UNIVERSITY OF WARSAW

UNIVERSITY'S INTEGRATED DEVELOPMENT PROGRAMME

Course syllabus

Course title	Advanced Python for cognitive scientists
Instructor(s)	Jarosław Paszek PhD
Contact details	j.paszek@mimuw.edu.pl
Affiliation	Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
Course format	lecture, class
Number of hours	45 hours
Number of ECTS credits	**4 ECTS credits**
Brief course description	The goal of the course is to build fluency in using Python programming language as a tool for scientific computing, data manipulation and visualization. We will introduce libraries which constitute a core of Python ecosystem for data analysis: numpy, scipy, pandas, matplotlib. After covering the basics, students will have the opportunity to hone their skills by working through a number of applications of the introduced tools in data analysis. Simultaneously, they will be improving their programming style and learning about good programming practices. Previous experience with Python is necessary.
Full course description	This course is designed as a continuation of an introductory course of Python programming. It is assumed that students know the basics of language syntax and are able to write simple programs on their own. In this class they will expand their knowledge of the language, get to know popular Python libraries, and learn practical applications of their skills. In addition to imperative style of programming, already known to students, concepts of high-level array programming (based on numpy and pandas libraries) are introduced. The focus is on scientific computing and exploratory data analysis. Libraries covered include numpy, scipy, pandas, matplotlib. Students learn important aspects of data literacy: data preprocessing, data manipulation, data visualization. These practical skills are prerequisites for delving deeper into issues of computational modeling and data science.
Learning outcomes	Student knows and understands: - popular Python libraries for data analysis (K_W04) - concepts of exploratory data analysis and data visualization (K_W08) Student is able to: - perform basic data analysis, build data preprocessing pipeline, program experimental procedure in Python (K_U02, K_U04).

	- search for proper tools and software libraries to solve a particular task, experiment with different approaches (K_K01, K_K02)
Learning activities and teaching methods	The class will be conducted in a computer laboratory. It will consist of programming exercises interspersed with short lectures and demonstrations. Homework assignments are to be expected.
List of topics/classes and bibliography	1. Programming languages and paradigms. Review of Python syntax. 2. Advanced iteration and data structures (modules: collections, itertools). 3. Writing beautiful code. Stylistic exercises. PEP8 standard. 4. File loading, text processing, regular expressions. 5. Array processing in numpy. Vectorizing operations. 6. Advanced numpy indexing, broadcasting. 7. Basics of scientific computing with numpy and scipy. Implementing gradient descent algorithm. 8. Cellular automata. Game of life. Image processing with convolution matrices. 9. Shelling's segregation mode: implementing simple agent-based simulation. 10. Data munging with pandas I. Data frames, grouping, aggregation. 11. Data munging with pandas II. Time series. 12. Data visualizations. 13. Exploratory data analysis example. 14. Optimizing performance. Pytorch and numba. Literature: ● Kevin Sheppard (2016). Introduction to Python for Econometrics, Statistics and Numerical Analysis: Third Edition https://www.kevinsheppard.com/images/b/b3/Python_introduction-2016.pdf ● Nicolas P. Rougier (2017). From Python to Numpy http://www.labri.fr/perso/nrougier/from-python-to-numpy/ ● Wes McKinney (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython: Second Edition
Assessment methods and criteria	**Laboratory exercises (40%)** At least two times during the semester students will be given simple problems to solve individually during class. They will be graded on the spot by the instructor. **Homework assignments (60%)** At least three times during the semester students will be assigned short problems to solve at home. These assignments may have more open format. Time to solve each assignment should be 1-2 weeks.
Attendance rules	Two unexcused absences are allowed in the semester. Further unexcused absences may result in lowering the grade. If due to absences students

	miss graded exercises in class, they may be given an additional homework assignment.
Prerequisites	“Introduction to programming in Python” class or equivalent.
Academic honesty	Students must respect the principles of academic integrity. Cheating and plagiarism (including copying work from other students, internet or other sources) are serious violations that are punishable and instructors are required to report all cases to the administration.
Remarks	