*Supplemental material for:*

**LandsatTS: an R package to facilitate retrieval, cleaning, cross-calibration, and phenological modeling of Landsat time-series data**

Logan T. Berner, Jakob J. Assmann, Signe Normand and Scott J. Goetz

# Function descriptions

Below we provide a description of each function in the *LandsatTS* package for R. Further details are 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 (Supplemental Table 1). 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.

Supplemental Table 1. 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 10,000 randomly sampled locations across the combined Arctic tundra and boreal forest biomes.

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 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). For example, we generated cross-calibration models for each spectral band and select spectral indices using *lsat\_calibrate\_poly()* with default settings and Landsat data sampled from 10,000 random locations across the combined Arctic tundra and boreal forest biomes (Supplemental Tables 2 and 3). Across bands and spectral indices, the regression models for adjusting Landsat 5 TM to match Landsat ETM+ had an average r2 of 0.97 ± 0.02 (± 1 SD) and reduced the average of the absolute median percent bias from 5.9% ± 5.5% to 0.2% ± 0.2%. Similarly, the regression models for adjusting Landsat 8 OLI to Landsat 7 ETM+ had an average r2 of 0.95 ± 0.03 and reduced the average of the absolute median percent bias from 4.4% ± 4.5% to 0.6% ± 0.5% (Supplemental Table 2). These cross-calibration models can be used in future research that focuses on monitoring and mapping forest and tundra ecosystems across northern high-latitude regions.

Supplemental Table 2. Summary of original biases, performance of polynomial regression models for cross-sensor calibration, and post-calibration biases relating spectral bands and indices from Landsat 5 TM and Landsat 8 OLI with Landsat 7 ETM+. Each model was trained and evaluated using Landsat data sampled from 10,000 random locations across the Arctic tundra and boreal forest biomes, with 75% of data used for training and the remaining 25% of data used for cross-validation.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Satellite sensor** | **Spectral**  **Band**  **or Index** | **Original Data** | | |  | **Cross-Validated Error Metrics** | | | |
| **RMSE** | **Median**  **bias** | **Median**  **% bias** |  | **r2** | **RMSE** | **Median**  **bias** | **Median**  **% bias** |
| Landsat 5 TM | BLUE | 0.01 | 0.01 | 13.3 |  | 0.94 | 0.01 | <0.01 | -0.2 |
|  | GREEN | 0.01 | 0.01 | 13.8 |  | 0.96 | 0.01 | <0.01 | -0.6 |
|  | RED | 0.01 | 0.01 | 14.8 |  | 0.97 | 0.01 | <0.01 | 0.0 |
|  | NIR | 0.02 | <0.01 | 0.8 |  | 0.94 | 0.01 | <0.01 | 0.2 |
|  | SWIR1 | 0.01 | <0.01 | 1.5 |  | 0.98 | 0.01 | <0.01 | 0.0 |
|  | SWIR2 | 0.01 | <0.01 | 1.5 |  | 0.98 | 0.01 | <0.01 | 0.2 |
|  | EVI2 | 0.03 | -0.01 | -4.3 |  | 0.97 | 0.02 | <0.01 | -0.2 |
|  | NBR | 0.03 | <0.01 | -1.1 |  | 0.98 | 0.03 | <0.01 | -0.4 |
|  | NDMI | 0.03 | -0.01 | -1.8 |  | 0.98 | 0.03 | <0.01 | -0.3 |
|  | NDVI | 0.05 | -0.04 | -5.6 |  | 0.98 | 0.03 | <0.01 | 0.2 |
|  | NDWI | 0.05 | 0.04 | -6.2 |  | 0.97 | 0.03 | <0.01 | 0.2 |
| Landsat 8 OLI | BLUE | 0.01 | -0.01 | -16.5 |  | 0.88 | 0.01 | <0.01 | 1.5 |
|  | GREEN | 0.01 | <0.01 | -2.6 |  | 0.91 | 0.01 | <0.01 | 0.6 |
|  | RED | 0.01 | <0.01 | -7.1 |  | 0.96 | 0.01 | <0.01 | 1.1 |
|  | NIR | 0.02 | 0.01 | 2.0 |  | 0.94 | 0.02 | <0.01 | 0.3 |
|  | SWIR1 | 0.01 | <0.01 | -2.6 |  | 0.97 | 0.01 | <0.01 | 0.0 |
|  | SWIR2 | 0.01 | <0.01 | 0.7 |  | 0.98 | 0.01 | <0.01 | 0.5 |
|  | EVI2 | 0.03 | 0.02 | 5.6 |  | 0.96 | 0.03 | <0.01 | -0.3 |
|  | NBR | 0.03 | <0.01 | 0.2 |  | 0.98 | 0.03 | <0.01 | -0.1 |
|  | NDMI | 0.03 | 0.02 | 4.3 |  | 0.97 | 0.03 | <0.01 | -1.1 |
|  | NDVI | 0.05 | 0.03 | 4.0 |  | 0.97 | 0.03 | <0.01 | -0.4 |
|  | NDWI | 0.04 | -0.02 | 3.0 |  | 0.95 | 0.04 | <0.01 | 0.2 |

