
unburn_row_df <- unburn_df %>%
  dplyr::select('row', 'cell_id') %>%
  mutate(P1 = row-1,
         P2 = row-1,
         P3 = row-1,
         P4 = row,
         P5 = row,
         P6 = row,
         P7 = row + 1,
         P8 = row + 1,
         P9 = row + 1)

# two new keys = pixel_id (P1, P2, etc) and row (the values for P1, P2, etc. above)
unburn_rows <- gather(unburn_row_df, pixel_id, row, P1:P9)

# columns
unburn_col_df <- unburn_df %>%
  dplyr::select('col', 'cell_id') %>%
  mutate(P1 = col-1,
         P2 = col,
         P3 = col+1,
         P4 = col-1,
         P5 = col,
         P6 = col+1,
         P7 = col-1,
         P8 = col,
         P9 = col+1)

# same method used for rows above, now for columns
unburn_cols <- gather(unburn_col_df, pixel_id, col, P1:P9)


# joining row and column data frames
unburn_row_col <- left_join(unburn_rows, unburn_cols, by = c('cell_id', 'pixel_id'))
unburn_cells_rc <- left_join(unburn_row_col, unburn_cells, by = 'cell_id')
unburn_locs_df <- unburn_cells_rc[order(unburn_cells_rc$cell_id),] %>%
  mutate(status = 'unburned',
```

```
        cell_id2 = cellFromRowCol(crop_test, unburn_cells_rc$row, unburn_cells_rc$col))

write_csv(unburn_locs_df, "MTBS_created/locations/unburn_nine_pixel_df.csv")

# Figure for presentation
band3 <- raster(file.list[4001])
band4 <- raster(file.list[4002])
ndvi <- (band4-band3)/(band4+band3)
ndvi[ndvi<=0]<-NA


setwd('/data/brooke_thesis')
burn_cells <- read_csv("MTBS_created/locations/burn_nine_pixel_df.csv")
unburn_cells <- read_csv("MTBS_created/locations/unburn_nine_pixel_df.csv")

lon <- xFromCol(ndvi, burn_cells$col)
lat <- yFromRow(ndvi, burn_cells$row)
xy <- cbind(lon, lat)
xy_df <- data.frame(xy) %>%
  na.omit()

spdf <- SpatialPointsDataFrame(coords = xy_df, data = xy_df,
                               proj4string = CRS("+proj=utm +zone=17 +datum=WGS84 +units=m
+no_defs +ellps=WGS84 +towgs84=0,0,0 "))

locs_sf <- st_as_sf(spdf)

lon2 <- xFromCol(ndvi, unburn_cells$col)
lat2 <- yFromRow(ndvi, unburn_cells$row)
xy2 <- cbind(lon2, lat2)
xy_df2 <- data.frame(xy2) %>%
  na.omit()

spdf2 <- SpatialPointsDataFrame(coords = xy_df2, data = xy_df2,
                               proj4string = CRS("+proj=utm +zone=17 +datum=WGS84 +units=m
+no_defs +ellps=WGS84 +towgs84=0,0,0 "))

locs_sf2 <- st_as_sf(spdf2)

plot(ndvi)
sample_locs <- sample_n(locs_sf, size = 40)
plot(st_geometry(sample_locs), add =TRUE)

writeRaster(ndvi, "MTBS_created/rasters/LS5_for_figures.tif")
st_write(locs_sf, "MTBS_created/locations/sample_burn_locs.shp")
st_write(locs_sf2, "MTBS_created/locations/sample_unburn_locs.shp")
```

landsat_extraction_functions.R

```
# this script uses the location (pixel files) created in adding_pixels
# to loop through and extract values from the landsat files we have
# this loop also crops the landsat files to our base extent

setwd('/data/brooke_thesis')

burn_pixels <- read_csv('MTBS_created/locations/burn_nine_pixel_df.csv')
# locs will be cell numbers for burned cells
burn_locs <- burn_pixels$cell_id2
unburn_pixels <- read_csv('MTBS_created/locations/unburn_nine_pixel_df.csv')
unburn_locs <- unburn_pixels$cell_id2


setwd('/data/landsat/SC')
##list of all landsat files scene path 16 row 37
file.list = list.files('/data/landsat/SC/',
                       pattern = "(sr_band|bt_band|pixel_qa|radsat_qa)")

# retriving base extent
landsat_rasters <- lapply(file.list, raster)
landsat_extents <- lapply(landsat_rasters, extent)

ext2df <- function(x){
  return(data_frame(xmin = x@xmin,
                    xmax = x@xmax,
                    ymin = x@ymin,
                    ymax = x@ymax))
}
landsat_extents <- lapply(landsat_extents, ext2df)
bind_extents <- bind_rows(landsat_extents)
# extents of all raster files and file names
bind_extents$ID <- file.list

# creating base extent for the landsat
xmin <- max(bind_extents[,1])
xmax <- min(bind_extents[,2])
ymin <- max(bind_extents[,3])
ymax <- min(bind_extents[,4])
bbox <- c(xmin, xmax, ymin, ymax)
base_ext <- extent(bbox)
```

```
burn_Func <- function(file, locs, ext) {
  library(tidyverse)
  r = raster(file)
  cr <- crop(r, ext)
  df = raster::extract(cr, locs, df = TRUE)
  names(df) <- c('id','value')
  df <- df %>% mutate(file = file,
                      cell_id_og = burn_pixels$cell_id,
                      cell_id_unique = burn_pixels$cell_id2,
                      pixel_id = burn_pixels$pixel_id,
                      row = burn_pixels$row,
                      col = burn_pixels$col,
                      Fire_ID = burn_pixels$Fire_ID,
                      severity = burn_pixels$severity,
                      Fire_Year = burn_pixels$Year,
                      Fire_Date = burn_pixels$ig_date,
                      Fire_Acres = burn_pixels$Acres,
                      Fire_Type = burn_pixels$FireType,
                      forest = burn_pixels$forest,
                      nlcd2001 = burn_pixels$nlcd2001,
                      nlcd2006 = burn_pixels$nlcd2006,
                      nlcd2011 = burn_pixels$nlcd2011)
  return(df)
}

example1 <- burn_Func(file = file.list[[1]], locs=burn_locs, ext = base_ext)
# applying burn function to all Landsat files
burn_locs_df.list <- lapply(file.list, burn_Func, locs=burn_locs, ext=base_ext)
burn_locs_df <- bind_rows(burn_locs_df.list)
write_csv(burn_locs_df,
"/data/brooke_thesis/MTBS_created/locations/burn_locs_landsat_extract_raw.csv")

# similar function for unburned locations
unburn_Func <- function(file, locs, ext) {
  library(tidyverse)
  r = raster(file)
  cr <- crop(r, ext)
  df = raster::extract(cr, locs, df = TRUE)
  names(df) <- c('id','value')
  df <- df %>% mutate(file = file,
                      cell_id = unburn_pixels$cell_id2,
                      pixel_id = unburn_pixels$pixel_id,
                      row = unburn_pixels$row,
                      col = unburn_pixels$col,
```

```
                    Fire_ID = unburn_pixels$Fire_ID_1,
                    fire_id2 = unburn_pixels$fire_id,
                    Fire_Year = unburn_pixels$Year,
                    Fire_Date = unburn_pixels$ig_date,
                    Fire_Acres =unburn_pixels$Acres,
                    Fire_Type = unburn_pixels$FireType,
                    forest = unburn_pixels$forest)
  return(df)
}

example1 <- unburn_Func(file = file.list[[1]], locs=unburn_locs, ext = base_ext)
# applying burn function to all Landsat files
unburn_locs_df.list <- lapply(file.list, unburn_Func, locs=unburn_locs, ext=base_ext)
unburn_locs_df <- bind_rows(unburn_locs_df.list)

# this dataset includes only fire id's also included in the burn datasets
write_csv(unburn_locs_df,
"/data/brooke_thesis/MTBS_created/locations/unburn_locs_landsat_extract_raw_with_change.cs
v")
```

Tidy_landsat_dfs.R

```
# this script takes the outputs from landsat_extraction_functions
# and creates tidy data frames that include columns for quality assessment, date,
# sensor, path/row, and band information from the file name information

# also processes data for phenology assessments of spectral indices nbr ndvi and sr
library(tidyverse)
library(dplyr)
library(lubridate)

setwd("/data/brooke_thesis")

# raw Landsat extract data (very large, do not run except if necessary)
burn_locs_df <- read_csv("MTBS_created/locations/burn_locs_landsat_extract_raw.csv")
unburn_locs_df <-
read_csv("MTBS_created/locations/unburn_locs_landsat_extract_raw_with_change.csv")

