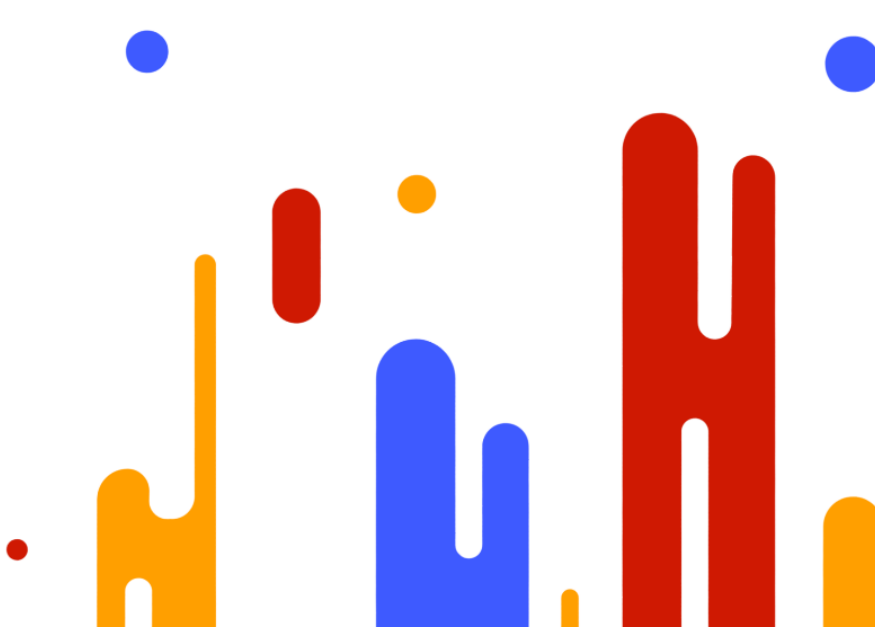


# Pacotes do R para criação de objetos de aprendizagem ativos em ensino reprodutível

José Maurício Schneedorf Ferreira da Silva



10.47247/LSCC/6063.145.8.6



**V&V Editora**

Santo André, São Paulo – Brasil

Tel./Whatsapp: (11) 94019-0635

E-mail: contato@vveditora.com

vveditora.com

@vveditora

**Expediente**

Coordenação Editorial: Marilena Rosalen

Coordenação de Área: Everton Viesba

Revisão: Letícia Moreira Viesba

Edição: Everton Viesba

Equipe editorial: Marco André, Larisse Maia Rodrigues, Juliana Bastos, Andressa Souza.

**Conselho Editorial**

Profa. Dra. Marilena Rosalen

Profa. Dra. Angela Martins Baeder

Profa. Dra. Eunice Nunes

Profa. Dra. Luciana A. Farias

Profa. Dra. Maria Célia S. Gonçalves

Profa. Dra. Rita C. Borges M. Amaral

Profa. Dra. Silvana Pasetto

Profa. Ma. Beatriz Milz

Profa. Ma. Marta Angela Marcondes

Prof. Dr. José Guilherme Franchi

Prof. Dr. Luiz Afonso V. Figueiredo

Prof. Dr. Flávio José M. Gonçalves

Prof. Dr. Giovano Candiani

Prof. Me. Arnaldo Silva Junior

Prof. Me. Pedro L. Castrillo Yagüe

Prof. Me. Everton Viesba-Garcia

Profa. Ma. Letícia Moreira Viesba

Profa. Ma. Erika Brunelli

**Organização**

Organização: Lara Santana Correia Costa e Claudinei Zagui Pareschi

**Parecer e revisão por pares**

Os textos que compõem esta obra foram submetidos para avaliação da Coordenação e/ou Conselho Editorial da V&V Editora, sendo aprovados na revisão por pares para publicação.

**Dados Internacionais de Catalogação na Publicação (CIP)**

x      Aprendizagem Ativa: caminhos para o protagonismo, a inovação e a construção do conhecimento. Lara Santana Correia Costa e Claudinei Zagui Pareschi (organizadores) – Santo André: V&V Editora, 2026.  
198 p. : 16 x 23 cm

Inclui bibliografia

ISBN 978-65-6063-145-8

DOI 10.47247/LSCC/6063.145.8

1. Um. 2. Dois. I. Três. II. Título.

CDD x

Elaborado por Maurício Amormino Júnior – CRB6/2422

## Ensino reprodutível (ER)

Um dos agravos bem conhecidos dos processos de ensino e aprendizagem envolve a transmissão adaptável de informações presentes em conteúdos didáticos. Não importa se com giz, marcador de quadro, ou lousa digital, os conteúdos são transmitidos periodicamente às turmas, por vezes às expensas únicas dos recursos mnemômicos do educador.

