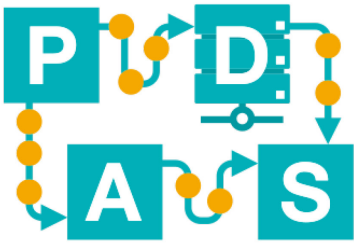
Crash Course in Python

Lecture 2

IDS-L2



Chair of Process and Data Science

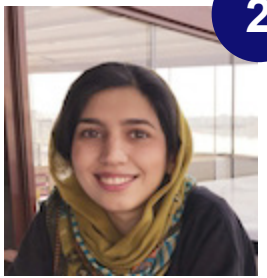
RWTH AACHEN UNIVERSITY

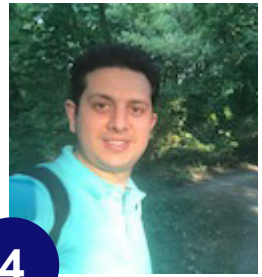
Outline of Today's Lecture

- Data science tools
- Python: features and design principles
- Basic examples
- Data structures manipulation

People involved

1. Wil van der Aalst
2. Mahsa Bafrani
3. Marco Pegoraro
4. Majid Rafiei
5. Yaguang Sun
6. Anja Syring





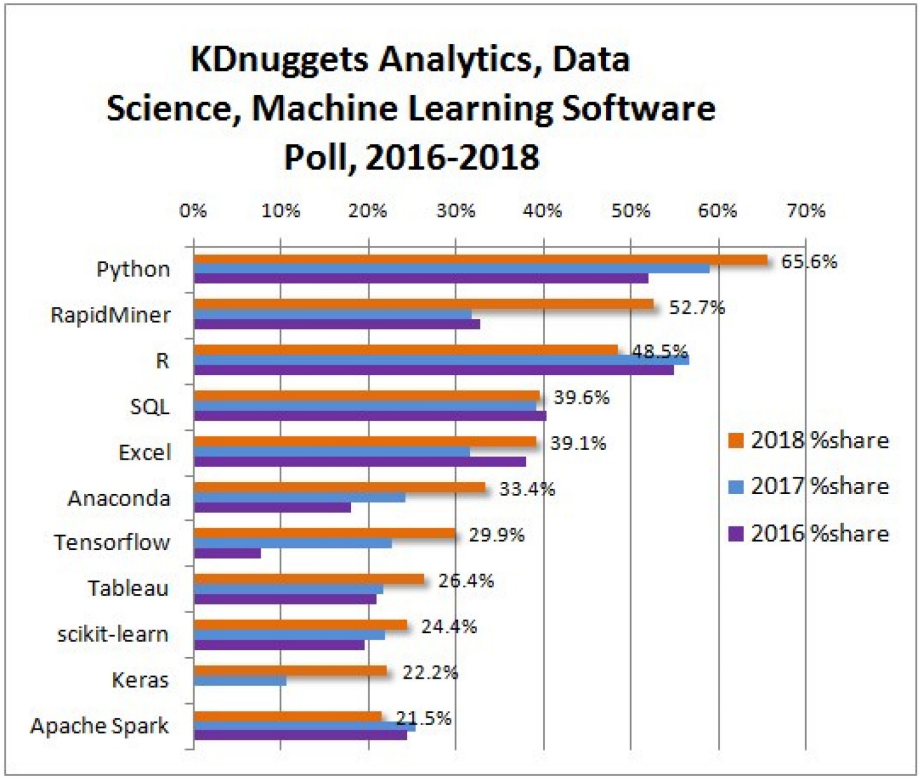
Chair of Process and Data Science

Data science tools

Thousands of data science tools

- Infrastructure (distributed computing and storage): Hadoop, Spark, mongoDB, etc.
- Programming / scripting languages tailored towards data analysis: Python, R, etc.
- Analysis tools using a visual workflow: Knime, RapidMiner, etc.
- BI tools: Tableau, Qlikview, PowerBI, etc.
- Statistical software: SAS, SPSS, etc.
- Traditional data mining: WEKA, etc.
- Specialized tools: Disco, Celonis, QPR, etc.

Kdnuggets Poll May 2018



KDnuggets Poll with over 2,300 voters (on average 7 tools per user), take with a grain of salt.

Motivation for Python

- Most widely used data science tool.
- One tool for the whole course (avoiding set-up time).
- Flexible, adaptable and allowing for bridges.
- Showing also a bit of the inside (not just pushing buttons).
- Open source, easy to get started.
- No visual workflow support.
- No UI tailored towards specific analysis tasks.
- Performance is good, but not the best in class.

Thousands of data science tools

- Infrastructure (distributed computing and storage): Hadoop, Spark, mongoDB, etc.
- Programming / scripting languages tailored towards data analysis: Python, R, etc.
- Analysis tools using a visual workflow: Knime, RapidMiner, etc.
- BI tools: Tableau, Qlikview, PowerBI, etc.
- Statistical software: SAS, SPSS, etc.
- Traditional data mining: WEKA, etc.
- Specialized tools: Disco, Celonis, QPR, etc.

Python: core features

Python: introduction

- First released in 1991 by Guido van Rossum
- General purpose
- Extremely popular: second most common language on Github
- Aimed to allow a very concise style of programming
  - Weakly and dynamically typed
  - Interpreted, object oriented
  - Imperative, but with strong functional support
  - Automatic memory management

Python: introduction

- **Designed to be accessible to many people**
  - **Easier than most languages to learn**
  - **Allows for very readable code**
- **Due to its conciseness and expressiveness it is appreciated in many scientific domains**
  - **e.g. Data Science, Machine Learning, Statistics**
- **One of the best languages to build and test algorithm prototypes**

Zen of Python

- The Zen of Python consists of a list of design principles for Python
- It has been included in the official documentation
- A selection of “tenets” describing these design principles is:

Zen of Python

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Errors should never pass silently.
- There should be one - and preferably only one - obvious way to do it.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.

