Lab 9: Spaceborne Lidar

Anthony Stewart

2023-05-23

Objectives:

- Introduction to full waveform lidar
- Introduction to Global Ecosystem Dynamics Investigation (GEDI) Lidar
  - https://gedi.umd.edu/ (https://gedi.umd.edu/)
  - https://gedi.umd.edu/return-of-the-gedi-space-based-forest-carbon-mapping-laser-array-saved/ (https://gedi.umd.edu/return-of-the-gedi-space-based-forest-carbon-mapping-laser-array-saved/)
- Introduction to The Ice, Cloud and land Elevation Satellite-2, or ICESat-2
  - https://www.nasa.gov/content/goddard/about-icesat-2 (https://www.nasa.gov/content/goddard/about-icesat-2)
- Downloading and using GEDI data
- Characterizing forested landscapes with full waveform lidar

Data and Software:

- LAB9_GEDI data
- `rGEDI` : An R Package for NASA's Global Ecosystem Dynamics Investigation (GEDI) Data Visualization and Processing https://cran.r-project.org/web/packages/rGEDI/vignettes/tutorial.html (https://cran.r-project.org/web/packages/rGEDI/vignettes/tutorial.html)
- ArcGIS Pro

What you will turn in:

Lab 9 Submission Quiz (https://canvas.uw.edu/courses/1633883/quizzes/1863823)

Welcome to Lab 9 for ESRM433/SEFS533 This lab is all about Spaceborn lidar. We have covered aerial lidar, and terrestrial lidar, now we will talk about spaceborne lidar scanning (SLS). Two prominent spaceborn lidar sensors are ICESat and GEDI.

PART 1a GEDI background

For this lab, we are going to focus on GEDI data. The GEDI sensor was launched to the International Space Station (ISS) in December of 2018 and is slated for a two year mission to scan terrestrial ecosystems. The data collected is limited to the orbit of the ISS so data isn't available for latitudes greater than 51.60.

For the following questions refer to: - GEDI's website https://gedi.umd.edu/ (https://gedi.umd.edu/)

GEDI videos

- https://www.youtube.com/watch?v=SSdDPFfUVIo (https://www.youtube.com/watch?v=SSdDPFfUVIo)

- https://www.youtube.com/watch?v=wxgrxvAKpTo (https://www.youtube.com/watch?v=wxgrxvAKpTo)
- https://www.youtube.com/watch?v=XFlm-TmhvjM (https://www.youtube.com/watch?v=XFlm-TmhvjM)

The lidar data from GEDI retains it's full waveform. ALS data is also based on the amount of energy returned from a laser pulse, but the full waveform is discarded and only the peaks in the waveform are preserved. These peaks are what we call returns when using terms like "discreate return lidar" vs "full wavefrom lidar"

The beam divergence of lasers is extremely important to keep in mind when discussion SLS. Beam divergence is in direct relationship to the distance the laser pulse travels. For TLS, objects are much closer to the scanner so beam divergence is likely to be a few mm to a few cm depending on the distance. For ALS, beam divergence can be ~10cm to ~30cm. This divergence is what allows for multiple returns from a single pulse.

QUESTION 1: Is GEDI data free to the public?

QUESTION 2: Where can you get GEDI data?

QUESTION 3: What is the footprint (resolution) of GEDI laser pulses? This is the same as beam divergence.

QUESITON 4: How many beams of data is GEDI collecting? What is their spacing along track and between tracks?

QUESTION 5: Is every square meter of the earths surface getting sampled by GEDI?

QUESTION 6: What are the three science questions for GEDI?

QUESTION 7: For Data Products, there are multiple Algorithm Theoretical Basis Documents (ATBDs) for GEDI data. We are most interested in L1B, L2A, & L2B. What is the name of the data product for those 3 ATBDs?

PART 2: Getting and processing GEDI data

Most of the following tutorial was taken from: https://cran.r-project.org/web/packages/rGEDI/vignettes/tutorial.html (https://cran.r-project.org/web/packages/rGEDI/vignettes/tutorial.html)

An important part of this class is to demonstrate how accessible lidar data is for researchers. This is certainly true about GEDI data, however, **GEDI data downloads are huge**. The smallest bundles are about **6GB**. This is in the highly compressed .h5 format. H5 is a Hierarchical Data format and if you want more info check out: https://www.neonscience.org/about-hdf5 (https://www.neonscience.org/about-hdf5)

Because of the size constraints, you will be provided with a small snippet of GEDI data to work with. A package for R has been developed to facilitate working with GEDI data, `rGEDI`

YOU DO NOT NEED TO RUN THE CODE BELOW FOR THIS LAB. It will take a while and is better used in your own time/projects

Warning: rGEDI WILL NOT WORK ON MacOS! The package hasn't seen many updates and will only work on Windows. Therefore you may need to work through Madrona/SEFS Spatial. Check out this link here for the github issue https://github.com/carlos-alberto-silva/rGEDI/issues/42 (https://github.com/carlos-alberto-silva/rGEDI/issues/42)

```
library(devtools)
devtools::install_git("https://github.com/carlos-alberto-silva/rGEDI", dependencies =
T)
```

```
These packages have more recent versions available.
It is recommended to update all of them.
Which would you like to update?

 1: All
 2: CRAN packages only
 3: None
 4: gtable       (0.3.1      -> 0.3.3     ) [CRAN]
 5: utf8         (1.2.2      -> 1.2.3     ) [CRAN]
 6: colorspace   (2.0-3      -> 2.1-0     ) [CRAN]
 7: fansi        (1.0.3      -> 1.0.4     ) [CRAN]
```

```
setwd("/Users/Anthony/OneDrive - UW/University of Washington/Teaching/SEFS433_Lidar/L
abs/Lab 9/")
library(rGEDI)
library(sf)
library(terra)
```

```
ul_lat<- study_extent["ymax"]
lr_lat<- study_extent["ymin"]
ul_lon<- study_extent["xmin"]
lr_lon<- study_extent["xmax"]
```

```
?gedifinder
gLevel1B<-gedifinder(product="GEDI01_B",ul_lat, ul_lon, lr_lat, lr_lon,version="002",
daterange=daterange)
gLevel2A<-gedifinder(product="GEDI02_A",ul_lat, ul_lon, lr_lat, lr_lon,version="002",
daterange=daterange)
gLevel2B<-gedifinder(product="GEDI02_B",ul_lat, ul_lon, lr_lat, lr_lon,version="002",
daterange=daterange)
```

```
gLevel1B
gLevel2A
gLevel2B
```

```
outdir <- ("Lab9/Lab9Data/GEDI_DL/")
```

OK THIS IS THE END OF THE REFERENCE CODE. Everything else below is going to be code we run in R

Part 3: Reading downloaded GEDI data

Check out rGEDI here: https://github.com/carlos-alberto-silva/rGEDI (https://github.com/carlos-alberto-silva/rGEDI)

Install the package here:

```
library(devtools)
devtools::install_git("https://github.com/carlos-alberto-silva/rGEDI", dependencies =
T)
```

load libraries and set working directory

```
#change to your own
setwd("/Users/Anthony/OneDrive - UW/University of Washington/Teaching/SEFS433_Lidar/L
abs/Lab 9/")

library(rGEDI)
library(sf)
library(terra)
```

