

# An introduction to agent-based modeling with Python

Installation guideline for the required software

Claudius Gräbner<sup>1,2</sup>

<sup>1</sup>Johannes Kepler University

<sup>1</sup>ZOE. Institute for Future-Fit Economies

Version 1.1. of July 23, 2018

For the course, the following software is required:

1. A python3 engine
2. The python modules numpy, scipy and matplotlib, and
3. An editor or IDE
4. The Jupyter package

These packages can be installed separately or as an integrated distribution. I detail now which options are available for the most common operating systems; all the software packages we recommend are free and open source. There are also proprietary options, but I do not cover them in this manual.

It is absolutely essential that the software is running the first day of the course. If you have problems with the installation, please contact me before the workshop! I will be more than happy to help you out!

[Below](#) I will also provide you with a test script. Please follow the instructions below to test whether your installation works properly and let me know if not.

# Contents

<b>1. Install Python and the required packages</b>	<b>3</b>
1.1. Python 2 or Python 3?	3
1.2. For Mac OS X and macOS	3
1.2.1. Install Anaconda	3
1.2.2. Install packages individually	3
1.3. For Windows	4
1.3.1. Install Anaconda	4
1.3.2. Install packages individually	4
1.4. For Linux	4
1.4.1. Individual Packages (python engine and libraries)	4
1.4.2. As a Distribution	5
<b>2. Install a suitable editor or IDE</b>	<b>5</b>
2.1. Windows and macOS	5
2.2. Linux	5
2.2.1. Debian/Ubuntu	6
2.2.2. Fedora	6
2.2.3. Archlinux	6
2.2.4. Gentoo	6
2.2.5. Other	6
<b>3. Install Jupyter</b>	<b>7</b>
3.1. Windows and macOS	7
3.2. Linux	7
<b>4. Run the test script</b>	<b>7</b>
<b>A. Installing Anaconda on Windows: Detailed Guideline</b>	<b>8</b>

# 1. Install Python and the required packages

## 1.1. Python 2 or Python 3?

There are currently two versions of Python: the ‘older’ version 2.X, and the newer version 3.X. Initially, the 2.X version should have been discarded by 2015, but it still enjoys wide popularity, especially among companies. Yet, Python 3.X has improved the language in many ways, and since support of the Python developers for Python 2.X will probably run out in 2020, there are a number of good reasons to learn Python 3.X. And this is what we are going to do in this course.

For more information on the differences between the two versions, and a guide to choose the right one see, for example, [the official website of the Python developers](#) or this useful [blog](#).

Of course, there are also very critical remarks on Python 3.X. If you are interested you can delve into the discussion by this [critical post](#) and this [rebuttal](#). But as indicated above, this course will be dedicated to Python 3.X, and not Python 2.X.

## 1.2. For Mac OS X and macOS

Mac typically already includes a Python engine and interpreter (cPython). However, this will be a version 2.X, not 3.X. As the syntax is different, and for the reasons provided [above](#), I strongly recommend to install a Python 3.X version alongside Python 2.X.

More precisely, I suggest installing the Anaconda package. It already includes all the libraries you will need for scientific computing. This facilitates things, particularly for beginners. But of course, it is also possible to install all packages individually.

### 1.2.1. Install Anaconda

Installing Anaconda on a Mac is *really* *reals*: just visit the following website and follow the [installation instructions](#). Please make sure that you have Python3, and not Python2!

### 1.2.2. Install packages individually

First, make sure you have the package manager MacPorts installed. If not, just visit this website and follow the instructions: <https://www.macports.org/install.php>

Then follow these steps to install Python3:

1. Open a terminal
2. Run

```
sudo port install py35-numpy py35-scipy py35-matplotlib
```

Note that 35 indicates version number 3.5 (i.e. Python 3.5). You may have to change this number (3.6, 3.7 would be more recent) but it should be consistent for all packages. Numpy, Scipy, and Matplotlib are the packages you need to install.

