# Team LumberHack


Team: Colin Wood, Jenna Pedro, Thomas Whitney, & Matthew Flanders

Client: Andrew J. Sanchez Meador, Ph.D

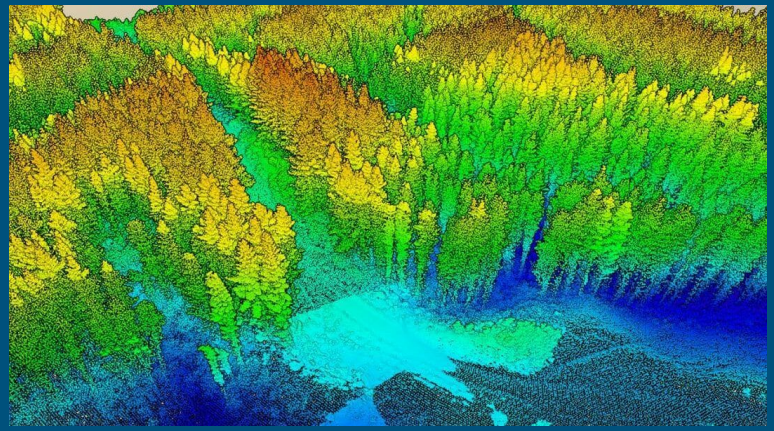
Mentor: Melissa D. Rose

# What is LiDAR

- Remote sensing using light pulses
  - Airborne and Mobile
- Maps data points into a point cloud
- Allows for much faster surveying than traditional methods

