ESRM433/SEFS533
Lab 3

Objectives:
- Create and assess LiDAR ground models
- Generating output for further analysis in ArcGIS Pro
- Projections and Coordinate Systems/

Data and Software:
- Lab3_Data zip folder, downloadable from Canvas
- Washington State Lidar Portal http://lidarportal.dnr.wa.gov/
  - Download data and store on Madrona Desktop at **ESRM433\LAB2**
  - Network directory name will be:
    - //fs-persona.sefs.uw.edu/student_redirect$/*YourNetID*/Desktop/ESRM433/LAB2
- RStudio. You can use via Madrona or download on your own computer here: https://rstudio.com/products/rstudio/download/
  - RStudio requires an installation of R as well: https://cran.rstudio.com/
  - A basic tutorial about using RStudio
    - https://education.rstudio.com/learn/beginner/
- lidR R package for lidar data manipulation https://github.com/Jean-Romain/lidR
  - lidR tutorials to supplement this lab: https://github.com/Jean-Romain/lidR/wiki
  - lidR reference book https://r-lidar.github.io/lidRbook/index.html
- ArcGIS Pro. You can use it on Madrona or download on your own computer.
  - **https://sites.uw.edu/arcgis/software/arcgis-desktop/arcgis-pro/**
- CloudCompare software. You can use it on Madrona or download on your own computer from:
  - https://www.danielgm.net/cc/
  - The current non-stereo stable version is the best to get for this class, but any version will work

What you will turn in:
- Submission Template in PDF format submitted via Canvas.

Welcome to Lab 3 for ESRM433/SEFS533!
This week in lab you will be learning how to create and assess LiDAR ground models, which are 3D representations of the Earth's surface. Ground models are one of the primary reasons that LiDAR data are acquired. Ground models developed from LiDAR data have incredibly high resolution and accuracy over very large areas. Even professional field surveys typically do not provide the quality of data that can be achieved using airborne LiDAR.

A ground model can also be called a Digital Terrain Model (DTM). A DTM is a type of Digital Elevation Model (DEM) that is the actual ground with any vegetation or buildings removed. The other common type of DEM is a Digital Surface Model (DSM) which is a DEM but keeps buildings and vegetation in it. More on DSMs next lab. High-quality ground models have numerous uses. Hydrological flow modeling, landslide risk/slope stability assessment, archaeological site detection, fluvial and slope geomorphology, and shoreline delineation/wasting assessment are some of the major applications. The availability of LiDAR ground models has allowed for substantial improvements in every one of these situations.

Creating ground models basically falls into two steps. First, all non-ground LiDAR returns have to be filtered out. Second, the returns representing the ground surface have to be converted from a collection of points to a 3D surface. The first step, filtering, represents most of the work, since it can be quite difficult to distinguish low, dense shrubs from the actual ground surface.

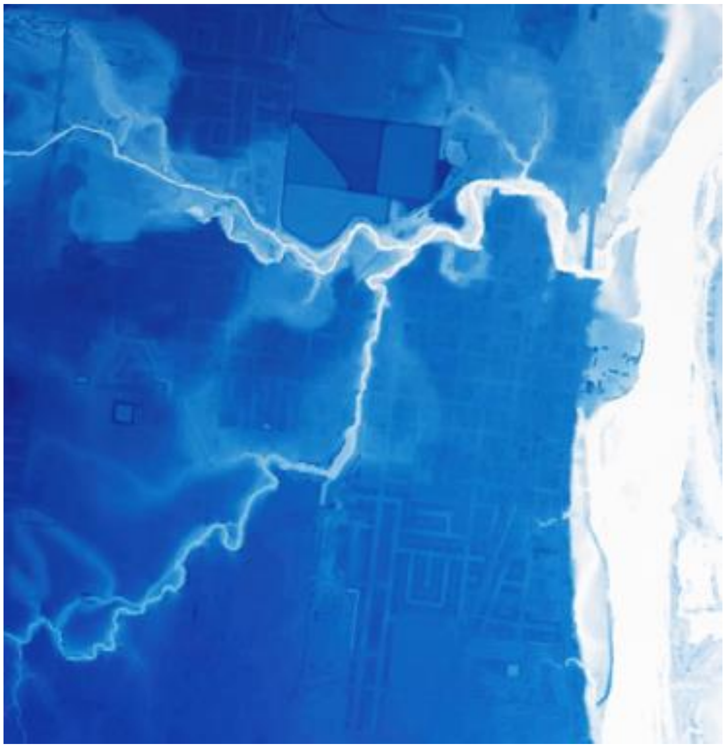
Note about working directories

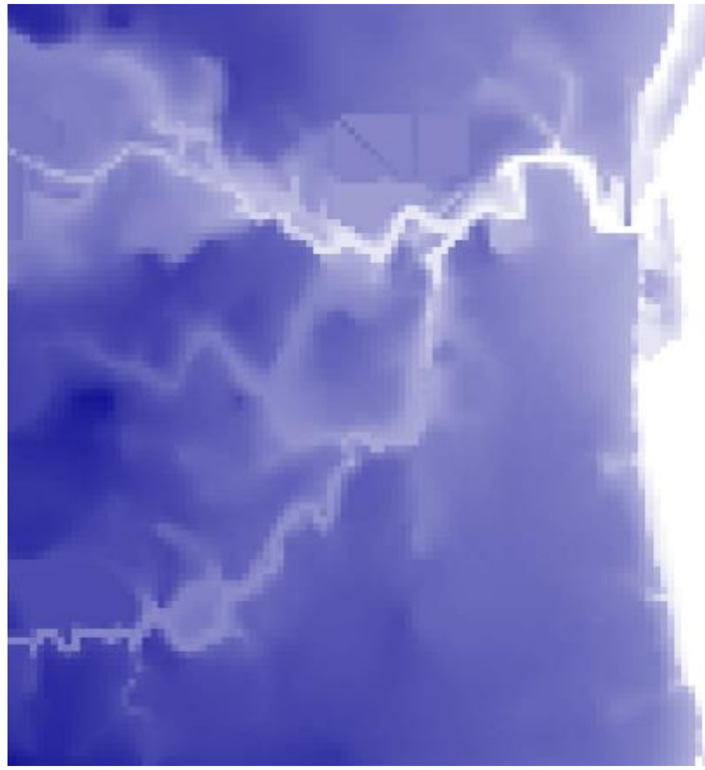
Remember that anything saved onto the C drive in your Madrona instance will be deleted once you log out. You have 20GB of storage in your networked folders (Downloads, Documents, Desktop). Normally I would not recommend keeping data and working from folders located on your Desktop, but in this case, I do suggest making a ESRM430 folder directly on your Desktop and creating subfolders within it for the different labs. The location of this folder in Madrona should be: //fs-persona.sefs.uw.edu/student_redirect$/*YourNetID*/Desktop/ESRM433
HOWEVER, you may find that RStudio, ArcGIS, and CloudCompare are especially slow if pulling data over a network. You can try to put your data directly on the C drive of your instance to work on during the lab, but if you do this, make sure you back up your work in your folder on your desktop before logging out, and I also recommend backing it up externally some place like your google drive. If you are moving data around, you may mess up file path names in places like RStudio so pay close attention to what data is where.

PART 1 Comparing LiDAR DEMs to 3DEP DEMs:

The general standard for digital elevation models (DEMs) across the US is 3DEP, the 3-Dimensional Elevation Program, which until recently was referred to as NED, the National Elevation Dataset. 3DEP DEMs are distributed by the US Geological Survey (USGS) at 10 m and 30 m resolutions across the whole US.

In this section you will make some general comparisons between the standard 3DEP DEM and a LiDAR DEM. Open WillametteRiver_1m_DOGAMI.pdf and WillametteRiver_10m_USGS.png so that you can view them side by side. These maps are renderings of the central Willamette Valley in Oregon, between the towns of Monmouth and Albany. The 1 m DEM is a LiDAR-derived ground model distributed by the Oregon Department of Geology and Mineral Industries (DOGAMI). DOGAMI is a great resource for publically-available LiDAR data in Oregon. The 10 m DEM is distributed by USGS and was derived by digitizing a scanned contour map (the original contours were created using photogrammetry from aerial stereo-photo pairs).





At the bottom of the DOGAMI map is some explanatory text, which reads:

> *This lidar-derived digital elevation model of the Willamette River displays a 50-foot elevation range, from low elevations (displayed in white) fading to higher elevations (displayed in dark blue). This visually replaces the relatively flat landscape of the valley floor with vivid historical channels, showing the dynamic movements the river has made in recent millennia.*

Look over these two maps and answer the following questions. You may also look at aerial photos on Google Maps if you desire:

https://www.google.com/maps/@44.7849986,-123.2504315,41998m/data=!3m1!1e3

Open the WillametteRiver_1m_DOGAMI.pdf and WillametteRiver_10m_USGS.png in the LAB3 data download. Don't just compare the small snips included above.

QUESTION 1: Describe the differences in level of detail between the two maps. Be specific, what kinds of features does the 1 m DEM capture that the 10 m DEM does not? Name and describe at least 3.

QUESTION 2: What consequences could these differences have in terms of answering research questions? For each difference you listed in question 1, try to name some situations in which it would better to have a finer-scale DEM and some situations where coarser-scale DEM would be fine.

ESRM433/SEFS533
Lab 3
PART 2: Vendor Supplied Ground Models

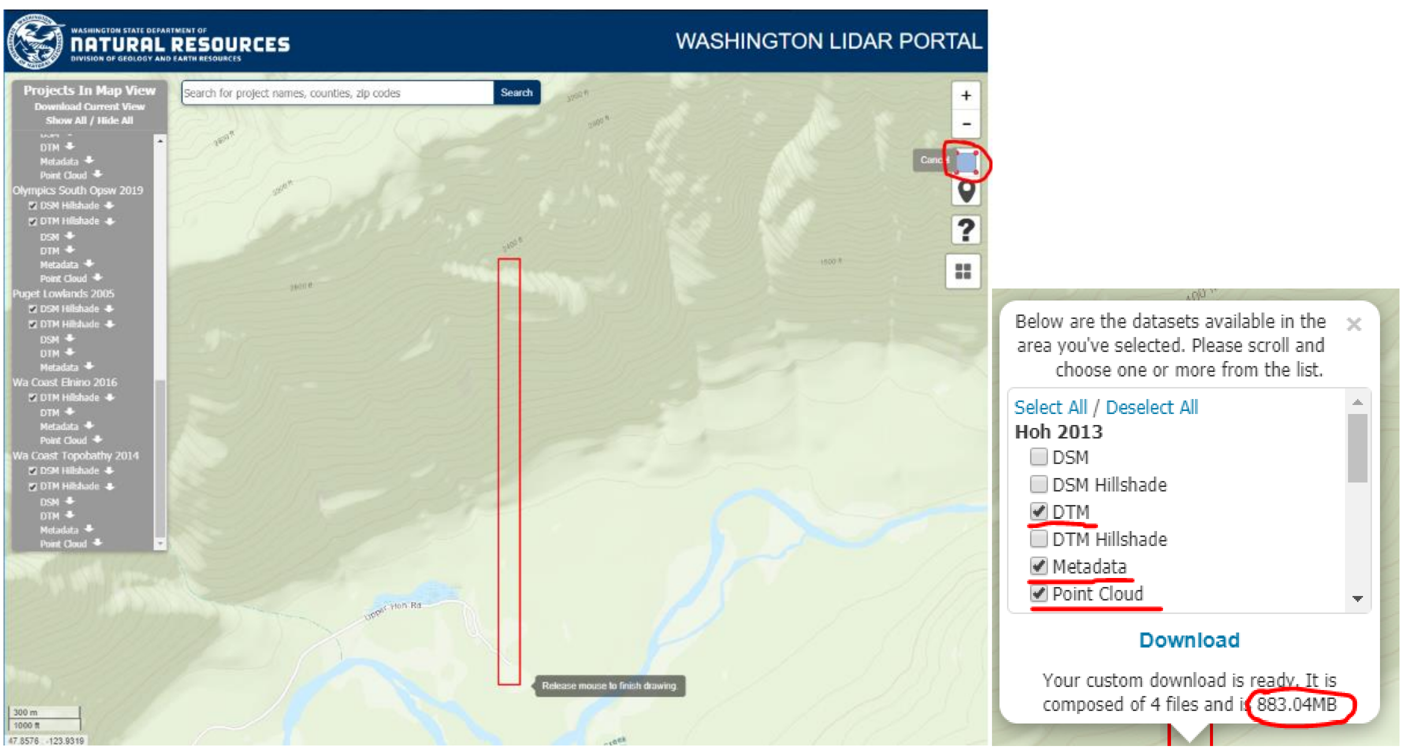
It is common practice for the vendors that acquire lidar data to provide a finished ground model of the acquisition as part of the deliverables. This may save you the time and effort of creating a ground model yourself, but you should still understand the process of creating one. For this step, we are going to look at a vendor created DTM for the East side of Olympic National Park. The first step is to download the data from the Washington DNR lidar portal.
Go to: https://lidarportal.dnr.wa.gov/#47.86450:-123.93295:15

Remember that the 47.86450:-123.93295:15 portion of the web address is the latitude and longitude of the window and the zoom level. This happens to be the Hoh Rain Forest Visitor Center. A beautiful area that you should check out at some point in the future, if you haven't been already.

