MIRANDASTECH · MANUAL PROFISSIONAL

# Análise de dados na pesquisa

Do dado bruto ao resultado relatado: arrumar a tabela, descrever, comparar grupos com tamanho de efeito e intervalo de confiança, correlacionar, cruzar categorias, gerar cada figura por script e codificar entrevistas com um *codebook* — **com ferramentas gratuitas, execuções reais e cada número explicado**.


| SÉRIE | VERSÃO | FORMATO | LICENÇA |
|---|---|---|---|
| Manuais MirandasTech · pesquisa científica | 1.0 · setembro de 2026 | – slides | CC BY 4.0 — cite a fonte |

**COMO USAR** CADA SLIDE É UM PASSO QUE VOCÊ EXECUTA NA PLANILHA, NO TERMINAL, NO NOTEBOOK OU NO TAGUETTE

# Para quem é — e o que você vai saber fazer ao final

Este manual é para **pós-graduandos que precisam analisar os dados da própria pesquisa sem formação em estatística nem em programação**. Cada número que aparece — DP, IC, p, d, $\eta^2$ — é explicado numa frase quando surge pela primeira vez. Os resultados quantitativos são reais, obtidos em 09/09/2026 com um conjunto de dados aberto (Palmer Penguins, CC0) e scripts, um notebook e uma planilha executados de verdade nesta máquina. Os textos de entrevista da parte qualitativa são **fictícios e ilustrativos**, escritos para a demonstração. O manual não fez login em nenhum serviço (Colab, Planilhas Google, jamovi Cloud, OSF, Zenodo): o que exige conta é descrito pela documentação oficial e dito como tal.

**REGRA**

Versões, nomes de botão e citações são os das páginas oficiais observadas em 09/09/2026; quando a sua tela divergir, vale a tela. As saídas de terminal, o notebook e o projeto do Taguette são execuções reais nesta máquina (Python 3.14.7, pandas 3.0.5, SciPy 1.18.1, JupyterLab 4.6.3, Taguette 1.5.2, LibreOffice Calc 25.2.7).

**ARMADILHA**

**Escolher o teste pelo p mais bonito.** Rodar «todos os testes» e relatar o que deu significativo não é análise, é pesca. A técnica vem do projeto (Parte 1); o p vem depois do tamanho de efeito e do IC (Parte 3); e o que foi testado a mais precisa de correção e de declaração (slide «Múltiplas comparações»).

**MANUAIS IRMÃOS**

## Ao final, você vai conseguir

- **Decidir** a técnica pelo projeto, não pelo dado — e arrumar a tabela no formato tidy (Wickham, 2014), com o dado bruto intocado e a limpeza documentada.
- **Descrever e comparar** na planilha (LibreOffice Calc, Planilhas Google), em Python com pandas e SciPy, num notebook Jupyter ou num programa de menu (jamovi, JASP) — e obter o mesmo resultado em duas ferramentas.
- **Ler** um resultado: n por grupo, média e DP, diferença com IC 95 %, tamanho de efeito e p — e saber quando o teste robusto entra ao lado do paramétrico.
- **Codificar** entrevistas no Taguette com um *codebook* escrito antes, exportar os destaques e sintetizar por tema com trechos (Braun & Clarke, 2006).
- **Relatar** conforme SAMPL (quantitativo) e COREQ/SRQR (qualitativo), com cada figura gerada por script e nenhum número digitado à mão.
- **Rodar** três scripts: `limpar_dados.py`, `analisar.py` e `codificar_taguette.py`.

> Este manual cuida da *análise*. O projeto que decide a técnica está no de Metodologia; a literatura vem de Busca e PRISMA; as referências, do Zotero; o texto, do LaTeX; o versionamento dos scripts e dados derivados, do Git; a leitura assistida, do Notebook; a declaração, do Uso de IA; e o Plugin junta tudo. Hub: mirandastech.com.br/pesquisa.

INTRODUÇÃO O ROTEIRO DO MANUAL SEGUE A ORDEM DE UMA ANÁLISE

# Mapa: do dado bruto ao resultado relatado — e onde cada parte do manual entra

1

### O projeto decide

A técnica de análise foi escolhida antes da coleta, com o método; aqui só se executa o que está no projeto

2

### Bruto intocado, limpeza documentada

Cópia do arquivo original; `limpar_dados.py` gera o arquivo limpo e o relatório do que foi feito; dado pessoal anonimizado (LGPD)

3

### Tabela arrumada

Uma variável por coluna, uma observação por linha (tidy): é o formato que a planilha, o pandas, o jamovi e o JASP leem sem retrabalho

4

### Descrever

n por grupo, ausentes, média e DP, mediana e IQR — na planilha ou no `analisar.py`

5

### Comparar, correlacionar, cruzar

Diferença com IC 95 % e tamanho de efeito; teste robusto ao lado; p exato; conferido em duas ferramentas

6

### Qualitativo

Codebook → Taguette → exportar destaques → `codificar_taguette.py` → síntese por tema com trechos

7

**Relatar**

SAMPL e COREQ/SRQR; toda figura gerada por script que lê o dado; nenhum número digitado à mão

| PARTE | O QUE COBRE | PASSOS |
|---|---|---|
| **1 Antes de analisar** | A técnica vem do projeto · dados arrumados · bruto intocado, limpeza e LGPD · o conjunto da demonstração · onde achar dados abertos | Passos 1 a 3 |
| **2 Planilha** | Quando basta · LibreOffice Calc e a tabela dinâmica · fórmulas e o n que muda o DP · Planilhas Google e Excel · erros de planilha | Passo 4 |
| **3 Python e notebooks** | pandas, SciPy, Matplotlib · Jupyter, JupyterLab e Colab · os scripts · como ler os números · pressupostos · múltiplas comparações · figura por script | Passos 4, 5 e 7 |
| **4 Programas de menu** | jamovi e JASP · R e RStudio · OpenRefine e Orange · o mesmo resultado em duas ferramentas | Passos 2, 4 e 5 |
| **5 Qualitativo com o Taguette** | Análise temática e codebook · instalar, criar projeto, destacar, etiquetar, exportar · `codificar_taguette.py` · COREQ e SRQR | Passo 6 |
| **6 Relatar e fechar** | | Passo 7 |

**COMO LER**

Quem tem uma tabela e nunca analisou nada começa na Parte 1 e faz a Parte 2 com a planilha aberta ao lado. Quem quer resultados com efeito e IC sem programar vai à Parte 4 e volta ao slide «Como ler os números» da Parte 3. Quem tem entrevistas vai à Parte 5. Todos passam pela Parte 6 antes de escrever.

**CONVENÇÕES**

Nomes de menu e de botão entre «aspas angulares». Comandos em blocos com botão «copiar». Capturas de páginas em 1920 × 938, feitas em 09/09/2026 sem login. Figuras de terminal, do JupyterLab e do Taguette com «execução real nesta máquina». Números decimais com vírgula, como no português. Trechos de entrevista sempre marcados como fictícios.

| PARTE | O QUE COBRE | PASSOS |
| --- | --- | --- |
| | SAMPL · o plugin · erros comuns · checklist · glossário · referências · como citar | |

PARTE 1 ANTES DE ANALISAR

# A técnica vem do projeto, não do dado; dados arrumados; dado bruto intocado, limpeza documentada e LGPD; o conjunto da demonstração; onde achar dados abertos

Seis slides sem nenhum teste estatístico: o que precisa estar decidido e organizado antes de abrir a planilha ou o terminal — porque a maior parte dos erros de análise acontece aqui, e não na fórmula.

01

# A técnica vem do projeto, não do dado: a pergunta e o tipo de variável escolhem o teste — antes da coleta

O projeto de pesquisa (manual irmão de Metodologia) já diz qual pergunta será respondida, com que variáveis e por qual técnica. A análise **executa** essa decisão. Trocar de técnica depois de ver os dados é possível — mas fica registrado, com o motivo, e vai para o texto. O que não se faz é rodar tudo e escolher o que «deu certo».

| PERGUNTA | VARIÁVEIS | TÉCNICA (O QUE O `ANALISAR.PY` FAZ) |
|---|---|---|
| **Uma medida difere entre dois grupos?** | Uma numérica, uma categórica com 2 níveis | t de Welch com IC 95 % e d de Cohen; Mann-Whitney ao lado |
| **...entre três ou mais grupos?** | Uma numérica, uma categórica com 3+ níveis | ANOVA com $\eta^2$; Kruskal-Wallis ao lado |
| **Duas medidas variam juntas?** | Duas numéricas | Pearson com IC; Spearman ao lado |
| **Duas classificações estão associadas?** | Duas categóricas | Qui-quadrado com V de Cramér; aviso se algum esperado < 5 |
| **O que as pessoas dizem sobre X?** | Texto de entrevista, resposta aberta | Análise temática com codebook (Parte 5) — não é estatística |

### CAMINHO DO PROJETO

- Pergunta → variáveis → técnica, escritas no projeto antes da coleta
- Instrumento desenhado para produzir exatamente essas variáveis
- Tamanho de efeito e IC decididos como resultado principal; o p acompanha
- Mudança de técnica: registrada com data e motivo

### CAMINHO DO DADO

- Coleta primeiro; «depois a gente vê o que dá para testar»
- Vários testes até um $p < 0{,}05$ aparecer
- Grupos redefinidos, outliers apagados, variáveis criadas depois de ver o resultado
- Nada disso declarado no texto

**REGRA**

Antes de abrir o dado, escreva em uma linha: «vou comparar *Y* entre os grupos de *G* com *teste*, e o resultado principal é a diferença com IC 95 %». Se essa linha não existe no projeto, o passo é voltar ao Metodólogo, não ao software. Wilson et al. (2017) chamam isso de prática «boa o bastante»: decidir antes, registrar tudo.

# Dados arrumados (tidy): cada variável numa coluna, cada observação numa linha, cada tipo de unidade numa tabela

Wickham (2014) deu nome ao formato que toda ferramenta deste manual espera. Uma tabela «larga», bonita para ler, esconde uma variável nos cabeçalhos; a tabela arrumada repete o valor da categoria em cada linha e deixa a ferramenta agrupar, filtrar e cruzar sem retrabalho. A tabela dinâmica da Parte 2, o `groupby` do pandas e os menus do jamovi partem da mesma forma.

ANTES — «LARGA»: A ESPÉCIE VIROU CABEÇALHO, A MASSA ESTÁ ESPALHADA EM TRÊS COLUNAS (ESQUEMA)

| ISLAND | ADELIE | CHINSTRAP | GENTOO |
|---|---|---|---|
| ilha A | massa… | massa… | — |
| ilha B | massa… | — | massa… |

DEPOIS — ARRUMADA: UMA LINHA POR PINGUIM, UMA COLUNA POR VARIÁVEL (AS COLUNAS REAIS DO PALMER PENGUINS)

| SPECIES | ISLAND | BODY_MASS_G | SEX |
|---|---|---|---|
| Adelie | ilha A | … | female |
| Adelie | ilha A | … | male |
| Gentoo | ilha B | … | female |

| SINAL DE TABELA DESARRUMADA | COMO ARRUMAR |
|---|---|
| **Valores no cabeçalho** («2024», «2025», «Adelie») | Vira uma coluna `ano` ou `species` e uma coluna com valor |
| **Duas variáveis numa célula** («… g / macho») | Duas colunas: `body_mass_g` `sex` |
| **Unidade no valor** («… g») | Unidade no nome da coluna; c só com o número |
| **Total ou média no meio das linhas** | Fora da tabela de dados: a ferramenta calcula |
| **Células mescladas, cabeçalho em duas linhas** | Uma linha de cabeçalho, sem mesclar; cada coluna com nom único |
| **Ausente escrito de três jeitos** («», «NA», «-») | Um único marcador de ausente `limpar_dados.py` conta e a |

REGRA

Nomes de coluna curtos, sem acento, sem espaço, com a unidade quando houver ( `body_mass_g` , `flipper_length_mm` ): o pandas, o R, o jamovi e o JASP leem sem ajuste, e o script que gera a figura não

> precisa de tradução. A tabela «larga» pode existir — como *saída* para o texto, gerada a partir da arrumada.

ANTES DE ANALISAR 3 DE 6 · WILSON ET AL. (2017) · THE TURING WAY (2024) · LGPD · MANUAL DE GIT

# Dado bruto intocado, limpeza documentada, dado pessoal fora do repositório — três regras que valem antes de qualquer número

1

### Bruto

O arquivo como saiu do instrumento, do sensor ou da exportação; guardado com data; nunca editado, nem «só para corrigir uma célula»

2

### Cópia de trabalho

Tudo o que se faz é sobre uma cópia; o bruto continua onde estava, para quem precisar refazer

3

### `limpar_dados.py`

Lê a cópia, lista tipos, ausentes, duplicatas, variantes de texto, vírgula decimal, valores fora de 1,5 × IQR — e não apaga nada

4

### Limpo + relatório

Um arquivo novo ( `limpo.csv` ) e o registro do que foi decidido sobre cada aviso: é isso que a análise lê

REGRA

Wilson et al. (2017): dados brutos intocados, um script para cada transformação, nomes de arquivo previsíveis. Toda decisão de limpeza («os 2 ausentes de massa ficaram fora da média», «a duplicata da linha 40

| O QUE | ONDE FICA | ENTRA NO GIT? |
|---|---|---|
| **Dado bruto com dado pessoal** (nome, CPF, e-mail, gravação, transcrição identificável) | Pasta protegida fora do repositório ( `dados_FORA_DO_GIT/` ), com acesso restrito | NUNCA |
| **Dado bruto sem dado pessoal** (sensor, conjunto aberto) | A mesma pasta, ou o repositório de dados de origem | SÓ SE FOR PEQUENO E A LICENÇA PE |
| **Dado limpo e anonimizado** + relatório de limpeza | Pasta de dados do projeto | SIM |
| **Scripts, notebook, codebook** | Pasta de scripts | SIM |
| | | SIM |

> foi removida porque…») vai para um arquivo de texto ao lado do dado e depois para o método. O que não está escrito não aconteceu.

| O QUE | ONDE FICA | ENTRA NO GIT? |
|---|---|---|
| **Tabelas e figuras de resultado** | Pasta de saídas, geradas por script | |

ARMADILHA

«**Anonimizei: troquei o nome por um código**». Um código ligado a uma tabela que ainda tem o nome não anonimiza nada; e uma transcrição de entrevista identifica pela história contada. Antes da análise, a tabela de correspondência fica fora do repositório e a transcrição passa por remoção de nomes, lugares e datas que identifiquem. A LGPD trata dado de pessoa identificável como pessoal mesmo em pasta «privada».

# O conjunto da demonstração: Palmer Penguins — 344 pinguins, três espécies, oito colunas, licença CC0

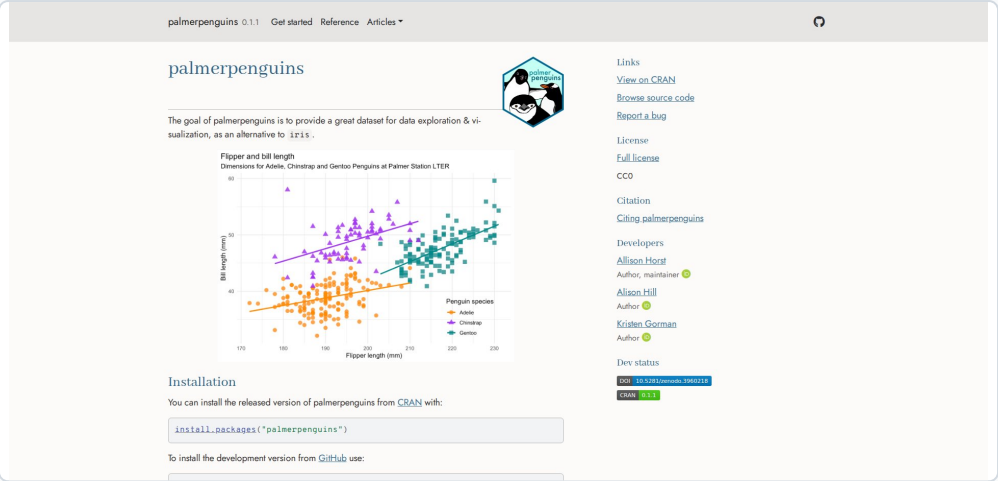



**FIGURA 1** allisonhorst.github.io/palmerpenguins (captura de 09/09/2026): o pacote e o CSV do conjunto Palmer Penguins, coletado por Kristen Gorman com a Palmer Station LTER e distribuído sob CC0. É o conjunto usado em todas as execuções deste manual — planilha, scripts, notebook.

| ITEM | VALOR |
|---|---|
| **Registros** | 344 pinguins de três espécies (Adelie, Chinstrap, Gentoo) em três ilhas do arquipélago Palmer, Antá |
| **Colunas** | `species`, `island`, `bill_length_mm`, `bill_depth_mm`, `flipper_length_mm`, `body_mass_g`, `sex`, `year` |
| **Ausentes** | 2 nas medidas, 11 em `sex` — por isso os n muda análise para análise (342 na correlação, 333 no qu quadrado) |
| **Origem** | Gorman, Williams e Fraser (2014), PLoS ONE, DOI 10.1371/journal.pone.0090081 |
| **Licença** | CC0 — pode ser copiado, redistribuído e usado em sem pedir permissão |

**REGRA**

Aprenda a ferramenta num conjunto aberto e pequeno antes de tocar no seu dado: com o Palmer Penguins você confere se o resultado bate com o deste manual; com o seu dado, um erro de ferramenta e um erro de dado ficam indistinguíveis. A pergunta da demonstração é

simples de propósito: «a massa corporal difere entre espécies e entre sexos?».

POR QUE NÃO «IRIS» OU UM CONJUNTO INVENTADO

Um conjunto com ausentes, três grupos de tamanhos diferentes e uma coluna categórica com falhas reproduz o que aparece em dados reais de pesquisa. Dados simulados só entram neste manual com o rótulo «fictício» — e nunca misturados aos reais.

# Onde achar dados abertos (1 de 2): o Portal Brasileiro de Dados Abertos e o SIDRA do IBGE

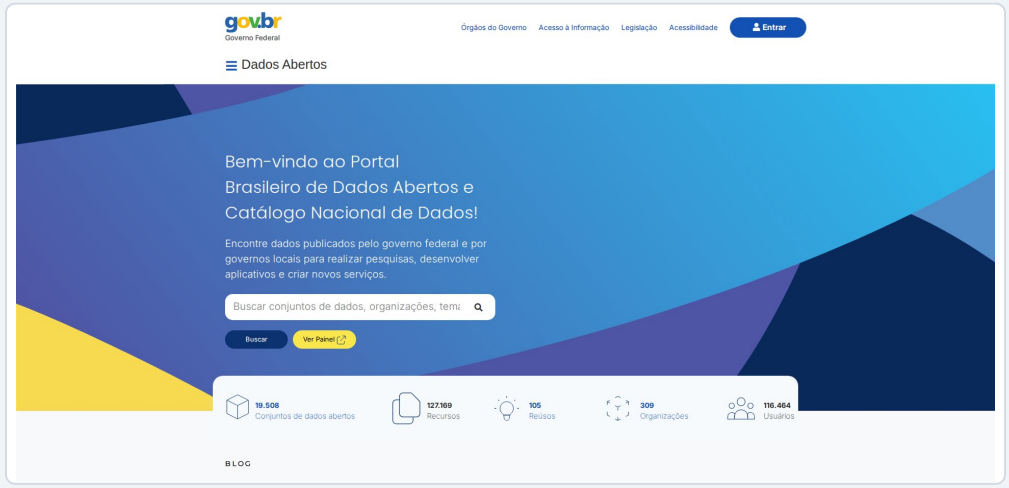



**FIGURA 2** dados.gov.br (captura de 09/09/2026): o «Portal Brasileiro de Dados Abertos e Catálogo Nacional de Dados» — a página inicial com a busca de conjuntos publicados por órgãos públicos. O manual não afirma quantos conjuntos há: o número não foi conferido nesta data.

