

Grand Challenges Default Platform Setup

Access the platform

To access the platform go to:

<https://grandchallenge.informaticslab.co.uk/hub/login>

If you do not have access to email Liam or Tim and we will grant access.

You have a choice of either Default or R. Select Default.

You should then come to this screen (You may need to click the Folder icon in the top left of the screen):

The screenshot shows a Jupyter notebook titled '1.0.0 Welcome.ipynb' in a web browser. The left sidebar displays a file explorer with a list of files and folders, including 'images', '1.0.0 Welcome.ipynb', '1.0.1 Notebook Intro.ipynb', '1.1.0.0 Installing packages .ipynb', '1.1.1.0 Iris Overview.ipynb', '1.1.3.0 ipyleaflet overview.ipynb', '1.2.0 Creating a new environment.ipynb', '1.3.0 Extracting Regional Data using Shap...', 'shape_utils.py', and 'sponsors.png'. The main content area of the notebook displays the 'COVID-19 Response Pangeo' welcome message. It includes logos for Azure, Met Office, The Environmental Futures & Big Data Impact Lab, and the European Union European Regional Development Fund. The text explains that Pangeo is a cloud-based data processing platform provided free of charge by Microsoft Azure, the Met Office, the Environmental Futures and Big Data Impact Lab, and the EU European Regional Development Fund for analysing weather data to explore links between the weather and the spread of COVID-19. It also provides information on updates, what Pangeo is, and how to explore the data.

How to use this document:

- General steps are general text.
- Essential aspects of the document are bolded, **read them if nothing else.**
- Red text is code. Input it exactly as shown (e.g. **'x = 3'** means in the terminal you input this code `x = 3`). There are also quotes around code, do not input the actual quote marks.
- `< >` is for your input. E.g. `https://grandchallenge.informaticslab.co.uk/user/<Github Username>/`
 - In this example, I would give my username making the URL I enter `https://grandchallenge.informaticslab.co.uk/user/berli/`

You can restart the server at (e.g. restart to go back to the default / r choice):

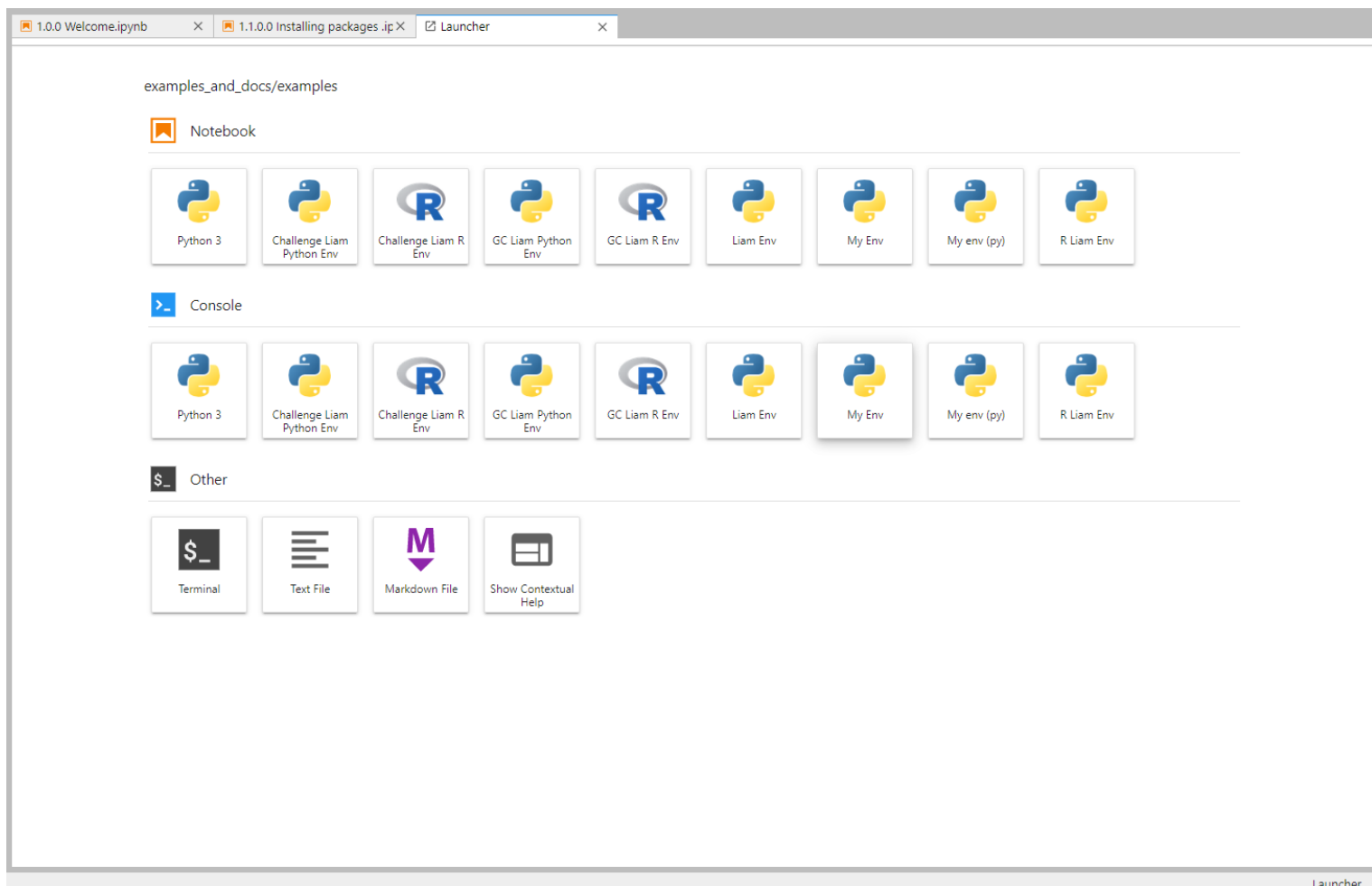
<https://grandchallenge.informaticslab.co.uk/hub/home>