Visiting the Hoh Rain Forest

Photo by Ken and Mary Campbell

The area and data that we want is the DTM for the acquisition, as well as two tiles of point cloud data. Use the polygon tool to select the area shown. This identifies that you want any data that is present within the bounds of the polygon.



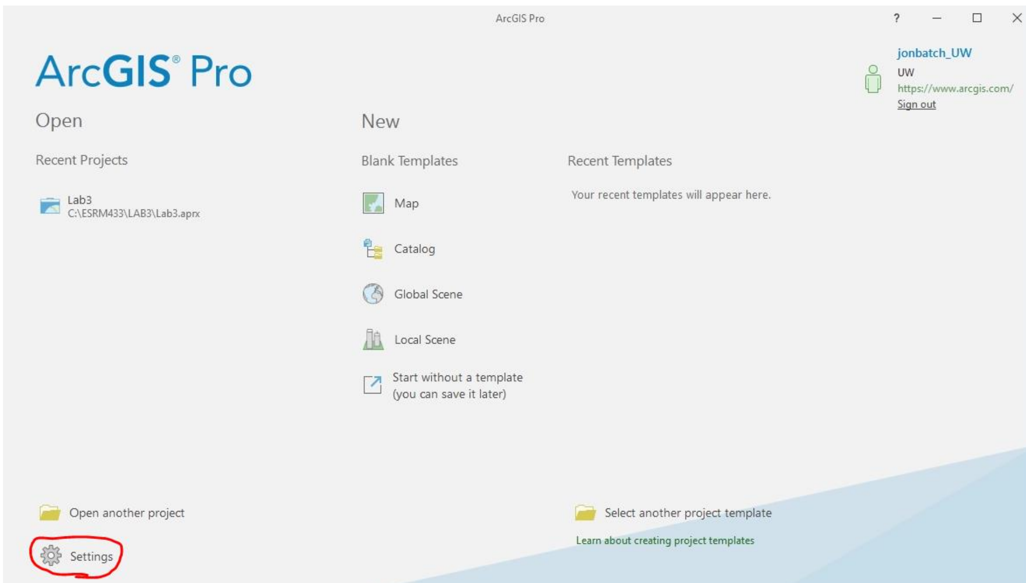
You want to download the DTM, Metadata, and Point Cloud for the Hoh 2013 acquisition. Do not download anything else. There are other acquisitions in the area but you only want to download these three things. Check you download size, it should be <u>883.04MB</u>. If it is bigger, you likely have more tiles selected for download. Redrawn your red polygon and make sure it is the size shown.

ESRM433/SEFS533
Lab 3

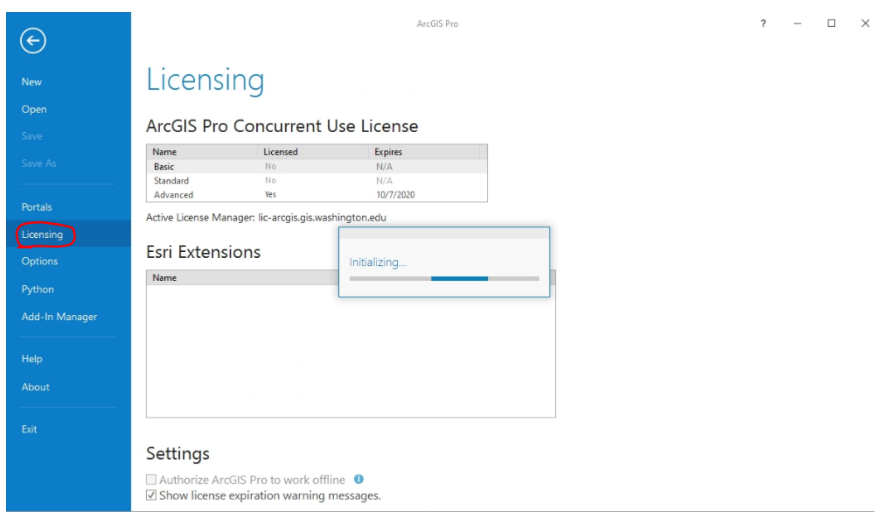
This is a rather large download so it may take a second. If it automatically downloads into your Downloads folder, cut and paste it into a folder you create at:
//fs-persona.sefs.uw.edu/student_redirect$/*YourNetID*/Desktop/ESRM433/LAB3

The first thing you should do is skim the metadata. Take a moment to note the point density and AOI. Having a full understanding of when the data was acquired, and the extent of the data will be beneficial to you.

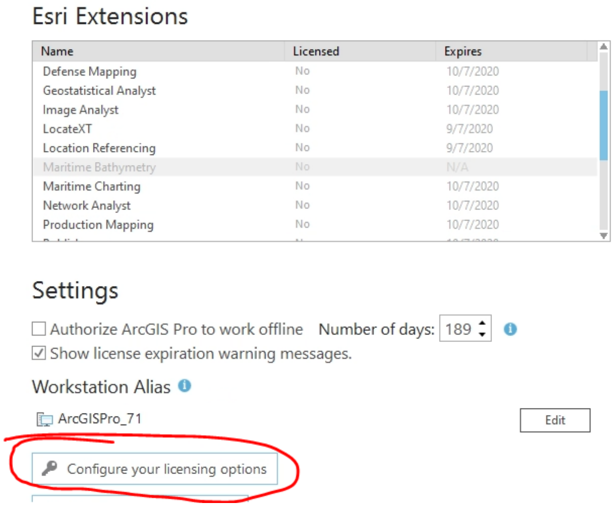
We are going to use ArcGIS Pro to look at the vendor supplied DTM. Start up ArcGIS Pro in Madrona or on your personal computer. As a UW student you are able to install a free copy of ArcGIS Pro on your own computer, but this lab will assume you are accessing ArcGIS Pro via Madrona.

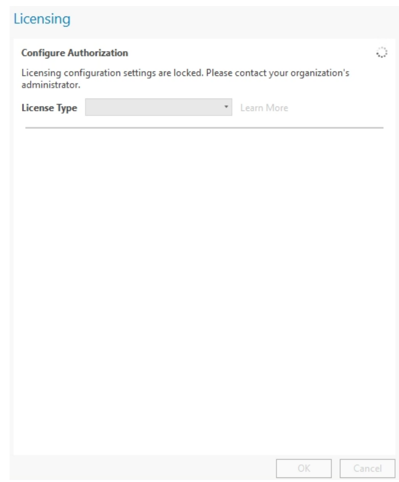


First you are going to select "Settings"
You need to sellect "Licensing"
For some reason, Madrona may take a while to populate the Esri Extensions box.

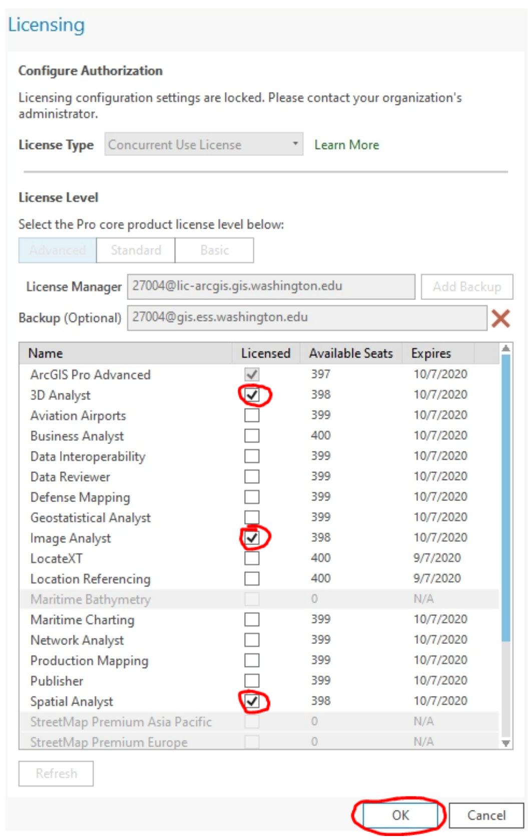


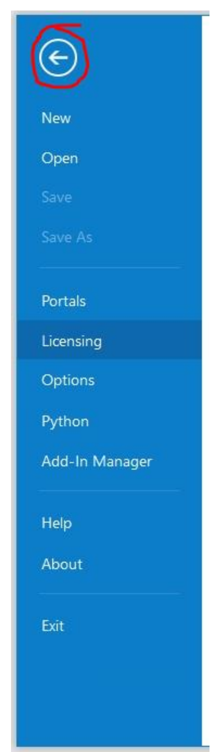
Once the Esri Extensions is populated, click on “configure your licensing options”. This may take a minute as ArcGIS checks the licensing options. Seeing a screen with a loading icon in the upper right hand corner is fine. Give a minute or two to get the licensing information.

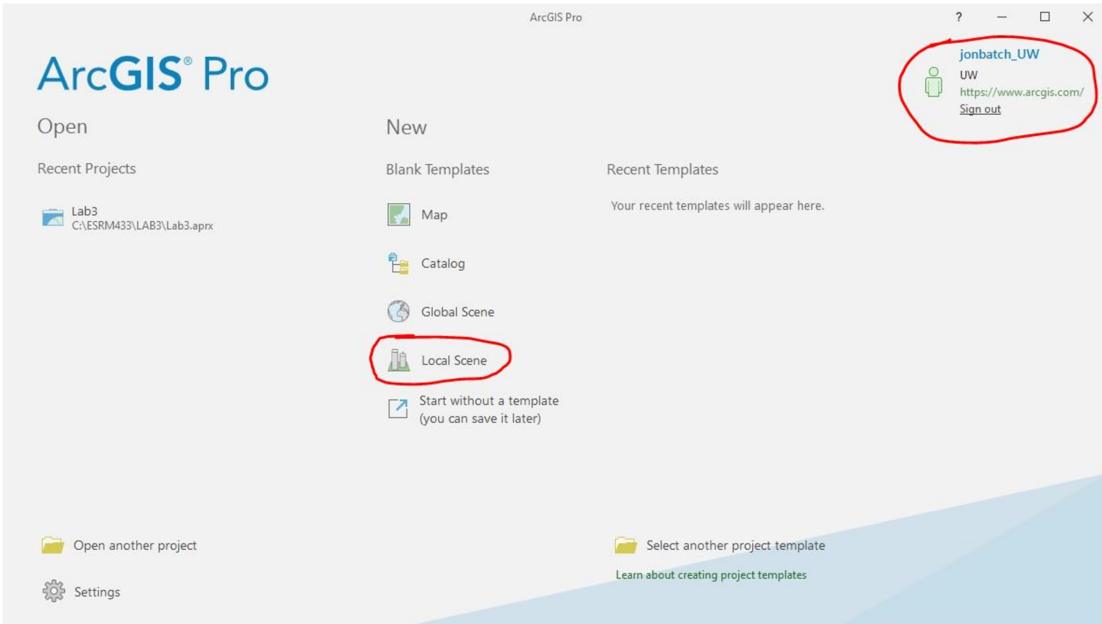




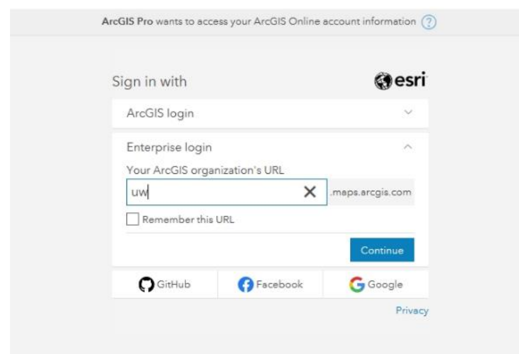
Once the licensing settings finally populate, you want to check the boxes for “3D Analyst”, “Image Analyst”, and “Spatial Analyst”. Click OK and then hit the back arrow to get back to the intro screen.







