EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Philosophische Fakultät
Seminar für Sprachwissenschaft

Data Science for Linguists

Session 1: IPython and Jupyter

Johannes Dellert

27 October, 2023

Table of Contents

What is Data Science?

- a **data scientist** is sometimes simply defined as a person with workable coding skills, a good background in statistics, and knowledge of a relevant domain
- these three areas map nicely to the prerequisites for this course:
  - Methods I: Programming
  - Methods II: Statistics
  - Linguistic Fundamentals (domain knowledge)
- but this definition would also fit a **data analyst**, who is typically described as using the very same skills to answer questions asked by other people
- the crucial difference is that a scientist will ask their own questions
- this course could be described as getting you on track towards combining these three areas of pre-existing knowledge in a productive way, allowing you to do data-based science:
  - ask meaningful questions that can be answered based on empirical data
  - develop strategies for data acquisition, preprocessing and reshaping
  - code statistical analyses which can provide answers to your questions

Table of Contents

Why is Data Science Relevant for Linguistics?

- many research questions asked in various branches of linguistics can only be answered based on large amounts of data (instead of e.g. hand-curated sets of example cases)
- linguistic data comes in various shapes depending on the subdiscipline:
  - audio and video recordings (the result of experiments of fieldwork)
  - eyetracking or other measurement data
  - scanned grammars describing many languages
  - digitalised dictionaries and lexical databases
  - curated typological databases (grammatical features across languages)
  - annotated corpora of various types (newspaper, literature, movie subtitles)
  - large amounts of raw text data
  - crowdsourced lexical and encyclopedic information (Wiktionary, Wikipedia)
- in modern science, there will be more relevant data than we could ever process manually
- exploratory data analysis is necessary to understand what is contained in a dataset
- statistical tests are necessary to decide whether there is a signal or only random noise
- modeling is necessary to understand the dynamics of complex systems

Table of Contents

Session 01: IPython and Jupyter

- Questions you will be able to answer after this session:
  - What are some of the advantages of IPython over vanilla Python?
  - How do I set up a Jupyter notebook, and what is the basic workflow?
  - How can I use Jupyter's capabilities to efficiently perform a small data analyis?

Session 02: Introduction to NumPy

- Questions you will be able to answer after this session:
  - Why is it standard practice to convert large datasets into arrays of numbers?
  - Why should I not use nested Python lists to represent numerical arrays?
  - How do I set up and populate large arrays using NumPy?
  - How can I slice, reshape, join and split NumPy arrays?
  - Why are universal functions preferable to loops?
  - How do I perform basic aggreggation tasks in NumPy?
  - How do broadcasting and masks work?
  - How can I use the options for fancy indexing in order to bin data?
  - How do I sort arrays along rows and columns?

Session 03: Pandas and Data Frames

- Questions you will be able to answer after this session:
  - What is a data frame, and why is it so useful?
  - What is the nature of `Series` and `Index` objects in Pandas?
  - How do I select and filter in order to work with subsets of my data?
  - What are my options for sorting and ranking data in my data frame?
  - How do I compute basic summarising and descriptive statistics?

Session 04: Data Cleaning and Preparation

- Questions you will be able to answer after this session:
  - How do I get data in various formats into my data frames?
  - What are some basic strategies for handling missing data?
  - How do I efficiently remove duplicates?
  - What are the best options for replacing certain values?
  - How do I detect and filter outliers?
  - How do I efficiently create random samples?
  - How can I work with categorical data?

Session 05: Linguistic Preprocessing

- Questions you will be able to answer after this session:
  - TODO

Session 06: Data Wrangling - Join, Combine, Reshape

- Questions you will be able to answer after this session:
  - What is hierarchical indexing?
  - How do I compute summary statisings by level?
  - How can I merge together data frames using the join operation?
  - What are the options for merging datasets, and how do I execute them?
  - How do I reshape data using hierarchical indexing?
  - How do I pivot data between long and wide formats?

Session 07: Data Aggregation and Grouping

- Questions you will be able to answer after this session:
  - What is the best way to think about group operations?
  - Which options for grouping are supported best by Pandas?
  - How do I perform column-wise and multiple function application?
  - How does the split-apply-combine work?
  - What is cross-tabulation, and what are its main uses?

Session 08: Visualisation with Seaborn

- Questions you will be able to answer after this session:
  - TODO

Session 09: Modeling and Prediction

- Questions you will be able to answer after this session:
  - recap of statistical modeling
  - recap of linear regression
  - polynomial regression
  - logistic regression

Session 10: Classification

- Questions you will be able to answer after this session:
  - Naive Bayes classification
  - Support Vector Machines
  - Decision Trees and Random Forests

Session 11: Clustering

- Questions you will be able to answer after this session:
  - How can clustering help me to infer some structure over a large number of datapoints?
  - k-Means Clustering
  - Gaussian Mixture Models
  - Unsupervised Learning

Session 12: Pattern Extraction and Density Estimation

- Questions you will be able to answer after this session:
  - network analysis
  - Principal Component Analysis
  - Manifold Learning
  - Kernel Density Estimation

Session 13: Statistical Inference

- Questions you will be able to answer after this session:
  - pitfalls of statistical tests
  - resampling methods
  - multiple testing

