Lab 5: Statistical Assesment of Lidar with on-the-ground Plot Data ESRM433 2023

TA: Anthony Stewart

2023-04-24

Welcome to 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:

- The Lab 5 Submission Quiz (https://canvas.uw.edu/courses/1633883/quizzes/1855505)

Introduction

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 (https://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."

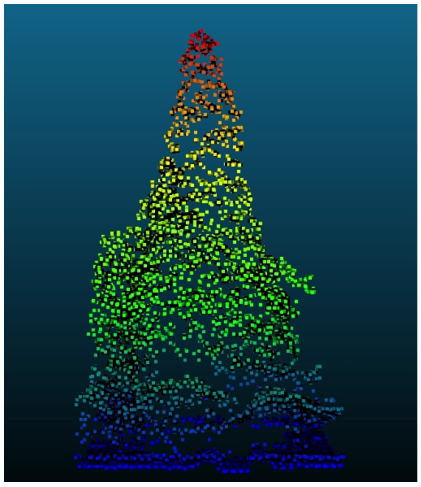
Another good resource is this introduction to statistics in R book here: (https://learningstatisticswithr.com /book/descriptives.html#interquartile-range (https://learningstatisticswithr.com /book/descriptives.html#interquartile-range))

Cloud Metrics

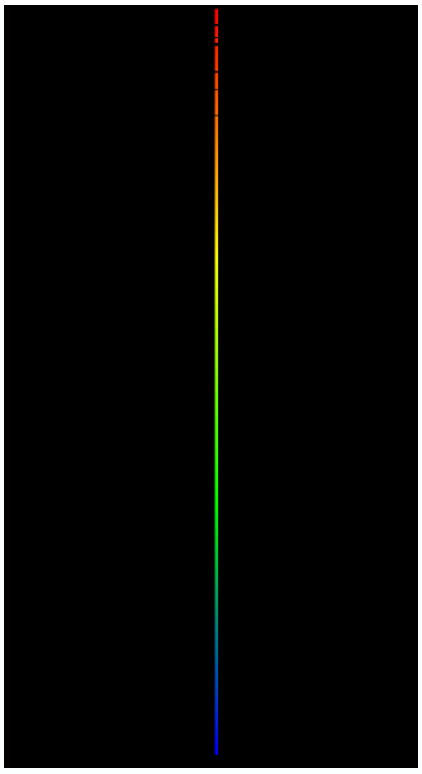
There are other methods to derive cloud metrics and we will cover those later in this lab. For now, we will be focusing on elevation.

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.



This is a Giant Sequoia Point Cloud



This would be the range of the point cloud height values if we stacked them all together in a single line

PART 1: Looking at Cloud Metrics

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.

You should be able to set up the working directory to a folder either on the desktop or U Drive.

So let's open up R Studio and import our `lidR` library and set our working directory:

```
# setwd("/Users/Anthony/OneDrive - UW/University of Washington/Teaching/SEFS433_Lidar
/Labs/")
# I've already done this in the document so I've commented it out

library(lidR)
library(rgl)
library(sf)
# you may need to do lib.loc to specify the directory if you get errors:
#library(lidR, lib.loc = "A DIRECTORY IF YOU GET AN ERROR")
```

You should have downloaded a `"LAB5Data"` folder from Canvas. In the data folder there is a relatively small las file: `Sequoia.las`

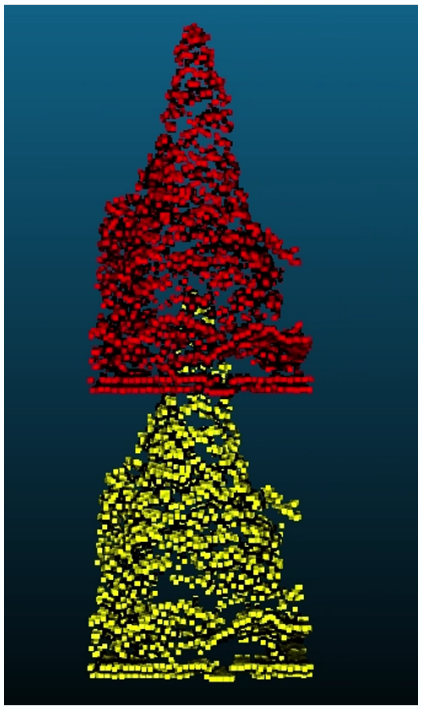
```
## [1] "LAB5Data/Sequoia.las"
```

Now, create a normalized las file from `Sequoia.las` and save it into your LAB5 folder:

```
las <- readLAS("Lab 5/LAB5Data/Sequoia.las")
lasNORM <- normalize_height(las, tin())
writeLAS(lasNORM, "Lab 5/LAB5Data/SequoiaNORM.las")
```

Remember: Normalizing height subtracts a DTM from the point to make the ground at 0 and the other heights relative to the ground

Now open up Cloud Compare to get a better visualization of the regular and normalized point clouds



I've set my colors for the trees 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.

Now back to R

Take a moment to familiarize yourself with `lidR::cloud_metrics` function. We're going to be using these quite a bit in the lab

```
?cloud_metrics
```

Since we are mostly concerned with Z for height values we can check that out within our `lasNORM` file

You can use `$` after a variable to subset it and see if there's more attributes or descriptor variables inside

```
lasNORM$Z[1:10] #This prints the first 10 Z values
```

```
##  [1]  3.00  3.52  3.63  4.42  4.14 10.74  0.00 12.25  0.00 14.33
```

Ok, so we know we have Z values for height in our point cloud let's do something with this.

You can do simple task with `cloud_metrics` and find out what the maximum Z value is, in your normalized point cloud:

```
cloud_metrics(lasNORM, ~max(Z)) # we are using feet in this case
```

```
## [1] 129.31
```

QUESTION 1: You are able to create your own metrics using `cloud_metrics`, but there are other existing functions that `lidR` can use, what are three of them?

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` for Z

To generate all `stdmetrics` for our las file, we need to run `cloud_metrics` with our `lasNORM` normalized point cloud file and specify that we want `.stdmetrics`

- I've gone further to make this a dataframe with the `as.data.frame` function to make the original list easier to work with

```
CMSequoia <- as.data.frame(cloud_metrics(lasNORM, .stdmetrics))
```

CMSequoia Over 2m

zmax	zmean	zsd	zskew	zkurt	zentropy	pzabovezmean	pzabove2	zq5	zq10	zq15	zq
129.3	37.1	31	0.6	2.7	NA	48.2	82.6	0	0.1	0.4	

Ok, there are a lot of metrics from this `cloud_metrics` function using `.stdmetrics`. Let's look at the help page:

```
?.stdmetrics
```

Description from the .stdmetrics

Predefined metrics functions intended to me used in `*_metric`s function such as pixel_metrics, cloud_metrics, crown_metrics, voxel_metrics and so on. Each function comes with a convenient shortcuts for lazy coding. The lidR package aims to provide an easy way to compute user-defined metrics rather than to provide them. However, for efficiency and to save time, sets of standard metrics have been predefined (see details). Every function can be computed by every `*_metrics` functions however * `stdmetrics` are more pixel-based metrics, `stdtreemetrics` are more tree-based metrics and `stdshapemetrics` are more point-based metrics. For example the metric `zmean` computed by `stdmetrics_z` makes sense when computed at the pixel level but brings no information at the voxel level.

