

Jon Danielsson
London School of Economics

Financial Risk Forecasting

Seminar Getting Started with Python, Anaconda and
Positron

financialriskforecasting.com

August 2026

Version 5.0

© 2011–2026 Jon Danielsson · CC BY-NC-ND 4.0

Introduction

The empirical work in this course will be done in Python. You may use any recent Python distribution and user interface, but at LSE we use the Anaconda distribution with Positron.

We discuss how to use AI assistance for course work in [The AI assistant](#) below.

From Week 4, you may need to produce reports in PDF format. [Making PDFs](#) explains the additional software required.

1 Using an LSE computer

1.1 Start a Python session with Positron and Anaconda

Everything is already installed on an LSE computer:

1. Open Positron.
2. Open the folder containing your work using **File > Open Folder**.
3. Click **Start Session** in the top-right corner.
4. Under **Start New Console Session**, select the Python entry labelled **Conda: base**.

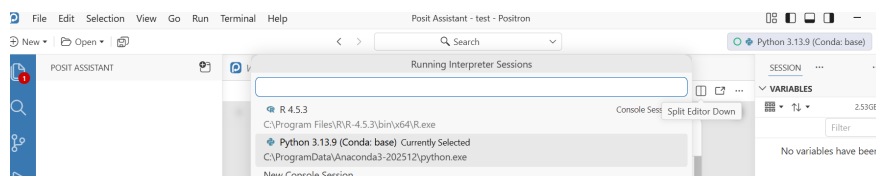


Figure 1: The Anaconda Python selected in Positron on Windows

Do not select one of the other Python installations listed by Positron. Positron remembers this choice for the folder, so you should only need to make it once.

Next, [install the two additional libraries](#), then [check that the setup works](#). These sections apply to both LSE computers and your own computer.

2 Installing on your own computer

If you want to install the course setup on your own computer, complete the three installation steps below: install Anaconda, install Positron and tell Positron to use Anaconda. Then [install the two additional libraries](#) and [check that the setup works](#).

Anaconda supplies Python and the libraries used for numerical work, statistics and plotting. Positron is the application in which you write and run the code. Both are free and run on macOS, Windows and Linux.

The [LSE Digital Skills Lab](#) runs free workshops, online courses and drop-in sessions on Python. Take software problems there. Questions about the course material belong in the seminars.

2.1 Install Anaconda

Download it from anaconda.com/download and run the installer. The download is around 1 GB, and it needs about 5 GB of disk once installed.

Choose the option to install for your user only. It needs no administrator password and is the more reliable of the two choices.

2.2 Install Positron

Download it from positron.posit.co/download and install it as you would any other application.

2.3 Tell Positron to use Anaconda

Positron does not run your code itself. It starts a session — a running copy of Python, waiting for something to do — and sends your code to it. The session is what actually executes anything, and it decides which Python is used and therefore which libraries are available.

That matters because a machine usually carries several Pythons. The operating system has one of its own, other software install others, and Anaconda adds another. Positron finds all of them and, left to itself, starts the newest, which is not necessarily the one you want. Anaconda's is the one with the libraries this course needs.

There are two ways to make sure it is the one used, and either is enough.

2.3.1 Set it as the default

The thorough way, done once and applying everywhere afterwards.

Open the settings editor: **Positron > Settings > Settings** on macOS, **File > Preferences > Settings** on Windows, or **Cmd-**, and **Ctrl-**, respectively. Search for `python.defaultInterpreterPath`, and put the location of Anaconda's Python in the box. Every folder you open from then on starts with Anaconda unless you say otherwise.

The difficulty is knowing what to type, because the location depends on how Anaconda was installed. On macOS it is usually `/opt/anaconda3/bin/python`, and on Windows usually `C:\Users\yourname\anaconda3\python.exe`, but an installation made for all users, or put somewhere other than the default, sits elsewhere.

Rather than guess, read it off your own machine. Do the second method below once, and the path appears next to the Conda entry in the list — that string, exactly as shown, is what belongs in the setting. Alternatively, run

```
import sys
sys.executable
```

in any session that is already using Anaconda, and it prints the path.

2.3.2 Or choose it for the folder you work in

The simpler way, and the one to use if the paths above do not match your machine.

