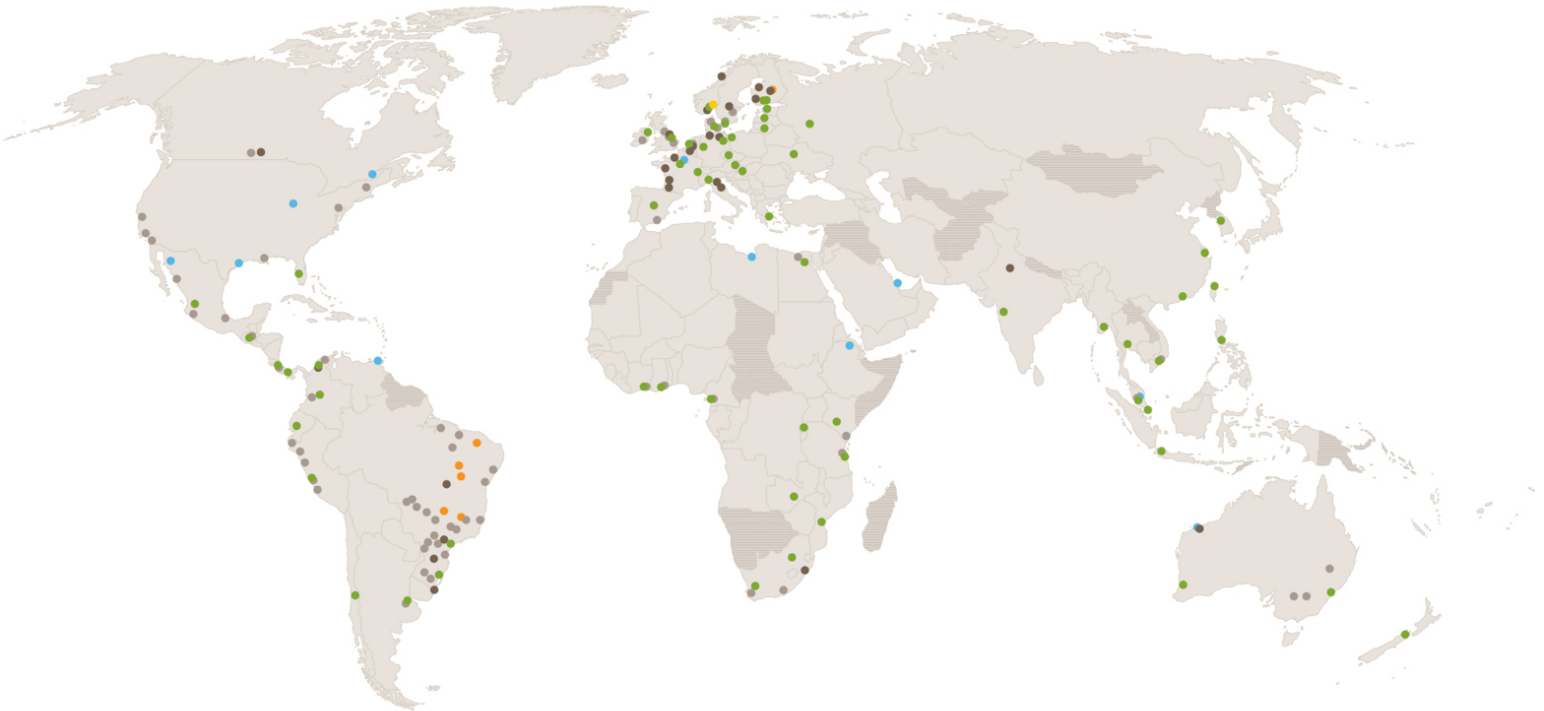
Knowledge grows

Knowledge grows

Richard Brosi _

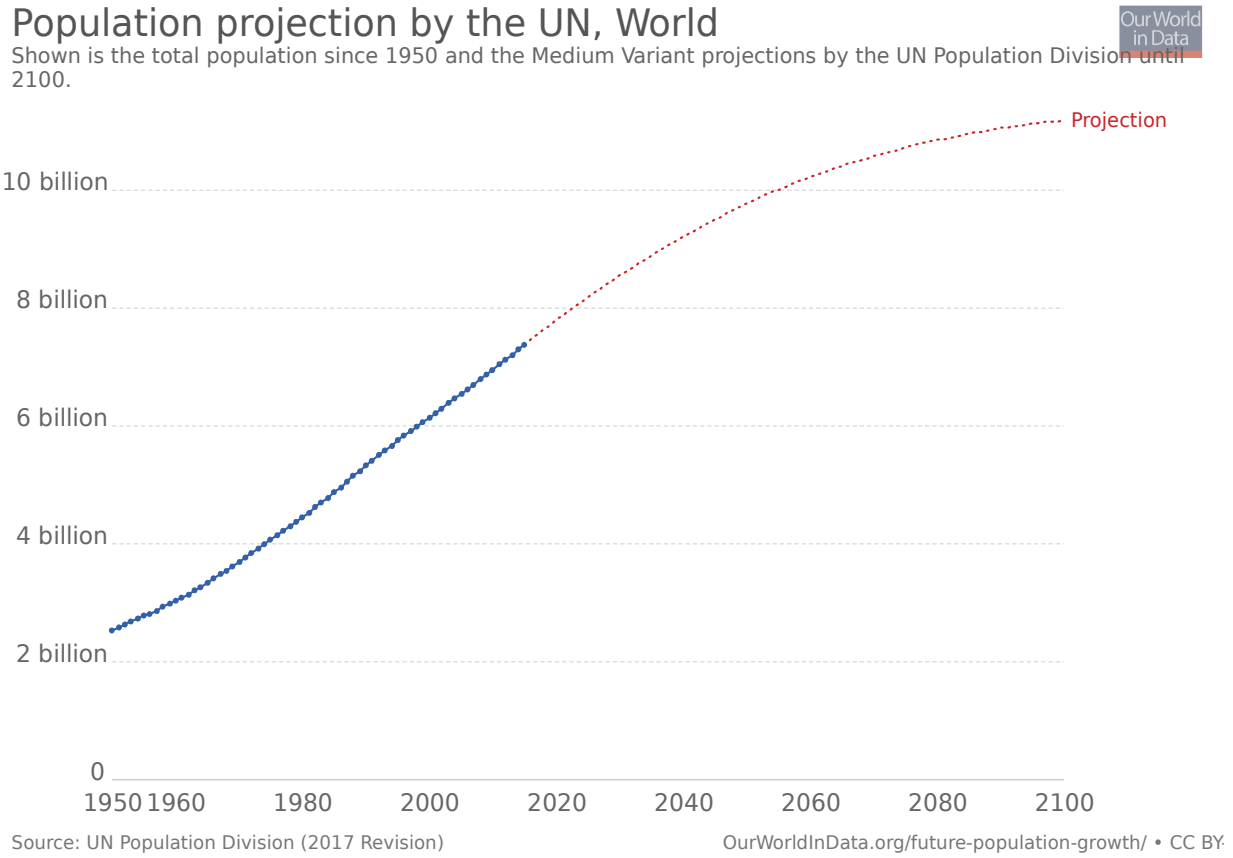
Yara _

Crop Nutrition

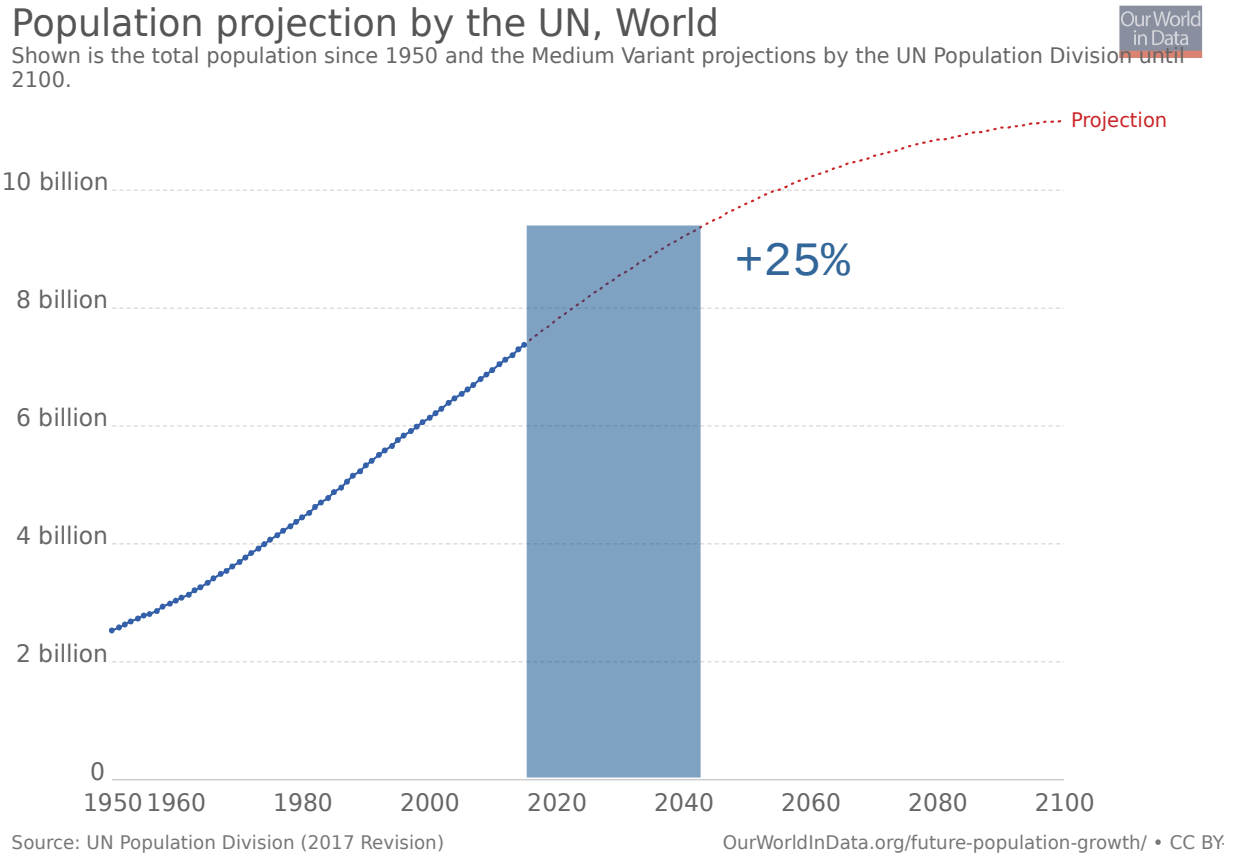


Yara _ | 16.000+ employees | USD 11.4bn revenue in 2017 |
| farming (fertilizer) & industrial chemicals |

It’s about food_

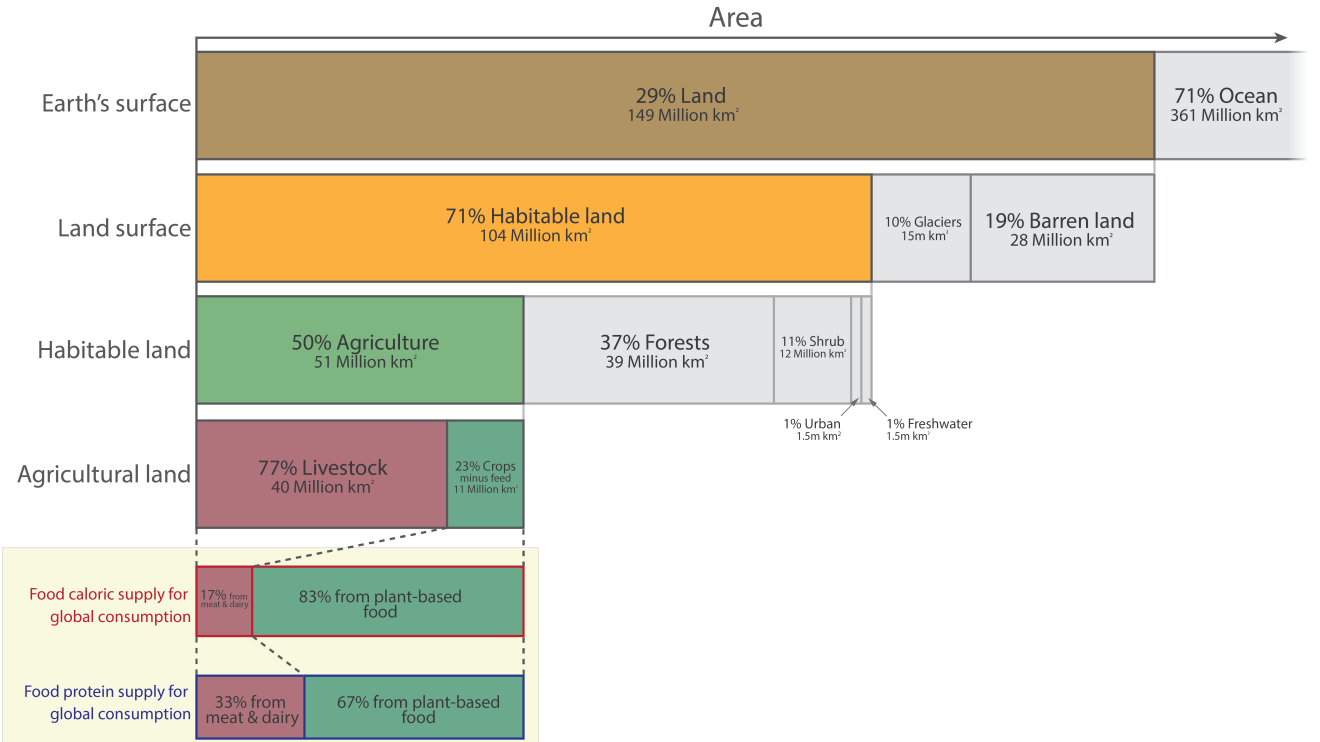


https://ourworldindata.org/future-population-growth


Population projection by the UN, World
Shown is the total population since 1950 and the Medium Variant projections by the UN Population Division until 2100.
Our World in Data
Projection
10 billion
8 billion
6 billion
4 billion
2 billion
0
+25%
1950
1960
1980
2000
2020
2040
2060
2080
2100
Source: UN Population Division (2017 Revision)
OurWorldInData.org/future-population-growth/ • CC BY

