Big Data and Automated Content Analysis

Week 1 – Wednesday

»First steps in the VM«

Damian Trilling

d.c.trilling@uva.nl
@damian0604
www.damiantrilling.net

Afdeling Communicatiewetenschap
Universiteit van Amsterdam

7 February 2018

Today

When point-and-click doesn't help you further:
The Linux command line

The tools

General idea

Whereever possbile, we use tools that are

- platform-independent
- free (as in beer and as in speech)
- open source

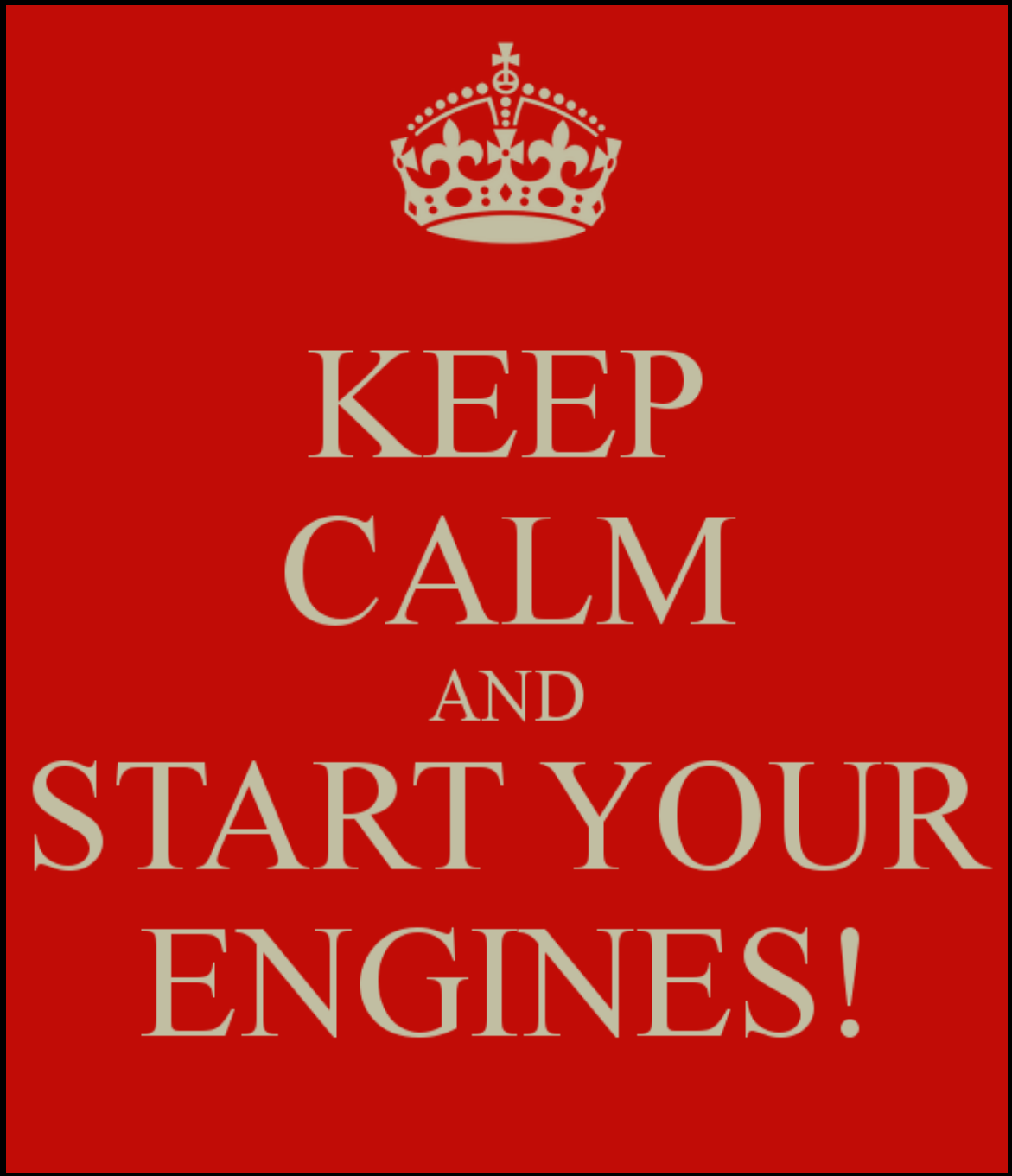
The tools

General idea

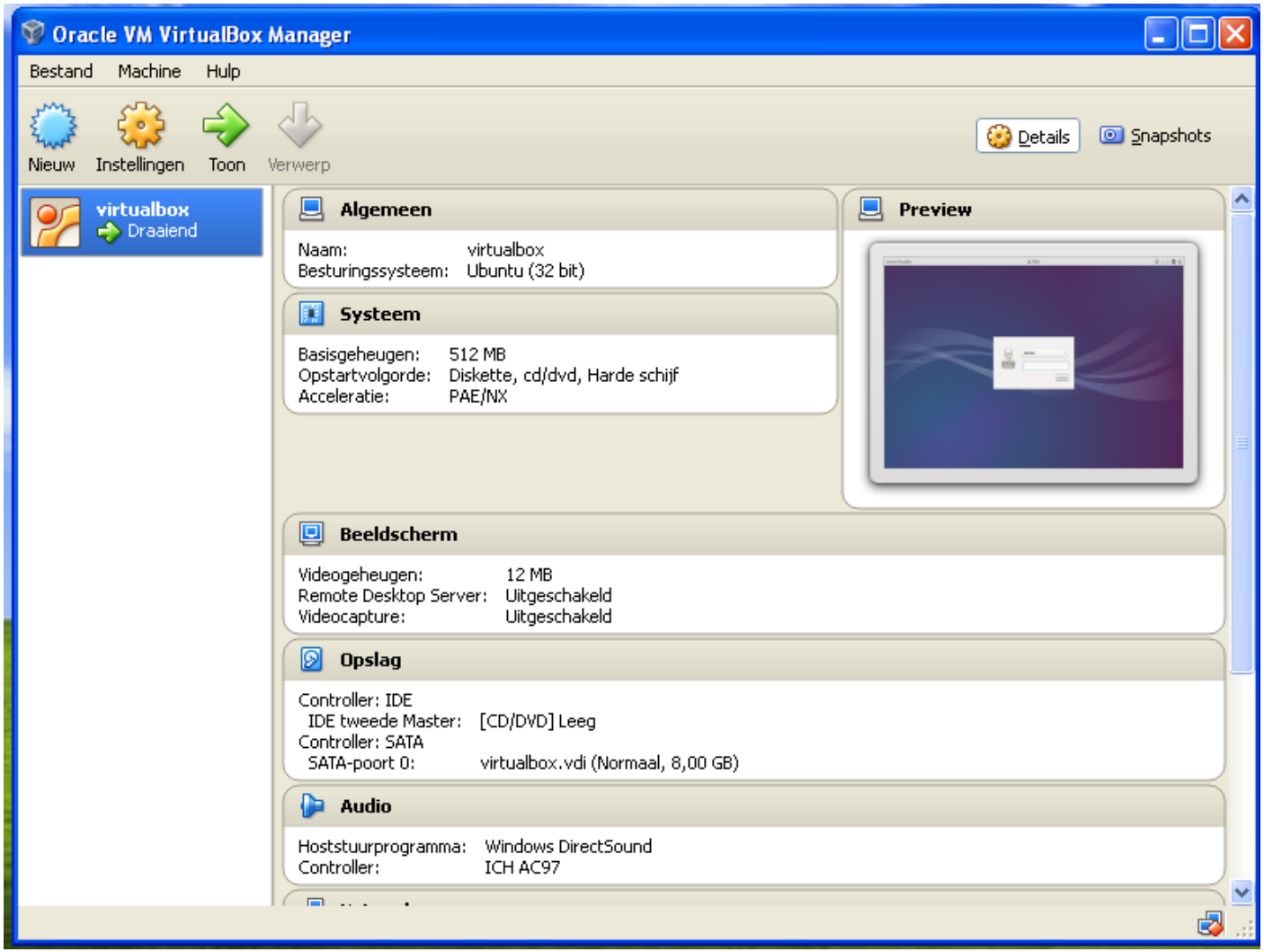
Whereever possbile, we use tools that are

- platform-independent
- free (as in beer and as in speech)
- open source

To make things easier, we work with a virtual machine in which everyone runs *the same* Linux version.