Make a new, empty folder for your work this term. On macOS, `Documents/FinancialRiskForecasting` is a sensible choice. On Windows, `C:\Users\yourname\FinancialRiskForecasting`.

Start Positron. With nothing open it shows a welcome screen, and the Explorer on the left says that no folder is open.

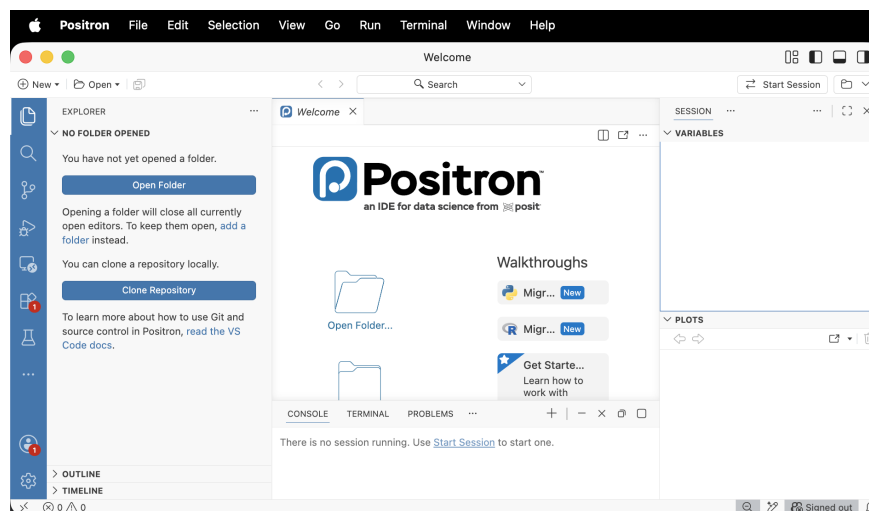


Figure 2: Positron on starting, with no folder open

Use **File > Open Folder**, or the Open Folder button in the Explorer, and select the folder you just made. Open the folder, not a file: Positron works on a folder, everything you write this term lives there, and every data file your code reads or writes is understood relative to it.

Nothing runs until a session is started, which is why the console says there is none. Click **Start Session** at the top right of the window. Positron then lists every Python it found. The one you want is labelled Conda, and its path contains `anaconda3`. Typing `con` into the box narrows the list to it.

Positron remembers this for the folder, so the choice is made once rather than weekly. The Python in use is named at the top right whenever a session is running, and it is worth a glance at the start of each seminar, since almost everything that goes wrong later comes back to it.

If a notification offers to change `terminal.integrated.inheritEnv`, accept it. It is standard advice for conda environments and affects only the terminal. The terminal is not used for ordinary Python work in this course, although it is used occasionally for software setup such as installing TinyTeX.

3 Install the two additional libraries

Anaconda includes almost all the libraries used in this course. With the Conda session running, the simplest method is to enter the following command in the Console pane:

```
%pip install arch eodhd
```

You can instead install `arch` and `eodhd` separately through Positron's Packages pane. Whichever method you use, make sure the active Python session is the Anaconda one. You only need to install the libraries once.

Installing into the Anaconda base installation is a simplification for a course that adds two libraries and never changes them. Work outside a course normally gives each project its own environment, so that one project's libraries cannot disturb another's.

4 Check that it works

With the Anaconda session running, enter the following in the Console pane.

```
import sys
import numpy, pandas, matplotlib, scipy, statsmodels, tabulate, arch, eodhd

print("Running from:", sys.executable)
```

If it raises no error and prints the path to the selected Anaconda Python, the setup is complete. That path is the important result. It proves that the same Anaconda Python received the installations and ran the course imports.

If it stops with `ModuleNotFoundError`, check the interpreter name at the top right, change it to Anaconda, and repeat the installation in the same session before running the check again.

5 Making PDFs

Nothing above is needed to run code, produce plots or write notebooks. It becomes relevant from Week 4, when the course starts producing reports, and for assignments submitted as PDF.

Quarto is the tool that turns a document into HTML, Word or PDF, and it comes with Positron, so there is nothing to install for it. HTML and Word need nothing further. PDF is the exception. It is produced through LaTeX, a separate typesetting system that Quarto calls out to.

If you already have a LaTeX distribution — MacTeX on macOS, MiKTeX or TeX Live on Windows — Quarto finds and uses it, and you need do nothing. These are large installations, several gigabytes, and are usually there because somebody installed them deliberately.

