Appendix B- Technical Walkthroughs

B1- ArcGIS Pro: Set coordinate systems, upload KML, MBG, find center of polygons

Last updated May 16, 2023

Basics

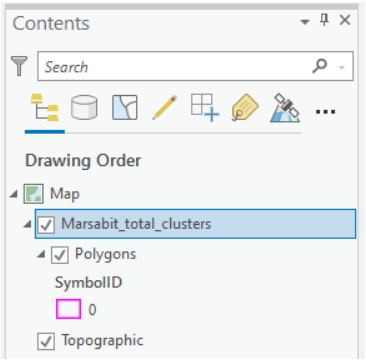
- It's helpful to make sure your feature names are saved sequentially in GoogleEarth- they read everything as alpha characters, not numbers, so you might want to save the feature names with leading zeros (A001, not A1). This will make it easier to match up output from ArcGIS to your master excel file
- Remember that Arc does not like spaces or weird characters in file or folder names
- All excel files should be .xls
- It's easiest to have all your files (KML, excel, Arc project) saved in one folder on the C: drive, but that means that you will have to use the same computer if you need to open them again, so keep that in mind
- Make sure your KML file is only the feature polygons- make sure the area boundaries, paths, points, etc are removed.
- The ArcGIS Pro resources are detailed, technical, and very helpful. If you have any questions, consult them first!
- Anything that is really important to get right will be in red

Map Coordinate System

- Open a new project, save it in your project folder in the C: drive
- Right click on *Map* in the *Contents* (leftmost pane) and select *Properties*. Navigate to *Coordinate Systems*
- Find your UTM zone and hemisphere under *Projected Coordinate Systems* and click *OK*
- The map will transform into an orange slice shape, with your study area in the center

Import the KML file

- In the *Geoprocessing* plane (look at the tabs on the bottom of the rightmost pane), search for KLM. *KML to Layer* should be the first tool.
    - Enter info. Output location should be your handy folder in the C: drive, if that's where you specified where your project should be saved
    - Don't worry too much about the output data name right now
- You should see the file in your *Contents* (leftmost pane). Open the descending files until you see *Polygons*

- 

- Right click on *Polygons*, select *Data*, then *Export Features*.
- This is where you want to give your features a good name. After entering the *Output Name*, select the *Environments* tab.
- Click *Output Coordinate System*. If you select *Current Map [Map]*, it should auto populate the UTM zone
- Ta da! Your polygons are now a shapefile and are projected to the correct coordinate system.

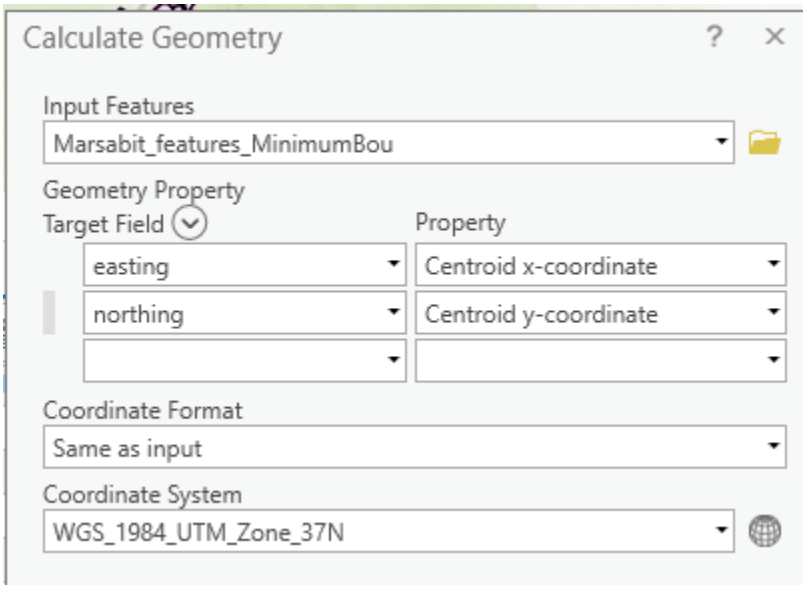
Minimum Bounding Geometry

- Search for *Minimum Bounding Geometry* in the *Geoprocessing* pane (look for the tabs on the bottom of the rightmost pane)
- Enter your newly created shape files as the *Input Features*
- Rename the *Output Feature Class* something memorable
    - *Geometry Type = Rectangle by area*
    - *Group Option = none*
    - Make sure *Add geometry characteristics as attribute to output* is selected
- Enveloping Rectangles are placed over your features. You'll see the new file in the *contents* pane
    - Optional- Click the symbol below it in the *contents pane* to change the symbology. I like having them as outlines, nothing filled in, so you can see the features under them
- Right click the new feature in *Contents* and open the attribute table
- Ta da! You now have the minimum bounding rectangle's long axis and short axis in meters, and the angle of the long axis.
    - Before exporting this info into an Excel sheet, there's one more tool to run

Find the center of the features

- When mapping, we record the coordinates of what we think is the center of the polygon into the master excel file. While this is very helpful when reviewing features, it's not precise enough for geospatial analysis. This tool will compute the exact center of each polygon
- If it's not already open, right click the MBG feature in the contents pane and open the attribute table
    - If you want, you can hide/delete some columns to make the table easier to read. The *OBJECTID*, *Name,* and MBG data should be kept
- Select *Add Field*
- A new tab will open where you can add new fields to the attribute table. On the bottom is the new field, highlighted in blue
- Add two new fields

    - *Field Name* and *Alias* = *Easting* and *Northing*
    - *Data Type = Double*
    - *Number Format = Numeric (*two decimal places)
- In the menu ribbon in the top of the program, select the *Fields* tab, then *Save.*
- Close the tab where you entered the new field. The attribute table for the MBG feature should still be open, with two new columns called *Easting* and *Northing*. They are filled with *<Null>* values.
- Right click the *Easting* heading and select *Calculate Geometry.* Fill out the box like this:



- *Ok*
- The table now has the easting and northings of the center points of each polygon.