Global surface area allocation for food production

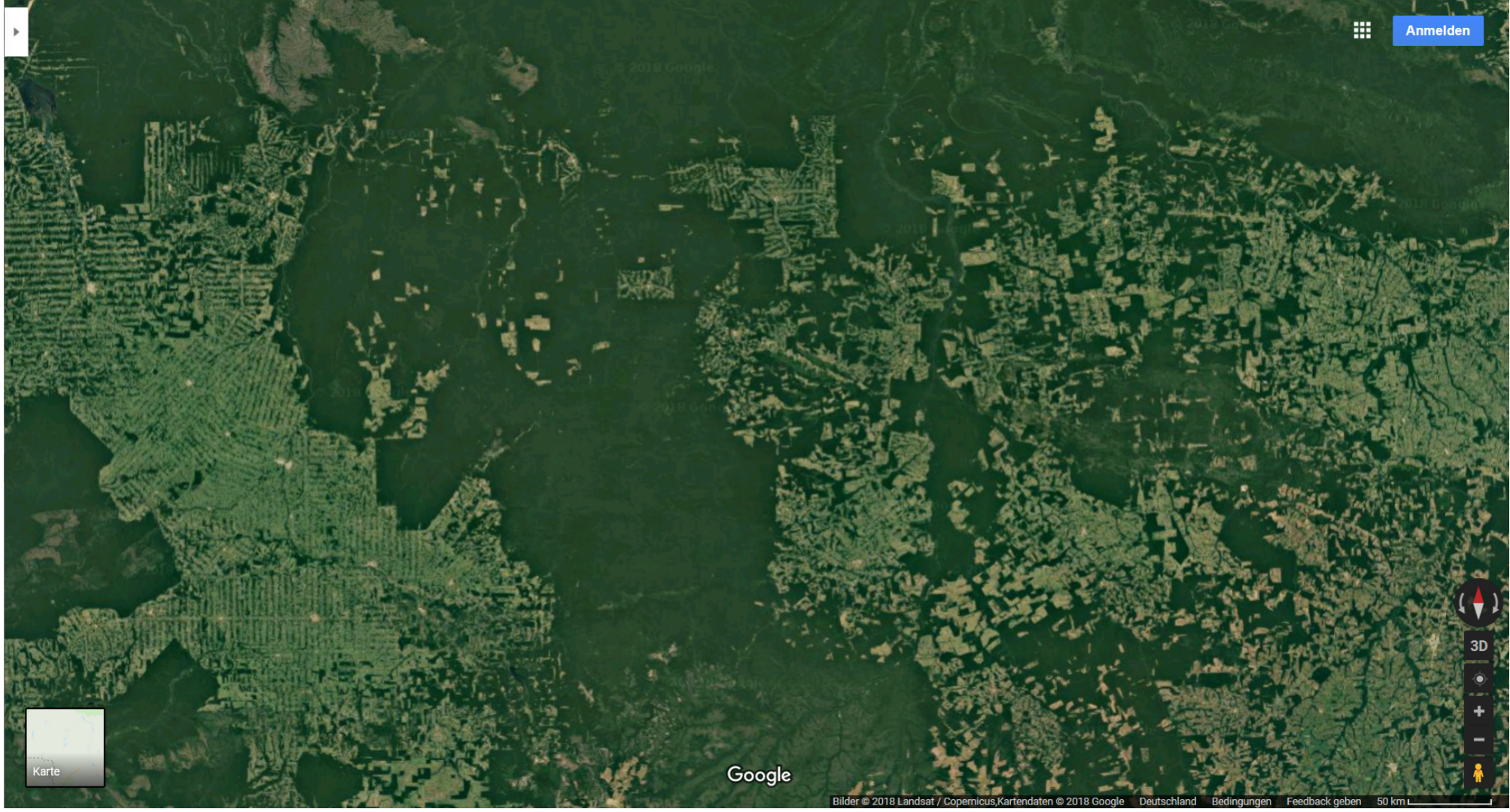
The breakdown of Earth surface area by functional and allocated uses, down to agricultural land allocation for livestock and food crop production, measured in millions of square kilometres. Area for livestock farming includes grazing land for animals, and arable land used for animal feed production. The relative production of food calories and protein for final consumption from livestock versus plant-based commodities is also shown.



Data source: based on UN Food and Agricultural Organization (FAO) Statistics.
The data visualization is available at OurWorldinData.org. There you find research and more visualizations on this topic.

This is today __ https://ourworldindata.org/wp-content/uploads/2013/10/Land-use-graphic-01-01-01.png

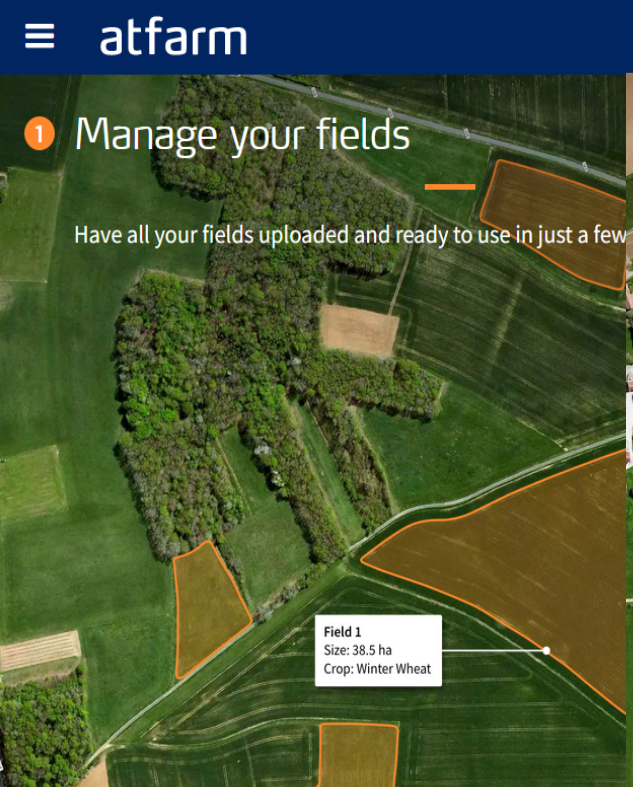
https://www.google.com/maps

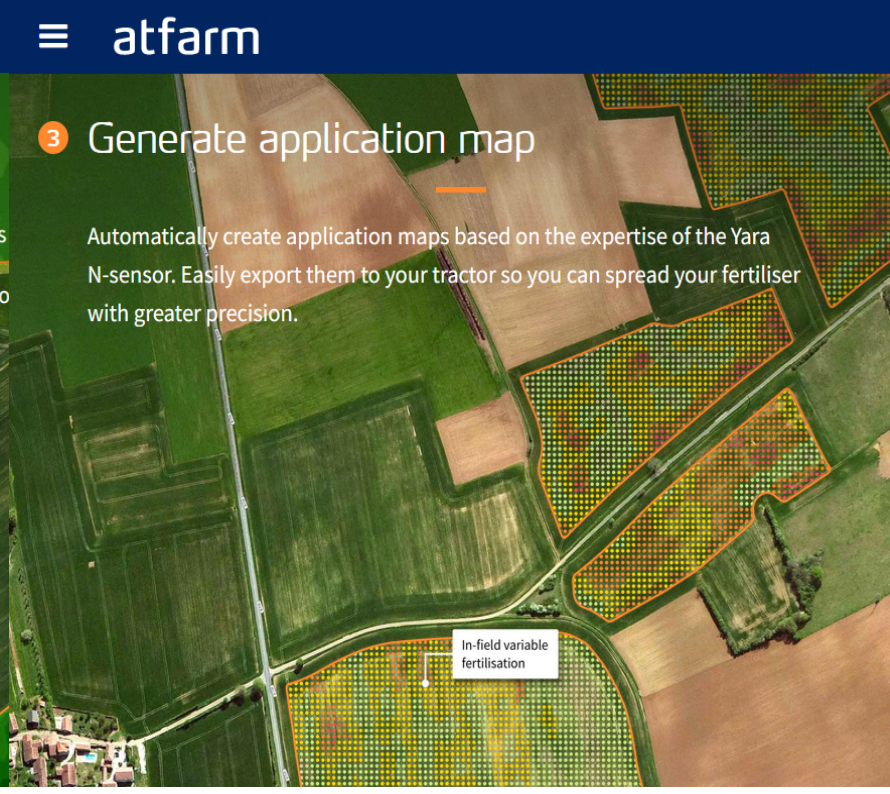


https://www.google.com/maps

It’s about food & **sustainable agriculture_**

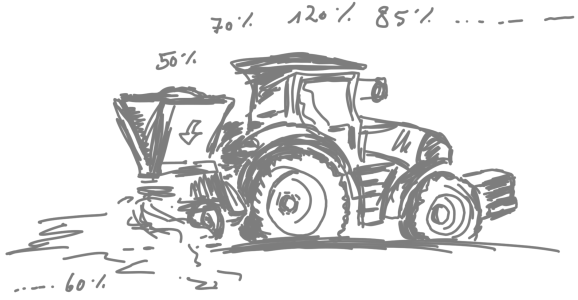
_ Yara | Digital Transformation





Precise Fertilization

- Satellite image analysis
- Apply just the optimum amount, no excess

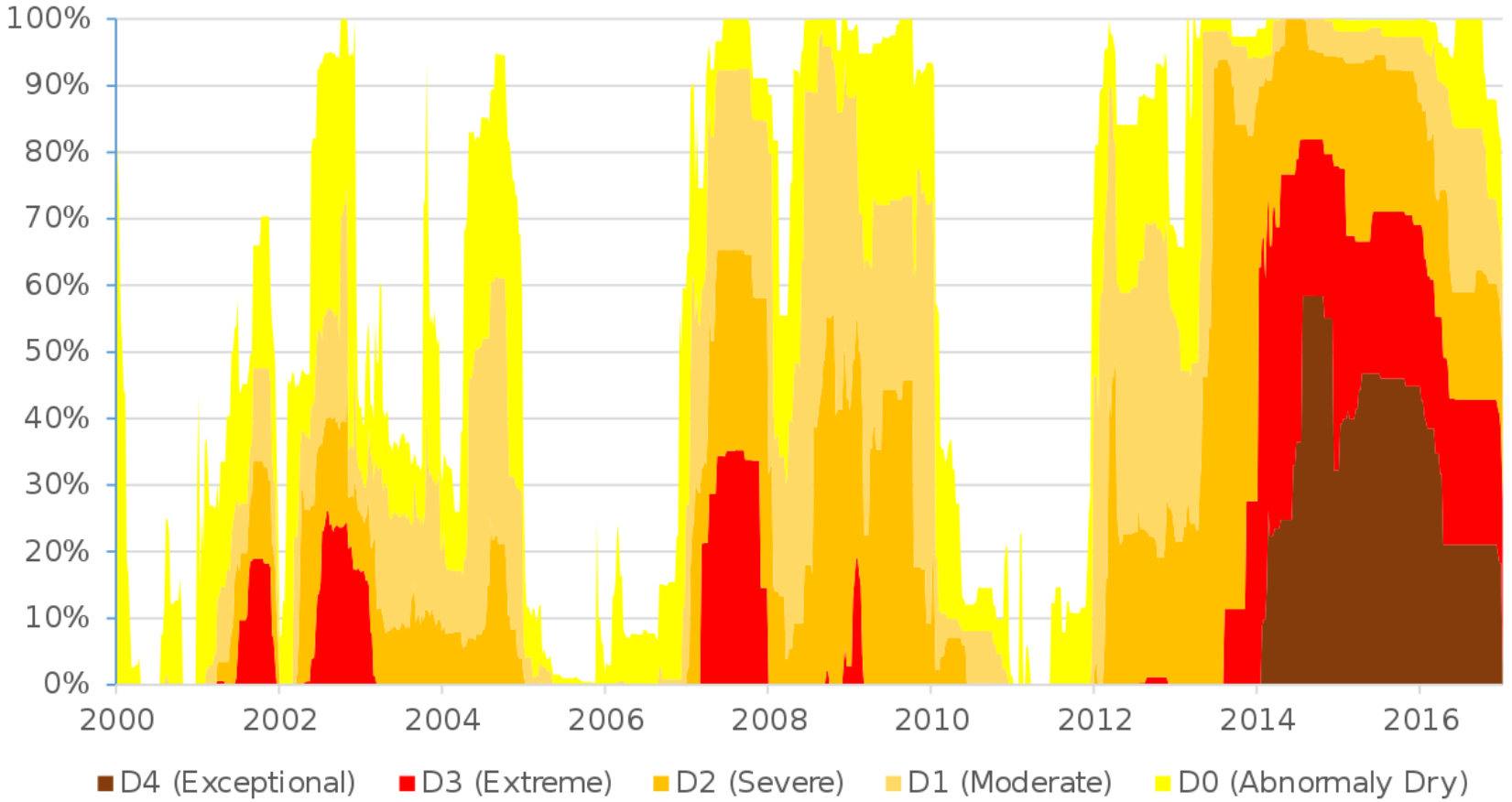


https://www.at.farm/

… currently covering more than 800’000 ha

Precise Irrigation

https://www.yara.com/water/ – Yara Water Solution – plant monitoring & irrigation scenario recommendation



