```
#--------------------------------
#Set working directory to ensure R can find the files we wish to import.
setwd("C:/Users/aalda/Desktop/All plots 2018")

#Installing and Loading all packages-------------

#install.packages("RStoolbox")
#install.packages("rasterVis")
#install.packages('raster')
#install.packages('gdalUtils')
#install.packages('tidyverse')
#install.packages('rgr')
#install.packages('uavRst')
#install.packages('rgdal') -Alex

library('gdalUtils')
library('RStoolbox')
library('rasterVis')
library('raster')
library('ggplot2')
library('rgr')
library('tidyverse')
library('rgdal') #alex wrote this.
#library('uavRst') #may not have been used.

# Load Data ---------------------------------------------------------------
#Raster

                   # We will create a character vector list of raster files using the list.files() function in the directory
named "All plots 2018". This list will be used to generate a Rasterstack.
files <- list.files()
files              #See the list of all files in the directory named "All plots 2018".
dbf.files <- files[grep(".tif", files, fixed=T)]        #Creates a file that is list of names only having .tif extensions. Grep
function finds ".tiff" pattern in the created "files" and fixed=T means pattern is a text string.
for(i in dbf.files) { assign(unlist(strsplit(i, "[.]"))[1], raster(i)) }    #

BH1_RGB<-stack("BH-01 RGB_modified.tif")       # (1) Import multi-band raster data, using the stack() function.
BH1_IR<-raster('BH-01 IR_modified.tif')        # (1) Import and create a Rasterlayer file using the raster function.
BL2_RGB<-stack("BL-02 RGB.tif")  #(2)
BL2_IR<-raster('BL-02 IR.tif')   #(2)
BH2_RGB<-stack("BH-02 RGB.tif")  #(3)
BH2_IR<-raster('BH-02 IR.tif')   #(3)
LGH1_RGB<-stack("LGH1-RGB_1-20_modified.tif")  #(4)
LGH1_IR<-raster('LGH1-IR_1-20_modified.tif')   #(4)
LGH2_RGB<-stack("LGH-2 RGB_modified.tif")      #(5)
LGH2_IR<-raster('LGH-2 IR_modified.tif')       #(5)
LGL1_RGB<-stack("LGL-1 RGB 1-20_modified.tif") #(6)
LGL1_IR<-raster('LG-L1 IR 1-20_modified.tif')  #(6)
LGL2_RGB<-stack("LGL-2-RGB_modified_2.tif")    #(7)
LGL2_IR<-raster('LGL-2 IR-modified.tif')       #(7)
OGH1_RGB<-stack("OGH-1 RGB2_modified.tif")     #(8)
OGH1_IR<-raster('OG-H2 IR_modified.tif')       #(8)
OGH2_RGB<-stack("OG-H2 RGB_modified.tif")      #(9)
OGH2_IR<-raster('OG-H2 IR_modified.tif')       #(9)



# (1) BH-01 ---------------------------------------------------------------


#align extent


BH1_IR_proj<-projectRaster(BH1_IR, BH1_RGB) #Alex: Project the data of a Raster object to a new RAster object with another
projection (crs). projectRaster(from, to).


#Shapefile

plot(BH1_IR_proj)                    #press-clear broom stick in plots tab, if you get a error that margins are too large.
Alex+ani: line 58-Error-plot.new has not been called yet, occured becuase plot name has not been called.
polygon<-shapefile("Polygons.shp")   #Alex: File format for storing geospatial data in polygon.shp.
plot(polygon, add=TRUE)              #Alex: This adds another raster ontop of another. This draws the boundary of the two
exclosures over the [BH1_IR_proj] image
BH1_shp_ug<-subset(polygon, PlotID=='BH-01 UG') #Ungrazed  #Alex: Returns (selected variables) subsets of vectors, matirces or
dataframes which meet conditions. Subset(object to be subsetted, logical expression indicating elements or rows to keep).
