Illegal Mining Detection using Remote Sensing.

Santiago Saavedra and Mauricio Romero

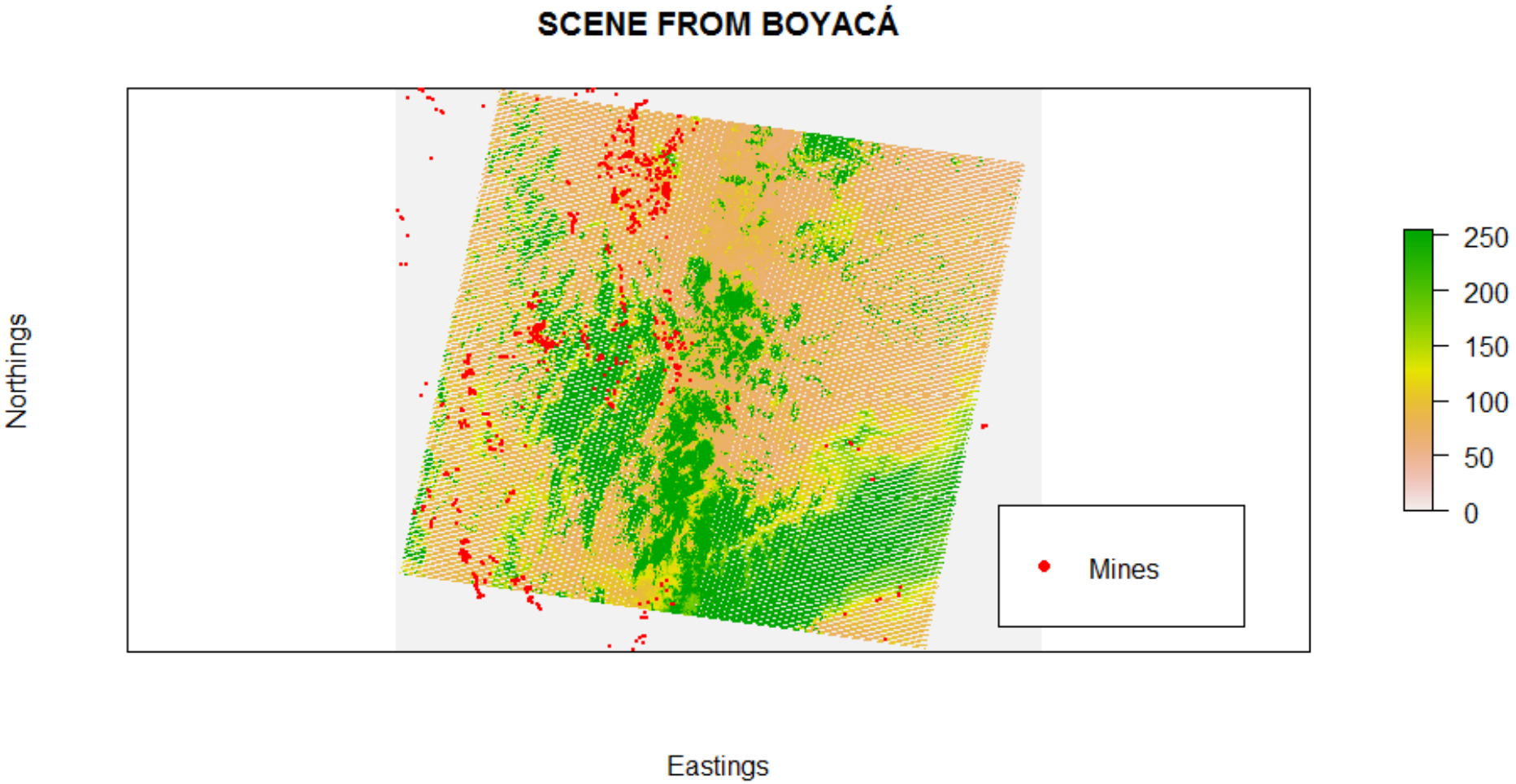
Table of Contents

1. Introduction
2. Data preprocessing
3. Cloud preprocessing
4. Data extraction
5. Principal Components Analysis
6. Subbagging and Logit Model
7. Raster of Predictions

Introduction

- Remote sensing of mining sites
- Landsat 7 data
- Machine Learning using Colombian mining census of 2010-2011.