_ California Drought Severity by Area | https://droughtmonitor.unl.edu/Data/DataTables.aspx | Phoenix7777

The Water Crises Aren't Coming—They're Here

For eons, the earth has had the same amount of water—no more, no less. What the ancient Romans used for crops and Nefertiti drank? It's the same stuff we bathe with. Yet with more than seven billion people on the planet, experts now worry we're running out of usable water. The symptoms are here: multiyear droughts, large-scale crop failures, a major city—Cape Town—on the verge of going dry, increasing outbreaks of violence, fears of full-scale water wars. The big question: How do we keep the H2O flowing?

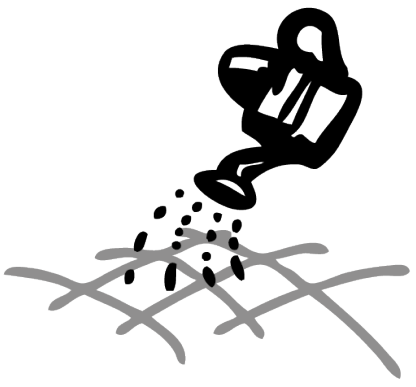
BY ALEC WILKINSON AUG 23, 2018 12.1K

> "We're overdrafting groundwater"
>
> "You have to start doing things differently."
>
> "**There's no more new water.**"

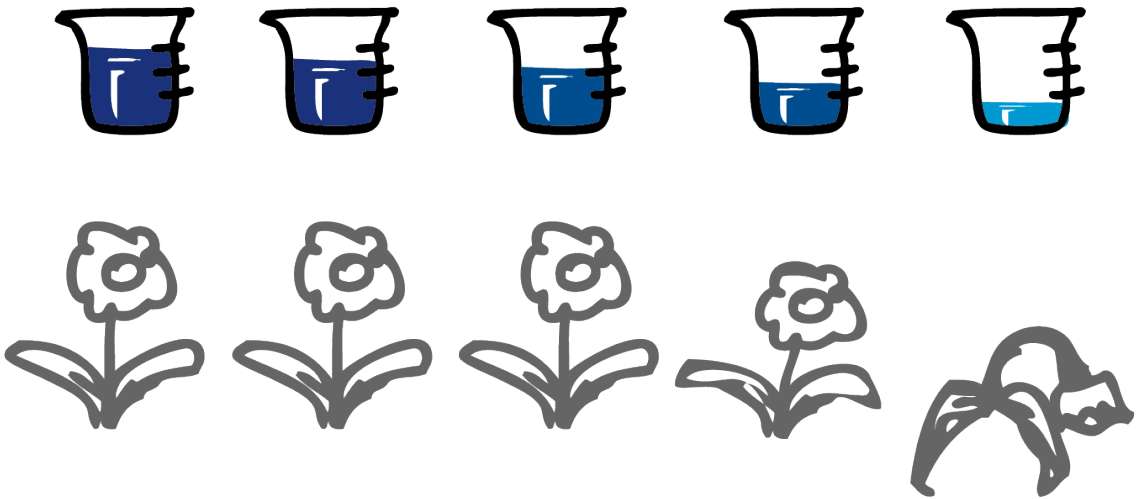
https://www.esquire.com/news-politics/a22627850/global-water-crisis/

_ Adaptive Irrigation & Drought Management

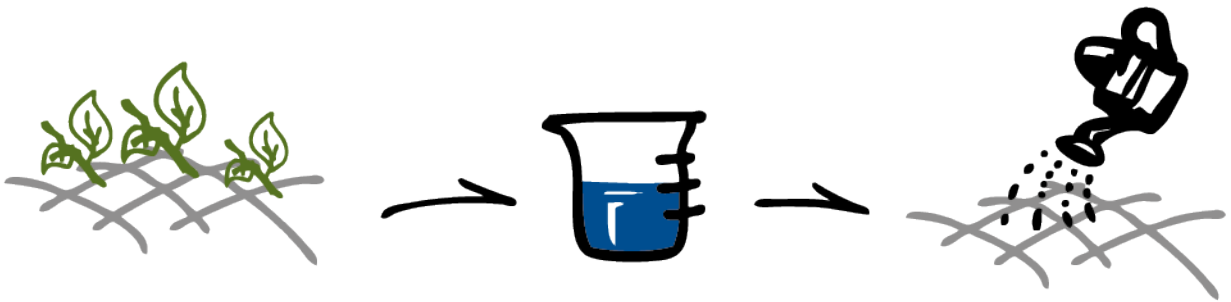
_



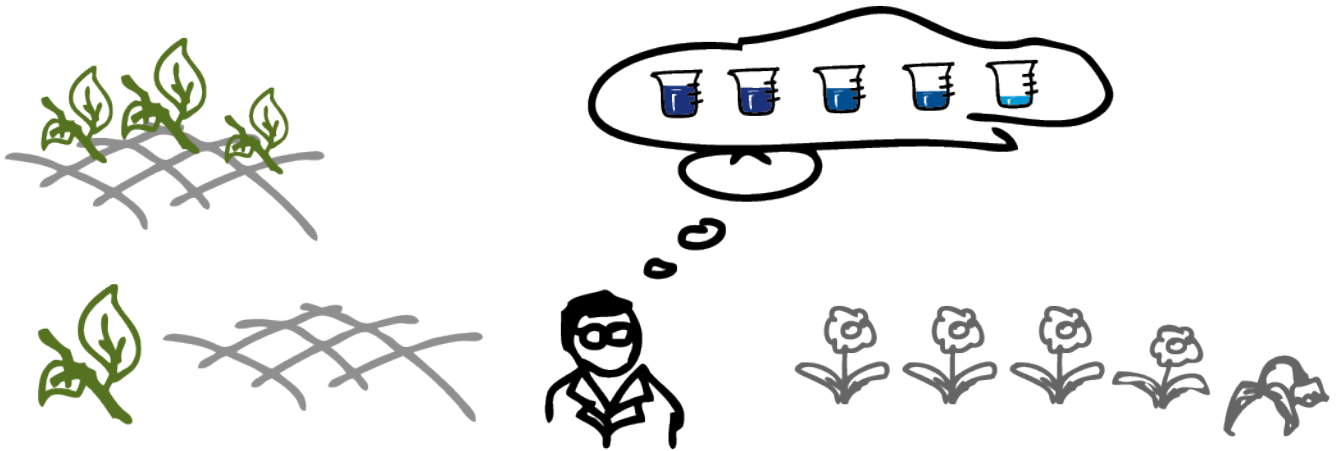
Aim: Precise & adaptive irrigation



_ How thirsty is the plant? – the plant’s “water status”

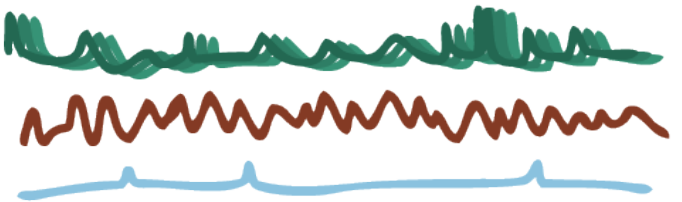


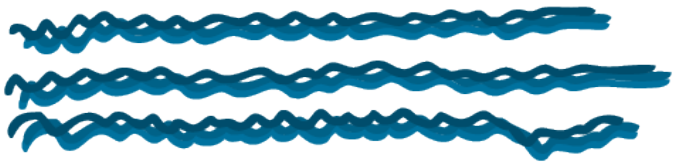
Using the water status for irrigation recommendations_



The Expert Knowledge_

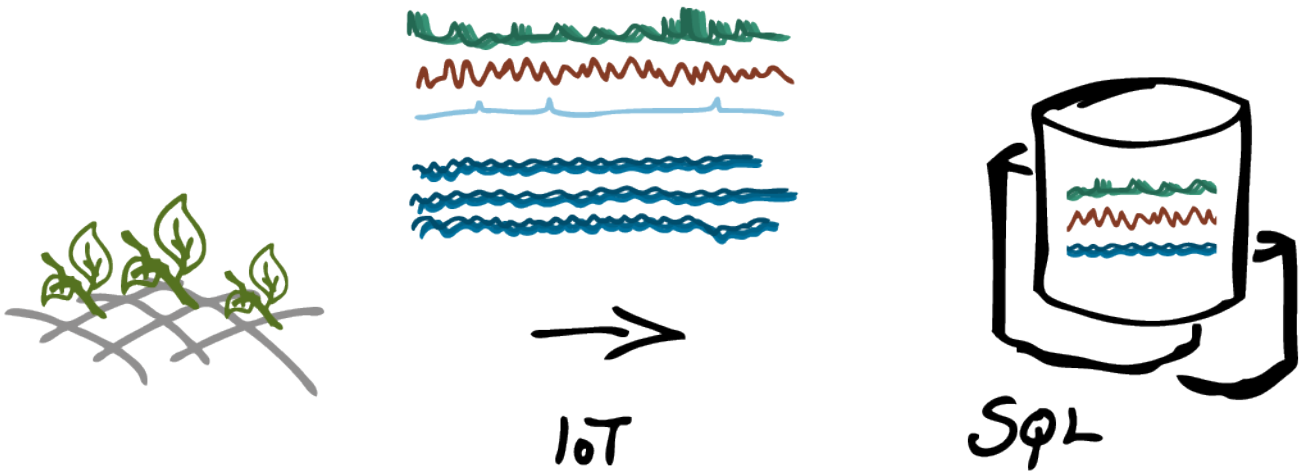
How to determine the water status automatically?





_ Gather data – in-field sensing hardware

_ Using R in a mid-sized data scenario


IoT
SQL

Why R?

People in the Agronomic Sector tend to speak R*

Existing Body of Code

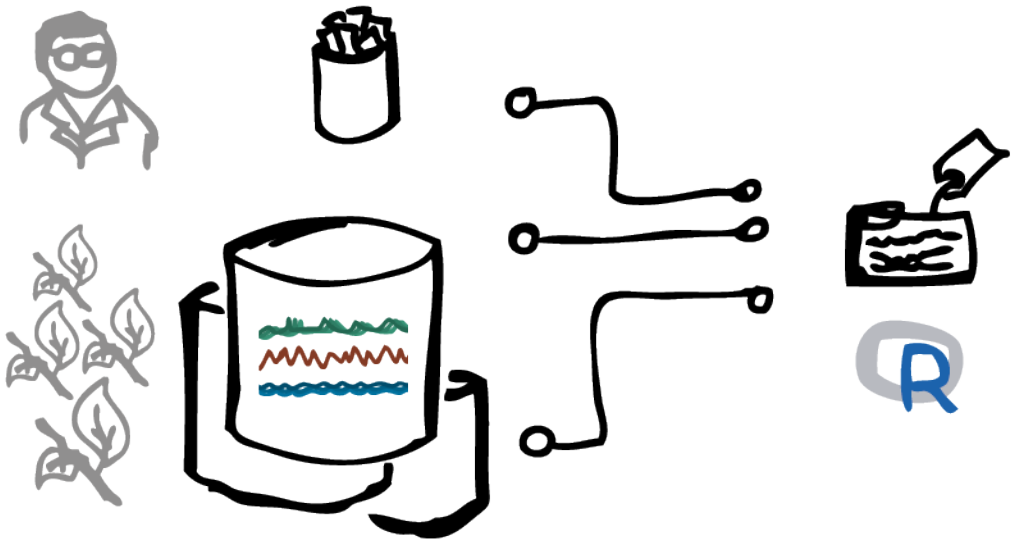
Broad array of CRAN packages in statistics, agronomy & biology

Very good documentation

Mid-sized data volumes

* a little bit _

https://shiny.rstudio.com/



Build model(s) from the labeled data_

R Interface to Python

The **reticulate** package provides a comprehensive set of tools for interoperability between Python and R. The package includes facilities for:

- Calling Python from R in a variety of ways including R Markdown, sourcing Python scripts, importing Python modules, and using Python interactively within an R session.
- Translation between R and Python objects (for example, between R and Pandas data frames, or between R matrices and NumPy arrays).
- Flexible binding to different versions of Python including virtual environments and Conda environments.