BH1_shp_g<-subset(polygon, PlotID=='BH-01 G')#Grazed

# BH1: Mask and clip rasters to polygon -------------------------------------------------

#Ungrazed

BH1_RGB_mask<-mask(BH1_RGB, BH1_shp_ug)
BH1_IR_mask<-mask(BH1_IR_proj, BH1_shp_ug)


BH1RGB_crop<- crop(BH1_RGB_mask, BH1_shp_ug)
plot(BH1RGB_crop)            #Alex: four images produced.(#1-image)
ex<-extent(BH1RGB_crop)

BH1IR_crop<- crop(BH1_IR_mask, ex)
plot(BH1IR_crop)             #Alex: one image produced.(#2-image)


#BH1-Ungrazed: Stack and Brick IR and RGB -------------------------------------------------------
```

```
BH1_stack<-stack(BH1RGB_crop, BH1IR_crop)
nlayers(BH1_stack)
BH1_stack<-writeRaster(BH1_stack, filename="C:/Users/aalda/Desktop/All plots 2018/BH1_Stack.tif", format="GTiff", overwrite=TRUE)




#BH1-Ungrazed: Calculate NDVI of Ungrazed ----------------------------------------------------------

BH1_ndvi_ungrazed<-((BH1_stack[[5]]-BH1_stack[[1]])/(BH1_stack[[5]]+BH1_stack[[1]]))
plot(BH1_ndvi_ungrazed)         #Alex: one image produced.(#3-image)
hist(BH1_ndvi_ungrazed)         #Alex: histogram produced. (#4-image)



# Do it all again for BH1 Grazed ----------------------------------------

#BH1: Mask and clip rasters to polygon --------------------------------------------------

#BH1-Grazed: Grazed

BH1_RGB_mask<-mask(BH1_RGB, BH1_shp_g)
BH1_IR_mask<-mask(BH1_IR_proj, BH1_shp_g)


BH1RGB_crop<- crop(BH1_RGB_mask, BH1_shp_g)
plot(BH1RGB_crop)          #Alex: four images produced. (#5-image)
ex<-extent(BH1RGB_crop)

BH1IR_crop<- crop(BH1_IR_mask, ex)
plot(BH1IR_crop)           #Alex: one image produced.(#6-image)


#BH1-Grazed: Stack and Brick IR and RGB --------------------------------------------------------

BH1_stack<-stack(BH1RGB_crop, BH1IR_crop)
nlayers(BH1_stack)          #Alex: there are 5 layers.

#BH1-Grazed: Calculate NDVI ---------------------------------------------------------

BH1_ndvi_grazed<-((BH1_stack[[5]]-BH1_stack[[1]])/(BH1_stack[[5]]+BH1_stack[[1]]))
plot(BH1_ndvi_grazed)           #Alex: one image produced. (#7-image)
BL2_IR_proj<-projectRaster(BL2_IR, BL2_RGB) #Alex: copied and pasted here.
BL2_shp_g<-subset(polygon, PlotID=='BL-02 G')#Grazed -Alex copied and pasted here.
BL2_IR_mask<-mask(BL2_IR_proj, BL2_shp_g)    #Alex: first run line 181 for BL2_shp_g and line 185 for BL2_IR_proj.


BL2_RGB_mask<-mask(BL2_RGB, BL2_shp_g)    #Alex: copieds line 229 and pasted it here
BL2_shp_g<-subset(polygon, PlotID=='BL-02 G')#Grazed   #Alex: copied and pasted it here
BL2RGB_crop<- crop(BL2_RGB_mask, BL2_shp_g)   #Alex: first run line 229 (BL2_RGB_mask<-mask(BL2_RGB, BL2_shp_g). Problem: Error
in file(fn, "rb") : cannot open the connection. Solution: run line 72 and line 16 (library (raster)).
plot(BL2RGB_crop)                              #Alex: four images in the shape of parallelgrams.(#8 image)
ex<-extent(BL2RGB_crop)

BL2IR_crop<- crop(BL2_IR_mask, ex)
plot(BL2IR_crop)                              #Alex: one image produce in the shape of parallelgram. (#9-image)


# Stack and Brick IR and RGB --------------------------------------------------------

BL2_stack<-stack(BL2RGB_crop, BL2IR_crop)
nlayers(BL2_stack)

# Calculate NDVI --------------------------------------------------------------

BL2_ndvi_grazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]]))
plot(BL2_ndvi_grazed)                           #Alex: one image produced in the shape of a parallelgram.(#10 image)
hist(BL2_ndvi_grazed)                           #Alex: histogram produced. (#11 image)

#compare to ungrazed
BL2_ndvi_ungrazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]]))  #alex:copied/pasted here.
BL2_ndvi_ungrazed<-as.data.frame(BL2_ndvi_ungrazed) #Alex: First run line 217 for BL2_ndvi_ungrazed which is (
BL2_ndvi_ungrazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]])) .
BL2_ndvi_grazed<-as.data.frame(BL2_ndvi_grazed)

BH_01_g<-tibble(
  Value=BL2_ndvi_grazed$layer,
  Treatment="Grazed"
)

BH_01_ug<-tibble(
  Value=BL2_ndvi_ungrazed$layer,
  Treatment="Ungrazed"
)

BH_01<-rbind(BH_01_g, BH_01_ug)
ggplot(data=BH_01, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')                      #Alex: two images produced.(#12 image)
hist(BH1_ndvi_grazed)                            #Alex: histogram produced. (#13 image)

#compare to ungrazed

BH1_ndvi_ungrazed<-as.data.frame(BH1_ndvi_ungrazed)
BH1_ndvi_grazed<-as.data.frame(BH1_ndvi_grazed)

BH_01_g<-tibble(
  Value=BH1_ndvi_grazed$layer,
  Treatment="Grazed"
)

BH_01_ug<-tibble(
  Value=BH1_ndvi_ungrazed$layer,
  Treatment="Ungrazed"
)

BH_01<-rbind(BH_01_g, BH_01_ug)
ggplot(data=BH_01, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')                          #Alex: two images produced. (#14 image)




# (2) BL-02 -------------------------------------------------------------------


#align extent

BL2_IR_proj<-projectRaster(BL2_IR, BL2_RGB)    #Alex: two images produced. (#15 image)

#Shapefile

plot(BL2_IR_proj)                          #Alex created this line.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)
BL2_shp_ug<-subset(polygon, PlotID=='BL-02 UG') #Ungrazed
BL2_shp_g<-subset(polygon, PlotID=='BL-02 G') #Grazed


