Geoscripting course introdduction

Dainius Masiliūnas and Jan Verbesselt

02 sausio, 2022

What comes to mind when you think of Geoscripting?

Answer on Wooclap in the link in the chat!

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

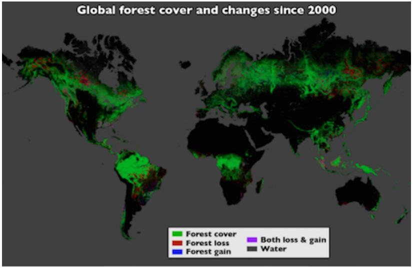
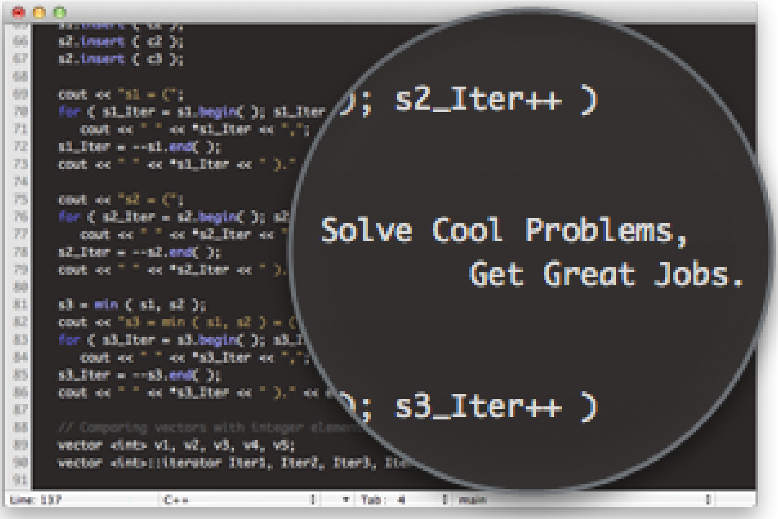


Image Credit: NASA Goddard, based on data from Hansen et al., 2013.

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

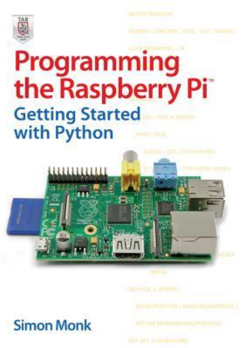
Can you think of other advantages?

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!

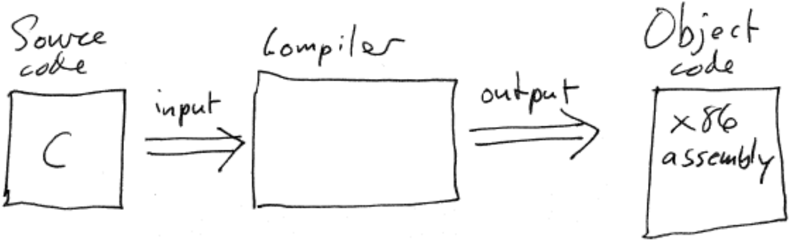
What is a scripting (interpreted) language?

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 a compiled language like C/C++/Fortran
  - these are languages that need to compile the whole program/app 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
model
bayesian
using
tools
methods
multivariate
interface
linear
statistical
generalized
based
test
time
series
inference
selection
random
statistics
clustering
distribution
algorithm
tests
method
distributions
spatial
multiple
network
modelling
classification
mixed
survival
nonparametric
testing
dynamic
networks
mixture
simulation
modeling
graphical
sampling
correlation
likelihood
sparse
design
hierarchical
visualization
implementation
graphics
experiments
datasets
variable
matrix
sets
plots
fitting
api
robust
genetic
gaussian
learning
density
prediction
utilities
binary
function

Many programs have support for

Python & R e.g.:

- QGIS
- GRASS
- PostGIS

. . . Cliffhanger!

How will you learn about all that in the course? Find out after this 5-minute break!

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?

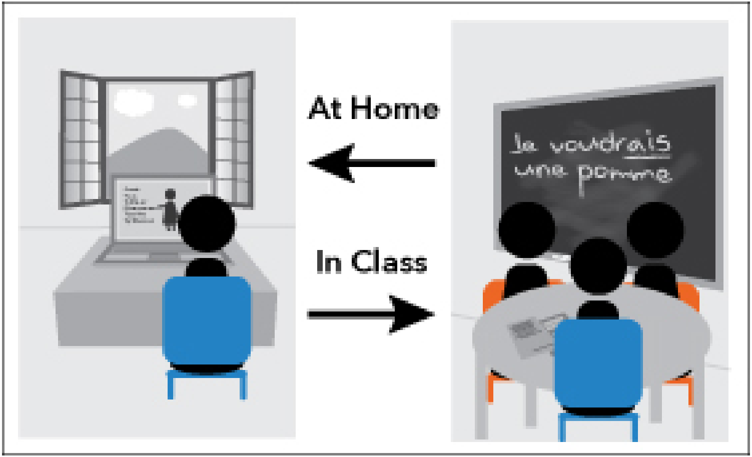
- Course coordinators: Dainius Masiliūnas, Jan Verbesselt
- Lecturers: Judith Verstegen (Python week) and Andrei Mîrț (Scoold & GEE)
- Teaching assistants: Yneke van Iersel, Victor Bonekamp, Daan Kivits
- Guest lecturers: Robert Masolele, Nandika Tsendbazar, Benjamin Brede