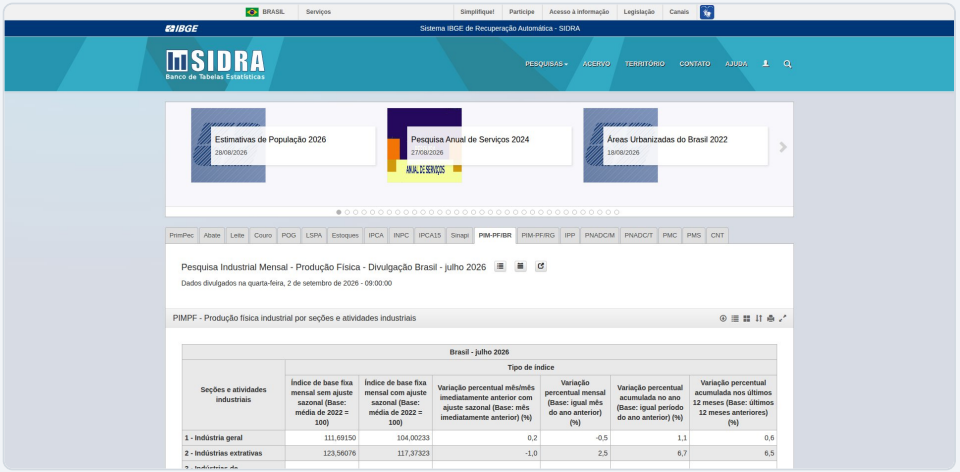



**FIGURA 3** sidra.ibge.gov.br (captura de 09/09/2026): a porta de entrada das tabelas do IBGE. O número da tabela e o período que você usar vão para o método, como qualquer fonte de dado.

| FONTE | SERVE PARA | O QUE CONFERIR ANTES DE USAR |
|---|---|---|
| **dados.gov.br** | Dados administrativos de órgãos públicos brasileiros: educação, saúde, | Licença do conjunto, data de atualização, dicionário de variáveis |

**REGRA**

Um dado aberto é bruto como qualquer outro: baixe, guarde intocado com a data e a URL, e passe pelo `limpar_dados.py`. No texto, cite a fonte, a tabela, a data de acesso e a licença — e diga o que foi filtrado ou recodificado. A exportação do SIDRA costuma vir «larga» (anos nas colunas): o slide «Dados arrumados» resolve.

| FONTE | SERVE PARA | O QUE CONFERIR ANTES DE USAR |
|---|---|---|
| | transporte, orçamento | |
| **SIDRA / IBGE** | Séries e tabelas oficiais por município, estado e país | A tabela e o período exatos (o número da tabela vai para o método) |

# Onde achar dados abertos (2 de 2): OSF e Zenodo — dados de outras pesquisas, com DOI e licença declarada

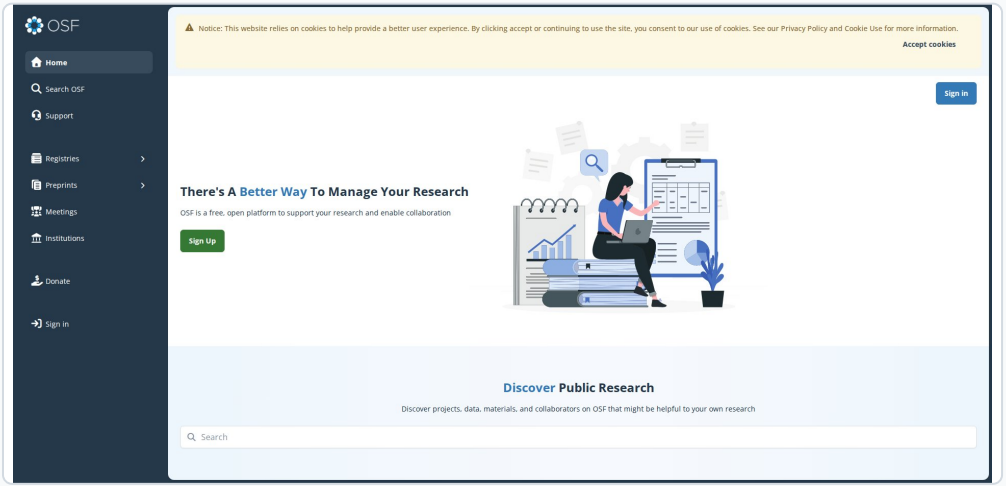



**FIGURA 4** osf.io (captura de 09/09/2026): «There's a better way to manage your research». O Open Science Framework hospeda projetos, pré-registros e materiais de pesquisa; muitos artigos depositam ali os dados que o texto analisa. Só a página pública de entrada foi capturada; o manual não fez login.

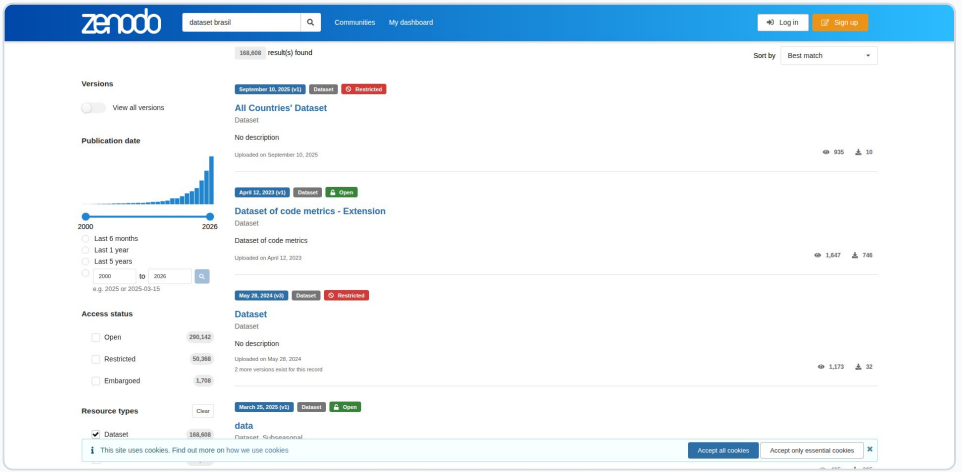



**FIGURA 5** A busca por datasets no Zenodo (captura de 09/09/2026): o repositório aberto que registra dados e código com DOI — e é assim que o seu dado derivado também pode ser publicado (manual irmão de Git, que mostra o DOI pelo Zenodo).

| FONTE | SERVE PARA | O QUE CONFERIR ANTES DE USAR |
|---|---|---|
| **OSF** | Dados, instrumentos e pré-registros de estudos publicados; replicações | Licença do componente, se o dado é bruto ou derivado, o artigo de origem |

**REGRA**

Reutilizar o dado de outra pesquisa é citar o dado (DOI, versão) *e* o artigo que o descreve — como este manual cita o Palmer Penguins pelo pacote e por Gorman, Williams e Fraser (2014). Sem licença explícita no registro, pergunte ao autor antes de analisar; «está na internet» não é licença.

| FONTE | SERVE PARA | O QUE CONFERIR ANTES DE USAR |
|---|---|---|
| **Zenodo** | Conjuntos de dados e código com DOI, de qualquer área | Versão do registro, licença e a descrição que o acompanha |

SEM NÚMEROS

Nem o OSF nem o Zenodo tiveram acervo, limites de tamanho ou planos conferidos nesta data; o manual só mostra as páginas públicas de entrada e de busca.

PARTE 2 PLANILHA

# Quando a planilha basta; LibreOffice Calc e a tabela dinâmica passo a passo; COUNTIF, AVERAGEIF e o n que muda o DP; Planilhas Google e Excel; erros de planilha

02

Cinco slides para quem vai ficar na planilha — e para quem vai sair dela sabendo por quê. O exemplo do desvio padrão que «não bate» com o Python é real, aconteceu na demonstração de 09/09/2026, e é a melhor aula sobre conferir o n.

PLANILHA 1 DE 5

# Quando a planilha basta: contar, somar, tirar média e DP por grupo, montar a tabela descritiva — e quando ela para de bastar

### BASTA

- Conferir a tabela: colunas, tipos, ausentes, valores estranhos — a olho e com filtro
- Descritivas: n, média, DP, mediana, mínimo e máximo por grupo (tabela dinâmica ou fórmulas)
- Tabela cruzada de duas categorias (contagens)
- Preparar a tabela arrumada que o jamovi, o JASP ou o pandas vão ler
- A tabela descritiva do texto, depois conferida numa segunda ferramenta

### PARA DE BASTAR

- Teste com IC 95 % e tamanho de efeito: dá para montar por fórmula, mas cada passo é uma célula que alguém pode sobrescrever
- Teste robusto (Mann-Whitney, Kruskal-Wallis, Spearman): sem função pronta
- Figura para o artigo: um gráfico de planilha não é gerado por script e não se refaz sozinho quando o dado muda
- Mais de umas centenas de linhas por muitas colunas: lento e fácil de errar
- Reprodutibilidade: ninguém sabe o que foi digitado e o que foi calculado

### REGRA

A planilha é boa para *olhar* o dado e para a primeira descritiva; é ruim para *guardar* a análise. Se você vai ficar nela, duas regras: nenhuma célula de resultado digitada à mão (tudo fórmula, com o intervalo visível) e o arquivo de dados separado do arquivo de cálculo. Se vai sair dela, saia depois de ter a tabela arrumada — o resto das ferramentas lê CSV.

### O QUE VEM A SEGUIR

LibreOffice Calc (gratuito, instalado localmente, é o que rodou nesta máquina), Planilhas Google (on-line, exige conta Google) e Excel (pago, Microsoft 365). Os passos da tabela dinâmica são os da ajuda oficial do Calc em português; a lógica é a mesma nos três.

# LibreOffice Calc e a tabela dinâmica: «Inserir – Tabela dinâmica» → «Seleção atual» → arrastar os cabeçalhos → OK

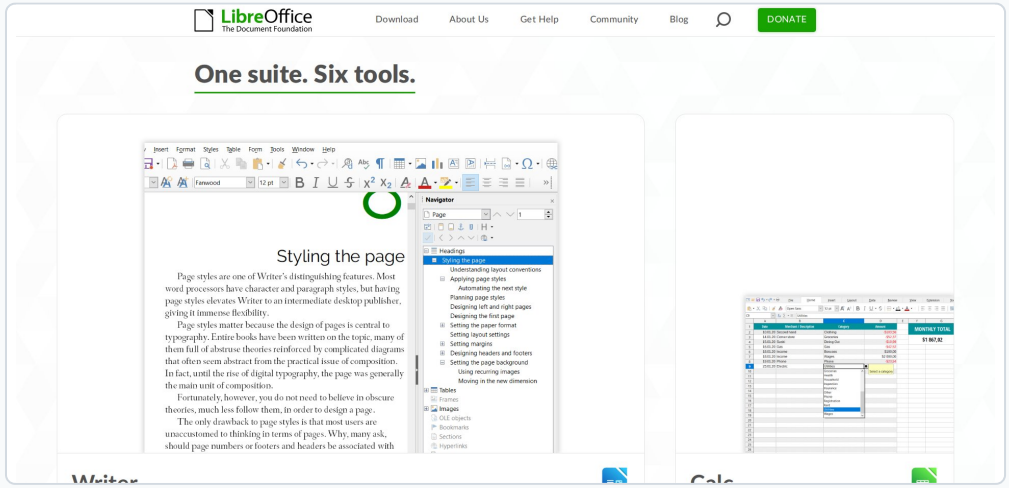



**FIGURA 6** libreoffice.org (captura de 09/09/2026): o Calc é a «planilha rica em recursos para analisar dados, calcular e criar visuais claros»; código sob Mozilla Public License v2.0; versão anunciada no site: LibreOffice 26.8. A planilha da demonstração foi recalculada nesta máquina pelo Calc 25.2.7, sem interface gráfica.


**FIGURA 7** «Criar tabelas dinâmicas», na ajuda do Calc em português (captura de 09/09/2026): o cursor no intervalo de dados → «Inserir – Tabela dinâmica» → «Seleção atual» → arrastar os cabeçalhos para Filtros, Campos de coluna, Campos de linha e Campos de dados → OK.

1

### Cursor no intervalo

Uma célula qualquer da tabela arrumada (com a linha de cabeçalho)

2

### «Inserir – Tabela dinâmica»

→ «Seleção atual» → OK: abre o assistente com os cabeçalhos como campos

**REGRA**

A tabela dinâmica é a primeira descritiva por grupo sem fórmula nenhuma: n (contagem), média e desvio padrão por espécie em quatro cliques. Só funciona sobre a tabela arrumada (Parte 1). Copie o resultado para o texto como *valores*, mas guarde a planilha com a tabela dinâmica viva — ela é a prova de como o número foi obtido.

3

**Arrastar**

`species` para Campos de linha; `body_mass_g` para Campos de dados — escolhendo a função (contagem, média, desvio padrão) no campo de dados

4

**OK**

A tabela dinâmica é criada; guarde a planilha com ela viva, não só o valor copiado

SEM CAPTURA DA INTERFACE

O Calc rodou nesta máquina sem tela (recalculo automático da planilha da demonstração), por isso não há captura da janela do assistente; os nomes dos passos são os da ajuda oficial em português (Figura 7).

# COUNTIF, AVERAGEIF e o n que muda o DP: a planilha e o Python deram a mesma média — e um desvio padrão diferente

Três palavras antes dos números: **n** é quantas observações entraram no cálculo; **média** é a soma dividida por n; **DP** (desvio padrão) é quanto os valores se afastam da média, na mesma unidade da variável — um DP de 458,57 g diz que a massa dos Adelie varia tipicamente uns 458,57 g em torno da média. A média ignora a célula vazia sozinha; o DP calculado à mão, não.

CALC — AS FÓRMULAS USADAS NA DEMONSTRAÇÃO (ESQUEMA; O SEPARADOR DE ARGUMENTOS SEGUE A CONFIGURAÇÃO REGIONAL DA SUA PLANILHA)

```
=COUNTIF(A:A; "Adelie")                          n: conta as linhas
da espécie (inclui a que tem massa vazia!)
=AVERAGEIF(A:A; "Adelie"; F:F)                   média da massa da
espécie (ignora a célula vazia)
=SQRT(SUMPRODUCT((A2:A345="Adelie")*(F2:F345-média)^2) / (n-1))
                                                 DP «à mão»: se n
veio do COUNTIF, o denominador conta a linha sem massa
```

| ESPÉCIE | PYTHON: N · MÉDIA · DP | CALC: LINHAS · MÉDIA · |
|---|---|---|
| **Adelie** | 151 · 3700,66 g · 458,57 | 152 · 3700,66 g · **457,05** |
| **Chinstrap** | 68 · 3733,09 g · 384,34 | 68 · 3733,09 g · 384,34 |
| **Gentoo** | 123 · 5076,02 g · 504,12 | 124 · 5076,02 g · **502,06** |

### O QUE ACONTECEU — DE VERDADE

As médias são idênticas nas duas ferramentas. O DP difere em Adelie e Gentoo porque a fórmula do Calc contou, no n do denominador, a linha em que a massa está ausente (n − 1 = 151 em vez de 150 para Adelie): o Python descarta a linha sem valor antes de calcular; o COUNTIF contou a espécie, não a massa. Chinstrap, sem ausente, bate exatamente. **Lição: antes de comparar resultados entre ferramentas, compare o n.**

### ARMADILHA

Publicar o DP da planilha «porque deu quase igual». A diferença aqui é pequena; num conjunto com muitos ausentes ela não é. Conferir o n por grupo, com os ausentes contados à parte, é o primeiro item do relato SAMPL (Parte 6).

# Planilhas Google e Excel: a mesma lógica, on-line com conta Google — ou paga, no Microsoft 365

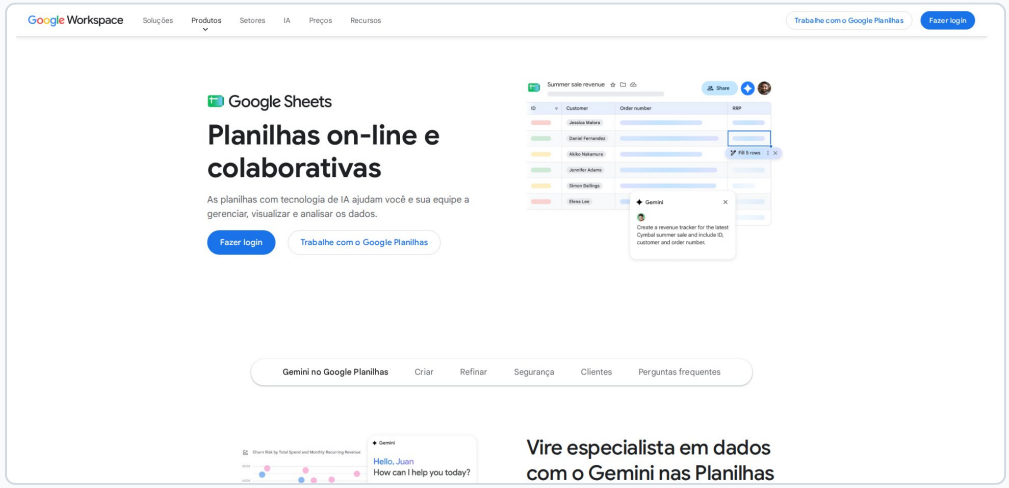



**FIGURA 8** A página do Planilhas Google no Google Workspace (captura de 09/09/2026): «Planilhas on-line e colaborativas»; a página promove o Gemini dentro das planilhas. Exige conta Google; a interface logada não foi capturada e o manual não cita limites de linhas ou de arquivo.

| FERRAMENTA | ONDE RODA · CUSTO | O QUE SABER |
|---|---|---|
| **LibreOffice Calc** | No seu computador · gratuito (MPL v2.0) | Instalado localmente, funciona s internet; é o que rodou nesta má (25.2.7); versão anunciada no si 26.8 |
| **Planilhas Google** | No navegador · conta Google | «Planilhas on-line e colaborativa descrição oficial; o arquivo fica n nuvem da Google — dado pessoa participante não entra sem anonimizar; a interface logada n capturada |
| **Excel** | Computador ou navegador · pago (Microsoft 365) | A página da Microsoft não respo na conferência de 09/09/2026: manual o cita só como opção pa sem preço nem versão |
| **GNU PSPP** | No seu computador · gratuito | Alternativa livre ao SPSS para qu vem dele; a página não pôde ser conferida nesta data (sem versã manual) |

REGRA

> Escolha pela regra do dado, não pela conveniência: dado com informação pessoal fica em ferramenta local (Calc) até a anonimização; o que já está anonimizado pode ir para a nuvem para colaboração. Em qualquer uma, exporte a tabela arrumada em CSV ao fim — é o formato que a Parte 3 e a Parte 4 leem.

PLANILHA 5 DE 5

# Erros de planilha: datas que viram números, vírgula decimal, célula digitada, cópia de valores — como aparecem e como evitar

| ERRO | COMO APARECE | CAUSA | COMO EVITAR |
|---|---|---|---|
| **Data vira número** (ou número vira data) | «45900» onde havia uma data; «1/2» virou «1 de fev.» | A planilha adivinha o tipo da célula ao colar ou importar | COLUNA COMO TEXTO NA IMPORTAÇÃO; DATAS EM AAAA-MM-DD |
| **Vírgula decimal** | «3,75» tratado como texto (alinhado à esquerda), média «errada» ou vazia | Arquivo com vírgula aberto em configuração com ponto, ou o contrário | UM SEPARADOR SÓ; O LIMPAR_DADOS.PY AVISA E CONVERTE |
| **Célula de resultado digitada** | Uma média que não muda quando o dado muda | Alguém «ajustou» um valor à mão | NENHUM NÚMERO DIGITADO: TUDO FÓRMULA OU SCRIPT |
| **Colar «só valores»** | A tabela de resultados perdeu as fórmulas; ninguém sabe de onde veio | Cópia para «limpar» a planilha | GUARDAR A VERSÃO COM FÓRMULAS; COLAR VALORES SÓ NA CÓPIA PARA O TEXTO |

| ERRO | COMO APARECE | CAUSA | COMO EVITAR |
|---|---|---|---|
| **Filtro aplicado ao calcular** | Média de «todos» que na verdade é de um subconjunto | Filtro ativo esquecido; funções que ignoram linhas ocultas | LIMPAR FILTROS ANTES; CONFERIR O N |
| **Ordenar uma coluna só** | As massas de um pinguim foram parar em outro | Seleção parcial antes de ordenar | ORDENAR SEMPRE A TABELA INTEIRA; MELHOR: NÃO ORDENAR O DADO, ORDENAR A SAÍDA |
| **Célula mesclada, cabeçalho duplo** | A importação no pandas, no jamovi ou no JASP quebra ou desloca colunas | Tabela feita para leitura, não para análise | UMA LINHA DE CABEÇALHO; SEM MESCLAR (SLIDE «DADOS ARRUMADOS») |
| **Ausente que vira zero** | Média puxada para baixo; n «cheio» | Célula vazia preenchida com 0 para «fechar» a tabela | AUSENTE É VAZIO, NÃO ZERO; CONTAR AUSENTES À PARTE |
| **n do COUNTIF ≠ n da medida** | DP diferente do da outra ferramenta (slide anterior) | Contou a categoria, não a coluna com valor | CONTAR A COLUNA DA VARIÁVEL; COMPARAR N ANTES DE COMPARAR RESULTADOS |
| **Gráfico da planilha no artigo** | Figura sem unidade no eixo, sem n, que não se refaz quando o dado muda | Feita à mão, colada como imagem | FIGURA POR SCRIPT (SLIDE «FIGURA POR SCRIPT, NUNCA À MÃO») |

PARTE 3 PYTHON E NOTEBOOKS

# pandas, SciPy e Matplotlib; Jupyter, JupyterLab e Colab; `limpar_dados.py` e `analisar.py` com saídas reais; como ler os números; pressupostos e testes robustos; múltiplas comparações; figura por script

Onze slides com o terminal e o notebook abertos. Todos os resultados são os da execução de 09/09/2026 sobre o Palmer Penguins (Python 3.14.7, pandas 3.0.5, SciPy 1.18.1, Matplotlib 3.11.1, seaborn 0.13.2, JupyterLab 4.6.3) — e cada número é explicado quando aparece.