# Wrong: BL2_shp_ug<-subset(polygon, PlotID=='BH-01 UG') #Ungrazed     #Alex: warming messages.
#wrong: BL2_shp_g<-subset(polygon, PlotID=='BH-01 G')#Grazed

# Mask and clip rasters to polygon ---------------------------------------------------

#Ungrazed

BL2_RGB_mask<-mask(BL2_RGB, BL2_shp_ug)
BL2_IR_mask<-mask(BL2_IR_proj, BL2_shp_ug)


BL2RGB_crop<- crop(BL2_RGB_mask, BL2_shp_ug)    #Alex: first run line 191 for BL2_shp_ug. Problem: Error in .local(x, y, ...) :
extents do not overlap
plot(BL2RGB_crop)                 #Alex: four images produced in shape of parallelgrams. (#16 image)
ex<-extent(BL2RGB_crop)

BL2IR_crop<- crop(BL2_IR_mask, ex)
plot(BL2IR_crop)                    #Alex: (check) one image not produced. (#17 image)


# Stack and Brick IR and RGB ---------------------------------------------------------

BL2_stack<-stack(BL2RGB_crop, BL2IR_crop)
nlayers(BL2_stack)                           #Alex: 5 layers.

# Calculate NDVI ---------------------------------------------------------------

BL2_ndvi_ungrazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]]))
plot(BL2_ndvi_ungrazed)                 #Alex: (check) one image produced. (#18 image)
hist(BL2_ndvi_ungrazed)                 #Alex: (check) histogram produced. (#19 image)



# Do it all again for BL2 Grazed -----------------------------------------------

# Mask and clip rasters to polygon ---------------------------------------------------

#Grazed

BL2_RGB_mask<-mask(BL2_RGB, BL2_shp_g)
BL2_IR_mask<-mask(BL2_IR_proj, BL2_shp_g)     #Ales wrote this line.

#pasted here.

BL2RGB_crop<- crop(BL2_RGB_mask, BL2_shp_g)   #Alex: first run line 229 (BL2_RGB_mask<-mask(BL2_RGB, BL2_shp_g). Problem: Error
in file(fn, "rb") : cannot open the connection. Solution: run line 72 and line 16 (library (raster)).
plot(BL2RGB_crop)                              #Alex: four images in the shape of parallelgrams.(#8 image)
ex<-extent(BL2RGB_crop)

BL2IR_crop<- crop(BL2_IR_mask, ex)
plot(BL2IR_crop)                              #Alex: one image produce in the shape of parallelgram. (#9-image)


# Stack and Brick IR and RGB ---------------------------------------------------------

BL2_stack<-stack(BL2RGB_crop, BL2IR_crop)
nlayers(BL2_stack)

# Calculate NDVI ---------------------------------------------------------------

BL2_ndvi_grazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]]))
plot(BL2_ndvi_grazed)                           #Alex: one image produced in the shape of a parallelgram.(#10 image)
hist(BL2_ndvi_grazed)                           #Alex: histogram produced. (#11 image)

#compare to ungrazed
```

```
BL2_ndvi_ungrazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]]))  #alex:copied/pasted here.
BL2_ndvi_ungrazed<-as.data.frame(BL2_ndvi_ungrazed) #Alex: First run line 217 for BL2_ndvi_ungrazed which is (
BL2_ndvi_ungrazed<-((BL2_stack[[5]]-BL2_stack[[1]])/(BL2_stack[[5]]+BL2_stack[[1]])) .
BL2_ndvi_grazed<-as.data.frame(BL2_ndvi_grazed)

#alex wrote this
BL_02_g<-tibble(
  Value= BL2_ndvi_ungrazed$layer,
  Treatment="Grazed"
)

BL_02_ug<-tibble(
  Value=BL2_ndvi_ungrazed$layer,
  Treatment= "Ungrazed"
)

BL_02<-rbind(BL_02_g, BL_02_ug)
ggplot(data= BL_02, aes (x= Treatment, y=Value))+
  geom_violin(scale='area')
hist(BL2_ndvi_grazed)    # Error in hist.default(BL2_ndvi_grazed) : 'x' must be numeric. Image#


#(3) BH-02 ----------------------------------------------------------------------------------------

#BH2_RGB<-stack("BH-02 RGB.tif")
#BH2_IR<-raster('BH-02 IR.tif')


#Align Extent  -make the projection of image.name_IR and image.name_RGB the same.

projection(BH2_IR)     #Projection of BH2_IR
projection(BH2_RGB)    #Projection of BH2_RGB
crs(BH2_IR)<-'+proj=utm +zone=10 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0'   #Assign the projection of BH2_RGB
to BH2_IR.
projection(BH2_IR)      #This shows that the crs of BH2_IR now has the same crs as BH2_RGB.