# pixels for burned and unburned samples
burn_pixels <- read_csv("MTBS_created/locations/burn_nine_pixel_df.csv")
unburn_pixels <- read_csv("MTBS_created/locations/unburn_nine_pixel_df.csv")

# joining the location dataframes to the pixel dataframes to get the original cell id for central
pixel
# which I need for grouping by patch in running the pcls model for phenology curve construction
burn_locs_df2 <- left_join(burn_locs_df, burn_pixels %>% dplyr::select(cell_id_og = cell_id,
```

```
cell_id2, pixel_id),
                        by = c('cell_id' = 'cell_id2', "pixel_id"="pixel_id"))

unburn_locs_df2 <- left_join(unburn_locs_df, unburn_pixels %>% dplyr::select(cell_id_og =
cell_id, cell_id2, pixel_id),
                          by = c('cell_id' = 'cell_id2', "pixel_id"="pixel_id"))

require(lubridate)
# yday() converts the string to julian day (1-366)

# burned locations
if(!file.exists("MTBS_created/locations/tidy_burn_locs_landsat_extract.csv")){
  burn_df <-  burn_locs_df2 %>%
    separate(file,
            into = c('sensor', 'junk1', 'path_row', 'date', 'proc_date', 'level', 'tier', 'type', 'band', 'ext'))
%>%
    dplyr::select(-junk1, -level, -tier, -proc_date, -ext) %>%
    # date diff= negative = post-fire
    dplyr::mutate(date_diff =  as.Date(as.character(Fire_Date), format="%Y-%m-%d")-
                  as.Date(as.character(date), format="%Y%m%d")) %>%
    dplyr::mutate(LS_year = substring(date, 1, 4),
                LS_julian_day = yday(as.Date(substring(date, 5, 8), "%m%d")),
                fire_julian_day = yday(as.Date(substring(Fire_Date, 6,10), "%m-%d"))) %>%
    dplyr::select(-Fire_Year)
  write_csv(burn_df, "MTBS_created/locations/tidy_burn_locs_landsat_extract.csv")
}

burn_df <- read_csv("MTBS_created/locations/tidy_burn_locs_landsat_extract.csv") %>%
  na.omit()

# burned locations by sensor
# landsat 8
if(!file.exists("MTBS_created/locations/tidy_burn_locs_landsat8.csv")){
  LS8 <- burn_df %>%
  filter(sensor =='LC08') %>%
  unite("type_band", c("type","band")) %>%
  group_by(LS_year, LS_julian_day) %>%
  spread(key = type_band, value=value) %>%
  mutate(clean = ((pixel_qa %in% c(322, 386, 834, 898, 1346)) |
                  pixel_qa %in% c(324, 388, 836, 900, 1348)),
         ndvi = (sr_band5-sr_band4)/(sr_band5+sr_band4),
         nbr = (sr_band5-sr_band6)/(sr_band5 + sr_band6),
         sr = (sr_band5/sr_band4),
         red_fz = (sr_band4-mean(sr_band4, na.rm = TRUE))/sd(sr_band4, na.rm = TRUE),# forest
index for band 4
         swir1_fz = (sr_band6-mean(sr_band6, na.rm = TRUE))/sd(sr_band6, na.rm =TRUE), #
```

```
forest index for band 6
        swir2_fz = (sr_band7-mean(sr_band7, na.rm = TRUE))/sd(sr_band7, na.rm = TRUE), #
forest index for band 7
        ifz = sqrt((1/3)*((red_fz^2)+(swir1_fz^2)+(swir2_fz^2))), # integrated forest z-score
(forestness index)
        t = as.integer(as.Date(as.character(date), format= "%Y%m%d")- as.Date("2009-01-01")),
        #EVI = 2.5 * ((Band 5 – Band 4) / (Band 5 + 6 * Band 4 – 7.5 * Band 2 + 1)) also
multiplied each band by the scale factor (.0001)
        evi = 2.5 * ((sr_band5*0.0001 - sr_band4*0.0001) / (sr_band5*0.0001 + 6 *
sr_band4*0.0001 - 7.5 * sr_band2*0.0001 + 1)),
        # burn dummy 1, pre fire = 0, post-fire = 1
        burn_dummy1 = ifelse(date_diff < 0, 1, 0),
        # date dummy for changing time since fire, starting with 2 years post fire (-730)
        burn_dummy2 = ifelse(burn_dummy1 == 1 & date_diff <= -730, 1, 0)) %>%
  filter(clean == TRUE, ndvi > 0, ndvi < 1) %>%
  arrange(t)
  write_csv(LS8, "MTBS_created/locations/tidy_burn_locs_landsat8.csv")
  }

# landsat 7
if(!file.exists("MTBS_created/locations/tidy_burn_locs_landsat7.csv")){
  LS7 <- burn_df %>%
  filter(sensor == 'LE07') %>%
  unite("type_band", c("type", "band")) %>%
  group_by(LS_year, LS_julian_day) %>%
  spread(key = type_band, value=value) %>%
  mutate(clean = ((pixel_qa %in% c(66, 130)) |
                  pixel_qa %in% c(68, 132)),
        ndvi = (sr_band4-sr_band3)/(sr_band4+sr_band3),
        nbr = (sr_band4-sr_band5)/(sr_band4 + sr_band5),
        sr = (sr_band4/sr_band3),
        red_fz = (sr_band3-mean(sr_band3, na.rm = TRUE))/sd(sr_band3, na.rm = TRUE),# forest
index for band 3
        swir1_fz = (sr_band5-mean(sr_band5, na.rm = TRUE))/sd(sr_band5, na.rm =TRUE), #
forest index for band 5
        swir2_fz = (sr_band7-mean(sr_band7, na.rm =TRUE))/sd(sr_band7, na.rm = TRUE), #
forest index for band 7
        ifz = sqrt((1/3)*((red_fz^2)+(swir1_fz^2)+(swir2_fz^2))),
        t = as.integer(as.Date(as.character(date), format= "%Y%m%d")- as.Date("2009-01-01")),
        evi =  2.5 * ((sr_band4*0.0001 - sr_band3*0.0001) / (sr_band4*0.0001 + 6 *
sr_band3*0.0001 - 7.5 * sr_band1*0.0001 + 1)),
        # burn dummy 1, pre fire = 0, post-fire = 1
        burn_dummy1 = ifelse(date_diff < 0, 1, 0),
        # date dummy for changing time since fire, starting with 2 years post fire (-730)
        burn_dummy2 = ifelse(burn_dummy1 == 1 & date_diff <= -730, 1, 0)) %>%
  filter(clean == TRUE, ndvi > 0, ndvi < 1) %>%
```

```
    arrange(t)
  write_csv(LS7, "MTBS_created/locations/tidy_burn_locs_landsat7.csv")
}

# landsat 5
if(!file.exists("MTBS_created/locations/tidy_burn_locs_landsat5.csv")){
  LS5 <- burn_df %>%
  filter(sensor == 'LT05') %>%
  unite("type_band", c("type", "band")) %>%
  group_by(LS_year, LS_julian_day) %>%
  spread(key = type_band, value=value) %>%
  mutate(clean = ((pixel_qa %in% c(66, 130)) |
                  pixel_qa %in% c(68, 132)),
         ndvi = (sr_band4-sr_band3)/(sr_band4+sr_band3),
         nbr = (sr_band4-sr_band5)/(sr_band4 + sr_band5),
         sr = (sr_band4/sr_band3),
         red_fz = (sr_band3-mean(sr_band3, na.rm = TRUE))/sd(sr_band3, na.rm = TRUE),# forest
index for band 3
         swir1_fz = (sr_band5-mean(sr_band5, na.rm = TRUE))/sd(sr_band5, na.rm =TRUE), #
forest index for band 5
         swir2_fz = (sr_band7-mean(sr_band7, na.rm =TRUE))/sd(sr_band7, na.rm = TRUE), #
forest index for band 7
         ifz = sqrt((1/3)*((red_fz^2)+(swir1_fz^2)+(swir2_fz^2))),
         t = as.integer(as.Date(as.character(date), format= "%Y%m%d")- as.Date("2009-01-01")),
         #EVI = 2.5 * ((Band 4 – Band 3) / (Band 4 + 6 * Band 3 – 7.5 * Band 1 + 1))
         evi =  2.5 * ((sr_band4*0.0001 - sr_band3*0.0001) / (sr_band4*0.0001 + 6 *
sr_band3*0.0001 - 7.5 * sr_band1*0.0001 + 1)),
         # burn dummy 1, pre fire = 0, post-fire = 1
         burn_dummy1 = ifelse(date_diff < 0, 1, 0),
         # date dummy for changing time since fire, starting with 2 years post fire (-730)
         burn_dummy2 = ifelse(burn_dummy1 == 1 & date_diff <= -730, 1, 0)) %>%
  filter(clean == TRUE, ndvi > 0, ndvi < 1) %>%
  arrange(t)
  write_csv(LS5, "MTBS_created/locations/tidy_burn_locs_landsat5.csv")
}

