Introduction to R Syntax[1]

Introduction to R Syntax as a key to language expression

Pablo E Adames
MEng, MSc, PEng

pablo.adames@alumni.ucalgary.ca

January 6nd and 7rd, 2025

[1] Last generated: January 6, 2025; 3:22pm

Outline

What is the context for this workshop?

We engage in the following...

- Science/engineering
- Collecting data in experiments to prove hypothesis
- Processing the data, concluding, communicating
- Using computer tools to do all of these
- Our tools are hardware and software
- Software lies on a spectrum:

What is the context for this workshop?

We engage in the following...

- Science/engineering
- Collecting data in experiments to prove hypothesis
- Processing the data, concluding, communicating
- Using computer tools to do all of these

- Our tools are hardware and software
- Software lies on a spectrum:

What is the context for this workshop?

We engage in the following...

- Science/engineering
- Collecting data in experiments to prove hypothesis
- Processing the data, concluding, communicating
- Using computer tools to do all of these

- Our tools are hardware and software
- Software lies on a spectrum:

The tools for data processing

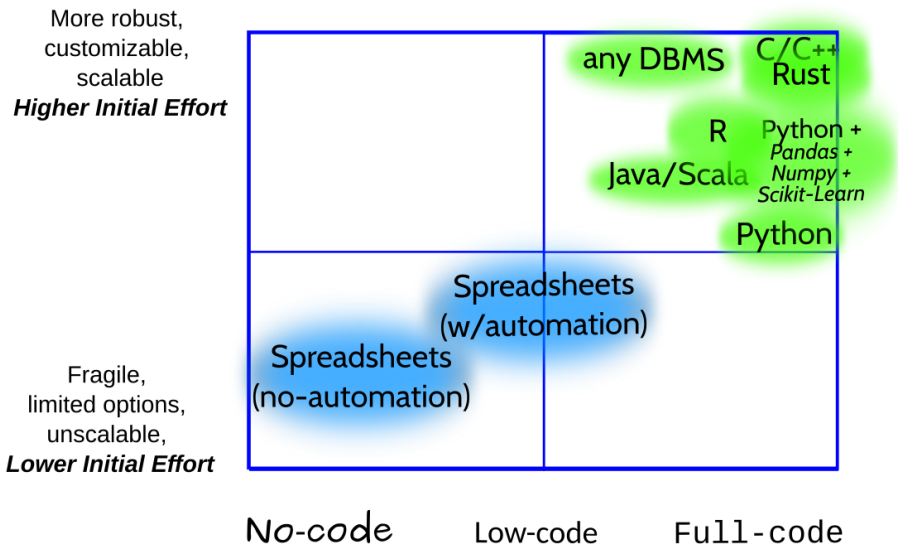
- Single-tier: own laptop/desktop/tablet and spreadsheets
- Multi-tier: remote server/computer clusters, database server, local programming editor and the terminal.

The tools for data processing

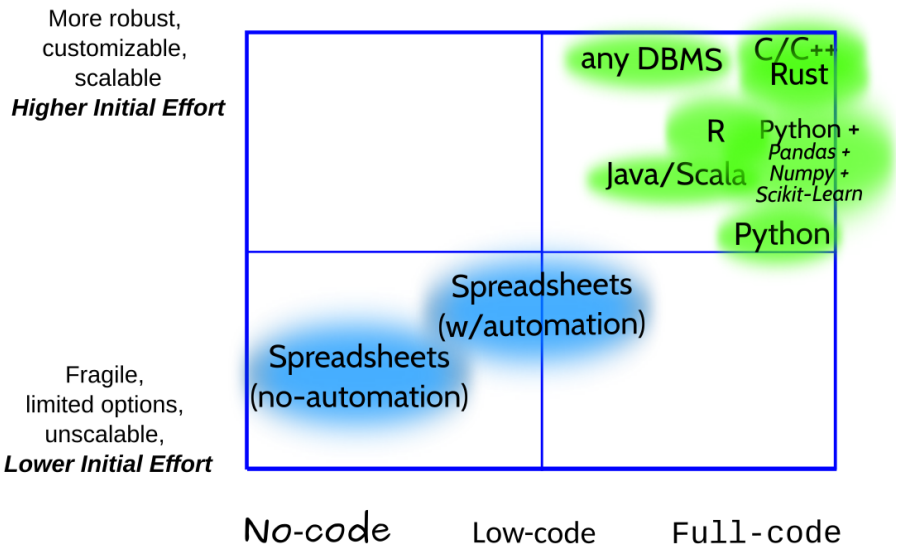
- Single-tier: own laptop/desktop/tablet and spreadsheets
- Multi-tier: remote server/computer clusters, database server, local programming editor and the terminal.

