

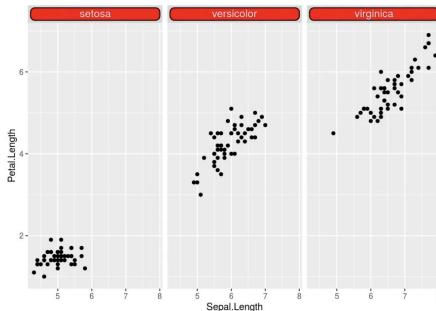


# Using Pins to Ensure Reproducibility with Datasets

@javierluraschi  
useR 2020 - RStudio PBC

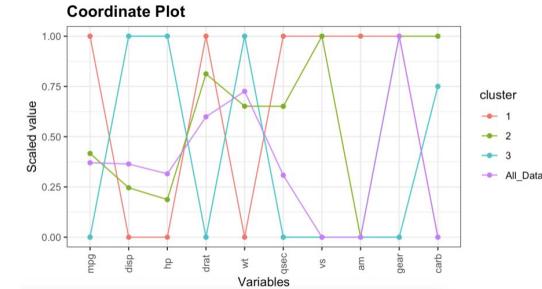
# Data Workflows - Today

June 16th posts in R-bloggers use iris, mtcars, have missing data paths and require manually downloading datasets.



```
#####
# ## IMPORT RAW DATA
32. log_info("Loading data")
33.
34. mydat = fread(data_path)
35.
36. #####
37.
```

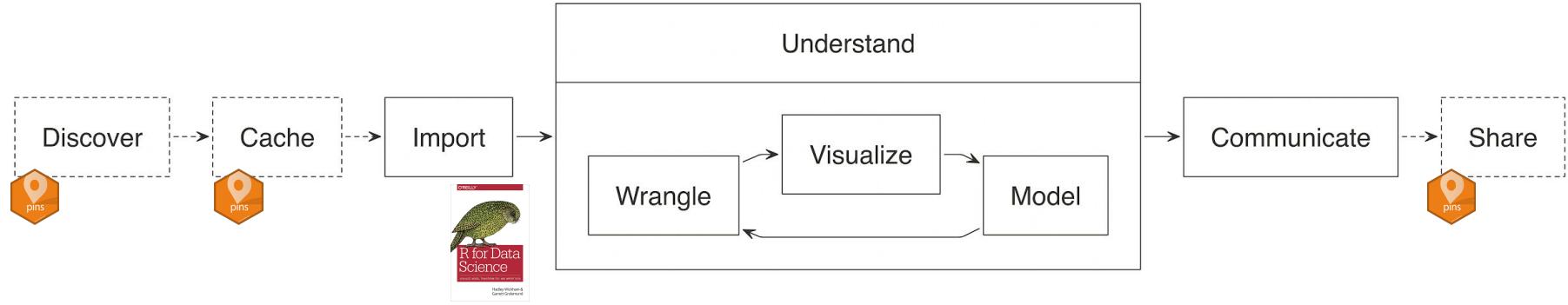
```
coord_plot(data=mtcars2, group_var="cluster",
group_func=median, print_table=TRUE)
```



```
# https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-relatives-a-lepidemie-de-covid-19/
fichier_covid <- "donnees/covid.csv"
```

# Data Workflows - In Data Science

We know from R for Data Science that in a typical project we usually need to import, tidy, understand and communicate knowledge.



However, it is often also required for us to discover which dataset to use, cache it locally, and share our datasets with colleagues.

# Data Workflows - Reproducibility

But more importantly, properly caching and sharing our datasets allows us to make Data Science more reproducible in tools like R Markdown and Jupyter.

The collage includes:

- A Jupyter logo with a hexagonal pattern background.
- A screenshot of the Jupyter Notebook interface showing code cells and output.
- A screenshot of the Jupyter website ([jupyter.org](https://jupyter.org)) featuring the Jupyter logo and navigation links.
- A screenshot of the R Markdown from R Studio interface, showing a plot of a volcano dataset.
- A large central image showing a plot of the Lorenz system with mathematical equations above it.
- A screenshot of the R Markdown website ([rmarkdown.rstudio.com](https://rmarkdown.rstudio.com)) with the tagline "Analyze. Share. Reproduce."
- A screenshot of the RStudio IDE interface showing a code editor and a plot.