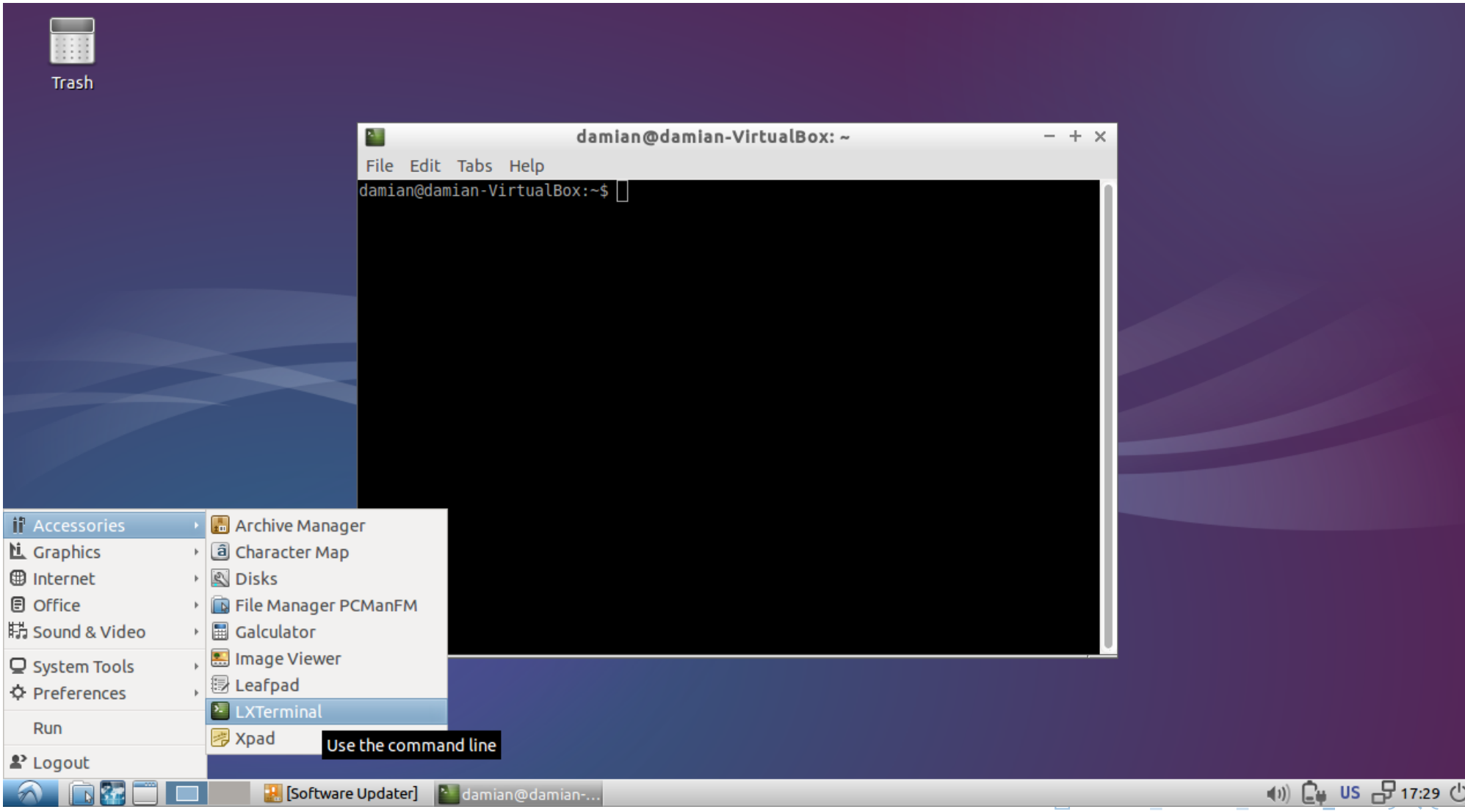

KEEP
CALM
AND
START YOUR
ENGINES!