✔ Depends on data set size/frequency
✔ Team size and research budget
✔ The effort is proportional to task

The effort Vs Code graph



The effort Vs Code graph



- ✔ No silver bullet to do it all!
- ✔ Programming language(s) are essential
- ✔ The effort is proportional to task

Written language is a technology

- Syntax is a key to unlock its power
- It's about how words get arranged to make sense to the receiver
- With the advent of Large Language Models (LLMs)
- Some say "computer programming is dead"

Written language is a technology

- Syntax is a key to unlock its power
- It's about how words get arranged to make sense to the receiver
- With the advent of Large Language Models (LLMs)
- Some say "computer programming is dead"

Written language is a technology

- Syntax is a key to unlock its power
- It's about how words get arranged to make sense to the receiver
- With the advent of Large Language Models (LLMs)
- Some say "computer programming is dead"

This does not apply to you if...

- ✔ You are (becoming) a career researcher
- ✔ You want to have precise control over the processing and visualization of your data

General purpose Vs Data Processing Syntax

R is data oriented

Statistical language, stats is about data and math. Its is natively vectorized and capable of graphics (no need of modules).

Systems-level languages

C/C++ and more recently Rust. They are used when real-time/high throughput is required

General purpose languages

Java/Scala are JVM languages used heavily for distributed data processing (Spark framework). Python also falls into this category.

General purpose Vs Data Processing Syntax

R is data oriented

Statistical language, stats is about data and math. Its is natively vectorized and capable of graphics (no need of modules).

Systems-level languages

C/C++ and more recently Rust. They are used when real-time/high throughput is required

General purpose languages

Java/Scala are JVM languages used heavily for distributed data processing (Spark framework). Python also falls into this category.

General purpose Vs Data Processing Syntax

R is data oriented

Statistical language, stats is about data and math. Its is natively vectorized and capable of graphics (no need of modules).

Systems-level languages

C/C++ and more recently Rust. They are used when real-time/high throughput is required

General purpose languages

Java/Scala are JVM languages used heavily for distributed data processing (Spark framework). Python also falls into this category.

Outline

Many R syntaxes

- An R package can define its own syntax
- Three main-stream syntaxes in use
- They are equally valid ways of expression

Excellent R syntax “unofficial” cheat sheet:

https://github.com/rstudio/cheatsheets/blob/main/syntax.pdf

Three main stream

1. The **dollar sign syntax,** sometimes called **base R syntax**, expected by most `base` R functions. It is characterized by the use of `dataset$variablename`, and is also associated with square bracket subsetting, as in `dataset[1,2]`. Almost all R functions will accept things passed to them in dollar sign syntax.

- Dollar syntax or basic R syntax
- Formula syntax, $y \sim x$
- Tidyverse syntax, part of the dplyr grammar

Three main stream

- Dollar syntax or basic R syntax
- Formula syntax, $y \sim x$
- Tidyverse syntax, part of the dplyr grammar

2. The **formula syntax**, used by modeling functions like `lm()`, `lattice` graphics, and `mosaic` summary statistics. It uses the tilde (~) to connect a response variable and one (or many) predictors. Many base R functions will accept formula syntax.

Three main stream

- Dollar syntax or basic R syntax
- Formula syntax, $y \sim x$
- Tidyverse syntax, part of the dplyr grammar

3. The **tidyverse syntax** used by `dplyr`, `tidyr`, and more. These functions expect data to be the first argument, which allows them to work with the "pipe" (`%>%`) from the `magrittr` package. Typically, `ggplot2` is thought of as part of the tidyverse, although it has its own flavor of the syntax using plus signs (+) to string pieces together. `ggplot2` author Hadley Wickham has said the package would have had different syntax if he had written it after learning about the pipe.