Reticulate embeds a Python session within your R session, enabling seamless, high-performance interoperability. If you are an R developer that uses Python for some of your work or a member of data science team that uses both languages, reticulate can dramatically streamline your workflow!

Python packages

You can install any required Python packages using standard shell tools like `pip` and `conda`. Alternately, reticulate includes a set of functions for managing and installing packages within virtualenvs and Conda environments. See the article on Installing Python Packages for additional details.

Calling Python

There are a variety of ways to integrate Python code into your R projects:

1. Python in R Markdown — A new Python language engine for R Markdown that supports bi-directional communication between R and Python (R chunks can access Python objects and vice-versa).
2. Importing Python modules — The `import()` function enables you to import any Python module and call it's functions directly from R.
3. Sourcing Python scripts — The `source_python()` function enables you to source a Python script the same way you would `source()` an R script (Python functions and objects defined within the script become directly available to the R session).
4. Python REPL — The `repl_python()` function creates an interactive Python console within R. Objects you create within Python are available to your R session (and vice-versa).

Why reticulate?

The package enables you to *reticulate* Python code into R, creating a new breed of project that weaves together the two languages.

Arrays

R matrices and arrays are converted automatically to and from NumPy arrays.

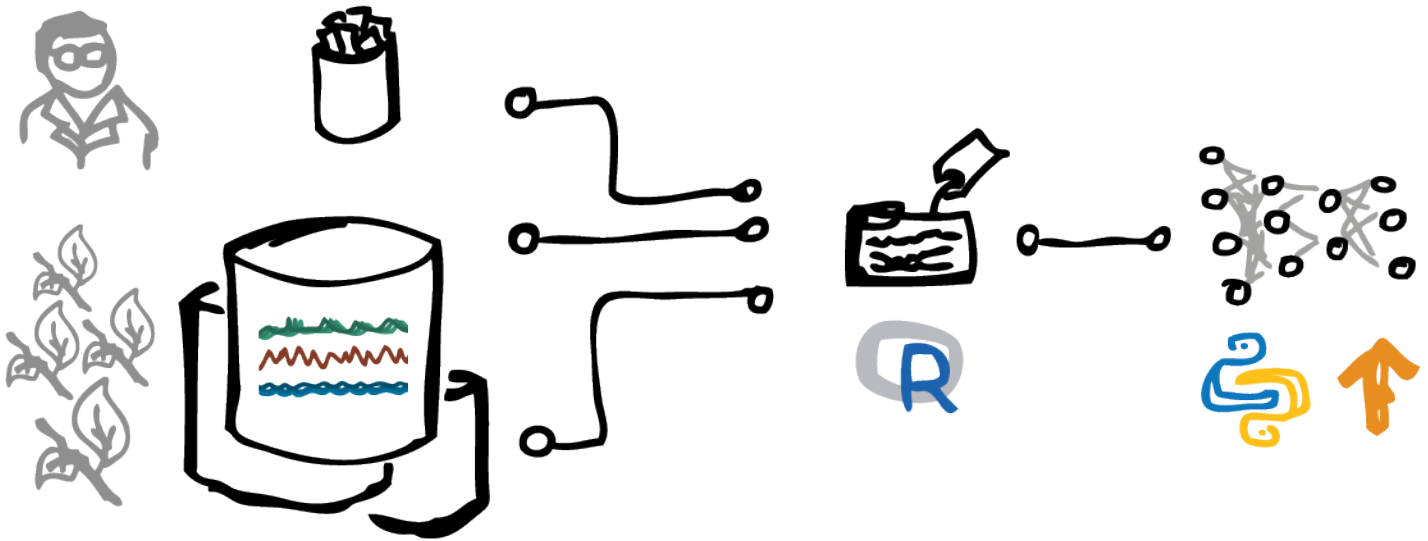
When converting from R to NumPy, the NumPy array is mapped directly to the underlying memory of the R array (no copy is made). In this case, the NumPy array uses a column-based in memory layout that is compatible with R (i.e. Fortran style rather than C style). When converting from NumPy to R, R receives a column-ordered copy of the NumPy array.

Data Frames

R data frames can be automatically converted to and from Pandas DataFrames. By default, columns are converted using the same rules governing R array <-> NumPy array conversion, but a couple extensions are provided:

R	Python
Factor	Categorical Variable
POSIXt	NumPy array with dtype = `datetime64[ns]`

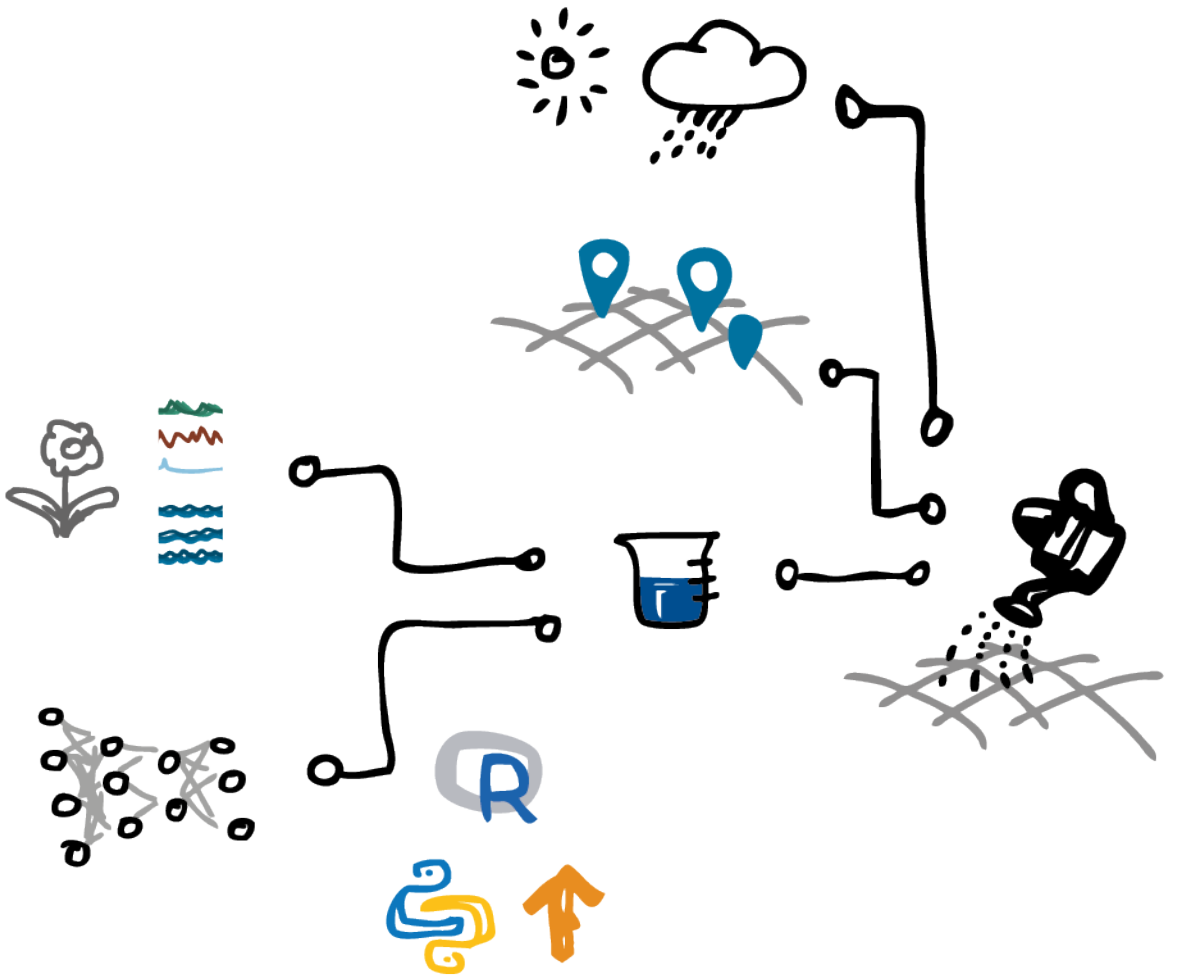
https://rstudio.github.io/reticulate/



https://rstudio.github.io/reticulate/

=> convenient access to Tensorflow, Keras, …

Build the model(s)_

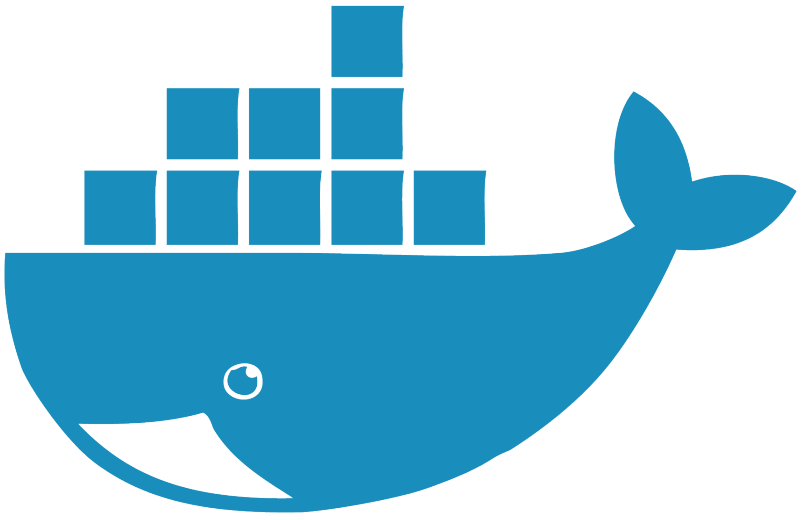


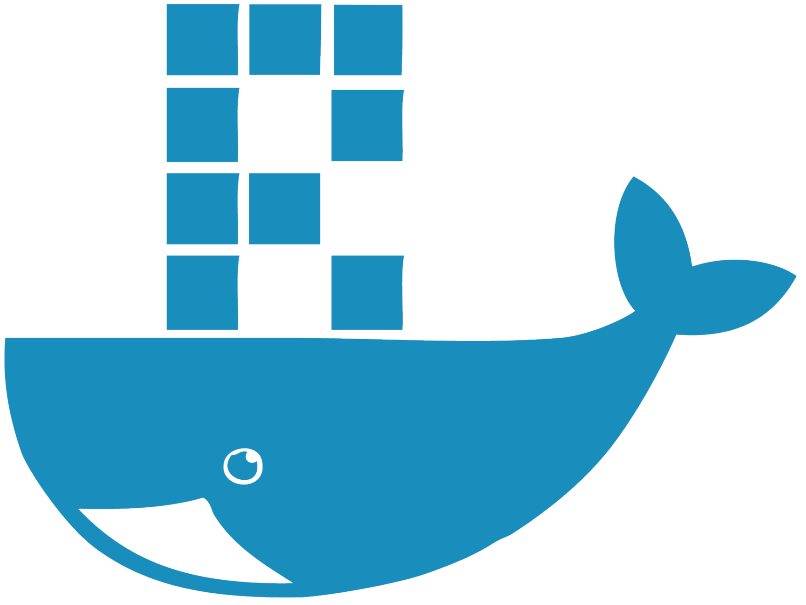
Online water status prediction_

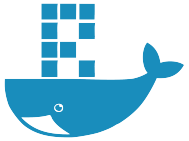
Reproducibility / Consistent environments

