Estimating Home Ranges and Core Ranges in R and QGIS

Assignment:

1. Create two maps in QGIS one for short-finned pilot whales (*Globicephala macrorhynchus*) and one for Cuvier's beaked whales (*Ziphius cavirostris*) displaying the track points, core range, and home range for all animals using kernel density estimations in R.
2. Descriptively compare pilot whale and Cuvier's beaked whale home ranges by completing the following:
   a) Describe the two species' home ranges and core ranges.
   b) Compare the two species' home ranges (similarities and differences)
   c) Give possible reasons as of to why there are similarities and differences between the two species' home ranges. Think about their physiology, diet, behavior, etc., as well as the continental shelf and Gulf Stream.

   We expect a paragraph but no more than one page. Please cite the sources you use.

Instructions:

Required sections must be done to complete the assignment (see yellow highlights).

I. Importing and Project Data in QGIS

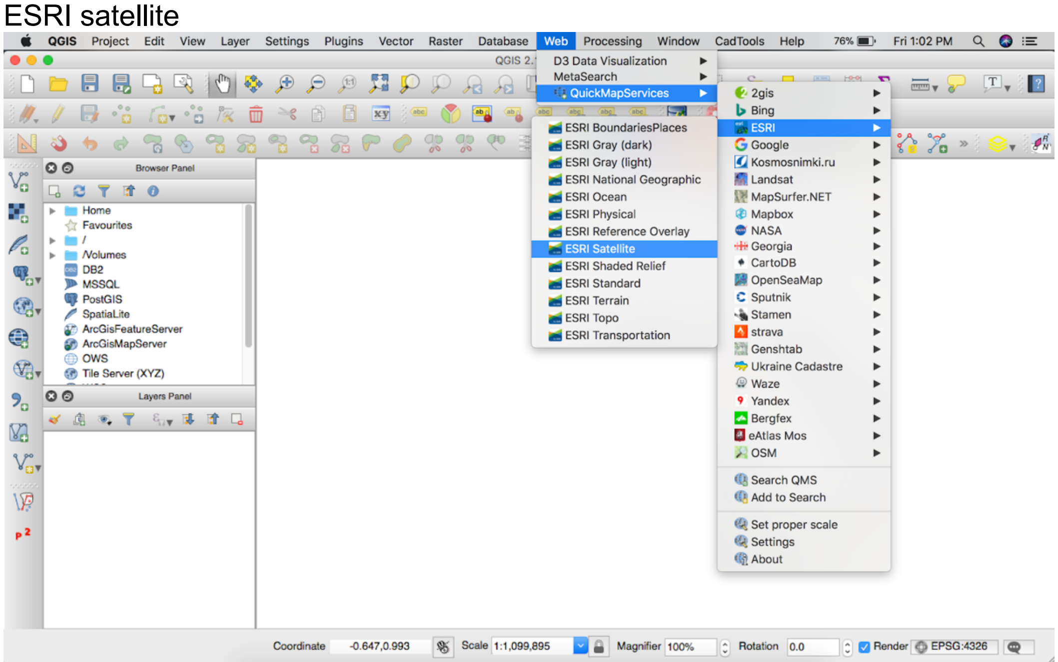
A. File Organization:

1. Make a Data Expeditions folder -> place the csv files in this folder & make a QGIS subfolder

B. Import Baseline Map Layer:

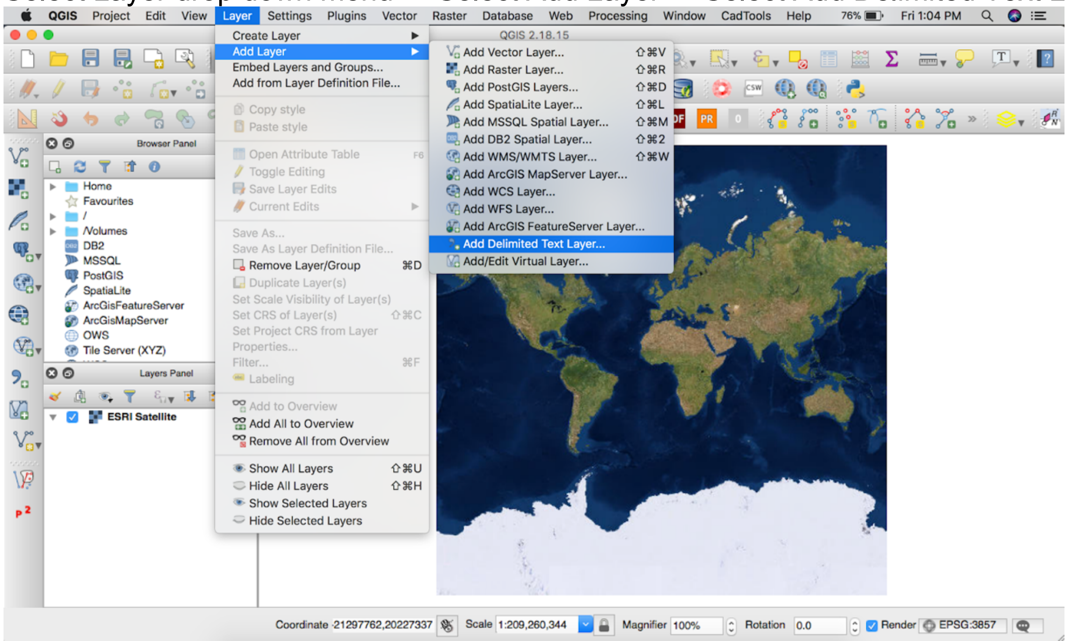
1. Open QGIS 2.18

2. Select the Web drop down menu -> Select Quick Map Services -> Select ESRI -> Select ESRI satellite

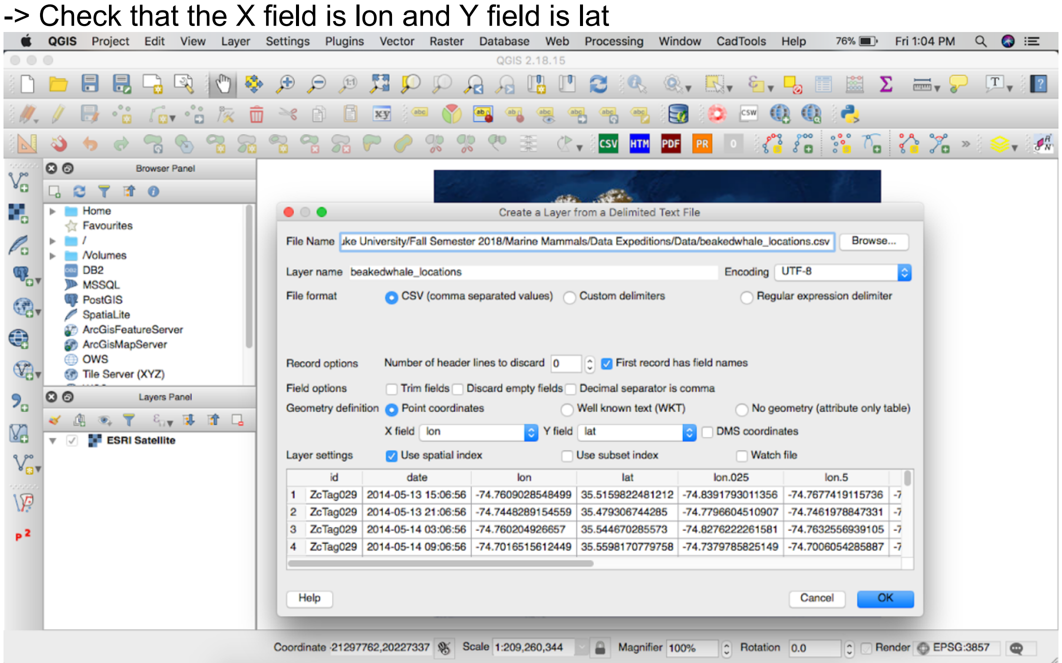


C. Import the CSV Files:

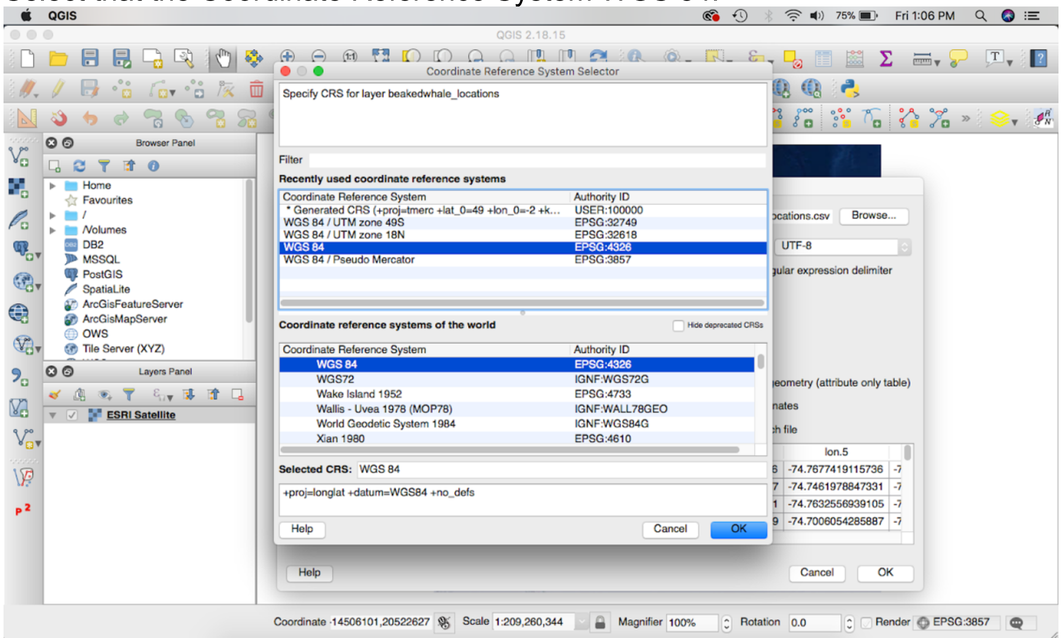
1. Select Layer drop down menu -> Select Add Layer -> Select Add Delimited Text Layer...



2. Select Browse...-> Select file beakedwhale_locations.csv in your Data Expeditions folder -> Check that the X field is lon and Y field is lat



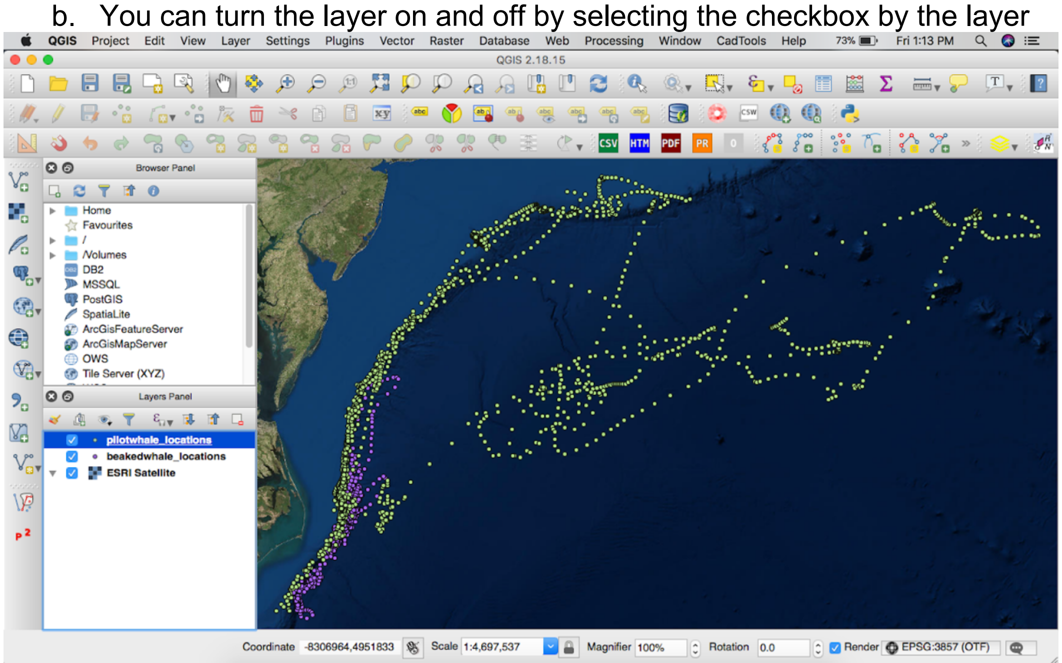
3. Select that the Coordinate Reference System WGS 84.