Export the data into an Excel Sheet

- In the *Geoprocessing* pane, search for *Table to Excel*
    - For input table, select your new MBG feature
    - Give your output table a memorable name, make sure it will save in your folder in the C: drive
- Ta da! You now have an excel file with the minimum bounding rectangle's long axis and short axis in meters, the angle of the long axis, and the coordinates of the center of each feature. You can copy/paste this info into your master file
    - In the Excel file, *Name* will equal your feature name from Google Earth. This is where it's useful to have everything in sequential order when you upload your KML, so you don't have to hunt and seek and make sure the data are being copy/pasted to the right feature

Upload polygon centers as points

- In your home folder, create a .xls file with the easting and northing of your center points. You can just copy/paste the coordinates from the excel table created in the last step. There should be no other data besides the easting and northing (table headers are ok, and will help you from getting them mixed up)

    - If you were clever enough to note the elevations, enter that in the third column.
- In the *geoprocessing* pane, search for *Excel to Table* and import the table into ArcGIS.
- The newly imported table will appear in the *Content* pane under *Standalone Tables*
- Right click the new table and select *Display XY data.*
    - *X Field = Easting*
    - *Y Field = Northing*
    - *Z Field = Elevation* (if you have it)
    - Give the output a memorable name
- Click the environments tab and choose the UTM zone for the coordinate system
- The map should populate with points in the center of your polygons. Choose a symbol less annoying than the blue dropped pin

B2- MATLAB Dendrogram and NNd clusters

Last updated May 16 2023

Anything in *italics* is code

Anything in red is a variable

Remember that everything in MATLAB is case sensitive and spaces around = are important

1. Define the array as X.
2. X = [x y z;] X = [easting northing elevation;] of polygon centers as defined in ArcGIS

X = [copy/paste from excel;]

3. Create the dendrogram. It *should* pop up. If it doesn't, highlight the two lines, right click, and select Evaluate Selection

treefigure = linkage (X,'single');

dendrogram(treefigure)

4. Evaluate the dendrogram and decide on the NNd and number of clusters
5. Define the cluster group as C. The last input is the # of clusters.

C = clusterdata(X,'Linkage', 'ward', 'SaveMemory', 'on', 'Maxclust',10);

6. Plot the clusters. This step is not necessary but it makes a pretty 3D plot

scatter3(X(:,1),X(:,2),X(:,3),10,C)

7. Create a matrix, so you can see what features fall into what cluster

Aug= [X C]

8. The coordinates will be displayed in scientific notation without enough sig figs because MATLAB is helpful like that.
9. These lines will display Aug with enough sig figs to decipher the coordinates

format long

Aug

10. They still won't be in the right UTM format, but you can copy/paste into Excel and multiply by 100,000. The last column is the cluster #
11. Now you have a table with coordinates and what cluster they are in. Yay! They will be in the same order you copied them in from Excel, but spot check a few to be sure.
12. Repeat the process for a different cluster group with a different number of clusters, starting with step 5, using a new variable for the new cluster group

B3- MATLAB 3-point Alignment Code

Last updated: May 16 2023

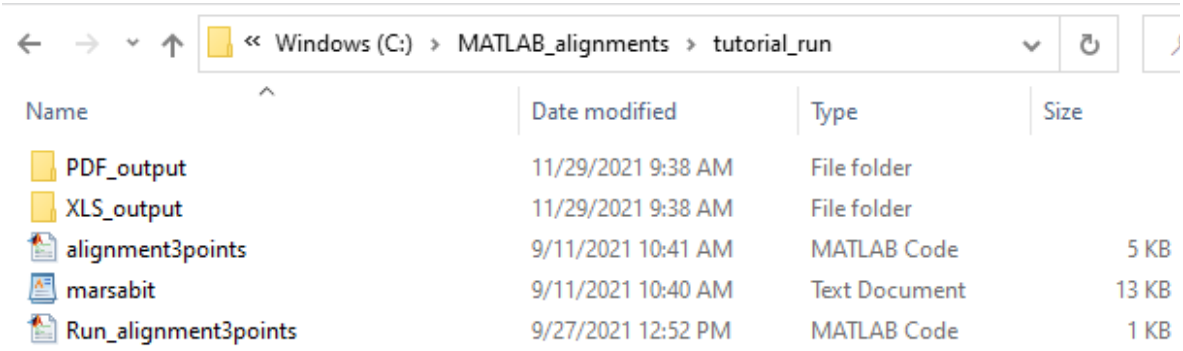
Code by Nicolas Le Corvec

Based on tutorial by James Muirhead

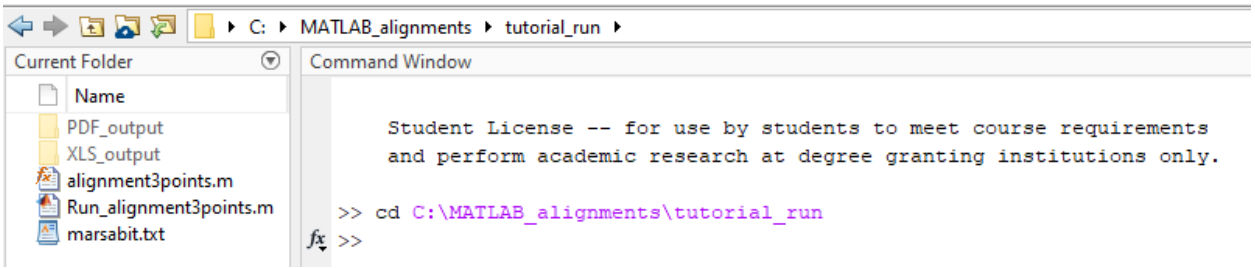
Files needed:

- alignment3points.m
- Run_alignment3points.m
- Tab delimited .txt with UTM coordinates of feature central points

