



Geoinformatics | Course Remote Sensing (1)

Schmitt | Ulloa

Summer Semester 2020

Practice 1: Software Products available for Remote Sensing and GIS data

Overview

Objectives: Familiarize with some of the most importants softwares available for visualization and processing of Remote Sensing (RS) data and GIS. Discuss about advantages and disadvantages for each program.

Data: For this practice, use the following files:

- Raster file: GeoTIFF (dop20_386_5818.tif)
- Vector file: Shapefile (Shape.shp)

Tasks: Open the data in the following programs:

- Paint
- IrfanView
- ERDAS IMAGINE
- ArcMap
- QGIS
- Python

Which software is better for which type of data? In which software can you visualize one band, multiple bands or a vector file? For each software try to extract the following information of your data whenever is possible: dimensions, coordinate reference system (CRS), histogram, pixel value, pixel size. Discuss the advantages and disadvantages of each software in terms of: price/availability, processing (i.e. the capacity of run all previous named functions), and versatility. Fill your results in the Table A.

Table A. Summary of software for RS and GIS data

Software	Load raster	Load vector	Visualization	Dimensions	CRS	Pixel value+size	Histogram	Price/availability	Versatility
Paint									
IrfanView									
ERDAS IMAGINE									
ArcMap									
QGIS									
Python									

Procedure

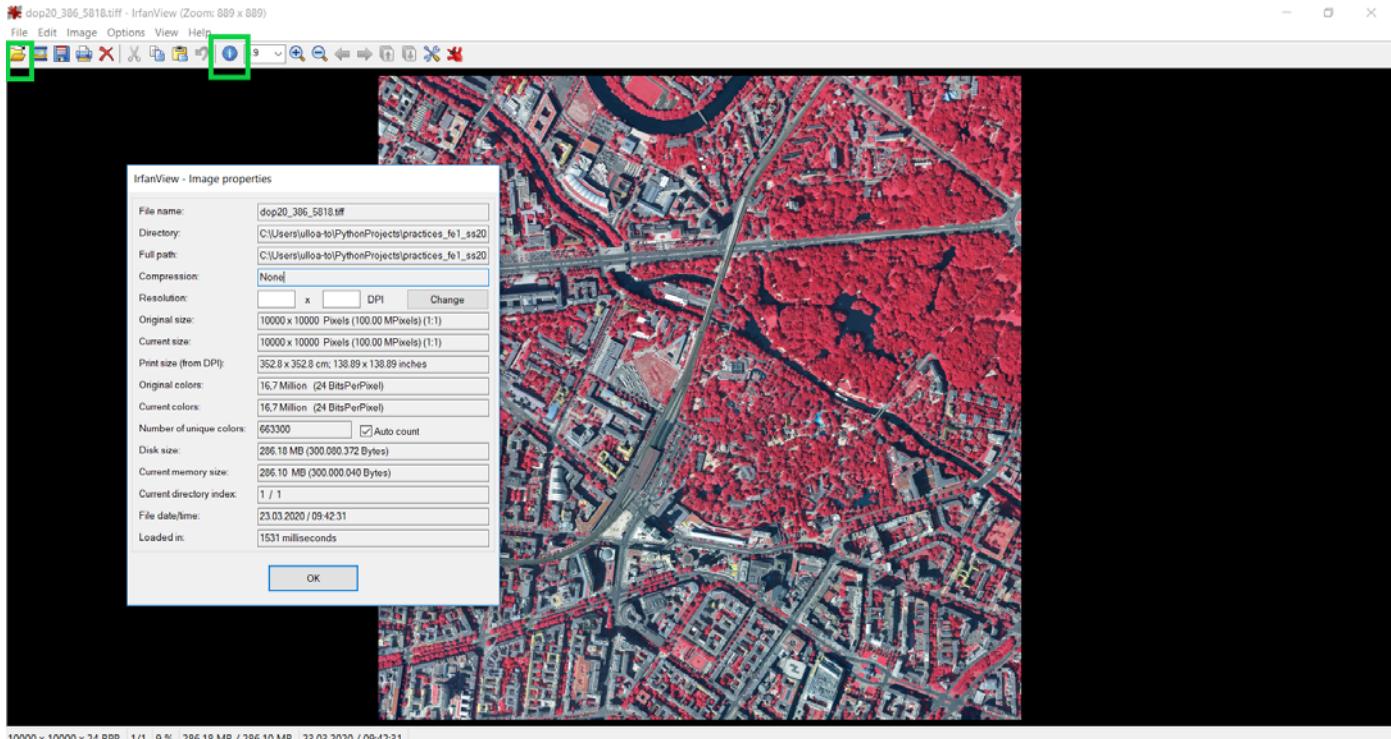
This is a step by step guide for the processing of files in each Software.

1. Paint

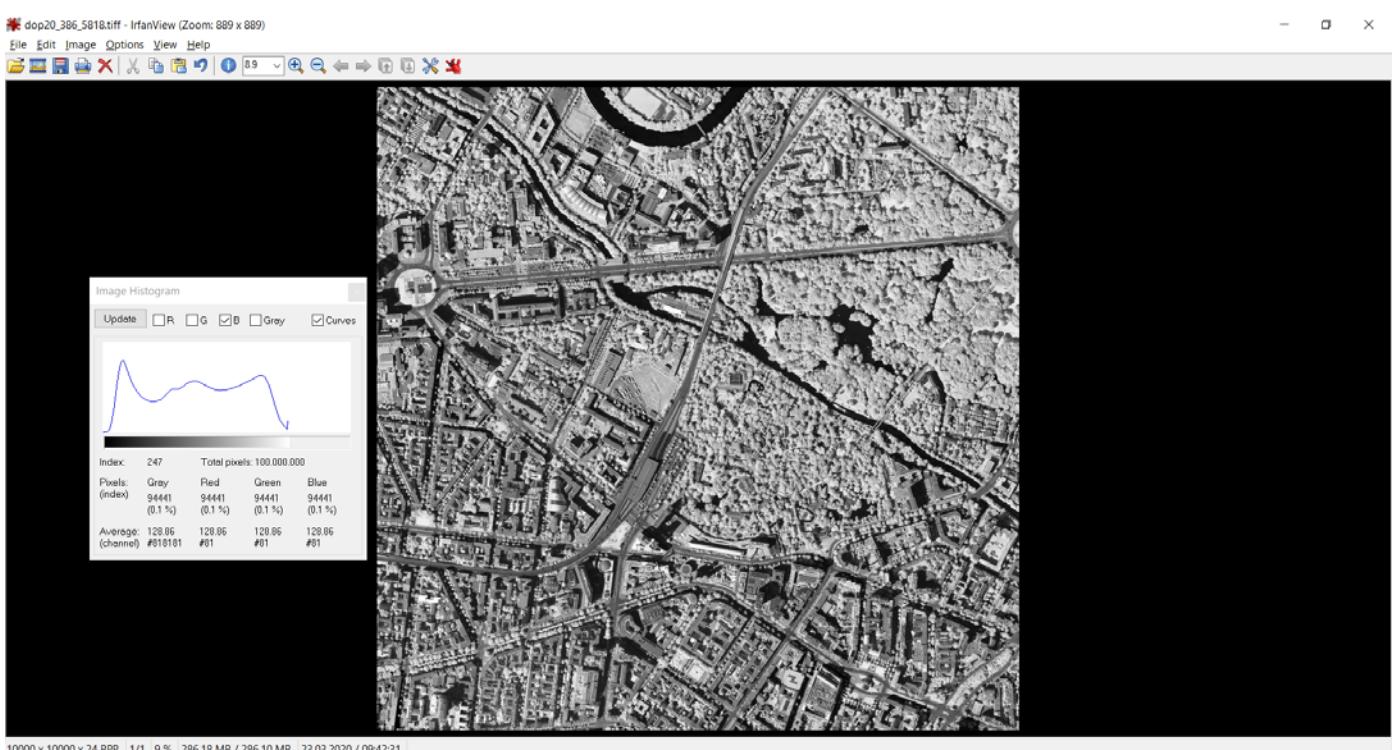
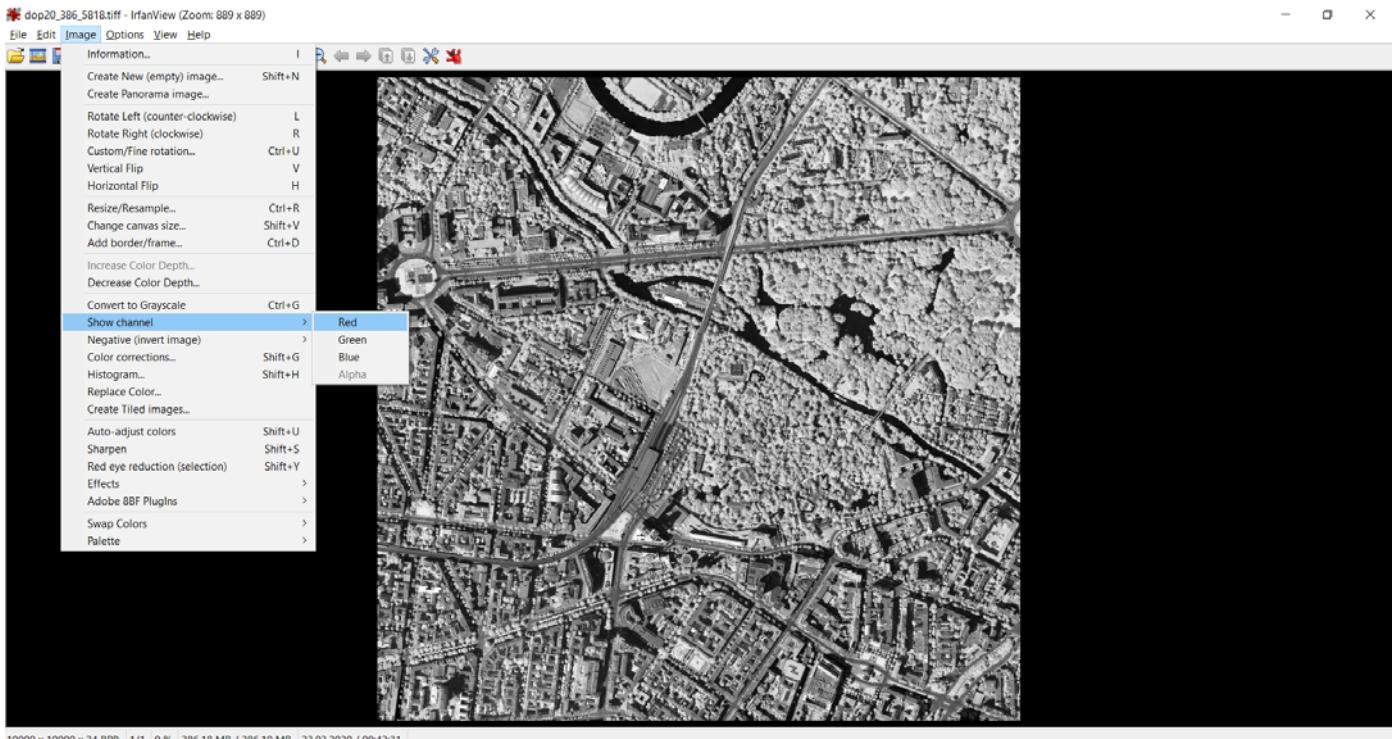
Open vector and raster files in Paint.