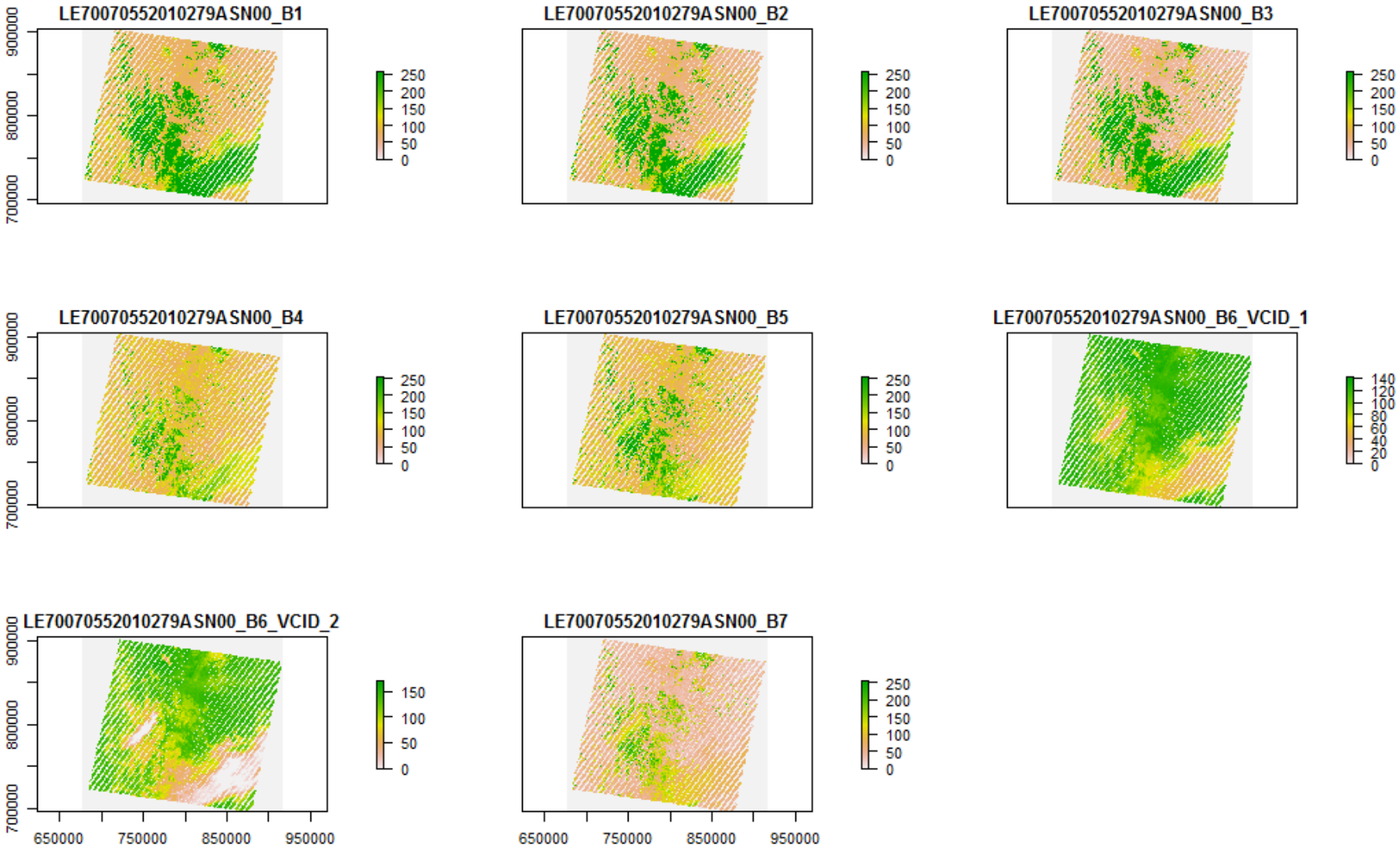
Introduction



Landsat 7

- Launched in 1999
- Multispectral images of the Earth
  - Eight color bands (30x30 mts) and one panchromatic band (60x60 mts)
  - In each band, values are coded in 8-bit data ranging from 1 to 255 according to reflected energy
- Each scene covers an area of about 49660 km^2
  - It takes 4 scenes to cover the median Colombian department.
  - Images are taken every 16 days
- Since 2003, due to Scan Line Corrector (SLC) failure, 22% of pixels damaged.
- **Cloud coverage**

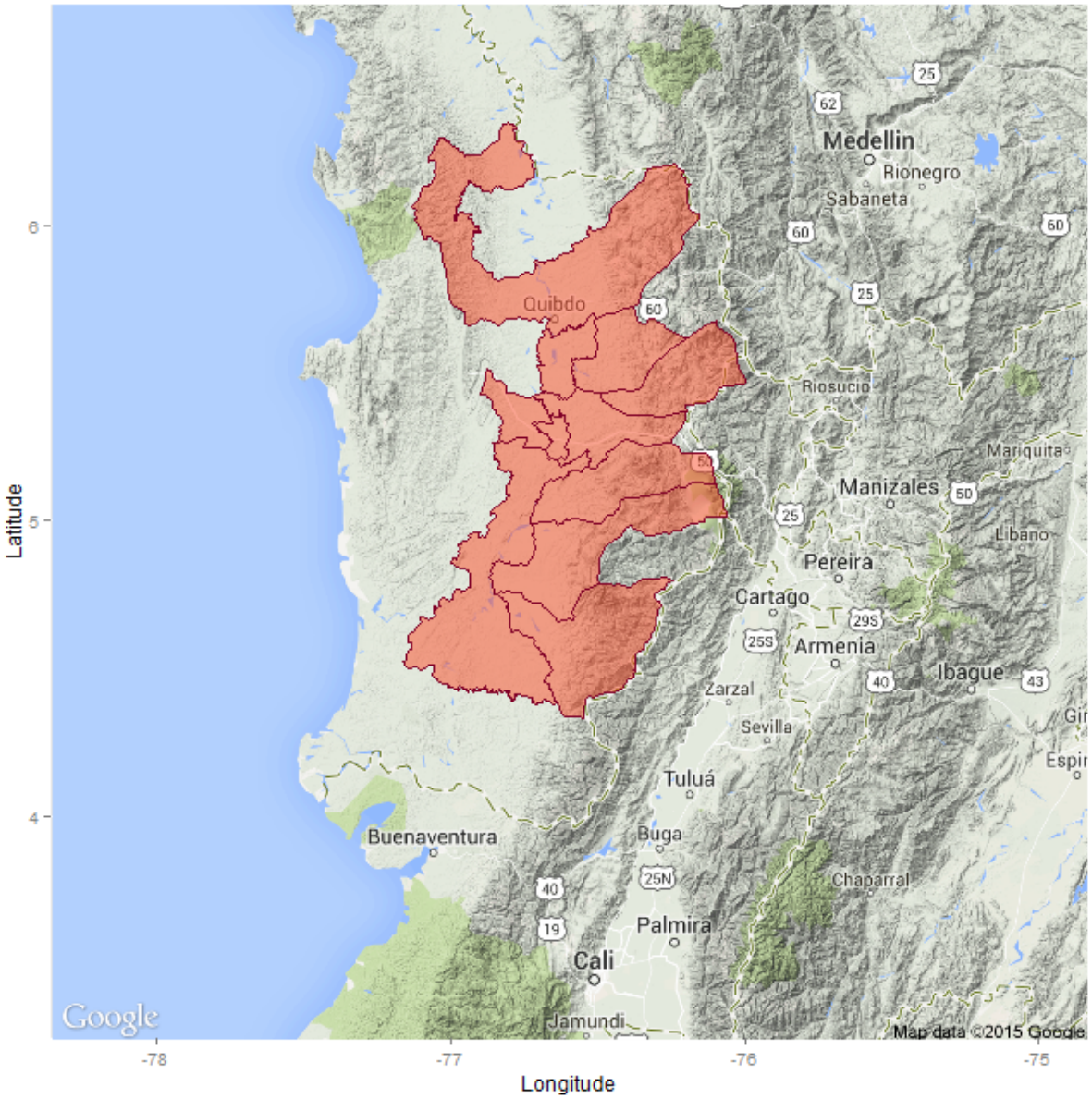
All layers in a Scene from Boyacá



Mining census 2010-2011

- Mining census in charge of the Mines and Energy Ministry of Colombia
- First phase covered 6 departments. Second phase covered 14.
- The census covered 14,357 observations mines across 20 departments, **62% of which are not formally registered**.
- Measurement problems with the georeferencing
  - Over 400 coordinates exceed the possible format.
  - Over 2000 observations are registered to different municipalities than the ones they are actually in

Municipalities in Chocó covered by census



New Municipalities

- The municipalities listed in the census were matched to a 2007 shapefile of Colombian municipalities using the DANE code.
- Some municipalities didn't have a match, because they were created after 2007.
- To fix this problem, we used a data base of DANE which provided information on the pre-2007 municipalities from which the new ones were formed.

