INTRODUCTION

Jan Kinne

Web Data Science

INSTRUCTOR JAN KINNE

- PhD in Geographic Data Science
- Co-founder of start-up ISTARI.AI
- Part-time researcher at ZEW Centre for European Economic Research in Germany
- Python user since 2013

WEB DATA SCIENCE

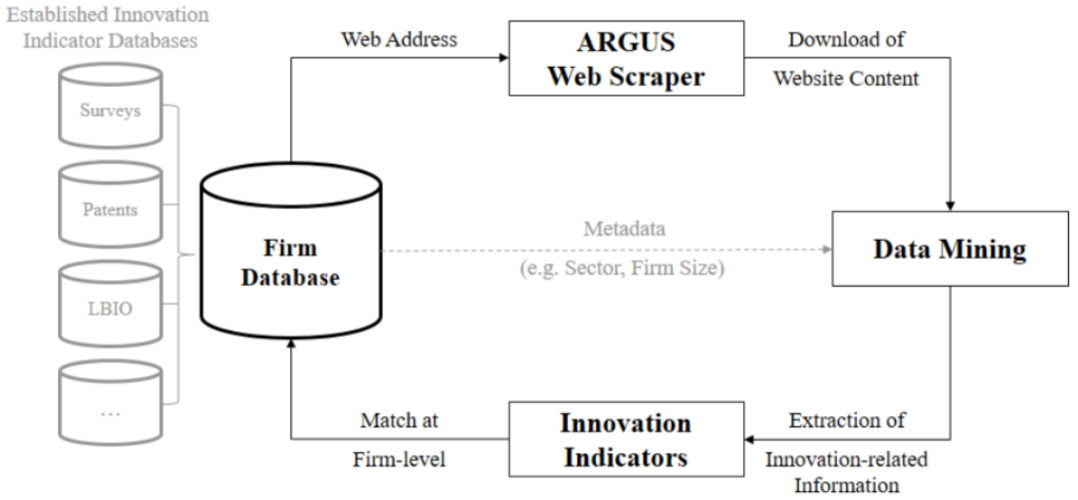
WEB DATA SCIENCE AKA WEB MINING

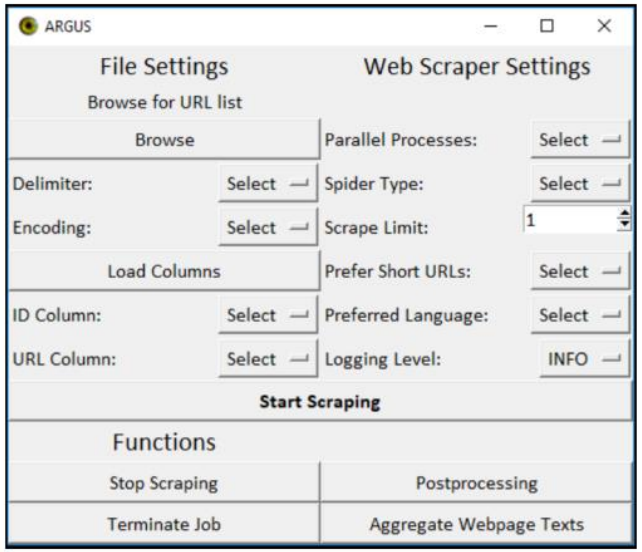
- Using **Data Science/Data Mining** tools in combination with **Web data**.

> *"'We are living in the information age' is a popular saying; however, we are actually living in the data age. [...] An explosively growing, widely available, and gigantic body of data makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining. Data mining has and will continue to make great strides in our journey from the data age toward the coming information age."*
>
> from "Data Mining: Concepts and Techniques", the first data mining book I read.