BH2_IR_proj<-BH2_IR                  #Conformation of changed projection.


#Shapefile

plot(BH2_IR_proj)                                  #single image produced.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)                            #Error: No boundaries displayed and no image with boundary produced.
BH2_shp_ug<- subset(polygon, PlotID == 'BH-02 UG')
BH2_shp_g<-subset(polygon, PlotID == 'BH-02_G')


#Mask and clip raster to polygon ---------------------------------

#BH2: UNGRAZED

length(BH2_RGB)
length(BH2_shp_ug)
BH2_RGB_mask<-mask(BH2_RGB, BH2_shp_ug)       #Error in x@polygons[[i]] : subscript out of bounds
BH2_IR_mask<-mask(BH2_IR_proj, BH2_shp_ug)    #Error in x@polygons[[i]] : subscript out of bounds

BH2RGB_crop<-crop(BH2_RGB_mask, BH2_shp_ug)
plot(BH2RGB_crop)
ex<-extent(BH2RGB_crop)

BH2IR_crop<-crop(BH2_IR_mask, ex)
plot(BH2IR_crop)

#BH2- Stack and Brick IR and RGB ----------------------------------

BH2_stack<-stack(BH2RGB_crop, BH2IR_crop)
nlayers(BH2_stack)

#BH2- Calculate NDVI for ungrazed ---------------------------------

BH2_ndvi_ungrazed<-((BH2_stack[[5]]-BH2_stack[[1]]) / (BH2_stack[[5]]+BH2_stack[[1]]))
Plot(BH2_ndvi_ungrazed)
hist(BH2_ndvi_ungrazed)

# Do it all again for Grazed ----------------------------------------------

#BH2: GRAZED -------------------------------------------------------

BH2_RGB_mask<-mask(BH2_RGB, BH2_shp_g)
BH2_IR_mask<-mask(BH2_IR_proj, BH2_shp_g)

BH2_RGB_crop<-crop(BH2_RGB_mask, BH2_shp_g)
plot(BH2RGB_crop)
ex<-extent(BH2RGB_crop)

BH2IR_crop<-crop(BH2_IR_mask, ex)
plot(BH2IR_crop)

#BH2- Stack and brick IR and RGB ----------------------------------

BH2_stack<-stack(BH2RGB_crop, BH2IR_crop)
nlayers(BH2_stack)

#BH2- Calculate NDVI for Grazed -----------------------------------

BH2_ndvi_grazed<-((BH2_stack[[5]]-BH2_stack[[1]])/(BH2_stack[[5]]+BH2_stack[[1]]))
plot(BH2_ndvi_grazed)
```

```
#BH2: compare grazed to ungrazed -----------------------------------

BH2_ndvi_ungrazed<-as.data.frame(BH2_ndvi_ungrazed)
BH2_ndvi_grazed<-as.data.frame(BH2_ndvi_grazed)

BH_02_g<-tibble(
  Value=BH2_ndvi_grazed$layer,
  Treatment="Grazed"
)

BH_02_ug<-tibble(
  Value=BH2_ndvi_ungrazed$layer,
  Treatment= "Ungrazed"
)

BH_02<-rbind(BH_02_g, BH_02_ug)
ggplot(data=BH_02, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')
hist(BH2_ndvi_grazed)




#(4) LGH-01 -----------------------------------------------------------------------------

#LGH1_RGB<-stack("LGH1-RGB_1-20_modified.tif")
#LGH1_IR<-raster('LGH1-IR_1-20_modified.tif')


#LGH1- Align Extent -------------------------------

LGH1_IR_proj<-projectRaster(LGH1_IR, LGH1_RGB)

#Shapefile ----------------------------------------

plot(LGH1_IR_proj)                                                            #single image produced.
polygon<-shapefile('C:/Users/aalda/Desktop/All plots 2018/Polygons.shp')
length(polygon)
plot(polygon, add=TRUE)                                                       #image with boundaries plotted.
#poly<-readOGR('C:/Users/aalda/Desktop/All plots 2018/Polygons.shp')
#poly@data


LGH1_shp_ug<-subset(polygon, PlotID =='LGH1 UG')  # Ungrazed
LGH1_shp_g<-subset(polygon, PlotID == 'LGH1 G')   #Grazed

#LGH1 - Mask and clip raster to polygon ----------------

#LGH1: Ungrazed

LGH1_RGB_mask<-mask(LGH1_RGB, LGH1_shp_ug)    # Error in x@polygons[[i]] : subscript out of bounds
LGH1_IR_mask<-mask(LGH1_IR_proj, LGH1_shp_ug)  # same error.