Um exemplo: o(a) professor(a) elabora uma aula com auxílio de pranchas de um aplicativo de apresentação (*slides*), e o transmite à turma com projetor multimídia. Ao semestre/ano seguinte ele *replica* a apresentação, ou a reproduz com alguma adaptação em texto/imagens das pranchas. Se por um lado esse aspecto permite personalizar o material didático, por outro arrisca engessar o mesmo face à dificuldades envolvidas em sua construção.

Ilustrando essas dificuldades: a nova versão do aplicativo poderá não aceitar o conteúdo da versão antiga, algo comum em aplicativos em linguagem Java e mais ainda em Macromedia/Adobe Shockwave Flash. Além disso, as novas imagens terão que ser ajustadas em tamanho/resolução/posicionamento adequados para inserção nas pranchas originais, assim como para novos subtítulos e ajustes em fonte/tamanho.

Nessa perspectiva, o *ensino reprodutível* (ER) vem paulatinamente ganhando força na literatura como uma solução de contorno potencialmente eficaz. O ER consiste numa abordagem ativa em ensino e aprendizagem envolvendo linguagem de programação, e que busca contemplar materiais didáticos com reprodutibilidade e adaptação voltados à educação. Em uma revisão recente intitulada “*Reproducibility in the Classroom*” (Dogucu, 2024), Mine conceitua e diferencia a reprodutibilidade da mera replicabilidade, e tece recomendações práticas para a incorporação da primeira.

Já para Ostblom e Timbers, faz-se primaz uma motivação para a importância de reprodutibilidade no ER mediada por colaboração, confiança e integridade científica (Ostblom; Timbers, 2022). Além disso, os autores também propõem o uso de projetos que se identifiquem com a realidade, bem como o uso de instrução guiada. E para essa, sugerem ferramentas e práticas reprodutíveis, como controle de versão com *Git*, uso de *Notebooks* interativos (sistema para códigos e texto, não o equipamento) e documentação adequada do algoritmo.

Num trabalho envolvendo a construção de projetos reprodutíveis em um curso introdutório de ciência de dados, Bean discorre sobre o histórico de ferramentas ao ER, desde a *literacia em programação* (alfabetização digital) e a programação letrada (Knuth, 1984) até o uso de *documentos dinâmicos*, como *RMarkdown* e *Jupyter Notebooks*. Em outro trabalho o autor discute os desafios e benefícios de implementar essa abordagem ao ensino superior (Bean, 2023),

elencando a linguagem R para organização de código, documentação e versionamento controlados.

## R & RStudio

R (R Core Team, 2023) consiste numa linguagem de programação desenvolvida nos anos 1990 na Universidade de Auckland (Nova Zelândia) como um ambiente de *software* gratuito para computação estatística e gráficos. Baseada na linguagem S previamente desenvolvida nos laboratórios Bell nos anos 1970, o R conta hoje com uma larga extensibilidade de *pacotes* para as mais variadas finalidades. Como a linguagem “roda” em um terminal simples, como o *CMD* do Windows ou *Terminal* do Linux, diversos ambientes gráficos foram criados para facilitar seu uso e apropriação técnica.

Entre esses *IDEs* (*Integrated Development Environment*) sobressaiu-se de modo exponencial nos últimos anos o ambiente *RStudio* (*RStudio Team, 2023*), originalmente criado por empresa homônima, e hoje sob a aprimoramento constante pela *Posit*, essa voltada a soluções abertas em *ciência de dados*.

## Objetos de aprendizagem em ER com R & RStudio

Entre os quase 22 mil pacotes atuais disponíveis no [site oficial do R](#), alguns possuem características passíveis para utilização em ensino-aprendizagem, como a produção de objetos didáticos (*LO, learning objects*; (Convertini *et al.*, 2006)). Dessa forma, pode-se empregar a plataforma R & RStudio para o desenvolvimento de trechos de códigos que, quando compilados, possam resultar em uma animação, simulação, gráfico ou mapa por exemplo. Adicionalmente, alguns pacotes possuem recursos adicionais para interatividade, essa resultante de uma ação direta do usuário sobre o objeto didático compilado, mais do que sobre o código utilizado.

Em repositórios de plataformas para gestão de conteúdo (*LCMS, Learning Content Management System*), tais objetos recebem uma classificação taxonômica de objetos de aprendizagem (*LO, Learning Objects*). Quando a interatividade decorre da ação do aprendiz sobre o objeto propriamente dito, ou ainda sobre o trecho de código, a categoria específica é denominada *S-Active*, ou *Self-Active*, sem dependência de colaboração ou comunicação com outros usuários (*taxonomia de OSEL*; Convertini *et al.* (2006)). Nesse caso, a diferença está apenas no tipo de objeto de aprendizagem, se passivo (*Generative LO*; uma animação, por ex) ou ativo (*Interactive LO*; um gráfico modificado por seleção de ítem num menu, por ex). Em ambos os casos, contudo, permanece a categoria *S-Active*, uma vez que o material permite que o usuário altere o código e/ou o objeto de aprendizagem.