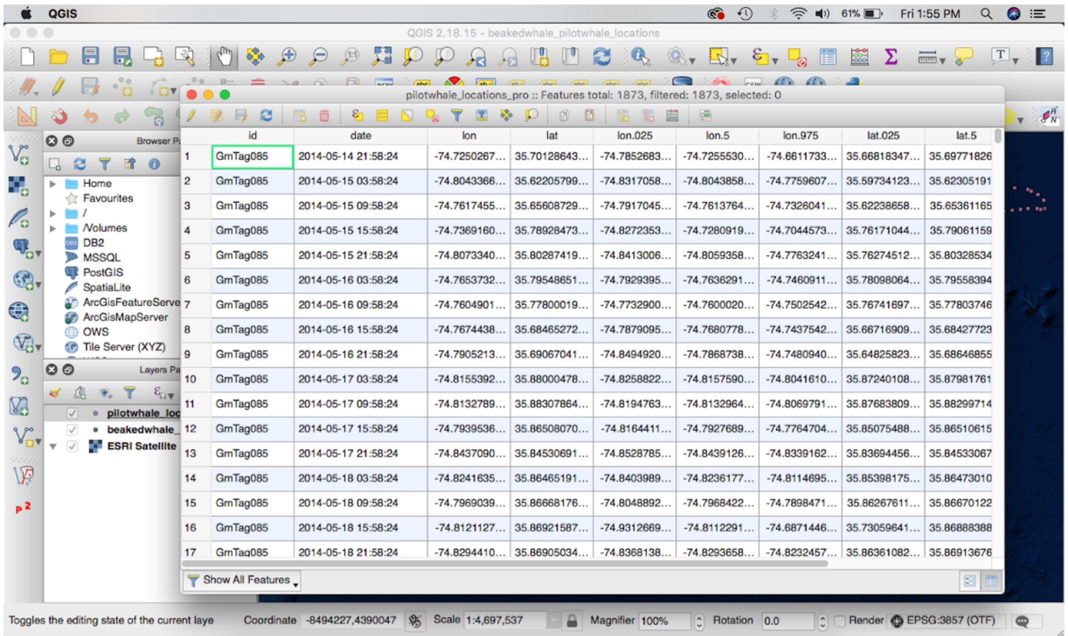
4. Repeat the steps 1-3 for the pilotwhales_location.csv data file

D. Examine the Data and Save the Project:

1. Right click the pilotwhale_locations layer -> select Zoom to Layer
   a. You can further zoom in and out using your touch pad or the +/- magnifying icons  in the toolbox panel
   b. You can turn the layer on and off by selecting the checkbox by the layer



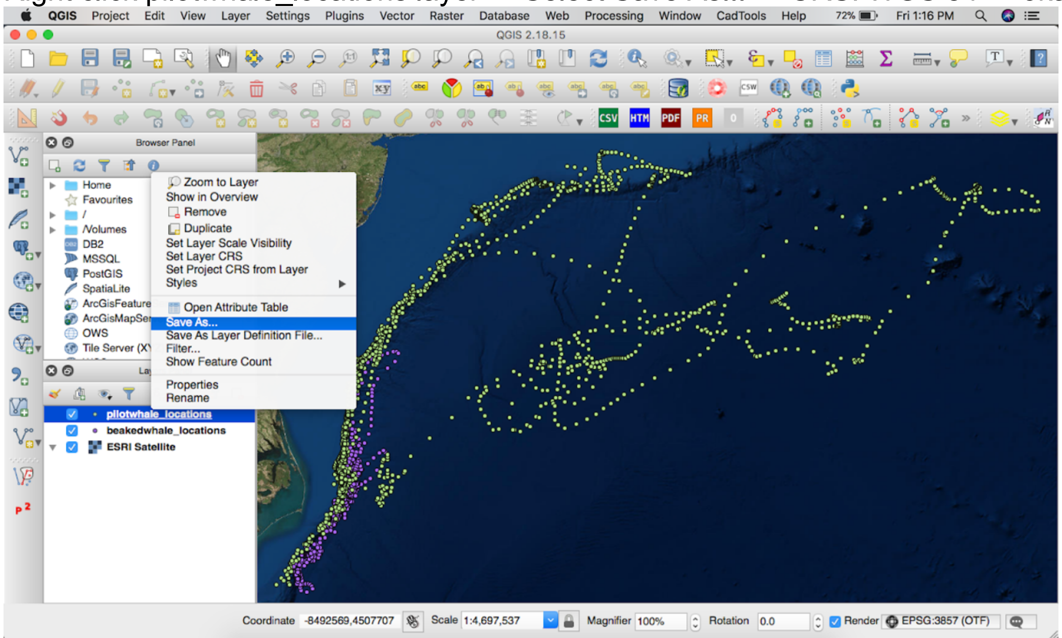
2. Right click the pilotwhale_locations and/or beakedwhale_locations layers -> Select Open Attribute Table.
    a. You will see that there is an id, date, lon, lat, and 6 other long and lat columns. Satellite Tag data often results in erroneous points. These points have been filtered out already using a Bayesian statistics model known as BSAM.
    b. There are track points for 5 individuals (e.g. id) of each species. The date is the date and time for each lon and lat coordinate recorded. section.



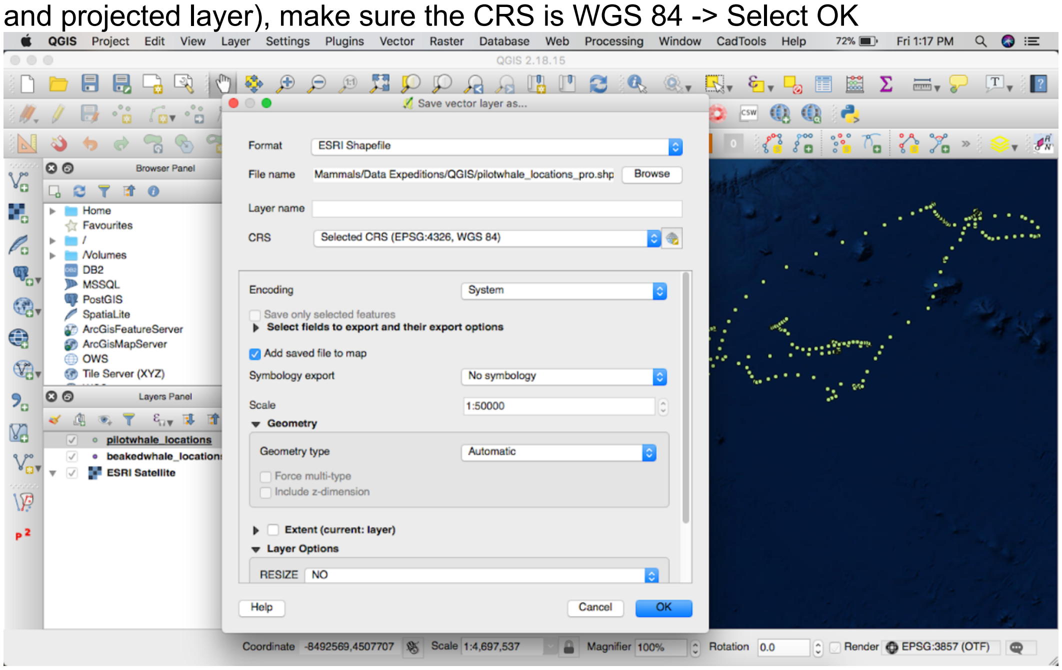
3. Save your project: Select the Project drop down menu -> Select Save As -> save in your QGIS folder with whatever file name you choose
    a. Notice that there are no other files in the QGIS folder. Each shape file layer needs to have a set of files (5) or else you will be unable to retain your work on other systems or manipulate those individual files' attribute tables.

E. Save and Reproject Telemetry Point Layers:

1. Right click pilotwhale_locations layer -> Select Save As... -> CRS: WGS 84 -> okay



2. Select Browse -> Navigate to your Data Expedition's QGIS folder -> Name the file pilotwhale_location_pro (the pro is short for projected to remind yourself this is a saved and projected layer), make sure the CRS is WGS 84 -> Select OK



3. Check your QGIS folder. Now, you'll see that there are 5 files (.dbf, .prj, .qpj, .shp, .shx) for this one layer.

4. Repeat steps 1 & 2 for the beakedwhale_locations layer

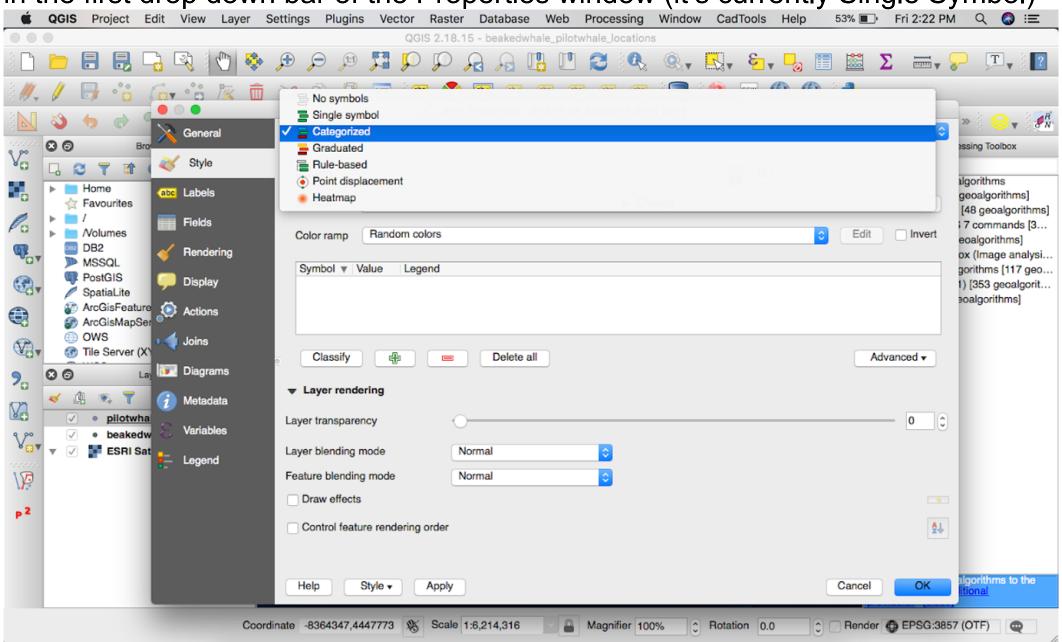
5. Remove the pilotwhale_locations and beakewhale_locations layer: Right click the layers -> Select remove