Document Goal

The goal of this document is to help you to set up **both** a python and R environment for the Grand Challenges events **before** the event starts.

Even if you feel you don't need both environment, set them both up. If you feel you can do it on the first day on the grand challenges rather than now, you would be right. However, it would upset me, and so I would appreciate you doing the set up as soon as possible in case anything goes wrong we can sort you out.

By the end of this guide, you should have two Notebooks on your launcher on the platform that allows quick access to both a Python and R coding environment.



Python Notebook Setup – Platform Guide Location /hexamples_and_docs/examples/1.1.0.0

Installing Packages

Creating A new Environment

Folder icon to get access.

Step 1:

First of all, you need to open a launcher. Click the + in the top left of the platform. Then on other click on the 'Terminal' button. A new terminal tab should now open. Take note of the file path as this will become important when installing packages.

Step 2:

In the terminal input, the command `'conda init bash'`

Now close the terminal, e.g. press the x on the tab.

Reopen another terminal. Click the + on the top left of the platform and select the 'Terminal' button from other. The number on the terminal tab will increment do not worry about this. E.g. Terminal 1 and then when you close it, it will be Terminal 2.

Step 3:

Into the terminal input `'conda create --name my-new-env -y'`

Of note in the above code is that you have two '-' before the name, this denotes an argument to the command, e.g. the next element of the sequence is the name for the new environment, in this example my-new-env. However, you may want to change this. For example, you may want GC-liam-env, making the above code `conda create --name my-new-env -y`.

Step 4:

Into the terminal input `'conda activate my-new-env'`

Once you have input this command, you have now activated your new environment, and you should see you file path change in the terminal. Please see the below screenshot.

```
root@jupyter-berrli:/home/jovyan# conda activate Challenge-Liam-Python-Env
(/root/my-conda-envs/Challenge-Liam-Python-Env) root@jupyter-berrli:/home/jovyan#
```

This now means that you are changing aspects of the conda environment itself. This is what you will want to do to install packages, but this will be covered later.

Putting an Environment on the launcher

Step 5:

Now that you have created an environment, you need to add it to your launcher. This will mean that when you click the + icon in the top left, you get access to a notebook with your environment being used.

Into the terminal input `'conda install ipykernel -y'`

Step 6:

Into the terminal input `'python -m ipykernel install --name "my-env" --display-name "My Env" --user'`

The string after `--display-name` will be the name that is displayed on the launcher, and so you might want to change this to something like "Liams Python environment"

Step 7:

If you refresh your web browser and open the launcher (+) then you should see that your new python environment should be available. If you click the new environment, it should take you to either a notebook (top row) or a terminal (second row). Click on the notebook.