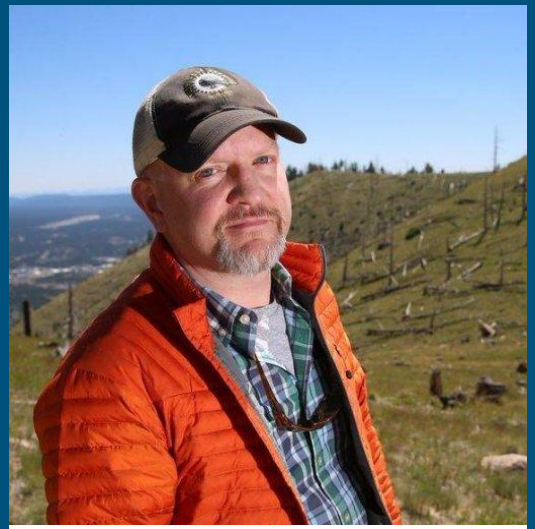
Airborne lidar

Mobile lidar



Point Cloud

# Client / Problem Statement



NORTHERN ARIZONA UNIVERSITY | Ecological Restoration Institute

Andrew J. Sanchez Meador, Ph.D

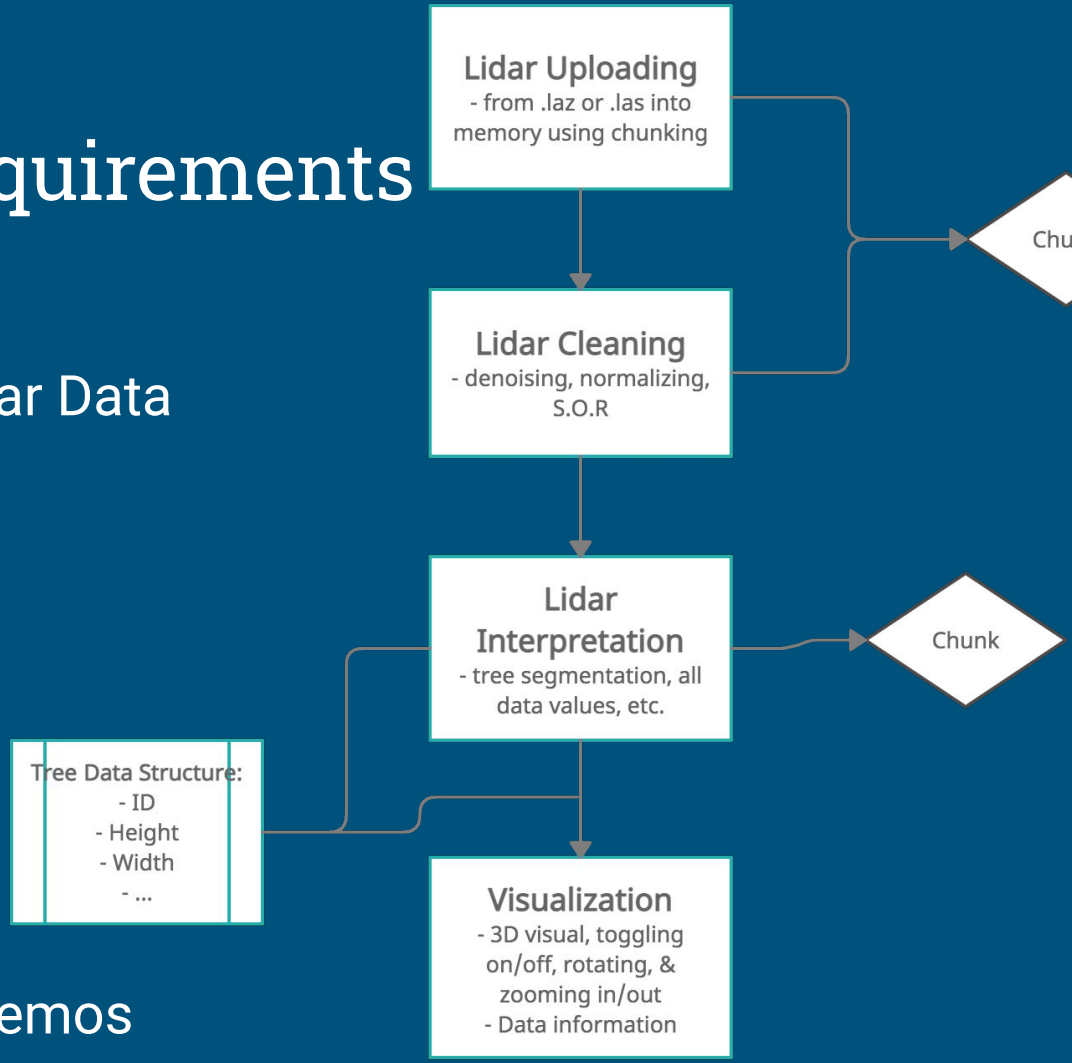
- Current tools are not mobile lidar focused
- Lack of automation with current software and methods
- Lidar data files are large (millions of data points)

# Solution Overview

- Expansion on similar R packages (lidR, TreeLS)
- Built specifically for mobile lidar
- One cohesive tool to improve workflow
  - Wrapped as a Shiny app

# Domain Level Requirements

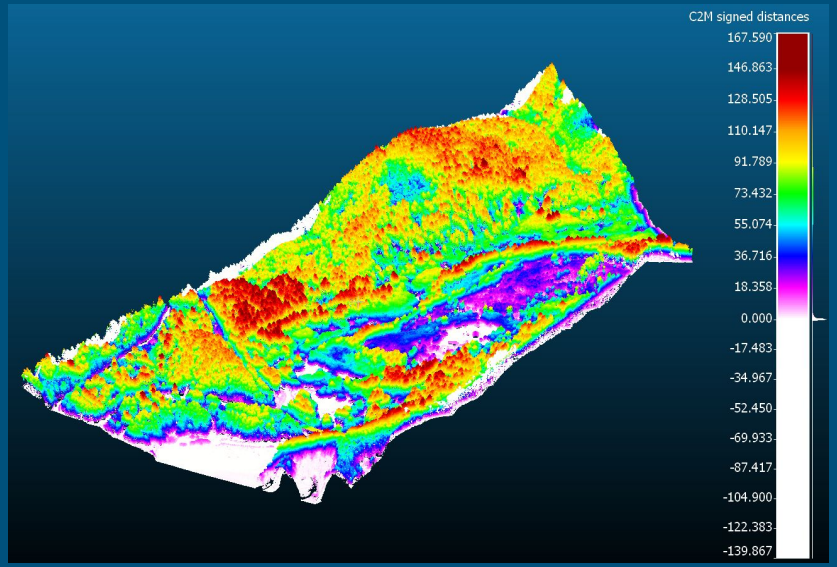
- Data Upload
- Clean/Normalized Lidar Data
- Interpretation of Data
  - Diameter
  - Height
  - Location
- Visualization
  - Tabular
  - 3D Point Cloud
- Client meetings and demos



# Key Functional Requirements

## Calculates ecologically relevant features

- Manually classify trees and segment them individually with CSF in CloudCompare
- Tree height: height-above ground method
- Volume: stem curve
- Above-ground biomass: cylinder shape fitting