LGH1_RGB_crop<-crop(LGH1_RGB_mask, LGH_shp_ug)
plot(LGH1RGB_crop)
ex<-extent(LGH1RGB_crop)

#LGH1: Stack and brick IR and RGB --------------------------

LGH1_stack<-stack(LGH1RGB_crop, LGH1IR_crop)
nlayers(LGH1_stack)

#LGH1: Calculate NDVI--------------------------------------

LGH1_ndvi_ungrazed<-((LGH1_stack[[5]]-LGH11_stack[[1]])/(LGH1_stack[[5]]+LGH1_stack[[1]]))
plot(LGH1_ndvi_ungrazed)
hist(LGH1_ndvi_ungrazed)

# Do it all again for LGH1 Grazed --------------------

#LGH1: Grazed

LGH1_RGB_mask<-mask(LGH1_RGB_mask, LGH1_shp_g)
LGH1_IR_mask<-mask(LGH1_IR_proj, LGH1_shp_g)

LGH1RGB_crop<-crop(LGH1_RGB_mask, LGH1_shp_g)
plot(LGH1RGB_crop)
ex<-extent(LGH1RGB_crop)

LGH

#LGH2: Stack and brick IR and RGB -----------------------

LGH2_stack<-stack(LGH2RGB_crop, LGH2IR_crop)

LGH1IR_crop<-crop(LGH1_IR_mask, ex)
plot(LGH1IR_crop)


#LGH1: Stack and brick IR and RGB --------------------

LGH1_stack<-stack(LGH1RGB_crop, LGH1IR_crop)
nlayers(LGH1_stack)


#LGH1: Calculate NDVI for Grazed --------------------
```

```
LGH1_ndvi_grazed<-((LGH1_stack[[5]]-LGH1_stack[[1]])/(LGH1_stack[[5]]+LGH1_stack[[1]]))
plot(LGH1_ndvi_grazed)


#LGH1: Compare Grazed to Ungrazed -------------------

LGH1_ndvi_ungrazed<-as.data.frame(LGH1_ndvi_ungrazed)
LGH1_ndvi_grazed<-as.data.frame(LGH1_ndvi_grazed)

LGH_01_g<-tibble(
  Value= LGH1_ndvi_grazed$layer,
  Treatment='Grazed'
)

LGH1_01_ug<-tibble(
  Value=LGH1_ndvi_ungrazed$layer,
  Treatment='Ungrazed'
)

LGH_01<-rbind(LGH_01_g, LGH_01_ug)
 ggplot(data=LGH_01, aes(x=Treatment, y=Value))+
 geom_violin(scale='area')
hist(LGH1_ndvi_grazed)




#(5) LGH-02 --------------------------------------------------------


#LGH2_RGB<-stack("LGH-2 RGB_modified.tif")
#LGH2_IR<-raster('LGH-2 IR_modified.tif')


# Align Extent--------------------------

LGH2_IR_proj<-projectRaster(LGH2_IR, LGH2_RGB)


# Shapefile ----------------------------

plot(LGH2_IR_proj)                                      #image of plot produced.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)                                 #image of plot and boundaried drawn.

LGH2_shp_ug<-subset(polygon, PlotID=='LGH-02 UG')  #Ungrazed
LGH2_shp_g<-subset(polygon, PlotID == 'LGH-02 G')  # Grazed


# LGH2: Mask and clip raster to polygon ------------

#LGH2: Ungrazed

LGH2RGB_mask<-mask(LGH2_RGB, LGH2_shp_ug)                 #Error in x@polygons[[i]] : subscript out of bounds
LGH2_IR_mask<-mask(LGH2_IR_proj, LGH2_shp_g)

LGH2RGB_crop<-crop(LGH2_RGB_mask, LGH2_shp_ug)
plot(LGH2RGB_crop)
ex<-extent(LGH2RGB_crop)

LGH2IR_crop<-crop(LGH2_IR_mask, ex)
plot(LGH2IR_crop)


#LGH2: Stack and brick IR and RGB -----------------------

LGH2_stack<-stack(LGH2RGB_crop, LGH2IR_crop)
nlayers(LGH2_stack)


#LGH2: Calculate NDVI of the Ungrazed ------------------

LGH2_ndvi_ungrazed<-((LGH2_stack[[5]]-LGH2_stack[[1]])/(LGH2_stack[[5]]+LGH2_stack[[1]]))
plot(LGH2_ndvi_ungrazed)
hist(LGH2_ndvi_ungrazed)


# Do it all again for LGH2 Grazed ------------------------

#LGH2: Grazed

LGH2_RGB_mask<-mask(LGH2_RGB, LGH2_shp_g)
LgH2_IR_mask<-mask(LGH2_IR_proj, LGH2_shp_g)

LGH2RGB_crop<-crop(LGH2_RGB_mask, LGH2_shp_g)
plot(LGH2RGB_crop)
ex<-extent(LGH2RGB_crop)

LGH2IR_crop<-crop(LGH2_IR_mask, ex)
plot(LGH2IR_crop)


#LGH2: Stack and brick IR and RGB ----------------------

LGH2_stack<-stack(LGH2RGB_crop, LGH2IR_crop)
nlayers(LGH2_stack)
```

```
#LGH2: Calculate NDVI ----------------------------------

LGH2_ndvi_grazed<-((LGH2_stack[[5]]-LGH2_stack[[1]])/ (LGH2_stack[[5]]+LGH2_stack[[1]]))
plot(LGH2_ndvi_grazed)


#LGH2: Compare Grazed to Ungrazed -------------------

LGH2_ndvi_ungrazed<-as.data.frame(LGH2_ndvi_ungrazed)
LGH2_ndvi_grazed<-as.data.frame(LGH2_ndvi_grazed)

LGH_02_g<-tibble(
  Value= LGH2_ndvi_grazed$layer,
  Treatment='Grazed'
)

LGH1_02_ug<-tibble(
  Value=LGH2_ndvi_ungrazed$layer,
  Treatment='Ungrazed'
)

LGH_02<-rbind(LGH_02_g, LGH_02_ug)
ggplot(data=LGH_02, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')
hist(LGH2_ndvi_grazed)



#(6) LGL-01 ----------------------------------------------------------

#LGL1_RGB<-stack("LGL-1 RGB 1-20_modified.tif")
#LGL1_IR<-raster('LG-L1 IR 1-20_modified.tif')


# Align Extent--------------------------

LGL1_IR_proj<-projectRaster(LGL1_IR, LGL1_RGB)


# Shapefile ----------------------------

plot(LGL1_IR_proj)                                  #image of plot produced.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)                              #image of plot and boundary produced.

