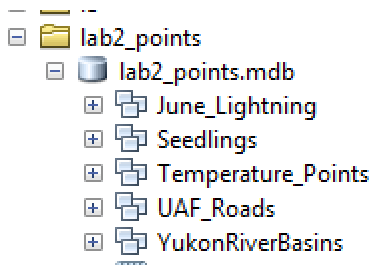
Lab2: More ArcGIS Points Analysis

In this lab we will work with points in several geodatabase feature class containers:

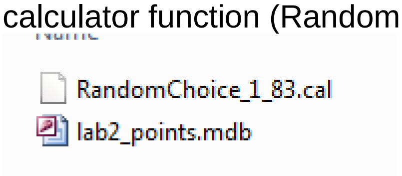


You will:

- Create uniformly spaced points within polygons
- Create random points within polygons and along lines
- Randomly choose from uniformly located points
- Create lightning density polygons from points
- Create a linear model of the effect of density on seedling height
- Interpolate point temperatures to a raster surface

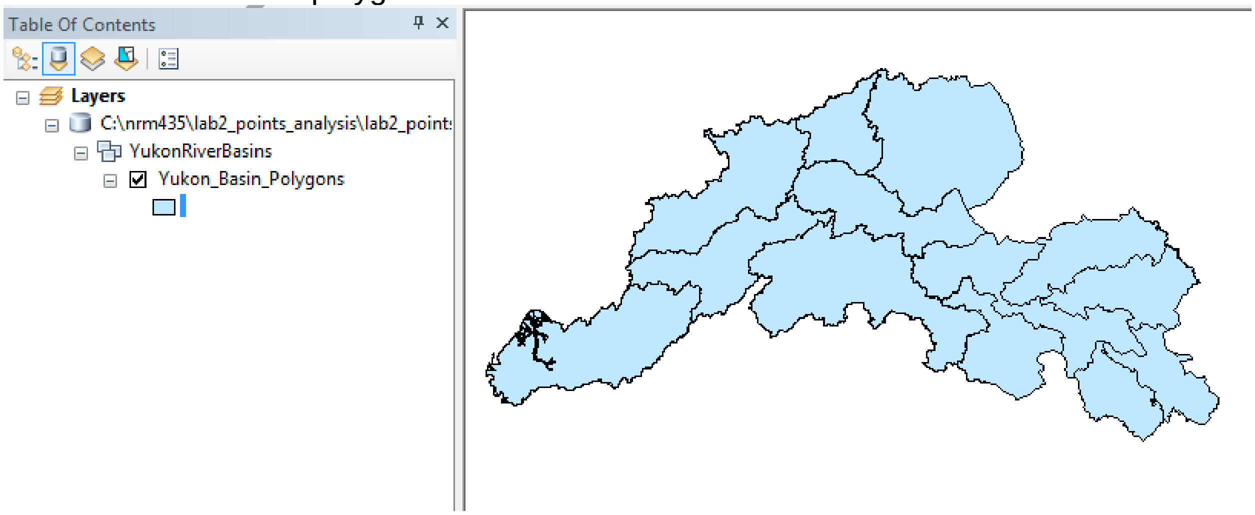
Download and unzip the geodatabase ***lab2_points.zip*** from
http://dverbyla.net/nrm435/data

The folder will contain a geodatabase container (lab2_points.mdb) and a field calculator function (RandomChoice_1_83.cal) you will use in arcmap:

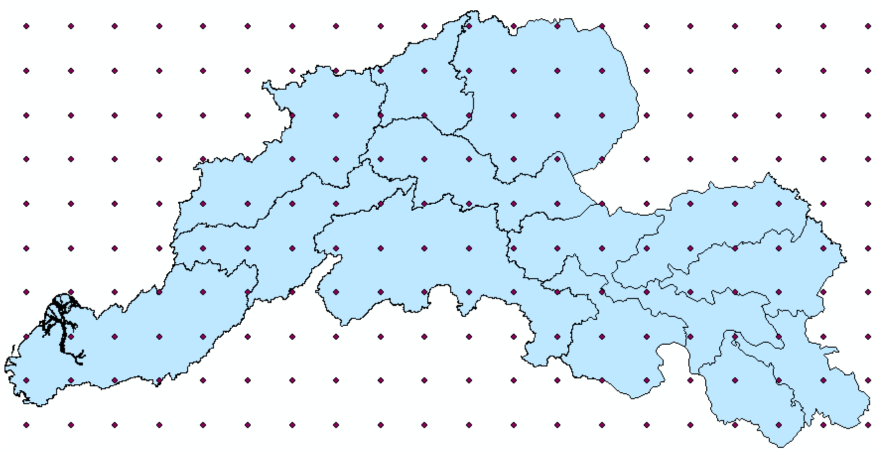


Generating Uniform Sample Locations

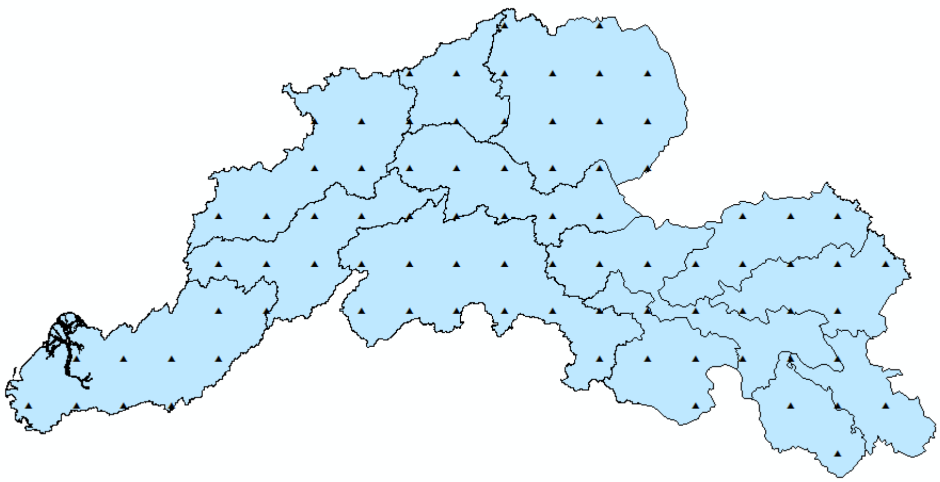
Add the Yukon Basin polygons to a new data frame:



We want a sample point located every 100km within the Yukon River Basin. First use the ***Create Fishnet*** geoprocessing tool to create points spaced every 100km.



Next use the ***Clip*** geoprocessing tool to create a point feature class of points inside the Yukon Basin.



Next, we want to randomly select ten of the points in the Yukon Basin.
Add a short integer field named PointID and compute PointID from the ObjectID field.

YukonBasin_100kmPoints

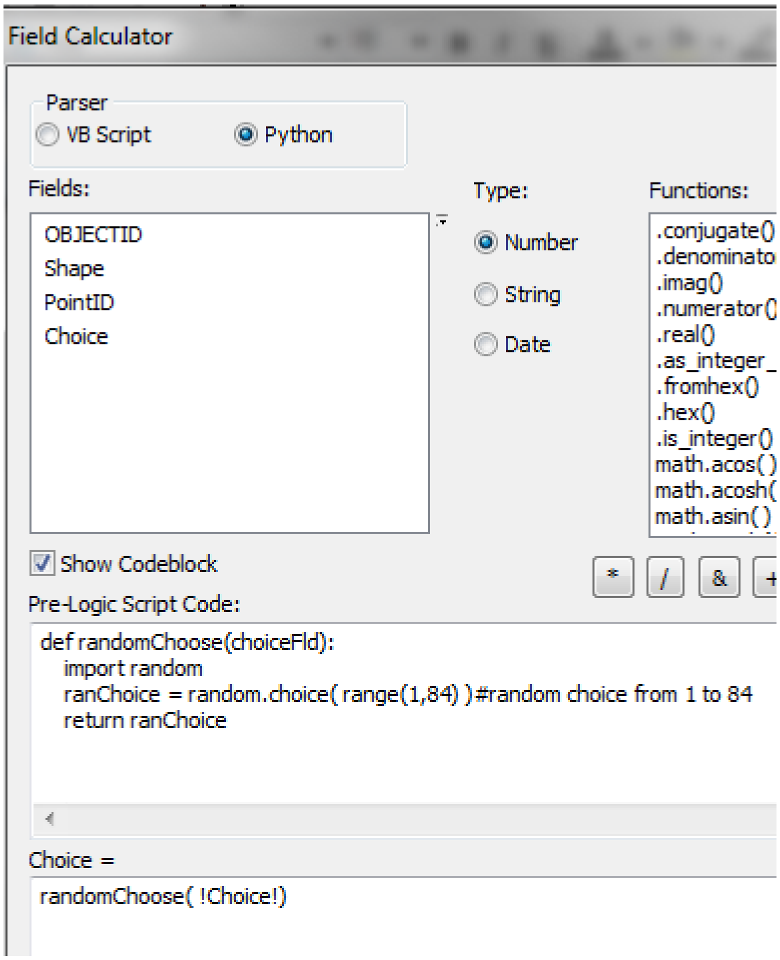
OBJECTID *	Shape *	PointID
1	Point	1
2	Point	2
3	Point	3
4	Point	4

