ESRM433/SEFS533
Lab 5

Objectives:

- Relating plot data to lidar data
- Using R for linear regression analysis
- This will be a crash course in simple linear regression and multiple linear regression for those of you that don't have a stats background… please help each other.

Data and Software:

- LAB5Data.zip folder, downloadable from Canvas
- R Studio
- CloudCompare

What you will turn in:

- Submission Template in PDF or DOCX file format submitted via Canvas

Welcome to Lab 5 for ESRM433/SEFS533!

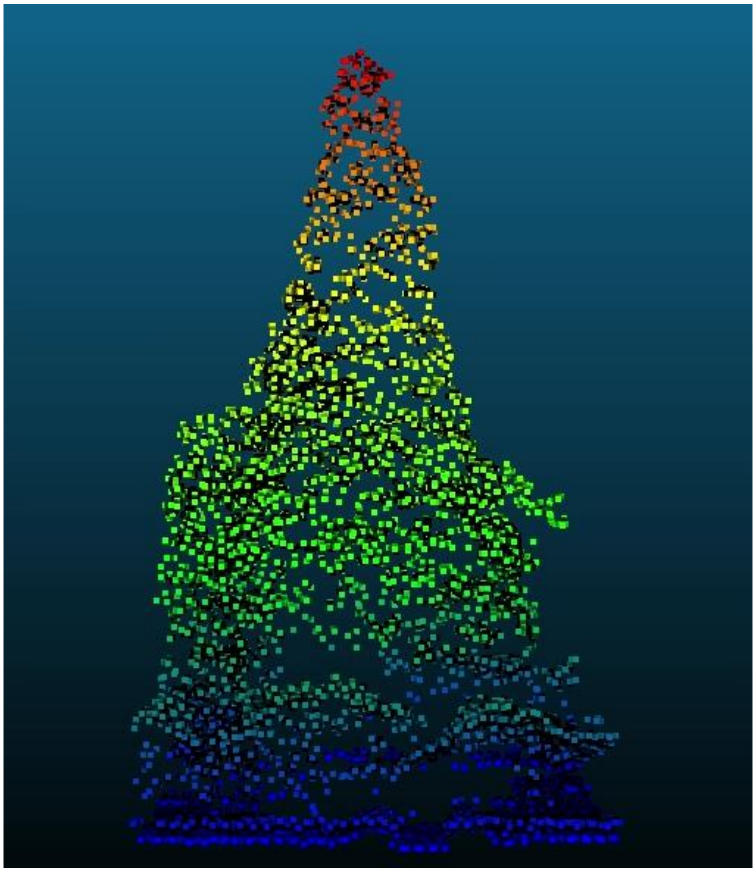
This lab is all about cloud metrics and relating them back to field data of typical forest inventory metrics.

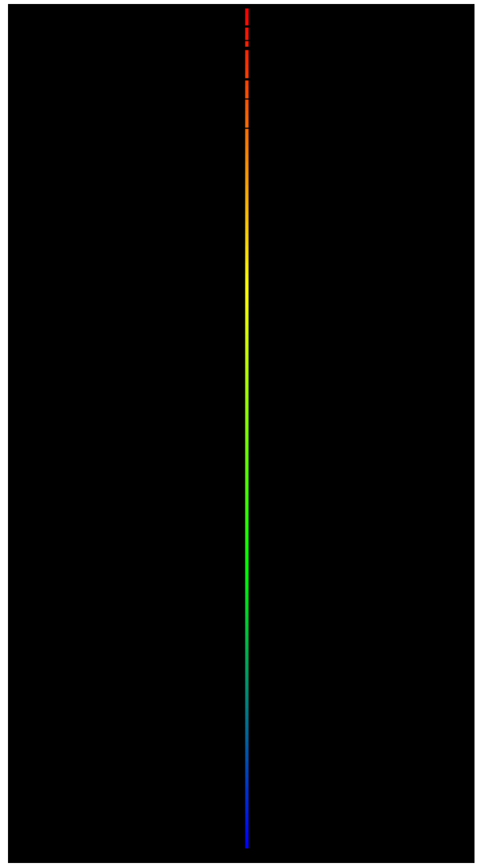
"Cloud Metrics" is a very broad term that can refer to any values derived from a point cloud, at a spatial resolution defined by the user. Typical cloud metrics relate to either the elevation or intensity of points within the defined area. Elevation is what we will be focusing on in this lab. We've used a very basic cloud metric in the previous lab in finding the z max (highest point) within a convex hull that represented a tree crown. Beyond just finding the maximum, mean, or minimum values within an area, we can also find the elevation of points in a defined percentile, and the distribution of points across all elevation values.
Quantile and percentile maybe used to describe the distribution depending on what software you are using. From Statsdirect.com

> "Quantiles are points in a distribution that relate to the rank order of values in that distribution.
> For a sample, you can find any quantile by sorting the sample. The middle value of the sorted sample (middle quantile, 50th percentile) is known as the median. The limits are the minimum and maximum values. Any other locations between these points can be described in terms of centiles/percentiles.
> Centiles/percentiles are descriptions of quantiles relative to 100; so the 75th percentile (upper quartile) is 75% or three quarters of the way up an ascending list of sorted values of a sample. The 25th percentile (lower quartile) is one quarter of the way up this rank order.
> Percentile rank is the proportion of values in a distribution that a particular value is greater than or equal to. For example, if a pupil is taller than or as tall as 79% of his classmates then the percentile rank of his height is 79, i.e. he is in the 79th percentile of heights in his class."
> https://www.statsdirect.com/help/nonparametric_methods/quantiles.htm

There are other methods to derive cloud metrics and we will cover those later in this lab. For now, we will be focusing on elevation.

If using elevation, it is important to remember that it is only the z values that are being used for the metrics. All x and y values are effectively ignored.
If you stacked all points in a single column (i.e. set all x and y values to 0), you would still get the same percentile values and distribution statistics as you would get if you were analyzing a full point cloud.





Let's take a look at the Giant Sequoia *(Sequoiadendron giganteum)* that is across the street from Anderson hall here at SEFS.

PART 1: Looking at Cloud Metrics

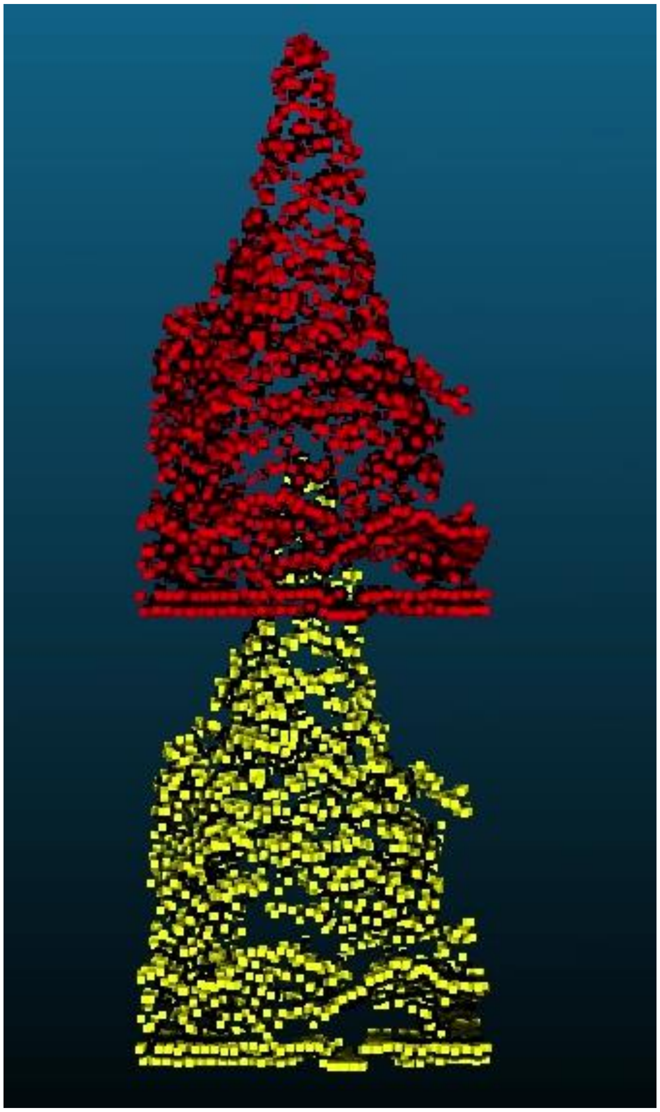
In the lab 5 data folder there is a relatively small las file: Sequoia.las
Make your LAB5 folder in your ESRM433 folder on your desktop and put all the lab data there.

We are going to normalize the point cloud in R, visualize the point cloud in CloudCompare, then compare the outputs from R to outputs from Excel.