2. IrfanView

Open the raster file in **File > Open**. The button **Image information** opens a window. Explore the parameters offered that provide information about the image.



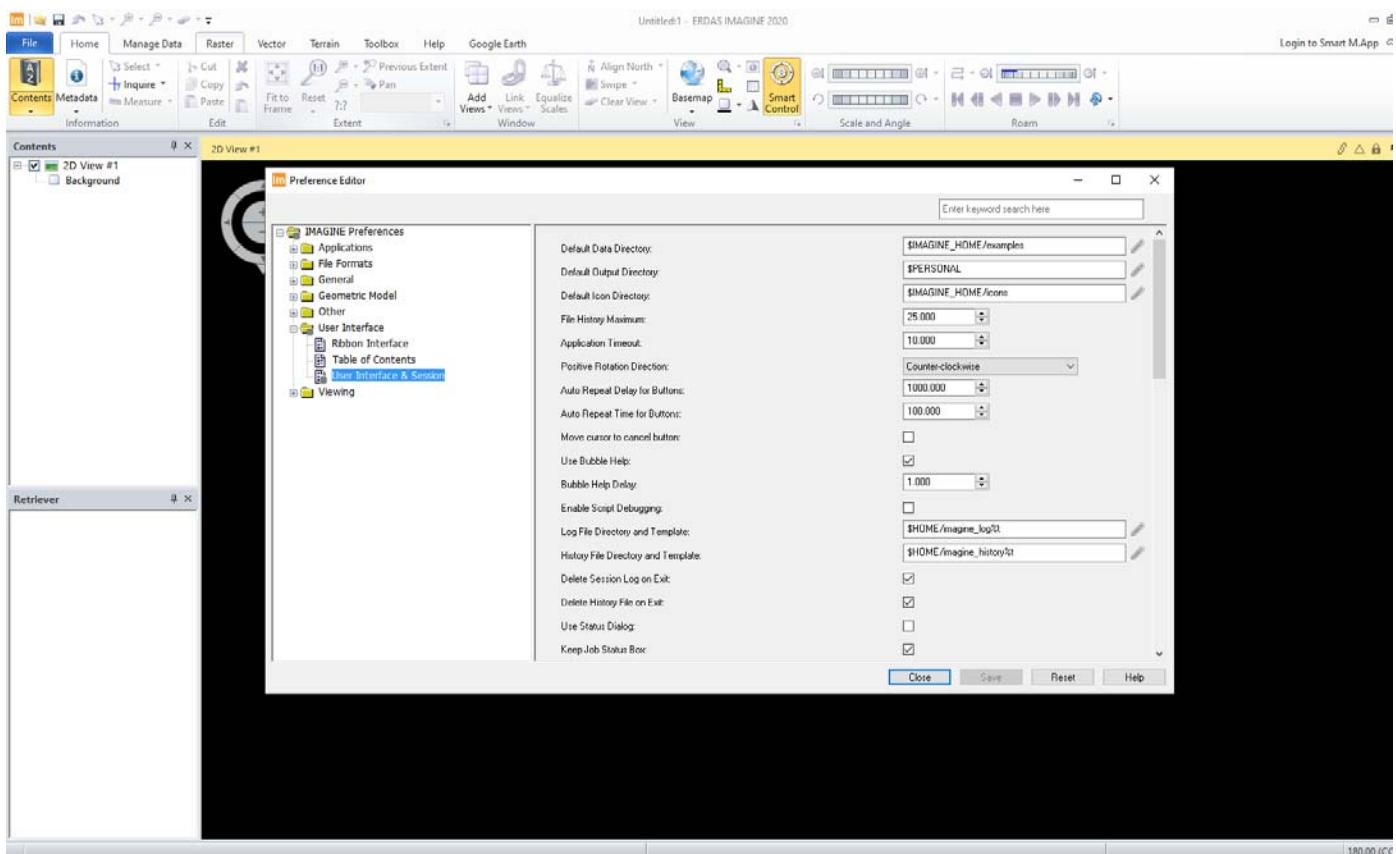
Under the Menu **Image**, there are many functions that allow you to extract information from the raster. For example, individual band visualization or histogram.



