# Abstract

The Landsat satellites provide decades of near-global surface reflectance measurements that are increasingly used to assess interannual changes in terrestrial ecosystem function. These assessments often rely on spectral indices related to vegetation greenness and productivity (e.g., NDVI). Nevertheless, multiple factors impede multi-decadal assessments of spectral indices using Landsat satellite data, including ease of data access and cleaning, as well as lingering issues with cross-sensor calibration and challenges with irregular timing of cloud-free acquisitions. To help address these problems, we developed the *LandsatTS* package for R. This software package facilitates sample-based time series analysis of surface reflectance and spectral indices derived from Landsat sensors. The package includes functions that enable the extraction of the full Landsat 5, 7, and 8 records from Collection 2 for point sample locations or small study regions using the Google Earth Engine accessed directly from R. Moreover, the package includes functions for (1) rigorous data cleaning, (2) cross-sensor calibration, (3) phenological modeling, and (4) time series analysis. For an example application, we show how *LandsatTS* can be used to assess changes in annual maximum vegetation greenness from 2000 to 2022 across the Noatak National Preserve in northern Alaska, USA. Overall, this software provides a suite of functions to enable broader use of Landsat satellite data for assessing and monitoring terrestrial ecosystem function during recent decades across local to global geographic extents.

# Background

Satellite remote sensing is crucial for assessing and monitoring how Earth’s terrestrial ecosystems have changed during recent decades (National Academies of Sciences 2018). The Landsat satellites are particularly valuable in this regard because they are the longest continuously running satellite program and were designed for terrestrial ecosystem monitoring at moderate spatial resolution (Wulder et al. 2019). The first Landsat satellite (Landsat 1) was launched in 1972 as a partnership between NASA and the US Geological Survey (USGS) and since that time a series of additional satellites have been launched, with the most recent being Landsat 9 in 2021. The Landsat satellites carry multispectral sensors that provide surface reflectance measurements used for a wide range scientific and land management applications (Wulder et al. 2019). These include global monitoring of forest canopy cover change (Hansen et al. 2013, Sexton et al. 2013), land cover and use change (Potapov et al. 2022) and surface water extent change (Pekel et al. 2016), as well as regional- to biome-scale assessments of how disturbance, land-use and climate change are impacting terrestrial ecosystems (e.g., Wulder et al. 2004, Powell et al. 2010, Ju and Masek 2016, Wang and Friedl 2019). Hence, the Landsat program has become a cornerstone of Earth surface monitoring. Yet there are challenges that hinder use of these data by ecologists, land managers, and other non-remote sensing specialists.

Here we present the *LandsatTS* (i.e., *Landsat Time Series*) software package for R that enables users to extract, process, and analyze time series of Landsat surface reflectance measurements for sample locations anywhere on Earth. *LandsatTS* enables extraction of Landsat 5, 7, and 8 surface reflectance measurements from the full Landsat Collection 2 dataset on Google Earth Engine (GEE; Gorelick et al. 2017). Furthermore, *LandsatTS* includes functions that facilitate (1) data cleaning, (2) cross-sensor calibration, (3) phenological modeling, and (4) time series analysis of vegetation greenness (Figure 1, Table 1). This software grew out of research projects focused on vegetation dynamics across northern high-latitude ecosystems (Berner et al. 2020, Berner and Goetz 2022) and is implemented within the free, open-source, and widely used R statistical computing environment (R Core Team 2021).

It has become easier to access and process Landsat data since the archive was made publicly available in 2008 (Wulder et al. 2012) and a copy of the archive subsequently hosted on GEE (Gorelick et al. 2017). The GEE cloud-computing platform enables users to access and process Landsat data using JavaScript and Python application program interfaces (APIs), as well as with R through the *rgee* package (Aybar et al. 2020). R is very popular among ecologists (Lai et al. 2019), yet other existing R packages only provide tools for processing individual Landsat scenes. For instance, *landsat* includes functions for radiometric and topographic correction of Landsat scenes (Goslee 2011), while *landsat8* includes functions for computing top of atmosphere reflectance, radiance, and/or brightness temperature on Landsat scenes (dos Santos 2017). Thus, the *rgee* package makes it easier for ecologists use the GEE platform and work with Landsat data. Nevertheless, it remains non-trivial to not only extract Landsat time series data using *rgee*, but also to thoroughly clean the extracted data to ensure that only high-quality measurements are used in analyses. *LandsatTS* therefore provides new tools for sample-based extraction of full Landsat data records using *rgee* to access the GEE. Furthermore, *LandsatTS* includes tools to rigorously clean Landsat data using both pixel-level CFmask flags (e.g., cloud, water; Zhu et al. 2015) and scene-level criteria (e.g., cloud cover, solar zenith angle). Consequently, *LandsatTS* helps further broaden the community of researchers who can utilize Landsat data for robust spatiotemporal analyses of terrestrial ecosystem dynamics.

Landsat time series analyses that use measurements from multiple sensors are hindered by systematic biases in spectral bands and indices among the Landsat 5 Thematic Mapper (TM), Landsat 7 Enhanced Thematic Mapper Plus (ETM+), and Landsat 8 Operational Land Imager (OLI) sensors (Ju and Masek 2016, Roy et al. 2016, Berner et al. 2020, Berner and Goetz 2022). If unaccounted for, these biases can introduce pronounced artificial trends into combined time series, such as spurious increases over time in spectral indices of vegetation greenness including the widely used Normalized Difference Vegetation Index (NDVI) (Sulla-Menashe et al. 2017). Prior approaches for cross-sensor calibration focused on linear corrections for individual spectral bands and select spectral indices (e.g., NDVI) using regional data (e.g., continental USA) from Landsat Collection 1 (e.g., Ju and Masek 2016, Roy et al. 2016). While valuable, these published cross-sensor calibration models do not account for potential non-linearities, may not be suitable for other regions, and may not be appropriate for the newer Landsat Collection 2 dataset. Therefore, *LandsatTS* includes functions to cross-calibrate spectral bands and indices among Landsat 5, 7, and 8 using either random forest machine learning or polynomial regression models. These models are fit using the user’s dataset. However, if the user’s dataset is too small to fit these models, then, if appropriate, the user can choose to fit models using pre-processed and staged Landsat data that were sampled from across the Arctic tundra and boreal forest biomes. Flexible implementation of cross-sensor calibration in the *LandsatTS* workflow enables the user to generate high quality time-series that are free from sensor-specific biases that can otherwise induce spurious trends.

Vegetation phenology controls ecosystem processes (e.g., photosynthesis) and is often assessed using spectral indices (e.g., NDVI) derived from satellite measurements (Helman 2018, Zeng et al. 2020). Nevertheless, efforts to assess vegetation phenology using the Landsat satellites are complicated by multiple factors that include (1) irregular timing of clear-sky acquisitions within a growing season and (2) changes in the annual number of clear-sky acquisitions across years as new satellites were launched. These challenges are especially acute in regions with short, cloudy growing seasons such as the Arctic, where the median number of clear-sky growing season measurements increased from 2 per year in 1995 to 7 per year in 2015 (Berner et al. 2020). Annual maximum vegetation greenness is an important metric of vegetation phenology related to productivity (Berner et al. 2020, Zeng et al. 2020, Boyd et al. 2021), yet this metric is sensitive to the timing and number of measurements made in a growing season. Consequently, simple calculations of this metric tend to be artificially low early in the Landsat record but less so during later years when more measurements are available, which can introduce a spurious positive trend into a time series (Berner et al. 2020). To address this issue, *LandsatTS* includes tools to estimate annual maximum vegetation greenness based on site-specific phenological modeling that iteratively fits cubic splines to vegetation greenness time series. Users interested in other aspects of vegetation phenology (e.g., timing of spring onset or fall senescence) could extract and process Landsat data using *LandsatTS*, but then capitalize on tools provided by other R packages, such as the new *phenofit* package that provides state-of-the-art tools for fitting phenological models (Kong et al. 2022). More broadly, while *LandsatTS* provides tools focused on generating high-quality vegetation greenness times series, it also enables users to undertake other analyses that rely on cleaned and cross-calibrated Landsat data.

*LandsatTS* includes an integrated suite of tools that were originally developed to assess long-term changes in vegetation greenness within the rapidly warming Arctic tundra and boreal forest biomes (Berner et al. 2020, Berner and Goetz 2022). This software implements a sample-based approach that we found is well-suited for assessing vegetation dynamics and evaluating ecological hypotheses in these cold northern biomes, while substantially reducing computational burden compared with wall-to-wall analyses. The sample-based approach is conducive to rigorous propagation of uncertainty using Monte Carlo simulations (Berner et al. 2020, Berner and Goetz 2022), which is important for improving confidence in remote sensing analyses but seldom carried out because of computational constraints (Myers-Smith et al. 2020). Furthermore, the sample-based approach has helped validate and interpret vegetation dynamics inferred from spectral indices by enabling comparisons between satellite and field measurements across widely distributed site networks (Boyd et al. 2019, Berner et al. 2020, Boyd et al. 2021, Walker et al. 2021). These tools have also been used to assess high-latitude vegetation responses to insect outbreaks (Boyd et al. 2019, Boyd et al. 2021), wildfires (Gaglioti et al. 2021), and permafrost degradation (Verdonen et al. 2020), as well as for syntheses focused on high-latitude disturbance regimes (Foster et al. 2022) and Arctic shrubification (Mekonnen et al. 2021). Among other applications, these tools could further be used to complement field-based ecosystems monitoring in protected areas, evaluate ecosystem impacts of extreme weather events (e.g., droughts), and improve local to global mapping efforts by enabling users to develop regression models for cross-sensor calibration. In summary, *LandsatTS* enables ecologists and other researchers to extract and process Landsat time series that can then be used to analyze vegetation phenology or for other user-defined applications. Below, we provide (1) an example application focused on vegetation dynamics across the Noatak National Preserve, USA, (2) instructions for package installation, and (3) descriptions of each function.