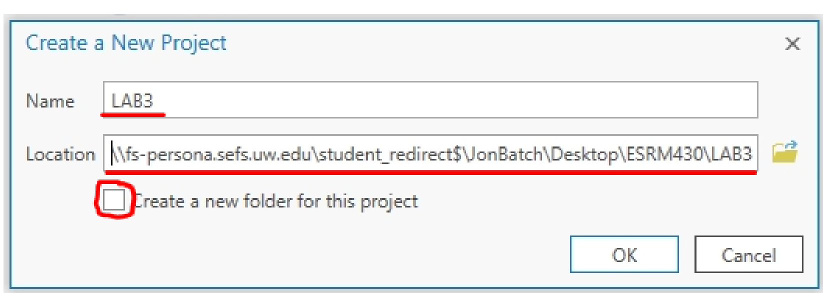
You can sign into ArcGIS Pro in the upper right-hand corner and use an enterprise login id of "UW". This should send you to a UW login screen. Where you can sign in to open up some additional options for maps in ArcGIS Pro.



Finally, you are going to create a new Local Scene. Local scenes are the 3D viewer of ArcGIS Pro with "Map" being a 2D viewer.
Name your new project "LAB3" and store in the location:
//fs-persona.sefs.uw.edu/student_redirect$/*YourNetID*/Desktop/ESRM433/LAB3
Uncheck the "Create a new folder for this project. You do not want to create a new folder.

If Madrona is struggling with the local scene, you can make a new 2D map instead.

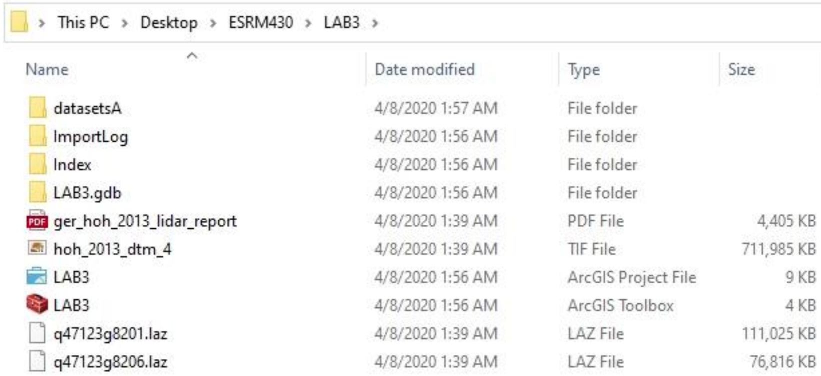
.



ESRM433/SEFS533

Lab 3

If you haven't already, unzip the custom_download zip file and drag the files you downloaded directly into your ESRM433\LAB3 folder. You can then delete the custom_download.zip and the unzipped custom_download folder. Your LAB3 folder should look like this:

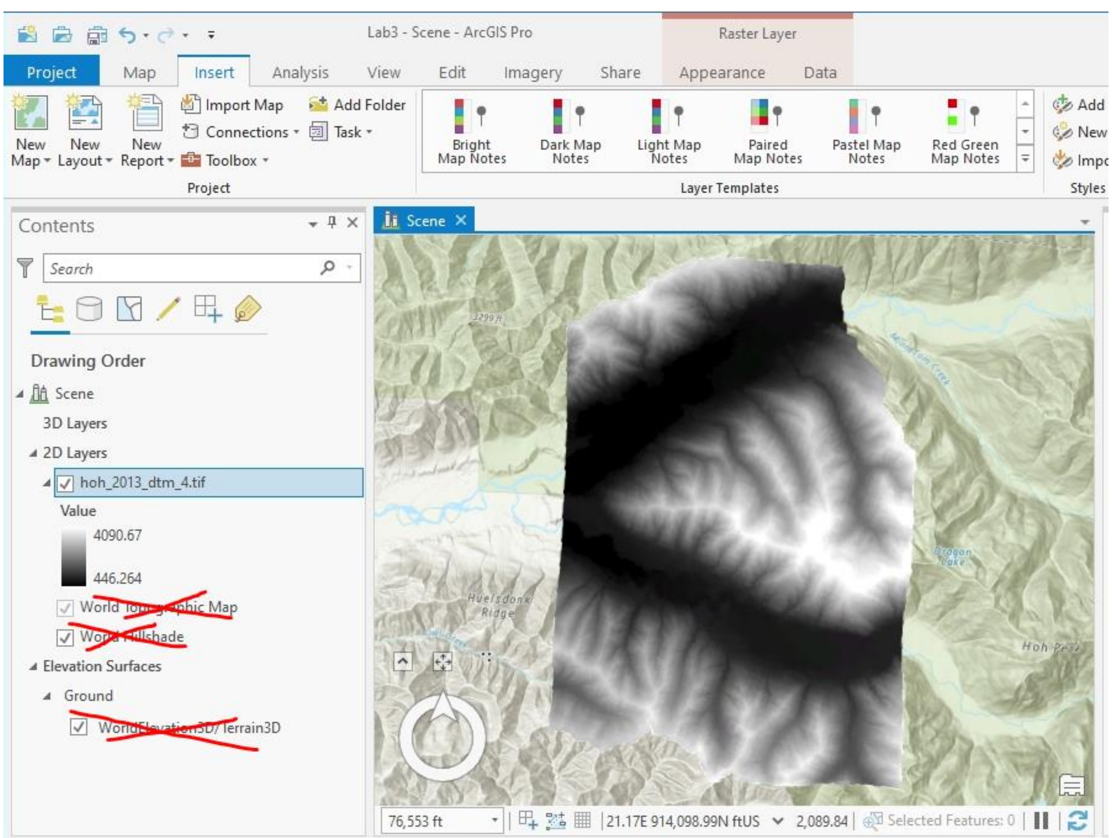


> This PC > Desktop > ESRM430 > LAB3 >

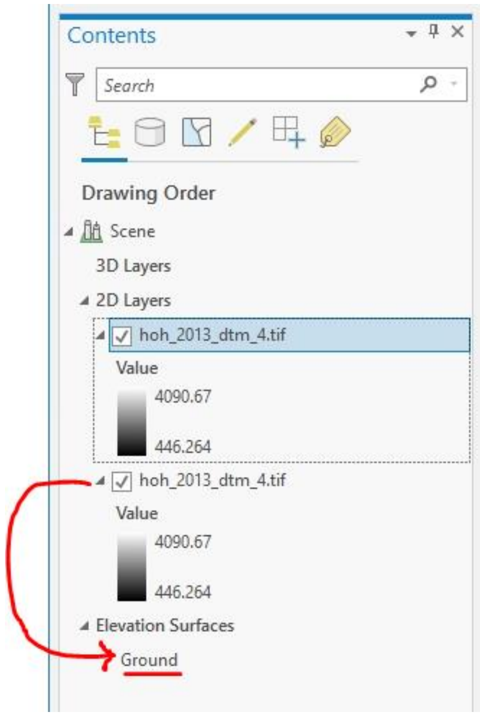
Name	Date modified	Type	Size
datasetsA	4/8/2020 1:57 AM	File folder	
ImportLog	4/8/2020 1:56 AM	File folder	
Index	4/8/2020 1:56 AM	File folder	
LAB3.gdb	4/8/2020 1:56 AM	File folder	
ger_hoh_2013_lidar_report	4/8/2020 1:39 AM	PDF File	4,405 KB
hoh_2013_dtm_4	4/8/2020 1:39 AM	TIF File	711,985 KB
LAB3	4/8/2020 1:56 AM	ArcGIS Project File	9 KB
LAB3	4/8/2020 1:56 AM	ArcGIS Toolbox	4 KB
q47123g8201.laz	4/8/2020 1:39 AM	LAZ File	111,025 KB
q47123g8206.laz	4/8/2020 1:39 AM	LAZ File	76,816 KB

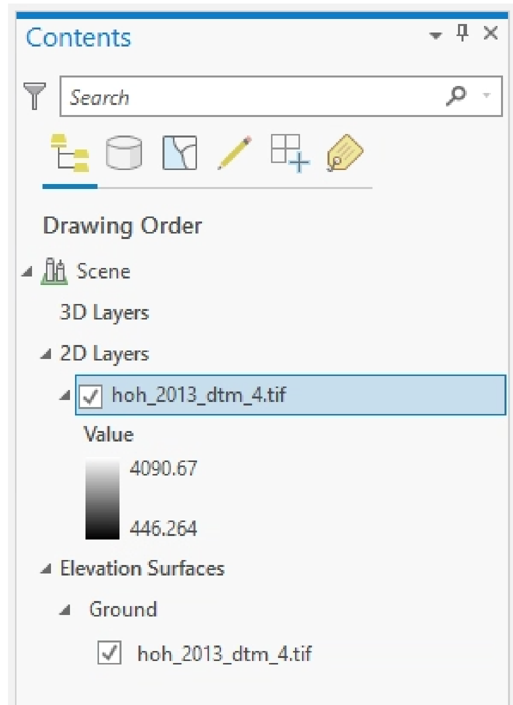
For instruction on how to navigate around in ArcGIS Pro, and especially a local scene, go to this ESRI topic page: https://pro.arcgis.com/en/pro-app/get-started/navigate-your-data.htm

Drag and drop your hoh_2013_dtm_4 file directly into your ArcGIS local scene. It will likely take a minute to load. It is ~712MB so not a small file. Once it loads, delete everything else in the scene. Delete the world topographic map, the world hillshade, and the WorldElevation3D/Terrain3D layers. You can just highlight them and press delete. We want to use our own, much higher resolution, DTM as the elevation surface.

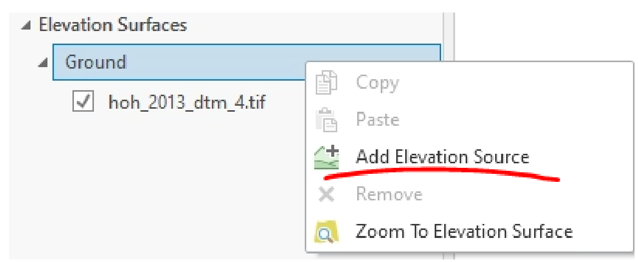


To set our DTM as the elevation surface, the easiest way to do it is to select hoh_203_dtm_4.tif layer and press Ctrl+c to copy it, and then press Ctrl+v to paste it (keystrokes are your friend). Then drag it onto the word “Ground”.

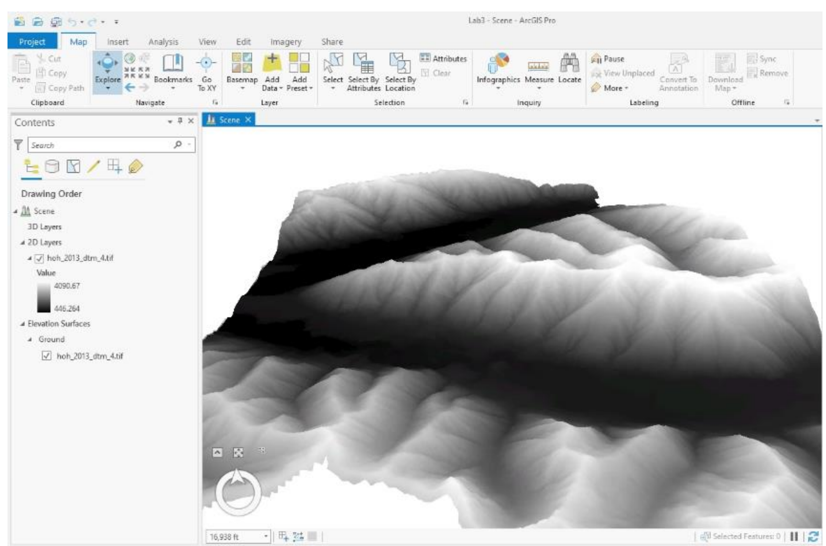




You can alternatively right click on “Ground” and select “Add Elevation Source” and navigate to the DTM in your file directory.

