

The ~~ILCI JupyterHub Breeding Analytics Hub~~ **Crop Improvement Analytic Platform** is a cloud-based environment where software tools, code, documentation, and data are available in one place.

## CIAP objectives:

- Co-create a collaborative environment for working together across institutions.
- Co-design, develop, and test custom software tools across institutions.
- Provide access to data, tools and analytic pipelines.
- Provide computational resources.
- Promote reproducible data science projects.
- Provide a platform for teaching and training.

## The Hub is a managed environment:

- Software, tools, and packages are curated, installed and kept up to date.
- This includes custom R packages and web services developed specifically for ILCI researchers.
- It is dynamic: new tools and customizations can be developed and added to suit users needs.

## The Hub is a shared space for collaboration, tool development, teaching and learning:

- Everyone is on the same server and in the same environment.
- Pass and share notebooks and data across users and institutions.
- Use templates, pre-populated notebooks that serve as tutorials and writeable notebooks.
- Use the platform as a place to develop and test new tools across many users.

## Computation is done on the cloud:

- Users aren't limited by the capacity of their computer or local server.
- Tools to submit large jobs in the background.
- Directories, data, and code are easily shared with other Hub users.

## The Hub makes use of computational notebooks:

- Code and documentation are recorded in one place.
- Data can be retrieved directly from online databases into computational notebooks.
- Create reproducible data analysis projects.

# Today's outline:

1. New onboarding
2. Tools and extensions
3. BrAPI helper

This is a hands-on workshop, so please be logged in and follow along and break stuff!

# New user onboarding agenda

1. Log in
2. Launch page
3. Create and name directories
4. Create and name notebooks
5. Execute Markdown and code
6. Save and export notebooks




# Step 1 - log in

1. Navigate here:

<https://ciap.ilci.scienceversa.com/hub/login>





2. In the username field enter the email address you used for the workshop sign-up form, then type a NEW password.



The image shows a login form with an orange header bar containing the text "Sign in". Below the header, there are two input fields. The first is labeled "Username:" and the second is labeled "Password:". Both fields are empty. At the bottom of the form is an orange button with the text "Sign in" in white.

## Step 2 - Create and name directories

- myName
  - myDataProject
    - Data
    - Notebooks
    - Documents
    - Output

/ econopouly / myDataProject /	
Name	Last Modified
 Data	4 seconds ago
 Documents	next year
 Notebooks	next year
 Output	next year

# Step 3 - Create, name, and save a notebook

The screenshot displays the Jupyter Notebook interface. The top menu bar includes File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. Below the menu is a toolbar with icons for creating a new notebook, opening a file, saving, and other actions. The left sidebar shows the file explorer with a search bar and a list of files. The main area displays the notebook editor with a code cell.

File Edit View Run Kernel Git Tabs Settings Help

Filter files by name

/ ... / myDataProject / Notebooks /

Name	Last Modified
myNotebook_20240703.ipynb	6 hours ago
Untitled.ipynb	next year

Untitled.ipynb

Code

BrAPI

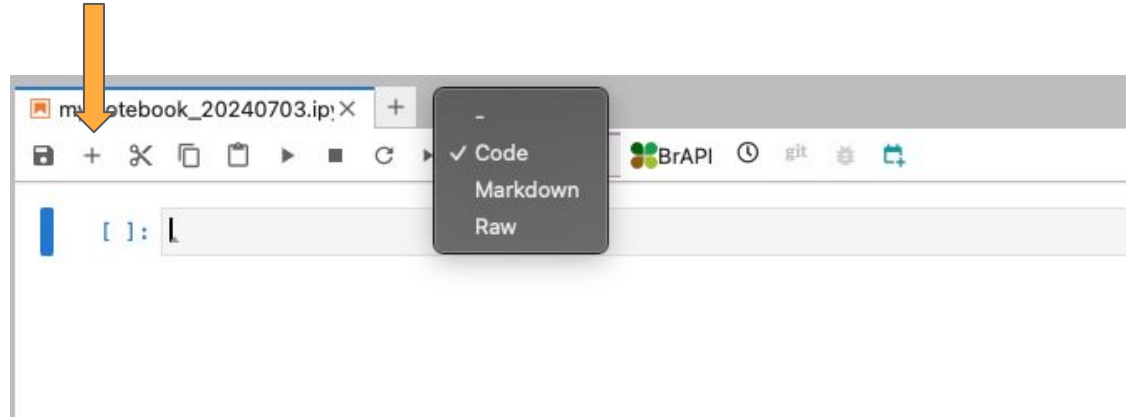
git

Notebook

[ ]:

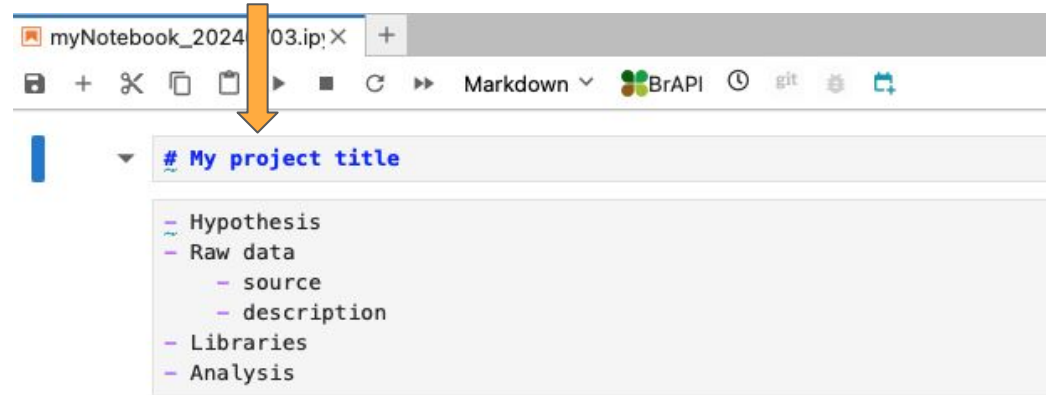
## Step 4 - Operate cells within notebooks

- A cell is an input field for text which can then be executed and interpreted.
- Notice that you can set each cell to: Code, Markdown, or Raw.
- Create additional cells with the + icon.



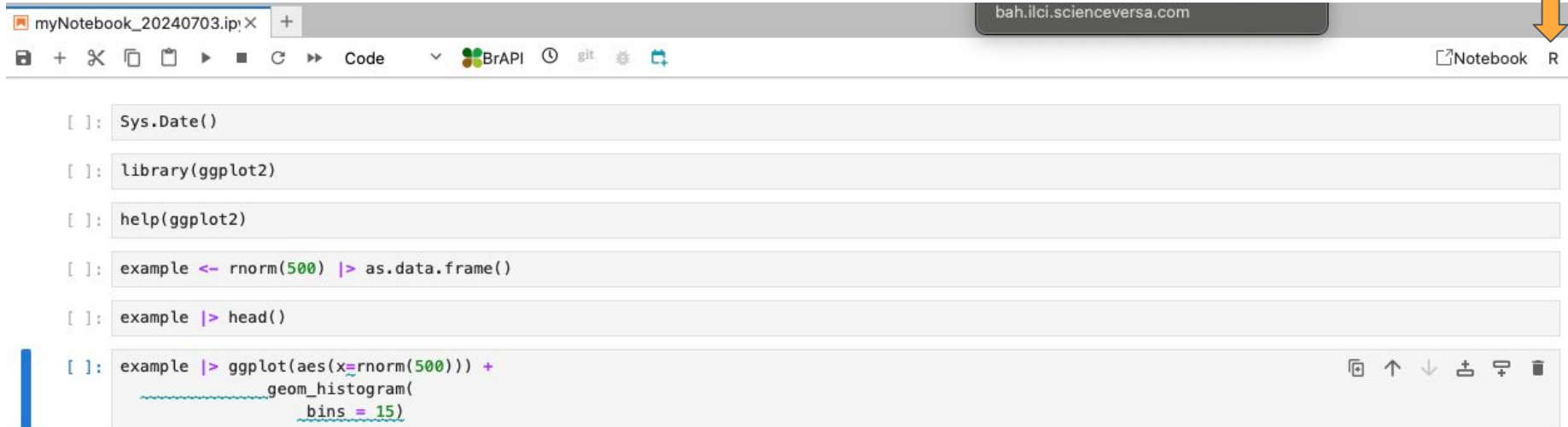
## Step 5 - Execute Markdown

- Markdown language allows you to format your text. Instead of clicking buttons to format text, Markdown syntax is used which upon execution formats your text.
- Set your cell to 'Markdown'.
- Execute with 'play' icon or shift or command + enter.



