

Laboratório de Biologia Computacional e Molecular

Centro de Biotecnologia da UFRGS
Universidade Federal do Rio Grande do Sul



R para Ciências da Vida (BCM13065) Aula 9

PPGBCM - UFRGS

Diego Bonatto
2024/2

Por que integrar R com Python?

The image is a screenshot of the Python.org website. At the top, there is a dark blue navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a lighter blue header section containing the Python logo, a yellow 'Donate' button, a search bar with a magnifying glass icon and a 'GO' button, and a 'Socialize' button. A secondary navigation bar below the header contains links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area is divided into two columns. The left column features a code editor with a yellow prompt icon and a dark background, displaying a Python 3 script for a Fibonacci series. The right column has a yellow header 'Functions Defined' and text explaining that the core of extensible programming is defining functions, with a link to 'More about defining functions in Python 3'. At the bottom of the right column are five numbered buttons (1-5).

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

[re:Invent](#)[Produtos](#)[Soluções](#)[Definição de preço](#)[Documentação](#)[Aprenda](#)[Rede de parceiros](#)[AWS Marketplace](#)[Capacitação de clientes](#)[Eventos](#)[Explore mais](#)[Sobre a AWS](#)[Fale conosco](#)[Suporte](#)[Português](#)[Minha conta](#)[Fazer login](#)[Crie uma conta da AWS](#)

O que é Python?

Quais são os benefícios do Python?

Como usar o Python?

Qual é a história do Python?

Quais são os principais recursos do Python?

O que são bibliotecas Python?

O que são frameworks Python?

O que são IDEs Python?

O que são SDKs do Python?

O que é o Boto3 em Python?

O que é o AWS PyCharm?

O que é Python?

O Python é uma linguagem de programação amplamente usada em aplicações da Web, desenvolvimento de software, ciência de dados e machine learning (ML). Os desenvolvedores usam o Python porque é eficiente e fácil de aprender e pode ser executada em muitas plataformas diferentes. O software Python pode ser baixado gratuitamente, integra-se bem a todos os tipos de sistema e agiliza o desenvolvimento.

Quais são os benefícios do Python?

Os benefícios do Python incluem:

- Os desenvolvedores podem ler e entender facilmente um programa Python, porque tem uma sintaxe básica semelhante à do inglês.
- O Python aumenta a produtividade dos desenvolvedores, porque eles podem escrever um programa Python usando menos linhas de código em comparação a muitas outras linguagens.
- O Python tem uma grande biblioteca-padrão que contém códigos reutilizáveis para quase todas as tarefas. Como resultado, os desenvolvedores não precisam escrever códigos do zero.
- Os desenvolvedores podem usar o Python facilmente com outras linguagens de programação populares, como Java, C e C++.
- A comunidade ativa do Python inclui milhões de desenvolvedores de suporte em todo o mundo. Se você tiver algum problema, poderá obter suporte rápido da comunidade.
- Muitos recursos úteis estão disponíveis na Internet se você quiser aprender Python. Por exemplo, você pode encontrar facilmente vídeos, tutoriais, documentação e guias do desenvolvedor.
- É possível fazer a portabilidade do Python em diferentes sistemas operacionais de computação, como Windows, macOS, Linux e Unix.



Python e bioinformática

- **Python é uma linguagem de programação popular para bioinformática porque é fácil de aprender, possui uma grande comunidade e oferece muitas ferramentas e pacotes para análise e visualização de dados.**
- **Onde o Python é usado em bioinformática?**
 - Análise do genoma: Python pode alinhar sequências de DNA e proteínas, identificar variações genéticas e realizar análises de expressão gênica.
 - Análise e visualização da estrutura de proteínas: Python pode analisar e visualizar estruturas de proteínas.
 - Aprendizado de máquina: Python pode classificar genes e prever estruturas de proteínas.
 - Visualização de dados: Python oferece pacotes como Matplotlib e Seaborn para visualização de dados biológicos.

Por que integrar R com Python?

- **Vantagens da Integração**

- **Complementaridade:** R é excelentemente conhecido por suas capacidades estatísticas e visualização de dados, enquanto Python se destaca em áreas como machine learning, processamento de texto e desenvolvimento web. Ao combinar ambas, você pode construir pipelines de dados mais completos e eficientes.
- **Flexibilidade:** A integração permite que você escolha a ferramenta mais adequada para cada tarefa. Por exemplo, você pode usar R para realizar análises exploratórias e criar visualizações, enquanto utiliza Python para construir modelos de machine learning mais complexos.
- **Comunidade e Ecossistema:** Ambas as linguagens possuem comunidades robustas e ecossistemas de pacotes em constante crescimento. Isso significa que você terá acesso a uma ampla variedade de recursos e soluções para problemas comuns.
- **Produtividade:** A integração pode acelerar o desenvolvimento de projetos, permitindo que você aproveite as bibliotecas e ferramentas de ambas as linguagens.
- **Compartilhamento de Dados:** A capacidade de compartilhar dados entre R e Python facilita a colaboração entre equipes e a reutilização de código.

Por que integrar R com Python?