Open up R studio and get your environment set up but setting your working directory and loading the lidR library.

Create a normalized las file from Sequoia.las and save it into your LAB5 folder:

```
LASfile <- ("LAB5/Sequoia.las")
las <- readLAS(LASfile)
lasNORM <- normalize_height(las,tin())
writeLAS(lasNORM, file = "LAB5/SequoiaNorm.las")
```



Load both Sequoia.las and SequoiaNorm.las into CloudCompare to visualize the normalization.

I set my colors with Edit > Colors > Set Unique

The red is the original Sequoia.las file and the yellow is SequoiaNorm.las. The height difference is elevation being subtracted from the points.

In R, take a moment to familiarize yourself with lidR's cloud_metrics function. You can do a simple task with cloud_metrics and find out what the maximum z value is, in your normalized point cloud:

```
?cloud_metrics
cloud_metrics(lasNORM, ~max(Z))
```

QUESTION 1: You are able to create your own metrics using cloud_metrics, but there are 4 existing functions that lidR can use, what are they? There are 6 papers cited for the metrics. List 4 of the papers used as reference for the different functions. Copy and pasting from the documentation is fine.

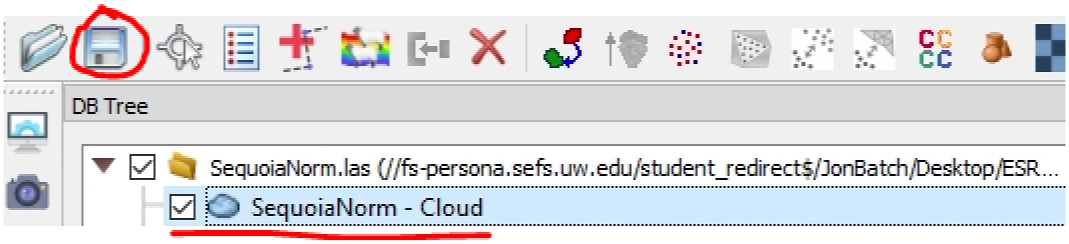
There are no "best" metrics to derive from a point cloud. The best metrics totally depend on the question being asked and must be ecologically defensible.
We will be focusing on .stdmetrics_z

To generate all stdmetrics for our las file, and then write them to a csv file, run:

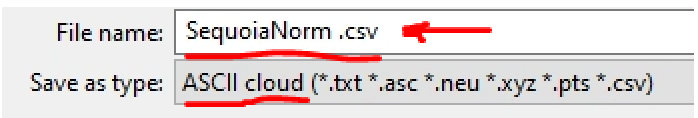
```
CMSequoia <- cloud_metrics(lasNORM, .stdmetrics)
write.csv(CMSequoia, file = "LAB5/CMSequoia.csv")
```

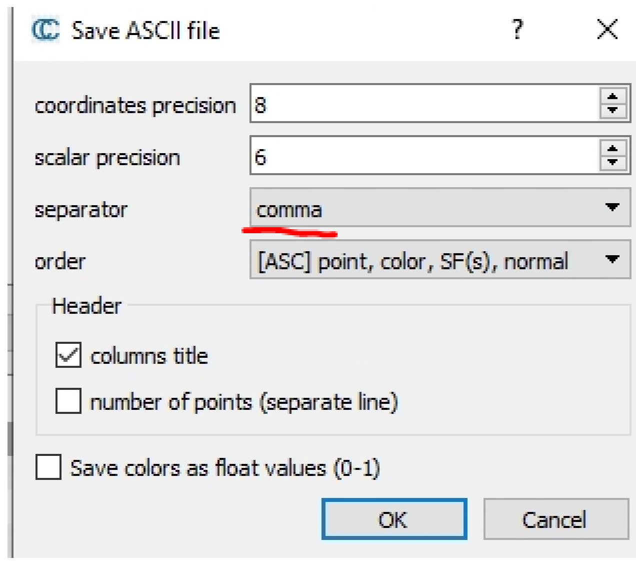
So far, we have only worked with lidar in a las or laz format. Lidar data can also be saved in an ASCII format that is readable by spreadsheet programs like Excel.

In CloudCompare, save your SequoiaNorm – Cloud as a csv file:

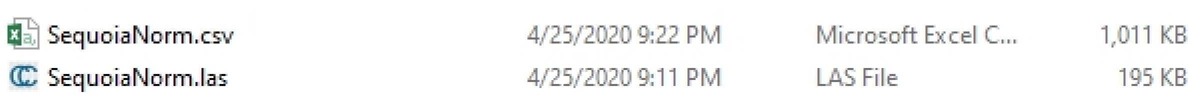


Make sure you type in the .csv and that you select comma as your separator.





Note how much bigger the .csv file is than the .las file. ASCII is a very inefficient storage format for lidar data, but it is helpful as you can directly edit the points in Excel…



Open your SequoiaNorm.csv and your CMSequoia.csv in Excel.

In your SequoiaNorm.csv the column headers are your las data fields such as PointSourceID, Intensity, Classification, etc.

In CMSequoia.csv the column headers are the standard output cloud metrics. There are a lot… The code to understanding them is in the ?cloud_metrics documentation:

- z: refers to the elevation
- i: refers to the intensity
- rn: refers to the return number
- q: refers to quantile
- a: refers to the ScanAngleRank or ScanAngle
- n: refers to a number (a count)
- p: refers to a percentage

pzabovezmean means “Percentage of points with an elevational value above the overall elevational mean”

All of the “z” values in this output were derived from the single “z” column in your SequoiaNorm.csv. All of the “I” values in CMSequoia.csv were derived from the single “Intensity: column in your SequoiaNorm.csv.

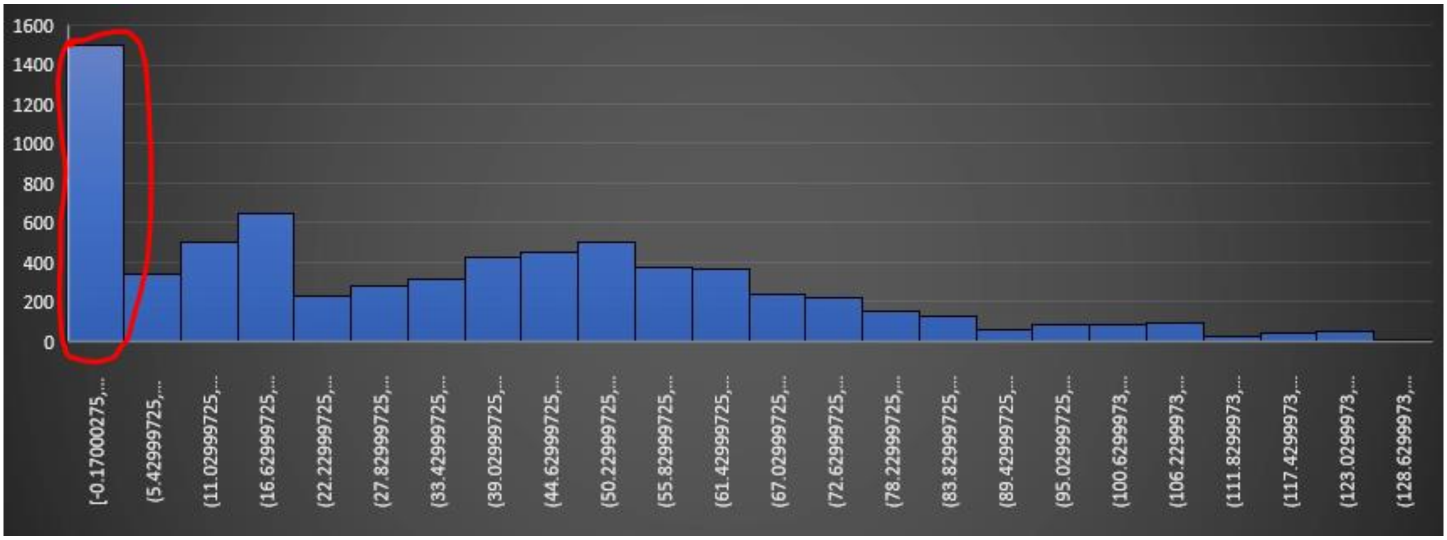
Highlight your Z column in SequoiaNorm.csv and create a histogram chart (should be suggested in your “recommended charts”)

A significant portion of the points are in the lowest bin. These are the ground points in the cloud. If we want information about the trees, we should remove the heavy weighting of the data by the ground points, so that we are only looking at points that are returns from the plants at a location.

QUESTION 2: Include a screenshot of your histogram for SequoiaNorm.csv. Briefly discuss the shape of the histogram and how it relates to vegetation.

