INSTITUTO BUTANTAN

Escola Superior Instituto Butantan

Introdução à bioestatística utilizando Python

Dia 02 - Instalando Python e Projeto com UV

PhD Flavio Lichtenstein

Bioinformatics, Systems Biology, and Biostatistics

Instituto Butantan – CENTD - Bioinformática

Janeiro 2026

centre of excellence in new target discovery

Porque mais um ambiente de gerenciamento de projetos e bibliotecas (pacotes Python)?

Porque mais um ambiente de gerenciamento de projetos e bibliotecas (pacotes Python)?

Porque temos que ter **reprodutibilidade computacional**

Controle de:

1. Versão Python
2. Versão de Código (junto a github)
3. Organização
    a. Bibliotecas/Pacotes
    b. Código fonte
    c. Dados

Além disto, UV é muito rápido

Porém, Anaconda gerencia pacotes R e Bioconductor, e UV somente Python.

Linux/Mac: instalando UV e Python (p.ex. versão 3.11)

Instalando uv:

```
$ wget -qO- https://astral.sh/uv/install.sh | sh
ou
$ pip install uv
```

Checando:

```
$ uv
```

Instalando Python em Linux:

```
$ uv python install 3.11
```

https://docs.astral.sh/uv/getting-started/installation/

Windows: instalando UV e Python (p.ex. versão 3.11)

Instalando uv:

```
PS> powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
ou
pip install uv
```

Checando:

```
$ uv
```

Instalando Python em Linux:

```
$ uv python install 3.11
```

https://docs.astral.sh/uv/getting-started/installation/

Define um diretório público e crie ‘biostat2026’

Diretório público em Linux:

~/biostat2026$

Diretório público em Windows:

PS> C:\Users\Public\biostat2026

A partir daqui nos referenciamos como biostat2026

Criando um projeto com uv

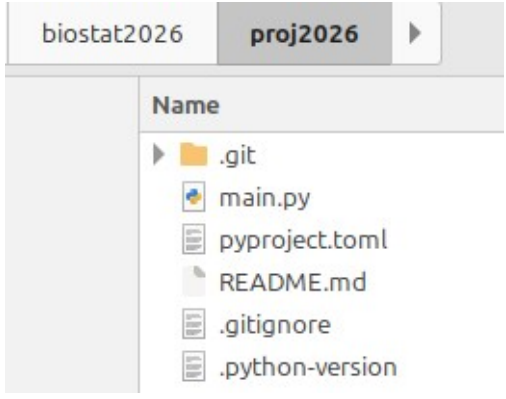
Criando ‘proj2026’

```
biostat2026$  uv init proj2026
# default
biostat2026$  uv init proj2026 --app
```

Projeto criado:

```
biostat2026$  tree proj2026/
```

```
proj2026/
├── main.py
├── pyproject.toml
└── README.md
```



Criando um projeto com uv

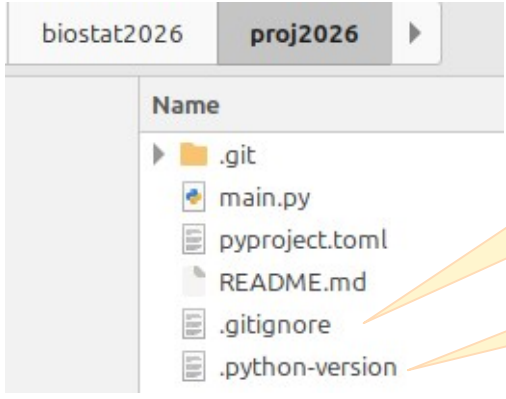
Criando ‘proj2026’

```
biostat2026$  uv init proj2026
# default
biostat2026$  uv init proj2026 --app
```