03

# Python com pandas, SciPy e Matplotlib: a tabela, os testes e a figura — gratuitos, instalados com um comando

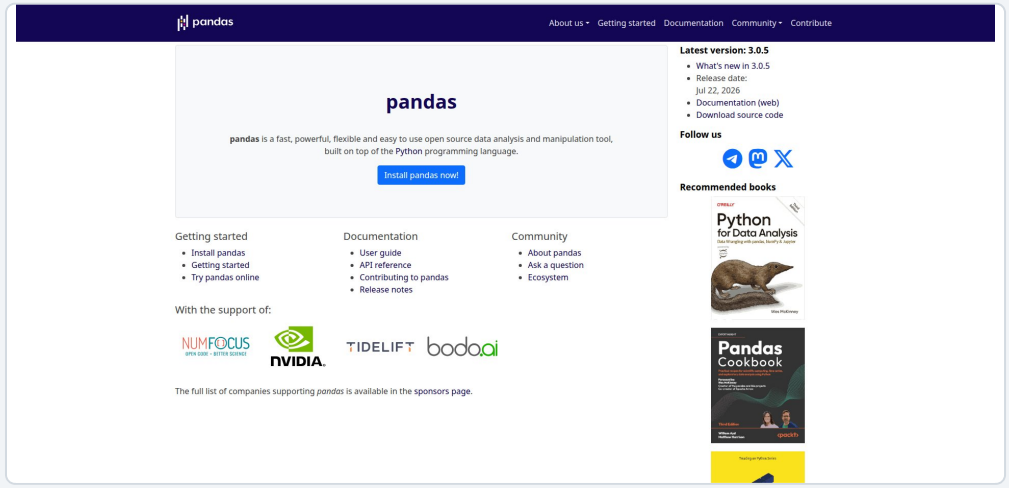



**FIGURA 9** pandas.pydata.org (captura de 09/09/2026): pandas 3.0.5 (22/07/2026), «ferramenta rápida, poderosa, flexível e fácil de usar para análise e manipulação de dados, de código aberto, construída sobre Python»; licença BSD-3; `pip install pandas`.

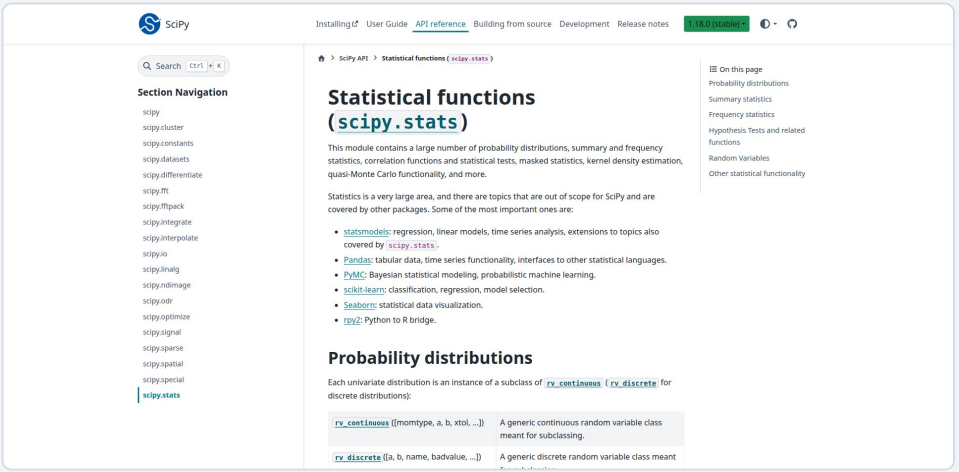



**FIGURA 10** A documentação de `scipy.stats` em docs.scipy.org (captura de 09/09/2026): distribuições, testes e correlação — é de onde o `analisar.py` tira o t de Welch, o Mann-Whitney, a ANOVA, o Kruskal-Wallis, o Pearson, o Spearman e o qui-quadrado.

TERMINAL — INSTALAR TUDO O QUE ESTE MANUAL USA (PYTHON 3.14.7 EM PYTHON.ORG/DOWNLOADS; AS VERSÕES ABAIXO SÃO AS DESTA MÁQUINA)

```
python3 --version                                   # 3.14.7
nesta máquina
pip install pandas scipy matplotlib seaborn jupyterlab
# pandas 3.0.5 · SciPy 1.18.1 · Matplotlib 3.11.1 · seaborn
0.13.2 · JupyterLab 4.6.3
```

| BIBLIOTECA | FAZ | NESTE MANUAL |
|---|---|---|
| **pandas** 3.0.5 | Lê CSV e planilha, arruma, agrupa, conta ausentes, descreve | `limpar_dados.py`, descritivas |
| **SciPy** 1.18.1 | `scipy.stats`: testes, correlação, distribuições | todos os testes do `analisar.p` |

| BIBLIOTECA | FAZ | NESTE MANUAL |
| --- | --- | --- |
| **Matplotlib** 3.11.1 | «Biblioteca abrangente para criar visualizações estáticas, animadas e interativas em Python» | a figura do `analisar.py --fi` |
| **seaborn** 0.13.2 | Gráficos estatísticos sobre o Matplotlib | o boxplot do notebook |

> **REGRA**
>
> Você não precisa «aprender Python» para usar a Parte 3: os scripts recebem o CSV e os nomes das colunas e imprimem o resultado. Precisa, sim, saber ler o que sai — é o que os slides «Como ler os números» e «Pressupostos» fazem. Quem quiser ir além tem o notebook (slide seguinte), onde cada linha pode ser alterada e reexecutada.

# Jupyter e JupyterLab: código, resultado e texto no mesmo arquivo — o notebook da demonstração, executado de verdade

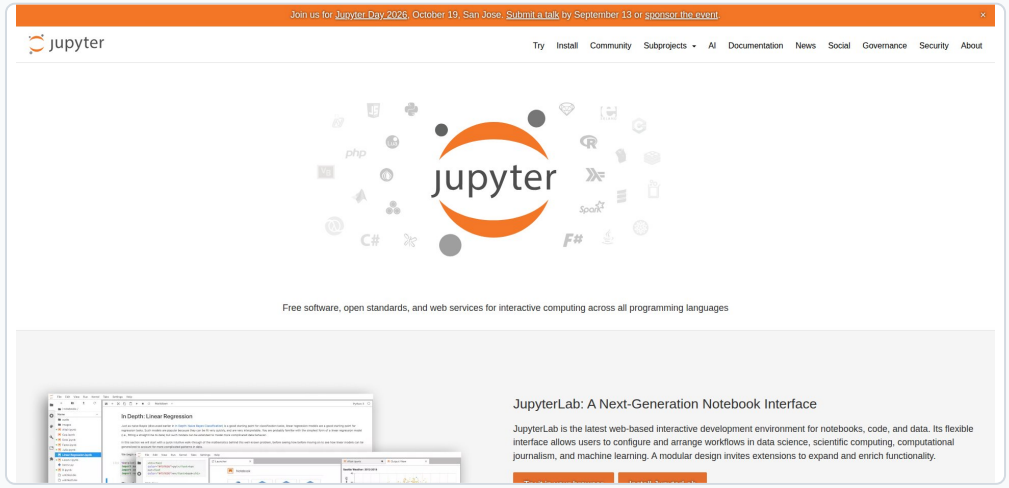



**FIGURA 11** jupyter.org (captura de 09/09/2026): «software livre, padrões abertos e serviços web para computação interativa em todas as linguagens». O JupyterLab é «o ambiente de desenvolvimento interativo baseado na web mais recente para notebooks, código e dados»; o Jupyter Notebook é a aplicação original; formato aberto em JSON (`.ipynb`); mais de 40 linguagens.

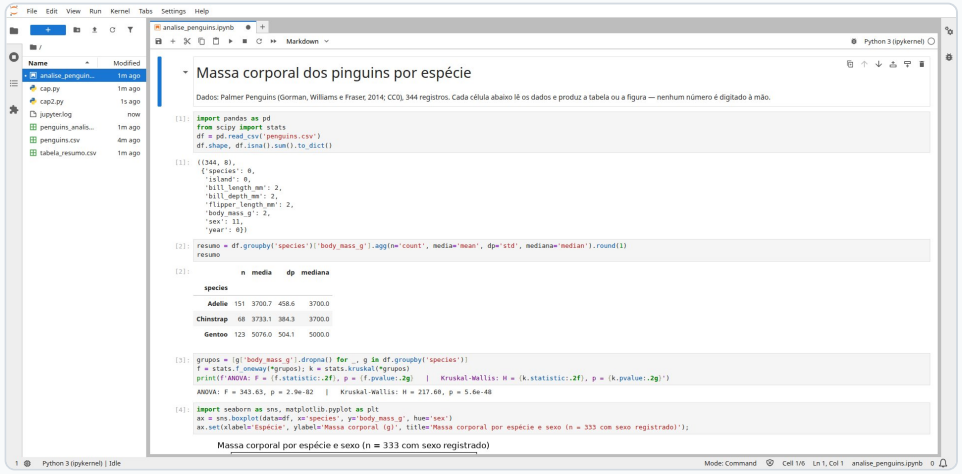



**FIGURA 12** O notebook da demonstração aberto no JupyterLab 4.6.3 local (captura de 09/09/2026, execução real nesta máquina): células executadas com o tamanho da tabela e os ausentes, o resumo de massa por espécie e a ANOVA com o Kruskal-Wallis — os mesmos números do `analisar.py` (slide «analisar.py: grupos»).

TERMINAL — ABRIR O JUPYTERLAB NA PASTA DO PROJETO (ABRE NO NAVEGADOR, EM LOCALHOST)

```
pip install jupyterlab      # uma vez
jupyter lab                 # abre a interface; Ctrl+C no
terminal encerra
```

| TERMO | O QUE É |
|---|---|
| **Notebook** (`.ipynb`) | Arquivo com células de código e de texto e as saídas gravadas; abre em qualquer JupyterLab e no Colab |
| **Célula** | Um bloco de código executado de uma vez; a saída ap logo abaixo |
| **Kernel** | |

| TERMO | O QUE É |
| --- | --- |
| | O processo Python que executa as células e guarda as variáveis na memória; reiniciá-lo apaga tudo o que est carregado |
| **JupyterLab × Notebook** | A mesma coisa por dentro; o Lab tem abas, terminal e navegador de arquivos ao lado |

**REGRA**

O notebook é onde se explora e se explica; o script é o que se entrega. Um resultado do texto pode nascer no notebook, mas termina num script (ou num notebook executado de cima a baixo, slide seguinte) que qualquer pessoa roda e obtém o mesmo número.

# O notebook de cima a baixo: um notebook só vale quando executa inteiro, do zero, e chega à mesma figura

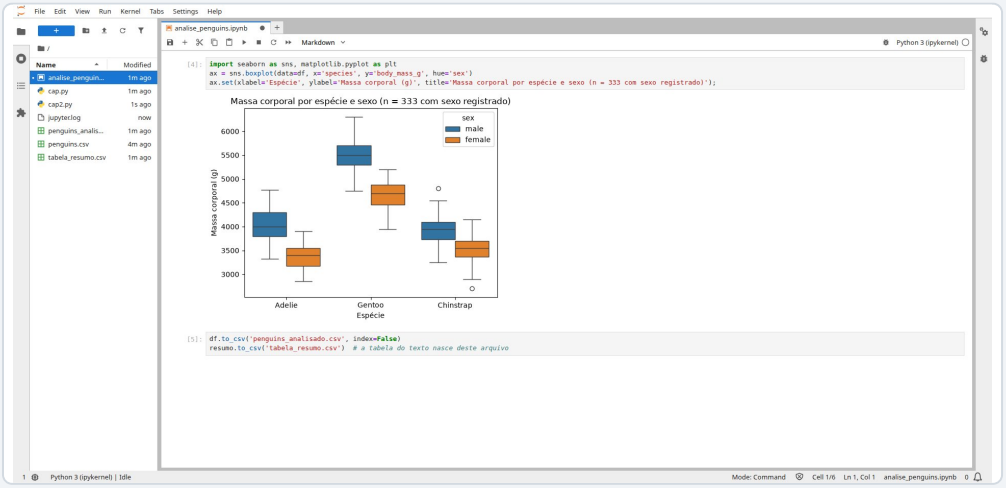



**FIGURA 13** A última célula do notebook da demonstração no JupyterLab (captura de 09/09/2026, execução real nesta máquina): o boxplot da massa corporal por espécie e sexo gerado com seaborn, logo abaixo do código que o produziu. A figura e o código estão no mesmo arquivo — quem abrir o notebook sabe exatamente como ela foi feita.

O QUE AS CÉLULAS DO NOTEBOOK FAZEM (ESQUEMA; A SAÍDA REAL ESTÁ NAS FIGURAS 12 E 13)

```
import pandas as pd
from scipy import stats
df = pd.read_csv("penguins.csv")
df.shape, df.isna().sum()                       # 344 linhas × 8 colunas; 2 ausentes nas medidas, 11 em sex
df.groupby("species")["body_mass_g"].describe()   # n, média, DP, mediana por espécie
g = [d.dropna() for _, d in df.groupby("species")["body_mass_g"]]
stats.f_oneway(*g); stats.kruskal(*g)           # ANOVA e Kruskal-Wallis
import seaborn as sns
sns.boxplot(data=df, x="species", y="body_mass_g", hue="sex")
```

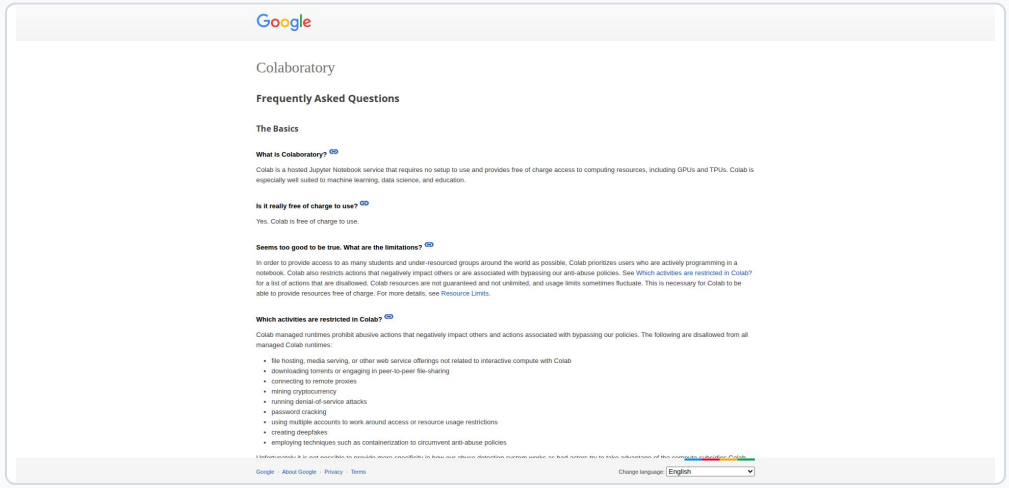
TERMINAL — EXECUTAR O NOTEBOOK INTEIRO SEM ABRIR A INTERFACE (FOI ASSIM QUE A DEMONSTRAÇÃO FOI CONFERIDA)

```
jupyter nbconvert --execute analise.ipynb     # executa todas as células, na ordem, num kernel novo
```

**ARMADILHA**

**O notebook que «funciona» só na sua tela.** Células executadas fora de ordem deixam variáveis na memória do kernel que não existem no arquivo; quem abrir do zero recebe erro na terceira célula. Antes de entregar: reinicie o kernel, execute tudo de cima a baixo — ou rode o `nbconvert --execute` acima. Se falhar, o notebook não estava pronto.

# Google Colab: o Jupyter hospedado, gratuito e sem instalar nada — com limites que «às vezes flutuam»


Google

Colaboratory

Frequently Asked Questions

The Basics

What is Colaboratory?

Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free of charge access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

Is it really free of charge to use?

Yes. Colab is free of charge to use.

Seems too good to be true. What are the limitations?

In order to provide access to as many students and under-resourced groups around the world as possible, Colab prioritizes users who are actively programming in a notebook. Colab also restricts actions that negatively impact others or are associated with bypassing our anti-abuse policies. See Which activities are restricted in Colab? for a list of actions that are disallowed. Colab resources are not guaranteed and not unlimited, and usage limits sometimes fluctuate. This is necessary for Colab to be able to provide resources free of charge. For more details, see Resource Limits.

Which activities are restricted in Colab?

Colab managed runtimes prohibit abusive actions that negatively impact others and actions associated with bypassing our policies. The following are disallowed from all managed Colab runtimes:

- file hosting, media serving, or other web service offerings not related to interactive compute with Colab
- downloading torrents or engaging in peer-to-peer file-sharing
- connecting to remote proxies
- mining cryptocurrency
- running denial-of-service attacks
- password cracking
- using multiple accounts to work around access or resource usage restrictions
- creating deepfakes
- employing techniques such as containerization to circumvent anti-abuse policies

Google · About Google · Privacy · Terms

Change language: English

**FIGURA 14** A FAQ do Colab (captura de 09/09/2026): «serviço de Jupyter Notebook hospedado que não exige configuração e dá acesso gratuito a recursos de computação, incluindo GPUs e TPUs». «Sim. O Colab é gratuito»; os recursos «não são garantidos nem ilimitados e os limites de uso às vezes flutuam». Exige conta Google; a interface logada não foi capturada.

| ITEM (FAQ, 09/09/2026) | O QUE DIZ |
|---|---|
| **Custo** | Gratuito; Colab Pro, Pro+ e Pay As You Go dão mais disponibilidade, por unidades de computação (o m não cita preços) |
| **Limites** | Não garantidos nem ilimitados; flutuam; o serviço prioriza quem está programando interativamente |
| **Execução longa** | O Pro+ permite execução contínua de até 24 h |
| **Conta** | Conta Google obrigatória; o notebook e os arquivos ficam no Google Drive |
| **O que roda** | O mesmo notebook `.ipynb` da Figura 12, num Jupyter hospedado pela Google |

**REGRA**

O Colab é o caminho mais curto para quem não pode instalar nada no computador (máquina da instituição, tablet). Duas restrições: dado com informação pessoal não sobe para a nuvem sem anonimizar; e o notebook precisa ser baixado e guardado no projeto (e versionado) — a cópia no Drive não é o registro da pesquisa.

SEM NÚMEROS DE LIMITE

> A FAQ não publica horas, memória ou GPU garantidas para o plano gratuito, e este manual não inventa: «flutuam» é a palavra da Google.