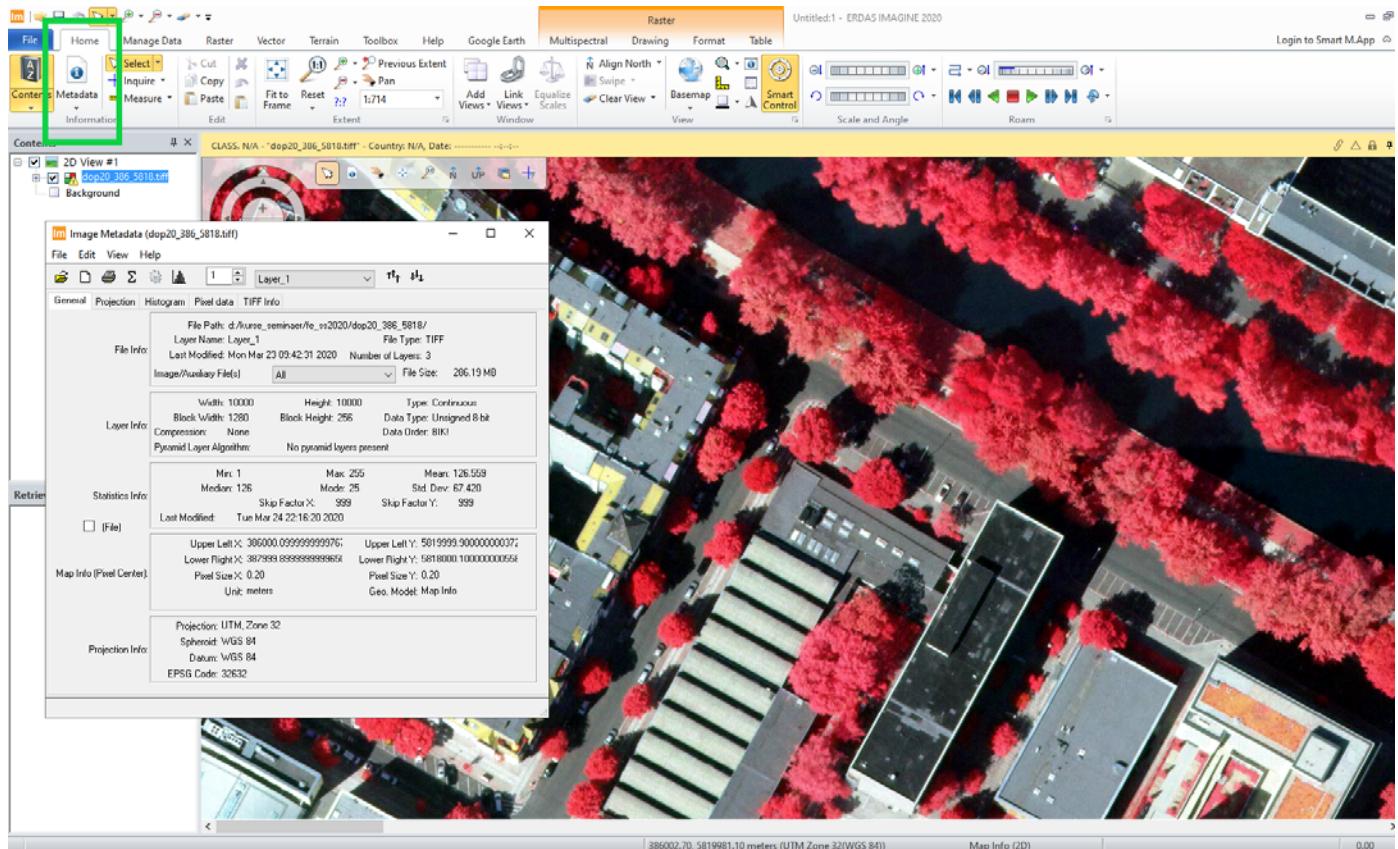
Follow the same procedure and open the vector file. Which commands allow to get which information?

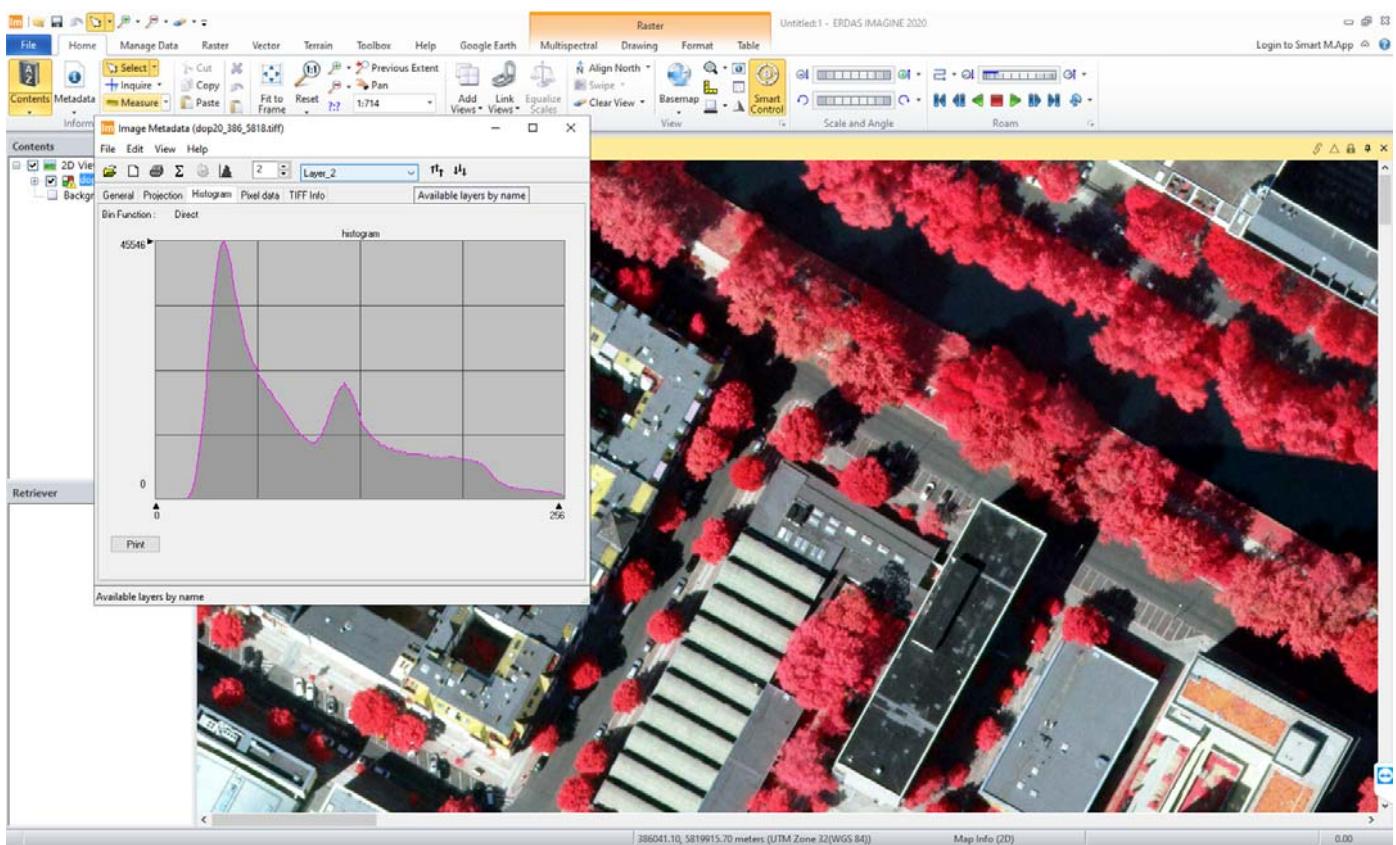
3. ERDAS IMAGINE

First, it is important that you indicate ERDAS where are your files and in which folder you are working. This is similar as defining a working directory. Later on, to load a raster or vector file, go to **File > Open > Raster Layer**.