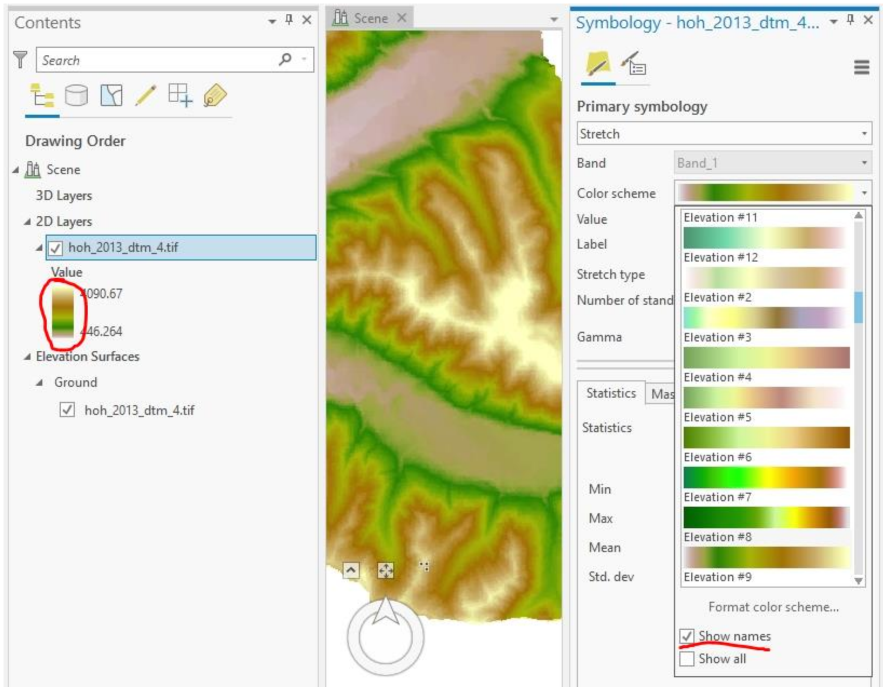


You should now be able to navigate around in your local scene. A mouse is extremely helpful in being able to fully explore the scene. If you don’t have a mouse, you can click on the tools in the bottom left hand corner to expand them and you can use them to navigate the 3D scene. More information about navigating the scene can be found here: https://pro.arcgis.com/en/pro-app/get-started/navigate-your-data.htm

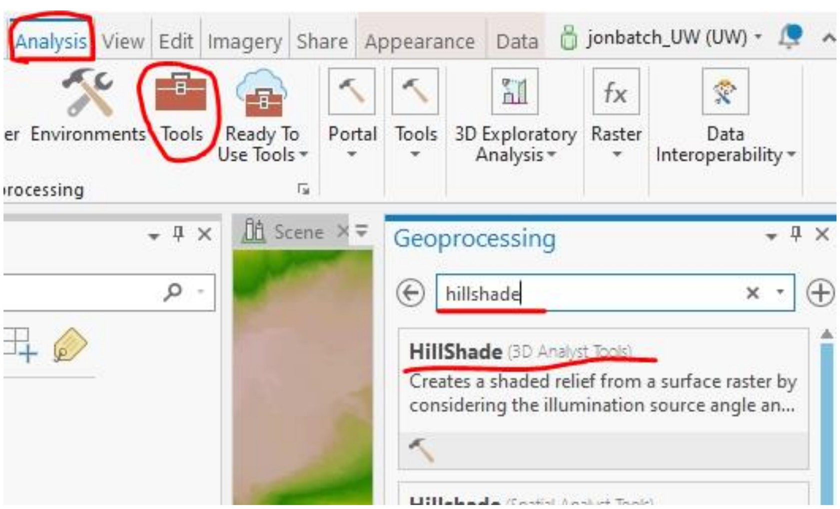


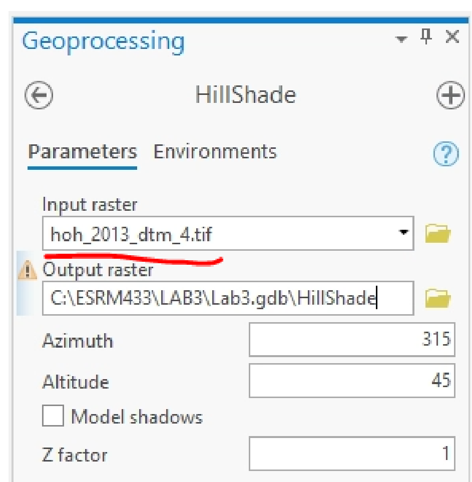
ESRM433/SEFS533
Lab 3

You can apply an elevation color scheme by double clicking on the color gradient in the contents menu. Pick any elevation color scheme you like.

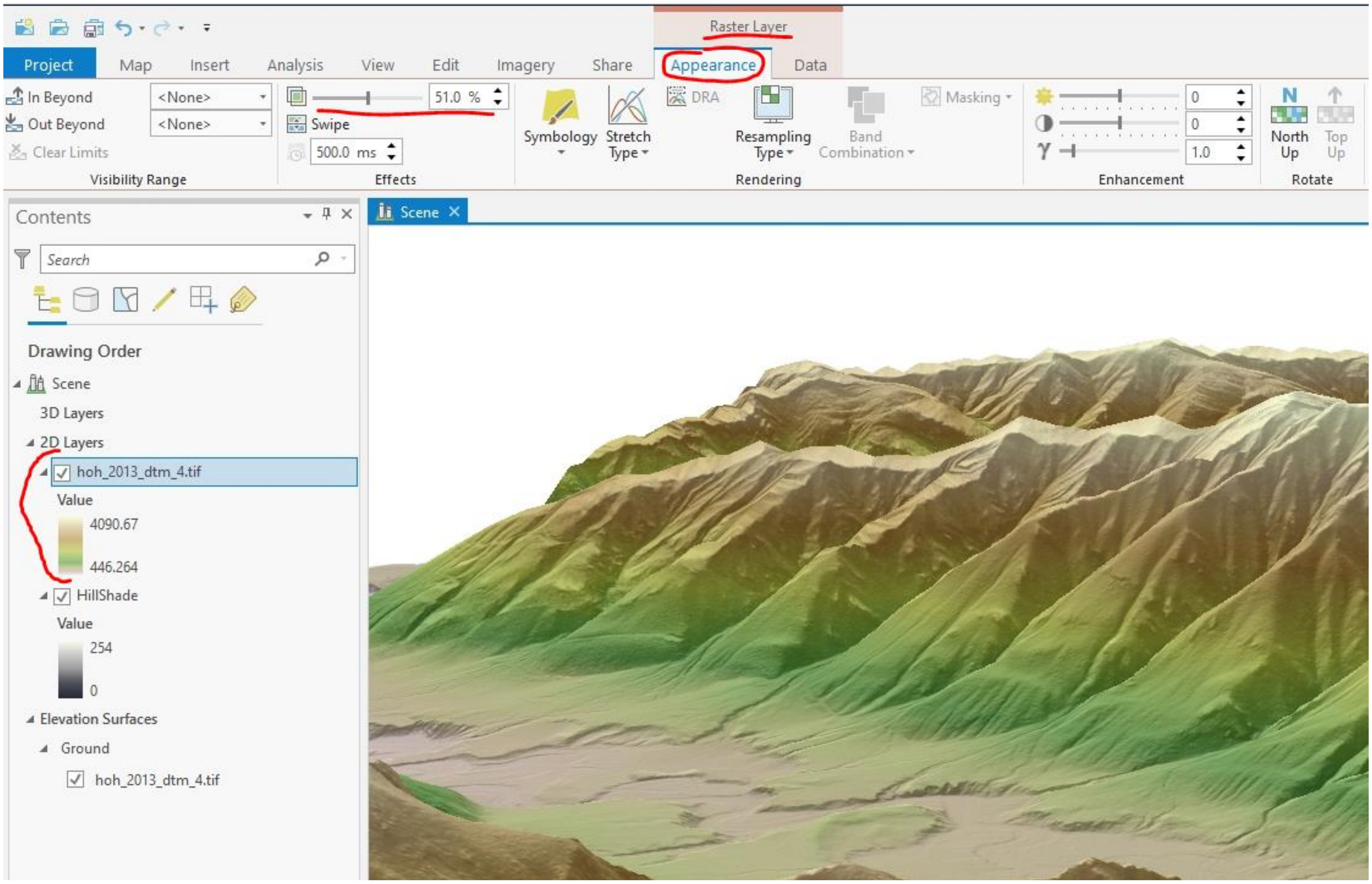


Now lets create a hillshade of the DTM. Go to Analysis > Tools > HillShade. Your input is the vendor supplied DTM and you can name the output raster anything you like. Note that it will be saved in the Lab3.gdb (geodatabase), which is fine.





Drag your Elevation colored DTM above your hillshade and set it's transparency to about 50%



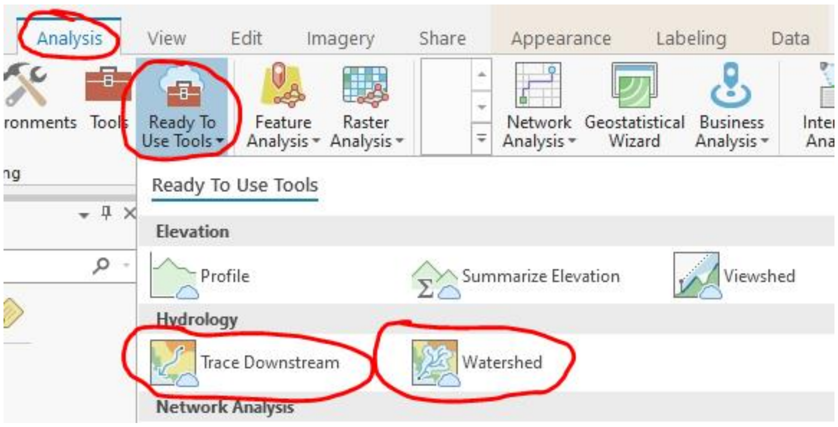
Take a moment to really explore the scene. Remember, this whole area is covered in trees, but we are seeing the bare ground in a heavily forested area. Note how well the braiding of the stream is captured. If you want to check out the scene with trees, you can do these same steps with the vendor DSM.

QUESTION 3: Take a screenshot of some part of the 3D image that is particularly interesting. Feel free to change the color scheme or the transparency. Compose an amazing piece of Lid-Art. Caption the screenshot with what the image is of.

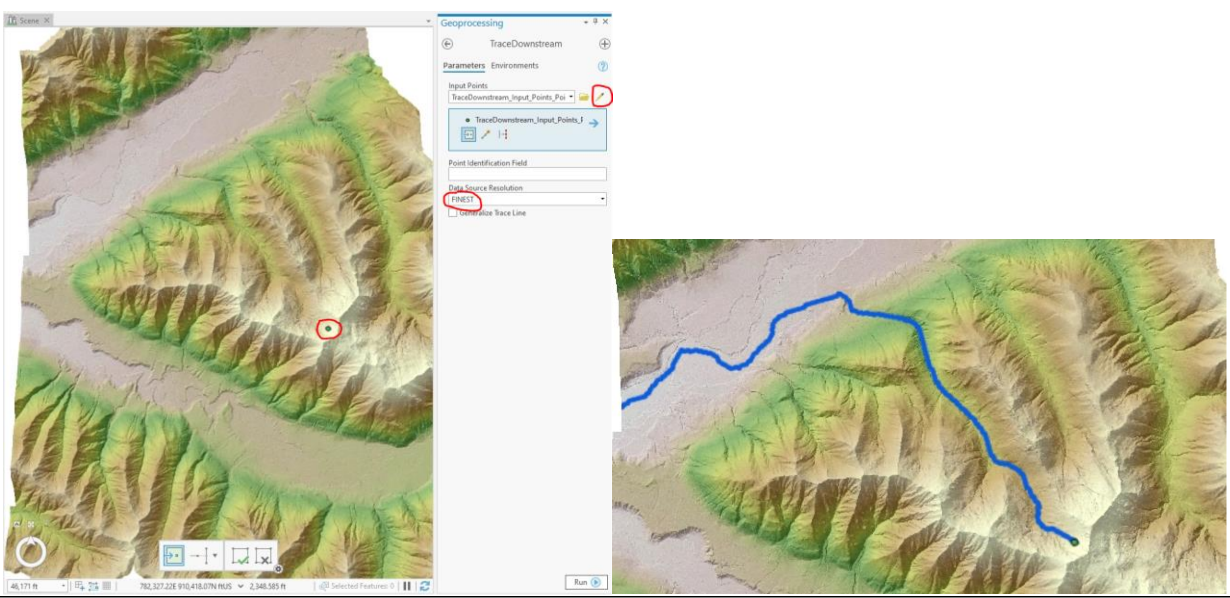
ESRM433/SEFS533
Lab 3

So far everything we have done has been a GIS exercise to make a cool looking 3D image. The point is to help you fully understand the detail possible with lidar. GIS programs like ArcGIS or QGIS have many tools that use DEMs to preform analysis for outputs like: Slope, Aspect, Solar Radiation, Viewshed, and many hydrology related tools. Going into depth about these tools is beyond the scope of this lab.