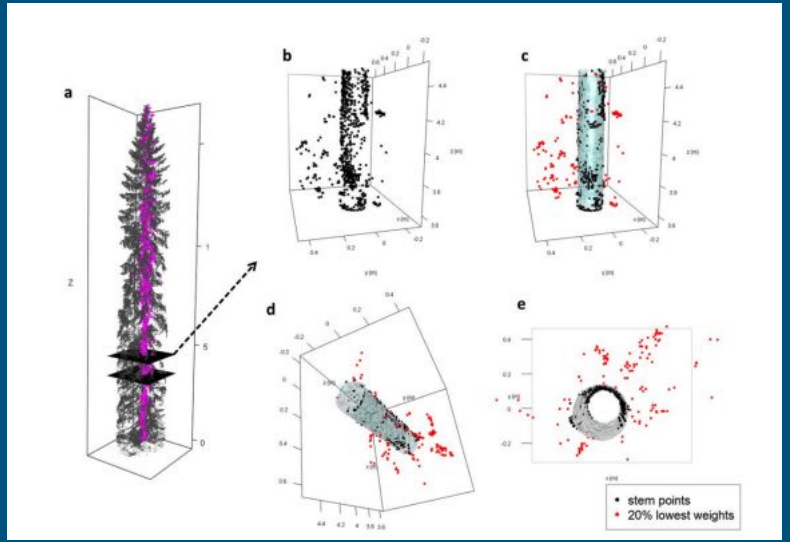



Cloth Simulation Filter(CSF) for ground classification

# Key Functional Requirements

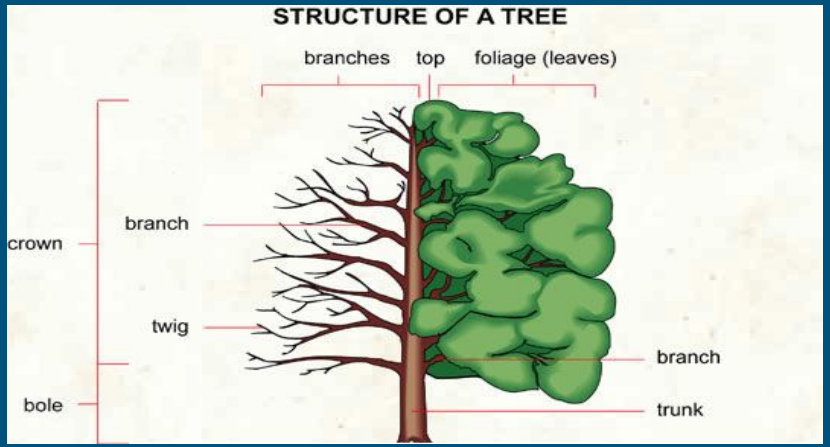
## Tree Diameter: Cylinder Shape Fitting

- **Uses RANSAC(Random Sample Consensus) to tree boles with attribute and error reporting**
  - Takes slice and individualizes trees
  - Cylinder fit with 1.37m at center




RANSAC Cylinder Fitting Process

# Key Functional Requirements




Classification of specific points

- **Such as the bole, branch, crown of tree, and others (including logs that are laying horizontal on the ground)**
  - Ground and bole are an important baseline
  - Allows for branch and crown classification later based on subtraction of removing ground and bole from the point cloud

# Key Functional Requirements

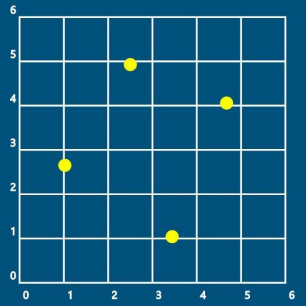
Identify key characteristics within a point cloud

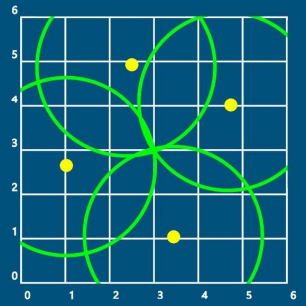
**Functional:**

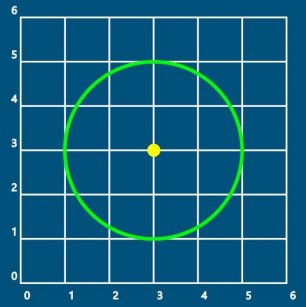
- Identify characteristics of points as being linear, planar, or spherical in orientation.
- Identify the center and radius of tree segments.
- Perform the operations quickly.