A FRAMEWORK FOR WEB SCRAPING FIRM WEBSITES

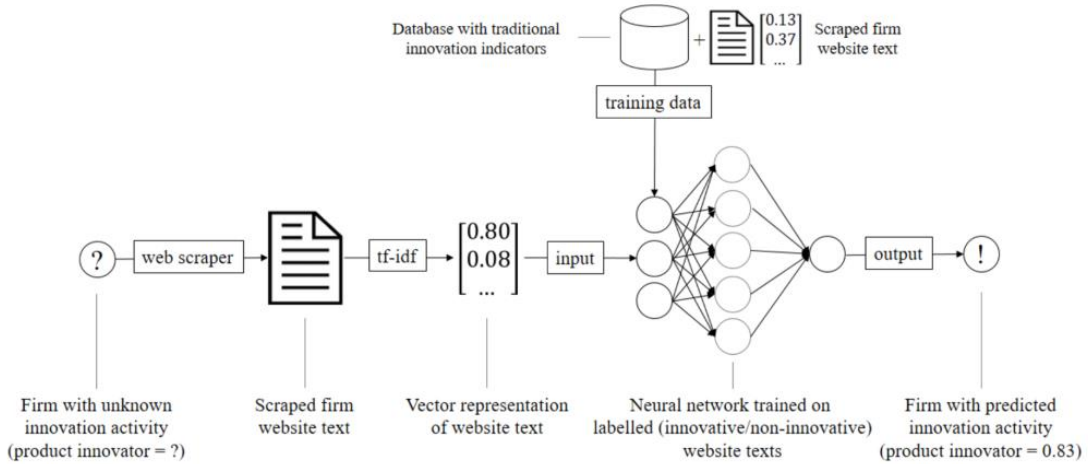
- Proposed a framework for **web-based innovation indicators** as an alternative to traditional indicators from patent or survey data.
- Developed the ARGUS **open source Python Scrapy based web scraper** for broad crawls.
- Investigated the population of firms with own website.
- https://doi.org/10.1007/s11192-020-03726-9