Mismatched municipalities

- Using the functions **gContains()** and **gDistance()** from package *Rgeos*, we were able to verify whether the coordinates of a given observation were contained in the registered municipality.
- **2.374** mines had a mismatch.
- It is possible, however, that some mines were registered as belonging to the municipality from which it's easier to access them by road. Which is not necessarily the one were they are geographically located.

Distribution of distance to municipality

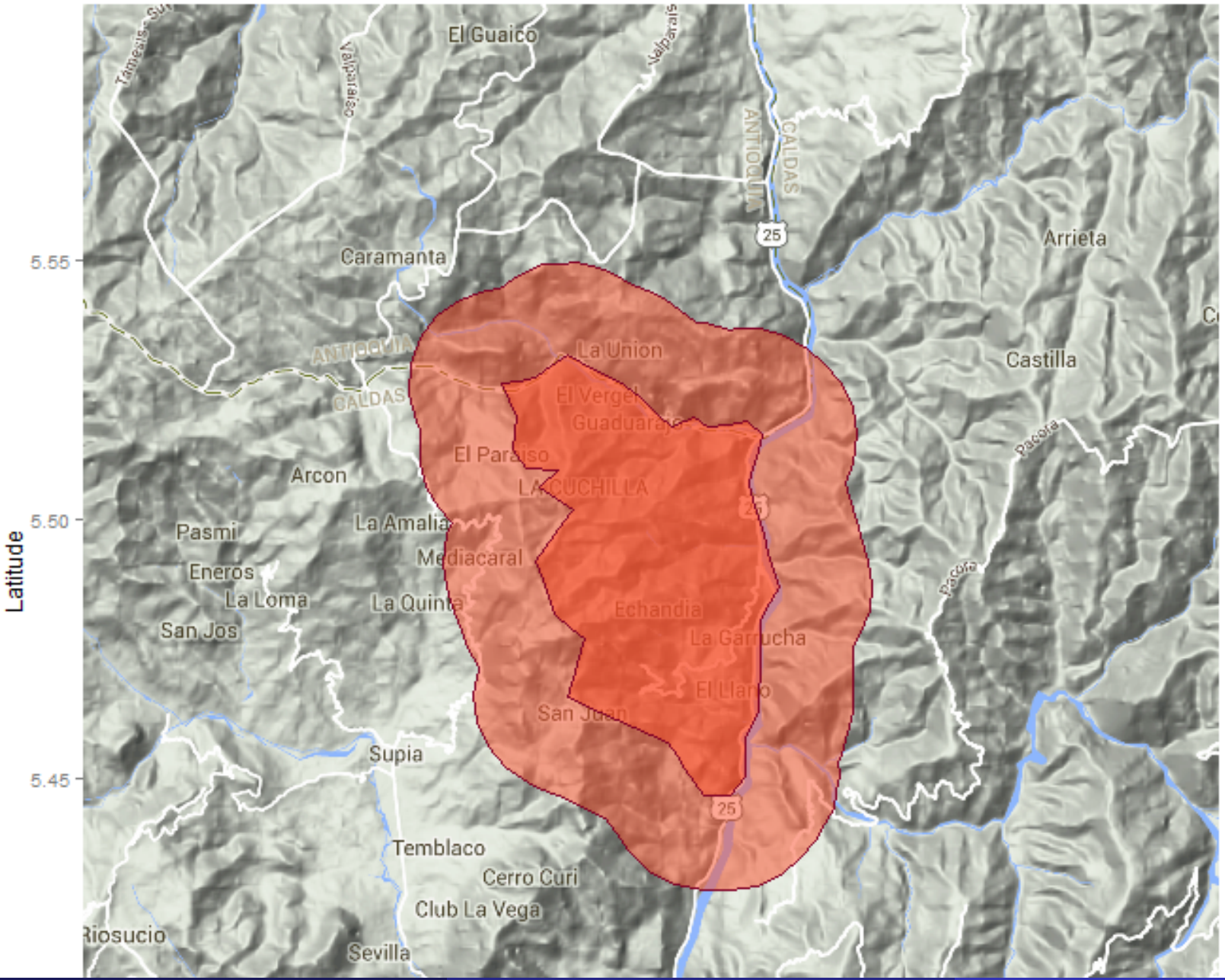
Distance	Mismatched mines
$<$ 10 M	62
$<$ 100 M	413
$<$ 1 KM	1275
$<$ 2 KM	1521
$<$ 10 KM	1965
$<$ 20 KM	2114

Mismatched municipalities

Solution

To avoid excluding from the analysis such mines that are tagged to the most accessible municipality, we created a 2 kilometer buffer around all municipalities, and then dropped all mines tagged to those municipalities that fell outside such buffer.

Municipality of Marmato, Caldas with Buffer



Mismatched municipalities

Potential bias

However, the mines that were excluded due to mismatching are systematically different from the others, as this mean-difference analysis reveals:

Mean Difference Analysis

Table 3: diferencia de medias

	Dependent variable:		
	met	num_frentes_exp	anos_prod
	(1)	(2)	(3)
accurate	0.017	0.161***	3.699***
	(0.014)	(0.054)	(0.366)
Constant	0.304***	1.070***	5.697***
	(0.013)	(0.051)	(0.350)
Observations	14,123	14,123	14,055
R^2	0.0001	0.001	0.007
Adjusted R^2	0.00004	0.001	0.007
Residual Std. Error	0.466 (df = 14121)	1.770 (df = 14121)	12.066 (df = 14053)
F Statistic	1.512 (df = 1; 14121)	9.091*** (df = 1; 14121)	102.289*** (df = 1; 14053)
Note:			*p<0.1; **p<0.05; ***p<0.01

