

# colrow: Handling SimU, CR, and LU data

*Pedro R. Andrade*

*setembro 21, 2018*

## Abstract

This package has functions to handle Simulation Units (SimU), ColRow (CR), and Large Unit (LU) data.

## Introduction

This package allows users to handle CR, SimU, and LU data. In order to start, it is necessary to install the package from GitHub.

```
devtools::install_github("pedro-andrade-inpe/colrow")
```

## Creating shapefiles for a given country

First load the package.

```
require(colrow)
```

```
## Loading required package: colrow
```

```
## colrow - Handling SimU, CR, and LU data
```

```
## Version 0.1-0 is now loaded
```

Download world data from here and save them in a single directory. Create a variable in R to store this directory.

```
dataDir <- "c:/Users/pedro/Dropbox/colrow/"
```

The content of the directory should be as shown below.

```
list.files(dataDir)
```

```
## [1] "COLROW30.aux"      "COLROW30.dbf"      "COLROW30.sbn"
## [4] "COLROW30.sbx"      "COLROW30.shp"      "COLROW30.shx"
## [7] "g2006_2.dbf"       "g2006_2.shp"       "g2006_2.shp.xml"
## [10] "g2006_2.shx"       "SimU_all.dbf"      "SimU_all.prj"
## [13] "SimU_all.sbn"      "SimU_all.sbx"      "SimU_all.shp"
## [16] "SimU_all.shp.xml"  "SimU_all.shx"
```

See the countries available. The code below shows only the first 10.

```
colrow::getCountries(dataDir)[1:10]
```

```
## [1] "Afghanistan"      "Aksai Chin"        "Albania"
## [4] "Algeria"          "American Samoa"    "Andorra"
## [7] "Angola"           "Anguilla"          "Antarctica"
## [10] "Antigua and Barbuda"
```

Then use get functions to generate the data for a given country. It must be written in the same way as one of the outputs of `getCountries()`.

```

country <- "Brazil"

myLU <- colrow::getLU(country, dataDir)

## Reading all countries
## Selecting Brazil
## Reading all SimUs
## Subsetting SimUs
## Mapping SimU to LU
## Merging data
## Computing union of SimUs within the same LU

myCR <- colrow::getCR(country, dataDir)

## Reading all countries
## Selecting Brazil
## Reading all SimUs
## Subsetting SimUs
## Mapping SimU to CR

mySimU <- colrow::getSimU(country, dataDir)

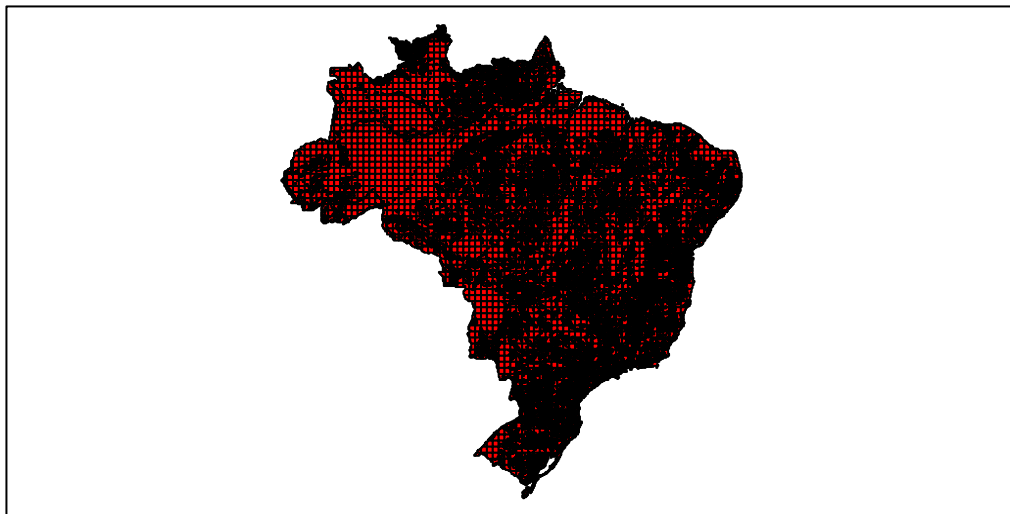
## Reading all countries
## Selecting Brazil
## Reading all SimUs
## Subsetting SimUs

To see that everything is correct, plot the data.

plot(sf::st_geometry(mySimU), main = "SimU", col = "red"); box()