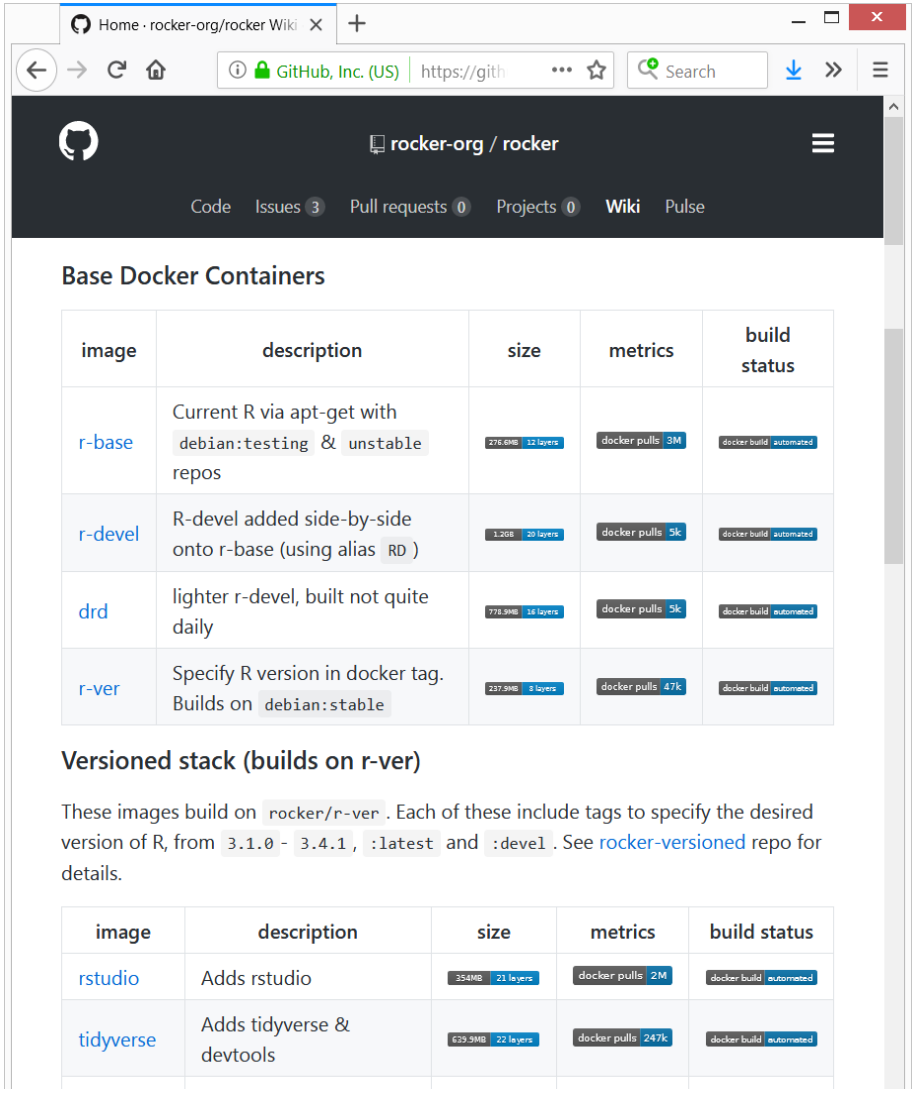
Portability / Scalability

All Is Code

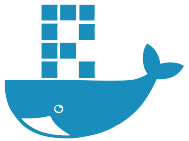








https://arxiv.org/abs/1710.03675

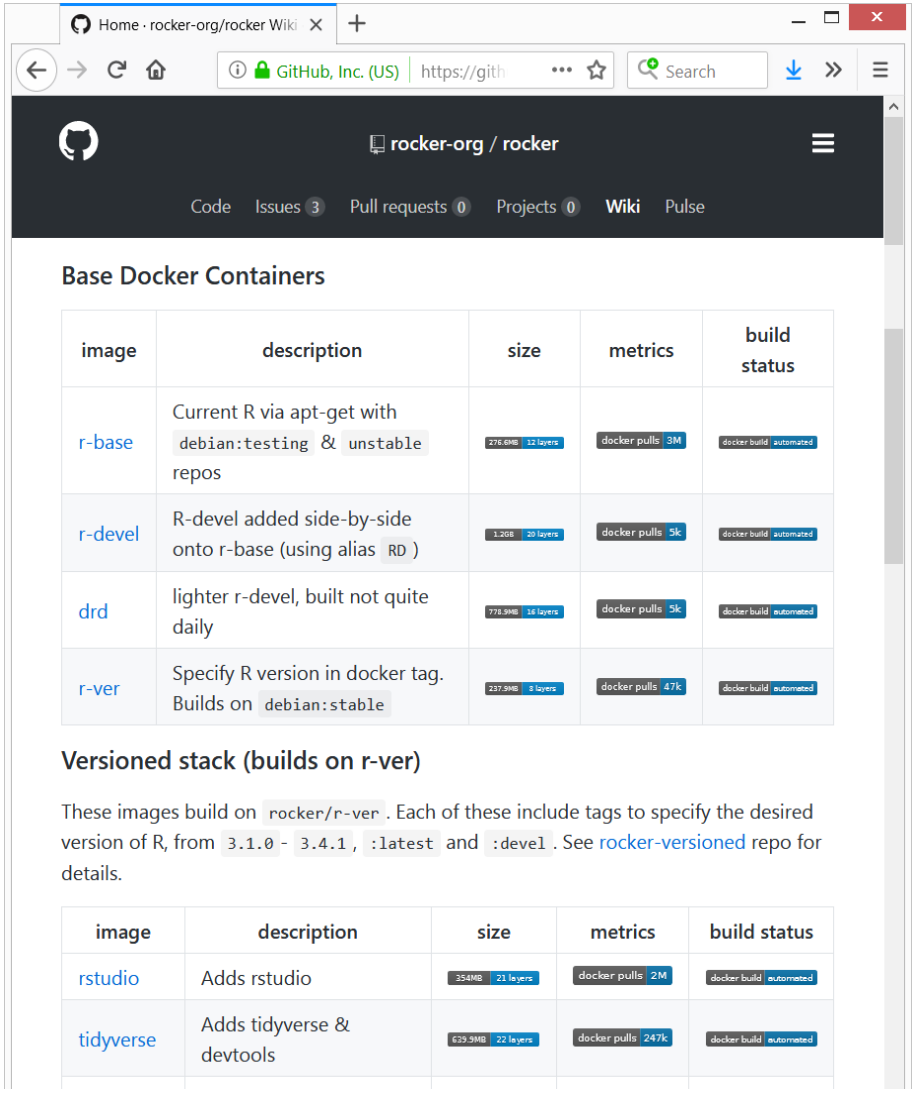


CRAN Time Machine

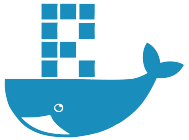
For the purpose of **reproducibility**, MRAN hosts **daily snapshots** of the CRAN R packages and R releases as far back as Sept. 17, 2014.

Use our **Time Machine** to browse CRAN contents from the past.

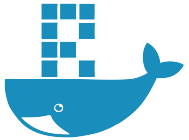
Browse Snapshots

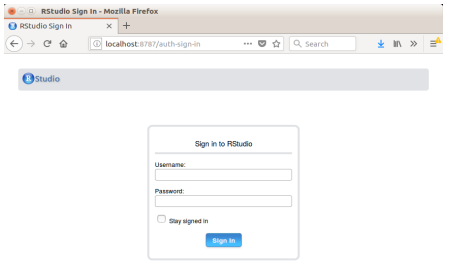


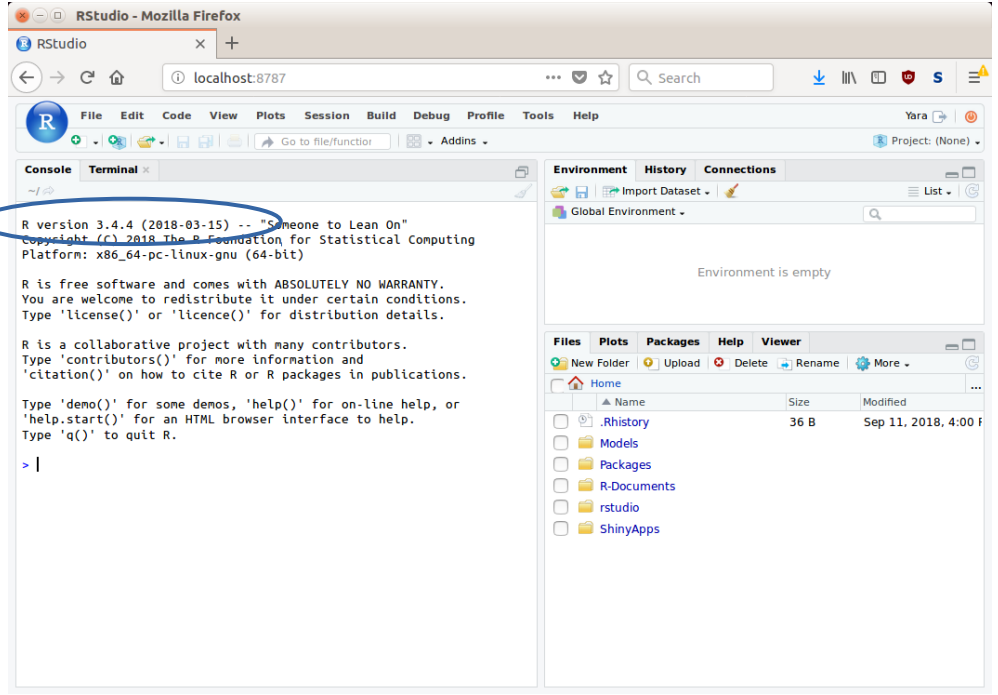
https://mran.microsoft.com/



R-docker for both development and production

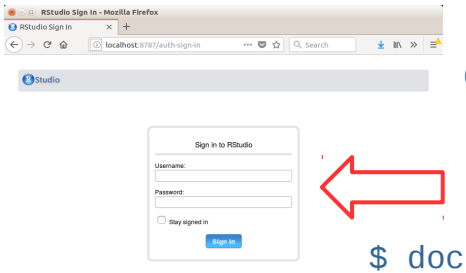






RStudio via browser_

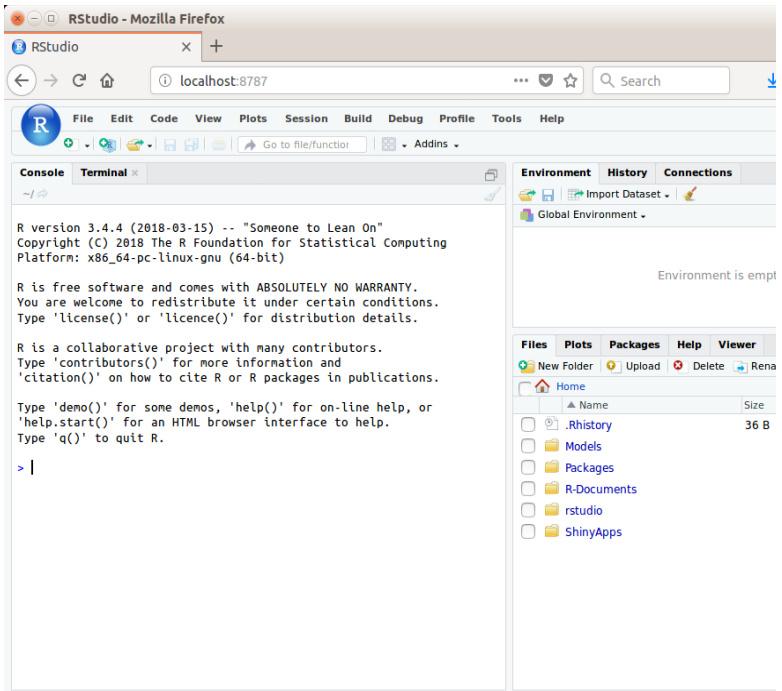
```
Dockerfile

FROM rocker/tidyverse:3.4.4
RUN  apt-get update && \
     apt-get -y –no-inst… && \
     install2.r –deps TRUE \
          DBI \
          ... 
          logging \
          ... 
          testthat \
          ...


$ docker build -f Dockerfile \
    -t tidyrocker .
```



CAVEAT 1:
...this password is an

Environment Variable

```
$ docker inspect MyRStudio
```

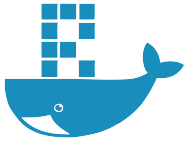


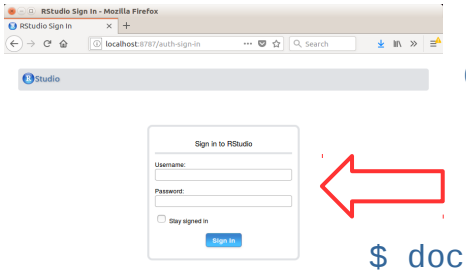
https://github.com/rocker-org/rocker/wiki/FAQ

I see RStudio using `http` instead of `https`. Is my password being transferred in plain text?

Nope. RStudio encrypts your credentials (using a JavaScript RSA implementation) It is possible to run RStudio server behind a proxy or over ssh tunnel instead (e.g. with all other ports behind a firewall) for additional security; see the RStudio documentation.

