

Introdução à Bioestatística para estudos Desenhos Experimentais  
em Biologia usando Python (2026)

ESIB  
CENTD – Instituto Butantan

PhD Flavio Lichtenstein

Bioinformatics, Systems Biology, and Biostatistics

Instituto Butantan – CENTD - Bioinformática

Janeiro 2026



# Qual o foco do curso?



chat-gpt: desenhe um jovem medieval de cútis branca, bochechas róseas, olhos marrons olhando vesgamente a uma flechada numa maçã vermelha posicionada acima de sua cabeça.

# Qual o foco do curso?



O foco principal:

1. Entender o que é ciência e um experimento científico: como definir variáveis, entender os efeitos de uma perturbação e onde a estatística entra nesta história.
2. Entender bioestatística

Focos secundários:

3. Iniciar ou aprimorar a estatística com a linguagem Python.

# Preciso entender o que é ciência?

Sim, “o que é ciência” é um dos focos principais do curso aliado a conceitos estatísticos.

A primeira aula é 100% dedicada a conceitos de ciência e ao método científico.

Para tanto vamos “gastar” algum tempo revendo a história da ciência, em especial a história de Darwin e a Evolução e a história de Mendel e os conceitos de hereditariedade.

Vamos entender como nascem hipóteses e como são desenhados alguns experimentos (especialmente os de Mendel).

Por fim, explicaremos o método científico e suas consequências.

# Preciso entender bioestatística?

Preciso entender bioestatística? - como assim?

Sim, bioestatística consta no título do curso, logo é um dos objetivos. A partir do segundo dia elucidaremos os principais conceitos de bioestatística.

Mas, atenção. Você não precisa saber Python ou necessariamente aderir a esta linguagem de programação. Usaremos Python, por acharmos que é a melhor ferramenta de trabalho, mas você pode discordar e desejar utilizar R ou soluções como SPSS, SAS, BioStat, SigmaPlot ou outras.

## Outras soluções:

- R
  - Projeto R: <https://www.r-project.org/>
  - Conceitos: <https://www.r-project.org/about.html>
  - Tutorial: <https://www.w3schools.com/r/>
  - RStudio: <https://posit.co/download/rstudio-desktop/>
  - Bioconductor: <https://www.bioconductor.org/>
- SAS: <https://www.sas.com/>
- SPSS: <https://www.ibm.com/br-pt/products/spss-statistics>
- BioStat: <https://www.analystsoft.com/br/products/biostat/>
- SigmaPlot: <https://grafiti.com/sigmaplot-detail/>

# Preciso entender Python?

O foco do curso é iniciar o estudante em conceitos de bioestatística e estudos experimentais. Para isto utilizaremos Python, mas você não precisa entender Python, nem ser iniciado, nem ser um especialista.

Se não desejar aprender Python, você poderá:

- somente acompanhar o curso.
- reproduzir os códigos oferecidos.

Para os que não estão interessados em Python, a priori, todos os códigos e resposta serão fornecidos. Somente desejamos que acompanhem atentamente aos exemplos e conceitos, estudem e sejam críticos. Não haverá desafios em Python!

Para aqueles interessados em Python recomendamos reproduzir e fazer os exercícios propostos.

Não checaremos as respostas (que serão fornecidas) nem haverá avaliações. Importante: os conceitos expostos são gerais e podem ser reproduzidos em outras linguagens como R ou pacotes estatísticos.

# Material do Curso

Para acessar o curso, você tem que ter um email do Google (gmail).

O motivo é que utilizaremos a plataforma “Classroom” do Google para ensinar e o “Google Colab” para rodar o código Python.

Uma vez convidado no Classroom, você pode acessá-lo em:

<https://classroom.google.com/w/NzkzMTMwNTgwMTY2/t/all>

Os códigos e dados estarão no GitHub e poderão ser acessados facilmente (ensinaremos estes conceitos). Os códigos são livres e abertos. Você pode baixar todo o conteúdo das aulas, copiar, mostrar e enviar a amigos.