# `limpar_dados.py`: tipos, ausentes, duplicatas, texto inconsistente, vírgula decimal e valores fora de 1,5 × IQR — avisa, não apaga

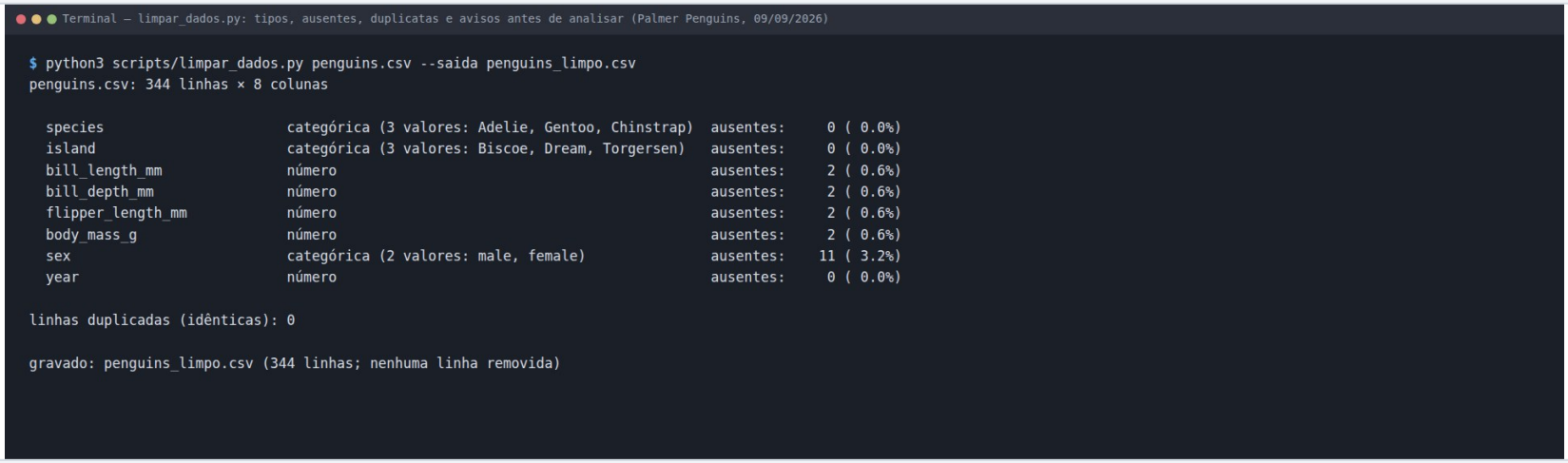



**FIGURA 15** `python3 limpar_dados.py penguins.csv` (captura de 09/09/2026, execução real nesta máquina): 344 × 8; tipo inferido por coluna; ausentes com contagem e percentual — 2 nas medidas, 11 em `sex`; 0 duplicatas; nenhuma linha foi apagada.

TERMINAL — USO

```
python3 scripts/limpar_dados.py dados.csv                    # só relatório
python3 scripts/limpar_dados.py dados.csv --sep ";" --saida limpo.csv   # CSV com ponto e vírgula; grava o arquivo limpo
python3 scripts/limpar_dados.py dados.xlsx --planilha 0      # primeira aba de uma planilha
```

| O QUE CONFERE | COMO APARECE |
|---|---|
| **Tipo por coluna** | |

Dois termos: a **mediana** é o valor do meio quando se ordena a coluna (metade acima, metade abaixo); o **IQR** (intervalo interquartil) é a distância entre o valor que deixa 25 % abaixo e o que deixa 75 % abaixo — a «caixa» do boxplot. Um valor além de 1,5 × IQR a partir da caixa é um candidato a **outlier**: um ponto a investigar, não a apagar.

> **REGRA**
>
> O script nunca apaga linhas. Cada aviso vira uma decisão sua, escrita no relatório de limpeza: o ausente fica ausente (não vira zero); a duplicata sai se for erro de digitação, fica se forem duas observações

| O QUE CONFERE | COMO APARECE |
| --- | --- |
| | Número, texto, data — e a coluna «numérica» que veio como texto por causa da vírgula |
| **Ausentes** | Contagem e % por coluna: célula vazia, «NA», «-» etc. |
| **Duplicatas** | Linhas inteiras repetidas |
| **Texto inconsistente** | Espaços sobrando e variantes de maiúsculas («Adelie» «adelie », «ADELIE») que viram três grupos |
| **Valores extremos** | Fora de 1,5 × IQR por coluna numérica — só avisa |

> reais; o valor extremo fica, salvo erro comprovado de medida — e, mesmo assim, o texto diz que saiu e por quê. A análise roda com e sem ele quando a dúvida persiste.

# `analisar.py` por grupo: descritivas, ANOVA com η² e Kruskal-Wallis — e a figura gerada pelo próprio script


**FIGURA 16** `python3 analisar.py penguins.csv --variavel body_mass_g --grupo species --figura massa_por_especie.png` (captura de 09/09/2026, execução real nesta máquina): descritivas por espécie e, por serem três grupos, ANOVA com η² e Kruskal-Wallis ao lado.

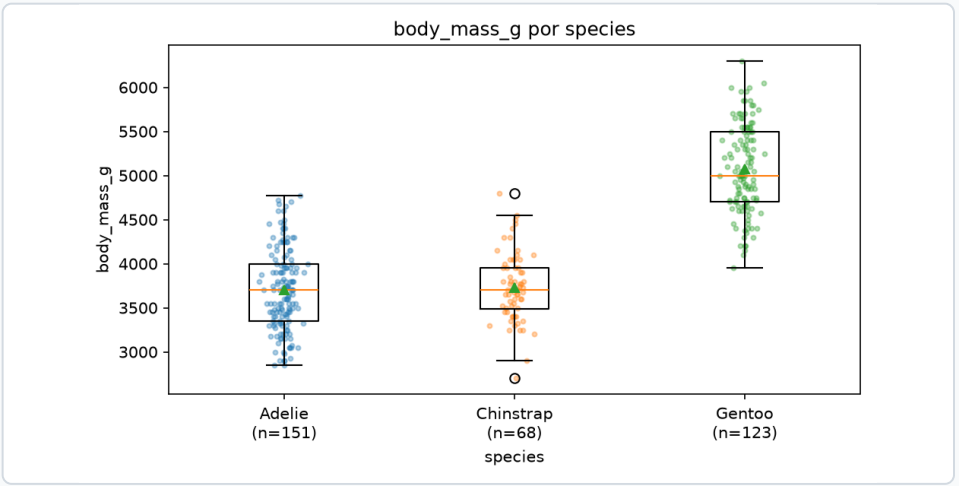



**FIGURA 17** `massa_por_especie.png`, gerada pelo `analisar.py --figura` (09/09/2026): boxplot com os pontos individuais, massa em gramas por espécie. Ninguém desenhou nada: se o CSV mudar, o mesmo comando refaz a figura.

| ESPÉCIE | N | MÉDIA (G) | D |
|---|---|---|---|
| **Adelie** | 151 | 3700,66 | 458,5 |
| **Chinstrap** | 68 | 3733,09 | 384,3 |
| **Gentoo** | 123 | 5076,02 | 504,1 |

ANOVA F(2, 339) = 343,63, p < 0,001, η² = 0,67 · Kruskal-Wallis H = 217,60

**COMO LER**

A **ANOVA** pergunta se as médias dos três grupos podem ser iguais; **F** é a razão entre a variação *entre* grupos e a variação *dentro* deles (os números entre parênteses são os graus de liberdade: 2 grupos a mais que um, 339 observações a menos que três). **p < 0,001** é a probabilidade de um F tão grande se as médias fossem iguais. **$\eta^2$ = 0,67** é o tamanho de efeito: 67 % da variação da massa é explicada pela espécie. O **Kruskal-Wallis** faz a mesma pergunta com as

> posições (postos) em vez das médias, sem supor distribuição normal; H = 217,60 aponta na mesma direção. O n é 342 porque 2 pinguins não têm massa.

# `analisar.py` para dois grupos, duas medidas e duas categorias: t de Welch com IC e d, Pearson com IC, qui-quadrado com V de Cramér

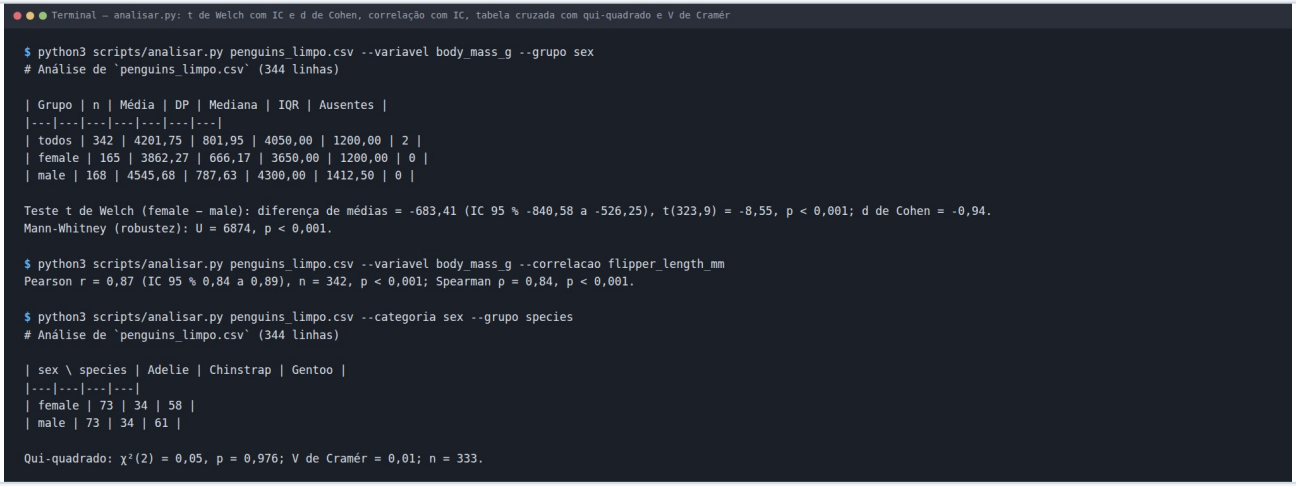



**FIGURA 18** Três chamadas do `analisar.py` (captura de 09/09/2026, execução real nesta máquina): massa por sexo (dois grupos → t de Welch, IC 95 %, d de Cohen, Mann-Whitney); massa × comprimento da nadadeira (Pearson com IC, Spearman); sexo × espécie (tabela cruzada, qui-quadrado, V de Cramér, aviso se algum esperado < 5).

TERMINAL — OS TRÊS COMANDOS DA FIGURA

```
python3 scripts/analisar.py penguins.csv --variavel body_mass_g
--grupo sex
python3 scripts/analisar.py penguins.csv --variavel body_mass_g
--correlacao flipper_length_mm
python3 scripts/analisar.py penguins.csv --categoria sex --grupo
species
# --markdown resultado.md grava a saída em Markdown para colar
no texto
```

COMO LER

**Diferença**: as fêmeas pesam, em média, 683,41 g a menos. O **IC 95 %** é a faixa compatível com os dados (de −840,58 a −526,25 g): não inclui o zero, então a diferença é distinguível de zero. O **t de Welch** compara as duas médias sem supor variâncias iguais; **d de Cohen** = **−0,94** é a diferença dividida pelo DP combinado — quase um desvio padrão, um efeito grande. O **Mann-Whitney** repete a comparação por postos. **r** = **0,87** é a correlação de Pearson (−1 a 1): massa e nadadeira crescem juntas, forte; **ρ de Spearman** faz o mesmo por

| PERGUNTA | RESULTADO (09/09/2026) |
|---|---|
| **Fêmeas × machos** (massa) | Diferença −683,41 g (IC 95 % −840,58 a −526,25); t d Welch t(323,9) = −8,55; d = −0,94; Mann-Whitney U = 6874 |
| **Massa × nadadeira** | Pearson r = 0,87 (IC 0,84 a 0,89); Spearman ρ = 0,84; = 342 |
| **Sexo × espécie** | $\chi^2(2) = 0,05$, p = 0,976; V de Cramér = 0,01; n = 333 |

> postos. No **qui-quadrado**, p = 0,976 e **V de Cramér** = **0,01** (0 a 1) dizem que a proporção de fêmeas e machos é praticamente a mesma nas três espécies. Os n mudam (342, 333) por causa dos ausentes de cada análise.

# Como ler os números: o tamanho de efeito diz quanto, o IC diz com que precisão, o p diz se é distinguível do acaso

| NÚMERO | O QUE É | COMO LER | NA DEMONSTRAÇÃO |
|---|---|---|---|
| **n** | Quantas observações entraram no cálculo | Por grupo, sempre; os ausentes à parte | 151 · 68 · 123 por espécie; 342 na correlação; 333 no qui-quadrado |
| **Média e DP** | Centro e dispersão, na unidade da variável | Sempre juntos; DP pequeno = grupo homogêneo | Gentoo 5076,02 g, DP 504,12 |
| **Mediana e IQR** | Centro e dispersão por posição; resistentes a extremos | Preferir quando a distribuição é assimétrica | Impressos pelo `analisar.py` ao lado da média |
| **Diferença / efeito** | O resultado principal, na unidade da variável | É o que responde à pergunta | −683,41 g entre fêmeas e machos |
| **IC 95 %** | Faixa de valores compatível com os dados | Largo = pouca precisão; se cruza o zero, a diferença não é distinguível de zero | −840,58 a −526,25 g |
| **d de Cohen** | Diferença ÷ DP combinado (sem unidade) | Compara efeitos entre estudos; pequeno, médio e grande são convenções, não leis | d = −0,94 |
| **$\eta^2$** | Fração da variação explicada pelos grupos (0 a 1) | O tamanho de efeito da ANOVA | $\eta^2$ = 0,67 |
| **r · ρ** | Força e sentido da relação entre duas medidas (−1 a 1) | Não diz que uma causa a outra | r = 0,87 (IC 0,84 a 0,89); ρ = 0,84 |

| NÚMERO | O QUE É | COMO LER | NA DEMONSTRAÇÃO |
|---|---|---|---|
| **V de Cramér** | Força da associação entre duas categorias (0 a 1) | O tamanho de efeito do qui-quadrado | V = 0,01 |
| **p** | Probabilidade de um resultado tão extremo se não houvesse efeito | Exato (p = 0,976), não «ns»; «< 0,001» só quando for menor; nunca sozinho | ANOVA p < 0,001; qui-quadrado p = 0,976 |

**REGRA**

A frase do resultado tem quatro partes, nesta ordem: o efeito na unidade da variável, o IC 95 %, o tamanho de efeito padronizado e o p exato — mais o n de cada grupo. Um p «significativo» com efeito minúsculo (V = 0,01 numa amostra grande) não vale um parágrafo; um efeito grande com IC largo pede mais dados, não mais adjetivos. É o que Lang e Altman (2015) pedem no SAMPL (Parte 6).

# Pressupostos e testes robustos: o paramétrico e o seu par por postos rodam juntos — e o texto diz o que foi conferido

| PARAMÉTRICO | SUPÕE | PAR ROBUSTO | NA DEMONSTRAÇÃ |
|---|---|---|---|
| **t de Welch** | Distribuição aproximadamente normal em cada grupo; não supõe variâncias iguais (por isso é o padrão do script, e não o t de Student) | **Mann-Whitney** | t(323,9) = −8,55 e U = 6874 apontar a mesma diferença |
| **ANOVA** | Normalidade dentro dos grupos e variâncias parecidas | **Kruskal-Wallis** | F = 343,63 e H = 217,60: mesma conclusão |
| **Pearson** | Relação linear; sensível a pontos extremos | **Spearman** | r = 0,87 e ρ = 0,84: a relação é monotônica quase linear |
| **Qui-quadrado** | Contagens esperadas ≥ 5 em cada célula | (o script avisa se alguma for < 5) | $\chi^2(2) = 0{,}05$; V = 0,01; n = 33 |

Um teste **robusto** (por postos, ou não paramétrico) troca os valores pelas suas posições na ordenação e por isso não depende da forma da distribuição nem se deixa puxar por um valor extremo. O `analisar.py` imprime sempre os dois: quando concordam, como aqui, o resultado não depende do pressuposto; quando divergem, é a distribuição (ou um outlier) que está falando — e o texto precisa dizer isso.

**REGRA**

Conferir pressuposto é olhar o dado, não coletar mais um p: o boxplot com pontos (Figura 17) mostra assimetria, caudas e valores extremos melhor que um teste de normalidade em amostras grandes. No método: «a normalidade foi avaliada graficamente; o teste robusto correspondente foi relatado ao lado». Se os dois discordam, relate os dois e explique.

**POR QUE WELCH E NÃO STUDENT**

O t de Student exige variâncias iguais nos dois grupos; o de Welch dispensa essa suposição e dá o mesmo resultado quando ela vale — os graus de liberdade fracionários (323,9) são a marca dele.

# Múltiplas comparações: cada teste a mais é uma chance a mais de «achar» algo — corrija e declare

Um p < 0,05 significa «isso aconteceria 1 vez em 20 sem efeito nenhum». Quem roda 20 testes espera, portanto, um p «significativo» por acaso. A ANOVA da demonstração diz que as espécies diferem; perguntar *quais pares* diferem (Adelie × Chinstrap, Adelie × Gentoo, Chinstrap × Gentoo) são três testes a mais — e é aí que entra a correção.

1
**Conte**
Quantas comparações o projeto previu; quantas você fez de fato

2
**Corrija**
Um método de correção para o conjunto (o mais simples divide o limiar de p pelo número de testes; os programas de menu e o SciPy oferecem outros)

3
**Declare**
No método: quantos testes, qual correção; no resultado: o p corrigido

4
**Separe**
O que era previsto (confirmatório) do que surgiu olhando o dado (exploratório) — o segundo gera hipótese, não conclusão

| SITUAÇÃO | O QUE FAZER |
|---|---|
| **ANOVA significativa, três grupos** | Comparações par a par com correção; o `analisar.py` para na ANOVA — o par é o passo seguinte, no programa de me no notebook |
| **Dez variáveis testadas contra o grupo** | Dez testes: corrija; ou, melhor, volte ao projeto — qual era a variável principal? |
| **Subgrupos criados depois de ver o dado** | Exploratório: relate como tal, sem p «significativo» como conclusão |
| **Um único teste previsto no projeto** | Nada a corrigir — e essa é a vantagem d decidir antes |

ARMADILHA

**Relatar só o que deu.** Rodar dez comparações e escrever sobre a única com p < 0,05, sem dizer que houve outras nove, é o erro que mais invalida resultado publicado. O registro de tudo o que foi testado fica no notebook ou no `--markdown` do script e vai para o material suplementar.

# Figura por script, nunca à mão: o arquivo que gera a figura lê o dado — e a refaz quando o dado muda

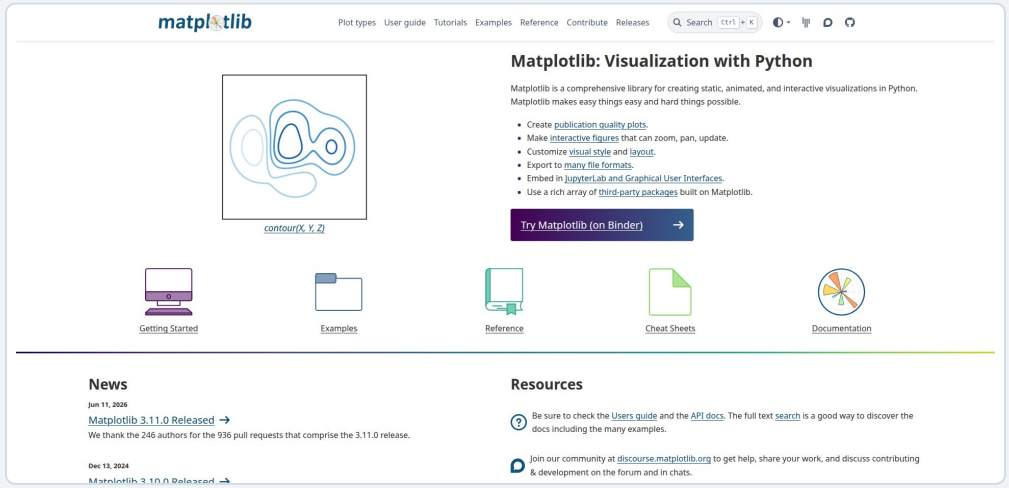



**FIGURA 19** matplotlib.org (captura de 09/09/2026): «biblioteca abrangente para criar visualizações estáticas, animadas e interativas em Python». É o que desenha a Figura 17 por dentro do `analisar.py` — e o que o seaborn usa no notebook (Figura 13).

