**Санкт-Петербургский филиал федерального государственного**

**автономного образовательного учреждения высшего образования**

**"Национальный исследовательский университет "Высшая школа экономики"**

**Рабочая программа дисциплины**

Прикладной анализ данных в Python / Data Analysis with Python

для направления 39.06.01 «Социологические науки»

подготовки научно-педагогических кадров в аспирантуре,

образовательная программа «Социологические науки»

Разработчики программы

Кольцова Е.Ю., к. социол.н., доцент, ekoltsova@hse.ru

Согласована Академическим советом Аспирантской школы по социологическим наукам

«17» октября 2018 г., протокол № 34

Санкт-Петербург - 2018

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения разработчика программы.*

**Аннотация**

|  |  |
| --- | --- |
| Название дисциплины | **Прикладной анализ данных в Python**  |
| Образовательная программа | «Социологические науки» Направление подготовки 39.06.01 «Социологические науки» |
| Тип дисциплины | По выбору |
| Требования к уровню знаний аспирантов, необходимых для освоения дисциплины (пререквизиты) | Английский язык |
| Объем з.е. | 4 |
| Объем в часах | Аудиторная работа | Самостоятельная работа  | Всего |
| 28 | 124 | 152 |
| Краткое описание курса | Python - один из самых популярных и быстро развивающихся языков программирования. Понятный и выразительный синтаксис, облегчающий изучение, и множество встроенных и сторонних библиотек сделали Python особенно популярным среди ученых и исследователей из разных областей. Python уже некоторое время является основным языком в машинном обучении и науке о данных, но поскольку социальные науки становятся все более ориентированными на цифровые технологии, он также получает признание среди социологов, экономистов, лингвистов и других социальных учёных. Этот курс создан для аспирантов, которые хотят научиться решать реальные проблемы, связанные с анализом и обработкой данных, но не имеют опыта программирования. Программа курса охватывает большую часть возможностей Python от базового синтаксиса до современных библиотек для машинного обучения и анализа данных. |
| Образовательные результаты по дисциплине | **Компетенция** | **Дескрипторы – основные признаки освоения (показатели достижения результата)** | **Формы и методы обучения, способствующие формированию и развитию компетенции** | **Форма контроля уровня сформированности компетенции** |
| Способность к самостоятельному обучению новым методам исследования и к их развитию при решении задач профессиональной деятельности (ОПК-3) | Знает современные средства анализа и визуализации данных, использует их корректно при решении задач в фундаментальных и прикладных областях социологии | Лекции и семинары и применением метода проектов, исследовательского метода | Работа в классе, презентация проекта, контрольная работа, финальное эссе |
| Способностью анализировать, верифицировать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию и работать в условиях неопределенности (ОПК-10) | Применяет знаний для отбора и оценки качества источников информации  | Лекции и семинары и применением метода проектов, исследовательского метода | Работа в классе, презентация проекта, контрольная работа, финальное эссе |
| Способность самостоятельно формулировать цели, ставить конкретные задачи конкретных научных исследований в фундаментальных и прикладных областях социологии и решать их с помощью современных исследовательских методов с использованием новейшего отечественного и зарубежного опыта и с применением современных технических средств и информационных технологий, в том числе на английском языке (ПК-3) | Проводит исследование на всех его этапах, объясняет необходимость выбранного методы сбора, анализа данных, выбирает корректные современные техники работы с данными  | Лекции и семинары и применением метода проектов, исследовательского метода | Работа в классе, презентация проекта, контрольная работа, финальное эссе |
| Способность использовать методы и теории социальных и гуманитарных наук при осуществлении экспертной, консалтинговой и аналитической, преподавательской, проектной, проектно-социальной, проектно-управленческой деятельности (ПК-4) | Проводит исследование на всех его этапах, объясняет необходимость выбранного методы сбора, анализа данных, выбирает корректные современные техники работы с данными, дает оценку качества и надежности проведенных исследований и | Лекции и семинары и применением метода проектов, исследовательского метода | Работа в классе, презентация проекта, контрольная работа, финальное эссе |
| Краткое содержание дисциплины | В первой части курса рассматривается история Python, обсуждаются его сильные и слабые стороны, а также особенности, которые делают Python чрезвычайно полезным для социологов. Далее изучаются основные концепции программирования, такие как переменные, операторы, типы данных, функции, парадигмы программирования и обработчики ошибок. В это части курса аспиранты также знакомятся с пакетами и модулями, интерактивной оболочкой и средами разработки.Во второй части курса Python рассматривается как инструмент для анализа данных. Основное внимание здесь уделяется специализированным программным пакетам, позволяющими трансформировать данные и использовать модели машинного обучения. |
| Образовательные технологии | Интерактивные лекции и семинары |
| Формы контроля | Тесты (50%).Итоговый экзамен (50%). |
| Литература | Основная 1. Bernard, J. (2016). *Python Recipes Handbook: A Problem-Solution Approach*. Apress. Retrieved from //www.springer.com/gp/book/9781484202425
2. Shai Vaingast S. (2014) *Beginning Python Visualization*. Apress. Retrieved from //www.springer.com

