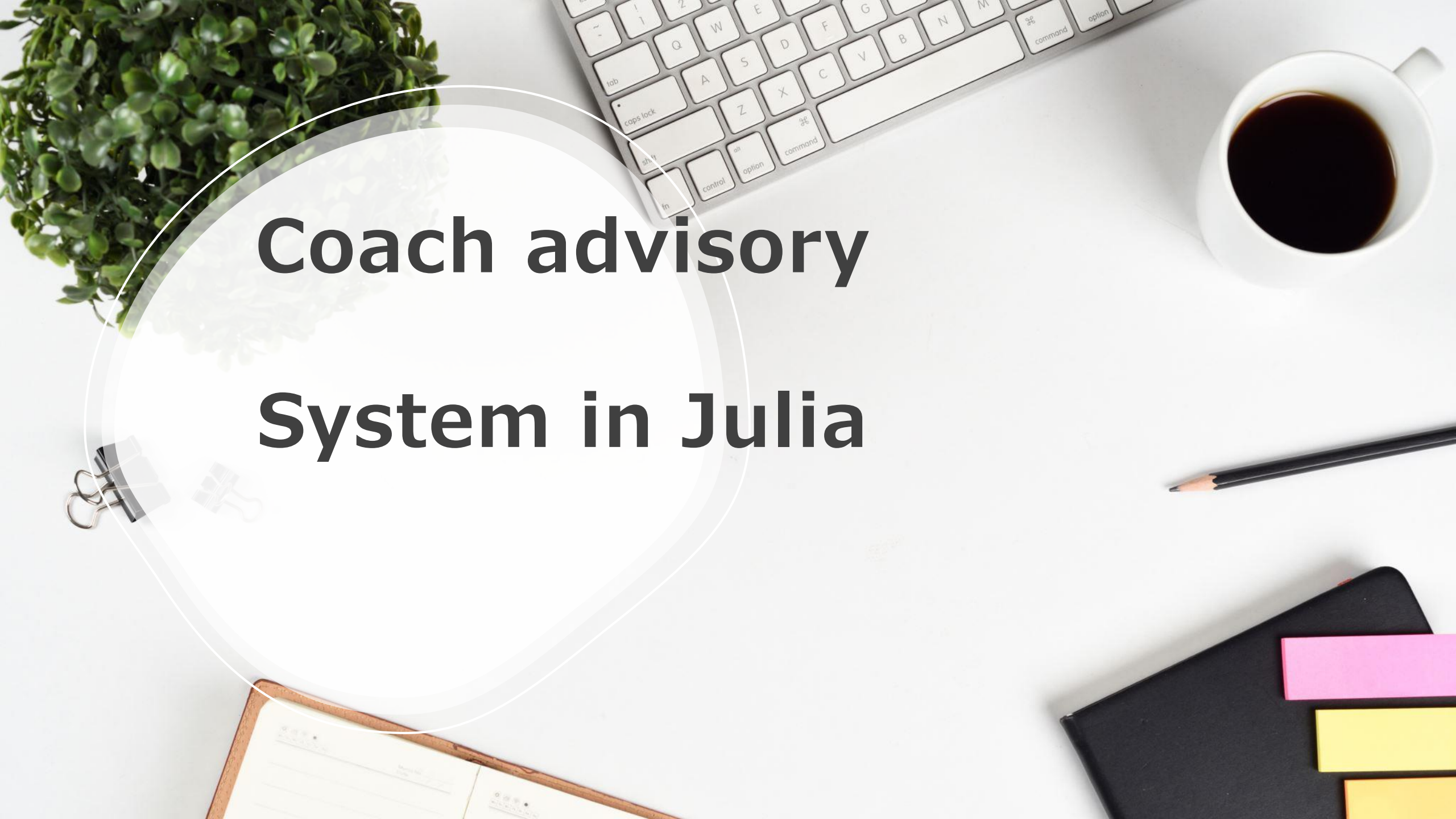
Coach advisory System in Julia



About Julia

Julia is a high-level, high-performance, dynamic programming language. While it is a general-purpose language and can be used to write any application, many of its features are well suited for numerical analysis and computational science.

It is made by Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman, who set out to create a free language that was both high-level and fast.

What is the problem ?