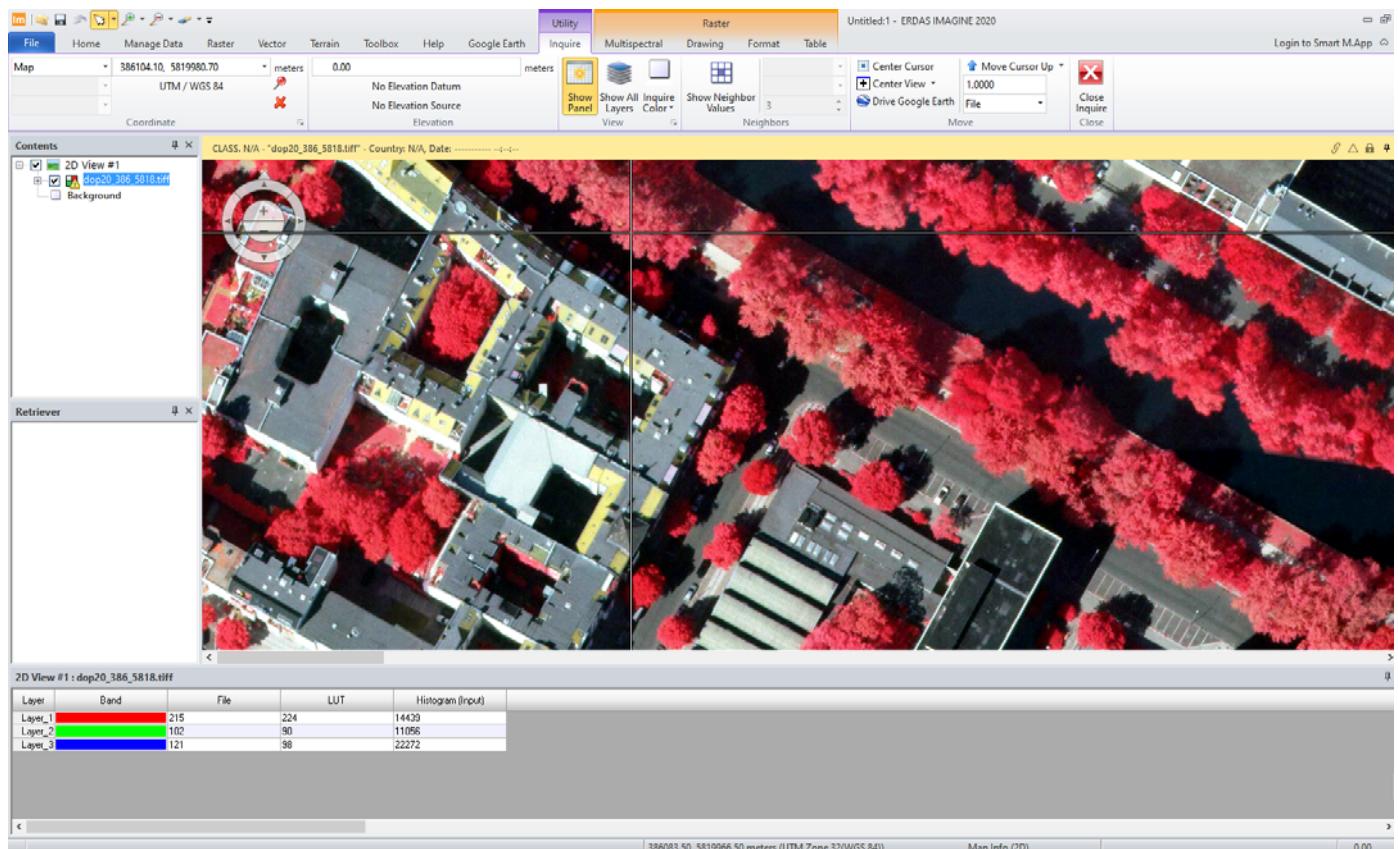


In the Menu, click on **Metadata**. Another window opens with different tabs and information about the image. Which parameters can you extract from here?





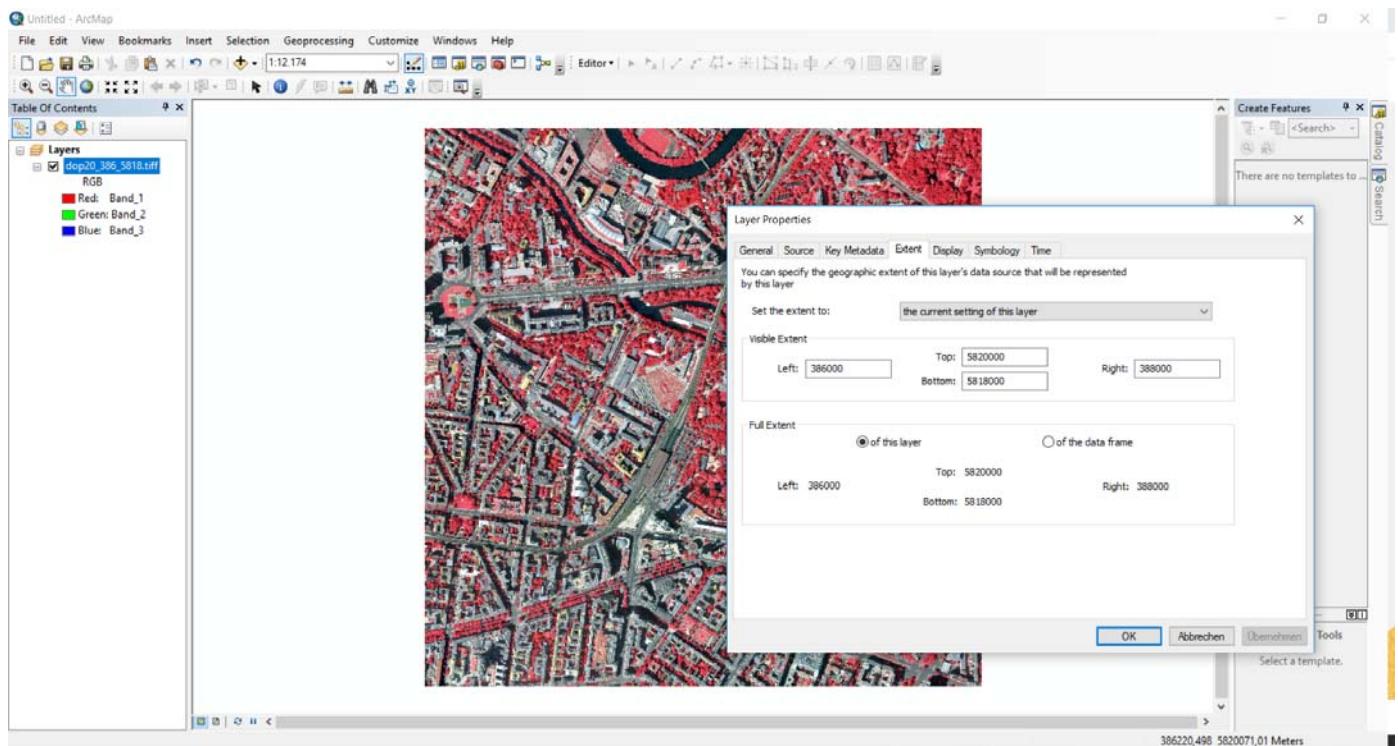
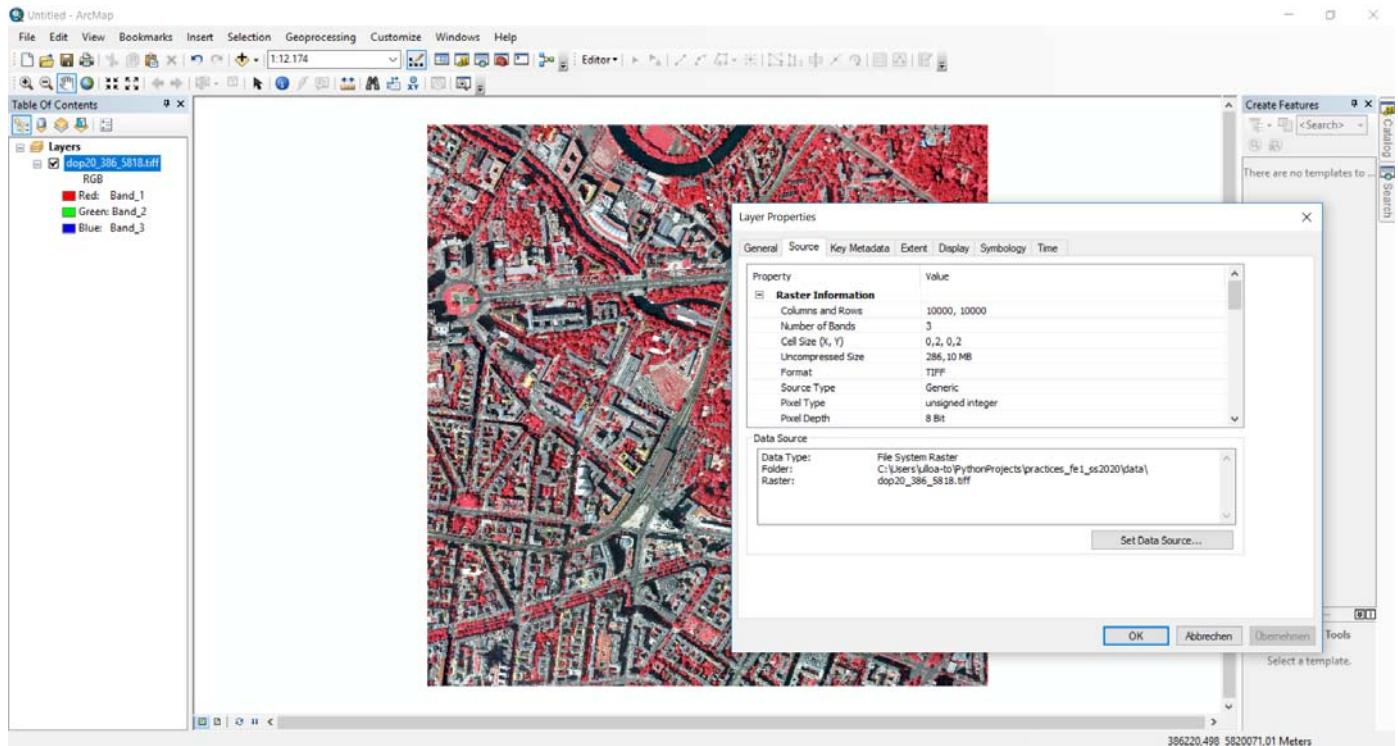
Under Home>Inquire , another window opens with information on every pixel per band.