## Pacotes de R para objetos ativos de aprendizagem

Dessa forma, considerando a taxonomia italiana de OSEL acima (Convertini *et al.*, 2006), gráficos, mapas e tabelas estáticos, bem como animações, simulações e cálculos criados/modificados por trechos de códigos em R, são considerados *objetos de aprendizagem generativos S-ativos (generative LO:S-Active)*. Por outro lado, quando esses mesmos objetos permitem alterações em si próprios, independentemente do uso da linguagem de programação ou biblioteca específica, tal como a necessidade de RStudio à demonstração do objeto, ou um arquivo HTML que não o necessite (*self-contained*), são então classificados como *objetos de aprendizagem interativos S-ativos (interactive LO:S-Active)*. No exemplo acima, arquivos HTML podem “rodar” de forma independente do programa devido à incorporação de elementos e bibliotecas específicas da linguagem Markdown que os compõe (JavaScript, por ex). À guisa de explicação, embora similares, um pacote difere de uma biblioteca por esta última consistir no conjunto de arquivos instalados em máquina, enquanto a primeira retrata o(s) arquivo(s) comprimido(s) necessários à instalação.

Entre as dezenas de milhares de pacotes do R, alguns apresentam ambos os potenciais, ou seja, tanto para criação de objetos generativos, como para objetos interativos. A [Figura 1](#) exemplifica alguns desses pacotes, para animações, gráficos, e tabelas.

Figura 1: Pacotes R para gráficos, tabelas, simulações e animações (estáticos ou interativos).

Pacotes R para gráficos, tabelas, simulações e animações (estáticos ou interativos)

Tipo e Pacotes		
Categoria	Tipo	Pacotes
Estático	Gráfico	ggplot2, lattice, graphics, grid, cowplot, patchwork, RColorBrewer, ggpubr
	Tabela	kableExtra, xtable, gt, huxtable, flextable
	Simulação	deSolve, markovchain, MASS, boot, SimDesign
	Animação	ganimate, animation
Interativo	Gráfico	plotly, leaflet, highcharter, echarts4r, dygraphs, googleVis, threejs, rgl
	Tabela	DT, reactable, formattable, shinyTable, handsontable
	Simulação	manipulate, shiny, shinydashboard, learnr, flexdashboard
	Animação	r2d3, htmlwidgets

## O website Bioquanti e objetos de aprendizagem ativos

Para as bibliotecas mencionadas neste capítulo e outras tantas, convida-se o leitor a uma “*passada de olho*” no website educacional [Bioquanti](#). Trata-se de um portal para diversos objetos de aprendizagem ativos em ensino reproduzível, e originalmente desenvolvido para o *ensino superior*, e com foco em *Bioquímica*, para gráficos, tabelas, cálculos, diagramas, simulações, e outros objetos didáticos de que trata a disciplina.

E como *Bioquanti* traz em sua contração lexical um aspecto *quantitativo* de abordagem, o [Bioquanti](#) insere alguns conteúdos aplicados em temas vistos também

em *Matemática, Físico Química e Química*, tais como *cálculo integral e diferencial, planejamento estatístico, análise estatística (descritiva e inferencial), modelagem linear e curvilínea de dados (incluindo ajuste não linear), e álgebra linear*, entre outros (Schneedorf, 2023). Atualmente o *Bioquanti* tem esboçado alguns objetos de aprendizagem ativos também para conteúdos do *ensino básico*, e nas áreas de *Ciências da Natureza, Matemática, Ciências Humanas e Linguagens*.

## Ilustrando alguns objetos de aprendizagem ativos

Dos pacotes elencados na [Figura 1](#), alguns se adequam mais facilmente ao ensino-aprendizagem de conteúdos escolares e universitários, tanto pela sintaxe facilitada e trecho de código reduzido, como pelo alvo normalmente envolvido (gráfico, simulação, tabela, mapa). Neste sentido abordaremos com algum detalhe uns poucos, listados abaixo:

1. Animação: `anim.plots`, `gganimate`
2. Simulação: `manipulate`
3. Gráfico interativo: `plotly`
4. Mapa interativo: `leaflet`

Essas bibliotecas instaladas possuem características distintas quanto ao ambiente interativo em que são geradas, bem como o seu produto final. Parte delas requer a plataforma *R & RStudio* para a visualização desse (`manipulate`). Outras o tem como uma sobreposição de imagens gerando uma animação, mas que não requer *R* nem *RStudio* para sua demonstração (`anim.plots`, `png animado`; `gganimate`, `gif animado`). E as demais também não requerem a plataforma para visualização, além de permitir ação direta do usuário junto ao objeto compilado, e em arquivo *HTML* autossuficiente (`plotly`, `DT`, `leaflet`, `flexdashboard`), e dessa forma permitindo fácil compartilhamento para fins educacionais.