Add another short integer field named choice and select the first ten rows.

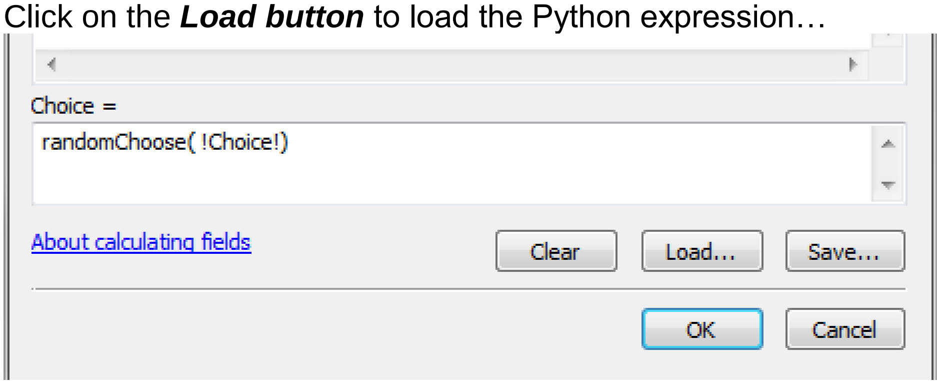
YukonBasin_100kmPoints

OBJECTID *	Shape *	PointID	Choice
1	Point	1	<Null>
2	Point	2	<Null>
3	Point	3	<Null>
4	Point	4	<Null>
5	Point	5	<Null>
6	Point	6	<Null>
7	Point	7	<Null>
8	Point	8	<Null>
9	Point	9	<Null>
10	Point	10	<Null>
11	Point	11	<Null>
12	Point	12	<Null>
13	Point	13	<Null>
14	Point	14	<Null>
15	Point	15	<Null>

There are 83 points to choose from. Use the Python field calculation to choose for each of the ***ten selected rows.***



Click on the ***Load button*** to load the Python expression…



FID *	Shape *	Id	Choice
1	Point	1	49
2	Point	2	9
3	Point	3	38
4	Point	4	59
5	Point	5	61
6	Point	6	50
7	Point	7	60
8	Point	8	83
9	Point	9	55
10	Point	10	73
11	Point	11	<Null>
12	Point	12	<Null>

Your choices will be different since these are random choices….

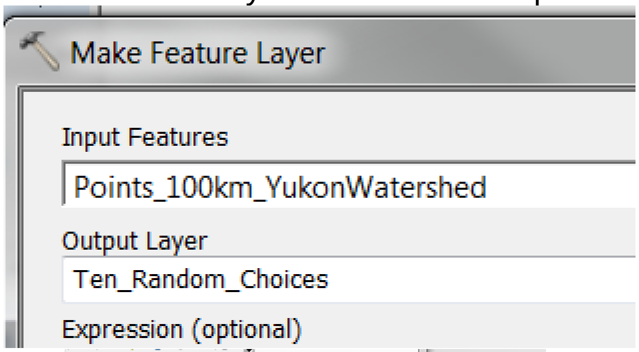
Sort your ten choices in your table

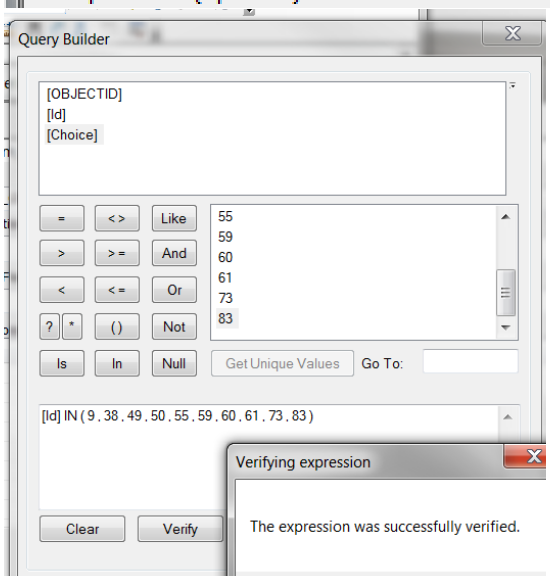
Points_100km_YukonWatershed

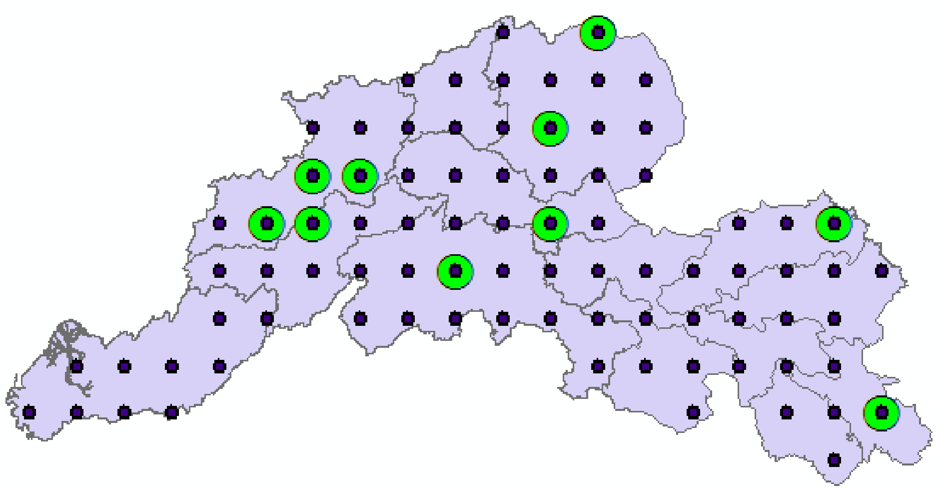
FID *	Shape *	Id	Choice
2	Point	2	9
3	Point	3	38
1	Point	1	49
6	Point	6	50
9	Point	9	55
4	Point	4	59
7	Point	7	60
5	Point	5	61
10	Point	10	73
8	Point	8	83

Then clear selection so all 83 points are available to query.

Make Feature Layer to show the ten points that were randomly chosen:







Ten_Random_Choice

Shape *	Id
Point	9
Point	38
Point	49
Point	50
Point	55
Point	59
Point	60
Point	61
Point	73
Point	83

Stratified Random Sampling

We want more random locations in larger basin polygons…
Compute the area of each basin polygon in square kilometers.

Yukon_Basin_Polygons

Shape *	SUBBASIN	KM2
Polygon	Chandalar River	35,513.6
Polygon	East Central Yukon	62,733.5
Polygon	Koyukuk River	81,326.2
Polygon	Lower Yukon	97,094.5
Polygon	Pelly River	49,022.5
Polygon	Porcupine River	116,398.7
Polygon	Stewart River	52,135.0
Polygon	Tanana River	115,840.3
Polygon	Teslin River	33,969.6
Polygon	Upper Yukon	71,395.0
Polygon	West Central Yukon	62,146.9
Polygon	White River	47,024.5
Polygon	Yukon Headwaters	31,782.4

We need a random point for every 10,000 km2 in each polygon. Create a field name NPts and compute the correct number of points for each row.

