

Inhaltsverzeichnis

Inhaltsverzeichnis	I
4 Entwicklungsumgebung	2
4.1 Anaconda Navigator	7
4.2 Jupyter Notebook Editor	7
4.3 Spyder Editor	13
4.4 R	22
4.5 Rtools	26
4.6 RStudio	30
4.7 Pakete in R installieren	38
4.8 Pakete in Python installieren	41
4.9 Python in Anaconda Prompt ausführen	44
4.10 Python als Executable in Windows Prompt ausführen	47
4.11 Python in RStudio ausführen	49
4.12 Paket pythonforbusiness	51
Literaturverzeichnis	54

4 Entwicklungsumgebung

Wie eine Entwicklungsumgebung gestaltet wird, hängt wesentlich davon ab, welche Werkzeuge Verwendung finden sollen. Im Folgenden wird vorgestellt, wie die Installation erfolgt, wenn Python (Python Software Foundation, 2025) in Verbindung mit der Integrated Development Environment (IDE) von **Spyder** (Spyder, 2024) und auch **RStudio Desktop** (Posit, 2025) genutzt werden sollen. Um auch Funktionen und Daten aus R-Paketen nutzen zu können, sind auch **R** (R Foundation, 2025) und **R-Tools** (R CRAN Team, 2024) Bestandteil der Entwicklungsumgebung.

Besonders geschickt ist die Installation der obigen Werkzeuge mit Hilfe des Distributionspakets **Anaconda**, welche die Programmiersprachen Python und R, die Entwicklungsumgebung Spyder und die Webanwendung Jupyter Notebook enthält. Das Anaconda Distributionspaket kann unter folgender URL <https://www.anaconda.com/download> (Anaconda, 2025) heruntergeladen werden (siehe Abb. 4-1).

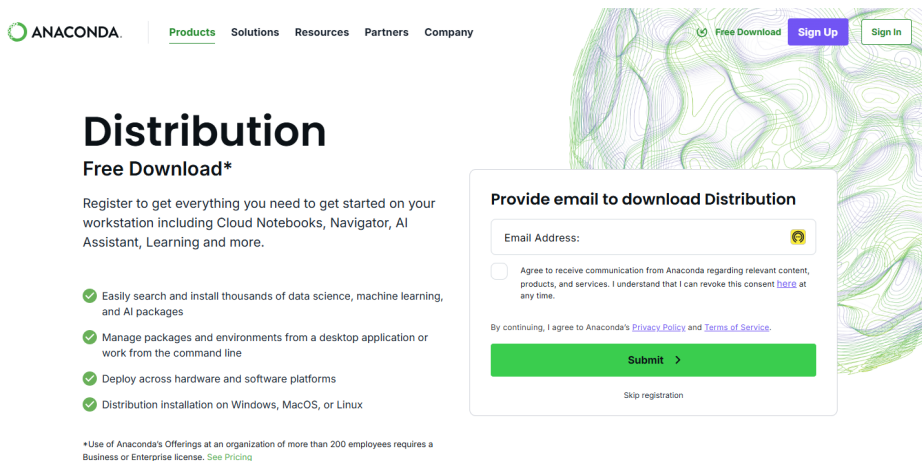


Abb. 4-1: Download

Wenn Sie auf „Download“ klicken können Sie die Installationsdatei speichern oder auch direkt ausführen. Bestätigen Sie, dass Sie die Installation ausführen möchten (siehe Abb. 4-2).



Abb. 4-2: Setup

Klicken Sie auf „Next“, um das Setup auszuführen. Stimmen Sie den Lizenzbedingungen zu, wählen das Installationsverzeichnis aus und registrieren Sie Anaconda3 als Ihre Standard-Python Umgebung (siehe Abb. 4-3).

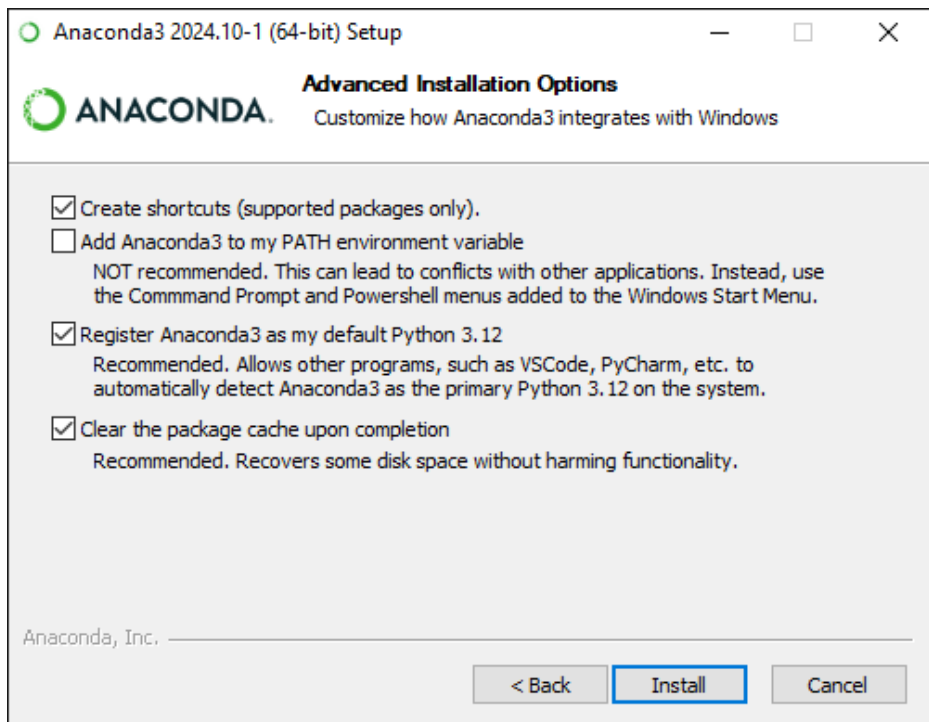


Abb. 4-3: Installationsoptionen

Nach der Installation wird diese als „completed“ bestätigt. Eine weitere Bestätigung wird angezeigt, nachdem Sie auf „Next“ geklickt haben (siehe Abb. 4-4).



Abb. 4-4: Bestätigung

Klicken Sie auf „Next“, um fortzufahren. Auf dem nächsten Fenster können Sie die Webseite mit „Getting Started“ öffnen. Dies können Sie bestätigen und fortfahren (siehe Abb. 4-5).

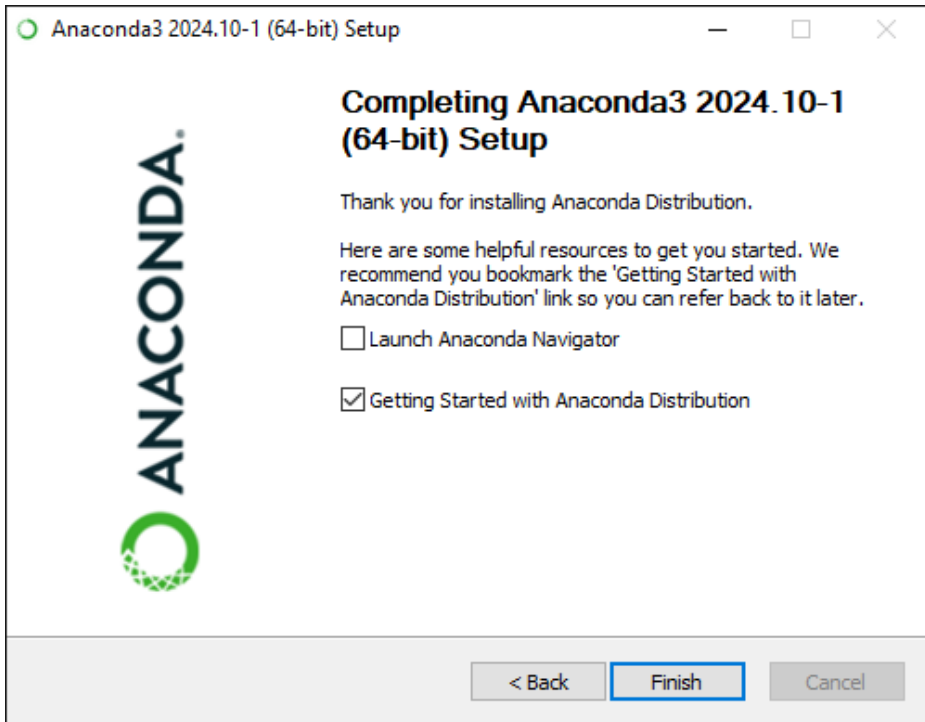


Abb. 4-5: Getting started

Auf der angezeigten Webseite finden Sie Tutorials und die Dokumentation zu Anaconda. Sie können jetzt alle Browser-Fenster schließen. Wenn Sie auf die Windows-Taste klicken, werden Ihnen die neu installierten Applikationen angezeigt: Anaconda Navigator, Anaconda Prompt, Jupyter Notebook und Spyder (siehe Abb. 4-6).

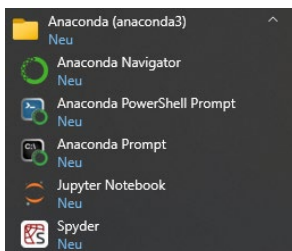


Abb. 4-6: Programme aufrufbar

4.1 Anaconda Navigator

Starten Sie auch den Anaconda Navigator über die Windows-Schaltfläche (siehe Abb. 4-8). Sie könnten hier noch weitere Programme installieren. Dies ist jedoch nicht erforderlich und die Version von RStudio, die hier zur Installation angeboten wird, ist oft eine ältere Version. Verlassen Sie den Navigator daher wieder.

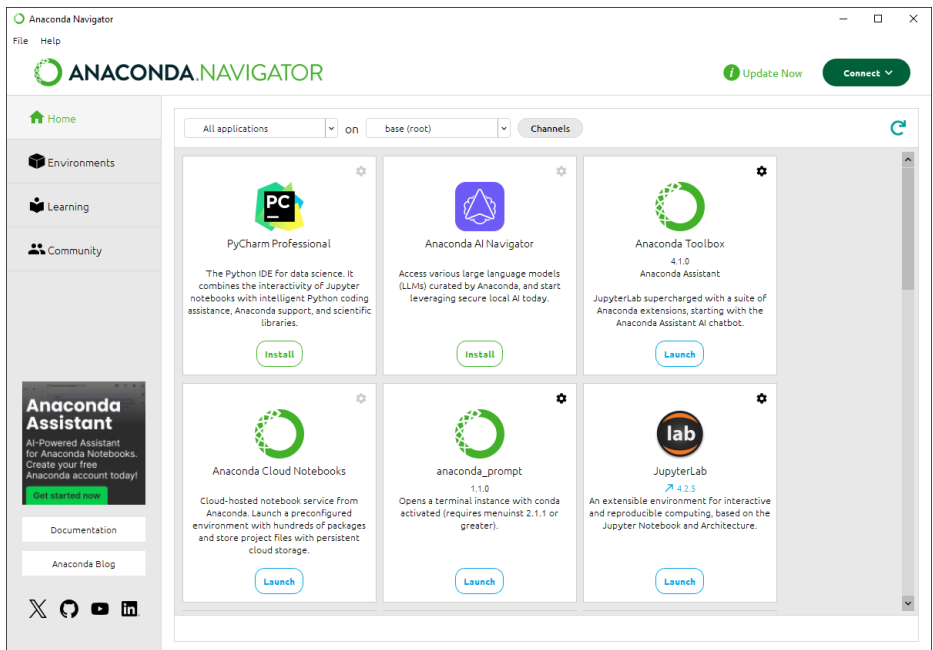
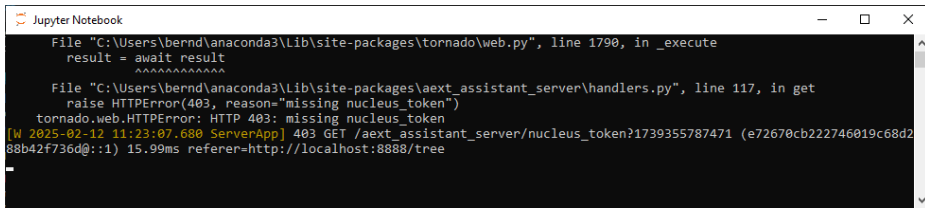


Abb. 4-7: Anaconda Navigator

Führen Sie jetzt einen Neustart Ihres Rechners aus, um u.a. die PATH-Einstellungen wirksam zu machen.

4.2 Jupyter Notebook Editor

Nach dem Restart starten Sie das Jupyter Notebook über die Windows-Schaltfläche. Ein Logfile wird angezeigt (siehe Abb. 4-8).



```
File "C:\Users\bernd\anaconda3\Lib\site-packages\tornado\web.py", line 1790, in _execute
    result = await result
    ^^^^^^^^^^^^^^^
File "C:\Users\bernd\anaconda3\Lib\site-packages\aeht_assistant_server\handlers.py", line 117, in get
    raise HTTPError(403, reason="missing nucleus_token")
tornado.web.HTTPError: HTTP 403: missing nucleus_token
[W 2025-02-12 11:23:07.680 ServerApp] 403 GET /aeht_assistant_server/nucleus_token?1739355787471 (e72670cb222746019c68d288b42f736d@::1) 15.99ms referer=http://localhost:8888/tree
```

Abb. 4-8: Log von Jupyter Notebook

Anschließend öffnet sich ein Fenster, in dem Ihnen die Verzeichnisstruktur Ihres Rechners angezeigt wird (siehe Abb. 4-9).

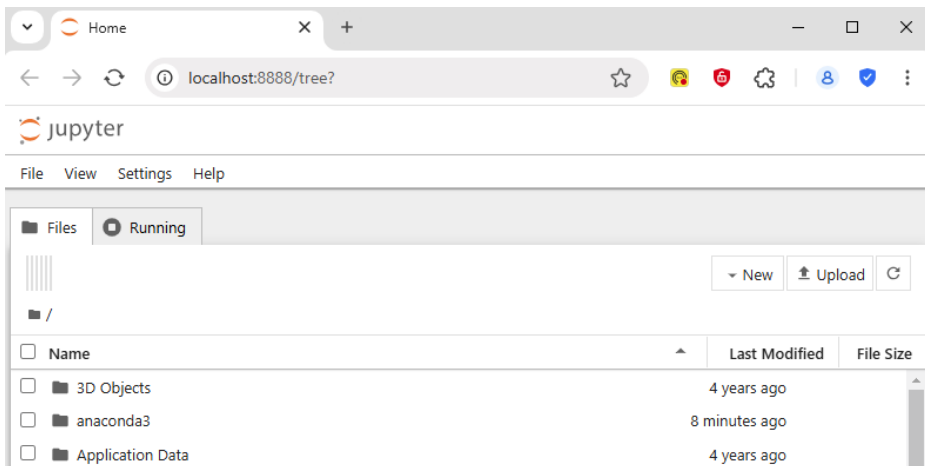


Abb. 4-9: Jupyter Notebook

Klicken Sie oben rechts im Menü auf „File/New/Notebook“. So entsteht ein neues Notebook für Python-Entwicklungen. Da es möglich ist mit verschiedenen Programmiersprachen und Versionen zu arbeiten, gilt es diese zunächst auszuwählen und als sogenannten Kernel für diese Datei zuzuordnen. Bestätigen Sie Python als Kernel mit „Select“ (siehe Abb. 4-10).

