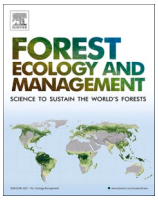
Forest Ecology and Management 503 (2022) 119813

Contents lists available at ScienceDirect

Forest Ecology and Management

journal homepage: www.elsevier.com/locate/foreco



Open-Source tools in R for forestry and forest ecology

Jeff W. Atkins [a,b,*], Atticus E.L. Stovall [c,d], Carlos Alberto Silva [e]

[a] USDA Forest Service Southern Research Station, New Ellenton, SC, USA
[b] Department of Biology, Virginia Commonwealth University, Richmond, VA 23284, USA
[c] Biospheric Sciences Laboratory, NASA Goddard Space Flight Center, Greenbelt, MD, USA
[d] Department of Geographical Sciences, University of Maryland, College Park, Maryland, MD 20740, USA
[e] Forest Biometrics and Remote Sensing Laboratory (Silva Lab), School of Forest, Fisheries and Geomatics Science, University of Florida, PO Box 110410 Gainesville, FL 32611, USA

ARTICLE INFO

Keywords:
R
Remote sensing
Mensuration
Software
Phenology
Inventory
Modeling
Statistics

ABSTRACT

Forestry and forest ecology research potentially lags behind related fields such as ecology, biodiversity, and conservation research in the employment of open-source software solutions, specifically the R programming language. A direct comparison of the last decade of published research literature from the top 20 ecology and forestry journals shows that R is utilized in over 30% of the literature for ecology, yet in less than 10% of the forestry literature. Open-source computing environments, such as R, Python, and Julia, increase the visibility and reproducibility of scientific research and foster collaborations through the removal of proprietary software restrictions. The lag in adoption of open-source software in forestry and forest ecology could be hindering collaboration, data sharing, and reproducibility. Here we survey the available packages in the R programming language with specific utility for forest-related research. We found more than 100 available packages which we systematically categorized by research category: community analysis; dendrochronology; forest mensuration and inventory; hydrology; informatics/IoT; modeling; phenology; and remote sensing. We present worked examples for a subgroup of R software packages for each category to demonstrate their potential and utility. In these examples we used open-source data sets of our own selection. Additionally, we collected this information into an R metapackage, *ForestAnalysisInR*, an R Shiny-based solution that allows users to query the R packages we have identified to find those best suited for their analysis needs in a quick and efficient way.

1. Introduction

Forestry and forest ecology have become increasingly data driven, advancing to the point where research relies heavily on software for analysis, modeling, visualization, and tabulation (Zou et al. 2019). Yet, much of the software in use among researchers and practitioners is proprietary in nature, which limits research reproducibility and restricts collaboration. Open-source software solutions offer a way to surmount these issues (Buck, 2015).

Reproducibility is a key feature of science (McNutt, 2014; Nosek et al. 2015) that can be limited by a reliance on proprietary software. Many proprietary software programs that are used among researchers in the natural sciences (e.g. PC-Ord, STATA, SAS, JMP etc.) are often prohibitively expensive for individual purchase, thus limiting access to those who are employed at more wealthy institutions that can afford licenses that can be shared among users. These limitations directly impact reproducibility and openness–as analyses performed on proprietary software cannot necessarily be shared or reproduced–as well as jeopardize the democratization of scientific research.

The global COVID-19 pandemic exposed yet another flaw in the sole reliance on proprietary software, as globally, many scientists and researchers lost in-person access to computing facilities and research software, or experienced significant disruptions to remote-access (Inouye et al. 2020; Carr et al. 2021). Further, the inequitable and variable availability of high-speed internet globally, created vast disparities in access to research capabilities, specifically remote proxies for any sponsored proprietary software requiring a university (or agency equivalent) VPN login. Open-source software environments such as R, Python, and Julia can surmount these limitations as they are free, do not require specific licenses or remote uplinks, and can be used across many platforms with shared compatibility. They also make it easier for researchers to reproduce analyses effectively and efficiently (Wilson et al.

* Corresponding author at: USDA Forest Service Southern Research Station, New Ellenton, SC, USA.
E-mail addresses: jeffrey.atkins@usda.gov (J.W. Atkins), atticus@umd.edu (A.E.L. Stovall), c.silva@ufl.edu (C. Alberto Silva).

https://doi.org/10.1016/j.foreco.2021.119813
Received 30 August 2021; Received in revised form 19 October 2021; Accepted 20 October 2021
Available online 28 October 2021
0378-1127/Published by Elsevier B.V.

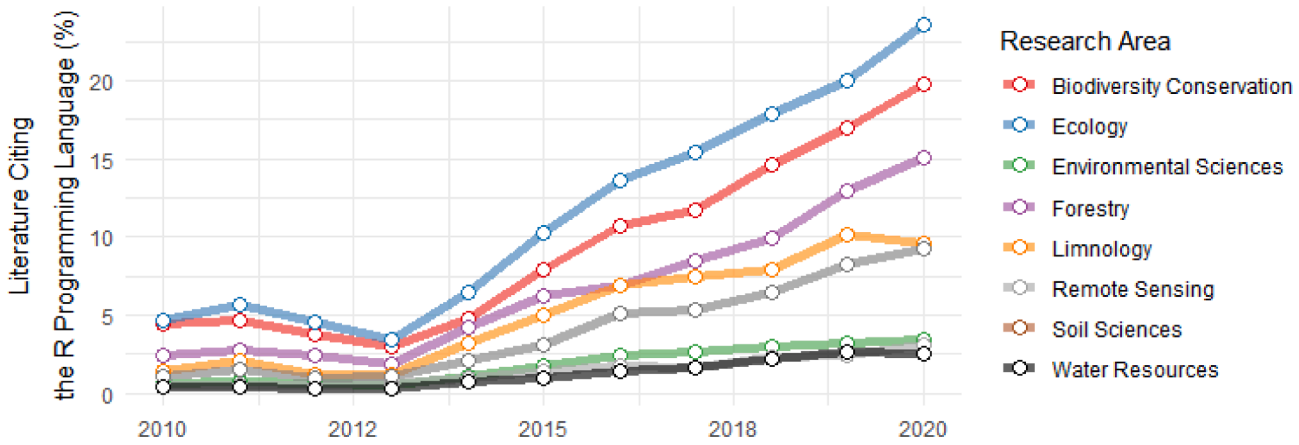


Fig. 1. Percentage of peer-reviewed articles indexed in Web of Science (WoS) published between 2010 and 2020 citing the R programming language. Total number of articles analyzed: 1,387,261; number of articles citing the R programming language: 62,902. WoS query was conducted using the cited author term "R Core Team", with 157 variations found among the literature. We restricted our analysis to document types "Articles," "Review Articles" and "Data Papers", under the assumption that citations of R within these document types most likey infers research usage. Full description of methodology in Supporting Information S1; Also see Data Availability.

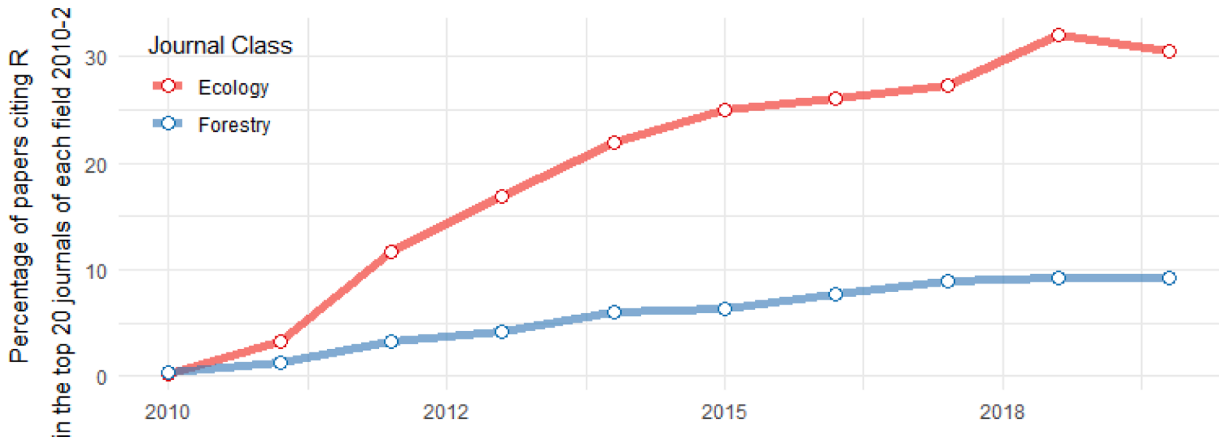


Fig. 2. Percentage of papers citing the R programming language by year from 2010 to 2019 within the top 20 journals of each field as ranked by Google Scholar metrics for 2020 (ordered by h5-index). See Supporting Information S1 for full methods.

2017).

R is used in many disciplines of the natural sciences (e.g., biology, ecology, forestry, botany), with its use steadily increasing over the preceding decade (Fig. 1). However, these increases in adoption have not been uniform among disciplines. In 2019, nearly 30% of the articles in the top 20 ecology journals cited usage of R, compared to only ~ 10% of articles in the top 20 forestry journals (Fig. 2.) –based on citation analysis from Web of Science (see Supporting Information S1 for methods and details). In part, this may be attributed to discipline-specific momentum or needs. R has broad utility, but is not always the optimal solution. The adoption of R in fields like hydrology (see Water Resources in Fig. 1) has been limited by the strong focus on and basis in engineering within the field resulting in a predominance in the use of MATLAB–a proprietary language utilized in both the public and private engineering sectors. Consequently, most students trained in hydrology are trained to use MATLAB. Many remote sensing scientists who work in open-source environments choose to work in Python because it has the added advantage of working relatively seamlessly with ArcGIS software (https://developers.arcgis.com/python/).