Use the `readLevel` functions to read in your .h5 GEDI data. We have three types we're going to explore:

- Level 1B
- Level 2A
- Level 2B

```
?readLevel1B
?readLevel2A
?readLevel2B
gedilevel1b <- readLevel1B("Lab9/Lab9Data/LAB9_GEDI/PF_level1b_clip.h5")
gedilevel2a <- readLevel2A("Lab9/Lab9Data/LAB9_GEDI/PF_level2a_clip.h5")
gedilevel2b <- readLevel2B("Lab9/Lab9Data/LAB9_GEDI/PF_level2b_clip.h5")
```

You now have GEDI data loaded into R! There is a lot of data packed into those h5 files. The first thing that we should do is to get the location information for the pulses.

getting the geo locations:

```
?getLevel1BGeo
level1BGeo <- getLevel1BGeo(level1b = gedilevel1b, select = c("elevation_bin0"))
```

```
##
  |
  |                                                                      |   0%
  |
  |=                                                                     |   2%
  |
  |===                                                                   |   4%
  |
  |====                                                                  |   6%
  |
  |======                                                                |   8%
  |
  |=======                                                               |  10%
```

```
head(level1BGeo)
```

```
##          shot_number latitude_bin0 latitude_lastbin longitude_bin0
## 1: 72230000300104806      46.89253         46.89253      -122.3994
## 2: 72230000300104807      46.89230         46.89230      -122.3987
## 3: 72230000300104808      46.89208         46.89207      -122.3980
## 4: 72230000300104809      46.89185         46.89184      -122.3974
## 5: 72230000300104810      46.89162         46.89162      -122.3967
## 6: 72230000300104811      46.89140         46.89139      -122.3960
##    longitude_lastbin elevation_bin0
## 1:         -122.3993       202.2359
## 2:         -122.3987       190.0121
## 3:         -122.3980       204.2407
## 4:         -122.3973       220.8789
## 5:         -122.3967       230.3352
```

That code pulls the lat and long for each shot (pulse) from the h5 file. The header of the new file shows you a snippet of the data. You have shot_number (every pulse has it's own ID number) and the lat, long, and elevation.

There are a few steps we need to do to convert the data into a format that we can output as a shapefile which is a more flexible format for us to use in vizualizing the spatial data of GEDI: - Convert `shot_number` from 'interger64' to 'character'

```
level1BGeo$shot_number <- as.character(level1BGeo$shot_number)
```

- Convert `level1BGeo` to a spatial object using the `sf` package and giving it a coordinate reference system (`crs`) with the EPSG code of 4326

```
level1BGeo_sf <- st_as_sf(level1BGeo, coords = c("longitude_bin0", "latitude_bin0"),
crs = 4326)
```

Now we can finally look at the locations of our individual shots

```
mapview(level1BGeo_sf, map.types = "Esri.WorldImagery")
```

And we can write the locations as an ESRI shapefile (`.shp`) to our computer and view it in another program like ArcGIS Pro

```
sf::write_sf(level1BGeo_sf, "Lab9/Lab9Data/GEDI_level1BGeo.shp")
```

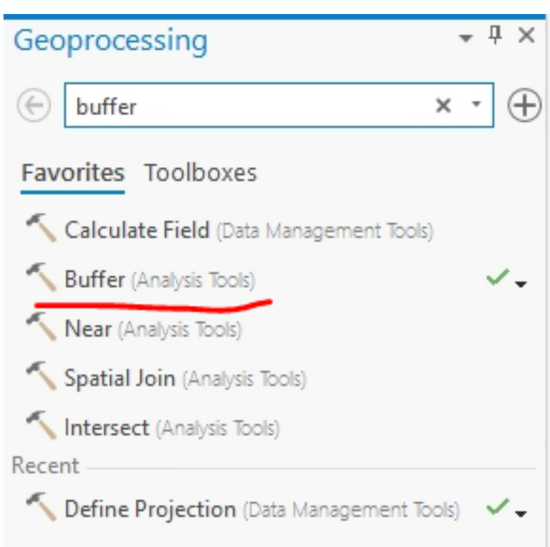
ArcGIS

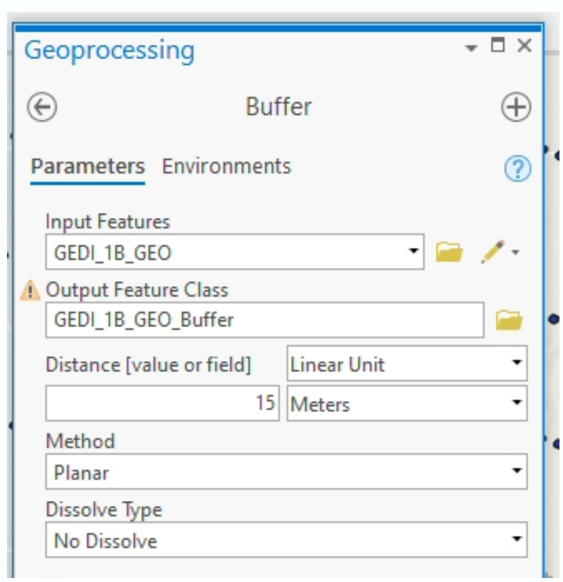
You should now have a shape file called `GEDI_level1BGeo.shp`

Open ArcGIS and make a new map project called LAB9. Insert a basemap that is world imagery.

Drag and drop your `GEDI_level1BGeo.shp` file into ArcGIS. You know that the diameter of each shot is 25 to 30m. Let's create a buffer around each point so we get a good idea of the area actually sampled by each shot.

Use the toolbox to search for buffer





Remember, your point size will change depending on your level of zoom. You need the buffers drawn to see the footprint of the shots.

QUESTION 8: Identify 3 different shots by their number. You can use the attribute table for this:

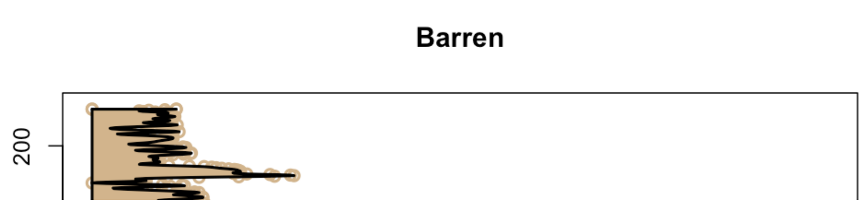
- **72230000300104862 - barren or cleared shot**
- **72230300300100204 - heavily forested shot**
- **72230300300100165 - light to moderate vegetated shot**

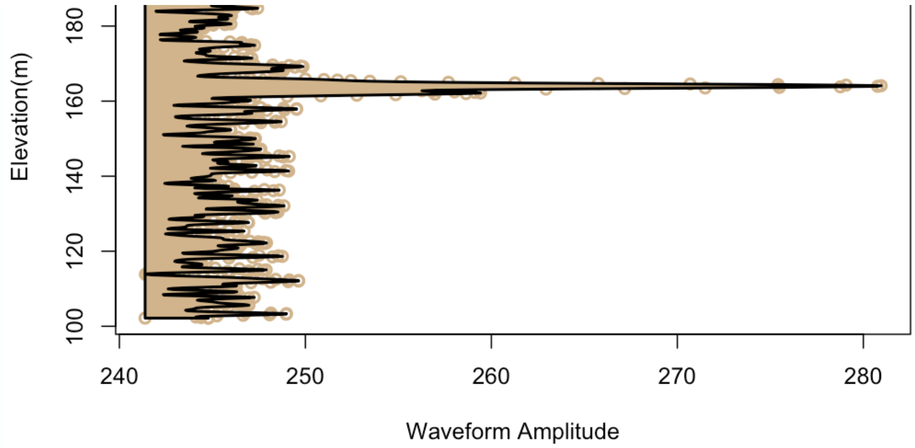
Include a screenshot of each shot (use the buffer, not the point).

