R project

Course: Monitoring ecosystem changes and functioning – 88271

Prof.re: Duccio Rocchini

Student: Simone Celebrin

A.A. 2019/2020

Almamater Studiorum Università di Bologna

Aims:

This spatial-ecological monitoring project wants to analyse some environmental controls using the R software.

Specifically, we will analyse:

- CO_2 — The temporal evolution of CO2 from 2000 to today, predicting the expected new CO2 levels
- NDVI — Diachronic analysis of the Normalized Difference Vegetation Index
- G.M — General Model used like proxy of land vulnerability to heat

1) CO_2 temporal evolution

In order to analyse CO_2 temporal evolution we download a spatial dataset composed of 17 images with 12 bands each, one for month.

> rlist<-list.files(pattern="odiac")

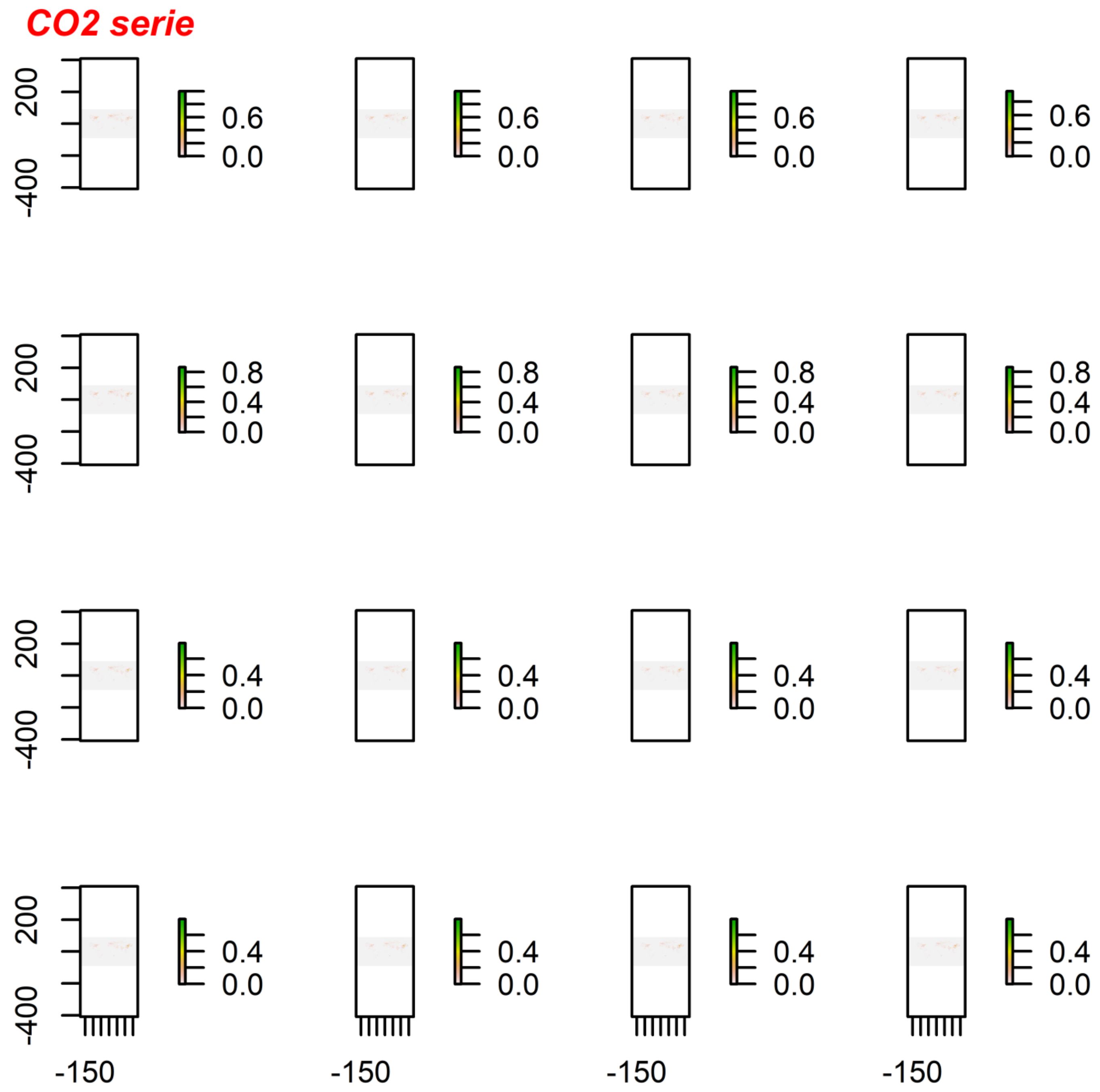
> import<-lapply(rlist,brick) In order to summarise the 12 bands into only one layer we do a PCA:

> PCAS<-lapply(import,rasterPCA)

The PC1s usually rapresent over the 99% variability so we use only them, and we standardize:

> PC1.s1<-(PCAS[[(n)]]mapPC1)/maxValue(PCAS[[(n)]]mapPC1) with n=(1:17)

At the end we create a stack in order to obtain a 17 bands tiff representing CO_2 evolution



CO2 series

In this image we can observe the evolution of the CO2 during the last 16 years.