QUESTION 3: What are the values for the elevation 95th and 25th percentile from your CMSequoia.csv?

If you run descriptive statistics in Excel on your z column for SequoiaNorm.csv, you should get the same values for the zmean, zsd, zskew, and zkurt as what is in CMSequoia.csv.



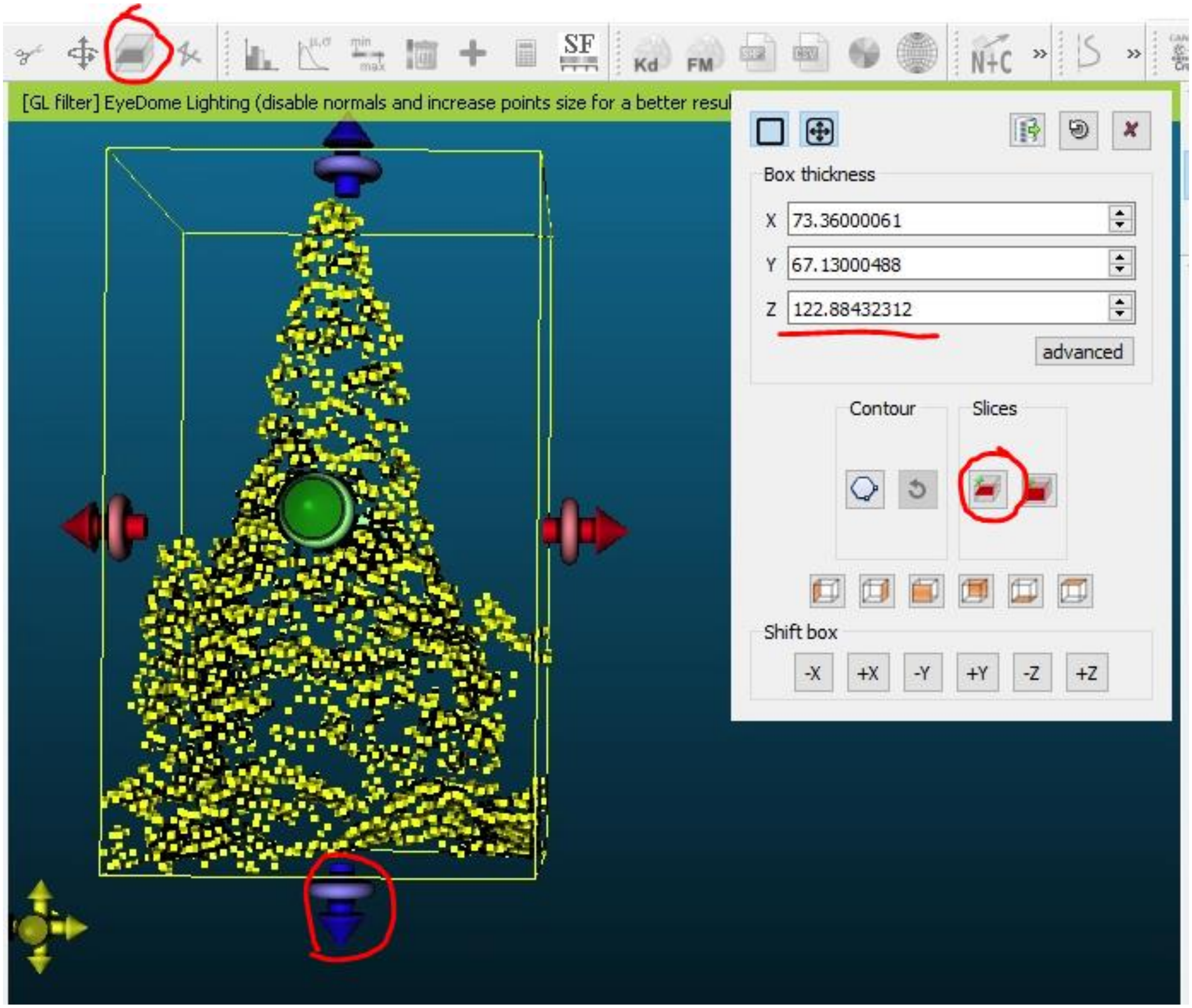
While it isn't a perfect solution, if we remove all the points below 2 meters from a normalized point cloud, we will likely be looking mostly at points from trees and tall shrubs. This is common practice in studies using lidar to look at forest structure.

To clip out the bottom 2 meters from our Sequoia lidar, run:

```
over2 = filter_poi(lasNORM, Z > 6.56)
CMSequoia2 <- cloud_metrics(over2, .stdmetrics)
write.csv(CMSequoia2, file = "LAB5/CMSequoia2.csv")
```

You used the lasfilter function before when you created the point cloud with only points that contained a treeID in the last lab. That above code also creates a new csv with stdmetrics for all points above 2m. Remember that data from the Washington DNR lidar portal is in feet, so 2 m = 6.56 ft.

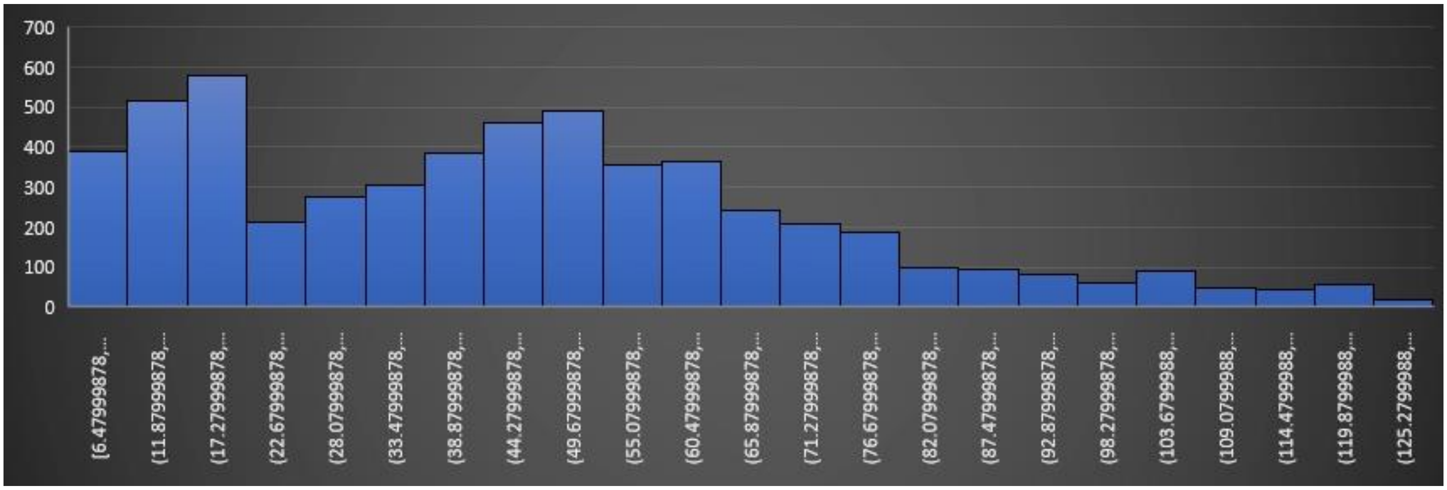
In CloudCompare, use the cross section tool to manually clip out the approximate bottom 2 m. Slide up the bottom blue arrow until the Z value is as close to 122.92 (129.48 – 6.56) as you are able.



Save your cropped point cloud as SequoiaNorm2.csv following the same steps as above, and open it in Excel. Follow the instructions to make a histogram from your SequoiaNorm2.csv column of z values.

QUESTION 4: What are the values for the elevation 95th and 25th percentile from your CMSequoia2.csv?

QUESTION 5: Include a screenshot of your Z column histogram and mark on it the approximate location of the 95th and 25th percentile.



QUESTION 6: Given your knowledge of how trees grow, why do you think there is a drop in the point count at right around the 25th percentile height value?

QUESTION 7: Why would p95 be used as a measure of tree height instead of the maximum z value?

QUESTION 8: Compare the stdmetrics from CMSequoia.csv and CMSequoia2.csv. Which metrics are similar and which ones differ the most? Why?

Take this opportunity to associate the histogram values and point distribution with the visual point cloud rendered in CloudCompare. Understanding what the cloud metric numbers actually represent in how a point cloud "looks" is important, but more important, is to understand how the point distributions in a point cloud relate to ecological parameters.

Interlude: Direct vs Derived lidar metrics & Understanding field metrics

A **direct** metric is something like p95 where it is a count of a number of points within a certain area, that meet a certain criterion. P95 is often used as a direct measurement for tree height. A **derived** metric is something that wasn't directly measured by lidar but that can be inferred. Metrics like Quadratic mean diameter (QMD) or tree counts are derived. The procedure you will follow is very common in the remote sensing world: first create models associating remote sensing metrics with ground-truth metrics at a number of plots, then use the wall-to-wall coverage of remotely sensed metrics to predict the model across large, continuous areas.