Projeto criado:

```
biostat2026$  tree proj2026/
```

```
proj2026/
├── main.py
├── pyproject.toml
└── README.md
```



controle de versão git

controle de versão python

Editando o projeto

Edite: pyproject.toml

```
[project]
name = "proj2026"
version = "0.1.0"
description = "digitar a descrição aqui"
readme = "README.md"
requires-python = ">=3.11"
dependencies = [ ]
```

digitar as depenências de pacotes Python, ou colocar de ‘requirements.txt’, ou rodar update de requirements

ou vai udicionand (add) os pacotes e uv controla isto para você

Editando o projeto

Edite: pyproject.toml

```
[project]
name = "proj2026"
version = "0.1.0"
description = "Aula de bioestatística (2016)"
readme = "README.md"
requires-python = ">=3.11"
dependencies = [
    "numpy",
    "pandas",
    "requests",
    "scikit-learn",
    "statsmodels",
    "matplotlib",
    "seaborn",
    "plotly",
    "kaleido",
    "jupyter",
    "biopython>=1.86",
    "biomart>=0.9.2",
    "openpyxl>=3.1.5",
    "dask>=2026.1.1",
    "decorator>=5.2.1",
    "docutils>=0.22.4",
    "entrypoints>=0.4",
    "flask>=3.1.2",
    "networkx>=3.6.1",
    "graphviz>=0.21",
    "spacy>=3.7.5",
    "scispacy>=0.5.5",
    "nltk>=3.9.2",
]

[tool.uv]
package = true

[build-system]
requires = ["setuptools>=64"]
build-backend = "setuptools.build_meta"

[tool.setuptools.packages.find]
where = ["src"]
```

Atualizando o projeto

Atualizando toml dependencies:

```
biostat2026$  uv sync --upgrade
```

similar a rodar pip sobre requirements.txt

‘pip install -r requirements.txt’

Estrutura do UV

Ao criar e depois atualizar o projeto

Atualiza as dependências

Cria um .env - similar ao Virtual Environment - onde está instala versão correta do Python e todos os pacotes

Também cria o uv.lock - inclui todos os pacotes (versões) e dependências → para reprodutibilidade

Experimente digitar:
$ uv tree

árvore de pacotes + dependências é gerada

Instalando e Removendo pacotes

Instalando pacotes:

```
biostat2026$  uv add flask
```

similar a pip install <package>

Removendo pacotes:

```
biostat2026$  uv remove flask

# forma antiga é preservada
biostat2026$  uv pip uninstall flask
```

Listando pacotes

Listando pacotes:

```
biostat2026$  uv pip list
```

```
biostat2026$  uv tree
```

https://docs.astral.sh/uv/reference/cli/#uv

Decolando JupyterLab

JupyterLab

```
biostat2026$  uv run --with jupyter jupyter lab
```

Copie fontes e dados

Fontes:

Pegue os códigos de ‘day02c - python - introdução’ em coloque em ‘src’

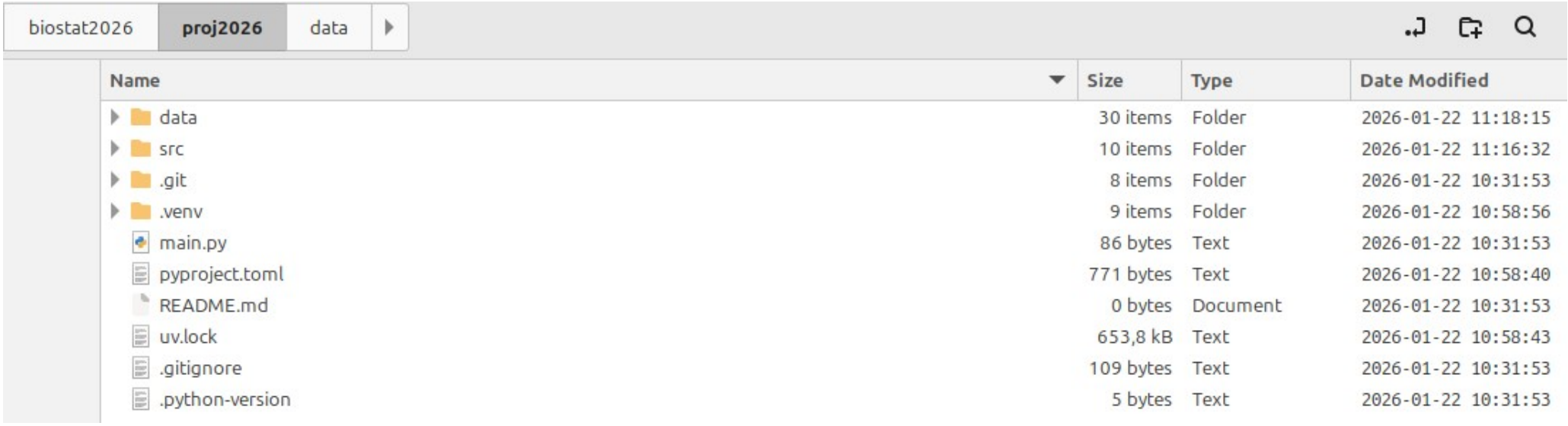
Dados:

Crie o diretório ‘data’ em paralelo com ‘src’

Baixe os dados fornecidos e copie em ‘data’

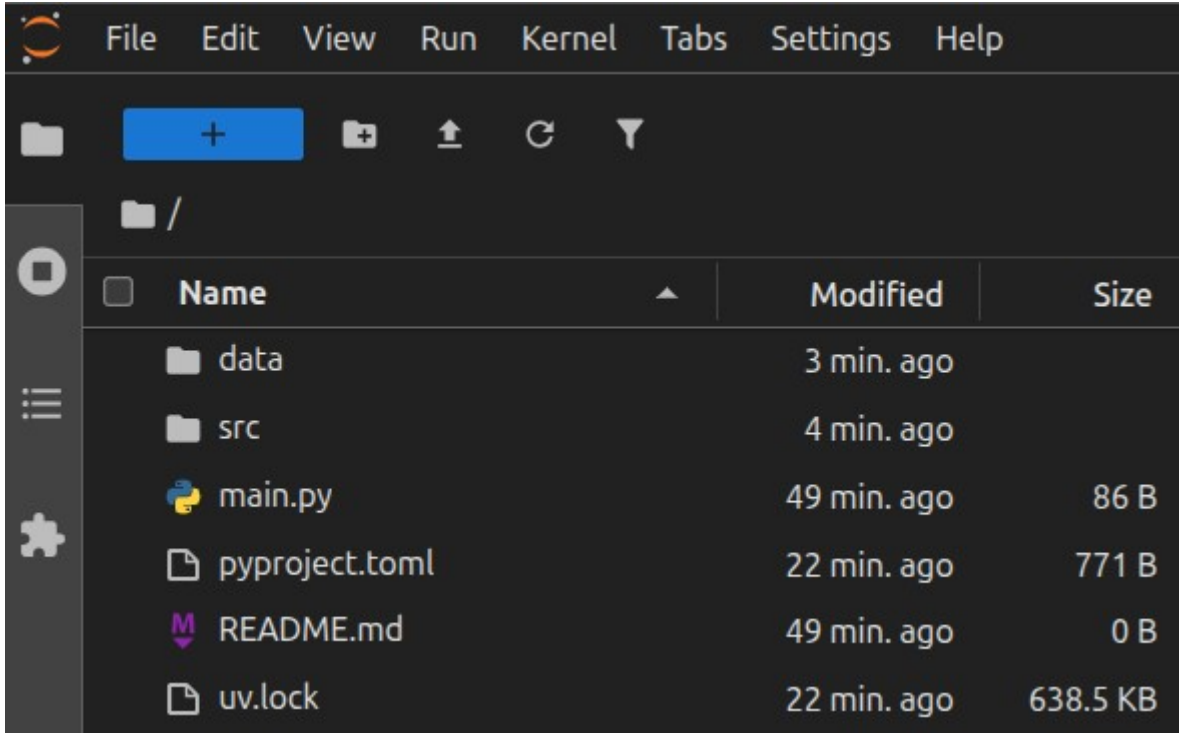
Explorer/Nemo

No seu gerenciador de arquivos poderá ver:



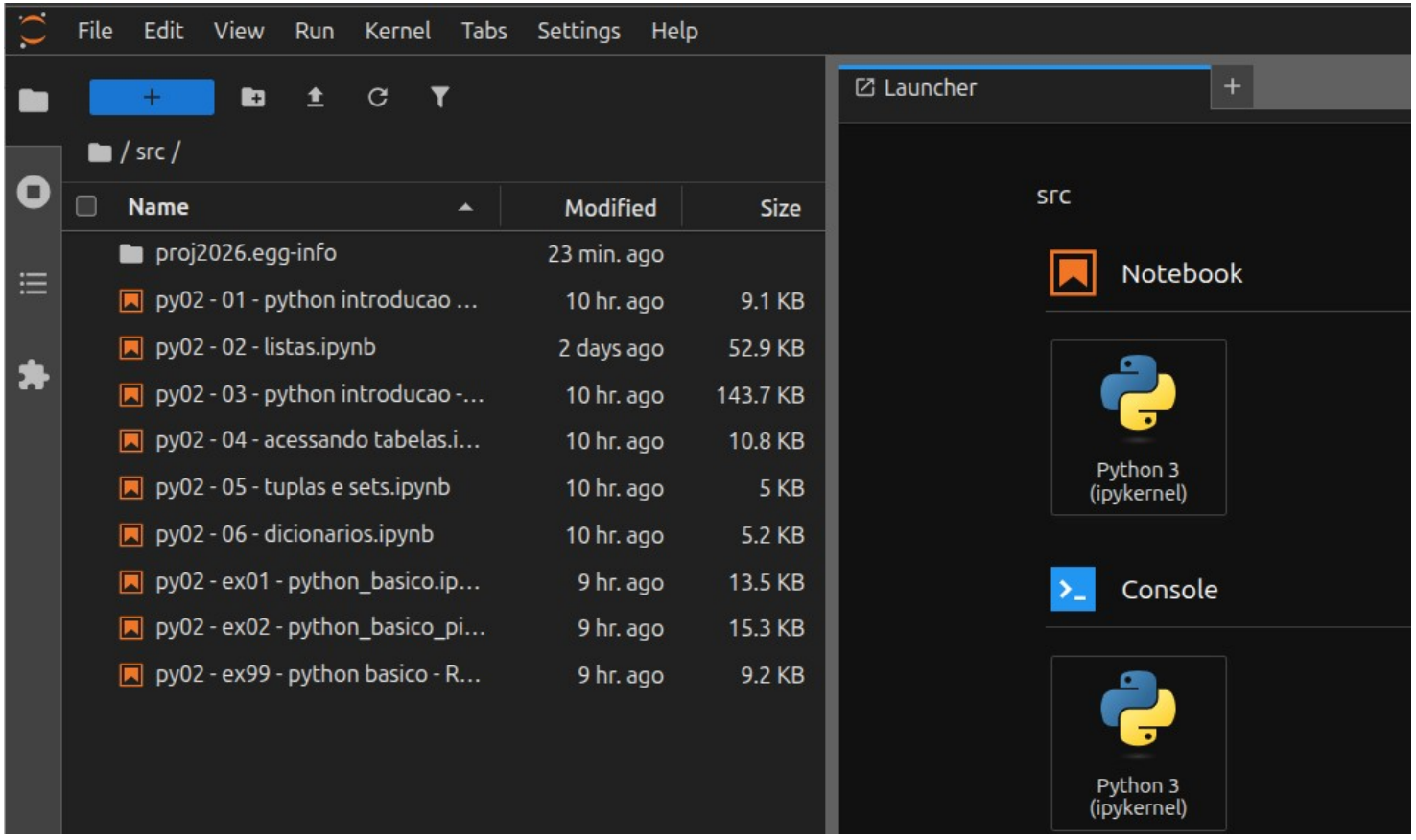
JupyterLab

No seu JupyterLab poderá ver:



JupyterLab

Clicando em ‘src’:



Obrigado
Perguntas?

PhD Flavio Lichtenstein

Bioinformatics & Systems Biology Lab
Molecular Biology Lab
Development and Innovation Center (CDI)

flavio.lichtenstein@butantan.gov.br

centre of excellence in new target discovery