

Data Mining using Python

— course introduction

Finn Årup Nielsen

DTU Compute
Technical University of Denmark

September 1, 2014

Data Mining using Python

DTU course 02819 Data mining using Python.

Previously called DTU course 02820 Python programming (study administration wanted another name).

Project course with a few introductory lectures, but mostly self-taught.

Deliverables: A report, a poster and an oral presentation at the poster about a Python program you write in a group.

Teacher: Finn Årup Nielsen

Tentative schedule for autumn 2014

1. September Installation

8. September. Introduction to the Python language.

15. September. Numerical NumPy, SciPy, MatPlot (“Python as Matlab”)

22. September. Databasing, web and text processing, “natural language processing”

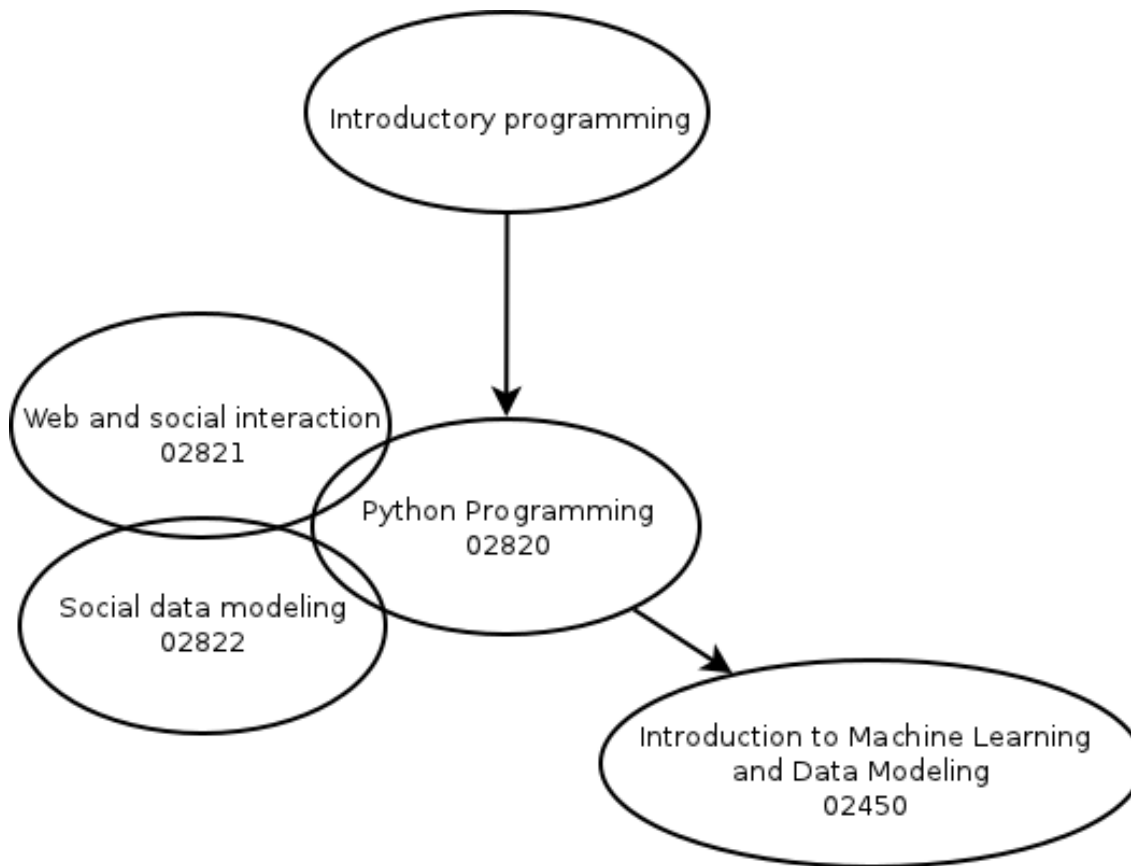
29. September Misc., e.g., GUI, Web serving

Project work for the rest of the time

December: Exam and report hand-in

See links to PDF on <http://www.compute.dtu.dk/courses/02820/>

Other courses



Introductory programming and mathematical modelling (linear algebra, statistics, machine learning)

Some overlap with 02805 (Social graphs and interaction), 02806 Social data analysis and visualization, 02821 (Web og social interaktion) and 02822 (Social data modellering).

If you take several 028xx courses be sure that you do **not** make a project that overlaps with projects in these courses in any way!

Project

Project: (Idea), design, implementation, testing, documentation.

Performed preferably in groups of two persons. Three is also ok.

Should preferably contain components of:

- Mathematical (numerical, computational, statistical or machine learning) modeling
- Internet/data/text mining

Poster

Construct a poster. Often A0/A1-sized.

“Defend” the poster, i.e., give a relatively short oral presentation of the poster and answer questions: Usually a ten minutes presentation for a two-person group with some questions afterwards.

Inspired from DTU course 02459 Machine Learning for Signal Processing



Why Python?