Part 4: GEDI Waveforms

We are now going to compare the full waveform lidar of each shot.

```
?getLevel1BWF

wfb2 <- getLevel1BWF(gedilevel1b, shot_number = "72230000300104862")

plot(wfb2, polygon = TRUE,  lwd = 2, col = "tan", xlab= "Waveform Amplitude", ylab =
"Elevation(m)", main = "Barren")
```






```
wfb2 <- getLevel1BWF(, shot_number = "") # fill in here
plot(wfb2, polygon = TRUE, type = "l", lwd = 2, col = "forestgreen", xlab= "Waveform
Amplitude", ylab = "Elevation(m)", main = "Forested")

wfb3 <- getLevel1BWF(, shot_number = "") #fill in here
plot(wfb3, polygon = TRUE, type = "l", lwd = 2, col = "lightgreen", xlab= "Waveform A
mplitude", ylab = "Elevation(m)", main = "Light Veg")
```

QUESTION 9: Include screenshots of your three waveforms. Make sure they are labeled “Barren”, “Forested”, and “Light Vegetation”. Discuss the similarities and differences between the waveforms. Remember, this is the full waveform and is not yet cropped with the ground and tree height identified.

QUESTION 10: You have now seen the output from L1B Give a more thorough description of what exactly L1B is

Lets take a look at the data contained in our L2A.

```
?getLevel2AM
level2AM <- getLevel2AM(gedilevel2a)
```

```
##
  |
  |                                                                      |   0%
  |
  |=========                                                             |  12%
  |
  |==================                                                    |  25%
  |
  |==========================                                            |  38%
  |
  |===================================                                   |  50%
  |
  |============================================                          |  62%
```

```
head(level2AM[, c("beam", "shot_number", "elev_highestreturn", "elev_lowestmode", "rh
100")])
```

```
##        beam       shot_number elev_highestreturn elev_lowestmode rh100
## 1: BEAM0000 72230000300104806           157.7767        121.6699 36.10
## 2: BEAM0000 72230000300104807           145.7397        122.0679 23.67
## 3: BEAM0000 72230000300104808           159.7441        122.7010 37.04
## 4: BEAM0000 72230000300104809           176.2708        131.9998 44.27
## 5: BEAM0000 72230000300104810           186.4405        155.6543 30.78
## 6: BEAM0000 72230000300104811           164.8258        160.5187  4.30
```

Repeat from above: Converting shot_number as "integer64" to "character"

```
level2AM$shot_number<-as.character(level2AM$shot_number)
```

Repeat from above: Converting Elevation and Height Metrics as data.table to `sf` spatial object

```
level2AM_sf<-st_as_sf(level2AM, coords = c("lon_lowestmode", "lat_lowestmode"), crs =
4326)
```

You can export all of the point location and information into an ESRI shape file, just like you did with the L1B data. This time when you click on each point, you will have a list of all the elevation and height metrics.

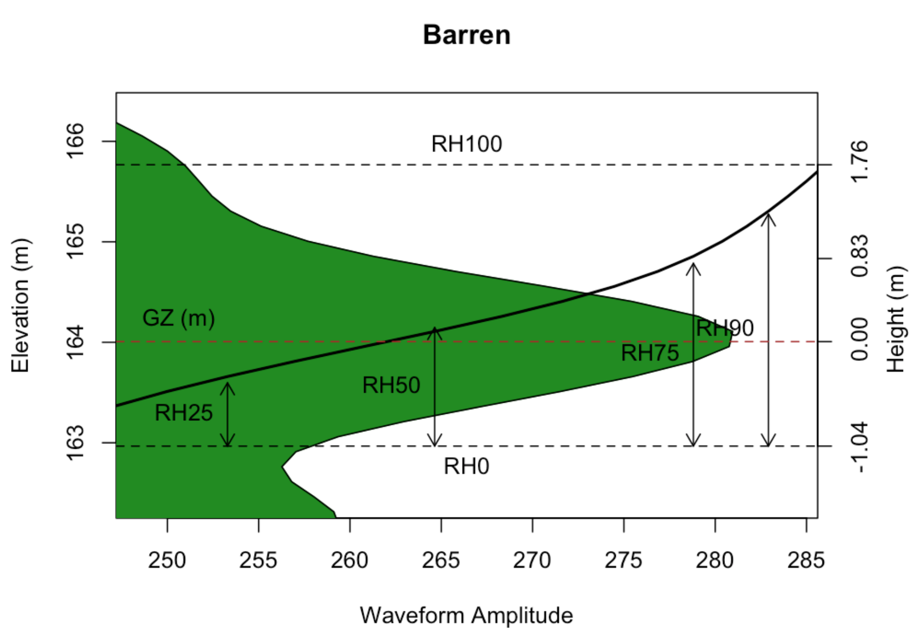
Export Elevation and Height metrics as an ESRI shapefile

```
sf::write_sf(level2AM_sf, "Lab9/Lab9Data/GEDI_level2AM.shp")
```

Lets plot the waveform L2A metrics for the Barren shot:

```
?plotWFMetrics

plotWFMetrics(gedilevel1b, gedilevel2a, "72230000300104862", rh=c(25, 50, 75, 90), ma
in = "Barren")
```

```
##
  |
  |                                                                     |   0%
  |
  |=========                                                            |  12%
  |
  |==================                                                   |  25%
  |
  |==========================                                           |  38%
  |
  |===================================                                  |  50%
  |
  |============================================                         |  62%
```



For information about specific the algorithms for Canopy Cover and Height refer to: https://lpdaac.usgs.gov/documents/588/GEDI_FCCVPM_ATBD_v1.0.pdf (https://lpdaac.usgs.gov/documents/588/GEDI_FCCVPM_ATBD_v1.0.pdf)

QUESTION 11: What exactly is the RH metric measuring? Check out: https://gedi.umd.edu/mission/technology/ (https://gedi.umd.edu/mission/technology/)

QUESTION 12: Create plots for your Barren, Forested, and Light vegetation shots. Compare and contrast the waveforms between the different land cover. If there isn't a clear difference between your plots, sample a new location. Include a screenshot of the three plots.

```
plotWFMetrics(gedilevel1b, gedilevel2a, "72230300300100204", rh=c(25, 50, 75, 90), ma
in = "Forest")
plotWFMetrics(gedilevel1b, gedilevel2a, "72230300300100165", rh=c(25, 50, 75, 90), ma
in = "Light Veg")
```