Session 13: Data Science Projects

- Questions you will be able to answer after this session:
  - setting up a project plan
  - data access and data ethics
  - sharing and collaboration

Table of Contents

IPython: Setup

- if you already have Python installed, installing IPython should be as easy as this:

```
$ pip3 install ipython
```

- otherwise, follow instructions on the webpage (`https://ipython.org/install.html`)
- IPython should now be callable via terminal using an `ipython` command (analogous to the `python` command of vanilla Python)

IPython: Improved Features

- advantages of IPython over vanilla Python:
  - much better copying and pasting of formatted Python code
  - more intelligent and readable output formatting
  - very good command completion and other options for saving keyboard strokes

IPython: Magic Commands

- IPython comes with a range of special **magic commands** prefixed by `%`:
  - `%run` allows you to execute external script files as part of the code
  - `%pwd` shows the current working directory
  - `%timeit` measures and reports how long a statement takes to execute

IPython: Debugging and Profiling

Table of Contents

Jupyter

- Jupyter provides useful interactive interfaces to IPython (and other kernels)
- most recent interface, and likely soon the standard for data analyses in Python: the **Jupyter Lab** (browser-based integrated environment for data science)
- we will rely on the classic and much simpler **Jupyter Notebook**, which is little more than a browser-based editor for project files called **notebooks**
- file format has the ending `.ipynb` (“IPython notebook”), this is a very convenient JSON-based format for sharing your data analysis projects with others

Jupyter: Setup

- installation should be just as easy as for IPython (if not: `https://jupyter.org/install`)

```
$ pip3 install notebook
```

- to run the notebook:

```
$ jupyter notebook
```

- after startup, you should see the notebook dashboard contents of your personal directory in the browser window you were directed to (it is actually hosted on the local machine)
- navigate to the directory where you want to create your first notebook file
- the New dropdown button is in a slightly unintuitive position (to me) on the upper right, this is where you create a new empty notebook (choosing IPython as the kernel if several options)

Jupyter Notebooks: Basic Usage

- at its core, a Jupyter notebook consists of a sequence of numbered **cells** in which you interact with the IPython interpreter which runs in the background
- cell you are currently editing is highlighted in green
- to execute a cell: Shift + Enter while it is selected
- to delete a cell: mark it (should be highlighted in blue), then Shift + Backspace
- there are also **Markdown cells** which allow you to insert formatted explanations in between your code cells
- closing the notebook will not shut down the server!

Jupyter Notebooks: Further Useful Features

- Notebooks can be exported to various formats: PDF, HTML, . . .

Table of Contents

Course Organization

- **practical seminar** consisting of 14 sessions, leads to completion of a data science project
- goals of this course format:
  - acquire ability to work with current standard tools of data science in Python
  - achieve a good overview of algorithms and Python libraries for data modeling tasks
  - taking a deep dive into a dataset of your own choice
  - practice in defining a data science project, and carrying it out within time constraints
- mandatory parts of coursework during the semester:
  - **attendance** (talk to me in case there are exceptional circumstances)
  - **assignments** (requirements and possibility of group work will depend on participants)
- structure of course sessions: introduction to new concepts during the first half (typically a presentation), work on an assignment building on the new concepts in the second half
- course concludes with a semester **project** (more information next time)
- by default, you receive a **graded 6 CP Schein**, but you can register it as ungraded
- initial **registration is via the Moodle**

Table of Contents

Assignment 1: Tasks

1) Set up IPython and Jupyter on your machine, following instructions for your operating system.
2) Create a new Jupyter notebook, and play around to familiarize yourself with the workflow.
3) Some simple tasks to brush up on your Python (in case you have not programmed in a while):
    a) Load the contents of the UTF-8 encoded file `sq-sample.txt` into a single string.
    b) Tokenize the text by splitting it at whitespaces and linebreaks, turn all tokens to lowercase and remove punctuation symbols (`,`, `.`, etc.), store result in a list.
    c) Write a function which converts a list of tokens into a `Counter` of token frequencies.
    d) Separate the tokenized text into ten chunks of equal size, run them through the function.
    e) Use `defaultdict` to create a map from tokens to a set of those chunk IDs where the token was among the top-625 tokens (i.e. roughly equivalent to A1 level).
4) Based on the objects resulting from 3), answer the following questions:
    a) How many words were in the top-625 across all of the ten chunks?
    b) Create a two-dimensional array of top-625 overlaps for each pair of chunks.
    c) Is there a chunk which seems to be especially divergent from the rest of the text?
    d) Does this result tell you anything useful about frequency lists?

Preliminary Course Plan

1	**27/10**	**IPython and Jupyter**
2	03/11	Introduction to NumPy
3	10/11	Pandas and Data Frames
4	17/11	Data Cleaning and Preparation
5	24/11	Linguistic Preprocessing
6	01/12	Data Wrangling: Join, Combine, Reshape
7	08/12	Data Aggregation and Grouping
8	15/12	Visualisation with Seaborn
9	22/12	Modeling and Prediction
10	12/01	Classification
11	19/01	Clustering
12	26/01	Pattern Extraction and Density Estimation
13	02/02	Statistical Inference
14	09/02	Data Science Projects

Questions

Questions?

Comments?

Suggestions?