

FULL STACK DATA SCIENCE IN R

MICHAŁ BURDUKIEWICZ

MI² DATA LAB, WARSAW UNIVERSITY OF TECHNOLOGY
.PROT
WHY R? FOUNDATION

1 Full stack data science

2 Data acquisition and processing

3 Model development

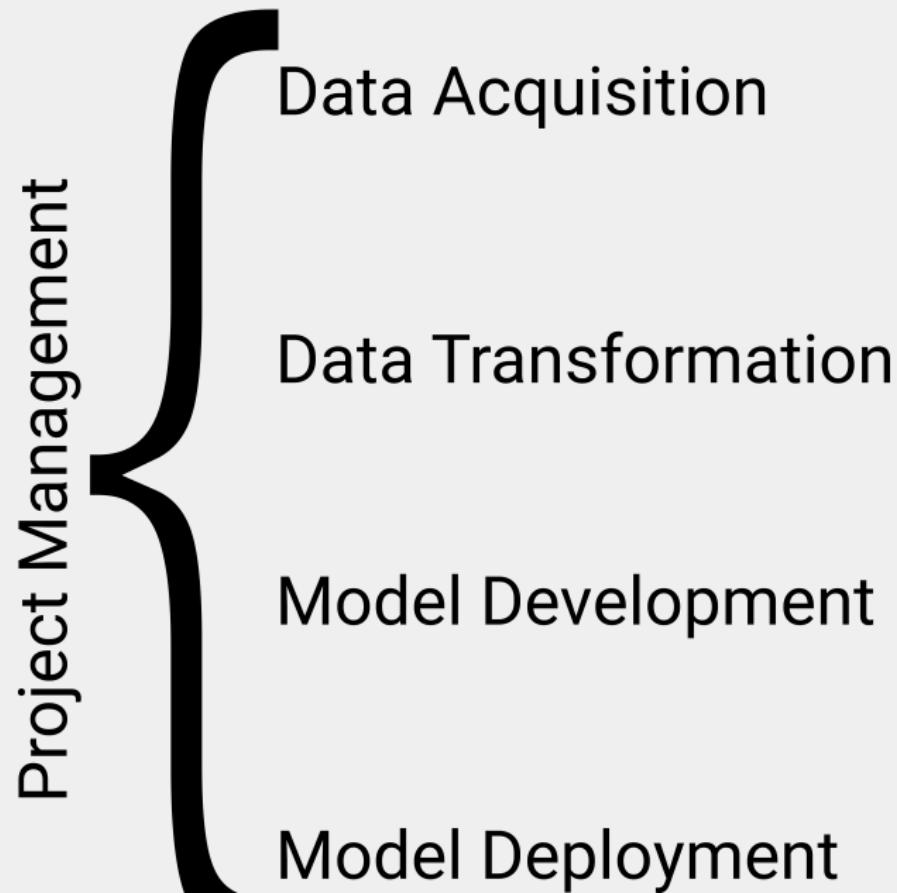
4 Model deployment

5 Project management

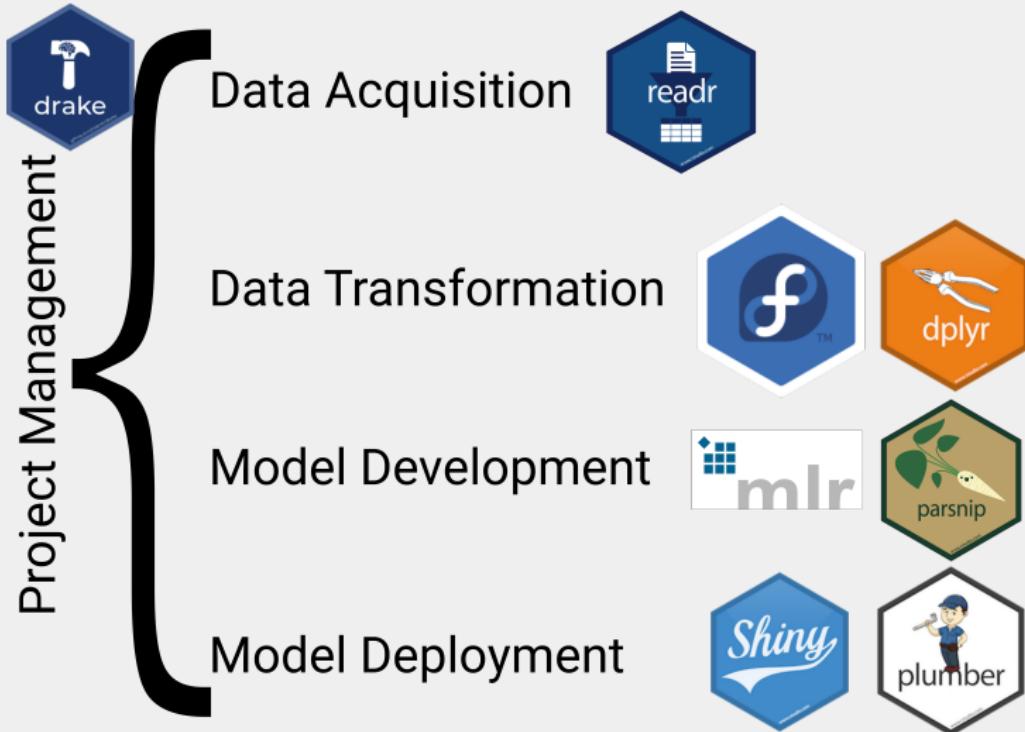
ABOUT ME

- bioinformatician (Warsaw University of Technology,
Brandenburg University of Technology Cottbus-Senftenberg,
.prot),
- founder of the Why R? Foundation and Wrocław R User Group,