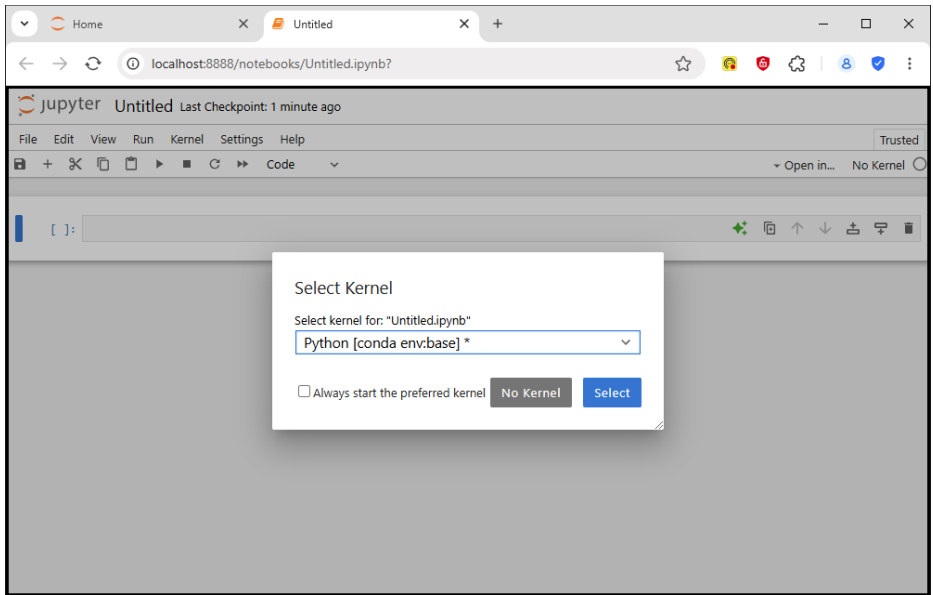


Abb. 4-10: Neues Notebook für Python

Jetzt können Sie im Jupyter Notebook Ihren Python-Code erfassen. Geben Sie `print("Hochschule Ansbach")` ein und führen es aus, indem Sie auf den Button „Run“ klicken (siehe Abb. 4-11).

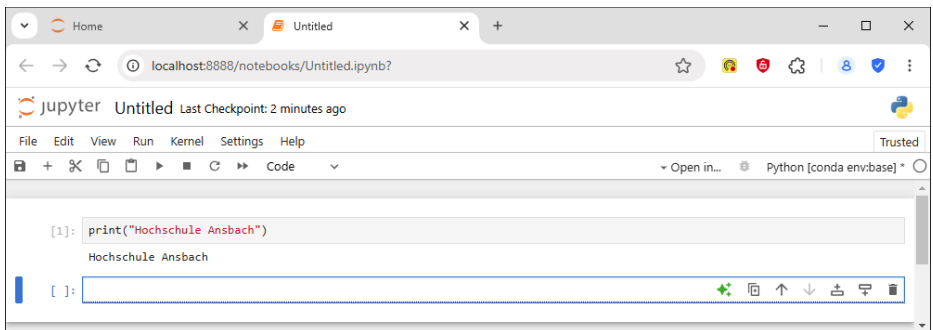


Abb. 4-11: Python-Code ausführen

Klicken Sie im Menü „File“ auf „Save Notebook as“ und speichern Sie Ihr erstes Programm ab unter dem Namen „0-ErstesPythonProgramm“ (siehe Abb. 4-12). Die Dateiergung von Jupyter Notebook ist „.ipynb“.

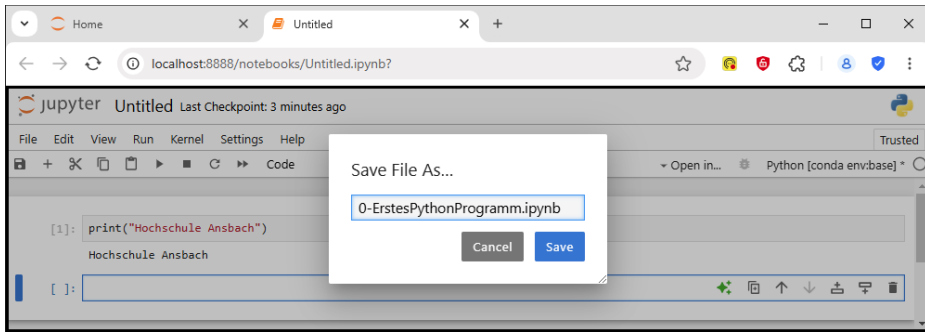


Abb. 4-12: Python-Code speichern

In einem Notebook kann auch ausführbarer Code (Code-Zelle) mit Text (Markdown-Zelle) kombiniert werden. Ob es sich um Markdown oder Code handelt, wird in der Befehlszeile im Dropdown ausgewählt. Überschriften-Ebenen lassen sich in Markdown über die Anzahl der vorangestellten „#“-Zeichen festlegen. Ergänzen Sie Markdown-Zellen mit Hilfe des „+“ vor und nach der Print-Anweisung und speichern Sie das Notebook mit „Save Notebook as“ unter dem Namen „1-ErstesPythonProgramm“ ab (siehe Abb. 4-13 und Abb. 4-14).

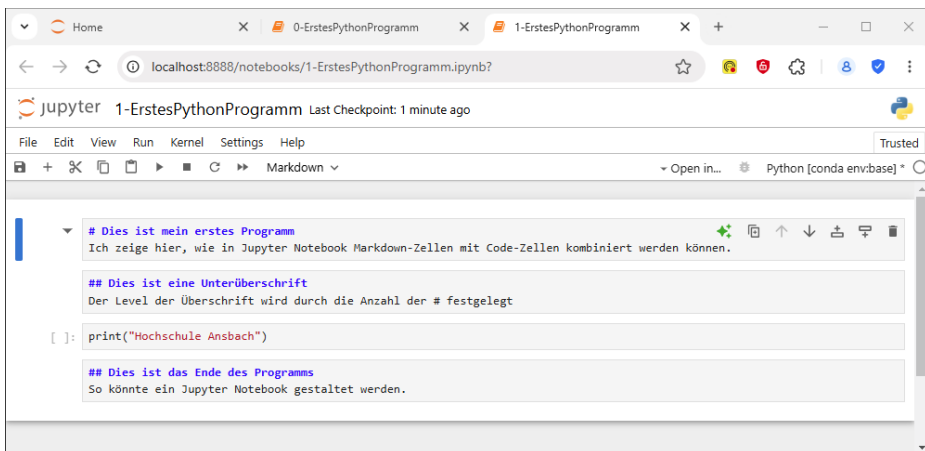


Abb. 4-13: Python Notebook mit Markdown vor Ausführung

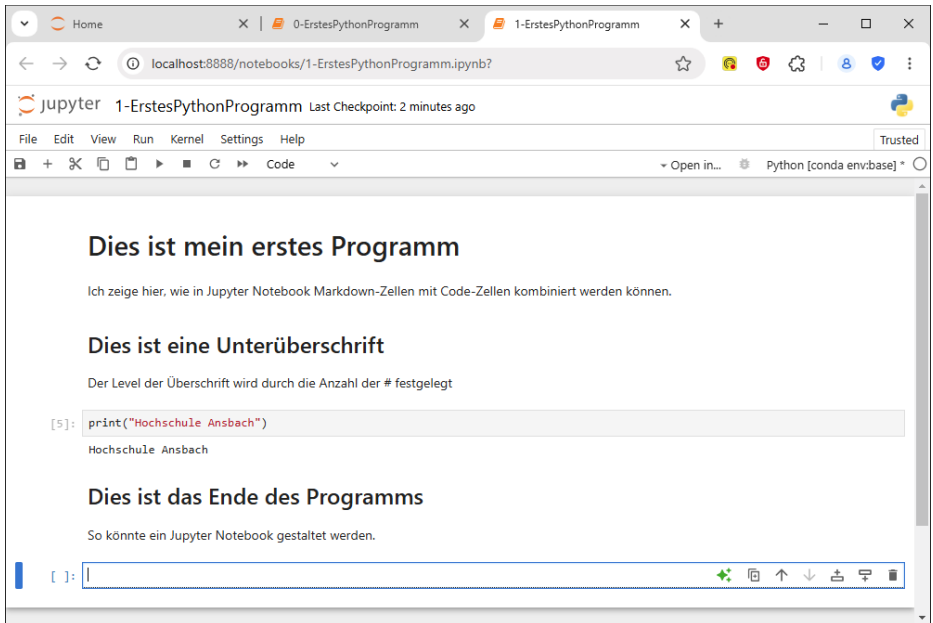


Abb. 4-14: Python Notebook mit Markdown nach Ausführung

Um das Verzeichnis zu modifizieren, welches von Jupyter Notebook beim Start geöffnet werden soll, legen Sie sich zunächst einen Shortcut der Anwendung auf den Desktop. Über das Kontextmenü des Shortcuts können Sie anschließend die Eigenschaften zur Angabe von dem „Ziel“ modifizieren (siehe Abb. 4-15).

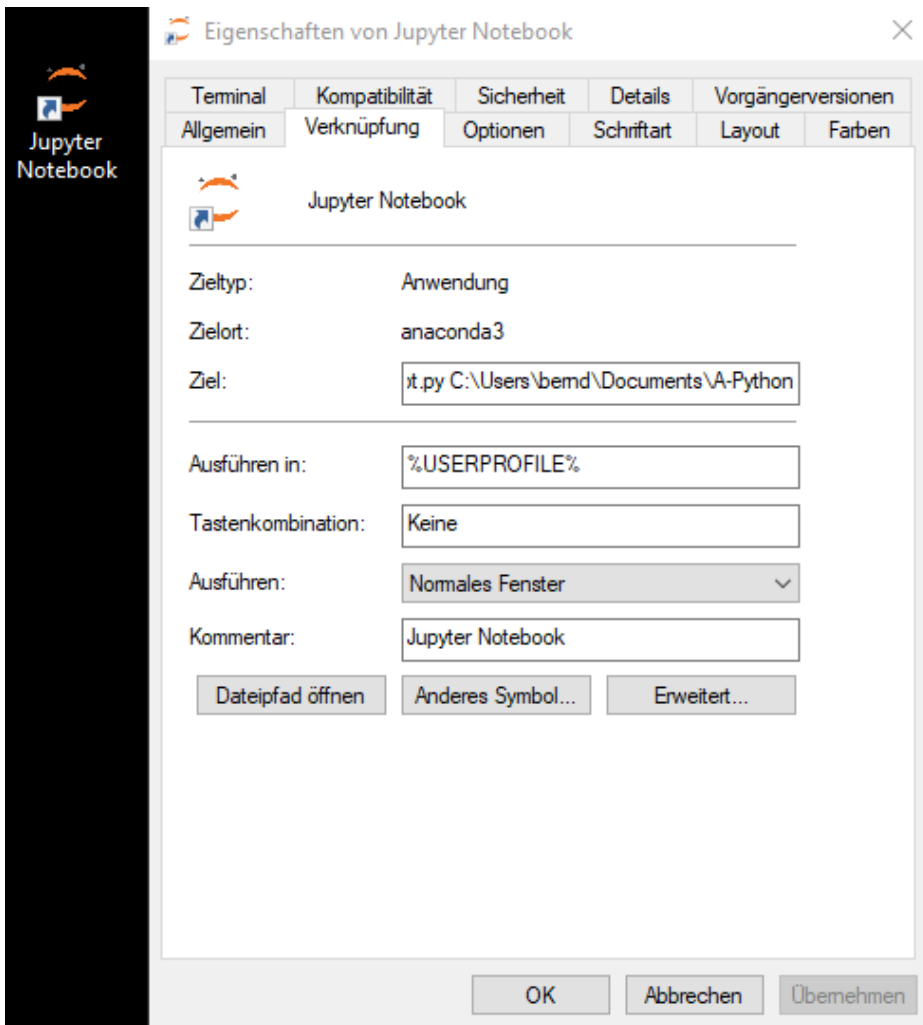


Abb. 4-15: Standardverzeichnis anpassen

Als integrierte Entwicklungsumgebung wird nachfolgend jedoch mit Spyder gearbeitet, da Spyder einige Vorteile bietet.

4.3 Spyder Editor

Klicken Sie auf Spyder, um den Python Editor zu öffnen (siehe Abb. 4-16). Wenn Ihnen angezeigt wird, dass eine neue Version von Spyder existiert, folgen Sie den vorgeschlagenen Installationsanweisungen.

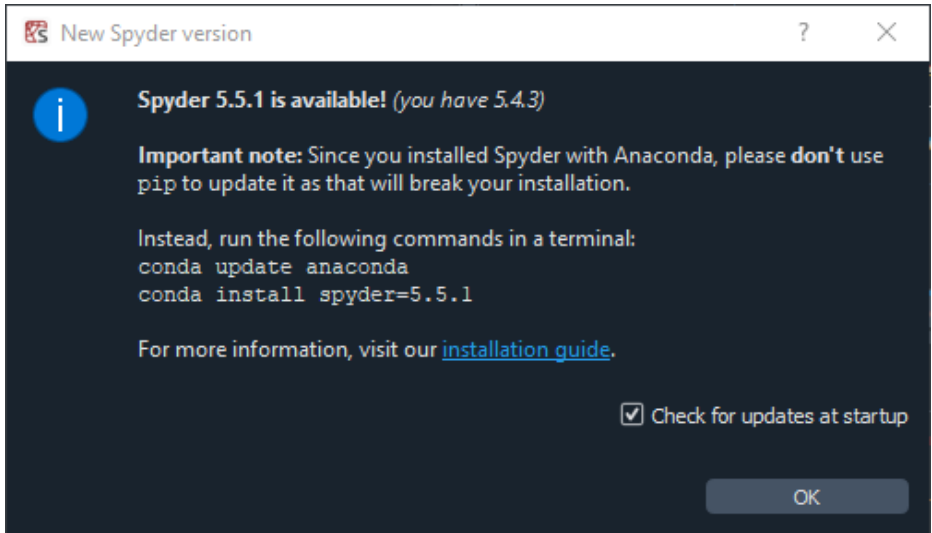
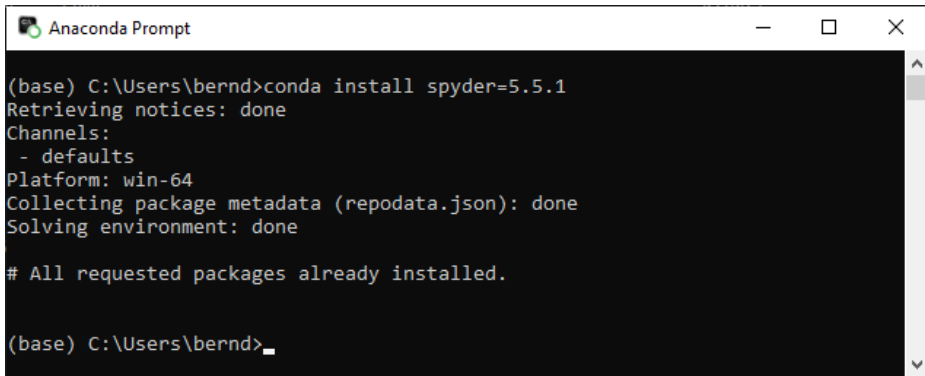


Abb. 4-16: Neue Spyder Version verfügbar

Um den Befehl `conda install spyder=5.5.1` im Terminal auszuführen, öffnen Sie den Anaconda Prompt (gerade unter neu installierten Applikationen angezeigt) und führen den Befehl aus (siehe Abb. 4-17). Auf Nachfrage bestätigen Sie ggfs., dass ein Update erfolgen soll.



```
(base) C:\Users\bernd>conda install spyder=5.5.1
Retrieving notices: done
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

# All requested packages already installed.

(base) C:\Users\bernd>
```

Abb. 4-17: Anaconda Prompt

Wenn Sie anschließend Spyder erneut starten und die Funktion `print("Hochschule Ansbach")` eingeben und den Code ausführen, indem Sie auf F5 oder „Run file“ klicken, dann sollte die Ausgabe in der Konsole erscheinen (siehe Abb. 4-18).

