

What is



?

Principally a collaborative software development project

But it is also:

- a software repository
- a bioinformatics support site
- data repository
- publisher for supplementary materials
- source for tutorials and instructional documentation

Managed and maintained by a core team of ~6 people, with contributions coming from all over the world



What is



?

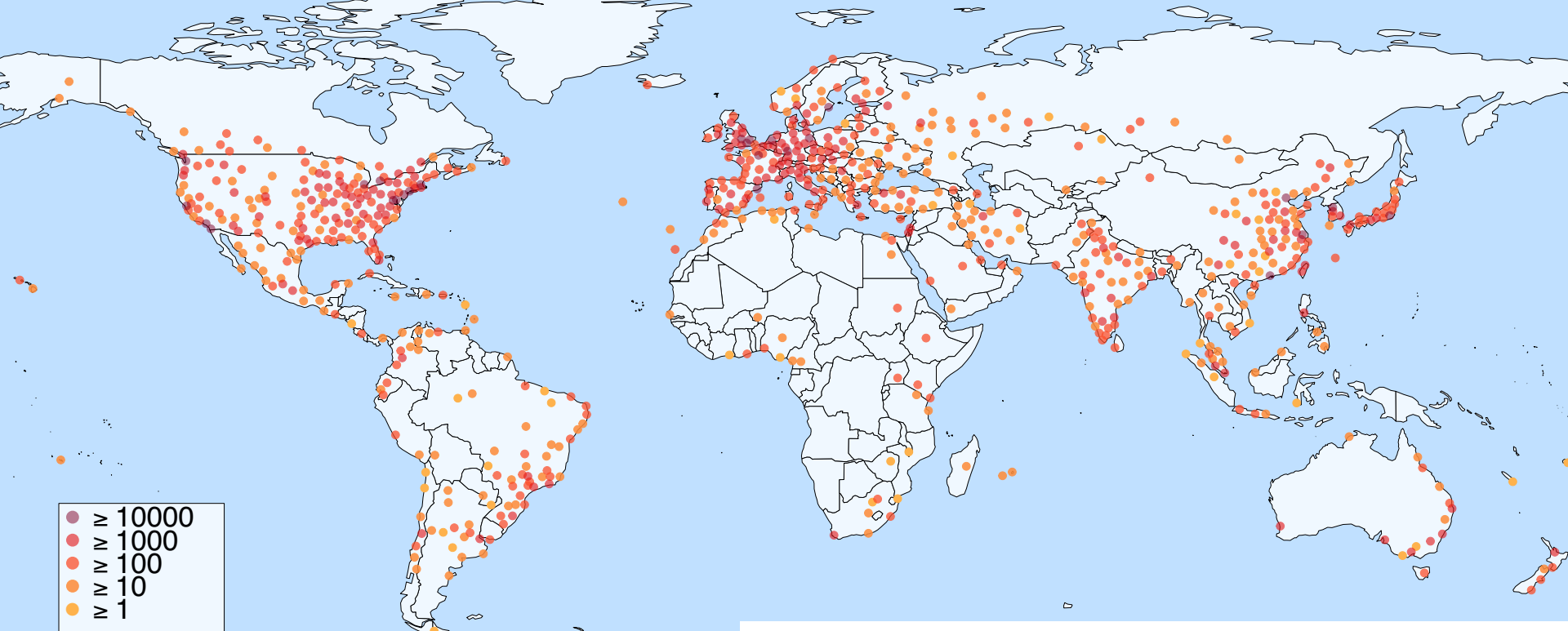
Started 2001 as a platform for analysis & understanding of microarray data

More than 1,600 packages. Domains of expertise:

- Sequencing (RNASeq, ChIPSeq, single-cell, called variants, ...)
- Microarrays (methylation, expression, copy number, ...)
- Flow cytometry
- Proteomics
- Multi-Omics data integration