- facilitator of McKinsey Tech Meetup.

FULL STACK DATA SCIENCE



FULL STACK DATA SCIENCE IN R



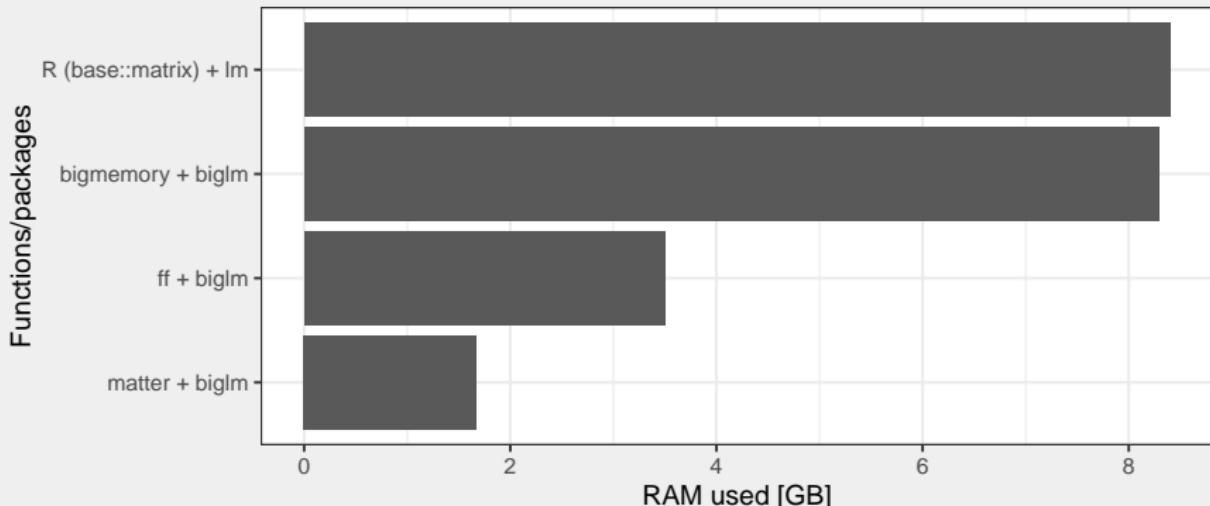
DATA ACQUISITION AND PROCESSING

DATA ACQUISITION

Import data to **R** in a *data.frame* (or similar) format.

Type	Package
Tabular	readr, xlsx, data.table::fread
Relational databases	RPostgreSQL, mongolite
Graph databases (e.g., neo4j)	None

BIG DATA PROCESSING



Comparative performance of matter for linear regression and calculation of the first two principal components on simulated datasets of 1.2 GB.