LGL1_shp_ug<-subset(polygon, PlotID=='LGL-01 UG')  #Ungrazed
LGL1_shp_g<-subset(polygon, PlotID == 'LGL-01 G')  # Grazed


#LGL1: Mask and clip raster to polygon ------------

#LGL1: Ungrazed

LGL1RGB_mask<-mask(LGL1_RGB, LGL1_shp_ug)                    #Error in x@polygons[[i]] : subscript out of bounds.
LGL1_IR_mask<-mask(LGL1_IR_proj, LGL1_shp_g)

LGL1RGB_crop<-crop(LGL1_RGB_mask, LGL1_shp_ug)
plot(LGL1RGB_crop)
ex<-extent(LGL1RGB_crop)

LGL1IR_crop<-crop(LGL1_IR_mask, ex)
plot(LGL1IR_crop)


#LGL1: Stack and brick IR and RGB -----------------------

LGL1_stack<-stack(LGL1RGB_crop, LGL1IR_crop)
nlayers(LGL1_stack)


#LGL1: Calculate NDVI of the Ungrazed ------------------

LGL1_ndvi_ungrazed<-((LGL1_stack[[5]]-LGL1_stack[[1]])/(LGL1_stack[[5]]+LGL1_stack[[1]]))
plot(LGL1_ndvi_ungrazed)
hist(LGL1_ndvi_ungrazed)


# Do it all again for LGH2 Grazed ------------------------

#LGH2: Grazed

LGH1_RGB_mask<-mask(LGH2_RGB, LGH2_shp_g)
LL_IR_mask<-mask(LGH2_IR_proj, LGH2_shp_g)

LGH2RGB_crop<-crop(LGH2_RGB_mask, LGH2_shp_g)
plot(LGH2RGB_crop)
ex<-extent(LGH2RGB_crop)

LGH2IR_crop<-crop(LGH2_IR_mask, ex)
plot(LGH2IR_crop)


#LGL1: Stack and brick IR and RGB ----------------------

LGL1_stack<-stack(LGL1RGB_crop, LGL1IR_crop)
```

```
nlayers(LGL1_stack)


#LGL1: Calculate NDVI ----------------------------------

LGH2_ndvi_grazed<-((LGH2_stack[[5]]-LGH2_stack[[1]])/ (LGH2_stack[[5]]+LGH2_stack[[1]]))
plot(LGH2_ndvi_grazed)


#LGL1: Compare Grazed to Ungrazed -------------------

LGL1_ndvi_ungrazed<-as.data.frame(LGL1_ndvi_ungrazed)
LGL1_ndvi_grazed<-as.data.frame(LGL1_ndvi_grazed)

LGL_01_g<-tibble(
  Value= LGL2_ndvi_grazed$layer,
  Treatment='Grazed'
)

LGL1_01_ug<-tibble(
  Value=LGL1_ndvi_ungrazed$layer,
  Treatment='Ungrazed'
)

LGL_01<-rbind(LGL_01_g, LGL_01_ug)
ggplot(data=LGL_01, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')
hist(LGL1_ndvi_grazed)




#(7) LGL-02 ----------------------------------------------------------

#LGL2_RGB<-stack("LGL-2-RGB_modified_2.tif")
#LGL2_IR<-raster('LGL-2 IR-modified.tif')


# Align Extent--------------------------

LGL2_IR_proj<-projectRaster(LGL2_IR, LGL2_RGB)


# Shapefile ----------------------------

plot(LGL2_IR_proj)                                     #image of plot produced.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)                                #image of plot and boundary produced.