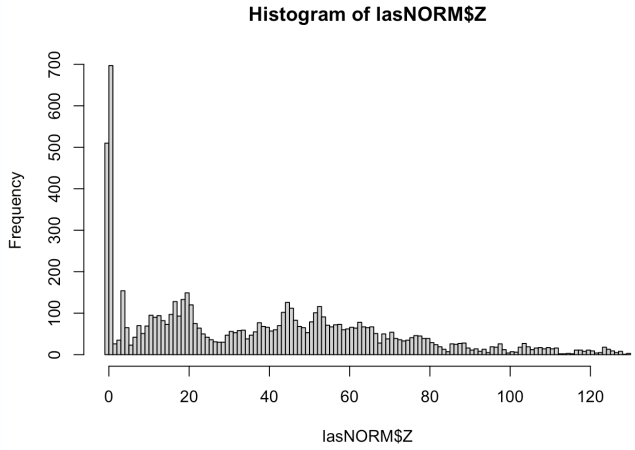
So we get some predefined metrics for point clouds which makes things easier.

QUESTION 2: In the ?stdmetrics help page there are details that lists the nomenclature used for the names of metrics. Provide descriptions for those names, then find a description for what `zpcumx` is.

Before we dive in a bit further, let's look back at our `lasNORM` point cloud and attempt to visualize some of the raw Z metrics using histograms

- histograms use percentiles defined earlier to show a distribution of data. Each percentile contains a count of the number of data points. For example, we can see how many observations cluster together in certain percentiles.

```
hist(lasNORM$Z, breaks = 100)
```



This is showing the frequency or count of the Z(height) data within the point cloud

QUESTION 3: Notice the huge spike in the values around Z=0. Why are we getting so many returns at this height (Z)?

In this lab, we're not interested in the ground *for now…* so let's filter those points out and focus on the trees. 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.

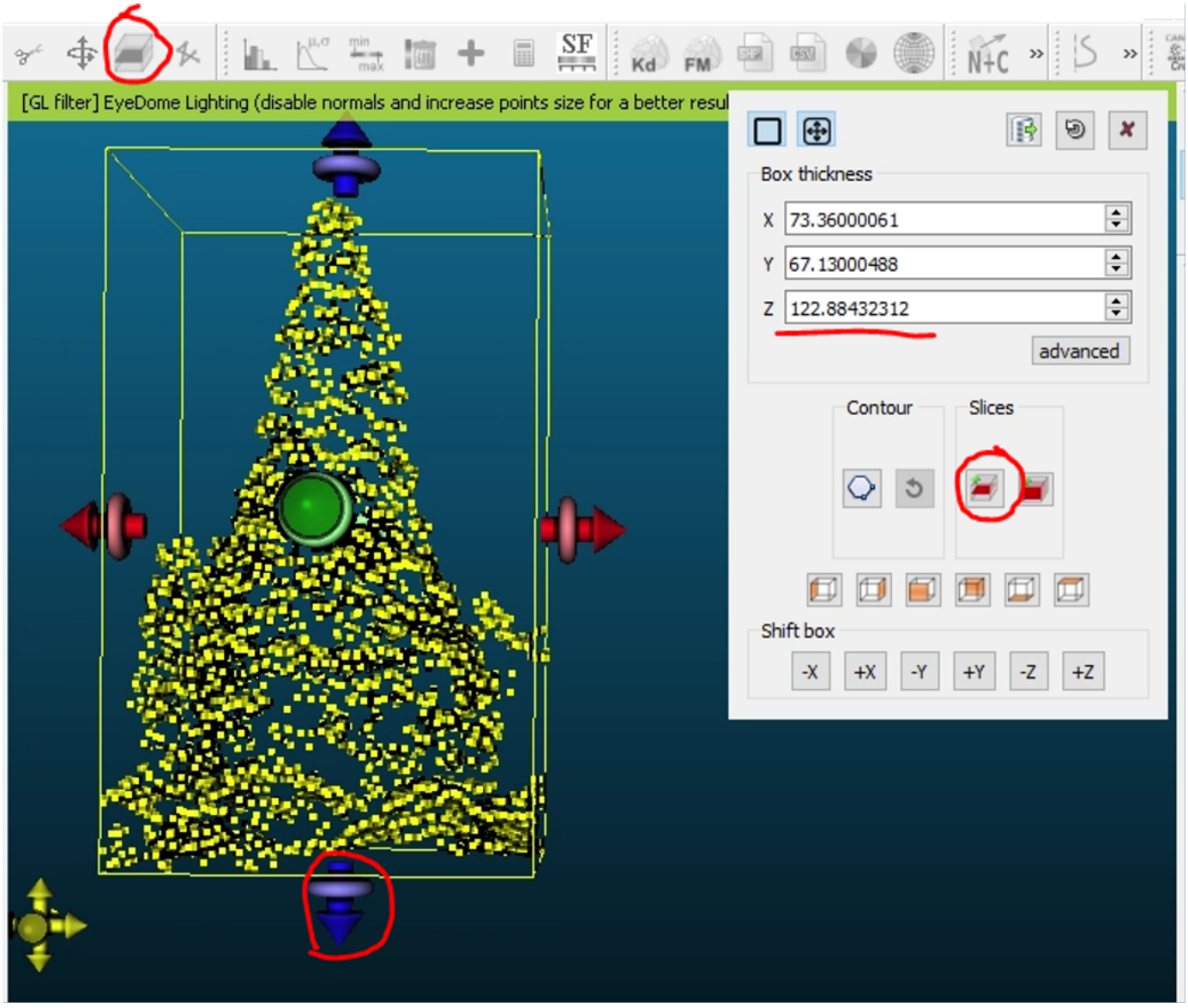
```
over2m <- filter_poi(lasNORM, Z >= 6.56)
CMSequoia_over2m <- as.data.frame(cloud_metrics(over2m, .stdmetrics))
```

CMSequoia Over 2m

zmax	zmean	zsd	zskew	zkurt	zentropy	pzabovezmean	pzabove2	zq5	zq10	zq15	z
129.3	47.1	27.6	0.6	2.9	0.9	46.7	100	10.7	13.8	16.8	

You can also do this in Cloud Compare so we're going to bring in our `lasNORM` point cloud (SequoiaNORM.las) into Cloud Compare and do a quick visualization exercise.

