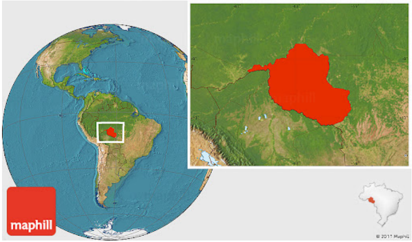
TEMPORAL ANALYSIS OF DEFORESTATION IN AN AREA IN RONDÔNIA

github.com/erikazanetti
A.Y. 2024-2025

Summary

1. Area
2. Objective
3. Methods
4. Analysis
5. Conclusion

The Rondônia area



- Area: 237,591 km²
- Original forest area (1500): 211,071 km²
- Current forest area (2020): 121,382 km² (51.09% of Rondônia)
- Accumulated deforestation (2001-2019): 31,660 km²

Area of study

- Area of study: $\sim$ 4100 km² (1.73% of Rondônia)
- Period: 2017-2023

Figure: Area of study in June 2023

Summary

Objective

Compare the forest loss rate of the area studied with the general rates of Rondônia.

Summary

Packages used

```
# Install required packages if not already installed
install.packages(c("terra", "ggplot2", "patchwork", "viridis", "devtools"))

# Load the packages
library(terra)     # For handling raster data (e.g., satellite images)
library(ggplot2)   # For creating graphs
library(patchwork) # For composing multiple graphs in a single plot
library(viridis)   # For color palettes
library(devtools)  # For managing R packages

# Install imageRy package from GitHub
install_github("ducciorocchini/imageRy")
library(imageRy)   # For manipulating raster images
```

Images used

- Downloaded from the Copernicus browser `https://browser.dataspace.copernicus.eu/`
- 4 images: 2 images from 2017 and 2 images from 2023 (true color and false color)

Summary

Importing images

First, we set the working directory, then we import the images for the years 2017 and 2023, while assigning them to an object.

```
setwd("C:/Users/Erika Zanetti/OneDrive - Alma Mater Studiorum Universit  di
     Bologna/spatial ecology in r")

r17 <- rast("2017-06-28-00_00_2017-06-28-23_59_Sentinel-2_L2A_True_color.jpg")
r17f <- rast("2017-06-28-00_00_2017-06-28-23_59_Sentinel-2_L2A_False_color.jpg")
r23 <- rast("2023-06-27-00_00_2023-06-27-23_59_Sentinel-2_L2A_True_color.jpg")
r23f <- rast("2023-06-27-00_00_2023-06-27-23_59_Sentinel-2_L2A_False_color.jpg")
```

After importing all the images, let's plot them all together in a multiframe.

```
par(mfrow=c(2,2))
plot(r17, main="True color 2017")
plot(r17f, main="False color 2017")
plot(r23, main="True color 2023")
plot(r23f, main="False color 2023")
```

Classifying the images

Assigning red, green and blue bands of 2017's true color's image and 2017's false color's NIR band to their relative objects. Uniting all the bands in one element.
Using the function im.classify() of the imageRy package to classify the images into 2 clusters (cleared forest and original forest)

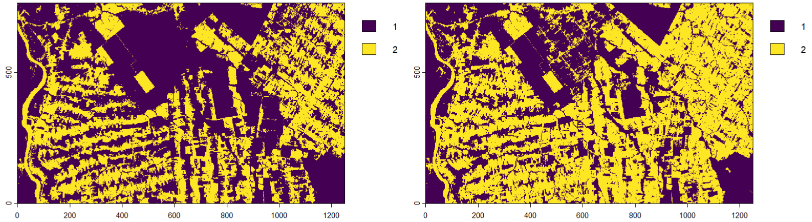
```
r17r <- r17[[1]]                           # Red band
r17g <- r17[[2]]                           # Green band
r17b <- r17[[3]]                           # Blue band
r17nir <- r17f[[1]]                        # NIR band
bandsr17 <- c(r17r, r17g, r17b, r17nir)    # Uniting all the bands

# Directly classifying the image with all the bands
bandsr17c <- im.classify(bandsr17,num_clusters=2)
```

Doing the same process for 2023's images.

Classifying the images

Plot the classified images of 2017 and 2023 in a multiframe using colors from the viridis palette.

```
clv <- viridis(100)
par(mfrow=c(1,2))
plot(bandsr17c, col=clv)
plot(bandsr23c, col=clv)
```



Classifying the images

Calculating the frequency and percentage of each class (cleared fores and original forest) for both 2017 and 2023.
Then calculating the forest loss percentage between 2017 and 2023.

```
f17 <- freq(bandsr17c)
tot17 <- ncell(bandsr17c)

f23 <- freq(bandsr23c)
tot23 <- ncell(bandsr23c)

perc17 = f17*100/tot17
# cleared forest = 41
# original forest = 59
perc23 = f23*100/tot23
# cleared forest = 56
# original forest = 44

forest_loss_percentage <- perc17 - perc23
```