Quadratic mean diameter

In forestry, quadratic mean diameter or QMD is a measure of central tendency which is considered more appropriate than arithmetic mean for characterizing the group of trees which have been measured. For n trees, QMD is calculated using the quadratic mean formula:

$$\sqrt{\frac{\sum D_i^{\,2}}{n}}$$

where D_i is the diameter at breast height of the i^{th} tree. Compared to the arithmetic mean, QMD assigns greater weight to larger trees. QMD is always greater than or equal to arithmetic mean for a given set of trees. QMD can be used in timber cruises to estimate the standing volume of timber in a forest, because it has the practical advantage of being directly related to basal area, which in turn is directly related to volume.

You will need to be familiar with basic forest ecology measurements and understand plot data collection.

Here is an example of very basic observations that can be measured at field sites, that can later be related to lidar data and extrapolated across an entire acquisition. These plots were 1/10 ha circles (17.85 m radius).

PlotNumber	Tree_No	TreeSpeciesCD	TreeDBH (in)	TreeHeight (ft)	Uncompacted Crown Ratio (%)	Compacted Live Crown Ratio (%)
201	1	PILA	14.2	40.7	90	80
201	2	PIJE	14.3	23.3	80	70
201	3	ABCO	26.5	64.7	100	70
201	4	PILA	12.6	35	95	90
201	5	PIJE	21.7	54	75	60
201	6	PIJE	8.5	33.5	65	55
201	7	PIJE	17.2	38	60	50
201	8	PIJE	12	6.8	0	0
404	1	PIJE	52.6	150.8	55	50
404	2	ABCO	16.5	38.8	95	90
404	3	ABCO	23.6	53.1	75	40
404	4	PIJE	47.7	62.9	45	35
404	5	ABCO	17	68.1	65	10
404	6	PIJE	31.8	80.1	75	65
404	7	PIJE	34.9	106.1	75	70
404	8	ABCO	6.2	18.4	100	95

Uncompacted crown ratio: The percent of the total length of the tree which supports a full crown of live or dead branches.

Compacted live crown ratio: The percent of the total length of the tree which supports a full, live crown. For trees that have uneven length crowns, ocularly transfer lower branches to fill holes in the upper portions of the crown, until a full, even crown is created.

Three measurements that are particularly helpful to relate to lidar data are Quadratic Mean Diameter (QMD), Basel Area (BA), and Trees Per Acre or Hectare (TPA or TPH). To complicate things, we have to pay close attention if the units are metric or imperial. The above units are imperial (us-ft) for trees but the plot size is defined in metric units. One day the antiquated imperial measurements will be abolished.

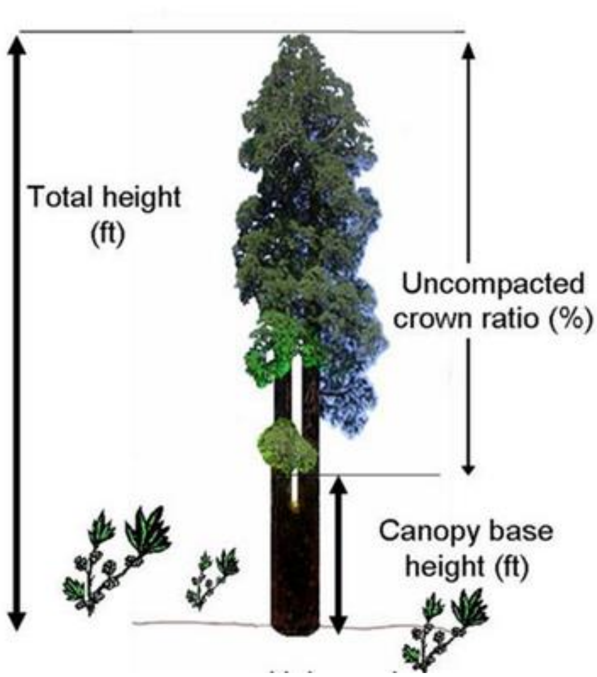
Basal area: Tree area in square feet or meters of the cross section at breast height of a single tree. When the basal areas of all trees in a stand are summed, the result is usually expressed as square feet of basal area per acre or square meters of basal area per hectare.

Plot 201

- QMD, using the equation provided above, is 16.8in per plot or 42.6cm per plot.
- For basal area and TPH we have to upscale our plot size from 1/10ha to a full hectare by simply multiplying our factored TPH or BA by 10. TPH is easy, there are 8 trees in 1/10ha so the TPH (assuming even distribution of trees) is 80. The TPA is slightly more difficult as there are 2.47 acres in one hectare. If the TPH is 80, then the TPA is 80/2.47. The TPA is 32.4.
- For basal area, we first figure out the area of the trees at DBH within the plot. Remember that the area of a circle is $A = \pi r^2$, and we have the diameter of all the trees. The BA per plot is $1767in^2$ or $12.27ft^2$. We can convert $12.27ft^2$ to $1.14m^2$ and then upscale for a metric BA of $11.4m^2$ per hectare or keep it imperial with $(12.27ft^2*10)/2.47$ which will give us $49.67ft^2$ per acre.
- The uncompacted crown ratio and compacted live crown ratio are expressed as percentages of the tree height. To calculate the average uncompacted crown length or average compacted live crown length is simply

$$\frac{\sum(T_i * R_i)}{n}$$

where T_i is the height of the i^{th} tree and R_i is either the uncompacted crown ratio or the compacted live crown ratio of the i^{th} tree. n is the number of trees.
Ignoring the one tree missing ratio information, the average uncompacted crown length for plot 201 is 34.04ft and the average compacted live crown length for plot 201 is 27.93ft.



- What might be more meaningful is to get the average crown height base from the crown ratio information. Ideally you would have a direct measurement of the crown base height from the field. This can be derived using the uncompacted crown ratio values and the height of trees.

$$\frac{\sum T_i - (T_i * R_i)}{n}$$

Ignoring the one tree missing ratio information, the average crown base height for plot 201 is 7.27ft

QUESTION 9: What is the QMD, BA, TPA/TPH and the average uncompacted crown length for plot 404? Give your answer in both imperial and metric units. Careful not to mix the units.

If an animal species is known to have certain habitat preferences, it may be possible to identify some of those preferences across a landscape using lidar.
For example, key habitat components to consider in the identification of spotted owl nesting/roosting habitat in eastern Washington includes (based on Gaines et al. 2015):

- Forest types: Dry forest, mesic forest, cold-moist forest
- Medium and Large trees, preferably Douglas-fir when appropriate to the forest type (>15 inches QMD [see below])
- High canopy closure (>70 percent) and two or more canopy layers
- Presence of mistletoe brooms
- Snags and Coarse Woody Debris (CWD) in variable abundance and a diversity of size classes, including large sizes

QUESTION 10: Not all of the habitat components believed to be preferred by spotted owls can be quantified using ALS, but some can be. Review the cloud metrics you've created. What habitat components for spotted owl nesting/roosting areas do you think lidar can be used to either directly measure, or can be derived using lidar metrics? Don't just list the components, discuss how lidar can be used to assess the component. There isn't one correct answer to this question, feel free to speculate. As long as the reasoning is sound, and the answer is well developed (i.e. not just a single sentence) you will get full credit.

PART 2: Linear Regression

You will be using linear regression to come up with relationships between LiDAR metrics and field metrics. If you are not familiar with linear models, here is a very brief background. Quantitative models try to determine the effect of an independent (explanatory) variable on a dependent (response) variable. They ask the question: for every unit that my independent variable changes, how many units does my dependent variable change by? Often you will have a few or several independent variables, and you want to know how each of those explains some of the observed changes in the dependent variable.

Linear models in particular try to explain the variation in the dependent variable as a linear combination of the independent variables. Geometrically, that means that each of your independent variables defines the slope of a line, and the sum of all of those lines is as close as possible to your dependent data points.

There are several common methods used to evaluate the quality of linear models. Today you will look at r^2 values and *p*-values. An r^2 value gives an indication of how much of the variance in your dependent variable is explained by the independent variables. It can range from 0 to 1, with 0 being no relationship and 1 being a perfect fit There is no specific guideline as to what is an "acceptable" r^2 value. It is relative to what the dependent and independent variables are. Usually, metrics like tree height – which is measured well by LiDAR – have good r^2 values of 0.9 or better. However, metrics like stand density – which is hard to measure with LiDAR – has lower r^2 values of 0.3-0.5.