```

## SimU



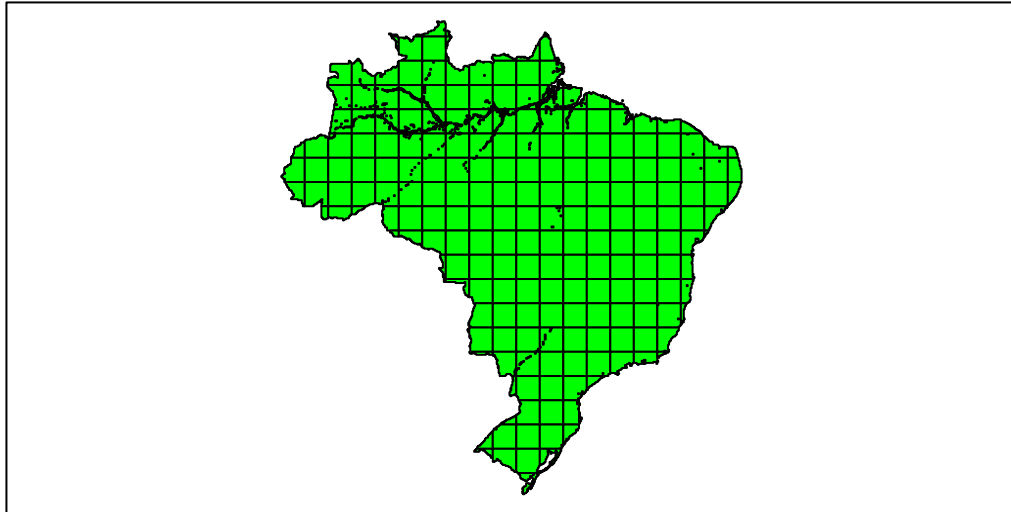
```
plot(sf::st_geometry(myCR), main = "CR", col = "blue"); box()
```

CR



```
plot(sf::st_geometry(myLU), main = "LU", col = "green"); box()
```

LU



Finally, save the data to be used by the processing functions. Each of this files has a column ID that will be used to match the objects from the CSV file.

```
sf::write_sf(myLU, paste0(country, "LU.shp"))
sf::write_sf(myCR, paste0(country, "CR.shp"))
sf::write_sf(mySimU, paste0(country, "SimU.shp"))
```

## Processing outputs

To process a file, it is necessary to choose the shapefile above that matches the representation used in the csv, a csv file and a vector of attribute names. These names must have one attribute called ID and another called VALUE. You can use `colrow::attrs()` to avoid quotes.

```
csvfile <- system.file("extdata/scenarios/FC/Land_Compare3_FC.csv", package = "colrow")
attributes <- colrow::attrs(COUNTRY, ID, ALTI, SLP, SOIL, USE, SCENARIO, YEAR, VALUE)

result <- colrow::processFile(
  "BrazilCR.shp",
  csvfile,
  attributes
)
```

```
## Reading shapefile: BrazilCR.shp
```

```
## Reading data file: C:/Users/pedro/Documents/R/win-library/3.5/colrow/extdata/scenarios/FC/Land_Compare3_FC.csv
```

```
## 6 objects belong to the shapefile but not to the csv file: CR273185, CR273186, CR274185, CR274186, CR275185, CR275186
```

```
## Ignored attributes: COUNTRY, ALTI, SLP, SOIL, SCENARIO
```

```
## Attributes to be joined: USE, YEAR
## USE: CrpLnd, PriFor, NatLnd, ForReg, GrsLnd, MngFor, PltFor
## YEAR: 2000, 2010, 2020, 2030, 2040, 2050
## Spreading the data
## 40 attributes were created
## Merging the data
## Replacing 34340 NA values by zero
```

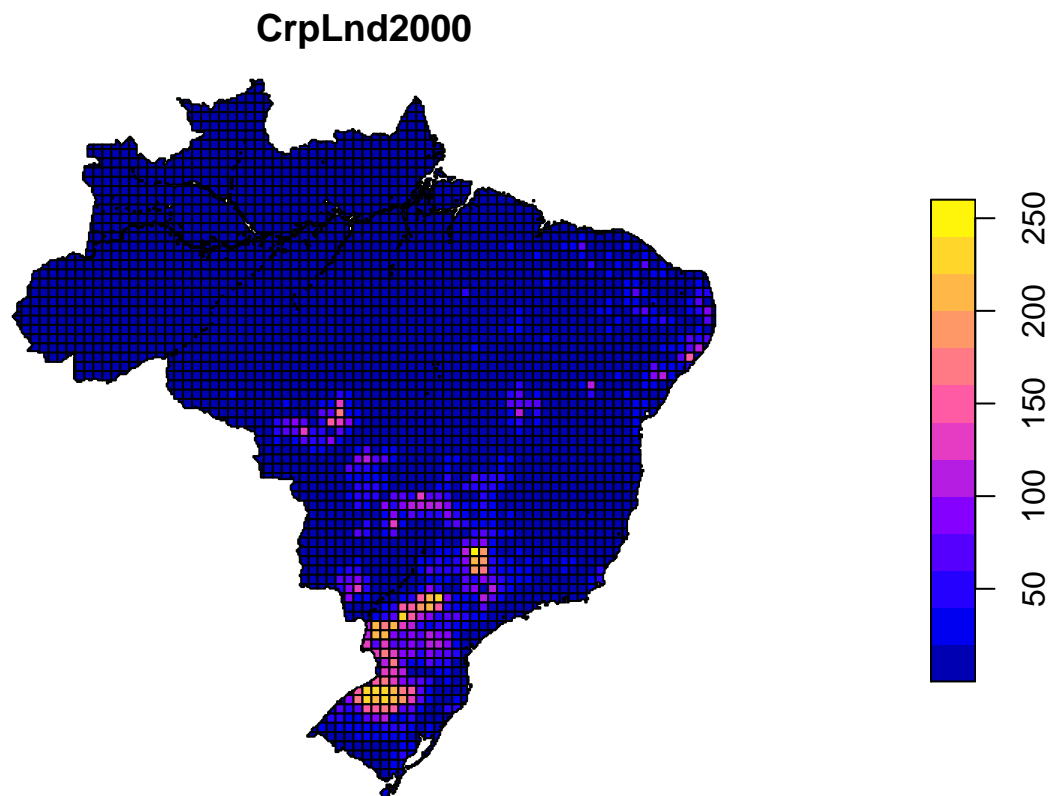
See the attributes created. It concatenates the attribute values to create the attribute names.

```
names(result)
```

```
## [1] "ID"          "CrpLnd2000" "CrpLnd2010" "CrpLnd2020" "CrpLnd2030"
## [6] "CrpLnd2040" "CrpLnd2050" "ForReg2030"  "ForReg2040"  "ForReg2050"
## [11] "GrsLnd2000"  "GrsLnd2010"  "GrsLnd2020"  "GrsLnd2030"  "GrsLnd2040"
## [16] "GrsLnd2050"  "MngFor2000"  "MngFor2010"  "MngFor2020"  "MngFor2030"
## [21] "MngFor2040"  "MngFor2050"  "NatLnd2000"  "NatLnd2010"  "NatLnd2020"
## [26] "NatLnd2030"  "NatLnd2040"  "NatLnd2050"  "PltFor2000"  "PltFor2010"
## [31] "PltFor2020"  "PltFor2030"  "PltFor2040"  "PltFor2050"  "PriFor2000"
## [36] "PriFor2010"  "PriFor2020"  "PriFor2030"  "PriFor2040"  "PriFor2050"
## [41] "geometry"
```