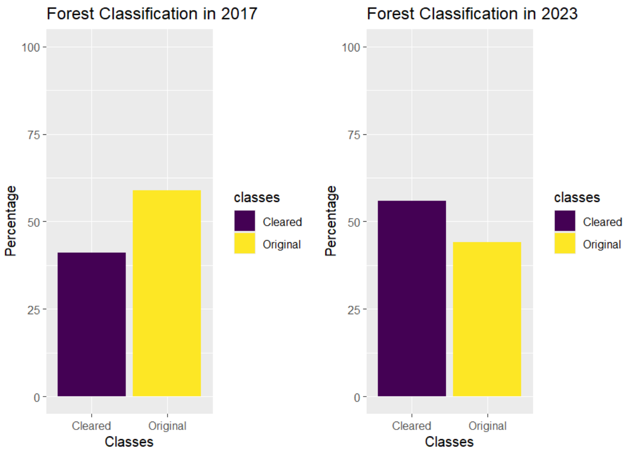
A 3% of forest loss is calculated.

Classifying the images

To make it even clearer, we can show the differences through bar graphs.

```
year_2017 <- c(41,59)
year_2023 <- c(56,44)
classes <- c("Cleared", "Original")
data <- data.frame(classes, year_2017, year_2023)

g1 <- ggplot(data, aes(x=classes, y=year_2017, fill=classes)) + geom_bar(stat="
     identity") + scale_fill_viridis_d(option = "D") + ylim(c(0, 100)) + labs(
     title = "Forest Classification in 2017", y = "Percentage", x = "Classes")
g2 <- ggplot(data, aes(x=classes, y=year_2023, fill=classes)) + geom_bar(stat="
     identity") +  scale_fill_viridis_d(option = "D") + ylim(c(0, 100)) + labs(
     title = "Forest Classification in 2023", y = "Percentage", x = "Classes")
g1 + g2
```

Classifying the images



DVI and NDVI

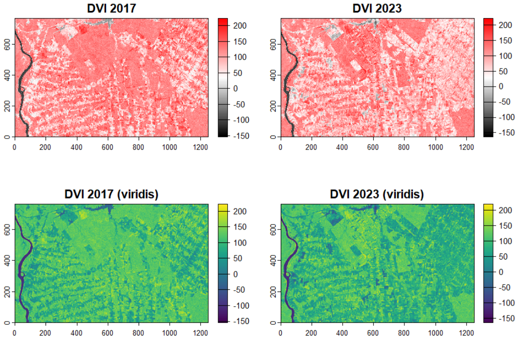
DVI and NDVI are used to monitor vegetation health and density.
Analyzing DVI first.
We can create a color scale from black to white to red with 100 as gradient.

```
cl <- colorRampPalette(c("black","white","red"))(100)

dvi17 = bandsr17[[4]] - bandsr17[[1]]
dvi23 = bandsr23[[4]] - bandsr23[[1]]
```

DVI and NDVI

Then, we plot the results using the previously created palette and using the viridis palette.

```
par(mfrow=c(2,2))
plot(dvi17, col=cl, main="DVI 2017")
plot(dvi23, col=cl, main="DVI 2023")
plot(dvi17, col=clv, main="DVI 2017 (viridis)")
plot(dvi23, col=clv, main="DVI 2023 (viridis)")
```



DVI and NDVI

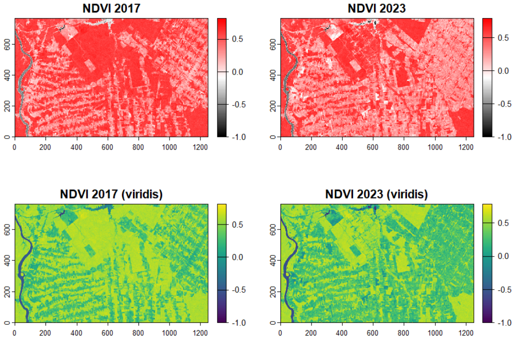
Now we can calculate the NDVI by doing (NIR-Red)/(NIR+Red).

```
ndvi17 = dvi17 / (bandsr17[[4]] + bandsr17[[1]])
ndvi23 = dvi23 / (bandsr23[[4]] + bandsr23[[1]])
```

DVI and NDVI

We plot the results using the previously created palette and using the viridis palette.

```
par(mfrow=c(2,2))
plot(ndvi17, col=cl, main="NDVI 2017")
plot(ndvi23, col=cl, main="NDVI 2023")
plot(ndvi17, col=clv, main="NDVI 2017 (viridis)")
plot(ndvi23, col=clv, main="NDVI 2023 (viridis)")
```

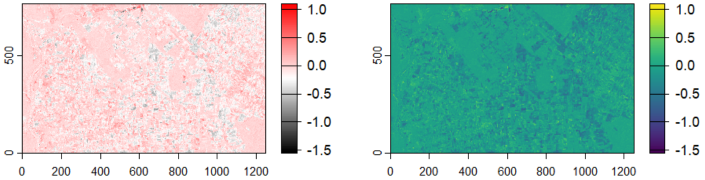


DVI and NDVI

NDVI difference:

- Positive values: vegetation improved;
- Negative values: vegetation worsened/decreased.