A *p*-value gives an indication of the significance of the model, and of the individual terms in the model. In this case significance means “probability that this result is due to chance.” The values for *p* also range from 0 to 1, where 0 would mean that the result is very significant and 1 would mean that the result has no significance. The *p*-values from linear models are simply taken from the model’s ANOVA table. The *p*-values for each term give some idea of how important the term was in the model. The *p*-value for the overall model gives some idea of whether the model is credible. Usually overall *p*-values for linear models are very low. If a *p*-value for a linear model is greater than 0.1, the model is probably not useful.

In the LAB5Data folder, there are two csv files, <u>cloudmetrics.csv</u> & <u>plotdata.csv</u>. These are plot data for 79 plots located on the east side of the cascade mountains. Plot data is from field measurements while cloud metrics are taken from the ALS of the plot locations.

You are first going to use R to do some basic multiple linear regression models using the data to look for relationships, and then in the next step you are going to use lidR to clip ALS data and to plot locations and use that data to run more basic statistics. Lidar data lends itself well to many statistical approaches from basic linear regression models to multivariate analysis. We will only be covering some of the basics in this class.

A basic introduction for multiple linear regression in R:
https://www.youtube.com/watch?v=q1RD5ECsSB0

For more information about regression analysis using lidar:
Hudak, Andrew T., et al. "Regression modeling and mapping of coniferous forest basal area and tree density from discrete-return lidar and multispectral satellite data." Canadian Journal of Remote Sensing 32.2 (2006): 126-138.

First thing you will want to do is to import your csv files into R. Run:

```
field <-(read.csv("LAB5/plotdata.csv"))
lidar <- (read.csv("LAB5/cloudmetrics.csv"))
```

You can then click on the data set you imported to get a better look at the columns and values contained.

field	79 obs. of 17 variables
lidar	79 obs. of 6 variables

ESRM433/SEFS533
Lab 5

LAB4.R* | lidar | field

	FileTitle	Percentage.all.returns.above.6.56	Elev.mean	Elev.P25	Elev.P95	Elev.stddev
63	d6096a22-7bfd-4fc6-ad87-11de4cf6fc6f	16.610169	3.738831	2.817910	5.917154	1.168026

For the lidar data, the file Title is just a randomized value used to represent each plot. Percentage.all.returns.above.6.56 is very descriptive of what it is, but more importantly, this is a good direct measurement of the canopy closure. If only 16.6% of all returns are above 2 meters, then the canopy is very open allowing most of the points to penetrate through.
The Elev metrics describe the Z values of all points within the plot.

You know that cloud_metrics in lidR derives many more statistics than this, but for this exercise, we are only going to focus on these 5.

LAB4.R* | lidar | field

TPA	BA	SDI	QMD.GT5	QMD.25pct	QMD.50pct	QMD.75pct	TPA.21	TPA.GT5	BA.GT5	SDI.GT5	TotalCu
40.08	63.01913	93.86550	16.979105	19.039870	19.764868	21.300000	10.02	40.08	63.01913	93.86550	1904.

The field data collected contains many more variables. The only ones that we are going to focus on in this lab are:
TPA, trees per acre
BA, basal area
QMD.GT5, quadratic mean diameter of trees greater than 5 inches in diameter

Let's first try some simple regression analysis with one independent variable.
Check out the lm function in R and identify the data you want to use as your dependent and independent variables.

```
?lm
Y <- field$QMD.GT5 #Dependent Variable. Identifies QMD.GT5 column from the field dataset
X <- lidar$Elev.P95 #Independent Variable. Identifes Elev.P95 column from the lidar dataset
```

For this, we are using QMD as the dependent variable and tree height (P95) as the independent variable.

Run a simple linear regression in R and then print the summary:

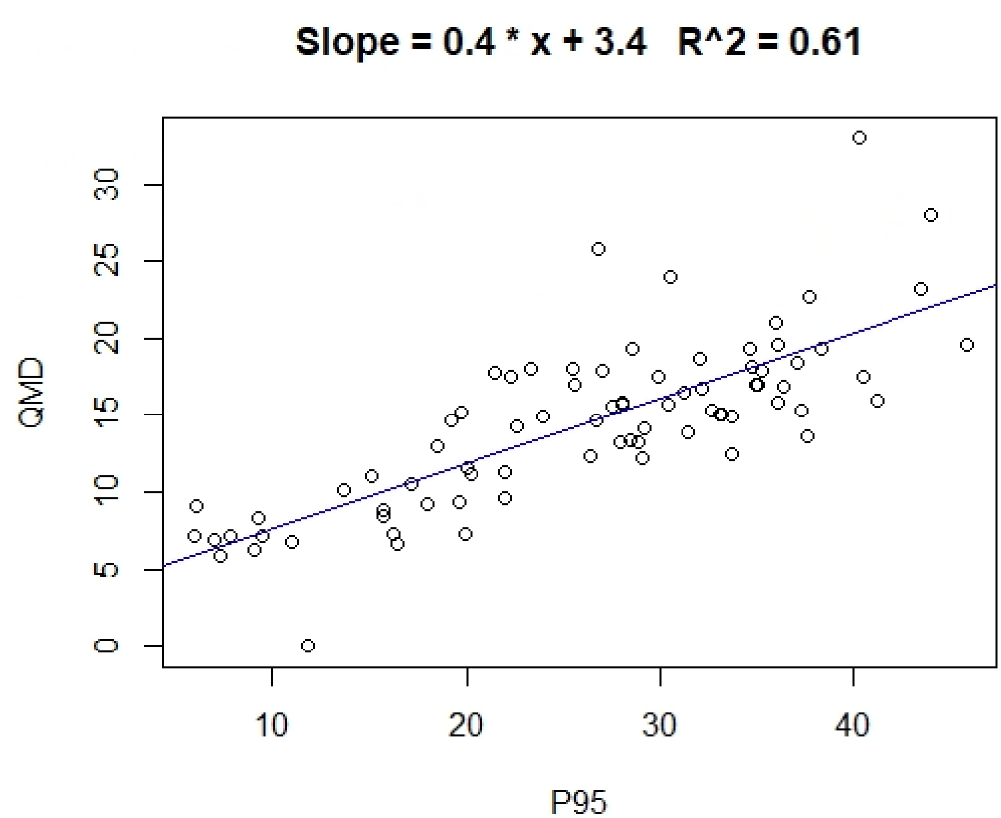
```
reg <- lm (Y ~ X) # reg is the regression output from Y & X
sum <- summary(reg) # sum is the summary statistics of the linear regression of Y & X
sum # prints the summary statistics. p-value and R-squared value are of the most interest to us
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.42303    1.07305    3.19  0.00206 **
X            0.42353    0.03824   11.07  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.418 on 77 degrees of freedom
Multiple R-squared:  0.6143,    Adjusted R-squared:  0.6093
F-statistic: 122.7 on 1 and 77 DF,  p-value: < 2.2e-16
```

We have a p-value of < 2.2e-16 which indicates a relationship between tree height and QMD. Our Y intercept is ~3.42 and our QMD value increases approximately 0.42 units with every one unit increase in P95. Our regression using approximate tree height, describes ~ 61% of the variance in the QMD for the plots.

Let's Plot it!

```
coeff <- coefficients(reg) #coefficients are the slope of the regression line
#eq will be the title of the plot with slope and R-Squared values
eq <- paste0("Slope=",round(coeff[2],1),"*x+",round(coeff[1],1)," R^2=",round(sum$r.squared, 2))
plot(X, Y, xlab="P95", ylab="QMD", main=eq) #plots the points and title
abline(reg, col="blue") #adds the regression line to the plot
```



With the above script, you can change what X and Y are, to look for relationships between other metrics.

```
Y <- field$BA #Dependent Variable.
X <- lidar$Elev.P25 #Independent Variable.
```

Once X and Y are changed, you can just run the rest of the lines of code unchanged.

QUESTION 11: What is the P-value, slope, and R^2 with BA as the dependent variable, and Elev.P25 as the independent variable? Include a screen shot of your output graph. Make sure to change (xlab="P25", ylab="BA"). Caption your output graph.

QUESTION 12: What lidar metric has a statistically significant relationship to TPA (i.e. p value less than 0.05). Remember, you can have a low p-value and still have a low R^2 value. How do you think the metric you identified could be ecologically related to TPA?

Often a combination of independent variables can produce a better model for a dependent variable. This is called multiple linear regression.