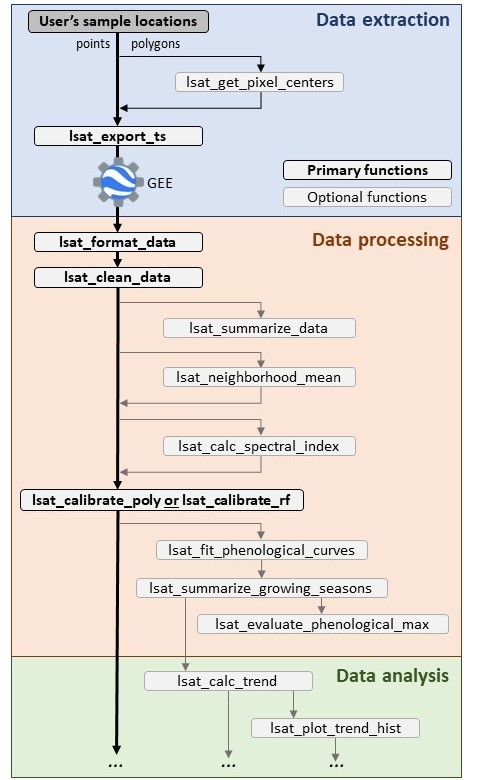


Figure 1. Schematic illustrating functions and typical workflow of the *LandsatTS* package. Each function is described in the main text and Table 1. *LandsatTS* has primary been used for assessments of interannual variability and trends in vegetation greenness. However, *LandsatTS* facilitates other Landsat time series analyses by providing tools for general data extraction and processing.

Table 1. Function names and descriptions. These are listed in the order typically used.

|  |  |  |
| --- | --- | --- |
| **Step** | **Function** | **Description** |
| Data extraction | lsat\_get\_pixel\_centers | (*Optional*) Retrieve point coordinates of all Landsat 8 pixel centers that fall within a polygon. |
|  | lsat\_export\_ts | Export full Landsat surface reflectance time series for a set of point coordinates using GEE accessed from R. |
| Data processing | lsat\_format\_data | Prepare data exported from GEE, including parsing satellite names and renaming and scaling bands. |
|  | lsat\_clean\_data | Filter out measurements based on presence of clouds, water, shadows, oblique view angles, and other criteria. |
|  | lsat\_summarize\_data | (*Optional*) Summarize data availability at each site, such as total number and years of observations. |
|  | lsat\_neighborhood\_mean | (*Optional*) For buffered sites, compute band-wise mean surface reflectance across grid cells within the buffer. |
|  | lsat\_calc\_spectral\_index | Calculate a variety of widely used spectral indices, such as the Normalized Difference Vegetation Index (NDVI). |
|  | lsat\_calibrate\_rf | Cross-calibrate bands or spectral indices from Landsat 5/8 to match Landsat 7 using Random Forest models. |
|  | lsat\_calibrate\_poly | Cross-calibrate bands or spectral indices from Landsat 5/8 to match Landsat 7 using polynomial regression. |
|  | lsat\_fit\_phenological\_curves | Characterize seasonal land surface phenology at each site by iteratively fitting flexible cubic splines. |
|  | lsat\_summarize\_growing\_seasons | Estimate various phenological metrics from fitted cubic splines, such as annual maximum vegetation greenness. |
|  | lsat\_evaluate\_phenological\_max | (*Optional*) Evaluate estimates of annual maximum vegetation greenness with measurement availability. |
| Data analysis | lsat\_calc\_trend | Calculate temporal trends using non-parametric Mann-Kendall trend tests and Theil-Sen slope indicators. |
|  | lsat\_plot\_trend\_hist | Plots a histogram of trends across sample sites |

# Example application: Vegetation greenness trends in the Noatak National Preserve, USA

Here we provide an example analysis of interannual changes in vegetation greenness from 2000 to 2022 within the Noatak National Preserve in northern Alaska, USA (Figure 2). The Noatak National Preserve is a vast wilderness of mountainous Arctic and alpine tundra that encompasses the largest undisturbed watershed in North America. The preserve is about 2.6 million hectares of roadless lands that were designated in 1980 to maintain ecological integrity, protect habitat and archeological resources, and provide opportunities for scientific research. Recent ecological research found climate warming substantially increased growth rates of white spruce (*Picea glauca*) and led to rapid expansion of trees and tall shrubs into tundra over the past half century in parts of the preserve (Suarez et al. 1999, Terskaia et al. 2020, Dial et al. 2022). The impacts of climate change are increasingly evident in the Noatak National Preserve and underscore the importance of sustained and cost-effective ecological monitoring and assessment.

Annual maximum vegetation greenness is related to tundra aboveground biomass and productivity, making it an important ecological metric that can be monitored using satellite remote sensing (Jia et al. 2003, Raynolds et al. 2012, Berner et al. 2018, Bhatt et al. 2021). We therefore demonstrate how multidecadal changes in annual maximum vegetation greenness can be readily assessed across the preserve using Landsat satellite data. In this section, we guide the reader through the analysis code with example output figures and tables that are generated by the *LandsatTS* functions.

## Part 1: Export Landsat time series from Google Earth Engine

To start, we create a random sample of points within the Noatak National Preserve and then

export Landsat time series for each sample point using GEE (Code Box 1). To facilitate our example, we include the preserve boundary as a simple feature polygon dataset (“noatak.sf”) in *LandsatTS*. Users could alternatively read in their own shapefile using *sf::st\_read()* or create a collection of spatial points (e.g., field sites) using *sf::st\_sf()* (Pebesma 2018). We load the preserve boundary dataset, create a simple random sample of *n* points within the boundary using the *sf::st\_sample* function, give each sample a unique identifier, and then create an interactive map showing preserve and sample point locations using *leaflet* (Figure 2) (Cheng et al. 2022). We then initialize GEE and submit a task to GEE that for each sample point exports all Landsat 5, 7, and 8 measurements made between day of year 152 (beginning of June) and 273 (end of September) from 1985 to 2022. For expediency, this example exports data for three random sample points, which took ~11 minutes and yielded ~800 B of data written to a folder called “earth\_engine” on the user’s Google Drive. Exporting four decades of summer Landsat data for 100 sample points took ~6 hours and yielded ~28 MB of data, while exporting data for 1,000 sample points took ~15 hours with four tasks running in parallel and yielded ~280 MB of data. To facilitate subsequent parts of this example, we include Landsat data for 100 sample points as a dataset (“noatak.dt”) in *LandsatTS*. Data export progress can be monitored using the GEE task manager in the web browser (https://code.earthengine.google.com/tasks) or with the R console using the ee\_monitoring() function provided by *rgee*. The CSV file(s) containing the raw exports need to be copied from the user’s Google Drive to the local machine that will carry out the subsequent processing using *LandsatTS*. The files can be copied manually or using the ee\_drive\_to\_local() function provided by *rgee*. Once the records are available locally, they need to be cleaned and processed into vegetation index time series as detailed in the next section.

Code Box 1: Export Landsat time series from Google Earth Engine

# Load required R packages

require(LandsatTS)

require(sf)

require(rgee)

require(tidyverse)

require(leaflet)

# Load the Noatak National Preserve simple feature polygon

data(noatak.sf)

# Create n random sample points within the Noatak National Preserve

n.pts <- 3

noatak.pts.sf <- st\_sample(x = noatak.sf, size = n.pts) %>% st\_sf()

# Add unique identifier to each point

noatak.pts.sf$sample\_id <- paste0('S\_', 1:n.pts)

# Make a basic interactive map showing Noatak National Preserve and sample points

leaflet() %>%

addProviderTiles('Esri.WorldImagery') %>%

addCircleMarkers(data = noatak.pts.sf,

color = 'white',

opacity = 0.9,

fillColor = 'fuchsia',

fillOpacity = 0.75,

weight = 1,

radius = 5) %>%

addPolygons(data = noatak.sf,

color = 'white',

weight = 3) %>%

addScaleBar(options = scaleBarOptions(imperial = F))

# Initialize Earth Engine

ee\_Initialize()

# Extract a time-series of surface reflectance measurements for each Landsat pixel

task\_list <- lsat\_export\_ts(pixel\_coords\_sf = noatak.pts.sf,

start\_date = "1985-06-01",

end\_date = "2022-09-30",

start\_doy = 152,

end\_doy = 273,

file\_prefix = 'noatak',

drive\_export\_dir = 'earth\_engine')

Map

Description automatically generated

Figure 2. Screenshot of a *leaflet* interactive map showing the Noatak National Preserve boundary in northern Alaska, USA, and 100 random sample points within the preserve. Landsat time series data were extracted for each of these sample points. Base map from ESRI World Imagery.

## Part 2: Format, clean, and summarize Landsat data in preparation for analysis

We load the Landsat data into R, format and clean the data, and then examine data availability. Here, we provide Landsat data for the 100 sample points as a dataset in *LandsatTS*; however, the dataset alternatively could be read into R as a data.table using the *fread()* function from the *data.table* package (Dowle and Srinivasan 2021). Once loaded into R, we format the dataset for analysis using *lsat\_format\_data()*, which formats column names and scales the band values, among other necessary formatting. We then clean the dataset using *lsat\_clean\_data()* to filter out clouds, snow, and water, as well as radiometric and geometric errors. For these field sites, *lsat\_clean\_data()* removed 78,625 of 99,600 observations (78.94%), including one sample point located in water. We then check the availability of clear-sky Landsat observations for the remaining 99 sample points using *lsat\_summarize\_data*(). On average (± 1 SD), each sample point had 212±48 clear-sky observations made between 1985 and 2022. The annual number of observations is typically small before the year 2000, as highlighted by the figure generated by the function (Figure 3).