RStudio via browser_





CAVEAT 1:
...this password is an

Environment Variable

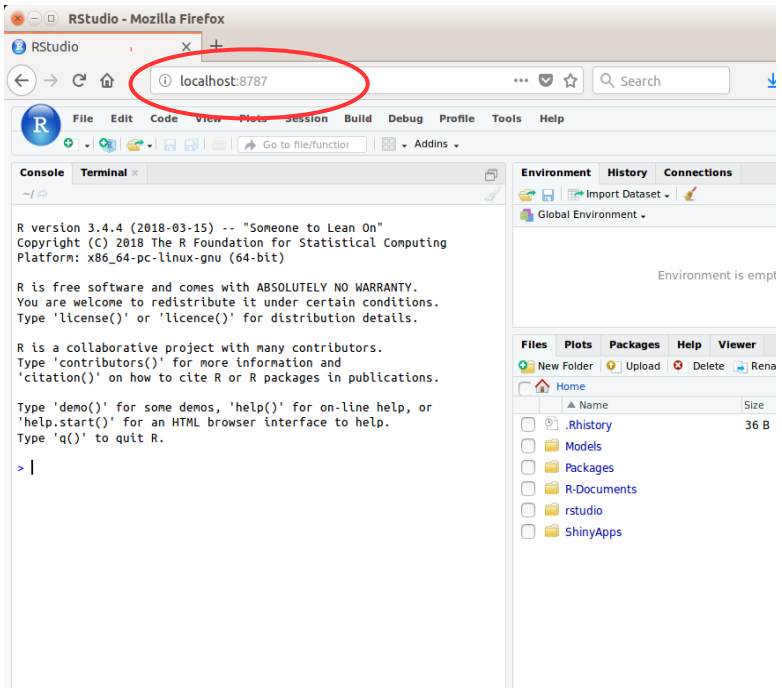
```
$ docker inspect MyRStudio
```

CAVEAT 2:
...no HTTPS

a) http behind ip filter/firewall

OR

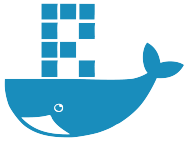
b) nginx as SSL reverse proxy



https://myserver:80

http://myintserver:8787

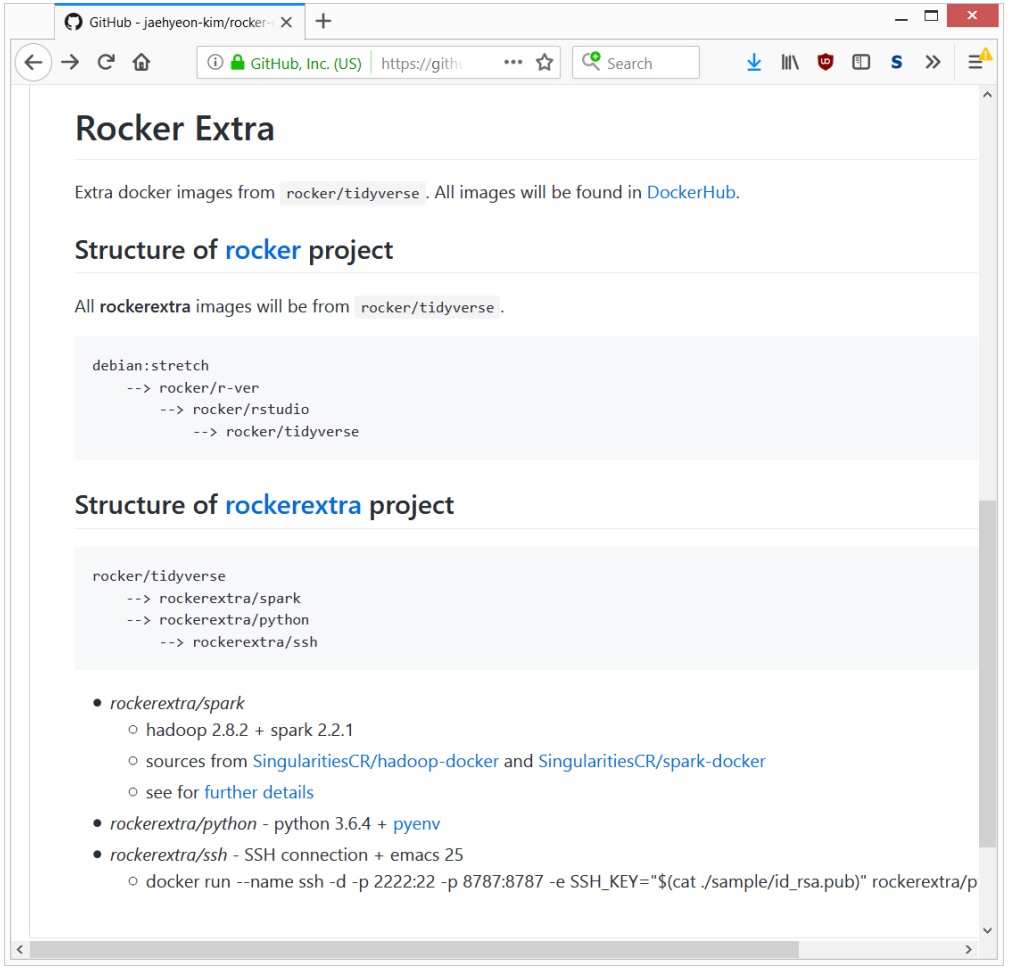
RStudio via browser_

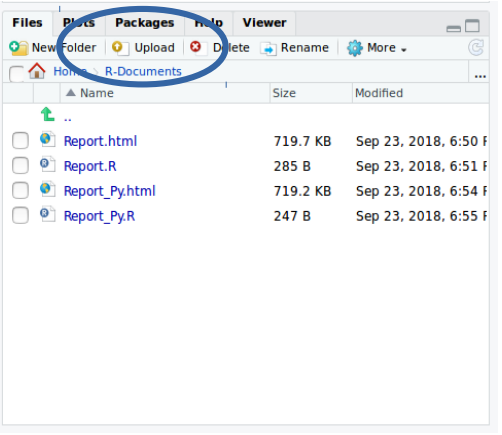


https://stackoverflow.com/questions/41422826/install-python-of-specific-version-system-wide-with-pyenv/43321411

https://github.com/pyenv/pyenv

https://github.com/jaehyeon-kim/rocker-extra

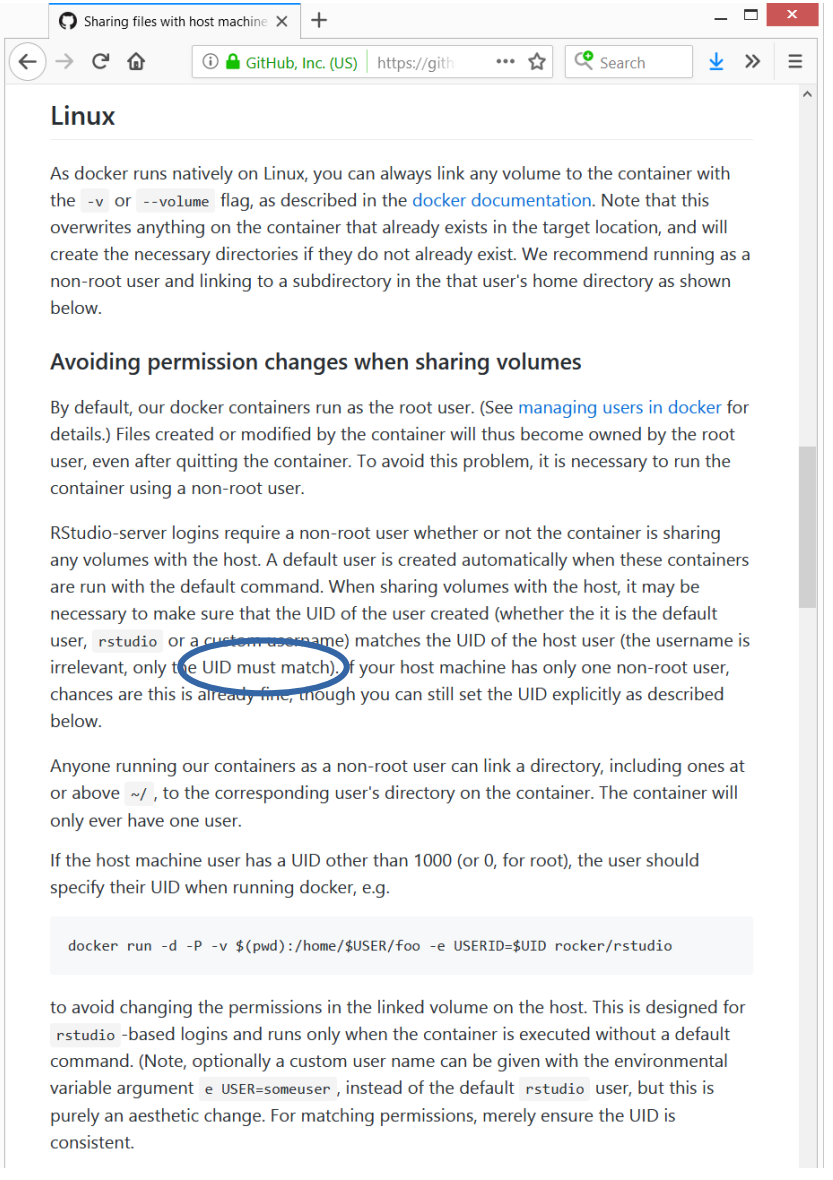


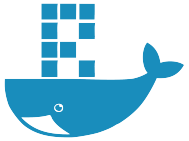


Use upload/download option

OR

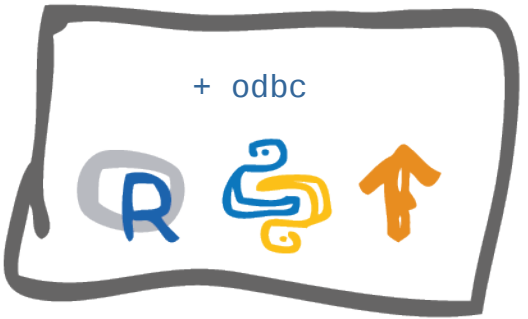
https://github.com/rocker-org/rocker/wiki/Sharing-files-with-host-machine

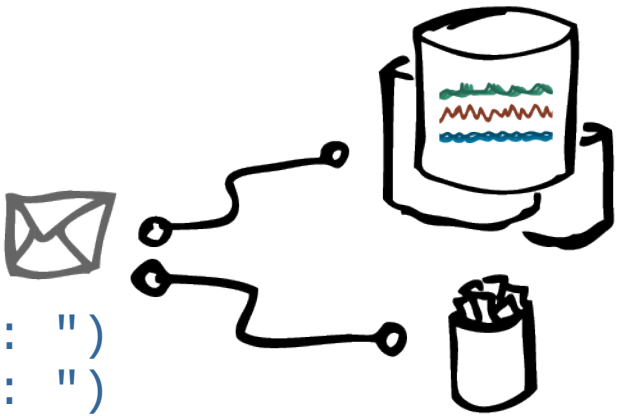




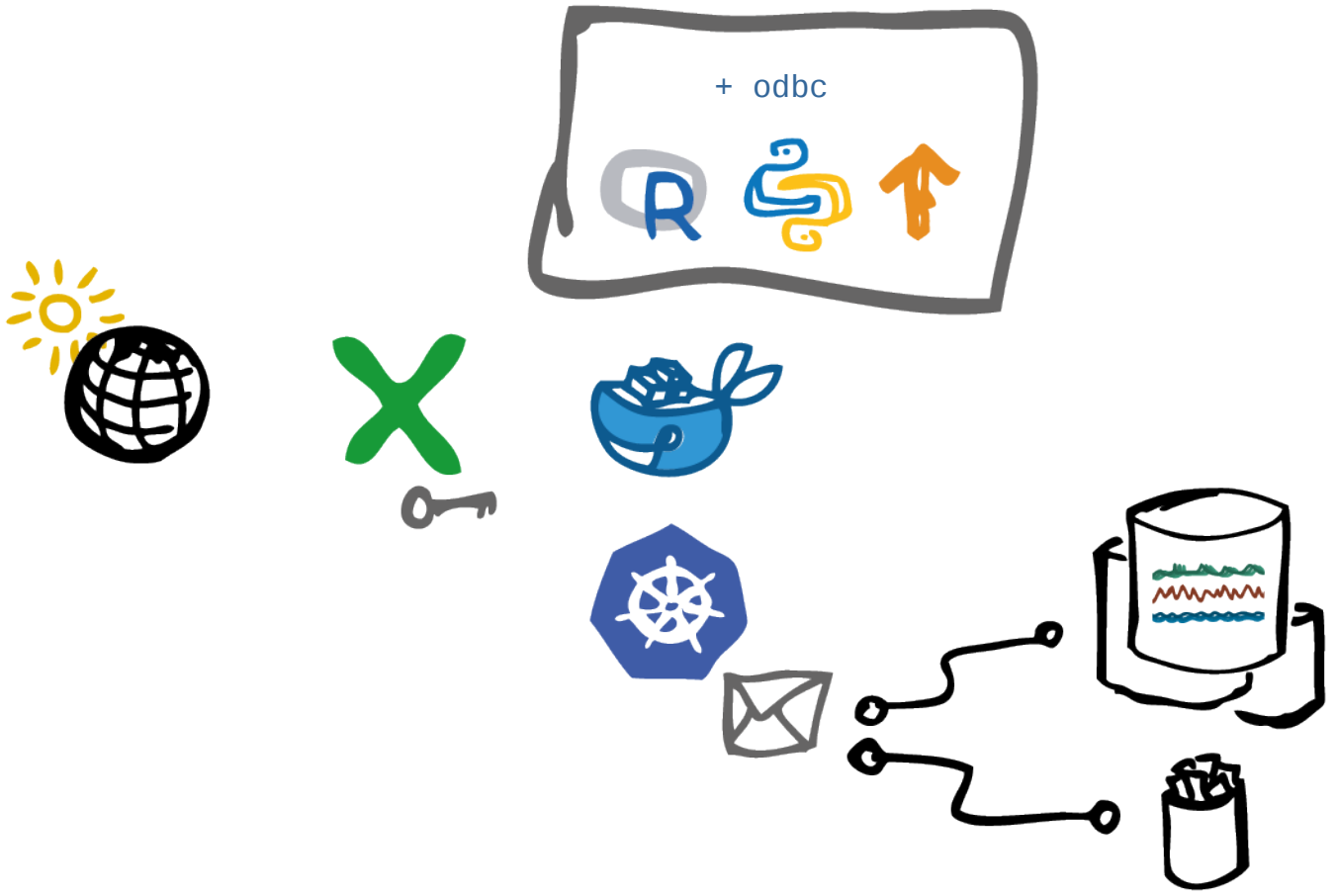
```
Another_Dockerfile_To_Add_MS_ODBC

FROM customrockerimage

RUN  apt-get update && … \
     curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add - && \
     curl https://packages.microsoft.com/config/debian/9/prod.list > \
        /etc/apt/sources.list.d/mssql-release.list && \
     apt-get update && \
     ACCEPT_EULA=Y apt-get -y install \
        msobdcsql17 \
        unixodbc-dev && \
            install2.r –deps TRUE \
                RODBC

COPY odbc.ini /etc/odbc.ini

$ docker build -f Another_Dockerfile_To_Add_MS_ODBC -t customrockerimage/msodbc .
```

