

OS DESAFIOS E NUANCES DO ENSINO DA LINGUAGEM DE PROGRAMAÇÃO R PARA ECÓLOGOS

Maurício Humberto Vancine^{1}, Pavel Dodonov², Bruno Vilela², Luisa Diele-Viegas³, Victor
Casagrande Souza⁴, Felipe Alvarez Silva Nunes⁴, Ana Julia Oliveira Silva⁴, Beatriz Milz⁵,
Cristina A. Kita⁶, Marco A. R. Mello⁶ & Renata L. Muylaert⁷*

¹ Universidade Estadual de Campinas (UNICAMP), Instituto de Biociências, Departamento de Biologia Animal, Campinas, SP, Brazil.

² Universidade Federal da Bahia, Instituto de Biologia, Laboratório de Ecologia Espacial, Salvador, Bahia, Brazil.

³ Universidade Federal da Bahia, Laboratório de (Bio)Diversidade no Antropoceno, Salvador, Bahia, Brazil.

⁴ Universidade Estadual Paulista (Unesp), Instituto de Biociências, Departamento de Biodiversidade, Rio Claro, SP, Brazil.

⁵ Universidade de São Paulo, Instituto de Energia e Ambiente, São Paulo, SP, Brazil.

⁶ Universidade de São Paulo, Instituto de Biociências, Departamento de Ecologia, São Paulo, Brazil.

⁷ The University of Sydney, Faculty of Science, Sydney School of Veterinary Science, Disease Ecology Laboratory, Sydney, New South Wales, Australia.

E-mails: mauricio.vancine@gmail.com (*autor correspondente); pdodonov@gmail.com;
bruno.vilela@ufba.br; luisa.mviegas@gmail.com; vicasagrand@outlook.com;
felipe.alvarez@unesp.br; anajuoliveira019@gmail.com; milz.bea@gmail.com;
c.akemikita@gmail.com; marmello@usp.br; renata.muylaert@sydney.edu.au

30 **Resumo**

31

32 A linguagem de programação R, valorizada por sua habilidade estatística, gerenciamento de
33 dados, visualização de dados e capacidades de aprendizado de máquina, tornou-se uma pedra
34 angular da ciência de dados aplicada à Ecologia. Sua natureza de código aberto e ampla gama
35 de ferramentas analíticas conquistaram uma ampla base de usuários, principalmente entre os
36 ecólogos. No entanto, ensinar R para estudantes de biologia e ecologia apresenta alguns
37 desafios-chave. Aqui, mergulhamos nas nuances de ensinar R em cursos de graduação e pós-
38 graduação em ecologia, focando na manipulação de dados, visualização de dados e estatísticas.
39 Oferecemos percepções de educadores em várias fases da carreira, com forte foco no contexto
40 brasileiro. Não existe uma única maneira de ensinar R que se adapte a todas as situações; no
41 entanto, algumas diretrizes gerais podem ser fornecidas e seguidas. Antes de iniciar essa jornada
42 educacional, é necessário abordar os fundamentos, incluindo infraestrutura, *hardware* e
43 *software*. Uma vez garantidos esses pré-requisitos, os estudantes enfrentam as complexidades
44 do ambiente de programação do R. Conceitos abstratos, idiossincrasias de codificação,
45 compatibilidade de pacotes e a interação entre código e dados precisam ser dominados. Também
46 discutimos o desafio de interpretar os resultados do R e integrá-los em análises estatísticas e
47 ecológicas. Além disso, consideramos o impacto dos desafios estruturais e as pressões globais
48 na educação em R, como a pandemia de COVID-19, assim como a influência das ferramentas
49 de inteligência artificial. Em conclusão, explorar o R na ecologia vislumbra um futuro em que
50 os profissionais possuem as habilidades analíticas para desbloquear soluções e aplicações
51 inovadoras. No final dessa jornada educacional, apresentamos nossa perspectiva sobre o futuro
52 do ensino de R e como ele pode ajudar a desenvolver a nossa ciência.

53

54 **Palavras-chave:** Brasil, codificação, ecologia, ciência de dados, ensino superior, programação,
55 educação científica.

56 **Introdução**

57

58 Discussões sobre um currículo canônico em Ecologia (Dormann & Mello 2023)
59 suscitaram debates sobre como ensinar fenômenos naturais, resolução de problemas e
60 programação, além de destacarem a necessidade de atualizar os métodos de análise de dados
61 ensinados em cursos de graduação e pós-graduação. O uso da linguagem de programação R (R
62 Core Team, 2024) para análise de dados ecológicos tem crescido continuamente nas últimas
63 duas décadas (Touchon & McCoy 2016) e tornou-se uma habilidade importante para ecólogos
64 (estendendo-se também a biólogos e profissionais de áreas correlatas) (Auker & Barthelmess
65 2020). No entanto, ensinar R pode ser desafiador, especialmente para alunos com pouca ou
66 nenhuma experiência em programação (Medeiros et al. 2019, Auker & Barthelmess 2020).

67 A análise de dados ecológicos no Brasil tradicionalmente dependia principalmente de
68 softwares proprietários, na maioria dos casos caros e inflexíveis. A mudança para a
69 programação começou a transformar esse cenário no início dos anos 2000, com o lançamento da
70 versão 1.0.0 do R (<https://cran.r-project.org/bin/windows/base/old>). As instituições brasileiras
71 começaram lentamente a integrar o R em seus currículos, embora de forma não sistemática ou
72 uniforme. Para facilitar essa transição, oficinas, cursos e recursos *on-line* proliferaram, tornando
73 as ferramentas de aprendizado de R acessíveis a estudantes e profissionais em todo o mundo.
74 Hoje, o R é uma ferramenta fundamental na pesquisa ecológica (Foulkes 2009, Lai et al. 2019,
75 Mello & Muylaert 2020, Da Silva et al. 2022, Wickham et al. 2023), e seu uso também cresce
76 em sala de aula. O uso do R pode capacitar as novas gerações de ecólogos a conduzir pesquisas
77 criativas, robustas e reprodutíveis, promovendo uma comunidade científica mais aberta,
78 colaborativa e inovadora.

79 Apesar desse potencial transformador, o ensino de R para estudantes de ecologia
80 apresenta muitos desafios, em grande parte moldados pelas concepções dos alunos sobre
81 matemática, estatística e programação (Auker & Barthelmess 2020). Muitos estudantes
82 ingressam nos cursos de ecologia (e também em cursos de biologia e áreas correlatas), com uma
83 grande paixão pela história natural, mas também com um considerável grau de ansiedade

relacionado a habilidades quantitativas. Isso pode criar uma barreira mental, levando os alunos a acreditar que a proficiência em R e programação está fora do seu alcance.

Outra dificuldade no ensino de R pode ser descrita como a diferença entre a "mentalidade de cliques - interfaces gráficas" e a "mentalidade de linha de comando". O uso de interfaces de usuário gráficas (*Graphical user interface* - GUIs) amigáveis incentiva o uso de menus e botões em vez da compreensão de como o código subjacente funciona. Embora as GUIs tornem a tecnologia amplamente acessível, elas limitam uma compreensão mais profunda (Selwyn 2022). Essa mentalidade parece ser ainda mais prevalente entre as gerações mais jovens devido à sua constante exposição a GUIs em smartphones e tablets. A dependência de GUIs é reforçada por ferramentas educacionais, tendências de software e uma ênfase cultural na gratificação instantânea (Yin & Shen 2024), e pode levar a um baixo engajamento em cursos de programação. Os alunos também podem achar a programação intimidadora e preferir usar ferramentas mais amigáveis e fáceis de usar (Farrell & Carey 2018). Para abordar essa questão, os programas educacionais estão incorporando tanto interfaces gráficas quanto as de linha de comando para melhorar a alfabetização computacional, fomentando a transição para uma "mentalidade de linha de comando", na qual os alunos escrevem e compreendem o código.

Em última análise, em nossa compreensão, a chave para o ensino de R para estudantes de ecologia reside em abordar suas concepções e inseguranças, fornecendo-lhes uma base sólida em ciência da computação e estratégias eficazes de resolução de problemas. Ao fazer isso, os instrutores podem capacitar os alunos a superar sua apreensão inicial e desenvolver confiança em programação. Isso não só aprimora a capacidade desses alunos de conduzir pesquisas ecológicas sofisticadas, mas também amplia seu conjunto de habilidades, tornando-os cientistas mais versáteis e com maior empregabilidade. Por meio de orientação estruturada e atividades de apoio, os alunos podem transformar sua hesitação inicial em uma forte proficiência em R, aproveitando seus recursos poderosos para impulsionar suas carreiras. Assim, o ensino bem-sucedido de R para estudantes de ecologia depende de uma abordagem abrangente que combine fundamentos sólidos, habilidades analíticas avançadas e estratégias de

aprendizagem ativa. Nas próximas seções, discutiremos algumas maneiras de alcançar esses objetivos. Pretendemos que nossas sugestões sejam aplicáveis a uma ampla gama de níveis de cursos, desde cursos introdutórios de graduação para alunos sem contato prévio com R ou estatística até cursos de mestrado e doutorado mais avançados.

Aumentando a alfabetização computacional

Para ajudar os alunos a superar esses problemas, os instrutores podem enfatizar a utilidade prática do R na pesquisa ecológica, demonstrando como ele simplifica a análise e a visualização de dados complexos, fomentando, em última instância, sua independência analítica e tornando suas investigações científicas mais robustas e perspicazes. Isso pode ser especialmente útil em cursos introdutórios ou para alunos que não tiveram contato prévio com R ou programação. Os instrutores também podem mostrar aos alunos que a familiaridade com programação é fundamental para a compreensão da literatura ecológica, visto que a maior parte das análises, incluindo os métodos, é atualmente codificada em R (Zuur et al. 2010, Zuur & Ieno 2016). Para terem sucesso como ecólogos profissionais, é imprescindível que os alunos compreendam como os dados são analisados em artigos científicos essenciais.

Uma compreensão básica de ciência da computação também é essencial para aprender R efetivamente. Isso inclui familiaridade com conceitos como sistemas operacionais, tipos de arquivo, estruturas de diretório, tipos de dados, estruturas de controle (por exemplo, laços e condicionais), vocabulário e sintaxe da linguagem, funções e argumentos, e a lógica de como os computadores executam o código. Especificamente para o R, isso também inclui a compreensão de funções e pacotes. Funções são trechos de código que executam uma tarefa específica, com uma estrutura semelhante a: `nome_funcao(argumento1, argumento2, ...)`. Podemos traçar paralelos com o conceito de uma função matemática $f(x)$: "f" é a função que realiza uma operação para modificar x; "x" representa os dados de entrada ou argumentos. Os pacotes são coleções de funções escritas em R, dados exemplificadores e documentação que descreve como

usá-las (Wickham & Bryan 2023). Eles contêm funções projetadas para executar tarefas específicas que vão além daquelas instaladas no R básico (*Base R*). Existem milhares de pacotes (descubra por si mesmo no R: ``nrow(available.packages())`` ou neste link: <https://cran.r-project.org/web/packages>) para uma ampla gama de finalidades; os pacotes de R mais populares entre os ecólogos incluem *lme4*, *vegan*, *nmle*, *ape* e *MuMIn* (Lai et al., 2023). Os pacotes podem estar disponíveis como versões revisadas no *Comprehensive R Archive Network* (CRAN) ou como versões em desenvolvimento no GitHub e em outras plataformas de controle de versão.