Mean Difference Analysis

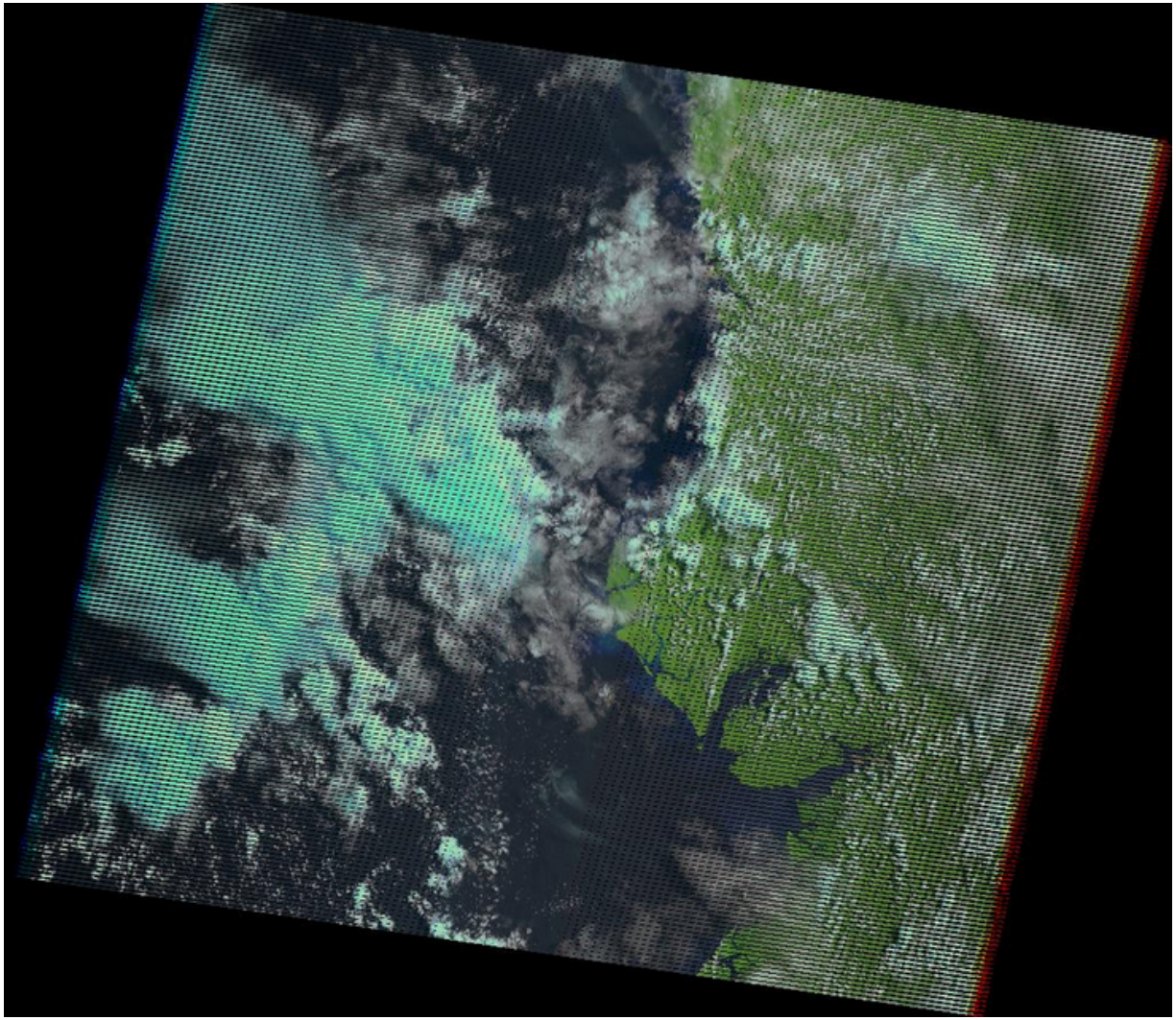
Table 4: diferencia de medias

	Dependent variable:		
	num_trabajadores	altitud	tipo_mina
	(1)	(2)	(3)
accurate	-2.292^{***}	-66.515^{*}	-0.099^{***}
	(0.570)	(34.001)	(0.015)
Constant	9.178^{***}	$1,250.940^{***}$	0.454^{***}
	(0.545)	(32.532)	(0.014)
Observations	14,107	14,122	14,123
R^2	0.001	0.0003	0.003
Adjusted R^2	0.001	0.0002	0.003
Residual Std. Error	18.847 (df = 14105)	1,124.128 (df = 14120)	0.480 (df = 14121)
F Statistic	16.161^{***} (df = 1; 14105)	3.827^{*} (df = 1; 14120)	46.534^{***} (df = 1; 14121)
Note:			*p<0.1; **p<0.05; ***p<0.01

Goal

- In order to train a model in "recognizing" mines, a cloudless image of the relevant territory for the relevant dates is needed.
- Very few scenes in this tropical part of the world have cloud coverage below 50%

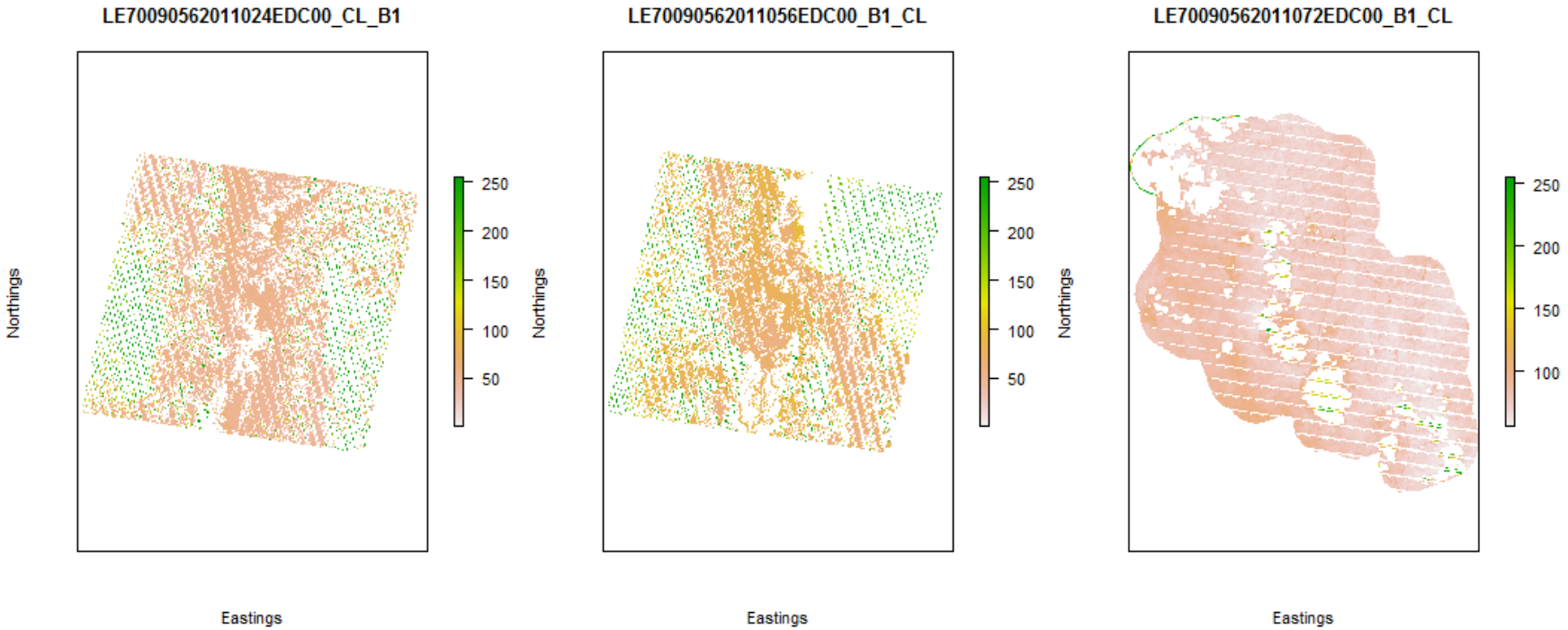
Clouds render some pixels useless



Cloud Masking

- Clouds can be identified, since they have values between 120 and 255 in the red band, and values between 102 and 128 in band 6.1, therefore a cloud mask can be easily built.
- To fill up the "holes" in scenes after removing the clouds, we take an average of the cloudless images of the entire 4 month period.
- As an added benefit, this also deals with those 22% damaged pixels and cancels out random deviations in colors due to haze, humidity and time of day.