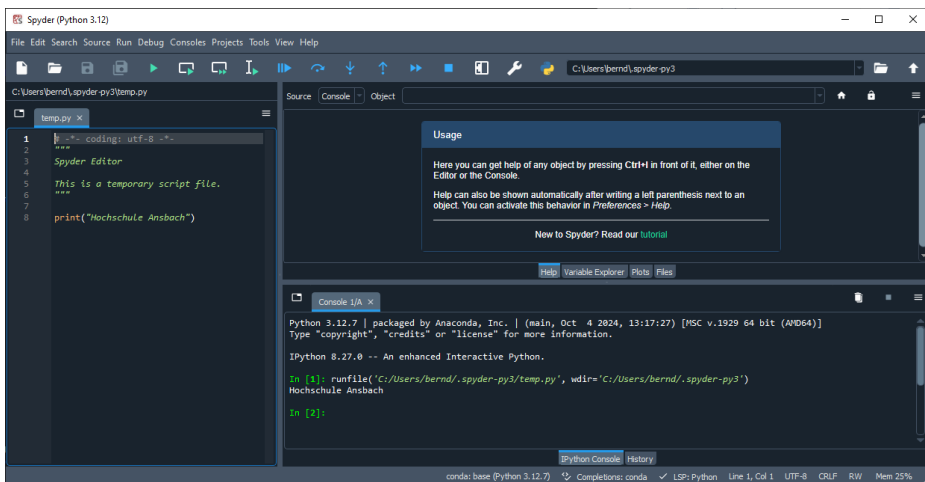


Abb. 4-18: Python Editor

In der Konsole (unten rechts) wird „Hochschule Ansbach“ angezeigt und damit haben Sie Ihr erstes Python-Programm erstellt.

Die Dokumentation zu der Spyder Entwicklungsumgebung inklusive Videoanleitungen finden Sie unter der URL <https://docs.spyder-ide.org/current/index.html> (Spyder, 2025), wie in Abb. 4-19 dargestellt.

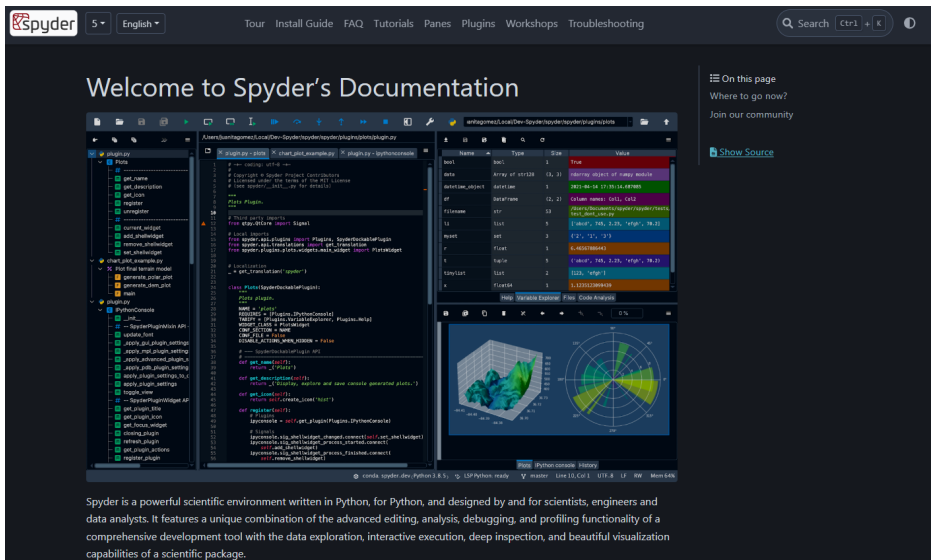


Abb. 4-19: Spyder Bedienungsanleitung (Spyder, 2025)

Das Layout von Spyder ist wie folgt geordnet in die Bereiche (siehe Abb. 4-20):

1. Zeilennummer und Codeanalyse-Warnungen. Click auf einer Zeile erzeugt einen Breakpoint (Debugger).
2. Scrollbalken.
3. Kontextmenü (Rechtsklick) zeigt mögliche Aktionen.
4. Optionsmenü-Icon erlaubt Editor-Einstellungen.
5. Pfad und Dateiname.
6. Tab-Bar zeigt alle geöffneten Dateien.

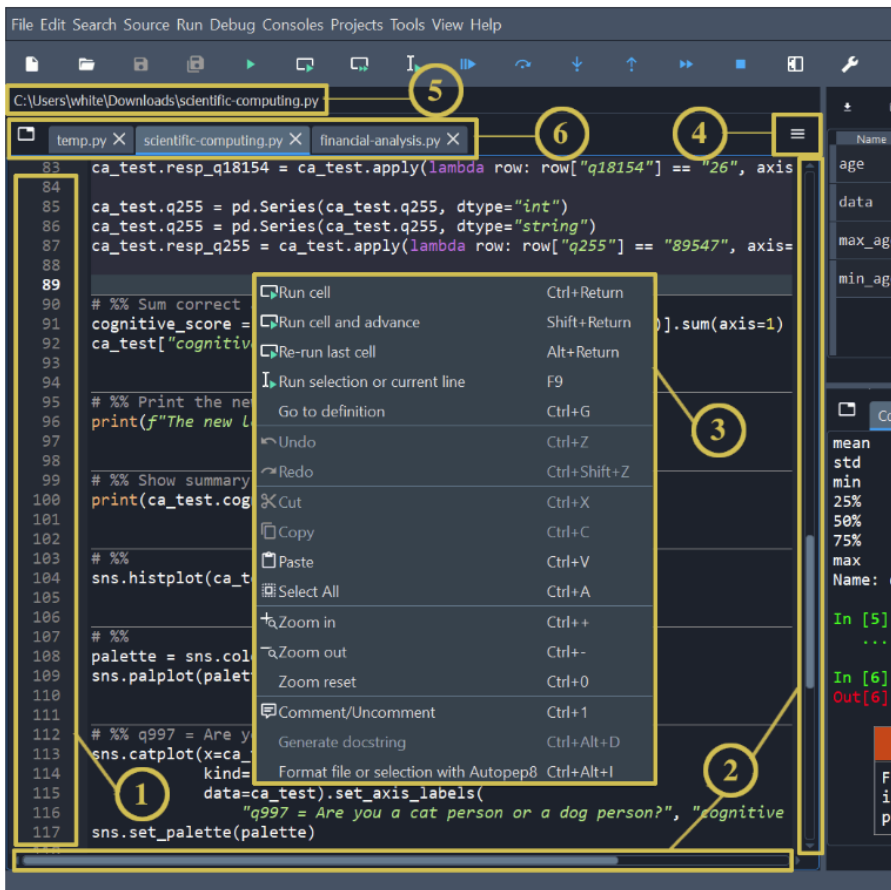


Abb. 4-20: Spyder Layout (Spyder, 2025)

Die Fenster in Spyder, sogenannte Panes, können beliebig angeordnet werden (Auswahl via Menü View/Panes). Auch die Anzeige der Toolbars kann individuell gestaltet werden (Auswahl via Menü View/Toolbars, siehe Abb. 4-21).

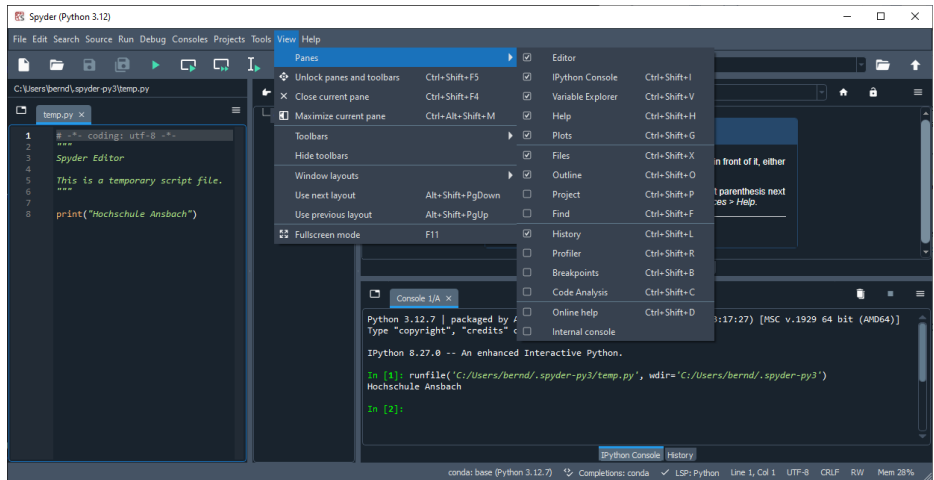


Abb. 4-21: Panes Optionen

Blöcke von Code (Code-Zellen) lassen sich in Spyder mit der Zeichenfolge `#%%` Name anlegen und werden in der Outline-Pane mit ihrem Namen und vorangestelltem `%` angezeigt. Blöcke können durch zusätzliche `%`-Zeichen untergeordneten Gliederungsebenen zugeordnet werden, z.B. `#%%%` Name für die dritte Ebene bzw. `#%%%%` Name für die vierte Ebene. Die Navigation zu Blöcken im Outline-Pane ist durch Click auf den Namen möglich (siehe Abb. 4-22).

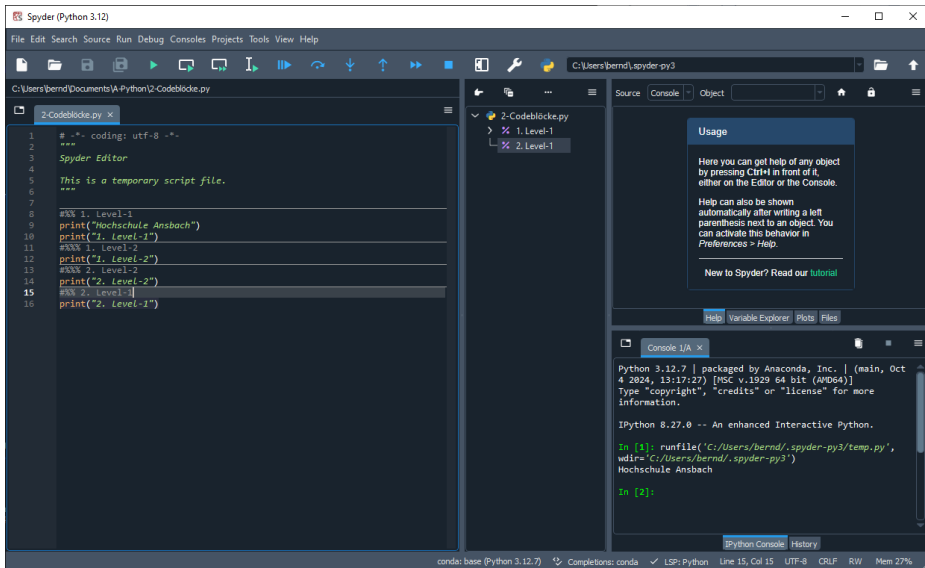


Abb. 4-22: Code-Blöcke und Outline-Pane vor Ausführung

Speichern Sie den obigen Code mit “File/Save as” als “2-Codeblöcke” ab. Die Skriptdateien mit Code werden von Spyder mit der Dateierdung “.py” gespeichert.

Mit den Icons “Run current cell” bzw. “Run current cell and go to the next one” oder via Tastatur mit Ctrl-Return bzw. Shift-Return können alle Anweisungen eines Blockes ausgeführt werden. Durch die Anweisungen erzeugte textuelle Ausgaben werden in der Konsole angezeigt (siehe Abb. 4-23).

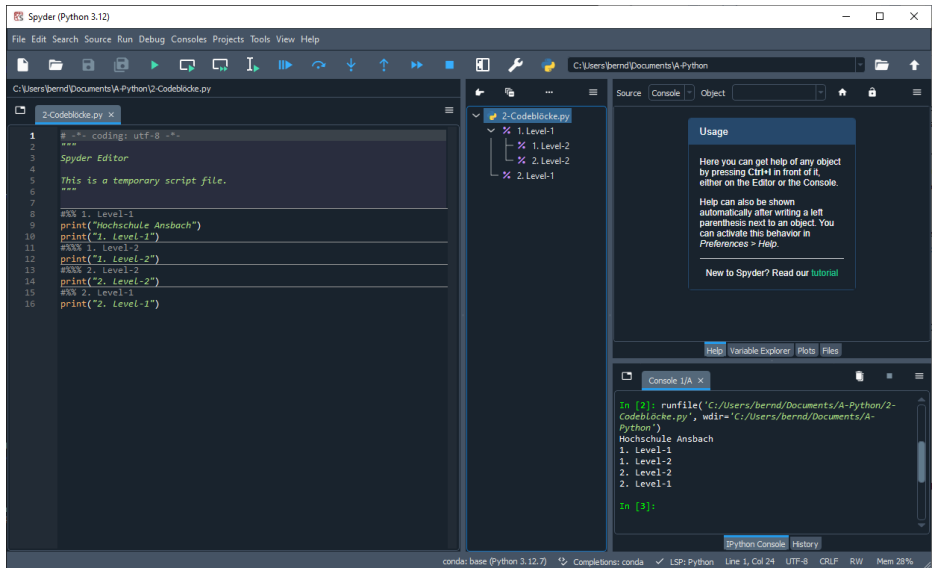


Abb. 4-23: Code-Blöcke, Outline-Pane und Konsole nach Ausführung

Durch Anweisungen erzeugte grafische Ausgaben werden in der **Plots-Pane** angezeigt (siehe Abb. 4-24). Die Variablen im Arbeitsspeicher werden in der **Variable Explorer-Pane** angezeigt (siehe Abb. 4-25). Im Variable Explorer werden der Name, Typ, Größe und Inhalt der Variablen angezeigt. Mit einem Doppelklick auf ein Objekt werden weitere Details zu einem Objekt angezeigt. Über die Toolbar ist auch die Speicherung aller aktuellen Variablen in einer .spydata-Datei möglich, so dass dieser Zustand zu einem späteren Zeitpunkt wiederhergestellt werden kann. Die Toolbar erlaubt auch das Löschen von Variablen.

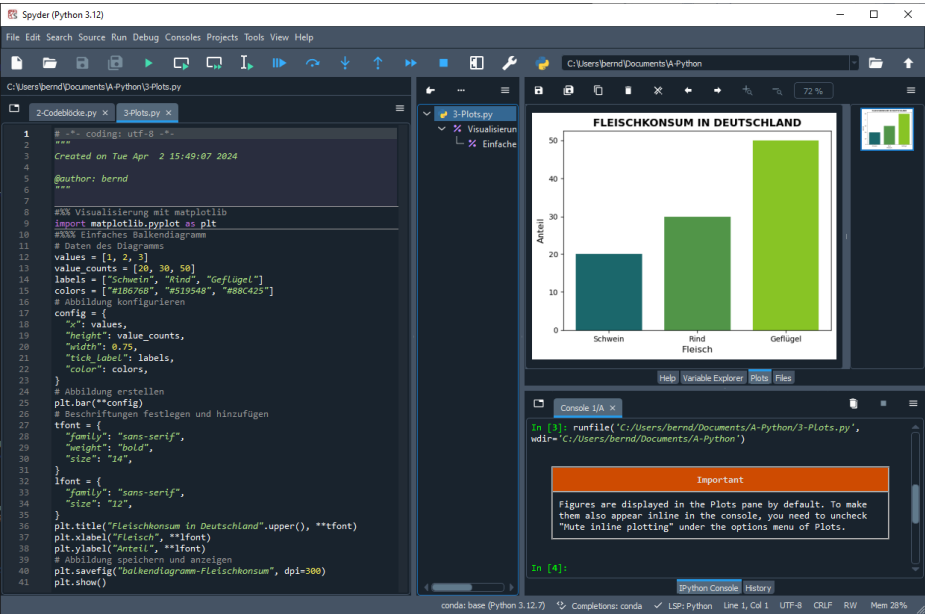


Abb. 4-24: Plots-Pane

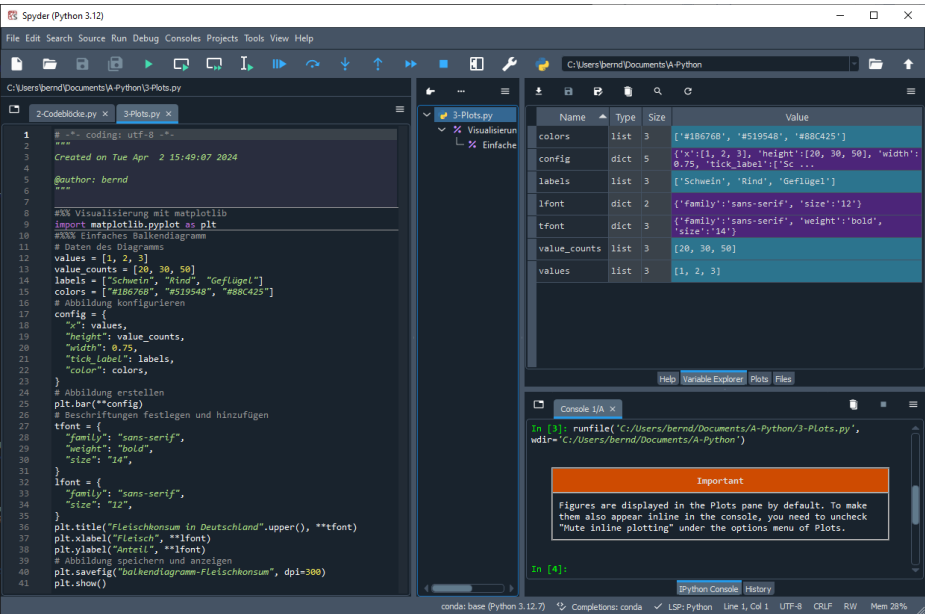


Abb. 4-25: Variable Explorer-Pane

Auf der **Help-Pane** kann die Dokumentation zu Python angezeigt werden (siehe Abb. 4-26). Die Dokumentation findet sich auch unter der URL <https://www.python.org/doc/> (Python Software Foundation, 2025).