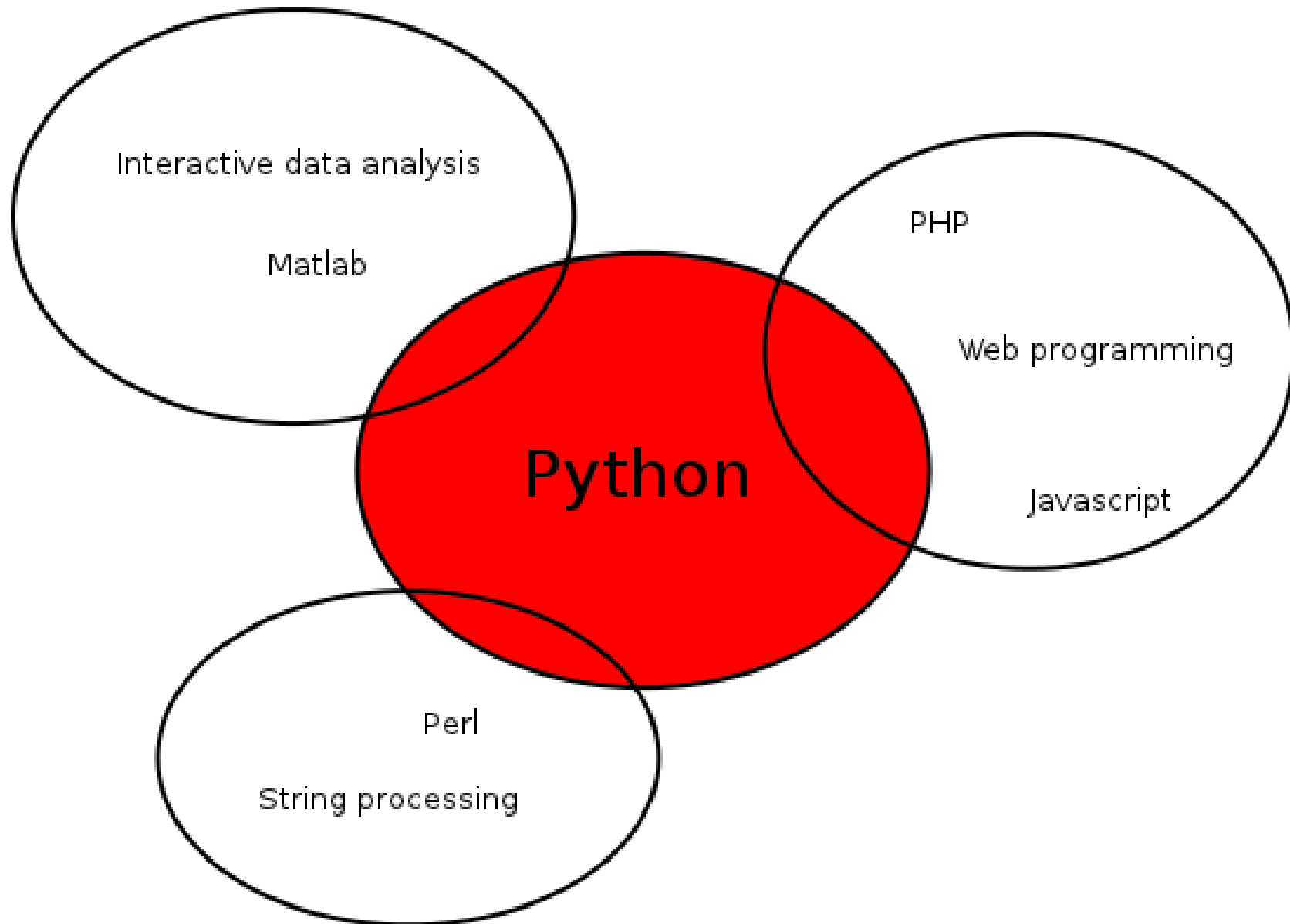
Interpreted, readable (usually clearer than Perl), interactive, many libraries, runs on many platforms, e.g., Nokia smartphones (hmmm...) and Apache Web servers.

With Python one can construct numerical programs, though with a bit more boilerplate than Matlab.

Google and Yahoo! is (has been?) using it. 2.73% of Open Source code written in Python (Black Duck Software, 2009).

“Without [Python] a project the size of Star Wars: Episode II would have been very difficult to pull off.” — <http://python.org/about/quotes/>

XKCD 353: “I wrote 20 short programs in Python yesterday. It was wonderful. Perl I’m leaving you.”



Why Python? Interactive language!

Interactive session

```
$ python
```

```
Python 2.4.4 (#2, Oct 22 2008, 19:52:44)
```

```
[GCC 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> 1+1
```

```
2
```

However, Matlab-like computation is not straightforward, e.g., what is the result of

```
>>> 1/2
```

Why Python? Interactive language!

Interactive session

```
$ python
```

```
Python 2.4.4 (#2, Oct 22 2008, 19:52:44)
```

```
[GCC 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> 1+1
```

```
2
```

However, Matlab-like computation is not straightforward, e.g., what is the result of

```
>>> 1/2
```

```
0 # Integer division! (in Python2 --- not Python3)
```

Example projects for inspiration

1. Characterize external links from DTU's Web-site.
2. Characterize internal link structure on DTU.
3. A search engine for DTU Web-pages.
4. Sentiment analysis of Tweets, blogs or news articles.
5. A wiki-based database for brain activations
6. A Web-service for visualization of brain activations.
7. Suggest one yourself

How do we evaluate the project?