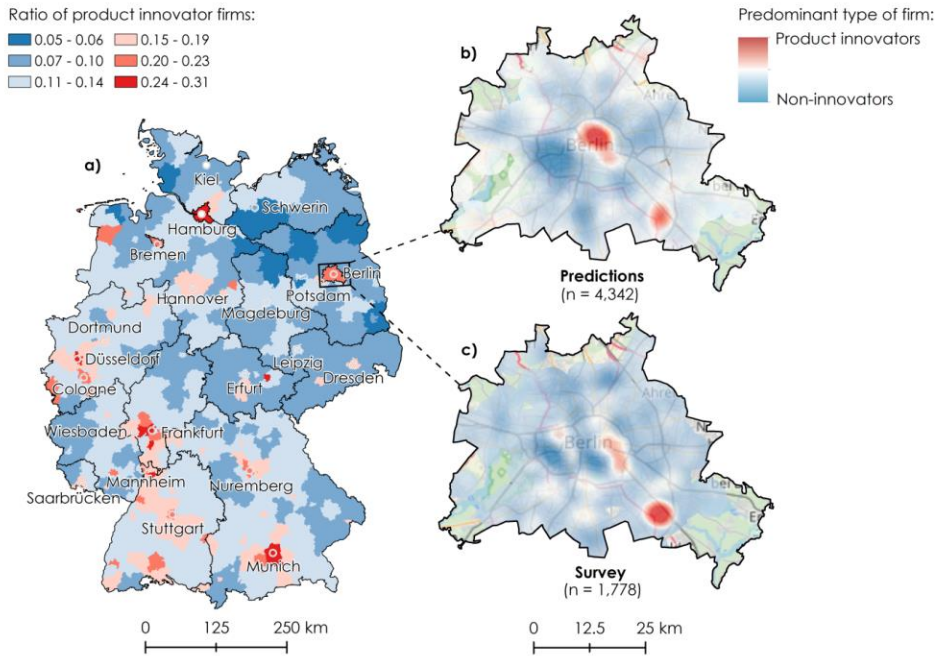




A WEB-BASED INNOVATION INDICATOR

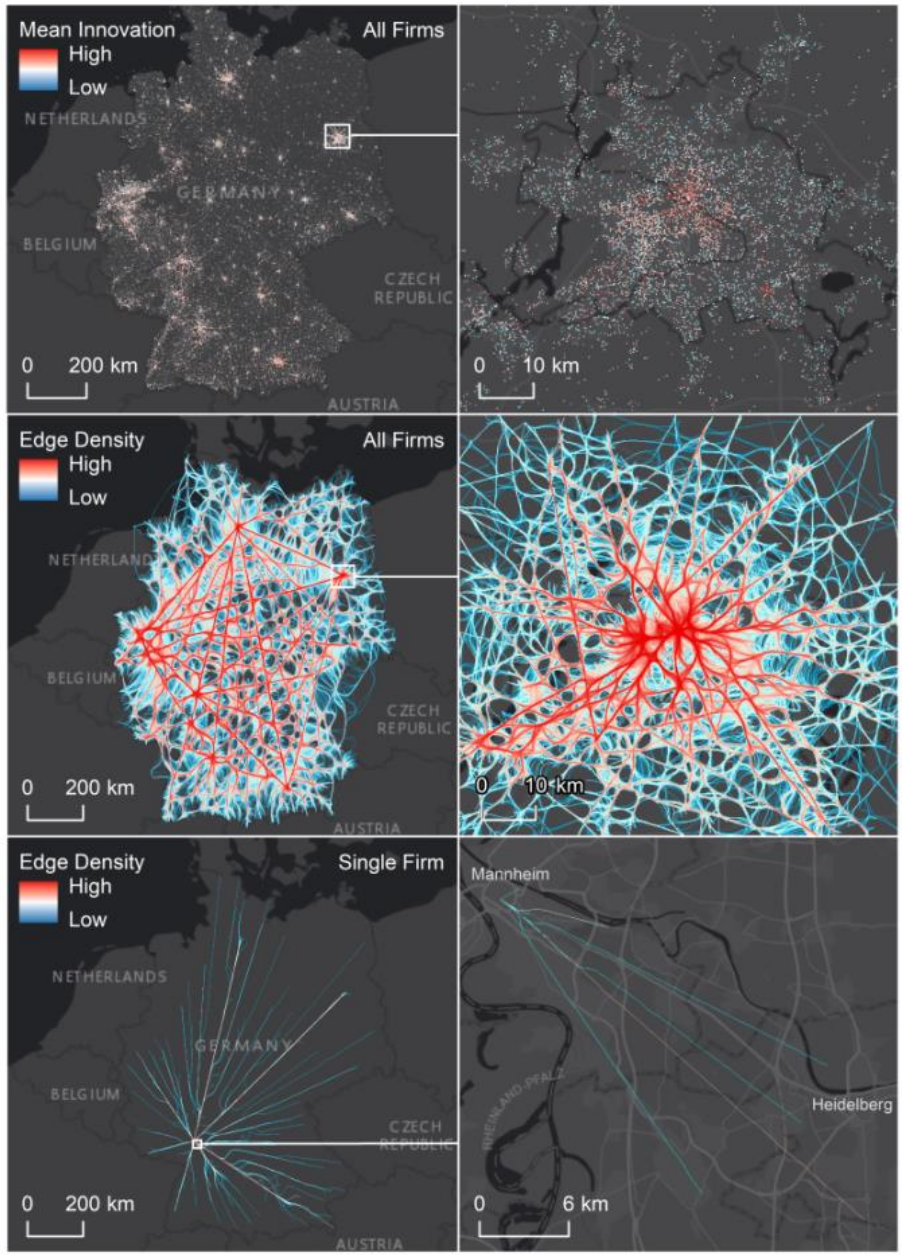
- Developed a web-based innovation indicator **based on company website texts**.
- **Machine learning model trained with** the website texts of firms that participated in a **traditional innovation survey**.
- Compared results to traditional innovation indicators.
- https://doi.org/10.1371/journal.pone.0249071





THE DIGITAL LAYER

- Used **hyperlinks** between firm websites to map **company networks**.
- Compared **linkages of innovative and non-innovative companies** in the Digital Layer.
- http://ftp.zew.de/pub/zew-docs/dp/dp20003.pdf