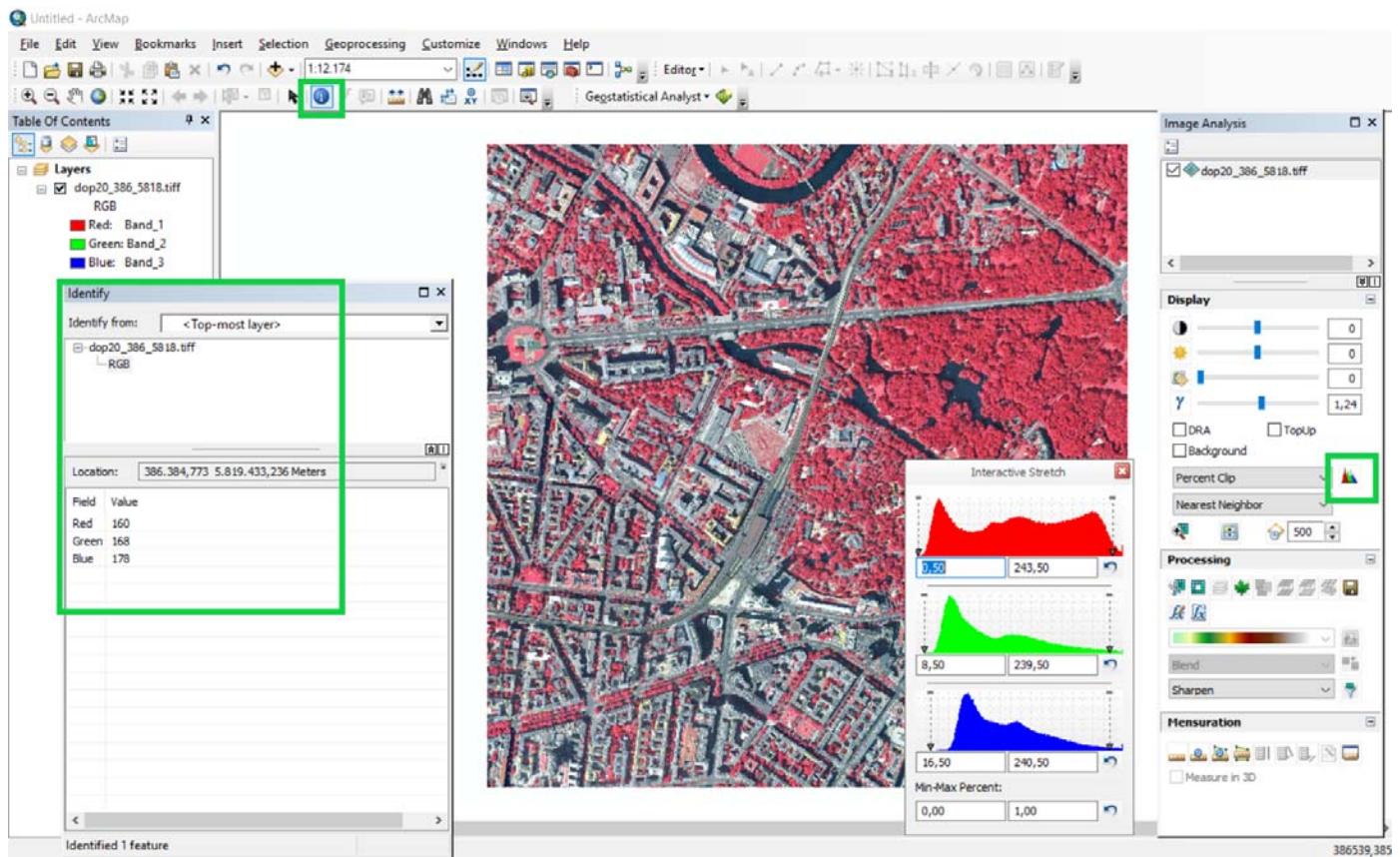
Follow the same procedure and open the vector file. Which commands allow to get which information?

4. ArcMap

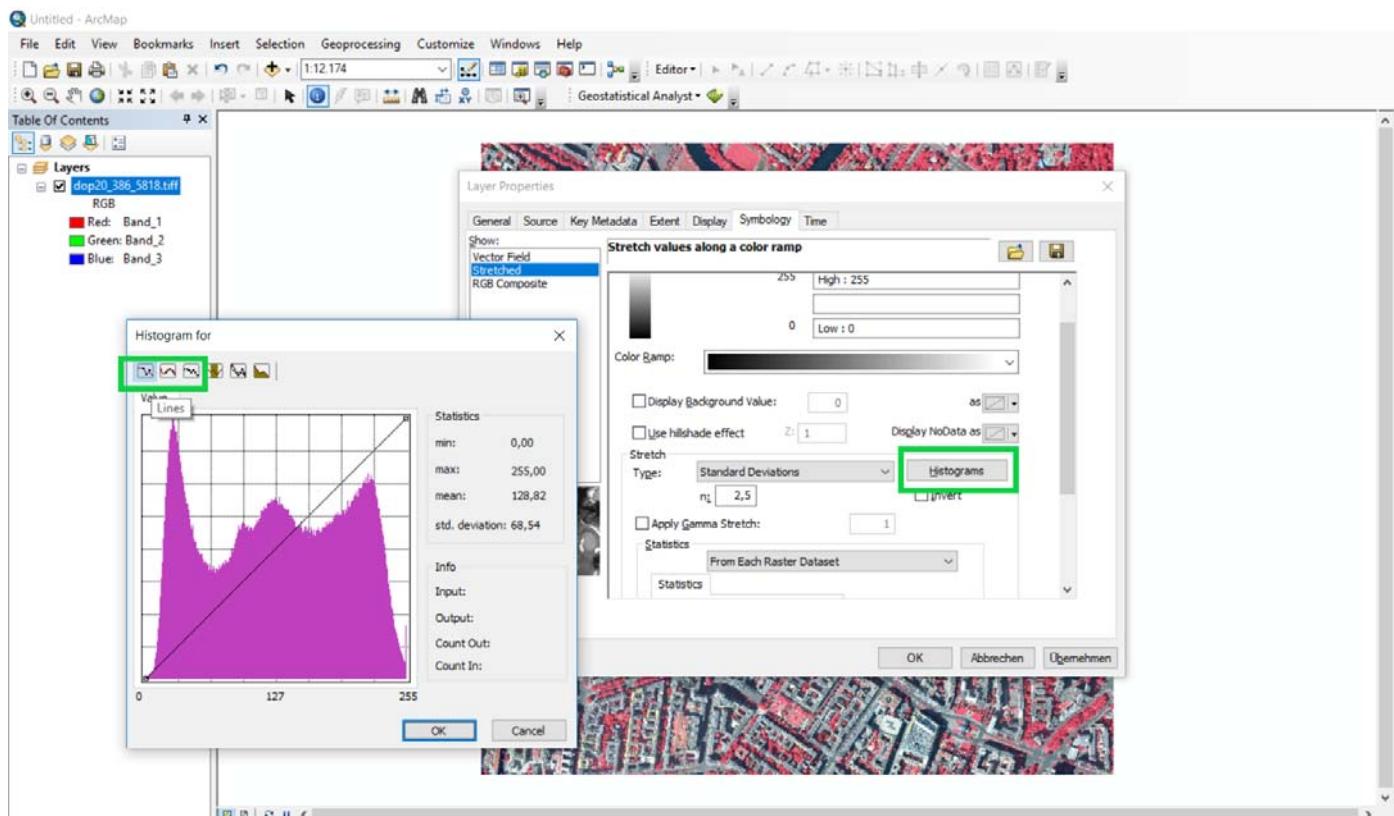
In ArcMap you can load a vector or raster file with the commands: **File > Add Data > Add Data**. Another option is to drag and drop your file on the console. Right click on the file, displays a menu. Under **Properties > Layer Properties** another window opens which allows you to access to many categories of information about your file. Check all the categories.