TERMINAL — A FIGURA DA DEMONSTRAÇÃO E A SUA

```
python3 scripts/analisar.py penguins.csv --variavel
body_mass_g --grupo species --figura massa_por_especie.png
python3 scripts/analisar.py limpo.csv --variavel minha_medida
--grupo meu_grupo --figura figuras/fig1.png
```

| TODA FIGURA PRECISA DE | POR QUÊ |
|---|---|
| **Eixos com nome e unidade** | «Massa corporal (g)», não «body_mass_g»; sem unidade o leitor não sabe o que está vendo |
| **n por grupo** | Na figura ou na legenda: uma caixa de 68 e uma 151 não pesam igual |
| **Os pontos, não só o resumo** | O boxplot com pontos mostra a distribuição; a b com a média esconde |
| **Um arquivo de script que a gera** | Wilson et al. (2017): cada figura sai de um script lê o dado; se o dado muda, o comando refaz |
| **Um nome de arquivo previsível** | O `--figura` recebe o nome do arquivo de saí demonstração, `massa_por_especie.png`); n como `fig1_massa_por_especie.png` dizem a figura é e de onde veio |

REGRA

> Nenhuma figura do texto vem de um gráfico de planilha copiado nem de um editor de imagem: o nome do script e do arquivo de dados que a geraram ficam registrados (no notebook, no README ou na legenda do material suplementar). Se alguém pedir «a figura com os dados de 2025», a resposta é um comando, não uma tarde de trabalho.

PARTE 4 PROGRAMAS DE MENU

# jamovi e JASP — quando usar; R e RStudio em um slide; OpenRefine e Orange para limpar e explorar; o mesmo resultado em duas ferramentas

Seis slides para quem prefere clicar a digitar: programas gratuitos que fazem os mesmos testes da Parte 3 por menu, com tabelas prontas para o texto. As páginas oficiais aparecem em capturas de 09/09/2026; jamovi e JASP não estão instalados nesta máquina, por isso não há capturas das janelas — só o que a documentação afirma.

04

# jamovi: «planilha estatística gratuita e aberta», desenvolvida com R — descritivas, t, ANOVA, correlação e regressão por menu

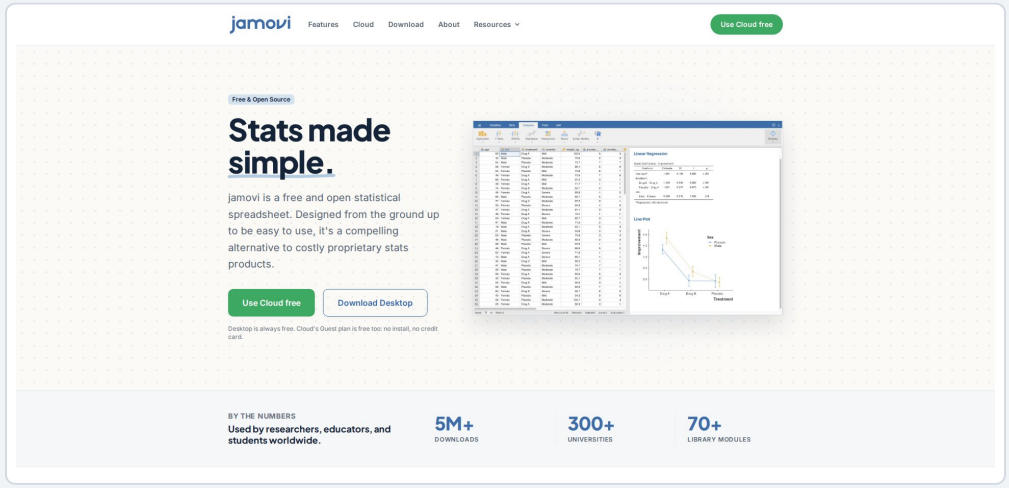



**FIGURA 20** jamovi.org (captura de 09/09/2026): «Stats made simple»; botões «Use Cloud free» e «Download Desktop». «Desenvolvido com R. Gratuito para sempre»; «Desktop é sempre gratuito». Versão mais recente publicada no GitHub: v2.7.30 (23/05/2026).

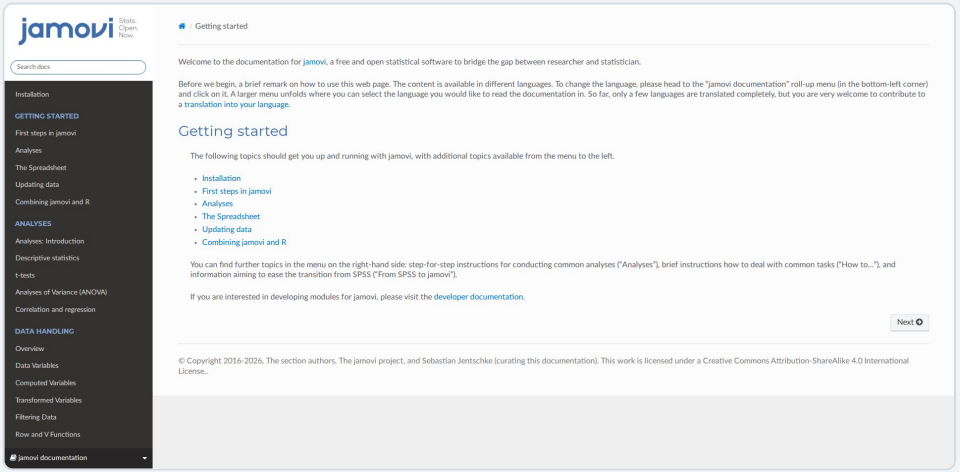



**FIGURA 21** docs.jamovi.org (captura de 09/09/2026): o guia do usuário — descritivas, testes t, ANOVA, correlação e regressão, gerenciamento de dados e um guia para quem vem do SPSS; licença CC BY-SA 4.0.

| ITEM (JAMOVI.ORG, 09/09/2026) | O QUE DIZ |
|---|---|
| **Custo** | Desktop sempre gratuito; o plano Guest do Cloud também: «sem instalação, sem cartão de crédito» |
| **Por dentro** | Desenvolvido com R: o resultado é o mesmo que o R daria, sem escrever código |
| **Documentação** | |

**QUANDO USAR**

Quando a técnica é uma das do slide «A técnica vem do projeto» e você não quer programar: o jamovi lê o CSV arrumado, mostra a tabela de dados de um lado e a saída do outro, e atualiza o resultado quando se muda uma opção. Peça sempre o tamanho de efeito e o IC nas opções da análise — eles não vêm marcados por padrão em todo programa de menu, e sem eles o relato não cumpre o SAMPL.

| ITEM (JAMOVI.ORG, 09/09/2026) | O QUE DIZ |
|---|---|
| | docs.jamovi.org: descritivas, t, ANOVA, correlação/ regressão; gerenciamento de dados; guia para quem vem do SPSS (CC BY-SA 4.0) |
| **Versão** | v2.7.30 no GitHub (23/05/2026) |

> **SEM CAPTURA DA JANELA**
>
> O jamovi Desktop não foi instalado nesta máquina; os nomes de menu internos não são citados porque não foram conferidos — a documentação da Figura 21 os traz.

# jamovi Cloud no navegador (sem instalar) e JASP: «A Fresh Way to Do Statistics», da Universidade de Amsterdã

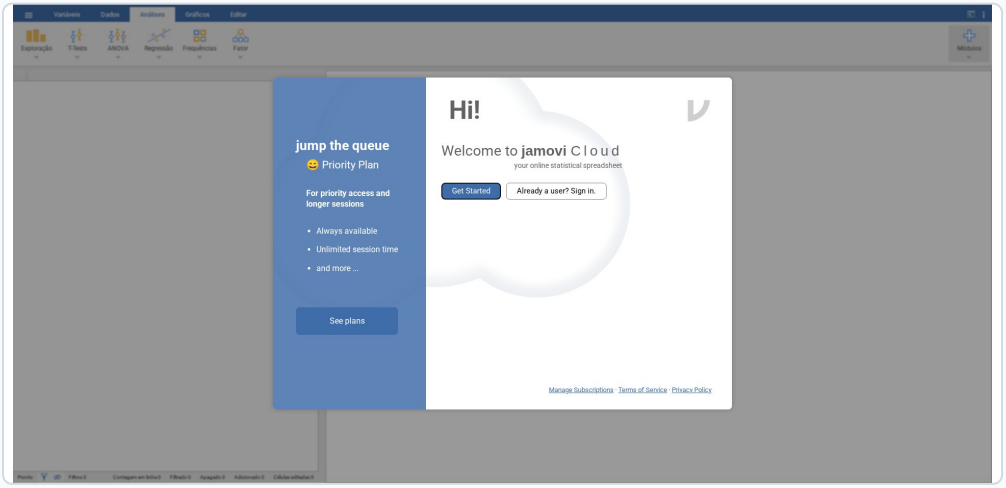



**FIGURA 22** cloud.jamovi.org (captura de 09/09/2026): «Welcome to jamovi Cloud», com «Get Started» e «Already a user? Sign in». É só a porta de entrada: o manual não criou conta nem usou o serviço; o plano Guest é gratuito segundo jamovi.org.


**FIGURA 23** jasp-stats.org (captura de 09/09/2026): «A Fresh Way to Do Statistics» — gratuito e de código aberto, desenvolvido na Universidade de Amsterdã (Departamento de Métodos Psicológicos, Eric-Jan Wagenmakers); análises clássicas e bayesianas; integração com o OSF; tabelas em formato APA.

| FERRAMENTA | VANTAGEM | O QUE SABER |
|---|---|---|
| **jamovi Cloud (Guest)** | Nada a instalar; abre no navegador | O arquivo de dados sobe para um servidor de terceiros: só dado anonimizado; para trabalho contínuo, o Desktop |
| **JASP** | Tabelas já em formato APA; análise bayesiana | Instalação local (slide seguinte); não precisa de internet para analisar |

**REGRA**

Os dois fazem o mesmo conjunto de testes deste manual; a escolha é de gosto e de ecossistema — jamovi para quem quer a planilha e a saída lado a lado com R por baixo, JASP para quem quer a tabela APA pronta e a versão bayesiana do mesmo teste. Em qualquer um, o resultado precisa bater com uma segunda ferramenta antes de ir para o texto (slide «Reproduzir o mesmo resultado»).

| FERRAMENTA | VANTAGEM | O QUE SABER |
| --- | --- | --- |
| | ao lado da clássica;<br>integração com o OSF | |

# JASP 0.98.1: Windows, macOS, Linux e ChromeOS; ~4 GB de disco, 4 GB de RAM; licença GNU AGPL v3; sem internet


**FIGURA 24** jasp-stats.org/download (captura de 09/09/2026): JASP 0.98.1 (07/07/2026; tag v0.98.1 no GitHub) — Windows (MSIX, MSI, Microsoft Store, WinGet), macOS (Apple Silicon e Intel), Linux (Flatpak/Flathub) e ChromeOS. A home ainda mostra «0.19.3» em alguns pontos; vale a página de download.

| REQUISITO (DOWNLOAD, 09/09/2026) | VALOR |
|---|---|
| **Versão** | 0.98.1 (07/07/2026) |
| **Plataformas** | Windows (MSIX/MSI/Store/WinGet); macOS (Appl Silicon e Intel); Linux (Flatpak/Flathub); ChromeO |
| **Disco · memória** | ~4 GB de disco; 4 GB de RAM no mínimo, 8 GB recomendados |
| **Licença** | GNU AGPL v3 — código aberto |
| **Internet** | Não é necessária para analisar |

### JAMOVI, QUANDO

- Você quer ver a tabela e o resultado juntos e mexer nas opções
- Vem do SPSS (há guia na documentação)
- Precisa rodar no navegador sem instalar (Cloud Guest)

### JASP, QUANDO

- Quer a tabela em formato APA pronta para o texto
- Quer a versão bayesiana do teste ao lado da clássica
- Já usa o OSF para guardar o projeto

NÃO CAPTURADO

> Nenhum dos dois programas foi instalado nesta máquina; os requisitos e as plataformas vêm da página de download, não de uma instalação real.

# R e RStudio em um slide: o «ambiente de software livre para computação estatística e gráficos» que está por baixo do jamovi

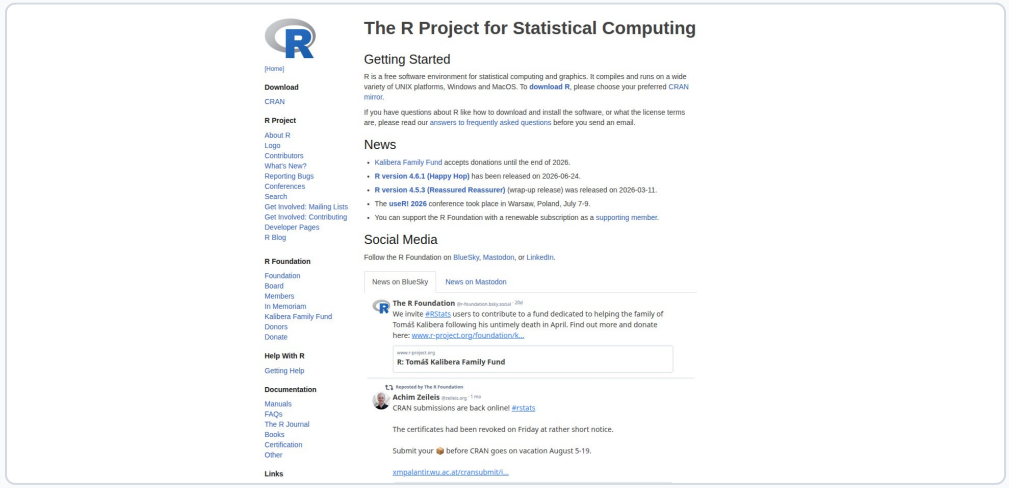



**FIGURA 25** r-project.org (captura de 09/09/2026): R 4.6.1 «Happy Hop» (24/06/2026), «ambiente de software livre para computação estatística e gráficos», para UNIX, Windows e macOS.

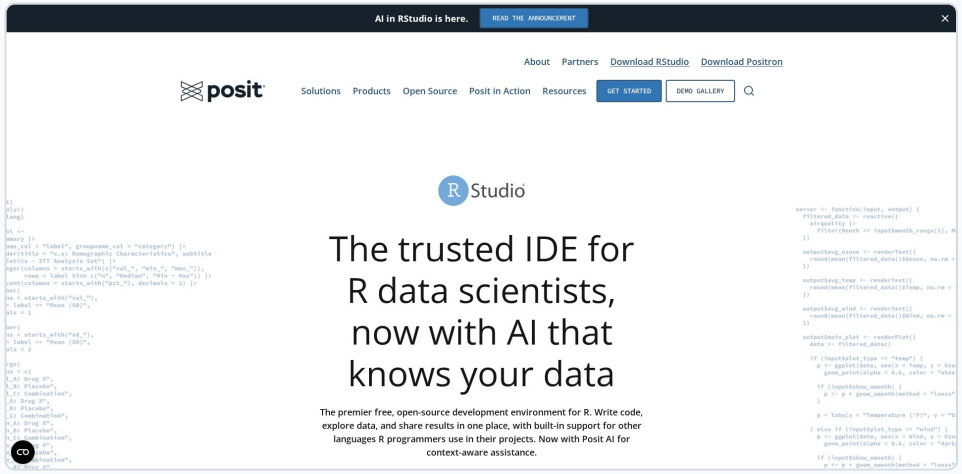



**FIGURA 26** posit.co (captura de 09/09/2026): o RStudio é «o IDE gratuito e de código aberto de referência para R» — editor, console, gráficos e arquivos numa janela. A Posit anuncia também um novo IDE com R e Python, não detalhado aqui.

| VÁ PARA O R QUANDO | POR QUÊ |
|---|---|
| **O orientador ou o grupo já usa R** | O script que eles leem é o que você entrega; a Parte 3 em Python tem equivalente linha a linha |
| **O modelo vai além dos testes deste manual** | Modelos mistos, séries, análises multivariadas têm pacotes consolidados no R |
| **Você quer ver o que o jamovi fez** | |

**REGRA**

R e Python fazem tudo o que este manual pede; não há razão para aprender os dois de uma vez. A regra é a mesma nos dois: o dado bruto intocado, um script para cada resultado, a figura gerada por código, o notebook (Jupyter roda R também) executado de cima a baixo. Este manual não traz comandos de R porque nada em R foi executado na demonstração — e não inventa saída.

| VÁ PARA O R QUANDO | POR QUÊ |
| --- | --- |
| | O jamovi é desenvolvido com R: o teste do menu corresponde a uma função do R |

VERSÕES

R 4.6.1 e a página do RStudio conferidas em 09/09/2026; a versão do RStudio não foi conferida nesta data e não é afirmada.

# OpenRefine para limpar «dados bagunçados» com histórico reproduzível; Orange para explorar visualmente antes de testar

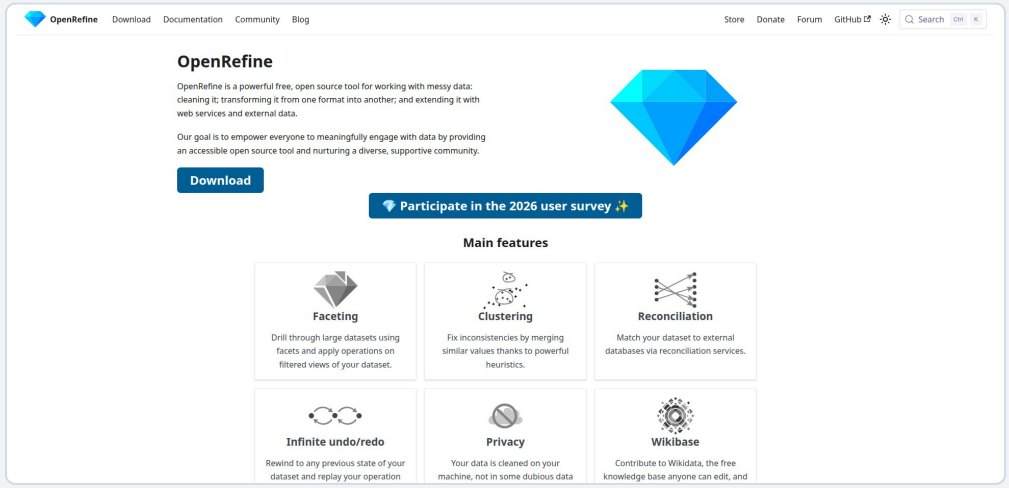



**FIGURA 27** openrefine.org (captura de 09/09/2026): «ferramenta gratuita e de código aberto para trabalhar com dados bagunçados: limpar, transformar de um formato em outro»; versão 3.10.1 (04/03/2026). O histórico de operações é reproduzível — a limpeza vira um registro, como pede a Parte 1.

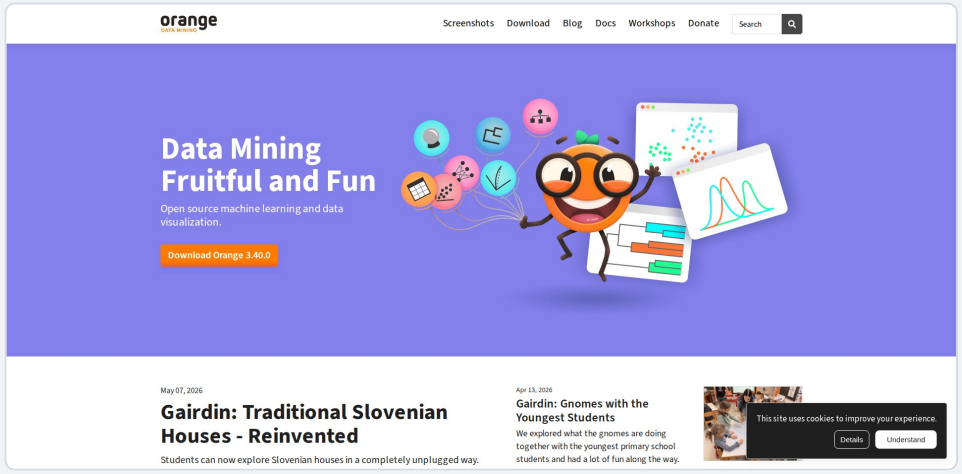



**FIGURA 28** orangedatamining.com (captura de 09/09/2026): Orange 3.40.0, mineração de dados visual — «Fruitful and Fun». Serve para explorar o dado sem escrever código, antes de aplicar a técnica do projeto.

| FERRAMENTA | SERVE PARA | NÃO SERVE PARA |
|---|---|---|
| **OpenRefine** 3.10.1 | «Dados bagunçados»: limpar texto inconsistente («Adelie», «adelie », «ADELIE»), transformar | Testes estatísticos; é a etapa antes do `limpar_dados.py`, quando o texto está muito sujo |

**REGRA**

Limpar no OpenRefine é aceitável porque o histórico de operações é reproduzível: guarde-o junto do relatório de limpeza e cite-o no método. Explorar no Orange é aceitável porque não decide nada: o que você vir ali vira uma pergunta a conferir com a técnica do projeto — não um resultado.

| FERRAMENTA | SERVE PARA | NÃO SERVE PARA |
| --- | --- | --- |
| | de um formato em outro, com histórico de operações reproduzível | |
| **Orange** 3.40.0 | Mineração de dados visual: olhar distribuições e relações sem escrever código | O relato: a figura do texto continua vindo de um script (Parte 3) |

PROGRAMAS DE MENU 6 DE 6 · DEMONSTRAÇÃO DE 09/09/2026

# Reproduzir o mesmo resultado em duas ferramentas: se a média bate e o DP não, o problema é o n — e você só descobre comparando

1

### Rode na primeira

O `analisar.py`, o notebook ou o programa de menu — com o mesmo arquivo limpo

2

### Repita na segunda

Planilha, jamovi, JASP ou o script: a mesma variável, o mesmo grupo, o mesmo teste

3

### Compare nesta ordem

n por grupo → ausentes → média → DP → estatística do teste → efeito → IC → p

4

### Divergiu?

Não é «arredondamento»: é n, ausente, variante do teste (Student × Welch), opção marcada — investigue e registre

| ESPÉCIE | PYTHON (N · MÉDIA · DP) | CALC (LINHAS · MÉDIA · DP) |
|---|---|---|
| **Adelie** | 151 · 3700,66 · 458,57 | 152 · 3700,66 · 457,05 |
| **Chinstrap** | 68 · 3733,09 · 384,34 | 68 · 3733,09 · 384,34 |
| **Gentoo** | 123 · 5076,02 · 504,12 | 124 · 5076,02 · 502,06 |

Na demonstração, a planilha e o Python concordaram na média e discordaram no DP de duas espécies. A causa (slide «COUNTIF, AVERAGEIF e o n que muda o DP») foi uma linha com massa ausente contada no n do denominador da planilha. Sem a segunda ferramenta, o DP da planilha iria para o texto — e ninguém perceberia.