In the plotdata.csv, there is a column "Carbon.AB". Let's create a multiple linear regression looking at all of our lidar metrics combined, and their potential relationship to Carbon.AB. In R you can run a multiple linear regression by simply adding more + terms. Run:

```
Mreg <-lm(field$Carbon.AB ~ lidar$Elev.stddev + lidar$Elev.mean +
lidar$Percentage.all.returns.above.6.56 + lidar$Elev.P95 + lidar$Elev.P25)
summary (Mreg)
```

If your top line ends in a "+" R will look to the next line of code. Alternatively, you can enter it all on line like like:

```
Mreg <-lm(field$Carbon.AB ~ lidar$Elev.stddev + lidar$Elev.mean + lidar$Percentage.all.returns.above.6.56 + lidar$Elev.P95 + lidar$Elev.P25)
```

```
Coefficients:
                                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)                             -67497.20   14819.20  -4.555 2.06e-05 ***
lidar$Elev.stddev                           34.41   11740.51   0.003  0.99767    
lidar$Elev.mean                          15684.08    9322.22   1.682  0.09676 .  
lidar$Percentage.all.returns.above.6.56    848.86     262.78   3.230  0.00186 ** 
lidar$Elev.P95                           -3380.24    3505.38  -0.964  0.33808    
lidar$Elev.P25                           -4558.77    7562.24  -0.603  0.54849    
```

We want to simplify our multiple linear regression so we are going to eliminate the metric with the highest p-value and run the regression again.

```
Mreg <-lm(field$Carbon.AB ~ lidar$Elev.mean +
          lidar$Percentage.all.returns.above.6.56 + lidar$Elev.P95 + lidar$Elev.P25)
```

```
Coefficients:
                                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)                              -67489.2    14464.8  -4.666 1.34e-05 ***
lidar$Elev.mean                           15667.6     7377.4   2.124  0.03704 *  
lidar$Percentage.all.returns.above.6.56     848.8      260.2   3.262  0.00168 ** 
lidar$Elev.P95                            -3386.5     2764.3  -1.225  0.22444    
lidar$Elev.P25                             4541.4     4643.4  -0.978  0.33125    
```

Repeat, removing Elev.p25

```
Mreg <-lm(field$Carbon.AB ~ lidar$Elev.mean + lidar$Percentage.all.returns.above.6.56 + lidar$Elev.P95)
```

```
Coefficients:
                                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)                              -70457.1    14138.8  -4.983 3.90e-06 ***
lidar$Elev.mean                            8682.6     1848.6   4.697 1.17e-05 ***
lidar$Percentage.all.returns.above.6.56     872.1      259.1   3.366   0.0012 ** 
lidar$Elev.P95                            -1040.7     1374.0  -0.757   0.4512    
```

Repeat, removing Elev.p95

```
Mreg <-lm(field$Carbon.AB ~ lidar$Elev.mean + lidar$Percentage.all.returns.above.6.56)
summary (Mreg)
```

```
Coefficients:
                                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)                              -75005.6    12764.2  -5.876 1.05e-07 ***
lidar$Elev.mean                            7419.9      796.8   9.312 3.38e-14 ***
lidar$Percentage.all.returns.above.6.56     824.8      250.7   3.290  0.00152 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 40180 on 76 degrees of freedom
Multiple R-squared:  0.7454,    Adjusted R-squared:  0.7387 
F-statistic: 111.2 on 2 and 76 DF,  p-value: < 2.2e-16
```

It is difficult to meaningfully plot the results of a multiple linear regression but combining Elev.mean and Percentage.all.returns.above.6.56 seem to have a stronger relationship to above ground carbon than either of them did separately.

There are more sophisticated statistical methods that should be applied to

QUESTION 13: Determine the two lidar metrics that model QMD.GT5 the best. Report the r-squared and p-value for your multiple linear regression. How do you think those two metrics ecologically relate to QMD?

PART 3: Extracting plot level Cloud Metrics

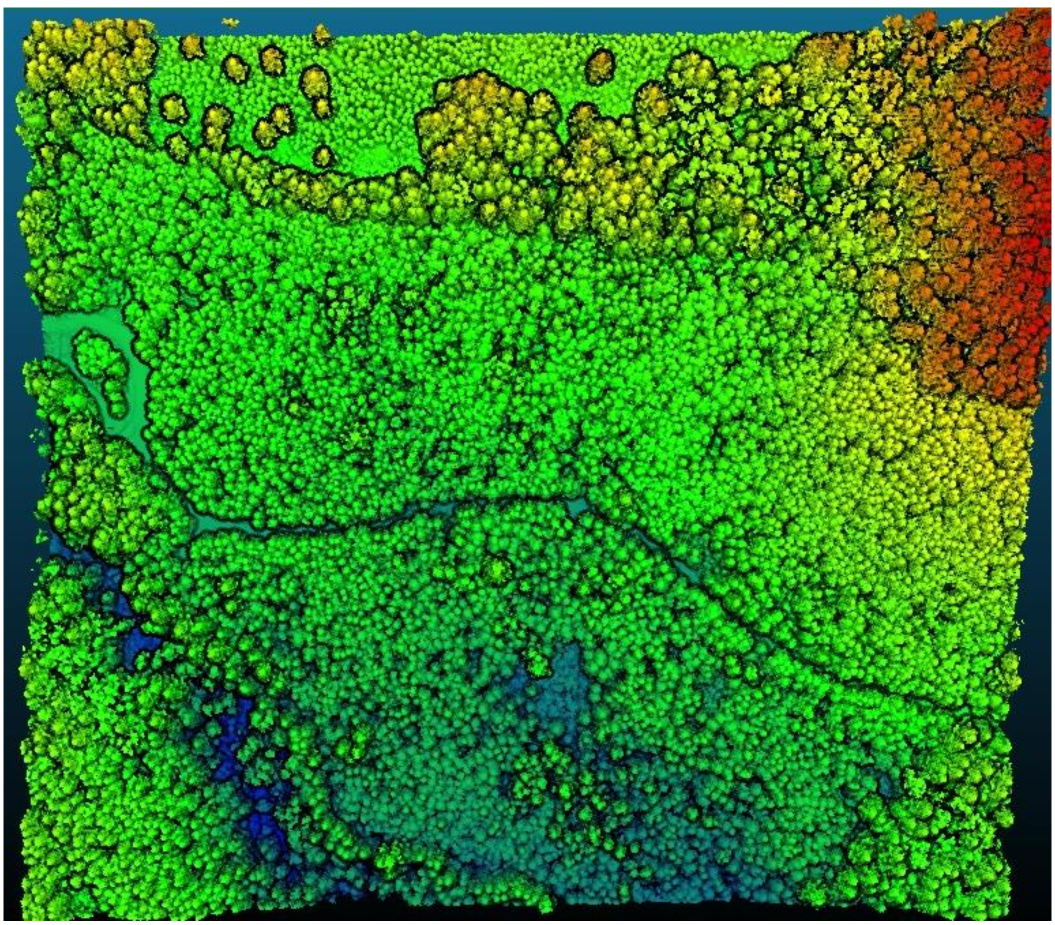
You are handed a small section of ALS data with the projection information removed to persevere the anonymity of the location. You are also given the locations of five plots within the ALS coverage. Your job is to clip out the point clouds at each of the plots, run cloud metrics on the plots, and look for relationships between lidar cloud metrics, and the field data collected at the plots. Ready? Go!

Clipping the data.

The first step is to import the CloudSection.las into R.

```
LASfile <- ("LAB5/CloudSection.las")
las <- readLAS(LASfile)
plot(las)
```

You can check out the point cloud in R, but also take a moment to load the point cloud into CloudCompare, you'll need to have it CloudCompare for a later question anyway.



ESRM433/SEFS533
Lab 5
Also check out the Plot2.csv in excel to see the field data, including the x and y coordinates for each plot.

Plot	X	Y	Radius	TPA	QMD	DBH	Height	HT40	LCR	SV632	BASAL	Volume
1	1636	-2636	60	44	13.3	12.97	77.8	79.6	62.1	3.43	42.7	1333
2	1430	-2230	60	103	12.1	11.79	76.6	85.1	50.4	6.21	81.9	2587
3	1216	-2425	60	181	10.8	10.65	77.2	82.1	46.3	7.7	116	3717
4	1279	-2725	60	243	10.5	10.36	76.9	84.2	42.2	9.84	147.4	4702
5	1139	-2174	60	160	10.4	10.17	76.7	86.8	46.2	6.31	94.2	3093

The field metrics are:

- TPA
  - Trees Per Acre
- QMD
  - Qudratic Mean Diameter
- DBH
  - Average diameter breast height
- Height
  - Average height of trees
- HT40
  - Average height of the 40 tallest trees
- LCR
  - Live Crown Ratio
- SV632
  - Scribner board foot volume to a 6-inch top diameter inside bark in 32-foot logs
- BASAL
  - The cross-sectional area of trees at breast height (1.3m or 4.5 ft above ground)
- Volume
  - Total estiamated volume of trees in plot

Using the X, Y, & radius values, create clips for each plot:

```
#create 5 circle clips and then normalize the clips
P1 <- clip_circle(las,1636,-2636,60)
P1n <- normalize_height(P1,tin())
plot(P1n)
P2 <- clip_circle(las,1430,-2230,60)
P2n <- normalize_height(P2,tin())
plot(P2n)
P3<- clip_circle(las,1216,-2425,60)
P3n <- normalize_height(P3,tin())
plot(P3n)
P4 <- clip_circle(las,1279,-2725,60)
P4n <- normalize_height(P4,tin())
plot(P4n)
P5 <- clip_circle(las,1139,-2174,60)
P5n <- normalize_height(P5,tin())
plot(P5n)
```

The x and y for the position data here is not any projection but only location relative to our CloudSelection point cloud. If you had UTM or lat/long for plots, all of these steps would be exactly them same. The only thing that matters is that your coordinate information for your plots match the coordinates for the las file you are clipping from. In this case, the data had a projected coordinate system and a large random value was simply subtracted from the x, y, and z values for all points. Subtract the same values from the plot location data and the relative locations stay the same.