## 1.3. For Windows

### 1.3.1. Install Anaconda

Short story: visit the following website and follow the [installation instructions](#). Please make sure that you have Python3, and not Python2! You may also use another distribution such as [WinPython](#).

Long story: since installing Anaconda on a Windows machine is not as straightforward as for a MAC, I provide you with a detailed step-by-step installation guideline [below](#).

### 1.3.2. Install packages individually

Unfortunately, Windows does not have a package manager. This makes it difficult to install packages individually so that they match and work well together. Installing a distribution is, therefore, the preferred option in the vast majority of cases.

If you want to install individual packages anyway, you need to download the python engine from the [python website](#).

The other packages are offered as binary on, for instance, [Christoph Grohlke's website](#).

Please be very careful which websites you trust and which websites you download software from.

## 1.4. For Linux

Many Linux distributions come with the python engine/interpreter (cPython) installed. It varies, however, which version (2.X or 3.X) it is. All packages are preferably installed using the distribution's proper package manager since it will then be included in future system updates. All common package managers offer the packages required for this course.

### 1.4.1. Individual Packages (python engine and libraries)

**Debian/Ubuntu type distributions** The standard python distribution is typically Python 2.X, hence you have to install Python 3.X additionally:

```
apt-get install python3 python3-numpy python3-scipy python3-matplotlib
```

**Fedora like distributions** Recent versions of Fedora have the python3 engine pre-installed as /usr/bin/python (If for some reason not, do "sudo yum install python"). The modules are available by:

```
sudo yum install python3-numpy python3-scipy python3-matplotlib
```

**Archlinux like distributions** Python 3.X is the standard python engine of the distribution. The modules can be installed like this:

```
pacman -Sv python-numpy, python-scipy, python-matplotlib
```

**Gentoo like distributions** Make sure that the appropriate version of Python 3.X is in in your USE FLAGS or defined in your portage make.conf (see [here](#)). Then:

```
emerge -pv python numpy scipy matplotlib
```

```
emerge -v python numpy scipy matplotlib
```

**Others** It is impossible to list instructions for all distributions. Most package managers feature the required packages. Otherwise, it is still possible to install modules using python's own pip package manager. Note that e.g. on older versions of Redhat/Fedora/CentOS, python3 (and associated modules) are not part of the official repositories. You may need to compile and install them manually. There are [manuals for this](#).

### 1.4.2. As a Distribution

You may install python as a distribution. This comes with the disadvantage that you will have several installations on your system for many packages. This effectively reverses one of the advantages of fully open and integrated systems. It does, however, have the advantage that you can install it completely in userspace, i.e. without root privileges.

Anaconda is an example, [see the website](#).

## 2. Install a suitable editor or IDE

Just select and install a suitable editor or IDE. There are many options. You may find a list of possible choices [here \(IDEs\)](#) or [here \(editors\)](#).

Some options I could recommend include

- [Spyder](#)
- [Geany](#)
- [Emacs](#)

Personally, I use Emacs and Spyder. During the course, I will rely exclusively on Spyder, which comes directly with Anaconda. It is much more accessible for beginners than Emacs, although the latter is much cooler;)

### 2.1. Windows and macOS

Installation packages can usually be downloaded directly from the editor/IDE websites. Note, however, that at least for Spyder the installation will become tricky if you do not install Anaconda (see [here](#)). And Emacs can be installed on macOS via the following terminal command (provided you use the macOS package manager MacPorts):

```
sudo port install emacs
```

### 2.2. Linux

Linux users may start with the following package manager commands, but keep in mind that some further system configuration may become necessary:

### 2.2.1. Debian/Ubuntu

```
apt-get install python3-spyder
```

```
apt-get install geany
```

```
apt-get install emacs
```

...

### 2.2.2. Fedora

```
yum install python3-spyder
```

```
yum install geany
```

```
yum install emacs
```

...