**REGRA**

Todo número que vai para o texto foi obtido em duas ferramentas independentes, ou pela mesma ferramenta em dois caminhos (script e notebook). O registro da conferência («média e DP por espécie conferidos no Calc em 09/09/2026; diferença no DP explicada pelo n») fica no relatório de análise. Não é desconfiança do software: é desconfiança da *opção* que você marcou.

**O QUE COSTUMA DIVERGIR**

DP «da população» (÷ n) contra «da amostra» (÷ n − 1); t de Student contra Welch; correlação com ou sem as linhas incompletas; percentual sobre o total com ou sem ausentes. Em todos os casos, o n resolve.

PARTE 5 QUALITATIVO COM O TAGUETTE

# Análise temática e o codebook antes de codificar; instalar e abrir o Taguette; criar projeto e adicionar documentos; destacar e etiquetar; ver por etiqueta e exportar; `codificar_taguette.py`; relatar com COREQ e SRQR

Oito slides com um projeto real no Taguette 1.5.2 local: três documentos, cinco etiquetas, doze destaques, exportação em CSV e a síntese por script. Os textos das «entrevistas» são **fictícios e ilustrativos**, escritos para a demonstração — nenhuma pessoa real foi entrevistada.

05

QUALITATIVO 1 DE 8 · BRAUN E CLARKE (2006), «USING THEMATIC ANALYSIS IN PSYCHOLOGY»

# Análise temática em seis fases (Braun e Clarke, 2006) — e o codebook escrito antes de destacar o primeiro trecho

1

**Familiarização**

Ler todas as transcrições, mais de uma vez, anotando impressões — sem etiquetar ainda

2

**Códigos iniciais**

Etiquetar trechos com códigos descritivos, sistematicamente, em todo o material

3

**Busca de temas**

Agrupar códigos em temas candidatos; reunir os trechos de cada um

4

**Revisão**

Conferir se cada tema se sustenta nos trechos e no conjunto; fundir, dividir, descartar

5

**Definição e nomeação**

Escrever o que cada tema é (e o que não é); nome curto

6

**Relato**

Cada tema com trechos que o ilustrem, ligados à pergunta; COREQ/SRQR (slide «Relatar»)

| ETIQUETA (DEMONSTRAÇÃO) | O QUE COBRE — ESCRITO ANTES DE CODIFICAR (EXEMPLO ILUSTRATIVO) |
|---|---|
| **previsibilidade** | Falas sobre antecipar falhas ou paradas e planejar com antecedência |
| **qualidade do dado** | Falas sobre sensores, registros e planilhas incompletos errados ou não confiáveis |
| **pessoas e treinamento** | Falas sobre quem usa a informação, resistência e capacitação |
| **custo** | Falas sobre gasto, investimento e retorno |
| **segurança** | Falas sobre risco a pessoas e equipamentos |

**REGRA**

O *codebook* é uma tabela — etiqueta, definição, exemplo, contraexemplo — escrita **antes** da codificação, a partir da pergunta de pesquisa e da primeira leitura, e versionada: toda etiqueta criada durante a codificação entra nele com data e motivo. Sem *codebook*, dois leitores etiquetam o mesmo trecho de jeitos diferentes e ninguém pode conferir. O Taguette exporta o *codebook* (slide «Ver por etiqueta e exportar»), mas quem o escreve é você.

**ARMADILHA**

**Tema = contagem.** «Custo apareceu 2 vezes, qualidade do dado 3» não é um resultado: um tema se sustenta pelo que os trechos dizem e por como respondem à pergunta, não pela frequência. A contagem serve para descrever a cobertura (em quantos documentos), não para ranquear temas.

# Instalar o Taguette: «ferramenta gratuita e de código aberto para análise qualitativa de dados» — instalador no macOS e Windows, `pip` no Linux

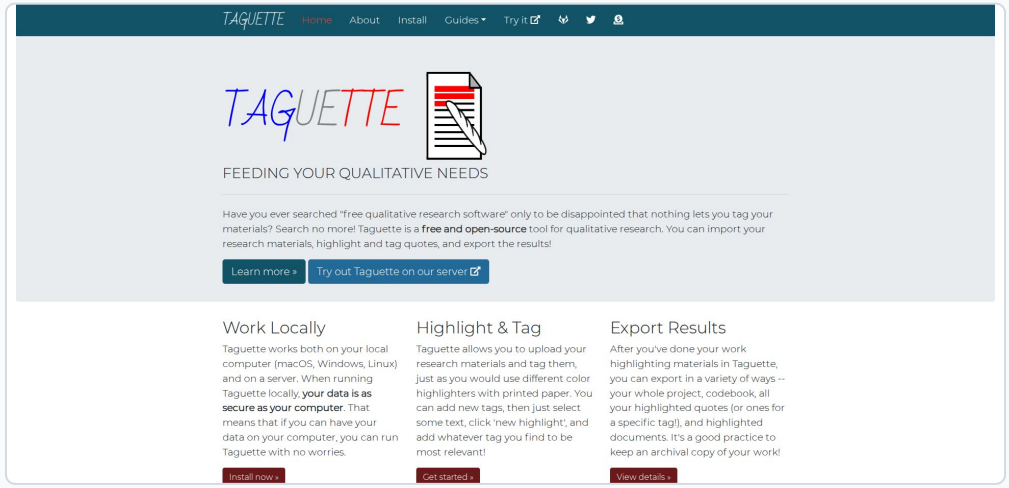



**FIGURA 29** taguette.org (captura de 09/09/2026): «Taguette é uma ferramenta gratuita e de código aberto para análise qualitativa de dados»; funciona localmente (macOS, Windows, Linux) e em servidor; licença BSD-3; há um servidor público gratuito em app.taguette.org, que exige conta e não foi usado.

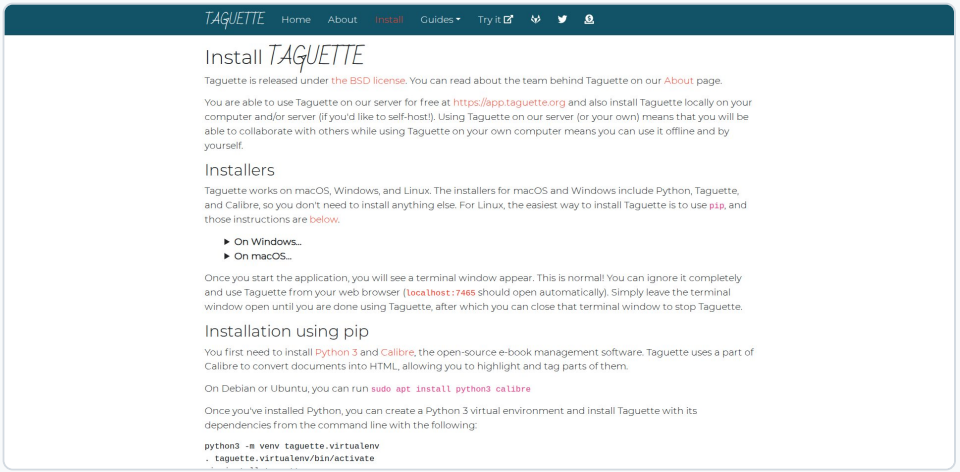



**FIGURA 30** taguette.org/install (captura de 09/09/2026): os instaladores para macOS e Windows incluem Python, Taguette e Calibre; no Linux, `pip install taguette`. Versão no PyPI: 1.5.2.

TERMINAL — LINUX (O QUE FOI FEITO NESTA MÁQUINA; NO MACOS E NO WINDOWS, USE O INSTALADOR)

```
pip install taguette
# nesta máquina, por causa de uma dependência antiga da versão
1.5.2:
uv tool install taguette --python 3.12 --with "setuptools<80"
```

| ITEM (TAGUETTE.ORG, 09/09/2026) | VALOR |
|---|---|
| **Custo · licença** | Gratuito; código aberto sob BSD-3 |
| **Onde roda** | No seu computador (macOS, Windows, Linux) ou num servidor; o servidor público app.taguette.org exige con |
| **Versão** | 1.5.2 (PyPI) — a que rodou nesta máquina |

```
taguette                        # abre o navegador em http://
localhost:7465/?token=…
```

| ITEM (TAGUETTE.ORG, 09/09/2026) | VALOR |
|---|---|
| **Documentos aceitos** | txt, docx, pdf, html e outros; os instaladores de macOS Windows trazem o Calibre junto |

**REGRA**

Transcrição de entrevista é dado pessoal até ser anonimizada: por isso o Taguette deste manual roda *localmente*, e o servidor público não é usado. O projeto do Taguette (banco local) fica na pasta protegida, fora do repositório; o que entra no Git é o *codebook*, o script e a síntese — e os destaques exportados só depois de conferir que nenhum trecho identifica alguém.

**ALTERNATIVAS**

NVivo, ATLAS.ti e MAXQDA (pagos) e QualCoder (livre) fazem a mesma codificação; não foram conferidos nesta data, por isso só os nomes. O Taguette exporta o *codebook* em QDC (REFI-QDA) para trocar com outros softwares.

# Abrir o Taguette: `localhost:7465` no navegador, «Single-user mode» — e «Welcome! Here are your projects»

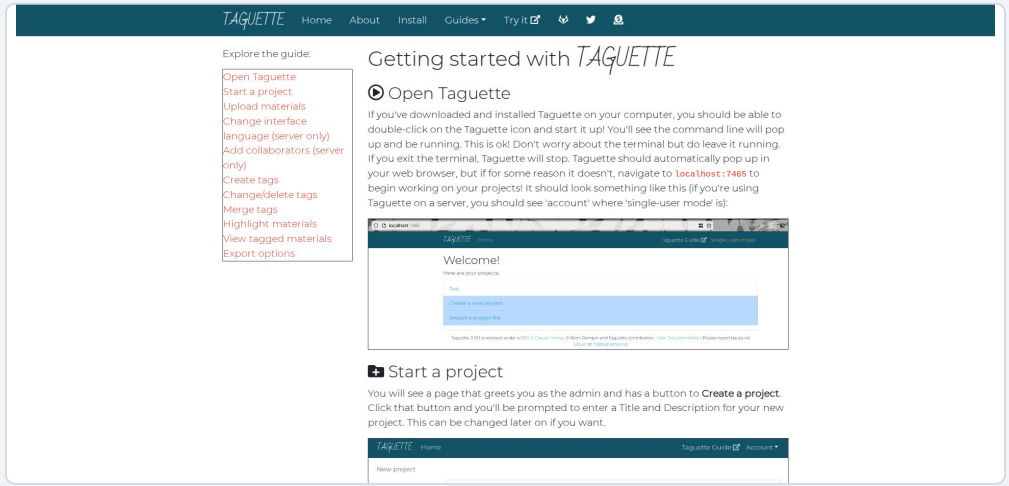



**FIGURA 31** «Getting started with Taguette» (captura de 09/09/2026): o guia oficial dos primeiros passos. Os slides seguintes seguem a mesma sequência com o projeto real desta máquina.

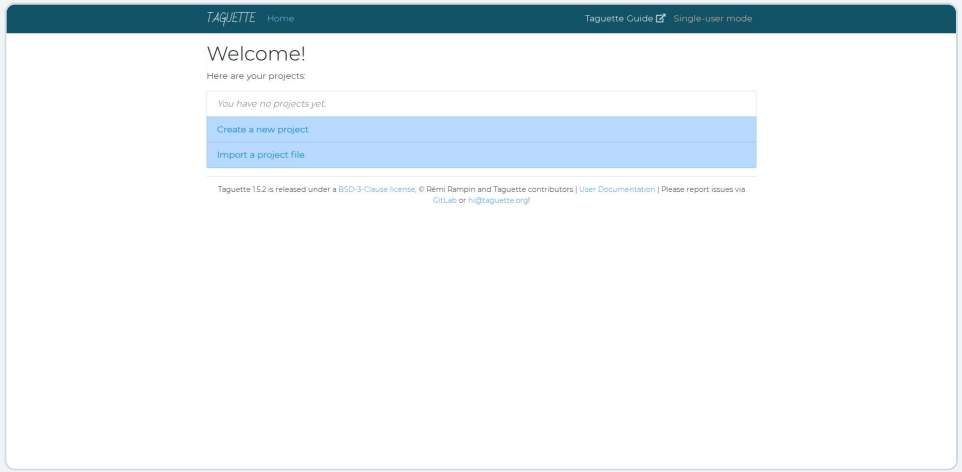



**FIGURA 32** A tela inicial do Taguette 1.5.2 nesta máquina (captura de 09/09/2026, execução real): «Welcome! Here are your projects», com «Create a new project» e «Import a project file». O endereço é `http://localhost:7465/?token=…` — o programa roda no seu computador e o navegador só o exibe.

| O QUE A TELA DIZ | O QUE SIGNIFICA |
|---|---|
| **localhost:7465** | Endereço da sua própria máquina; nada sai para a internet |
| **?token=…** | Chave de sessão gerada ao abrir, que faz parte do endereço mostrado |
| **Single-user mode** | Sem contas: o projeto é de quem está no computador |

**REGRA**

Um projeto por estudo, não por entrevista: os documentos ficam juntos para que a aba «Highlights» mostre uma etiqueta atravessando todos eles (slide «Ver por etiqueta»). Antes do primeiro documento, o *codebook* já está escrito e as etiquetas criadas com os mesmos nomes.

| O QUE A TELA DIZ | O QUE SIGNIFICA |
| --- | --- |
| **«Import a project file»** | Abre um projeto exportado antes (o seu de outr máquina, ou o de um colega) |

# Criar o projeto e adicionar documentos: «Create a new project» → Name, Description → «Create» → «Add a document»

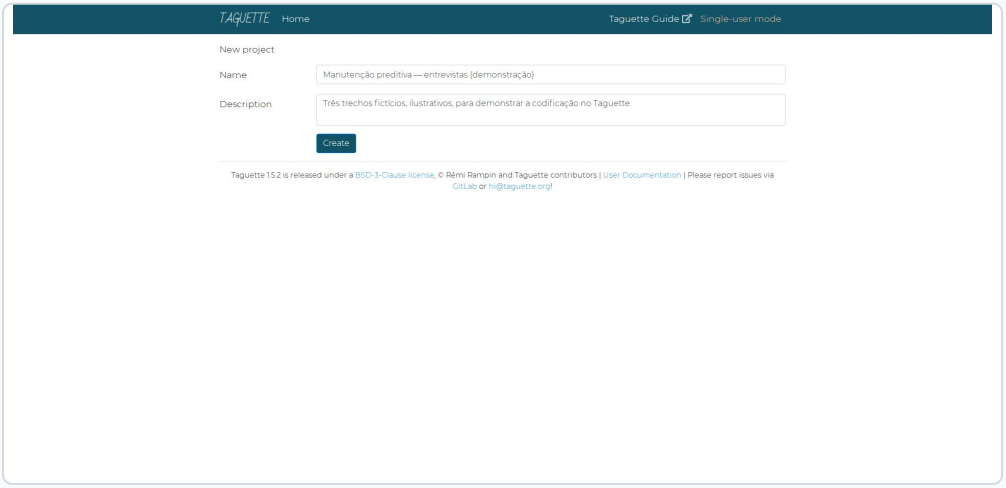



**FIGURA 33** O formulário «New project» preenchido (captura de 09/09/2026, execução real nesta máquina): «Name» e «Description» → «Create». A descrição é o lugar para registrar a pergunta de pesquisa e a versão do codebook.

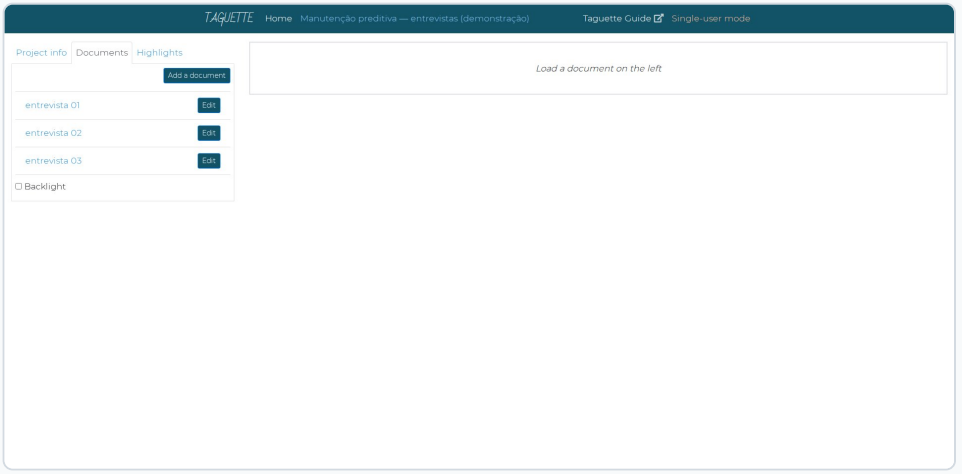



**FIGURA 34** O projeto com os três documentos (captura de 09/09/2026, execução real): abas «Project info», «Documents» e «Highlights»; «Add a document» (nome, descrição e o arquivo — txt, docx, pdf, html…); «Backlight» realça tudo o que já foi destacado. Os documentos são entrevistas **fictícias**: um técnico de manutenção, uma supervisora de operações e um engenheiro de confiabilidade, escritos para a demonstração.

1

**«Create a new project»**

Name, Description → «Create»

2

**«Add a document»**

Nome (ex.: `entrevista_01`), descrição (perfil, data, duração — sem nome de pessoa), arquivo

**REGRA**

O nome do documento é o identificador que vai aparecer na exportação e no texto («entrevista_02»): curto, sem nome de pessoa, com uma tabela à parte — fora do repositório — ligando o identificador ao perfil e à data. A descrição do documento guarda o

3

**Etiquetas**

Criar as do codebook antes de destacar; a etiqueta «interesting» («Further review required») vem com todo projeto

4

**Repetir por documento**

Todas as transcrições no mesmo projeto, anonimizadas antes de subir

contexto (função, tempo de casa, data da entrevista) que o COREQ pede no relato.

FORMATO

Transcrições em `.txt`, `.docx`, `.pdf` ou `.html` são aceitas no «Add a document». O texto aparece à direita, pronto para destacar (slide seguinte).