Дополнительная1. Hetland, M. L. (2017). Beginning Python: From Novice to Professional (3rd ed.). Apress. Retrieved from //www.springer.com/gp/book/9781484200292
2. Gerrard, P. (2016). *Lean Python: Learn Just Enough Python to Build Useful Tools*. Apress. Retrieved from //www.springer.com/gp/book/9781484223840
3. Langtangen H.P. (2006) Python Scripting for Computational Science. Springer. Retrieved from //www.springer.com
4. Unpingco J. (2016) Python for Probability, Statistics, and Machine Learning. Springer. Retrieved from //www.springer.com
 |
| Преподаватель | Кольцова Елена Юрьевна, к.социол. н., доцент |

**Course Syllabus**

|  |  |
| --- | --- |
| Title of the course | **Data Analysis with Python (offered in English)** |
| Title of the Academic Programme  | Sociological Sciences |
| Type of the course  | Elective |
| Prerequisites | English Language |
| ECTS workload | 4 ECTS |
| Total indicative study hours | Directed Study | Self-directed study  | Total |
| 28 | 124 | 152 |
| Course Overview | Python is one of the most popular and rapidly developing programming languages. A clear syntax which facilitates learning and a plethora of built-in and third-party libraries made Python especially popular among academics and researchers of all kinds. Python has already been the first-choice language in Machine Learning and Data Science for a while, but as far as Social Sciences are becoming more digitally-oriented it is getting in demand by sociologists, economists, linguists, and other social researchers. This course is created for PhD students who want to learn how to solve real-world data-related problems with Python programming environment but have no experience in programming. The course syllabus covers most of Python functionality from basics syntax to the modern libraries for machine learning and data analysis. |
| Intended Learning Outcomes (ILO) | Being able to manage the complexity of the program using the techniques functional and object-oriented programming. Being able to write Python programs covering basic need of data scientist, namely:1. Loading data from external sources (.txt, .csv, .json, .sav, etc.).2. Scrape data from the web.3. Data wrangling and processing.4. Data visualisation.5. Models building. |
| Teaching and Learning Methods | Interactive lectures and seminars  |
| Content and Structure of the Course |
| **№** | **Topic / Course Chapter** | **Total** | **Directed Study** | **Self-directed Study** |
| 1 | Review of Python's history and features. Expressions: objects and operators | 19 | 4 | 15 |
| 2 | Statements and modules | 21 | 4 | 17 |
| 3 | Network requests and web scrapping | 24 | 4 | 20 |
| 4 | Errors Handling. Input/output. Programming paradigms: Functional and object-oriented programming. | 24 | 4 | 20 |
| 5 | Linear algebra with NumPy and Data munging with Pandas | 19 | 4 | 15 |
| 6 | Visualization | 24 | 4 | 20 |
| 7 | Machine learning with Scikit-learn and StatsModels | 21 | 4 | 17 |
| **Total study hours** | 152 | 28 | 124 |
| Indicative Assessment Methods and Strategy  | Tests (50%).Final exam (50%). |
| Readings / Indicative Learning Resources | Mandatory Bernard, J. (2016). *Python Recipes Handbook: A Problem-Solution Approach*. Apress. Retrieved from //www.springer.com/gp/book/9781484202425Shai Vaingast S. (2014) *Beginning Python Visualization*. Apress. Retrieved from //www.springer.comOptionalHetland, M. L. (2017). Beginning Python: From Novice to Professional (3rd ed.). Apress. Retrieved from //www.springer.com/gp/book/9781484200292 Gerrard, P. (2016). *Lean Python: Learn Just Enough Python to Build Useful Tools*. Apress. Retrieved from //www.springer.com/gp/book/9781484223840Langtangen H.P. (2006) Python Scripting for Computational Science. Springer. Retrieved from //www.springer.comUnpingco J. (2016) Python for Probability, Statistics, and Machine Learning. Springer. Retrieved from //www.springer.com |
| Indicative Self- Study Strategies | **Type** | **+/–** | **Hours** |
| Reading for seminars / tutorials (lecture materials, mandatory and optional resources) | + | 90 |
| Assignments for seminars / tutorials / labs | - |  |
| E-learning / distance learning (MOOC / LMS) | - |  |
| Fieldwork | - |  |
| Project work | - |  |
| Other (please specify) | - |  |
| Preparation for the exam | + | 34 |
| Academic Support for the Course | Academic support for the course is provided via LMS, where PhD students can find: guidelines and recommendations for doing the course; guidelines and recommendations for self-study; samples of assessment materials |
| Facilities, Equipment and Software | Personal computer with installed Python 3 interpreter. |
| Course Instructor | Elena Koltsova |