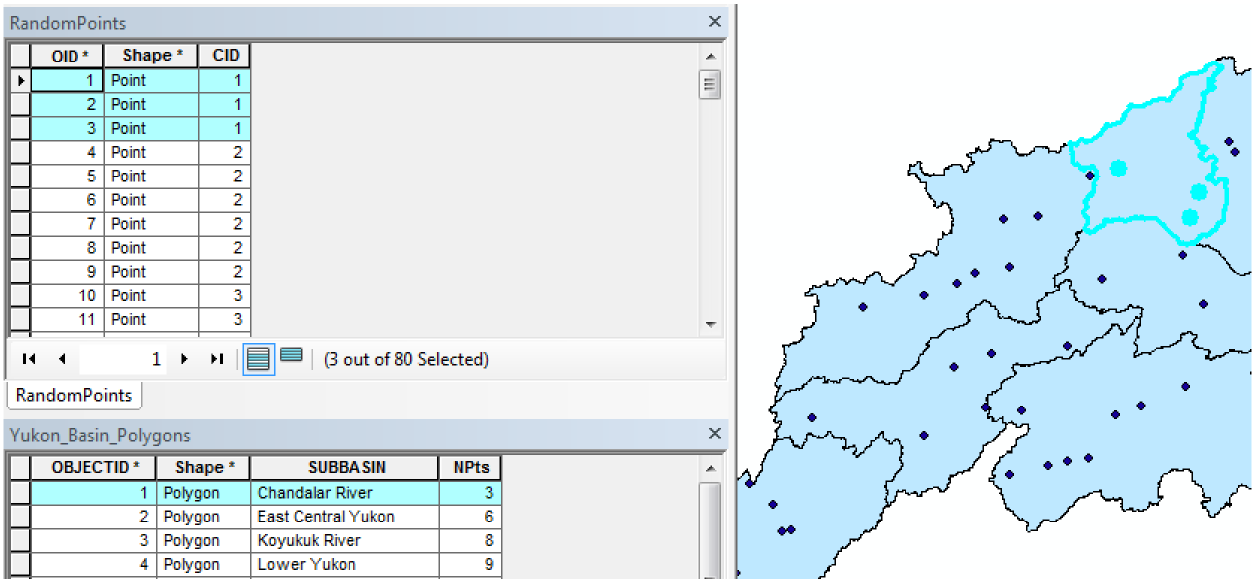
KM2	NPts
35,513.6	3
62,733.5	6
81,326.2	8
97,094.5	9
49,022.5	4
116,398.7	11
52,135.0	5
115,840.3	11
33,969.6	3
71,395.0	7
62,146.9	6
47,024.5	4
31,782.4	3

Show Codeblock

NPts =

```
Int ( [KM2]/10000 )
```

Next, use the **Create Random Points** geoprocessing tool to create random points in each basin polygon from your NPts field values.



RandomPoints

OID *	Shape *	CID
1	Point	1
2	Point	1
3	Point	1
4	Point	2
5	Point	2
6	Point	2
7	Point	2
8	Point	2
9	Point	2
10	Point	3
11	Point	3

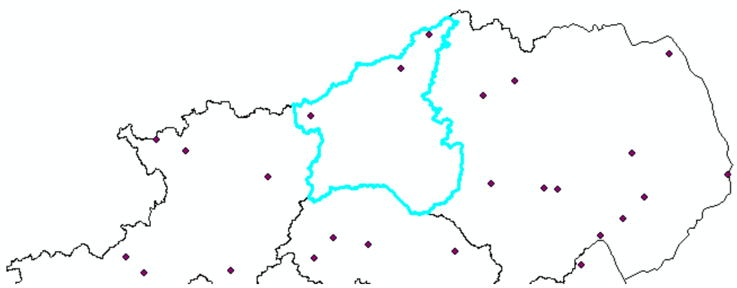
1 (3 out of 80 Selected)

RandomPoints

Yukon_Basin_Polygons

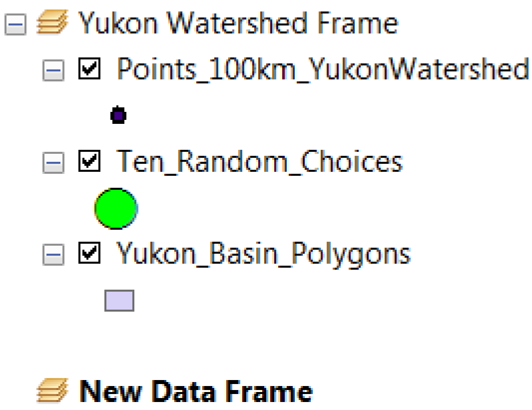
OBJECTID *	Shape *	SUBBASIN	NPts
1	Polygon	Chandalar River	3
2	Polygon	East Central Yukon	6
3	Polygon	Koyukuk River	8
4	Polygon	Lower Yukon	9

Your locations will differ since the points are randomly located..



Points Randomly Located Along Lines

From the Insert menu, select Data Frame to create a new data frame



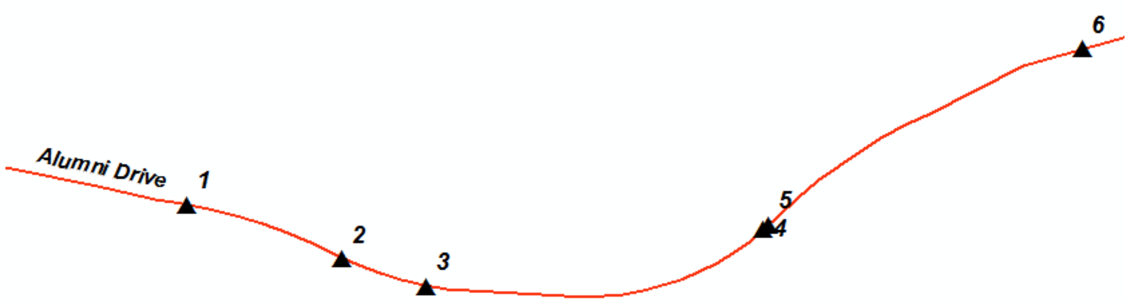
and add the UAF roads to your new data frame. Do a Definition Query to make a lines longer than 500 meters visible.

For these lines, we want to create a random point for each 100 meters of line. So create a field ***NPts*** and compute this field.

str_name	Shape_Length	NPts
Alumni Drive	601	6
Ambler Lane	192	1
Ballaine	864	8
Chatanika	121	1
CIGO	880	8
Columbia Circle	152	1

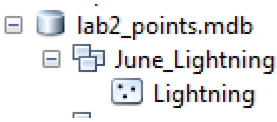
Next, use the ***Create Random Points*** geoprocessing tool to create random points on each line from your NPts values.

As an example, there were 6 points randomly located along Alumni Drive:



Point Count Density

Create a new data frame and add the lightning point feature class to your data frame.

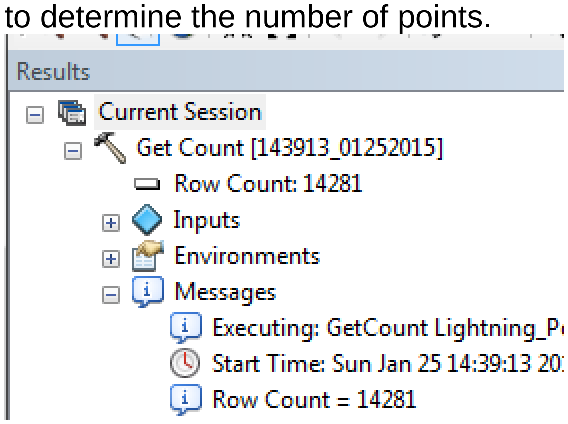


These points represent all the lightning strikes detected by the Alaska Fire Service lightning detection network in June 2014.