# Destacar e etiquetar: selecionar o trecho, escolher a etiqueta — o destaque fica ligado ao documento, à posição e à etiqueta

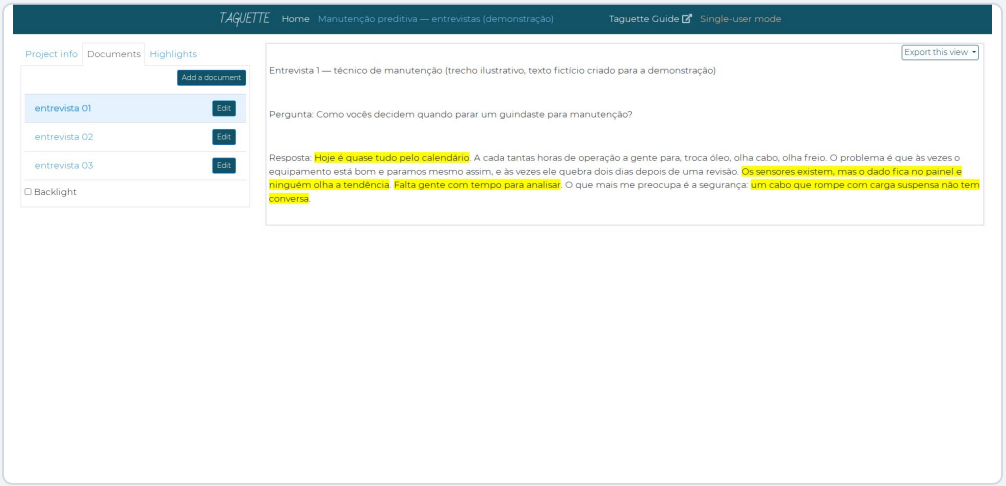



**FIGURA 35** A entrevista 01 (texto **fictício**) aberta no Taguette (captura de 09/09/2026, execução real nesta máquina): quatro trechos destacados em amarelo, cada um com a sua etiqueta; à esquerda, a lista de etiquetas do projeto. Na demonstração, os destaques foram criados pela mesma API que o botão da interface usa.

1

### Abrir o documento

Aba «Documents» → o texto aparece à direita

2

### Selecionar o trecho

Com o mouse, a frase ou o parágrafo que responde à pergunta — nem uma palavra, nem a página inteira

3

### Escolher a etiqueta

Uma ou mais etiquetas do codebook para o mesmo trecho (coocorrência, que o script conta)

4

### Etiqueta nova?

Só se o codebook for atualizado com definição e motivo; senão, «interesting» e decidir depois

**REGRA**

Um destaque é uma *decisão*: aquele trecho responde àquela definição do *codebook*. Trecho longo demais dilui o tema; curto demais perde o contexto que o leitor precisa para concordar com você. Codifique todo o material com o mesmo *codebook*; se ele mudar no meio, volte ao primeiro documento e recodifique — é a fase 4 de Braun e Clarke.

FICTÍCIO, SEMPRE

Os trechos que aparecem nesta parte foram escritos para a demonstração e não citam pessoa, empresa ou lugar real; ao reproduzi-los no texto, mantenha o rótulo.

# Ver por etiqueta e exportar: a aba «Highlights» atravessa os documentos; «Export this view» salva os destaques; o codebook sai em CSV, DOCX ou REFI-QDA

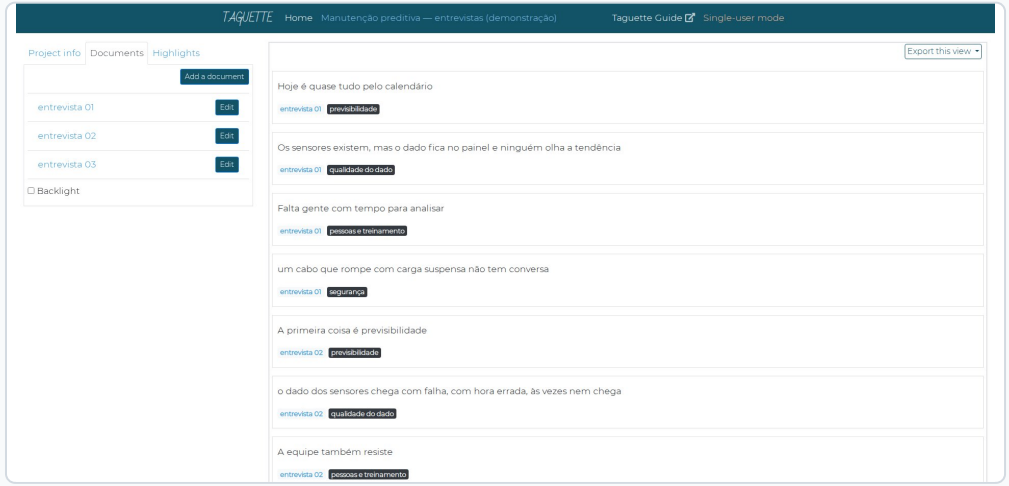



**FIGURA 36** A aba «Highlights» com todos os destaques (captura de 09/09/2026, execução real nesta máquina): cada trecho com o documento e a etiqueta. Doze destaques em três documentos, cinco etiquetas do codebook — a etiqueta padrão «interesting» ficou sem uso.

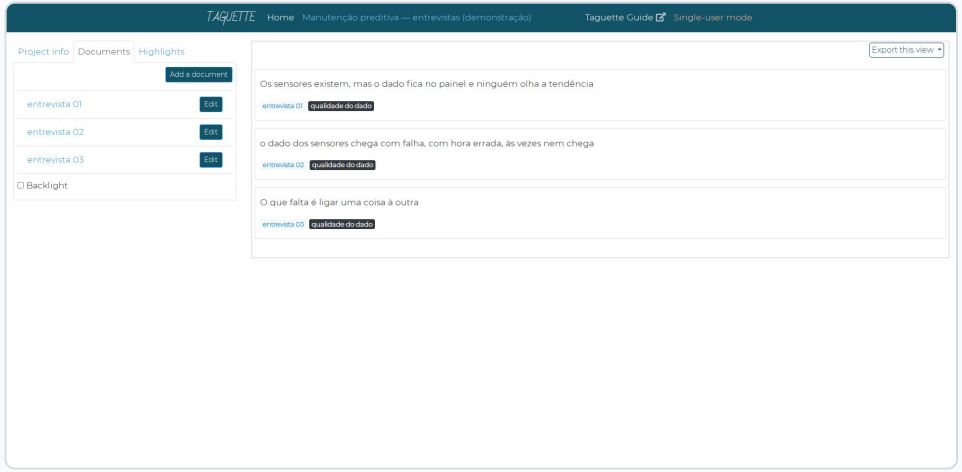



**FIGURA 37** Os destaques da etiqueta «qualidade do dado» (captura de 09/09/2026, execução real): três trechos, um por documento — é a leitura por tema que a fase 3 de Braun e Clarke pede. «Export this view» salva exatamente essa lista.

| EXPORTAÇÃO | FORMATOS | PARA QUÊ |
|---|---|---|
| **Destaques** («Export this view») | CSV, XLSX, DOCX, HTML… | O `destaques.csv` (id, document, tag, content) que o `codificar_taguette.py` lê; o DOCX para o orientador ler |
| **Codebook** | CSV, XLSX, DOCX, HTML, | O `codebook.csv` (tag, descriptior number of highlights, number of |

| ETIQUETA (CODEBOOK.CSV REAL) | DESTAQUES | DOCUME |
|---|---|---|
| qualidade do dado | 3 | |
| pessoas e treinamento | 3 | |
| previsibilidade | 2 | |

| EXPORTAÇÃO | FORMATOS | PARA QUÊ |
|---|---|---|
| | PDF e QDC (REFI-QDA) | documents); o QDC para abrir em outro software |
| **Projeto** | Arquivo de projeto | «Import a project file», na tela inicia o abre em outra máquina — a do segundo codificador, por exemplo |

| ETIQUETA (CODEBOOK.CSV REAL) | DESTAQUES | DOCUME |
|---|---|---|
| custo | 2 | |
| segurança | 2 | |
| interesting | 0 | |

**REGRA**

Exporte os destaques e o *codebook* ao fim de cada sessão, com a data no nome do arquivo: a exportação é o registro auditável da codificação. O QDC (REFI-QDA) é o formato de troca entre programas de análise qualitativa — é o que permite ao segundo codificador usar outro software.

QUALITATIVO 7 DE 8 · EXECUÇÃO REAL NESTA MÁQUINA, 09/09/2026 · PYTHON 3, PANDAS

# `codificar_taguette.py`: etiqueta × documento, trechos por etiqueta e coocorrência a partir do que o Taguette exportou

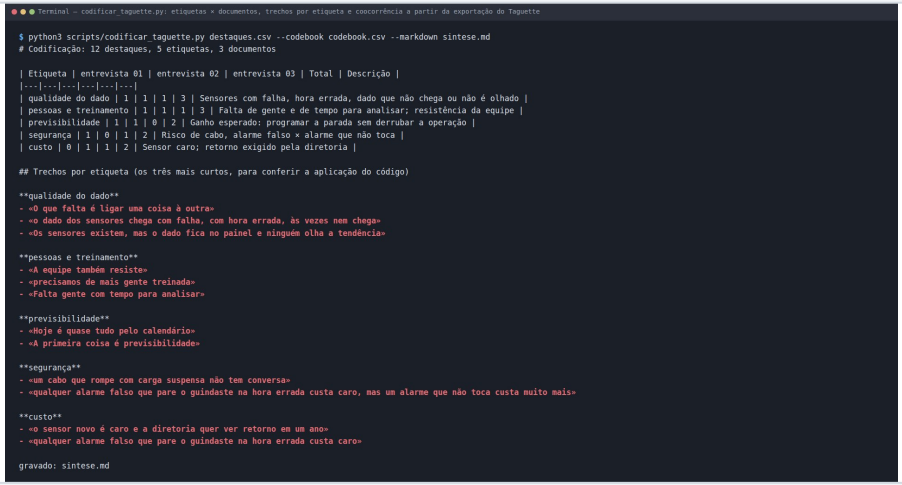



**FIGURA 38** `python3 codificar_taguette.py destaques.csv --codebook codebook.csv` (captura de 09/09/2026, execução real nesta máquina): a matriz etiqueta × documento, os totais, a descrição de cada etiqueta vinda do codebook, os três trechos mais curtos por etiqueta e a coocorrência (o mesmo trecho com duas etiquetas). Os trechos são **fictícios**.

TERMINAL — USO

```
python3 scripts/codificar_taguette.py destaques.csv --
codebook codebook.csv
python3 scripts/codificar_taguette.py destaques.csv --
codebook codebook.csv --markdown sintese.md   # grava a
síntese para o texto
```

| SAÍDA | PARA QUÊ |
|---|---|
| **Etiqueta × documento** | Cobertura: em quais entrevistas cada tema aparec (não «quantas vezes é verdade») |
| **Descrição** | A definição do codebook, ao lado dos números, pa não esquecer o que a etiqueta significa |
| **Três trechos mais curtos** | Candidatos a citação no texto — os curtos cabem r parágrafo; os longos vão para o suplemento |
| **Coocorrência** | Trechos com duas etiquetas: onde os temas se toc matéria-prima da fase 4 (revisão) |

REGRA

O script organiza; não interpreta. A síntese em Markdown é o ponto de partida do capítulo de resultados: para cada tema, a definição, em quantos documentos apareceu e os trechos — e então *você* escreve o

> que o tema responde à pergunta. Cada trecho citado no texto leva o identificador do documento («entrevista_02»), como o Taguette exporta.

# Relatar o qualitativo: COREQ (32 itens) ou SRQR, um segundo codificador numa amostra, e trechos no texto — com documento e etiqueta

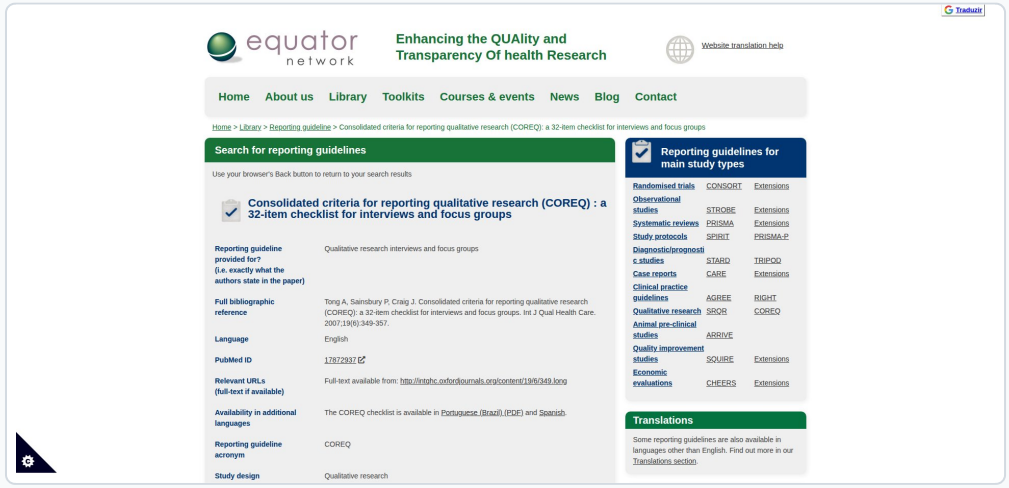



**FIGURA 39** O COREQ na EQUATOR Network (captura de 09/09/2026): Tong, Sainsbury e Craig (2007), «Consolidated criteria for reporting qualitative research», checklist de 32 itens para entrevistas e grupos focais — equipe e reflexividade, desenho do estudo, análise e achados; há tradução para o português na EQUATOR.

| O TEXTO PRECISA DIZER | ONDE ESTE MANUAL A |
|---|---|
| **Quem codificou, com que experiência e relação com os participantes** | Reflexividade (COREQ, domínio 1): escreva an codificar |
| **Como o codebook foi construído e mudou** | O codebook versionad (slide «Análise temátic |
| **Quantos codificadores; como as divergências foram resolvidas** | Segundo codificador nu amostra de documento com o mesmo codeboo divergências discutidas registradas |
| **Software usado** | Taguette 1.5.2; exporta em CSV; `codificar_taguett` |
| **Temas com trechos ilustrativos e identificador do participante** | Os trechos de `destaques.csv`, co documento («entrevista_02») — nu nome |
| **Achados ligados à pergunta; casos discordantes** | A coocorrência e os tre que não couberam em |

| O TEXTO PRECISA DIZER | ONDE ESTE MANUAL A |
| --- | --- |
| | nenhum tema entram r<br>relato, não somem |

**REGRA**

Um trecho no texto tem três partes: a citação entre aspas, o identificador do documento e a etiqueta ou o tema — e, quando fictício, o rótulo. Ex.: «[trecho fictício]» (entrevista_01, tema «qualidade do dado»). O SRQR (O'Brien et al., 2014) é a alternativa quando o estudo não é entrevista ou grupo focal; os dois estão na EQUATOR (Parte 6).

PARTE 6 RELATAR E FECHAR

# SAMPL — o que toda estatística relatada precisa ter; tudo num só plugin; erros comuns; checklist; glossário; referências; como citar

06

O resultado só existe quando está relatado de um jeito que outra pessoa consiga conferir: n, efeito, IC, p exato, pressupostos, figura por script, trechos com identificador. Depois, o que costuma dar errado, a lista para conferir, os termos e as fontes.

# SAMPL: o que toda estatística relatada precisa ter — e a EQUATOR, onde estão os guias de relato de cada tipo de estudo

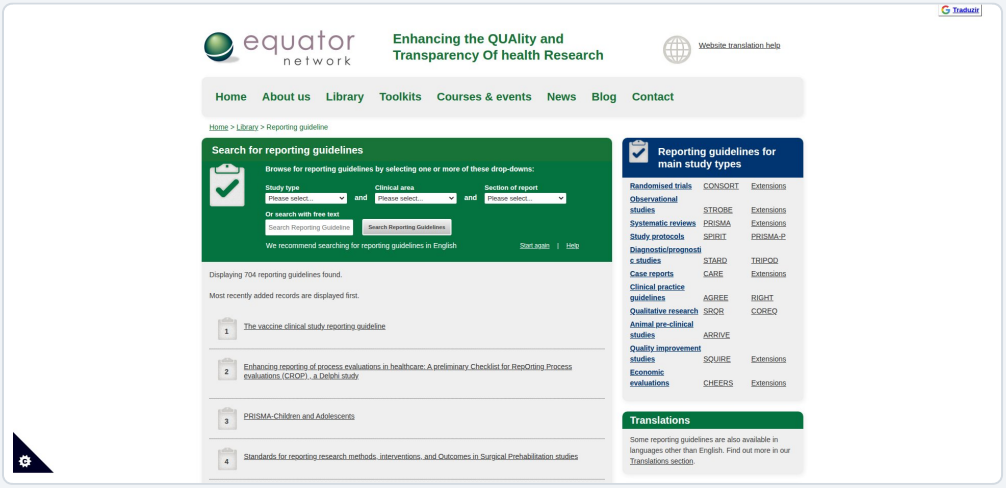



FIGURA 40 A biblioteca de guias da EQUATOR Network (captura de 09/09/2026): CONSORT (ensaios), STROBE (observacionais), PRISMA (revisões), COREQ e SRQR (qualitativa), SAMPL (estatística). O guia do seu tipo de estudo é lido antes de escrever os resultados.

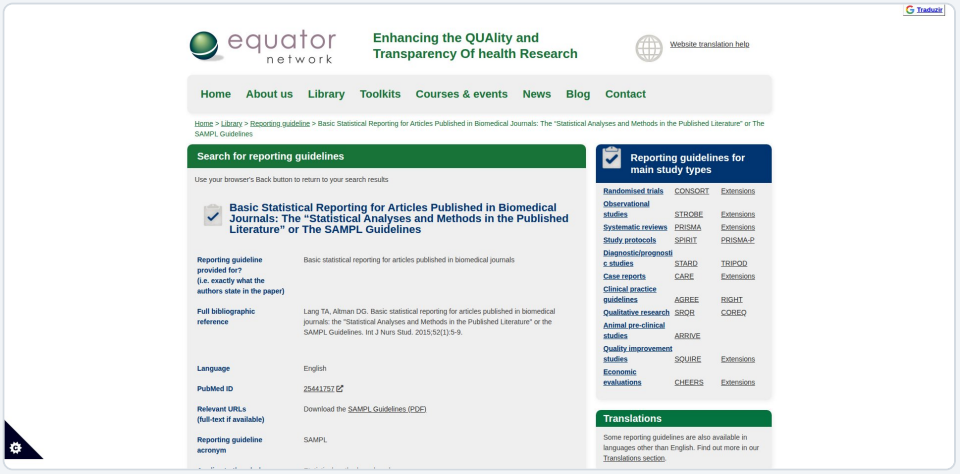



FIGURA 41 O SAMPL na EQUATOR (captura de 09/09/2026): Lang e Altman (2015), «Statistical Analyses and Methods in the Published Literature» — o relato estatístico básico para artigos: o que dizer sobre cada método e cada resultado.

| TODO RESULTADO TRAZ | EXEMPLO COM A DEMONSTRAÇÃO |
|---|---|
| **n por grupo e ausentes** | 151 Adelie, 68 Chinstrap, 123 Gentoo 2 sem massa |
| **Descritivas completas** | Média e DP (e mediana e IQR quando assimétrico) por grupo |
| **Efeito na unidade + IC 95 %** | −683,41 g (IC 95 % −840,58 a −526,25) entre fêmeas e machos |

**UMA FRASE DE RESULTADO, NO PADRÃO**

«As fêmeas pesaram, em média, 683,41 g menos que os machos (IC 95 % −840,58 a −526,25; t de Welch $t(323{,}9) = -8{,}55$; $d = -0{,}94$; Mann-Whitney $U = 6874$). A massa diferiu entre as espécies (ANOVA $F(2, 339) = 343{,}63$, $p < 0{,}001$, $\eta^2 = 0{,}67$; Kruskal-Wallis $H = 217{,}60$; $n = 342$).»

Números da execução de 09/09/2026 (Figuras 16 e 18); a forma segue Lang e Altman (2015).

| TODO RESULTADO TRAZ | EXEMPLO COM A DEMONSTRAÇÃO |
|---|---|
| **Tamanho de efeito padronizado** | d = −0,94; $\eta^2$ = 0,67; V = 0,01 |
| **Teste, estatística, graus de liberdade, p exato** | t de Welch t(323,9) = −8,55; $\chi^2(2)$ = 0,05, p = 0,976 |
| **Pressupostos e teste robusto** | Mann-Whitney U = 6874; Kruskal-Wallis H = 217,60 ao lado |
| **Software e versão** | Python 3.14.7, pandas 3.0.5, SciPy 1.18.1; ou jamovi/JASP com versão |

**REGRA**

No método: cada técnica com o motivo, o software com versão, o que foi feito com ausentes e valores extremos, a correção para múltiplas comparações. No resultado: a frase acima, para cada pergunta do projeto, na ordem do projeto. Nada de «p < 0,05» genérico, nada de resultado sem n, nada de figura sem script.