Considerando que a maioria dos estudantes iniciantes em ecologia (por exemplo, alunos de graduação cursando suas primeiras disciplinas de ecologia), bem como alguns estudantes mais avançados (tais como alunos de pós-graduação que não tiveram contato prévio com o R), precisam adquirir essa base (Braga et al. 2023), os instrutores devem começar com esses fundamentos antes de se aprofundarem em particularidades mais complexas do R. Introduzir esses conceitos por meio de exemplos ecológicos relevantes pode preencher a lacuna entre o conhecimento teórico e a aplicação prática, promovendo uma compreensão mais clara da lógica de programação.

A programação também envolve lidar com *bugs* e erros, o que significa que os alunos devem desenvolver uma abordagem sistemática para depuração e solução de problemas. Incentivar uma "mentalidade de linha de comando", que veja os erros não como falhas, mas como oportunidades de aprendizado, pode aprimorar significativamente suas habilidades de resolução de problemas (Kruger & Dunning 1999). Os instrutores devem ensinar os alunos a interpretar mensagens de erro, usar ferramentas de depuração e empregar estratégias como comentários no código e testes incrementais para identificar e corrigir problemas (Zuur et al. 2010). Ambientes de aprendizagem colaborativa, onde os alunos podem compartilhar e discutir desafios comuns, oferecem suporte adicional e reduzem a frustração com a depuração (McCartney 2016).

Tipos de cursos

165

166 Antes de ensinar R, é importante definir o tipo de curso que será ministrado: o curso é
167 focado no próprio R (por exemplo, um curso de programação em R) ou utiliza o R como
168 ferramenta para ensinar outros assuntos (por exemplo, um curso de estatística que usa R, mas
169 que também poderia usar um software com interface gráfica)? Podemos também ter uma
170 situação intermediária – um curso que visa ensinar os conceitos e suas aplicações no ambiente
171 R, no qual o aprendizado de R é parte essencial do curso, mas o curso em si não se limita a isso.
172 Assim, é fundamental esclarecer como o R se encaixa no curso — é o assunto principal ou uma
173 ferramenta? — e comunicar isso aos alunos desde o início. Também é essencial ter objetivos
174 claros em mente e avaliar a infraestrutura disponível, tanto na instituição de ensino quanto nas
175 residências dos alunos. Por exemplo, nem todas as universidades possuem laboratórios de
176 informática que os alunos possam usar livremente, os quais facilitam enormemente o
177 aprendizado de R, especialmente para alunos de famílias de baixa renda.

178 Ensinar a linguagem de programação R para estudantes de ecologia também requer uma
179 abordagem estruturada que equilibre cursos básicos e avançados, incorporando metodologias
180 ativas para aprimorar os resultados de aprendizagem (Deslauriers et al. 2019, Hoffman &
181 Wright 2024). Para aprender R com sucesso, os alunos devem estar preparados para se envolver
182 profundamente com os aspectos conceituais e práticos da linguagem. Cursos básicos focados em
183 R devem estabelecer uma base sólida, introduzindo os alunos aos conceitos fundamentais de
184 programação e à sintaxe do R (veja, por exemplo, “Uso da Linguagem R para Análise de
185 Dados”: <https://uspdigital.usp.br/janus/Disciplina?sgldis=BIE5782&> e “Programação em R”:
186 <https://www.coursera.org/learn/r-programming>). Esses cursos geralmente abordam tipos de
187 dados, a lógica das funções, seus parâmetros e argumentos, manipulação de dados e
188 visualizações básicas em R. Conceitos abstratos devem ser relacionados a dados ecológicos e
189 questões de pesquisa; por exemplo, pequenos conjuntos de dados empíricos podem ser usados
190 para criar objetos e aplicar funções a eles (um exemplo usado por um de nós consiste em dez
191 valores de cobertura de bivalves associados a dez valores que representam a qualidade da água,

coletados na mesma cidade onde a universidade está localizada, em locais conhecidos pela maioria dos alunos). Outros dados já organizados como Data Papers (por exemplo, ATLANTIC SERIES - [https://esajournal.onlinelibrary.wiley.com/doi/toc/10.1002/\(ISSN\)1939-9170.AtlanticPapers](https://esajournal.onlinelibrary.wiley.com/doi/toc/10.1002/(ISSN)1939-9170.AtlanticPapers)) também podem ser usados, principalmente porque geralmente são extensos (com muitas linhas e colunas), podem exigir a junção de diferentes tabelas e quase sempre apresentam dados ausentes e até mesmo erros de digitação que podem ser desafios interessantes de manipulação de dados pelos alunos. A prática, com exercícios regulares envolvendo conjuntos de dados do mundo real, também é crucial.

Alunos em cursos de R mais avançados (veja, por exemplo, “Modelagem de Regressão na Prática”: <https://www.coursera.org/learn/regression-modeling-practice>) devem encontrar tópicos mais complexos, como métodos estatísticos avançados, modelagem e a criação de funções personalizadas. Esses cursos exigem um nível mais elevado de envolvimento e maiores habilidades de resolução de problemas. Para ter sucesso, os alunos precisam dominar os fundamentos e estar preparados para lidar com análises complexas. Cursos avançados devem enfatizar a aplicação do R a problemas ecológicos relevantes, incentivando os alunos a desenvolver códigos (*scripts*) e funções personalizados. A capacidade de solucionar problemas e otimizar o código de forma independente é fundamental nesta etapa, assim como a capacidade de interpretar e comunicar resultados complexos de forma eficaz. Cursos mais avançados podem incluir a criação de pacotes, o que exige conhecimento avançado de funções e suas descrições.

Por outro lado, um curso que utiliza o R como ferramenta e não como assunto principal pode enfatizar um pequeno conjunto de habilidades e funções básicas, permitindo que os alunos deem os primeiros passos no aprendizado do R. Essas habilidades podem incluir a leitura de planilhas eletrônicas de dados no R, estatística descritiva básica, visualização de dados e testes estatísticos simples, com foco em dados e aplicações do mundo real. Dessa forma, mesmo sem adquirir conhecimento detalhado da sintaxe e das estruturas de dados do R, o aluno poderá utilizá-lo para análises básicas. Ainda assim, é necessário algum entendimento da linguagem

219 subjacente (por exemplo, o que são objetos e funções) e, se possível, ensinar a criação de
220 funções do zero pode ser extremamente benéfico.

221 Além dos cursos básicos e avançados de R, essa linguagem de programação também
222 pode auxiliar no ensino de disciplinas genuinamente ecológicas. Por exemplo, na Universidade
223 de São Paulo (USP), existem aulas práticas sobre ecologia animal elaborados como cadernos em
224 R em um curso de graduação presencial (veja "Tópicos Avançados em Ecologia Animal":
225 <https://uspdigital.usp.br/jupiterweb/obterDisciplina?nomdis=&sgldis=bie0315>). Nessas práticas,
226 os alunos reproduzem análises de dados encontradas em artigos científicos de ecologia e são
227 incentivados a conectar conceitos e análises por meio de uma combinação de aulas expositivas,
228 leituras, questionários e atividades práticas. A mesma estratégia é utilizada por nós em um curso
229 a distância sobre ciência de redes, que também inclui cadernos em R em suas aulas práticas
230 (veja "Redes Ecológicas": <https://www.coursera.org/learn/redesecologicas>).

231 As metodologias ativas desempenham um papel fundamental no ensino de todos esses
232 cursos. Essas abordagens, que incluem sala de aula invertida, aprendizagem entre pares e
233 aprendizagem baseada em projetos, exigem que os alunos sejam protagonistas no processo de
234 aprendizagem (Deslauriers et al. 2019, e veja a proposta no Cap. 1 de Da Silva et al. 2022 -
235 <https://analises-ecologicas.com/cap1#como-ensinar-e-aprender-com-esse-livro>). Os alunos
236 devem estar dispostos a se envolver proativamente, trabalhar em colaboração e abraçar a
237 natureza iterativa da programação. Sabemos que nem todos os alunos possuem esse nível de
238 motivação; no entanto, em nossa experiência, aulas com atividades práticas, nas quais os alunos
239 conseguem resolver problemas e ver seu código funcionando, podem aumentar o engajamento.
240 Além disso, manter os alunos motivados a aprender programação pode ser um desafio
241 significativo, visto que nem todos têm aptidão natural para conceitos lógicos e abstratos.
242 Portanto, acreditamos que exemplos e aplicações práticas são bons motivadores para que os
243 alunos mantenham um alto nível de interesse.

244 Em uma sala de aula invertida, os alunos podem assistir a palestras gravadas, ler artigos
245 por conta própria e, em seguida, participar de exercícios práticos de programação durante o

tempo de aula presencial. Na aprendizagem baseada em projetos, os alunos podem se organizar em grupos e trabalhar no desenvolvimento de um projeto de pesquisa ou na resolução de um problema de forma autônoma, com o apoio e a orientação dos instrutores. Os projetos em grupo podem ser bem definidos ou mais abertos, simulando cenários reais de pesquisa ecológica, e podem ser concluídos em uma única aula ou se estender por várias semanas. Ao integrar essas metodologias, os educadores podem criar um ambiente de aprendizagem dinâmico e colaborativo que incentiva os alunos a assumirem a responsabilidade por seu próprio processo de aprendizagem.

Por fim, pode haver grande variação na velocidade de aprendizagem e no nível de habilidade entre os alunos, e pode ser necessário usar estratégias para maximizar a experiência de aprendizagem para todos, como aulas de nivelamento, podem aumentar as chances de proporcionar uma experiência uniforme para todos. Trabalhar em atividades básicas e avançadas que abrangem diferentes níveis de habilidade simultaneamente também é uma possibilidade. Além disso, alunos mais habilidosos ou com experiência prévia podem ser convidados a atuar como "monitores temporários" e ajudar seus colegas.

Ensinando o pensamento computacional

Um dos desafios mais significativos que os alunos enfrentam ao aprender R não é dominar a sintaxe da linguagem (todos sabemos que mesmo usuários avançados recorrem a mecanismos de busca (por exemplo, Google) ou inteligência artificial generativa (por exemplo, ChatGPT) regularmente), mas sim compreender como abordar problemas complexos (Martinez-Blasco et al. 2023). Os alunos frequentemente ficam paralisados nas etapas iniciais porque tentam resolver o problema inteiro de uma só vez e não sabem por onde começar. A questão fundamental é operacionalizar o problema, dividindo-o em problemas menores e mais fáceis de resolver, um processo também chamado de "pensamento computacional" (Wing 2006). Sugerimos uma estrutura simples e prática, adaptada para ajudar os alunos a dividir os

273 problemas em R em partes menores e mais gerenciáveis (veja Wu et al. 2024 para proposições
274 gerais): (1) definir o problema; (2) definir as entradas e saídas; (3) decompor o processamento
275 em etapas; e (4) traduzir essas etapas em código.

276 *Definir o problema:* O primeiro passo é definir o problema em si e operacionalizá-lo.
277 Isso pode ser simples, como criar uma função para calcular a riqueza de espécies em múltiplos
278 locais, ou mais abstrato, como descrever a biodiversidade de várias áreas protegidas. Neste
279 último caso, precisamos esclarecer as métricas de biodiversidade a serem utilizadas antes de
280 tentar abordar a questão em si. Portanto, devemos operacionalizar nossas variáveis (definir e
281 especificar claramente os indicadores para conceitos abstratos), assim como fazemos no teste de
282 hipóteses. Essa é uma habilidade que pode ser explorada desde cursos básicos até avançados,
283 com problemas de diferentes níveis de complexidade.

