**Tracking Forest Height Growth Over Time with ICESat-2 ATL08**

William Thomas Poncy

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Forestry

Valeria A. Thomas (Co-advisor)

Randolph H. Wynne (Co-advisor)

P. Corey Green

May 13, 2025

Blacksburg, VA

Keywords: remote sensing, forests, LiDAR, ICESat-2

Copyright 2025 William T. Poncy

**Tracking Forest Height Growth Over Time with ICESat-2 ATL08**

William Poncy

ABSTRACT

Accurate quantification of forest structure is necessary for forest inventory, growth prediction, and carbon stock estimates. Since late 2018, the ICESat-2 mission has employed a photon-counting LiDAR system to estimate along-track ground and canopy heights above the WGS 84 ellipsoid. This mission provides repeat coverage in 100 m data segments across the Southeastern United States, a region selected for its high timber productivity. In this research, we use multiple years of ICESat-2 ATL08 data to identify canopy height growth over time in a coastal region of North Carolina. ATL08 canopy height estimates demonstrate strong alignment (R2=0.88, RMSE=2.64 meters) with coincident airborne laser scanning. Equivalence tests reveal that locations sampled by ICESat-2 are homogenous within a margin of 2 meters. FIA remeasurement plots within our study area provide an expected net growth of 1.68 meters over a five-year timeframe, and a rate of change of 0.34 meters per year. Exchanging space for time reveals canopy height growth is detectable with five years of ATL08 data. However, the influence between land cover type and disturbance history on canopy height growth is uncertain, likely due to convolved nature of these factors and sampling pattern of ICESat-2 data collection. Ultimately, this research can serve as a proof-of-concept for using multiple years of spaceborne LiDAR to detect changes in forest structure. Future research should explore the detection of canopy height growth with spaceborne LiDAR in other regions of the globe, as solid results could demonstrate the potential of the ATL08 product in global forest structural monitoring in the face of a changing climate.

**Tracking Forest Height Growth Over Time with ICESat-2 ATL08**

William Poncy

GENERAL ABSTRACT

~~To accomplish this we 1) validate ATL08 canopy height measurements with coincident airborne lidar data across our study area. Next, we 2) assess the homogeneity of our study area forests to identify spatial biases in canopy height. After, we 3) establish expected rates of change in canopy height from FIA remeasurement plots. We then 4) exchange space for time, and use multiple years of ATL08 data to identify the emergence of canopy height change across our study area. Finally, we 5) remove observations from locations demonstrating spatial bias in canopy height, to assess how much canopy height change is driven by spatial bias. Exchanging space for time shows statistically significant trends in height change, namely growth, over time in the study area. Removing observations demonstrating spatial bias shows muffled trends in height change over time, likely due to a shorter timeframe and smaller sample size. Further research should explore the emergence of temporal trends in other regions of the globe and across ATL08 collection characteristics, as solid results could demonstrate the potential of the ATL08 product in tracking forest height change over time.~~

**Table of Contents**

List of figures…..…………………………………………………………………………vi

List of tables….………………………………………………………………………….vii

1. Introduction…………………………………………………………………………....1
   1. Forests………………………………………………………………………….....1
   2. Forest Structure…………………………………………………………………...1
   3. Forest Structure with LiDAR……………………………………………………..2
   4. Airborne Laser Scanning……………………………………………………...….3
   5. Spaceborne LiDAR……………………………………………………………….4
   6. Research Questions……………………………………………………………….6
2. Methods………………………………………………………………………………..6
   1. Study Area………………………………………………………………………..6
   2. Data Products……………………………………………………………………..8
      1. ALS………………………………………………………………………..8
      2. ICESat-2 ATL08…………………………………………………………..8
      3. National Land Cover Database…………………………………………..10
      4. Landscape Change Monitoring System………………………………….11
      5. Forest Inventory & Analysis Program…………………………………...11
   3. Data Processing…………………………………………………………………12
      1. ATL08……………………………………………………………………12
      2. Extracting ALS Data Within ATL08 Segments…………………………14
      3. Retrieving Study Area FIA Data…………………………………………17
   4. Statistical Methods Used………………………………………………………..18
      1. Linear Regression Model………………………………………………...18
      2. Equivalence Test…………………………………………………………18
      3. Wilcoxon Rank-Sum Test……………….……………………………….19
      4. Mann Kendall Test……………………………………………………….19
      5. Bonferroni Correction……………………………………………………20
   5. Phase 1: Validating ATL08 Canopy Heights with ALS………………………...20
   6. Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection……………...21
      1. Sample Representation………………….………………………………..22
      2. Sample Location Equivalence……….……………………………...……22
   7. Phase 3: Reference Canopy Height Growth…………………………………….23
   8. Phase 4: Identifying Canopy Height Growth with ATL08……………………...24
3. Results………………………………………………………………………………..25
   1. Phase 1: Validating ATL08 Canopy Heights with ALS………………………...25
   2. Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection ...…………...26
      1. Sample Representation……….…………………………………………..26
      2. Sample Location Equivalence……………………………………………28
   3. Phase 3: Reference Canopy Height Growth…………………………………….30
   4. Phase 4: Identifying Canopy Height Growth with ATL08……………...………32
4. Discussion……………………………………………………………………………38
5. Conclusion…………………………………………………………………………...44
6. Bibliography…………………………………………………………………………47
7. Appendix……………………………………………………………………………..69
   1. ~~Python Script to Extract Canopy Height Metrics from ALS Data……………...69~~
   2. ~~Python Script to Extract Canopy Height Metrics from ATL08 Granules………77~~
   3. ~~R Script to Access North Carolina FIADB and Conduct Analyses of Phase 3…85~~
   4. ~~Python Script to Filter ATL08 Data and Extract Land Characteristics…………98~~
   5. ~~Python Script to Create Polygons of ATL08 Segments……………………….106~~
   6. ~~R Script for Analyses of Phases 1, 2, 4, and 5…………………………………112~~

**List of Figures**

**Figure 1.** Study area in North Carolina……..………………………………………….....7

**Figure 2.** ATL08segments within study area after data quality filters………………….14

**Figure 3.** Workflow for extracting ALS data within ATL08 polygons…………………15

**Figure 4.** ALS data extracted within ATL08 segments…………………………………15

**Figure 5.** Overlapping collection scheme of ALS projects……...……………………...16

**Figure 6.** Counties of North Carolina within study area……...…………………………17

**Figure 7.** Misinterpretation of spatial biases in canopy height as temporal change…….21

**Figure 8.** Population of forests throughout study area…………………………………..23

**Figure 9.** Correlation of canopy heights estimated by ATL08 and ALS………………..26

**Figure 10.** Distribution of sample canopy heights………………………………………27

**Figure 11.** Grid of sample equivalence tests across bound values………………………28

**Figure 12.** Distribution of population canopy heights…………………………..………29

**Figure 13.** Grid of population equivalence tests across bound values…………………..30

**Figure 14.** Distribution of actual height increment values from FIA plots……………...31

**Figure 15.** Distribution of six-year actual height net growth values from FIA plots……32

**Figure 16.** Spatial distribution of segments across study area by group………….……..39

**Figure 17.** Proportion of segments in EPA Level 2 Ecoregions by group………………40

**Figure 18.** ATL08 canopy height estimation in disturbed vs. undisturbed stands………42**~~List of Tables~~**

**~~Table 1.~~** ~~Summary statistics of total and actual height measurement…………………...30~~

**~~Table 2.~~** ~~Results of Phase 4 regression……………………………………………...…...32~~

**~~Table 3.~~** ~~Results of Phase 4 t-test ………………………….………................................33~~

**~~Table 4.~~** ~~Results of~~~~Phase 4 generalized additive model…...……….…………………...34~~

**~~Table 5.~~** ~~Results of Phase 4 Mann-Kendall test………………………………….……...35~~

**~~Table 10.~~** ~~Number and proportion of ATL08 segments by group…………….….……...38~~

1. **Introduction**
   1. **Forests**

Supporting forest ecosystems is critical to combating climate change (Psistaki et al., 2024). Forests are the backbone of the land carbon sink (Pan et al., 2024), covering ~31% of the terrestrial land area and sequestering almost twice as much carbon as they emit (FAO and UNEP, 2020; Harris & Gibbs, 2021). Forests impact not only global carbon cycle dynamics but mediate local and microclimates through evapotranspiration and incident solar radiation. Furthermore, forests provide notable benefits to human health, lowering blood pressure and muscle tension while facilitating recovery from attention fatigue (Karjalainen et al., 2009).

However, disturbance regimes serve to alter forest structural traits (height, crown closure, etc.) thereby disrupting the age structure and spatial continuity of forests (Sturtevant & Fortin, 2021). These disturbances can result in stress-induced mortality, changes in species composition, and reduced tree regeneration capacity (Khaine & Woo, 2015). Disturbance regimes themselves are sensitive to climatic shifts, with warming temperatures likely to increase the frequency of fire, insect, and pathogen disturbances in coming years. Despite these rising pressures, current modeling techniques inadequately represent in the complex interplay between vegetation dynamics, climate pattens, and disturbance agents (Seidl et al., 2017). To effectively assess ecosystem resilience in the face of a changing disturbance landscape, dynamic ecosystem models require information on forest structural diversity over large scales (Mitchell et al., 2023).

* 1. **Forest Structure**

Forest structure is generally characterized by the horizontal and vertical distribution of its plant and organic components throughout space, described by variations in species and age classes, canopy layers, and diameter classes (Zenner & Hibbs, 2000). Forest structure emerges from the interplay of biophysical and ecological processes throughout time, and has regulatory effects on the microclimate and quality of ecosystem services (Hui et al., 2019). Variables describing forest structure, such as basal area and tree height heterogeneity, have demonstrated greater influence on forest productivity than species diversity (Bohn & Huth, 2017). Despite variability across ecoregions, forest structural complexity is positively correlated with ecosystem stability and productivity (X. Liu et al., 2024). However, uncertainties remain in the 3D structure of global forests, and by extension, the global carbon budget (Hall et al., 2011). As such, accurate quantification of global forest structure is necessary to model ecosystem processes, carbon allocation, and growth dynamics to enhance forest resilience in response to mounting disturbance pressures (Beland et al., 2019).

* 1. **Forest Structure with LiDAR**

Light detection and ranging (LiDAR) is an active remote sensing technique that uses laser pulses to calculate distance to objects and map out in high detail the environment surrounding the sensor (Harrap & Lato, 2010). Importantly, LiDAR data avoids the georeferencing of optical remote sensing-where the 3D world is projected onto an 2D image-and the inevitable geometric distortion (P. Dong & Chen, 2017). The waveform of LiDAR energy returned to the sensor can be recorded as peaks in the waveform curve (discrete return), arrival time of individual photons reflected from the laser pulse (photon counting), or the entire distribution (full waveform) (Podest, 2021). LiDAR sensors can be mounted to terrestrial, airborne, and spaceborne platforms, resulting in application across domains for efforts in landslide detection, autonomous vehicle navigation, and archaeological discovery (Jaboyedoff et al., 2012; Risbøl & Gustavsen, 2018; Roriz et al., 2022).

LiDAR has seen growing use in remote sensing of forests, primarily due to its rapid collection and precise quantification of 3D forest structure. Whereas passive remote sensing techniques cannot penetrate the canopy layer, LiDAR can describe the subcanopy vegetation and underlying topography of a forest ecosystem, which have strong influence on drought response and post-disturbance stability (Jarron et al., 2020; H. Zhao et al., 2022). LiDAR is also used for estimating aboveground biomass of forest ecosystems more precisely than optical sensors, an essential task for quantifying the global carbon budget and upholding international climate and emissions agreements (Neuenschwander et al., 2023; Zolkos et al., 2013). Commonly, LiDAR data is used to describe the structural complexity of forests by quantifying canopy cover, foliage height diversity, top rugosity, leaf area index, and leaf area density, which can then be used to infer ecosystem functions (Atkins et al., 2023). This study will focus on canopy height as the primary forest structural characteristic, as tree height is data is fundamental to forest management activities for estimating stem volume, describing stand health, and modeling forest biomass (F. Chen et al., 2023; Mielcarek et al., 2020).

* 1. **Airborne Laser Scanning**

For research at scales larger than forest stands, remotely sensed data is often combined with field inventories to generate a comprehensive view of forest structure (Lamping et al., 2021). Airborne laser scanning (ALS) is a form of LiDAR remote sensing that distributes emitted light along an aircraft’s flight path (Maltamo et al., 2014). ALS sensor technology can collect data at hundreds of points per square meter, leading to a growing adoption for quantifying vegetation structure by overcoming difficulties of traditional terrestrial sampling techniques in remote or topographically complex areas (Sumnall et al., 2022; Wilkes et al., 2015). Data from ALS is also used in forest inventory efforts, commonly with an area-based-approach or through individual tree detection (Xiang et al., 2024). The spatial coverage offered by airborne laser scanning is invaluable for forest research across ecological gradients, with programs like the U.S. Geological Survey’s (USGS) 3D Elevation Program and the National Ecological Observatory Network’s Airborne Observation Platform providing freely available LiDAR data for a range of forested environments.

While the accuracy of height estimations from LiDAR are dependent upon the forest species composition, airplane flight height, and scanner pulse density, LiDAR sensors may underestimate tree height due to the unlikely nature of laser pulses returning from the top of a tree (X. Yu et al., 2004; K. Zhao et al., 2018). While single-area unit ALS forest measurements may demonstrate bias, site-index and aboveground biomass estimations improve as spatial scale and plot size increase (Duncanson et al., 2020; Meyer et al., 2013; Noordermeer et al., 2018, 2020). Multiple studies have found that tree height can be estimated by ALS to within half a meter for pine species (Andersen et al., 2006; Roberts et al., 2005). ALS from the USGS 3D Elevation Program, which is used in this study, has demonstrated an error of 1-2 meters for estimating tree height and dominant height of forests (Oh et al., 2022; Ribas-Costa, Gastón, & Cook, 2024).

Ultimately forest height change is itself heavily influenced by external factors, and estimation of year-over-year height change with ALS could be subject to non-trivial error (Guerra-Hernández et al., 2021; Socha et al., 2017). Identifying height change with repeat ALS requires ample time between collections for the change to exceed the noise associated in single-year height measurements. Yu et al. (2004) observed that plot-level growth could be ascertained at a scale of 10-15 centimeters in a 21-month timeframe. In a study of spruce, pine, and birch, Hyyppä et al. (2003) found a standard error of less than 5 centimeters in estimating height growth at the stand level. In a stand of temperate, mature red pine, Hopkinson et al. (2008) observed an approximate growth rate of 0.4 meters per year, and that the LiDAR-estimated growth error falls below an acceptable uncertainty value of 10% after 3 years.

Despite impressive performance in estimating forest height and growth, ALS data is not without its shortcomings, with collection costs prohibiting repeat or global coverage (Hancock et al., 2021; M. Liu et al., 2019). Moreover, differences in ALS acquisition parameters, namely point density, leads to variability in forest structural estimation (LaRue et al., 2022)

* 1. **Spaceborne LiDAR**

Estimates of forest aboveground biomass and density are critical to understanding the impacts of land use change on the global carbon cycle (Dubayah et al., 2022). To overcome limitations present in ALS, such as inconsistent collections and lack of wall-to-wall coverage, spaceborne LiDAR missions have been commissioned facilitating regional-to-global scale ecosystem insights (Coops et al., 2021). These missions exchange repeated measurements of small, specific areas for broad spatial coverage, sampling earth’s surface during continued orbit throughout time. When harmonized with other data sources (optical imagery, ALS), spaceborne LiDAR data enabled creation of create gridded biomass products at national and regional scales, filling gaps of field-based or ALS campaigns (F. Chen et al., 2023; Dubayah et al., 2022; Neuenschwander et al., 2024)

NASA’s Ice, Cloud and Land Elevation Satellite (ICESat) mission collected global waveform LiDAR data from 2003 to 2009 with a laser altimeter system, demonstrating strong performance in elevation retrieval with relative accuracy and precision of ~2 and ~14 centimeters (Gong et al., 2011; Shuman et al., 2006). While the ICESat mission’s main objectives were to quantify changes in ice sheet elevation (Markus et al., 2017), the derived data products saw use in efforts measuring land topography, vegetation canopy heights, and atmospheric composition (Schutz et al., 2005). More recently, NASA launched the Global Ecosystem Dynamics Investigation (GEDI) mission in late 2018 to directly assessing land surface carbon balance and biodiversity using waveform LiDAR (Dubayah et al., 2020). Building on the success of the first ICESat mission, NASA also launched the ICESat-2 mission in September 2018 to continue measurements of ice sheet elevation through improvements in beam design, along-track sampling rate, and footprint diameter (Markus et al., 2017).

* 1. **ICESat-2**

The ICESat-2 satellite employs the Advanced Topographic Laser Altimeter System (ATLAS), a photon-counting LiDAR instrument, to measure earth’s surface (Carabajal & Boy, 2020). ATLAS operates at a 532 nm (green) laser wavelength, optimized for maximal photon detection with current technology (Neumann et al., 2019a). Prior to exiting the satellite, ATLAS splits the laser into six beams arranged into three beam pairs, with each beam pair containing a strong and weak beam with an energy transmission ratio of 4:1 (Neumann et al., 2019a)

For greater coverage of Earth’s surface, beams within a pair are separated by 90 meters and beam pairs are separated by ~3 kilometers (Markus et al., 2017). ATLAS’s reduced laser power requirement allows a smaller payload aboard the ICESat-2 bus (Sun et al., 2020). With a spacecraft velocity of ~7 km/s and a laser frequency of 10 kHz, the ATLAS instrument achieves an along-track sampling interval of 70 centimeters. This generates strong overlap between shots to determine terrain slope along and across the orbital track. In contrast to the GEDI mission, which collects data only within latitudes ±51.6°, ICESat-2 provides near-global coverage between 88° N and S (Markus et al., 2017; Pronk et al., 2024). Moreover, ICESat-2’s orbit altitude of ~500 km and 91-day repeat cycle facilitate analyses of seasonal variation for its coverage areas (Wang et al., 2024).

Ultimately, the consistency of surface measurements by the ATLAS instrument is influenced by atmospheric conditions and land cover attributes. Over highly reflective surfaces like land ice, up to 10 signal photons may be returned per laser pulse. Over vegetated areas with lower surface reflectance, the energy and detection ratio of ATLAS may drop to ~2.5:1, with only 0-4 signal photons returned per laser pulse (Neuenschwander et al., 2023, 2024).

Though primarily designed for ice sheet monitoring, ICESat-2 measurements have been developed for characterizing ocean elevation, inland water height, and terrestrial vegetation. This research will employ the ICESat-2 L3A Land and Vegetation Height (ATL08) Version 6 data product, which estimates ground and canopy elevations in 100-meter segments.

Commonly, assessing the performance of ATL08 segments in estimating canopy height requires comparison against coincident ALS. While studies commonly report an underestimation of canopy heights by ATL08 segments-primarily due to the sampling rate of the ATLAS instrument-the magnitude of error varies with forest conditions (A. Liu et al., 2021; Malambo & Popescu, 2021; Neuenschwander et al., 2020). Comparison of ATL08 canopy height estimations to ALS data has shown mean biases ranging from -1.71 meters across diverse biomes, with strong performance in conifer forests, to 3.05 meters in boreal forests.

ATL08 segments collected at nighttime with the strong beam of the satellite consistently yield the lowest height estimation errors due to reduced atmospheric interference and greater energy output. Use of weak beam data is generally discouraged for canopy height estimation (F. Chen et al., 2023; Guerra-Hernández et al., 2022; A. Liu et al., 2021; Neuenschwander et al., 2020; Rai et al., 2024; J. Yu et al., 2022). Moreover, canopy height estimation errors are minimized in a range of 40-80% canopy cover. Sparse vegetation has an inherently low probability of generating returns to adequately describe canopy height, while in dense canopy cover the algorithm for the ATL08 data product struggles to interpolate the underlying terrain, introducing error into the resulting canopy height estimation (Neuenschwander et al., 2020).

Ultimately, the accuracy of ATL08 forest height estimation is a function of many factors, including acquisition characteristics (e.g., beam strength, time of collection), terrain variability, forest density, geolocation accuracy, atmospheric noise, local disturbance history, and temporal coincidence to reference data (Fernandez-Diaz et al., 2022; Malambo & Popescu, 2021; Neuenschwander et al., 2023; Rai et al., 2024).

These challenges notwithstanding, ICESat-2’s near-global coverage and high geolocation accuracy (<5m) offer a unique advantage over GEDI to track forest dynamics over time (Luthcke et al., 2021; Neuenschwander & Magruder, 2019). The question remains as to whether spaceborne LiDAR can fully describe changes in vegetation structure when and where ALS data is not available (B. Li et al., 2022; Malambo & Popescu, 2021). By leveraging its strengths and effectively handling the limitations in its data quality, this study aims to showcase ICESat-2’s potential for forest structural monitoring in the context of a shifting climate.

* 1. **Research Questions**

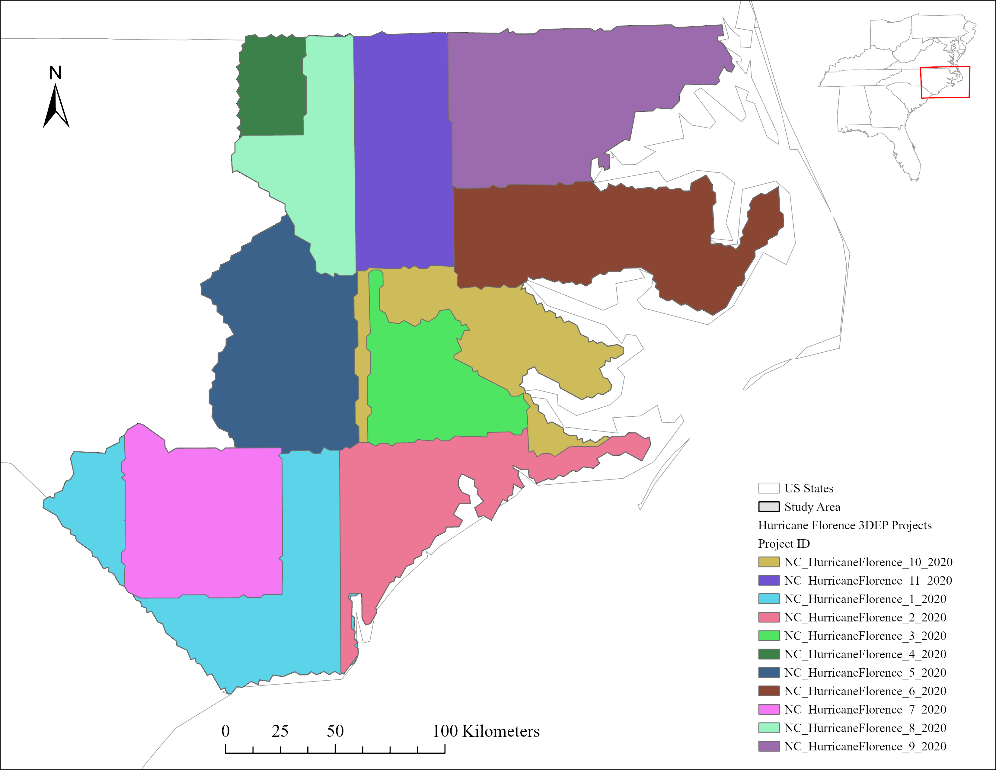
Our research aims to use multiple years of spaceborne LiDAR data to monitor changes in forest structure. To that end, we aim to answer the following research questions:

1. To what extent can ICESat-2 ATL08 detect forest height change over time?
2. How is this ability influenced by forest cover type and disturbance history?
3. **Methods**

Our approach for answering the research questions is organized into four sequential phases-validation, addressing bias, FIA reference, and change detection-each described in subsections below.

* 1. **Study Area**

Hurricane Florence, a slow moving category one hurricane, made landfall in coastal North Carolina on September 14th, 2018 (Callaghan, 2020). Florence brought record-breaking levels of rain, surpassing 30 inches of rain in some regions of North Carolina, resulting in 53 fatalities and an estimated $16-44 billion in damages to public and private infrastructure (Griffin et al., 2020; Paul et al., 2019). In response, the USGS 3D Elevation Program (3DEP), under the Additional Supplemental Appropriations for Disaster Relief Act of 2019 (H.R. 2157), collected high-resolution ALS data to quantify the extent of hurricane damage and support infrastructure recovery efforts (Columbia Environmental Research Center, 2019). The study area of this research is defined as the collective boundaries of the 3DEP Hurricane Florence ALS projects (Figure 1). The study area comprises ~5.1 million total hectares in the Eastern portion of North Carolina.



*Figure 1: Boundaries of USGS 3D Elevation Program ALS projects collected in response to Hurricane Florence. Projects were flown in late 2019 and early 2020.*

This study area was selected for several factors. While repeat coverage by ICESat-2 is unlikely for the stand-or-plantation-scale, the chosen study area is large enough to ensure repeat coverage even after data quality filters. Moreover, the collection of data to validate ATL08 canopy height estimations can be financially cumbersome. However, the chosen study area contains extensive ALS data across ecological gradients.

Topography has noticeable effects on ATL08 canopy height estimations (A. Liu et al., 2021; Malambo & Popescu, 2021). The study area demonstrates low terrain variability, with an elevation standard deviation of 27.3 meters and an average slope of 0.78 degrees. As of 2023, the land cover of the study area is predominantly Woody Wetlands (29%), Cultivated Crops (26%), and Evergreen Forest (19%) as determined from the National Land Cover Database (NLCD). Finally, the chosen study area falls within the P*inus taeda* (loblolly pine) historical natural range. As such, the regular tree spacing, consistent tree heights, and fast tree growth observed by plantations in the study region should be more conducive to accurate ATL08 canopy height estimation (Baker & Langdon, 1990).