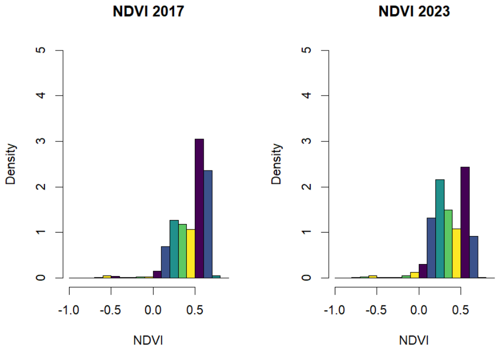
```
difNDVI = ndvi23 - ndvi17
par(mfrow=c(1,2))
plot(difNDVI,col=cl)
plot(difNDVI,col=clv)
```



DVI and NDVI

Visualizing NDVI difference through histograms for 2017 and 2023.

```
par(mfrow=c(1,2))
hist17 <- hist(ndvi17,main="NDVI 2017", xlab = "NDVI", nclass=20, freq=F, ylim=c
      (0,5), col=viridis(5))
hist23 <- hist(ndvi23,main="NDVI 2023", xlab = "NDVI", nclass=20, freq=F, ylim=c
      (0,5), col=viridis(5))
```



PCA analysis for spatial variability

First, we calculate the PCA for 2017 and find the percentage of variability explained by each axis.
Then, we combine PC1 and PC2, because together they explain over 89% of the variability.
Finally, we calculate the standard deviation (SD) of the combined principal components (PC1 and PC2) from the PCA analysis, using a moving window (3x3 grid) to smooth the results. The focal() function applies the SD operation over the window.

```
pca17 <- im.pca(bandsr17)
tot17 <- sum(36.612812, 31.971524, 4.963559, 3.076220)
c(36.612812*100/tot17  # 47.78236 % of variability explained by the first axis
  31.971524*100/tot17  # 41.72515 % of variability explained by the second axis
  4.963559*100/tot17   # 6.477803 % of probability explained by the third axis
  3.076220*100/tot17   # 4.014689 % of probability explained by the fourth axis
  )
pc17comb <- pca17[[1]] + pca17[[2]]
pcsd17 <- focal(pc17comb, matrix(1/9, 3, 3), fun=sd)
```

PCA analysis for spatial variability

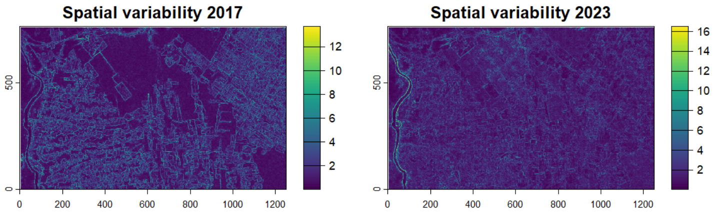
We do the same process for 2023.

```
pca23 <- im.pca(bandsr23)
tot23 <- sum(36.612812, 31.971524, 4.963559, 3.076220)
c(44.297913*100/tot23  # 57.81197 % of variability explained by the first axis
  20.766979*100/tot23  # 27.10241 % of variability explained by the second axis
  4.477552*100/tot23   # 5.843529 % of probability explained by the third axis
  3.009718*100/tot23   # 3.927899 % of probability explained by the fourth axis
  )
pc23comb <- pca23[[1]] + pca23[[2]]
pcsd23 <- focal(pc23comb, matrix(1/9, 3, 3), fun=sd)
```

PCA analysis for spatial variability

The results of the standard deviation (spatial variability) for both 2017 and 2023 are visualized side-by-side.

```
par(mfrow=c(1,2))
plot(pcsd17, col=clv, main="Spatial variability 2017")
plot(pcsd23, col=clv, main="Spatial variability 2017")
```



Summary

Conclusions

Comparing these values, we can infer that your area might not be as severely affected as the overall state of Rondônia:

- Rondônia's average annual forest loss (2001-2019) = 1.02% per year
- Studied area's annual forest loss (2018-2023) = 0.6% per year.

The studied area has experienced less deforestation per year compared to the broader trends observed in Rondônia from 2001 to 2019.
However, it's still important to understand the larger dynamics, as deforestation trends can vary significantly within smaller areas depending on the local causes (e.g., land use, agriculture, etc.).

Sources

- https://forestchampions.org/jxd_reports/en_Rond%C3%B4nia_Brazil.pdf
- https://www.globalforestwatch.org/