284 *Definir as entradas e as saídas:* O próximo passo é determinar os dados de entrada, ou
285 informações, que devem ser fornecidos pelo usuário para resolver o problema. Por exemplo, se
286 quisermos calcular a riqueza de espécies de dois ou mais locais, precisamos de uma matriz de
287 comunidades, onde as linhas representam os locais e as colunas representam as espécies, e cada
288 valor indica a abundância ou presença de uma espécie em um determinado local. Em seguida,
289 definimos a saída, que deve ser definida nesta etapa, pois é importante prever claramente a
290 estrutura do resultado antes de obtê-lo. Isso permite que os indivíduos alinhem seus esforços
291 com seus objetivos finais, promovendo eficiência e eficácia na resolução de problemas. No
292 nosso exemplo, a saída esperada é a riqueza de espécies por local. Definir a entrada e a saída
293 antecipadamente também permite que o código seja modular, composto por unidades ou
294 módulos independentes e reutilizáveis, cada um com uma função específica, o que aumenta a
295 eficiência e a colaboração na ciência (por exemplo, Kass 2018).

296 *Decompor o processamento em etapas:* Agora podemos prosseguir com a divisão do
297 problema em etapas de processamento. Em vez de pedir aos alunos que calculem a riqueza de
298 espécies, precisamos orientá-los a enxergar a tarefa maior como uma sequência de etapas
299 menores e gerenciáveis. Uma abordagem é fornecer-lhes uma versão simplificada dos dados de

300 entrada e pedir que calculem a riqueza de espécies manualmente, prestando atenção às etapas
301 mentais utilizadas no cálculo e anotando-as, incluindo as informações de entrada e saída. Isso
302 pode resultar em algo como:

303

- 304 1. Dados de entrada: matriz de comunidade.
- 305 2. Identifique as espécies com abundância maior que 0 em um determinado local.
- 306 3. Conte o número de espécies únicas identificadas na etapa 1.
- 307 4. Registre/armazene esse número.
- 308 5. Repita esse processo para todos os locais.
- 309 6. Dados de saída: valores de riqueza de espécies por local.

310

311 Essas instruções escritas para executar uma tarefa também são conhecidas como
312 pseudocódigo: um protocolo simplificado e informal, escrito em linguagem natural, que é usado
313 para descrever a lógica e as etapas de um algoritmo de forma mais acessível (Petre & Winder
314 1988). Ele não está vinculado a nenhuma linguagem de programação específica e evita sintaxes
315 complexas, sendo útil para planejar e comunicar como um script ou função funcionará antes de
316 escrever o código propriamente dito (Bellamy 1994).

317 *Traduzir essas etapas em código:* Agora que o problema de criar uma função para
318 calcular a riqueza de espécies em múltiplos locais foi amplamente decomposto em etapas
319 gerenciáveis, seu pseudocódigo pode ser traduzido para código R. O pseudocódigo também
320 pode servir como documentação, aumentando a clareza e a reprodutibilidade do código. Os
321 alunos podem então testar a função, otimizar seu desempenho e até mesmo explorar algoritmos
322 completamente diferentes. Por exemplo, os alunos poderiam aprimorar o algoritmo utilizando
323 vetorização e evitando a etapa 5, que exige uma estrutura mais complexa envolvendo um laço
324 de repetição (ou *loop* – uma estrutura de controle para um conjunto de instruções a serem
325 executadas repetidamente sobre uma sequência de valores). Essa abordagem estruturada
326 também permite que os alunos adquiram habilidades essenciais de resolução de problemas,

como pensamento abstrato, planejamento sistemático e refinamento iterativo, que são de grande valor para lidar com uma ampla gama de desafios que os alunos podem encontrar em suas carreiras.

Ensino *on-line*

O isolamento social causado pelos *lockdowns* durante a pandemia de COVID-19 impactou significativamente o ensino superior, forçando a transição do ensino presencial para o *on-line* e, posteriormente, para o híbrido em universidades de todo o mundo (Chow et al. 2021). Após a pandemia, muitas atividades de ensino continuaram a ser realizadas remotamente ou semi-presencialmente, frequentemente utilizando a experiência adquirida durante esse período. Professores com pouca experiência em recursos de ensino digital (SEAD-UFBA 2020b) tiveram que converter cursos presenciais para versões totalmente remotas da noite para o dia. Abaixo, apresentamos algumas recomendações para esses cursos, com base em nossa experiência com o ensino *on-line* antes, durante e depois da COVID-19.

O primeiro passo é avaliar quais ferramentas estão disponíveis para cada aluno. Nem todos os alunos têm acesso a computadores em casa. Por exemplo, uma pesquisa realizada pela Universidade Federal da Bahia em 2020 (SEAD-UFBA 2020a) mostrou que aproximadamente um quarto de seus alunos tinha pouca ou nenhuma condição para o aprendizado *on-line*: apenas 18% tinham acesso a um computador pessoal de mesa e 72% a um notebook, e cerca de 1% não tinha acesso a nenhum dispositivo digital. Aproximadamente 20% dos alunos tinham acesso limitado à internet. Esses dados oferecem um panorama da situação precária enfrentada por muitos estudantes, especialmente em países em desenvolvimento. Embora as condições variem bastante entre as universidades, os educadores devem compreender que nem todos os alunos têm acesso a um computador com internet rápida e estável, e aqueles que têm acesso podem precisar compartilhar o dispositivo com outras pessoas em suas casas ou vizinhanças. Assim, para cursos de graduação e pós-graduação em comunidades vulneráveis (ver **Desafios**

354 **estruturais**), recomendamos que, antes de escolher as estratégias e ferramentas de
355 aprendizagem, seja realizada uma pesquisa institucional com os alunos para avaliar a
356 infraestrutura disponível. Para cursos de extensão, uma alternativa seria listar os equipamentos
357 necessários. Também é importante ter em mente que alguns pacotes podem consumir muita
358 memória para computadores mais antigos, e os requisitos mínimos para a realização do curso
359 devem ser mencionados no plano de ensino, incluindo alternativas *on-line* como o *Posit Cloud*
360 (anteriormente *RStudio Cloud*) (<https://posit.cloud>).

361 Além disso, a maioria dos alunos não terá duas telas disponíveis. Se precisarem
362 acompanhar a demonstração do instrutor e escrever seu próprio código simultaneamente,
363 provavelmente ficarão perdidos no processo. Uma alternativa é separar os momentos de ensino
364 e de programação/escrita de código: o instrutor primeiro faz a demonstração e, em seguida, os
365 alunos têm tempo para trabalhar em seu próprio código, com base em suas anotações.
366 Recomenda-se fortemente fornecer material de preparação antecipadamente em formato PDF,
367 arquivos R e R Markdown/Quarto, incluindo listas de verificação de tarefas.

368 Outra dificuldade é que, enquanto em um curso presencial o instrutor pode circular pela
369 sala de aula, verificando o código, dando sugestões e corrigindo erros, isso se torna um desafio
370 em uma aula *on-line*, principalmente se o número de instrutores e assistentes for insuficiente
371 para atender a todos os alunos. As ferramentas *on-line* comumente utilizadas tendem a
372 apresentar baixa interatividade e engajamento, dificultando a obtenção de ajuda ou a
373 comunicação de problemas devido à atenção limitada (muitas vezes, o aluno não tem permissão
374 para compartilhar sua tela por padrão). Problemas de conexão com a internet interrompem o
375 aprendizado e frustram alunos e instrutores. Uma maneira de lidar com esses desafios é usar
376 salas de apoio menores, onde os alunos podem entrar, compartilhar sua tela com um instrutor ou
377 assistente e obter ajuda para resolver seus problemas. Outra alternativa é que os alunos com
378 dificuldades compartilhem sua tela com toda a turma, para que todos, incluindo o instrutor,
379 possam ajudar a solucionar os problemas. Essa última alternativa pode ser particularmente

interessante, pois, ao ajudar os colegas a resolverem seus problemas, os alunos podem aprimorar suas próprias habilidades.

As atividades em grupo possibilitam a participação de alunos que têm acesso apenas a *smartphones* ou *tablets*, como observamos em nossos cursos durante a pandemia. Em vez de cada aluno escrever seu próprio código individualmente, eles podem ser divididos em grupos menores, nos quais um aluno compartilha sua tela com o grupo e todos contribuem. O instrutor pode interagir com cada grupo por um curto período para orientar os alunos. Essa última abordagem é especialmente interessante para estratégias ativas, como a aprendizagem baseada em projetos. Após uma breve explicação, os alunos podem receber um problema para resolver em R, como analisar um conjunto de dados para responder a uma pergunta. Os alunos então se dividem em grupos, cada grupo utilizando uma sala virtual diferente ou uma chamada separada. Enquanto cada grupo trabalha no problema, o instrutor interage com cada grupo individualmente e, ao final da aula, todos os grupos se reúnem e compartilham suas soluções.

Por fim, é importante ter uma lista bem organizada de recursos *on-line*, como textos e videoaulas, para que os alunos possam continuar aprendendo fora do horário de aula. Isso é válido também para o ensino presencial, mas ainda mais para cursos *on-line* síncronos (veja **Desafios estruturais**). Cursos totalmente assíncronos também são uma opção, especialmente para atividades de extensão. Para esses cursos, recomendamos o uso de uma combinação de vídeos, textos, questionários e exercícios práticos para os alunos resolverem, individualmente ou em grupo.

Monitoria

Como muitos alunos têm pouca ou nenhuma experiência em programação, a presença de professores convidados e monitores para auxiliar o professor principal pode melhorar significativamente o processo de aprendizagem. Em primeiro lugar, os alunos podem se sentir mais à vontade para fazer perguntas a um monitor do que a um professor, pois os monitores

407 podem oferecer atenção personalizada e um ambiente mais acessível, especialmente em turmas
408 maiores e mais formais, conduzidas por professores. Mesmo em turmas menores, com 15 a 20
409 alunos, pode ser impossível para um único professor dar atenção a todos os alunos. Nesses
410 casos, os monitores podem oferecer horários de atendimento individual ou em grupo e estimular
411 sessões de aprendizagem entre pares, aprimorando a compreensão dos alunos e promovendo um
412 senso de comunidade e trabalho em equipe. Além disso, os iniciantes podem ter dificuldades
413 com a sintaxe básica do R, como parênteses ou aspas ausentes, ou nomes de funções incorretos.
414 As mensagens de erro do R também podem ser confusas no início, tornando a depuração
415 desafiadora, demorada e frustrante. Ao identificar rapidamente os problemas de sintaxe, os
416 monitores podem otimizar o aprendizado.

417 Para estudantes com pouco conhecimento de estatística e planejamento
418 experimental/amostragem, compreender e aplicar conceitos estatísticos à análise de dados é um
419 desafio. No entanto, a experiência prática com o R pode auxiliar na aprendizagem desses
420 conceitos, especialmente se essa experiência for guiada por monitores. Os monitores podem
421 ajudar os alunos oferecendo sessões de apoio sobre conceitos e aplicações por meio da análise
422 de dados e auxiliando-os a compreender o que é solicitado em suas tarefas. Se necessário, os
423 monitores também podem incentivar boas práticas de gerenciamento de dados (Broman & Woo
424 2018) durante essas sessões, como os princípios de *tidy data* (Wickham 2014), para tornar suas
425 planilhas eletrônicas organizadas e legíveis por humanos e máquinas (Harrison 2014, Cooper &
426 Hsing 2017). Os monitores também podem auxiliar na interpretação biológica dos resultados
427 dos modelos estatísticos.

428 Embora existam muitos recursos disponíveis na internet para aprender R, a qualidade e
429 a acessibilidade desses recursos variam bastante. Os monitores podem orientar os alunos
430 indicando boas fontes de informação, incluindo livros, fóruns e cursos on-line (consulte o
431 Material Suplementar). A natureza exata dessa contribuição pode variar, podendo incluir, por
432 exemplo, o fornecimento de pseudocódigo para os alunos aplicarem; o auxílio ao raciocínio dos

alunos por meio de perguntas orientadoras; a ajuda na resolução de problemas específicos do R; ou outras formas de apoio, dependendo da natureza do curso.

