Lecture 1

- Using the command line
    - The Unix philosophy and `bash`
    - Remote computing with `ssh`
    - Version control with `git`
    - Documents with $\LaTeX$
    - Automation with `make`

Lecture 2

- The IPython notebook
    - Markdown cells
    - Code cells
    - The display system
    - IPython magic
    - Widgets and the interact decorator
    - Interfacing with other languages
- Programming in Python
    - Basic types
    - Basic collections
    - Control flow
    - Functions
    - Classes
    - Modules
    - The standard library
    - PyPI and `pip`
    - Importing other modules
    - Using `conda` virtual environments

Computer lab 1

- Exercise 1: Using `bash` for data scrubbing
- Exercise 2: Using `git` for collaborative editing
- Exercise 3: Generating a report with `make`, $\LaTeX$ and python

- Exercise 4: Functions to calculate mean, variance and Pearson correlation coefficient

Lecture 3

- Functional programming
    - Functions are first class objects
    - Pure functions
    - Iterators
    - Generators
    - Anonymous functions with lambda
    - Recursion
    - Decorators
    - The `operator` module
    - The `itertools` module
    - The `functools` module
    - The `toolz` module
    - Constructing a lazy data pipeline
- Working with text
    - string methods
    - The `string` module
    - The `re` module

Lecture 4

- Obtaining data
    - CSV with `csv`
    - JSON with `json`
    - Web scraping with `scrapy`
    - HDF5 with `pyhdf`
    - Relational databases and SQL with `sqlite3`
    - The `datasets` module
- Scrubbing data
    - Removing comments
    - Filtering rows
    - Filtering columns
    - Fixing inconsistencies
    - Handling missing data
    - Removing unwanted information
    - Derived information
    - Sanity check and visualization

Computer lab 2

- Exercise 1: Moving data from CSV to JSON to HDF to sqlite3
- Exercise 2: Querying a relational database
- Exercise 3: Constructing a functional pipeline using generators
- Exercise 4: Project Euler puzzle 1

Lecture 5

- Using `numpy`
  - Data types
  - Creating arrays
  - Indexing
  - Broadcasting
    - Outer product
  - Ufuncs
  - Generalized Ufuncs
  - Linear algebra in numpy
    - Calculating covariance matrix
    - Solving least squares linear regression
  - I/O in numpy
- Using `pandas`
  - Reading and writing data
  - Split-apply-combine
  - Merging and joining
  - Working with time series
- Using `blaze`

Lecture 6

- Graphics in Python
  - Using `matplotlib`
  - Using `seaborn`
  - Using `bokeh`
  - Using `daft`

Computer lab 3

- Exercise 1: Make a 12 by 12 times table chart without looping
- Exercise 2: Generate a 3×3 magic square
- Exercise 3: Working with some data set in `pandas`
- Exercise 4: Plotting the scatter matrix for the Iris data set in `matplotlib`, `seaborn` and `bokeh`

Lecture 7

- From math to computing
    - Computer representation of numbers
    - Overflow, underflow, catastrophic cancellation
    - Stability
    - Conditioning
    - Direct translation of symbols to code is dangerous
- The purpose of computing is insight not numbers
- Use of the computer in statistics
    - Estimating parameters (point and interval estimates)
    - Estimating functions
    - Approximating functions (especially PDFs)
    - Feature extraction, class discovery and dimension reduction
    - Classification and regression
    - Simulations and computational inference

Lecture 8

- Algorithmic efficiency and big $\mathcal{O}$ notation
- Some data structures
    - Performance of built-in data structures
    - Graphs and trees
- Algorithmic patterns
    - Divide and conquer
    - Decrease and conquer
    - Greedy algorithms
    - Dynamic programming
    - Hill climbing
    - Reduction and transformation
    - Monte Carlo

Computer lab 4

Exercise 1: Project Euler puzzle 2
Exercise 2: Project Euler puzzle 3

Exercise 3: Project Euler puzzle 4
Exercise 4: Project Euler puzzle 14

Lecture 9

- Numerical linear algebra
    - Simultaneous linear equations
    - Column space, row space and rank
    - Column space interpretation is most useful
    - Rank, basis, span
    - Norms and distance
    - Trace and determinant
    - Eigenvalues and eigenvectors
    - Inner product
    - Outer product
    - Einstein summation notation
- Matrices as linear transforms
    - Types of matrices
        - Square and non-square
        - Singular
        - Positive definite
        - Idempotent and projections
        - Orthogonal and orthonormal
        - Symmetric
        - Hermitian
        - Transition
    - Matrix geometry illustrated
- Matrix decompositions
    - LU (Gaussian elimination)
    - QR
    - Spectral
    - SVD
    - Cholesky
- Using `scipy.linalg`
- BLAS and LAPACK

Lecture 10

- Projections, ordination, change of coordinates
    - PCA in detail

    - PCA with eigendecomposition
    - PCA with SVD
    - Related methods
        - ICA
        - LSA
        - Factor analysis

Computer lab 5

- Exercise 1: Solving a least squares problem using Cholesky decomposition
- Exercise 2: Implement latent semantic indexing
- Exercise 3: Implement k-means clustering
- Exercise 4: Use latent semantic indexing to reduce a set of documents to 2D then use k-means to cluster them, and finally plot the result

Lecture 11

- Approximating functions
    - Orthogonal function basis
    - Splines
    - Kernel density estimation
- Estimating functions
    - Minimizing least squares
        - Linear regression example and the normal equations
    - Maximum likelihood
        - Logistic regression example
    - Bayesian posterior probability