The observed differences in R usage within forest-related research may be attributable to many factors. The learning-curve that accompanies any computational skill can be significant and lack of access to educational resources or technical guidance to aid in skills acquisition can limit adoption. Online resources such as Data Carpentry (https://datacarpentry.org/) and NEON (https://www.neonscience.org/resources/learning-hub) exist to help fill this need specifically for research, while R-Studio offers additional online resources for broader audiences (https://www.rstudio.com/resources/training/). Discipline specific needs and/or aversion to adopting new software may also be contributing factors. Software environments, open or otherwise, are different and each carries distinct advantages and disadvantages given the needs of a specific field. Often decisions for choosing software do not rest with the individual workers and are made at the institutional level, where choices between open-source or proprietary software are made based on security and technical support, in addition to cost (Dhir, 2017). Proprietary software documentation is often far more consistent than that of open-source software which often relies upon community-based solutions. For non-research applications, graphical user interface (GUI) or point-and-click software solutions (e.g., Excel, JMP, Origin) that require minimal or no coding may be preferred. Cloud-based solutions are also emerging. Global Forest Watch (https://www.globalforestwatch.org/), which monitors global patterns of deforestation, is powered by Google Earth Engine. There are also "no-code" cloud-based platforms that allow researchers to design apps (Adalo, Backendless), web interfaces (Drapcode), and data models (Google's AutoML) or even automate certain tasks (Processia), all with minimal time investment and required knowledge base. There are many options for end-users to find the tool to match the job. However, the nature of forest-based

research is changing due in part to the increasing reliance upon and impact of "big data" and "disruptive technologies" (Kubik, 2020). Coding skills are necessary for researchers to meet the needs of this changing landscape. Given these considerations, it is our intention in this manuscript to highlight the functionality that R has for forestry and forest ecology-based research, rather than advocating for one software over another. We acknowledge that R, as any other software, is not a one-size-fits all solution. For example, MATLAB excels at parallel computing processes in comparison to R, and many users find data visualization more intuitive (Ozgur et al. 2017). Python is often considered easier to learn than either R or MATLAB and given its broad use across all computational fields and the speed at which it enables bulk processing, is in high demand in many industries (Ozgur et al. 2017).

R is a free, publicly available software environment for statistical computation and graphics development capable of running on Windows, MacOS, and many UNIX platforms (R Core Team, 2021). The novel innovation of R, and its commercial predecessor S, is a focus on interactive data-analysis without sacrificing the capability to create longer, more traditional programs (Ihaka and Gentleman, 1996). R is a powerful tool for scientists as it combines the ability to create publication quality graphics and perform rigorous (and reproducible) statistical analyses in one platform, is both free and supported by a large and diverse user community and facilitates open-science best practices. The utility of R for ecological research is increasingly recognized (Hesselberth et al. 2021)

This R user community is a central advantage to the environment, particularly through the development of "packages" – the "fundamental unit of shareable code" –that vastly improve the R environment. Packages are bundles of code, data, documentation, and tests that can be installed into a user's R environment based on the user's needs. A package often simplifies some function or process that a user needs. Many of the issues that are encountered in the process of analyzing data or writing code are issues that are shared by many; thus, members of the user community will often "solve" some common issue by writing a function or series of functions that can be compiled into a package. A package can then be shared with all other users, thus simplifying many issues and increasing productivity and efficiency. Packages can be accessed directly within the R environment where they are downloaded from CRAN (the Comprehensive R Archive Network; https://cran.r-project.org/), a clearinghouse for R packages with stringent and robust quality control. R-Forge (https://r-forge.r-project.org/) is another central platform for R packages with more community features including mailing lists, bug tracking, and message boards to add package and code development. There are also additional packages available via public repositories such as GitHub. However, these packages are often "developmental" and have not been subjected to the standardization and QA/QC processes involved when packages are submitted to CRAN. As such, developmental packages, while often useful, may vary in quality, documentation, and efficiency.

There has been a marked increase in the number and sophistication of packages specifically focused on common analyses and issues faced with the disciplines of forestry, forest ecology and forest science over the last few years. In this manuscript we present the results of a survey of the available packages specific for forest-based research. We have collated these findings in both an included table (Supporting Information S2) and also created *ForestAnalysisInR* (Atkins et al., 2021a,b), an R metapackage designed using the R Shiny platform to allow users to identify R packages for their specific needs, as well as link to tutorials on using R for forestry and forest ecology research. We then highlight select forestry-focused packages covering a broad swath of use cases, providing worked examples for each with independent data sets.

2. R packages

We used a systematic keyword search of package descriptions to identify available packages on CRAN and R-Forge using the keywords: *canopy, ecology, forest, forestry, inventory, lidar, mensuration, measurement, modeling, modelling, phenology,* and *remote sensing*. During our survey of R packages, we found 105 software packages relevant to forest and forest ecology research–96 on CRAN, 3 on R-Forge, and 6 on GitHub. Packages hosted on GitHub were included based on knowledge of recent publications. We collated this information into the aforementioned *ForestAnalysisInR* metapackage—also including the ability for community members to notify the authors of new packages or packages that were omitted by mistake via contact/submission hyperlink.

To make order of the universe of R packages for forestry provided by the community, we classified packages into groups based on the primary function and the research specific need(s) each package attempts to address. These group classifications are used to organize the worked examples that follow (Sections 3.1-3.6). We used eight classification groupings but only include worked examples for groups with more than one identified package—leading us to exclude Hydrology and Informatics/IoT. Groups are as follows, where *n* represents the number of packages within that category:

Community Analyses (n = 11) - These packages include functions to perform community analyses including ordinations, principal component analysis, and functional diversity analyses (e.g. *vegan, CommEcol, FD*).

Dendrochronology (n = 12) - Tree-ring analysis, chronologies, fire-history reconstructions (e.g. *burnr*), and climatology (e.g. *dplR*).

Hydrology (n = 1) – Hydrology related to forest-based research, including ecohydrology and related disciplines (e.g., *ecohydrology*).

Informatics/IoT (n = 1) – Functionality specifically within the realm of informatics and Internet of Things (IoT) technologies at the interface of forestry (e.g., *TreeTalkersCheck*).

Inventory/Mensuration (n = 27) - Forest measurements, biometrics, and allometries. This group also includes packages that contain forest inventory data from networks or projects (e.g. *rFIA, fgeo, BIEN*).

Modeling/Simulation (n = 10) - Simulation and forest modeling. This may also include specific statistical treatments of or methods for standard forestry data, such as inventory or growth modeling (e.g. *forestfit*).

Phenology (n = 13) – Working with phenology data, typically in the form of RGB image analysis or the integration of auxiliary data necessary to interpret and analyze RGB images (e.g. *phenopix, phenocamr*).

Remote Sensing (n = 36) - Interacting, processing and analyzing remote sensing data, including light detection and ranging, or LiDAR data (e.g. *rGEDI, lidR, rLiDAR, ForestGapR, forestr*); hemispherical camera data (e.g. *Sky*), hyperspectral data (e.g. *hyperspec, hsad*), satellite imagery (*raster, rgdal*).

3. Examples of applications

In the following sections, we provide worked examples using a subset of packages within each group. Many of these packages have detailed vignettes and tutorials online to help users. We have attempted to provide all the necessary information either in text or in supplemental information for users to find and access these materials. Several packages also have dedicated software or applications papers that introduce the package and provide detailed information on the scientific basis of the package and its functions, how it works, how to use it. We have made an honest attempt to be as thorough as possible in compiling this information and any omissions are unintentional.

3.1. Community Analysis: Species diversity using NEON data and the vegan package in R

Community analysis is essential in forest research as it focuses on the interactions among and between species and individuals. The *vegan* package in R, short for "vegetation analysis", is one of the most popular and widely cited of all R packages (Oksanen et al. 2020). Working in *vegan* is fairly straightforward, though is a bit different from many other

Table 1
Example output from the *vegan* package diversity() function for four NEON terrestrial sites located in the Southeast and Neotropical Domains.

Index	DSNY	JORN	LAJA	OSBS
Shannon Diversity Index	1.83	0.90	2.41	1.86
Simpson's Diversity Index	0.73	0.43	0.87	0.69
Inverse Simpson's Diversity Index	3.71	1.75	7.62	3.18

operations in R. *Vegan* prefers data to be organized in a form similar to a classic matrix format one would find in MATLAB or IDL environments. See Supporting Information Section S3 for the full worked example with all code included.

Here we use NEON data from four sites located in the Southeast and Neotropical domains: 1) DSNY - Disney Wilderness; 2) JORN - Jornada Experimental Forest; 3) LAJA - Lajas Experimental Station; and 4) OSBS - Ordway Swisher Biological Station, operated by the University of Florida (NEON Woody Plant Vegetation Structure Data, 2021). Once we import these data into R and arrange them in a wide data format we can use *vegan's* diversity() function to calculate Shannon, Simpson, and Inverse Simpson diversity indices (Table 1):

```
vegan::diversity(species.matrix, index = "shannon")
vegan::diversity(species.matrix, index = "simpson")
vegan::diversity(species.matrix, index = "invsimpson")
```

From these data we can then create a rarefaction curve, an assessment of species richness per a given number of samples or observations—here where each stem is an observation. Rarefaction curves are a useful tool in community analysis as they approximate the number of plots or samples needed to assess the species richness of a given system. Species richness comparisons among sites are only statistically valid when adjusted for the number of observations. Rarefaction curves are one means to make these comparisons. There are a couple of steps to get to this point using *vegan.*

First, we sum the number of individuals per species across our matrix to calculate the total number of observed individuals per site using rowSums(species.matrix) where species.matrix is our formatted and concatenated NEON data, then using min(rowSums(species.matrix)) we can isolate the site with the fewest number of observed individuals. These values are used to arrive at the expected species richness if all plots had the same number of observed individuals. We can then build our rarefaction dataset from these two datasets using *vegan's* rarefy() and then plot those data using rarecurve() (Fig. 3). See Supporting Information Section S4 for the full worked example with all code included.