Two tools that we are going to check out using our lidar derived DTM are the hydrology tools in the "Ready to use Tools" toolbox.



The Trace Downstream tool uses a hosted digital elevation model (DEM) to trace downstream flow paths. The resolution of the DEM used depends on the location of the analysis and will be recorded in the `Data Resolution` field of the result layer. https://doc.arcgis.com/en/arcgis-online/analyze/trace-downstream.htm

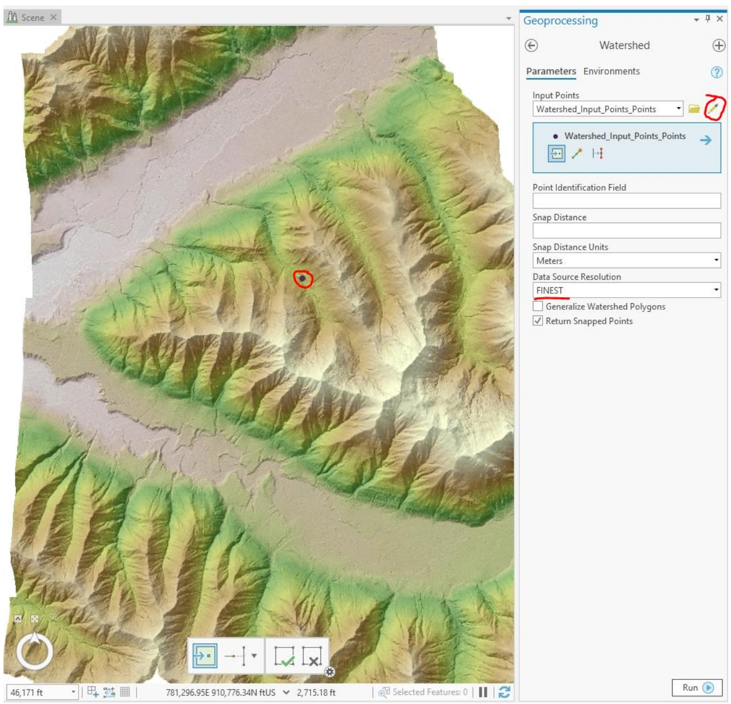


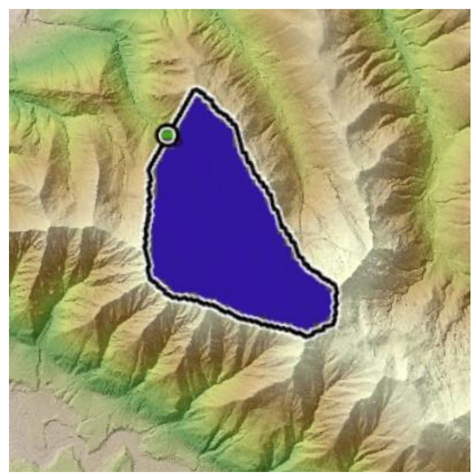
The Create Watersheds tool uses a hosted digital elevation model (DEM) to create water catchment areas. The resolution of the DEM used depends on the location of the analysis and will be recorded in the `DataResolution` field of the result layer.

For the result watersheds to be relevant, the input points must be located on drainage lines. The **Search distance to nearest drainage** option will move the input points to the nearest drainage line within the search distance. If you do not specify a search distance, the tool will compute and use a conservative search distance.

For analysis purposes, drainage lines have been precomputed by Esri using standard hydrologic models. If there is no drainage line within the search distance, the location containing the highest flow accumulation within the search distance is used.

https://doc.arcgis.com/en/arcgis-online/analyze/create-watersheds.htm





QUESTION 4: Use the Trace Downstream and the Watershed tools and include screen shots of the output from each tool. Try using more than just a single point to generate an output.

Save your Lab3 ArcGIS project, but leave it open as we are going to be bringing products from CloudCompare and lidR into it.

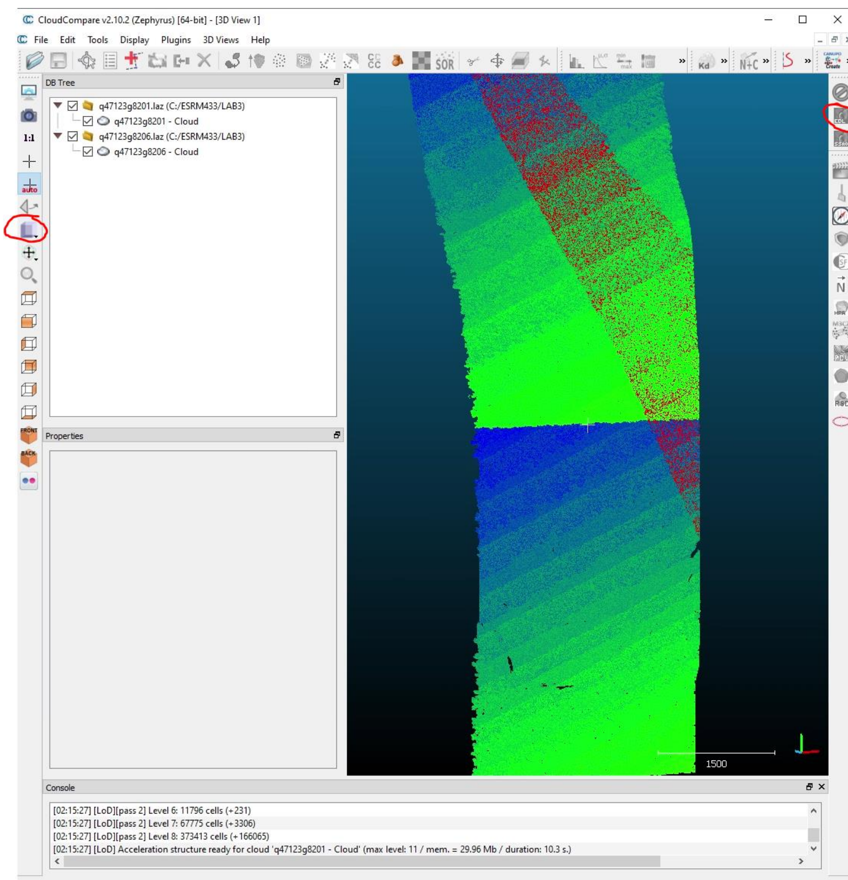
ESRM433/SEFS533

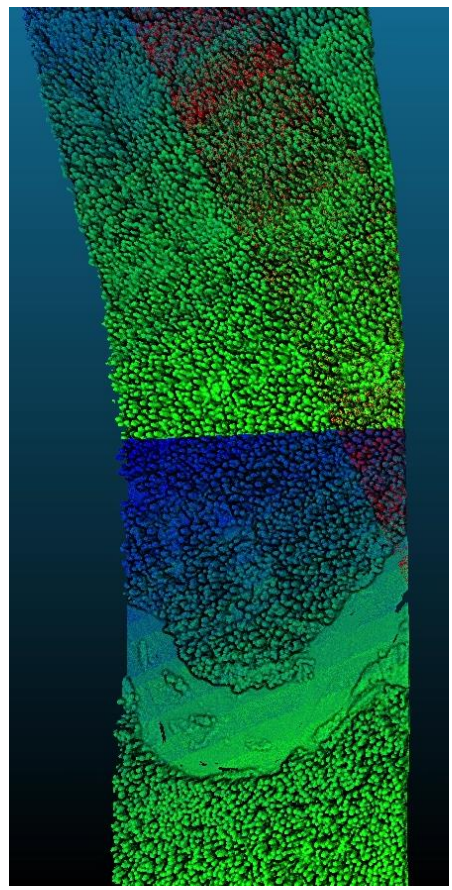
Lab 3

PART 3: Generating a DTM

Now that you can generate amazing 3D maps and have some experience with some of the tools that use DEMs, how is a DTM actually made from a lidar point cloud? The tiles that you downloaded are only a very small section of the overall area but they still have almost 40 million points between the two tiles.

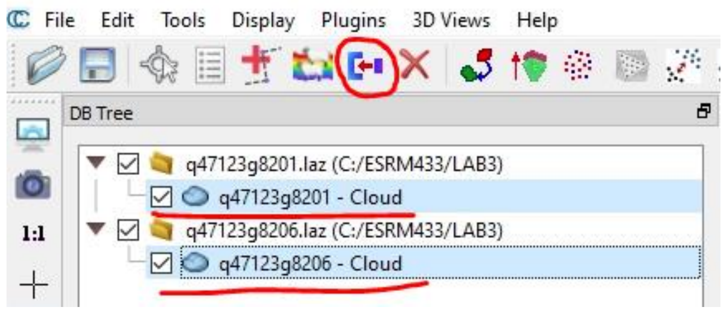
Open CloudCompare and import the two laz files into the program. They will take a moment to load. When they are done loading, they will likely not look like much. Make sure your view is set to object-centered perspective and your turn on the EDL shader. The point clouds should become more recognizable. You now see how ALS acquisitions are broken up into multiple tiles. This is mainly due to the computational power required to deal with lidar data. You are likely noticing how Madrona is struggling with only two tiles being visualized. To get a cohesive end product, the tiles have to be joined together. Automating this process is the main reason we will be using lidR for the majority of actual lidar processing, but to conceptually go through the process, we are first going to use CloudCompare.





ESRM433/SEFS533
Lab 3

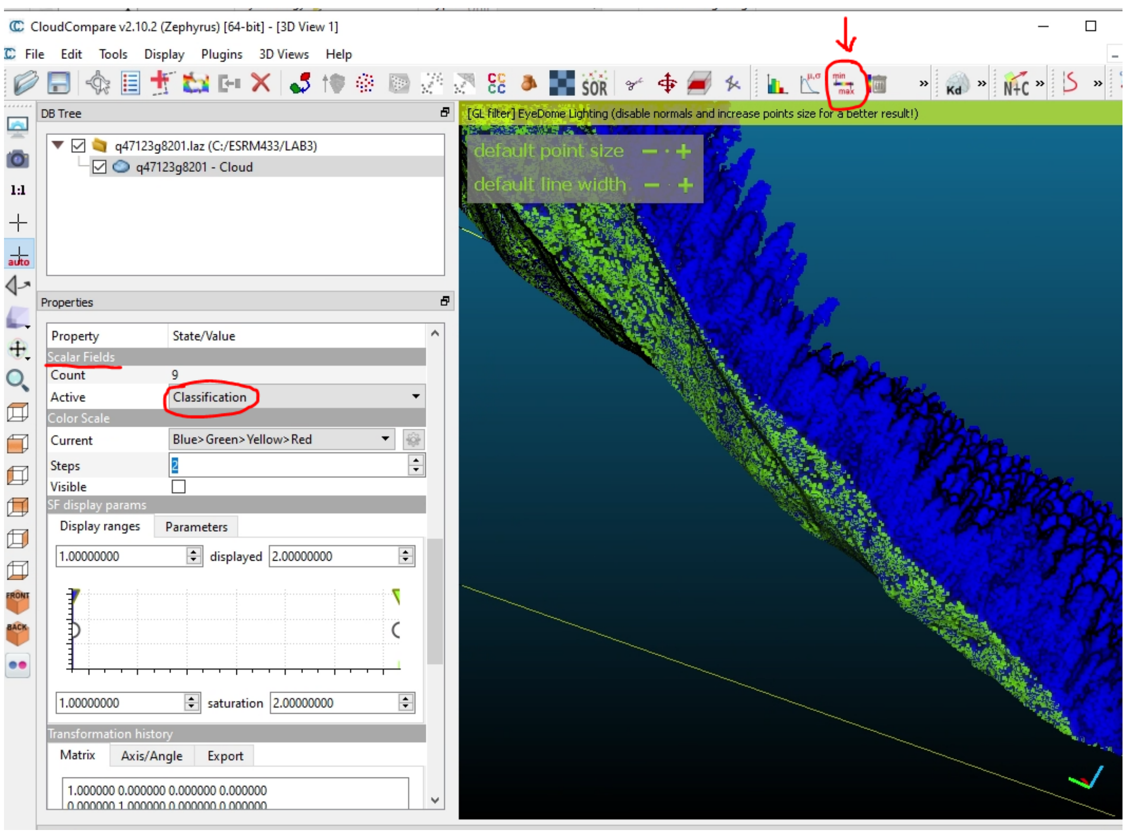
Merging multiple clouds is as simple as pressing a single button. With both clouds selected, press the "Merge Multiple Clouds" button.