Supplemental Table 3. Coefficients for polynomial regression models to cross-calibrate spectral bands and indices from Landsat 5 TM and Landsat 8 OLI with Landsat 7 ETM+. The regression models are first (Y = β0 + β1X), second (Y = β0 + β1X + β2X2), or third (Y = β0 + β1X + β2X2 + β3X3) order polynomials, where the order was selected using BIC. The models were trained and cross-validated using Landsat data randomly sampled from 10,000 locations across the combined Arctic tundra and boreal forest biomes.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Satellite sensor** | **Spectral**  **Band**  **or Index** | **Coefficients** | | | |  | **Standard Error of Coefficients** | | | |
| **β0** | **β1** | **β2** | **β3** |  | **β0** | **β1** | **β2** | **β3** |
| Landsat 5 TM | BLUE | -0.0007 | 0.8288 | 1.8833 | -6.0679 |  | 0.0005 | 0.0245 | 0.3161 | 1.0995 |
|  | GREEN | -0.0011 | 0.8733 | 0.3266 |  |  | 0.0004 | 0.0090 | 0.0392 |  |
|  | RED | -0.0068 | 1.0075 |  |  |  | 0.0002 | 0.0026 |  |  |
|  | NIR | 0.0057 | 0.9686 |  |  |  | 0.0009 | 0.0036 |  |  |
|  | SWIR1 | 0.0010 | 0.9791 |  |  |  | 0.0005 | 0.0025 |  |  |
|  | SWIR2 | 0.0025 | 0.9240 | 0.3858 | -0.5121 |  | 0.0006 | 0.0129 | 0.0778 | 0.1256 |
|  | EVI2 | 0.0022 | 1.0560 | -0.0553 |  |  | 0.0014 | 0.0092 | 0.0150 |  |
|  | NBR | 0.0013 | 1.0234 | -0.0363 |  |  | 0.0010 | 0.0049 | 0.0071 |  |
|  | NDMI | 0.0054 | 0.9993 |  |  |  | 0.0005 | 0.0022 |  |  |
|  | NDVI | 0.0232 | 0.8985 | 0.3852 | -0.2847 |  | 0.0036 | 0.0293 | 0.0678 | 0.0467 |
|  | NDWI | -0.0492 | 0.9052 | -0.2866 | -0.2563 |  | 0.0044 | 0.0360 | 0.0869 | 0.0636 |
| Landsat 8 OLI | BLUE | 0.0042 | 1.0950 | -0.4681 |  |  | 0.0003 | 0.0125 | 0.0973 |  |
|  | GREEN | -0.0005 | 1.0412 |  |  |  | 0.0003 | 0.0044 |  |  |
|  | RED | -0.0012 | 1.1592 | -0.9166 | 1.8795 |  | 0.0004 | 0.0155 | 0.1735 | 0.4975 |
|  | NIR | 0.0221 | 0.8442 | 0.1811 |  |  | 0.0018 | 0.0139 | 0.0270 |  |
|  | SWIR1 | 0.0071 | 0.9698 | 0.0685 |  |  | 0.0008 | 0.0082 | 0.0210 |  |
|  | SWIR2 | 0.0028 | 0.9212 | 0.5277 | -0.8244 |  | 0.0005 | 0.0113 | 0.0724 | 0.1250 |
|  | EVI2 | 0.0030 | 0.7845 | 0.6026 | -0.4981 |  | 0.0025 | 0.0254 | 0.0798 | 0.0751 |
|  | NBR | -0.0095 | 0.9744 | 0.1344 | -0.1085 |  | 0.0012 | 0.0051 | 0.0209 | 0.0229 |
|  | NDMI | -0.0196 | 1.0111 | 0.0923 | -0.2415 |  | 0.0007 | 0.0039 | 0.0140 | 0.0359 |
|  | NDVI | -0.0120 | 0.8323 | 0.2010 |  |  | 0.0034 | 0.0124 | 0.0110 |  |
|  | NDWI | -0.0037 | 0.8090 | -0.2425 |  |  | 0.0046 | 0.0181 | 0.0173 |  |

## 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.

# References

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

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., 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.

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.

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.

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.

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.

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.

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).

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.

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.

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.

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.

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.

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.

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.