Some Scenes from Caldas after removing Clouds



Cloudless composition

Implementation

The command **clouds()** from package *Landsat* allows the user to adjust a tolerance level in the identification of clouds, and produces the desired mask. Every cloudless scene is then saved to disk.

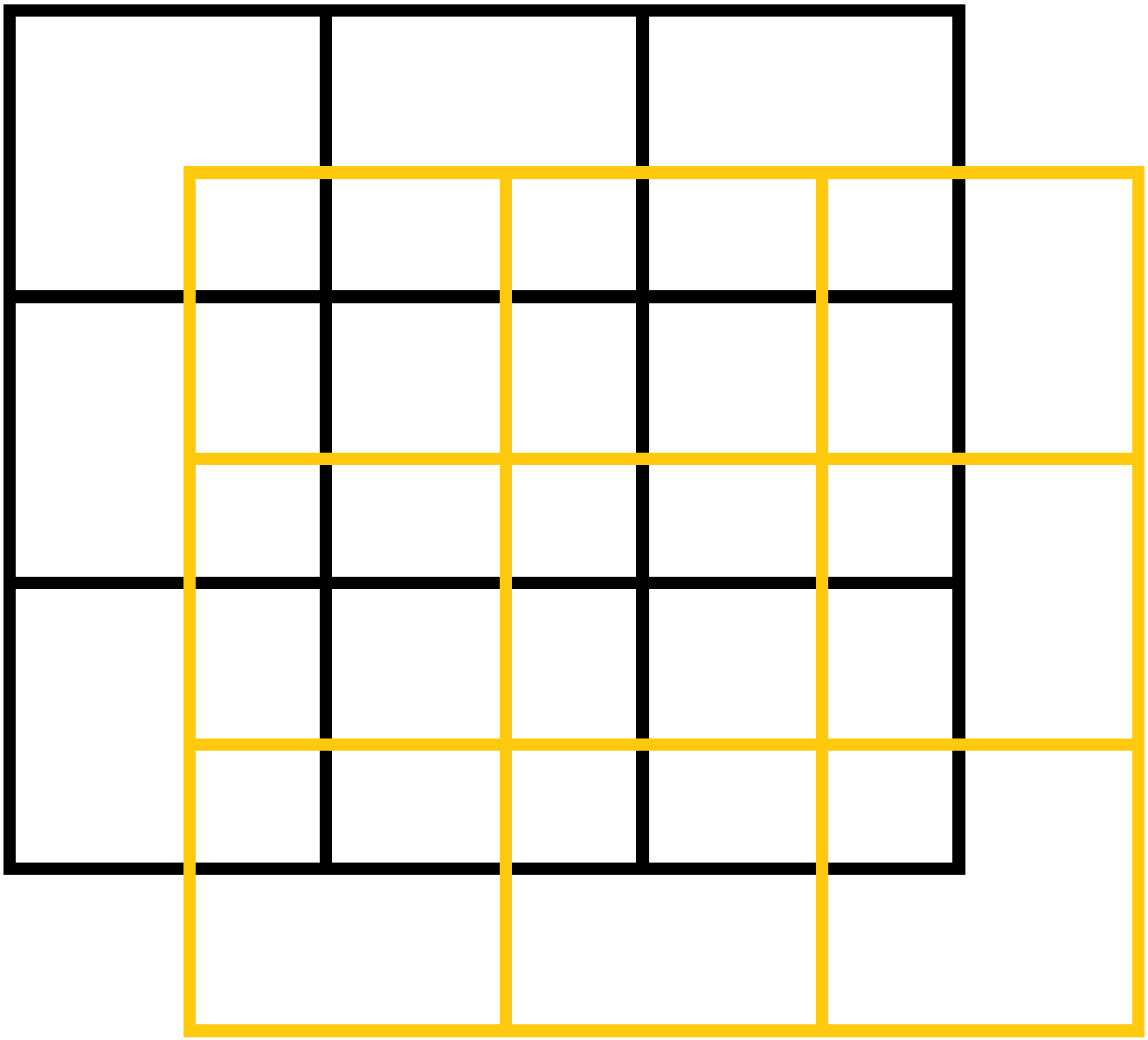
Complications

Damaged pixels are inconveniently set to 0. This is problematic when taking an average through different scenes, since 0 values are averaged with real values and corrupt the result. It's necessary to set 0 values to **NA**. To reduce computation time, the cloud masking and cloud composition process is performed only on municipalities covered by the mining census. To do this we mask the scenes to the shape of such municipalities.

Cloudless mosaic

- In order to merge or mosaic Rasters, these need to be "compatible". That is, have the same resolution and origin.
- Two rasters are compatible if the "extension" of the grid of pixels of one overlaps perfectly with the grid of the other.
- Scenes for the same department do not have "compatible" extents, in a series of satellite captures of the same location, slight shifts occur.
- In order to create the mosaic, the different scenes need to be made compatible.

Example of unaligned Rasters with equal resolution



Cloudless mosaic

Implementation

The process of making different Rasters compatible is called **resampling**. It takes a target raster with the desired resolution and origin and interpolates the values of centres using a bilinear interpolation of the values in the corners (which can be retrieved from the original raster). R implements this with the call **resample(input.raster, target.raster, method="bilinear")**.

Limitations

This process is very computationally expensive, and increasingly expensive depending on the size of the Raster. To make things feasible, the resolution of every raster is previously reduced by a factor of 3. This is accomplished with the function **aggregate(input.raster, fact=3, fun=mean, expand=TRUE, na.rm=TRUE)** of the package *Raster*. This process may reduce prediction power.