# unburned locations
if(!file.exists("MTBS_created/locations/tidy_unburn_locs_landsat_extract.csv")){
  unburn_df <-  unburn_locs_df2 %>%
    separate(file,
             into = c('sensor', 'junk1', 'path_row', 'date', 'proc_date', 'level', 'tier', 'type', 'band', 'ext'))
%>%
    dplyr::select(-junk1, -level, -tier, -proc_date, -ext) %>%
    # date diff= negative = post-fire
    dplyr::mutate(date_diff =  as.Date(as.character(Fire_Date), format="%Y-%m-%d")-
                  as.Date(as.character(date), format="%Y%m%d")) %>%
```

```
    dplyr::mutate(LS_year = substring(date, 1, 4),
                LS_julian_day = yday(as.Date(substring(date, 5, 8), "%m%d")),
                fire_julian_day = yday(as.Date(substring(Fire_Date, 6,10), "%m-%d"))) %>%
    dplyr::select(-Fire_Year)

  write_csv(unburn_df, "MTBS_created/locations/tidy_unburn_locs_landsat_extract.csv")
}

#unburned locations
unburn_df <- read_csv("MTBS_created/locations/tidy_unburn_locs_landsat_extract.csv")

# landsat 8
if(!file.exists("MTBS_created/locations/tidy_unburn_locs_landsat8.csv")){
  LS8 <- unburn_df %>%
    filter(sensor =='LC08') %>%
    unite("type_band", c("type","band")) %>%
    group_by(LS_year, LS_julian_day) %>%
    spread(key = type_band, value=value) %>%
    mutate(clean = ((pixel_qa %in% c(322, 386, 834, 898, 1346)) |
                  pixel_qa %in% c(324, 388, 836, 900, 1348)),
          ndvi = (sr_band5-sr_band4)/(sr_band5+sr_band4),
          nbr = (sr_band5-sr_band6)/(sr_band5 + sr_band6),
          sr = (sr_band5/sr_band4),
          red_fz = (sr_band4-mean(sr_band4, na.rm = TRUE))/sd(sr_band4, na.rm = TRUE),#
forest index for band 4
          swir1_fz = (sr_band6-mean(sr_band6, na.rm = TRUE))/sd(sr_band6, na.rm =TRUE), #
forest index for band 6
          swir2_fz = (sr_band7-mean(sr_band7, na.rm =TRUE))/sd(sr_band7, na.rm = TRUE), #
forest index for band 7
          ifz = sqrt((1/3)*((red_fz^2)+(swir1_fz^2)+(swir2_fz^2))),
          t = as.integer(as.Date(as.character(date), format= "%Y%m%d")- as.Date("2009-01-01")),
          evi = 2.5 * ((sr_band5*0.0001 - sr_band4*0.0001) / (sr_band5*0.0001 + 6 *
sr_band4*0.0001 - 7.5 * sr_band2*0.0001 + 1)),
          # burn dummy 1, pre fire = 0, post-fire = 1
          burn_dummy1 = ifelse(date_diff < 0, 1, 0),
          # date dummy for changing time since fire, starting with 2 years post fire (-730)
          burn_dummy2 = ifelse(burn_dummy1 == 1 & date_diff <= -730, 1, 0)) %>%
  filter(clean == TRUE, ndvi > 0, ndvi < 1) %>%
  arrange(t)
  write_csv(LS8, "MTBS_created/locations/tidy_unburn_locs_landsat8.csv")
}

# landsat 7
if(!file.exists("MTBS_created/locations/tidy_unburn_locs_landsat7.csv")){
  LS7 <- unburn_df %>%
    filter(sensor == 'LE07') %>%
```

```
    unite("type_band", c("type", "band")) %>%
    group_by(LS_year, LS_julian_day) %>%
    spread(key = type_band, value=value) %>%
    mutate(clean = ((pixel_qa %in% c(66, 130)) |
                    pixel_qa %in% c(68, 132)),
           ndvi = (sr_band4-sr_band3)/(sr_band4+sr_band3),
           nbr = (sr_band4-sr_band5)/(sr_band4 + sr_band5),
           sr = (sr_band4/sr_band3),
           red_fz = (sr_band3-mean(sr_band3, na.rm = TRUE))/sd(sr_band3, na.rm = TRUE),#
forest index for band 3
           swir1_fz = (sr_band5-mean(sr_band5, na.rm = TRUE))/sd(sr_band5, na.rm =TRUE), #
forest index for band 5
           swir2_fz = (sr_band7-mean(sr_band7, na.rm =TRUE))/sd(sr_band7, na.rm = TRUE), #
forest index for band 7
           ifz = sqrt((1/3)*((red_fz^2)+(swir1_fz^2)+(swir2_fz^2))),
           t = as.integer(as.Date(as.character(date), format= "%Y%m%d")- as.Date("2009-01-01")),
           evi =  2.5 * ((sr_band4*0.0001 - sr_band3*0.0001) / (sr_band4*0.0001 + 6 *
sr_band3*0.0001 - 7.5 * sr_band1*0.0001 + 1)),
           # burn dummy 1, pre fire = 0, post-fire = 1
           burn_dummy1 = ifelse(date_diff < 0, 1, 0),
           # date dummy for changing time since fire, starting with 2 years post fire (-730)
           burn_dummy2 = ifelse(burn_dummy1 == 1 & date_diff <= -730, 1, 0)) %>%
    filter(clean == TRUE, ndvi > 0, ndvi < 1) %>%
    arrange(t)
  write_csv(LS7, "MTBS_created/locations/tidy_unburn_locs_landsat7.csv")
}

# landsat 5
if(!file.exists("MTBS_created/locations/tidy_unburn_locs_landsat5.csv")){
  LS5 <- unburn_df %>%
    filter(sensor == 'LT05') %>%
    unite("type_band", c("type", "band")) %>%
    group_by(LS_year, LS_julian_day) %>%
    spread(key = type_band, value=value) %>%
    mutate(clean = ((pixel_qa %in% c(66, 130)) |
                    pixel_qa %in% c(68, 132)),
           ndvi = (sr_band4-sr_band3)/(sr_band4+sr_band3),
           nbr = (sr_band4-sr_band5)/(sr_band4 + sr_band5),
           sr = (sr_band4/sr_band3),
           red_fz = (sr_band3-mean(sr_band3, na.rm = TRUE))/sd(sr_band3, na.rm = TRUE),#
forest index for band 3
           swir1_fz = (sr_band5-mean(sr_band5, na.rm = TRUE))/sd(sr_band5, na.rm =TRUE), #
forest index for band 5
           swir2_fz = (sr_band7-mean(sr_band7, na.rm =TRUE))/sd(sr_band7, na.rm = TRUE), #
forest index for band 7
           ifz = sqrt((1/3)*((red_fz^2)+(swir1_fz^2)+(swir2_fz^2))),
```

```
        t = as.integer(as.Date(as.character(date), format= "%Y%m%d")- as.Date("2009-01-01")),
        evi =  2.5 * ((sr_band4*0.0001 - sr_band3*0.0001) / (sr_band4*0.0001 + 6 *
sr_band3*0.0001 - 7.5 * sr_band1*0.0001 + 1)),
        # burn dummy 1, pre fire = 0, post-fire = 1
        burn_dummy1 = ifelse(date_diff < 0, 1, 0),
        # date dummy for changing time since fire, starting with 2 years post fire (-730)
        burn_dummy2 = ifelse(burn_dummy1 == 1 & date_diff <= -730, 1, 0)) %>%
    filter(clean == TRUE, ndvi > 0, ndvi < 1) %>%
    arrange(t)
  write_csv(LS5, "MTBS_created/locations/tidy_unburn_locs_landsat5.csv")
}