CO2 prevision

Using historical images saw before we create a prediction of future CO2 levels

```
> source("predictionCo2.r")
```

The used “predictionCo2.r” code is written on a .txt file, recalled by source function

```
> require(raster)
> require(rgdal)
# define the extent
> ext<- c(-180, 180, -90, 90)
> extension <- crop(serieC02, ext)
# make a time variable (to be used in regression)
> time <- 1:nlayers(serieC02)
# run the regression
> fun <- function(x) {if (is.na(x[1])){ NA } else {lm(x ~ time)$coefficients[2] }}
> predicted.co2 <- calc(extension, fun)
```

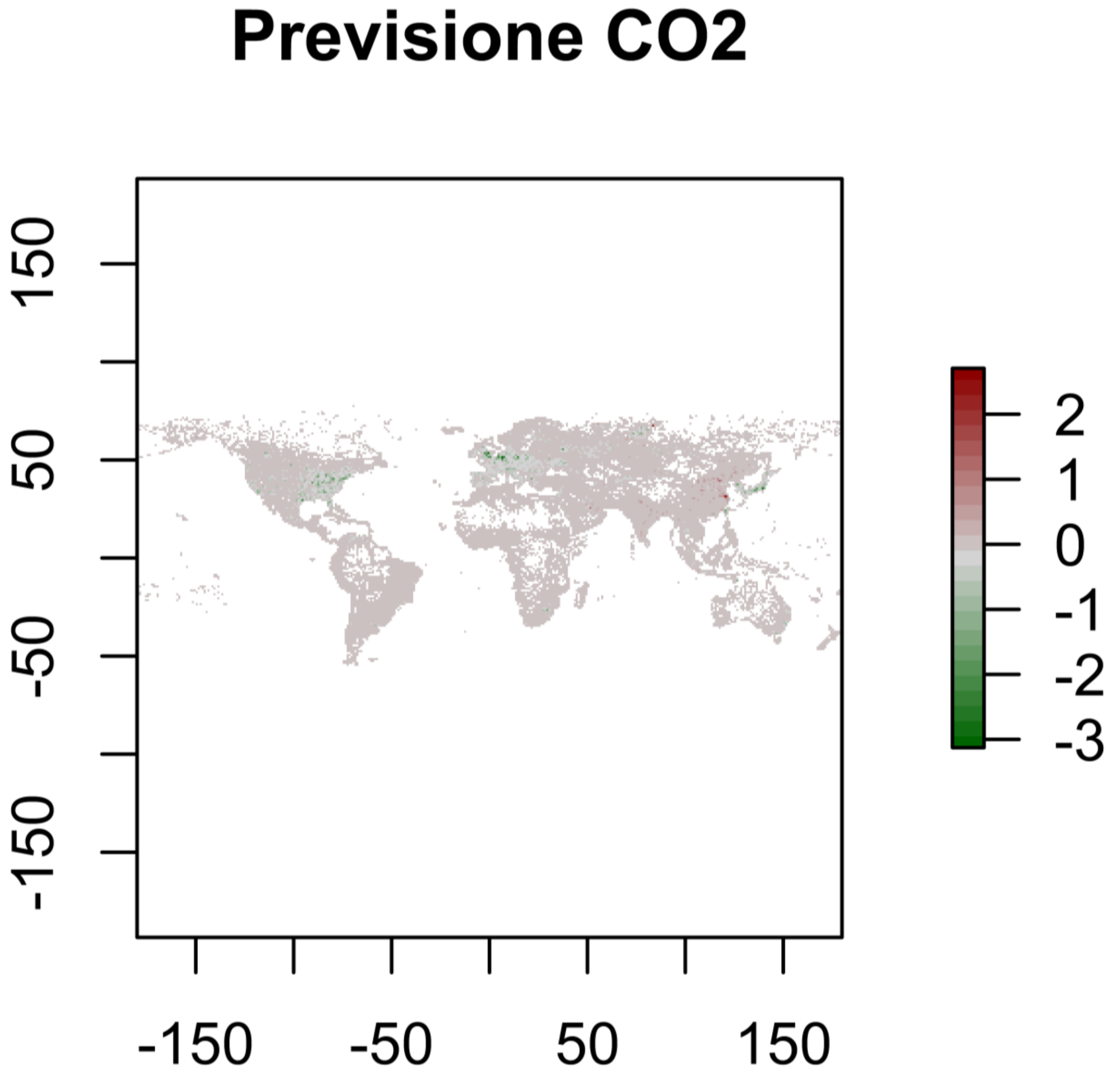
CO2 prevision

The raster CO2 prediction has a coloured background, so to transform background values into NA values and obtain a white background we use the following code:

```
>click(predicted.co2, n=Inf, id=FALSE, xy=FALSE, cell=FALSE, type="n", show=TRUE)
>x<-reclassify(predicted.co2, cbind(6.75477e-05,6.75478e-05,NA))
```