Let's switch to Linux!



Tools: The linux command line

a.k.a. the `terminal`, `shell` or, more specifically, `bash`



Tools: The linux command line

Tools: The linux command line

Why?

Tools: The linux command line

Why?

- Direct access to your computer's functions

Tools: The linux command line

Why?

- Direct access to your computer's functions
- In contrast to point-and-click programs, command line programs can easily be linked to each other, scripted, ...

Tools: The linux command line

Why?

- Direct access to your computer's functions
- In contrast to point-and-click programs, command line programs can easily be linked to each other, scripted, ...
- Suitable for handling even huge files

Tools: The linux command line

Why?

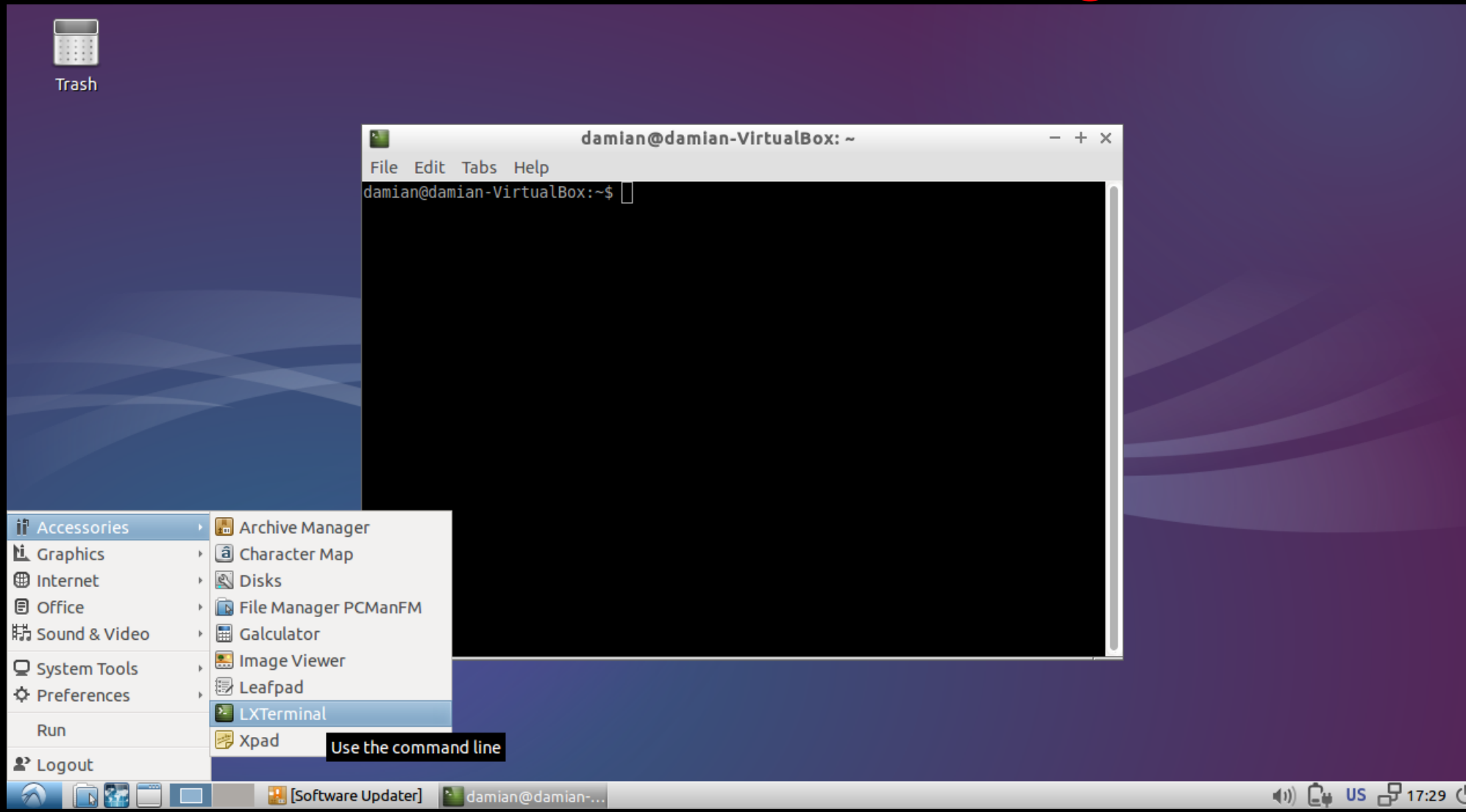
- Direct access to your computer's functions
- In contrast to point-and-click programs, command line programs can easily be linked to each other, scripted, ...
- Suitable for handling even huge files
  - You simply cannot open them in many GUI programs
  - ...or it takes ages
  - The command line allows you to do such things without problems