# all data combined and ready for phenology assessments
if(!file.exits("MTBS_created/locations/tidy_processed_locs_all_data.csv")){
  LS7_burned <- read_csv("MTBS_created/locations/tidy_burn_locs_landsat7.csv") %>%
    dplyr::select(sensor, cell_id, pixel_id, date, Fire_ID, forest, severity,
           Fire_Date, Fire_Acres, Fire_Type, ndvi, sr, nbr, burn_dummy1,
           burn_dummy2, t, date_diff, sr_band1, sr_band2, sr_band3, sr_band4,
           sr_band5, sr_band7, red_fz, swir1_fz, swir2_fz, ifz, evi)
  LS7_unburned <- read_csv("MTBS_created/locations/tidy_unburn_locs_landsat7.csv") %>%
    dplyr::select(sensor, cell_id, pixel_id, date, forest,
           Fire_Date, Fire_Acres, Fire_Type, ndvi, sr, nbr, burn_dummy1,
           burn_dummy2, t, date_diff, sr_band1, sr_band2, sr_band3, sr_band4,
           sr_band5, sr_band7, red_fz, swir1_fz, swir2_fz, ifz, evi)
  LS5_burned <- read_csv("MTBS_created/locations/tidy_burn_locs_landsat5.csv") %>%
    dplyr::select(sensor, cell_id, pixel_id, date, Fire_ID, forest, severity,
           Fire_Date, Fire_Acres, Fire_Type, ndvi, sr, nbr, burn_dummy1,
           burn_dummy2, t, date_diff, sr_band1, sr_band2, sr_band3, sr_band4,
           sr_band5, sr_band7, red_fz, swir1_fz, swir2_fz, ifz, evi)
  LS5_unburned <- read_csv("MTBS_created/locations/tidy_unburn_locs_landsat5.csv") %>%
    dplyr::select(sensor, cell_id, pixel_id, date, forest,
           Fire_Date, Fire_Acres, Fire_Type, ndvi, sr, nbr, burn_dummy1,
           burn_dummy2, t, date_diff, sr_band1, sr_band2, sr_band3, sr_band4,
           sr_band5, sr_band7, red_fz, swir1_fz, swir2_fz, ifz, evi)
  LS8_burned <- read_csv("MTBS_created/locations/tidy_burn_locs_landsat8.csv") %>%
    dplyr::select(sensor, cell_id, pixel_id, date, Fire_ID, forest, severity,
           Fire_Date, Fire_Acres, Fire_Type, ndvi, sr, nbr, burn_dummy1,
           burn_dummy2, t, date_diff, sr_band1, sr_band2, sr_band3, sr_band4,
           sr_band5, sr_band7, red_fz, swir1_fz, swir2_fz, ifz, evi)
  LS8_unburned <- read_csv("MTBS_created/locations/tidy_unburn_locs_landsat8.csv") %>%
    dplyr::select(sensor, cell_id, pixel_id, date, forest,
           Fire_Date, Fire_Acres, Fire_Type, ndvi, sr, nbr, burn_dummy1,
           burn_dummy2, t, date_diff, sr_band1, sr_band2, sr_band3, sr_band4,
           sr_band5, sr_band7, red_fz, swir1_fz, swir2_fz, ifz, evi)
  ## BURNED PROCESSING
  # some pixel cell id's appear multiple times in the data set, due to overlap during pixel
```

```
aggregation
  burn_dup_cells <- burn_pixels[duplicated(burn_pixels$cell_id2),]

  all_burn <- rbind(LS7_burned, LS5_burned, LS8_burned) %>%
    left_join(burn_pixels %>% dplyr::select("cell_id_og" = "cell_id", "pixel_id", "cell_id2" ),
              by = c("cell_id" = "cell_id2", "pixel_id" = "pixel_id")) %>%
    mutate(burn_status = "burned") %>%
    #removing duplicate cells
    filter(!cell_id %in% burn_dup_cells$cell_id2)

  ## UNBURNED PROCESSING
  unburn_dup_cells <- unburn_pixels[duplicated(unburn_pixels$cell_id2),] # duplicate cells
  # fire df needed to add real Fire ID's to unburned, not sure why these did not transfer
  fire_attributes <- read_csv("MTBS_created/locations/unburn_sample_fire_attributes.csv")
  all_unburn <- rbind(LS7_unburned, LS5_unburned, LS8_unburned) %>%
    left_join(unburn_pixels %>% dplyr::select("cell_id_og" = "cell_id", "pixel_id", "cell_id2" ),
              by = c("cell_id" = "cell_id2", "pixel_id" = "pixel_id")) %>%
    filter(!cell_id %in% unburn_dup_cells$cell_id2) %>%
    mutate(severity = 0,
           burn_status = "unburned") %>%
    left_join(fire_attributes %>% dplyr::select(Fire_ID, ig_date, FireType, Acres),
              by = c("Fire_Date" = "ig_date",
                     "Fire_Type" = "FireType",
                     "Fire_Acres" = "Acres"))

  # data frame of all burned and unburned sampels from
  all_data <- rbind(all_burn, all_unburn) %>%
    mutate(patch_id = group_indices(.,cell_id_og))  # patch id based on original cells ids
  write_csv(all_data, "MTBS_created/locations/tidy_all_data_ungrouped_phenology.csv")
  # grouping by patch id and other important variables
  grouped_data <- all_data %>%
    group_by(patch_id, t, date, Fire_ID, Fire_Type, Fire_Acres, Fire_Date, burn_dummy1,
burn_dummy2,
             date_diff, forest, burn_status, date_diff, severity, sensor) %>%
    summarise_at(c("ndvi", "ifz", "evi", "nbr"), mean, na.rm=TRUE) %>%
    ungroup()
  write_csv(grouped_data, "MTBS_created/locations/tidy_all_data_grouped_phenology.csv")
}
```

Phen_funs.R

```
# all phenology functions plus initial grouped data frame
# run at the beginning of all clean_models scripts

# function for creating sin, cos, and deriv matrices for xmat_fun
matrix_fun <- function(t, k, trig, deriv){
```

```
  if(trig == 'sin') {
    x <- sin(((2*pi*k)/365)*t)
  }
  if(trig == 'cos') {
    x <- cos(((2*pi*k)/365)*t)
  }
  if(deriv == TRUE) {
    x <- x * (((2*pi*k)/365)^2)
  }
  return(x)
}
# function for creating X variable matrix for any set of time (t)
xmat_fun <- function(t){
  xparts_s <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = FALSE))
  sin_mat <- do.call(cbind, xparts_s)
  xparts_c <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = FALSE))
  cos_mat <- do.call(cbind, xparts_c)
  Xmat <- cbind(rep(1, length(t)),
                t,
                t^2,
                sin_mat,
                cos_mat)
  Xmat
} # returns Xmat
# function for creating penalty matrix for any set of time (t)
penalty_fun <- function(t){
  xparts_s_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = TRUE))
  sin_deriv_mat <- do.call(cbind, xparts_s_deriv)
  xparts_c_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = TRUE))
  cos_deriv_mat <- do.call(cbind, xparts_c_deriv)
  Pen_mat <- cbind(rep(0, length(t)),
                   rep(0, length(t)),
                   rep(2, length(t)),
                   sin_deriv_mat,
                   cos_deriv_mat)
  S <- list(crossprod(Pen_mat))
  S
} # returns S
# generates two penalty matrices (one for pre fire and one for post fire, as list
# use for pcls (S)
pen_mat_fun <- function(t, burn_dummy){
  xparts_s_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = TRUE))
  sin_deriv_mat <- do.call(cbind, xparts_s_deriv)
  xparts_c_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = TRUE))
  cos_deriv_mat <- do.call(cbind, xparts_c_deriv)
  Pen_mat <- cbind(rep(0, length(t)),
```