Cloudless mosaic

Implementation

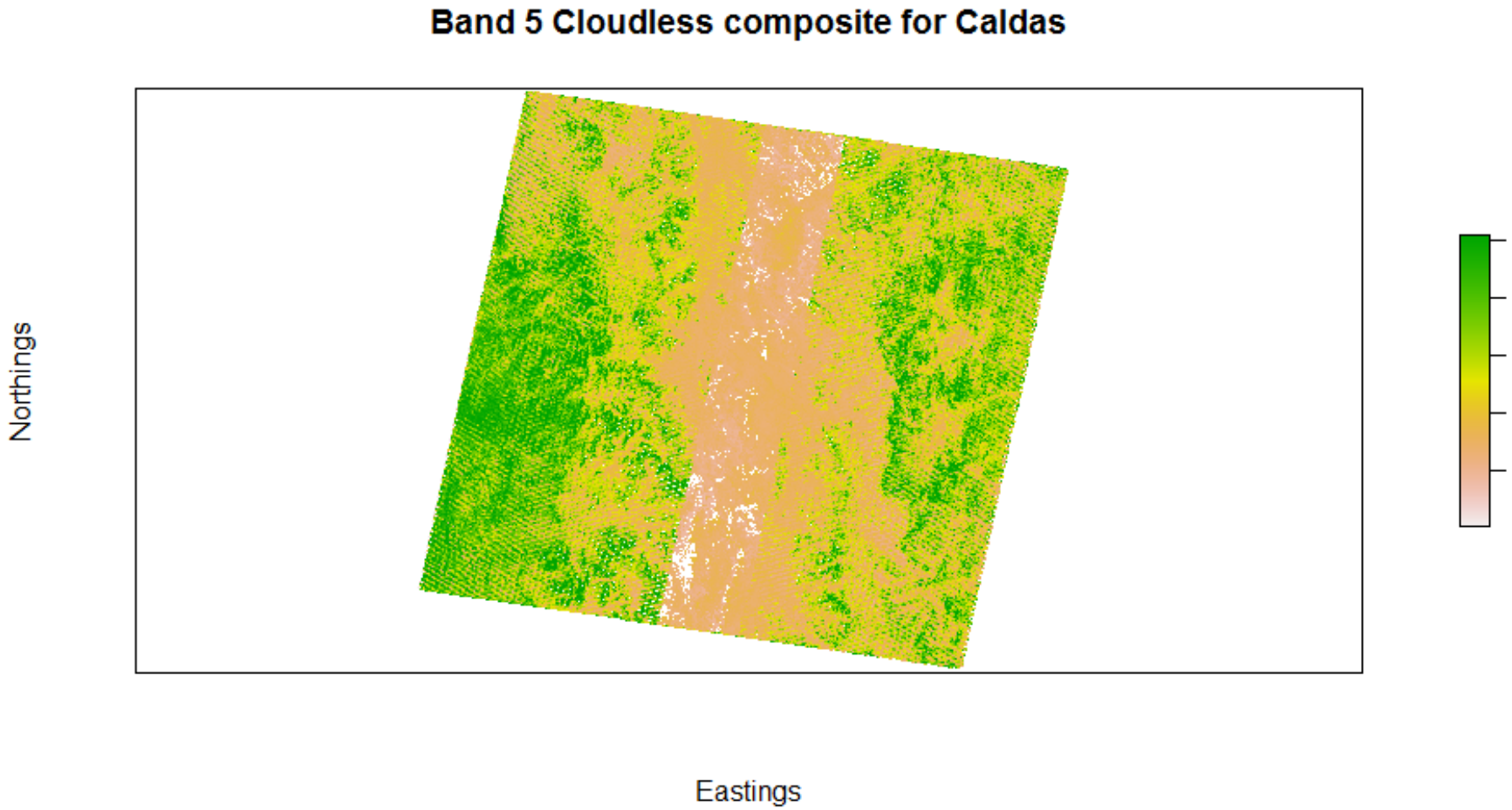
The compilation of compatible rasters is then done using the call **do.call(mosaic, resampled.rasters)** which results in a large cloudless raster for each department.

Limitations

Putting together the mosaic requires writing each scene several times in RAM. This can overload even highly endowed computers. To avoid this, it's necessary to "split" the process a few times.

Mosaic with no clouds

After the mosaic, we obtain a virtually cloudless composite of the scene:



Supervised classification

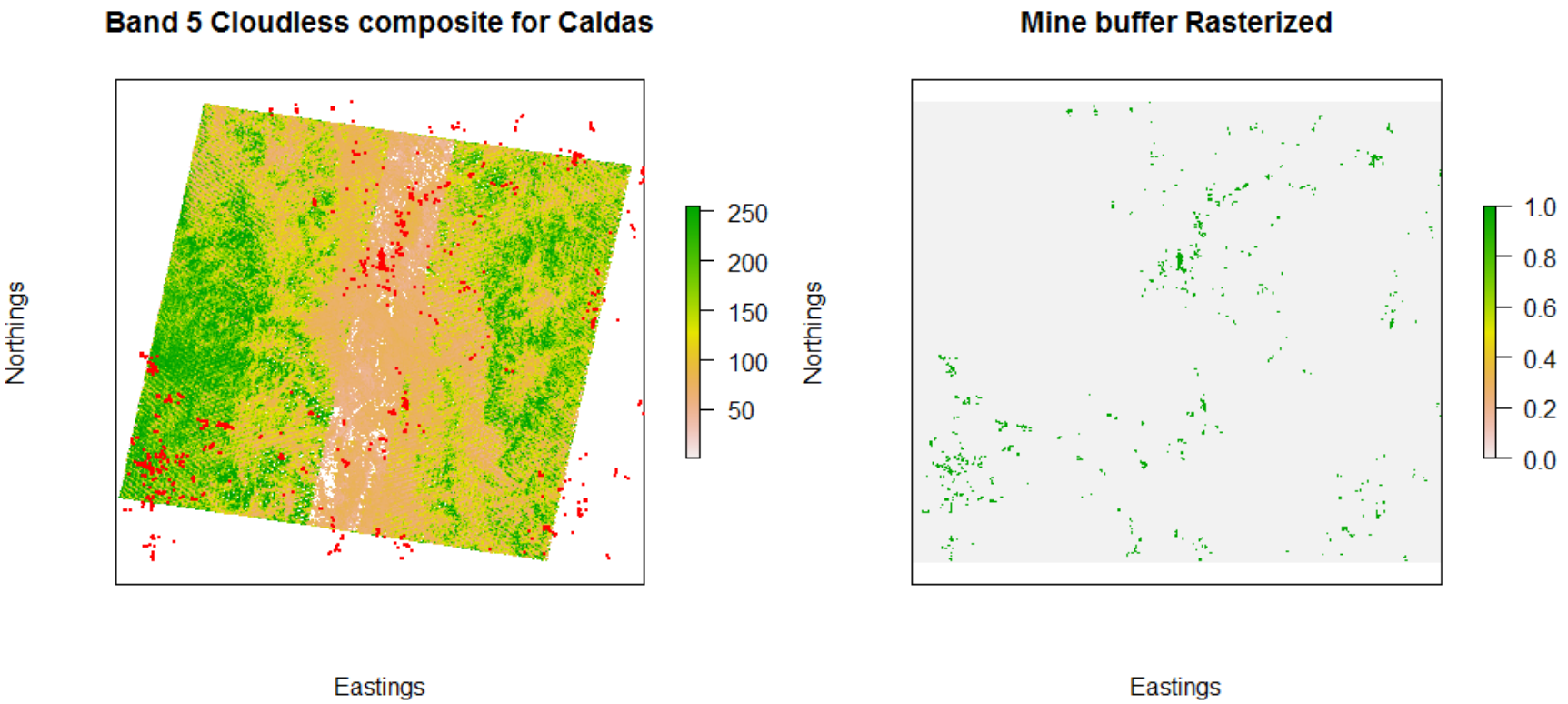
- We want a model that can classify every pixel as having a mine or not.
- The coordinates of a mine are a single point in space. Mines occupy a certain area.
- Using the function **gBuffer()** of the package *Rgeos*, we create a 200 meter buffer about the mines and signal those areas as having a mine.
- Then we create an additional layer in the large cloudless raster signaling the mines.