The last clip of data that you have is L2B. Just as with the other data clips we will bring the data into R and then convert it into a format that can be output as a shapefile:

```
?getLevel2BVPM
```

```
level2BVPM<-getLevel2BVPM(gedilevel2b)
```

```
##
  |
  |                                                                      |   0%
  |
  |=========                                                             |  12%
  |
  |==================                                                    |  25%
  |
  |==========================                                            |  38%
  |
  |===================================                                   |  50%
  |
  |============================================                          |  62%
```

```
head(level2BVPM[,c("beam","shot_number","pai","fhd_normal","omega","pgap_theta","cove
r")])
```

```
##    beam       shot_number        pai fhd_normal omega pgap_theta      cover
## 1:    0 72230000300104806 2.00382900   3.279090     1  0.3670606 0.63274115
## 2:    0 72230000300104807 0.85066009   3.127365     1  0.6534669 0.34642449
## 3:    0 72230000300104808 2.84563279   3.293170     1  0.2409268 0.75883543
## 4:    0 72230000300104809 4.03077650   3.169244     1  0.1331845 0.86654395
## 5:    0 72230000300104810 2.19123483   3.193733     1  0.3342184 0.66557312
## 6:    0 72230000300104811 0.05784217   1.555367     1  0.9714843 0.02850675
```

Converting `shot_number` as “integer64” to “character”

```
level2BVPM$shot_number <- as.character(level2BVPM$shot_number)
```

Converting GEDI Vegetation Profile Biophysical Variables from data.table to sf spatial object

```
level2BVPM_sf <- st_as_sf(level2BVPM, coords = c("longitude_lastbin", "latitude_lastb
in"), crs = 4326)
```

Exporting to ESRI shapefile

```
write_sf(level2BVPM_sf, "Lab9/Lab9Data/GEDI_level2BVPM.shp")
```

Part 5: Level 2B: Plant Area Index (PAI)

`rGEDI` uses a metric called *Plant Area Index (PAI)*. This is similar to leaf area index but it doesn’t distinguish between live green leaves and woody tree stems. https://en.wikipedia.org/wiki/Leaf_area_index (https://en.wikipedia.org/wiki/Leaf_area_index) https://lpdaac.usgs.gov/documents

/588/GEDI_FCCVPM_ATBD_v1.0.pdf (https://lpdaac.usgs.gov/documents/588/GEDI_FCCVPM_ATBD_v1.0.pdf)

PAI differs from height metrics as it relies on the amount of energy that filters through the different layers of forest canopy. A forest with shorter trees but the trees have very full crowns and there is a full midstory layer of vegetation could have a much higher PAI than a forest with very tall trees, but a small crown ratio and little to no midstory vegetation.

- Tang, Hao, et al. “Retrieval of vertical LAI profiles over tropical rain forests using waveform lidar at La Selva, Costa Rica.” Remote Sensing of Environment 124 (2012): 242-250.
- Zhao, Feng, et al. “Measuring effective leaf area index, foliage profile, and stand height in New England forest stands using a full-waveform ground-based lidar.” Remote Sensing of Environment 115.11 (2011): 2954-2964

LAI and PAI can be expressed as ratios with a value of 1 indicating that the area of plant matter surfaces is equal to the spatial planar area of the sample location. A value less than 1 will have less vegetation, while a value greater than 1 will have more vegetation. There is no upper limit to the lai values but a value approaching 10 would indicate a very dense forest. A value of 0 would be bare ground.

You can look up your PAI values in the biophysical data by sorting through the table. The easiest way to do it in R is to quickly load in the library `dplyr`. This isn’t a remote sensing package, just a package for working with R data

```
library(dplyr)

Barren <- level2BVPM |> filter(shot_number == "72230000300104862")
Barren$pai

Forest <- level2BVPM |> filter(shot_number == "72230300300100204")
Forest$pai

Lightveg <- level2BVPM |> filter(shot_number == "72230300300100165")
Lightveg$pai
```

QUESTION 13: What is the PAI for your Barren, Forested, and Light Vegetation shots?

We can create profiles of the PAI for the individual GEDI beams. Think of this as a slice that is taken of the vegetation along the beams path. You can get information about the vegetation height as well as the density of the vegetation. This is the core of calculating the biomass of an area.

let’s get the PAI profiles from the GEDI Level2B

```
?getLevel2BPAIProfile
level2BPAI_Profile <- getLevel2BPAIProfile(gedilevel2b)
```

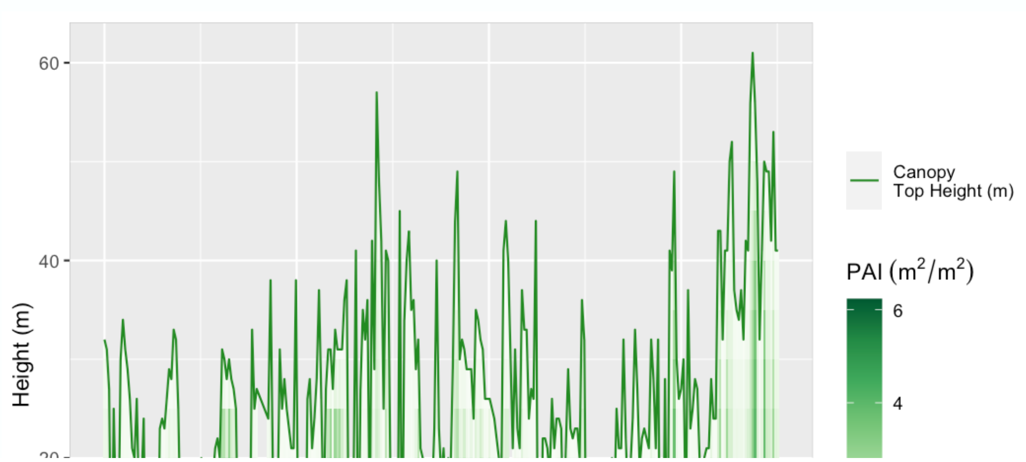
```
##
  |
  |                                                                      |   0%
  |
  |=========                                                             |  12%
  |
  |==================                                                    |  25%
  |
  |==========================                                            |  38%
  |
  |===================================                                   |  50%
  |
  |============================================                          |  62%
```

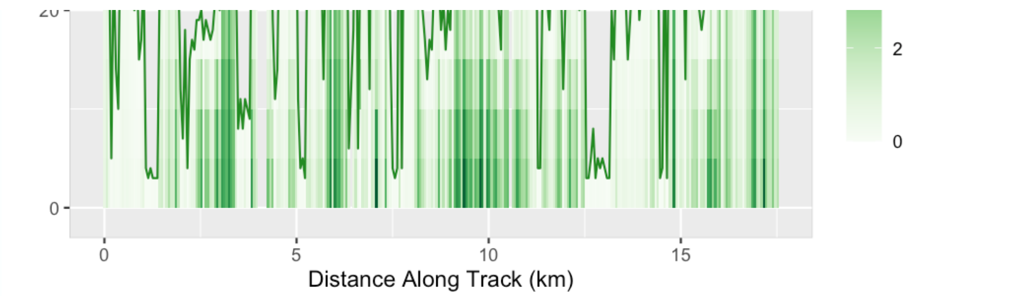
```
head(level2BPAI_Profile[,c("beam", "shot_number","pai_z0_5m","pai_z5_10m")])
```

```
##        beam       shot_number  pai_z0_5m pai_z5_10m
## 1: BEAM0000 72230000300104806 2.00382900  1.6610067
## 2: BEAM0000 72230000300104807 0.85066009  0.7201815
## 3: BEAM0000 72230000300104808 2.84563279  2.6006696
## 4: BEAM0000 72230000300104809 4.03077650  3.0375919
## 5: BEAM0000 72230000300104810 2.19123483  1.9147891
## 6: BEAM0000 72230000300104811 0.05784217  0.0000000
```