# Step 6 - Execute code

- Notebooks support coding in various languages, including R.
- Set your cell to 'Code'. Make sure your notebook is set to 'R'.
- Execute with 'play' icon or shift or command + enter.



The screenshot displays a web-based Jupyter Notebook interface. The browser's address bar shows the URL `bah.ilci.scienceversa.com`. The notebook's title bar indicates the file name `myNotebook_20240703.ipynb`. The interface includes a toolbar with icons for saving, opening, and running code. The notebook is currently set to the 'Code' mode, and the language is configured to 'R'. A vertical orange arrow on the right side of the interface points to the 'R' language selector in the top right corner. The notebook contains six code cells, each starting with `[ ]:`. The code in the cells is as follows:

```
[ ]: Sys.Date()

[ ]: library(ggplot2)

[ ]: help(ggplot2)

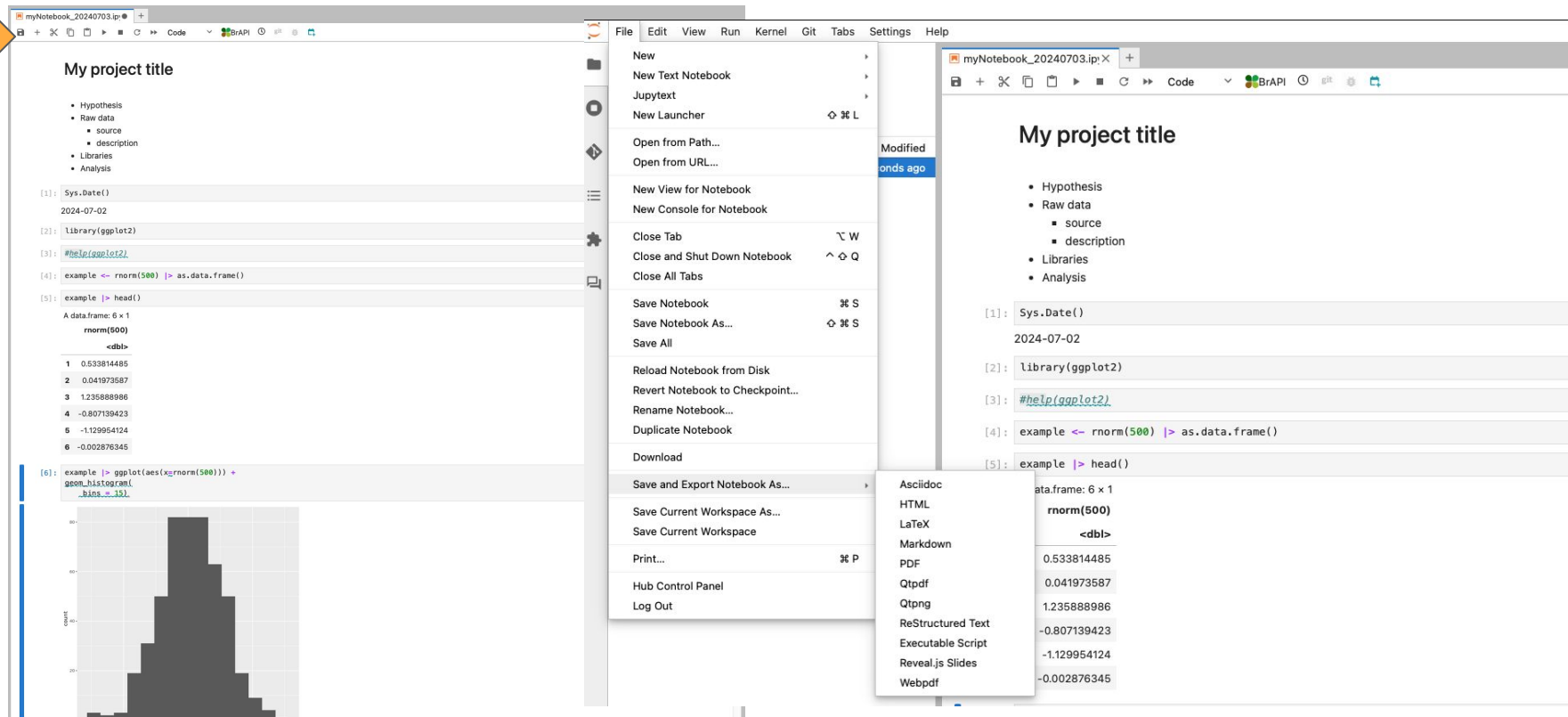
[ ]: example <- rnorm(500) |> as.data.frame()

[ ]: example |> head()

[ ]: example |> ggplot(aes(x=rnorm(500))) +  
  geom_histogram(  
    bins = 15)
```