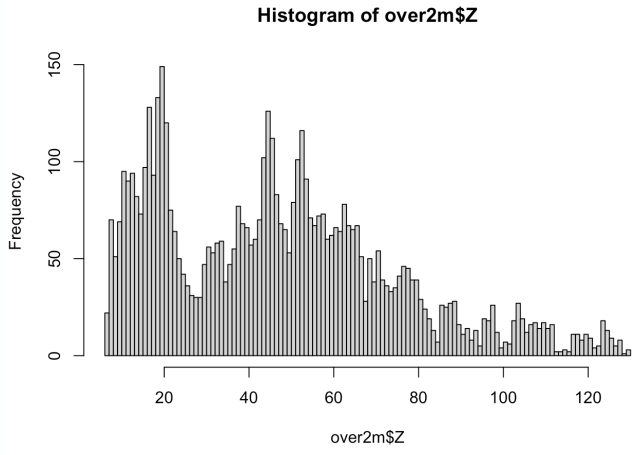
In CloudCompare, use the cross section tool to manually clip out the approximate bottom 2 m. Slide up the bottom blue arrow until the Z value is as close to 122.92 (129.48 – 6.56) as you are able.



QUESTION 4: Take a screenshot of your filtered point cloud in Cloud Compare like the figure above

Let's look at another histogram this time with the `over2m` data that we created by filtering out points less than 2m (6.56 ft)

```
hist(over2m$Z, breaks = 100)
```



QUESTION 5: Include a screenshot of your histogram and mark on it the approximate location of the 95th, 50th, and 10th percentile. Hint: there's an R function in the stats package that can do this

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 25-40 height values?

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 CMSequoia_over2m.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:

> We 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.

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

PlotNumber	Tree_No	TreeSpeciesCD	TreeDBH (in)	TreeHeight (ft)	Uncompacted Crown Ratio (%)	Compacted Live Crown Ratio (%)
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

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

Trees Per Acre or Hectare (TPA or TPH)

- This is an important metric for forestry but it is also one of the easiest to measure. We can use a plot area and simply count the trees within it. Then divide by the area of the plot in acres or hectares

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:

$$QMD = \sqrt{\frac{\sum D_i^2}{n}}$$

- where D_i is the diameter at breast height of the ith 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.

Uncompacted crown ratio:

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

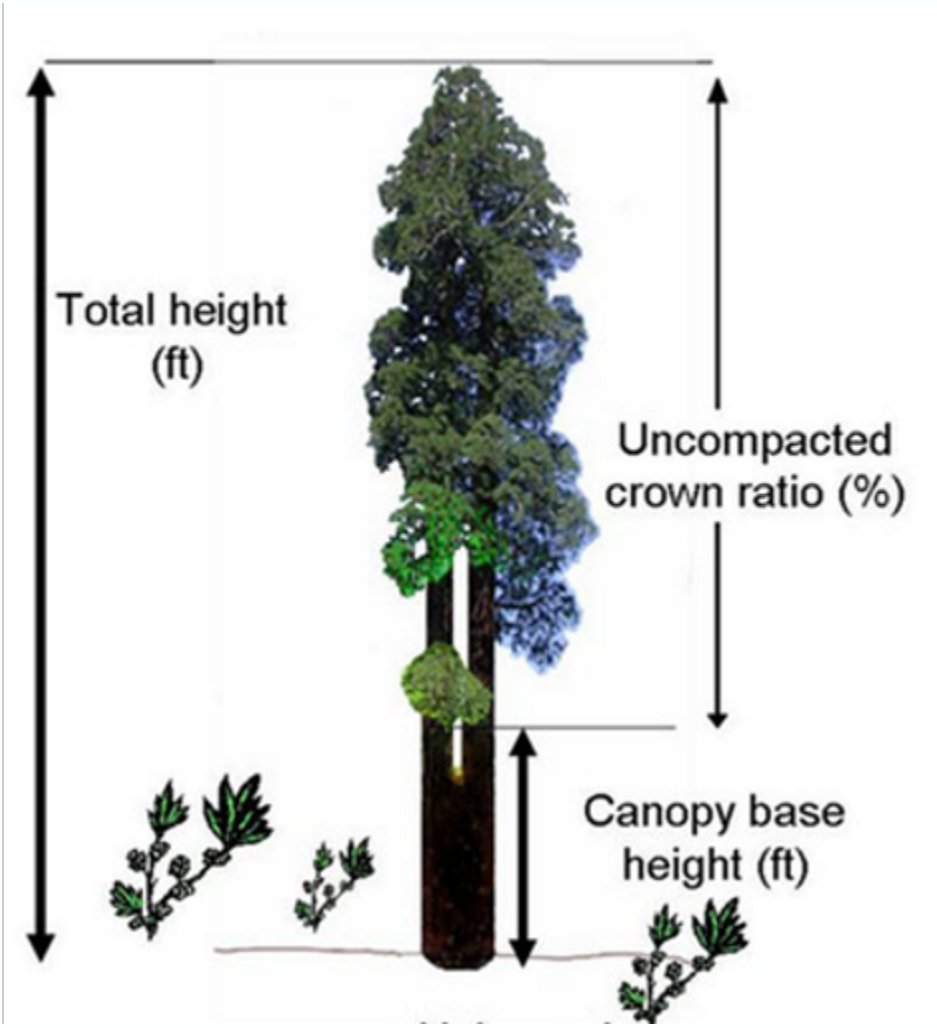
Compacted live crown ratio:

- 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.

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.



A diagram of what these metrics represent for a tree

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

Let's look back at **Plot 201**

- QMD, using the equation provided above, is 16.8in per plot or 42.6cm per plot.
- For $BasalArea$ and TPH we have to upscale our plot size from 1/10ha to a full acre 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 $BasalArea$, we first figure out the area of the trees at DBH within the plot. Remember that the area of a circle is A=πr^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 $\frac{(12.27ft^2*10)}{2.47}$ 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 Ti is the height of the ith tree and Ri is either the uncompacted crown ratio or the compacted live crown ratio of the ith 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 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)
- 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 listed above 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: 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!

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

```
las_proj <- readLAS("Lab 5/LAB5Data/CloudSection.las")
#plot(las_proj)
```

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.

Also check out the Plot2.csv in R or excel to see the field data, including the x and y coordinates for each plot.

