Training plan for 3.1 Experimental Design and Basic Statistics

Prior to the course

Students are required to attend online courses from Harvard University in the following link: http://online-learning.harvard.edu/course/data-analysis-life-sciences-1-statistics-and-r-0?keywords=statistics

with the following lectures:

PH525.1x:	Statistics and R for the Life Sciences
PH525.2x:	Introduction to Linear Models and Matrix Algebra
PH525.3x:	Statistical Inference and Modeling for High-throughput Experiments
PH525.4x:	High-Dimensional Data Analysis
PH525.5x:	Introduction to Bioconductor: annotation and analysis of genomes and genomic assays
PH525.6x:	High-performance computing for reproducible genomics
PH525.7x:	Case studies in functional genomics

Students from non-statistics background are required to take the first lecture only, while those from statistics background are required to take the first three lectures, with an option to take the fourth lecture.

Course plan

The first session will cover basic statistics and the importance of experimental design. The second session will cover 'hands-on' session where students will learn statistics using R, which will be covered in the previous session.

Course syllabus

1. Population vs. sample
2. Types of data
3. Normal distribution
4. One-sample and two-sample tests
5. Intro to linear models and anova (Note: this is simply introducing linear regression and anova models)
6. Importance of randomization

This module aims to give students grounding in basic statistics

Objectives

1. To describe the difference between population and sample
2. To describe different types of data and numerical and visual summaries
3. To introduce normal distribution and its properties

4. To describe t-distribution, one-sample and two-sample t-test
5. To introduce a simple linear regression model (one predictor)
6. To introduce experimental design and the importance of randomisation

Learning outcomes

On completion of this module, participants should be able to:

1. Understand the importance of defining population and considering a representative sample
2. Understand different types of data, and to present numerical and visual summaries using appropriate plots/figures
3. Understand the normal distribution and its characteristics
4. Perform one-sample and two-sample t-tests, and draw conclusion from the test results
5. Investigate linear relationship between two variables using simple linear regression model
6. Understand the importance of randomisation and simple experimental design
7. Perform descriptive analysis, one-sample and two-sample t-tests, and fit a linear regression model using R

Training plan for 3.2 Regression Analysis

Prior to the course

All students are required to study the material from 3.1 Basic Statistics module.

Students from mathematics, statistics, or physics background are required to learn Chapter 1, 2, 4, and 6 of the book “In All Likelihood” by Yudi Pawitan.

Course plan

There will be no special session on the implementation of the method in R as the students are assumed to be able to learn this independently after learning R (and python) in the previous training.

Course Syllabus

1. Linear regression
2. Logistic regression
3. Cox regression

Regression Analysis aims to give students the knowledge on statistical concept and applications of the linear regression, logistic regression, and Cox regression models.

Objectives

1. To describe both simple and multiple linear regression models
2. To describe the logistic regression model
3. To describe the Cox regression model
4. To describe the interpretation of the models and their parameters
5. To describe the issue of modelling with multiple variables

Learning Outcomes

On completion of this module, participants should be able to:

1. Understand when each of the regression models shall be implemented given a relevant research problem and data
2. Understand how to interpret the model parameter estimates and how a prediction is done (given new data)
3. Understand the issues on modelling with several predictor variables
4. Understand the types of test on the model parameters
5. Investigate model adequacy in all of the above regression models

Training plan for 3.3 Missing data

This course aims to provide a basic and practical introduction to deal with missing data in simple (linear regression) situations. The statistical concepts and underlying assumptions of imputation methods will be briefly discussed. Students will engage in basic hands-on exercises to implement the methods with R.

Objectives:

1. To recognize the types and patterns of missing data;
2. To understand and apply multiple imputation methods to deal with missing data: MAR (missing at random) and DL (detection limit) cases;
3. (For students with statistics background, and optional for others) To apply imputation methods based on a random forest;
4. To discuss how to present imputation methods and results.

Learning outcomes:

1. Students will understand the key statistical concepts, underlying assumptions and the relative merits of different statistical methods for dealing with missing data;
2. Students will be able to make a valid statistical analysis in various missing data situations with the R software;
3. Students can provide a clear report on the statistical analysis using imputation methods.

Prior knowledge:

1. Linear regression and statistical inference (effect estimate, confidence interval, and P-value);
2. Can perform data analysis using linear regression with R;
3. Papers to read:
    a. [Gentle_Intorduction.pdf] Review: A gentle introduction to imputation of missing values
    b. [DL.pdf] Evaluation of regression methods when immunological measurements are constrained by detection limits
    c. [MissForest.pdf] MissForest – nonparametric missing value imputation for mixed-type data

Which R-package download/install from CRAN.