THE COURSE

AIM OF THE COURSE

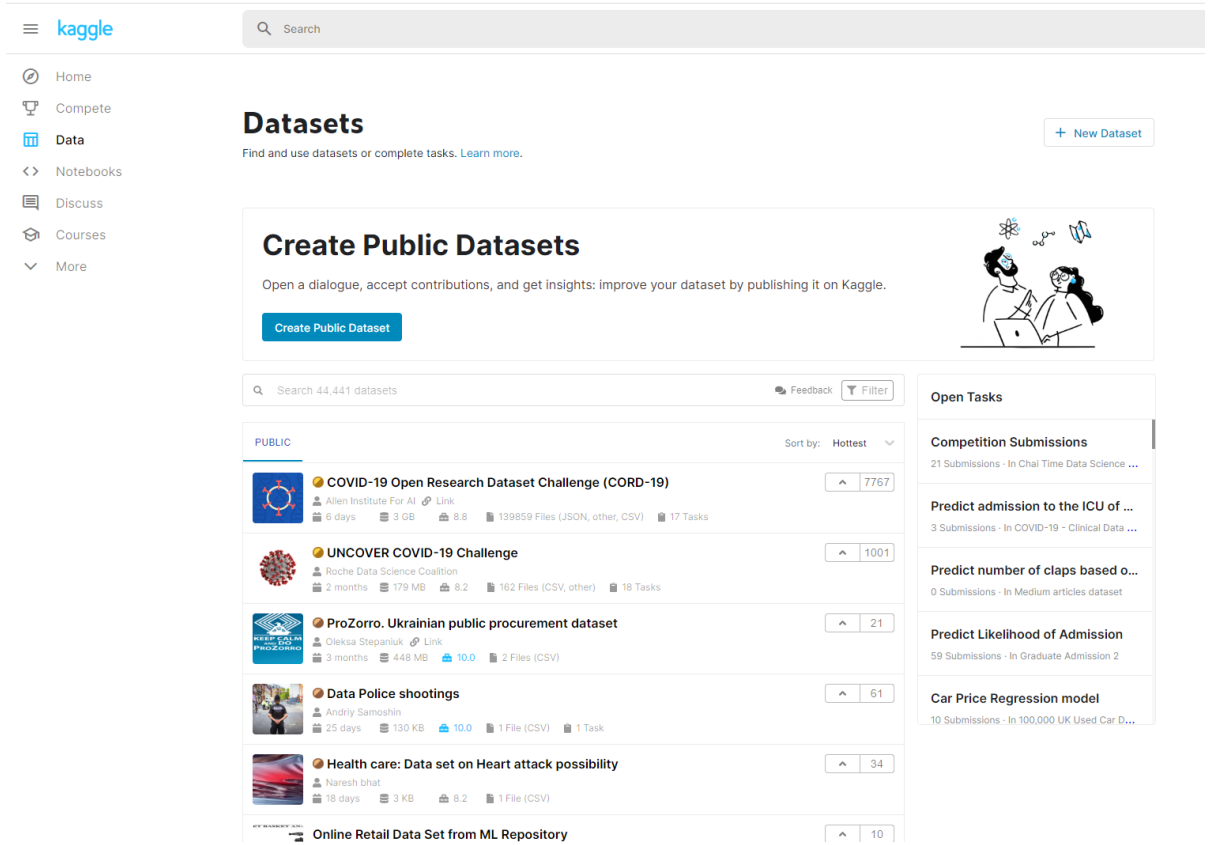
- **Hands on** introduction to **Python**.
- Focus on **web data analysis**, not on classical programming.
- Overview of the **many fields of application** from data manipulation and analysis to visualization to web scraping.
- Mix of **self-study/learning-by-doing** and presentations with live coding.
- Creating a **solid basis for further self-study**.
- Give you as many **practical suggestions** as possible.

AGENDA

- Day 01 – Introduction to Python I
  - Jupyter Notebooks
  - Python basics
  - Loading and handling data with Pandas
  - Descriptive statistics
  - Data plotting
- Day 02 – Introduction to Python II
  - More Pandas
  - More plotting
- Day 03 – Mapping
  - Mapping with Geopandas
  - Geodata basics
  - Geocoding
- Day 04 – Web Scraping
  - HTML basics
  - Web crawling and web scraping
- Day 05 – Text Mining
  - Machine learning basics
  - Building a text mining pipeline
- Day 06 – Student Project Presentations

STUDENT PROJECTS

- Students will:
  - conduct an independent **Exploratory Data Analysis** (EDA).
  - choose one or several datasets from the **Kaggle** open database.
  - apply the learned skills to explore the dataset.
  - **present and discuss intermediate results** at the beginning of day 02 to day 05.
  - **present their final results** on day 06.
- Grading will be based on these projects.



STUDENT PROJECTS

- Your quest 💬:
  Your **supervisor** asks you to **explore a data set**. A week later, your supervisor wants to see your results. Present your results in the form of a **clearly formatted, commented and structured Jupyter Notebook**. Make sure your supervisor understands your main working steps and how you achieved your results, without getting lost in unnecessary detail.