Geoscripting learning objectives

- Demonstrate scripting skills by writing clear, documented, and reproducible scripts to read, write and visualise spatial data
- Apply version control supporting effective collaboration in a team
- Use documentation, search tools, and contribute to discussion platforms with reproducible examples as an aid to solve scripting challenges effectively
- Be willing to use and create open-source software with an appropriate software license
- Combine functions from novel libraries to efficiently process spatial data
- Create an integrated software project that solves a spatial challenge

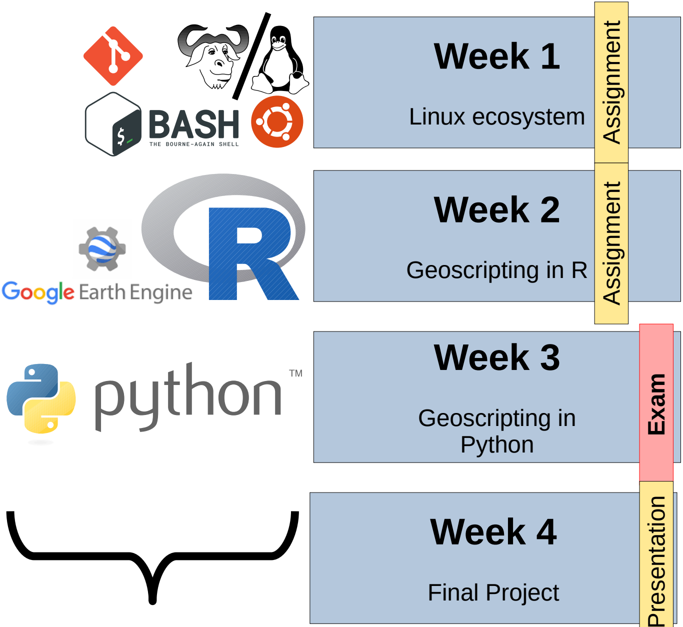
Course concept

Flipped Classroom concept:



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

How will we learn all that?



Week 1

Linux ecosystem

Assignment

Week 2

Geoscripting in R

Assignment

Week 3

Geoscripting in Python

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 - 16:00: Discussion session
- 16:00 - 17:00: Guest presentation & wrap-up

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 assignment, graded (**30%**), **not** peer reviewed but receive feedback from autotests, mostly practice.
- Final project: in pairs, graded (**50%**) and peer reviewed.
  - Think of a geoscripting topic feasible within 3.5 days with 4 people

Starter material will be made available for each group/person on GitLab: https://git.wur.nl/geoscripting-2022. Every change in the starter is automatically submitted to Brightspace (CodeGrade), rubrics and automatic feedback is available there.

Team work & helping each other

- Log into GitLab and set up 2-factor authentication: https://git.wur.nl/-/profile/account
- Set up teams of two:
  - Create or join a group on Brightspace > Exercises > Exercise 1
  - For project work, you will work in groups of 4, which we will set based on your chosen topics

Heeelp!

Confused?! Always start in Microsoft Teams!

- Announcements in the General channel!
- If you like, you can create a new channel for your team's internal communication
- Tabs at the top link to all content!
- Support in the Questions and Answers 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 - 10:00: Introduction presentation
- 10:00 - 12:00: Self-study (geoscripting-wur.github.io)
- 12:00 - 13:00: Lunch break
- 13:00 - **15:00**: Use git to submit your thoughts on what you would like to work on during the course
- 15:00 - **15:30**: Review the idea of two other teams
- 15:30 - 16:00: Discussion session
- 16:00 - 17:00: Interactive Git presentation

Deadlines are always **hard**!

Peer review

- When you do peer review, check the exercise of the day on Brightspace.
- You get access to the solution of the teams you are supposed to review.
- Review according to the rubric on Brightspace (CodeGrade) and the exercise/assignment description.
  - You can use in-line comments and general comments.
  - Don't review the review.
  - In upcoming exercises, reproducibility is a prerequisite!

Summary

- Geoscripting is an essential skill in research, as well as highly sought-after in the industry
- Scripting saves you time in the long run
- Geoscripting is an intensive, whole-day course where you learn best by doing
- Deliverables are all practice, little theory!
- Mind the deadlines, and don't forget to review others!

Learning outcomes of the rest of today:

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

Questions?