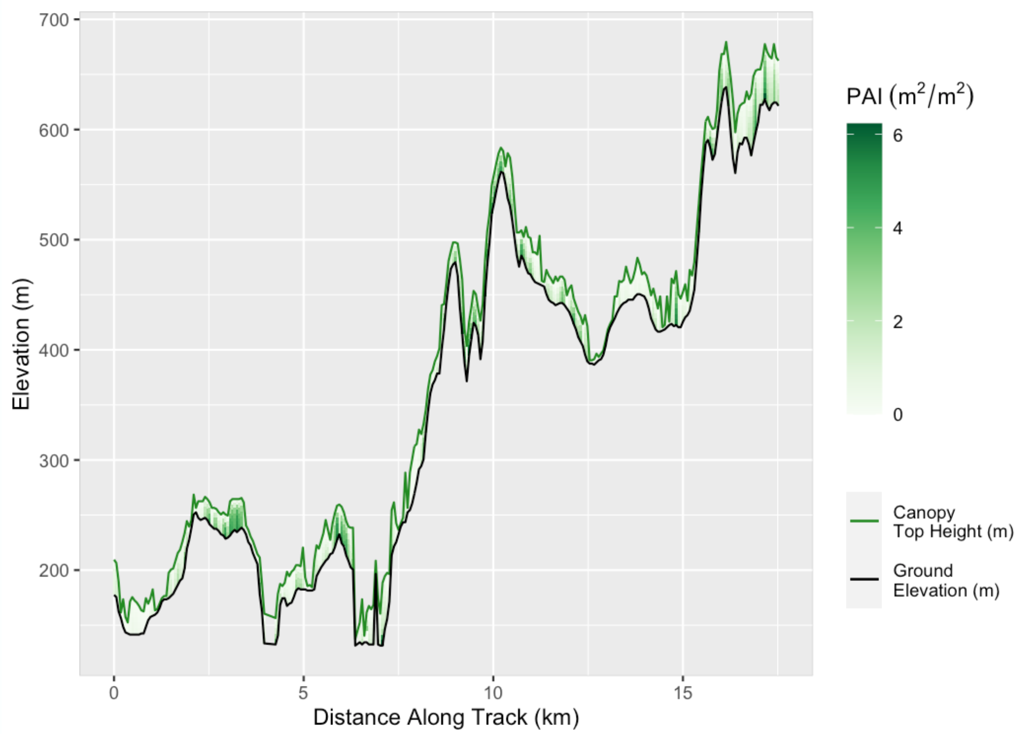
You can create a profile of the different beams from GEDI. You can determine the name of the beam by selecting a point in ArcGIS just like how you determined the shot number. You can “normalize” the profile by removing the elevation from the profile `(elev=FALSE)` .

```
?plotPAIProfile

# The full power beams are 0101, 0110, 1000, 1011
gPAIprofile <- plotPAIProfile(level2BPAI_Profile, beam = "BEAM0101", elev = FALSE)
gPAIprofile <- plotPAIProfile(level2BPAI_Profile, beam = "BEAM0101", elev = TRUE)

#coverage beams are:0000, 0001, 0010, 0011
gPAIprofile <- plotPAIProfile(level2BPAI_Profile, beam = "BEAM0001", elev = FALSE)
gPAIprofile <- plotPAIProfile(level2BPAI_Profile, beam = "BEAM0001", elev = TRUE)
```

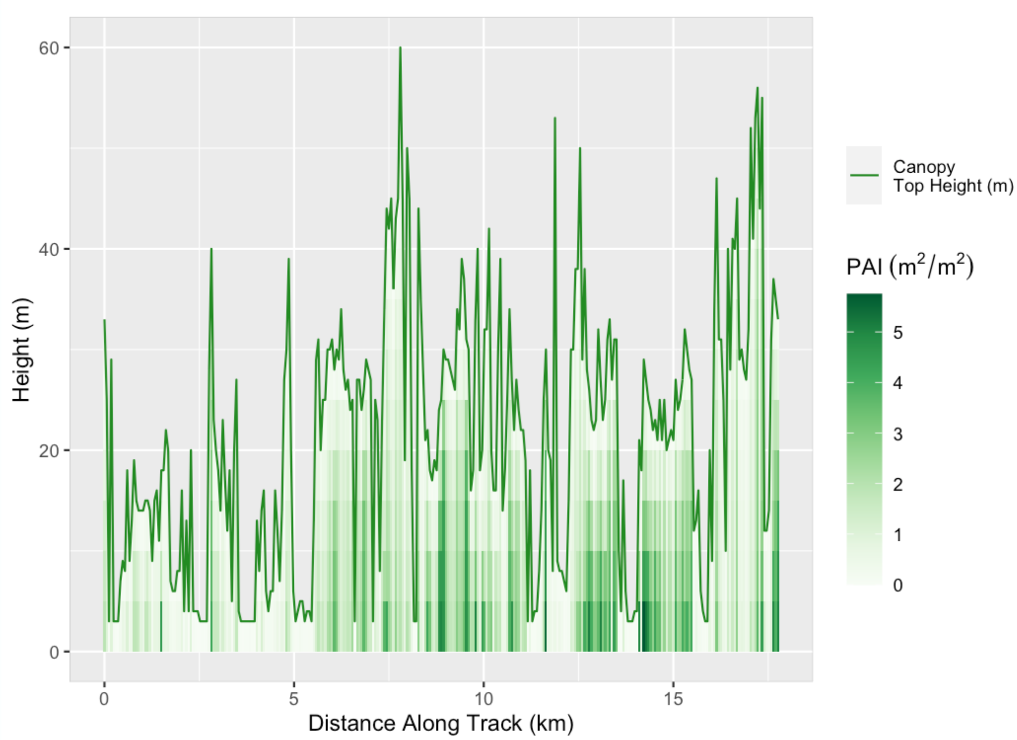




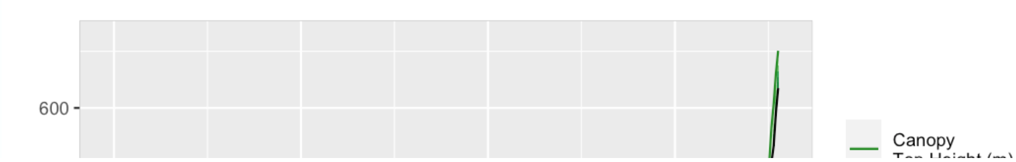
20
0
2
0
0
5
10
15
Distance Along Track (km)