Plot and Field Data

Plot	X	Y	Radius	TPA	QMD	DBH	Height	HT40	LCR	SV632	BASAL	Volume
1	1636	-2636	60	44	13.3	13.0	77.8	79.6	62.1	3.4	42.7	1333
2	1430	-2230	60	103	12.1	11.8	76.6	85.1	50.4	6.2	81.9	2587
3	1216	-2425	60	181	10.8	10.7	77.2	82.1	46.3	7.7	116.0	3717
4	1279	-2725	60	243	10.5	10.4	76.9	84.2	42.2	9.8	147.4	4702
5	1139	-2174	60	160	10.4	10.2	76.7	86.8	46.2	6.3	94.2	3093

The field metrics are:

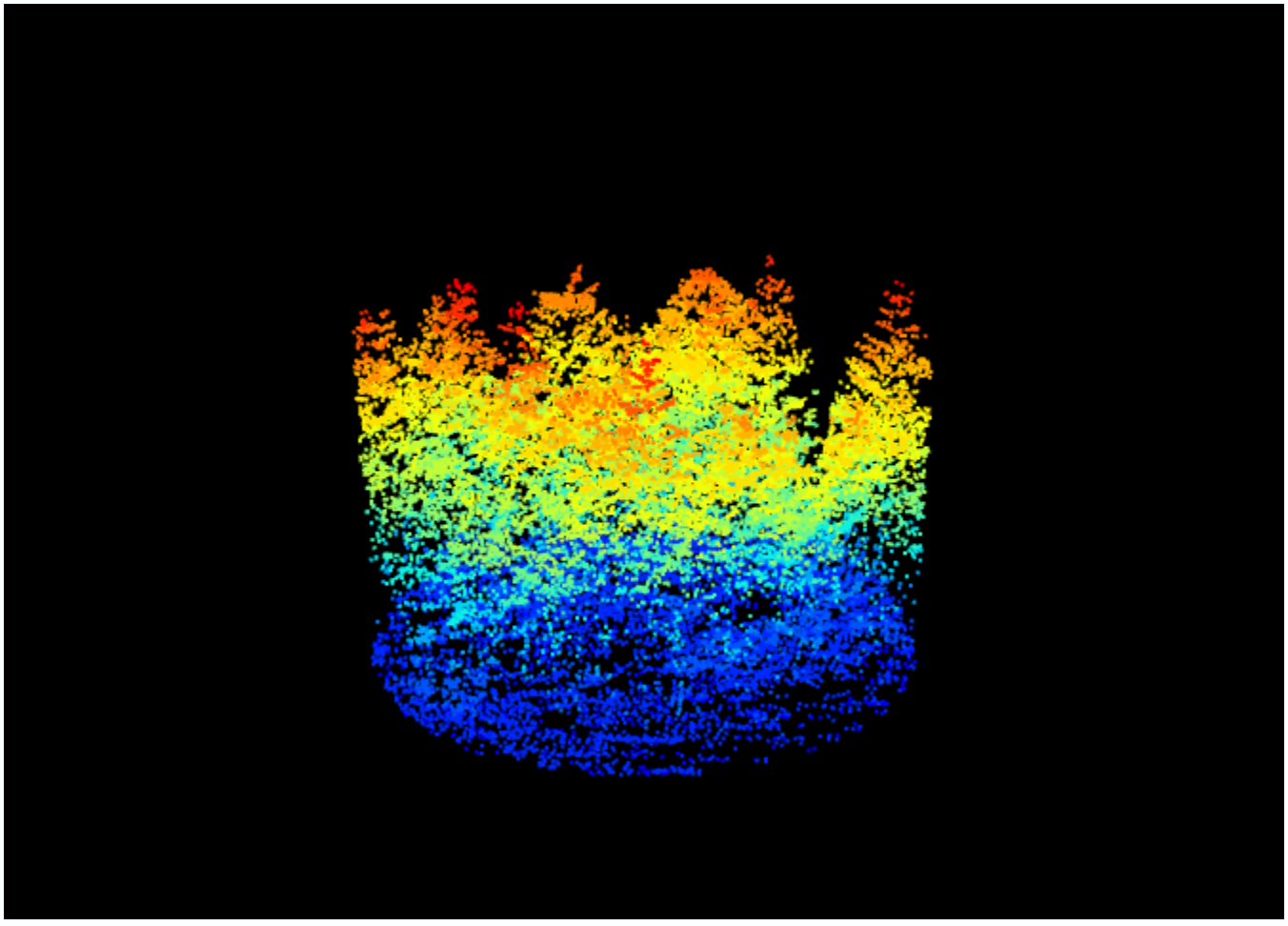
- TPA: Trees Per Acre
- QMD: Qudratic Mean Diameter
- DBH: Average diameter breast height
- Height: Average height of trees
- HT40: Average height of the 40 tallest trees
- LCR: Live Crown Ratio
- SV632: Scribner board foot volume to a 6-inch top diameter inside bark in 32-foot logs
- BASAL: The cross-sectional 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:

```
P1 <- clip_circle(las_proj, 1636, -2636, 60)
P1n <- normalize_height(P1, tin())

P2 <- clip_circle(las_proj, 1430, -2230, 60)
P2n <- normalize_height(P2, tin())

P3 <- clip_circle(las_proj, 1216, -2425, 60)
P3n <- normalize_height(P3, tin())

P4 <- clip_circle(las_proj, 1279, -2725, 60)
P4n <- normalize_height(P4, tin())

P5 <- clip_circle(las_proj, 1139, -2174, 60)
P5n <- normalize_height(P5, tin())
```