Finalmente, em um tópico relacionado, o aprendizado entre pares fora da sala de aula também é algo excelente. Um exemplo é o GERE (Grupo de Estudos em R e Estatística) [<https://github.com/grupo-estudos-r-estatistica>] da Unesp – Rio Claro, um grupo de estudos em estatística e R que foi criado em 2024 em resposta à forte resistência ao pensamento matemático em cursos de graduação em ecologia. Inicialmente apoiado por alguns membros, a participação no grupo cresceu significativamente por meio das mídias sociais e do boca-a-boca, atingindo uma média de vinte participantes por reunião. Guiado pelo livro "Análise Ecológica em R" (Da Silva et al. 2022), o grupo realizou sessões práticas duas vezes por semana, em dois horários diferentes, manhã e noite, com o mesmo conteúdo, abrangendo tópicos desde o básico de R até a análise de modelos lineares, e incentivou o compartilhamento de projetos de pesquisa.

Desafios estruturais

O ensino de programação em R para estudantes de ecologia no Brasil e em outros países geopoliticamente periféricos pode ser dificultado por desafios estruturais persistentes, decorrentes de uma série de problemas de infraestrutura, instabilidade política e econômica, e restrições logísticas. Por exemplo, as universidades no Brasil frequentemente enfrentam problemas com conexão intermitente à internet, o que prejudica as aulas de programação, especialmente quando são necessárias atualizações de pacotes e *download* de dados. Isso também limita os cursos *on-line* síncronos, nos quais, devido à instabilidade da internet, quedas de energia e outros problemas imprevisíveis, os alunos podem ficar impossibilitados de participar de algumas aulas.

Outro problema são os laboratórios de informática desatualizados ou com recursos limitados (incluindo número insuficiente de cadeiras e mesas) e o suporte técnico insuficiente. Muitos laboratórios de informática estão equipados com máquinas obsoletas que não

conseguem executar com eficiência softwares modernos, incluindo as versões mais recentes do R e seus diversos pacotes. Isso força alunos e instrutores a gastarem um tempo precioso da aula resolvendo problemas técnicos em vez de se concentrarem nos conceitos e aplicações de programação.

A falta de suporte de Tecnologia da Informação (TI) adequado (assistência técnica para sistemas de computador) também representa um desafio, visto que o pessoal especializado pode ser insuficiente ou indisponível, o que significa que os problemas técnicos não podem ser resolvidos prontamente, levando a atrasos e interrupções durante as aulas. Além disso, os instrutores geralmente não possuem privilégios administrativos para atualizar ou instalar *softwares* nos computadores da universidade, limitando suas possibilidades de solucionar problemas. Isso é particularmente problemático quando atualizações de *software* ou a instalação de novos pacotes são necessárias para executar determinados scripts em R. Configurar ambientes de programação, solucionar problemas de software e garantir que todos os alunos tenham acesso às ferramentas necessárias tornam-se, portanto, tarefas complexas.

A instabilidade política no Brasil (veja <https://dynamiccecolgy.wordpress.com/2017/11/28/guest-post-doing-ecology-on-a-rollercoaster-in-brazil> para uma perspectiva sobre o tema) frequentemente leva a greves e protestos de professores e alunos – muitas vezes no final dos semestres acadêmicos, quando tópicos avançados de programação em R são geralmente abordados. Por causa disso, partes cruciais do currículo podem ser ministradas de forma apressada ou superficial. Greves de professores e alunos também podem atrasar a correção de trabalhos e o fornecimento de feedback. Sem um retorno sobre as atividades em tempo hábil, os alunos podem ter dificuldades para acompanhar os requisitos do curso e ficar para trás na compreensão de conceitos complexos de programação. Às vezes, partes inteiras do currículo não são sequer ministradas devido a restrições de tempo.

Por fim, questões logísticas, incluindo transporte e segurança, impactam a frequência e a participação dos alunos, especialmente nos cursos noturnos. Muitos estudantes enfrentam

487 dificuldades para se deslocar até o campus devido ao transporte público precário e a problemas
488 de segurança (em casos extremos, a falta de segurança leva à suspensão das aulas -
489 [https://www.correio24horas.com.br/minha-bahia/cursos-na-ufba-suspendem-aulas-em-meio-a-](https://www.correio24horas.com.br/minha-bahia/cursos-na-ufba-suspendem-aulas-em-meio-a-violencia-no-alto-das-pombas-e-calabar-0923)
490 [violencia-no-alto-das-pombas-e-calabar-0923](https://www.correio24horas.com.br/minha-bahia/cursos-na-ufba-suspendem-aulas-em-meio-a-violencia-no-alto-das-pombas-e-calabar-0923)). Esses problemas resultam em frequência
491 irregular, fazendo com que os alunos percam aulas importantes e prejudicando a continuidade
492 do aprendizado. A falta de continuidade na frequência significa que os alunos frequentemente
493 perdem palestras e sessões práticas importantes. Em uma disciplina como programação em R,
494 onde os conceitos se baseiam uns nos outros, perder mesmo algumas aulas pode prejudicar
495 significativamente o desempenho do aluno.

496 Enfrentar esses desafios exige esforços consistentes para melhorar e manter a
497 infraestrutura, fornecer suporte de TI confiável e mitigar as preocupações logísticas e de
498 segurança no campus e no transporte público, além de grandes habilidades de planejamento e
499 adaptabilidade por parte dos instrutores.

500

501 **Diversidade e inclusão**

502

503 Consideramos a inclusão e a equidade como um processo contínuo que envolve
504 estruturas, sistemas, diretrizes e nossas práticas de ensino. A aplicação dos princípios de
505 equidade e inclusão em nossas práticas de ensino pode variar de acordo com a estrutura e os
506 sistemas institucionais, e as diretrizes para o ensino superior podem diferir em termos de oferta
507 curricular e avaliações (Tabela S4). Reconhecer o multiculturalismo e a ampla diversidade de
508 estudantes no Brasil é um passo importante para a inclusão, e o avanço dos esforços para
509 disseminar práticas equitativas deve estar na vanguarda do desenvolvimento de diretrizes para a
510 prática docente no país. As barreiras à alfabetização computacional estão enraizadas na
511 desigualdade geral de alfabetização no Brasil e se estendem às barreiras linguísticas, visto que
512 grande parte do material para programação em R e a maioria das descrições de funções estão em

inglês. Além disso, as disparidades de gênero são comuns nos cursos de programação de graduação (Forrester et al. 2022).

Uma iniciativa valiosa que aborda essas questões é a R-Ladies (<https://rladies.org>), uma organização global que visa promover a diversidade de gênero na comunidade R. A organização foi fundada em 2012 por Gabriela de Queiroz, uma brasileira que estudava nos Estados Unidos na época. Desde então, a organização cresceu e possui grupos locais em todos os continentes, incluindo grupos ativos no Brasil, como R-Ladies São Paulo (<https://rladies-sp.org>), R-Ladies Goiânia (<https://www.rladiesgyn.com>) e R-Ladies Belo Horizonte (<https://rladiesbh.com.br>). Os grupos locais organizam atividades gratuitas, incluindo eventos on-line ou presenciais, grupos de estudo e clubes de leitura, focados no ensino de R. Eles também oferecem recursos on-line gratuitos, incluindo tutoriais completos, exercícios de programação e workshops gravados sobre uma ampla gama de tópicos, desde programação básica em R até análise de dados espaciais. Além disso, a R-Ladies mantém blogs e canais de mídia social ativos, onde as membros promovem a inclusão e a diversidade e compartilham seus conhecimentos, experiências e atualizações sobre os últimos desenvolvimentos na comunidade R.

Considerações adicionais e recursos

Ambientes de Desenvolvimento Integrado (IDEs) e editores de texto

A linguagem de programação R vem instalada com um console de linha de comando, chamado RGui, no Windows e diretamente no terminal em sistemas operacionais Unix (GNU/Linux e macOS). No entanto, existem outras maneiras de escrever códigos em R por meio de Ambientes de Desenvolvimento Integrados (do inglês *Integrated Development Environments* - IDEs), que auxiliam no preenchimento automático de código, geração automática de código, detecção de erros e destaque de sintaxe. Alguns IDEs foram desenvolvidos especificamente para o R, como o RStudio (RStudio Team 2020), que combina um console R, um editor com destaque de sintaxe e função de preenchimento automático, além

de abas para gráficos, histórico, Git e ajuda. O RStudio pode ser ensinado juntamente com o R, mas é importante distinguir entre os dois. Recentemente, a Posit está trabalhando em um IDE de ciência de dados de próxima geração chamado Positron™ (<https://github.com/posit-dev/positron>), que se tornou um IDE independente de linguagem, desenvolvido pela Posit (anteriormente RStudio). Outros IDEs são mostrados na Tabela S1.

tidyverse e tidymodels

O R passou por adaptações nos últimos anos, incluindo uma nova abordagem para manipulação de dados, chamada *Tidy Data* (Wickham 2014), que se concentra na limpeza e organização dos dados. Com base nessas ideias, o *tidyverse* (<https://www.tidyverse.org>) foi implementado no R como uma coleção de pacotes (por exemplo, *dplyr*, *ggplot2*, *magrittr*, *readr*, *tibble* e *tidyr*) projetados para importação, manipulação, exploração, visualização, análise e comunicação de dados (Wickham et al. 2019). Esse conjunto de pacotes altera significativamente as operações de fluxo de dados no R, introduzindo o conceito de encadeamento de funções (conhecido pelo termo em inglês *pipe* e pelo símbolo `%>%`) e programação funcional (e.g. *purrr*, Wickham 2019). Seguindo a mesma abordagem, o *tidymodels* (Kuhn & Wickham 2020) (<https://www.tidymodels.org>) é uma coleção de pacotes para modelagem e aprendizado de máquina.

Existem dúvidas sobre se essas modificações vieram para ficar ou se seria benéfico reverter alguns aspectos para a versão base do R (chamada de *Base R*) (Carscadden & Martin 2022, Çetinkaya-Rundel et al. 2022). Isso levanta uma questão importante: o R deve ser ensinado começando pela versão *Base R*, focando em suas funções e sintaxe, ou é mais vantajoso começar com o *tidyverse*? Acreditamos que ainda não há uma resposta definitiva para essa pergunta, mas sugerimos que, se o tempo e os recursos permitirem, o ensino comece com o *Base R* e, posteriormente, avance para o *tidyverse* (Da Silva et al., 2022). Muitos conceitos utilizados no *tidyverse* podem ser confusos se as classes e os tipos de objetos fundamentais não forem explicados primeiro na versão *Base R*.

ggplot2

Na abordagem *tidyverse*, o pacote *ggplot2* (Wickham 2016) é amplamente utilizado para a criação de gráficos (Nordmann et al. 2022). Baseado na gramática dos gráficos (Wilkinson et al. 2005), o pacote *ggplot2* possui sua própria sintaxe baseada em camadas, exigindo funções específicas que são conectadas usando o operador `+`. A dicotomia entre o *Base R* e o *tidyverse* também se manifesta aqui, com desafios de ensino relacionados às funções de gráficos no *Base R* e à sintaxe e funções do *ggplot2*. Os códigos do *ggplot2* podem rapidamente se tornar extensos e, por vezes, difíceis de compreender para iniciantes, devido à abordagem em camadas do *ggplot2* e à necessidade de especificação explícita dos componentes do gráfico, como temas, dados, estética e geometrias. Como mencionado anteriormente, acreditamos que não há uma resposta definitiva sobre se a visualização deve ser ensinada usando o *Base R* ou o *ggplot2*, mas recomendamos explorar ambos, se o tempo permitir.