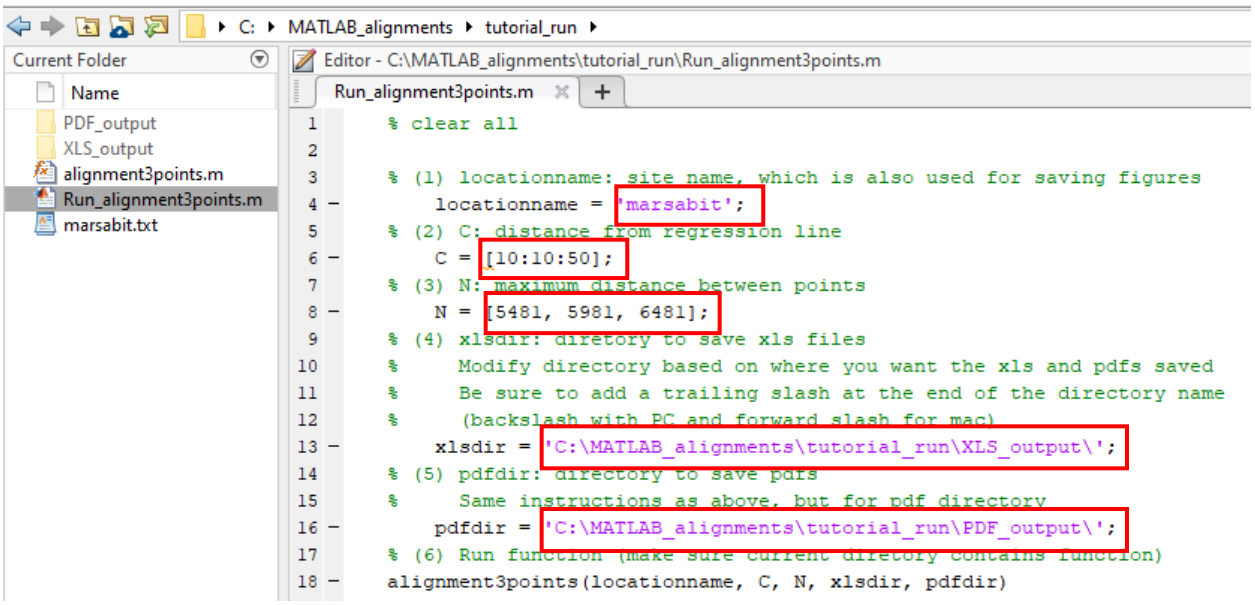
This script will identify three points that form an alignment within specific width (C) and length (N) tolerances. Start by getting all your files in one folder. Also create a folder for PDF and Excel outputs.



Use the **cd** command in MATLAB to change directories.



Open the Run_alignment3points.m code



- Locationname =
  - .txt file with feature center points
- C =
  - Width tolerance. The number in the center is the interval- 10:10:50 means it will run starting with a width tolerance of 10 meters and will increase by 10 meters until 50 meters- i.e. 10, 20, 30, 40, 50 meters.
- N =
  - Length tolerance. In this example, I used commas instead of colons- this means it will only run for 5481, 5981, and 6481 meters. If you want, you could use the layout as above.
- xlsdir =
  - Path to the Excel output folder. Do not forget the backslash at the end!
- pdfdir =
  - Path to the PDF output folder. Do not forget the backslash at the end!

Run the code. In the command window, you will see Run_alignment3points and “Busy” in the very lower left-hand corner. In a few moments, a figure will pop up with the first C and N combination. Depending on how many width and length tolerances are set, it may take a few minutes for the entirely of the code to run. When the “Busy” disappears, the code has finished running.

Tada! You now have Excel sheets with the orientations of 3-point alignments that fall in the specific width and length tolerances and PDFs of the alignments. Add a column of zeros and save as a tab delimited .txt to create stereonets with Stereonet 11.

B4- MATLAB KDE

Last updated: May 16, 2023

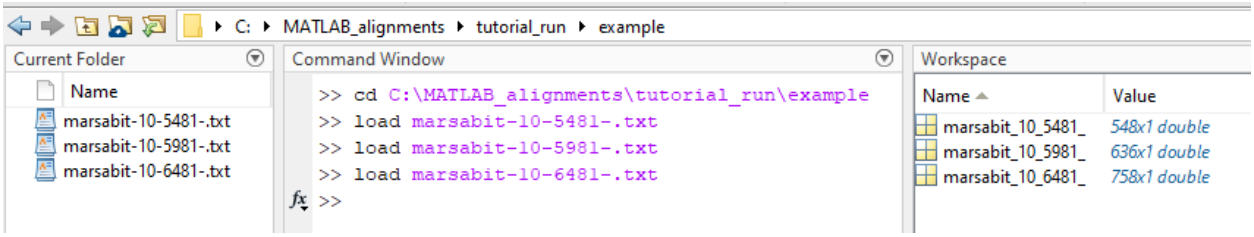
Based on a tutorial by James Muirhead.

Best to start with this Intro to Kernel Density Estimation video. I also found these articles helpful:

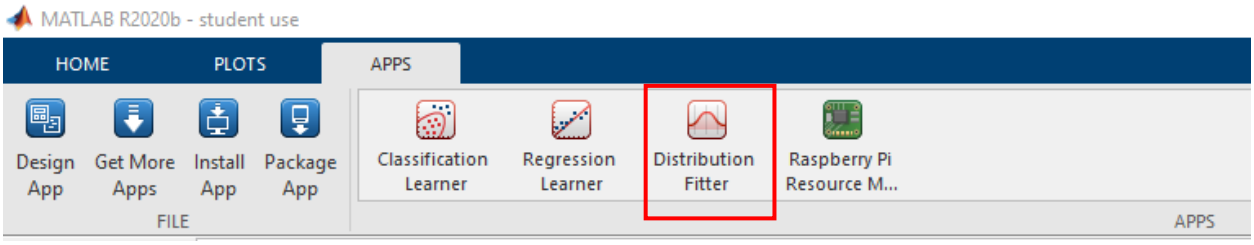
- Kernel Density Estimation
- Estimating Data Clusters with Kernel Density Estimation

This process uses the Excel files of linear array orientations created by Le Corvec's 3-point alignment code. Save them as a text (tab delimited) file. Once you are familiar with the process, I found it easiest to process in batches by width tolerances.

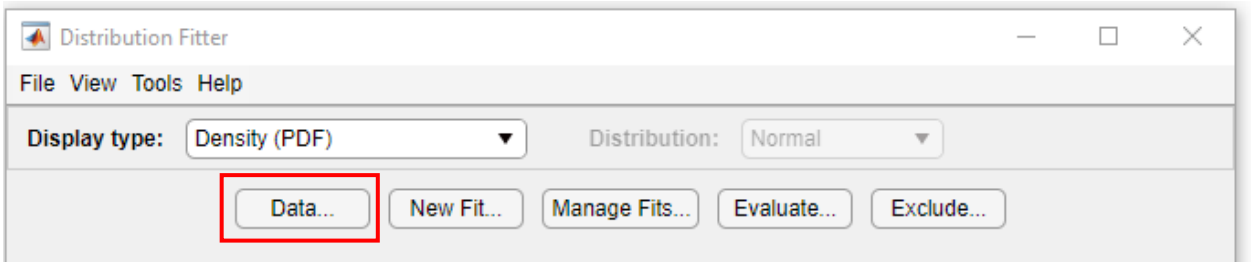
- Use the **cd** command to change directory to your working folder.
- Use the **load** command to load the first .txt
- After loading the text files, they will populate the Workspace



