Modelling Journey

BIO401-01/598-02

2021-03-17 Wed

Course Progression

GeoComputation

- GIS concepts

Course Progression

GeoComputation

- GIS concepts
- Linux environment

Course Progression

GeoComputation

- GIS concepts
- Linux environment
- Geo Objects : raster + vector files

Course Progression

GeoComputation

- GIS concepts
- Linux environment
- Geo Objects : raster + vector files
- Manipulation tools : gdal/ogr, pktools

Course Progression

GeoComputation

- GIS concepts
- Linux environment
- Geo Objects : raster + vector files
- Manipulation tools : gdal/ogr, pktools

GeoModelling

- GeoMath

Course Progression

GeoComputation

- GIS concepts
- Linux environment
- Geo Objects : raster + vector files
- Manipulation tools : gdal/ogr, pktools

GeoModelling

- GeoMath
- GeoStats

Course Progression

GeoComputation

- GIS concepts
- Linux environment
- Geo Objects : raster + vector files
- Manipulation tools : gdal/ogr, pktools

GeoModelling

- GeoMath
- GeoStats
- GeoCoding

Modelling Process

- Problem Statement : project definition, data collection

Modelling Process

- Problem Statement : project definition, data collection
- Model Construction

Modelling Process

- Problem Statement : project definition, data collection
- Model Construction
  1. Explorative
     Data exploration : missing data, correction, manipulation
     Geocomput tools, math/stats, programming

Modelling Process

- Problem Statement : project definition, data collection
- Model Construction
  1. Explorative
     Data exploration : missing data, correction, manipulation
     Geocomput tools, math/stats, programming
  2. Mathematical
     Modelling knowledge, Programming skills

Modelling Process

- Problem Statement : project definition, data collection
- Model Construction
  1. Explorative
     Data exploration : missing data, correction, manipulation
     Geocomput tools, math/stats, programming
  2. Mathematical
     Modelling knowledge, Programming skills
- Model Analysis
  solution determination

Modelling Process

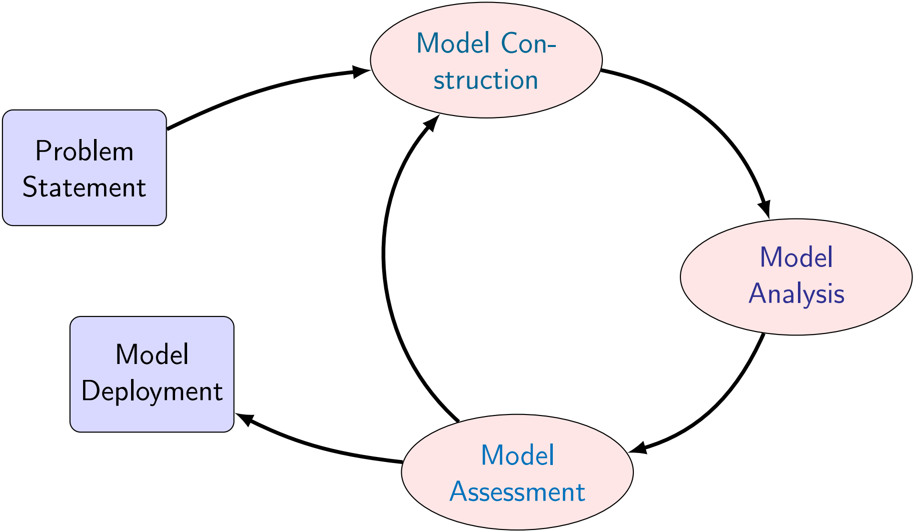
- Problem Statement : project definition, data collection
- Model Construction
  1. Explorative
     Data exploration : missing data, correction, manipulation
     Geocomput tools, math/stats, programming
  2. Mathematical
     Modelling knowledge, Programming skills
- Model Analysis
  solution determination
- Model Assessment
  fidelity, cost, complexity, flexibility, etc.