Lightning

OBJECTID *	Shape *	StrikeDay
1	Multipoint	6/13/2014
2	Multipoint	6/6/2014
3	Multipoint	6/6/2014
4	Multipoint	6/6/2014
5	Multipoint	6/6/2014
6	Multipoint	6/6/2014

A multipoint can be more than one strike having the same X,Y location and StrikeDay value. So first, use the ***Multipart to Singlepart*** geoprocessing tool to create a single point for each strike. Then run the ***Get Count*** geoprocessing tool to determine the number of points.



So there were a 14,281 strikes in June

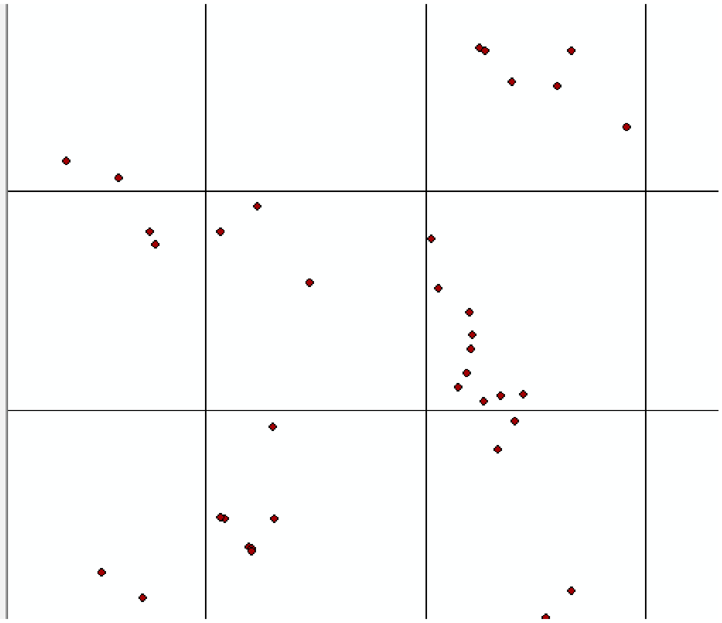
We want to create a map of lightning density per 100 sq km. You can do this using square polygons and determining the point count inside each square.

Use the ***Create Fishnet*** geoprocessing tool to create 100 sq. km. square polygons in the extent of your lightning strikes.

Add a double precision field and calculate geometry to check that your squares are 100 sq. km each.

Fishnet10km2

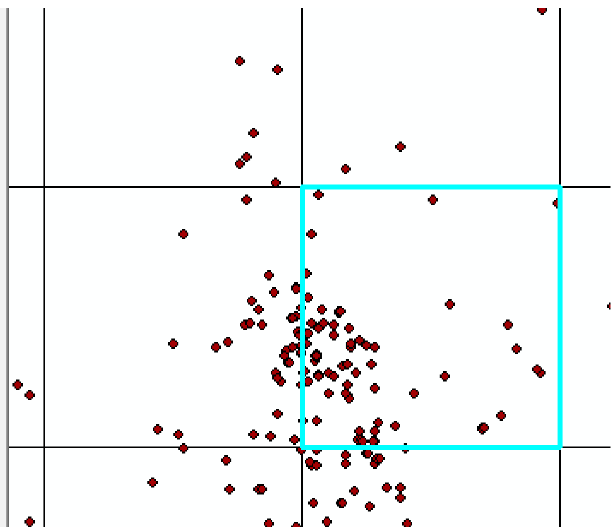
Shape *	SQ_KM
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100
Polygon	100



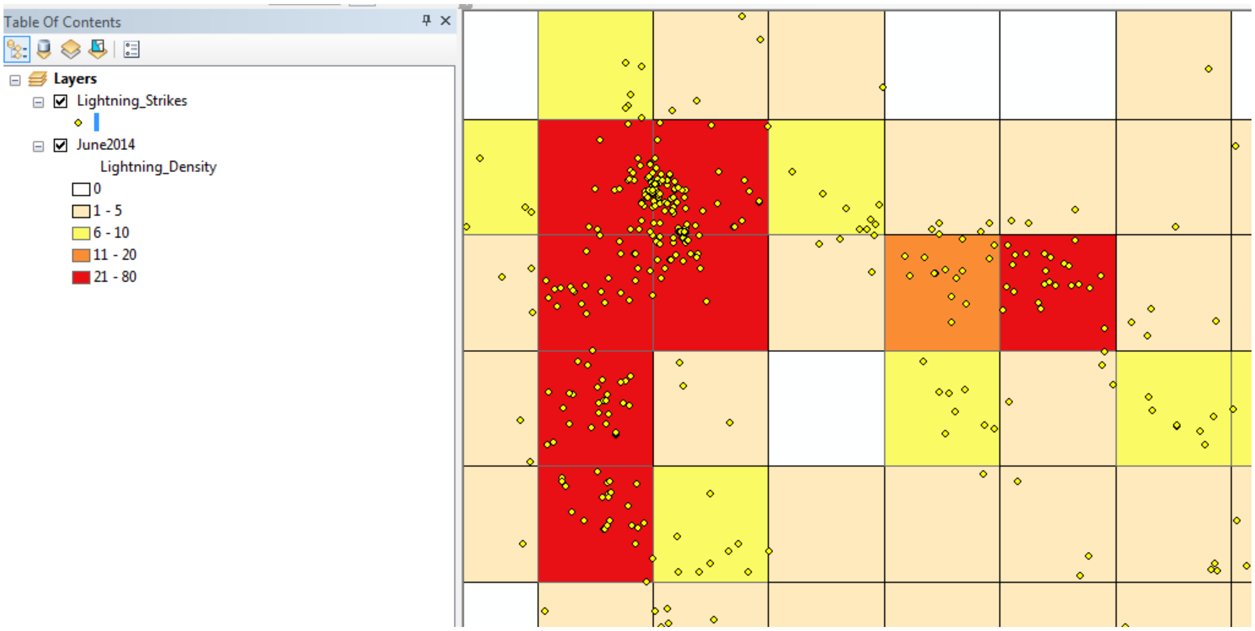
Next, use the ***Spatial Join*** geoprocessing tool to determine the number of lightning strikes inside each 100 sq. km. polygon.
Sort descending by Join_Count to find the square with the highest lightning density:

Fishnet10km2_StrikeCounts

Shape *	Join_Count	SQ_KM
Polygon	79	100
Polygon	63	100
Polygon	57	100
Polygon	53	100
Polygon	49	100
Polygon	48	100
Polygon	44	100
Polygon	37	100
Polygon	36	100
Polygon	36	100
Polygon	36	100
Polygon	36	100
Polygon	35	100
Polygon	35	100

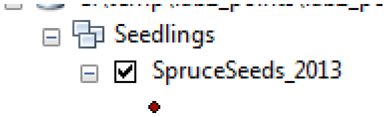


Symbolize your lightning density layer with the following 5 classes:

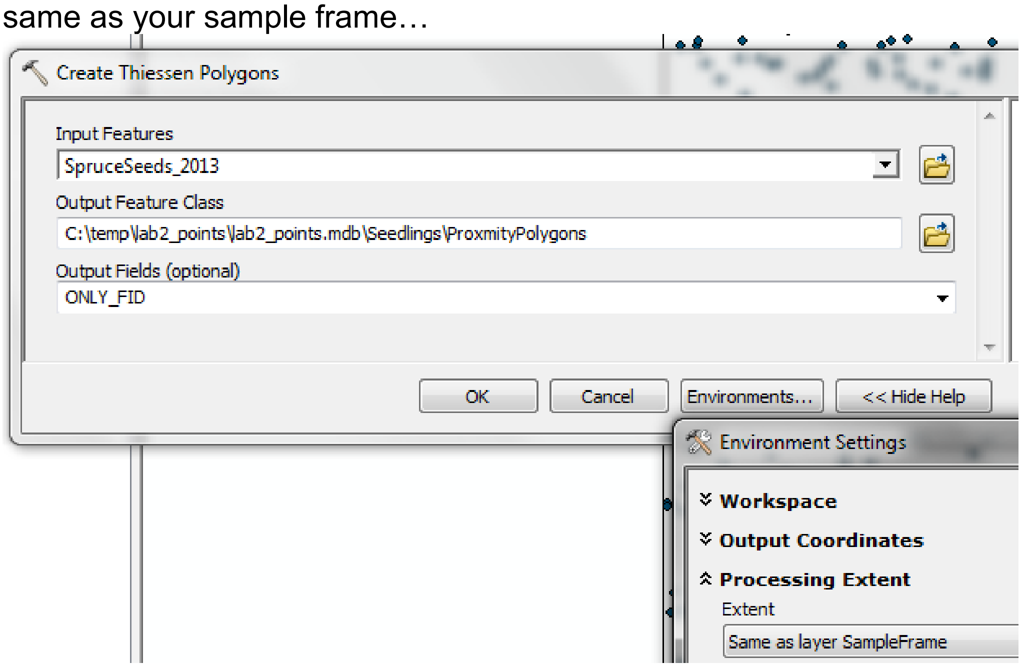


Density Effect

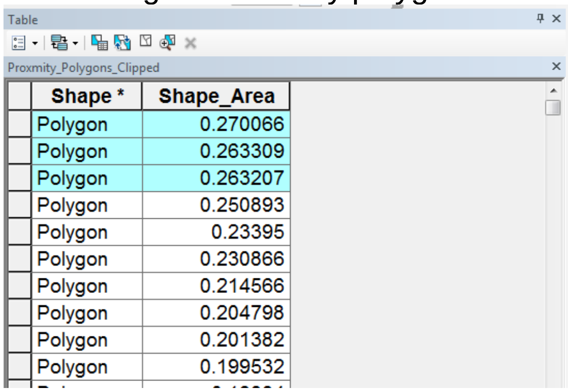
Create a new data frame and add the seedlings points to your new data frame.



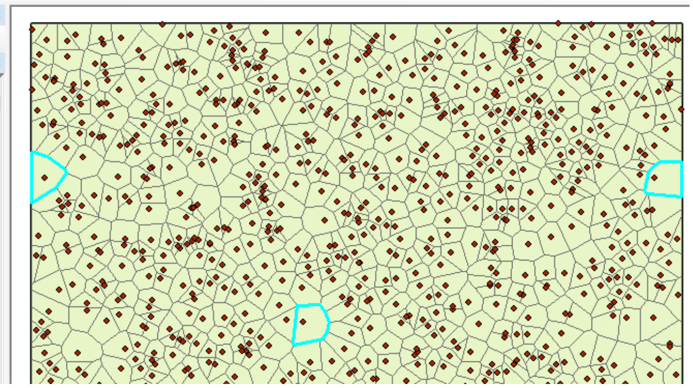
Use the **Create Thiessen Polygons** geoprocessing tool to create proximity polygons. Before you execute the tool, specify the processing extent to be the same as your sample frame…



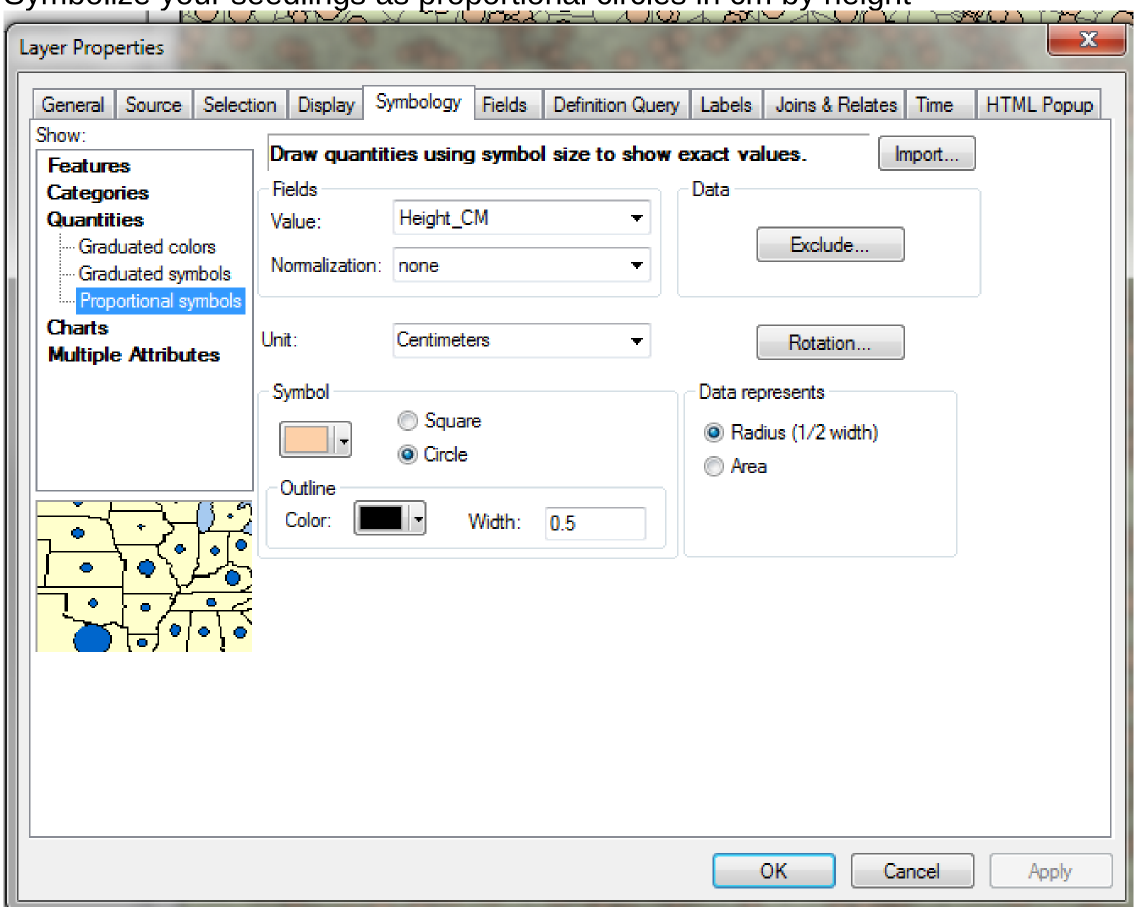
Three Largest Proximity polygons:



Shape *	Shape_Area
Polygon	0.270066
Polygon	0.263309
Polygon	0.263207
Polygon	0.250893
Polygon	0.23395
Polygon	0.230866
Polygon	0.214566
Polygon	0.204798
Polygon	0.201382
Polygon	0.199532



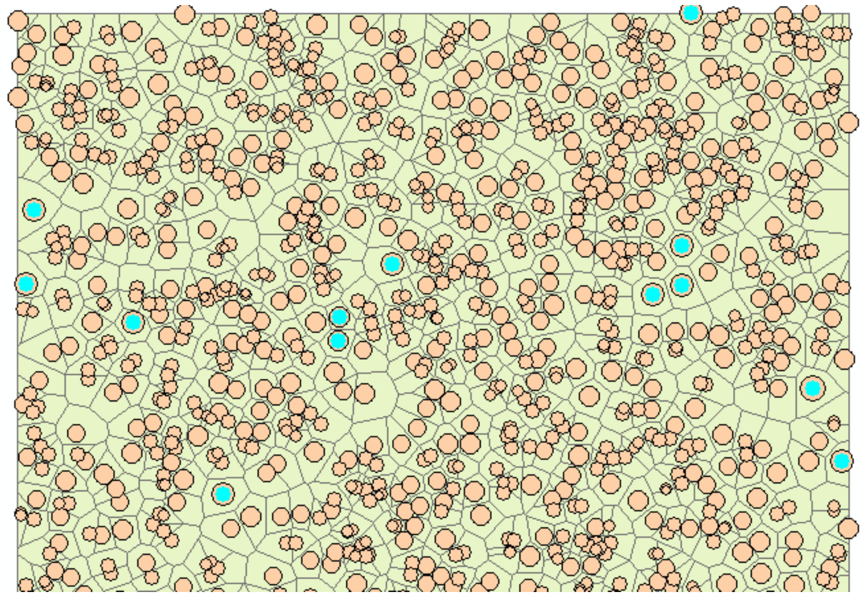
Symbolize your seedlings as proportional circles in cm by height



There seems to be a relationship of greater height growth at lower seedling density….