As bibliotecas são panoramicamente apresentadas a seguir. Optou-se, por uma questão prática e de limitação deste texto, a apresentá-las com nome, descrição curta, um exemplo para o ensino básico ou para o ensino superior, e fonte de consulta. Também por limitação de espaço, os conceitos envolvidos nos exemplos devem ser buscados em texto técnicos. Complementarmente, optou-se por ilustrar trechos curtos de código, sem configurações extras ou alternativas, de modo a facilitar sua apresentação, simplificar os passos de execução, e estimular o usuário à mudanças pontuais pra um novo objeto didático, “*alma*” do *ensino reproduzível*. Não obstante, a *web* é fartamente ilustrada com inúmeros exemplos de gráficos, tabelas, simulações e animações, tanto para os pacotes exemplificados a seguir, como os demais elencados à [Figura 1](#).

## Compilando e “rodando” trechos de código

Há duas formas básicas para se compilar o código e observar o objeto de aprendizagem: ou por R & RStudio instalados em computador, ou em nuvem. Nesse caso, pode-se tentar alguns provedores que oferecem a linguagem R para compilação remota, tais como o site do mantenedor do [RStudio](#) para computação em nuvem.

Para executar um código, também existem várias maneiras. Segue uma simples:

1. Copiar o código junto ao ícone de pasta ao canto superior direito do trecho;
2. Abrir um `*script*` no `*RStudio*`: `*File --> New File --> R Script*`;
3. Colar o código no `*script*` aberto no `*RStudio*`;
4. Compilar o código: `*Ctrl+A --> Ctrl Enter*`

## Animação

### Pacote `anim.plots`

A biblioteca [anim.plots](#) (Hugh-Jones, 2022) permite uma animação de gráficos simples criados com o pacote `graphics` da instalação base do R. Algumas animações são visualizadas no [site](#) do desenvolvedor. O objeto de aprendizagem apresenta-se ativo em função da escrita/modificação do código, embora não seja interativo, e materializa-se numa sequência animada de gráficos do pacote básico de R. Por tratar-se de animação, não é passível incluí-la num documento convencional como este, embora o arquivo gerado possa ser armazenado e acessado como *GIF* (padrão), *SWF*, *HTML*, or *LaTeX*.

### Efeito da temperatura na equação de Clapeyron (Gases Ideais)

Segue um exemplo envolvendo o efeito da alteração de temperatura na *Lei de Gases Ideais*.

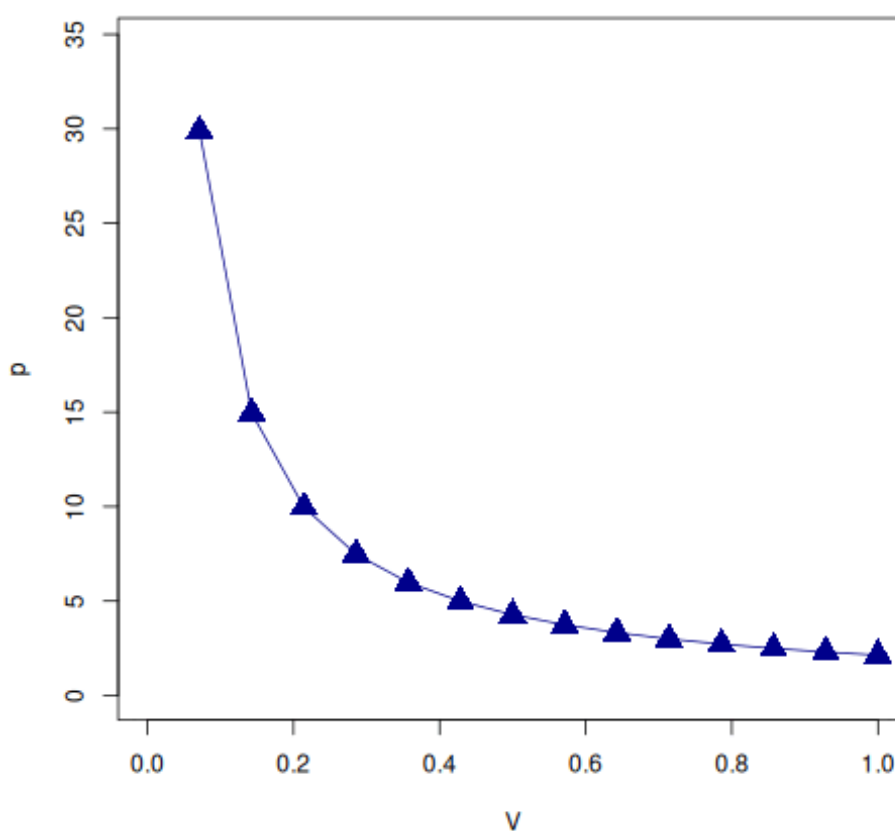
```
# Pacote necessário; instala automaticamente se tiver
if (!requireNamespace("anim.plots", quietly = TRUE))
  install.packages("anim.plots")

library(anim.plots)
# Definição dos parâmetros
n <- 1      # Número de mols
R <- 0.0821 # Constante universal dos gases (em L·atm/(mol·K))
t <- 300    # Temperatura inicial (em Kelvin)

anim.gas.ideal <- anim.curve(n*R*t/x, # equação de gás ideal
```

```
times=1:30, n=15, type="o", # repetição de quadros, no. de pontos,  
e tipo de gráfico  
col="darkblue", cex=2, pch=17, # cor, tamanho e tipo de símbolo  
xlab="V", ylab="p", # etiquetas de eixos  
speed = 5) # velocidade de animação)  
  
anim.save(anim.gas.ideal, "anim.gas.ideal.gif") # salvamento de arquivo
```

Figura 2: Representação estática de animação de gases ideais conduzida com pacote `anim.plots`. Para observar a animação, clique neste [link](#).