Supervised classification

Implementation

To create the "Mine" layer, we rasterize the shapefile of the buffer around the mines. This is done with the function **rasterize()** of the package *Raster*. This function takes the resolution and origin of a template raster; in this case we can take any of the layers of the cloudless raster.

Creating a "mine" layer



Extraction

- The input for the classification model is a data frame that contains a row for each pixel and in the columns it has the value of each band and the dummy variable "Mine".
- To get further prediction power in the future, it is convenient to have the municipality code of each pixel. This way we can attach more variables at the municipality level.
- We would also like to select 80% of municipalities as **training sample** and the rest as a **test sample** to perform cross validation.

Extraction

extract()

For each department we extract, municipality by municipality using the command **extract()** of the package *Raster*, and setting as a parameter the municipality to be extracted. To the resulting data.frame we add a column with the municipality DANE code and in 80% of municipalities (chosen randomly) we add a column with a boolean set to **TRUE**. In the others, we set it to **FALSE**.

Extraction

The resulting data.frame looks like this:

	row.names	ID	composite__B1_CL	composite__B2_CL	composite__B3_CL	composite__B4_CL	composite__B5_CL	composite__B6_VCID_1_CL	composite__B6_VCID_2_CL	composite_B7_CL	mina
1	13	4	92	78	74	87	82	120	128	53	1
2	14	4	84	70	61	96	83	117	124	46	1
3	15	4	90	75	70	93	79	119	127	47	1
4	16	4	88	74	68	91	79	117	124	45	1
5	17	5	81	67	57	98	84	127	143	44	1
6	18	5	88	72	67	80	74	125	138	43	1
7	19	5	82	67	58	98	80	126	140	42	1
8	20	5	91	77	73	83	70	123	134	41	1
9	21	6	90	77	74	82	74	124	136	49	1
10	22	6	93	79	78	81	77	122	134	50	1

Principal Components

- The combined data.frame of all municipalities is quite large, running the model on fewer variables
  can result in considerably shorter computation times.
- Additionally, different bands of a satellite images are often highly correlated making them redundant.
- Performing Principal Component Analysis we effectively reduce the dimension of the problem and
  eliminate potential problems derived from high correlation between the independent variables.

Principal Components

Implementation

The principal components of the bands are obtained using the command **prcomp()** of the basic R package. Tolerance is set as to only include components whose standard deviations exceed 10% of the main component's standard deviation.

Principal Components for Caldas Example

In a pilot example for Caldas, we obtain 3 components:

row.names	PC1	PC2	PC3
composite__B1_CL	0.4330225	0.2537093	-0.2263035
composite__B2_CL	0.4648452	0.1089961	-0.2117483
composite__B3_CL	0.5117249	0.09004993	-0.3279787
composite__B4_CL	0.2844758	-0.1551854	0.7448691
composite__B5_CL	0.376036	-0.4745024	0.2104693
composite__B6_VCID_1_CL	-0.05624693	-0.3819897	-0.2163561
composite__B6_VCID_2_CL	-0.09909938	-0.6830012	-0.3807399
composite_B7_CL	0.3150144	-0.2323806	0.07402143

Unbalanced classification

- Our training sample has very unbalanced classes. Over 99.9% of observations are not mines.
- Simply running a Logit model could yield a prediction of **NO** everywhere and have an error rate of less than 0.1%
- The solution is to take several "balanced" subsamples, and calibrate the model there. Then take an average of the resulting coefficients. This is known as "Subbagging".