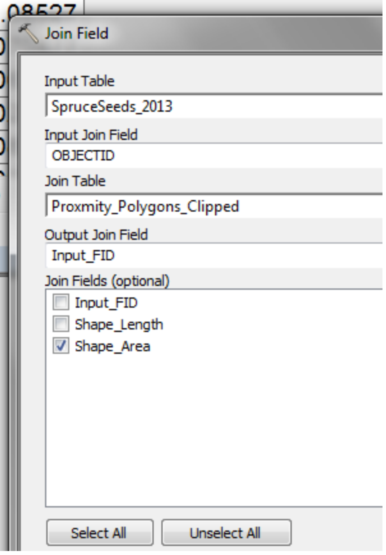
SpruceSeeds_2013

	Shape *	Height_CM
	Point	16.27505
	Point	14.727754
	Point	14.624436
	Point	14.233239
	Point	13.921007
	Point	13.748956
	Point	13.61879
	Point	13.421898
	Point	13.359847
	Point	13.273629
	Point	13.173228
	Point	13.169284
	Point	13.121603
	Point	13.041489
	Point	13.001646



We will test this relationship by developing a linear model Height = a + b(Density) where Density is the area of the proximity polygon for each seedling.

Use the ***Join Field*** geoprocessing tool to join the area of each proximity polygon to the seedling points.



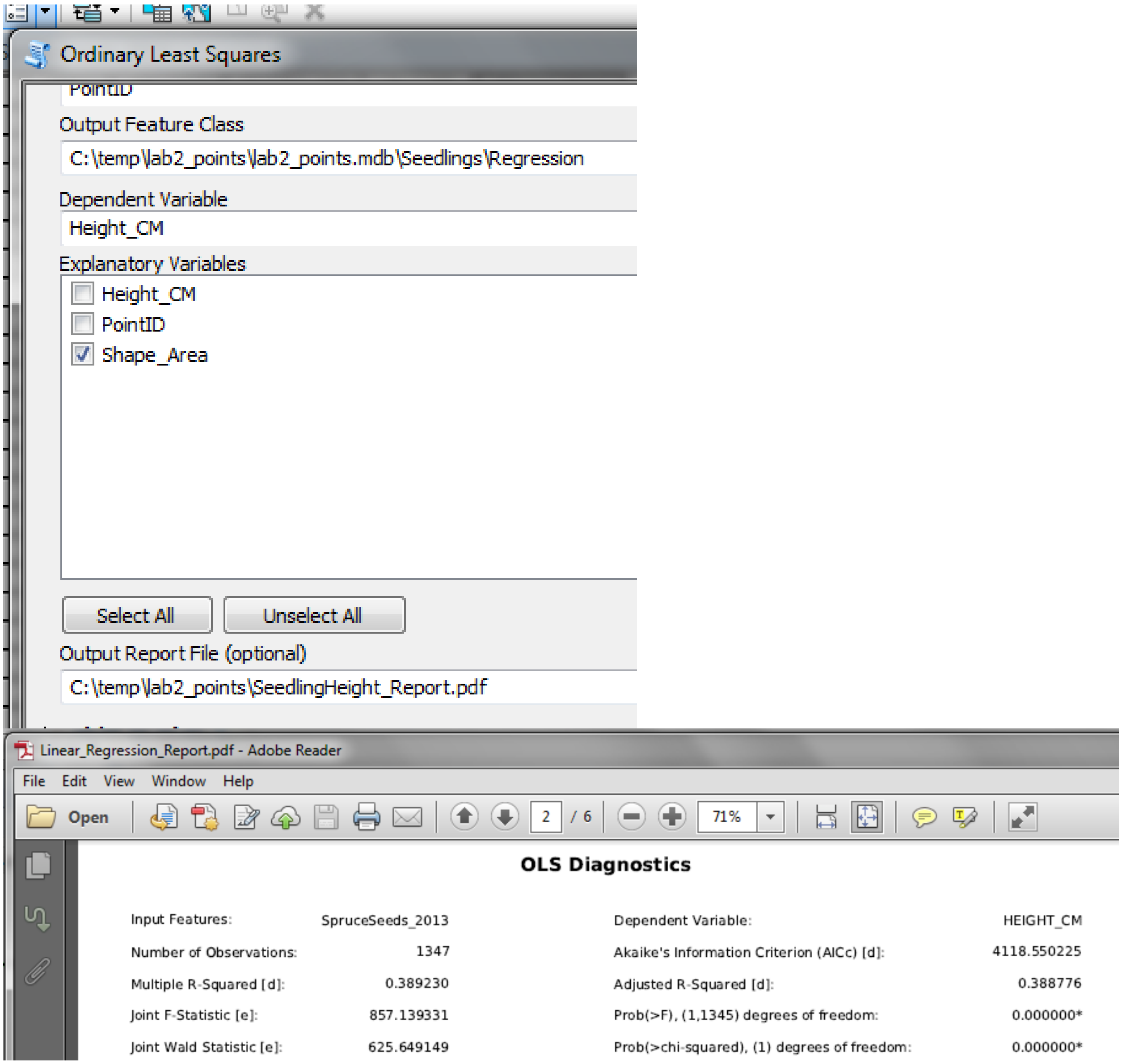
Proxmity_Polygons_Clipped

	Shap	Input_FID	Shape_Area
	Polyg	1	0.071539
	Polyg	2	0.052468
	Polyg	3	0.073451
	Polyg	4	0.08527
	Polyg	5	0.0
	Polyg	6	0.0
	Polyg	7	0.0
	Polyg	8	0.0

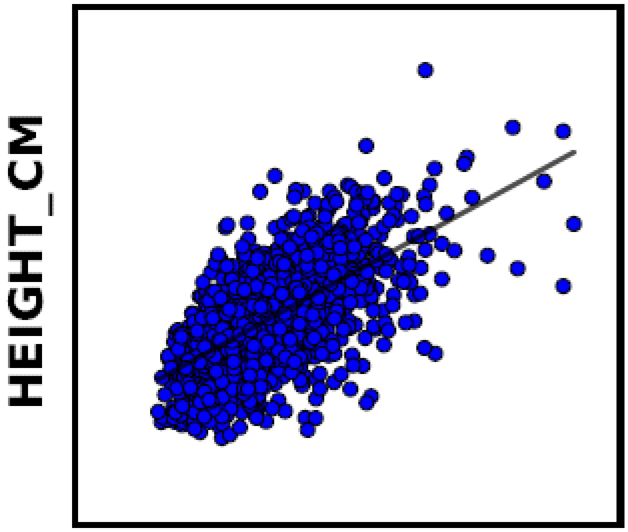
SpruceSeeds_2013

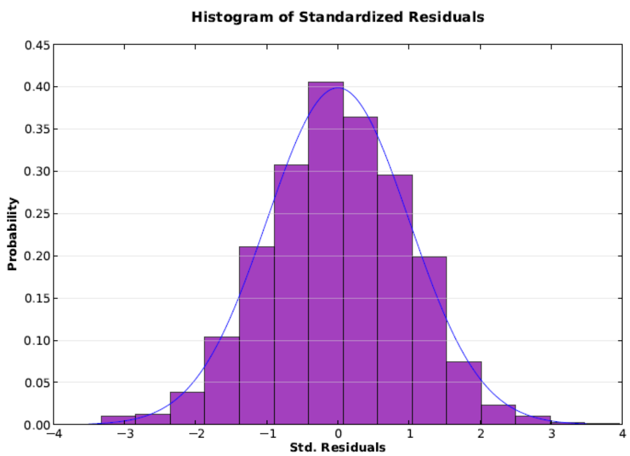
	OID	Shap	Height_	PointID
	1	Point	9.4064	1
	2	Point	9.9535	2
	3	Point	9.3492	3
	4	Point	9.1778	4
	5	Point	10.588	5
	6	Point	8.9479	6
	7	Point	7.7449	7

Then use the ***Ordinary Least Squares*** geoprocessing tool to create a linear model predicting seedling height as a function of the proximity polygon area…the larger the area, the greater the seedling height due to less seedling competition..