A [Figura 2](#) ilustra de modo estático a animação oferecida com o `chunk` (trecho de código) acima. Se desejar observar a animação propriamente dita, clique no `link` indicado na legenda da figura. Como curiosidade, experimente modificar alguma configuração do gráfico (cor, símbolo, no. de quadros, velocidade) ou parâmetro da equação (no. de mols, temperatura inicial), e compile o objeto para uma produto diferente (“*elã vital*” do ensino reproduzível).

## Pacote gganimate

A biblioteca `gganimate` é visualmente mais sofisticada que `anim.plots`, e atua somente sobre outra biblioteca instalada, `ggplot2`, o sistema de produção de gráficos com qualidade de publicação técnico-científica mais utilizado pelos *amantes* do R. O `ggplot2` é baseado na *gramática de gráficos* de Wilkinson (Wilkinson, 2012), que preconiza a elaboração gráfica ideal como uma deposição em camadas (tema, coordenadas, tipos de gráfico, estatísticas, elementos estéticos). Além disso, o `gganimate` possui um conjunto bem maior de controles para animação, quando comparado ao `anim.plots` (Pederse, 2025).

## Ordem de uma reação química

O *chunk* a seguir ilustra uma animação com `gganimate` para a produção de um gráfico cujo perfil é alterado em função do valor de ordem da reação.

```
# 1. Instalação facultativa dos pacotes necessários (só instala se não
tiver)

if (!requireNamespace("gganimate", quietly = TRUE))
install.packages("gganimate") # pacote de animação
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2") #
pacote para gráfico ggplot2
if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dplyr") #
pacote para manipulação de vetores (como uma coluna de Excel)
if (!requireNamespace("magrittr", quietly = TRUE)) install.packages("magrittr") #
pacote para uso do operador "pipe" %>%, para legibilidade e organização do
código

# 2. Carregamento dos pacotes
library(ggplot2)
library(gganimate)
library(dplyr)
library(magrittr)

# 3. Geração dos dados para o gráfico
k <- 1
Reagente <- seq(0, 1, length.out = 100) # cria um vetor para reagente
ordens <- seq(1, 2, by = 0.01) # cria um vetor para a ordem de reação
dados <- expand.grid(Reagente = Reagente, Ordem = ordens) %>% # interpola
valores dos gr
mutate(Velocidade = k * Reagente^Ordem)
```

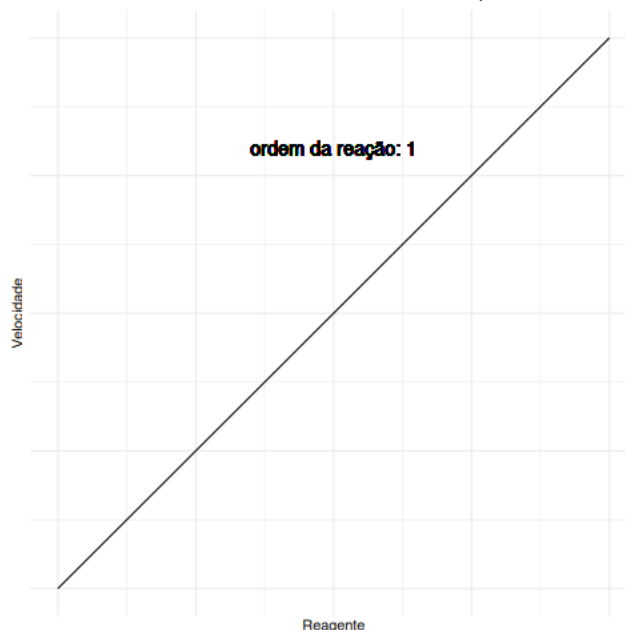
```

# Gráfico estático (constrói um gráfico com o pacote `ggplot2` necessário ao
`gganimate`)
ggplot(dados, aes(x = Reagente, y = Velocidade, group = Ordem)) + # estabelece
gráfico
  geom_line() + # acrescenta uma linha
  geom_text(aes(x = 0.5, y = max(Velocidade) * 0.8, # acrescenta e posiciona o
texto
  label = paste("ordem da reação:", round(Ordem, 1))), size = 5) + #
apresenta a ordem de reação em cada quadro
  transition_states(Ordem) + # aplica a animação
  theme_minimal() # seleciona um tema simples

anim_save("ordem_reacao.gif") # salva o gráfico como um GIF animado

```

Figura 3: Representação de uma animação com o pacote `gganimate`, para ordem de reação química. Como acima, para observar a animação, clique neste [link](#).