Let's check out a plot of a plot

```
plot(P5n)
```



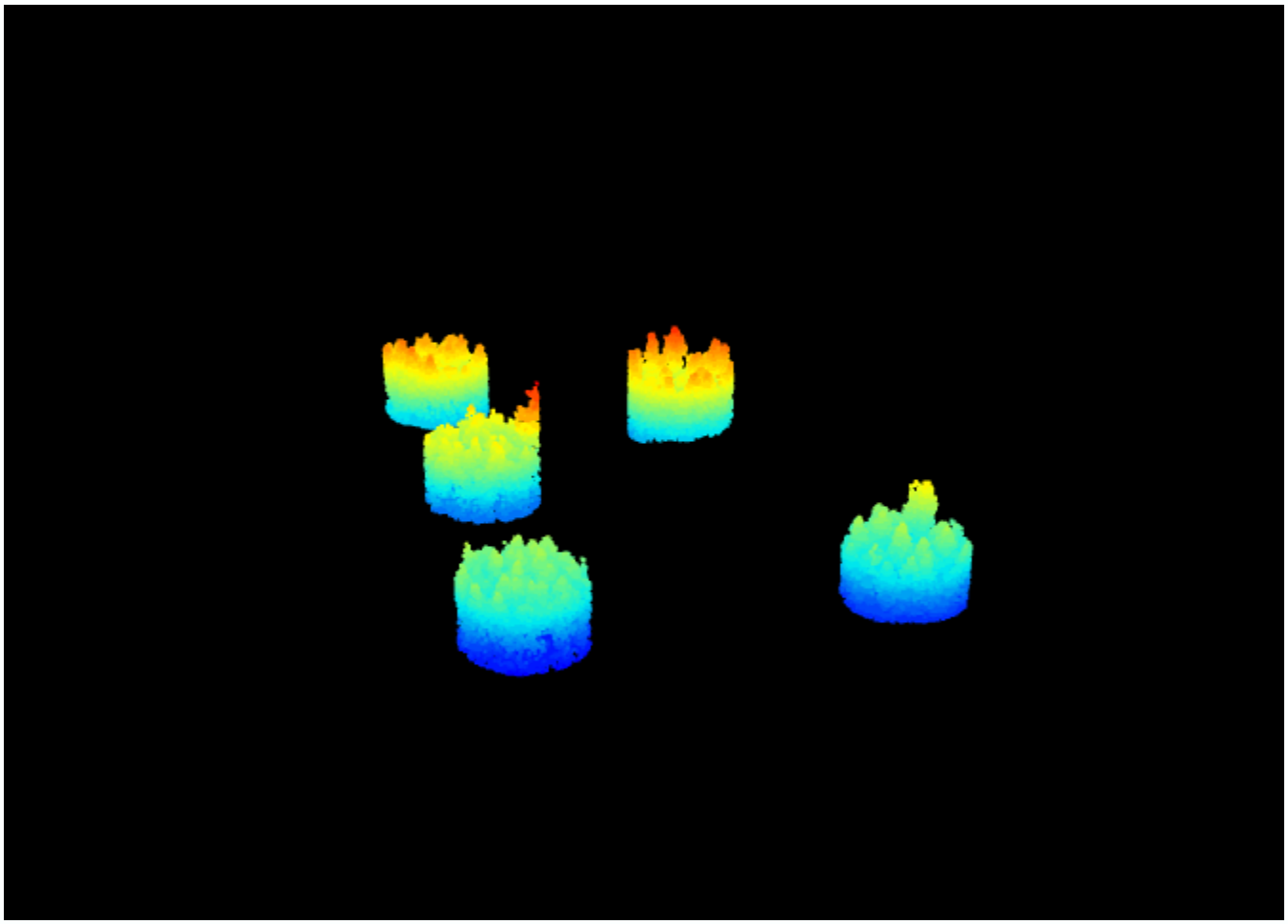
The x and y for the position data here is not any projection but only location relative to our CloudSelection point cloud. If you had UTM or lat/long for plots, all of these steps would be exactly them same. The only thing that matters is that **your coordinate information for your plots match the coordinates 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.

`Lasclip` and `lasnormalize` should both be familiar to you now but you can always use `?lasclip` or `?lasnormalize` for more information

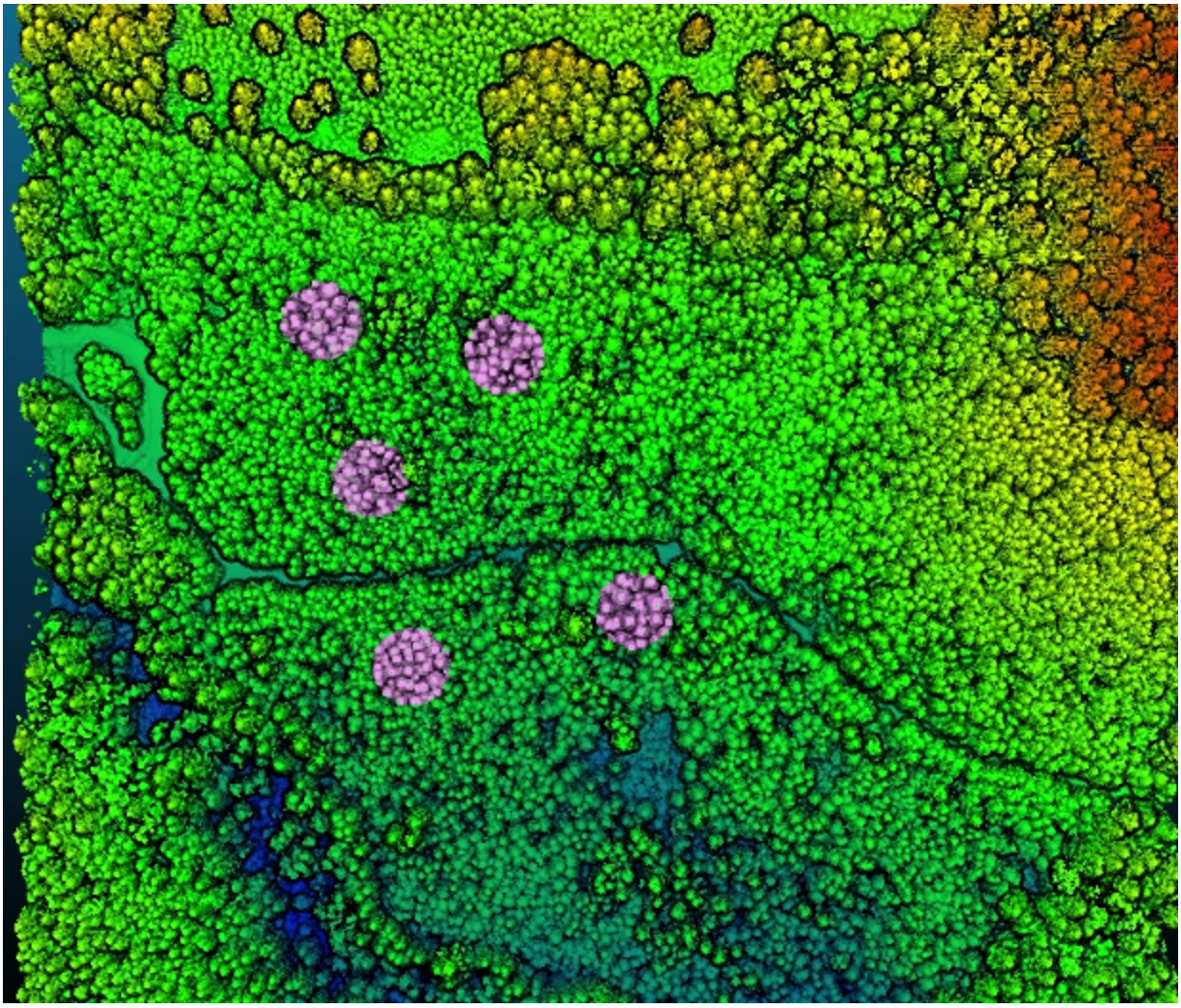
Go ahead and create a new las file with your clip data

```
lasPLOTS <- rbind(P1, P2, P3, P4, P5)
plot(lasPLOTS)
writeLAS(lasPLOTS, file = "Lab 5/lasPLOTS.laz") #writing out the file to save
```



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 `lasPLOTS.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 lasPLOTS.



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).

```
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)
```

```
P_CM <- rbind.data.frame(P1m, P2m, P3m, P4m, P5m)
```

Cropping the data as we did with `filter_poi` 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 (6.56 f
t)  and dividing them by the number of all points (sum(Z>-1))
# The number of points uses the Z> -1 to include any slightly negative points in the
point cloud 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))

P_CC <- rbind(P1c, P2c, P3c, P4c, P5c)
```

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

```
plot2 <- read.csv("Lab 5/LAB5Data/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.