**Language of choice**  
Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.

R Markdown documents are fully reproducible. Use a productive [notebook interface](#) to weave together narrative text and code to produce elegantly formatted output. Use [multiple languages](#) including R, Python, and SQL.

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Your data tells a story. Tell it with R Markdown. Turn your analyses into high quality documents, reports, presentations and dashboards.

# Data Workflows - Ideally



- Learn a **single tool** regardless of where data lives.
- Easy **switch storage** and share it anywhere.
- Easy to build **automated workflows**.
- Can share datasets of **any size**.
- Can easily use **interesting datasets!**
- Data loads **fast** and works **offline**.

# What is the package?



- **Pin** remote resources locally with `pin()`, work offline and cache results.
- **Discover** new resources across different boards using `pin_find()`.
- **Share resources** in local folders, GitHub, Kaggle, and RStudio Connect by registering new boards with `board_register()`.

## Links

Download from CRAN at  
<https://cloud.r-project.org/package=pins>

Browse source code at  
<https://github.com/rstudio/pins>

Report a bug at  
<https://github.com/rstudio/pins/issues>

## License

Apache License 2.0

## Developers

Javier Luraschi  
Author, maintainer

[All authors...](#)

## Dev status

build passing

CRAN 0.4.1

codecov 77%

downloads 3954/month

lifecycle maturing

chat on gitter

stars 107

# How can I use Pins?

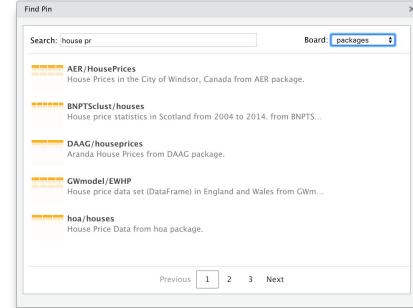
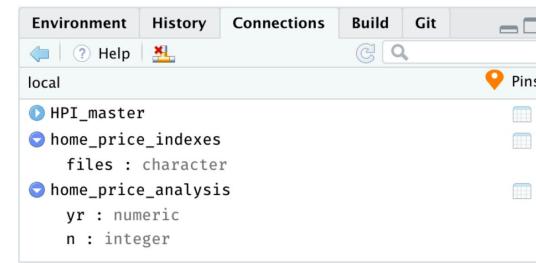
Use `pin()` to save objects, `pin_get()` to retrieve them and `pin_find()` to search.

```
pin(1:10, name = "numbers")
pin_get("numbers")

pin("http://cs231n.stanford.edu/tiny-imagenet-200.zip", name = "tinyimagenet")
pin_get("tinyimagenet")

pin_find()
```

It also supports various RStudio add-ins!



# What are Boards?

A board is a storage location, like your local hard drive, but there are also many additional places where you can store your datasets like RStudio Connect, Microsoft Azure, Google Cloud, Digital Ocean, Amazon S3, GitHub and Kaggle.



# How can I use Boards?

To use pins with boards, first register the board, then specify the board in pin().

```
board_register("_____")  
  
pin(1:10, name = "numbers", board = "_____")  
pin_get("numbers")  
  
pin("http://cs231n.stanford.edu/tiny-imagenet-200.zip ",  
    name = "tinyimagenet", board = "_____")  
pin_get("tinyimagenet")  
  
pin_find()
```

To use pins with boards, first register the board, then specify the board in pin().

# Boards and Pins

Pins works in the same way across all storage services; however, each service provides a slightly different user interface, permissions, workflows, etc.

The image displays four screenshots illustrating the 'pins' interface across different platforms:

- Google Sheets:** Shows a table titled 'mtcars' with columns: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. A sidebar on the left shows 'My Pins' and 'Recent'. A note at the top says 'This document is shared with "javierluraschi".' A 'Share' button is visible at the top right.
- AWS S3:** Shows the 'pinscontainer' bucket in the 'Amazon S3' console. It lists objects: 'iris', 'mtcars', and 'data.txt'. The 'iris' object has a preview showing a scatter plot. The 'Actions' dropdown menu is open. The top navigation bar shows 'Overview', 'Properties', 'Permissions', and 'Management'.
- Databricks:** Shows a notebook titled 'Content / mtcars'. It contains a code cell with R code: 'library(ggplot2); ggplot(mtcars, aes(wt, mpg)) + geom\_point()'. Below the code cell is a preview of a scatter plot. The sidebar on the left shows 'Download File' and 'My Pins'.
- Github:** Shows a repository page for 'javierluraschi/mtcars'. Key statistics are: 466 commits, 2 branches, 0 releases, 1 contributor. A pull request summary states: 'This branch is 464 commits ahead, 1 commit behind master.' Recent activity includes a pull request from 'javierluraschi' and updates to 'iris', 'mtcars', and 'data.txt' files by 'javierluraschi'.