3.2. Dendrochronology: Tree-ring chronology construction

There are many packages for working with dendrological and dendrochronological data in R. Here we use the package *dplR* (Bunn, 2008; Bunn et al. 2021) to analyze an example tree-ring data set

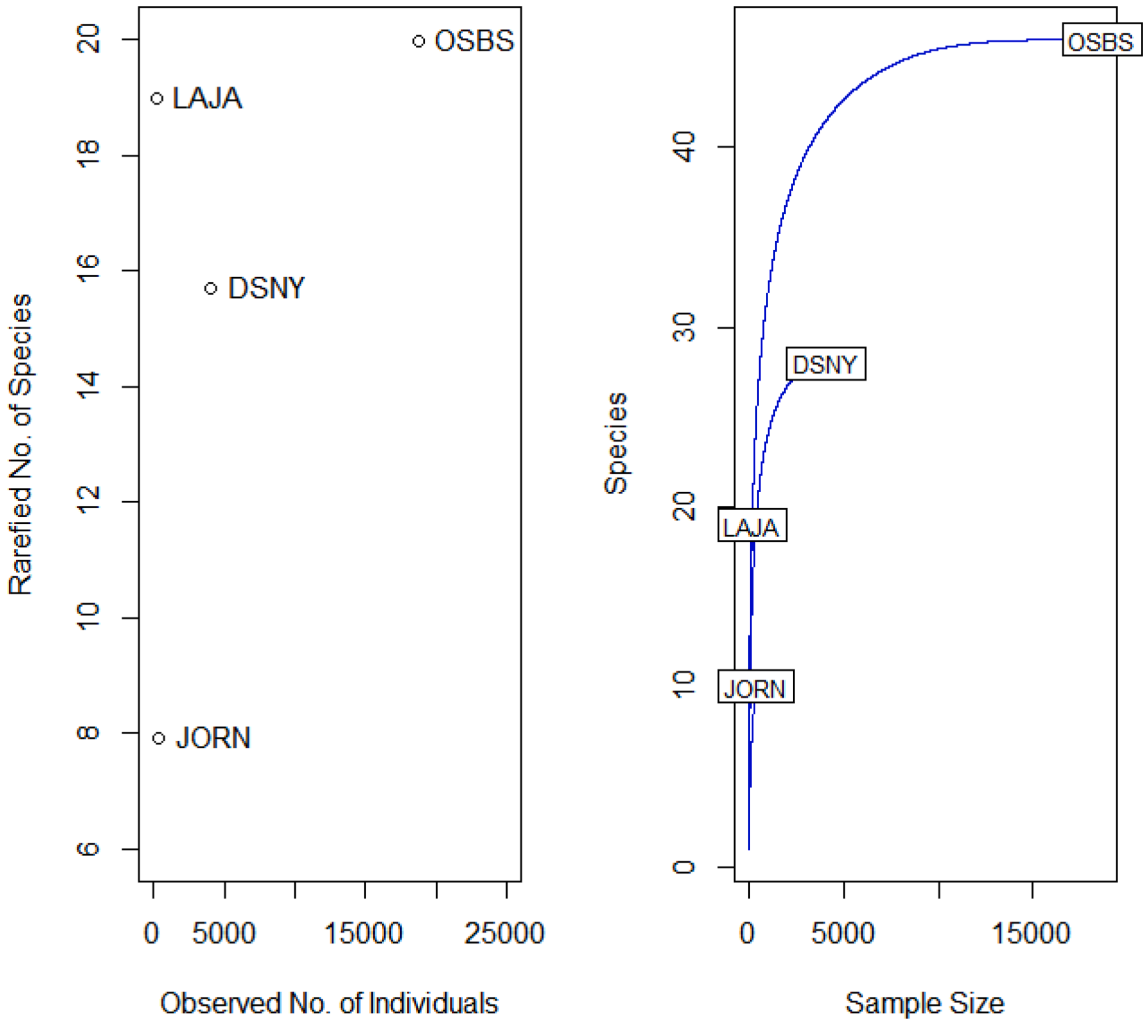


Fig. 3. In this example, though LAJA has 19 observed species, when adjusted for sampling intensity, the difference in species richness increases as seen at right. Rarefaction curves are most effective when species dominance is low or when beta diversity is high. Here we have included all species, including both the understory and overstory. OSBS is a long-leaf pine savanna ecosystem, one of the more biodiverse ecosystems in North America, which our data reflect.

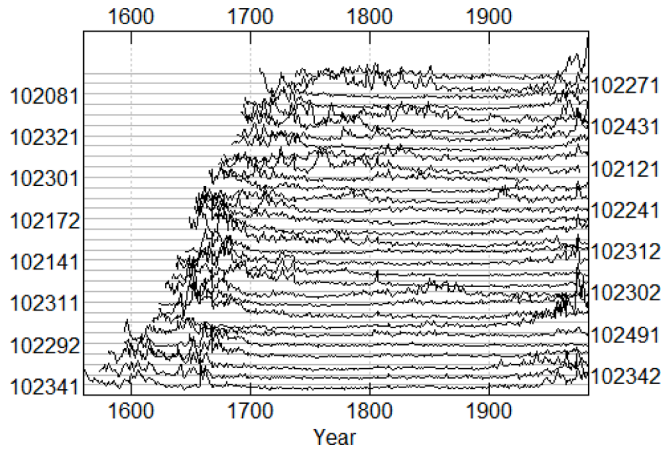


Fig. 4. Spaghetti plot created using the *dplR* package in R from 31 tree cores taken from the Kelsey Tract (Cook, 2002 - https://doi.org/10.25921/7hwh-aw70). The plot shows raw values of ring widths over time. Data accessed via NOAA from the International Tree Ring Data Bank and World Data Center for Paleoclimatology archives.

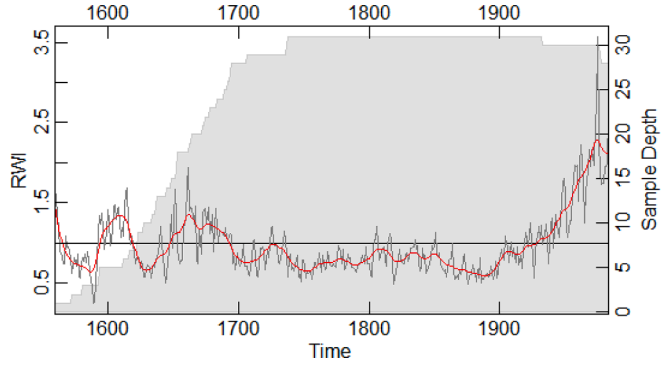


Fig. 5. Ring-width index (RWI) plot using a 20-year smoothing spline created using the *dplR* package in R showing the chronology of the Cook, 2002 Kelsey Tract data (Cook, 2002).

downloaded from the Tree ring data from the International Tree Ring Data Bank and World Data Center for Paleoclimatology archives, specifically the Kelsey Tract dataset collected in western North Carolina, USA (Cook, 2002), taken from a stand of *Tsuga canadensis* with a date range from 1560 to 1983.

Tree-ring data can come in many different file types, which may necessitate work on the user to find which package works for them and their project. That said, we found many of these to be well-documented and easily learned after following package tutorials and vignettes.

Using *dplR*, we first import our data with the read.rwl() function which is fairly forgiving of different ring-width or tree-core data types. The subsequent function, rwl.report() then provides summary statistics including the number of series in the data set (31), number of measurements (10281), average series length (331.6 years), range (424 years), span (1560-1983), and then series specific statistics including mean series intercorrelation (0.53) and standard deviation (0.06)–parenthetical values are specific to the Kelsey Tract data. Further, a description of any missing rings or years are then included (i.e. absent rings). First, we plot our data using a spaghetti plot which can be accessed using plot(data = your.data, plot.type = "spag") in base R. This allows us to visualize the data with tree-ring width plotted over time by each series (Fig. 4).

Then, to build a tree-ring chronology, we detrend the ring-width data using the detrend() function. There are many established detrending methods, here *dplR* provides options for: "Spline", "ModNegExp", "Mean", "Ar", "Friedman", "ModHugershoff". For our data set, we used the "ModNegExp" method. Then, using the chron() function on our detrended data, we build our chronology and then plot (Fig. 5).

