Carpentries Training Options (English)

This is a list and description of the potential Data and Software training available for the DataUp participants to select.

For more information visit The Carpentries website.

Core Lessons	Brief Description	Lessons	Speaking Language
	Data Lessons		
Data in Ecology	This workshop uses a **tabular ecology dataset** from the **Portal Project Teaching Database** and teaches data cleaning, management, analysis, and visualization. The Ecology workshop can be taught using R or Python as the base language.	1. Ecology Workshop Overview; 2. Data Organization in Spreadsheets for Ecologists; 3. Data Cleaning with OpenRefine for Ecologists; 4. Data Management with SQL for Ecologists; 5. Data Analysis and Visualization in R for Ecologists; 6. Data Analysis and Visualization in Python for Ecologists	English
Data in Genomics	The focus of this workshop is on working with genomics data, and data management and analysis for genomics research, including best practices for organization of bioinformatics projects and data, use of command line utilities, use of command line tools to analyze sequence quality and perform variant calling, and connecting to and using cloud computing.	1. Genomics Workshop Overview; 2. Project Organization and Management for Genomics; 3. Introduction to the Command Line for Genomics; 4. Data Wrangling and Processing for Genomics; 5. Introduction to Cloud Computing for Genomics	English

Social Science	This workshop uses a **tabular interview dataset** from the **SAFI Teaching Database** and teaches data cleaning, management, analysis and visualization. *There are no pre-requisites, and the materials assume no prior knowledge about the tools.* We use a single dataset throughout the workshop to model the data management and analysis workflow that a researcher would use. The Social Sciences workshop can be taught using R as the base language.	1. Social Science Workshop Overview; 2. Data Organization in Spreadsheets for Social Scientists; 3. Data Cleaning with OpenRefine for Social Scientists; 4. Data Analysis and Visualization with R for Social Scientists	English
Geospatial Data	This workshop is co-developed with the National Ecological Observatory Network **(NEON)**. It focuses on working with geospatial data - managing and understanding spatial data formats, understanding coordinate reference systems, and working with raster and vector data in R for analysis and visualization.	1. Geospatial Workshop Overview; 2. Introduction to Geospatial Concepts; 3. Introduction to R for Geospatial Data; 4. Introduction to Geospatial Raster and Vector Data with R	English
Core Lessons	**Brief Description**	**Schedule Topics**	**Speaking Language**
	Software Lessons		
The Unix Shell	It's a power tool that allows people to do complex things with just a few keystrokes. More importantly, it helps them combine existing programs in new ways and automate repetitive tasks so they aren't typing the same things over and over again. Use of the shell is fundamental to using a wide range of other powerful tools and computing resources (including "high-performance computing" supercomputers). These lessons will start you on a path towards using these resources effectively.	1. Intro to Shell; 2. Navigating Files and Directories; 3. Working With Files and Directories; 4. Pipes and Filters; 5. Loops; 6. Shell Scripts; 7. Finding Things;	English
Version Control with Git	Version control is the lab notebook of the digital world: it's what professionals use to keep track of what they've done and to collaborate with other people. Every large software	1. Automated Version Control; 2. Setting Up Git; 3. Creating a Repository;	English

	development project relies on it, and most programmers use it for their small jobs as well. And it isn't just for software: books, papers, small data sets, and anything that changes over time or needs to be shared can and should be stored in a version control system	4. Tracking Changes; 5. Exploring History; 6. Ignoring Things; 7. Remotes in GitHub; 8. Collaborating; 9. Conflicts; 10. Open Science; 11. Licensing; 12. Citation; 13. Hosting; 14. Supplemental: Using Git from RStudio	
Programming with Python	This introduction to Python is built around a common scientific task: data analysis.	1. Python Fundamentals; 2. Analyzing Patient Data; 3. Repeating Actions with Loops; 4. Storing Multiple Values in Lists; 5. Analyzing Data from Multiple Files; 6. Making Choices; 7. Creating Functions; 8. Errors and Exceptions; 9. Defensive Programming; 10. Debugging; 11. Command-Line Programs	English
Programming with R	The best way to learn how to program is to do something useful, so this introduction to R is built around a common scientific task: data analysis. Our real goal isn't to teach you R, but to teach you the basic concepts that all programming depends on.	1. Analyzing Patient Data; 2. Creating Functions; 3. Analyzing Multiple Data Sets; 4. Making Choices; 5. Command-Line Programs; 6. Best Practices for Writing R Code; 7. Dynamic Reports with knitr; 8. Making Packages in R; 9. Introduction to RStudio; 10. Addressing Data; 11. Reading and Writing CSV Files; 12. Understanding Factors; 13. Data Types and Structures;	English

		14. The Call Stack 15. Loops in R	
Using Databases and SQL	Text files are easiest to create, and work well with version control, but then we would have to build search and analysis tools ourselves. Spreadsheets are good for doing simple analyses, but they don't handle large or complex data sets well. Databases, however, include powerful tools for search and analysis, and can handle large, complex data sets. These lessons will show how to use a database to explore the expeditions' data.	1. Selecting Data; 2. Sorting and Removing Duplicates; 3. Filtering; 4. Calculating New Values; 5. Missing Data; 6. Aggregation; 7. Combining Data; 8. Data Hygiene; 9. Creating and Modifying Data; 10. Programming with Databases - Python; 11. Programming with Databases - R	English
R for Reproducible Scientific Analysis	The goal of this lesson is to teach novice programmers to write modular code and best practices for using R for data analysis. R is commonly used in many scientific disciplines for statistical analysis and its array of third-party packages. We find that many scientists who come to Software Carpentry workshops use R and want to learn more. The emphasis of these materials is to give attendees a strong foundation in the fundamentals of R, and to teach best practices for scientific computing: breaking down analyses into modular units, task automation, and encapsulation	1. Introduction to R and RStudio; 2. Project Management With RStudio; 3. Seeking Help; 4. Data Structures; 5. Exploring Data Frames; 6. Subsetting Data; 7. Control Flow; 8. Creating Publication-Quality Graphics with ggplot2; 9. Vectorization; 10. Functions Explained; 11. Writing Data; 12. Splitting and Combining Data Frames with plyr; 13. Dataframe Manipulation with dplyr; 14. Dataframe Manipulation with tidyr; 15. Producing Reports With knitr; 16. Writing Good Software	English

Programming with MATLAB	Our real goal isn't to teach you MATLAB, but to teach you the basic concepts that all programming depends on.	1. Working With Variables; 2. Arrays; 3. Plotting data; 4. Writing MATLAB Scripts; 5. Repeating With Loops; 6. Making Choices; 7. Creating Functions; 8. Defensive Programming	English
Automation and Make	Make is a tool which can run commands to read files, process these files in some way, and write out the processed files. For example, in software development, Make is used to compile source code into executable programs or libraries, but Make can also be used to run analysis scripts on raw data files to get data files that summarize the raw data; run visualization scripts on data files to produce plots; and to parse and combine text files and plots to create papers.	1. Introduction; 2. Makefiles; 3. Automatic Variables; 4. Dependencies on Data and Code; 5. Pattern Rules; 6. Variables; 7. Functions; 8. Self-Documenting Makefiles;	English