Possible dimensions for evaluation of project?

Coding style

Bad: Variables are given incoherent names. Indentations are inconsistent.

Good: Variables are given intuitive and readable names. Code has been checked with flake8 and pylint.

Evaluation: Reusability of code

Bad: Input variable values are hard-coded. Code is repeated to make it look 'big'.

Good: Code is in meaningful modules. It is no problem to apply the data mining on new data. Part of the code can be used in other contexts.

Evaluation: Amount of data

Bad: The system is only able to handle a small amount of prespecified data and not likely anything else.

Good: The system use a 'large' amount data. The system use a database or other structured way of accessing a large amount of data.

Evaluation: Data mining effort

Bad: Simple analysis is performed. No use of Numpy, Scipy or other data mining package. Data is just entered, stored and 'copied around'.

Good: Machine learning or other complex analysis is performed.

Evaluation: Testing

Bad: No tests.

Good: A part of the code is tested.

Better: As much as feasible of the code is tested and with a variety of input and with the standard tools of Python testing. Testing coverage is computed and reported. Testing is performed on multiple versions of Python.

Evaluation: Documentation

Bad: There is no documentation. No use of docstring.

Good: Docstrings are used.

Better: Docstrings are used and used according to Numpy and other conventions. The documentation is checked with the pep257 program and no errors are found. Online documentation is generated with sphinx is available.

Evaluation: ‘Well-presented’ results

Bad: A plot in Excel is used with unlabeled axes.

Good: Data analysis results and other presentation with a number of Python tools, Matplotlib, etc., utilized in depth.

Better: A responsive interactive environment (perhaps web-based) is made where the user can navigate the result such zooming and panning as well as get the data results in a suitable format for further processing.

Evaluation: other dimensions

Effective and 'good' code. Shows a good command of Python . . .

Amount of code (but not code that is constructed to look big, by unnecessary repetitions and bad implementation).

Relevance of project: Is there a interesting (scientific) result or possibility for commercial application?

Originality of project . . . !?

More information

Learning objective: “Identify relevant learning material”. You yourself need to identify the appropriate Python documentation!

<http://www.python.org/>

The Python Tutorial <http://docs.python.org/tutorial/>

Internet search engines: Google, Bing or Yahoo.

[Stack Overflow](#), . . .

Google for error messages, “[Python tutorial](#)”

[MATLAB commands in numerical Python \(NumPy\)](#) by Vidar Bronken Gundersen if you know Matlab or R.

Free books

Dive into Python, ([Pilgrim, 2004](#)). Free, old and good.

With `sudo aptitude install diveintopython` it is available at `file:///usr/share/doc/diveintopython/html/index.html`

Think Python: How to Think Like a Computer Scientist and *How to Think Like a Computer Scientist*. Covers the basics of the Python language and Tkinter GUI. Also available as Wikibooks: *Think Python* and *How to Think Like a Computer Scientist: Learning with Python 2nd Edition*.

General books

Practical Programming. An introduction to computer science using Python, (Campbell et al., 2009): Introductory programming. Good if you are unsure.

Python cookbook (Martelli et al., 2005): Short program examples for somewhat specific problems. Too specific.

Specialized books relevant for the course

Programming collective intelligence (Segaran, 2007): Python and machine learning with data from the Web.

Natural language processing with Python (Bird et al., 2009): Text mining with Python. On paper and available online from <http://nltk.org>

Programming the Semantic Web (Segaran et al., 2009)

Mining the Social Web (Russell, 2011) Used(?) in on DTU courses. Maybe good.

Bioinformatics Programming Using Python, (Model, 2009). Introductory book to Python programming with emphasis on bioinformatics.

Data analysis and numerics books

Kevin Sheppard's *Introduction to Python for Econometrics, Statistics and Data Analysis* on 381 pages covers both Python basics and Python-based data analysis with Numpy, SciPy, Matplotlib and Pandas, — and it is not just relevant for econometrics ([Sheppard, 2014](#)).

([Langtangen, 2005](#); [Langtangen, 2008](#)): Python book with many examples especially for numerical processing. 2005 edition not fully up to date on numerical Python. 2008 version should be [available online](#) through DTU library

My draft *Data Mining with Python*.