3.3. Inventory and Mensuration: Working with forest inventory and analysis data from the USFS forest inventory analysis program with rFIA

Forest inventory and mensuration analysis often relies on basic mathematical equations or operations. Basal area for example is calculated as the area of a circle where the radius (r) in the equation πr^2 is provided by diameter-at-breast height values that are ubiquitous in forest inventory data collections. Calculating standard forestry and stand structure metrics and variables is thus relatively straightforward in R and requires only beginner to intermediate coding skills. Users, once they learn basic data import skills and summary functions, can create high quality, reproducible analyses in R from only base functions and

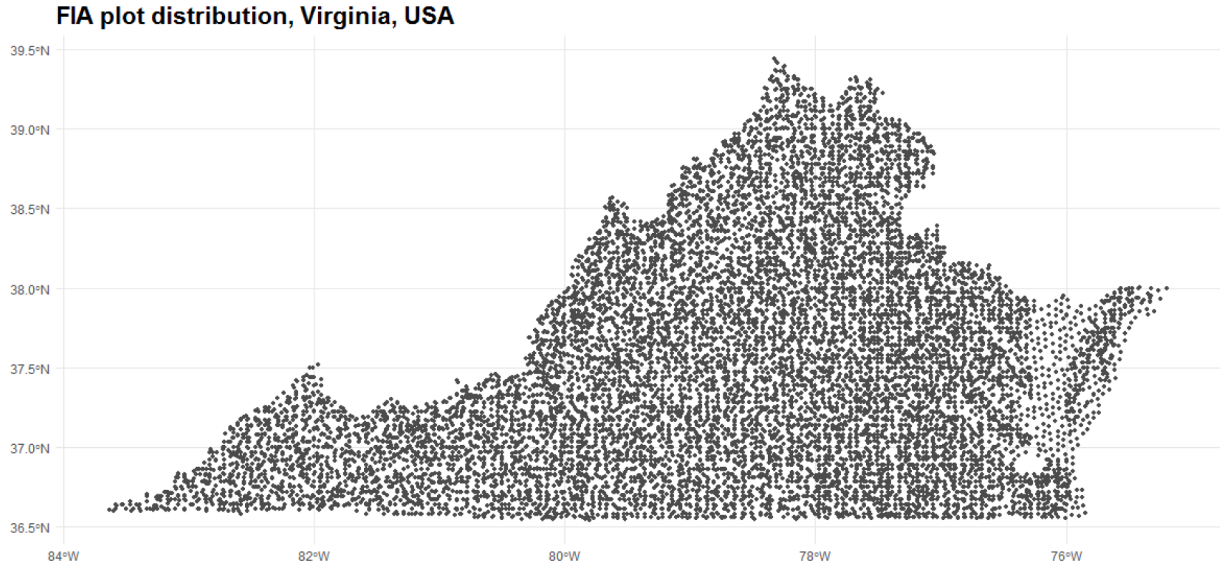


Fig. 6. FIA plot distribution in Virginia, USA, plotted using the plotFIA() command in *rFIA*. Each grey dot represents one FIA sampling plot. Many FIA plots are located on private land. To ensure privacy of landowners, USDA Forest Service introduces error into the ×, y coordinate positions of up 1.6 km (Coulston and Reams, 2004).

Fig. 7. Basal area per acre (BAA) averaged by size class (sizeClass) for all FIA data within the boundaries of Shenandoah National Park.

packages (i.e. no extra add-ons) in minutes. However, the options in R far exceed these and offer the potential of advanced analyses and capabilities.

Many of the packages we classified as Inventory/Mensuration were specific to certain databases or projects. R packages like *fgeo* (Lepore et al. 2019) or *rFIA* (Stanke et al. 2020) for example, allow users to directly access and import data from the Smithsonian ForestGEO data set and the United States Forest Services Forest Inventory and Analysis program directly into the R environment. Subsequently, users can then query, filter, and even analyze these datasets. Here, using the *rFIA* package, we analyze size class distributions of trees in Shenandoah National Park. See Supporting Information Section S4 for the full worked example with all code included.

First, after loading *rFIA*, we use the readFIA() function from the package to download FIA data. For this analysis, we used the additional argument states which allows us to specify data only from a state of interest. Here we are using the US state of Virginia:

readFIA(dir = ’./data/fia’, states = ’VA’)

This command imports all the data into a folder the user specifies, here we are using the nested subfolder ’/data/fia’. These data are not necessarily small, as this function downloads 59 files, totaling approximately 1.5 gigabytes, and includes 1.07 million individual tree measurements beginning in 2003 as well as many other data types and metadata (see https://rfia.netlify.app/publication/ems/) (Fig. 6)

The *rFIA* packages includes utilities to calculate forest and stand structural attributes, including trees per acre and basal area per acre: tpa (); tree biomass and carbon per acre: biomass(); stand structural stage distribution: standStruct(); vegetation cover by canopy layer: vegStruct (); estimates of average annual DBH, basal area, height, and net volume growth rates for individual stems, along with average annual basal area and net volume growth per acre: vitalRates(). Additionally, the package includes the ability to subset data using standard GIS shapefiles and the ability to create summary statistics.

In our example, we imported the boundary shapefile from the Shenandoah National Park (SNP) (https://public-nps.opendata.arcgis.com/). Then, using the clipFIA() function with the shapefile as the ‘mask‘, we clipped the entire Virginia FIA data set, to just those data found in the boundaries of SNP. Then using the tpa() function, we calculated basal area per acre (BAA) by size class (Fig. 7):

(tpa(snp.clip, bySizeClass = TRUE)

3.4. Modelling and Simulation: DBH distributions of two Great Lakes forest types using ForestFIT

In forestry it is often necessary to create models that describe some aspect of forest structure such as age-class, diameter, or height distributions. These models are important to forest management and policy decisions as well as ecological research as inputs into forest growth

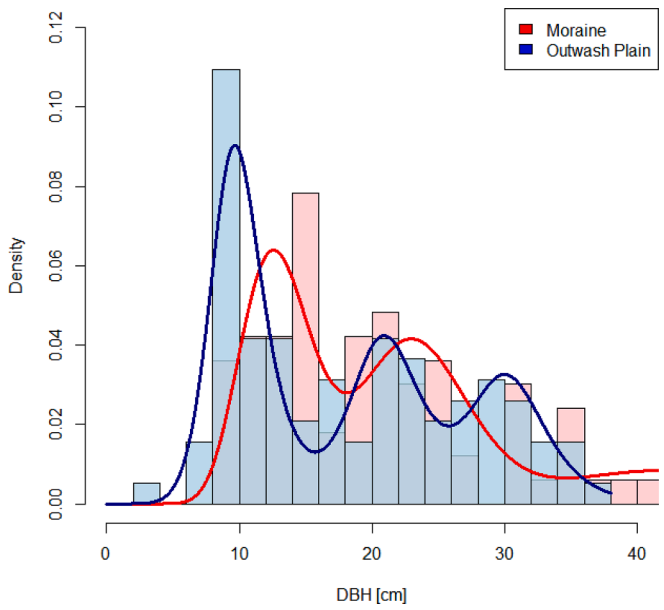


Fig. 8. Two DBH (diameter-at-breast-height) models based on forest inventory data from two, same-age forest stands located at the University of Michigan Biological Station (Gough et al., 2021; Atkins et al. 2020). The Moraine site (FoRTE plot A04W) is located on a glacial moraine with nutrient rich soils and is dominated by large ~ 120-year-old big-tooth aspen (*Populous grandifolia*). The Outwash Plain site (FoRTE plot D03E) is located ~ 3 km to the east on the outwash plain where soils are less fertile. The site is dominated by smaller northern red oak. Both stands were clear cut and burned in ~ 1900 as were many forests of the Great Lakes region of North America. Lines represent plant density models fitted from *forestfit* using a log-normal distribution. See Supporting Information Section S5 for the full worked example with all code included. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

models (Shugart and West 1980). R has found purchase within the forest growth and yield community as well (Mehtätalo and Lappi, 2020; Robinson and Harmann, 2011). The US Forest Services Forest Vegetation Simulator (FVS) model, used broadly by both researchers as well as land managers, has been ported into R as Open-FVS/rFVS (https://sourceforge.net/p/open-fvs/wiki/rFVS/#rfvs).

In this example, using the *ForestFit* (Teimouri, 2021), we calculate DBH distribution models for two forest types in northern Michigan, USA. See Supporting Information Section S6 for the full worked example with all code included.

Our data for this analysis is imported from *fortedata* (Atkins et al. 2020), an R package that includes multiple data sets created during a large, forest disturbance manipulation experiment at the University of Michigan Biological Station, initiated in 2018–the Forest Resilience Threshold Experiment, or FoRTE (Gough et al., 2021). We are using forest inventory data from pre-disturbance surveys of two distinct landscape types within the study area–one forest plot from a glacial moraine deposit which is nutrient rich and populated primarily by mature (greater than100 years old), early successional species aspen and white birch (FoRTE replicate group A), and another plot from the outwash plain landscape type, a nutrient poor area populated primarily by younger (15–80 years old) late successional species including red maple, oak, red and white pine (FoRTE replicate group D) (Pearsall et al. 1995; Scheurmann et al. 2018). First we import the data from *fortedata* and sort to two plots of interest (A04W for the moraine site and D03E for the outwash plain). We also filter to health_status = “L” to include only live trees:

```
# filter to plot
inv %>%
filter(subplot_id == “A04W” & health_status == “L”) %>%
```

Table 2
Model fit parameters for DBH distributions for the FoRTE landscape type comparisons. All values are AIC values from fitmixture() output. Bolded values represent the best model fits, the lowest AIC values.

Model distribution	Moraine	Outwash Plain
log-normal	623.0	671.9
log-logistic	624.0	**668.8**
weibull	626.2	672.6
gamma	**622.9**	669.8

```
data.frame() -> moraine
inv %>%
filter(subplot_id == "D03E" & health_status == "L") %>%
data.frame() -> outwash.plain
```

On our sorted data we use the fitmixture() function in *ForestFit*, to model the diameter distribution of our data. The package includes multiple finite mixture distributions, designated as family = … in the function. Options include: "birnbaum-saunders", "burrxii", "chen", "f", "Frechet", "gamma", "gompetrz", "log-normal", "log-logistic", "lomax", "skew-normal", and "weibull". For our analysis, we opted for "log-normal". The output from this function includes vectors of the estimated weight, shape, and scale of model parameters. Then, further, it provides a sequence of goodness-of-fit measures: Akaike Information Criterion (AIC), Consistent Akaike Information Criterion (CAIC), Bayesian Information Criterion (BIC), Hannan-Quinn information criterion (HQIC), Anderson-Darling (AD), CramerVon Misses (CVM), Kolmogorov-Smirnov (KS), and log-likelihood (log-likelihood) statistics. We tested four distribution families for each of the moraine and outwash plain plots, with the number of components, K, set to 3 (See Supporting Information Section S4). For our example data set (Fig. 8), we ranked models using the AIC values (Akaike, 1974) output from fitmixture() (Table 2.)