At the bottom right of the interface, there is a toolbar with icons for running the code (a play button), undo, redo, and other standard editing functions.

# Step 7 - Save and export your notebook



The screenshot displays the JupyterLab interface with a notebook titled 'myNotebook\_20240703.ipynb'. The notebook content includes a project title, a table of contents, and a series of code cells. The first code cell shows the current date. The second cell loads the 'ggplot2' library. The third cell displays the help for 'ggplot2'. The fourth cell generates a random normal distribution and converts it to a data frame. The fifth cell displays the first few rows of the data frame. The sixth cell creates a histogram of the data.

The 'File' menu is open, showing options for creating new notebooks, opening existing ones, and saving. The 'Save and Export Notebook As...' option is highlighted, which has opened a secondary menu with the following export formats:

- AsciiDoc
- HTML
- LaTeX
- Markdown
- PDF
- Qtpdf
- Qtpng
- ReStructured Text
- Executable Script
- Reveal.js Slides
- Webpdf

The notebook content is as follows:

```
My project title
```

- Hypothesis
- Raw data
  - source
  - description
- Libraries
- Analysis

```
[1]: Sys.Date()
2024-07-02

[2]: library(ggplot2)

[3]: #help(ggplot2)

[4]: example <- rnorm(500) |> as.data.frame()

[5]: example |> head()
A data.frame: 6 x 1
  rnorm(500)
  <dbl>
1  0.533814485
2  0.041973587
3  1.235888986
4 -0.807139423
5 -1.129954124
6 -0.002876345

[6]: example |> ggplot(aes(x=rnorm(500))) +
  geom_histogram(
    bins = 33)
```

The histogram shows the distribution of the random normal data, with a peak around 0.5.

# Templates

- What are Templates?
  - Pre-designed notebooks serving as a starting point for creating new notebooks.
  - Pre-configured with code and markdown components.
  - Allows quick project initiation without setting up the basic structure.
- Benefits
  - **Consistency:** Ensures uniform layout and structure across notebooks.
  - **Efficiency:** Saves time by eliminating the need to set up the initial framework.
  - **Best Practices:** Incorporates recommended coding and documentation standards.
  - **Reusability:** Can be reused across different projects, facilitating replication of successful methodologies.
- Demo
  - <https://github.com/agostof/ILCI-NotebookTemplates>



# Job Scheduler

- Overview
  - Automates the execution of Jupyter Notebooks.
  - Facilitates scheduled tasks, automating routine analyses, reports, and data processing.
- Benefits
  - **Reliability:** Execute code and notebooks without consistent internet connectivity
  - **Scalability:** Capable of handling multiple scheduled tasks concurrently, supporting scalability in data operations.
  - **Productivity:** Automates repetitive tasks, freeing up time for more complex analyses.
- [Demo](#)

# Libraries - CIAP ecosystem

- Community-wide access to popular R/python libraries:
  - **tidyverse** - general data manipulation and visualization
  - **sommer** - mixed model analyses
  - **QBMS** - breeding database support
  - **rTASSEL/rPHG2** - genetics toolkit, GWAS, PHG integration
  - *...and many more!*
- Extensible!
  - Something you want but you don't see? **Ask!**