Source: Bemis KA (2019). matter: A framework for rapid prototyping with binary data on disk. R package version 1.10.0, <https://github.com/kuwisdalu/matter>.

BIG DATA PROCESSING



Memory overhead is the maximum memory used during the execution minus the memory in use upon completion.

Source: Bemis KA (2019). matter: A framework for rapid prototyping with binary data on disk. R package version 1.10.0, <https://github.com/kuwisdalu/matter>.

MODEL DEVELOPMENT

mlr: a standardized interface to machine learning in **R** (by Bernd Bischl and Michael Lang).

Alternatives: **caret**, **parsnip**.

WHY MLR?

- amazing documentation,
- separation of learner and task,
- a wide array of tasks: *Classification*, *Regression*, *Cost-sensitive*, Survival, Clustering, Multilabel, Imbalanced data, Functional data, Spatial data.
- 71 performance measures.
- MBO.

MLR TASKS

```
iris[1L:2, ]  
  
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1         5.1        3.5       1.4        0.2  setosa  
## 2         4.9        3.0       1.4        0.2  setosa  
iris.task  
  
## Supervised task: iris-example  
## Type: classif  
## Target: Species  
## Observations: 150  
## Features:  
##     numerics    factors    ordered functionals  
##             4          0          0          0  
## Missings: FALSE  
## Has weights: FALSE  
## Has blocking: FALSE  
## Has coordinates: FALSE  
## Classes: 3  
##     setosa versicolor virginica  
##           50          50          50  
## Positive class: NA
```

MLR LEARNERS

```
listLearners()
```

```
##          class                      name short.name
## 1      classif.ada                  ada Boosting     ada
## 2  classif.adaboostm1              ada Boosting M1 adaboostm1
## 3 classif.bartMachine Bayesian Additive Regression Trees bartmachine
## 4      classif.binomial            Binomial Regression binomial
## 5      classif.boosting           Adabag Boosting    adabag
## 6      classif.bst                Gradient Boosting   bst
##          package type installed numerics factors ordered missings weights
## 1  ada,rpart classif    FALSE    TRUE    TRUE  FALSE  FALSE  FALSE
## 2      RWeka classif    FALSE    TRUE    TRUE  FALSE  FALSE  FALSE
## 3  bartMachine classif    FALSE    TRUE    TRUE  FALSE  TRUE  FALSE
## 4      stats classif     TRUE    TRUE    TRUE  FALSE  FALSE  TRUE
## 5 adabag,rpart classif    FALSE    TRUE    TRUE  FALSE  TRUE  FALSE
## 6    bst,rpart classif    FALSE    TRUE  FALSE  FALSE  FALSE  FALSE
##          prob oneclass twoclass multiclass class.weights featimp oobpreds
## 1    TRUE     FALSE    TRUE    FALSE        FALSE  FALSE  FALSE
## 2    TRUE     FALSE    TRUE     TRUE        FALSE  FALSE  FALSE
## 3    TRUE     FALSE    TRUE    FALSE        FALSE  FALSE  FALSE
## 4    TRUE     FALSE    TRUE    FALSE        FALSE  FALSE  FALSE
## 5    TRUE     FALSE    TRUE     TRUE        FALSE  TRUE  FALSE
## 6   FALSE     FALSE    TRUE    FALSE        FALSE  FALSE  FALSE
##          functionals single.functional    se licens rcens icens
## 1      FALSE        FALSE FALSE FALSE FALSE FALSE
## 2      FALSE        FALSE FALSE FALSE FALSE FALSE
## 3      FALSE        FALSE FALSE FALSE FALSE FALSE
## 4      FALSE        FALSE FALSE FALSE FALSE FALSE
## 5      FALSE        FALSE FALSE FALSE FALSE FALSE
## 6      FALSE        FALSE FALSE FALSE FALSE FALSE
## ... (#rows: 167, #cols: 24)
```