Then to see bigger values we multiply the raster images by 100

```
>x100<-x*100
>plot(x100, col=cl)
```



Other interesting images obtained by CO2 time serie:

1. CO2 difference: we calculate the CO2 difference between 2018 and 2002:

```
> dif<- (serieC02$PC1s_all.17 - serieC02$PC1s_all.1)
> click(dif, n=Inf, id=FALSE, xy=FALSE, cell=FALSE, type="n", show=TRUE)
> d <- reclassify(dif, cbind(0.0007800222,NA))
> plot(d)
```

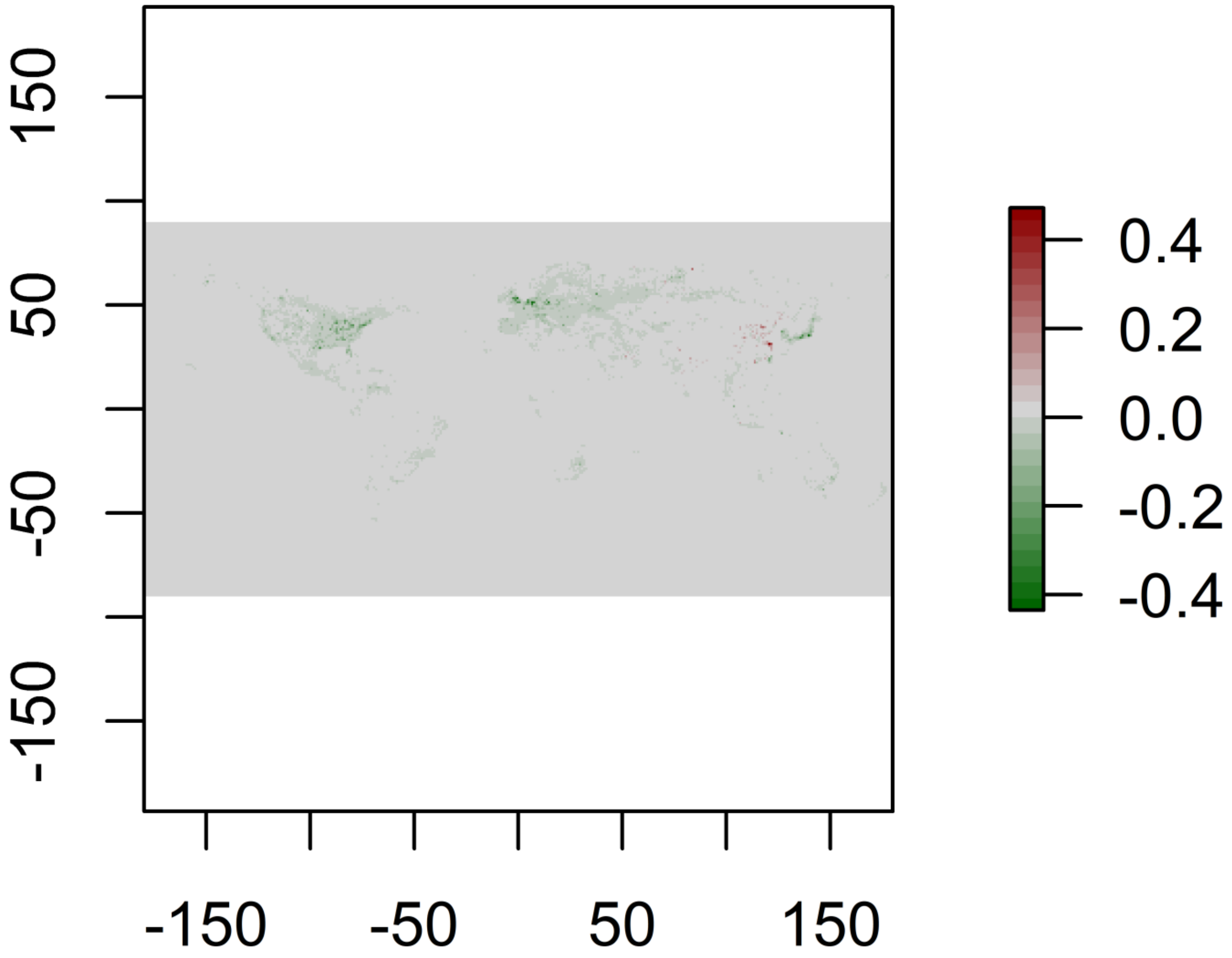
2. Correlation between prevision and difference:

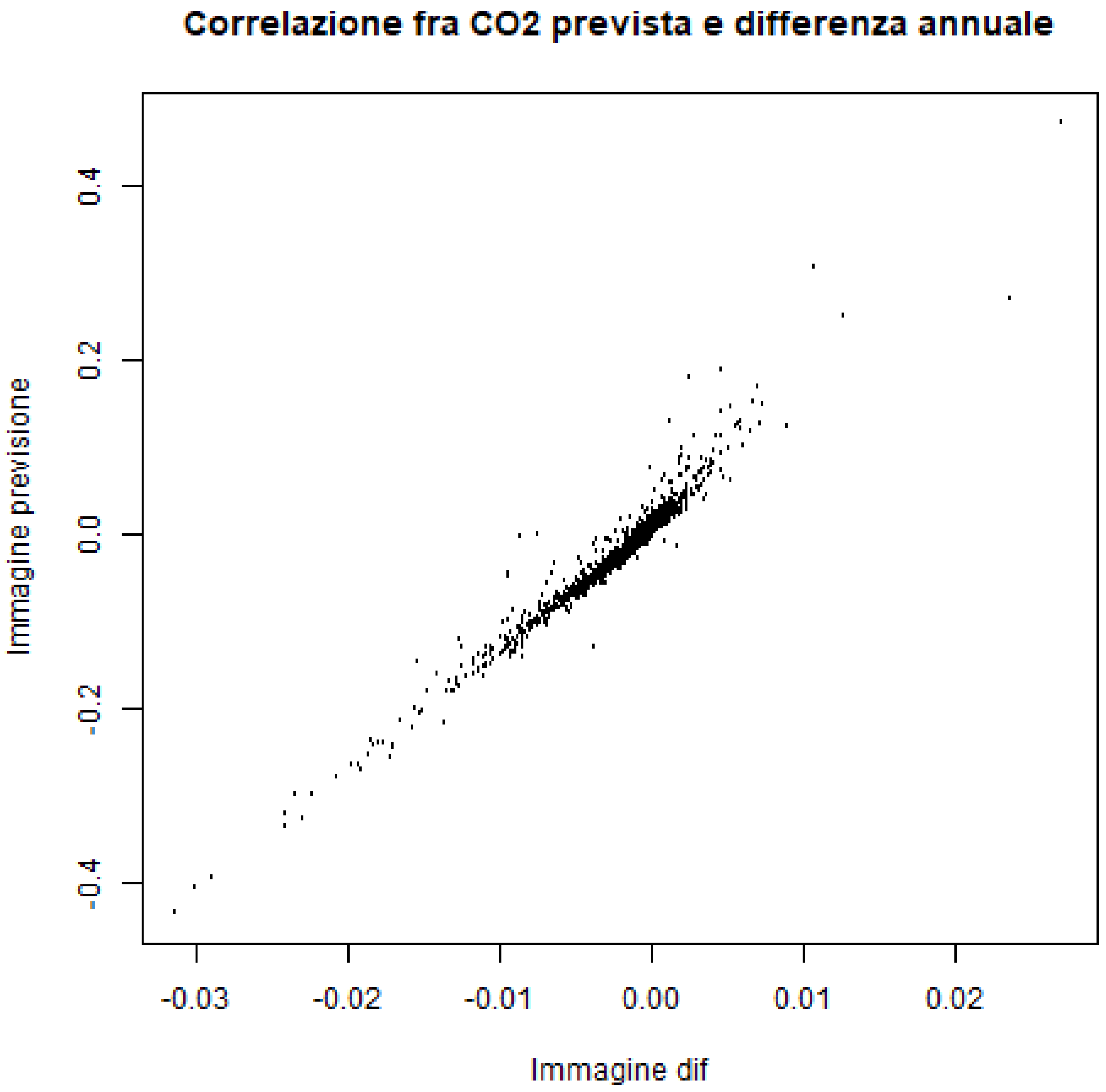
```
> plot(x,d)
```

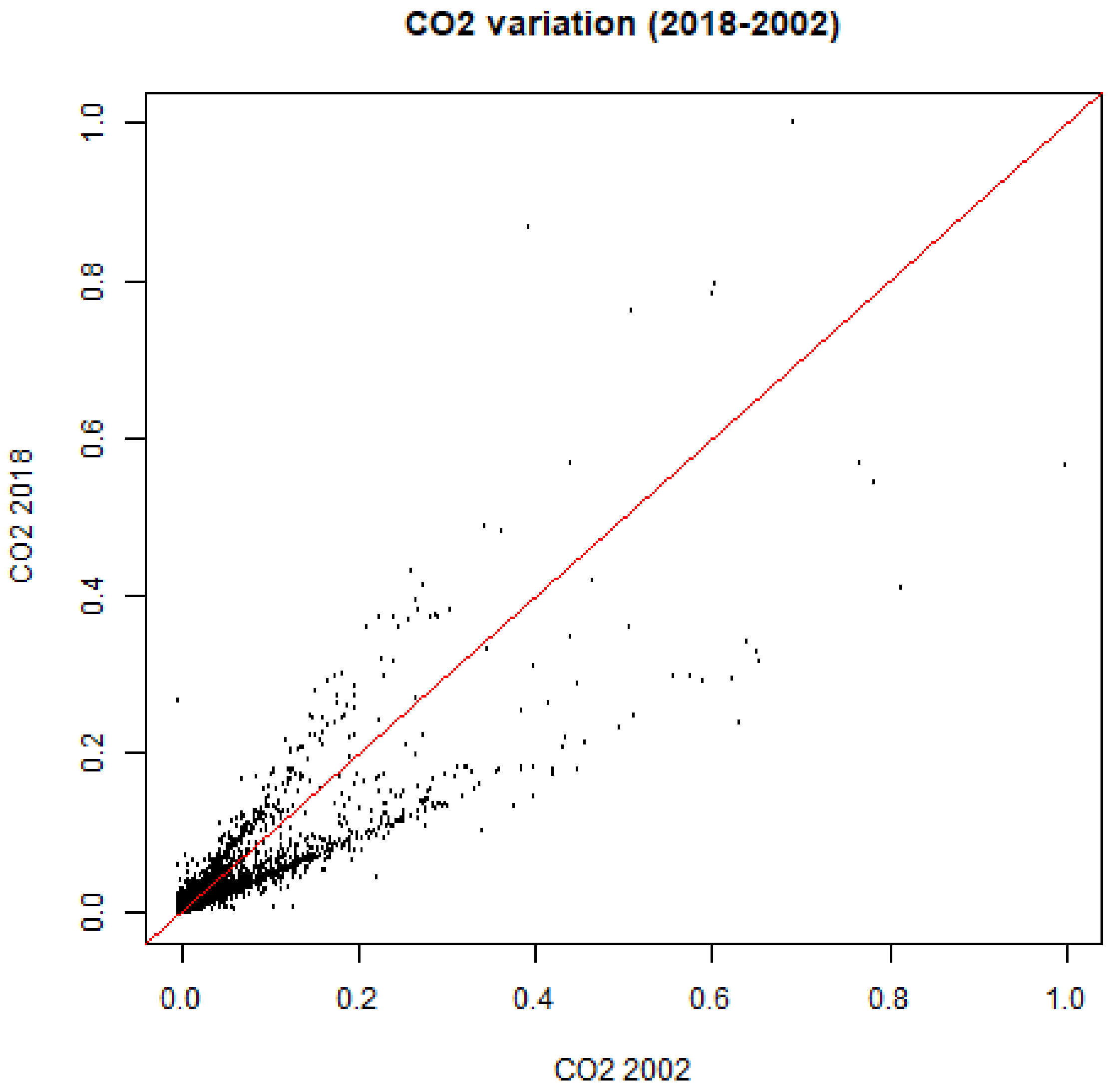
3. Time trend: we put image 1 on x ax and image 17 on aix y and we obtain a 45° line that describes correspondence 1 to 1. In this graph the data will be under this line if the image 1 has higher values.

```
> plot(serieC02$PC1s_all.1, serieC02$PC1s_all.17, main="CO2 variation (2018-2002)", ylab="CO2 2018", xlab="CO2 2002")
> abline(0,1,col="red")
```

Differenza CO2 2018-2002




Correlazione fra CO2 prevista e differenza annuale
Immagine previsione
Immagine dif
-0.4
-0.2
0.0
0.2
0.4
-0.03
-0.02
-0.01
0.00
0.01
0.02


CO2 variation (2018-2002)
CO2 2018
CO2 2002
0.0
0.2
0.4
0.6
0.8
1.0

NDVI EVOLUTION OVER THE TIME

NDVI: simple graphical indicator that can be used to analyze remote sensing measurements, assessing whether the target being observed contains live green vegetation.