### 2.2.3. Archlinux

```
pacman -Sv spyder
```

```
pacman -Sv geany
```

```
pacman -Sv emacs
```

...

### 2.2.4. Gentoo

```
emerge -pv spyder
```

```
emerge -v spyder
```

```
emerge -pv geany
```

```
emerge -v geany
```

```
emerge -pv emacs
```

```
emerge -v emacs
```

You may need to set `ACCEPT_KEYWORDS` appropriately for portage to allow the use of packages in testing.

### 2.2.5. Other

The availability and maintenance of IDEs and editor packages differs between distributions. Check out what is available for your distribution. Some editors can also be installed with pip; jupyter is an example, see [here](#).

### 3. Install Jupyter

You need to make sure that jupyter is installed. In case you installed Python3 via Anaconda, Jupyter is already installed.

In case you did not installed Anaconda you need to install Jupyter via your package manager (Linux) or via the Python package manager pip (macOS and Windows). You may test whether everything is working fine by starting a notebook according to: [this manual](#).

Let me know if there are any error messages!

#### 3.1. Windows and macOS

Jupyter must be installed via the python package manager pip. It comes with all new python versions. To install jupyter via pip just follow the instructions [here](#).

#### 3.2. Linux

Linux users may install jupyter via their respective package managers, for Ubuntu/Debian-like distributions, for example, this would be:

```
apt-get install python3-jupyter
```

For for Arch-like distributions it would be:

```
pacman -Sv jupyter
```

And for Gentoo-like distributions:

```
emerge -pv jupyter
```

```
emerge -v jupyter
```

Please check out tutorials in the WWW for your particular distribution.

### 4. Run the test script

Please download the test script from the course page. Alternatively, you can get it [from my homepage](#).

Save the file in a folder and execute it using Python. To to this, you use a terminal to get to the location of the file and type:

```
python test-complexity.py
```

If you get the message that everything is all right, great! You are well prepared for the course.

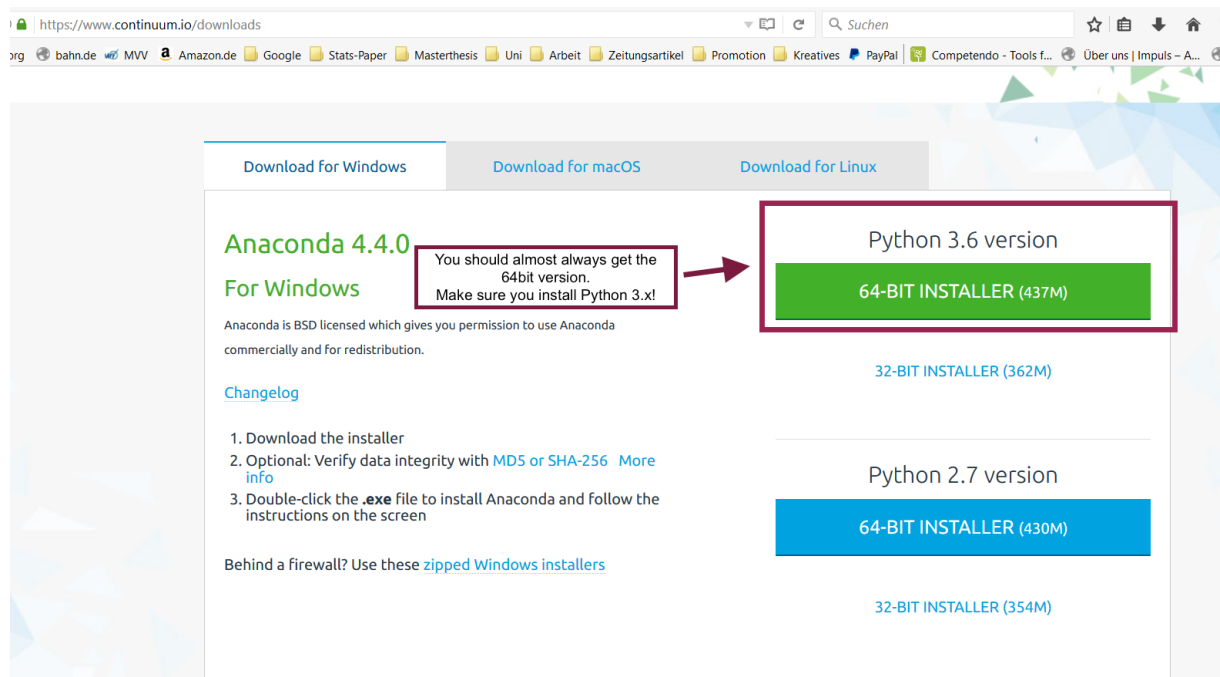
If not, please send the error message via email to [Claudius](#).