Code Box 2: Format, clean, and summarize Landsat data in preparation for analysis

# Load required R packages

require(LandsatTS)

require(data.table)

require(tidyverse)

require(sf)

require(leaflet)

require(mapview)

# Load Landsat data for Noatak sites, or read in file using data.table::fread().

data(noatak.dt)

# Format the exported data

noatak.dt <- lsat\_format\_data(noatak.dt)

# Clean the data by filtering out clouds, snow, water, etc.

noatak.dt <- lsat\_clean\_data(noatak.dt)

# Summarize the availability of Landsat data for each pixel

lsat\_summarize\_data(noatak.dt)

# Continue to Code Box 3...

Chart

Description automatically generated

Figure 3. Annual availability of quality screened summer Landsat observations summarized across sample points in the Noatak National Preserve as returned by the function *lsat\_summarize\_data()*. Summaries are based on observations acquired between day of year 152 (beginning of June) and 273 (end of September). Note the limited availability of observations before the year 2000. Lines with points denote median counts while shaded bands encompass the 2.5th to 97.5th percentiles of counts among sample points.

*Part 3: Generate cross-calibrated time series of annual maximum vegetation greenness*

To generate time series of annual maximum vegetation greenness for each sample point, we (1) compute NDVI, (2) cross-calibrate NDVI among Landsat sensors, and then (3) estimate annual maximum NDVI (NDVImax) using phenological modeling. First, we calculate NDVI using *lsat\_calc\_spectral\_index()*, which supports calculating a variety of commonly used spectral indices. There are systematic differences in NDVI among Landsat sensors, so next we calibrate NDVI from Landsat 5 TM and Landsat 8 OLI to match Landsat 7 ETM+, which has measurements that temporally overlap with the other two sensors. We cross-calibrate NDVI among sensors using *lsat\_calibrate\_poly()* to fit and apply polynomial regression models. As the number of field sites in this dataset is rather small, we use a pre-processed dataset of Landsat observations that were randomly sampled from across northern high-latitudes ecosystems and are included for this purpose with *LandsatTS*. The function generates and returns a series of graphs (Figure 4) and tabular data (Table 3) that help with evaluating model performance and can optionally be written to a user-specified directory. As desired, calibration visually (Figure 4) and statistically (Table 3) reduced the bias between Landsat 7 NDVI and Landsat 5 and 8 NDVI.

As a step towards estimating annual NDVImax, we fit phenological models to the calibrated NDVI time series using *lsat\_fit\_phenological\_curves()*. The function automatically returns a figure with Landsat observations and fitted phenological curves for nine random sample locations in the dataset (Figure 5). Each phenological curve characterizes the seasonal progression of NDVI using observations pooled over a multi-year period (here a 7-year moving window) and should be smooth and hump-shaped. Beware of phenological curves with long straight lines that could suggest inadequate seasonal distribution of data used when fitting the curves. Phenological models were not fit for three sites that were minimally vegetated (NDVI < 0.15) because it is challenging to extract a meaningful vegetation phenology signal under these conditions. After fitting phenological models for 22 field sites, we then generated growing season summary statistics, including estimates of NDVImax, using *lsat\_summarize\_growing\_seasons()*. The *lsat\_evaluate\_phenological\_max()* can be used to output a figure that allows for visually assessing the performance of modelled NDVImax (Figure 6). In the case of the Noatak example dataset, modeled estimates of NDVImax tendto be biased slightly low (~1%) when only one or two observations are available from a growing season (Figure 6), yet there were rarely such few observations during the period from 2000 to 2021 (Figure 3). The final step following the cross-calibration and phenological modelling is the time series analysis.

Code Box 3: Cross-calibration and phenological modelling

# ... continuing from Code Box 2

# Compute the Normalized Difference Vegetation Index (NDVI)

noatak.dt <- lsat\_calc\_spectral\_index(noatak.dt, si = 'ndvi')

# Cross-calibrate NDVI among sensors using polynomial regression

noatak.dt <- lsat\_calibrate\_poly(noatak.dt,

band.or.si = 'ndvi',

train.with.highlat.data = T,

overwrite.col = T)

# Fit phenological models (cubic splines) to each time series

noatak.pheno.dt <- lsat\_fit\_phenological\_curves(noatak.dt, si = 'ndvi')

# Summarize growing season characteristics

noatak.gs.dt <- lsat\_summarize\_growing\_seasons(noatak.pheno.dt, si = 'ndvi')

# Evaluate estimates of annual maximum NDVI

noatak.gs.eval.dt <- lsat\_evaluate\_phenological\_max(noatak.pheno.dt, si = 'ndvi')

# Continue to Code Box 4...

Chart

Description automatically generated

Figure 4. Relationships between Landsat 7 NDVI and both (a) Landsat 5 NDVI and (b) Landsat 8 NDVI using (left panels) original data and (right panels) data that were calibrated using polynomial regression models. Each point is a sample location from the Arctic – Boreal domain with temporally overlapping measurements from pairs of Landsat sensors. Orange diagonal lines depict 1:1 relationships. Model performance metrics are provided in Table 3. Cross-calibration substantially reduces biases between sensors.

Table 3. Summary of original biases, performance of polynomial regression models for cross-sensor calibration, and post-calibration biases in NDVI between Landsat 7 ETM+ and either Landsat 5 TM or Landsat 8 OLI. Each model was trained using 75% of available data selected at random and then cross-validated using the remaining 25% of data.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Satellite**  **sensor** | **Number of sites** | |  | **Original Data** | | |  | **Cross-Validated Error Metrics** | | | |
| **Train** | **Eval.** |  | **RMSE** | **Median**  **bias** | **Median**  **% bias** |  | **r2** | **RMSE** | **Median bias** | **Median**  **% bias** |
| Landsat 5 TM | 5237 | 1746 |  | 0.052 | -0.04 | -6.1 |  | 0.974 | 0.032 | <0.01 | <0.1 |
| Landsat 8 OLI | 5927 | 1976 |  | 0.050 | 0.03 | 4.9 |  | 0.965 | 0.035 | <0.01 | <0.1 |

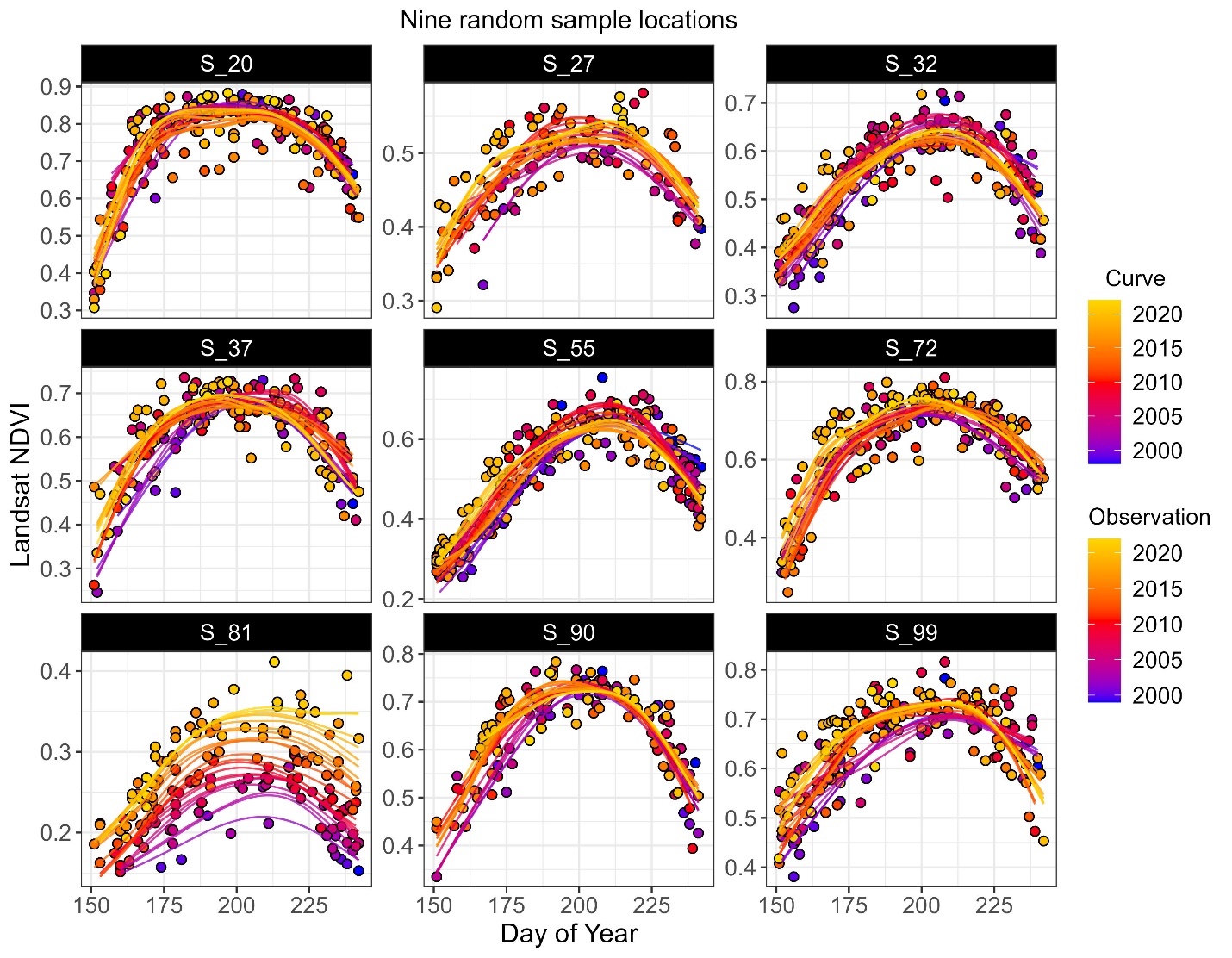


Figure 5. Seasonal progression of Landsat NDVI and phenological curves for nine random sample points in the Noatak National Preserve. Each dot is an observation that is colored by the year of acquisition ranging between 1985 and 2022. Each line represents a phenological curve that was fit to observations pooled over a 7-year window centered on the focal year as indicated by the color of the line. Color coding helps illustrate how individual curves are fit to observations. These figures can visually highlight long-term changes in phenology and canprovide a quick visual assessment of how well curves are being fit to observations, especially when the function is run using the parameter *test.run =* TRUE.