Other books

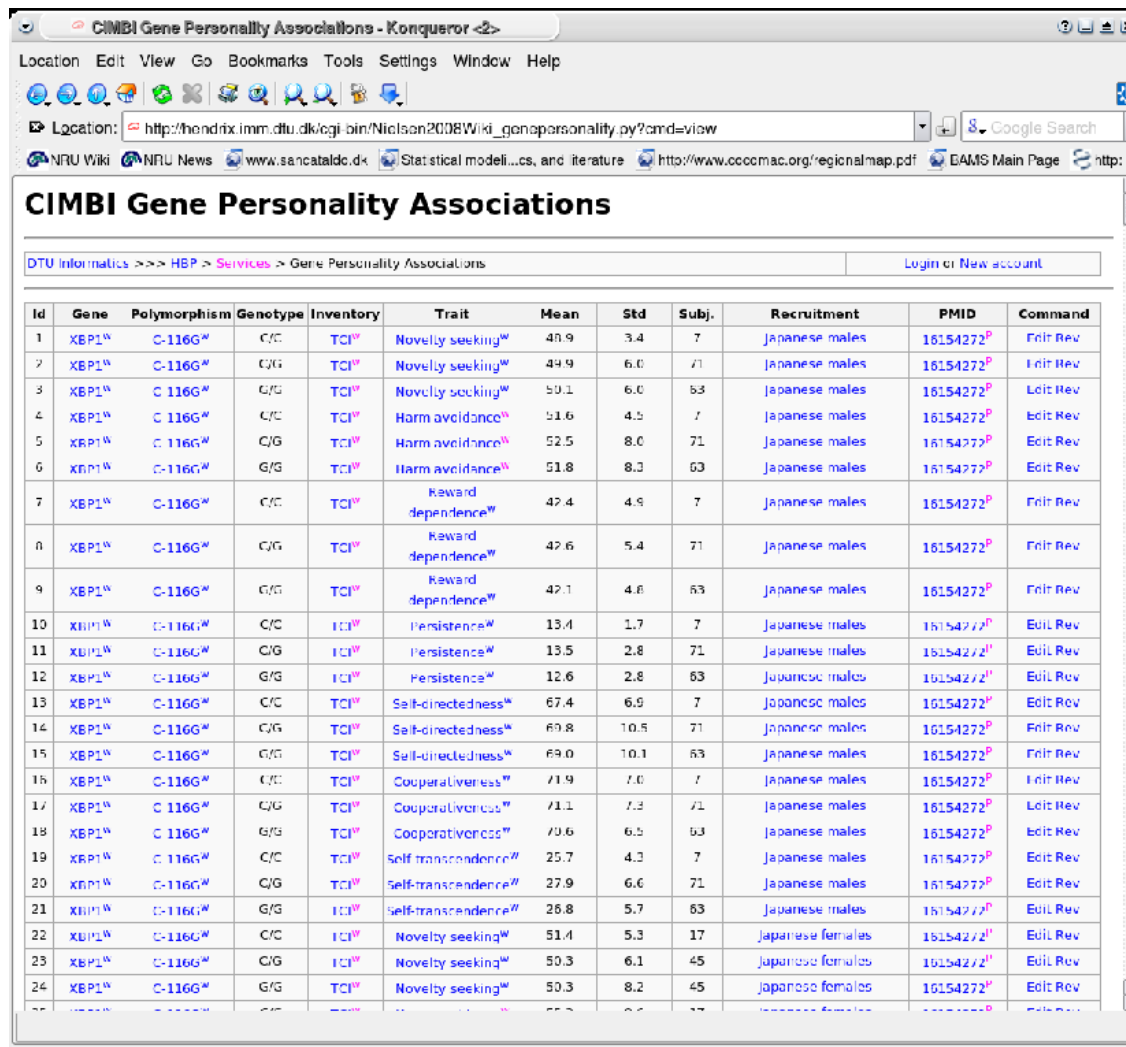
Other O'Reilly titles: Python in a Nutshell, Python Pocket Reference, Learning Python, Programming Python?

Other books that I know of:

Mobile Python ([Scheible and Tuulos, 2007](#)): On Nokia smartphone. Dead end.

Python Essential References ([Beazley, 2000](#)): Introduction and list of Python functions with small examples. Somewhat old and not recommendable.

Example: A fielded wiki . . .