* 1. **Data Products**

This research utilizes ICESat-2 ATL08 canopy height data to assess changes in forest height over time. To focus exclusively on forested areas, land cover data from the Annual National Landcover Database was utilized was used to identify forested regions within the study area. The disturbance history of these areas was derived from the Landscape Change Monitoring System developed by the U.S. Forest Service. ALS data from the USGS 3DEP was used as ground truth reference data to validate ATL08 canopy height estimates, assess spatial biases in ICESat-2 coverage locations. Finally, data from the U.S. Forest Service’s Forest Inventory and Analysis program provide local, repeat measurement plot surveys for determining changes in canopy height throughout time expected within our study area.

* + 1. **ALS**

ALS data from the Hurricane Florence region is contained in 11 projects of the USGS 3DEP flown between December 7th, 2019, and February 28th, 2020. All projects utilized Reigl VQ1560i or Reigl VQ1560ii instruments to collect data, meeting the Quality Level 1 requirements specified by the 3DEP LiDAR Base Specification. This guarantees that reported heights have an average error (RMSE) within 10 centimeters, and an average point density of 8 points per square meter. In consideration of the strong vertical accuracy and high point density of this ALS data, we determined it to be adequate in serving as ‘ground truth’ for validating ATL08 canopy heights. USGS 3DEP LiDAR Point Clouds was accessed on May 17, 2025, from https://registry.opendata.aws/usgs-lidar. A custom Python script (Appendix 7.1) was created in Python 3.12.5 (Rossum & Drake, 2010) to stream the 3DEP data from the Entwine Point Tiles (EPT) format in during data processing.

* + 1. **ICESat-2 ATL08**

Photon measurements from the ATLAS instrument are aggregated into several data products. The Global Geolocated Photon Data (ATL03) product records heights above the WGS84 ellipsoid for photons detected by ATLAS (Neumann et al., 2023). From the geolocated photon data, the ICESat-2 L3A Land and Vegetation Height (ATL08) data product is derived through a Differential, Regressive, and Gaussian Adaptive Nearest Neighbor (DRAGANN) method. This process filters out background noise and estimates the land surface and vegetation height, labeling individual photons as noise, ground, canopy, or top of canopy (Malambo & Popescu, 2024; Neuenschwander et al., 2020; Neuenschwander & Pitts, 2019).

The ATL08 product is used in this research as it reports several canopy parameters in 20-meter and 100-meter segments, including the mean, minimum, maximum, and median of relative and absolute height of canopy photons. Metrics are also reported for relative and absolute canopy heights at the following percentiles: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95. ATL08’s primary canopy height metric, *h\_canopy*, uses a segment’s 98th percentile canopy relative heights to represent the top of canopy, as maximum canopy height may include background noise.

For this study, all ATL08 data collected between October 14th, 2018, and January 17th, 2025 within the extent of the study area was downloaded from the NASA Earthdata Search (<https://search.earthdata.nasa.gov/search>), yielding 831 ATL08 granules for further analysis in the HDF5 file format. These granules were processed in a custom Python script (see Appendix 7.2) to extract metrics related to the vegetation, terrain, and spacecraft operation of each 100-meter segment

* + 1. **National Land Cover Database**

To aid research on land surface characteristics within human-environmental systems, the USGS Resources Observation and Science Center developed the National Land Cover Database (NLCD). This suite of data products used Landsat imagery to provide information on land cover and change processes from 2001 to 2021. NLCD products have been used widely in remote sensing research, with applications in land management, risk mitigation, and environmental planning (Homer et al., 2020). In 2024, the USGS released the Annual National Land Cover Database Collection 1 Science Products, improving legacy NLCD datasets by providing annual land cover classification and land change assessments for each year from 1985 to 2023 (U.S. Geological Survey, 2024). These products are created through a framework that combines continuous change detection, geospatial deep learning, and probability-based post-processing.

Within the collection, the Land Cover product categorizes the land surface into broad natural and artificial surface cover types for each mapping year, employing a modified Anderson Level II classification system to identify 16 land cover classes. While ATL08 data includes segment land cover information in the *segment\_landcover* attribute-derived from the Copernicus Land Cover (ANC18) data product at a 100-meter resolution-the NLCD Land Cover dataset was chosen for this research for its higher spatial resolution and alignment with historical disturbance data provided by the U.S. Forest Service. The NLCD data used in this research was accessed from the Multi-Resolution Land Characteristics Consortium data archive (https://www.mrlc.gov/data). Land cover for the contiguous United States was downloaded for the most recent year available, 2023, to provide the most up-to-date representation of forest extent within the study area. This data is provided as a single-band raster in the TIFF file format, with a 30-meter cell spatial resolution

* + 1. **Landscape Change Monitoring System**

The Landscape Change Monitoring System (LCMS) is a suite of remotely sensed-data products developed by the U.S. Forest Service to monitor landscape dynamics across the United States (Housman et al., 2021). LCMS provides three annual products using an ensemble of predictor models to characterize ecosystem processes over four decades of change: change, land cover, and land use.

LCMS relies on input imagery from Landsat 4, 5, 7, and 8 (Collection 1 Tier 1) and Sentinel-2a and -2b (level 1C top of atmosphere reflectance data). From these inputs, clouds and cloud shadows are masked with the CFmask, cloudScore, 2Cloudless, and Temporal Dark Outlier Mask algorithms (Chastain et al., 2019; Foga et al., 2017; Housman et al., 2021). The processed outputs are assembled into an annual time series and segmented using the LandTrendr and CCDC methods (Kennedy et al., 2010, 2018; Zhu & Woodcock, 2014). The segmented data is incorporated with terrain data in a random forest model to generate predictions for land change, land cover, and land use.

The land change product is focused on vegetation cover, with subproducts categorizing change processes into fast loss, slow loss, and gain. Fast loss events refer to an abrupt disturbance to vegetation cover (fire, harvesting, mechanical disturbance), slow loss events indicate gradual structural or spectral decline, while gain represents vegetation recovery. A summary product is provided for each of these change processes at a 30-meter spatial resolution, with pixel values identifying the most recent year of occurrence of the given change between 1985 and 2023. For this research, the Fast Loss change summary product was downloaded for the Conterminous United States from the Landscape Change Monitoring System data archive (https://apps.fs.usda.gov/lcms-viewer/home.html#download)

* + 1. **Forest Inventory & Analysis Program**

The Forest Inventory & Analysis (FIA) program is an effort by the U.S. Forest Service’s Research and Development Branch aimed at monitoring the distribution and health of forest resources. With data collected in all 50 states, U.S. territories, and Freely Associated States (Republic of Marshall Islands, Federated States of Micronesia, Republic of Palau), FIA provides data on forest resources domestically and abroad. A core tenet of the FIA is the Nationwide Forest Inventory (NFI), a nationally distributed network of forest inventory plots that receive repeat measurements every 5-10 years. Depending on the plot, the NFI provides data on land use, soil, down woody material, understory vegetation, and individual tree height measurements. Data for the NFI plots within the study area was obtained from the North Carolina Forest Inventory and Analysis database using a custom R script (see Appendix 7.3). Though precise FIA plot locations are fuzzed for privacy concerns (U.S. Department of Agriculture Forest Service, 2024), the county of each plot record is provided with Census Bureau FIPS codes, a level of spatial-specificity sufficient for this research

* 1. **Data Processing**

All spatial data sources, with exception of FIA data, were first projected from their native coordinate systems to the NAD 1983 North Carolina State Plane projection system (EPSG:32119). This projection system maintained a horizontal distance unit of meters- necessary for comparing ATL08 and ALS canopy height estimations-while minimizing geometric distortion across the extent of the study area in North Carolina.

* + 1. **ATL08**

Using LiDAR to identify statistically meaningful change in canopy height over time requires change significant enough to surpass the margin of error associated with single-point-in-time canopy height estimations. In the case of ICESat-2, this is especially challenging given the issues of atmospheric noise sensitivity, probability of canopy height underestimation by the ATL08 algorithm, and sampling scheme of the satellite. As such, careful handling of all potential sources of error in ATL08 data is necessary to optimize its application for given analyses (Feng et al., 2023; B. Li et al., 2022; W. Li et al., 2020; A. Liu et al., 2021; Rai et al., 2024; Wang et al., 2024; Xi et al., 2022).

ATL08 data was processed in a custom Python script to filter observations and intersect with land cover characteristics (see Appendix 7.4). First, ATL08 segments with invalid (3.402e+38) canopy height values were removed. To minimize biases introduced by low vegetation or high noise, canopy heights were then filtered to values between 2 meters and a global maximum (Cao et al., 2016; Rai et al., 2024). Following the approach of Malambo & Popescu (2024), a global maximum for canopy heights was calculated as *kP, k* is a constant factor of (1.2) and *P* is the 98th percentile of all ATL08 canopy heights in our dataset. For our study area, the calculated global maximum was 39.42 meters. This was effective in removing erroneously high canopy heights estimated by the ATL08 algorithm. Additional filters selected only segments collected by strong beams, at nighttime, within the months of May to September to minimize the effects of solar noise and vegetation phenology on canopy height estimation (Neuenschwander et al., 2020). As done by Malambo & Popescu (2024), segments with a multiple scattering warning flag greater than zero or a cloud confidence flag above one were removed to mitigate possible atmospheric interference.

After these preliminary filters on data quality, ATL08 segments were geolocated with the GeoPandas library from the *latitude* and *longitude* attributes, identifying center coordinates of signal photons for each segment (Jordahl et al., 2020; Neuenschwander et al., 2023). These points were intersected with annual NLCD land cover to determine the land cover of each segment’s centroid for the year 2023. Using the most recent year of NLCD available ensures that land cover classification reflects the latest conditions within the ICESat-2 mission lifetime. Forested segments were identified by selecting only NLCD values of 41 (Deciduous Forest), 42 (Evergreen Forest), 43 (Mixed Forest), or 90 (Woody Wetlands). The centroids of these forest segments were then intersected with the annual LCMS Fast Loss change product from the year 2023 to provide insight into the disturbance history of these segments. For segments with a disturbance event reported by the Fast Loss product, the years since the last disturbance was calculated as the difference between the year of ICESat-2 collection and the most recent disturbance year. For this research, segments with disturbances occurring *after* the year of collection were removed, as ATL08 canopy height values may no longer reflect the forest conditions present. Additionally, segments with less than 3 years since the last disturbance event were removed to provide at least 3 years of forest regrowth and recovery.

A map of the united states

AI-generated content may be incorrect.

*Figure 2: ATL08 segments collected in study area after data quality filters. Segments colored by year of collection to demonstrate ICESat-2 sampling pattern.*

* + 1. **Extracting ALS Data Within ATL08 Segments**

For this research it is necessary to extract 2020 ALS data within each individual ATL08 segment, regardless of the segment’s collection year. To generate the rectangular segments of the ATL08 product, a polygon of 100 x 11 meters was created around the centroid coordinates of each segment, and rotated to align with the ICESat-2 track inclination using a custom python script (see Appendix 7.5; Rai et al., 2024). These polygons were then reprojected to the Web Mercator projection system, which the ALS data uses in its native storage format (Hobu, Inc., 2025). These polygons serve as cropping geometries for the ALS data.

ALS data was processed in a custom pipeline (Figure 3) using the *PDAL* library (see Appendix 7.1; Butler et al., 2024). First ALS data was streamed from any 3DEP project intersecting a given polygon using the polygon’s coordinate bounds. The Simple Morphological Filter (SMRF) was used to interpolate the ground surface as it minimized Type I errors in ground classification, which could’ve resulted in inaccurate canopy height estimation (Pingel et al., 2013). Heights above the interpolated ground surface were then calculated using a nearest neighbor approach. This normalized point cloud was cropped to the geometry of the respective polygon to match the polygon’s along-track rotation (Figure 4).

A diagram of a diagram

AI-generated content may be incorrect.

*Figure 3: Workflow of python script used to extract ALS data collected within ATL08 polygons.*

A satellite image of a field

AI-generated content may be incorrect.

*Figure 4: ALS data extracted within ATL08 segments over a plantation area. ALS data is only non-ground returns with reasonable vegetation height as described below.*

For each ATL08 segment, the normalized ALS point cloud was converted to a Pandas DataFrame to calculate metrics for return heights above ground (The pandas development team, 2024). We selected only non-ground returns with heights above ground between 2 meters and the previously determined global maximum 39.42 meters to match the filters applied to ATL08 data. These returns were deemed as *vegetation* returns, from which several metrics were calculated for the return heights including minimum, maximum, mean, median, and the following percentiles: 90, 95, 98, 99. Finally, the ratio of vegetation returns to total returns provided an approximation of vegetation density (Neuenschwander et al., 2020). These ALS metrics were then joined to the corresponding ATL08 segment using a unique identifier.

Due to the overlapping collection scheme of the ALS data, many ATL08 segments intersected with multiple 3DEP projects (Figure 5), resulting in the calculation of vegetation metrics for multiple ALS projects within a single segment.

A screen shot of a computer screen

AI-generated content may be incorrect.

*Figure 5: Track of ATL08 segments (white polygons) passing through overlapping area (green) of separate ALS projects (blue, orange).*

To remove duplicate segments, only the ALS project providing the highest number of vegetation returns per segment was retained. Additionally, the original ALS project boundaries are generalized, as the initial crop of ATL08 data to the 3DEP boundaries resulted in several segments with zero ALS returns within their bounds. These were also removed from analyses, resulting in a final “working set” of53,905ATL08 segments. In summary, our working set contains 6 years of ICESat-2 ATL08 segments (2019-2024), from which vegetation height metrics were also calculated from 2020 airborne laser scanning collected within each segment’s geometry.

* + 1. **Retrieving Study Area FIA Data**

A copy of the North Carolina Forest Inventory and Analysis database (FIADB) was created in PostgreSQL using a custom repository that provides scripts for manipulating FIA data in a local environment (radt0005, 2025/2025). North Carolina FIA remeasurement plots were accessed using a custom R script (see Appendix 7.3). This script selected only those in the evaluation group 372023, corresponding to plots in North Carolina (state code 37) remeasured during 2023. Using data from the most recent FIA remeasurement interval of 2023 provides the most up-to-date insight into the expected patterns of forest growth within the study area. These plots were further subset to only those from the 40 counties within the study area using county code attributes (Figure 6).

A map of the state of maine

AI-generated content may be incorrect.

*Figure 6: Counties of North Carolina within study area.*

Unique plot sequence numbers were extracted from FIA plots within the study area counties. For these sequence numbers, the plot, condition, and tree tables were extracted from the North Carolina FIADB. These tables were then joined into a single R dataframe to provide a detailed table of *current* measurements.

To obtain previous FIA measurements, the current record of a remeasurement plot was matched to the record of the previous inventory using the previous plot sequence number. For these previous plot sequence numbers, the same plot, condition, and tree data records were gathered from the FIADB. These tables were also joined into a single R dataframe to provide a detailed table of *historical* measurements.

Finally, the current and historical measurements were joined by matching state, survey unit, county, plot, subplot and species-specific individual tree records. This provided a single R dataframe of individual tree growth data on remeasured FIA plots to establish baseline rates of canopy height change.

* 1. **Statistical Methods Used**

General descriptions for the statistical methods used in this research are provided below. For further information, readers should refer to the references cited herein. Unless otherwise stated, all statistical tests used a significance (alpha) level of 0.05. Analyses were conducted in R Statistical Software ( v4.3.2; R Core Team, 2023) and JMP (*JMP Student Edition*, 1989).

* + 1. **Linear Regression Model**

Linear regression is a statistical method that aims to represent the relationship between a dependent and explanatory variable(s) by fitting a linear equation to the observations (esri, 2025). Linear regression employs an equation following the form:

Where *Y* is the dependent variable, *X* is the explanatory variable, *b* is the slope of the line, and *a* is the intercept (*Y* when *x* = 0). Linear regression is often done using a least squares approach, where the fitted line attempts to minimize the sum of the squared residuals (Department of Statistics, 1997). However, least squares approaches can be highly sensitive to outliers (C. Yu & Yao, 2017), a non-trivial issue in the context of canopy height estimation using LiDAR, which may generate errant returns. The least squares estimation of a regression slope is ultimately a *weighted average* of pairwise slopes, and an extreme slope value between a data point and an outlier will influence the slope estimate (Goldstein-Greenwood, 2023). As such, this research will employ a method of “robust regression”, the Theil-Senregression (Sen, 1968; Theil, 1992). The Theil-Sen regression calculates the slope between a predictor and response variable as the *median* of slopes between each pair of points in the data set, providing a correlation estimate that is more insensitive to outliers.

* + 1. **Equivalence Test**

An insignificant p-value of a null-hypothesis significance test (NHST) simply indicates the absence of evidence of a difference between treatments, which is not the same as stating equivalence between treatments (Altman & Bland, 1995). Failure in rejecting the null hypothesis of an NHST does not automatically make it true, merely that there is insufficient evidence to support the alternate (Wachs, 2015). Put simply, absence of evidence is not evidence of absence.

An equivalence test is a subtype of interval hypothesis testing, which tests against the null hypothesis that differences between group averages are larger than a margin of equivalence (Shtaynberger & Bar, 2023). Equivalence tests are used to show that group averages are equivalent within a margin that is *practically* important, as it is never truly possible to show that an effect size (Δ) is zero (Lakens, 2022). As such, when sharing results of an equivalence test it is common practice to report the equivalence margin the data is tested against.

This research will employ a two one-sided tests (TOST) procedure for equivalence testing. In TOST, an upper (ΔU) and lower (-ΔL) bound of equivalence specify the smallest effect size of interest. TOST utilize two simultaneous null hypotheses (Lakens, 2017), following the formulas:

With this setup, we can reject the presence of meaningful effects only if *both* tests yield p-values below 0.05 (Lakens, 2022), and consider the groups practically equivalent within the margin provided (JMP Support, 2024).

* + 1. **Wilcoxon Rank-Sum Test**

A t-test is a parametric test that evaluates the means of one or two populations. A one sample t-test evaluates if a population’s mean differs from a known value, a two-sample t-test evaluates if two populations differ significantly from each other, and a paired t-test evaluates significant difference of paired measurements (JMP Statistical Discovery, 2025). Given that this research aims to identify changes in forest height between two points in time with sampled measurements, a two-sample t-test is appropriate. However, this research will employ a non-parametric alternative to the two-sample t-test, the Wilcoxon rank-sum test, which tests purely on the order (ranks) of observations from the two samples (Wild & Seber, 2000). The Wilcoxon statistic, W, is the smaller of the rank totals between the two samples (Hole, 2015).

* + 1. **Mann-Kendall Test**

The Mann-Kendall test is a non-parametric test that assesses whether a time series has a monotonic upward or downward trend, without needing data to be normally distributed or linear (Kendall & Gibbons, 1990; Mann, 1945; Meals et al., 2011). The test itself is rank-based, unaffected by presence of extreme values in the dataset (Ringard et al., 2019). This makes it especially useful for time series of biological, chemical, or financial data, as it can identify whether trends of perceived growth or decline are statistically significant despite fluctuations. The test statistic S is calculated with the equation:

This equation determines the sign of the difference between each observation (*yj*) and the previous observation (*yi*), returing a -1, 0, or 1, if the differenceis negative, zero, or positive, respectively (EarthSoft, Inc., 2024). This S value provides the number of increasing occurrences in the dataset, with positive S values indicative of an upward trend over time. The Z value provides a more practical look at the strength of the trend, calculated with:

where S – 1 is used if S > 0, or S – 1 is used if S < 0. Finally, Kendall’s Tau coefficient indicates the correlation of observations and their order in time (S. Chen et al., 2022), with values ranging from -1 (negative correlation) to 1 (positive correlation).

* + 1. **Bonferroni Correction**

In a single hypothesis test with a significance level of 0.05, the probability of a Type I error (false positive) is 5%. However, when running multiple significance tests (a family) the Type I error rate increases considerably. In our case, we are running 15 pairwise equivalence tests at each equivalence bound, increasing the family-wise error rate to 54% (Frost, 2023). The Bonferroni correction controls the likelihood of false positives by dividing the significance level by the number of tests performed (Amplitude, Inc., 2025). This correction determines a new significance level for our multiple hypothesis tests to be evaluated against.

* 1. **Phase 1: Validating ATL08 Canopy Heights with ALS**

Prior to identifying change over time, it is necessary to establish a strong relationship between ATL08 canopy heights and those from reference data at a single point in time. Doing so will allow us to reliably use ATL08 canopy height estimations throughout the remainder of this research. To accomplish this, Phase 1 uses linear regression to compare ATL08 canopy height estimations against ALS-derived canopy heights within the bounds of each ATL08 segment (see Section 2.3.2).

To align with the timing of ALS data, ATL08 segments in the study area were restricted to only those acquired during the summer of 2020, resulting in 5,003 segments. Though the ALS data was collected in the winter months of late 2019 to early 2020, this was deemed to provide an adequate temporal coincidence with the ATL08 data. These segments will serve as a *validation set* to assess the accuracy of the ATL08 product in canopy height estimation.

For the validation set of segments, the correlation between ATL08 and ALS canopy height metrics was assessed with a Theil-Sen regression model using the *RobustLinearReg* package in R (Hurtado, 2020). For the most direct comparison, we compared the 98th percentile ATL08 canopy height (*h\_canopy)* variableto the 98th percentile height of ALS vegetation returns. The ATL08 height was used as the response (Y) variable, while ALS height was used as the predictor (X) variable.

* 1. **Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection**

Using ICESat-2 or GEDI data to track change in forest height over time is challenged by the transect sampling patterns of spaceborne LiDAR (see Figure 2; Mulverhill et al., 2022). ICESat-2 intentionally employs an off-nadir pointing at mid-latitudes to fill gaps between tracks and obtain dense coverage of terrestrial vegetation data, prohibiting repeated measurement of individual forest stands (Markus et al., 2017; Neuenschwander et al., 2023; Neumann et al., 2019b). As a result, each year of ATL08 segment data is a distinct sample of the study area (Figure 7). With this in mind, we refer to each year of ATL08 segments as a ‘samples’ for this phase of the approach. A black background with yellow and purple spots

AI-generated content may be incorrect.

*Figure 7: Separate plots showing spatial distribution of each year’s ATL08 segments throughout the study area, unique colors chosen for each year of segments. Each year of ATL08 segments do not align, but shift spatially to provide greater surface coverage.*

Since spatial differences in forest height may be confounded with change over time, we must address two possible sources of biases. First, we verify that each annual sample represents the overall distribution of forests throughout the entire study area. However, this will only provide confidence in any individual sample. To use the samples together to track change over time, we also verify that the footprints of each sample are comparable with each other. By adequately addressing the bias that may exist within ICESat-2 data collection, we can move across space and throughout time to identify canopy height change, with confidence that trends emerge *even though* the samples are distinct, and not *because* the samples are distinct.

* + 1. **Sample Representation**

We begin by randomly selecting 50,000 pixels from all forests in our study area (Figure 8).

A screenshot of a map

AI-generated content may be incorrect.

*Figure 8: Layout showing A) all forested pixels throughout the study area colored by forest cover type, and B) a random selection of 50,000 forest pixels. Selected pixels have an ATL08 segment created at their center coordinates, resulting in the red speckled effect.*

For each selected pixel, we create an ATL08 segment at its center (see Section 2.3.2) and extract ALS canopy heights within the segment. These combined extractions serve as the *population* of canopy heights across the entire study area.

To establish individual samples as representative, we compare ALS canopy heights within the footprints of each sample (Figure 7) against the population. If each sample’s distribution of ALS canopy heights falls within the overall population distribution, we can proceed with confidence that ICESat-2 collection yields representative samples of our study area.

* + 1. **Sample Location Equivalence**

Next, we establish the locations of the samples as equivalent by comparing their distribution of ALS canopy heights against *each other,* rather than against population. For this we use a bootstrapped TOST technique using the *TOSTER* package in R (Caldwell, 2022; Lakens, 2017), which was robust to possible violations of the standard t-test (Caldwell, 2025).

In this case, the equivalence margin of our TOST corresponds to tolerable difference (in meters) in canopy height between the footprints of samples. These tests were performed across equivalence bound values of 2, 1.5, 1, 0.5, 0.25, and 0.1 meters, spanning a range of interpretations for “equivalence” in canopy height. These equivalence tests in this phase use a p-value that employs the Bonferroni correction technique (see Section 2.4.7), using a p-value significance level of α = *0.05 / 15* or *0.0033*.

Ultimately, these tests aim not to demonstrate perfect equivalence among samples, but rather to identify samples with noticeable divergence from the rest. Samples with poor performance in equivalence testing are removed temporal analyses, under the assumption of spatial differences in forest conditions that would confound change over time.

* 1. **Phase 3: Reference Canopy Height Growth**

Phase 3 determines expected canopy height growth within the study area using FIA remeasurement plots. For this we use *actual* height measurements from FIA plots-rather than *total* height measurements may involve subjective estimation for trees with missing tops-as actual height measurements use the canopy surface sampled by LiDAR data (U.S. Department of Agriculture Forest Service, 2024). Moreover, we use measurements only from live trees by removing dead trees using the FIA tree status code.

Importantly, FIA measurements are collected at the individual-tree level. For this phase we aggregate tree up to the plot-level to align with the spatial scale of ATL08 segments. From each FIA plot, we calculate the median values of increment and net change in canopy height.