# Tudo num só plugin: a skill `analise-de-dados` roda e explica; o aluno decide a técnica, interpreta e escreve

O plugin Pesquisa MirandasTech (para Claude Code e Codex) coloca os papéis dos manuais dentro do assistente, com um estado único (`PROGRESSO.md`) e um gerente — o Pesquisador — que só fecha uma fase com evidência. A skill `analise-de-dados` segue este manual: lê no `PROGRESSO.md` a técnica prevista, roda o `limpar_dados.py` e discute cada aviso, roda o `analisar.py` conforme o projeto, reproduz o resultado numa segunda ferramenta, organiza a codificação do Taguette com o `codificar_taguette.py` — e explica cada número. Quem decide a técnica, interpreta o resultado e escreve é o aluno.

| PAPEL | O QUE FAZ COM ESTE MANUAL | SLIDES |
|---|---|---|
| **Pesquisador** | Confere que a técnica está no projeto, que existe arquivo limpo com relatório, que cada tabela e figura tem script, que o resultado tem efeito e IC e que a síntese qualitativa tem trechos | Partes 1, 3, 5 e 6 |
| **Skill analise-de-dados** | Roda os três scripts, explica a saída, reproduz em duas ferramentas, monta a frase no padrão SAMPL para o aluno revisar | Partes 2 a 6 |
| **Metodólogo** | | Parte 1 |

CLAUDE CODE — PEDIDOS NA PASTA DO PROJETO (O PLUGIN LÊ PROGRESSO.MD)

```
Pesquisador, onde paramos?
Rode o limpar_dados.py em dados/coleta.csv e me explique cada aviso.
Compare a variável tempo_parada entre os grupos do projeto e me explique o IC e o d.
Reproduza a média e o DP por grupo na planilha e me diga se o n bate.
Organize os destaques exportados do Taguette com o codebook.
/pesquisa:status
/pesquisa:proxima
```

**REGRA**

O agente **recusa** quatro pedidos: apagar os pontos «que estragam o gráfico» (mostra o efeito com e sem e registra a decisão do aluno); rodar todos os testes e dizer qual deu significativo (a técnica vem do projeto); inventar dados «parecidos» sem o rótulo «simulado»; e resumir entrevistas em temas sem *codebook*. Ele nunca digita um número no texto: todo valor nasce de um script ou de uma exportação, com o arquivo de origem nomeado.

| PAPEL | O QUE FAZ COM ESTE MANUAL | SLIDES |
|---|---|---|
| | Decide a técnica com o aluno, antes da coleta; registra mudanças | |
| **Orientador (uso de IA)** | Declara o uso do assistente na análise e no texto | Último slide |

ONDE CADA MANUAL ENTRA

Metodologia decide a técnica; Busca e PRISMA trazem a literatura; Zotero guarda as referências; este manual analisa; Git versiona scripts e dados derivados; LaTeX escreve; Notebook ajuda a ler; Uso de IA declara; o plugin mantém a ordem. Hub: mirandastech.com.br/pesquisa.

# Erros comuns — como aparecem e como corrigir

| ERRO | COMO APARECE | CAUSA | CORREÇÃO |
|---|---|---|---|
| **p sem efeito** | «Houve diferença significativa (p < 0,05)» e nada mais | Confundir «distinguível do acaso» com «importante» | EFEITO NA UNIDADE + IC 95 % + D OU H² + P EXATO |
| **Apagar o outlier** | O ponto «estranho» some da tabela e da figura sem registro | Achar que o gráfico ficou «mais limpo» | INVESTIGAR; MANTER SALVO ERRO COMPROVADO; RELATAR COM E SEM |
| **Teste escolhido pelo p** | Student, Welch e Mann-Whitney rodados; só o «que deu» no texto | Técnica decidida depois de ver o resultado | TÉCNICA DO PROJETO; PARAMÉTRICO + ROBUSTO LADO A LADO |
| **Figura de planilha colada** | Eixo sem unidade, sem n; imagem que não se refaz | Gráfico feito à mão no Excel/Calc e copiado | ANALISAR.PY --FIGURA OU NOTEBOOK; SCRIPT GUARDADO |
| **Número digitado** | Uma média no texto que não bate com a tabela | Valor copiado à mão, dado atualizado depois | TUDO VEM DO --MARKDOWN DO SCRIPT OU DA EXPORTAÇÃO |
| **Dado pessoal na planilha** | Nome, CPF ou e-mail de participante na tabela de análise | Anonimização adiada | ANONIMIZAR ANTES; BRUTO FORA DO REPOSITÓRIO (LGPD) |
| **Sem n por grupo** | «Os grupos diferiram» sem dizer quantos em cada | Tabela descritiva incompleta | N E AUSENTES POR GRUPO EM TODA TABELA |
| **Média sem DP** | | | MÉDIA ± DP; MEDIANA E IQR SE ASSIMÉTRICO |

| ERRO | COMO APARECE | CAUSA | CORREÇÃO |
|---|---|---|---|
| | «Média de 3700,66 g» solta | Só a tabela dinâmica com «média» | |
| **Correlação como causa** | «A nadadeira aumenta a massa» a partir de r = 0,87 | Leitura causal de uma associação | `«VARIAM JUNTAS»; CAUSA EXIGE DESENHO, NÃO R` |
| **Codificar sem codebook** | Etiquetas criadas ao sabor da leitura; ninguém reproduz | Começar a destacar antes de definir | `CODEBOOK ESCRITO E VERSIONADO ANTES; RECODIFICAR SE MUDAR` |
| **Tema sem trecho** | «Emergiu o tema custo» sem citação nem identificador | Síntese por contagem | `TRECHO + DOCUMENTO + ETIQUETA; COREQ/SRQR` |
| **Notebook não executado de cima a baixo** | Erro na terceira célula quando outra pessoa abre | Células rodadas fora de ordem; kernel com variáveis antigas | `REINICIAR O KERNEL E EXECUTAR TUDO; NBCONVERT --EXECUTE` |
| **DP «diferente» entre ferramentas** | 457,05 na planilha, 458,57 no Python | n do denominador contou a linha sem valor | `COMPARAR O N ANTES DE COMPARAR O RESULTADO` |

# Checklist final: antes de escrever os resultados

Marque conforme concluir. O estado fica salvo neste navegador — não é compartilhado nem enviado a lugar nenhum.

- [ ] **LIMPEZA** `limpar_dados.py` rodado; cada aviso (ausentes, duplicatas, variantes, vírgula, extremos) com decisão escrita no relatório; nada apagado sem registro
- [ ] **TIDY** Uma variável por coluna, uma observação por linha, nomes de coluna curtos com unidade; salvo em CSV
- [ ] **TÉCNICA** Decidida no projeto antes da coleta, em uma frase: variável, grupos, teste, resultado principal com IC; mudanças registradas com motivo
- [ ] **BRUTO** Arquivo original intocado, com data e origem, fora do repositório se tiver dado pessoal; cópia de trabalho separada
- [ ] **RESULTADO** Efeito na unidade, IC 95 %, tamanho de efeito padronizado (d, $\eta^2$, r, V), estatística com graus de liberdade e p exato; teste robusto ao lado
- [ ] **PRESSUPOSTOS E CORREÇÃO** Distribuição olhada na figura; número de testes contado; correção aplicada e declarada; exploratório separado do confirmatório
- [ ] **LGPD** Dado pessoal anonimizado antes de qualquer análise; tabela de correspondência fora do repositório; nada na nuvem sem anonimizar
- [ ] **DESCRITIVAS** n por grupo e ausentes; média e DP; mediana e IQR quando assimétrico — em toda tabela
- [ ] **QUALITATIVO**
- [ ] **RELATO E IA**
- [ ] **DUAS FERRAMENTAS**
- [ ] **FIGURAS**

| | | | |
|---|---|---|---|
| Codebook escrito antes e versionado; Taguette local; destaques e codebook exportados com data; síntese por tema com trechos e identificador; segundo codificador numa amostra | SAMPL (quantitativo) e COREQ/SRQR (qualitativo) conferidos item a item; software com versão; uso do assistente declarado conforme o manual de Uso de IA | Cada número do texto reproduzido numa segunda ferramenta ou caminho; n comparado primeiro; divergências explicadas | Todas geradas por script que lê o dado; eixos com unidade; n na figura ou na legenda; script e arquivo de dados registrados |

# Glossário

Tudo que aparece no manual sem ser explicado no próprio slide — e o que foi explicado uma vez e vale reler.

## Dados e descrição

**Tidy (dados arrumados)**
Cada variável numa coluna, cada observação numa linha, cada tipo de unidade numa tabela (Wickham, 2014).

**Variável**
Uma característica medida ou classificada: uma coluna da tabela arrumada (`body_mass_g`, `species`).

**Observação**
Uma unidade medida: uma linha da tabela (um pinguim, um participante, uma máquina).

**Ausente**
Valor não registrado; fica vazio (não vira zero) e é contado à parte; muda o n de cada análise.

**Outlier (valor extremo)**
Valor muito afastado dos demais; candidato a investigação, não a exclusão; o `limpar_dados.py` avisa os que passam de 1,5 × IQR.

**IQR (intervalo interquartil)**
Distância entre o valor que deixa 25 % abaixo e o que deixa 75 % abaixo: a «caixa» do boxplot.

**Média**
Soma dos valores dividida por n; sensível a extremos.

**Mediana**
O valor do meio da coluna ordenada; resistente a extremos.

**DP (desvio padrão)**
Quanto os valores se afastam da média, na unidade da variável; acompanha toda média relatada.

**n**
Número de observações que entraram no cálculo; por grupo, sempre.

**IC 95 % (intervalo de confiança)**
Faixa de valores compatível com os dados para a diferença ou o efeito; se não inclui o zero, a diferença é distinguível de zero.

**p**
Probabilidade de observar um resultado tão extremo quanto o obtido se não houvesse efeito; não mede o tamanho do efeito; relatado exato.

**Tamanho de efeito**
Quanto o efeito vale, na unidade da variável (diferença) ou padronizado (d, $\eta^2$, r, V).

**d de Cohen**
Diferença entre duas médias dividida pelo desvio padrão combinado.

**$\eta^2$ (eta ao quadrado)**
Fração da variação total explicada pelos grupos (0 a 1); tamanho de efeito da ANOVA.

## Testes, notebooks e qualitativo

**t de Welch**
Compara duas médias sem supor variâncias iguais; graus de liberdade fracionários (323,9).

**Mann-Whitney**
Compara dois grupos por postos (posições), sem supor distribuição normal; par robusto do t.

**ANOVA**
Compara três ou mais médias; F é a razão entre a variação entre grupos e dentro deles.

**Kruskal-Wallis**
Compara três ou mais grupos por postos; par robusto da ANOVA.

**Pearson (r)**
Correlação linear entre duas medidas, de −1 a 1; não implica causa.

**Spearman (ρ)**
Correlação por postos; par robusto do Pearson; mede relação monotônica.

**Qui-quadrado ($\chi^2$)**
Testa a associação entre duas variáveis categóricas numa tabela cruzada; exige esperados ≥ 5.

**Teste robusto (não paramétrico)**
Usa as posições dos valores em vez dos valores; não depende da forma da distribuição.

**Múltiplas comparações**
Vários testes no mesmo estudo aumentam a chance de um p «significativo» por acaso; exigem correção e declaração.

**Notebook**
Arquivo `.ipynb` (JSON aberto) com células de código, texto e saídas; abre no JupyterLab e no Colab.

**Kernel**
O processo que executa as células e guarda as variáveis; reiniciá-lo limpa a memória.

**Codebook**
Tabela com etiqueta, definição, exemplo e contraexemplo, escrita antes de codificar e versionada.

**Etiqueta (tag)**
O código aplicado a um trecho no Taguette; cada uma vem do codebook.

**Destaque (highlight)**
Um trecho selecionado e etiquetado; a exportação traz id, documento, etiqueta e conteúdo.

**Análise temática**
Método de Braun e Clarke (2006) em seis fases: familiarização, códigos, busca de temas, revisão, definição e nomeação, relato.

**Coocorrência**

**V de Cramér**

Força da associação entre duas variáveis categóricas (0 a 1); tamanho de efeito do qui-quadrado.

O mesmo trecho com duas etiquetas; onde os temas se tocam.

**REFI-QDA (QDC)**

Formato de troca de codebooks entre programas de análise qualitativa; o Taguette exporta o codebook em QDC.

**SAMPL · COREQ · SRQR**

Guias de relato da EQUATOR: estatística (Lang e Altman, 2015); entrevistas e grupos focais (Tong et al., 2007); qualitativa em geral (O'Brien et al., 2014).

# Referências (1 de 2): os dados, os métodos e os guias de relato


1. GORMAN, K. B.; WILLIAMS, T. D.; FRASER, W. R. Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus Pygoscelis). PLoS ONE, 2014. doi.org/10.1371/journal.pone.0090081

2. PALMERPENGUINS. Palmer Penguins: pacote e CSV, CC0; dados coletados por Kristen Gorman com a Palmer Station LTER. Acesso em 9 set. 2026. allisonhorst.github.io/palmerpenguins

3. WICKHAM, H. Tidy Data. Journal of Statistical Software, v. 59, n. 10, 2014. doi.org/10.18637/jss.v059.i10

4. WILSON, G.; BRYAN, J.; CRANSTON, K.; KITZES, J. et al. Good enough practices in scientific computing. PLOS Computational Biology, 2017. doi.org/10.1371/journal.pcbi.1005510

5. LANG, T. A.; ALTMAN, D. G. Basic statistical reporting for articles published in biomedical journals: the "Statistical Analyses and Methods in the Published Literature" or the SAMPL Guidelines. International Journal of Nursing Studies, v. 52, n. 1, p. 5-9, 2015. doi.org/10.1016/j.ijnurstu.2014.09.006

6. BRAUN, V.; CLARKE, V. Using thematic analysis in psychology. Qualitative Research in Psychology, 2006. doi.org/10.1191/1478088706qp063oa

7. TONG, A.; SAINSBURY, P.; CRAIG, J. Consolidated criteria for reporting qualitative research (COREQ): a 32-item checklist for interviews and focus groups. International Journal for Quality in Health Care, v. 19, n. 6, p. 349-357, 2007. doi.org/10.1093/intqhc/mzm042

8. O'BRIEN, B. C.; HARRIS, I. B.; BECKMAN, T. J. et al. Standards for Reporting Qualitative Research (SRQR). Academic Medicine, 2014. doi.org/10.1097/ACM.0000000000000388

9. THE TURING WAY COMMUNITY; SCRIBERIA. The Turing Way. Zenodo, 2024. CC BY 4.0. doi.org/10.5281/zenodo.3332807

10. EQUATOR NETWORK. Reporting guidelines: CONSORT, STROBE, PRISMA, COREQ, SRQR, SAMPL. Acesso em 9 set. 2026. equator-network.org

# Referências (2 de 2): as ferramentas — páginas oficiais e versões observadas em 09/09/2026


1. PYTHON. Python 3.14.7 (05/08/2026). python.org. Acesso em 9 set. 2026. python.org/downloads

2. PANDAS. pandas 3.0.5 (22/07/2026), BSD-3. Acesso em 9 set. 2026. pandas.pydata.org

3. SCIPY. scipy.stats (SciPy 1.18.1 nesta máquina); MATPLOTLIB (3.11.1); SEABORN (0.13.2). Acesso em 9 set. 2026. docs.scipy.org matplotlib.org

4. PROJECT JUPYTER. Jupyter; JupyterLab 4.6.3 nesta máquina. Acesso em 9 set. 2026. jupyter.org

5. GOOGLE. Colaboratory — Perguntas frequentes. Acesso em 9 set. 2026. research.google.com/colaboratory/faq

6. R PROJECT. R 4.6.1 «Happy Hop» (24/06/2026); POSIT. RStudio. Acesso em 9 set. 2026. r-project.org posit.co

7. JAMOVI. jamovi (v2.7.30, 23/05/2026); jamovi docs (CC BY-SA 4.0); jamovi Cloud. Acesso em 9 set. 2026. jamovi.org docs.jamovi.org cloud.jamovi.org

8. JASP. JASP 0.98.1 (07/07/2026), GNU AGPL v3. Universidade de Amsterdã. Acesso em 9 set. 2026. jasp-stats.org jasp-stats.org/download

9. THE DOCUMENT FOUNDATION. LibreOffice Calc (26.8 no site; 25.2.7 nesta máquina), MPL v2.0; Ajuda do LibreOffice: Criar tabelas dinâmicas (pt-BR). Acesso em 9 set. 2026. libreoffice.org help.libreoffice.org

10. GOOGLE. Planilhas Google (Google Workspace). Acesso em 9 set. 2026. workspace.google.com/products/sheets

11. OPENREFINE. OpenRefine 3.10.1 (04/03/2026); ORANGE. Orange 3.40.0. Acesso em 9 set. 2026. openrefine.org orangedatamining.com

12. TAGUETTE. Taguette 1.5.2, BSD-3; Install; Getting started with Taguette. Acesso em 9 set. 2026. taguette.org taguette.org/install

13. BRASIL. Portal Brasileiro de Dados Abertos e Catálogo Nacional de Dados; IBGE. SIDRA; OSF. Open Science Framework; ZENODO. Busca de datasets. Acesso em 9 set. 2026. dados.gov.br sidra.ibge.gov.br osf.io zenodo.org


SOBRE AS DATAS E O QUE NÃO ESTÁ AQUI

Versões, citações e nomes de botão foram lidos nas páginas acima em 09/09/2026 e mudam sem aviso. Não há preço de Excel, Colab Pro, jamovi Cloud pago, NVivo, ATLAS.ti ou MAXQDA; não há limite numérico do Colab gratuito (a FAQ diz que flutuam); não há versão do PSPP nem número de conjuntos do dados.gov.br — porque não foram conferidos nessa data. Nenhum resultado estatístico além dos da execução de 09/09/2026 aparece no manual.

# Como citar este manual, declaração de uso de IA e licença

### COMO CITAR

MATIAS, J. P. M. *Análise de dados na pesquisa: passo a passo.* MirandasTech, v1.0, set. 2026. CC BY 4.0.

Este manual: dados.mirandastech.com.br (PDF em /manual.pdf). Manuais irmãos: git.mirandastech.com.br · latex.mirandastech.com.br · busca.mirandastech.com.br · prisma.mirandastech.com.br · zotero.mirandastech.com.br · metodologia.mirandastech.com.br · notebook.mirandastech.com.br · manual.mirandastech.com.br · plugin.mirandastech.com.br · hub: mirandastech.com.br/pesquisa

### LICENÇA

Conteúdo, capturas, o notebook e os scripts `limpar_dados.py`, `analisar.py` e `codificar_taguette.py`: CC BY 4.0 — use, copie, adapte e redistribua com atribuição. O conjunto Palmer Penguins é CC0. pandas, Jupyter, Google, Posit, jamovi, JASP, The Document Foundation, Taguette, OpenRefine, Orange, Microsoft e as demais ferramentas e serviços citados são marcas e projetos de terceiros, sem vínculo nem endosso deste manual.

### DECLARAÇÃO DE USO DE IA

Este manual foi escrito com assistência de IA (Claude, Anthropic) na estruturação, na redação, na automação das capturas de tela e na execução dos scripts e do notebook, sob as regras do manual de Uso de IA na pesquisa científica. Todos os fatos, números, versões, citações e nomes de botão foram conferidos nas fontes indicadas em 09/09/2026. Nenhuma captura foi feita com login. Os dados quantitativos são reais e abertos (Palmer Penguins, CC0); os textos qualitativos são fictícios e ilustrativos, escritos para a demonstração. Decisões de conteúdo, seleção das fontes e responsabilidade pelo texto são do autor.

Sem garantia: interfaces, planos, limites e versões mudam; cada fato traz a data em que foi verificado. Quando a sua tela divergir, vale a tela. A escolha da técnica, o tratamento dos seus dados (ausentes, extremos, anonimização), a interpretação dos resultados e a declaração de uso de IA da sua pesquisa são seus e seguem o seu programa, o seu orientador, o guia de relato da sua área e, quando houver, o comitê de ética.