

Tools for graphing and A.I

Jupyter notebooks & others

Overview

- What are jupyter notebooks?
- Notebook interface
- Sample applications
- Plotting
- Other tools

What is Jupyter notebook?

- Concept based on Lab book (pen & paper)
 - procedures, data, calculations, and findings.
- Jupyter notebook (Electronic Lab book)
 - software code, computational output, explanatory text, and rich content in a single environment.
 - Supports many programming languages although 3 most used are python, R and Julia.
 - Can export to many formats: HTML, LaTeX, PDF, etc..
 - Platform independent (works inside a web-browser).
 - File extension: ipynb

Obtaining jupyter notebooks

Personal & group use

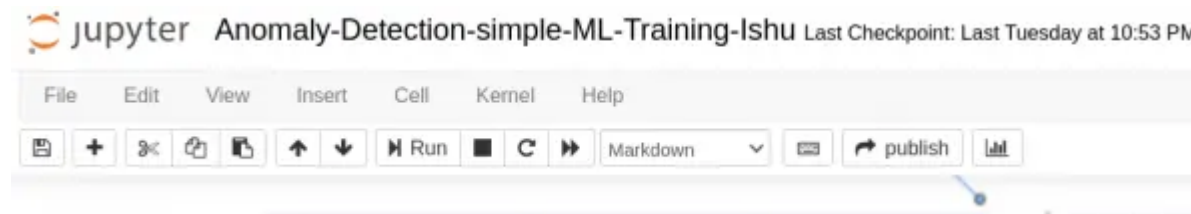
- Installing anaconda
 - <https://www.anaconda.com/download/>
- Python (and use pip to install)
- Install jupyterHUB for classroom use
 - <https://jupyter.org/hub>

Some free On-line

- WACREN
 - <https://notebooks.wacren.net/>
- Collab
 - <https://colab.research.google.com/>
- Azure
 - <https://notebooks.azure.com>
- Others
 - <https://github.com/jupyter/tmpnb>

Jupyter Interface Menu

- **File:** New (add notebooks, etc), Save, upload data and notebooks
- **Edit:** Operates on cells: add, delete, move, paste, etc
- **View:** Syntax highlighting, etc..
- **Kernel:** Internal engine or application: python3, anaconda, gnuplot, etc
- **Help:** Context sensitive help, able to provide help on functions and *imported modules (matplotlib, pandas, etc)*



Pandas and time series CSV data files

- Import pandas as pd
- Read CSV data of 1st march 2024
 - `df = pd.read_csv('240301.csv', parse_dates=['Local_Time'], date_format='%Y-%m-%d %H:%M:%S', index_col='Local_Time')`
 - *With the `read_csv` function, you can specify which column to use as index and how to interpret dates..*
- Read CSV data of 1nd of march 2024
 - `df2 = pd.read_csv('240302.csv', parse_dates=['Local_Time'], date_format='%Y-%m-%d %H:%M:%S', index_col='Local_Time')`
- Merge both days into one dataframe
 - `df3 = pd.merge(df,df2, left_index=True, right_index=True)`

Working with Dataframes

- Obtaining information about your dataframe
 - `df.info()`
- Showing first 5 entries
 - `df.head()`
- Showing last 5 entries
 - `df.tail()`
- Calculating standard deviation and mean for all columns in dataframe
 - `df.std()`, `df.mean()`
- Functions can also be applied per column (*NOTE: PH_value is column name*)
 - `df['PH_value'].std()`
 - `df['PH_value'].mean()`

Simple plot using matplotlib

- Import modules and create a figure handle named ax
 - `import matplotlib.pyplot as plt`
 - `import matplotlib.ticker as ticker`
 - `fig, ax = plt.subplots()`
- Plot a variable as a line (NOTE: issue multiple plots statement for each variable)
 - `ax.plot(df3.PH_calc)`
- Plot without axis labels
 - `ax.set_xticks([])`
 - `ax.set_yticks([])`
- OR plot with minimal axis labels
 - `ax.xaxis.set_major_locator(ticker.MultipleLocator(400))`
 - `ax.yaxis.set_major_locator(ticker.MultipleLocator(50))`
 - `ax.tick_params(axis='x', labelrotation=90)`

Hands-on tasks

- Plotting multiple variables on same graph
 - Use a single plot statement with different variables or
 - Use separate plot statements for each variable
- Hands-on tasks
 - Compare PH values for 2 separate days
 - *Hint: load the datafiles into 2 separate dataframes*
 - Compare PH values for every Tuesday in a month
 - *Hint: load 4 data files into 4 different data files*
 - Plot only the first 6 hours of a chosen day
 - *Hint: truncate dataframe*
 - Plot between time range 12:00 and 18:00
 - *Hint: subslice dataframe*

From data to A.I

- Jupyter notebooks can be used to combine data, software code, explanatory notes, graphs, etc together.
- Additional modules that are useful for A.I tasks include
 - tensorflow, keras, neuralnetwork, pytouch, etc

Other software

- R studio (desktop application) download from <https://www.rstudio.com/>
- GNU Octave (Opensource alternative to MATLAB)
 - <https://octave-online.net/>