Reprodutibilidade, código limpo e guias de estilo

A reprodutibilidade pode ser vista como um gradiente (Figueiredo et al. 2022) e inclui considerações sobre pacotes necessários, dados, código e uma descrição do ambiente R. A legibilidade do código é fundamental para orientar o leitor, mas se o código não funcionar, sua utilidade pode ser bastante limitada. Alguns programadores de R preferem minimizar o código ao máximo, o que pode comprometer a compreensão. Por outro lado, códigos longos podem ser considerados mais propensos a erros, simplesmente porque escrever mais caracteres aumenta as chances de erros de digitação. O ponto ideal para uma alta compreensão do código em uma análise específica depende do estilo, mas dependerá fortemente do código funcionar corretamente, da inclusão de comentários e de metadados adequados.

As práticas de código limpo (*Clean Code*) são essenciais para a legibilidade e a manutenibilidade do código (Martin 2011), e os guias de estilo para o *Base R* (<https://google.github.io/styleguide/Rguide.html>) e o *tidyverse* (<https://style.tidyverse.org>)

podem ajudar a padronizar os códigos. No entanto, ensinar a criação de funções desde o início dos cursos pode ser altamente benéfico para que os alunos aprimorem sua compreensão da estrutura e da sintaxe da linguagem.

R Markdown e Quarto

O R Markdown (<https://rmarkdown.rstudio.com>) é um formato de documento dinâmico que permite a integração de texto, blocos de código e resultados de códigos executados, tudo em um único documento. Ele também possibilita a criação de documentos, sites e aplicativos interativos com o Shiny, que podem ser atualizados automaticamente com novos dados (Xie et al. 2019). O Quarto (<https://quarto.org>) é um sistema de publicação científica e técnica de código aberto (*open source*) e representa a nova geração do R Markdown. Tanto o R Markdown quanto o Quarto podem aprimorar a reprodutibilidade, a transparência e a disseminação dos resultados de pesquisa. Eles podem ser usados tanto para o ensino e a preparação de material didático, quanto como uma ferramenta para que os alunos integrem texto e código na criação de seus próprios materiais de estudo.

Controle de versão (git e GitHub)

O controle de versão é um sistema que rastreia as alterações em arquivos e diretórios ao longo do tempo, permitindo que múltiplos usuários colaborem em projetos (Braga et al. 2023). O Git (<https://www.git-scm.com>) é o software mais comum para gerenciamento de controle de versão e oferece suporte à integração com repositórios remotos, como o GitHub (<https://github.com>); ambos também estão integrados ao RStudio. Ele rastreia as alterações, permitindo reversão, comparação, resolução de conflitos e colaboração por meio de ramificações e fusões. O controle de versão fornece *backup* por meio de *commits* e repositórios remotos, facilitando a revisão por pares por meio de *pull requests*. Ensinar os conceitos de controle de versão é crucial para preparar os alunos para projetos colaborativos.

621 ***Ferramentas de Inteligência Artificial Generativa (IA)***

622 O surgimento de *chatbots* (programas de computador que simulam conversas humanas)
623 baseados em grandes modelos de linguagem (conhecido em inglês como *Large Language*
624 *Models*) (IA generativa) a partir do final de 2022 (por exemplo,
625 <https://openai.com/index/chatgpt>) foi um marco transformador em ferramentas computacionais
626 e compreensão da linguagem natural, influenciando massivamente diversos domínios, incluindo
627 a educação (Cooper et al. 2024). Os modelos de IA generativa criam e revisam conteúdo
628 rapidamente (código de computador, texto narrativo, imagem, filme e áudio), incluindo
629 percepções a partir de dados e fluxos de trabalho de código complexos, a partir de instruções
630 iterativas fornecidas pelo usuário, com base nos padrões que a ferramenta aprendeu e nos dados
631 com os quais foi treinada. O uso de ferramentas generativas no ensino de R para estudantes
632 pode aprimorar a experiência de aprendizagem, fornecendo assistência e explicações
633 personalizadas em tempo real (Yilmaz & Karaoglan Yilmaz 2023). Essas ferramentas podem
634 ajudar os alunos a compreender conceitos e sintaxe complexos em R, oferecendo *feedback*
635 imediato, explicações lógicas, exemplos em diferentes linguagens (como traduzir pseudocódigo
636 para R e depois para Python) e dicas para solução de problemas. Elas também podem criar
637 fluxos de trabalho para abordar problemas relevantes para estudos ecológicos. Tudo isso pode
638 promover uma compreensão mais profunda de R, simplificar o processo de aprendizagem e
639 capacitar os alunos a aplicar suas habilidades de programação em pesquisas ecológicas do
640 mundo real.

641 No entanto, o uso de IA generativa no ensino pode gerar problemas como a
642 dependência excessiva da IA (comprometendo o pensamento crítico e as habilidades de
643 resolução independente de problemas), preocupações com a integridade acadêmica devido ao
644 potencial plágio, menor engajamento com conceitos fundamentais, diminuição da motivação
645 para aprender e reter conhecimento, e informações incorretas, conhecidas como “alucinações”,
646 fornecidas pela IA (veja, por exemplo: [https://revistapesquisa.fapesp.br/universidades-](https://revistapesquisa.fapesp.br/universidades-brasileiras-discutem-regras-de-uso-de-inteligencia-artificial)
647 [brasileiras-discutem-regras-de-uso-de-inteligencia-artificial](https://revistapesquisa.fapesp.br/universidades-brasileiras-discutem-regras-de-uso-de-inteligencia-artificial)). O acesso desigual às ferramentas

de IA pode exacerbar ainda mais as disparidades nos resultados de aprendizagem, enquanto questões éticas persistem em relação à propriedade do conteúdo e à equidade no ambiente acadêmico. Equilibrar os benefícios da IA na educação exige o enfrentamento desses desafios e a criação de diretrizes institucionais claras para garantir o uso responsável e eficaz em sala de aula.

As reações dos professores universitários à IA variam desde a proibição total até a adoção irrestrita, tornando seu uso obrigatório em trabalhos acadêmicos (veja <https://revistapesquisa.fapesp.br/pesquisadores-usam-inteligencia-artificial-em-tarefas-academicas>). As discussões abrangem a integridade acadêmica, a conduta ética, a natureza da experiência de aprendizagem e o que se espera dos alunos em seus trabalhos e resultados de aprendizagem diante de um mercado de trabalho em constante transformação. Para os alunos, isso implica apresentar trabalhos originais, citar corretamente as fontes ao utilizar conhecimento existente, seguir as diretrizes éticas em pesquisa e cumprir as instruções. Violações da integridade acadêmica comprometem a confiança nos padrões acadêmicos e o valor da educação. Ainda assim, manter a integridade acadêmica é um desafio devido à pressão por bom desempenho, à falta de diretrizes claras de conduta, ao fácil acesso à tecnologia, às diferenças culturais nas normas acadêmicas e ao rápido avanço da IA.

As universidades devem enfatizar e esclarecer os padrões esperados para garantir que os alunos desenvolvam o pensamento crítico e as habilidades de comunicação essenciais para suas futuras carreiras, ao mesmo tempo que mantêm uma conduta ética e respeitam a propriedade intelectual. Os instrutores podem adotar uma postura positiva em relação ao uso da IA no ensino, partindo do princípio de que as ferramentas de IA vieram para ficar. Por exemplo, a IA generativa pode ser usada como assistente de programação em alguns tutoriais, com discussões em sala de aula sobre a experiência dos alunos (Tabela S4).

Recursos adicionais

Existe uma ampla gama de conteúdo aberto e de alta qualidade para apoiar e simplificar o ensino de temas e processos complexos em estatística e o funcionamento da linguagem de programação R. Listamos os principais livros e recursos nas Tabelas S2 e S3, respectivamente. Também compilamos uma lista de tipos de avaliação e aplicações para cursos de ecologia que utilizam o R (Tabela S4).

Considerações finais

Ensinar e aprender R tem o potencial de contribuir enormemente para o desenvolvimento de habilidades profissionais, transformando a carreira de ecólogos (podemos atestar isso por experiência própria). Assim, convidamos os educadores a se juntarem a nós na jornada de apresentar aos estudantes de ecologia a "mentalidade da linha de comando". O R possui uma comunidade internacional engajada que enriquece a experiência educacional e torna essa jornada muito mais fácil.

Como em qualquer jornada, porém, ensinar R para estudantes de ecologia apresenta desafios e oportunidades significativas. Os alunos frequentemente enfrentam uma curva de aprendizado acentuada, lidando com a sintaxe do R, e os instrutores precisam utilizar uma variedade de recursos personalizados e projetos envolventes para ajudá-los a superar esses obstáculos. No entanto, dominar o R abre muitas oportunidades para estudantes de ecologia na academia, na indústria e no governo.

A proficiência em R permite que os alunos aproveitem ao máximo essas oportunidades, abordando questões ecológicas com precisão e transparência e aumentando sua competitividade no mercado de trabalho. Assim, embora o caminho para aprender R possa ser desafiador, os benefícios duradouros para os estudantes de ecologia são substanciais, não apenas aprimorando suas habilidades analíticas, mas também contribuindo para o crescimento coletivo da comunidade global de R e da Ecologia como disciplina científica.

Referências

- Auker, L. A., & Barthelmess, E. L. 2020. Teaching R in the undergraduate ecology classroom: approaches, lessons learned, and recommendations. *Ecosphere*, 11(4). DOI: 10.1002/ecs2.3060
- Bellamy, R. K. E. 1994. What Does Pseudo-Code Do? A Psychological Analysis of the use of Pseudo-Code by Experienced Programmers. *Human-Computer Interaction*.
- Braga, P. H. P., Hébert, K., Hudgins, E. J., Scott, E. R., Edwards, B. P. M., Sánchez Reyes, L. L., Grainger, M. J., Foroughirad, V., Hillemann, F., Binley, A. D., Brookson, C. B., Gaynor, K. M., Shafiei Sabet, S., Güncan, A., Weierbach, H., Gomes, D. G. E., & Crystal-Ornelas, R. 2023. Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution. *Methods in Ecology and Evolution*, 14(6), 1364–1380. DOI: 10.1111/2041-210X.14108
- Carscadden, K., & Martin, A. 2022. To Tidy or not When Teaching R Skills in Biology Classes. Version 5. *International Journal of Higher Education*, 11(5), 39. DOI: 10.5430/ijhe.v11n5p39
- Çetinkaya-Rundel, M., Hardin, J., Baumer, B. S., McNamara, A., Horton, N. J., & Rundel, C. 2022. An educator's perspective of the tidyverse. *Technology Innovations in Statistics Education*, 14(1). DOI: 10.5070/T514154352
- Chow, F., Calixto, C. P. G., & Mello, M. A. R. 2021. Do ensino remoto emergencial ao ensino híbrido no curso de Ciências Biológicas: a nossa visão a partir do Instituto de Biociências da Universidade de São Paulo (IB-USP). Version Supl 1. *Medicina (Ribeirão Preto)*, 54(Supl 1), e-185554. DOI: 10.11606/issn.2176-7262.rmrp.2021.185554
- Cooper, N., Clark, A. T., Lecomte, N., Qiao, H., & Ellison, A. M. 2024. Harnessing large language models for coding, teaching and inclusion to empower research in ecology and evolution. *Methods in Ecology and Evolution*, n/a(n/a). DOI: 10.1111/2041-210X.14325

728 Cooper, N., & Hsing, P.-Y. (Eds.). 2017. A guide to reproducible code in ecology and
729 evolution. British Ecological Society.

730 Da Silva, F. R., Gonçalves-Souza, T., Paterno, G. B., Provete, D. B., & Vancine, M. H. 2022.
731 Análises Ecológicas no R. PE: NUPEEA: p. 640.