Chart, box and whisker chart

Description automatically generated

Figure 6. Raw estimates of annual maximum NDVI (NDVImax) are biased low when only a few Landsat observations are available from a given growing season, whereas phenologically modeled estimates of NDVImax are minimally impacted by the availability of observations. The figure summarizes how raw and modeled estimates of NDVImax differ from observed NDVImax based on number of observations, as determined using *lsat\_evaluate\_phenological\_max().*

## Part 3: Analyze vegetation greenness time series

Finally, we evaluate the interannual trend in NDVImax from 2000 to 2022 for each sample point. We calculate temporal trends using the *lsat\_calc\_trend()* function that implements and summarizes non-parametric trend assessments (Table 4). Note how we use the “yrs” argument to restrict the time series analysis to the years between 2000-2022 to avoid using the low number of observations in the record prior to the turn of the millennium. We then create a histogram of recent NDVImax trends using *lsat\_plot\_trend\_hist()* (Figure 7) and also create an interactive map showing the trend at each sample point (Figure 8). These figures indicate extensive greening across the study area in recent decades*.*

Code Box 4: Analyze and visualize vegetation greenness time series

# ... continuing from Code Box 3

# Compute temporal trend in annual NDVImax for each sample point

noatak.trend.dt <- lsat\_calc\_trend(noatak.gs.dt, si = 'ndvi.max', yrs = 2000:2022)

# Plot histogram of trends across sample points

lsat\_plot\_trend\_hist(noatak.trend.dt, xlim = c(-21,21))

# Create an interactive map showing NDVI trends

colors.dt <- data.table(trend.cat = c("greening","no\_trend","browning"),

trend.color = c("green","white","brown"))

noatak.trend.dt <- noatak.trend.dt[colors.dt, on = 'trend.cat']

noatak.trend.sf <- st\_as\_sf(noatak.trend.dt,

coords = c("longitude", "latitude"),

crs = 4326)

leaflet() %>%

addProviderTiles('Esri.WorldImagery') %>%

addPolylines(data = noatak.sf, color = 'white', weight = 3) %>%

addCircleMarkers(data = noatak.trend.sf,

color = 'white',

weight = 1,

opacity = 0.9,

fillColor = ~trend.color,

fillOpacity = 0.5,

radius = ~sqrt(abs(total.change.pcnt))\*3) %>%

setView(-160, 68, zoom = 7) %>%

addLegend('bottomright',

colors = colors.dt$trend.color,

labels = colors.dt$trend.cat,

title = 'NDVImax trend',

opacity = 1)

# End of code examples

Table 4. Abridged summary of NDVImax trends from 2000 to 2022 for each sample point (Sample ID) as generated using the function *lsat\_calc\_trend()*.Trends were assessed for each sample point by removing temporal autocorrelation and then applying a Mann-Kendall trend test (tau statistic and p-value provided). Slopes were calculated using the Theil-Sen slope estimators.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sample ID** | **Latitude** | **Longitude** | **N** | **Slope** | **Intercept** | **Tau** | **P-value** | **Total change** | **Total change (%)** |
| S\_1 | 67.70765 | -157.404 | 22 | 0.00109 | 0.5918 | 0.181 | 0.2639 | 0.025 | 4.2 |
| S\_10 | 68.23443 | -158.416 | 23 | 0.00127 | 0.6144 | 0.091 | 0.5728 | 0.029 | 4.7 |
| S\_11 | 67.8104 | -157.097 | 21 | 0.0017 | 0.6366 | 0.105 | 0.5376 | 0.039 | 6.1 |
| S\_12 | 67.81419 | -160.017 | 23 | 0.00155 | 0.6943 | 0.108 | 0.4986 | 0.036 | 5.2 |
| S\_13 | 68.12915 | -161.226 | 23 | 0.00209 | 0.5268 | 0.541 | < 0.001 | 0.048 | 9.1 |
| S\_14 | 68.26632 | -157.32 | 23 | 0.00067 | 0.2369 | 0.403 | 0.0095 | 0.015 | 6.3 |
| S\_15 | 67.87087 | -156.911 | 22 | 0.00073 | 0.6307 | 0.01 | 0.9759 | 0.017 | 2.7 |
| S\_16 | 68.18229 | -156.824 | 23 | 0.00048 | 0.6445 | 0.065 | 0.693 | 0.011 | 1.7 |
| S\_17 | 67.64494 | -158.002 | 23 | 0.00314 | 0.6726 | 0.541 | < 0.001 | 0.072 | 10.7 |
| S\_18 | 67.94227 | -161.809 | 23 | -0.00086 | 0.7419 | -0.152 | 0.3377 | -0.020 | -2.7 |
| S\_19 | 67.76848 | -162.447 | 23 | 0.00623 | 0.5918 | 0.784 | < 0.001 | 0.025 | 4.2 |

Chart, histogram

Description automatically generated

Figure 7. Histogram of relative change in Landsat NDVImax from 2000 to 2022 among sample points across the Noatak National Preserve. Relative changes in percent are calculated based on the Theil-Sen slope and intercept estimates (Table 4).

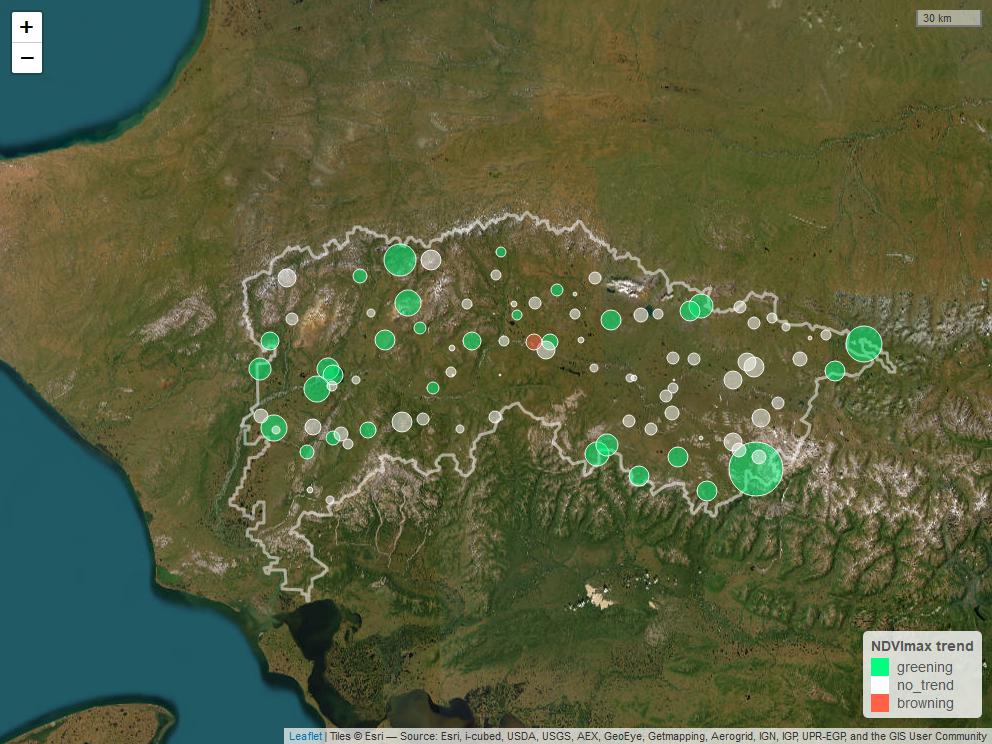


Figure 8. Screenshot of a *leaflet* interactive map showing the trends in NDVImax from 2000 to 2022 for sample points in the Noatak National Preserve. Base map from ESRI World Imagery.

*Results and interpretation of the example analysis*

Our analysis showed annual maximum vegetation greenness (i.e., NDVImax) increased 5.5 ± 10.8% (mean ± 1 SD) from 2000 to 2022 across sample points in the Noatak National Preserve (Figure 7). During these years, vegetation greenness increased by at least 10% at 20% of sample points. Vegetation greenness systematically (α = 0.10) increased at 32% of sample points, decreased at 1% of sample points, and exhibited no systematic change at the remaining 67% of sample points. Greening was especially prevalent in western parts of the preserve, as well as along in the northern foothills of the Brooks Range (Figure 8).