The problem is that you have dataset of matches played and you want the coaches to determine if their team are going to win the next match –based on the statistics of the last seasons-or not.

So we decided that we will use a decision tree to solve this kind of problem.

First things first What is a decision tree?

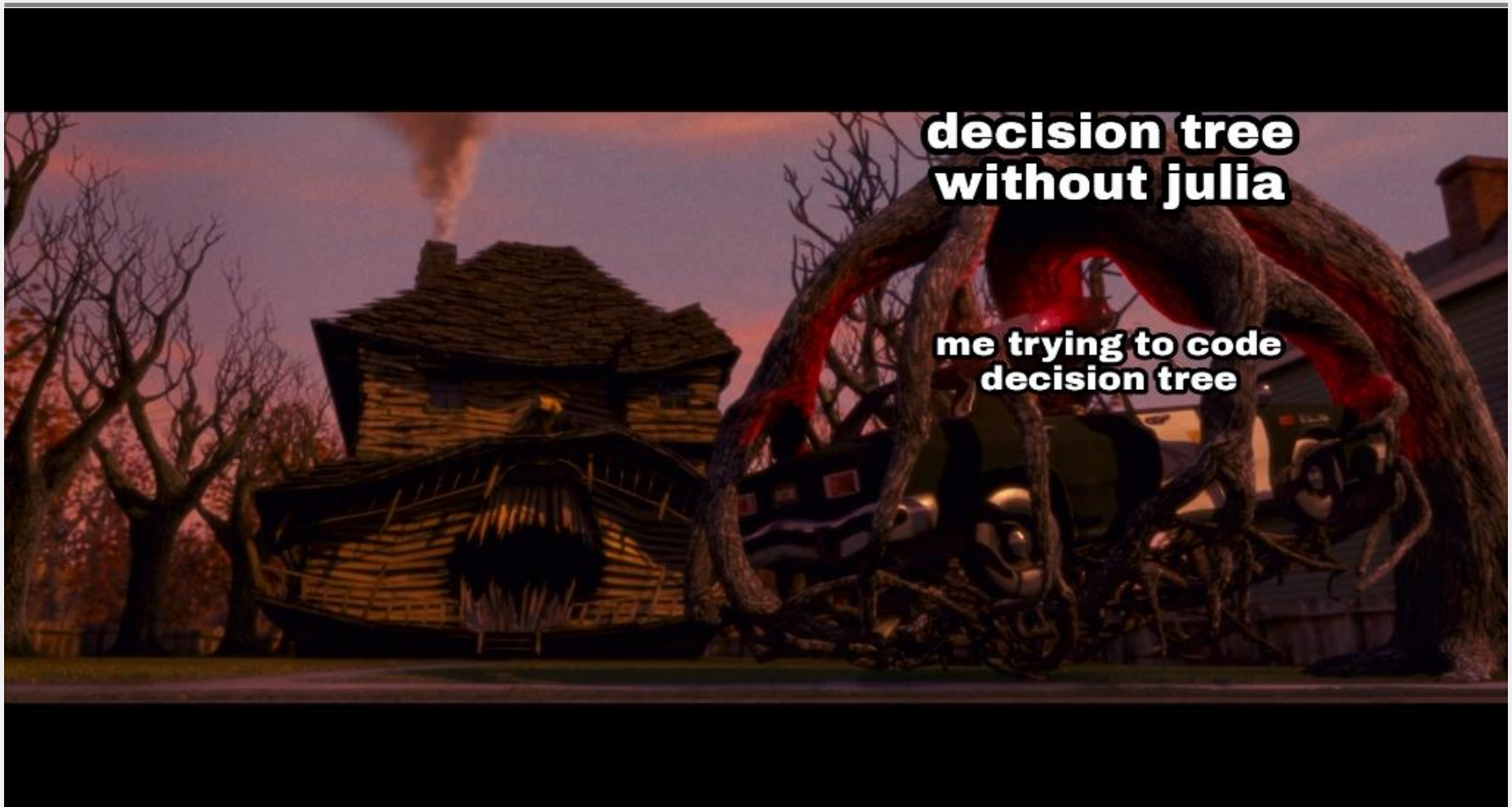
A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Why did you use Julia when you have other languages?