Modelling Process

- Problem Statement : project definition, data collection
- Model Construction
  1. Explorative
     Data exploration : missing data, correction, manipulation
     Geocomput tools, math/stats, programming
  2. Mathematical
     Modelling knowledge, Programming skills
- Model Analysis
  solution determination
- Model Assessment
  fidelity, cost, complexity, flexibility, etc.
- Model Deployment
  presentation : map output

Modelling Process

- Iterative Process



Model Construction

Problem Statement

Model Analysis

Model Deployment

Model Assessment

Camping Analogy

Basically camping tool set
Barbeque grill
Skewers
Charcoal
Tent
Tongs for charcoal
Tongs
Bamboo for blower
Lighter
Gloves
Ignition chemical
Mat
Blue Tarp
Sleeping Bag
FUEL
4ℓ
Fuel
Kitchen knife set
Lantern
Cooler (48L)
Ice pack
Pack-Away Picnic Table Set

http://survival-mastery.com

Class Structure

- previous class review + Q&A
- course materials for the day
- homework deployment

Homework

Policy

- issued on Mon and/or Wed each week
- due by class hours on Mon of the *immediate* following week
- collaboration allowed except for the bonus question

Homework

Policy

- issued on Mon and/or Wed each week
- due by class hours on Mon of the *immediate* following week
- collaboration allowed except for the bonus question

Typical problem set

- 3 basic questions (80%)
- 1 advanced question (20%)
- 1 bonus question (10%)

Homework

Policy

- issued on Mon and/or Wed each week
- due by class hours on Mon of the *immediate* following week
- collaboration allowed except for the bonus question

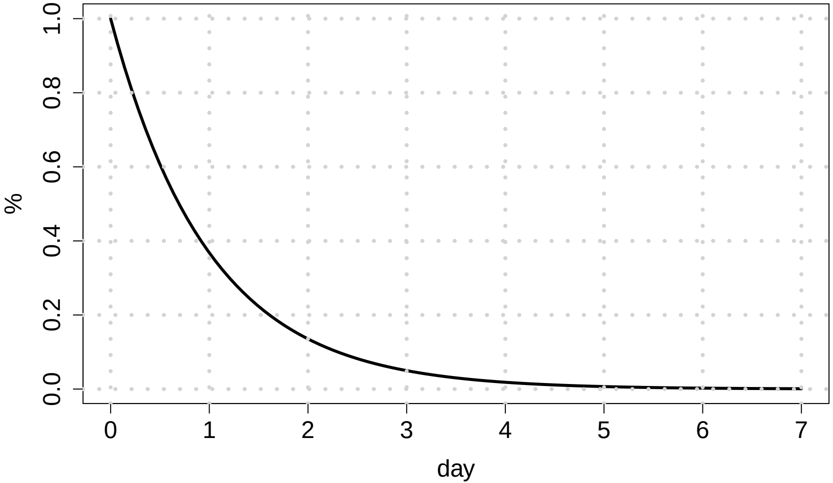
Typical problem set

- 3 basic questions (80%)
- 1 advanced question (20%)
- 1 bonus question (10%)

Feedback

- graded HW returned *one week after* the due date
- answer sheets provided *one week after* the HW due date
- HW systematically reviewed in class *one week after* the due date

Homework : Turn-in time fitting



Communications

Office Hours

- Time : 4 pm on every Tue and Thur (EDT)
- Location : *check the slack channel*

Free style

- anytime, slack channel

Git

- created by *Linus Torvalds* in 2005

Git

- created by *Linus Torvalds* in 2005
- distributed version control : each directory as a full-fledged repo

Git

- created by *Linus Torvalds* in 2005
- distributed version control : each directory as a full-fledged repo
- used for changes tracking and work coordination among collaborators

Git

- created by *Linus Torvalds* in 2005
- distributed version control : each directory as a full-fledged repo
- used for changes tracking and work coordination among collaborators

Example

```
$ cd ~/SE_data
$ ls -a
.   ..   exercise   .git   lectures   README.md
```