These remotely sensed changes suggest tundra productivity and biomass increased in recent decades across large parts of the Noatak National Preserve. These changes are consistent with observed warming-induced expansion of trees and tall shrubs in the preserve (Tape et al. 2006, Terskaia et al. 2020, Dial et al. 2022), as well as with rising summer temperatures increasing the productivity of existing vegetation in this cold tundra environment (Suarez et al. 1999, Berner et al. 2020, Dial et al. 2022). This preserve is also one of the most fire-prone regions in the Arctic and observed greening trends could partially be related to historical fires causing near-surface permafrost thaw, nutrient release, and subsequent shrub proliferation (Gaglioti et al. 2021). Greening in the preserve generally mirrors changes that have been observed more broadly across the Arctic tundra biome, though greening was more prevalent in the preserve than the broader Arctic (32% vs 27% of sample points) (Berner et al. 2020, Mekonnen et al. 2021).

This example analysis was based on Landsat data from 100 random sample points, yet nearly identical results were obtained when the analysis was performed using 1,000 sample points. Further insight into recent ecological changes could be garnered using a higher sample density with samples stratified by land cover type, ecological land unit, management unit, or other factors (e.g., Gaglioti et al. 2021, Berner and Goetz 2022). Nevertheless, Landsat data from even a relatively small random sample (n = 100) enabled robust inference about recent ecological changes that occurred over the past two decades within one of the most remote protected areas in the United States.

# Package installation

The R package *LandsatTS* is publicly available through a GitHub code repository. Users will need to have installed the R software environment on their computer. The *LandsatTS* package is operating system agnostic and can be installed from within R using the *install\_github()* function from the *devtools* package:

devtools::install\_github("logan-berner/LandsatTS")

The installation will compile the package from source code on the user’s computer. As the *LandsatTS* package itself is exclusively written in R code, no additional software is required.

To use the data extraction and preparation functions, users will need an account on GEE, and to have installed and configured the *rgee* package to access GEE from R. Please see the GEE (<https://earthengine.google.com/>) and *rgee* (<https://r-spatial.github.io/rgee/>) websites for details on signing up for an account and configuring *rgee*, respectively.

All other external package dependencies are configured and automatically dealt with by *devtools* during the installation. These required packages include (*LandsatTS* tested with version cited): *crayon* v1.4.2 (Csárdi 2021), *data.table* v1.14.2 (Dowle and Srinivasan 2021), *dplyr* v1.0.7 (Wickham et al. 2021), *ggplot2* v3.3.5 (Wickham 2016), *ggpubr* v0.4.0 (Kassambara 2020), *magrittr* v2.0.1 (Bache and Wickham 2020), *mapview* v2.10.0 (Appelhans et al. 2021), *purr* v0.3.4 (Henry and Wickham 2020), *R.utils* v2.11.0 (Bengtsson 2021), *ranger* v0.13.1 (Wright and Ziegler 2017), *sf* v1.0-4 (Pebesma 2018), *stats* v4.1.1 (R Core Team 2021), *stringr* v1.4.0 (Wickham 2019), *tidyr* v1.1.4 (Wickham 2021), *zoo* v1.8.9 (Zeileis and Grothendieck 2005), *zyp* v0.10-1.1 (Bronaugh and Werner 2019).

# Function descriptions

Below we provide a description of each function, with further details provided in the package manual that is available both within R and as Supplemental Material.

## Export point-coordinate Landsat time series from Google Earth Engine using lsat\_export\_ts()

The function *lsat\_export\_ts()* exports Landsat 5, 7 and 8 surface reflectance measurements for each sample location over a user-defined period by querying the Landsat Collection 2 archived on GEE. Sample locations typically represent (1) center coordinates of field sites, (2) a census of all Landsat pixels from a small area of interest, or (3) a random sample from a large region. If the user wishes to extract Landsat data for all pixels in a small area of interest (e.g., 5 km x 5 km), then the central coordinates of each pixel can be obtained using *lstat\_get\_pixel\_centers()* and then those sample locations are passed to *lsat\_export\_ts()*. It is important to stress this function only works for sample locations (point coordinates) that must be supplied as a simple feature (*sf*) collection of point geometries.

The function issues one or more tasks to GEE that export the data in the form of comma separated value (CSV) files to the user’s Google Drive. The number of tasks issued varies depending on the number of sample locations for which the Landsat record is to be extracted. Data extractions that involve many sample locations are prone to errors and may exceed user limits set by GEE. Therefore, the function will chunk the sample locations into small groups (by default 250 sites) and for each chunk will issue a separate export task to GEE. The function returns a list of *rgee* task objects, which can be used to query the progress of the exports and subsequently retrieve the data from the user’s Google Drive.

Please note that *lsat\_export\_ts()* has not been tested for data extractions exceeding 105 Landsat pixels (~90 km2). It took about two weeks to extract four decades of summer Landsat data for 105 pixels sampled from across the boreal forest biome. This data extraction yielded ~41.6 million multispectral measurements that required ~15 Gb of hard drive storage (Berner and Goetz 2022). *LandsatTS* enables large data extractions but is not infinitely scalable.

*Optional: Get central coordinates of pixels within a polygon using lsat\_get\_pixel\_centers()*

The function *lsat\_get\_pixel\_centers()* facilitates extracting data for all Landsat pixels in a small area of interest (e.g., < 5 km x 5 km) by determining the central coordinates of all Landsat pixels that fall within a user-specified polygon. The user-specified polygon is supplied to the function as a simple feature collection. The function determines the Landsat Worldwide Reference System (WRS) scene whose center is closest to the center of the user-specified polygon. It then extracts the center coordinates for all pixels that overlap with the user-specified polygon from the first Landsat 8 scene on record available on GEE. A buffer can be specified to include additional pixels beyond the polygon boundary. The function returns the pixel centers as a simple feature object that can then be passed to the *lsat\_export\_ts()* function for the extraction of the Landsat time series. Please note this function is not designed to be used for sampling polygons that would exceed tens of thousands of Landsat pixels. The number of pixels in large polygons can quickly become too difficult to handle in the subsequent export and processing workflow, and such polygons may also extend beyond the area of the Landsat scene (185 km x 180 km) used to determine the pixel centers. For large areas, we recommend a random or regular subsampling of point locations such as done in prior studies (Berner et al. 2020, Berner and Goetz 2022).

## Format data for analysis using lsat\_format\_data ()

The function *lsat\_format\_data()* takes the GEE exports generated by *lsat\_export\_ts()* and prepares the data for the subsequent *LandsatTS* workflow. These preprocessing tasks include parsing coordinates and other information, renaming columns, and scaling band values. The GEE exports need to be passed to the function in the form of a *data.table* object. *lsat\_format\_data()* returns a *data.table* object that can then be passed on to *lsat\_clean\_data()* for the next step in the processing workflow. Please note that all *LandsatTS* functions handling a *data.table* object require a column called “sample.id” that uniquely identifies each location. If this column is not called “sample.id”, please modify accordingly.

## Clean surface reflectance data using lsat\_clean\_data()

The function *lsat\_clean\_data()* filters measurements to those made under clear-sky conditions. This function allows the user to filter measurements based on pixel quality flags and scene criteria. The USGS provides pixel quality flags based on the CFMask algorithm (Zhu et al. 2015) and information on each scene (e.g., cloud cover). The default settings for *lsat\_clean\_data()* will filter out measurements flagged as snow or water, as well as measurements acquired at high solar zenith angle (>60°), those with high geolocation uncertainty (>15 m), or those acquired as part of scenes with extensive cloud cover (>80%). Additionally, optional water masking is provided based on maximum surface water extent from the Landsat-based JRC Global Surface Water Dataset (Pekel et al. 2016). The main input supplied to *lsat\_clean\_data()* is a *data.table* of Landsat records for individual sample locations (specified by a sample.id column) - usually the direct output of *lsat\_format\_data()* - and returns cleaned records in the form of an updated *data.table*, along with a console message summarizing the number and percentage of measurements removed during cleaning (generally >70%).

## Compute neighborhood mean surface reflectance using lsat\_neighborhood\_mean()

The function *lsat\_neighborhood\_mean()* computes the mean band-specific reflectance across a neighborhood of pixels for measurements at each period in time. This is helpful when each of the user’s sample locations was buffered to include a neighborhood of Landsat pixels (e.g., 3 x 3 pixels). If there are neighborhood pixels with no data (i.e., NA values), then the function omits those pixels and computes the mean across the remaining pixels. The main input to this function is a *data.table* of Landsat records for buffered sample locations. The function returns a new *data.table* with mean reflectance for each band at each point in time at every sample location. If used, the function should be called immediately after *lsat\_clean\_data()*.

## Summarize data availability for each site using lsat\_summarize\_data()

The function *lsat\_summarize\_data()* takes a *data.table* of Landsat records and returns a summary *data.table* that provides information on the time period and number of observations available for each sample location. It also generates a figure showing the annual median (2.5th and 97.5th percentile) number of observations available from each satellite summarized across all sample locations. The figure is plotted to the current graphics device and can be saved by calling the function *ggsave()*.

## Calculate spectral indices using lsat\_calc\_spectral\_index()

The function *lsat\_calc\_spectral\_index()* calculates a variety of common spectral indices. The function currently supports calculating 15 spectral indices, including the Normalized Difference Vegetation Index (NDVI), 2-band Enhanced Vegetation Index (EVI2), and others (Table 2). Note the function can only compute one spectral index at a time. As an input it requires a *data.table* with Landsat records and a string indicating the spectral index to be calculated. The function then returns the *data.table* updated with a new column containing the spectral index for each observation.