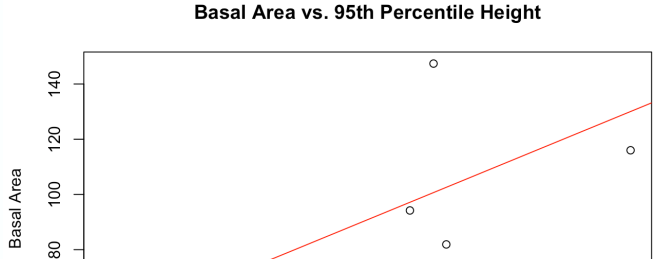
```
df <- cbind.data.frame(plot2, P_CC, P_CM)
```

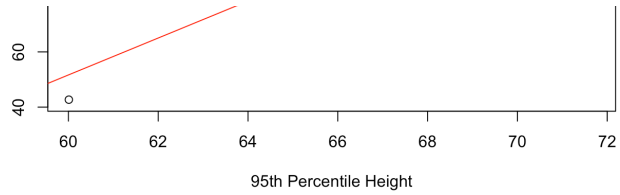
Combined Dataframe for Plot Data, Cloud Metrics, and Canopy Metrics

	Plot	X	Y	Radius	TPA	QMD	DBH	Height	HT40	LCR	SV632	BASAL	Vo
P1c	1	1636	-2636	60	44	13.3	13.0	77.8	79.6	62.1	3.4	42.7	1
P2c	2	1430	-2230	60	103	12.1	11.8	76.6	85.1	50.4	6.2	81.9	2
P3c	3	1216	-2425	60	181	10.8	10.7	77.2	82.1	46.3	7.7	116.0	3
P4c	4	1279	-2725	60	243	10.5	10.4	76.9	84.2	42.2	9.8	147.4	4
P5c	5	1139	-2174	60	160	10.4	10.2	76.7	86.8	46.2	6.3	94.2	3

From this dataframe we can start examining some relationships between lidar data and plot collected data

```
plot(df$zq95, df$BASAL, main = "Basal Area vs. 95th Percentile Height",
     xlab = "95th Percentile Height", ylab = "Basal Area")
abline(lm(df$BASAL ~df$zq95), col = "red")
```





In summary, the above plot shows our extracted lidar data which is the 95th percentile of Z/height returns vs. the Basal Area which is measured in the plot.

This is key!

Measuring with both Lidar and Plot measurements allows us to assess how well we are capturing tree and forest characteristics. Once we have enough of these plot data we could potentially scale up plot metrics across an entire lidar acquisition.

QUESTION 11: From your dataframe, choose 3 lidar vs. plot data relationships that might make ecological sense and plot them like above in the Basal Area vs. 95th Percentile. Try to plot the regression lines to show the slope. Take screenshots of all three of these plots and provide ecological reasoning as for why you chose them

Wait, how do we know whether or not this is a good relationship between plot and lidar data? Well that question brings us to the next section…

Interlude: Short Introduction to Linear Regression

You will be using linear regression to determine a relationship between LiDAR metrics and field metrics. If you are not familiar with linear models, this is a very brief background but read this book section here on regression (https://learningstatisticswithr.com/book/regression.html):

- Basic linear models try to fit a relationship between 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. The common form of a basic linear model is:

$$y = mx + b$$

or in statistical notation

$$\hat{Y}_i = \beta_1 X_i + \beta_0$$

- Where $\hat{Y}_i$ (“Y hat” due to the “^”)is our prediction of data based on the relationship from our predictor variable X_i time the slope β_1 plus the intercept β_0. The $_i$ denotes an individual number like the when

$i = 3, X_3$ is our 3rd X observation.

We then evaluate the **fit** between our predicted data $\hat{Y}_i$ and our observed data Y_i *notice no hat* which is usually a collected sample dataset. For example, today we will examine how well our lidar collected data explain some field collected data.

> Example) How well does the **95th percentile of Z** explain the **Quadratic Mean Diameter (QMD)**?

In this example our predictor variable X_i is our **lidar 95th percentile of Z** and our Y_i is the sampled **QMD** in the field data set. The intercept β_0 is estimated where the **lidar 95th percentile of Z** is 0

There are several common methods used to evaluate the quality of linear models. Today you will look at r^2 values and p-values.

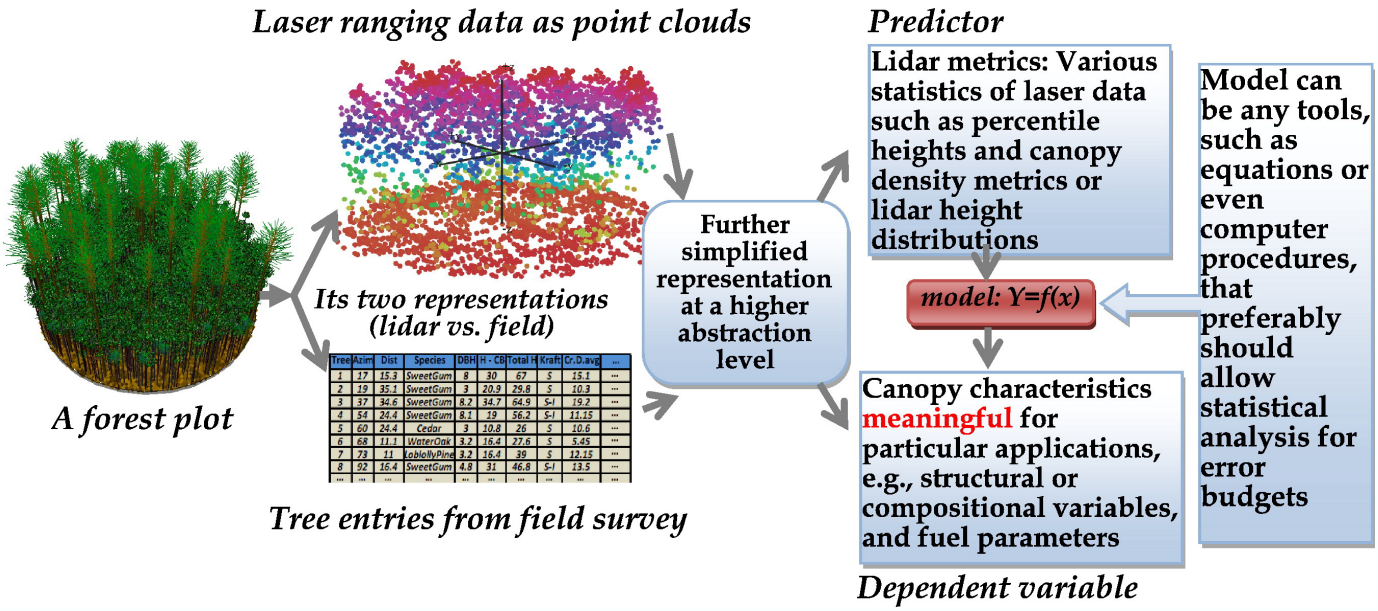
- An r^2 value gives an indication of how much of the variance in your dependent variable is explained by the independent variables. It can range from 0 to 1, with 0 being no relationship and 1 being a perfect fit There is no specific guideline as to what is an "acceptable" r^2 value. It is relative 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.
- There was a section here on p-values but p-values have a long history in science and they are often times arbitrary. I would encourage you to take a more in-depth statistics class that explains this better. But overall p-values are not the end all be all for explaining a relationship. But we will use them to explain some things in linear regression here where a "low" p-value indicates significance This is a good book section here that explains P-values, Hypotheses, Type 1 & 2 Error, and significance (https://learningstatisticswithr.com/book/hypothesistesting.html)