LGL2_shp_ug<-subset(polygon, PlotID=='LGL-02 UG')  #Ungrazed
LGL2_shp_g<-subset(polygon, PlotID == 'LGL-02 G')  # Grazed


# LGL2: Mask and clip raster to polygon ------------

#LGL2: Ungrazed

LGL2_RGB_mask<-mask(LGL2_RGB, LGL2_shp_ug)           #Error in x@polygons[[i]] : subscript out of bounds
LGL2_IR_mask<-mask(LGL2_IR_proj, LGL2_shp_g)

LGL2RGB_crop<-crop(LGL2_RGB_mask, LGL2_shp_ug)
plot(LGL2RGB_crop)
ex<-extent(LGL2RGB_crop)

LGL2IR_crop<-crop(LGL2_IR_mask, ex)
plot(LGL2IR_crop)


####LGL2: Stack and brick IR and RGB -----------------------

LGL2_stack<-stack(LGL2RGB_crop, LGL2IR_crop)
nlayers(LGL2_stack)


#LGL2: Calculate NDVI of the Ungrazed ------------------

LGL2_ndvi_ungrazed<-((LGL2_stack[[5]]-LGL2_stack[[1]])/(LGL2_stack[[5]]+LGL2_stack[[1]]))
plot(LGL2_ndvi_ungrazed)
hist(LGL2_ndvi_ungrazed)


# Do it all again for LGH2 Grazed ------------------------

#LGL2: Grazed

LGL2_RGB_mask<-mask(LGL2_RGB, LGL2_shp_g)
LGL2_IR_mask<-mask(LGL2_IR_proj, LGL2_shp_g)

LGL2RGB_crop<-crop(LGL2_RGB_mask, LGL2_shp_g)
plot(LGL2RGB_crop)
ex<-extent(LGL2RGB_crop)

LGL2IR_crop<-crop(LGL2_IR_mask, ex)
plot(LGL2IR_crop)


#LGL2: Stack and brick IR and RGB ----------------------
```

```
LGL2_stack<-stack(LGL2RGB_crop, LGL2IR_crop)
nlayers(LGL2_stack)


#LGL2: Calculate NDVI ---------------------------------

LGL2_ndvi_grazed<-((LGL2_stack[[5]]-LGL2_stack[[1]])/ (LGL2_stack[[5]]+LGL2_stack[[1]]))
plot(LGL2_ndvi_grazed)


#LGL2: Compare Grazed to Ungrazed -------------------

LGL2_ndvi_ungrazed<-as.data.frame(LGL2_ndvi_ungrazed)
LGL2_ndvi_grazed<-as.data.frame(LGL2_ndvi_grazed)

LGL_02_g<-tibble(
  Value= LGL2_ndvi_grazed$layer,
  Treatment='Grazed'
)

LGL_02_ug<-tibble(
  Value=LGL2_ndvi_ungrazed$layer,
  Treatment='Ungrazed'
)

LGL_02<-rbind(LGL_02_g, LGL_02_ug)
ggplot(data=LGL_01, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')
hist(LGL2_ndvi_grazed)




#(8) OGH-01 ----------------------------------------------------------


#OGH1_RGB<-stack("OGH-1 RGB2_modified.tif")
#OGH1_IR<-raster('OG-H2 IR_modified.tif')

# Align Extent--------------------------

OGH1_IR_proj<-projectRaster(OGH1_IR, OGH1_RGB)


# Shapefile ----------------------------

plot(OGH1_IR_proj)                                 #image of plot produced.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)                         #Error- boundary of enclosure and plot is misaligned but produced.

OGH1_shp_ug<-subset(polygon, PlotID=='OGH-01 UG')  #Ungrazed
OGH1_shp_g<-subset(polygon, PlotID == 'OGH-01 G')  # Grazed


# OGH1: Mask and clip raster to polygon ------------

#OGH1: Ungrazed

OGH1_RGB_mask<-mask(OGH1_RGB, OGH1_shp_ug)            #Error in x@polygons[[i]] : subscript out of bounds
OGH1_IR_mask<-mask(OGH1_IR_proj, OGH1_shp_g)

OGH1RGB_crop<-crop(OGH1_RGB_mask, OGH1_shp_ug)
plot(OGH1RGB_crop)
ex<-extent(OGH1RGB_crop)

OGH1IR_crop<-crop(OGH1_IR_mask, ex)
plot(OGH1IR_crop)


#OGH1: Stack and brick IR and RGB -----------------------

OGH1_stack<-stack(OGH1RGB_crop, OGH1IR_crop)
nlayers(OGH1_stack)


#OGH1: Calculate NDVI of the Ungrazed ------------------

OGH1_ndvi_ungrazed<-((OGH1_stack[[5]]-OGH1_stack[[1]])/(OGH1_stack[[5]]+OGH1_stack[[1]]))
plot(OGH1_ndvi_ungrazed)
hist(OGH1_ndvi_ungrazed)


# Do it all again for OGH1 Grazed ------------------------

#OGH1: Grazed

OGH1_RGB_mask<-mask(OGH1_RGB, OGH1_shp_g)