MLR LEARNERS

```
listLearners(obj = iris.task)
```

```
##          class name
## 1    classif.cforest Random forest based on conditional inference trees
## 2    classif.ctree      Conditional Inference Trees
## 3  classif.featureless      Featureless classifier
## 4    classif.fnn        Fast k-Nearest Neighbour
## 5  classif.gausspr      Gaussian Processes
## 6    classif.knn       k-Nearest Neighbor
##   short.name package type installed numerics factors ordered missings
## 1     cforest   party classif    TRUE    TRUE    TRUE    TRUE
## 2      ctree   party classif    TRUE    TRUE    TRUE    TRUE
## 3 featureless    mlr classif    TRUE    TRUE    TRUE    TRUE
## 4       fnn     FNN classif    TRUE    TRUE FALSE FALSE
## 5  gaussspr kernlab classif    TRUE    TRUE FALSE FALSE
## 6      knn    class classif    TRUE    TRUE FALSE FALSE
##   weights prob oneclass twoclass multiclass class.weights featimp
## 1    TRUE  TRUE    FALSE    TRUE    TRUE    FALSE    TRUE
## 2    TRUE  TRUE    FALSE    TRUE    TRUE    FALSE FALSE
## 3   FALSE  TRUE    FALSE    TRUE    TRUE    FALSE FALSE
## 4   FALSE FALSE    FALSE    TRUE    TRUE    FALSE FALSE
## 5   FALSE TRUE    FALSE    TRUE    TRUE    FALSE FALSE
## 6   FALSE FALSE    FALSE    TRUE    TRUE    FALSE FALSE
##   oobpreds functionals single.functional    se lrens rcens icens
## 1    FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE FALSE
## 2    FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE FALSE
## 3    FALSE    TRUE    FALSE FALSE FALSE FALSE FALSE FALSE
## 4    FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE FALSE
## 5    FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE FALSE
## 6    FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE FALSE
## ... (#rows: 19, #cols: 24)
```

MLR LEARNERS

```
iris.task <- makeClassifTask(data = iris, target = "Species")

lrn_rng <- makeLearner("classif.ranger")
# ranger: the fastest random forest implementation in R
# Marvin Wright, the author of ranger, is a keynote speaker of Why R? 2019

cv <- makeResampleDesc("CV", iters = 3)

resample(learner = lrn_rng, task = iris.task, resampling = cv)

## Resampling: cross-validation

## Measures: mmce

## [Resample] iter 1: 0.0400000
## [Resample] iter 2: 0.0600000
## [Resample] iter 3: 0.0400000

##
## Aggregated Result: mmce.test.mean=0.0466667

##
## Resample Result
## Task: iris
## Learner: classif.ranger
## Aggr perf: mmce.test.mean=0.0466667
## Runtime: 0.0884285
```

BAYESIAN OPTIMIZATION

$$\max_{x \in \mathbb{R}} f(x)$$

The *posterior* probability of a model (or theory, or hypothesis) M given evidence (or data, or observations) E is proportional to the likelihood of E given M multiplied by the *prior* probability of M :