```
                 rep(0, length(t)),
                 rep(2, length(t)),
                 sin_deriv_mat,
                 cos_deriv_mat)
  change_mat1 <- Pen_mat * burn_dummy
  S_mat1 <- kronecker(matrix(c(1,0), 1, 2), Pen_mat)
  S_mat2 <- kronecker(matrix(c(0,1), 1, 2), change_mat1)
  pen_change_mat <- list(crossprod(S_mat1), crossprod(S_mat2)) # 2 penalty matrices in this list
  pen_change_mat
}
# t = time, burn_dummy1 = separetes pre and post fire
pen_burn_fun <- function(t, burn_dummy1, burn_dummy2){
  # sin derivative from t time in data
  xparts_s_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = TRUE))
  sin_deriv_mat <- do.call(cbind, xparts_s_deriv)
  # cos derivative from t time in data
  xparts_c_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = TRUE))
  cos_deriv_mat <- do.call(cbind, xparts_c_deriv)
  Pen_mat <- cbind(rep(0, length(t)),
                 rep(0, length(t)),
                 rep(2, length(t)),
                 sin_deriv_mat,
                 cos_deriv_mat)
  change_mat1 <- Pen_mat * burn_dummy1
  change_mat2 <- Pen_mat * burn_dummy2
  S_mat1 <- kronecker(matrix(c(1,0,0), 1, 3), Pen_mat)
  S_mat2 <- kronecker(matrix(c(0,1,0), 1, 3), change_mat1)
  S_mat3 <- kronecker(matrix(c(0,0,1), 1, 3), change_mat2)
  pen_change_mat <- list(crossprod(S_mat1), crossprod(S_mat2), crossprod(S_mat3)) # 3
penalty matrices in this list
  pen_change_mat
}
xmat_mat_fun <- function(t, burn_dummy){
  xparts_s <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = FALSE))
  sin_mat <- do.call(cbind, xparts_s)
  xparts_c <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = FALSE))
  cos_mat <- do.call(cbind, xparts_c)
  Xmat <- cbind(rep(1, length(t)), # this will be removed for fire model
              t,
              t^2,
              sin_mat,
              cos_mat)
  xmat_change <- cbind(Xmat,(Xmat*burn_dummy))
  xmat_change
}# function for burned data
# t = time, burn_dummy1 = separetes pre and post fire, burn_dummy2 = separates
```

```
# recent burn from burn after a certain time period (specified in data filtering)
xmat_burn_fun <- function(t, burn_dummy1, burn_dummy2){
  xparts_s <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = FALSE))
  sin_mat <- do.call(cbind, xparts_s)
  xparts_c <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = FALSE))
  cos_mat <- do.call(cbind, xparts_c)
  Xmat <- cbind(rep(1, length(t)),
                t,
                t^2,
                sin_mat,
                cos_mat)
  xmat_change <- cbind(Xmat,(Xmat*burn_dummy1), (Xmat*burn_dummy2))
  xmat_change
}
# penalty matrix function that outputs a penalty matrix that is the sum
# of two penalty matrices multiplied by respective smoothing parameters
# for calculating standard deviation
pen_mat_sd <- function(t, burn_dummy, sp1, sp2){
  xparts_s_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = TRUE))
  sin_deriv_mat <- do.call(cbind, xparts_s_deriv)
  xparts_c_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = TRUE))
  cos_deriv_mat <- do.call(cbind, xparts_c_deriv)
  Pen_mat <- cbind(rep(0, length(t)),
                   rep(0, length(t)),
                   rep(2, length(t)),
                   sin_deriv_mat,
                   cos_deriv_mat)
  change_mat1 <- Pen_mat * burn_dummy
  S_mat1 <- kronecker(matrix(c(1,0), 1, 2), Pen_mat)
  S_mat2 <- kronecker(matrix(c(0,1), 1, 2), change_mat1)
  pen_change_mat <- list(crossprod(S_mat1), crossprod(S_mat2)) # 2 penalty matrices in this list
  pen_mats <- (pen_change_mat[[1]] * sp1) + (pen_change_mat[[2]] * sp2)
  pen_mats
}
# outputs penalty matrix for only time and one smoothing parameter (no fire difference)
pen_mat_sd2 <- function(t, sp1){
  xparts_s_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = TRUE))
  sin_deriv_mat <- do.call(cbind, xparts_s_deriv)
  xparts_c_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = TRUE))
  cos_deriv_mat <- do.call(cbind, xparts_c_deriv)
  Pen_mat <- cbind(rep(0, length(t)),
                   rep(0, length(t)),
                   rep(2, length(t)),
                   sin_deriv_mat,
                   cos_deriv_mat)
  S <- list(crossprod(Pen_mat))
```

```
  pen_mat <- (S[[1]] * sp1)
}
pen_mat_sd3 <- function(t, burn_dummy1, burn_dummy2, sp1, sp2, sp3){
  xparts_s_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'sin', deriv = TRUE))
  sin_deriv_mat <- do.call(cbind, xparts_s_deriv)
  xparts_c_deriv <- lapply(1:20, function(x) matrix_fun(t, x, 'cos', deriv = TRUE))
  cos_deriv_mat <- do.call(cbind, xparts_c_deriv)
  Pen_mat <- cbind(rep(0, length(t)),
                   rep(0, length(t)),
                   rep(2, length(t)),
                   sin_deriv_mat,
                   cos_deriv_mat)
  change_mat1 <- Pen_mat * burn_dummy1
  change_mat2 <- Pen_mat * burn_dummy2
  S_mat1 <- kronecker(matrix(c(1,0,0), 1, 3), Pen_mat)
  S_mat2 <- kronecker(matrix(c(0,1,0), 1, 3), change_mat1)
  S_mat3 <- kronecker(matrix(c(0,0,1), 1, 3), change_mat2)
  pen_change_mat <- list(crossprod(S_mat1), crossprod(S_mat2), crossprod(S_mat3)) # 3
penalty matrices in this list
  pen_mats <- (pen_change_mat[[1]] * sp1) + (pen_change_mat[[2]] * sp2)  +
(pen_change_mat[[3]] * sp3)
  pen_mats
}

# Data prepped for phenology assessments (tidy_landsat_dfs.R)
# data that has been aggregated by spectral indices
study_fires_df <- read_csv('MTBS_created/study_fire_stats.csv')

# Data prepped for phenology assessments (tidy_landsat_dfs.R)
# data that has been aggregated by spectral indices
grouped_data <- read_csv("MTBS_created/locations/tidy_all_data_grouped_phenology.csv")
%>%
  mutate(image_julian_day = yday(as.Date(substring(date, 5, 8), "%m%d")),
         LS_year = substring(date, 1, 4)) %>%
  filter(ndvi > 0, Fire_ID %in% study_fires_df$Fire_ID) %>%
  na.omit()

# oak gum cypress
oak_df <- grouped_data %>%
  filter(forest == 600, burn_status == 'burned')

# loblolly
lob_df <- grouped_data %>%
  filter(forest == 160, burn_status == 'burned')

# STANDARD DEVIATION AND STANDARD ERROR
```

```
# function for calculating standard deviation of predictions
pred_sd <- function(xmat, pred_xmat, pen_mat){
  half_var <- solve(t(xmat) %*% xmat + pen_mat, t(xmat))
  coef_var <- half_var %*% t(half_var) # variance of coefficients
  pred_var <- diag(pred_xmat %*% coef_var %*% t(pred_xmat)) # beta variance
  pred_sd = sqrt(pred_var)
}
# xmat, real ndvi, predicted ndvi
# output is the reidual standard error
pred_se <- function(xmat, ndvi, ndvi_pred, pred_sd){
  resid_se = sum((ndvi-ndvi_pred)^2)/(nrow(xmat)-ncol(xmat))
  pred_se = resid_se * pred_sd
}

# a function that takes a list of beta and ID's (in this case fire)
# and converts them to predicted NDVI based on pcls
# returns a data frame of pre-fire NDVI, post-fire NDVI, and change between the two
# only works with (pre and post fire data, no time steps)
beta_fun2 <- function(beta, xpred, t){
  beta2 <- beta
  beta2[44:86] = 0 # first half of coef preserved, pre-fire
  beta2[2:3] = 0 # removing time trend
  beta3 <- beta
  beta3[1:43] = 0 # second half of coef preserved, post-fire
  beta3[45:46] = 0 # removing time trend
  pre_pred <- as.list(xpred %*% beta2) # pre fire nbr curve
  change <- as.list(xpred %*% beta3) # change less than 2 years post fire
  data_frame(pre_pred = as.numeric(pre_pred), # change
             post_pred = as.numeric(pre_pred) + as.numeric(change),
             change = as.numeric(change),
             t = t)
}

# function for outputting curve predictions for pre fire and two time steps after fire
beta_fun3 <- function(beta, xpred, t){
  beta2 <- beta
  beta2[44:129] = 0 # first third of coef preserved, pre-fire
  beta2[2:3] = 0 # removing time trend
  beta3 <- beta
  beta3[1:43] = 0 # second third of coef preserved, less than 2 years after fire
  beta3[45:46] = 0 # removing time trend
  beta3[85:129] = 0
  beta4 <- beta
  beta4[1:86] = 0 # last third, 2+ years after fire
  beta4[88:89] = 0 # removing time trend
  pre_pred <- as.list(xpred %*% beta2) # pre fire NDVI curve
```