A [Figura 3](#) ilustra estaticamente o gráfico final produzido com o pacote `gganimate`. Para observá-lo em movimento, com o perfil alterando-se em função da ordem de reação, clique no [link](#) da legenda da figura. Como acima, experimente alterar alguma variável ou comando do `chunk` para observar um resultado diferente.

## Simulação - pacote `manipulate`

Entre vários pacotes de simulação do R, o `manipulate` apresenta-se de forma simples para a simulação de equações. Diferente dos demais apreciados neste texto, o produto final só pode ser visualizado e modificado dentro do ambiente do R &

RStudio. Dessa forma, para que um estudante possa visualizar um objeto de aprendizagem, é necessária a instalação ou uso remoto da plataforma por esse.

O `manipulate` (Allaire, 2025) permite simular o comportamento de uma equação por um gráfico elaborado junto ao pacote básico `graphics` do R. A animação do pacote, contrariamente aos anteriores, é realizada por ação de *mouse* direta do usuário no objeto de aprendizagem. Isso permite ao mesmo visualizar graficamente as alterações sofridas numa expressão matemática por modificação de valores ou qualidades mediadas por um controle deslizante (*slider*), um menu de escolha (*picker*), uma caixa de checagem (*checkbox*), ou um botão (*button*).

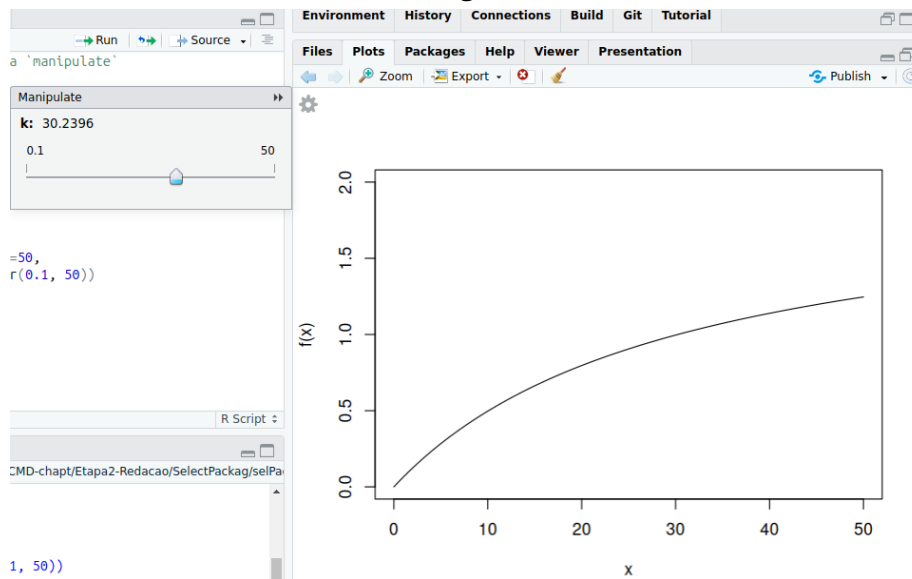
### Cinética enzimática de Michaelis-Menten

Segue um exemplo ilustrativo do pacote `manipulate` para um comportamento enzimático baseado na equação de Michaelis-Menten.

```
# 1. Instalação facultativa da biblioteca `manipulate`  
  
if (!requireNamespace("manipulate", quietly = TRUE))  
  install.packages("manipulate") # pacote de animação  
  
library(manipulate) # carregamento da biblioteca  
  
# 2. Dados da equação ( $y = V_m \cdot x / (K_m + x)$ ) - Eq. Michaelis-Menten  
Km = 1 # valor da constante de MM  
x <- 1:100 # vetor de concentração de substrato  
  
manipulate(curve(2*x/(Km+x), # aplicação do pacote  
  ylab="f(x)", from=0, to=50,  
  ylim=c(0,2)), Km = slider(0.1, 50)) # criação de controle deslizante para  
valores de Km
```

Como o `manipulate` só opera dentro do ambiente R & RStudio, não é possível apresentar seu potencial como outrora (arquivos de animação no [website Bioquanti](#)). Mas pode-se ilustrar seu efeito a partir da [Figura 4](#) que segue.

Figura 4: Recorte de tela do RStudio apresentando um gráfico de Michaelis-Mentem. O perfil é alterado em função de valores para  $K_m$  selecionados na barra deslizante à esquerda da imagem.



Dessa forma, o pacote `manipulate` permite o desenvolvimento de um objeto de aprendizagem não apenas ativo (mudança de código, mudança de produto final), como também *interativo*. E pela taxonomia de OSEL (Convertini *et al.*, 2006), um *objeto de aprendizagem interativo S-ativo (interactive LO:S-Active)*.