$$P(M|E) \propto P(E|M)P(M)$$

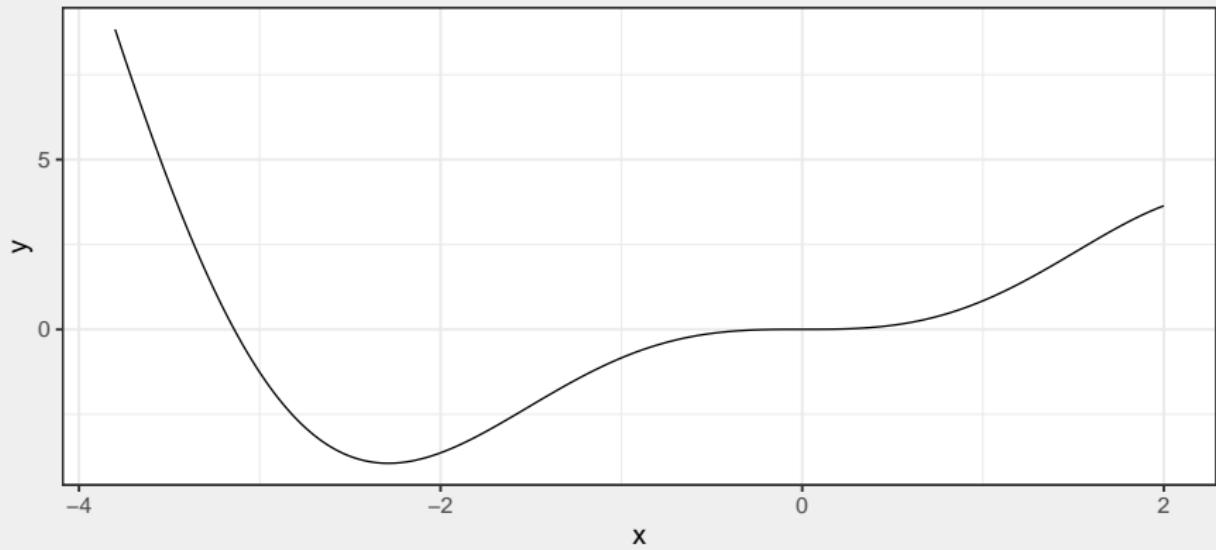
BAYESIAN OPTIMIZATION

Find local extrema of arbitrary functions.

BAYESIAN OPTIMIZATION

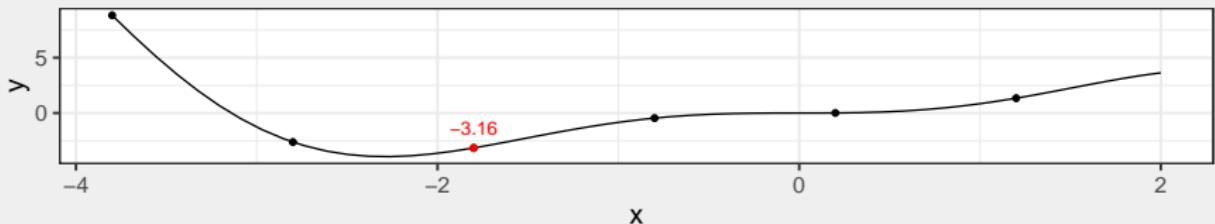
$$y = \sin(x) \times x^2$$

Local minimum: -3.9453.