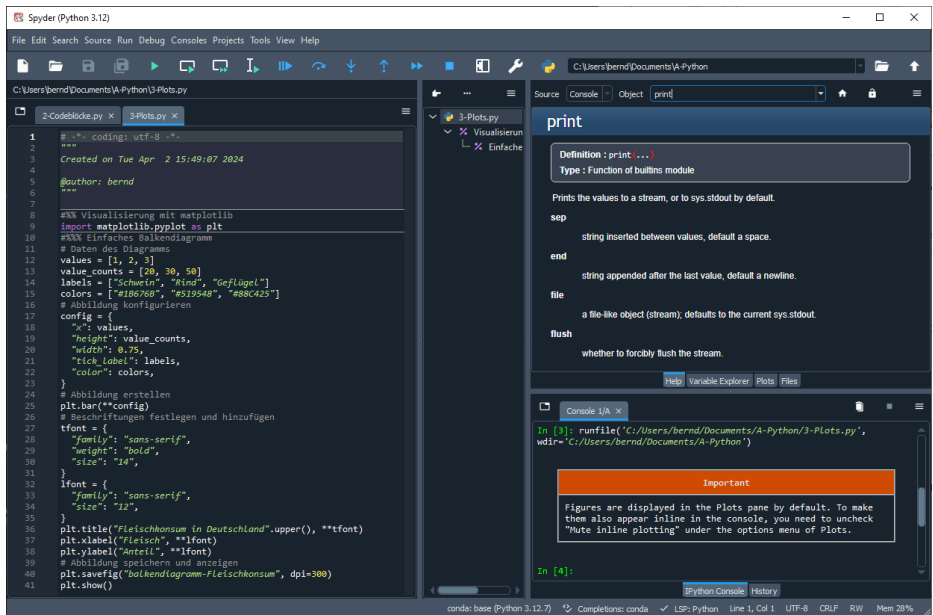


Abb. 4-26: Help-Pane

Wenn Dateien geöffnet oder gespeichert werden, so erfolgt dies im **Arbeitsverzeichnis (Working Directory)**. Dies kann im Menü Tools/Preferences festgelegt werden (siehe Abb. 4-27). Im File-Pane werden die Dateien des Arbeitsverzeichnisses dann angezeigt.

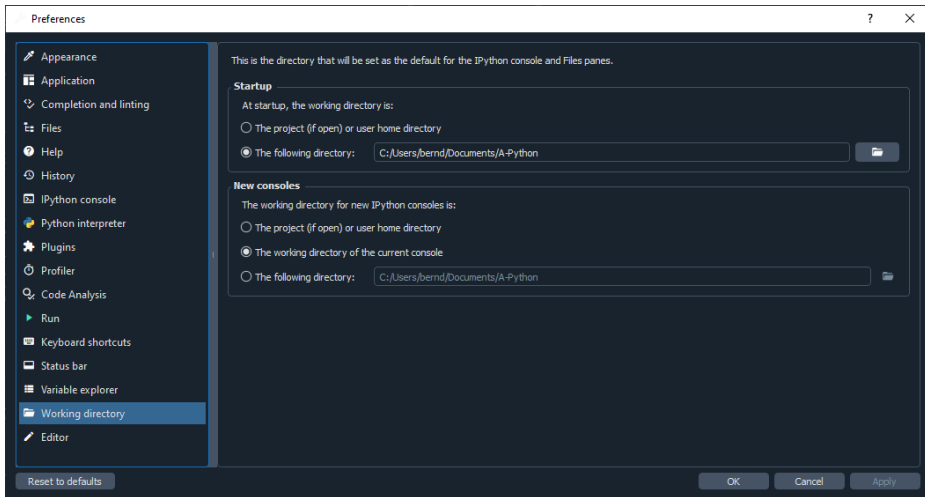


Abb. 4-27: Arbeitsverzeichnis

4.4 R

Um auch Daten und Funktionen von R in Python verwenden zu können, gilt es im Vorfeld R zu installieren. Die aktuelle Version kann von der Webseite des R-Projects (R Foundation, 2025) heruntergeladen werden (siehe Abb. 4-28).



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Conferences

Search

Get Involved: Mailing Lists

Get Involved: Contributing

Developer Pages

R Blog

R Foundation

Foundation

Board

Members

Donors

Donate

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- The **useR! 2025** conference will take place at Duke University, in Durham, NC, USA, August 8-10.
- **R version 4.4.2 (Pile of Leaves)** has been released on 2024-10-31.
- We are deeply sorry to announce that our friend and colleague Friedrich (Fritz) Leisch has died. [Read our tribute to Fritz here.](#)
- **R version 4.3.3 (Angel Food Cake)** (wrap-up of 4.3.x) was released on 2024-02-29.
- You can support the R Foundation with a renewable subscription as a [supporting member](#).

News via Mastodon



useR_conf

UseR! 2025 Call for Proposals is now open! We invite R users and developers from around our region and the world to submit abstracts for useR! 2025.
Deadline: Monday 3rd March 2025

Abb. 4-28: Das R Projekt

Klicken Sie zunächst auf den Link „download R“. Ihnen wird dann eine Liste der Server des CRAN-Netzwerks angezeigt, von denen R geladen werden kann. Wählen Sie einen Server in ihrer Nähe aus, indem Sie z.B. auf den Link „<https://ftp.fau.de/cran>“ der FAU Universität klicken (siehe Abb. 4-29).

Germany

<https://mirror.howtolearnlanguage.info/>

<https://ftp.fau.de/cran/>

<https://cran.datenrettung360.de/>

<https://ftp.gwdg.de/pub/misc/cran/>

<https://mirror.dogado.de/cran/>

<https://cran.uni-muenster.de/>

<https://mirror.clientvps.com/CRAN/>

<https://mirror.kamp.de/cran/>

dogado GmbH

Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Hetzner Online GmbH

GWDG Göttingen

dogado GmbH

University of Münster, Germany

ClientVPS

KAMP Netzwerkdienste GmbH

Abb. 4-29: Wahl des CRAN Servers

Die R-Software wird dann von dem ausgewählten Server heruntergeladen, wenn Sie auf den Link „Download R for ...“ (Wählen Sie Ihr Betriebssystem) klicken (siehe Abb. 4-30).

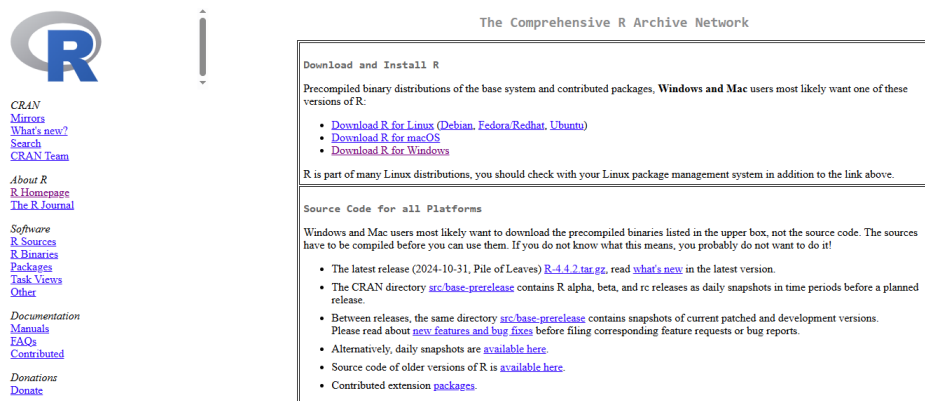


Abb. 4-30: Download R for ...

Klicken Sie anschließend auf „install R for the first time“ (siehe Abb. 4-31).

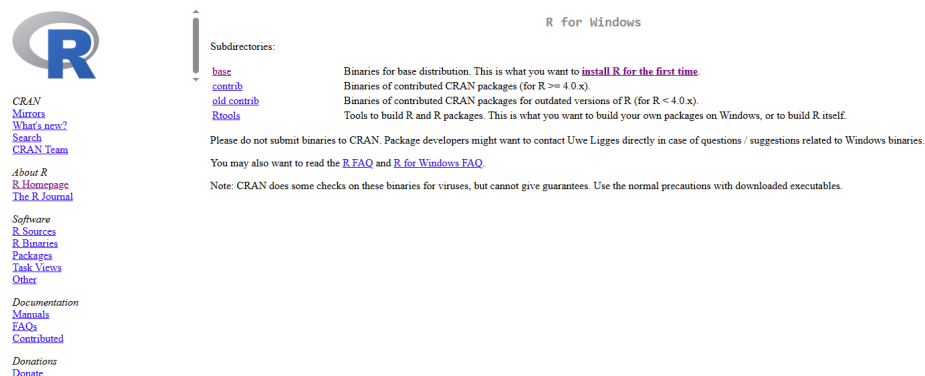


Abb. 4-31: Installieren von R

Es wird Ihnen automatisch die aktuelle Version von R zum Download angeboten. Klicken Sie auf den entsprechenden Link, z.B. „Download R 4.4.2 for Windows“ (siehe Abb. 4-32).



R-4.4.2 for Windows

[Download R-4.4.2 for Windows](#) (83 megabytes, 64 bit)

[README on the Windows binary distribution](#)

[New features in this version](#)

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from [here](#).

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [<CRAN_MIRROR>bin/windows/base/release.html](#).

Last change: 2024-11-01

Abb. 4-32: Download starten

Klicken Sie anschließend auf „Öffnen“, um die Installation fortzuführen. Wählen Sie anschließend die Sprache aus, in der die Installation erfolgen soll. Dann werden Ihnen die Lizenzbedingungen angezeigt. Bestätigen Sie diese durch Klicken auf „Weiter“, um die Installation fortzusetzen. Wählen Sie den Zielordner aus und bestätigen Sie dies mit „Weiter“. Wählen Sie anschließend die Komponenten aus, die Sie installieren möchten. Es ist empfehlenswert die Einstellungen wie vorgeschlagen zu installieren. Sie können die Startoptionen unverändert lassen. Um einen Ordner im Startmenü anlegen zu lassen, klicken Sie auf den Button „Weiter“. Die Versionsnummer sollte im Registry gespeichert werden und die Dateien mit der Endung .RData sollten mit R verknüpft werden, so dass diese später mit R geöffnet werden. Die erfolgreiche Installation wird Ihnen bestätigt und Sie können dies mit „Fertigstellen“ bestätigen.

4.5 Rtools

Für einige der Funktionalitäten von R und dessen Pakete sind spezielle Werkzeuge erforderlich. Laden Sie sich daher die aktuelle Version von Rtools (R CRAN Team, 2024) herunter (siehe Abb. 4-33).

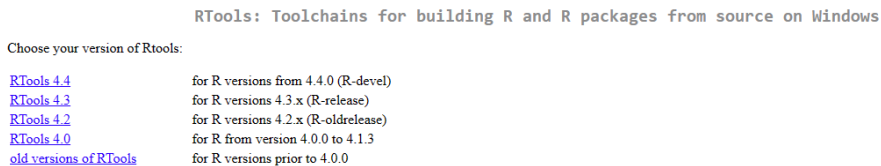


Abb. 4-33: R-Tools Versionen

Klicken Sie z.B. auf den Link „Rtools43_installer“ (siehe Abb. 4-34) und führen Sie die Installationsdatei anschließend aus.

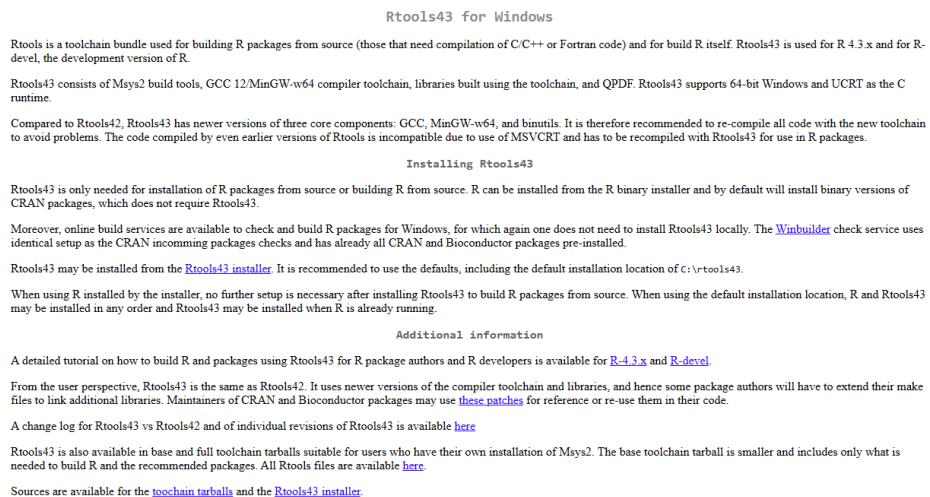


Abb. 4-34: Rtools Installation

Wählen Sie den Zielordner aus und bestätigen Sie dies mit „Weiter“ (siehe Abb. 4-35).

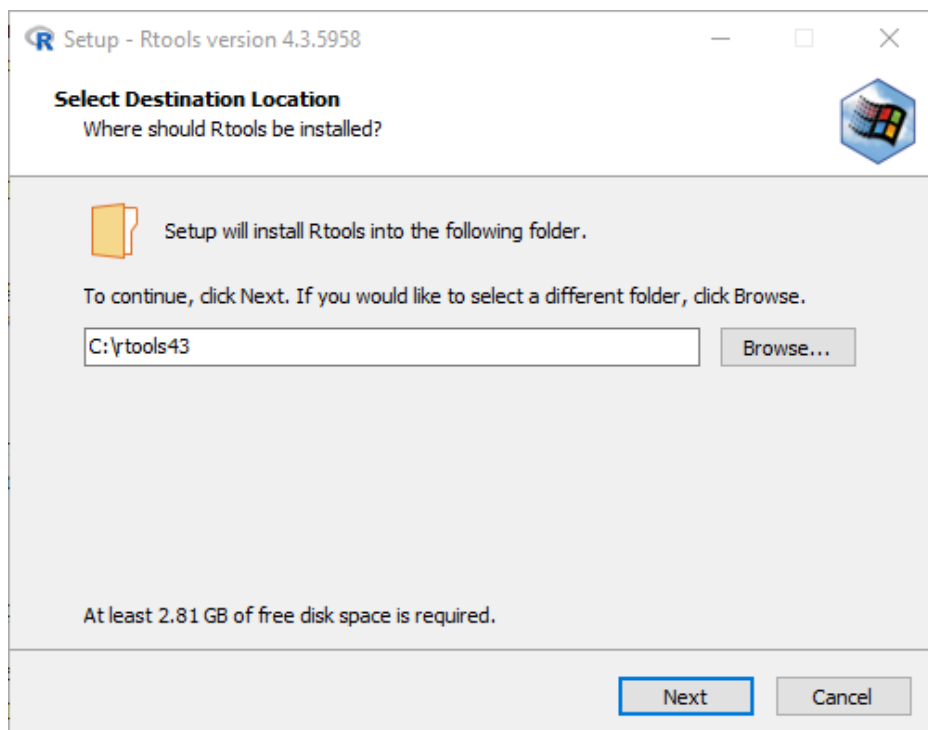


Abb. 4-35: Zielordner für Installation

Wählen Sie anschließend die Optionen so, dass die Rtools-Version zur Registry hinzugefügt wird (siehe Abb. 4-36).