The output pdf has model information and plots…the model explains 38.9 percent of the variation in seedling height.



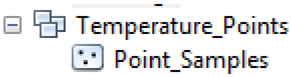


Ideally the histogram of your residuals would match the normal curve, indicated above in blue. If the histogram looks very different from the normal curve, you may have a biased model. If this bias is significant it will also be

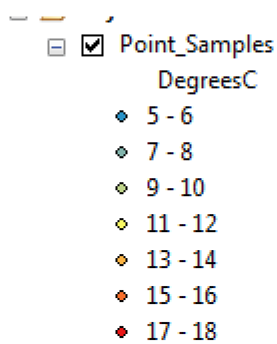
The model is not biased as the model errors peak at 0.

Point Values Interpolated To Surface

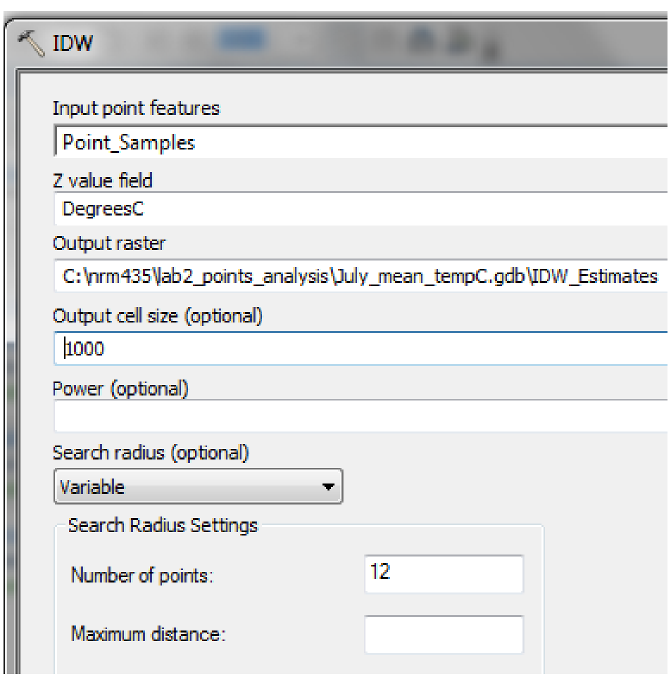
Create a new data frame and add Point_Samples to your data frame.

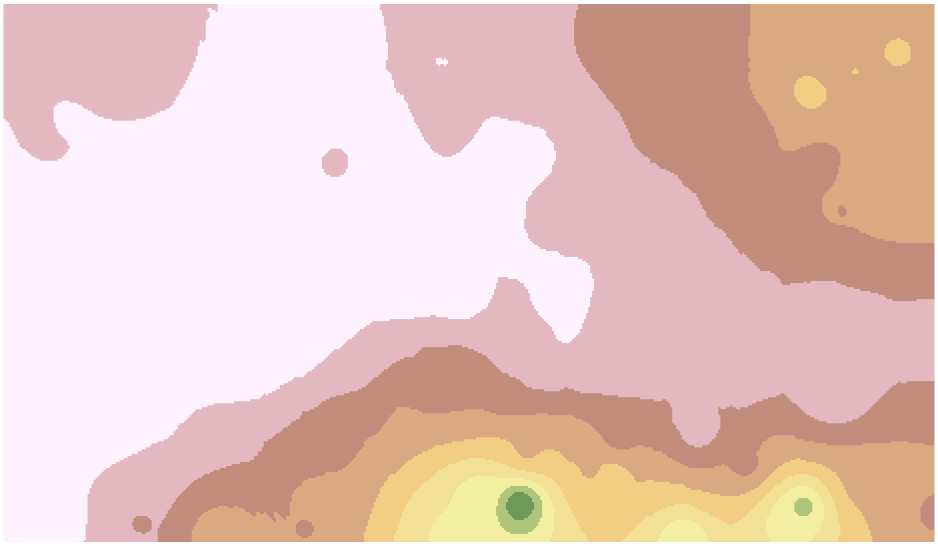


The points represent mean July temperature for some locations in interior Alaska. We will use 2 interpolation methods to create a raster of temperature estimates for this area. Both methods are "exact" in that the output surface will predict the correct temperature value at the point locations. Adjust your symbology of your points as follows:



The first method is ***Inverse Distance Weighted*** interpolation or IDW. The method assigns a greater weight to points that are closer to the output pixel for the 12 points closest to each pixel. We will take the default power value of 2 and output to a 1000 meter cell size raster.





How accurate are these estimates? Use the ***Extract Values To Points*** geoprocessing tool to extract the pixel value estimates to your Points_Validation

- Temperature_Points
  - Point_Samples
  - Points_Validation

Extract_IDW

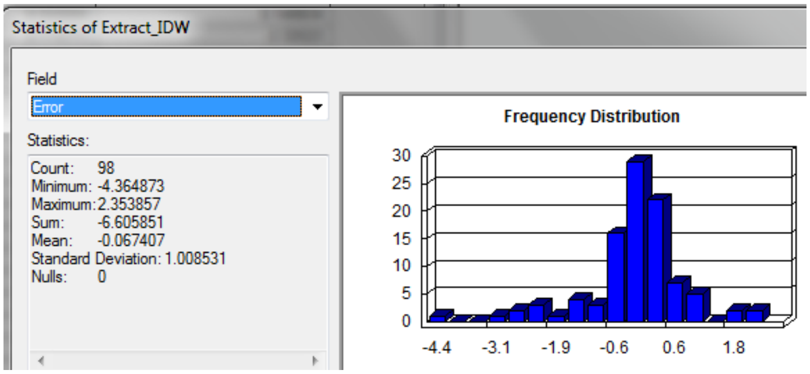
Shape *	DegreesC	RASTERVALU
Point	14.7	13.37663
Point	15.9	14.908942
Point	15.7	15.229673
Point	17.799999	17.470284
Point	15.4	15.608465

There are 2 points that are outside the extent of the IDW raster, and they have a raster value of -9999. Use a Definition Query to exclude these 2 points.

Extract_IDW

Shape *	DegreesC	RASTERVALU
Point	13.4	-9999
Point	16	-9999
Point	5.3	5.672365
Point	9.7	9.550066
Point	12.1	9.774771
Point	9.8	10.408037

Add a double precision field named Error and use the field calculator to compute error as the actual temperature minus the estimated (IDW interpolated) temperature.