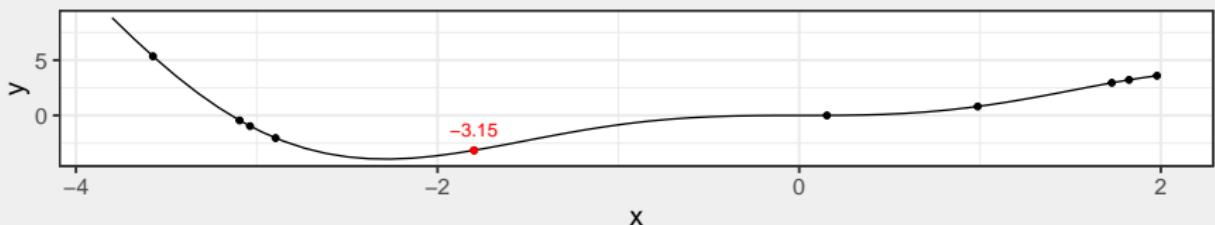


BAYESIAN OPTIMIZATION

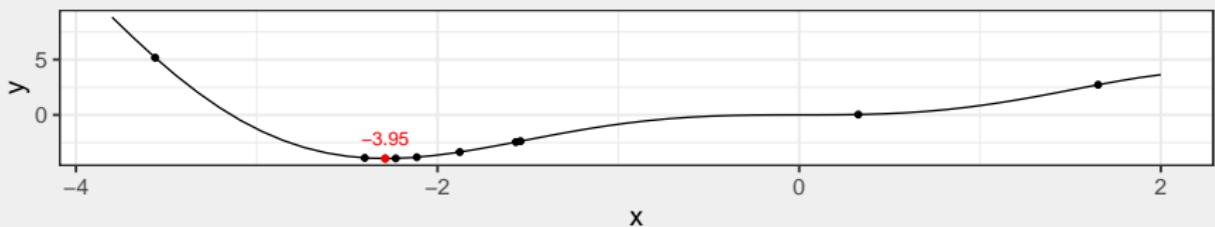
Grid search



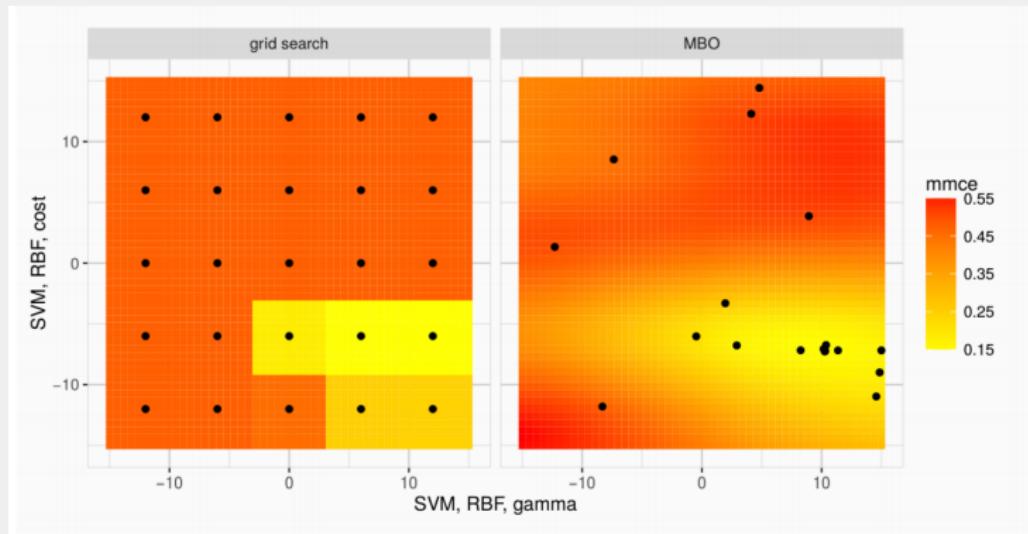
Random search



MBO



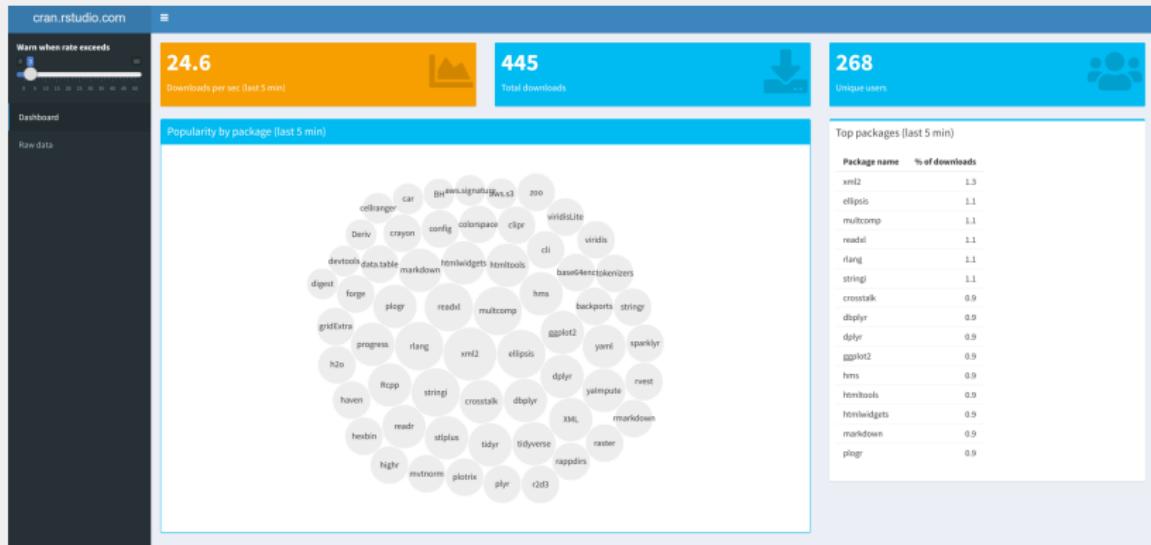
BAYSIAN OPTIMIZATION



Source: Bernd Bischl

MODEL DEPLOYMENT

SHINY



Limitations:

1. stability decreases with the number of concurrent users,
2. speed.

SHINYPROXY AND KUBERNETES

ShinyProxy: docker-based containers for Shiny apps.

ShinyProxy containers can be themselves run in a container and deployed clustered container managers (such as Kubernetes or Swarm).