Git

- created by *Linus Torvalds* in 2005
- distributed version control : each directory as a full-fledged repo
- used for changes tracking and work coordination among collaborators

Example

```
$ cd ~/SE_data
$ ls -a
.  ..  exercise  .git  lectures  README.md
```

Basic Practice

- only the first time

```
$ mv ~/SE_data /media/sf_LVM_Shared/my_SE_data
# (working copy for yourself, taking notes, etc.)

$ git clone https://github.com/selvaje/SE_data
# (source copy, no work inside here)
```

Git

Basic Practice

- routine after the first time

```
$ cd ~/SE_data
$ git pull # (sync. w/ server)

$ rsync -hvrPt --ignore-existing  ~/SE_data/* \
  /media/sf_LVM_Shared/my_SE_data
#(sync. only new files)

$ cd /media/sf_LVM_Shared/my_SE_data # (work here)
```

- Common practice to separate source and working copies
- Important : NOT working in the source copy

Git (optional)

- git repo setup
- good for professional development
- good for collaboration

Git (optional)

- git repo setup
- good for professional development
- good for collaboration

Initialisation

```
$ mkdir my_Project ; cd my_Project

$ git config --global user.name "your name"
$ git config --global user.email "your email"

$ git init
Initialized empty Git repository in ...

$ ls -a
.   ..   .git
```

Git (optional)

Add files

```
$ touch README.md
$ git status
Untracked files:
(use "git add <file>..." to include in what will be
committed)
README.md

$ git add README.md ; git status
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file:   README.md

$ git commit -m "added README" ; git status
nothing to commit, working tree clean
```

Git (optional)

Modify file contents

```
$ echo -e "Project for BIO401-01/598-02\n" > README.md
$ git status
(use "git add <file>..." to update what will be committed)
modified:   README.md

$ git add README.md ; git commit -m "modified README"
[master 002362a] modified README
1 file changed, 2 insertions(+)
$ git status
nothing to commit, working tree clean
```

Git (optional)

Modify file contents

```
$ echo -e "Project for BIO401-01/598-02\n" > README.md
$ git status
(use "git add <file>..." to update what will be committed)
modified:   README.md

$ git add README.md ; git commit -m "modified README"
[master 002362a] modified README
1 file changed, 2 insertions(+)
$ git status
nothing to commit, working tree clean
```

Move or remove files

```
$ git mv <old file> <new file>
$ git rm <filename>
remember to commit after mv or rm actions
```

Git (optional)

Link repo to GitHub

```
create a GitHub account
create a repo on GitHub
follow the instructions on the GitHub setup page

$ git remote add origin git@github.com:/your/project
$ git push -u origin master
```

Git (optional)

Link repo to GitHub

```
create a GitHub account
create a repo on GitHub
follow the instructions on the GitHub setup page

$ git remote add origin git@github.com:/your/project
$ git push -u origin master
```

Sync. w/ GitHub

```
$ git pull # download
$ git push # upload
```

Git (optional)

Link repo to GitHub

```
create a GitHub account
create a repo on GitHub
follow the instructions on the GitHub setup page

$ git remote add origin git@github.com:/your/project
$ git push -u origin master
```

Sync. w/ GitHub

```
$ git pull # download
$ git push # upload
```

ref : Git version control training

Programming Language, R

Language of the choice

- Class survey
- Popularity, Community support
- Appropriateness

Programming Language, R

Language of the choice

- Class survey
- Popularity, Community support
- Appropriateness

Reference books (optional)

- The Art of R Programming: A Tour of Statistical Software Design, Norman Matloff (2011)
- ggplot2: Elegant Graphics for Data Analysis, Hadley Wickham (2016)
- Spatial Ecology and Conservation Modeling Applications with R Robert Fletcher, Marie-Josée Fortin (2018)
- Spatial Data Science with R, Robert J. Hijmans
- Geocomputation with R, Robin Lovelace, Jakub Nowosad, Jannes Muenchow (2021)