Reminder this is **not** a statistics course and the above explanations are just to get us to a starting point in the lab. Statistics is mostly magic in that it requires guided expertise and time spent in large books to master it

> Like other occult techniques of divination, the statistical method has a private jargon deliberately contrived to obscure its methods from non-practitioners.

> – Attributed to G. O. Ashley

Part 3: Implementing linear regression in R



Zhao, Kaiguang, Sorin Popescu, Xuelian Meng, Yong Pang, and Muge Agca. “Characterizing Forest Canopy Structure with Lidar Composite Metrics and Machine Learning.” Remote Sensing of Environment 115, no. 8 (August 15, 2011): 1978–96. https://doi.org/10.1016/j.rse.2011.04.001 (https://doi.org/10.1016/j.rse.2011.04.001).

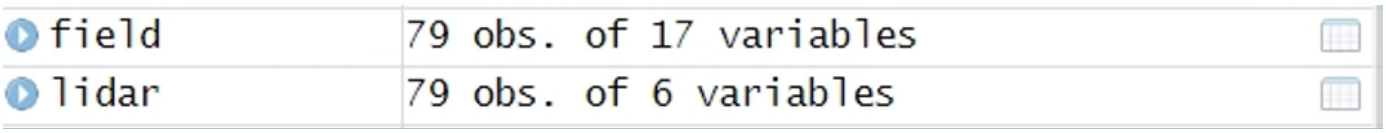
In the LAB5Data folder, there are two csv files, cloudmetrics.csv & plotdata.csv. These are plot data for 79 plots located on the east side of the cascade mountains. Plot data is from field measurements while cloud metrics are taken from the ALS of the plot locations.

We want to know how our **lidar** data corresponds and predicts our **plot** metrics

Ok let’s get back to coding by bringing in the data

```
plotdata <- read.csv("Lab 5/LAB5Data/plotdata.csv")
lidardata <- read.csv("Lab 5/LAB5Data/cloudmetrics.csv")
```

You can then click on the data set in the Environment tab usually on the upper right of the R Studio screen to get a better look at the columns and values contained.



{width = 50%}

Let’s check out the plot data

```
head(plotdata)
```

Plot Data

PlotID	TPA	BA	SDI	QMD.GT5	QMD.25pct	QMD.50pct	QMD.75
01bd3f24-4bc9-4ff6-a0f1-be759c05173e	40.1	63.0	93.9	17.0	19.0	19.8	21.3

PlotID	TPA	BA	SDI	QMD.GT5	QMD.25pct	QMD.50pct	QMD.75
04530f04-7687-4a37-91f2-f3b619e4a0c4	120.2	74.9	103.7	17.9	20.4	24.4	33.6
0a6d8d0e-5ae7-4193-b76a-b3c48d09fb76	70.1	21.3	40.8	10.1	11.4	11.4	14.0
1510bf7d-bed3-4268-8594-0effd115f6d2	190.4	243.0	349.8	15.3	18.7	21.7	27.4
15d7be42-1cd6-4a9a-8502-d0a65e3c6e2e	50.1	35.0	58.8	11.3	12.3	13.7	15.0
16c1e180-c3a7-439c-af7a-2bfb2faabc53	120.2	163.8	249.3	15.8	17.6	18.9	20.3

Ok lot's of data here. Let's simplify and focus on:

- `PlotID` : unique names for plots
- `TPA` , trees per acre
- `BA` , basal area
- `QMD.GT5` , quadratic mean diameter of trees greater than 5 inches in diameter

Let's check out the lidar data

```
head(lidardata)
```

Lidar Data

FileTitle	Percentage.all.returns.above.6.56	Elev.mean	Elev.P25	Elev.P95
01bd3f24-4bc9-4ff6-a0f1-be759c05173e	24.2	16.9	13.6	25.6
04530f04-7687-4a37-91f2-f3b619e4a0c4	66.8	15.3	6.7	35.3
0a6d8d0e-5ae7-4193-b76a-b3c48d09fb76	22.3	6.7	3.7	13.7
1510bf7d-bed3-4268-8594-0effd115f6d2	71.4	24.5	19.4	37.3
15d7be42-1cd6-4a9a-8502-d0a65e3c6e2e	31.3	13.8	9.4	22.0

FileTitle	Percentage.all.returns.above.6.56	Elev.mean	Elev.P25	Elev.P95
16c1e180-c3a7-439c-af7a-2bfb2faabc53	69.0	19.3	15.7	28.1

For the lidar data, the file

- `FileTitle` : 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.
- `Elev` metrics describe the `Z` values of all points within the plot.

Let's find out through a linear model if Plot Basal Area or `BA.GT5` can be predicted by Lidar `Percentage.all.returns.above.6.56`

Linear models in R can be built with the `lm` function

Check out the `lm` function by using `?lm` in R and identify the data you want to use as your dependent and independent variables.

```
?lm
Y <- plotdata$BA.GT5
X <- lidardata$Percentage.all.returns.above.6.56
lmod <- lm(Y ~ X)
lmod
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Coefficients:
## (Intercept)            X
##     -40.126        2.968
```

We now have this linear model object `lmod` and if you highlight and run it it gives you the above formula with Coefficients: Intercept and X. These are the β_0 and β_1 from the Interlude

Notably, our β_1 for X is positive but our β_0 the intercept is negative. We'll get to that intercept later on…

Using the `summary` function will evaluate the model

```
summary(lmod)
```

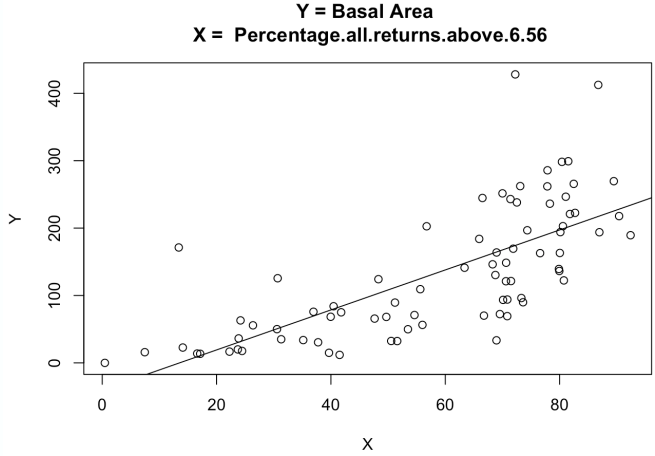
```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.067  -46.522   -8.894   32.624  253.765
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -40.1255    20.8630  -1.923   0.0581 .
## X             2.9680     0.3326   8.922 1.69e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68.23 on 77 degrees of freedom
## Multiple R-squared:  0.5083, Adjusted R-squared:  0.5019
## F-statistic: 79.61 on 1 and 77 DF,  p-value: 1.692e-13
```