Why Python?

Python Features

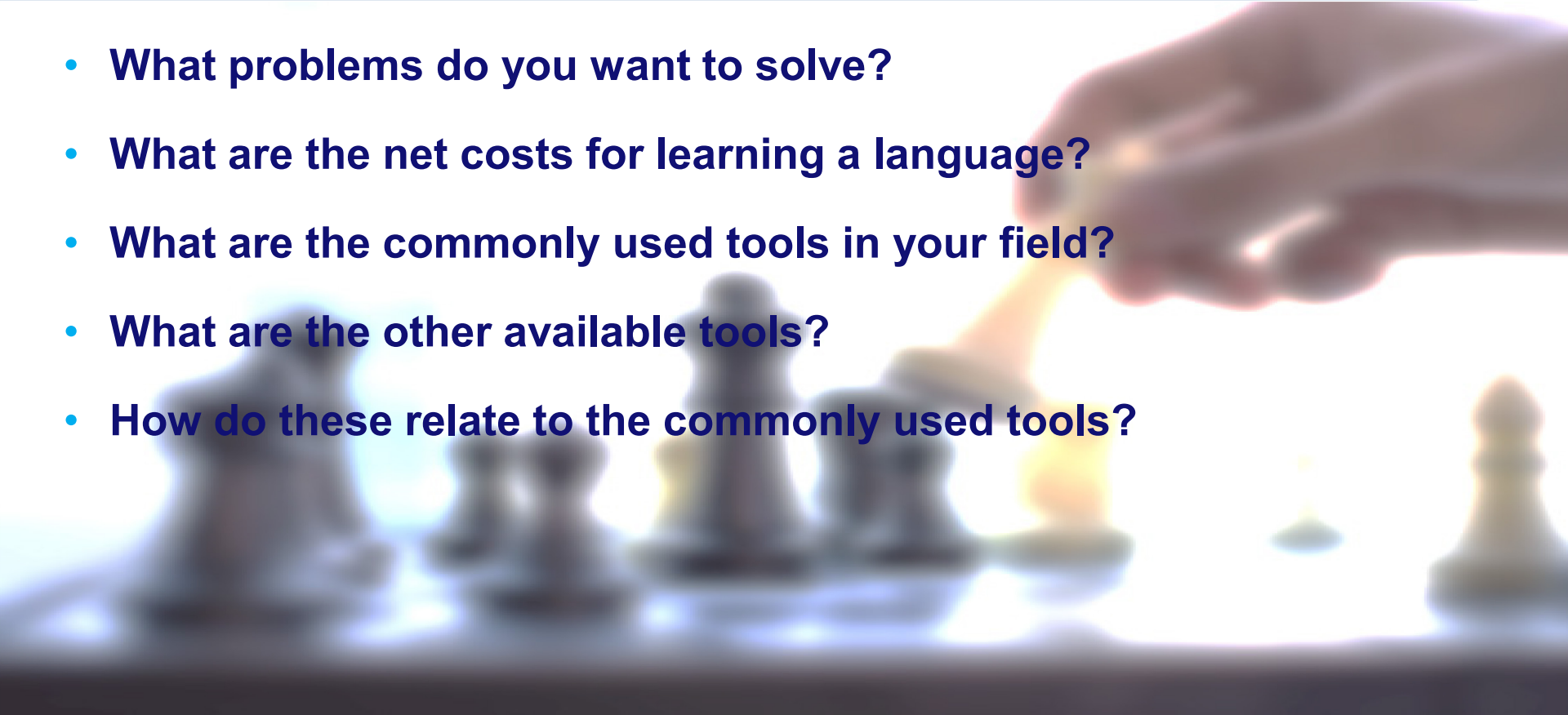
1) Easy to Learn and Use	Developer-friendly high level programming language.
2) Expressive Language	More understandable and readable
3) Interpreted Language	Easier debugging and thus suitable for beginners.
4) Cross-platform Language	Run in different platforms such as Windows, Linux, Unix and Macintosh etc.
5) Free and Open Source	The Source-code is available
6) Object-Oriented Language	Python supports object oriented concepts
7) Extensible	Other languages such as C/C++ can be used to compile the code It can be used further in our python code.
8) Large Standard Library	Python has a large and broad library Provides rich set of module and functions for rapid application development.
9) GUI Programming Support	Develop of graphical user interfaces

When?

- Integrated data analysis with web apps
- Statistics code needs to be incorporated into a production database
- Being a fully-fledged programming language
- To implement algorithms for production use

Time to Decide

- What problems do you want to solve?
- What are the net costs for learning a language?
- What are the commonly used tools in your field?
- What are the other available tools?
- How do these relate to the commonly used tools?



Examples

Conclusion

Python: a summary

- **Python: history and features**
- **Basics of the language (syntax and variables)**
- **Basics of data structure and their manipulation**

Python: the instruction

- **The first instruction of the course will be dedicated to Python:**
  - **More advanced data structure manipulation**
  - **Basics of OOP in Python**
  - **Basics of Pythonic coding style**
  - **Overview of the data science Python packages for the assignments**

Relevant Literature

- P. Barry, “Head First Python” (somewhat playful)
- M. Lutz and D. Ascher, “Learning Python”
- J. Knupp, “Writing Idiomatic Python”