Geoscripting course introdduction

Dainius Masiliūnas and Jan Verbesselt

03 January, 2021

Did you know?!

Scripting can be used to process satellite data easily to monitor deforestation globally

- Google Earth Engine for global deforestation monitoring - https://earthengine.google.org/#intro

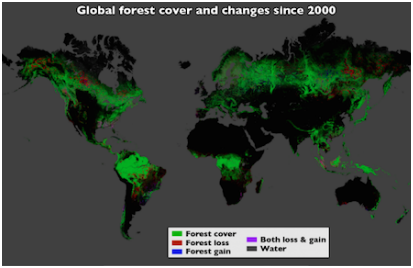
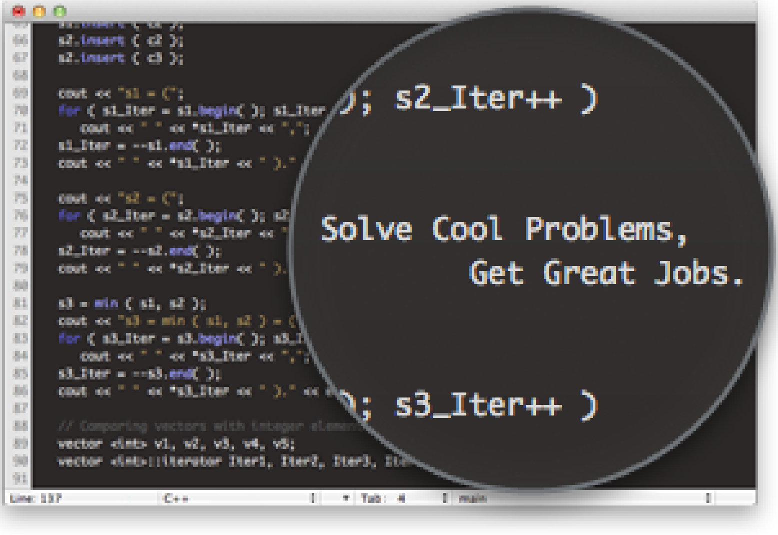


Image Credit: NASA Goddard, based on data from Hansen et al., 2013.

Why geoscripting?



```
s2.insert ( c2 );
s2.insert ( c3 );

cout << "s1 = (";
for ( s1_Iter = s1.begin( ); s1_Iter
   cout << " " << *s1_Iter << ",";
s1_Iter = --s1.end( );
cout << " " << *s1_Iter << " )."

cout << "s2 = (";
for ( s2_Iter = s2.begin( ); s2
   cout << " " << *s2_Iter << "
s2_Iter = --s2.end( );
cout << " " << *s2_Iter << " ).

s3 = min ( s1, s2 );
cout << "s3 = min ( s1, s2 ) = (
for ( s3_Iter = s3.begin( ); s3_I
   cout << " " << *s3_Iter << ",";
s3_Iter = --s3.end( );
cout << " " << *s3_Iter << " )." << e

// Comparing vectors with integer elemen
vector <int> v1, v2, v3, v4, v5;
vector <int>::iterator Iter1, Iter2, Iter3, It
```

```
); s2_Iter++ )

Solve Cool Problems,
        Get Great Jobs.

); s3_Iter++ )
```

Why geoscripting?

- Reproducible: you keep track of what you have done
- Efficient: you can write a script to do something and repeat your analysis x1000...
  - Good for finding errors

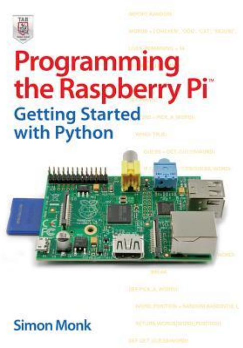
Can you think of other advantages?

Why geoscripting?

- Enable collaboration
  - a huge social R and Python community

Why geoscripting?

- Build your own tools and functions (e.g. automated web-processing, etc.)



See demos!

The Geoscripting course

Goal:

- Scripting to solve spatial challenges!
- **Learn by doing**
- **Learn how to keep on learning**

A team of `R` & `Python` experts who use geo-scripting languages every day helped to develop this course

Who are we?

