N.B.: the following program is a preliminary curriculum. The final program will be available thorugh the official web-page at the "Programme" section.

Open Science

Aims:

To consider the importance of Open Science.

Learning Outcomes:

At the end of this course a student will

- have reflected on the tools that enable Open Science,
- have reflected on the impact Open Science on their own research and future career.

Course content:

- The goals of Open Science
    - reproducibility,
    - “nullius in verba”,
    - speeding up research.
- Open sessions to reflect on personal impact of Open Science.

Open Research Data

Aims:

To have an understanding of the principles of research data management and the impact of Openness and Sharing in Research

Learning Outcomes:

At the end of this course a student will

- understand the data curation life-cycle,
- understand the principles and importance of annotation,
- how to publish data.

Course content:

- Incentives for curation.
- The data curation life-cycle.
- Methods for data publishing (such as Zenodo). Examples from different communities.
- Metadata standards.

Data Carpentry

Aims:

To have an introductory understanding of how to manipulate data as stored in SQL databases.

Learning Outcomes:

At the end of this course a student will

- understand what an SQL database is,
- be able to perform basic queries of an SQL databases,
- perform aggregation commands such as GROUP BY,
- combine data from tables using commands such as JOIN.

Course Content:

- Introduction to SQLite Manager, SQL and relational databases.
- Motivation of the use of relational databases.
- Import files to SQL.
- SELECTing data from SQL.
- Filtering data from SQL.
- COUNT, GROUP BY commands.
- JOINS, ALIASES.

Software Carpentry

Aims:

To have an introductory understanding of programming and software engineering skills to manipulate data and analyse data in reproducible fashion.

Learning Outcomes:

At the end of this course a student will

- have an introductory understanding of the Unix shell,
- be able to execute simple commands in R,
- be able to use Git.

Course content:

- Introduction to the Unix shell.
- File concepts in Unix.
- Combining Unix commands, pipes and filters.
- Shell scripts.
- Functions in R.
- Conditionals in R.
- Command line R programs.
- Best practices in R.
- Building your own R packages.
- Setting up Git.
- Creating a Git repository.
- Tracking changes in Git.
- Collaboration and Open Science with Git.

Analysis

Aims:

To have an understanding of the principles necessary to analyse data in terms of determining significance, performing regression and clustering.

Learning Outcomes:

At the end of this course a student will

- understand how to look for significance beyond applying basic significance tests,
- understand principles of Machine Learning,
- apply techniques such as clustering on multi-dimensional data.

Course content:

- Modelling data and significance,
- Machine Learning - supervised and unsupervised learning,
- Machine Learning - cross-validation,
- Clustering data.

Visualisation

Aims:

To have an understanding of the principles of visualising data.

Learning Outcomes:

At the end of this course a student will

- understand how to use R to perform visualisation,
- be able to perform a critical assessment of effective visualisation techniques.

Course content:

- Data wrangling.
- Visualisation packages in R (such as ggplot2).
- [Optional] Visualisation in Python.
- Workshop based approaches to critical assessment of visualisation.

Computational Infrastructures

Aims:

To have an introductory understanding of cloud computing platforms and their use.

Learning Outcomes:

At the end of this course a student will

- understand the basic concepts of cloud computing,
- be able to launch a Virtual Machine on a cloud platform,
- be able use more advanced features such as batch schedulers or containers.

Course content:

- Introduction to cloud computing concepts such as IaaS and PaaS.
- Secure authentication
- Launching a VM on an IaaS cloud.
- Deploying scripts.
- Interacting with mass storage.
- Use of batch schedulers of containers.