```
  change_less2 <- as.list(xpred %*% beta3) # change less than 2 years post fire
  change_more2 <- as.list(xpred %*% beta4) # change more than 2 years post fire
  data_frame(pre_pred = as.numeric(pre_pred),
        post_less2 = as.numeric(pre_pred) + as.numeric(change_less2),
        post_more2 = as.numeric(pre_pred) + as.numeric(change_more2),
        change_less2 = as.numeric(change_less2),
        change_more2 = as.numeric(change_more2),
        t = t)
}

# PLOTTING BASED ON ANNUAL PREDICTIONS
base_plot <- scale_x_continuous(name = "Month",
                breaks = c(1,91,182,274,335),
                labels = c('Jan','April',"July","Oct","Dec"),
                minor_breaks = c(31, 60, 121, 152,
                          213, 244, 305),
                limits=c(0,367), expand=c(0,0))

# setting up prediction matrix for annual predictions and plotting results
tpred <- seq(0:366) # annual
pred_mat1 <- xmat_fun(t = tpred)
pred_mat2 <- cbind(pred_mat1, pred_mat1) # this is for pre and post fire models
pred_mat3 <- cbind(pred_mat1, pred_mat1, pred_mat1)
```

species_difference_models_ndvi.R

```
# for figures in thesis
# species differences for all data

setwd('/data/brooke_thesis')
library(tidyverse)
library(purrr)
library(dplyr)
library(tidyr)
library(broom)
library(readr)
library(caret)
library(ggplot2)
library(dplyr)
library(magrittr)
library(mgcv)
library(dismo)
library(lubridate)

source('MTBS_scripts/clean_models/phen_funs.R')
```

```
# aggregated data
# pre-burn sites for the burned samples
pre_burn <- grouped_data %>%
  filter(burn_dummy1 == 0, burn_status == 'burned')

# converting forest column to character
pre_burn$forest <- as.character(as.numeric(pre_burn$forest))

# data exploration
raw_ndvi <- ggplot(data = pre_burn, aes(x = image_julian_day, y = ndvi)) +
  geom_point(alpha = 1/5, aes(color = forest)) +
  ylab("NDVI") + xlab("Day of Year") +
  ylim(0, 1) +
  labs(title = "Raw NDVI, 9-pixel patches, pre-fire") +
  base_plot +
  scale_color_manual(name="Forest Group",
                     labels = c("Loblolly-Shortleaf",
                                "Oak-Gum-Cypress"),
                     values = c("160"="dark green",
                                "600"="orange"))
ggsave('all_data_by_species_ndvi.png')


raw_nbr <- ggplot(data = pre_burn, aes(x = image_julian_day, y = nbr)) +
  geom_point(alpha = 1/5, aes(color = forest)) +
  ylab("NDVI") + xlab("Day of Year") +
  ylim(-.5, .8) +
  labs(title = "Raw NBR, 9-pixel patches") +
  base_plot +
  scale_color_manual(name="Forest Group",
                     labels = c("Loblolly-Shortleaf",
                                "Oak-Gum-Cypress"),
                     values = c("160"="dark green",
                                "600"="orange"))
ggsave('all_data_by_species_nbr.png')

# loblolly aggregated across patches
loblolly <- pre_burn %>%
  filter(forest == 160)

# oak aggregated across patches
oak <- pre_burn %>%
  filter(forest == 600)

# GLOBAL MODELS
```

```
#lob
lob_xmat <- xmat_fun(loblolly$image_julian_day)
lob_mod <- lm(loblolly$ndvi~lob_xmat-1)
lob_pen_mat <- penalty_fun(loblolly$image_julian_day)

# oak
oak_xmat <- xmat_fun(oak$image_julian_day)
oak_mod <- lm(oak$ndvi~oak_xmat-1)
oak_pen_mat <- penalty_fun(oak$image_julian_day)

# PCLS OBJECTS
# lob
lob.pcls <- list(y = loblolly$ndvi, # response variable
        w = rep(1, length(loblolly$image_julian_day)),
        X = lob_xmat,
        C = matrix(0,0,0),
        S = lob_pen_mat,
        off = c(0),
        p = coef(lob_mod),
        sp = c(10000),
        Ain = matrix(0,1,43),
        bin = 0)

# oak
oak.pcls <- list(y = oak$ndvi, # response variable
        w = rep(1, length(oak$image_julian_day)),
        X = oak_xmat,
        C = matrix(0,0,0),
        S = oak_pen_mat,
        off = c(0),
        p = coef(oak_mod),
        sp = c(10000),
        Ain = matrix(0,1,43),
        bin = 0)

# BETA and NDVI predictions
# lob
lob_beta <- pcls(lob.pcls)
lob_beta[2:3] = 0 # removing time trend
loblolly$ndvi_pred = lob_xmat %*% lob_beta

# oak
oak_beta <- pcls(oak.pcls)
oak_beta[2:3] = 0
oak$ndvi_pred = oak_xmat %*% oak_beta
```

```
# setting up prediction matrix for annual predictions and plotting results
tpred <- seq(0:366) # annual
pred_mat1 <- xmat_fun(t = tpred)

# LOBLOLLY
# penalty matrix constructed for  calculating standard deviation, loblolly
lob_sd_penalty <- pen_mat_sd2(loblolly$image_julian_day, 10000)

# calculating standard error and deviation
lob_pred_sd <- pred_sd(lob_xmat, pred_mat1, lob_sd_penalty)
names(lob_pred_sd) <- "sd"
lob_pred_se <- pred_se(lob_xmat, loblolly$ndvi, loblolly$ndvi_pred, lob_pred_sd)
names(lob_pred_se) <- "se"


# OAK
# penalty matrix constructed for  calculating standard deviation, loblolly
oak_sd_penalty <- pen_mat_sd2(oak$image_julian_day, 10000)

# calculating standard error and deviation
oak_pred_sd <- pred_sd(oak_xmat, pred_mat1, oak_sd_penalty)
names(oak_pred_sd) <- "sd"
oak_pred_se <- pred_se(oak_xmat, oak$ndvi, oak$ndvi_pred, oak_pred_sd)
names(oak_pred_se) <- "se"

# PLOTTING BASED ON ANNUAL PREDICTIONS
species_curves <- data_frame(t = tpred,
                        lob_smooth = as.vector(pred_mat1 %*% lob_beta),
                        oak_smooth = as.vector(pred_mat1 %*% oak_beta),
                        lob_se = as.vector(lob_pred_se),
                        oak_se = as.vector(oak_pred_se)) %>%
  gather(key = "Forest", value = "ndvi_pred",
         lob_smooth, oak_smooth) %>%
  mutate(upper = ifelse(Forest == 160, ndvi_pred + (2*lob_se),
                        ndvi_pred + (2*oak_se)),
         lower = ifelse(Forest == 160, ndvi_pred - (2*lob_se),
                        ndvi_pred - (2*oak_se))) %>%
  mutate(season = ifelse(t %in% 60:151, 'spring',
                         ifelse(t %in% 152:243, 'summer',
                                ifelse(t %in% 244:334, 'autumn', 'winter'))))

write_csv(species_curves, 'species_ndvi_stats.csv')

# PLOTTING
base_species_plot <- ggplot(data = species_curves, aes(x = t, y = ndvi_pred)) +
  geom_line(aes(linetype = Forest)) +
```

```
  geom_ribbon(aes(x = t, ymin = lower, ymax = upper, linetype = Forest), alpha = .5) +
  ylab("NDVI") + ylim(.45, .8) +
  base_plot +
  scale_linetype_discrete(name  ="Forest Group",
                        breaks=c("lob_smooth", "oak_smooth"),
                        labels=c("Loblolly-Shortleaf Pine", "Oak-Gum-Cypress"))

ggsave('smooth_species_comp.png')

#ggsave('too_smooth_species_comp.png')


unsmooth_curves <- data_frame(t = tpred,
                          lob_unsmooth = as.vector(pred_mat1 %*% coef(lob_mod)),
                          oak_unsmooth = as.vector(pred_mat1 %*% coef(oak_mod))) %>%
  gather(key = "Curve", value = "ndvi_pred",
        lob_unsmooth, oak_unsmooth)

unsmooth_species_plot <- ggplot(data = unsmooth_curves) +
  geom_line(aes(x = t, y = ndvi_pred, linetype = Curve)) +
  ylab("NDVI") +
  ylim(.45, .8) +
  base_plot +
  scale_linetype_discrete(name  ="Forest Group",
                        breaks=c("lob_unsmooth", "oak_unsmooth"),
                        labels=c("Loblolly-Shortleaf Pine", "Oak-Gum-Cypress"))
ggsave('unsmooth_species_comp.png')
```

lob_models_ndvi.R