Three main stream

1. The **dollar sign syntax,** sometimes called **base R syntax**, expected by most `base` R functions. It is characterized by the use of `dataset$variablename`, and is also associated with square bracket subsetting, as in `dataset[1,2]`. Almost all R functions will accept things passed to them in dollar sign syntax.

- Dollar syntax or basic R syntax
- Formula syntax, $y \sim x$
- Tidyverse syntax, part of the dplyr grammar

We will focus on the base R syntax

Outline

Data Representation

Native data structures in R according to the data type they can store and the number of dimensions they use (Wickham 2015, 13).

Dimensions	Homogeneous	Heterogeneous
One	Vector	List
Two	Matrix	Data frame
Three or more	Array	

There are no scalars and everything is an object.

Recycling vectors

Element-wise multiplication

$$\begin{bmatrix} 59 & 70 & -9 & 38 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 59 & 70 & -9 & 0 \end{bmatrix}$$

Recycling vectors

Recycling a smaller vector

$$\begin{bmatrix} 59 & 70 & -9 & 38 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 59 & 0 & -9 & 0 \end{bmatrix}$$

Recycling vectors

Recycling explained...

$$\begin{bmatrix} 59 & 70 & -9 & 38 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 59 & 0 & -9 & 0 \end{bmatrix}$$

Recycling vectors

Recycling a smaller vector

$$\begin{bmatrix} 59 & 70 & -9 & 38 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 59 & 0 & -9 & 0 \end{bmatrix}$$

R Code | Start Over | Run Code

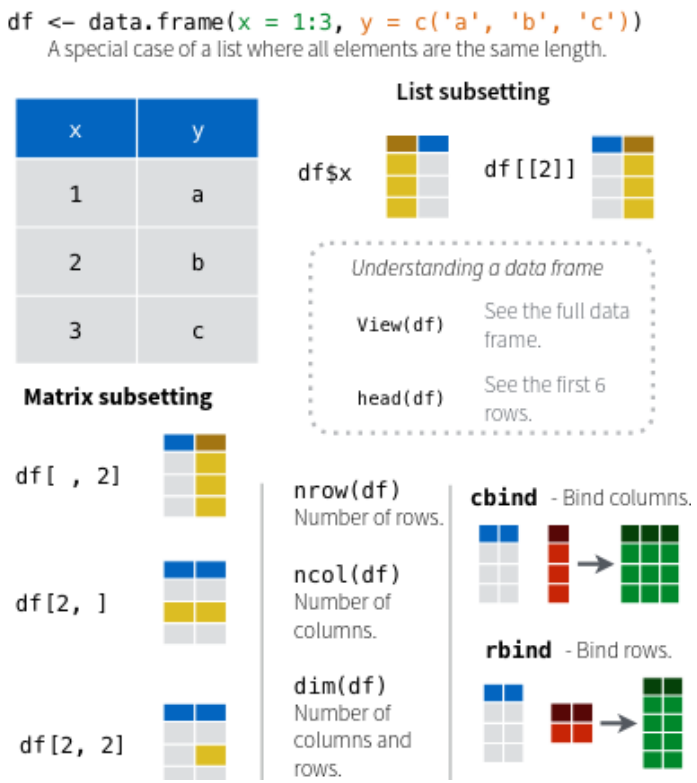
```
1 c(59, 70, -9, 38) * c(1,1,1,0)
2 c(59, 70, -9, 38) * c(1,0)
```

```
[1] 59 70 -9  0
```

```
[1] 59  0 -9  0
```

Recycling is a feature not a bug!

Subsetting



This is native to the language, no need of libraries.

Outline

Roadmap

- Command line usage (little/no code reuse)
- Writing/executing scripts
- Reproducible data analysis (knitr)
- Write your user functions
- Write your tests
- Package authoring
- Visualization tools/deployments
- Data processing pipeline (Data Eng)
- Online presentations/websites

Order does not matter. Journey length as required by your career/profession

R technologies

- Computational statistics language
- Tools for reproducible science
- Tools for publishing (papers, books, blogs, etc)
- Tools for sharing exploratory analysis (Shinny/HTML docs)
- Tools for reporting results (dashboards, websites)
- Tools for data engineering (cloud, CI/CD)

Evolving and modern tooling ecosystem: Posit, RPubs, Github

Batteries included: low barrier to entry

Thank you

That was my last slide. . .

Now let's get to the workshop!