- Course coordinator: Dainius Masiliūnas, (Jan Verbesselt)
- Lecturers: Corné Vreugdenhil (Python week) and Andrei Mîrț (Scoold & GEE)
- Teaching assistants: Matthew Needle, Kamiel Verhelst, Thanasis Antonopoulos
- Guest lecturers: Arend Ligtenberg, Nandika Tsendbazar, Judith Verstegen, Sabina Roșca, Benjamin Brede, Robert Masolele, Wanda de Keersmaecker, Milutin Milenković

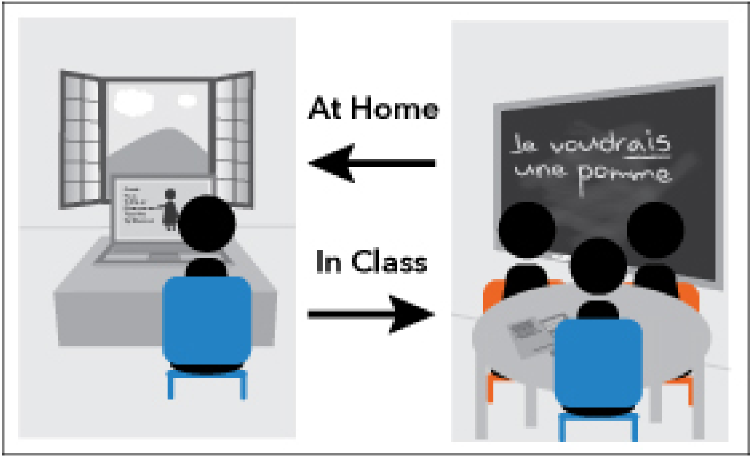
Geoscripting learning objectives

. . . are changing! In your case, by the end of the course, you should be able to:

- Demonstrate scripting skills by writing clear, documented and reproducible scripts to read, write and visualise spatial data
- Use documentation, search tools and contribute to discussion platforms with reproducible examples as an aid to solve scripting challenges effectively
- Combine functions from novel libraries to efficiently process spatial data
- Create an integrated software project that solves a spatial challenge

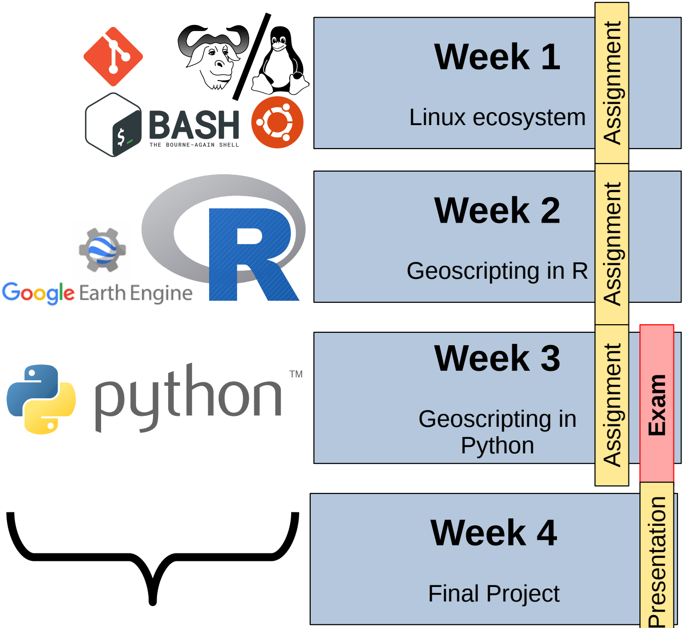
Course concept

Flipped Classroom concept:



- Experts from companies, research institutes and university will present (guest lectures)

How will we learn all that?



Week 1

Linux ecosystem

Assignment

Week 2

Geoscripting in R

Assignment

Week 3

Geoscripting in Python

Assignment

Exam

Week 4

Final Project

Presentation

What we will do every day

- 08:30 - 11:00: Self-study (individually)
- 11:00 - 12:00: Work on exercise of the day (in pairs)
- 12:00 - 13:00: Lunch break
- 13:00 - 15:00: Work on exercise of the day (in pairs)
- 15:00 - 15:30: Review exercise of two other teams
- 15:30 - 16:30: Discussion session
- 16:30 - 17:00: Guest lecture

Exceptions: Assignment days (Thursdays), the exam day and the project week: whole day work!