3.5. Phenology: Mapping spring onset in the Virginia Piedmont with PhenoCam data using phenocamr

Advances in software and remote sensing capabilities have catalyzed phenological research. Large data networks such as the PhenoCam network, provide users with high quality, consistent remote sensing imagery taken from tower-mounted cameras at 133 sites worldwide. Many open-source solutions, particularly in the R software environment, are helping to broaden this research.

Here we examine *phenocamr* (Hufkens et al. 2018), an R package that interfaces with and helps to analyze PhenoCam data directly from the network's databases. A more in-depth tutorial is available via the National Ecological Observatory Network's (NEON) Data Tutorial Series (https://www.neonscience.org/phenocam-phenor-modeling). Here we show how to analyze PhenoCAM data, taken from the PhenoCAM network for the Pace site, a mixed-temperate site located in the central Virginia Piedmont of the US (Richardson et al. 2018; Atkins et al. 2020). See Supporting Information Section S7 for the full worked example with all code included.

When working with *phenocamr*, data from a site can be imported using the download_phenocam() function which asks for: site, veg_type, frequency, phenophase, and out_dir. Frequency refers to the smoothing window value to be used. For this analysis, we are using a 3-day smoothing average. When phenophase = TRUE, a separate file that calculates phenological transition dates is created in the out_dir, the directory where data are saved:

```
download_phenocam(site = "pace",
veg_type = "DB",
frequency = 3,
phenophase = TRUE,
out_dir = "./data/phenology")
```

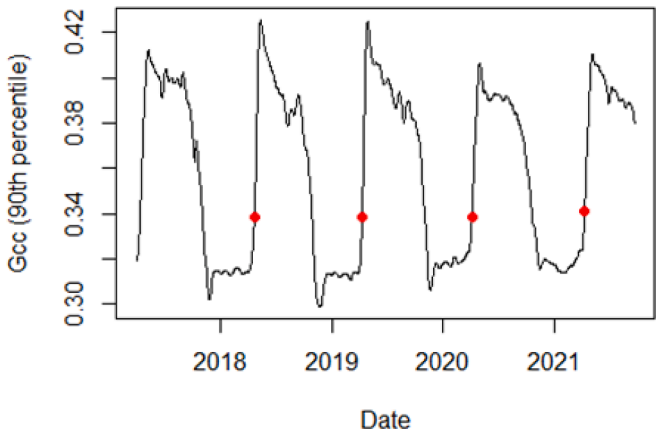


Fig. 9. Green chromatic coordinate values from the Pace Estate Tower (site = "pace"). The black line represents a 3-day *Gcc* moving average, and the red points represent the spring green-up transition dates (i.e. leaf-out). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Within the PhenoCam data there are multiple phenological metrics, but we are focusing on *Gcc*, the green chromatic coordinate. *Gcc* is the ratio of green to the entire red, green, blue values collected by a standard RGB camera. A higher *Gcc* value indicates more green, like other vegetation indices like the normalized difference vegetation index (NDVI) which uses the infrared bands, but *Gcc* uses only visible light. Specifically, the data set includes: gcc_50, gcc_75, gcc_90–the 50th, 75th and 90th percentiles (for all images passing the selection criteria) of the mean Gcc (by image) for the region of interest. This download script provides another file of transition dates–indicating when *Gcc* quantile values exceed 25% of their maximum value. After downloading these data, we then import both the 3-day averaged values and the transition dates. To assess green up or when leaf-out occurs, we filter our transition date data to only include the rising values–this is within the direction column of the PhenoCam data set and is based on that moving average, i.e. a rising transition date is a date that is occurring in the spring based on some statistically defined dormant period. Conversely, a falling value is post growing season, when *GCC* values begin to decline during leaf senescence. After this filtering procedure we can then merge these data visually and see seasonal trends (Fig. 9).

3.6. Remote Sensing: Analyzing forest succession in R with 'lidR' and 'ForestGapR'

Remote sensing science requires handling large data, often hundreds to thousands of megabytes in size per file, with analyses typically including multiple files requiring complex operations, atmospheric correction, height normalization, band algebra–necessitating both access to sufficient computing and skills development resources. The R environment and package development community have helped to meet this need. In this example, we show a brief analysis of forest succession using NEON Airborne Observatory (AOP) discrete return lidar data acquired at the Blandy Experimental Farm (NEON site BLAN) in 2019 (NEON Discrete return LiDAR point cloud, 2021). Blandy is a 300-ha biological field station owned by the University of Virginia that includes 120 ha of pasture and cropland, 40 ha of woodland, the 60 ha Virginia State Arboretum, and 80 ha of old fields in early, middle, and late succession (Bowers, 1997; Aneece et al. 2017). We focus this tutorial on one of these successional chronosequences.

Many remote sensing centered packages have other package dependencies, meaning they necessitate the installation of additional packages. For this example, we will be using functions from the *lidR*, *ForestGapR*, and *sf* packages. However, additionally, it is necessary to have the *devtools*, *ggplot2*, *raster* and *viridis* packages installed.

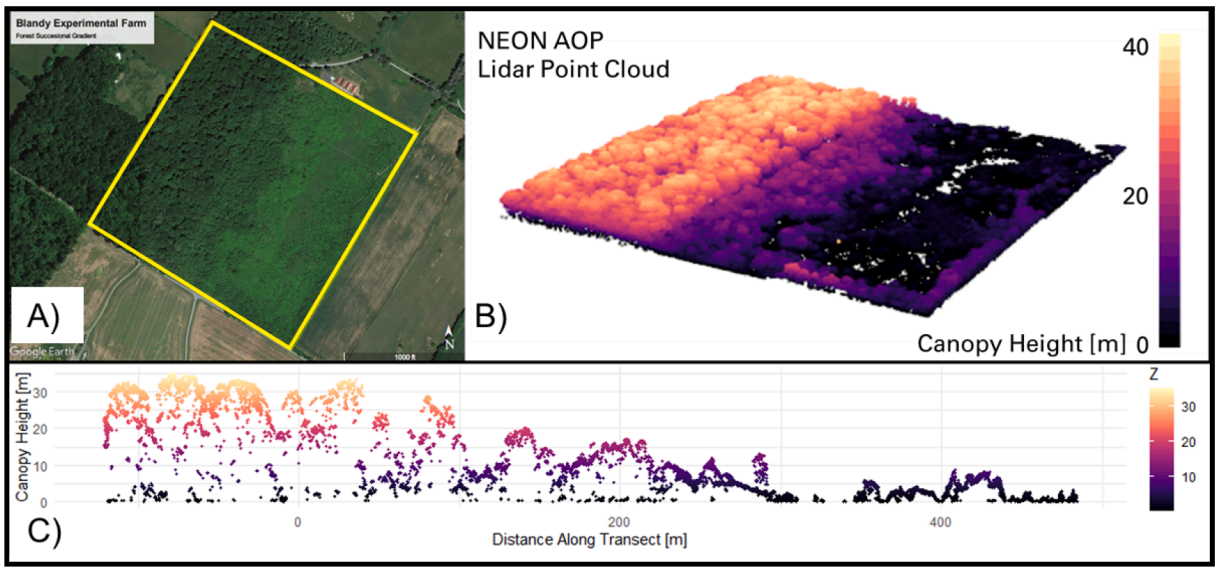


Fig. 10. National Ecological Observatory (NEON) aerial observation platform (AOP) discrete return lidar point cloud for the Blandy Experimental Farm successional field. A) Google Earth Image showing the field in 2018; B) Canopy height colored lidar point cloud; C) Transect showing decreasing canopy height with forest age with a 1 m wide buffer.

Hesselbarth et al. (2021) present a comprehensive review of R packages and approaches for remote sensing and landscape ecology more broadly which may also be of utility and interest. Here we focus primarily on using R for working with lidar data given the growing, broader interest in the utility of lidar for forestry and forest ecology.

Raw lidar point clouds often come in the .las file format, which the *lidR* package works with quite readily. For this tutorial, our example data set is a mosaicked point cloud of four separate .las files chosen because they include the area of the Blandy Experimental Farm that includes the old field abandonment study, specifically the southwestern successional chronosequence which includes an early successional field abandoned in 2001, a middle successional field abandoned in 1986, and a late successional field abandoned in 1910. Wang et al. (2010) and Aneece et al. (2017) provide greater context and detail on these successional chronosequences.

Point clouds can be imported using the readLAS() function from the lidR package. For many analyses, it is imperative to *normalize* point cloud data, creating a digital terrain model of the ground elevation that is then subtracted from each point with in the lidar point cloud. The elevation or height values in LiDAR data (the *Z* column in .las files corresponding to measured height from ground) are measures of distance that are elevation from sea level. Normalizing point clouds flattens them and sets the datum to 0. A set of functions exist in the *lidR* package to perform these operations (e.g. grid_terrain, normalize_height, grid_canopy; See Supporting Information Section S8).

Our point cloud includes some areas outside of our study area, necessitating that we clip the point cloud using a standard shapefile which can be uploaded in the workspace using the st_read() function from the *sf* package and then convert the coordinates of that shapefile into the same UTM coordinates as our point cloud using the st_transform () function. Then we want to clip the region of interest using clip_roi() from the 'lidR' package and then plot (Fig. 10).

For visualization purposes, we have also cut a transect out of this data using the clip_transect() function in *lidR*. We have also plotted these data using color palettes from the *viridis* package, a package that contains color palettes that are more accessible for those with color blindness or related issues.