Increment represents the annual, year-over-year growth for individual trees calculated as:

Net change scales this annual increment to match the timeframe of our ATL08 segments, as FIA plots have remeasurements periods ranging from 5-10 years (U.S. Department of Agriculture Forest Service, 2023). It is calculated as:

Records with negative net change were removed, possibly resulting from errors in the FIA sampling protocol or damage from disturbance events. We summarize the distributions of increment and net change into minimum, 1st quartile, mean, median, 3rd quartile, and maximum values. We also visualize the distributions of increment and net change with histograms to provide insight into typical annual and cumulative forest growth in the study area. These reference metrics serve not as “targets” to hit when tracking growth with ATL08 segments, but rather a gut-check for additional context.

* 1. **Phase 4: Identifying Canopy Height Growth with ATL08**

Phase 4 uses multiple statistical approaches to identify canopy height growth in ATL08 segment data. Growth trends are stratified across the following three factors:

1. **Land cover type**: Deciduous Forest, Evergreen Forest, Mixed Forest, Woody Wetlands
2. **Disturbance presence:** No, Yes
3. **Time since disturbance:** 3-10 Years, 10-20 Years, 20-30 Years, 30+ Years

First, we use a Wilcoxon Rank Sum test to determine net change in canopy height comparing the first and last years of ATL08 segments. This one-sided test uses an alternative hypothesis that canopy heights in the last year are significantly *greater* than those in the first year, indicative of growth over time. For the Wilcoxon tests we report the W statistic and the difference in mean canopy height between the first and last year of segments.

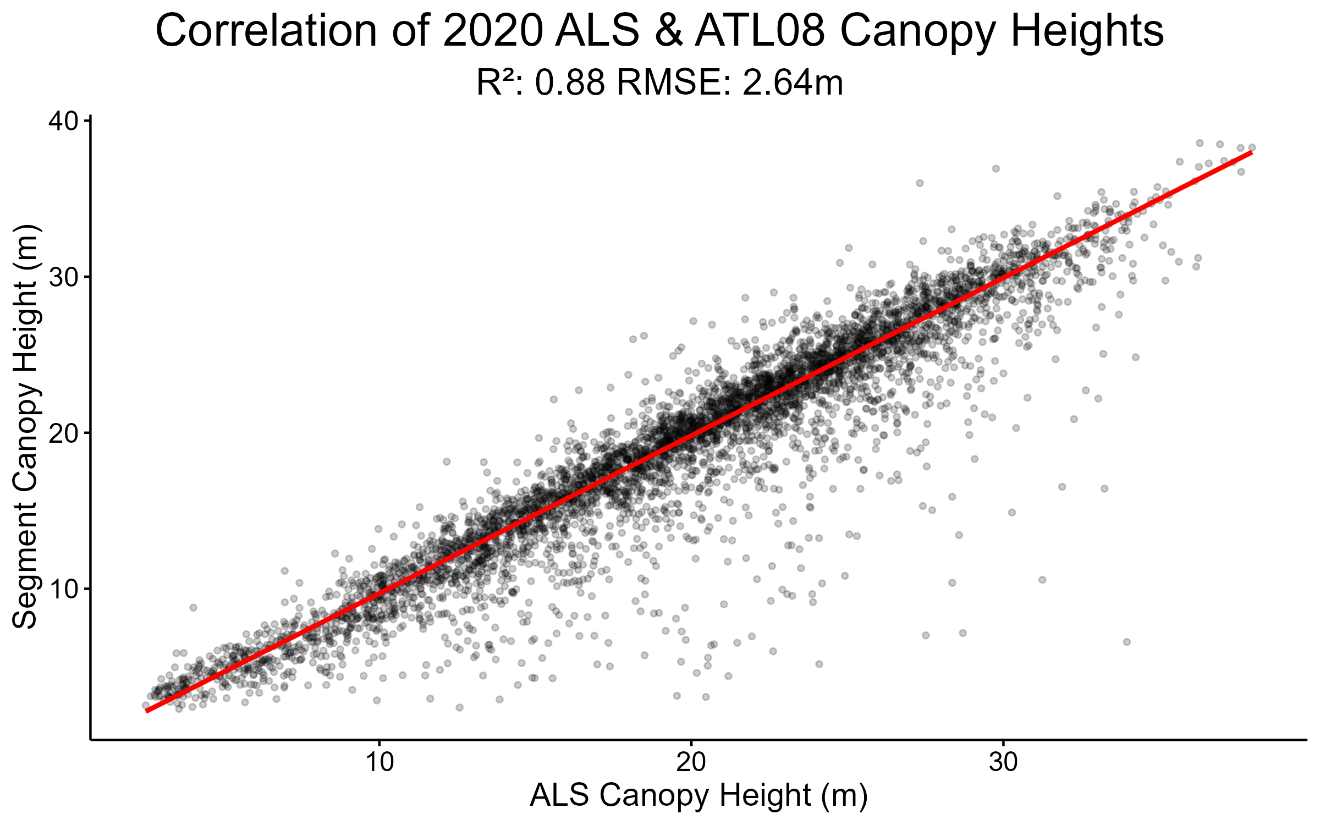
Next, we employ a Theil-Sen regression to estimate incremental change in canopy height over time modeling the response of ATL08 canopy heights (Y) to the year (X). From this model we report the slope, approximating the rate of change from year-to-year, and the intercept, representing the baseline canopy height at the start of the segment timeframe.

Finally, we use a one-sided Mann-Kendall (MK) test was used to identify monotonic trends in canopy height over time. The MK test is also one-sided, using an alternative hypothesis that later values are significantly *greater* than earlier values, suggestive of growth over time. From the MK test results we report the Z, S, and Tau statistics.

Due to the large sample size in our ATL08 segment dataset, tests in Phase 4 were run with a bootstrapping technique. Each test was run for 10,000 iterations on an independent sample of 500 observations taken with replacement. This accommodates the unequal sample sizes between years of ATL08 data, and ensures that any ATL08 segment is an independent pull from the dataset. Additionally, using 10,000 iterations stabilized the variability inherent in the bootstrapping technique. Statistics reported in Phase 4 are the median values of the 10,000 iterations.

1. **Results**
   1. **Phase 1: Validating ATL08 Canopy Heights with ALS**

ATL08 segment canopy height estimations demonstrate strong correlation with reference ALS data at a single point in time (Figure 8). With an R2 of 0.88 and an RMSE of 2.64 meters, ATL08 segments are adequately estimating canopy heights within the study area. The greater dispersion of points below the regression (red) line is indicative of an overall underestimation of canopy height by ATL08 in comparison to ALS data.

**

*Figure 9: Correlation of canopy heights estimated by ALS and ATL08. Canopy heights are reported in meters above ground. Red line indicates trend line from a Theil-Sen regression using ATL08 (Y) and ALS (X) canopy heights*

* 1. **Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection**
     1. **Sample Representation**

When comparing ALS canopy heights in the locations of each sample against ALS canopy heights throughout entire study area, the samples appear representative of the broader population (Figure 10). Each sample’s canopy height distribution falls within the bounds of the overall population distribution, indicated by the overarching white histogram. Moreover, the distributions of the individual samples demonstrate reasonable alignment with each other.

A graph of a height

AI-generated content may be incorrect.

* + 1. **Sample Location Equivalence**

Bootstrapped TOST show that the footprints of samples are practically equivalent within two meters of canopy height (Figure 13), as below this bound not all samples demonstrate equivalence.

A group of blue and yellow squares

AI-generated content may be incorrect.

*Figure 11: Grid of plots showing results of equivalence tests between locations samples, where each subplot represents a unique equivalence bound value, with a TOST run for each group pair. Tests were run using a bootstrapped TOST technique. Blue cells indicate that locations in Sample 1 (X) and Sample 2 (Y) are equivalent within the bound specified in the subplot title. Yellow cells indicate Samples 1 and 2 are not equivalent, as at least one of the null hypotheses in the TOST was not rejected.*

At more strict margins of equivalence such as 0.1 or 0.25, none of our samples demonstrate equivalence in canopy height with each other. Based on the results of the equivalence tests in Figure 11, the 2019 sample demonstrates noticeably lower equivalence than other samples. If the 2019 sample is removed from consideration, all other samples demonstrate equivalence at a margin as tight as 1 meter in canopy height. With this in mind, we remove the 2019 sample from temporal analyses. While this sample may have represented the population, spatial differences in canopy height between this sample and the others could impact trends over time. As a result, we’ve minimized the sampling bias present in our ATL08 segments, but have limited the timeframe for tracking change to only 5 years of data (2020-2024)

* 1. **Phase 3: FIA-Derived Canopy Height Change**

Table 1 shows plot-level summary statistics for increment and net change in canopy height from FIA remeasurement plots.

*Table 1: Summary statistics for total and actual height measurements. Units reported as original FIA measurements were in feet.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **Units** | **Min.** | **1st Qu.** | **Median** | **Mean** | **3rd Qu.** | **Max.** |
| Increment | Meters per Year | 0 | 0.20 | 0.34 | 0.41 | 0.53 | 1.76 |
| Net Change | Meters | 0 | 1.03 | 1.68 | 2.07 | 2.63 | 8.82 |

Figure 11 shows the distribution of actual height increment from remeasurement plots. The median value indicates that across the range of forest types in our study area, one can expect an additional 0.34 meters of tree height growth each year at the plot level.

A graph of blue bars

AI-generated content may be incorrect.

*Figure 12: Distribution of height increment values from FIA remeasurement plots. Red line indicates median value.*

Finally, Figure 12 shows the distribution of values for net change in actual height over five years. This distribution is centered at 1.68 meters with a strong right skew, with values ranging from 0 to 8.82 meters of tree growth in a 5-year period.

A graph of a graph

AI-generated content may be incorrect.

*Figure 13: Distribution of five-year net change in height from FIA remeasurement plots. Red line indicates median value.*

* 1. **Phase 4: Assessing Canopy Height Change with ATL08**

**First net change (wilcoxon rank test)**

Table 7 shows the results of net change in canopy height between 2020 and 2024 using Wilcoxon rank sum tests. All factor levels demonstrated significant net change over time except for forests that experienced disturbance 20-30 years ago. All factors experienced net growth, as indicated by positive mean difference values. Cohen’s d effect sizes are in the small to medium range, from 0.06 in forests with 20-30 Years Since Disturbance to 0.64 in forests with 30+ Years Since Disturbance.

**Then incremental (TSR)**

Linear models show statistically significant positive rates of change for canopy height over time for only five of ten factor levels (Table 6). Linear models estimate the greatest rate of change for canopy heights over time in Woody Wetlands forest cover (0.43 meters/year), followed by Evergreen Forest (0.41 meters/year), Mixed Forest (0.39 meters/year), and finally Deciduous Forest (0.31 meters/year). Disturbed forests show a higher rate of change (0.49 meters/year) than forests that had not experienced disturbance (0.36 meters/year). Rate of change did not increase in tandem with time since disturbance. It was highest in the 3-10 years since disturbance age group with a rate of 0.64 meters/year, lower in the 10-20 years group with a rate of 0.33 meters/year, lower still in the 20-30 years group with a rate of 0.14 meters/year, and higher in the 30+ years group with a rate of 0.55 meters/year.

**Then monotonic (MK)**

Finally, MK tests show statistically significant monotonic trend of canopy heights over time for most factor levels (Table 9). Insignificant trends were shown in undisturbed forests, and forests that had experienced disturbance 10-20 or 20-30 years ago. Tau values range from 0.05 for Woody Wetlands, the lowest agreement between increase of time and canopy heights, to 0.07 for Mixed Forest. Disturbed forests have a slightly higher Tau of 0.06 than undisturbed forests of 0.05. Tau values are highest in recently disturbed (<10 Years) or mature (30+ Years) than forests with moderate time since disturbance.

*Table 9: Results of Mann-Kendall tests for each level of each factor. Bolded and italicized values are statistically significant (p < 0.05).*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factor** | **Level** | **Sample Size** | **Z** | **S** | **Tau** |
| Forest Cover Type | Deciduous Forest | 1337 | ***1.84*** | ***6864*** | ***0.06*** |
| Forest Cover Type | Evergreen Forest | 18609 | ***1.86*** | ***6953*** | ***0.06*** |
| Forest Cover type | Mixed Forest | 5598 | ***2.51*** | ***9367*** | ***0.08*** |
| Forest Cover Type | Woody Wetlands | 25206 | 1.61 | 6015 | 0.05 |
| Disturbance Presence | Yes | 28597 | ***1.98*** | ***7405*** | ***0.06*** |
| Disturbance Presence | No | 22153 | 1.56 | 5818 | 0.05 |
| Years Since Disturbance | <10 Years | 7451 | ***2.62*** | ***9794*** | ***0.08*** |
| Years Since Disturbance | 10-20 Years | 9476 | 1.47 | 5499 | 0.04 |
| Years Since Disturbance | 20-30 Years | 7999 | 1.07 | 3983.5 | 0.03 |
| Years Since Disturbance | 30+ Years | 3671 | ***3.52*** | ***13130.5*** | ***0.11*** |

1. **Discussion**

The ability of ICESat-2 to reliably detect change in forest height throughout time is vital for broad scale forest monitoring, overcoming the temporal limitations of repeat ALS data collection and spatial limitations of field inventories. However, the performance of ATL08 canopy height estimation can vary across multitude of topographic, atmospheric, and ecological conditions. As such, any study utilizing ATL08 canopy height estimations must compare them against reference data sources.

We first demonstrated that ATL08 sufficiently estimates canopy heights within our study area (R2 = 0.88, RMSE = 2.64) using coincident high-resolution ALS data. The minor underestimation against reference ALS is to be expected from the literature (Neuenschwander et al., 2020; Rai et al., 2024; Q. Yu et al., 2024). As posited in Section 2.1, the strong performance of height estimation by ATL08 was likely due to the consistent tree height and regular tree spacing found in forest plantations within our study area. Ultimately, ATL08 canopy height estimations in the study area at a single point in time were deemed adequate for identifying trends throughout time.

Next we assessed the spatial biases in ICESat-2 tracks by using ATL08 segments as spatial groups across which to compare 2020 ALS data. All groups of ALS data demonstrated equivalence within a margin of 2 meters, showing a considerable level of agreement across a study area of over 5 million hectares. When tightening the equivalence margin to 1.5 meters, group A did not demonstrate equivalence with groups C, E or F, indicating some spatial difference in canopy height between those locations. Still, equivalence shown in 75% of groups within a 1.5-meter margin speaks to the forest cover homogeneity across the study area. If group A is removed from consideration, all groups demonstrate equivalence to each other within an even tighter 1-meter margin.

Table 6 shows the count and proportion of segments within each group. Since these segments serve as locations across which we assess canopy height equivalence with ALS, it is possible that group A segments are not entirely representative of the study area, comprising less than 6 percent of our working dataset.

*Table 10: Number and proportion of ATL08 segments by group.*

|  |  |  |
| --- | --- | --- |
| **Group** | **Number of Segments** | **Proportion of Total (%)** |
| A | 3,155 | 5.85 |
| B | 5,026 | 9.32 |
| C | 11,661 | 21.63 |
| D | 14,904 | 27.65 |
| E | 6,848 | 12.70 |
| F | 12,311 | 22.84 |

The spatial distribution of each group’s ATL08 segments shown in Figure 13 provides additional context for the sample equivalence tests. Ideally, each year’s segments should cover a similar vertical and horizontal gradient across the study area. Coverage by group A is the lowest, as to be expected from the sample size, with notable absence in the Western regions of the study area. It is possible these areas contain forest conditions that are key for a representative sample.

A purple and yellow lights

AI-generated content may be incorrect.

*Figure 14: Separate plots showing each spatial distribution of each group’s ATL08 segments throughout the study area, unique colors chosen for each year of segments.*

Pursuing this idea further, the Environmental Protection Agency, in collaboration with other federal, state, and international resource management agencies, have divided the United States into regions of similar ecosystems, or *Ecoregions*. Broadly speaking, Ecoregions are measurement units of reasonably consistent geology, vegetation, climate, wildlife, etc. These Ecoregions are available at multiple scales of granularity, from the generalized Level I Ecoregions to the detailed Level 4 Ecoregions. Figure 14 shows the breakdown of each year’s ATL08 segments into the Level 2 Ecoregions found within our study area.

A graph of a number of individuals

AI-generated content may be incorrect.

*Figure 15: Stacked bar plot showing proportion of ATL08 segments in each EPA Level 2 Ecoregions per year of data. Percent values are relative to each year’s total sample size*

The sample equivalence tests identify the group A canopy heights as most different from the others. As shown in Figure 17, the group A has the distribution which is most heavily skewed from a 50:50 split between Level 2 Ecoregions, and the lowest proportion of data collected in the Southern USA Plains. This highlights the importance of ecological continuity, implying that a more even distribution between Level 2 Ecoregions could’ve yielded equivalence between A and the other groups at a tighter margin. When considering spatial scales best suited for forest monitoring with ATL08 data, this presents EPA ecoregions as a possible recommendation.

The equivalence of ALS groups demonstrated in Phase 2 was deemed sufficient for our purposes, especially considering the RMSE of 2.64 meters in Phase 1, as it is unlikely that ATL08 measurements would be heavily influenced by spatial biases in canopy height falling within this margin of error.

Phase 3 established reference canopy height changes within our study area from FIA remeasurement plots. The distribution of actual height increment (Figure 14) is heavily concentrated in values below half a meters per year, possibly indicative of predominantly young forests across our study area, as forest height growth tends to decrease after forest stand maturity (Bennett, 1963; Tang et al., 2014). Total height net change (2.69 m/year) and actual height net change (2.70 m/year) are highly similar, indicating minimal FIA measurement error from subjective estimation when trees have broken or missing tops (U.S. Department of Agriculture Forest Service, 2024).

Dead or retired trees were removed during FIA data processing to focus solely on living-tree growth. While ATL08 data cannot differentiate between live or dead standing trees-and one can argue for retaining all standing trees to more holistically represent the forest conditions surveyed by LiDAR-including dead or retired trees in FIA analysis would have lowered values of both height change and increment. Ultimately, we removed these trees from FIA remeasurement plots to select the best subset to demonstrate forest height growth, just as the heavy filtering of ATL08 data aimed to achieve.

Phase 4 combined parametric and non-parametric approaches to determine if trends in canopy height change emerged across 6 years (2019-2024) of ATL08 data. The rates of change, indicated by linear regression slopes, were highest in plantation-like conditions (Evergreen Forest and Woody Wetlands). Growth estimates in Mixed Forests were most in line with reference FIA data, speaking to the breadth of spatial coverage by the FIA program. Limiting FIA remeasurement plots to plantation-like forest types would have resulted in greater values in both height change and increment, and closer alignment with the linear regression slope or t-test mean difference of Evergreen Forest or Woody Wetlands.

Disturbed and undisturbed forests show highly similar patterns of canopy height change, with comparable values for the linear regression slope, t-test Cohen’s D, and MK Z statistic, and identical MK tau values. A lower t-test mean difference in disturbed forests than undisturbed forests is likely due to an amplified underestimation of canopy height in sparser stands (Figure 15), as a sparser canopy has a lower probability of generating returns throughout the canopy profile. In comparison, a denser canopy with a more homogenous top surface (present in undisturbed forests, or plantations with ample recovery from disturbance) will yield a more consistent canopy height.

A screenshot of a video game

AI-generated content may be incorrect.

*Figure 16: Diagram showing canopy surface (black line) in disturbed (left) vs. undisturbed (right) forests. Due to the canopy openness, photons returned to the ATLAS sensor (red dots) are more likely to underestimate true 98th percentile canopy height in a disturbed forest. This also applies to mixed forests with greater tree heterogeneity, where the probability of accurately representing 98th percentile canopy height is far lower than in a plantation with more consistent tree heights.*

Interestingly, the only factor level demonstrating statistically significant non-linear smoothing from the GAMs was disturbed forests. This implies that even within a 6-year timeframe ATL08 data may be picking up non-linear patterns of forest recovery from disturbance. However, the GAM results were ineffective in demonstrating nonlinear canopy height change for all other factor levels, as none of the disturbance age groups themselves demonstrated statistically significant non-linear smoothing. It is possible that these groups may be too broad in their division of years. In consideration of the lackluster results from the GAMs, it is likely that the non-linear patterns of forest height growth for any one given stand smoothed out to more linear patterns over the extent of our study area. Though focusing on smaller regions with more consistent stand age would yield a lower sample size, it could reveal more nonlinear patterns of canopy height change in ATL08 data.

The two sample t-tests and MK tests show that both net and monotonic trends were emergent. This was not applicable only to Deciduous Forest, which only showed statistical significance in net change. While these forests undoubtedly experienced growth, it is likely that the heterogenous canopy presented challenges for the ATL08 algorithm to detect year-over-year change.

It should be noted the undeniable role of the study area in influencing the results, which was specifically chosen for its gentle topographic relief and known presence of forest plantations. In mountainous regions of greater terrain variability or tropical regions of greater canopy complexity, the ATL08 algorithm will struggle to identify ground photons (J. Dong et al., 2021; Fernandez-Diaz et al., 2022), likely failing to yield sensible results for canopy height change over time. Moreover, our study area is located within the loblolly pine range-a region of exceptional forest productivity (Ribas-Costa et al., 2024). This region undoubtedly provides canopy height changes greater than that to be expected in other U.S. forests, where canopy height change over 6-years may not reach a level detectable in ATL08 data. As such, we must note that users should strongly consider the dominant topography, forest conditions, and forest productivity of their study area if attempting to identify canopy height change with ATL08 data.

As a final assessment, Phase 5 removed segments demonstrating spatial bias in canopy height from our working dataset. Though this limited our dataset to only 5 years (2020-2024), these segments were collected across locations with canopy heights that are equivalent within a margin of 1m from 2020 ALS (Figure 11). With this adjusted dataset, significant net canopy height change was still emergent with exception of forests that experienced disturbance 20-30 years ago. Linear and monotonic trends in canopy height change were less prevalent, with inconsistent significance across factor levels in the regression and MK test.

From these new results we surmise that some, not all, of the canopy height change originally seen in Phase 4 was driven by spatial biases embedded in 2019 segments. To reiterate, when 2019 data is included, the canopy heights across track locations were only equivalent within a range of 2 meters as determined by ALS, but the trends of canopy height change are more pronounced. When 2019 data is removed, the canopy heights across track locations are equivalent within a range of 1 meter as determined by ALS, but trends were muffled by the shorter time frame and lower sample size.

Testing equivalence between treatments requires careful definition of a margin of “practical” equivalence-a range within treatment results are close enough to be considered the same in practice. In this research, this margin is the maximum difference in canopy height (in meters) across ICESat-2 track locations we would tolerate when exchanging space for time. Since canopy height is emergent from multiple ecological factors and canopy height observations are limited by the precision of the measurement instrument, we found it appropriate to test equivalence across a range of equivalence bound values.

Other studies have used a flexible percentage-based approach for paired measurements, deeming a region within +25% of the slope and intercept means as practically equivalent (Corrao et al., 2022; Falkowski et al., 2008). While this research does not employ paired measurements for equivalence testing, a percentage-based approach for equivalence testing could accommodate the range of forest cover types sampled by ICESat-2’s global coverage. A recent study by Guo et al. (2025) found equivalence for tree height measurements between instruments at a tolerance above 1.5m in trees with a “pyramidal and well-defined crown”-similar to that of pine plantations in our study area. While their study focused on measurement errors arising from field crew experience, an argument could be made that the numerous sources of error in ATL08 data present similar inconsistencies, despite our attempts to control it in this study.

Ultimately, trends emergent from analyses of multitemporal ATL08 data always be influenced by *some* spatial differences in forest height, purely due to the nature of ICESat-2’s sampling pattern. In absence of a widely accepted standard, researchers may derive their own definition of canopy height equivalence to properly balance the duration over which they’re assessing change, the amount and spatial coverage of ATL08 data available, and the noise of ATL08 canopy height estimates within their study area.

In revisiting our initial research questions, we find that ATL08 data can detect both net and incremental change in canopy height even within a period of only 5 years. However, no clear pattern emerged when stratifying by forest cover type and disturbance history, as these factors are likely convolved within each other.

1. **Conclusion**

Land management decisions at a local scale affect biodiversity and climatic processes, and contribute collectively to global climate change (Griffith et al., 2003). Understanding regional differences in environmental resource use and degradation is therefore critical to ensure effective land management that is locale-specific (Omernik & Griffith, 2014). When considering environmental scales best suited for ICESat-2 analyses, one must choose a spatial scale that will maximize data availability, while maintaining environmental continuity for optimal canopy height estimation by the ATL08 algorithm. In consideration of spatial biases in canopy height presented in the discussion, future research could stratify ATL08 data by EPA Ecoregions to explore canopy height changes.

Based on the results of Phase 4, an argument could be made that ICESat-2 holds the potential to track changes in forest height throughout time. However, this ability will be reliant upon several factors, not only those influencing ATL08 canopy height estimations at a single point in time (Malambo & Popescu, 2021; Rai et al., 2024), but their amplified effects throughout time. Additionally, land change patterns over a given area will hold strong influence on the ATL08 algorithm’s ability to interpolate the ground surface, possibly obscuring valid canopy height change.

This study applied several quality filters to ATL08 data that were non-trivial in their reduction of the sample size, controlling for beam strength, time of collection, and seasonality to select the best subset of ATL08 data. While strong beam, nighttime data collected in leaf off conditions have historically provided the highest accuracy in static canopy height estimation, future studies should maintain all ATL08 segments within a study area to test the emergence of temporal trends *across* factors of ATL08 data and environmental conditions. This would extend the work of Rai et. al (2024), A. Liu et al. (2021), Malambo & Popescu (2021), Neuenschwander et al. (2020), and others across multiple timesteps.

Spatial biases may have arisen in Phase 2 from post-Hurricane Florence recovery of the forests in our study area, as the occurrence of Hurricane Florence (September 2018) aligns closely with the beginning of ICESat-2 data collection (December 2018). Further work is required to dig into the spatial stratification of the segments using a Hurricane Florence flood extent product (NC OneMap, 2020), which could determine if Hurricane Florence inundation is impacting the canopy height values reported by the 2020 ALS.