**Non-functional:**

- Calculate eigenvalues to determine how groups of points are orientation in 3d space.
- Preform Hough transformation to determine the center and radius from a horizontal slice of points.
- Perform computations in parallel.



# Key Functional Requirements

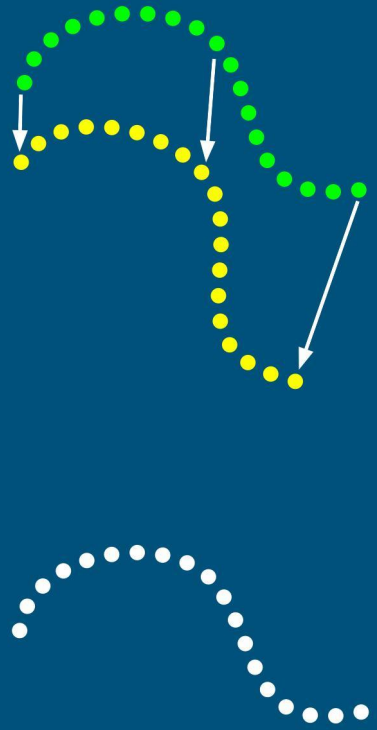
Registration of two, temporally distinct but overlapping, point clouds

**Functional:**

- Load two point clouds and align them for comparison.

**Non-Functional:**

- Alignment by physical markers, or by orienting in the GUI.
- Then use point cloud registration tools such as the CloudCompare Iterative Closest Point (ICP) algorithm to fine tune the alignment.



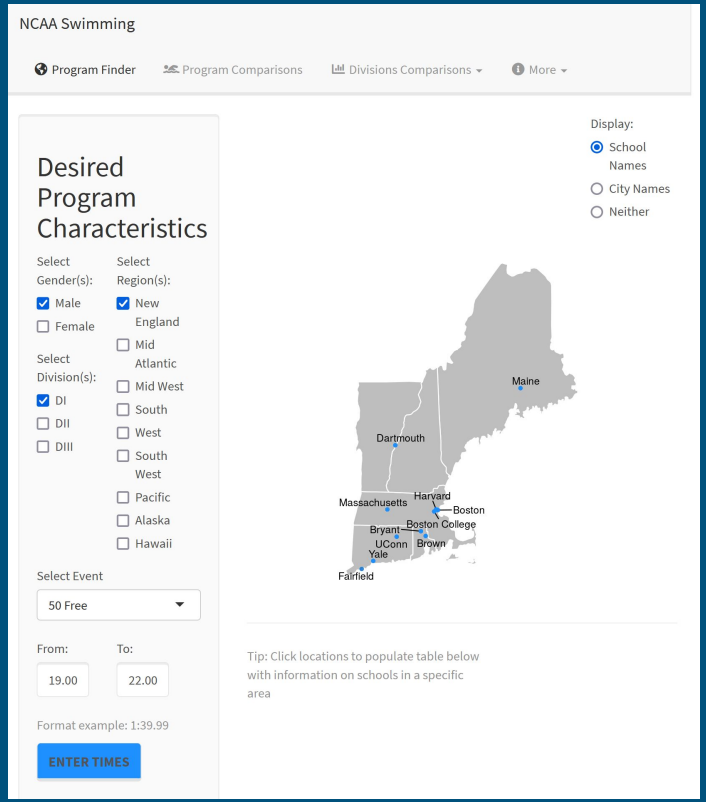
Simple example of ICP

# Key Functional Requirements

User interface

**Functional:**

- Simple interface for processing data.
- Parameter modification fields.
- 3d visualization.
- Tabular representation data.