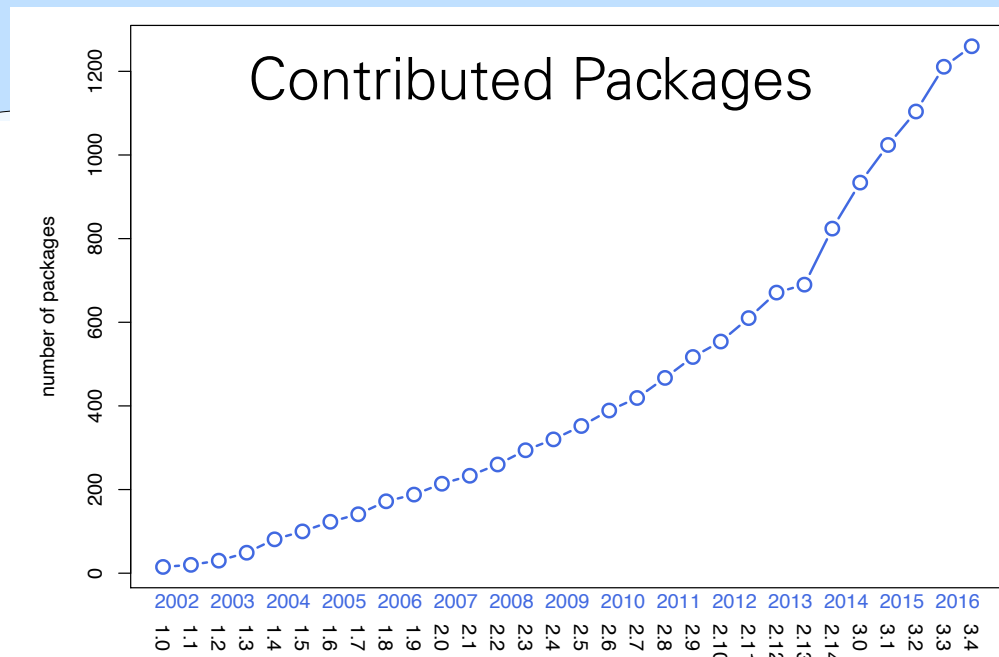
## Important themes

- Reproducible research
- Interoperability between packages & workflows  
... even from different authors
- Usability



Site users - per location

World largest bioinformatics project  
 10,000s users  
 >18,000 papers in PubmedCentral