Os códigos serão divididos em tópicos (aulas), além de um diretório de dados (data), e um diretório com as respostas dos exercícios. Aos alunos que pretendem aprender e se exercitar em Python, recomendamos que façam os exercícios, e só depois de resolvido olhem as respostas. Os outros, podem rodar as respostas e acompanhar os resultados.

# Foco do Curso

Nossa meta é cobrir conceitos básicos de Python sem avançar muito nesta linguagem.

O curso de Bioestatística tem como meta abranger alguns problemas de análise de dados de um laboratório de biologia, medicina ou relacionado a temas de saúde.

Utilizaremos Google-Colab (<https://colab.research.google.com/>) ← clique aqui, sendo que em muitos casos não entraremos em profundidade em conceitos específicos ou avançados, mas somente como utilizá-los na prática. Este ambiente, na nuvem, permite acessar determinada versão do Python (como a 3.11); e também pode rodar códigos de outras linguagens como R, mas este não será nosso foco.

Este mesmo ambiente pode ser utilizado localmente (no seu computador), e ofereceremos uma aula extra com `uv`, para os interessados. Este é o administrador de pacotes mais recente e rápido, mas você pode utilizar Anaconda ou Virtual Environment.

UV - <https://docs.astral.sh/uv/>

Anaconda <https://www.anaconda.com>

Virtual Environment - venv - <https://docs.python.org/3/library/venv.html>

## Agenda do curso

# Agenda

Agenda distribuída em 8 dias:

1. Ciência, método científico, experimentos controlados
2. Estatística inicial e história
  - a. Introdução ao Python(para todos)
  - b. Google-Colab (para todos)
  - c. Aula extra desenvolvimento de Python local (interessados em Python)
3. Estatística:
  - a. Variáveis randômicas
  - b. Frequência, simetria, quartis, outliers, boxplots ...
4. Distribuições:
  - a. Distribuições Discretas
  - b. Distribuições Contínuas + introdução a Pandas
5. Teste de hipóteses
6. Testes de hipóteses paramétricos
7. Testes de hipóteses não paramétricos
8. Regressões e causalidade

Python: se você não tem interesse,  
a apresentação se encerra por aqui

# O que é Python?

Linguagem de computador com as seguintes características:

- Ótima portabilidade
- Fácil utilização
- Sem custos (gratuito) e código livre (com raras exceções)
- Utilizado por iniciantes até profissionais de alto nível
- Possui poderosa biblioteca
- Pode ser utilizado simultaneamente num mesmo computador sob diferentes ambientes:
  - Google Colab (com instalação de novas bibliotecas usando pip)
  - Anaconda, Virtual environment ou UV
  - Docker

# O que é Python?

Linguagem de computador com as seguintes características:

- Ótima portabilidade
- Fácil utilização
- Sem custos (gratuito) e código livre (com raras exceções)
- Utilizado por iniciantes até profissionais de alto nível
- Possui poderosa biblioteca
- Pode ser utilizado simultaneamente num mesmo computador sob diferentes ambientes:
  - Google-Colab (com instalação de novas bibliotecas usando pip)
  - Anaconda, Virtual environment ou UV
  - Docker

Pô! Explica melhor tudo isto!

# O que é Python? - portabilidade

O que é portabilidade? É a capacidade de rodar códigos fontes em diferentes computadores e sistemas operacionais. Ou seja, se você criou algo em no seu Laptop Mac, pode rodar o mesmo (com poucas diferenças) num Servidor Linux. Importante: a utilização Python em áreas científicas raramente utiliza Windows.

- Sistemas operacionais:
  - Linux, Windows, macOS ...
- Computadores:
  - PC, laptop
  - Servidor
  - Supercomputador

# O que é Python? - é fácil!

Python é uma linguagem de computador de fácil utilização:


- Linguagem interpretada
- Sintaxe simples
- Estruturada:
  - Texto
  - Funcional
  - Orientado a objetos
- O Python tem diversos editores gratuitos (vamos usar o Jupyter Notebook) - também recomendamos o Visual Studio Code (da Microsoft, livre e gratuito) - mas, não teremos tempo para ensiná-lo, neste curso.