1. mice: Multivariate Imputation by Chained Equations
2. missForest: Nonparametric Missing Value Imputation using Random Forest

Training plan for 3.4 Feature selection

The goal of this course module is to give students an introduction into methods for the analysis of high dimensional datasets.

Objectives:

1. To introduce students to challenges and pitfalls when analysing high dimensional datasets
2. To present different approaches for analysing high dimensional datasets

Learning outcomes:

1. Students will understand the shortcomings of standard regression techniques when analysing high dimensional data
2. Students will become aware of various statistical approaches to deal with high dimensional datasets

Training plan for 3.5 Programming R and Python

Prior to the course

For the R module students should have R and RStudio installed on their laptops. Rstudio can be found at https://www.rstudio.com/
For python, the Anaconda distribution (with python 3.6) should be installed and this can be found at https://www.anaconda.com/download/

Students are required to attend online courses from Software Carpentry in the following link:
http://swcarpentry.github.io/r-novice-inflammation/
http://swcarpentry.github.io/python-novice-inflammation/

Course syllabus

1. Programming approaches
2. Basics of programming
3. Data structures and data tables manipulation
4. Bootstrapping and randomization methods

Programming in R and Python aims to give students grounding in basic programming using R and Python, highlighting similarities and differences in their use.

Objectives

1. To explain organization of a code baseline
2. To introduce basic programming concepts of encapsulation, iteration and code reuse
3. To describe how to implement automated data pipelines
4. To introduce concepts of data structure (tidy data) and manipulation (tidyverse)
5. To introduce programming approach for analysis reproducibility and reliability
6. To describe randomization strategies for uncertainty evaluation

Learning outcomes

On completion of this module, participants should be able to:

1. Write data processing pipelines using iteration, encapsulation and reusable code
2. Manipulate simple databases in a form suitable for analysis and visualization
3. Summarize, describe and visualize the content of a database
4. Using randomization strategies to improve uncertainty estimation and visualization
5. Write reusable code whose correctness can be proven with tests.

Training plan for 3.6 Programming Git

The goal of the **Programming Git** course module is to introduce the students to the Git version control system and give them basic hands-on experience with it.

Prerequisites

- Students should have Git installed on their laptops (or have shell access to a Linux server with Git installed), see the instructions at
  https://swcarpentry.github.io/workshop-template/#shell
  https://swcarpentry.github.io/workshop-template/#git
- Students need to have a text editor installed on their laptops, for example Notepad++, Emacs, Vim, Sublime Text, see for example
  https://swcarpentry.github.io/workshop-template/#editor
- Some basic knowledge of/familiarity with the Linux/Bash command line will be very useful.
- Before the course, students should have gone through the following chapters of the Git course we will use in class:
  http://swcarpentry.github.io/git-novice/01-basics/
  http://swcarpentry.github.io/git-novice/02-setup/
  http://swcarpentry.github.io/git-novice/10-open/
  http://swcarpentry.github.io/git-novice/11-licensing/
  http://swcarpentry.github.io/git-novice/12-citation/

Objectives

1. The students will be introduced to the concept of version control, using Git as an example of a distributed version control system, so that they can use Git in their own workflow
2. The students will learn how to use Git to track changes in files (e.g. analysis scripts)
3. The students will learn the difference between the working directory and the repository
4. The students will learn how to show the differences between two versions of the same file and how to revert to an older version of a file
5. Depending on time/student level, more advanced concepts like branches and remote repositories will be discussed

Learning outcomes

1. The students can set up and use a Git repository to track changes in e.g. analysis scripts (R, Python):
   - Creating a repository
   - Adding files to the repository
   - Committing changes to the repository
   - Investigating the history of files using the Git log

- Comparing different versions of a file
- Checking out an older version of a file

2. Depending on their individual starting level and/or progress during the course, students will also be able to work with branches and remote repositories.

Training plan for 3.7 Network analysis

This course aims to provide a basic and practical introduction to network theory, introducing basic concepts, methods, and showing examples of applications in a biological context. Students will engage in basic hands-on exercises to do network analysis and it will be communicated a list of useful software in R and Python for the analysis and the visualization of networks.

Objectives:

1. To learn the basic definitions associated to a network;
2. To understand the context in which they can be applied, and with which purpose;
3. To have a survey of the software toolboxes that provide algorithms for network analysis.

Learning outcomes:

1. Students will understand the main features of a network;
2. Students will be able to build networks starting from specific data types (e.g. patient multidimensional profiling, multivariate time series);
3. Students will be able to use fundamental network algorithms (e.g. for centrality measures and clustering) available in high-level language packages (Python NetworkX, R?).

Prior knowledge:

1. Vector and matrix algebra (transpose, sum and product of a matrix, eigenvalue problem);
2. Basic statistical concepts: descriptive statistics, comparison tests (e.g. Student’s), covariance and correlation.