# Versioning

You can optionally enable or disable tracking versions, some boards support this by default, in others you opt-in when the board is registered.

```
board_register("github", repo = "javierluraschi/datasets", branch = "datasets")

pin(iris, name = "versioned", board = "github", commit = "use iris...")
pin(mtcars, name = "versioned", board = "github", commit = "slight
preference...")

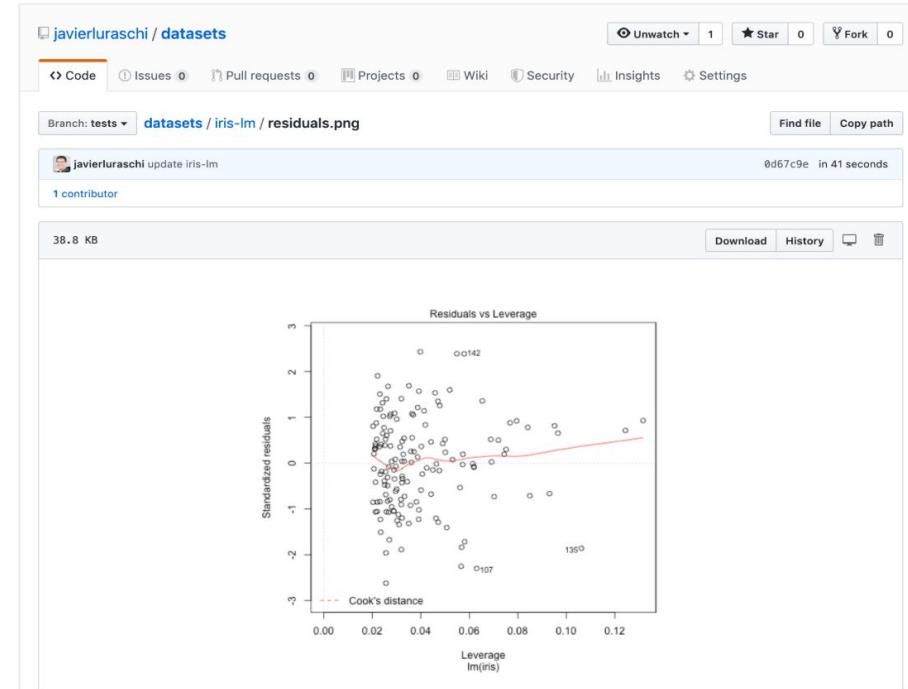
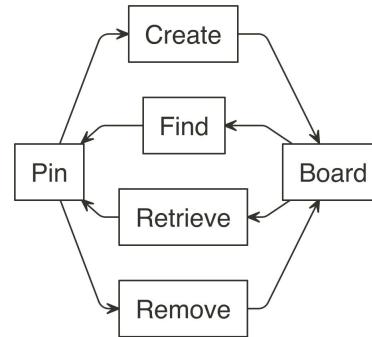
pin_versions("versioned", board = "github")

# A tibble: 2 x 4
  version created      author      message
  <chr>    <chr>        <chr>        <chr>
1 6e6c320 2020-04-02T21:28:07Z javierluraschi slight preference to mtcars
2 01f8ddf 2020-04-02T21:27:59Z javierluraschi use iris as the main dataset
```

```
pin_get("versioned", version = "6e6c320")
```

# Extending Pins and Boards

You can store R objects with specific file formats and also create new boards by implementing the pins API.



# Website Boards

Pins uses [datatxt.org](http://datatxt.org) to describe datasets, a simple YAML file to locate resources and optional metadata. You can then auto-generate data websites (like [cellar.kasa.ai](http://cellar.kasa.ai)) with the datatxt package.