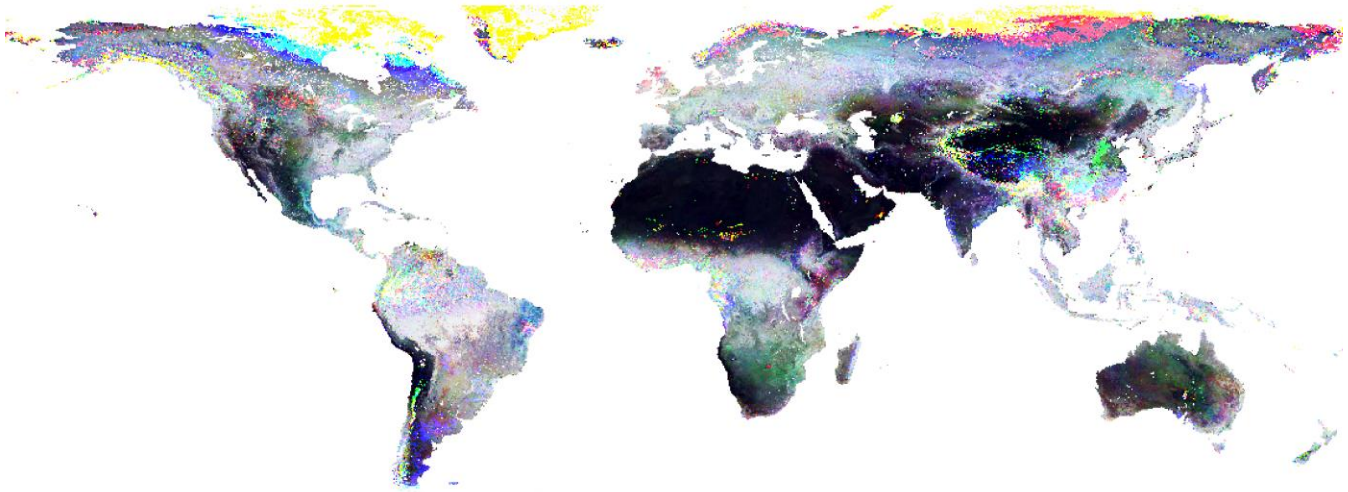
```
> rNlist<-list.files(pattern="c_gls_NDV")
> importN<-lapply(rNlist,raster)
> NDVI.multitemp<-stack(importN)
> plot(NDVI.multitemp$Normalized.Difference.Vegetation.Index.1KM.1)
```

To understand the background values

```
> click(NDVI.multitemp$Normalized.Difference.Vegetation.Index.1KM.1, n=Inf, id=FALSE, xy=FALSE, cell=FALSE, type="n", show=TRUE)
```

Let's transform background values into NA values

```
> NDVI.multitempR<- calc(NDVI.multitemp, fun=function(x){ x[x > 0.936000] <- NA; return(x)} )
```