Table 2. Spectral indices that can be computed using the *lsat\_calc\_spectral\_index()* function.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Formula** | **Citation** |
| Enhanced Vegetation Index | EVI |  | Huete et al. (2002) |
| Enhanced Vegetation Index (2-band) | EVI2 |  | Jiang et al. (2008) |
| Moisture Stress Index | MSI |  | Rock et al. (1986) |
| Near Infrared Vegetation Index | NIRv |  | Badgley et al. (2017) |
| Normalized Burn Ratio | NBR |  | Key and Benson (1999) |
| Normalized Difference Infrared Index | NDII |  | Hardisky et al. (1983) |
| Normalized Difference Moisture Index | NDMI |  | Gao (1996) |
| Normalized Difference Vegetation Index (red) | NDVI |  | Rouse et al. (1974) |
| Normalized Difference Vegetation Index (green) | gNDVI |  | Gitelson and Merzlyak (1998) |
| Normalized Difference Vegetation Index (kernel) | kNDVI | )2) | Camps-Valls et al. (2021) |
| Normalized Difference Water Index | NDWI |  | McFeeters (1996) |
| Plant Senescence Reflectance Index | PSRI |  | Merzlyak et al. (1999) |
| Soil Adjusted Vegetation Index | SAVI | 1.5 \* | Huete (1988) |
| Soil Adjusted Total Vegetation Index | SATVI |  | Marsett et al. (2006) |
| Wide Dynamic Range Vegetation Index | WDRVI |  | Gitelson (2004) |

## Cross-calibrate spectral data across sensors using lsat\_calibrate\_rf()

The function *lsat\_calibrate\_rf()* will calibrate individual bands or spectral indices from Landsat 5 TM and Landsat 8 OLI to match Landsat 7 ETM+ using random forest models following the approach developed by Berner et al. (2020). Further cross-sensor calibration is needed because there are systematic differences in individual bands and spectral indices among Landsat sensors that must be addressed when combining data from multiple sensors (Ju and Masek 2016, Roy et al. 2016, Berner et al. 2020, Berner and Goetz 2022). Here, the Landsat 7 ETM+ is used as a benchmark because it temporally overlaps with the other two sensors. Cross-calibration can only be performed on one band or spectral index at a time and requires having data from 100s to preferably many 1,000s of sample locations to train the random forest models. There is an option for users to train the random forest models using pre-processed Landsat data from ~6000 randomly sampled locations across the Arctic – Boreal domain.

The overall approach involves determining the median spectral reflectance at a sample location during a portion of the growing season using Landsat 7 and Landsat 5/8 data that were collected the same years. A random forest model is then trained to predict Landsat 7 reflectance from Landsat 5/8 reflectance. Random forest models are ensembles of regression trees (Breiman 2001) that here are trained using a fast implementation provided by the *ranger* package (Wright and Ziegler 2017). If the user’s dataset includes both Landsat 5 and 8, then the function will train a random forest model for each sensor. The function evaluates model performance using both out-of-bag and cross-validated approaches. Please see Berner et al. (2020) for further details.

The main inputto *lsat\_calibrate\_rf()* is a *data.table* of Landsat records for sample locations and a string specifying the name of the band or spectral index to be cross-calibrated. By default, *lsat\_calibrate\_rf()* will return a *data.table* with a new column containing the cross-calibrated data. If requested using the *write.output* parameter, the function creates a user-specified output directory that contains (1) trained random forest models, (2) a CSV file with model evaluation metrics, and (3) a multi-panel figure comparing sensors pre- and post-calibration. In any case, model evaluation metrics are returned to the console and the figure plotted in the active graphics device. If the default setting to add a new column with the cross-calibrated data is used, then either use those data in the subsequent functions (e.g., ndvi.xcal) or, once satisfied, manually overwrite the uncalibrated data to simplify subsequent column names.

## Cross-calibrate spectral data across sensors using lsat\_calibrate\_poly()

The function *lsat\_calibrate\_poly()* behaves similarly to *lsat\_calibrate\_rf()* but fits polynomial regression models rather than random forest models. The function automatically fits first-, second- and third-order polynomial regression models (i.e., Y = β0 + β1X + β2X2 + β3X3). It then automatically selects the most parsimonious fit using Bayesian Information Criterion (BIC), applies the most parsimonious model for cross-sensor calibration, and then returns regression model coefficients and cross-validation metrics. Initial testing showed *lsat\_calibrate\_poly()* and *lsat\_calibrate\_rf()* produce very similar results (r2 = 0.97), have similar run times, and both effectively mitigate biases among Landsat sensors, yet an advantage of the more recently developed *lsat\_calibrate\_poly()* function is it generates regression model coefficients that can be more readily applied to other datasets or incorporated into other software (e.g., GEE).

## Fit phenological curves to vegetation greenness time series using lsat\_fit\_phenological\_curves()

The function *lsat\_fit\_phenological\_curves()* provides information on the phenological timing of every Landsat observation relative to multi-year estimates of annual maximum vegetation greenness at each sample location. Specifically, the function models seasonal land surface phenology at each sample location using cubic splines iteratively fit to vegetation greenness (e.g., NDVI) time series within successive moving windows. The magnitude and timing of annual maximum vegetation greenness are determined for each period by first pooling observations over years within each moving-window and then fitting cubic splines to observations that have been sorted by day of year. Often there are too few observations from an individual year to fit a reliable phenological curve, therefore the function enables users to pool observations over multiple years when fitting each cure. The default is a 7-year moving-window centered on the focal year, but the width of the moving window can be made shorter or longer if there are many or few observations in the data record. For each time period, a cubic spline is initially fit that describes vegetation greenness for each day of year during the growing season. To screen outliers, each observation of vegetation greenness is compared against the model fitted values for that day of year and if the deviation is greater than a user-specified difference (default is a 30% difference), then the observation is removed, and the cubic spline is re-fit. This is repeated until no observations exceed the user-specified threshold. The phenological status of each remaining observation is then determined relative to the modeled maximum vegetation greenness during the multi-year period. Additional details are provided in Berner et al. (2020).

The function takes as input a *data.table* with irregular time series of vegetation greenness observations at each sample location, as well as several parameters (e.g., moving window width, minimum number of observation needed to fit a cubic spline, cubic spline flexibility). The function returns a new *data.table* with phenological information for each remaining observation that occurred during a time period with adequate data for modeling surface phenology (i.e., typically fewer observations will be returned than are provided to the function). Among other output, the returned *data.table* provides for each observation the modeled estimates of (1) vegetation greenness for that day of year and for peak summer; (2) vegetation greenness for that day of year as a fraction of annual maximum vegetation greenness; (3) day of year when annual maximum vegetation greenness occurred; and (4) expected difference in vegetation greenness between that day of year and peak summer. The function also returns a figure to the current graphic device that shows seasonal progression of Landsat observations and modeled surface phenology for a random subset of nine sample locations. The user can optionally output a CSV that includes for each sample location the vegetation greenness predicted for each day of year during each period by the cubic splines. Furthermore, the function includes an optional “test run” mode that will run the function on a random subset of nine sample locations and return a figure showing model fits, thus allowing the user to quickly experiment with different parameter settings. Note the function was designed to characterize seasonal phenology in terrestrial ecosystems with a single growing season and thus may not be suitable for use in ecosystems with multiple growing seasons. Also, the function was designed for spectral indices that are typically positive (e.g., NDVI). If using a spectral index that is typically negative (e.g., NDWI) then multiply the index by -1 before running the *lsat\_fit\_phenological\_curves()* and *lsat\_summarize\_growing\_seasons()* functions and then back-transform afterwards.

## Derive annual growing season metrics using lsat\_summarize\_growing\_seasons()

The function *lsat\_summarize\_growing\_seasons()* estimates several annual growing season metrics from vegetation greenness time series and modeled land surface phenology derived from Landsat satellite observations. The function’s main input is the *data.table* generated by *lsat\_fit\_phenological\_curves()* and user-specified parameters including the name of the spectral index and the phenological cut-off for an observation to be considered part of the growing season. Specifically, an observation is considered to be part of the growing season if the modeled vegetation greenness for that day of year is within a user-specified fraction of modeled annual maximum vegetation greenness (by default 0.75). The function returns a new *data.table* that includes for each sample location the annual mean, median, and 90th percentile vegetation greenness computed from observations during each growing season. The function also returns phenologically modeled estimates of the magnitude and timing (day of year) of annual maximum vegetation greenness. For each sample location, annual maximum vegetation greenness is estimated by first adjusting individual observations by the expected difference in vegetation greenness between that day of year and peak summer, and then taking the median of phenologically adjusted values within each growing season. Please see Berner et al. (2020) for additional details.

## Assess estimates of maximum vegetation greenness using lsat\_evaluate\_phenological\_max()

The function *lsat\_evaluate\_phenological\_max()* assesses how estimates of annual maximum vegetation greenness vary with the number of Landsat observations when derived from raw observations and after phenological modeling. Raw estimates of annual maximum vegetation greenness are sensitive to the number of observations available from a growing season, but phenological modeling tends to substantially reduce this dependency (Berner et al. 2020). The main input to the function is a *data.table* with Landsat records and phenological information generated by *lsat\_fit\_phenological\_curves().* The function assumes the “actual” annual maximum vegetation greenness at a sample location is captured by having at least a user-specific number of observations (e.g., ≥ 7). For each site, the function extracts years with at least the user-specified number of growing season observations and then repeatedly compares how raw and phenologically modeled estimates of annual maximum vegetation greenness differ from actual annual maximum vegetation greenness as progressively smaller subsets of observations are used. The function returns a figure to the current graphic device that summarizes how raw and modeled estimates of annual maximum vegetation greenness differ from actual conditions when there are between 1 and n-1 Landsat observations from a single growing season. This lets the user determine how much annual estimates of maximum vegetation greenness are impacted by the number of available growing season observations.