The screenshot shows a web page with a header "Summary" and tabs for "Cellar" and "Datasets". Below the tabs is a section titled "Datasets" containing a bulleted list of datasets:

- [FIMA NFIP Redacted Claims Data Set](#)
- [French Third-Party Liability \(Claims\)](#)
- [French Third-Party Liability \(Policies\)](#)
- [Schedule P Data](#)
- [SOA Lapse Study Data](#)

On the right side, there is a summary table with the following data:

Name	Piped data
Number of rows	26639
Number of columns	2
Column type frequency:	
numeric	2
Group variables	
None	

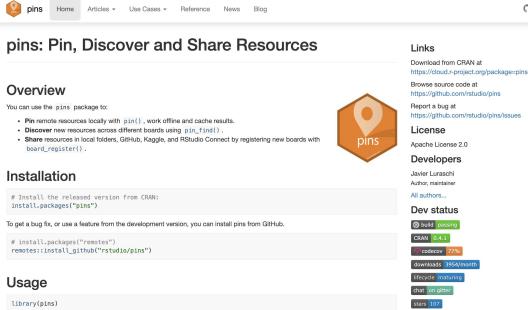
Below the summary table is a "Variable type: numeric" section with a table showing statistical summaries for two columns:

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
policy_id	0	1	2279863.83	1577201.81	139	1087642.50	2137413	3180162.00	6113971	
claim_amount	0	1	2278.54	29297.48	1	686.81	1172	1228.08	4075401	

At the bottom, there is a "Source" section with the text "R package 'CASdatasets', <http://dutango.free.fr/pub/RRepos/web/CASdatasets-index.html>".

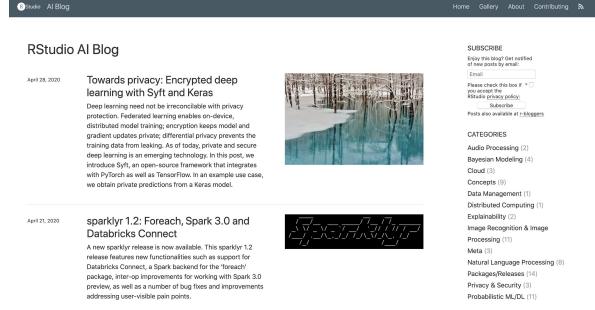
```
pins::board_register("http://data.cellar.kasa.ai")
```

# Thank You!



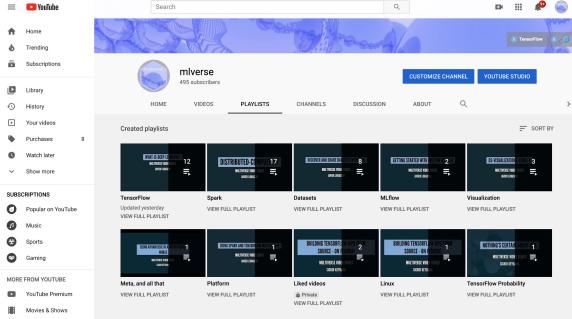
The screenshot shows the homepage of the pins R package. It features a header with navigation links: Home, Articles, Use Cases, Reference, News, and Blog. Below the header is a section titled "pins: Pin, Discover and Share Resources". This section includes an "Overview" section with a note about the package's purpose, an "Installation" section with R code for CRAN and GitHub installation, and a "Usage" section with a library(pins) command. On the right side, there's a sidebar with links to the CRAN page, GitHub source code, and a pinned issue. A large orange hexagonal logo with the word "pins" is positioned on the left.

[pins.rstudio.com](https://pins.rstudio.com)



The screenshot shows a blog post from the RStudio AI Blog. The title is "Towards privacy: Encrypted deep learning with Syft and Keras". The post discusses Federated learning and its role in privacy protection. It includes a screenshot of a neural network diagram. The post is dated April 23, 2020. Below the main content is a sidebar with categories like Data Management, Cloud, and Concepts, and a "SUBSCRIBE" form.

[blogs.rstudio.com/ai](https://blogs.rstudio.com/ai/2020/04/towards-privacy-encrypted-deep-learning-with-syft-and-keras/)



The screenshot shows the YouTube channel page for "mlverse". The channel has 403 subscribers. It features a banner image of a brain, a video thumbnail for "MLflow", and a grid of other video thumbnails. The sidebar includes sections for Home, Videos, Playlists, Channels, Discussion, and About. The "PLAYLISTS" tab is selected, showing playlists for TensorFlow, Spark, Datasets, MLflow, and Visualization.

[youtube.com/c/mlverse](https://www.youtube.com/c/mlverse)



@javierluraschi



@kevinykuo



@alexkgold



@dfalbel



@zkajdan



@yI790