II. Visualize the Data

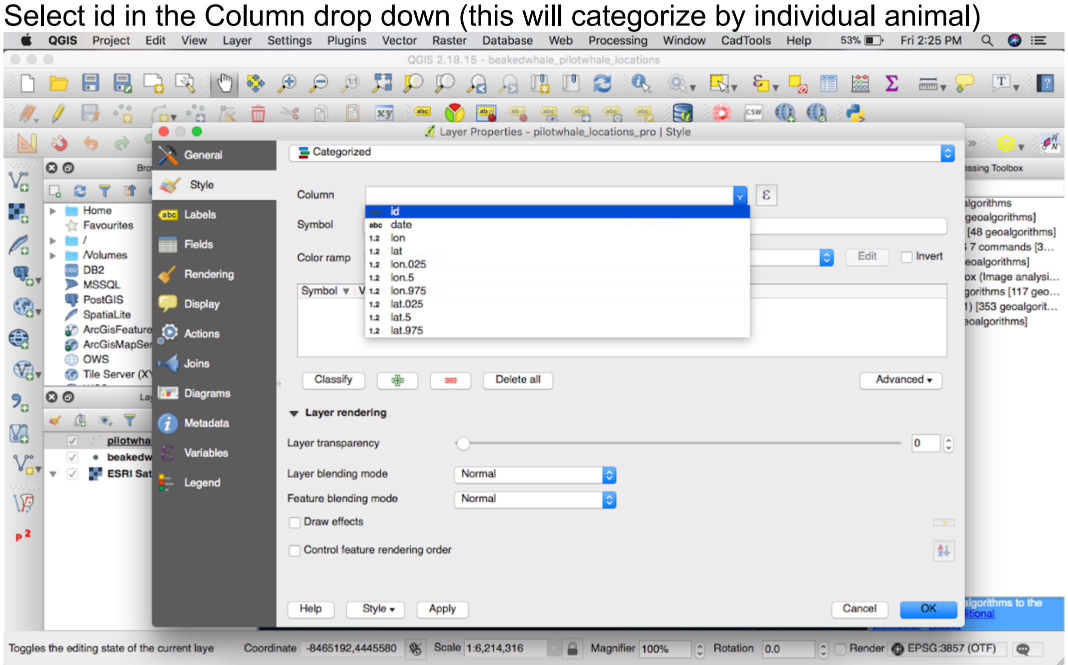
A. Categorize Layers and Look at Individual Variation (Required):

Currently both the pilot whale and beaked whale layers do not differentiate the points by individual (e.g. id), to look at individual variation in location:

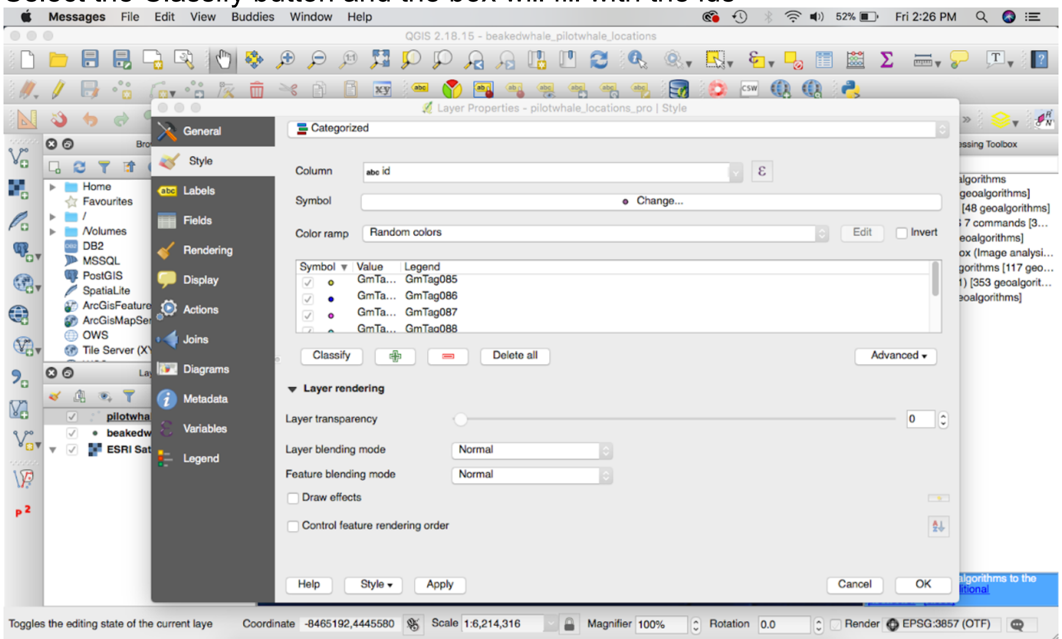
1. Right click the pilotwhale_locations_pro layer -> Select Properties -> Select Categorized in the first drop down bar of the Properties window (it's currently Single Symbol)



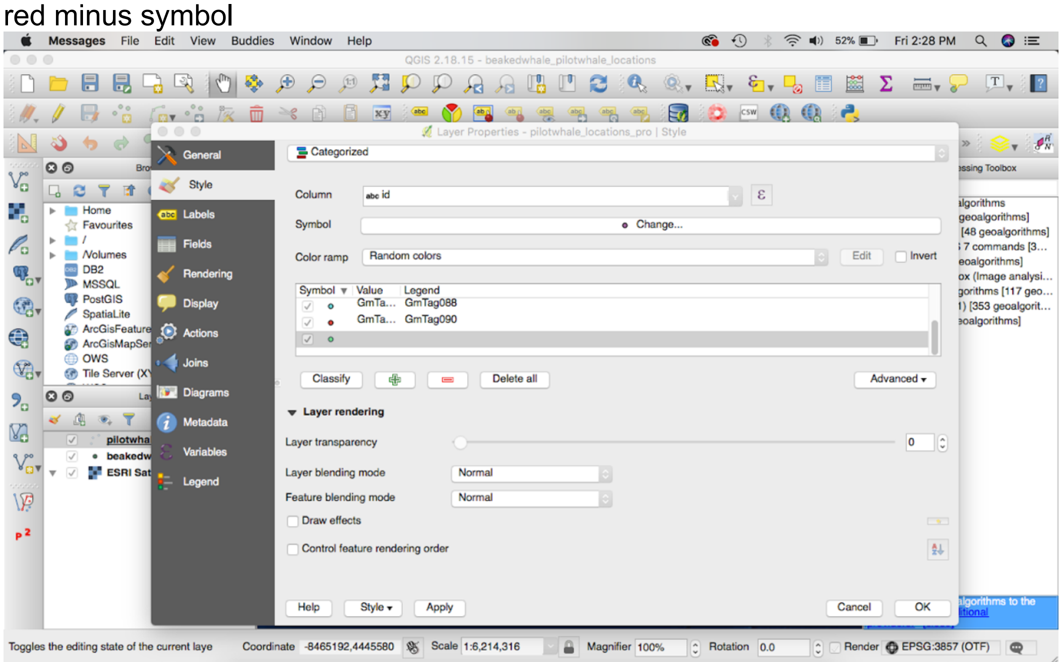
2. Select id in the Column drop down (this will categorize by individual animal)



3. Select the Classify button and the box will fill with the ids

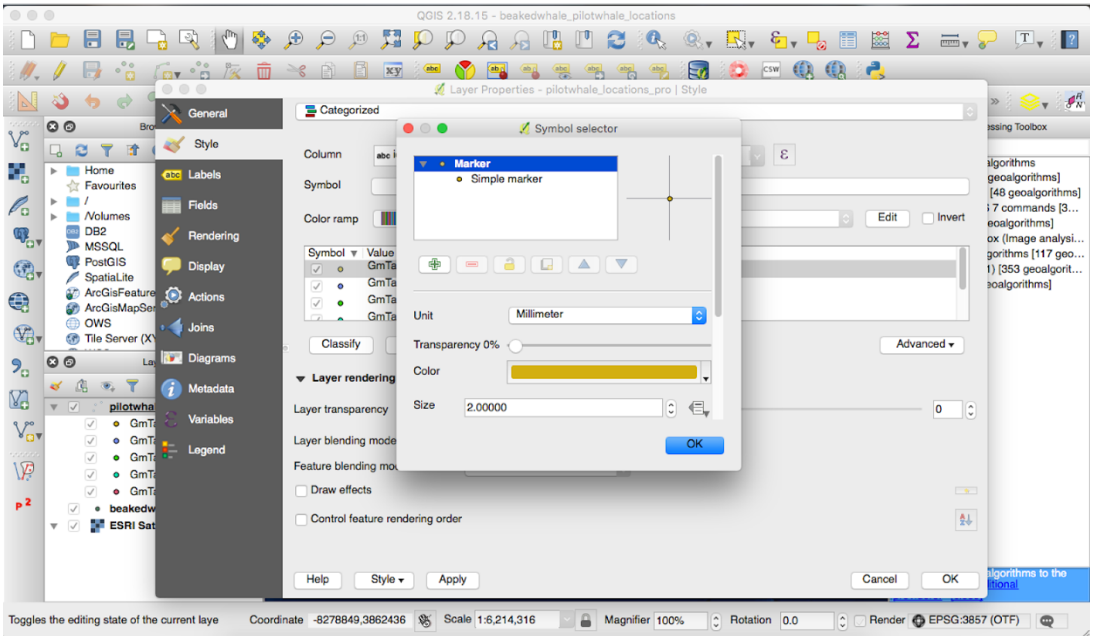


4. Remove the row for the point without any value or legend by selecting the row and the red minus symbol



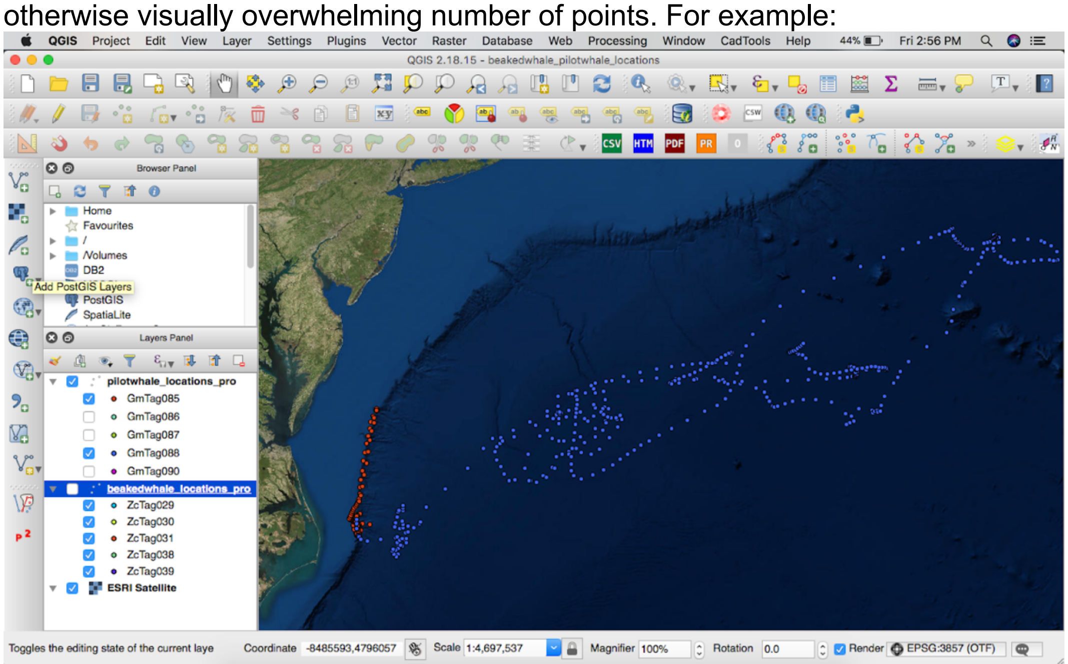
5. You can change the color, size, and transparency of points for each individual by double clicking the Symbol icon (i.e. ∘) in each individual's row and making adjustments in the Symbol Selector window. You can also change the color by selecting preset colors from the Color ramp drop down in the Layer Properties window. You can also change the entire layers transparency by adjusting the Layer transparency in the Layer Properties

window.