Ultimately, this research aims to understand structural changes of vegetation throughout time using modern active remote sensing missions. In understanding these changes at the regional scale, this research can aid governing agencies and local stakeholders with spatially explicit management strategies. Such strategies can protect vegetation and natural resources against site-specific threats, supporting conservation efforts and land-use planning in the face of a changing climate.

1. **Bibliography**

Altman, D. G., & Bland, J. M. (1995). Statistics notes: Absence of evidence is not evidence of absence. *BMJ*, *311*(7003), 485–485. https://doi.org/10.1136/bmj.311.7003.485

Amplitude, Inc. (2025). *Bonferroni Correction Explained: Managing Multiple Testing in Statistics*. https://amplitude.com/explore/experiment/what-is-bonferroni-correction

Andersen, H.-E., Reutebuch, S. E., & McGaughey, R. J. (2006). A rigorous assessment of tree height measurements obtained using airborne lidar and conventional field methods. *Canadian Journal of Remote Sensing*, *32*(5), 355–366. https://doi.org/10.5589/m06-030

Atkins, J. W., Costanza, J., Dahlin, K. M., Dannenberg, M. P., Elmore, A. J., Fitzpatrick, M. C., Hakkenberg, C. R., Hardiman, B. S., Kamoske, A., LaRue, E. A., Silva, C. A., Stovall, A. E. L., & Tielens, E. K. (2023). Scale dependency of lidar‐derived forest structural diversity. *Methods in Ecology and Evolution*, *14*(2), 708–723. https://doi.org/10.1111/2041-210X.14040

Baker, J. B., & Langdon, G. O. (1990). Pinus Taeda L. Loblolly Pine. In *Silvics of North America* (Vol. 1, pp. 497–512). U.S. Dept. of Agriculture, Forest Service.

Beland, M., Parker, G., Sparrow, B., Harding, D., Chasmer, L., Phinn, S., Antonarakis, A., & Strahler, A. (2019). On promoting the use of lidar systems in forest ecosystem research. *Forest Ecology and Management*, *450*, 117484. https://doi.org/10.1016/j.foreco.2019.117484

Bennett, F. (1963). *Gorwth and Yield of Slash Pine Plantations* (No. SE-1). U.S. Department of Agriculture, Forest Service Southern Forest Experiment Station.

Bohn, F. J., & Huth, A. (2017). The importance of forest structure to biodiversity–productivity relationships. *Royal Society Open Science*, *4*(1), 160521. https://doi.org/10.1098/rsos.160521

Butler, H., Andrew Bell, Michael P. Gerlek, chambbj, Pete Gadomski, Connor Manning, Mateusz Łoskot, Bas Couwenberg, Norman Barker, Paul Ramsey, Julia Dark, Nicolas Chaulet, Even Rouault, Kyle Mann, Claire Foster, Guilhem Villemin, Michael Rosen, Grigory, Ogi Moore, … Aaron Yonas. (2024). *PDAL/PDAL: 2.7.1* (Version 2.7.1) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.10884408

Caldwell, A. R. (2022). *Exploring Equivalence Testing with the Updated TOSTER R Package*. PsyArXiv. https://doi.org/10.31234/osf.io/ty8de

Caldwell, A. R. (2025). *Robust TOST Procedures* [Vigenette]. https://cran.r-project.org/web/packages/TOSTER/vignettes/robustTOST.html

Callaghan, J. (2020). Extreme rainfall and flooding from Hurricane Florence. *Tropical Cyclone Research and Review*, *9*(3), 172–177. https://doi.org/10.1016/j.tcrr.2020.07.002

Cao, L., Coops, N. C., Innes, J. L., Sheppard, S. R. J., Fu, L., Ruan, H., & She, G. (2016). Estimation of forest biomass dynamics in subtropical forests using multi-temporal airborne LiDAR data. *Remote Sensing of Environment*, *178*, 158–171. https://doi.org/10.1016/j.rse.2016.03.012

Carabajal, C. C., & Boy, J.-P. (2020). ICESAT-2 ALTIMETRY AS GEODETIC CONTROL. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B3-2020*, 1299–1306. https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1299-2020

Chastain, R., Housman, I., Goldstein, J., Finco, M., & Tenneson, K. (2019). Empirical cross sensor comparison of Sentinel-2A and 2B MSI, Landsat-8 OLI, and Landsat-7 ETM+ top of atmosphere spectral characteristics over the conterminous United States. *Remote Sensing of Environment*, *221*, 274–285. https://doi.org/10.1016/j.rse.2018.11.012

Chen, F., Zhang, X., Wang, L., Du, B., Dang, S., & Wang, L. (2023). Systematic Evaluation of Multi-Resolution ICESat-2 Canopy Height Data: A Case Study of the Taranaki Region. *Remote Sensing*, *15*(24), 5686. https://doi.org/10.3390/rs15245686

Chen, S., Ghadami, A., & Epureanu, B. I. (2022). Practical guide to using Kendall’s *τ* in the context of forecasting critical transitions. *Royal Society Open Science*, *9*(7), 211346. https://doi.org/10.1098/rsos.211346

Columbia Environmental Research Center. (2019). *2019 Disaster Relief Act: USGS recovery activities* (USGS Numbered Series Nos. 2019–3066; Fact Sheet, p. 4). U.S. Geological Survey.

Coops, N. C., Tompalski, P., Goodbody, T. R. H., Queinnec, M., Luther, J. E., Bolton, D. K., White, J. C., Wulder, M. A., Van Lier, O. R., & Hermosilla, T. (2021). Modelling lidar-derived estimates of forest attributes over space and time: A review of approaches and future trends. *Remote Sensing of Environment*, *260*, 112477. https://doi.org/10.1016/j.rse.2021.112477

Corrao, M. V., Sparks, A. M., & Smith, A. M. S. (2022). A Conventional Cruise and Felled-Tree Validation of Individual Tree Diameter, Height and Volume Derived from Airborne Laser Scanning Data of a Loblolly Pine (P. taeda) Stand in Eastern Texas. *Remote Sensing*, *14*(11), 2567. https://doi.org/10.3390/rs14112567

Department of Statistics. (1997). *Linear Regression* (No. 101; Course List for 1997-98). Yale University. http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

Dong, J., Ni, W., Zhang, Z., Sun, G., 中国科学院空天信息创新研究院 遥感科学国家重点实验室, 北京 100094 Aerospace Information Research Institute, Chinese Academy of Sciences, State Key Laboratory of Remote Sensing Science, Beijing 100094, China, 中国科学院大学 资源与环境学院, 北京 100049 School of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China, & 美国马里兰大学地理系 马里兰州College Park, 美国 20742 Department of Geographical Sciences, University of Maryland, College Park, Maryland, USA, 20742. (2021). Performance of ICESat-2 ATL08 product on the estimation of forest height by referencing to small footprint LiDAR data. *National Remote Sensing Bulletin*, *25*(6), 1294–1307. https://doi.org/10.11834/jrs.20219449

Dong, P., & Chen, Q. (2017). *LiDAR Remote Sensing and Applications* (1st ed.). CRC Press. https://doi.org/10.4324/9781351233354

Dubayah, R., Armston, J., Healey, S. P., Bruening, J. M., Patterson, P. L., Kellner, J. R., Duncanson, L., Saarela, S., Ståhl, G., Yang, Z., Tang, H., Blair, J. B., Fatoyinbo, L., Goetz, S., Hancock, S., Hansen, M., Hofton, M., Hurtt, G., & Luthcke, S. (2022). GEDI launches a new era of biomass inference from space. *Environmental Research Letters*, *17*(9), 095001. https://doi.org/10.1088/1748-9326/ac8694

Dubayah, R., Blair, J. B., Goetz, S., Fatoyinbo, L., Hansen, M., Healey, S., Hofton, M., Hurtt, G., Kellner, J., Luthcke, S., Armston, J., Tang, H., Duncanson, L., Hancock, S., Jantz, P., Marselis, S., Patterson, P. L., Qi, W., & Silva, C. (2020). The Global Ecosystem Dynamics Investigation: High-resolution laser ranging of the Earth’s forests and topography. *Science of Remote Sensing*, *1*, 100002. https://doi.org/10.1016/j.srs.2020.100002

Duncanson, L., Neuenschwander, A., Hancock, S., Thomas, N., Fatoyinbo, T., Simard, M., Silva, C. A., Armston, J., Luthcke, S. B., Hofton, M., Kellner, J. R., & Dubayah, R. (2020). Biomass estimation from simulated GEDI, ICESat-2 and NISAR across environmental gradients in Sonoma County, California. *Remote Sensing of Environment*, *242*, 111779. https://doi.org/10.1016/j.rse.2020.111779

EarthSoft, Inc. (2024, July 8). *Mann-Kendall Tests*. EQuIS 7.24.3. https://help.earthsoft.com/lib\_mann-kendall.htm

esri. (2025, April 18). How Linear regression algorithm works. *Feature and Tabular Analysis Toolset Concepts*. https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-linear-regression-works.htm

Falkowski, M. J., Smith, A. M. S., Gessler, P. E., Hudak, A. T., Vierling, L. A., & Evans, J. S. (2008). The influence of conifer forest canopy cover on the accuracy of two individual tree measurement algorithms using lidar data. *Canadian Journal of Remote Sensing*, *34*(sup2), S338–S350. https://doi.org/10.5589/m08-055

FAO and UNEP. (2020). *The State of the World’s Forests 2020*. FAO and UNEP. https://doi.org/10.4060/ca8642en

Feng, T., Duncanson, L., Montesano, P., Hancock, S., Minor, D., Guenther, E., & Neuenschwander, A. (2023). A systematic evaluation of multi-resolution ICESat-2 ATL08 terrain and canopy heights in boreal forests. *Remote Sensing of Environment*, *291*, 113570. https://doi.org/10.1016/j.rse.2023.113570

Fernandez-Diaz, J. C., Velikova, M., & Glennie, C. L. (2022). Validation of ICESat-2 ATL08 Terrain and Canopy Height Retrievals in Tropical Mesoamerican Forests. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 2956–2970. https://doi.org/10.1109/JSTARS.2022.3163208

Foga, S., Scaramuzza, P. L., Guo, S., Zhu, Z., Dilley, R. D., Beckmann, T., Schmidt, G. L., Dwyer, J. L., Joseph Hughes, M., & Laue, B. (2017). Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sensing of Environment*, *194*, 379–390. https://doi.org/10.1016/j.rse.2017.03.026

Frost, J. (2023, July 17). What is the Bonferroni Correction and How to Use It. *Hypothesis Testing*. https://statisticsbyjim.com/hypothesis-testing/bonferroni-correction/

Goldstein-Greenwood, J. (2023, April 28). Theil-Sen Regression: Programming and Understanding an Outlier-Resistant Alternative to Least Squares. *UVA Library StatLab*. https://library.virginia.edu/data/articles/theil-sen-regression-programming-and-understanding-an-outlier-resistant-alternative-to-least-squares

Gong, P., Li, Z., Huang, H., Sun, G., & Wang, L. (2011). ICESat GLAS Data for Urban Environment Monitoring. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(3), 1158–1172. https://doi.org/10.1109/TGRS.2010.2070514

Griffin, M., Malsick, M., Mizzell, H., & Moore, L. (2020). Historic Rainfall and Record-Breaking Flooding from Hurricane Florence in the Pee Dee Watershed. *The Journal of South Carolina Water Resources*, *6*, 28–35. https://doi.org/10.34068/JSCWR.06.03

Griffith, J. A., Stehman, S. V., & Loveland, T. R. (2003). Landscape Trends in Mid-Atlantic and Southeastern United States Ecoregions. *Environmental Management*, *32*(5), 572–588. https://doi.org/10.1007/s00267-003-0078-2

Guerra-Hernández, J., Arellano-Pérez, S., González-Ferreiro, E., Pascual, A., Sandoval Altelarrea, V., Ruiz-González, A. D., & Álvarez-González, J. G. (2021). Developing a site index model for P. Pinaster stands in NW Spain by combining bi-temporal ALS data and environmental data. *Forest Ecology and Management*, *481*, 118690. https://doi.org/10.1016/j.foreco.2020.118690

Guerra-Hernández, J., Narine, L. L., Pascual, A., Gonzalez-Ferreiro, E., Botequim, B., Malambo, L., Neuenschwander, A., Popescu, S. C., & Godinho, S. (2022). Aboveground biomass mapping by integrating ICESat-2, SENTINEL-1, SENTINEL-2, ALOS2/PALSAR2, and topographic information in Mediterranean forests. *GIScience & Remote Sensing*, *59*(1), 1509–1533. https://doi.org/10.1080/15481603.2022.2115599

Hall, F. G., Bergen, K., Blair, J. B., Dubayah, R., Houghton, R., Hurtt, G., Kellndorfer, J., Lefsky, M., Ranson, J., Saatchi, S., Shugart, H. H., & Wickland, D. (2011). Characterizing 3D vegetation structure from space: Mission requirements. *Remote Sensing of Environment*, *115*(11), 2753–2775. https://doi.org/10.1016/j.rse.2011.01.024

Hancock, S., McGrath, C., Lowe, C., Davenport, I., & Woodhouse, I. (2021). Requirements for a global lidar system: Spaceborne lidar with wall-to-wall coverage. *Royal Society Open Science*, *8*(12), 211166. https://doi.org/10.1098/rsos.211166

Harrap, R., & Lato, M. (2010, June 1). *An Overview of LIDAR: collection to applications*. Norwegian Geotechnical Institute.

Harris, N., & Gibbs, D. (2021). *Forests Absorb Twice As Much Carbon As They Emit Each Year*. https://www.wri.org/insights/forests-absorb-twice-much-carbon-they-emit-each-year

Hobu, Inc. (2025). *USGS 3DEP LiDAR Point Clouds* [S3 Bucket; Entwine Point Tiles]. https://registry.opendata.aws/usgs-lidar

Hole, G. (2015). The Wilcoxon test. In *Research Skills One* (1.0).

Homer, C., Dewitz, J., Jin, S., Xian, G., Costello, C., Danielson, P., Gass, L., Funk, M., Wickham, J., Stehman, S., Auch, R., & Riitters, K. (2020). Conterminous United States land cover change patterns 2001–2016 from the 2016 National Land Cover Database. *ISPRS Journal of Photogrammetry and Remote Sensing*, *162*, 184–199. https://doi.org/10.1016/j.isprsjprs.2020.02.019

Housman, I., Campbell, L., Heyer, J., Goetz, W., Finco, M., Megown, K., & Pugh, N. (2021). US Forest Service landscape change monitoring system methods. *USDA General Technology and Applications Center, Salt Lake City, UT*.

Hui, G., Zhang, G., Zhao, Z., & Yang, A. (2019). Methods of Forest Structure Research: A Review. *Current Forestry Reports*, *5*(3), 142–154. https://doi.org/10.1007/s40725-019-00090-7

Hurtado, S. (2020). *RobustLinearReg: Robust Linear Regressions* (p. 1.2.0) [Dataset]. https://doi.org/10.32614/CRAN.package.RobustLinearReg

Jaboyedoff, M., Oppikofer, T., Abellán, A., Derron, M.-H., Loye, A., Metzger, R., & Pedrazzini, A. (2012). Use of LIDAR in landslide investigations: A review. *Natural Hazards*, *61*(1), 5–28. https://doi.org/10.1007/s11069-010-9634-2

Jarron, L. R., Coops, N. C., MacKenzie, W. H., Tompalski, P., & Dykstra, P. (2020). Detection of sub-canopy forest structure using airborne LiDAR. *Remote Sensing of Environment*, *244*, 111770. https://doi.org/10.1016/j.rse.2020.111770

JMP Statistical Discovery. (2025). The t-test. *Statistics Knowledge Portal*. https://www.jmp.com/en/statistics-knowledge-portal/t-test

*JMP Student Edition* (Version 18.2.0). (1989). [Computer software]. SAS Institute Inc.

JMP Support. (2024). *Equivalence Test Reports*. JMP Statistical Discovery LLC. https://www.jmp.com/support/help/en/18.1/index.shtml#page/jmp/equivalence-test-reports.shtml

Jordahl, K., Bossche, J. V. D., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., Maxalbert, Bilogur, A., … Leblanc, F. (2020). *geopandas/geopandas: V0.8.1* (Version v0.8.1) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.3946761

Karjalainen, E., Sarjala, T., & Raitio, H. (2009). Promoting human health through forests: Overview and major challenges. *Environmental Health and Preventive Medicine*, *15*(1), 1–8. https://doi.org/10.1007/s12199-008-0069-2

Kendall, M. G., & Gibbons, J. D. (1990). *Rank correlation methods* (5th ed). E. Arnold ; Oxford University Press.

Kennedy, R. E., Yang, Z., & Cohen, W. B. (2010). Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr — Temporal segmentation algorithms. *Remote Sensing of Environment*, *114*(12), 2897–2910. https://doi.org/10.1016/j.rse.2010.07.008

Kennedy, R. E., Yang, Z., Gorelick, N., Braaten, J., Cavalcante, L., Cohen, W. B., & Healey, S. (2018). Implementation of the LandTrendr Algorithm on Google Earth Engine. *Remote Sensing*, *10*(5), 691. https://doi.org/10.3390/rs10050691

Khaine, I., & Woo, S. Y. (2015). An overview of interrelationship between climate change and forests. *Forest Science and Technology*, *11*(1), 11–18. https://doi.org/10.1080/21580103.2014.932718

Lakens, D. (2017). Equivalence Tests: A Practical Primer for *t* Tests, Correlations, and Meta-Analyses. *Social Psychological and Personality Science*, *8*(4), 355–362. https://doi.org/10.1177/1948550617697177

Lakens, D. (2022). *Improving Your Statistical Inferences*. https://doi.org/10.5281/ZENODO.6409077

Lamping, J. E., Zald, H. S. J., Madurapperuma, B. D., & Graham, J. (2021). Comparison of Low-Cost Commercial Unpiloted Digital Aerial Photogrammetry to Airborne Laser Scanning across Multiple Forest Types in California, USA. *Remote Sensing*, *13*(21), 4292. https://doi.org/10.3390/rs13214292

LaRue, E. A., Fahey, R., Fuson, T. L., Foster, J. R., Matthes, J. H., Krause, K., & Hardiman, B. S. (2022). Evaluating the sensitivity of forest structural diversity characterization to LiDAR point density. *Ecosphere*, *13*(9), e4209. https://doi.org/10.1002/ecs2.4209

Li, B., Zhao, T., Su, X., Fan, G., Zhang, W., Deng, Z., & Yu, Y. (2022). Correction of Terrain Effects on Forest Canopy Height Estimation Using ICESat-2 and High Spatial Resolution Images. *Remote Sensing*, *14*(18), 4453. https://doi.org/10.3390/rs14184453

Li, W., Niu, Z., Shang, R., Qin, Y., Wang, L., & Chen, H. (2020). High-resolution mapping of forest canopy height using machine learning by coupling ICESat-2 LiDAR with Sentinel-1, Sentinel-2 and Landsat-8 data. *International Journal of Applied Earth Observation and Geoinformation*, *92*, 102163. https://doi.org/10.1016/j.jag.2020.102163

Liu, A., Cheng, X., & Chen, Z. (2021). Performance evaluation of GEDI and ICESat-2 laser altimeter data for terrain and canopy height retrievals. *Remote Sensing of Environment*, *264*, 112571. https://doi.org/10.1016/j.rse.2021.112571

Liu, M., Popescu, S., & Malambo, L. (2019). Feasibility of Burned Area Mapping Based on ICESAT−2 Photon Counting Data. *Remote Sensing*, *12*(1), 24. https://doi.org/10.3390/rs12010024

Liu, X., Feng, Y., Hu, T., Luo, Y., Zhao, X., Wu, J., Maeda, E. E., Ju, W., Liu, L., Guo, Q., & Su, Y. (2024). Enhancing ecosystem productivity and stability with increasing canopy structural complexity in global forests. *Science Advances*, *10*(20), eadl1947. https://doi.org/10.1126/sciadv.adl1947

Luthcke, S. B., Thomas, T. C., Pennington, T. A., Rebold, T. W., Nicholas, J. B., Rowlands, D. D., Gardner, A. S., & Bae, S. (2021). ICESat‐2 Pointing Calibration and Geolocation Performance. *Earth and Space Science*, *8*(3), e2020EA001494. https://doi.org/10.1029/2020EA001494

Malambo, L., & Popescu, S. (2024). Mapping vegetation canopy height across the contiguous United States using ICESat-2 and ancillary datasets. *Remote Sensing of Environment*, *309*, 114226. https://doi.org/10.1016/j.rse.2024.114226

Malambo, L., & Popescu, S. C. (2021). Assessing the agreement of ICESat-2 terrain and canopy height with airborne lidar over US ecozones. *Remote Sensing of Environment*, *266*, 112711. https://doi.org/10.1016/j.rse.2021.112711

Maltamo, M., Næsset, E., & Vauhkonen, J. (Eds.). (2014). *Forestry Applications of Airborne Laser Scanning: Concepts and Case Studies* (Vol. 27). Springer Netherlands. https://doi.org/10.1007/978-94-017-8663-8

Mann, H. B. (1945). Nonparametric Tests Against Trend. *Econometrica*, *13*(3), 245. https://doi.org/10.2307/1907187

Markus, T., Neumann, T., Martino, A., Abdalati, W., Brunt, K., Csatho, B., Farrell, S., Fricker, H., Gardner, A., Harding, D., Jasinski, M., Kwok, R., Magruder, L., Lubin, D., Luthcke, S., Morison, J., Nelson, R., Neuenschwander, A., Palm, S., … Zwally, J. (2017). The Ice, Cloud, and land Elevation Satellite-2 (ICESat-2): Science requirements, concept, and implementation. *Remote Sensing of Environment*, *190*, 260–273. https://doi.org/10.1016/j.rse.2016.12.029

Meals, D., Spooner, J., Dressing, S., & Harcum, J. (2011). *Statistical Analysis for Monotonic Trends* (No. 6; Tech Notes, p. 23). U.S. Environmental Protection Agency. https://www.epa.gov/sites/default/files/2016-05/documents/tech\_notes\_6\_dec2013\_trend.pdf

Meyer, V., Saatchi, S. S., Chave, J., Dalling, J. W., Bohlman, S., Fricker, G. A., Robinson, C., Neumann, M., & Hubbell, S. (2013). Detecting tropical forest biomass dynamics from repeated airborne lidar measurements. *Biogeosciences*, *10*(8), 5421–5438. https://doi.org/10.5194/bg-10-5421-2013

Mielcarek, M., Kamińska, A., & Stereńczak, K. (2020). Digital Aerial Photogrammetry (DAP) and Airborne Laser Scanning (ALS) as Sources of Information about Tree Height: Comparisons of the Accuracy of Remote Sensing Methods for Tree Height Estimation. *Remote Sensing*, *12*(11), 1808. https://doi.org/10.3390/rs12111808

Mitchell, J. C., Kashian, D. M., Chen, X., Cousins, S., Flaspohler, D., Gruner, D. S., Johnson, J. S., Surasinghe, T. D., Zambrano, J., & Buma, B. (2023). Forest ecosystem properties emerge from interactions of structure and disturbance. *Frontiers in Ecology and the Environment*, *21*(1), 14–23. https://doi.org/10.1002/fee.2589

Mulverhill, C., Coops, N. C., Hermosilla, T., White, J. C., & Wulder, M. A. (2022). Evaluating ICESat-2 for monitoring, modeling, and update of large area forest canopy height products. *Remote Sensing of Environment*, *271*, 112919. https://doi.org/10.1016/j.rse.2022.112919

NC OneMap. (2020). *Hurricane Florence Flood Extent Across the Piedmont and Coastal Plain of North Carolina* [Imagery Dataset]. https://services.nconemap.gov/secure/rest/services/Florence\_Flood\_Extent/ImageServer

Neuenschwander, A., Duncanson, L., Montesano, P., Minor, D., Guenther, E., Hancock, S., Wulder, M. A., White, J. C., Purslow, M., Thomas, N., Mandel, A., Feng, T., Armston, J., Kellner, J. R., Andersen, H. E., Boschetti, L., Fekety, P., Hudak, A., Pisek, J., … Stereńczak, K. (2024). Towards global spaceborne lidar biomass: Developing and applying boreal forest biomass models for ICESat-2 laser altimetry data. *Science of Remote Sensing*, *10*, 100150. https://doi.org/10.1016/j.srs.2024.100150

Neuenschwander, A., Guenther, E., White, J. C., Duncanson, L., & Montesano, P. (2020). Validation of ICESat-2 terrain and canopy heights in boreal forests. *Remote Sensing of Environment*, *251*, 112110. https://doi.org/10.1016/j.rse.2020.112110

Neuenschwander, A., & Magruder, L. A. (2019). Canopy and Terrain Height Retrievals with ICESat-2: A First Look. *Remote Sensing*, *11*(14), 1721. https://doi.org/10.3390/rs11141721

Neuenschwander, A., & Pitts, K. (2019). The ATL08 land and vegetation product for the ICESat-2 Mission. *Remote Sensing of Environment*, *221*, 247–259. https://doi.org/10.1016/j.rse.2018.11.005

Neuenschwander, A., Pitts, K., Jelley, B. J., Robbins, J., Markel, J., Popescu, S., Nelson, R., Harding, D., Pederson, Klotz, B., & Sheridan, R. (2023). *Ice, Cloud, and Land Elevation Satellite (ICESat-2) Project Algorithm Theoretical Basis Document (ATBD) for Land—Vegetation Along-Track Products (ATL08), version 6*. https://doi.org/10.5067/8ANPSL1NN7YS