Collaborative  
and distributed development  
Open source

Lower barrier of entry  
Train  
Turn users into  
developers

Data import,  
preprocessing  
Integration of  
data types

Based on R  
Interoperable components  
Rapid development  
Code re-use

Publication of software  
Computational reproducibility

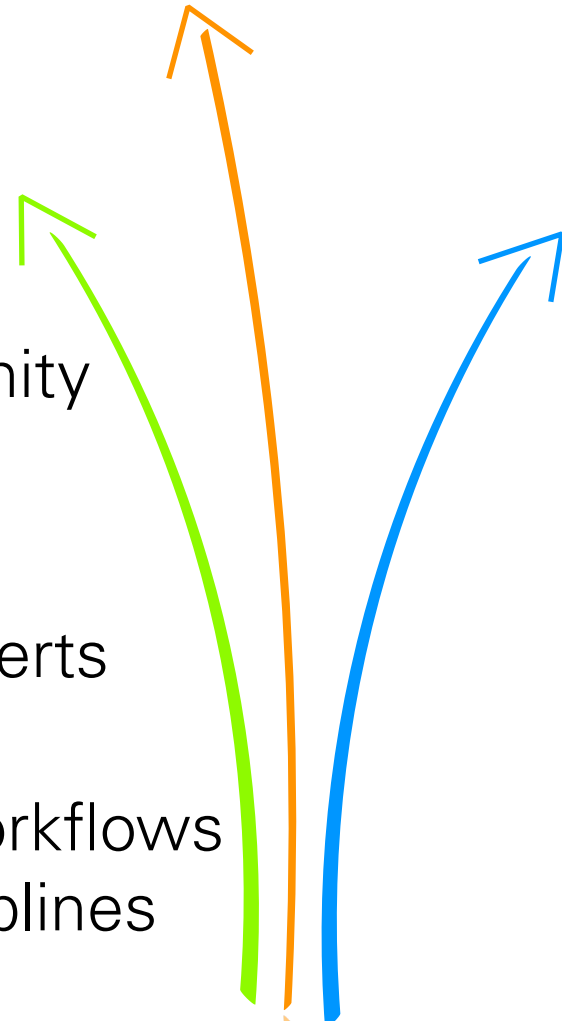
# Motivating principles

→ Provide a compelling **user** experience:  
documentation, demos, tutorials

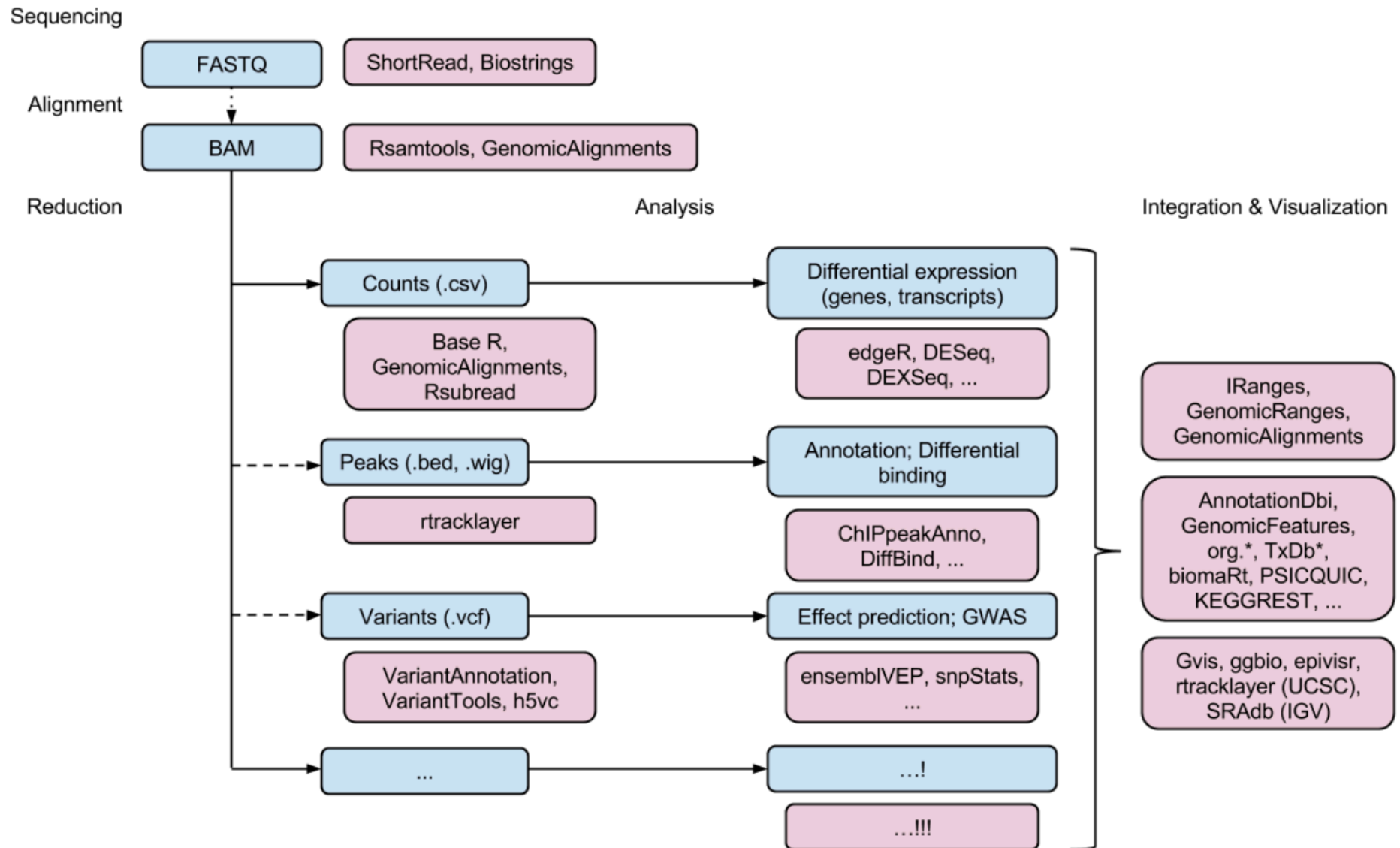
- workflows
- package vignettes
- function manual pages

→ Support active & open **developer** community

- training on software development & programming techniques
- distributed development by domain experts  
(→ interoperability)
- common data structures that enable workflows  
integrating multiple data types and disciplines



# Workflows for HT Sequencing



# Bioconductor Classes: e.g., Summarized Experiment



- Synchronized handling of multiple tables / matrices (subsetting, dangling pointers)
- Encapsulation: separation of interface from implementation
- Validity checking: enforces contracts with the user
- Specialized highly efficient methods for manipulation (e.g. GRanges class)

```
subsetByOverlaps(se, roi) assays(se)  
assay(se, n = 2) assay(subsetByOverlaps(se, roi))  
assay(se[, se$dex == "trt"])
```

# Annotation & Datasets

Annotation of genes, transcripts, proteins, pathways, metabolites; GO, Reactome, Pubmed, ...