Um pacote mais amplo para gráficos, e com muita flexibilidade para simulações é o `plotly`, descrito a seguir.

## Gráfico interativo - pacote `plotly`

A biblioteca `plotly` é seguramente das mais utilizadas para visualização de dados de modo interativo (*interactive LO:S-Active*), quer por gráficos, como por mapas. Não está restrita ao ambiente do R, mas também expande-se à linguagem `Python` de programação. Como o pacote está baseado na biblioteca `plotly.js`, um sistema próprio para `JavaScript`, seu produto final de compilação pode ser apresentado de forma interativa num arquivo `HTML`. Isso permite ao `plotly` um vasto emprego na *web*, tanto para gráficos e mapas, como para painéis (*dashboards*) interativos.

A biblioteca permite a elaboração de um grande conjunto de gráficos com qualidade para publicação científica (Sievert, 2020), incluindo de linhas, dispersão, de área, barras, barras de erro, *boxplots*, histogramas, *heatmaps*, subplots, eixos múltiplos e gráficos 3D baseados em `WebGL` (Sievert, 2025). Objetos digitais criados com `plotly` possibilitam seu compartilhamento em arquivo `HTML` autossuficiente. Ou seja, o usuário pode interagir com o objeto, sem a necessidade da plataforma R & RStudio. Nesse caso, quando voltados ao *ensino reprodutível*, pode-se nomear tais objetos como *Objetos Interativos em Ensino Reprodutível*, ou OIER.

## Função seno

Um OIER para função seno é apresentado a seguir. O objeto final compreende o gráfico senoide junto a um botão de *PLAY* para alteração num multiplicador à equação.

```
# Instalação facultativa do pacote necessário
if (!requireNamespace("plotly", quietly = TRUE)) install.packages("plotly") #
pacote de animação

library(plotly) # carregamento do pacote

# Geração de dados
x <- seq(0, 2 * pi, length.out = 500) # definição de valores e quantidade de pontos

# Função para gerar y com base no parâmetro a
func <- function(a, x) {
  return(a * sin(x))
}

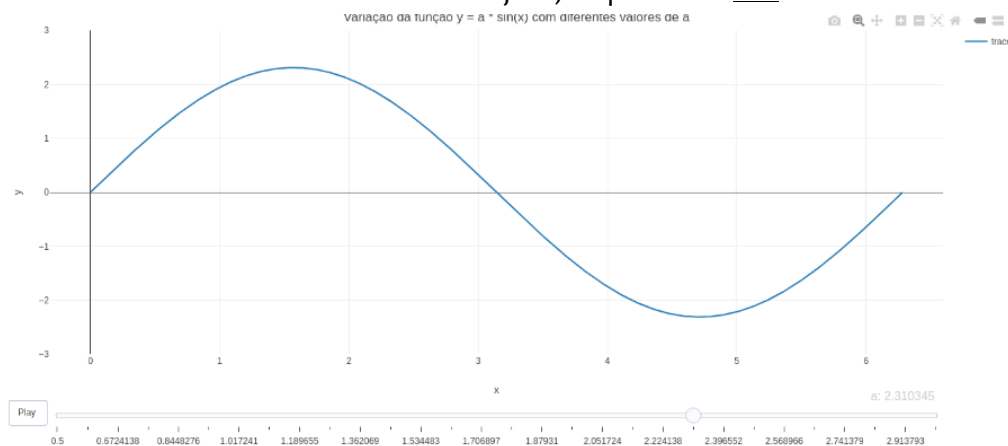
# Valores do parâmetro 'a' que irá variar para apresentação de perfis gráficos
diferentes
a_values <- seq(0.5, 3, length.out = 30)

# Criação de um data frame (planilha) com todos os valores de a e x para plotar
data <- data.frame(
  x = rep(x, times = length(a_values)),
  y = unlist(lapply(a_values, function(a) func(a, x))),
  a = rep(a_values, each = length(x))
)

# Criação do gráfico interativo com animação
plot_ly(data,
  x = ~x,
  y = ~y,
  frame = ~a, # parâmetro a variar
  type = 'scatter',
  mode = 'lines') %>%
layout(
  title = "Variação da função  $y = a * \sin(x)$  com diferentes valores de a",
  xaxis = list(title = "x"),
  yaxis = list(title = "y", range = c(-3, 3))
)
```

A [Figura 5](#) apresenta de modo estático o *OIER* produzido pelo trecho de código acima. Novamente, clique no *link* para observar sua interatividade.

Figura 5: *OIER* produzido com a função seno:  $y = a * \text{seno}(x)$ . Para observar o potencial de interatividade do objeto, clique neste [link](#)



## Mapa interativo - pacote leaflet

O pacote [leaflet](#) é bastante conhecido para a criação de mapas interativos. Também [abrange a biblioteca JavaScript além do pacote htmlwidgets](#), permitindo uma interatividade direta sobre o objeto construído. O leaflet é bem sofisticado (Cheng; Karambelkar; Xie, 2025), permitindo a inserção de *pop-ups*, dicas flutuantes, seleções interativas, marcadores com distintos formatos, visualização geoespacial, e inserção de camadas de mapas-base (*Google Maps*, *OpenStreetMap*, por ex).