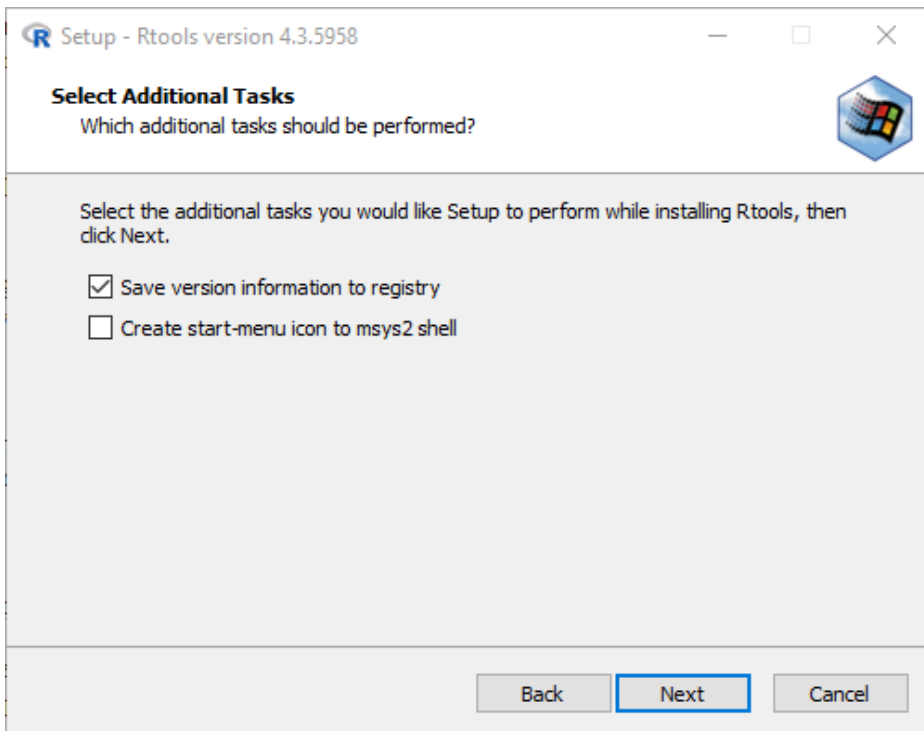


Abb. 4-36: Rtools-Version zu Registry hinzufügen

Bestätigen Sie den Start der Installation mit „Install“ (siehe Abb. 4-37).

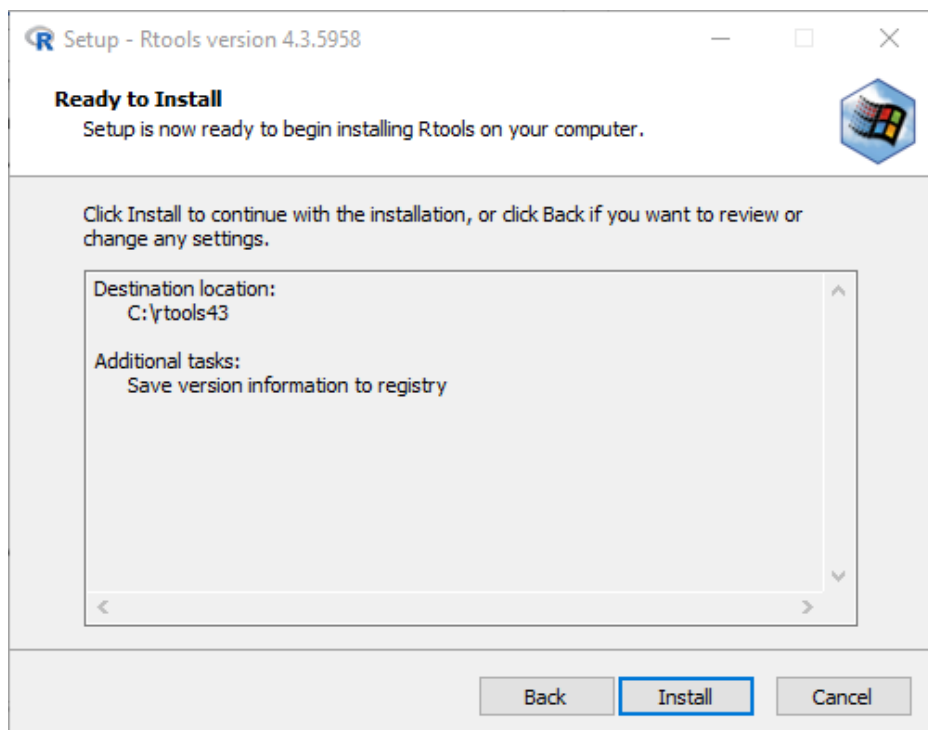


Abb. 4-37: Installation ausführen

Die erfolgreiche Installation wird Ihnen bestätigt und Sie können dies mit „Finish“ bestätigen (siehe Abb. 4-38).

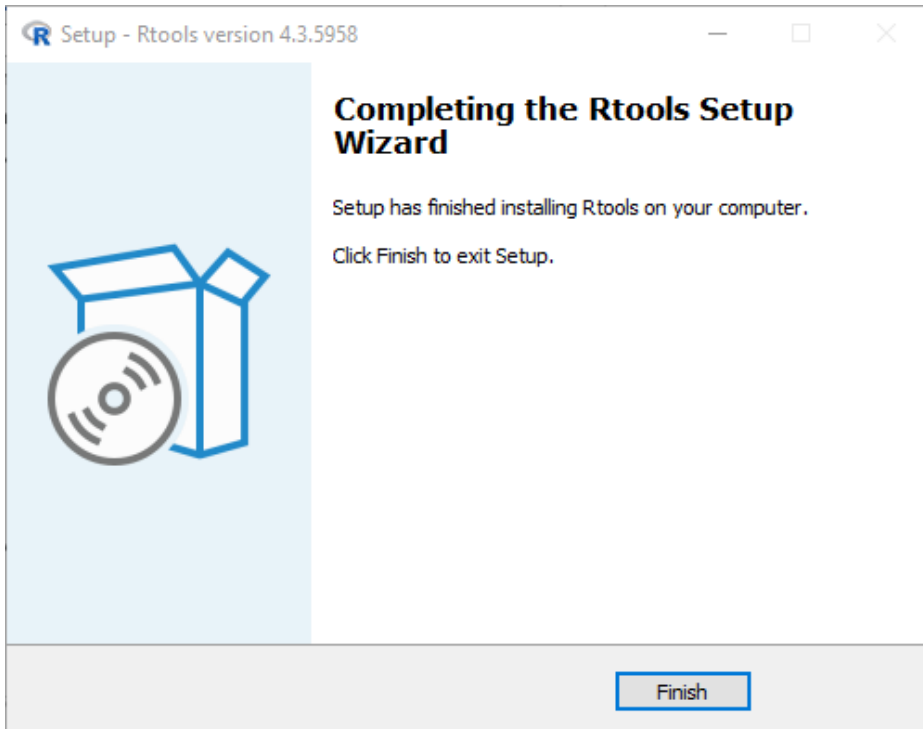


Abb. 4-38: Rtools erfolgreich installiert

4.6 RStudio

Eine etablierte Benutzeroberfläche für Anwender von R und Python ist RStudio, welches Sie sich von der Webseite von RStudio (Posit, 2025) herunterladen können. Klicken Sie auf „Products“ und wählen dann „RStudio IDE“ aus (siehe Abb. 4-39).

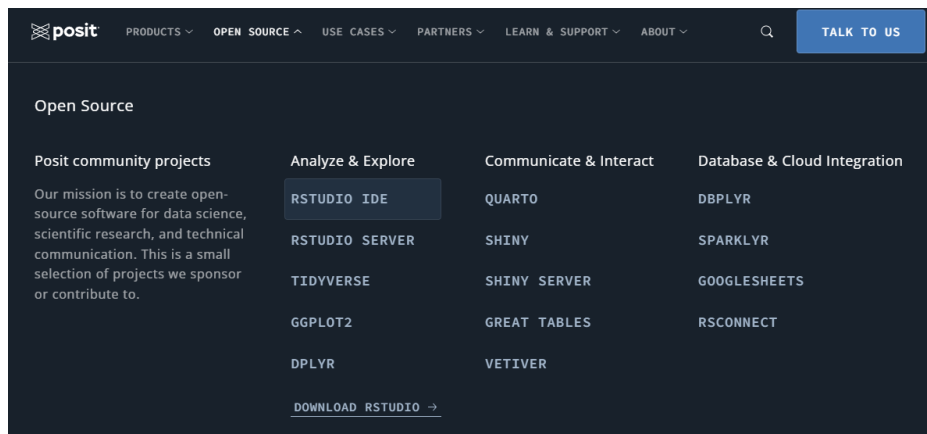


Abb. 4-39: RStudio

Auf dem nächsten Bildschirm scrollen Sie nach unten und wählen dann „Download RStudio Desktop“. Diese Anwendung ist für den Einzelplatzrechner geeignet (siehe Abb. 4-40).

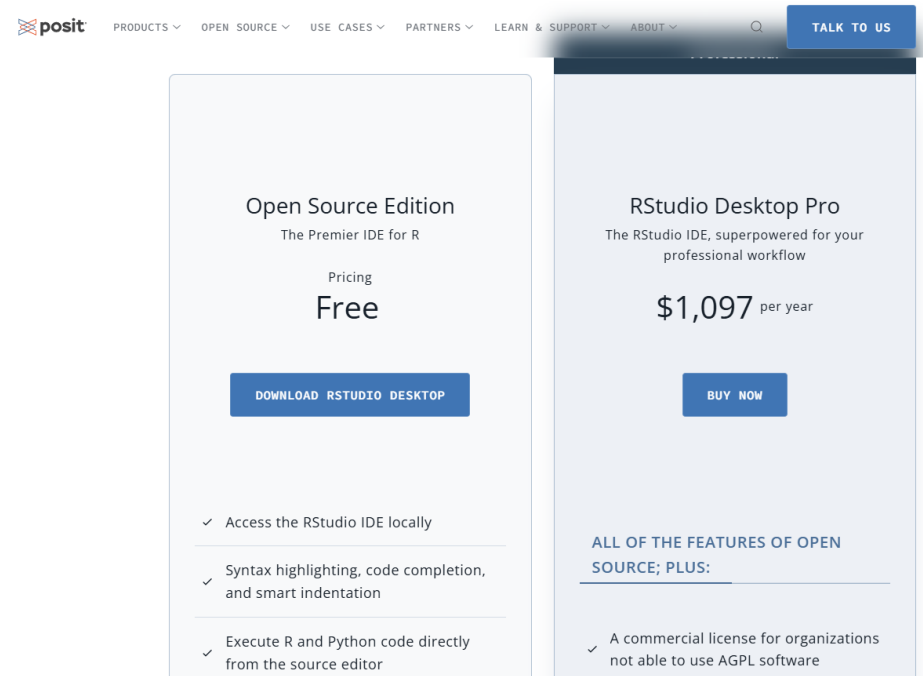


Abb. 4-40: RStudio Desktop

Wählen Sie auch auf dem nachfolgenden Screen die Version für Ihr Betriebssystem aus und klicken auf den entsprechenden Link, z.B. „RStudio-2024.12.0+467.exe“. Klicken Sie auf den Button „Weiter“, um die Installation zu starten. Bestimmen Sie das Zielverzeichnis für die Installation. Wählen Sie anschließend den Ordner des Startmenüs aus, in dem RStudio hinzugefügt werden soll und klicken Sie dann auf „Installieren“. Die erfolgreiche Installation wird Ihnen bestätigt und Sie können dies mit „Fertigstellen“ bestätigen.

Nachfolgend wird Ihnen vorgestellt, wie Sie RStudio starten und beenden können und erweiterte Funktionalitäten mit Hilfe von Paketen nutzbar machen. Wenn Sie die Basisversion von R starten, dann erscheint die RGui (Gui = Graphical User Interface) mit der Konsole und ein einfaches Menü (siehe Abb. 4-41).

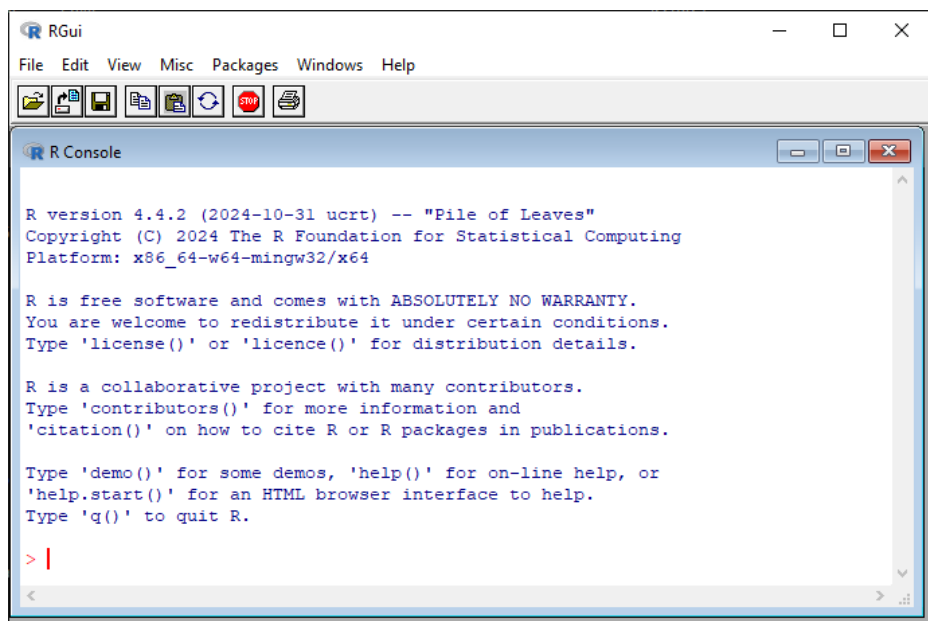


Abb. 4-41: Gui der R-Basisversion

Die einfachste Möglichkeit mit R zu arbeiten ist die Eingabe von Befehlen in die Konsole. Nach der Ausführung eines Befehls wird dann dessen Ausgabe, sofern der Befehl etwas ausgibt, in der Konsole angezeigt.

Eine komfortablere Benutzeroberfläche als das Gui der R-Basisversion ist RStudio. Im Folgenden wird daher RStudio verwendet (siehe Abb. 4-42).