- **Desvantagens**

- **Curva de Aprendizado:** Aprender a trabalhar com duas linguagens exige um investimento de tempo e esforço.
- **Complexidade:** A integração pode adicionar complexidade ao seu projeto, especialmente se você precisar gerenciar diferentes ambientes e dependências.
- **Overhead:** A comunicação entre R e Python pode introduzir alguma sobrecarga, impactando o desempenho em alguns casos.

Ferramentas de integração

- **RPy2:** Um pacote R que permite chamar funções Python a partir de R.
- **Reticulate:** Um pacote R mais recente, que oferece uma interface mais moderna e flexível para a integração.
- **Python com rpy2:** Utilizar o pacote rpy2 em Python para chamar funções R.
- **Plataformas de Big Data:** Ferramentas como Spark e Hadoop permitem executar código tanto em R quanto em Python, facilitando a integração em grandes conjuntos de dados.



rpy2 - R in Python



rpy2 is an interface to **R** running embedded in a **Python** process.

pypi v3.5.17 **on Pypi**

Questions and issues

Consider having a look at the documentation. Otherwise questions should preferably be asked on the rpy mailing-list on SourceForge, or on **StackOverflow**. Bugs, or wishes, can also go to the **Github page** for the project.

rpy2 in the wild

Source code

Repository

The source code is also in a public repository on **Github**.

License

The same license as R: GPLv2 or greater.



R Interface to Python

The **reticulate** package provides a comprehensive set of tools for interoperability between Python and R. The package includes facilities for:

- Calling Python from R in a variety of ways including R Markdown, sourcing Python scripts, importing Python modules, and using Python interactively within an R session.
- Translation between R and Python objects (for example, between R and Pandas data frames, or between R matrices and NumPy arrays).
- Flexible binding to different versions of Python including virtual environments and Conda environments.



Reticulate embeds a Python session within your R session, enabling seamless, high-performance interoperability. If you are an R developer that uses Python for some of your work or a member of data science team that uses both languages, reticulate can dramatically streamline your workflow!

Getting started

Installation

Install the **reticulate** package from CRAN as follows:

```
install.packages("reticulate")
```

Python version

By default, reticulate uses an isolated python virtual environment named "r-reticulate".

The `use_python()` function enables you to specify an alternate python, for example:

```
library(reticulate)
use_python("/usr/local/bin/python")
```

The `use_virtualenv()` and `use_condaenv()` functions enable you to specify versions of Python in virtual or Conda environments, for example:

Links

[View on CRAN](#)

[Browse source code](#)

[Report a bug](#)

License

Apache License 2.0

Citation

[Citing reticulate](#)

Developers

Tomasz Kalinowski
Contributor, maintainer

Kevin Ushey
Author

JJ Allaire
Author

RStudio
Copyright holder, funder

Yuan Tang
Author, copyright holder 

[More about authors...](#)

Quando integrar?

- **Projetos de grande escala:** Quando você precisa lidar com grandes volumes de dados e uma variedade de tarefas analíticas.
- **Equipes multidisciplinares:** Se sua equipe inclui especialistas em ambas as linguagens, a integração pode facilitar a colaboração.
- **Desenvolvimento de aplicações:** Quando você precisa criar aplicações que combinam as capacidades de ambas as linguagens.

[Home](#)[About](#)[Get](#)[LaTeX3](#)[Publications](#)[Help](#)[News](#)

LaTeX – A document preparation system

LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents. LaTeX is available as [free software](#).

You don't have to pay for using LaTeX, i.e., there are no license fees, etc. But you are, of course, invited to support the maintenance and development efforts through a [donation to the TeX Users Group \(choose LaTeX Project contribution\)](#) if you are satisfied with LaTeX.

You can also sponsor the work of LaTeX team members through the [GitHub sponsor program](#) at the moment for [Frank](#), [David](#) and [Joseph](#). Your contribution goes without any reductions by GitHub to the developers in support of the

Recent News

27 November, 2024

[Better multicolumn marks: Pre-release of LaTeX 2025-06-01 is available for testing](#)

1 November, 2024

[LaTeX 2024-11-01 released and distributed](#)

8 September, 2024

[TUG Conference 2024 in Prague: talks and accessibility workshop](#)

8 July, 2024

[How to make accessible PDF](#)

13 June, 2024

[LaTeX 2024-06-01 PL1 released and distributed](#)

24 April, 2024

[Development releases for the L3 Programming Layer](#)

27 March, 2024

[ISO PDF/UA-2 standard released - Examples](#)

An introduction to LaTeX

LaTeX, which is pronounced «Lah-tech» or «Lay-tech» (to rhyme with «blech» or «Bertolt Brecht»), is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing.

LaTeX is not a word processor! Instead, LaTeX encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content. For example, consider this document:

```
Cartesian closed categories and the price of eggs
Jane Doe
September 1994

Hello world!
```

To produce this in most typesetting or word-processing systems, the author would have to decide what layout to use, so would select (say) 18pt Times Roman for the title, 12pt Times Italic for the name, and so on. This has two results: authors wasting their time with designs; and a lot of badly designed documents!