Step 8:

To confirm that your notebook is working as intended input the code `'print(3)'` and while the cursor is in the cell at the top, press Shift+Enter. The code should now run, and 3 be printed under the cell, and the cursor move to the second cell.

Step 9:

Please refer to the cheat sheet and linked tutorial for a guide on how to use Jupyter Notebooks.

Activating Environment Section

Whenever you open a new terminal from the launcher (+) you will be put into the top-level environment. If you want to make changes to a specific environment (such as install packages) then you will need to input the command, `'conda activate my-new-env'`

If you are currently in an active environment (which can be told from the file path (the text next to where the blinking cursor is in the terminal)), then you can stop editing the environment with the command `'conda deactivate'`, and then repeat the previous step.

R Notebook Setup - Platform Guide Location /hexamples_and_docs/examples/1.2.0 Creating a new environment

DO NOT JUMP TO THIS SECTION, SET UP PYTHON FIRST

Step 1:

First of all, you need to open a launcher. Click the + in the top left of the platform. Then on other click on the 'Terminal' button. This should now open a terminal tab. Take note of the file path as this will become important when installing packages.

Step 2:

Input into the terminal `'ENV_NAME=myenv'`

This will create a variable in the terminal to refer to your environment name.

Step 3:

Input into the terminal `'conda create -n $ENV_NAME \`
`-c conda-forge \`
`python ipywidgets nbconvert jupyterlab'`

Step 4:

Input into the terminal `'conda activate $ENV_NAME'`

As with the python set up, you should now see the file path on the terminal.

Step 5:

To get the needed R packages in the environment input the code `'conda install -c conda-forge r r-tidymodels r-ggplot2'`

Step 6:

Input into the terminal `'conda install -c conda-forge r-irkernel'`

Step 7:

Input into the terminal `'Rscript -e 'IRkernel::installspec(name = Sys.getenv("ENV_NAME"),`
`displayname = "GC-R-Env")'`

To explain the above code, you can replace the `Sys.getenv("ENV_NAME")` with the environment name you set up, the previous code just pulls the `ENV_NAME` variable from the system, and the `displayname` can also be changed to whatever you want to be displayed on the launcher. For example, for me to set up a conda environment called 'Renv' (I initially set `ENV_NAME = Renv`) with the launcher display name of GC Lima R Env, I would input the command:

Don't include an apostrophe in the display name as this will break the code, e.g. don't use Liam's Environment but rather make the names Liams environment etc.

If for some reason you are not able to get the command working above, then it may be due to differences in the characters when copying for the pdf. Please type in the above code into the terminal. Below can be seen a screenshot of working code I used to create an environment (imitate this with your desired names etc.)

```
Rscript -e 'IRkernel::installspec(name = "Renv", displayname = "GC Liam R Env")'
```