Subbagging

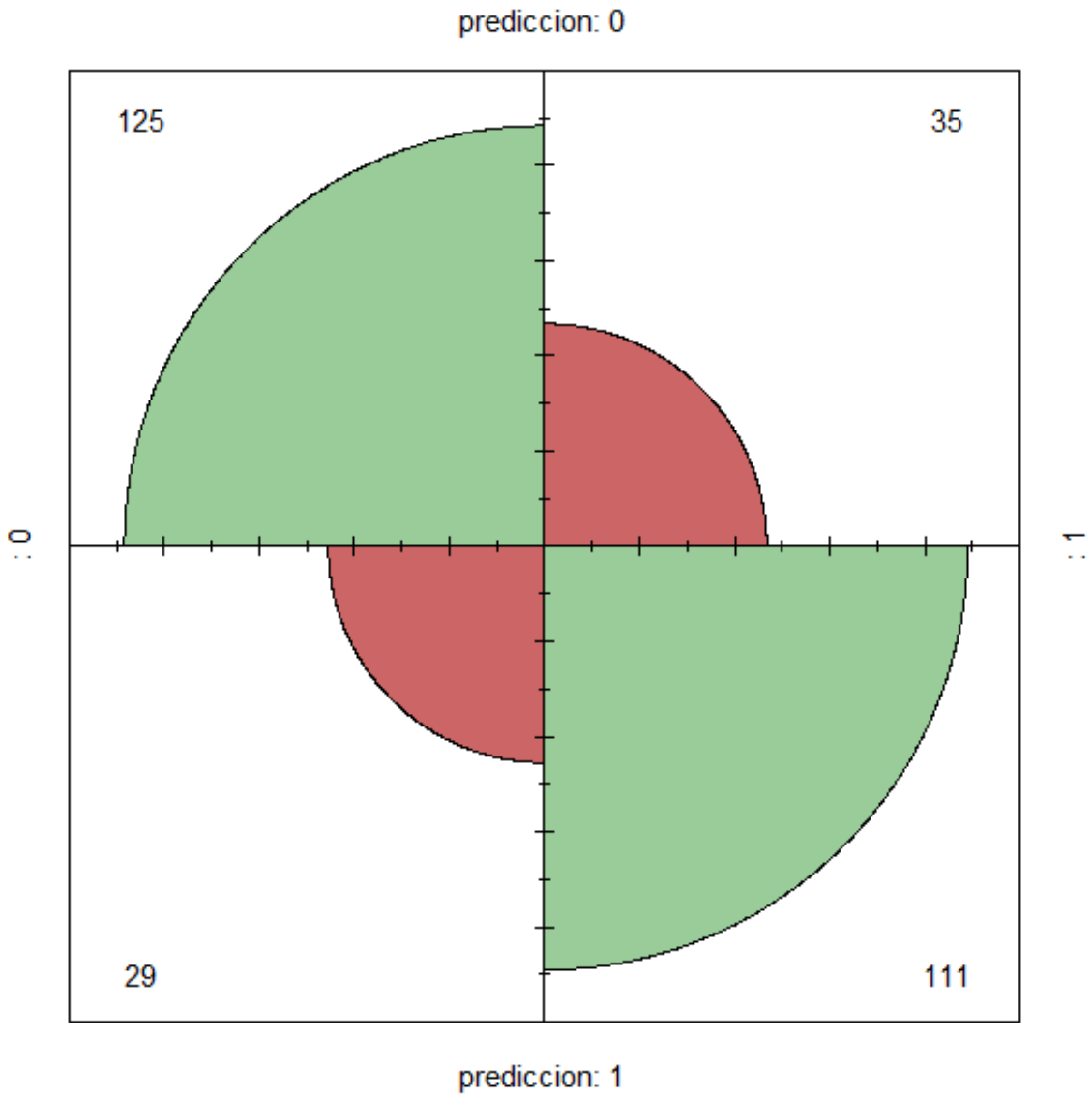
- To do the Subbagging, we create 1000 index vectors consisting of the observations with mine and an equal number of randomly selected observations with no mine.
- For each of them we calibrate a Logit model and store the resulting β in a list.
- We then take the average of the estimated coefficients to be our true calibration.

Cross Validation

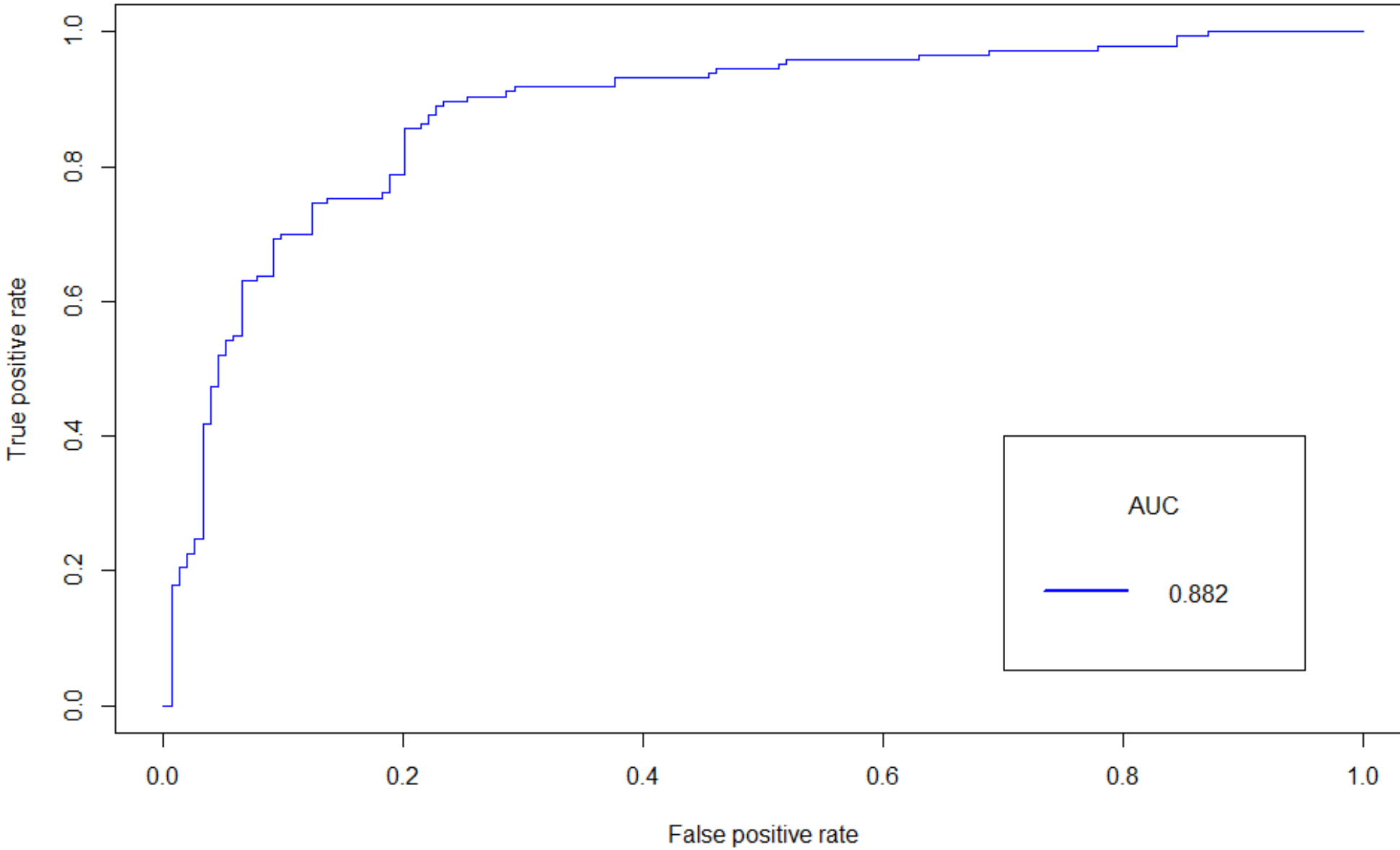
Implementation

In order to test our model, we run it on the test municipalities, producing a column of **predicted probabilities** which we can use to compute a **confussion matrix**, and measure the **power of prediction** with a **ROC curve**.

Confussion Matrix for Caldas



ROC curve for Caldas



Predicting

- Using the estimated coefficient of the previous excercise and the transformation matrix of the PCA
  we can build a Raster that "maps" the prediction of a mine over an existing Raster.
- In the following example we map the predictions of our model over a scene that covers part of the department of Vichada.

Prediction on Vichada

IMAGE MISSING