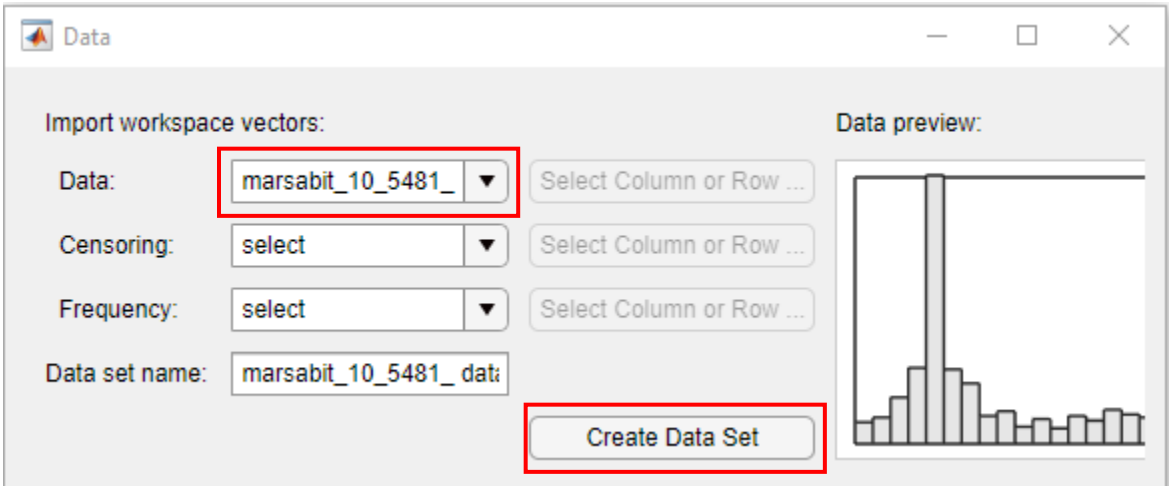
- Navigate to APPS > APPS > Distribution Fitter and open the app. You may need to install it.



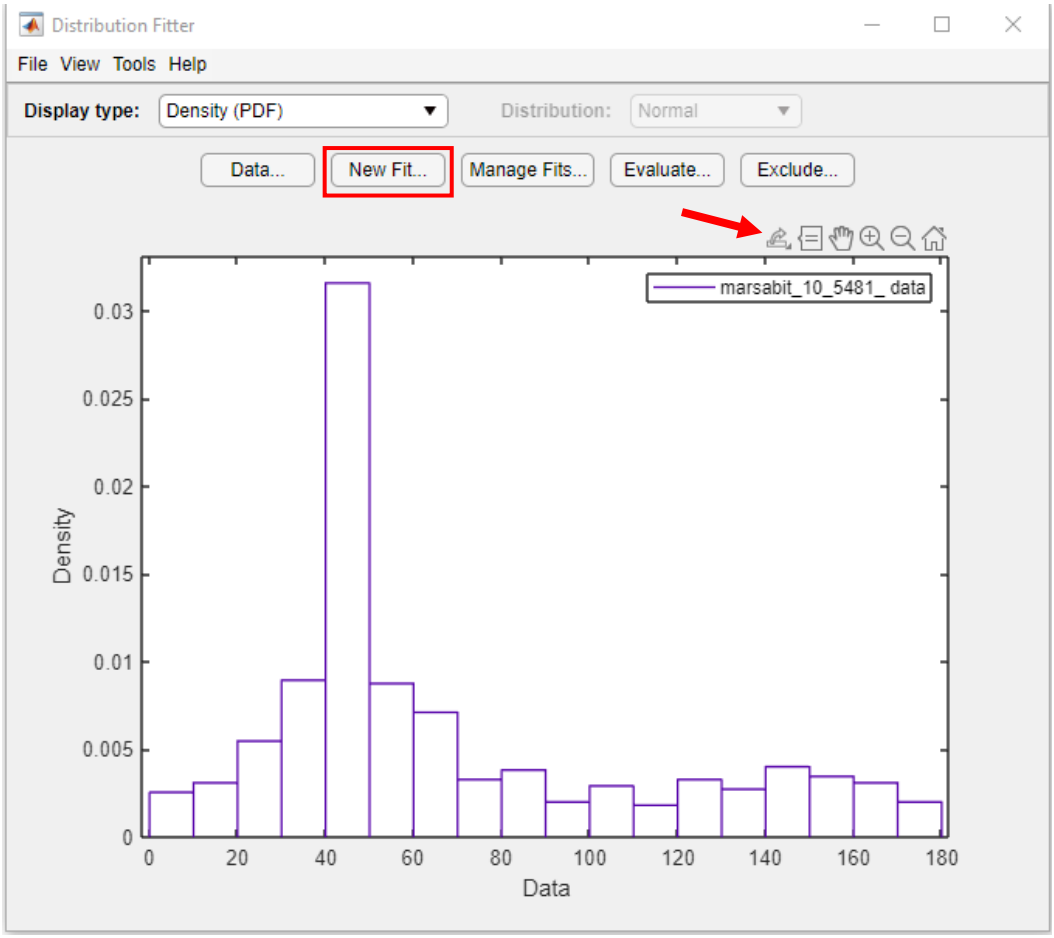
- Select *Data*.



- Select the first text file and *Create Data Set.*

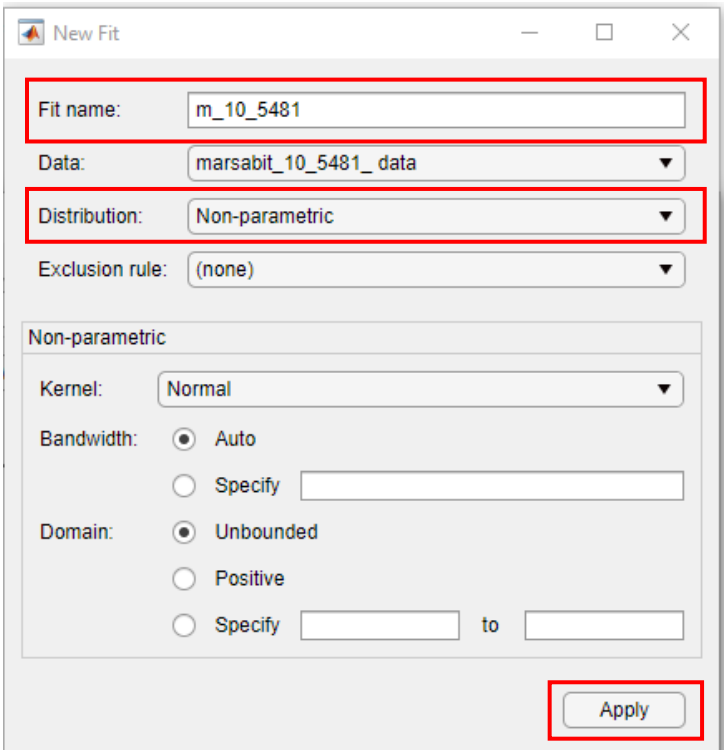


- Back in the *Distribution Fitter* window, a histogram of the data will appear. Hovering over the histogram will make a menu appear in the upper right. Clicking the first icon will allow you to export a PDF.
- Select *New Fit.*

