7a. Overview and Goals

Martin Alfaro
PhD in Economics

The first part of the website has laid the groundwork for working with Julia. This demanded introducing fundamental data types, such as scalars, vectors, and tuples. Alongside these, we've covered essential programming constructs, including functions, conditionals, and for-loops. While these concepts may vary in syntax and usage across different programming languages, their underlying principles remain universal.

In the second part of the website, we'll shift our attention to one of Julia's most distinctive strengths: **high-performance computing**. When paired with its intuitive syntax and interactive nature, this feature makes Julia an ideal choice for scientific applications.

The domain of high-performance computing is vast and complex. Moreover, each subject has idiosyncratic features that make certain optimizations more or less relevant. Given this breadth, I've made deliberate choices about what to include and exclude. The challenge lay in striking the right balance between providing sufficient background knowledge for explaining a technique, while avoiding unnecessary specificity.

Considering this inherent trade-off, I've chosen the subjects with the goal of equipping readers with practical knowledge for optimizing code, without overwhelming them with excessive detail. In particular, the primary focus will be on what I consider to be the essentials for performance in Julia: **type stability** and **reductions in memory allocations**. The former in particular constitutes a prerequisite for achieving high performance in Julia, making it necessary for any further optimization.

The discussion of high performance in Julia will lead us to consider its type system. Nonetheless, some valuable concepts related to it have been left out. In particular, the concept of `struct`, which allows users to create their own custom objects, won't be covered. There are two reasons for this omission. First, while important for project development, the subject can be bypassed when analyzing high performance, without compromising its understanding. Second, the section included on types is already long enough—adding more subjects could divert the reader's attention away from the primary focus, which is learning high-performance techniques.