6. Repeat the steps 1-5 for the beakedwhale_locations_pro layer.

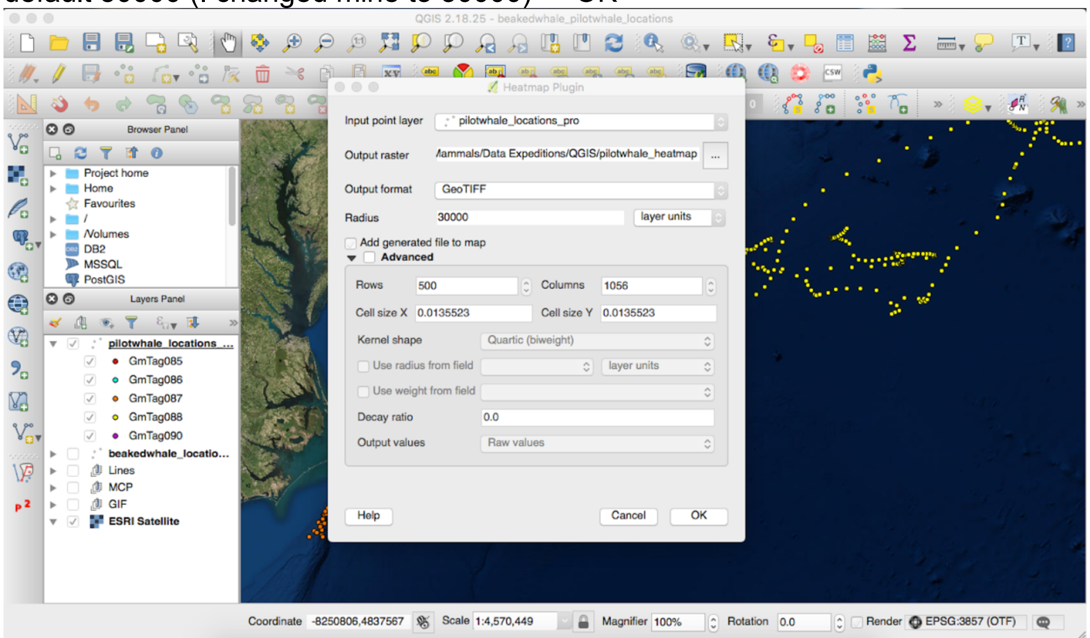
7. You can now visualize each individual pilot and beaked whales' track points. Select the checkbox by individuals in the Layers Panel to turn the pilotwhale_locations_pro and beakedwhale_locations_pro layers on and off & nested individual pilot and beaked whale tracks on and off (i.e. GMTag085, ZcTag029). Play around with this to break down the otherwise visually overwhelming number of points. For example:



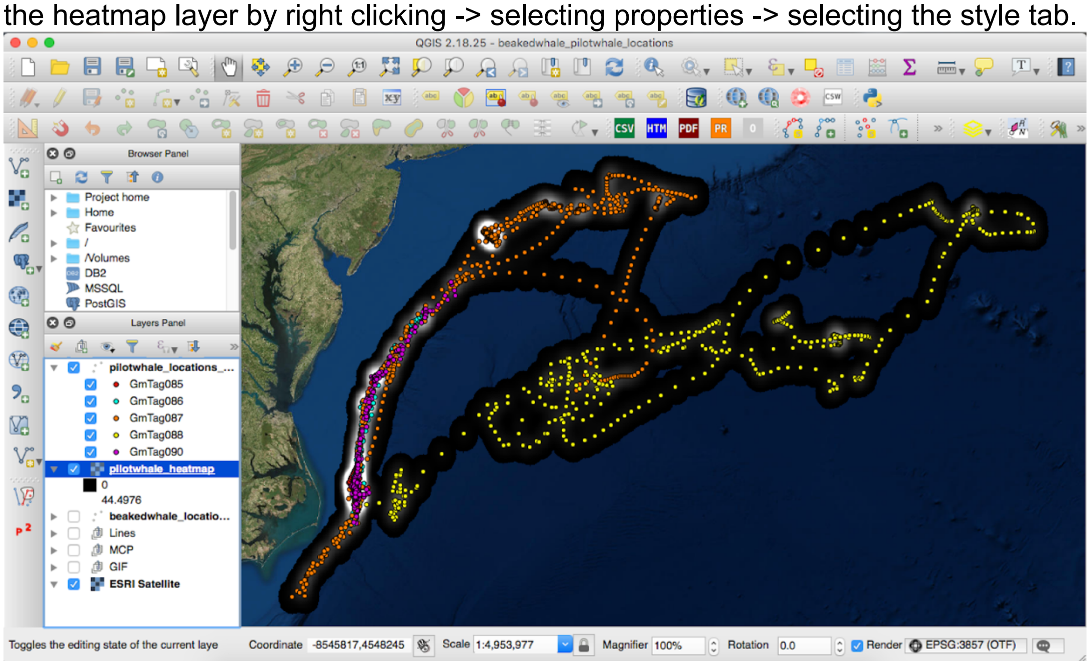
B. Using Heat Map to Look at Point Density (Optional):

1. Select the Raster drop down menu -> Heatmap -> Heatmap

2. Select the pilotwhale_locations_pro as the input point layer -> Output raster should be saved to your QGIS folder and named pilotwhale_heatmap -> Radius can remain as the default 50000 (I changed mine to 30000) -> OK



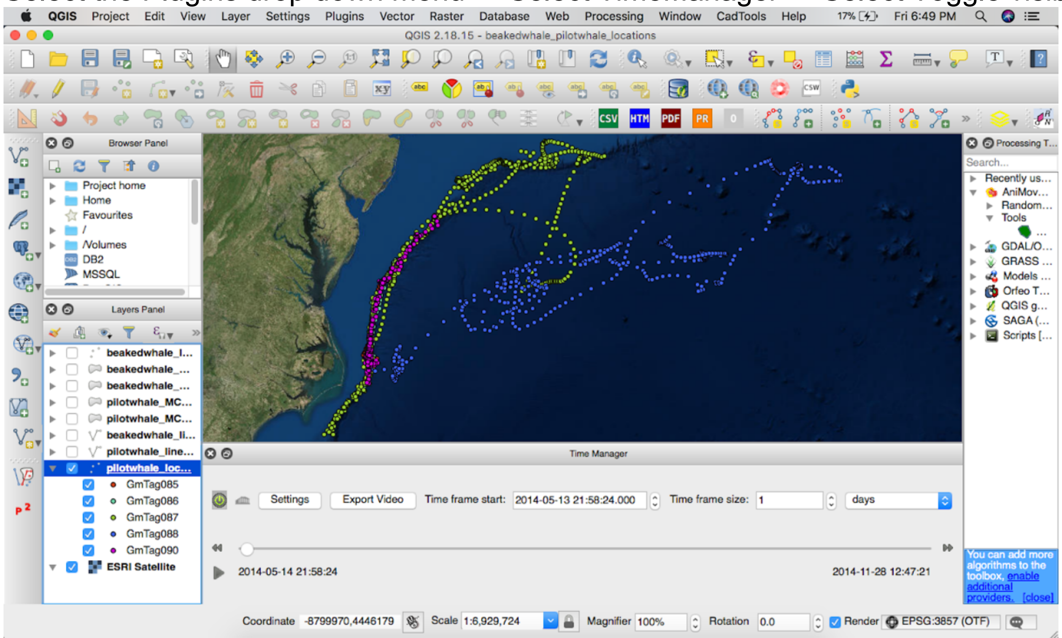
3. The white area is the area with the highest density of points. You can overlay the pilotwhale_locations_pro layer over the heatmap by dragging it on top of the pilotwhale_heatmap layer in the Layers Panel. Again, you can also adjust the coloring of the heatmap layer by right clicking -> selecting properties -> selecting the style tab.



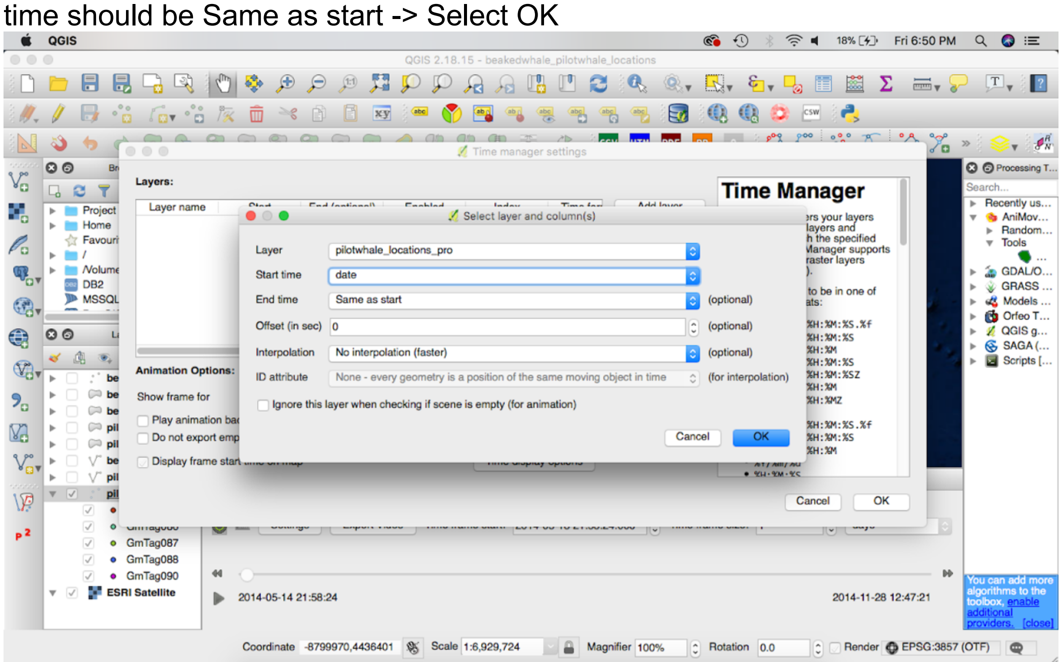
4. Repeat steps 1-3 for the beakedwhale_locations_pro layer.

C. Time Manager (Optional):

1. Select the Plugins drop down menu -> Select TimeManager -> Select Toggle visibility



2. Select the Settings button -> Select the pilotwhale_locations_pro or beakedwhale_locations_pro layer -> Select date from the Start time drop down -> End time should be Same as start -> Select OK

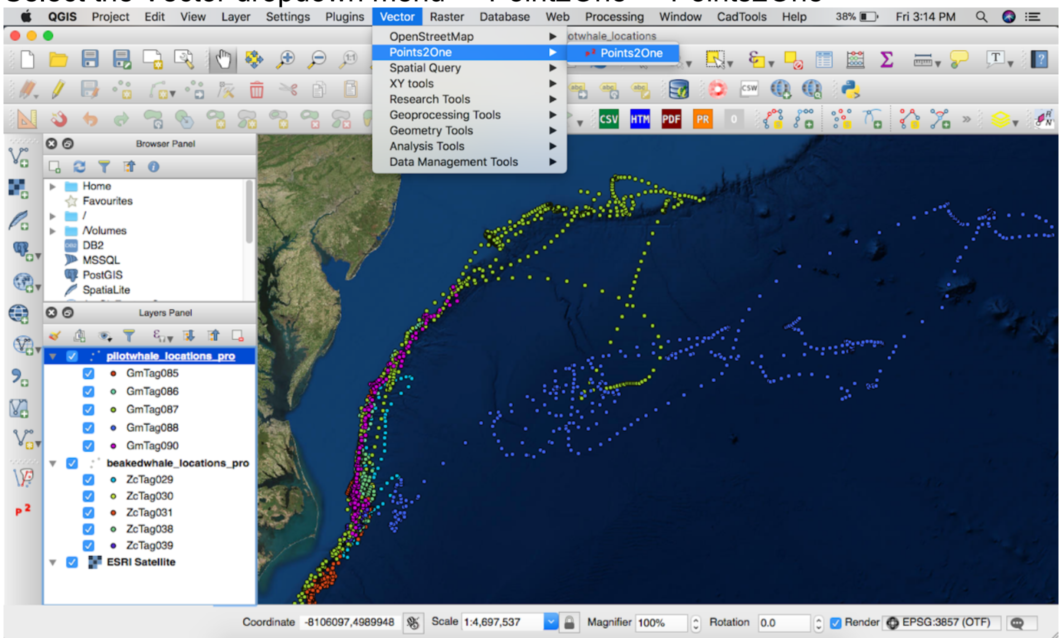


3. Enter the time frame size and units you want your animation to run at -> Select the play button and watch the animation go! -> Adjust the time frame size and units until you're satisfied
   a. For pilot whales, try a Time frame of 6 with the unit being hours