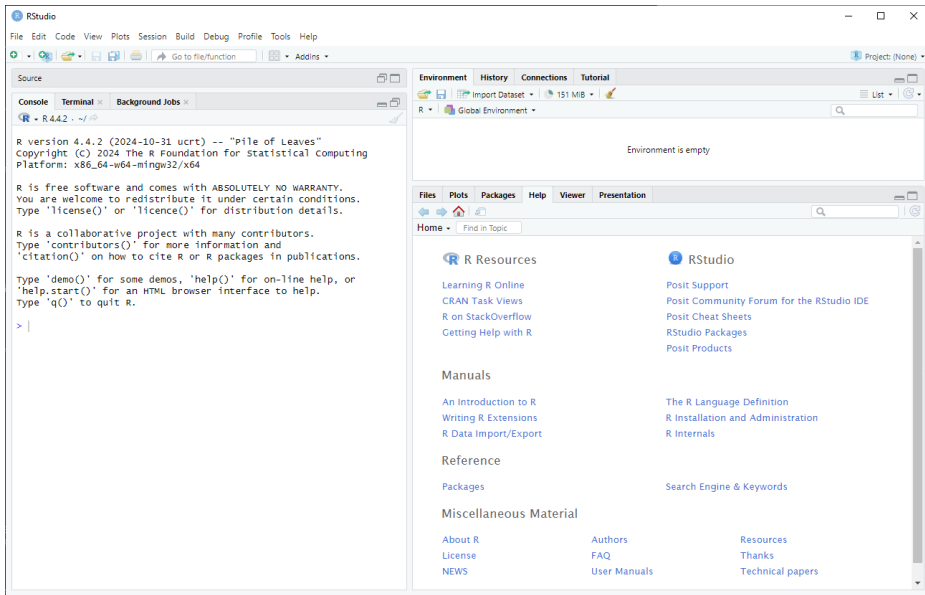


Abb. 4-42: Gui von RStudio

Auch in RStudio werden Befehle entweder in der Konsole, einem Fenster zur Befehlseingabe, oder in einer Skriptdatei eingegeben. Eine **Skriptdatei** hat den Vorteil, dass die darin enthaltenen Befehle für eine spätere erneute Ausführung gespeichert und somit wiederverwendet werden können. Wenn ein Befehl in der Konsole (Quadrant links in Abb. 4-42) eingegeben wird, dann erfolgt dessen Ausführung nach Drücken der Enter-Taste. Kann der Befehl nach der Eingabe von Enter nicht abgeschlossen werden, z.B. weil eine Klammer oder ein Parameter fehlt, zeigt R statt der Ausgabe ein + an, weil der Befehl so nicht ausgeführt werden kann. Um einen Befehl aus einer Skriptdatei auszuführen, gilt es zunächst eine Skriptdatei über das Menü **File/New File/R Script** anzulegen (siehe Abb. 4-43).

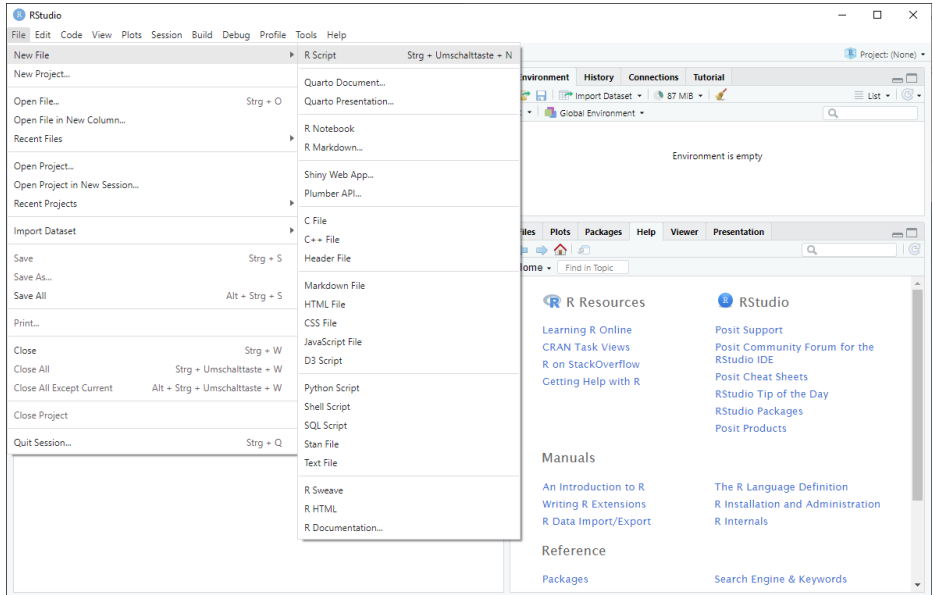


Abb. 4-43: Neue Skriptdatei anlegen

In dem neu geöffneten Fenster mit dem R-Skript kann ein Befehl, z.B. `1+2`, ausgeführt werden, indem der Cursor auf die Zeile positioniert (Quadrant oben links in Abb. 4-44) und dann `Ctrl+Enter` gedrückt wird oder alternativ durch Drücken auf den Button **Run**. Nach der Ausführung eines Befehls wird dann dessen Ausgabe, sofern der Befehl etwas ausgibt, in der Konsole angezeigt. Um eine R-Anweisung abubrechen, wenn Sie zu lange zur Ausführung benötigt, kann die `Esc`-Taste gedrückt werden.

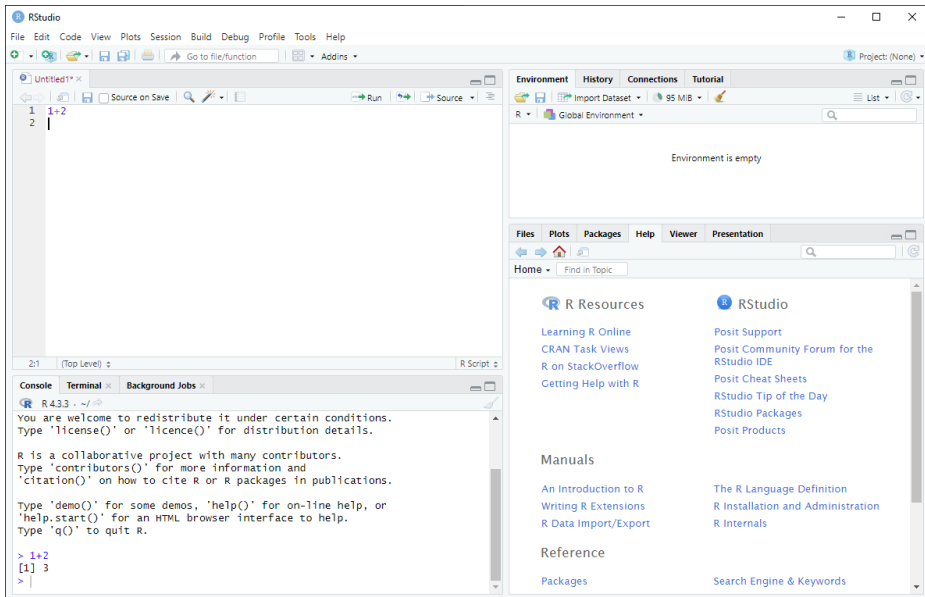


Abb. 4-44: RStudio mit Skriptfenster

Wenn einer der zuletzt verwendeten Befehle erneut ausgeführt werden soll, kann die Taste \uparrow verwendet werden. Um eine Abfolge von Prozessschritten bei der Datenanalyse immer wieder auszuführen, empfiehlt sich auf jeden Fall das Arbeiten mit Skriptdateien. Die Befehle aus dem Skriptfenster können über das Menü **File/Save As...** als Skriptdatei gespeichert werden und zu einem späteren Zeitpunkt über das Menü **File/Open File...** wieder geladen werden (siehe Abb. 4-45).

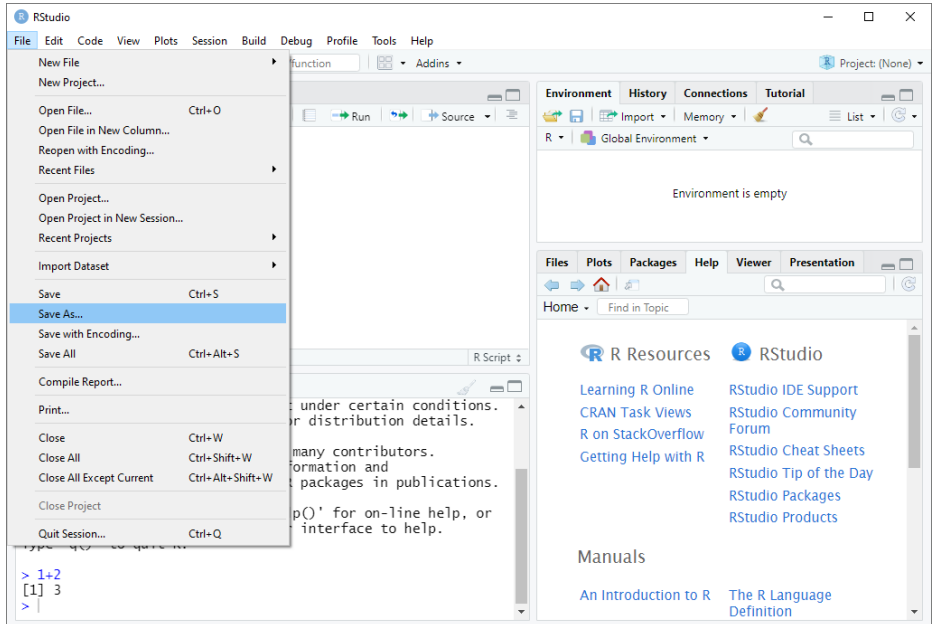


Abb. 4-45: Skriptdatei speichern

Speichern Sie das Skript mit dem Namen „0-ErstesRProgramm“ ab. Die Dateierweiterung für R-Skriptdateien ist „.R“. Im Rahmen des Skriptfensters steht nach dem Speichern der Name der Skriptdatei (siehe Abb. 4-46). Es ist möglich mehrere Skriptdateien gleichzeitig zu öffnen. Diese werden dann nebeneinander angezeigt.

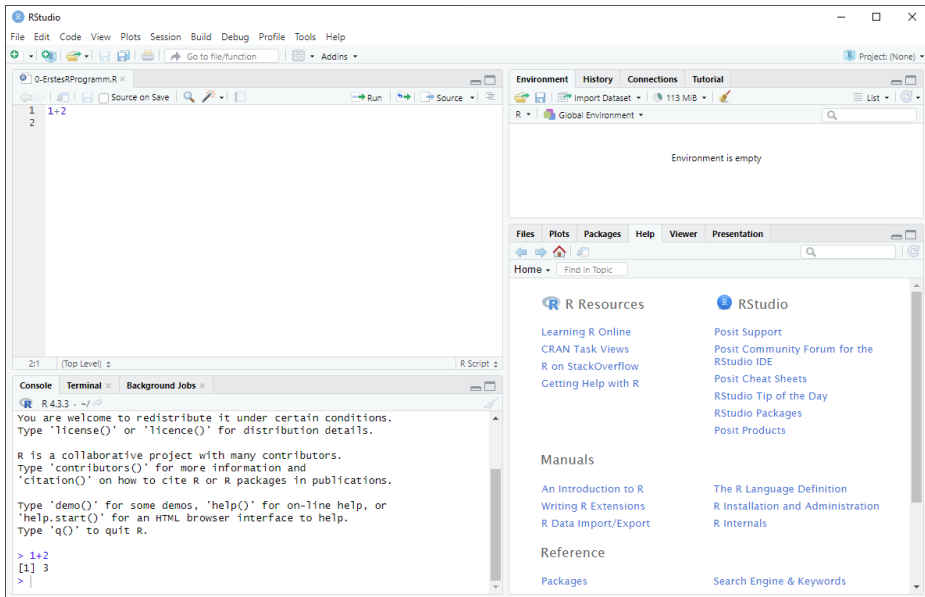


Abb. 4-46: Skriptdatei 0-ErstesRProgramm.R

Eine R-Session kann in RStudio über das Menü mit **File/Quit Session** oder mit der Funktion `q()` beendet werden. Wenn R beendet wird, folgt im Standard die Frage, ob der Workspace (alle Objekte/Variablen/Funktionen im Global Environment) gespeichert werden sollen. Bei jedem Start von R wird der Workspace geladen. Wenn es also gewünscht ist, dass die Arbeit später an gleicher Stelle fortgesetzt werden soll, wo diese aufgehört wurde, dann ist es sinnvoll den Workspace zu speichern. Möchte man nicht bei jedem Beenden von R gefragt werden, ob der Workspace gespeichert werden soll, kann man dies auch im Menü **Tools/Global Options/General** dauerhaft einstellen (Optionen: Always, Never, Ask).

4.7 Pakete in R installieren

Es existieren mehr als 15.000 sogenannte Pakete (Packages) mit erweiterten Funktionen für R, die üblicherweise über das CRAN abrufbar sind und

mit der Funktion `install.packages("Paket")` von dort abgerufen und in das Dateiverzeichnis R-Library auf dem lokalen Rechner kopiert werden können. Die Angabe des Paketnamens muss bei dieser Funktion in Anführungszeichen erfolgen. Die Installation ist auf einem Rechner nur ein einziges Mal erforderlich. Es empfiehlt sich den Parameter `dependencies=TRUE` zu verwenden, so dass automatisch auch alle anderen Packages installiert werden, auf deren Funktionen und Inhalte das installierte Paket zurückgreift.

Eine Liste der bereits installierten Pakete kann mit der Funktion `installed.packages()` angezeigt werden.

Nach jedem Start von R muss die Funktion `library(Paket)` aufgerufen werden, um das Paket in den Arbeitsspeicher zu laden bzw. dies zu aktivieren. Vor Ausführung dieses Befehls können die in einem Package enthaltenen Funktionen nicht genutzt werden. Ist ein Paket für die Ausführung eines Skriptes erforderlich, so ist die Funktion `require()` geeignet, um bei Bedarf sowohl die Installation als auch das Laden eines Paketes zu veranlassen.

Das Paket **tidyverse** (Posit, 2024b) ist ein sogenanntes Kompositpaket, welches die Pakete `ggplot2`, `dplyr`, `tidyr`, `readr`, `purrr`, `tibble`, `stringr` und `forcats` beinhaltet. Anstatt diese einzeln installieren und laden zu müssen wird dies mit einer Anweisung über das Paket **tidyverse** erreicht. Es ist empfehlenswert dieses Paket für Data-Science-Projekte zu laden. Für wesentliche Funktionen des Machine Learnings empfiehlt sich ebenfalls die Installation des Kompositpakets `tidymodels` (Posit, 2024a), welches die Pakete `rsample`, `parsnip`, `recipes`, `workflows`, `tune`, `yardstick`, `broom` und `dials` beinhaltet.

Werden bei dem Laden eines Pakets Konflikte angezeigt, wie nachfolgend beim Laden von `tidyverse`, dass die Funktion `filter` nicht nur im Paket

dplyr, sondern auch im Paket **stats** existiert, dann gilt es bei Bedarf dem Funktionsaufruf von `filter` aus dem jeweiligen Paket den Paketnamen mit zwei Doppelpunkten voranzustellen, z.B. `dplyr::filter()` bzw. `stats::filter()`. Dies stellt sicher, dass die jeweils gewünschte `filter`-Funktionen vom System ausgeführt wird.

Packages installieren und aktivieren, Hilfe zu Packages

```
#- Pakete-----
#install.packages("tidyverse")      # Installation Package (einmalig)
library(tidyverse)                  # Aktivieren Package (bei jedem R-Start)
#?tidyverse                          # Hilfe zu Package
#help(tidyverse)                    # Hilfe zu Package
#install.packages("tidymodels")     # Installation Package (einmalig)
library(tidymodels)                 # Aktivieren Package (bei jedem R-Start)
```

Wenn ein Package geladen bzw. aktiviert ist, können die darin enthaltenen Funktionen aufgerufen werden. Mit `?Paket` bzw. `help(Paket)` kann die Hilfe zu einem Package oder mit `?Funktion` bzw. `help(Funktion)` die Hilfe zu einer Funktion aus dem Paket aufgerufen werden, z.B. `?tidyverse` für die Hilfe zu dem Paket **tidyverse** oder `?geom_boxplot` für die Hilfe zu der Funktion `geom_boxplot()` aus dem Paket **ggplot2**.

Da sich die Funktionalität der R-Pakete permanent weiterentwickelt, ist es ab und zu sinnvoll die installierten Pakete auf den neuesten Stand zu bringen. Dies ist mit der Funktion `update.packages()` möglich.

Sollte man gewisse Pakete nicht mehr benötigen, so kann man diese mit der Funktion `remove.packages()` jederzeit entfernen.

Die Installation, Aktualisierung, Aktivierung oder das Entfernen von Paketen ist ebenso über RStudio möglich (Quadrant unten rechts in Abb. 4-47).