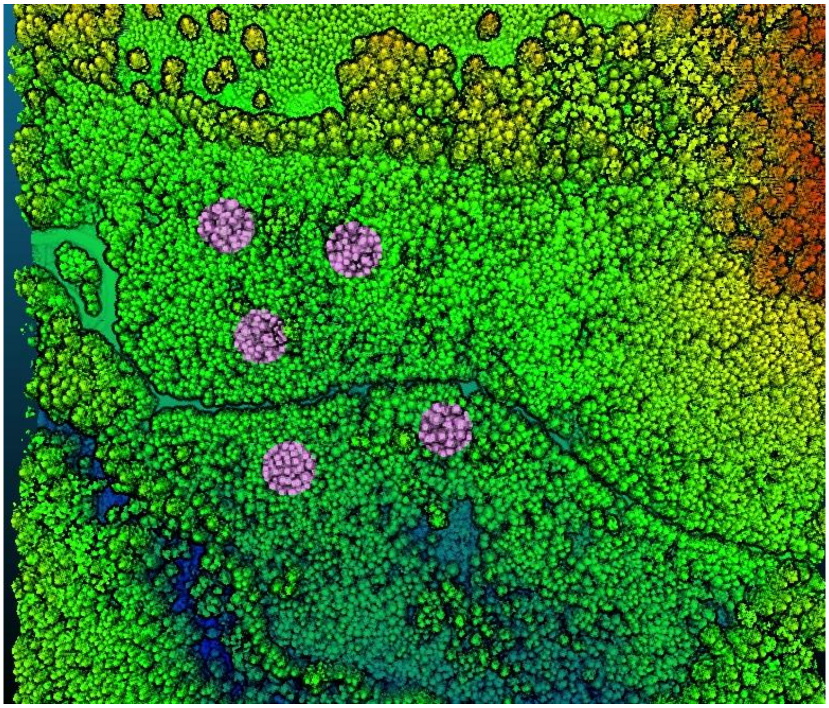
clip and normalize should both be familiar to you now but you can always use ?lclip_circle or ?normalize_height for more information.

Go ahead and create a new las file with your clip data

```
#Combines all the non-normalized clips
#and creates a single las file from the clips
PLOTSlas <-rbind(P1,P2,P3,P4,P5)
plot(PLOTSlas)
writeLAS(PLOTSlas, file = "LAB5/PLOTSlas.laz")
```

The rbind command joins the separate las sets together. Note that it is combining the non-normalized clips. You should have a new file in your LAB5 folder.

Take a moment and import your new PLOTSlas.laz file into CloudCompare and display it with your CloudSelection.las file. This will help you visualize where the plots are, and how the forest looks in a broader context. Make sure you change your colors. I would suggest a height ramp for the CloudSelection and a solid color for the PLOSTlas.



QUESTION 14: Include a screenshot of your Plotslas and CloudSelection clouds rendered in CloudCompare. Pick an interesting view and make sure to color the point clouds so the plots can be seen. Play around with point size and colors. Include a caption for your screenshot.

Now that we have our clips, we need to create the cloud metrics for each one, and we also want to only use the points above 2m (6.56ft).

```
#Creates standard cloud metrics from the normalized clips
#only uses the points above 2m (6.56ft)
?cloud_metrics
P1m <- cloud_metrics(filter_poi(P1n,Z>6.56), .stdmetrics)
P2m <- cloud_metrics(filter_poi(P2n,Z>6.56), .stdmetrics)
P3m <- cloud_metrics(filter_poi(P3n,Z>6.56), .stdmetrics)
P4m <- cloud_metrics(filter_poi(P4n,Z>6.56), .stdmetrics)
P5m <- cloud_metrics(filter_poi(P5n,Z>6.56), .stdmetrics)
```

We now need all those cloud metrics in a single data frame.

```
#combines all of the clip cloud metrics into a single data frame
#rbind is merging (bind) data by stacking rows (r)
CM <- rbind.data.frame(P1m,P2m,P3m,P4m,P5m)
```

Cropping the data as we did with lasfilter to only include points above 2m, we need to run a separate command to look at the non-filtered plot point clouds to figure out the canopy cover, we will also have to do this because the default in stdmetrics assumes that the data is in meters so it looks at Z values greater than 2, but we need z values greater than 6.56. This is also an example of how you can make custom metrics from a point cloud:

```
#creates a metric for canopy cover by counting the number of points
#above 2m (sum(Z>6.56)) and deviding by the number of all points (sum(Z>-1))
#number of points is >-1 as a few poitns may have a slightly negitive value
#due to the tin used to create the normalized point cloud.
P1c <- cloud_metrics(P1n, ~sum(Z>6.56)/sum(Z>-1))
P2c <- cloud_metrics(P2n, ~sum(Z>6.56)/sum(Z>-1))
P3c <- cloud_metrics(P3n, ~sum(Z>6.56)/sum(Z>-1))
P4c <- cloud_metrics(P4n, ~sum(Z>6.56)/sum(Z>-1))
P5c <- cloud_metrics(P5n, ~sum(Z>6.56)/sum(Z>-1))
```

Now combine our canopy cover metrics into one data frame.

```
#combineds our canopy clousure values into one data frame
CC <- rbind(P1c,P2c,P3c,P4c,P5c)
```

Use the following command to change the data name to CC.

```
colnames(CC) <- c("CC")
```

So we have created all of our cloud metrics that we want to use. We still need to import the field plot data:

```
#reads in the csv file of our plot data and creates a data frame
plot2 <- read.csv("LAB5/plot2.csv")
```

We have three different data frames for our data, we can combine them all into one data frame just for ease. This step isn't necessary, but it might make your life easier.

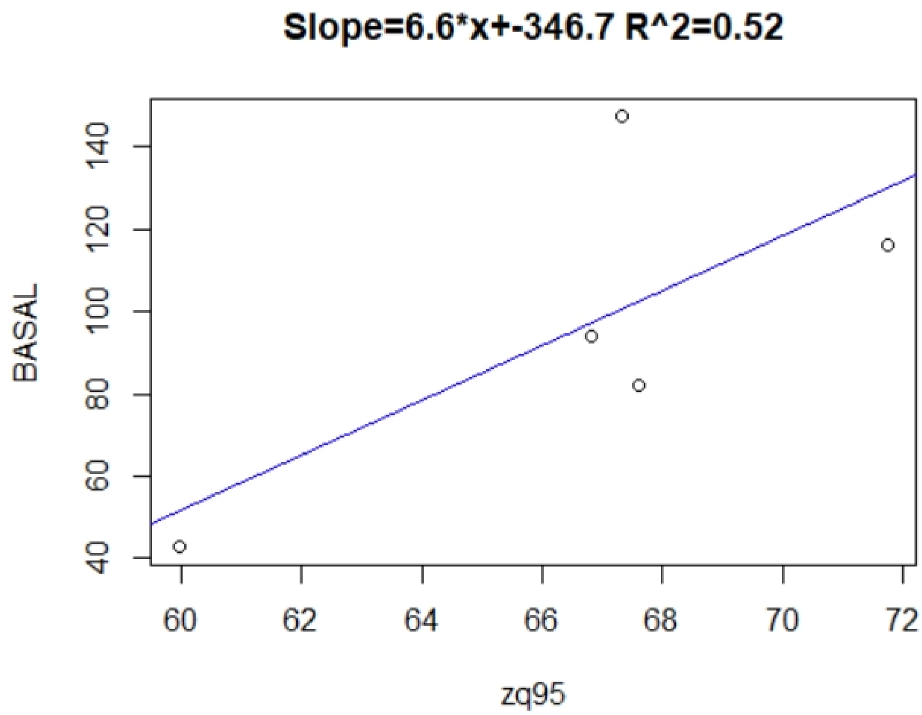
```
#combines our outputs togeter for one datafrom.
#bindes the columns instead of the rows
D <-cbind.data.frame(plot2,CC,CM)
```