**Аnnex 1**

**Assessment Methods** **and Criteria**

**Assessment Methods**

|  |  |  |
| --- | --- | --- |
| **Types of Assessment** | **Forms of Assessment** | **Year 2**  |
| **Part 1** |
| Formative Assessment | Tests | 1 |
| Summative Assessment | Exam  | 1 |

**Assessment Criteria**

**Written Assignments (Tests)**

|  |  |
| --- | --- |
| **Grades** | **Assessment Criteria** |
| «Excellent» (8-10) | Has a clear argument, which addresses the topic and responds effectively to all aspects of the task. Fully satisfies all the requirements of the task; rare minor errors occur;  |
| «Good» (6-7) | Responds to most aspects of the topic with a clear, explicit argument. Covers the requirements of the task; may produce occasional errors. |
| «Satisfactory» (4-5) | Generally addresses the task; the format may be inappropriate in places; display little evidence of (depending on the assignment): independent thought and critical judgement include a partial superficial coverage of the key issues, lack critical analysis, may make frequent errors. |
| «Fail» (0-3) | Fails to demonstrate any appropriate knowledge. |

**Course Content**

**Part 1: Python basics.**

The first part of the course reviews the history of Python and discusses its strengths and weaknesses and features that make Python extremely useful for social scientists. It also covers the basic programming concepts such as variables, data types operators, functions and so on. Moreover, it gives an introduction to the applied tools essential for every programmer such as interactive shell, IDEs, and packages.

Duration: 10 hours.

**Lesson 1. Review of Python's history and features. Expressions: objects and operators**

1.1. History and development.

1.2. Features and philosophy.

1.3. Python as a primary language for scientific computing.

1.4. How to run Python program: command-line interface (CLI), notepad-like editors, integrated development environments (IDEs) and Jupyter.

1.5. Materials and resources for self-study.

1.6. Structure of Python program.

1.7. Expressions: data types and operators.

1.8. Control point: quiz.

*Example question:*

*1. Select the correct statements:*

*a) Python is compiled language.*

*b) Python has dynamic typing.*

*c) Python is low-level language.*

*d) Python is slow language.*

**Lesson 2. Statements and modules.**

2.1. if tests. while and for loops.