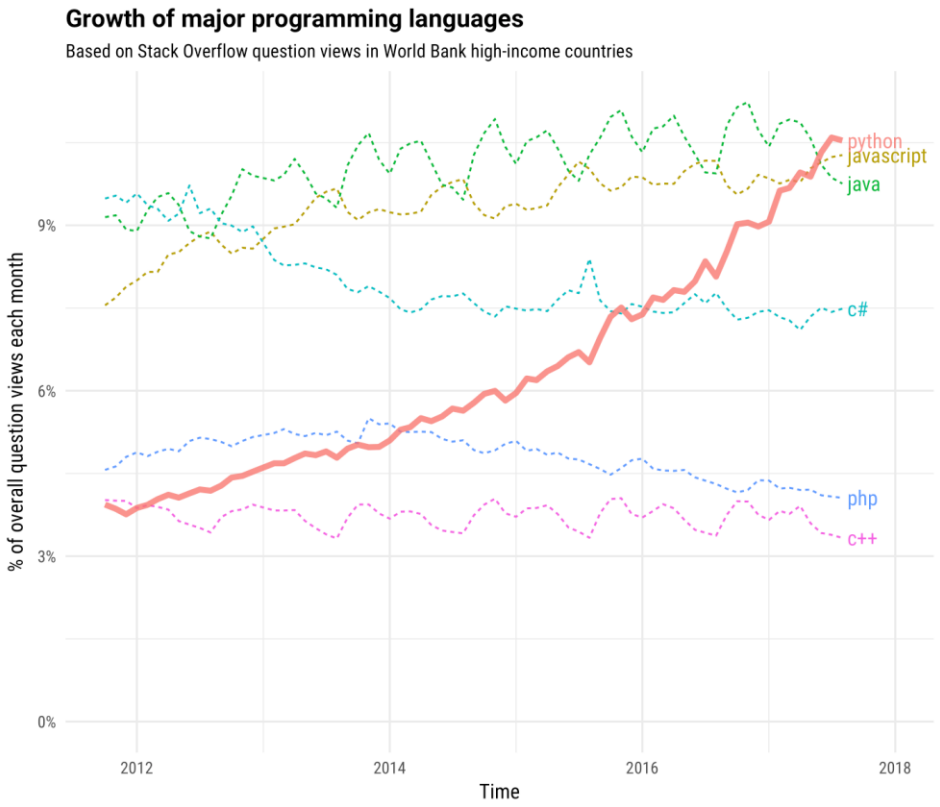
- Rewards 💯:
  - max 1XP: Loading and preprocessing data (e.g. fixing wrong data types). Data exploration with descriptive statistics and plots.
  - max 1XP: Scraping a Web data source of your choice.
  - max 1XP: Mapping your dataset.
  - max 1XP: Apply a machine learning model to your data.
  - max 1XP: Executive summary with the main findings of your data science project and presentation (10 mins) day 06.
- 6 – **XP** = Grade 

THE TOOLS

python™

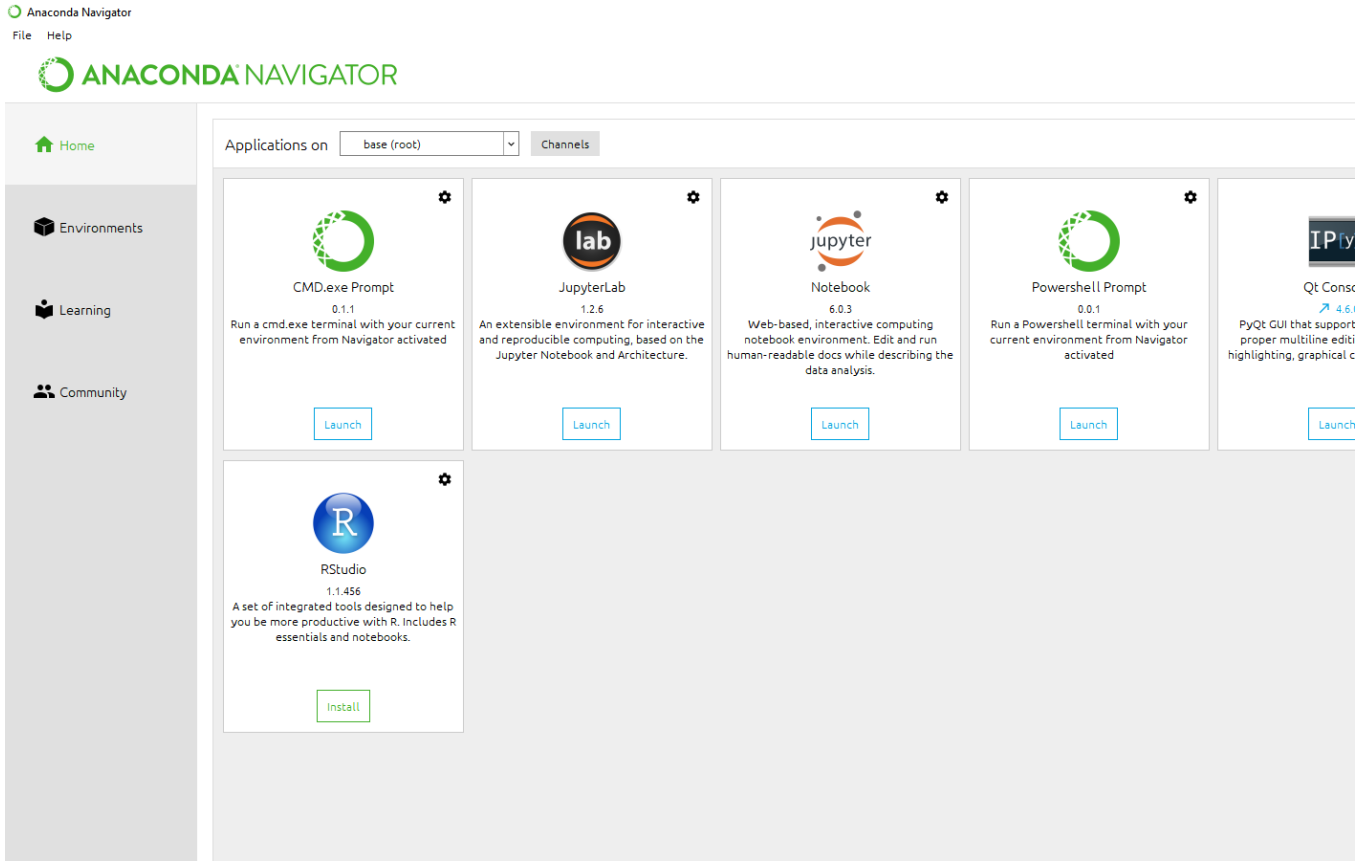
- **High-level**, general-purpose programming language.
- High code **readability**.
- Great for quick **prototyping**.
- Over 200,000 **packages** for extended functionality.
- Great for **consistent data science workflows** from data acquisition to data analysis to visualization.
- Very high and increasing **popularity**.

```
[1]: print("Hello world!")

Hello world!

[2]: names = ["Gandalf", "Frodo", "Arwen"]

for name in names:
    print(name, "is a character from 'The Lord of the Rings'")

Gandalf is a character from 'The Lord of the Rings'
Frodo is a character from 'The Lord of the Rings'
Arwen is a character from 'The Lord of the Rings'
```