Basic plot.

```
plot(result["CrpLnd2000"])
```



It is also possible to save the output when processing a file. In this case, the function does not return anything.

```
colrow::processFile(
  "BrazilCR.shp",
  csvfile,
  attributes,
  "brazilOutput.shp"
)
```

```
## Reading shapefile: BrazilCR.shp
## Reading data file: C:/Users/pedro/Documents/R/win-library/3.5/colrow/extdata/scenarios/FC/Land_Compa
## 6 objects belong to the shapefile but not to the csv file: CR273185, CR273186, CR274185, CR274186, C
## Ignored attributes: COUNTRY, ALTI, SLP, SOIL, SCENARIO
## Attributes to be joined: USE, YEAR
## USE: CrpLnd, PriFor, NatLnd, ForReg, GrsLnd, MngFor, PltFor
## YEAR: 2000, 2010, 2020, 2030, 2040, 2050
## Spreading the data
## 40 attributes were created
## Merging the data
## Replacing 34340 NA values by zero
## Renaming attributes according to convertList
## No attribute has more than 10 characters
## Writing output file: brazilOutput.shp
```

Note that shapefiles have limits: 255 or less columns, 10 or less characters in the attribute names. When working with large attribute names, it is necessary to simplify the names.

```
convert <- list(
  CrpLnd = "cr", PriFor = "pr",
  NatLnd = "nl", ForReg = "fr",
  GrsLnd = "gl", MngFor = "mf",
  PltFor = "pl"
)
```

If you want to simplify the years to two characters, just run:

```
for(year in paste(seq(2000, 2050, 10))) # from 2000, 2010, ..., 2050
  convert[[year]] = substr(year, 3, 4) # to 00, 10, ..., 50

unlist(convert)
```

```
## CrpLnd PriFor NatLnd ForReg GrsLnd MngFor PltFor 2000 2010 2020
## "cr" "pr" "nl" "fr" "gl" "mf" "pl" "00" "10" "20"
## 2030 2040 2050
## "30" "40" "50"
```

```
result <- colrow::processFile(
  "BrazilCR.shp",
  csvfile,
  attributes,
  "brazilOutput.shp",
  convert
)
```

```
## Reading shapefile: BrazilCR.shp
## Reading data file: C:/Users/pedro/Documents/R/win-library/3.5/colrow/extdata/scenarios/FC/Land_Compa
## 6 objects belong to the shapefile but not to the csv file: CR273185, CR273186, CR274185, CR274186, C
```

```
## Ignored attributes: COUNTRY, ALTI, SLP, SOIL, SCENARIO
## Attributes to be joined: USE, YEAR
## USE: CrpLnd, PriFor, NatLnd, ForReg, GrsLnd, MngFor, PltFor
## YEAR: 2000, 2010, 2020, 2030, 2040, 2050
## Spreading the data
## 40 attributes were created
## Merging the data
## Replacing 34340 NA values by zero
## Renaming attributes according to convertList
## No attribute has more than 10 characters
## Writing output file: brazilOutput.shp
```

## Plotting outputs

More plotting examples.

## Final remarks

If you have suggestions or want to report an error, please visit the [GitHub page](#) here.