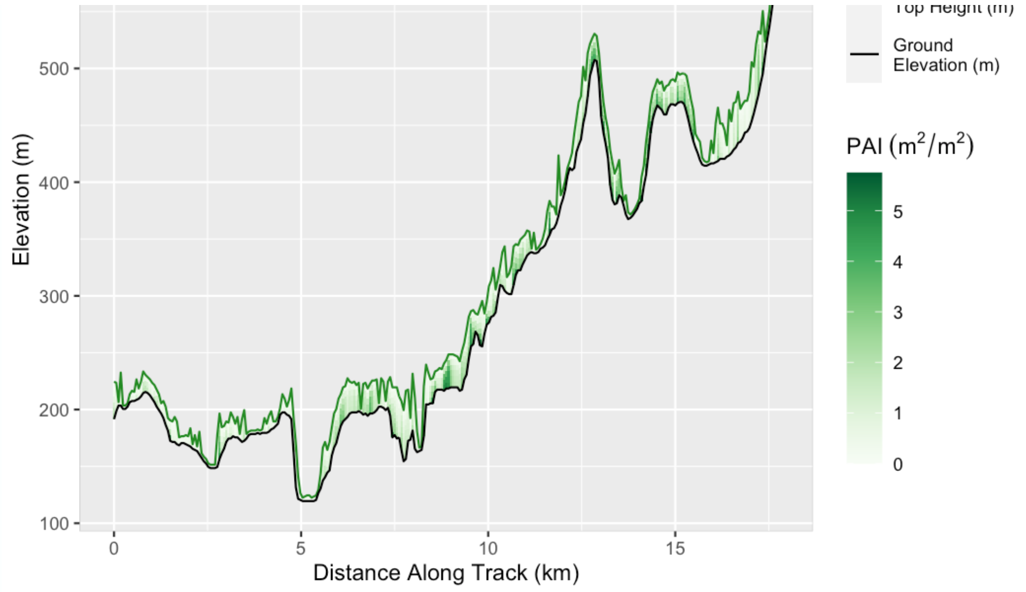
Elevation (m)
700
600
500
400
300
200
0
5
10
15
Distance Along Track (km)
PAI (m^2/m^2)
6
4
2
0
Canopy
Top Height (m)
Ground
Elevation (m)



Height (m)
60
40
20
0
0
5
10
15
Distance Along Track (km)
Canopy
Top Height (m)
PAI (m^2/m^2)
5
4
3
2
1
0



600
Canopy



QUESTION 18: Select a different beam. Include a screen shot of the beam with imagery base map and identify the beam in the image. Include screenshots of the PAI profile with and without the elevation. Caption the images describing what PAI is.

Part 6: Grid Metrics

Lastly we are going to output a raster of the GEDI data. This is very similar to the grid metrics we used with the ALS data. Consider that we only have spot data for a relatively small proportion of our area. We have to define a grid step size that will each grid cell will include at a minimum one of the GEDI shots. Within each cell, the values of each shot within will used to determine the value assigned to the entire cell. Typical metrics are Max, Min, Mean, and standard deviation.

We need to create a function within R for those metrics:

```
mySetofMetrics <- function(x){
  metrics = list(
    min = min(x),
    max = max(x),
    mean = mean(x), #average
    sd = sd(x) #standard deviation
  )
}
```

You defined your own metrics using lidR and this is no different.

We are going to grid our metrics into a raster. Make sure to check out `?gridStatsLevel2am.` The output is a raster brick like we used before. The resolution is much trickier. The resolution output is in decimal degrees. This will output a different resolution raster depending on the latitude of the sample area. You can use a latitude and longitude calculator to figure out the length of a degree of longitude and latitude at different latitudes. http://www.csgnetwork.com/degreelenllavcalc.html (http://www.csgnetwork.com/degreelenllavcalc.html)

At our location (latitude ~46.8), 1 degree longitude is ~ 76339m and 1 degree latitude is ~ 111167m.

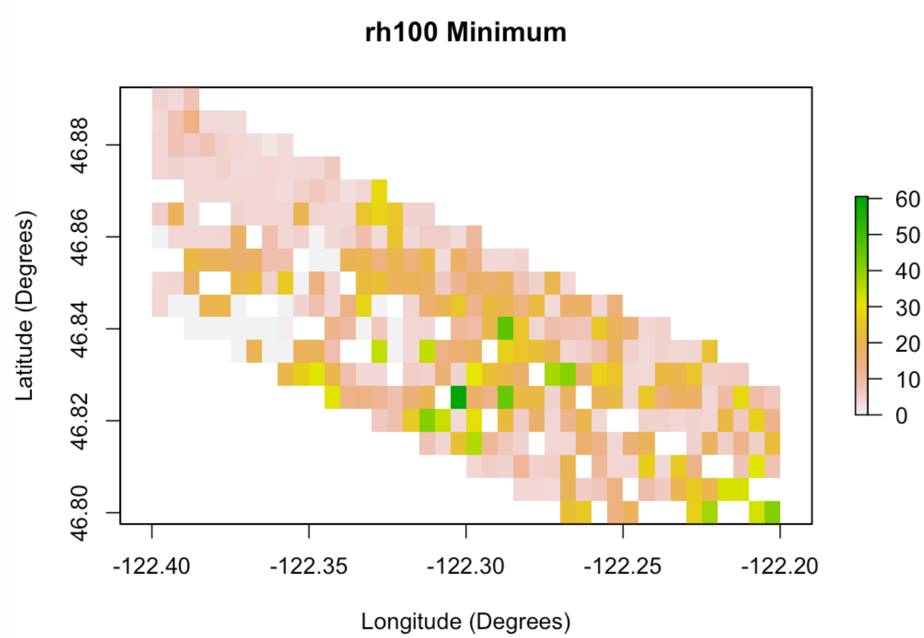
```
?gridStatsLevel2AM
rh100metrics <- gridStatsLevel2AM(level2AM = level2AM, func = mySetofMetrics(rh100),
res = 0.005)
```

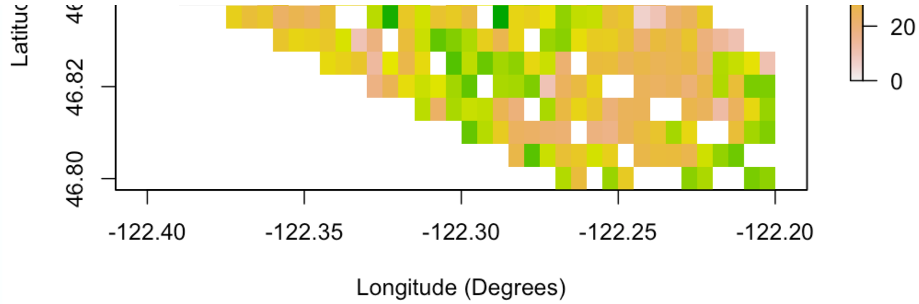
QUESTION 14. At our Pack forest site, what is the length of 0.005 degrees in meters for longitude and latitude?