732 Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., & Kestin, G. 2019. Measuring actual
733 learning versus feeling of learning in response to being actively engaged in the classroom.
734 Proceedings of the National Academy of Sciences, 116(39), 19251–19257. DOI:
735 10.1073/pnas.1821936116

736 Dormann, C. F., & Mello, M. A. R. 2023. Why we need a Canonical Ecology Curriculum. Basic
737 and Applied Ecology, 71, 98–109. DOI: 10.1016/j.baae.2023.05.009

738 Farrell, K. J., & Carey, C. C. 2018. Power, pitfalls, and potential for integrating computational
739 literacy into undergraduate ecology courses. Ecology and Evolution, 8(16), 7744–7751.
740 DOI: 10.1002/ece3.4363

741 Figueiredo, L., Scherer, C., & Cabral, J. S. 2022. A simple kit to use computational notebooks
742 for more openness, reproducibility, and productivity in research. PLOS Computational
743 Biology, 18(9), e1010356. DOI: 10.1371/journal.pcbi.1010356

744 Forrester, C., Schwikert, S., Foster, J., & Corwin, L. 2022. Undergraduate R Programming
745 Anxiety in Ecology: Persistent Gender Gaps and Coping Strategies. CBE—Life Sciences
746 Education, 21(2), ar29. DOI: 10.1187/cbe.21-05-0133

747 Foulkes, A. S. 2009. Applied Statistical Genetics with R: For Population-based Association
748 Studies. New York, NY: Springer New York. DOI: 10.1007/978-0-387-89554-3

749 Harrison, K. (Ed.). 2014. A guide to data management in ecology and evolution. British
750 Ecological Society.

751 Hoffman, A. M., & Wright, C. 2024. Ten simple rules for teaching an introduction to R. PLOS
752 Computational Biology, 20(5), e1012018. DOI: 10.1371/journal.pcbi.1012018

753 Kruger, J., & Dunning, D. 1999. Unskilled and unaware of it: How difficulties in recognizing
 754 one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social*
 755 *Psychology*, 77(6), 1121–1134. DOI: 10.1037/0022-3514.77.6.1121
 756 Kuhn, M., & Wickham, H. 2020. Tidymodels: a collection of packages for modeling and
 757 machine learning using tidyverse principles. (Retrieved on from
 758 <https://www.tidymodels.org>).
 759 Lai, J., Cui, D., Zhu, W., & Mao, L. 2023. The Use of R and R Packages in Biodiversity
 760 Conservation Research. Version 12. *Diversity*, 15(12), 1202. DOI: 10.3390/d15121202
 761 Lai, J., Lortie, C. J., Muenchen, R. A., Yang, J., & Ma, K. 2019. Evaluating the popularity of R
 762 in ecology. *Ecosphere*, 10(1), e02567. DOI: 10.1002/ecs2.2567
 763 Martin, R. C. 2011. *The clean coder: a code of conduct for professional programmers*. Upper
 764 Saddle River, NJ: Prentice Hall: p. 210.
 765 Martinez-Blasco, M., Serrano, V., Prior, F., & Cuadros, J. 2023. Analysis of an event study
 766 using the Fama–French five-factor model: teaching approaches including spreadsheets and
 767 the R programming language. *Financial Innovation*, 9(1), 76. DOI: 10.1186/s40854-023-
 768 00477-3
 769 McCartney, M. 2016. Mistakes as a pathway to learning. *Science*, 352(6290), 1186.3-1187.
 770 DOI: 10.1126/science.352.6290.1186-c
 771 Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. 2019. A Systematic Literature Review on
 772 Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions*
 773 *on Education*, 62(2), 77–90. DOI: 10.1109/TE.2018.2864133
 774 Mello, M. A. R., & Muylaert, R. L. 2020. Network Science as a Framework for Bat Studies. In:
 775 21. Network Science as a Framework for Bat Studies. pp. 373–390. University of Chicago
 776 Press.
 777 Nordmann, E., McAleer, P., Toivo, W., Paterson, H., & DeBruine, L. M. 2022. Data
 778 Visualization Using R for Researchers Who Do Not Use R. *Advances in Methods and*

779 Practices in Psychological Science, 5(2), 25152459221074654. DOI:
780 10.1177/25152459221074654

781 Pagani-Núñez, E., Yan, M., Hong, Y., Zeng, Y., Chen, S., Zhao, P., & Zou, Y. 2022.
782 Undergraduates' perceptions on emergency remote learning in ecology in the post-pandemic
783 era. Ecology and Evolution, 12(3), e8659. DOI: 10.1002/ece3.8659

784 Petre, M., & Winder, R. 1988. Issues governing the suitability of programming languages for
785 programming tasks. In: Proceedings of the Fourth Conference of the British Computer
786 Society on People and computers IV. pp. 199–215. USA: Cambridge University Press.

787 R Core Team. 2024. R: A Language and Environment for Statistical Computing. R Foundation
788 for Statistical Computing, Vienna, Austria. <https://www.R-project.org>.

789 RStudio Team. 2020. RStudio: Integrated Development Environment for R.
790 <http://www.rstudio.com>.

791 SEAD-UFBA. 2020a. As condições para aprendizagem online dos estudantes de graduação da
792 UFBA em tempos de pandemia da Covid-19. Superintendência de educação a distância da
793 Universidade Federal da Bahia.

794 SEAD-UFBA. 2020b. Diagnóstico das competências digitais dos professores da
795 UFBA. Superintendência de educação a distância da Universidade Federal da Bahia.

796 Selwyn, N. 2022. Education and technology: key issues and debates. Third edition Third edition
797 ed. London New York Oxford New Delhi Sydney: Bloomsbury Academic: p. 222.

798 Touchon, J. C., & McCoy, M. W. 2016. The mismatch between current statistical practice and
799 doctoral training in ecology. Ecosphere, 7(8), e01394. DOI: 10.1002/ecs2.1394

800 Wickham, H. 2014. Tidy Data. Journal of Statistical Software, 59(10). DOI:
801 10.18637/jss.v059.i10

802 Wickham, H. 2016. ggplot2: Elegant Graphics for Data Analysis. 2 ed. Springer International
803 Publishing. DOI: 10.1007/978-3-319-24277-4

804 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Golemund, G.,
805 Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K.,

806 Ooms, J., Robinson, D., Seidel, D., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo,
807 K., & Yutani, H. 2019. Welcome to the Tidyverse. *Journal of Open Source Software*, 4(43),
808 1686. DOI: 10.21105/joss.01686

809 Wickham, H., & Bryan, J. 2023. *R Packages: Organize, Test, Document, and Share Your Code*.
810 2º edição. O'Reilly Media.

811 Wickham, H. 2019. *Advanced R*. 2º edição. Boca Raton: Chapman and Hall/CRC: p. 588.

812 Wickham, H., Çetinkaya-Rundel, M., & Golemund, G. 2023. *R for Data Science: Import, Tidy,*
813 *Transform, Visualize, and Model Data*. 2nd edition ed. Beijing Boston Farnham Sebastopol
814 Tokyo: O'Reilly Media: p. 576.

815 Wilkinson, L., Wills, D., Rope, D., Norton, A., & Dubbs, R. 2005. *The Grammar of Graphics*.
816 2nd 2005 ed. edição 2nd 2005 ed. edição ed. New York: Springer.

817 Wing, J. M. 2006. Computational thinking. *Communications of the ACM*, 49(3), 33–35. DOI:
818 10.1145/1118178.1118215

819 Wu, T.-T., Asmara, A., Huang, Y.-M., & Permata Hapsari, I. 2024. Identification of Problem-
820 Solving Techniques in Computational Thinking Studies: Systematic Literature Review. *Sage*
821 *Open*, 14(2), 21582440241249897. DOI: 10.1177/21582440241249897

822 Xie, Y., Allaire, J. J., & Golemund, G. 2019. *R Markdown: the definitive guide*. Boca Raton:
823 CRC Press, Taylor and Francis Group: p. 1.

824 Yilmaz, R., & Karaoglan Yilmaz, F. G. 2023. The effect of generative artificial intelligence
825 (AI)-based tool use on students' computational thinking skills, programming self-efficacy
826 and motivation. *Computers and Education: Artificial Intelligence*, 4, 100147. DOI:
827 10.1016/j.caeai.2023.100147

828 Yin, B., & Shen, Y. 2024. Development and Validation of the Compensatory Belief Scale for
829 the Internet Instant Gratification Behavior. *Heliyon*, 10(1). DOI:
830 10.1016/j.heliyon.2024.e23972

831 Zuur, A. F., & Ieno, E. N. 2016. A protocol for conducting and presenting results of regression-
832 type analyses. *Methods in Ecology and Evolution*, 7(6), 636–645. DOI: 10.1111/2041-
833 210X.12577

834 Zuur, A. F., Ieno, E. N., & Elphick, C. S. 2010. A protocol for data exploration to avoid
835 common statistical problems. *Methods in Ecology and Evolution*, 1(1), 3–14. DOI:
836 10.1111/j.2041-210X.2009.00001.x

837

838 **Financiamento**

839

840 **MHV**: processos nº 2022/01899-6 e nº 2024/19865-6, Fundação de Amparo à Pesquisa do
841 Estado de São Paulo (FAPESP). **BV**: Bolsa de produtividade do CNPq (311213/2023-5). **CAK**:
842 Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, 88887.802356/2023-
843 00) e processo nº 2023/17728-9, Fundação de Amparo à Pesquisa do Estado de São Paulo
844 (FAPESP). **MARM**: Fundação Alexander von Humboldt (AvH, 1134644), Conselho Nacional
845 de Desenvolvimento Científico e Tecnológico (CNPq, 304498/2019-0), processos nº
846 2018/20695-7 e nº 2023/02881-6 Fundação de Amparo à Pesquisa do Estado de São Paulo
847 (FAPESP). **RLM**: Australian Research Council Australian Laureate Fellowship (FL240100037)
848 financiada pelo Governo Australiano.

849

850 **CRedit declaração de contribuição de autoria**

851

852 **MHV**: Conceituação, Supervisão, Redação – rascunho original, Redação – revisão e edição.
853 **PD**: Conceituação, Redação – rascunho original, Redação – revisão e edição. **BV**:
854 Conceituação, Redação – rascunho original, Redação – revisão e edição. **LDV**: Redação –
855 revisão e edição. **VCS**: Conceituação, Redação – rascunho original, Redação – revisão e edição.
856 **FASN**: Redação – revisão e edição. **AJOS**: Redação – rascunho original, Redação – revisão e
857 edição. **BM**: Redação – rascunho original, Redação – revisão e edição. **MVCV**: Redação –

858 revisão e edição. **CAK**: Conceituação, Redação – rascunho original, Redação – revisão e

859 edição. **MARM**: Redação – rascunho original, Redação – revisão e edição. **RLM**:

860 Conceituação, Supervisão, Redação – rascunho original, Redação – revisão e edição.

861

862 **Agradecimentos**

863

864 Agradecemos às comunidades globais do R e do *Stack Overflow* por sempre nos ajudarem

865 quando encontramos dificuldades. Nossos monitores de graduação e pós-graduação, com sua

866 perspectiva atualizada sobre os anseios e desafios das gerações mais jovens, também

867 desempenharam um papel fundamental em nos ajudar a aprender a ensinar R em sala de aula.

868 Por fim, somos profundamente gratos aos alunos brilhantes e entusiasmados que encontramos

869 em todas as turmas, cuja curiosidade e sede de conhecimento nos motivam a aprimorar nosso

870 ensino um pouco mais a cada dia.

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885 **Material Suplementar**

886
887
888

Tabela S1. Lista selecionada de interfaces para programação em R.