Id	Gene	Polymorphism	Genotype	Inventory	Trait	Mean	Std	Subj.	Recruitment	PMID	Command
1	XBP1A	C-116G	C/C	TCI	Novelty seeking	48.9	3.4	7	japanese males	16154272 ^P	Edit Rev
2	XBP1A	C-116G	C/G	TCI	Novelty seeking	44.9	6.0	71	japanese males	16154272 ^P	Edit Rev
3	XBP1A	C-116G	G/G	TCI	Novelty seeking	59.1	6.0	63	japanese males	16154272 ^P	Edit Rev
4	XBP1A	C-116G	C/C	TCI	Harm avoidance	51.6	4.5	7	japanese males	16154272 ^P	Edit Rev
5	XBP1A	C-116G	C/G	TCI	Harm avoidance	52.5	8.0	71	japanese males	16154272 ^P	Edit Rev
6	XBP1A	C-116G	G/G	TCI	Harm avoidance	51.8	8.3	63	japanese males	16154272 ^P	Edit Rev
7	XBP1A	C-116G	C/C	TCI	Reward dependence	42.4	4.9	7	japanese males	16154272 ^P	Edit Rev
8	XBP1A	C-116G	C/G	TCI	Reward dependence	42.6	5.4	71	japanese males	16154272 ^P	Edit Rev
9	XBP1A	C-116G	G/G	TCI	Reward dependence	42.1	4.8	63	japanese males	16154272 ^P	Edit Rev
10	XBP1A	C-116G	C/C	TCI	Persistence	13.4	1.7	7	japanese males	16154272 ^P	Edit Rev
11	XBP1A	C-116G	C/G	TCI	Persistence	13.5	2.8	71	japanese males	16154272 ^P	Edit Rev
12	XBP1A	C-116G	G/G	TCI	Persistence	12.6	2.8	63	japanese males	16154272 ^P	Edit Rev
13	XBP1A	C-116G	C/C	TCI	Self-directedness	67.4	6.9	7	japanese males	16154272 ^P	Edit Rev
14	XBP1A	C-116G	C/G	TCI	Self-directedness	69.8	10.5	71	japanese males	16154272 ^P	Edit Rev
15	XBP1A	C-116G	G/G	TCI	Self-directedness	69.0	10.1	63	japanese males	16154272 ^P	Edit Rev
16	XBP1A	C-116G	C/C	TCI	Cooperativeness	71.9	7.0	7	japanese males	16154272 ^P	Edit Rev
17	XBP1A	C-116G	C/G	TCI	Cooperativeness	71.1	7.3	71	japanese males	16154272 ^P	Edit Rev
18	XBP1A	C-116G	G/G	TCI	Cooperativeness	70.6	6.5	63	japanese males	16154272 ^P	Edit Rev
19	XBP1A	C-116G	C/C	TCI	Self-transcendence	25.7	4.3	7	japanese males	16154272 ^P	Edit Rev
20	XBP1A	C-116G	C/G	TCI	Self-transcendence	27.9	6.6	71	japanese males	16154272 ^P	Edit Rev
21	XBP1A	C-116G	G/G	TCI	Self-transcendence	26.8	5.7	63	japanese males	16154272 ^P	Edit Rev
22	XBP1A	C-116G	C/C	TCI	Novelty seeking	51.4	5.3	17	japanese females	16154272 ^P	Edit Rev
23	XBP1A	C-116G	C/G	TCI	Novelty seeking	50.3	6.1	45	japanese females	16154272 ^P	Edit Rev
24	XBP1A	C-116G	G/G	TCI	Novelty seeking	50.3	8.2	45	japanese females	16154272 ^P	Edit Rev

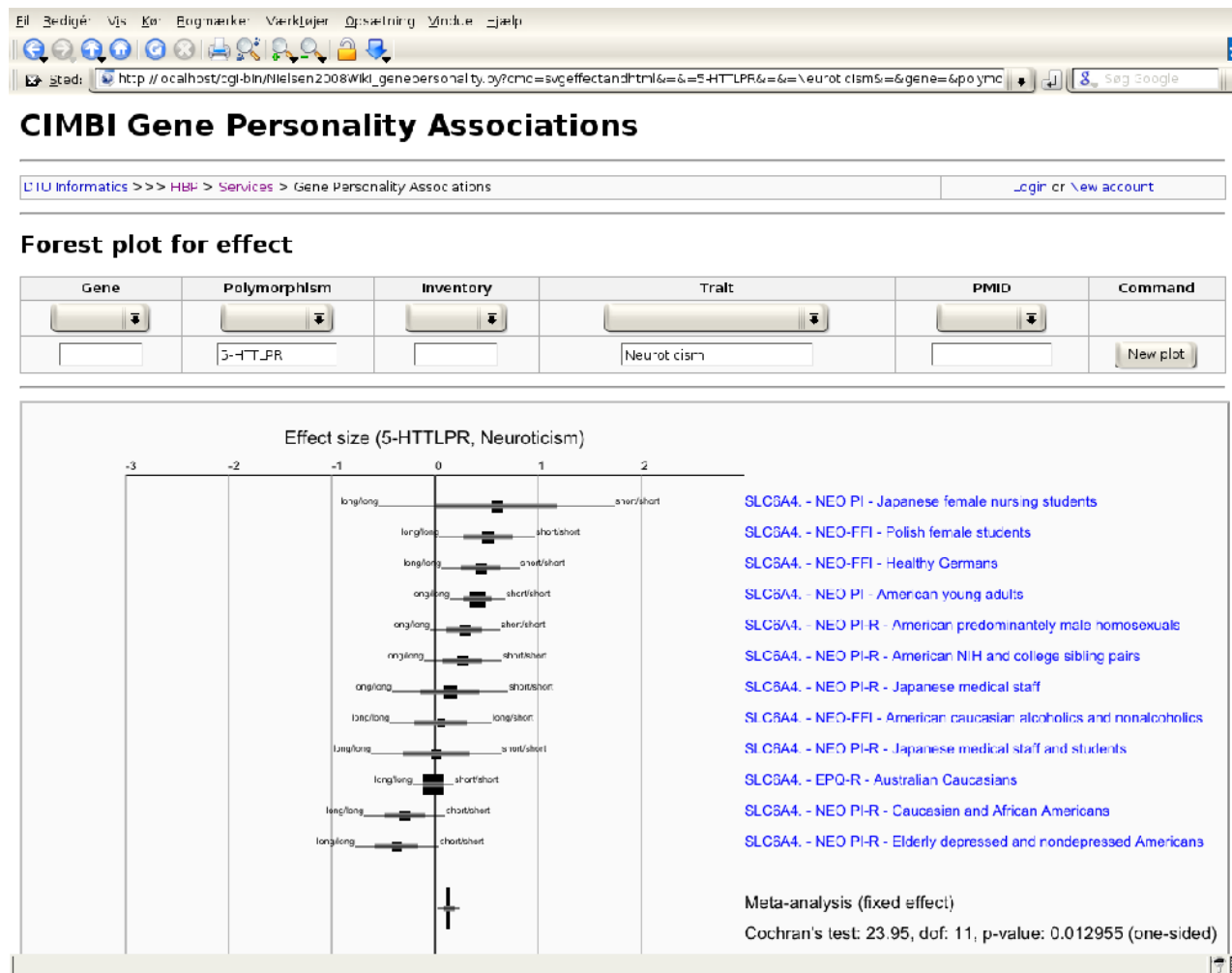
Web script in Python implementing a fielded wiki for personality genetics.