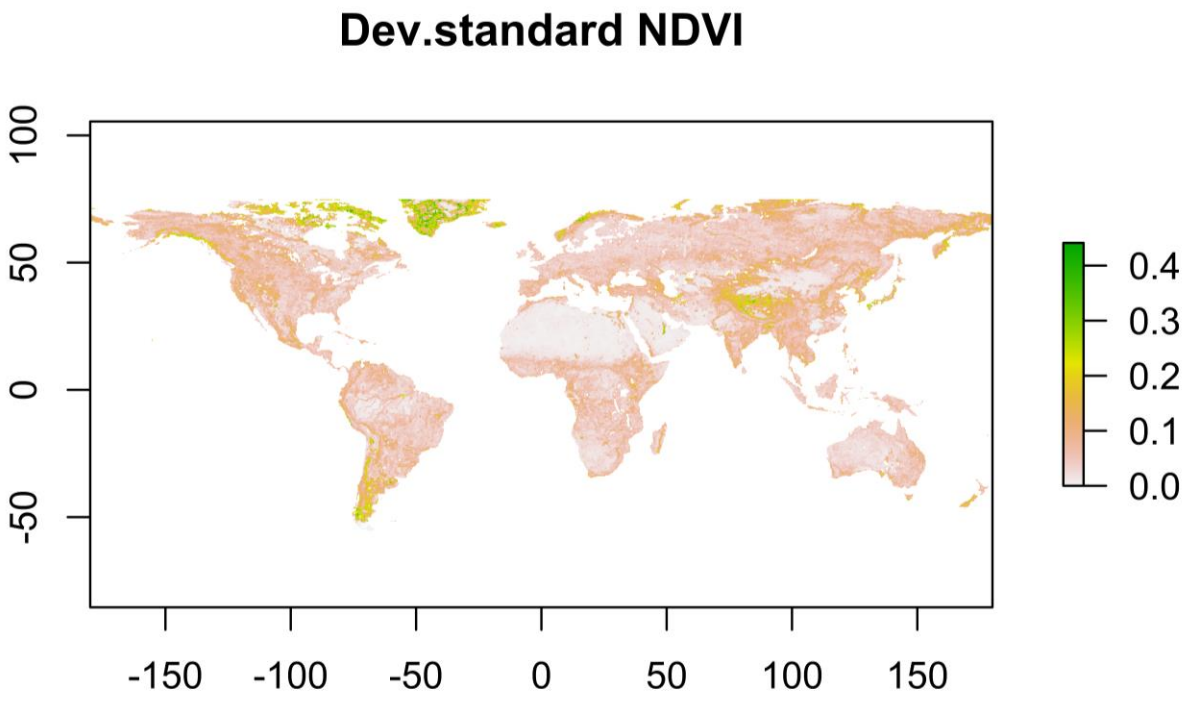
NDVI plot in RGB

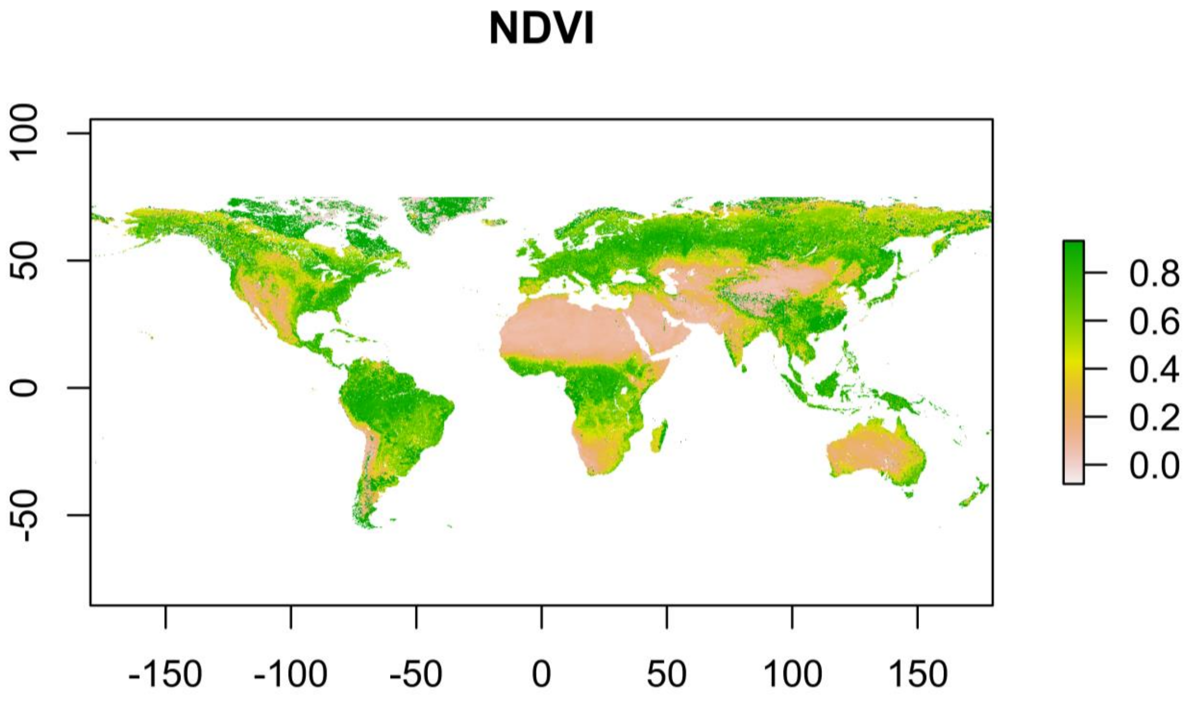
We plot 3 different NDVI periods in rgb, so where there are higher values the image takes the red, green or blue color. So we understand in which year there's been the higher values and where.

Red = 1998

Blue= 2010

Green= 2020





NDVI STANDARD DEVIATION

We want to understand the NDVI standard variation.

let's create a moving window

window <- matrix(1, nrow = 5, ncol = 5)

Focal is the function to move the window, it calculate standard deviation values (in this case) for the neighborhood of focal cells.

> aggr_NDVI <-aggregate (NDVI.multitempR$Normalized.Difference.Vegetation.Index.1KM.6, fact=10)

>sd_str<- focal(aggr_NDVI, w=window, fun=sd)

>plot(NDVI.multitempR$NDVI_corr.6, main="NDVI")