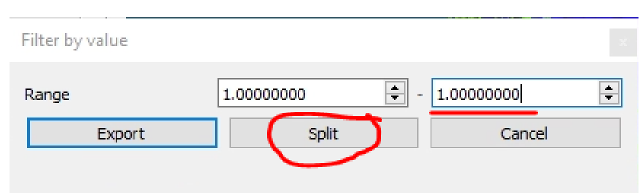
You do not want to create a scalar field with the original cloud information.

Once the clouds are joined together, we are going to separate the ground points and non-ground points based on the vendor classification that is already encoded in the las file. Remember the default las field "Classification"? We are going to take advantage of this pre-existing classification. With your joined cloud selected, make the active scalar field "Classification". The first thing you should notice is that the ground is green, but all the trees are blue. This is a good sign.

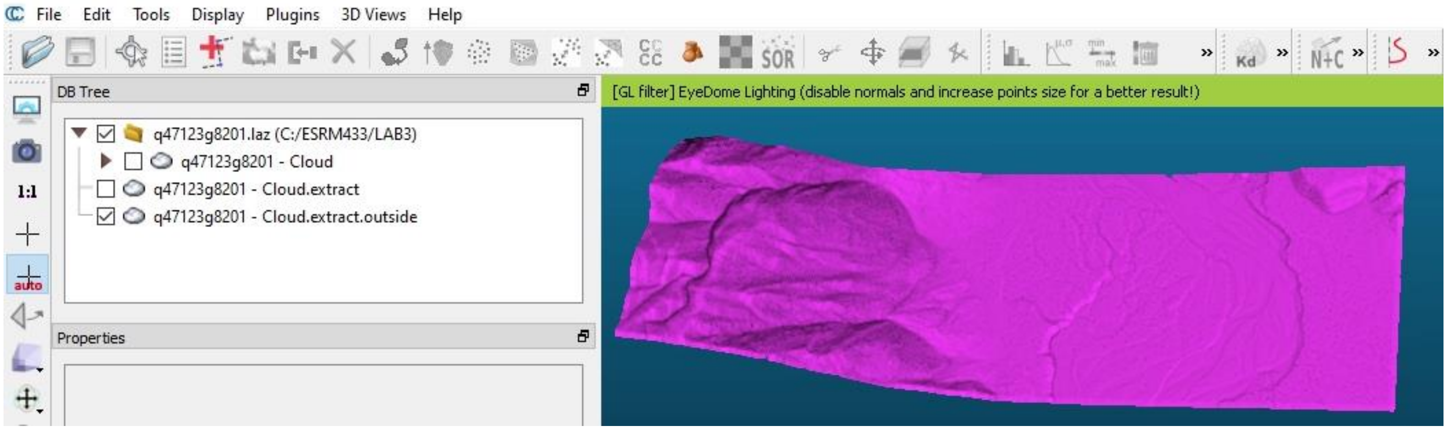
Click on the "filter points by value" button . This button should be in your tool bar, but it can also be found in Edit > Scalar fields > filter by value.



There are only two values in the classification.
1 = Unassigned
2 = Ground

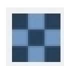


Set the range to 1 – 1 and Split the cloud.
This will take a moment and you will likely get a "not responding" note on CloudCompare. That is OK, just wait.



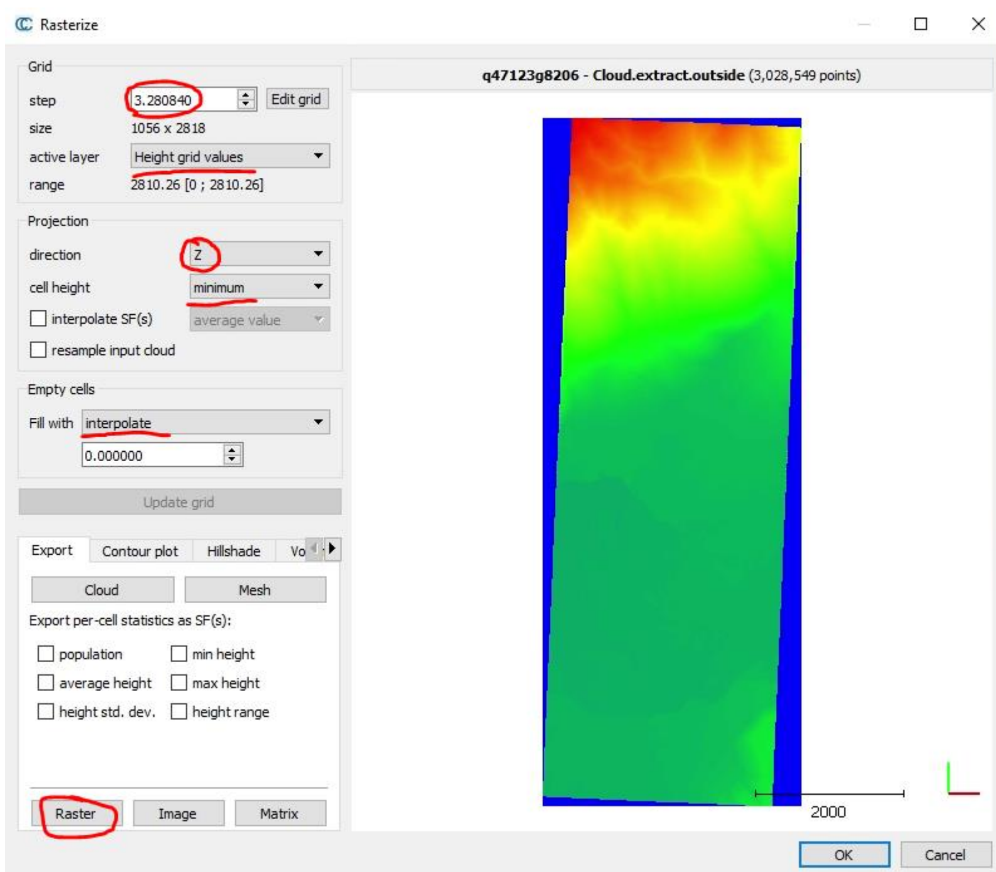
You could have set the range to 2-2 and still had a similar split as there are only two classification values in this point cloud. Conventionally, classification 2 is for ground in a las point cloud, but these values can be changed so don't assume that class 2 will always be ground.

If you turn off your "not-ground" layer, you should see what looks like a small section of the DTM that we were working with in ArcGIS. However, remember that this isn't a solid surface. It is only a tightly packed collection of single points. If we want to create a raster from this data, you can use the raster tool within CloudCompare just as you did in lab 2.

With your ground layer selected, open the raster tool. 

A few things to remember, all the data downloaded from the Washington DNR Website is in The projected coordinate system Washington State Plane South. The units are feet, not meters. If we want to create a 1 meter resolution raster, your Grid step needs to be 3.28084.
You can try to run a grid step of 1 to create a 1ft resolution raster, but it will take a LONG time for Madrona to process, so stick to creating a 1m resolution raster for now.

We want a DTM so we want the active layer to be height grid values and the direction to be Z.



Cell height, this is something you will have to think about. There isn't necessarily one correct answer. You just have to be able to justify your choice. We only have the ground points selected, but we are placing a 1m grid over the points so within each grid cell, do you want to use the height of the lowest, highest, or average of all points to assign that one cell a value? Each raster pixel will only have one value.

We want a solid surface with no holes, so we need to fill the void space with some values. Once again you have a choice between the lowest value withing the extent of the raster, the highest value, the average value, some specified value, or to interpolate. Interpolation using the values of the surrounding nearest pixels to generate a value for any cell that is missing a value.
When you fill the empty cells, CloudCompare makes a nice rectangle with the data and due to our projection, our tiles are a little crooked. That is why we get that odd extra fill space. For this exercise, we aren't going to worry about it.

When you have your values set, update the grid. This will take a moment. Notice that you can also generate a contour plot, hillshade, volume and a few other options. For a more extensive break down on the raster tool, go to: https://www.cloudcompare.org/doc/wiki/index.php?title=Rasterize

Once your grid is updated, click on Raster in the bottom left hand corner. We want to Export heights.
Save the file as CloudCompareDTM_1m and make sure you are saving it in Desktop>ESRM433>LAB3 folder.

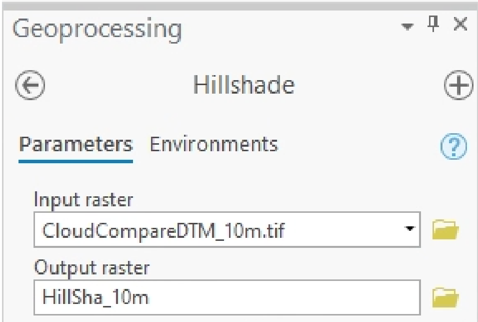
Do the same process again but change the resolution to 10m and save that DTM as CloudCompareDTM_10m

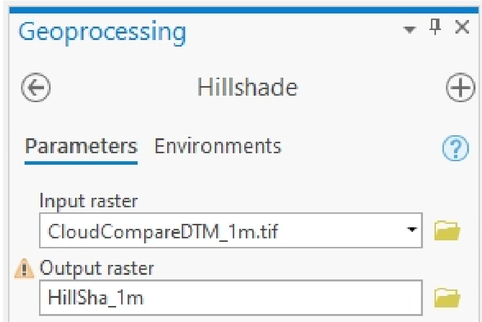
QUESTION 5: For the cell height, did you use minimum, maximum, or average values? Why? How would selecting the different values change the output raster as you decrease the spatial resolution (i.e. increase the grid step size)?

Drag and drop your two generated DTMs into your local scene in ArcGIS. Select yes when it asks if you want to build pyramids. Your two created DTMs should overlay directly on top of a portion of your already created Model from the DNR lidar data.

To really see the difference in the resolution, make hillshades of both 1m and 10m DTMs.

If Madrona is having issues running the local scene, you can do these steps in a 2D map. See video for more explanation.





QUESTION 6: Zoom into a location and grab screenshots of both the 1m resolution DTM Hillshade and the 10m resolution DTM Hillshade,

QUESTION 7: How do you think this resolution difference would impact the output from the hydrology tools you used earlier (Trace Downstream & Watershed)? The tools within ArcGIS likely don't have the sensitivity to actually produce a significantly different output, but more conceptually, how would a reduced spatial resolution impact the ability to trace stream paths?

Interlude to further discuss Coordinate Systems and Map Projections

This is more of a GIS conversation, but we can't continue without have some conversation about Coordinate Systems and Map Projections. Repeatedly it has been mentioned that data downloaded from the Washington DNR lidar site is in Washington State Plane South, but what does that actually mean, and what happens if you have data with mixed projections or there is an error in your coordinate system that is defined for one of your data sets?

Coordinate systems mismatches or errors are extremely common to come across when dealing with remote sensing data. Checking your data and being able to recognize errors is extremely important. As you can imagine, if data isn't lining up properly or is being placed in the wrong location, any analysis using that data will likely fail.

A good introductory video can be found here:
https://www.esri.com/videos/watch?videoid=PICwxT0fTHQ&title=introducing-coordinate-systems-and-map-projections

It is a little long and not all is applicable, but if you haven't dealt with map coordinate systems and projections before, I would suggest watching it and doing some additional research yourself.

A short description is that there are two basic types of coordinate systems, projected coordinate systems and geographic coordinate systems. Geographic coordinate systems are based on a spheroid and utilize angular units (degrees). Projected coordinate systems are based on a plane (the spheroid projected onto a 2D surface) and utilize linear units (feet, meters, etc.).

ESRM433/SEFS533
Lab 3

We will likely be only dealing with projected coordinate systems in this class. If you right click on the Hoh DTM you downloaded from the DNR site and go to Properties > Source > Spatial Reference you should see this:

Spatial Reference

Projected Coordinate System	NAD 1983 HARN StatePlane Washington South FIPS 4602 Feet
Projection	Lambert Conformal Conic
WKID	2927
Authority	EPSG
Linear Unit	US Survey Feet (0.3048006096012192)
False Easting	1640416.666666667
False Northing	0.0
Central Meridian	-120.5
Standard Parallel 1	45.83333333333334
Standard Parallel 2	47.33333333333334
Latitude Of Origin	45.33333333333334

I will be referring to the above figure in the last portion of this lab below.

This is all the information about how to interpret the numbers associated with the file that tells the GIS programs where to place the rasters or point clouds or any geographically tagged data. You looked at some of this data in the last lab when you were checking out the header for the las files using lidR. The key things to note are that this data is using NAD 1983 HARN and is StatePlane Washington South and that the units are feet. There is also a EPSG code of 2927. EPSG stands for European Petroleum Survey Group and is an organization that maintains a geodetic parameter database with standard codes, the EPSG codes, for coordinate systems, datums, spheroids, units and such alike. ... Every geographic object (coordinate system, spheroid, unit etc.) gets assigned a unique number.