PROJECT MANAGEMENT

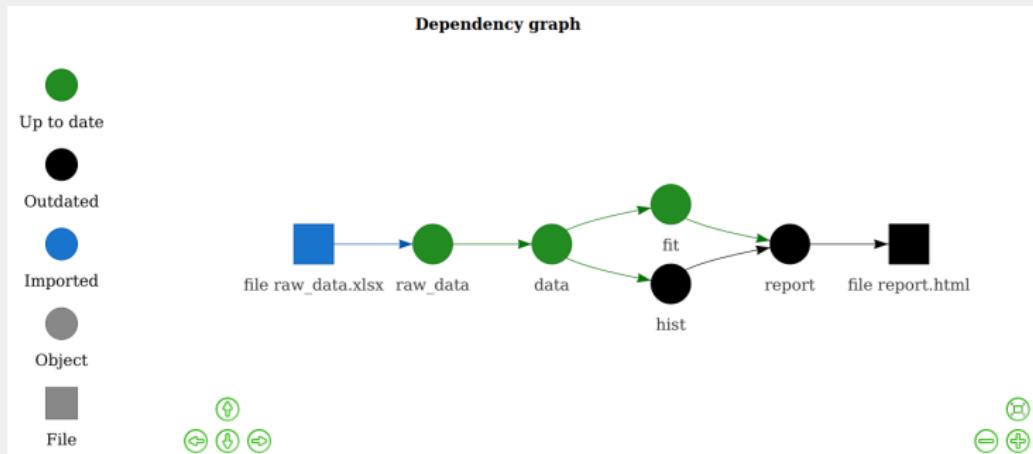
make for R projects.

GNU Make is a tool which controls the generation of executables and other non-source files of a program from the program's source files.

Source: GNU Make web page.

```
plan <- drake_plan(  
  raw_data = read_excel(file_in("raw_data.xlsx")),  
  data = raw_data %>%  
    mutate(Species = fct_inorder(Species)),  
  hist = create_plot(data),  
  fit = lm(Sepal.Width ~ Petal.Width + Species, data),  
  report = rmarkdown::render(  
    knitr_in("report.Rmd"),  
    output_file = file_out("report.html"),  
    quiet = TRUE  
)  
)
```

DRAKE



Parallelization:

```
# multicore scaling
make(plan, jobs = 12)

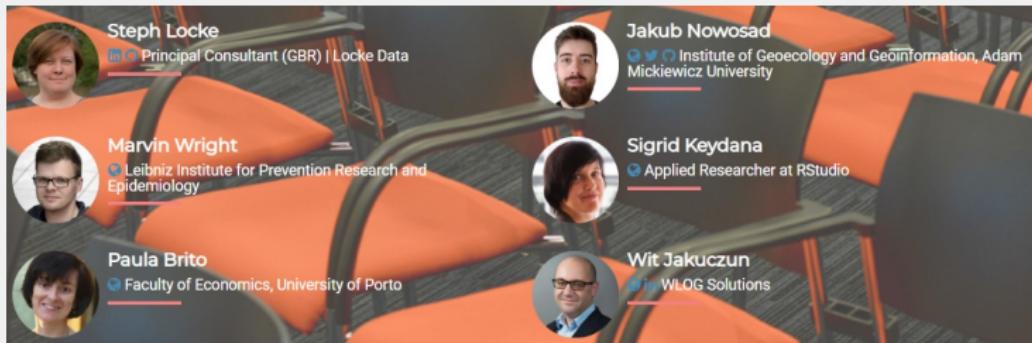
# or enable supercomputing clusters with
# the future package
drake_hpc_template_file("torque_batchtools tmpl")
library(future.batchtools)
future::plan(batchtools_torque,
            template = "torque_batchtools tmpl",
            workers = 8000)
make(plan, parallelism = "future_lapply")
```

WHY R? 2019



whyr.pl/2019

WHY R? 2019



Workshops: Jan Gorecki (data.table, H₂O), Matteo Fasiolo (University of Bristol).

whyr.pl/2019

MI² DATA LAB

MI² Data Lab (<https://mi2.mini.pw.edu.pl/>), Faculty of Mathematics and Computer Science, Warsaw University of Technology.



Contact: michal@whyr.pl.

Presentation: <https://github.com/michbur/fsds>.