If you do not, Quarto can install a small one for itself. In Positron's terminal, run once:

```
quarto install tinytex
```

TinyTeX is about a 100 MB download rather than several gigabytes, carrying the couple of hundred LaTeX packages that documents usually need and fetching anything further as it meets it. It is installed locally, used only by Quarto, and does not disturb anything else on the machine.

Note that this does not happen by itself. Positron includes Quarto, but Quarto includes no LaTeX and will not install one for you. Without either a distribution already present or that one command, rendering to PDF stops with an error.

You will know which case you are in the first time you render a PDF. It either works, or Quarto stops and tells you that no LaTeX installation was found.

6 The AI assistant

Positron has an AI assistant built in. It is optional, it does nothing until you connect it to a provider, and the course does not require it.

What it offers is a chat panel beside your code, which can explain what a piece of code does, say what an error message means, and write code from a description of what you want. It will also suggest completions as you type, and, if you let it, edit files directly. Browser-based AI tools can do much the same when code is pasted into them. Both are genuinely useful and both produce plausible answers that can be wrong.

To connect one, run **Authentication: Configure Language Model Providers** from the command palette: **Cmd-Shift-P** on macOS, **Ctrl-Shift-P** on Windows.

Some providers offer student access, while others require a paid developer account. A consumer chat subscription and developer access are often separate products, so check the provider's current terms before attaching a payment method. No particular provider is required for this course.

Whatever you connect, your code and your questions go to that provider. Its privacy and charging terms apply. Do not place private data, passwords or API tokens in an AI conversation. An assistant allowed to inspect or edit open files may also receive code in those files, so remove secrets before giving it that access.

7 The libraries

For reference, this is what the course uses and where each is introduced. Five come with Anaconda, and the setup instructions add the other two.

Library	Used for	First used	In Anaconda
numpy	arrays and numerical calculation	Week 1	yes

Library	Used for	First used	In Anaconda
pandas	data frames, reading and writing data files	Week 1	yes
matplotlib	plots	Week 1	yes
eodhd	downloading market data from EOD Historical Data	Week 1	no
scipy	distributions, optimisation and matrix routines	Week 3	yes
statsmodels	statistical tests and autocorrelation plots	Week 3	yes
arch	GARCH and other volatility models	Week 6	no

8 When something goes wrong

Start with the final line of the error message. It names the type of error and usually identifies the library, file, variable or line of code involved.

8.1 Check the Python session

Most setup problems occur because Positron is running a different Python. Check the session name at the top right. It should identify **Conda: base**. If it does not, follow [Start a Python session with Positron and Anaconda](#), then repeat [Check that it works](#).

8.2 A library cannot be found

`ModuleNotFoundError` means that the active Python cannot find a named library. First check that the session is **Conda: base**. If the missing library is **arch** or **eodhd**, follow [Install the two additional libraries](#), then run the setup check again.

8.3 A file, variable or column cannot be found

- **FileNotFoundError**: check the filename and confirm that the folder containing your work and data files is open in Positron.
- **NameError**: the named variable does not exist in the current session. Run the earlier code that creates it.
- **KeyError**: the requested column or ticker is not present. Check its spelling and inspect the available column names.

If the state of the session is unclear, restart it and run the necessary code again from the beginning.

8.4 Python cannot read the code

SyntaxError means that Python cannot read the code as written. Check the indicated line and the line immediately above it for missing brackets, quotation marks or punctuation. If the error followed a recent edit, undo that edit and run the code again.

8.5 Anaconda is not listed

Run **Interpreter: Discover All Interpreters** from the command palette — **Cmd-Shift-P** on macOS or **Ctrl-Shift-P** on Windows — and look again.

On an LSE computer, do not attempt to reinstall Anaconda. Restart Positron and, if Anaconda is still absent, take the problem to the [LSE Digital Skills Lab](#). On your own computer, confirm that Anaconda is installed and reinstall it in the default location if it is not.

8.6 The console will not start

If the console closes as soon as it opens, restart Positron once and select **Conda: base** again. If the problem repeats on an LSE computer, use another LSE computer and report the faulty installation to the LSE Digital Skills Lab. On your own computer, reinstall Anaconda rather than selecting an unrelated Python installation.