Persistence with a small SQLite database.

Some of the Python libraries used: cgi, Cookie, math, pysqlite2, scipy, sha.

One Python script with 2269 lines of code.

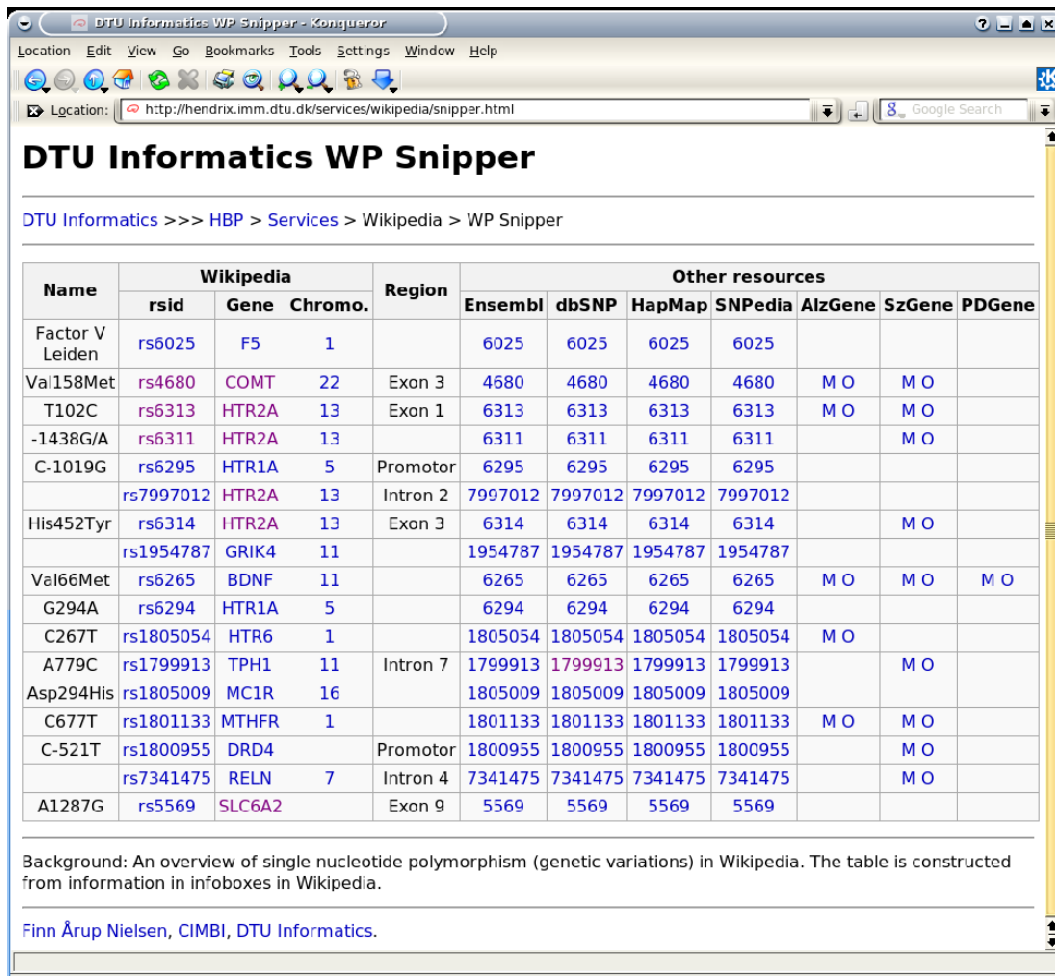
Example: ... A fielded wiki



Computation of *effect sizes* (a statistical value) and comparison to statistical distributions.