Tools: The linux command line

Why?

- Direct access to your computer's functions
- In contrast to point-and-click programs, command line programs can easily be linked to each other, scripted, ...
- Suitable for handling even huge files
  - You simply cannot open them in many GUI programs
  - ...or it takes ages
  - The command line allows you to do such things without problems
- It is reproducible (ever tried to explain to your parents on the phone where they have to click?)

There are endless tutorials, cheat sheets, videos . . . online. Google it!



Exercise

Take the book.
Follow the instructions in Chapter 2.
Observe how you could do the same thing with the graphical interface.

A language, not a program:
Python

Python

What?

- A language, not a specific program
- Huge advantage: flexibility, portability
- One of *the* languages for data analysis. (The other one is R.)

Python

What?

- A language, not a specific program
- Huge advantage: flexibility, portability
- One of *the* languages for data analysis. (The other one is R.)

Python

What?

- A language, not a specific program
- Huge advantage: flexibility, portability
- One of *the* languages for data analysis. (The other one is R.)

Which version?

We use Python 3.
`http://www.google.com` or `http://www.stackexchange.com` still offer a lot of Python2-code, but that can easily be adapted. Most notable difference: In Python 2, you write `print "Hi"`, this has changed to `print ("Hi")`

If it's not a program, how do you work with it?

Interactive mode

- Just type `ipython3` (if not available: `python3`)on the command line, and you can start entering Python commands

  (You can leave again by entering `quit()`)
- Great for quick try-outs, but you cannot even save your code

If it's not a program, how do you work with it?

Interactive mode

- Just type `ipython3` (if not available: `python3`)on the command line, and you can start entering Python commands

  (You can leave again by entering `quit()`)
- Great for quick try-outs, but you cannot even save your code

An editor of your choice

- Write your program in any text editor, save it as `myprog.py`
- and run it from the command line with `./myprog.py` or `python3 myprog.py`

If it's not a program, how do you start it?

An IDE (Integrated Development Environment)

- Provides an interface
- Both quick interactive try-outs and writing larger programs
- We use spyder, which looks a bit like RStudio (and to some extent like Stata)

