Efficient implementations of statistical test accounting for confounders on large-scale datasets

Supervisor: Fabian Woller

Several statistical test and models for assessing associations and dependencies between variables of different data types are widely used and implemented in several Python statistics packages (Scipy, pandas, Pingouin, …). However, most of these implementations lack intrinsic parallelization and are not well-suited for processing large-scale datasets with a significant amount of missingness. At the same time, for a meaningful analysis of (large, mixed-type) population cohort data, it is of general interest to employ statistical methods that are able to correct for confounding variables and effects within the data (Partial Correlation, ANCOVA, Multiple Linear Regression, …). In this project, we want to implement a selection of the above methods in a parallelized, more efficient way that allows us to perform statistical analyses on such large input data in less time than currently possible with any available Python tools.

The intended statistical tests and methods are supposed to be implemented in both Python (using the JIT-compiler Numba) and/or C++ in combination with pybind11. For a given test to be implemented, the rough steps could read as follows:

- Become familiar with underlying theory and mechanism of statistical test / model
- Investigate existing Python implementations and their efficiency
- Figure out parallelizability of respective statistical test
- Implement parallelized version in Python using Numba and/or in C++ with Python bindings
- Assess scalability / runtime efficiency / memory efficiency of the implemented test on simulated and/or real-world data

Requirements:

- Ability/Motivation to become familiar with the underlying theory and mathematical foundation of statistical tests
- Strong Python skills; ideally also C++ skills in combination with OpenMP
- Familiarity with basic concepts of parallelization