Deliverables

- Daily exercises: in pairs, not graded, but peer reviewed; assignments and exam are very similar, so good practice!
- Weekly assignments: in pairs, graded (**20%**) and peer reviewed; peer review is graded too!
- Exam: individual, graded (**30%**), **not** peer reviewed but receive feedback from autotests, all practice.
- Final project: in pairs, graded (**50%**) and peer reviewed.
  - Think of a geoscripting topic feasible within 3.5 days with 2 people

Starter material will be made available for each group/person on GitHub: https://github.com/geoscripting-2021. Every change in the starter is automatically submitted to Brightspace (CodeGrade), rubrics and automatic feedback is available there.

Team work & helping each other

- Fill out your GitHub username by filling in a form
- Teams of two:
  - Create or join a group on Brightspace > Exercises > Exercise 1
  - For project work there is a possibility to change your groups

Heeelp!

Confused?! Always start in Microsoft Teams!

- Announcements in the General channel!
- Each pair gets a channel! So you (and we) can talk to each other directly!
- Tabs at the top link to all content!
- Support in the Support channel! Simply join one (or more) of the ongoing meetings and raise your hand!
- Offline support on the Scoold Q&A platform!

What we will do today

- 08:30 - 9:15: Introduction presentation
- 09:30 - 10:00: Introduction to version control
- 10:00 - 12:00: Self-study
- 12:00 - 13:00: Lunch break
- 13:00 - **15:00**: Think about a topic for your project and submit your thoughts
- 15:00 - **15:30**: Review the idea of two other teams
- 15:30 - 16:00: Discussion session

Deadlines are always **hard**!

Peer review

- When you do peer review, use the peer review team selector!
- You get a link to the repository of the teams you are supposed to review.
- Go to “Issues” and open an issue; enter your team name in the title.
- Review according to the rubric on Brightspace (CodeGrade) and the exercise/assignment description.

Learning outcomes of today:

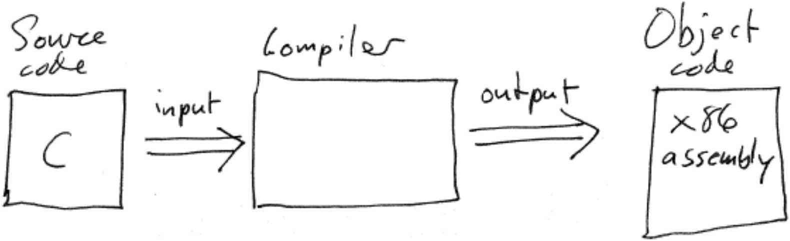
- Getting started with Linux, Git
- Submit an exercise to CodeGrade!
- Think of the project topic

What is a scripting language?

- A scripting language can interpret and automate the execution of tasks, instead of clicking!!!
- A scripting language is the glue, between different commands, functions, and objectives without the need to compile it for each OS (e.g. windows)
- This is different from like C/C++/Fortran
  - these are languages that need to be compiled first.

Compiling

Passing the source code through a compiler, which translates the `high-level language` instructions into an `exe`.



The main scripting languages for GIS and Remote sensing currently are:

- R
- Python
- GRASS (grass function are included in QGIS)
- JavaScript for geoweb scripting
- Matlab
- IDL (ENVI)
- ...

Python versus R

- Python is a general purpose programming language
- R is particularly strong in statistical computing and graphics
- Syntactic differences between R and Python can be confusing
- There are many R and Python packages for spatial analyses

R packages

data
analysis
models
functions
regression
package
estimation
using
tools
model
bayesian
methods
multivariate
linear
interface
time
series
based
test
statistical
generalized
clustering
distribution
distributions
algorithm
tests
method
selection
random
statistics
spatial
multiple
network
modelling
classification
survival
mixed
nonparametric
testing
dynamic
inference
mixture
fitting
simulation
modeling
robust
likelihood
sparse
design
hierarchical
graphical
datasets
sampling
correlation
visualization
implementation
graphics
networks
sets
variable
matrix
plots
files
api
density
prediction
gaussian
learning
genetic
utilities
binary
function
analyses
effects

Many programs have support for

Python & R e.g.:

- QGIS
- GRASS
- PostGIS

Time for a break!

After that:

- Git presentation
- Try out Linux
- Git self-study
- Exercise 1