Neumann, T. A., Brenner, A., Hancock, D., Robins, J., Saba, J., Harbeck, K., Gibbons, A., Lee, J., Luthcke, S., & Rebold, T. (2023). *Ice, Cloud, and Land Elevation Satellite (ICESat-2) Project Algorithm Theoretical Basis Document (ATBD) for Global Geolocated Photons ATL03, version 6*. https://doi.org/10.5067/GA5KCLJT7LOT

Neumann, T. A., Martino, A. J., Markus, T., Bae, S., Bock, M. R., Brenner, A. C., Brunt, K. M., Cavanaugh, J., Fernandes, S. T., Hancock, D. W., Harbeck, K., Lee, J., Kurtz, N. T., Luers, P. J., Luthcke, S. B., Magruder, L., Pennington, T. A., Ramos-Izquierdo, L., Rebold, T., … Thomas, T. C. (2019a). The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System. *Remote Sensing of Environment*, *233*, 111325. https://doi.org/10.1016/j.rse.2019.111325

Neumann, T. A., Martino, A. J., Markus, T., Bae, S., Bock, M. R., Brenner, A. C., Brunt, K. M., Cavanaugh, J., Fernandes, S. T., Hancock, D. W., Harbeck, K., Lee, J., Kurtz, N. T., Luers, P. J., Luthcke, S. B., Magruder, L., Pennington, T. A., Ramos-Izquierdo, L., Rebold, T., … Thomas, T. C. (2019b). The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System. *Remote Sensing of Environment*, *233*, 111325. https://doi.org/10.1016/j.rse.2019.111325

Noordermeer, L., Bollandsås, O. M., Gobakken, T., & Næsset, E. (2018). Direct and indirect site index determination for Norway spruce and Scots pine using bitemporal airborne laser scanner data. *Forest Ecology and Management*, *428*, 104–114. https://doi.org/10.1016/j.foreco.2018.06.041

Noordermeer, L., Gobakken, T., Næsset, E., & Bollandsås, O. M. (2020). Predicting and mapping site index in operational forest inventories using bitemporal airborne laser scanner data. *Forest Ecology and Management*, *457*, 117768. https://doi.org/10.1016/j.foreco.2019.117768

Oh, S., Jung, J., Shao, G., Shao, G., Gallion, J., & Fei, S. (2022). High-Resolution Canopy Height Model Generation and Validation Using USGS 3DEP LiDAR Data in Indiana, USA. *Remote Sensing*, *14*(4), 935. https://doi.org/10.3390/rs14040935

Omernik, J. M., & Griffith, G. E. (2014). Ecoregions of the Conterminous United States: Evolution of a Hierarchical Spatial Framework. *Environmental Management*, *54*(6), 1249–1266. https://doi.org/10.1007/s00267-014-0364-1

Pan, Y., Birdsey, R. A., Phillips, O. L., Houghton, R. A., Fang, J., Kauppi, P. E., Keith, H., Kurz, W. A., Ito, A., Lewis, S. L., Nabuurs, G.-J., Shvidenko, A., Hashimoto, S., Lerink, B., Schepaschenko, D., Castanho, A., & Murdiyarso, D. (2024). The enduring world forest carbon sink. *Nature*, *631*(8021), 563–569. https://doi.org/10.1038/s41586-024-07602-x

Paul, S., Ghebreyesus, D., & Sharif, H. O. (2019). Brief Communication: Analysis of the Fatalities and Socio-Economic Impacts Caused by Hurricane Florence. *Geosciences*, *9*(2), 58. https://doi.org/10.3390/geosciences9020058

Pingel, T. J., Clarke, K. C., & McBride, W. A. (2013). An improved simple morphological filter for the terrain classification of airborne LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *77*, 21–30. https://doi.org/10.1016/j.isprsjprs.2012.12.002

Podest, E. (2021, March 16). *The Fundamentals of LiDAR*. ARSET - Use of Solar Induced Fluorescence and LIDAR to Assess Vegetation Change and Vulnerability, Jet Propulsion Laboratory, California Institute of Technology.

Pronk, M., Eleveld, M., & Ledoux, H. (2024). Assessing Vertical Accuracy and Spatial Coverage of ICESat-2 and GEDI Spaceborne Lidar for Creating Global Terrain Models. *Remote Sensing*, *16*(13), 2259. https://doi.org/10.3390/rs16132259

Psistaki, K., Tsantopoulos, G., & Paschalidou, A. K. (2024). An Overview of the Role of Forests in Climate Change Mitigation. *Sustainability*, *16*(14), Article 14. https://doi.org/10.3390/su16146089

R Core Team. (2023). *R: A Language and Environment for Statistical Computing* [Computer software]. R Foundation for Statistical Computing. https://www.R-project.org

radt0005. (2025). *FIADB\_Direct* [R]. https://github.com/radt0005/FIADB\_Direct (Original work published 2025)

Rai, N., Ma, Q., Poudel, K. P., Himes, A., & Meng, Q. (2024). Evaluating the Uncertainties in Forest Canopy Height Measurements Using ICESat-2 Data. *Journal of Remote Sensing*, *4*, 0160. https://doi.org/10.34133/remotesensing.0160

Ribas-Costa, V. A., Gastón, A., Bloszies, S. A., Henderson, J. D., Trlica, A., Carter, D. R., Rubilar, R., Albaugh, T. J., & Cook, R. L. (2024). Nature vs. nurture: Drivers of site productivity in loblolly pine (Pinus taeda L.) forests in the southeastern US. *Forest Ecology and Management*, *572*, 122334. https://doi.org/10.1016/j.foreco.2024.122334

Ribas-Costa, V. A., Gastón, A., & Cook, R. L. (2024). Modeling dominant height with USGS 3DEP LiDAR to determine site index in even-aged loblolly pine ( *Pinus taeda* L.) plantations in the southeastern US. *Forestry: An International Journal of Forest Research*, cpae034. https://doi.org/10.1093/forestry/cpae034

Ringard, J., Chiriaco, M., Bastin, S., & Habets, F. (2019). Recent trends in climate variability at the local scale using 40 years of observations: The case of the Paris region of France. *Atmospheric Chemistry and Physics*, *19*(20), 13129–13155. https://doi.org/10.5194/acp-19-13129-2019

Risbøl, O., & Gustavsen, L. (2018). LiDAR from drones employed for mapping archaeology – Potential, benefits and challenges. *Archaeological Prospection*, *25*(4), 329–338. https://doi.org/10.1002/arp.1712

Roberts, S. D., Dean, T. J., Evans, D. L., McCombs, J. W., Harrington, R. L., & Glass, P. A. (2005). Estimating individual tree leaf area in loblolly pine plantations using LiDAR-derived measurements of height and crown dimensions. *Forest Ecology and Management*, *213*(1–3), 54–70. https://doi.org/10.1016/j.foreco.2005.03.025

Roriz, R., Cabral, J., & Gomes, T. (2022). Automotive LiDAR Technology: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 6282–6297. https://doi.org/10.1109/TITS.2021.3086804

Rossum, G. van, & Drake, F. L. (2010). *The Python language reference* (Release 3.0.1 [Repr.]). Python Software Foundation.

Schutz, B. E., Zwally, H. J., Shuman, C. A., Hancock, D., & DiMarzio, J. P. (2005). Overview of the ICESat Mission. *Geophysical Research Letters*, *32*(21), 2005GL024009. https://doi.org/10.1029/2005GL024009

Seidl, R., Thom, D., Kautz, M., Martin-Benito, D., Peltoniemi, M., Vacchiano, G., Wild, J., Ascoli, D., Petr, M., Honkaniemi, J., Lexer, M. J., Trotsiuk, V., Mairota, P., Svoboda, M., Fabrika, M., Nagel, T. A., & Reyer, C. P. O. (2017). Forest disturbances under climate change. *Nature Climate Change*, *7*(6), 395–402. https://doi.org/10.1038/nclimate3303

Sen, P. K. (1968). Estimates of the Regression Coefficient Based on Kendall’s Tau. *Journal of the American Statistical Association*, *63*(324), 1379–1389. https://doi.org/10.1080/01621459.1968.10480934

Shtaynberger, J., & Bar, H. (2023). *Equivalence Testing*. Cornell Statistical Consulting Unit. https://cscu.cornell.edu/wp-content/uploads/equiv.pdf

Shuman, C. A., Zwally, H. J., Schutz, B. E., Brenner, A. C., DiMarzio, J. P., Suchdeo, V. P., & Fricker, H. A. (2006). ICESat Antarctic elevation data: Preliminary precision and accuracy assessment. *Geophysical Research Letters*, *33*(7), 2005GL025227. https://doi.org/10.1029/2005GL025227

Socha, J., Pierzchalski, M., Bałazy, R., & Ciesielski, M. (2017). Modelling top height growth and site index using repeated laser scanning data. *Forest Ecology and Management*, *406*, 307–317. https://doi.org/10.1016/j.foreco.2017.09.039

Sturtevant, B. R., & Fortin, M.-J. (2021). Understanding and Modeling Forest Disturbance Interactions at the Landscape Level. *Frontiers in Ecology and Evolution*, *9*, 653647. https://doi.org/10.3389/fevo.2021.653647

Sumnall, M. J., Albaugh, T. J., Carter, D. R., Cook, R. L., Hession, W. C., Campoe, O. C., Rubilar, R. A., Wynne, R. H., & Thomas, V. A. (2022). Effect of varied unmanned aerial vehicle laser scanning pulse density on accurately quantifying forest structure. *International Journal of Remote Sensing*, *43*(2), 721–750. https://doi.org/10.1080/01431161.2021.2023229

Sun, T., Qi, J., & Huang, H. (2020). Discovering forest height changes based on spaceborne lidar data of ICESat-1 in 2005 and ICESat-2 in 2019: A case study in the Beijing-Tianjin-Hebei region of China. *Forest Ecosystems*, *7*(1), 53. https://doi.org/10.1186/s40663-020-00265-w

Tang, J., Luyssaert, S., Richardson, A. D., Kutsch, W., & Janssens, I. A. (2014). Steeper declines in forest photosynthesis than respiration explain age-driven decreases in forest growth. *Proceedings of the National Academy of Sciences*, *111*(24), 8856–8860. https://doi.org/10.1073/pnas.1320761111

The pandas development team. (2024). *pandas-dev/pandas: Pandas* (Version v2.2.3) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.3509134

Theil, H. (1992). A Rank-Invariant Method of Linear and Polynomial Regression Analysis. In B. Raj & J. Koerts (Eds.), *Henri Theil’s Contributions to Economics and Econometrics* (Vol. 23, pp. 345–381). Springer Netherlands. https://doi.org/10.1007/978-94-011-2546-8\_20

U.S. Department of Agriculture Forest Service. (2023, January 13). *Forest Inventory and Analysis*. Research. https://research.fs.usda.gov/programs/fia

U.S. Department of Agriculture Forest Service. (2024). *The Forest Inventory and Analysis Database User Guide (NFI): Vol. Database Description*.

U.S. Geological Survey. (2024). *Annual National Land Cover Database (NLCD)* (Science Product User Guide No. LSDS-2103; p. 43). https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/media/files/LSDS-2103%20Annual%20National%20Land%20Cover%20Database%20%28NLCD%29%20Collection%201%20Science%20Product%20User%20Guide%20-v1.0%202024\_10\_15%201.pdf

Wachs, S. (2015). *What Is Equivalence Testing & When Should We Use It?* Integral Concepts, Inc. https://www.integral-concepts.com/wp-content/media/What-is-Equivalence-Testing-and-When-Should-We-Use-It.pdf

Wang, X., Liang, X., Gong, W., Häkli, P., & Wang, Y. (2024). Accuracy fluctuations of ICESat-2 height measurements in time series. *International Journal of Applied Earth Observation and Geoinformation*, *135*, 104234. https://doi.org/10.1016/j.jag.2024.104234

Wild, C. J., & Seber, G. A. F. (2000). The Wilcoxon Test. In *Chance encounters: A first course in data analysis and inference* (p. 10). Wiley.

Wilkes, P., Jones, S. D., Suarez, L., Haywood, A., Woodgate, W., Soto-Berelov, M., Mellor, A., & Skidmore, A. K. (2015). Understanding the Effects of ALS Pulse Density for Metric Retrieval across Diverse Forest Types. *Photogrammetric Engineering & Remote Sensing*, *81*(8), 625–635. https://doi.org/10.14358/PERS.81.8.625

Xi, Z., Xu, H., Xing, Y., Gong, W., Chen, G., & Yang, S. (2022). Forest Canopy Height Mapping by Synergizing ICESat-2, Sentinel-1, Sentinel-2 and Topographic Information Based on Machine Learning Methods. *Remote Sensing*, *14*(2), 364. https://doi.org/10.3390/rs14020364

Xiang, B., Wielgosz, M., Kontogianni, T., Peters, T., Puliti, S., Astrup, R., & Schindler, K. (2024). Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3D deep learning. *Remote Sensing of Environment*, *305*, 114078. https://doi.org/10.1016/j.rse.2024.114078

Yu, C., & Yao, W. (2017). Robust linear regression: A review and comparison. *Communications in Statistics - Simulation and Computation*, *46*(8), 6261–6282. https://doi.org/10.1080/03610918.2016.1202271

Yu, J., Nie, S., Liu, W., Zhu, X., Lu, D., Wu, W., & Sun, Y. (2022). Accuracy Assessment of ICESat-2 Ground Elevation and Canopy Height Estimates in Mangroves. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5. https://doi.org/10.1109/LGRS.2021.3107440

Yu, Q., Ryan, M. G., Ji, W., Prihodko, L., Anchang, J. Y., Kahiu, N., Nazir, A., Dai, J., & Hanan, N. P. (2024). Assessing canopy height measurements from ICESat-2 and GEDI orbiting LiDAR across six different biomes with G-LiHT LiDAR. *Environmental Research: Ecology*, *3*(2), 025001. https://doi.org/10.1088/2752-664X/ad39f2

Yu, X., Hyyppä, J., Kaartinen, H., & Maltamo, M. (2004). Automatic detection of harvested trees and determination of forest growth using airborne laser scanning. *Remote Sensing of Environment*, *90*(4), 451–462. https://doi.org/10.1016/j.rse.2004.02.001

Zenner, E. K., & Hibbs, D. E. (2000). A new method for modeling the heterogeneity of forest structure. *Forest Ecology and Management*, *129*(1–3), 75–87. https://doi.org/10.1016/S0378-1127(99)00140-1

Zhao, H., Wu, J., Wang, A., Guan, D., & Liu, Y. (2022). Microtopography mediates the climate–growth relationship and growth resilience to drought of Pinus tabulaeformis plantation in the hilly site. *Frontiers in Plant Science*, *13*, 1060011. https://doi.org/10.3389/fpls.2022.1060011

Zhao, K., Suarez, J. C., Garcia, M., Hu, T., Wang, C., & Londo, A. (2018). Utility of multitemporal lidar for forest and carbon monitoring: Tree growth, biomass dynamics, and carbon flux. *Remote Sensing of Environment*, *204*, 883–897. https://doi.org/10.1016/j.rse.2017.09.007

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

Zolkos, S. G., Goetz, S. J., & Dubayah, R. (2013). A meta-analysis of terrestrial aboveground biomass estimation using lidar remote sensing. *Remote Sensing of Environment*, *128*, 289–298. https://doi.org/10.1016/j.rse.2012.10.017

1. **Appendix**
   1. **Python Script to Extract Canopy Height Metrics from ALS Data in EPT Format**

# THIS SCRIPT GATHERS USGS 3DEP LIDAR DATA IN PARALLEL

# USE IT FOR COMPARING SPACEBORNE AND AIRBORNE CANOPY HEIGHT OBSERVATIONS

import pandas as pd

import pdal

import geopandas as gpd

import json

from scipy import stats

import numpy as np

import copy

import os

from concurrent.futures import ProcessPoolExecutor

from pathlib import Path

import glob

pipelineTemplate = {

    "pipeline" : [

        # READS DATA FROM EPT

        {

            "type": "readers.ept",

            "filename":"" , # BLANK FILENAME TO ACCOMMODATE SPECIFIC 3DEP PROJECT

            "bounds": "", # READING WTIHIN BOUNDS OF SPECIFIC POLYGON

        },

        # removed the elm filter

        # removed the outliers filter

        # HANDLES SMRF ERROR WITH NO RETURNS

        {

            "type": "filters.assign",

            "value": [

              "ReturnNumber = 1 WHERE ReturnNumber < 1",

              "NumberOfReturns = 1 WHERE NumberOfReturns < 1"

            ]

        },

        # REPROJECT TO OUR STUDY AREA CRS: NAD83 NC STATE PLANE

        # {

        #     "type": "filters.reprojection",

        #     "out\_srs": "EPSG:32119",

        #     "tag": "reprojectUTM"

        # },

        # SMRF FILTER TO IDENTIFY GROUND PER PINGEL ET. AL (2013)

        {

          "type":"filters.smrf",

          "ignore":"Classification[7:7]",

          "slope":0.2,

          "window":16,

          "threshold":0.45,

          "scalar":1.2

        },

        # CUTOFF FOR SMRF VS CSF FILTER ROUTES

        {

            "type":"filters.hag\_nn"

        },

        # CROPPING TO BOUNDS OF SPECIFIC POLYGON

        {

            "type":"filters.crop",

            "polygon":""

        },

        # UNCOMMENT IF YOU WANT TO WRITE CROPPED POINT CLOUD FOR TESTING

        # {

        #     "type":"writers.las",

        #     "filename":""

        # }

    ]

}

def calculateMetrics(pointCloudArray, outputUniqueID):

    # Empty object to hold vegetation metrics:

    geosegmentStats = None

    # Getting the normalized heights out as numpy array

    allReturns = pd.DataFrame(pointCloudArray[0])

    # number of total points

    totalPoints = len(allReturns)

    # This is the case where it falls outside the EPT returns, it'll have no returns regardless of the heights

    if totalPoints < 1:

        print(f'NO RETURNS AT ALL FOR THIS UID {outputUniqueID}, FALLS OUTSIDE OF EPT COLLECTION')

        return None

    # These two queries to only vegetation points in the height range of interest between 2 meters and global maxima (39.423 meters)

    vegReturns = allReturns.query('Classification != 2') # removes ground to get only vegetation returns

    validReturns = vegReturns.query('HeightAboveGround > 2 and HeightAboveGround < 39.392207153320314') # valid height range removes shrubs & high noise (birds/powerlines)

    heights = validReturns['HeightAboveGround'] # Now we can just grab the vegetation heights to do statistics

    totalVegPoints = len(heights)

    # This is the case where there is only ground and no valid vegetation (powerlines)

    # This is still interesting as it shows ICESat has spatial inaccuracies

    if totalVegPoints < 1:

        # Output the results as a dictionary, makes it easy to reformat later

        geosegmentStats = {

            'outputID' : outputUniqueID,

            'total\_points': totalPoints, # this will use the number of GROUND returns

            'total\_veg\_points': 0, # this will use the number of GROUND returns

            'veg\_proportion': 0,

            'max': 0,

            'min': 0,

            'mean': 0,

            'median': 0,

            'percentile\_90':0,

            'percentile\_95':0,

            'percentile\_98':0,

            'percentile\_99':0

        }

    # This is the case where there actually is vegetation

    elif totalPoints >= 1:

        # Calculate height metrics

        max\_height = np.max(heights)

        min\_height = np.min(heights)

        mean\_height = np.mean(heights)

        median\_height = np.median(heights)

        # Height percentiles

        p90,p95, p98, p99 = np.percentile(heights,[90,95,98,99])

        # Have to convert to a pandas series to run these stats

        pdSeries = pd.Series(heights)

        # Output the results as a dictionary, makes it easy to reformat later

        geosegmentStats = {

            'outputID' : outputUniqueID,

            'total\_points': totalPoints, # this will use the number of GROUND returns

            'total\_veg\_points': totalVegPoints, # this will use the number of VEGETATION returns

            'veg\_proportion': totalVegPoints / totalPoints,

            'max': max\_height,

            'min': min\_height,

            'mean': mean\_height,

            'median': median\_height,

            'percentile\_90':p90,

            'percentile\_95':p95,

            'percentile\_98':p98,

            'percentile\_99':p99

        }

    metricsDf = pd.DataFrame([geosegmentStats])

    return metricsDf

def executePipeline(dataframeRow):

    uid = dataframeRow['segmentUID']

    eptAsset = dataframeRow['asset']

    projectID = dataframeRow['projectID']

    outputID = dataframeRow['outputID']

    # creating copy of pipeline template for specific sample

    pipelineConfig = copy.deepcopy(pipelineTemplate)

    # geometry wkt in original crs as its already in a column

    wkt = dataframeRow.geometry.wkt

    # geometry bounds in format: ([xmin, xmax], [ymin, ymax], [zmin, zmax])

    boundsString = dataframeRow['bounds']

    # boundsString = f"([{bounds[0]}, {bounds[2]}], [{bounds[1]}, {bounds[3]}])"

    # adjusting pipeline specific to the row

    pipelineConfig['pipeline'][0]['bounds'], pipelineConfig['pipeline'][0]['filename'], pipelineConfig['pipeline'][4]['polygon'] = boundsString, eptAsset, wkt

    # uncomment this if you want to write .las file for testing

    # pipelineConfig['pipeline'][6]['filename'] = f"D:\\IceSat\\Final\\segments\\pointCloudTesting\\{outputID}.las"

    # formats custom pipeline into JSON

    pipelineJSON = json.dumps(pipelineConfig)

    try:

        pipeline = pdal.Pipeline(pipelineJSON) # converts JSON into PDAL pipeline object

        pipeline.execute() # executes pipelins

        PCarray = pipeline.arrays # gets array of return data

        metrics = calculateMetrics(PCarray,outputID) # feeds array of return data into function for calculating vegetation metrics

        # finally returns vegetation metrics array to main function

        return metrics

    except Exception as e:

        return None

def main():

    # 1. Read & ensure correct CRS

    polygons = gpd.read\_parquet(r"D:\IceSat\Final\segments\workingSetPolygonsWITHOUTALSMetrics.parquet")

    print(f"\nProcessing {len(polygons)} polygons...")

    print('\nCreating bounds in epsg 3857')

    polygons3857 = polygons.to\_crs(epsg=3857)

    bounds = polygons3857.geometry.bounds

    # print(bounds)

    polygons3857['bounds'] = bounds.apply(

        lambda row: (

        f"([{row['minx']}, {row['maxx']}], "

        f"[{row['miny']}, {row['maxy']}])"

    ), axis=1)

    # print(polygons3857)

    rows = [row for \_, row in polygons3857.iterrows()]

    print('\nNow extracting EPT data')

    # Process rows in parallel

    with ProcessPoolExecutor(max\_workers=15) as executor:

        results = list(executor.map(executePipeline, rows))

    # creating list of results from pipeline processing

    results = [df for df in results if df is not None]

    # stacking these resulting dataframes

    combinedMetrics = pd.concat(results, axis=0).reset\_index(drop=True)

    print("Merging with polygons")

    merged = polygons3857.merge(combinedMetrics, on="outputID", how='left')

    mergedDroppedNAs = merged.dropna(subset=['total\_veg\_points'])

    mergedMaxVegPoints = mergedDroppedNAs.loc[mergedDroppedNAs.groupby('segmentUID')['total\_veg\_points'].idxmax()].reset\_index(drop=True) # dropping multi-project polygons by selecting project with max returns

    print("Writing to file")

    reprojectedMaxVegPoints = mergedMaxVegPoints.to\_crs(32119)

    reprojectedMaxVegPoints.to\_parquet(r"D:\IceSat\Final\segments\newWorkingSetPolygonsWITHALSMetrics.parquet")

    reprojectedMaxVegPoints.to\_file(r"D:\IceSat\Final\segments\newWorkingSetPolygonsWITHALSMetrics.GeoJSON")

    print("Done. Results written to newWorkingSetPolygonsWITHALSMetrics.parquet")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

* 1. **Python Script to Extract Canopy Height Metrics from ATL08 Granules**

import pandas as pd

import geopandas as gpd

from scipy import stats

import numpy as np

import os

from concurrent.futures import ProcessPoolExecutor, as\_completed

from pathlib import Path

import glob

import h5py as h5

# Change this to directory of raw .h5 files.

rawGranuleDirectory = r"D:\IceSat\Final\granules\\*.h5"

# Change this to the directory of extracted ICESat-2 granules. Ouputs will be in .parquet format

outputDirectory = r"D:\IceSat\Final\extracted"

# Here we define a project-wide CRS for all our ICESat-2 segments, as well as any geometry we want to clip our segments to

# For our purposes, we're picking albers equal area conic

projectCRS = 32119

# Commonly, we need to clip ICESat-2 data to only the segments collected within a focus area. Provide a path for a shapefile to read in

# For this research we'll use Little's loblolly range

clippingGeometry = gpd.read\_file(r"D:\IceSat\Final\mapping\ALSdata\dissolvedProjects.parquet").to\_crs(projectCRS)

def extractFileData(filePath):

    try:

        with h5.File(filePath, 'r') as file:

            filename = file.filename # grabbing the file name from the H5 file object

            outputDataframePath = filename.replace('granules', 'extracted').replace('h5', 'parquet') # file extension for the extacted parquet

            # print('input:', filename, 'output:', outputDataframePath)

            validBeams = getBeams(file)

            allBeamDataframes = []

            for beam in validBeams:

                # Reading in ICESat-2 data from key groups to reduce file opening/closing

                land\_segments = file[beam + '/land\_segments']

                orbit\_info = file['orbit\_info']

                ancillary\_data = file['/ancillary\_data']

                rowCount = len(land\_segments['delta\_time'][:])

                data = {

                # |---------------------- COLLECTION DATA ---------------|

                    'delta\_time': land\_segments['delta\_time'][:],

                    'filename': [filename] \* rowCount,

                    'sc\_orient': [orbit\_info['sc\_orient'][0]] \* rowCount,

                    'beam': [beam] \* rowCount,

                    'segment\_landcover' : land\_segments['segment\_landcover'][:],

                    'rgt' : land\_segments['rgt'][:],

                    'start\_cycle' : [ancillary\_data['start\_cycle'][0]] \* rowCount,

                    'end\_cycle' : [ancillary\_data['end\_cycle'][0]] \* rowCount,

                    'segment\_id\_beg' : land\_segments['segment\_id\_beg'][:],

                    'segment\_id\_end' : land\_segments['segment\_id\_end'][:],

                    'lat' : land\_segments['latitude'][:],

                    'lon' : land\_segments['longitude'][:],

                # |---------------------- CANOPY DATA ---------------|

                    'h\_canopy':land\_segments['canopy/h\_canopy'][:],

                    'canopy\_openness':land\_segments['canopy/canopy\_openness'][:],

                    'h\_mean\_canopy':land\_segments['canopy/h\_mean\_canopy'][:],

                    'h\_median\_canopy':land\_segments['canopy/h\_median\_canopy'][:],

                    'h\_dif\_canopy':land\_segments['canopy/h\_dif\_canopy'][:],

                    'h\_min\_canopy':land\_segments['canopy/h\_min\_canopy'][:],

                    'h\_max\_canopy':land\_segments['canopy/h\_max\_canopy'][:],

                    'toc\_roughness':land\_segments['canopy/toc\_roughness'][:],

                    'h\_canopy\_quad':land\_segments['canopy/h\_canopy\_quad'][:],

                    'n\_ca\_photons':land\_segments['canopy/n\_ca\_photons'][:],

                    'n\_toc\_photons':land\_segments['canopy/n\_toc\_photons'][:],

                    'centroid\_height':land\_segments['canopy/centroid\_height'][:],

                    'h\_canopy\_uncertainty':land\_segments['canopy/h\_canopy\_uncertainty'][:],

                    'can\_noise':land\_segments['canopy/can\_noise'][:],

                    'dem\_h':land\_segments['dem\_h'][:],

                    'h\_te\_best\_fit':land\_segments['terrain/h\_te\_best\_fit'][:],

                    'h\_te\_interp':land\_segments['terrain/h\_te\_interp'][:],

                    'h\_te\_mean':land\_segments['terrain/h\_te\_mean'][:],

                    'h\_te\_median':land\_segments['terrain/h\_te\_median'][:],

                    'h\_te\_max':land\_segments['terrain/h\_te\_max'][:],

                    'h\_te\_min':land\_segments['terrain/h\_te\_min'][:],

                    'n\_te\_photons':land\_segments['terrain/n\_te\_photons'][:],

                    'terrain\_slope':land\_segments['terrain/terrain\_slope'][:],

                    # |---------------------- QUALITY FLAGS ---------------|

                    'brightness\_flag':land\_segments['brightness\_flag'][:],

                    'cloud\_flag':land\_segments['cloud\_flag\_atm'][:],

                    'cloud\_fold\_flag':land\_segments['cloud\_fold\_flag'][:],

                    'dem\_flag':land\_segments['dem\_flag'][:],

                    'dem\_removal\_flag':land\_segments['dem\_removal\_flag'][:],

                    'layer\_flag':land\_segments['layer\_flag'][:],

                    'msw\_flag':land\_segments['msw\_flag'][:],

                    'night\_flag':land\_segments['night\_flag'][:],

                    'ph\_removal\_flag':land\_segments['ph\_removal\_flag'][:],

                    'psf\_flag':land\_segments['psf\_flag'][:],

                    'sat\_flag':land\_segments['sat\_flag'][:],

                    'segment\_watermask':land\_segments['segment\_watermask'][:],

                    'snr':land\_segments['snr'][:],

                    'terrain\_flag':land\_segments['terrain\_flg'][:],

                    'urban\_flag':land\_segments['urban\_flag'][:],

                    # |---------------------- GEOSEGMENT DATA ---------------|

                    # LATITUDE COORDS

                    'lat\_0\_20':land\_segments['latitude\_20m/'][:, 0],

                    'lat\_20\_40':land\_segments['latitude\_20m/'][:, 1],

                    'lat\_40\_60':land\_segments['latitude\_20m/'][:, 2],

                    'lat\_60\_80':land\_segments['latitude\_20m/'][:, 3],

                    'lat\_80\_100':land\_segments['latitude\_20m/'][:, 4],

                    # LONGITUDE COORDS

                    'long\_0\_20':land\_segments['longitude\_20m/'][:, 0],

                    'long\_20\_40':land\_segments['longitude\_20m/'][:, 1],

                    'long\_40\_60':land\_segments['longitude\_20m/'][:, 2],

                    'long\_60\_80':land\_segments['longitude\_20m/'][:, 3],

                    'long\_80\_100':land\_segments['longitude\_20m/'][:, 4],

                    # CANOPY HEIGHTS

                    'h\_canopy\_0\_20':land\_segments['canopy/h\_canopy\_20m/'][:, 0],

                    'h\_canopy\_20\_40':land\_segments['canopy/h\_canopy\_20m/'][:, 1],

                    'h\_canopy\_40\_60':land\_segments['canopy/h\_canopy\_20m/'][:, 2],

                    'h\_canopy\_60\_80':land\_segments['canopy/h\_canopy\_20m/'][:, 3],

                    'h\_canopy\_80\_100':land\_segments['canopy/h\_canopy\_20m/'][:, 4],

                    # TERRAIN ESTIMATES

                    'h\_te\_best\_fit\_0\_20':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 0],

                    'h\_te\_best\_fit\_20\_40':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 1],

                    'h\_te\_best\_fit\_40\_60':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 2],

                    'h\_te\_best\_fit\_60\_80':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 3],

                    'h\_te\_best\_fit\_80\_100':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 4],

                    # |---------------------- CANOPY HEIGHT METRICS ---------------|

                    'canopy\_h\_metrics\_10':land\_segments['canopy/canopy\_h\_metrics/'][:,0],

                    'canopy\_h\_metrics\_15':land\_segments['canopy/canopy\_h\_metrics/'][:,1],

                    'canopy\_h\_metrics\_20':land\_segments['canopy/canopy\_h\_metrics/'][:,2],

                    'canopy\_h\_metrics\_25':land\_segments['canopy/canopy\_h\_metrics/'][:,3],

                    'canopy\_h\_metrics\_30':land\_segments['canopy/canopy\_h\_metrics/'][:,4],

                    'canopy\_h\_metrics\_35':land\_segments['canopy/canopy\_h\_metrics/'][:,5],

                    'canopy\_h\_metrics\_40':land\_segments['canopy/canopy\_h\_metrics/'][:,6],

                    'canopy\_h\_metrics\_45':land\_segments['canopy/canopy\_h\_metrics/'][:,7],

                    'canopy\_h\_metrics\_50':land\_segments['canopy/canopy\_h\_metrics/'][:,8],

                    'canopy\_h\_metrics\_55':land\_segments['canopy/canopy\_h\_metrics/'][:,9],

                    'canopy\_h\_metrics\_60':land\_segments['canopy/canopy\_h\_metrics/'][:,10],

                    'canopy\_h\_metrics\_65':land\_segments['canopy/canopy\_h\_metrics/'][:,11],

                    'canopy\_h\_metrics\_70':land\_segments['canopy/canopy\_h\_metrics/'][:,12],

                    'canopy\_h\_metrics\_75':land\_segments['canopy/canopy\_h\_metrics/'][:,13],

                    'canopy\_h\_metrics\_80':land\_segments['canopy/canopy\_h\_metrics/'][:,14],

                    'canopy\_h\_metrics\_85':land\_segments['canopy/canopy\_h\_metrics/'][:,15],

                    'canopy\_h\_metrics\_90':land\_segments['canopy/canopy\_h\_metrics/'][:,16],

                    'canopy\_h\_metrics\_95':land\_segments['canopy/canopy\_h\_metrics/'][:,17]

                }

                # Converting dictionary to dataframe to hold granule data

                beamDataframe = pd.DataFrame(data)

                # Describing spacecraft orientation

                beamDataframe['orientation'] = beamDataframe['sc\_orient'].map({0: 'backward', 1: 'forward', 2: 'transition'})

                # Gathering datetime information based on delta time, uses seconds since 2018/01/01 00:00:00

                beamDataframe['datetime'] = pd.to\_datetime(beamDataframe['delta\_time'], unit='s', origin='2018-01-01')

                beamDataframe['year'] = beamDataframe['datetime'].dt.year

                beamDataframe['month'] = beamDataframe['datetime'].dt.month

                # Determining beam strength of collection

                beamDataframe['beam\_strength'] = beamDataframe.apply(determineBeamStrength, axis=1)

                # Assigning a unique ID to the segment describing its collection scheme

                beamDataframe['segmentUID'] = beamDataframe[['start\_cycle', 'rgt', 'segment\_id\_beg', 'beam', 'year']].astype(str).agg('\_'.join, axis=1) # creating unique ID for each 100m segment

                # Append current beam dataframe to array holding all beam dataframes

                allBeamDataframes.append(beamDataframe)

            # Stacking all beam dataframes into single dataframe

            finalDataframe = pd.concat(allBeamDataframes, ignore\_index=True)

            # Here we convert our beam dataframe into a geodataframe to allow for clipping

            fileGeodataframe = gpd.GeoDataFrame(

                finalDataframe, geometry=gpd.points\_from\_xy(finalDataframe.lon, finalDataframe.lat), crs="EPSG:4326"

            ).to\_crs(projectCRS)

            # Clipping to our geometry of interest

            clippedGeodataframe = fileGeodataframe.clip(clippingGeometry).reset\_index(drop=True)

            # Skips writing a file for files that have no ICESat-2 segments after clipping to our geometry

            if clippedGeodataframe.empty:

                return (f'empty dataframe after clip for {filename}')

            # If its not empty, we write to parquet

            clippedGeodataframe.to\_parquet(outputDataframePath) # writing our clipped geodataframe to parquet file

            return (f'Successful extraction of {outputDataframePath} to dataframe of length {len(clippedGeodataframe)}')

    # Error handling if the file could not be opened

    except IOError as errorMessage:

        return f'for {filePath} the following error was received: {errorMessage}'

# Function to determine the strength of the beam based on spacecraft orientation

def determineBeamStrength(row):

    if row['orientation'] == 'forward' and row['beam'] in ['gt1r', 'gt2r', 'gt3r']:

        return 'strong'

    elif row['orientation'] == 'backward' and row['beam'] in ['gt1l', 'gt2l', 'gt3l']:

        return 'strong'

    else:

        return 'weak'

# Functiont to gather beams for a file that actually have land\_segments data

def getBeams(fileObject):

    allBeams = [beam for beam in list(fileObject.keys()) if 'gt' in beam]

    landSegmentBeams = []

    for beam in allBeams:

        if '{}/land\_segments/'.format(beam) in fileObject:

            landSegmentBeams.append(beam)

    return landSegmentBeams

# Function to extract data from ICESat-2 h5 files in parallel

def processGranulesInParallel(filePaths):

    results = []

    with ProcessPoolExecutor(max\_workers=11) as executor:

        future\_to\_file = {executor.submit(extractFileData, file): file for file in filePaths}

        for future in as\_completed(future\_to\_file):

            file = future\_to\_file[future]

            try:

                result = future.result()

                results.append(result)

            except Exception as exc:

                results.append(f"Failure: Exception while processing {file} - {str(exc)}")

    return results

# Main function

def main():

    rawGranules = glob.glob(rawGranuleDirectory)

    print(f'\n|--------------------- There are {len(rawGranules)} granules to process, beginning extraction -----------------|')

    results = processGranulesInParallel(rawGranules)

    # results = list(map(extractFileData, rawGranules))

    for result in results:

        print(result)

    print(f'\n|--------------------- Extraction complete -----------------|')

# Entry point check

if \_\_name\_\_ == "\_\_main\_\_":

    main()

* 1. **R-Script to Access North Carolina FIADB and Conduct Analyses of Phase 3**

---

title: "FIA Study Area Query"

output: word\_document

date: "2025-03-31"

execute:

warning: False

---

# Imports

```{r}

library(tidyverse)

library(dplyr)

library(RobustLinearReg)

library(ggplot2)

library(glue)

data\_path = file.path("C:/Users/poncy/Desktop/Masters/Research/FIA/FIADB\_Direct/Working")

source(file.path(file.path(data\_path,'FIADB\_Direct\_source\_all.R')))

```

# Plot theme

```{r}

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.text = element\_text(size=12, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

```

# Current measurements

```{r}

NC\_plots\_AGB <- PLOT\_obs(EVAL\_GRP = 372023, ATTRIBUTE\_NBR = 10,GRP\_BY\_ATTRIB = c('unitcd','countycd')) |>

filter(COUNTYCD %in% c(13, 15, 17, 19, 29, 31, 41, 47, 49, 53, 55, 61, 65, 69, 73, 79, 83, 91, 95,101, 103, 107, 117, 127, 129, 131, 133, 137, 139, 141, 143, 147, 155, 163, 177, 181, 185, 187, 191, 195)

)

# unique plot sequence numbers for EVAL\_GRP 372023 within study area

PLT\_CN <- unique(NC\_plots\_AGB$PLT\_CN) # 1483 unique plot sequence numbers

# this grabs plots from the PLOT table with sequnce numbers matching our NC\_plots\_AGB

plot <- GET\_record(TABLE\_NAME = 'plot',VAR\_NAME = 'cn',VAR\_VALUES = PLT\_CN) |>

rename(PLT\_CN = CN)

# this grabs conditions from the CONDITION table with sequnce numbers matching our NC\_plots\_AGB

cond <- GET\_record(TABLE\_NAME = 'cond',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PLT\_CN)

# unique plot conditions, as there are multiple plot sequences in cond table

PLT\_CN\_ALL <- unique(cond$PLT\_CN)

# this only gets plots which match the unique sequence numbers

plot <- plot |> filter(PLT\_CN %in% PLT\_CN\_ALL)

# i think this gets individual tree records

tree <- GET\_record(TABLE\_NAME = 'tree',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PLT\_CN\_ALL) |>

rename(TRE\_CN = CN)

### Removing dates

plot <- plot[,-grep("\_DATE",names(plot))]

cond <- cond[,-grep("\_DATE",names(cond))]

tree <- tree[,-grep("\_DATE",names(tree))]

### Joining plot, cond, tree

plot\_cond <- left\_join(plot,cond)

plot\_cond\_tree <- left\_join(plot\_cond,tree)

# test <- anti\_join(tree,plot\_cond\_tree) # Trees from plots with partial non-SYP fortypecd

### Creating current df

current\_tree <- plot\_cond\_tree |>

dplyr::select(PLT\_CN,PREV\_PLT\_CN,STATECD,UNITCD,COUNTYCD,MEASYEAR,REMPER,CONDID,COND\_STATUS\_CD,FORTYPCD,STDORGCD,

STDORGSP,DSTRBCD1,DSTRBYR1, TRTCD1,TRE\_CN,PREV\_TRE\_CN,SUBP,TREE,STATUSCD,SPCD,DIA,HT,ACTUALHT,CCLCD,

DAMLOC1,DAMTYP1,DAMSEV1) |>

rename(PLT\_CN2 = PLT\_CN,

MEASYEAR2 = MEASYEAR,

CONDID2 = CONDID,

COND\_STATUS\_CD2 = COND\_STATUS\_CD,

FORTYPCD2 = FORTYPCD,

STDORGCD2 = STDORGCD,

STDORGSP2 = STDORGSP,

DSTRBCD12 = DSTRBCD1,

DSTRBYR12 = DSTRBYR1,

TRTCD12 = TRTCD1,

TRE\_CN2 = TRE\_CN,

STATUSCD2 = STATUSCD,

DIA2 = DIA, HT2 = HT,

ACTUALHT2 = ACTUALHT,

CCLCD2 = CCLCD,

DAMLOC12 = DAMLOC1,

DAMTYP12 = DAMTYP1,

DAMSEV12 = DAMSEV1)

```

# Previous measurements

```{r}

# grabs the previous sequence numbers for each plot

PREV\_PLT\_CN <- unique(plot$PREV\_PLT\_CN) # 1485 plots

# then gets the plot records for those previous plot records

prev\_plot <- GET\_record('plot','cn',PREV\_PLT\_CN) |>

rename(PLT\_CN = CN)

# then grabs the conditions for the previous plot records

prev\_cond <- GET\_record(TABLE\_NAME = 'cond',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PREV\_PLT\_CN) |>

filter(FORTYPCD %in% c(141, 142, 161, 162, 166, 171, 402, 403, 404, 406, 407, 409, 501, 503, 504, 505, 506, 507, 508, 511, 512, 514, 515, 516, 517, 519, 520, 962, 995, 601, 602, 605, 606, 607, 608, 609, 701, 702, 703, 704, 705, 706, 708, 709))

# finally, joining tree data for previous plot records

prev\_tree <- GET\_record(TABLE\_NAME = 'tree',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PREV\_PLT\_CN) |>

rename(TRE\_CN = CN)

# drop \_DATE variables from previous plot records

prev\_plot <- prev\_plot[,-grep("\_DATE",names(prev\_plot))]

prev\_cond <- prev\_cond[,-grep("\_DATE",names(prev\_cond))]

prev\_tree <- prev\_tree[,-grep("\_DATE",names(prev\_tree))]

### Joining previous plot, cond, tree

# joinging condiiton and tree data from previous plot records

prev\_plot\_cond <- left\_join(prev\_plot,prev\_cond)

prev\_plot\_cond\_tree <- left\_join(prev\_plot\_cond,prev\_tree)

### Creating previous df

# establishing a df of previous tree measurments

previous\_tree <- prev\_plot\_cond\_tree |>

dplyr::select(PLT\_CN,STATECD,UNITCD,COUNTYCD,MEASYEAR,CONDID,COND\_STATUS\_CD,FORTYPCD,STDORGCD,

STDORGSP,DSTRBCD1,DSTRBYR1, TRTCD1,TRE\_CN,SUBP,TREE,STATUSCD,SPCD,DIA,HT,ACTUALHT,CCLCD,

DAMLOC1,DAMTYP1,DAMSEV1)

```

# Joining remeasurements

```{r}

## 4: Joining current & previous measurments

fullDf <- left\_join(current\_tree,previous\_tree,

by=join\_by(STATECD,UNITCD,COUNTYCD,SUBP,TREE,SPCD,

PREV\_PLT\_CN == PLT\_CN,PREV\_TRE\_CN == TRE\_CN)) |>

mutate(DIA\_INC = (DIA2-DIA)/REMPER,

DIA\_CHNG = DIA2-DIA,

# Calculating change measurements

HT\_CHNG = HT2-HT, # net change in height (feet)

HT\_INC = (HT2-HT)/REMPER, # height increment (feet)

NET\_CHNG\_6YR = (HT2-HT)/REMPER \* 6, # scaling rate to net change (feet)

ACTUALHT\_CHNG = ACTUALHT2-ACTUALHT, # net change in actual height (feet)

ACTUALHT\_INC = (ACTUALHT2-ACTUALHT)/REMPER, # actual height increment (feet)

NET\_ACTUALCHNG\_6YR = (ACTUALHT2-ACTUALHT)/REMPER\* 6, # scaling rate to net actual change (feet)

# Getting measurements in meters

HT\_m = HT\* .3048,

HT2\_m = HT2 \* .3048,

ACTUALHT\_m = ACTUALHT \* .3048,

ACTUALHT2\_m = ACTUALHT2 \* .3048,

# Calculating change measurements in meters

HT\_CHNGm = (HT2\_m-HT\_m), # net change in height (meters)

HT\_INCm = (HT2\_m-HT\_m)/REMPER, # height increment (meters)

NET\_CHNG\_6YR\_m = (HT2\_m-HT\_m)/REMPER \* 6, # scaling rate to net change (M)

ACTUALHT\_CHNGm = ACTUALHT2\_m-ACTUALHT\_m, # net change in actual height (meters)

ACTUALHT\_INCm = (ACTUALHT2\_m-ACTUALHT\_m)/REMPER, # actual height increment (meters)

NET\_ACTUALCHNG\_6YR\_m = (ACTUALHT2\_m-ACTUALHT\_m)/REMPER \* 6, # scaling rate to net actual change (m)

# i think this piece accounts for dia\_inc only in planted stands, as natural are multiplied by 0

DIAINC\_STDORG = DIA\_INC\*STDORGCD) |>

filter(STATUSCD <= 1 & STATUSCD2 <= 1) # STATUS CODE to dead or retired trees

```

# Adding characteristics

```{r}

## Mapping forest type

fortypMapping <- c(

# Evergreens

"141" = "Evergreen Forest", "142" = "Evergreen Forest", "161" = "Evergreen Forest",

"162" = "Evergreen Forest", "166" = "Evergreen Forest", "171" = "Evergreen Forest",

"606" = "Evergreen Forest",

# then deciduous

"501" = "Deciduous Forest", "503" = "Deciduous Forest", "504" = "Deciduous Forest",

"505" = "Deciduous Forest", "506" = "Deciduous Forest", "507" = "Deciduous Forest",

"508" = "Deciduous Forest", "511" = "Deciduous Forest", "512" = "Deciduous Forest",

"514" = "Deciduous Forest", "515" = "Deciduous Forest", "516" = "Deciduous Forest",

"517" = "Deciduous Forest", "519" = "Deciduous Forest", "520" = "Deciduous Forest",

"701" = "Deciduous Forest", "702" = "Deciduous Forest", "703" = "Deciduous Forest",

"704" = "Deciduous Forest", "705" = "Deciduous Forest", "706" = "Deciduous Forest",

"708" = "Deciduous Forest", "709" = "Deciduous Forest", "962" = "Deciduous Forest",

# then mixed

"402" = "Mixed Forest", "403" = "Mixed Forest", "404" = "Mixed Forest", "406" = "Mixed Forest",

"407" = "Mixed Forest", "409" = "Mixed Forest",

# then woody wetlands

"601" = "Woody Wetlands", "602" = "Woody Wetlands", "605" = "Woody Wetlands",

"607" = "Woody Wetlands", "608" = "Woody Wetlands", "609" = "Woody Wetlands",

# i'll leave unknown just as-is

"999" = "Unknown"

)

# then applying the mappng

fullDf <- fullDf |>

mutate(NLCD\_2023\_Label = recode(as.character(FORTYPCD), !!!fortypMapping)

)

## Mapping disturbance presence

# ADDING DISTURBANCE PRESENCE

fullDf$DisturbancePresence <- ifelse(fullDf$DSTRBCD12 == 0, 'No', 'Yes')

# ADDING YEARS SINCE DISTURBANCE

fullDf$YearsSinceDisturbance <- fullDf$MEASYEAR2 - fullDf$DSTRBYR12

# here marking continuous disturbances (9999 for disturbance year) and were negative in prev calculation

fullDf$YearsSinceDisturbance <- ifelse(fullDf$YearsSinceDisturbance < 0, 'Continuous', fullDf$YearsSinceDisturbance)

```

# Filtering

```{r}

filtered <- fullDf |>

filter(HT\_CHNGm >= 0, na.rm=TRUE) |>

filter(ACTUALHT\_CHNGm >=0, na.rm=TRUE)

```

# First, summarizing

```{r}

# total height inc m

ht\_inc\_summary <- data.frame(Statistic = names(summary(filtered$HT\_INCm)),

Value = as.numeric(summary(filtered$HT\_INCm)),

Variable = "Total Height Inc (m/yr)")

# actual height inc m

actualht\_inc\_summary <- data.frame(Statistic = names(summary(filtered$ACTUALHT\_INCm)),

Value = as.numeric(summary(filtered$ACTUALHT\_INCm)),

Variable = "Actual Height Inc (m/yr)")

# 6-year net total height change m

netchng\_6yr\_summary <- data.frame(Statistic = names(summary(filtered$NET\_CHNG\_6YR\_m)),

Value = as.numeric(summary(filtered$NET\_CHNG\_6YR\_m)),

Variable = "6-Year Net Total Height Change (m)")

# 6-year net actual height change m

netactualchng\_6yr\_summary <- data.frame(Statistic = names(summary(filtered$NET\_ACTUALCHNG\_6YR\_m)),

Value = as.numeric(summary(filtered$NET\_ACTUALCHNG\_6YR\_m)),

Variable = "6-Year Net Actual Height Change (m)")

# Stack them together

summaryTable <- bind\_rows(

ht\_inc\_summary,

actualht\_inc\_summary,

netchng\_6yr\_summary,

netactualchng\_6yr\_summary

)

# Rearrange so Variable is the first column

summaryTable <- summaryTable |> select(Variable, everything()) |>

pivot\_wider(

names\_from = Statistic,

values\_from = Value

)

summaryTable

```

# Regression

```{r}

# first doing height change by remeasurement period

tsr <- theil\_sen\_regression(ACTUALHT\_CHNGm ~ REMPER, data=filtered)

coef(tsr)

summary(tsr)

tsrSlope <- coef(tsr)[2]

```

# Graphing

## Actual height change by remper

```{r}

actualHTCHNGPlot <- ggplot(filtered, aes(REMPER, NET\_ACTUALCHNG\_6YR\_m))+

geom\_point(alpha=0.1)+

geom\_abline(slope = tsrSlope, intercept = 0.4609171, color = "red", size = 1) +

labs(title = "Actual Height Change by Remeasurement Period",

subtitle= glue("Regression Slope: {round(tsrSlope, 2)} m"),

y = "Actual Height Change (m)",

x = "Remeasurement Period (years)")+

theme\_minimal() +

customTheme

ggsave(

"D:/IceSat/writing/thesis/figures/o3\_actualheightchange\_remperplot.png",

plot = actualHTCHNGPlot,

bg = "white",

width = 8, height = 5, units = "in", dpi = 300

)

actualHTCHNGPlot