Interfaces	Ano de lançamento	Link
<i>R IDEs</i>		
RStudio	2011	https://www.rstudio.com
Positron	2024	https://github.com/posit-dev/positron
Tinn-R	2003	https://tinn-r.org/en
R Commander	2005	https://socialsciences.mcmaster.ca/jfox/Misc/Rcmdr
RKward	2002	https://rkward.kde.org
Rattle GUI	2009	https://rattle.togaware.com
Bio7	2012	https://bio7.org
<i>Other IDEs</i>		
Jupyter Notebook	2014	https://jupyter.org
Eclipse (StatET)	2010	https://projects.eclipse.org/projects/science.statet
Google Colab	2017	https://colab.research.google.com
Visual Studio (VS)	2015	https://code.visualstudio.com
<i>Code</i>		
<i>Editores de texto</i>		
vi/Vim	1976	https://www.vim.org/download.php
Emacs ESS	1997	https://ess.r-project.org
Sublime Text	2008	https://www.sublimetext.com
Notepad++	2003	https://notepad-plus-plus.org/downloads
<i>Statistical frameworks</i>		
Jamovi	2017	https://www.jamovi.org
JASP	2016	https://jasp-stats.org

889
890
891
892

Tabela S2. Lista selecionada de livros didáticos essenciais para programação em R. Adaptado de Da Silva *et al.* (2022) e Giorgi *et al.* (2022).

Livro	Descrição
“Análises Ecológicas no R” by Fernando R. da Silva, Thiago Gonçalves-Souza, Gustavo B. Paterno, Diogo B. Provete, Maurício H. Vancine	Contém 15 capítulos que abrangem desde como baixar e instalar o R, passando por manipulação de dados, análise de diversidade, estatística básica, métodos multidimensionais e visualização de dados geoespaciais.
“Ciência de Dados com R: introdução” by Saulo Guerra, Paulo Felipe de Oliveira, Robert McDonnell e Sillas Gonzaga	Este livro foi criado para o curso de Introdução à Ciência de Dados com R, oferecido pelo Instituto Brasileiro de Pesquisa e Análise de Dados (IBPAD).
“Ciência de Dados em R” by Athos Damiani, Beatriz Milz, Caio Lente Daniel Falbel, Fernando Correa Julio Trecenti, Nicole Ludovice, Tereza Lacerda, William Amorim	O objetivo deste livro é guiá-lo por todas as etapas da análise de dados, utilizando o R como ferramenta. Ele contém ferramentas para importar, organizar, manipular, visualizar e modelar bancos de dados, além de ferramentas para comunicar e automatizar os resultados.
"Introdução à Linguagem R seus fundamentos e sua prática" by Pedro Duarte Faria	O objetivo deste livro é ensinar os fundamentos da linguagem R e como eles se aplicam à prática da análise de dados.
“R para Ciência de Dados” by Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund	Livro escrito pelo criador do ggplot2 e do tidyverse, que fornece conhecimento estruturado sobre o tidyverse, bem como soluções para operações de ciência de dados. Tradução para o português.
“Teaching Tech Together” by Greg Wilson	Oferece práticas baseadas em evidências para a criação e a execução de aulas eficazes e para a construção de uma comunidade de professores em torno delas.
“The R book” by Michael J. Crawley	Um ótimo livro para iniciantes, apresentando exemplos práticos e os fundamentos das estruturas e funções do R e do aprendizado de máquina.

“R for Data Science” by Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund	Este livro, escrito pelo criador do ggplot2 e do tidyverse, fornece conhecimento estruturado sobre o tidyverse, bem como soluções para operações de ciência de dados.
“Use R!” by Robert Gentleman, Kurt Hornik and Giovanni Parmigiani	Série de livros com quase 100 volumes dedicada ao uso prático e focado da linguagem de programação R.
“The R Series” by John Chambers, Torsten Hothorn, Duncan Temple Lang and Hadley Wickham	Outra coleção de livros sobre R com 62 títulos, abrangendo uma ampla gama de tópicos, desde conceitos de programação até aplicações do R.
“Highland Statistics Ltd. books” by Alain Zuur and Elena Ieno (https://www.highstat.com/index.php/books2)	Coleção de livros sobre a linguagem R, com 12 títulos, abrangendo tópicos iniciais de programação em R, mas principalmente tópicos em modelos estatísticos, como: GLM, GAM, GLMM, GAMM e modelos com inflação de zeros.

Tabela S3. Lista selecionada dos principais recursos para aprender a programar em R e outras ferramentas.

Tipos de recursos	Referências
Código no R	Ihaka & Gentleman (1996), Tippmann (2015), Lai <i>et al.</i> (2019), Giorgi <i>et al.</i> (2022)
Ensino de R	Barraquand <i>et al.</i> (2014), Touchon & McCoy (2016), Johnston <i>et al.</i> (2019), Auker & Barthelmess (2020), Lawlor <i>et al.</i> (2022), Tucker <i>et al.</i> (2022), Hoffman & Wright (2024)
Pacotes no R	Lortie <i>et al.</i> (2020), Bivand (2021), Hesselbarth <i>et al.</i> (2021), Lai <i>et al.</i> (2023), Sillero <i>et al.</i> (2023)
Gráficos no R	Wickham (2010), Nordmann <i>et al.</i> (2022), The R Graph Gallery (https://r-graph-gallery.com), from Data to Viz (https://www.data-to-viz.com), The Data Visualisation Catalogue (https://datavizcatalogue.com/index.html)
R Markdown	Ben Baumer <i>et al.</i> (2014), Benjamin Baumer & Udwin (2015), Bauer (2018)
Comunidade R	Fox (2009), Mair <i>et al.</i> (2015), Bollmann <i>et al.</i> (2017)
tidyverse	Wickham (2014), Wickham <i>et al.</i> (2019), Carscadden & Martin (2022), Çetinkaya-Rundel <i>et al.</i> (2022)
Código limpo	Filazzola & Lortie (2022)
Reprodutibilidade	Figueiredo <i>et al.</i> (2022), Kita <i>et al.</i> (2022), Keller <i>et al.</i> (2023)
Estatística	Schielzeth (2010), Freeman <i>et al.</i> (2014), Greenland <i>et al.</i> (2016), Touchon & McCoy (2016), Gonçalves-Souza <i>et al.</i> (2019), Feng <i>et al.</i> (2020), Hagan <i>et al.</i> (2020), LaTourrette <i>et al.</i> (2021), Tredennick <i>et al.</i> (2021), Juavinett (2022), Cuddington <i>et al.</i> (2023), Hendry (2023)
Ilustrações	Allison Horst (https://github.com/allisonhorst/stats-illustrations)
git/GitHub	Perez-Riverol <i>et al.</i> (2016), Perkel (2016), Beckman <i>et al.</i> (2021), Braga <i>et al.</i> (2023)

Tabela S4. Tipos de avaliação para o ensino superior e os mais associados às aulas de programação em R na área de Ecologia. Os tipos de avaliação foram adaptados das diretrizes de ensino da Universidade Estadual Paulista (Unesp), no Brasil, e da Universidade Massey, na Nova Zelândia.

Tipo de avaliação	Descrição	Aplicações em R para a ecologia
Programas de computador	Animação e exibição por computador, design, programação, modelagem e outros trabalhos de informática.	Código ou exercício personalizado para a aplicação das análises ecológicas mais comuns.
Composições criativas	Animações, filmes, modelos, têxteis, sites e outras composições.	Utilizando o R para criar gráficos ou mapas espaço-temporais.
Exames	Uma avaliação agendada pelo professor. A avaliação pode ser presencial, on-line, oral, prática, de campo, escrita ou em outro formato. As informações sobre data e local da avaliação podem ser disponibilizadas por meio de um portal on-line para estudantes (Moodle, Stream), mas as datas	Testes de lógica ou exames tradicionais que utilizam termos funcionais ou o significado biológico para manipular dados de uma maneira específica ou interpretar os resultados de uma análise.

	geralmente são confirmadas durante as aulas.	
Apresentações orais	Debates, demonstrações, exposições, entrevistas, propostas orais, dramatizações, discursos e outras apresentações.	Apresentações sobre o fluxo de trabalho para a investigação de um projeto.
Participação	O aluno poderá ser avaliado com base em sua participação em atividades como discussões on-line (fóruns), laboratórios, debates, tutoriais, exercícios e seminários.	Discussões on-line, participação em aulas ou em atividades em grupo.
Portfólio	Portfólios criativos, de aprendizagem, on-line, narrativos, fotográficos, escritos e outros.	Relatórios em Markdown.
Prática	Viagens de estudo, trabalho de campo, visitas, estágios, seminários, workshops, trabalho voluntário e outras atividades.	Eventos e workshops sobre R.
Simulação	Simulações baseadas em tecnologia ou em experiência.	Utilizando ou desenvolvendo código baseado em simulações.
Teste	Testes laboratoriais, on-line, de múltipla escolha, de resposta curta, orais e outros tipos de testes.	Testes on-line ou presenciais, ou testes criados em R.
Trabalho escrito	Ensaio, projetos em grupo ou individuais, propostas, relatórios, resenhas, exercícios de escrita e outras tarefas escritas..	Tarefas em projetos que utilizam R para análise de dados. Tarefas de programação com o auxílio de inteligência artificial.

Referências

- Auker, L. A., & Barthelmess, E. L. 2020. Teaching R in the undergraduate ecology classroom: approaches, lessons learned, and recommendations. *Ecosphere*, 11(4). DOI: 10.1002/ecs2.3060
- Barraquand, F., Ezard, T. H. G., Jørgensen, P. S., Zimmerman, N., Chamberlain, S., Salguero-Gómez, R., Curran, T. J., & Poisot, T. 2014. Lack of quantitative training among early-career ecologists: a survey of the problem and potential solutions. *PeerJ*, 2, e285. DOI: 10.7717/peerj.285
- Bauer, P. C. 2018. Writing a reproducible paper in R Markdown., 17.
- Baumer, Ben, Cetinkaya-Rundel, M., Bray, A., Loi, L., & Horton, N. J. 2014. R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics.
- Baumer, Benjamin, & Udwin, D. 2015. R Markdown. *WIREs Computational Statistics*, 7(3), 167–177. DOI: 10.1002/wics.1348
- Beckman, M. D., Çetinkaya-Rundel, M., Horton, N. J., Rundel, C. W., Sullivan, A. J., & Tackett, M. 2021. Implementing Version Control With Git and GitHub as a Learning

Objective in Statistics and Data Science Courses. *Journal of Statistics and Data Science Education*, 29(sup1), S132–S144. DOI: 10.1080/10691898.2020.1848485

Bivand, R. S. 2021. Progress in the R ecosystem for representing and handling spatial data. *Journal of Geographical Systems*, 23(4), 515–546. DOI: 10.1007/s10109-020-00336-0

Bollmann, S. S., Cook, D., Dumas, J., Fox, J., Josse, J., Keyes, O., Strobl, C., Turner, H., & Debelak, R. 2017. A First Survey on the Diversity of the R Community. Version 2. *The R Journal*, 9(2), 542–551. DOI: 10.5167/uzh-233651

Braga, P. H. P., Hébert, K., Hudgins, E. J., Scott, E. R., Edwards, B. P. M., Sánchez Reyes, L. L., Grainger, M. J., Foroughirad, V., Hillemann, F., Binley, A. D., Brookson, C. B., Gaynor, K. M., Shafiei Sabet, S., Güncan, A., Weierbach, H., Gomes, D. G. E., & Crystal-Ornelas, R. 2023. Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution. *Methods in Ecology and Evolution*, 14(6), 1364–1380. DOI: 10.1111/2041-210X.14108

Broman, K. W., & Woo, K. H. 2018. Data organization in spreadsheets. *The American Statistician*, 72(1), 2–10. DOI: 10.1080/00031305.2017.1375989

Carscadden, K., & Martin, A. 2022. To Tidy or not When Teaching R Skills in Biology Classes. Version 5. *International Journal of Higher Education*, 11(5), 39. DOI: 10.5430/ijhe.v11n5p39

Çetinkaya-Rundel, M., Hardin, J., Baumer, B. S., McNamara, A., Horton, N. J., & Rundel, C. 2022. An educator’s perspective of the tidyverse. *Technology Innovations in Statistics Education*, 14(1). DOI: 10.5070/T514154352

Cuddington, K., Abbott, K. C., Adler, F. R., Aydeniz, M., Dale, R., Gross, L. J., Hastings, A., Hobson, E. A., Karatayev, V. A., Killion, A., Madamanchi, A., Marraffini, M. L., McCombs, A. L., Samyono, W., Shiu, S.-H., Watanabe, K. H., & White, E. R. 2023. Challenges and opportunities to build quantitative self-confidence in biologists. *BioScience*, 73(5), 364–375. DOI: 10.1093/biosci/biad015

945 Da Silva, F. R., Gonçalves-Souza, T., Paterno, G. B., Provete, D. B., & Vancine, M. H. 2022.
 946 Análises Ecológicas no R. PE: NUPEEA: p. 640.