The fact that the group governing the standardization of codes for coordinate systems is a petroleum group only speaks to the fact that this industry was one of the first to need to be able to have a way to convert between coordinate systems. They became the default largely because there was no need to duplicate the effort they had already put in.

With the EPSG number 2927, you can tell a program like ArcGIS how to interpret coordinate information contained within data like a GeoTiff.

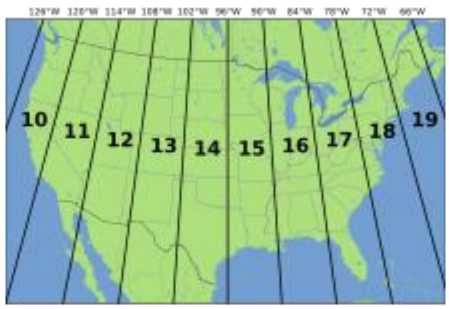
Go to https://epsg.io/

You can enter the code 2927 to find out the information about the NAD83(HARN) / Washington South (ftUS) attributes. The information here, should be the same as what is provided by ArcGIS. This website is a convenient way to get information about projections without having to access a GIS program.

For more information about map projections and coordinate systems check out:
https://www.gislounge.com/projections-and-coordinate-systems/

QUESTION 8: Look up the EPSG code 2856. It may look similar to code 2927 but there is one key difference under the Attributes. What is different between 2856 and 2927? This one difference necessitated the creation of separate codes. Discuss how this difference changed the Center coordinates, and the Projected bounds.

QUESTION 9: What is the EPSG code for the coordinate system NAD83(HARN) / UTM zone 10N?



This is another common coordinate system for Western Washington, note that eastern Washington is in zone 11N. This can be a source of confusion if you are working around the crest of the Cascade Mountains.

Always include your datum and zone information (or just the EPSG code) anytime you record position information!

PART 4: lidR and DTMs

IMPORTANT FIRST STEP. You need to make sure you are using the 64bit R and not the 32bit R. When starting R studio a message will appear telling you if you are using 32bit or 64bit. If you are using 32bit, you must go to Tools > Global Options and change the R version to the machine's default version of R (64bit)

In R, a # indicates a comment that won't be run in a line. I've included an appendix at the end of this lab that has # comments for every line of code from labs 2 & 3 for you to refer to.

We are going to start up RStuido. An easy and simple way that will give you a small head start is to open the Lab2.R file that should be in your LAB2 folder. This will bring in several lines of code that you are going to want to reuse.

Remember to set your working directory to your Madrona desktop, ESRM433 folder. Also remember to not use spaces in your folder and files names, and that CaPItaLizatiON matters.

```
install.packages("lidR")
install.packages("rgl")
library(lidR)
library(rgl)

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

########### Lab 2 ###############
LASfile <- ("Lab2/WA_031_rp.laz")
las <- readLAS(LASfile)
plot(las)
print(las)
print(las@header)
las_check(las)

plot(las, color="Intensity",trip=100)
plot(las, color="RGB")
```

Run a line of script by placing your curser on the line and press Ctrl+Enter.
Run the lines:

```
library(lidR)
library(rgl)
```

You shouldn’t need to download and install the packages again, just make sure that the two packages are loaded by running the library lines.

If you are having issues, make sure your setwd is pointing to the correct location, and you can rerun all the and library lines.

For information about packages within R (we will be using the packages lidR and raster extensively), you can run a help command:

```
help(package = lidR)
help(package = raster)
```

`help`	This tells R that you want the help documentation about something
`(package = lidR)`	This identifies that you want the help documentation about a package, and that package is lidR (or raster).

Take a moment to familiarize yourself with how the help documentation is laid out. If you want to explore all of the options within lidR, this is a good resource. You can look up the terms we have used so far like “readLAS”. If you get hung up on syntax for a command, or can’t figure out why something isn’t working, use this help documentation.

Back to our code…

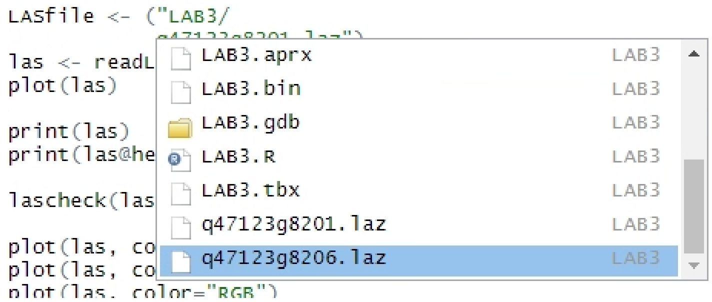
Change the line:

```
LASfile <- (“LAB2/wa_031_rp.laz”)
```

Change it to:

```
LASfile <- (“LAB3/q47123g8201.laz”)
```

You can see an example in the screenshot above. Also remember tab completion. If you type `("LAB3/` then press tab, a menu should pop up showing you all of the files and folders located in the LAB3 folder. This is assuming that you've already made your working directory your Madrona Desktop > ESRM433 folder. You can press tab and just click on the file you want.



Run the lines:

```
LASfile <- ("LAB3/q47123g8201.laz")
las <- readLAS(LASfile)
plot(las)
```

Plotting the file will take a moment and it might not be obvious that anything happened, but look for a popup window in your windows menu bar. 

If you get an error here, make sure you are using the 64 bit version of R and not the 32 bit. See the video for more information.

The rgl package just isn't as good as CloudCompare for point cloud visualization but you can still check out your point clouds in this manner from within RStudio. Once you've checked your lidar point cloud, you can close the RGL device popup window.

Lets check out the specifics about our laz file.
Run the lines:

```
print(las)
projection(las)
```

Pay attention to the line "coord. ref."

```
coord. ref.   : +proj=lcc +lat_1=47.33333333333334 +lat_2=45.83333333333334 +lat_0=45.33333333333334
 +lon_0=-120.5 +x_0=500000 +y_0=0 +datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0
```

Most of the values should look familiar. Compare the values given here to the values in the screen shot in the "Interlude to discuss Coordinate Systems and Map Projections". The one very important difference is the linear unit (+units). We know that data downloaded from the DNR portal should be EPSG: 2927 with units being feet. lidR thinks the units are meters. We will need to address this issue before we can compare our output from lidR with other data. This would be easy to miss and if you were processing a large amount of data we might spend a lot of time waiting for the processing and then realize our error only when looking where the output was geographically.

We can check just the number of the CRS using the code:

```
epsg(las)
```

This just output the epsg number of the layer. With out data, it is wrong.
We need to change the epsg code associated with our lidar data with the line of code:

```
st_crs(las) <- 2927
```

<u>This does not reproject the data, it just defines what the current projection is. It is much more computationally demanding to reproject data.</u>

You can check that the projection information was changed with running the lines again:

```
projection(las)
epsg(las)
```

OK, we are now ready to make ourselves some ground models using lidR!
We are going to use "rasterize_terrain" that is part of the lidR package. For information about grid_terrain, you can use the help documentation for lidR help(package = lidR), or you can simply type:

```
?rasterize_terrain
```

The ? is another way to access help documentation. "?" is good for specific functions while "help" is better for the full documentation for a package.
rasterize_terrain uses different algorithms to create a DTM. In CloudCompare, we just set a grid and used the point heights within each grid cell to create a raster. In rasterize_terrain, the DTM is created using much more sophisticated methods. The three options are: knnidw, tin, and kriging.

`algorithm`	function. A function that implements an algorithm to compute spatial interpolation. `lidR` implements knnidw, tin, and kriging (see respective documentation and examples).

QUESTION 10: Using the links and documentation from ?rasterize_terrain, provide the description information of the three algorithms. You aren't expected to know exactly how each algorithm works, but you should be able to find the description information using the help documentation.

Note that to run kriging, you have to install the package 'gstat' first. This can be done with:

```
install.packages("gstat")
library(gstat)
```

We are going to run knnidw and tin. You can run all three if you like, but the processing time can be somewhat long so be prepared to wait if you try kriging. You can tell RStudio to run both lines of code by highlighting them both before hitting Ctrl+Enter. This will take a while but as long as the little red stop sign in up, RStudio is running. Type:
USE res=3 and not 3.28 as I initially say in the lab video. R will likely crash if you use 3.28

```
dtm1 = rasterize _terrain(las, res=3, algorithm = knnidw(k=6L, p=2))
dtm2 = rasterize _terrain(las, res=3, algorithm = tin())
```

`dtm1 =` This is what we are going to call our output from grid_terrain. In this instance a "=" is being used but it serves the same function as a "<-". While there is some difference between "=" and "<-", the can often be used interchangeably.
`rasterize _terrain` This is the tool that we are using to create a DTM
`(las,` The data that we have already loaded into R that will be used to create the DTM

`res=3.28,` The resolution of the output raster. We want to create a 1m resolution end raster so use a resolution value of 3. You could leave this out and the default is 1 but that will also at least triple the processing time.
`algorithm =` The type of algorithm that you want to use (e.g. tin or knnidw)

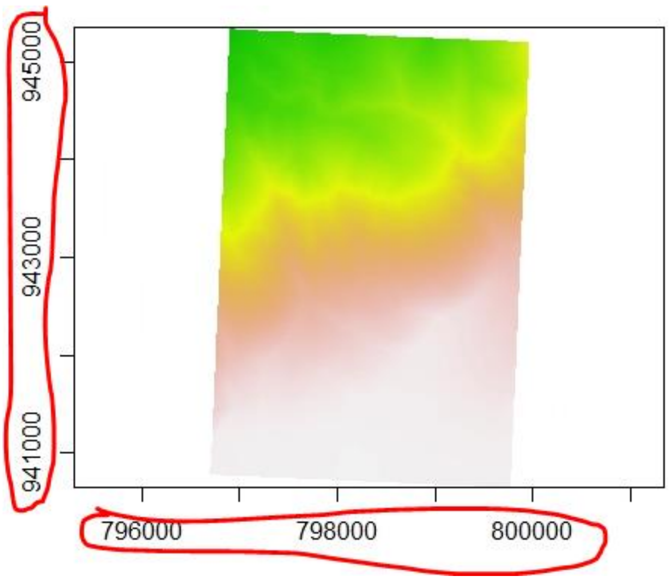
QUESTION 11: Refer back to the ?rasterize_terrain documentation. With the knnidw algorithm, what does k=6L, p=2 denote?

Once RStuido is done running, lets look at the output. Run:

```
plot(dtm1, main='knnidw')
plot(dtm2, main='tin')
```

The two plots likely look the same. The point cloud data is very high resolution and there isn't that much of a difference between the two methods at this scale.

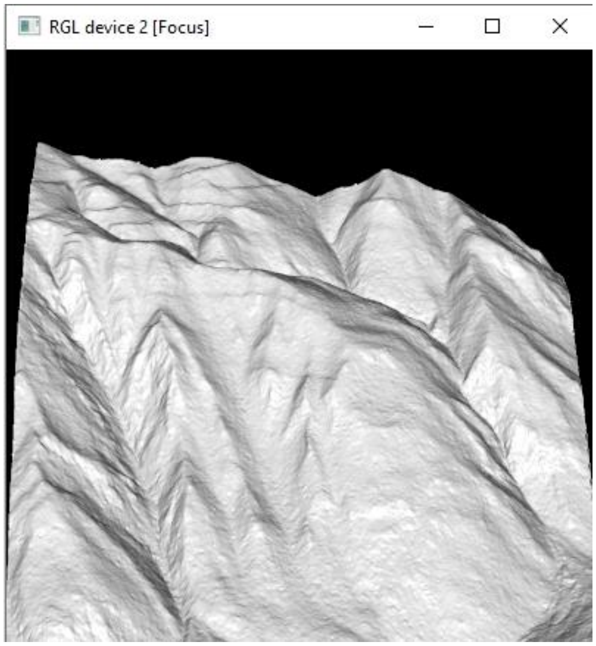
QUESTION 12: Include screen shots of your tin plot and your knnidw plot. Are you able to see any differences between the two plots? Make sure your title is visible in your screen shot.



QUESTION 13: Keeping in mind that the output raster has a CRS associated with it, what do the values along the x and y axis represent? What units are they in?

Lets take a look at the outputs in 3D. Run (remember you can use ?plot_dtm3d for more information):

```
plot_dtm3d(dtm1)
plot_dtm3d(dtm2)
```



QUESTION 14: include screen shots of your two 3D rendered DTMs. Compare how they differ and how they are the same. Do you think one is better? Why?

QUESTION 15: Do the models look like reasonable representations of the ground? Why or why not?