Ok let me answer you in a funny way and I will explain it in moments...

Trees in real world Vs trees in CS

Decision Tree without Julia be like:



Decision Trees in Julia be like

Let’s begin getting our hands dirty:

First to implement the problem we have. You will use a huge dataset. When it comes to dealing with big datasets Julia has a game changer called DataFrames. Using DataFrames you can read, write and traverse data and it will be visualized to see how are you doing in the process of cleaning or even using the dataset. So this a great advantage that we used in Julia.

Let’s begin getting our hands dirty(continued):

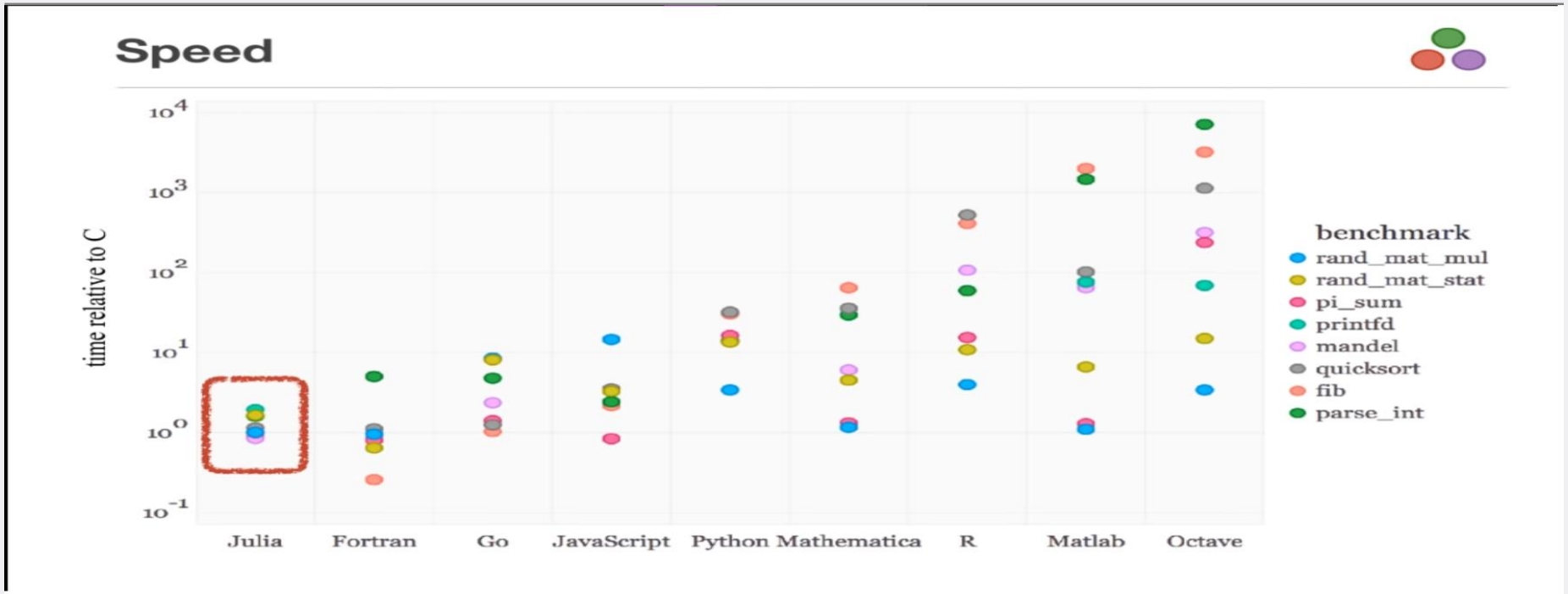
Another advantage that we used in Julia while programming the decision tree is **DecisionTree().** It is great advantage that Julia has as you can build decision trees easily, quickly and technically good using this built-in lib.

So the main advantage out of the last two examples is that Julia has powerful built-in libs and its community are contributing to it continuously.

Let’s begin getting our hands dirty(continued):

Another advantage is that Julia is relatively fast when compared with other languages.

Let’s begin getting our hands dirty(continued):



Let’s begin getting our hands dirty(continued):

Another advantage:

Julia takes other programming languages packages explained in the next meme😂😂😂

Let’s begin getting our hands dirty(continued):