A spatial resolution of hundreds of meters doesn't sound that good, but considering that GEDI data is aiming for near global coverage, the resolution becomes more reasonable.

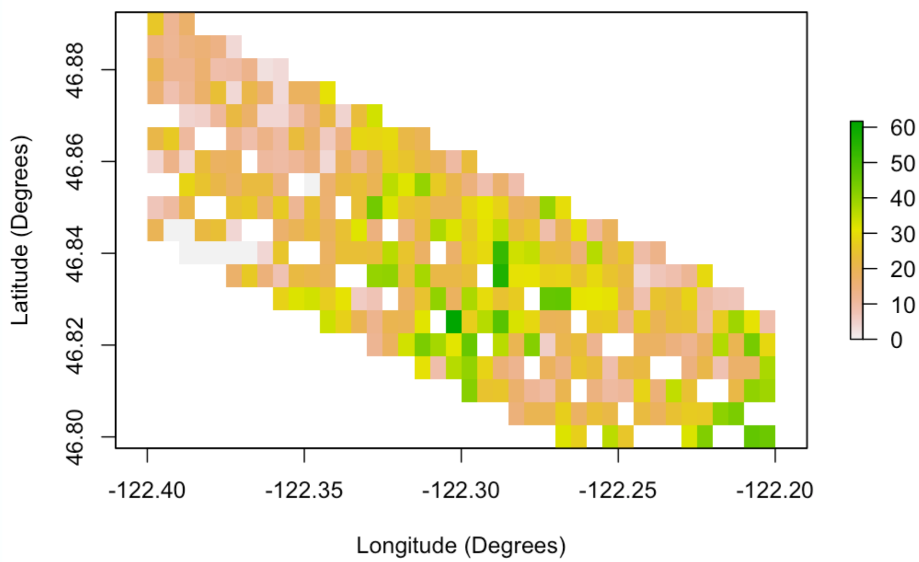
Lets plot our height metrics:

```
?levelplot

plot(rh100metrics$min, main = "rh100 Minimum", xlab = "Longitude (Degrees)", ylab = "
Latitude (Degrees)")
plot(rh100metrics$max, main = "rh100 Maximum", xlab = "Longitude (Degrees)", ylab = "
Latitude (Degrees)")
plot(rh100metrics$mean, main = "rh100 Mean", xlab = "Longitude (Degrees)", ylab = "La
titude (Degrees)")
plot(rh100metrics$sd, main = "rh100 Standard Deviation", xlab = "Longitude (Degree
s)", ylab = "Latitude (Degrees)")
```

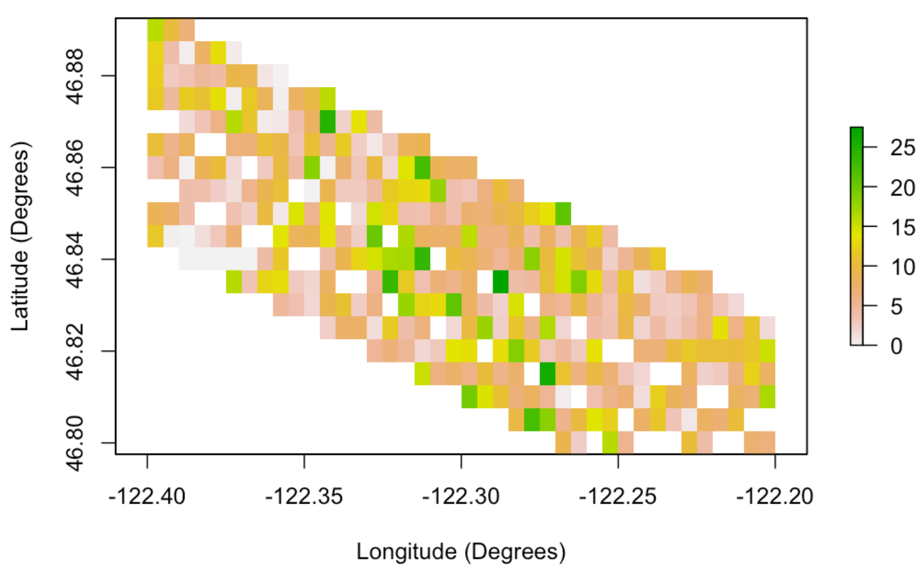




rh100 Mean



rh100 Standard Deviation



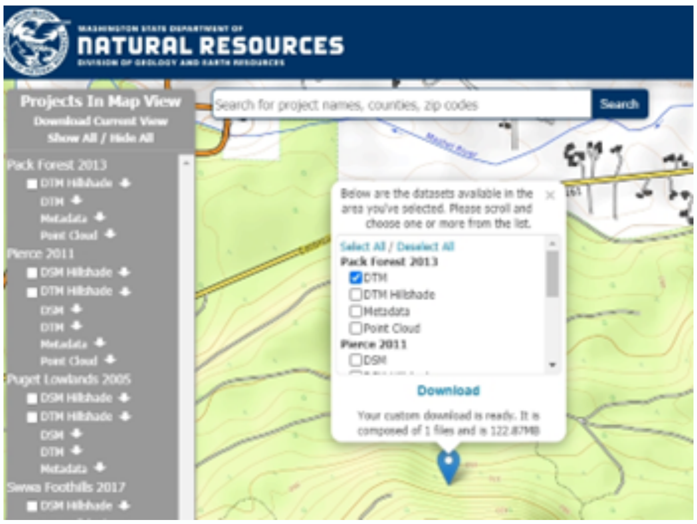
Now we can save these as rasters for viewing in ArcGIS Pro

```
crs(rh100metrics) <- "EPSG:4326"

writeRaster(rh100metrics$min, "Lab9/Lab9Data/rh100_min.tif", overwrite = TRUE)
writeRaster(rh100metrics$max, "Lab9/Lab9Data/rh100_max.tif", overwrite = TRUE)
writeRaster(rh100metrics$mean, "Lab9/Lab9Data/rh100_mean.tif", overwrite = TRUE)
writeRaster(rh100metrics$sd, "Lab9/Lab9Data/rh100_sd.tif", overwrite = TRUE)
```

QUESTION 15: Submit 4 screenshots of your four rh100 plots. Caption them with a full description of what they represent.

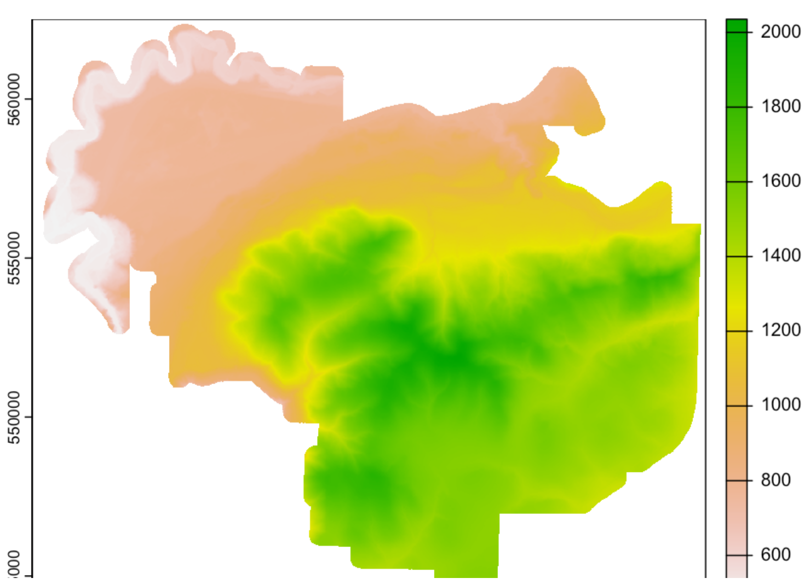
Part 7: Compare to lidar…

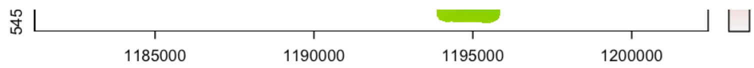
From the Washington DNR lidar portal, grab the DTM from PackForest if you don't already have it. Just the one file, should be 122.87MB.