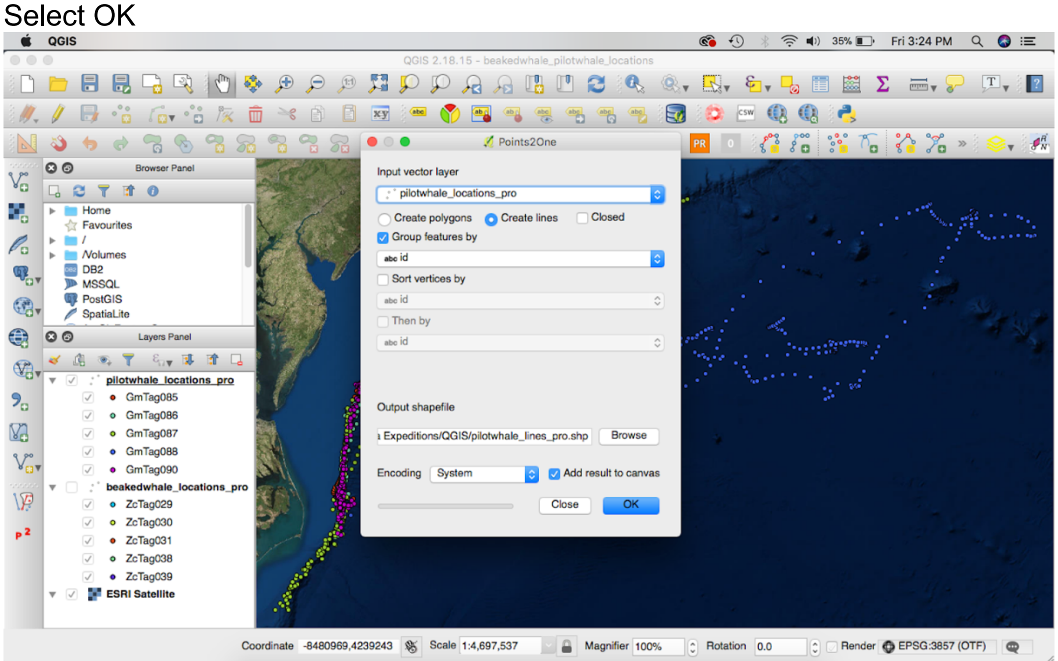
4. See Section IV. Making Maps in QGIS to learn how to export your animated map as a GIF

D. Convert Points into Lines and Look at Individuals' Paths (Optional):

1. Select the Vector dropdown menu -> Point2One -> Points2One

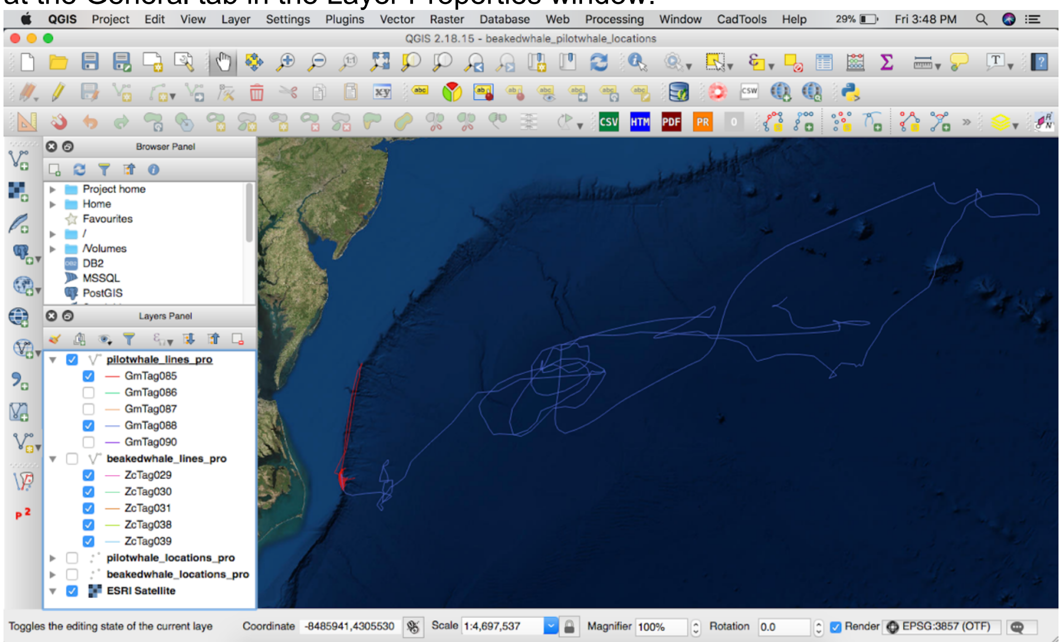


2. Points2One window: Select pilotwhale_locations_pro as the input vector layer -> Select Create lines -> Check Group features by id -> Browse and save the file as pilotwhale_lines_pro in the QGIS folder -> Check Add result to canvas create lines -> Select OK



3. Repeat steps 1 & 2 for the beakedwhale_locations_pro layer

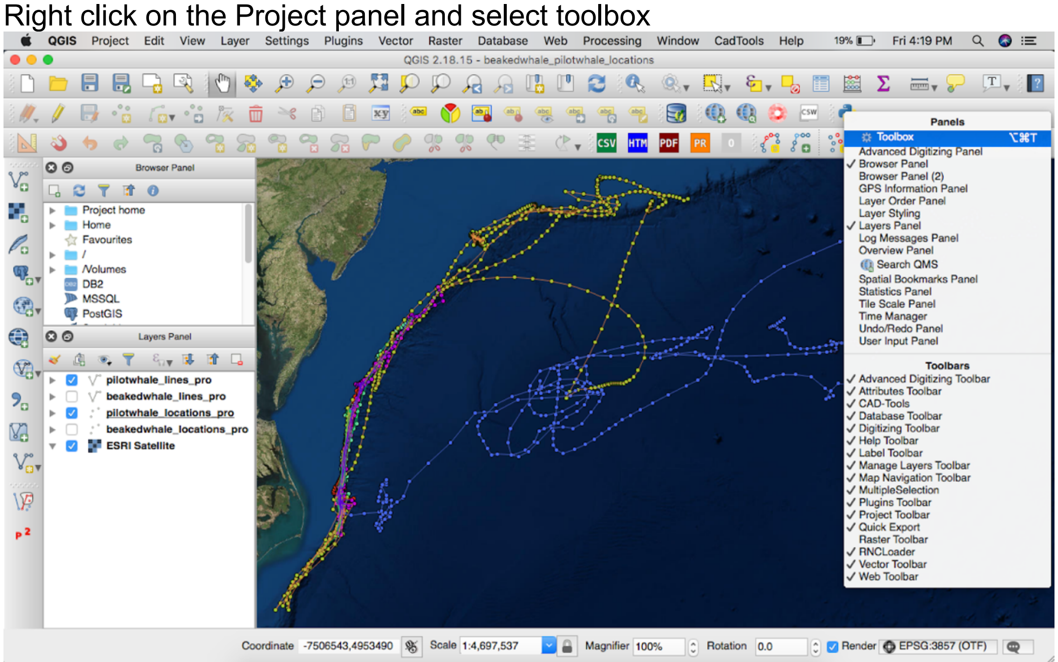
4. Again, both the pilotwhale_lines_pro and beakedwhale_lines_pro layers are showing up as one line and are not displaying lines per individual. Repeat the steps in the Categorize Layers and Look at Individual Variation section for both the pilot and beaked whale layers. Try to make sure that each individual has the same color for its location points and lines. While doing this also make sure that the CRS is still WGS84 by looking at the General tab in the Layer Properties window.



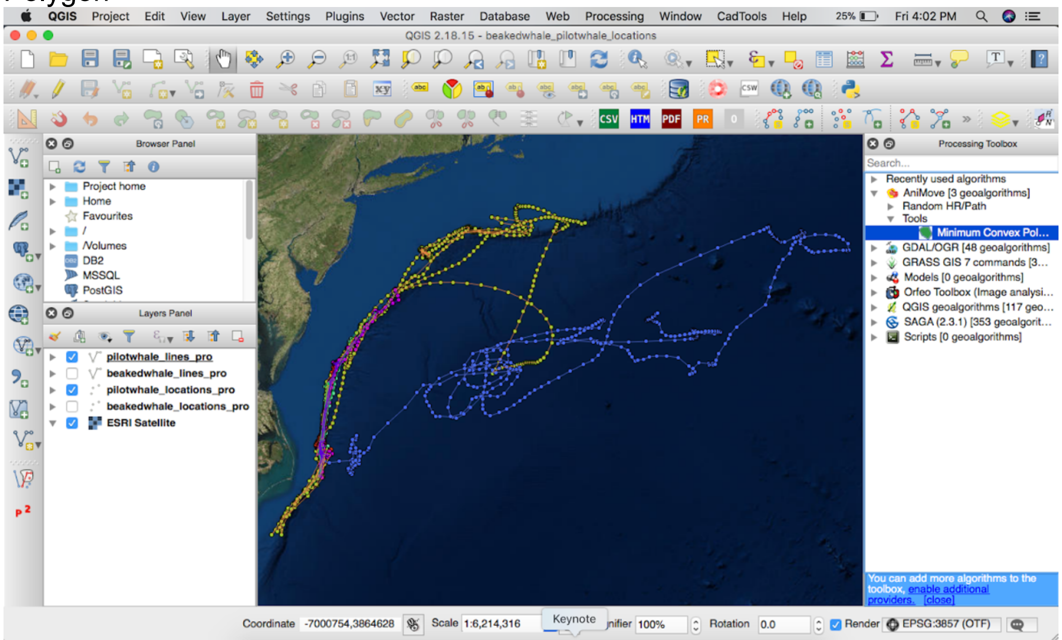
III. Estimating MCP and Kernel Density in QGIS and R:

A. Using the Minimum Convex Polygon Tool in QGIS (Optional):

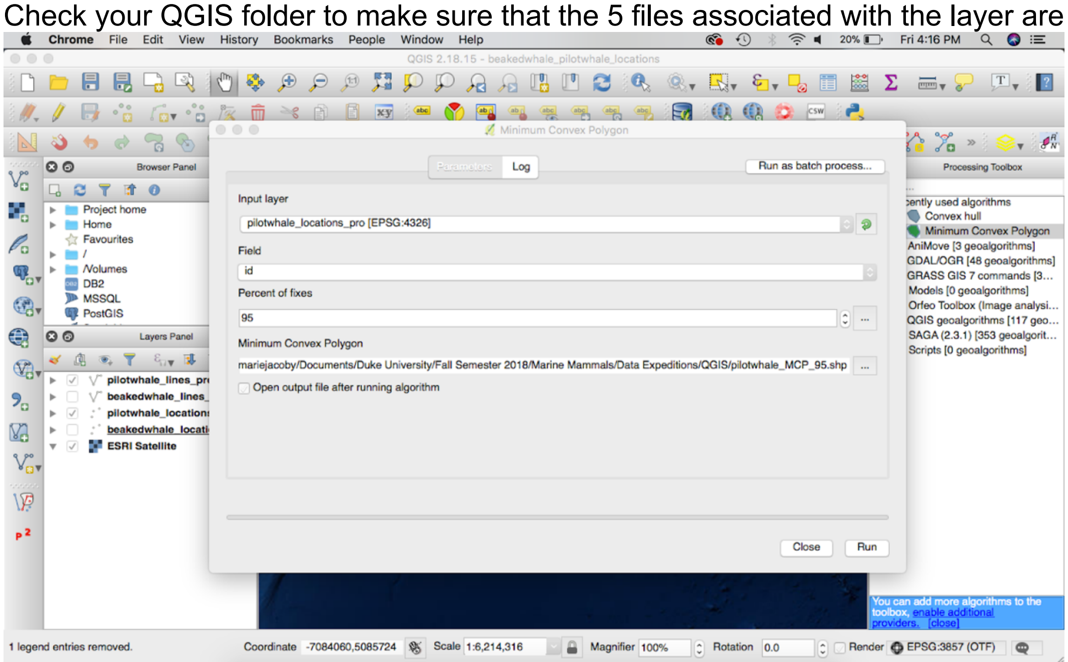
1. Right click on the Project panel and select toolbox