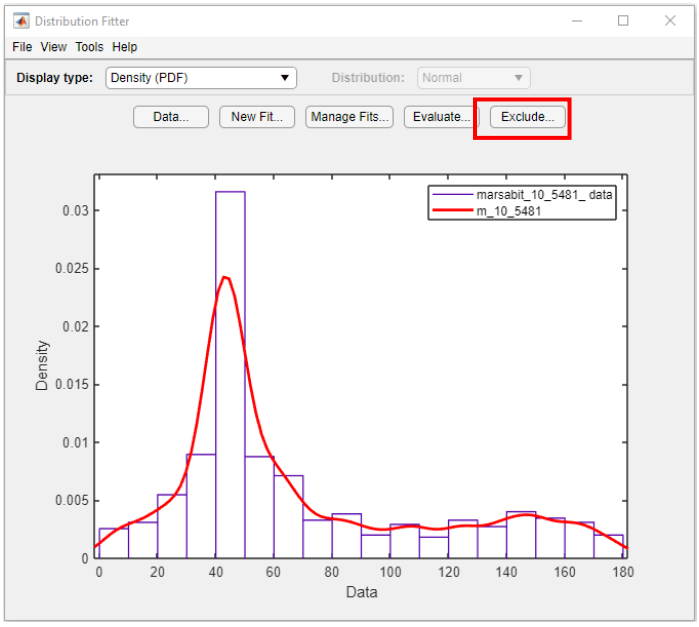


- This is where the Kernel Density Estimate fit is applied.
- Give the fit a descriptive name.

- Select *Non-parametric* as the distribution.
- Select *Apply*, then *Close.*

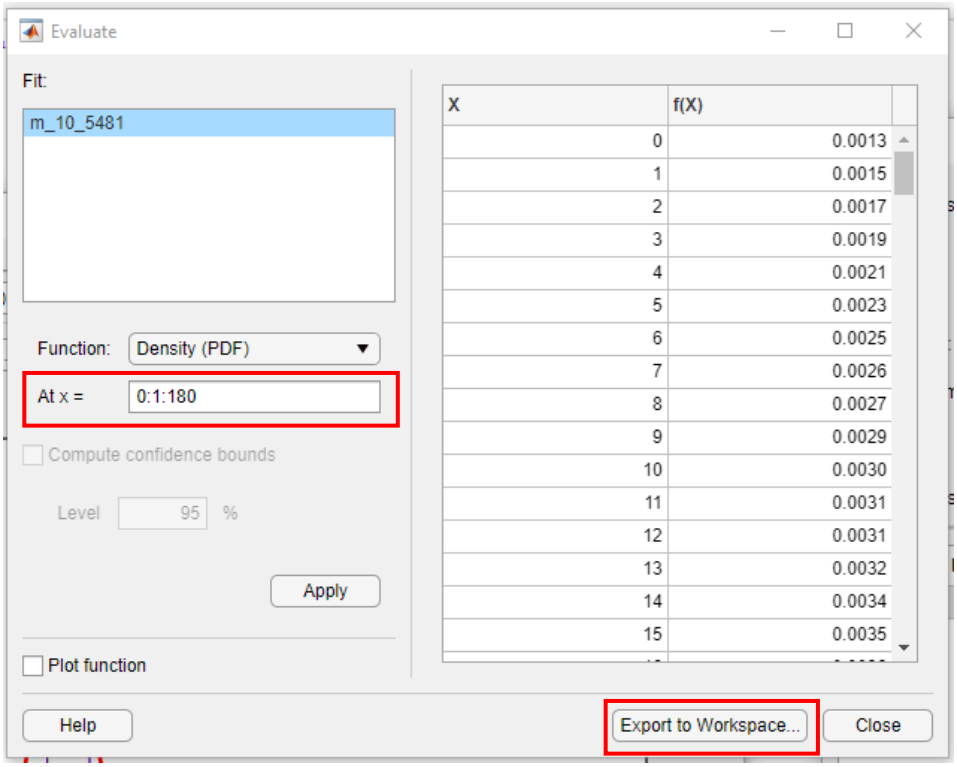


- The Kernel Density Estimate is now fit around the histogram.
- Select *Evaluate* to export to Excel.



- Set *At x =* to 0:1:180. This mean x will start at 0 and increase by one until reaching 180.
- Select *Export to Workspace* and give the file a descriptive name.

- Must start with a letter, does not like dashes, underscores ok
- In this example, I saved the results as *result_10_5481*

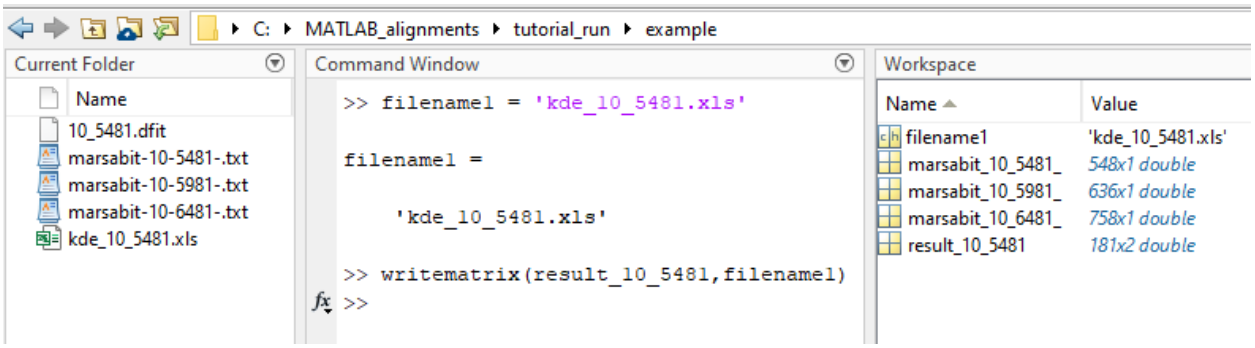


- Close the *Distribution Fitter* app. Save the fit, just in case.
- The results of the distribution is in the workspace.
- Use the following code to export to Excel:

```
filename1 = 'kde_10_5481.xls'

writematrix(results_10_5481, filename1)
```

  - filename1 is whatever you want to name the Excel file.
  - *result_10_5481* is what I called my exported KDE.
- You can see the new Excel file populate in the current folder.