One method of analyzing lidar data is to work with structural diversity and complexity metrics which can distill complex, rich data sets like lidar point clouds–which often contain millions of data points–into tractable metrics and indices. This is analogous to established methods in landscape ecology and forestry in using landscape metrics such as contagion and connectivity, or ecological methods such as vegetation indices (e.g. NDVI, EVI, SAVI) which can be derived from spectrometer data (e.g. Landsat, MODIS).

lidR includes a function grid_metrics() which can be modified using custom functions or used with built in functions, to calculate LiDAR metrics that are "gridded" at the specified resolution. Here we first define a custom myMetrics()function. Within this, we create inputs for return height (*z*), return number (*rn*), return intensity (*i*), and have the function return four lidar forest structural metrics that have been used broadly in forest research, especially for describing successional gradients and forest age:

1. Foliar height diversity (FHD) – distribution of canopy cover/biomass/leaf area among forest strata, expressed as a diversity index (MacArthur and MacArthur, 1961). The equation for FHD is in the form of the Shannon Diversity Index and is the sum across all canopy layers of the proportion of vegetation within that layer (ρ_i) multiplied by the natural log of the proportion of vegetation in that layer ($\ln\rho_i$).

$$FHD = \sum_{i=1}^{n=H_{max}} \rho_i \times \ln\rho_i$$

2. Vertical Distribution Ratio (VDR) - describes the vertical distribution of canopy elements and ranges from 0 to 1. Forests with dense canopies, but little understory exhibit low VDR values, while Forests with evenly distributed canopies have VDR values approaching 1 (Goetz et al. 2007).

$$VDR = \frac{Z_{max} - Z_{median}}{Z_{max}}$$

3. Top rugosity (R_T) - a measure of canopy complexity defined as the standard deviation of the outer canopy height (Parker and Russ, 2004).

4. p95 – the 95th height percentile of canopy height measures over a given area or distance.

We then pass this custom function to the grid_metrics() function, which produces a raster stack with each layer a raster of our calculated metric at the resolution declared in the res = option in grid_metrics(), here we use 10, corresponding to a 10 x 10 meter grid–the units are defined by the units of the .las file. Our .las file is in UTM coordinates, thus in meters. Be aware of this! We then plot the rasters (Fig. 11).

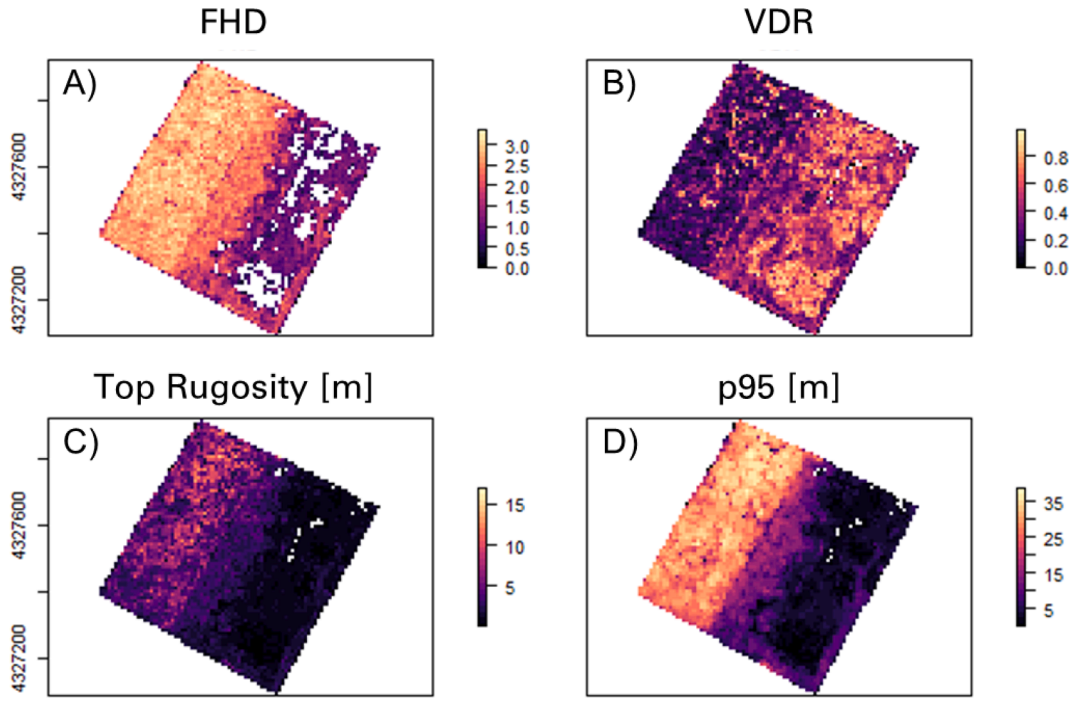


Fig. 11. Panel figure showing foliage height diversity (FHD), vertical distribution ratio (VDR), Top Rugosity, and 95th percentile return height (p95) for the Blandy SW successional field.

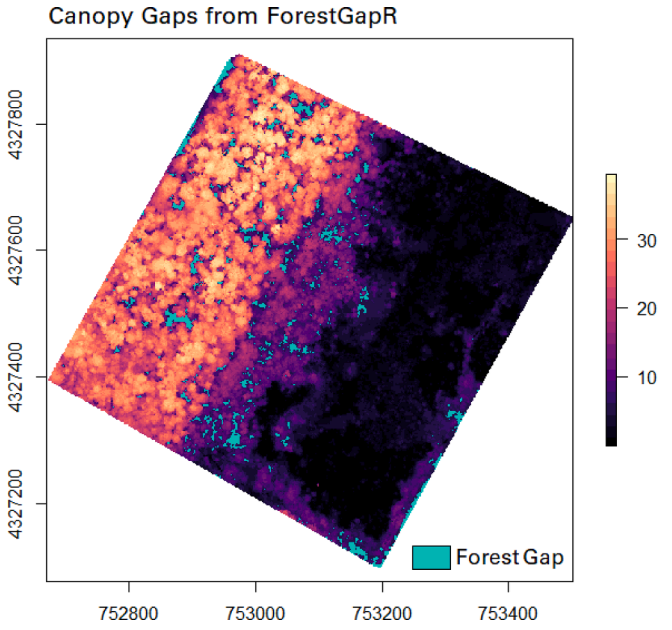


Fig. 12. Forest canopy gaps-detected layer (in teal) over the airborne lidar-derived canopy height model (units in meters).

Analyzing forest gap development using 'ForestGapR'

Forest canopy gaps are defined as openings, at a certain height threshold (e.g. 10 m), in forest canopies caused due to natural or anthropogenic disturbances (Silva et al. 2019) and are often developed as a forest ages. Detection of forest canopy gaps has been rapidly advanced using lidar (Asner et al. 2013). While forest canopy gap detection methods exist using raw lidar 3D point clouds, *ForestGapR* package in R (Silva et al. 2019) works directly with airborne lidar-derived canopy height models (CHM) which are in raster file format (e.g. .tif, .asc) –a file format that is two to three orders or more smaller in file size than .las or .laz files and far more readily available and accessible (Fig. 12).

We can use the getForestGaps() function to detect forest gaps in our CHM files. To do this, we must define our CHM with chm_layer =, our height threshold (threshold =) and our gap size (size =). For this example we are using a height threshold of 5 m and a minimum of 1 m in area and a maximum of 1,000 m, with anything greater than 1,000 m ignored:

```
gaps_blandy <- ForestGapR::getForestGaps(chm_layer = chm, threshold = 5, size = c(1, 1000))
```

To plot our gaps we can use base R plotting and add our gaps layer to a plot of the CHM:

```
plot(chm, col = viridis::magma(24))
plot(gaps_blandy, col = "#00F0D4", add = TRUE, main="Forest Canopy Gap", legend = FALSE)
```

Additional analysis on forest canopy gap-size frequency and spatial distribution as well as canopy gap dynamics can be run within ForestGapR using the GapStats() , GapSizeFDist() , GapSpatPattern() and GapChangeDec() functions (Silva et al. 2019).

4. The ForestAnalysisInR R metapackage

As we have detailed, there are over 80 packages as of October 2021 broadly or specifically useful to working with forestry data in R. We have created what we are terming a metapackage–an R package that is aimed at helping users find other packages containing specific functions to perform the analysis they are searching for. The *ForestAnalysisInR* package can be downloaded directly from GitHub and installed using install_github() function from the *devtools* package. This package launches an R Shiny application, a GUI interface that allows users to identify and search a dynamic table we have generated in the preparation of this manuscript. This package can be run locally via direct download:

```
devtools::install_github("atkinsjeff/ForestAnalysisInR")
library(ForestAnalysisInR)
## Launch the app
launchRFA()
```