947 Feng, X., Qiao, H., & Enquist, B. J. 2020. Doubling demands in programming skills call for
 948 ecoinformatics education. *Frontiers in Ecology and the Environment*, 18(3), 123–124. DOI:
 949 10.1002/fee.2179

950 Figueiredo, L., Scherer, C., & Cabral, J. S. 2022. A simple kit to use computational notebooks
 951 for more openness, reproducibility, and productivity in research. *PLOS Computational*
 952 *Biology*, 18(9), e1010356. DOI: 10.1371/journal.pcbi.1010356

953 Filazzola, A., & Lortie, C. 2022. A call for clean code to effectively communicate science.
 954 *Methods in Ecology and Evolution*, 2041–210X.13961. DOI: 10.1111/2041-210X.13961

955 Fox, J. 2009. Aspects of the Social Organization and Trajectory of the R Project. *The R Journal*,
 956 1(2), 5–13. DOI: 10.32614/RJ-2009-014

957 Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., &
 958 Wenderoth, M. P. 2014. Active learning increases student performance in science,
 959 engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23),
 960 8410–8415. DOI: 10.1073/pnas.1319030111

961 Giorgi, F. M., Ceraolo, C., & Mercatelli, D. 2022. The R Language: An Engine for
 962 Bioinformatics and Data Science. Version 5. *Life*, 12(5), 648. DOI: 10.3390/life12050648

963 Gonçalves-Souza, T., Provete, D. B., Garey, M. V., da Silva, F. R., & Albuquerque, U. P. 2019.
 964 Going Back to Basics: How to Master the Art of Making Scientifically Sound Questions. In:
 965 U. P. Albuquerque, R. F. P. de Lucena, L. V. F. Cruz da Cunha, & R. R. N. Alves (Eds.),
 966 *Methods and Techniques in Ethnobiology and Ethnoecology*. pp. 71–86. New York, NY:
 967 Springer New York.

968 Greenland, S., Senn, S. J., Rothman, K. J., Carlin, J. B., Poole, C., Goodman, S. N., & Altman,
 969 D. G. 2016. Statistical tests, P values, confidence intervals, and power: a guide to
 970 misinterpretations. *European Journal of Epidemiology*, 31(4), 337–350. DOI:
 971 10.1007/s10654-016-0149-3

972 Hagan, A. K., Lesniak, N. A., Balunas, M. J., Bishop, L., Close, W. L., Doherty, M. D., Elmore,
 973 A. G., Flynn, K. J., Hannigan, G. D., Koumpouras, C. C., Jenior, M. L., Kozik, A. J.,
 974 McBride, K., Rifkin, S. B., Stough, J. M. A., Sovacool, K. L., Sze, M. A., Tomkovich, S.,
 975 Topcuoglu, B. D., & Schloss, P. D. 2020. Ten simple rules to increase computational skills
 976 among biologists with Code Clubs. *PLOS Computational Biology*, 16(8), e1008119. DOI:
 977 10.1371/journal.pcbi.1008119
 978 Hendry, A. P. 2023. Prediction in ecology and evolution. *BioScience*, biad083. DOI:
 979 10.1093/biosci/biad083
 980 Hesselbarth, M. H. K., Nowosad, J., Signer, J., & Graham, L. J. 2021. Open-source Tools in R
 981 for Landscape Ecology. *Current Landscape Ecology Reports*, 6(3), 97–111. DOI:
 982 10.1007/s40823-021-00067-y
 983 Hoffman, A. M., & Wright, C. 2024. Ten simple rules for teaching an introduction to R. *PLOS*
 984 *Computational Biology*, 20(5), e1012018. DOI: 10.1371/journal.pcbi.1012018
 985 Ihaka, R., & Gentleman, R. 1996. R: A Language for Data Analysis and Graphics. *Journal of*
 986 *Computational and Graphical Statistics*, 5(3), 299. DOI: 10.2307/1390807
 987 Johnston, L. W., Bonsma-Fisher, M., Ostblom, J., Hasan, A. R., Santangelo, J. S., Coome, L.,
 988 Tran, L., Andrade, E. S. de, & Mahallati, S. 2019. A graduate student-led participatory live-
 989 coding quantitative methods course in R: Experiences on initiating, developing, and
 990 teaching. *Journal of Open Source Education*, 2(16), 49. DOI: 10.21105/jose.00049
 991 Juavinett, A. L. 2022. The next generation of neuroscientists needs to learn how to code, and we
 992 need new ways to teach them. *Neuron*, 110(4), 576–578. DOI: 10.1016/j.neuron.2021.12.001
 993 Keller, A., Ankenbrand, M. J., Bruelheide, H., Dekeyzer, S., Enquist, B. J., Erfanian, M. B.,
 994 Falster, D. S., Gallagher, R. V., Hammock, J., Kattge, J., Leonhardt, S. D., Madin, J. S.,
 995 Maitner, B., Neyret, M., Onstein, R. E., Pearse, W. D., Poelen, J. H., Salguero-Gomez, R.,
 996 Schneider, F. D., Tóth, A. B., & Penone, C. 2023. Ten (mostly) simple rules to future-proof
 997 trait data in ecological and evolutionary sciences. *Methods in Ecology and Evolution*, 14(2),
 998 444–458. DOI: 10.1111/2041-210X.14033

999 Kita, C. A., Florez-Montero, G., Montoya-Bustamante, S., Muylaert, R. L., Zapata-Mesa, N., &
1000 Mello, M. A. R. 2022. Ten simple rules for reporting information on species interactions.
1001 PLOS Computational Biology, 18(8), e1010362. DOI: 10.1371/journal.pcbi.1010362
1002 Lai, J., Cui, D., Zhu, W., & Mao, L. 2023. The Use of R and R Packages in Biodiversity
1003 Conservation Research. Diversity, 15(12), 1202. DOI: 10.3390/d15121202
1004 Lai, J., Lortie, C. J., Muenchen, R. A., Yang, J., & Ma, K. 2019. Evaluating the popularity of R
1005 in ecology. Ecosphere, 10(1), e02567. DOI: 10.1002/ecs2.2567
1006 LaTourrette, K., Stengel, A., & Clarke, J. 2021. Student-led workshops: Filling skills gaps in
1007 computational research for life scientists. Natural Sciences Education, 50(1), e20052. DOI:
1008 10.1002/nse2.20052
1009 Lawlor, J., Banville, F., Forero-Muñoz, N.-R., Hébert, K., Martínez-Lanfranco, J. A., Rogy, P.,
1010 & MacDonald, A. A. M. 2022. Ten simple rules for teaching yourself R. PLOS
1011 Computational Biology, 18(9), e1010372. DOI: 10.1371/journal.pcbi.1010372
1012 Lortie, C. J., Braun, J., Filazzola, A., & Miguel, F. 2020. A checklist for choosing between R
1013 packages in ecology and evolution. Ecology and Evolution, 10(3), 1098–1105. DOI:
1014 10.1002/ece3.5970
1015 Mair, P., Hofmann, E., Gruber, K., Hatzinger, R., Zeileis, A., & Hornik, K. 2015. Motivation,
1016 values, and work design as drivers of participation in the R open source project for statistical
1017 computing. Proceedings of the National Academy of Sciences, 112(48), 14788–14792. DOI:
1018 10.1073/pnas.1506047112
1019 Nordmann, E., McAleer, P., Toivo, W., Paterson, H., & DeBruine, L. M. 2022. Data
1020 Visualization Using R for Researchers Who Do Not Use R. Advances in Methods and
1021 Practices in Psychological Science, 5(2), 25152459221074654. DOI:
1022 10.1177/25152459221074654
1023 Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V.,
1024 Fufezan, C., Ternent, T., Eglen, S. J., Katz, D. S., Pollard, T. J., Konovalov, A., Flight, R.
1025 M., Blin, K., & Vizcaíno, J. A. 2016. Ten Simple Rules for Taking Advantage of Git and

1026 GitHub. PLOS Computational Biology, 12(7), e1004947. DOI:
 1027 10.1371/journal.pcbi.1004947
 1028 Perkel, J. 2016. Democratic databases: science on GitHub. *Nature*, 538(7623), 127–128. DOI:
 1029 10.1038/538127a
 1030 Schielzeth, H. 2010. Simple means to improve the interpretability of regression coefficients.
 1031 *Methods in Ecology and Evolution*, 1(2), 103–113. DOI: 10.1111/j.2041-210X.2010.00012.x
 1032 Sillero, N., Campos, J. C., Arenas-Castro, S., & Barbosa, A. M. 2023. A curated list of R
 1033 packages for ecological niche modelling. *Ecological Modelling*, 476, 110242. DOI:
 1034 10.1016/j.ecolmodel.2022.110242
 1035 Tippmann, S. 2015. Programming tools: Adventures with R. Version 7532. *Nature*, 517(7532),
 1036 109–110. DOI: 10.1038/517109a
 1037 Touchon, J. C., & McCoy, M. W. 2016. The mismatch between current statistical practice and
 1038 doctoral training in ecology. *Ecosphere*, 7(8), e01394. DOI: 10.1002/ecs2.1394
 1039 Tredennick, A. T., Hooker, G., Ellner, S. P., & Adler, P. B. 2021. A practical guide to selecting
 1040 models for exploration, inference, and prediction in ecology. *Ecology*, 102(6). DOI:
 1041 10.1002/ecy.3336
 1042 Tucker, M. C., Shaw, S. T., Son, J. Y., & Stigler, J. W. 2022. Teaching Statistics and Data
 1043 Analysis with R. *Journal of Statistics and Data Science Education*, 0(0), 1–15. DOI:
 1044 10.1080/26939169.2022.2089410
 1045 Wickham, H. 2010. A Layered Grammar of Graphics. *Journal of Computational and Graphical*
 1046 *Statistics*, 19(1), 3–28. DOI: 10.1198/jcgs.2009.07098
 1047 Wickham, H. 2014. Tidy Data. *Journal of Statistical Software*, 59(10). DOI:
 1048 10.18637/jss.v059.i10
 1049 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Golemund, G.,
 1050 Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K.,
 1051 Ooms, J., Robinson, D., Seidel, D., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo,

1052 K., & Yutani, H. 2019. Welcome to the Tidyverse. *Journal of Open Source Software*, 4(43),
1053 1686. DOI: 10.21105/joss.01686