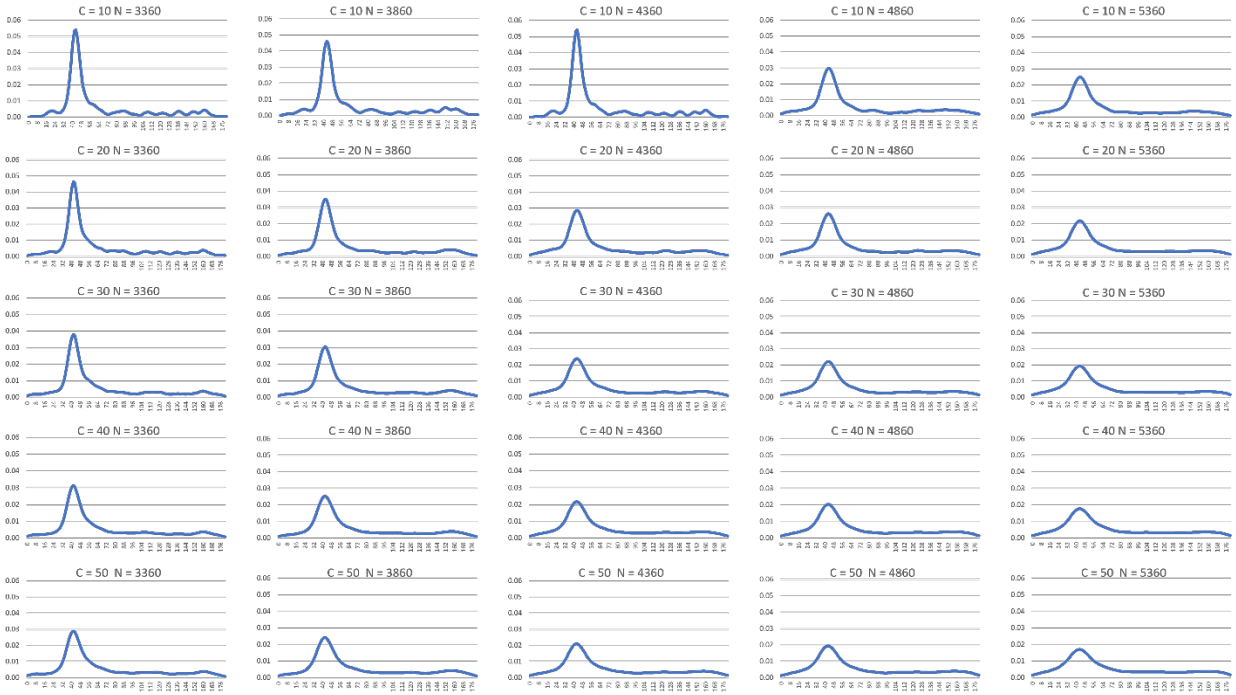
- Tada! Now go for a walk or something.

B5- EXCEL Identify KDE Peaks

Last updated: May 20, 2023

This process utilizes Kernel Density Estimates created by the Density Fitter App in MATLAB.

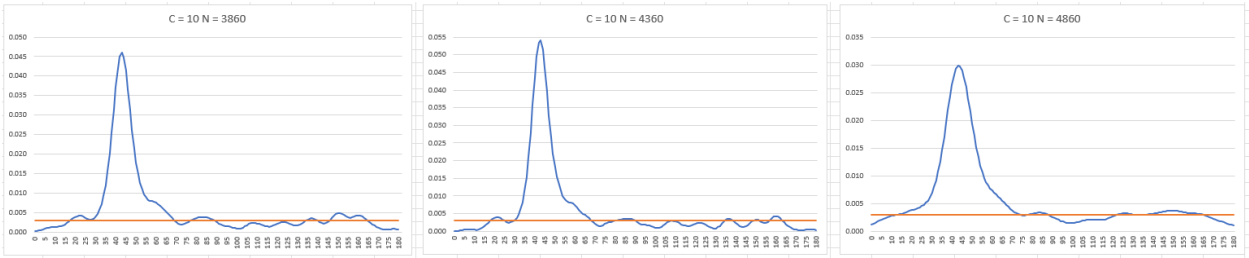
- Create line charts in Excel of each KDE fit created in MATLAB. Organize by width and length tolerances, like the rose diagrams in previous exercises.