Using the button **Identify** from the Menu, allows you to click on the image and retrieves information about your file. Which information is possible to extract here? Moreover, under **Windows > Image Analysis** is possible to plot a histogram for all bands of the RasterStack.



Another way of visualizing the histogram is under Right click > Layer Properties > Stretched . In this section, you can try different ways of [stretching](https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/working-with-the-histogram-when-displaying-raster-data.html) (<https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/working-with-the-histogram-when-displaying-raster-data.html>) your data.



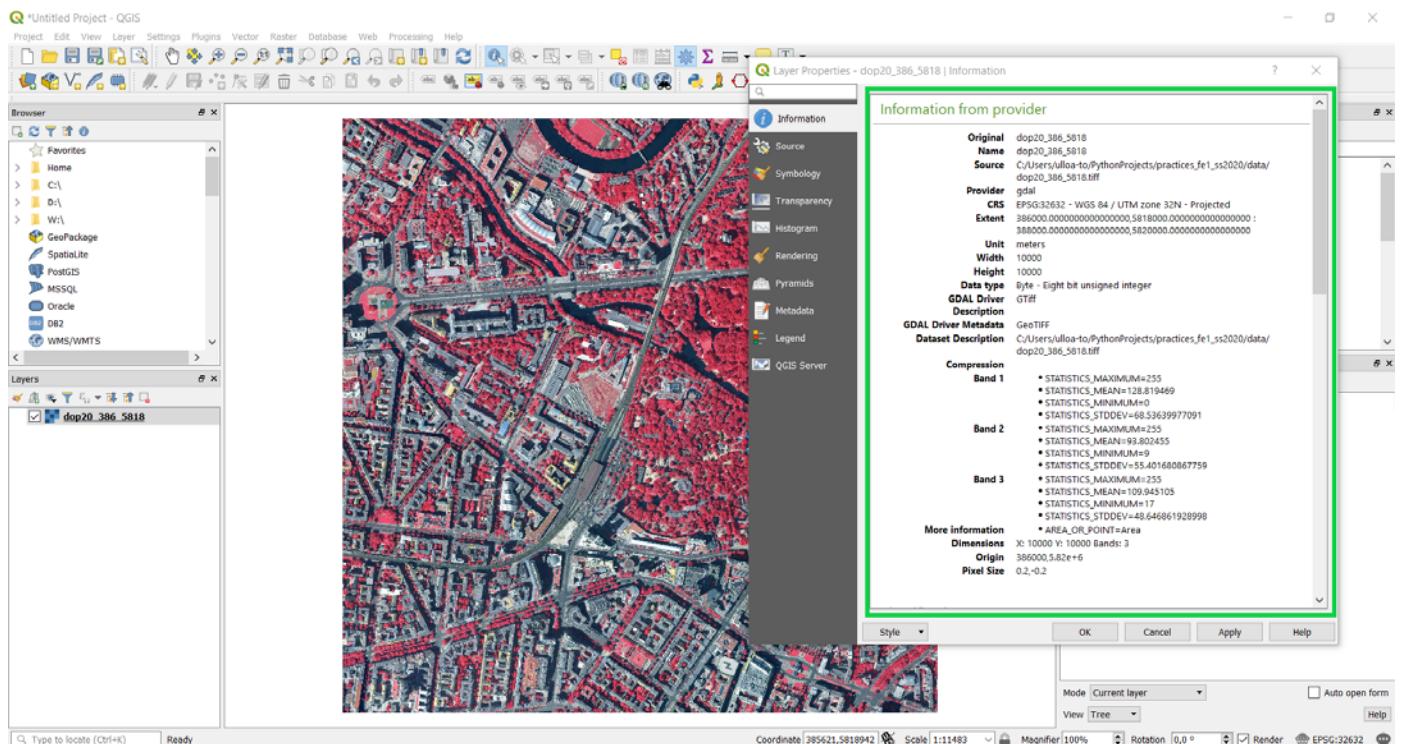
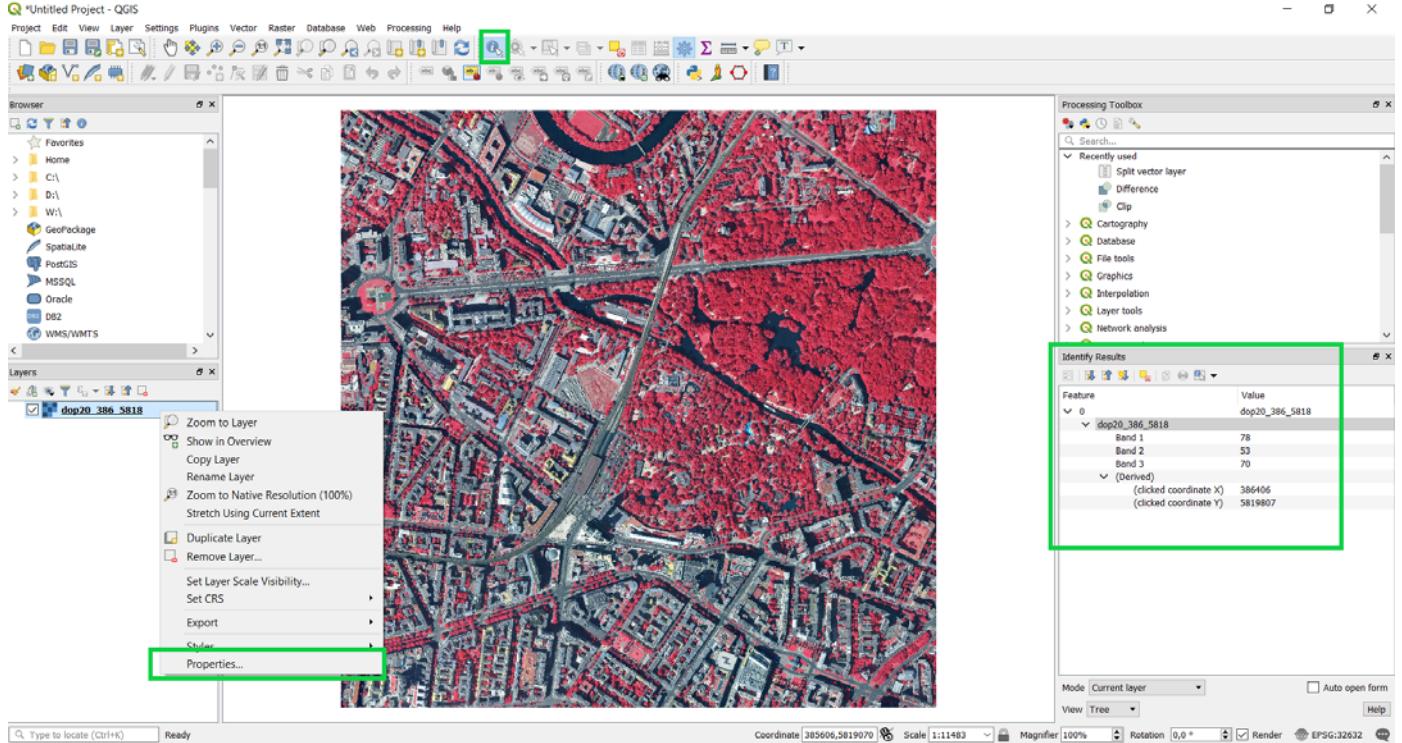
Follow the same procedure and open the vector file. Which commands allow to get which information?