The following code is just a repeat from the code above. Note that our cloud metrics have different names than before. The names above were generated using the lidar software 'Fusion', now we are using lidR. Fusion's p95 is the same as lidR's zq95.

```
Y <- D$BASAL #Dependent Variable. Basal area from our field data
X <- D$zq95 #Independent Variable. Height where 95% of all points are below

reg <- lm (Y ~ X) # reg is the regression output from Y & X
sum <- summary(reg) # sum is the summary statistics of the linear regression of Y & X
sum # prints the summary statistics. p-value and R-squared value are of the most interest to us

coeff <- coefficients(reg) #coefficients are the slope of the regression line
#eq will be the title of the plot with slope and R-Squared values
eq <- paste0("Slope=",round(coeff[2],1),"*x+",round(coeff[1],1)," R^2=",round(sum$r.squared, 2))
plot(X, Y, xlab="zq95", ylab="BASAL", main=eq) #plots the points and title
abline(reg, col="blue") #adds the regression line to the plot
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -346.673    244.653  -1.417    0.251
X              6.644      3.662   1.814    0.167

Residual standard error: 31.1 on 3 degrees of freedom
Multiple R-squared:  0.5231,	Adjusted R-squared:  0.3642
F-statistic: 3.291 on 1 and 3 DF,  p-value: 0.1673
```

Setting a threshold for significance at 0.05, there is not enough evidence to show a significant relationship between zq95 and Basal area for our plots. However, we have a very small sample size and when you look at the data plotted out, it looks like if we removed the one outlier, there would likely be a statistically significant relationship. Just need more data.



We have a ton of metrics now. You don't have to go through each and every single metric (unless you want to). Instead lets just focus on the cloud metrics:

- zmean
- zsd
- zq25
- zq95
- CC --- this is our created canopy cover metric. It is essentially the same as the percentage above 2m that you used before.

QUESTION 15: We are interested in TPA, report the R^2 and p-values for the five cloud metrics above in relation to TPA. For the cloud metric that has the best relationship with TPA, include a screenshot of the plot with axes labeled and a description. Make sure you discuss the ecological relationship between the cloud metric and the field data.

QUESTION 16: For question 13 you determined the two lidar metrics that modeled QMD.GT5 the best. Try those two metrics using this new data in a multiple linear regression to model QMD. Are they good predictors in our new study site? Why or why not?

QUESTION 17: What cloud metric(s) have the best relationship to QMD? Two things to keep in mind… A multiple linear regression may not produce a better relationship than a simple linear regression. Our sample size is very small so you can't just add all the metrics and start subtracting metrics like we did before. You'll just have to try different combinations. Make sure you provide an ecological answer as to why the metrics would be related.

QUESTION 18: In the example given, we used a simple linear regression to look for a relationship between Basal area and zq95. We had a p-value > 0.05. Do any of our five selected cloud metrics have a statistically significant relationship to Basal area? Does a multiple linear regression to include 2 independent variables improve the statistical relationship or, in this case, does a single independent variable provide the best result? Report the R^2 and p-value for the regression that preforms the best and discuss the ecological reasoning.

GRADUATE STUDENT

You cited several papers in question 1. Write a short paper summary on one of them. One or two paragraphs will suffice.

R Appendix

```
##################################
########### Lab 5 ###############
##################################

library(lidR)

setwd("//fs-persona.sefs.uw.edu/student_redirect$/jonbatch/Desktop/ESRM433/")

LASfile <- ("Lab5/Sequoia.las")
las <- readLAS(LASfile)
lasNORM <- normalize_height(las,tin())
writeLAS(lasNORM,file="Lab5/SequoiaNORM.las")

?cloud_metrics
cloud_metrics(lasNORM, ~max(Z))
CMSequoia <- cloud_metrics(lasNORM, .stdmetrics)
write.csv(CMSequoia,file = "Lab5/CMSequoia.csv")

over2 <- filter_poi(lasNORM, Z > 6.56)
CMSequoia2<-cloud_metrics(over2, .stdmetrics)
write.csv(CMSequoia2, file = "Lab5/CMSequoia2.csv")

field <- read.csv("Lab5/plotdata.csv")
lidar <- read.csv("Lab5/cloudmetrics.csv")

?lm
Y<-field$QMD.GT5
X<-lidar$Elev.P95

reg <- lm(Y~X)
sum <- summary(reg)
sum
coeff<- coefficients(reg)
eq<-paste0("Slope = ",round(coeff[2],1),"*x+",round(coeff[1],1)," R^2=" , round(sum$r.squared, 2))
plot(X,Y, xlab="P95", ylab="QMD", main=eq)
abline(reg, col= "blue")

Mreg <- lm(field$Carbon.AB ~ lidar$Elev.stddev + lidar$Elev.mean +
             lidar$Percentage.all.returns.above.6.56 + lidar$Elev.P95 + lidar$Elev.P25)
summary (Mreg)

Mreg <- lm(field$Carbon.AB ~  lidar$Elev.mean+ lidar$Percentage.all.returns.above.6.56
           + lidar$Elev.P95 + lidar$Elev.P25)
summary (Mreg)

Mreg <- lm(field$Carbon.AB ~  lidar$Elev.mean+ lidar$Percentage.all.returns.above.6.56
           + lidar$Elev.P95)
summary (Mreg)
#continue until all combos are run

```

```
LASfile<- ("Lab5/CloudSection.las")
las <- readLAS(LASfile)
plot(las)

P1 <- clip_circle(las, 1636,-2636, 60)
P1n <- normalize_height(P1,tin())
plot(P1n)
P2 <- clip_circle(las, 1430,-2230, 60)
P2n <- normalize_height(P2,tin())
plot(P2n)
P3 <- clip_circle(las, 1216,-2425, 60)
P3n <- normalize_height(P3,tin())
plot(P3n)
P4 <- clip_circle(las, 1279,-2725, 60)
P4n <- normalize_height(P4,tin())
plot(P4n)
P5 <- clip_circle(las, 1139,-2174, 60)
P5n <- normalize_height(P5,tin())
plot(P5n)

PLOTSlas <- rbind(P1,P2,P3, P4,P5)
plot(PLOTSlas)
writeLAS(PLOTSlas, file = "Lab5/PLOTSlas.las")

?cloud_metrics
P1m <- cloud_metrics(filter_poi(P1n,Z>6.56), .stdmetrics)
P2m <- cloud_metrics(filter_poi(P2n,Z>6.56), .stdmetrics)
P3m <- cloud_metrics(filter_poi(P3n,Z>6.56), .stdmetrics)
P4m <- cloud_metrics(filter_poi(P4n,Z>6.56), .stdmetrics)
P5m <- cloud_metrics(filter_poi(P5n,Z>6.56), .stdmetrics)
CM<-rbind.data.frame(P1m,P2m,P3m,P4m,P5m)

P1c <- cloud_metrics(P1n, ~sum(Z>6.56)/sum(Z>-1))
P2c <- cloud_metrics(P2n, ~sum(Z>6.56)/sum(Z>-1))
P3c <- cloud_metrics(P3n, ~sum(Z>6.56)/sum(Z>-1))
P4c <- cloud_metrics(P4n, ~sum(Z>6.56)/sum(Z>-1))
P5c <- cloud_metrics(P5n, ~sum(Z>6.56)/sum(Z>-1))
CC<-rbind.data.frame(P1c,P2c,P3c,P4c,P5c)
colnames(CC) <- c("CC")

plot2<-read.csv("Lab5/plot2.csv")

D <- cbind.data.frame(plot2,CC,CM)

Y<-D$BASAL
X<-D$zq95

reg <- lm(Y~X)
sum <- summary(reg)
sum
coeff<- coefficients(reg)
```

```
eq<-paste0("Slope = ",round(coeff[2],1),"*x+",round(coeff[1],1)," R^2=" , round(sum$r.squared, 2))
plot(X,Y, xlab="zq95", ylab="BASAL", main=eq)
abline(reg, col= "blue")

reg <- lm(D$TPA ~ D$zmean)
summary(reg)
reg <- lm(D$TPA ~ D$sd)
summary(reg)
reg <- lm(D$TPA ~ D$zq25)
summary(reg)
reg <- lm(D$TPA ~ D$zq95)
summary(reg)
reg <- lm(D$TPA ~ D$CC)
summary(reg)

Q16<-lm(D$QWMD ~D$ + D$)
```