Chaining Dockerfiles_



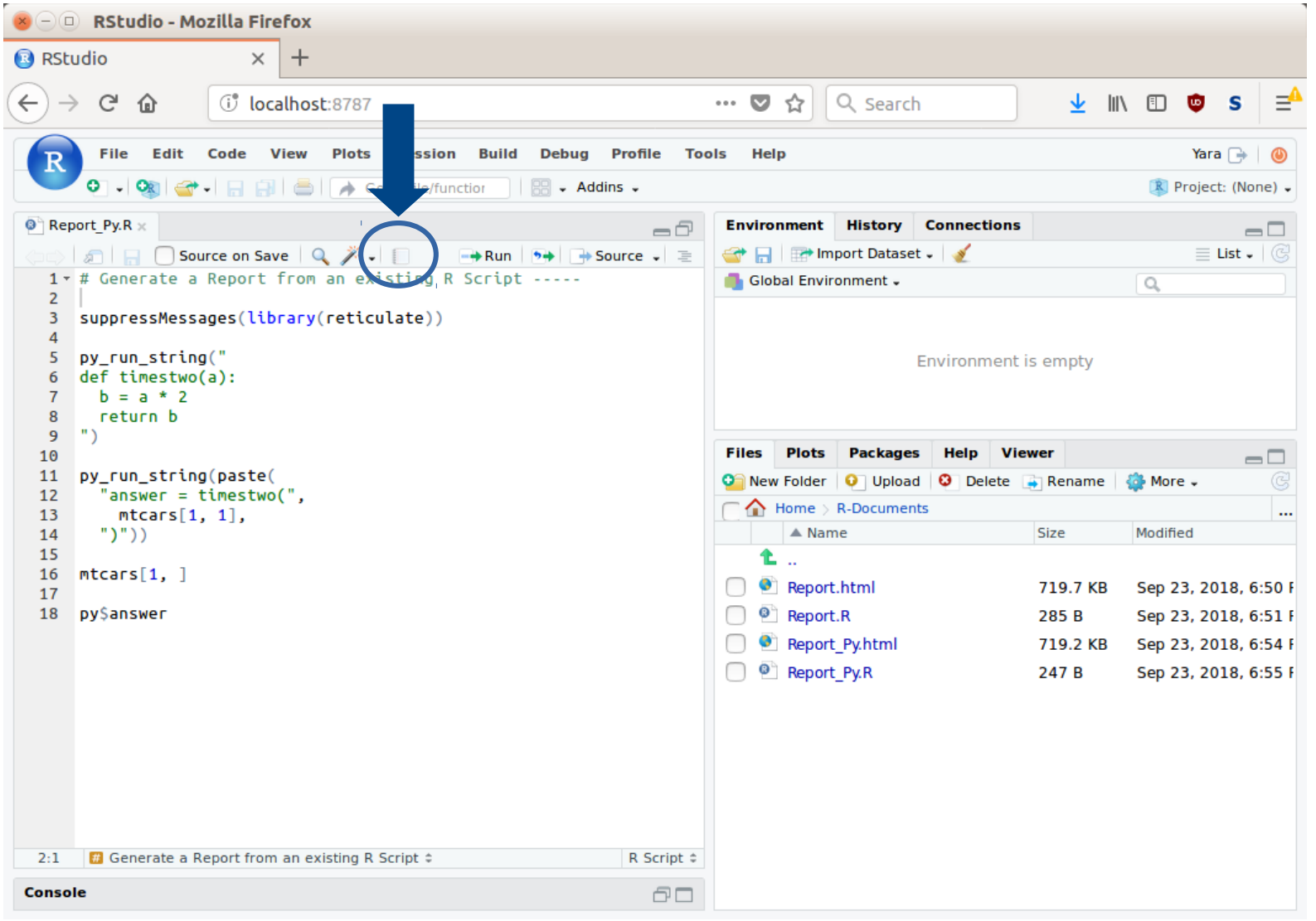


```
rstudioapi::askForPassword("pwd: ")
          getPass::getPass("pwd: ")
```

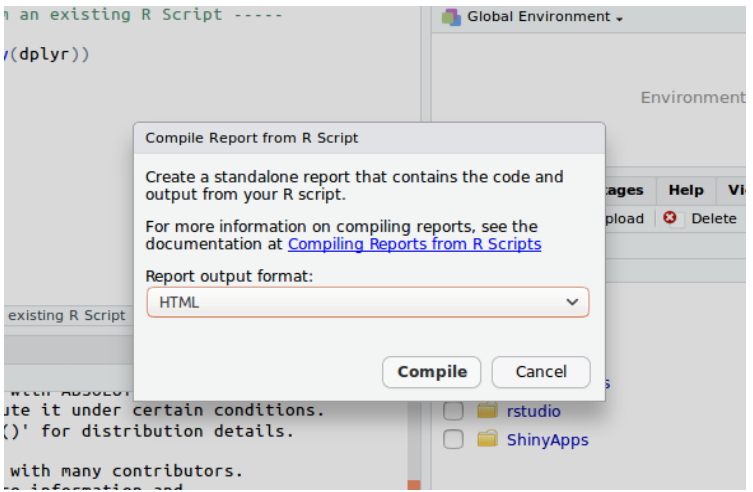


DB access – non-interactive – https://kubernetes.io/docs/concepts/configuration/secret/

Some things I found useful…



RStudio Server_



~/R-Documents/Report_Py.html - Mozilla Firefox

localhost:8787/?view=rmarkdown

Report_Py.html | Open in Browser | Find | Publish

Report_Py.R

Yara

Sun Sep 23 16:56:44 2018

```
# Generate a Report from an existing R Script -----

suppressMessages(library(reticulate))

py_run_string("
def timestwo(a):
  b = a * 2
  return b
")

py_run_string(paste(
  "answer = timestwo(",
    mtcars[1, 1],
  ")"))

mtcars[1, ]
```

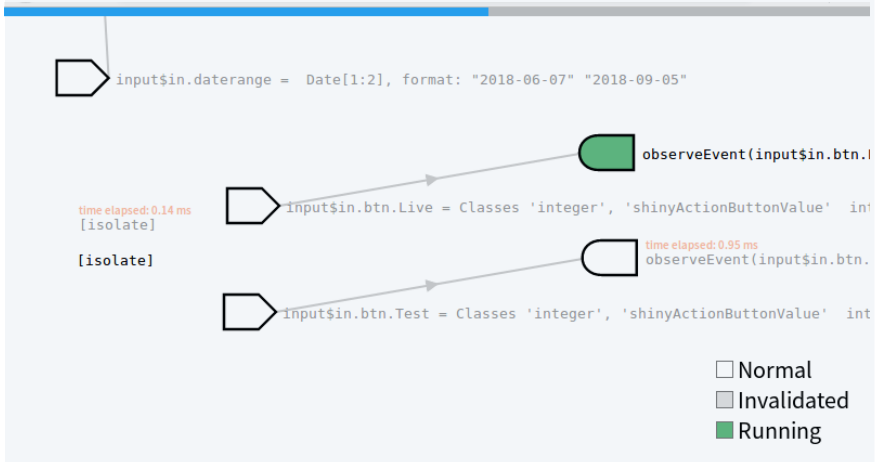
```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4  21   6  160 110  3.9 2.62 16.46  0  1    4    4
```

```
py$answer
```

```
## [1] 42
```

```
…
options(shiny.reactlog=TRUE)
…
```