## A. Installing Anaconda on Windows: Detailed Guideline

This is a more detailed installation guideline for Anaconda for Windows 10. It probably works for Windows 8 in the same way, but no guarantee for that. Please let me know if there are any problems.

First thing is to get a suitable terminal emulator for Windows, e.g. [cmdr](#).

Then, download the Anaconda installer from [this page](#). Make sure you install Python 3.X., as shown in screenshot 1.



Screenshot 1: Make sure to install Python 3.x.

We strongly recommend to check the integrity of the downloaded file via the MH5 sums! A useful and easy-to-use tool is [this one](#).

Now you are ready to install Anaconda!

Thus, double click on the downloaded .exe file and install Anaconda with the default settings (only for the current user as as the standard Python version). In case you are already more experienced, you may also change the default options, but for beginners we recommend the default settings.

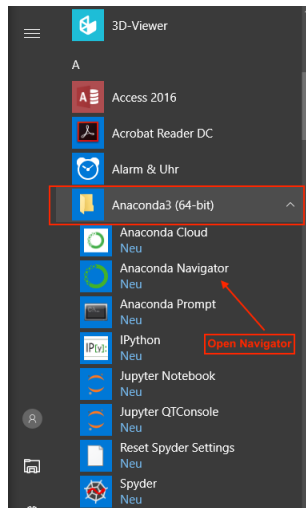
If everything went well you should have the Anaconda folder in your star menu as shown in screenshot 2.

After clicking on “Anaconda Navigator” you should see the main navigation window as shown in screenshot 3.

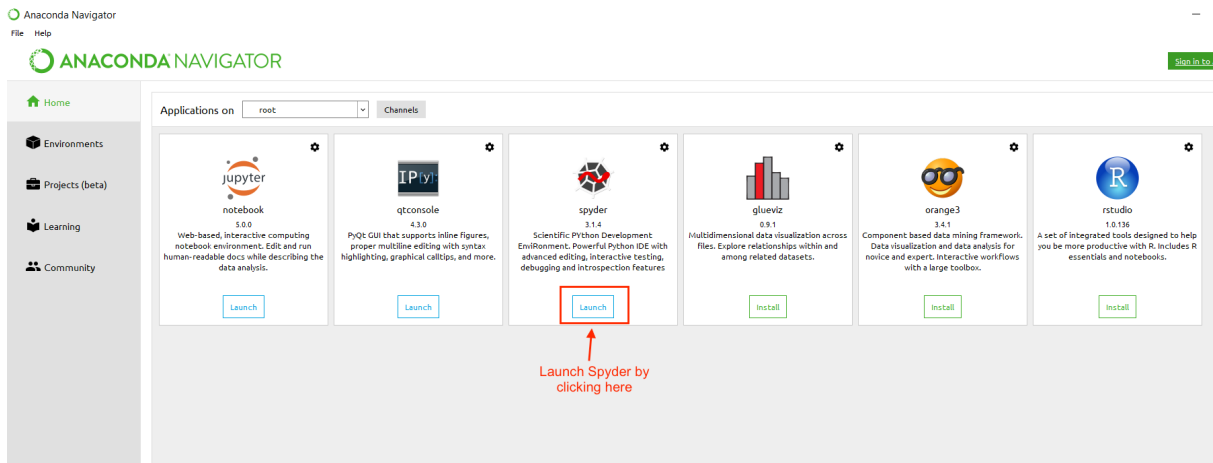
Here you can choose which application to open. We will have a closer look at the applications in the course, but the most relevant ones for us are:

- The Jupyter notebooks, which we use to experiment with code and write annotated scripts.
- Spyder, a integrated development environment (IDE), which we use to develop code and execute it.





Screenshot 2: Anaconda should now be in your start menu.



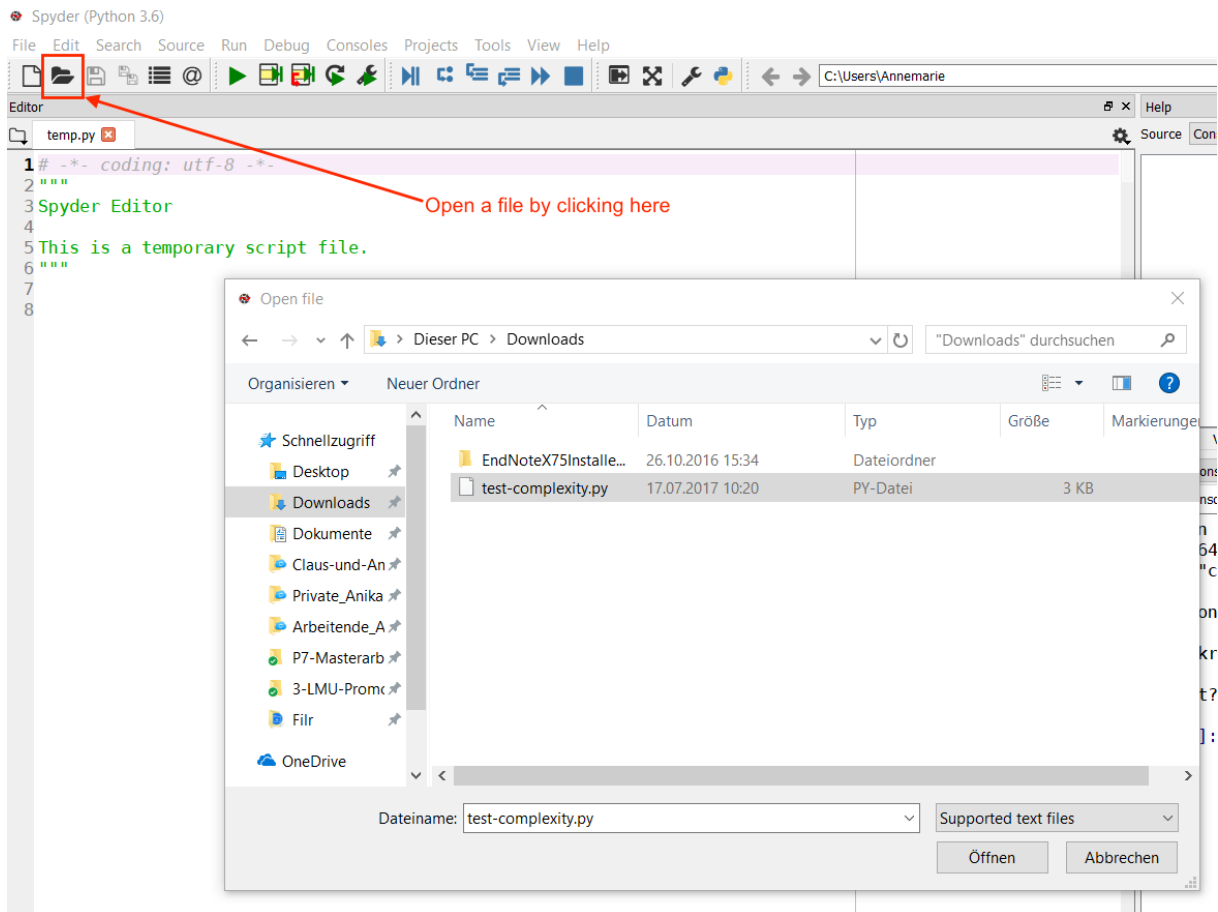
Screenshot 3: The navigation window of Anaconda Navigator.