Sequences

You don't have to download text files from NCBI / EBI and parse them into R - use ready made packages with nice interfaces.

[ExperimentHub](#): published datasets already curated into efficient R objects, with documentation



# Modes of documentation

- Manual Pages (for each function)
- Vignettes: Narrative overviews on what you can do with a package
- Workflows: end-to-end descriptions of a scientific question
- F1000Research papers, Bioinformatics application notes: peer-reviewed, citable

# Support Forum -

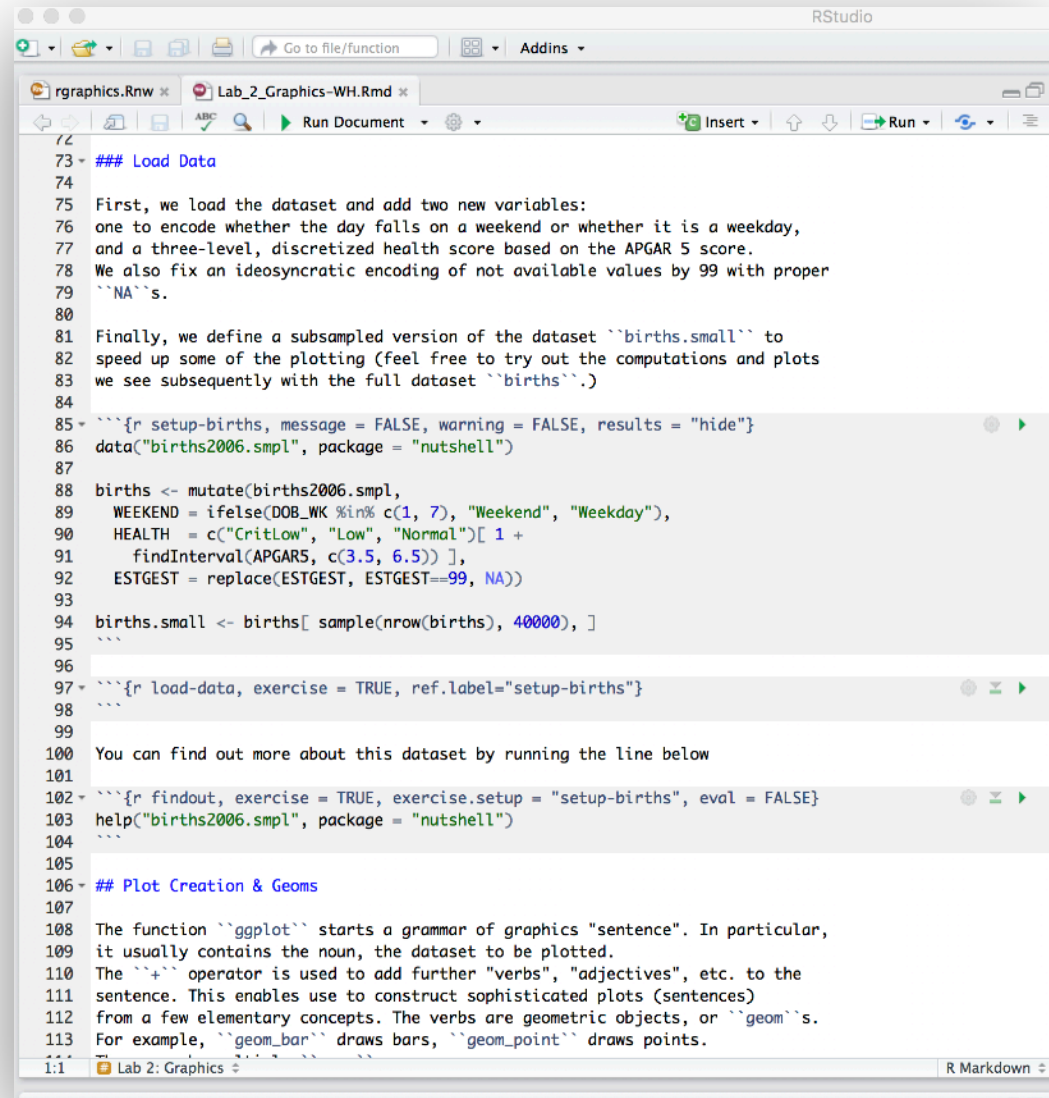
## Etiquette (Posting Guide)