## Compute interannual trends in vegetation greenness using lsat\_calc\_trend()

The function *lsat\_calc\_trend()* computes a temporal trend in annual time series of vegetation greenness for each sample location over a user-specified period. This function pre-whitens each time series (i.e., removes temporal autocorrelation) (Yue et al. 2002) and then computes Mann-Kendall trend tests and Theil-Sen slope indicators as implemented by the *zyp.yuepilon()* function from the *zyp* package (Bronaugh and Werner 2019). The function takes as input a *data.table* with annual time series of vegetation greenness, or other spectral index, for each sample location. The function returns (1) a new *data.table* that summarizes the interannual trend at each sample location; (2) a console message summarizing trends across all sample locations. Specifically, the new *data.table* summarizes for each sample location the trend slope, intercept, Kendall’s tau, and p-value, as well as total absolute and relative change in vegetation greenness and other information (e.g., number of years with observations). The console message summarizes the mean (±1 SD) relative change in vegetation greenness across all sample locations, as well as the percentage of samples sites that greened, browned, or had no trend based on a user-specified critical value (default α = 0.10).

## Plot histogram of vegetation greenness trends using lsat\_plot\_trend\_hist()

The function *lsat\_plot\_trend\_hist()* creates a histogram depicting the total percent change in vegetation greenness, or other spectral index, among sample locations. The function takes the *data.table* that is output by the function *lsat\_calc\_trend()* and returns a figure that is plotted to the current graphics device.

# Conclusions

The *LandsatTS* package for R facilitates extracting and processing Landsat surface reflectance time series, as well as generating and analyzing metrics of vegetation greenness and other spectral indices. We demonstrated the functionality of this software by analyzing multidecadal changes in vegetation greenness across the Noatak National Preserve, USA, but would like to highlight that these tools are also well suited for sample-based analyses of vegetation dynamics across geographic regions ranging from individual field sites to entire terrestrial biomes (e.g., Berner et al. 2020, Berner and Goetz 2022). To date, *LandsatTS* has been used for ecological studies focused on the Arctic tundra and boreal forest biomes, but many of the functions could be used for studies focused on lower latitude ecosystems, especially ecosystems without a multi-modal growing season. Overall, this software provides a suite of functions to enable broader use of Landsat satellite data for assessing and monitoring Earth’s land surface over the past four decades in a sample-based framework suitable for local to global geographic extents.

# References

Appelhans, T., F. Detsch, C. Reudenbach, and S. Woellauer. 2021. mapview: Interactive Viewing of Spatial Data in R. R package version 2.10.0. <https://CRAN.R-project.org/package=mapview>.

Aybar, C., Q. Wu, L. Bautista, R. Yali, and A. Barja. 2020. rgee: An R package for interacting with Google Earth Engine. Journal of Open Source Software **5**:2272.

Bache, S. M., and H. Wickham. 2020. magrittr: A Forward-Pipe Operator for R. R package version 2.0.1. <https://CRAN.R-project.org/package=magrittr>.

Badgley, G., C. B. Field, and J. A. Berry. 2017. Canopy near-infrared reflectance and terrestrial photosynthesis. Science Advances **3**:e1602244.

Bengtsson, H. 2021. R.utils: Various Programming Utilities. R package version 2.11.0. <https://CRAN.R-project.org/package=R.utils>.

Berner, L. T., and S. J. Goetz. 2022. Satellite observations document trends consistent with a boreal forest biome shift. Global Change Biology **28**:3275–3292.

Berner, L. T., P. Jantz, K. D. Tape, and S. J. Goetz. 2018. Tundra plant aboveground biomass and shrub dominance mapped across the North Slope of Alaska. Environmental Research Letters **13**:035002.

Berner, L. T., R. Massey, P. Jantz, B. C. Forbes, M. Macias-Fauria, I. H. Myers-Smith, T. Kumpula, G. Gauthier, L. Andreu-Hayles, B. Gaglioti, P. J. Burns, P. Zetterberg, R. D'Arrigo, and S. J. Goetz. 2020. Summer warming explains widespread but not uniform greening in the Arctic tundra biome. Nature communications **11**:4621.

Bhatt, U. S., D. A. Walker, M. K. Raynolds, J. E. Walsh, P. A. Bieniek, L. Cai, J. C. Comiso, H. E. Epstein, G. V. Frost, and R. Gersten. 2021. Climate drivers of Arctic tundra variability and change using an indicators framework. Environmental Research Letters **16**:055019.

Boyd, M. A., L. T. Berner, P. Doak, S. J. Goetz, B. M. Rogers, D. Wagner, X. J. Walker, and M. C. Mack. 2019. Impacts of climate and insect herbivory on productivity and physiology of trembling aspen (Populus tremuloides) in Alaskan boreal forests. Environmental Research Letters **14**:085010.

Boyd, M. A., L. T. Berner, A. C. Foster, S. J. Goetz, B. M. Rogers, X. J. Walker, and M. C. Mack. 2021. Historic declines in growth portend trembling aspen death during a contemporary leaf miner outbreak in Alaska. Ecosphere **12**:e03569.

Breiman, L. 2001. Random Forests. Machine Learning **45**:5-32.

Bronaugh, D., and A. Werner. 2019. zyp: Zhang + Yue-Pilon trends package. R package version 0.10-1.1. <https://CRAN.R-project.org/package=zyp>.

Camps-Valls, G., M. Campos-Taberner, Á. Moreno-Martínez, S. Walther, G. Duveiller, A. Cescatti, M. D. Mahecha, J. Muñoz-Marí, F. J. García-Haro, and L. Guanter. 2021. A unified vegetation index for quantifying the terrestrial biosphere. Science Advances **7**:eabc7447.

Cheng, J., B. Karambelkar, and Y. Xie. 2022. leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version 2.1.1. <https://CRAN.R-project.org/package=leaflet>.

Csárdi, G. 2021. crayon: Colored Terminal Output. R package version 1.4.2. <https://CRAN.R-project.org/package=crayon>.

Dial, R. J., C. T. Maher, R. E. Hewitt, and P. F. Sullivan. 2022. Sufficient conditions for rapid range expansion of a boreal conifer. Nature **608**:546-551.

dos Santos, A. 2017. landsat8: Landsat 8 Imagery Rescaled to Reflectance, Radiance and/or Temperature. R package version 0.1-10. <https://CRAN.R-project.org/package=landsat8>.

Dowle, M., and A. Srinivasan. 2021. data.table: Extension of `data.frame`. R package version 1.14.2. <https://CRAN.R-project.org/package=data.table>.

Foster, A. C., J. A. Wang, G. V. Frost, S. J. Davidson, E. Hoy, K. W. Turner, O. Sonnentag, H. Epstein, L. T. Berner, A. H. Armstrong, M. Kang, B. M. Rogers, E. Campbell, K. R. Miner, K. M. Orndahl, L. L. Bourgeau-Chavez, D. A. Lutz, N. French, D. Chen, J. Du, T. A. Shestakova, J. K. Shuman, K. Tape, A.-M. Virkkala, C. Potter, and S. Goetz. 2022. Disturbances in North American boreal forest and Arctic tundra: impacts, interactions, and responses. Environmental Research Letters **17**:113001.

Gaglioti, B., L. T. Berner, B. M. Jones, K. M. Orndahl, A. P. Williams, L. Andreu‐Hayles, R. D’Arrigo, S. J. Goetz, and D. H. Mann. 2021. Tussocks enduring or shrubs greening: Alternate responses to changing fire regimes in the Noatak River Valley, Alaska. Journal of Geophysical Research: Biogeosciences **126**:e2020JG006009.

Gao, B.-C. 1996. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. Remote Sensing of Environment **58**:257-266.

Gitelson, A. A. 2004. Wide dynamic range vegetation index for remote quantification of biophysical characteristics of vegetation. Journal of plant physiology **161**:165-173.

Gitelson, A. A., and M. N. Merzlyak. 1998. Remote sensing of chlorophyll concentration in higher plant leaves. Advances in Space Research **22**:689-692.

Gorelick, N., M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment **202**:18-27.

Goslee, S. 2011. Analyzing remote sensing data in R: The Landsat Package. The Journal of Statistial Software **43**.

Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. 2013. High-Resolution Global Maps of 21st-Century Forest Cover Change. science **342**:850.

Hardisky, M., V. Klemas, and M. Smart. 1983. The influence of soil salinity, growth form, and leaf moisture on the spectral radiance of Spartina alterniflora. Photogrammetric Engineering & Remote Sensing **49**:77-83.

Helman, D. 2018. Land surface phenology: What do we really ‘see’ from space? Science of the Total Environment **618**:665-673.

Henry, L., and H. Wickham. 2020. purrr: Functional Programming Tools. R package version 0.3.4. <https://CRAN.R-project.org/package=purrr>.

Huete, A., K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira. 2002. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. Remote Sensing of Environment **83**:195-213.

Huete, A. R. 1988. A soil-adjusted vegetation index (SAVI). Remote Sensing of Environment **25**:295-309.

Jia, G. J., H. E. Epstein, and D. A. Walker. 2003. Greening of arctic Alaska, 1981–2001. Geophysical Research Letters **30**:2067.

Jiang, Z., A. R. Huete, K. Didan, and T. Miura. 2008. Development of a two-band enhanced vegetation index without a blue band. Remote Sensing of Environment **112**:3833-3845.