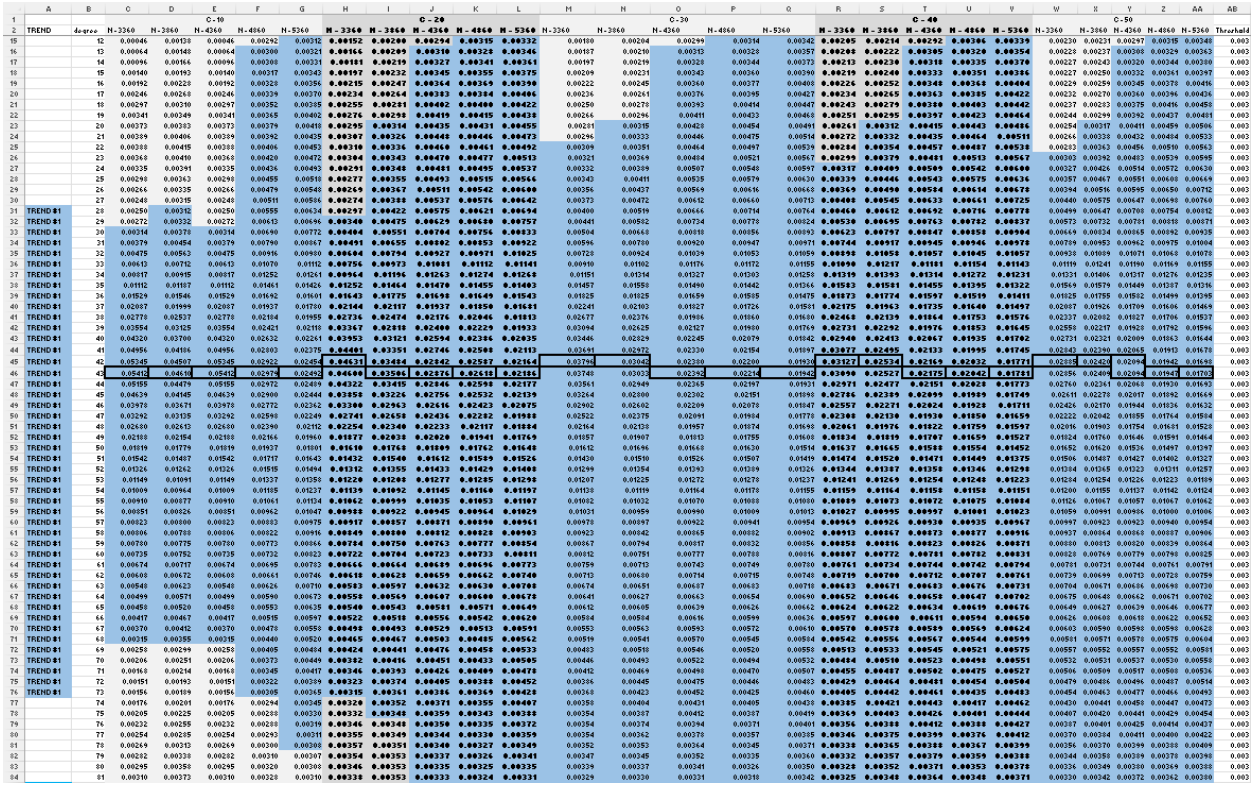
- Just by looking, can you identify any peaks that repeat in more than one chart? Does there appear to be a threshold most peaks rise above?
- Combine all kernel density estimates into one table. Find some of the peaks you identified in the charts. Can you find a threshold that encompasses most of them?

	C = 10					C = 20				
degree	N = 3360	N = 3860	N = 4360	N = 4860	N = 5360	N = 3360	N = 3860	N = 4360	N = 4860	N = 5360
6	0.00049	0.00099	0.00049	0.00231	0.00249	**0.00123**	**0.00158**	**0.00202**	**0.00232**	**0.00242**
7	0.00053	0.00112	0.00053	0.00245	0.00263	**0.00125**	**0.00168**	**0.00219**	**0.00248**	**0.00259**
8	0.00052	0.00121	0.00052	0.00258	0.00275	**0.00126**	**0.00176**	**0.00235**	**0.00263**	**0.00275**
9	0.00047	0.00127	0.00047	0.00268	0.00285	**0.00128**	**0.00182**	**0.00250**	**0.00277**	**0.00290**
10	0.00041	0.00130	0.00041	0.00277	0.00294	**0.00133**	**0.00188**	**0.00265**	**0.00290**	**0.00305**
11	0.00039	0.00133	0.00039	0.00285	0.00303	**0.00140**	**0.00193**	**0.00279**	**0.00303**	**0.00319**
12	0.00046	0.00138	0.00046	0.00292	0.00312	**0.00152**	**0.00200**	**0.00294**	**0.00315**	**0.00332**
13	0.00064	0.00148	0.00064	0.00300	0.00321	**0.00166**	**0.00209**	**0.00310**	**0.00328**	**0.00346**
14	0.00096	0.00166	0.00096	0.00308	0.00331	**0.00181**	**0.00219**	**0.00327**	**0.00341**	**0.00361**

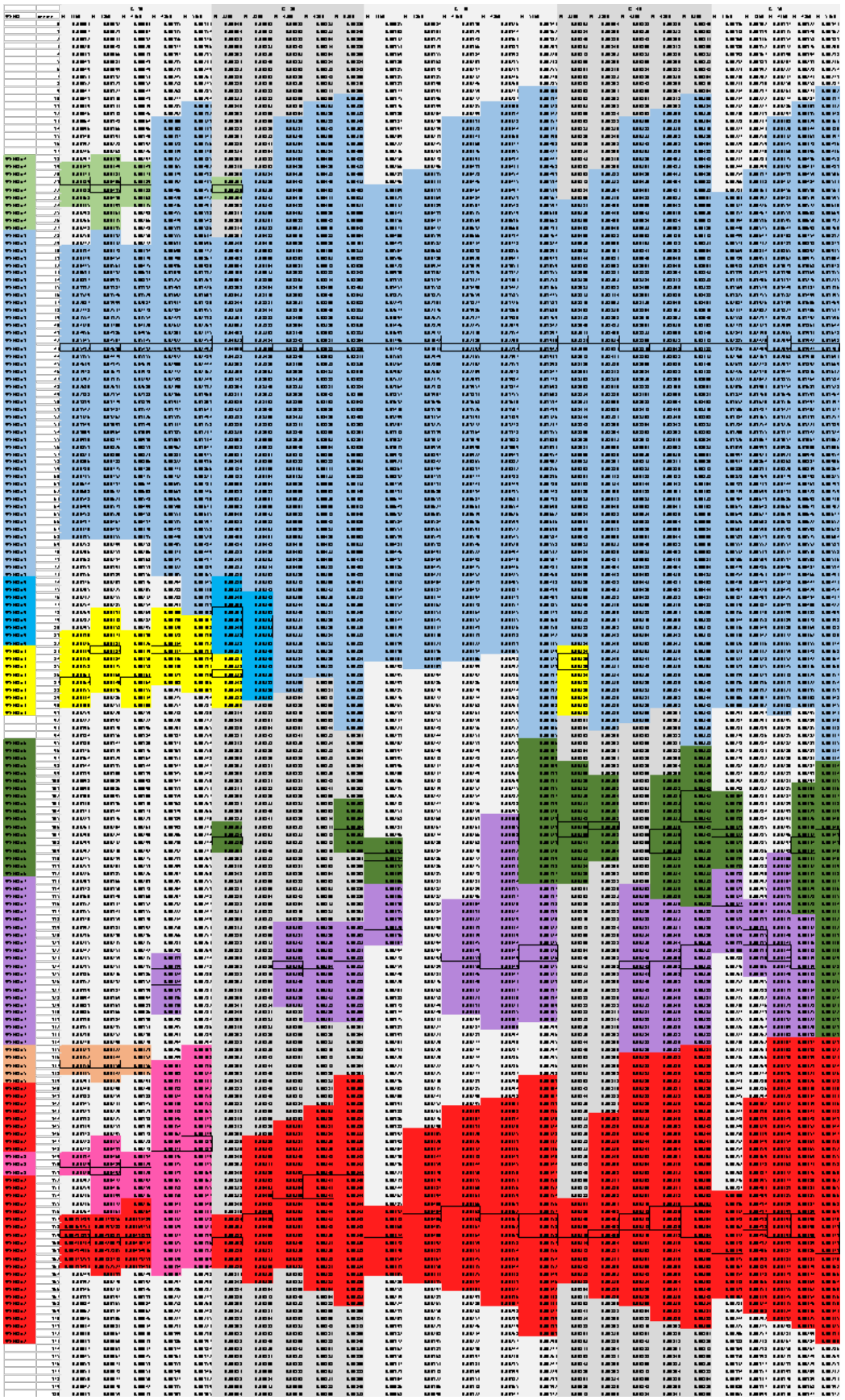
    - Don’t spend too much time on this step. Find a threshold and run with it. You can adjust this later.
- In the table, add a column with the threshold. For this example, I will use 0.003. Add it to all the charts. Take another look at the peaks. Does this threshold encompass most of the peaks that repeat from chart to chart?