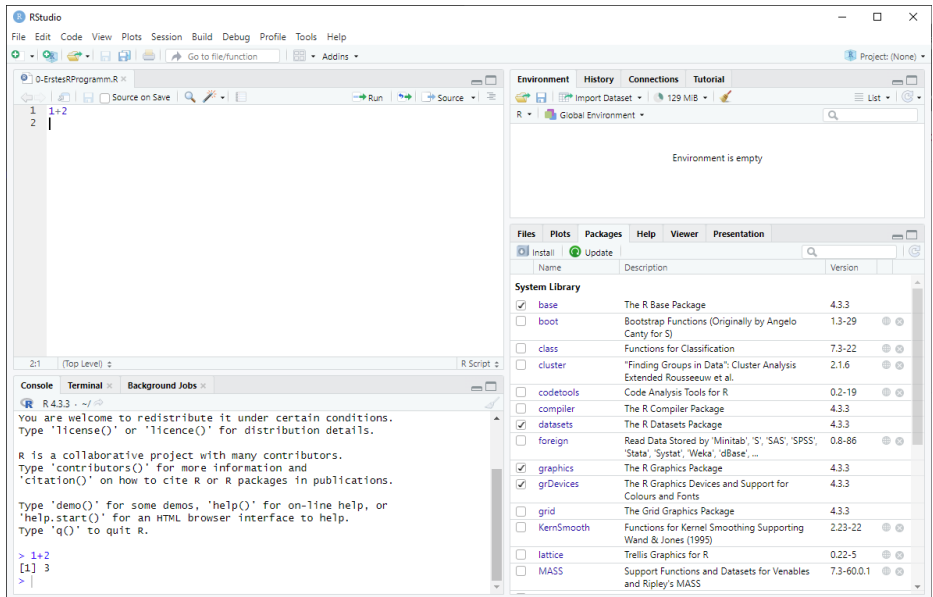
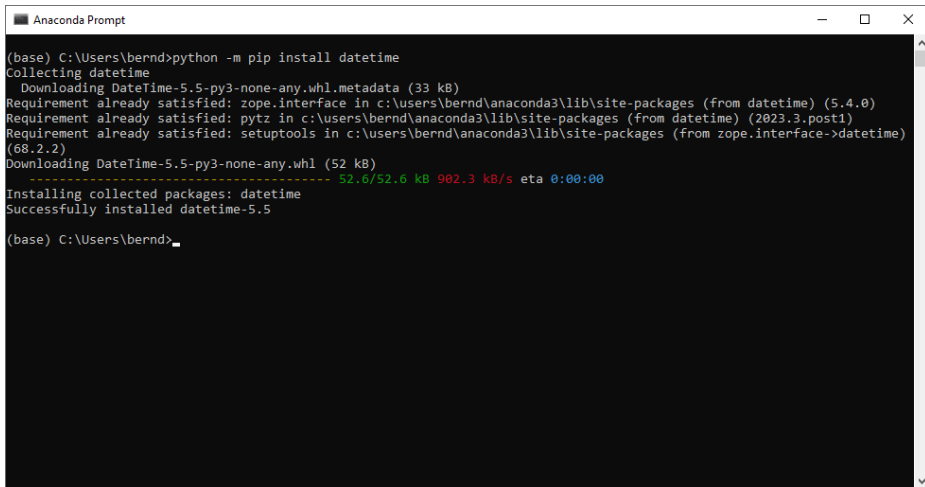


Abb. 4-47: Installation und Aktivierung von Paketen

4.8 Pakete in Python installieren

Ebenso wie in R gilt es auch in Python Pakete zu installieren (einmalig) und zu aktivieren (nach jedem Start einer Python-Session). Die Installation von Paketen erfolgt über den Anaconda Prompt (siehe Abb. 4-6). In der Kommandozeile wird die Funktion `python -m pip install paketname` ausgeführt (siehe Abb. 4-48). Der Upgrade eines Pakets kann mit der Funktion `python -m pip install -upgrade paketname` erreicht werden.



```
(base) C:\Users\bernd>python -m pip install datetime
Collecting datetime
  Downloading DateTime-5.5-py3-none-any.whl.metadata (33 kB)
Requirement already satisfied: zope.interface in c:\users\bernd\anaconda3\lib\site-packages (from datetime) (5.4.0)
Requirement already satisfied: pytz in c:\users\bernd\anaconda3\lib\site-packages (from datetime) (2023.3.post1)
Requirement already satisfied: setuptools in c:\users\bernd\anaconda3\lib\site-packages (from zope.interface->datetime) (68.2.2)
Downloading DateTime-5.5-py3-none-any.whl (52 kB)
----- 52.6/52.6 kB 902.3 kB/s eta 0:00:00
Installing collected packages: datetime
Successfully installed datetime-5.5

(base) C:\Users\bernd>
```

Abb. 4-48: Installation von Paket über Anaconda Prompt

Die Aktivierung eines Pakets erfolgt im Python-Code mit der Funktion `import paketname`. Sollen die Funktionen des Pakets über eine Abkürzung leichter aufgerufen werden, so ist es möglich bei der Funktion mit dem Parameter `as` eine Abkürzung zu bestimmen, z.B. `import paketname as pt`. Wie dies mit Spyder umgesetzt wird ist in Abb. 4-49 ersichtlich.

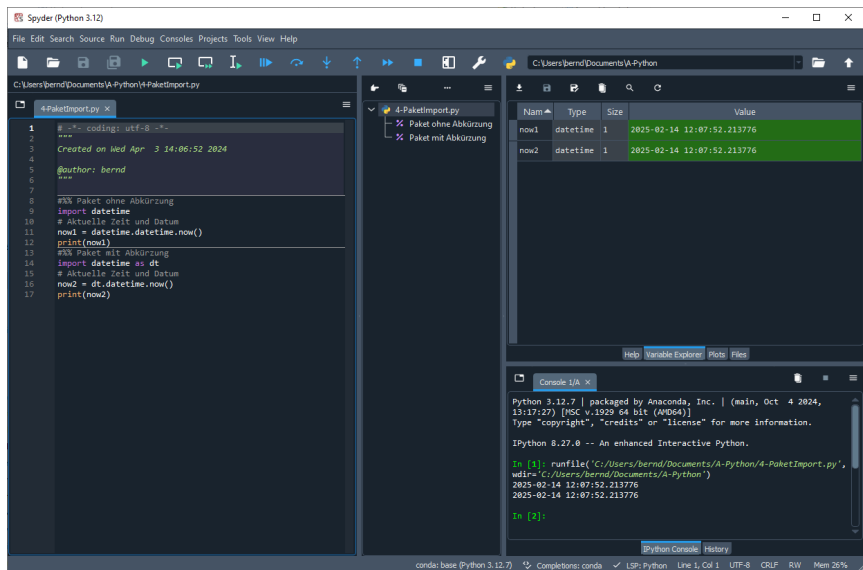


Abb. 4-49: Aktivierung und Ausführung von Paket in Spyder

In RStudio kann dies gleichermaßen erfolgen, wenn zuvor die Konfiguration von RStudio abgeschlossen wurde, die in Abschnitt 4.11 beschrieben ist (siehe Abb. 4-50).

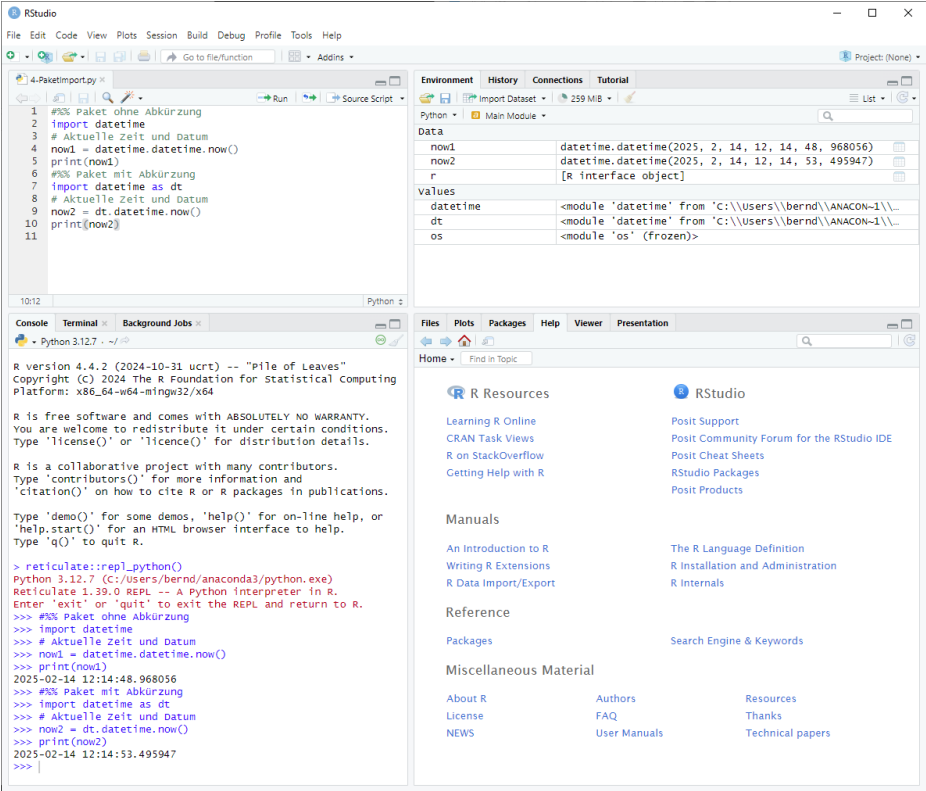


Abb. 4-50: Aktivierung und Ausführung von Paket in RStudio

4.9 Python in Anaconda Prompt ausführen

Wenn Sie in Spyder eine Python-Skriptdatei mit dem Namen 5-Test-Scatter.py speichern (Python-Code siehe Abb. 4-51), dann lässt sich dieses Skript anschließend entweder in Spyder ausführen oder alternativ auch aus der Kommandozeile über den Anaconda Prompt (siehe Abb. 4-6).

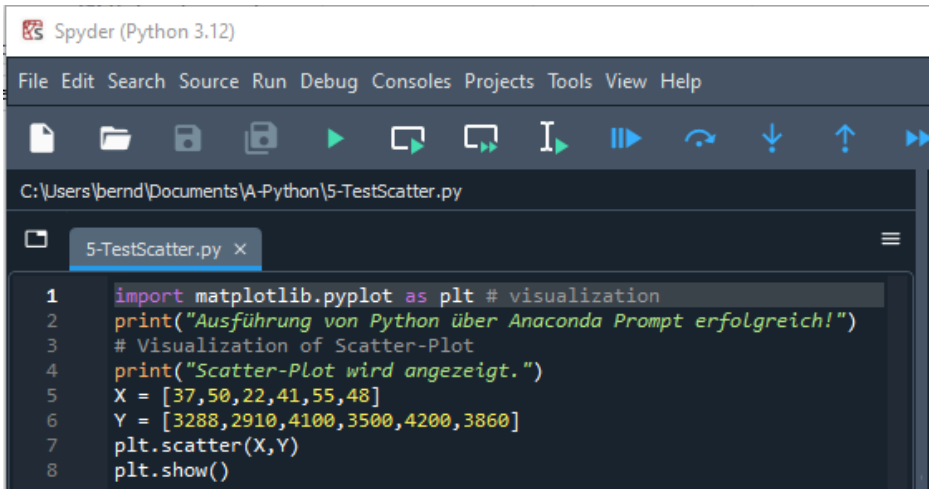


Abb. 4-51: Python Skript

Der Pfad, in dem das Python Skript gespeichert wurde, wird in Spyder angezeigt. Alternativ kann im Datei-Explorer die Skriptdatei markiert werden und mit der linken Maustaste lassen sich dann die Eigenschaften anzeigen (siehe Abb. 4-52).

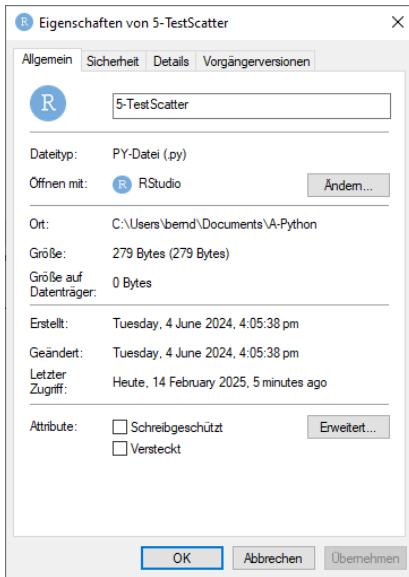


Abb. 4-52: Pfadangabe der Python Skriptdatei

Die Angabe des Ortes (Verzeichnis) muss dann korrekt in die folgende Anweisung übernommen werden. In der Kommandozeile des Anaconda Prompt wird die Funktion `python Documents/A-Python/5-TestScatter.py` ausgeführt (siehe Abb. 4-53).

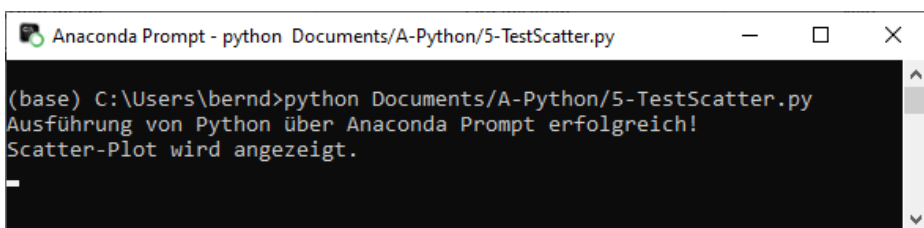


Abb. 4-53: Anaconda Prompt Ausführung von Python Skript

Neben der Textausgabe erfolgt auch die Ausgabe des Scatter-Plots (siehe Abb. 4-54).

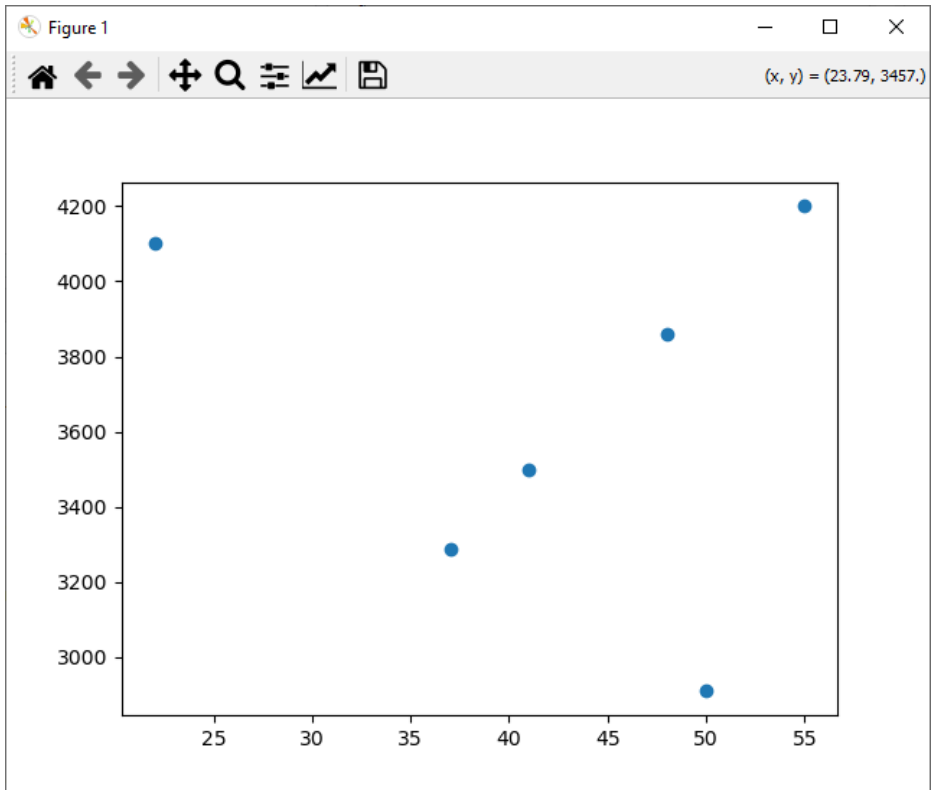


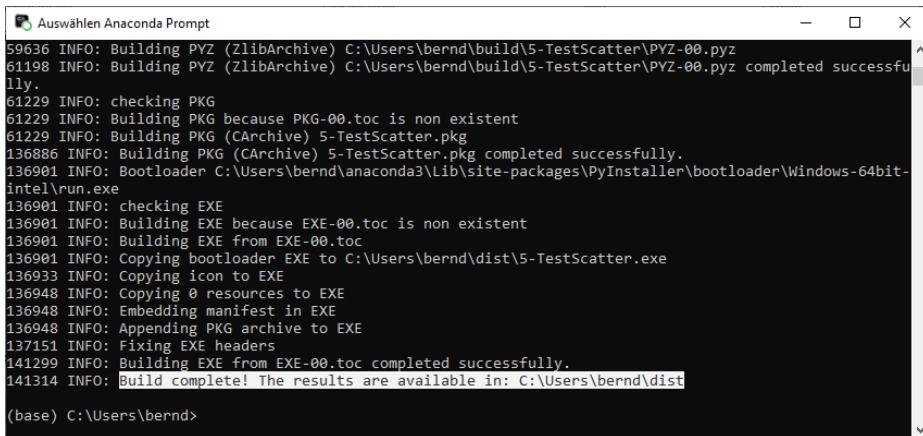
Abb. 4-54: Scatter-Plot

So lässt sich Python-Code auch ohne den Aufruf einer Entwicklungsumgebung wie Spyder ausführen.