```
# for figures in thesis
# pre vs post fire + time steps

setwd('/data/brooke_thesis')
library(tidyverse)
library(purrr)
library(dplyr)
library(tidyr)
library(broom)
library(readr)
library(caret)
library(ggplot2)
library(dplyr)
library(magrittr)
library(mgcv)
```

```
library(dismo)
library(lubridate)
library(gghighlight)

source('MTBS_scripts/clean_models/phen_funs.R')

# global models
# for loblolly
lob_xmat <- xmat_mat_fun(lob_df$image_julian_day, lob_df$burn_dummy1)
lob_mod <- lm(lob_df$ndvi~lob_xmat-1)
lob_pen_mat <- pen_mat_fun(lob_df$image_julian_day, lob_df$burn_dummy1)

# PCLS with one burn dummy for pre and post fire
lob2.pcls <- list(y = lob_df$ndvi, # response variable
                w = rep(1, length(lob_df$image_julian_day)),
                X = lob_xmat,
                C = matrix(0,0,0),
                S = lob_pen_mat,
                off = c(0,0),
                p = coef(lob_mod),
                sp = c(10000, 100000),
                Ain = matrix(0,1,86),
                bin = 0)

lob_beta2 <- pcls(lob2.pcls)

# ndvi predictions
lob_df$ndvi_pred = lob_xmat %*% lob_beta2

# LOBLOLLY
# penalty matrix constructed for  calculating standard deviation, loblolly
lob_sd_penalty <- pen_mat_sd(lob_df$t, lob_df$burn_dummy1,
                             10000, 100000)

## PREPARING DATA FOR PLOTTING

# calculating standard error and deviation
lob_pred_sd <- pred_sd(lob_xmat, pred_mat2, lob_sd_penalty)
names(lob_pred_sd) <- "sd"
lob_pred_se <- pred_se(lob_xmat, lob_df$ndvi, lob_df$ndvi_pred, lob_pred_sd)
names(lob_pred_se) <- "se"

# loblolly pre and post fire curves, annual predictions
lob_curves_data <- beta_fun2(lob_beta2, pred_mat2, tpred) %>%
  mutate(pred_se = as.vector(lob_pred_se))
```

```
write_csv(lob_curves_data,'phenology_curves/loblolly_ndvi_stats.csv')

# loblolly curves with standard error, lower, and upper bounds
lob_tidy <- lob_curves_data %>%
  gather(key = "Burn_stat", value = "ndvi_pred",
         pre_pred, post_pred, change) %>%
  mutate(lower = ndvi_pred - (2*pred_se),
         upper = ndvi_pred + (2*pred_se))


# PLOTTING BASED ON ANNUAL PREDICTIONS
lob_plot <- ggplot(data=lob_tidy, aes(x = t, y = ndvi_pred)) +
  geom_line(aes(linetype = Burn_stat)) +
  geom_ribbon(aes(ymin=lower, ymax=upper, linetype = Burn_stat), alpha =0.1) +
  ylab("NDVI") + xlab("Day of Year") +
  ylim(-.25, .9) +
  geom_hline(yintercept = 0) +
  base_plot +
  scale_linetype_manual(name ="Burn Status",
                        breaks = c("pre_pred", "post_pred", "change"),
                        labels=c("Pre-Fire", "Post-Fire", "Change"),
                        values = c("pre_pred" = "solid",
                                   "post_pred" = "dashed",
                                   "change" = "dotted"))

ggsave('lob_pre_post_fire_ndvi.png')


# global models
# for loblolly
# looking at two time steps after fire
lob_xmat2 <- xmat_burn_fun(lob_df$image_julian_day, lob_df$burn_dummy1,
lob_df$burn_dummy2)
lob_mod2 <- lm(lob_df$ndvi~lob_xmat2-1)
lob_pen_mat2 <- pen_burn_fun(lob_df$image_julian_day, lob_df$burn_dummy1,
lob_df$burn_dummy2)

# PCLS with one burn dummy for pre and post fire
lob3.pcls <- list(y = lob_df$ndvi, # response variable
                  w = rep(1, length(lob_df$image_julian_day)),
                  X = lob_xmat2,
                  C = matrix(0,0,0),
                  S = lob_pen_mat2,
                  off = c(0,0, 0),
                  p = coef(lob_mod2),
                  sp = c(10000, 100000, 100000),
```

```
            Ain = matrix(0,1,129),
            bin = 0)

lob_beta3 <- pcls(lob3.pcls)


lob_df$ndvi_pred2 = lob_xmat2 %*% lob_beta3

# STANDARD DEVIATION AND STANDARD ERROR

# LOBLOLLY
# penalty matrix constructed for  calculating standard deviation, loblolly
lob_sd_penalty <- pen_mat_sd3(lob_df$image_julian_day, lob_df$burn_dummy1,
lob_df$burn_dummy2,
                    10000, 100000, 100000)

# calculating standard error and deviation
lob_pred_sd2 <- pred_sd(lob_xmat2, pred_mat3, lob_sd_penalty)
names(lob_pred_sd2) <- "sd"
lob_pred_se2 <- pred_se(lob_xmat2, lob_df$ndvi, lob_df$ndvi_pred2, lob_pred_sd2)
names(lob_pred_se2) <- "se"


## PREPARING DATA FOR PLOTTING
# loblolly pre and post fire curves
lob_curves_data2 <- beta_fun3(lob_beta3, pred_mat3, tpred) %>%
  mutate(pred_se = as.vector(lob_pred_se2))

write_csv(lob_curves_data2, 'phenology_curves/loblolly_ndvi_time_steps_stats.csv')

# loblolly curves with standard error, lower, and upper bounds
lob_tidy2 <- lob_curves_data2 %>%
  gather(key = "Burn_stat", value = "ndvi_pred",
         pre_pred, post_less2, post_more2, change_less2, change_more2) %>%
  mutate(lower = ndvi_pred - (2*pred_se),
         upper = ndvi_pred + (2*pred_se))

lob_plot2 <- ggplot(data=lob_tidy2, aes(x = t, y = ndvi_pred)) +
  geom_line(aes(linetype = Burn_stat)) +
  geom_ribbon(aes(ymin=lower, ymax=upper, linetype = Burn_stat), alpha =0.1) +
  ylab("NDVI") + xlab("Day of Year") +
  geom_hline(yintercept = 0) +
  ylim(-.25, .9) +
  base_plot +
  scale_linetype_manual(name ="Burn Status",
                   breaks = c("pre_pred", "post_less2", "post_more2",
```

```
                  "change_less2", "change_more2"),
            labels=c("Pre-Fire", "Less than 2 years post-fire",
                  "More than 2 years post-fire", "Change less than 2 years post-fire",
                  "Change more than 2 years post-fire"),
            values = c("pre_pred" = "solid",
                  "post_less2" = "dashed",
                  "post_more2" = "dotted",
                  "change_less2" = "dashed",
                  "change_more2" = "dotted"))
ggsave('lob_pre_post_fire_time_steps_ndvi.png')
```

phenometrics.R

```
setwd('/data/brooke_thesis/')
library(tidyverse)
library(purrr)
library(dplyr)
library(tidyr)
library(broom)
library(readr)
library(caret)
library(ggplot2)
library(dplyr)
library(magrittr)
library(mgcv)
library(dismo)
library(lubridate)
library(gghighlight)
library(data.table)
library(formattable)
library(DT)

# all curves
lob_oak <- read_csv("phenology_curves/species_ndvi_stats.csv")
lob_pre_post_ndvi <- read_csv('phenology_curves/loblolly_ndvi_stats.csv')
oak_pre_post_ndvi <- read_csv('phenology_curves/oak_ndvi_stats.csv')
lob_pre_post_nbr <- read_csv('phenology_curves/loblolly_nbr_stats.csv')
oak_pre_post_nbr <- read_csv('phenology_curves/oak_nbr_stats.csv')

# time steps
lob_time_ndvi <- read_csv('phenology_curves/loblolly_ndvi_time_steps_stats.csv')
oak_time_ndvi <- read_csv('phenology_curves/oak_ndvi_time_steps_stats.csv')
lob_time_nbr <- read_csv('phenology_curves/loblolly_nbr_time_steps_stats.csv')
oak_time_nbr <- read_csv('phenology_curves/oak_nbr_time_steps_stats.csv')
```