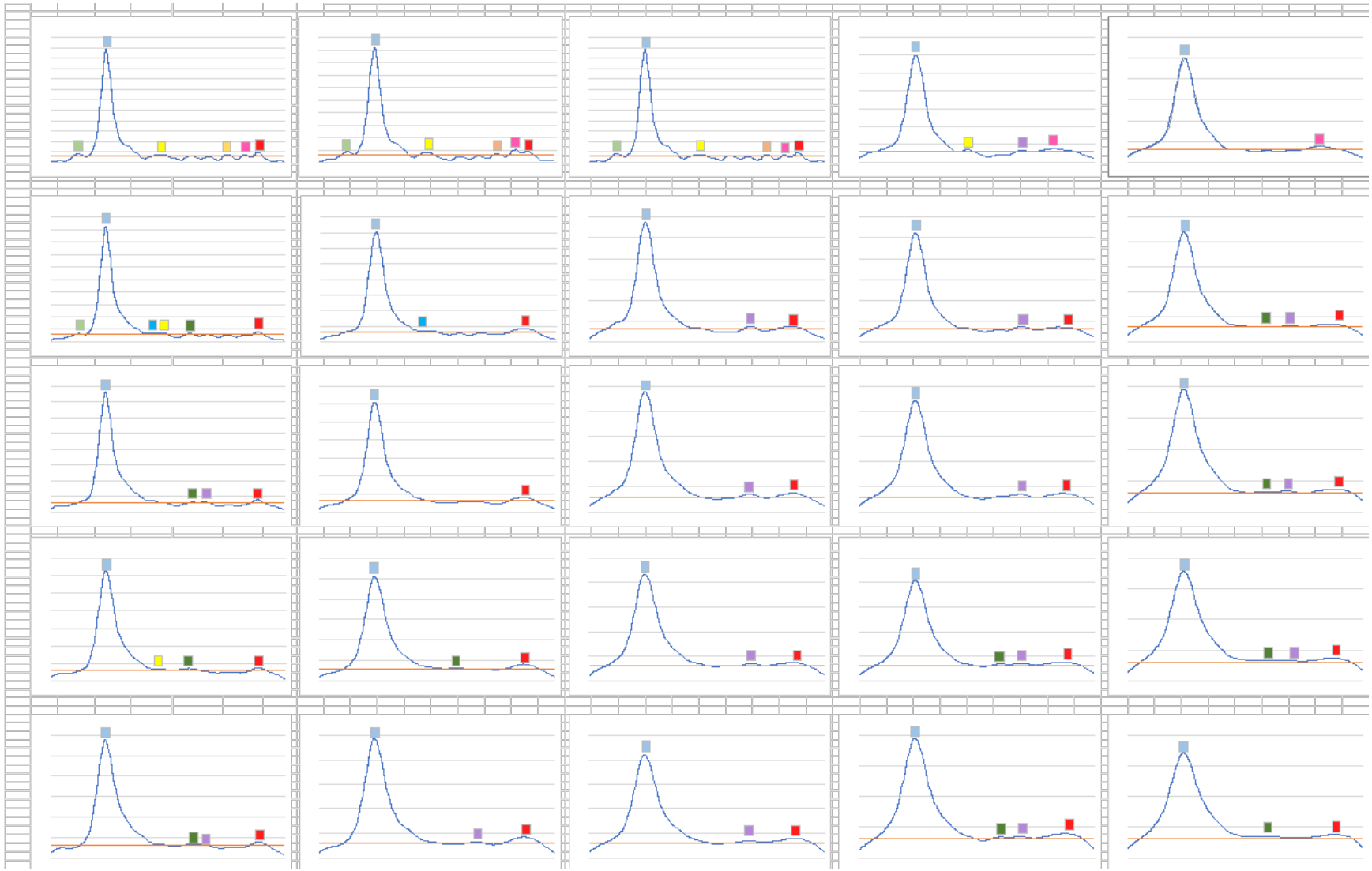


- Fine tune the threshold from here. You don’t want it to encompass ALL the peaks, just the repeating ones. If it’s too low, you will catch a lot of noise. Too high, and you may miss peaks.
- When you are happy with the threshold, label the largest, most repeating peak in each chart. Identify the peak in the table. Highlight all the densities that are part of this peak and greater than the threshold. Highlight the highest value in the peak somehow- I outlined the cell in bold. There should be a crescendo of increasing values until the highest value is reached, then a decrescendo of decreasing values.
    - If values start to increase again, then it is the start of a new peak.



- Repeat the process for every peak that is above the threshold. I found it helpful to color code everything, so the cell color in the table matches the label in the chart.





- Create a table with relevant info on the identified peaks

Peak #	degree min	degree max	average peak	peak StDev	highest density	occurrences	%
1	9	100	42.65	0.49	0.0541	25	100%
2	134	173	157.49	2.26	0.0043	23	92%
7	108	138	118.74	2.36	0.0037	15	60%
6	98	136	104.55	1.99	0.0035	11	44%
3	78	91	84.88	1.64	0.0039	7	28%
8	138	167	147.17	1.47	0.0049	5	20%
4	18	27	24.50	0.58	0.0048	4	16%
5	135	139	140.00	0.00	0.0036	3	12%
9	74	89	79.5	1.29	0.0036	2	8%

- From this info, I can conclude that peaks 1 and 2 are statistically significant and represent actual trends on Marsabit. Maaaaaaybe peak 7.