Generation of interactive and hyperlinked plots in SVG (an XML format)

Structured information from Wikipedia



Name	Wikipedia			Region	Other resources						
	rsid	Gene	Chromo.		Ensembl	dbSNP	HapMap	SNPedia	AlzGene	SzGene	PDGene
Factor V Leiden	rs6025	F5	1		6025	6025	6025	6025			
Val158Met	rs4680	COMT	22	Exon 3	4680	4680	4680	4680	MO	MO	
T102C	rs6313	HTR2A	13	Exon 1	6313	6313	6313	6313	MO	MO	
-1438G/A	rs6311	HTR2A	13		6311	6311	6311	6311		MO	
C-1019G	rs6295	HTR1A	5	Promotor	6295	6295	6295	6295			
	rs7997012	HTR2A	13	Intron 2	7997012	7997012	7997012	7997012			
His452Tyr	rs6314	HTR2A	13	Exon 3	6314	6314	6314	6314		MO	
	rs1954787	GRIK4	11		1954787	1954787	1954787	1954787			
Val66Met	rs6265	BDNF	11		6265	6265	6265	6265	MO	MO	MO
G294A	rs6294	HTR1A	5		6294	6294	6294	6294			
C267T	rs1805054	HTR6	1		1805054	1805054	1805054	1805054	MO		
A779C	rs1799913	TPH1	11	Intron 7	1799913	1799913	1799913	1799913		MO	
Asp294His	rs1805009	MC1R	16		1805009	1805009	1805009	1805009			
C677T	rs1801133	MTHFR	1		1801133	1801133	1801133	1801133	MO	MO	
C-521T	rs1800955	DRD4		Promotor	1800955	1800955	1800955	1800955		MO	
	rs7341475	RELN	7	Intron 4	7341475	7341475	7341475	7341475		MO	
A1287G	rs5569	SLC6A2		Exon 9	5569	5569	5569	5569			

Background: An overview of single nucleotide polymorphism (genetic variations) in Wikipedia. The table is constructed from information in infoboxes in Wikipedia.

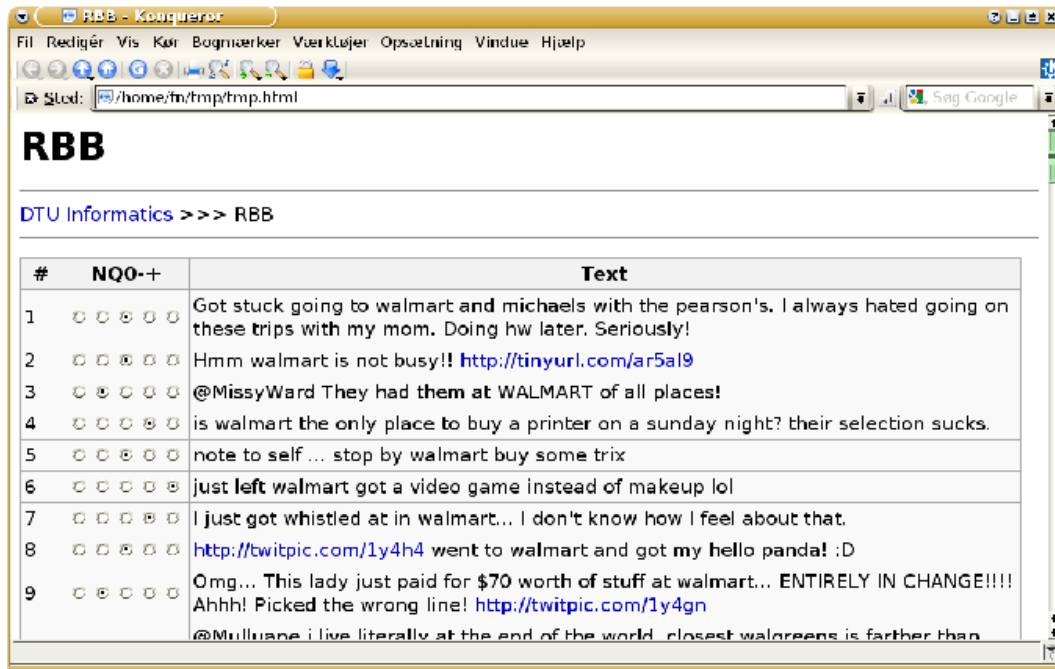
[Finn Årup Nielsen, CIMBI, DTU Informatics.](#)

Get Wikipedia pages that contain a specific template, download the page, extract information from the templates and render the result on an HTML page.

Python libraries: json, re, urllib2

Around 25 Python lines to get the data, and around 120 to render the result.

Web script for Twitter annotation



The screenshot shows a web browser window titled "RBB - Konqueror". The address bar shows a local file path. The page content includes the title "RBB" and a sub-header "DTU Informatics >>> RBB". Below this is a table with two columns: "# NQ0-+" and "Text". The table contains 9 rows of tweets, each with a numerical ID and a list of small circular icons representing annotations. The text of the tweets is visible in the "Text" column.