2. Toolbox processing window: Select AniMove -> Select Tools -> select Minimum Convex Polygon



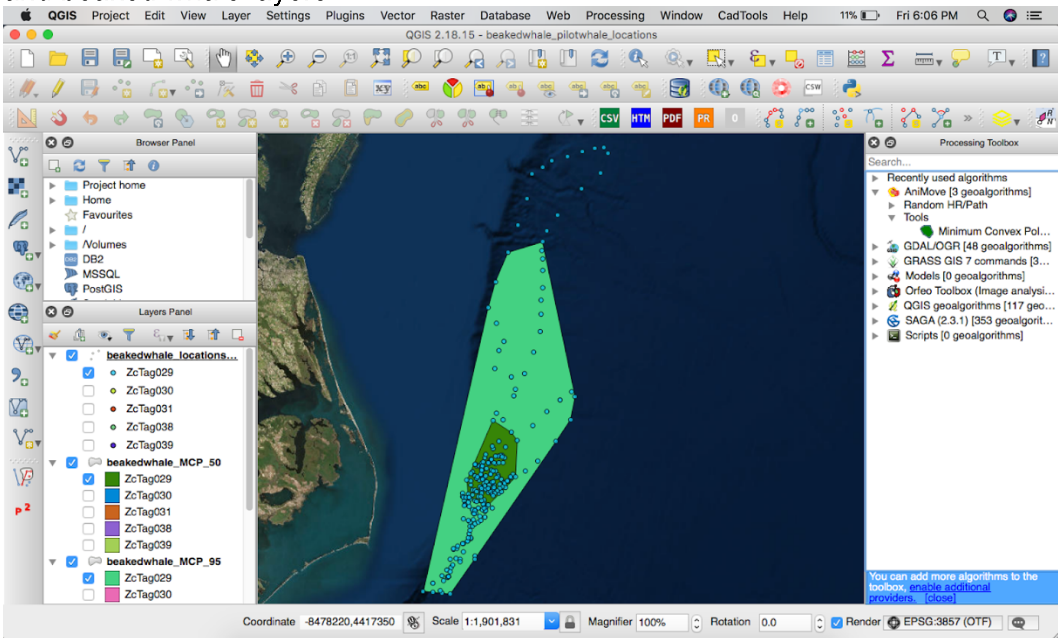
3. Select pilotwhale_locations_pro [EPSG:4326] in the Input Layer drop down -> Select id in the Field drop down -> Percent of fixes needs to be 95 or 50 (for 95% or 50%) -> Select… for the Minimum Convex Polygon -> Select Save to file and find the QGIS folder and save the file as pilotwhale_MCP_95 or pilotwhale_MCP_50 -> Select Run -> Check your QGIS folder to make sure that the 5 files associated with the layer are there



4. Repeat steps 1 & 2 for the beakedwhale_locations_pro layer

5. Again, the minimum convex layer will need to be categorized by individual. Repeat the steps in the Categorize Layers and Look at Individual Variation section for both the pilot

and beaked whale layers.



B. Estimating 50% and 95% MCP and Kernel Density in R:

1. Open RStudio
2. Use the following code:

Inputting Files and Loading Packages:

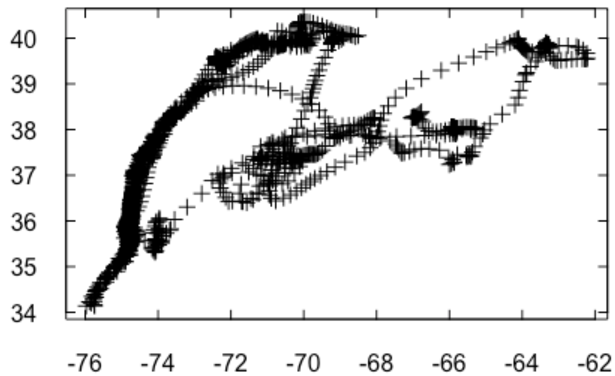
```
First, save your R Script in the same folder where the data files are s
tored.

#In a new R script, you should always remove objects from the environme
nt using the following code:
rm(list=ls())

#Set the working directory so that R knows where to read csv files from
:
Session -> Set Working Directory -> To Source File Location

#Or you can set the working directory by using the following code and y
our file path:
setwd("~/Documents/Duke University/Fall Semester 2018/Marine Mammals/Da
ta Expeditions/Data")

#You have to install and/or load a few packages that will allow you to
calculate, map, and export kernel density estimates and minimum convex
polygons.

#If you have not installed packages adehabitatHR, rgdal, or raster, use
the following code:
install.packages(“adehabitatHR”)
install.packages(“rgdal”)
```

```r
install.packages("raster")
install.packages("rgeos")

#If you have previously installed these packages, load the libraries:
library(adehabitatHR)
library(rgdal)
library(raster)
library(rgeos)

#Now you need to read in the data files by assigning them to a variable
using the read.csv function (we will walk you thorugh the pilotwhale da
taset):
pwdat <- read.csv("pilotwhale_locations.csv")

#Go ahead and take a look at the data by using the summary function:
summary(pwdat)

#You can also view the data frame by selecting it in the Environment wi
ndow
```

Estimating KDE and MCP for All Pilot Whales:

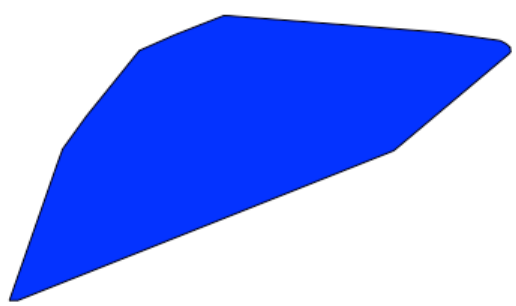
I. Prepare the Data:

```r
#First, create a spatial points variable for lon and lat so that R reco
gnizes these two columns as geographic coordinates:
pw.all.sp <- SpatialPoints(pwdat[c("lon", "lat")])


#Make sure the variable is projected as CRS = WGS84
proj4string(pw.all.sp) = CRS("+init=epsg:4326")


#Reproject to the appropriate CRS (in this case it will remain as WGS84
) and assign as a new variable:
pw.all.sp <- spTransform(pw.all.sp,CRS("+init=epsg:4326"))

#Take a look at the locations by using the plot function:
plot(pw.all.sp)
#Add the lon and lat axes:
axis(1, tcl = .3)
axis(2, las = 1, tcl = .3)
axis(3, labels = NA, tcl = .3)
axis(4, labels = NA, tcl = .3)
box()
```



II. Minimum Convex Polygon (Optional):

```
#Now, you are ready to calculate the minimum convex polygon (100%) for
all of the pilot whales using the mcp function:
pw.all.mcp <- mcp(pw.all.sp, percent = 100, unin = "m", unout = "km2")

#Plot it:
plot(pw.all.mcp, col="blue")
```

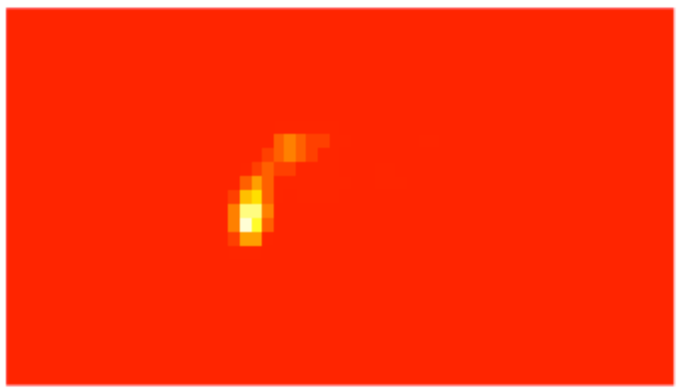


```
#For the home range MCP you will want to use 95% (instead of 100%):
pw.all.95.mcp <- mcp(pw.all.sp, percent = 95, unin = "m", unout =
"km2")
#Plot it:
plot(pw.all.95.mcp)

#For core range MCP you will want to use 50% (instead of 100%):
pw.all.50.mcp <- mcp(pw.all.sp, percent = 50, unin = "m", unout =
"km2")
#Plot it:
plot(pw.all.50.mcp)

#Write your 95% and 50% MCP into a shapefile that can be brought into Q
GIS using the writeOGR function:
```

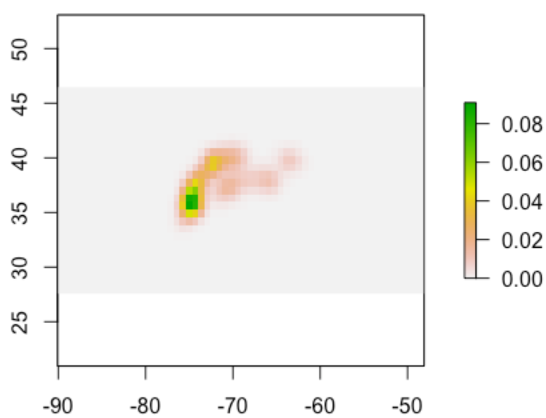
```
writeOGR(pw.all.95.mcp, dsn = ".", layer = "PilotWhale_All_95_MCP", dri
ver="ESRI Shapefile")
#A dsn value of "." will output the file in your working directory fold
er, layer is the file name, driver is the type of file that is being ma
de.
#Repeat for the 50% MCP.
```

III. Kernel Density Estimate:

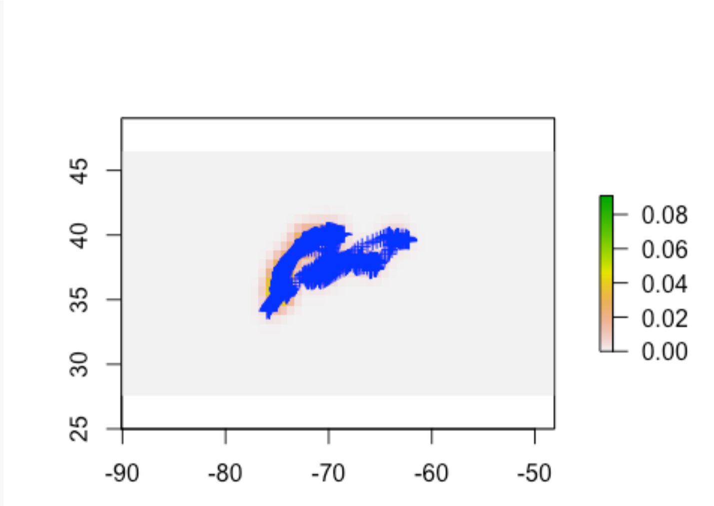
```
#Another (and better way) to estimate home range and core range is usin
g a kernel density estimate via the kernelUD function:
pw.all.kde <- kernelUD(pw.all.sp, h="href")
#Plot it:
image(pw.all.kde)
```



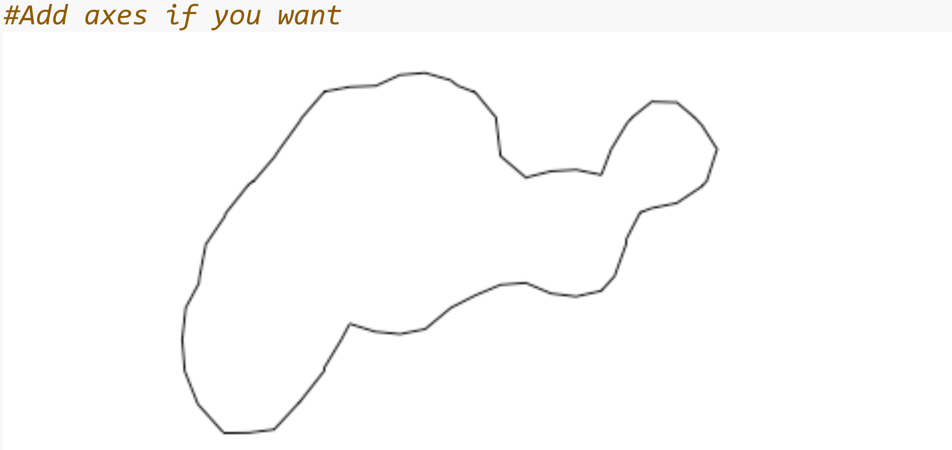
```
#Turn the KDE image into a raster file:
pw.all.rast <- (raster(as(pw.all.kde, "SpatialPixelsDataFrame")))
#Plot it:
plot(pw.all.rast)
```




```
#Overlay the track points:
plot(pw.all.sp, add = T, col="blue")
```




```r
#If you want to export a visual, you can select Export in the Plots window.

#Output the kernel density as a rasterized geo tiff that can be brought into QGIS using the writeRaster function:
writeRaster(pw.all.rast, "pilotwhale_all_kde.tif") #this file will be stored in the folder you set as your working directory

#Estimate home range KDE (95%) using the getveticshr function:
pw.all.95.kde <- getverticeshr(pw.all.kde, percent = 95, unin = "m", unout = "km2")
#Plot it:
plot(pw.all.95.kde)
#Add axes if you want
```