```

## Increment density

```{r}

# increment in actual height

actualHTINCPlot <- ggplot(filtered, aes(x = ACTUALHT\_INCm)) +

geom\_histogram(aes(y = ..density..),

colour = 1, fill='white') +

geom\_density(fill='lightblue', alpha=0.6)+

geom\_vline(xintercept=median(filtered$ACTUALHT\_INCm), color='red', lwd=.75)+

annotate("text", x = .65, y = 1.5, size=5, face='bold', color='black', label = glue("Median: {round(median(filtered$ACTUALHT\_INCm), 2)

} m"))+

labs(title = "Kernel Density of Actual Height Increment",

x = "Actual Height Increment (m)",

y = "Density")+

theme\_minimal() +

customTheme

ggsave(

"D:/IceSat/writing/thesis/figures/o3\_increment\_density\_plot.png",

plot = actualHTINCPlot,

bg = "white",

width = 8, height = 5, units = "in", dpi = 300

)

actualHTINCPlot

```

## Net change density

```{r}

# increment in actual height

actualHTNetChangePlot <- ggplot(filtered, aes(x = NET\_ACTUALCHNG\_6YR\_m)) +

geom\_histogram(aes(y = ..density..),

colour = 1, fill='white') +

geom\_density(fill='lightblue', alpha=0.6)+

geom\_vline(xintercept=median(filtered$NET\_ACTUALCHNG\_6YR\_m), color='red', lwd=.75)+

annotate("text", x = 4, y = .29, size=5, face='bold', color='black', label = glue("Median: {round(median(filtered$NET\_ACTUALCHNG\_6YR\_m), 2)

} m"))+

labs(title = "Kernel Density of Six-Year Actual Height Net Change",

x = "Net Change (m)",

y = "Density")+

theme\_minimal() +

customTheme

ggsave(

"D:/IceSat/writing/thesis/figures/o3\_net\_change\_density\_plot.png",

plot = actualHTNetChangePlot,

bg = "white",

width = 8, height = 5, units = "in", dpi = 300

)

actualHTNetChangePlot

```

# Testing

```{r}

# then regrsssion of height change (m) by remeasurement period

tsrCols <- filtered |>

select(ACTUALHT\_CHNGm, REMPER)

summary(theil\_sen\_regression(ACTUALHT\_CHNGm ~ REMPER, data=tsrCols))

```

```{r}

filtered |>

filter(REMPER > 5.9 & REMPER < 7.1) |>

select(ACTUALHT\_INCm) |>

summary()

```

* 1. **Python Script to Filter ATL08 Data and Extract Land Characteristics**

#!/usr/bin/env python

# coding: utf-8

# In[1]:

from osgeo import gdal

import numpy as np

import geopandas as gpd

import pandas as pd

import rasterio

import os

import dask.dataframe as dd

from tqdm.notebook import trange, tqdm

import glob

pd.set\_option('display.max\_columns', None)

import copy

from collections import Counter

projectCRS = 32119

rasters = []

# # 1. Filtering

# Reading in extracted parquets

# In[2]:

extractedGranules = glob.glob("D:/IceSat/Final/extracted/\*.parquet")

extractedGdfs = []

for granulePath in tqdm(extractedGranules):

    gdf = gpd.read\_parquet(granulePath)

    extractedGdfs.append(gdf)

print(len(extractedGdfs))

extracted = pd.concat(extractedGdfs, ignore\_index=True)

extracted

# In[3]:

extracted.h\_canopy.max()

# Calculating global cutoff

# In[4]:

# First removing noise canopy values

valid = extracted.query('h\_canopy < 1000').reset\_index(drop=True)

globalCutoff = valid['h\_canopy'].quantile(.98) \* 1.2

globalCutoff

# Then applying all filters

# In[5]:

# Assembles query conditions, which can be commented, removed, added.

# Query column names must match column naming convention in extracted files.

queryConditions = [

    f"h\_canopy > 2 and h\_canopy < {globalCutoff}",

    "night\_flag == 1",

    "month >= 5 and month <= 9",

    "beam\_strength == 'strong'",

    "cloud\_flag <= 1",

    "msw\_flag <= 0"

]

# Filters our dataframe based on provided query conditions

filtered = extracted.query(" and ".join(queryConditions)).reset\_index(drop=True)

filtered['yearCentered'] = filtered['year'] - filtered['year'].mean()

filtered

# # 2. Intersecting with imagery

# Reading in our florence study area

# In[6]:

florence = gpd.read\_parquet(r"D:\IceSat\Final\mapping\ALSdata\dissolvedProjects.parquet")

florence

# Reading all filtered granules. It will be larger than ~52,000 as its not forest filtered

# # First intersecting with Level 4 EPA ecoregions

# In[7]:

l4eco = gpd.read\_file(r"D:\IceSat\Final\mapping\shapefiles\us\_eco\_l4\_state\_boundaries\Level4Ecoregions.shp")

l4eco

# Intersecting all segment centroids in the loblolly region with the EPA level 2 ecoregions

# In[8]:

florenceCentroidsWithEco = gpd.sjoin(filtered, l4eco[['geometry','NA\_L2NAME', 'NA\_L3NAME', 'US\_L4NAME']], how='left', predicate='within').drop('index\_right', axis=1)

florenceCentroidsWithEco.rename(columns={'NA\_L2NAME': 'Level2Ecoregion', 'NA\_L3NAME': 'Level3Ecoregion', 'US\_L4NAME': 'Level4Ecoregion'}, inplace=True)

florenceCentroidsWithEco

# # Function to intersect raster with segments

# In[9]:

def intersectWithRaster(dataframe, rasterPath, columnName, adjustmentValue):

    print(f'intersecting with {rasterPath}')

    with rasterio.open(rasterPath) as raster:

        if dataframe.crs != raster.crs:

            print('CRS mismatch')

            return None

        else:

            coordinates = [(x, y) for x, y in zip(dataframe["geometry"].x, dataframe["geometry"].y)]

            print('sampling coords')

            dataframe[f"{columnName}"] = [val[0] if val else None for val in raster.sample(coordinates)]

            if adjustmentValue:

                print('adjusting raster values by', adjustmentValue)

                dataframe[f"{columnName}"] = dataframe[f"{columnName}"].apply(lambda x: x + adjustmentValue if pd.notna(x) else x)

            dataframe[f"{columnName}"] = dataframe[f"{columnName}"].astype("Int64")

    print('done')

    return dataframe

# # First, intersecting with most recent fast loss

# In[10]:

florenceWithFastLoss = intersectWithRaster(florenceCentroidsWithEco,

                                     rasterPath = r"D:\IceSat\Final\mapping\imagery\LCMSLandChange\FastLossExtracted.tif",

                                     columnName = 'FastLossYear',

                                     adjustmentValue = False)

florenceWithFastLoss['FastLossYear'] = florenceWithFastLoss['FastLossYear'] + 1970

florenceWithFastLoss

# In[15]:

florenceWithFastLoss['FastLossYear'].value\_counts()

# Assigning undisturbed cells, removing disturbances after 2016. \

# \*\*\*Update 03/10 this was changed to include the year of ICESat-2 collection\*\*\*

# In[16]:

def groupFastLossYear(row):

    if row['FastLossYear'] == 1842:

        return 'Undisturbed'

    elif row['FastLossYear'] > row['year']:

        return 'After Collection'

    else:

        age = row['year'] - row['FastLossYear']

        if age <= 3:

            return '0-3 Years'

        elif age <= 10:

            return '<10 Years'

        elif age <= 20:

            return '10-20 Years'

        elif age <= 30:

            return '20-30 Years'

        else:

            return '30+ Years'

florenceWithFastLossAgeGroup = florenceWithFastLoss.assign(FastLossAgeGroup=florenceWithFastLoss.apply(groupFastLossYear, axis=1))

florenceWithFastLossAgeGroup['FastLossAgeGroup'].value\_counts()

# In[17]:

florenceWithFastLossAgeGroup['DisturbancePresence'] = np.where(

    florenceWithFastLossAgeGroup['FastLossAgeGroup'].isin(['After Collection', '0-3 Years', 'Undisturbed']),

    'No',

    'Yes'

)

florenceWithFastLossAgeGroup['DisturbancePresence'].value\_counts()

# # Assigning NLCD landcover

# In[18]:

florenceWithNLCD = intersectWithRaster(florenceWithFastLossAgeGroup,

                                      rasterPath = r"D:\IceSat\Final\mapping\imagery\NLCDLandCover\NLCDExtracted.tif",

                                      columnName = 'NLCD\_2023',

                                      adjustmentValue = False)

florenceWithNLCD['NLCD\_2023\_Label'] = florenceWithNLCD['NLCD\_2023'].map({41: 'Deciduous Forest', 42: 'Evergreen Forest', 43: 'Mixed Forest', 90: 'Woody Wetlands'})

florenceWithNLCD

# Assigning Flood Mask

# In[19]:

florenceWithFlooding = intersectWithRaster(florenceWithNLCD,

                                           rasterPath = r"D:\IceSat\Final\mapping\imagery\FlorenceFloodExtent\FlorenceFloodExtentReprojected.tif",

                                           columnName = 'FloodBinary',

                                           adjustmentValue = False)

florenceWithFlooding['FloodImpact'] = florenceWithFlooding['FloodBinary'].apply(

    lambda x: 'Unaffected' if pd.isna(x) else 'Affected' if x == 1 else None

)

florenceWithFlooding

# In[20]:

florenceWithFlooding['FloodImpact'].value\_counts()

# Recoding nlcd, year, and flooding impact

# In[21]:

forestCentroids = florenceWithFlooding[florenceWithFlooding["NLCD\_2023"].isin([41, 42, 43, 90])].reset\_index(drop=True)

print(len(forestCentroids))

forestCentroids.head()

# In[22]:

forestCentroids['DisturbancePresence'].value\_counts()

# # Writing out the forest centroids within the study area

# In[23]:

forestCentroids.to\_parquet(r"D:\IceSat\Final\segments\workingSetCentroids.parquet")

forestCentroids.to\_file(r"D:\IceSat\Final\segments\workingSetCentroids.GeoJSON")

# In[ ]:

* 1. **Python Script to Create Polygons of ATL08 Segments**

#!/usr/bin/env python

# coding: utf-8

# In[2]:

from osgeo import gdal

import numpy as np

import geopandas as gpd

import pandas as pd

import rasterio

import os

import dask.dataframe as dd

from tqdm.notebook import trange, tqdm

import glob

from pystac import Catalog, get\_stac\_version

import pystac

from pystac\_client import Client

import colorcet as cc

pd.set\_option('display.max\_columns', None)

import pystac\_client

from pystac import CatalogType

import fiona

import shapely

from sklearn.linear\_model import LinearRegression

import copy

from shapely.geometry import Point, LineString, Polygon, MultiPolygon, Polygon

from shapely.affinity import rotate

projectCRS = 32119

rasters = []

# Reading in our EPT projects to join with centroids

# In[3]:

undissolvedProjects = gpd.read\_parquet(r"D:\IceSat\Final\mapping\ALSdata\undissolvedProjects.parquet")

undissolvedProjects = undissolvedProjects.rename(columns={'id':'projectID'})

undissolvedProjects

# Reading in Florence centroids

# In[4]:

workingSet = gpd.read\_parquet(r"D:\IceSat\Final\segments\workingSetCentroids.parquet")

print(len(workingSet))

workingSet.head()

# In[5]:

eastCoastEnvelope = gpd.read\_file(r"D:\IceSat\V12\mapping\data\roughEastCoastEnvelope.shp")

if eastCoastEnvelope.crs != projectCRS:

    print('reprojecting envelope')

    eastCoastEnvelope = eastCoastEnvelope.to\_crs(projectCRS)

allGroundTrackKMLs = []

parentDir = r"D:\IceSat\ReferenceGroundTrackKMLs"

for root, dirs, files in os.walk(parentDir):

    for file in files:

        if file.endswith(".kml"):

            allGroundTrackKMLs.append(os.path.join(root, file))

def determineAngleFromKML(dataframe):

    dataframe = dataframe.sort\_values('datetime')

    cycleNumber = dataframe['start\_cycle'].iloc[0]

    rgtNumber = str(dataframe['rgt'].iloc[0]).zfill(4)

    # Determining relevant kml file to read from

    matchingFile = next((file for file in allGroundTrackKMLs if f'RGT\_{rgtNumber}\_cycle{cycleNumber}' in os.path.basename(file)), None)

    # print('Cycle:', cycleNumber, 'RGT:', rgtNumber, os.path.basename(matchingFile))

    kml = gpd.read\_file(matchingFile, driver='libKML')

    # # Creating a dataframe for our kml geometry and renaming columns

    kmlDf = pd.DataFrame(kml.geometry[0].coords)

    kmlDf.columns = ['longitude', 'latitude', 'z']

    # # Converting dataframe to a geodataframe and clipping to our kentucky intersection

    kmlGdf = gpd.GeoDataFrame(

        kmlDf, geometry=gpd.points\_from\_xy(kmlDf.longitude, kmlDf.latitude), crs="EPSG:4326"

    ).to\_crs(projectCRS).clip(eastCoastEnvelope)

    kmlGdf = kmlGdf.to\_crs(32119)

    # Getting first row, i.e., the first moment the track passes through the east coast

    print('LENGTH:',len(kmlGdf))

    firstRow = kmlGdf.iloc[0]

    # Getting last row, i.e., the last moment the track passes through the east coast

    lastRow = kmlGdf.iloc[-1]

    # VERTICAL DIRECTION

    # moving south to north

    if firstRow['latitude'] < lastRow['latitude']:

        # print('Moving SOUTH to NORTH, bottom to top')

        #HORIZONTAL DIRECTION

        # the earliest segment is more WESTWARD than the last, platform moving SW to NE

        if firstRow['longitude'] < lastRow['longitude']:

            # return ('rotation should be +2')

            return -2

        # HORIZONTAL DIRECTION

        # ealiest segment is more EASTWARD than the last, platform is moving SE to NW

        elif firstRow['longitude'] > lastRow['longitude']:

            # primt('rotation should be -2')

            return 2

    # VERTICAL DIRECTION

    # moving north to south

    elif firstRow['latitude'] > lastRow['latitude']:

        # print('Moving NORTH to SOUTH, top to bottom\n')

        # HORIZONTAL DIRECTION

        # earliest segment is more WESTWARD than last, platform is moving NW to SE

        if firstRow['longitude'] < lastRow['longitude']:

            return 2

        # HORIZONTAL DIRECTION

        # earliest segment is more EASTWARD than last, platform is moving NE to SW

        elif firstRow['longitude'] > lastRow['longitude']:

            return -2

def createSegments(geodataframe, cycleRGT):

    angle = determineAngleFromKML(geodataframe)

    geodataframe['geometry'] = geodataframe.apply(lambda row: shapely.box(row['geometry'].x - 5.5, row['geometry'].y - 50, row['geometry'].x + 5.5, row['geometry'].y + 50), axis=1)

    # now we assign the previously determined rotaion angle to a column value

    geodataframe['rotation\_angle'] = angle

    # finally, rotating each row's polygon geometry by the corresponding value in the rotation\_angle column

    geodataframe['geometry'] = geodataframe.apply(lambda row: shapely.affinity.rotate(row['geometry'], row['rotation\_angle'], origin=(row['geometry'].centroid.x, row['geometry'].centroid.y)), axis=1)

    return geodataframe

def createPolygons(dataframe, geosegmentFlag):

    # Making a dictionary to hold a dataframe for each unique combination of ICESat-2 cycle and reference ground track, there's 143 unique combinations of these variables

    geosegmentDict = {}

    dataframesWithPolygons = []

    for (startCycle, rgt), cycleRGTDataframe in dataframe.groupby(['start\_cycle', 'rgt']):

        geosegmentDict[(startCycle, rgt)] = cycleRGTDataframe

        print('\nCreating segments for cycle:', startCycle, 'rgt:', rgt, 'dataframe length', len(cycleRGTDataframe))

        cycleRGT = f"{startCycle},{rgt}"

        dfWithPolygons = createSegments(cycleRGTDataframe, cycleRGT)

        dataframesWithPolygons.append(dfWithPolygons)

        # print('Done creating segments\n')

    # # Stacking them vertically after into one final dataframe

    finalDataframe = pd.concat(dataframesWithPolygons, ignore\_index=True)

    if geosegmentFlag:

        finalDataframe = pivotToGeosegments(finalDataframe)

    return finalDataframe

polygons = createPolygons(workingSet, geosegmentFlag=False)

polygons

# In[6]:

polygonsWithEPT = polygons.sjoin(undissolvedProjects[['geometry']], how='left', predicate='intersects')

polygonsFull = polygonsWithEPT.merge(undissolvedProjects[['asset', 'projectID']], left\_on='index\_right', right\_index=True, how='left').reset\_index(drop=True)

polygonsFull['outputID'] = polygonsFull['segmentUID'] + polygonsFull['projectID']

polygonsFull

# In[7]:

polygonsFull.to\_parquet(r"D:\IceSat\Final\segments\workingSetPolygonsWITHOUTALSMetrics.parquet")

polygonsFull.to\_file(r"D:\IceSat\Final\segments\workingSetPolygonsWITHOUTALSMetrics.GeoJSON")

* 1. **R Script for Analyses of Phases 1, 2, 4, and 5**

---

title: "FinalAnalysis"

format: html

editor: visual

---

# Libraries

```{r setup, warning=FALSE, message=FALSE}

library(tidyverse)

library(RobustLinearReg)

library(caret)

library(mgcv)

library(arrow)

library(dplyr)

library(cowplot)

library(broom)

library(glue)

library(trend)

library(TOSTER)

```

# Reading Data

```{r}

# Our working set: polygons for all years of ICESat-2 data with > 3 years recovery, or disturbance before icesat-2 collection

workingSet <- read\_parquet("D:/IceSat/Final/segments/newWorkingSetPolygonsWITHALSMetrics.parquet") |>

filter(!(FastLossAgeGroup %in% c("0-3 Years", "After Collection")))

# Our validation set of only polygons collected in 2020

validationSet <- workingSet |>

filter(year == 2020)

# removing 2019 from consideration for spatial biases

filteredSet <- workingSet |>

filter(!(year==2019))

# sample of 50K forest cells which have ALS metrics

fiftyThousandSample <- read\_parquet("D:/IceSat/Final/testing/fiftyThousandSamplePolygonsWithALSMetrics.parquet")

```

# Global Variables

```{r}

# Plotting theme

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.text = element\_text(size=12, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

# Number of iterations

numberOfIterations <- 10000

```

# 1: Validation

## Functions

### Regression

This correlation ATL08 segments in 2020, regardless of forest cover type or disturbance history.

```{r}

validationTSRModel <- theil\_sen\_regression(h\_canopy ~ percentile\_98, data=validationSet)

validationTSRPredictions <- predict(validationTSRModel)

validationTSRrSquared <- summary(validationTSRModel)$r.squared

validationTSRRMSE <- sqrt(mean(validationTSRModel$residuals^2))

summary(validationTSRModel)

```

### Correlation plot

```{r}

mainCorrelationPlot <- ggplot(validationSet, aes(x = percentile\_98, y = h\_canopy)) +

geom\_point(color = "black", size = 1.25, alpha = 0.2) +

geom\_line(aes(y = validationTSRPredictions), color = 'red', linewidth = 1) +

labs(

title = "Correlation of 2020 ALS & ATL08 Canopy Heights",

subtitle = glue("R²: {round(validationTSRrSquared, 2)} RMSE: {round(validationTSRRMSE, 2)}m"),

x = "ALS Canopy Height (m)",

y = "ATL08 Canopy Height (m)",

# colour = "Forest Cover Type"

) +

customTheme +

guides(color = guide\_legend(override.aes = list(size = 10, color = "white")))

ggsave(

"D:/IceSat/writing/thesis/figures/o1\_main\_plot.png",

plot = mainCorrelationPlot,

bg = "white",

width = 8, height = 5, units = "in", dpi = 300

)

mainCorrelationPlot

```

# 2: Bias Assessment

## Recoding ALS Groups

```{r}

workingSet <- workingSet %>%

mutate(yearRecoded = recode\_factor(

year,

`2019` = "A",

`2020` = "B",

`2021` = "C",

`2022` = "D",

`2023` = "E",

`2024` = "F"

))

```

## ATL08 Data

### Year Histograms

```{r}

alsHeightsYearSummary <- workingSet %>%

group\_by(yearRecoded) %>%

summarize(medianALSHeight = median(percentile\_98, na.rm=TRUE), meanALSHeight = mean(percentile\_98, na.rm = TRUE))

fullALSHist <- ggplot(workingSet, aes(x = percentile\_98, color = as.factor(yearRecoded), fill = as.factor(yearRecoded))) +

geom\_density(size = 1, alpha = 0.1) +

geom\_vline(data = alsHeightsYearSummary, aes(xintercept = medianALSHeight, color = as.factor(yearRecoded)),

linetype = "dashed", size = 1) +

labs(title = glue("Density of 2020 ALS Canopy Heights by Group"),

x = "ALS Canopy Height (m)",

y = "Density",

color = "Group",

fill = "Group") +

theme\_minimal()+

customTheme

ggsave("D:/IceSat/writing/thesis/figures/o2\_year\_KDE\_plot.png", plot = fullALSHist, bg = "white", width = 8, height = 5, units = "in", dpi = 300)

fullALSHist

```

### Equivalence Grid

#### Bootstrap TOST

```{r}

unique\_years <- sort(unique(workingSet$yearRecoded))

allPairResults <- list()

equivBounds <- c(2.5, 2, 1.5, 1, 0.5, 0.25, 0.1)

# The outer loop and inner loops combine to get the unique pairs of years

# Outer loop

for (i in 1:(length(unique\_years)-1)) {

# Inner loop

for (j in (i+1):length(unique\_years)) {

# Assigning years

year1 <- unique\_years[i]

year2 <- unique\_years[j]

# Print after defining year1 and year2

print(glue("\n\nyear 1: {year1}, year 2: {year2}"))

# Subsetting our working set to only year1 or year2

currentPairSample <- workingSet |>

filter(yearRecoded %in% c(year1, year2))

print(glue("length of current df {nrow(currentPairSample)}"))

# Finally, the innermost loop of our equivalence bounds, the years and sample won't change across eqb values

currentPairResults <- list()

for(currentBound in equivBounds){

print(glue("Testing {year1} against {year2} with an EQB of: {currentBound}\n"))

# Running TOST with the current year pair and current EQB

currentBoundTOST <- boot\_t\_TOST(formula = percentile\_98 ~ as.factor(yearRecoded),

data = currentPairSample,

paired = FALSE,

var.equal = FALSE,

eqb = currentBound,

eqbound\_type="raw"

)

currentBoundBootResults <- data.frame(

yearA = year1,

yearB = year2,

equivalenceBound = currentBound,

pairLength = nrow(currentPairSample),

lower\_p\_value = currentBoundTOST$TOST$p.value[2],

upper\_p\_value = currentBoundTOST$TOST$p.value[3],

result = currentBoundTOST$decision$combined,

lower\_t\_value = currentBoundTOST$TOST$t[2],

upper\_t\_value = currentBoundTOST$TOST$t[3],

lower\_raw\_ci = currentBoundTOST$effsize$lower.ci[1],

upper\_raw\_ci = currentBoundTOST$effsize$upper.ci[1]

)

# appending results of current EQB to full list for current year pair

currentPairResults[[length(currentPairResults) + 1]] <- currentBoundBootResults

}

# stacking all EQB results for current year pair

combinedCurrentPairResults <- bind\_rows(currentPairResults)

# then adding all EQB results of current year pair to list of all year pairs

allPairResults[[length(allPairResults) + 1]] <- combinedCurrentPairResults

}

}

# Combine all results into a dataframe

finalTOSTResults <- bind\_rows(allPairResults)

# View the final dataframe

finalTOSTResults

```

#### Bonferroni Correction

```{r}

# recalculating bonferroni corrected p-value

bonferroniPValue <- 0.05 / 15

finalTOSTGrid <- finalTOSTResults |>

mutate(Decision = if\_else(lower\_p\_value < bonferroniPValue & upper\_p\_value < bonferroniPValue, "Yes", "No")) |>

select(yearA, yearB, Decision, equivalenceBound)

```

#### Charting Bootstrapt EQT

```{r}

allEQBGrid <- finalTOSTGrid |>

filter(equivalenceBound != 2.5) |>

ggplot(aes(yearA, yearB, fill= as.factor(Decision))) +

# geom\_tile()+

geom\_tile(color = "lightgray", size = 0.3) +

facet\_wrap(~equivalenceBound)+

# theme\_minimal()+

scale\_fill\_manual(values = c("Yes" = "darkgreen", "No" = "red")) +

labs(title = glue("ALS Group Equivalence by Bound (Meters)"),

subtitle = glue("Bonferroni Corrected p-value: {round(bonferroniPValue, 4)}"),

x = "Group 1",

y = "Group 2",

fill = "Equivalent?") +

customTheme+

theme(

axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

)

allEQBGrid

ggsave("D:/IceSat/writing/thesis/figures/o2\_ALS\_EQB\_Grid.png", plot = allEQBGrid, bg = "white", width = 8, height = 5, units = "in", dpi = 300)

minTOSTGrid <- finalTOSTGrid %>%

group\_by(yearA, yearB) %>%

summarise(minEquivalence = min(equivalenceBound[Decision == 'Yes'], na.rm = TRUE)) %>%

ungroup()

minTOSTGrid