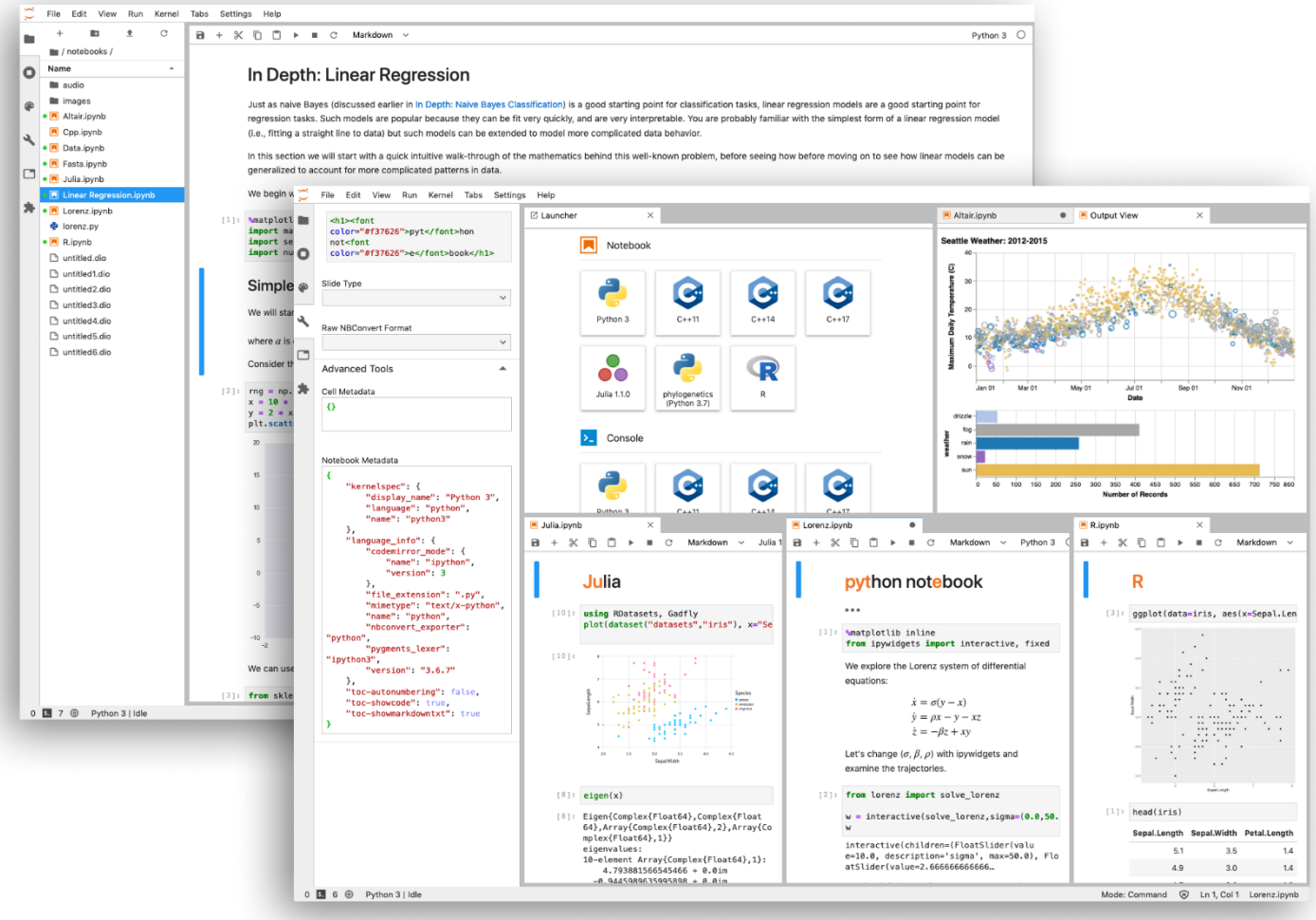
ANACONDA PYTHON

- Free and open-source **distribution** of Python.
- Aims to **simplify package management**.
- Includes the most important Python data science **packages out-of-the-box**.
- Available for Windows, Linux, and Mac.
- Includes the Anaconda Navigator **graphical user interface** for launching applications and manage packages and environments.



JUPYTER NOTEBOOK

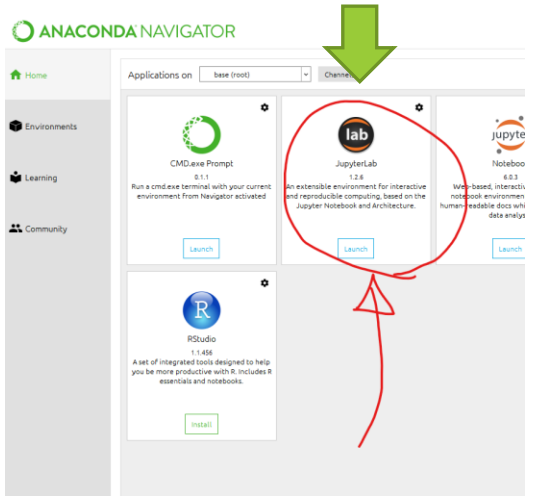
- Open-source, web-based and **interactive application**.
- Create and share **documents that contain code, equations, visualizations, and text**.
- Perfect for **Exploratory Data Analysis**.
- Great for **structuring the workflow** of a whole data science project.
- Works with over 40 programming languages.
- We will use **JupyterLab** as our interface to work with Jupyter notebooks.