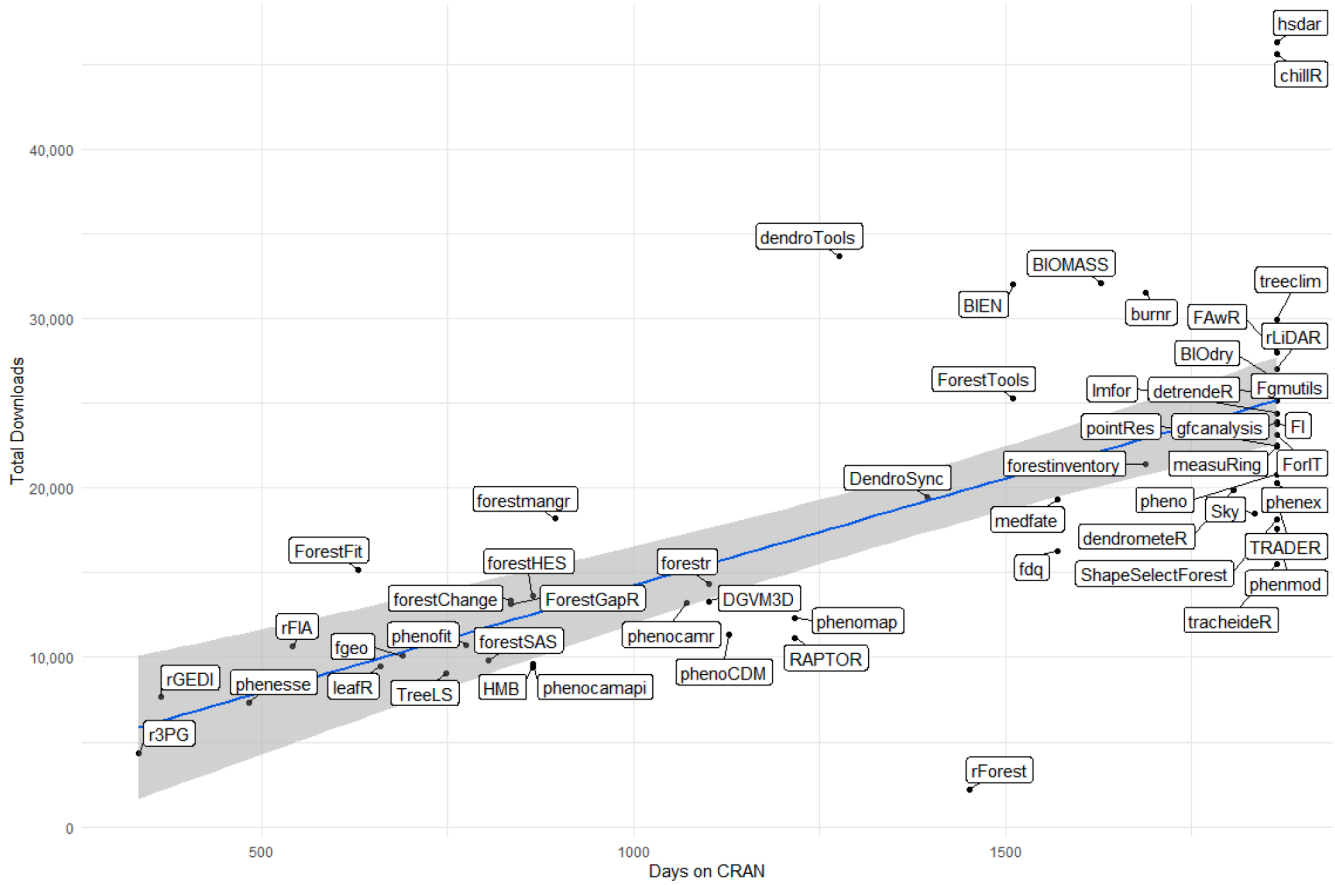


Fig. 13. Note, both packages *phenopix* (total downloads 134,802, days on CRAN 1,365) and *lidR* (total downloads 85,967; days on CRAN 1,541) were removed due to an incredibly high downloads per day ratio which placed them well out of the range of the plot. Plot current as of August 1, 2021.

This package will be updated periodically to reflect change in the R package environment, specifically including changes in package availability. A Google form submission link is included in the R Shiny app interface to allow community members to suggest the inclusion or update of any relevant R packages. The R Shiny app can be accessed on the web at https://atkinsjeff.shinyapps.io/ForestAnalysisInR/,

5. Final considerations

Open-source software solutions, such as the R programming language, offer a multitude of functions, algorithms, and data processing capabilities to the research community. However, R is broadly underutilized in forestry and forest ecology relative to ecology (Fig. 2) (Russel, 2021). Forestry and forest ecology are already data intensive areas of study, requiring strong computational skills. Yet, in an era of big data and broad collaboration, a reliance on proprietary software creates limitations that can be surmounted by open source, cross-platform solutions (Wilson et al. 2017). A large and growing user community exists within the natural science disciplines and educational materials to help onboard new users are expanding rapidly. Many universities are switching statistics courses from software such as SPSS, Excel, SAS, JMP, to R and Python. A survey of Canadian universities found that in the field of Psychology, R is the second or third most common software used in statistics courses at the introductory, intermediary, and graduate level (Davidson et al. 2019). We are aware of no similar study in the natural sciences, but a comparison of questions posed on the website Cross Validated, a website for crowdsourcing answers to statistics questions used readily by both students in statistics classes and practitioners in the field, shows that R far and away exceeds all other statistics software with 25,804 active questions posed compared to Python (3,953), Excel (412), SAS (686), JMP (46), and STATA (1,374). While this is not a perfect proxy by any means, it is an indication of the prevalence of R in statistical computing. However, despite the tremendous growth in the use of R over the last decade, it is primarily used in the academic sphere, which is a major consideration for education (Robinson, 2017). For students to be successful in securing employment, they must have the training needed for their desired occupation.

R and similar open-source software solutions also afford a level of continuity that proprietary software does not. As a rule, open-source software platforms are decentralized and run by consortiums or non-profit groups. Whereas proprietary software solutions are typically owned and issued by corporations and for-profit entities. Any proprietary software can be discontinued at any time for either market reasons, corporate restructuring, or replacement. These disruptions affect scientific research. For example, when Google discontinued Google Fusion Tables in 2019, many researchers found themselves in a lurch. Google Fusion Tables had an avid following within the research community and filled a necessary niche in linking data between Google Sheets and Google Earth Engine. This allowed researchers to create maps rather easily from tabular data and to share these data and pipelines. When the software was discontinued, this created disruptions that resulted in broken analysis pipelines and data products, as well as lost productivity. Open-source software are not at the whims of market or cultural trends however, and over time, have proven to be consistent and resilient.

R's greatest strength, the diverse amount of user and community generated software packages that expand the capabilities of the base language, however, can be a weakness. For users to get the most utility from R, they need to have access to and the ability to learn how to use packages pertinent to their research. Hasselberth et al. (2021) highlight

this issue in the field of landscape ecology—increasing awareness and access to the additional packages that are available can greatly increase usage. This creates strong potential for additive, downstream effects including but not limited to increases in reproducibility of studies, data access, reuse of data, and collaboration within and among research groups.

Despite the number of available packages in R for forest-specific research, greater potential exists . Many common forest statistical needs are not met by forestry specific R packages, though may be addressed by more general packages. For example, the *Vegan* (Oksanen et al. 2020) package in R includes many different functions to deal with common forest community statistics. Packages focusing on forestry specific statistics could be incredibly useful. Regarding remote sensing, packages dealing with hyperspectral or terrestrial LiDAR are also needed (Calders et al. 2021).

Package maintenance also becomes an issue. R is a dynamic language, and packages periodically need to be updated to ensure compatibility. This maintenance, as is too often the case with data management, goes unfunded and under supported. Mechanisms to support the community are necessary. Additional support could also help expand internal documentation within packages (i.e. expand help files and examples) as well as help create additional tutorials and educational resources online. Many R packages that we surveyed have been downloaded 10 s of thousands of times (Fig. 13) and have contributed to the analyses underlying thousands of scientific papers, management plans, dissertations, class projects, etc.

6. Conclusion

While there are many options of open-source software solutions for research, the R programming language offers a viable and rich option for forestry and forest ecology research with a diverse and plentiful user community and development base. Further adoption of R in forest-research could increase the visibility and reproducibility of research, while also increasing productivity.

Data and Code Availability

Code and data in support of ForestAnalysisInR and the bibliometrics analysis can be found at: https://github.com/atkinsjeff/ForestAnalysisInR and the R Shiny interface is available at: https://atkinsjeff.shinyapps.io/ForestAnalysisInR/

Code and analysis for worked examples can be found in Supporting Information.

CRediT authorship contribution statement

Jeff W. Atkins: Conceptualization, Methodology, Software, Formal analysis, Visualization. **Atticus E.L. Stovall:** Conceptualization, Methodology, Software, Validation. **Carlos Alberto Silva:** Conceptualization, Methodology, Software, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The National Ecological Observatory Network is a program sponsored by the National Science Foundation and operated under cooperative agreement by Battelle. This material is based in part upon work supported by the National Science Foundation through the NEON Program. AELS acknowledges support from the NASA Postdoctoral Program Fellowship and the National Aeronautics and Space Administration (grant no. 80NSSC17K0110).

We would also like to extend our gratitude to the editor, associate editor, and reviewers who provided helpful feedback and perspective that strengthened this manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.foreco.2021.119813.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE transactions on automatic control 19 (6), 716–723.

Aneece, I.P., Epstein, H., Lerdau, M., 2017. Correlating species and spectral diversities using hyperspectral remote sensing in early-successional fields. Ecology and evolution 7 (10), 3475–3488.

Asner, G. P., Kellner, J. R., Kennedy-Bowdoin, T., Knapp, D. E., Anderson, C., & Martin, R. E. (2013). Forest Canopy Gap Distributions in the Southern Peruvian Amazon. PLoS ONE, 4, e60875. https://doi.org/10.1371/journal.pone.0060875.

Atkins, J.W., Stovall, A.E., Yang, X., 2020. Mapping Temperate Forest Phenology Using Tower, UAV, and Ground-Based Sensors. Drones 4 (3), 56.

Atkins, J.W., Silva, C.A., Stovall, A.E.L., 2021a. ForestAnalysisInR: An R Metapackage for Forestry and Forest Ecology Analyses. R package version (1). https://github.com/atkinsjeff/ForestAnalysisInR.

Atkins, J.W., Agee, E., Barry, A., Dahlin, K.M., Dorheim, K., Grigri, M.S., Bond-Lamberty, B., 2021b. The fortedata R package: open-science datasets from a manipulative experiment testing forest resilience. Earth System Science Data 13 (3), 943–952.

Bowers, M.A., 1997. University of Virginia's blandy experimental farm. Bulletin of the Ecological Society of America 78 (3), 220–225.

Buck, S., 2015. Solving Reproducibllity. Science 348 (6242), 1403.

Bunn AG (2008). "A dendrochronology program library in R (dplR)." Dendrochronologia, 26(2), doi: 10.1016/j.dendro.2008.01.002 (URL:.

Bunn, A., Korpela, M., Biondi, F., Campelo, F., Merian, P., Qeadan, F., Zang, C., 2021. dplR: Dendrochronology Program Library in R. R package version 1.7.2. https://CRAN.R-project.org/package=dplR.