Step 8

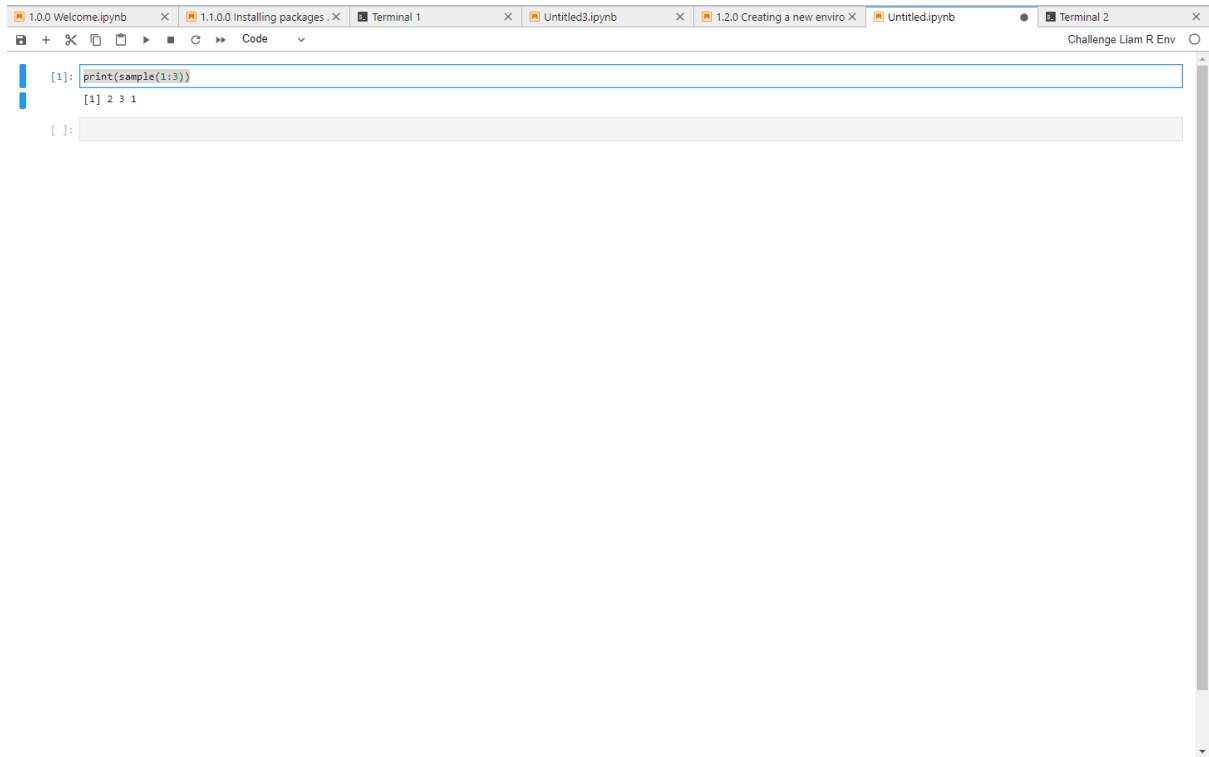
If you refresh your web browser and open the launcher, you should now see that you have a new button the first and second row that have the R logo. Click on that to open an R notebook.

Step 9:

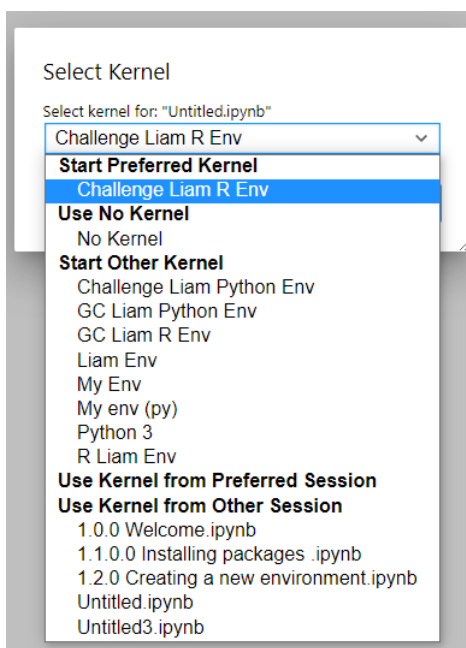
To confirm that your notebook is working as intended input the code `'print(sample(1:3))'` and while the cursor is in the cell at the top, press Shift+Enter. The code should now run and have [1] followed by three numbers get printed under the cell, and the cursor moves to the second cell.

Kernels and Custom Packages

Jupyter Notebooks run code using something called a kernel. The kernel that a notebook is using can be seen at the top right of the notebook. In the screenshot below it can be seen that the notebook is currently using the 'Challenge Liam R Env' kernel.



If you click the kernel text in the top right of the screen, you can change which kernel you are using. This change is achieved through the menu seen below.



As you proceed with your work, a typical task will be to install new packages. This is achieved by opening a terminal as done in the previous section and then inputting the command '`conda activate target-env`'

In this case, target-env is the environment (kernel) that you want to add a package to. When you input the above command, your file path should change as before, and now you are modifying the given environment. This means that you can then add packages to this environment.

To find out what command to input into the terminal, you can simply search google for the package name and conda. For example, if you wanted to install Sidecar, then you would search for conda sidecar and would reach this webpage <https://anaconda.org/conda-forge/sidecar>. You can then simply install the package with the code included on the webpage e.g. '`conda install -c conda-forge sidecar`'

The same can then be done with R packages however there is a r- before the package name to indicate it is an r package, eg. To install tidyverse, you would go to <https://anaconda.org/r/r-tidyverse>