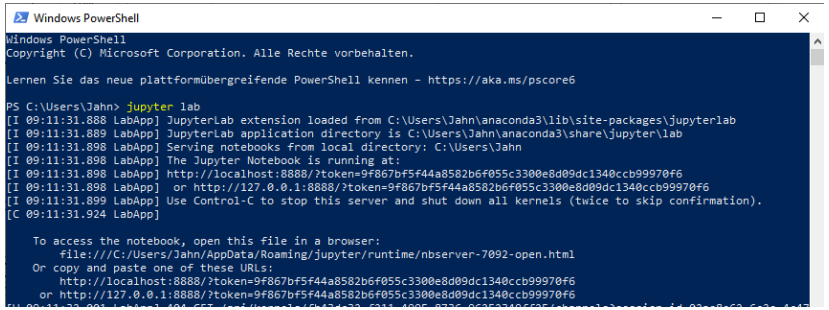


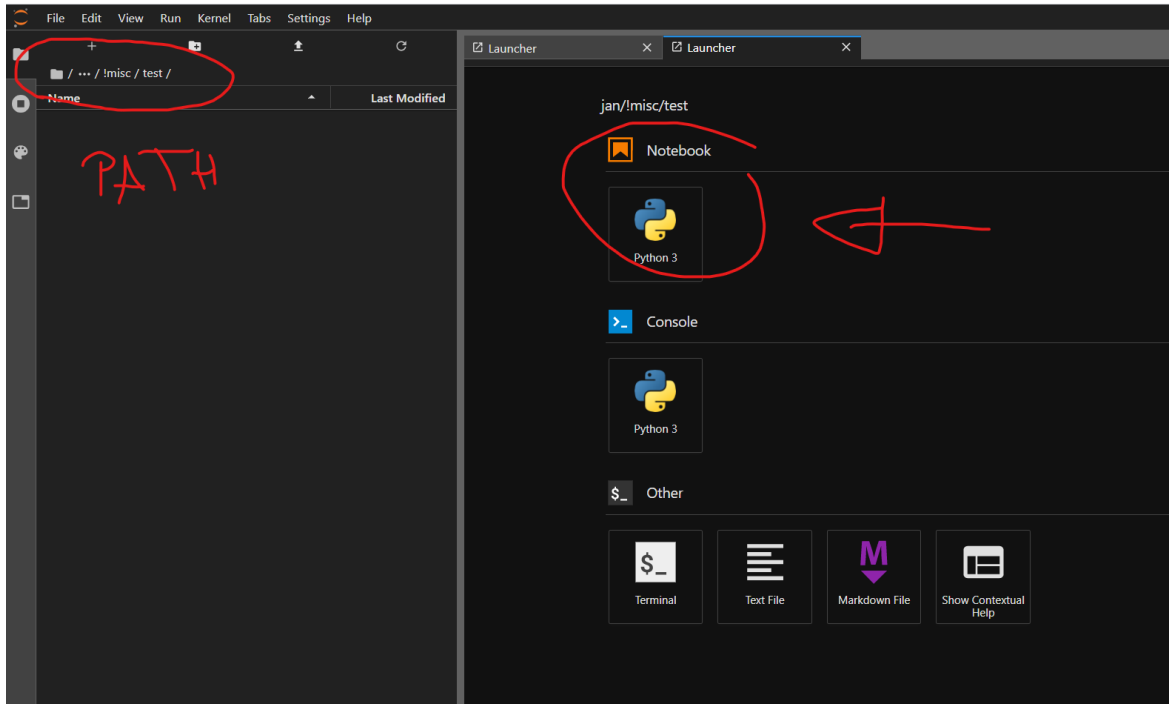
STARTING JUPYTER LAB

- Launch from **Anaconda Navigator**
- Launch from shell/**command prompt**:
  - Launch Windows PowerShell, Linux Shell, or macOS Terminal
  - **Navigate to directory**
  - Type **jupyter lab** and hit enter
- JupyterLab server will be started
- Default browser opens up
- If not: use address shown in JupyterLab server window

LAB (!) not Notebook!

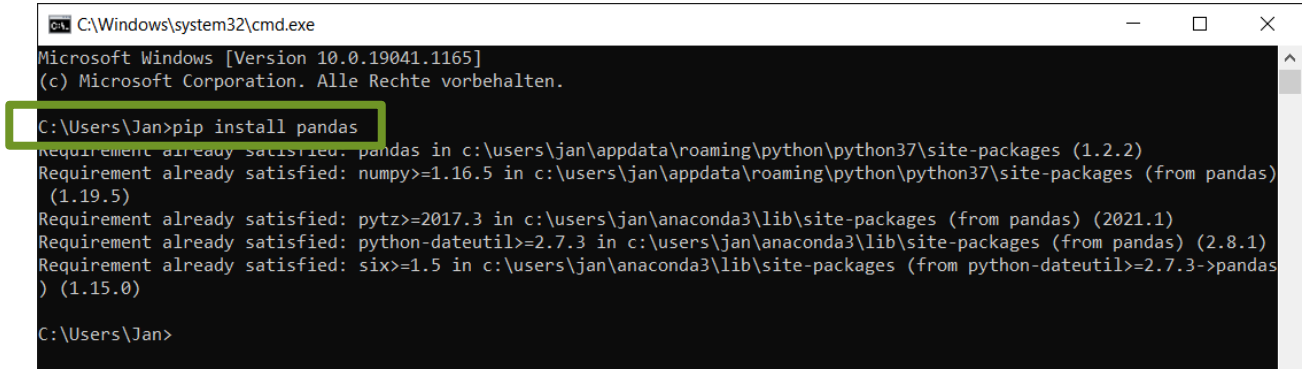






INSTALL PACKAGES

- Install additional Python packages using **pip** or **conda** package installers.
  - `pip install pandas`
  - `conda install pandas`
- Installs packages from **online repositories**.
- **Installs all dependencies** (other needed packages) automatically.
- Allows for down- and upgrading of packages.
- Conda can be used to maintain different **environments** (i.e. Python installations with packages).



GET HELP

- Use Google and include the Python package you are working with in the search string.
- Look for **stackoverflow** search results (usually top hit).
- Check out the **top answers** and the correct answers
  (marked with a green tick)