To check whether everything works fine, open Spyder. Within Spyder open the test file, which you can get from the course page. Alternatively, you can get it [from my homepage](#). You can open the file as indicated in screenshot [4](#).

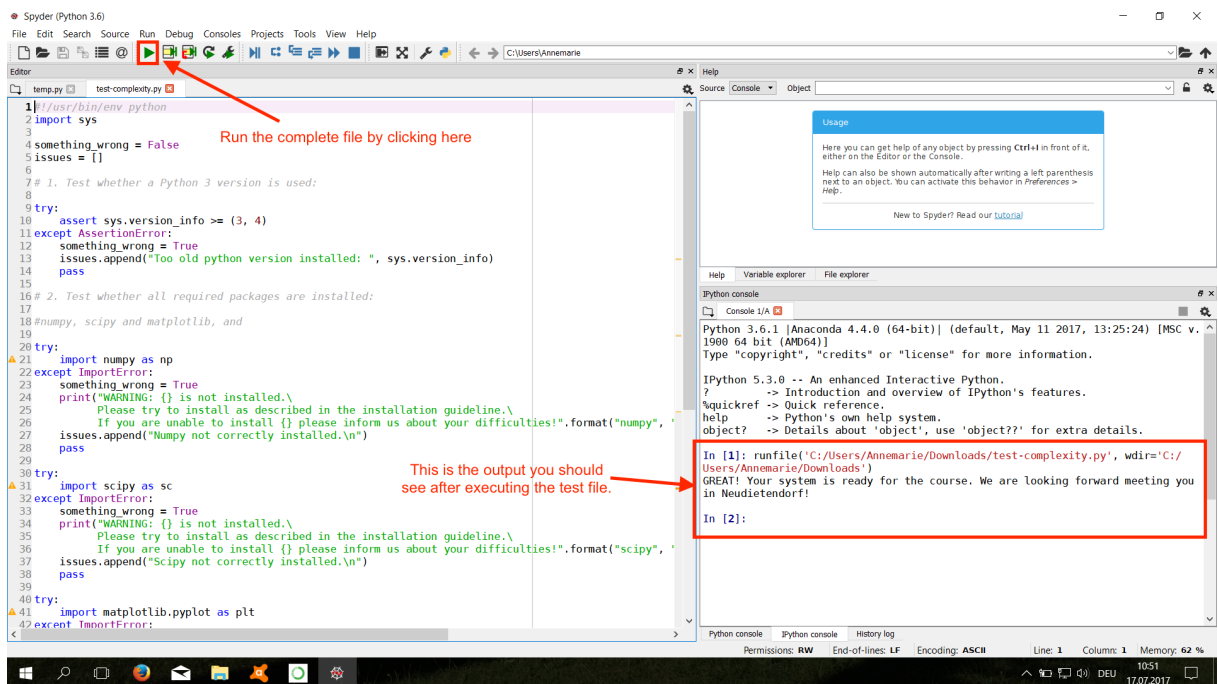
Then execute it by clicking on the button “Run file”, as indicated in screenshot [5](#).

If a window pops up that lets you choose your Python installation, choose the 3.x one (which should be default) and click run.

If you see the message as shown in the bottom right in screenshot [5](#) everything is working fine! If not, please contact Claudius!



Screenshot 4: Open the test file from within the Spyder IDE.



Screenshot 5: Open the test file from within the Spyder IDE.