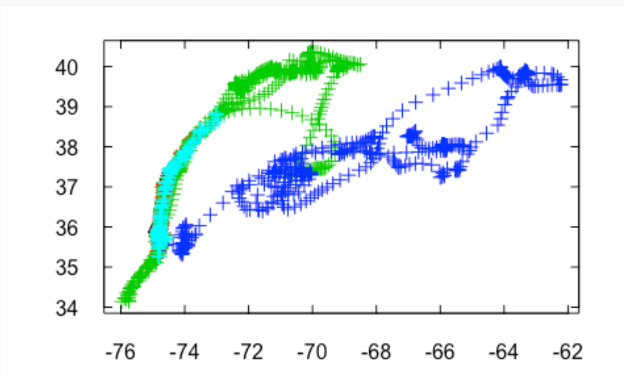
```r
#Estimate core range KDE (50%) using the getveticshr function:
pw.all.50.kde <- getverticeshr(pw.all.kde, percent = 50, unin = "m", unout = "km2")
#Plot it:
plot(pw.all.50.kde)
#Add axes if you want
```

```
#The Kernel Density Estimates overlap with land!
#To remove land_from your 95% kde, first read in the shapefile of the U
S (e.g"NOS80k.shp") using the readOGR function:
us <- readOGR("NOS80k.shp")

#Transform/reproject to WGS84, so that it has the same CRS as everythin
g else:
us.pro <- spTransform(us,CRS("+init=epsg:4326"))


#Remove land using the gdifference function:
pw.all.95.fkde <- gDifference(pw.all.95.kde, us.pro, byid=TRUE, drop_lo
wer_td=TRUE)
#Turn the new kde into a spatial polygon data frame, so that it can be
outputted as a shapefile:
pw.all.95.fkde <- as(pw.all.95.fkde, "SpatialPolygonsDataFrame")

#To remove land from your 50% kde, use the gdifference function again:
pw.all.50.fkde <- gDifference(pw.all.50.kde, us.pro, byid=TRUE, drop_lo
wer_td=TRUE)
#Turn the new kd into a spatial polygon data frame:
pw.all.50.fkde <- as(pw.all.50.fkde, "SpatialPolygonsDataFrame")

#You can also do this QGIS using the Difference Vector Geoprocessing to
ol (although it can be glitchy)

#Write your new 95% and 50% KDE into a shapefile that can be brought in
to QGIS using the writeOGR function:
#95% KDE shapefile:
writeOGR(pw.all.95.fkde, dsn = ".", layer = "PilotWhale95KDE_Combined_F
inal", driver="ESRI Shapefile")

#50% KDE shapefile:
writeOGR(pw.all.50.fkde, dsn = ".", layer = "PilotWhale50KDE_Combined_F
inal", driver="ESRI Shapefile")

#Congrats, you've used R to map and estimate MCP and KDE for pilot whal
es!
```

```
IV. Repeat to get KDE and MCP for beaked whales (optional)
```

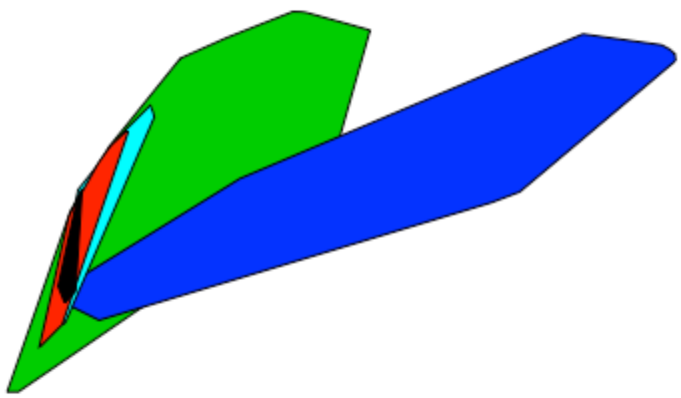
Estimating KDE and MCP for Individual Pilot Whales (optional):

```
#Create another variable for converting the longitude and latitude into a spatial data frame so that you can look each individual's home range:
pw.spdf <- pwdat

#Use the SpatialPointsDataFrame function in the rgdal package to have R recognize the columns lon and lat as geographic coordinates:
pw.spdf <- SpatialPointsDataFrame(coords=(cbind(pw.spdf$lon, pw.spdf$lat)), data = pw.spdf["id"], proj4string = CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"))

#Use spTransform to make sure that the data is properly projected (e.g. CRS: WGS84)
pw.spdf <- spTransform(pw.spdf, CRS("+init=epsg:4326"))

#Take a look at the locations by using the plot function:
plot(pw.spdf, col = pw.spdf$id)
#Add the lat and long axes:
axis(1, tcl = .3)
axis(2, las = 1, tcl = .3)
axis(3, labels = NA, tcl = .3)
axis(4, labels = NA, tcl = .3)
box()
```



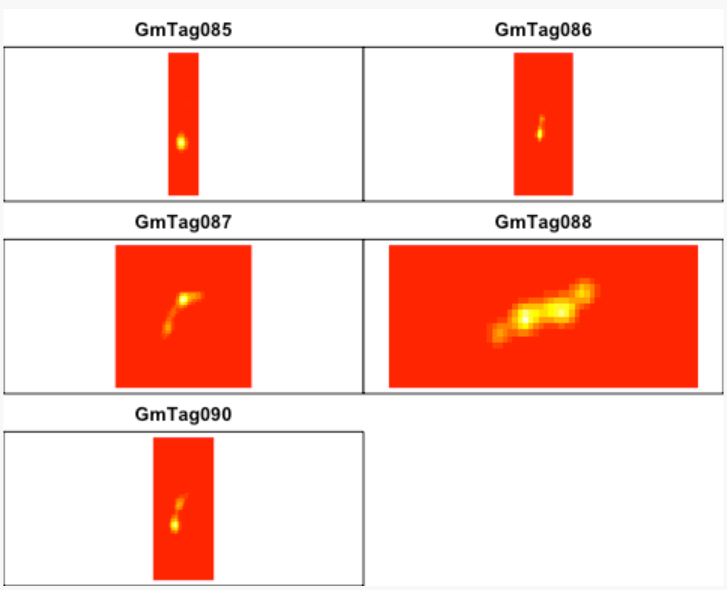

```
#To look at one individual, such as GmTag085, create a new variable isolating that indivdiual:
gm85 <- pw.spdf[pw.spdf$id=="GmTag085", ]
plot(gm85)
```

```
#And add axes using the code above

#Now, you can calculate the minimum convex polygon (100%) for each indi
vidual using the mcp function:
pw.mcp <- mcp(pw.spdf[, "id"], percent = 100, unin = "m", unout = "km2"
)
#Plot your MCP
plot(pw.mcp, col = c(1:5))
```



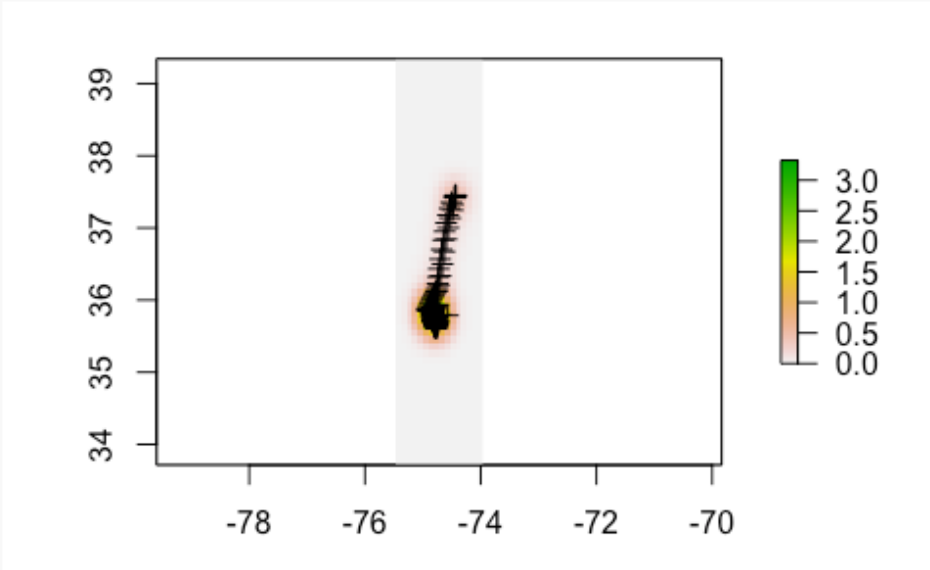
```
#Add lat and lon axes using the above code
#Plot one animal at a time:
plot(pw.mcp[1,]) #1 represents individual "GmTag085", #2 represents ind
ividual "GmTag086", etc.

#For home range MCP you want to use 95% (instead of 100%):
pw.95.mcp <- mcp(pw.spdf[, "id"], percent = 95, unin = "m", unout = "km
2")
#Use the code above for plotting and looking at one individual at a tim
e

#For core range MCP you want to use 50% (instead of 100%):
pw.50.mcp <- mcp(pw.spdf[, "id"], percent = 50, unin = "m", unout = "km
2")
#Use the code above for plotting and looking at one individual at a tim
e
```

```r
#You can look at the area for each individual's 100% MCP using:
as.data.frame(pw.mcp)

##                  id         area
## GmTag085 GmTag085 3.924514e-07
## GmTag086 GmTag086 1.531439e-06
## GmTag087 GmTag087 1.974954e-05
## GmTag088 GmTag088 1.940682e-05
## GmTag090 GmTag090 2.012345e-06

#Do the same for the 95% and 50% MCP by using the appropriate variables
.

#Write your MCP into a shapefile that can be brought into QGIS using th
e writeOGR function:
writeOGR(pw.95.mcp, dsn = ".", layer = "PilotWhale95MCP", driver="ESRI
Shapefile")
#dsn "." will output the file in your working directory folder, layer i
s the file name, driver is the type of file that is being made
#Repeat for the 50% MCP

#Now, estimate each individual's kernel desnity using the kernelUD func
tion:
pw.kde <- kernelUD(pw.spdf[,"id"], h="href")
#Look at the kernel density estimates:
image(pw.kde)
```




```r
#Turn the kde image into a raster file for each individual, such as GMT
ag085:
gm85.rast <- (raster(as(pw.kde$GmTag085, "SpatialPixelsDataFrame")))
#Look at GMTag085's kernel density with it's points overlayed:
```