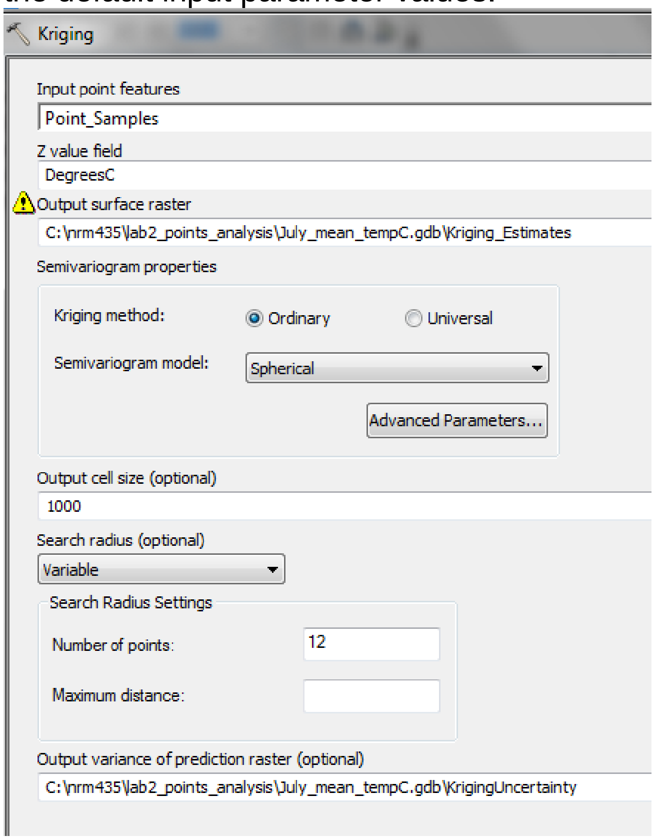

Statistics of Extract_IDW
Field
Error
Statistics:
Count: 98
Minimum: -4.364873
Maximum: 2.353857
Sum: -6.605851
Mean: -0.067407
Standard Deviation: 1.008531
Nulls: 0
Frequency Distribution
30
25
20
15
10
5
0
-4.4
-3.1
-1.9
-0.6
0.6
1.8

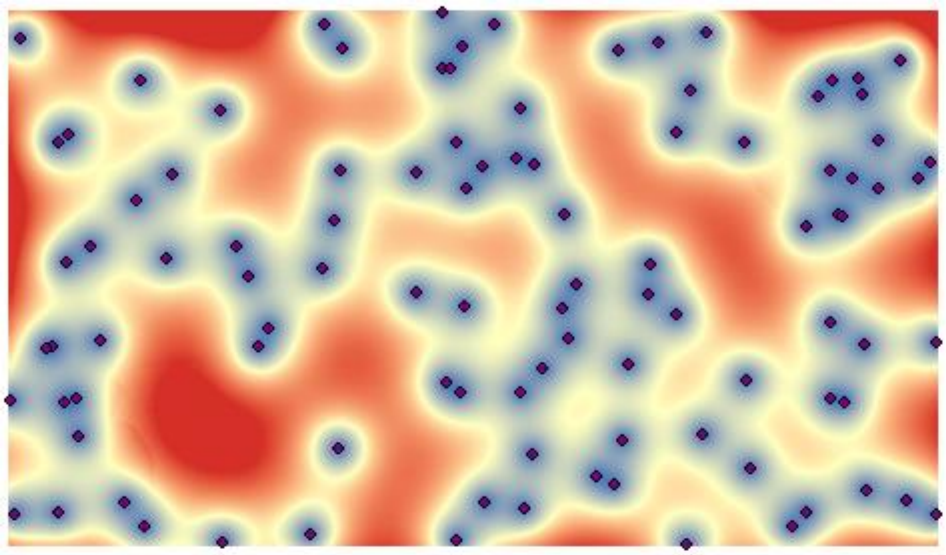
Kriging is a geostatistical method based on statistical models that include autocorrelation—that is, the statistical relationships among the measured points. Kriging has 2 advantages over IDW interpolation:

1) Because it models spatial autocorrelation, it typically produces more accurate estimates compared to IDW interpolation.

2) Sometimes we do not have validation points. With Kriging, a prediction raster is also output, allowing the user to assess the likely uncertainty of the estimates. Typically estimates will have higher uncertainty in neighborhoods where there are few points close together and the point values are highly variable.

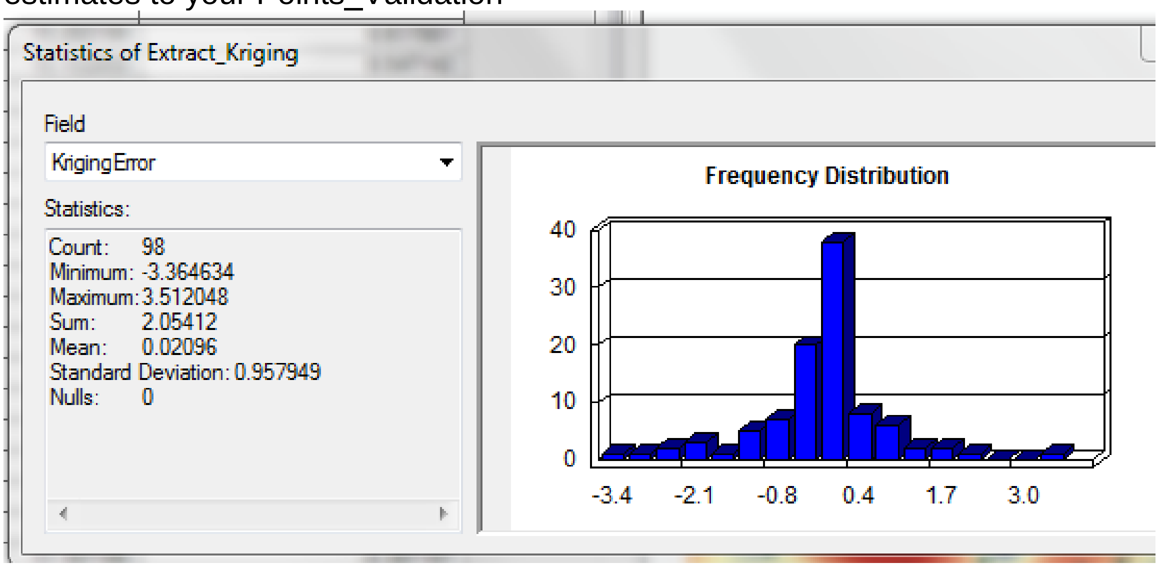
Run the ***Kriging*** geoprocessing tool, outputting to 1000 meter pixels and using the default input parameter values.

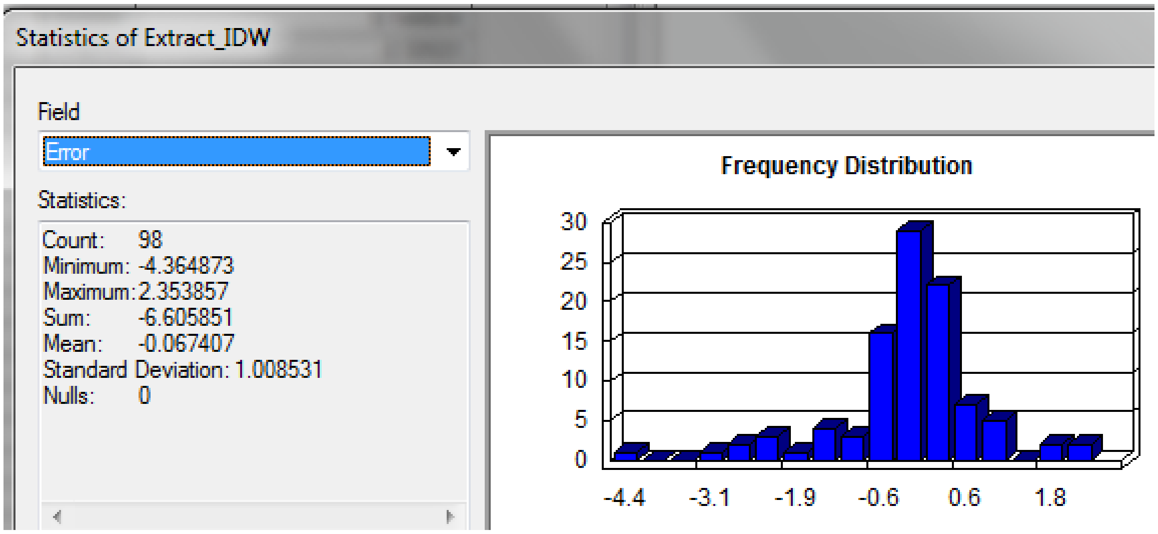




Notice from the uncertainty raster the areas with highest uncertainty are areas where there were no input points and areas of low uncertainty had many input points.

Use your validation points to assess the accuracy of your Kriging estimates. Use the ***Extract Values To Points*** geoprocessing tool to extract the pixel value estimates to your Points_Validation





So the mean error from Kriging was 0.020 degrees compared to the mean error from IDW of -0.067 degrees.