- Make sure you use most recent versions
- Read the documentation
- Use Google to see if a similar question has already been asked
- Prepare a minimal working example and post its code
- Remember your manners when reporting “bugs” or “missing features”
- Use descriptive subject line and precise language
- Post `'devtools::session_info()'`

<https://www.bioconductor.org/help/support/posting-guide/>

# Scientific software should be assessed by similar criteria as a scientific publication

- Reproducible
- Peer-reviewed
- Easy to access by other researchers & society
- Builds on the work of others
- Others will build their work on top of it



```
72
73 ## Load Data
74
75 First, we load the dataset and add two new variables:
76 one to encode whether the day falls on a weekend or whether it is a weekday,
77 and a three-level, discretized health score based on the APGAR 5 score.
78 We also fix an idiosyncratic encoding of not available values by 99 with proper
79 ``NA``s.
80
81 Finally, we define a subsampled version of the dataset ``births.small`` to
82 speed up some of the plotting (feel free to try out the computations and plots
83 we see subsequently with the full dataset ``births``.)
84
85 ```{r setup-births, message = FALSE, warning = FALSE, results = "hide"}
86 data("births2006.smpl", package = "nutshell")
87
88 births <- mutate(births2006.smpl,
89   WEEKEND = ifelse(DOB_WK %in% c(1, 7), "Weekend", "Weekday"),
90   HEALTH = c("CritLow", "Low", "Normal")[ 1 +
91     findInterval(APGAR5, c(3.5, 6.5)) ],
92   ESTGEST = replace(ESTGEST, ESTGEST==99, NA))
93
94 births.small <- births[ sample(nrow(births), 40000), ]
95 ```
96
97 ```{r load-data, exercise = TRUE, ref.label="setup-births"}
98 ```
99
100 You can find out more about this dataset by running the line below
101
102 ```{r findout, exercise = TRUE, exercise.setup = "setup-births", eval = FALSE}
103 help("births2006.smpl", package = "nutshell")
104 ```
105
106 ## Plot Creation & Geoms
107
108 The function ``ggplot`` starts a grammar of graphics "sentence". In particular,
109 it usually contains the noun, the dataset to be plotted.
110 The ``+`` operator is used to add further "verbs", "adjectives", etc. to the
111 sentence. This enables use to construct sophisticated plots (sentences)
112 from a few elementary concepts. The verbs are geometric objects, or ``geom``s.
113 For example, ``geom_bar`` draws bars, ``geom_point`` draws points.
```

# Code re-use

Writing good software is hard

Existing, well-used and maintained software contains fewer bugs

Common problems are already solved

Avoid re-implementation — produce interfaces

Focus on new things

→ Lots of package interdependencies (>1000 packages, 100s developers)

# Don't reinvent the wheel

Shared code base, maintained by core team

Bioconductor already has code to:

- Read common file formats
- Represent common data types e.g. Genomic Ranges, Summarized Experiments
- Load genomes and annotation
- etc.

Let users become  
developers

# What are the benefits from using the Bioconductor development environment?

Standardised and powerful data structures for representing datasets incl. metadata

Many tools for data I/O and preprocessing. Access to databases of primary data and annotation (ExperimentHub)

Support for writing good documentation

Support for supporting your users



# What are the benefits from using the Bioconductor development environment?

Free code review

Package system

Daily checks - continuous integration

Version control system

Release & devel branches

Six monthly release cycle

Stable version for most users,  
but easy to make new features  
public





# Why R?

- high-level, interpreted programming language
- rapid prototyping, creativity, flexibility and reproducibility
- scientific and statistical computing capabilities
- publication quality graphics system ('grammar of graphics')
- convenient data I/O & wrangling
- mature package management system
- inter-language interfaces (C, C++, Java, JavaScript)
- lots of momentum with recent language innovations (RStudio, tidyverse, Jupyter, commercial adaptations, ...)

LISP/Scheme inside