OGH1_IR_mask<-mask(OGH1_IR_proj, OGH1_shp_g)

OGH1RGB_crop<-crop(OGH1_RGB_mask, OGH1_shp_g)
plot(OGH1RGB_crop)
ex<-extent(OGH1RGB_crop)

OGH1IR_crop<-crop(OGH1_IR_mask, ex)
plot(OGH1IR_crop)
```

```
#OGH1: Stack and brick IR and RGB ----------------------

OGH1_stack<-stack(OGH1RGB_crop, OGH1IR_crop)
nlayers(OGH1_stack)


#OGH1: Calculate NDVI ---------------------------------

OGH1_ndvi_grazed<-((OGH1_stack[[5]]-OGH1_stack[[1]])/ (OGH1_stack[[5]]+OGH1_stack[[1]]))
plot(OGH1_ndvi_grazed)


#OGH1: Compare Grazed to Ungrazed -------------------

OGH1_ndvi_ungrazed<-as.data.frame(OGH1_ndvi_ungrazed)
OGH1_ndvi_grazed<-as.data.frame(OGH1_ndvi_grazed)

OGH_01_g<-tibble(
  Value= OGH1_ndvi_grazed$layer,
  Treatment='Grazed'
)

OGH_01_ug<-tibble(
  Value=OGH1_ndvi_ungrazed$layer,
  Treatment='Ungrazed'
)

OGH_01<-rbind(OGH_01_g, OGH_01_ug)
ggplot(data=OGH_01, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')
hist(OGH1_ndvi_grazed)




#(9) OGH-02 ----------------------------------------------------------



#OGH2_RGB<-stack("OG-H2 RGB_modified.tif")
#OGH2_IR<-raster('OG-H2 IR_modified.tif')


# Align Extent--------------------------

OGH2_IR_proj<-projectRaster(OGH2_IR, OGH2_RGB)


# Shapefile ----------------------------

plot(OGH2_IR_proj)                                    #image of plot produced.
polygon<-shapefile("Polygons.shp")
plot(polygon, add=TRUE)                               #image of boundary out of bounds of plot but both are displayed.

OGH2_shp_ug<-subset(polygon, PlotID=='OGH-02 UG')  #Ungrazed
OGH2_shp_g<-subset(polygon, PlotID == 'OGH-02 G')  # Grazed


#OGH2: Mask and clip raster to polygon -------------

#OGH2: Ungrazed

OGH2_RGB_mask<-mask(OGH2_RGB, OGH2_shp_ug)                #Error in x@polygons[[i]] : subscript out of bounds
OGH2_IR_mask<-mask(OGH2_IR_proj, OGH2_shp_g)

OGH2RGB_crop<-crop(OGH2_RGB_mask, OGH2_shp_ug)
plot(OGH2RGB_crop)
ex<-extent(OGH2RGB_crop)

OGH2IR_crop<-crop(OGH2_IR_mask, ex)
plot(OGH2IR_crop)


#OGH2: Stack and brick IR and RGB -----------------------

OGH2_stack<-stack(OGH2RGB_crop, OGH2IR_crop)
nlayers(OGH2_stack)


#OGH2: Calculate NDVI of the Ungrazed ------------------

OGH2_ndvi_ungrazed<-((OGH2_stack[[5]]-OGH2_stack[[1]])/(OGH2_stack[[5]]+OGH2_stack[[1]]))
plot(OGH2_ndvi_ungrazed)
hist(OGH2_ndvi_ungrazed)


# Do it all again for LGH2 Grazed ------------------------

#OGH2: Grazed

OGH2_RGB_mask<-mask(OGH2_RGB, OGH2_shp_g)
OGH2_IR_mask<-mask(OGH2_IR_proj, OGH2_shp_g)

OGH2RGB_crop<-crop(OGH2_RGB_mask, OGH2_shp_g)
plot(OGH2RGB_crop)
ex<-extent(OGH2RGB_crop)
```

```
OGH2IR_crop<-crop(OGH2_IR_mask, ex)
plot(OGH2IR_crop)


#OGH2: Stack and brick IR and RGB -----------------------

OGH2_stack<-stack(OGH2RGB_crop, OGH2IR_crop)
nlayers(OGH2_stack)


#OGH2: Calculate NDVI -----------------------------------

OGH2_ndvi_grazed<-((OGH2_stack[[5]]-OGH2_stack[[1]])/ (OGH2_stack[[5]]+OGH2_stack[[1]]))
plot(OGH2_ndvi_grazed)


#OGH2: Compare Grazed to Ungrazed --------------------

OGH2_ndvi_ungrazed<-as.data.frame(OGH2_ndvi_ungrazed)
OGH2_ndvi_grazed<-as.data.frame(OGH2_ndvi_grazed)

OGH_02_g<-tibble(
  Value= OGH2_ndvi_grazed$layer,
  Treatment='Grazed'
)

OGH_02_ug<-tibble(
  Value=OGH2_ndvi_ungrazed$layer,
  Treatment='Ungrazed'
)

OGH_02<-rbind(OGH_02_g, OGH_02_ug)
ggplot(data=OGH_01, aes(x=Treatment, y=Value))+
  geom_violin(scale='area')
hist(OGH2_ndvi_grazed)
```