```

## 50K Sample

### ALS Historgrams

```{r}

fiftyThousandSampleHeightsYearSummary <- fiftyThousandSample %>%

group\_by(yearCategory) %>%

summarize(medianALSHeight = median(percentile\_98, na.rm=TRUE), meanALSHeight = mean(percentile\_98, na.rm = TRUE))

fiftyThousandSamplefullALSHist <- ggplot(fiftyThousandSample, aes(x = percentile\_98, color = as.factor(yearCategory), fill = as.factor(yearCategory))) +

geom\_density(size = 1, alpha = 0.1) +

geom\_vline(data = fiftyThousandSampleHeightsYearSummary, aes(xintercept = medianALSHeight, color = as.factor(yearCategory)),

linetype = "dashed", size = 1) +

labs(title = glue("Density of 2020 ALS Canopy Heights by Group"),

x = "ALS Canopy Height (m)",

y = "Density",

color = "Group",

fill = "Group") +

theme\_minimal()+

customTheme

ggsave("D:/IceSat/writing/thesis/figures/o2\_sample\_yearCategory\_KDE\_plot.png", plot = fiftyThousandSamplefullALSHist, bg = "white", width = 8, height = 5, units = "in", dpi = 300)

fiftyThousandSamplefullALSHist

```

### Equivalence Grid

#### Bootstrap TOST

```{r}

unique\_yearCategories <- sort(unique(fiftyThousandSample$yearCategory))

allPairResults <- list()

equivBounds <- c(2.5, 2, 1.5, 1, 0.5, 0.25, 0.1)

# The outer loop and inner loops combine to get the unique pairs of years

# Outer loop

for (i in 1:(length(unique\_yearCategories)-1)) {

# Inner loop

for (j in (i+1):length(unique\_yearCategories)) {

# Assigning years

year1 <- unique\_yearCategories[i]

year2 <- unique\_yearCategories[j]

# Print after defining year1 and year2

print(glue("\n\nyear 1: {year1}, year 2: {year2}"))

# Subsetting our working set to only year1 or year2, not sampling cause boot\_TOST does it for us

currentPairSample <- fiftyThousandSample |>

filter(yearCategory %in% c(year1, year2))

print(glue("length of current df {nrow(currentPairSample)}"))

# Finally, the innermost loop of our equivalence bounds, the years and sample won't change across eqb values

currentPairResults <- list()

for(currentBound in equivBounds){

print(glue("Testing {year1} against {year2} with an EQB of: {currentBound}\n"))

# Running TOST with the current year pair and current EQB

currentBoundTOST <- boot\_t\_TOST(formula = percentile\_98 ~ as.factor(yearCategory),

data = currentPairSample,

paired = FALSE,

var.equal = FALSE,

eqb = currentBound,

eqbound\_type="raw"

)

currentBoundBootResults <- data.frame(

yearA = year1,

yearB = year2,

equivalenceBound = currentBound,

pairLength = nrow(currentPairSample),

lower\_p\_value = currentBoundTOST$TOST$p.value[2],

upper\_p\_value = currentBoundTOST$TOST$p.value[3],

result = currentBoundTOST$decision$combined,

lower\_t\_value = currentBoundTOST$TOST$t[2],

upper\_t\_value = currentBoundTOST$TOST$t[3],

lower\_raw\_ci = currentBoundTOST$effsize$lower.ci[1],

upper\_raw\_ci = currentBoundTOST$effsize$upper.ci[1]

)

# appending results of current EQB to full list for current year pair

currentPairResults[[length(currentPairResults) + 1]] <- currentBoundBootResults

}

# stacking all EQB results for current year pair

combinedCurrentPairResults <- bind\_rows(currentPairResults)

# then adding all EQB results of current year pair to list of all year pairs

allPairResults[[length(allPairResults) + 1]] <- combinedCurrentPairResults

}

}

# Combine all results into a dataframe

finalSampleTOSTResults <- bind\_rows(allPairResults)

# View the final dataframe

finalSampleTOSTResults

```

#### Bonferroni Correction

```{r}

bonferroniPValue <- 0.05 / 15

finalSampleTOSTGrid <- finalSampleTOSTResults |>

mutate(Decision = if\_else(lower\_p\_value < bonferroniPValue & upper\_p\_value < bonferroniPValue, "Yes", "No")) |>

select(yearA, yearB, Decision, equivalenceBound)

```

#### Charting bootstrap EQT

```{r}

sampleAllEQBGrid <- finalSampleTOSTGrid |>

filter(equivalenceBound != 2.5) |>

ggplot(aes(yearA, yearB, fill= as.factor(Decision))) +

# geom\_tile()+

geom\_tile(color = "lightgray", size = 0.3) +

facet\_wrap(~equivalenceBound)+

# theme\_minimal()+

scale\_fill\_manual(values = c("Yes" = "darkgreen", "No" = "red")) +

labs(title = glue("ALS Group Equivalence by Bound (Meters)"),

subtitle = glue("Bonferroni Corrected p-value: {round(bonferroniPValue, 4)}"),

x = "Group 1",

y = "Group 2",

fill = "Equivalent?") +

customTheme+

theme(

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # angled axis values for legibility

)

sampleAllEQBGrid

ggsave("D:/IceSat/writing/thesis/figures/o2\_ALS\_Sample\_EQB\_Grid.png", plot = sampleAllEQBGrid, bg = "white", width = 8, height = 5, units = "in", dpi = 300)

minSampleTOSTGrid <- finalSampleTOSTGrid %>%

group\_by(yearA, yearB) %>%

summarise(minEquivalence = min(equivalenceBound[Decision == 'Yes'], na.rm = TRUE)) %>%

ungroup()

minSampleTOSTGrid

```

# 3: Reference Growth

## Alternate Script

# 4: Change Quantification

Section 4 will have 2019 data included, section 5 will not \## Functions \### TSR by Factor

```{r}

tsrByfactor <- function(dataframe, factor, factorLevel, heightsColumn){

print(glue("Running theil-sen regression for {factor}: {factorLevel}"))

currentLevelDataframe <- dataframe |>

arrange(.data[[factor]]) |>

filter(.data[[factor]] == factorLevel)

allTSRResults = list()

targetN <- min(500, nrow(currentLevelDataframe))

for(currentIteration in 1:numberOfIterations){

if (currentIteration %% 1000 == 0) {

print(glue("Current iteration: {currentIteration}"))

}

currentIterationSample <- sample\_n(currentLevelDataframe, targetN, replace=TRUE)

tsrModel <- theil\_sen\_regression(reformulate("yearCentered", response = heightsColumn), data = currentIterationSample)

modelIterationSummary <- tidy(tsrModel)

modelIterationSummary$iteration <- currentIteration

modelIterationSummary$sampleSize <- targetN

modelIterationSummary$factor <- factor

modelIterationSummary$factorLevel <- factorLevel

allTSRResults[[currentIteration]] <- modelIterationSummary

}

allTSRResultsDf <- do.call(rbind, allTSRResults)

# Aggregate results

aggregatedResults <- allTSRResultsDf %>%

group\_by(term) %>%

summarize(

factor = unique(factor),

factorLevel = unique(factorLevel),

# mean\_estimate = mean(estimate, na.rm = TRUE),

median\_estimate = median(estimate, na.rm = TRUE),

# mean\_p\_value = mean(p.value, na.rm = TRUE),

median\_p\_value = median(p.value, na.rm = TRUE)

)

return(aggregatedResults)

}

```

### Net T-Test

```{r}

calculate\_cohens\_d <- function(x, y) {

n1 <- length(x)

n2 <- length(y)

sd1 <- sd(x, na.rm = TRUE)

sd2 <- sd(y, na.rm = TRUE)

pooled\_sd <- sqrt(((n1 - 1) \* sd1^2 + (n2 - 1) \* sd2^2) / (n1 + n2 - 2))

if (pooled\_sd > 0) {

return((mean(y, na.rm = TRUE) - mean(x, na.rm = TRUE)) / pooled\_sd)

} else {

return(NA)

}

}

netTByFactor <- function(dataframe, factor, level, heightsColumn, firstYear, lastYear){

years <-

# Pulling heights from our first year of data

firstYearHeights <- dataframe |>

filter(year == firstYear) |>

filter(.data[[factor]] == level) |>

pull(.data[[heightsColumn]])

# Pulling heights from our last year of data

lastYearHeights <- dataframe |>

filter(year == lastYear) |>

filter(.data[[factor]] == level) |>

pull(.data[[heightsColumn]])

allTTestResults <- data.frame(

t\_statistic = numeric(),

p\_value = numeric(),

mean\_diff = numeric(),

lower\_ci = numeric(),

upper\_ci = numeric(),

cohens\_d = numeric()

)

targetN <- min(500, min(length(firstYearHeights), length(lastYearHeights)))

# 10000 iterations of 500 samples seems fine

for(currentIteration in 1:numberOfIterations){

if (currentIteration %% 1000 == 0) {

print(glue("Current iteration: {currentIteration}"))

}

firstYearSample <- sample(firstYearHeights, targetN, replace=TRUE)

lastYearSample <- sample(lastYearHeights, targetN, replace=TRUE)

tTest <- wilcox.test(lastYearSample, firstYearSample, alternative='greater')

# here gotta calculate cohen's d between the first year and last year heights

cohens\_d\_value <- calculate\_cohens\_d(firstYearSample, lastYearSample)

# storing results into df

allTTestResults <- rbind(allTTestResults, data.frame(

t\_statistic = tTest$statistic,

p\_value = tTest$p.value,

mean\_diff = mean(lastYearSample) - mean(firstYearSample),

cohens\_d = cohens\_d\_value

))

}

# then aggregating results for more meaninful stats

aggregatedResults <- allTTestResults %>%

summarize(

sampleSize = targetN,

startYear = firstYear,

endYear = lastYear,

firstYearSamples = length(firstYearHeights),

lastYearSamples = length(lastYearHeights),

# mean\_t\_statistic = mean(t\_statistic, na.rm = TRUE),

median\_t\_statistic = median(t\_statistic, na.rm = TRUE),

# mean\_p\_value = mean(p\_value, na.rm = TRUE),

median\_p\_value = median(p\_value, na.rm = TRUE),

# mean\_diff = mean(mean\_diff, na.rm = TRUE),

median\_mean\_diff = median(mean\_diff, na.rm = TRUE),

# mean\_cohens\_d = mean(cohens\_d, na.rm = TRUE),

median\_cohens\_d = median(cohens\_d, na.rm = TRUE)

)

return(aggregatedResults)

}

```

### GAM

```{r}

gamByFactor <- function(dataframe, categoryVariable, heightsColumn, kValue){

gamFormula <- as.formula(glue("{heightsColumn} ~ s(yearCentered, by = as.factor({categoryVariable}), k = {kValue}) + as.factor({categoryVariable})"))

gamFit <- tryCatch({

gam(gamFormula, data = dataframe, method = "REML")

}, error = function(e) NULL)

gamSummary <- summary(gamFit)

edfData <- data.frame(

category = rownames(gamSummary$s.table),

edf = gamSummary$s.table[, "edf"],

pval = gamSummary$s.table[,"p-value"],

devianceExplained = gamSummary$dev.expl,

baseline = gamSummary$p.coeff

)

return(edfData)

}

bootStrapGam <- function(df, categoryVar, heightsCol, kVal){

df <- df %>% arrange(.data[[categoryVar]])

targetN <- min(500, nrow(df))

allIterationResults = list()

for(currentIteration in 1:numberOfIterations){

if (currentIteration %% 1000 == 0) {

print(glue("Current iteration: {currentIteration}"))

}

currentIterationSample <- sample\_n(df, targetN, replace=TRUE)

currentIterationResult <- gamByFactor(currentIterationSample, categoryVar, heightsCol, kVal)

allIterationResults[[currentIteration]] <- currentIterationResult

}

allResultsDf <- do.call(rbind, allIterationResults)

aggregatedResults <- allResultsDf %>%

group\_by(category) %>%

summarize(

# mean\_edf = mean(edf, na.rm = TRUE),

median\_edf = median(edf, na.rm = TRUE),

# mean\_pval = mean(pval, na.rm = TRUE),

median\_pval = median(pval, na.rm = TRUE),

# mean\_devianceExplained = mean(devianceExplained, na.rm = TRUE),

median\_devianceExplained = median(devianceExplained, na.rm = TRUE),

# mean\_baseline = mean(baseline, na.rm = TRUE),

median\_baseline = median(baseline, na.rm = TRUE)

)

return(aggregatedResults)

}

# samp <- workingSet |> sample\_n(4000)

# gam(h\_canopy ~ s(yearCentered, by=as.factor(NLCD\_2023\_Label)), data=samp, method="REML")

```

### MK Test

```{r}

mkTestByFactor <- function(dataframe, factor, level, heightsColumn){

# Pulling canopy heights chronologically sorted for given factor

heightsSorted <- dataframe |>

filter(.data[[factor]] == level) |>

arrange(datetime) |>

pull(.data[[heightsColumn]])

allMKResults <- data.frame(

sampleSize = numeric(),

z\_statistic = numeric(),

p\_value = numeric(),

S = numeric(),

varS = numeric(),

tau = numeric()

)

targetN <- min(500, length(heightsSorted))

# 10000 iterations of bootstrapping

for(currentIteration in 1:numberOfIterations){

if (currentIteration %% 1000 == 0) {

print(glue("Current iteration: {currentIteration}"))

}

# Sample indices sort them to preserve time order

indices <- sort(sample(seq\_along(heightsSorted), targetN, replace = TRUE))

bootstrapSample <- heightsSorted[indices]

mk <- mk.test(bootstrapSample, alternative = "greater")

# add sen's slope here

# Store the results

allMKResults <- rbind(allMKResults, data.frame(

sampleSize = targetN,

z\_statistic = as.numeric(mk$statistic),

p\_value = as.numeric(mk$p.value),

S = as.numeric(mk$estimates["S"]),

varS = as.numeric(mk$estimates["varS"]),

tau = as.numeric(mk$estimates["tau"])

))

}

aggregatedResults <- allMKResults %>%

summarize(

sampleSize = targetN,

# mean\_z\_statistic = mean(z\_statistic, na.rm = TRUE),

median\_z\_statistic = median(z\_statistic, na.rm = TRUE),

# mean\_p\_value = mean(p\_value, na.rm = TRUE),

median\_p\_value = median(p\_value, na.rm = TRUE),

# mean\_s = mean(S, na.rm = TRUE),

median\_s = median(S, na.rm = TRUE),

# mean\_varS = mean(varS, na.rm = TRUE),

median\_varS = median(varS, na.rm = TRUE),

# mean\_tau = mean(tau, na.rm=TRUE),

median\_tau = median(tau, na.rm=TRUE)

)

return(aggregatedResults)

}

```

## 4.1 TSR Regression Models

### By forest cover

Deciduous

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Deciduous Forest", "h\_canopy")

```

Evergreen

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Evergreen Forest", "h\_canopy")

```

Mixed

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Mixed Forest", "h\_canopy")

```

Woody Wetlands

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Woody Wetlands", "h\_canopy")

```

### By disturbance presence

Yes

```{r}

tsrByfactor(workingSet, "DisturbancePresence", "Yes", "h\_canopy")

```

No

```{r}

tsrByfactor(workingSet, "DisturbancePresence", "No", "h\_canopy")

```

### By disturbance age

\< 10 Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "<10 Years", "h\_canopy")

```

10-20 Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "10-20 Years", "h\_canopy")

```

20-30 Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "20-30 Years", "h\_canopy")

```

30+ Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "30+ Years", "h\_canopy")

```

## 4.2 Net change t-test

### By forest cover

Deciduous

```{r}

netTByFactor(workingSet, "NLCD\_2023\_Label", "Deciduous Forest", "h\_canopy", 2019, 2024)

```

Evergreen

```{r}

netTByFactor(workingSet, "NLCD\_2023\_Label", "Evergreen Forest", "h\_canopy", 2019, 2024)

```

Mixed

```{r}

netTByFactor(workingSet, "NLCD\_2023\_Label", "Mixed Forest", "h\_canopy", 2019, 2024)

```

Woody Wetlands

```{r}

netTByFactor(workingSet, "NLCD\_2023\_Label", "Woody Wetlands", "h\_canopy", 2019, 2024)

```

### By disturbance presence

Yes

```{r}

netTByFactor(workingSet, "DisturbancePresence", "Yes", "h\_canopy", 2019, 2024)

```

No

```{r}

netTByFactor(workingSet, "DisturbancePresence", "No", "h\_canopy", 2019, 2024)

```

### By disturbance age

\< 10 Years

```{r}

netTByFactor(workingSet, "FastLossAgeGroup", "<10 Years", "h\_canopy", 2019, 2024)

```

10-20 Years

```{r}

netTByFactor(workingSet, "FastLossAgeGroup", "10-20 Years", "h\_canopy", 2019, 2024)

```

20-30 Years

```{r}

netTByFactor(workingSet, "FastLossAgeGroup", "20-30 Years", "h\_canopy", 2019, 2024)

```

30+ Years

```{r}

netTByFactor(workingSet, "FastLossAgeGroup", "30+ Years", "h\_canopy", 2019, 2024)

```

## 4.3 Generalized Additive Models

### By forest cover

```{r}

bootStrapGam(workingSet, 'NLCD\_2023\_Label', 'h\_canopy', 5)

```

### By disturbance presence

```{r}

bootStrapGam(workingSet, 'DisturbancePresence', 'h\_canopy', 5)

```

### By disturbance age

```{r}

bootStrapGam(workingSet, 'FastLossAgeGroup', 'h\_canopy', 5)

```

## 4.4 Mann Kendall Test

### By forest cover

Deciduous

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Deciduous Forest","h\_canopy")

```

Evergreen

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Evergreen Forest","h\_canopy")

```

Mixed

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Mixed Forest","h\_canopy")

```

Woody Wetlands

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Woody Wetlands","h\_canopy")

```

### By disturbance presence

Yes

```{r}

mkTestByFactor(workingSet, "DisturbancePresence", "Yes","h\_canopy")

```

No

```{r}

mkTestByFactor(workingSet, "DisturbancePresence", "No","h\_canopy")

```

### By disturbance age

\< 10 Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "<10 Years","h\_canopy")

```

10-20 Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "10-20 Years","h\_canopy")

```

20-30 Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "20-30 Years","h\_canopy")

```

30+ Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "30+ Years","h\_canopy")

```

# 5. Filtered Change Quantification

Section 5 will exclude 2019 data, so as to explore the impact of spatial biases on growth trends

## 5.1 TSR Regression Models

### By forest cover

Deciduous

```{r}

tsrByfactor(filteredSet, "NLCD\_2023\_Label", "Deciduous Forest", "h\_canopy")

```

Evergreen

```{r}

tsrByfactor(filteredSet, "NLCD\_2023\_Label", "Evergreen Forest", "h\_canopy")

```

Mixed

```{r}

tsrByfactor(filteredSet, "NLCD\_2023\_Label", "Mixed Forest", "h\_canopy")

```

Woody Wetlands

```{r}

tsrByfactor(filteredSet, "NLCD\_2023\_Label", "Woody Wetlands", "h\_canopy")

```

### By disturbance presence

Yes

```{r}

tsrByfactor(filteredSet, "DisturbancePresence", "Yes", "h\_canopy")

```

No

```{r}

tsrByfactor(filteredSet, "DisturbancePresence", "No", "h\_canopy")

```

### By disturbance age

\< 10 Years

```{r}

tsrByfactor(filteredSet, "FastLossAgeGroup", "<10 Years", "h\_canopy")

```

10-20 Years

```{r}

tsrByfactor(filteredSet, "FastLossAgeGroup", "10-20 Years", "h\_canopy")

```

20-30 Years

```{r}

tsrByfactor(filteredSet, "FastLossAgeGroup", "20-30 Years", "h\_canopy")

```

30+ Years

```{r}

tsrByfactor(filteredSet, "FastLossAgeGroup", "30+ Years", "h\_canopy")

```

## 5.2 Net change t-test

### By forest cover

Deciduous

```{r}

netTByFactor(filteredSet, "NLCD\_2023\_Label", "Deciduous Forest", "h\_canopy", 2020, 2024)

```

Evergreen

```{r}

netTByFactor(filteredSet, "NLCD\_2023\_Label", "Evergreen Forest", "h\_canopy", 2020, 2024)

```

Mixed

```{r}

netTByFactor(filteredSet, "NLCD\_2023\_Label", "Mixed Forest", "h\_canopy", 2020, 2024)

```

Woody Wetlands

```{r}

netTByFactor(filteredSet, "NLCD\_2023\_Label", "Woody Wetlands", "h\_canopy", 2020, 2024)

```

### By disturbance presence

Yes

```{r}

netTByFactor(filteredSet, "DisturbancePresence", "Yes", "h\_canopy", 2020, 2024)

```

No

```{r}

netTByFactor(filteredSet, "DisturbancePresence", "No", "h\_canopy", 2020, 2024)

```

### By disturbance age

\< 10 Years

```{r}

netTByFactor(filteredSet, "FastLossAgeGroup", "<10 Years", "h\_canopy", 2020, 2024)

```

10-20 Years

```{r}

netTByFactor(filteredSet, "FastLossAgeGroup", "10-20 Years", "h\_canopy", 2020, 2024)

```

20-30 Years

```{r}

netTByFactor(filteredSet, "FastLossAgeGroup", "20-30 Years", "h\_canopy", 2020, 2024)

```

30+ Years

```{r}

netTByFactor(filteredSet, "FastLossAgeGroup", "30+ Years", "h\_canopy", 2020, 2024)

```

## 5.3 Generalized Additive Models

### By land cover

```{r}

bootStrapGam(filteredSet, 'NLCD\_2023\_Label', 'h\_canopy', 5)

```

### By disturbance presence

```{r}

bootStrapGam(filteredSet, 'DisturbancePresence', 'h\_canopy', 5)

```

### By disturbance age

```{r}

bootStrapGam(filteredSet, 'FastLossAgeGroup', 'h\_canopy', 5)

```

## 5.4 Mann Kendall Test

### By forest cover

Deciduous

```{r}

mkTestByFactor(filteredSet, "NLCD\_2023\_Label", "Deciduous Forest","h\_canopy")

```

Evergreen

```{r}

mkTestByFactor(filteredSet, "NLCD\_2023\_Label", "Evergreen Forest","h\_canopy")

```

Mixed

```{r}

mkTestByFactor(filteredSet, "NLCD\_2023\_Label", "Mixed Forest","h\_canopy")

```

Woody Wetlands

```{r}

mkTestByFactor(filteredSet, "NLCD\_2023\_Label", "Woody Wetlands","h\_canopy")

```

### By disturbance presence

Yes

```{r}

mkTestByFactor(filteredSet, "DisturbancePresence", "Yes","h\_canopy")

```

No

```{r}

mkTestByFactor(filteredSet, "DisturbancePresence", "No","h\_canopy")

```

### By disturbance age

\< 10 Years

```{r}

mkTestByFactor(filteredSet, "FastLossAgeGroup", "<10 Years","h\_canopy")

```

10-20 Years

```{r}

mkTestByFactor(filteredSet, "FastLossAgeGroup", "10-20 Years","h\_canopy")

```

20-30 Years

```{r}

mkTestByFactor(filteredSet, "FastLossAgeGroup", "20-30 Years","h\_canopy")

```

30+ Years

```{r}

mkTestByFactor(filteredSet, "FastLossAgeGroup", "30+ Years","h\_canopy")

```

# 6. Additional Testing

## Coverage by L2E

```{r}

result\_list <- list()

# Loop through unique years

for (group in unique(workingSet$yearRecoded)) {

# Subset data for the current year

subset\_df <- workingSet[workingSet$yearRecoded == group, ]

# Get value counts for Level2Ecoregion

counts <- as.data.frame(table(subset\_df$Level2Ecoregion))

# Rename columns for clarity

colnames(counts) <- c("Level2Ecoregion", "Count")

# Add year column

counts$Category <- group

# Store in list

result\_list[[as.character(group)]] <- counts

}

# Combining into single df

final\_df <- bind\_rows(result\_list)

# Arranging readability

final\_df <- final\_df |> arrange(Category, Level2Ecoregion)

final\_df <- final\_df |>

mutate(Level2Ecoregion = recode(Level2Ecoregion, 'MISSISSIPPI ALLUVIAL AND SOUTHEAST USA COASTAL PLAINS' = "Mississippi Alluvial & Southeast USA Coastal Plains", 'SOUTHEASTERN USA PLAINS' = "Southern USA Plains")) |>

group\_by(Category) |>

mutate(Proportion = Count / sum(Count) \* 100)

l2ProportionPlot <- ggplot(final\_df, aes(fill = Level2Ecoregion, y = Count, x = Category)) +

geom\_bar(position = "fill", stat = "identity") +

geom\_text(

aes(label = sprintf("%.1f%%", Proportion / 100 \* 100)),

position = position\_fill(vjust = 0.5),

size = 4,

face = 'bold',

color = 'white'

) +

scale\_fill\_manual(

values = c(

"Mississippi Alluvial & Southeast USA Coastal Plains" = "#002673",

"Southern USA Plains" = "#267300"

),

labels = function(x) str\_wrap(x, width = 20)

) +

labs(

title = "Level 2 Ecoregion Segment Proportion by Year of Collection",

x = "Group",

y = "Proportion of Segments",

fill = "Level 2 Ecoregion"

) +

scale\_y\_continuous(labels = scales::percent) +

theme\_minimal() +

customTheme

ggsave(

"D:/IceSat/writing/thesis/figures/l2\_ecoregion\_proportion\_plot.png",

plot = l2ProportionPlot,

bg = "white",

width = 8, height = 5, units = "in", dpi = 300

)

l2ProportionPlot

```