Bring the DTM into R:

```
PFDTM <- rast("Lab9/Lab9Data/pack_forest_2013_dtm_1.tif")
plot(PFDTM)
```





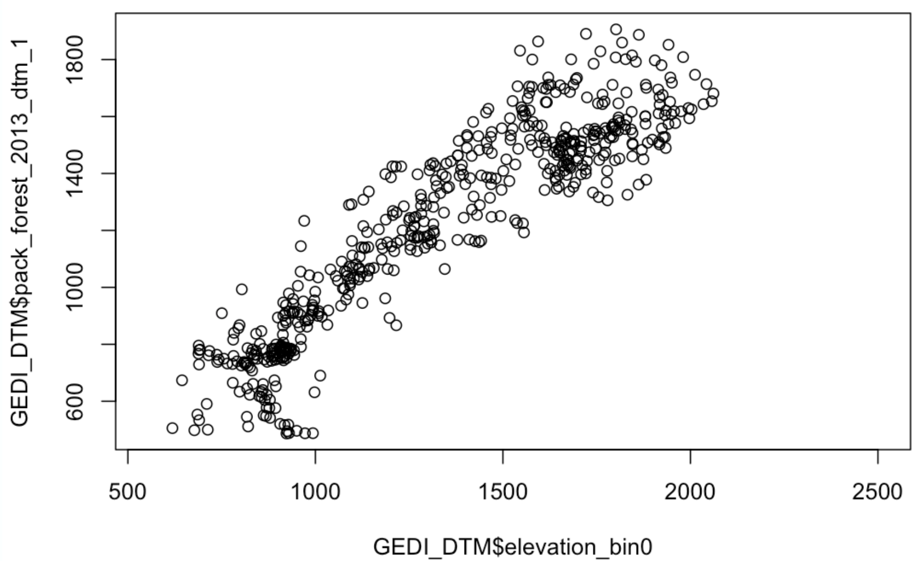
Lets check how well the lidar DTM agrees with the elevation values that were captured in the GEDI level 1b data. We already have the level 1b data in a spatial data frame so we can overlay the points across the lidar DTM and extract the DTM values at each point.

This code extracts the DTM values with the GEDI datapoints we converted to an `sf` spatial object

```
level1BGeo_sf_prj <- st_transform(level1BGeo_sf, crs(PFDTM))

GEDI_DTM <- terra::extract(PFDTM, level1BGeo_sf_prj, bind=TRUE)
GEDI_DTM <- na.omit(GEDI_DTM) #remove all NA rows
GEDI_DTM$elevation_bin0 <- GEDI_DTM$elevation_bin0*3.28
```

Now we can plot the elevation from GEDI and the elevation from the DTM together

```
plot(GEDI_DTM$elevation_bin0, GEDI_DTM$pack_forest_2013_dtm_1)
```



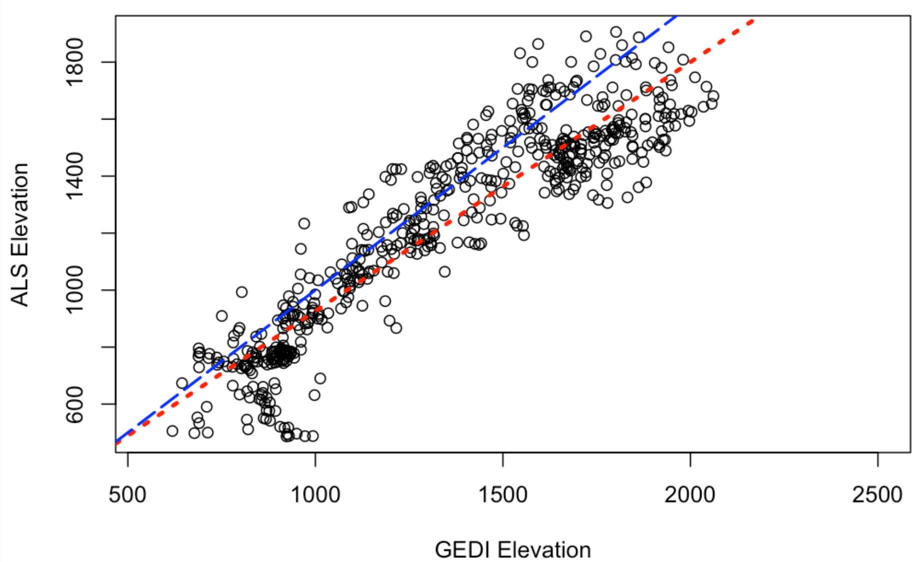
Let’s evaluate further and make a linear model

```
reg <- lm(GEDI_DTM$pack_forest_2013_dtm_1 ~ GEDI_DTM$elevation_bin0)
summary(reg)
```

```
##
## Call:
## lm(formula = GEDI_DTM$pack_forest_2013_dtm_1 ~ GEDI_DTM$elevation_bin0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -432.83  -87.02  -10.53   89.40  428.18
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             52.54234   22.09838   2.378   0.0178 *
## GEDI_DTM$elevation_bin0  0.87388    0.01598  54.699   <2e-16 ***
## ---
```

Now let's make a new plot with the regression line

```
plot(GEDI_DTM$elevation_bin0, GEDI_DTM$pack_forest_2013_dtm_1, xlab = "GEDI Elevatio
n", ylab="ALS Elevation")
abline(reg, lty=3, col ="red", lwd = 3)
abline(0, 1, col = "blue", lty = 5, lwd=2)
```



QUESTION 16: Submit a screenshot of the scatterplot but using different lty and lwd. Report adjusted R^2. Provide a full caption of the figure.

LAST STEP!

Use the code below to write your GEDI points to a ESRI shapefile

```
write_sf(level1BGeo_sf_prj, "Lab9/Lab9Data/level1BGeo_sf_prj.shp")
```

```
## Warning in abbreviate_shapefile_names(obj): Field names abbreviated for ESRI
## Shapefile driver
```

QUESTION 17: Bring your ALS DTM and your level1BGeo_sf_prj point layers in ArcGIS, use the symbology in ArcGIS to change the colors of the level1BGeo_sf_prj symbols to be graduated colors with the elevation_bin0 field. Create a map with level1BGeo_sf_prj points overlaid on the pack_forest_2013_dtm. Fully caption the figure.