Calders, K., Adams, J., Armston, J., Bartholomeus, H., Bauwens, S., Bentley, L.P., Chave, J., Danson, F.M., Demol, M., Disney, M., Gaulton, R., Krishna Moorthy, S.M., Levick, S.R., Saarinen, N., Schaaf, C., Stovall, A., Terryn, L., Wilkes, P., Verbeeck, H., 2020. Terrestrial laser scanning in forest ecology: Expanding the horizon. Remote Sensing of Environment 251, 112102. https://doi.org/10.1016/j.rse.2020.112102.

Carr, R.M., Lane-Fall, M.B., South, E., Brady, D., Momplaisir, F., Guerra, C.E., Montoya-Williams, D., Dalembert, G., Lavizzo-Mourey, R., Hamilton, R., 2021. Academic careers and the COVID-19 pandemic: Reversing the tide. Science Translational Medicine 13 (584). https://doi.org/10.1126/scitranslmed.abe7189.

Cook, E.R., 2002-04-26. NOAA/WDS Paleoclimatology - Cook - Kelsey Tract - TSCA - ITRDB NC005 [indicate subset used]. NOAA National Centers for Environmental Information. https://doi.org/10.25921/7hwh-aw70 (Accessed 19 July 2021).

Cook, E.R. (2002-04-26): NOAA/WDS Paleoclimatology - Cook - Kelsey Tract - TSCA - ITRDB NC005. [Site 15464]. NOAA National Centers for Environmental Information. https://doi.org/10.25921/7hwh-aw70. June 14, 2021.

Coulston, J. W. and G.A. Reams (2004). The effect of blurred plot coordinates on interpolating forest biomass: a case study. In: Proceedings of the joint meeting of the 15th annual conference of the International Environmetrics Society and the 6th international symposium on spatial accuracy assessment in natural resources and environmental sciences.

Davidson, H., Jabbari, Y., Patton, H., O'Hagan, F., Peters, K., Cribbie, R., 2019. Statistical software use in Canadian university courses: Current trends and future directions. Teaching of Psychology 46 (3), 246–250.

Dhir, S., et al., 2017. Adoption of open-source software versus proprietary software: An exploratory study. Strategic Change 26 (4), 363–371.

Goetz, S., Steinberg, D., Dubayah, R., Blair, B., 2007. Laser remote sensing of canopy habitat heterogeneity as a predictor of bird species richness in an eastern temperate forest, USA. Remote Sensing of Environment 108 (3), 254–263.

Gough, C.M., Atkins, J.W., Bond-Lamberty, B., Agee, E.A., Dorheim, K.R., Fahey, R.T., Grigri, M.S., Haber, L.T., Mathes, K.C., Pennington, S.C., Shiklomanov, A.N., Tallant, J.M., 2021. Forest Structural Complexity and Biomass Predict First-Year Carbon Cycling Responses to Disturbance. Ecosystems 24 (3), 699–712.

Hesselbarth, M.H.K., Nowosad, J., Signer, J., Graham, L.J., 2021. Open-source tools in R for landscape ecology. Current Landscape Ecology Reports 6 (3), 97–111. https://doi.org/10.1007/s40823-021-00067-y.

Hufkens, K., Basler, D., Milliman, T, Melass, E.K., Richardson, A.D. (2018). An integrated phenology modelling framework in R: modelling vegetation phenology with phenor Methods in Ecology & Evolution, 9(2), 1-10.

Ihaka, R., Gentleman, R., 1996. R: a language for data analysis and graphics. Journal of computational and graphical statistics 5 (3), 299–314.

Inouye, D.W., Underwood, N., Inouye, B.D., Irwin, R.E., 2020. Support early-career field researchers. Science 368 (6492), 724–725.

Kubik, G. H. (2020). Technology as a driver of future change in the forest sector: projected roles for disruptive and emergent technologies. In: Dockry, Michael J.; Bengston, David N.; Westphal, Lynne M., comps. Drivers of change in US forests and forestry over the next 20 years. Gen. Tech. Rep. NRS-P-197., 50-58.

Lepore, M., Gabriel Arellano, Richard Condit, Stuart Davies, Matteo Detto, Erika Gonzalez-Akre, Pamela Hall, Kyle Harms, Valentine Herrmann, David Kenfack,

Suzanne Lao, Sean McMahon, Sabrina Russo, Kristina Anderson-Teixeira, Graham Zemunik and Daniel Zuleta (2019). fgeo: Analyze Forest Diversity and Dynamics. R package version 1.1.4. https://CRAN.R-project.org/package=fgeo.

MacArthur, R.H., MacArthur, J.W., 1961. On bird species diversity. Ecology 42 (3), 594–598.

McNutt, M., 2014. Reproducibility. Science 343 (6168), 229. https://doi.org/10.1126/science:1250475.

NEON (National Ecological Observatory Network). Woody plant vegetation structure, RELEASE-2021 (DP1.10098.001). https://doi.org/10.48443/e3qn-xw47. Dataset accessed from https://data.neonscience.org on August 01, 2021.

NEON (National Ecological Observatory Network). Discrete return LiDAR point cloud, RELEASE-2021 (DP1.30003.001). https://doi.org/10.48443/6e8k-3343. Dataset accessed from https://data.neonscience.org on August 01, 2021.

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., ... & Yarkoni,. (2015). Promoting an open research culture. Science, 348(6242), 1422-1425.

Oksanen, J., F. Guillaume, Blanchet, M. F., Roeland Kindt, Pierre Legendre, Dan McGlinn, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, Eduard Szoecs and Helene Wagner (2020). vegan: Community Ecology Package. R package version 2.5-7. https://CRAN.R-project.org/package=vegan.

Mehtätalo, Lauri, Lappi, Juha, 2020. Biometry for forestry and environmental data: With examples in R. Chapman and Hall/CRC.

Ozgur, C., Colliau, T., Rogers, G., Hughes, Z., 2017. MatLab vs. Python vs. R. Journal of Data Science 15 (3), 355–371.

Parker, G.G., Russ, M.E., 2004. The canopy surface and stand development: assessing forest canopy structure and complexity with near-surface altimetry. Forest Ecology and Management 189 (1-3), 307–315.

Pearsall, D. R. (1995). Landscape ecosystems of the University of Michigan Biological Station: ecosystem diversity and ground-cover diversity (Doctoral dissertation, University of Michigan).

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Richardson, A.D., K. Hufkens, T. Milliman, D.M. Aubrecht, M. Chen, J.M. Gray, M.R. Johnston, T.F. Keenan, S.T. Klosterman, M. Kosmala, E.K. Melaas, M.A. Friedl, S. Frolking, M. Abraha, M. Alber, M. Apple, B.E. Law, D. Baldocchi, T.A. Black, P. Blanken, D.M. Browning, S. Bret-Harte, N. Brunsell, S.P. Burns, E. Cremonese, A.R. Desai, A.L. Dunn, D.M. Eissenstat, S.E. Euskirchen, L.B. Flanagan, B. Forsythe, J. Gallagher, L. Gu, D.Y. Hollinger, J.W. Jones, J. King, O. Langvall, J.H. McCaughey, P.J. McHale, G.A. Meyer, M.J. Mitchell, M. Migliavacca, Z. Nesic, A. Noormets, K. Novick, J. O'Connell, A.C. Oishi, W.W. Oswald, T.D. Perkins, R.P. Phillips, M.D. Schwartz, R.L. Scott, O. Sonnentag, J.E. Thom, and J. Verfaillie. 2018. PhenoCam Dataset v1.0: Vegetation Phenology from Digital Camera Imagery, 2000-2015. ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/1511.

Robinson, D. (2017) The Impressive Growth of R https://stackoverflow.blog/2017/10/10/impressive-growth-r/.

Russel, MB (2020) "Nine Tips to Improve Your Everyday Forest Data Analysis", Journal of Forestry, Volume 118, Issue 6, November 2020, Pages 636–643, https://doi.org/10.1093/jofore/fvaa034.

Robinson, A.P., Harmann, J.D., 2011. Forest analytics with R. Springer, New York, pp. 1–339.

Scheuermann, C.M., Nave, L.E., Fahey, R.T., Nadelhoffer, K.J., Gough, C.M., 2018. Effects of canopy structure and species diversity on primary production in upper Great Lakes forests. Oecologia 188 (2), 405–415.

Shugart Jr, H. H., & West, D. C. (1980). Forest succession models. BioScience, 30(5), 308-313.

Silva, C.A., Pinage, E., Mohan, M., Valbuena, R., Almeida, D., Broadbent, E., Jaafar, W., Papa, D., Cardil, A., Klauberg, C., 2019. ForestGapR: An R Package for Airborne Laser Scanning-derived Tropical Forest Gaps Analysis. Methods Ecol Evolution. 10, 1347–1356. https://doi.org/10.1111/2041-210X.13211.

Stanke, H., Finley, A.O., Weed, A.S., Walters, B.F., Domke, G.M., 2020. rFIA: An R package for estimation of forest attributes with the US Forest Inventory and Analysis database. Environmental Modelling & Software 127, 104664. https://doi.org/10.1016/j.envsoft.2020.104664.

Teimouri, M. (2021). ForestFit: Statistical Modelling for Plant Size Distributions. R package version 0.7.1. https://CRAN.R-project.org/package=ForestFit.

Wang, Jin, Epstein, Howard, Wang, Lixin, 2010. Soil CO_2 flux and its controls during secondary succession. J. Geophys. Res.: Biogeosci. 115 (G2).

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., Teal, T.K., Ouellette, F., 2017. Good enough practices in scientific computing. PLoS computational biology 13 (6), e1005510. https://doi.org/10.1371/journal.pcbi.100551010.1371/journal.pcbi.1005510.g001.

Zou, W., Jing, W., Chen, G., Lu, Y., Song, H., 2019. A survey of big data analytics for smart forestry. IEEE Access 7, 46621–46636.