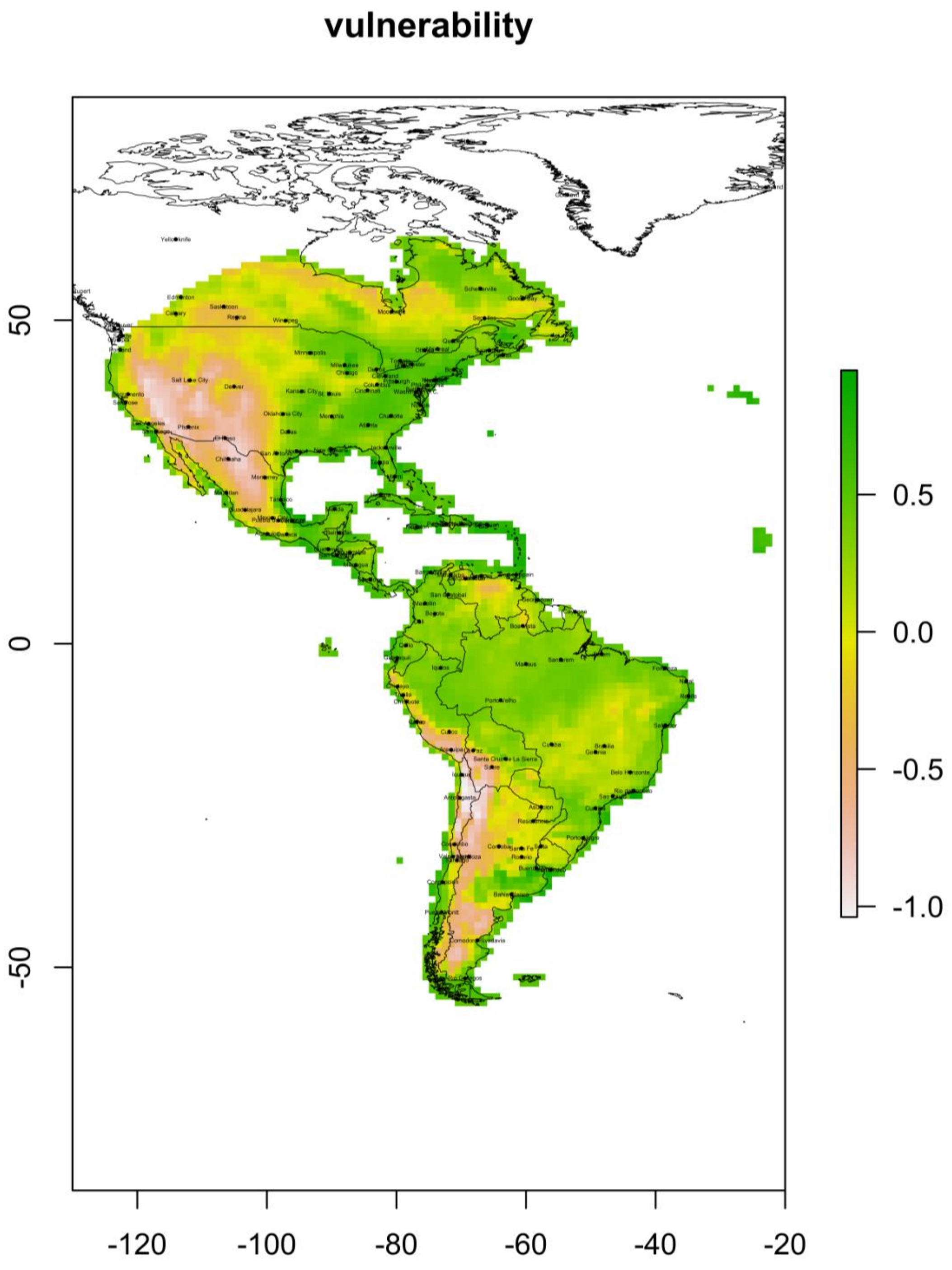
GENERAL MODEL:

Let's create a general model that represent into one layer: NDVI, C02, Temperature and built ground cover. In this way with a PCA we can summarize into one layer these variables. We obtain 4 dimensions PCA: PC1, PC2, PC3, PC4, hat can be use like a proxy of land vulnerability to heat.

```
> NDVI2020<-raster("c_gls_NDVI_202006010000_GLOBE_PROBAV_V2.2.1.nc")
> serieC02<-brick("PC1s_all.tif")
> CO2ult<-serieC02$PC1s_all.1
>Temper2020<-raster("c_gls_LST10-DC_202006110000_GLOBE_GEO_V1.2.1.nc")
>Costru2020<- raster("lulc-human-modification-terrestrial-systems_geographic.tif")
```

We give to images the same size and resolution of CO2ult (the smallest)

```
> NDVI2020r <- resample(NDVI2020, CO2ult, resample='bilinear’)
> Temper2020r<-resample(Temper2020, CO2ult, resample='bilinear’)
> Costru2020r<-resample(Costru2020, CO2ult, resample='bilinear’)
> modelvariables<-stack(NDVI2020r,Temper2020r,CO2ult,Costru2020r)
```



VULNERABILITY

We do the PCA

> vuln<-rasterPCA(modelvariables)

Use PC1 and PC2 and standardize them

> vulnPC1_stand<-(vulnmapPC1)/maxValue(vulnmapPC1)

> vulnPC2_stand<-(vulnmapPC2)/maxValue(vulnmapPC2)

> vulntot<-(vulnPC1_stand+vulnPC2_stand)

> vulntot_stand<-vulntot/maxValue(vulntot)

Then we crop only the Americas

> ext <- c(-130, -20, -80, 80)

> extension <- crop(vulntot_stand, ext)

And we put the main cities on the map and the borders of the states

Thanks for the attention