## Mapa-múndi\* e algumas cidades

O trecho de código que segue ilustra o emprego do pacote leaflet para uma representação interativa do mapa-múndi, e com visualização iniciada nas coordenadas do Brasil.

```
if(!requireNamespace("leaflet", quietly = TRUE))
install.packages("leaflet") # instalação facultativa

library(leaflet)
leaflet() %>% # carrega-se o pacote
  addTiles() %>% # adiciona-se as "peças" (tiles)
  setView(lng=-56.0949, lat=-15.5989, zoom = 4) %>% # adiciona-se as
coordenadas desejadas e o nível de ampliação
  addMarkers(lng=-45.950139, lat=-21.425917, # adiciona-se um marcador
  popup="<b>Aqui fica a UNIFAL-MG !</b><br><a
href='https://www.washingtonpost.com'>-Me</a>") # adiciona-se uma
mensagem ao marcador
```

Figura 6: Mapa-múndi com visualização inicial em coordenadas do Brasil, tal como criado por pacote plotly. Perceba sua interatividade clicando neste [link](#).



## A elaboração do capítulo

A formatação de texto (*RMarkdown*), figuras, códigos, gerenciamento bibliográfico, e compilação para documento dinâmico como arquivo DOCX, foram conduzidos com R (versão 4.3.3, fev/2024) em ambiente *RStudio* (versão 2024.09.1 Build 394), e pacote *quarto* (versão 1.4.4). Outros pacotes mencionados no texto foram utilizados para ilustrar os OIER.

## Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

## Referências

ALLAIRE, J. **manipulate: Interactive Plot Manipulation**. [S. l.: s. n.], 2025. Disponível em: <https://cran.r-project.org/web/packages/manipulate/manipulate.pdf>. Acedido em: 24 jan. 2025.

BEAN, B. L. Teaching Reproducibility to First Year College Students: Reflections From an Introductory Data Science Course. **Journal on Empowering Teaching Excellence**, [s. l.], vol. 7, n.º 2, p. 5, 2023.

CHENG, J.; KARAMBELKAR, B.; XIE, Y. **leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library**. [S. l.: s. n.], 2025. Disponível em: <https://cran.r-project.org/web/packages/leaflet/index.html>. Acedido em: 24 jan. 2025.

CONVERTINI, V. *et al.* The OSEL taxonomy for the classification of learning objects. **Interdisciplinary Journal of E-Learning and Learning Objects**, [s. l.], vol. 2, n.º 1, p. 125–138, 2006.

DOGUCU, M. Reproducibility in the Classroom. **Annual Review of Statistics and Its Application**, [s. l.], vol. 12, 2024.

HUGH-JONES, D. **Package 'anim.plots': Simple Animated Plots for R**. [S. l.: s. n.], 2022. Disponível em: <https://cran.r-project.org/web/packages/anim.plots/anim.plots.pdf>. X

KNUTH, D. E. Literate programming. **The computer journal**, [s. l.], vol. 27, n.º 2, p. 97–111, 1984.

OSTBLOM, J.; TIMBERS, T. Opinionated practices for teaching reproducibility: motivation, guided instruction and practice. **Journal of Statistics and Data Science Education**, [s. l.], vol. 30, n.º 3, p. 241–250, 2022.

PEDERSE, T. L. **gganimate: A Grammar of Animated Graphics**. [S. l.: s. n.], 2025. Disponível em: <https://cran.r-project.org/web/packages/gganimate/gganimate.pdf>. Acedido em: 24 jan. 2025.

R CORE TEAM. **R: A Language and Environment for Statistical Computing**. Vienna, Austria: R Foundation for Statistical Computing, 2023. Disponível em: <https://www.R-project.org/>. X

RSTUDIO TEAM. **RStudio: Integrated Development Environment for R**. Boston, MA: Posit Software, PBC, 2023. Disponível em: <https://posit.co/products/open-source/rstudio/>. X

SCHNEEDORF, J. M. Bioquanti, um website interativo para ensino-aprendizagem em Bioquímica. **Revista de Ensino de Bioquímica**, [s. l.], vol. 21, n.º 1, p. 110–124, 2023.

SIEVERT, C. **Interactive web-based data visualization with R, plotly, and shiny**. [S. l.]: Chapman; Hall/CRC, 2020.

SIEVERT, C. **plotly: Create Interactive Web Graphics via 'plotly.js'**. [S. l.: s. n.], 2025. Disponível em: <https://cran.r-project.org/web/packages/plotly/plotly.pdf>. Acedido em: 24 jan. 2025.

WILKINSON, L. **The grammar of graphics**. [S. l.]: Springer, 2012.