#	NQ0-+	Text
1	○ ○ ○ ○ ○	Got stuck going to walmart and michael's with the pearson's. I always hated going on these trips with my mom. Doing hw later. Seriously!
2	○ ○ ○ ○ ○	Hmm walmart is not busy!! http://tinyurl.com/ar5a19
3	○ ○ ○ ○ ○	@MissyWard They had them at WALMART of all places!
4	○ ○ ○ ○ ○	is walmart the only place to buy a printer on a sunday night? their selection sucks.
5	○ ○ ○ ○ ○	note to self ... stop by walmart buy some trix
6	○ ○ ○ ○ ○	just left walmart got a video game instead of makeup lol
7	○ ○ ○ ○ ○	I just got whistled at in walmart... I don't know how I feel about that.
8	○ ○ ○ ○ ○	http://twitpic.com/1y4h4 went to walmart and got my hello panda! :D
9	○ ○ ○ ○ ○	Omg... This lady just paid for \$70 worth of stuff at walmart... ENTIRELY IN CHANGE!!!! Ahhh! Picked the wrong line! http://twitpic.com/1y4gn @Mulluane i live literally at the end of the world. closest walgreens is farther than

CGI program that searches Twitter with a user-defined query, obtain tweets and present them in a Web form for manual annotation and stores the result in a SQL database.

Python libraries: codecs, json, re, cgi, urllib2, pysqlite2, xml.

500 Python lines.

Temporal sentiment analysis

DTU-forsker afkoder Twitter-beskeder med 1.200 linjer Python-kode

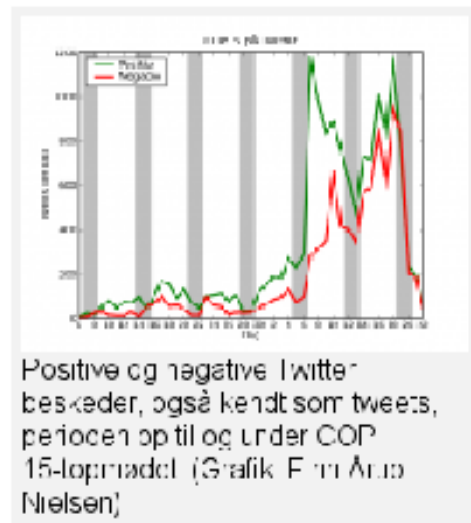
Twitter-beskeder og blog-indlæg har stor betydning for, hvordan virksomheders omdømme ser ud online. Danske forskere arbejder på at skabe et digitalt stemningsbarometer ud fra syndfloden af oplysninger online.

AF: [Mikkel Møller](#), [Torsdag 29. juli 2009 kl. 06:59](#)
 _MNL: [Python](#) [social](#) [SDI](#) [WAR](#)

—vad enten en virksomhed opfører sig socialt ansvarligt eller som en miljøforbryder, skal den nok blive set og hørt via nettes underskov af blogs, wikier og Twitter feeds.

Derfor arbejder et hold af forskere på CBS og DTU på at kunne aflæse virksomheders ry og omdømme allerede fra de bidder af tekst, der skrives om dem for eksempel på blogs og gennem Twitter.

Idéen med denne form for sentimentanalyse, som det hedder – eller stemningsanalyse på modersmålet – er at kunne give et billede af om der skrives godt, skidt eller begge dele om en virksomhed lige på det store, kontrollerbare net.



Download tweets from Twitter microblog searching on 'COP15' (United Nation climate conference in December 2009)

Compare words against a word list with valence (positive/negative) valence for each word.

Sum up positive and negative valence for each day and plot a graph.

Python libraries: SQLite, re, simplejson, ...

Online topic-sentiment mining

http://neuro.imm.dtu.dk/cgi-bin/brede_str_nmf

References

- Beazley, D. M. (2000). *Python Essential Reference*. The New Riders Professional Library. New Riders, Indianapolis.
- Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly, Sebastopol, California. ISBN 9780596516499.
- Campbell, J., Gries, P., Montojo, J., and Wilson, G. (2009). *Practical Programming: An Introduction to Computer Science Using Python*. The Pragmatic Bookshelf, Raleigh.
- Langtangen, H. P. (2005). *Python Scripting for Computational Science*, volume 3 of *Texts in Computational Science and Engineering*. Springer. ISBN 3540294155.
- Langtangen, H. P. (2008). *Python Scripting for Computational Science*, volume 3 of *Texts in Computational Science and Engineering*. Springer, Berlin, third edition edition. ISBN 978-3-642-09315-9.
- Martelli, A., Ravenscroft, A. M., and Ascher, D., editors (2005). *Python Cookbook*. O'Reilly, Sebastopol, California, 2nd edition.
- Model, M. L. (2009). *Bioinformatics Programming Using Python*. O'Reilly, Köln. ISBN 978-0-596-15450-9.
- Pilgrim, M. (2004). *Dive into Python*.
- Russell, M. A. (2011). *Mining the Social Web*. O'Reilly. ISBN 978-1-4493-8834-8.
- Scheible, J. and Tuulos, V. (2007). *Mobile Python: Rapid Prototyping of Applications on the Mobile Platform*. Wiley, 1st edition. ISBN 9780470515051.
- Segaran, T. (2007). *Programming Collective Intelligence*. O'Reilly, Sebastopol, California.
- Segaran, T., Evans, C., and Taylor, J. (2009). *Programming the Semantic Web*. O'Reilly. ISBN 978-0-596-15381-6.
- Sheppard, K. (2014). *Introduction to Python for Econometrics, Statistics and Data Analysis*. Self-published, University of Oxford, version 2.1 edition.