4.10 Python als Executable in Windows Prompt ausführen

Um eine Python Skriptdatei in eine ausführbare .exe-Datei zu konvertieren kann das Paket **pyinstaller** verwendet werden. Dazu sollten Sie zunächst im Anaconda Prompt die Anweisung `pip install pyinstaller` ausführen. Anschließend kann die im letzten Abschnitt erstellte Python Skriptdatei mit der Anweisung `pyinstaller --onefile Documents/A-`

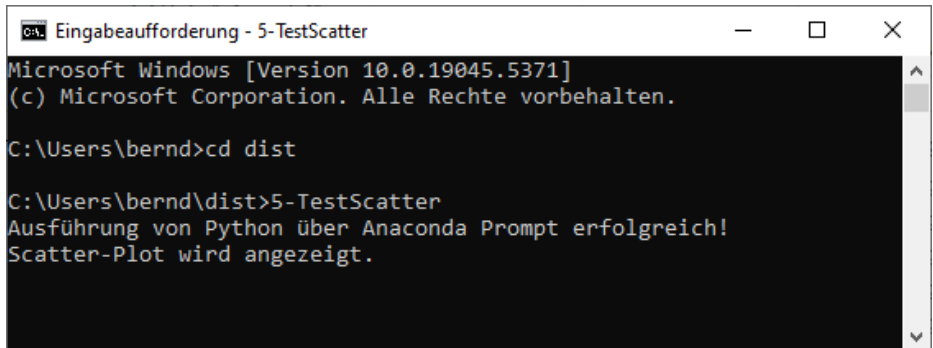
Python/5-TestScatter.py (der Parameter onefile gibt an, dass wir nur eine Datei umwandeln wollen) in eine .exe-Datei konvertieren. Wenn nach einer Weile die Meldung „Building EXE from EXE-00.toc completed successfully“ erscheint, dann war die Umwandlung erfolgreich. Die ausführbare Datei wird in dem Verzeichnis dist abgelegt, welches auch im Anaconda Prompt angezeigt wird (siehe Abb. 4-55).



```
Auswählen Anaconda Prompt
59636 INFO: Building PYZ (ZlibArchive) C:\Users\bernd\build\5-TestScatter\PYZ-00.pyz
61198 INFO: Building PYZ (ZlibArchive) C:\Users\bernd\build\5-TestScatter\PYZ-00.pyz completed successfully.
61229 INFO: checking PKG
61229 INFO: Building PKG because PKG-00.toc is non existent
61229 INFO: Building PKG (CArchive) 5-TestScatter.pkg
136886 INFO: Building PKG (CArchive) 5-TestScatter.pkg completed successfully.
136901 INFO: Bootloader C:\Users\bernd\anaconda3\Lib\site-packages\PyInstaller\bootloader\Windows-64bit-intel\run.exe
136901 INFO: checking EXE
136901 INFO: Building EXE because EXE-00.toc is non existent
136901 INFO: Building EXE from EXE-00.toc
136901 INFO: Copying bootloader EXE to C:\Users\bernd\dist\5-TestScatter.exe
136933 INFO: Copying icon to EXE
136948 INFO: Copying 0 resources to EXE
136948 INFO: Embedding manifest in EXE
136948 INFO: Appending PKG archive to EXE
137151 INFO: Fixing EXE headers
141299 INFO: Building EXE from EXE-00.toc completed successfully.
141314 INFO: Build complete! The results are available in: C:\Users\bernd\dist
(base) C:\Users\bernd>
```

Abb. 4-55: Anaconda Prompt

Die ausführbare Datei lässt sich jetzt aus diesem Verzeichnis auch in ein anderes Verzeichnis verschieben. Dort, wo die Datei abgelegt ist, kann sie dann jederzeit über die Windows Kommandozeile ausgeführt werden (siehe Abb. 4-56).



```
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\bernd>cd dist

C:\Users\bernd\dist>5-TestScatter
Ausführung von Python über Anaconda Prompt erfolgreich!
Scatter-Plot wird angezeigt.
```

Abb. 4-56: Ausführung von TestScatter.exe-Datei

Nach dem Schließen des Abbildungsfensters ist das Fenster der Eingabeaufforderung bereit für weitere Kommandoeingaben.

4.11 Python in RStudio ausführen

Um mit RStudio auch Python-Anweisungen ausführen zu können gilt es in RStudio über das Menü „Tools/Global Options/Python“ mit „Select“ die installierte Python-Version zu selektieren (siehe Abb. 4-57) und anschließend mit „Apply“ zu bestätigen.

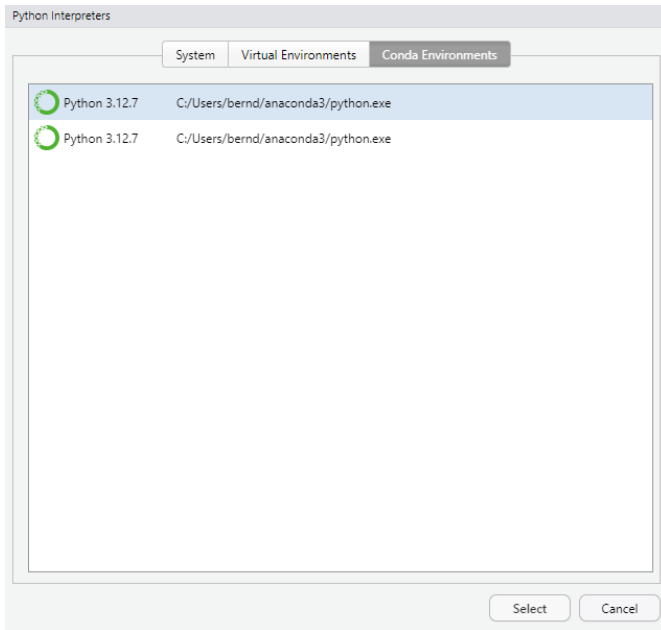


Abb. 4-57: Python-Pfad in RStudio

Darüber hinaus muss das R-Paket **reticulate** installiert und aktiviert werden.

Ausführung von Python-Code in RStudio vorbereiten

```
#install.packages("reticulate")    # Installation Package (einmalig)
library(reticulate)               # Aktivieren Package (bei jedem R-Start)
```

Um Python-Code in R-Studio auszuführen, muss zunächst Python als Skriptsprache ausgewählt werden. Dies erfolgt über das Menü mit „File/New File/Python Script“. Der dort eingefügte Python-Code kann dann wie gewohnt mit Run ausgeführt werden.

Python-Code in RStudio ausführen

```
zahlenliste = list(range(1,30,3))
print(zahlenliste)
## [1, 4, 7, 10, 13, 16, 19, 22, 25, 28]
```

Die neu erstellte Variable wird im Environment unter Main Module angezeigt und die Ausgabe erfolgt in der Konsole (siehe Abb. 4-58).

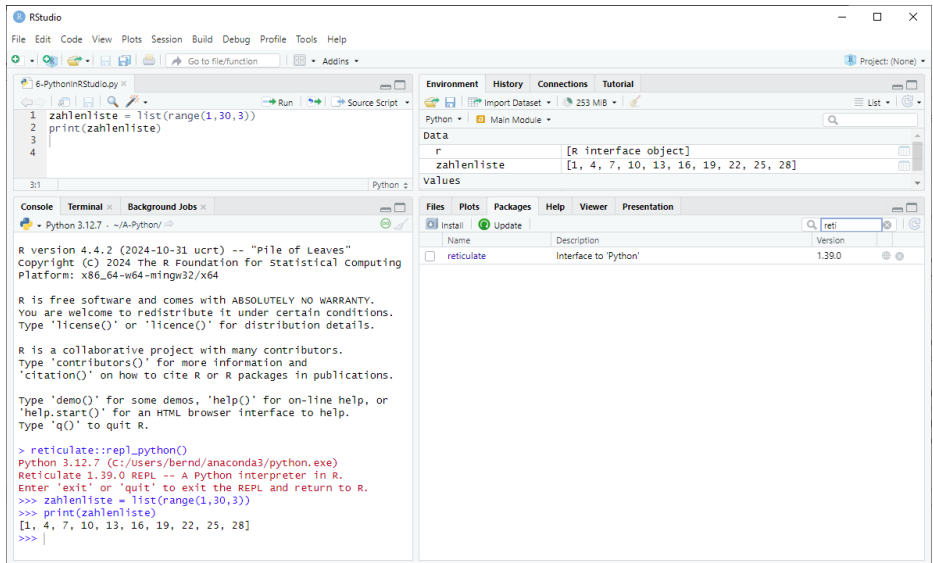


Abb. 4-58: Python-Code in RStudio

4.12 Paket pythonforbusiness

Die hier im Buch verwendeten Datasets, sofern nicht explizit auf andere Pakete verwiesen wird, befinden sich in dem von mir erstellten R-Paket **pythonforbusiness**. Sie können dieses Paket mit den folgenden Anweisungszeilen in RStudio installieren:

- `install.packages("devtools")`
- `install.packages("learnr")`
- `install.packages("reticulate")`
- `library(reticulate)`
- `reticulate::py_install("pandas")`
- `reticulate::py_require("pandas")`
- `py_module_available("pandas")` # Sollte TRUE anzeigen
- `library(learnr)`
- `install.packages("devtools")`

- `library(devtools)`
- `devtools::install_github("bheesen/pythonforbusiness",force=TRUE)`
- `library(pythonforbusiness)`

Nach erfolgreicher Installation können Sie die enthaltenen Daten und Funktionen im Ausschnitt „Global Environment“ nach Auswahl des Pakets „package pythonforbusiness“ anzeigen lassen und mit diesen arbeiten (siehe Abb. 4-59).

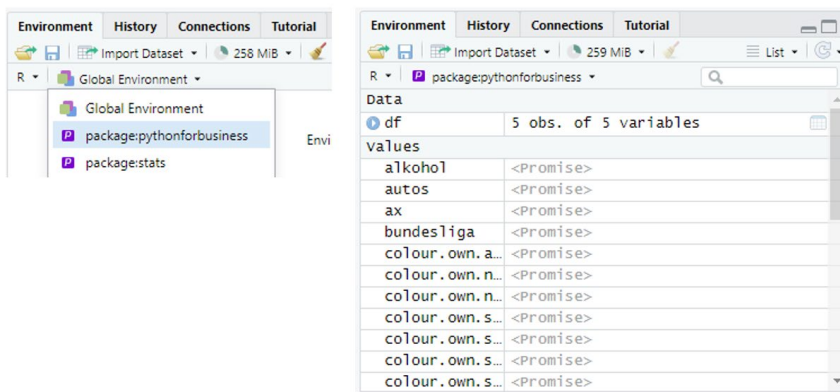


Abb. 4-59: R-Paket pythonforbusiness

Begleitend zu den Inhalten des Buches sind auch Tutorials zum Erlernen der Programmiersprache Python in dem R-Paket **pythonforbusiness** enthalten. Sie können die Tutorials, die mit dem Paket **learnr** (Schloerke, 2024) erstellt wurden, mit den folgenden Anweisungszeilen in RStudio starten:

- `ml.tutorial(name = "py.syntax")`
- `ml.tutorial(name = "py.datenstrukturen")`
- `ml.tutorial(name = "py.operationen")`
- `ml.tutorial(name = "py.kontrollstrukturen")`
- `ml.tutorial(name = "py.funktionen")`
- `ml.tutorial(name = "py.standardfunktionen")`

Wenn Sie ein Tutorial starten, dann öffnet sich ein Browser-Fenster, welches Ihnen interaktiv Inhalte präsentiert und auch die Ausführung von Anweisungen unterstützt (siehe Abb. 4-60).

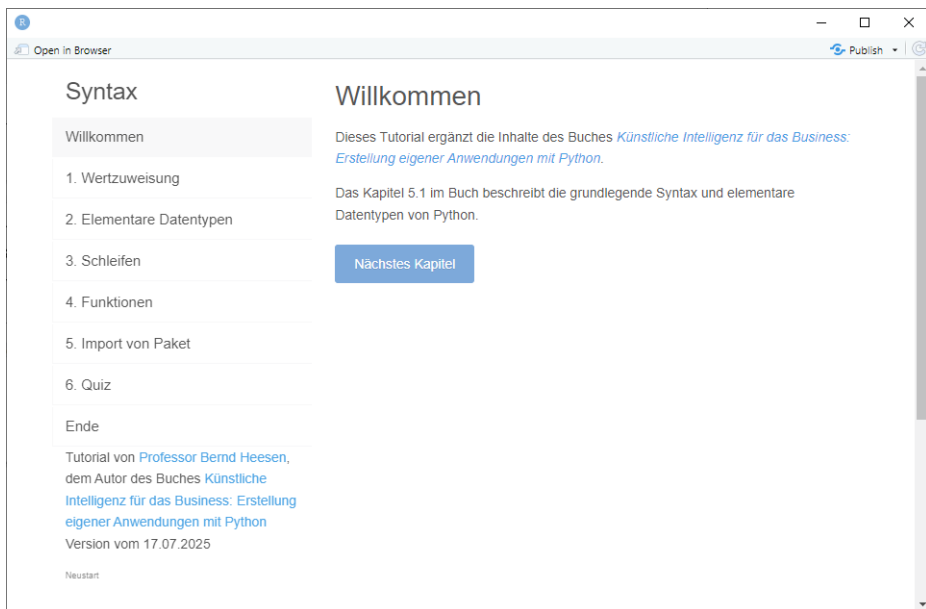


Abb. 4-60: Python Tutorial aus dem R-Paket `pythonforbusiness`

Literaturverzeichnis

- Anaconda. (2025). Abgerufen am 12. Februar 2025 von <https://www.anaconda.com/download>
- Posit. (2024a). *Tidymodels*. Abgerufen am 02. April 2024 von <https://www.tidymodels.org/>
- Posit. (2024b). *Tidyverse*. Abgerufen am 02. April 2024 von <https://www.tidyverse.org/>
- Posit. (2025). *RStudio IDE: The most trusted IDE for open source data science*. Abgerufen am 12. Februar 2025 von <https://posit.co/products/open-source/rstudio>
- Python Software Foundation. (2025). *Python*. Abgerufen am 18. Mai 2025 von <https://www.python.org>
- R CRAN Team. (2024). *RTools: Toolchains for building R and R packages from source on Windows*. Abgerufen am 24. March 2024 von <https://cran.r-project.org/bin/windows/Rtools>
- R Foundation. (2025). *The R Project for Statistical Computing*. Abgerufen am 12. Februar 2025 von <https://www.r-project.org>
- Schloerke, B. (2024). *Learnr*. Abgerufen am 24. March 2024 von Learnr: <https://rstudio.github.io/learnr>
- Spyder. (2024). *Spyder: The scientific Python development environment*. Abgerufen am 23. March 2024 von <https://www.spyder-ide.org/>
- Spyder. (2025). *Welcome to Spyder's Documentation*. Abgerufen am 12. Februar 2025 von <https://docs.spyder-ide.org/current/index.html>