5. QGIS

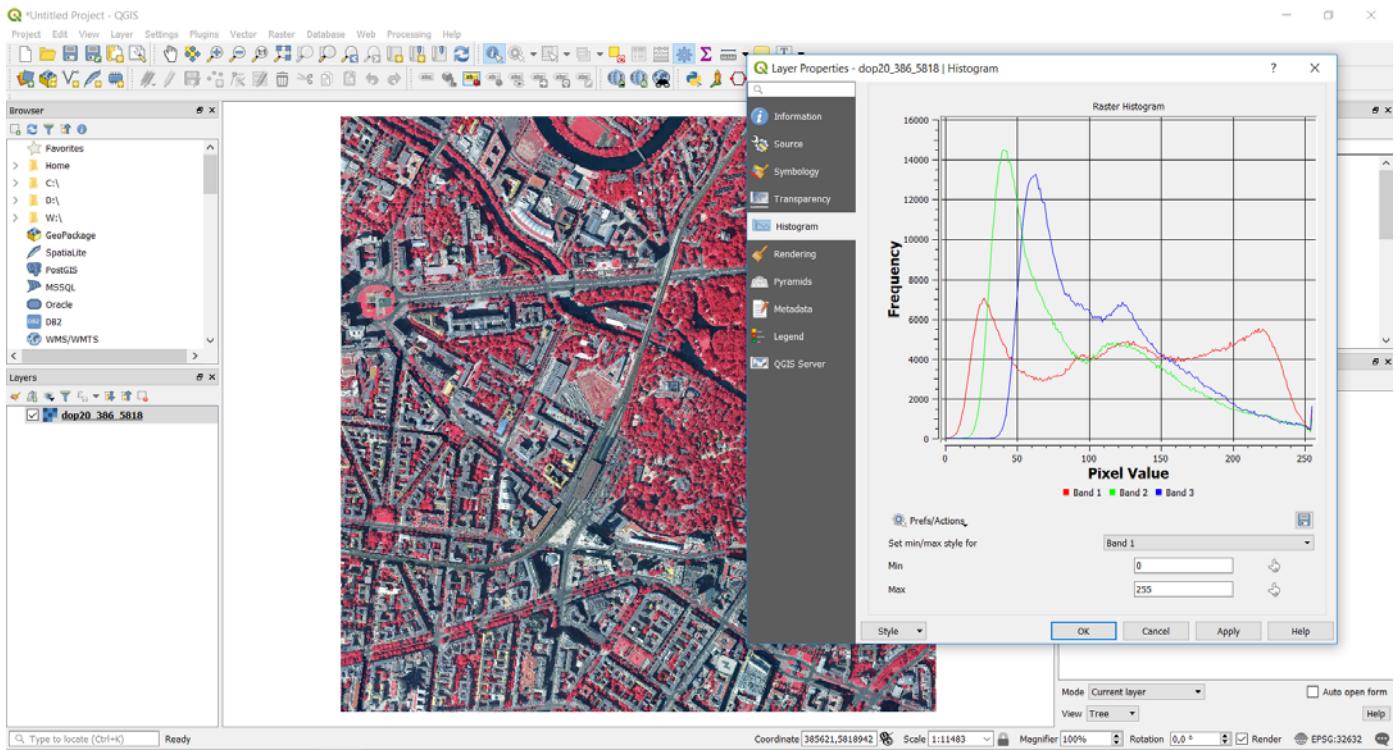
In QGIS you can load a vector or raster file with the commands: Project > Open > Choose File . Another option is to drag and drop your file on the console.

Right click on the file, displays a menu. Under Properties another window opens which allows you to access to many categories of information about your file. Check all the categories.

Using the button Identify Features from the Menu, allows you to click on the image and retrieves infomation about your file. Which information is possible to extract here?



Under Histogram it is possible to plot a histogram of the 3 bands of your file. Can you do the same for one band?



Follow the same procedure and open the vector file. Which commands allow to get which information?

6. Python

There are diverse programming languages that can be used for processing of RS and GIS data. Python and R are very common, among others. In Python, use the following code and extract the necessary information for raster and vector files.

In [1]:

```
print("Hello world")
```

Hello world

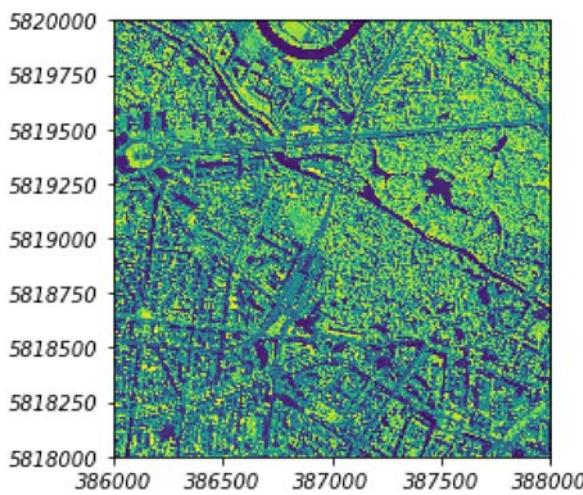
In [6]:

```

import os
import rasterio
from rasterio.plot import show
import numpy
import matplotlib.pyplot as plt
%matplotlib inline
import geopandas as gp

#TODO insert your root directory here
src = r"C:\Users\ulloa-to\PythonProjects\practices_fe1_ss2020\practice1\data"
rasterfile = os.path.join(src, "dop20_386_5818.tif")
raster = rasterio.open(rasterfile)
show(raster)

```



Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1da5eff12e8>
```

Access to raster properties

In [4]:

```
#Filename
raster.name
```

Out[4]:

```
'C:\\\\Users\\\\ulloa-to\\\\PythonProjects\\\\practices_fe1_ss2020\\\\practice1\\\\data\\\\dop20_386_5818.tif'
```

In [5]:

```
#Number of Bands
raster.count
```

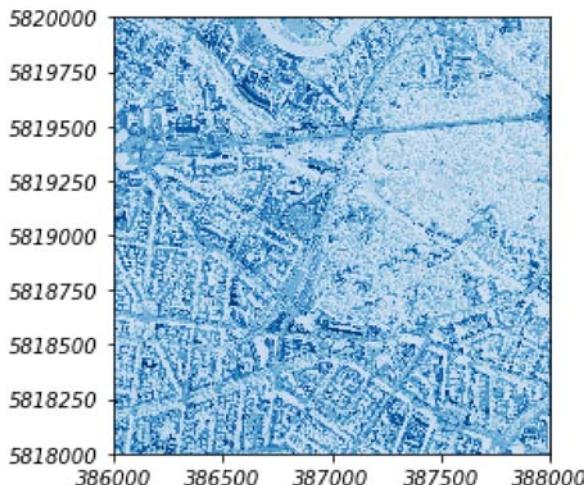
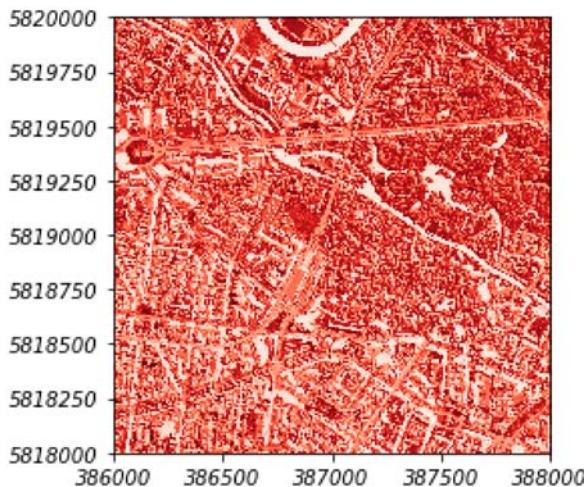
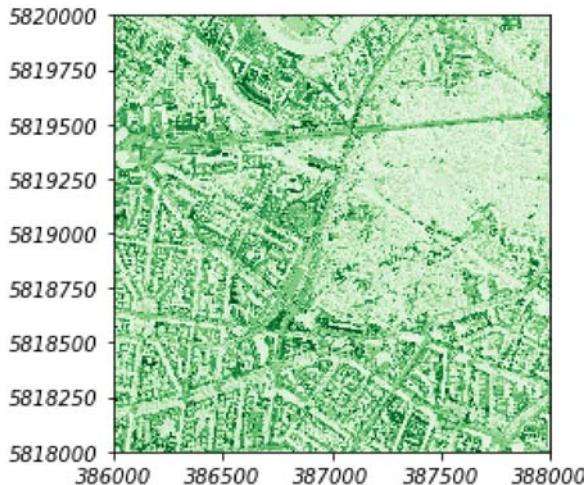
Out[5]:

```
3
```

Visualization of Bands

In [6]:

```
#Green  
show((raster, 2), cmap='Greens')  
#Red  
show((raster, 1), cmap='Reds')  
#Infrared  
show((raster, 3), cmap='Blues')
```



Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c1cd81d198>
```

Raster dimensions

In [9]:

```
#Bounding Box
b = raster.bounds
#width
w = raster.width
#height
h = raster.height
#Resolution
horizontal_res = (b[2] - b[0]) / w

print("The boundaries of the raster are:", b, "\nWidth:", w, "\nHeight:", h, "\nResolution:"
```

```
The boundaries of the raster are: BoundingBox(left=386000.0, bottom=5818000.0, right=388000.0, top=5820000.0)
Width: 10000
Height: 10000
Resolution: 0.2
```

Coordinate Reference System

In [10]:

```
##Georeference
#TODO probably this is a good place to talk about units
#Coordinate Reference System
raster.crs

print("The CRS of the raster is: {}".format(raster.crs))
```

```
The CRS of the raster is: EPSG:32632
```

Pixel values

In [10]:

```
#Navigate through Band(s)
#indexes of bands
raster.indexes
ir_band = raster.read(3)
# (Numpy) Format
{ i: dtype for i, dtype in zip(raster.indexes, raster.dtypes) }
#uint8 := 0..255
ir_band
numpy.min(ir_band)
numpy.max(ir_band)
#Pixel value

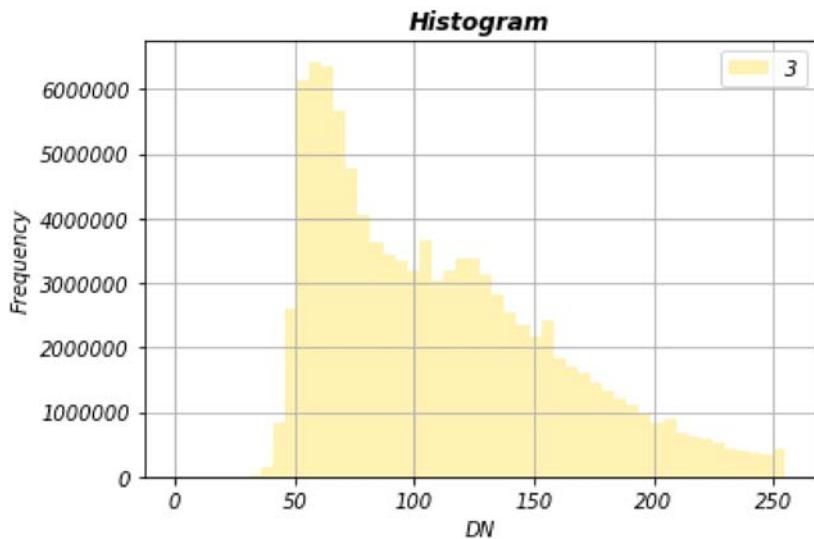
print("Minimum pixel value:", numpy.min(ir_band))
print("Maximum pixel value:", numpy.max(ir_band))
```

Minimum pixel value: 0
 Maximum pixel value: 255

Plot histogram of 3 bands

In [11]:

```
from rasterio.plot import show_hist
#Show Infrared Channel
show_hist((raster, 3), bins=50, lw=0.0, stacked=False, alpha=0.3, histtype='stepfilled', tit
```



Load vector and read properties

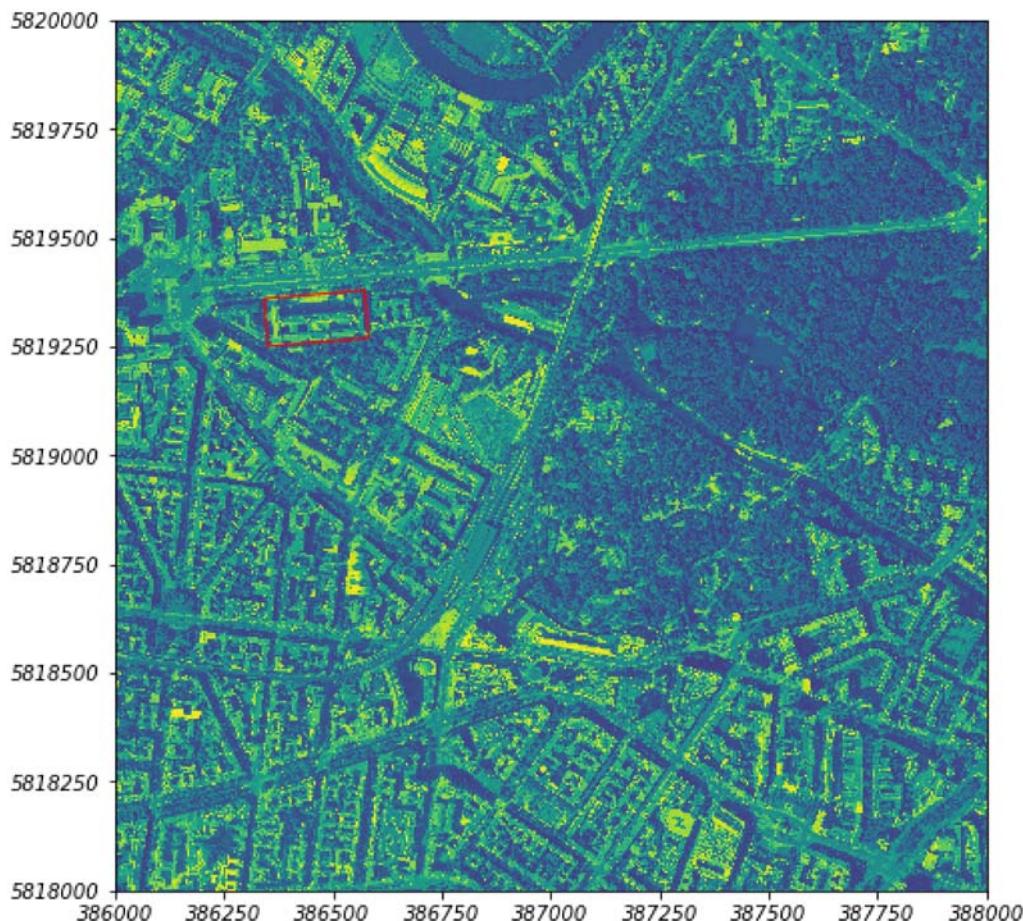
In [12]:

```
## Plot Shapefile over Raster
#Read Shapefile
shapefile = os.path.join(src, "Shape.shp")
shape = gp.read_file(shapefile)
#Properties
shape.crs
shape['geometry']
print(shape['geometry'][0])
```

```
POLYGON ((386341.94931751 5819360.158391257, 386571.9613358536 5819379.35568
4905, 386582.6264989916 5819270.21551546, 386350.4814480204 5819250.66271637
3, 386341.94931751 5819360.158391257))
```

In [13]:

```
#Plot
fig, ax = plt.subplots(figsize = (12, 8))
#Plot Raster
show((raster, 3), ax=ax)
# Plot Shapefile
shape.plot(color='none', edgecolor = 'red', ax = ax, alpha=1)
plt.show()
```



This tutorial was prepared with the support from Gabriel Cevallos. March 2020