OK, you now know how to generate a DTM from a single lidar tile that already has points classified as “ground” by the vendor. Rasterize_terrain doesn’t identify the ground points itself, it uses the preexisting classification already encoded into the las file. What if you have lidar data that doesn’t have the ground points classified?

You can use help(package = lidR) to search for tools that have “ground” associated with them. Searching the help documentation is extremely helpful to find the correct tool and to see what tools are related. In this case we want “classify_ground”.

QUESTION 16: Using ?classify_ground you can gain information about the classify_ground tool. There are three algorithms that classify_ground can use. Pmf, mcc, and csf. You can gain more information about these algorithms by following the links. Make sure to read the descriptions. What paper is pmf based on? What paper is csf based on? What paper is mcc based on?

QUESTION 17: The Progressive Morphological Filter (PMF) needs two arguments defined to work, ws & th. What are these arguments? Directly quoting from the ?pmf documentation is OK.

QUESTION 18: The Cloth Simulation Filter (CSF) has 6 possible arguments. What are they? Directly quoting from the ?csf documentation is OK. Note that when an argument has a default value listed, the argument can be left out of the line of script and the default value will be used.

As you can imagine, running a ground classification filter on lidar data can be extremely demanding and time consuming. Therefore, we aren’t going to run las ground on the entirety of our data. First we are going to clip out a section of the data to just as a test to see how the ground filters preform. The clip function is also the same function you would use to clip out areas for research plots or anytime you have a specific location you want to clip out. Clip can take input geometry in the form of shape files. Make sure you check out ?clip
For our clip to test the ground filtering algorithms, we don’t have a specific location of interest, other than selecting an area on the hill that has trees. We need to know the spatial extent of our data so that we can select a location within the area. Luckily, you already know how to find out the extent of the lidar data. Run:

```
print(las)
```

The 'extent' line in the output tells you the minimum and maximum x and y values for the data as well as the area. Run:

```
Lasclip <- clip_circle(las, 797000, 944000, 100)
```

`Lasclip <-`	This defines the new object 'Lasclip' as the output from the following function
`clip_circle`	This is the lclip type (others are rectangle, polygon, shape, etc)
`(las, 797000, 944000, 100)`	This is the object to clip (las), the x center (797000), the y center (944000), and the radius (100)

Check out your new clip:

```
plot(Lasclip)
```

You will need to install the R package that allows for CSF to run. Run the lines of code:

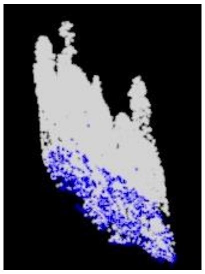
```
Install.packages("RCSF")
Library(RCSF)
```

We can now run pmf and csf algorithms on our clip without it taking forever to process. We will then compare the output from pmf and csf. Run:

```
ws <- seq(3,12,3)
th <- seq(0.1, 1.5, length.out = length(ws))
laspmf <- classify_ground(Lasclip, pmf(ws, th))

mycsf <- csf(TRUE, 1, 1, time_step = 1)
lascsf <- classify_ground (Lasclip, mycsf)
```

You now have two different clips of lidar data that have had their points reclassified from the original classification in the las file. You should get a message that there were already points classified as ground in the las file and that they were reclassified before the new classification was applied.

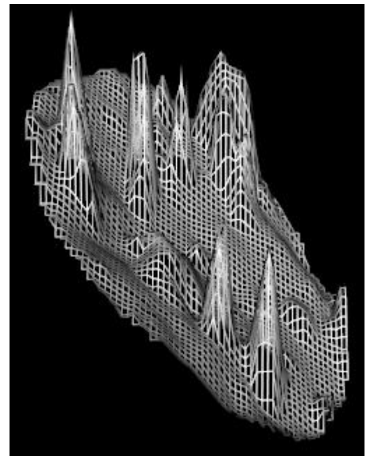


You can plot your new point cloud clips, coloring the points by classification by running:

```
plot(laspmf, color = "Classification")
plot(lascsf, color = "Classification")
```

Create a DTM from your clips by running:

```
dtmPMF = rasterize_terrain(laspmf, res=3, algorithm = tin())
plot_dtm3d(dtmPMF)
dtmCSF = rasterize_terrain(lascsf, res=3, algorithm = tin())
plot_dtm3d(dtmCSF)
```



QUESTION 19: Include a screenshot of your generated DTMs for laspmf and lascsf. Why do you think there were so many stray points left to make the dtmPMF "spikey"?

There isn't one best way to classify ground points in a lidar point cloud. For the pmf method, the window size and threshold heights needed for an open area will be very different that the values that should be used in a heavily forested area. The Cloth Simulation Filter may have provided us with a better result, but the processing time was considerably more. When processing lidar data, a heavily specialized computer with multiple cores will often be left running for days to chew though an entire acquisition.

I would encourage you all to play around with the different values for the PMF and CSF to see if you can get better results. You can also run the classification on the entire point cloud, but this will take time. The CSF is very slow. This is entirely "for fun" and not part of anything that will be graded, but if you do, let me know how it turns out.
You can also output the laz file of your laspmf and lascsf and compare the classification of the points in CloudCompare. To output the laz file, run:

```
writeLAS(laspmf, file = "Lab3/laspmf.laz")
```

Now that we have all this RStudio work under our belt, how can we quickly run a script that will combine our two tiles, correct the projection information, create a DTM and then output the raster so that it can be used in a GIS program?

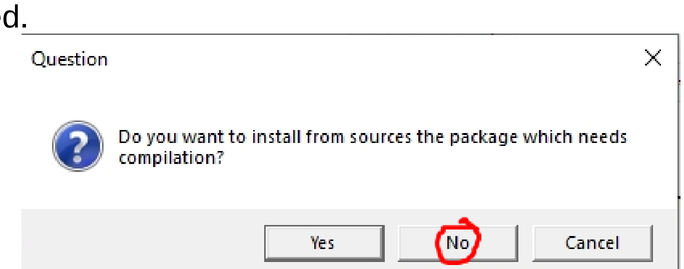
The answer is using readLAScatalog in lidR.
To use readLAScatalog, we need to install the package 'sf' (Simple Features)
As with the other packages, sf is already installed on madrona so just run:

```
library(sf)
```

If you are on your own computer, you will have to first run:

```
install.packages("sf")
```

There is a version of the sf package that is binary and needs compilation. You do not want that. Select No if you are asked.



Then run:

```
library(sf)
```

<u>One thing that we will want to change, is that we want our las files to be alone in their own folder. Copy the two las files you downloaded and put them in a subfolder called TILES. ESRM433>LAB3>TILES</u>

Run:
```
ctg <- readLAScatalog("Lab3/TILES/")
```
This command should look very familiar. Instead of telling R that you want to associate a single file with a name (ctg in this case), you are telling R that you want ctg to represent all the files in the folder LAB3/TILES as identified with the function 'readLAScatalog'

```
las_check(ctg)
```
This is the same function that you ran on the single file earlier, but it is smart enough to recognize that 'ctg' is a collection of las files in a catalog.

```
plot(ctg)
```
Instead of plotting the points in the lasfiles, this plots the location of the tiles and the amount of overlap (if any). You can install the mapview to view the tiles on a map, but that isn't required for this lab, we will get to that in future labs.

```
projection(ctg)
st_crs(ctg) <- 2927
```
Both of these are exactly the same as well. Checking the projection information of the catalog 'ctg' and then changing it using the epsg code

Note. In the lab video I initially talk about how rasterize_terrain is the updated version of grid_terrain, HOWEVER, rasterize_terrain is not working with LasCatalogs so we need to still use grid_terrain for this step.

```
TILEdtm = grid_terrain(ctg, res = 3, algorithm = tin())
```
This is the same rasterize_terrain as run earlier, but once again, it is smart enough to recognize that this is a collection of tiles. It is still relying on the vendor defined ground points to create the dtm, but this will make a seamless DTM from all the tiles in the catalog.

```
plot(TILEdtm, main='TILE')
plot_dtm3d(TILEdtm)
```
Both of these commands are identical as you ran before as you are now only dealing with a single raster DTM that can be plotted in either 2D or 3D.

Note. To use the following code to write the raster output, you need to install a package and load two packages:
Install.package("rgdal")
library(raster)
library(rgdal)
see the R code appendix.

```
writeRaster(TILEdtm,filename=file.path("LAB3/TILEdtm.tif"),format="GTiff",overwrite=TRUE)
```

Last command to 'writeRaster'. This takes your created raster 'TILEdtm' and creates a geoTIFF (GTiff) in the LAB3 folder. Overwrite=TRUE indicates that if there is already a file in your folder named "TILEdtm.tif", it will overwrite it.

QUESTION 20: Include 2 screenshots. A screenshot of your output from plot(ctg) and your output from plot(TILEdtm, main='TILE').

QUESTION 21: How do you expect the quality of ground models to change under different canopy conditions?

QUESTION 22: Under what sort of conditions would a low-density LiDAR acquisition provide satisfactory ground models? Under what sort of conditions would a high-density acquisition be required?

Last step to go full circle. Take your output TILEdtm.tif file and import it into ArcGIS. You should be able to just drag and drop it.

QUESTION 23: In ArcGIS, compare the vendor supplied DTM, the 1m DTM created with CloudCompare, and the TILEdtm from lidR. Are there any differences you notice? Zoom into a feature and include a screen shot the feature from each of the 3 dtms.

QUESTION 24: Assume you were given a laz file called NoClassified.laz. As indicated in the name, the file has no ground points classified. You are told that the projection should be Washington StatePlane South with units in feet, but that the file doesn't have an encoded coordinate system. You have a very fast computer so processing time is not an issue. Write out a section of R code that would take this file and classify ground vs not ground, and then create a DTM in the correct projection. Please write the code in a succinct manner in RStudio and include a screenshot. Comment each line of code with a #description. See appendix below for an example of a screen shot of R.

GRADUATE STUDENT

You cited 3 papers in question 16. Write a short paper summary on one of them. One or two paragraphs will suffice.

DELETE ALL EXTRANIOUS DATA LIKE ORIGINAL ZIP FILES OR ANY DUPLICATES YOU HAVE MADE.

R Appendix

```
install.packages("lidR")
install.packages("rgl")
install.packages("RCSF")
install.packages("rgdal")

library(lidR)
library(rgl)

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

########### Lab 2 ###############
LASfile <- ("Lab2/WA_031_rp.laz")
las <- readLAS(LASfile)
plot(las)
print(las)
print(las@header)
las_check(las)

plot(las, color="Intensity",trip=100)
plot(las, color="RGB")

########### Lab 3 ###############
library(lidR)
library(rgl)
library(RCSF)
library(sf)
library(raster)
library(rgdal)

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

help(package=lidR)
help(package=raster)

LASfile <-("Lab3/TILES/q47123g8206.laz")
las <- readLAS(LASfile)
summary(las)
plot(las)

print(las)
projection(las)
epsg(las)
st_crs(las) <- 2927
```

```r

?rasterize_terrain
dtm1 = rasterize_terrain(las, res=3, algorithm = knnidw(k = 6L, p = 2))
dtm2 = rasterize_terrain(las, res=3, algorithm = tin())
plot(dtm1, main='knnidw')
plot(dtm2, main='tin')
plot_dtm3d(dtm1)
plot_dtm3d(dtm2)

?classify_ground
Lasclip<-clip_circle(las,797000,944000,100)
plot(Lasclip)

ws <- seq(3,12, 3)
th <- seq(0.1, 1.5, length.out = length(ws))
laspmf<-classify_ground(Lasclip,pmf(ws,th))
mycsf <- csf(TRUE,1,1,time_step = 1)
lascsf <- classify_ground(Lasclip, mycsf)

plot(laspmf, color = "Classification")
plot(lascsf, color = "Classification")

dtmPMF = rasterize_terrain(laspmf, res = 3, algorithm = tin())
plot_dtm3d(dtmPMF)
dtmCSF = rasterize_terrain(lascsf, res = 3, algorithm = tin())
plot_dtm3d(dtmCSF)

writeLAS(laspmf, file="Lab3/laspmf.laz")

ctg <- readLAScatalog("Lab3/TILES/")
las_check(ctg)
plot(ctg)
st_crs(ctg) <- 2927
projection(ctg)

TILEdtm <- grid_terrain(ctg,3,tin())
projection(TILEdtm)
plot(TILEdtm, main='TILE')
plot_dtm3d(TILEdtm)
writeRaster(TILEdtm,filename = "Lab3/Tiledtm.tif",format="GTiff",overwrite=TRUE)

```