<https://code.visualstudio.com/>

# O que é Python? - linguagem sem custos

## Sem custos

- Não há custos para utilizá-lo e ainda usualmente está sob código livre o que aumenta o aprendizado se desejar aprender um dado algoritmo “mais a fundo”
- É uma linguagem poderosa e de cooperação (veja, por exemplo, o PyMOL)  
<https://www.youtube.com/watch?v=h5wKppcyzOw>
- Alguns links básicos desta linguagem:
  - Python org: <https://www.python.org/>
  - Python doc: <https://docs.python.org/3/library/>
  - Wiki: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

 clique nas  
linhas em azul

# O que é Python? - todos podem usar Python

Pode ser utilizado por qualquer pessoa:

- Programadores iniciantes
- Usuários com baixo conhecimento em informática
- Usuários com bom conhecimento em informática
- Programadores experientes que utilizam OO (Orientação por Objetos)
- Cientistas na área de computação
- Programação de alta performance (HPC): veja o Santos Dumont

(LNCC) <https://www.youtube.com/channel/UCD5S2EfH6z0SKeO1YCNztzg/videos>

# O que é Python? - bibliotecas

As bibliotecas (ou pacotes) expandem as possibilidades do Python.  
Há bibliotecas como:

- Gráficas:

<https://matplotlib.org/3.1.1/gallery/index.html>

<https://seaborn.pydata.org/examples/index.html>

<https://plotly.com/python/>

- Numéricas, científicas e de aprendizado de máquina:

<https://numpy.org/>

<https://www.scipy.org/>

<https://scikit-learn.org/stable/>

# O que é Python? - ambientes de controle de bibliotecas

Em “situações gerais”, no Google Colab você não precisa se incomodar em gerir versões de diferentes bibliotecas, além do mais, quando você abre um Google Colab muitas das bibliotecas já estão prontas e instaladas.

Mas, o que é biblioteca? E o que é Gestão de Versões de Bibliotecas?

Vamos dar um passo atrás. Falamos em “situações gerais”, o que quer dizer isto? Para um programador iniciante, este vai abrir o ambiente do Google Colab (ambiente de desenvolvimento em Python), eventualmente instalar uma ou duas bibliotecas com o comando pip, e rodar seus programas.

Mas, para um desenvolvedor sênior, este usa diversas bibliotecas e algumas podem se tornar incompatíveis com as outras na evolução temporal. Ou seja, a medida que o tempo passa o Python e Sistemas Operacionais evoluem (melhoram e ficam mais poderosos). Pode ser que uma ou outra funcionalidade se torne defasada, ou seja, não é atualizada na nova versão e fique incompatível com outros pacotes mais modernos e atualizados. Logo, um sistema gestor de pacotes como Anaconda, Virtual Environment ou UV se torna necessário para garantir esta compatibilidade. Há ainda o problema de reprodutibilidade, que discutiremos nas próximas aulas.

# O que é Python? - ambientes de controle de bibliotecas

Em situações bem avançadas, normalmente em ambiente Linux, você será obrigado a controlar as versões de bibliotecas Linux, além das versões de bibliotecas Python. Neste caso poderá usar Anaconda, Virtual Environment, UV com ou sem Docker.

Docker é um administrador de ambientes virtuais em que você encapsula tudo o que precisa dentro de um módulo e ele monta um microambiente para sua aplicação. Porém, neste curso não falaremos de ambientes mais avançados. (<https://www.docker.com/>).

Três ótimos gerenciadores de versões Python são:

- Anaconda: <https://www.anaconda.com/>
- Virtual Environment: <https://docs.python.org/3/library/venv.html>
- UV: <https://docs.astral.sh/uv/>

normalmente utilizados em máquinas locais ou servidores para evitar conflitos de versões. Neste curso apresentaremos uma aula extra caso você queira usar o Python em uma máquina local e não na nuvem, usando o UV.

Seja bem vinda(o)

Em caso de dúvidas escreva no chat para o moderador

Algumas perguntas serão respondidas e depositadas no Q&A  
Possivelmente, devido ao grande número de alunos não conseguiremos  
responder a todas as perguntas, mas tentaremos.

PhD Flavio Lichtenstein

Bioinformatics, Systems Biology, and Biostatistics

Instituto Butantan – CENTD - Bioinformática

Janeiro 2026