```
# unburned
lob_oak_ub <- read_csv("species_ndvi_stats_unburn.csv")
lob_change_ndvi_ub <- read_csv("phenology_curves/loblolly_ndvi_unburn_stats.csv")
oak_change_ndvi_ub <- read_csv("phenology_curves/oak_ndvi_stats_unburn.csv")
lob_change_nbr_ub <- read_csv("phenology_curves/loblolly_nbr_stats_unburn.csv")
oak_change_nbr_ub <- read_csv("phenology_curves/oak_nbr_stats_unburn.csv")

# returns phenology metrics for data frames that include "pred" variable
phen_metric_fun1 <- (function(df){
  phen_metrics <- df %>%
    summarize("Max" = max(pred),
              "Min" = min(pred),
              "Seasonal Amplitude" = max(pred) - min(pred),
              "Date of Max" = t[which.max(pred)],
              "Date of Min" = t[which.min(pred)],
              "Spring Onset" = t[max(which(t < 179 & pred < (min(pred) +
                                                      .5 * ((max(pred)) -
                                                           min(pred)))))],
              "Senescence" = t[min(which(t > 179 & pred < (min(pred) +
                                                           .5 * ((max(pred)) -
                                                                min(pred)))))])
  phen_metrics
})

lob <- lob_oak %>%
  dplyr::filter(Forest == 'lob_smooth') %>%
  dplyr::select(pred = ndvi_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Forest group" = "Loblolly-shortleaf pine")

oak <- lob_oak %>%
  dplyr::filter(Forest == 'oak_smooth') %>%
  dplyr::select(pred = ndvi_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Forest group" = "Oak-gum-cypress")

lob_pre_ndvi <- lob_pre_post_ndvi %>%
  dplyr::select(-post_pred, pred=pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Pre-fire")
oak_pre_ndvi <- oak_pre_post_ndvi %>%
  dplyr::select(-post_pred, pred=pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Pre-fire")
```

```
lob_post_ndvi <- lob_pre_post_ndvi %>%
  dplyr::select(-pre_pred, pred=post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Post-fire")
oak_post_ndvi <- oak_pre_post_ndvi %>%
  dplyr::select(-pre_pred, pred=post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Post-fire")
lob_pre_nbr <- lob_pre_post_nbr %>%
  dplyr::select(-post_pred, pred=pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Pre-fire")
lob_post_nbr <- lob_pre_post_nbr %>%
  dplyr::select(-pre_pred, pred = post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Post-fire")
oak_pre_nbr <- oak_pre_post_nbr %>%
  dplyr::select(-post_pred, pred = pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Pre-fire")
oak_post_nbr <- oak_pre_post_nbr %>%
  dplyr::select(-pre_pred, pred = post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Post-fire")

# table outputs
lob_comb <- rbind(lob_pre_ndvi, lob_post_ndvi,
                  lob_pre_nbr, lob_post_nbr)

formattable(lob_comb)
datatable(lob_comb)

oak_comb <- rbind(oak_pre_ndvi, oak_post_ndvi,
                  oak_pre_nbr, oak_post_nbr)

formattable(oak_comb)
datatable(oak_comb)


species_comb <- rbind(lob, oak)
```

```
formattable(species_comb)


# time steps
lob_pre_ndvi <- lob_time_ndvi %>%
  dplyr::select(-post_less2, -post_more2, pred = pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Pre-fire")
lob_less2_ndvi <- lob_time_ndvi %>%
  dplyr::select(-pre_pred, -post_more2, pred = post_less2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Less than 2 years after fire")
lob_more2_ndvi <- lob_time_ndvi %>%
  dplyr::select(-pre_pred, -post_less2, pred = post_more2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "More than 2 years after fire")

lob_pre_nbr <- lob_time_nbr %>%
  dplyr::select(-post_less2, -post_more2, pred = pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Pre-fire")
lob_less2_nbr <- lob_time_nbr %>%
  dplyr::select(-pre_pred, -post_more2, pred = post_less2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Less than 2 years after fire")
lob_more2_nbr <- lob_time_nbr %>%
  dplyr::select(-pre_pred, -post_less2, pred = post_more2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "More than 2 years after fire")

# table outputs
lob_comb_time <- rbind(lob_more2_ndvi, lob_less2_ndvi,lob_more2_nbr, lob_less2_nbr)

formattable(lob_comb_time)
datatable(lob_comb_time)

oak_pre_ndvi <- oak_time_ndvi %>%
  dplyr::select(-post_less2, -post_more2, pred = pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
```

```
        "Burn status" = "Pre-fire")
oak_less2_ndvi <- oak_time_ndvi %>%
  dplyr::select(-pre_pred, -post_more2, pred = post_less2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
        "Burn status" = "Less than 2 years after fire")
oak_more2_ndvi <- oak_time_ndvi %>%
  dplyr::select(-pre_pred, -post_less2, pred = post_more2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
        "Burn status" = "More than 2 years after fire")

oak_pre_nbr <- oak_time_nbr %>%
  dplyr::select(-post_less2, -post_more2, pred = pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
        "Burn status" = "Pre-fire")
oak_less2_nbr <- oak_time_nbr %>%
  dplyr::select(-pre_pred, -post_more2, pred = post_less2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
        "Burn status" = "Less than 2 years after fire")
oak_more2_nbr <- oak_time_nbr %>%
  dplyr::select(-pre_pred, -post_less2, pred = post_more2, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
        "Burn status" = "More than 2 years after fire")

# table outputs oak
oak_comb_time <- rbind(oak_more2_ndvi, oak_less2_ndvi,oak_more2_nbr, oak_less2_nbr)
oak_comb_time <- oak_comb_time[c(8,9,1,2,3,4,5,6,7)]
formattable(oak_comb_time)
datatable(oak_comb_time)



study_fires <- read_csv('MTBS_created/study_fire_stats.csv')

# UNBURN

lob_ub <- lob_oak_ub %>%
  dplyr::filter(Forest == 'lob_smooth') %>%
  dplyr::select(pred = ndvi_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Forest group" = "Loblolly-shortleaf pine")
```

```
oak_ub <- lob_oak_ub %>%
  dplyr::filter(Forest == 'oak_smooth') %>%
  dplyr::select(pred = ndvi_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Forest group" = "Oak-gum-cypress")

lob_pre_ndvi_ub <- lob_change_ndvi_ub %>%
  dplyr::select(-post_pred, pred=pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Pre-fire")
oak_pre_ndvi_ub <- oak_change_ndvi_ub %>%
  dplyr::select(-post_pred, pred=pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Pre-fire")
lob_post_ndvi_ub <- lob_change_ndvi_ub %>%
  dplyr::select(-pre_pred, pred=post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Post-fire")
oak_post_ndvi_ub <- oak_change_ndvi_ub %>%
  dplyr::select(-pre_pred, pred=post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NDVI",
         "Burn status" = "Post-fire")
lob_pre_nbr_ub <- lob_change_nbr_ub %>%
  dplyr::select(-post_pred, pred=pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Pre-fire")
lob_post_nbr_ub <- lob_change_nbr_ub %>%
  dplyr::select(-pre_pred, pred = post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Post-fire")
oak_pre_nbr_ub <- oak_change_nbr_ub %>%
  dplyr::select(-post_pred, pred = pre_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Pre-fire")
oak_post_nbr_ub <- oak_change_nbr_ub %>%
  dplyr::select(-pre_pred, pred = post_pred, t) %>%
  phen_metric_fun1() %>%
  mutate("Spectral index" = "NBR",
         "Burn status" = "Post-fire")
```

```
# table outputs
lob_comb_ub <- rbind(lob_pre_ndvi_ub, lob_post_ndvi_ub,
                     lob_pre_nbr_ub, lob_post_nbr_ub)

formattable(lob_comb_ub)
datatable(lob_comb_ub)

oak_comb_ub <- rbind(oak_pre_ndvi_ub, oak_post_ndvi_ub,
                     oak_pre_nbr_ub, oak_post_nbr_ub)

formattable(oak_comb_ub)
datatable(oak_comb_ub)


species_comb_ub <- rbind(lob_ub, oak_ub)
formattable(species_comb_ub)
```

Vita

Miranda Brooke Rose was born in Memphis, TN, to parents Todd and Teresa Rose. Brooke attended Henry County High School in Paris, TN. In 2016, Brooke earned a Bachelor of Science degree in Environmental Science, summa cum laude. As an undergraduate, Brooke used GIS to study hydropower development in China and land cover change in protected areas in Africa. Through a grant funded by the U.S. Forest Service, she focused on disturbance and recovery processes in southeastern forests as a master's student at the University of Tennessee. Brooke completed her requirements for the Master of Science degree in August 2019. She will start a PhD program in Plant Biology at the University of California, Riverside in Fall 2019.