Example shiny web app

# Key Requirements

Environmental

- R programming language
- Utilize multiple CPU cores

Stretch Goals

- Vignettes explain the functionality of the package and the Shiny app
- Package uploaded and passing all CRAN checks so it can be an official “CRAN” package

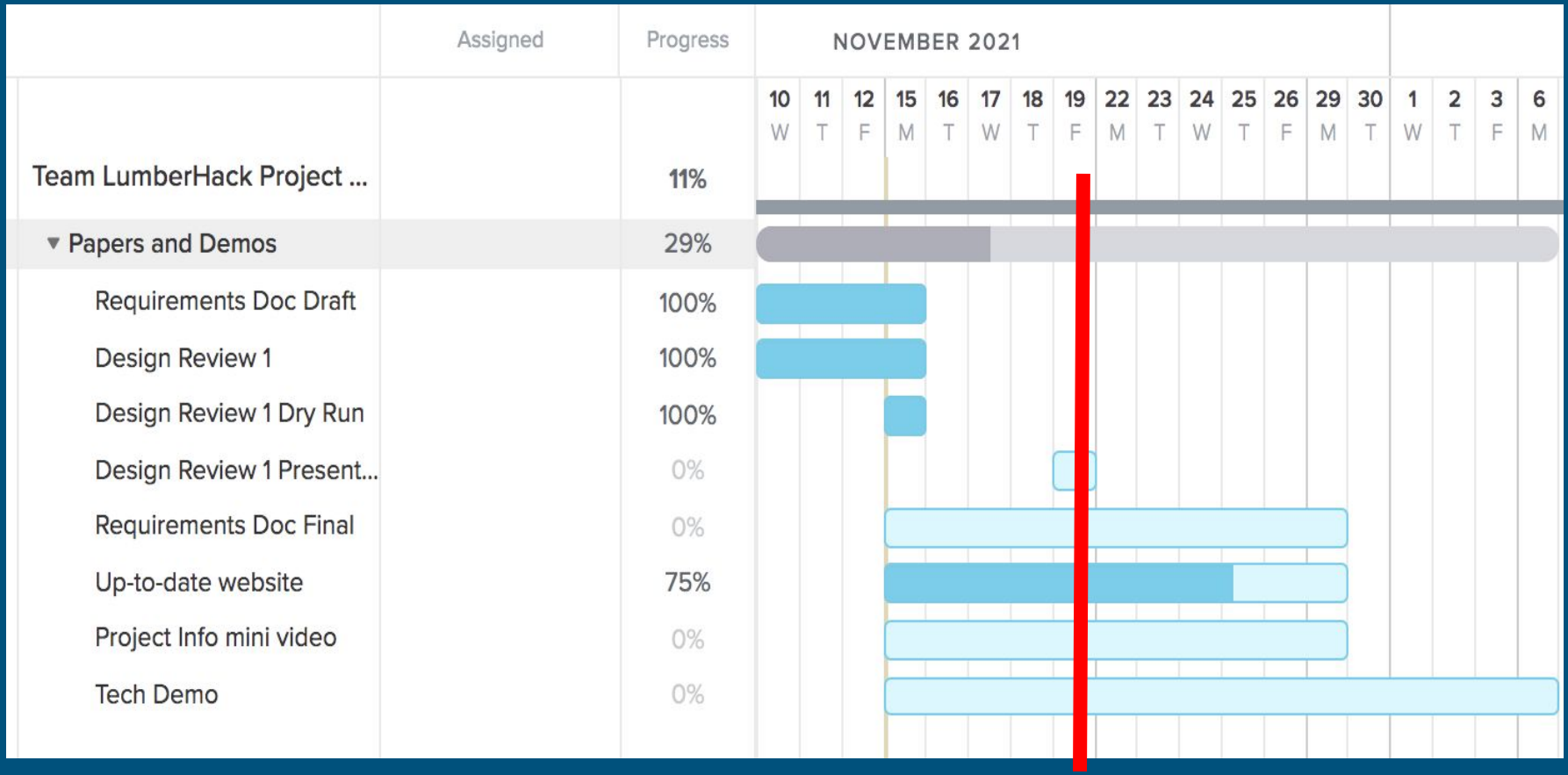
# Key Risks

- Inaccurate or invalid data processing and measurements
  - This would affect the credibility of our product, possibly deterring research use