The summary shows a lot of information. But the main things to point out are:

- X or our `Percentage.all.returns.above.6.56` is a noted as a significant predictor based on a t statistic and Pr(>|t|) which is basically a p-value
- There are two forms of R^2 and they both explain about 50% of the variation in the model which is pretty good.
- Our model has an overall significant p-value of 1.692e-13 which means we have a relationship!

Let's plot our variables together and see if it makes sense. We should have a positive slope based on the positive β_1 for X

```
plot(X, Y, main = "Y = Basal Area \n X =  Percentage.all.returns.above.6.56")
abline(lm(Y ~ X)) # a is the intercept and b is the slope
```



This is a fairly good relationship! But the R^2 is still about 50%. Let's add all the variables and see if we can explain a lot of the variance

```
lmod_all <- lm(plotdata$BA ~  lidardata$Percentage.all.returns.above.6.56 +
                              lidardata$Elev.mean +
                              lidardata$Elev.P25 +
                              lidardata$Elev.P95 +
                              lidardata$Elev.stddev)

summary(lmod_all)
```

```
##
## Call:
## lm(formula = plotdata$BA ~ lidardata$Percentage.all.returns.above.6.56 +
##     lidardata$Elev.mean + lidardata$Elev.P25 + lidardata$Elev.P95 +
##     lidardata$Elev.stddev)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.705 -31.581  -9.519  20.014 231.062
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                 -59.6084    19.8547  -3.002
## lidardata$Percentage.all.returns.above.6.56   1.5866     0.3521   4.507
## lidardata$Elev.mean                          22.2651    12.4899   1.783
## lidardata$Elev.P25                          -10.4622    10.1319  -1.033
## lidardata$Elev.P95                           -3.0577     4.6965  -0.651
## lidardata$Elev.stddev                        -7.7159    15.7299  -0.491
##                                             Pr(>|t|)
## (Intercept)                                  0.00367 **
## lidardata$Percentage.all.returns.above.6.56 2.46e-05 ***
## lidardata$Elev.mean                          0.07880 .
## lidardata$Elev.P25                           0.30520
## lidardata$Elev.P95                           0.51705
## lidardata$Elev.stddev                        0.62523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.36 on 73 degrees of freedom
## Multiple R-squared:  0.7014, Adjusted R-squared:  0.6809
## F-statistic: 34.29 on 5 and 73 DF,  p-value: < 2.2e-16
```

Well now we increased our R^2 but we have some redundancy in our model. This is indicated by the lack of significance for some of our parameters, particularly: `Elev.P25` , `Elev.P95` , and `Elev.stddev`

So let's remove these and keep just the `Elev.mean` and `Percentage.all.returns.above.6.56`

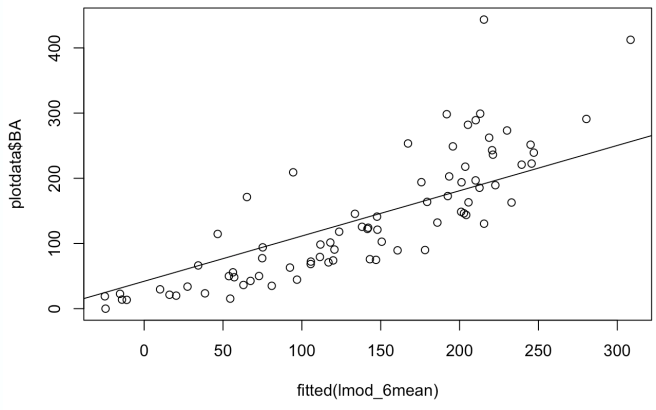
```
lmod_6mean <- lm(plotdata$BA ~  lidardata$Percentage.all.returns.above.6.56 +
                              lidardata$Elev.mean)

summary(lmod_6mean)
```

```
##
## Call:
## lm(formula = plotdata$BA ~ lidardata$Percentage.all.returns.above.6.56 +
##     lidardata$Elev.mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.172 -32.718  -8.936  20.975 228.016
##
## Coefficients:
##                                            Estimate Std. Error t value
## (Intercept)                                -66.7765    17.1199  -3.901
## lidardata$Percentage.all.returns.above.6.56   1.5584     0.3363   4.634
## lidardata$Elev.mean                           7.1791     1.0687   6.717
##                                            Pr(>|t|)
## (Intercept)                                0.000206 ***
## lidardata$Percentage.all.returns.above.6.56 1.46e-05 ***
## lidardata$Elev.mean                        2.98e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.89 on 76 degrees of freedom
## Multiple R-squared:  0.6945, Adjusted R-squared:  0.6865
## F-statistic:  86.4 on 2 and 76 DF,  p-value: < 2.2e-16
```

Now our R^2 is still high and we have significant predictors: `Elev.mean` and `Percentage.all.returns.above.6.56`

Let's plot this relationship

```
plot(fitted(lmod_6mean), plotdata$BA, )
abline(lm(fitted(lmod_6mean) ~ plotdata$BA))
```



This is sort of hard to visualize the multiple predictors in the model but they basically sum together to fit the regression line. Notice that it's a bit different from the first single variable linear model we did.

So for you to try this out, I want you to evaluate lidar prediction variables for `Carbon.AB` which is Carbon in

Aboveground Biomass.

QUESTION 12: First build a linear model to evaluate `Carbon.AB` as the dependent variable (Y) with the `lm` function and one variable of your choice as the predictor/independent variable (X) . Take a screenshot of the model `summary()` and plot the results of the model

QUESTION 13: Why did you choose the variable in Question 11? What about this lidar metric makes it suitable for predicting Carbon.AB? In other words, what is the hypothesis?

QUESTION 14: Next build a multiple linear regression for `Carbon.AB` with all the lidar variables. Then remove insignificant variables until you are left with only significant ones. Print the model `summary()` and ake a screenshot

BONUS 1: plot the results of the linear model in QUESTION 13 with the regression line.

BONUS 2: You might have noticed that our models often have a negative intercept where the regression line goes through the Y-axis. What does this mean when we evaluate a model that uses lidar metrics as the predictor?

GRADUATE STUDENTS: Find one scientific paper that uses lidar metrics and plot data similar to this lab. Provide a 2 sentence summary and a citation