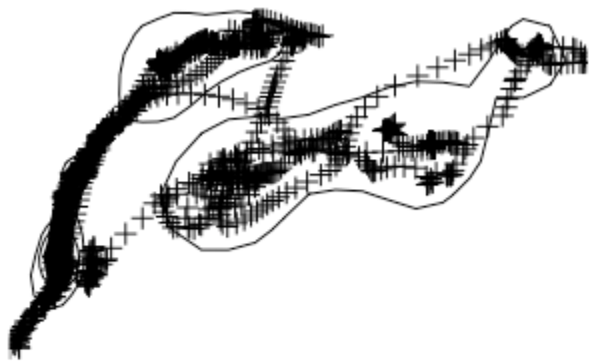
```r
plot(gm85.rast)
plot(gm85, add=T) #remember that you created variable gm85 above
```




```r
#If you want to export the graph, you can select Export in the Plots wi
ndow.
#Output the kernel density as a rasterized geo tiff using the writeRast
er function:
writeRaster(gm85.rast, "gm85.tif")

#Let's continue estimating KDE home range (95%) and core range (50%) fo
r each indivdiual using the getveticeshr function:
#Home range (95%):
pw.95.kde <- getverticeshr(pw.kde, percent = 95)
#Plot it:
plot(pw.95.kde)
#Overlay the track points:
plot(pw.spdf, add=T)
```

```r
#If you want, you can also add the rasters of kde
#Assign axes for lat and lon using the code above
```

```r
#Core range (50%):
pw.50.kde <- getverticeshr(pw.kde, percent = 50)
#Plot it:
plot(pw.50.kde)
#Overlay the track points:
plot(pw.spdf, add=T)
```

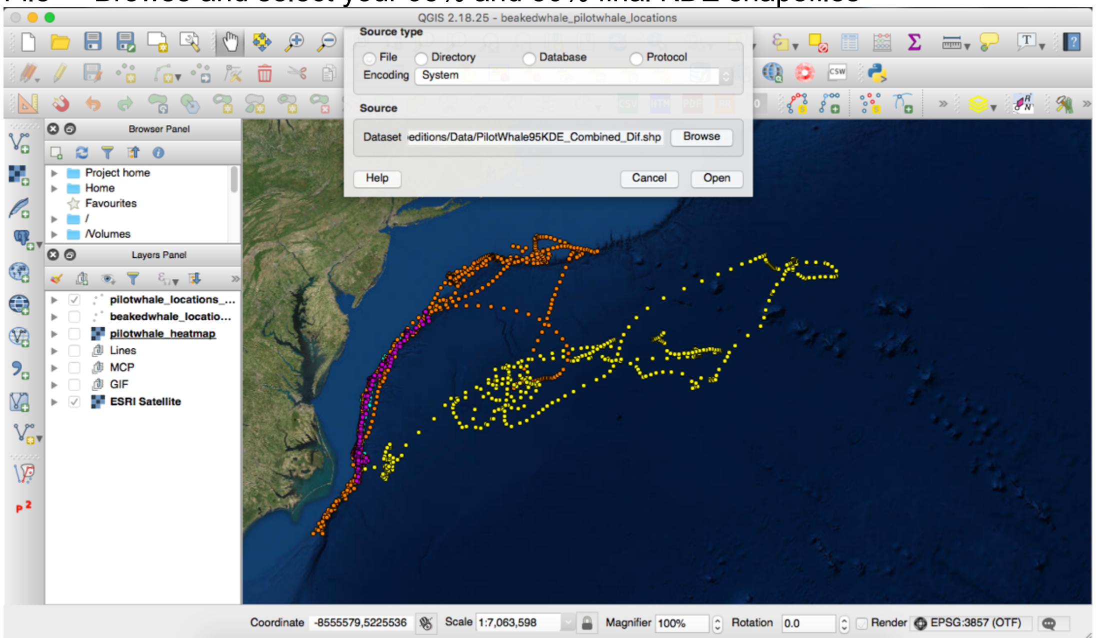


```r
#If you want, you can also add the rasters of kde
#Assign axes for lat and lon using the code above

#Don't forget to remove land from the kernel density estimates using
the gdifference function!
#You already read in the and transformed the shapefile of the US, so
you can skip that step.
#To remove land from your 95% kde, use the gdifference function again:
pw.95.fkde <- gDifference(pw.95.kde, us.pro, byid=TRUE,
drop_lower_td=TRUE)
#Turn the new kd into a spatial polygon data frame:
pw.95.fkde <- as(pw.95.fkde, "SpatialPolygonsDataFrame")

#To remove land from your 50% kde, use the gdifference function again:
pw.50.fkde <- gDifference(pw.50.kde, us.pro, byid=TRUE,
drop_lower_td=TRUE)
#Turn the new kd into a spatial polygon data frame:
pw.50.fkde <- as(pw.50.fkde, "SpatialPolygonsDataFrame")

#Write your new 95% and 50% KDE into a shapefile that can be brought
into QGIS using the writeOGR function:
#95% KDE shapefile:
writeOGR(pw.95.fkde, dsn = ".", layer = "PilotWhale95KDE_Final",
driver="ESRI Shapefile")
#50% KDE shapefile:
writeOGR(pw.50.fkde, dsn = ".", layer = "PilotWhale50KDE_Final",
driver="ESRI Shapefile")
```

```
#Congrats, you've used R to map and estimate MCP and KDE for each individual pilot whale!
#Now, repeat for the beaked whales!
```

C. Import Vector Layers in QGIS:

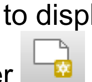
1. Select the Layer drop down menu -> Add layer -> Add Vector Layer… -> Source Type: File -> Browse and select your 95% and 50% final KDE shapefiles



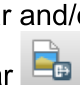
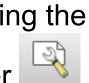
2. Save the layer: Right click the layer in the Layers Panel -> Save as… -> Save in your QGIS folder

IV. Making Maps in QGIS

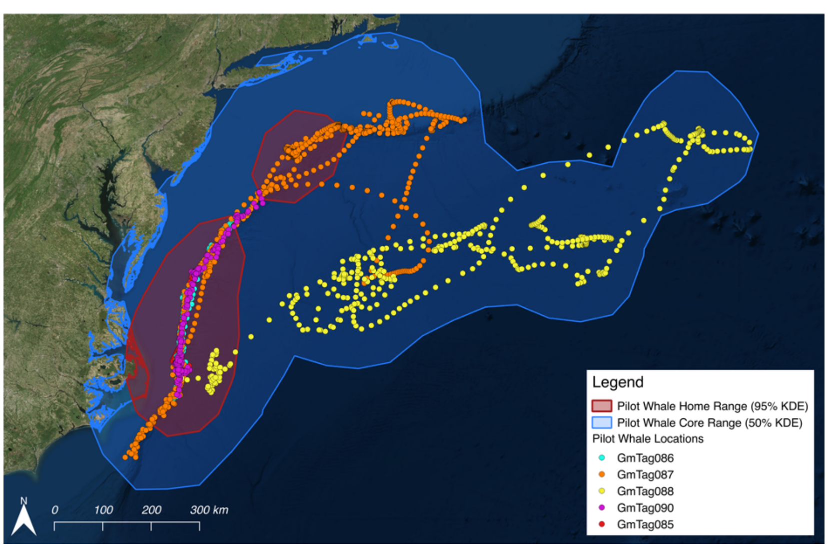
A. Map Making with the Print Composer:

1. Select the layers you want to display in your map
2. Select New Print Composer  in the Project Toolbar -> Enter a Composer Title (such as pilotwhale_homerange) -> The Print Composer window will open
3. Select Add new map  in the left toolbar-> drag a rectangle on the canvas -> the map will appear and you will now see Map 0 in Items panel on the right
    a. Note: You can change the names of the items by double clicking the item
4. Panning: Select Move item content  in the left toolbar -> Pan the map by clicking and dragging
5. Adjusting Scale: Select Map 0 in the Items panel on the right -> Go to the Item properties panel -> Main Properties -> Adjust the Scale value
6. Add a legend: Select Add new legend  in the toolbar -> Click and drag a legend on the canvas -> Select Legend in the Items panel -> In the Item properties panel below expand the Legend items and deselect Auto update -> Select Filter Legend By Map Content  to only display the active layers -> select the ESRI Satellite layer and

Delete  (the basemap layer isn't necessary to display) -> Change the name of layers so that they are more intuitive by selecting the layer you want to rename and Edit  -> Adjust the font size, color and fill color using the Item properties panel

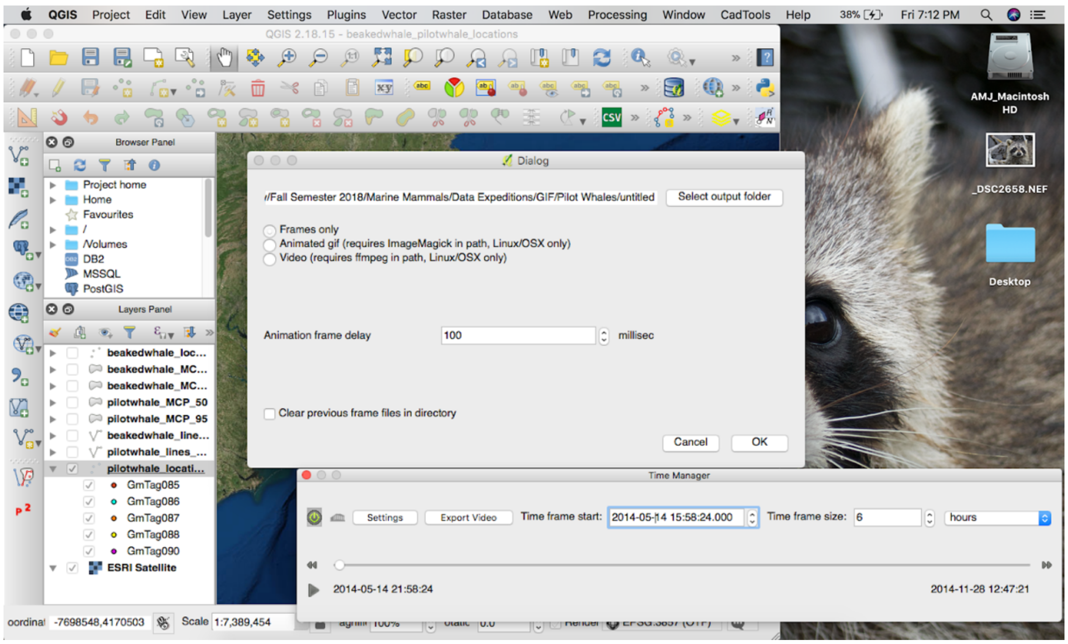
7. Move item and Adjust Size: Select Select/Move item  in the toolbar -> Select the item you want to move or adjust the size of
8. Add a Scalebar: Select Add new scalebar  in the toolbar -> Click and drag a scalebar on the canvas -> Adjust the style, font color, etc by selecting the <scale bar> item in the Items panel and changing the properties in the Item Properties panel below -> Try changing the Style in Main Properties and the various fonts in Font and Colors as well as the # of km
9. Add an N Arrow: Select Add image  in the toolbar -> Click and drag a rectangle on the canvas -> Select <picture> in the Item panel -> In the Item properties panel below expand the select search directories -> Select  -> Expand the SVG Parameters in the Item Properties -> Change the Outline color to white -> Adjust the size
10. Add Text/Map Title: Select Add new label in the toolbar -> Click and drag a rectangle on the canvas -> Select QGIS in the Item panel -> In the Item properties panel expand Main properties -> Delete QGIS and enter in your map title -> Adjust the size of the font by expanding Appearance in the Item properties and selecting Font
11. Save Template: Select Save project in the top toolbar and/or Save as template
12. Export Map: Select Export as image in the top toolbar  -> A resolution of 300 is good (especially if you intend to print the map), do not do less than 72 -> Save in selected location with your selected file name
13. Reopening the template after closing the print composer window: Select Composer Manager  in the project toolbar -> Select the saved template -> The Print Composer window will open with the template -> Select the map layer in Items -> Select update preview in Item properties

Pilot Whale Home Range and Core Range



B. Making a GIF (Optional):
Being able to communicate your data to a broader audience is critical. Creating an animated map is an excellent way to visualize time series data and make it accessible to many audiences.

1. See II. Visualize the Data in QGIS, C. Time Manager
2. Adjust the dimensions of your map: Separate the time manager panel so that it's a separate window -> adjust the dimension of QGIS, what you see will be outputted into hundreds of .png files
3. Export the images: Select Export Video -> Select output folder (I highly recommend making one folder for all of the png files - mine was called Pilot Whales) -> Select Frames only -> Select OK -> To stop recording/outputting frames select the green power button



4. Use Photoshop or another program of your choice to create the GIF
   a. Here is the link for photoshop: https://helpx.adobe.com/photoshop/how-to/make-animated-gif.html