LaTeX is based on the idea that it is better to leave document design to document designers, and to let authors get on with writing documents. So, in LaTeX you would input this document as:

```
\documentclass{article}
\title{Cartesian closed categories and the price of eggs}
\author{Jane Doe}
\date{September 1994}
\begin{document}
  \maketitle
  Hello world!
```

Por que integrar LaTeX com R?

- **Vantagens da Integração**

- **Relatórios Reproduzíveis:** Ao integrar R e LaTeX, você pode criar documentos que são completamente reproduzíveis. Isso significa que qualquer pessoa com acesso ao código e aos dados pode gerar o mesmo relatório, garantindo a transparência e a verificabilidade dos resultados.
- **Formatação de Alta Qualidade:** LaTeX é conhecido por sua capacidade de produzir documentos com uma formatação impecável. Ao combinar com R, você pode criar gráficos, tabelas e equações de alta qualidade diretamente no seu documento.
- **Automatização:** A integração permite automatizar a geração de relatórios, economizando tempo e reduzindo o risco de erros manuais. Você pode criar templates e gerar relatórios personalizados com base em diferentes conjuntos de dados.
- **Flexibilidade:** LaTeX oferece uma grande variedade de estilos e templates, permitindo que você adapte seus documentos às normas de diferentes publicações.
- **Integração de Código:** O código R pode ser incorporado diretamente no documento LaTeX, permitindo que você veja o código e os resultados lado a lado. Isso facilita a depuração e a compreensão do processo de análise.

Por que integrar LaTeX com R?

- **Desvantagens**

- Curva de Aprendizado: Aprender tanto R quanto LaTeX pode exigir um investimento de tempo considerável, especialmente para quem está começando.
- Complexidade: A configuração inicial e a criação de templates podem ser complexas, especialmente para projetos maiores.
- Dependências: A geração de documentos finais pode depender de várias ferramentas e pacotes, o que pode tornar o processo mais frágil e sujeito a erros.

Por que integrar LaTeX com R?

- **Formas de Integração**

- RMarkdown: É a forma mais popular de integrar R e LaTeX. RMarkdown permite que você escreva documentos em um formato semelhante ao Markdown, mas com a capacidade de executar código R e gerar saída em diversos formatos, incluindo PDF (via LaTeX), HTML e Word.
- Sweave: Um dos primeiros pacotes a permitir a integração de R e LaTeX, Sweave é mais antigo que RMarkdown, mas ainda é utilizado por muitos usuários.
- Pacotes de Conversão: Existem pacotes como o xtable que permitem converter objetos do R em tabelas LaTeX.

R Markdown

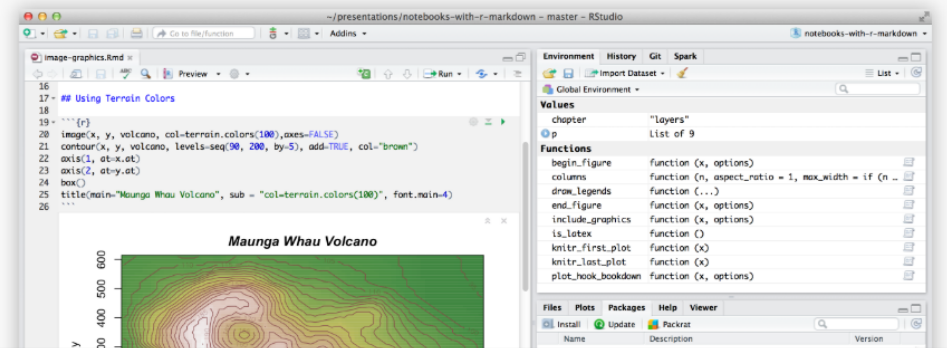
from  Studio

[Get Started](#) [Gallery](#) [Formats](#) [Articles](#) [Book](#) [References](#) 

Analyze. Share. Reproduce.

Your data tells a story. Tell it with R Markdown.
Turn your analyses into high quality documents,
reports, presentations and dashboards.

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.



rmarkdown

R-CMD-check failing CRAN 2.29 codecov 64%

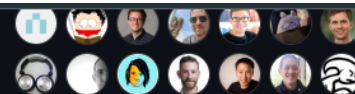
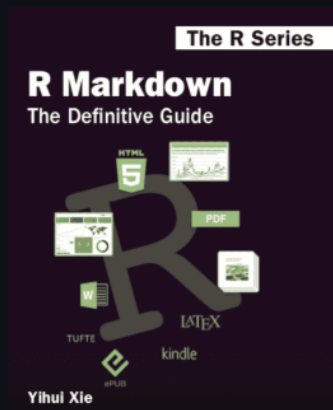


The **rmarkdown** package helps you create dynamic analysis documents that combine code, rendered output (such as figures), and prose. You bring your data, code, and ideas, and R Markdown renders your content into a polished document that can be used to:

- Do data science interactively within the RStudio IDE,
- Reproduce your analyses,
- Collaborate and share code with others, and
- Communicate your results with others.

R Markdown documents can be rendered to many output formats including HTML documents, PDFs, Word files, slideshows, and more, allowing you to focus on the content while R Markdown takes care of your presentation.

Books



+ 115 contributors

Languages