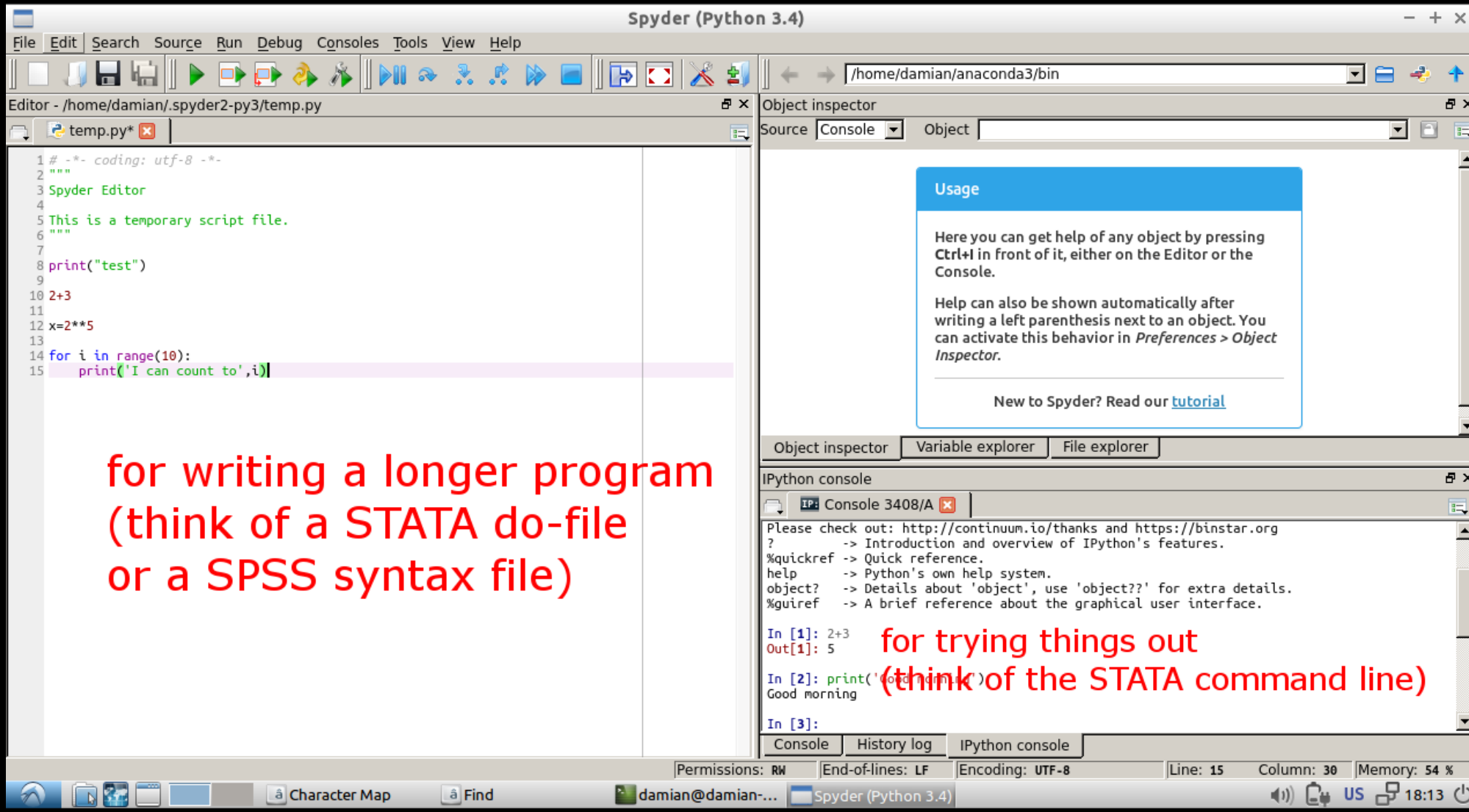
If it's not a program, how do you start it?

An IDE (Integrated Development Environment)

- Provides an interface
- Both quick interactive try-outs and writing larger programs
- We use spyder, which looks a bit like RStudio (and to some extent like Stata)

Jupyter Notebook

- Runs in your browser
- Stores results and text along with code
- Great for *interactive* playing with data and for sharing results