Ju, J., and J. G. Masek. 2016. The vegetation greenness trend in Canada and US Alaska from 1984–2012 Landsat data. Remote Sensing of Environment **176**:1-16.

Kassambara, A. 2020. ggpubr: 'ggplot2' Based Publication Ready Plots. R package version 0.4.0. <https://CRAN.R-project.org/package=ggpubr>.

Key, C. H., and N. C. Benson. 1999. The Normalized Burn Ratio (NBR): A Landsat TM radiometric measure of burn severity. United States Geological Survey, Northern Rocky Mountain Science Center.(Bozeman, MT).

Kong, D., T. R. McVicar, M. Xiao, Y. Zhang, J. L. Peña-Arancibia, G. Filippa, Y. Xie, and X. Gu. 2022. phenofit: An R package for extracting vegetation phenology from time series remote sensing. Methods in Ecology and Evolution **13**:1508-1527.

Lai, J., C. J. Lortie, R. A. Muenchen, J. Yang, and K. Ma. 2019. Evaluating the popularity of R in ecology. Ecosphere **10**:e02567.

Marsett, R. C., J. Qi, P. Heilman, S. H. Biedenbender, M. C. Watson, S. Amer, M. Weltz, D. Goodrich, and R. Marsett. 2006. Remote sensing for grassland management in the arid southwest. Rangeland Ecology & Management **59**:530-540.

McFeeters, S. K. 1996. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. International Journal of Remote Sensing **17**:1425-1432.

Mekonnen, Z. A., W. J. Riley, L. T. Berner, N. J. Bouskill, M. S. Torn, G. Iwahana, A. L. Breen, I. H. Myers-Smith, M. G. Criado, Y. Liu, E. S. Euskirchen, S. J. Goetz, M. C. Mack, and R. F. Grant. 2021. Arctic tundra shrubification: a review of mechanisms and impacts on ecosystem carbon balance. Environmental Research Letters **16**:053001.

Merzlyak, M. N., A. A. Gitelson, O. B. Chivkunova, and V. Y. Rakitin. 1999. Non‐destructive optical detection of pigment changes during leaf senescence and fruit ripening. Physiologia plantarum **106**:135-141.

Myers-Smith, I. H., J. T. Kerby, G. K. Phoenix, J. W. Bjerke, H. E. Epstein, J. J. Assmann, C. John, L. Andreu-Hayles, S. Angers-Blondin, P. S. A. Beck, L. T. Berner, U. S. Bhatt, A. D. Bjorkman, D. Blok, A. Bryn, C. T. Christiansen, J. H. C. Cornelissen, A. M. Cunliffe, S. C. Elmendorf, B. C. Forbes, S. J. Goetz, R. D. Hollister, R. de Jong, M. M. Loranty, M. Macias-Fauria, K. Maseyk, S. Normand, J. Olofsson, T. C. Parker, F.-J. W. Parmentier, E. Post, G. Schaepman-Strub, F. Stordal, P. F. Sullivan, H. J. D. Thomas, H. Tømmervik, R. Treharne, C. E. Tweedie, D. A. Walker, M. Wilmking, and S. Wipf. 2020. Complexity revealed in the greening of the Arctic. Nature Climate Change **10**:106-117.

National Academies of Sciences. 2018. Thriving on Our Changing Planet: A Decadal Strategy for Earth Observation from Space. The National Academies Press, Washington, DC.

Pebesma, E. J. 2018. Simple features for R: standardized support for spatial vector data. The R Journal **10**:439-446.

Pekel, J.-F., A. Cottam, N. Gorelick, and A. S. Belward. 2016. High-resolution mapping of global surface water and its long-term changes. Nature **540**:418-422.

Potapov, P., M. C. Hansen, A. Pickens, A. Hernandez-Serna, A. Tyukavina, S. Turubanova, V. Zalles, X. Li, A. Khan, F. Stolle, N. Harris, X.-P. Song, A. Baggett, I. Kommareddy, and A. Kommareddy. 2022. The Global 2000-2020 Land Cover and Land Use Change Dataset Derived From the Landsat Archive: First Results. Frontiers in Remote Sensing **3**.

Powell, S. L., W. B. Cohen, S. P. Healey, R. E. Kennedy, G. G. Moisen, K. B. Pierce, and J. L. Ohmann. 2010. Quantification of live aboveground forest biomass dynamics with Landsat time-series and field inventory data: A comparison of empirical modeling approaches. Remote Sensing of Environment **114**:1053-1068.

R Core Team. 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Raynolds, M. K., D. A. Walker, H. E. Epstein, J. E. Pinzon, and C. J. Tucker. 2012. A new estimate of tundra-biome phytomass from trans-Arctic field data and AVHRR NDVI. Remote Sensing Letters **3**:403-411.

Rock, B., J. Vogelmann, D. Williams, A. Vogelmann, and T. Hoshizaki. 1986. Remote detection of forest damage. BioScience **36**:439-445.

Rouse, J., R. Haas, J. Schell, and D. Deering. 1974. Monitoring vegetation systems in the Great Plains with ERTS. NASA special publication **351**:309-317.

Roy, D. P., V. Kovalskyy, H. K. Zhang, E. F. Vermote, L. Yan, S. S. Kumar, and A. Egorov. 2016. Characterization of Landsat-7 to Landsat-8 reflective wavelength and normalized difference vegetation index continuity. Remote Sensing of Environment **185**:57-70.

Sexton, J. O., X.-P. Song, M. Feng, P. Noojipady, A. Anand, and C. Huang. 2013. Global, 30-m resolution continuous fields of tree cover: landsat-based rescaling of MODIS vegetation continuous fields with lidar-based estimates of error. Int J Digit Earth **6**.

Suarez, F., D. Binkley, M. W. Kaye, and R. Stottlemyer. 1999. Expansion of forest stands into tundra in the Noatak National Preserve, northwest Alaska. Ecoscience **6**:465-470.

Tape, K., M. Sturm, and C. Racine. 2006. The evidence for shrub expansion in Northern Alaska and the Pan-Arctic. Global Change Biology **12**:686-702.

Terskaia, A., R. J. Dial, and P. F. Sullivan. 2020. Pathways of tundra encroachment by trees and tall shrubs in the western Brooks Range of Alaska. Ecography **43**:769-778.

Verdonen, M., L. T. Berner, B. C. Forbes, and T. Kumpula. 2020. Periglacial vegetation dynamics in Arctic Russia: decadal analysis of tundra regeneration on landslides with time series satellite imagery. Environmental Research Letters **15**:105020.

Walker, X. J., H. D. Alexander, L. T. Berner, M. A. Boyd, M. M. Loranty, S. M. Natali, and M. C. Mack. 2021. Positive response of tree productivity to warming is reversed by increased tree density at the Arctic tundra-taiga ecotone. Canadian Journal of Forest Research **51**:1323-1338.

Wang, J. A., and M. A. Friedl. 2019. The role of land cover change in Arctic-Boreal greening and browning trends. Environmental Research Letters **14**:125007.

Wickham, H. 2016. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlang New York.

Wickham, H. 2019. stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. <https://CRAN.R-project.org/package=stringr>.

Wickham, H. 2021. tidyr: Tidy Messy Data. R package version 1.1.4. <https://CRAN.R-project.org/package=tidyr>.

Wickham, H., R. Francois, H. Lionel, and K. Müller. 2021. dplyr: A Grammar of Data Manipulation. R package version 1.0.7. <https://CRAN.R-project.org/package=dplyr>.

Wright, M. N., and A. Ziegler. 2017. Ranger: a fast implementation of random forests for high dimensional data in C++ and R. Journal of statistical software **77**:1-17.

Wulder, M., R. Skakun, W. Kurz, and J. White. 2004. Estimating time since forest harvest using segmented Landsat ETM+ imagery. Remote Sensing of Environment **93**:179-187.

Wulder, M. A., T. R. Loveland, D. P. Roy, C. J. Crawford, J. G. Masek, C. E. Woodcock, R. G. Allen, M. C. Anderson, A. S. Belward, W. B. Cohen, J. Dwyer, A. Erb, F. Gao, P. Griffiths, D. Helder, T. Hermosilla, J. D. Hipple, P. Hostert, M. J. Hughes, J. Huntington, D. M. Johnson, R. Kennedy, A. Kilic, Z. Li, L. Lymburner, J. McCorkel, N. Pahlevan, T. A. Scambos, C. Schaaf, J. R. Schott, Y. Sheng, J. Storey, E. Vermote, J. Vogelmann, J. C. White, R. H. Wynne, and Z. Zhu. 2019. Current status of Landsat program, science, and applications. Remote Sensing of Environment **225**:127-147.

Wulder, M. A., J. G. Masek, W. B. Cohen, T. R. Loveland, and C. E. Woodcock. 2012. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. Remote Sensing of Environment **122**:2-10.

Yue, S., P. Pilon, B. Phinney, and G. Cavadias. 2002. The influence of autocorrelation on the ability to detect trend in hydrological series. Hydrological processes **16**:1807-1829.

Zeileis, A., and G. Grothendieck. 2005. zoo: S3 Infrastructure for Regular and Irregular Time Series. Journal of statistical software **14**:1-27.

Zeng, L., B. D. Wardlow, D. Xiang, S. Hu, and D. Li. 2020. A review of vegetation phenological metrics extraction using time-series, multispectral satellite data. Remote Sensing of Environment **237**:111511.

Zhu, Z., S. Wang, and C. E. Woodcock. 2015. Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images. Remote Sensing of Environment **159**:269-277.