2.2. Iterations and Comprehensions.

2.3. Functions. Build-in functions.

2.4. Modules and packages. **pip** as a package manager. Review of the most useful packages.

2.4. Control point: quiz and practical assignment.

*Write a function that would count z-values for a given list of digits.*

**Lesson 3. Network requests and web scrapping.**

3.1. The structure of modern web.

3.2. http requests with **requests** module. Web-scrapping.

3.3. Parsing of HTML pages.

**Lesson 4. Errors Handling. Input/output. Programming paradigms: Functional and object-oriented programming.**

4.1. Errors handling and exceptions. try/except blocks and assert statement.

4.2. Input and Output. Working with files.

4.3. Control point: practical assignment.

4.4. What is programming paradigm and we need it.

4.5. Functional programming with Python: lambda functions, map and filter, functools package.

4.6. OOP programming.

*Load the tweets from the text file. Count the most frequent words. Filter out the words that occur in more than half tweets or that occur in the corpus less than 4 times. Count tf-idf statistic for the rest of the words.*

**Part 2: Python for data analysis**

The second part of the course deeps into Python as a tool for data analysis. The primary focus here is on the specialized programming packages allowing to do data analysis in a slick and easy way.

**Lesson 5. Linear algebra with NumPy and Data munging with Pandas.**

5.1. Why we need to use linear algebra? Vectors, matrices and basic operations.

5.2. New data types and methods introduced by NumPy.

5.3. Basic datatypes of Pandas: DataFrames and Series.

5.4. Input and Output in Pandas. Working with Excel files.

5.5. Indexing and Selecting Data in Pandas. In this section, we will focus on how to slice, dice, and generally get and set subsets of Pandas objects.

5.6. Grouping and aggregating data.

5.7. Reshaping & Pivoting data.

5.8. Merge, join, and concatenate.

5.9. Working with missing data.

5.10. Practical task.

*Load the image as a three-dimensional NumPy array. Cut the bottom of image. Transform the image to the grayscale image. Transform the image to the negative image. Guess, what you need to do with the NumPy array to create blurred/noisy/black-and-white image.*

*Make your data tidy.* *Tidy data sets main concept is to arrange data in a way that each variable is a column and each observation (or case) is a row. Tidy data provide standards and concepts for data cleaning, and with tidy data there's no need to start from scratch and reinvent new methods for data cleaning. Implement principles of Tidy data in Pandas on the real data using built-in functionality you studied before.*

**6. Visualization.**

6.1. Matplotlib as a core library for data visualization in Python.

6.2. Built-in visualization functionality in Pandas.

6.3. Seaborn. Python visualization library based on Matplotlib, which provides a high-level interface for drawing attractive statistical graphics.

6.3. Control point: practical assignment.

*Visualize the distributions of the variables from a given data set. Compare different plot and*

**7. Machine learning with Scikit-learn and StatsModels**

7.1. Linear regression.

7.2. Generalized Linear Models.

7.3. ANOVA.

7.4. Classification with logistic regression SVM, tree-based algorithms

*Using a given dataset create a model that explains the variation of the target variable. Choose the right model if this variable has numeric/binary/categorical statistical data type.*

**Special conditions for organization of learning process for PhD students with special needs**

The following types of comprehension of learning information (including e-learning and distance learning) can be offered to PhD students with disabilities (by their written request) in accordance with their individual psychophysical characteristics:

1. *for persons with vision disorders:* a printed text in enlarged font; an electronic document; audios (transferring of learning materials into the audio); an individual advising with an assistance of a sign language interpreter; individual assignments and advising.
2. *for persons with hearing disorders: a* printed text; an electronic document; video materials with subtitles; an individual advising with an assistance of a sign language interpreter; individual assignments and advising.
3. *for persons with muscle-skeleton disorders: a* printed text; an electronic document; audios; individual assignments and advising.