Spyder (Python 3.4)
File Edit Search Source Run Debug Consoles Tools View Help
/home/damian/anaconda3/bin
Editor - /home/damian/.spyder2-py3/temp.py
temp.py*
-*- coding: utf-8 -*-
"""
Spyder Editor
This is a temporary script file.
"""
print("test")
2+3
x=2**5
for i in range(10):
print('I can count to',i)
for writing a longer program
(think of a STATA do-file
or a SPSS syntax file)
Object inspector
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Object Inspector.
New to Spyder? Read our tutorial
Object inspector Variable explorer File explorer
IPython console
Console 3408/A
Please check out: http://continuum.io/thanks and https://binstar.org
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%guiref -> A brief reference about the graphical user interface.
In [1]: 2+3
Out[1]: 5
In [2]: print('Good morning')
Good morning
In [3]:
for trying things out
(think of the STATA command line)
Console History log IPython console
Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 15 Column: 30 Memory: 54 %
Character Map Find damian@damian-... Spyder (Python 3.4)
US 18:13

ACA-workshop_day1 - Mozilla Firefox

ACA-workshop_day1

localhost:8888/notebooks/ACA-workshop_day1.ipynb

csv writer dialect excel

jupyter ACA-workshop_day1 Last Checkpoint: Last Monday at 2:01 PM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help

Python 3

Code CellToolbar

In [ ]:
```
import csv
import re
from nltk.sentiment import vader
from nltk.corpus import stopwords
import nltk
```

Download the data

We will use a dataset by Schumacher et al. (2016). From the abstract:

> This paper presents EUSpeech, a new dataset of 18,403 speeches from EU leaders (i.e., heads of government in 10 member states, EU commissioners, party leaders in the European Parliament, and ECB and IMF leaders) from 2007 to 2015. These speeches vary in sentiment, topics and ideology, allowing for fine-grained, over-time comparison of representation in the EU. The member states we included are Czech Republic, France, Germany, Greece, Netherlands, Italy, Spain, United Kingdom, Poland and Portugal.

Schumacher, G, Schoonvelde, M., Dahiya, T., Traber, D, & de Vries, E. (2016): *EUSpeech: a New Dataset of EU Elite Speeches.* doi:10.7910/DVN/XPCVEI

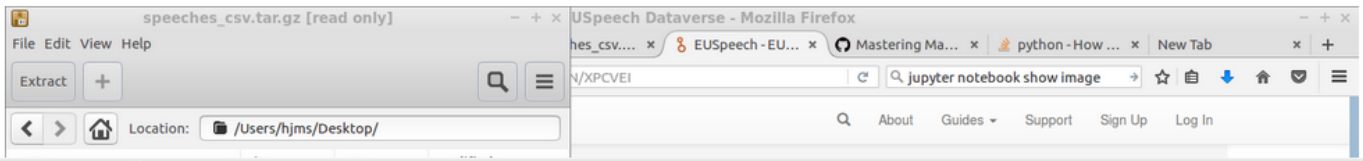
Download and unpack the following file:

```
speeches_csv.tar.gz
```

In the .tar.gz file, you find a .zip file. Extract the whole folder to your home directory. See below a screenshot of how this looks like in Lubuntu (double-click on "speeches_csv.zip" in the left window, then the right window will open. Click on "Extract")

In [1]:
```
from IPython.display import Image
Image("https://github.com/damian0604/bdaca/raw/master/ipynb/euspeech_download.png")
```

Out[1]:



damian@dami... [Spyder (Pytho... damian@dami... ACA-workshop... neg 16:46

Let's start up a Python environment and write a Hello-world-program!

Start playing!

1. Run a program that greets you.

The code for this is

```
1 print("Hello world")
```

After that, do some calculations. You can do that in a similar way:

```
1 a=2
2 print(a*3)
```

Just play around.

Repeat your exercise in different environments (command line shell, spyder, jupyter notebook).

Start playing!

2. Write a program that converts centimeters to inches

Take a variable with the number of centimeters as input and print a nicely formatted answer that gives the value in inches. 1 inch = 2.54 cm.

Printing several things in a row can be done like this:

```
1 print("The answer is",42)
```

Additional ressources

Codecademy course on Python

`https://www.codecademy.com/learn/python`

Next meetings

Week 2: Getting started with Python

Monday, 12–2

Lecture.
Chapter 4.

Wednesday, 14–2

Lab Session.
Exercise A1.