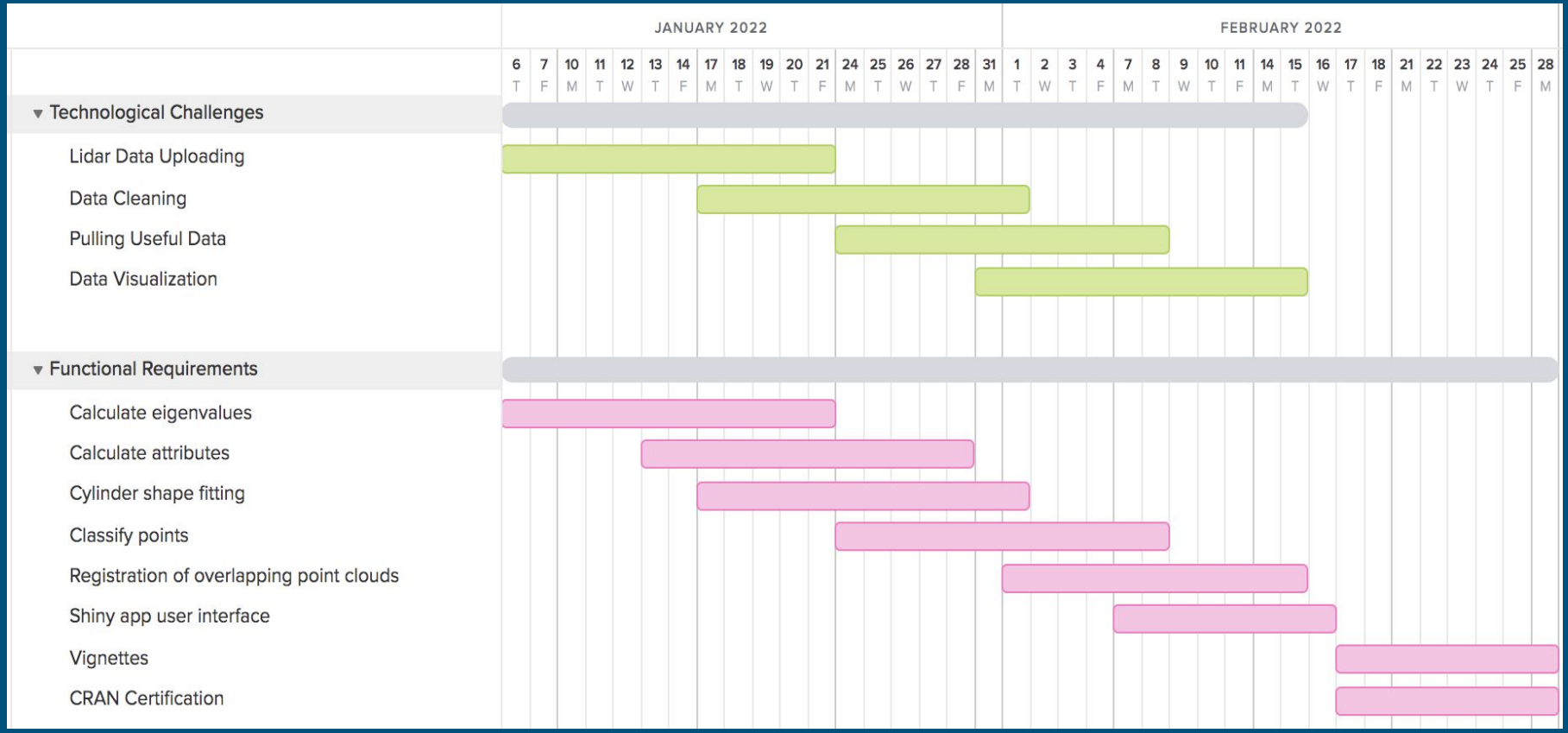
- Code is poorly developed and/or documented
  - Collaborative development from other researchers less likely
  - Code maintenance is difficult, project may eventually become obsolete
  - Non-specialists likely to be frustrated learning to use the software

- Poor user interface design
  - The UI is the face of our product and must thus be intuitive, capable, and non-buggy
  - If any of the above goals are not met, users are likely to be frustrated

# Schedule (Fall)



# Schedule (Spring)



# Looking Ahead

Moving forward:

- We'll have a complete understanding of requirements and will begin development full-time
- Create a full project skeleton more concretely (code and documentation)
- Prepare for the tech demo
  - Create a "dummy" Shiny app so the team can gain familiarity
  - Test various parts of the lidar data processing workflow

# Conclusion

Our project has the potential to:

- Streamline the lidar workflow for forest researchers
- Make lidar data interpretation more accessible for the non-specialist

To accomplish the above we envision:

- An R package that performs the core data processing and calculations
  - Tree height, diameter, crown size, and other calculations
- A Shiny web framework that visualizes the desired outputs
  - Statistics as desired, 2-D and 3-D visualizations tailored to the processed data