Lecture 12

- Constrained and unconstrained optimization
- Discrete and continuous optimization
- Root finding and optimization in 1D
    - Taylor series
    - Root finding with Newton methods
    - Root finding with bisection and Brent's method
    - Line search optimization

Computer lab 6

- Exercise 1: Given a sequence of function values, write a program to perform kernel density estimation using several kernels
- Exercise 2: Use line search optimization to solve a 1D logistic regression problem
- Exercise 3: Implement the secant method for finding roots in 1D
- Exercise 4: Implement Newton's method and find an approximate solution to several equations (including ones that diverge)

Lecture 13

- Multivariate optimization
    - GD and SGD
    - Newton's method and IRLS for GLMs
- Other methods
    - Non-gradient based (e.g. Nelder-Mead)
    - Global optimization
    - Discrete optimization
        - Integer programming
        - Combinatorial optimization
- Packages for optimization
    - Using `scipy.optimization`
    - Using `statsmodels`
    - Using `scikits-learn`
    - Others: `cvxopt` and `pyopt`
- General approach to optimization
    - Know the problem
    - Multiple random starts
    - Combining algorithms
    - Graphing progress

Lecture 14

- The EM algorithm (1)
    - Convex and concave functions
    - Jansen's inequality
    - Missing data setup
    - Toy example - coin flipping with 2 biased coins

Computer lab 7

Exercise 1: Write the SGD function to solve a multivariate logistic regression problem using maximum likelihood
Exercise 2: Write the EM algorithm to solve another toy problem
Exercise 3: Playing with scipy.optimize

Exercise 4: Playing with scikits-learn

Lecture 15

- The EM algorithm (2)
    - Gaussian mixture model
    - EM for Bayesians - MAP of posterior distribution
    - Other applications of EM
    - EM variants

Lecture 16

- Monte Carlo methods
    - Random number generators
    - Generating random variates from a distribution
    - Quadrature and volume estimation
    - Estimate confidence intervals (bootstrap)
    - Compare competing statistics (statistical simulation - e.g. power)
    - Compare models (cross-validation)
    - Hypothesis testing (permutation-resampling)

Computer lab 8

- Exercise 1: Modify the EM algorithm for GMMs to find the MAP estimate of the posterior distribution
- Exercise 2: Use k-fold cross-validation to evaluate which is the best model for a given data set
- Exercise 3: Estimate the distribution of the slope in a linear regression model by bootstrapping on the residuals
- Exercise 4: Find the type-1 error for $\alpha = 0.05$ by using permutation resampling to correct for multiple testing

Lecture 17

- Conducting a simulation experiment (case study)
- Experimental design
    - Variables to study
    - Levels of variables (factorial, Latin hypercube)
    - Code documentation
    - Recording results
    - Reporting
    - Reproducible analysis with `make` and $\LaTeX$

Lecture 18

- MCMC (1)
    - Toy problem - rats on drugs
    - Monte Carlo estimation
    - Importance sampling
    - Metropolis-Hasting
    - Gibbs sampling
    - Hamiltonian sampling
    - Assessing convergence
    - Using `pystan`
    - Using `pymc2`
    - Using `emcee`

Computer lab 9

- Exercise 1: Writing a Gibbs sampler for change point detection
- Exercise 2: Using `pystan`
- Exercise 3:Using `pymc2`
- Exercise 4: Using `emcee`

Lecture 19

- MCMC (2)
    - Gaussian mixture model revisited
    - Gibbs sampling
    - Infinite mixture model with the Dirichlet process
    - Simulated tempering
    - MC^3

Lecture 20

- Profiling
    - Premature optimization is the root of all evil
    - Using `%time` and `%timeit`
    - Profiling with `%prun`
    - Line profiler
    - Memory profiler
- Code optimization
    - Use appropriate data structure
    - Use appropriate algorithm
    - Use known Python idioms
    - Use optimized modules
    - Caching and memoization
    - Vectorize and broadcast
    - Views
    - Stride tricks

Computer lab 10

- Exercise 1 The label-switching problem
- Exercise 2:Classifying points with the GMM:
- Exercise 3: Profiling source code
- Exercise 4 Optimizing source code

Lecture 21

- JIT compilation with `numba`
- Optimization with `cython`
- Wrapping C code
- Wrapping C++ code
- Wrapping Fortran code

Lecture 22

- Why modern CPUs are starving and what can be done about it
- Parallel programming patterns
- Amdahl's and GustClassifying points with the Gustafson's laws
- Parallel programming examples
    - JIT compilation with `numba`
    - Toy example - fractals
    - Using `joblib`
    - Using `multiprocessing`
    - Using `IPython.Parallel`
    - Using `MPI4py`

Computer lab 11

Exercise 1: Optimizing EM code with numba
Exercise 2: Optimizing EM code with Cython
Exercise 3: Parallel processing of embarrassingly parallel code
Exercise 4: Parallel processing of code requiring intr-process communication

Lecture 23

- GPU computing
    - Introduction to CUDA
    - Vanilla matrix multiplication
    - Matrix multiplication with shared memory

    - JIT compilation with `numba`
    - Matrix multiplication in OpneCL

Lecture 24

- Map-reduce and Spark
    - Problem - k-mer counting for DNA sequences
    - Small scale map-reduce using Python
    - Using `hadoop` with `mrjob`
    - Using `spark` with `pyspark`
    - Using `MLib` for large-scale machine learning

Computer lab 12

Exercise 1: Coding fractals in CUDA
Exercise 2: Something more statistical in CUDA
Exercise 3: Word count in map-reduce
Exercise 4: K-mer count with map reduce with E Coli and human genome

Data sets
- ecoli genome
- human genome