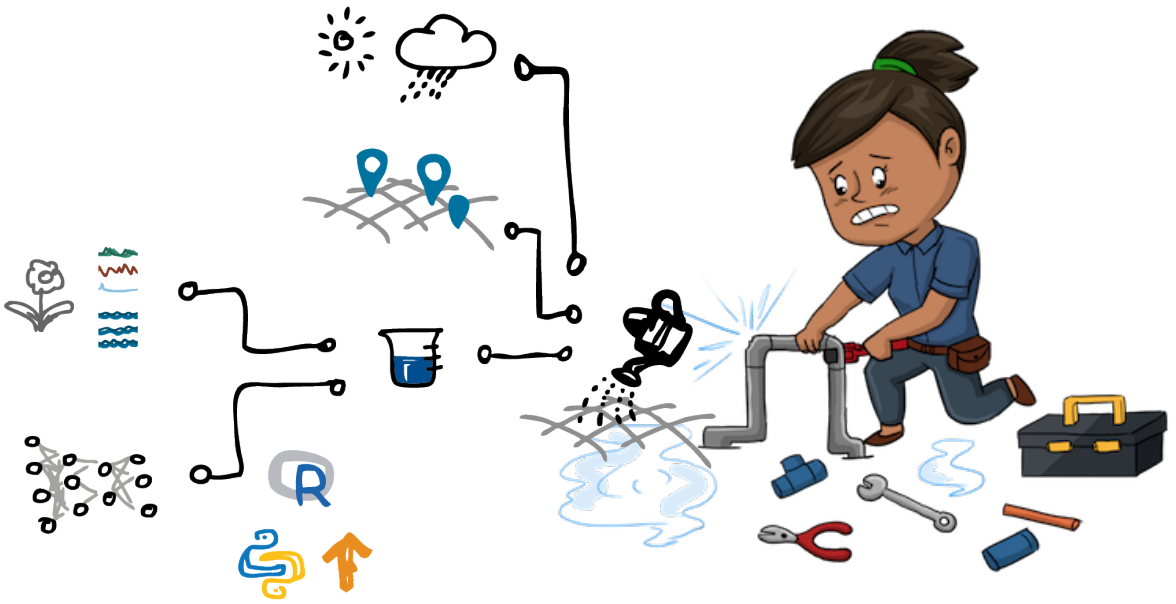
→ *Press* Ctrl + F3



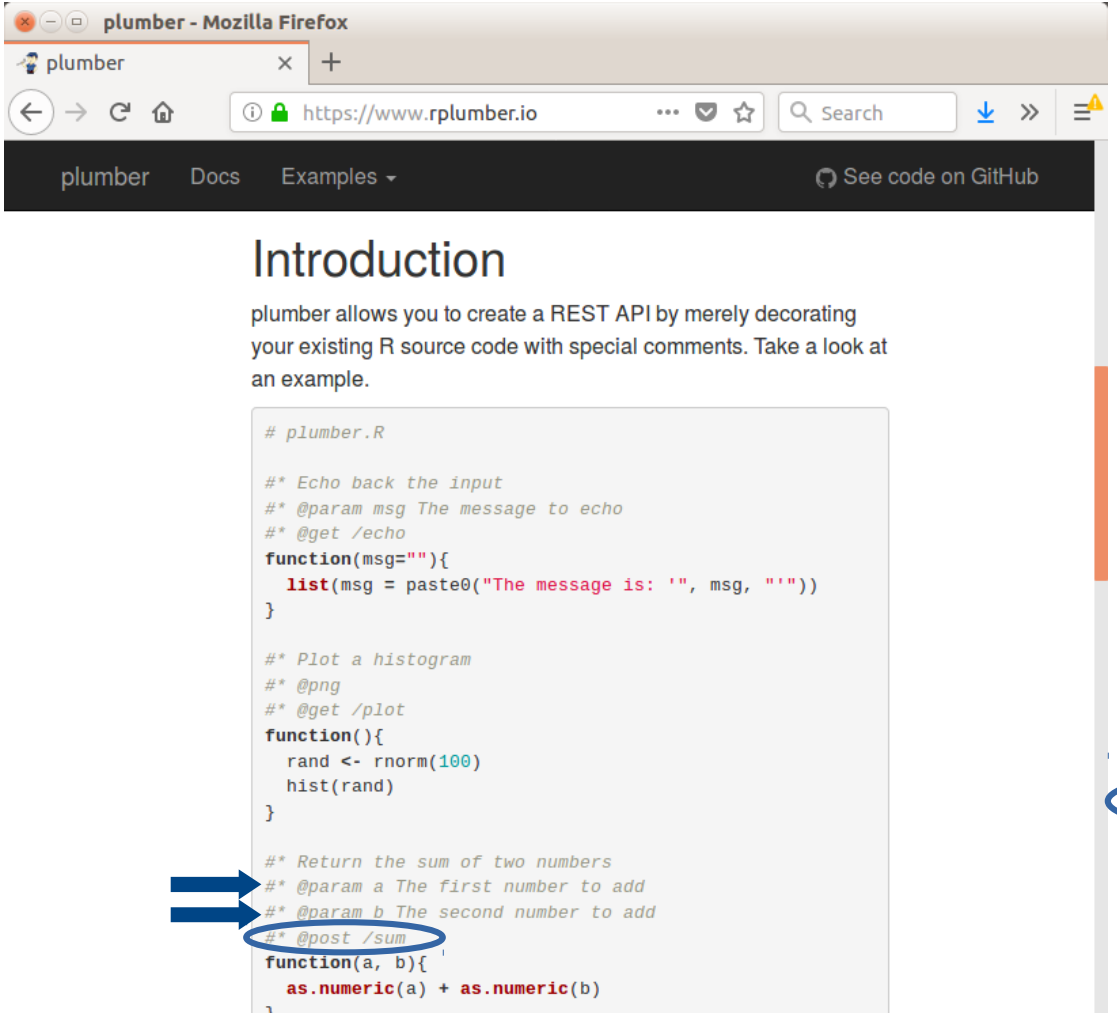
https://shiny.rstudio.com/articles/debugging.html

```
<shiny-server.conf>

run_as shiny;

app_init_timeout 240;
app_idle_timeout 120;

preserve_logs true;

Server {
    listen 12345;
    ...
}
```


```
Dockerfile

…
COPY shiny-server.conf /etc/shiny-server/shiny-server.conf
…
```



https://www.rplumber.io/
https://www.rplumber.io/docs/




```
> library(plumber)
> r <- plumb("plumber.R")  # Where 'plumber.R' is the lo
cation of the file shown above
> r$run(port=8000)
```

You can visit this URL using a browser or a terminal to run your R function and get the results. For instance http://localhost:8000/plot will show you a histogram, and http://localhost:8000/echo?msg=hello will echo back the 'hello' message you provided.

Here we're using `curl` via a Mac/Linux terminal.

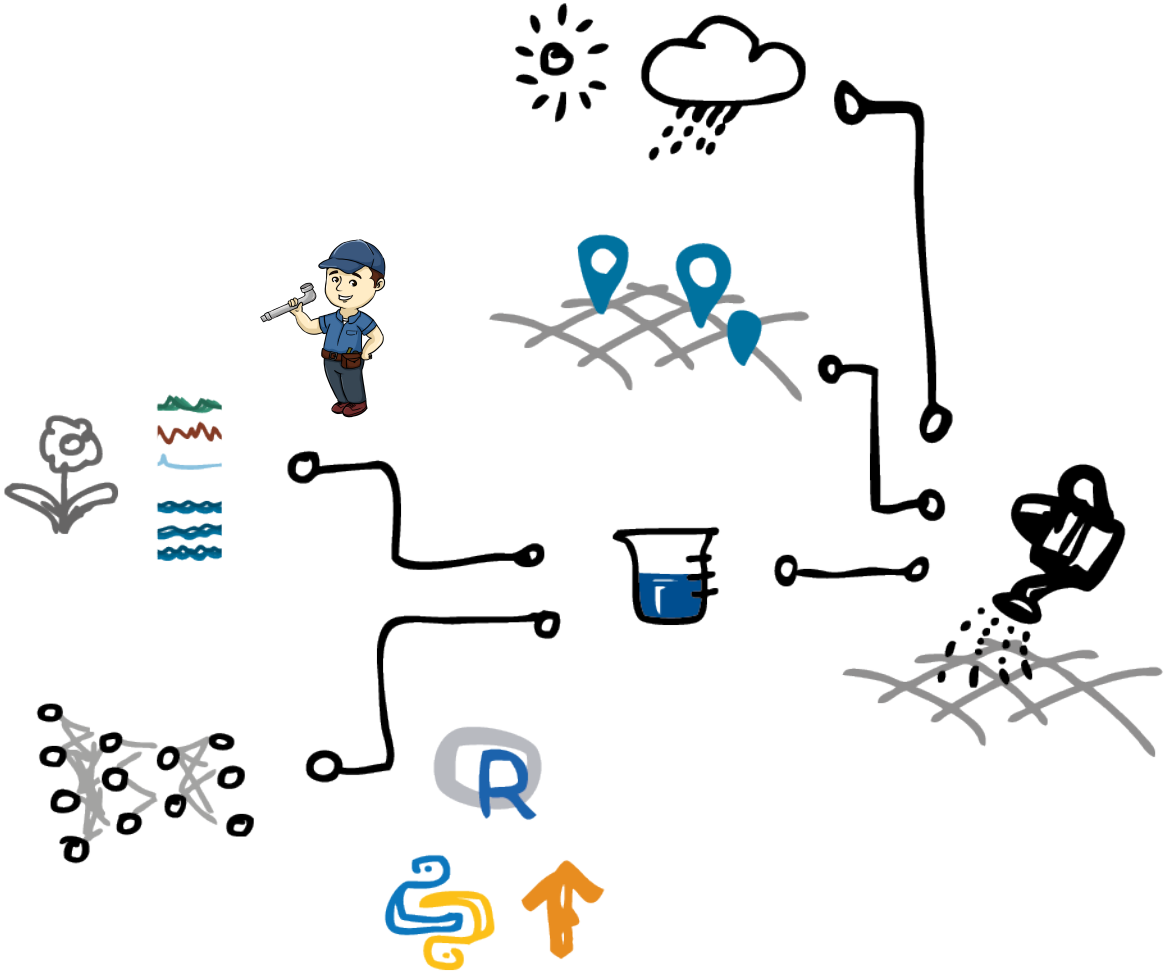
```
$ curl "http://localhost:8000/echo"
 {"msg":["The message is: '"]}
$ curl "http://localhost:8000/echo?msg=hello"
 {"msg":["The message is: 'hello'"]}
```

As you might have guessed, the request's query string parameters are forwarded to the R function as arguments (as character strings).

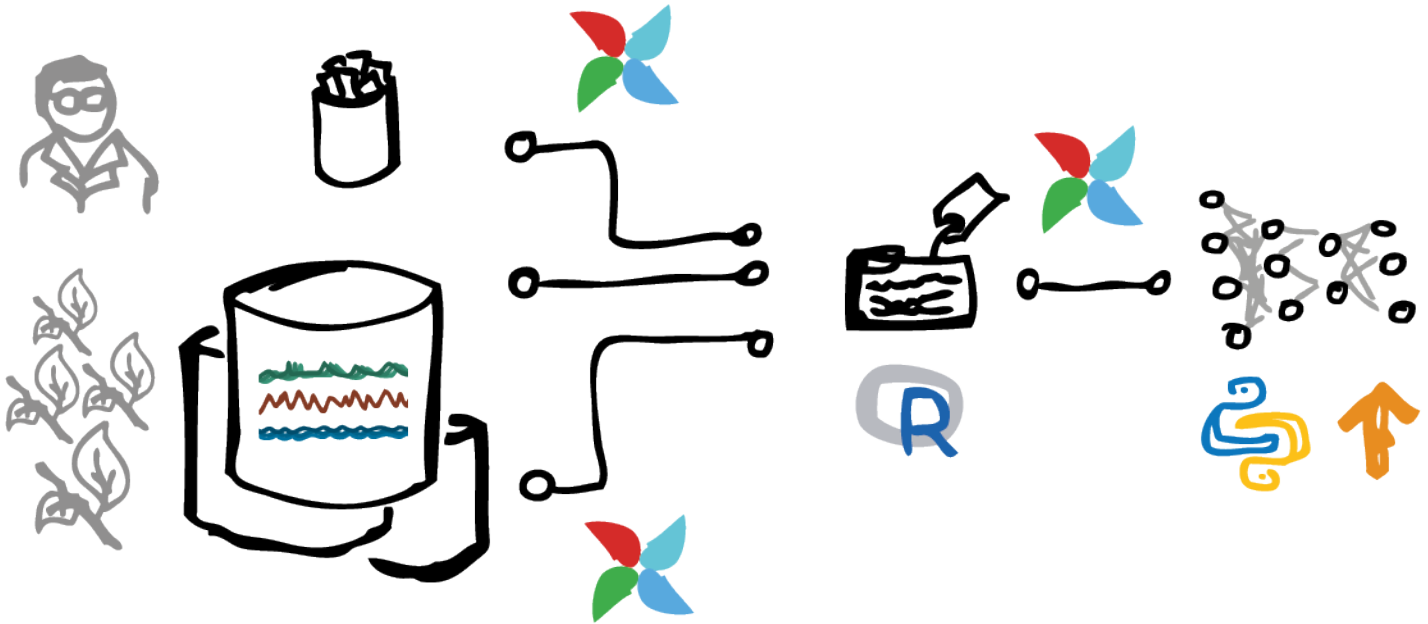
```
$ curl --data "a=4&b=3" "http://localhost:8000/sum"
 [7]
```

If you're still interested, check out our more thorough documentation.

REST api_



Water status prediction as R-based web service_

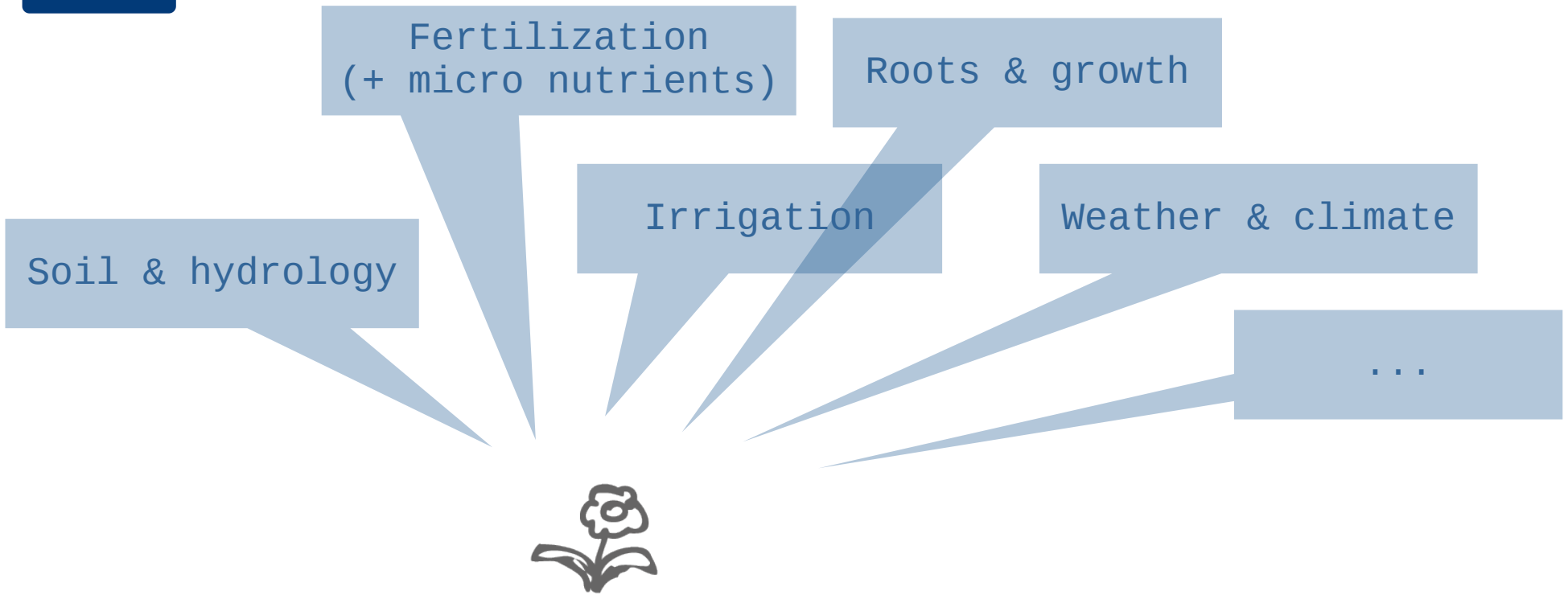


https://airflow.apache.org/
https://airflow.apache.org/ui.html

… from seed to harvest

Yara Digital is starting a new project initiative:

Complete characterization of the crop cycle



Come talk to us!

Claudia Stötzel
Jesus Martinez
Richard Brosi

Knowledge grows


NPK